

var@ice_clear_vsi_promisc

remove_list_head

var@ice_remove_vsi_lkup_fltr

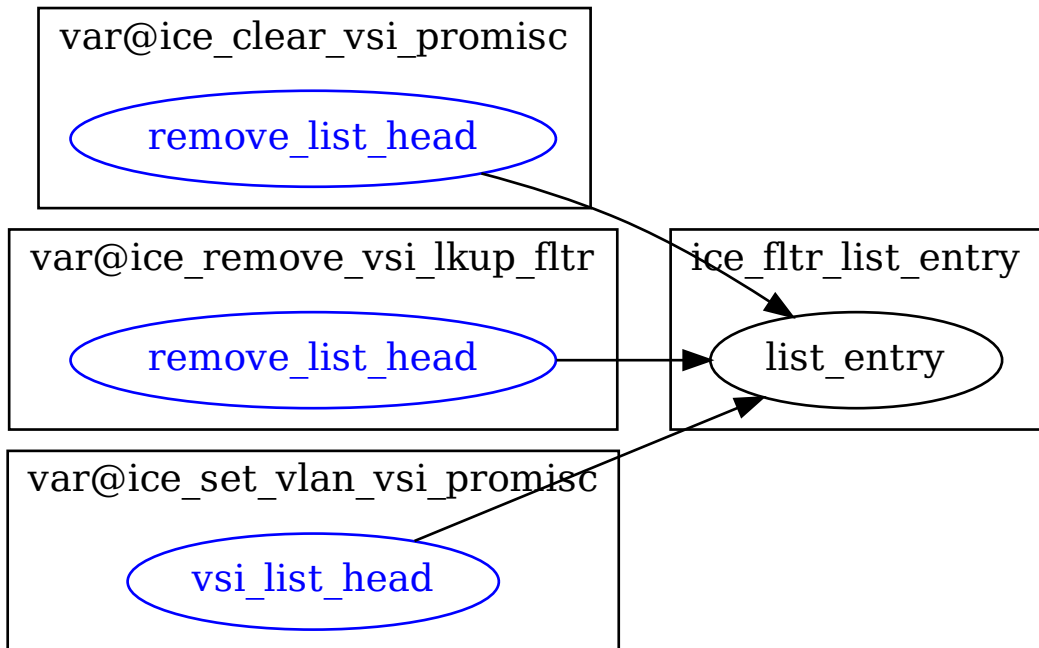
remove_list_head

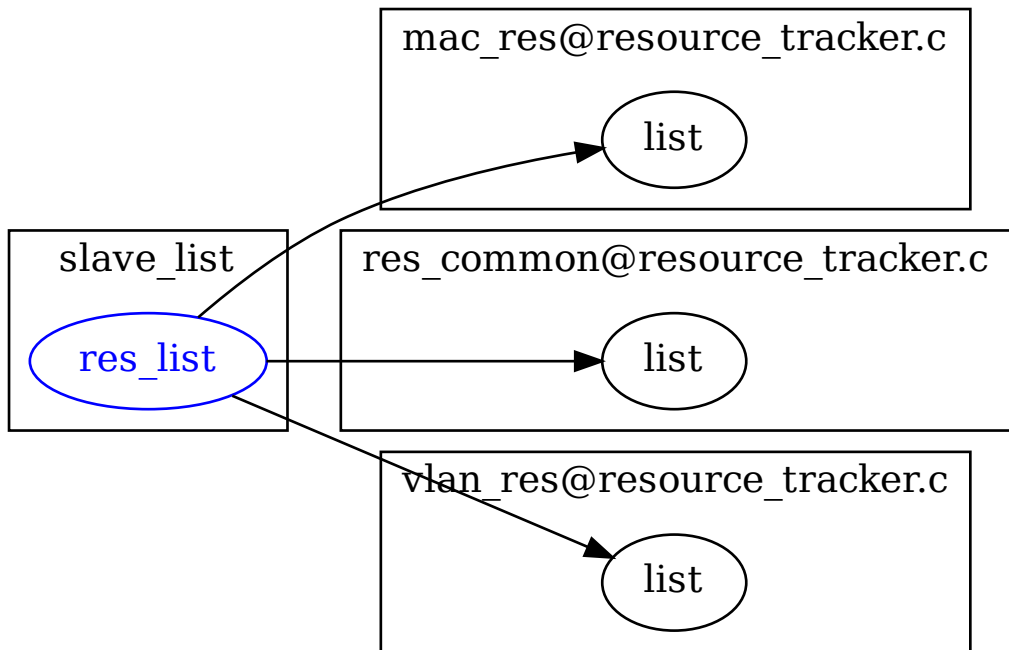
var@ice_set_vlan_vsi_promisc

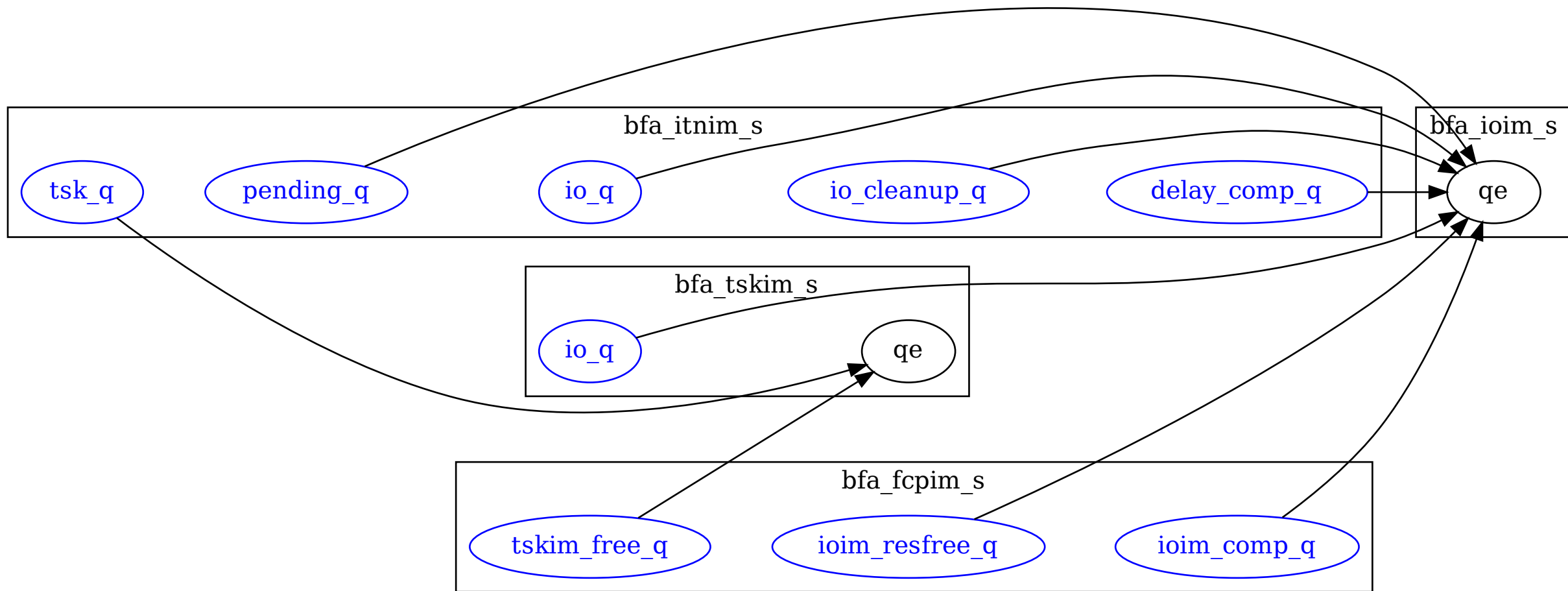
vsi_list_head

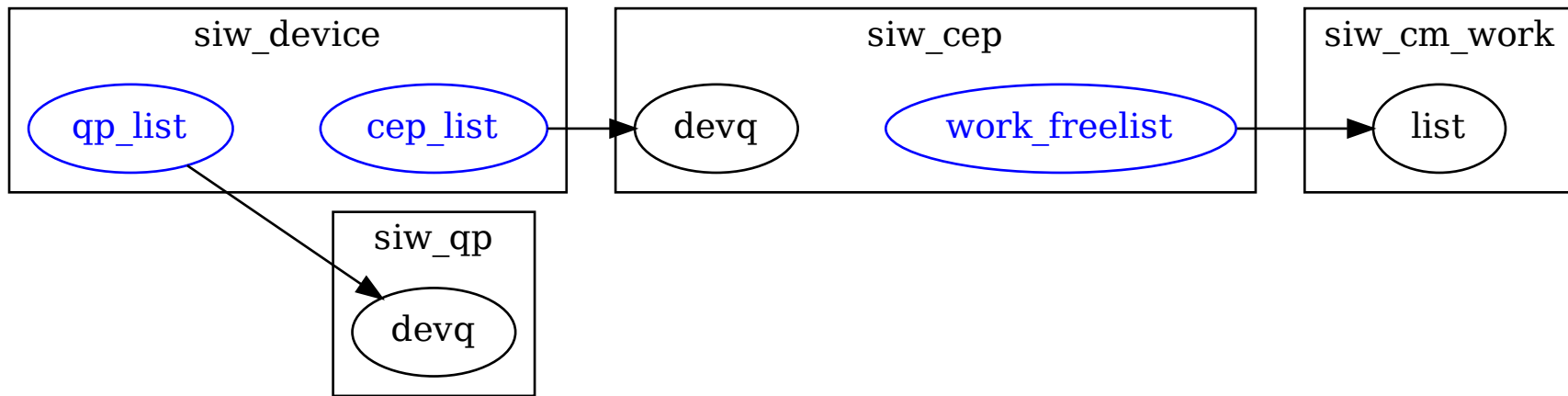
ice_fltr_list_entry

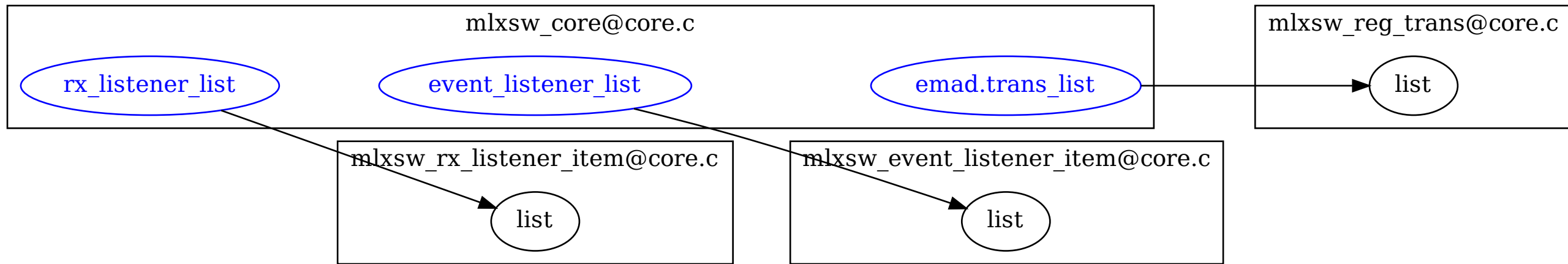
list_entry

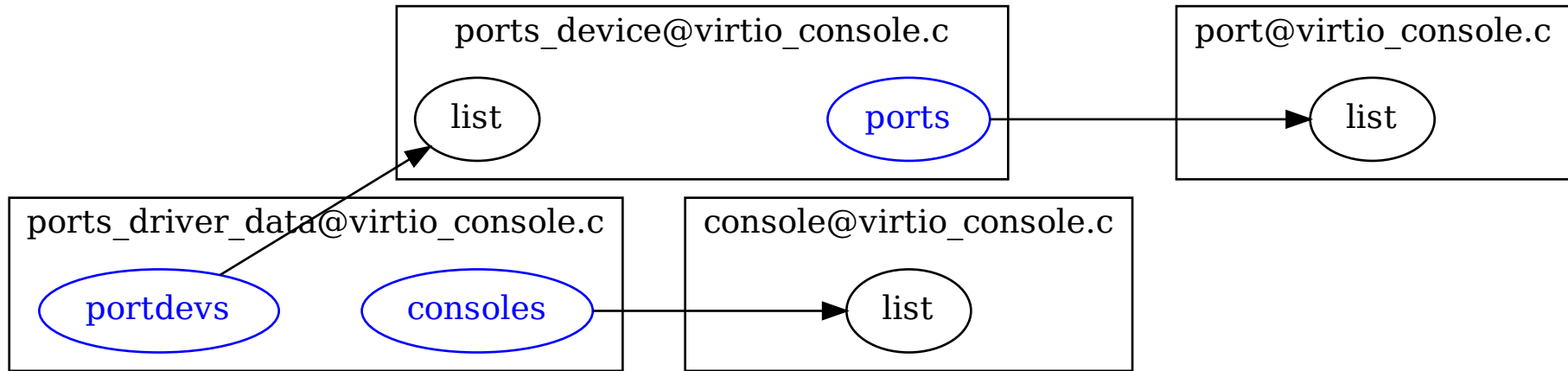


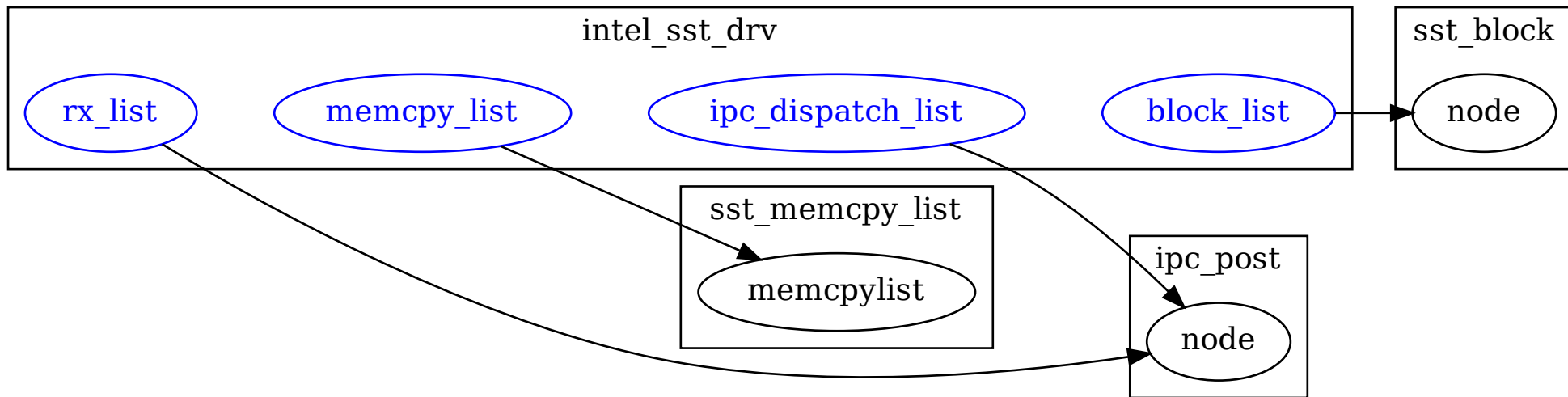


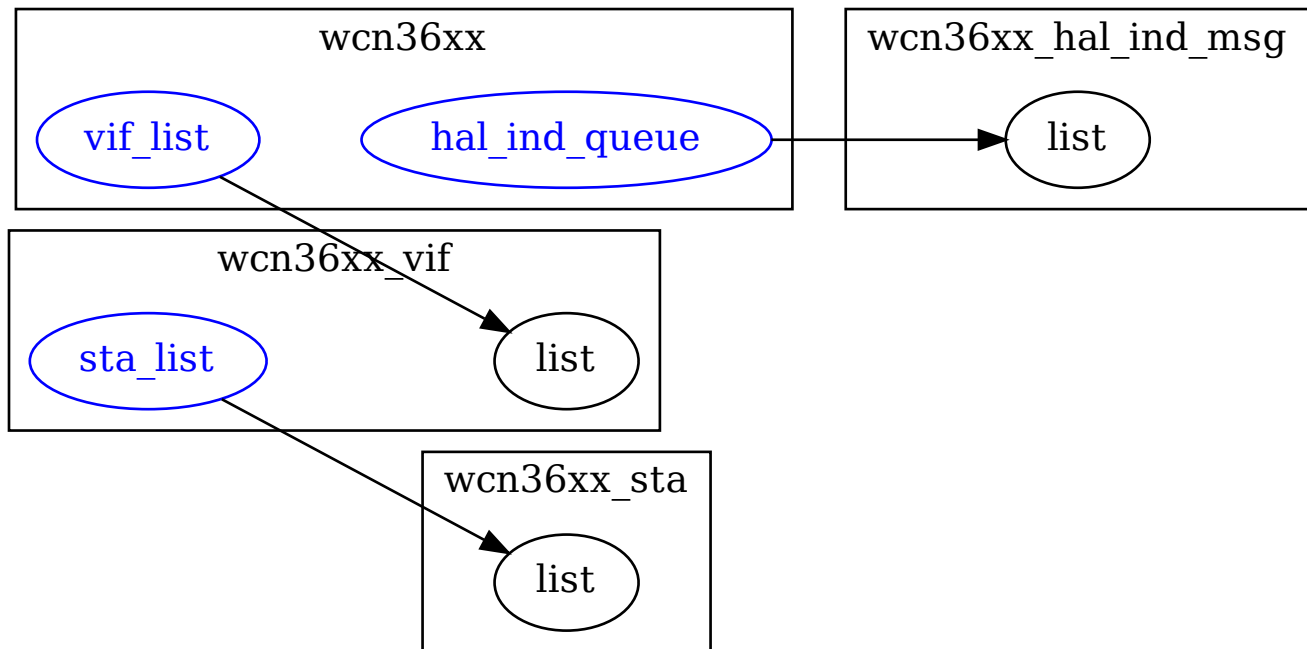


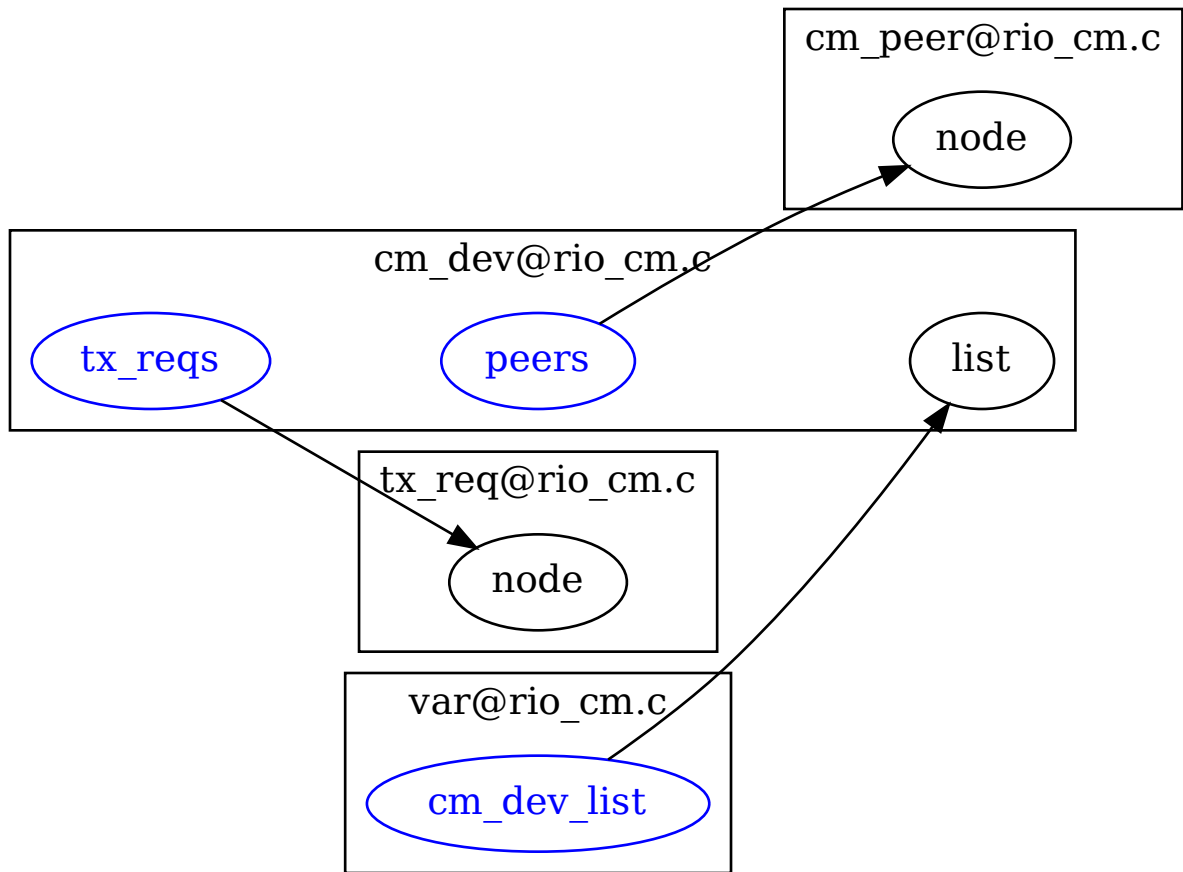


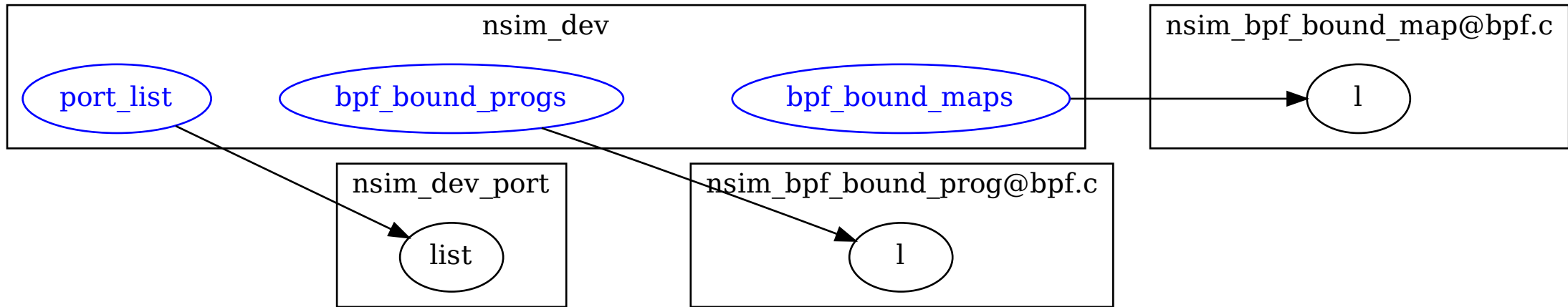


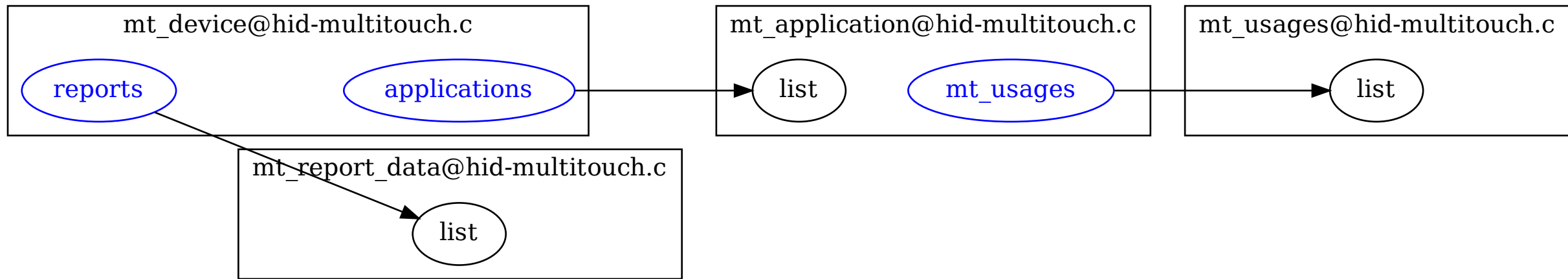


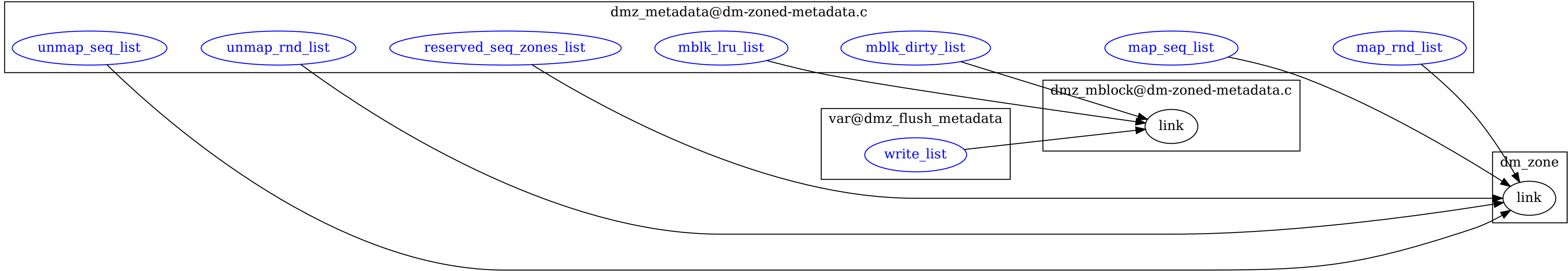


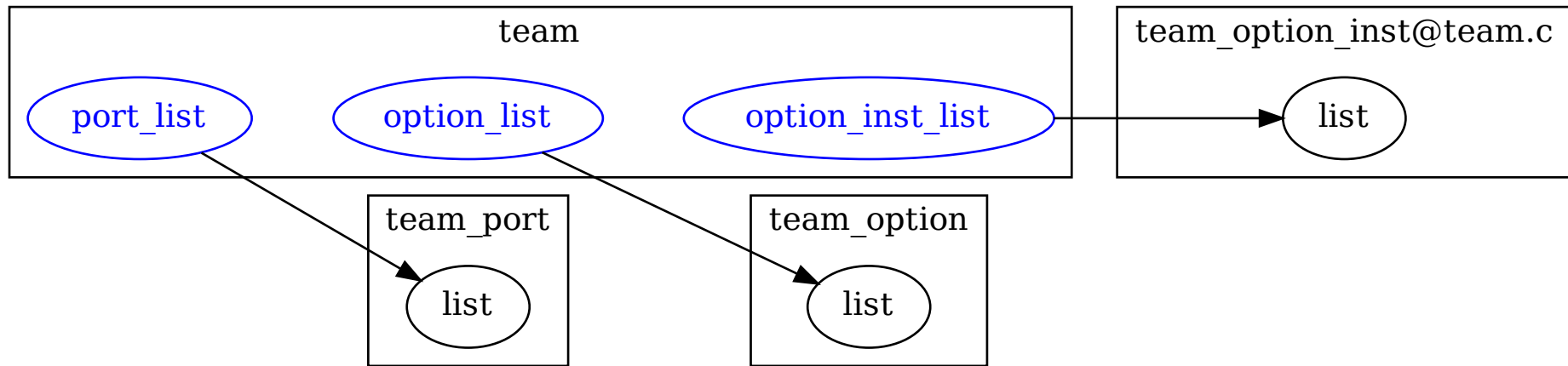


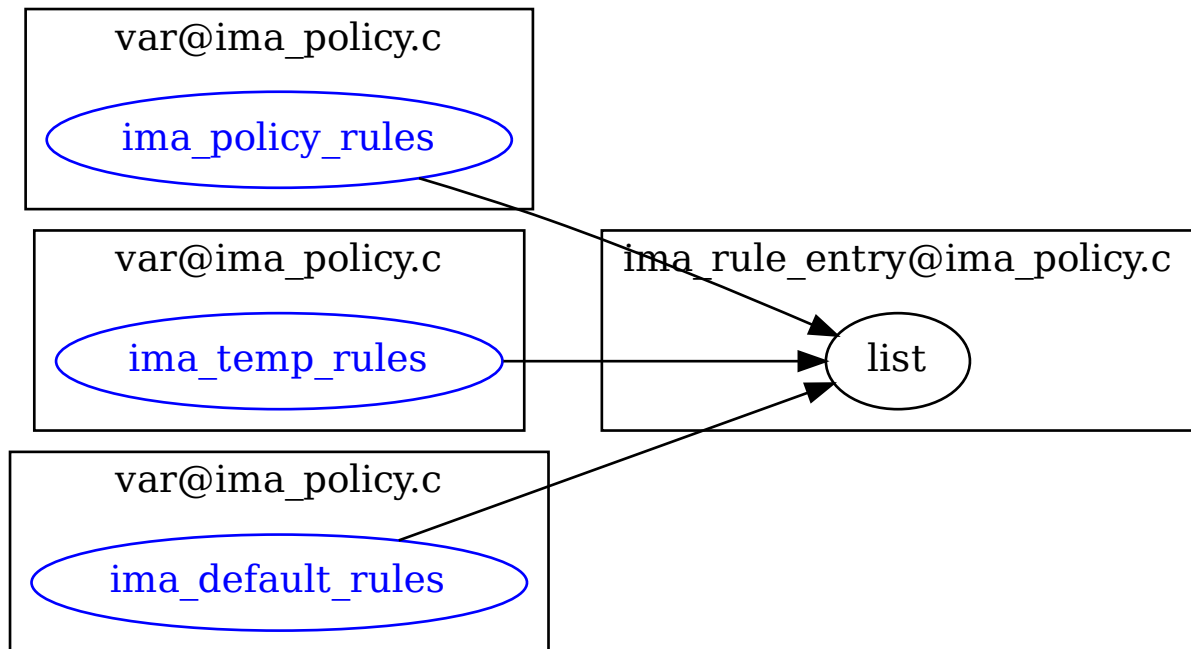


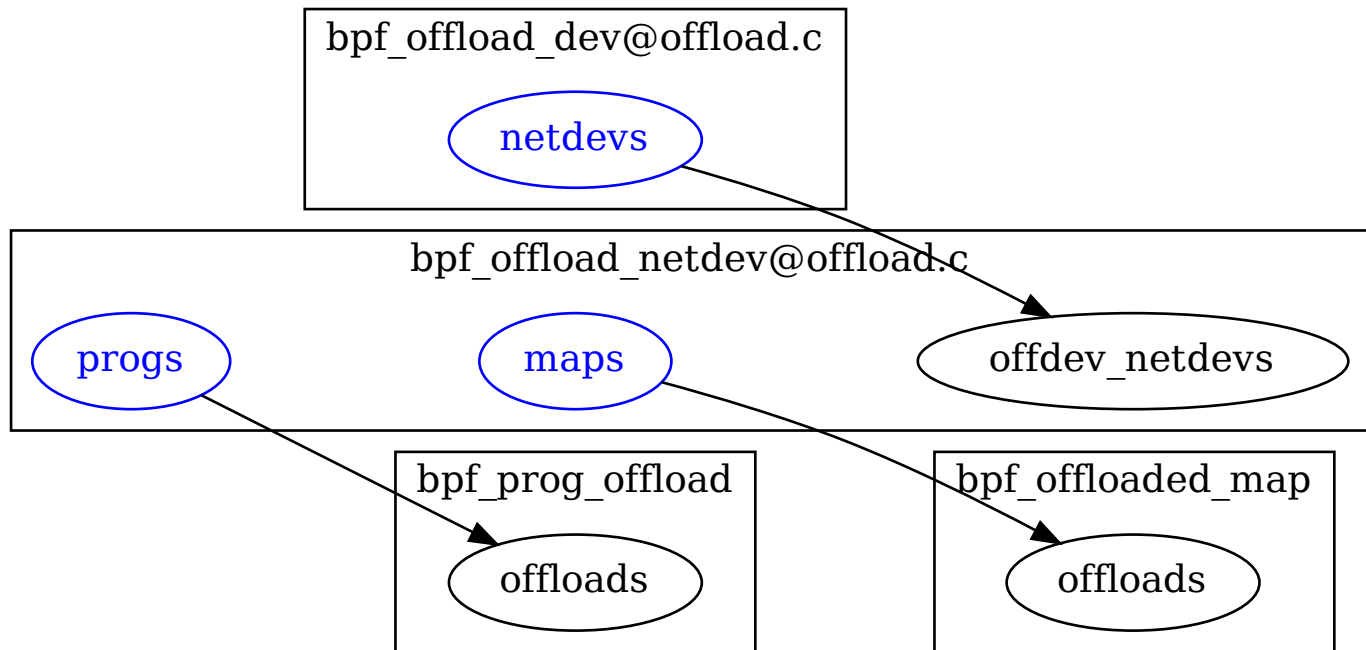


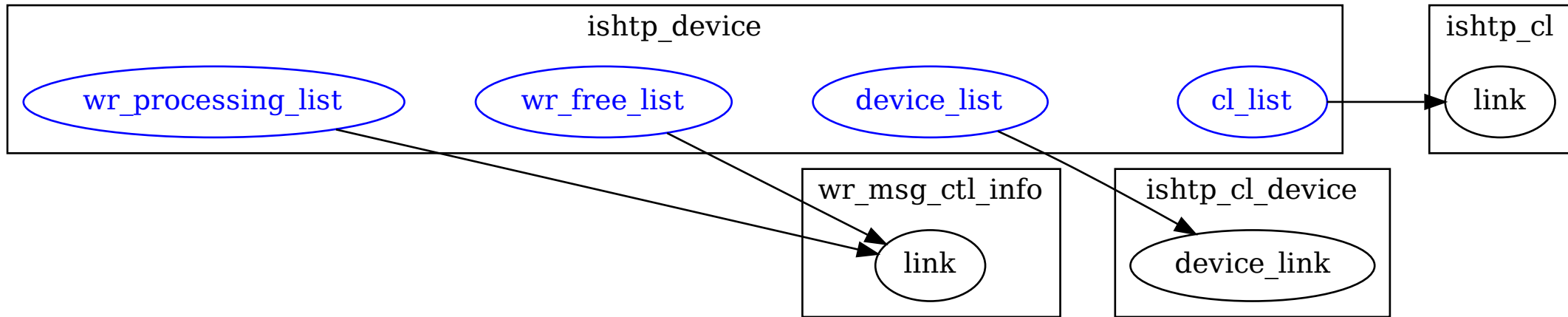


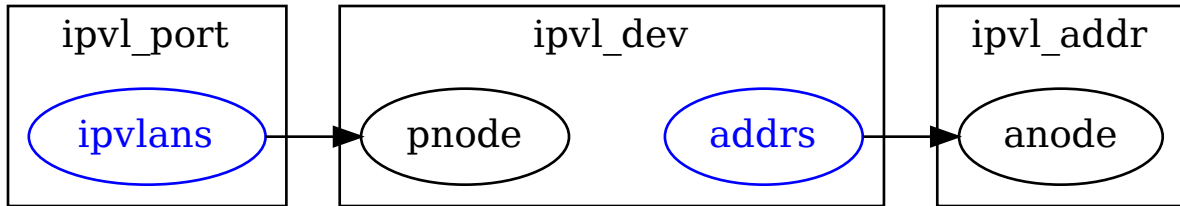












var@mxl5xx.c

mxllist

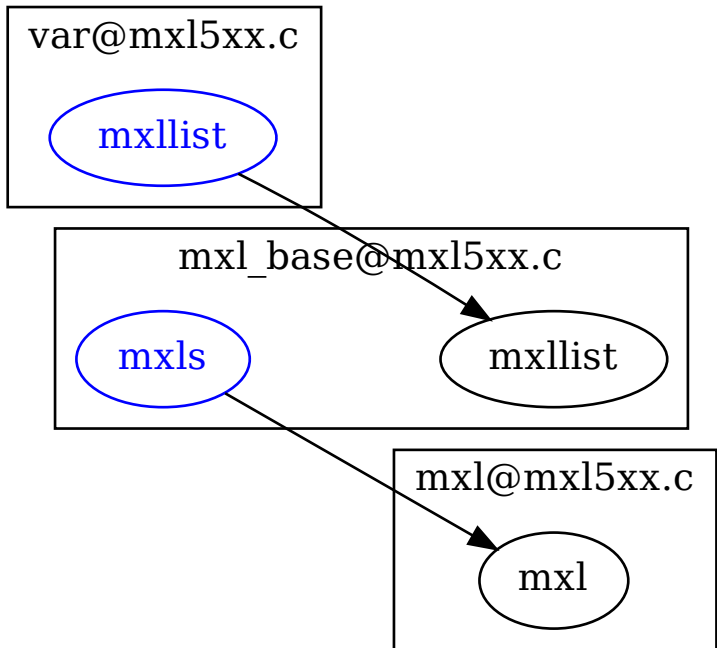
mxl_base@mxl5xx.c

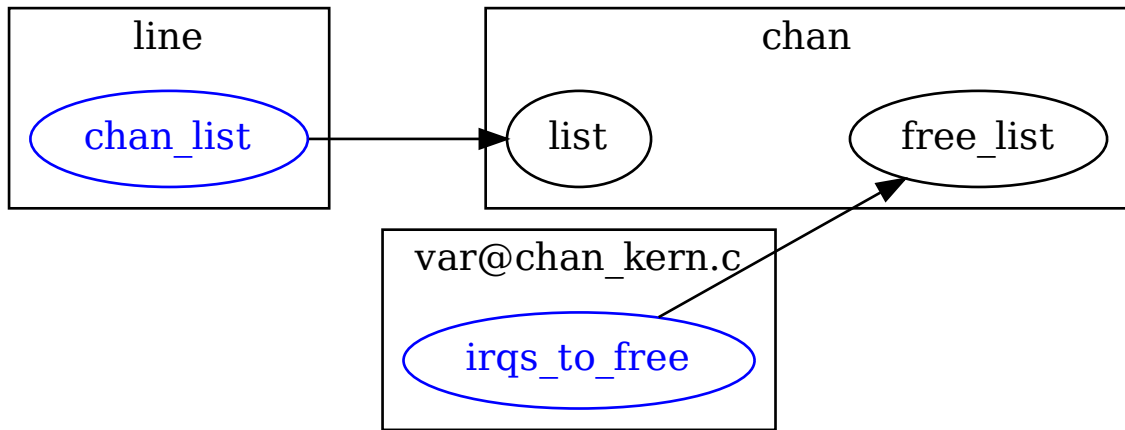
mxls

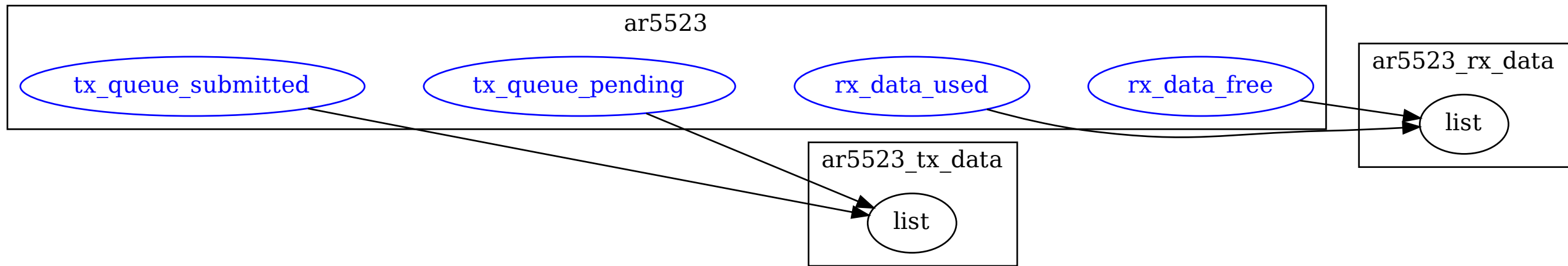
mxllist

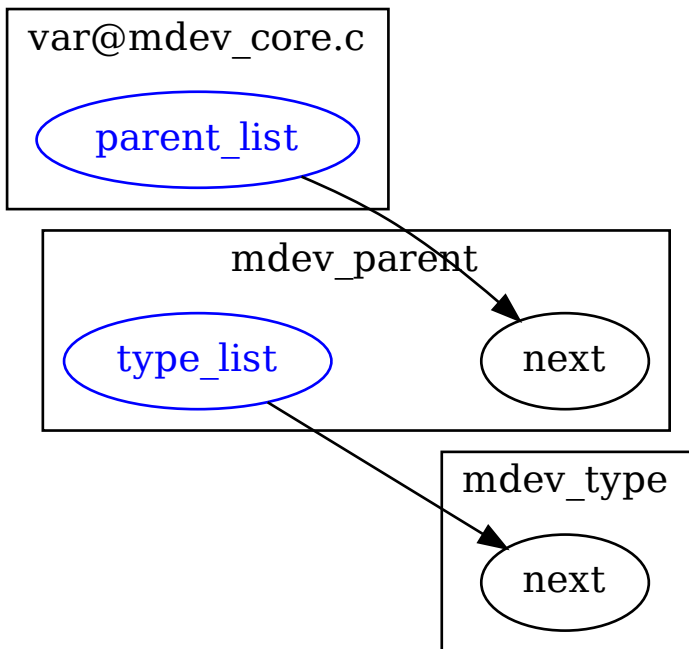
mxl@mxl5xx.c

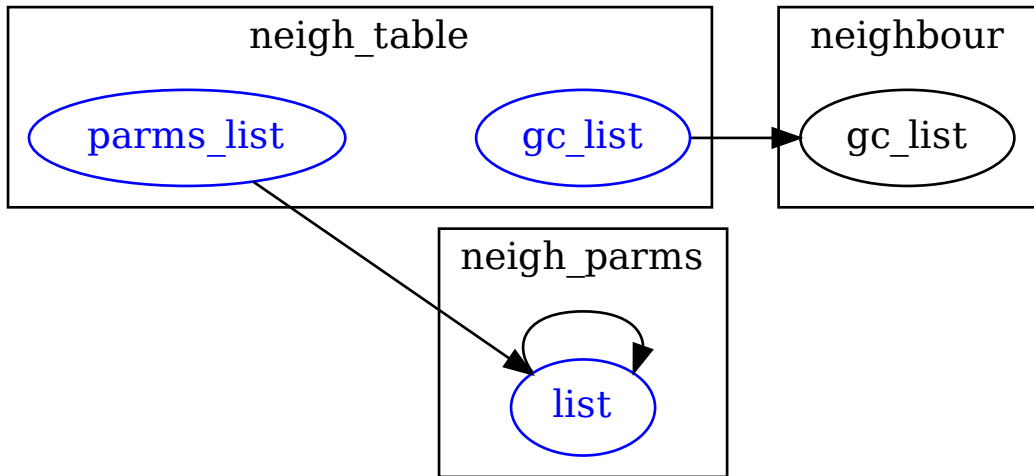
mxl

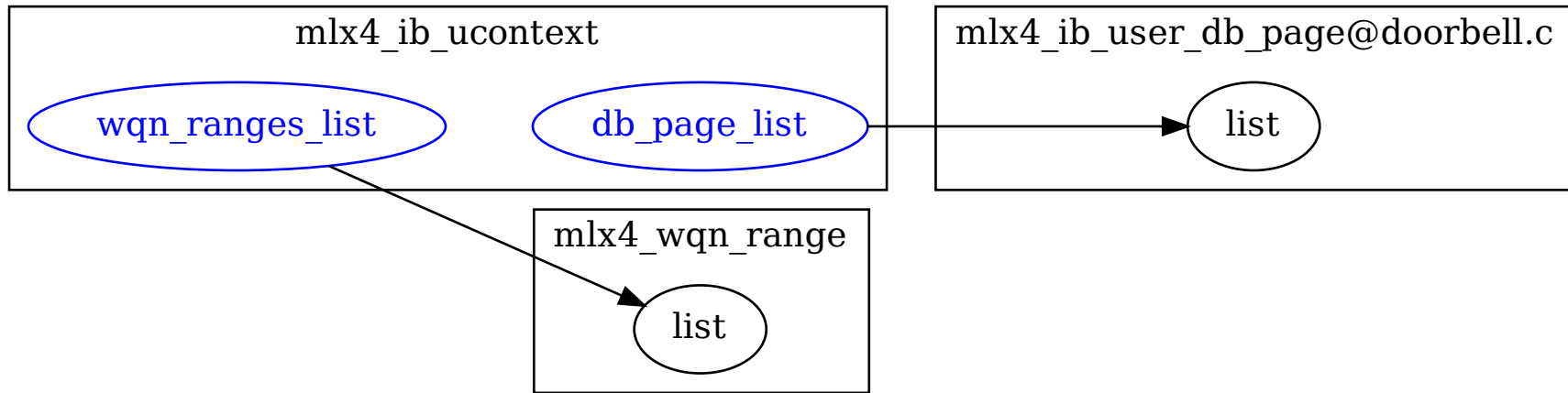


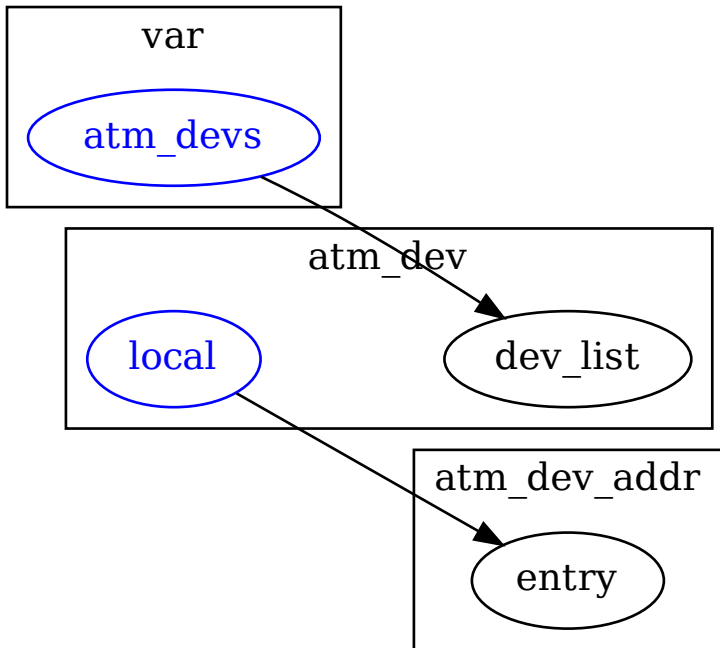


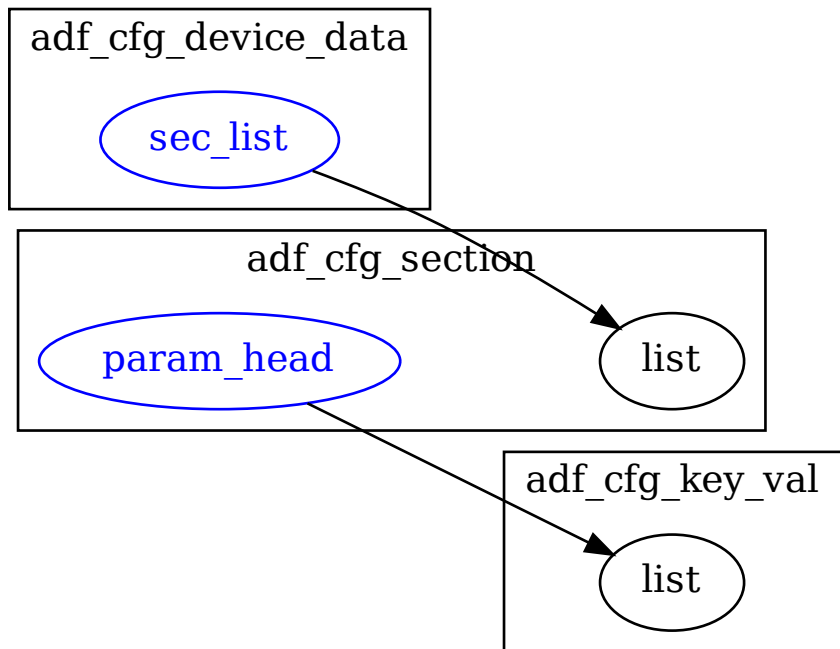


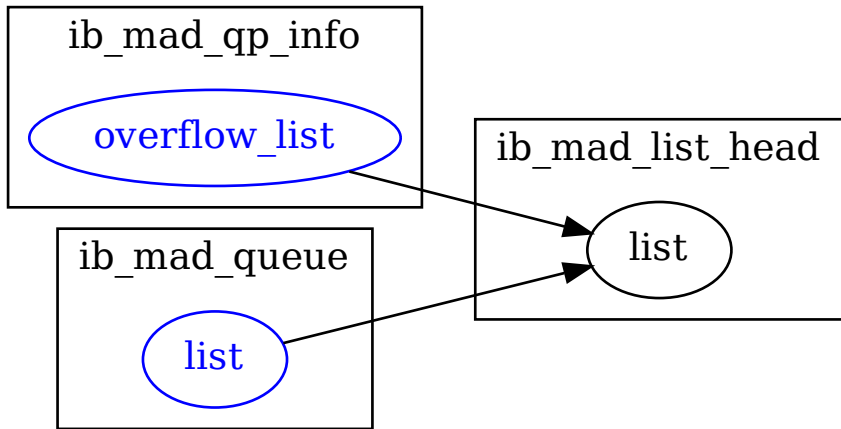


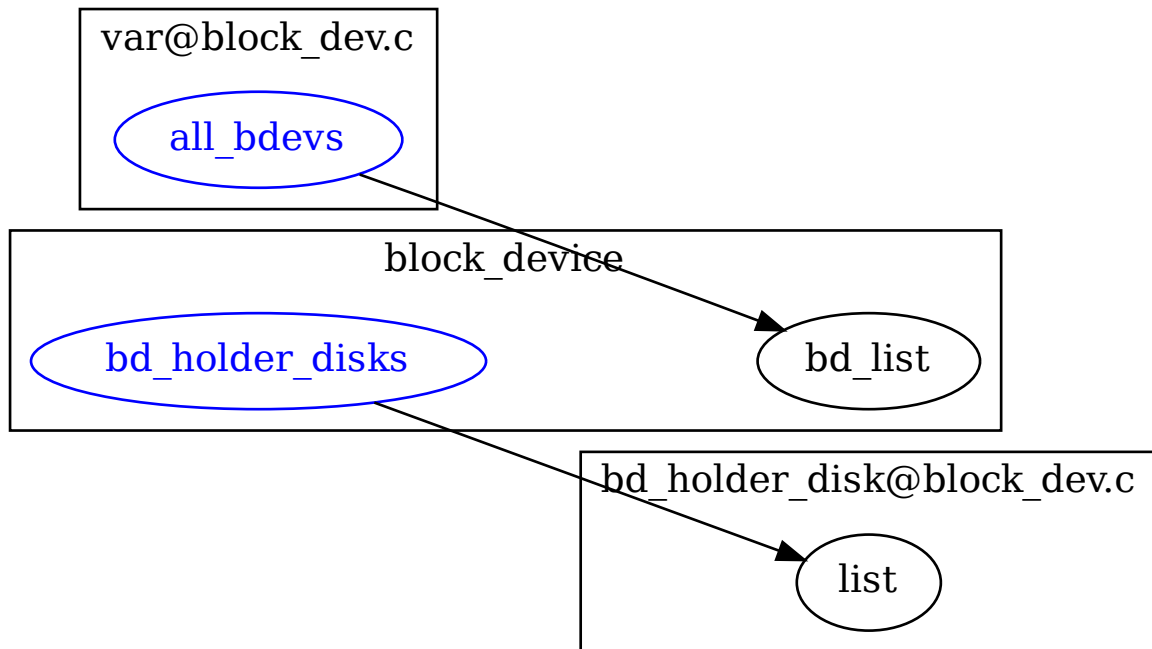


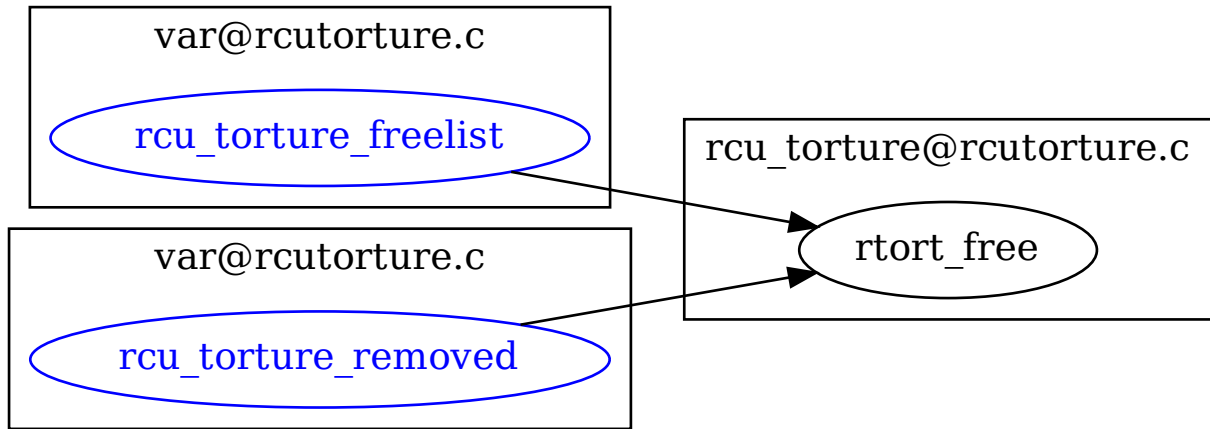












xfs_mru_cache@xfs_mru_cache.c

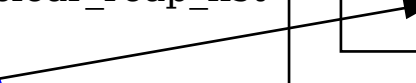
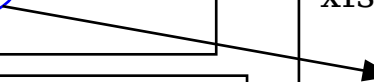
reap_list

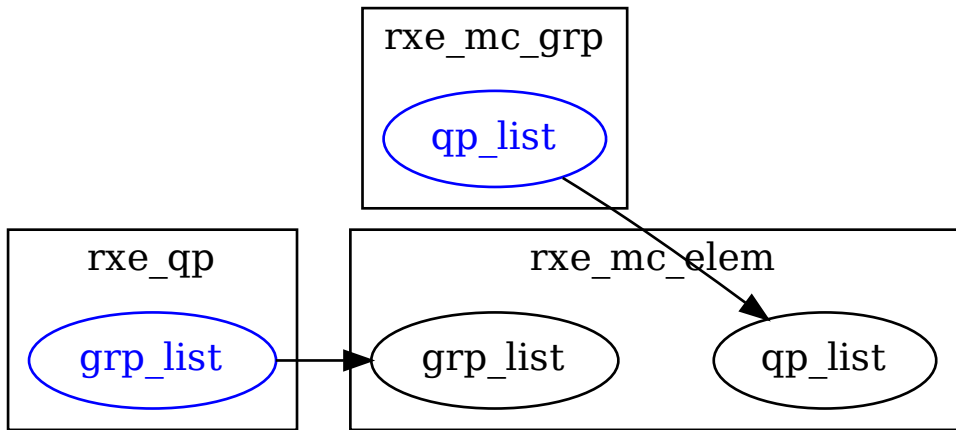
var@_xfs_mru_cache_clear_reap_list

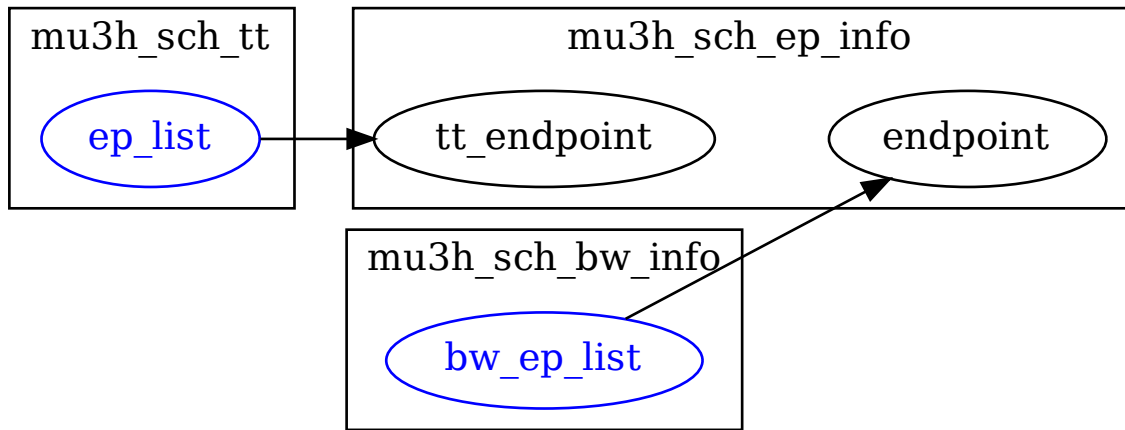
tmp

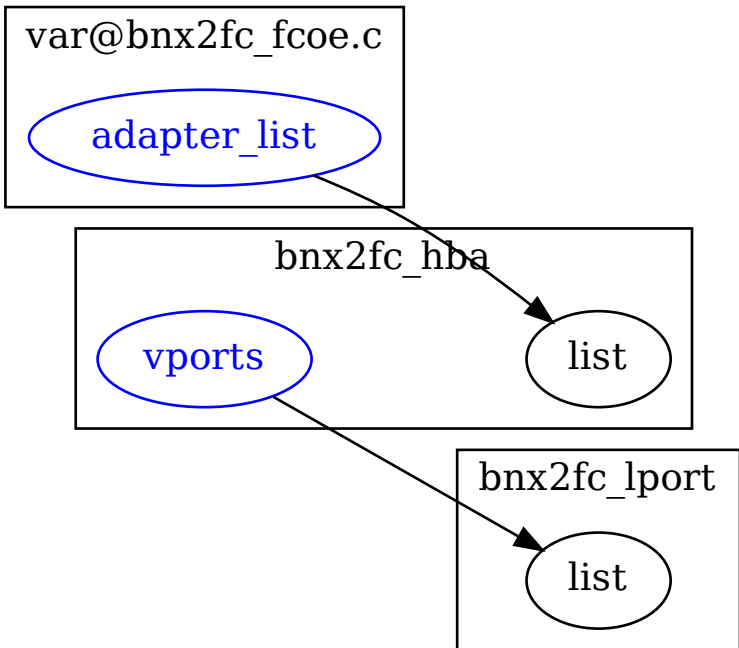
xfs_mru_cache_elem

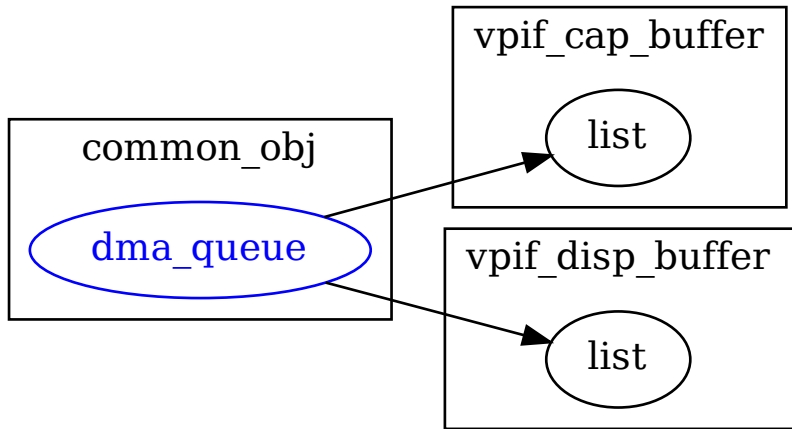
list_node

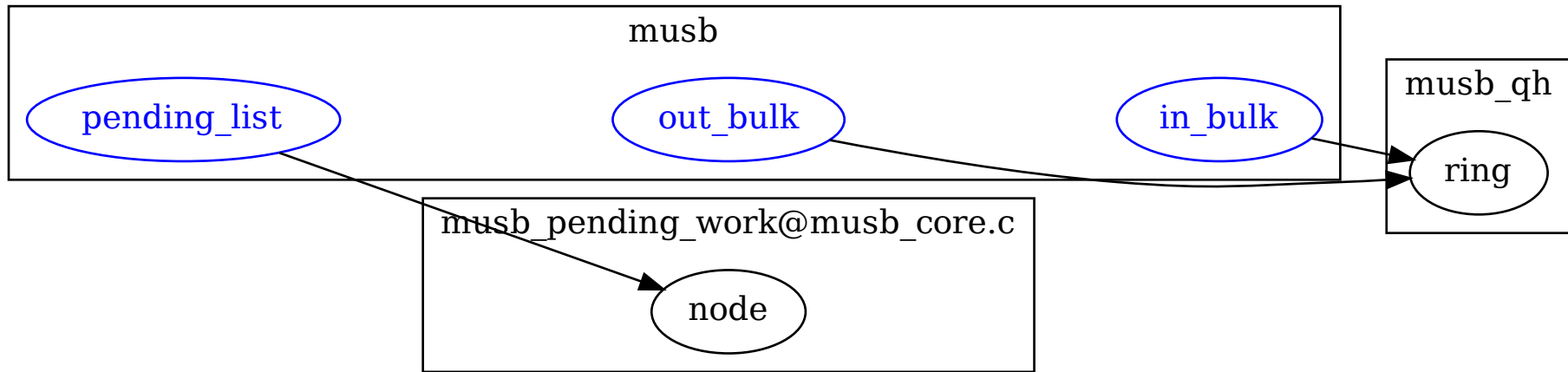


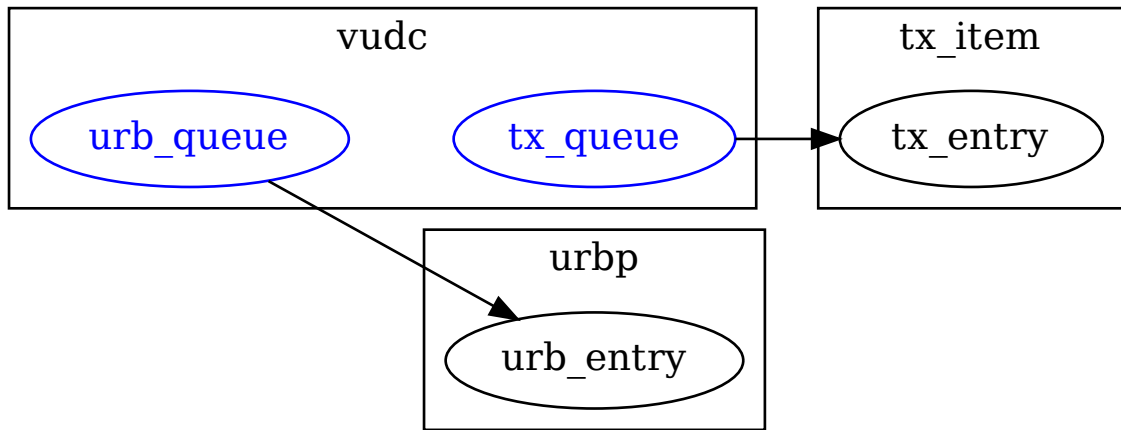


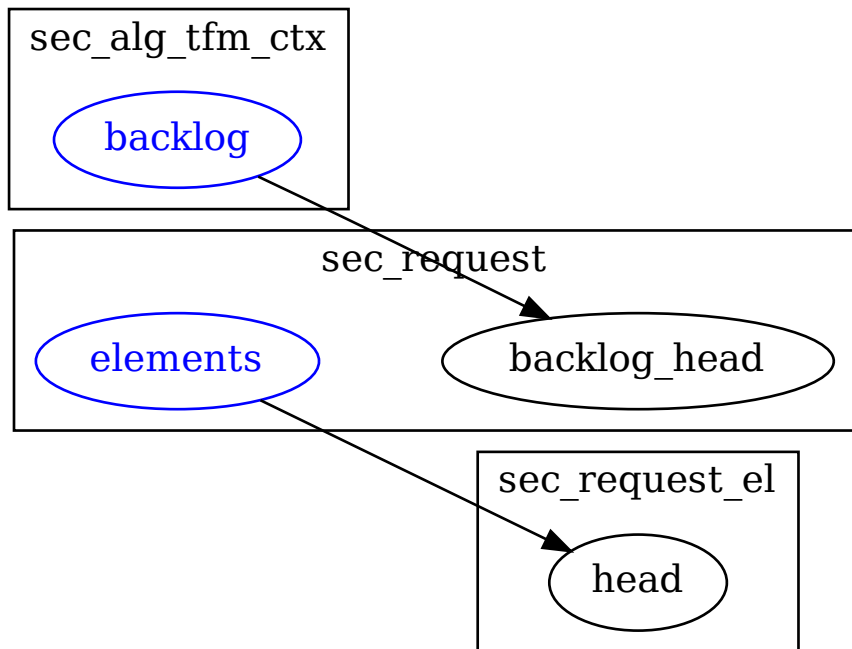


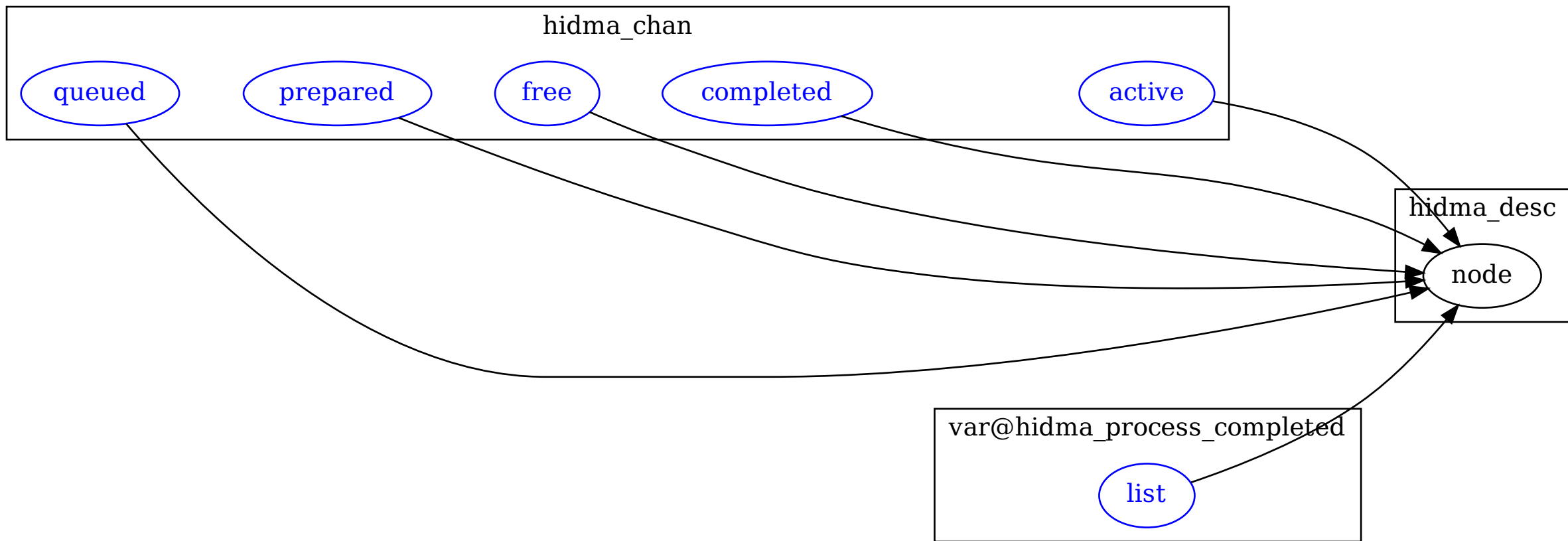


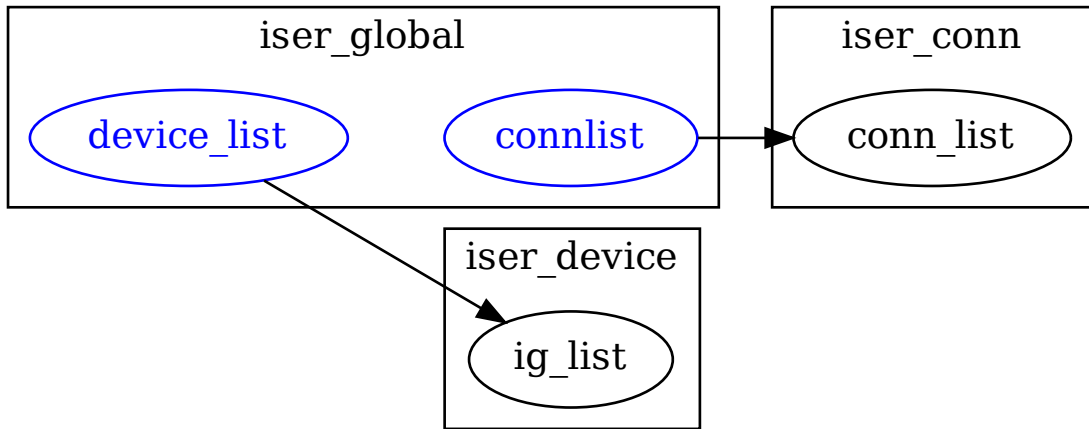


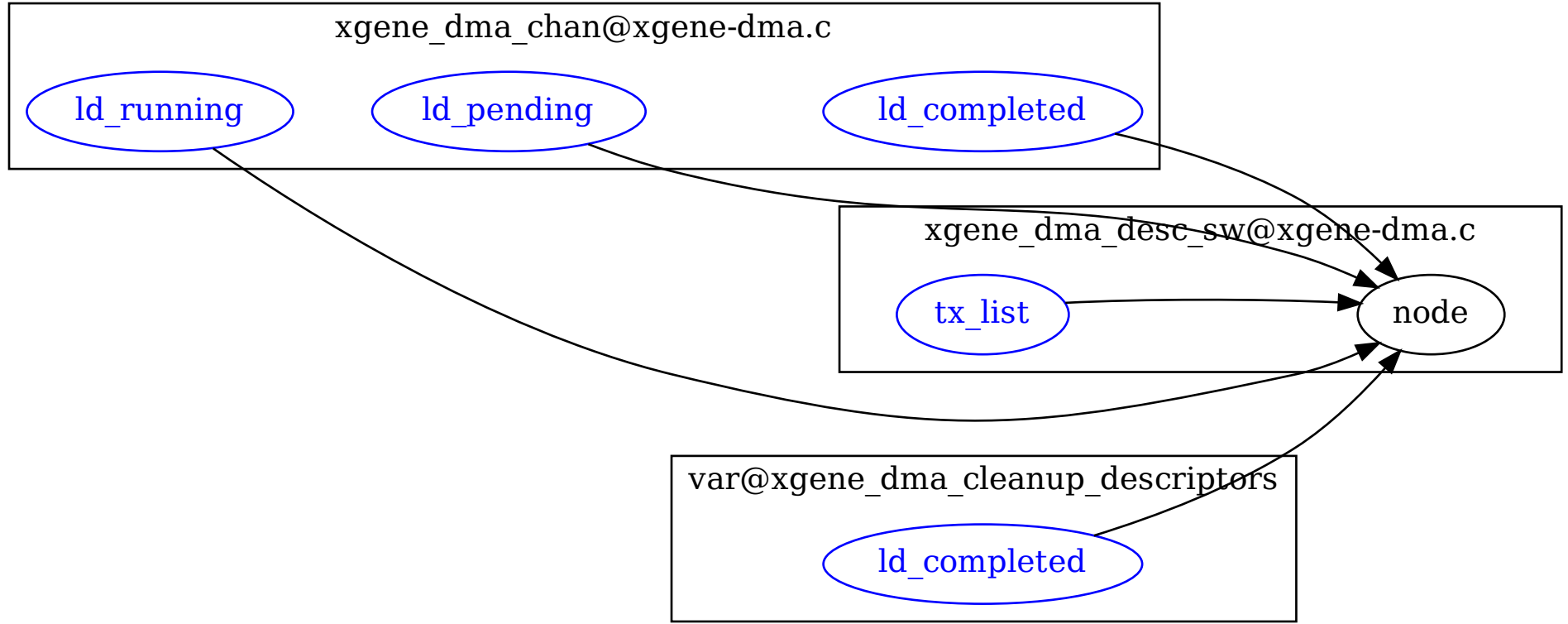


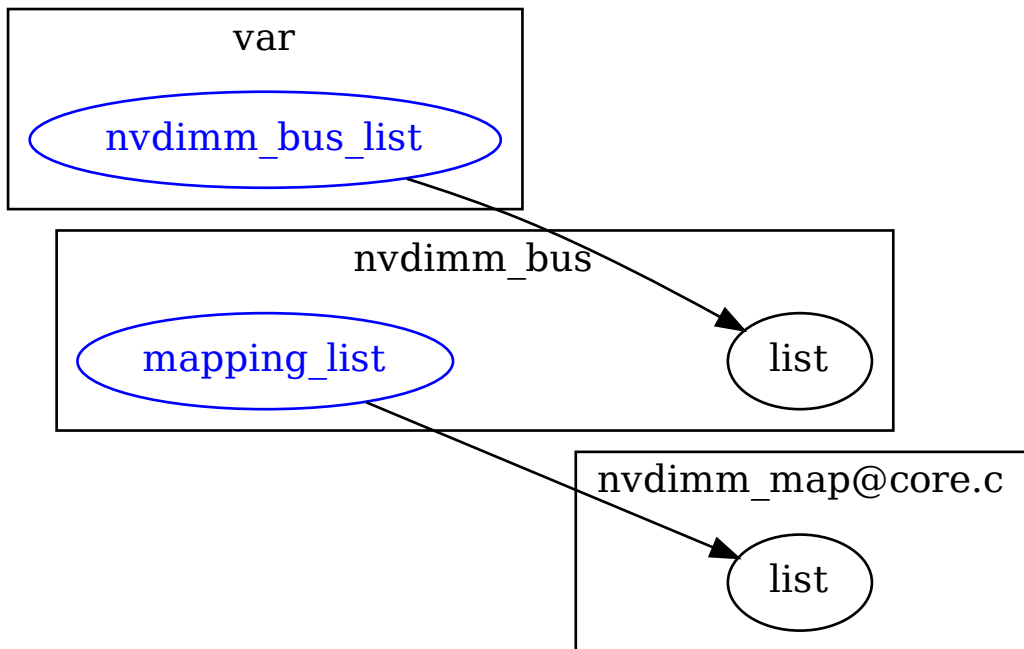


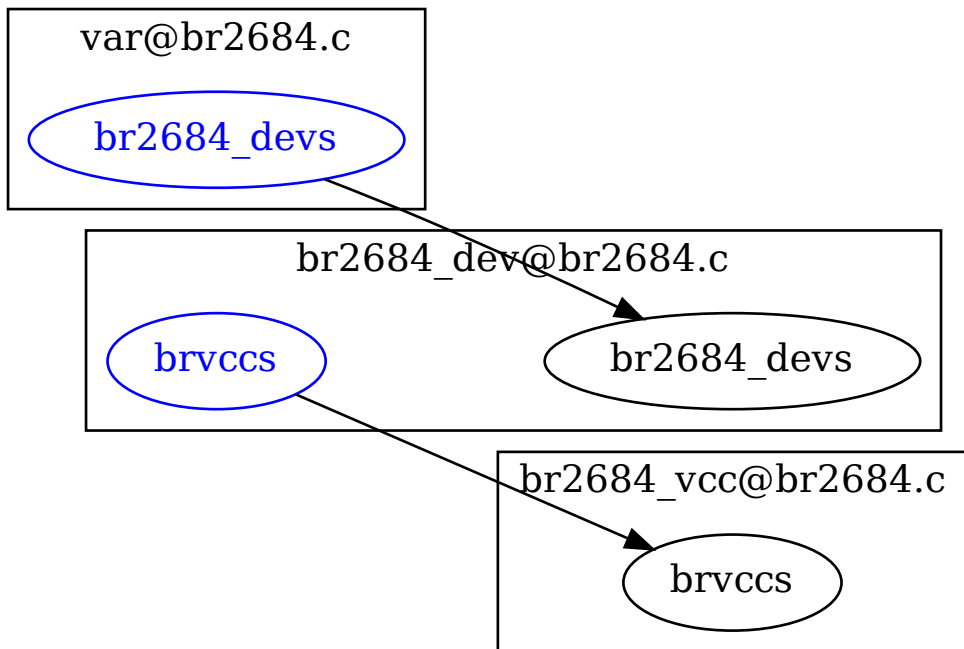


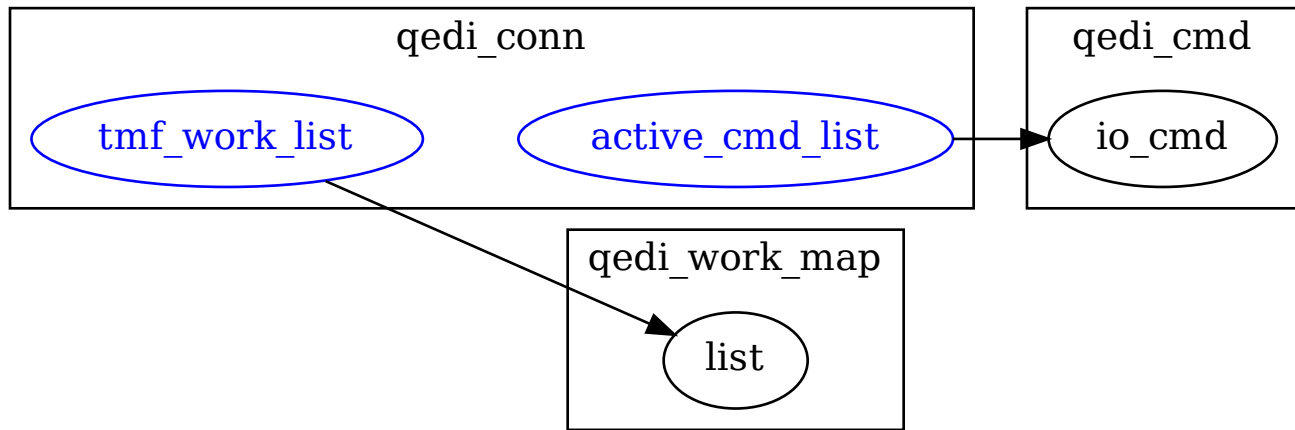


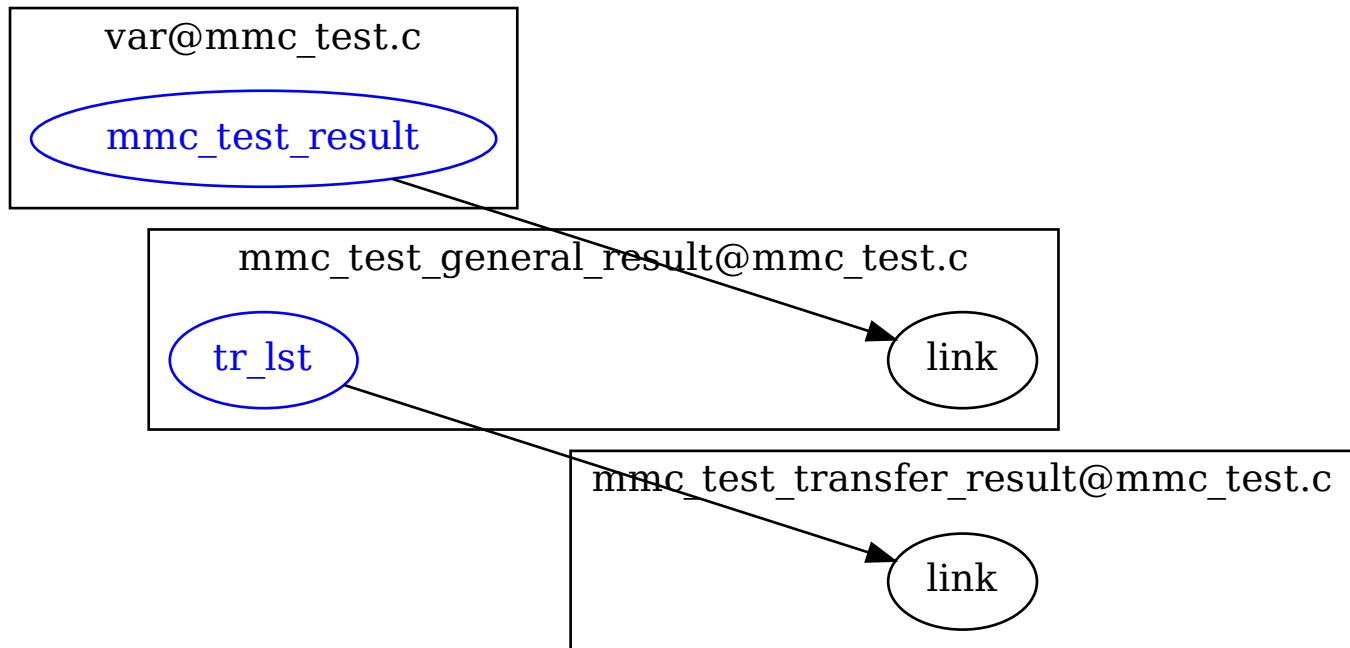


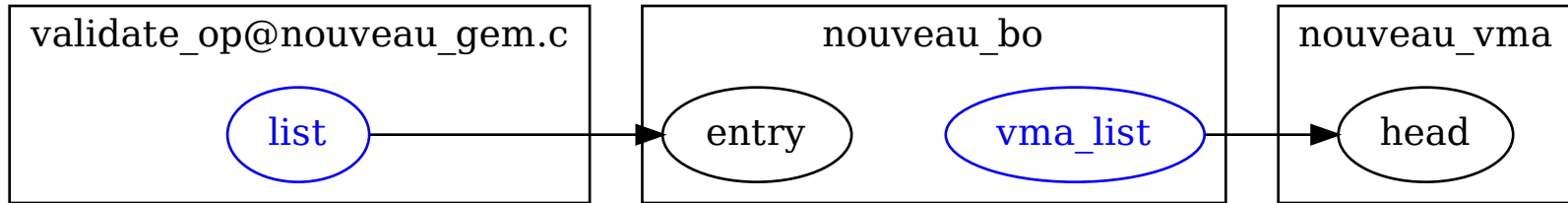


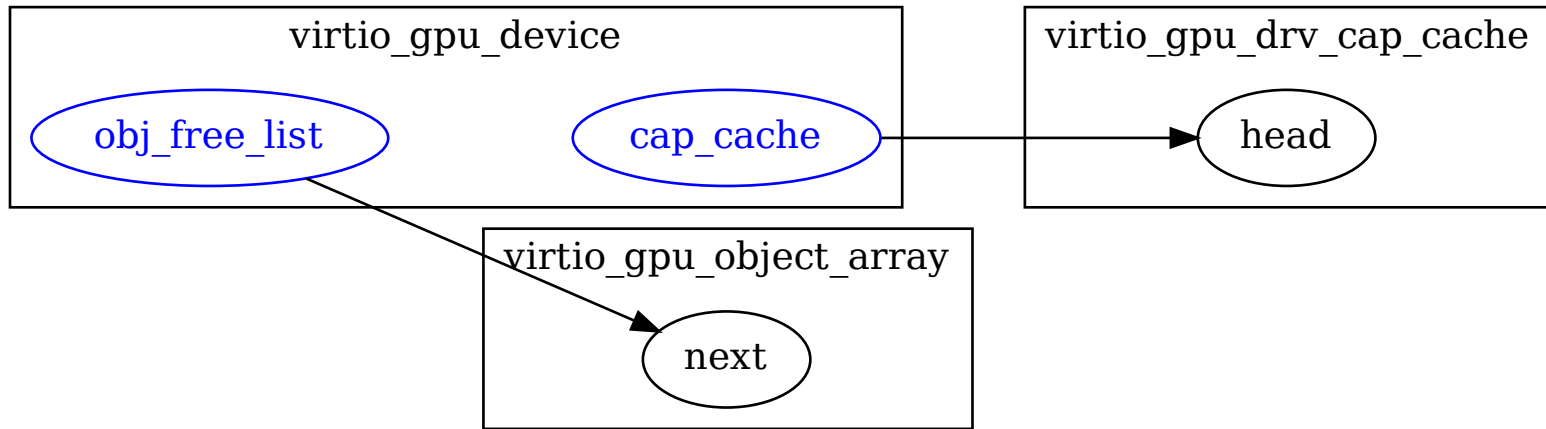


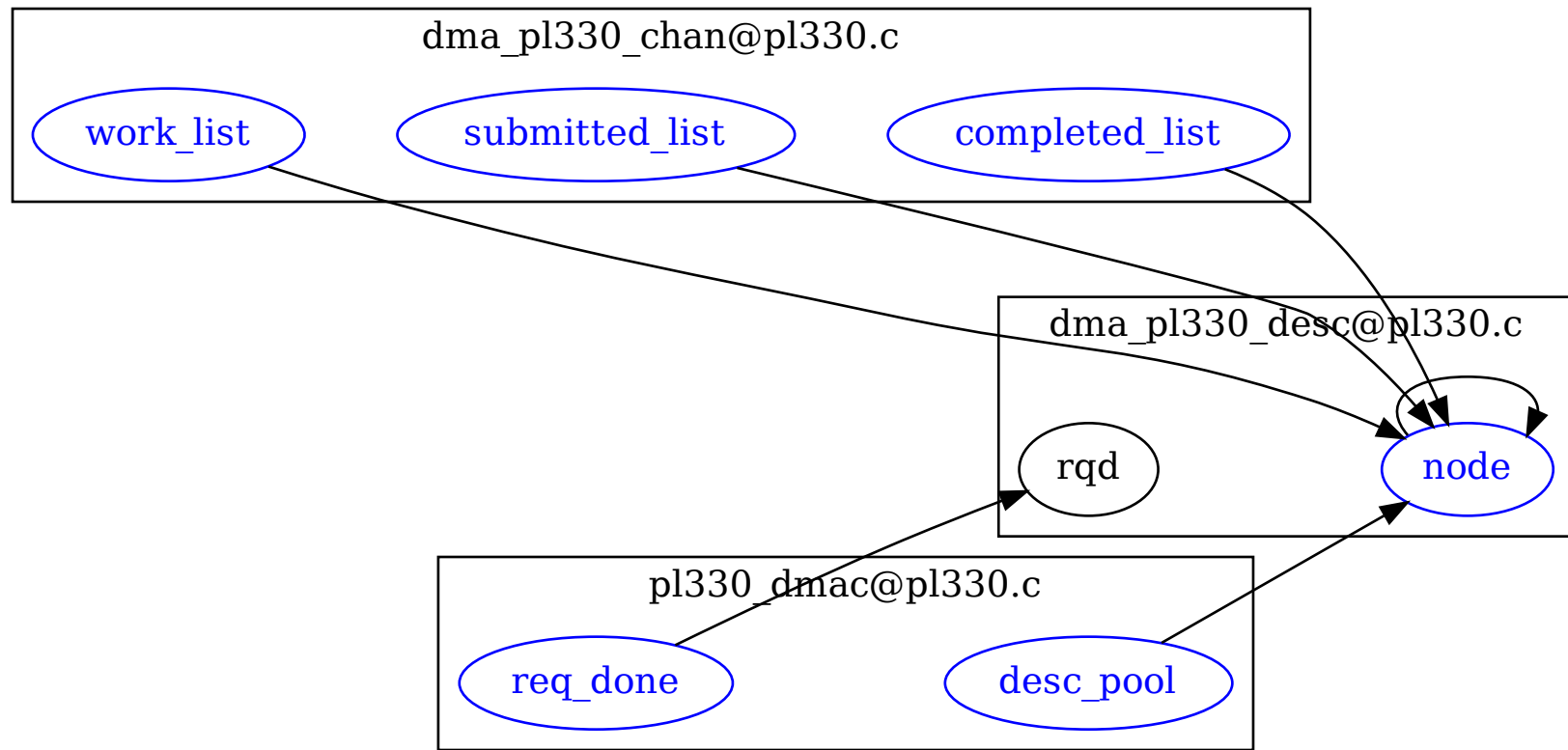


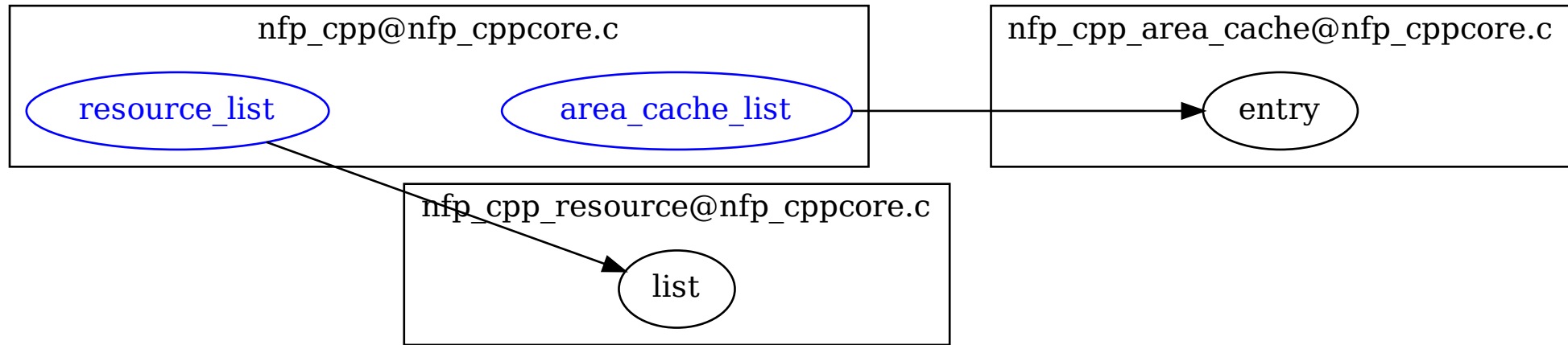


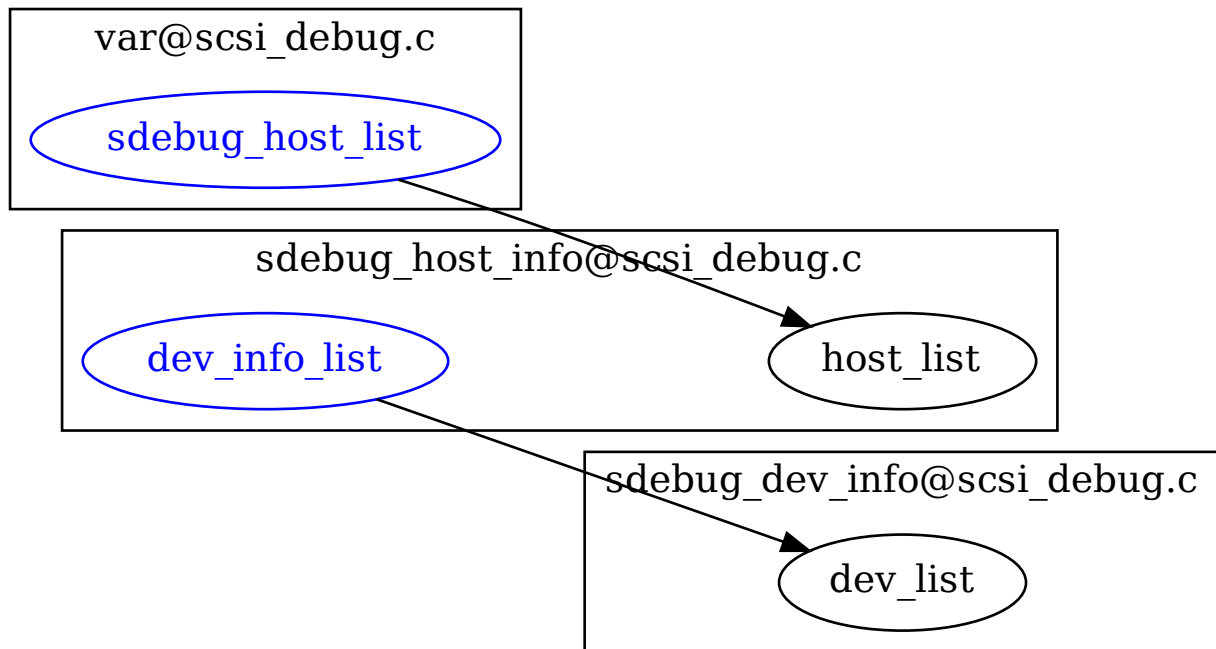


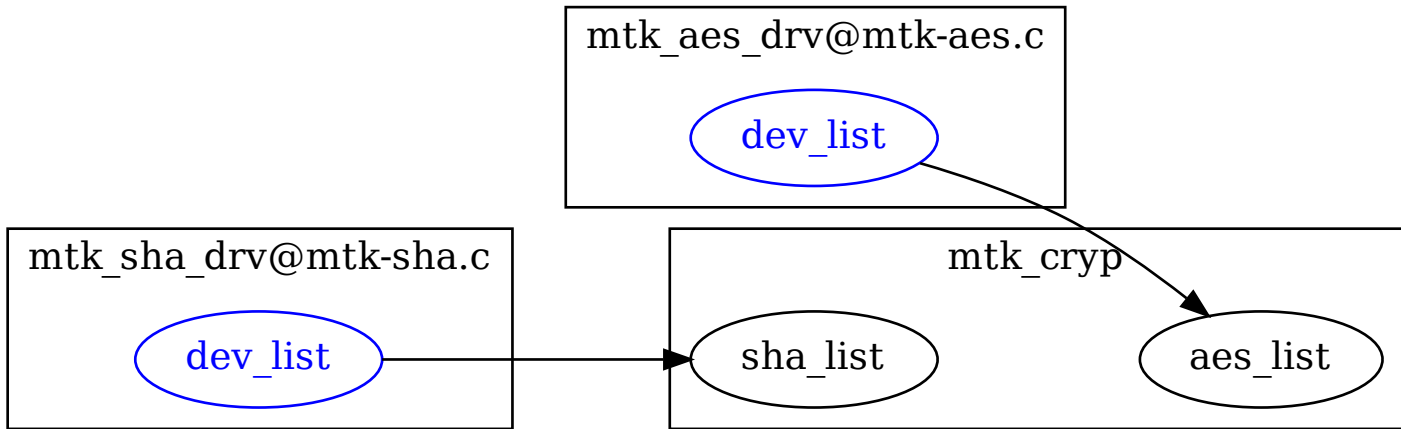


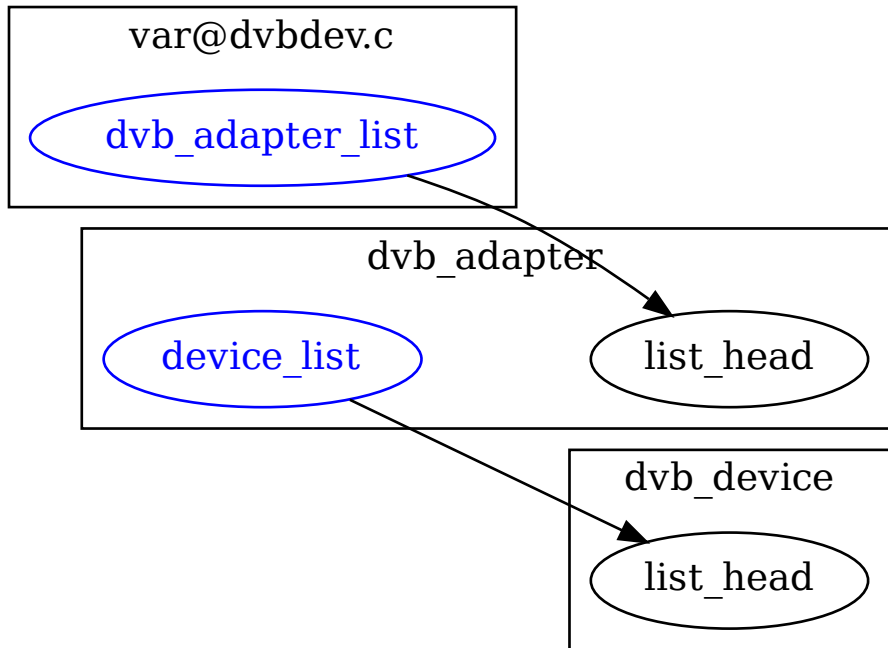


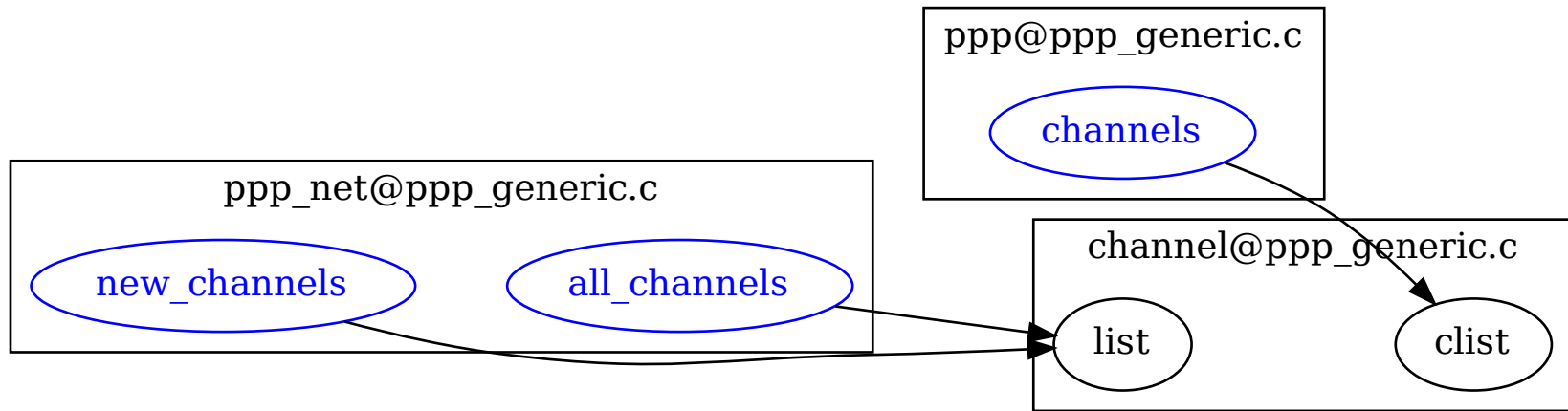


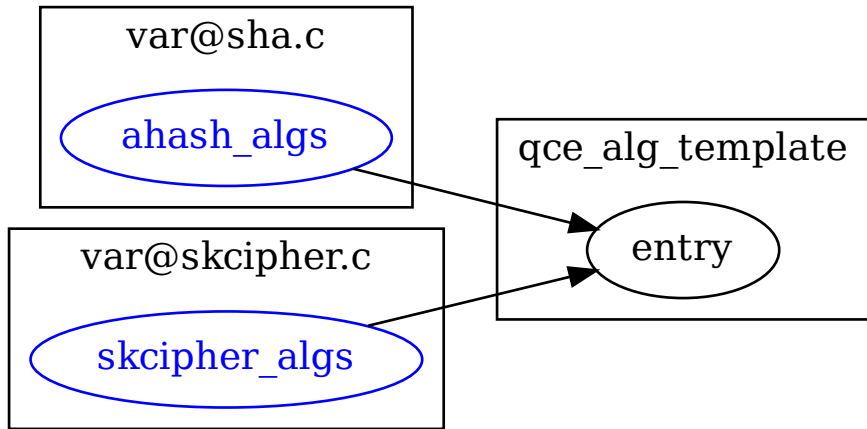


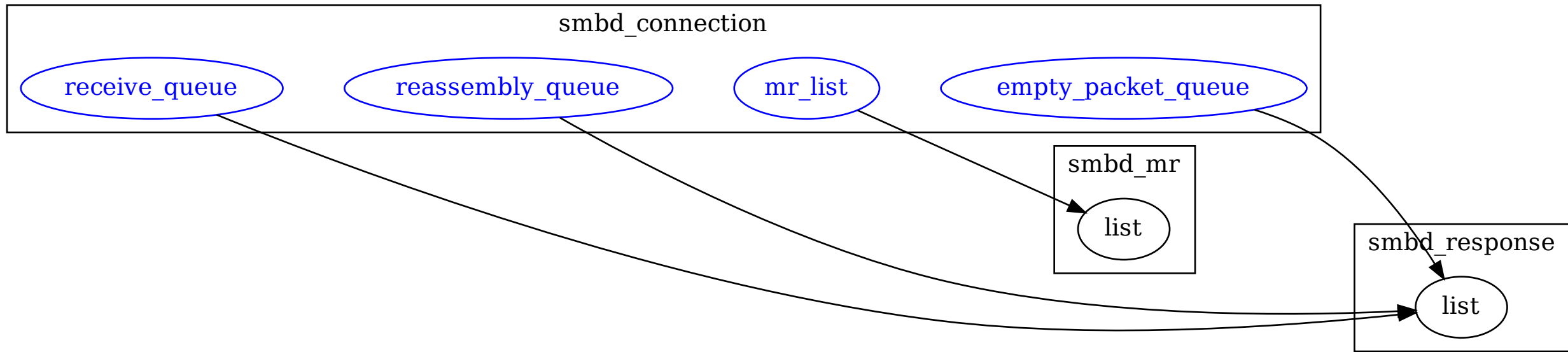


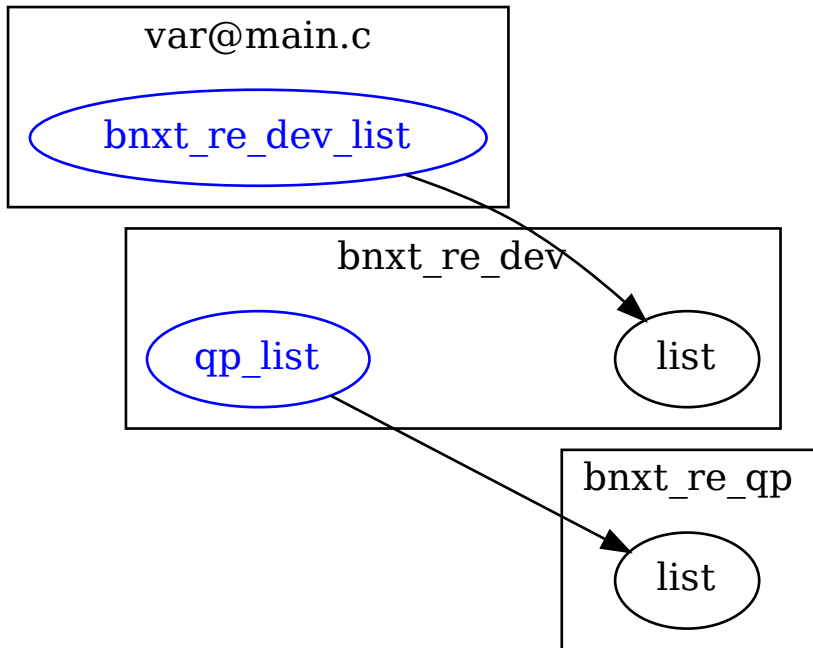


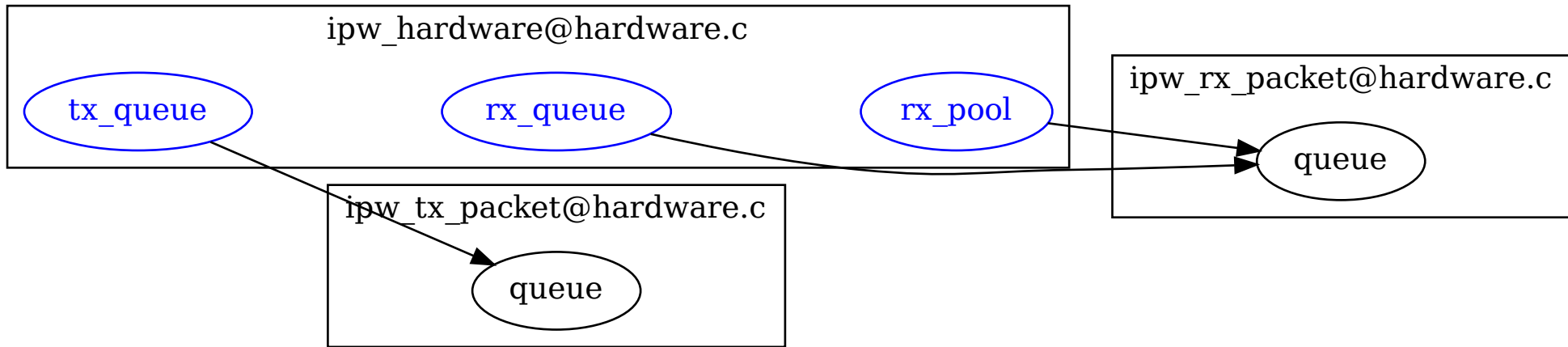


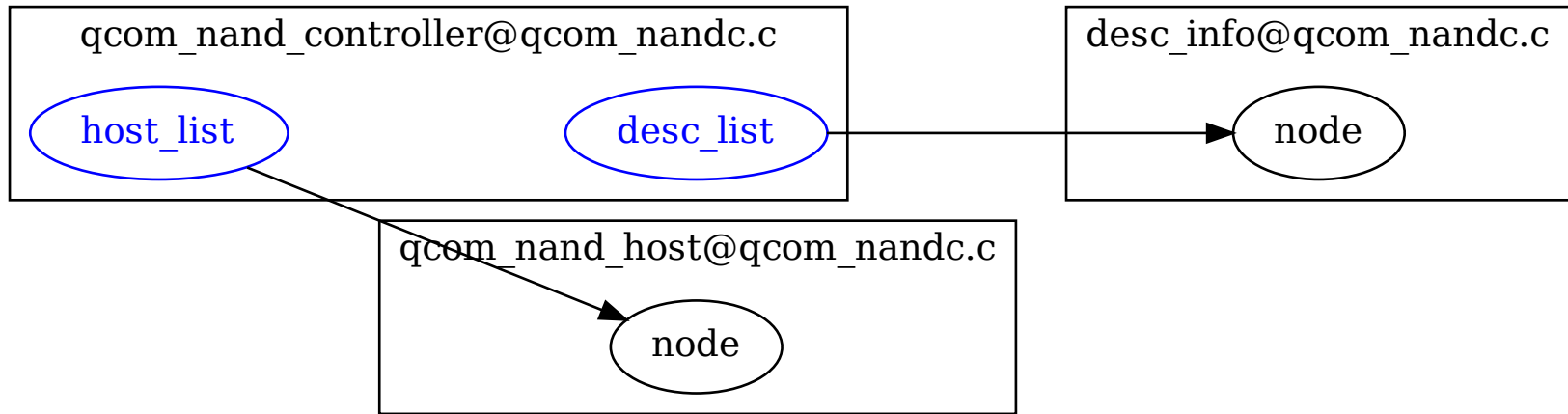


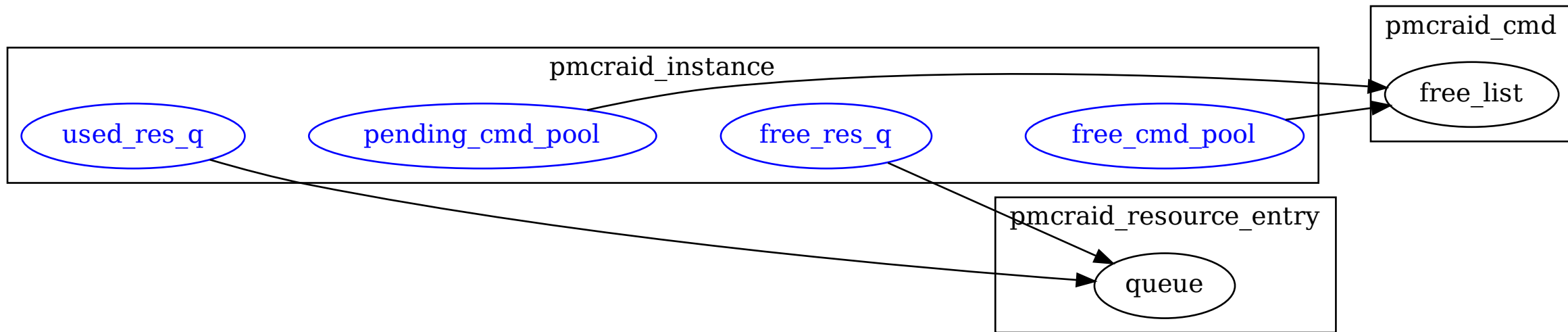


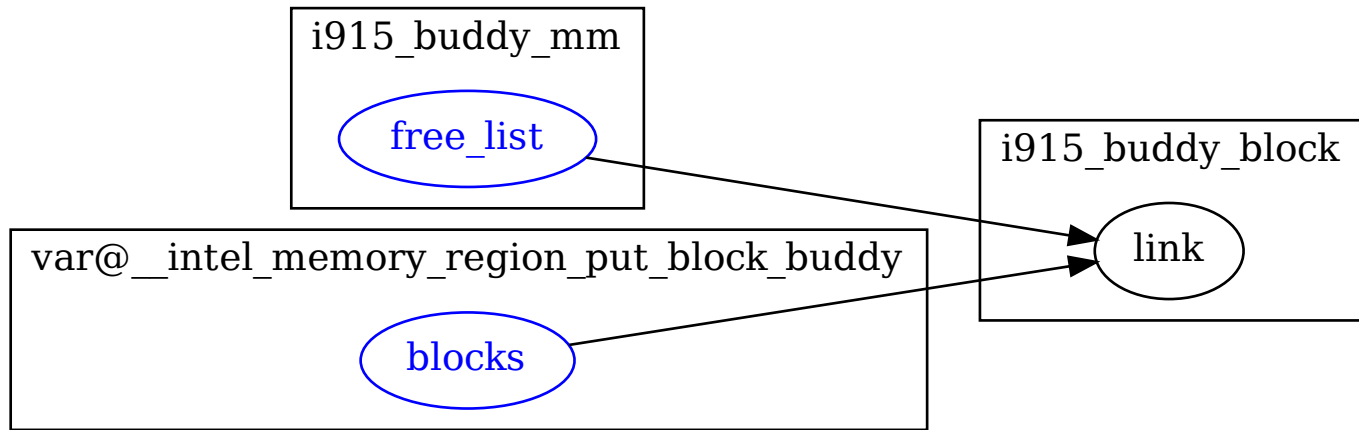


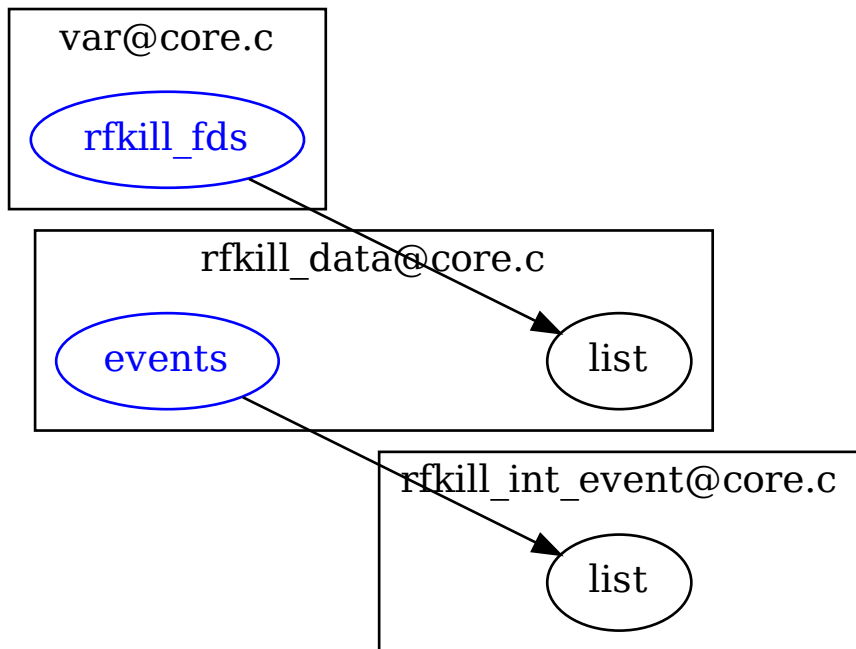


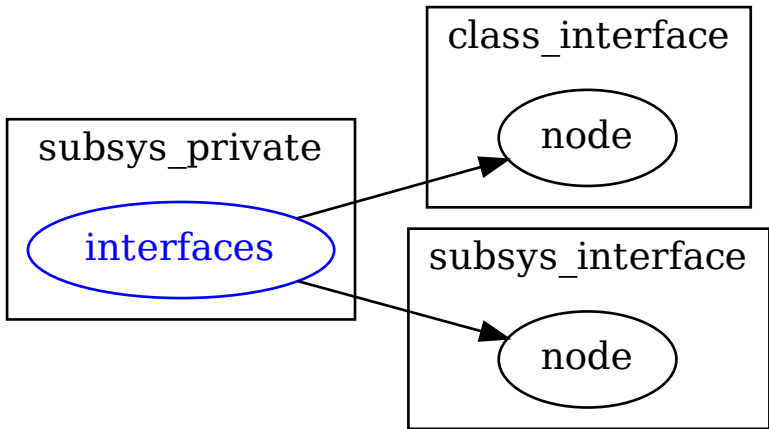


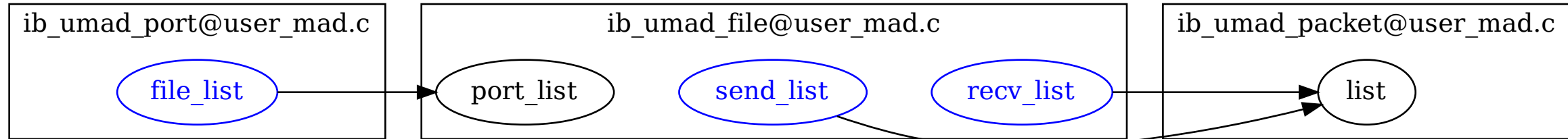


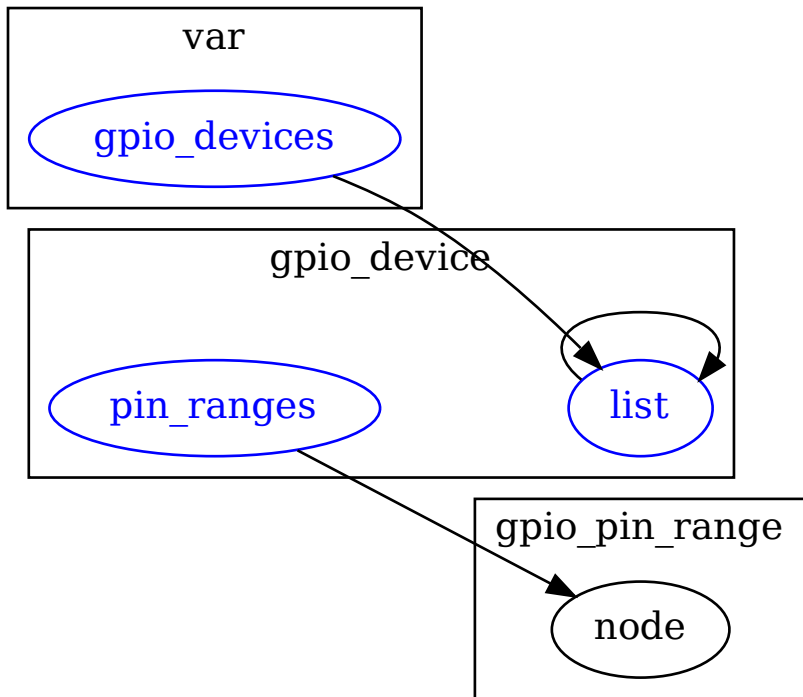


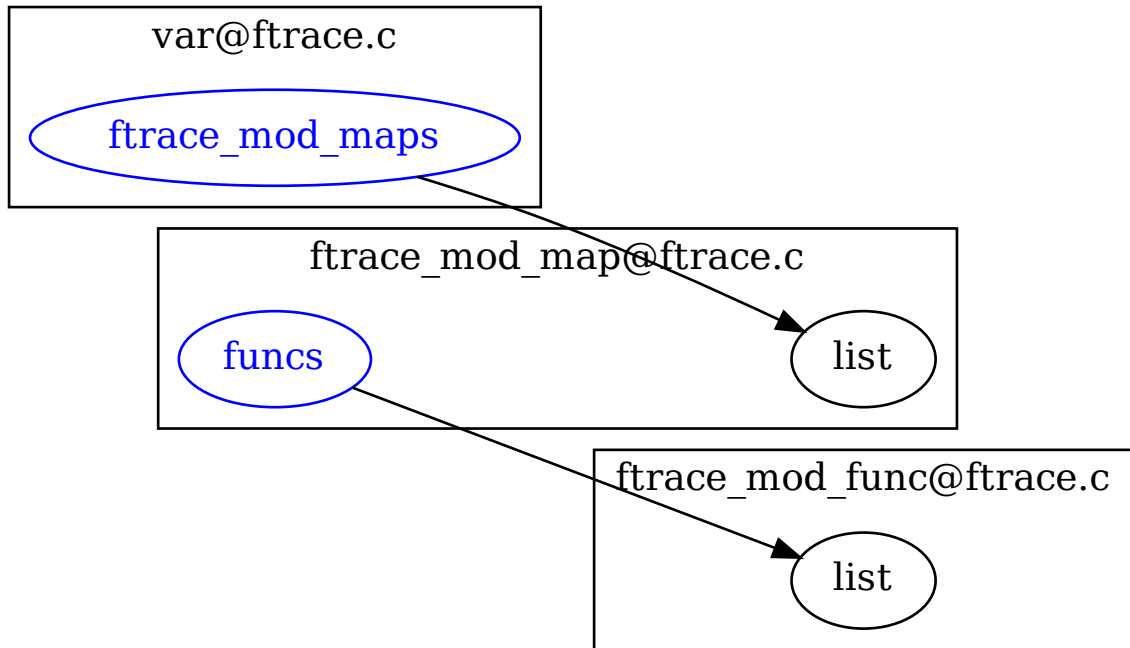


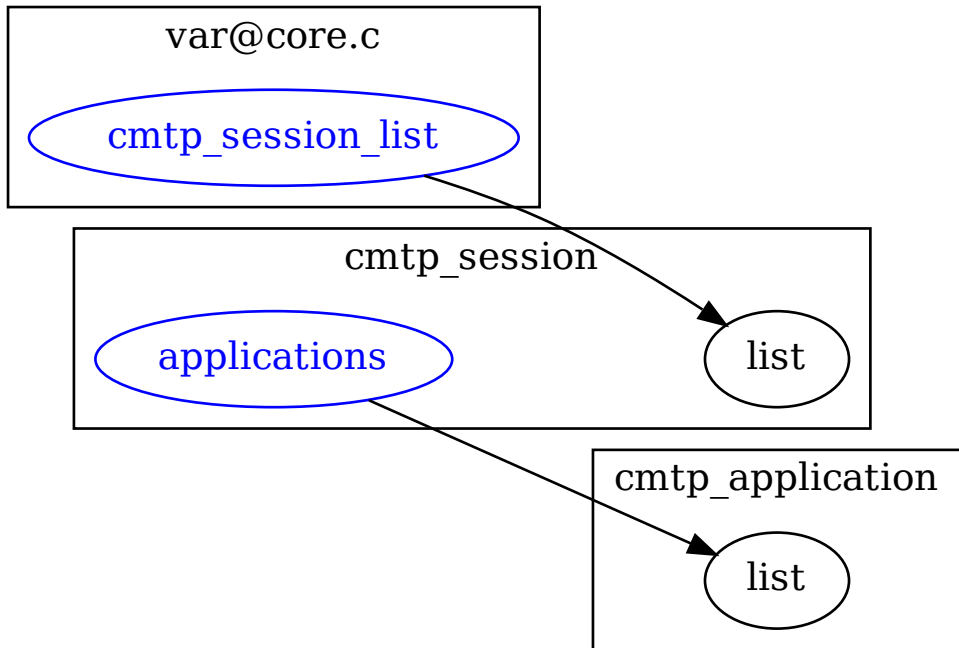


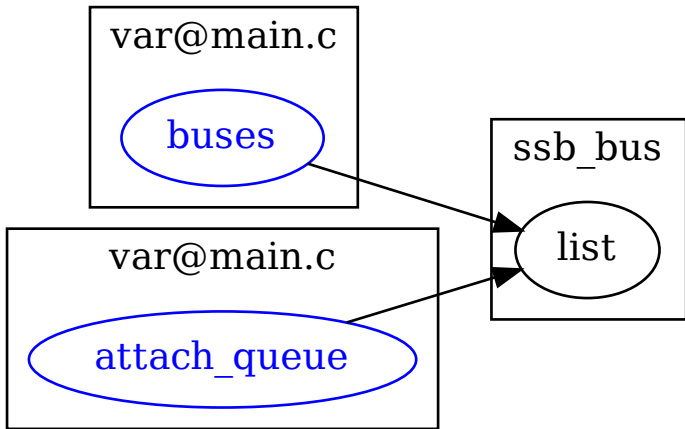


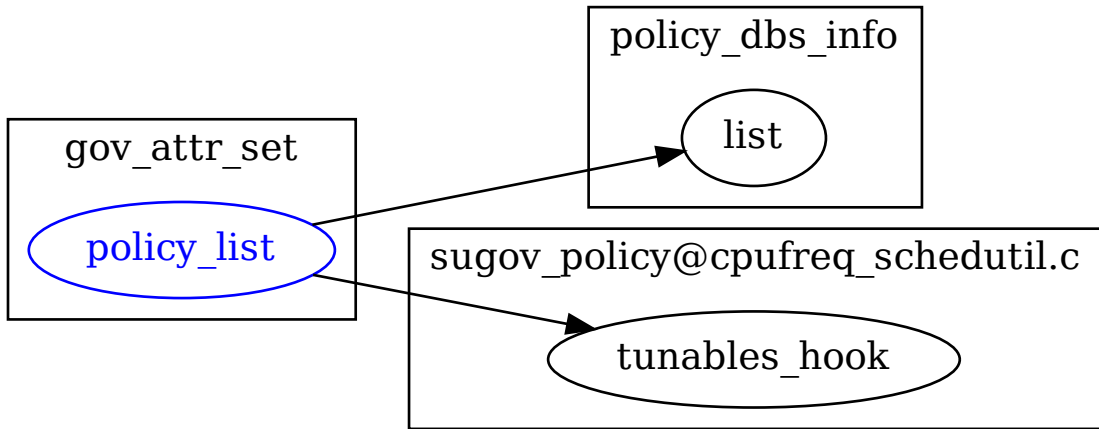


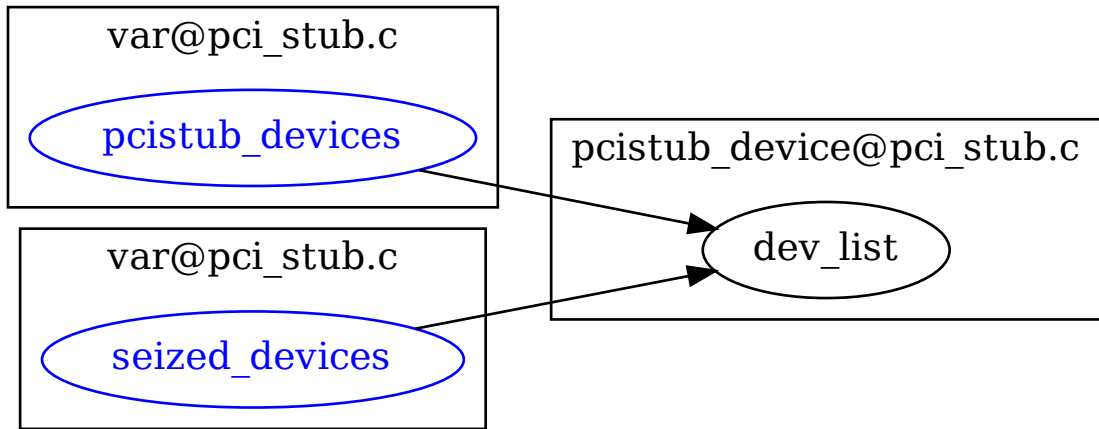


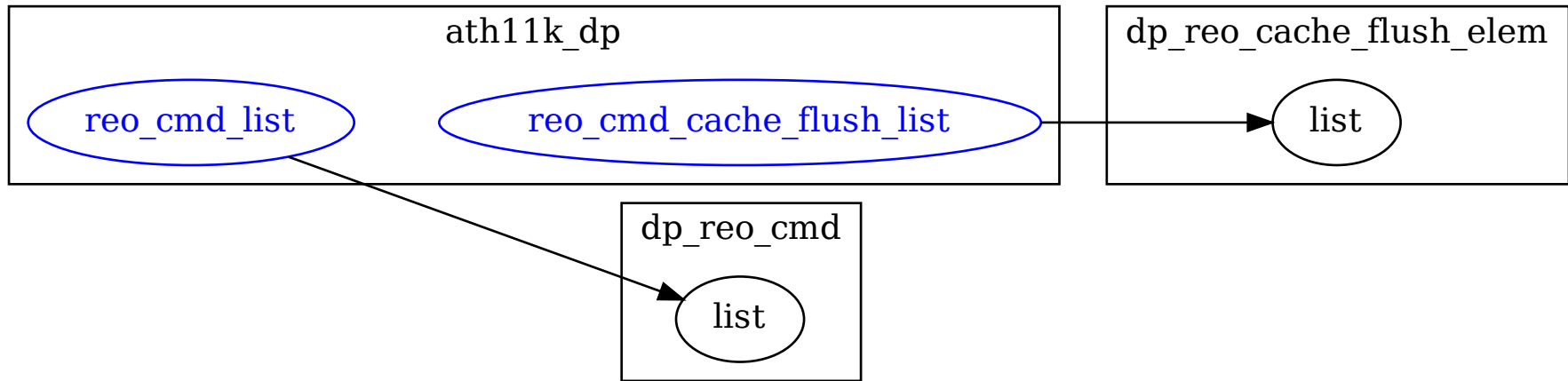


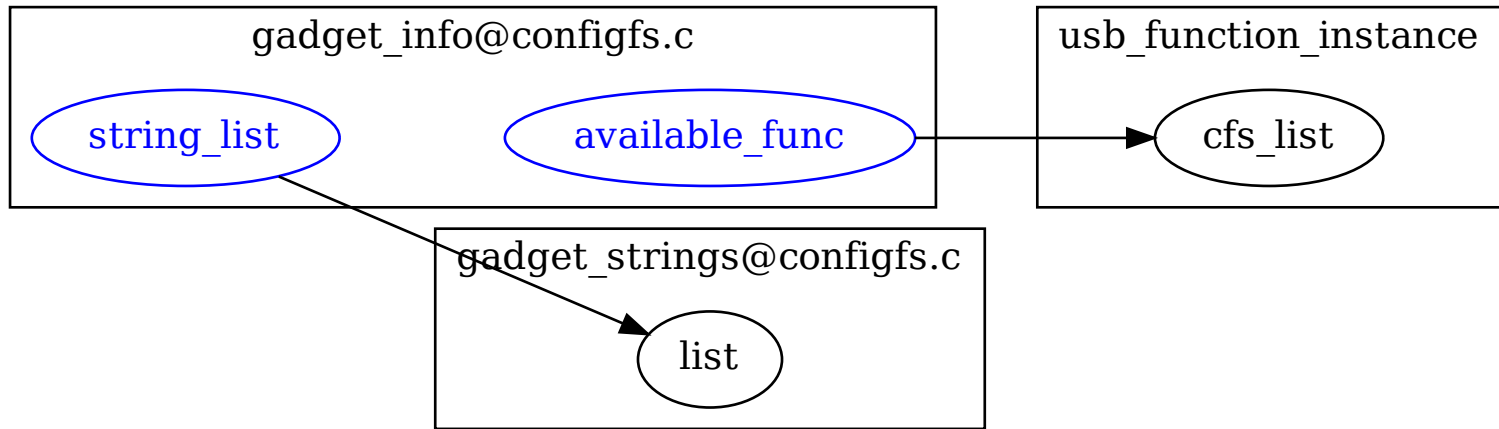


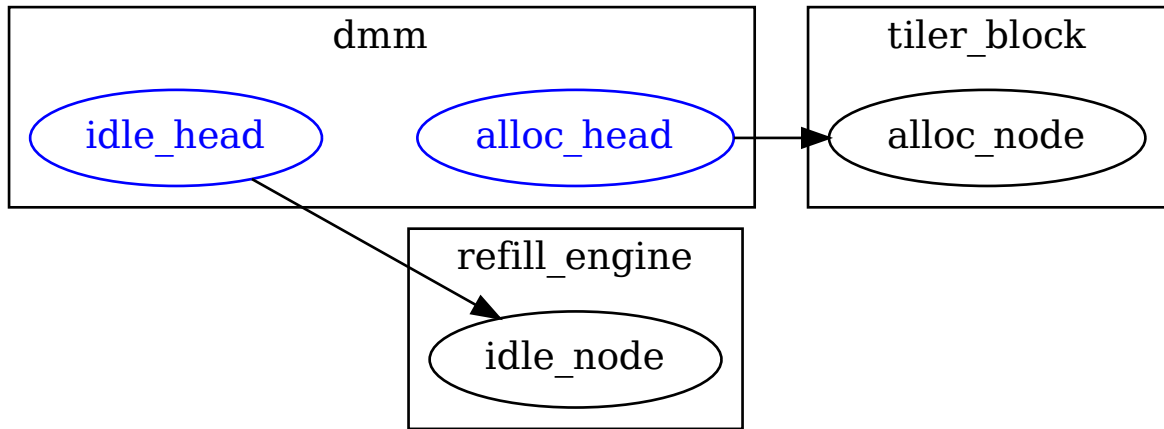


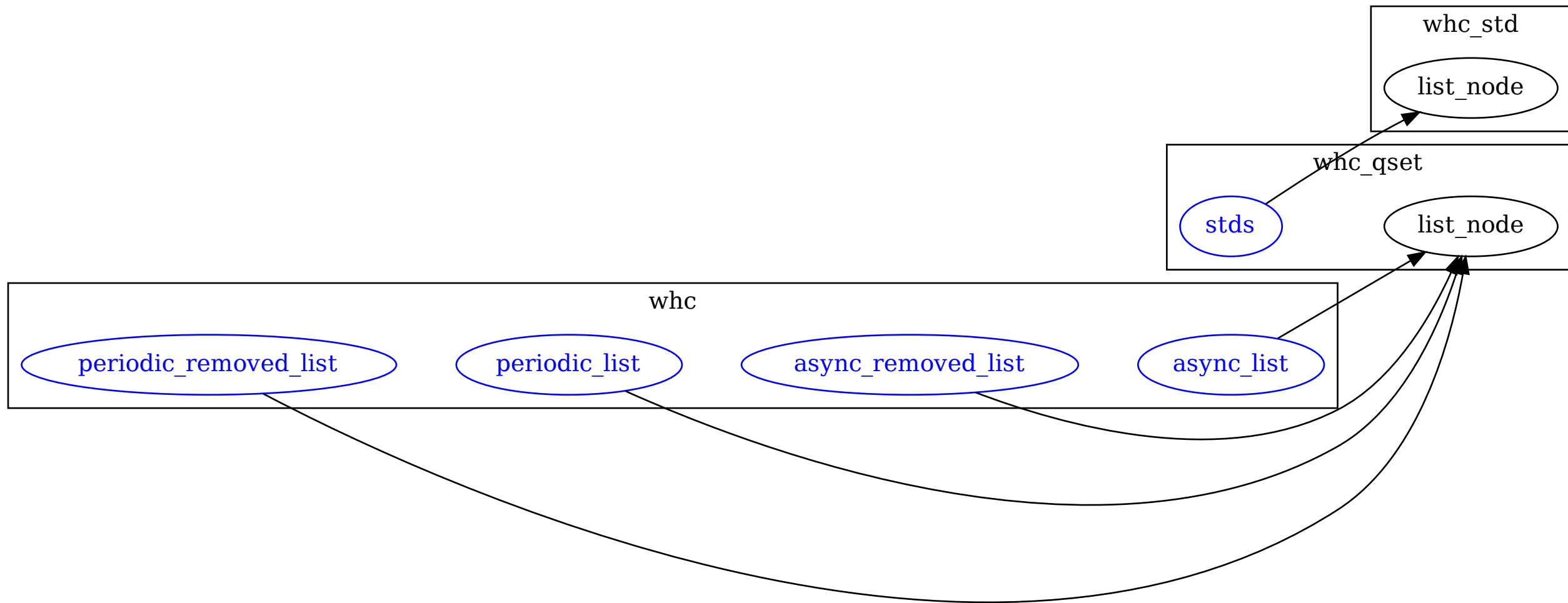


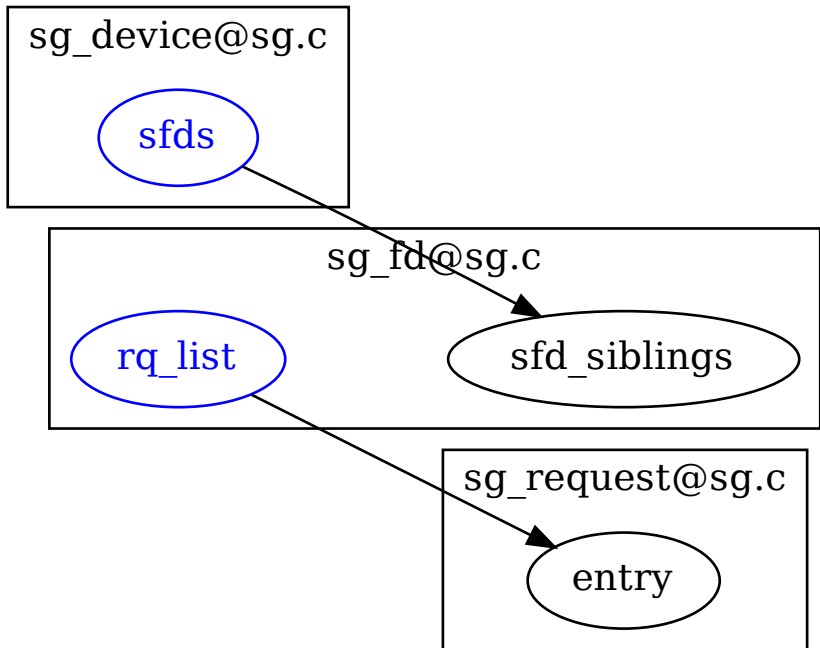


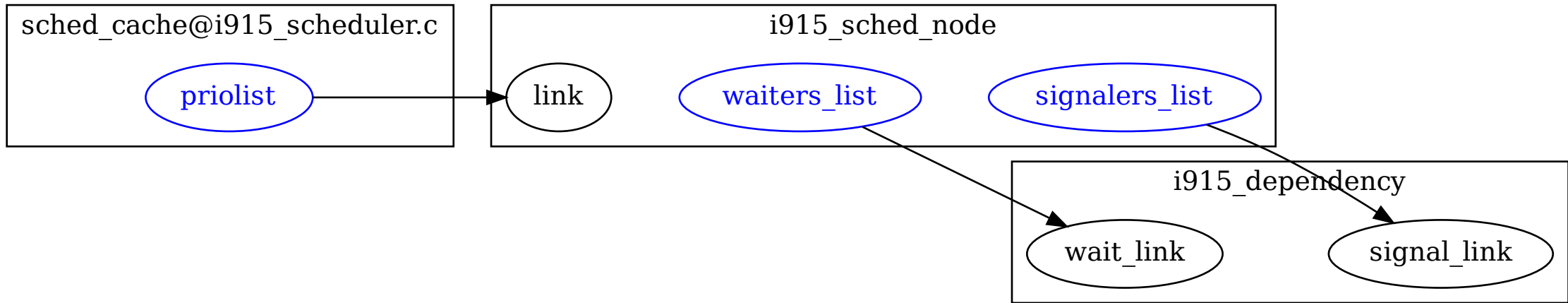


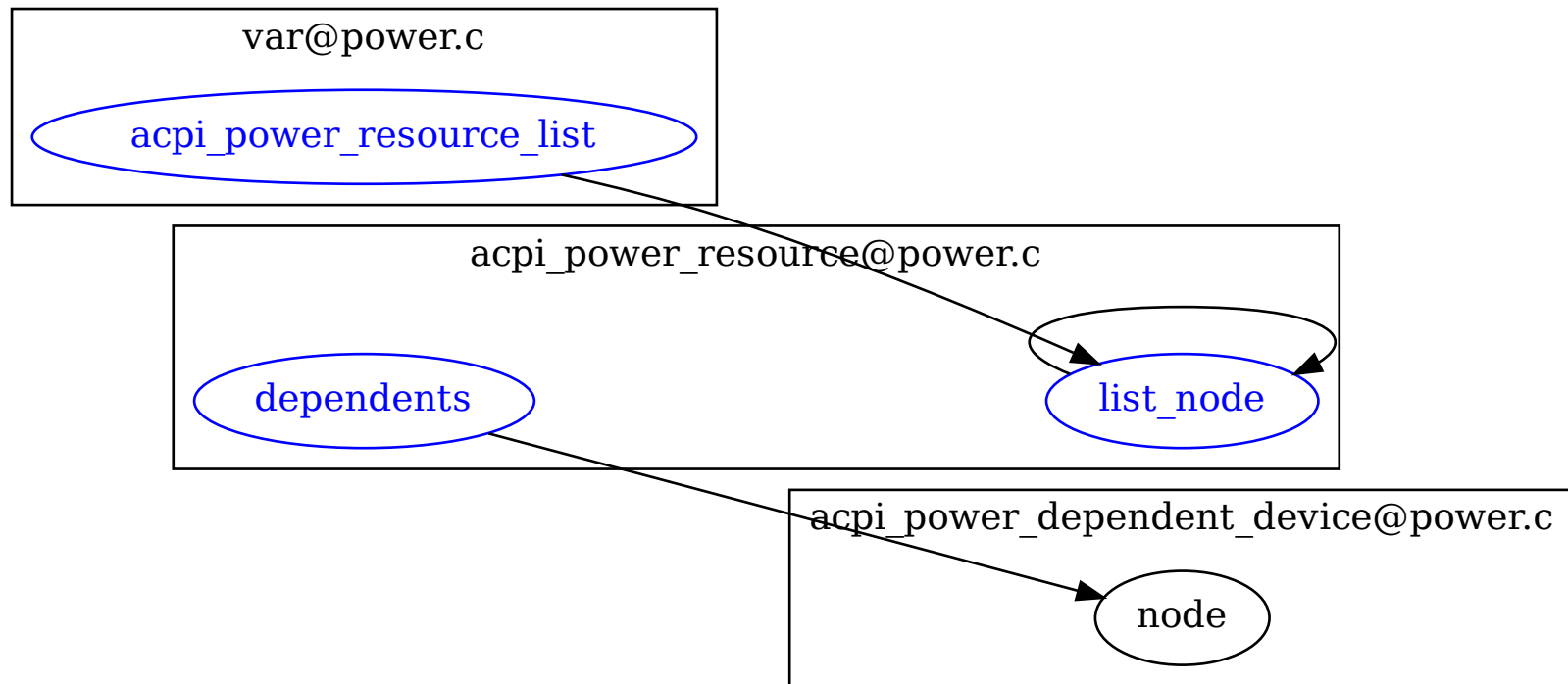


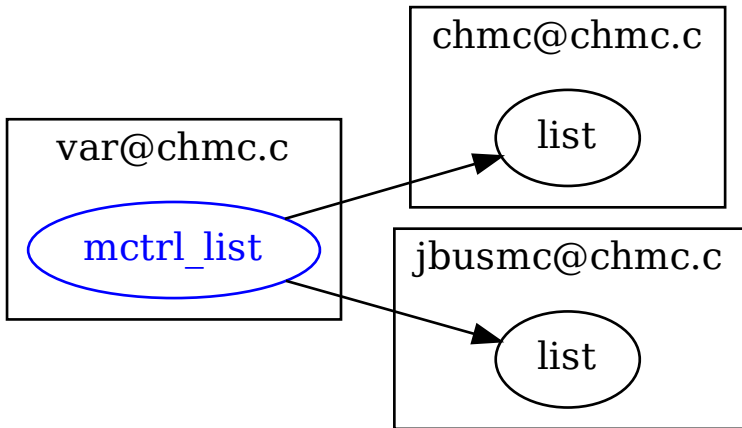


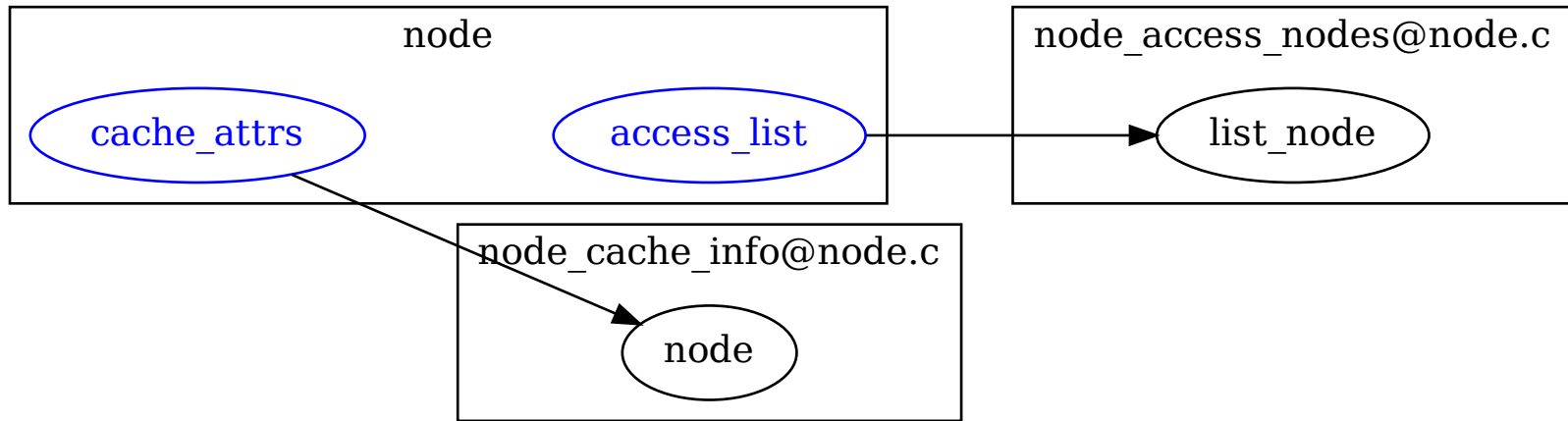


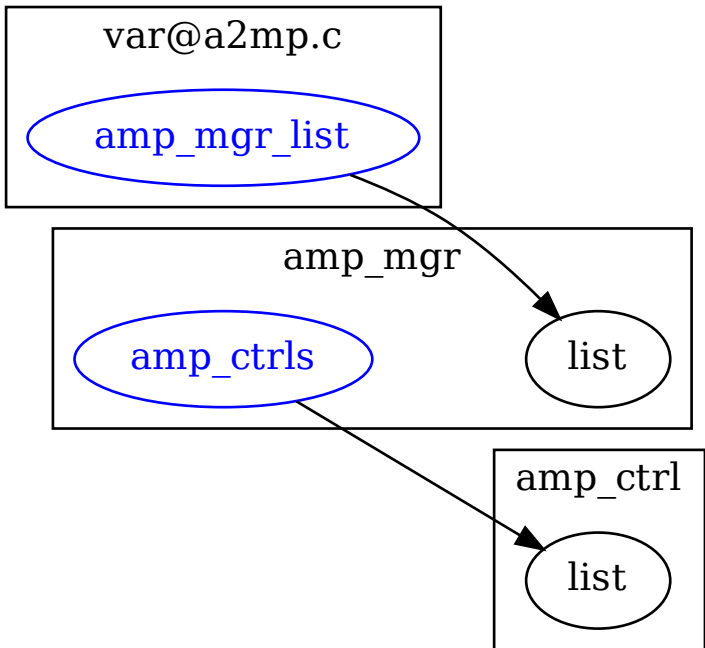


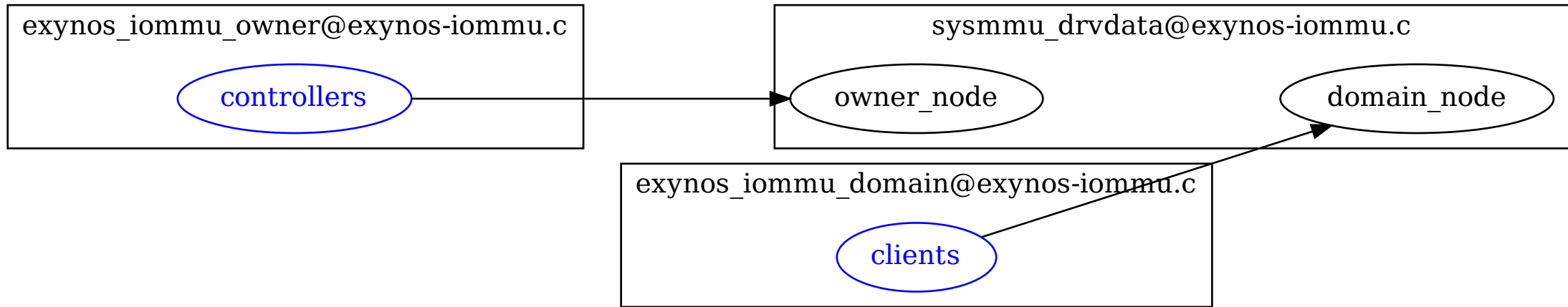


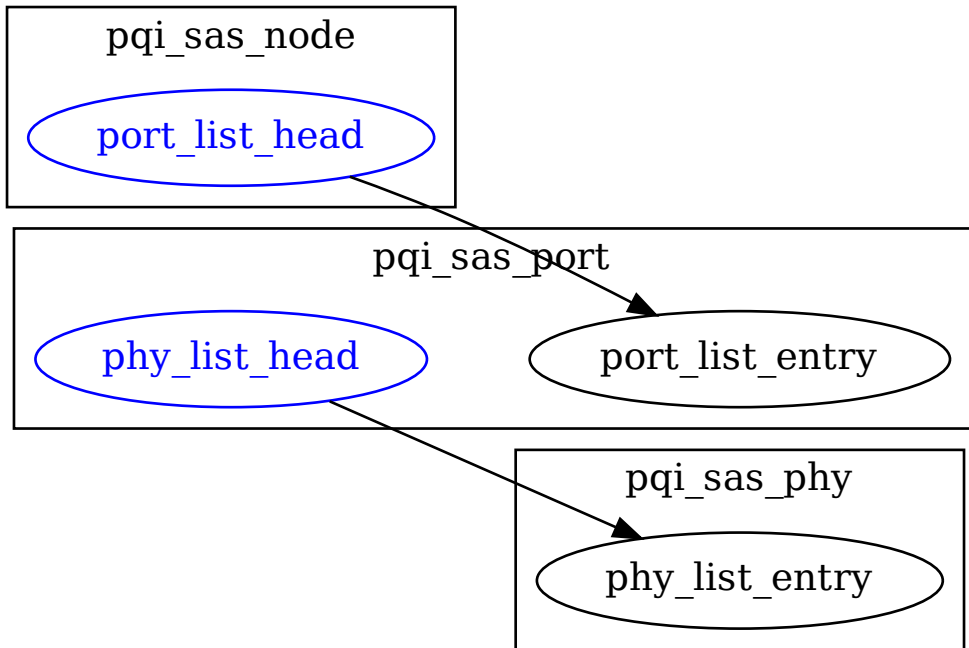


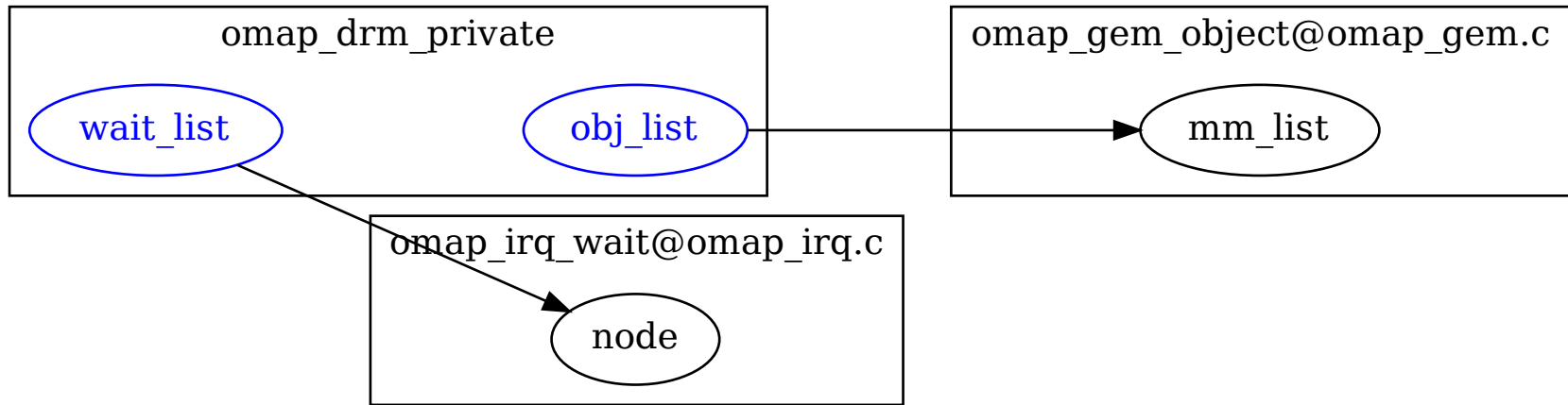


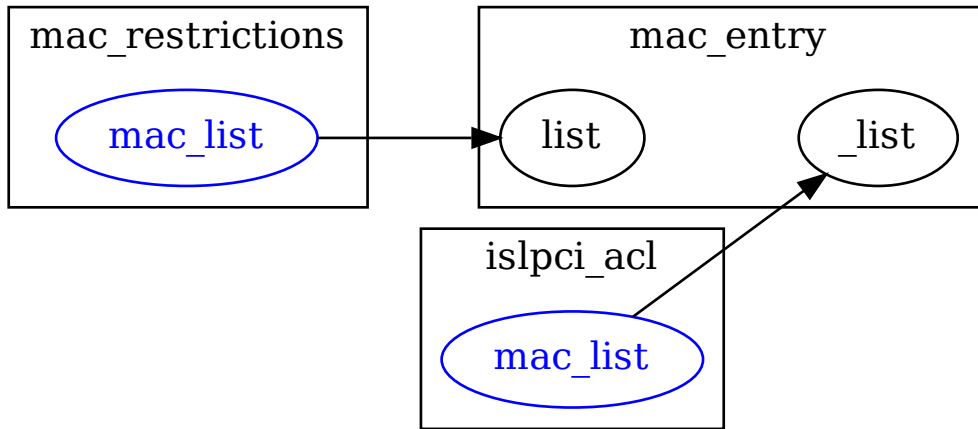


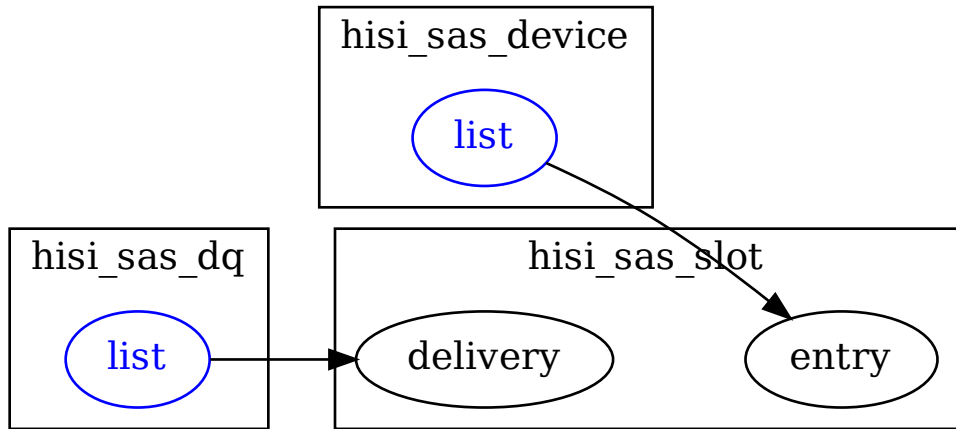


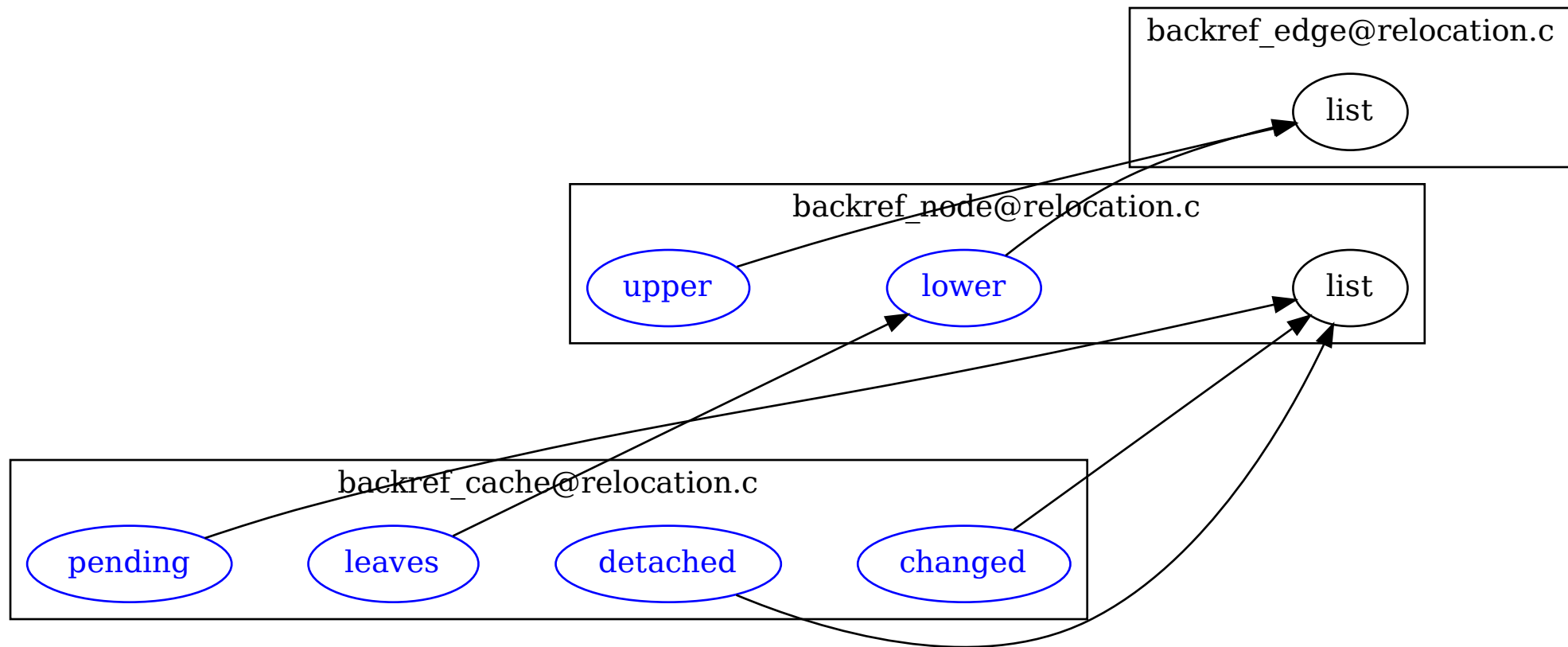


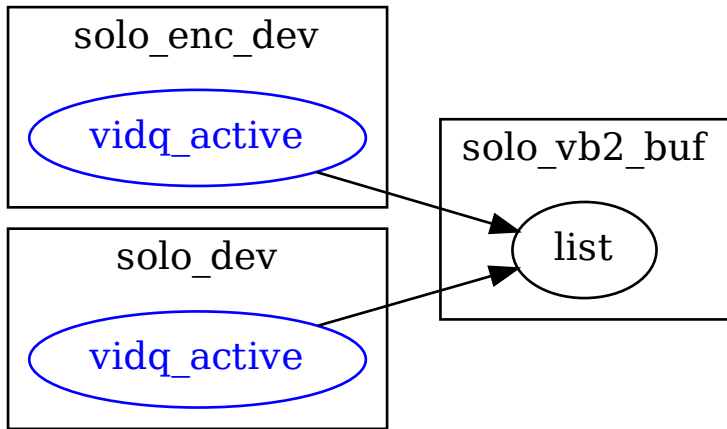


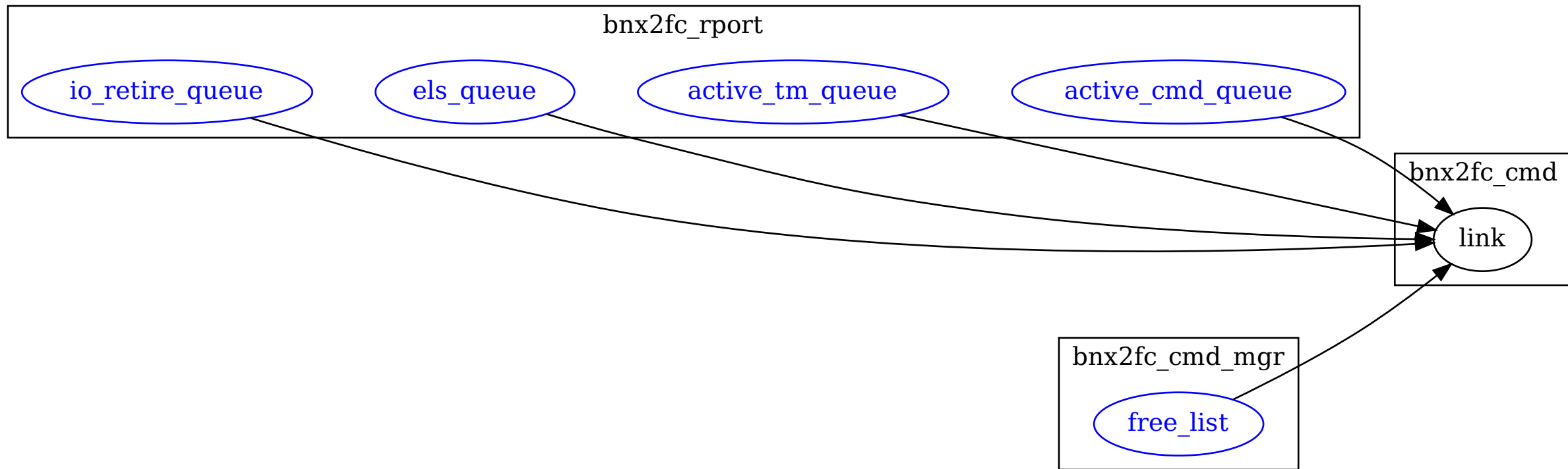


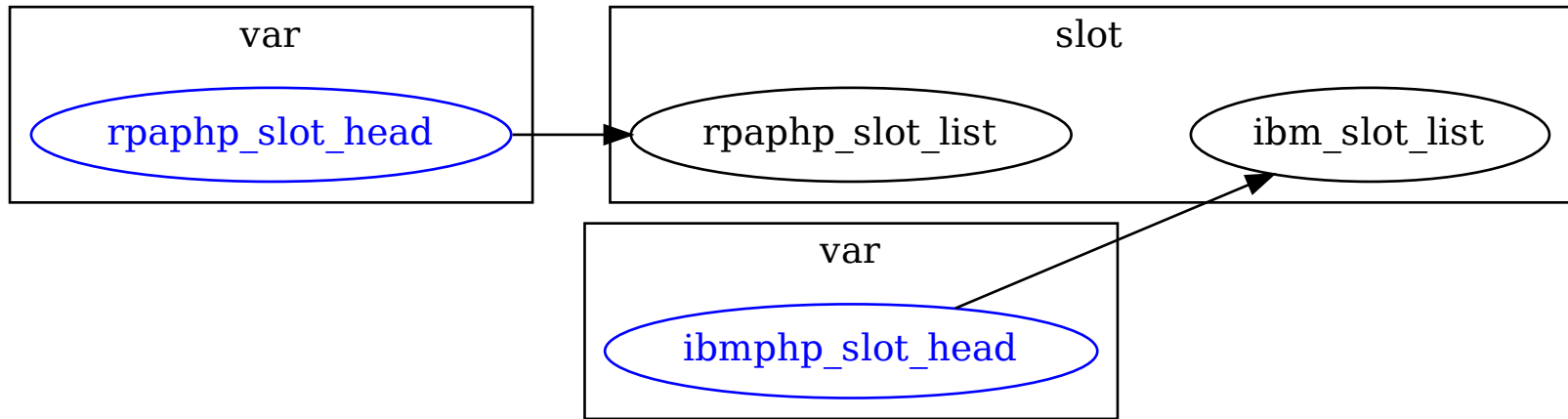


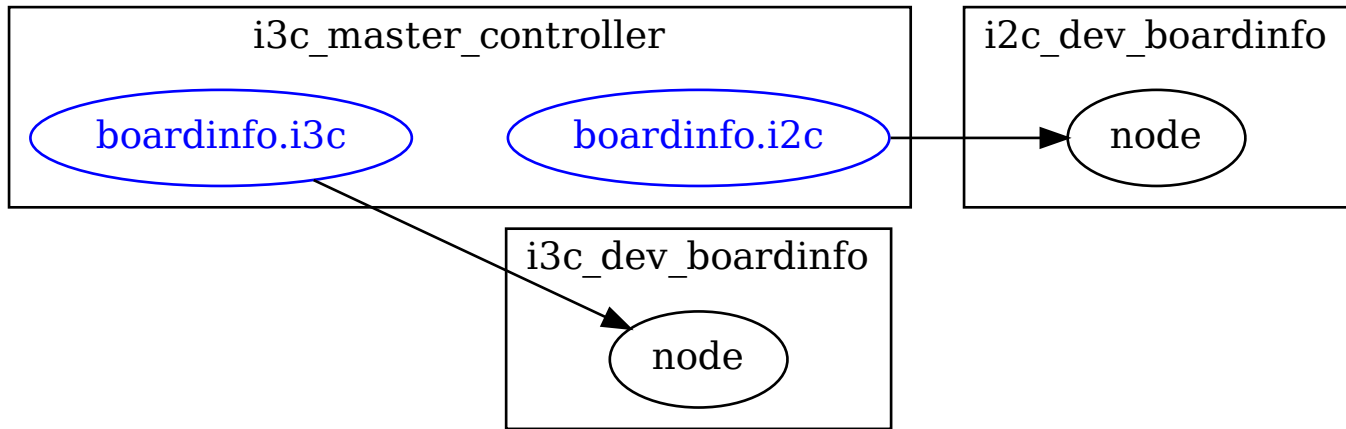


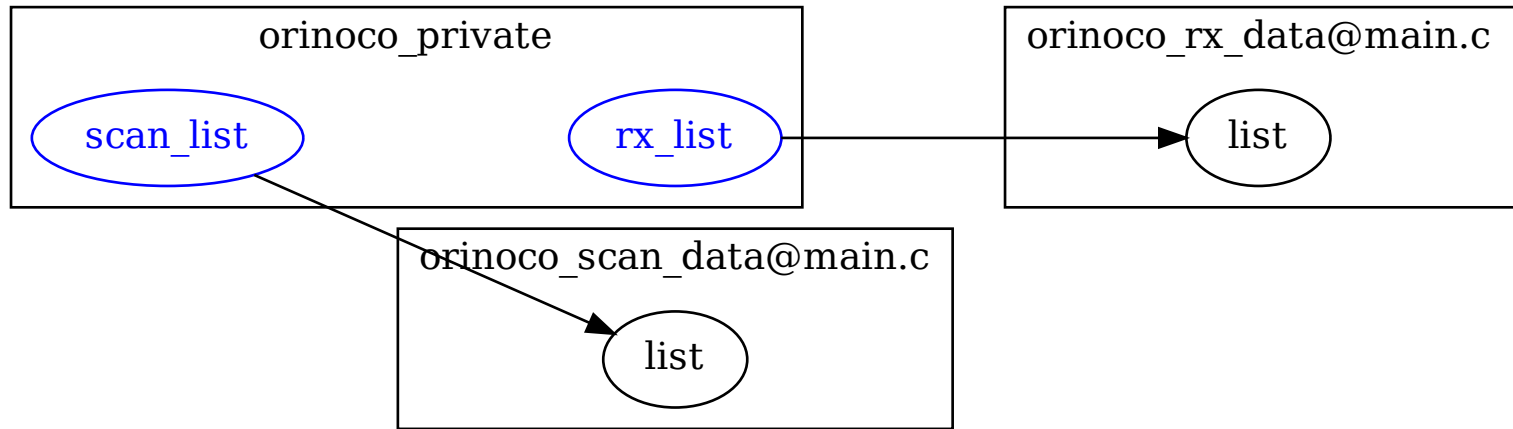


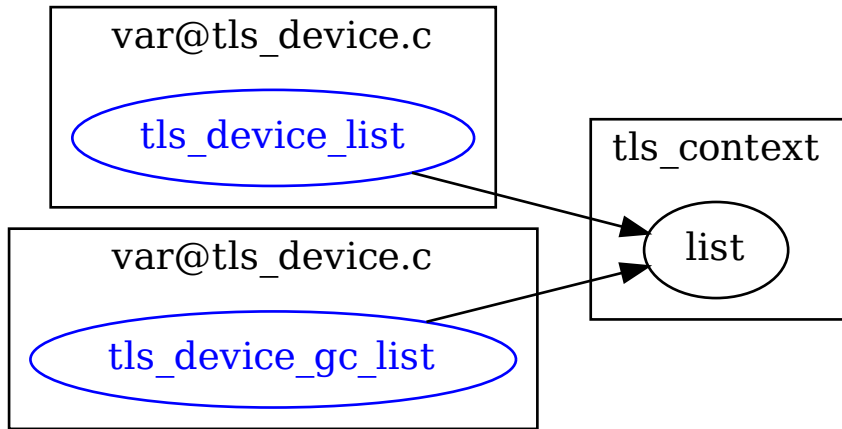


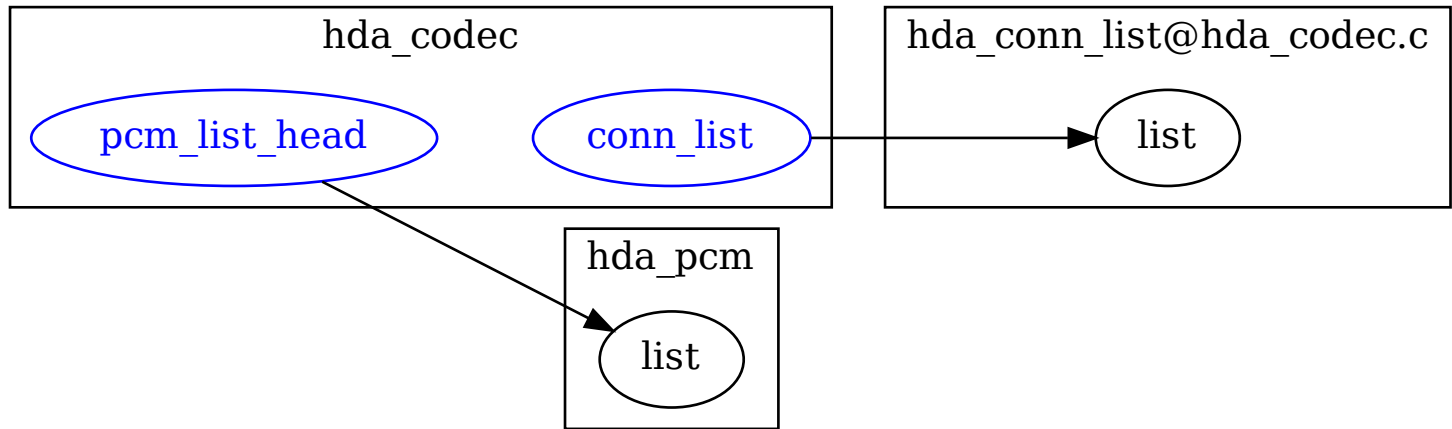


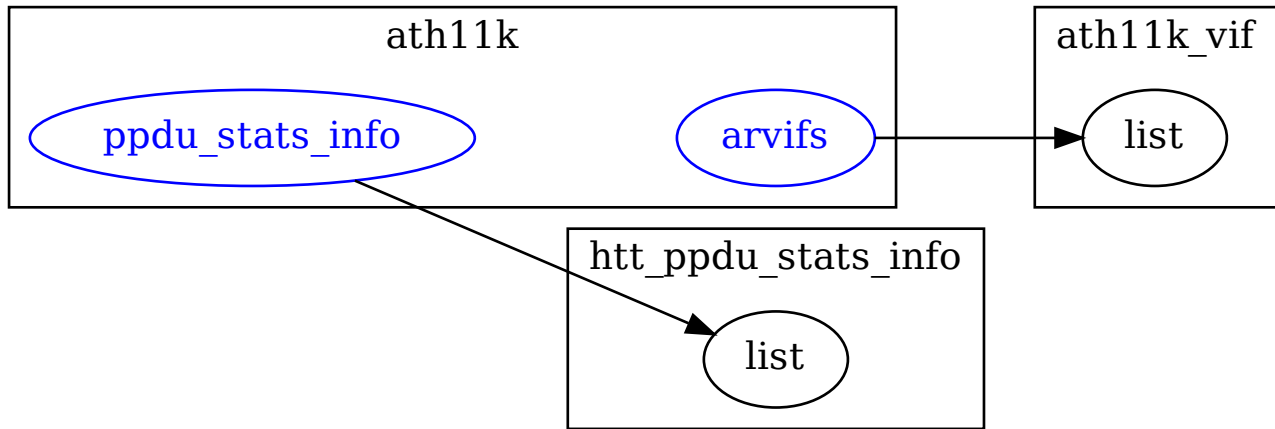


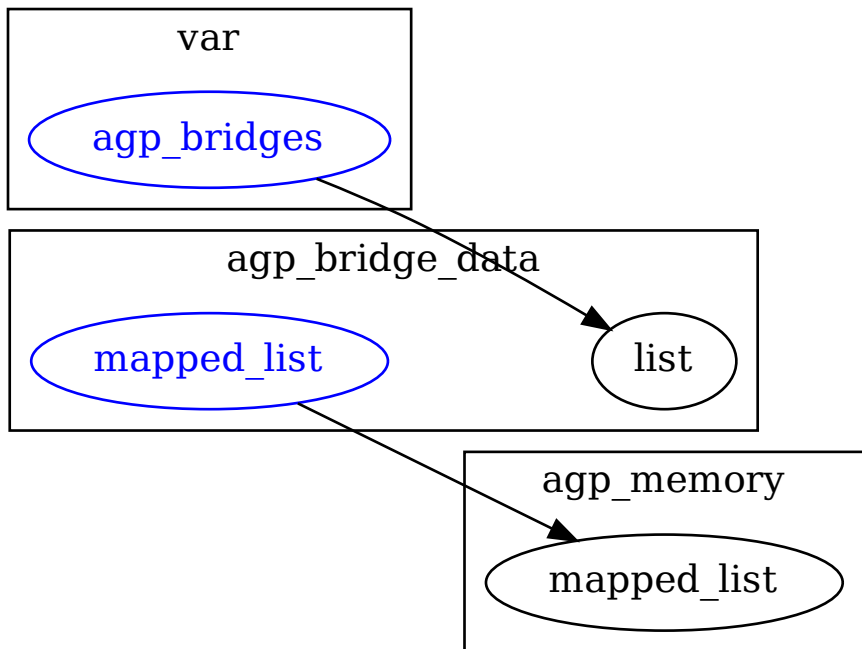


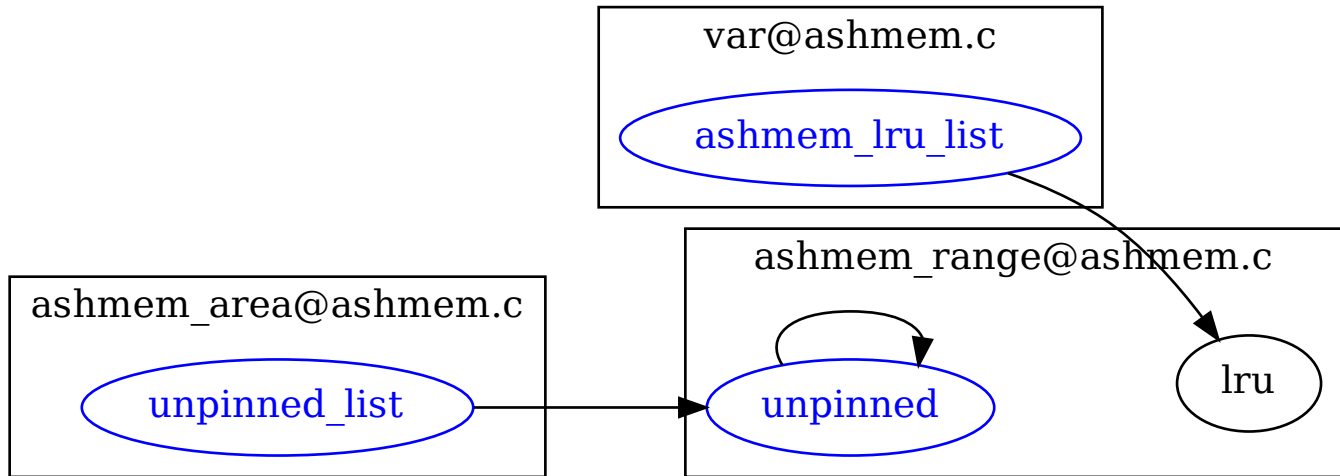


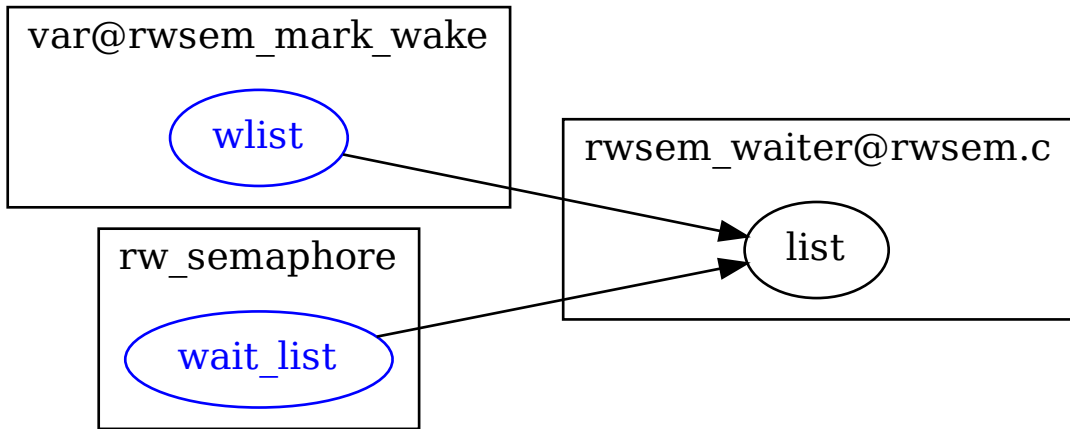


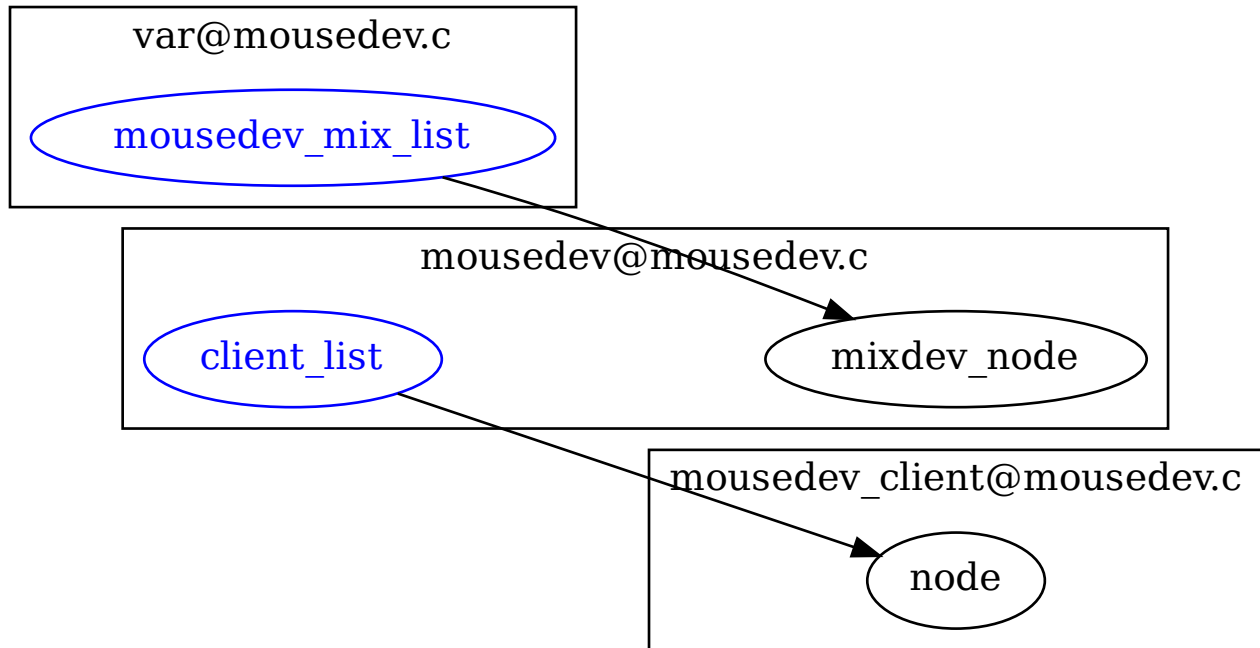


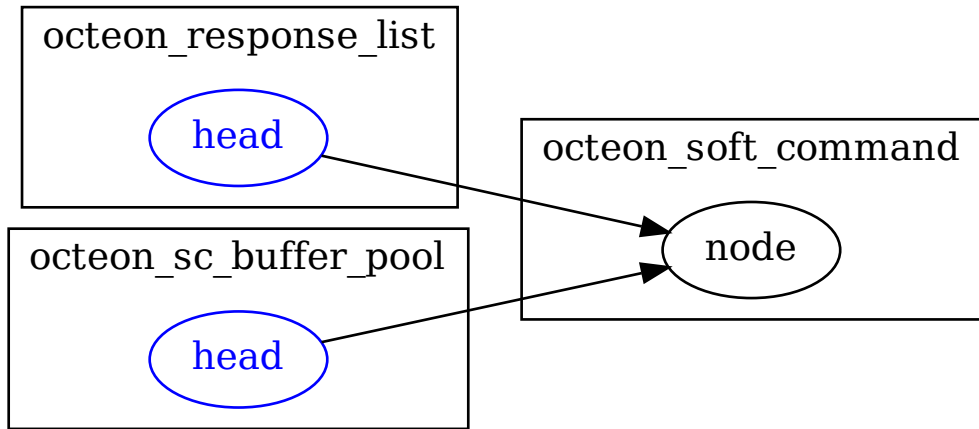


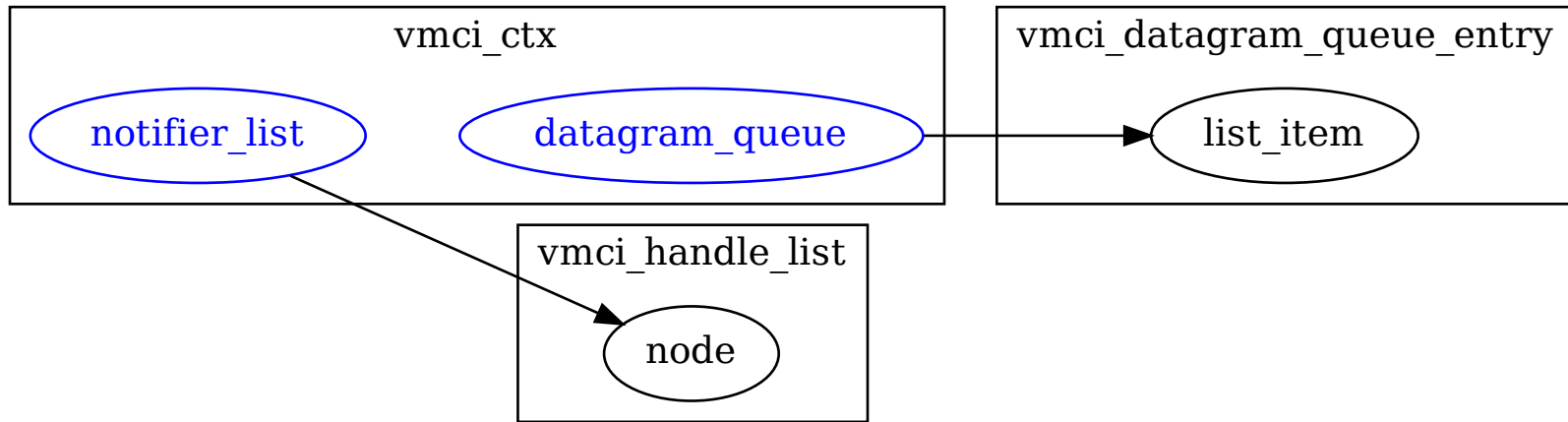


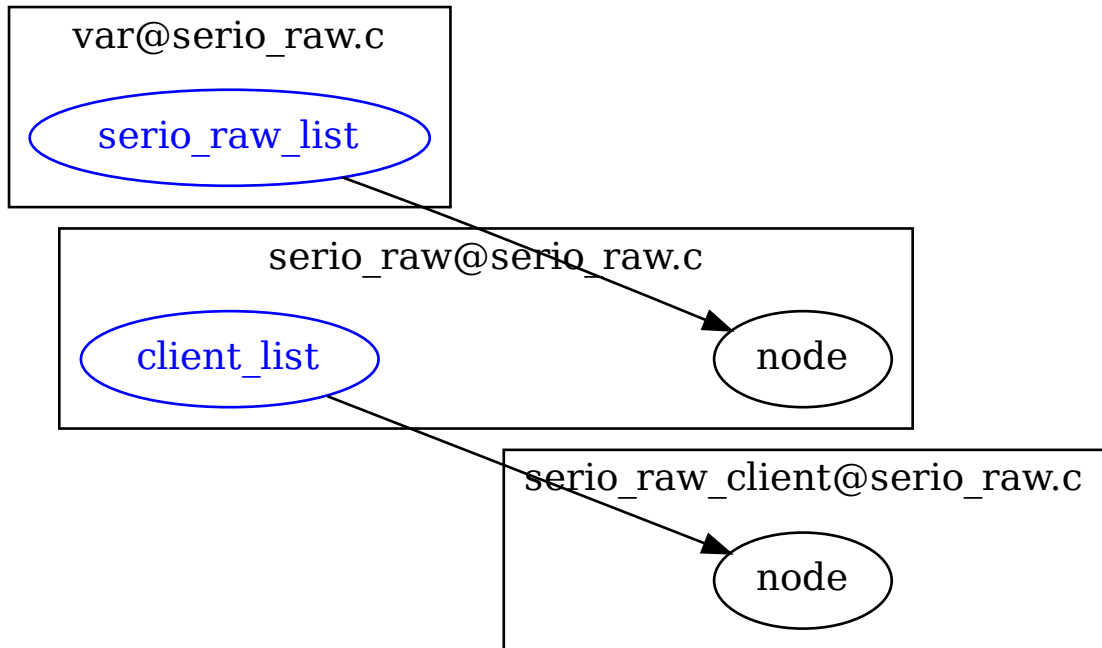


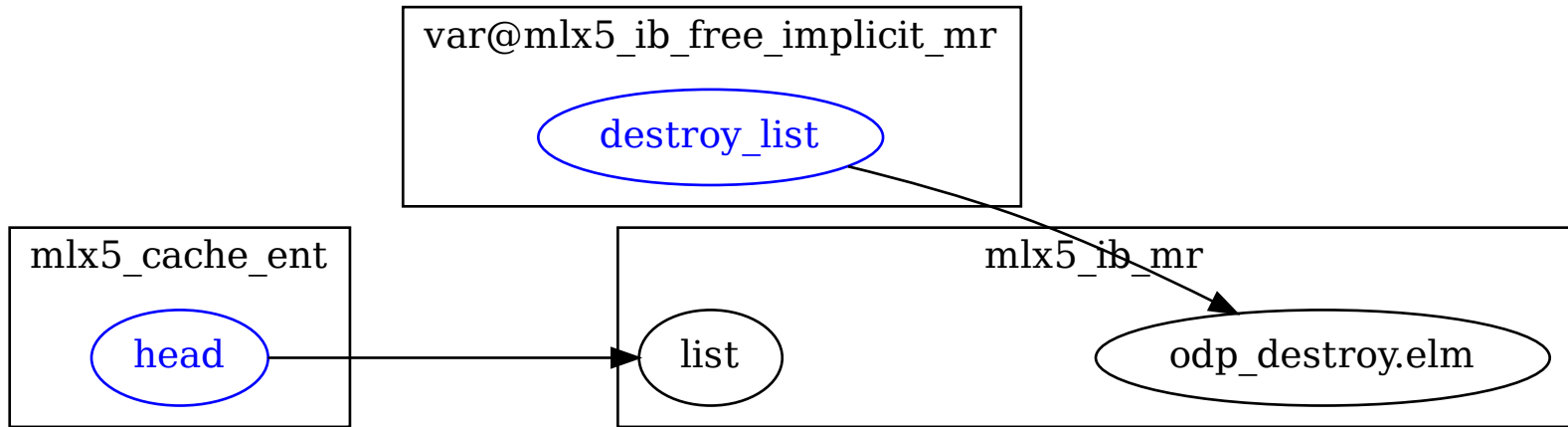


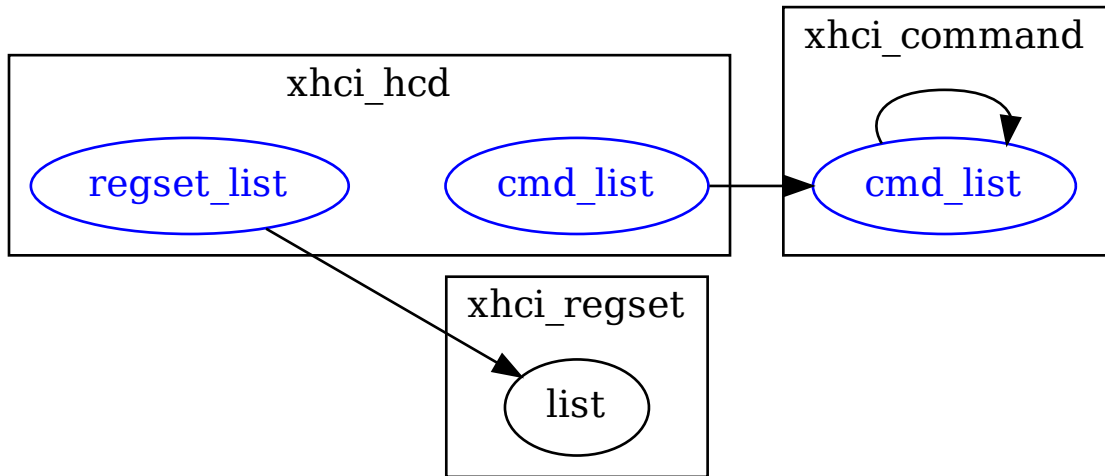


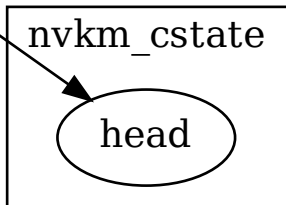
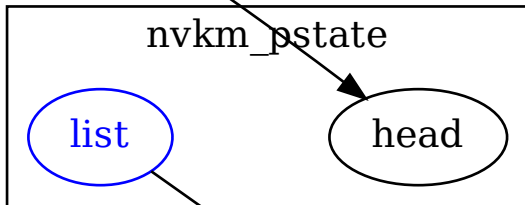
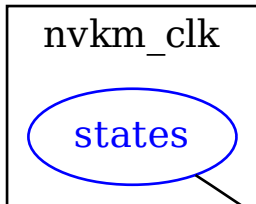


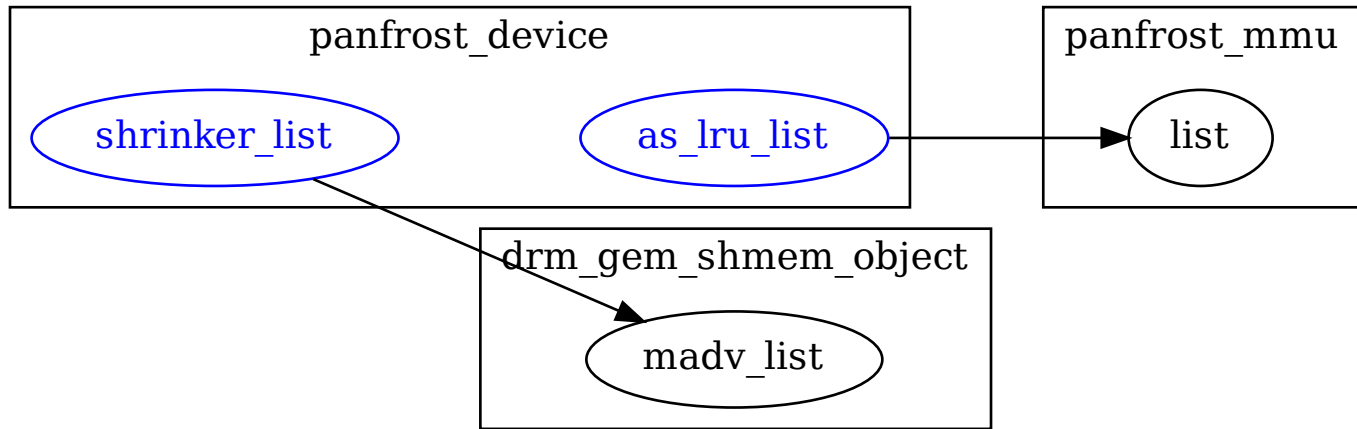


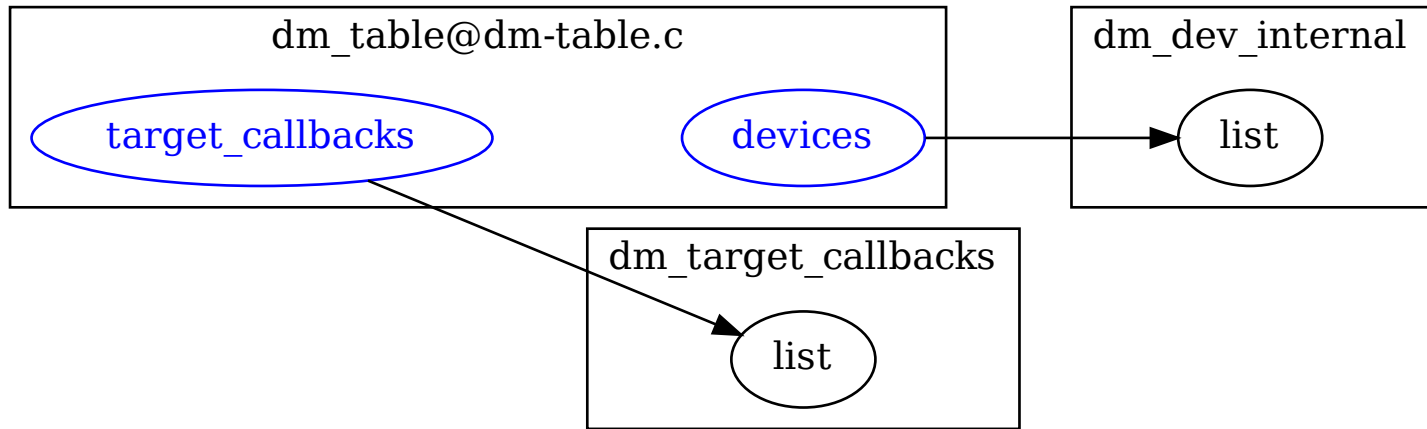


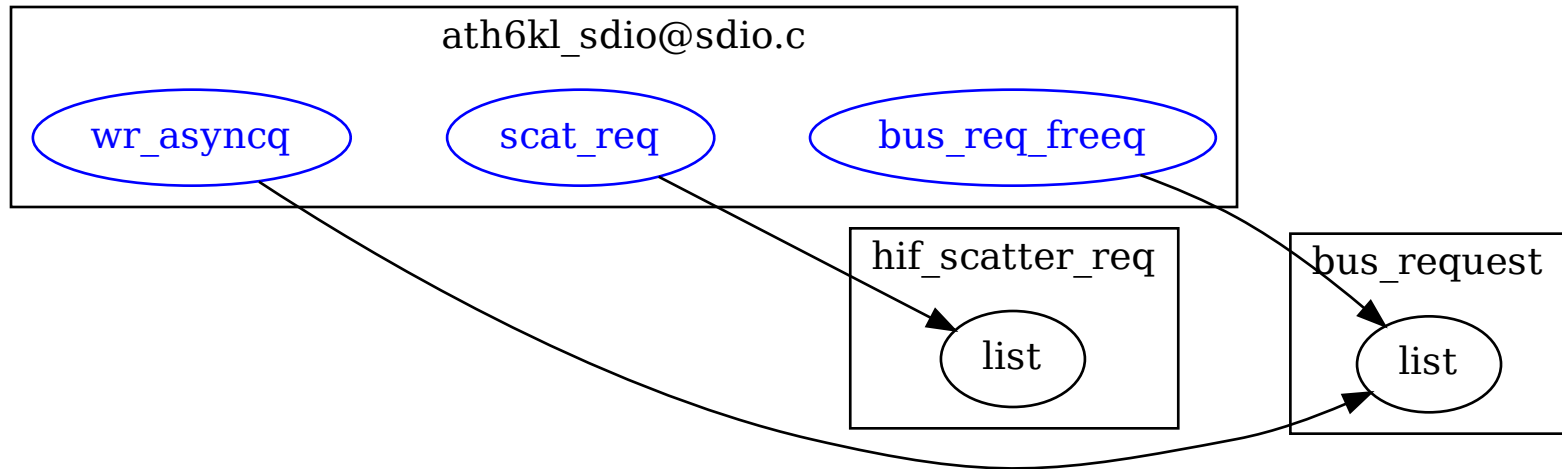


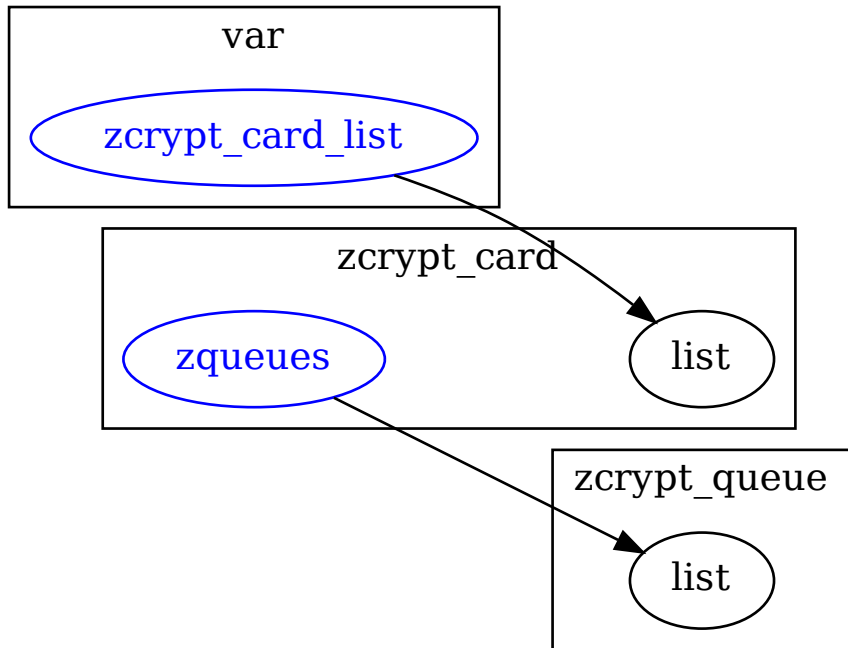


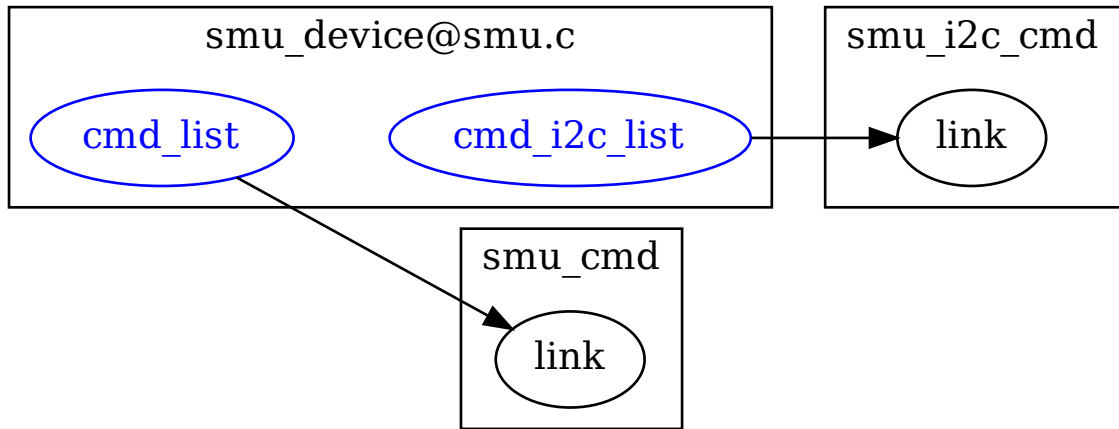


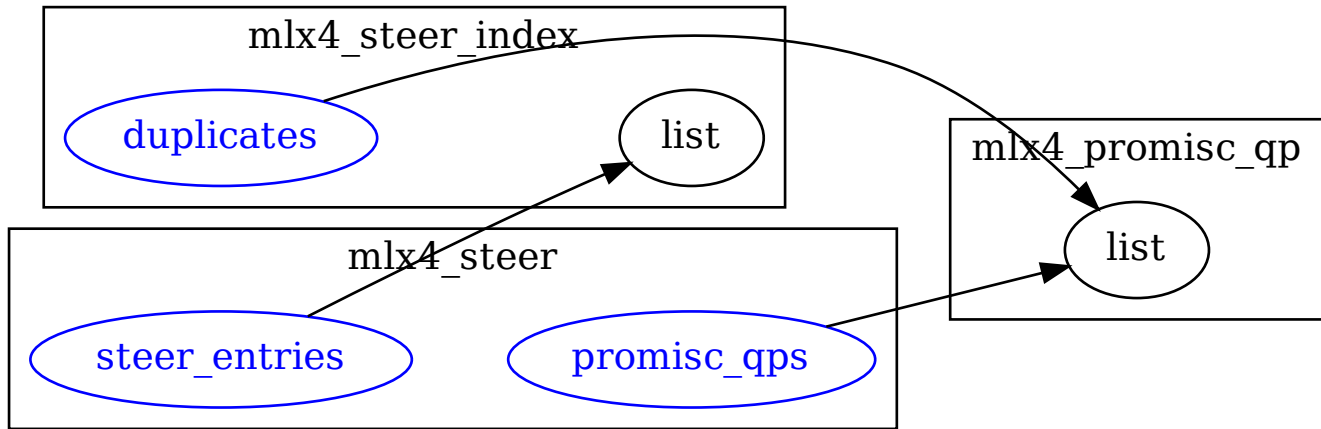


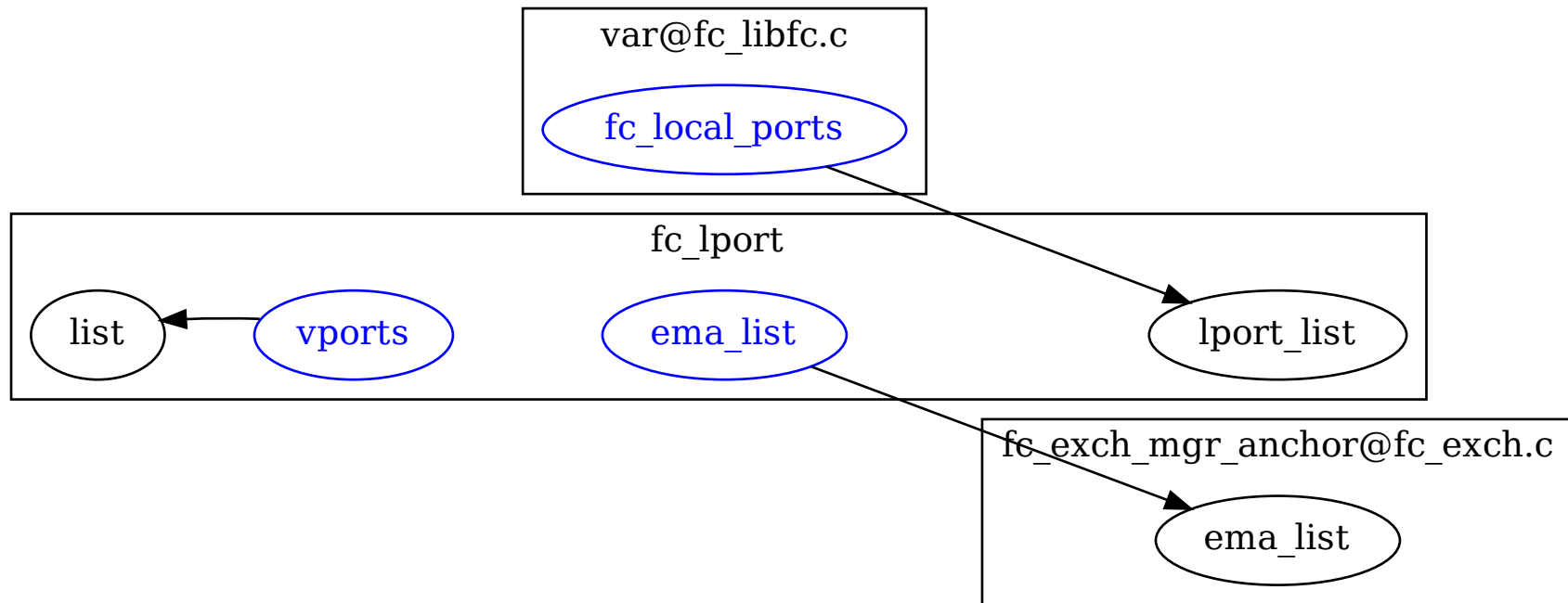


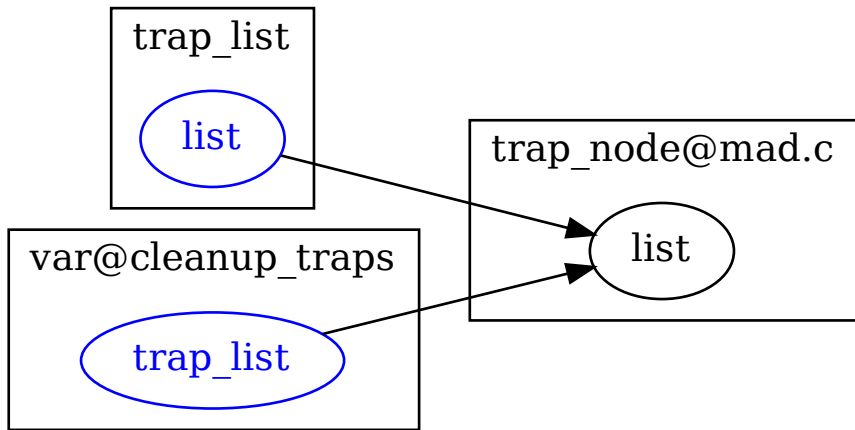


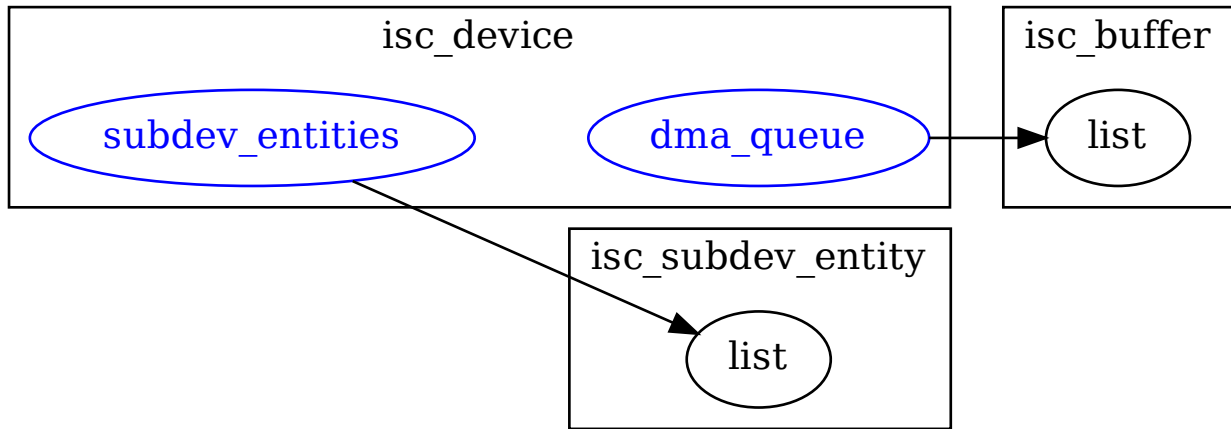


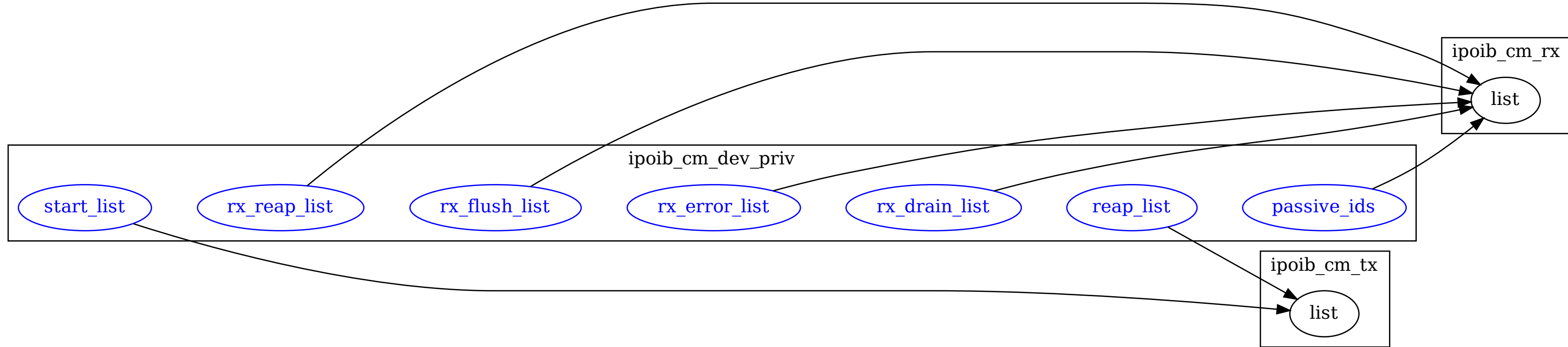












var@list_test_list_for_each_entry

list

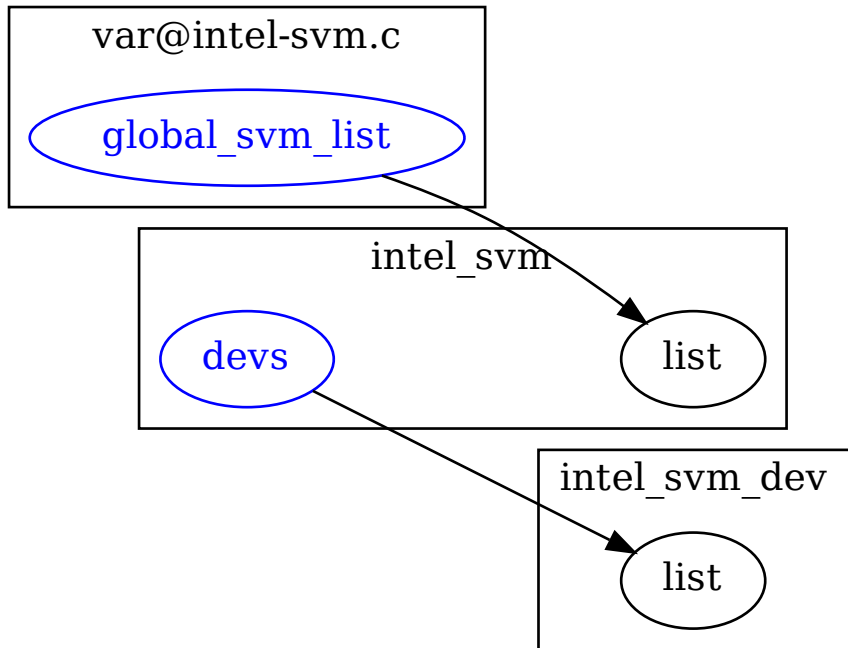
var@list_test_list_for_each_entry_reverse

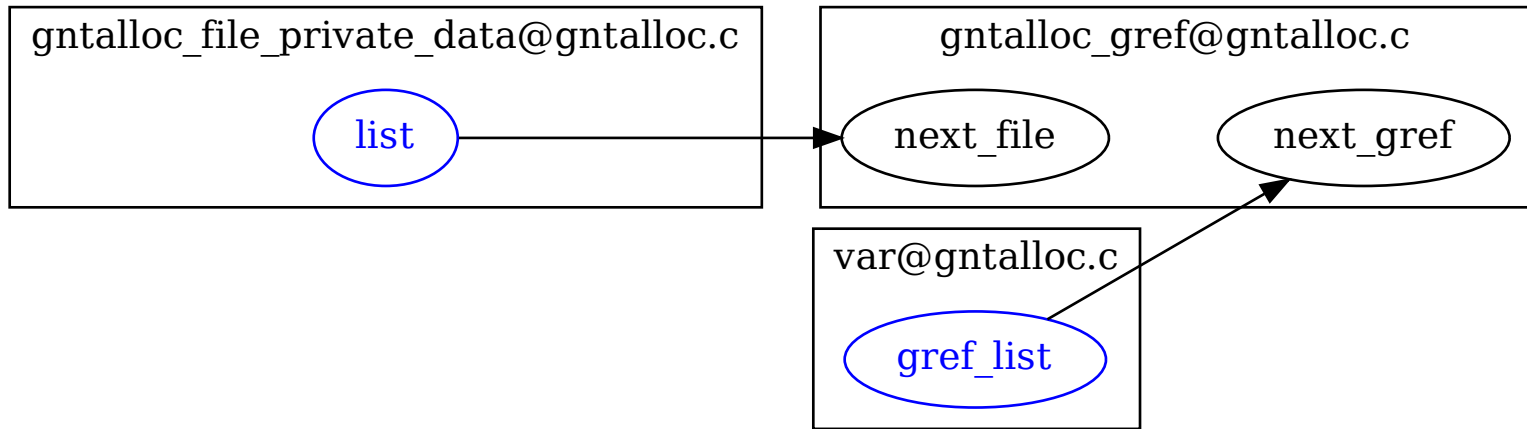
list

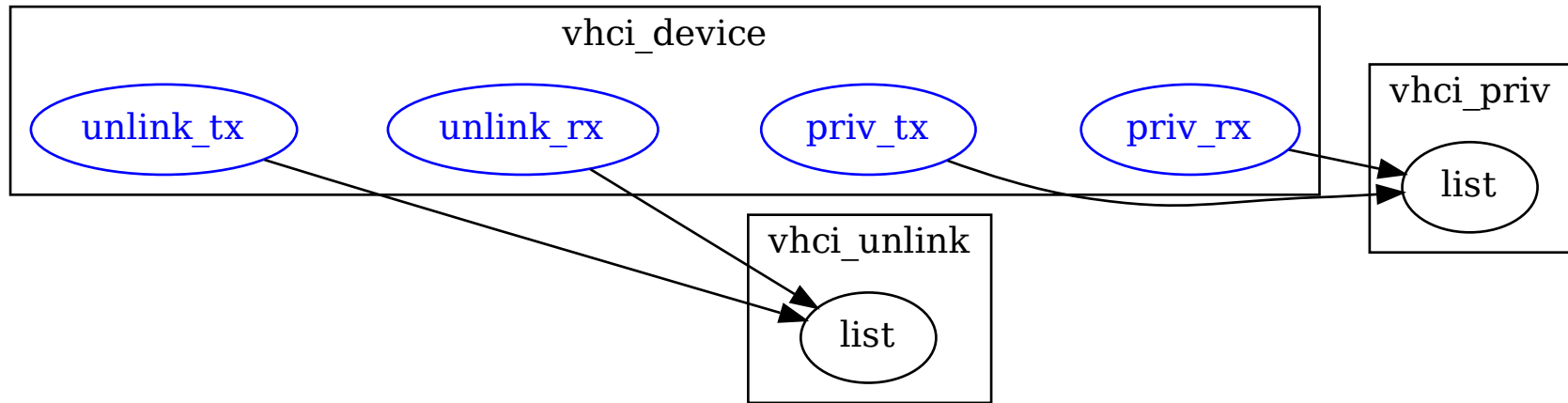
list_test_struct@list-test.c

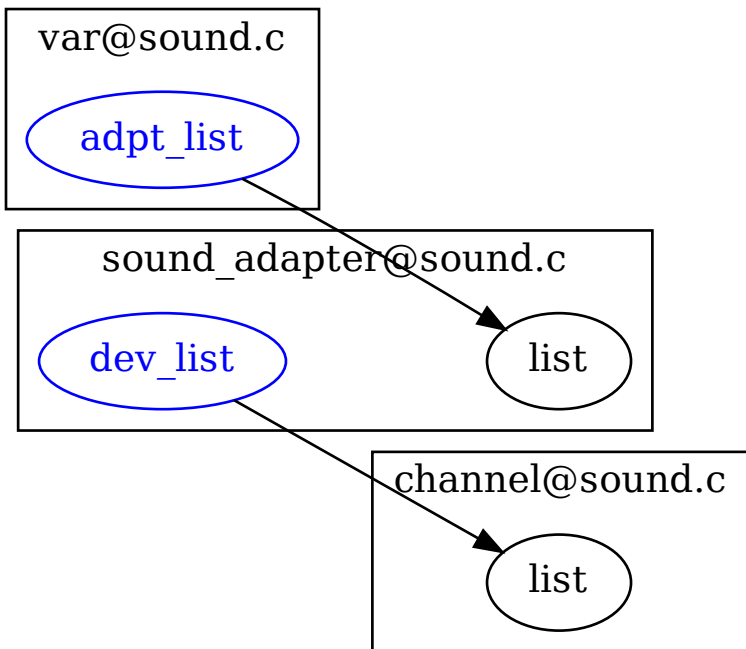
list

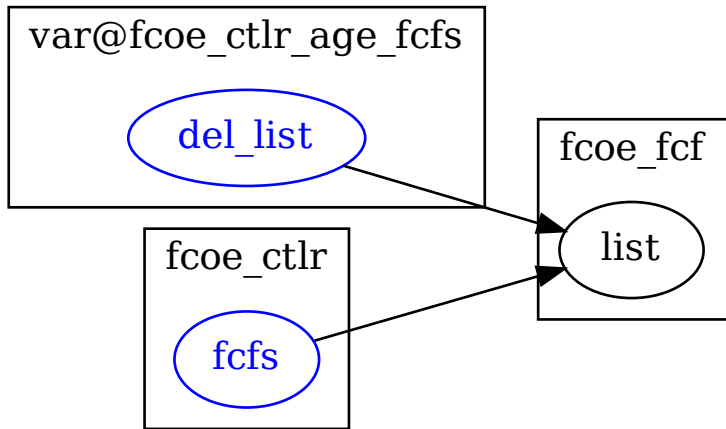
The diagram illustrates the relationship between three variables named 'list'. On the left, there are two rectangular boxes. The top box contains the text 'var@list_test_list_for_each_entry' and a blue oval containing the word 'list'. The bottom box contains the text 'var@list_test_list_for_each_entry_reverse' and another blue oval containing the word 'list'. On the right, there is a rectangular box containing the text 'list_test_struct@list-test.c' and a black oval containing the word 'list'. Two arrows originate from the blue ovals on the left and point to the black oval on the right. The arrow from the top-left oval points to the upper part of the right oval, and the arrow from the bottom-left oval points to the lower part of the right oval, indicating that both left variables point to the same 'list' variable in the rightmost scope.

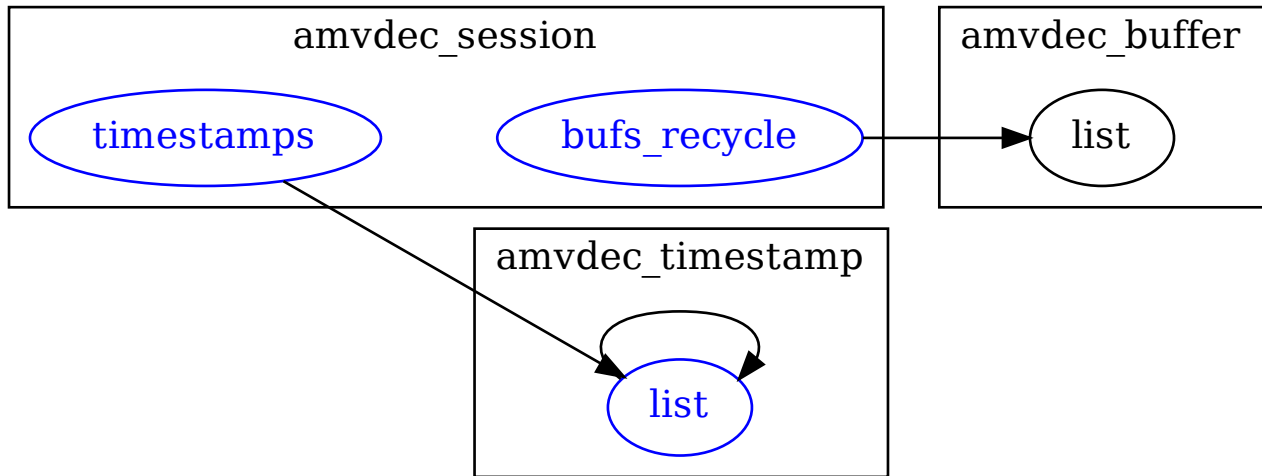


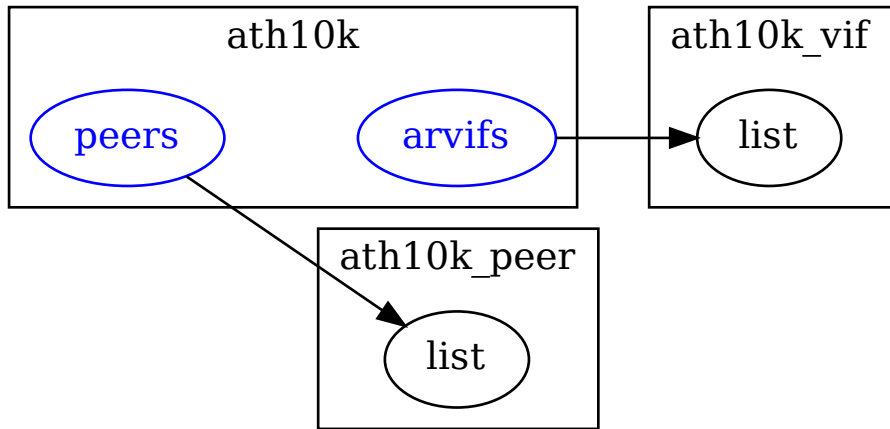


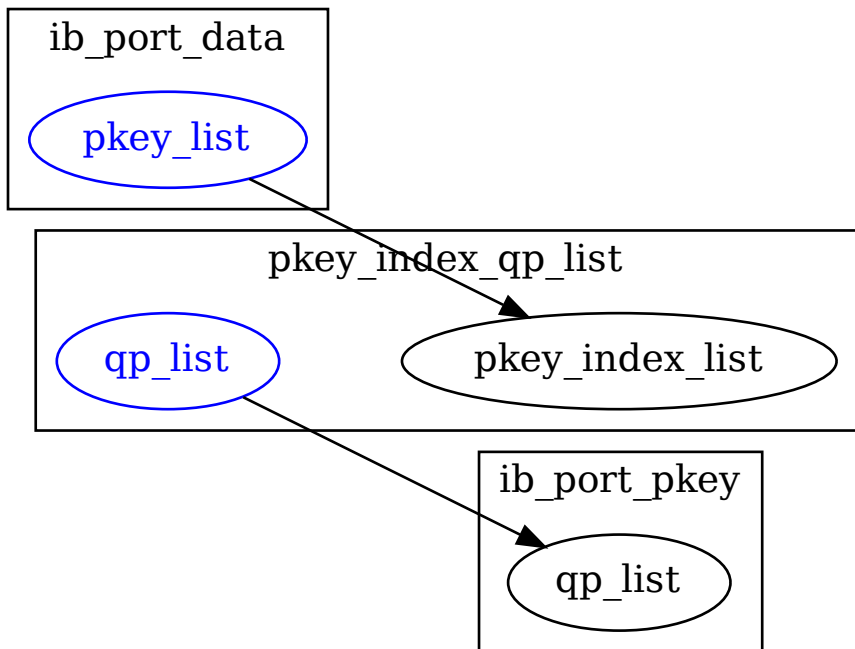


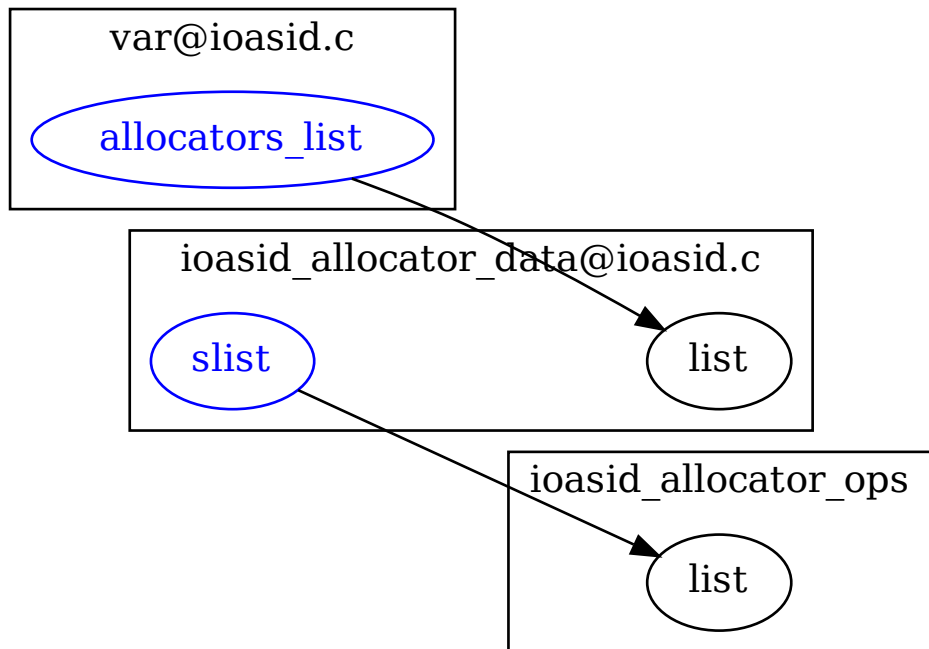












multipath@dm-mpath.c

priority_groups

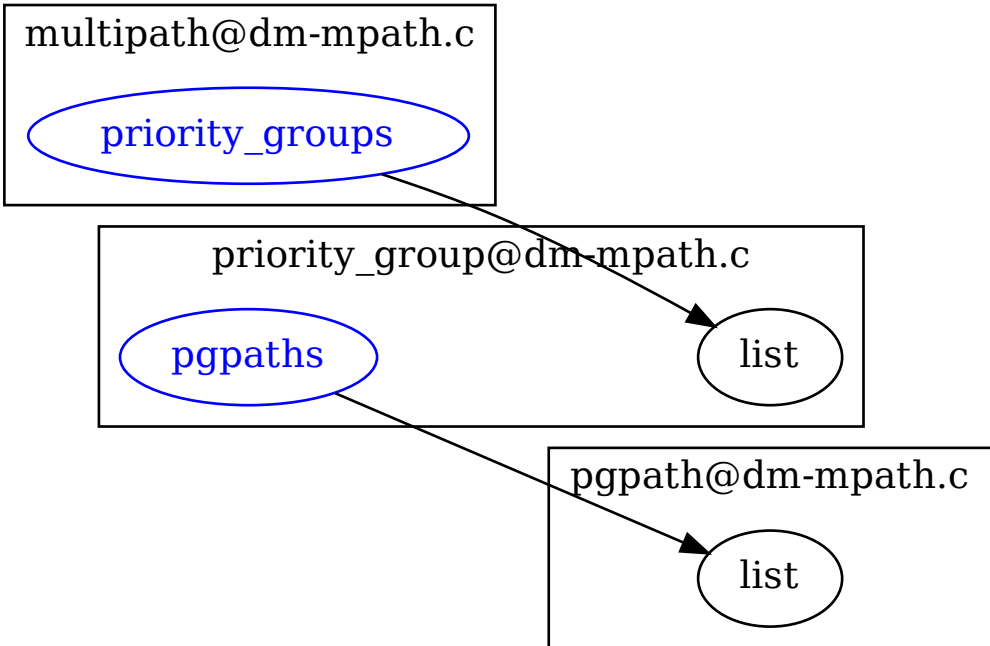
priority_group@dm-mpath.c

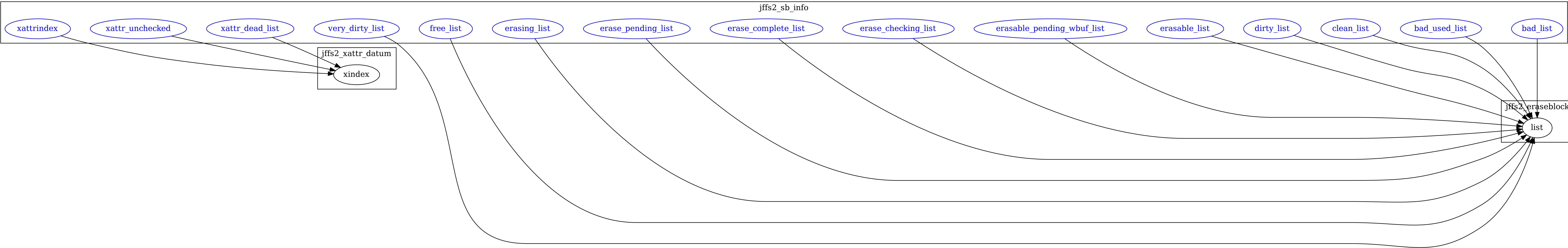
pgpaths

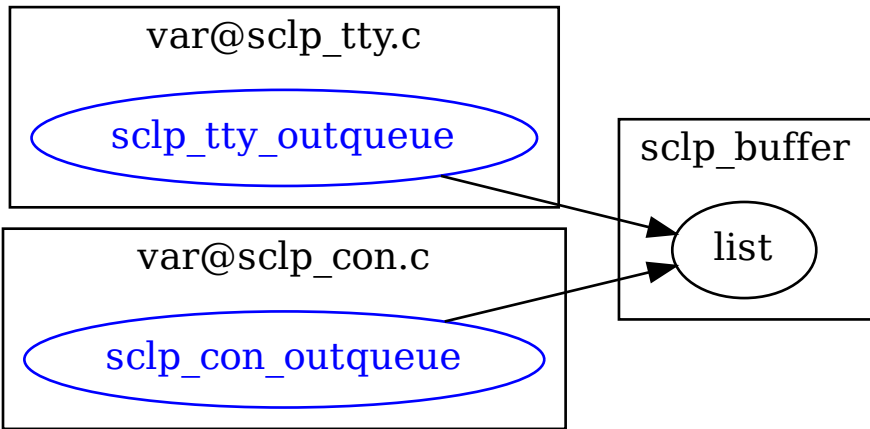
list

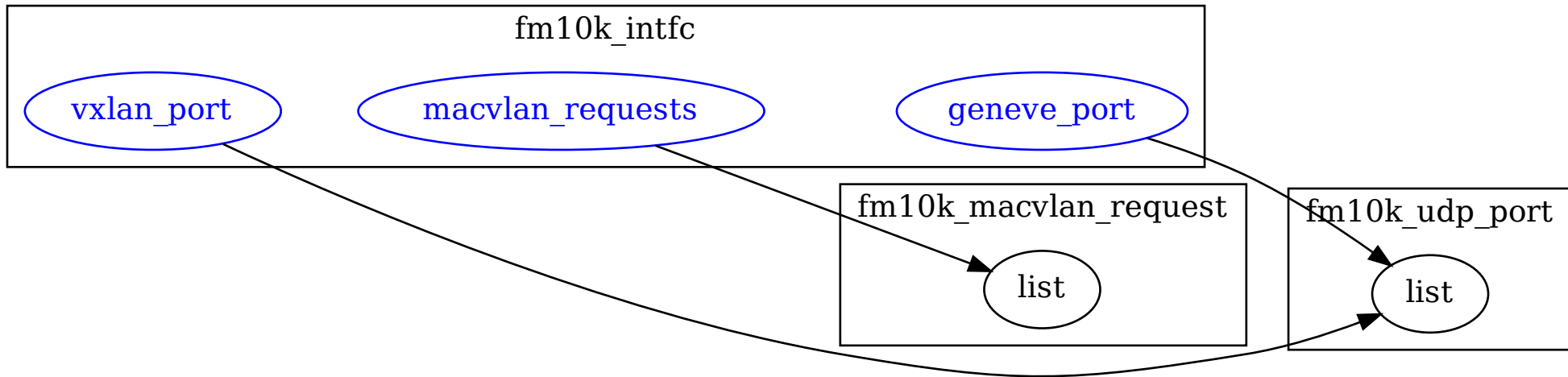
pgpath@dm-mpath.c

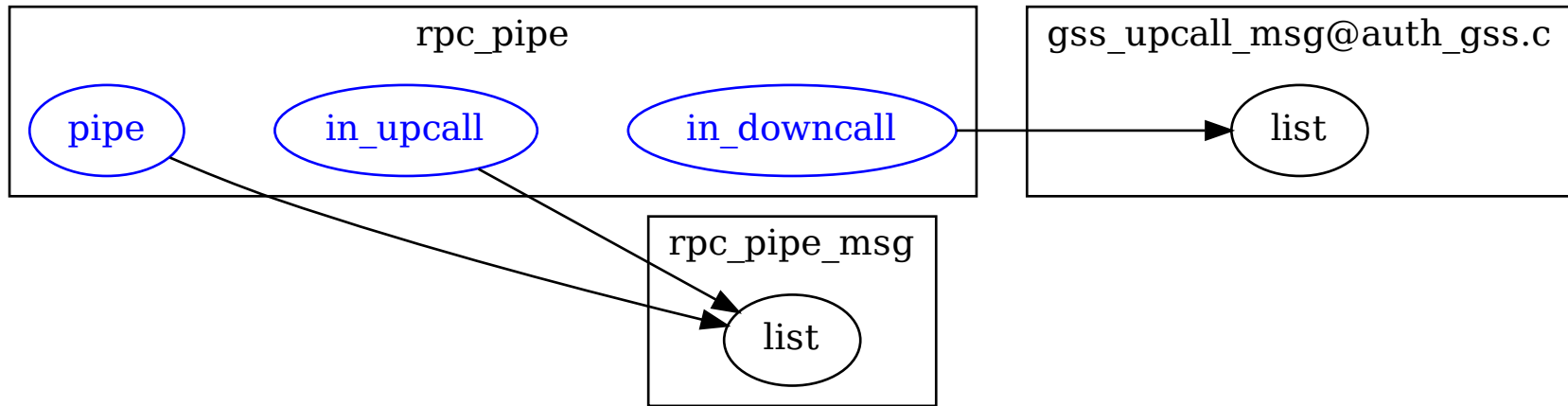
list

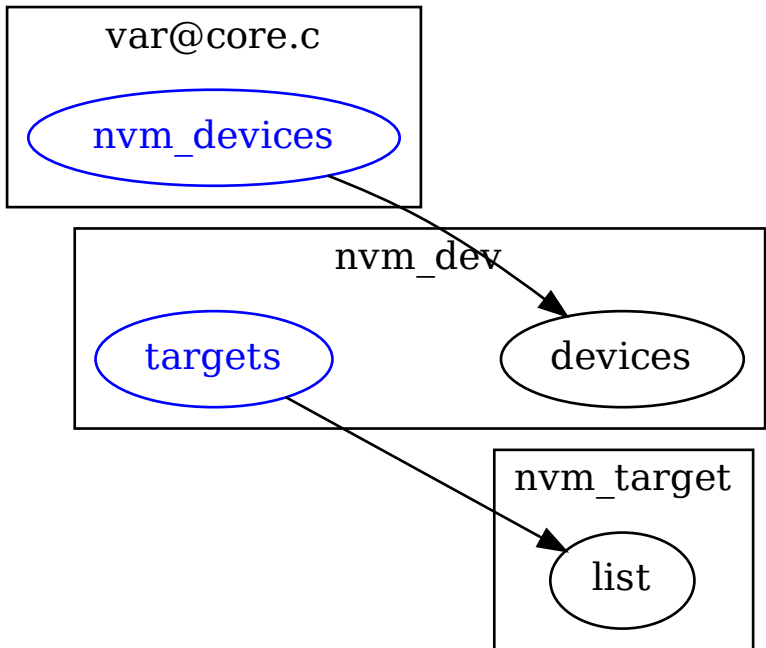


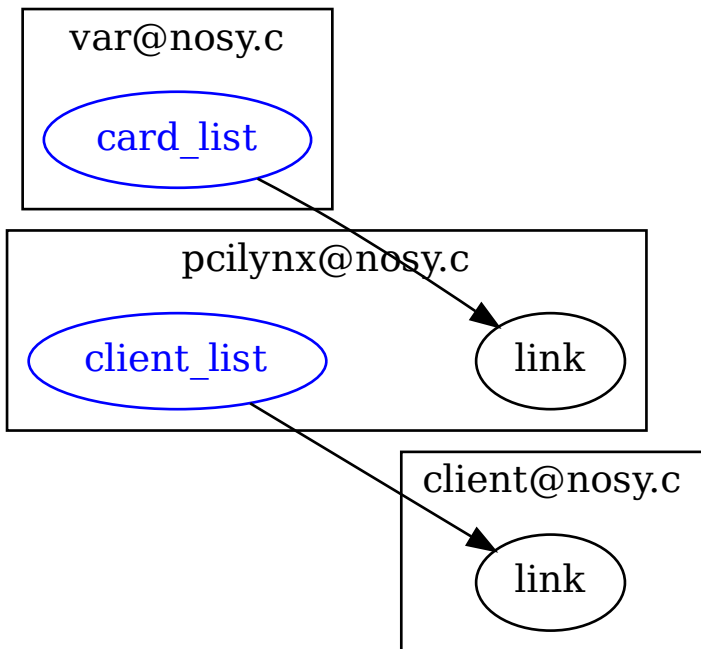


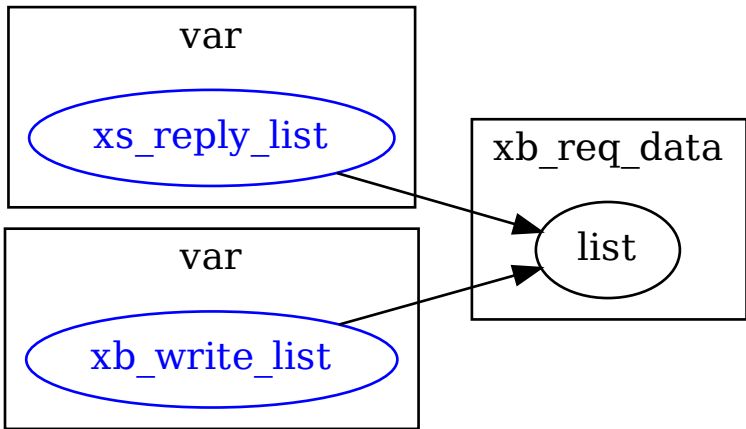


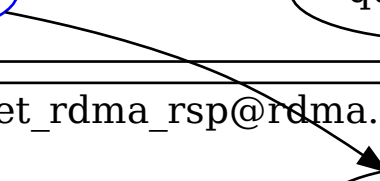
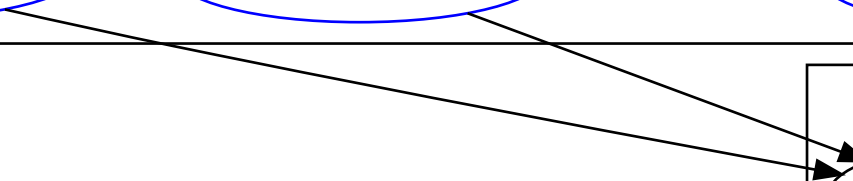
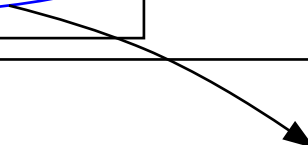
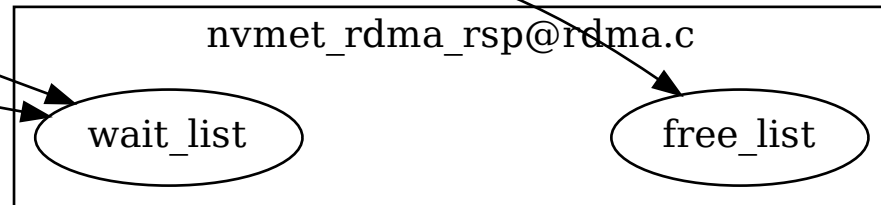
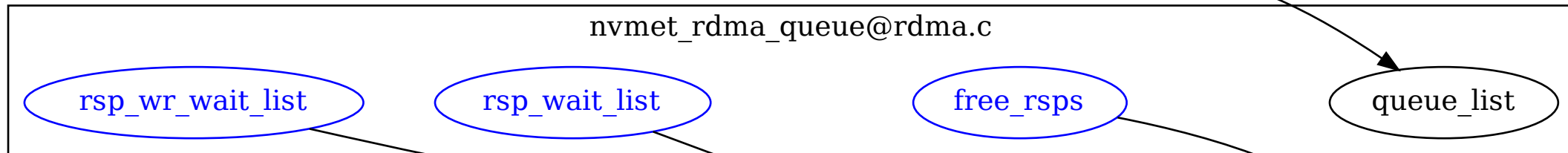
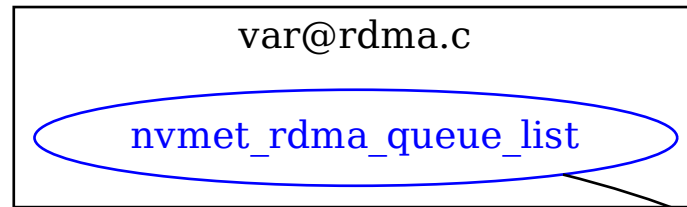


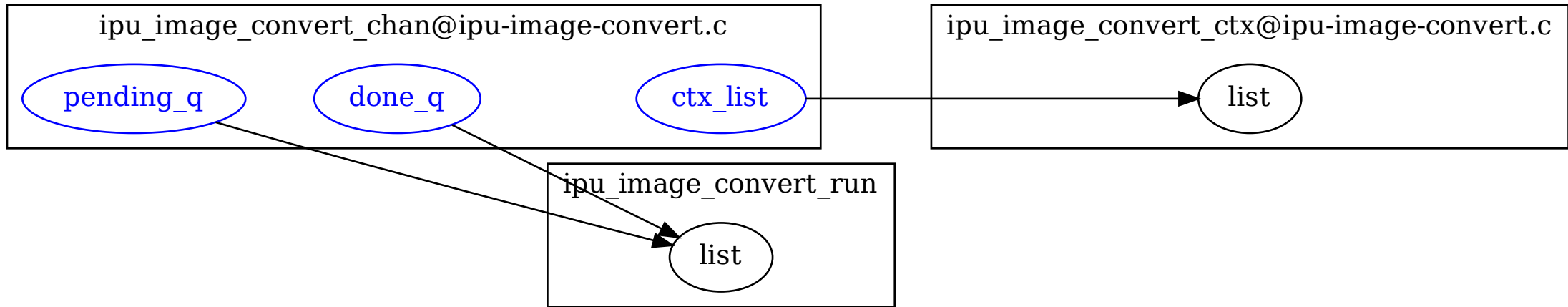


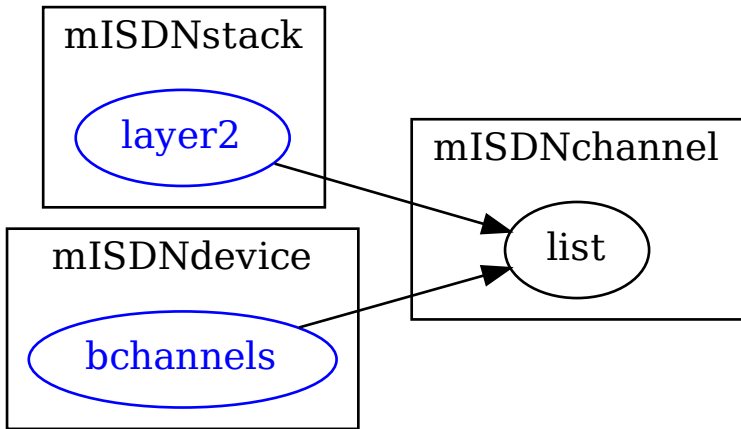


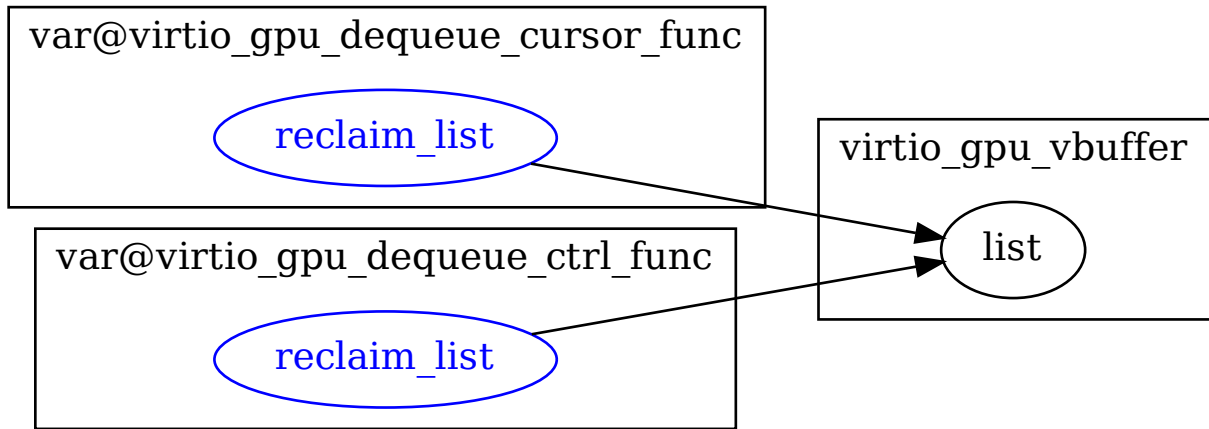


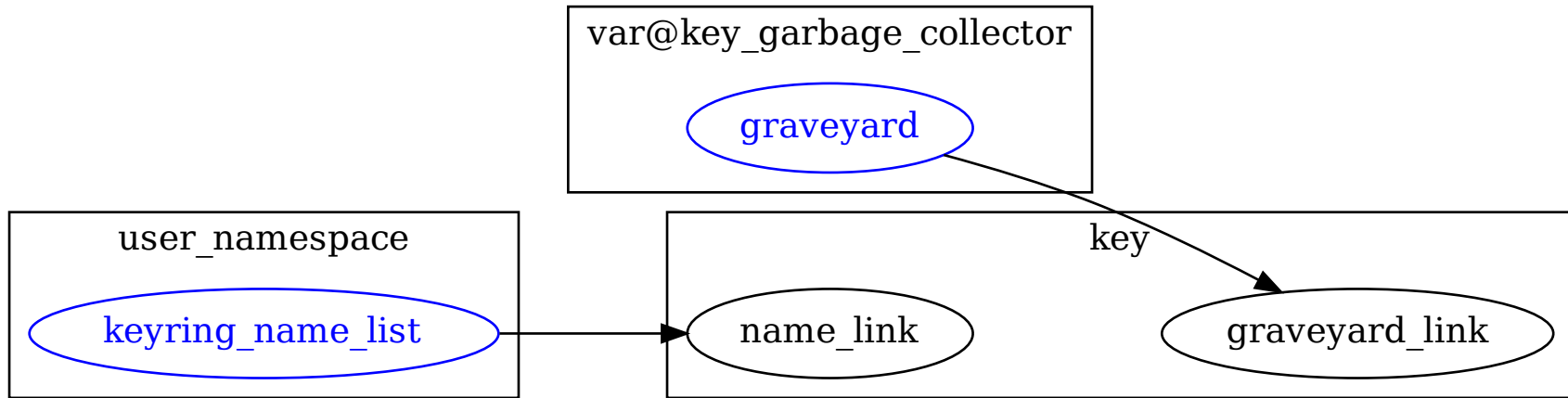


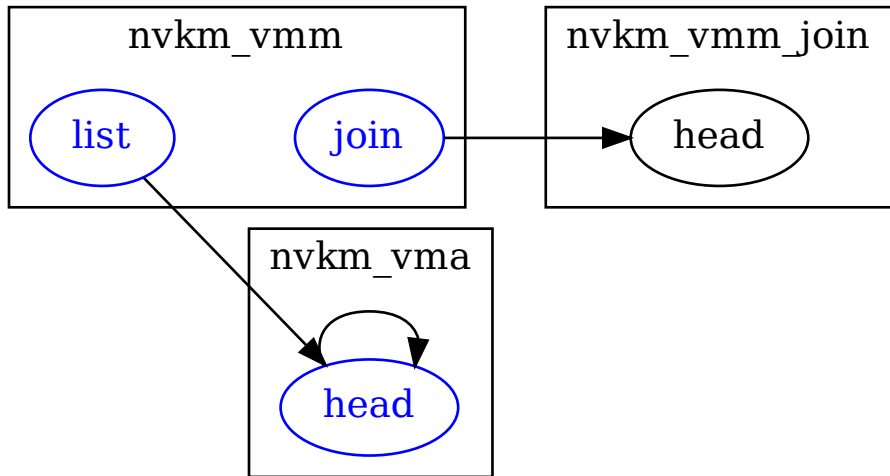


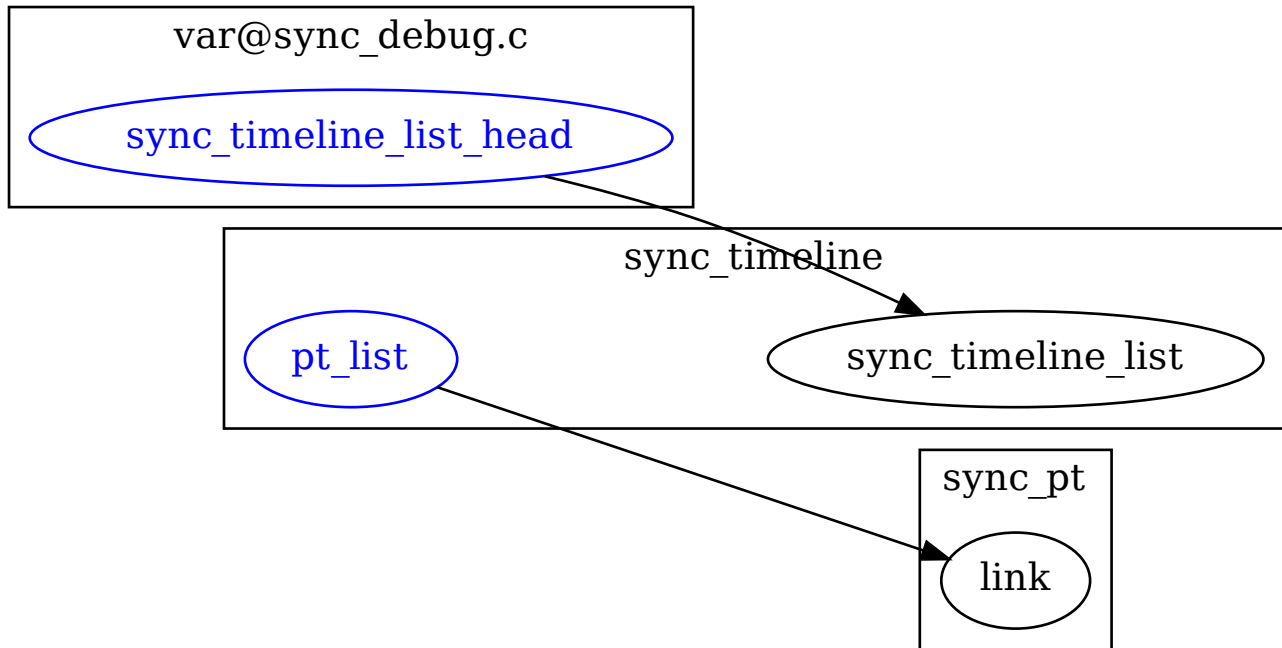


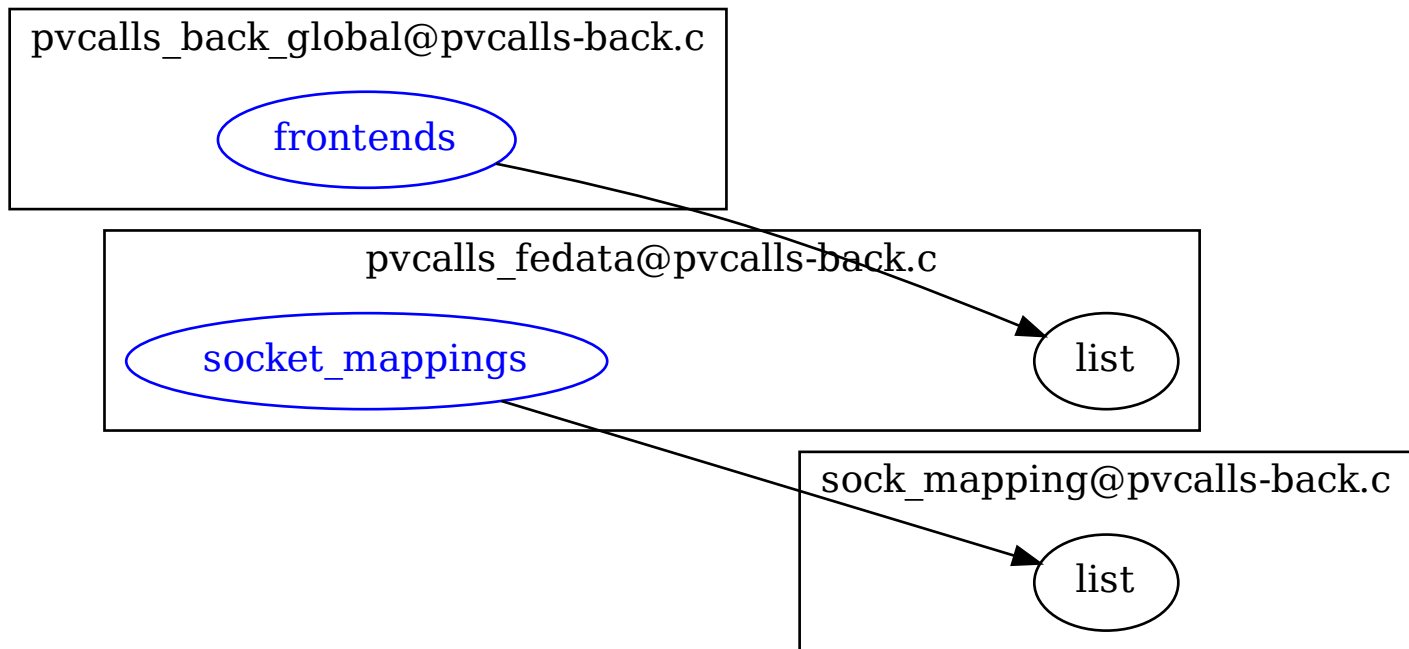


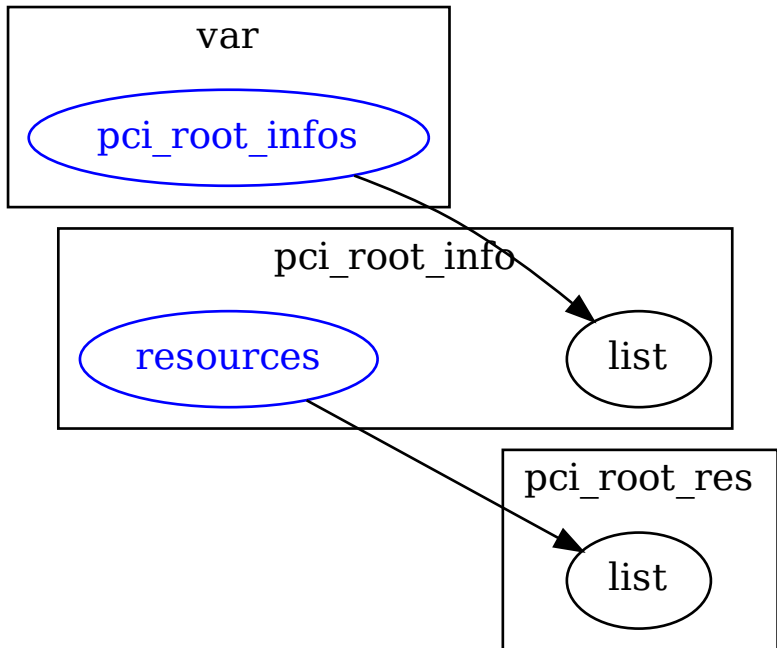


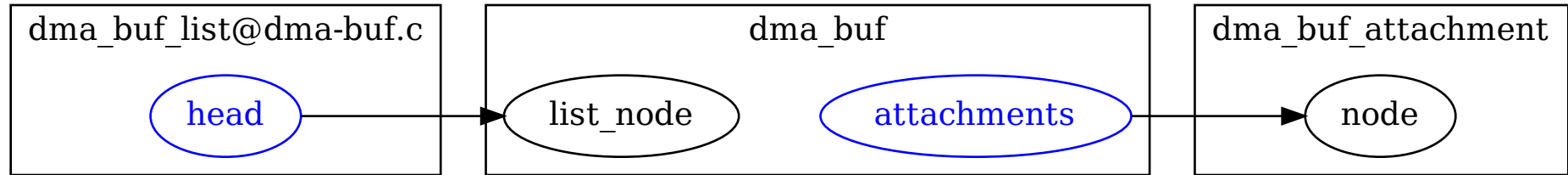


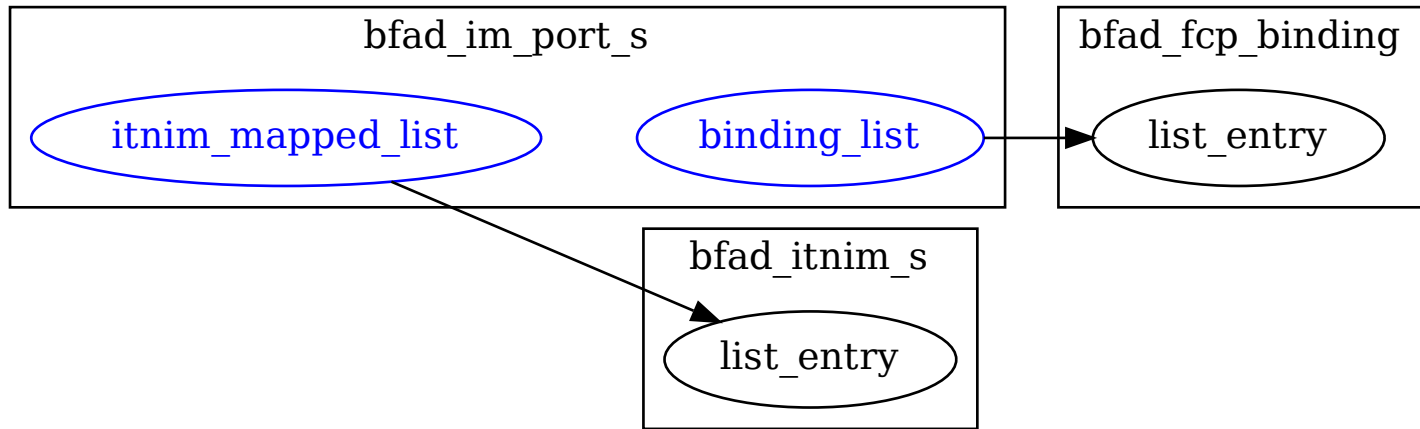


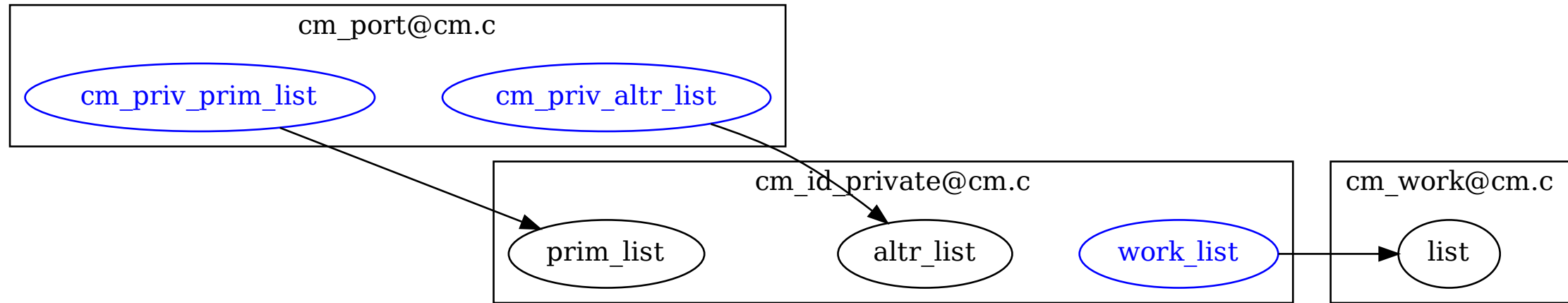


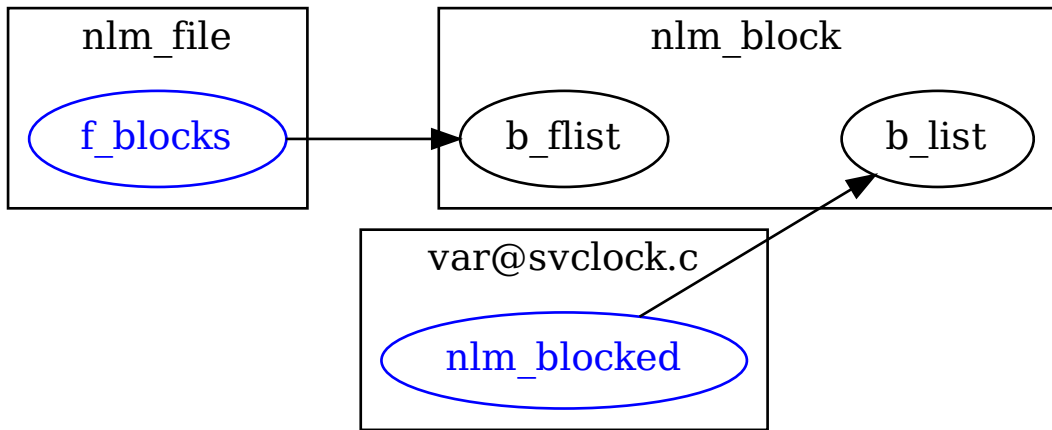


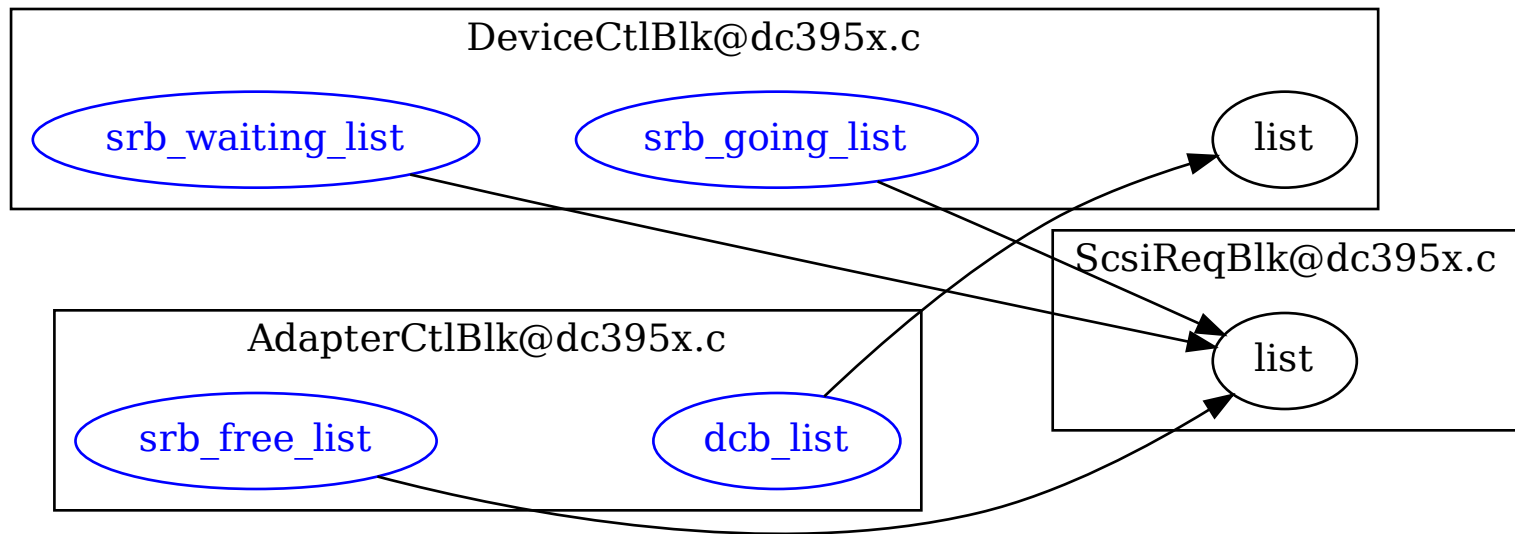


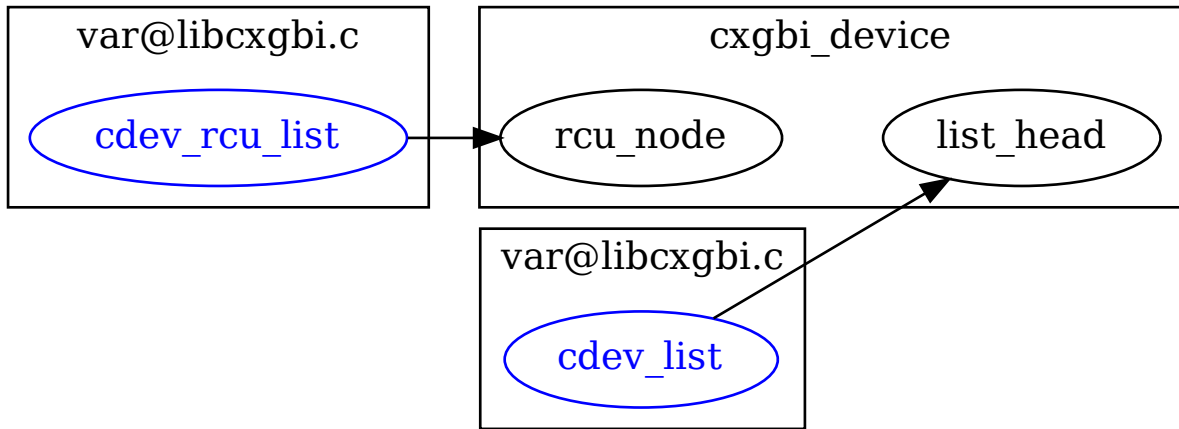


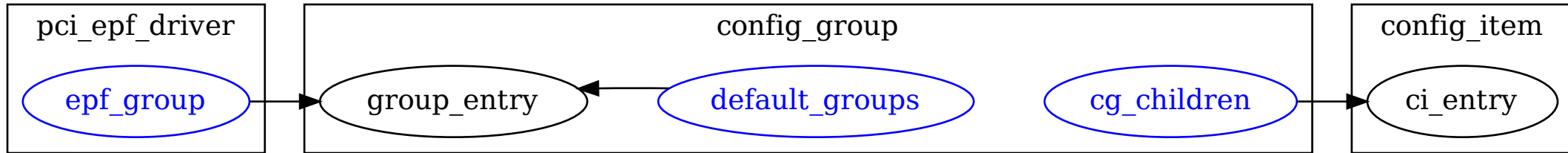


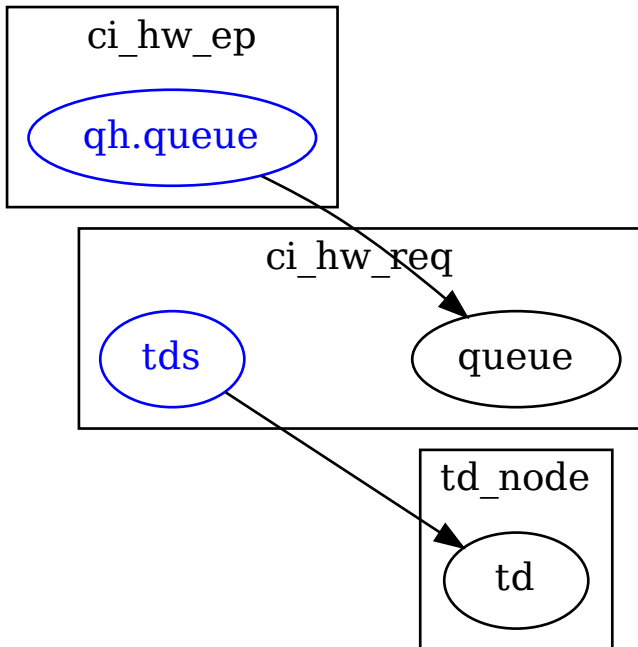


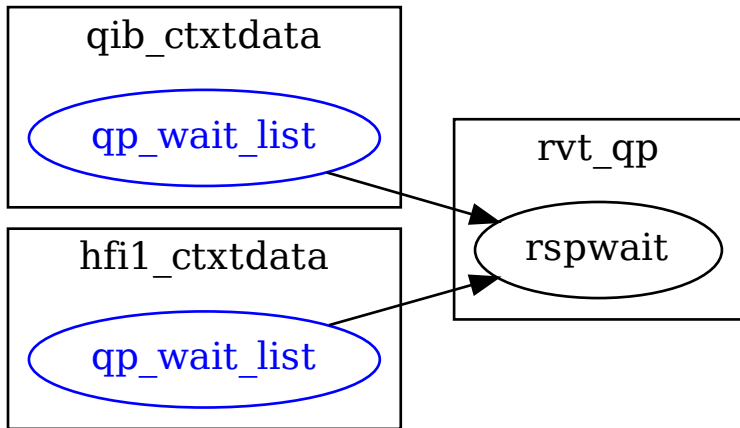


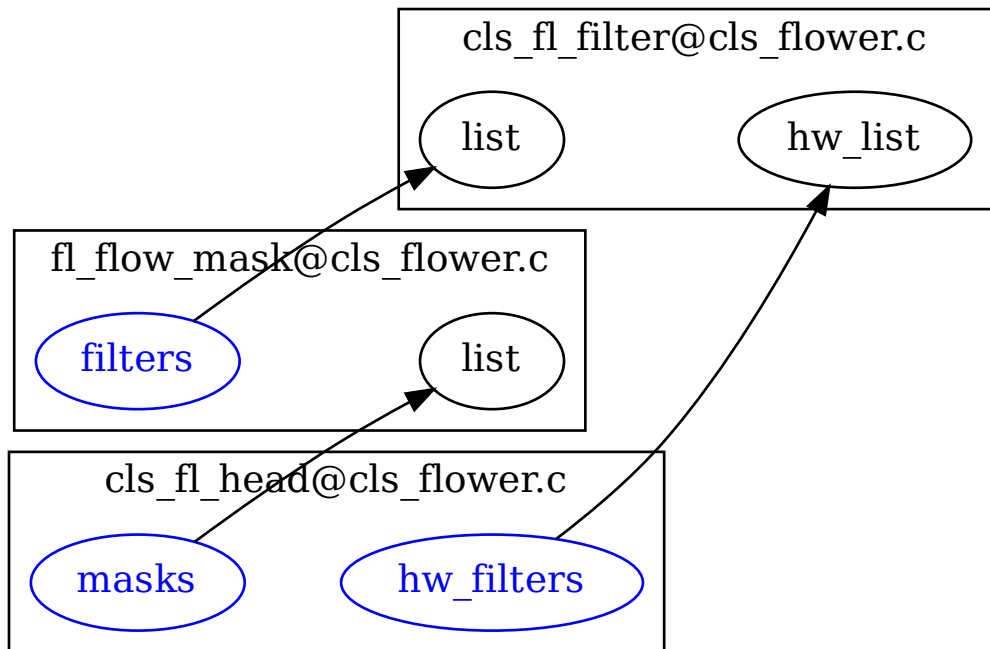


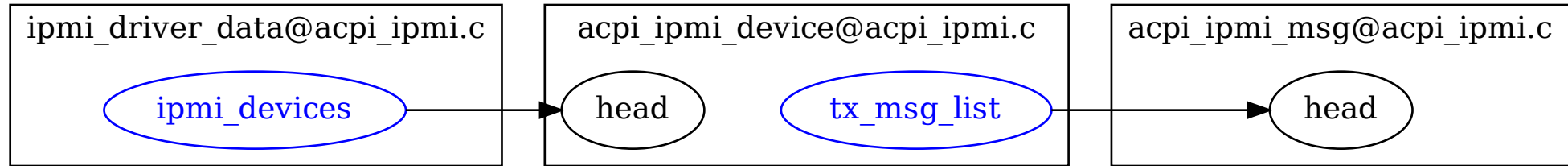


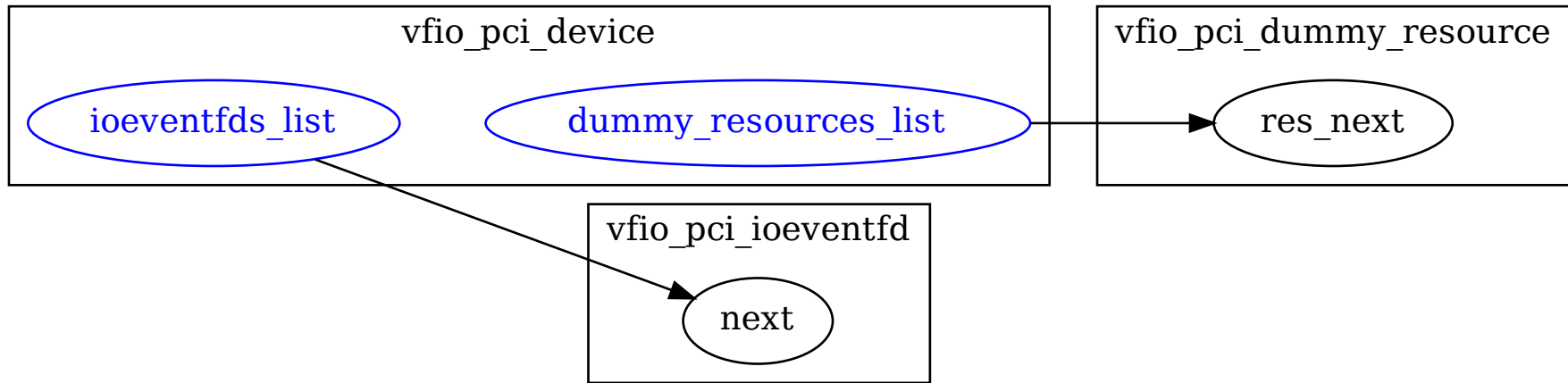


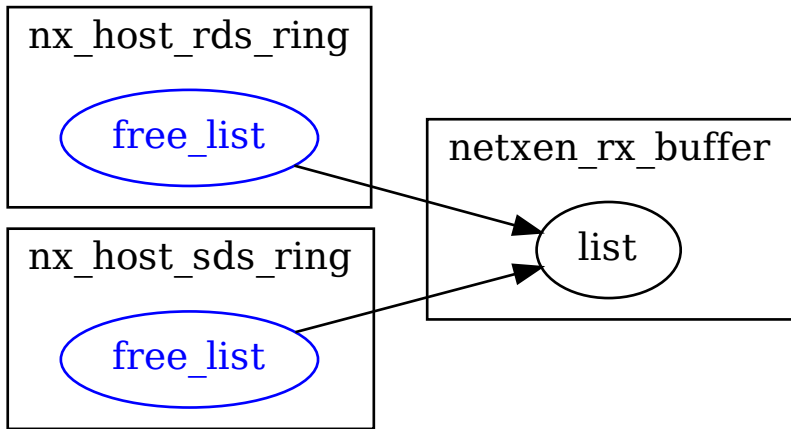


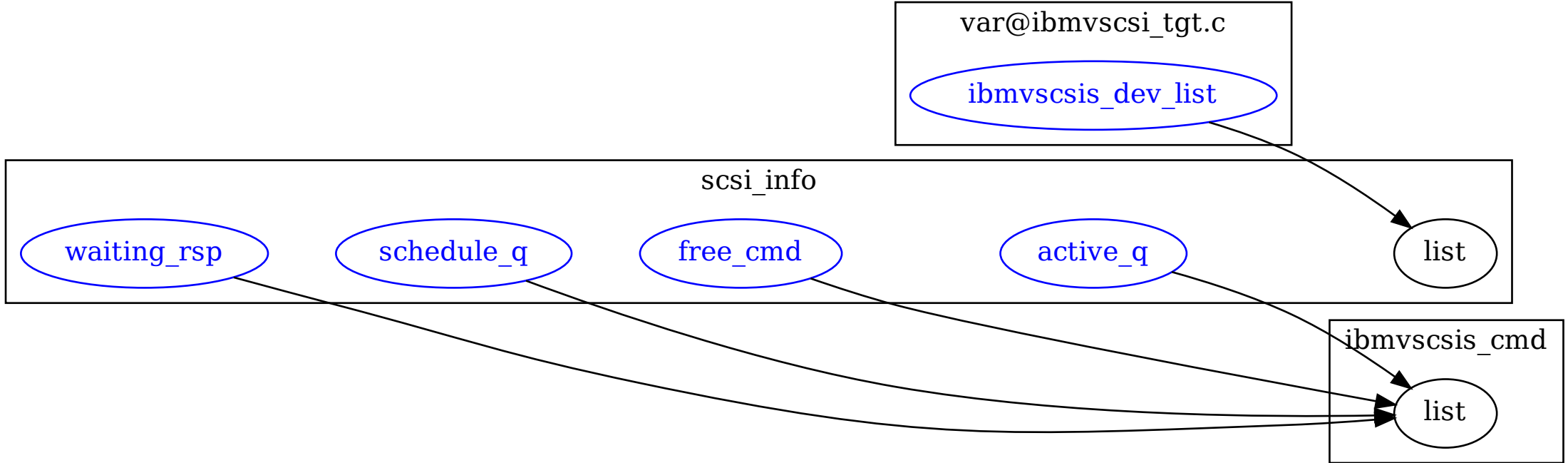
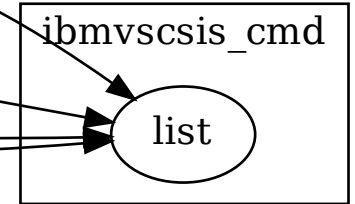
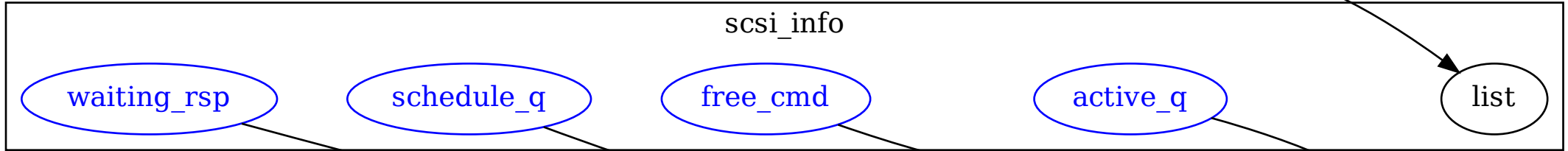
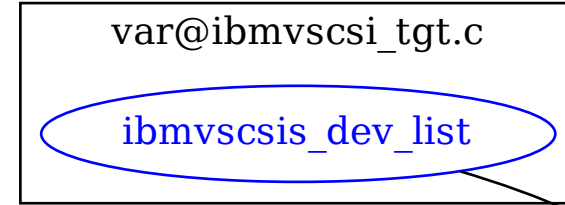


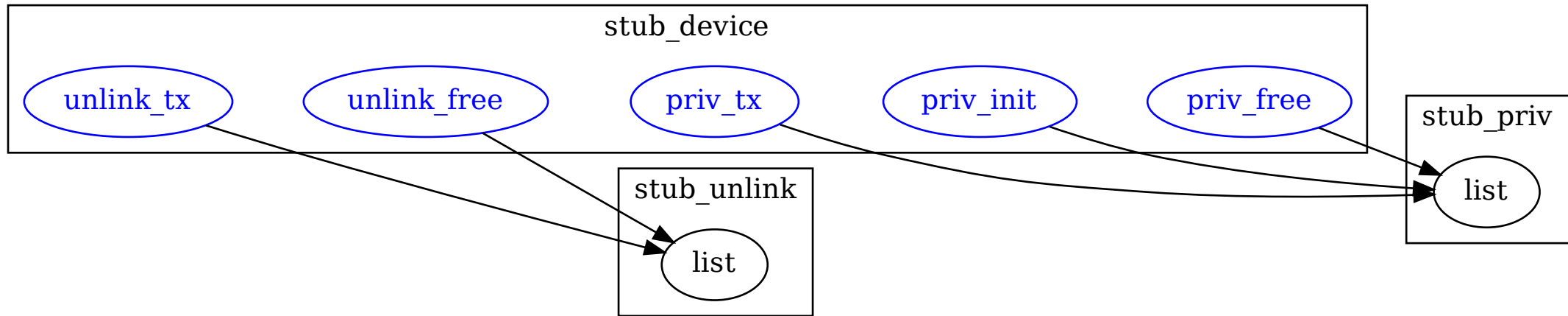


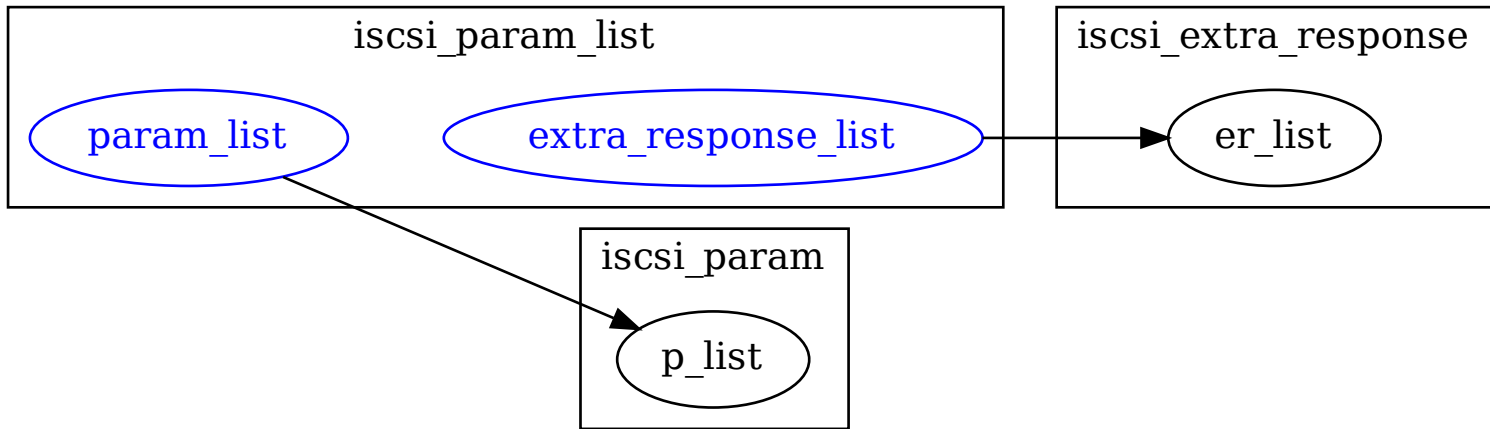


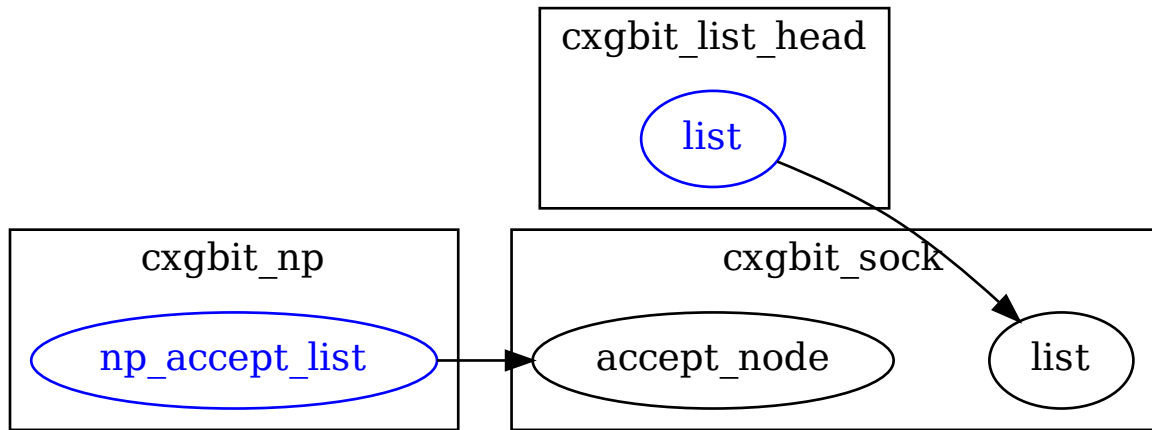


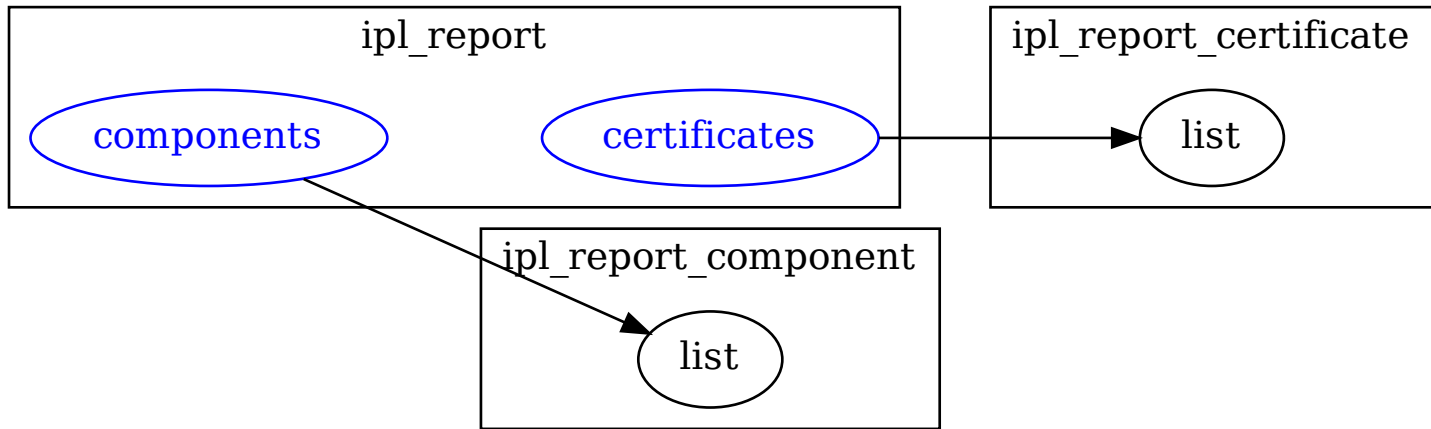


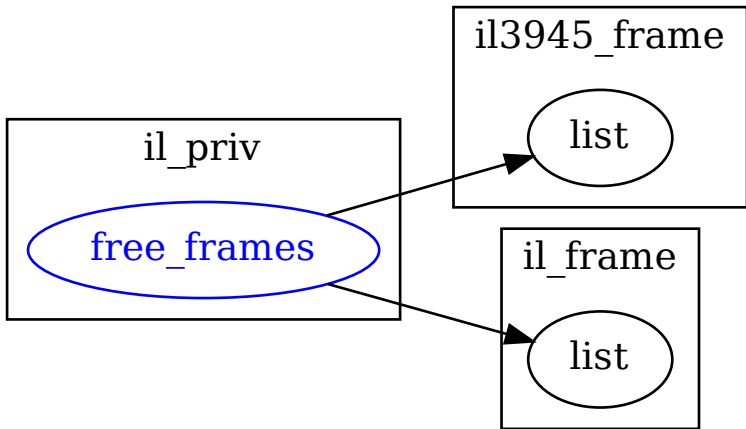


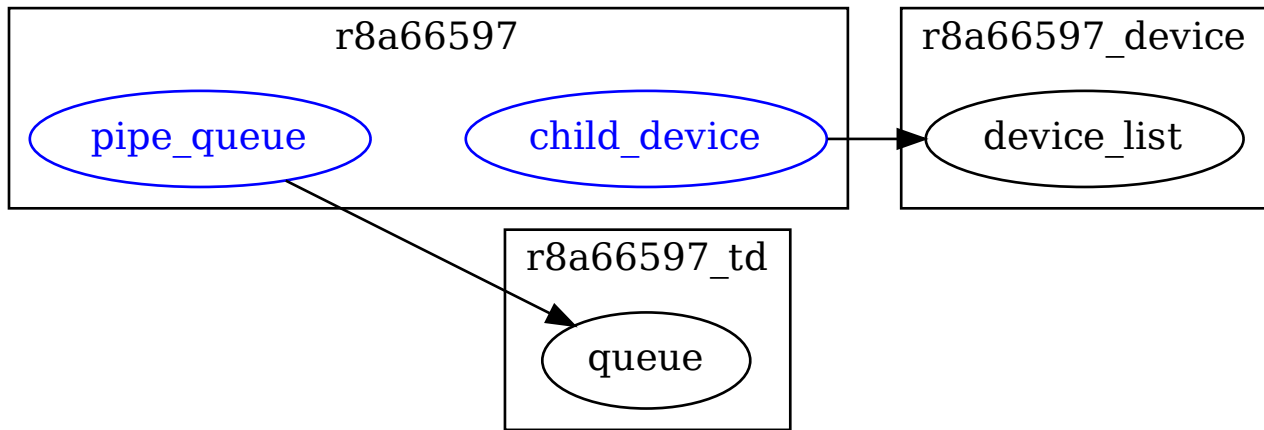


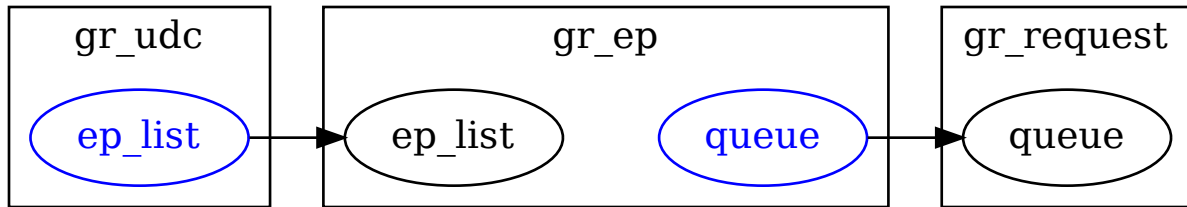


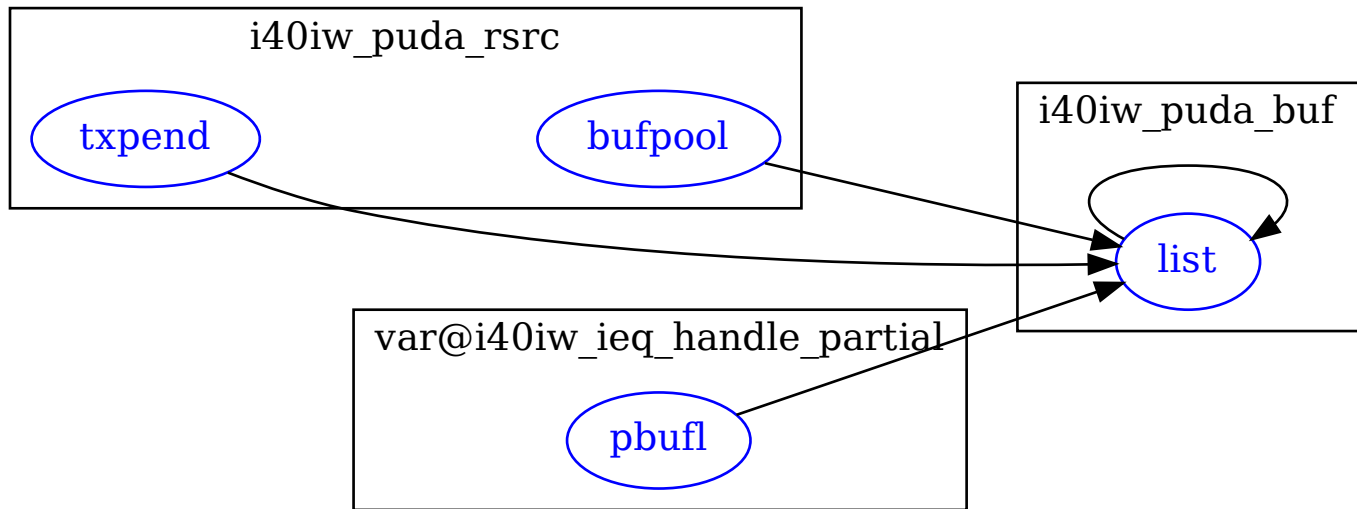


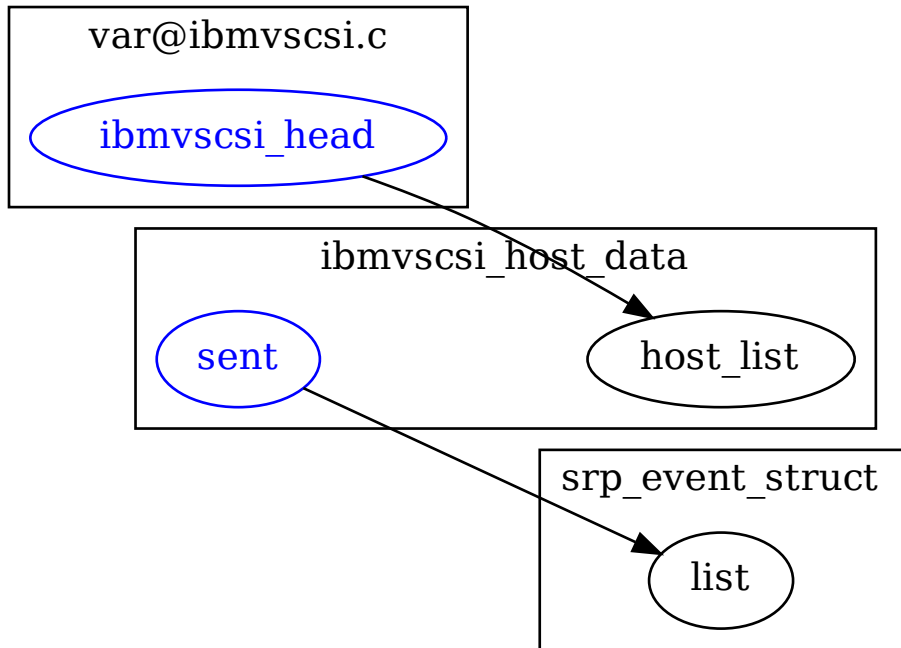


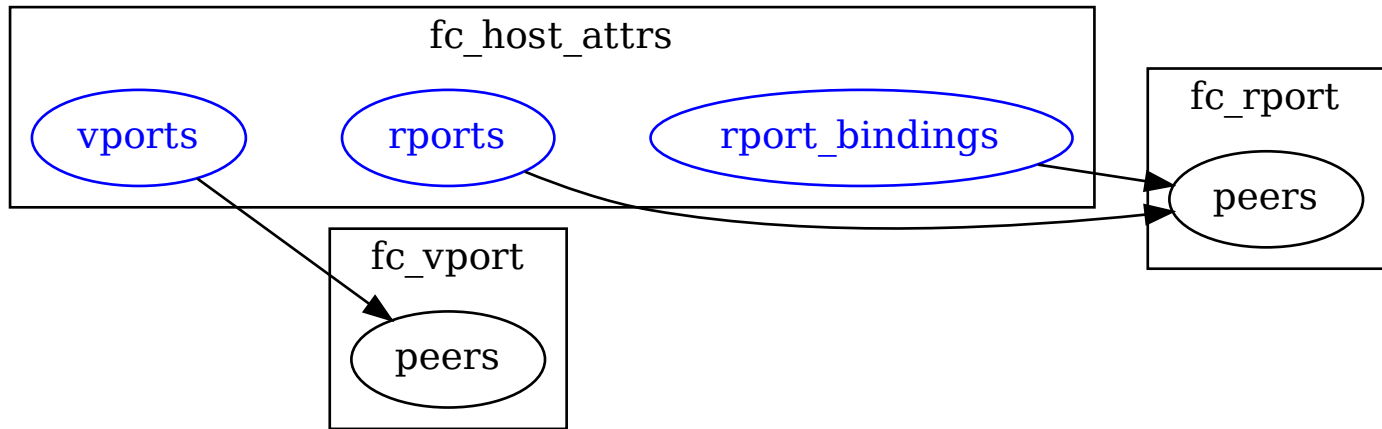


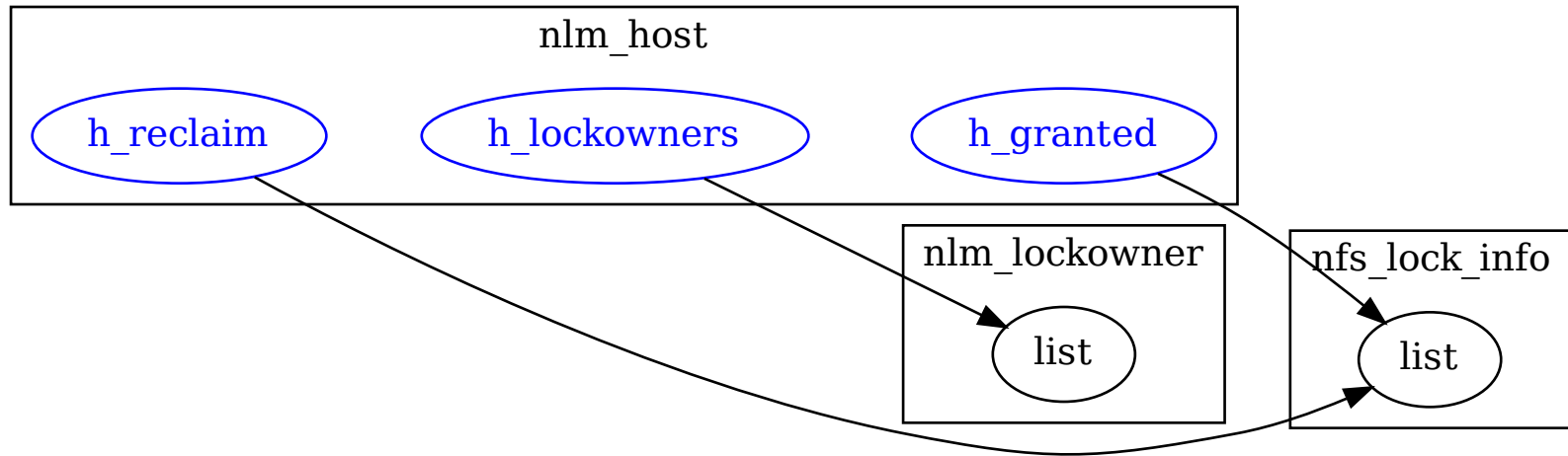


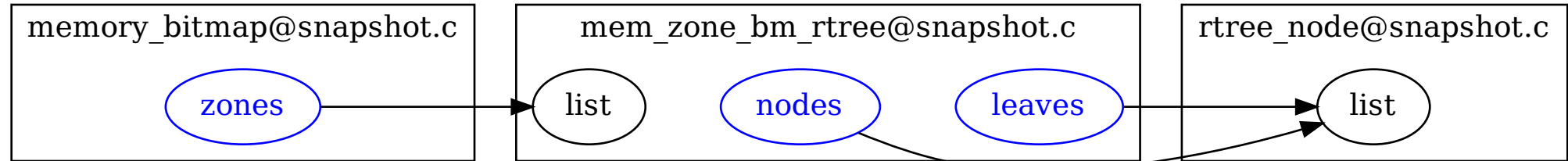


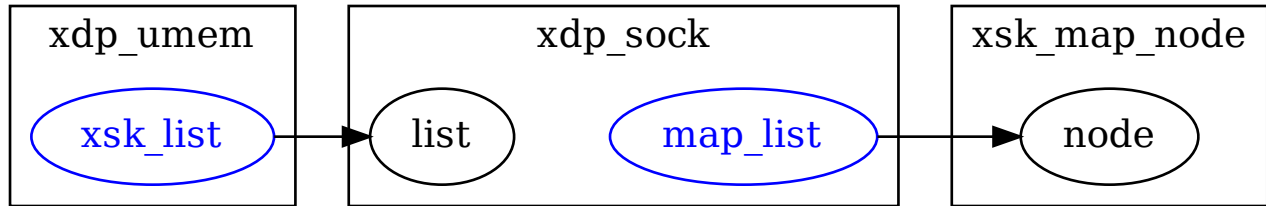


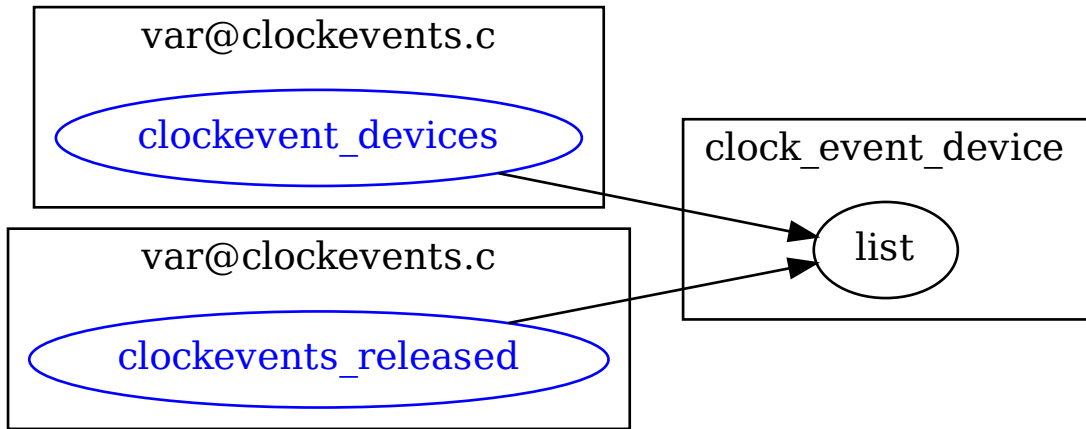












var@thermal_hwmon.c

thermal_hwmon_list

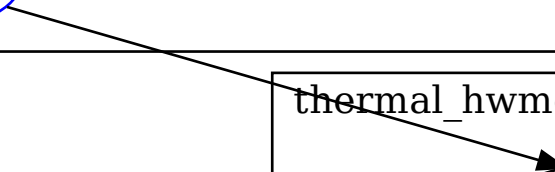
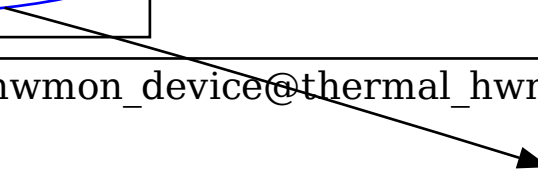
thermal_hwmon_device@thermal_hwmon.c

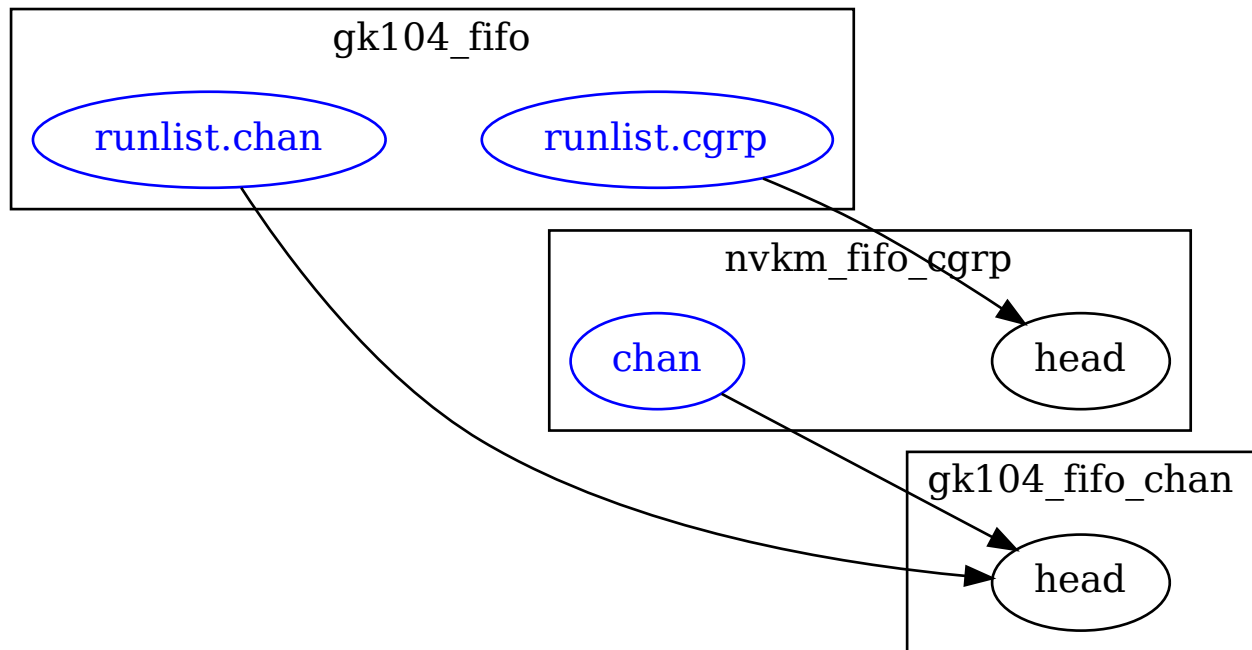
tz_list

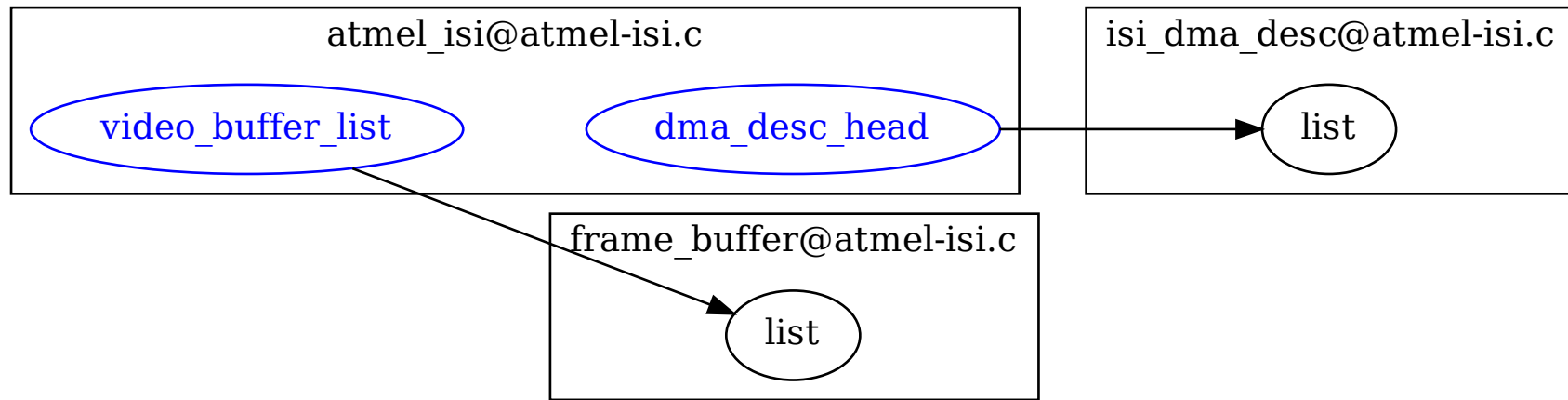
node

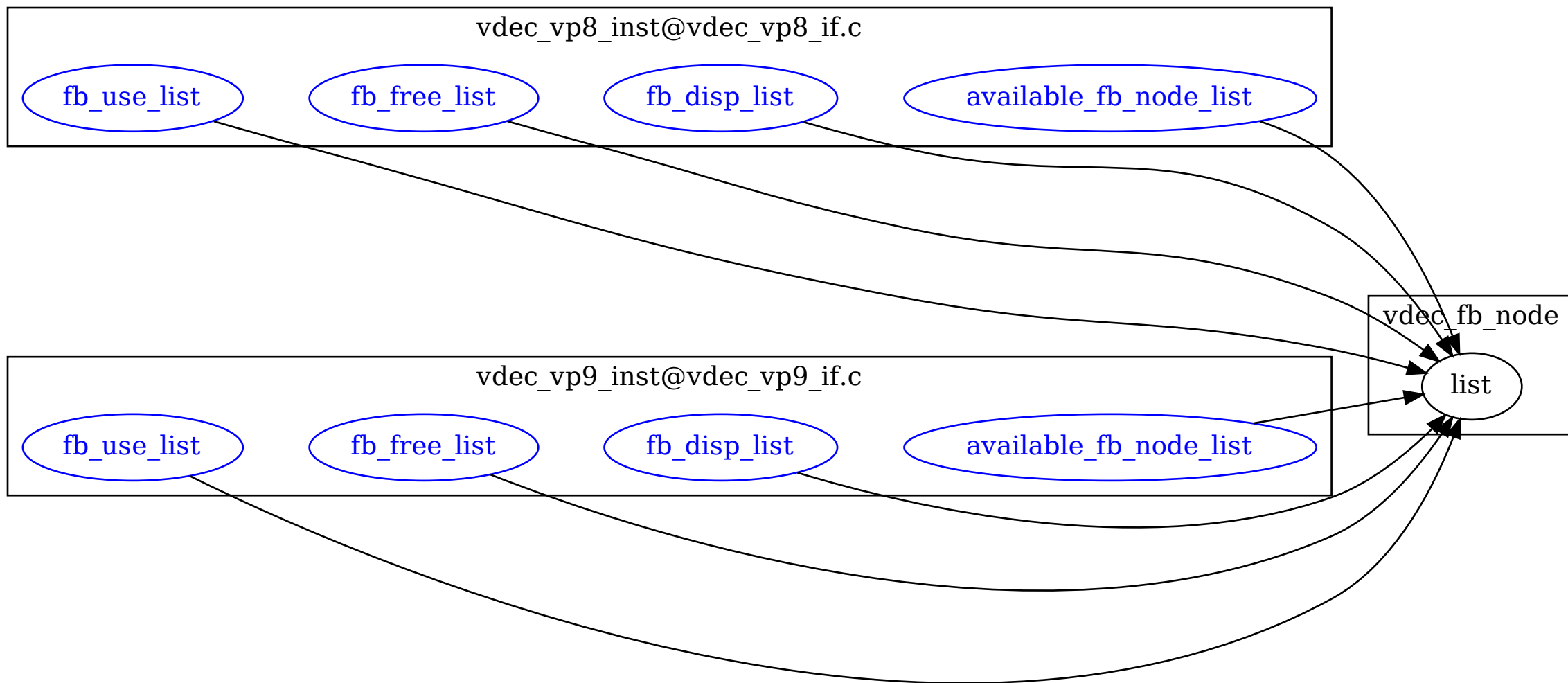
thermal_hwmon_temp@thermal_hwmon.c

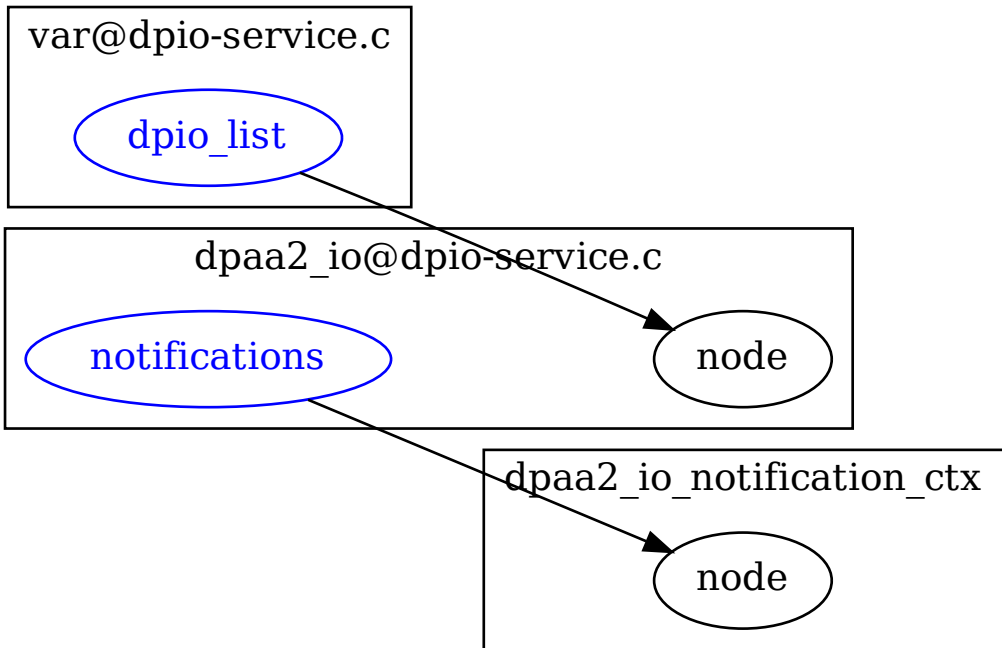
hwmon_node

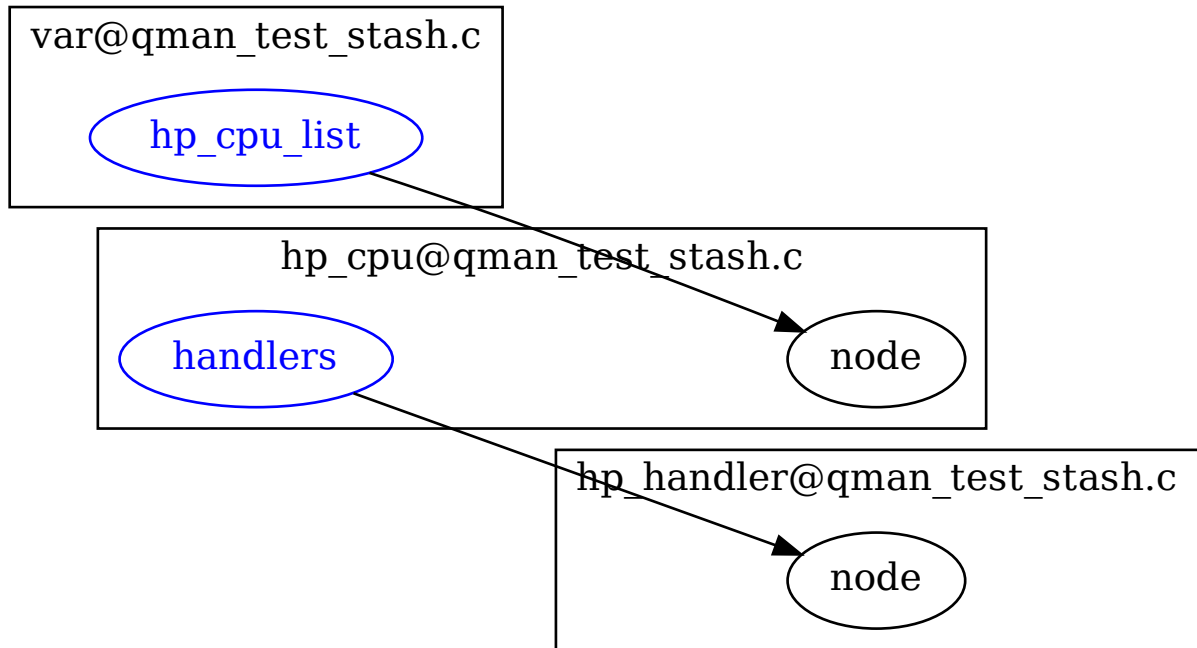


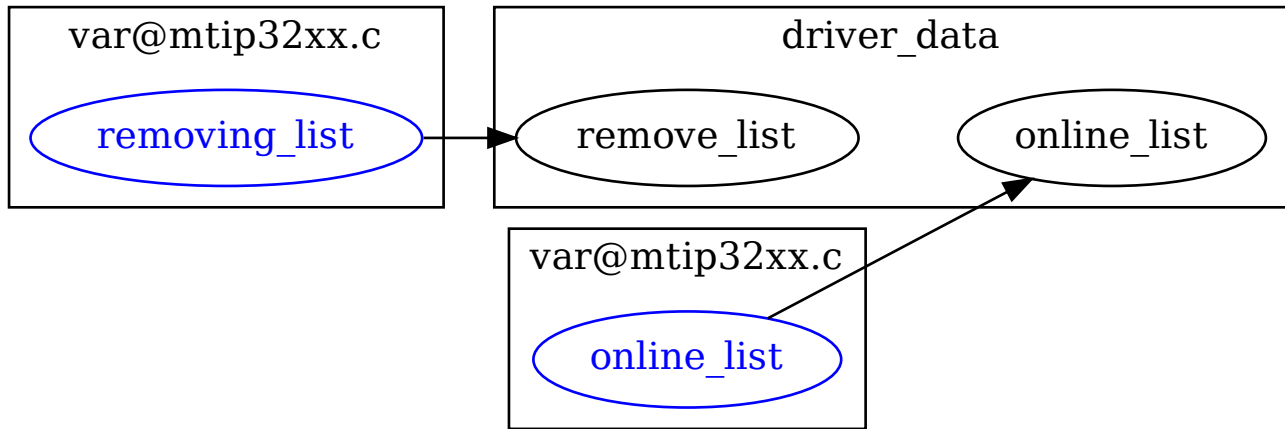


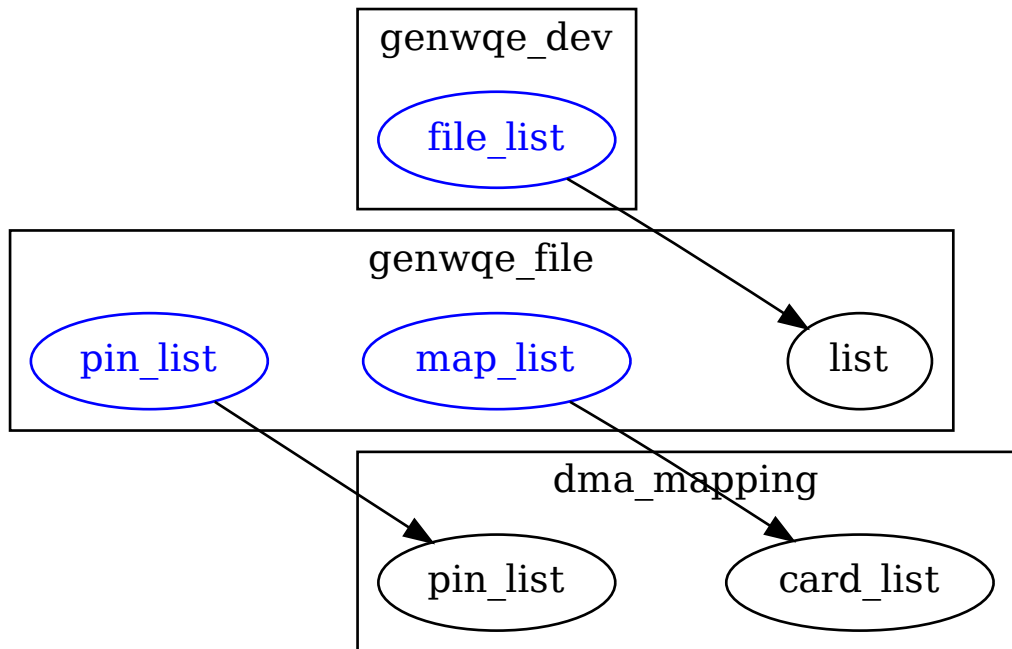












xchk_refcnt_check@refcount.c

fragments

var@xchk_refcountbt_process_rmap_fragments

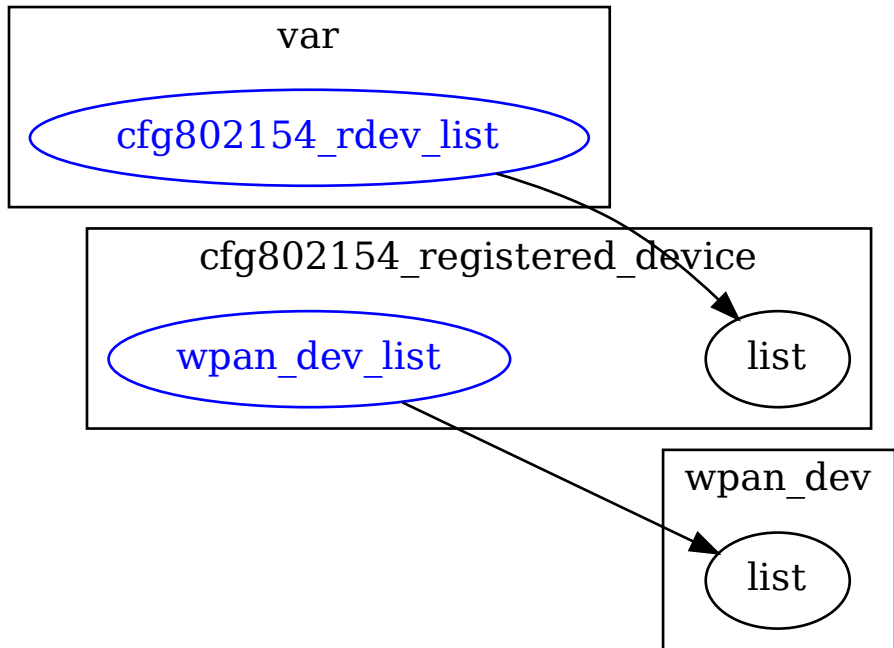
worklist

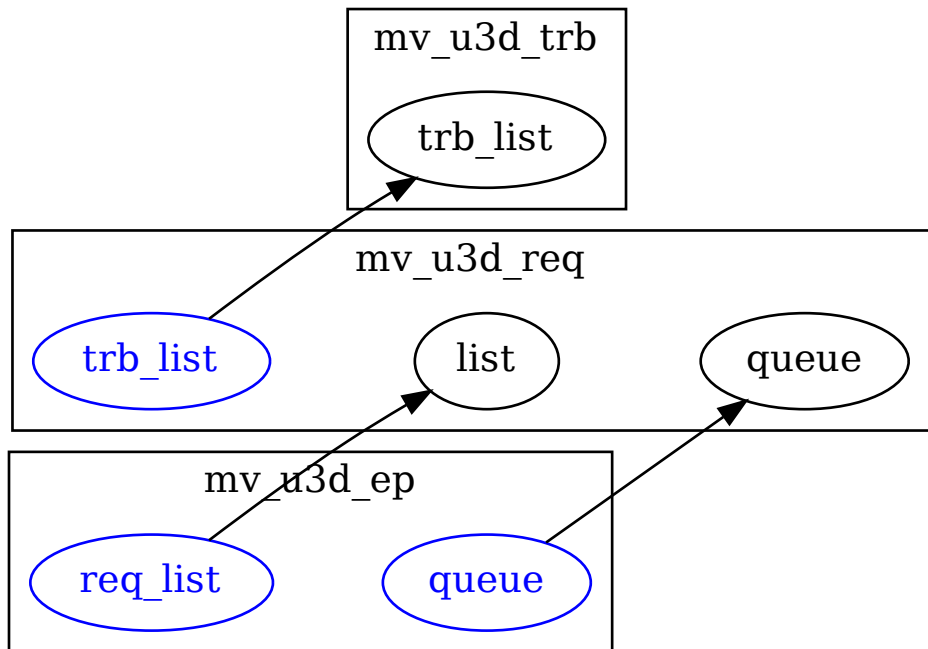
xchk_refcnt_frag@refcount.c

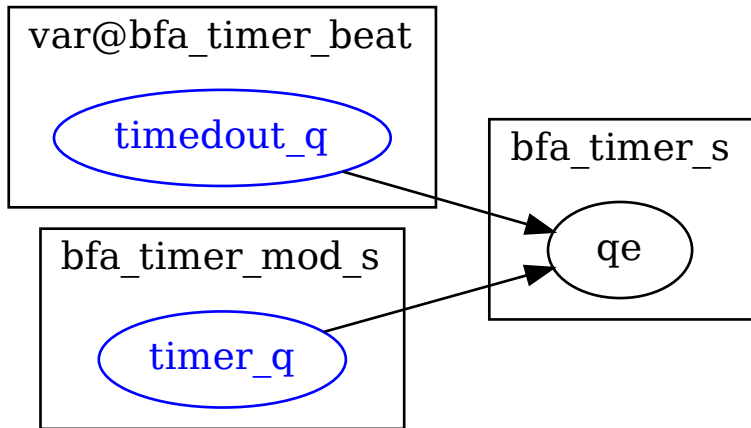
list

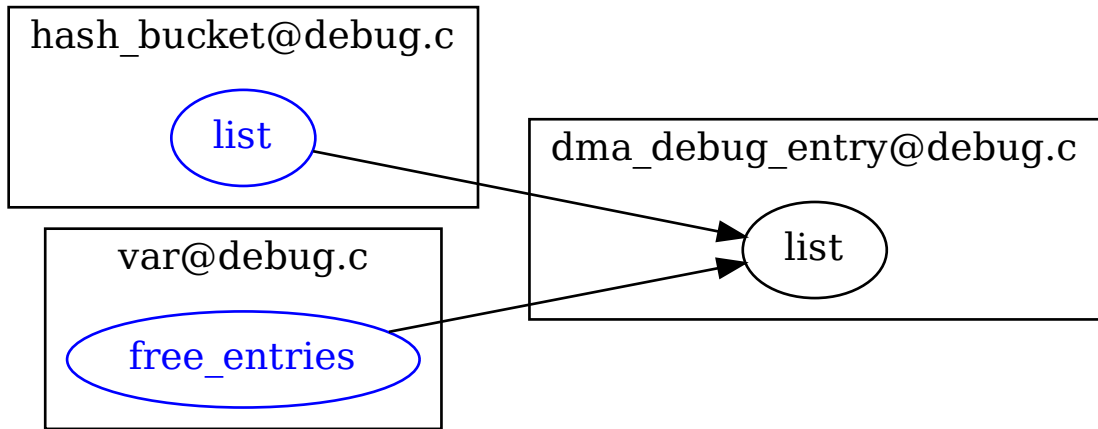
```
graph LR; subgraph Box1 [xchk_refcnt_check@refcount.c]; fragments; end; subgraph Box2 [var@xchk_refcountbt_process_rmap_fragments]; worklist; end; subgraph Box3 [xchk_refcnt_frag@refcount.c]; list; end; fragments --> list; worklist --> list;
```

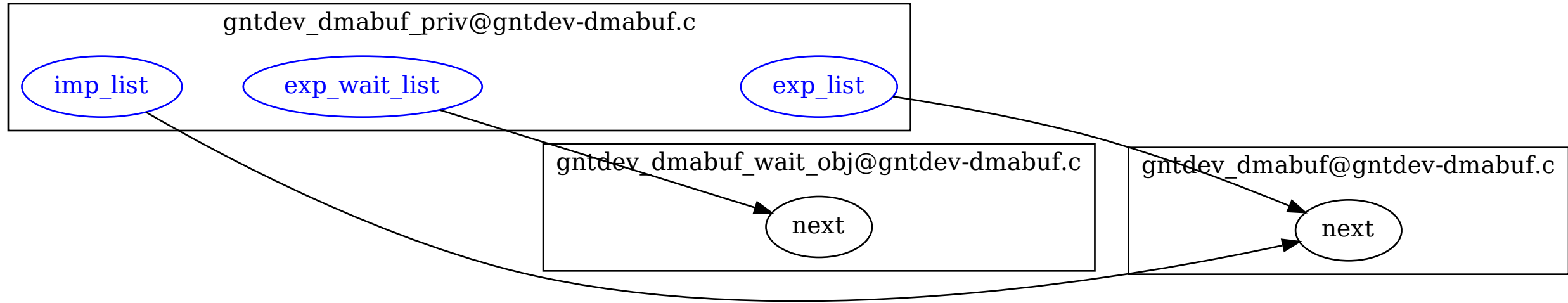
The diagram illustrates pointer relationships between variables in three source files. It consists of three rectangular boxes. The top-left box, labeled 'xchk_refcnt_check@refcount.c', contains a blue oval labeled 'fragments'. The bottom-left box, labeled 'var@xchk_refcountbt_process_rmap_fragments', contains a blue oval labeled 'worklist'. The right box, labeled 'xchk_refcnt_frag@refcount.c', contains an oval labeled 'list'. Two arrows point from the 'fragments' and 'worklist' ovals to the 'list' oval, indicating that both 'fragments' and 'worklist' point to the same memory location 'list'.

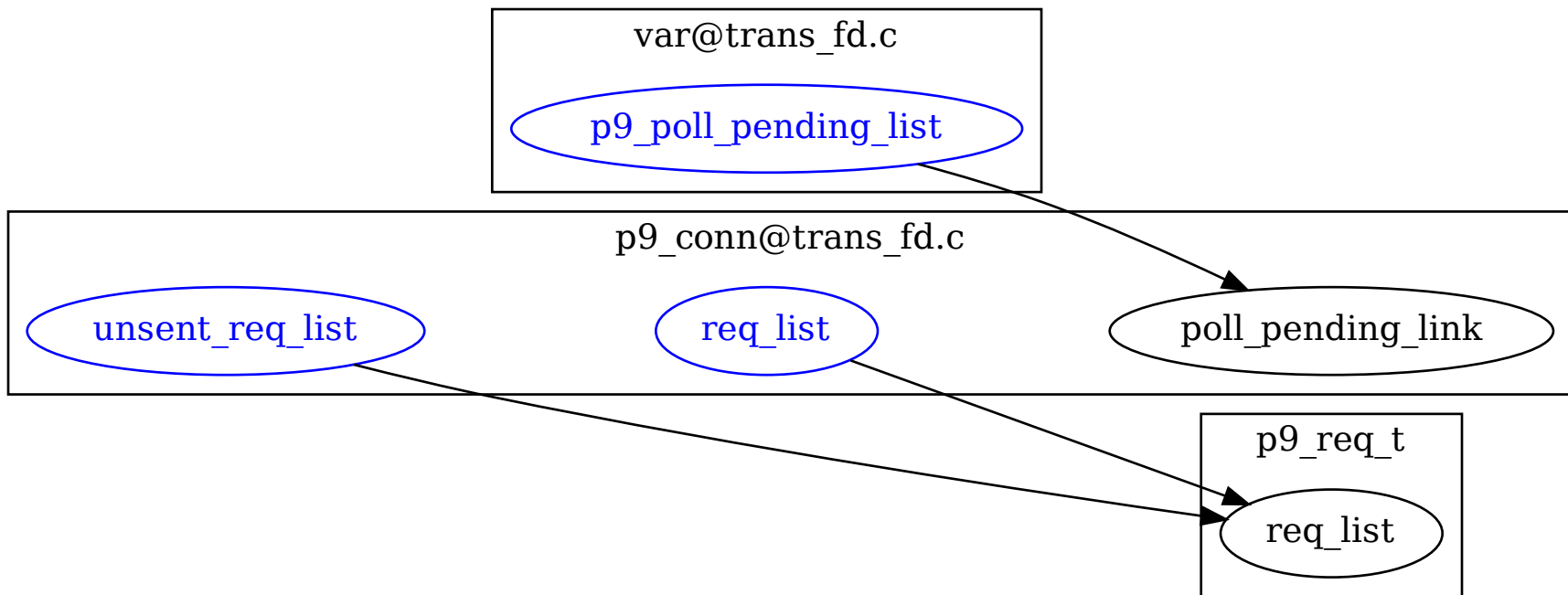


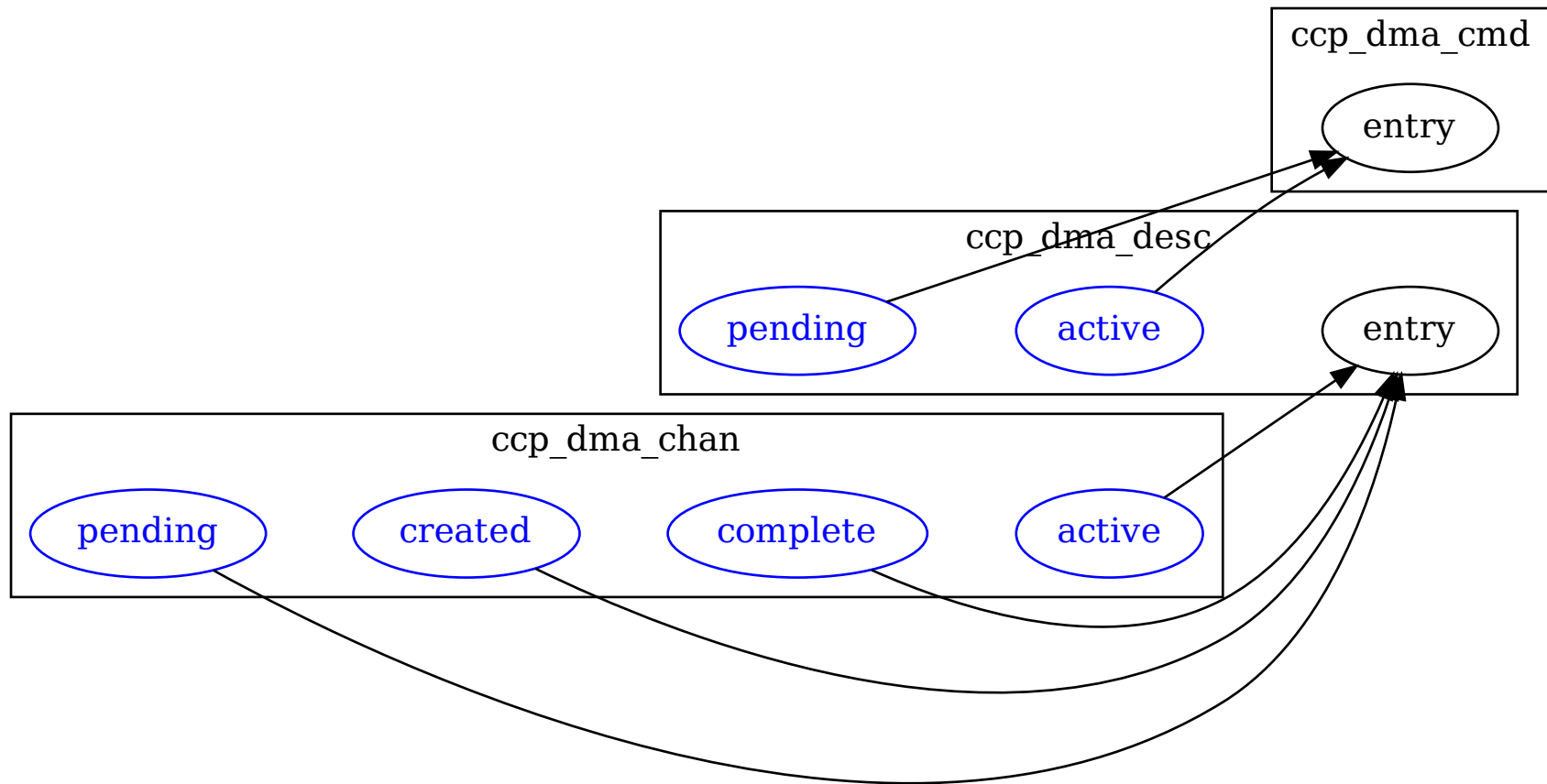


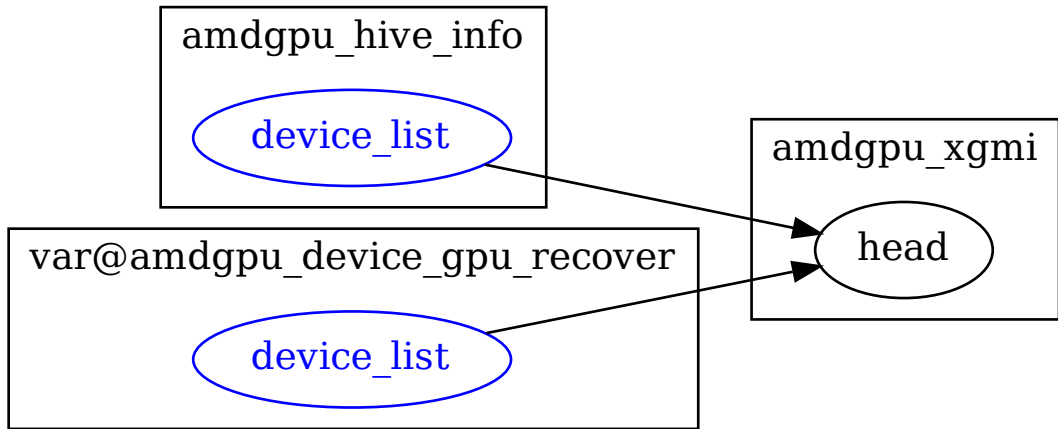


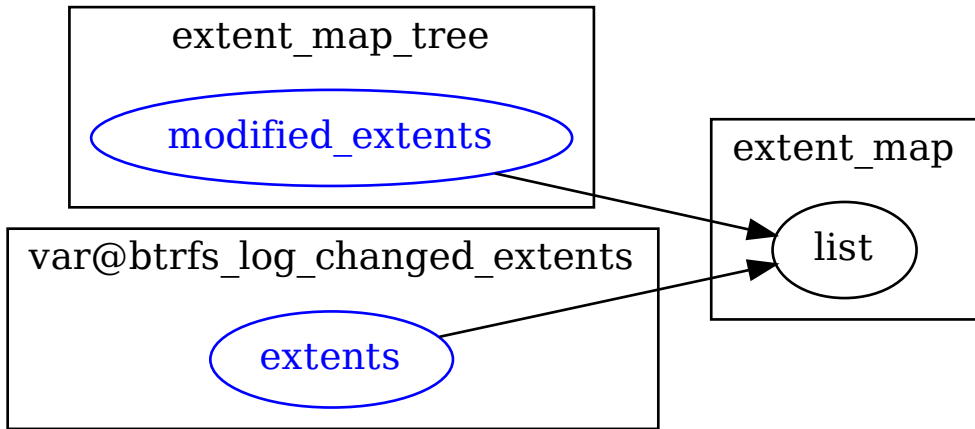


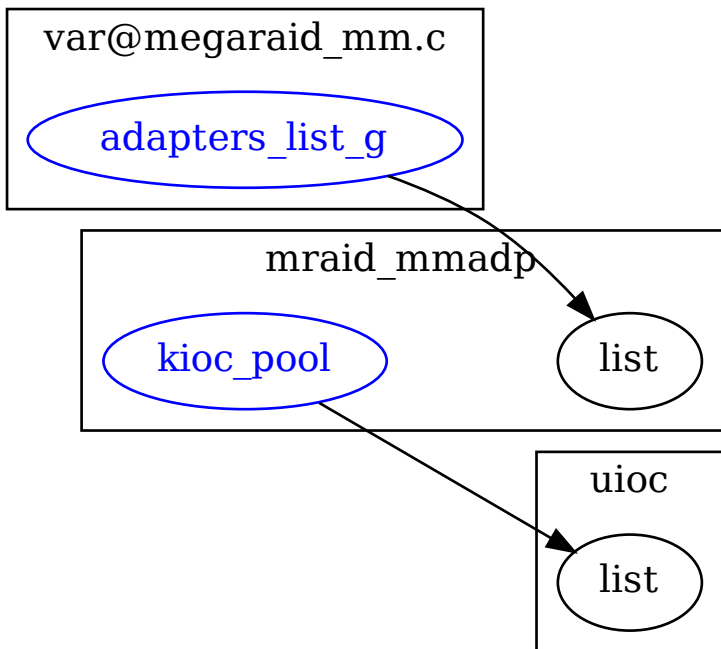


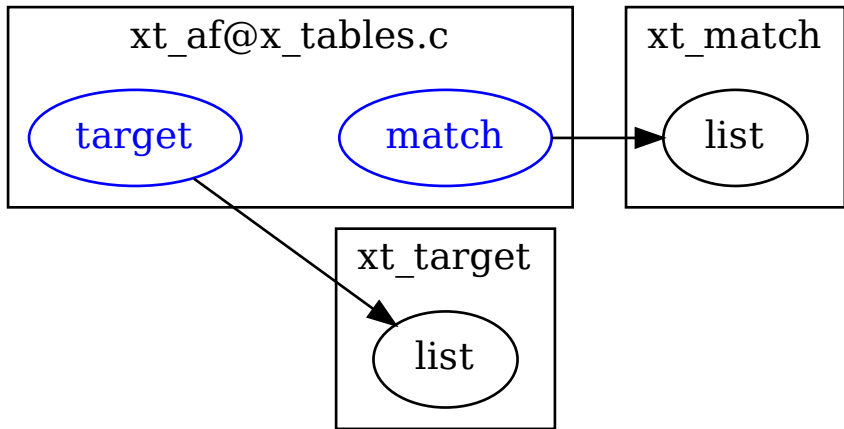


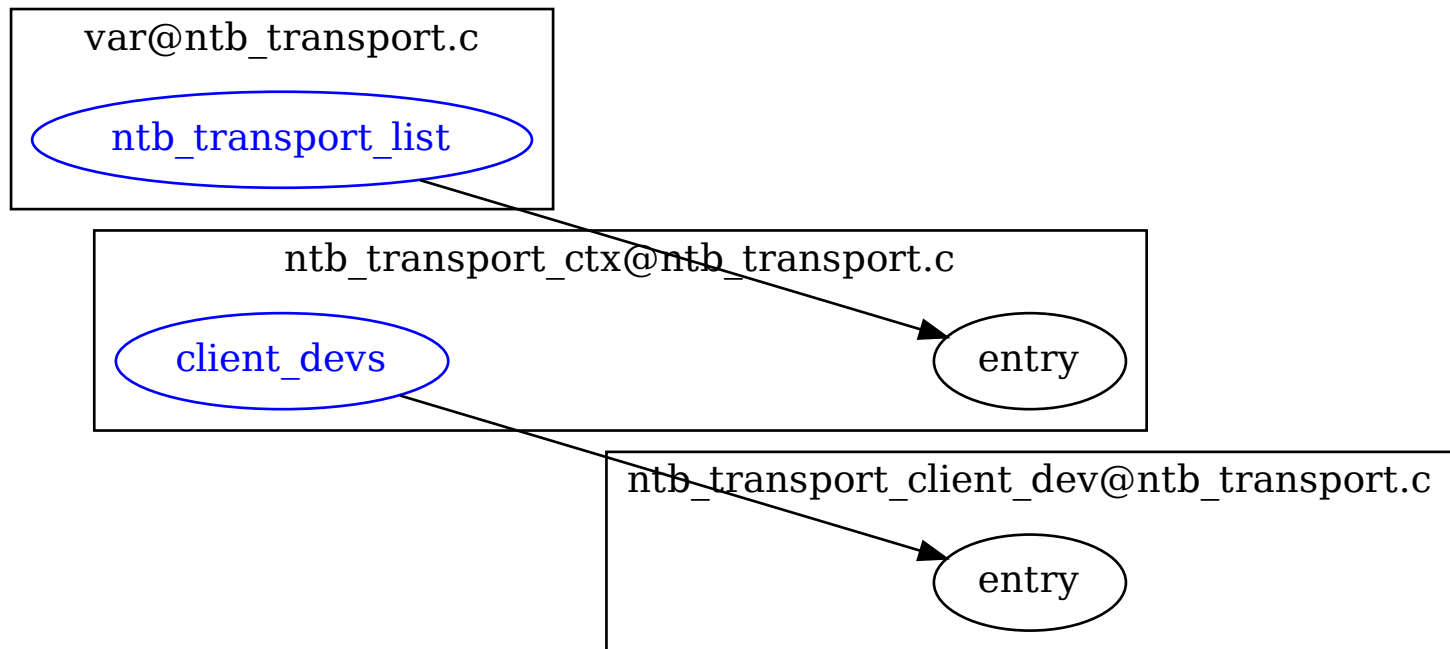


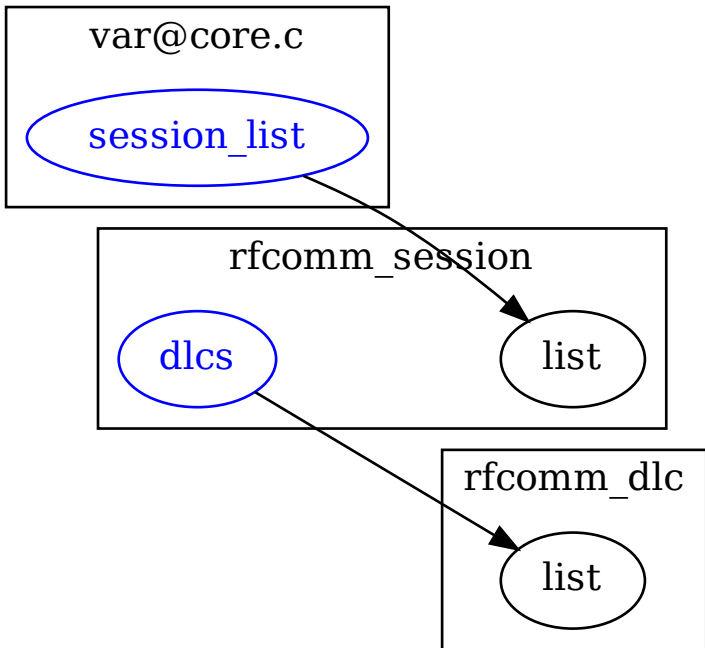


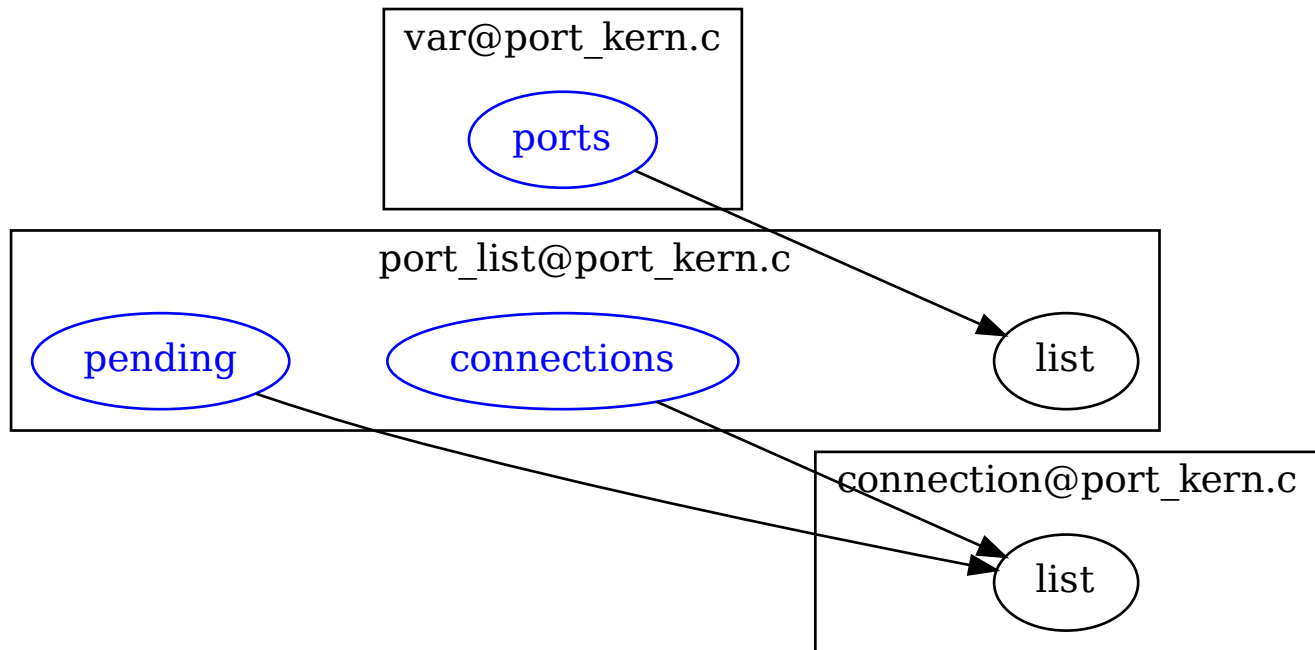


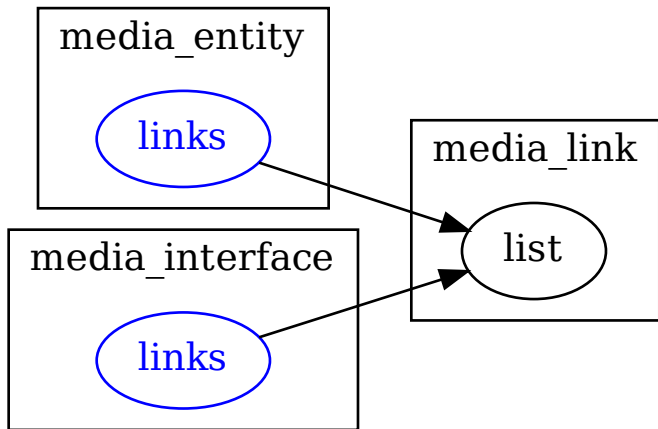


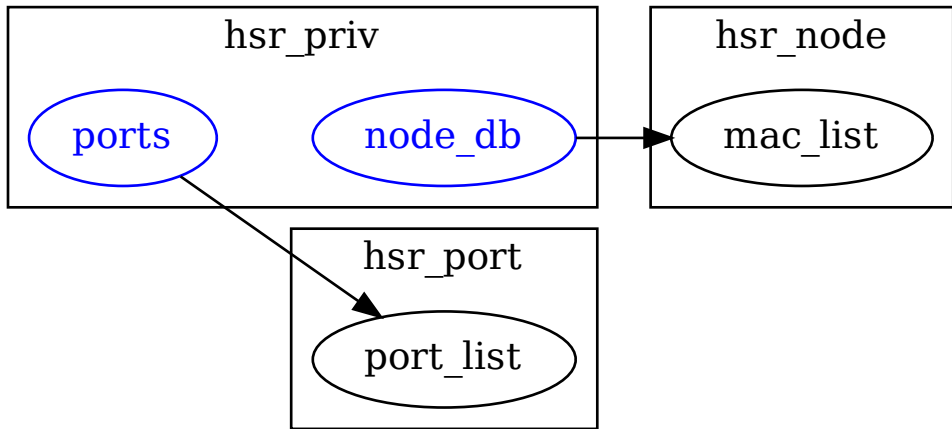


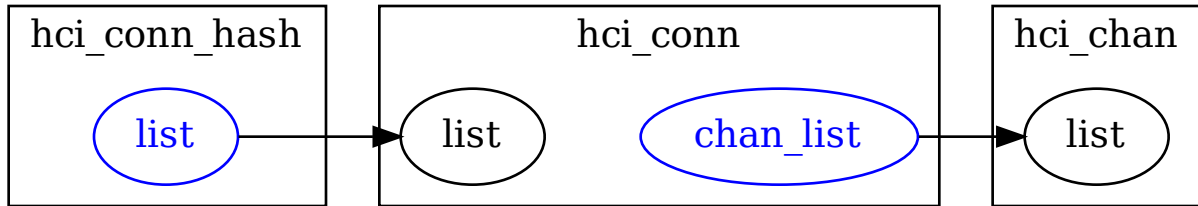


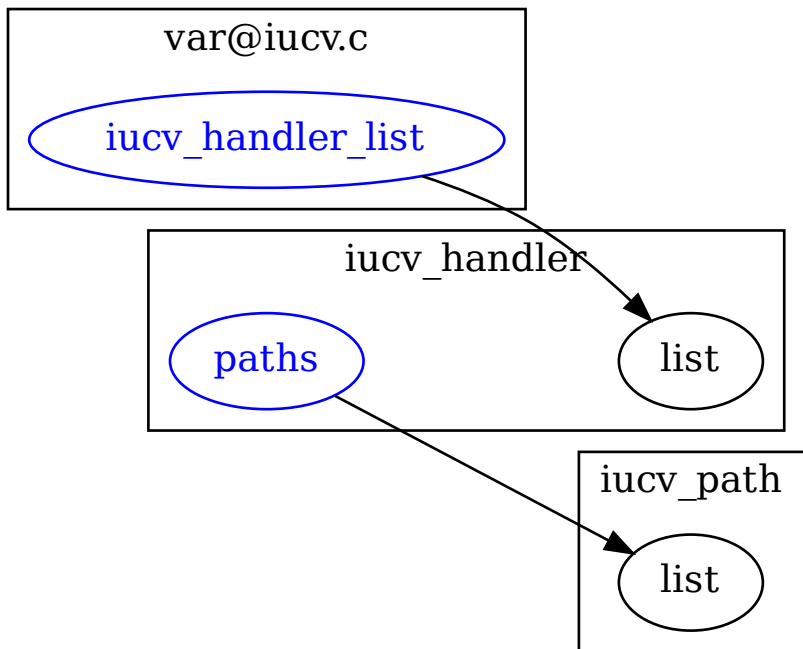


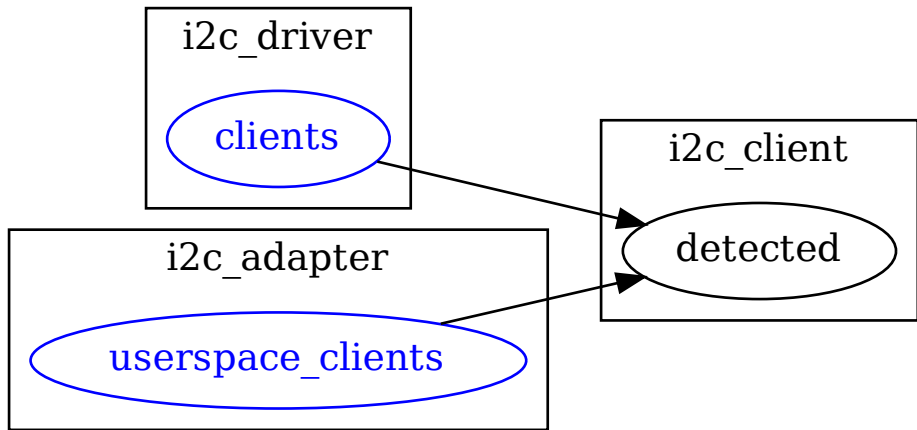


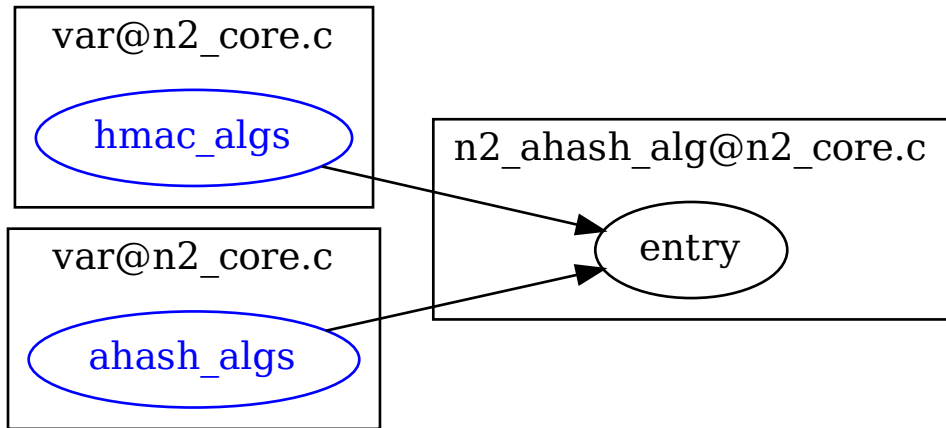


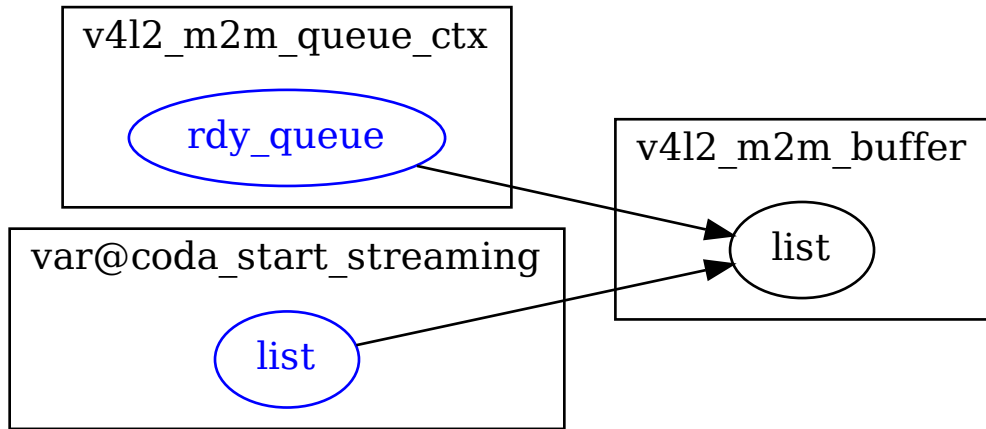


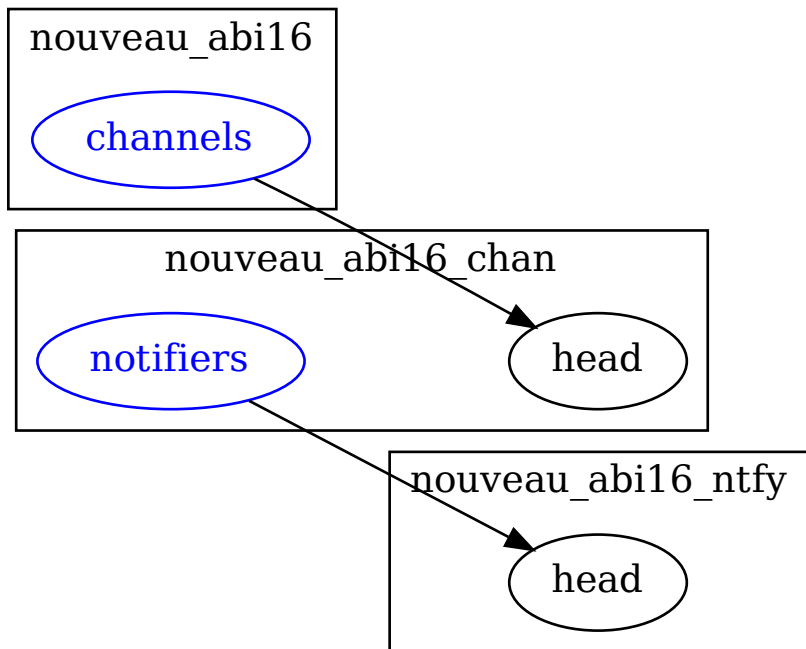


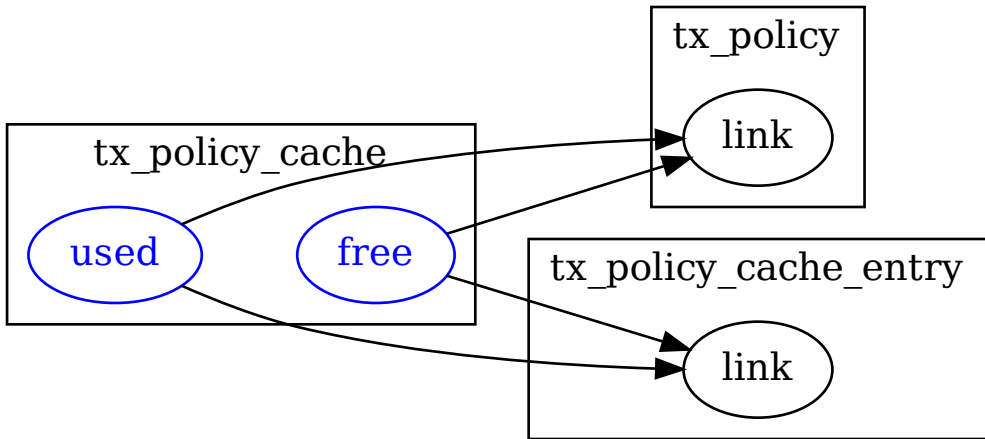


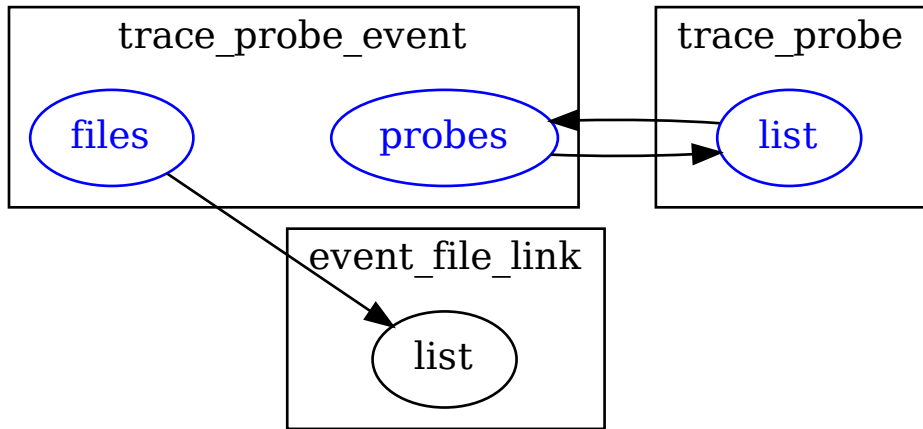


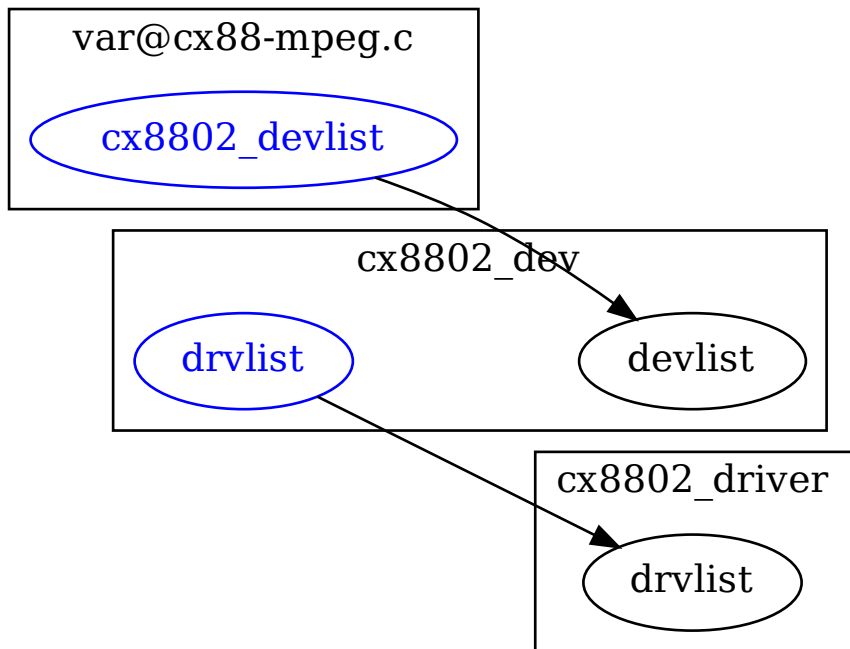


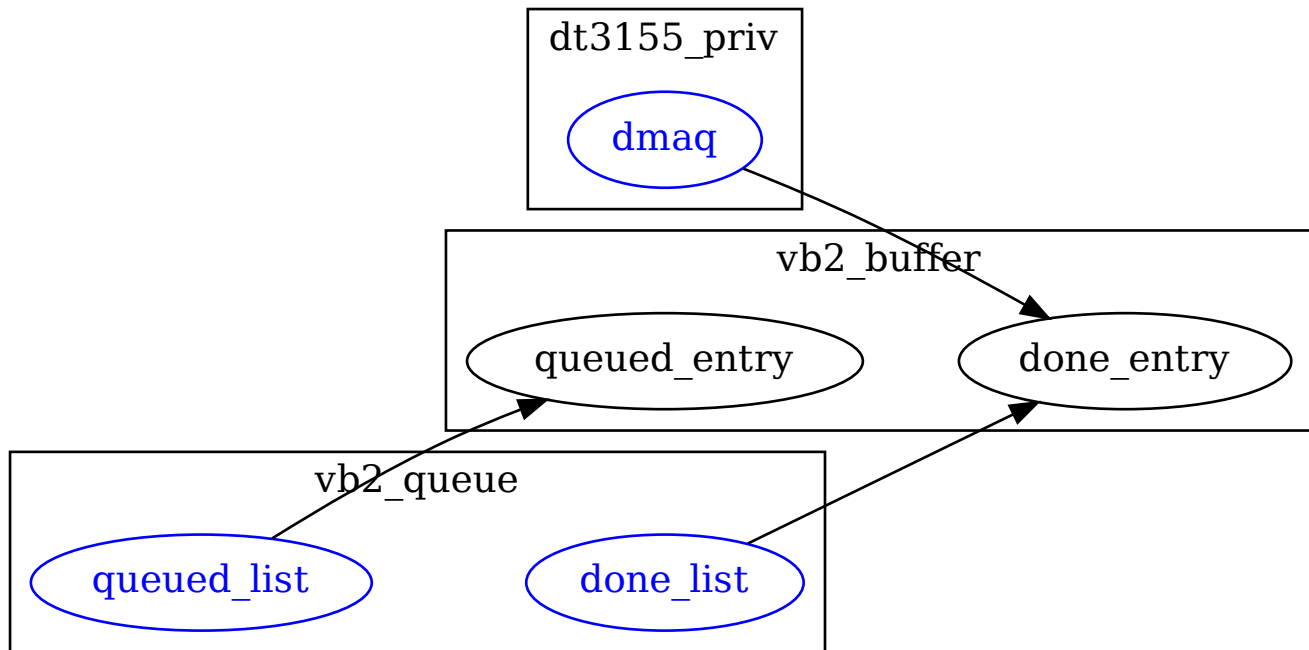












var@irq-gic-v3-its.c

its_nodes

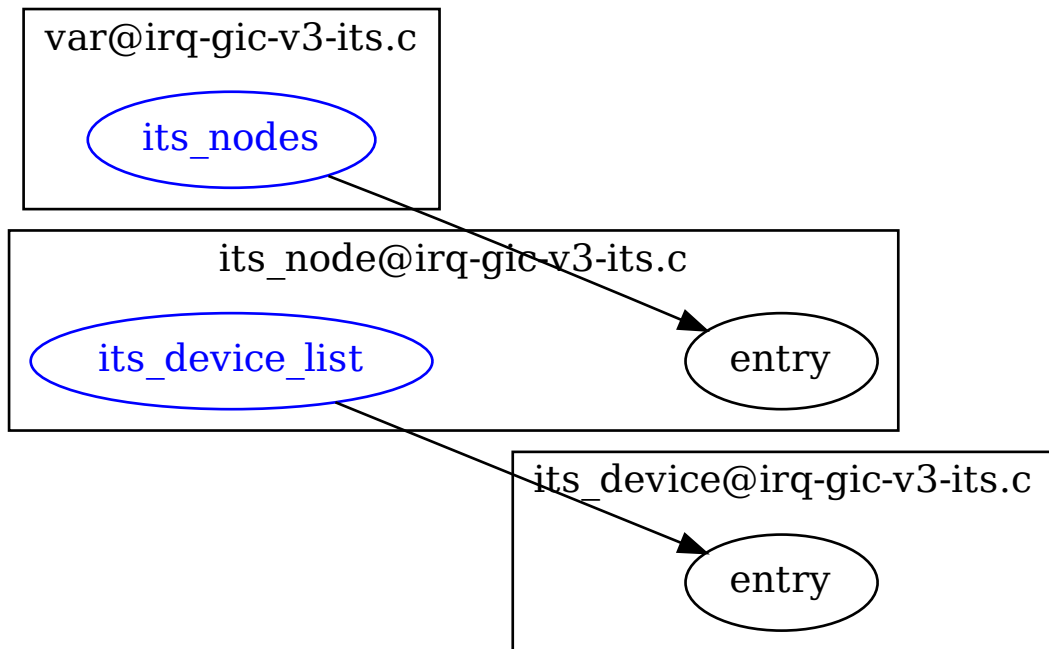
its_node@irq-gic-v3-its.c

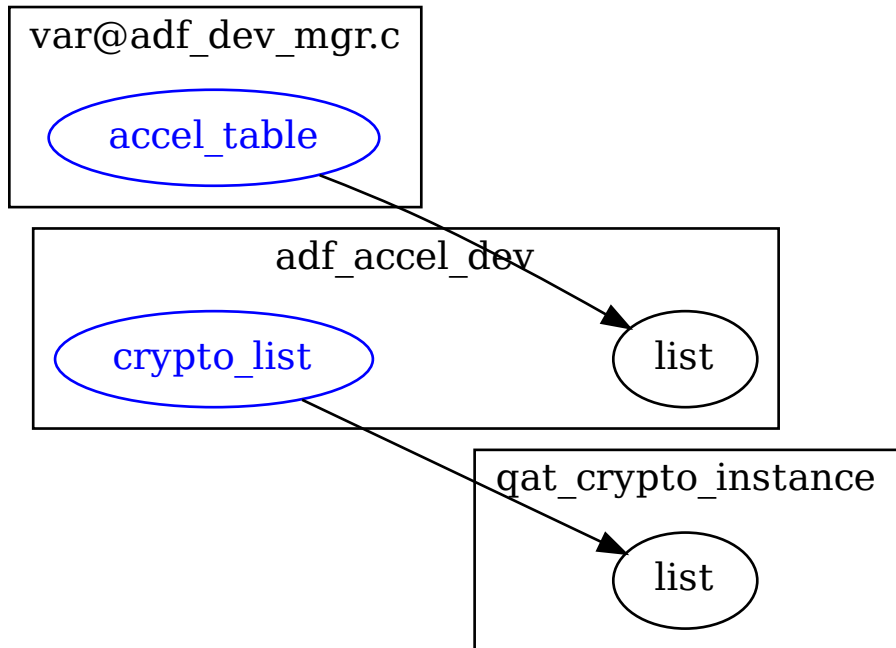
its_device_list

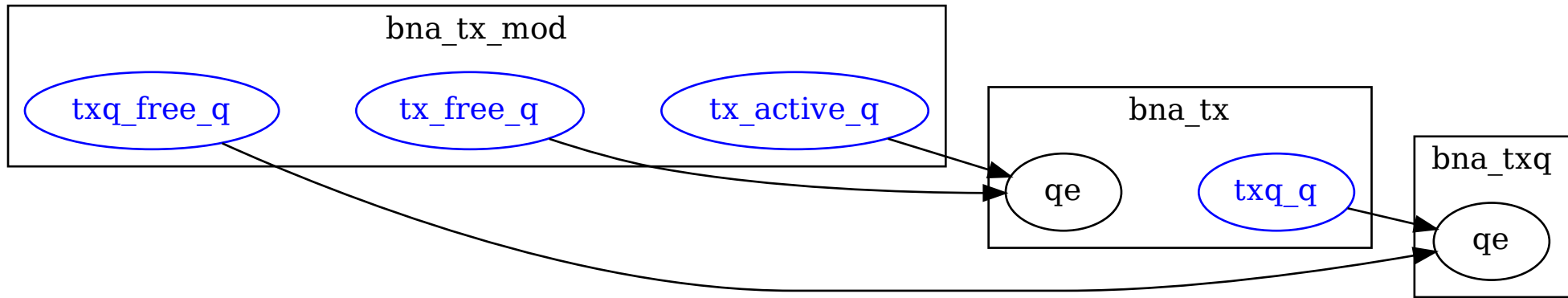
entry

its_device@irq-gic-v3-its.c

entry







var@target_core_alua.c

lu_gps_list

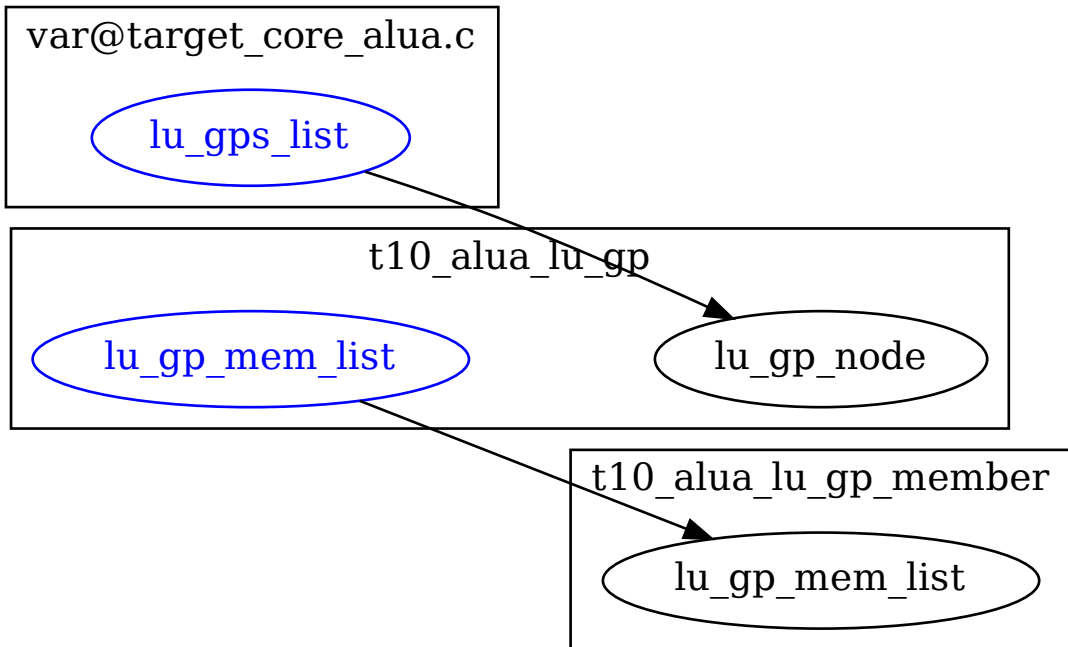
t10_alua_lu_gp

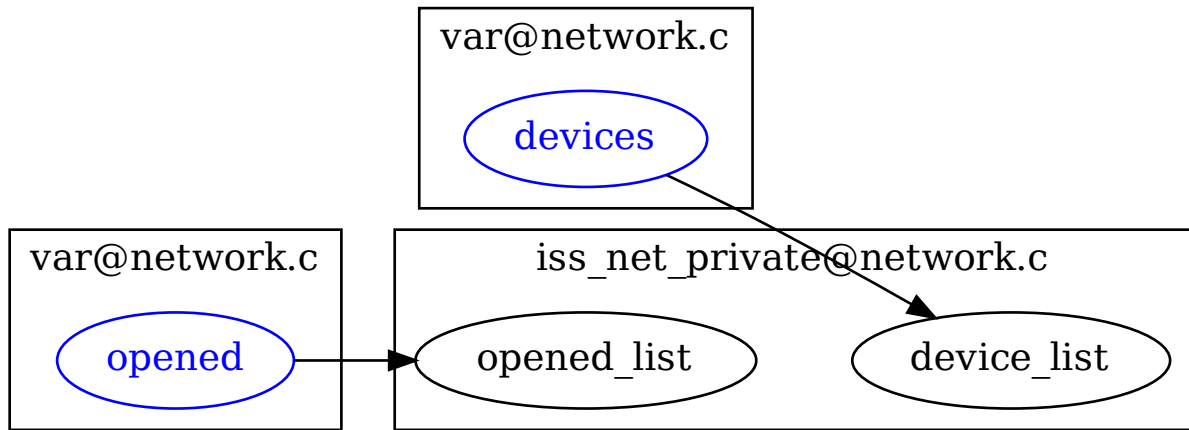
lu_gp_mem_list

lu_gp_node

t10_alua_lu_gp_member

lu_gp_mem_list





var@qla4xxx_ddb_login_st

list_nt

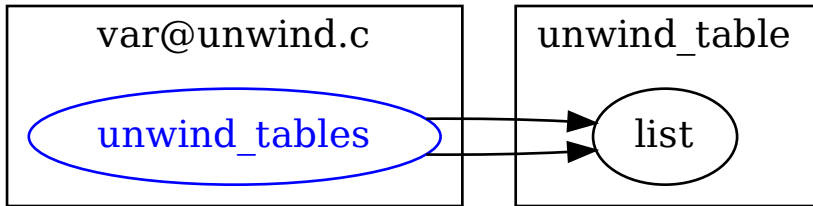
var@qla4xxx_build_ddb_list

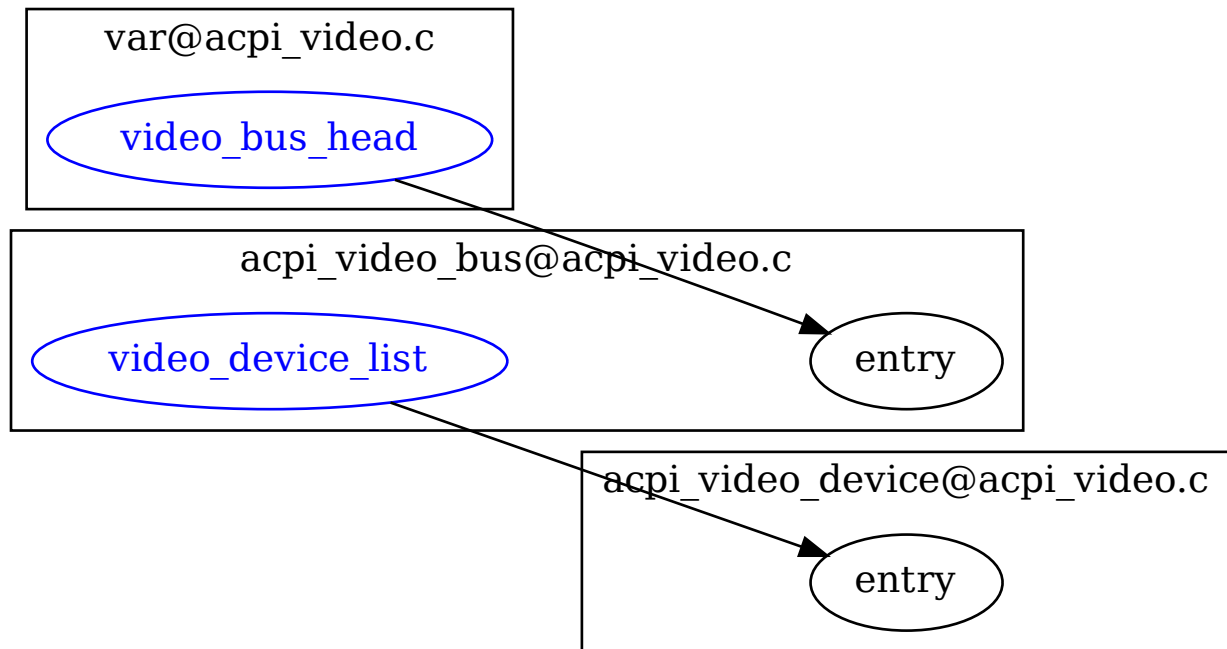
list_st

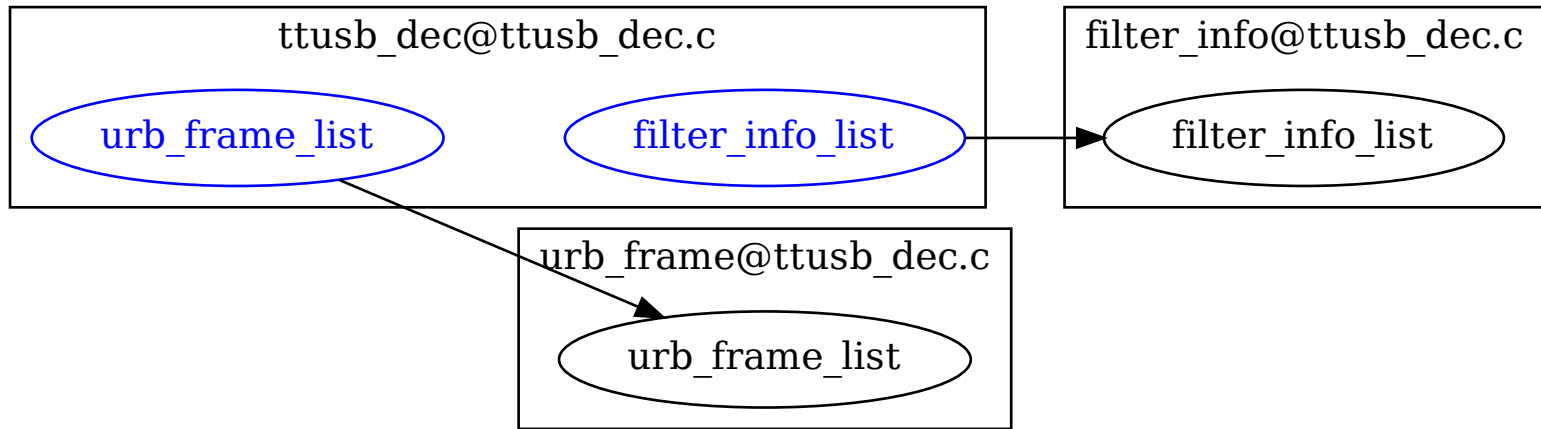
qla_ddb_index

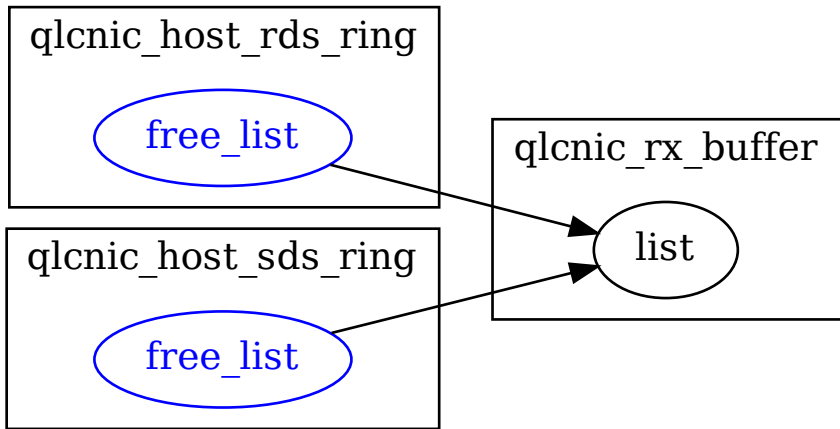
list

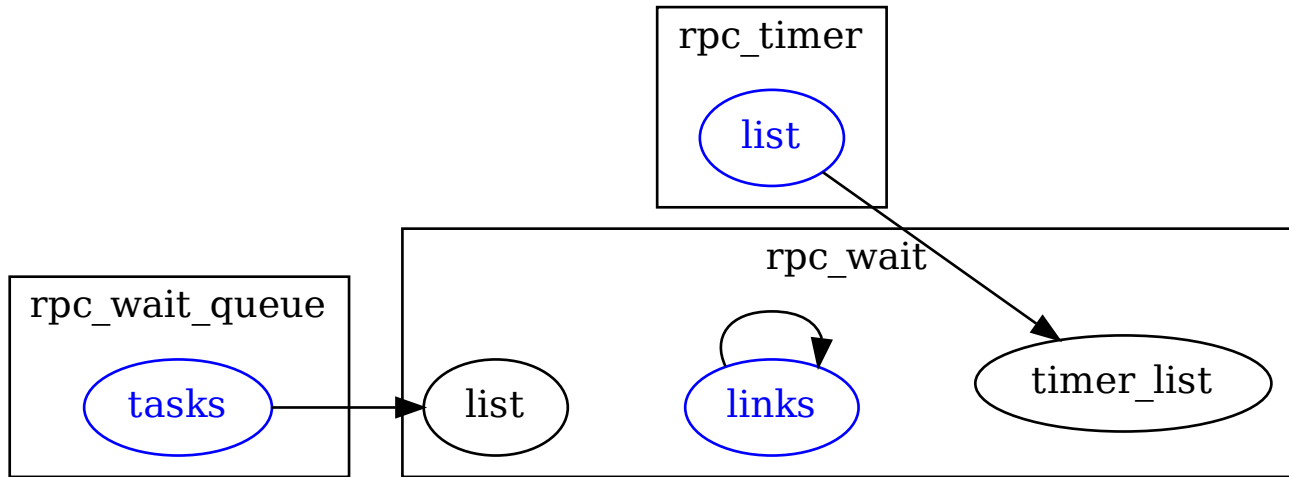
```
graph LR; subgraph Box1 [ ]; list_nt; end; subgraph Box2 [ ]; list_st; end; subgraph Box3 [ ]; qla_ddb_index; list; end; list_nt --> list; list_st --> list;
```

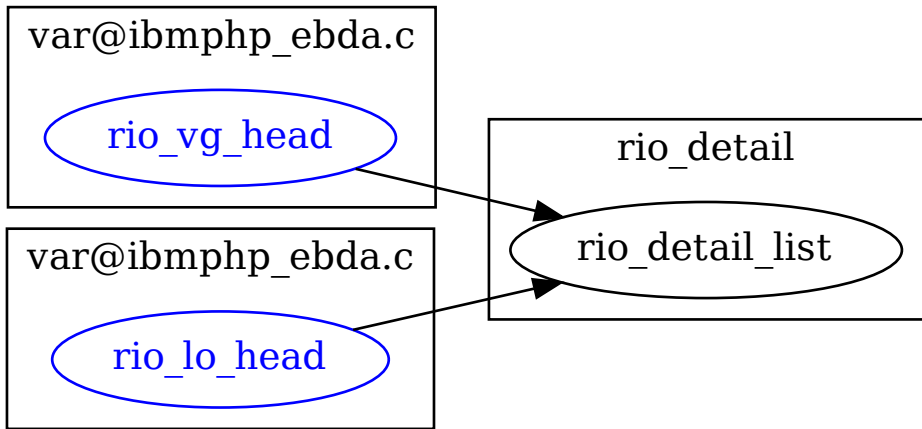


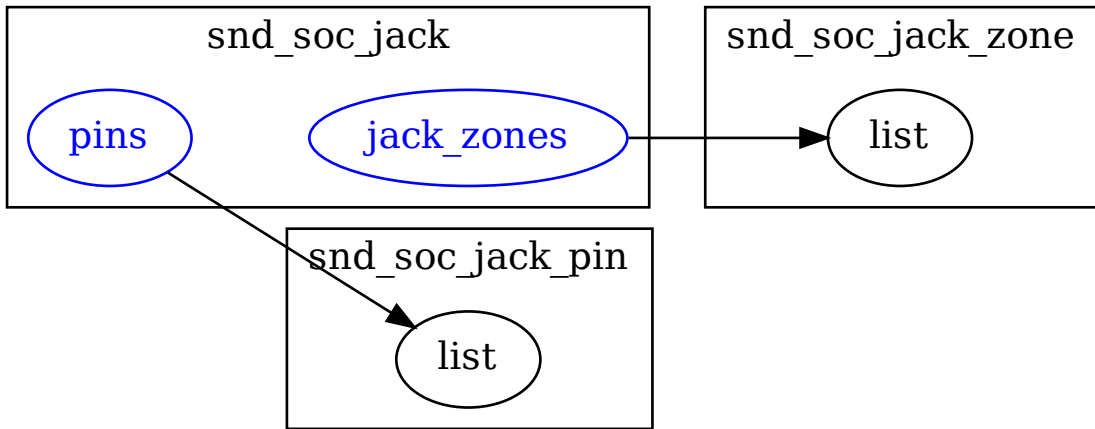


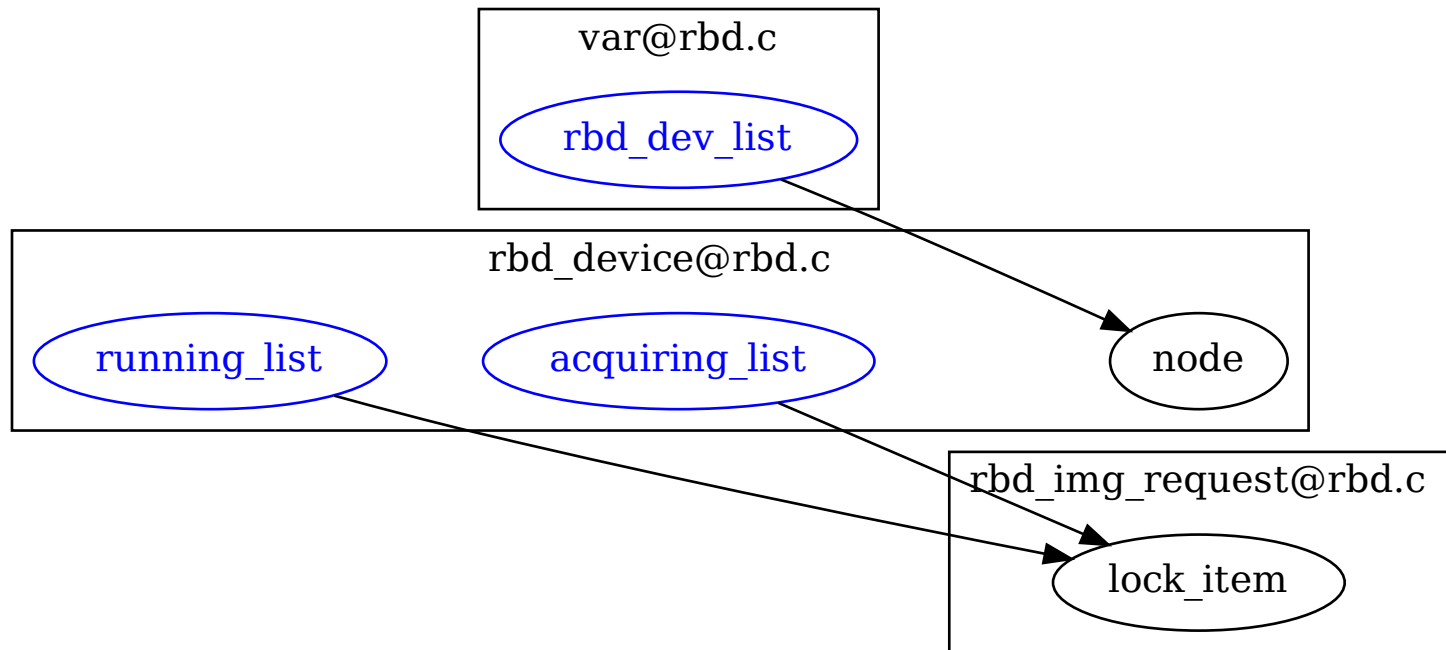


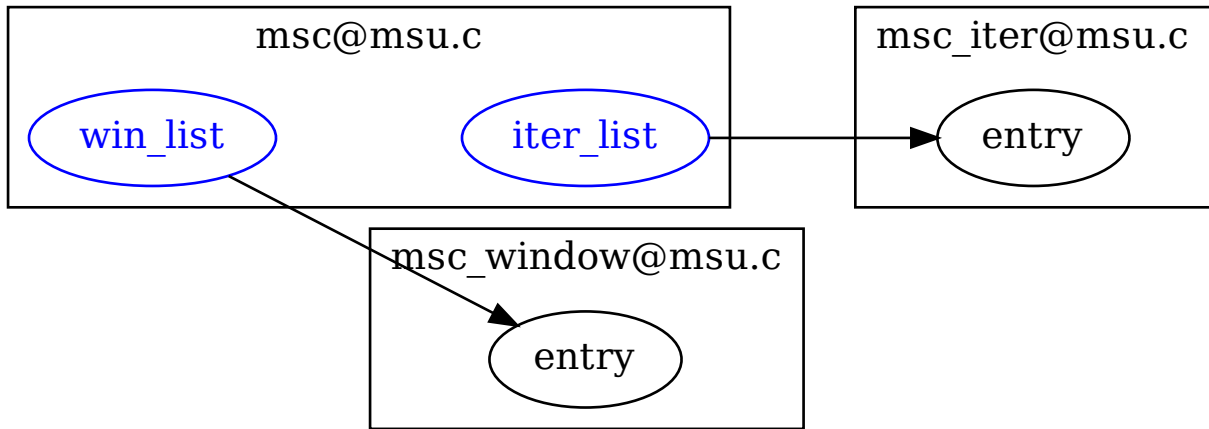


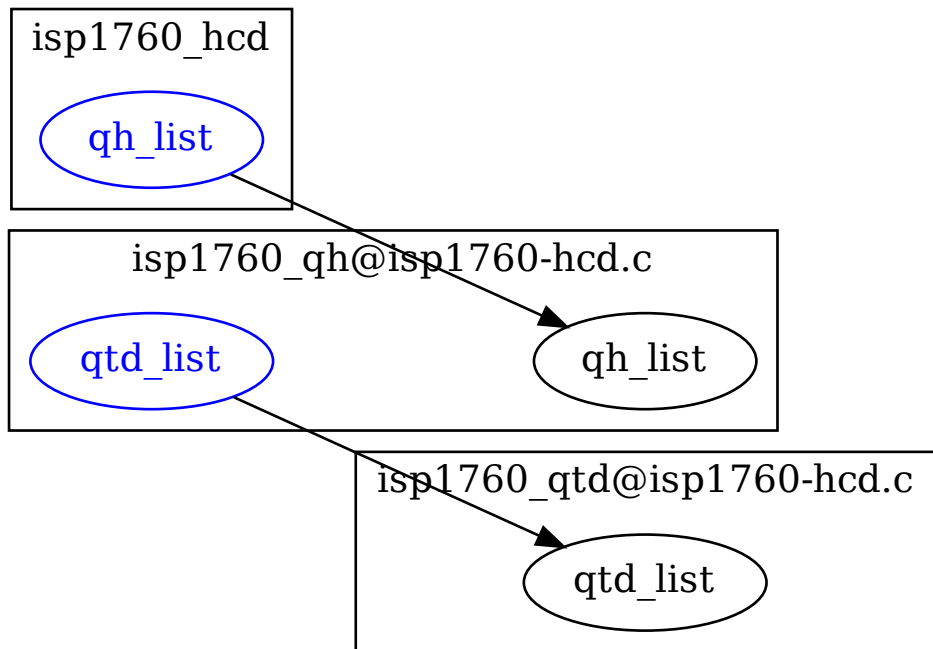


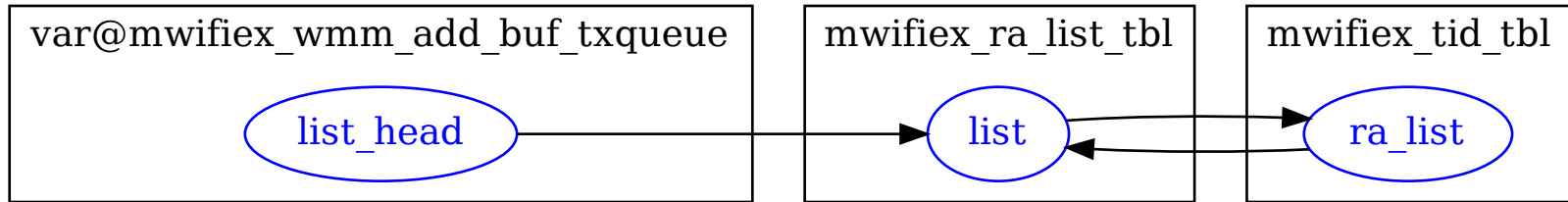


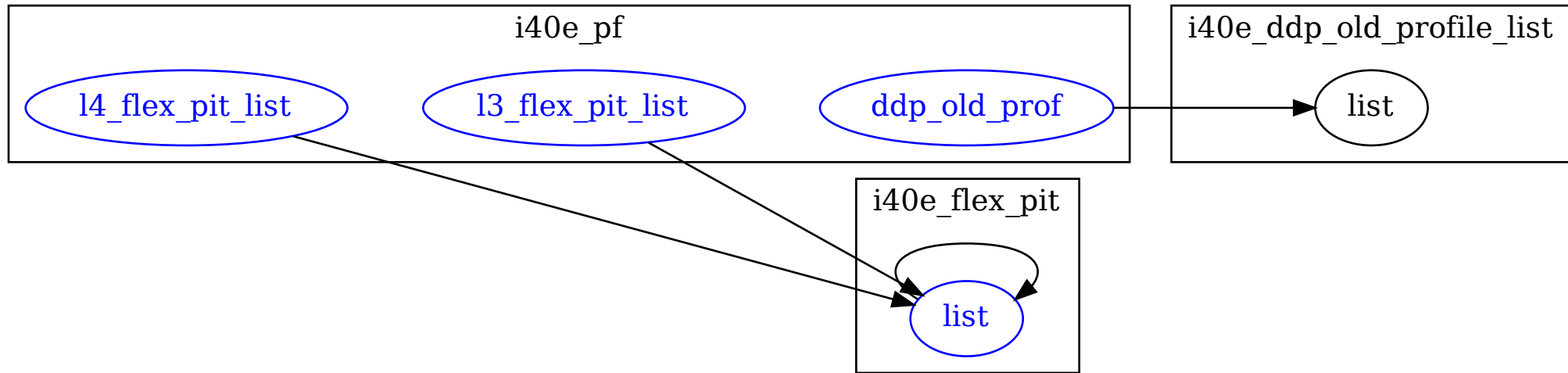


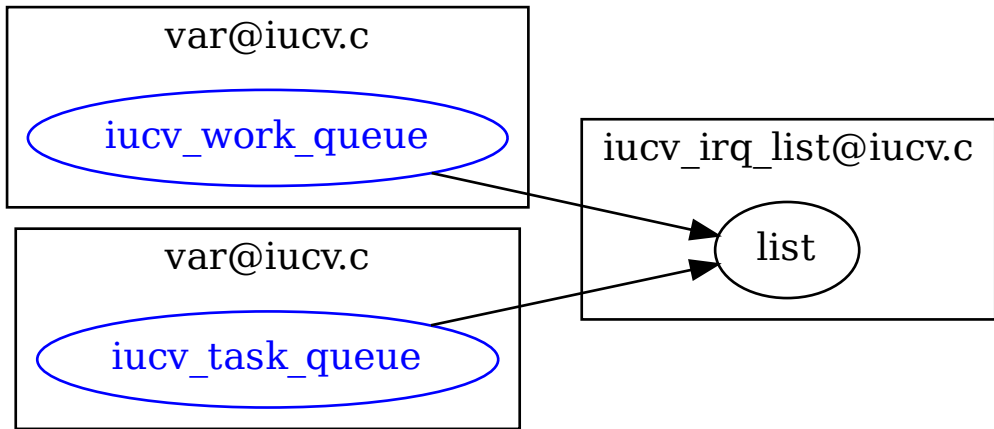


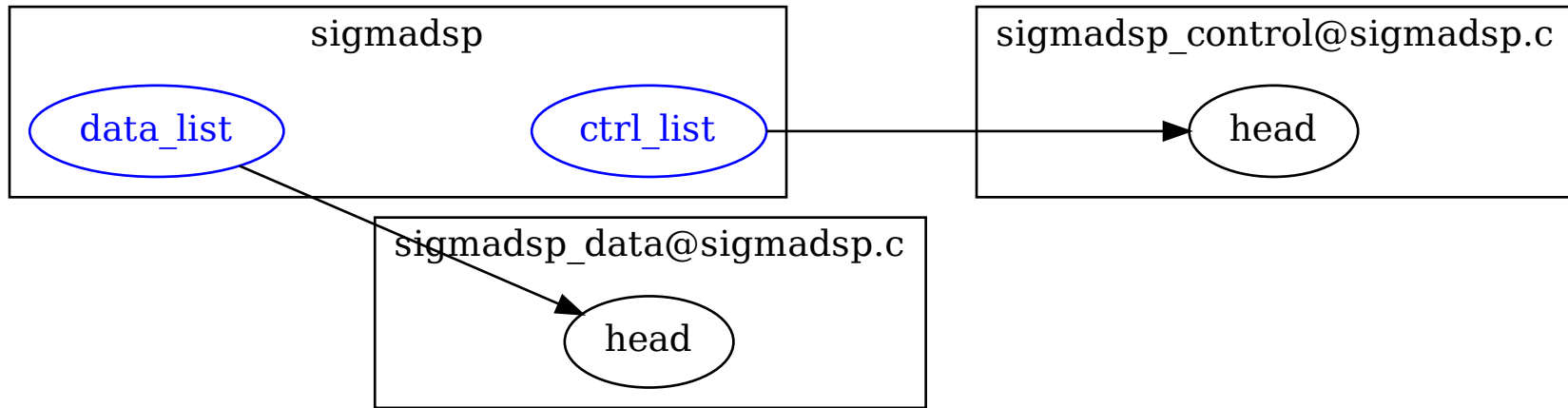


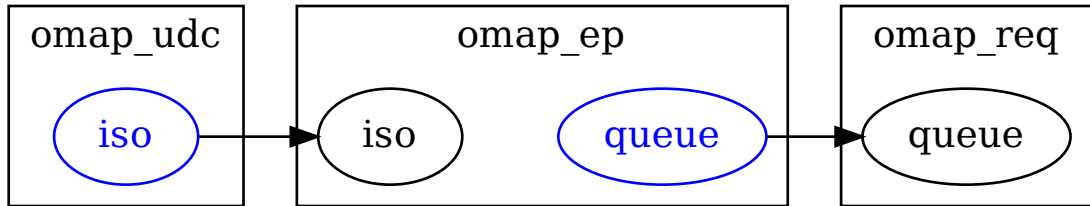


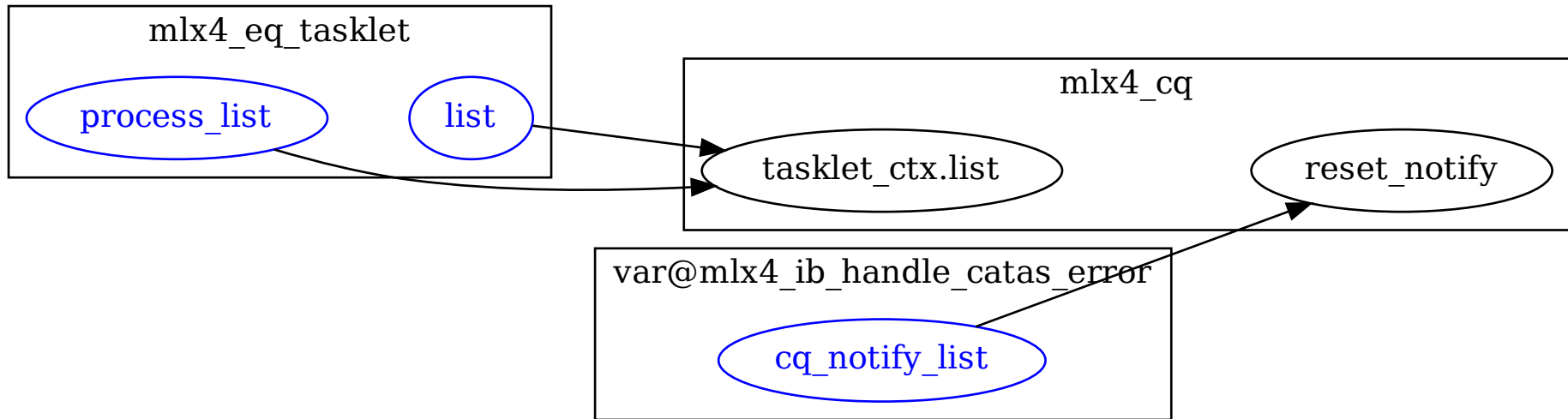


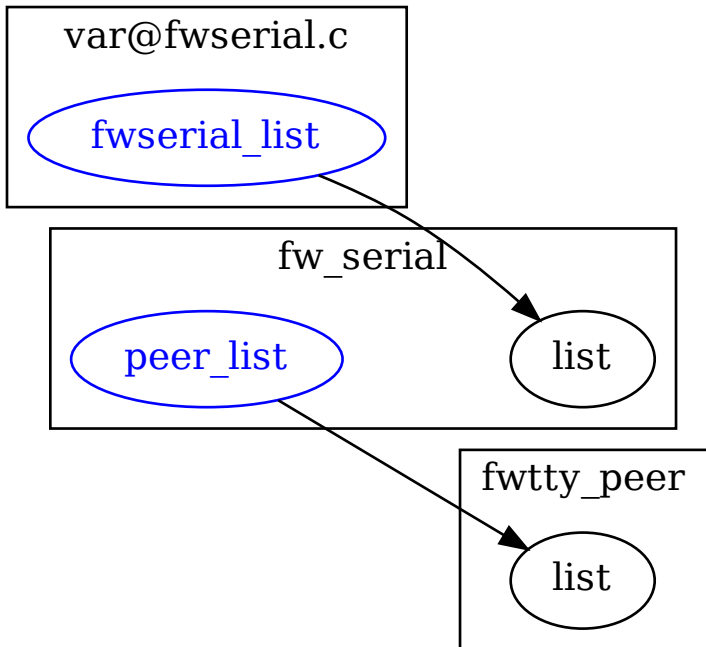


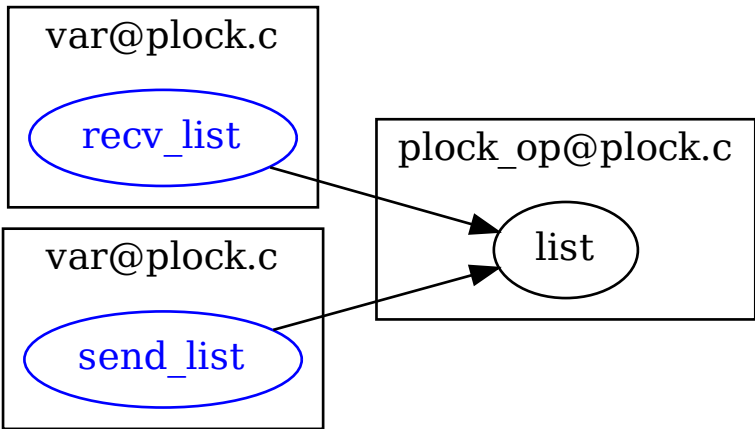


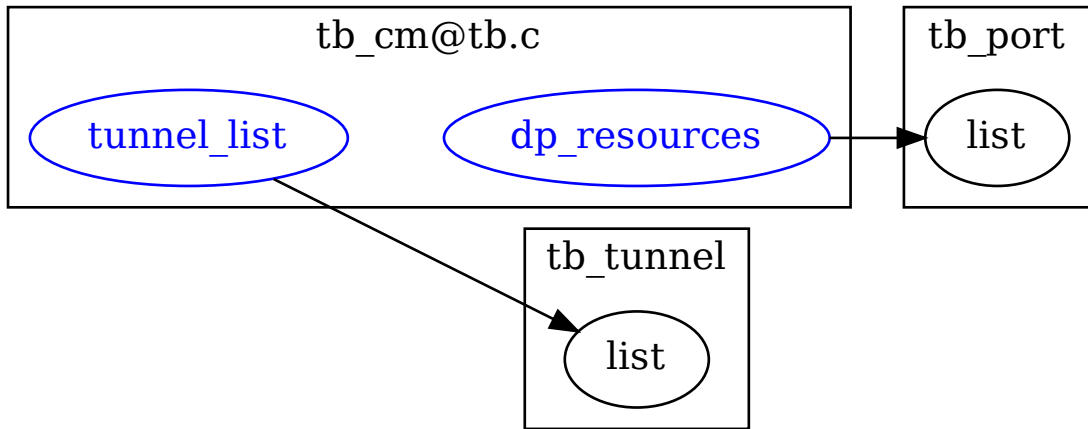


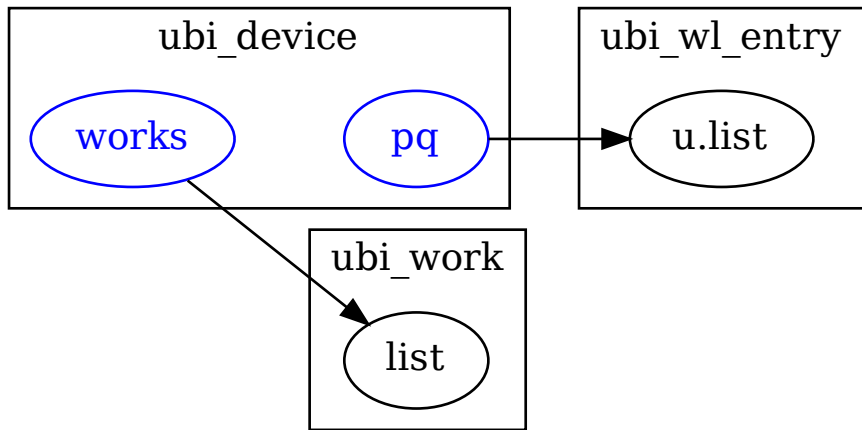


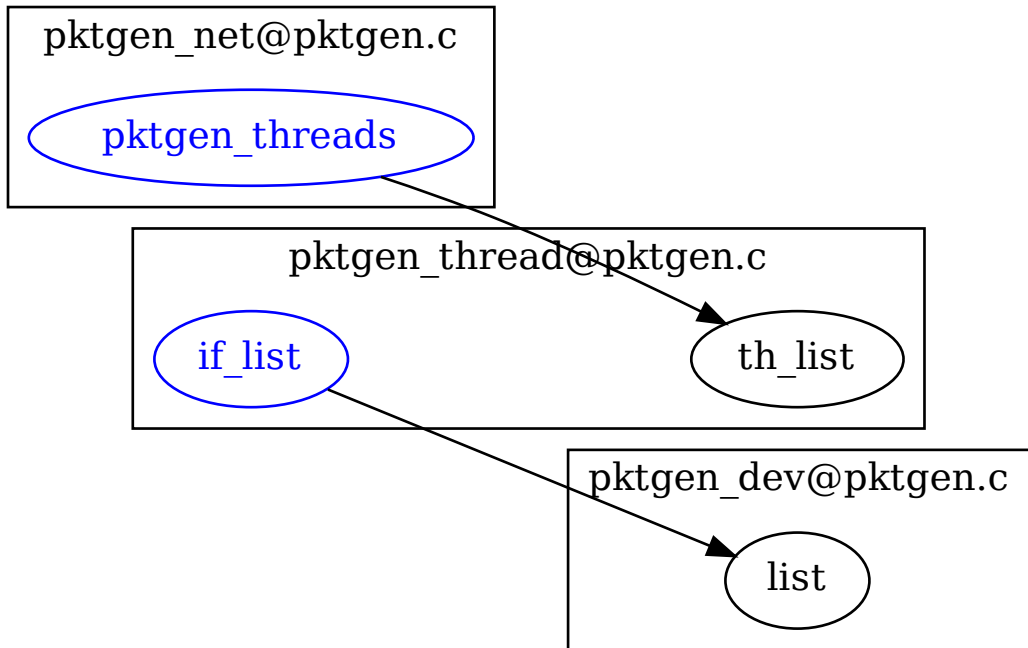


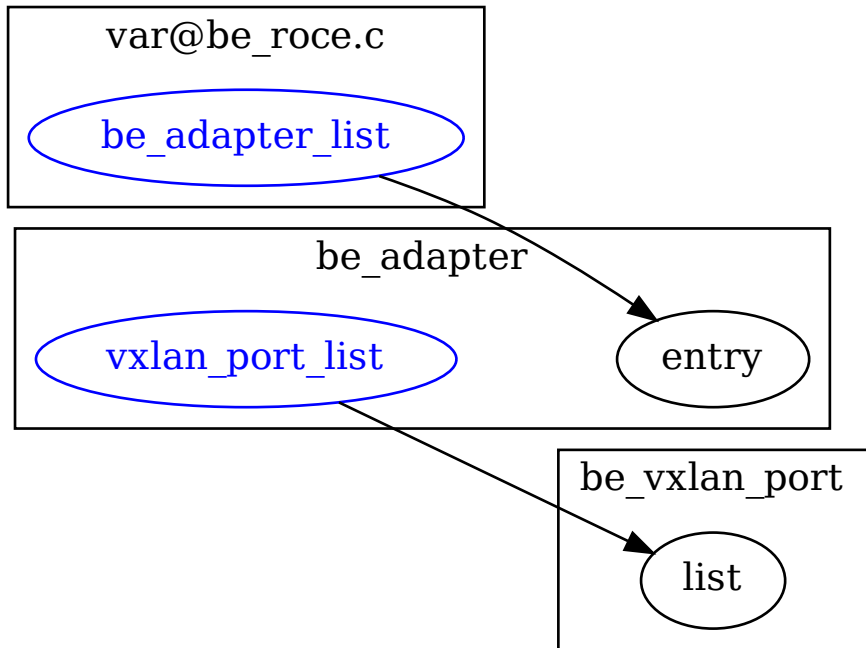


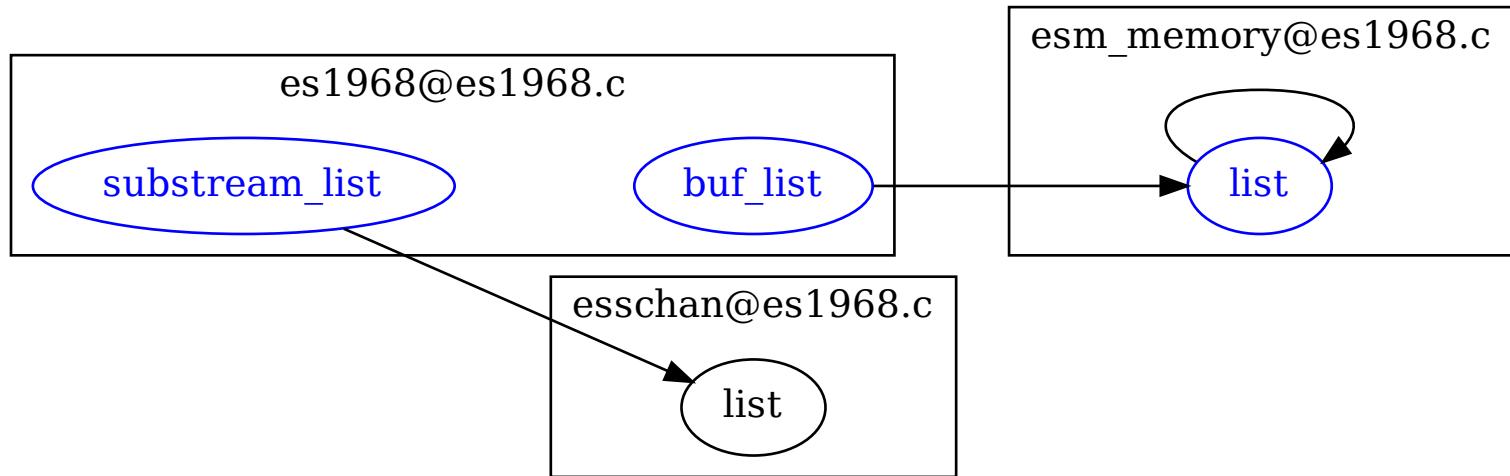


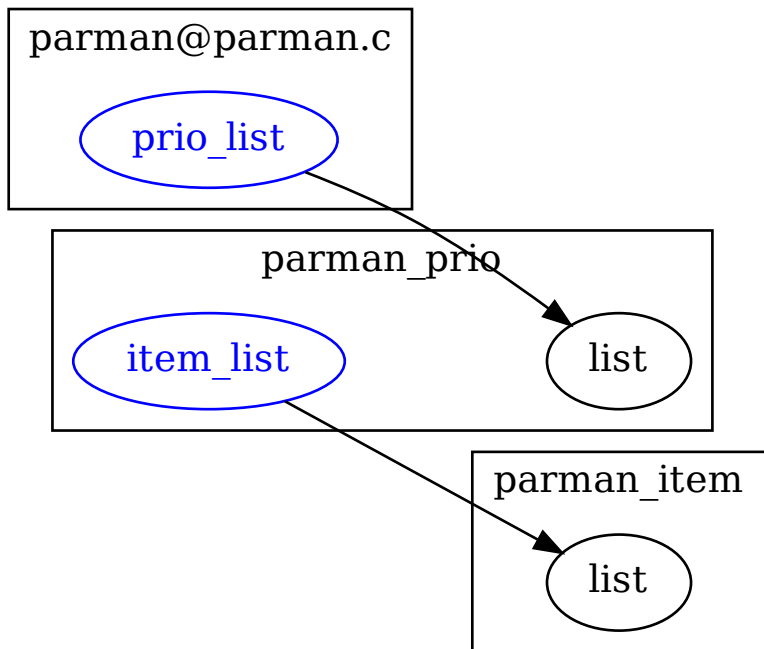


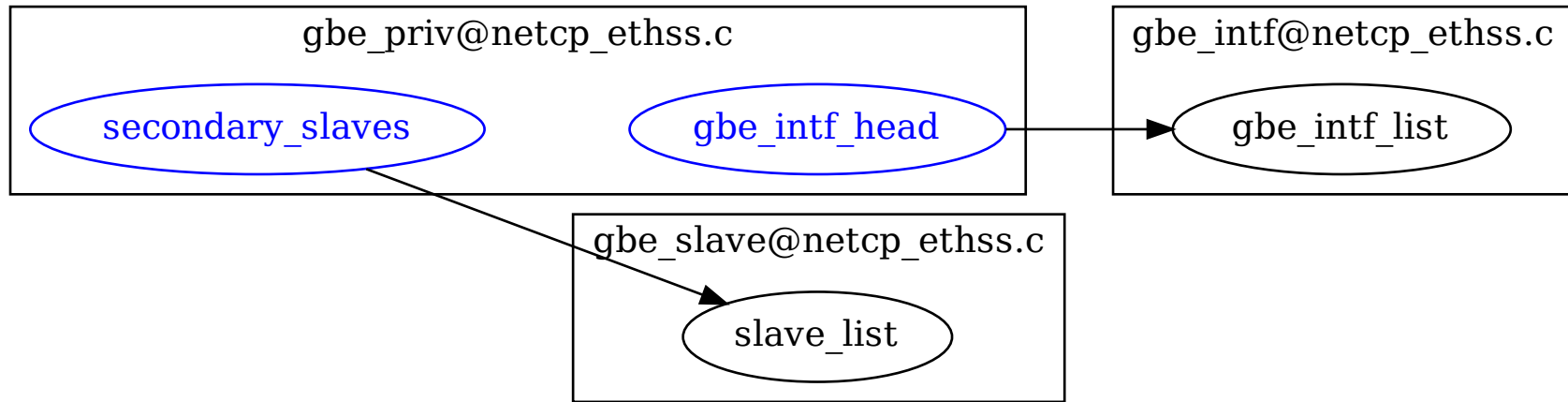


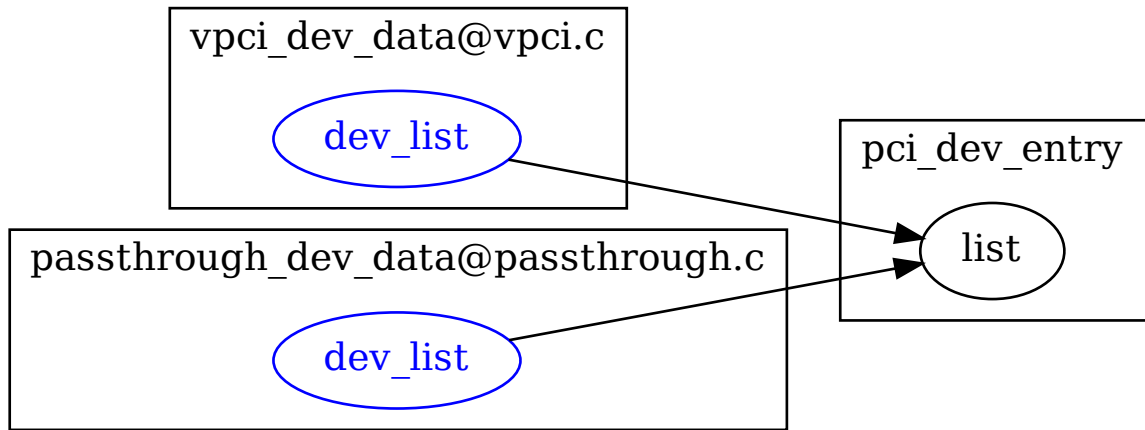


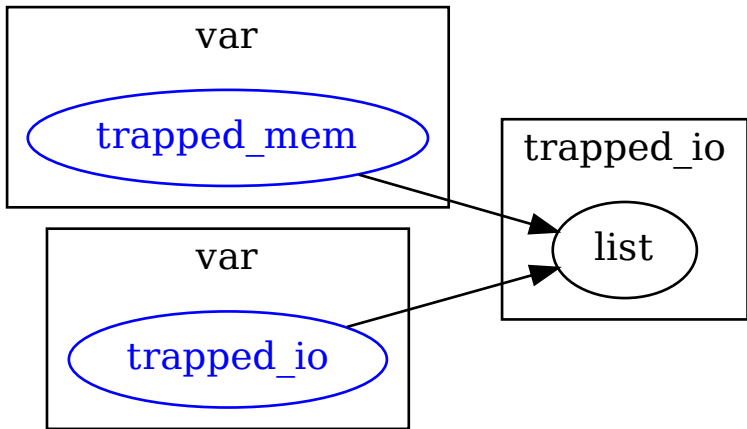


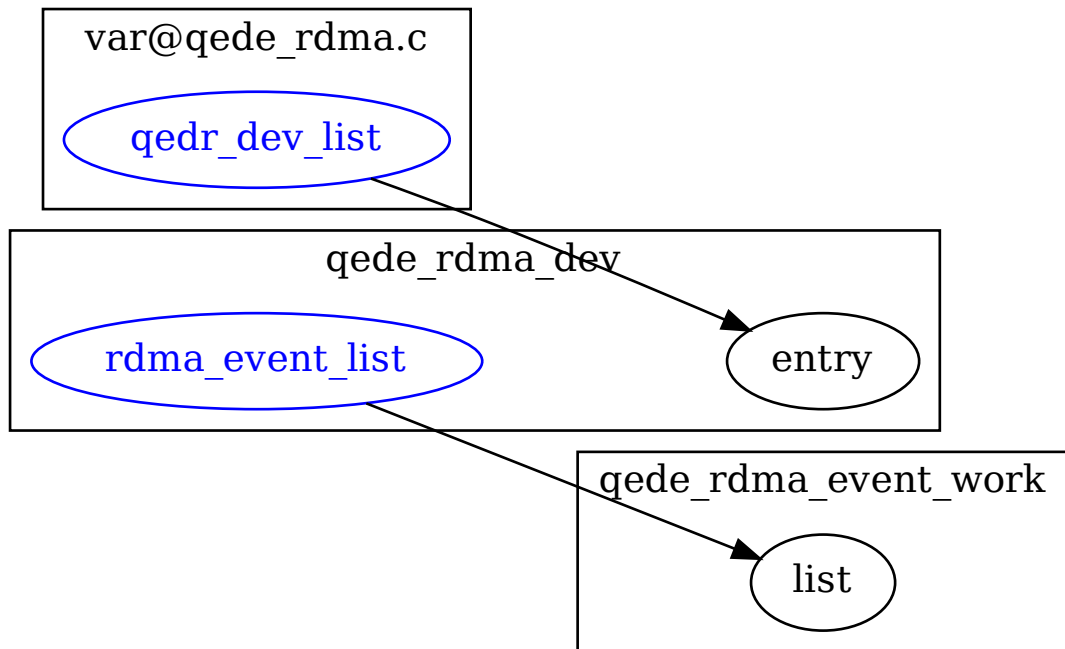


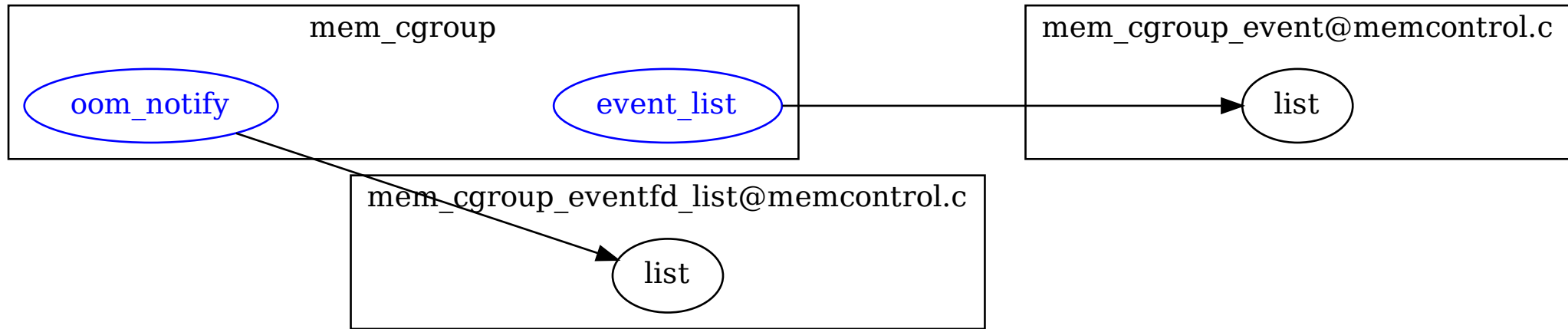


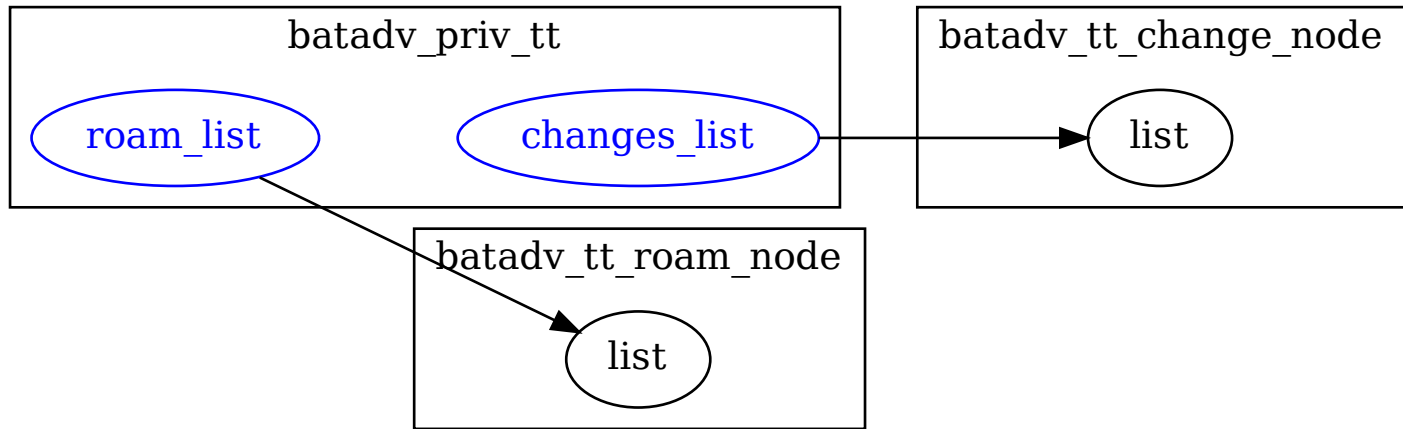


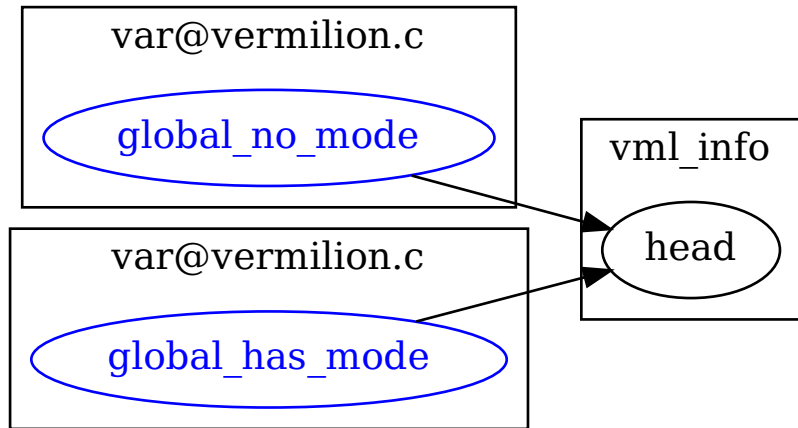


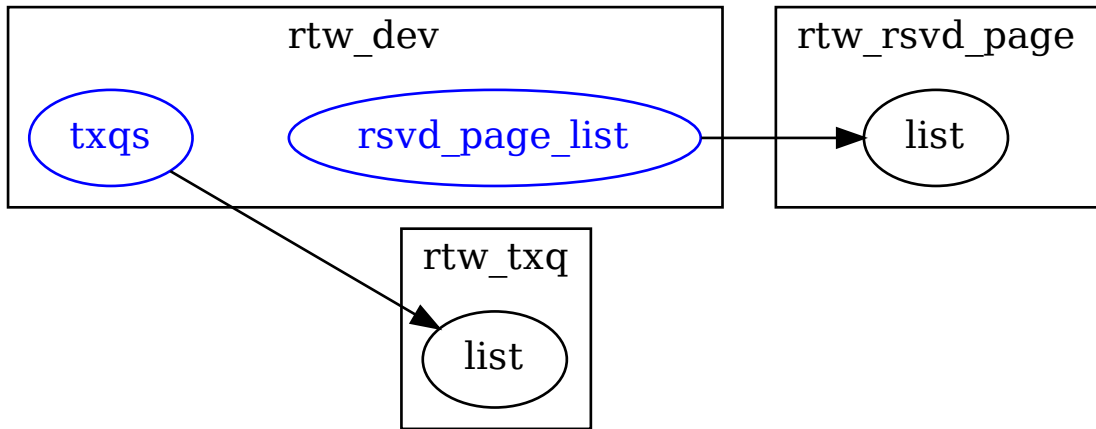


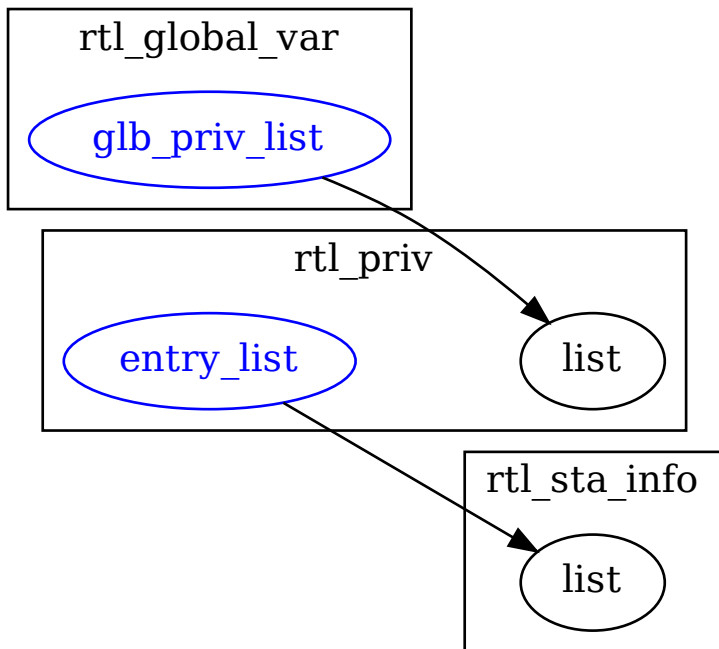


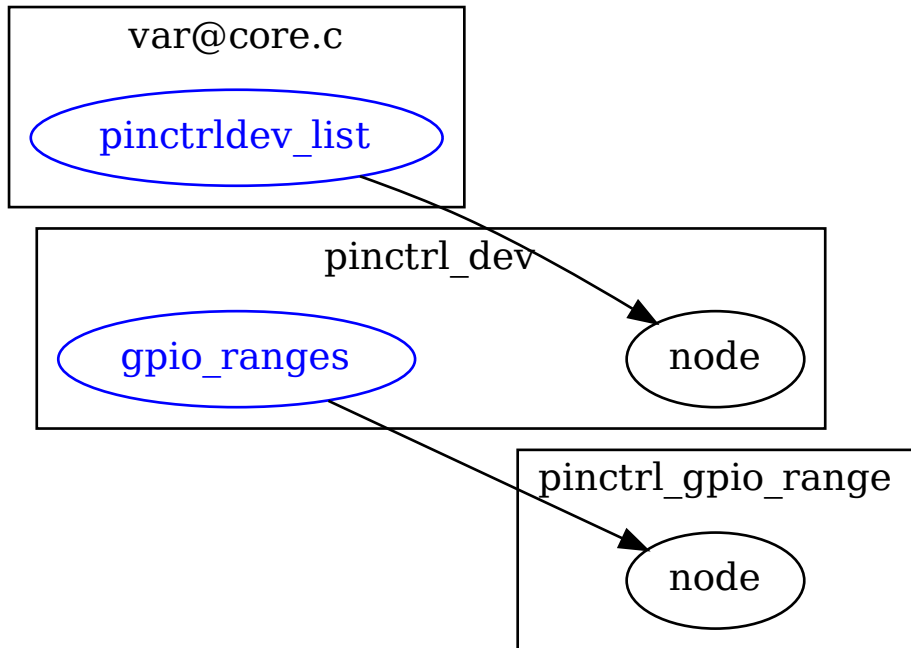


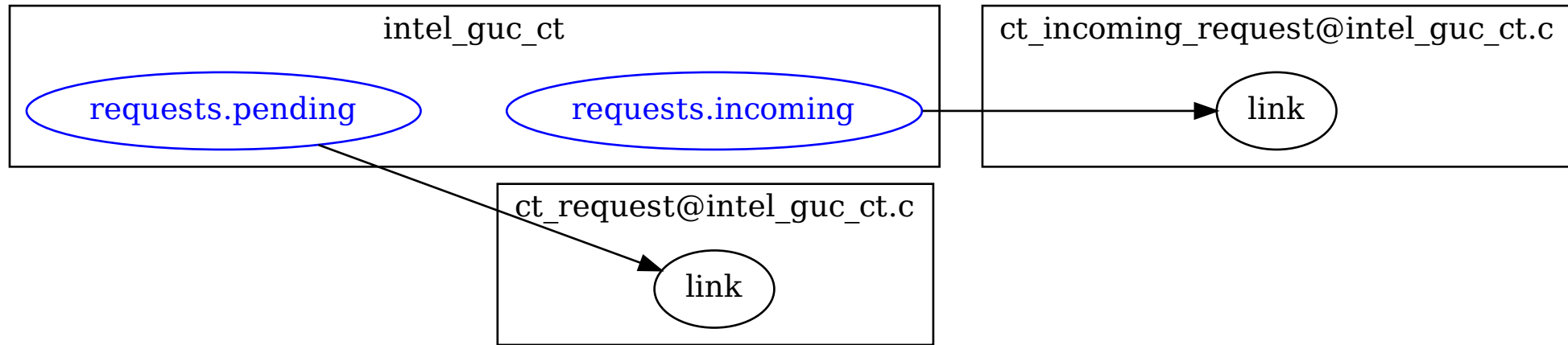


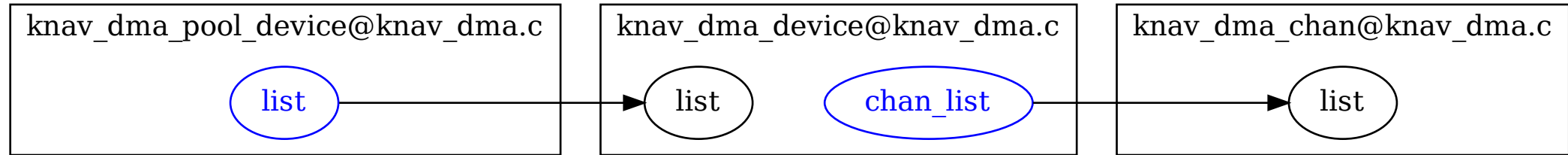


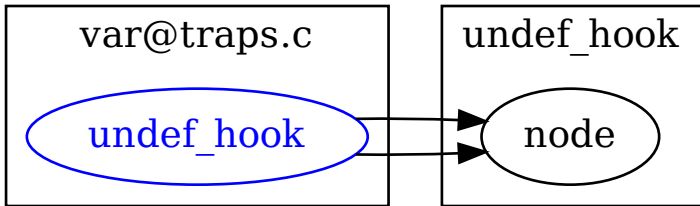


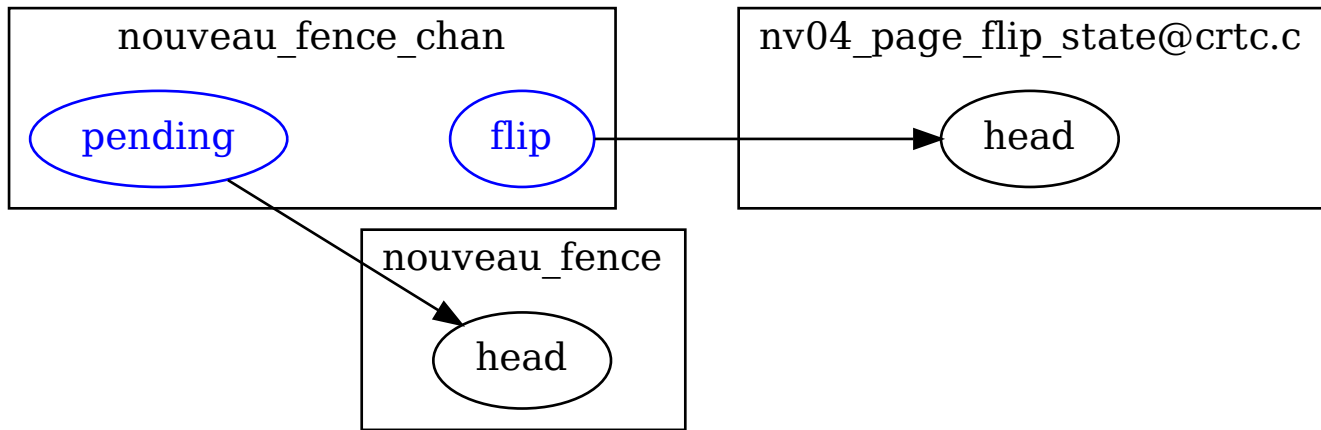


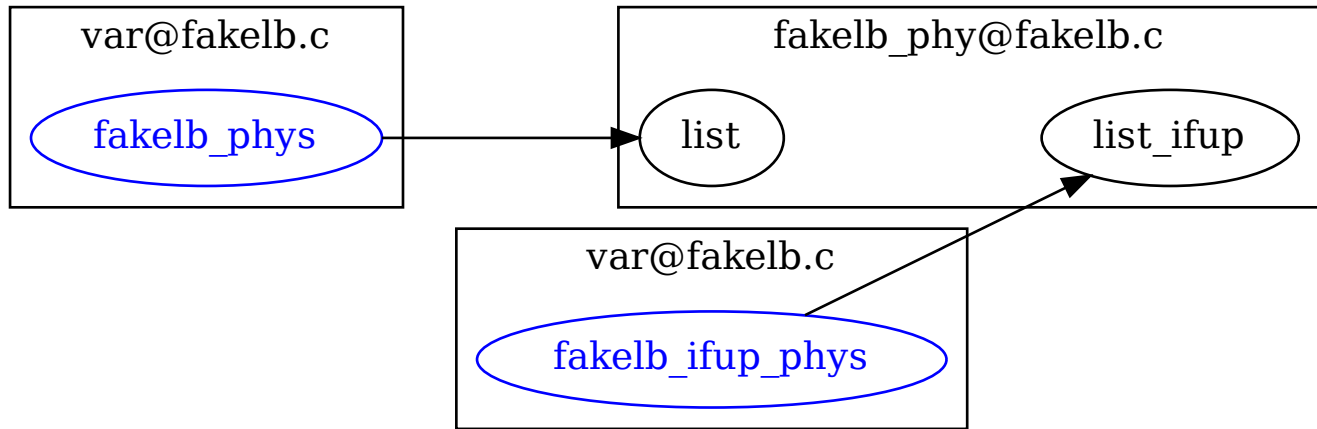


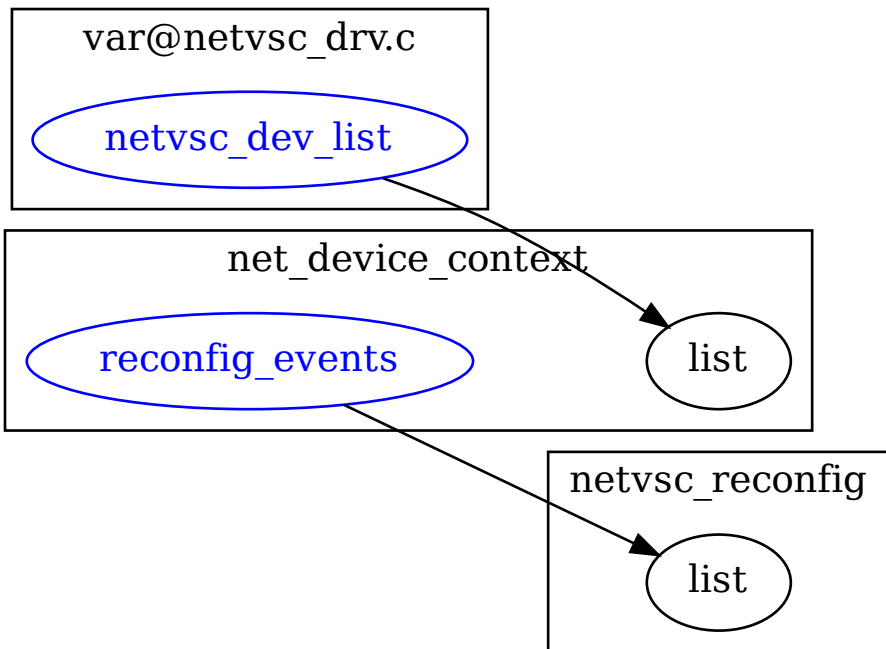


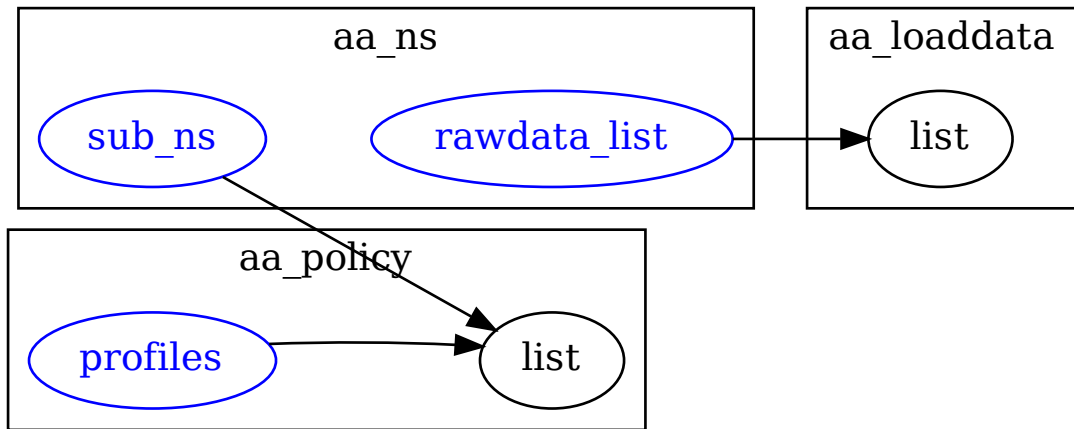


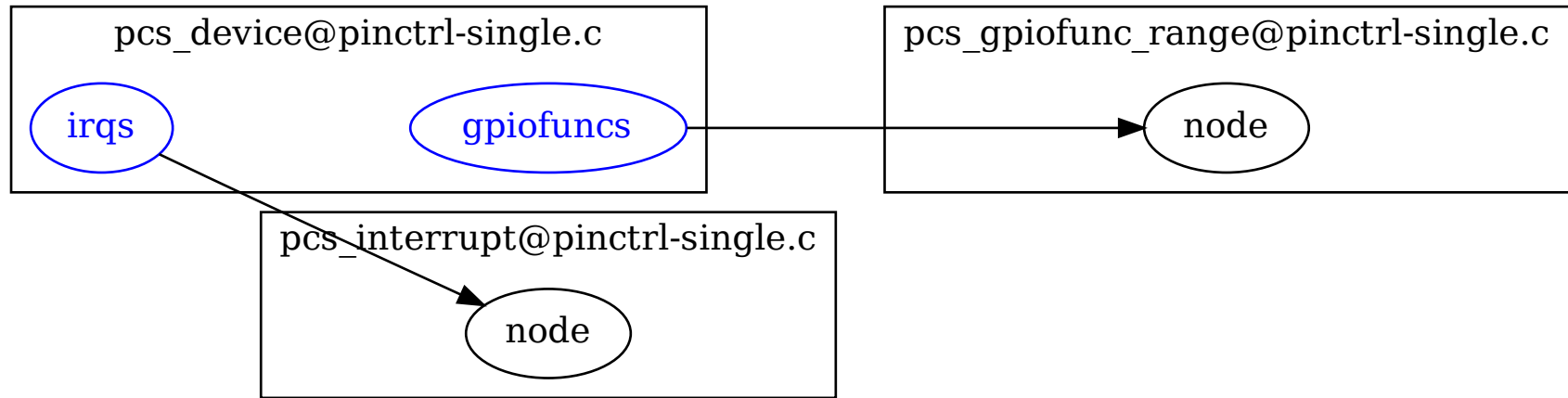


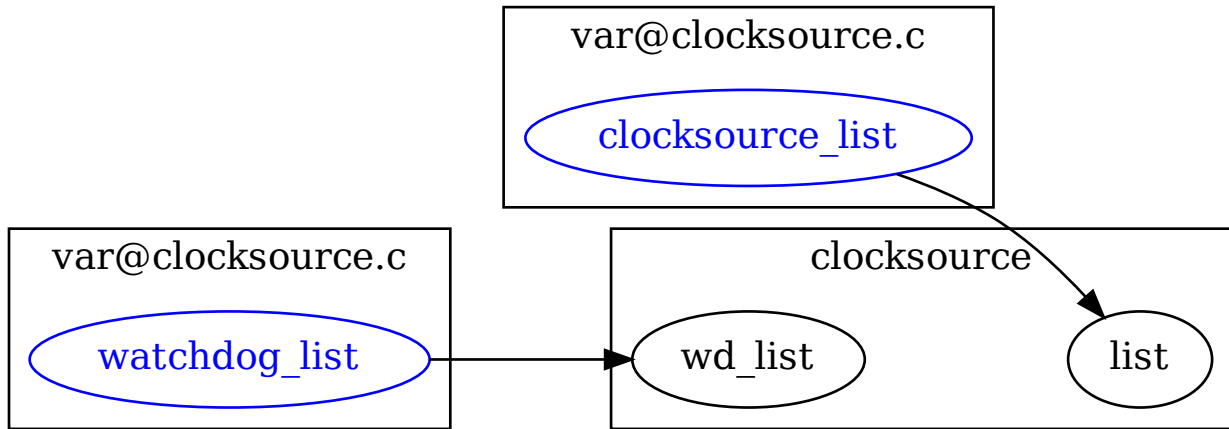


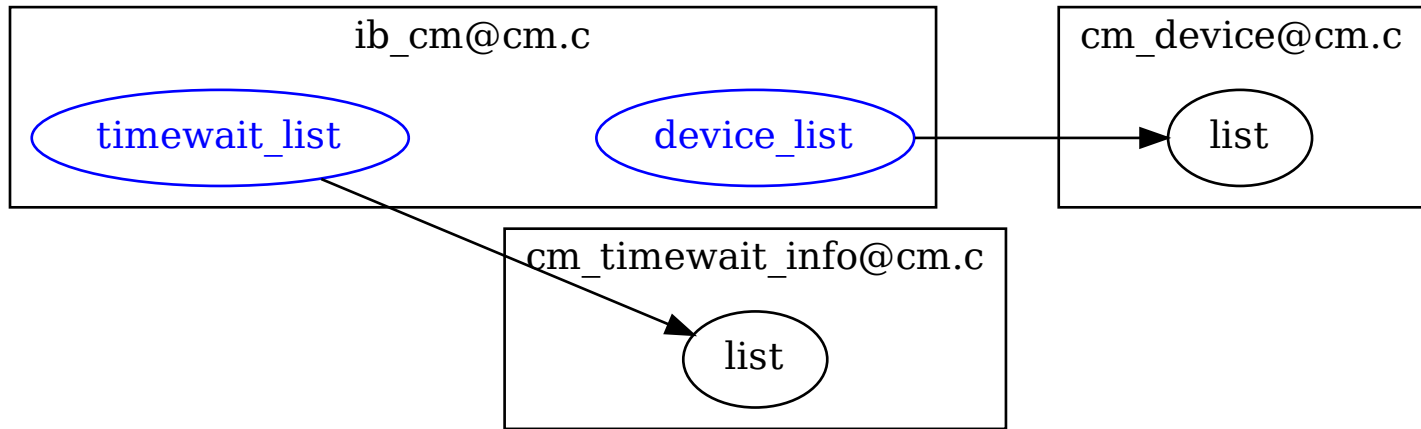


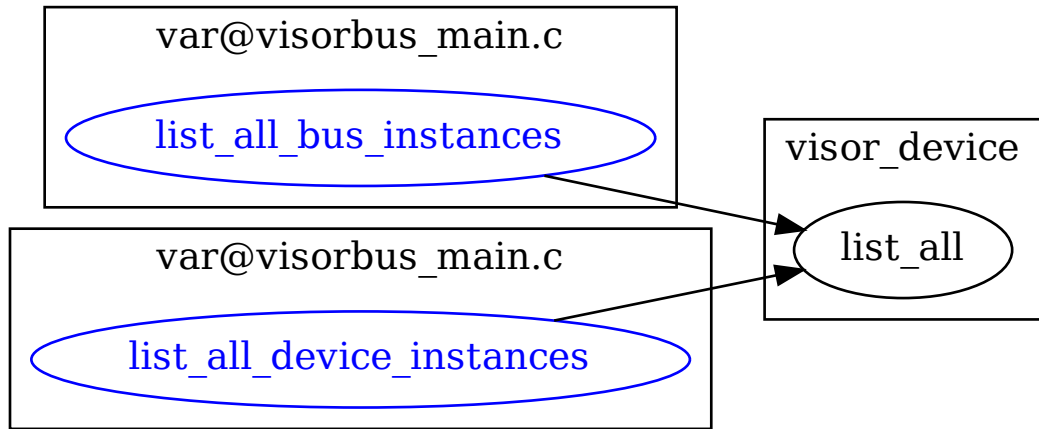


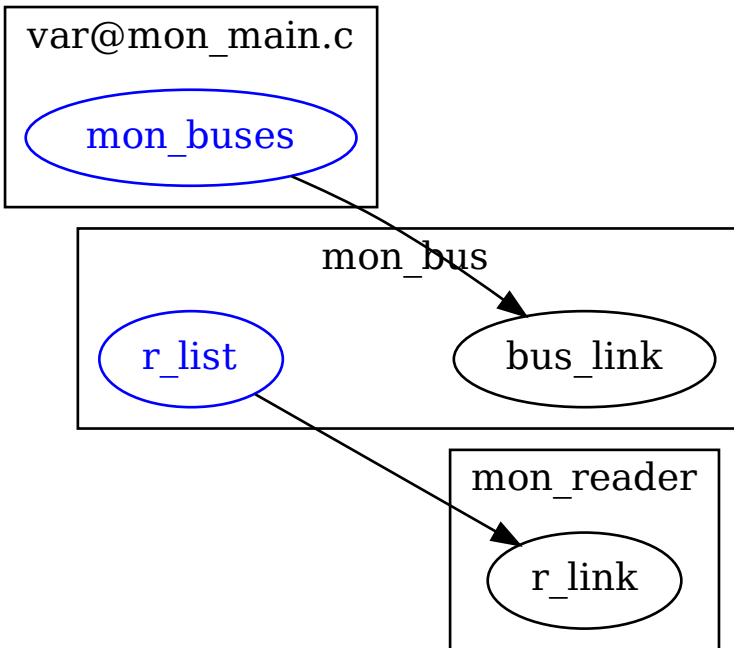


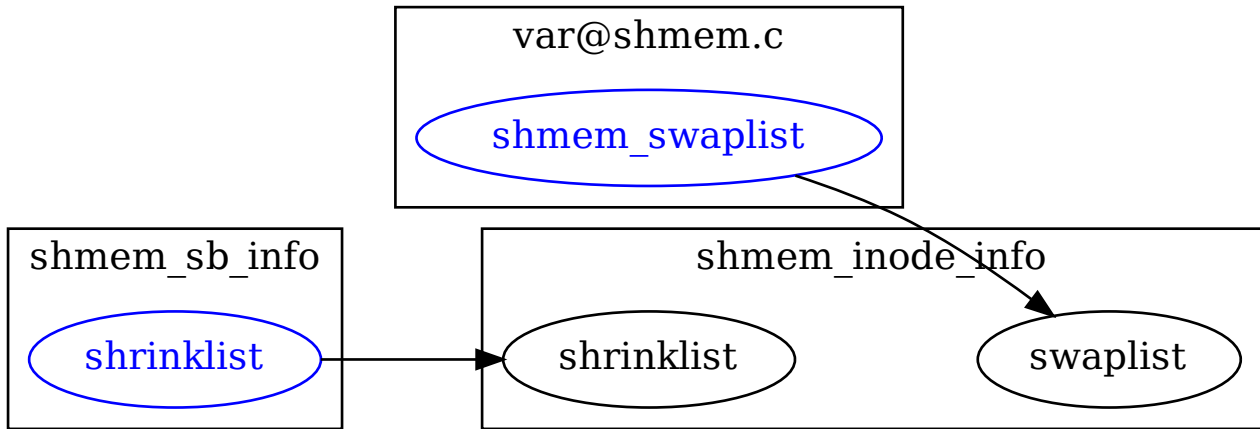


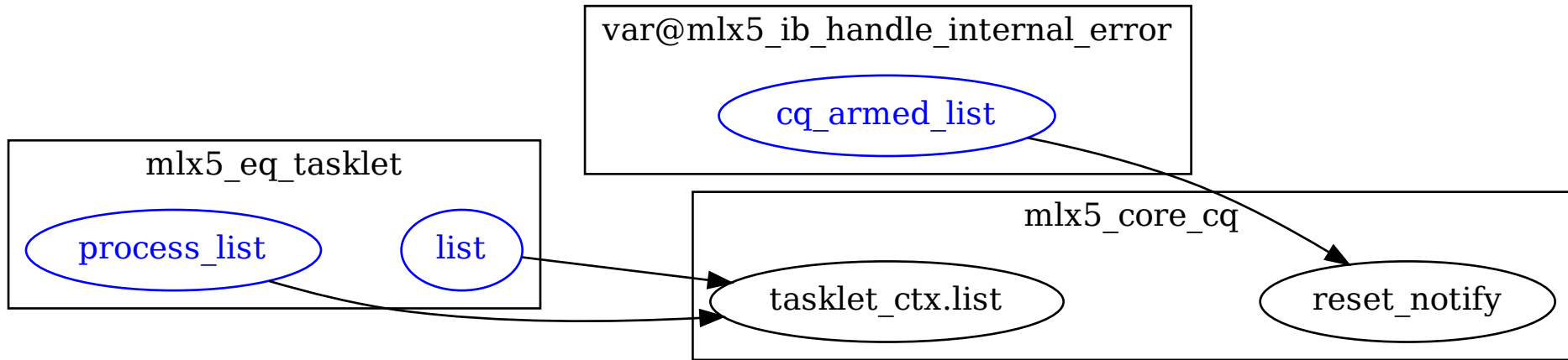


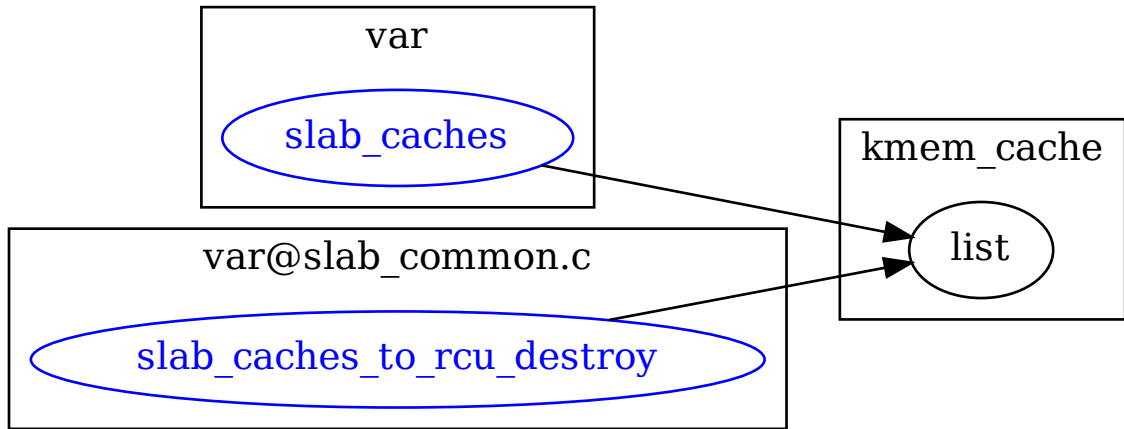


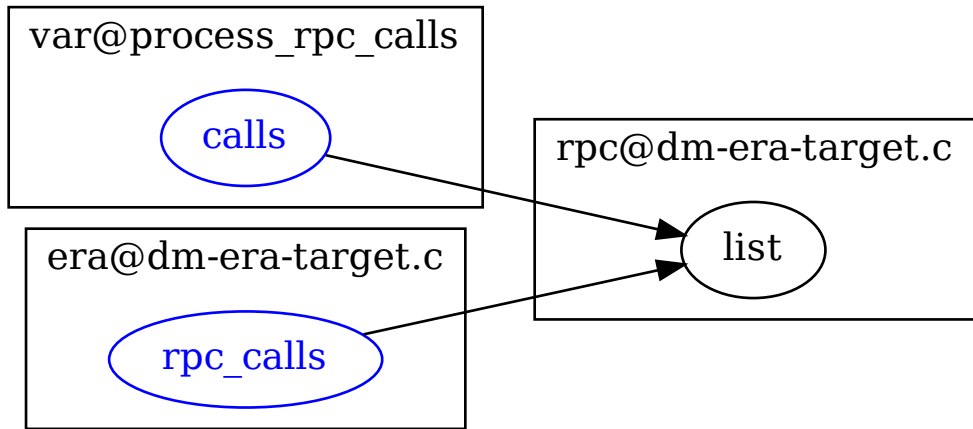


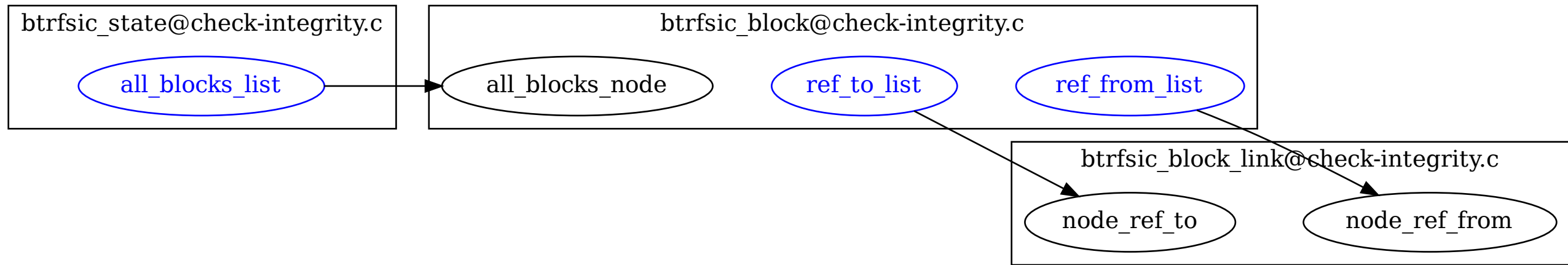


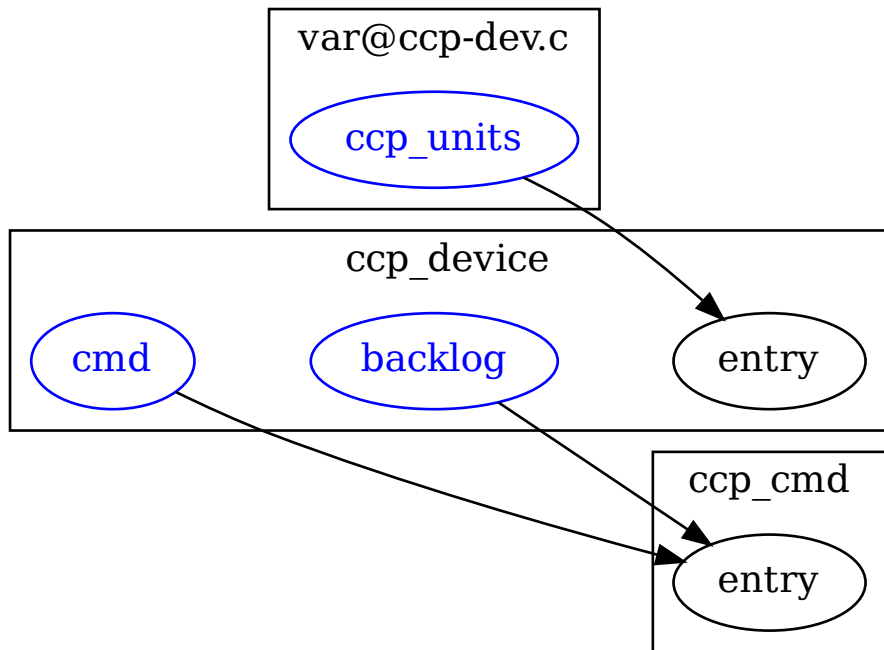


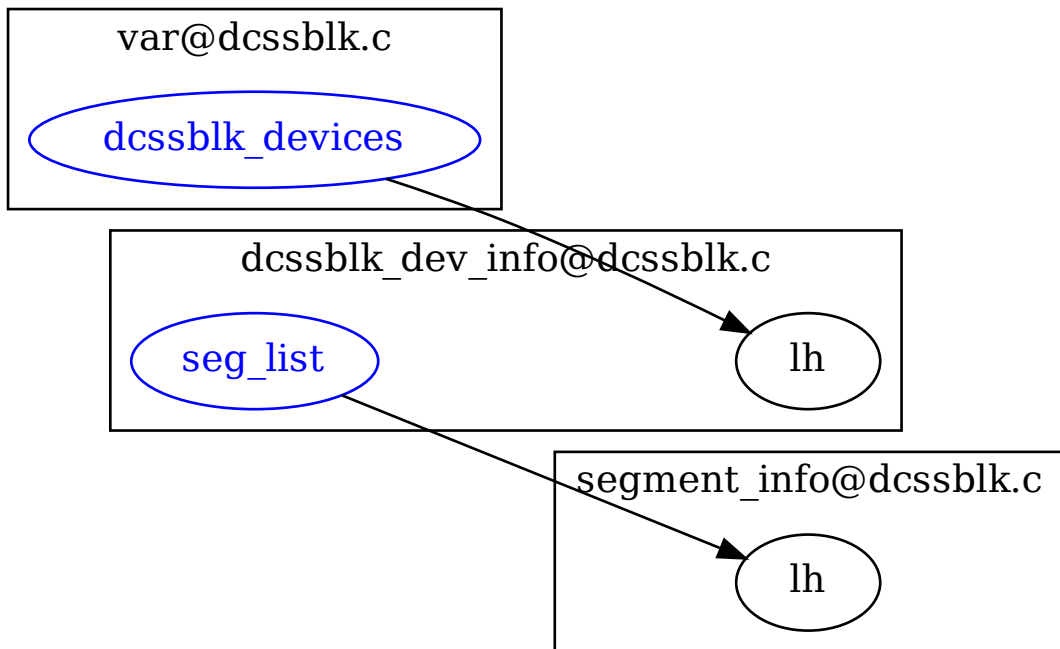


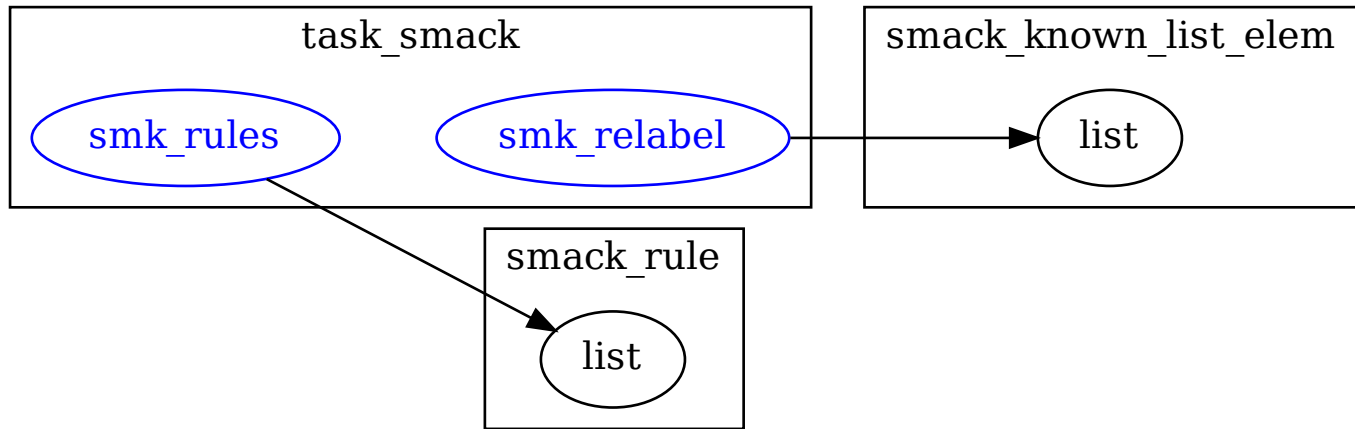


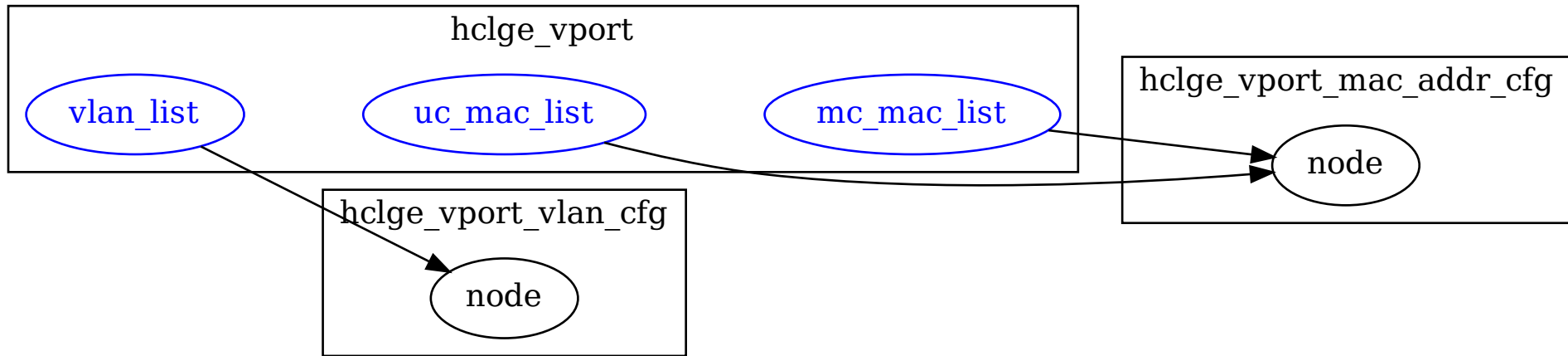


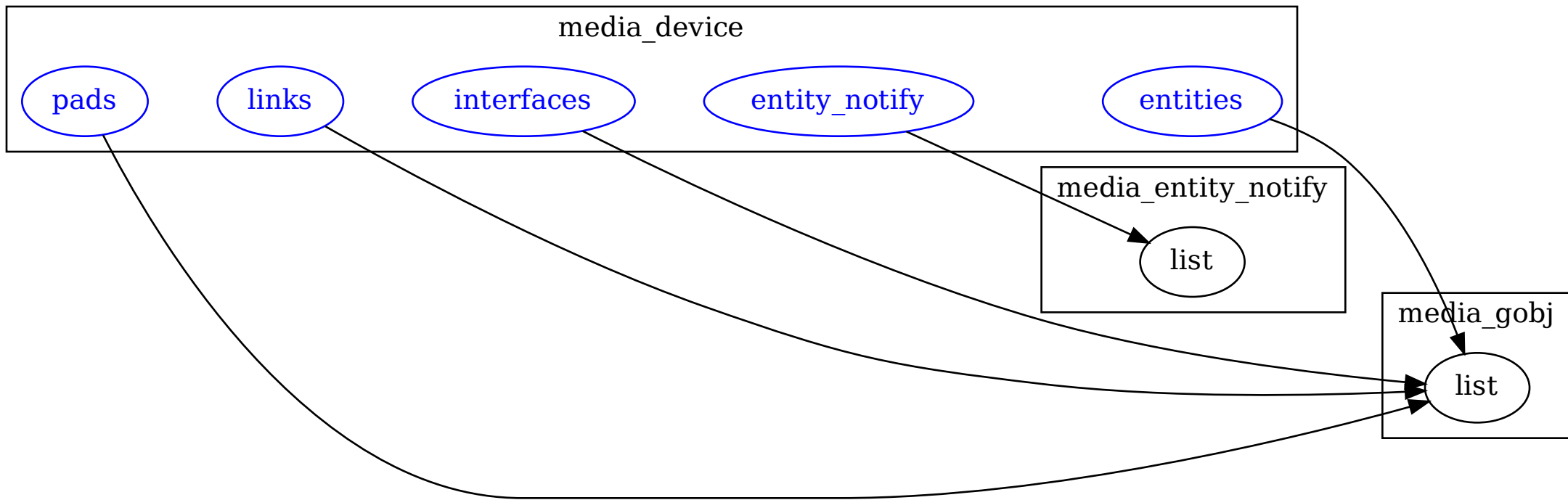


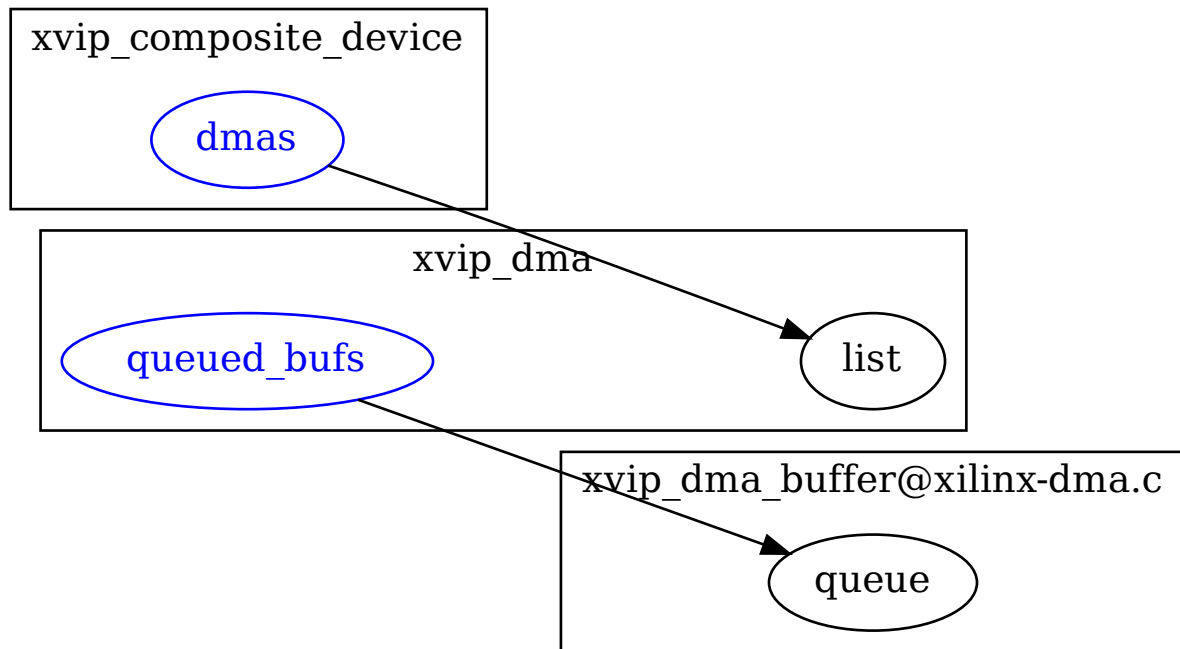


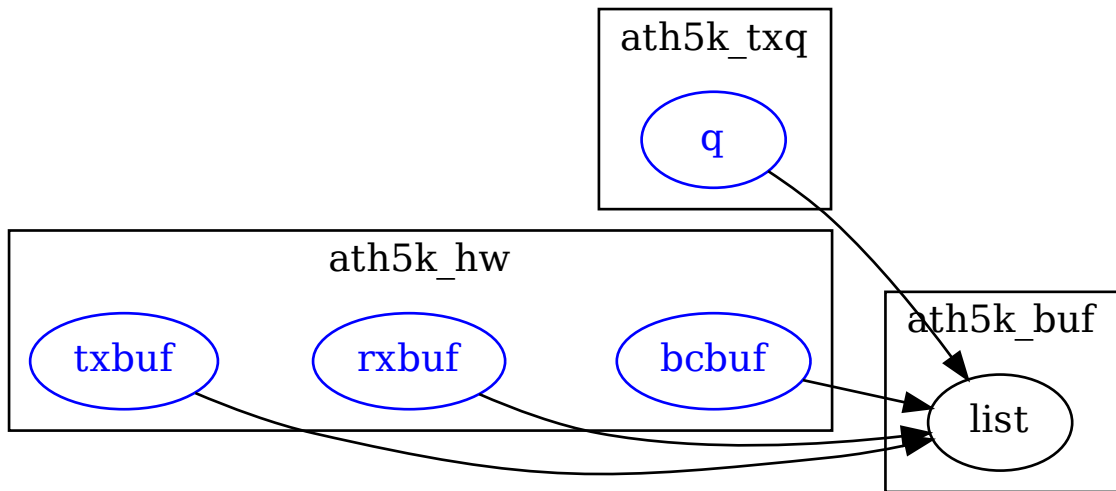


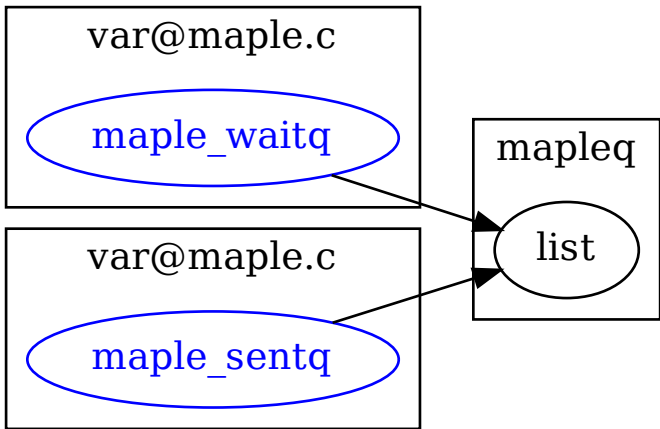


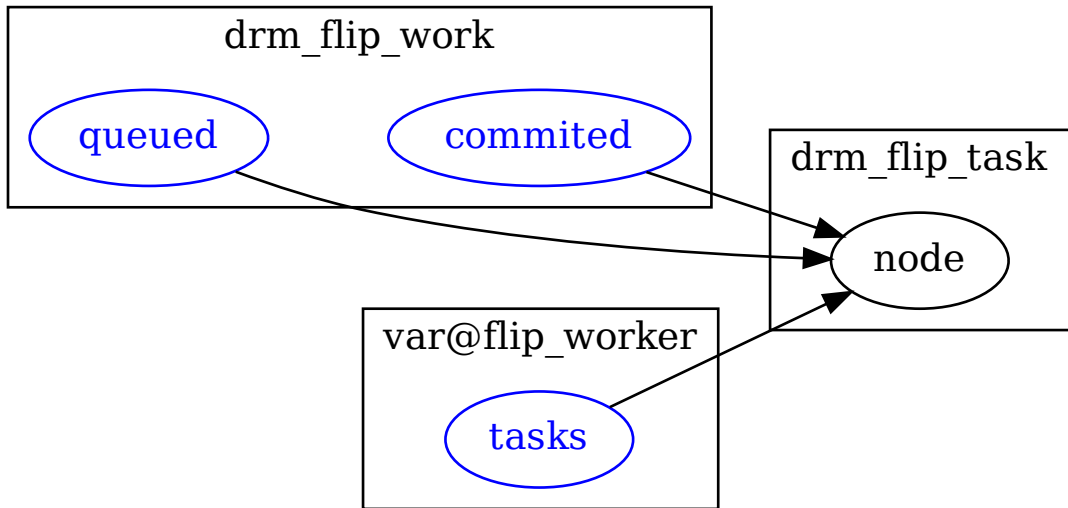


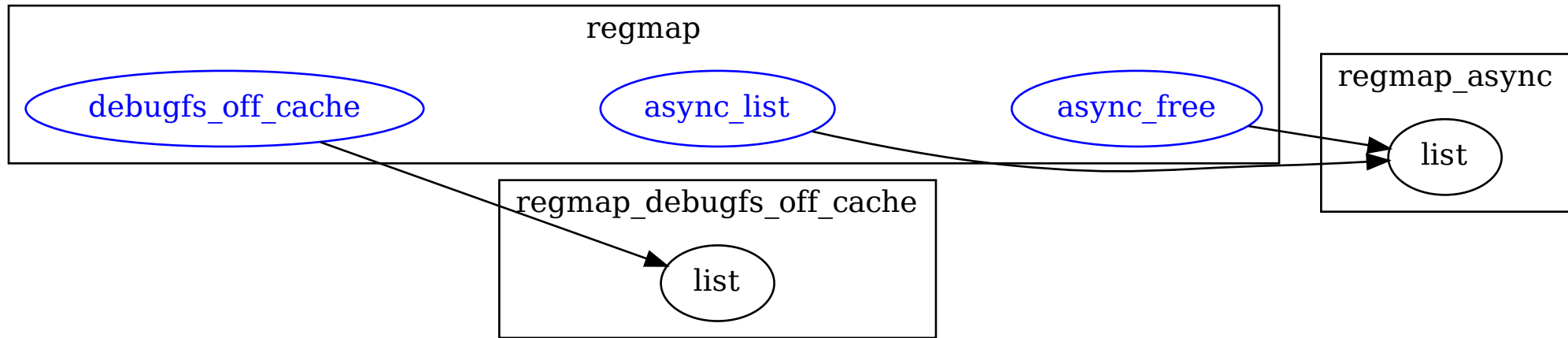


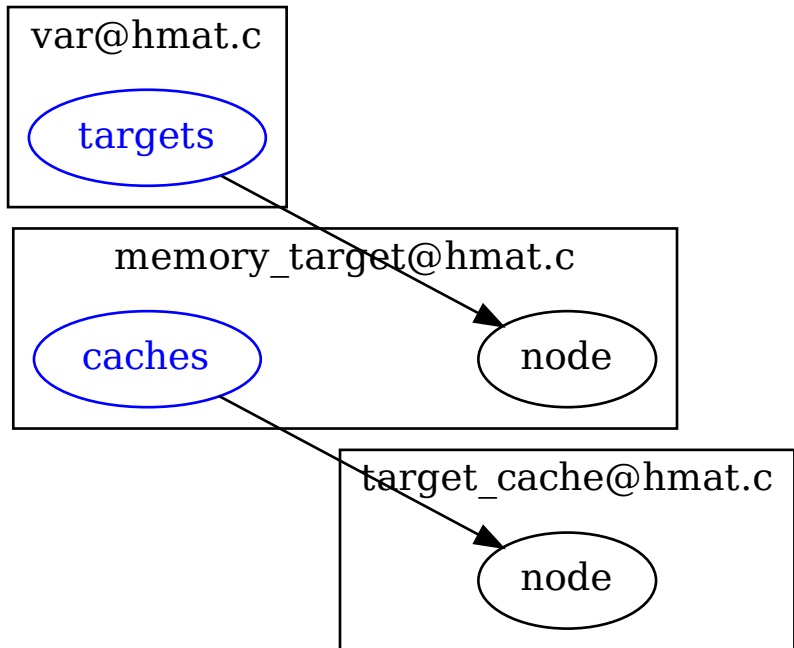


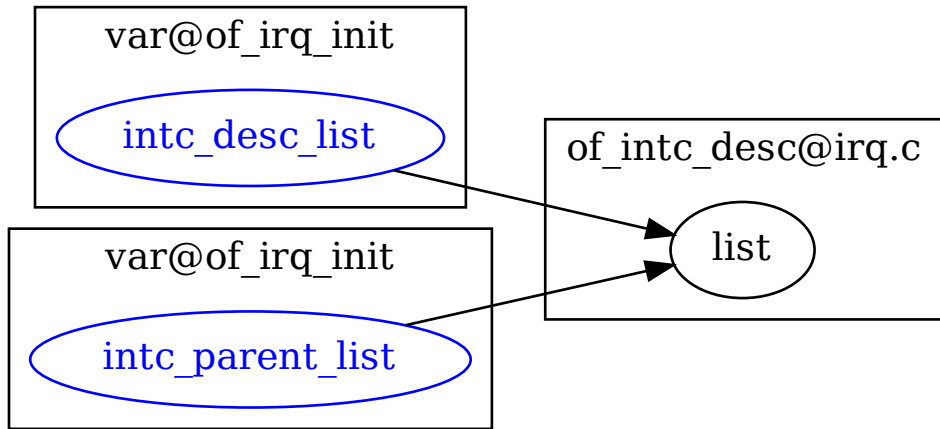


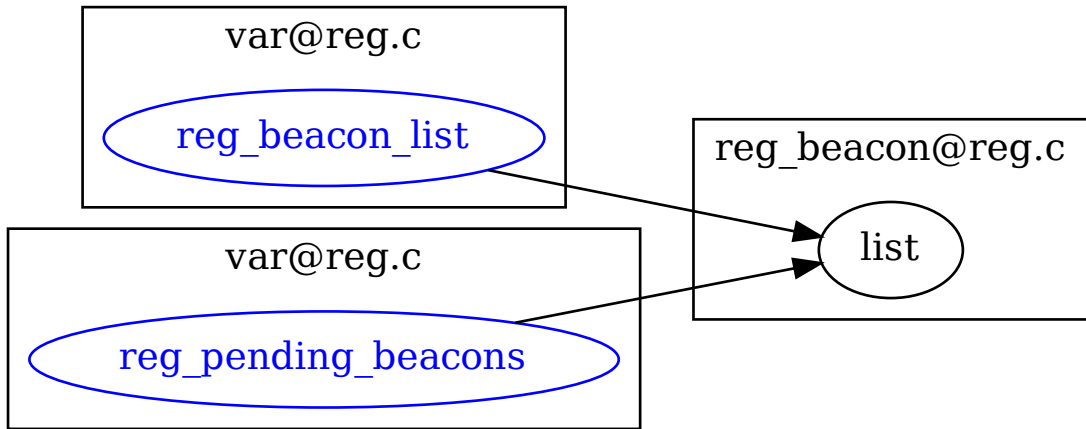












var@batadv_mcast_mla_bridge_get

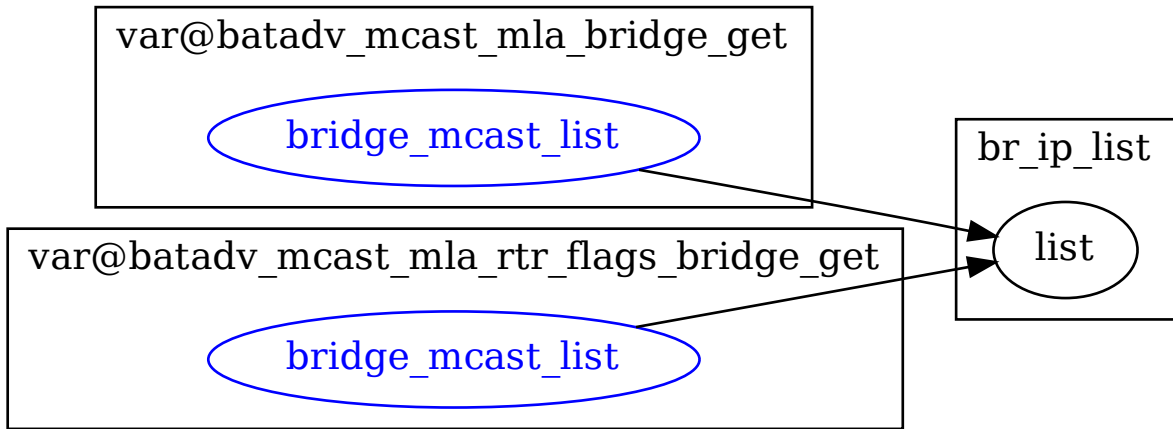
bridge_mcast_list

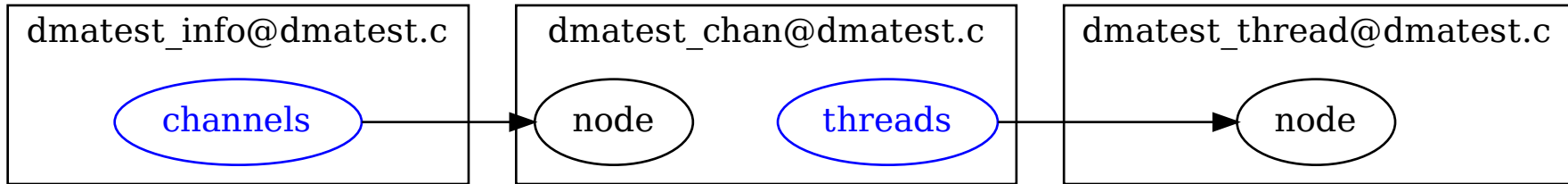
var@batadv_mcast_mla_rtr_flags_bridge_get

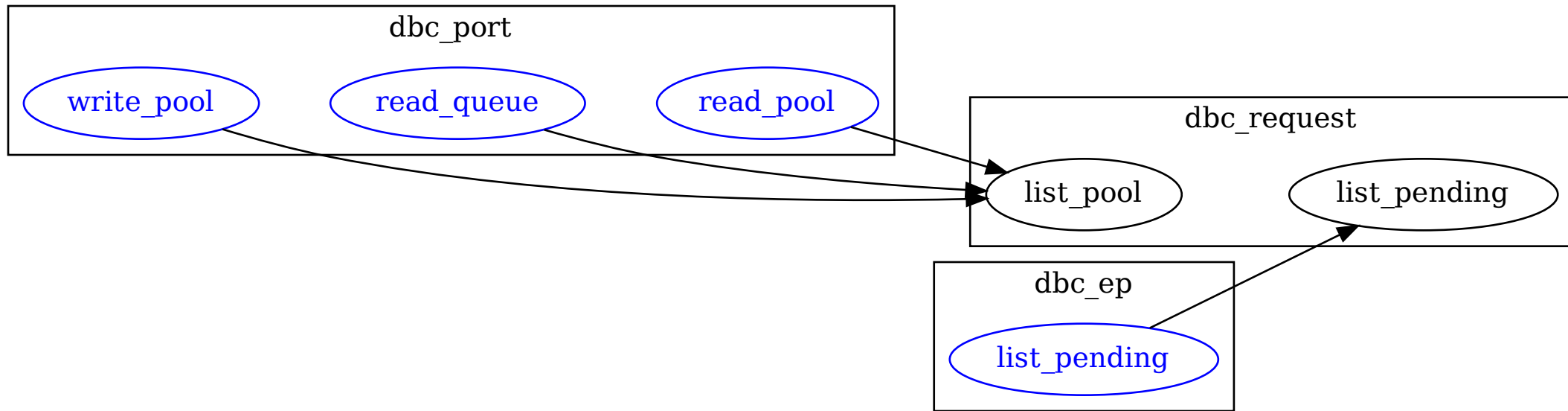
bridge_mcast_list

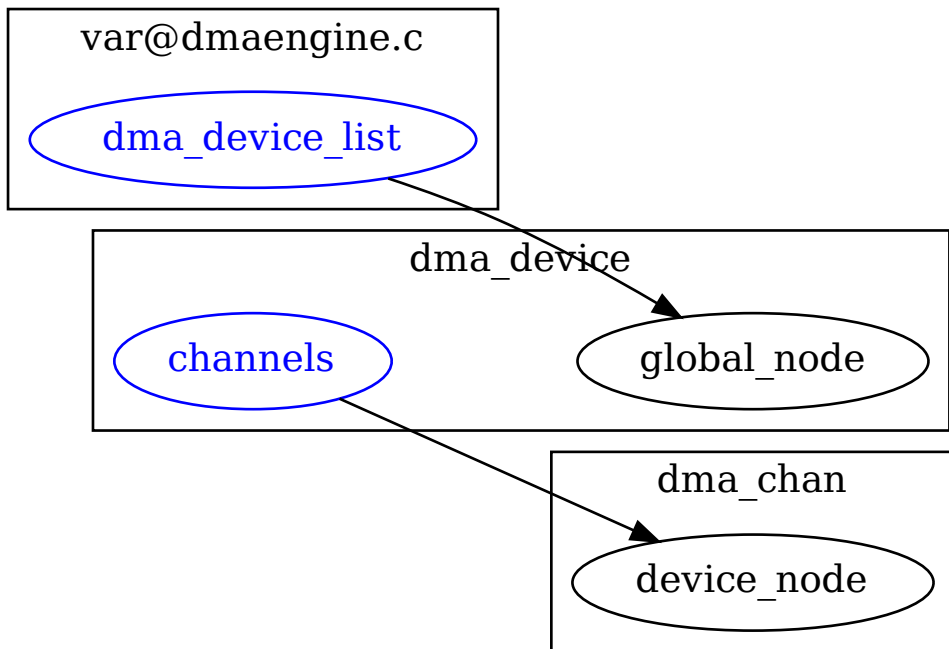
br_ip_list

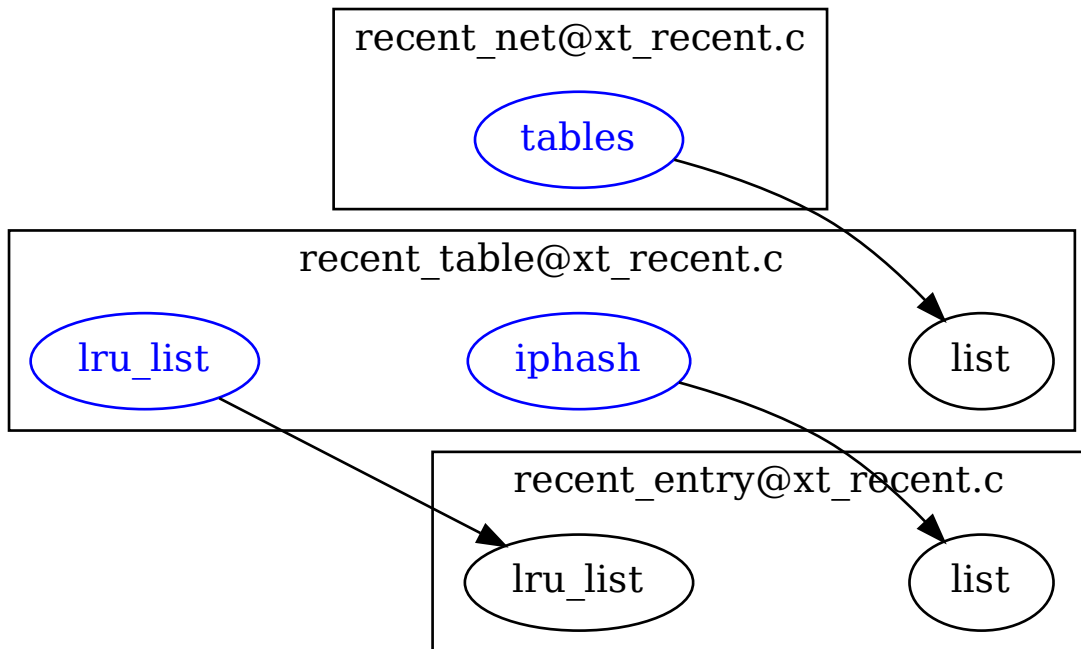
list

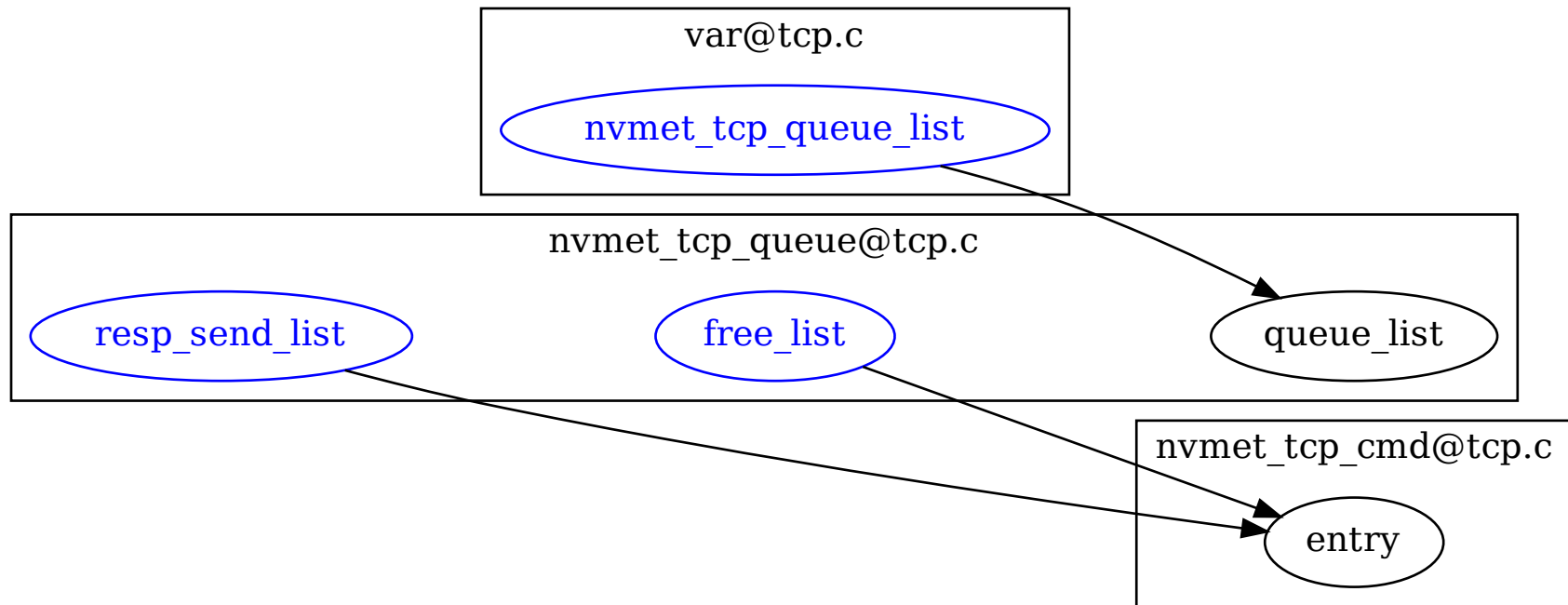


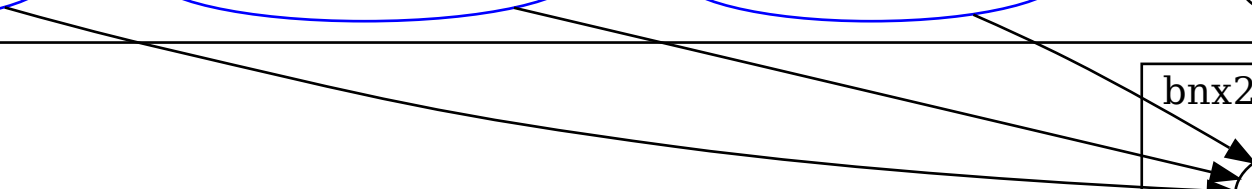
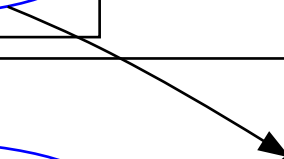
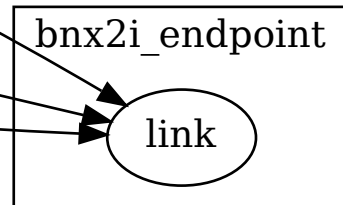
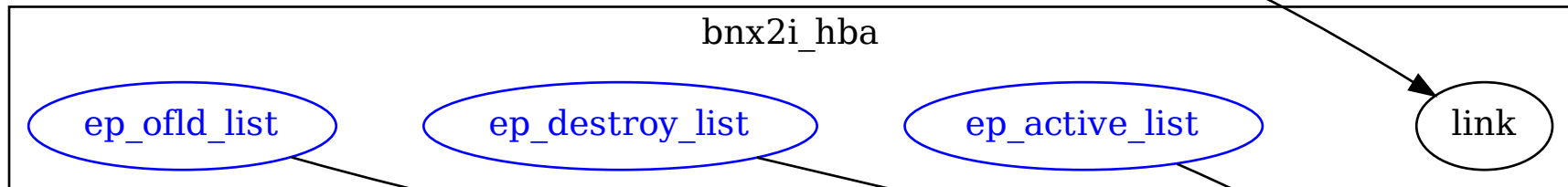
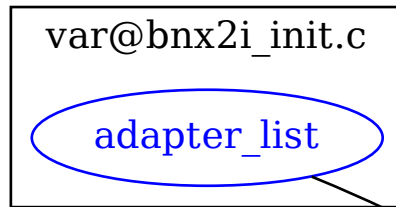


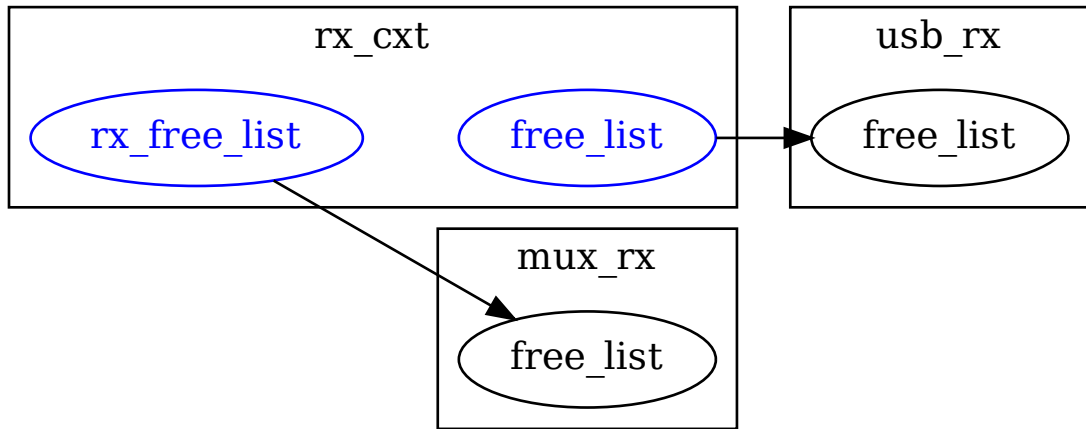


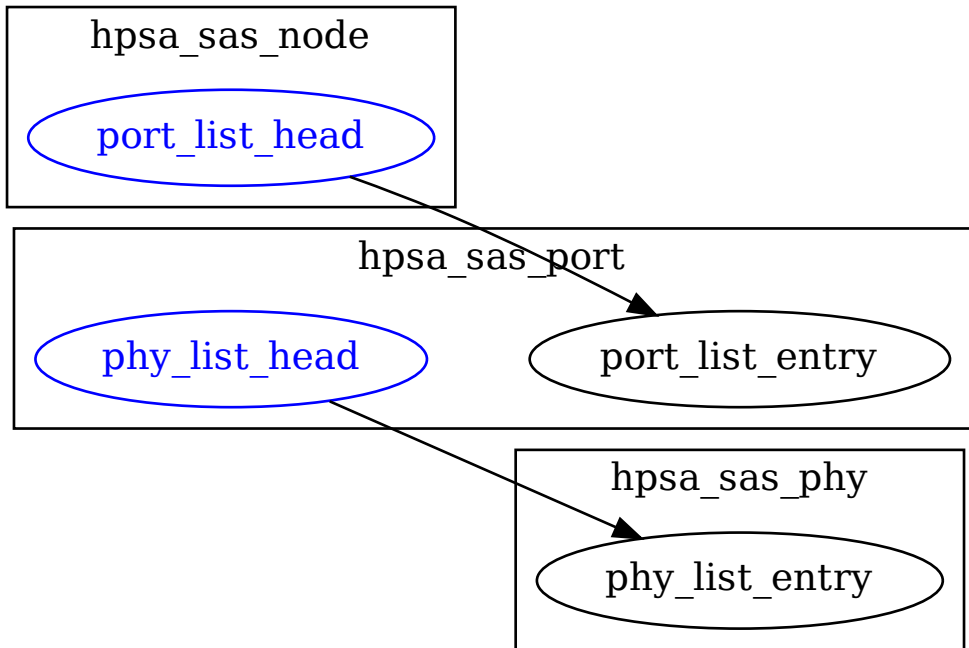


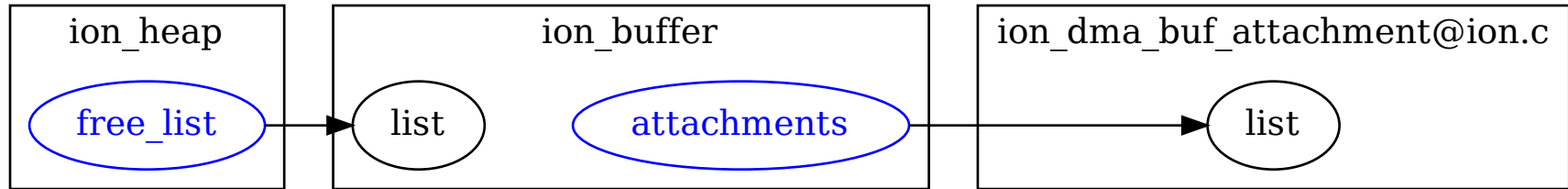


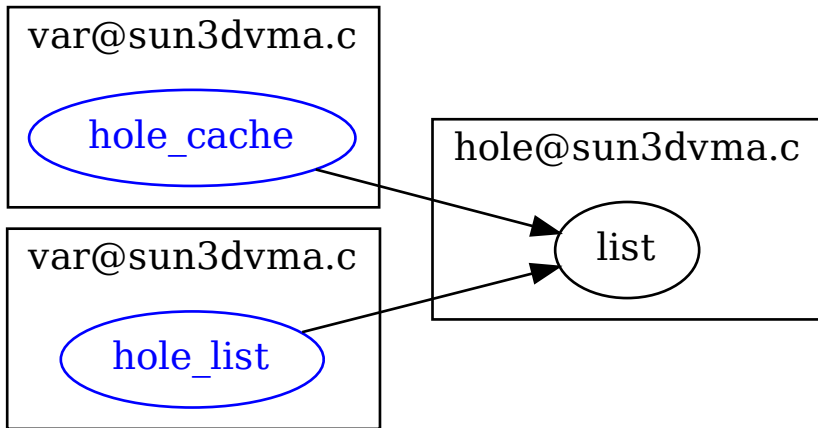


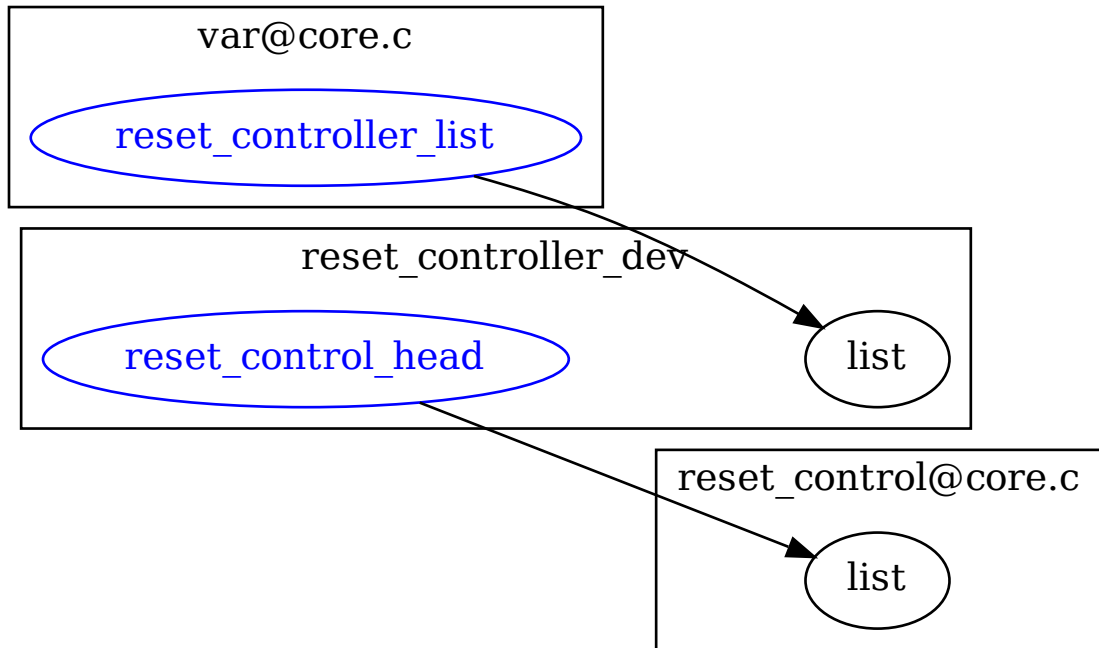


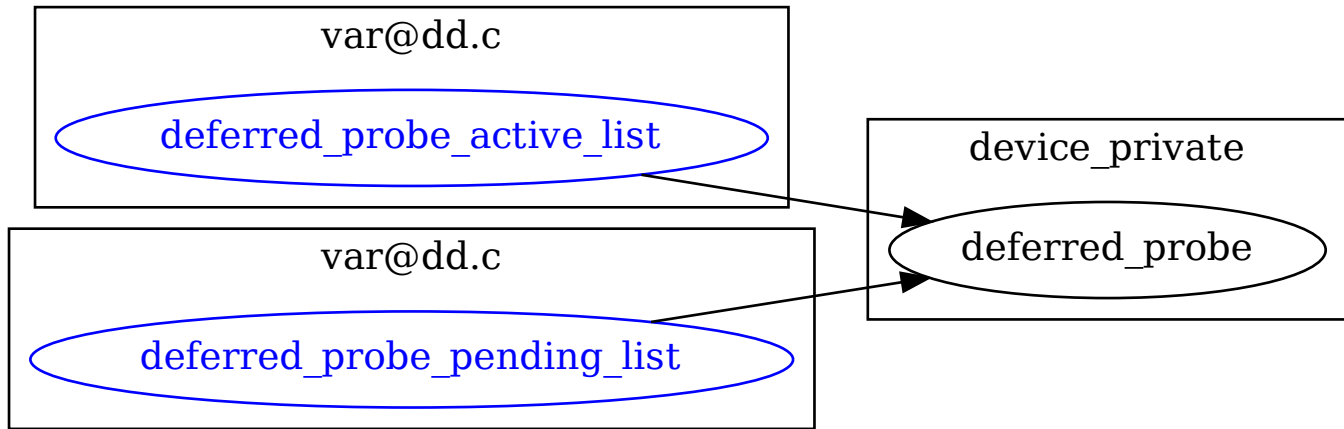


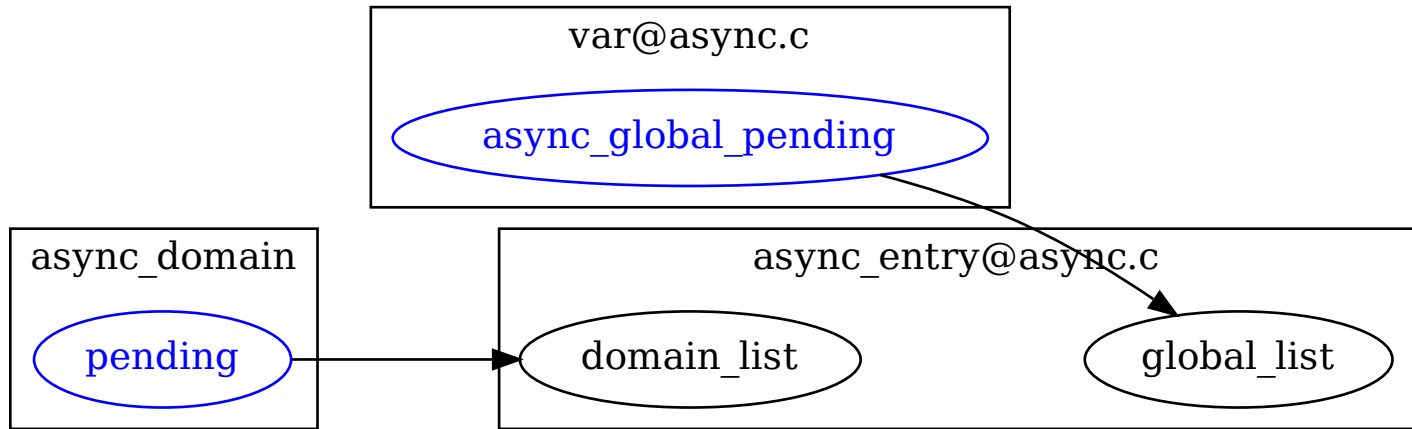


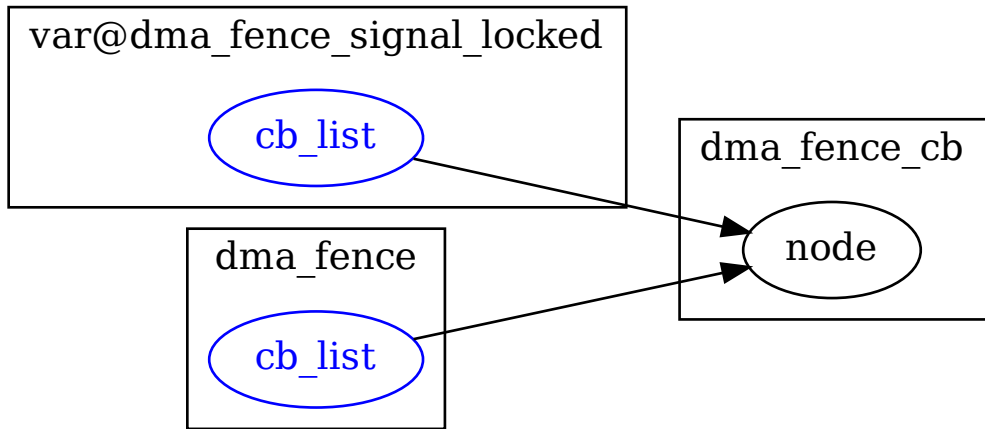


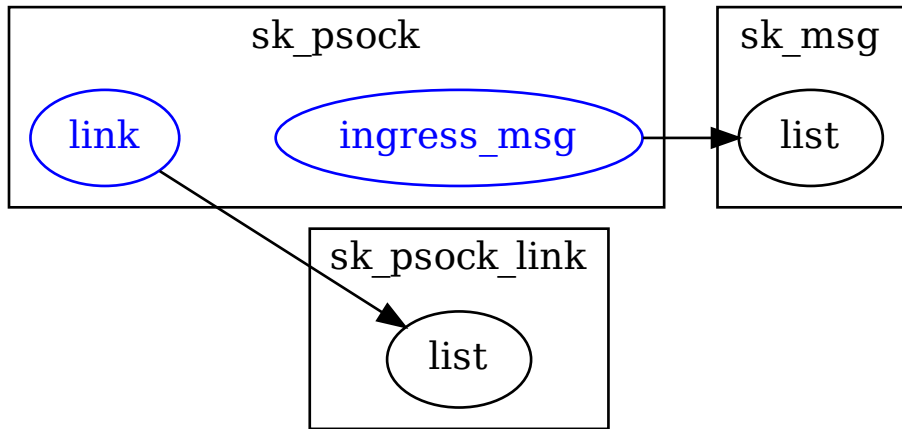


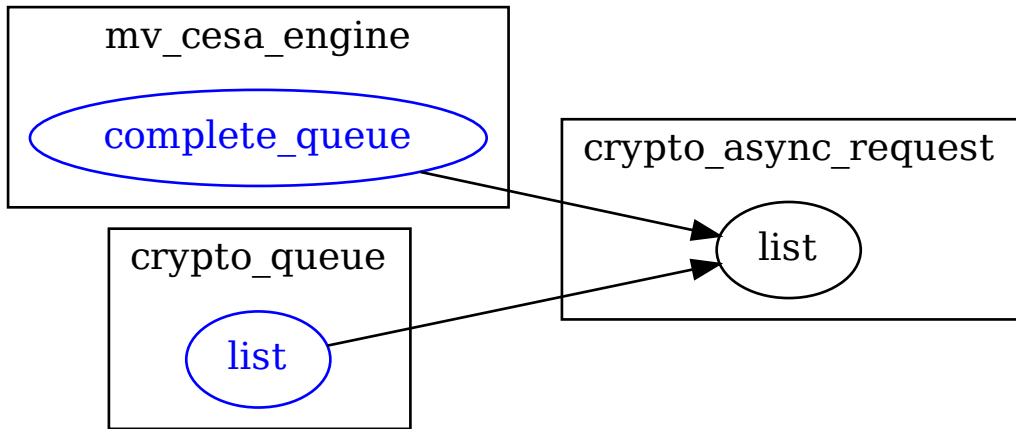


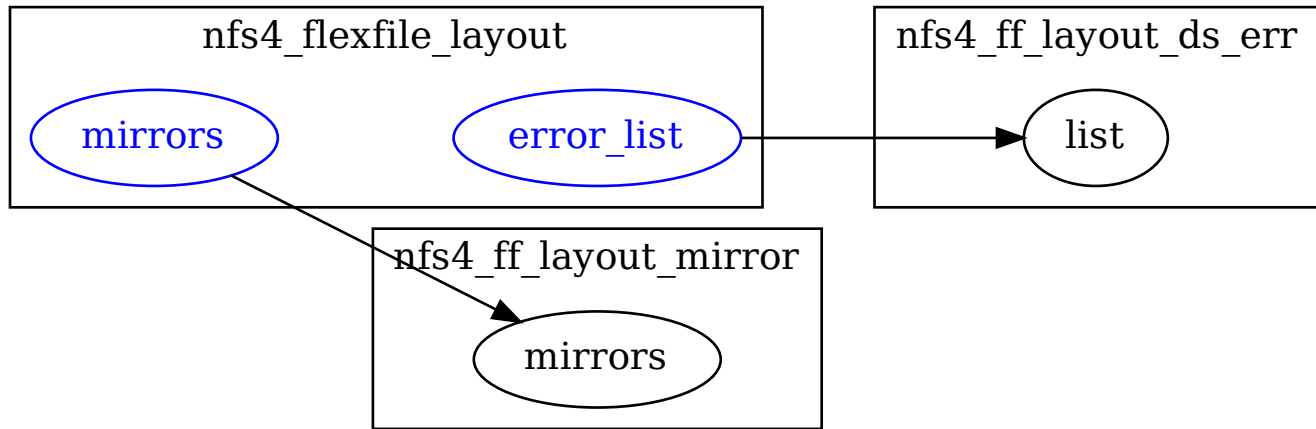












var@qla_target.c

qla_tgt_glist

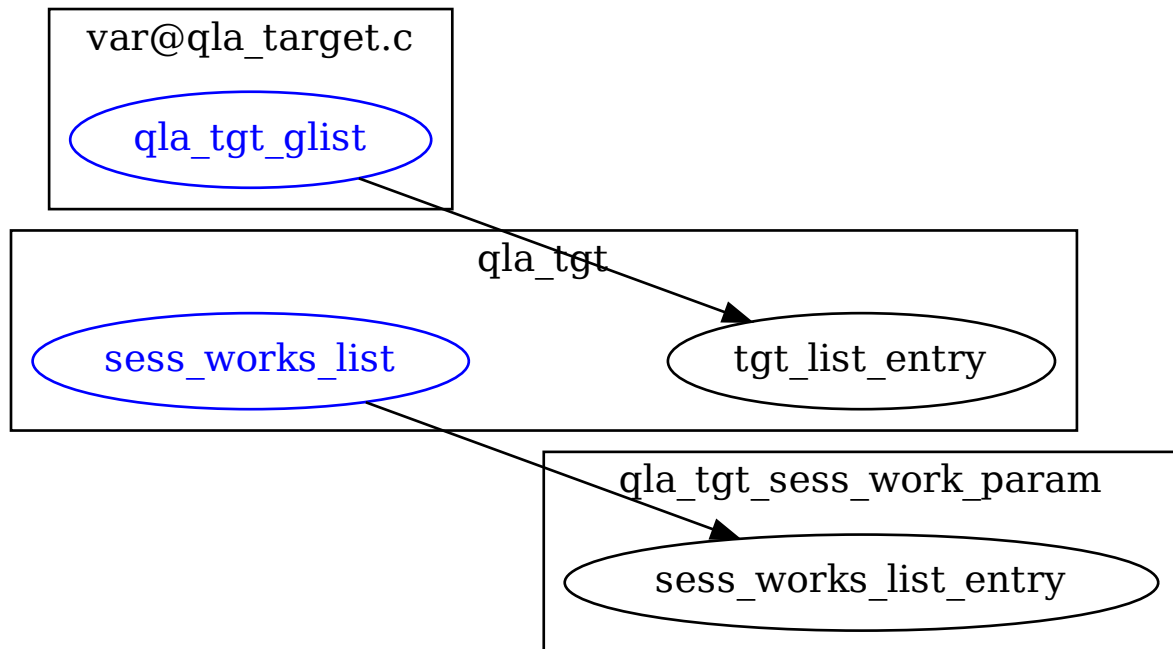
qla_tgt

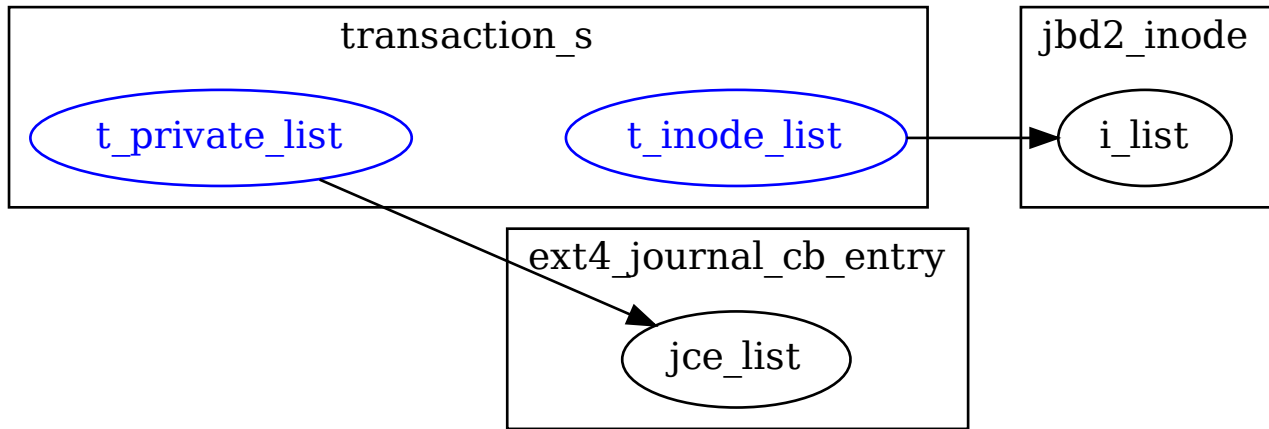
sess_works_list

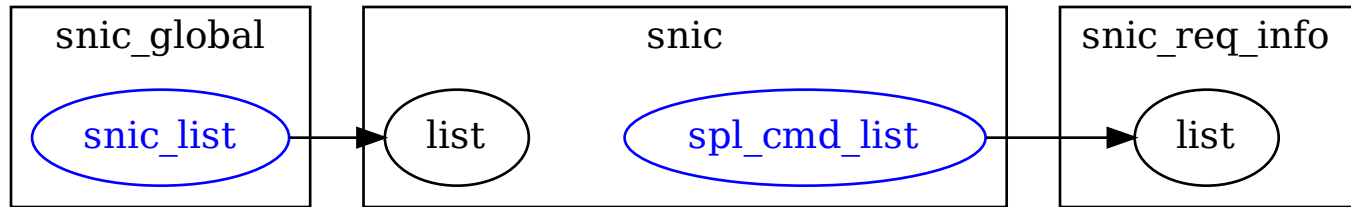
tgt_list_entry

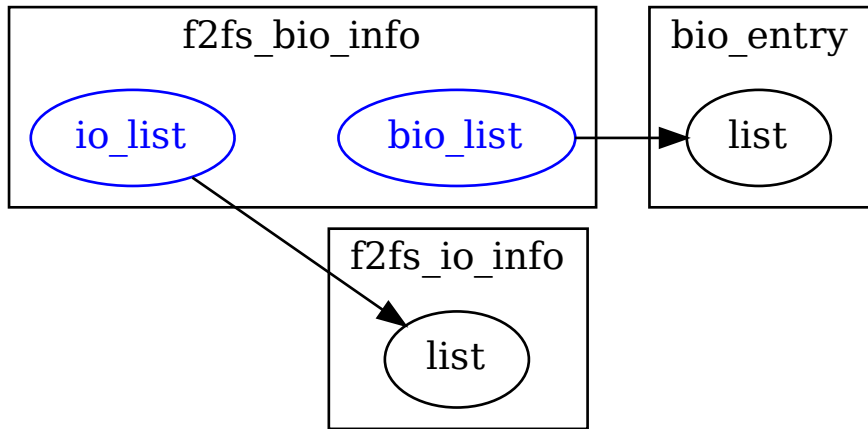
qla_tgt_sess_work_param

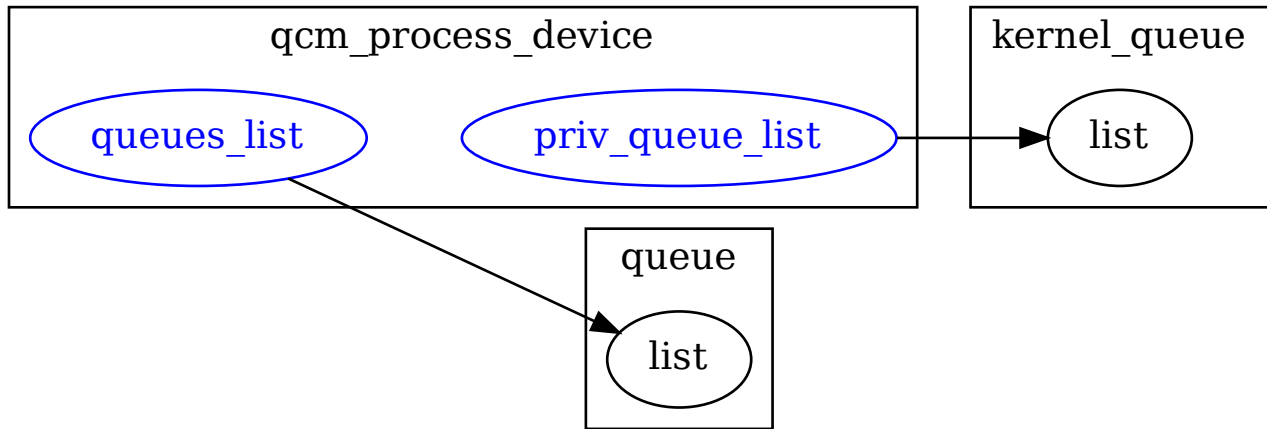
sess_works_list_entry

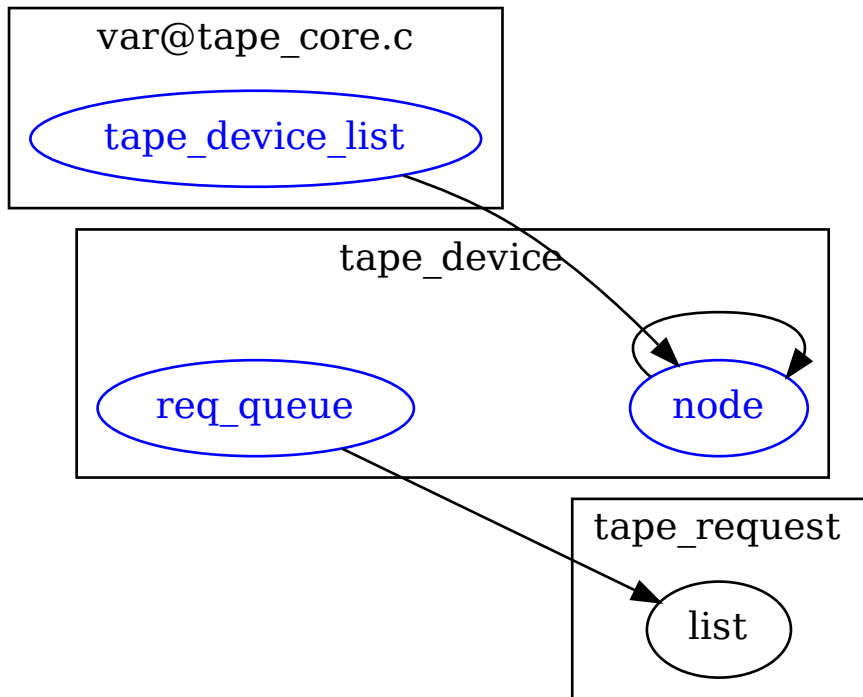


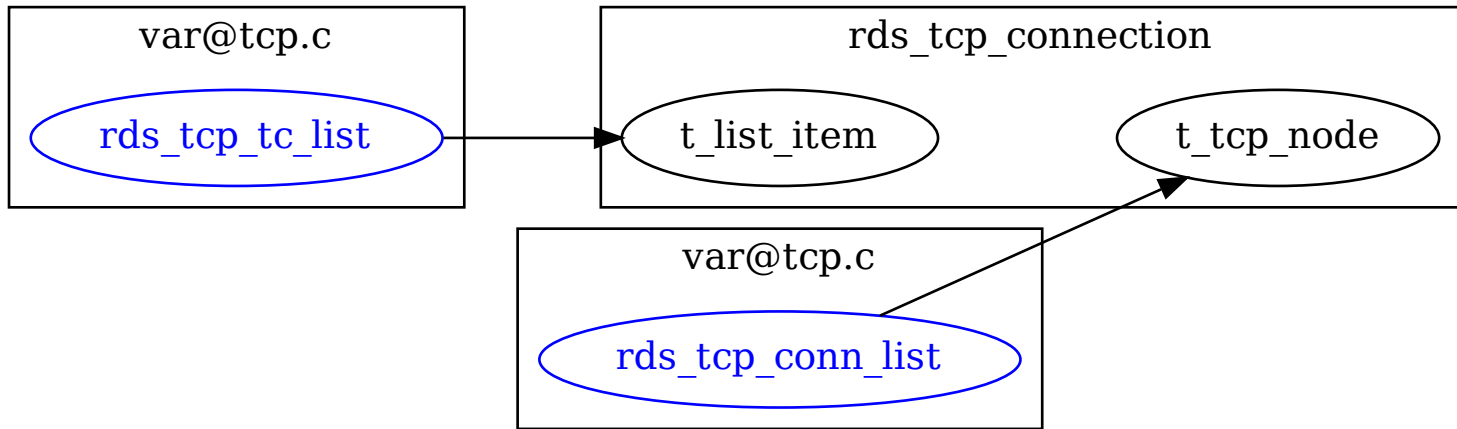


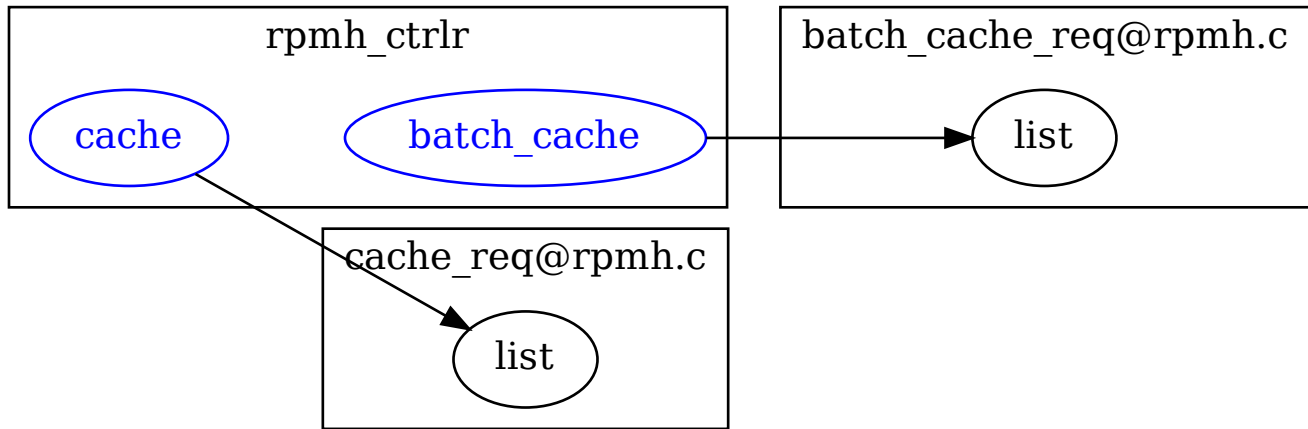


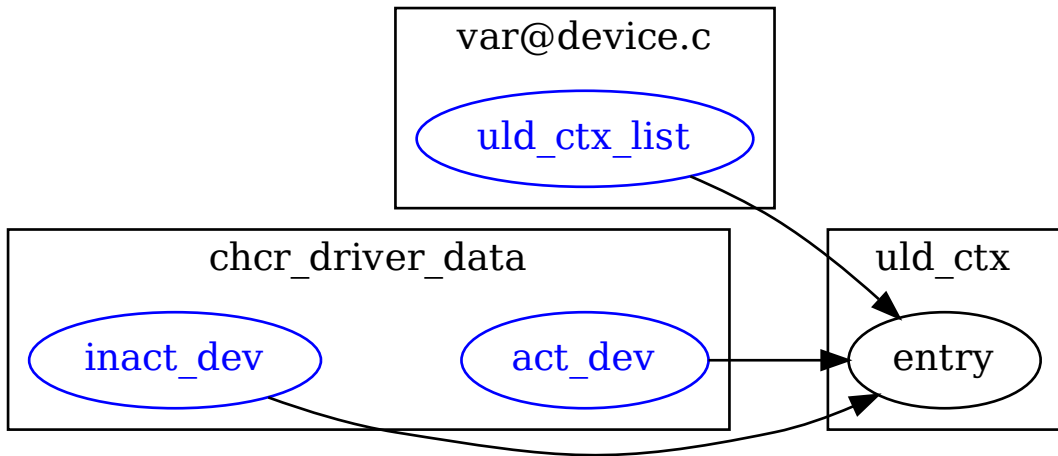


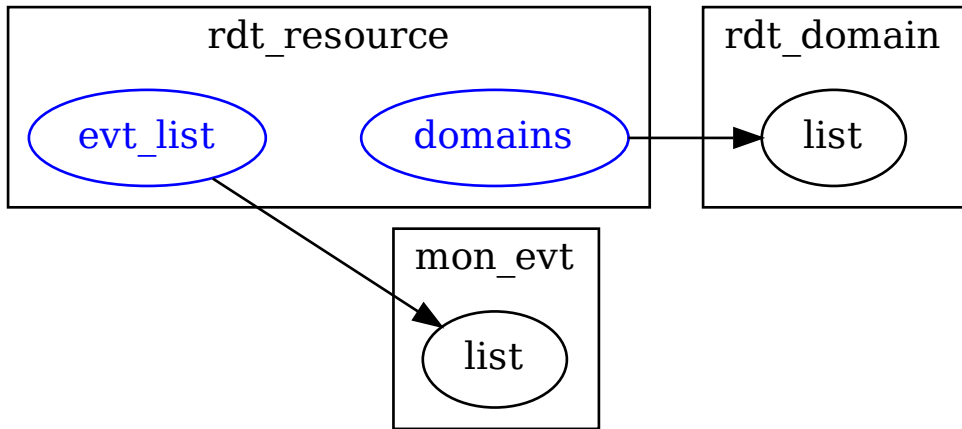


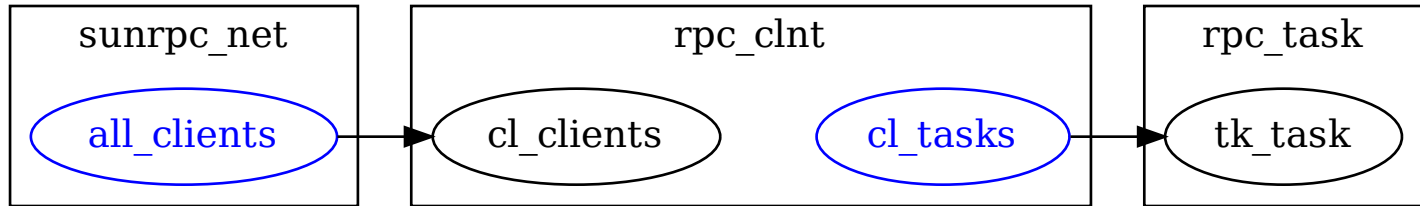


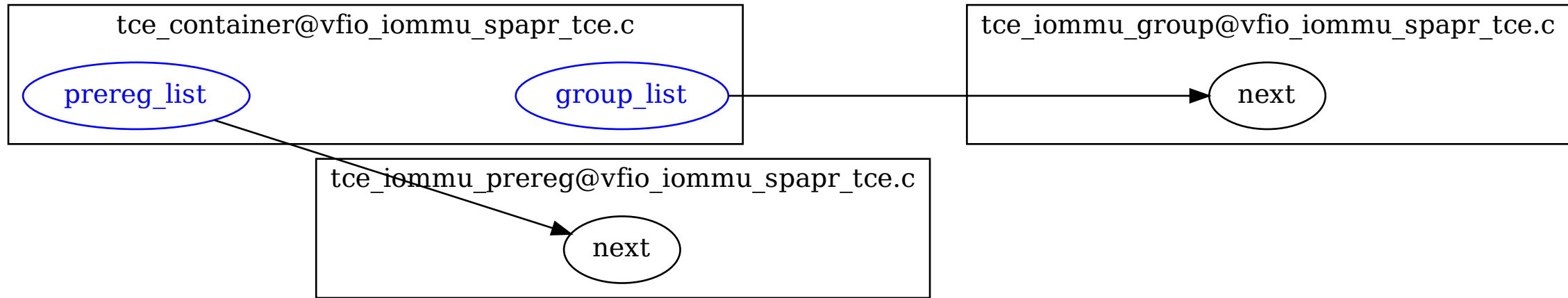


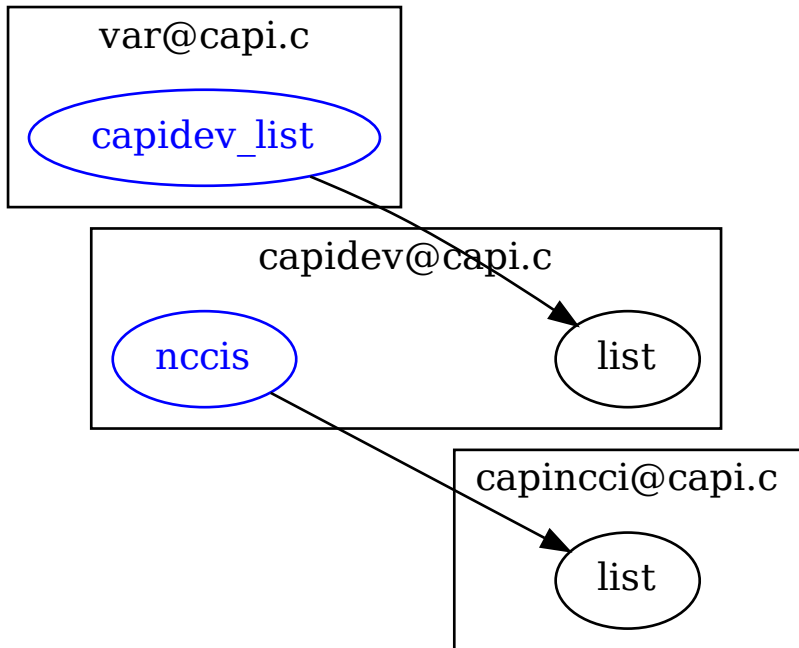


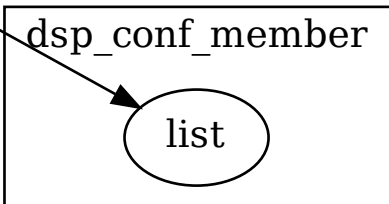
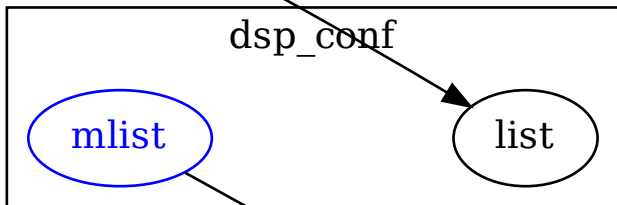
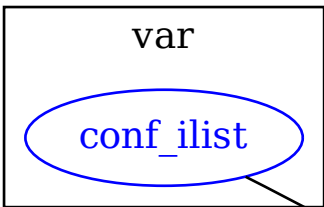


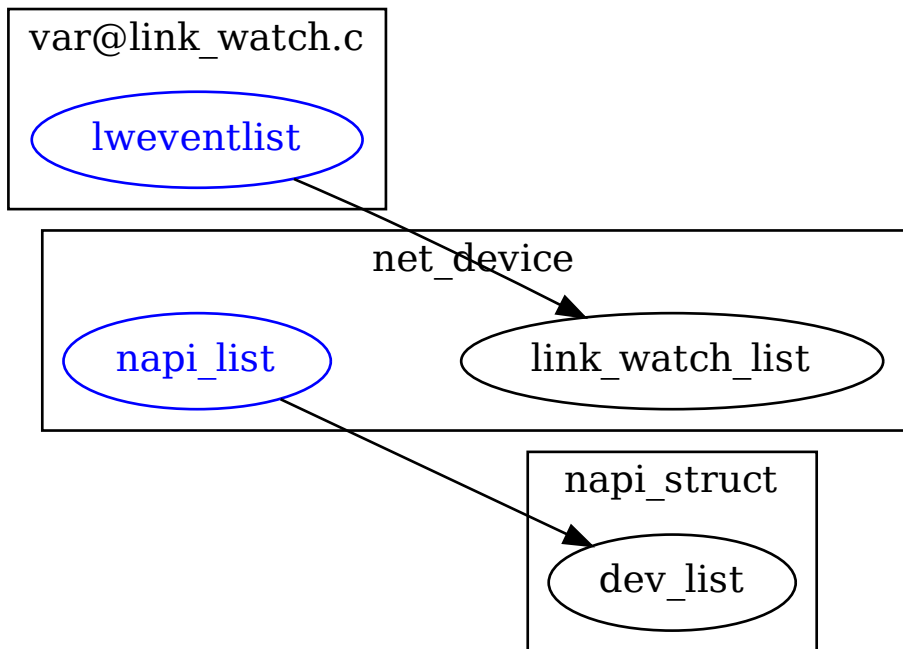


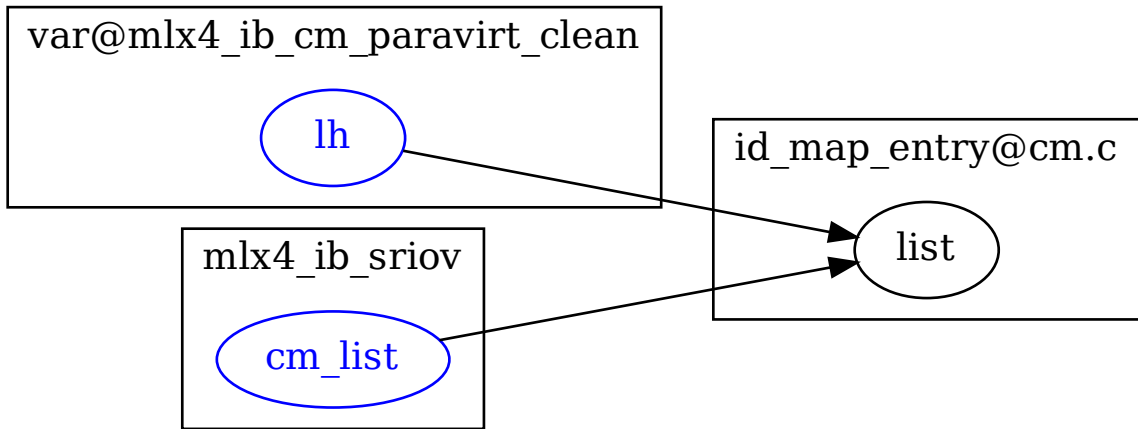


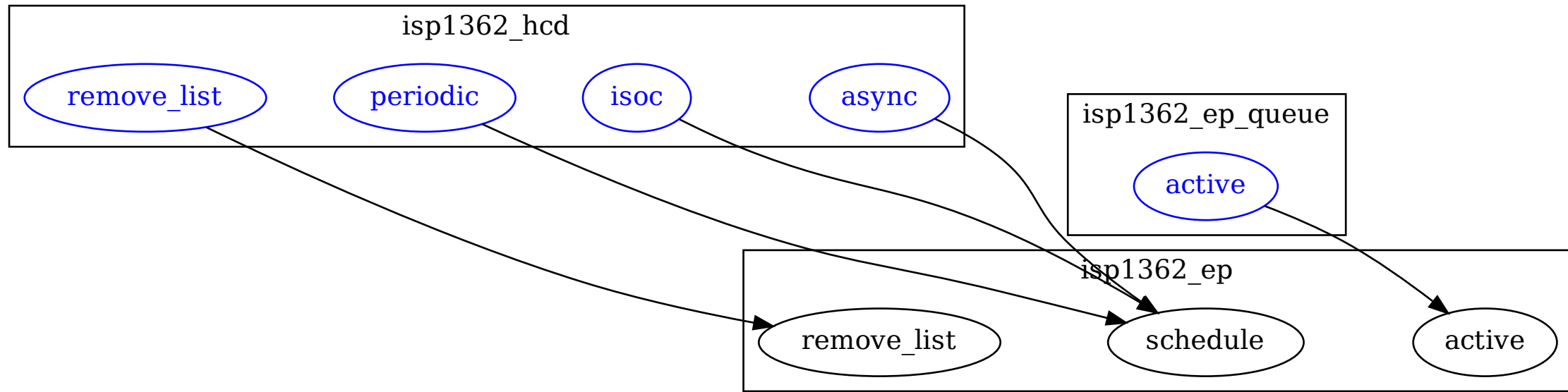


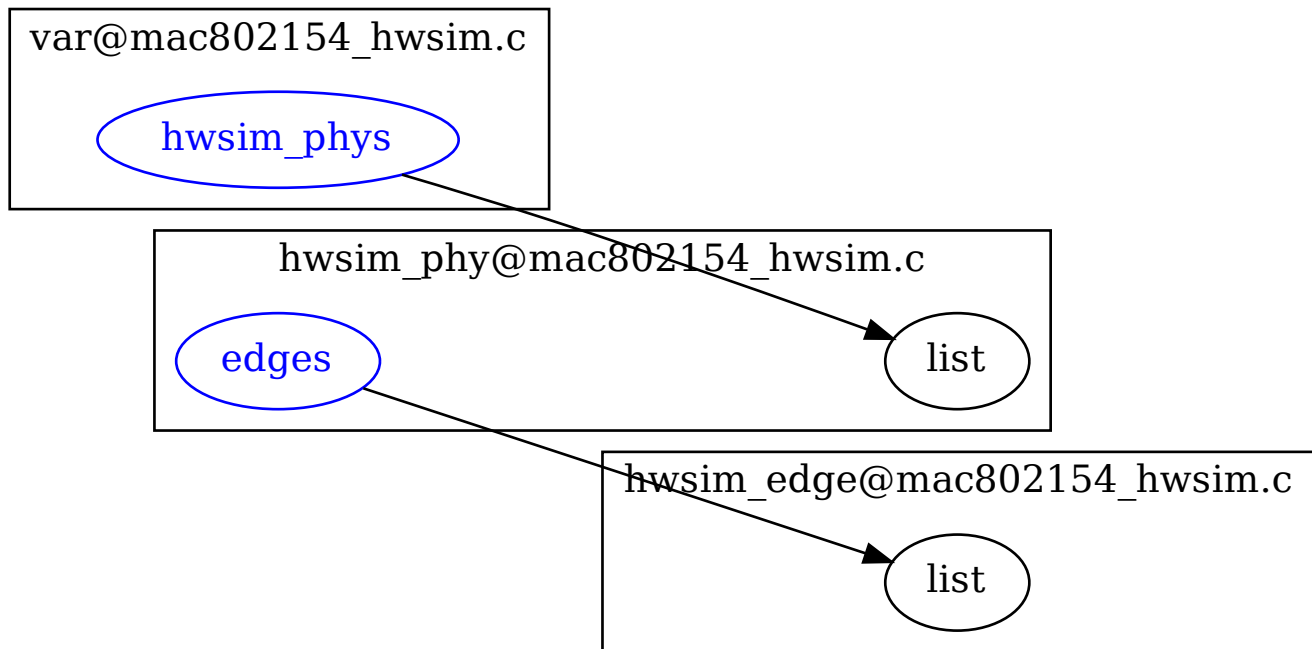


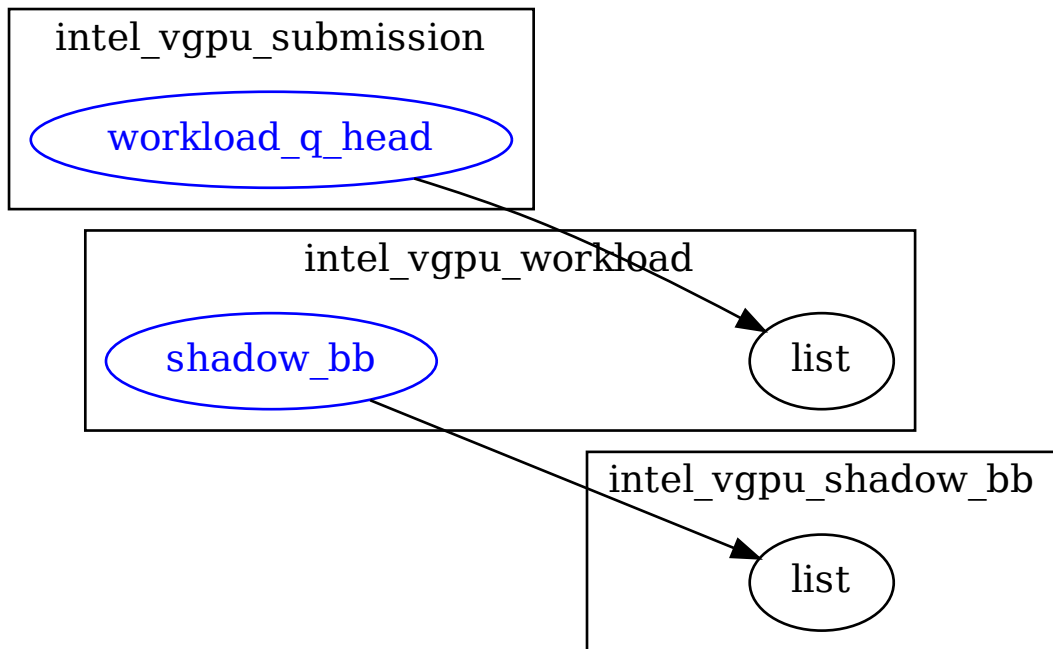


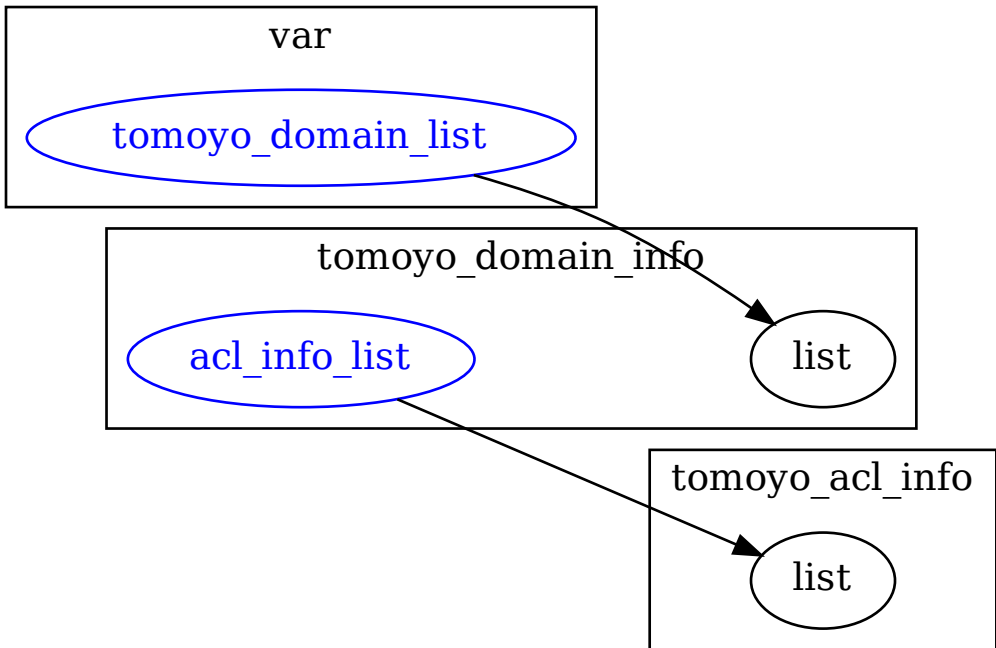


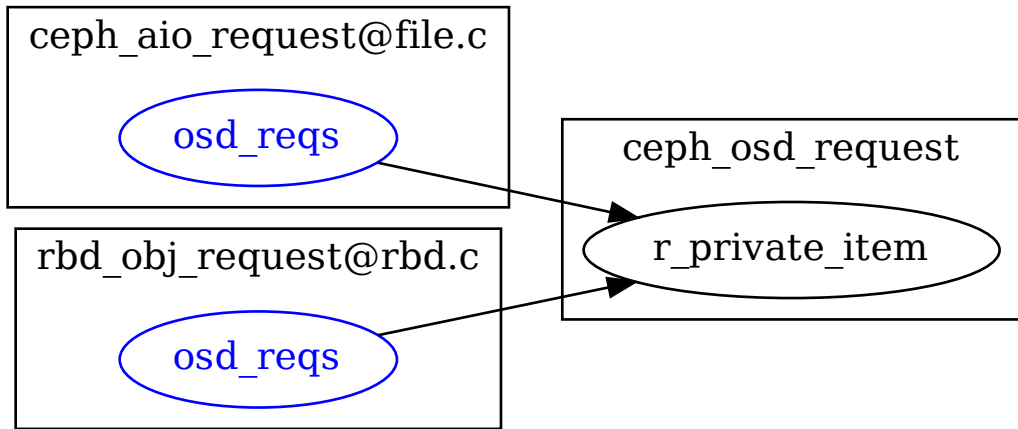


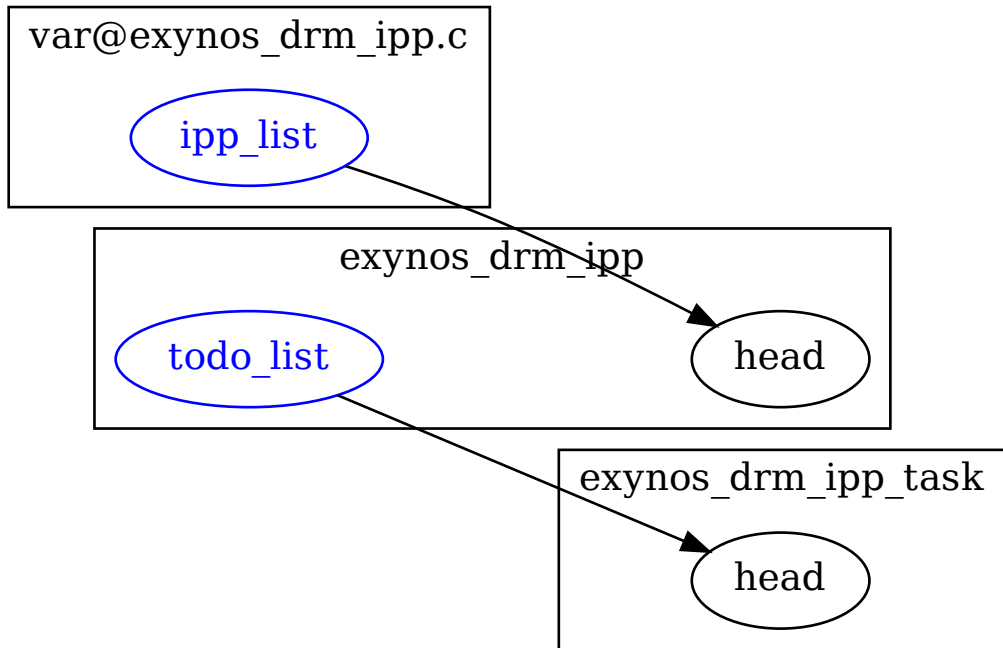


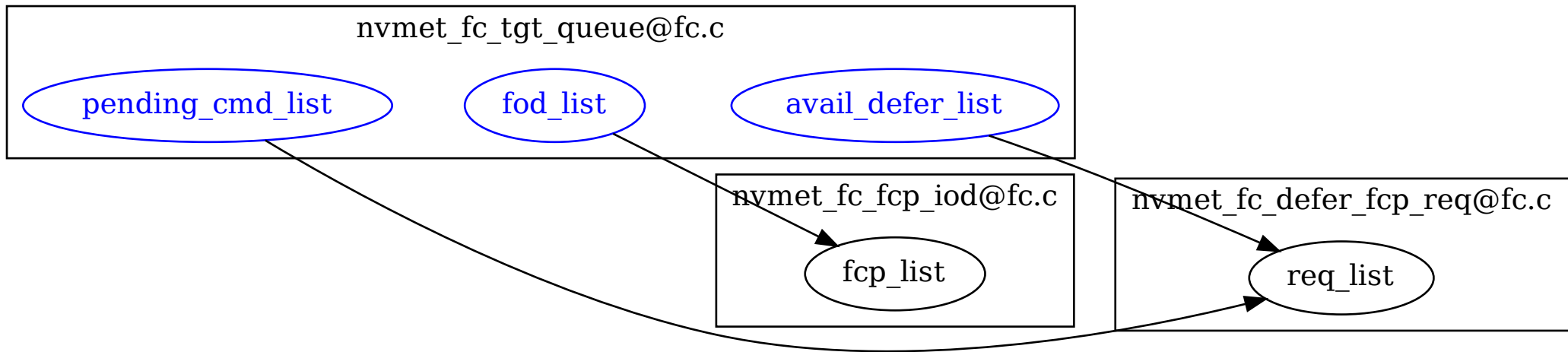


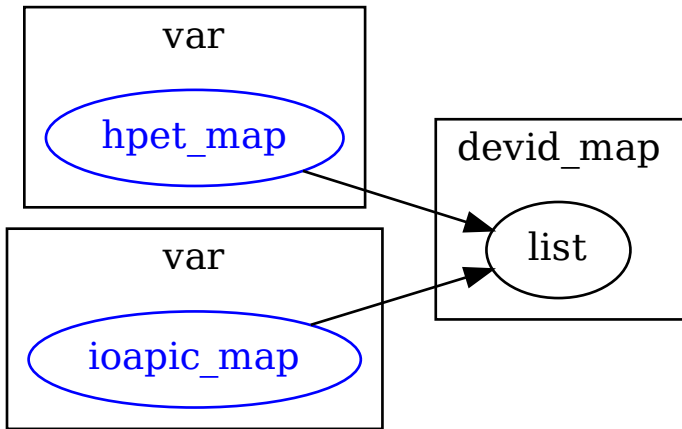


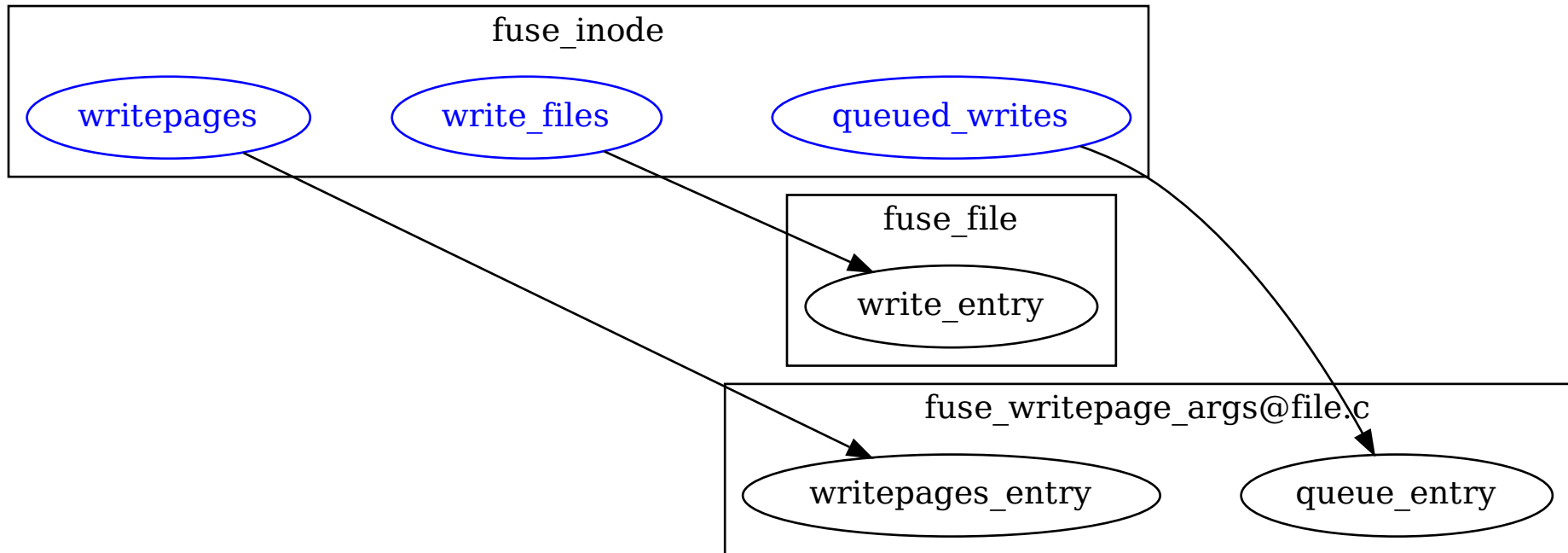


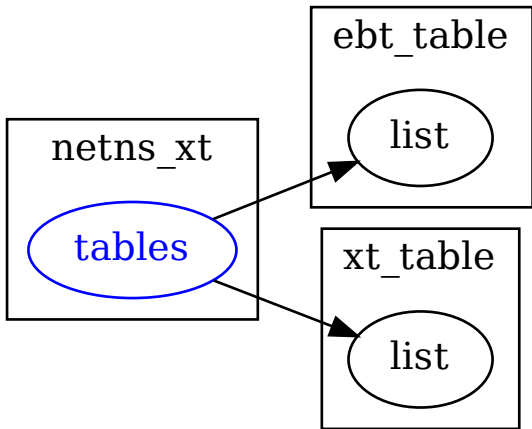


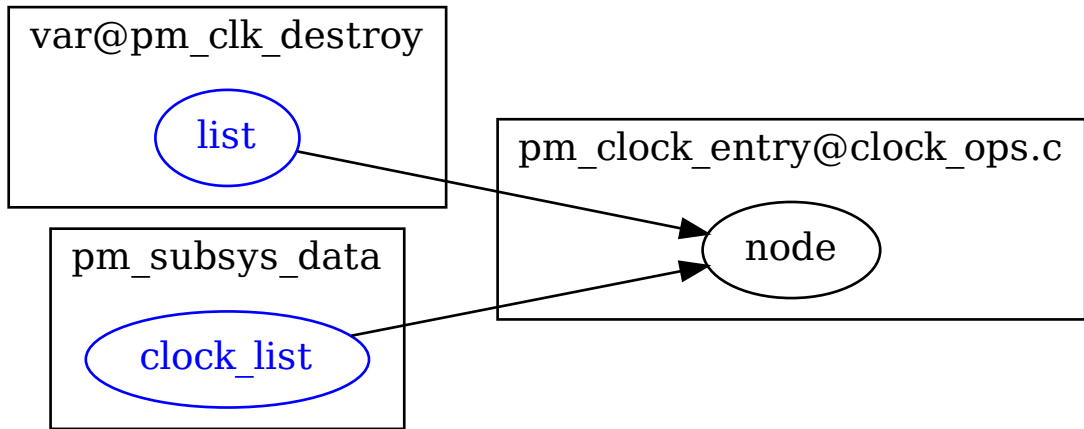


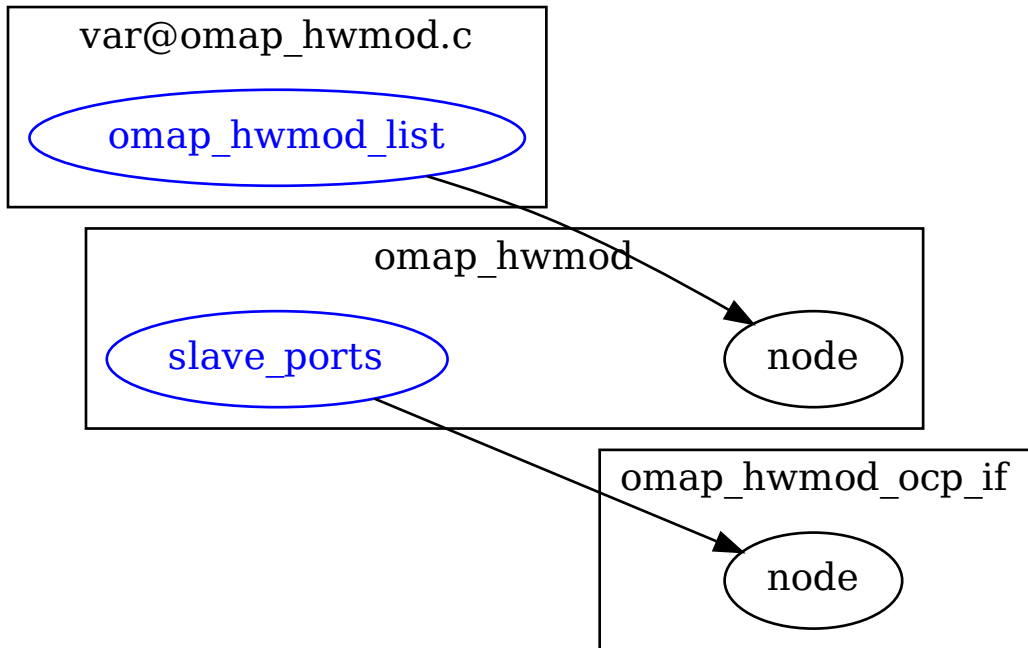


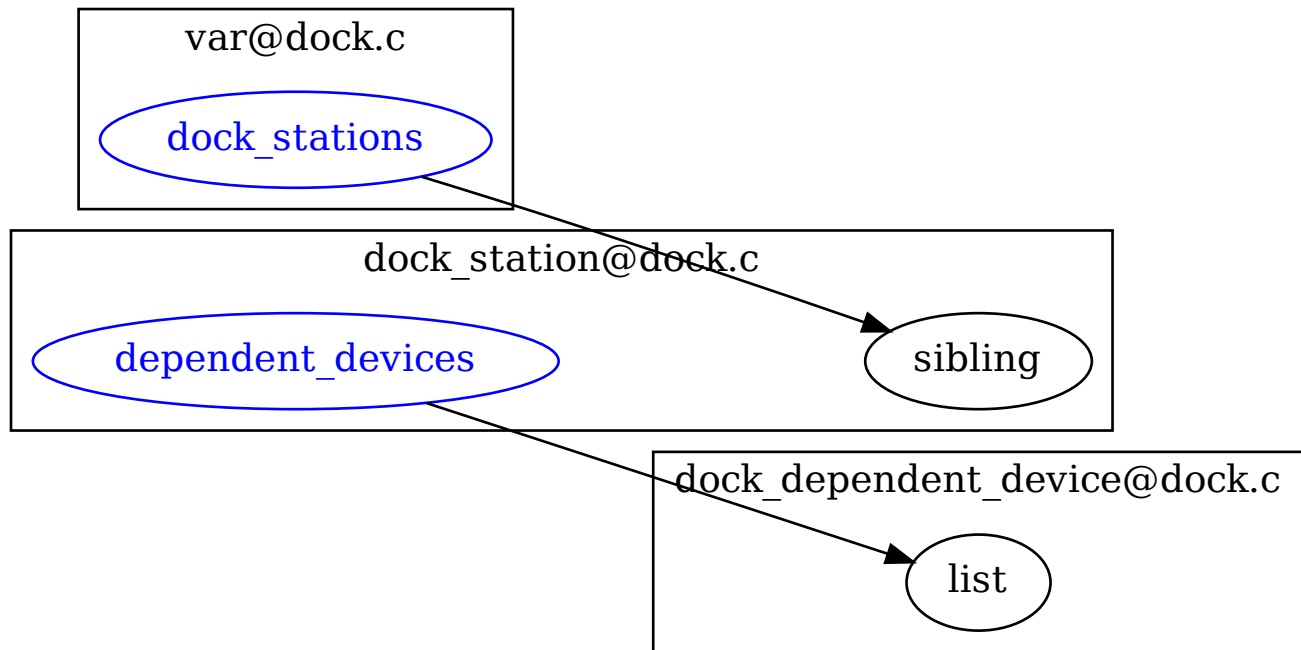


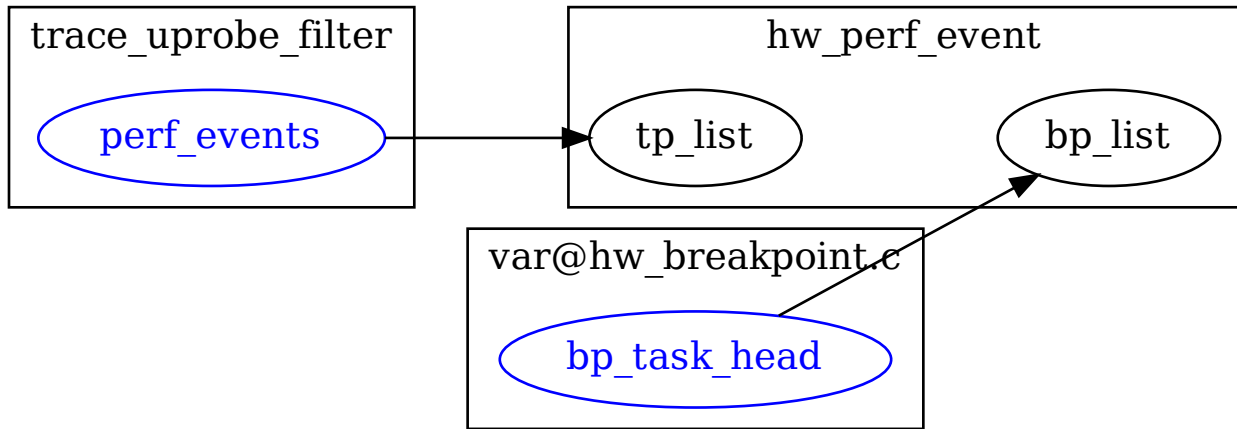


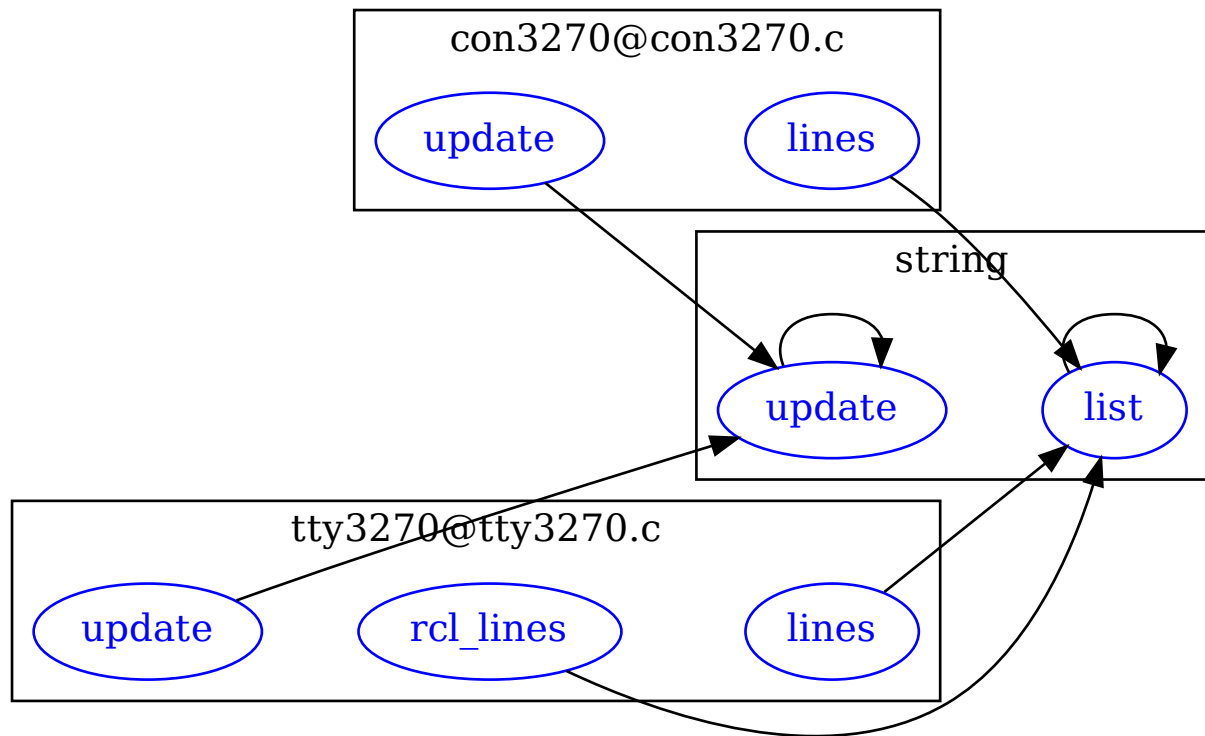


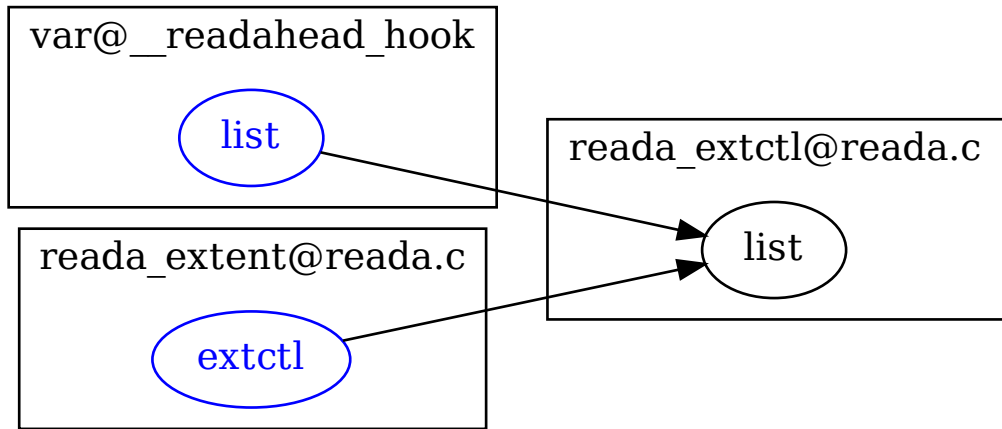


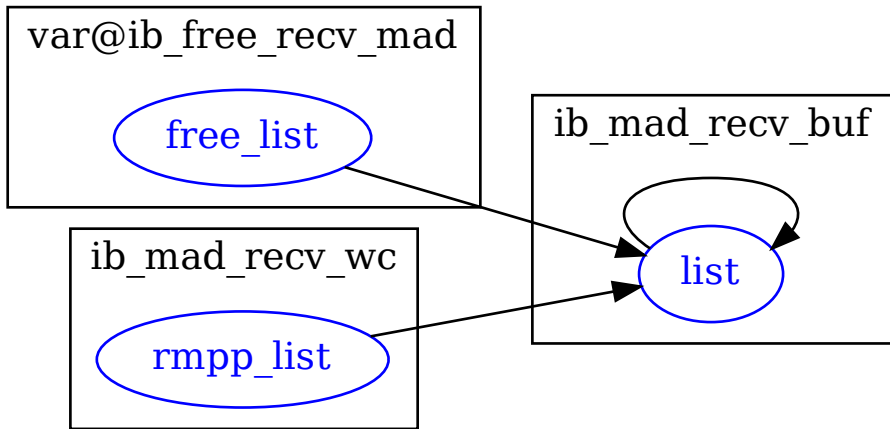


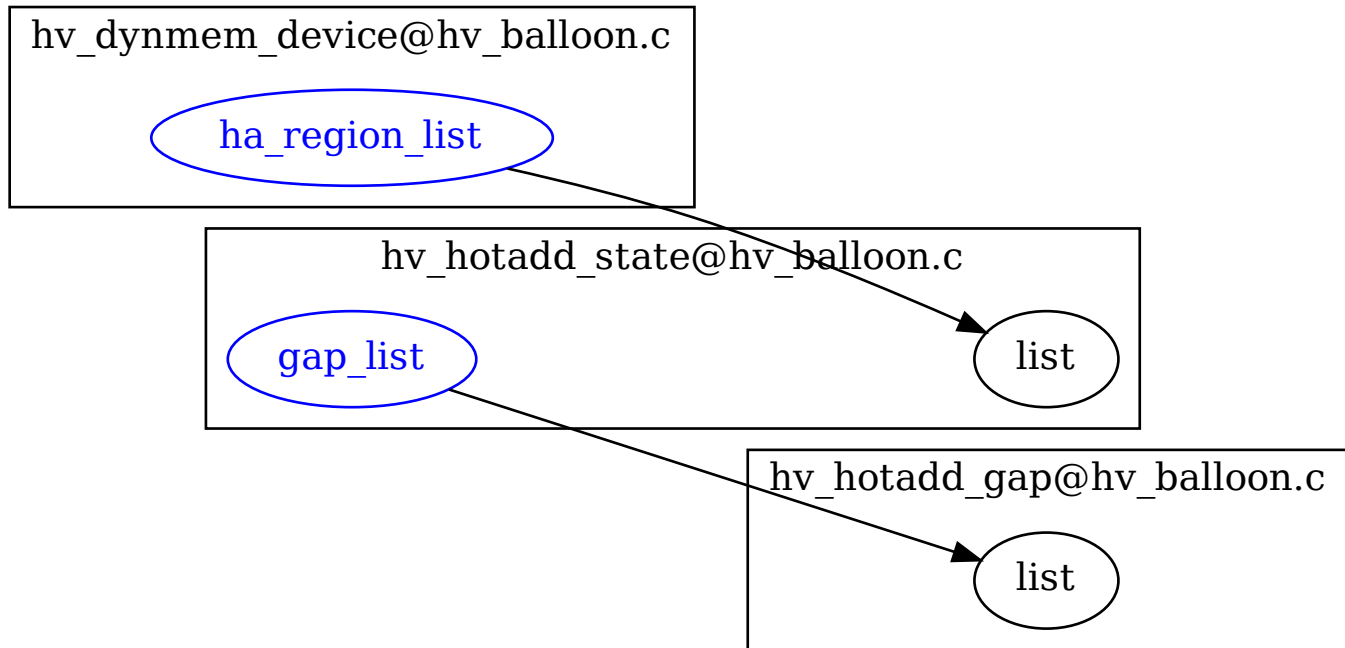


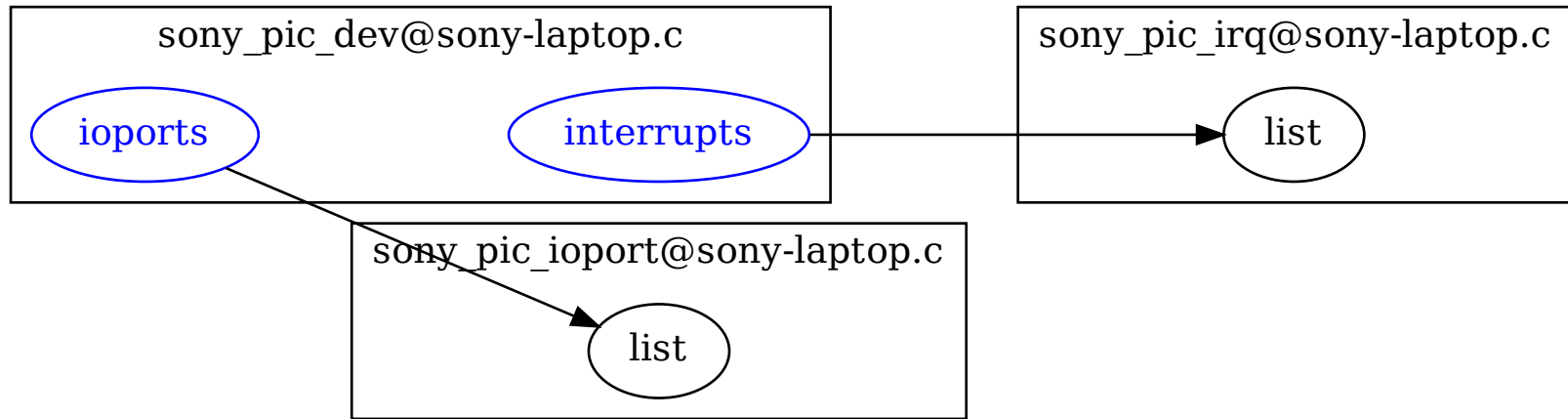


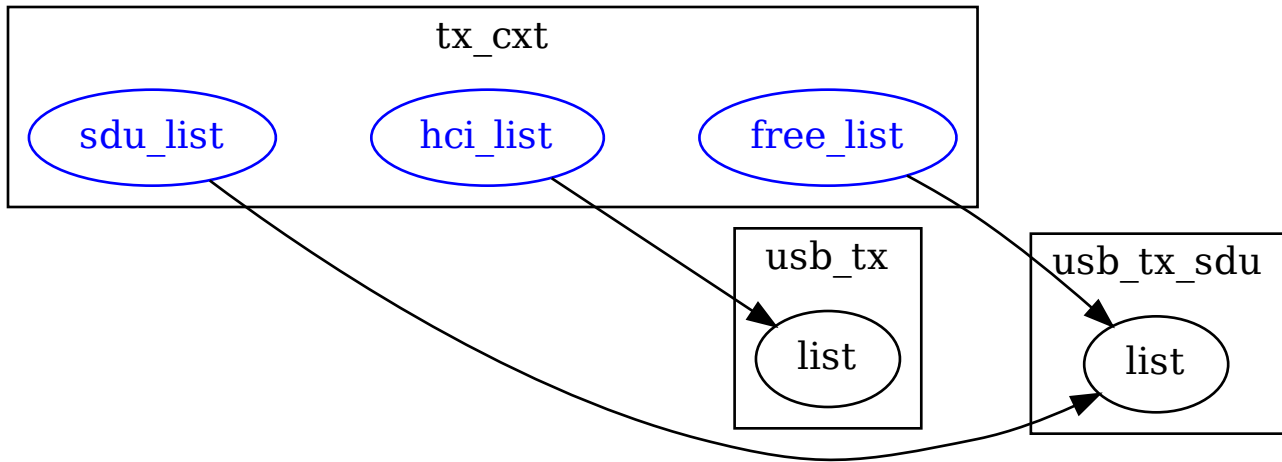


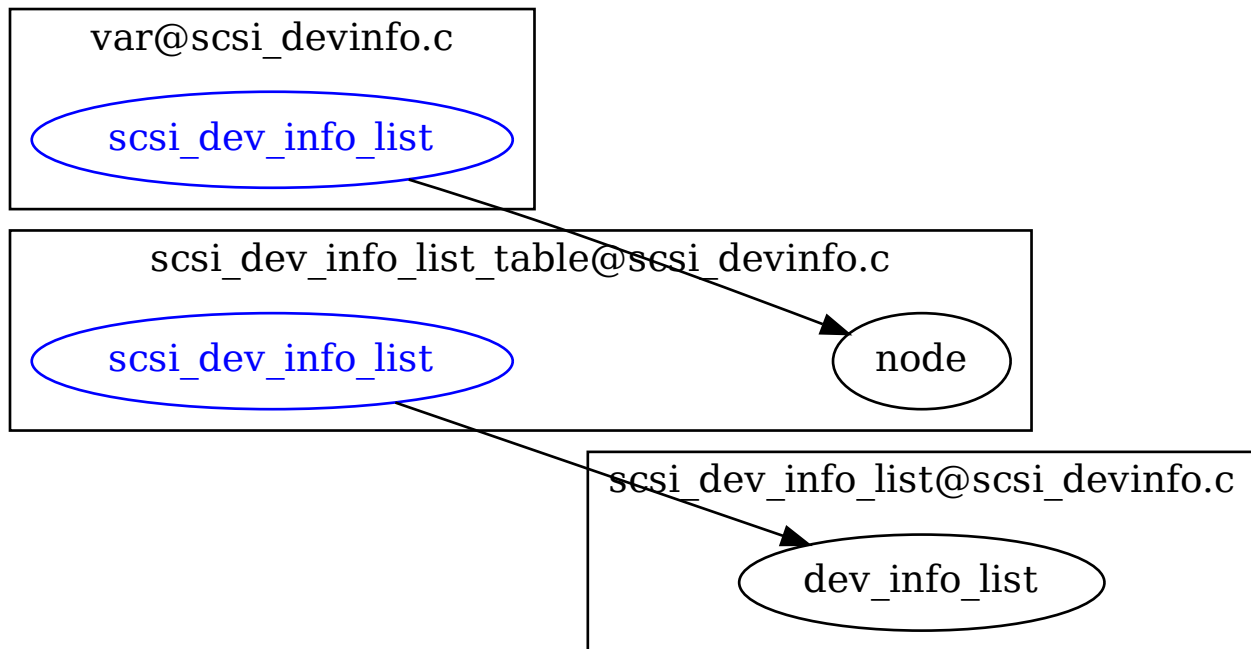


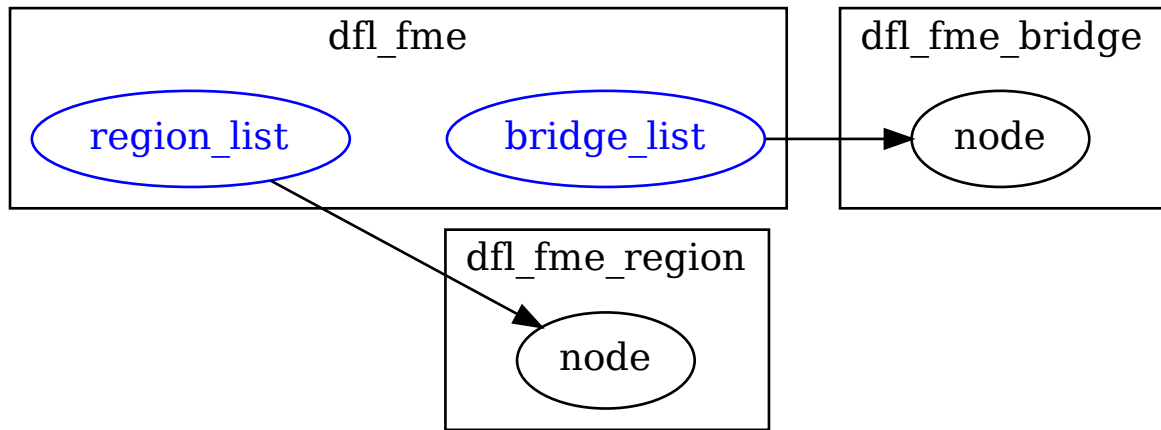


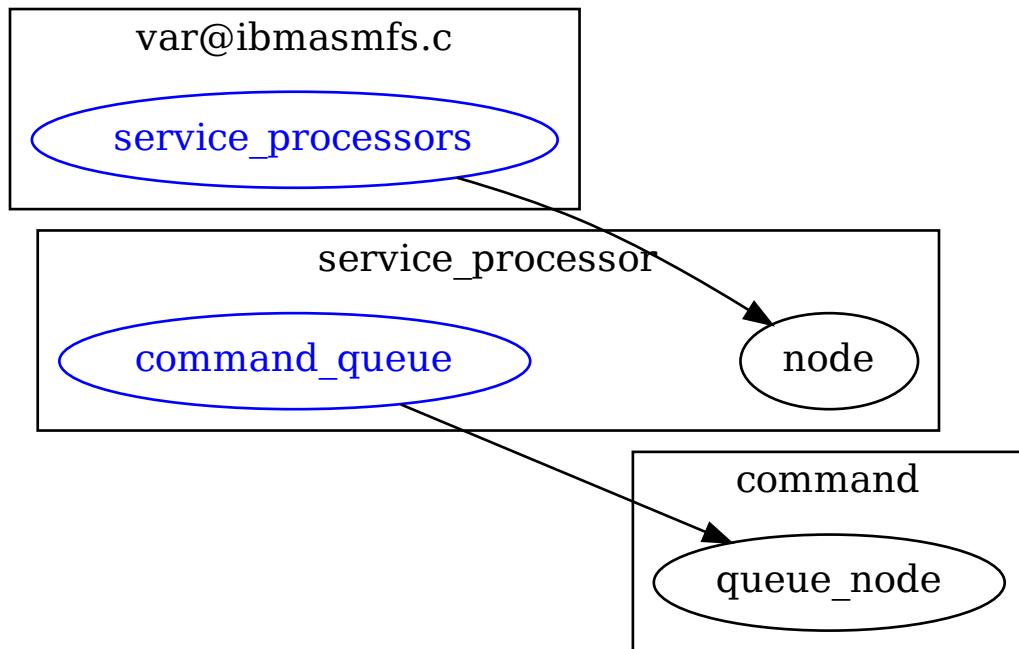


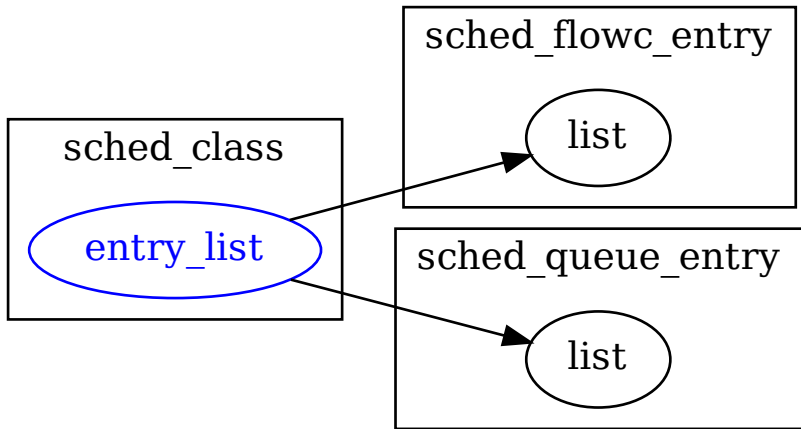


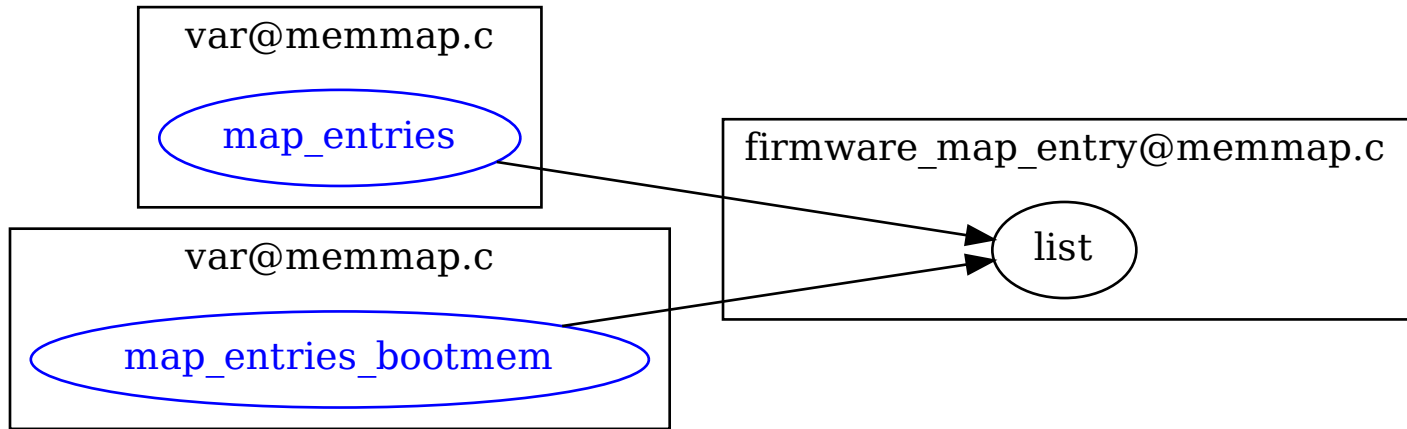


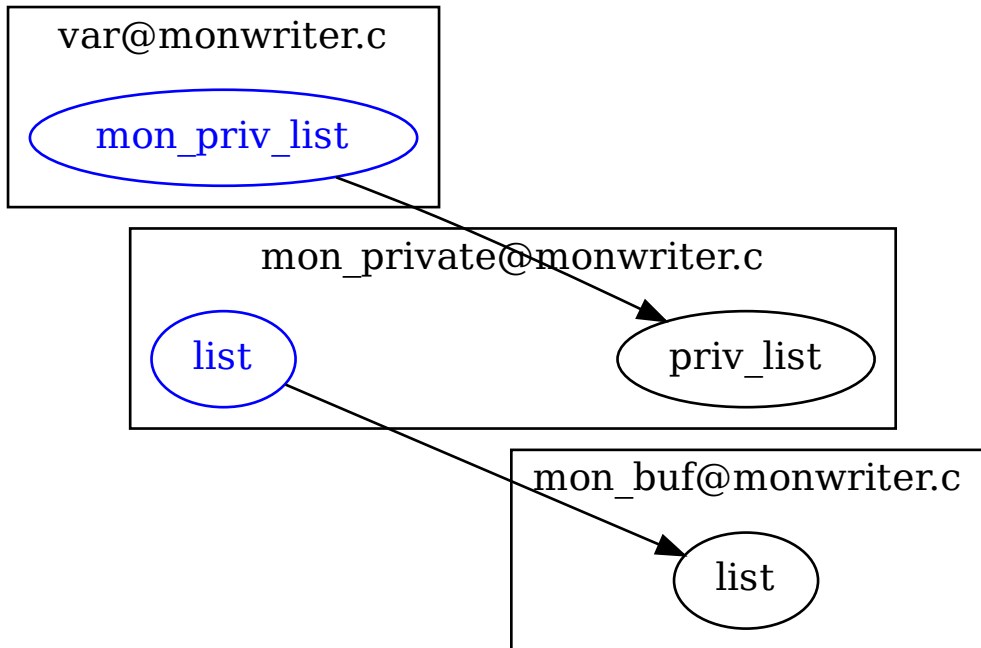


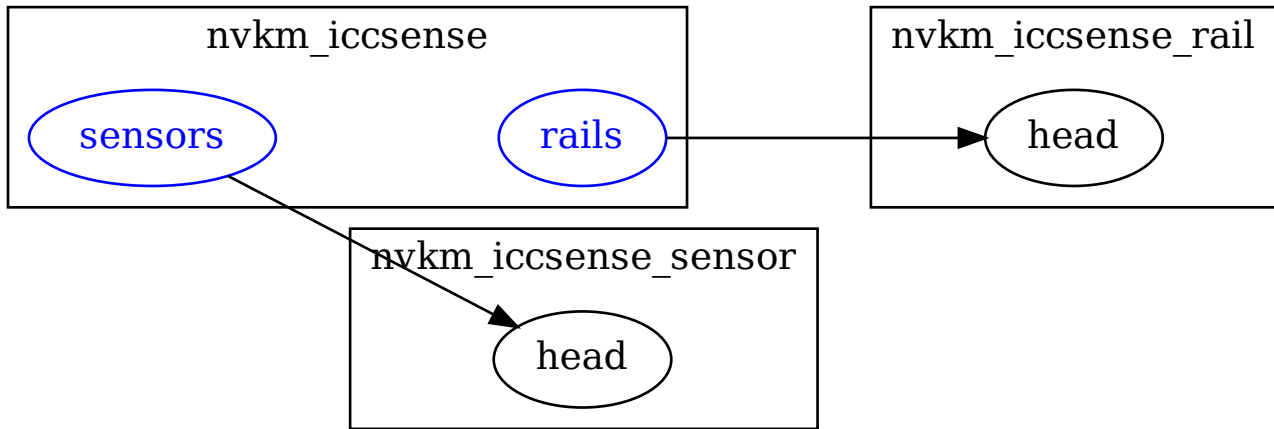


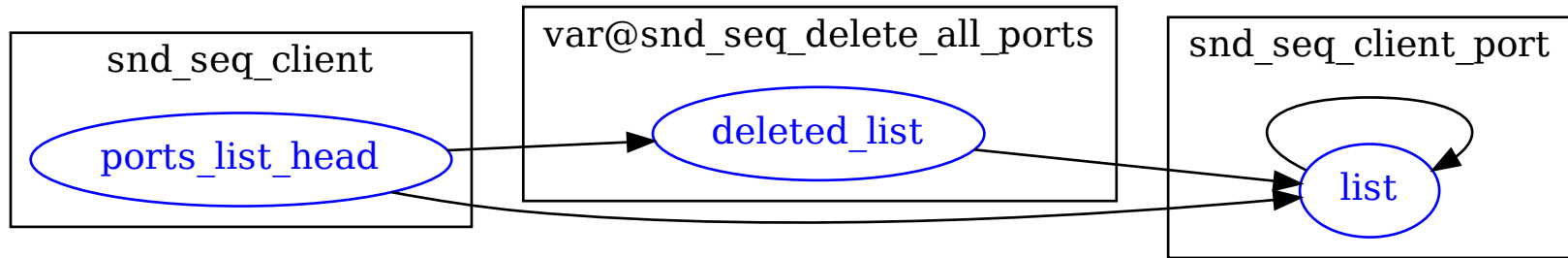


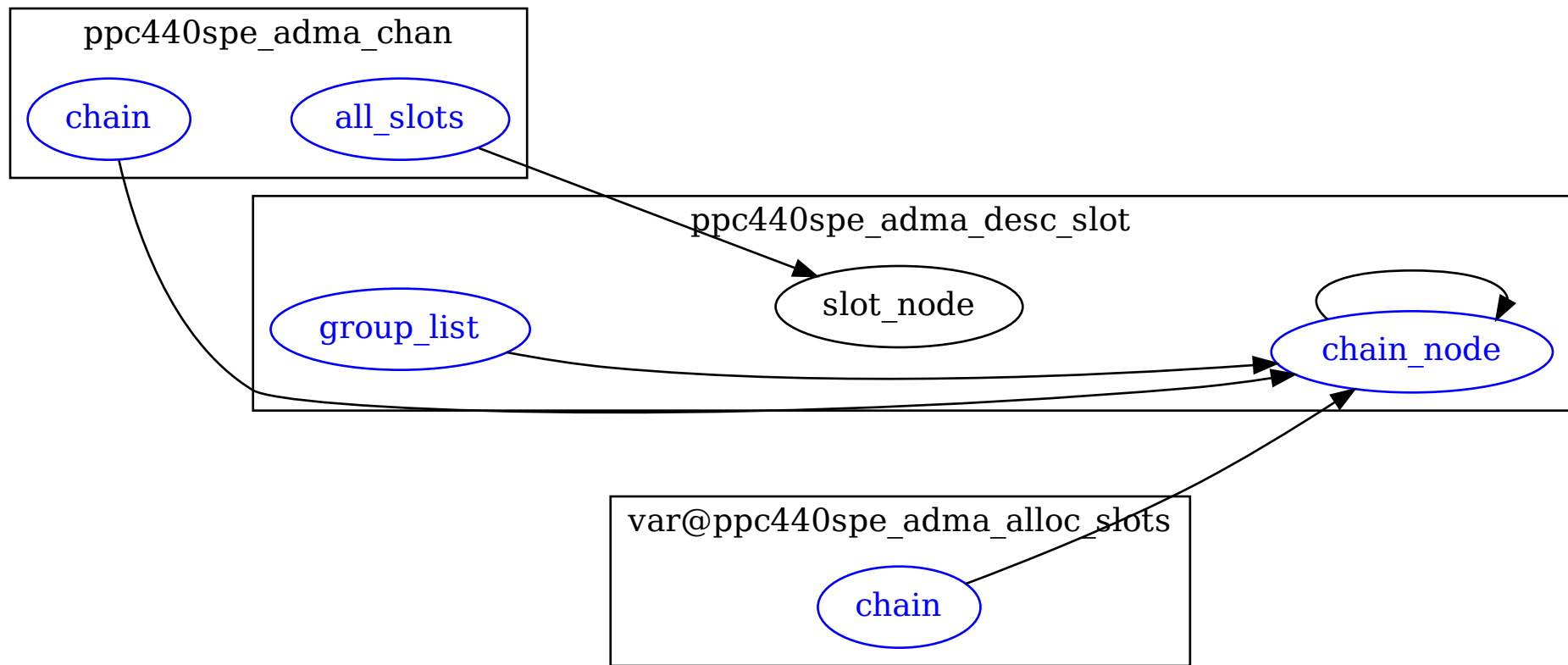


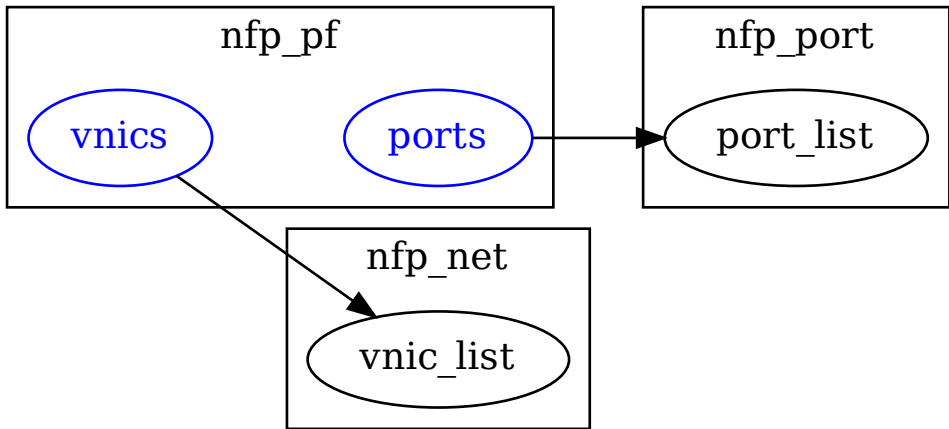


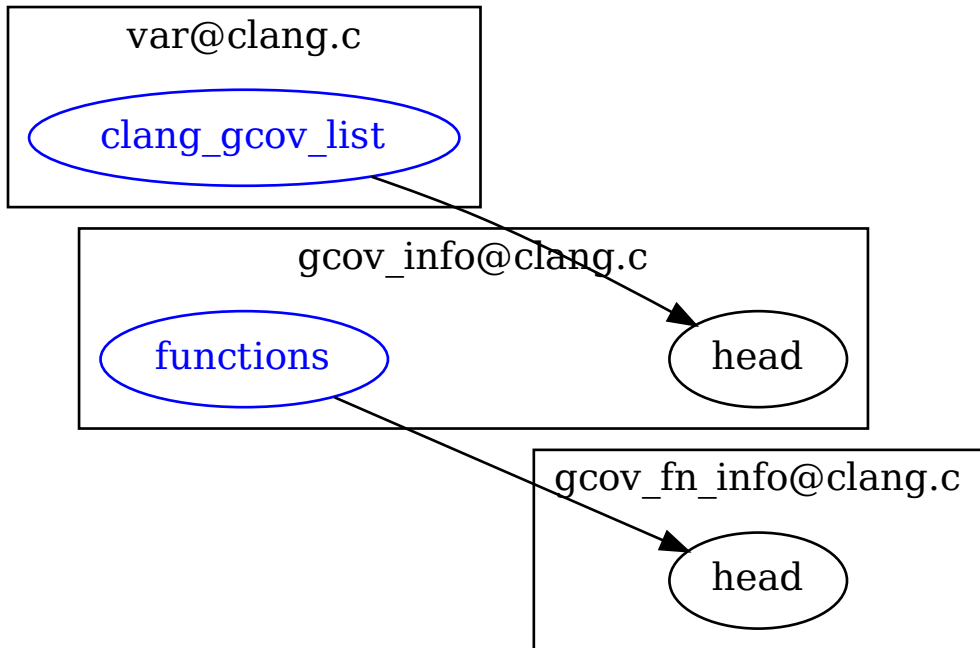


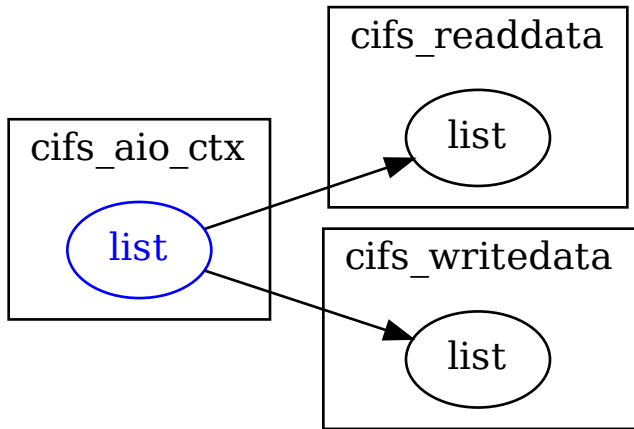


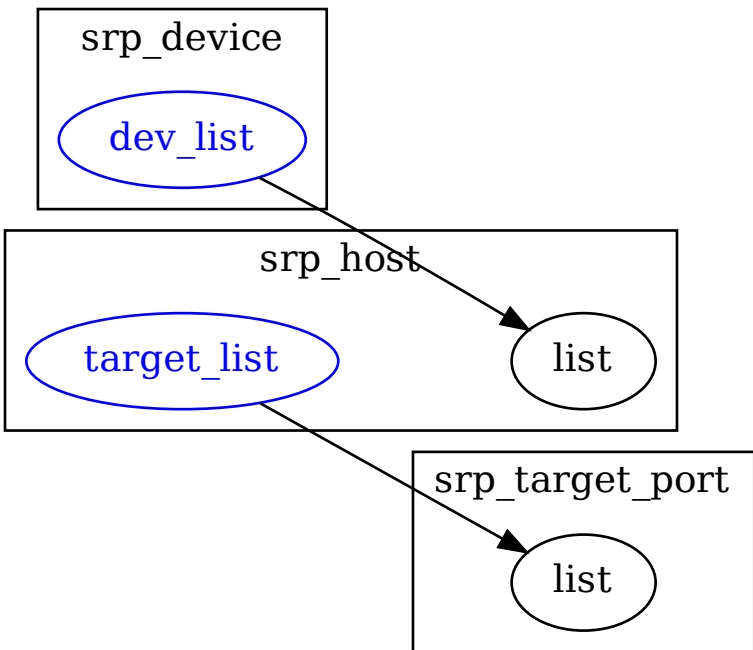


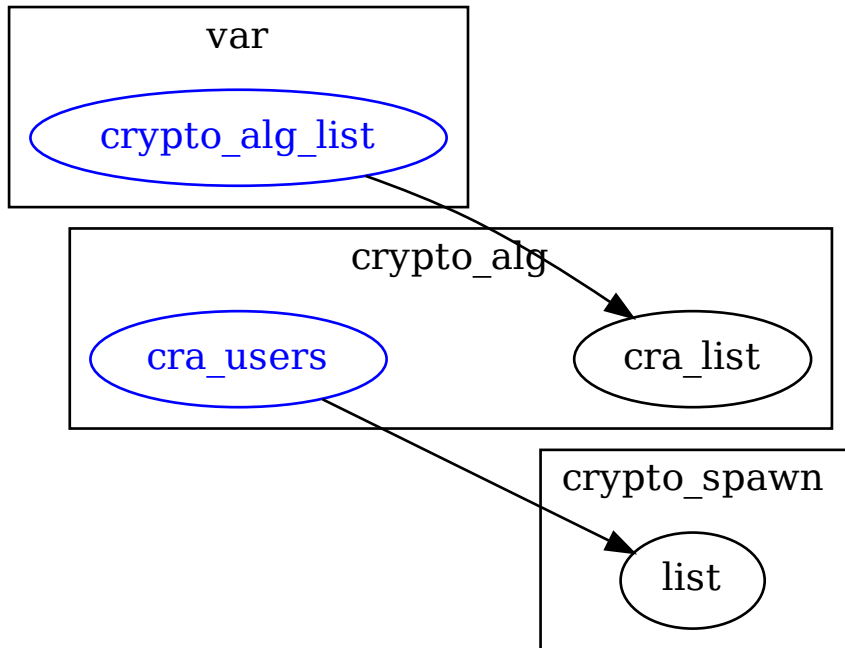


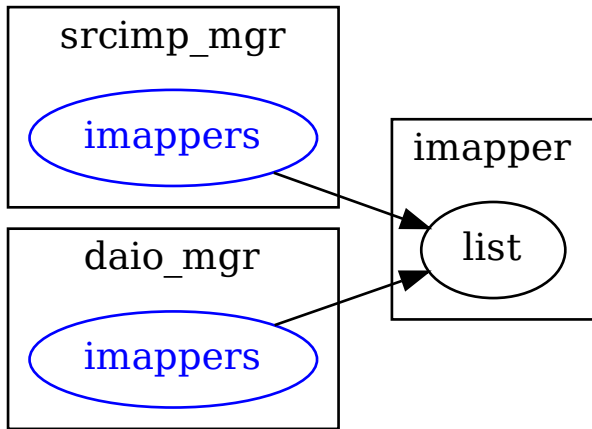












var@btrfs_batch_insert_items

head

var@btrfs_batch_delete_items

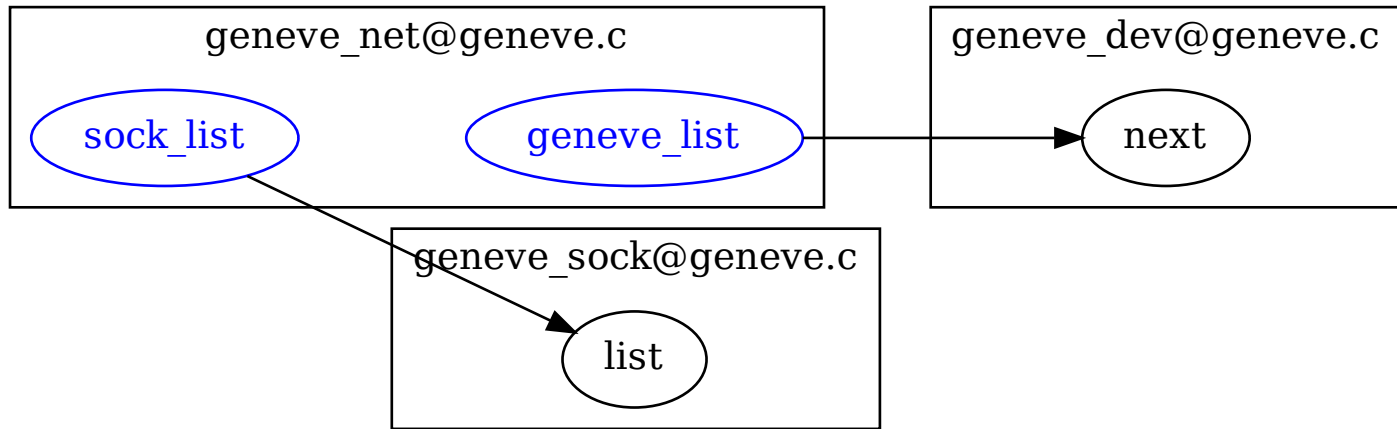
head

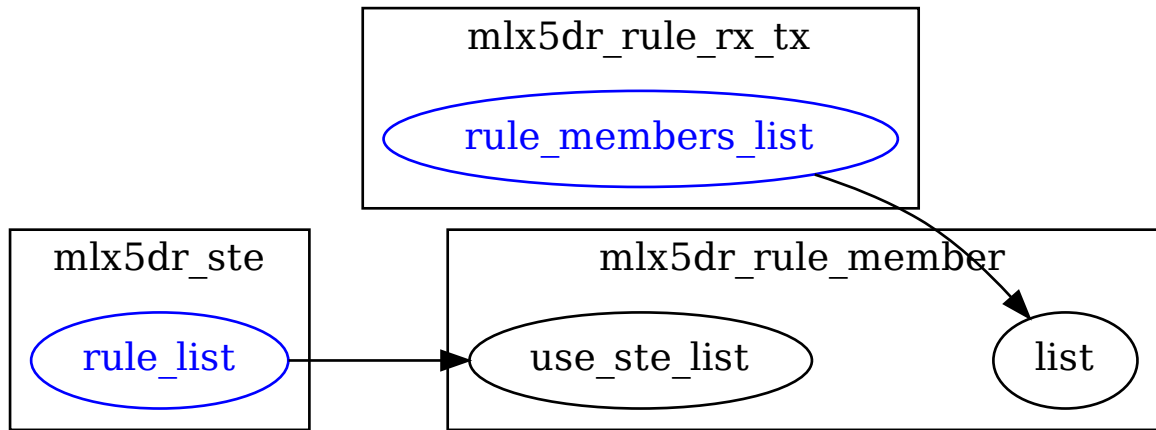
btrfs_delayed_item

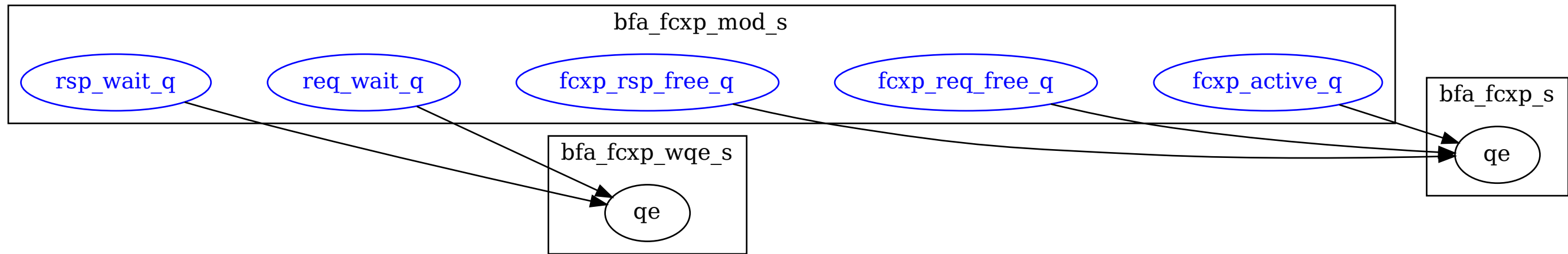
tree_list

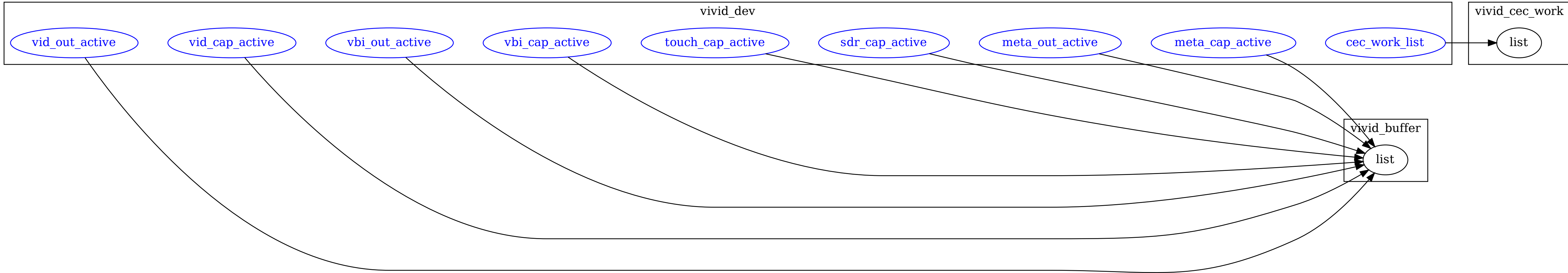
```
graph LR; A1([head]) --> B([tree_list]); A2([head]) --> B;
```

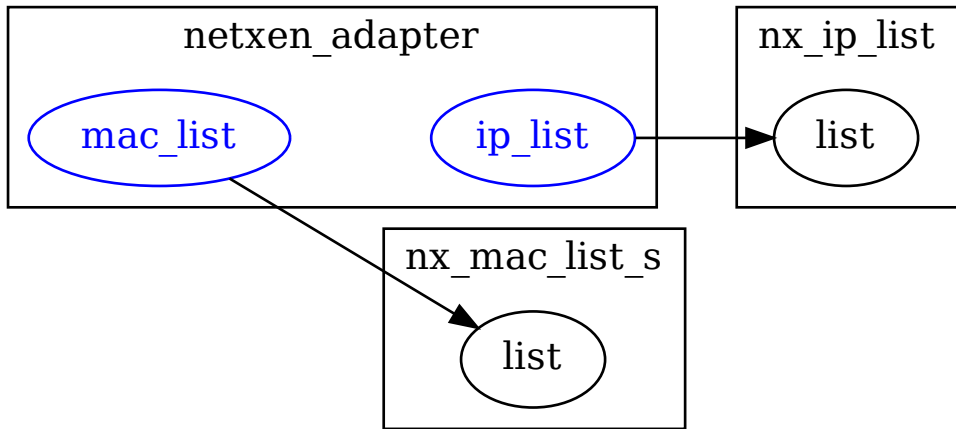
The diagram illustrates two variables, `var@btrfs_batch_insert_items` and `var@btrfs_batch_delete_items`, each containing a `head` field. Both `head` fields point to the `tree_list` field within the `btrfs_delayed_item` structure.

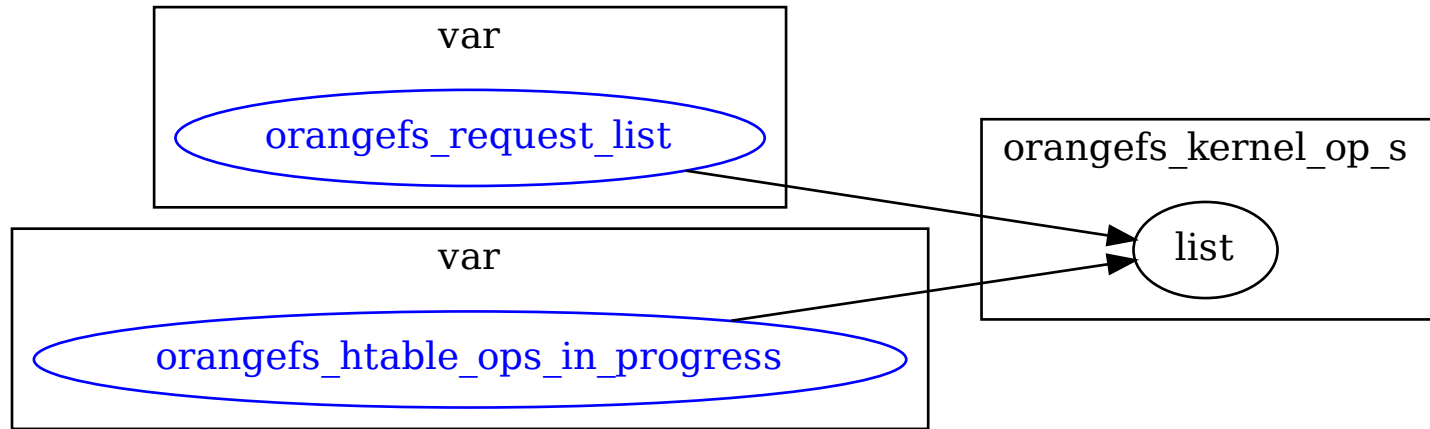


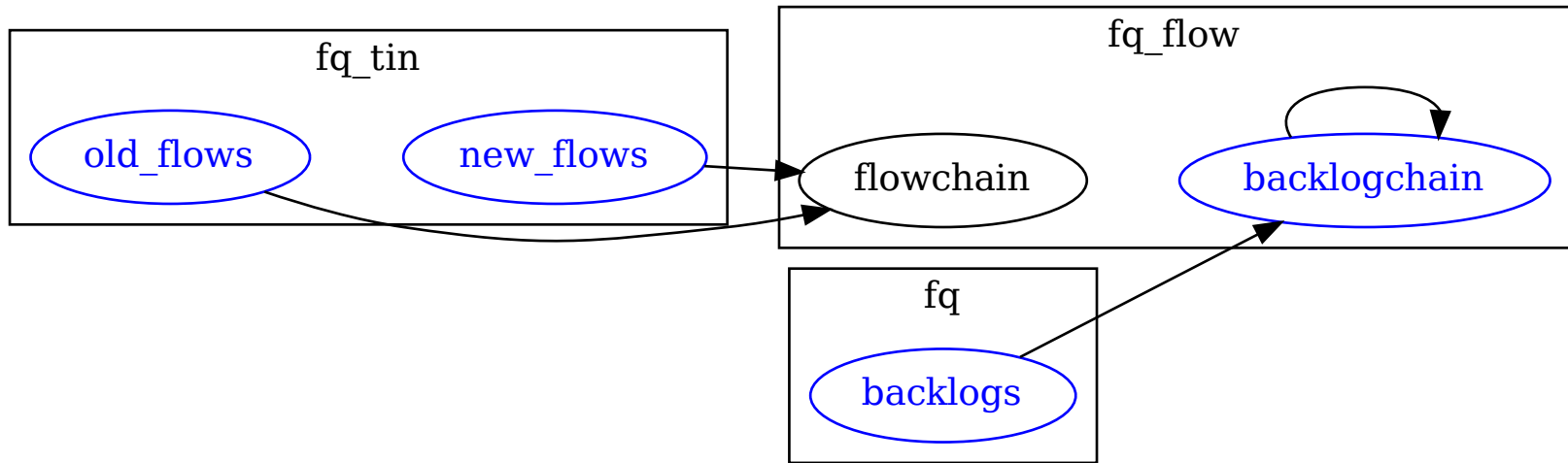


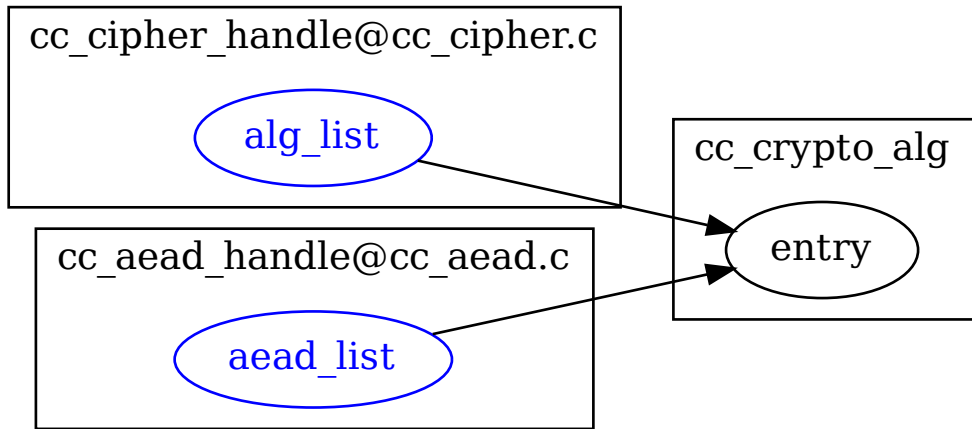


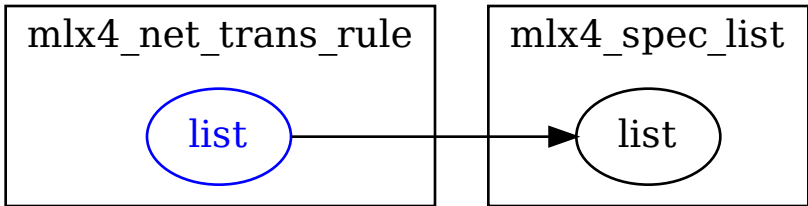


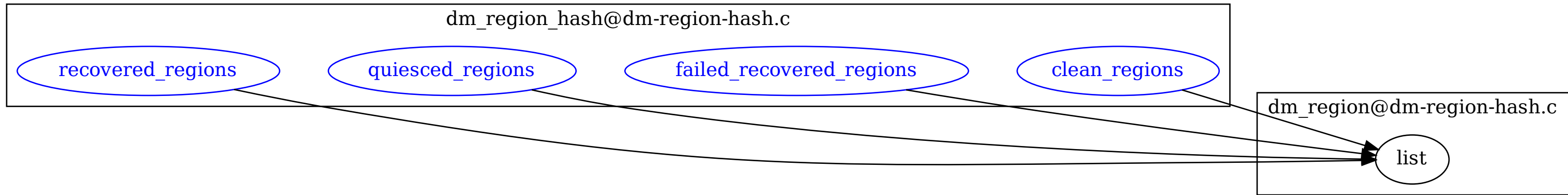












var@bq27xxx_battery.c

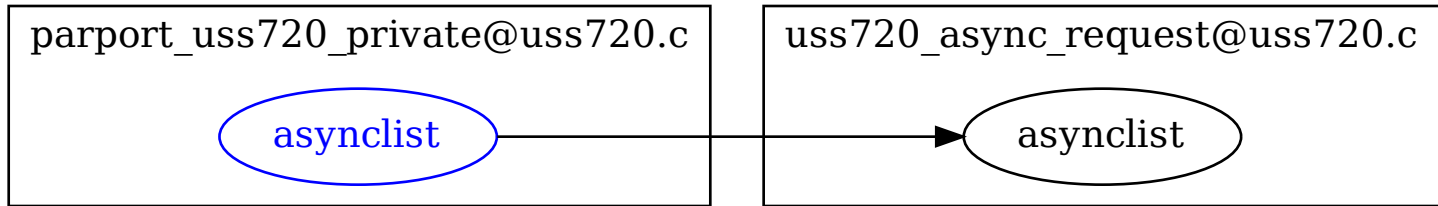
bq27xxx_battery_devices

bq27xxx_device_info

list

```
graph LR; A["var@bq27xxx_battery.c  
bq27xxx_battery_devices"] --> B["bq27xxx_device_info  
list"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@bq27xxx_battery.c' at the top and 'bq27xxx_battery_devices' at the bottom. The text 'bq27xxx_battery_devices' is enclosed in a blue oval. The right box contains the text 'bq27xxx_device_info' at the top and 'list' at the bottom. The text 'list' is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.



var@sysfs.c

acpi_table_attr_list

acpi_table_attr@sysfs.c

node

```
graph LR; subgraph var_sysfs_c [var@sysfs.c]; attr_list([acpi_table_attr_list]); end; subgraph acpi_table_attr_sysfs_c [acpi_table_attr@sysfs.c]; node([node]); end; attr_list --> node;
```

The diagram illustrates a pointer relationship between two variables in the sysfs filesystem. On the left, a box labeled 'var@sysfs.c' contains an oval labeled 'acpi_table_attr_list'. On the right, a box labeled 'acpi_table_attr@sysfs.c' contains an oval labeled 'node'. A horizontal arrow points from the 'acpi_table_attr_list' oval to the 'node' oval, indicating that the 'acpi_table_attr_list' variable points to the 'node' object.

acpi_ged_device@evged.c

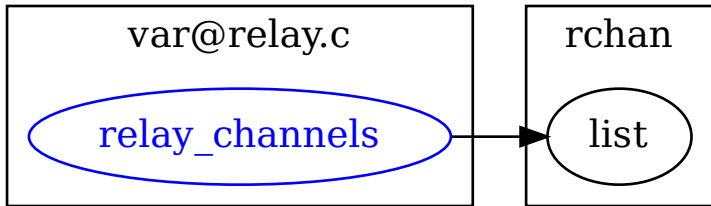
event_list

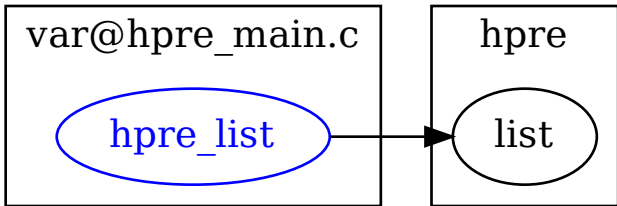
acpi_ged_event@evged.c

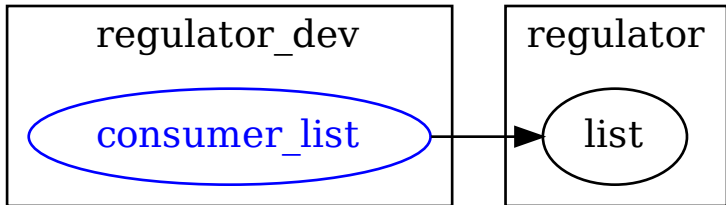
node

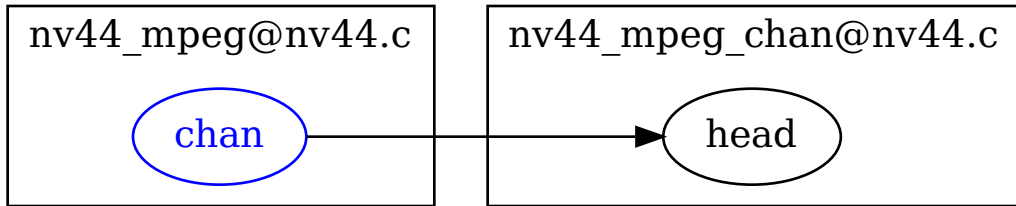
```
graph LR; subgraph acpi_ged_device [acpi_ged_device@evged.c]; event_list; end; subgraph acpi_ged_event [acpi_ged_event@evged.c]; node; end; event_list --> node;
```

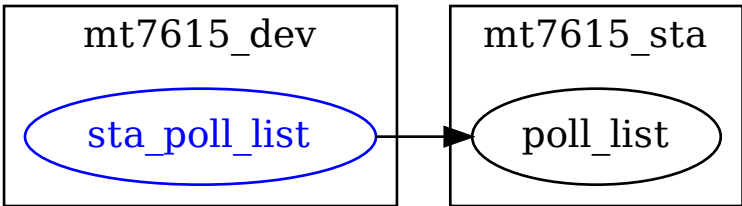
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'acpi_ged_device@evged.c' contains an oval labeled 'event_list' with a blue border. On the right, a rectangular box labeled 'acpi_ged_event@evged.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'event_list' oval to the left side of the 'node' oval, indicating that 'event_list' points to 'node'.

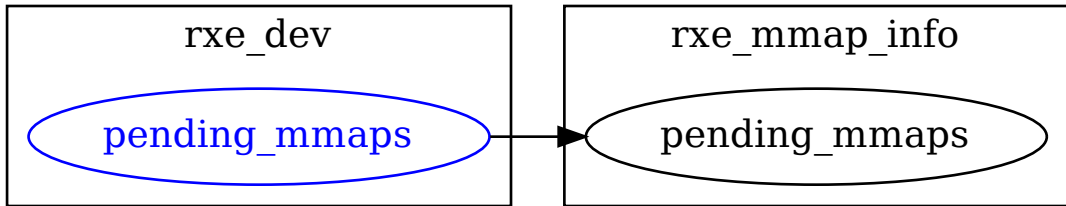












var@cpufreq_cooling.c

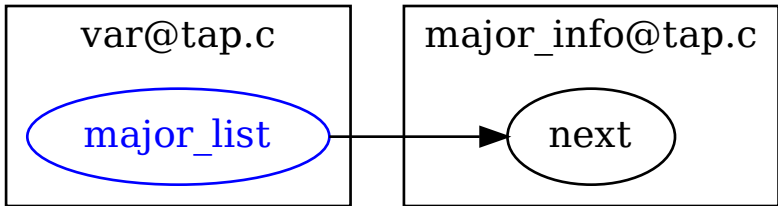
cpufreq_cdev_list

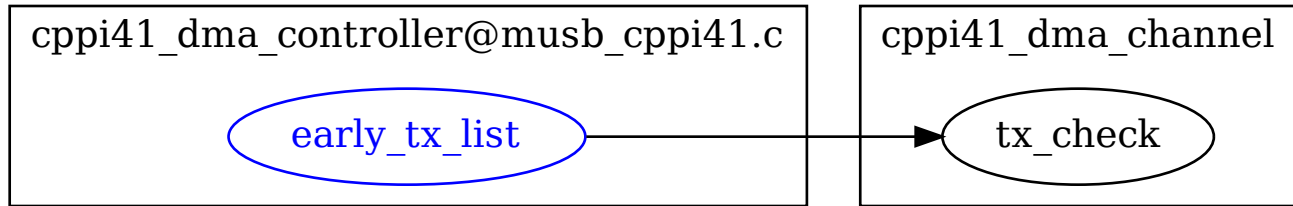
cpufreq_cooling_device@cpufreq_cooling.c

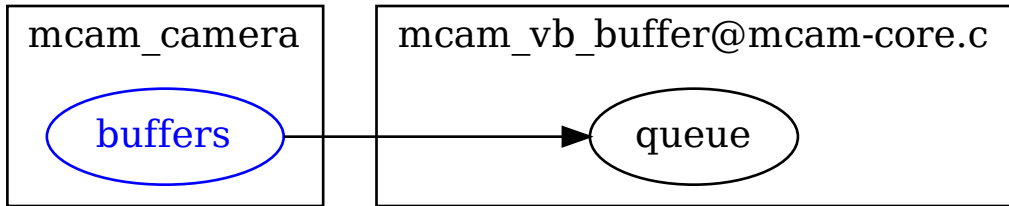
node

```
graph LR; A[cpufreq_cdev_list] --> B((node))
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@cpufreq_cooling.c'. Inside this box, the variable 'cpufreq_cdev_list' is highlighted with a blue oval. On the right, another rectangular box represents the source file 'cpufreq_cooling_device@cpufreq_cooling.c'. Inside this box, there is an oval labeled 'node'. A horizontal arrow points from the right side of the 'cpufreq_cdev_list' oval to the left side of the 'node' oval, indicating that 'cpufreq_cdev_list' points to 'node'.







var@dynamic_debug.c

ddebug_tables

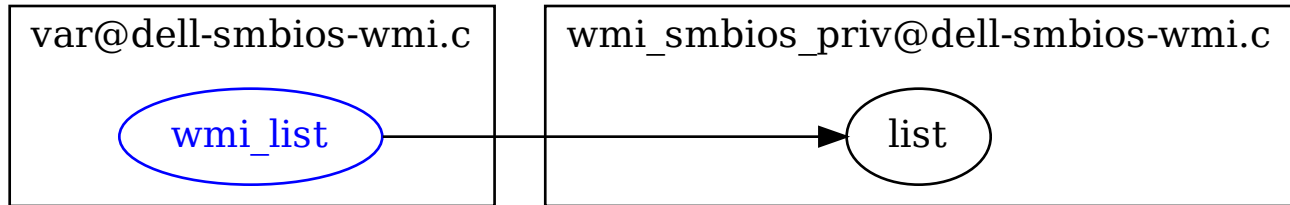
ddebug_table@dynamic_debug.c

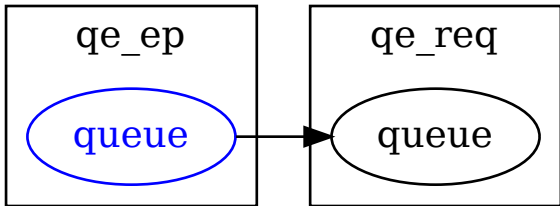
link

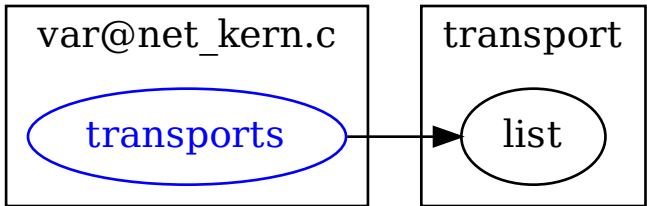


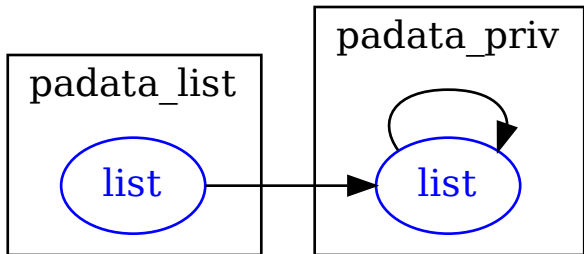
```
graph LR; A(var@dynamic_debug.c) --> B(ddebug_table@dynamic_debug.c); subgraph A; C(ddebug_tables); end; subgraph B; D(link); end; C --> D;
```

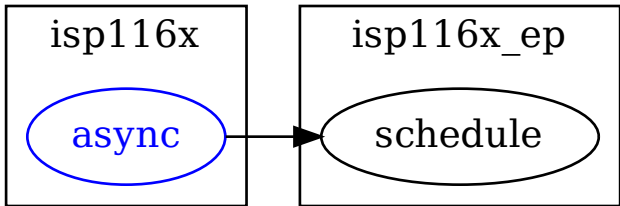
The diagram illustrates a relationship between two variables. On the left, a box labeled 'var@dynamic_debug.c' contains an oval labeled 'ddebug_tables'. On the right, a box labeled 'ddebug_table@dynamic_debug.c' contains an oval labeled 'link'. A horizontal arrow points from the 'ddebug_tables' oval to the 'link' oval, indicating a link or pointer relationship.

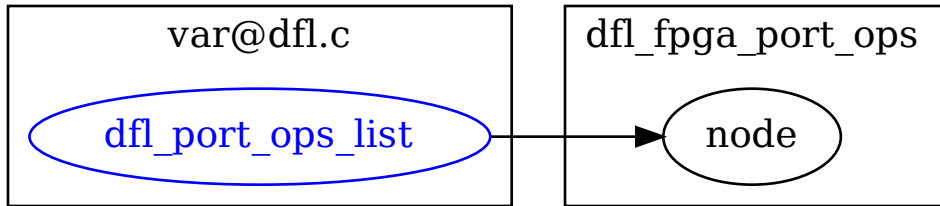


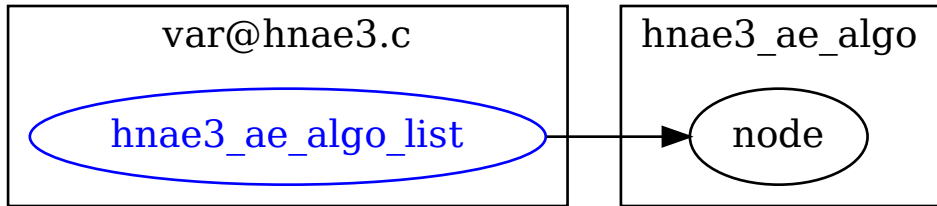


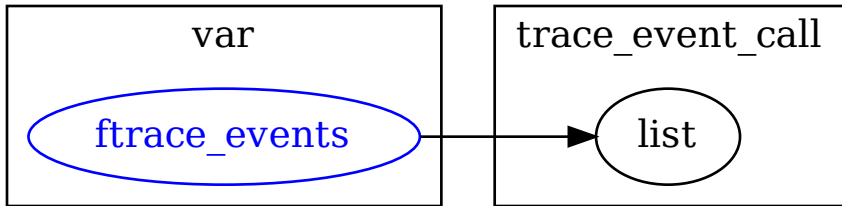


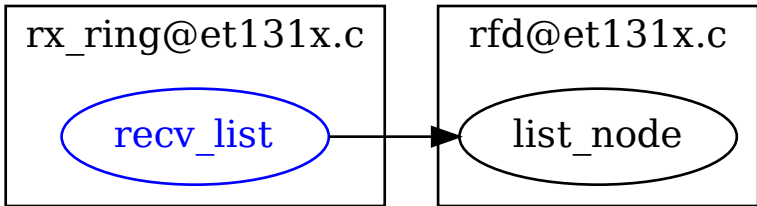












skl_nau8825_private@skl_nau88l25_max98357a.c

hdmi_pcm_list

skl_hdmi_pcm@skl_nau88l25_max98357a.c

head

```
graph LR; subgraph S1 [skl_nau8825_private@skl_nau88l25_max98357a.c]; H1(hdmi_pcm_list); end; subgraph S2 [skl_hdmi_pcm@skl_nau88l25_max98357a.c]; H2(head); end; H1 --> H2;
```

basic_head@cls_basic.c

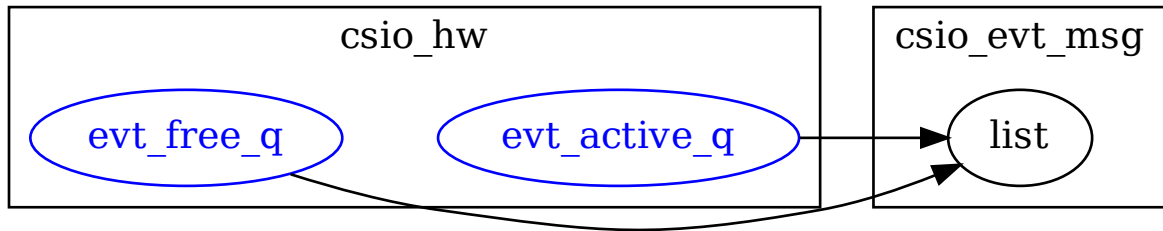
flist

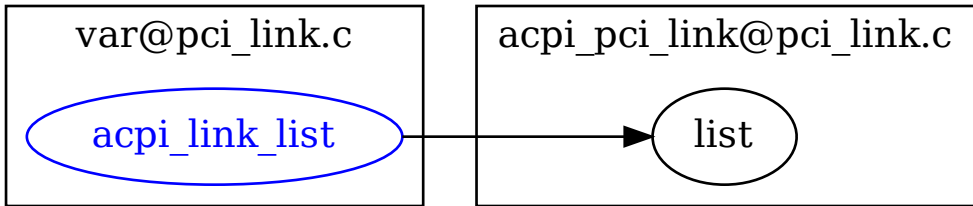
basic_filter@cls_basic.c

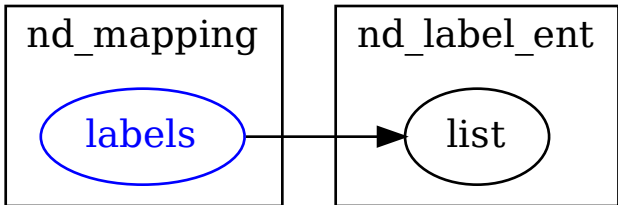
link

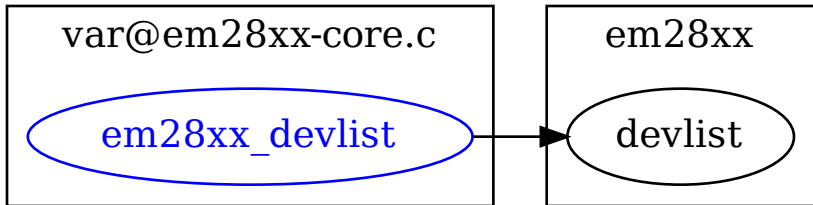
```
graph LR; A[flist] --> B[link];
```

The diagram consists of two rectangular boxes. The left box contains the text 'basic_head@cls_basic.c' at the top and a blue oval containing the text 'flist' below it. The right box contains the text 'basic_filter@cls_basic.c' at the top and a black oval containing the text 'link' below it. A horizontal arrow points from the right side of the 'flist' oval to the left side of the 'link' oval.













qcom_glink@qcom_glink_native.c

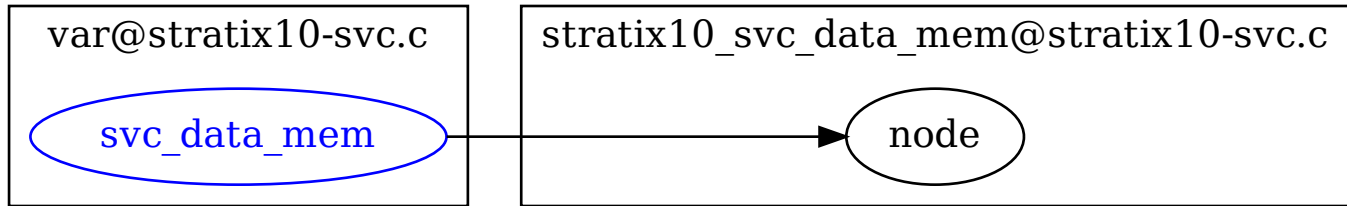
rx_queue

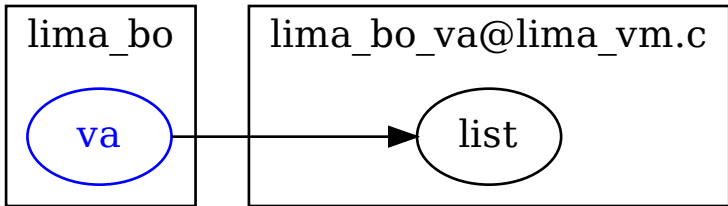
glink_defer_cmd@qcom_glink_native.c

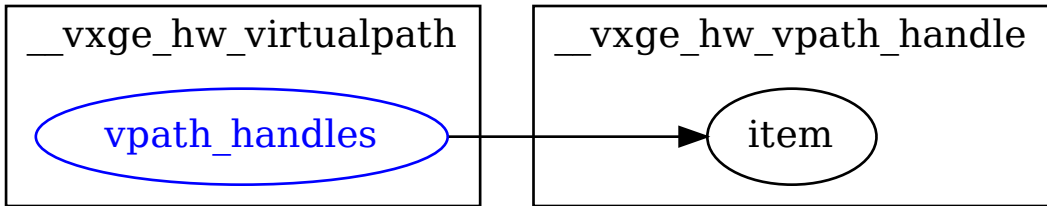
node

```
graph LR; rx_queue(rx_queue) --> node(node);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'qcom_glink@qcom_glink_native.c' contains an oval labeled 'rx_queue' in blue text. On the right, a box labeled 'glink_defer_cmd@qcom_glink_native.c' contains an oval labeled 'node'. A horizontal arrow points from the 'rx_queue' oval to the 'node' oval, indicating that 'rx_queue' points to 'node'.







var@tuner-core.c

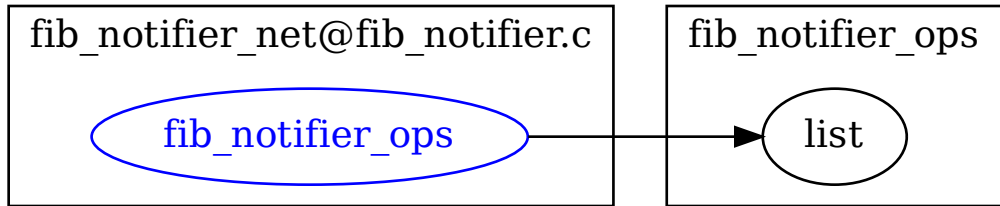
tuner_list

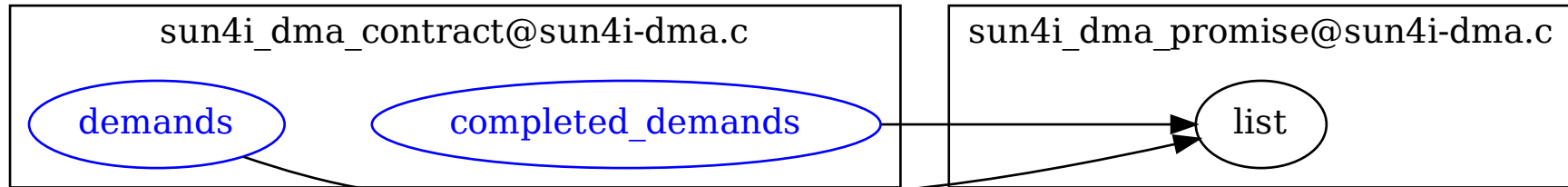
tuner@tuner-core.c

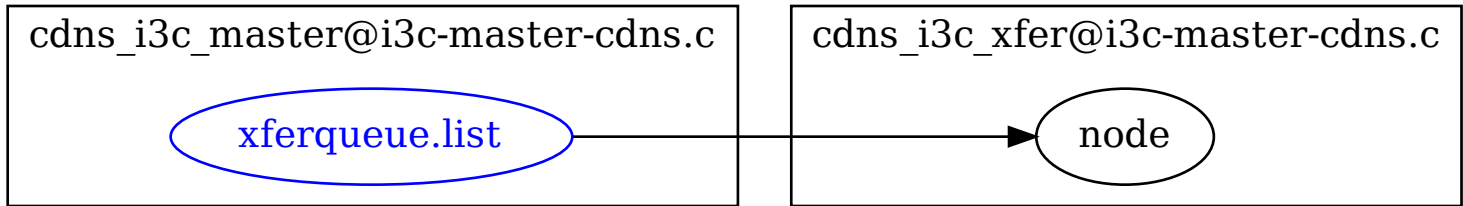
list

```
graph LR; subgraph var [var@tuner-core.c]; tuner_list; end; subgraph tuner [tuner@tuner-core.c]; list; end; tuner_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@tuner-core.c' contains a blue oval labeled 'tuner_list'. On the right, a box labeled 'tuner@tuner-core.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'tuner_list' oval to the 'list' oval, indicating that 'tuner_list' points to 'list'.







var@sclp_vt220.c

sclp_vt220_outqueue

sclp_vt220_request@sclp_vt220.c

list

```
graph LR; subgraph File1 [var@sclp_vt220.c]; A(sclp_vt220_outqueue); end; subgraph File2 [sclp_vt220_request@sclp_vt220.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@sclp_vt220.c`. Inside this box, the variable `sclp_vt220_outqueue` is enclosed in a blue oval. On the right, another rectangular box represents the source file `sclp_vt220_request@sclp_vt220.c`. Inside this box, the variable `list` is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that `sclp_vt220_outqueue` points to `list`.

pxa_camera_dev@pxa_camera.c

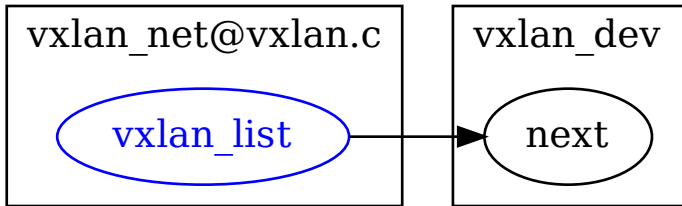
capture

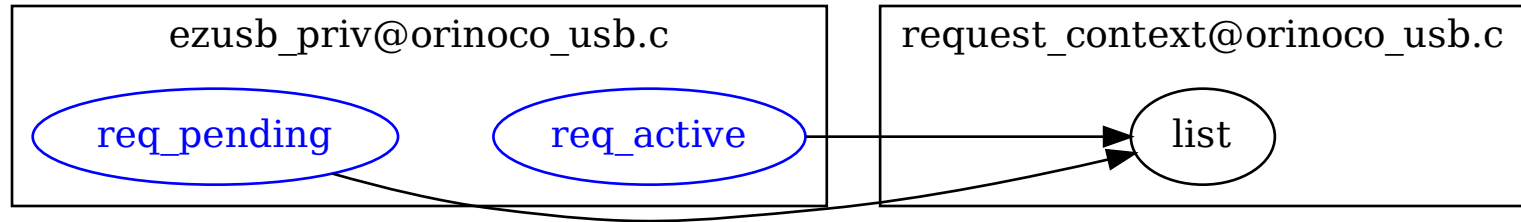
pxa_buffer@pxa_camera.c

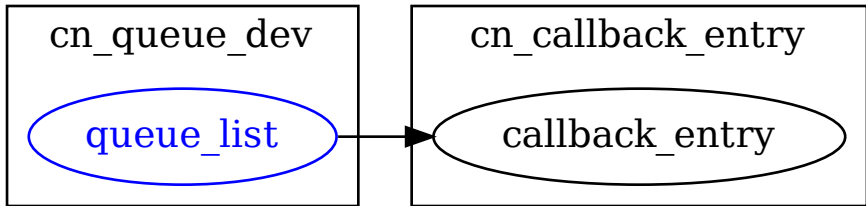
queue

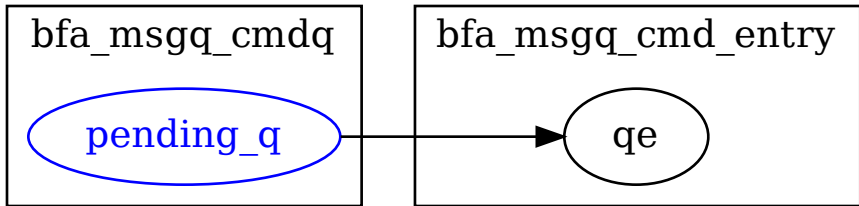
```
graph LR; subgraph pxa_camera_dev [pxa_camera_dev@pxa_camera.c]; capture; end; subgraph pxa_buffer [pxa_buffer@pxa_camera.c]; queue; end; capture --> queue;
```

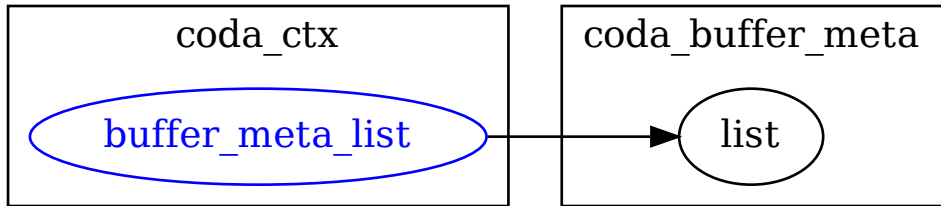
The diagram illustrates a function call sequence between two modules. On the left, a box labeled 'pxa_camera_dev@pxa_camera.c' contains an oval labeled 'capture'. On the right, a box labeled 'pxa_buffer@pxa_camera.c' contains an oval labeled 'queue'. A horizontal arrow points from the 'capture' oval to the 'queue' oval, indicating a call from the camera device to the buffer's queue function.

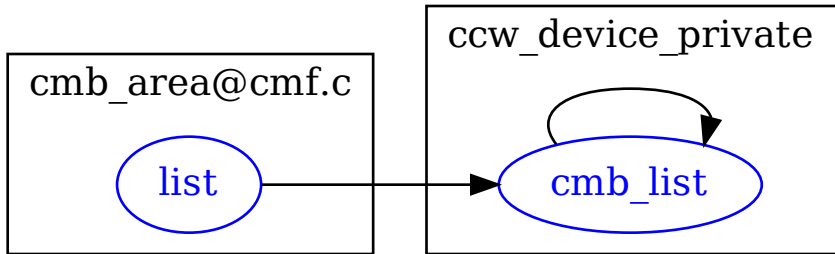


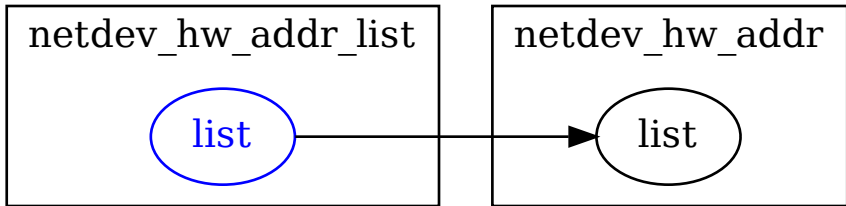


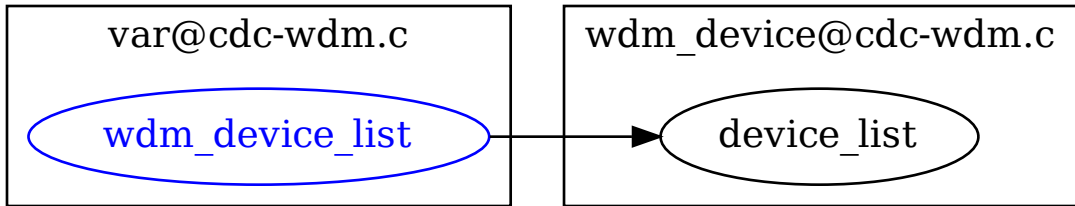












qp_list@vmci_queue_pair.c

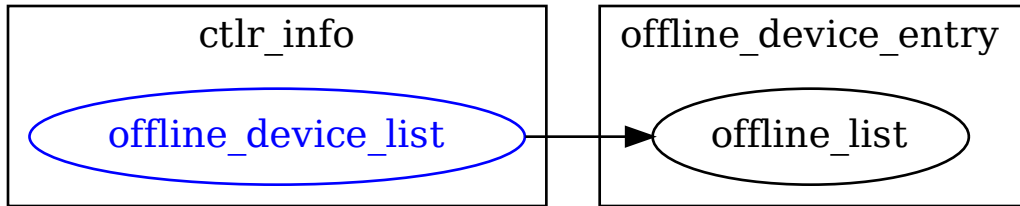
head

qp_entry@vmci_queue_pair.c

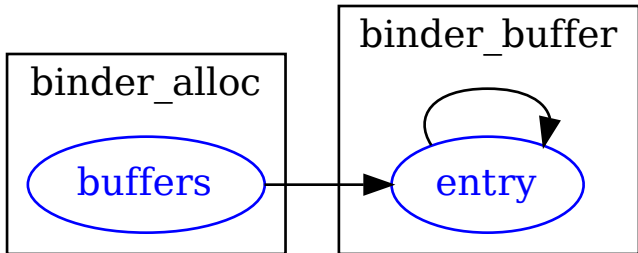
list_item

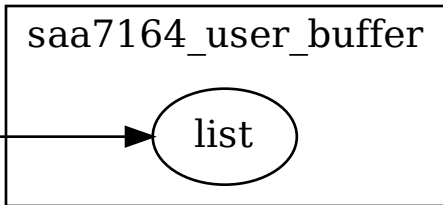
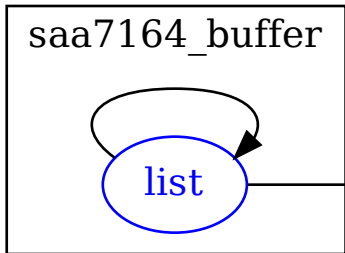
```
graph LR; subgraph qp_list [qp_list@vmci_queue_pair.c]; head((head)); end; subgraph qp_entry [qp_entry@vmci_queue_pair.c]; list_item((list_item)); end; head --> list_item;
```

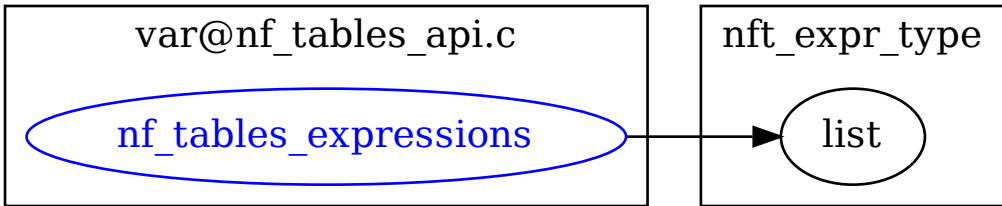
The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'qp_list@vmci_queue_pair.c' contains an oval labeled 'head'. On the right, a box labeled 'qp_entry@vmci_queue_pair.c' contains an oval labeled 'list_item'. A horizontal arrow points from the 'head' oval to the 'list_item' oval, indicating that 'head' points to the 'list_item' structure.

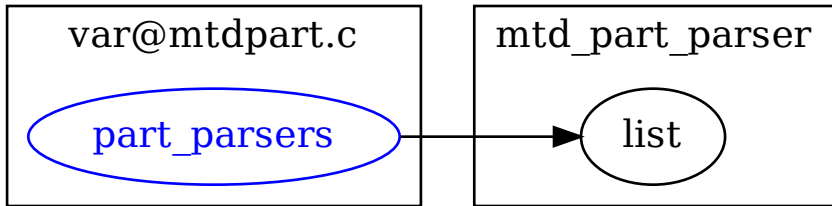


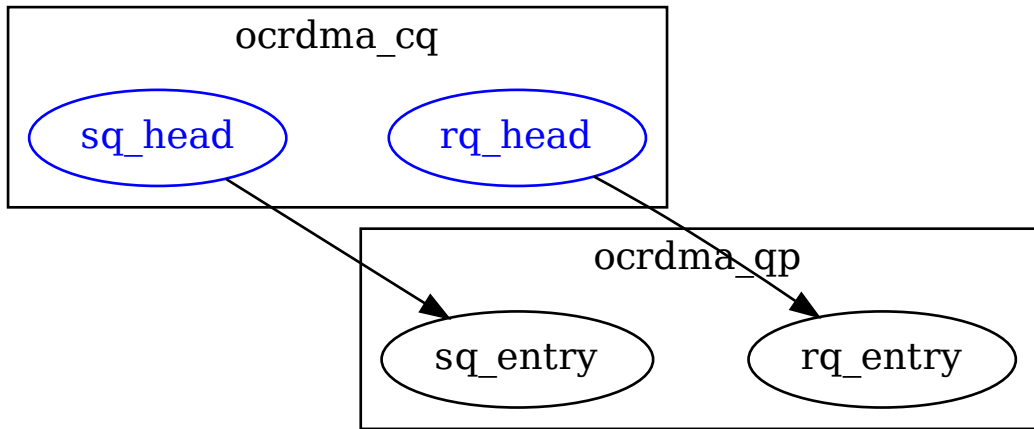


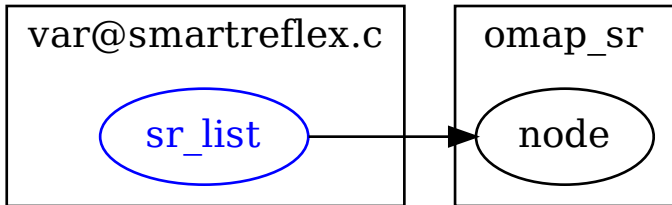


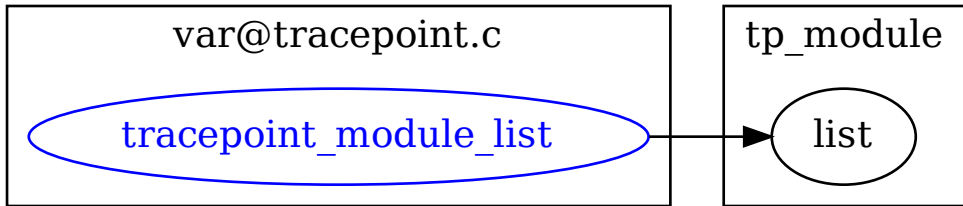


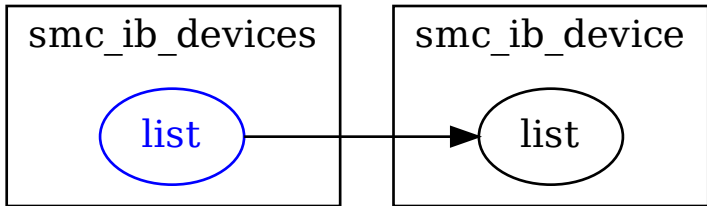


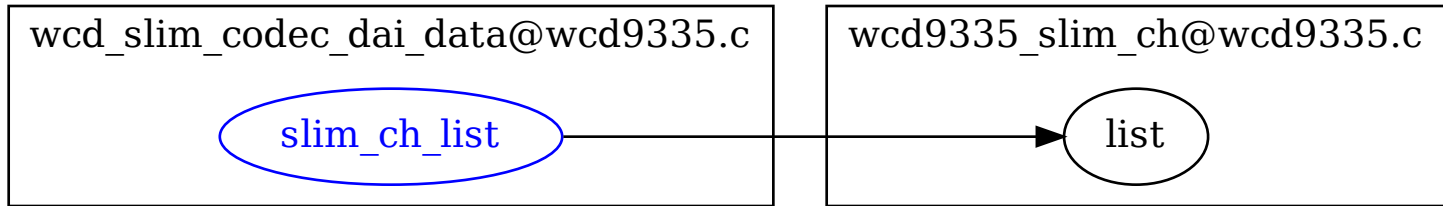


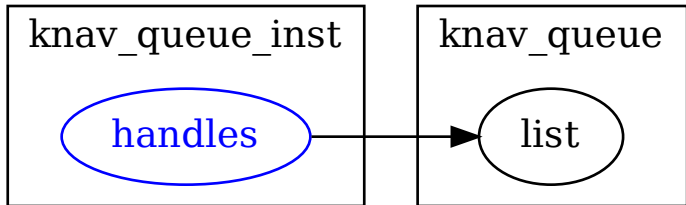












var@stv0910.c

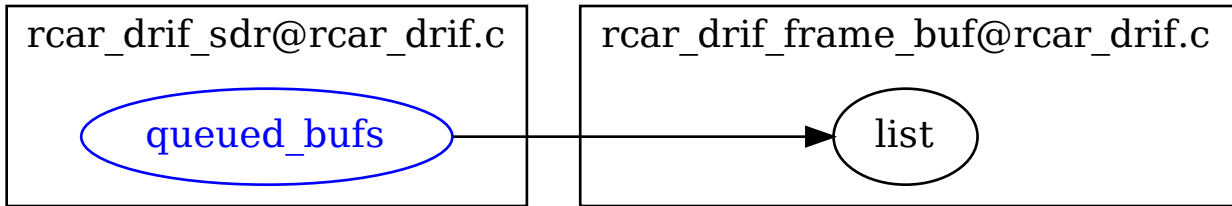
stvlist

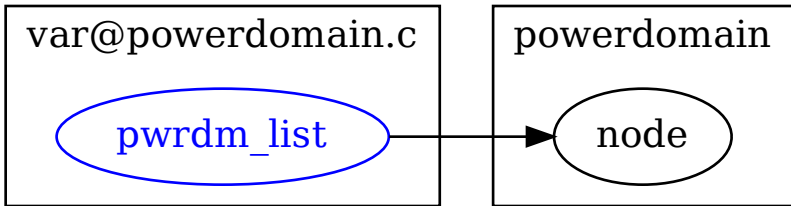
stv_base@stv0910.c

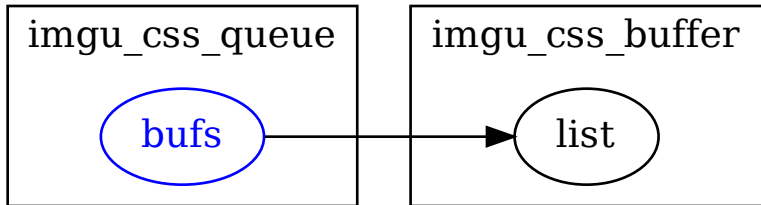
stvlist

```
graph LR; A([stvlist]) --> B([stvlist]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@stv0910.c' contains an oval labeled 'stvlist' with a blue border. On the right, a box labeled 'stv_base@stv0910.c' contains an oval labeled 'stvlist' with a black border. A horizontal arrow points from the 'stvlist' oval in the first box to the 'stvlist' oval in the second box, indicating that the first variable points to the second.







var@wacom_sys.c

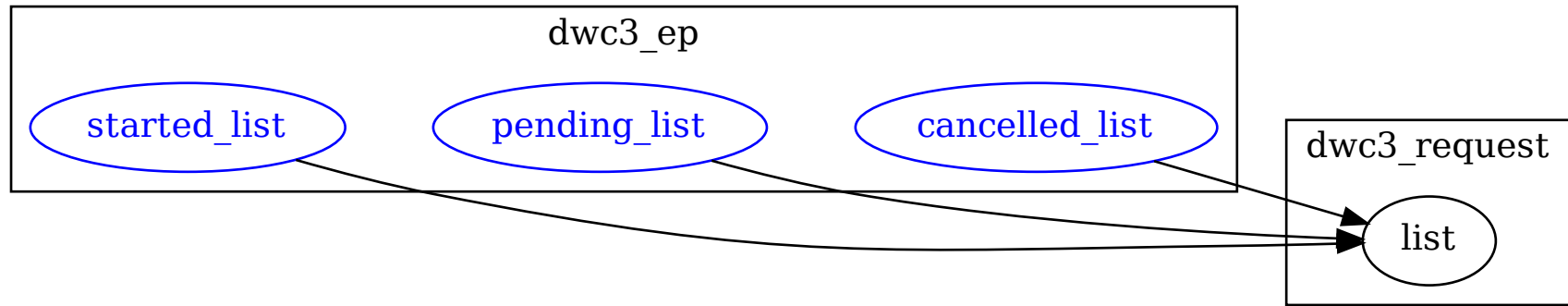
wacom_udev_list

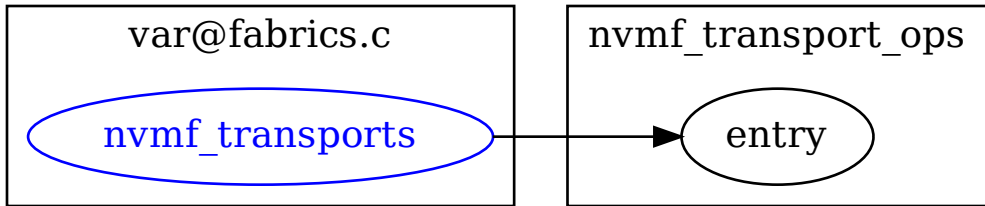
wacom_hdev_data@wacom_sys.c

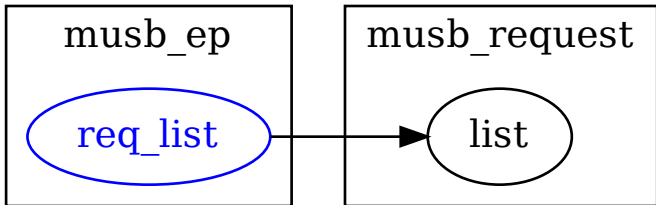
list

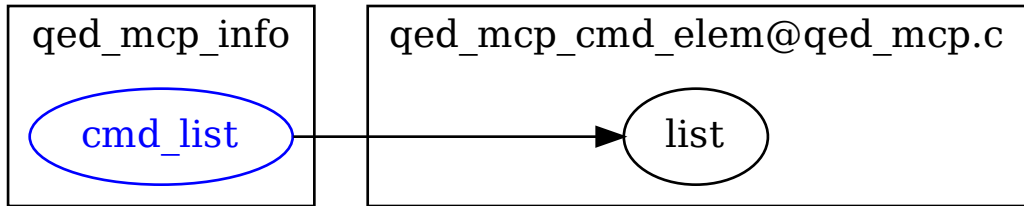
```
graph LR; subgraph Box1 [var@wacom_sys.c]; wacom_udev_list; end; subgraph Box2 [wacom_hdev_data@wacom_sys.c]; list; end; wacom_udev_list --> list;
```

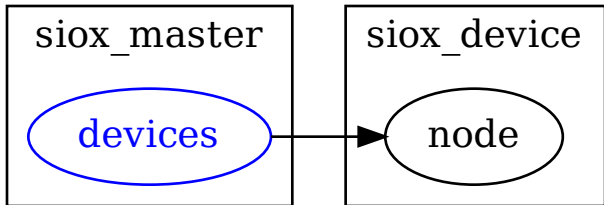
The diagram consists of two rectangular boxes. The left box is labeled 'var@wacom_sys.c' and contains an oval labeled 'wacom_udev_list'. The right box is labeled 'wacom_hdev_data@wacom_sys.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'wacom_udev_list' oval to the 'list' oval.











var@wm0010_firmware_load

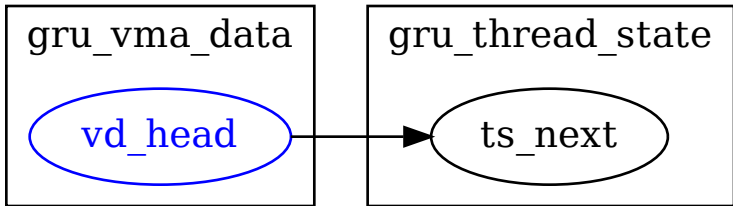
xfer_list

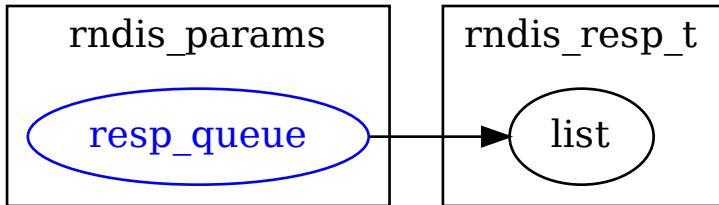
wm0010_boot_xfer@wm0010.c

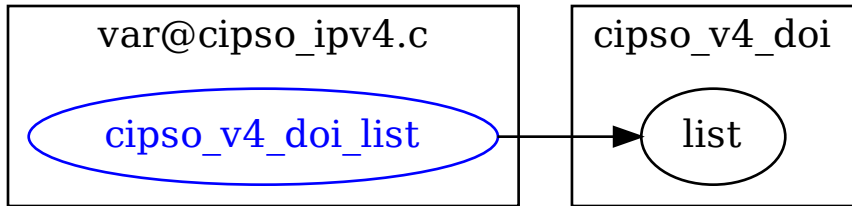
list

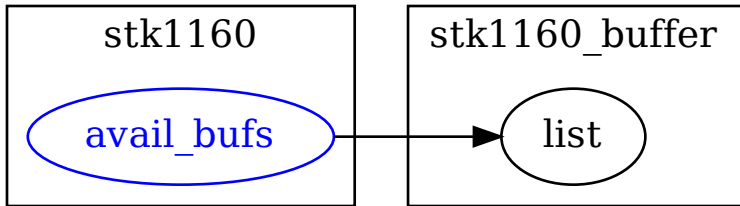
```
graph LR; subgraph Box1 [var@wm0010_firmware_load]; xfer_list((xfer_list)); end; subgraph Box2 [wm0010_boot_xfer@wm0010.c]; list((list)); end; xfer_list --> list;
```

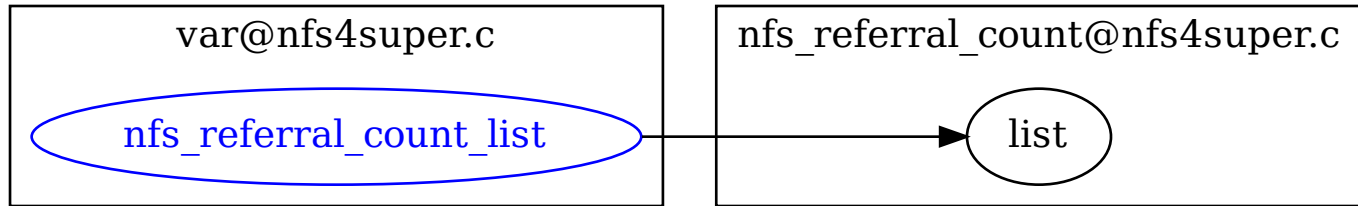
The diagram consists of two rectangular boxes. The left box is labeled 'var@wm0010_firmware_load' and contains an oval labeled 'xfer_list' in blue text. The right box is labeled 'wm0010_boot_xfer@wm0010.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the 'xfer_list' oval to the 'list' oval, indicating a pointer relationship.











chardev_priv@cros_ec_chardev.c

events

ec_event@cros_ec_chardev.c

node

```
graph LR; subgraph Box1 [chardev_priv@cros_ec_chardev.c]; events([events]); end; subgraph Box2 [ec_event@cros_ec_chardev.c]; node([node]); end; events --> node;
```

The diagram consists of two rectangular boxes. The left box is titled 'chardev_priv@cros_ec_chardev.c' and contains a blue oval labeled 'events'. The right box is titled 'ec_event@cros_ec_chardev.c' and contains a black oval labeled 'node'. A horizontal arrow points from the right side of the 'events' oval to the left side of the 'node' oval, indicating a pointer relationship.

var@stack_user.c

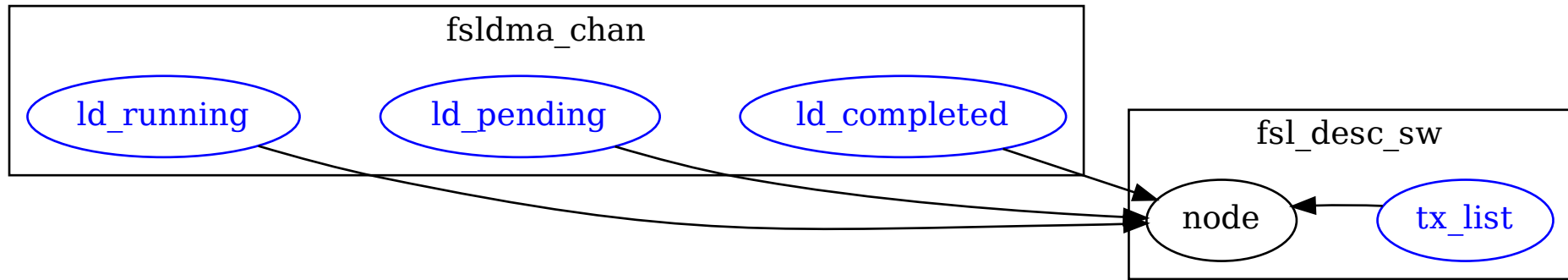
ocfs2_live_connection_list

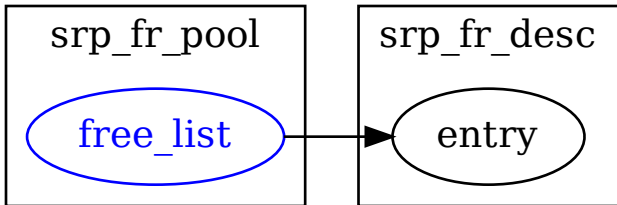
ocfs2_live_connection@stack_user.c

oc_list

```
graph LR; A[ocfs2_live_connection_list] --> B((oc_list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@stack_user.c'. Inside this box, the variable 'ocfs2_live_connection_list' is highlighted with a blue oval. On the right, another rectangular box represents the source file 'ocfs2_live_connection@stack_user.c'. Inside this box, the variable 'oc_list' is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that 'ocfs2_live_connection_list' points to 'oc_list'.





var@regmap-debugfs.c

regmap_debugfs_early_list

regmap_debugfs_node@regmap-debugfs.c

link

```
graph LR; subgraph var [var@regmap-debugfs.c]; early_list([regmap_debugfs_early_list]); end; subgraph node [regmap_debugfs_node@regmap-debugfs.c]; link([link]); end; early_list --> link;
```


var@main.c

blacklisted_initcalls

blacklist_entry@main.c

next

```
graph LR; subgraph var [var@main.c]; blacklisted_initcalls; end; subgraph blacklist_entry [blacklist_entry@main.c]; next; end; blacklisted_initcalls --> next;
```

The diagram illustrates a pointer relationship between two memory locations. On the left, a box labeled 'var@main.c' contains an oval labeled 'blacklisted_initcalls'. On the right, a box labeled 'blacklist_entry@main.c' contains an oval labeled 'next'. A horizontal arrow points from the 'blacklisted_initcalls' oval to the 'next' oval, indicating that the 'blacklisted_initcalls' variable points to the 'next' field of a 'blacklist_entry' structure.

var@test_kmod.c

reg_test_devs

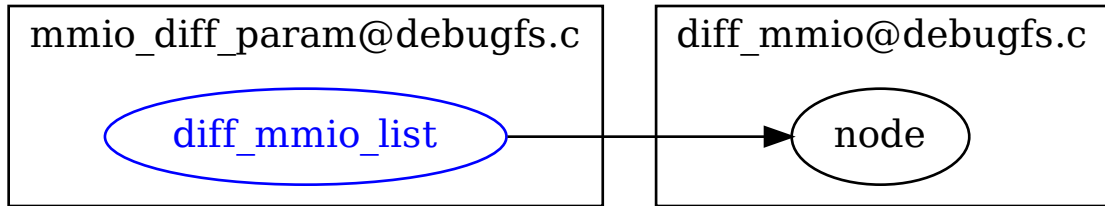
kmod_test_device@test_kmod.c

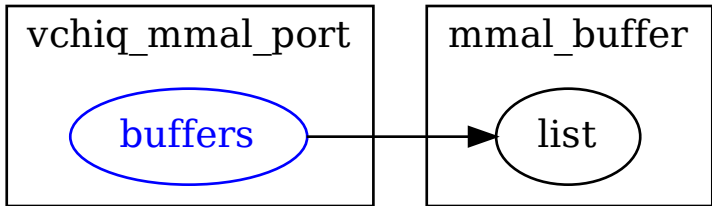
list

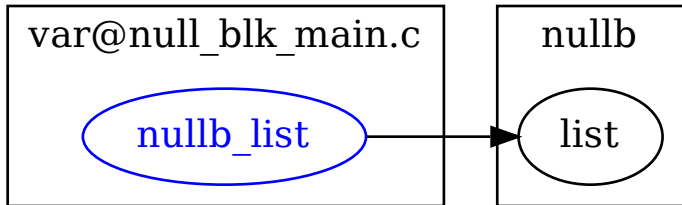
```
graph LR; subgraph File1 [var@test_kmod.c]; A([reg_test_devs]); end; subgraph File2 [kmod_test_device@test_kmod.c]; B((list)); end; A --> B;
```

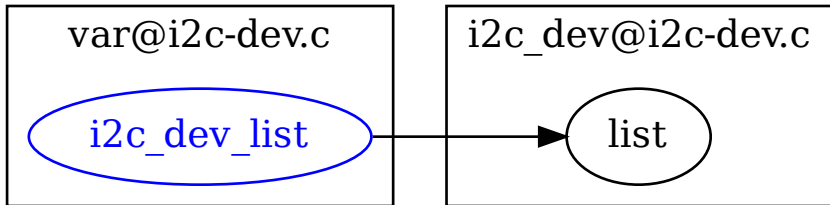
The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file 'var@test_kmod.c', containing a variable 'reg_test_devs' which is highlighted with a blue oval. On the right, a box represents the source file 'kmod_test_device@test_kmod.c', containing a variable 'list' which is enclosed in a black oval. A horizontal arrow points from the 'reg_test_devs' oval to the 'list' oval, indicating that 'reg_test_devs' points to 'list'.

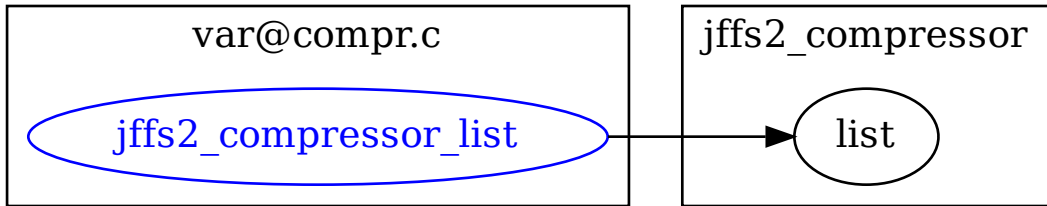


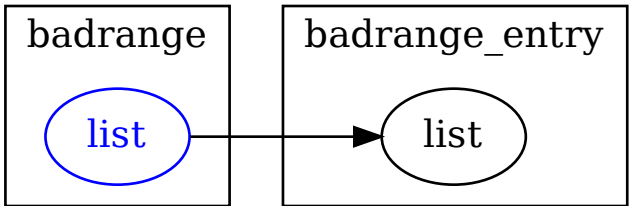


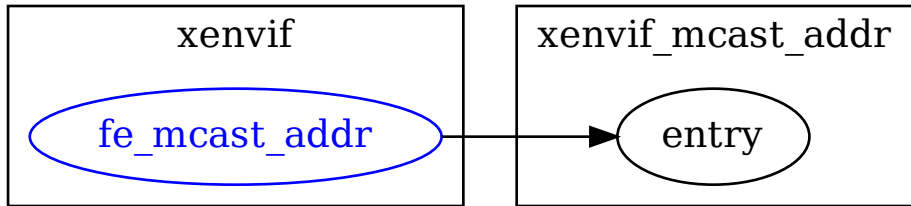


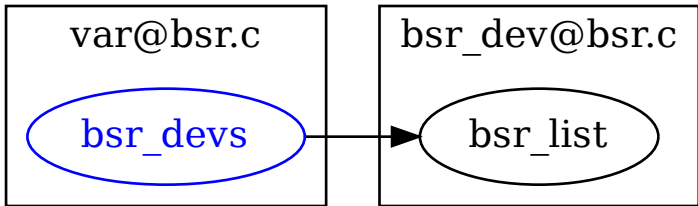


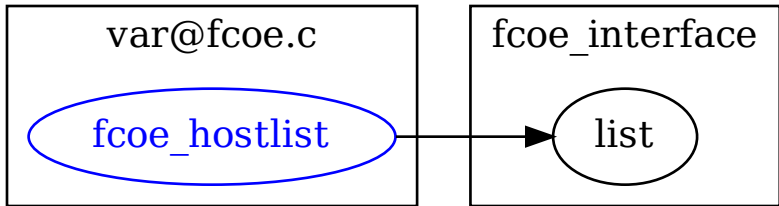


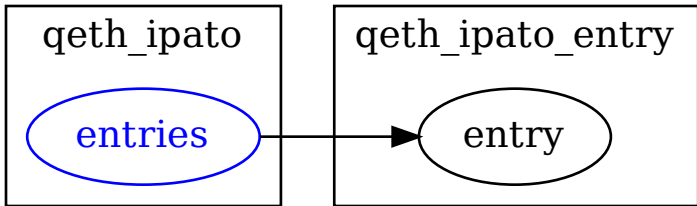


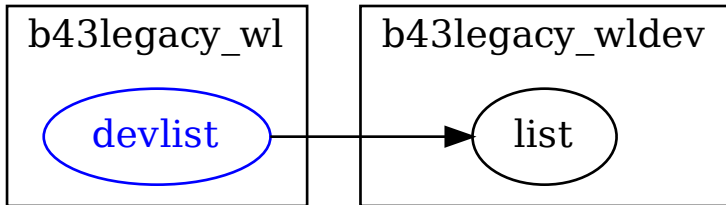












var@amd_iommu_v2.c

state_list

device_state@amd_iommu_v2.c

list

```
graph LR; subgraph var ["var@amd_iommu_v2.c"]; state_list; end; subgraph device_state ["device_state@amd_iommu_v2.c"]; list; end; state_list --> list;
```


The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@amd_iommu_v2.c' contains an oval labeled 'state_list'. On the right, a box labeled 'device_state@amd_iommu_v2.c' contains an oval labeled 'list'. A horizontal arrow points from the 'state_list' oval to the 'list' oval, indicating that 'state_list' points to 'list'.

msdos_inode_info

cache_lru

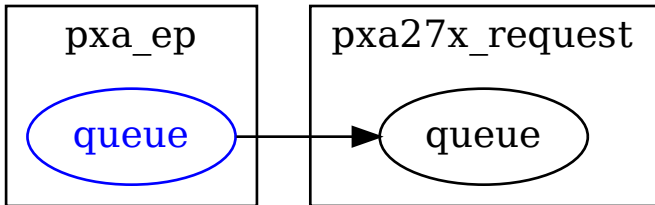
fat_cache@cache.c

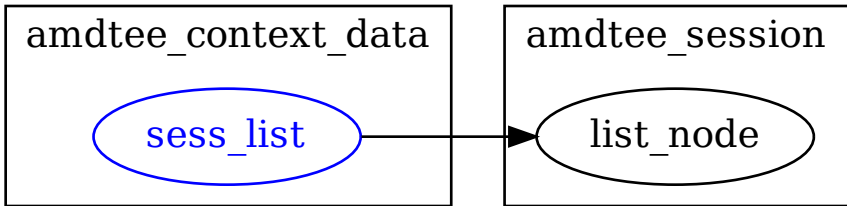
cache_list

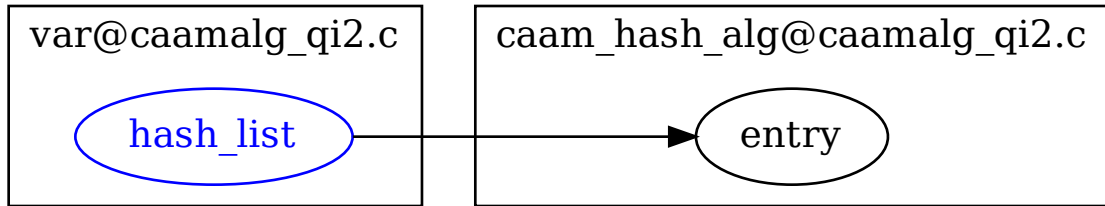


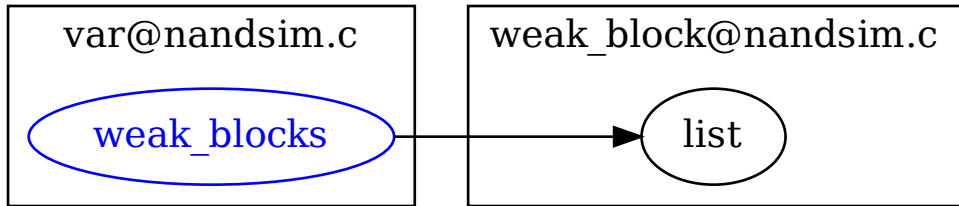
```
graph LR; subgraph A [msdos_inode_info]; C1(cache_lru); end; subgraph B [fat_cache@cache.c]; C2(cache_list); end; C1 --> C2;
```

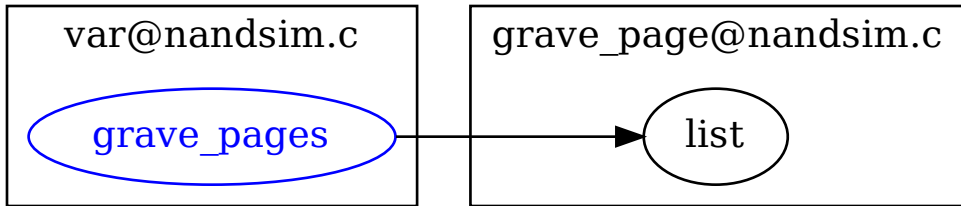
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'msdos_inode_info' contains an oval labeled 'cache_lru'. On the right, a rectangular box labeled 'fat_cache@cache.c' contains an oval labeled 'cache_list'. A horizontal arrow points from the 'cache_lru' oval to the 'cache_list' oval, indicating that 'cache_lru' points to 'cache_list'.











var@kobject_uevent.c

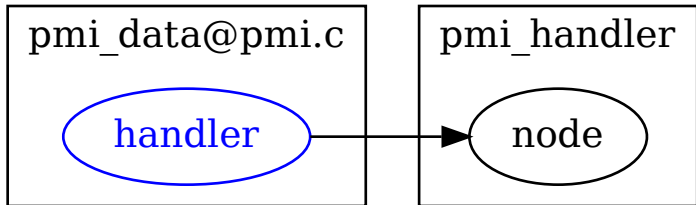
uevent_sock_list

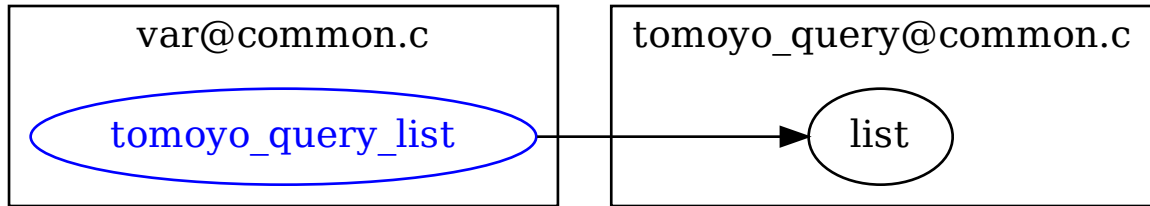
uevent_sock@kobject_uevent.c

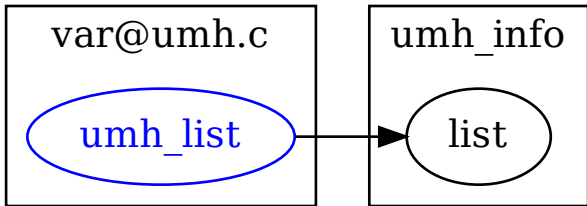
list

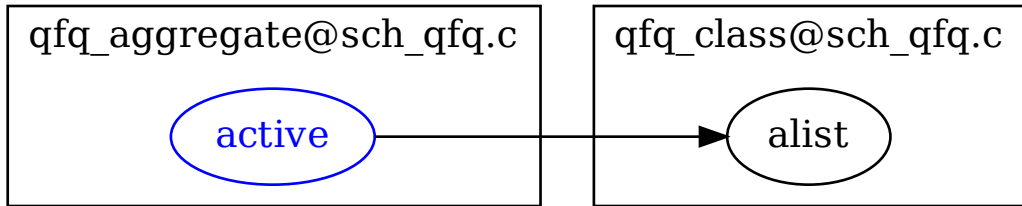
```
graph LR; A[uevent_sock_list] --> B((list));
```

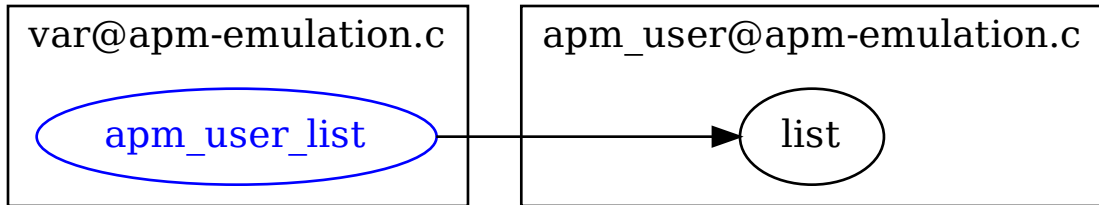
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@kobject_uevent.c' at the top and 'uevent_sock_list' below it. The 'uevent_sock_list' text is enclosed in a blue oval. On the right, another rectangular box contains the text 'uevent_sock@kobject_uevent.c' at the top. Inside this box, there is a black oval containing the text 'list'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that 'uevent_sock_list' points to 'list'.











var@dm-path-selector.c

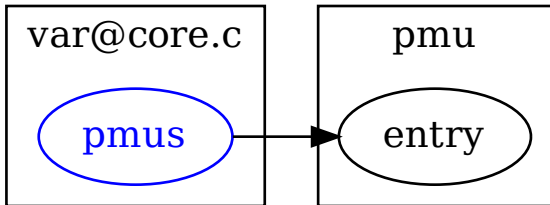
path_selectors

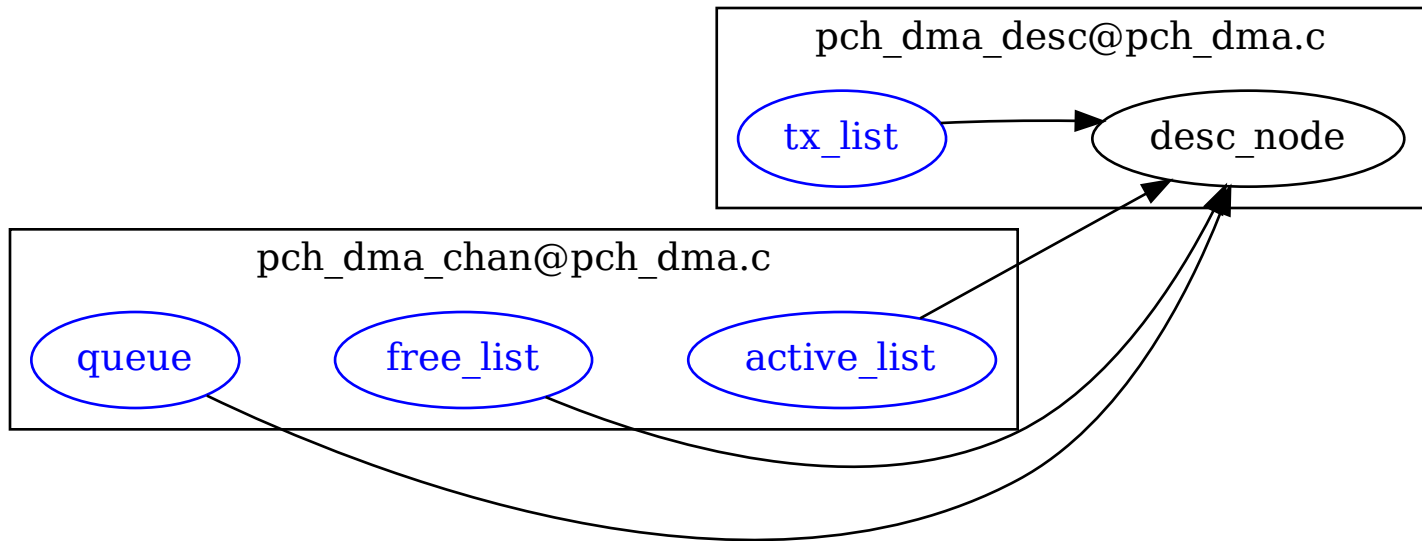
ps_internal@dm-path-selector.c

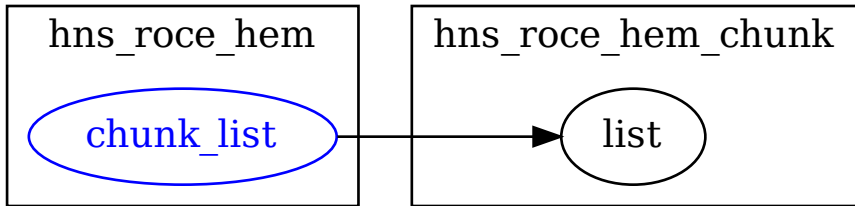
list

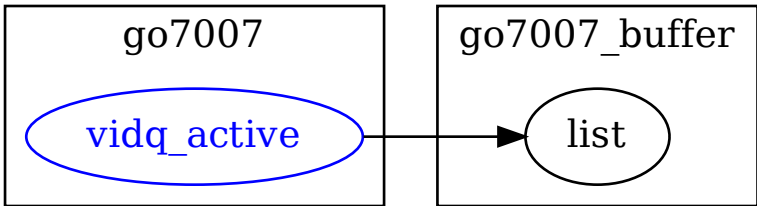
```
graph LR; A["var@dm-path-selector.c  
_path_selectors"] --> B["ps_internal@dm-path-selector.c  
list"]
```

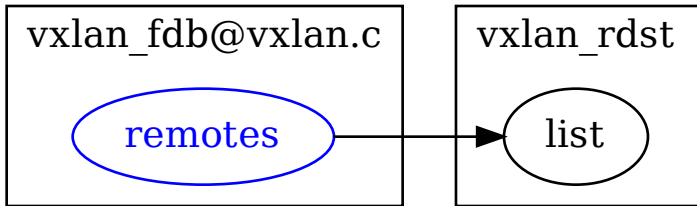
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@dm-path-selector.c' contains an oval labeled 'path_selectors'. On the right, a box labeled 'ps_internal@dm-path-selector.c' contains an oval labeled 'list'. A horizontal arrow points from the 'path_selectors' oval to the 'list' oval, indicating that the 'var' module's 'path_selectors' variable points to the 'list' variable within the 'ps_internal' module.

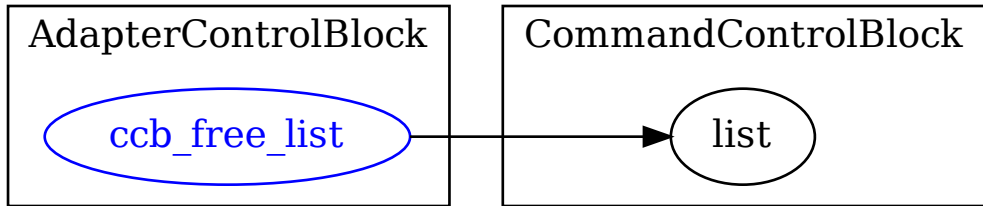


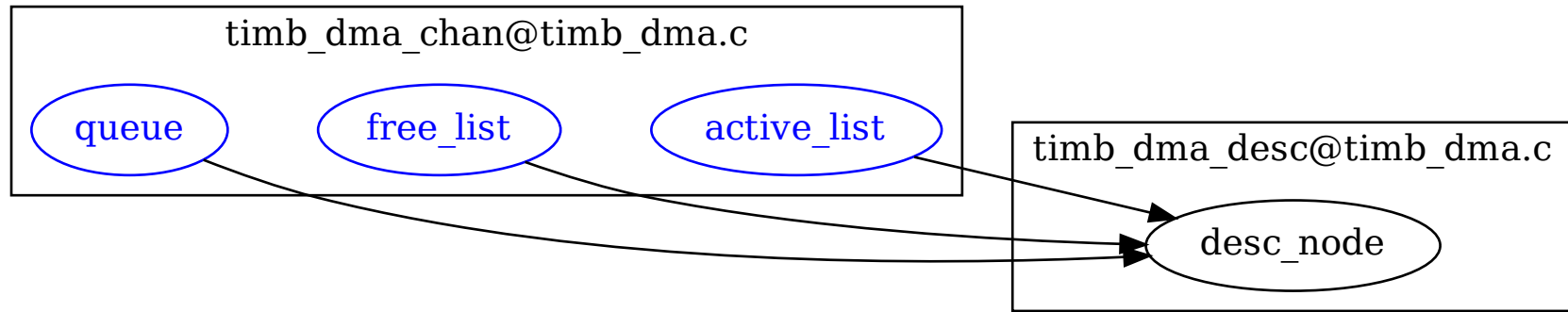


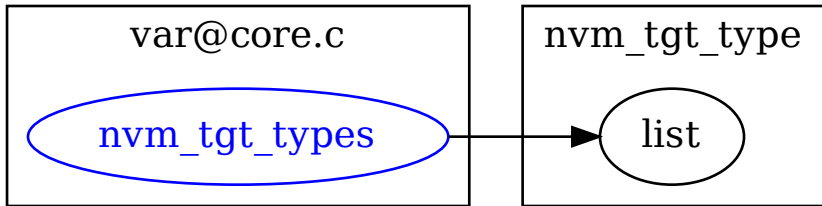


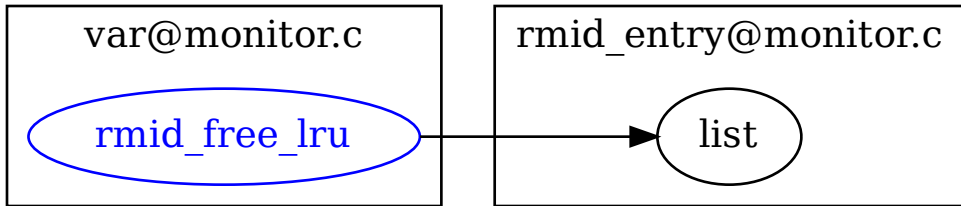


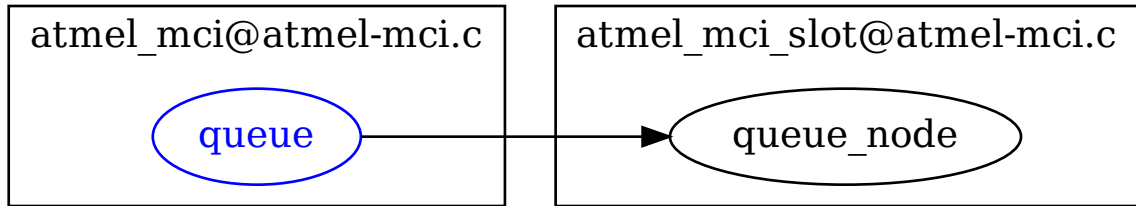












var@opal-memory-errors.c

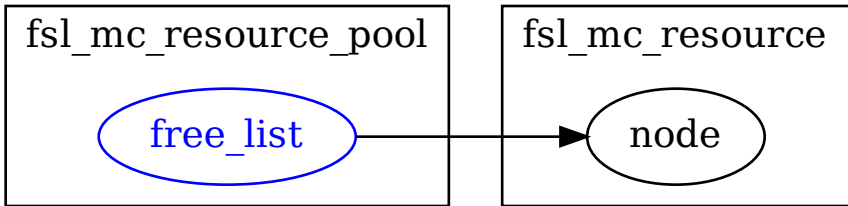
opal_memory_err_list

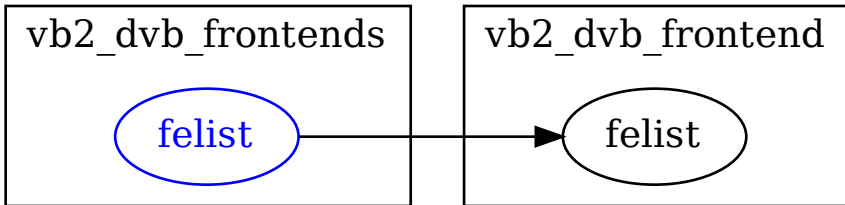
OpalMsgNode@opal-memory-errors.c

list

```
graph LR; A[opal_memory_err_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@opal-memory-errors.c' contains a blue oval labeled 'opal_memory_err_list'. On the right, a rectangular box labeled 'OpalMsgNode@opal-memory-errors.c' contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'opal_memory_err_list' oval to the left side of the 'list' oval, indicating that 'opal_memory_err_list' points to 'list'.





var@authentication.c

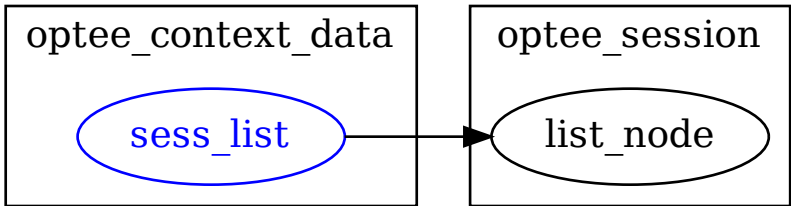
cap_list

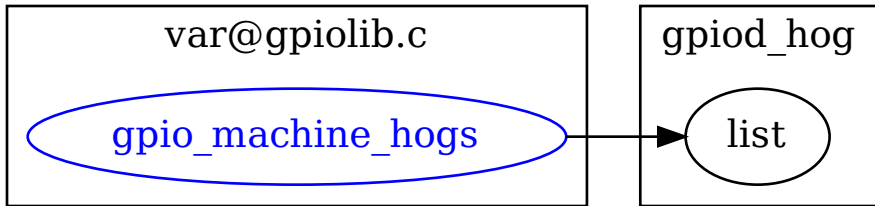
gb_cap@authentication.c

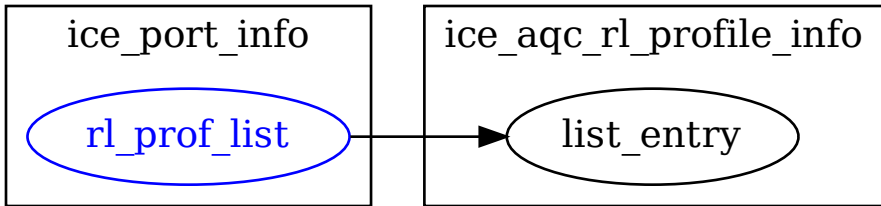
node

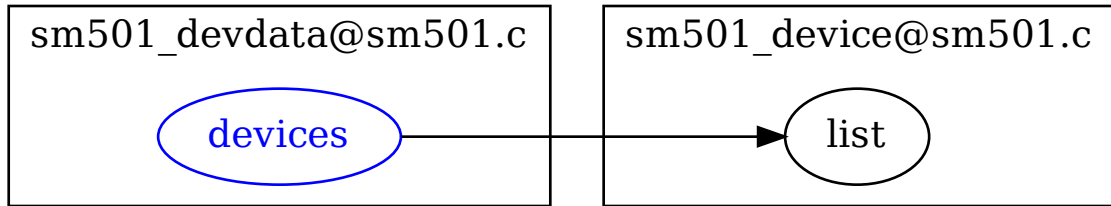
```
graph LR; subgraph var_authentication_c [var@authentication.c]; cap_list((cap_list)); end; subgraph gb_cap_authentication_c [gb_cap@authentication.c]; node((node)); end; cap_list --> node;
```

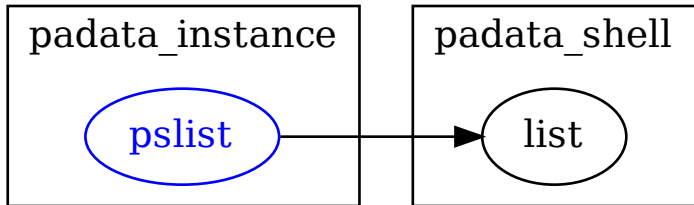
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@authentication.c' contains an oval labeled 'cap_list'. On the right, a box labeled 'gb_cap@authentication.c' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'cap_list' oval to the left side of the 'node' oval, indicating that 'cap_list' points to 'node'.











pvscsi_adapter@vmw_pvscsi.c

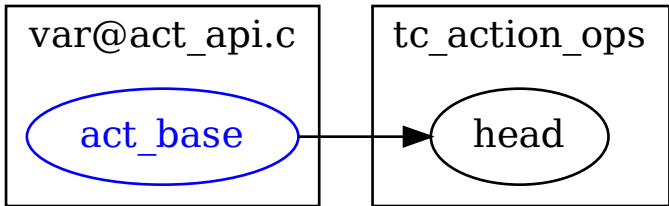
cmd_pool

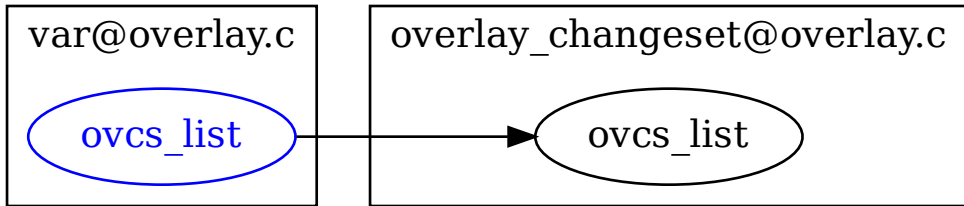
pvscsi_ctx@vmw_pvscsi.c

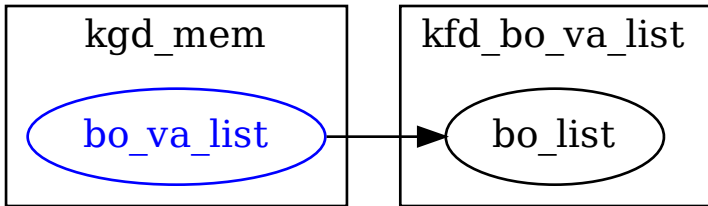
list

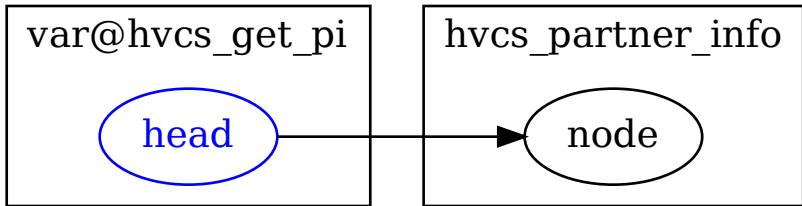
```
graph LR; subgraph Adapter [pvscsi_adapter@vmw_pvscsi.c]; cmd_pool; end; subgraph Context [pvscsi_ctx@vmw_pvscsi.c]; list; end; cmd_pool --> list;
```

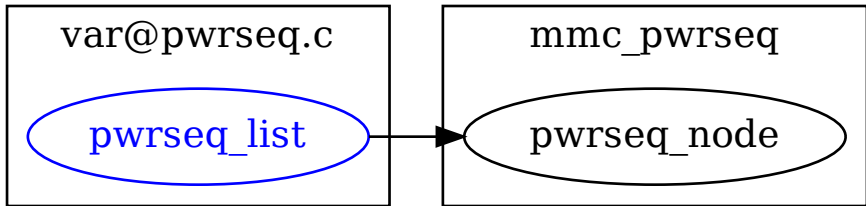
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box represents the `pvscsi_adapter@vmw_pvscsi.c` module, containing an oval labeled `cmd_pool`. On the right, another rectangular box represents the `pvscsi_ctx@vmw_pvscsi.c` module, containing an oval labeled `list`. A horizontal arrow points from the `cmd_pool` oval to the `list` oval, indicating that `cmd_pool` points to `list`.











var@btrfs_fallocate

reserve_list

falloc_range@file.c

list

```
graph LR; A[reserve_list] --> B(list);
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@btrfs_fallocate' at the top and an oval containing 'reserve_list' in blue text below it. The right box contains the text 'falloc_range@file.c' at the top and an oval containing 'list' in black text below it. A horizontal arrow points from the right side of the 'reserve_list' oval to the left side of the 'list' oval.

brcmnand_controller@brcmnand.c

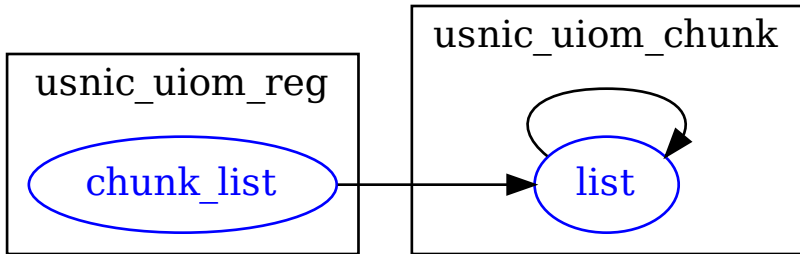
host_list

brcmnand_host@brcmnand.c

node

```
graph LR; subgraph Controller [brcmnand_controller@brcmnand.c]; HL((host_list)); end; subgraph Host [brcmnand_host@brcmnand.c]; N((node)); end; HL --> N;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'brcmnand_controller@brcmnand.c' contains a blue oval labeled 'host_list'. On the right, a box labeled 'brcmnand_host@brcmnand.c' contains a black oval labeled 'node'. A horizontal arrow points from the 'host_list' oval to the 'node' oval, indicating that the 'host_list' variable in the controller module points to the 'node' variable in the host module.



stub_chip@i2c-stub.c

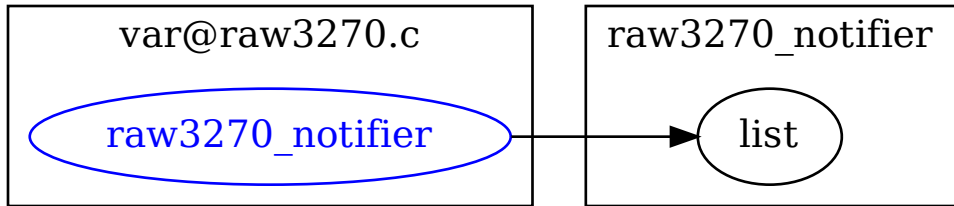
smbus_blocks

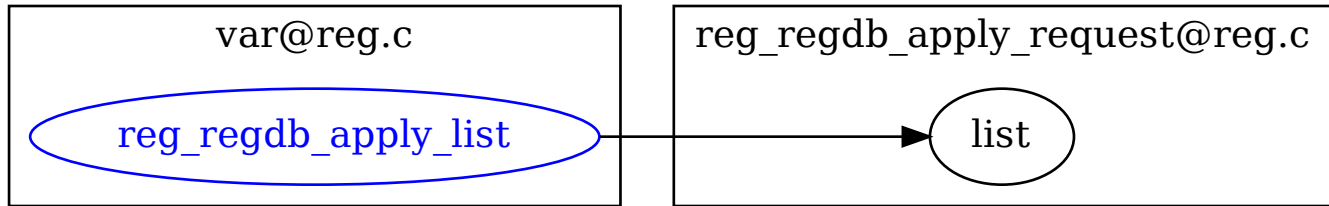
smbus_block_data@i2c-stub.c

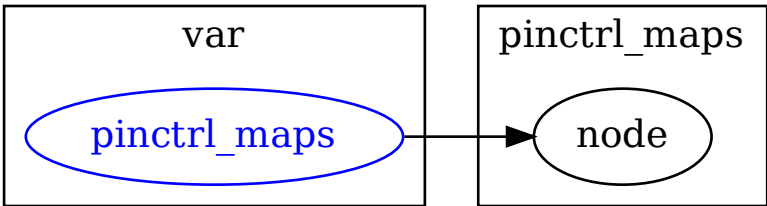
node

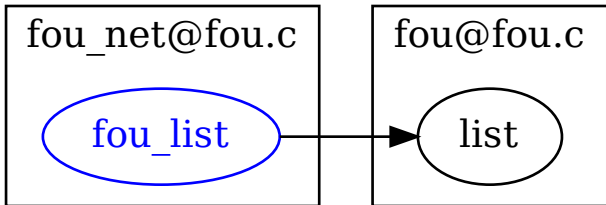
```
graph LR; subgraph StubChip [stub_chip@i2c-stub.c]; smbus_blocks; end; subgraph SmbusBlockData [smbus_block_data@i2c-stub.c]; node; end; smbus_blocks --> node;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'stub_chip@i2c-stub.c' contains an oval labeled 'smbus_blocks' with a blue border. On the right, a rectangular box labeled 'smbus_block_data@i2c-stub.c' contains an oval labeled 'node'. A horizontal arrow points from the 'smbus_blocks' oval to the 'node' oval, indicating that 'smbus_blocks' points to 'node'.









var@rdma.c

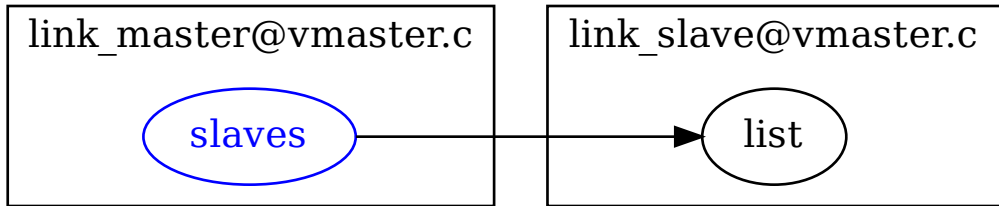
device_list

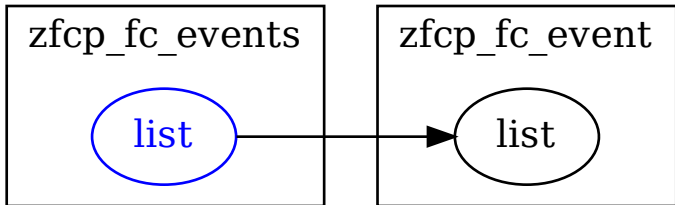
nvmet_rdma_device@rdma.c

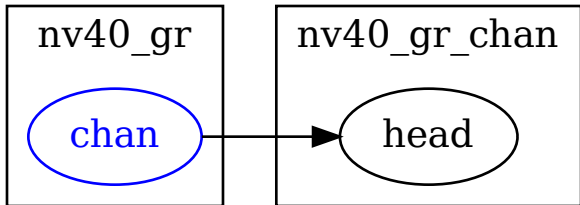
entry

```
graph LR; subgraph var_at_rdma_c [var@rdma.c]; device_list; end; subgraph nvmet_rdma_device_at_rdma_c [nvmet_rdma_device@rdma.c]; entry; end; device_list --> entry;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@rdma.c' contains an oval labeled 'device_list'. On the right, a box labeled 'nvmet_rdma_device@rdma.c' contains an oval labeled 'entry'. A horizontal arrow points from the 'device_list' oval to the 'entry' oval, indicating that 'device_list' points to 'entry'.







var@k3-ringacc.c

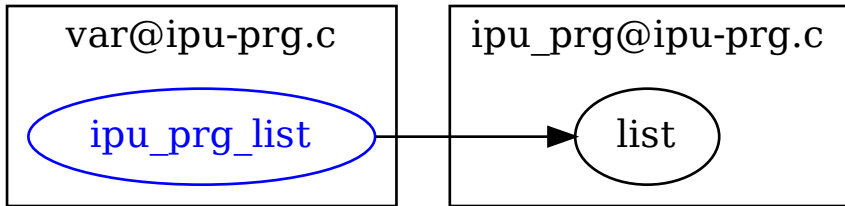
k3_ringacc_list

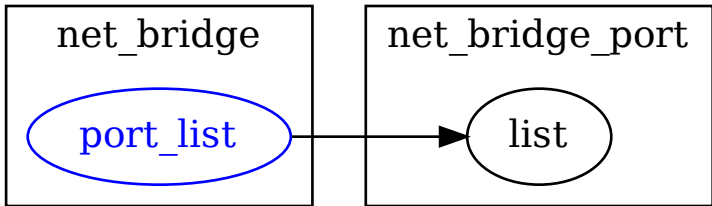
k3_ringacc@k3-ringacc.c

list

```
graph LR; subgraph Module1 [var@k3-ringacc.c]; k3_ringacc_list; end; subgraph Module2 [k3_ringacc@k3-ringacc.c]; list; end; k3_ringacc_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@k3-ringacc.c' contains a blue oval labeled 'k3_ringacc_list'. On the right, a box labeled 'k3_ringacc@k3-ringacc.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'k3_ringacc_list' oval to the 'list' oval, indicating that the variable 'k3_ringacc_list' in the first module points to the 'list' variable in the second module.





var@pm34xx.c

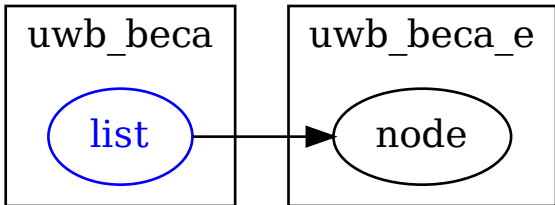
pwrst_list

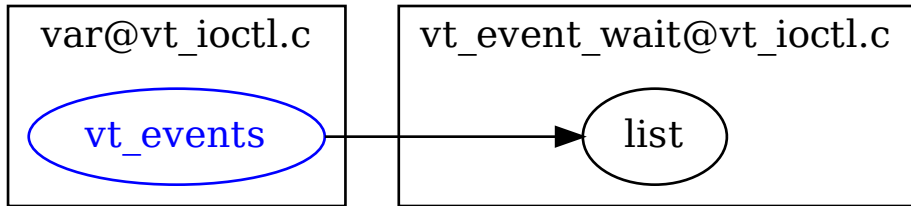
power_state@pm34xx.c

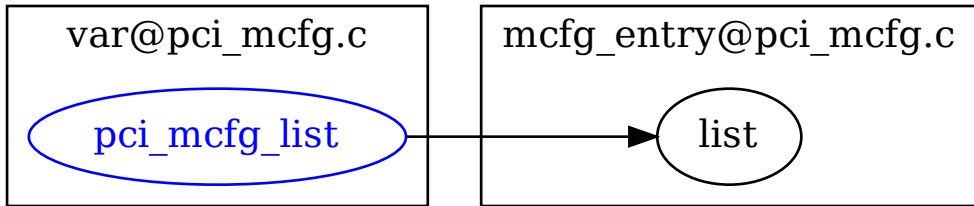
node

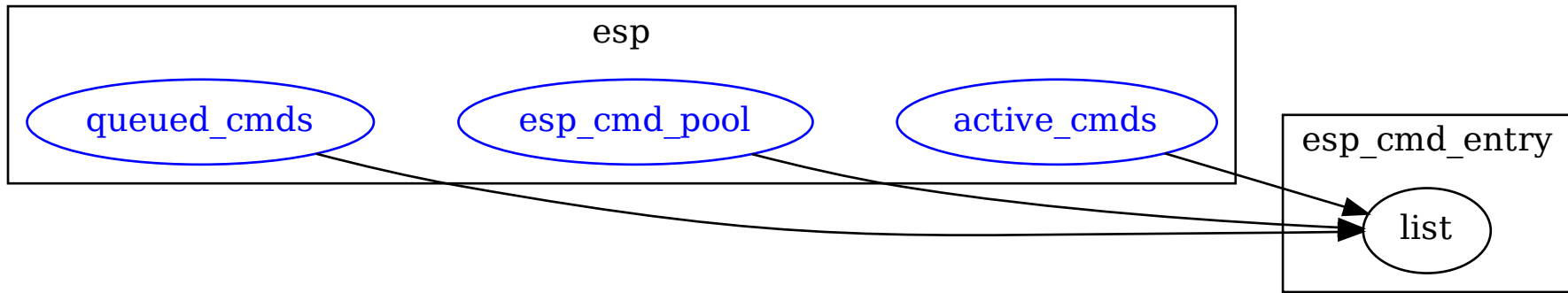
```
graph LR; subgraph var_pm34xx_c [var@pm34xx.c]; pwrst_list; end; subgraph power_state_pm34xx_c [power_state@pm34xx.c]; node; end; pwrst_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@pm34xx.c' contains an oval labeled 'pwrst_list'. On the right, a box labeled 'power_state@pm34xx.c' contains an oval labeled 'node'. A horizontal arrow points from the 'pwrst_list' oval to the 'node' oval, indicating that 'pwrst_list' points to 'node'.









var@aer_inject.c

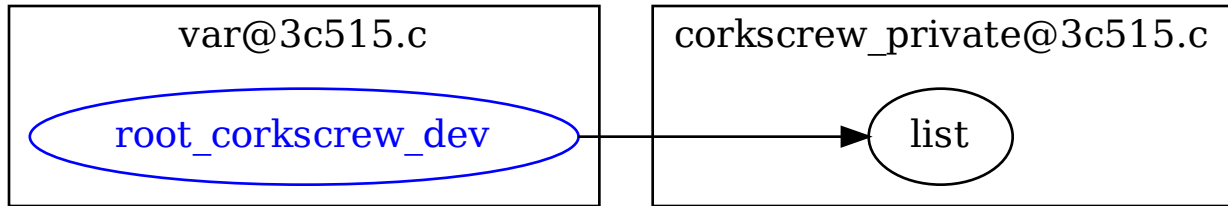
pci_bus_ops_list

pci_bus_ops@aer_inject.c

list

```
graph LR; subgraph "var@aer_inject.c"; pci_bus_ops_list; end; subgraph "pci_bus_ops@aer_inject.c"; list; end; pci_bus_ops_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@aer_inject.c' contains the variable 'pci_bus_ops_list', which is highlighted by a blue oval. On the right, a box labeled 'pci_bus_ops@aer_inject.c' contains a variable 'list' inside a black oval. A horizontal arrow points from the 'pci_bus_ops_list' oval to the 'list' oval, indicating that 'pci_bus_ops_list' points to 'list'.



var@vector_kern.c

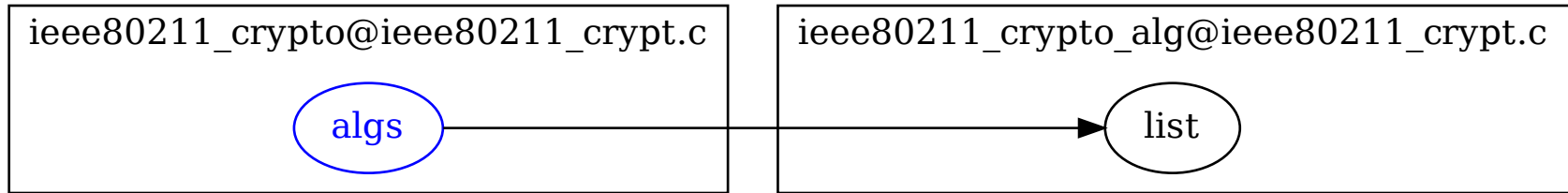
vec_cmd_line

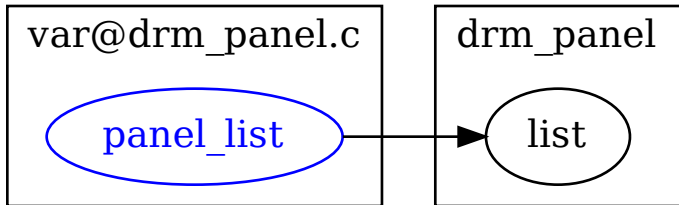
vector_cmd_line_arg@vector_kern.c

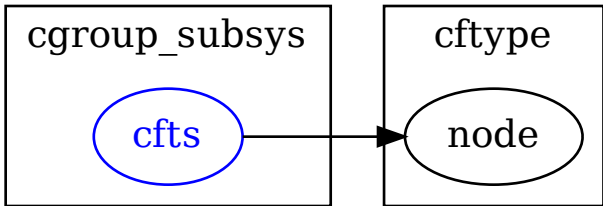
list

```
graph LR; subgraph Box1 [var@vector_kern.c]; V[vec_cmd_line]; end; subgraph Box2 [vector_cmd_line_arg@vector_kern.c]; L((list)); end; V --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@vector_kern.c' and contains an oval labeled 'vec_cmd_line' in blue text. The right box is labeled 'vector_cmd_line_arg@vector_kern.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'vec_cmd_line' oval to the left side of the 'list' oval.







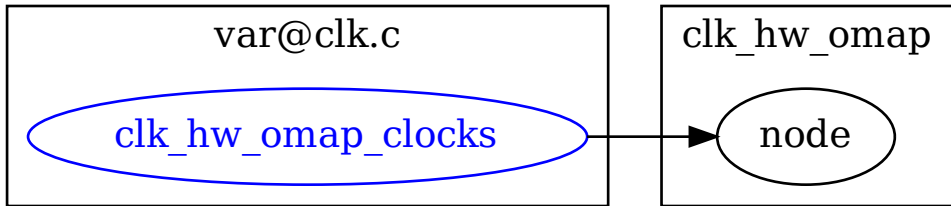
var@drm_fb_helper.c

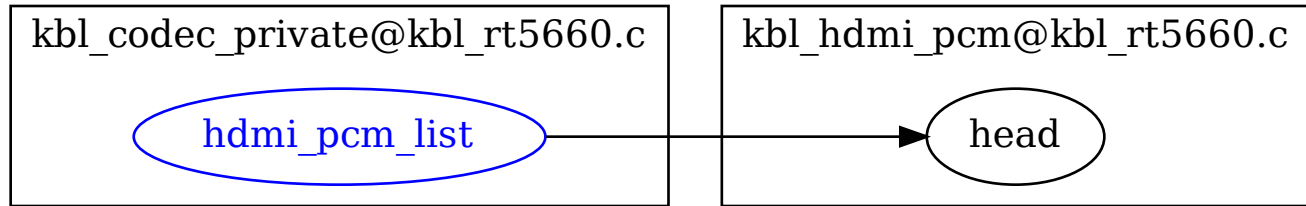
kernel_fb_helper_list

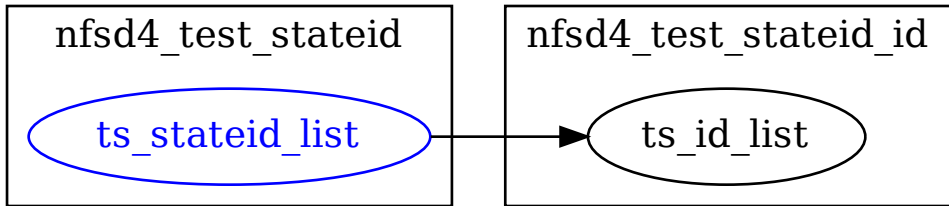
drm_fb_helper

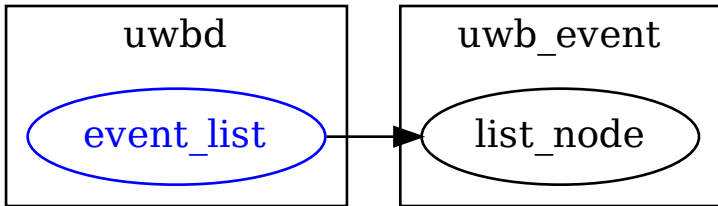
kernel_fb_list

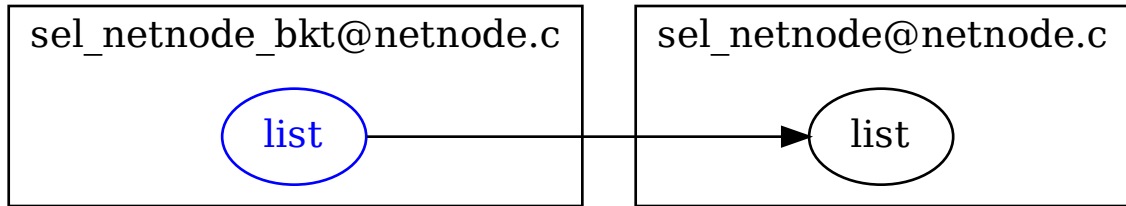


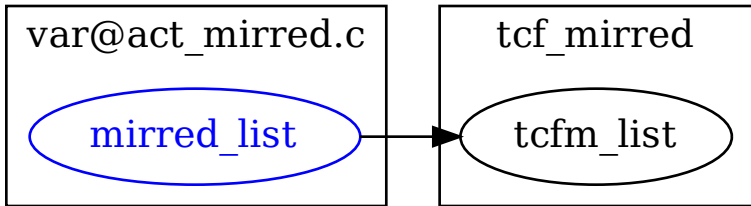


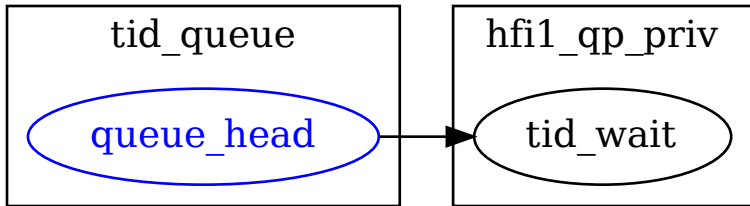












async_chunk@inode.c

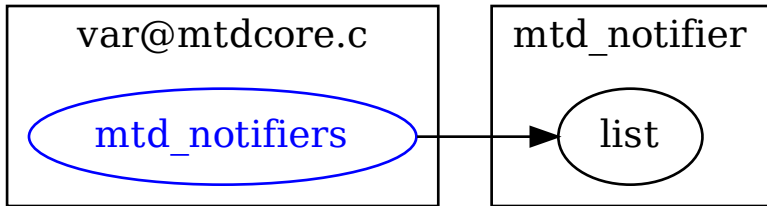
extents

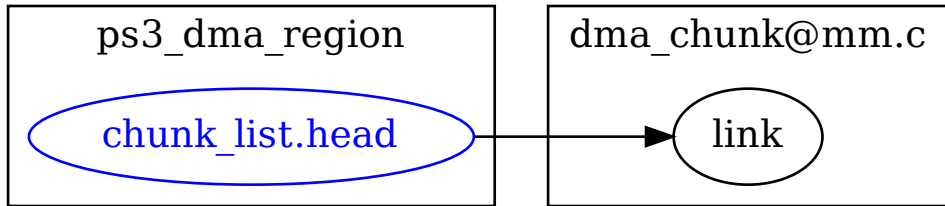
async_extent@inode.c

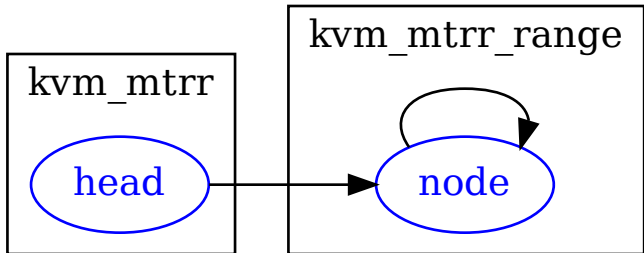
list

```
graph LR; subgraph "async_chunk@inode.c"; extents; end; subgraph "async_extent@inode.c"; list; end; extents --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'async_chunk@inode.c' contains an oval labeled 'extents'. On the right, a box labeled 'async_extent@inode.c' contains an oval labeled 'list'. A horizontal arrow points from the 'extents' oval to the 'list' oval, indicating that the 'extents' variable points to the 'list' structure.







var@vmci_event.c

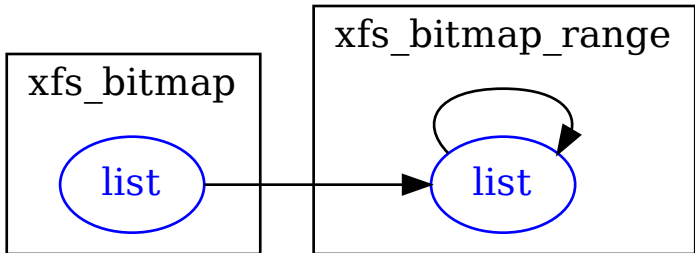
subscriber_array

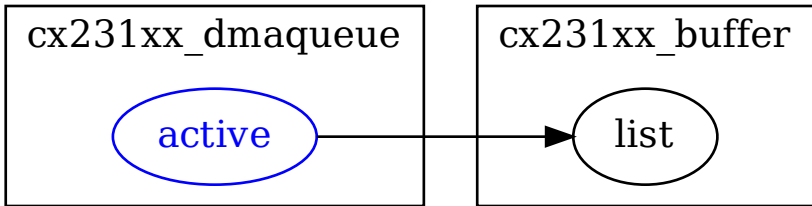
vmci_subscription@vmci_event.c

node

```
graph LR; subgraph var ["var@vmci_event.c"]; sub["subscriber_array"]; end; subgraph vmci_subscription ["vmci_subscription@vmci_event.c"]; node((node)); end; sub --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@vmci_event.c' contains an oval labeled 'subscriber_array' in blue text. On the right, a rectangular box labeled 'vmci_subscription@vmci_event.c' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'subscriber_array' oval to the left side of the 'node' oval, indicating that 'subscriber_array' points to 'node'.





ip_vs_dest_set@ip_vs_lblcr.c

list

ip_vs_dest_set_elem@ip_vs_lblcr.c

list



atmel_aes_drv@atmel-aes.c

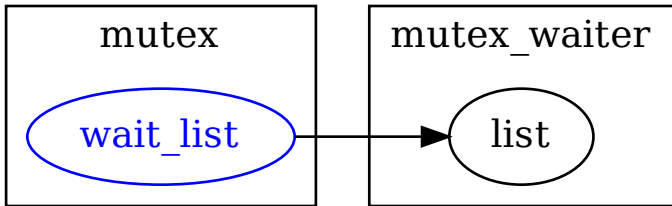
dev_list

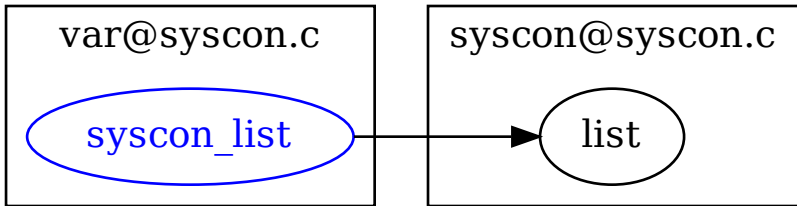
atmel_aes_dev@atmel-aes.c

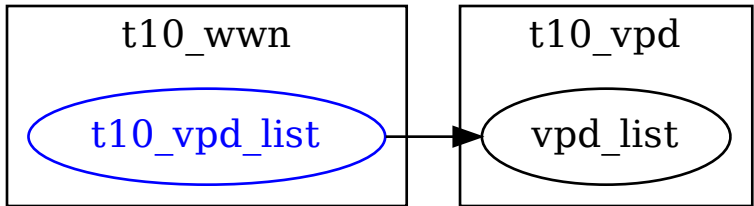
list

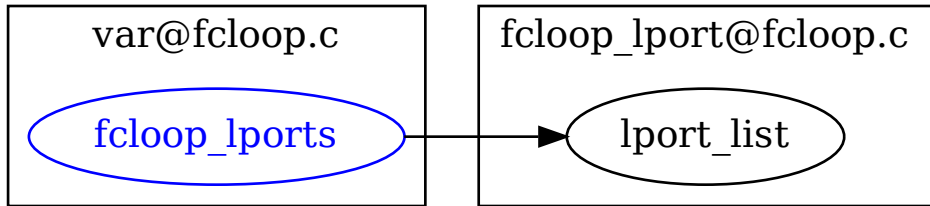
```
graph LR; subgraph Box1 [atmel_aes_drv@atmel-aes.c]; dev_list([dev_list]); end; subgraph Box2 [atmel_aes_dev@atmel-aes.c]; list([list]); end; dev_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'atmel_aes_drv@atmel-aes.c' contains a variable 'dev_list' enclosed in a blue oval. On the right, a box labeled 'atmel_aes_dev@atmel-aes.c' contains a variable 'list' enclosed in a black oval. A horizontal arrow points from the 'dev_list' oval to the 'list' oval, indicating that 'dev_list' points to 'list'.









var@zcrypt_ccamisc.c

cca_info_list

cca_info_list_entry@zcrypt_ccamisc.c

list

```
graph LR; A[cca_info_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@zcrypt_ccamisc.c' at the top and 'cca_info_list' in a blue oval below it. On the right, another rectangular box contains the text 'cca_info_list_entry@zcrypt_ccamisc.c' at the top. Inside this box, there is a black oval containing the text 'list'. A horizontal arrow points from the right side of the 'cca_info_list' oval to the left side of the 'list' oval, indicating that 'cca_info_list' points to 'list'.

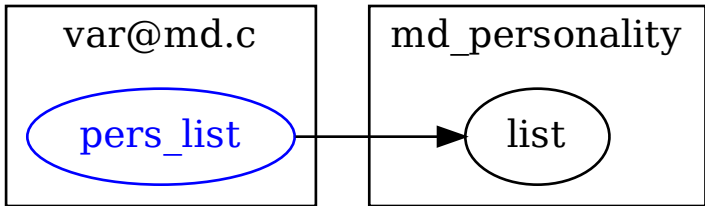
var@ipmi_ssif.c

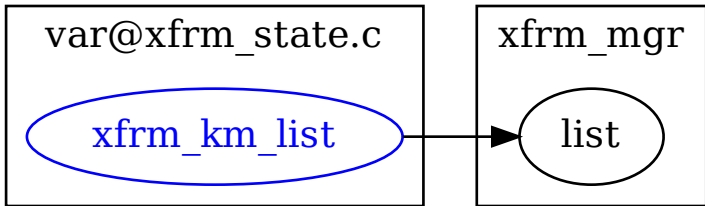
ssif_infos

ssif_addr_info@ipmi_ssif.c

link

A horizontal arrow points from the 'ssif_infos' oval in the left box to the 'link' oval in the right box.





owl_dma_txd@owl-dma.c

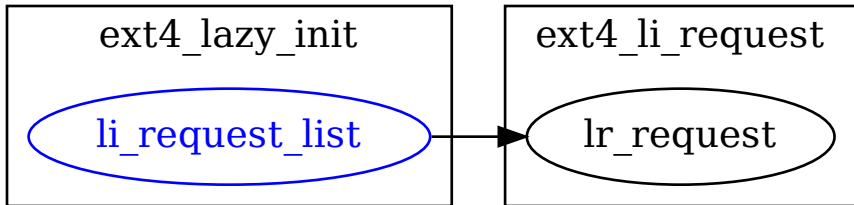
lli_list

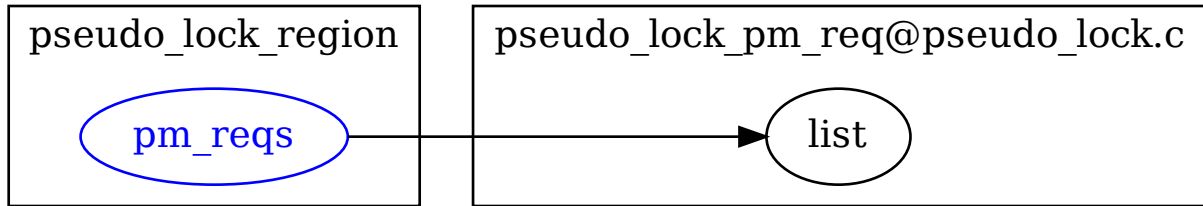
owl_dma_lli@owl-dma.c

node

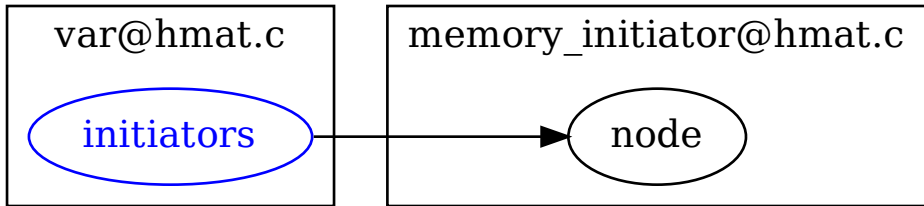
```
graph LR; subgraph tx_struct [owl_dma_txd@owl-dma.c]; lli_list((lli_list)); end; subgraph lli_struct [owl_dma_lli@owl-dma.c]; node((node)); end; lli_list --> node;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'owl_dma_txd@owl-dma.c' contains an oval labeled 'lli_list'. On the right, a box labeled 'owl_dma_lli@owl-dma.c' contains an oval labeled 'node'. A horizontal arrow points from the 'lli_list' oval to the 'node' oval, indicating that 'lli_list' points to 'node'.









var@thermal_core.c

thermal_governor_list

thermal_governor

governor_list

```
graph LR; A[thermal_governor_list] --> B[governor_list]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@thermal_core.c' contains an oval labeled 'thermal_governor_list' with a blue border. On the right, a box labeled 'thermal_governor' contains an oval labeled 'governor_list' with a black border. A black arrow points from the right side of the 'thermal_governor_list' oval to the left side of the 'governor_list' oval, indicating that 'thermal_governor_list' points to 'governor_list'.

var@dasd_eer.c

bufferlist

eerbuffer@dasd_eer.c

list

```
graph LR; subgraph Module1 [var@dasd_eer.c]; BL(bufferlist); end; subgraph Module2 [eerbuffer@dasd_eer.c]; L(list); end; BL --> L;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@dasd_eer.c' contains an oval labeled 'bufferlist' in blue text. On the right, a box labeled 'eerbuffer@dasd_eer.c' contains an oval labeled 'list' in black text. A horizontal arrow points from the 'bufferlist' oval to the 'list' oval, indicating that the 'bufferlist' variable in the first module points to the 'list' variable in the second module.

brcmf_chip_priv@chip.c

cores

brcmf_core_priv@chip.c

list



var@ipmi_si_intf.c

smi_infos

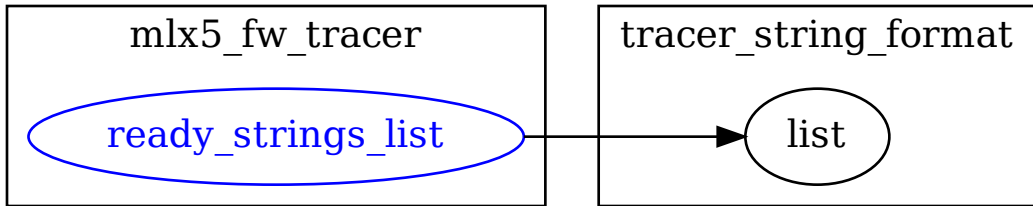
smi_info@ipmi_si_intf.c

link



```
graph LR; subgraph Box1 [var@ipmi_si_intf.c]; A((smi_infos)); end; subgraph Box2 [smi_info@ipmi_si_intf.c]; B((link)); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@ipmi_si_intf.c' and contains an oval labeled 'smi_infos' in blue text. The right box is labeled 'smi_info@ipmi_si_intf.c' and contains an oval labeled 'link'. A horizontal arrow points from the 'smi_infos' oval in the first box to the 'link' oval in the second box.



var@lowcomms.c

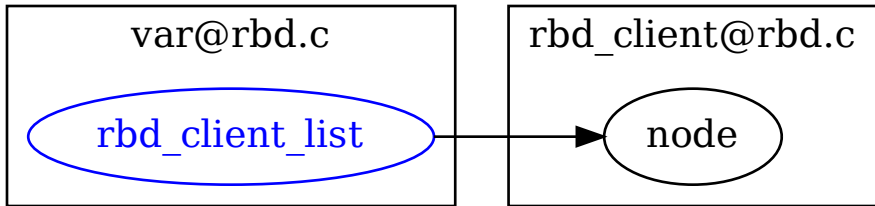
dln_node_addrs

dln_node_addr@lowcomms.c

list

```
graph LR; subgraph File1 [var@lowcomms.c]; A(dln_node_addrs); end; subgraph File2 [dln_node_addr@lowcomms.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@lowcomms.c`. Inside this box, the variable `dln_node_addrs` is enclosed in a blue oval. On the right, another rectangular box represents the source file `dln_node_addr@lowcomms.c`. Inside this box, the variable `list` is enclosed in a black oval. A horizontal arrow points from the blue oval `dln_node_addrs` to the black oval `list`, indicating that `dln_node_addrs` points to `list`.



brcmf_cfg80211_info

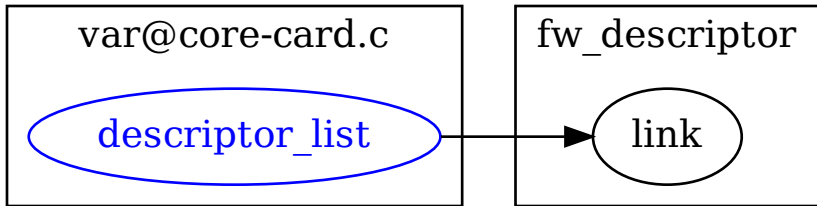
vif_list

brcmf_cfg80211_vif

list

```
graph LR; subgraph brcmf_cfg80211_info; vif_list; end; subgraph brcmf_cfg80211_vif; list; end; vif_list --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'brcmf_cfg80211_info' contains an oval labeled 'vif_list' with a blue border. On the right, a rectangular box labeled 'brcmf_cfg80211_vif' contains an oval labeled 'list'. A horizontal arrow points from the 'vif_list' oval to the 'list' oval, indicating that 'vif_list' points to 'list'.



stm32_hash_drv@stm32-hash.c

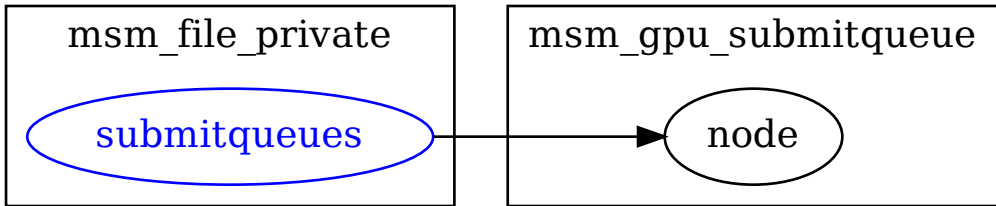
dev_list

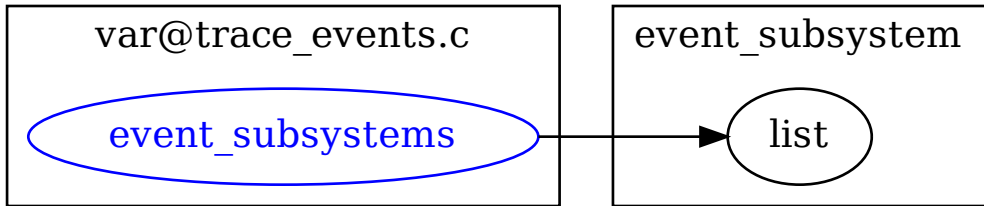
stm32_hash_dev@stm32-hash.c

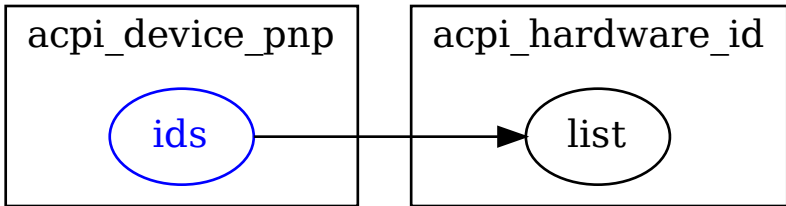
list

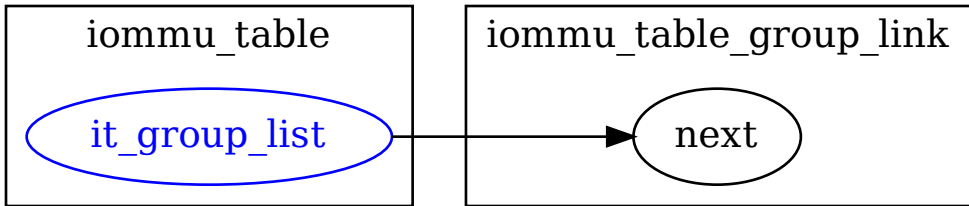
```
graph LR; subgraph Box1 [stm32_hash_drv@stm32-hash.c]; dev_list((dev_list)); end; subgraph Box2 [stm32_hash_dev@stm32-hash.c]; list((list)); end; dev_list --> list;
```

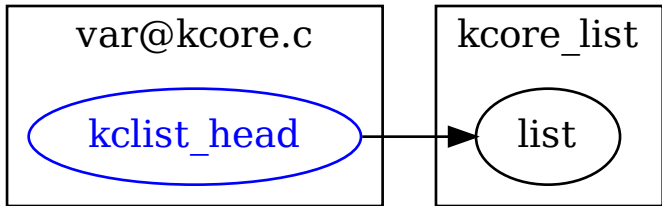
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'stm32_hash_drv@stm32-hash.c' contains a variable 'dev_list' which is circled in blue. On the right, a box labeled 'stm32_hash_dev@stm32-hash.c' contains a variable 'list' circled in black. A horizontal arrow points from the right side of the 'dev_list' oval to the left side of the 'list' oval, indicating that 'dev_list' points to 'list'.











var@matroxfb_base.c

matroxfb_driver_list

matroxfb_driver

node

```
graph LR; A[matroxfb_driver_list] --> B((node));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@matroxfb_base.c' contains an oval labeled 'matroxfb_driver_list' with a blue border. On the right, a rectangular box labeled 'matroxfb_driver' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'matroxfb_driver_list' oval to the left side of the 'node' oval, indicating that 'matroxfb_driver_list' points to 'node'.

tegra_xudc_ep@tegra-xudc.c

queue

tegra_xudc_request@tegra-xudc.c

list

```
graph LR; subgraph Box1 [tegra_xudc_ep@tegra-xudc.c]; queue((queue)); end; subgraph Box2 [tegra_xudc_request@tegra-xudc.c]; list((list)); end; queue --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'tegra_xudc_ep@tegra-xudc.c' and contains an oval labeled 'queue' with a blue border. The right box is labeled 'tegra_xudc_request@tegra-xudc.c' and contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'queue' oval to the 'list' oval, indicating a pointer relationship.

mt7621_pcie@pci-mt7621.c

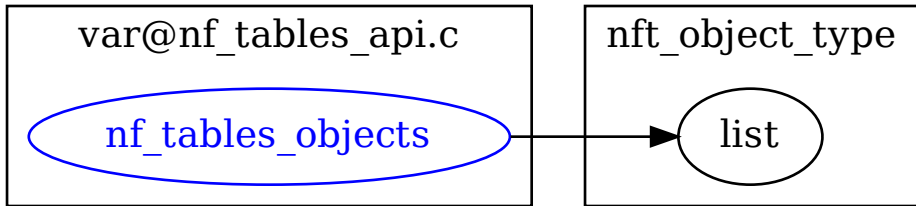
ports

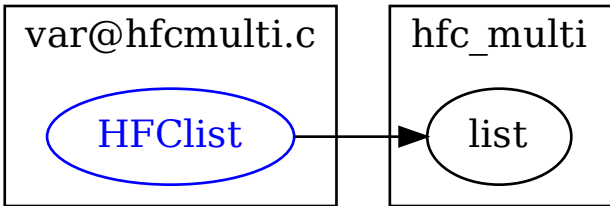
mt7621_pcie_port@pci-mt7621.c

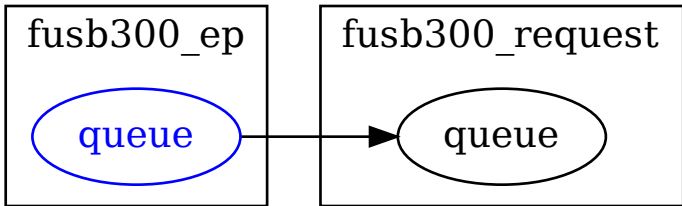
list

```
graph LR; A["mt7621_pcie@pci-mt7621.c  
ports"] --> B["mt7621_pcie_port@pci-mt7621.c  
list"]
```

The diagram illustrates a pointer relationship between two modules. On the left, a box represents the module `mt7621_pcie@pci-mt7621.c`, which contains a variable named `ports` (highlighted with a blue oval). On the right, a box represents the module `mt7621_pcie_port@pci-mt7621.c`, which contains a variable named `list` (highlighted with a black oval). A horizontal arrow points from the `ports` variable in the first module to the `list` variable in the second module, indicating that the `ports` variable points to the `list` variable.







dev_cgroup@device_cgroup.c

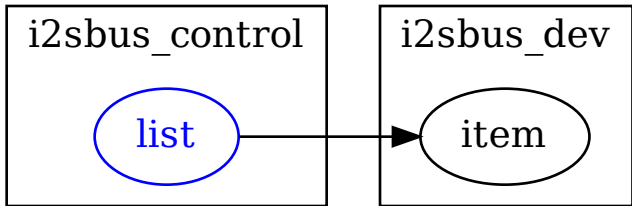
exceptions

dev_exception_item@device_cgroup.c

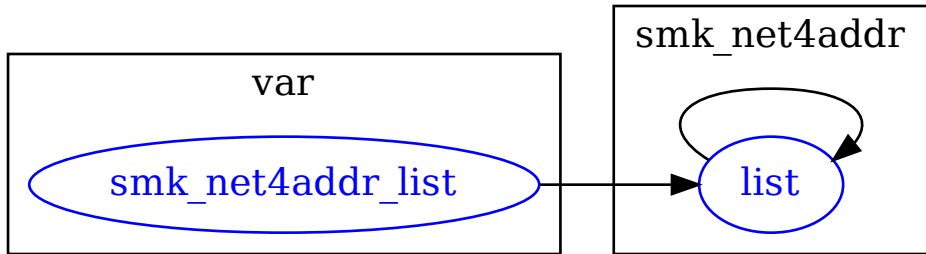
list

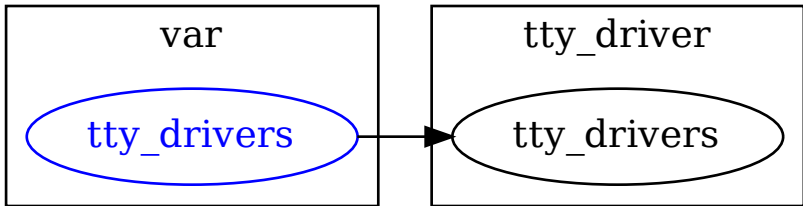
```
graph LR; A[dev_cgroup@device_cgroup.c] --> B[dev_exception_item@device_cgroup.c]; subgraph A; E([exceptions]); end; subgraph B; L([list]); end; E --> L;
```

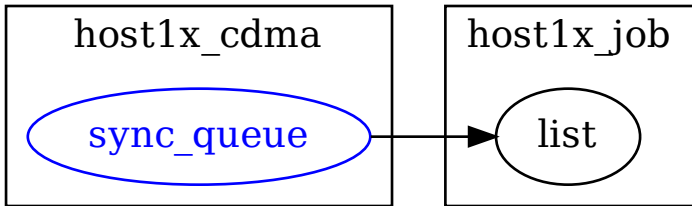
The diagram consists of two rectangular boxes. The left box contains the text 'dev_cgroup@device_cgroup.c' at the top and a blue oval containing the word 'exceptions' below it. The right box contains the text 'dev_exception_item@device_cgroup.c' at the top and a black oval containing the word 'list' below it. A horizontal arrow points from the right side of the 'exceptions' oval to the left side of the 'list' oval.

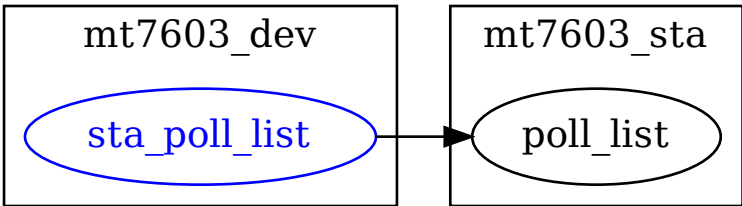












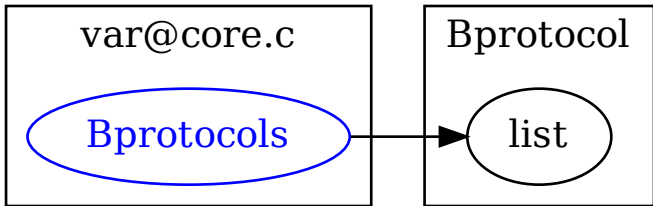
var@nx-842-powernv.c

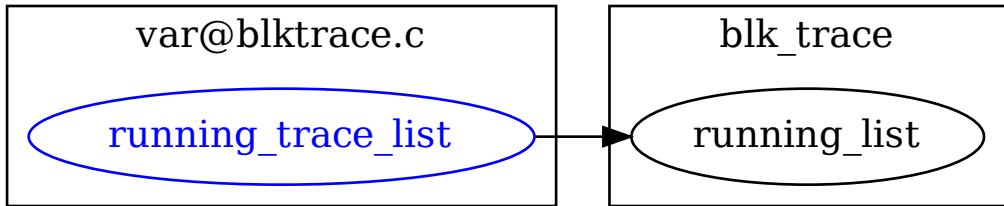
nx842_coproc

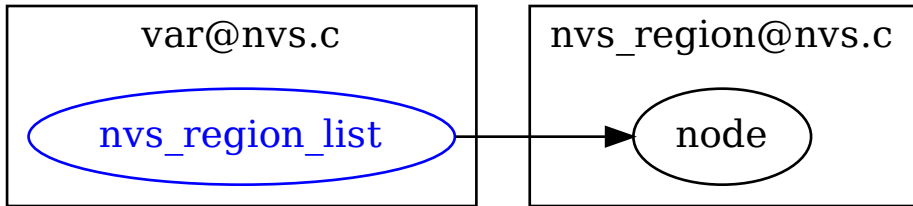
nx842_coproc@nx-842-powernv.c

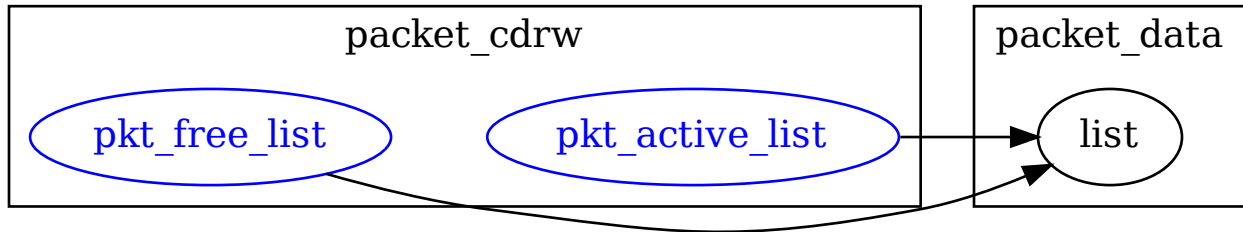
list

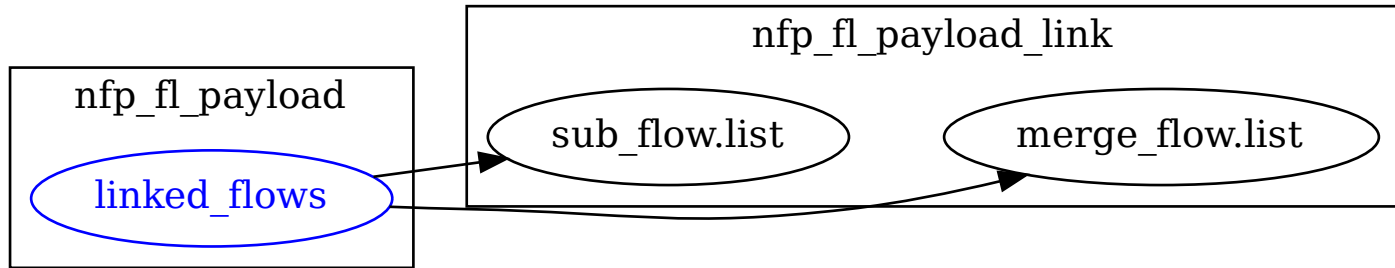
The diagram consists of two rectangular boxes. The left box contains the text 'var@nx-842-powernv.c' at the top and 'nx842_coproc' at the bottom, which is enclosed in a blue oval. The right box contains the text 'nx842_coproc@nx-842-powernv.c' at the top and 'list' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.











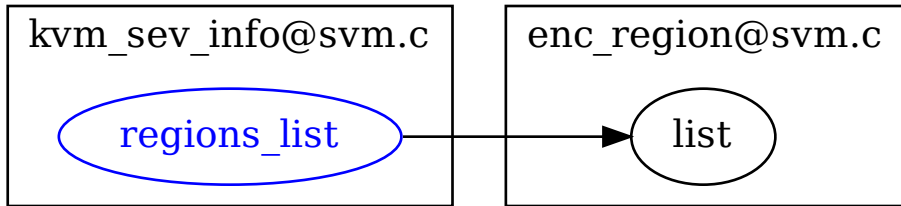
s3c_hsudc_ep@s3c-hsudc.c

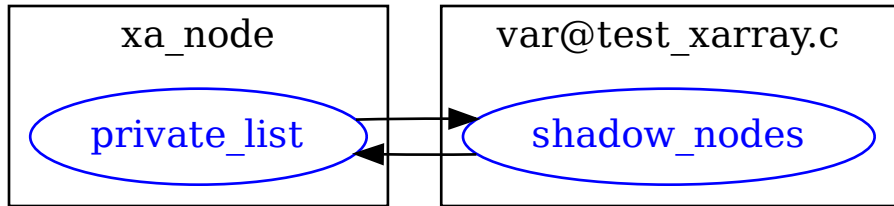
queue

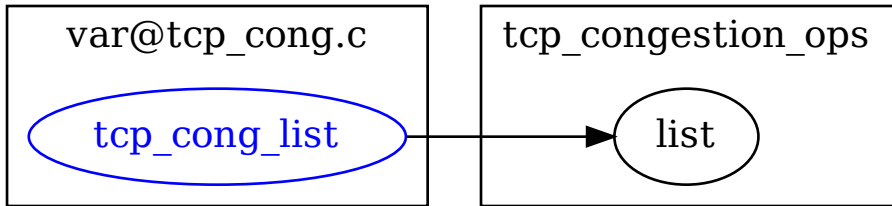
s3c_hsudc_req@s3c-hsudc.c

queue









var@s390_pci_hpc.c

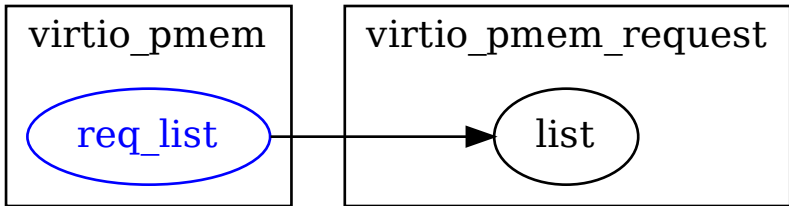
s390_hotplug_slot_list

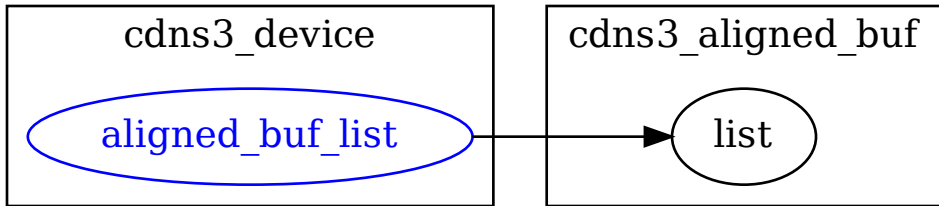
slot@s390_pci_hpc.c

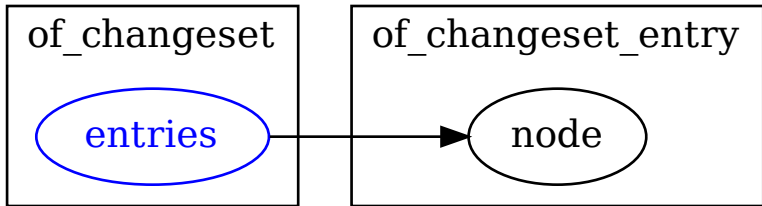
slot_list

```
graph LR; A["var@s390_pci_hpc.c  
s390_hotplug_slot_list"] --> B["slot@s390_pci_hpc.c  
slot_list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@s390_pci_hpc.c' and a blue oval containing 's390_hotplug_slot_list'. On the right, another rectangular box contains the text 'slot@s390_pci_hpc.c' and a black oval containing 'slot_list'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that the variable 's390_hotplug_slot_list' points to the variable 'slot_list'.







var@smscoreapi.c

g_smscore_devices

smscore_device_t

entry

```
graph LR; subgraph var_at_smscoreapi_c [var@smscoreapi.c]; g_smscore_devices; end; subgraph smscore_device_t_struct [smscore_device_t]; entry; end; g_smscore_devices --> entry;
```

The diagram illustrates a pointer relationship. On the left, a rectangular box labeled 'var@smscoreapi.c' contains an oval labeled 'g_smscore_devices' with a blue border. On the right, another rectangular box labeled 'smscore_device_t' contains an oval labeled 'entry' with a black border. A horizontal arrow points from the right side of the 'g_smscore_devices' oval to the left side of the 'entry' oval, indicating that 'g_smscore_devices' points to 'entry'.

cal_dmaqueue@cal.c

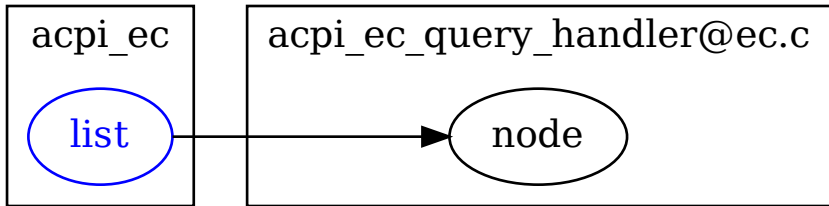
active

cal_buffer@cal.c

list

```
graph LR; subgraph "cal_dmaqueue@cal.c"; active((active)); end; subgraph "cal_buffer@cal.c"; list((list)); end; active --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'cal_dmaqueue@cal.c' and contains a blue oval labeled 'active'. The right box is labeled 'cal_buffer@cal.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'active' oval to the left side of the 'list' oval.

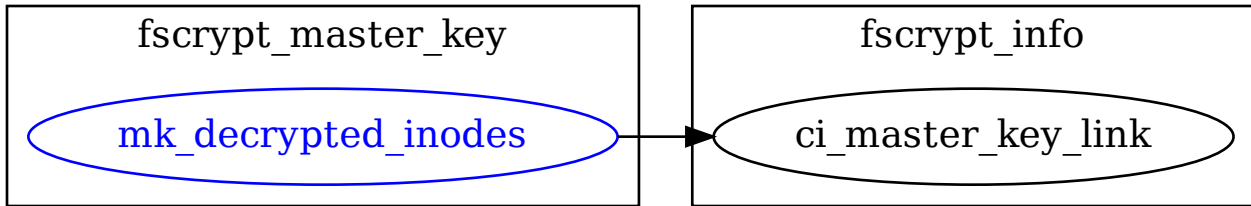


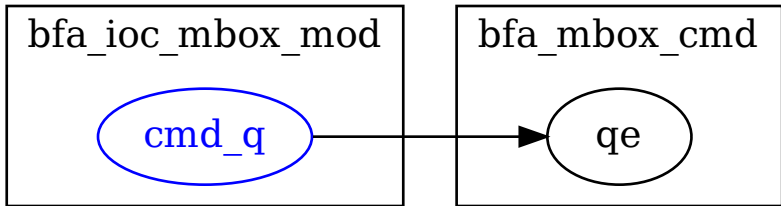
fscrypt_master_key

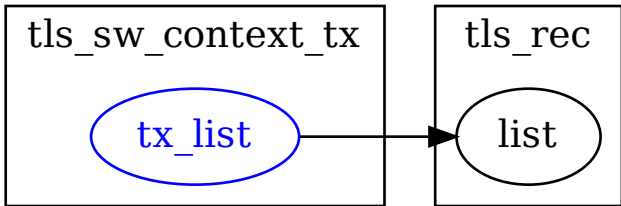
mk_decrypted_inodes

fscrypt_info

ci_master_key_link







stm32_cryp_list@stm32-cryp.c

dev_list

stm32_cryp@stm32-cryp.c

list



atmel_lcdfb_pdata

pwr_gpios

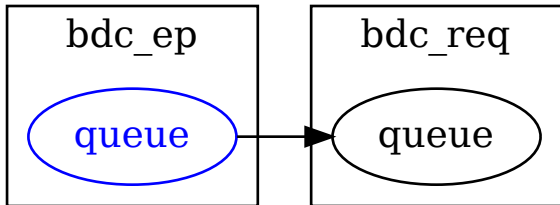
atmel_lcdfb_power_ctrl_gpio@atmel_lcdfb.c

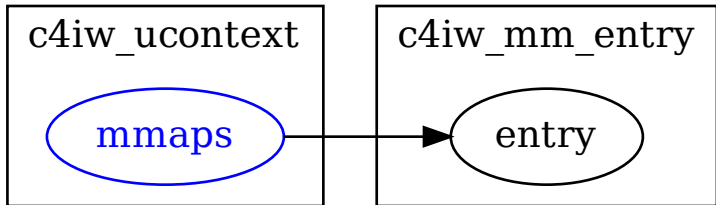
list

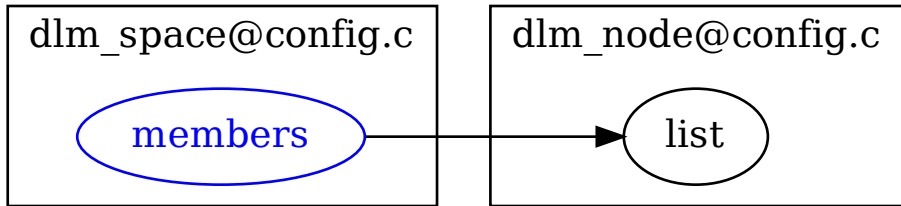


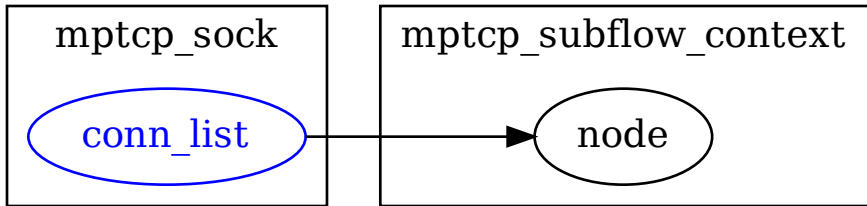
```
graph LR; subgraph Box1 [atmel_lcdfb_pdata]; pwr_gpios; end; subgraph Box2 [atmel_lcdfb_power_ctrl_gpio@atmel_lcdfb.c]; list; end; pwr_gpios --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'atmel_lcdfb_pdata' contains an oval labeled 'pwr_gpios'. On the right, a box labeled 'atmel_lcdfb_power_ctrl_gpio@atmel_lcdfb.c' contains an oval labeled 'list'. A horizontal arrow points from the 'pwr_gpios' oval to the 'list' oval, indicating that 'pwr_gpios' points to 'list'.









mlxsw_sp_fib@spectrum_router.c

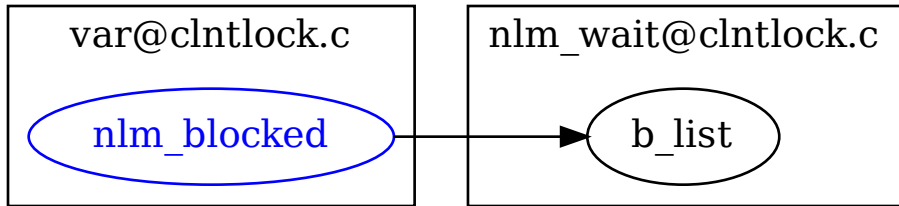
node_list

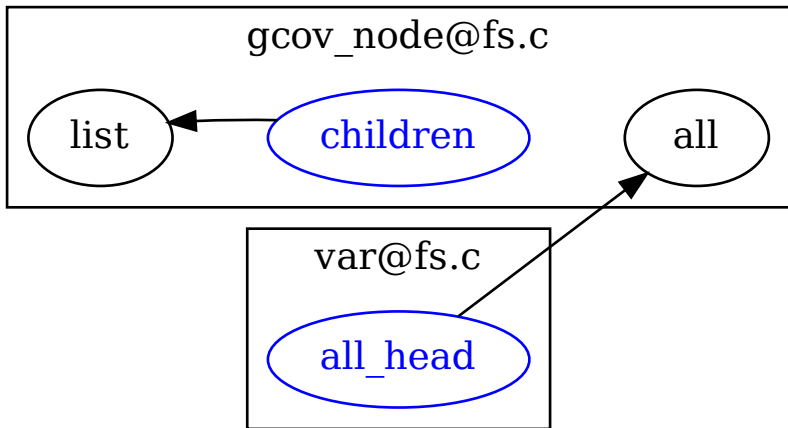
mlxsw_sp_fib_node@spectrum_router.c

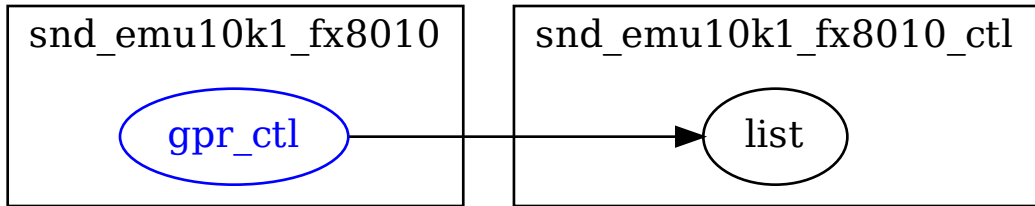
list

```
graph LR; A[mlxsw_sp_fib@spectrum_router.c] --> B((list)); B --- C[mlxsw_sp_fib_node@spectrum_router.c];
```

The diagram illustrates a pointer relationship between two code files. On the left, a box labeled 'mlxsw_sp_fib@spectrum_router.c' contains a blue oval labeled 'node_list'. On the right, a box labeled 'mlxsw_sp_fib_node@spectrum_router.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'node_list' oval to the 'list' oval, indicating that 'node_list' points to 'list'. Additionally, a horizontal line connects the two boxes, passing through the arrow.







sst_hsw@sst-haswell-ipc.c

stream_list

sst_hsw_stream@sst-haswell-ipc.c

node

```
graph LR; subgraph Box1 [ ]; direction TB; A[sst_hsw@sst-haswell-ipc.c]; B((stream_list)); end; subgraph Box2 [ ]; direction TB; C[sst_hsw_stream@sst-haswell-ipc.c]; D((node)); end; B --> D;
```

The diagram consists of two rectangular boxes. The left box contains the text 'sst_hsw@sst-haswell-ipc.c' at the top and an oval containing 'stream_list' below it. The right box contains the text 'sst_hsw_stream@sst-haswell-ipc.c' at the top and an oval containing 'node' below it. A horizontal arrow points from the right side of the 'stream_list' oval to the left side of the 'node' oval.

var@BusLogic.c

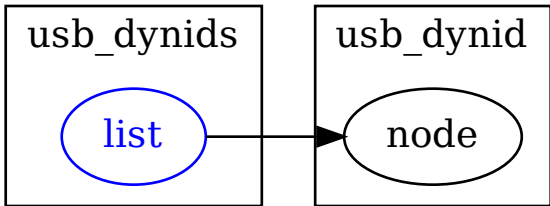
blogic_host_list

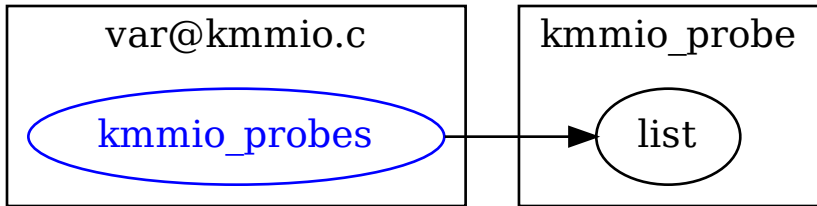
blogic_adapter

host_list

```
graph LR; A[blogic_host_list] --> B[host_list];
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@BusLogic.c' contains an oval labeled 'blogic_host_list' with a blue border. On the right, a rectangular box labeled 'blogic_adapter' contains an oval labeled 'host_list' with a black border. A horizontal arrow points from the right side of the 'blogic_host_list' oval to the left side of the 'host_list' oval, indicating that 'blogic_host_list' points to 'host_list'.





clusterip_net@ipt_CLUSTERIP.c

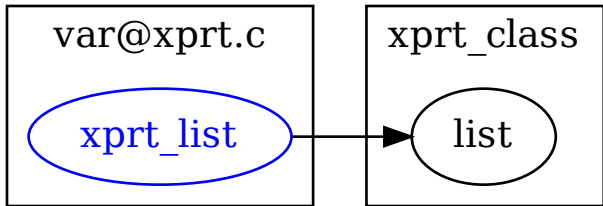
configs

clusterip_config@ipt_CLUSTERIP.c

list

```
graph LR; A[configs] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'clusterip_net@ipt_CLUSTERIP.c' contains a blue oval labeled 'configs'. On the right, a box labeled 'clusterip_config@ipt_CLUSTERIP.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'configs' oval to the 'list' oval, indicating that 'configs' points to 'list'.



var@cxgb3_offload.c

client_list

cxgb3_client

client_list

The diagram consists of two rectangular boxes. The left box is labeled 'var@cxgb3_offload.c' and contains a blue oval labeled 'client_list'. The right box is labeled 'cxgb3_client' and contains a black oval labeled 'client_list'. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box, indicating a pointer relationship.

var@security.c

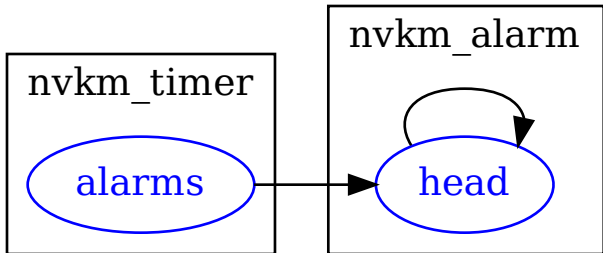
mad_agent_list

ib_mad_agent

mad_agent_sec_list

```
graph LR; A[var@security.c] --> B[ib_mad_agent]; A --- C(mad_agent_list); B --- D(mad_agent_sec_list); C --- D;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@security.c' contains an oval labeled 'mad_agent_list' with a blue border. On the right, a rectangular box labeled 'ib_mad_agent' contains an oval labeled 'mad_agent_sec_list' with a black border. A black arrow points from the right side of the 'mad_agent_list' oval to the left side of the 'mad_agent_sec_list' oval, indicating that the variable 'mad_agent_list' points to the variable 'mad_agent_sec_list'.



counter_device_attr_group

attr_list

counter_device_attr@counter.c

1



var@qemu_fw_cfg.c

fw_cfg_entry_cache

fw_cfg_sysfs_entry@qemu_fw_cfg.c

list

```
graph LR; A[fw_cfg_entry_cache] --> B((list));
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@qemu_fw_cfg.c' at the top and 'fw_cfg_entry_cache' at the bottom, which is enclosed in a blue oval. The right box contains the text 'fw_cfg_sysfs_entry@qemu_fw_cfg.c' at the top and 'list' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.

kbl_rt5663_private@kbl_rt5663_max98927.c

hdmi_pcm_list

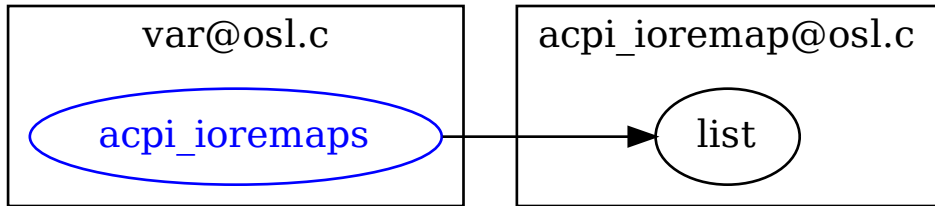
kbl_hdmi_pcm@kbl_rt5663_max98927.c

head



```
graph LR; subgraph LeftBox [kbl_rt5663_private@kbl_rt5663_max98927.c]; hdmi_pcm_list; end; subgraph RightBox [kbl_hdmi_pcm@kbl_rt5663_max98927.c]; head; end; hdmi_pcm_list --> head;
```



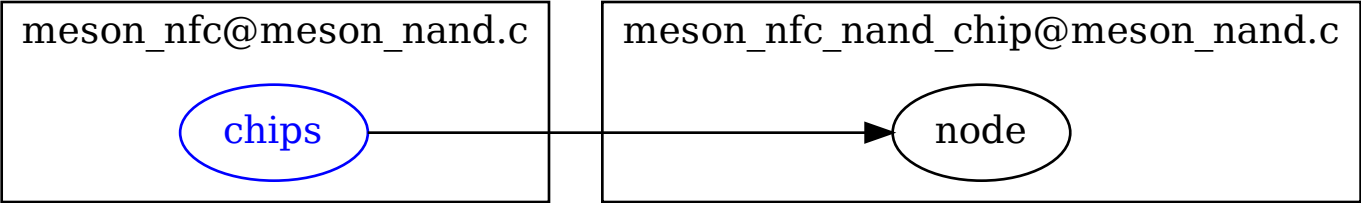


meson_nfc@meson_nand.c

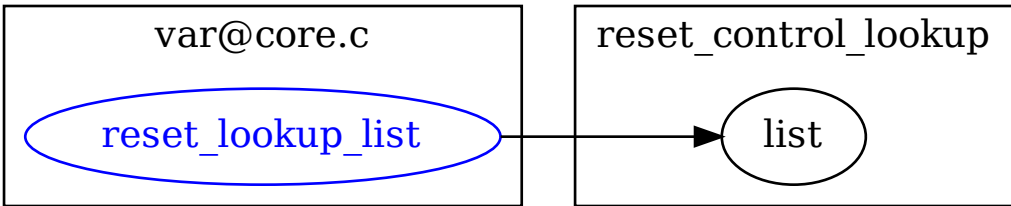
chips

meson_nfc_nand_chip@meson_nand.c

node



```
graph LR; subgraph Box1 [meson_nfc@meson_nand.c]; chips((chips)); end; subgraph Box2 [meson_nfc_nand_chip@meson_nand.c]; node((node)); end; chips --> node;
```



atk_data@asus_atk0110.c

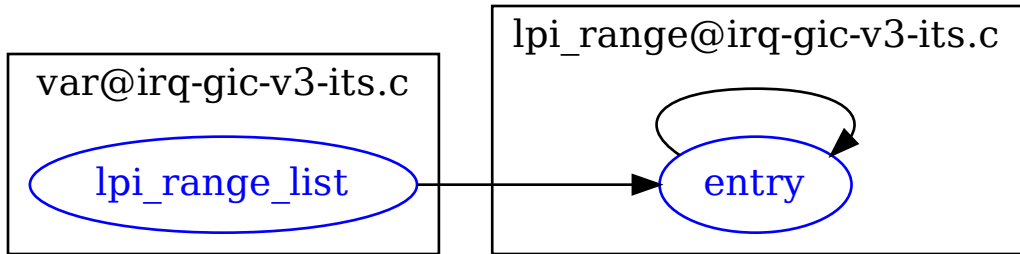
sensor_list

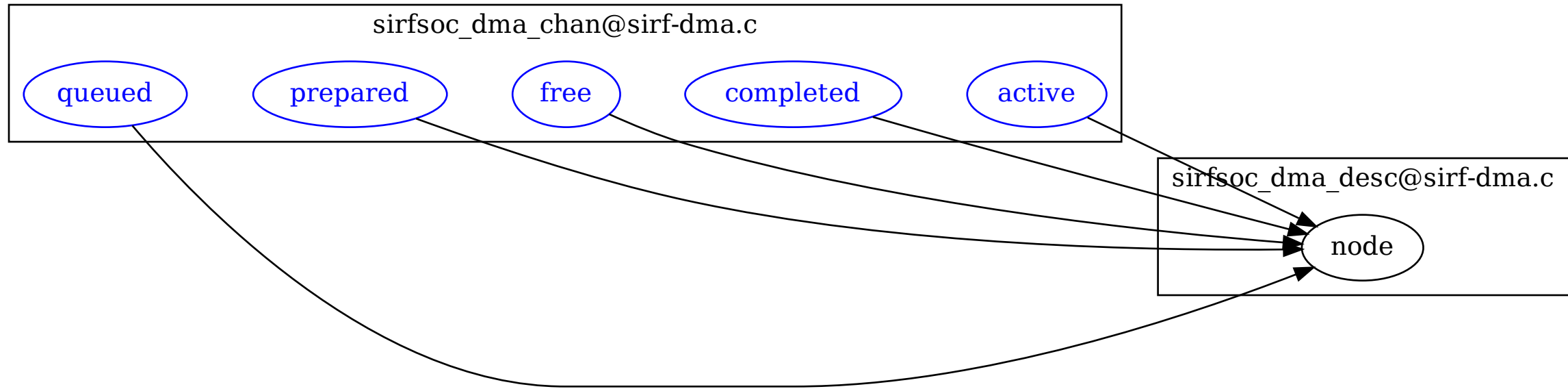
atk_sensor_data@asus_atk0110.c

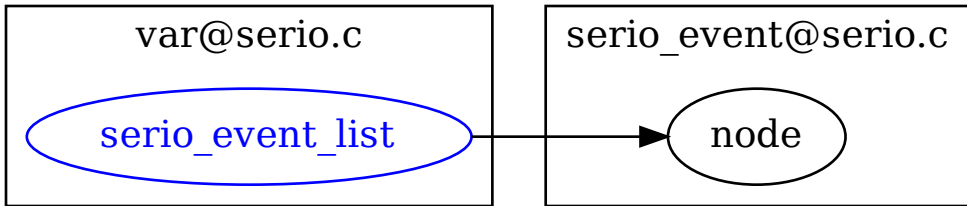
list

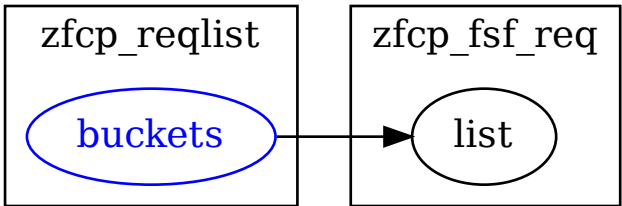
```
graph LR; subgraph Box1 [atk_data@asus_atk0110.c]; SL(sensor_list); end; subgraph Box2 [atk_sensor_data@asus_atk0110.c]; L((list)); end; SL --> L;
```

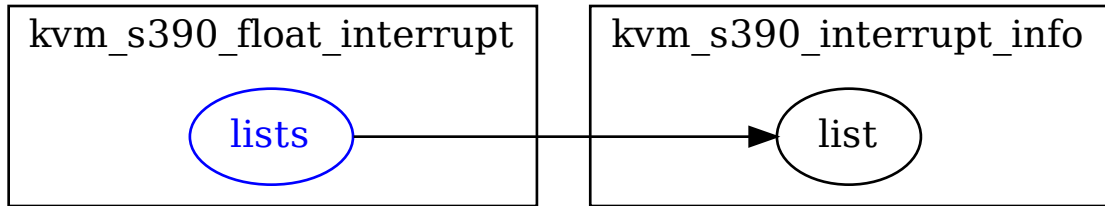
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'atk_data@asus_atk0110.c'. Inside this box, the text 'sensor_list' is enclosed in a blue oval. On the right, another rectangular box contains the text 'atk_sensor_data@asus_atk0110.c'. Inside this box, the text 'list' is enclosed in a black oval. A horizontal arrow points from the blue oval 'sensor_list' in the first box to the black oval 'list' in the second box, indicating that 'sensor_list' points to 'list'.

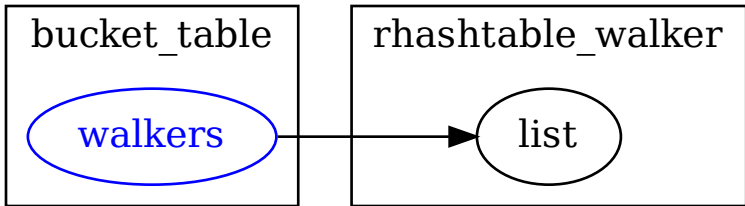


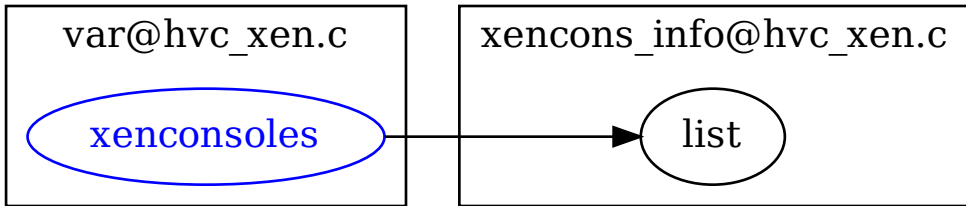


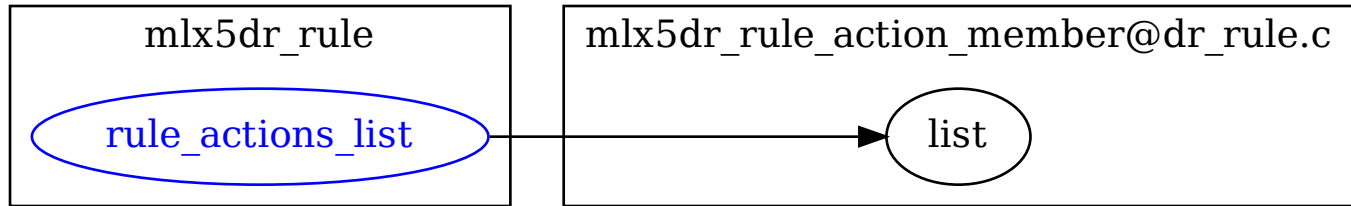


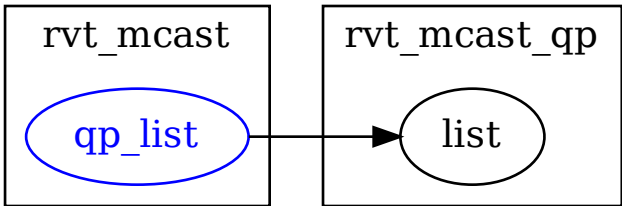


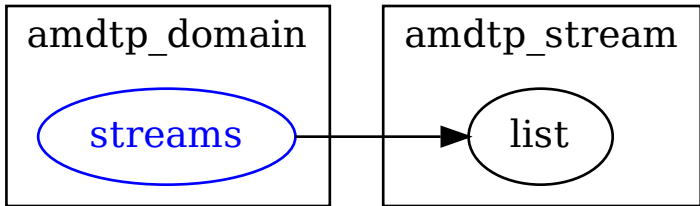


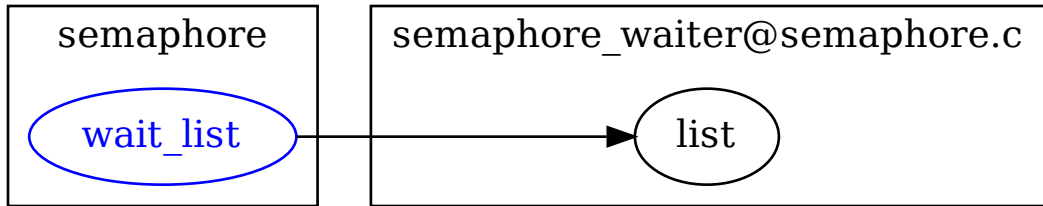












cpcap_charger_ddata@cpcap-charger.c

irq_list

cpcap_interrupt_desc@cpcap-charger.c

node

```
graph LR; A[cpcap_charger_ddata@cpcap-charger.c] -- irq_list --> B[cpcap_interrupt_desc@cpcap-charger.c]; B -- node --> C[ ];
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'cpcap_charger_ddata@cpcap-charger.c' contains an oval labeled 'irq_list'. On the right, a box labeled 'cpcap_interrupt_desc@cpcap-charger.c' contains an oval labeled 'node'. A horizontal arrow points from the 'irq_list' oval to the 'node' oval, indicating that 'irq_list' points to 'node'.

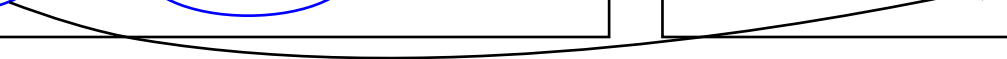
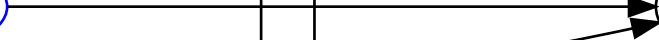
background_tracker@dm-cache-background-tracker.c

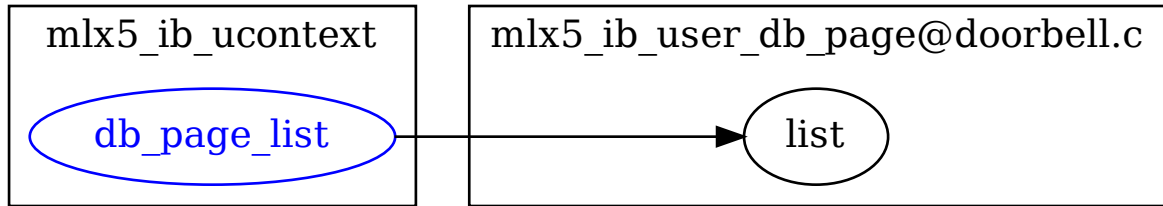
queued

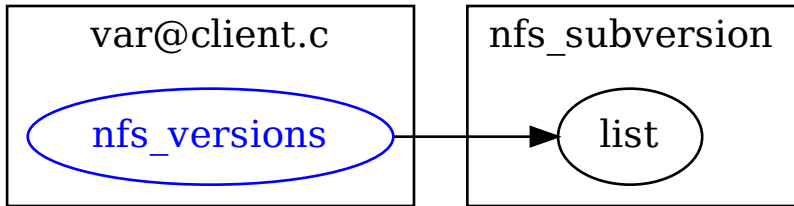
issued

bt_work@dm-cache-background-tracker.c

list







glink_channel@qcom_glink_native.c

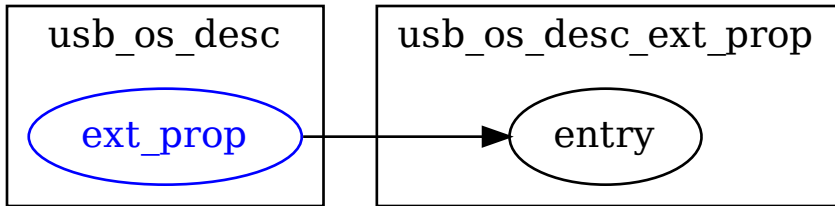
done_intents

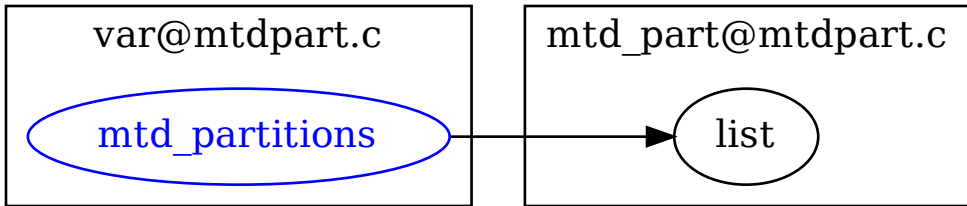
glink_core_rx_intent@qcom_glink_native.c

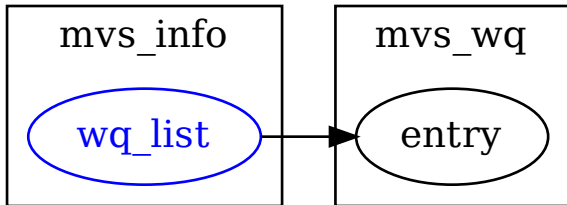
node

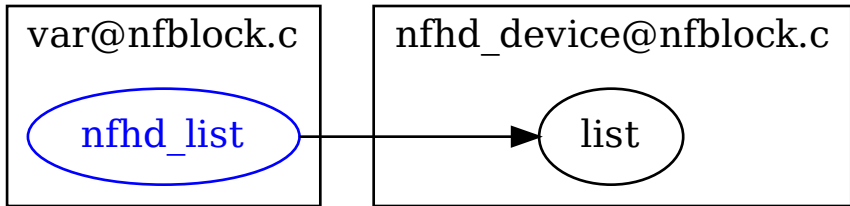
```
graph LR; A(glink_channel@qcom_glink_native.c) --> B(glink_core_rx_intent@qcom_glink_native.c); subgraph A; C(done_intents); end; subgraph B; D(node); end; C --> D;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'glink_channel@qcom_glink_native.c' contains an oval labeled 'done_intents'. On the right, a box labeled 'glink_core_rx_intent@qcom_glink_native.c' contains an oval labeled 'node'. A horizontal arrow points from the 'done_intents' oval to the 'node' oval, indicating that 'done_intents' points to 'node'.









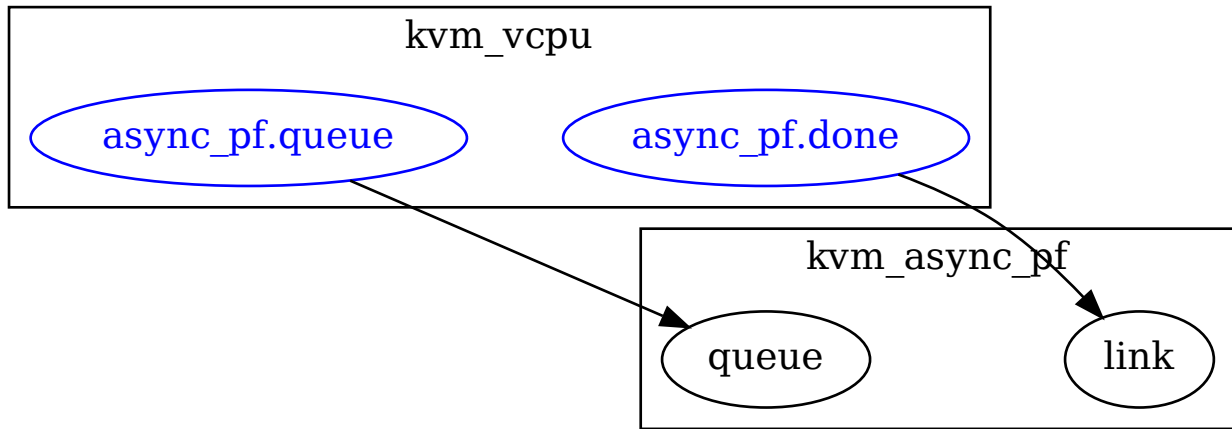
var@cpufreq.c

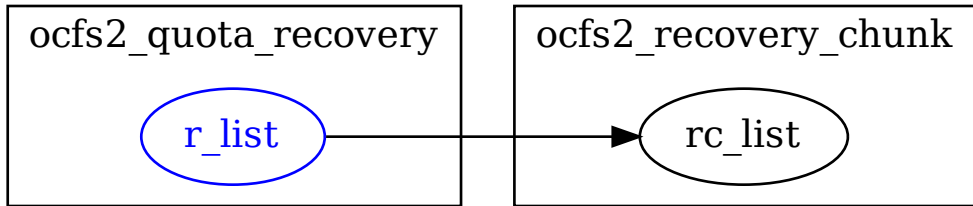
cpufreq_governor_list

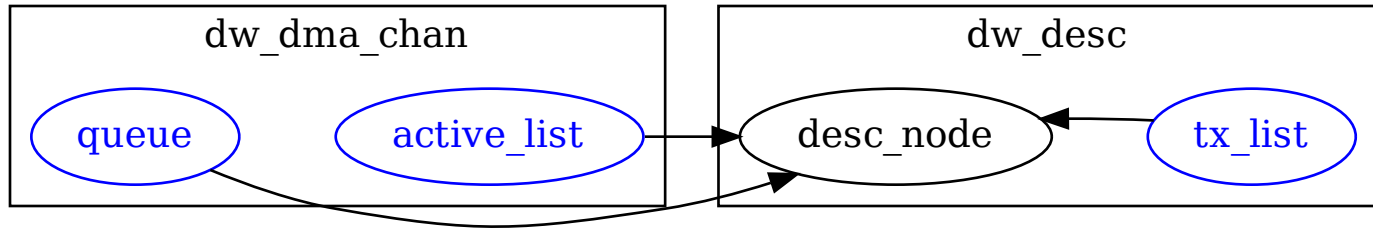
cpufreq_governor

governor_list









var@fw-management.c

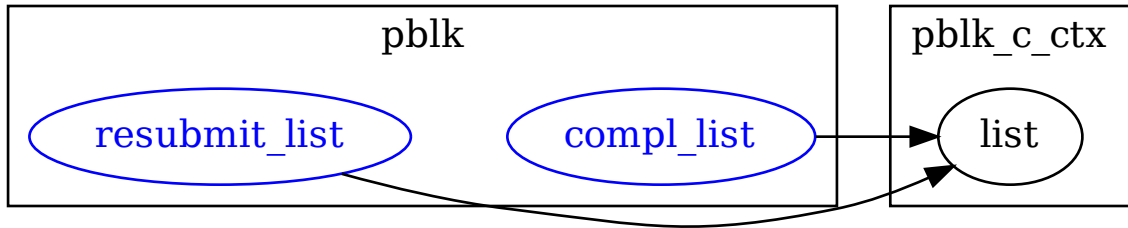
fw_mgmt_list

fw_mgmt@fw-management.c

node

```
graph LR; subgraph Box1 [var@fw-management.c]; fw_mgmt_list; end; subgraph Box2 [fw_mgmt@fw-management.c]; node; end; fw_mgmt_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@fw-management.c' contains an oval labeled 'fw_mgmt_list' with a blue border. On the right, a rectangular box labeled 'fw_mgmt@fw-management.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'fw_mgmt_list' oval to the left side of the 'node' oval, indicating that 'fw_mgmt_list' points to 'node'.



var@scsi_transport_iscsi.c

sesslist

iscsi_cls_session

sess_list

```
graph LR; A[var@scsi_transport_iscsi.c] --> B[iscsi_cls_session]; subgraph A; A1(sesslist); end; subgraph B; B1(sess_list); end; A1 --> B1;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@scsi_transport_iscsi.c' contains a blue oval labeled 'sesslist'. On the right, a box labeled 'iscsi_cls_session' contains a black oval labeled 'sess_list'. A horizontal arrow points from the 'sesslist' oval to the 'sess_list' oval, indicating that the 'sesslist' variable points to the 'sess_list' variable.

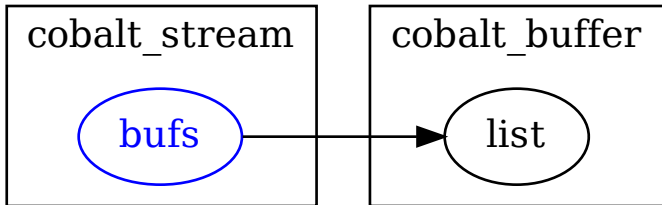
var@of-dma.c

of_dma_list



of_dma

of_dma_controllers



vimc_cap_device@vimc-capture.c

buf_list

vimc_cap_buffer@vimc-capture.c

list

```
graph LR; subgraph "vimc_cap_device@vimc-capture.c"; buf_list; end; subgraph "vimc_cap_buffer@vimc-capture.c"; list; end; buf_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'vimc_cap_device@vimc-capture.c' contains a variable 'buf_list' enclosed in a blue oval. On the right, a box labeled 'vimc_cap_buffer@vimc-capture.c' contains a variable 'list' enclosed in a black oval. A horizontal arrow points from the 'buf_list' oval to the 'list' oval, indicating that 'buf_list' points to 'list'.

ttm_object_file@ttm_object.c

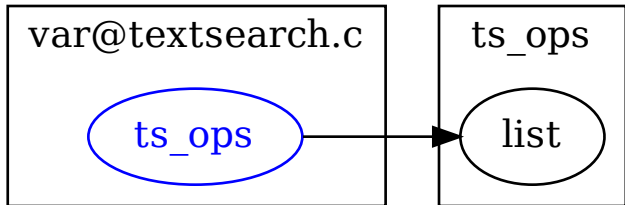
ref_list

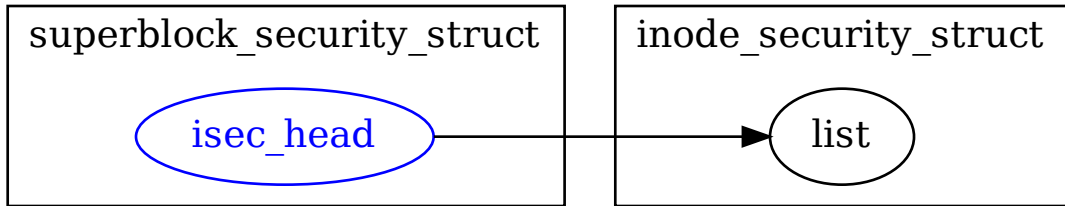
ttm_ref_object@ttm_object.c

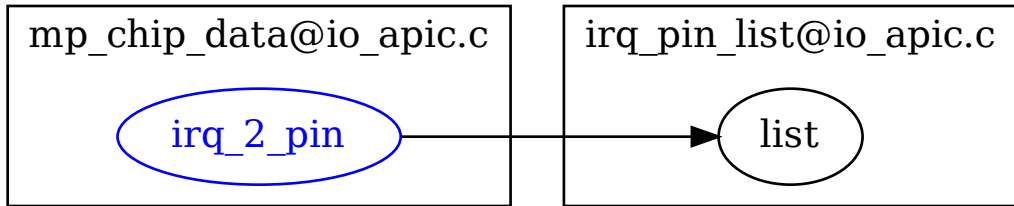
head

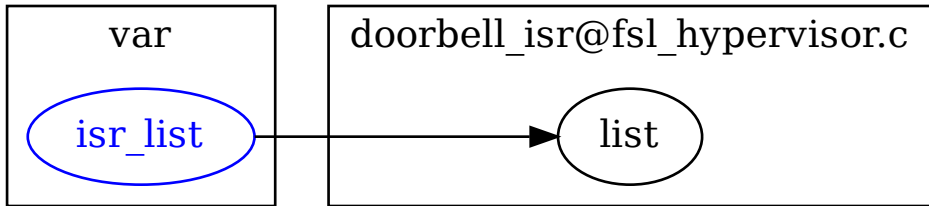
```
graph LR; A[ttm_object_file@ttm_object.c] --> B[ttm_ref_object@ttm_object.c]; subgraph A; R([ref_list]); end; subgraph B; H([head]); end; R --> H;
```

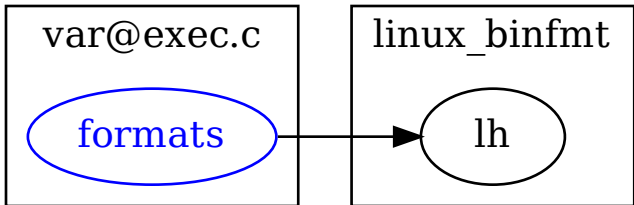
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'ttm_object_file@ttm_object.c' contains an oval labeled 'ref_list'. On the right, a box labeled 'ttm_ref_object@ttm_object.c' contains an oval labeled 'head'. A horizontal arrow points from the 'ref_list' oval to the 'head' oval, indicating that the 'ref_list' variable in the first module points to the 'head' variable in the second module.

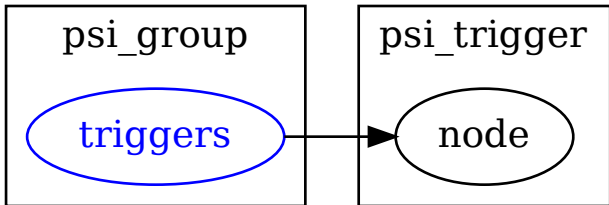












mon_reader_text@mon_text.c

e_list

mon_event_text@mon_text.c

e_link



var@md.c

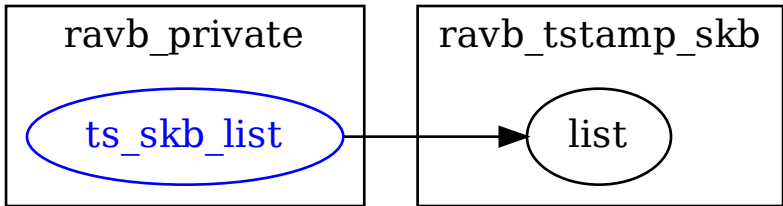
all_detected_devices

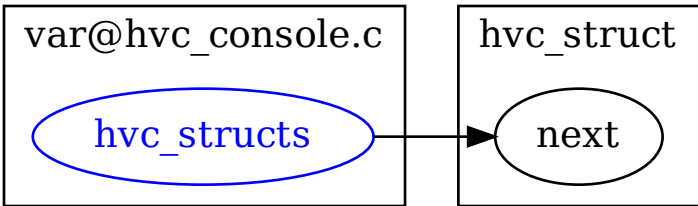
detected_devices_node@md.c

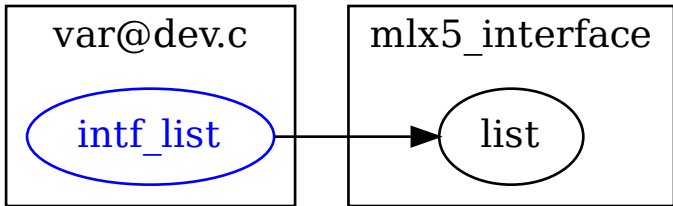
list

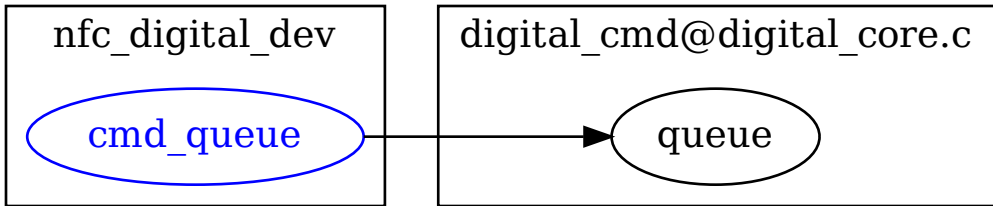
```
graph LR; A[all_detected_devices] --> B((list));
```

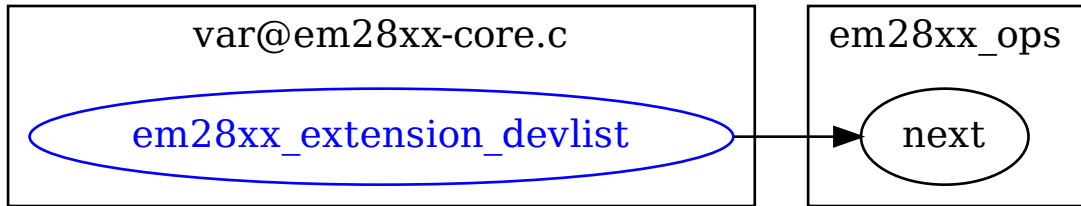
The diagram consists of two rectangular boxes. The left box is labeled 'var@md.c' and contains an oval labeled 'all_detected_devices' in blue text. The right box is labeled 'detected_devices_node@md.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'all_detected_devices' oval to the left side of the 'list' oval, indicating a pointer relationship.

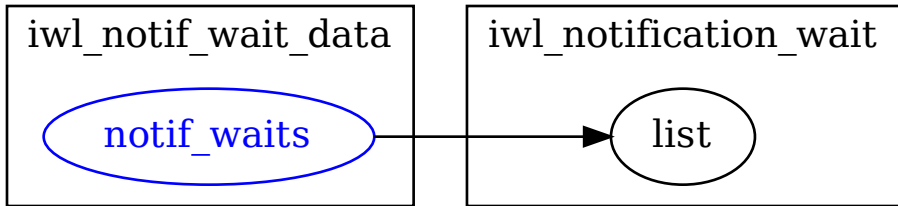


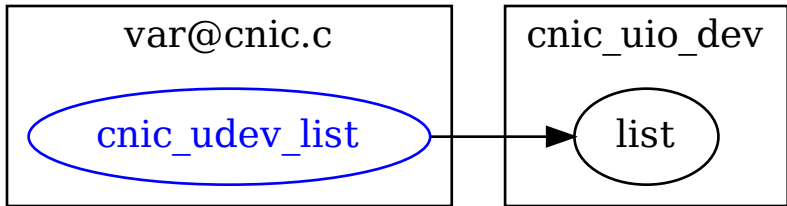












var@omap-mailbox.c

omap_mbox_devices

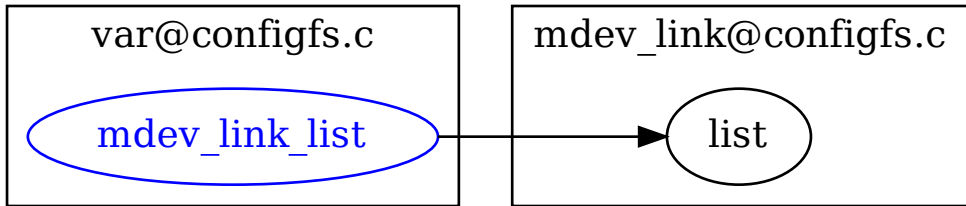
omap_mbox_device@omap-mailbox.c

elem

```
graph LR; subgraph var [var@omap-mailbox.c]; omap_mbox_devices; end; subgraph omap_mbox_device [omap_mbox_device@omap-mailbox.c]; elem; end; omap_mbox_devices --> elem;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@omap-mailbox.c' contains a blue oval labeled 'omap_mbox_devices'. On the right, a box labeled 'omap_mbox_device@omap-mailbox.c' contains an oval labeled 'elem'. A horizontal arrow points from the 'omap_mbox_devices' oval to the 'elem' oval, indicating that the 'omap_mbox_devices' variable points to the 'elem' element.





var@lapbether.c

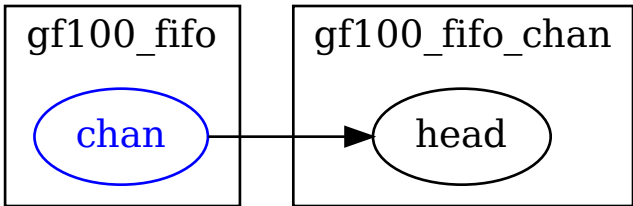
lapbeth_devices

lapbethdev@lapbether.c

node

```
graph LR; subgraph Box1 [var@lapbether.c]; A([lapbeth_devices]); end; subgraph Box2 [lapbethdev@lapbether.c]; B([node]); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@lapbether.c' and contains an oval labeled 'lapbeth_devices' in blue text. The right box is labeled 'lapbethdev@lapbether.c' and contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'lapbeth_devices' oval to the left side of the 'node' oval.



iommu_dma_cookie@dma-iommu.c

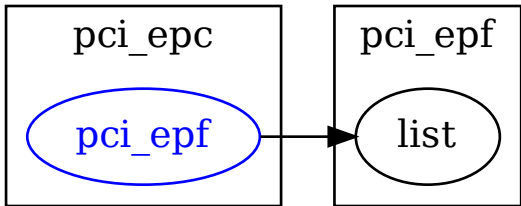
msi_page_list

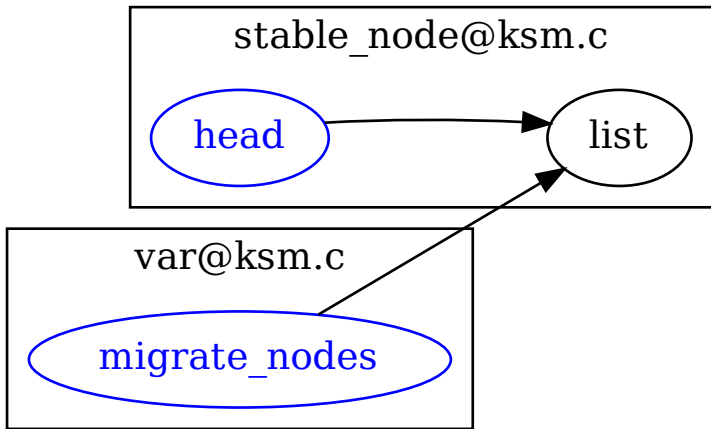
iommu_dma_msi_page@dma-iommu.c

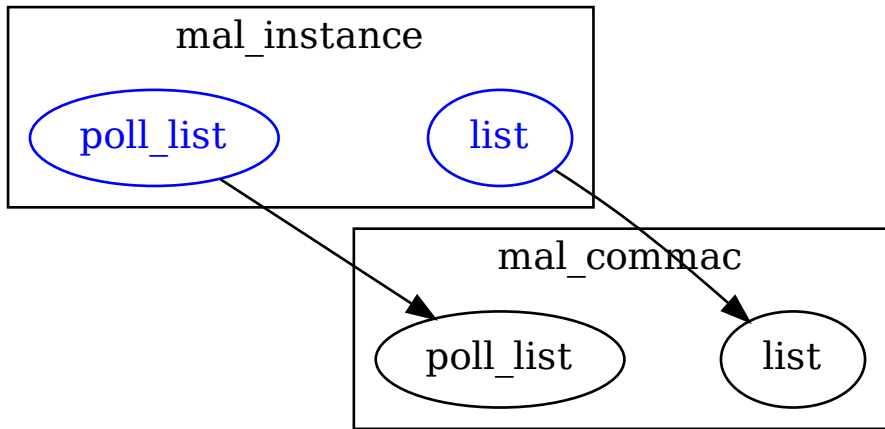
list

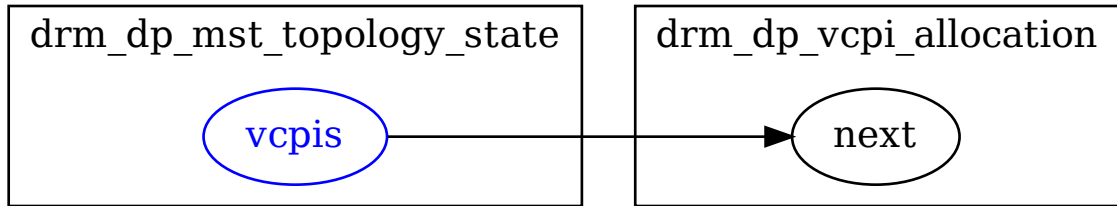
```
graph LR; A[iommu_dma_cookie@dma-iommu.c] --> B[iommu_dma_msi_page@dma-iommu.c]; B --> C((list));
```

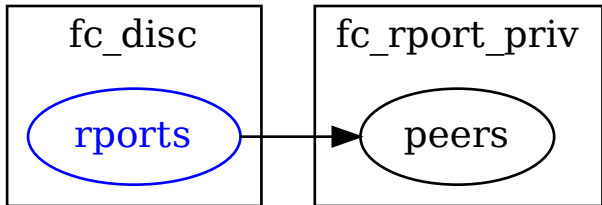
The diagram illustrates a pointer relationship between two iommu structures. On the left, a box labeled 'iommu_dma_cookie@dma-iommu.c' contains a blue oval labeled 'msi_page_list'. On the right, a box labeled 'iommu_dma_msi_page@dma-iommu.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'msi_page_list' oval to the 'list' oval, indicating that the 'msi_page_list' pointer in the first structure points to the 'list' in the second structure.











dummy_hcd@dummy_hcd.c

urbp_list

urbp@dummy_hcd.c

urbp_list

var@ftrace_free_mem

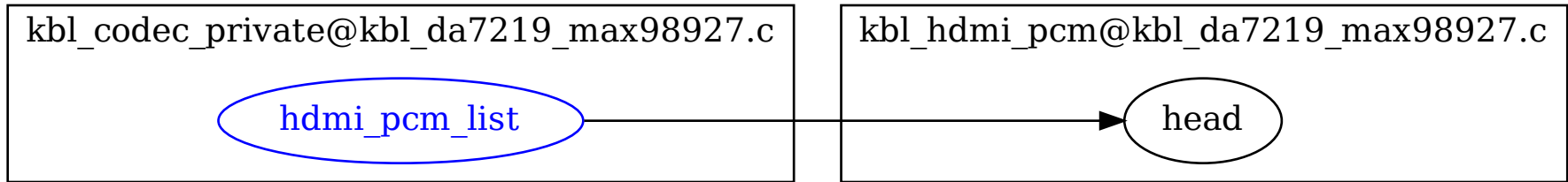
clear_hash

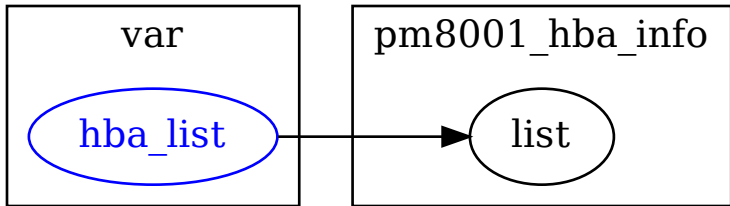
ftrace_init_func@ftrace.c

list

```
graph LR; subgraph Box1 [var@ftrace_free_mem]; CH((clear_hash)); end; subgraph Box2 [ftrace_init_func@ftrace.c]; L((list)); end; CH --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@ftrace_free_mem' and contains a blue oval labeled 'clear_hash'. The right box is labeled 'ftrace_init_func@ftrace.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'clear_hash' oval to the left side of the 'list' oval, indicating a pointer or reference relationship.





var@crash_dump.c

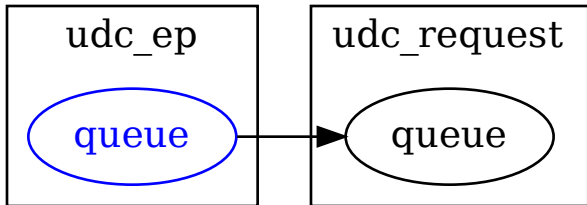
dump_save_areas

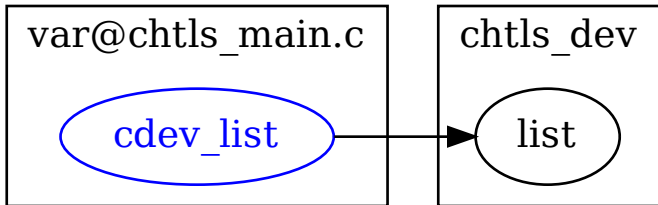
save_area@crash_dump.c

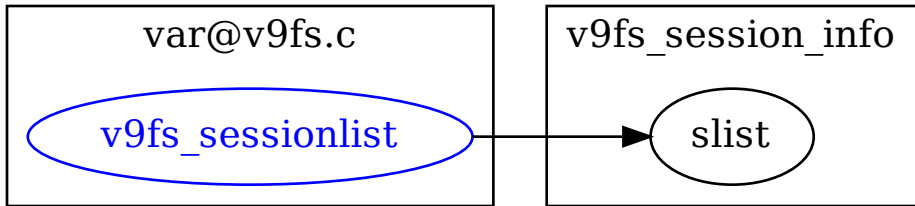
list

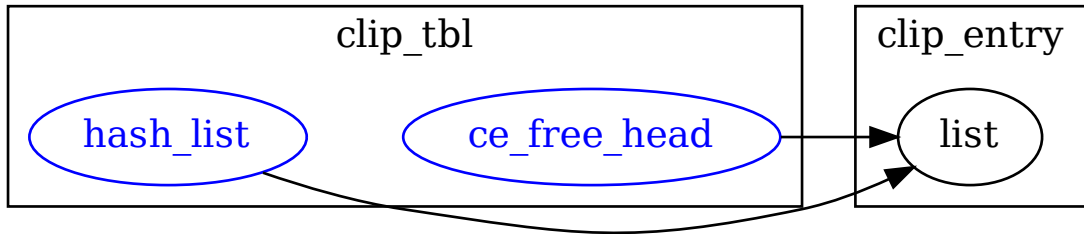
```
graph LR; subgraph var ["var@crash_dump.c"]; dump_save_areas; end; subgraph save_area ["save_area@crash_dump.c"]; list; end; dump_save_areas --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@crash_dump.c' contains an oval labeled 'dump_save_areas' with a blue border. On the right, a rectangular box labeled 'save_area@crash_dump.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dump_save_areas' oval to the 'list' oval, indicating that 'dump_save_areas' points to 'list'.









sur40_state@sur40.c

buf_list

sur40_buffer@sur40.c

list

```
graph LR; subgraph "sur40_state@sur40.c"; buf_list; end; subgraph "sur40_buffer@sur40.c"; list; end; buf_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'sur40_state@sur40.c' contains an oval labeled 'buf_list'. On the right, a box labeled 'sur40_buffer@sur40.c' contains an oval labeled 'list'. A horizontal arrow points from the 'buf_list' oval to the 'list' oval, indicating that 'buf_list' points to 'list'.

hifn_device@hifn_795x.c

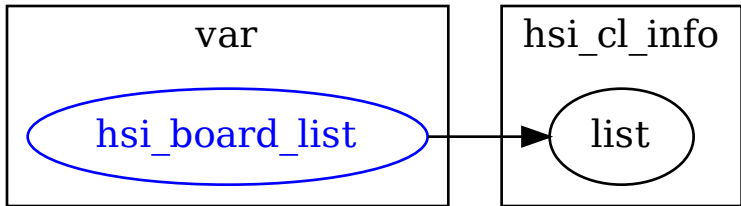
alg_list

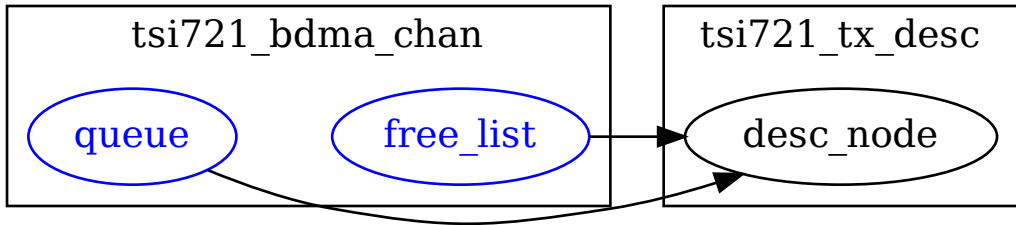
hifn_crypto_alg@hifn_795x.c

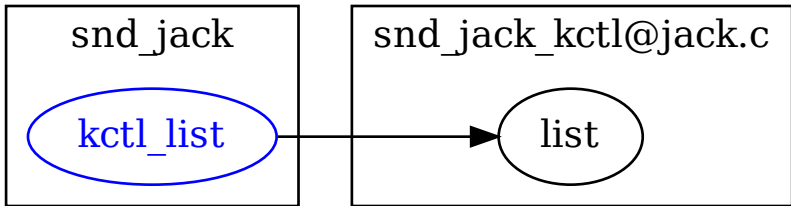
entry

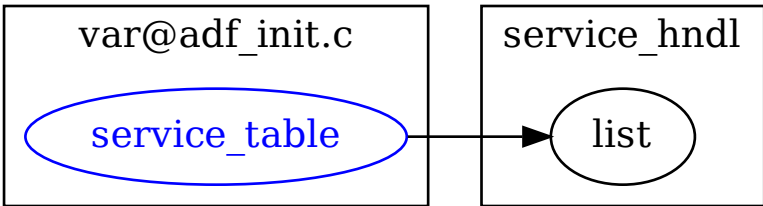
```
graph LR; A(hifn_device@hifn_795x.c) --> B(hifn_crypto_alg@hifn_795x.c); subgraph A; C([alg_list]); end; subgraph B; D([entry]); end; C --> D;
```

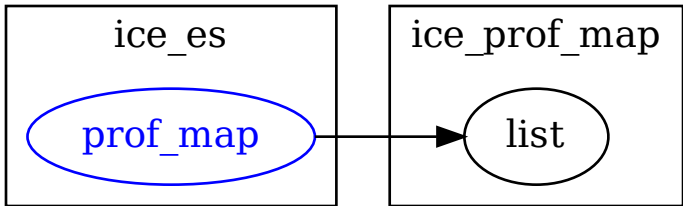
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'hifn_device@hifn_795x.c' contains an oval labeled 'alg_list' with a blue border. On the right, a rectangular box labeled 'hifn_crypto_alg@hifn_795x.c' contains an oval labeled 'entry' with a black border. A horizontal arrow points from the right side of the 'alg_list' oval to the left side of the 'entry' oval, indicating that 'alg_list' points to 'entry'.

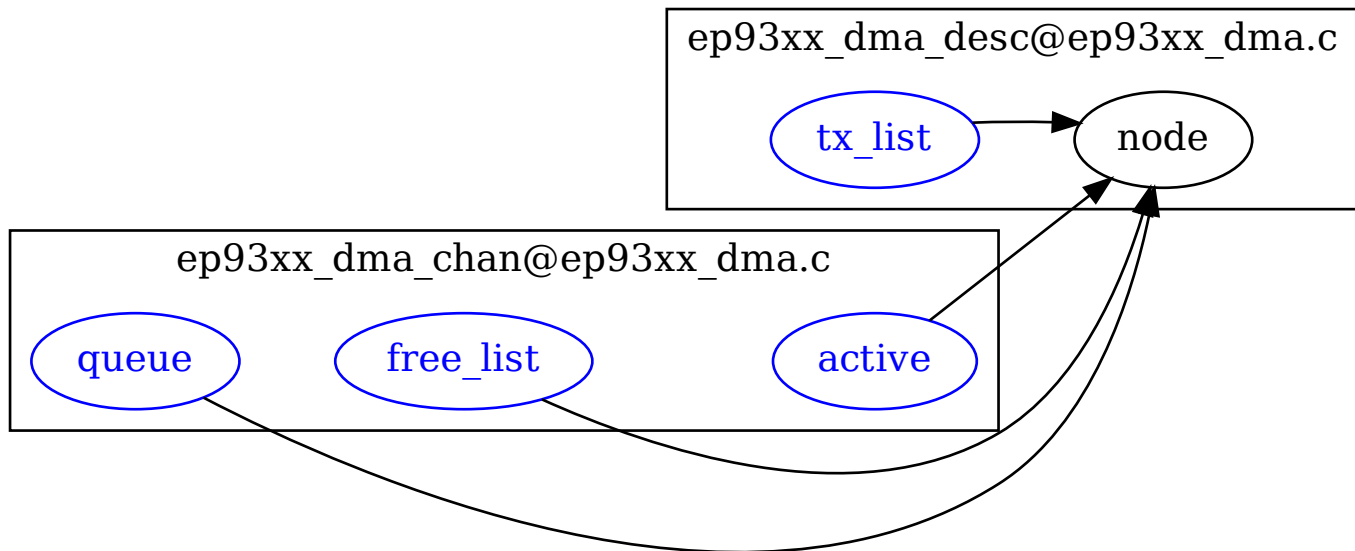


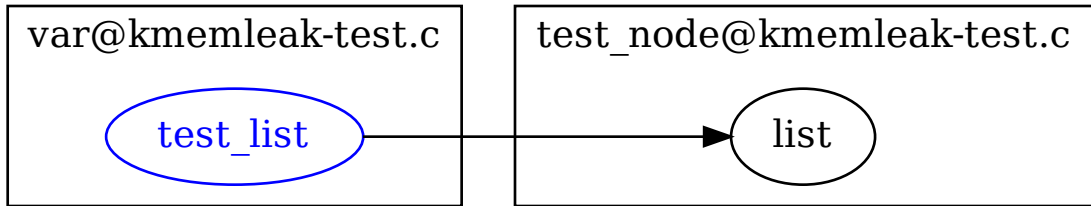


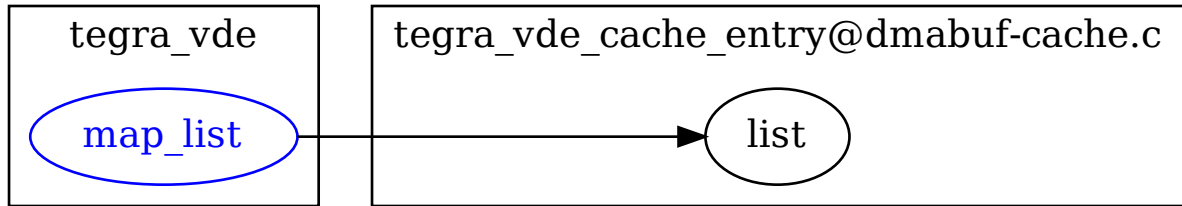












altr_arria10_edac

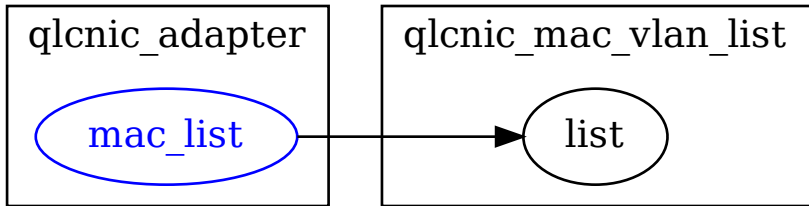
a10_ecc_devices

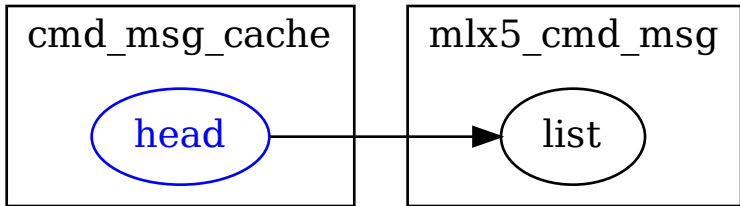
altr_edac_device_dev

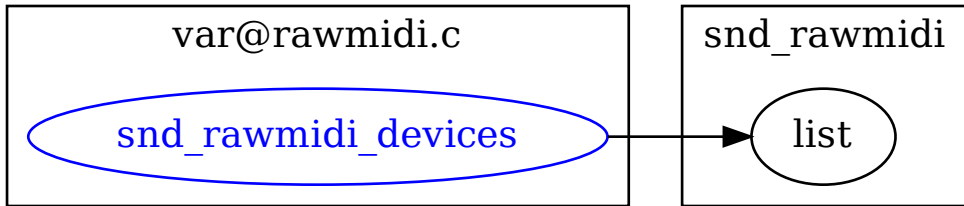
next

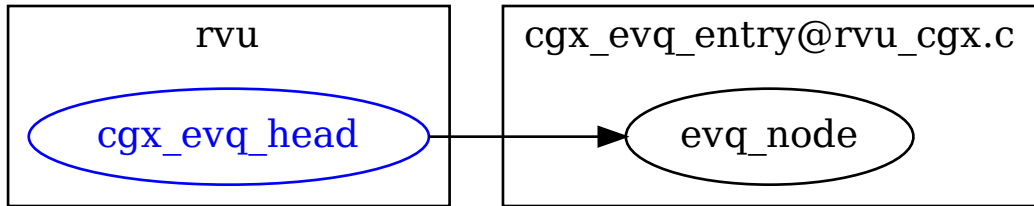
```
graph LR; subgraph Box1 [altr_arria10_edac]; A10([a10_ecc_devices]); end; subgraph Box2 [altr_edac_device_dev]; Next([next]); end; A10 --> Next;
```

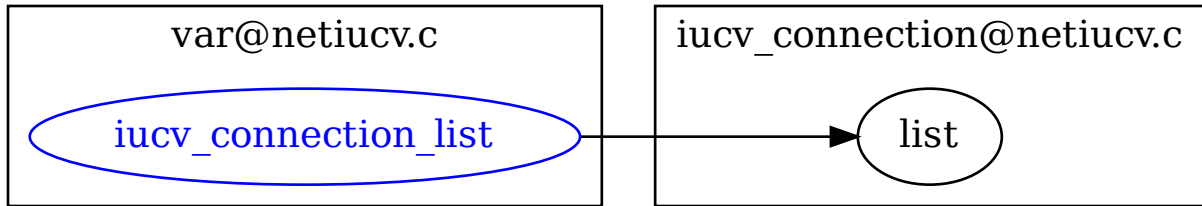
The diagram consists of two rectangular boxes. The left box is labeled 'altr_arria10_edac' and contains an oval labeled 'a10_ecc_devices' with a blue border. The right box is labeled 'altr_edac_device_dev' and contains an oval labeled 'next' with a black border. A horizontal arrow points from the right side of the 'a10_ecc_devices' oval to the left side of the 'next' oval.











var@bcmstb_gisb.c

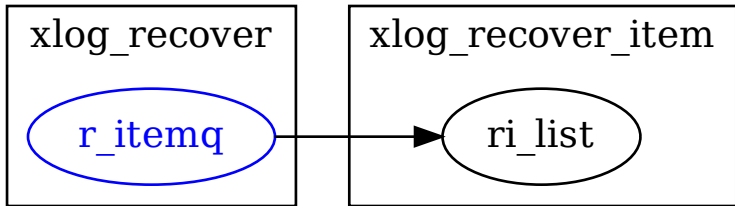
bcmstb_gisb_arb_device_list

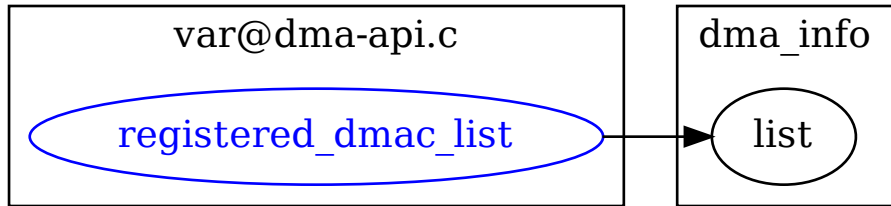
bcmstb_gisb_arb_device@bcmstb_gisb.c

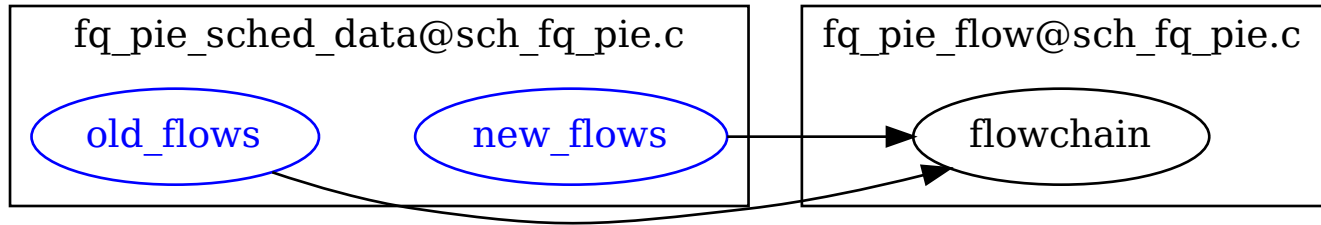
next

```
graph LR; subgraph File1 [var@bcmstb_gisb.c]; A([bcmstb_gisb_arb_device_list]); end; subgraph File2 [bcmstb_gisb_arb_device@bcmstb_gisb.c]; B([next]); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the file `var@bcmstb_gisb.c`, containing the variable `bcmstb_gisb_arb_device_list`, which is highlighted by a blue oval. On the right, a box represents the file `bcmstb_gisb_arb_device@bcmstb_gisb.c`, containing the variable `next`, which is enclosed in an oval. A horizontal arrow points from the blue oval in the first box to the oval in the second box, indicating that `bcmstb_gisb_arb_device_list` points to `next`.







var@xfs_refcount_recover_cow_leftovers

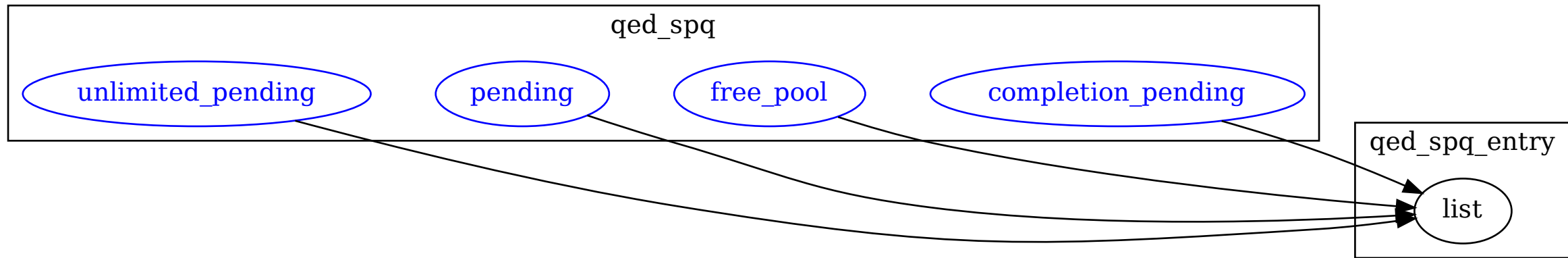
debris

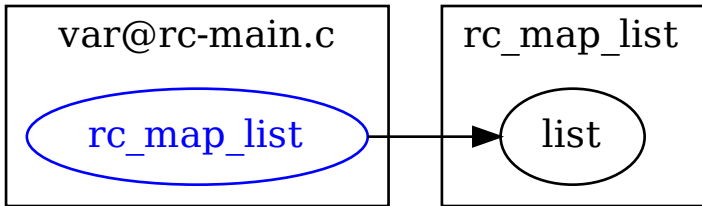
xfs_refcount_recovery@xfs_refcount.c

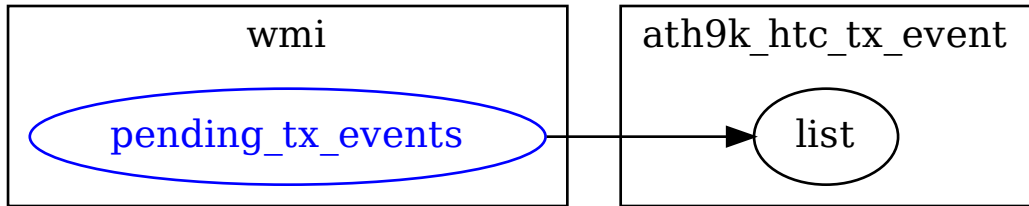
rr_list

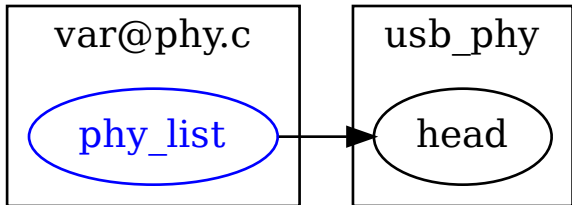
```
graph LR; debris([debris]) --> rr_list([rr_list]);
```

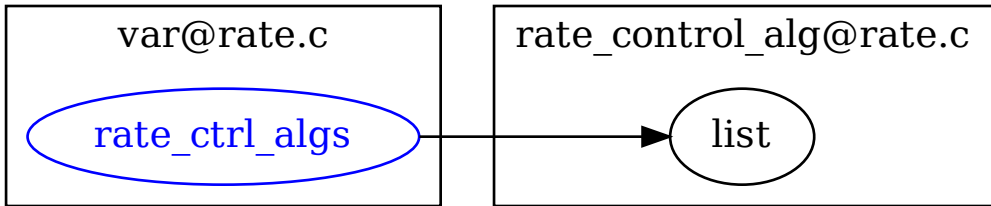
The diagram consists of two rectangular boxes. The left box contains the text 'var@xfs_refcount_recover_cow_leftovers' at the top and an oval labeled 'debris' in the center. The right box contains the text 'xfs_refcount_recovery@xfs_refcount.c' at the top and an oval labeled 'rr_list' in the center. A horizontal arrow points from the 'debris' oval in the left box to the 'rr_list' oval in the right box.

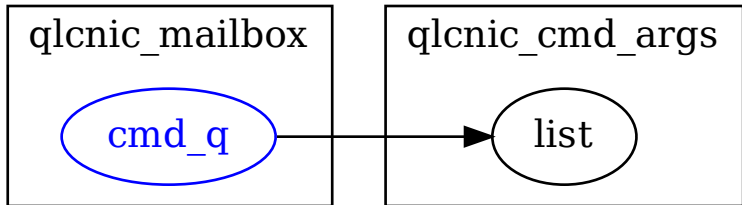


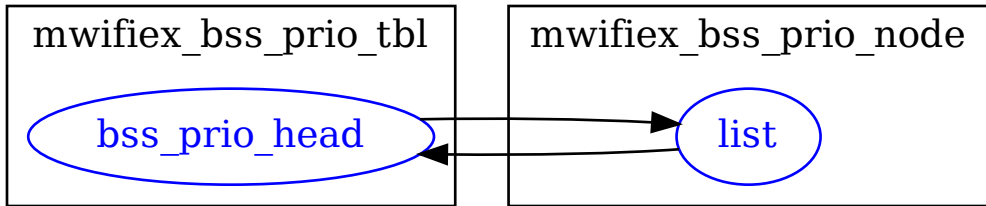


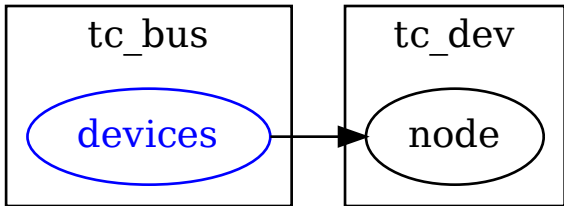




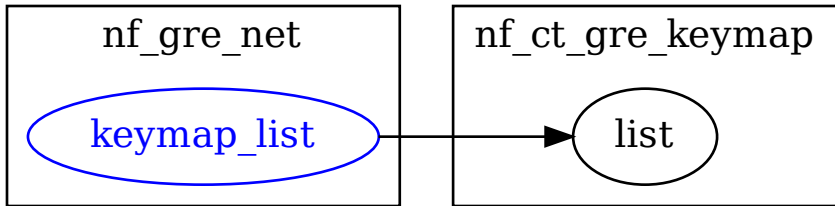


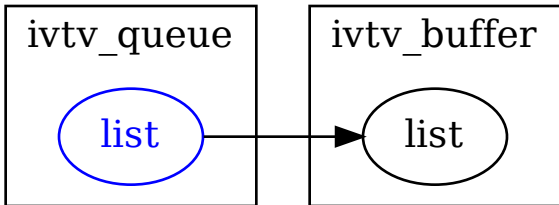


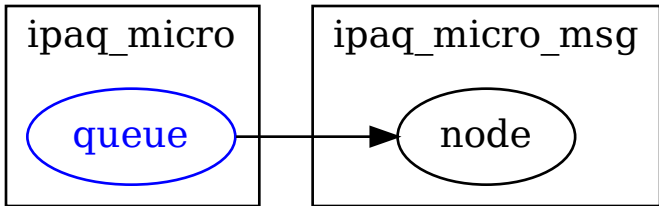












var@sclp_sd.c

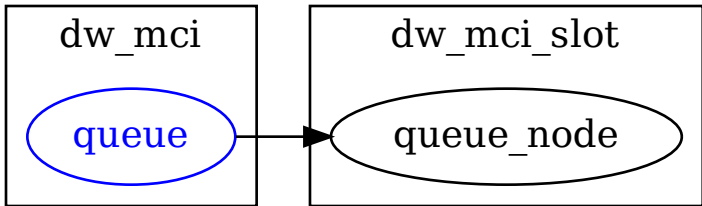
sclp_sd_queue

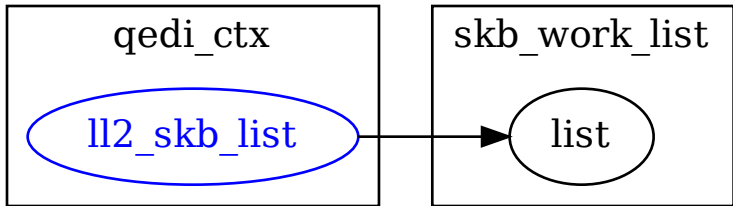
sclp_sd_listener@sclp_sd.c

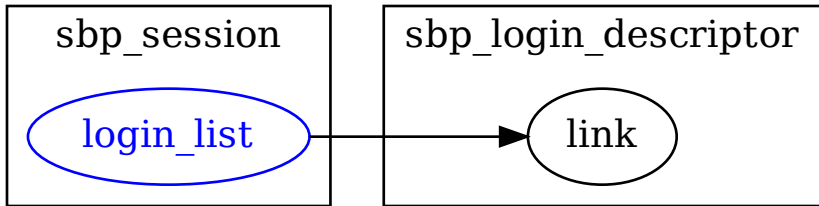
list

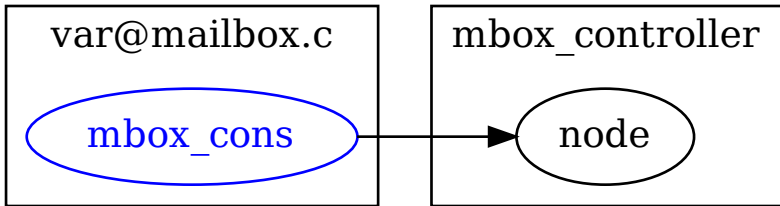
```
graph LR; subgraph var_sclp_sd_c [var@sclp_sd.c]; sclp_sd_queue; end; subgraph sclp_sd_listener_sclp_sd_c [sclp_sd_listener@sclp_sd.c]; list; end; sclp_sd_queue --> list;
```

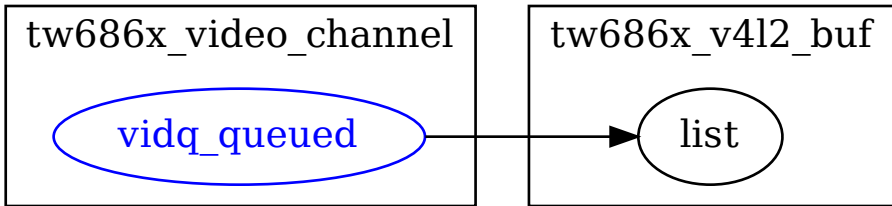
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@sclp_sd.c' contains an oval labeled 'sclp_sd_queue'. On the right, a rectangular box labeled 'sclp_sd_listener@sclp_sd.c' contains an oval labeled 'list'. A horizontal arrow points from the 'sclp_sd_queue' oval to the 'list' oval, indicating that 'sclp_sd_queue' points to 'list'.











var@ipmi_msghandler.c

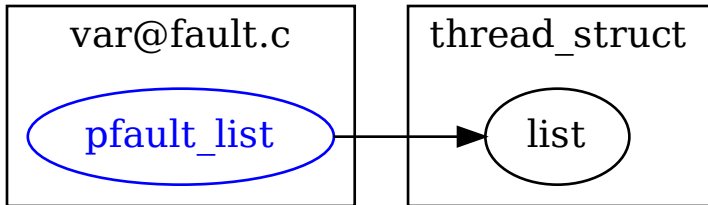
smi_watchers

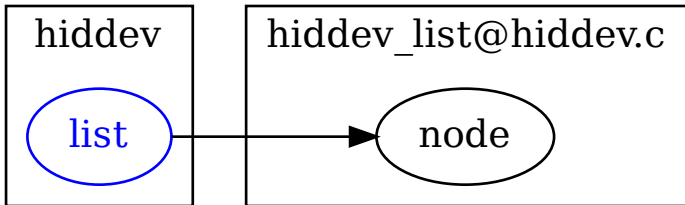
ipmi_smi_watcher

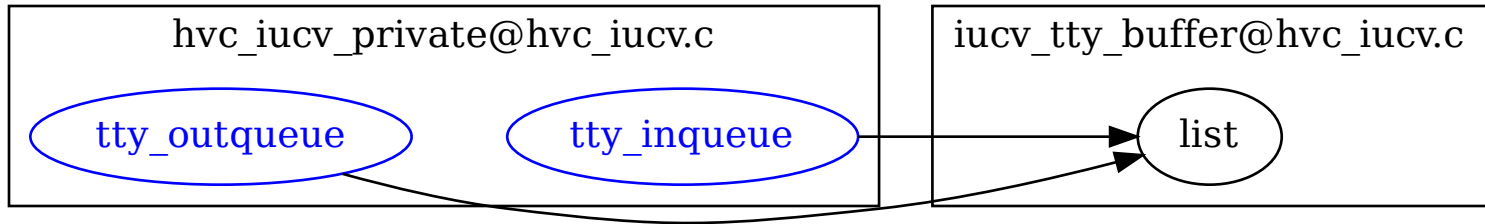
link

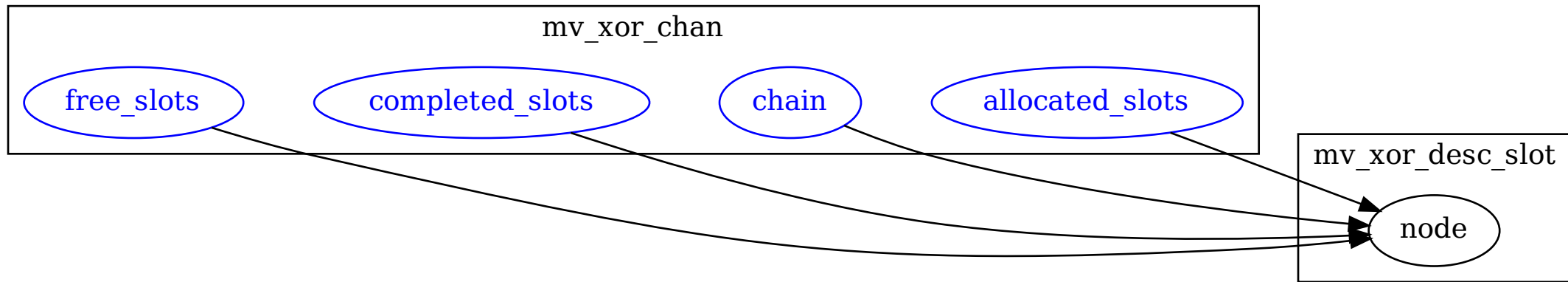
```
graph LR; subgraph Box1 [var@ipmi_msghandler.c]; SMI_WATCHERS(smi_watchers); end; subgraph Box2 [ipmi_smi_watcher]; LINK(link); end; SMI_WATCHERS --> LINK;
```

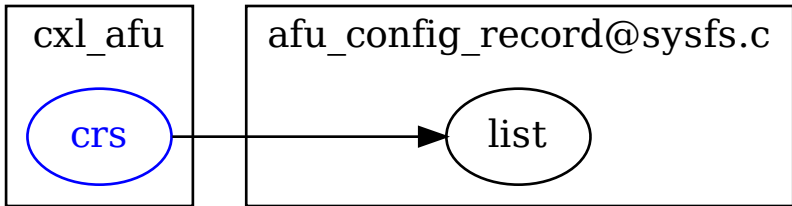
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@ipmi_msghandler.c' contains an oval labeled 'smi_watchers' in blue text. On the right, a box labeled 'ipmi_smi_watcher' contains an oval labeled 'link'. A horizontal arrow points from the right side of the 'smi_watchers' oval to the left side of the 'link' oval, indicating that 'smi_watchers' points to 'link'.

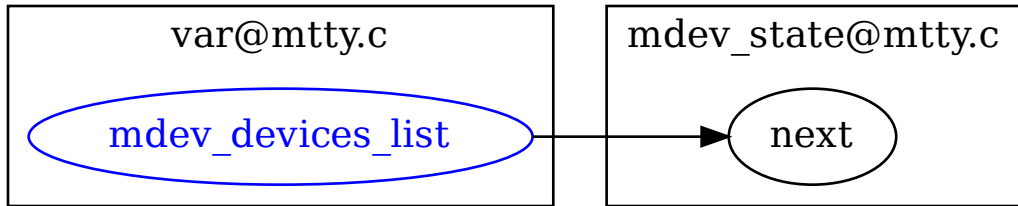


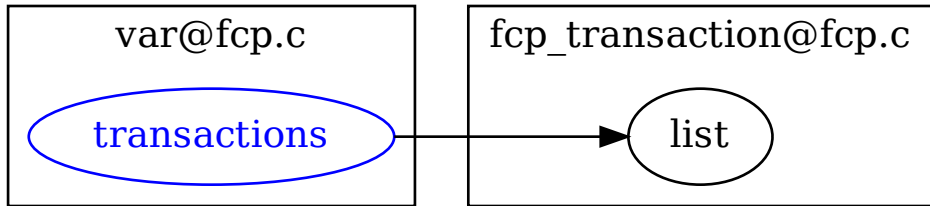












var@mmio-mod.c

trace_list

remap_trace@mmio-mod.c

list

```
graph LR; subgraph var; trace_list; end; subgraph remap_trace; list; end; trace_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@mmio-mod.c' contains an oval labeled 'trace_list'. On the right, a box labeled 'remap_trace@mmio-mod.c' contains an oval labeled 'list'. A horizontal arrow points from the 'trace_list' oval to the 'list' oval, indicating that 'trace_list' points to 'list'.

mlxsw_sp_acl_tcam_group@spectrum_acl_tcam.c

region_list

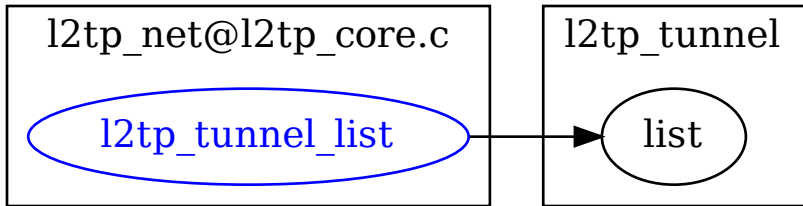
mlxsw_sp_acl_tcam_region

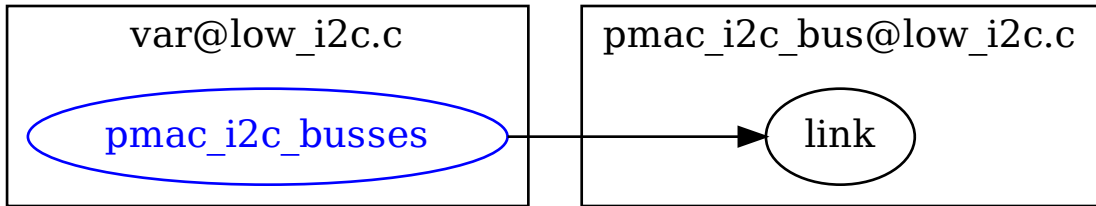
list

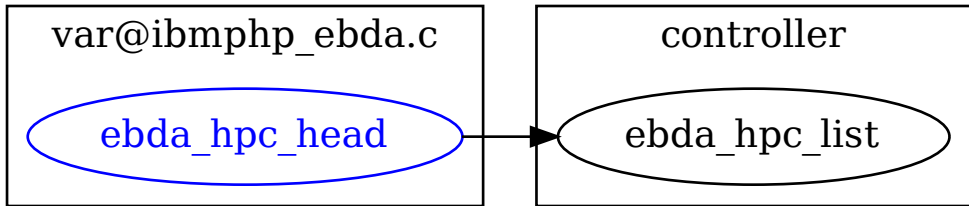
```
graph LR; A[mlxsw_sp_acl_tcam_group@spectrum_acl_tcam.c] --> B((list));
```

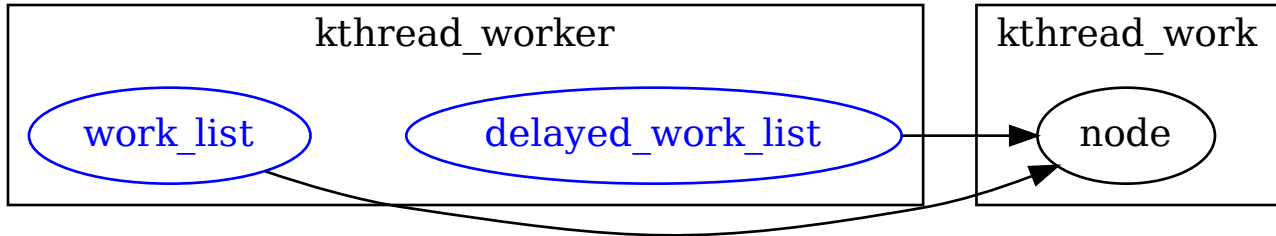
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box contains the text 'mlxsw_sp_acl_tcam_group@spectrum_acl_tcam.c'. Inside this box, the text 'region_list' is enclosed in a blue oval. A horizontal arrow points from the right side of this box to a second rectangular box on the right. This second box contains the text 'mlxsw_sp_acl_tcam_region'. Inside this box, the text 'list' is enclosed in a black oval. The arrow indicates that the 'region_list' variable in the first structure points to the 'list' variable in the second structure.











var@sta2x11-fixup.c

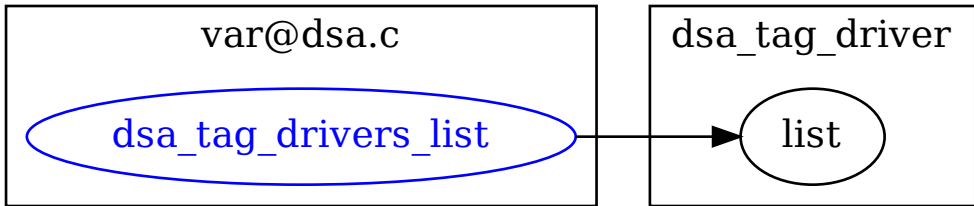
sta2x11_instance_list

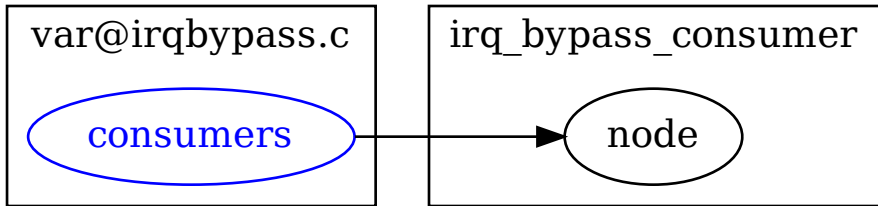
sta2x11_instance@sta2x11-fixup.c

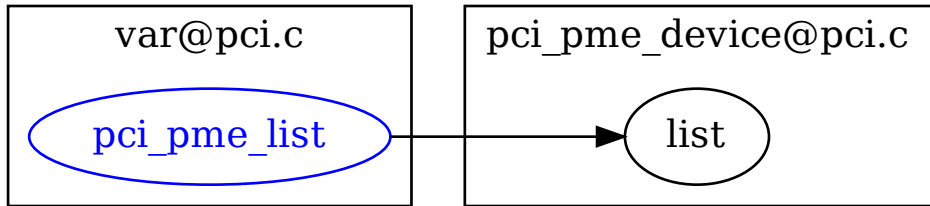
list

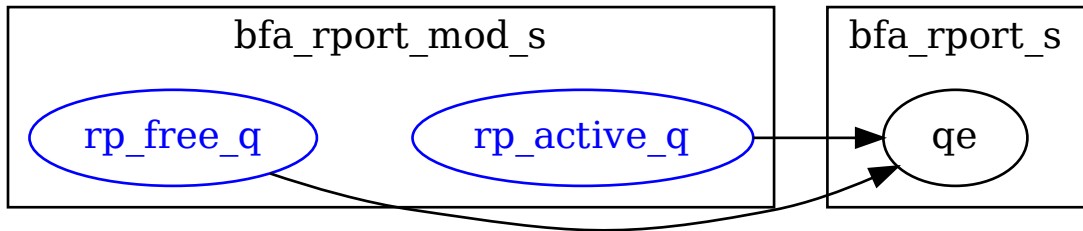
```
graph LR; A[sta2x11_instance_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@sta2x11-fixup.c' at the top and 'sta2x11_instance_list' below it. The 'sta2x11_instance_list' text is enclosed in a blue oval. A horizontal arrow points from the right side of this blue oval to a black oval on the right. This black oval is located inside a second rectangular box that also contains the text 'sta2x11_instance@sta2x11-fixup.c' at the top and 'list' below it.









var@scsi_transport_iscsi.c

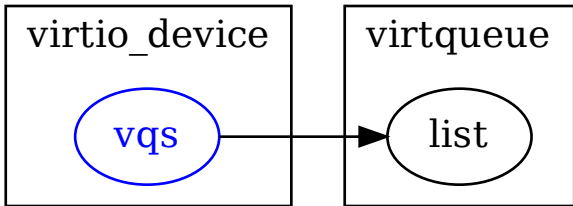
connlist

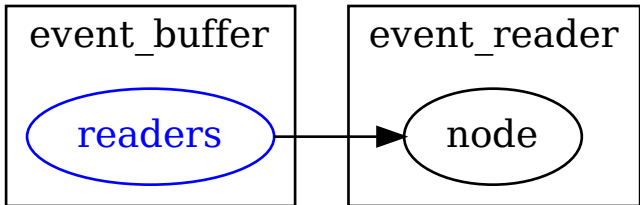
iscsi_cls_conn

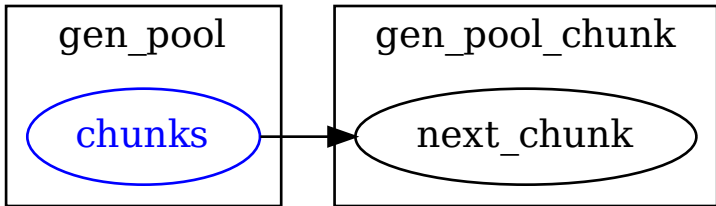
conn_list

```
graph LR; A[connlist] --> B[conn_list];
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@scsi_transport_iscsi.c' contains an oval labeled 'connlist' with a blue border. On the right, a rectangular box labeled 'iscsi_cls_conn' contains an oval labeled 'conn_list' with a black border. A horizontal arrow points from the 'connlist' oval to the 'conn_list' oval, indicating that 'connlist' points to 'conn_list'.





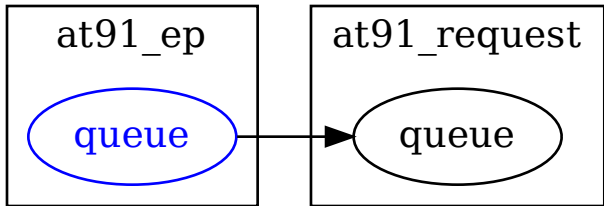


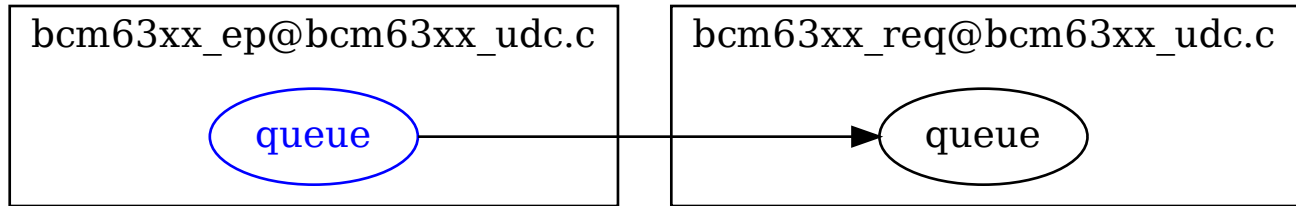
nfscache.c

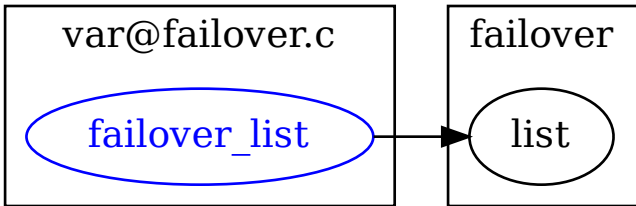
lru_head

svc_cacherep

c_lru







var@generic-chip.c

gc_list

irq_chip_generic

list

```
graph LR; gc_list((gc_list)) --> list((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@generic-chip.c' contains an oval labeled 'gc_list' in blue text. On the right, a box labeled 'irq_chip_generic' contains an oval labeled 'list' in black text. A horizontal arrow points from the 'gc_list' oval to the 'list' oval, indicating that 'gc_list' points to 'list'.

var@fail_function.c

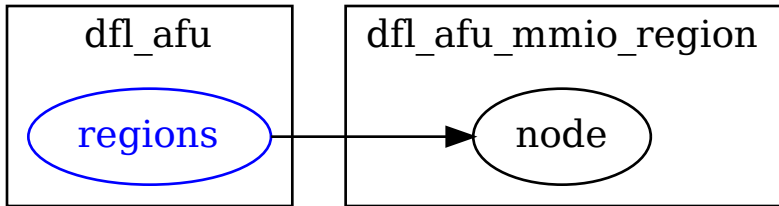
fei_attr_list

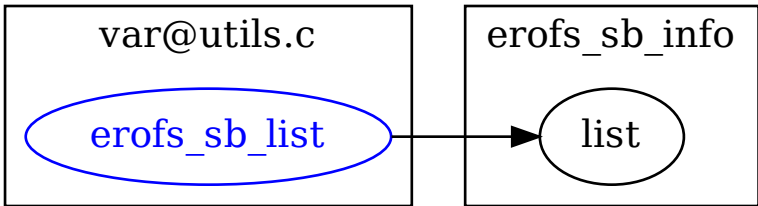
fei_attr@fail_function.c

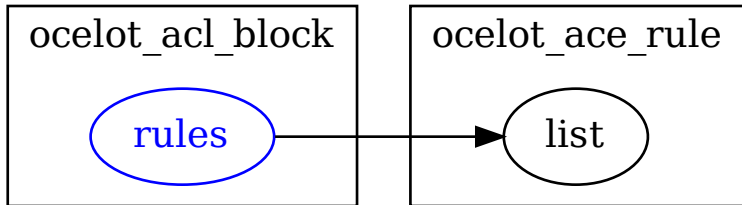
list

```
graph LR; A[fei_attr_list] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@fail_function.c' contains a blue oval labeled 'fei_attr_list'. On the right, a box labeled 'fei_attr@fail_function.c' contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'fei_attr_list' oval to the left side of the 'list' oval, indicating that 'fei_attr_list' points to 'list'.





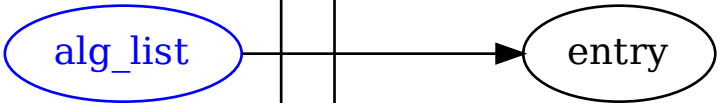


talitos_private

alg_list

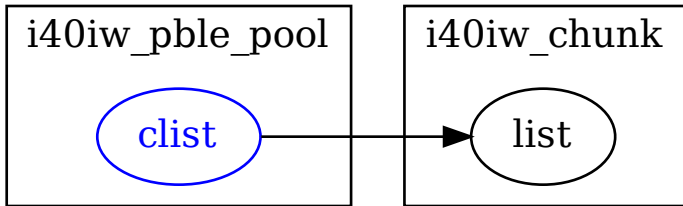
talitos_crypto_alg@talitos.c

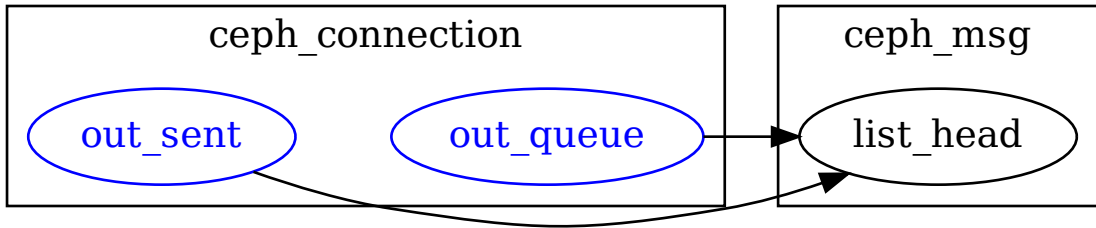
entry

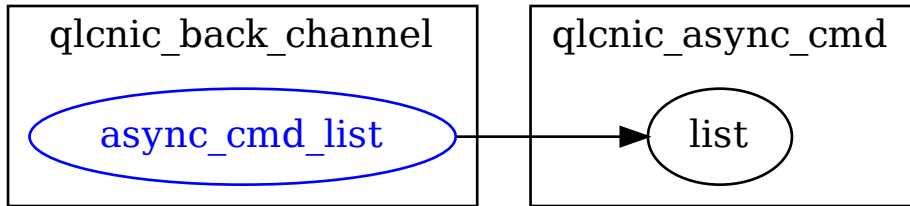


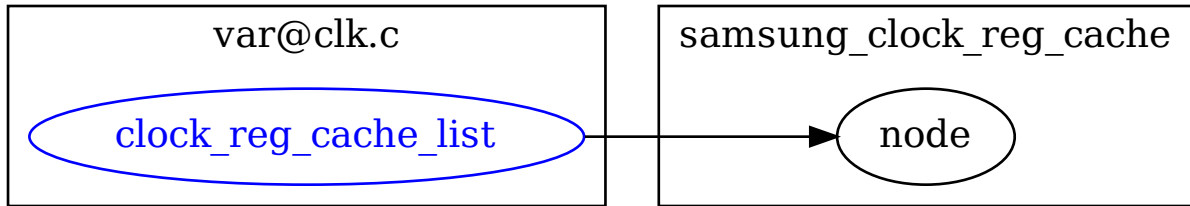
```
graph LR; subgraph talitos_private; alg_list; end; subgraph talitos_crypto_alg_at_talitos_c [talitos_crypto_alg@talitos.c]; entry; end; alg_list --> entry;
```

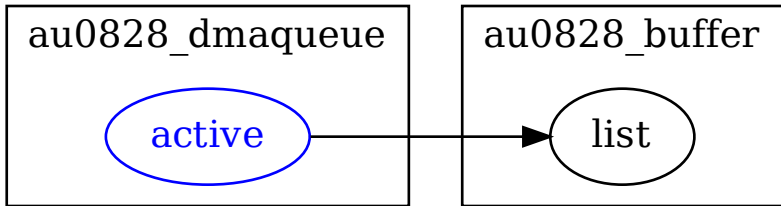
The diagram illustrates a pointer relationship between two memory spaces. On the left, a box labeled 'talitos_private' contains an oval labeled 'alg_list'. On the right, a box labeled 'talitos_crypto_alg@talitos.c' contains an oval labeled 'entry'. A horizontal arrow points from the 'alg_list' oval to the 'entry' oval, indicating that 'alg_list' points to the 'entry' in the 'talitos_crypto_alg@talitos.c' space.

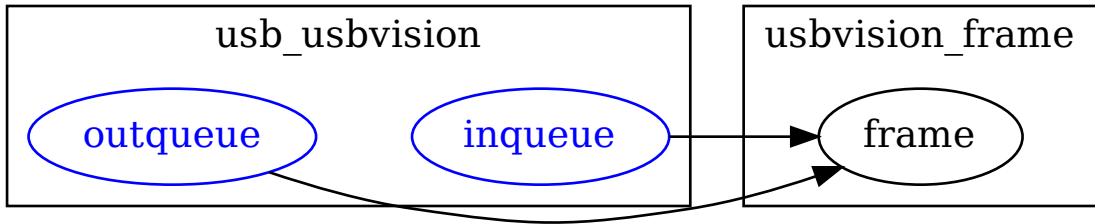


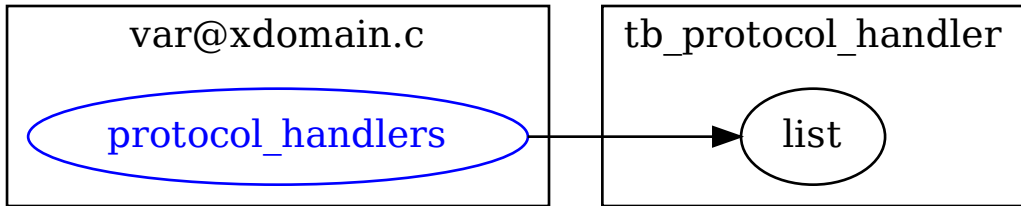












video_i2c_data@video-i2c.c

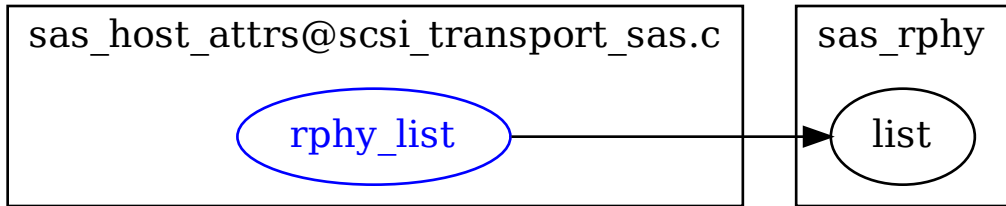
vid_cap_active

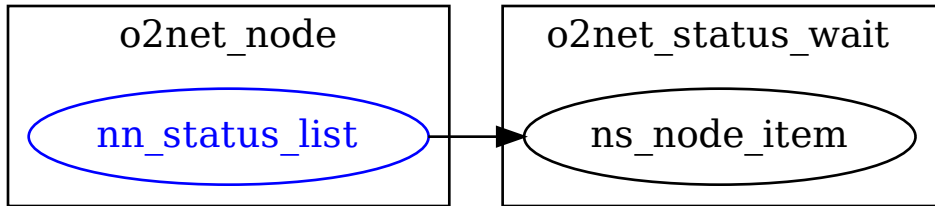
video_i2c_buffer@video-i2c.c

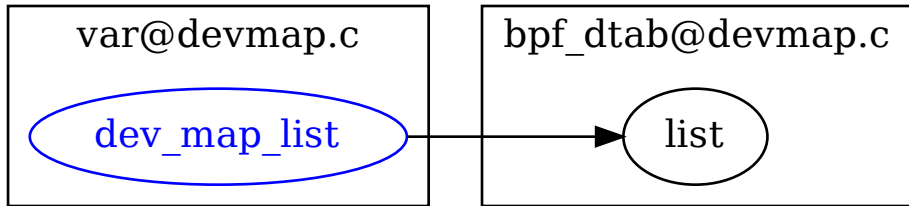
list

```
graph LR; subgraph Module1 [video_i2c_data@video-i2c.c]; vid_cap_active; end; subgraph Module2 [video_i2c_buffer@video-i2c.c]; list; end; vid_cap_active --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'video_i2c_data@video-i2c.c' contains an oval labeled 'vid_cap_active'. On the right, a box labeled 'video_i2c_buffer@video-i2c.c' contains an oval labeled 'list'. A horizontal arrow points from the 'vid_cap_active' oval to the 'list' oval, indicating that 'vid_cap_active' points to the 'list' structure.







var@mISDNinfineon.c

Cards

inf_hw@mISDNinfineon.c

list

```
graph LR; subgraph Box1 [var@mISDNinfineon.c]; Cards; end; subgraph Box2 [inf_hw@mISDNinfineon.c]; list; end; Cards --> list;
```


The diagram consists of two rectangular boxes. The left box is labeled 'var@mISDNinfineon.c' and contains an oval labeled 'Cards' with a blue border. The right box is labeled 'inf_hw@mISDNinfineon.c' and contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'Cards' oval to the 'list' oval, indicating a pointer or reference relationship.

i3c_generic_ibi_pool@master.c

free_slots

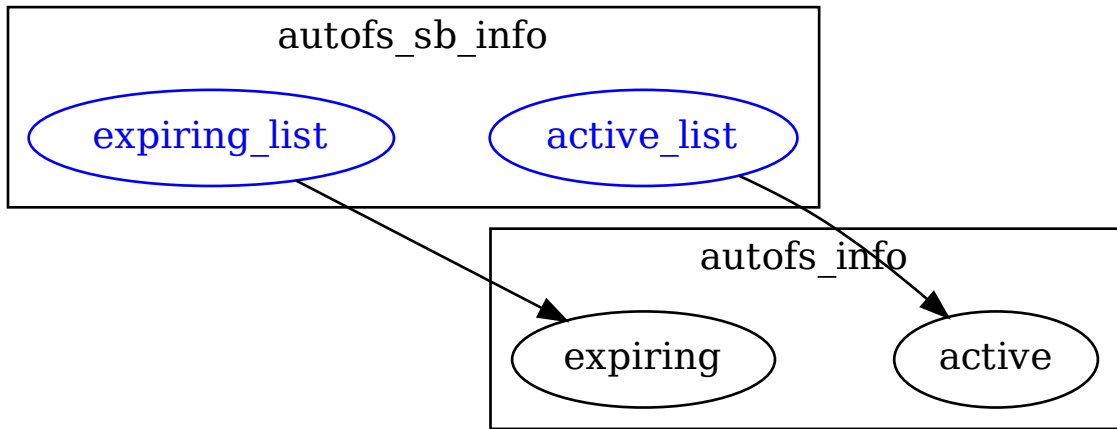
i3c_generic_ibi_slot@master.c

node



```
graph LR; subgraph "i3c_generic_ibi_pool@master.c"; free_slots; end; subgraph "i3c_generic_ibi_slot@master.c"; node; end; free_slots --> node;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'i3c_generic_ibi_pool@master.c' contains an oval labeled 'free_slots'. On the right, a box labeled 'i3c_generic_ibi_slot@master.c' contains an oval labeled 'node'. A horizontal arrow points from the 'free_slots' oval to the 'node' oval, indicating that 'free_slots' points to 'node'.



pvcalls_bedata@pvcalls-front.c

socket_mappings

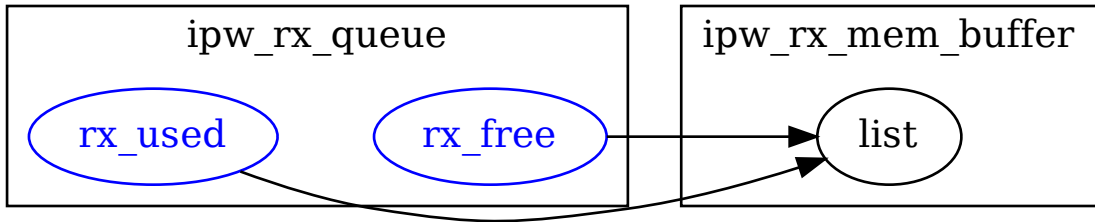
sock_mapping@pvcalls-front.c

list



```
graph LR; subgraph LeftBox [pvcalls_bedata@pvcalls-front.c]; SM(socket_mappings); end; subgraph RightBox [sock_mapping@pvcalls-front.c]; L((list)); end; SM --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'pvcalls_bedata@pvcalls-front.c' and contains a blue oval labeled 'socket_mappings'. The right box is labeled 'sock_mapping@pvcalls-front.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'socket_mappings' oval to the left side of the 'list' oval.



var@livepatch-shadow-mod.c

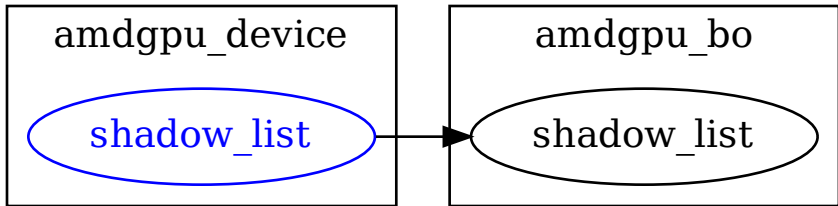
dummy_list

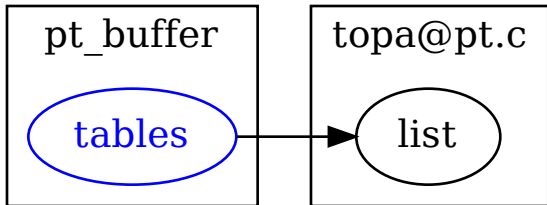
dummy@livepatch-shadow-mod.c

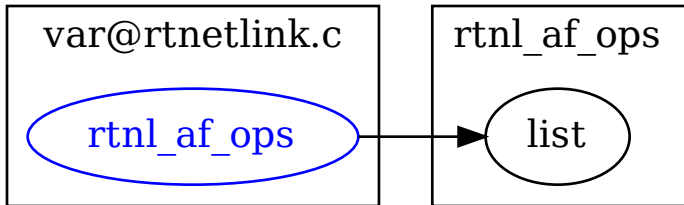
list

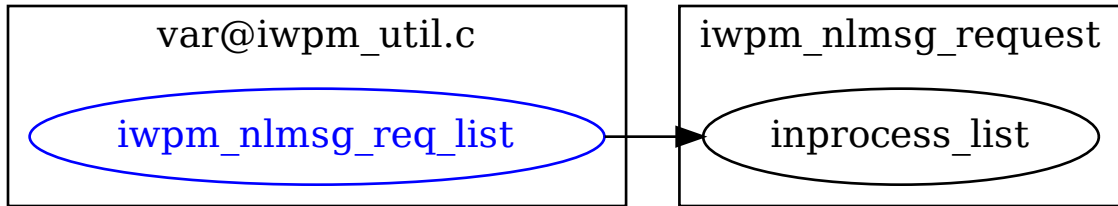
```
graph LR; subgraph Module1 [var@livepatch-shadow-mod.c]; dummy_list; end; subgraph Module2 [dummy@livepatch-shadow-mod.c]; list; end; dummy_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@livepatch-shadow-mod.c' contains a blue oval labeled 'dummy_list'. On the right, a box labeled 'dummy@livepatch-shadow-mod.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'dummy_list' oval to the 'list' oval, indicating that 'dummy_list' points to 'list'.









var@dell-smbios-base.c

smbios_device_list

smbios_device@dell-smbios-base.c

list

```
graph LR; subgraph File1 [var@dell-smbios-base.c]; A(smbios_device_list); end; subgraph File2 [smbios_device@dell-smbios-base.c]; B((list)); end; A --> B;
```

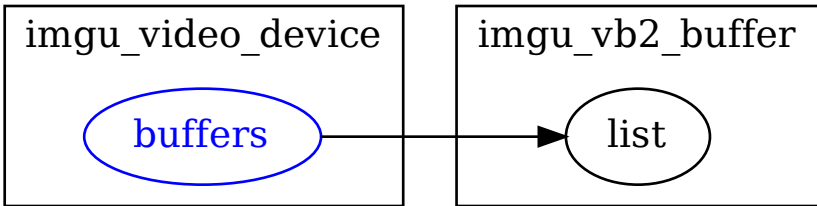
The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file `var@dell-smbios-base.c`, containing the variable `smbios_device_list` which is highlighted with a blue oval. On the right, a box represents the source file `smbios_device@dell-smbios-base.c`, containing a variable `list` enclosed in an oval. A horizontal arrow points from the `smbios_device_list` oval in the first box to the `list` oval in the second box, indicating that `smbios_device_list` points to the `list` variable.

imgu_video_device

buffers

imgu_vb2_buffer

list



iommu_group@iommu.c

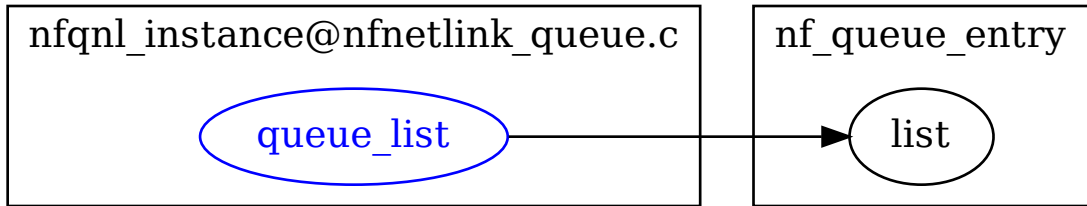
devices

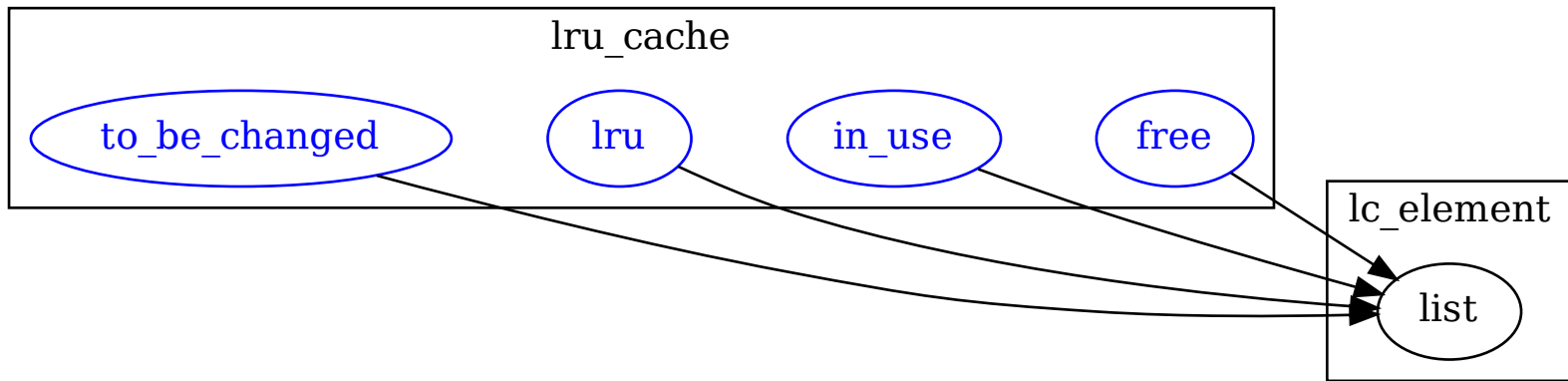
group_device@iommu.c

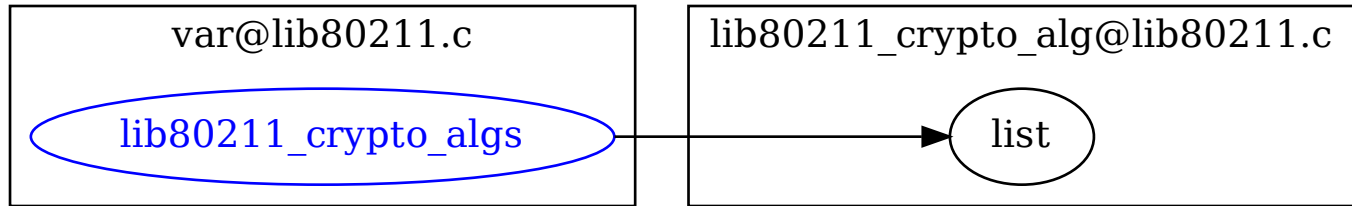
list

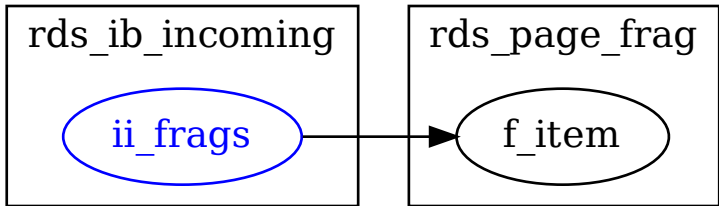
```
graph LR; subgraph iommu_group [iommu_group@iommu.c]; devices; end; subgraph group_device [group_device@iommu.c]; list; end; devices --> list;
```

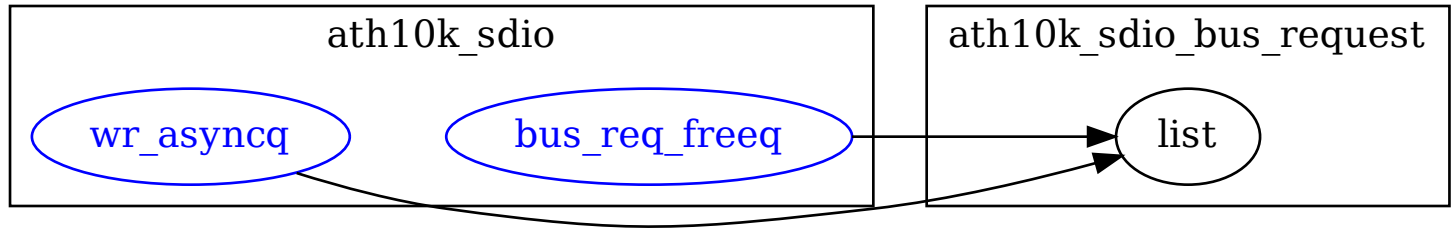
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'iommu_group@iommu.c' contains an oval labeled 'devices'. On the right, a rectangular box labeled 'group_device@iommu.c' contains an oval labeled 'list'. A horizontal arrow points from the 'devices' oval to the 'list' oval, indicating that the 'devices' variable points to the 'list' structure.

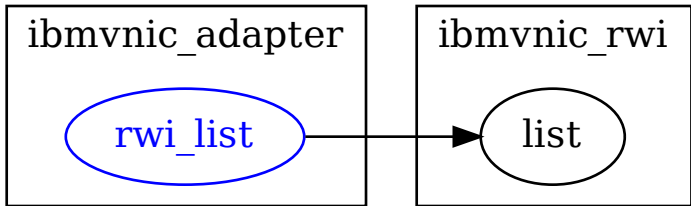


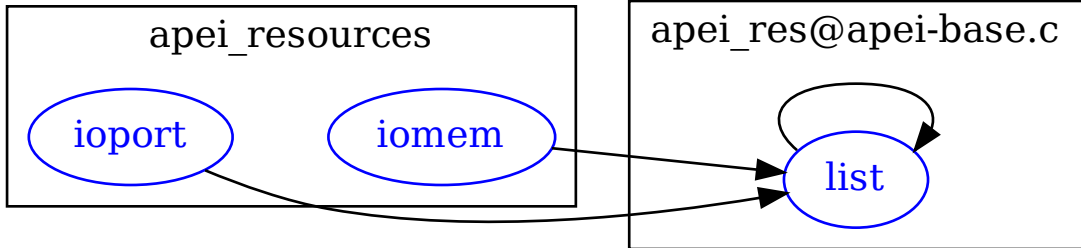


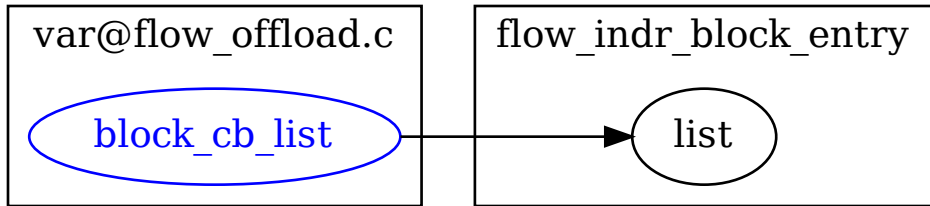












capture_priv@imx-media-capture.c

ready_q

imx_media_buffer

list

```
graph LR; subgraph Box1 [capture_priv@imx-media-capture.c]; ready_q([ready_q]); end; subgraph Box2 [imx_media_buffer]; list([list]); end; ready_q --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'capture_priv@imx-media-capture.c' and contains an oval labeled 'ready_q' in blue text. The right box is labeled 'imx_media_buffer' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'ready_q' oval to the left side of the 'list' oval, indicating a pointer relationship.

var@xen.c

dev_domain_list

xen_device_domain_owner@xen.c

list

```
graph LR; A[dev_domain_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@xen.c' contains an oval labeled 'dev_domain_list' in blue text. On the right, a rectangular box labeled 'xen_device_domain_owner@xen.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dev_domain_list' oval to the 'list' oval, indicating that 'dev_domain_list' points to 'list'.

var@xfrm_ipcomp.c

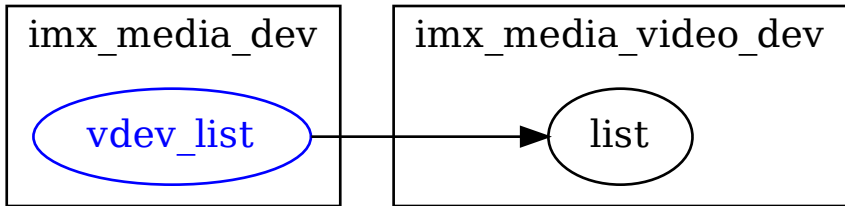
ipcomp_tfms_list

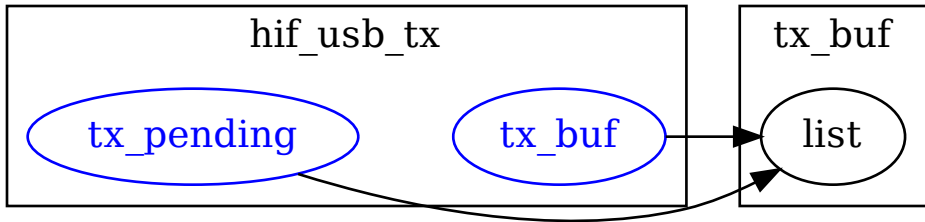
ipcomp_tfms@xfrm_ipcomp.c

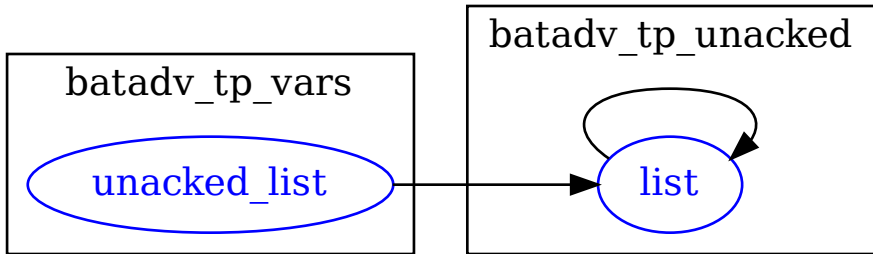
list

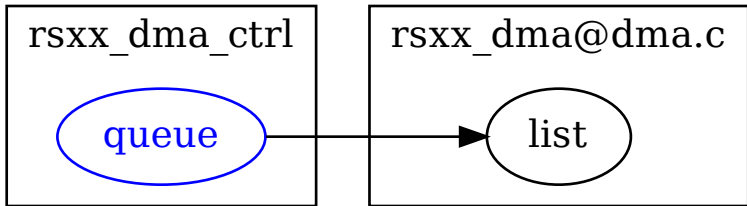
```
graph LR; A["var@xfrm_ipcomp.c  
ipcomp_tfms_list"] --> B["ipcomp_tfms@xfrm_ipcomp.c  
list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@xfrm_ipcomp.c' and a blue oval containing 'ipcomp_tfms_list'. On the right, another rectangular box contains the text 'ipcomp_tfms@xfrm_ipcomp.c' and a black oval containing 'list'. A horizontal arrow points from the blue oval in the first box to the black oval in the second box, indicating that the variable 'ipcomp_tfms_list' points to the variable 'list'.









usbport_trig_data@ledtrig-usbport.c

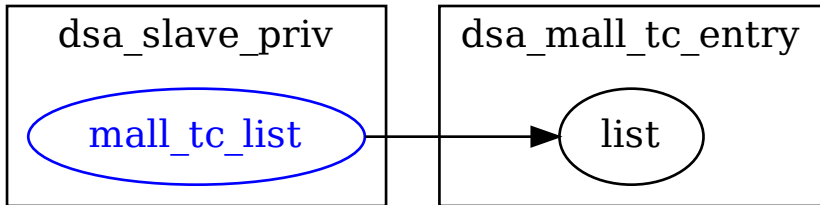
ports

usbport_trig_port@ledtrig-usbport.c

list



```
graph LR; subgraph Module1 [usbport_trig_data@ledtrig-usbport.c]; ports; end; subgraph Module2 [usbport_trig_port@ledtrig-usbport.c]; list; end; ports --> list;
```



var@ti_sci_scan_clocks_from_dt

clks

sci_clk@sci-clk.c

node

```
graph LR; clks((clks)) --> node((node));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@ti_sci_scan_clocks_from_dt' contains an oval labeled 'clks' with a blue border. On the right, a box labeled 'sci_clk@sci-clk.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the 'clks' oval to the 'node' oval, indicating that 'clks' points to 'node'.

var@nf_conntrack_helper.c

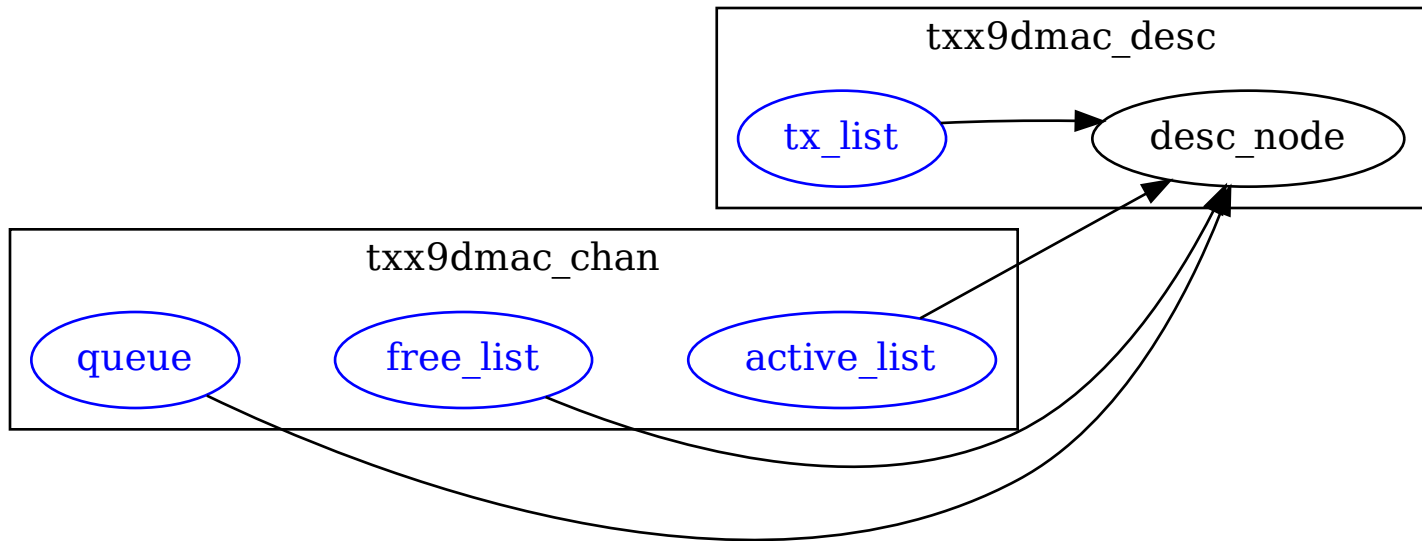
nf_ct_nat_helpers

nf_conntrack_nat_helper

list

```
graph LR; A[nf_ct_nat_helpers] --> B((list));
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@nf_conntrack_helper.c' at the top and 'nf_ct_nat_helpers' in a blue oval below it. The right box contains the text 'nf_conntrack_nat_helper' at the top and 'list' in a black oval below it. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.



var@windfarm_smu_controls.c

smu_fans

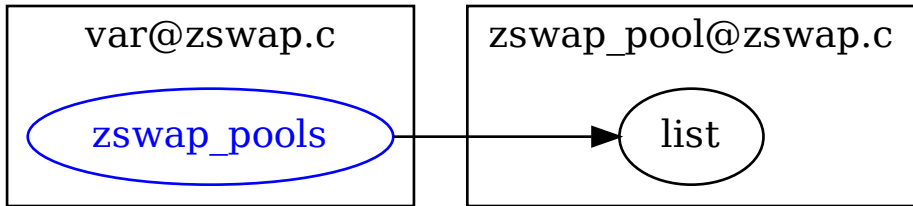
smu_fan_control@windfarm_smu_controls.c

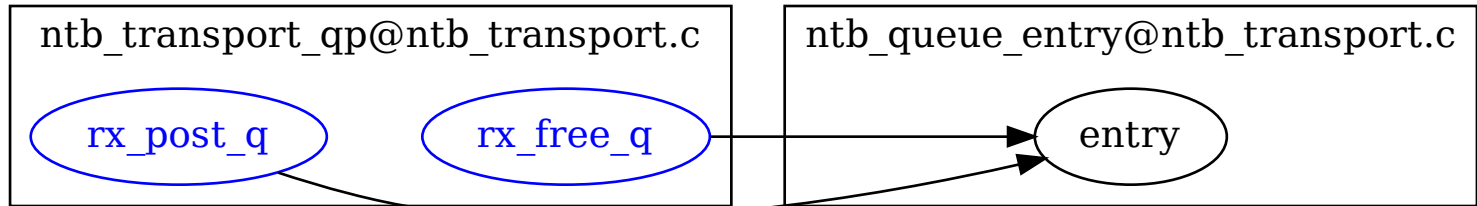
link

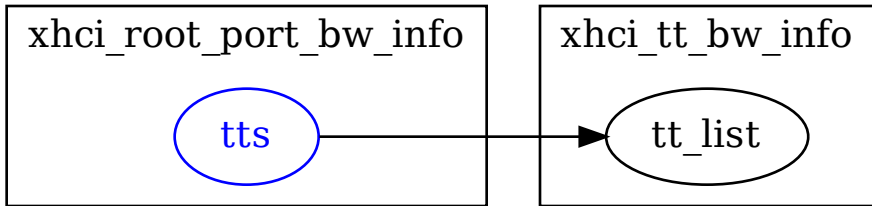


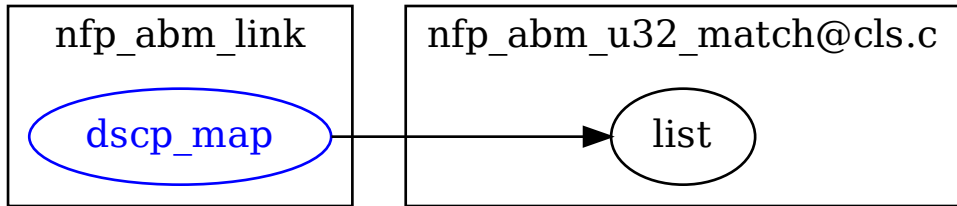
```
graph LR; A([smu_fans]) --> B([link]);
```

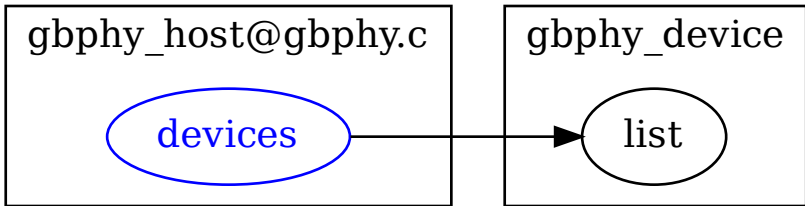
The diagram consists of two rectangular boxes. The left box contains the text 'var@windfarm_smu_controls.c' at the top and a blue oval containing 'smu_fans' below it. The right box contains the text 'smu_fan_control@windfarm_smu_controls.c' at the top and a black oval containing 'link' below it. A horizontal arrow points from the 'smu_fans' oval to the 'link' oval.











ceu_device@renesas-ceu.c

capture

ceu_buffer@renesas-ceu.c

queue

```
graph LR; A[ceu_device@renesas-ceu.c] -- capture --> B[queue]; subgraph B [ceu_buffer@renesas-ceu.c]; B; end
```

The diagram illustrates a data flow process. On the left, a rectangular box labeled 'ceu_device@renesas-ceu.c' contains an oval labeled 'capture' with a blue border. A horizontal arrow points from the right side of the 'capture' oval to the left side of another oval labeled 'queue'. This 'queue' oval is located within a larger rectangular box on the right labeled 'ceu_buffer@renesas-ceu.c'.

ibmpex_driver_data@ibmpex.c

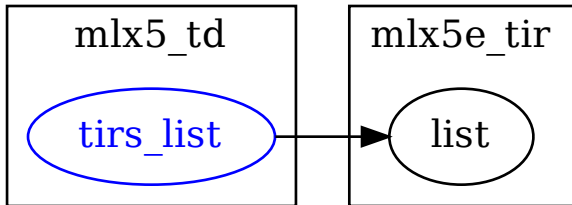
bmc_data

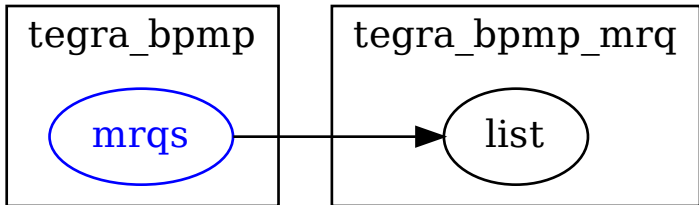
ibmpex_bmc_data@ibmpex.c

list

```
graph LR; subgraph "ibmpex_driver_data@ibmpex.c"; bmc_data((bmc_data)); end; subgraph "ibmpex_bmc_data@ibmpex.c"; list((list)); end; bmc_data --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'ibmpex_driver_data@ibmpex.c' contains an oval labeled 'bmc_data'. On the right, a box labeled 'ibmpex_bmc_data@ibmpex.c' contains an oval labeled 'list'. A horizontal arrow points from the 'bmc_data' oval to the 'list' oval, indicating that the 'bmc_data' variable points to the 'list' structure.





omap_clkctrl_provider@clkctrl.c

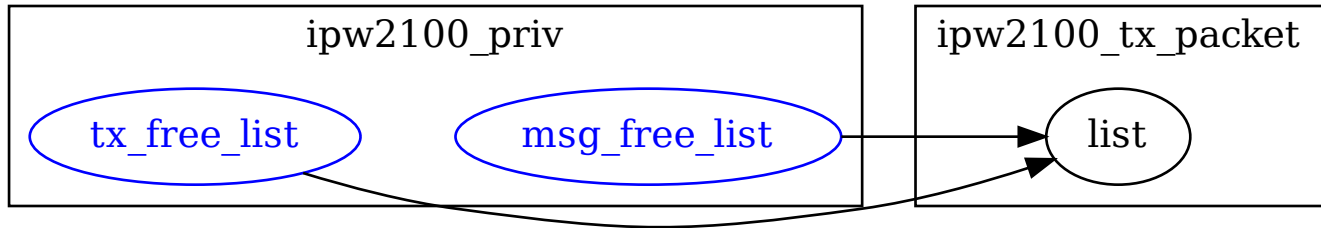
clocks

omap_clkctrl_clk@clkctrl.c

node

```
graph LR; subgraph Module1 [omap_clkctrl_provider@clkctrl.c]; direction TB; clocks((clocks)); end; subgraph Module2 [omap_clkctrl_clk@clkctrl.c]; direction TB; node((node)); end; Module1 --> Module2;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'omap_clkctrl_provider@clkctrl.c' contains an oval labeled 'clocks' with a blue border. On the right, a box labeled 'omap_clkctrl_clk@clkctrl.c' contains an oval labeled 'node'. A horizontal arrow points from the 'clocks' oval to the 'node' oval, indicating that the 'clocks' variable in the first module points to the 'node' variable in the second module.



sched_gate_list@sch_taprio.c

entries

sched_entry@sch_taprio.c

list



kernfs_open_node@file.c

files

kernfs_open_file

list

```
graph LR; subgraph Box1 [kernfs_open_node@file.c]; files((files)); end; subgraph Box2 [kernfs_open_file]; list((list)); end; files --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'kernfs_open_node@file.c' and contains a blue oval labeled 'files'. The right box is labeled 'kernfs_open_file' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'files' oval to the left side of the 'list' oval.

capiminor@capi.c

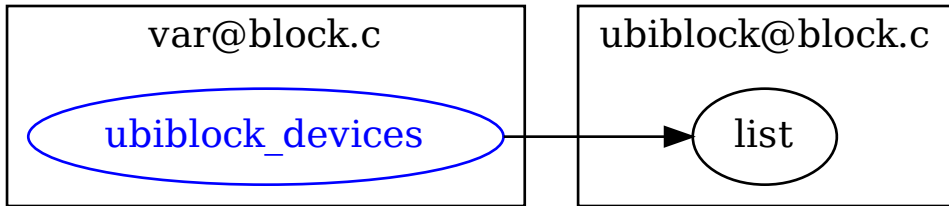
ackqueue

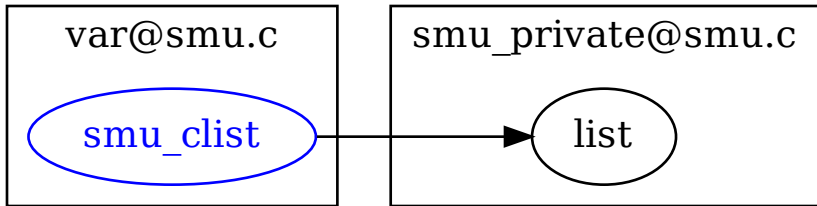
ackqueue_entry@capi.c

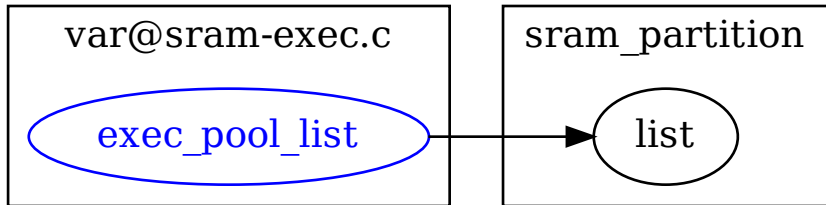
list

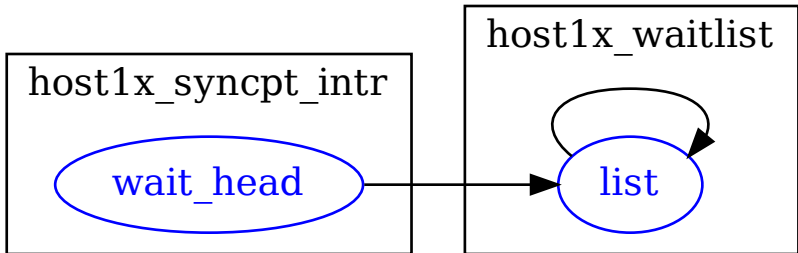
```
graph LR; subgraph "capiminor@capi.c"; A([ackqueue]); end; subgraph "ackqueue_entry@capi.c"; B([list]); end; A --> B;
```

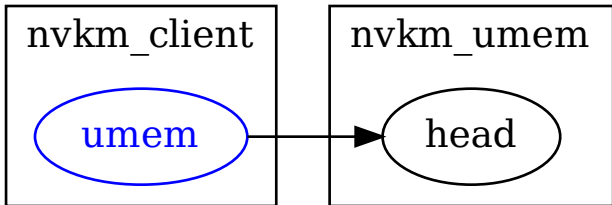
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'capiminor@capi.c' contains an oval labeled 'ackqueue'. On the right, a rectangular box labeled 'ackqueue_entry@capi.c' contains an oval labeled 'list'. A horizontal arrow points from the 'ackqueue' oval to the 'list' oval, indicating that the 'ackqueue' variable points to the 'list' structure.

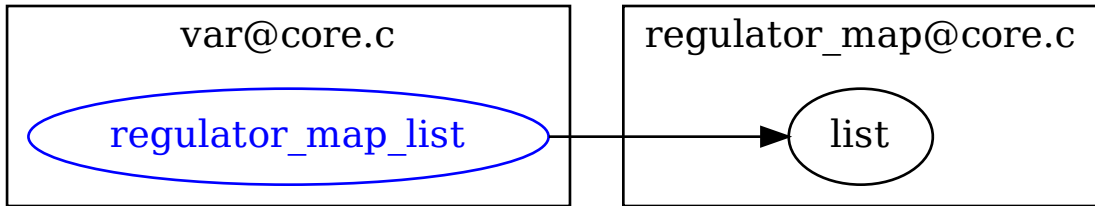












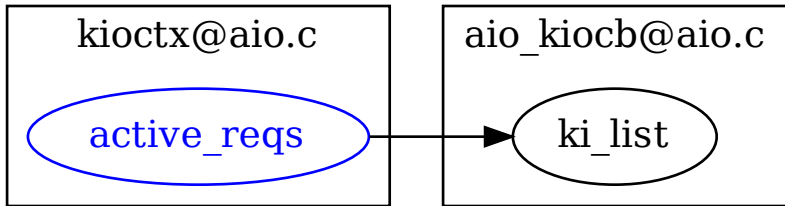
var@bpf_trace.c

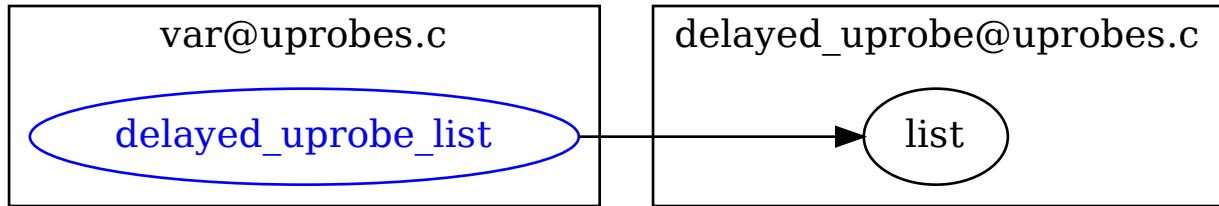
bpf_trace_modules

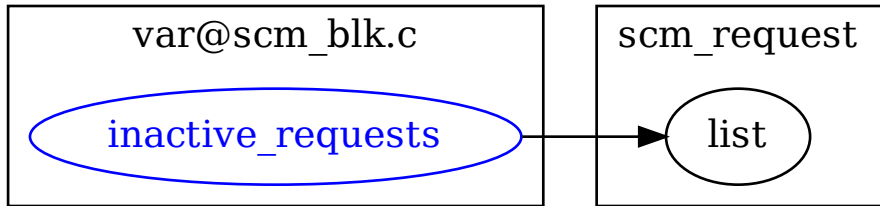
bpf_trace_module@bpf_trace.c

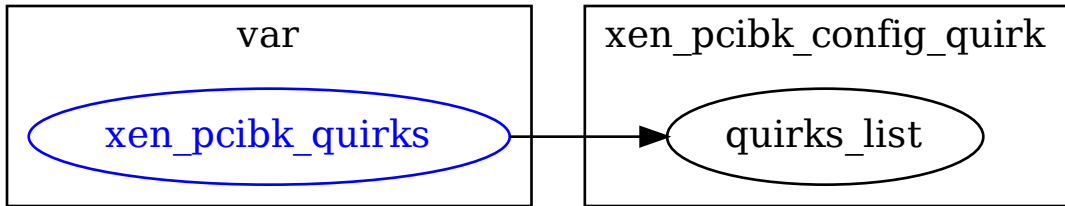
list

```
graph LR; subgraph var ["var@bpf_trace.c"]; bpf_trace_modules; end; subgraph module ["bpf_trace_module@bpf_trace.c"]; list; end; bpf_trace_modules --> list;
```

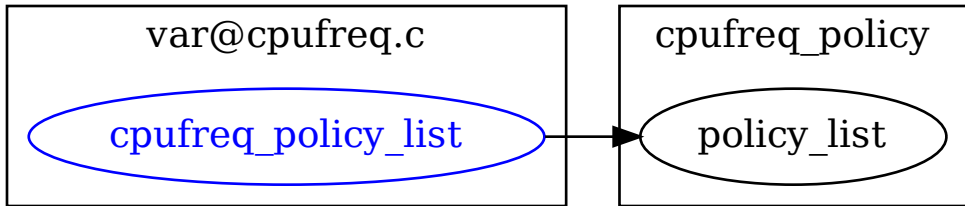


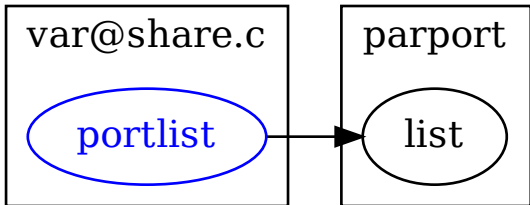


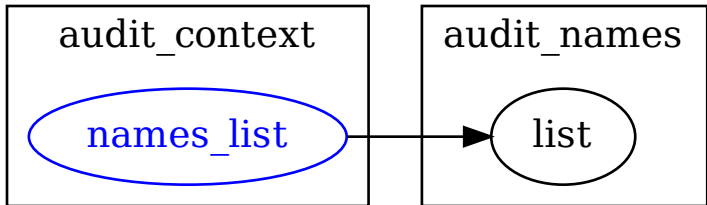


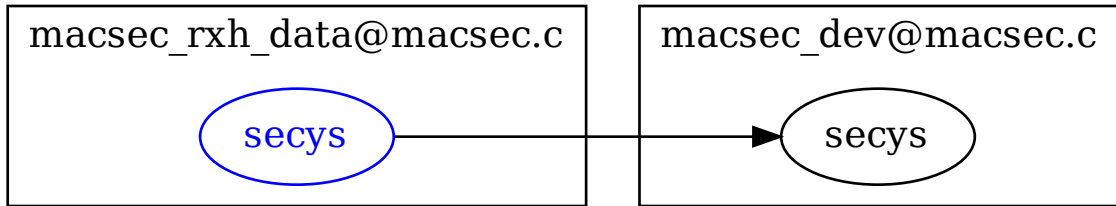












var@snapshot.c

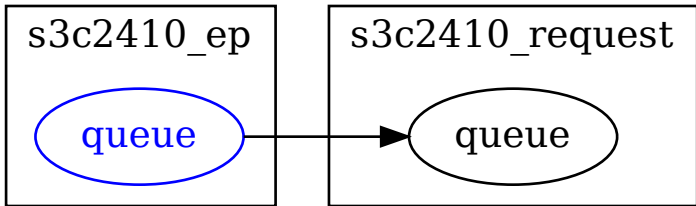
nosave_regions

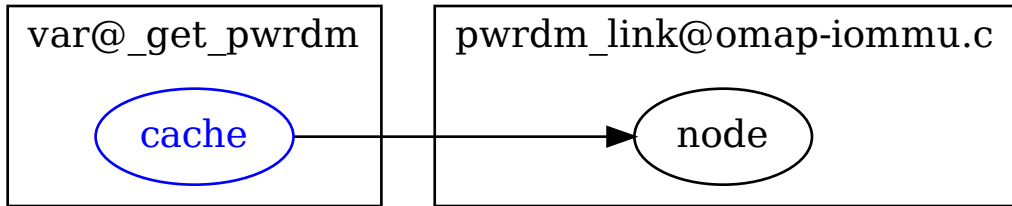
nosave_region@snapshot.c

list

```
graph LR; subgraph var_snapshot_c [var@snapshot.c]; nosave_regions; end; subgraph nosave_region_snapshot_c [nosave_region@snapshot.c]; list; end; nosave_regions --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@snapshot.c' contains a blue oval labeled 'nosave_regions'. On the right, a box labeled 'nosave_region@snapshot.c' contains an oval labeled 'list'. A horizontal arrow points from the 'nosave_regions' oval to the 'list' oval, indicating that the 'nosave_regions' variable in the first module points to the 'list' variable in the second module.





var@component.c

component_list

component@component.c

node

```
graph LR; subgraph var [var@component.c]; direction TB; CL([component_list]); end; subgraph comp [component@component.c]; direction TB; N([node]); end; CL --> N;
```

The diagram illustrates a pointer relationship between two components. On the left, a box labeled 'var@component.c' contains an oval labeled 'component_list' in blue text. On the right, a box labeled 'component@component.c' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'component_list' oval to the left side of the 'node' oval, indicating that 'component_list' points to 'node'.

omap_sham_drv@omap-sham.c

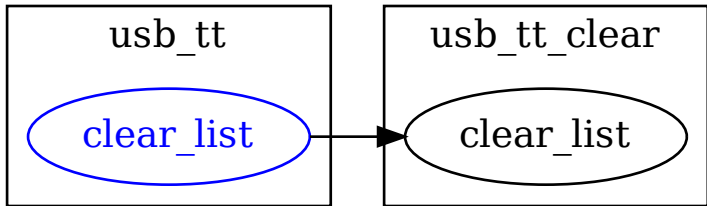
dev_list

omap_sham_dev@omap-sham.c

list

```
graph LR; subgraph "omap_sham_drv@omap-sham.c"; dev_list; end; subgraph "omap_sham_dev@omap-sham.c"; list; end; dev_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'omap_sham_drv@omap-sham.c' contains a blue oval labeled 'dev_list'. On the right, a box labeled 'omap_sham_dev@omap-sham.c' contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'dev_list' oval to the left side of the 'list' oval, indicating that 'dev_list' points to 'list'.



var@dma-heap.c

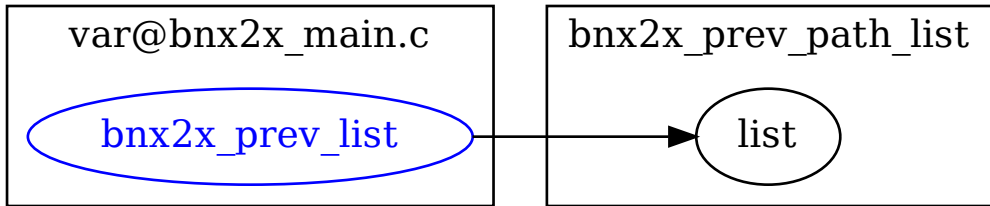
heap_list

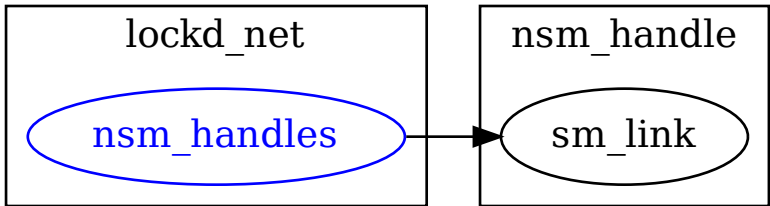
dma_heap@dma-heap.c

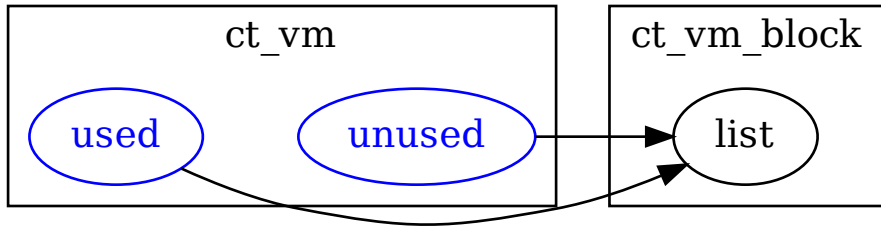
list

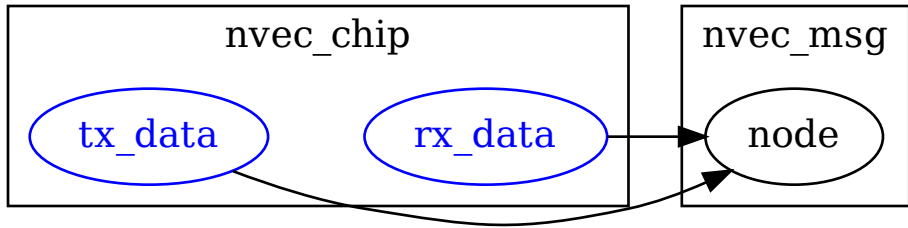
```
graph LR; subgraph var_at_dma_heap_c [var@dma-heap.c]; heap_list; end; subgraph dma_heap_at_dma_heap_c [dma_heap@dma-heap.c]; list; end; heap_list --> list;
```

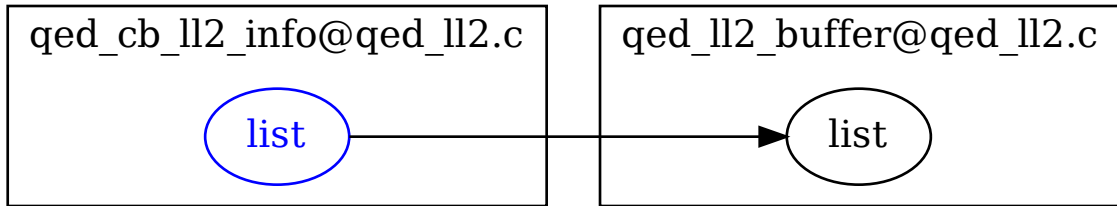
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@dma-heap.c' contains a blue oval labeled 'heap_list'. On the right, a box labeled 'dma_heap@dma-heap.c' contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'heap_list' oval to the left side of the 'list' oval, indicating that 'heap_list' points to 'list'.

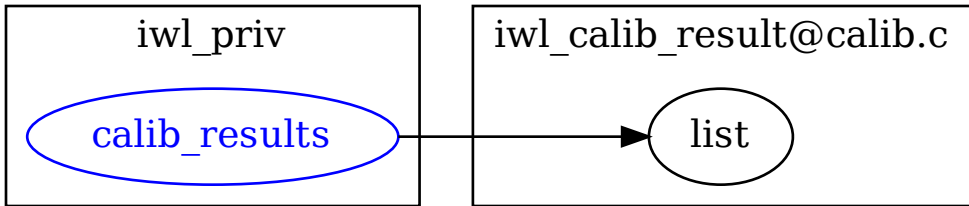


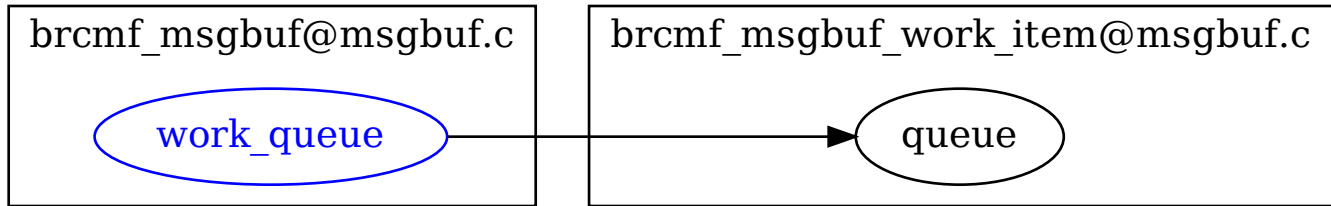


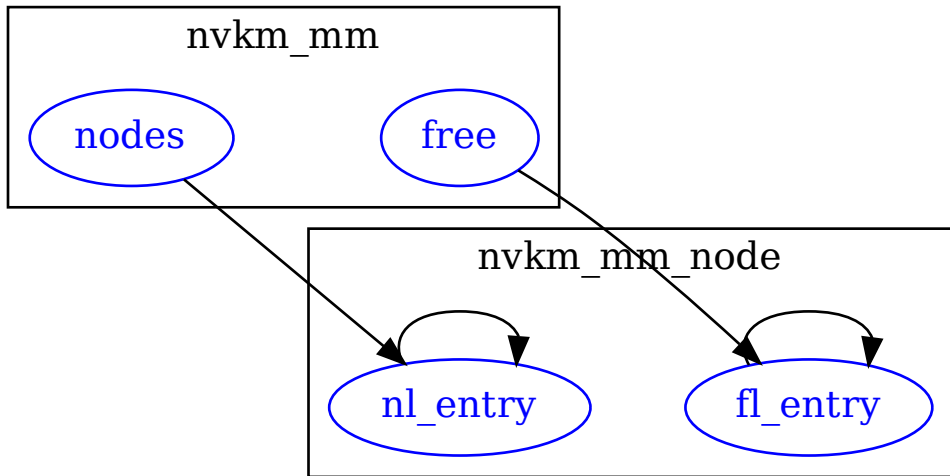


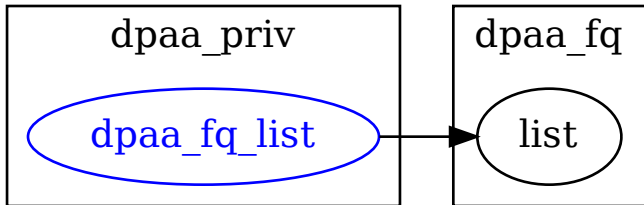


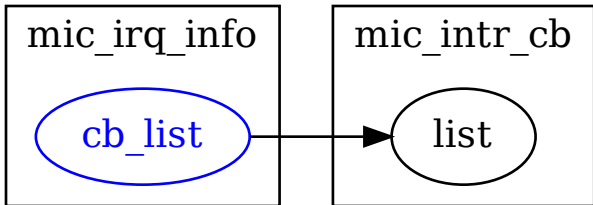


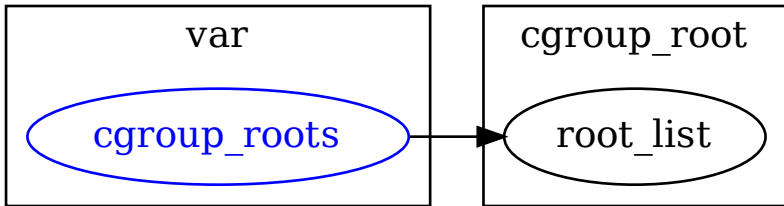


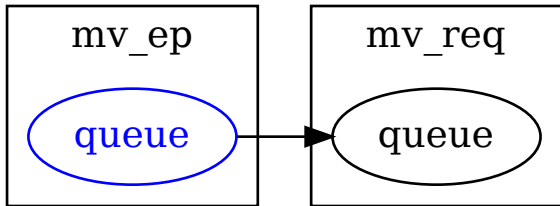


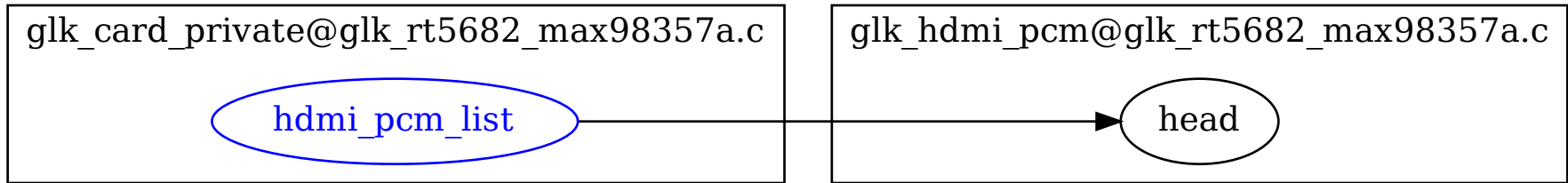


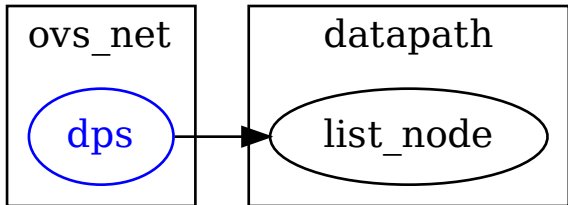


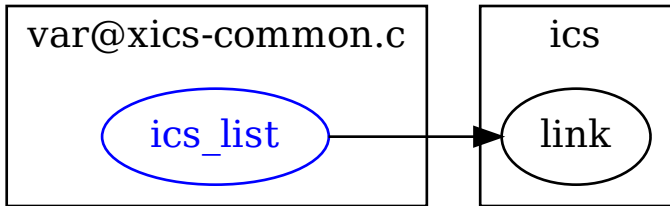


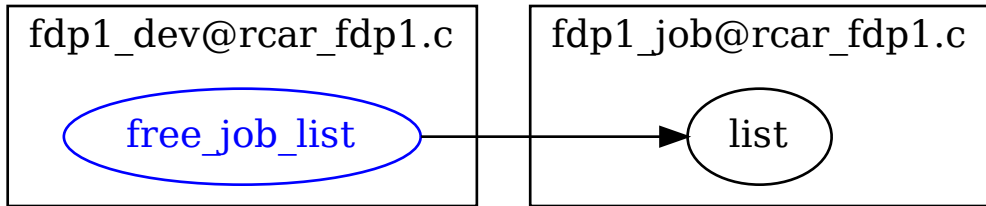


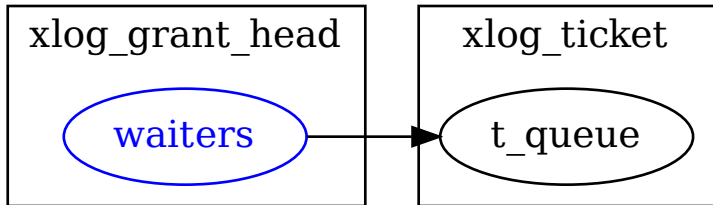












cpcap_battery_ddata@cpcap-battery.c

irq_list

cpcap_interrupt_desc@cpcap-battery.c

node

```
graph LR; A[cpcap_battery_ddata@cpcap-battery.c] -- irq_list --> B[cpcap_interrupt_desc@cpcap-battery.c]; B -- node --> C[ ];
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'cpcap_battery_ddata@cpcap-battery.c' contains an oval labeled 'irq_list'. On the right, a box labeled 'cpcap_interrupt_desc@cpcap-battery.c' contains an oval labeled 'node'. A horizontal arrow points from the 'irq_list' oval to the 'node' oval, indicating that 'irq_list' points to 'node'.

var@initramfs.c

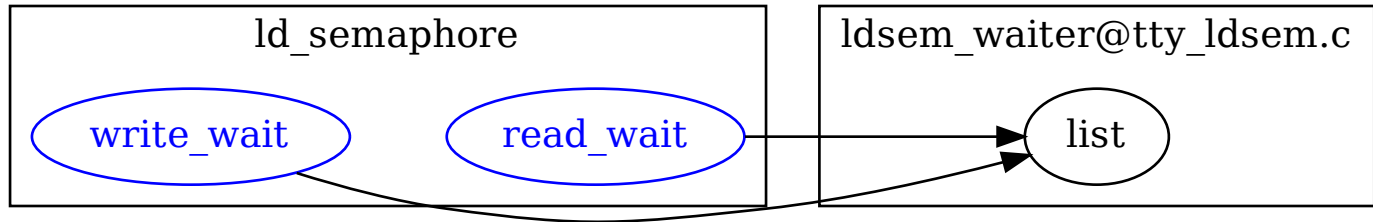
dir_list

dir_entry@initramfs.c

list

```
graph LR; subgraph var_initramfs_c [var@initramfs.c]; dir_list; end; subgraph dir_entry_initramfs_c [dir_entry@initramfs.c]; list; end; dir_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@initramfs.c' contains an oval labeled 'dir_list'. On the right, a box labeled 'dir_entry@initramfs.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dir_list' oval to the 'list' oval, indicating that 'dir_list' points to 'list'.



sta2x11_vip@sta2x11_vip.c

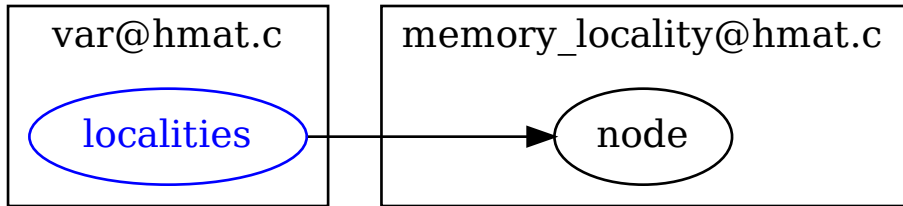
buffer_list

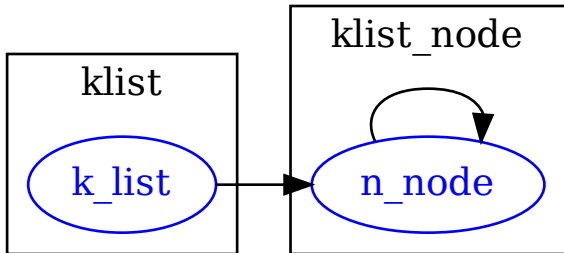
vip_buffer@sta2x11_vip.c

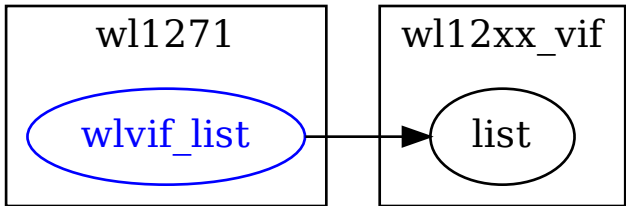
list

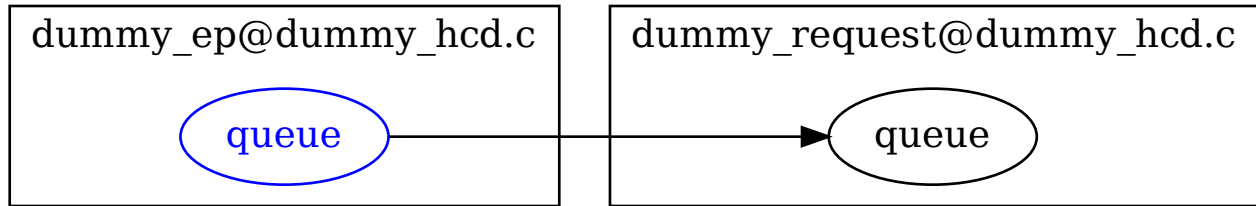
```
graph LR; subgraph Module1 [sta2x11_vip@sta2x11_vip.c]; direction TB; BL(buffer_list); end; subgraph Module2 [vip_buffer@sta2x11_vip.c]; direction TB; L(list); end; BL --> L;
```

The diagram illustrates a pointer relationship between two modules. On the left, a rectangular box represents the module 'sta2x11_vip@sta2x11_vip.c', containing an oval labeled 'buffer_list' in blue text. On the right, another rectangular box represents the module 'vip_buffer@sta2x11_vip.c', containing an oval labeled 'list'. A horizontal arrow points from the 'buffer_list' oval to the 'list' oval, indicating that 'buffer_list' points to 'list'.









urb_list@smscufx.c

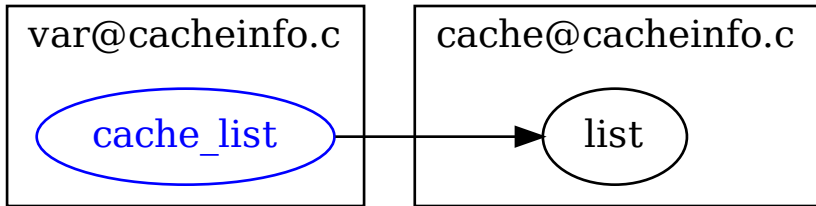
list

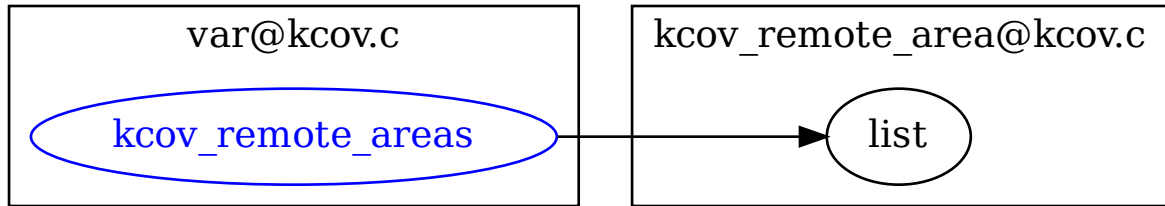
urb_node@smscufx.c

entry

```
graph LR; subgraph "urb_list@smscufx.c"; list((list)); end; subgraph "urb_node@smscufx.c"; entry((entry)); end; list --> entry;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'urb_list@smscufx.c' contains a blue oval labeled 'list'. On the right, a box labeled 'urb_node@smscufx.c' contains a black oval labeled 'entry'. A horizontal arrow points from the 'list' oval to the 'entry' oval, indicating that the 'list' variable points to the 'entry' structure.





var@speedfax.c

Cards

sfax_hw@speedfax.c

list

```
graph LR; subgraph Box1 [var@speedfax.c]; Cards([Cards]); end; subgraph Box2 [sfax_hw@speedfax.c]; list([list]); end; Cards --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@speedfax.c' and contains an oval labeled 'Cards' in blue text. The right box is labeled 'sfax_hw@speedfax.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'Cards' oval to the left side of the 'list' oval.

var@ibmphp_ebda.c

opt_vg_head

opt_rio

opt_rio_list



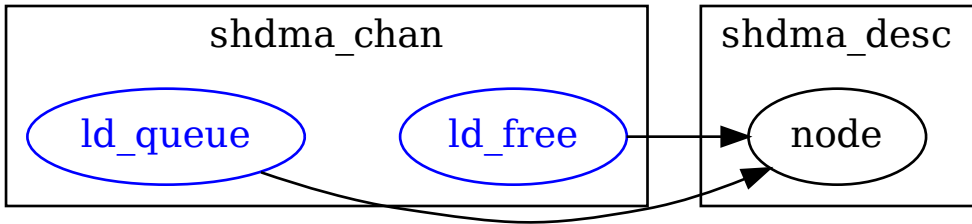
var@sync_debug.c

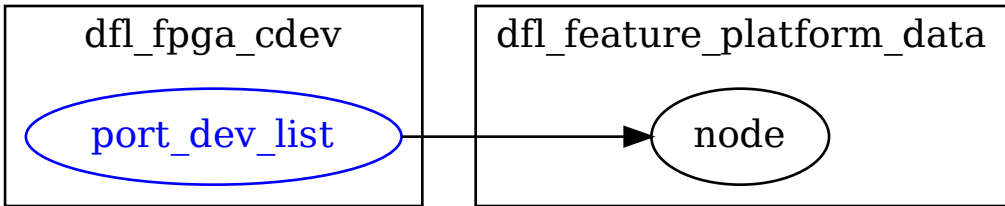
sync_file_list_head

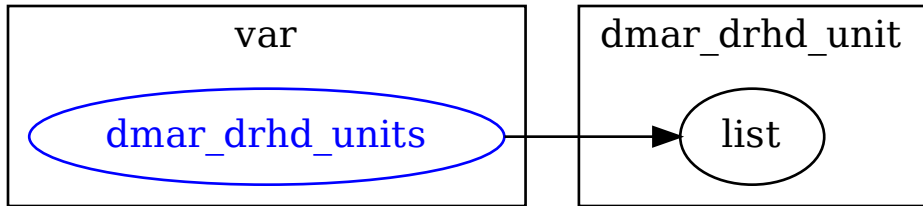
sync_file

sync_file_list









block_lock@dm-block-manager.c

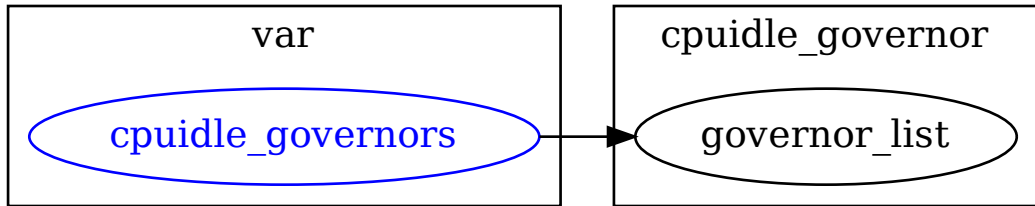
waiters

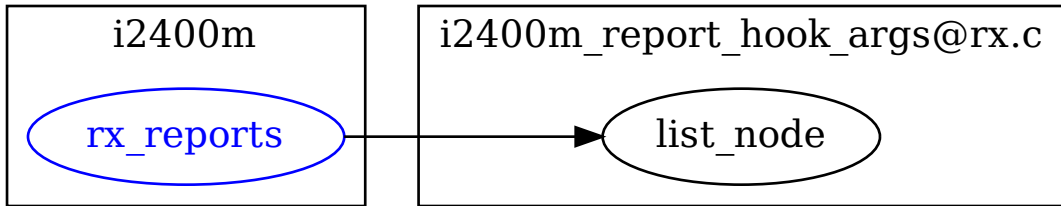
waiter@dm-block-manager.c

list

```
graph LR; subgraph "block_lock@dm-block-manager.c"; waiters((waiters)); end; subgraph "waiter@dm-block-manager.c"; list((list)); end; waiters --> list;
```

The diagram consists of two rectangular boxes. The left box is titled 'block_lock@dm-block-manager.c' and contains a blue oval labeled 'waiters'. The right box is titled 'waiter@dm-block-manager.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'waiters' oval to the left side of the 'list' oval, indicating a pointer relationship.





skl_rt286_private@skl_rt286.c

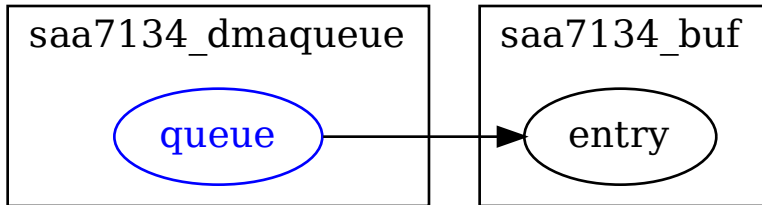
hdmi_pcm_list

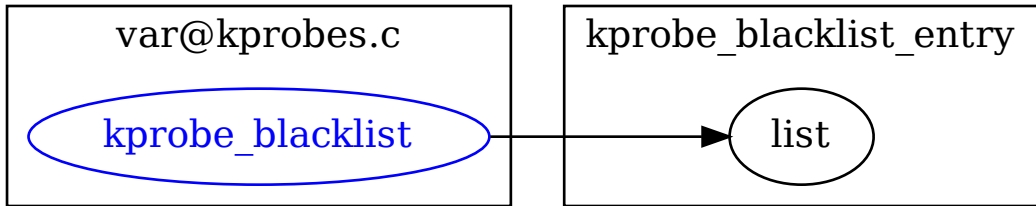
skl_hdmi_pcm@skl_rt286.c

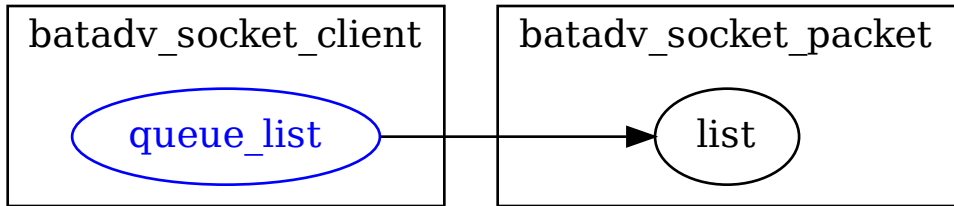
head

```
graph LR; A[skl_rt286_private@skl_rt286.c] --> B[skl_hdmi_pcm@skl_rt286.c]; subgraph A [skl_rt286_private@skl_rt286.c]; C(hdmi_pcm_list); end; subgraph B [skl_hdmi_pcm@skl_rt286.c]; D(head); end; C --> D;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'skl_rt286_private@skl_rt286.c' contains an oval labeled 'hdmi_pcm_list' in blue text. On the right, a box labeled 'skl_hdmi_pcm@skl_rt286.c' contains an oval labeled 'head' in black text. A horizontal arrow points from the 'hdmi_pcm_list' oval to the 'head' oval, indicating that the 'hdmi_pcm_list' variable points to the 'head' variable.







var@filecache.c

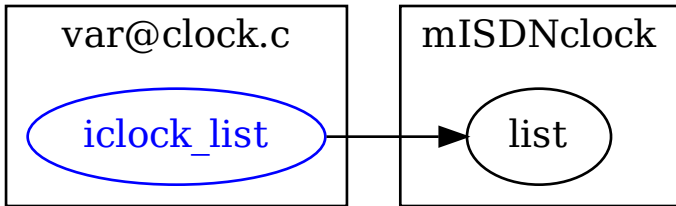
laundrettes

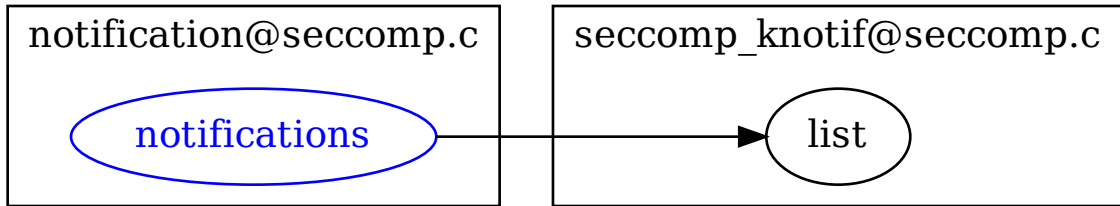
nfsd_fcach_disposal@filecache.c

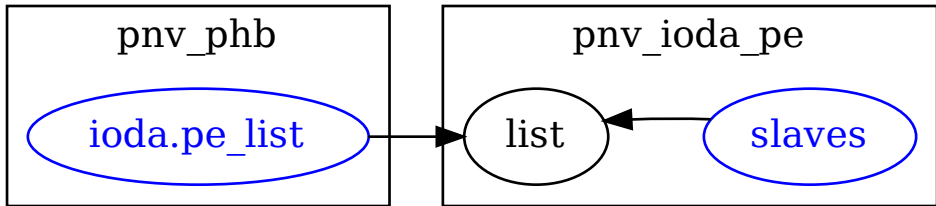
list

```
graph LR; subgraph Box1 [var@filecache.c]; laundrettes; end; subgraph Box2 [nfsd_fcach_disposal@filecache.c]; list; end; laundrettes --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@filecache.c' and contains a blue oval labeled 'laundrettes'. The right box is labeled 'nfsd_fcach_disposal@filecache.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'laundrettes' oval to the left side of the 'list' oval.







tee_net@xt_TEE.c

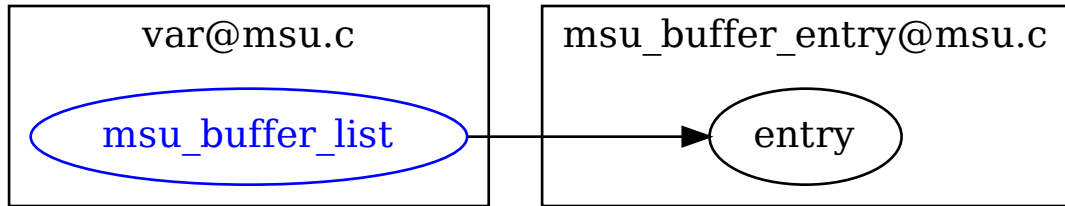
priv_list

xt_tee_priv@xt_TEE.c

list

```
graph LR; subgraph Module1 [tee_net@xt_TEE.c]; priv_list; end; subgraph Module2 [xt_tee_priv@xt_TEE.c]; list; end; priv_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'tee_net@xt_TEE.c' contains an oval labeled 'priv_list'. On the right, a box labeled 'xt_tee_priv@xt_TEE.c' contains an oval labeled 'list'. A horizontal arrow points from the 'priv_list' oval to the 'list' oval, indicating that 'priv_list' points to 'list'.



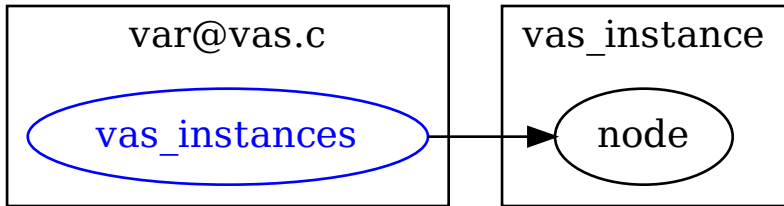
ets_sched@sch_ets.c

active

ets_class@sch_ets.c

alist





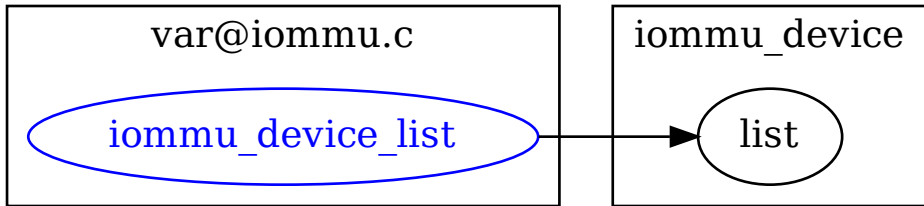
var@dm-exception-store.c

exception_store_types

dm_exception_store_type

list

The diagram consists of two rectangular boxes. The left box contains the text 'var@dm-exception-store.c' and a blue oval containing the text '_exception_store_types'. The right box contains the text 'dm_exception_store_type' and an oval containing the text 'list'. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the 'list' oval in the right box.



var@avmfritz.c

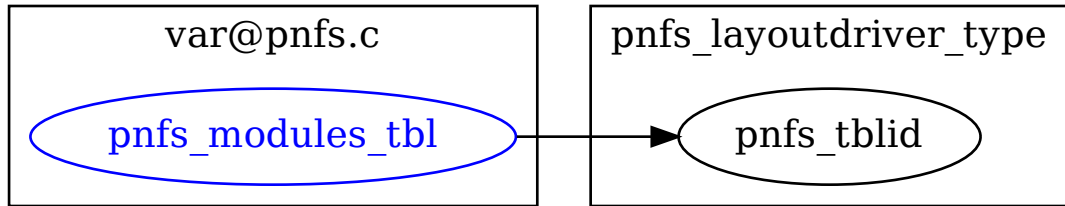
Cards

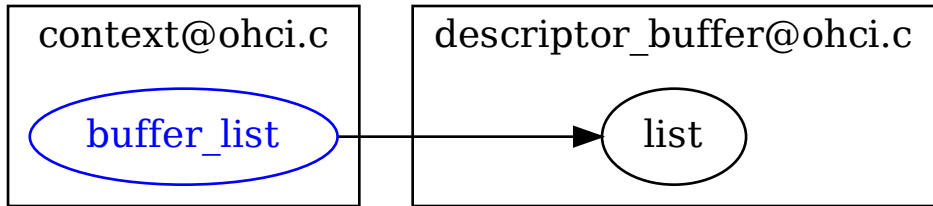
fritzcard@avmfritz.c

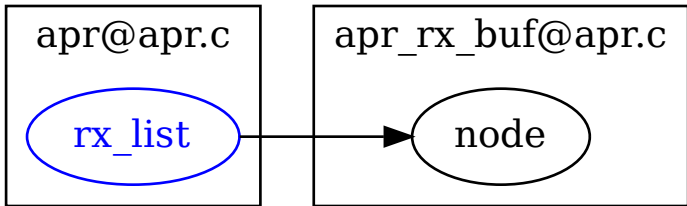
list

```
graph LR; subgraph Box1 [var@avmfritz.c]; Cards([Cards]); end; subgraph Box2 [fritzcard@avmfritz.c]; list([list]); end; Cards --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@avmfritz.c' and contains an oval labeled 'Cards' in blue text. The right box is labeled 'fritzcard@avmfritz.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'Cards' oval to the left side of the 'list' oval.







roccat_device@hid-roccat.c

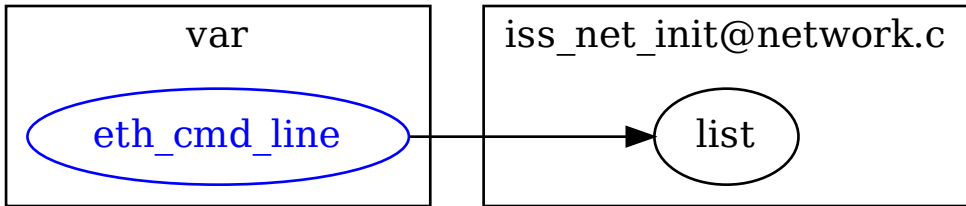
readers

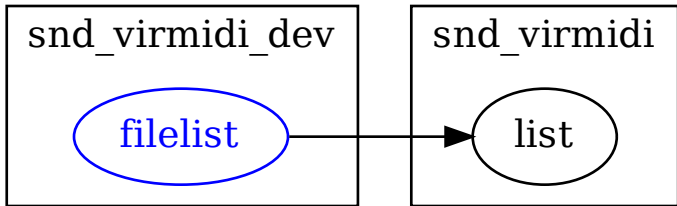
roccat_reader@hid-roccat.c

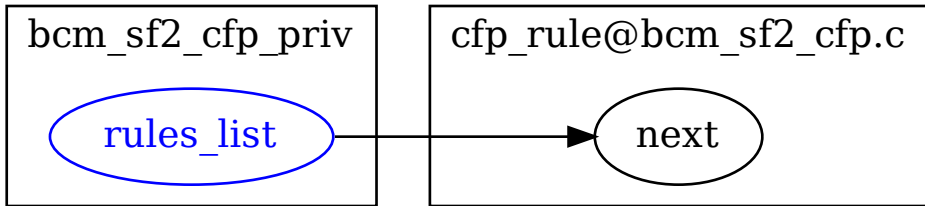
node

```
graph LR; subgraph Box1 [roccat_device@hid-roccat.c]; readers; end; subgraph Box2 [roccat_reader@hid-roccat.c]; node; end; readers --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'roccat_device@hid-roccat.c' and contains an oval labeled 'readers' in blue text. The right box is labeled 'roccat_reader@hid-roccat.c' and contains an oval labeled 'node' in black text. A horizontal arrow points from the 'readers' oval in the first box to the 'node' oval in the second box.







acpi_memory_device@xen-acpi-memhotplug.c

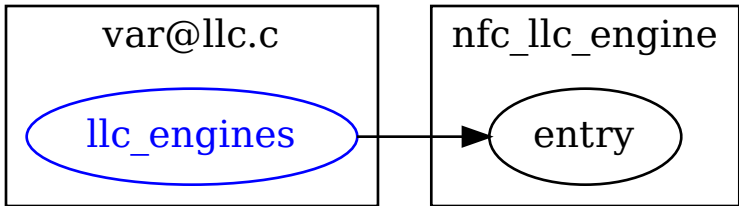
res_list

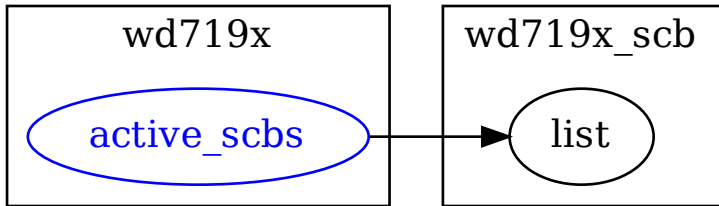
acpi_memory_info@xen-acpi-memhotplug.c

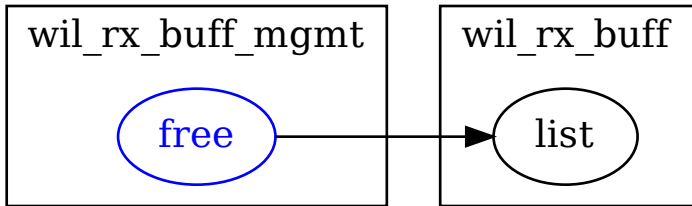
list

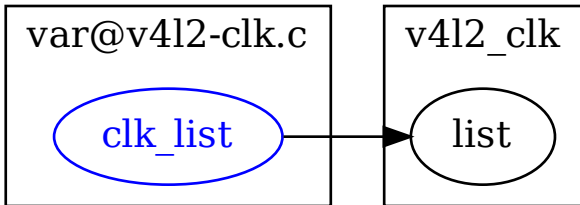
```
graph LR; A[acpi_memory_device@xen-acpi-memhotplug.c] -- res_list --> B[acpi_memory_info@xen-acpi-memhotplug.c]; B -- list --> C((list));
```

The diagram illustrates a pointer relationship between two modules. The first module, `acpi_memory_device@xen-acpi-memhotplug.c`, contains a variable `res_list` (highlighted in a blue oval). An arrow points from `res_list` to the second module, `acpi_memory_info@xen-acpi-memhotplug.c`. Inside the second module, there is a variable `list` (enclosed in a black oval) which is the target of the arrow.









sa11x0_dma_dev@sa11x0-dma.c

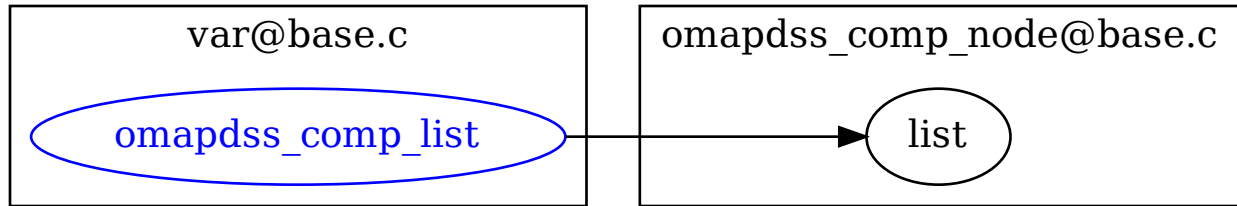
chan_pending

sa11x0_dma_chan@sa11x0-dma.c

node

```
graph LR; subgraph "sa11x0_dma_dev@sa11x0-dma.c"; direction TB; A(chan_pending); end; subgraph "sa11x0_dma_chan@sa11x0-dma.c"; direction TB; B(node); end; A --> B;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box represents the 'sa11x0_dma_dev@sa11x0-dma.c' structure, containing an oval labeled 'chan_pending' in blue text. On the right, another rectangular box represents the 'sa11x0_dma_chan@sa11x0-dma.c' structure, containing an oval labeled 'node'. A horizontal arrow points from the 'chan_pending' oval to the 'node' oval, indicating that the 'chan_pending' pointer in the first structure points to the 'node' in the second structure.



var@dfs_cache.c

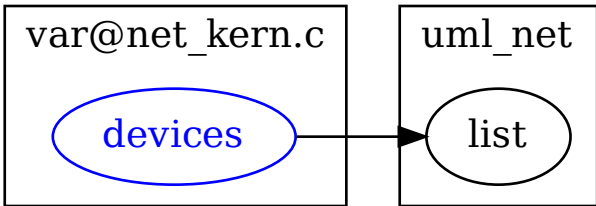
vol_list

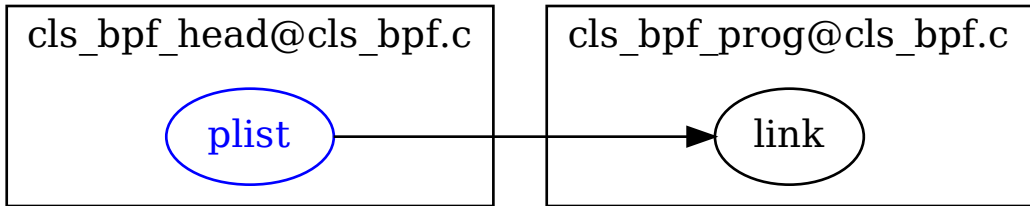
vol_info@dfs_cache.c

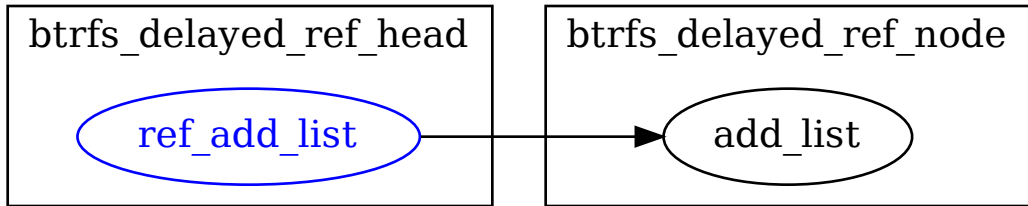
list

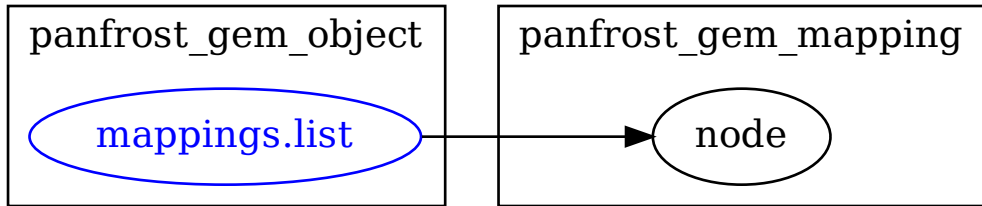
```
graph LR; subgraph var_at_dfs_cache_c [var@dfs_cache.c]; vol_list; end; subgraph vol_info_at_dfs_cache_c [vol_info@dfs_cache.c]; list; end; vol_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@dfs_cache.c' contains an oval labeled 'vol_list'. On the right, a box labeled 'vol_info@dfs_cache.c' contains an oval labeled 'list'. A horizontal arrow points from the 'vol_list' oval to the 'list' oval, indicating that 'vol_list' points to 'list'.









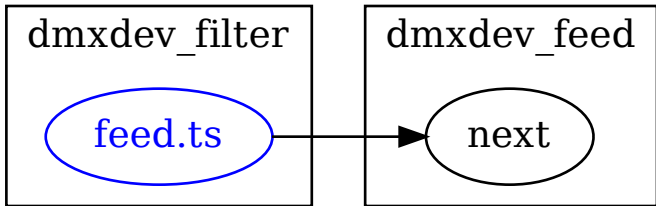
dm_integrity_c@dm-integrity.c

wait_list

dm_integrity_range@dm-integrity.c

wait_entry





var@stratix10-svc.c

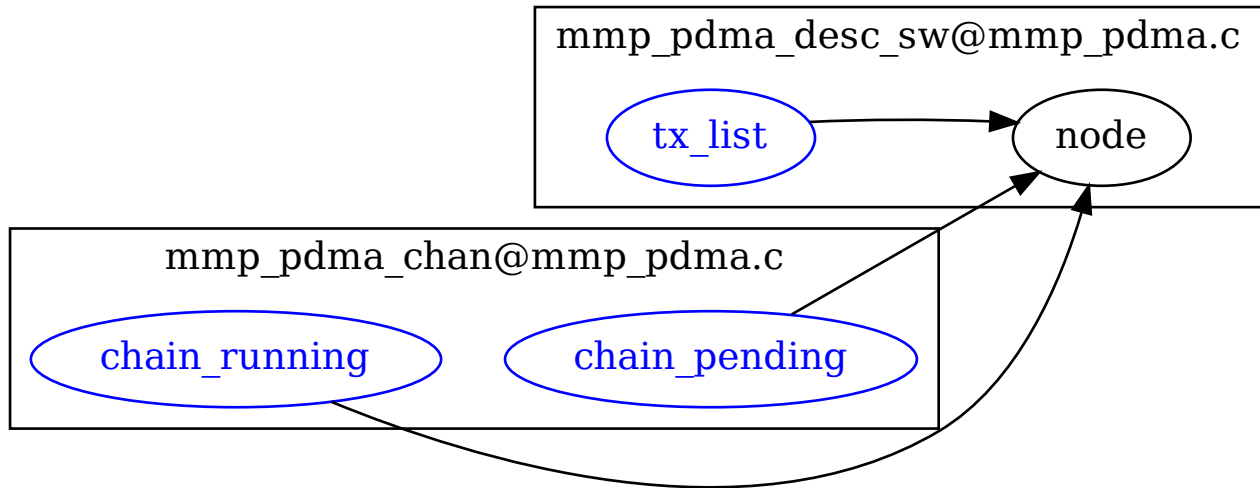
svc_ctrl

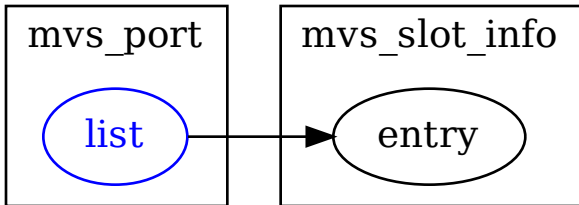
stratix10_svc_controller@stratix10-svc.c

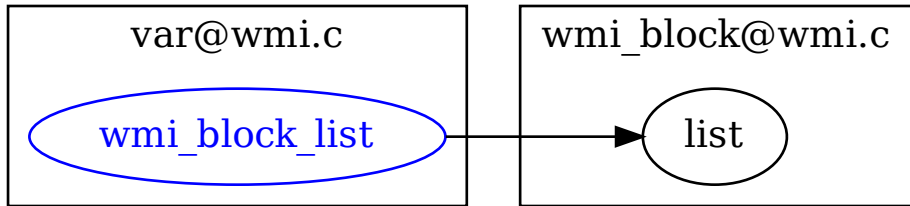
node

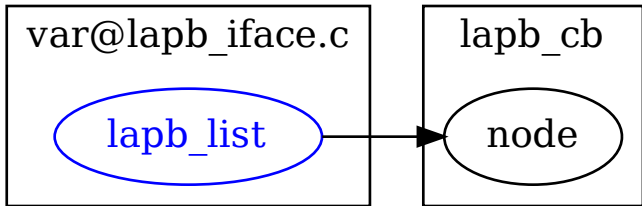
```
graph LR; subgraph Box1 [var@stratix10-svc.c]; direction TB; svc_ctrl([svc_ctrl]); end; subgraph Box2 [stratix10_svc_controller@stratix10-svc.c]; direction TB; node([node]); end; Box1 --> Box2;
```

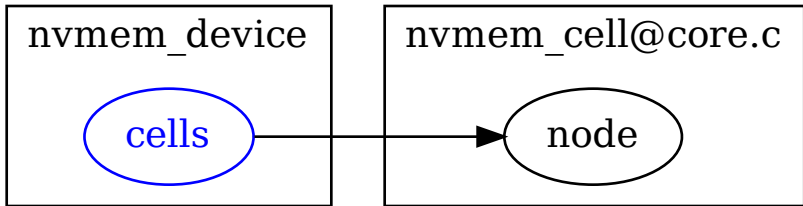
The diagram consists of two rectangular boxes. The left box is labeled 'var@stratix10-svc.c' and contains an oval labeled 'svc_ctrl'. The right box is labeled 'stratix10_svc_controller@stratix10-svc.c' and contains an oval labeled 'node'. A horizontal arrow points from the 'svc_ctrl' oval in the first box to the 'node' oval in the second box.











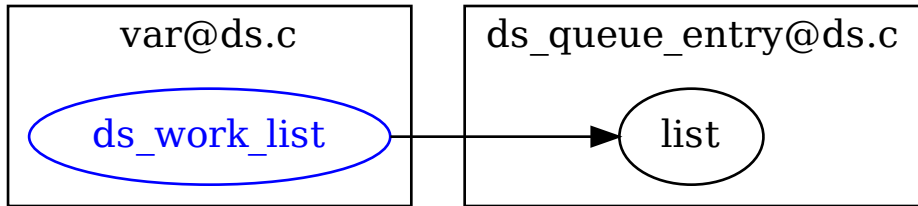
mlx5dr_icm_pool@dr_icm_pool.c

icm_mr_list

mlx5dr_icm_mr@dr_icm_pool.c

mr_list





var@xilinx-vtc.c

xvtc_list

xvtc_device@xilinx-vtc.c

list

```
graph LR; subgraph var [var@xilinx-vtc.c]; xvtc_list; end; subgraph xvtc_device [xvtc_device@xilinx-vtc.c]; list; end; xvtc_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@xilinx-vtc.c' contains an oval labeled 'xvtc_list'. On the right, a box labeled 'xvtc_device@xilinx-vtc.c' contains an oval labeled 'list'. A horizontal arrow points from the 'xvtc_list' oval to the 'list' oval, indicating that 'xvtc_list' points to 'list'.

vsc8531_private@mscc.c

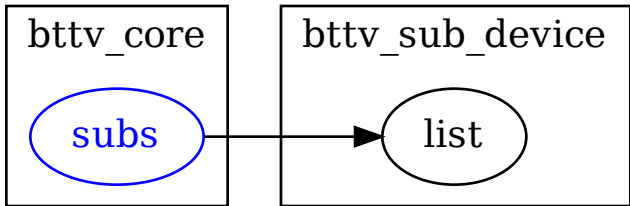
macsec_flows

macsec_flow@mscc.c

list

```
graph LR; subgraph Box1 [vsc8531_private@mscc.c]; direction TB; A([macsec_flows]); end; subgraph Box2 [macsec_flow@mscc.c]; direction TB; B([list]); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'vsc8531_private@mscc.c' and contains a blue oval labeled 'macsec_flows'. The right box is labeled 'macsec_flow@mscc.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'macsec_flows' oval to the left side of the 'list' oval.



gk104_ram@ramgk104.c

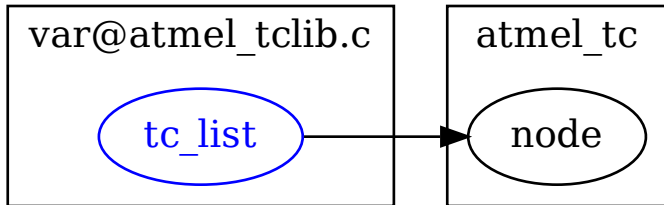
cfg

nvkm_ram_data

head

```
graph LR; subgraph Box1 [gk104_ram@ramgk104.c]; cfg((cfg)); end; subgraph Box2 [nvkm_ram_data]; head((head)); end; cfg --> head;
```

The diagram consists of two rectangular boxes. The left box is labeled 'gk104_ram@ramgk104.c' and contains an oval labeled 'cfg' with a blue border. The right box is labeled 'nvkm_ram_data' and contains an oval labeled 'head'. A horizontal arrow points from the 'cfg' oval to the 'head' oval.



var@xenbus_client.c

xenbus_valloc_pages

xenbus_map_node@xenbus_client.c

next

```
graph LR; subgraph File1 [var@xenbus_client.c]; V[xenbus_valloc_pages]; end; subgraph File2 [xenbus_map_node@xenbus_client.c]; N((next)); end; V --> N;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@xenbus_client.c`. Inside this box, the variable `xenbus_valloc_pages` is enclosed in a blue oval. On the right, another rectangular box represents the source file `xenbus_map_node@xenbus_client.c`. Inside this box, the variable `next` is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that `xenbus_valloc_pages` points to `next`.

var@dmi-sysfs.c

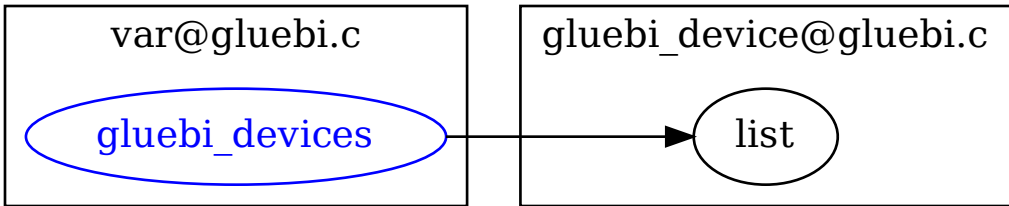
entry_list

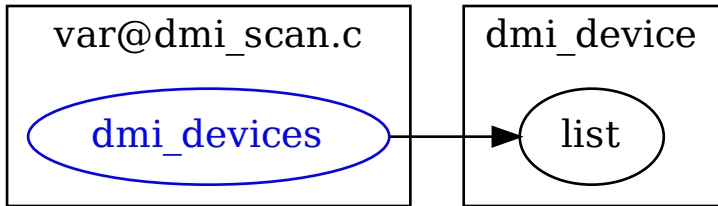
dmi_sysfs_entry@dmi-sysfs.c

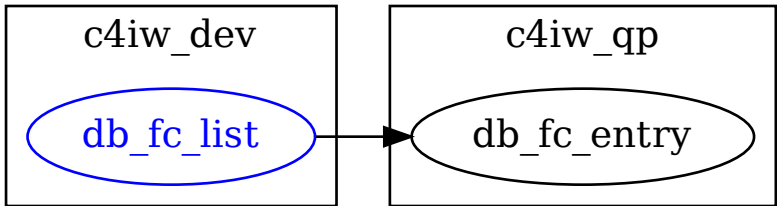
list

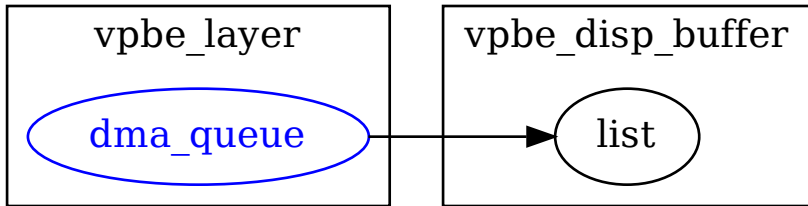
```
graph LR; A["var@dmi-sysfs.c<br/>entry_list"] --> B["dmi_sysfs_entry@dmi-sysfs.c<br/>list"]
```

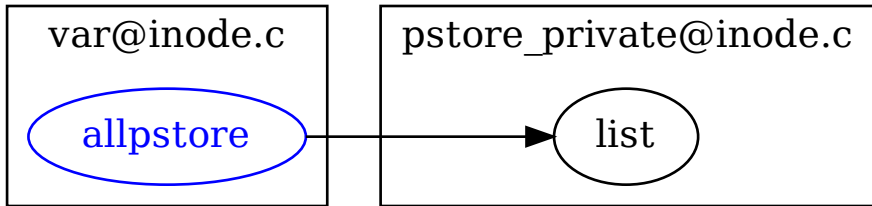
The diagram consists of two rectangular boxes. The left box contains the text 'var@dmi-sysfs.c' at the top and 'entry_list' in a blue oval below it. The right box contains the text 'dmi_sysfs_entry@dmi-sysfs.c' at the top and 'list' in a black oval below it. A horizontal arrow points from the right side of the 'entry_list' oval to the left side of the 'list' oval.

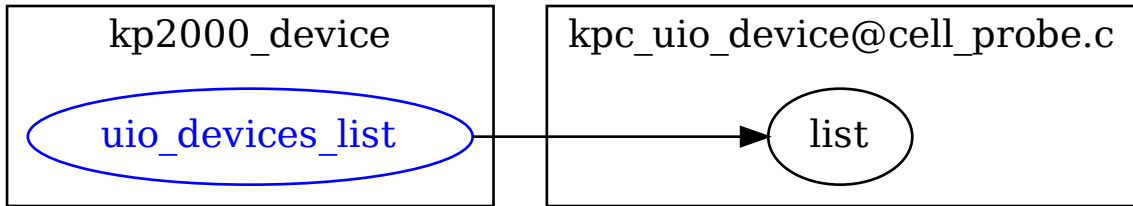


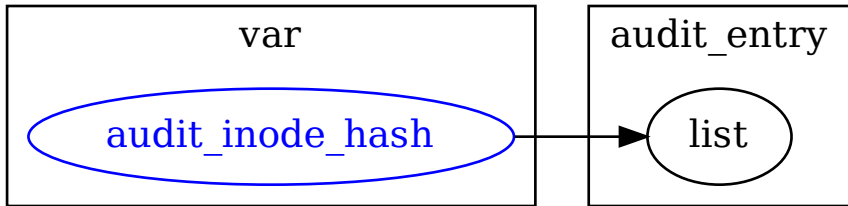


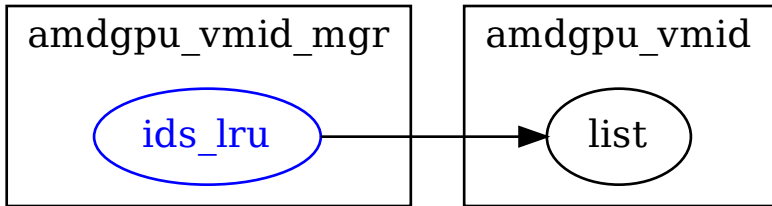


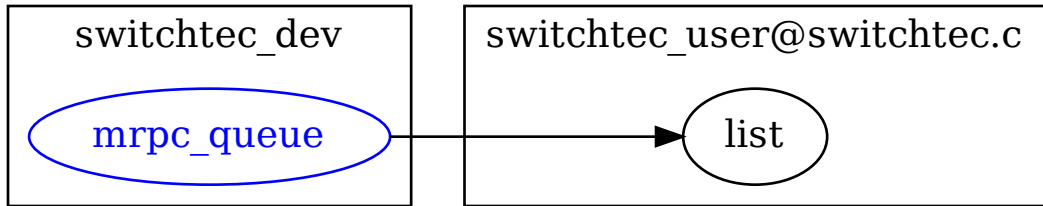












devx_async_event_queue@devx.c

event_list

devx_async_data@devx.c

list



```
graph LR; subgraph devx_async_event_queue [devx_async_event_queue@devx.c]; event_list((event_list)); end; subgraph devx_async_data [devx_async_data@devx.c]; list((list)); end; event_list --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'devx_async_event_queue@devx.c' contains an oval labeled 'event_list'. On the right, a box labeled 'devx_async_data@devx.c' contains an oval labeled 'list'. A horizontal arrow points from the 'event_list' oval to the 'list' oval, indicating that 'event_list' points to 'list'.

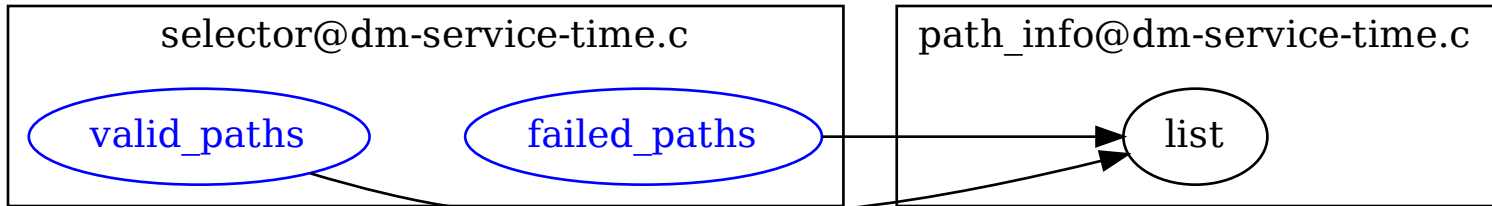
selector@dm-service-time.c

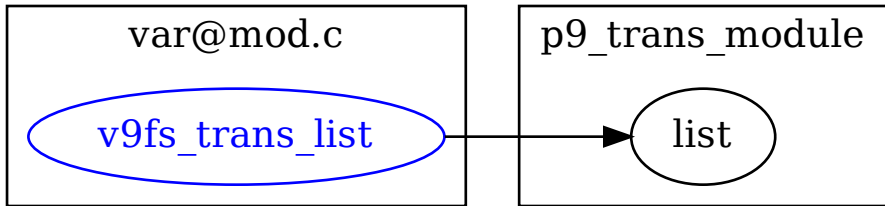
valid_paths

failed_paths

path_info@dm-service-time.c

list





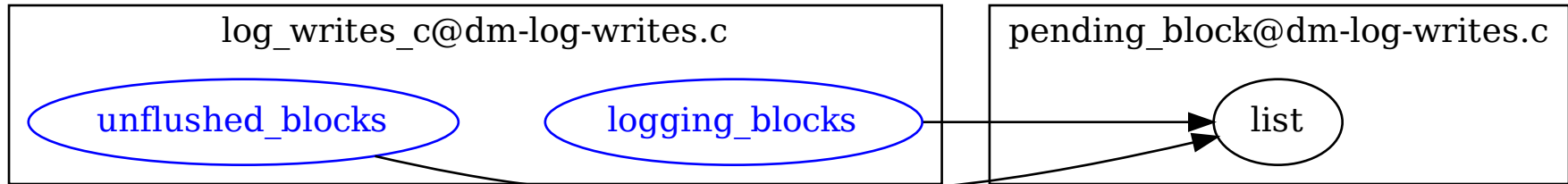
sel_netport_bkt@netport.c

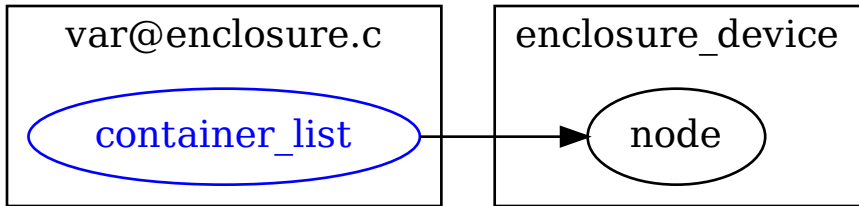
list

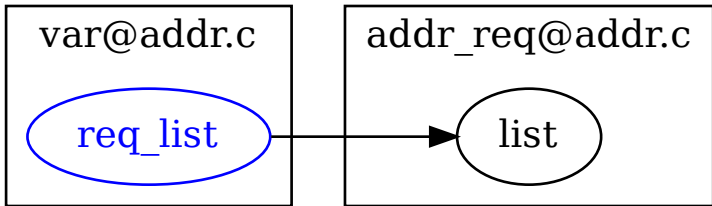
sel_netport@netport.c

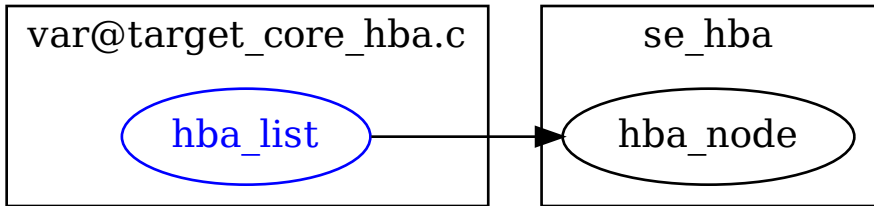
list











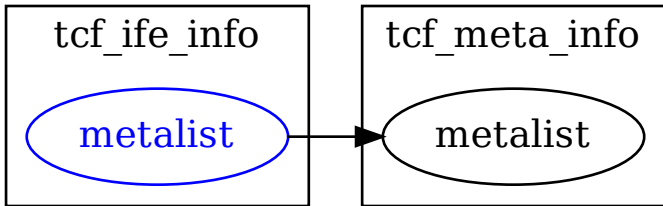
sof_card_private@sof_rt5682.c

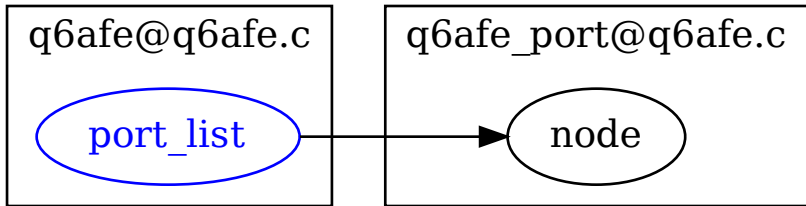
hdmi_pcm_list

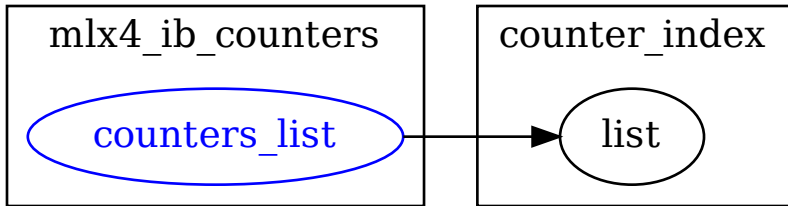
sof_hdmi_pcm@sof_rt5682.c

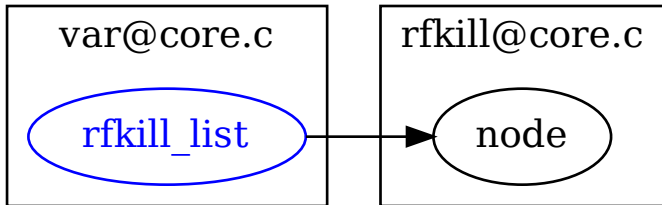
head











var@xt_IDLETIMER.c

idletimer_tg_list

idletimer_tg@xt_IDLETIMER.c

entry

```
graph LR; A["var@xt_IDLETIMER.c  
idletimer_tg_list"] --> B["idletimer_tg@xt_IDLETIMER.c  
entry"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@xt_IDLETIMER.c' contains a blue oval labeled 'idletimer_tg_list'. On the right, a box labeled 'idletimer_tg@xt_IDLETIMER.c' contains a black oval labeled 'entry'. A horizontal arrow points from the right side of the 'idletimer_tg_list' oval to the left side of the 'entry' oval, indicating that 'idletimer_tg_list' points to 'entry'.

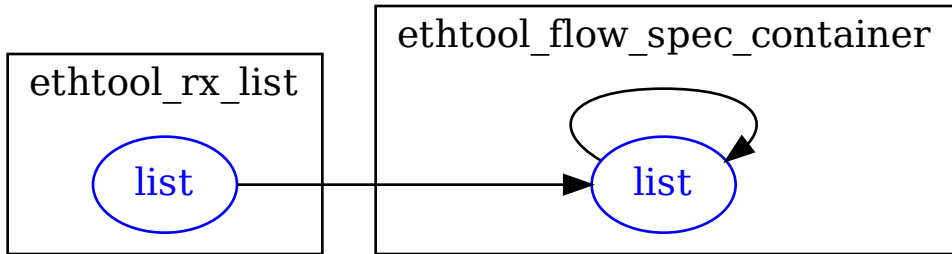
cdns_nand_ctrl@cadence-nand-controller.c

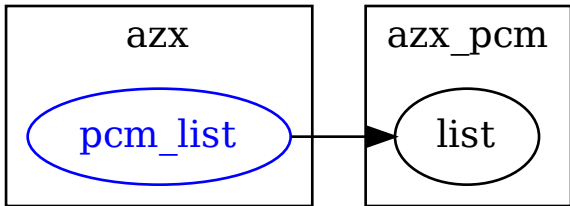
chips

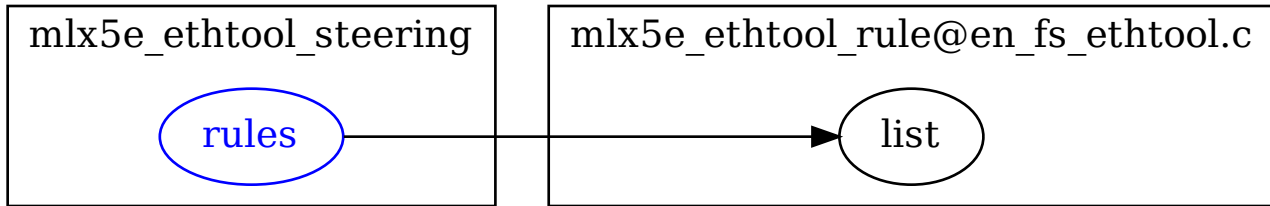
cdns_nand_chip@cadence-nand-controller.c

node

```
graph LR; subgraph LeftBox [cdns_nand_ctrl@cadence-nand-controller.c]; chips; end; subgraph RightBox [cdns_nand_chip@cadence-nand-controller.c]; node; end; chips --> node;
```







devlink_fmsg@devlink.c

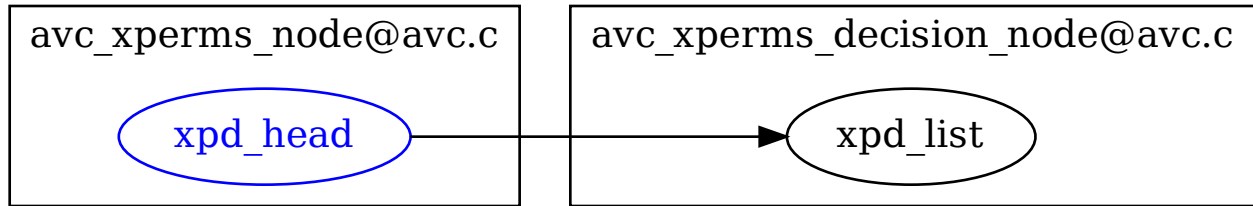
item_list

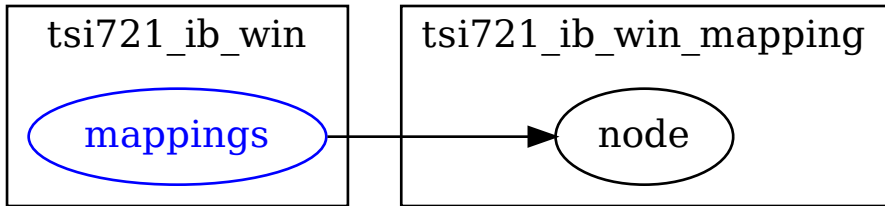
devlink_fmsg_item@devlink.c

list

```
graph LR; A[devlink_fmsg@devlink.c] --> B[devlink_fmsg_item@devlink.c]; subgraph A [devlink_fmsg@devlink.c]; A1([item_list]); end; subgraph B [devlink_fmsg_item@devlink.c]; B1([list]); end; A1 --> B1;
```

The diagram illustrates a pointer relationship between two structs. On the left, a box represents the `devlink_fmsg@devlink.c` struct, containing a blue oval labeled `item_list`. On the right, a box represents the `devlink_fmsg_item@devlink.c` struct, containing a black oval labeled `list`. A horizontal arrow points from the `item_list` oval to the `list` oval, indicating that the `item_list` member of the first struct points to the `list` member of the second struct.





var@switchdev.c

deferred

switchdev_deferred_item@switchdev.c

list

```
graph LR; subgraph Box1 [var@switchdev.c]; deferred([deferred]); end; subgraph Box2 [switchdev_deferred_item@switchdev.c]; list([list]); end; deferred --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@switchdev.c' and contains a blue oval labeled 'deferred'. The right box is labeled 'switchdev_deferred_item@switchdev.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'deferred' oval to the left side of the 'list' oval.

var@vector_kern.c

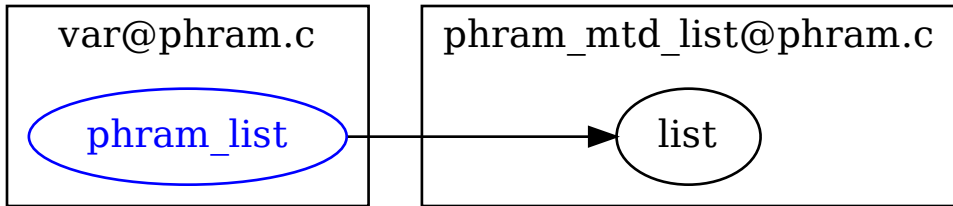
vector_devices

vector_device@vector_kern.c

list



```
graph LR; subgraph Box1 [var@vector_kern.c]; direction TB; VD([vector_devices]); end; subgraph Box2 [vector_device@vector_kern.c]; direction TB; L([list]); end; VD --> L;
```



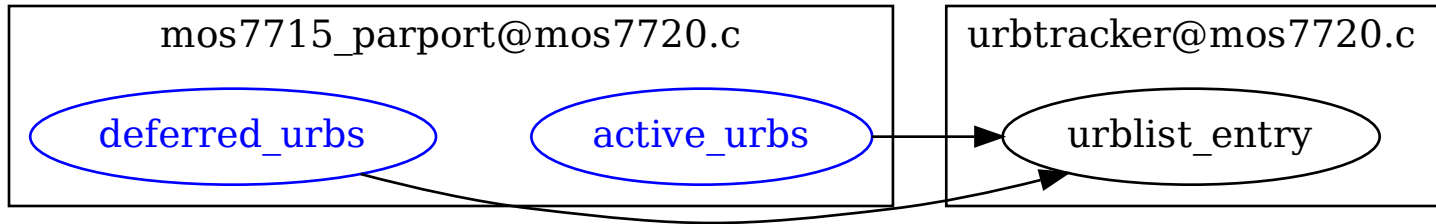
mos7715_parport@mos7720.c

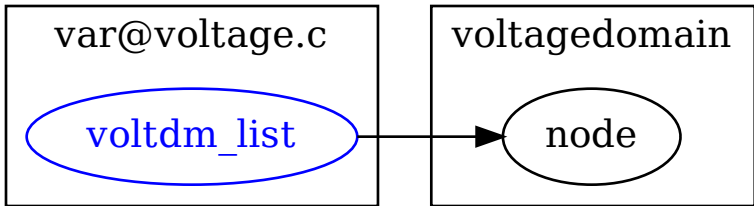
deferred_urbs

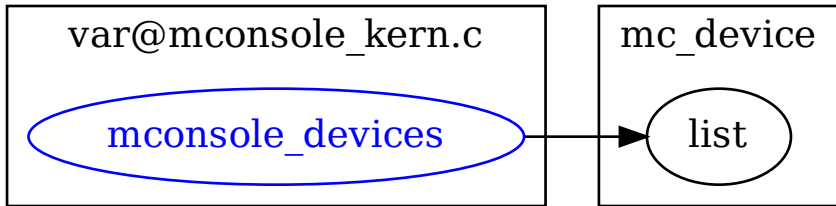
active_urbs

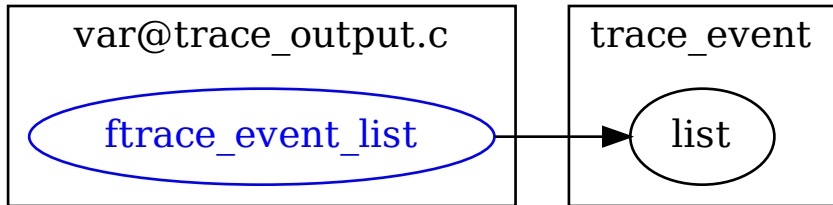
urbtracker@mos7720.c

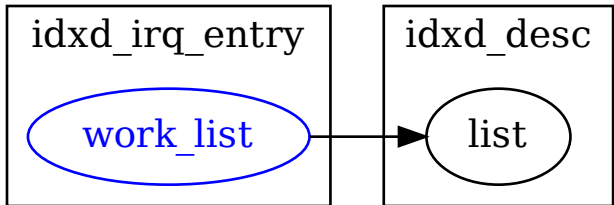
urblast_entry











mlxsw_sp_acl_erp_table@spectrum_acl_erp.c

atcam_erps_list

mlxsw_sp_acl_erp@spectrum_acl_erp.c

list



```
graph LR; A(atcam_erps_list) --> B(list)
```

mlxsw_sp_fid_family@spectrum_fid.c

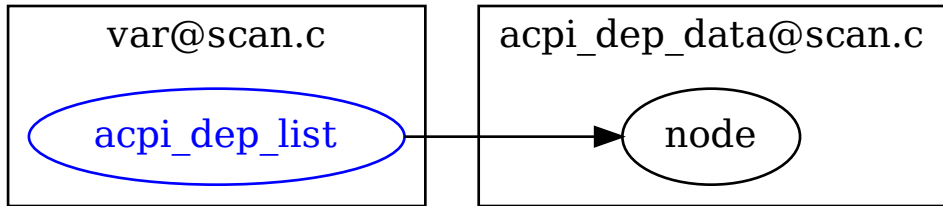
fids_list

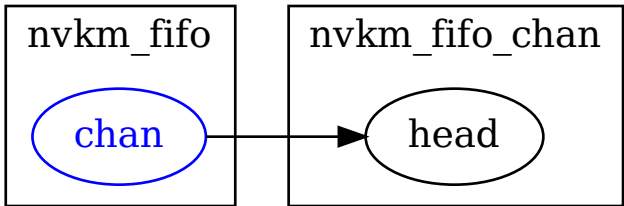
mlxsw_sp_fid@spectrum_fid.c

list

```
graph LR; A["mlxsw_sp_fid_family@spectrum_fid.c  
fids_list"] --> B["mlxsw_sp_fid@spectrum_fid.c  
list"]
```

The diagram illustrates a pointer relationship between two modules. On the left, a box represents the module `mlxsw_sp_fid_family@spectrum_fid.c`, which contains a variable `fids_list` (highlighted in a blue oval). On the right, a box represents the module `mlxsw_sp_fid@spectrum_fid.c`, which contains a variable `list` (highlighted in a black oval). A horizontal arrow points from the `fids_list` variable in the left module to the `list` variable in the right module, indicating that `fids_list` points to `list`.





var@loop.c

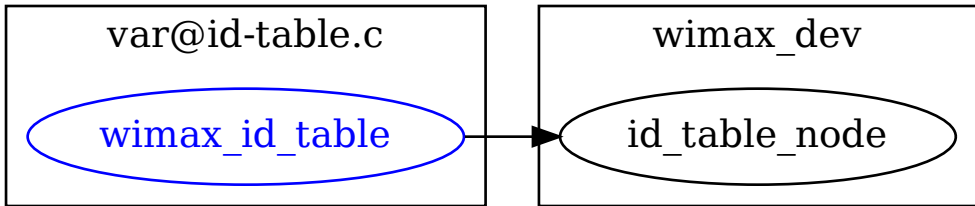
loop_conns

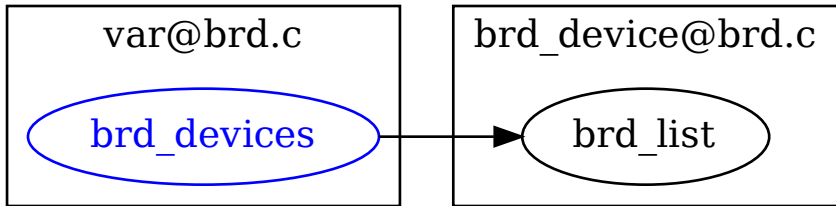
rds_loop_connection@loop.c

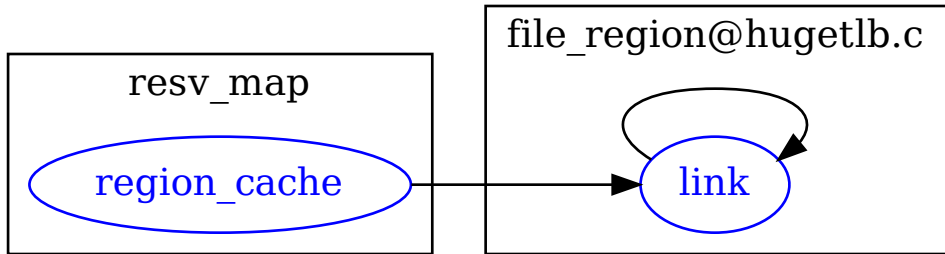
loop_node

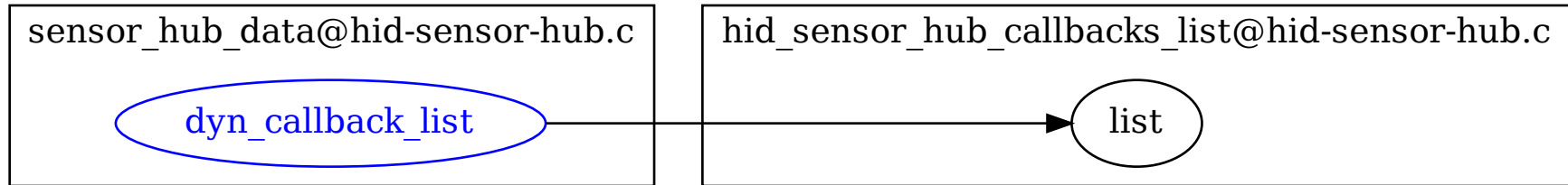
```
graph LR; subgraph var_at_loop_c [var@loop.c]; loop_conns; end; subgraph rds_loop_connection_at_loop_c [rds_loop_connection@loop.c]; loop_node; end; loop_conns --> loop_node;
```

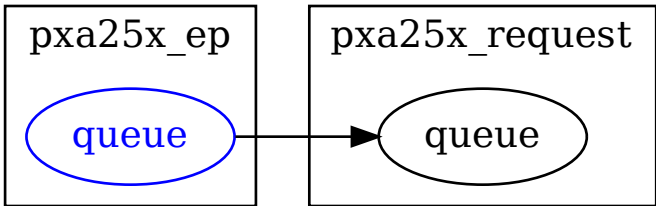
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@loop.c' contains an oval labeled 'loop_conns'. On the right, a box labeled 'rds_loop_connection@loop.c' contains an oval labeled 'loop_node'. A horizontal arrow points from the 'loop_conns' oval to the 'loop_node' oval, indicating that 'loop_conns' points to 'loop_node'.











sh_vou_device@sh_vou.c

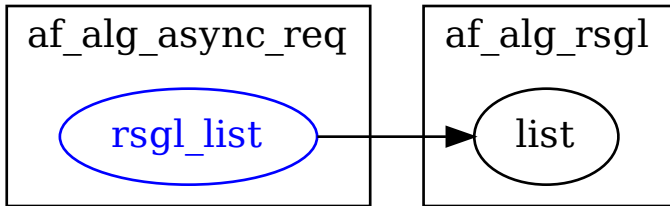
buf_list

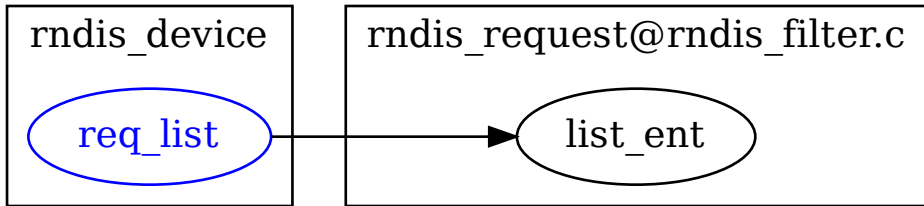
sh_vou_buffer@sh_vou.c

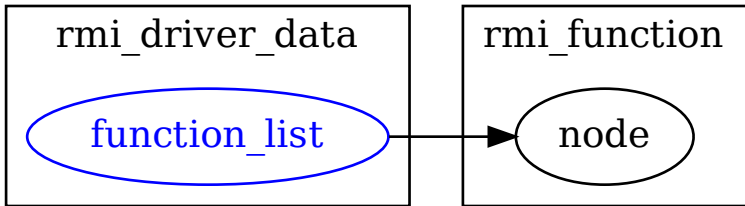
list

```
graph LR; subgraph Device [sh_vou_device@sh_vou.c]; buf_list(buf_list); end; subgraph Buffer [sh_vou_buffer@sh_vou.c]; list(list); end; buf_list --> list;
```

The diagram illustrates a pointer relationship between two memory spaces. On the left, a box labeled 'sh_vou_device@sh_vou.c' contains a blue oval labeled 'buf_list'. On the right, a box labeled 'sh_vou_buffer@sh_vou.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'buf_list' oval to the 'list' oval, indicating that 'buf_list' points to the memory location of 'list'.







sel_ib_pkey_bkt@ibpkey.c

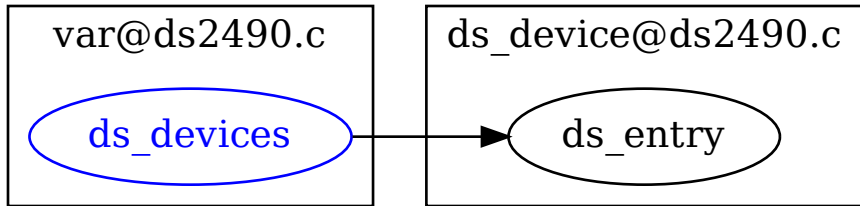
list

sel_ib_pkey@ibpkey.c

list







sun6i_dma_dev@sun6i-dma.c

pending

sun6i_vchan@sun6i-dma.c

node

```
graph LR; subgraph Box1 [sun6i_dma_dev@sun6i-dma.c]; pending; end; subgraph Box2 [sun6i_vchan@sun6i-dma.c]; node; end; pending --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'sun6i_dma_dev@sun6i-dma.c' and contains an oval labeled 'pending' with a blue border. The right box is labeled 'sun6i_vchan@sun6i-dma.c' and contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'pending' oval to the left side of the 'node' oval.

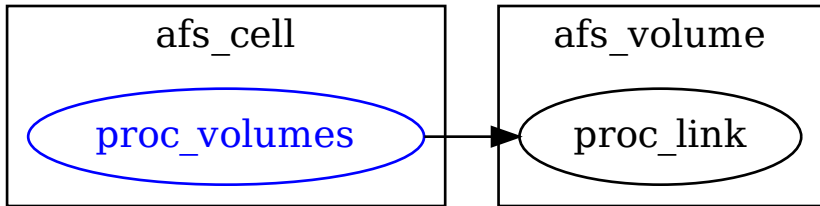
ioc@blk-iocost.c

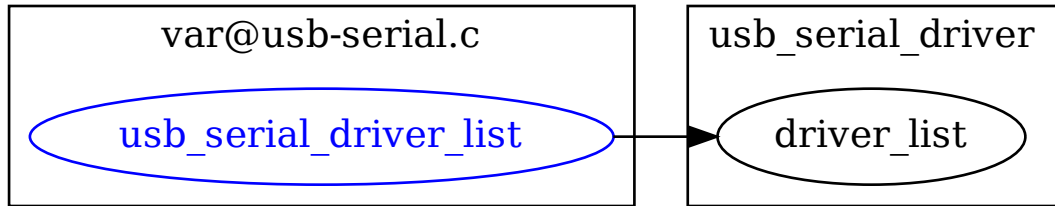
active_iocgs

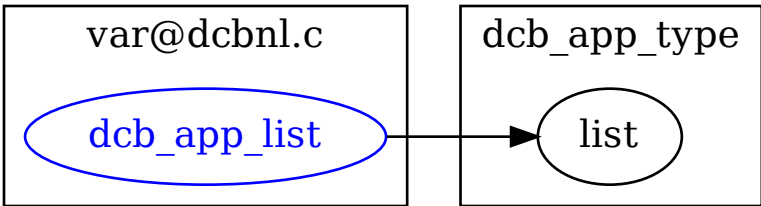
ioc_gq@blk-iocost.c

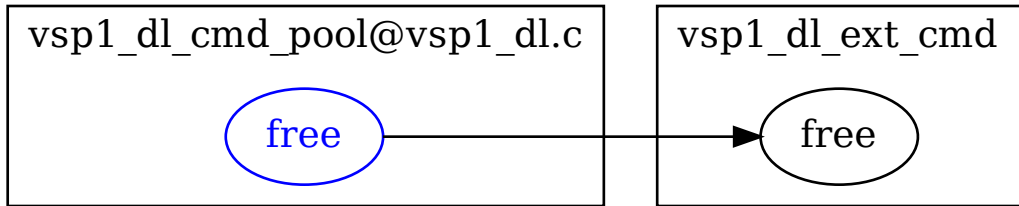
active_list

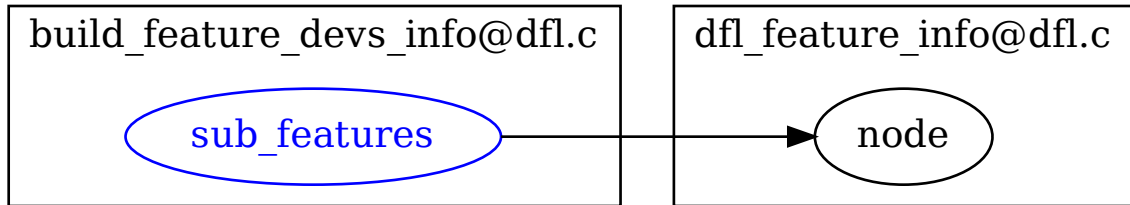












var@windfarm_smu_sensors.c

smu_ads

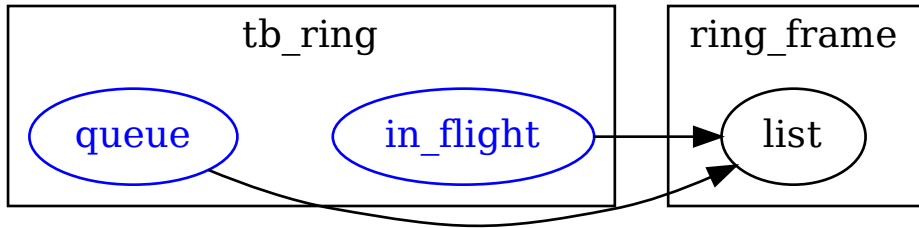
smu_ad_sensor@windfarm_smu_sensors.c

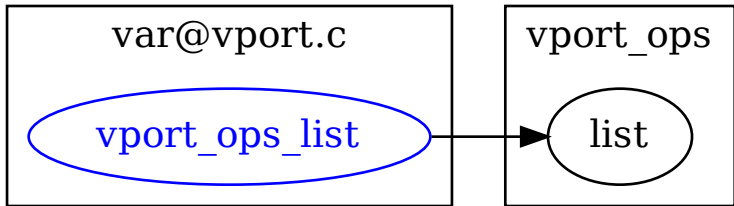
link

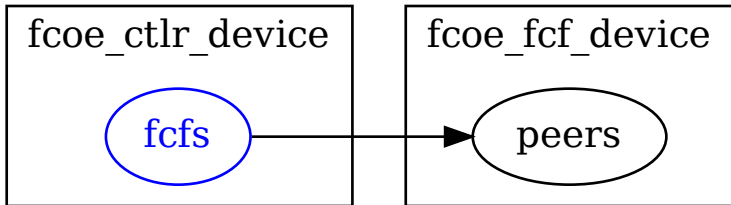
```
graph LR; A[smu_ads] --> B[link];
```

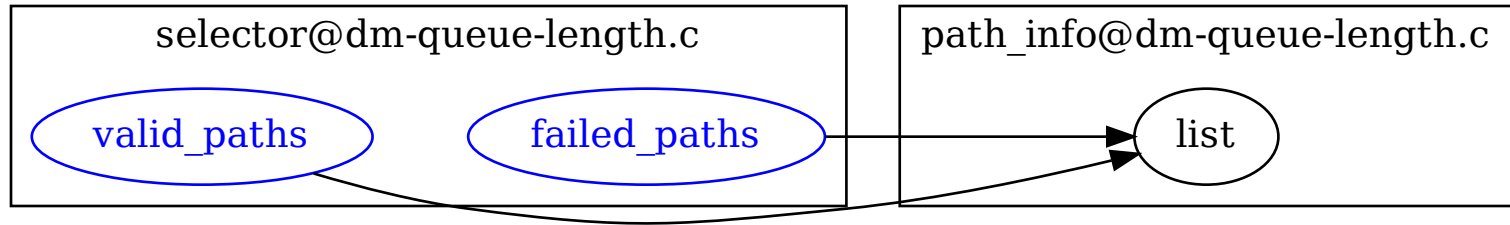
The diagram consists of two rectangular boxes. The left box contains the text 'var@windfarm_smu_sensors.c' at the top and a blue oval containing 'smu_ads' below it. The right box contains the text 'smu_ad_sensor@windfarm_smu_sensors.c' at the top and a black oval containing 'link' below it. A horizontal line with an arrowhead at the right end connects the right side of the 'smu_ads' oval to the left side of the 'link' oval.

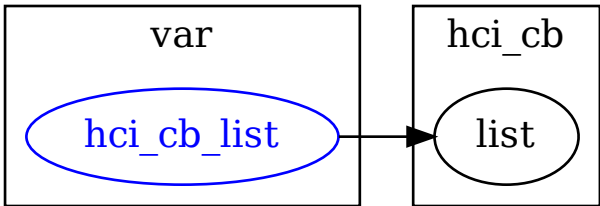












zstd_workspace_manager@zstd.c

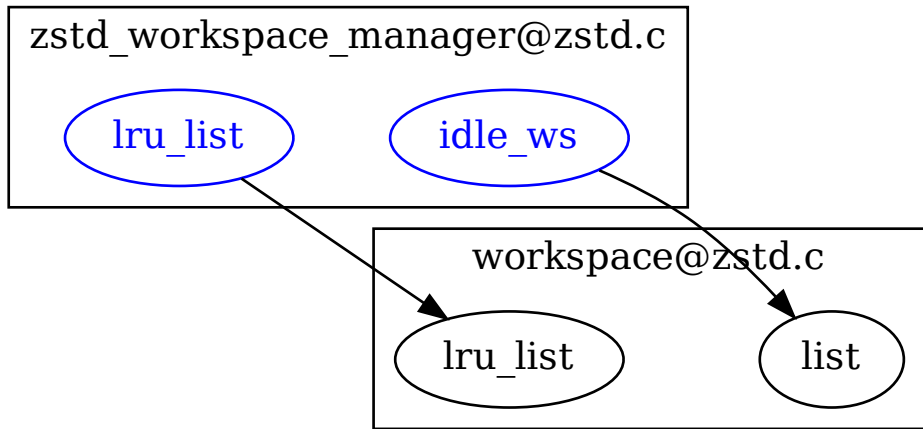
lru_list

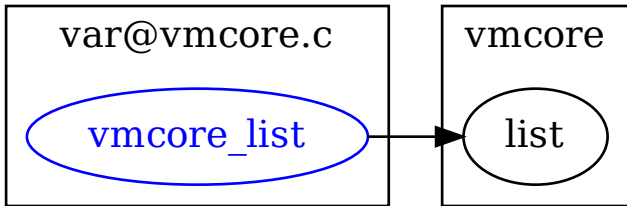
idle_ws

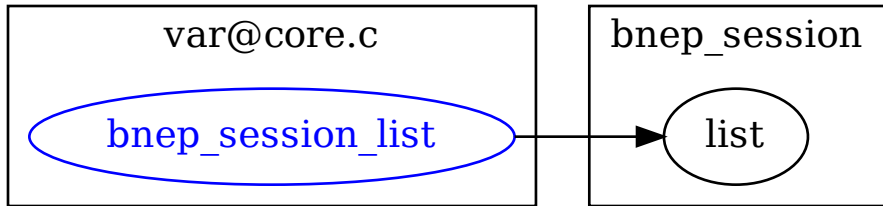
workspace@zstd.c

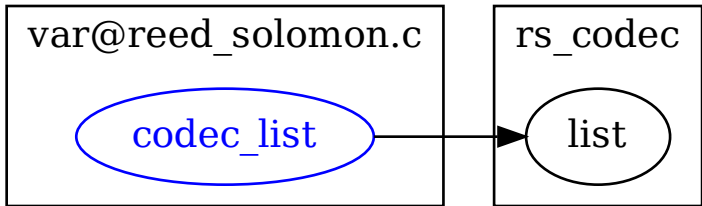
lru_list

list









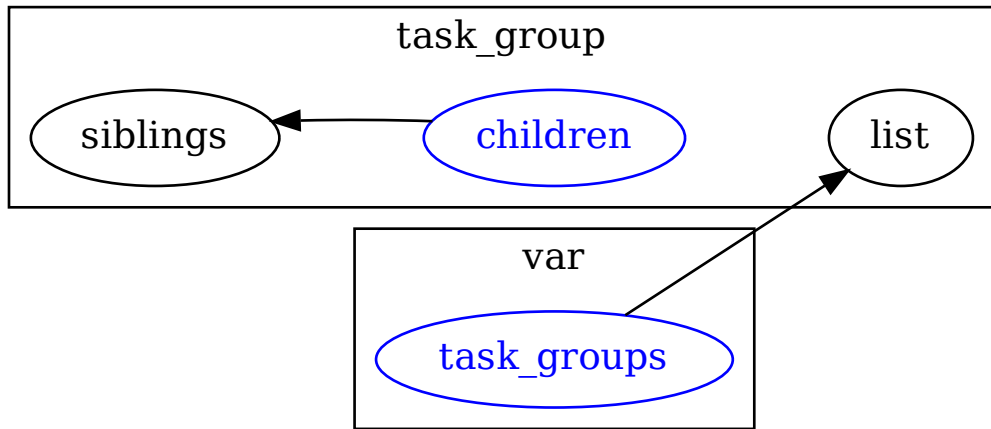
skl_nau88125_private@skl_nau88125_ssm4567.c

hdmi_pcm_list

skl_hdmi_pcm@skl_nau88125_ssm4567.c

head





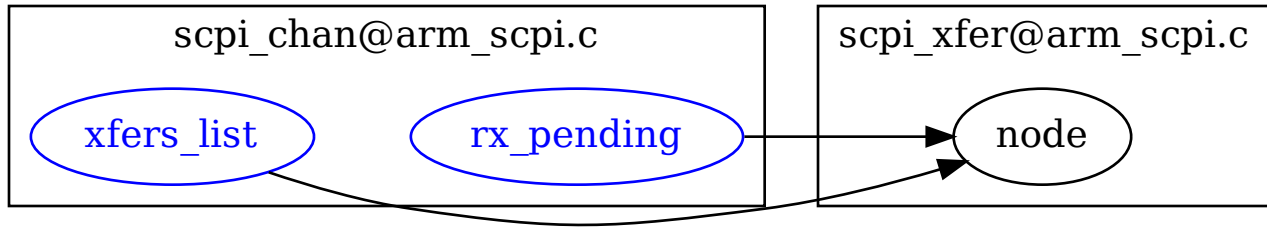
renesas_usb3_ep@renesas_usb3.c

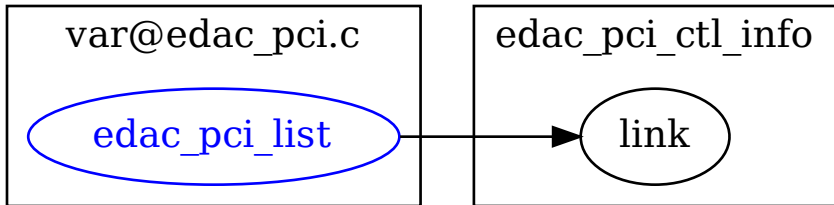
queue

renesas_usb3_request@renesas_usb3.c

queue







var@matroxfb_base.c

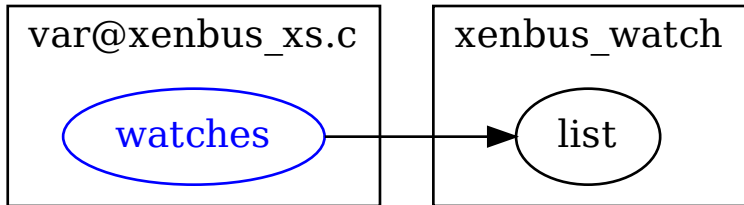
matroxfb_list

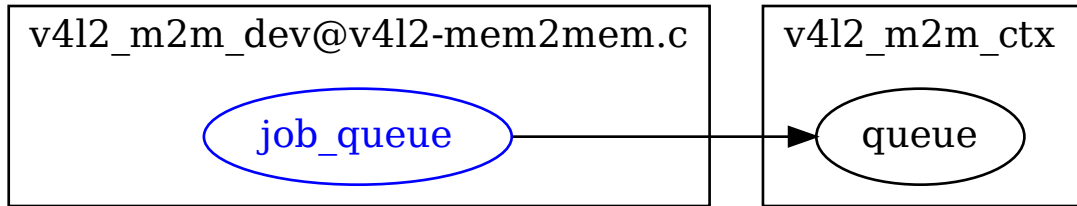
matrox_fb_info

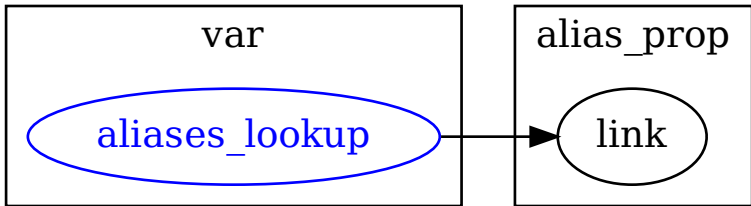
next_fb

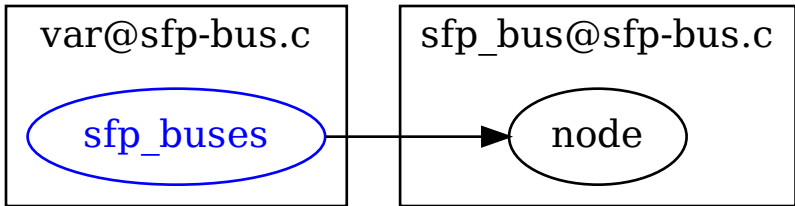
```
graph LR; A[matroxfb_list] --> B[next_fb];
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@matroxfb_base.c' at the top and 'matroxfb_list' in the center. The 'matroxfb_list' text is enclosed in a blue oval. On the right, another rectangular box contains the text 'matrox_fb_info' at the top and 'next_fb' in the center. The 'next_fb' text is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval to the left side of the black oval, indicating that 'matroxfb_list' points to 'next_fb'.









var@opal-prd.c

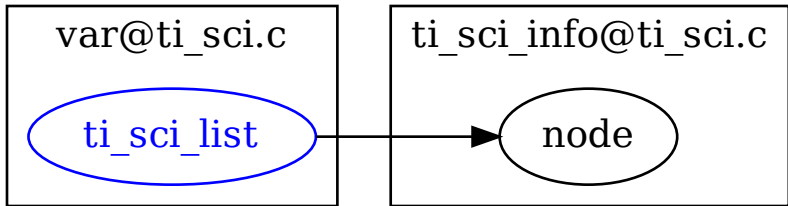
opal_prd_msg_queue

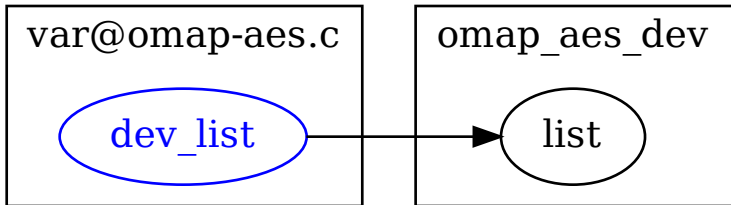
opal_prd_msg_queue_item@opal-prd.c

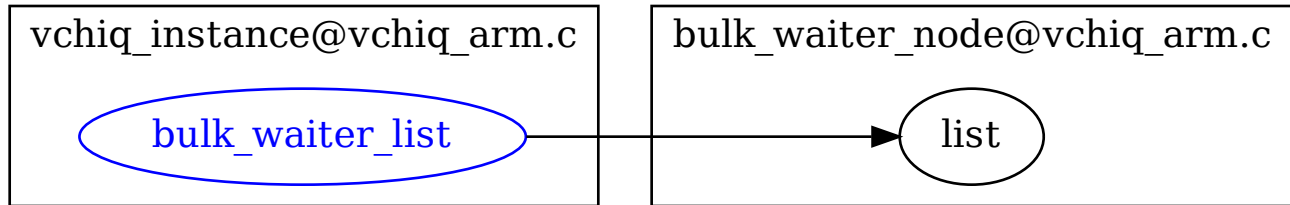
list

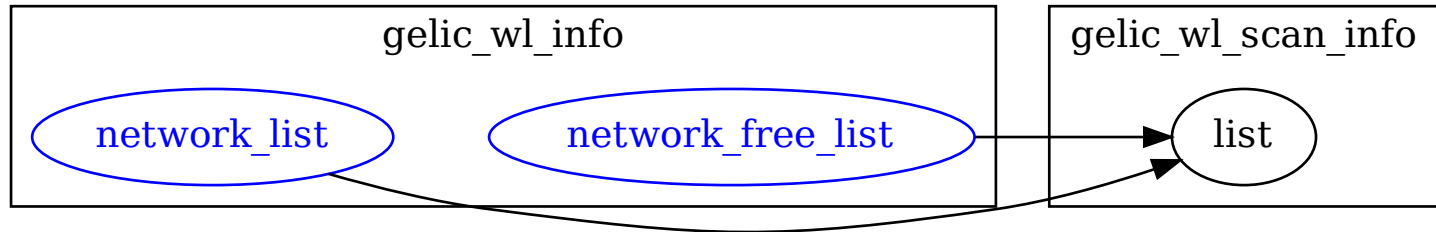


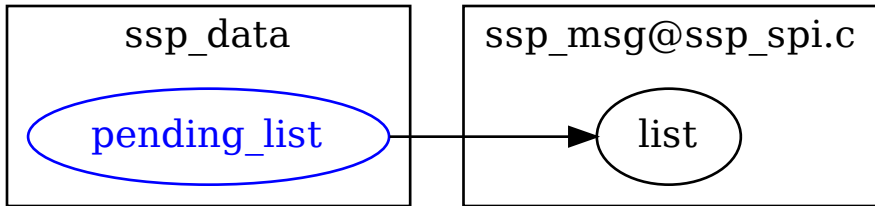
```
graph LR; subgraph Module1 [var@opal-prd.c]; opal_prd_msg_queue; end; subgraph Module2 [opal_prd_msg_queue_item@opal-prd.c]; list; end; opal_prd_msg_queue --> list;
```

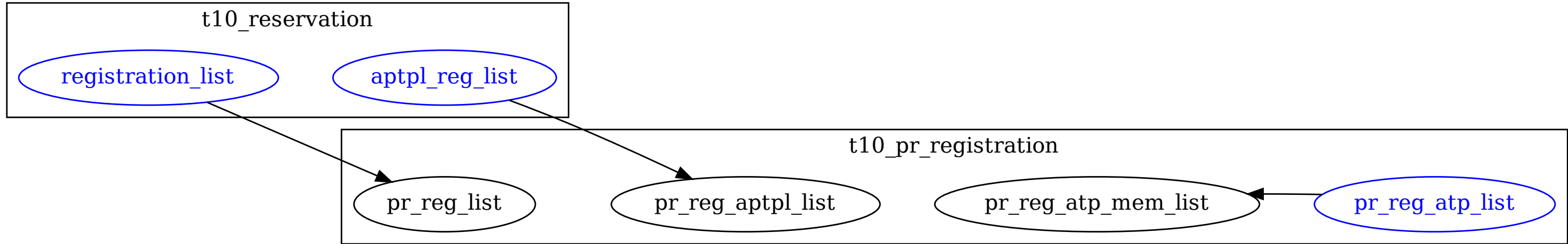












ieee802154_local

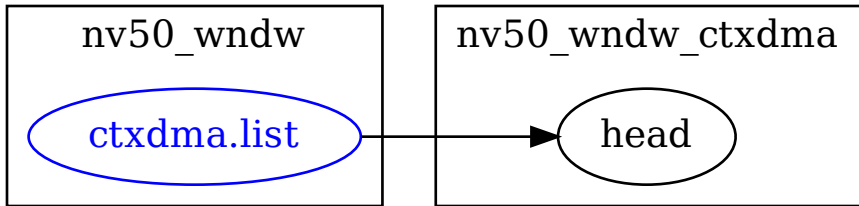
interfaces

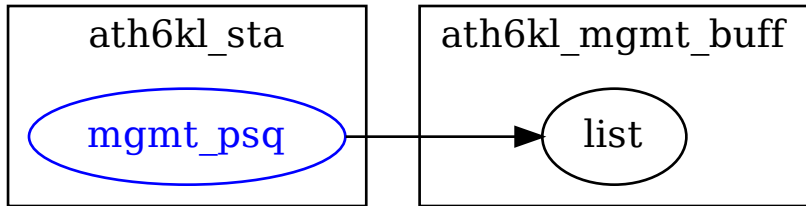
ieee802154_sub_if_data

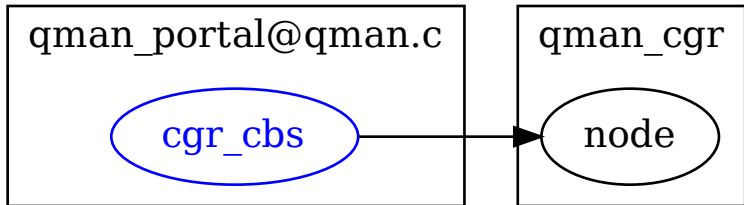
list

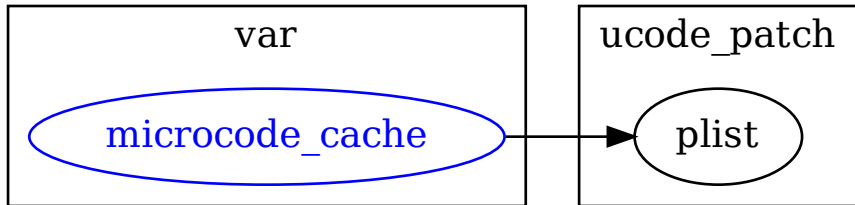
```
graph LR; subgraph ieee802154_local; interfaces; end; subgraph ieee802154_sub_if_data; list; end; interfaces --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'ieee802154_local' contains an oval labeled 'interfaces'. On the right, a rectangular box labeled 'ieee802154_sub_if_data' contains an oval labeled 'list'. A horizontal arrow points from the 'interfaces' oval to the 'list' oval, indicating that the 'interfaces' variable points to the 'list' structure.









var@vgaarb.c

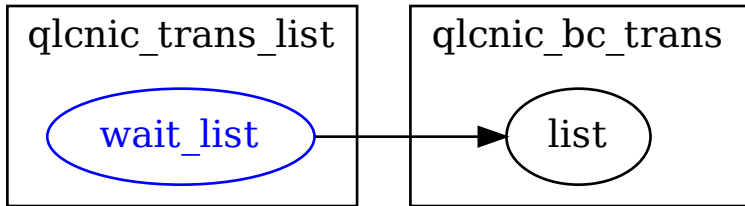
vga_user_list

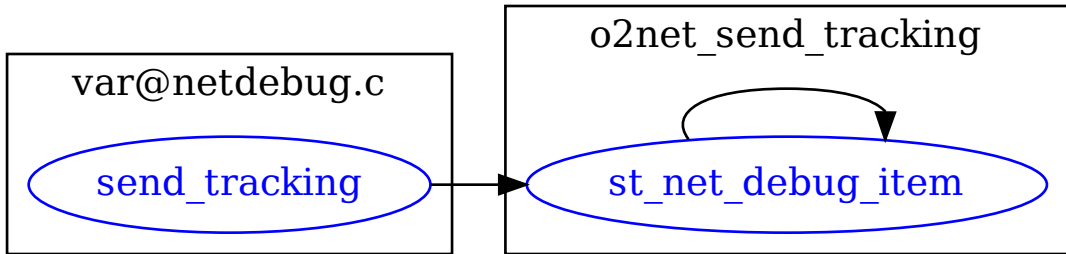
vga_arb_private@vgaarb.c

list

```
graph LR; subgraph Box1 [var@vgaarb.c]; vga_user_list; end; subgraph Box2 [vga_arb_private@vgaarb.c]; list; end; vga_user_list --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@vgaarb.c' and contains an oval labeled 'vga_user_list' with a blue border. The right box is labeled 'vga_arb_private@vgaarb.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'vga_user_list' oval to the left side of the 'list' oval.





var@omap-iommu.c

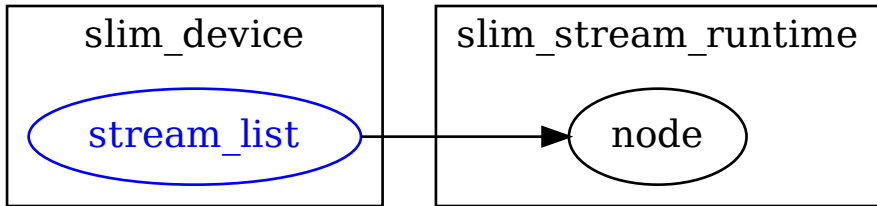
orphan_dev_list

orphan_dev@omap-iommu.c

node

```
graph LR; A[orphan_dev_list] --> B((node));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@omap-iommu.c' contains an oval labeled 'orphan_dev_list' with a blue border. On the right, a rectangular box labeled 'orphan_dev@omap-iommu.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'orphan_dev_list' oval to the left side of the 'node' oval, indicating that 'orphan_dev_list' points to 'node'.



var@nf_tables_api.c

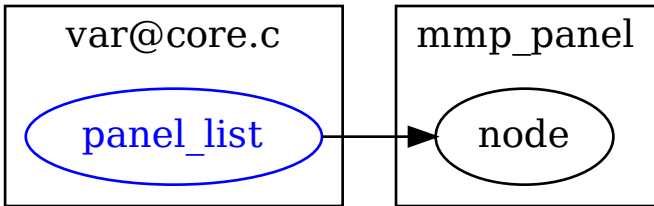
nf_tables_flowtables

nf_flowtable_type

list

```
graph LR; A[nf_tables_flowtables] --> B((list));
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@nf_tables_api.c' at the top and 'nf_tables_flowtables' in the center, which is enclosed in a blue oval. The right box contains the text 'nf_flowtable_type' at the top and 'list' in the center, which is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.



var@intel-iommu.c

dmar_rmrr_units

dmar_rmrr_unit@intel-iommu.c

list

```
graph LR; subgraph Box1 [var@intel-iommu.c]; dmar_rmrr_units; end; subgraph Box2 [dmar_rmrr_unit@intel-iommu.c]; list; end; dmar_rmrr_units --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@intel-iommu.c' contains a variable 'dmar_rmrr_units' which is highlighted with a blue oval. On the right, a box labeled 'dmar_rmrr_unit@intel-iommu.c' contains a variable 'list' which is enclosed in a black oval. A horizontal arrow points from the 'dmar_rmrr_units' oval to the 'list' oval, indicating that 'dmar_rmrr_units' points to 'list'.

wf_sat@windfarm_smu_sat.c

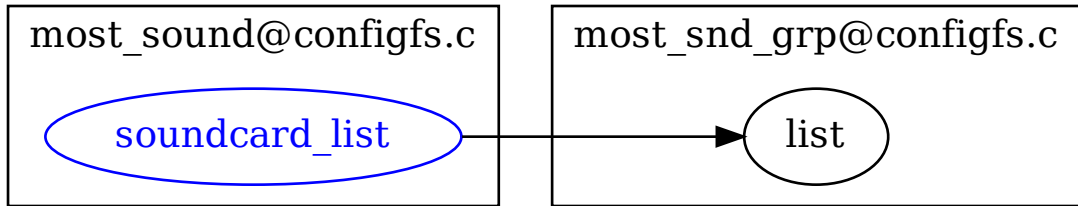
sensors

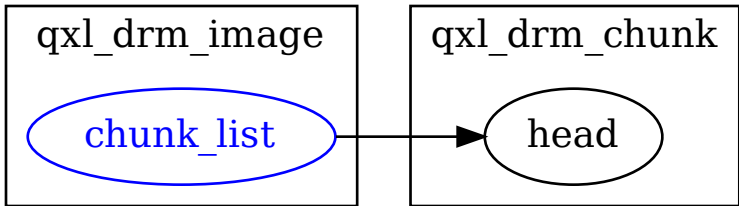
wf_sat_sensor@windfarm_smu_sat.c

link



```
graph LR; subgraph Module1 [wf_sat@windfarm_smu_sat.c]; sensors((sensors)); end; subgraph Module2 [wf_sat_sensor@windfarm_smu_sat.c]; link((link)); end; sensors --> link;
```





var@ima_template.c

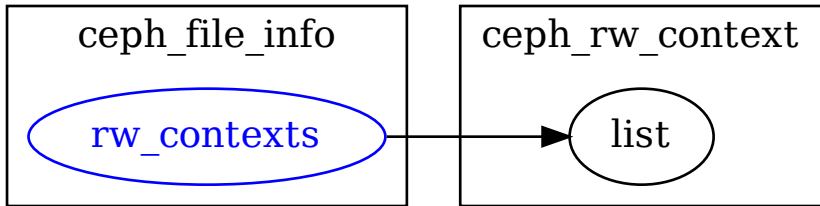
defined_templates

ima_template_desc

list

```
graph LR; subgraph Box1 [var@ima_template.c]; defined_templates; end; subgraph Box2 [ima_template_desc]; list; end; defined_templates --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@ima_template.c' and contains an oval labeled 'defined_templates' in blue text. The right box is labeled 'ima_template_desc' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'defined_templates' oval to the left side of the 'list' oval.



var@ab8500_fg.c

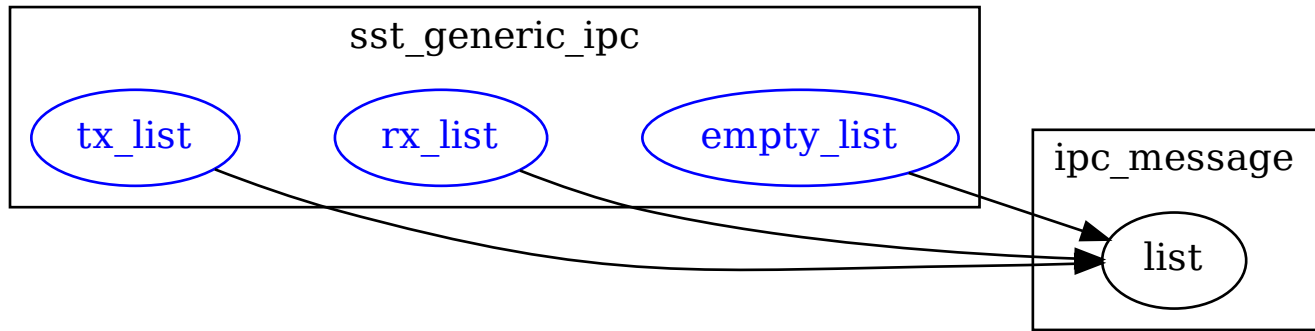
ab8500_fg_list

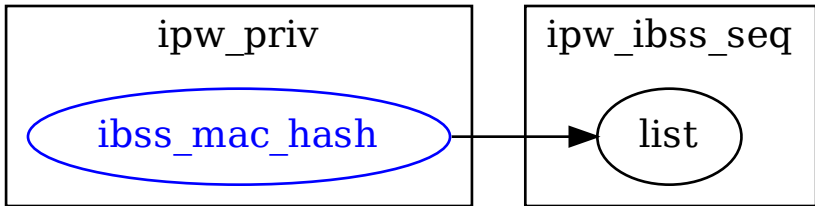
ab8500_fg@ab8500_fg.c

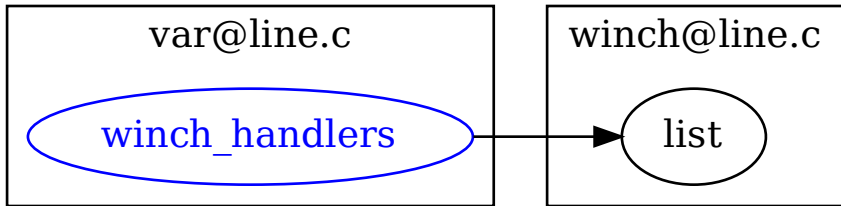
node

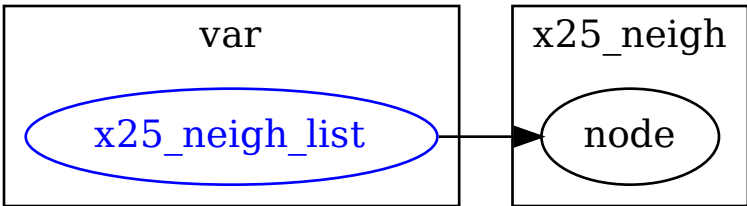
```
graph LR; subgraph Box1 [var@ab8500_fg.c]; direction TB; A([ab8500_fg_list]); end; subgraph Box2 [ab8500_fg@ab8500_fg.c]; direction TB; B([node]); end; A --> B;
```

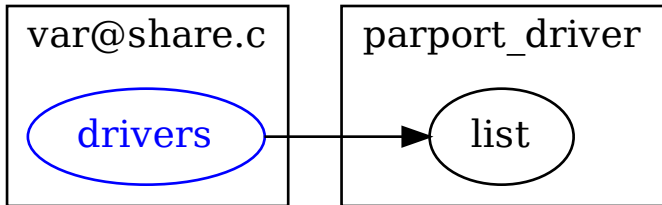
The diagram consists of two rectangular boxes. The left box is labeled 'var@ab8500_fg.c' and contains a blue oval labeled 'ab8500_fg_list'. The right box is labeled 'ab8500_fg@ab8500_fg.c' and contains a black oval labeled 'node'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box.

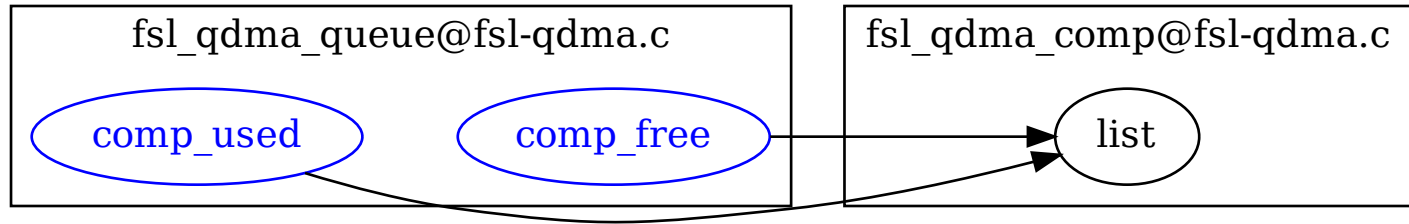


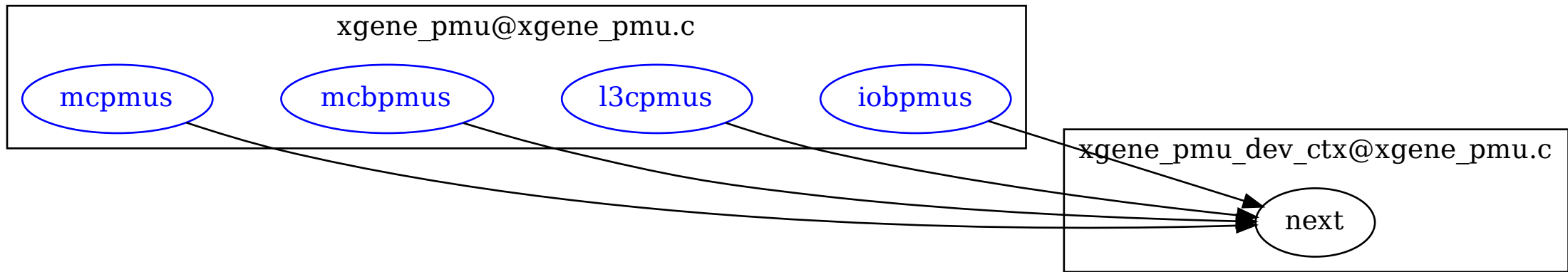


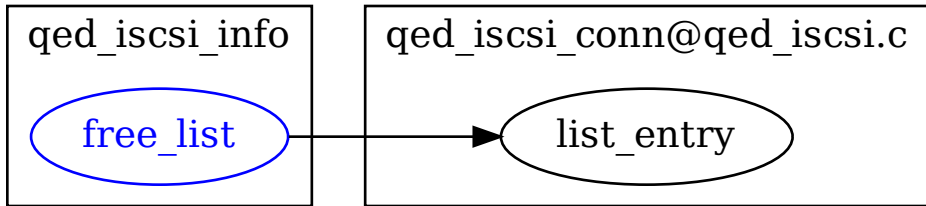


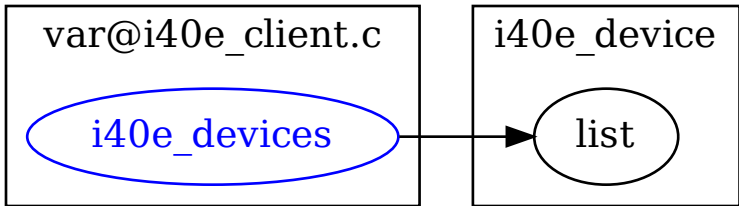


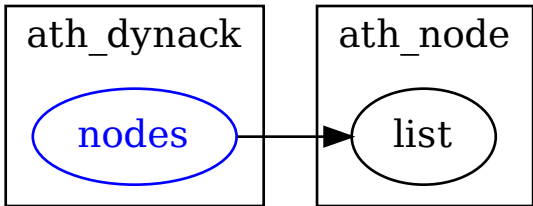


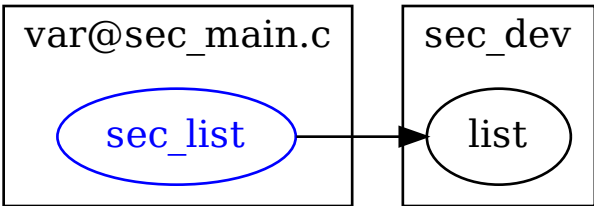


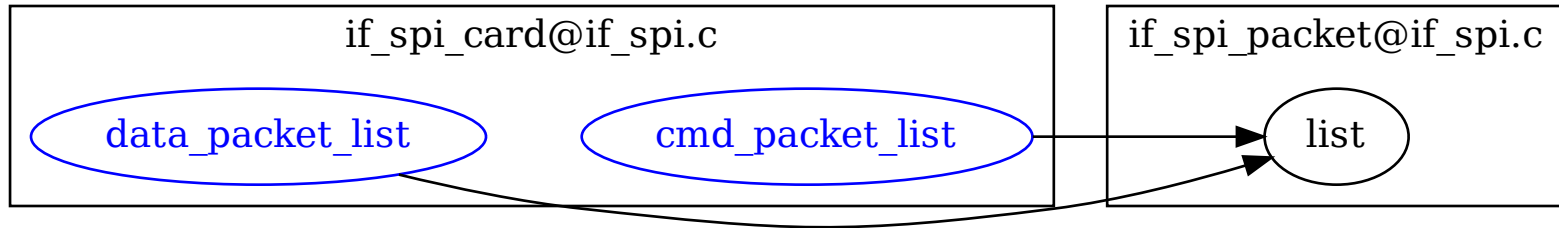


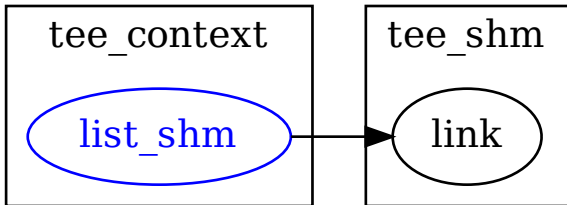


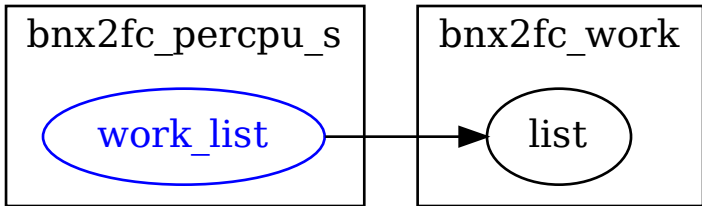


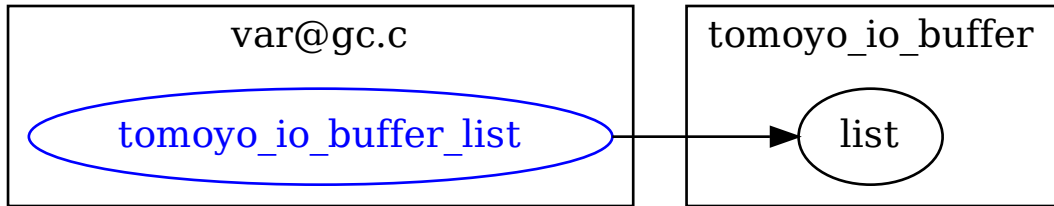












dmabounce_device_info@dmabounce.c

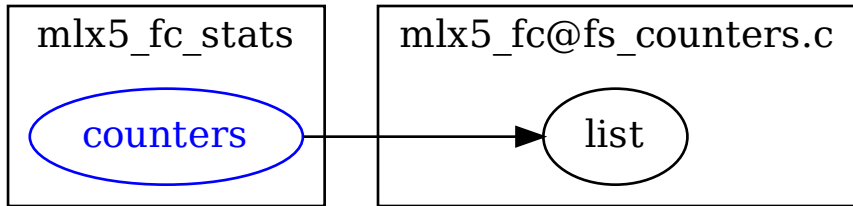
safe_buffers

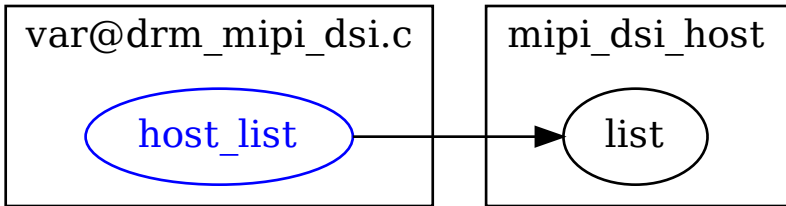
safe_buffer@dmabounce.c

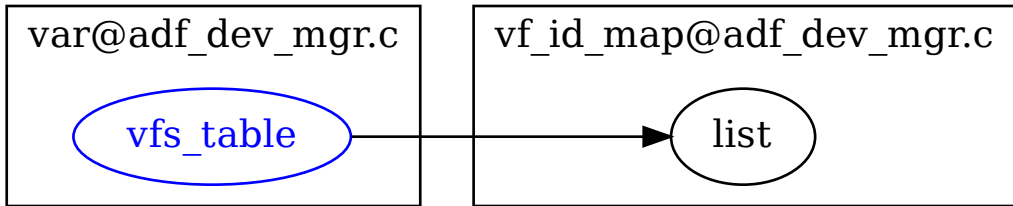
node

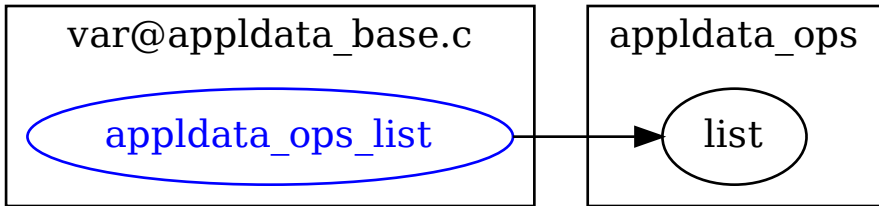
```
graph LR; subgraph Module1 [dmabounce_device_info@dmabounce.c]; safe_buffers; end; subgraph Module2 [safe_buffer@dmabounce.c]; node; end; safe_buffers --> node;
```

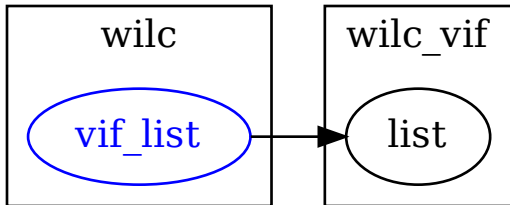
The diagram illustrates a pointer relationship between two modules. On the left, a box represents the module 'dmabounce_device_info@dmabounce.c', which contains a blue oval labeled 'safe_buffers'. On the right, a box represents the module 'safe_buffer@dmabounce.c', which contains a black oval labeled 'node'. A horizontal arrow points from the 'safe_buffers' oval to the 'node' oval, indicating that 'safe_buffers' points to 'node'.

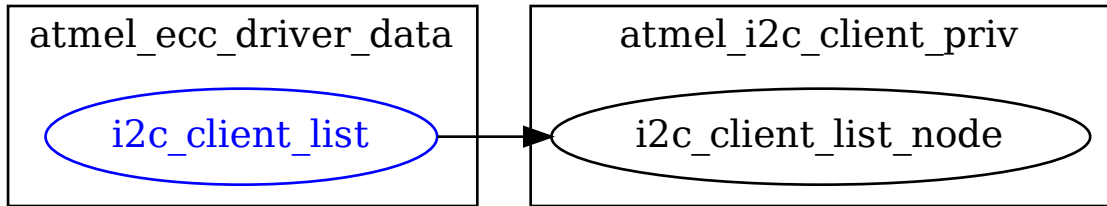












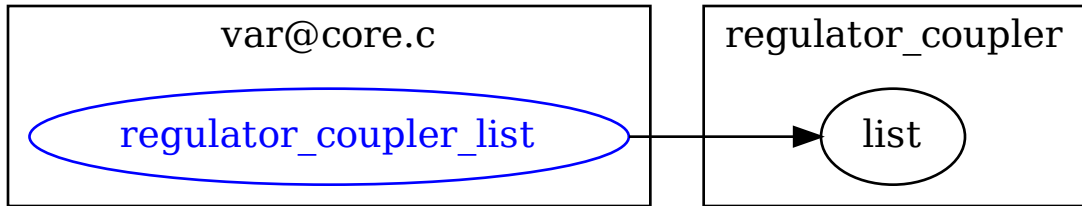
pfect_lru@test_lru_dist.c

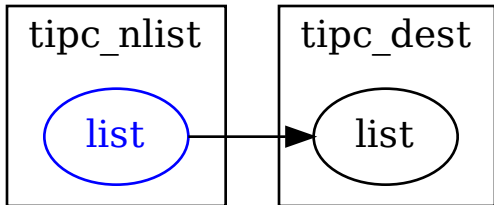
list

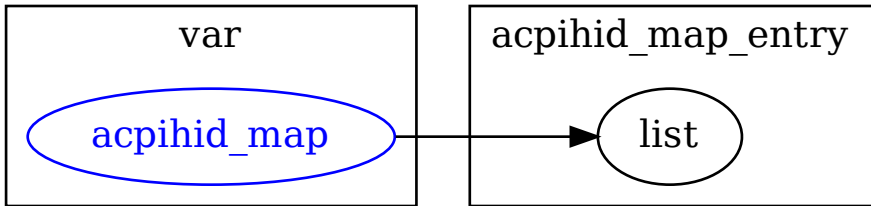
pfect_lru_node@test_lru_dist.c

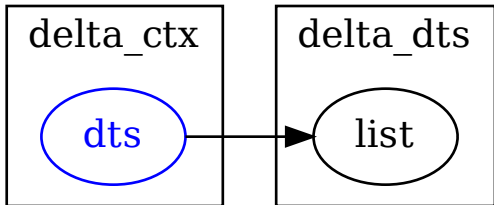
list

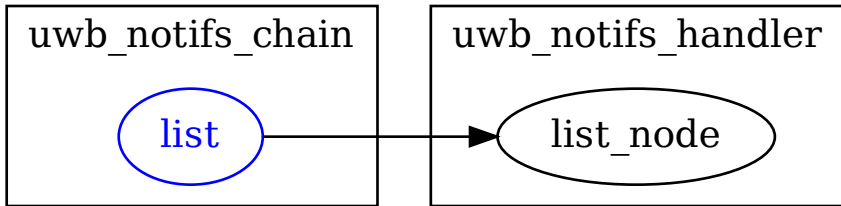


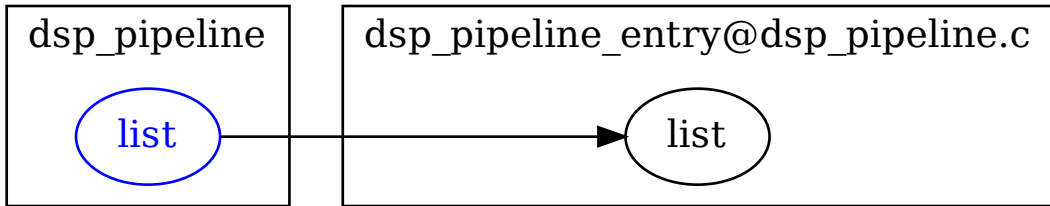


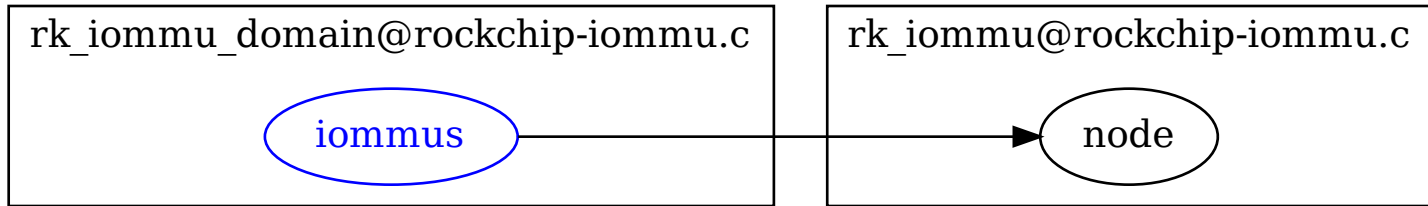


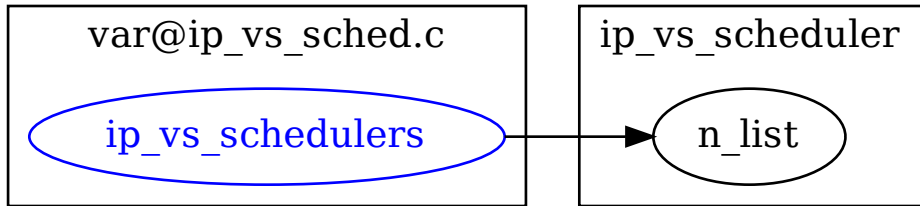












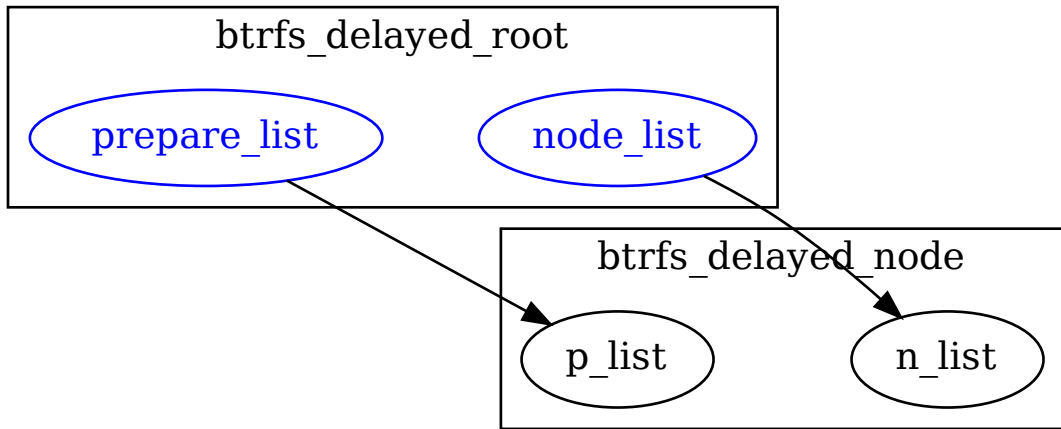
rcar_thermal_common@rcar_thermal.c

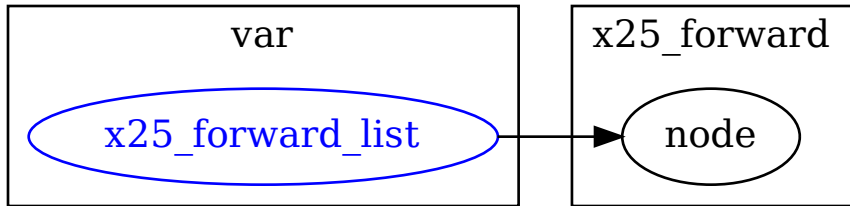
head

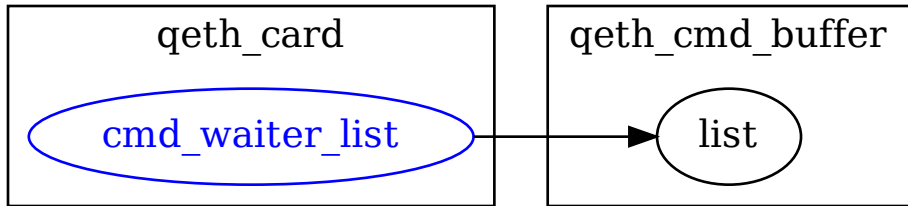
rcar_thermal_priv@rcar_thermal.c

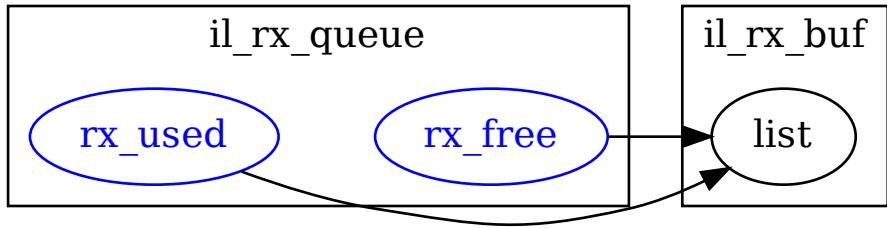
list

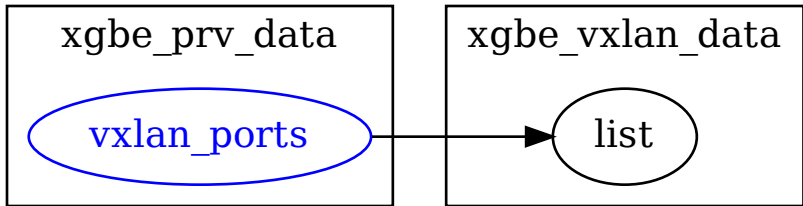


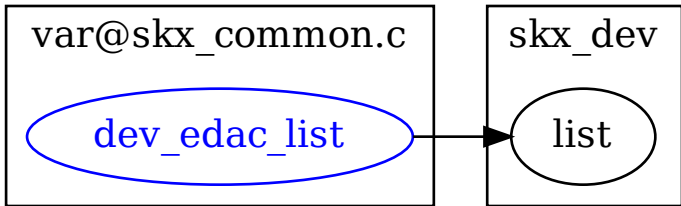












var@w83793.c

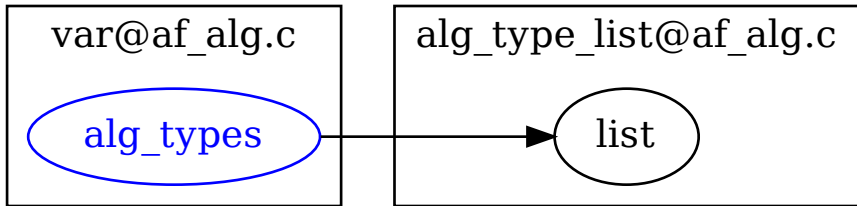
watchdog_data_list

w83793_data@w83793.c

list

```
graph LR; A[watchdog_data_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@w83793.c' contains an oval labeled 'watchdog_data_list' with a blue border. On the right, a rectangular box labeled 'w83793_data@w83793.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'watchdog_data_list' oval to the left side of the 'list' oval, indicating that 'watchdog_data_list' points to 'list'.



acpi_device_wakeup

resources

acpi_power_resource_entry@power.c

node

```
graph LR; subgraph acpi_device_wakeup; resources; end; subgraph acpi_power_resource_entrypower_c [acpi_power_resource_entry@power.c]; node; end; resources --> node; node --> node;
```

The diagram consists of two rectangular boxes. The left box is titled 'acpi_device_wakeup' and contains an oval labeled 'resources'. The right box is titled 'acpi_power_resource_entry@power.c' and contains an oval labeled 'node'. A horizontal arrow points from the 'resources' oval to the 'node' oval. Additionally, a curved arrow originates from the top of the 'node' oval and points back to its top, forming a self-loop.

var@asymmetric_type.c

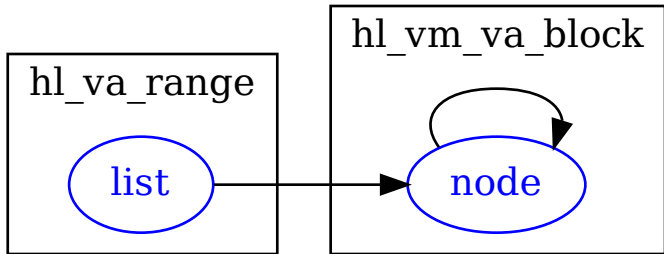
asymmetric_key_parsers

asymmetric_key_parser

link

```
graph LR; subgraph Box1 [var@asymmetric_type.c]; A(asymmetric_key_parsers); end; subgraph Box2 [asymmetric_key_parser]; B((link)); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@asymmetric_type.c' and contains a blue oval labeled 'asymmetric_key_parsers'. The right box is labeled 'asymmetric_key_parser' and contains a black oval labeled 'link'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box.



var@control.c

snd_control_ioctls

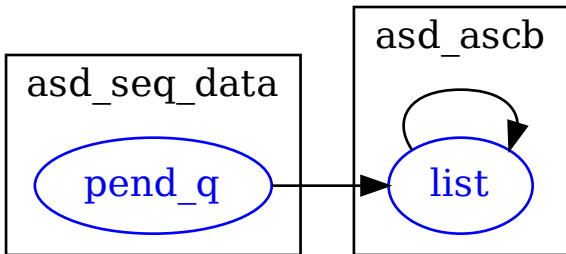
snd_kctl_ioctl@control.c

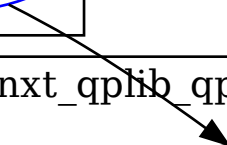
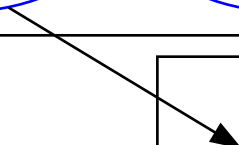
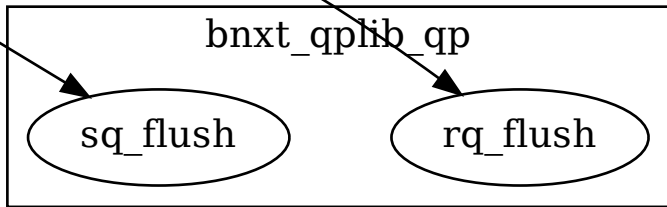
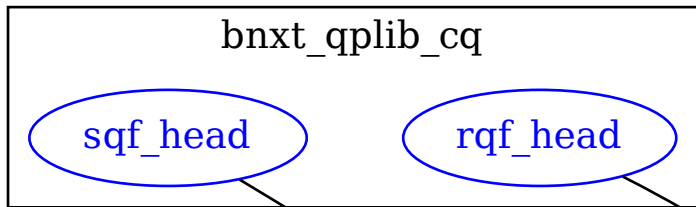
list

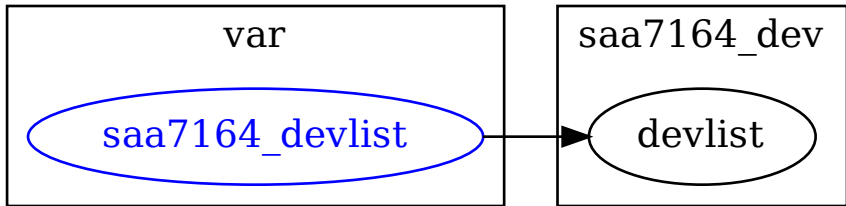
```
graph LR; subgraph Box1 [var@control.c]; direction TB; A(snd_control_ioctls); end; subgraph Box2 [snd_kctl_ioctl@control.c]; direction TB; B((list)); end; A --> B;
```

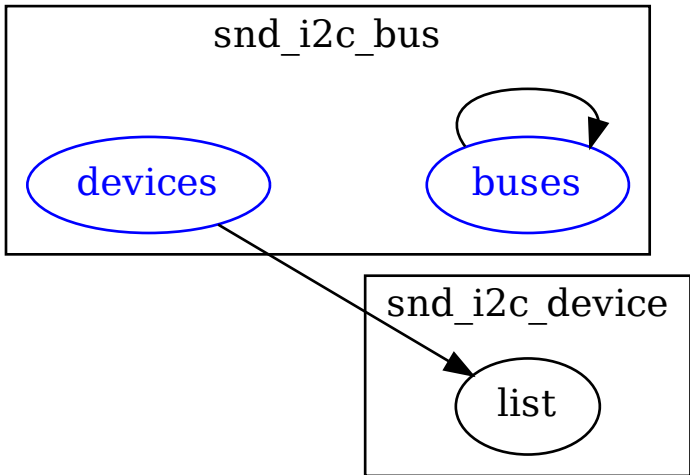
The diagram consists of two rectangular boxes. The left box is labeled 'var@control.c' and contains a blue oval labeled 'snd_control_ioctls'. The right box is labeled 'snd_kctl_ioctl@control.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'snd_control_ioctls' oval to the left side of the 'list' oval.











ttm_pool_manager@ttm_page_alloc_dma.c

pools

device_pools@ttm_page_alloc_dma.c

pools



khugepaged_scan@khugepaged.c

mm_head

mm_slot@khugepaged.c

mm_node



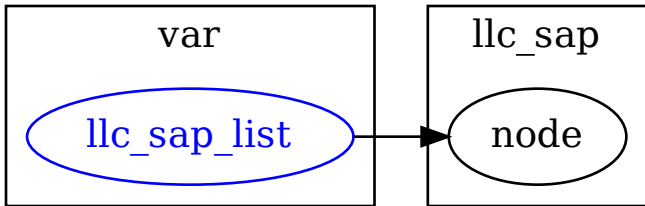
dfs_pattern_detector

channel_detectors

channel_detector@dfs_pattern_detector.c

head

The diagram consists of two rectangular boxes. The left box is labeled 'dfs_pattern_detector' and contains a blue oval labeled 'channel_detectors'. The right box is labeled 'channel_detector@dfs_pattern_detector.c' and contains a black oval labeled 'head'. A horizontal arrow points from the 'channel_detectors' oval in the left box to the 'head' oval in the right box.



var@cache.c

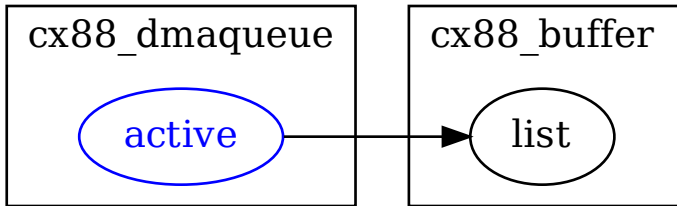
fscache_cache_tag_list

fscache_cache_tag

link

```
graph LR; A[fscache_cache_tag_list] --> B((link));
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@cache.c' and contains an oval labeled 'fscache_cache_tag_list' in blue text. The right box is labeled 'fscache_cache_tag' and contains an oval labeled 'link'. A horizontal arrow points from the right side of the 'fscache_cache_tag_list' oval to the left side of the 'link' oval.



tegra_smmu@tegra-smmu.c

groups

tegra_smmu_group@tegra-smmu.c

list

```
graph LR; subgraph Box1 [tegra_smmu@tegra-smmu.c]; groups([groups]); end; subgraph Box2 [tegra_smmu_group@tegra-smmu.c]; list([list]); end; groups --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'tegra_smmu@tegra-smmu.c' and contains an oval labeled 'groups' in blue text. The right box is labeled 'tegra_smmu_group@tegra-smmu.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the right side of the 'groups' oval to the left side of the 'list' oval, indicating a pointer relationship.

var@fcoe_transport.c

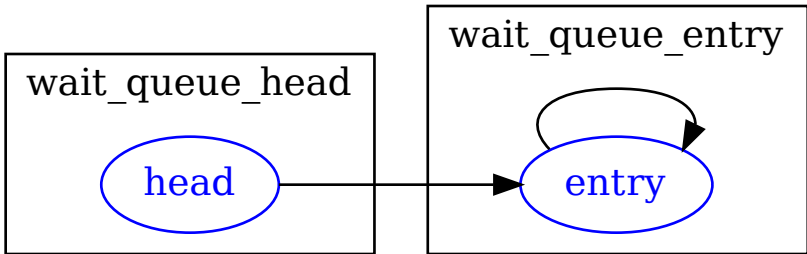
fcoe_netdevs

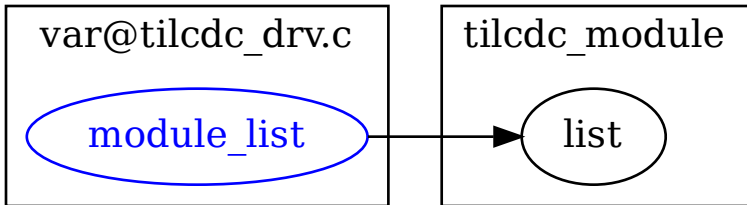
fcoe_netdev_mapping

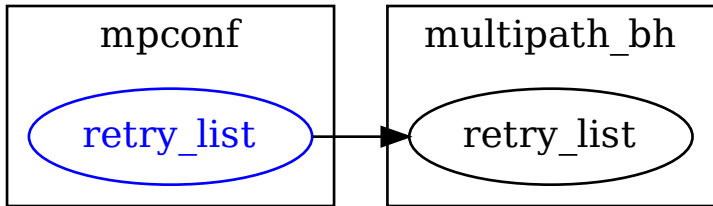
list

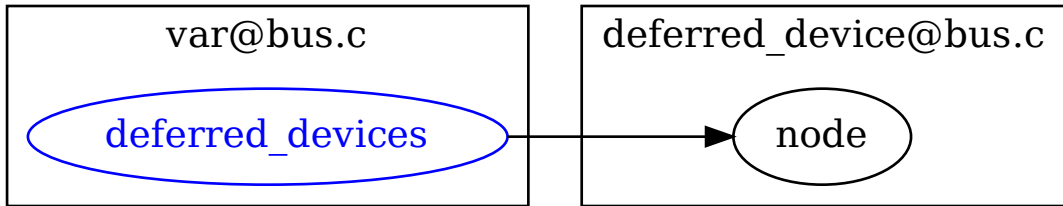
```
graph LR; A["var@fcoe_transport.c  
fcoe_netdevs"] --> B["fcoe_netdev_mapping  
list"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@fcoe_transport.c' at the top and 'fcoe_netdevs' in a blue oval below it. The right box contains the text 'fcoe_netdev_mapping' at the top and 'list' in a black oval below it. A horizontal arrow points from the right side of the 'fcoe_netdevs' oval to the left side of the 'list' oval.









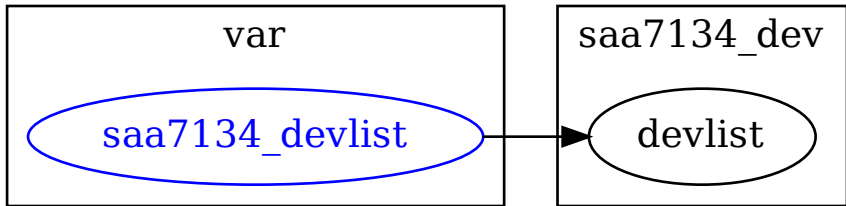
var@gss_mech_switch.c

registered_mechs

gss_api_mech

gm_list





var@trace_dynevent.c

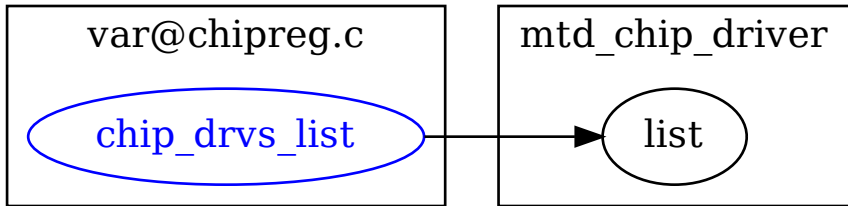
dyn_event_ops_list

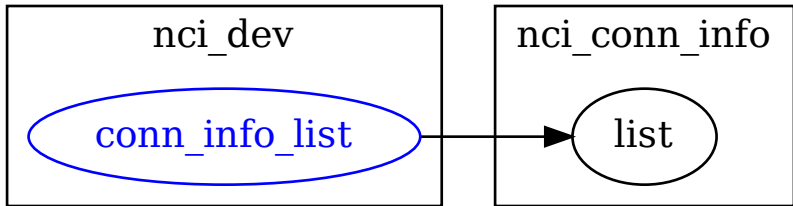
dyn_event_operations

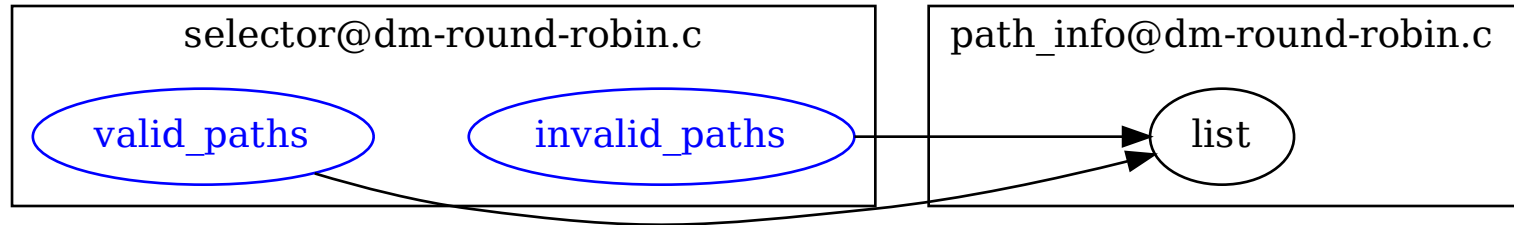
list

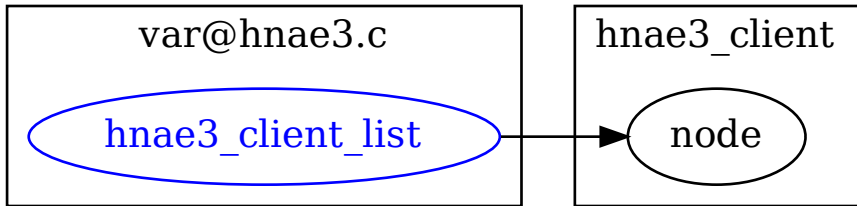
```
graph LR; A[var@trace_dynevent.c] --> B((list)); subgraph A_Box [var@trace_dynevent.c]; A; end; subgraph B_Box [dyn_event_operations]; B; end;
```

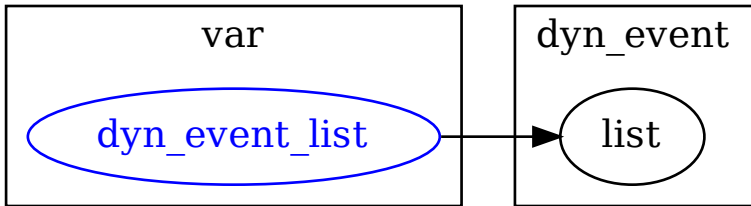
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@trace_dynevent.c' contains an oval labeled 'dyn_event_ops_list' in blue text. On the right, a rectangular box labeled 'dyn_event_operations' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'dyn_event_ops_list' oval to the left side of the 'list' oval, indicating that the variable 'dyn_event_ops_list' points to the 'list' structure within the 'dyn_event_operations' context.











var@core-transaction.c

address_handler_list

fw_address_handler

link

```
graph LR; A[address_handler_list] --> B((link));
```

The diagram illustrates a pointer relationship between two modules. On the left, a rectangular box labeled 'var@core-transaction.c' contains an oval labeled 'address_handler_list' with a blue border. On the right, a rectangular box labeled 'fw_address_handler' contains an oval labeled 'link'. A horizontal arrow points from the right side of the 'address_handler_list' oval to the left side of the 'link' oval, indicating that 'address_handler_list' points to 'link'.

var@ima_queue_keys.c

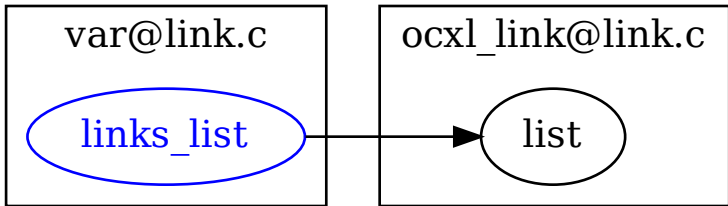
ima_keys

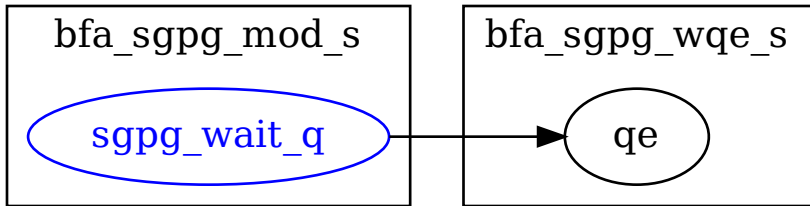
ima_key_entry

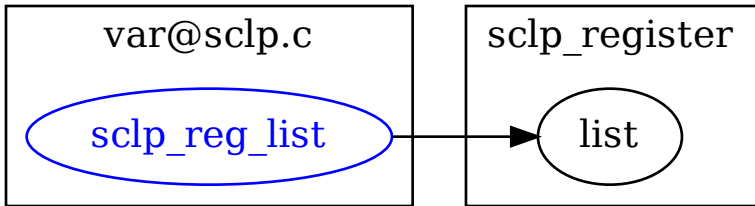
list

```
graph LR; A[ima_keys] --> B(list);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'var@ima_queue_keys.c' contains an oval labeled 'ima_keys' with a blue border. On the right, a rectangular box labeled 'ima_key_entry' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'ima_keys' oval to the left side of the 'list' oval, indicating that 'ima_keys' points to 'list'.







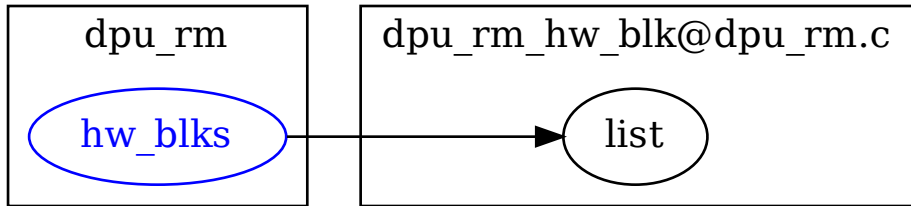
max3421_hcd@max3421-hcd.c

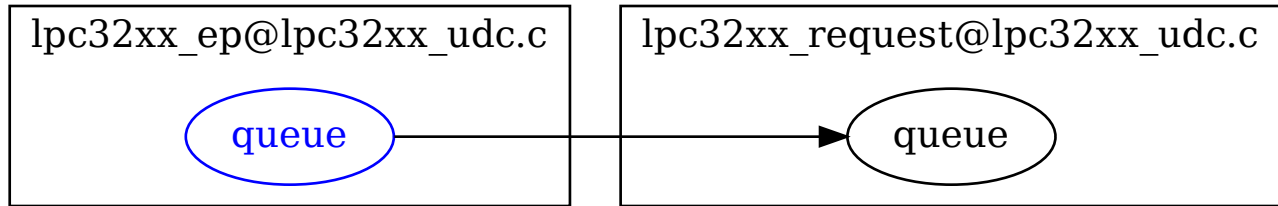
ep_list

max3421_ep@max3421-hcd.c

ep_list







var@bnx2fc_fcoe.c

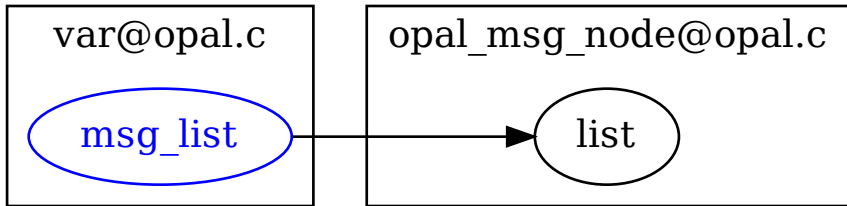
if_list

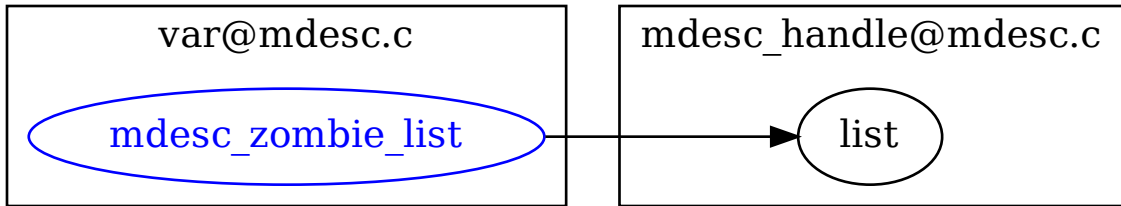
bnx2fc_interface

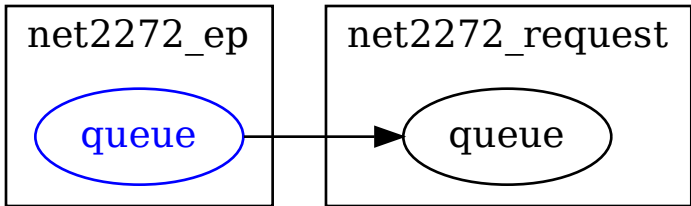
list

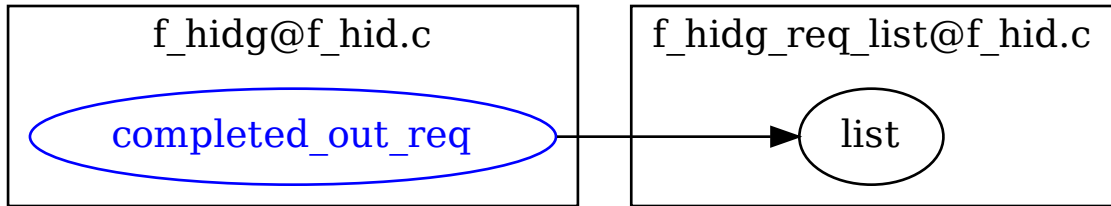
```
graph LR; subgraph var_at_bnx2fc_fcoe_c [var@bnx2fc_fcoe.c]; if_list([if_list]); end; subgraph bnx2fc_interface [bnx2fc_interface]; list([list]); end; if_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@bnx2fc_fcoe.c' contains an oval labeled 'if_list' with a blue border. On the right, a box labeled 'bnx2fc_interface' contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'if_list' oval to the 'list' oval, indicating that 'if_list' points to 'list'.









amd76xrom_window@amd76xrom.c

maps

amd76xrom_map_info@amd76xrom.c

list

```
graph LR; maps([maps]) --> list([list]);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'amd76xrom_window@amd76xrom.c' contains an oval labeled 'maps'. On the right, a box labeled 'amd76xrom_map_info@amd76xrom.c' contains an oval labeled 'list'. A horizontal arrow points from the 'maps' oval to the 'list' oval, indicating that 'maps' points to 'list'.

var@pvrdma_main.c

pvrdma_device_list

pvrdma_dev

device_link



s390_domain@s390-iommu.c

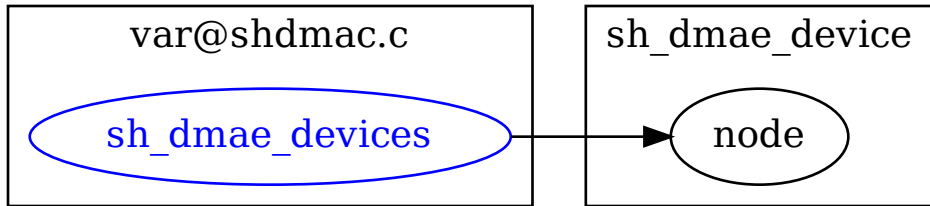
devices

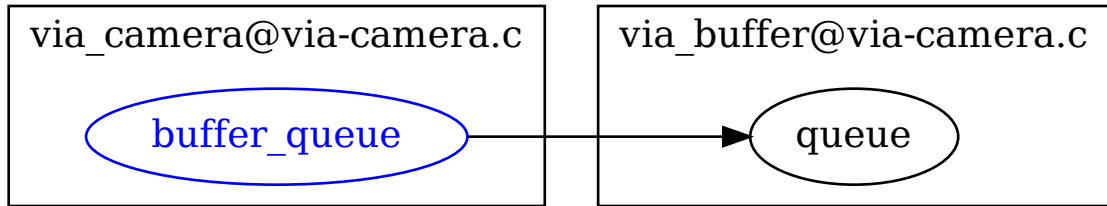
s390_domain_device@s390-iommu.c

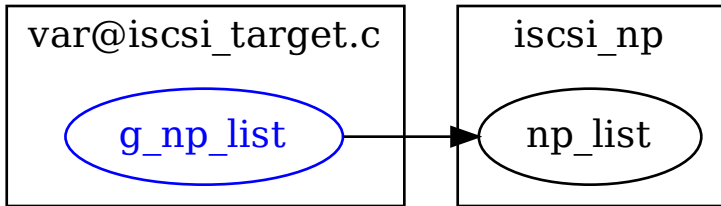
list

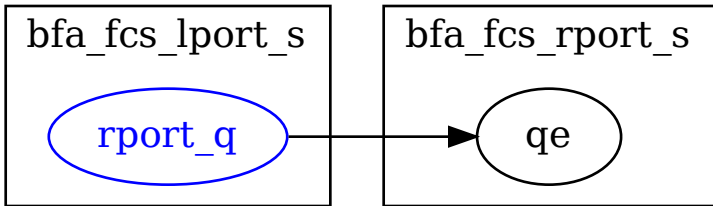
```
graph LR; subgraph Module1 [s390_domain@s390-iommu.c]; direction TB; D([devices]); end; subgraph Module2 [s390_domain_device@s390-iommu.c]; direction TB; L([list]); end; D --> L;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 's390_domain@s390-iommu.c' contains an oval labeled 'devices'. On the right, a box labeled 's390_domain_device@s390-iommu.c' contains an oval labeled 'list'. A horizontal arrow points from the 'devices' oval to the 'list' oval, indicating that the 'devices' variable in the first module points to the 'list' variable in the second module.









netup_dma

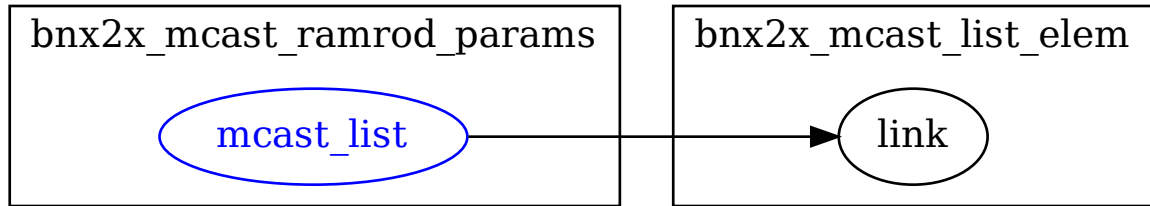
free_buffers

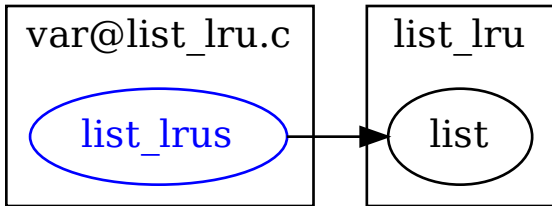
netup_unidvb_buffer@netup_unidvb_core.c

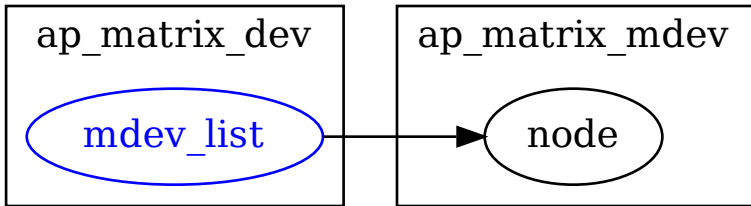
list

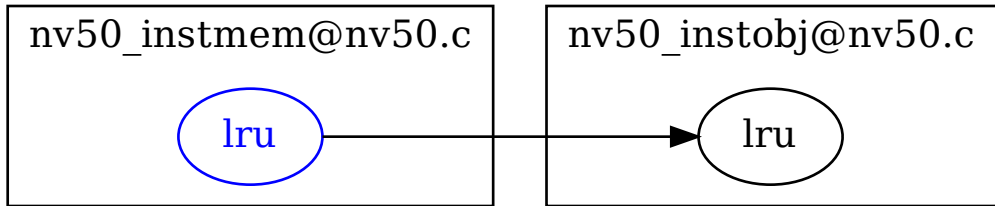
```
graph LR; subgraph netup_dma; free_buffers; end; subgraph netup_unidvb_buffer_at_netup_unidvb_core_c [netup_unidvb_buffer@netup_unidvb_core.c]; list; end; free_buffers --> list;
```

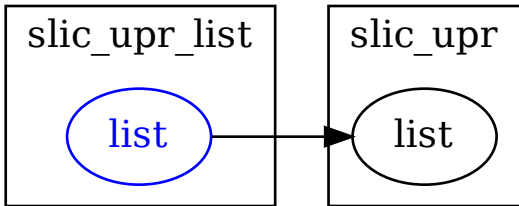
The diagram consists of two rectangular boxes. The left box is labeled 'netup_dma' and contains an oval labeled 'free_buffers'. The right box is labeled 'netup_unidvb_buffer@netup_unidvb_core.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'free_buffers' oval in the left box to the 'list' oval in the right box.

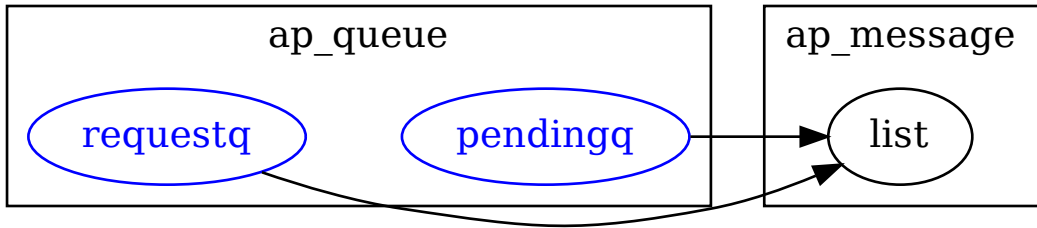












xfs_qm_isolate@xfs_qm.c

dispose

xfs_dquot

q_lru

```
graph LR; A[xfs_qm_isolate@xfs_qm.c] --> B((dispose)); B --> C[xfs_dquot]; C --> D((q_lru));
```

var@message.c

set_config_list

set_config_request@message.c

node



```
graph LR; A([set_config_list]) --> B([node]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@message.c' contains an oval labeled 'set_config_list' with a blue border. On the right, a rectangular box labeled 'set_config_request@message.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'set_config_list' oval to the left side of the 'node' oval, indicating that 'set_config_list' points to 'node'.

var@mconsole_kern.c

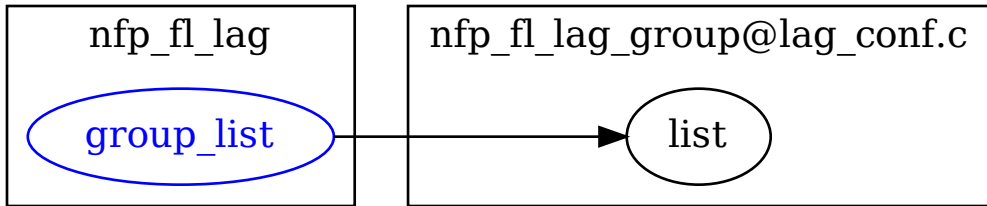
mc_requests

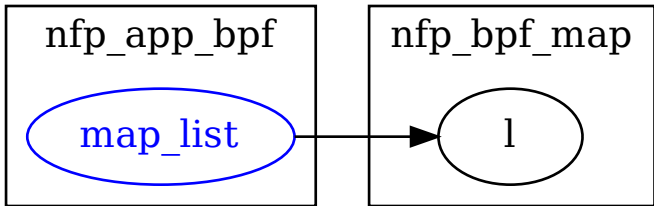
mconsole_entry

list

```
graph LR; subgraph Box1 [var@mconsole_kern.c]; mc_requests; end; subgraph Box2 [mconsole_entry]; list; end; mc_requests --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@mconsole_kern.c' and contains an oval labeled 'mc_requests' with a blue border. The right box is labeled 'mconsole_entry' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'mc_requests' oval to the left side of the 'list' oval.





var@tm6000-core.c

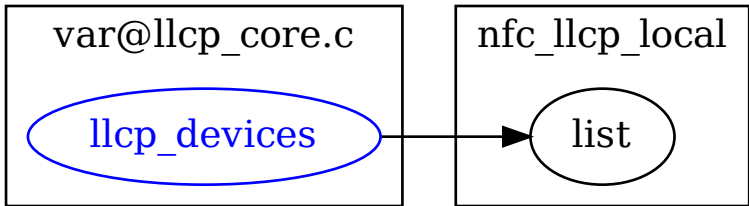
tm6000_extension_devlist

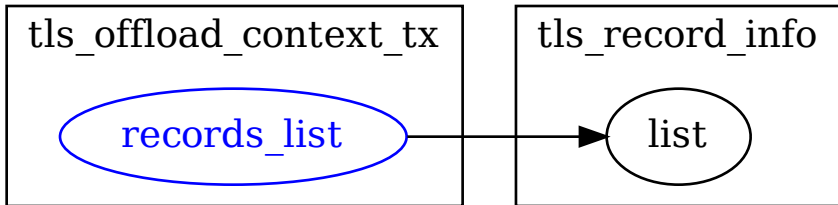
tm6000_ops

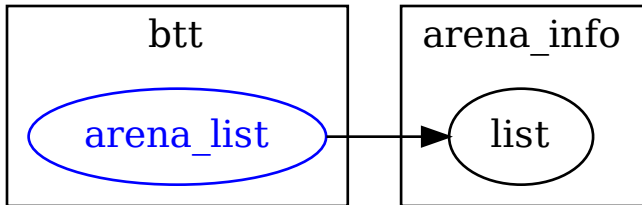
next

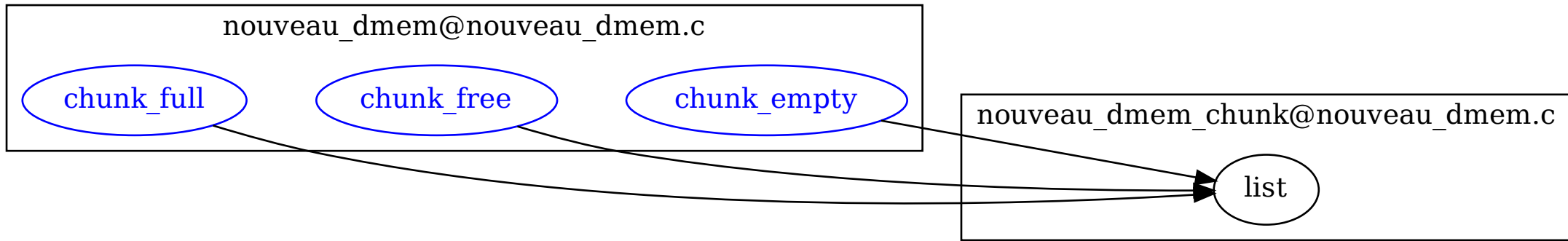
```
graph LR; A[tm6000_extension_devlist] --> B((next));
```

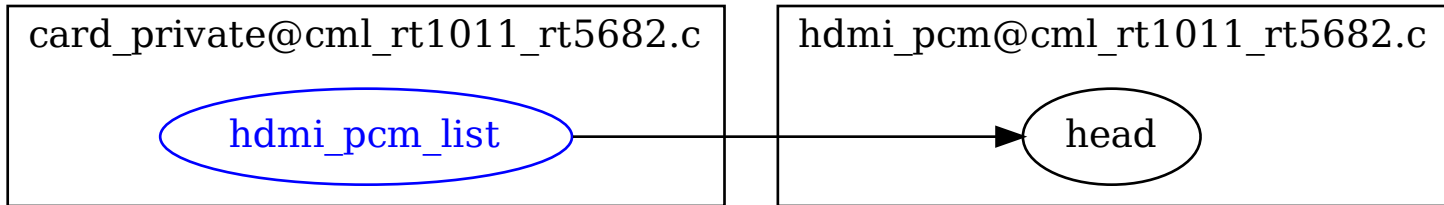
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'var@tm6000-core.c' contains an oval labeled 'tm6000_extension_devlist' with a blue border. On the right, a rectangular box labeled 'tm6000_ops' contains an oval labeled 'next' with a black border. A black arrow points from the right side of the 'tm6000_extension_devlist' oval to the left side of the 'next' oval, indicating that the 'tm6000_extension_devlist' variable points to the 'next' field within the 'tm6000_ops' structure.











mlxsw_sp_acl@spectrum_acl.c

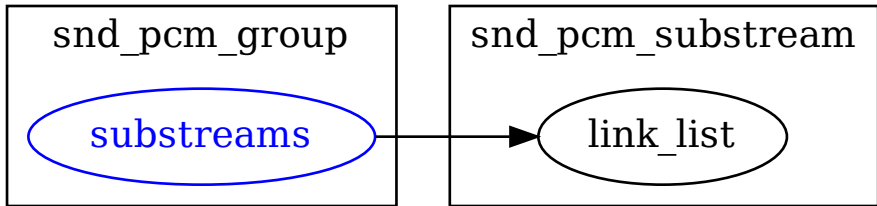
rules

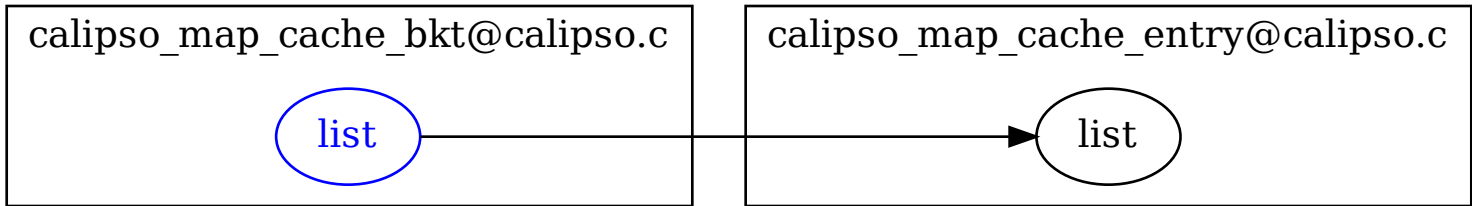
mlxsw_sp_acl_rule@spectrum_acl.c

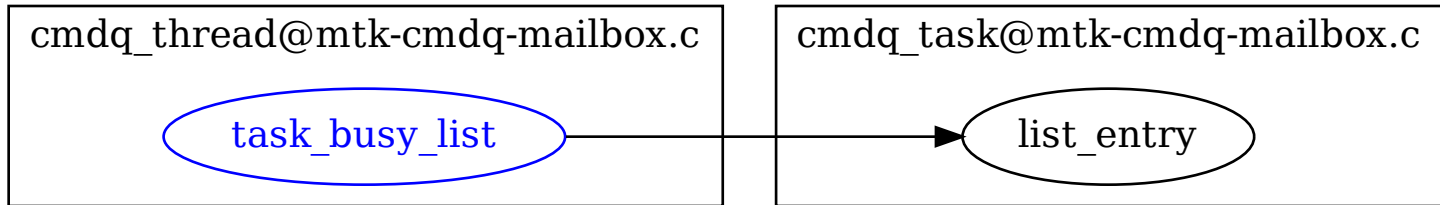
list

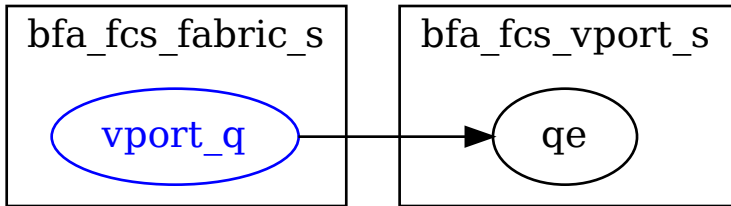
```
graph LR; rules(rules) --> list(list);
```

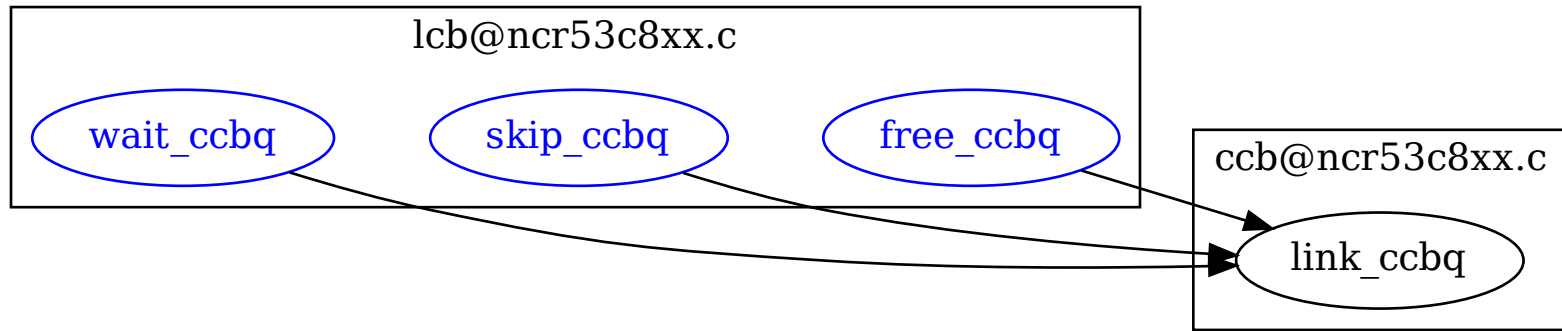
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'mlxsw_sp_acl@spectrum_acl.c' contains an oval labeled 'rules'. On the right, a box labeled 'mlxsw_sp_acl_rule@spectrum_acl.c' contains an oval labeled 'list'. A horizontal arrow points from the 'rules' oval to the 'list' oval, indicating that the 'rules' variable points to the 'list' variable.

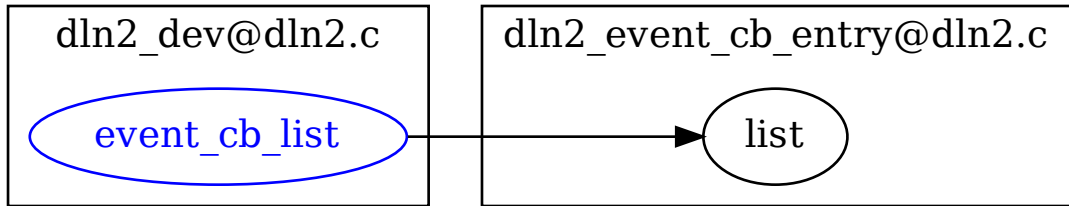


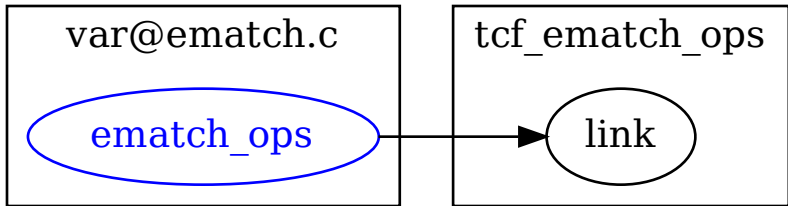


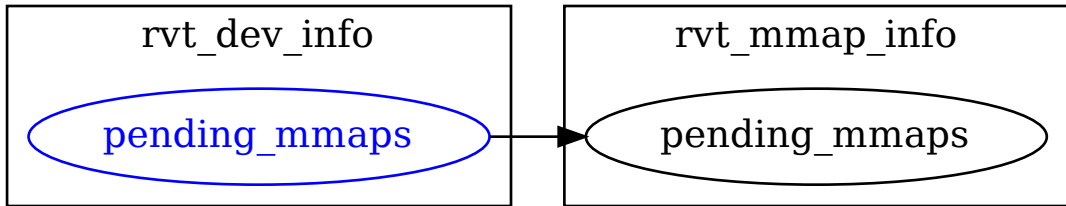


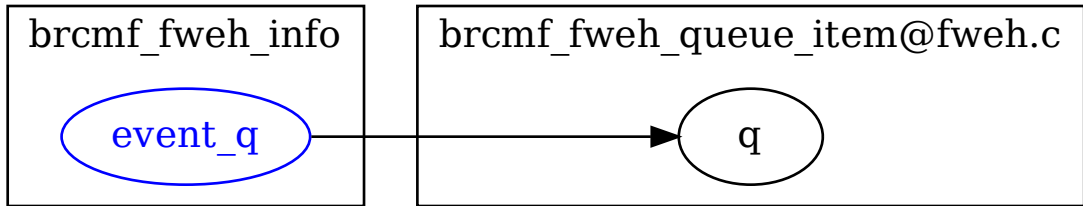












var@switch.c

nvm_auth_status_cache

nvm_auth_status@switch.c

list

```
graph LR; subgraph var_at_switch_c [var@switch.c]; nvm_auth_status_cache; end; subgraph nvm_auth_status_at_switch_c [nvm_auth_status@switch.c]; list; end; nvm_auth_status_cache --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@switch.c' contains an oval labeled 'nvm_auth_status_cache'. On the right, another rectangular box labeled 'nvm_auth_status@switch.c' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'nvm_auth_status_cache' oval to the left side of the 'list' oval, indicating that 'nvm_auth_status_cache' points to 'list'.

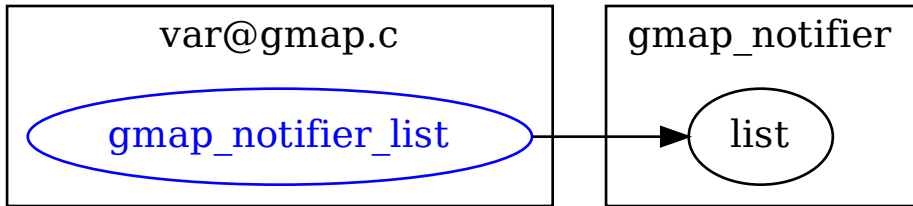
via_file_private

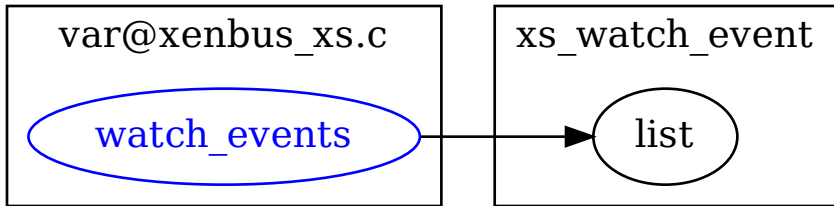
obj_list

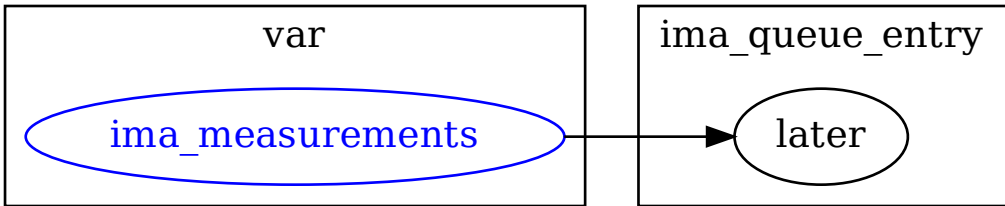
via_memblock@via_mm.c

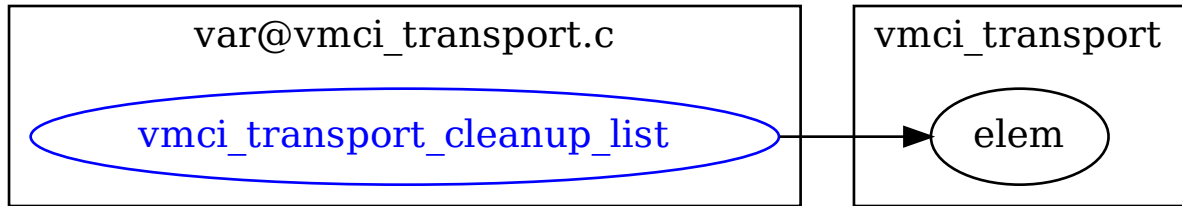
owner_list



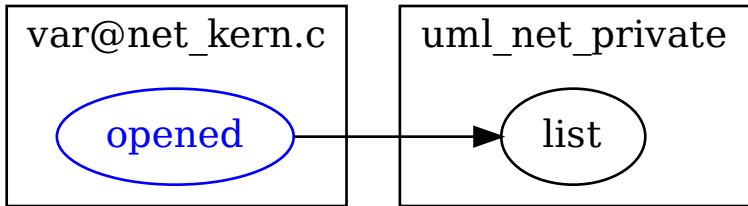


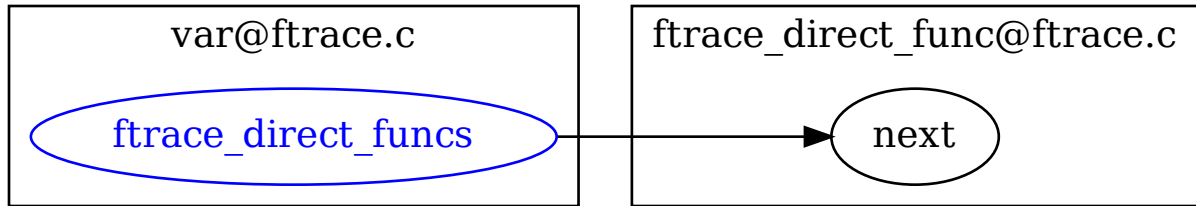


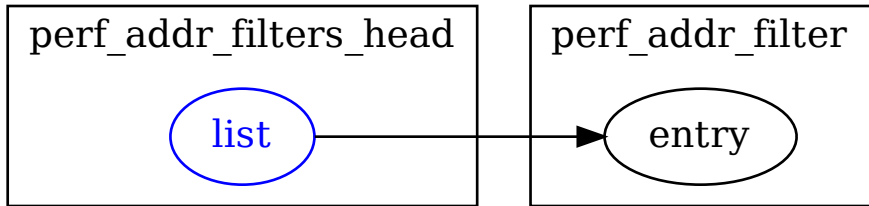


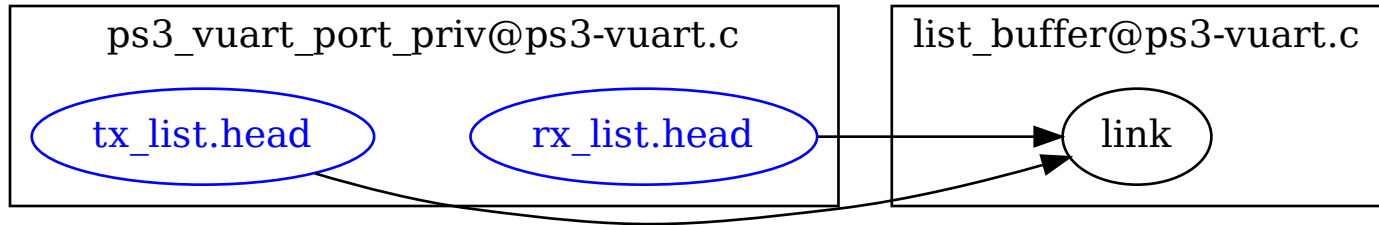


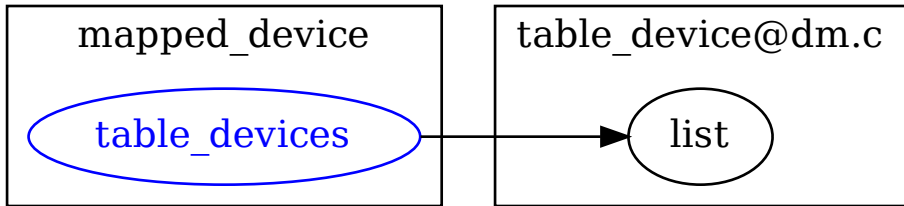


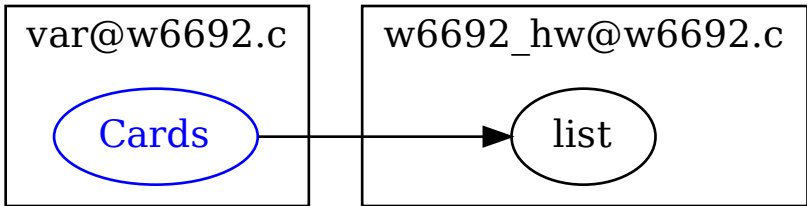


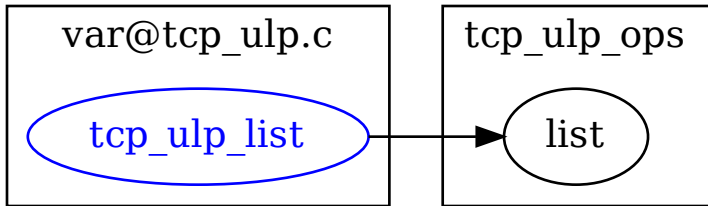


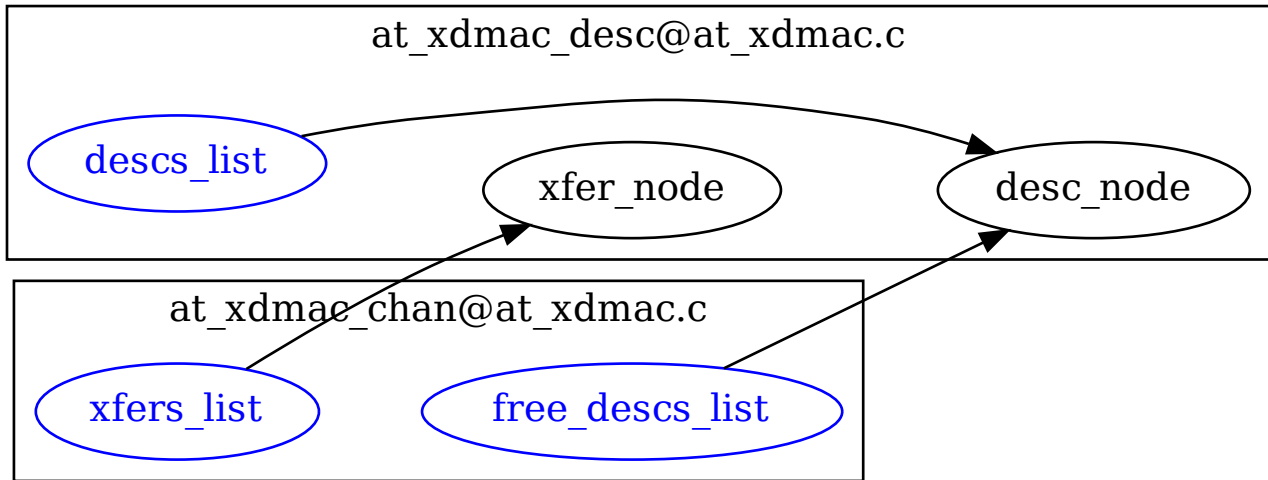


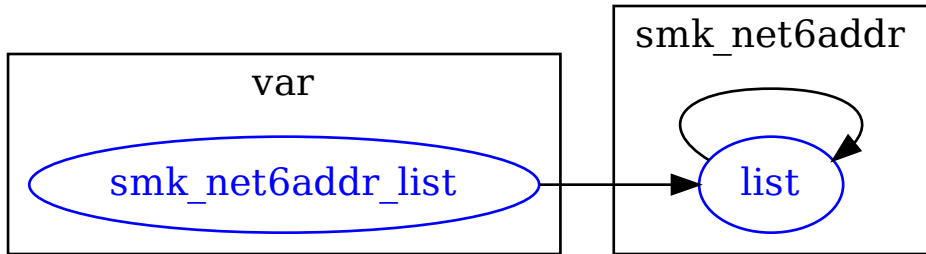


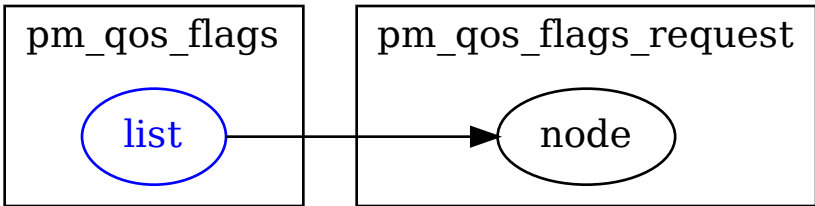


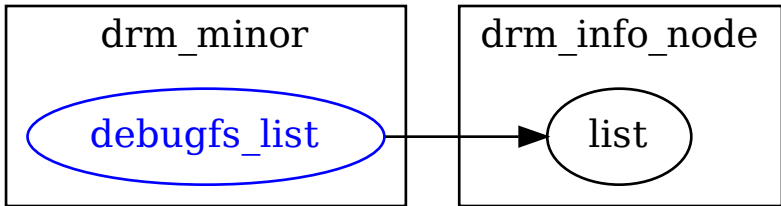


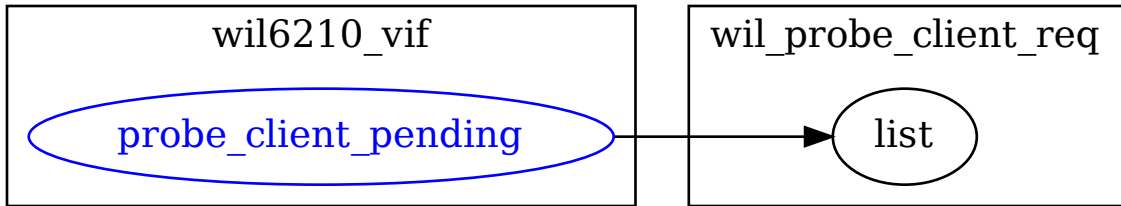


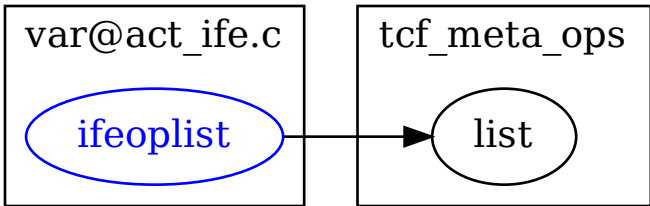












amdgpu_display_manager

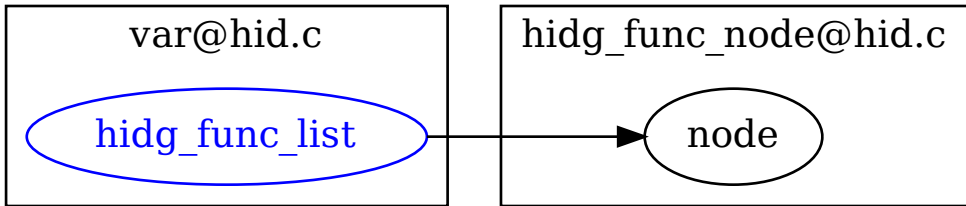
irq_handler_list_high_tab

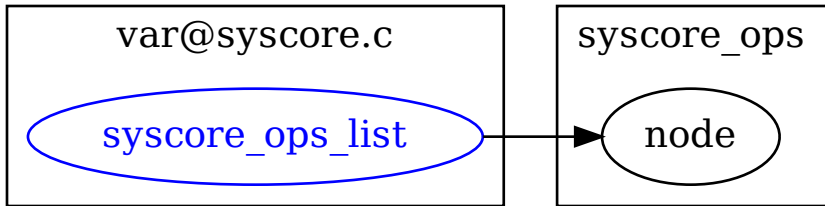
amdgpu_dm_irq_handler_data@amdgpu_dm_irq.c

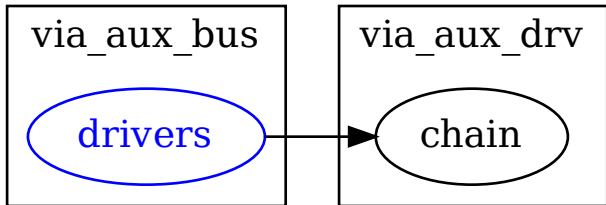
list

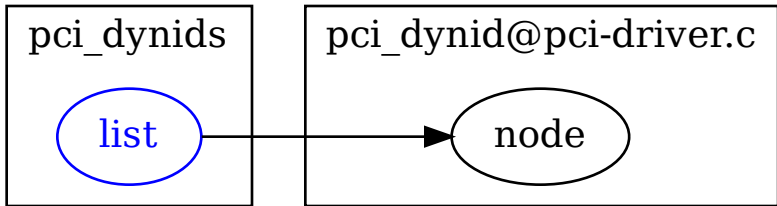
```
graph LR; subgraph A [amdgpu_display_manager]; direction TB; A1[irq_handler_list_high_tab]; end; subgraph B [amdgpu_dm_irq_handler_data@amdgpu_dm_irq.c]; direction TB; B1((list)); end; A1 --> B1;
```

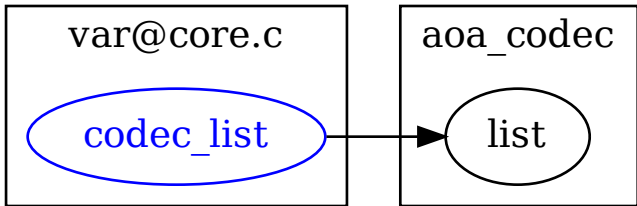
The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'amdgpu_display_manager' contains a variable 'irq_handler_list_high_tab' which is highlighted by a blue oval. On the right, a box labeled 'amdgpu_dm_irq_handler_data@amdgpu_dm_irq.c' contains a variable 'list' enclosed in a black oval. A horizontal arrow points from the blue oval in the first box to the black oval in the second box, indicating that 'irq_handler_list_high_tab' points to 'list'.











var@pci_slot.c

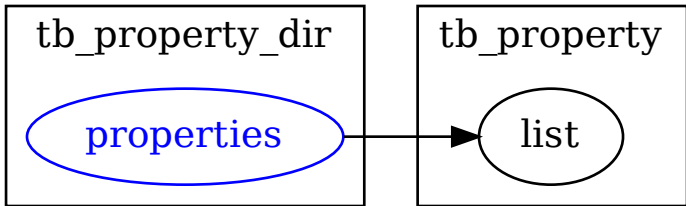
slot_list

acpi_pci_slot@pci_slot.c

list

```
graph LR; subgraph var_pci_slot_c [var@pci_slot.c]; slot_list(slot_list); end; subgraph acpi_pci_slot_pci_slot_c [acpi_pci_slot@pci_slot.c]; list(list); end; slot_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@pci_slot.c' contains a blue oval labeled 'slot_list'. On the right, a box labeled 'acpi_pci_slot@pci_slot.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'slot_list' oval to the 'list' oval, indicating that 'slot_list' points to 'list'.



ti_sci_inta_irq_domain@irq-ti-sci-inta.c

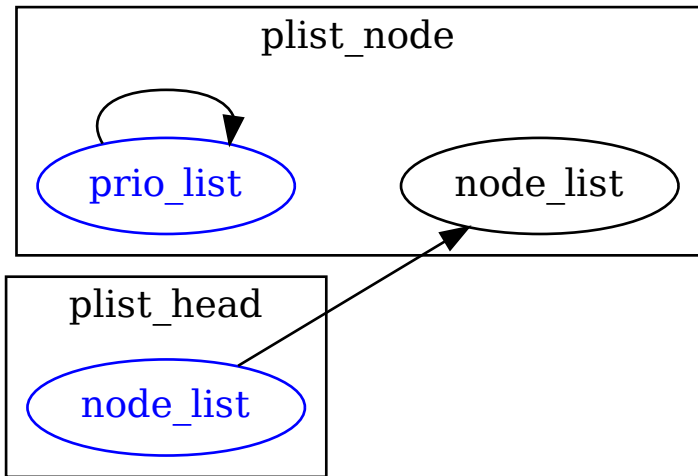
vint_list

ti_sci_inta_vint_desc@irq-ti-sci-inta.c

list

```
graph LR; A[ti_sci_inta_irq_domain@irq-ti-sci-inta.c] --> B[ti_sci_inta_vint_desc@irq-ti-sci-inta.c]; subgraph A; A1((vint_list)); end; subgraph B; B1((list)); end; A1 --> B1;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'ti_sci_inta_irq_domain@irq-ti-sci-inta.c' contains an oval labeled 'vint_list'. On the right, a box labeled 'ti_sci_inta_vint_desc@irq-ti-sci-inta.c' contains an oval labeled 'list'. A horizontal arrow points from the 'vint_list' oval to the 'list' oval, indicating that 'vint_list' points to 'list'.



var@i7core_edac.c

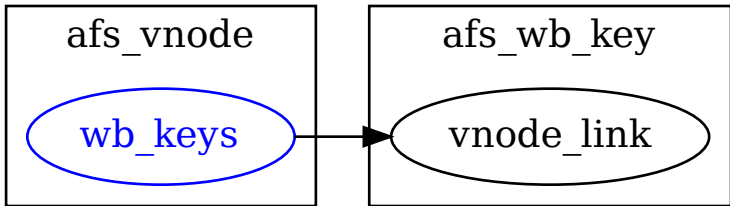
i7core_edac_list

i7core_dev@i7core_edac.c

list

```
graph LR; A["var@i7core_edac.c  
i7core_edac_list"] --> B["i7core_dev@i7core_edac.c  
list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@i7core_edac.c' contains an oval labeled 'i7core_edac_list'. On the right, a box labeled 'i7core_dev@i7core_edac.c' contains an oval labeled 'list'. A horizontal arrow points from the 'i7core_edac_list' oval to the 'list' oval, indicating that the variable 'i7core_edac_list' points to the variable 'list'.



delay_c@dm-delay.c

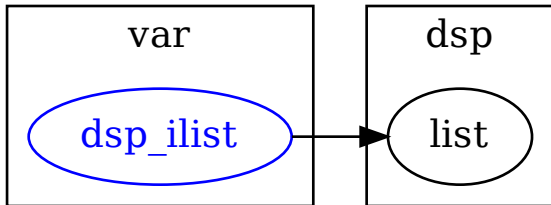
delayed_bios

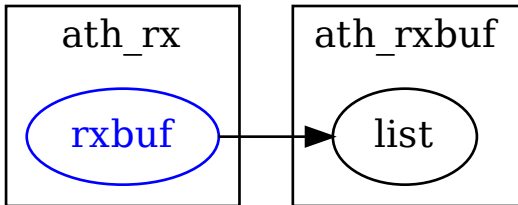
dm_delay_info@dm-delay.c

list

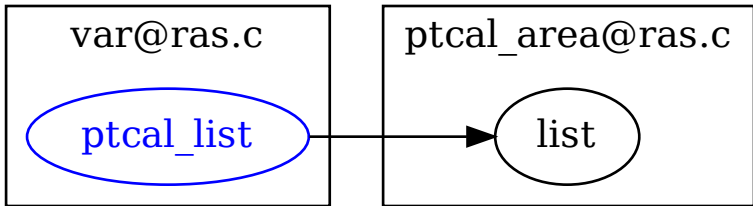
```
graph LR; subgraph Module1 [delay_c@dm-delay.c]; delayed_bios; end; subgraph Module2 [dm_delay_info@dm-delay.c]; list; end; delayed_bios --> list;
```

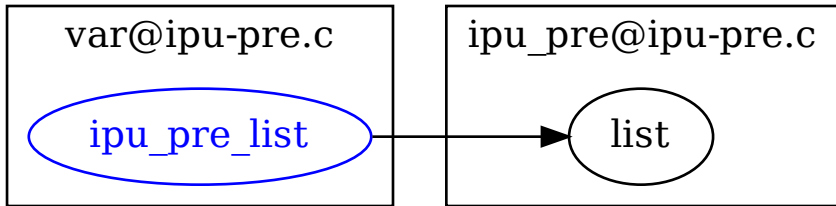
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'delay_c@dm-delay.c' contains an oval labeled 'delayed_bios'. On the right, a box labeled 'dm_delay_info@dm-delay.c' contains an oval labeled 'list'. A horizontal arrow points from the 'delayed_bios' oval to the 'list' oval, indicating that 'delayed_bios' points to 'list'.

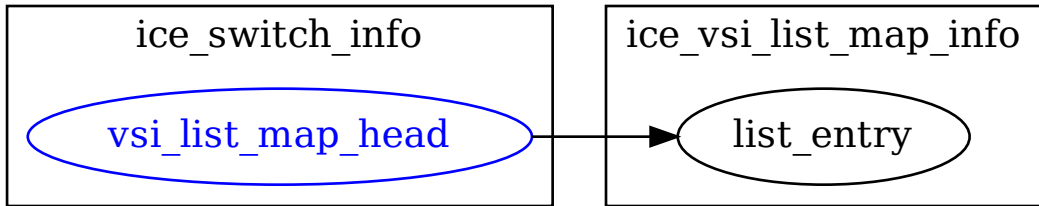


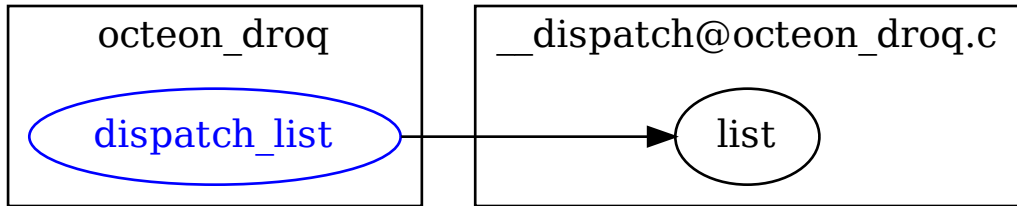












var@error-inject.c

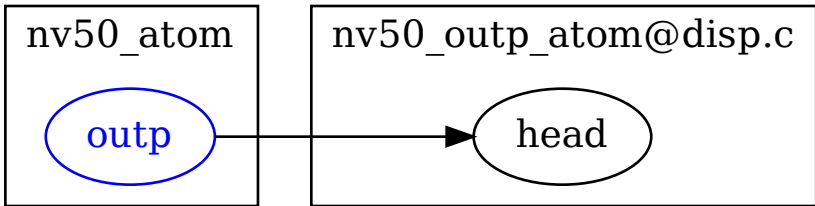
error_injection_list

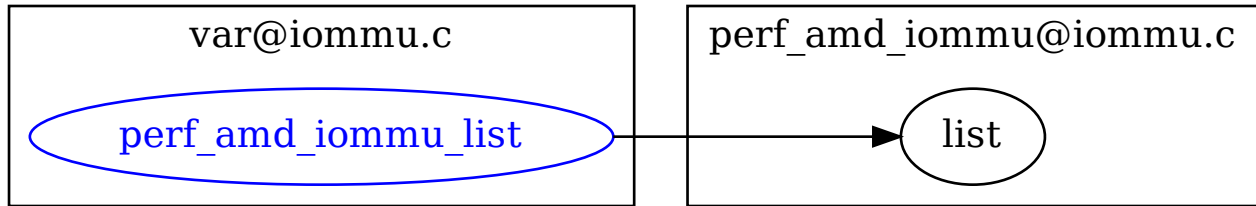
ei_entry@error-inject.c

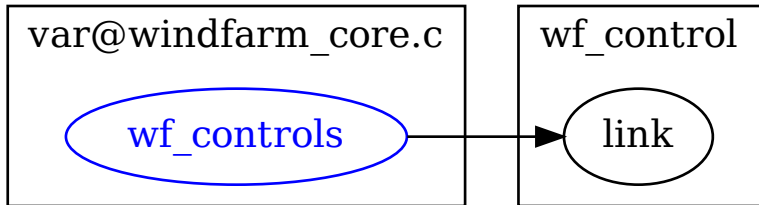
list

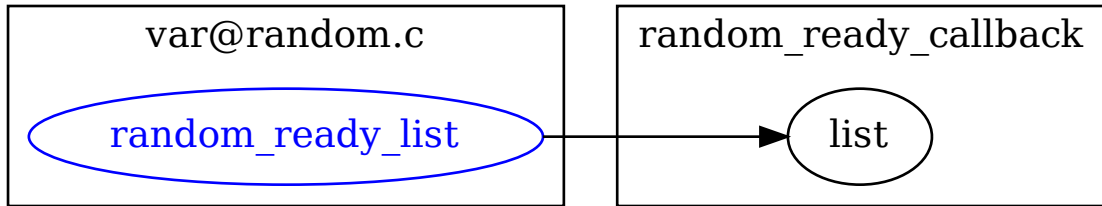
```
graph LR; subgraph "var@error-inject.c"; A([error_injection_list]); end; subgraph "ei_entry@error-inject.c"; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@error-inject.c'. Inside this box, the variable 'error_injection_list' is enclosed in a blue oval. On the right, another rectangular box represents the source file 'ei_entry@error-inject.c'. Inside this box, the variable 'list' is enclosed in a black oval. A horizontal arrow points from the right side of the 'error_injection_list' oval to the left side of the 'list' oval, indicating that 'error_injection_list' points to 'list'.









stm32_crc_list@stm32-crc32.c

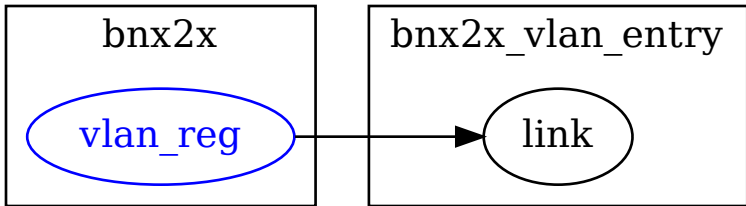
dev_list

stm32_crc@stm32-crc32.c

list



```
graph LR; subgraph Box1 [ ]; direction TB; F1[stm32_crc_list@stm32-crc32.c]; V1((dev_list)); end; subgraph Box2 [ ]; direction TB; F2[stm32_crc@stm32-crc32.c]; V2((list)); end; V1 --> V2;
```



var@pm44xx.c

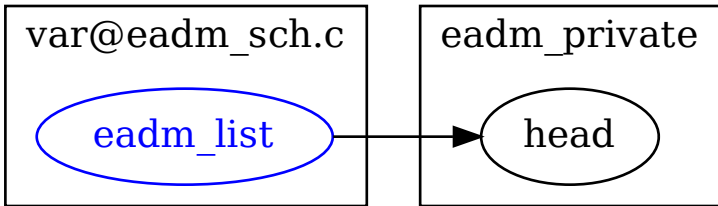
pwrst_list

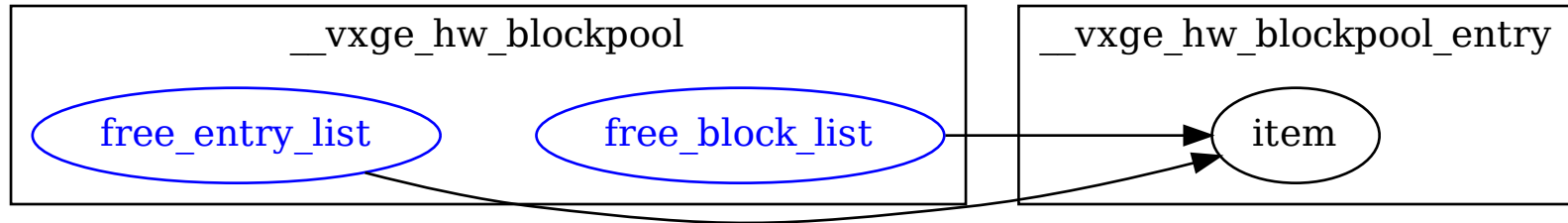
power_state@pm44xx.c

node

```
graph LR; subgraph var [var@pm44xx.c]; pwrst_list; end; subgraph power_state [power_state@pm44xx.c]; node; end; pwrst_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@pm44xx.c' contains an oval labeled 'pwrst_list'. On the right, a box labeled 'power_state@pm44xx.c' contains an oval labeled 'node'. A horizontal arrow points from the 'pwrst_list' oval to the 'node' oval, indicating that 'pwrst_list' points to 'node'.





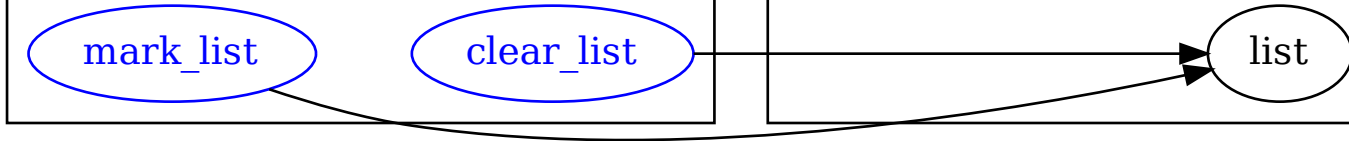
log_c@dm-log-userspace-base.c

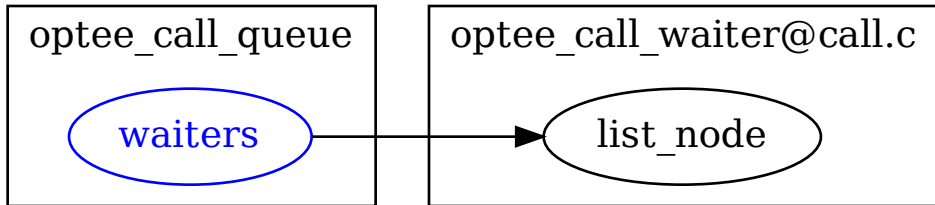
mark_list

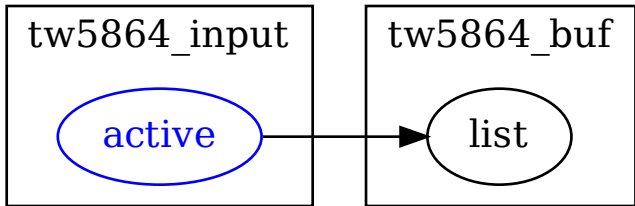
clear_list

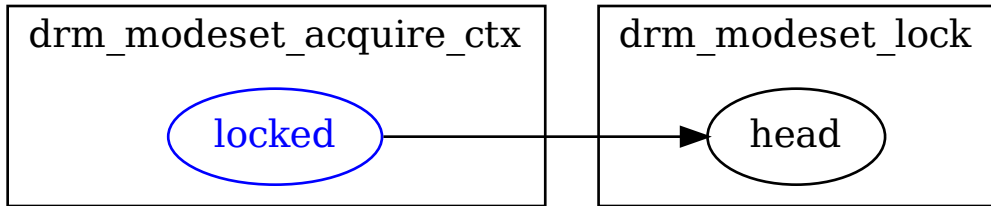
dm_dirty_log_flush_entry@dm-log-userspace-base.c

list









var@inkern.c

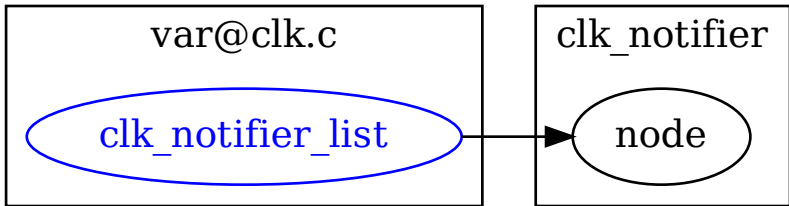
iio_map_list

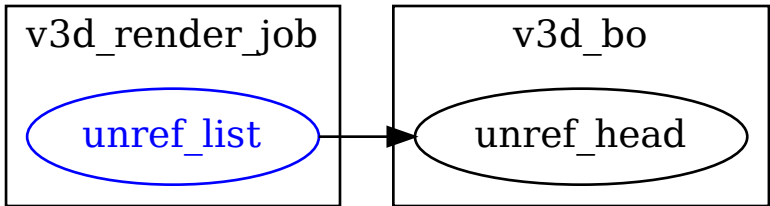
iio_map_internal@inkern.c

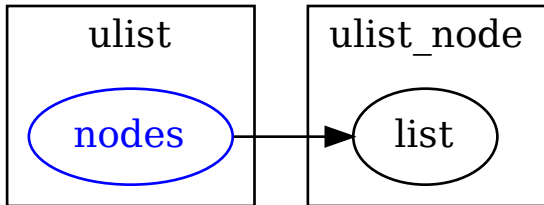
1

```
graph LR; A["var@inkern.c  
iio_map_list"] --> B["iio_map_internal@inkern.c  
1"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@inkern.c' and a blue oval containing 'iio_map_list'. The right box contains the text 'iio_map_internal@inkern.c' and a black oval containing the number '1'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box.







var@hci_sock.c

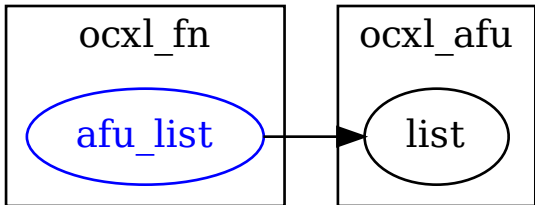
mgmt_chan_list

hci_mgmt_chan

list

```
graph LR; A[var@hci_sock.c] --> B(hci_mgmt_chan); subgraph A [var@hci_sock.c]; C(mgmt_chan_list); end; subgraph B [hci_mgmt_chan]; D((list)); end; C --> D;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@hci_sock.c' contains an oval labeled 'mgmt_chan_list' in blue text. On the right, a box labeled 'hci_mgmt_chan' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'mgmt_chan_list' oval to the left side of the 'list' oval, indicating that 'mgmt_chan_list' points to 'list'.



uniphier_pinctrl_priv@pinctrl-uniphier-core.c

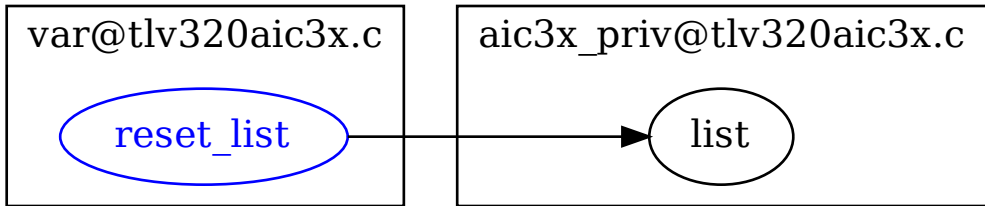
reg_regions

uniphier_pinctrl_reg_region@pinctrl-uniphier-core.c

node

```
graph LR; A[uniphier_pinctrl_priv@pinctrl-uniphier-core.c] --> B[uniphier_pinctrl_reg_region@pinctrl-uniphier-core.c]; B --> C((node));
```

The diagram illustrates a pointer relationship between two structures. On the left, a box represents the `uniphier_pinctrl_priv@pinctrl-uniphier-core.c` structure, which contains a blue oval labeled `reg_regions`. On the right, a box represents the `uniphier_pinctrl_reg_region@pinctrl-uniphier-core.c` structure, which contains a black oval labeled `node`. A horizontal arrow points from the `reg_regions` oval to the `node` oval, indicating that the `reg_regions` pointer in the first structure points to the `node` in the second structure.



var@heartbeat.c

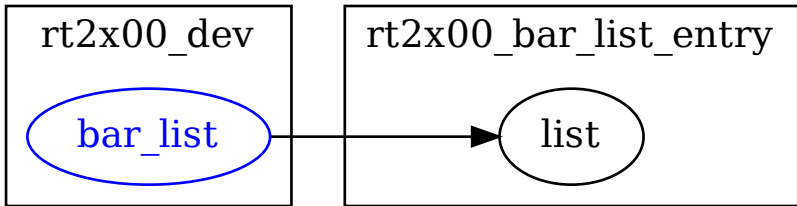
o2hb_all_regions

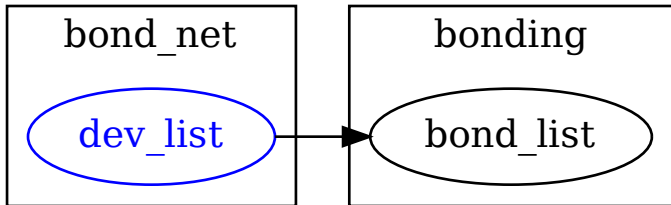
o2hb_region@heartbeat.c

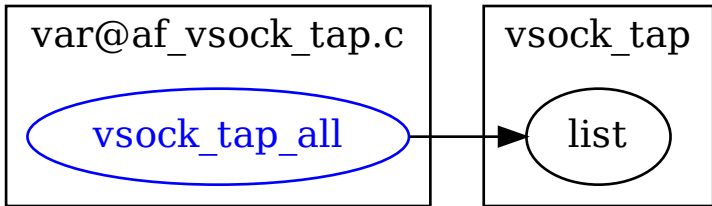
hr_all_item

```
graph LR; subgraph var_at_heartbeat_c [var@heartbeat.c]; o2hb_all_regions; end; subgraph o2hb_region_at_heartbeat_c [o2hb_region@heartbeat.c]; hr_all_item; end; o2hb_all_regions --> hr_all_item;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@heartbeat.c' contains an oval labeled 'o2hb_all_regions'. On the right, a box labeled 'o2hb_region@heartbeat.c' contains an oval labeled 'hr_all_item'. A horizontal arrow points from the 'o2hb_all_regions' oval to the 'hr_all_item' oval, indicating that 'o2hb_all_regions' points to 'hr_all_item'.









tegra_hsp@tegra-hsp.c

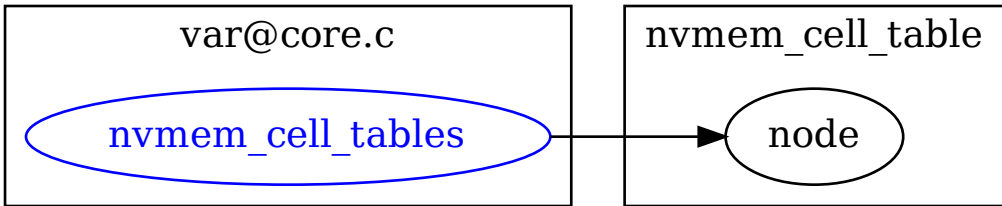
doorbells

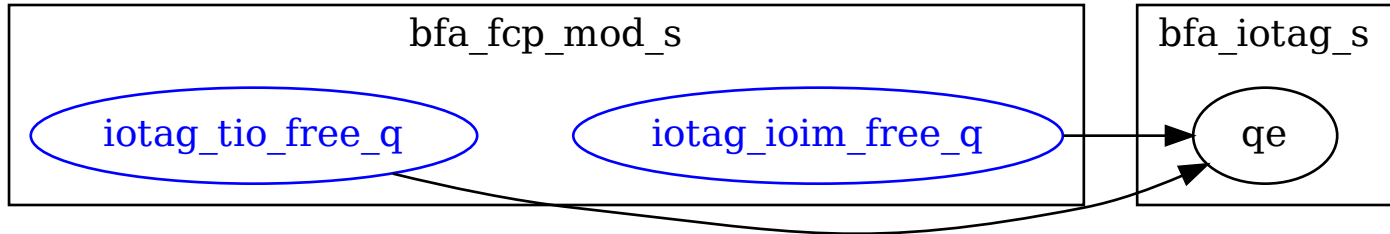
tegra_hsp_doorbell@tegra-hsp.c

list

```
graph LR; subgraph Box1 [tegra_hsp@tegra-hsp.c]; doorbells; end; subgraph Box2 [tegra_hsp_doorbell@tegra-hsp.c]; list; end; doorbells --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'tegra_hsp@tegra-hsp.c' and contains an oval labeled 'doorbells' in blue text. The right box is labeled 'tegra_hsp_doorbell@tegra-hsp.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the 'doorbells' oval in the first box to the 'list' oval in the second box.





binder_transaction@binder.c

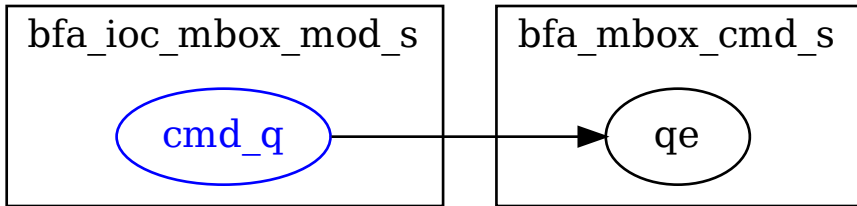
fd_fixups

binder_txn_fd_fixup@binder.c

fixup_entry

```
graph LR; A(binder_transaction@binder.c) --> B(binder_txn_fd_fixup@binder.c); A --> C(fixup_entry);
```

The diagram consists of two rectangular boxes. The left box is labeled 'binder_transaction@binder.c' and contains an oval labeled 'fd_fixups' with a blue border. The right box is labeled 'binder_txn_fd_fixup@binder.c' and contains an oval labeled 'fixup_entry' with a black border. A horizontal arrow points from the 'fd_fixups' oval to the 'fixup_entry' oval, indicating a pointer relationship.



esb2rom_window@esb2rom.c

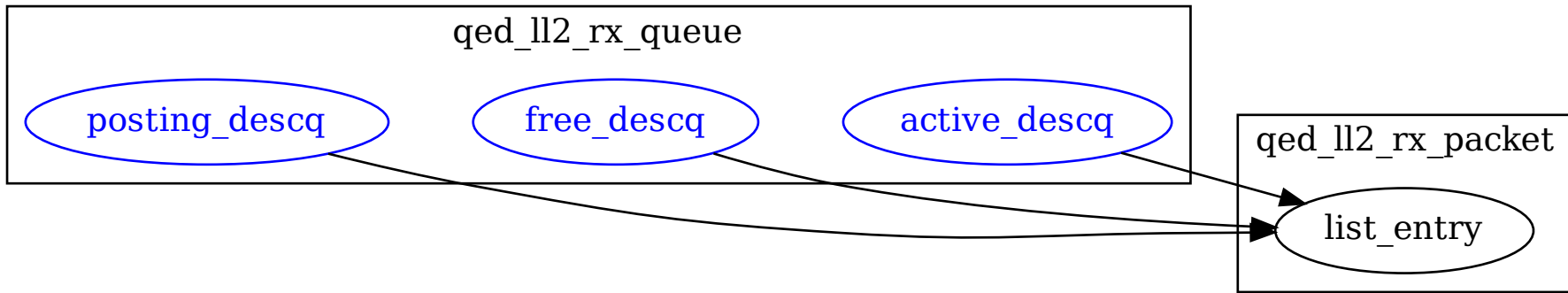
maps

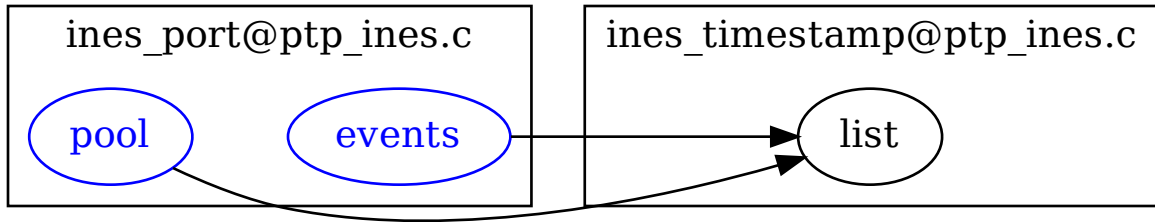
esb2rom_map_info@esb2rom.c

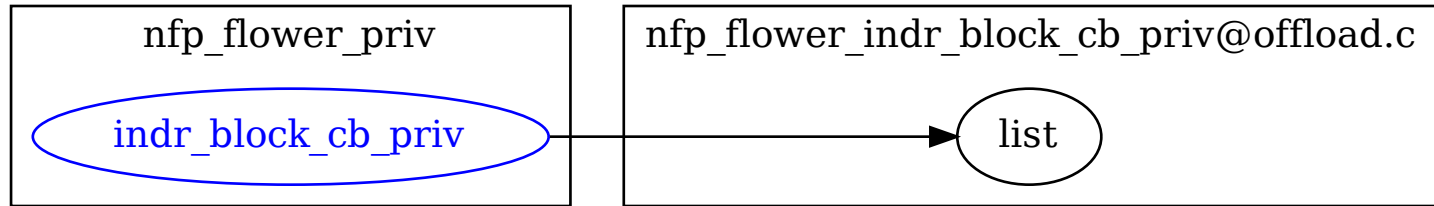
list

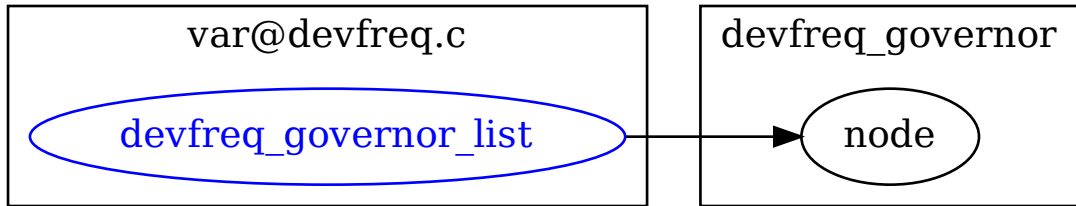
```
graph LR; subgraph Box1 [esb2rom_window@esb2rom.c]; maps([maps]); end; subgraph Box2 [esb2rom_map_info@esb2rom.c]; list([list]); end; maps --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'esb2rom_window@esb2rom.c' and contains an oval labeled 'maps' in blue text. The right box is labeled 'esb2rom_map_info@esb2rom.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the right side of the 'maps' oval to the left side of the 'list' oval, indicating a pointer relationship.









var@saa7134-core.c

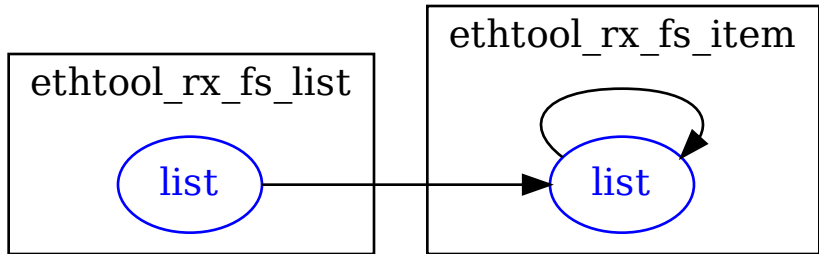
mops_list

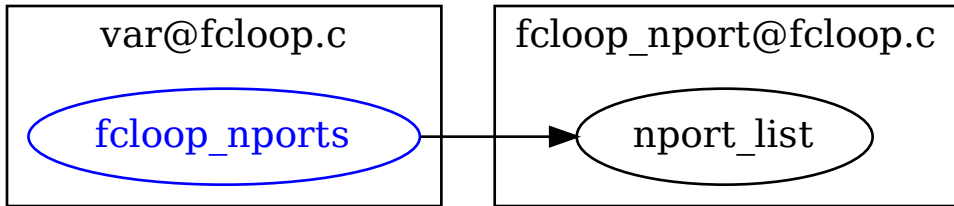
saa7134_mpeg_ops

next

```
graph LR; A[var@saa7134-core.c] --> B[saa7134_mpeg_ops]; subgraph A; A1(mops_list); end; subgraph B; B1(next); end; A1 --> B1;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@saa7134-core.c' contains an oval labeled 'mops_list'. On the right, a box labeled 'saa7134_mpeg_ops' contains an oval labeled 'next'. A horizontal arrow points from the 'mops_list' oval to the 'next' oval, indicating that 'mops_list' points to the 'next' field within the 'saa7134_mpeg_ops' structure.





var@dm-log-userspace-transfer.c

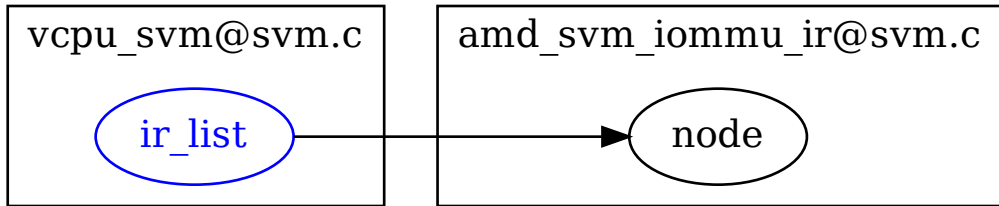
receiving_list

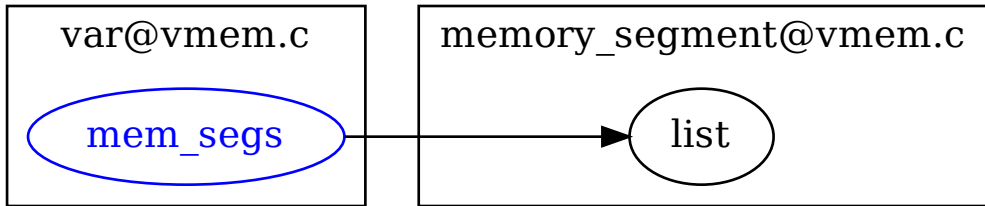
receiving_pkg@dm-log-userspace-transfer.c

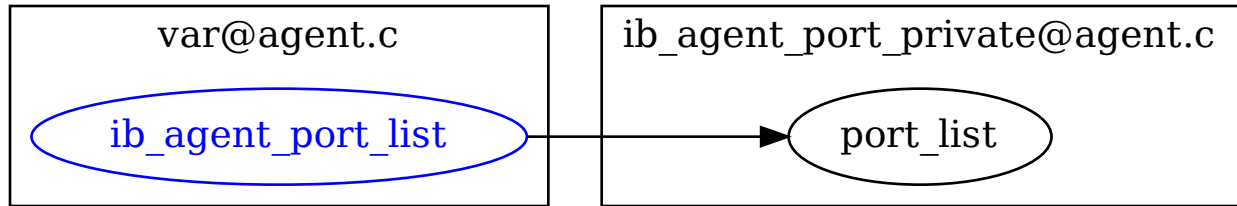
list

```
graph LR; A[receiving_list] --> B((list));
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@dm-log-userspace-transfer.c' contains an oval labeled 'receiving_list' in blue text. On the right, a box labeled 'receiving_pkg@dm-log-userspace-transfer.c' contains an oval labeled 'list'. A horizontal arrow points from the 'receiving_list' oval to the 'list' oval, indicating that 'receiving_list' points to 'list'.







kvm_vfio@vfio.c

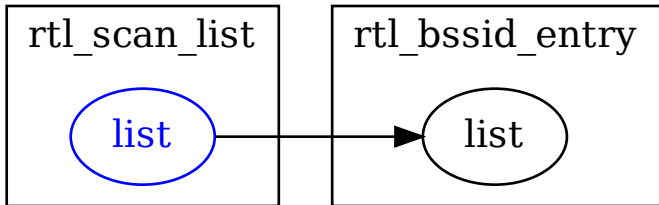
group_list

kvm_vfio_group@vfio.c

node

```
graph LR; subgraph "kvm_vfio@vfio.c"; GL([group_list]); end; subgraph "kvm_vfio_group@vfio.c"; N([node]); end; GL --> N;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'kvm_vfio@vfio.c' contains an oval labeled 'group_list'. On the right, a box labeled 'kvm_vfio_group@vfio.c' contains an oval labeled 'node'. A horizontal arrow points from the 'group_list' oval to the 'node' oval, indicating that 'group_list' points to 'node'.



var@klist.c

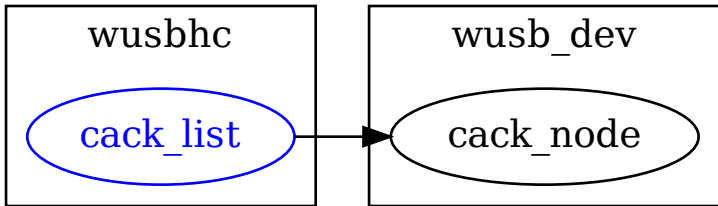
klist_remove_waiters

klist_waiter@klist.c

list

```
graph LR; subgraph var_klist_c [var@klist.c]; klist_remove_waiters; end; subgraph klist_waiter_klist_c [klist_waiter@klist.c]; list; end; klist_remove_waiters --> list;
```

The diagram illustrates a pointer relationship between two code blocks. On the left, a box labeled 'var@klist.c' contains an oval labeled 'klist_remove_waiters'. On the right, a box labeled 'klist_waiter@klist.c' contains an oval labeled 'list'. A horizontal arrow points from the 'klist_remove_waiters' oval to the 'list' oval, indicating that the function 'klist_remove_waiters' points to the 'list' variable.



nouveau_svm@nouveau_svm.c

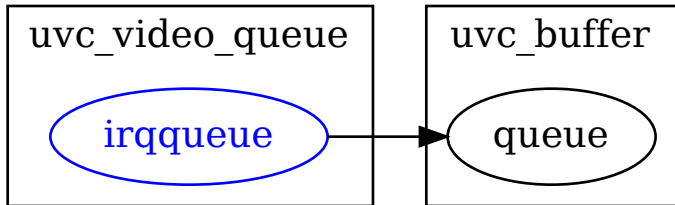
inst

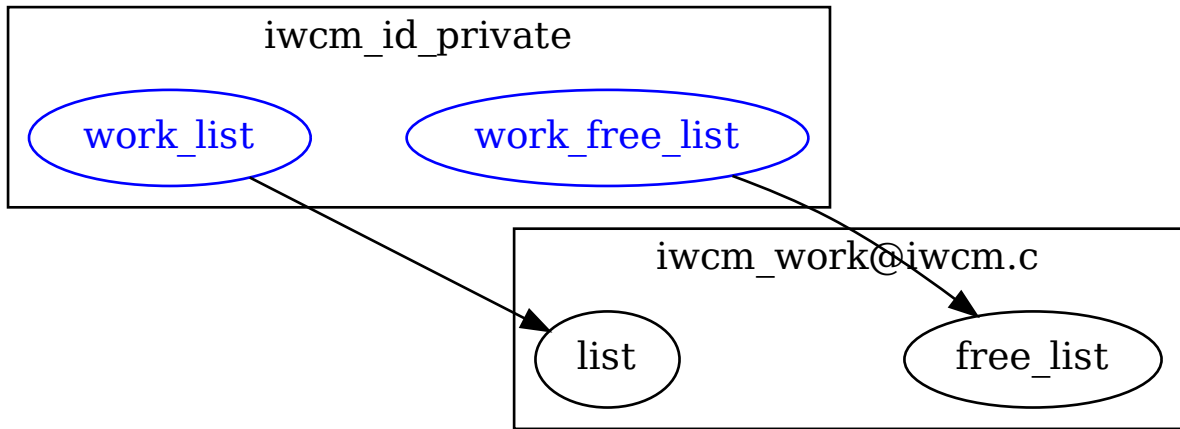
nouveau_ivmm@nouveau_svm.c

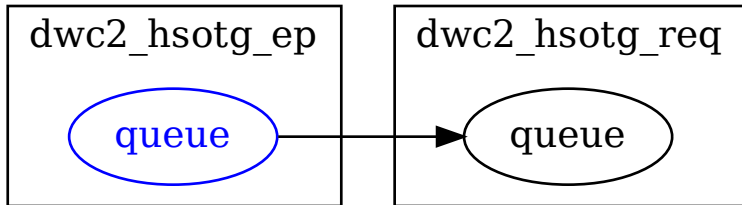
head

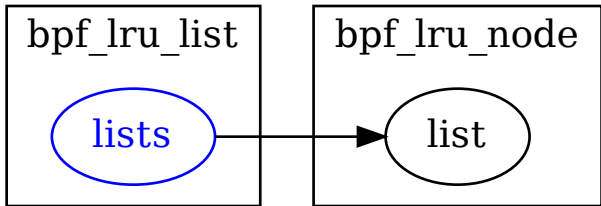
```
graph LR; subgraph NouveauSVM [nouveau_svm@nouveau_svm.c]; inst((inst)); end; subgraph NouveauIVMM [nouveau_ivmm@nouveau_svm.c]; head((head)); end; inst --> head;
```

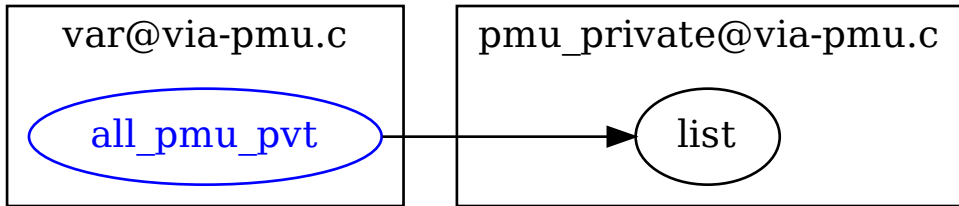
The diagram consists of two rectangular boxes. The left box is labeled 'nouveau_svm@nouveau_svm.c' and contains a blue oval labeled 'inst'. The right box is labeled 'nouveau_ivmm@nouveau_svm.c' and contains a black oval labeled 'head'. A horizontal arrow points from the right side of the 'inst' oval to the left side of the 'head' oval, crossing the boundary between the two boxes.











var@omapdss-boot-init.c

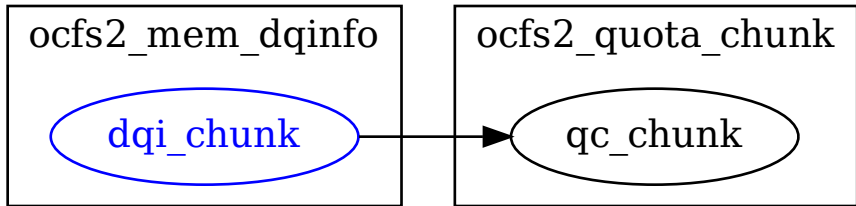
dss_conv_list

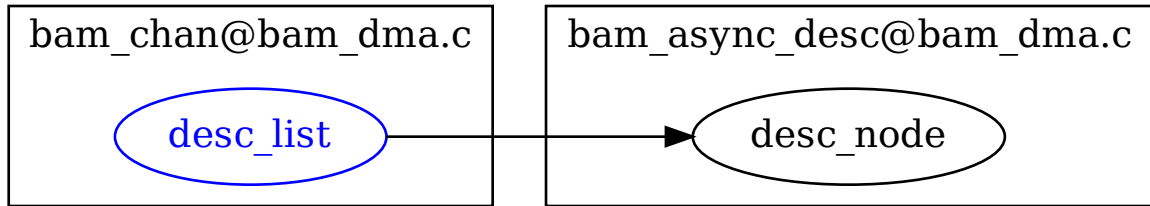
dss_conv_node@omapdss-boot-init.c

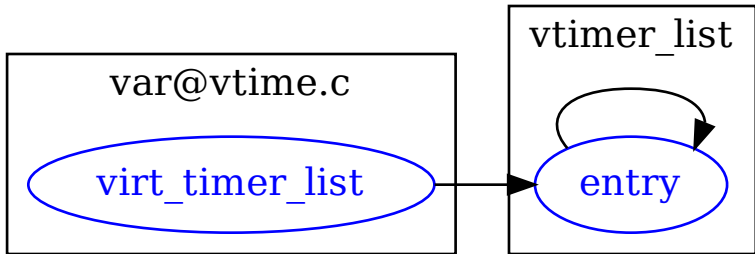
list

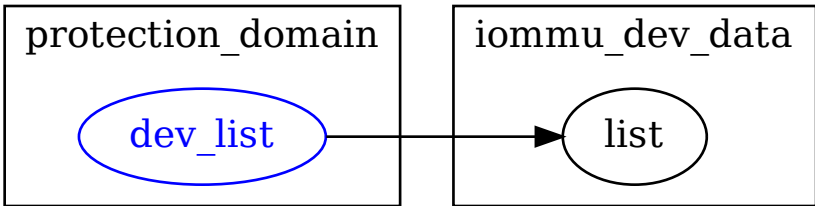
```
graph LR; A([dss_conv_list]) --> B([list]);
```

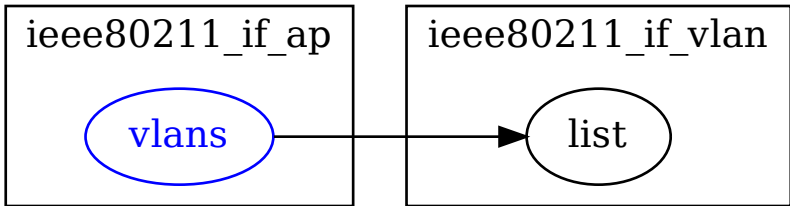
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@omapdss-boot-init.c'. Inside this box, the text 'dss_conv_list' is enclosed in a blue oval. On the right, another rectangular box contains the text 'dss_conv_node@omapdss-boot-init.c'. Inside this box, the text 'list' is enclosed in a black oval. A horizontal arrow points from the blue oval 'dss_conv_list' to the black oval 'list', indicating that 'dss_conv_list' points to 'list'.

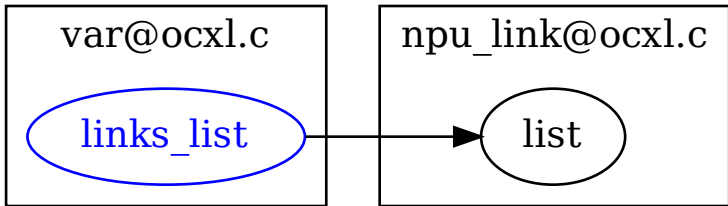


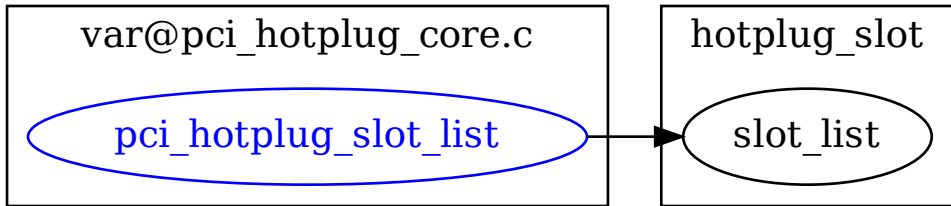












var@ab8500_btemp.c

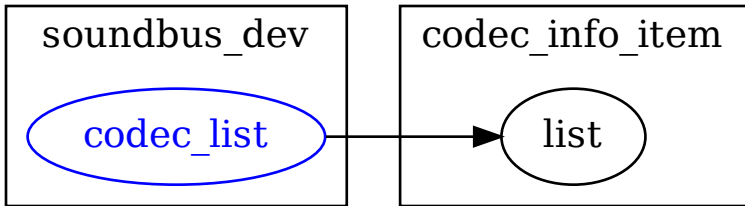
ab8500_btemp_list

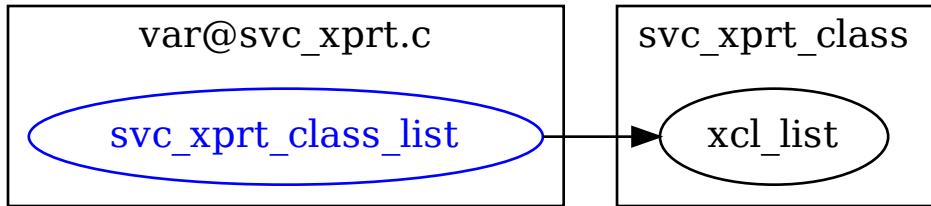
ab8500_btemp@ab8500_btemp.c

node

```
graph LR; A["var@ab8500_btemp.c  
ab8500_btemp_list"] --> B["ab8500_btemp@ab8500_btemp.c  
node"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@ab8500_btemp.c' at the top and 'ab8500_btemp_list' at the bottom, which is enclosed in a blue oval. The right box contains the text 'ab8500_btemp@ab8500_btemp.c' at the top and 'node' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.





var@attribute_container.c

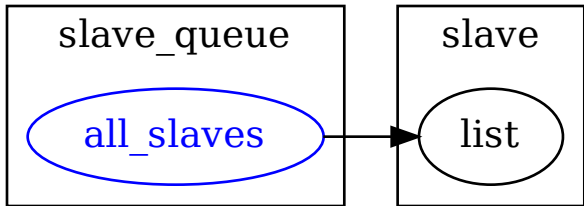
attribute_container_list

attribute_container

node

```
graph LR; subgraph var [var@attribute_container.c]; direction TB; A(attribute_container_list); end; subgraph attr [attribute_container]; direction TB; B((node)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@attribute_container.c' contains an oval labeled 'attribute_container_list' which is highlighted with a blue border. On the right, a rectangular box labeled 'attribute_container' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'attribute_container_list' oval to the left side of the 'node' oval, indicating that 'attribute_container_list' points to 'node'.



nsim_fib_data@fib.c

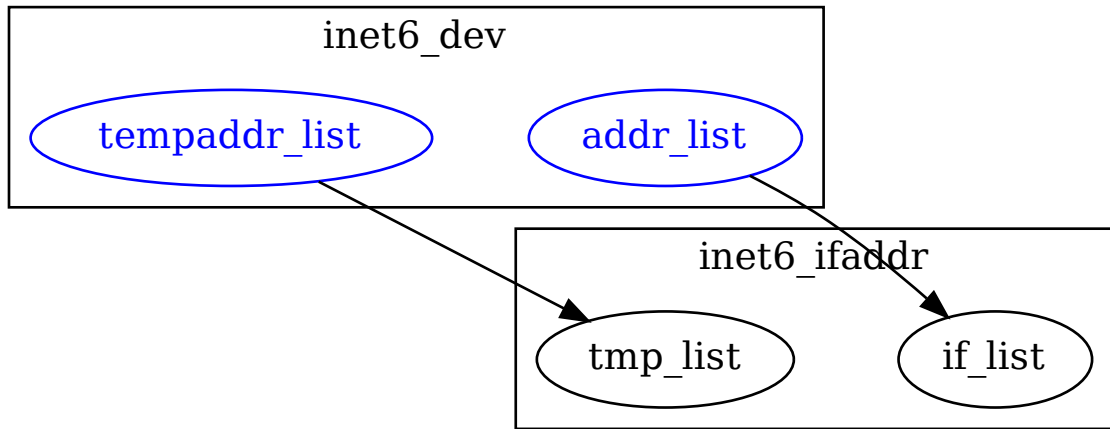
fib_rt_list

nsim_fib_rt@fib.c

list

```
graph LR; subgraph Module1 [nsim_fib_data@fib.c]; fib_rt_list; end; subgraph Module2 [nsim_fib_rt@fib.c]; list; end; fib_rt_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'nsim_fib_data@fib.c' contains an oval labeled 'fib_rt_list'. On the right, a box labeled 'nsim_fib_rt@fib.c' contains an oval labeled 'list'. A horizontal arrow points from the 'fib_rt_list' oval to the 'list' oval, indicating that the 'fib_rt_list' variable in the first module points to the 'list' variable in the second module.

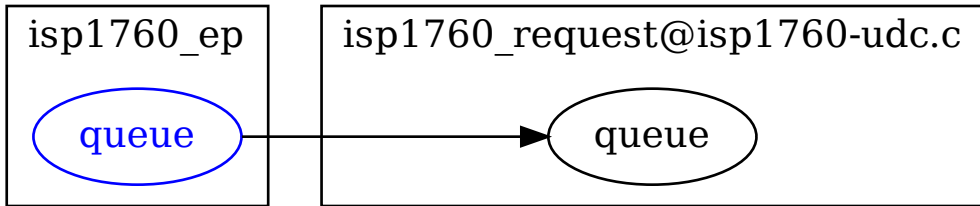


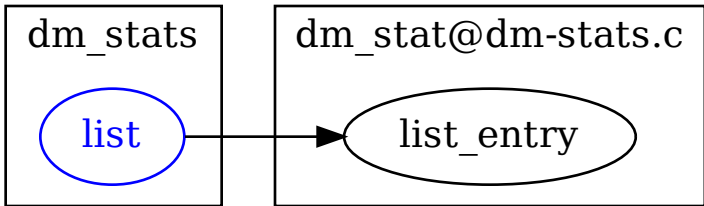
isp1760_ep

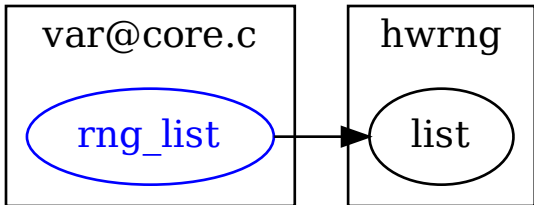
queue

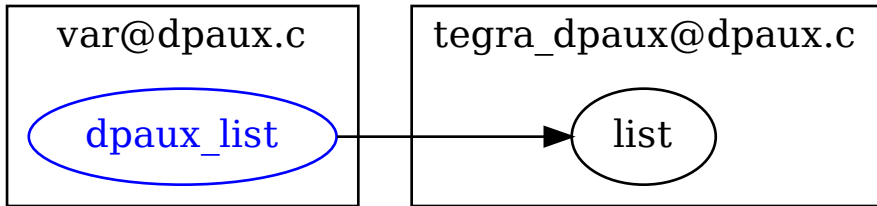
isp1760_request@isp1760-udc.c

queue









var@smscoreapi.c

g_smscore_registry

smscore_registry_entry_t@smscoreapi.c

entry

```
graph LR; subgraph var ["var@smscoreapi.c"]; g["g_smscore_registry"]; end; subgraph smscore_registry_entry_t ["smscore_registry_entry_t@smscoreapi.c"]; entry["entry"]; end; g --> entry;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@smscoreapi.c' contains the variable 'g_smscore_registry', which is highlighted with a blue oval. On the right, a box labeled 'smscore_registry_entry_t@smscoreapi.c' contains the variable 'entry', which is highlighted with a black oval. A horizontal arrow points from the blue oval of 'g_smscore_registry' to the black oval of 'entry', indicating that 'g_smscore_registry' points to 'entry'.

mlxsw_afk@core_acl_flex_keys.c

key_info_list

mlxsw_afk_key_info@core_acl_flex_keys.c

list

```
graph LR; A["mlxsw_afk@core_acl_flex_keys.c  
key_info_list"] --> B["mlxsw_afk_key_info@core_acl_flex_keys.c  
list"]
```

The diagram consists of two rectangular boxes connected by a horizontal arrow. The left box contains the text 'mlxsw_afk@core_acl_flex_keys.c' and a blue oval containing 'key_info_list'. The right box contains the text 'mlxsw_afk_key_info@core_acl_flex_keys.c' and an oval containing 'list'. A horizontal arrow points from the blue oval in the left box to the oval in the right box.

_MPT_SCSI_HOST

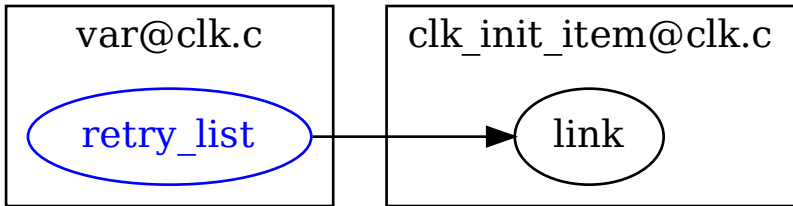
target_reset_list

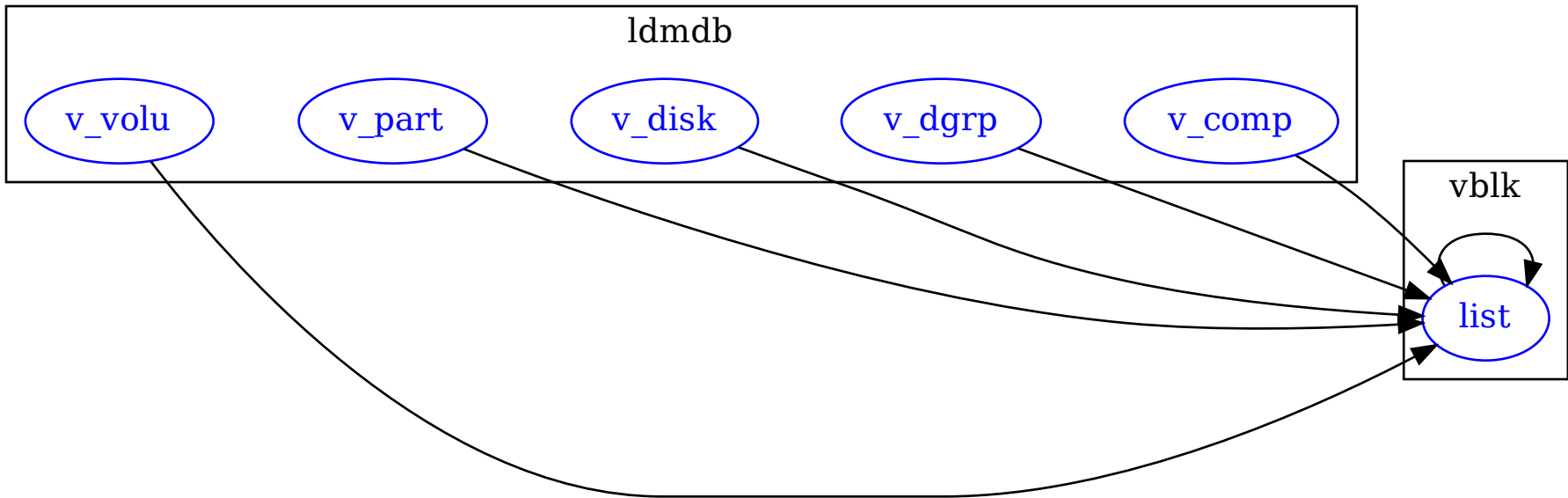
mptsas_target_reset_event

list

```
graph LR; A["_MPT_SCSI_HOST<br/>target_reset_list"] --> B["mptsas_target_reset_event<br/>list"]
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled `_MPT_SCSI_HOST` contains an oval labeled `target_reset_list`. On the right, a rectangular box labeled `mptsas_target_reset_event` contains an oval labeled `list`. A horizontal arrow points from the `target_reset_list` oval to the `list` oval, indicating that the `target_reset_list` pointer points to the `list` structure within the `mptsas_target_reset_event` object.





var@zcrypt_ep11misc.c

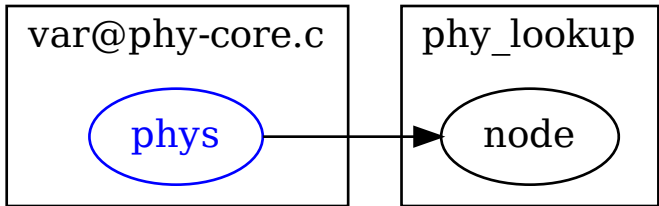
card_list

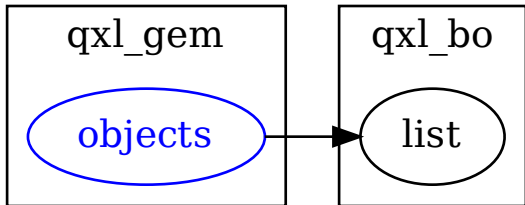
card_list_entry@zcrypt_ep11misc.c

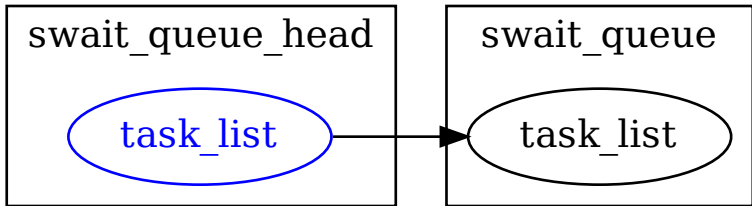
list

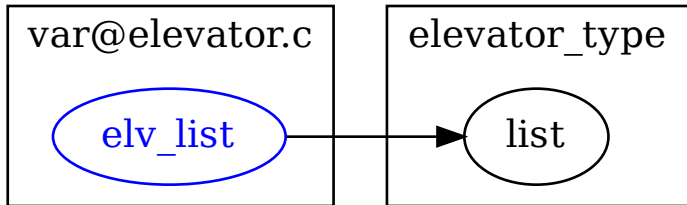
```
graph LR; A["var@zcrypt_ep11misc.c  
card_list"] --> B["card_list_entry@zcrypt_ep11misc.c  
list"]
```

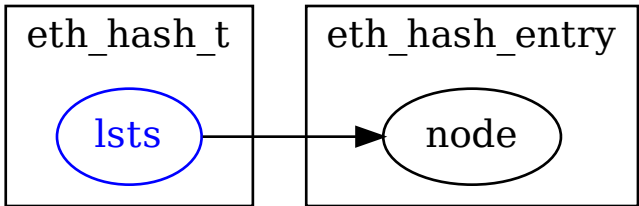
The diagram consists of two rectangular boxes. The left box contains the text 'var@zcrypt_ep11misc.c' at the top and 'card_list' in a blue oval below it. The right box contains the text 'card_list_entry@zcrypt_ep11misc.c' at the top and 'list' in a black oval below it. A horizontal arrow points from the right side of the left box to the left side of the right box, specifically pointing to the 'list' oval.

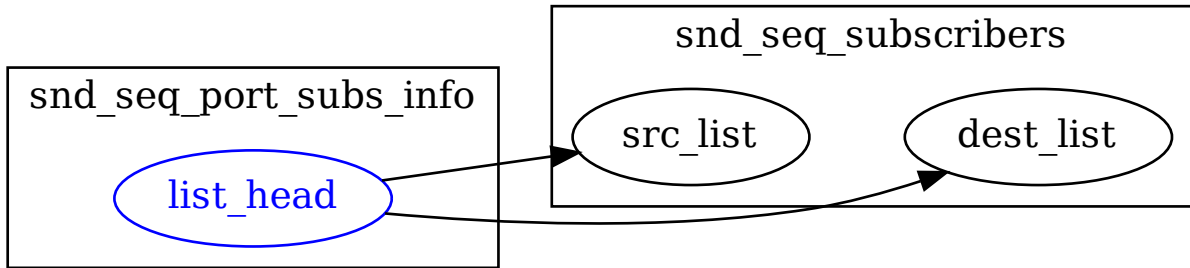


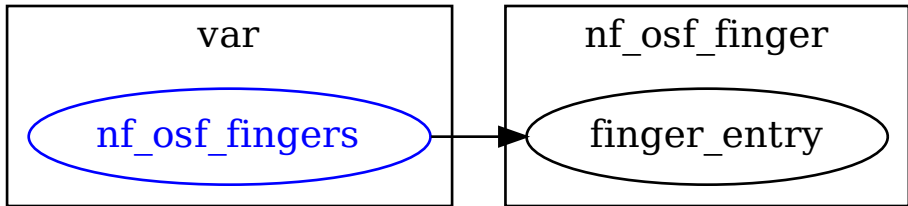












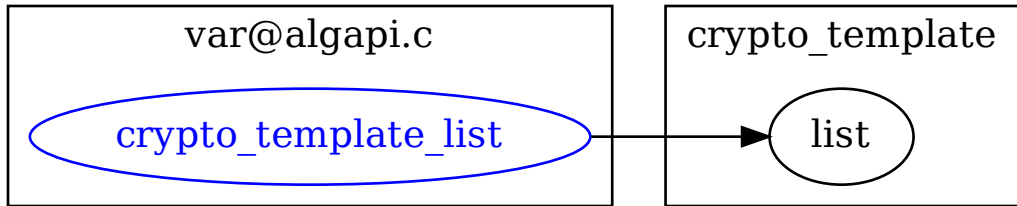
var@charger-manager.c

cm_list

charger_manager

entry





mtk_nfc@mtk_nand.c

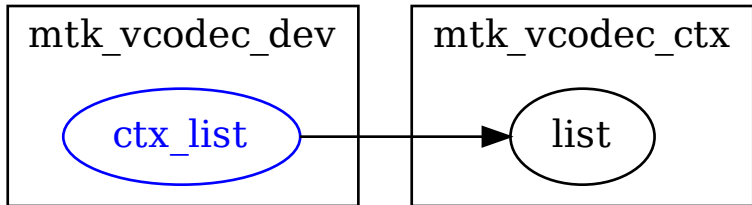
chips

mtk_nfc_nand_chip@mtk_nand.c

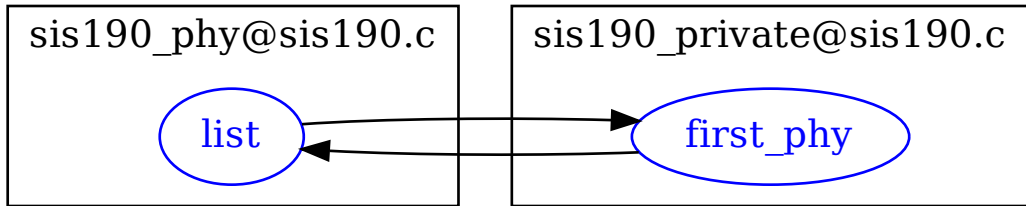
node

```
graph LR; A["mtk_nfc@mtk_nand.c  
chips"] --> B["mtk_nfc_nand_chip@mtk_nand.c  
node"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'mtk_nfc@mtk_nand.c' at the top and a blue oval containing the word 'chips' below it. The right box contains the text 'mtk_nfc_nand_chip@mtk_nand.c' at the top and a black oval containing the word 'node' below it. A horizontal arrow points from the right side of the 'chips' oval to the left side of the 'node' oval, indicating a pointer or reference relationship.







uvcg_streaming_header@uvc_configfs.c

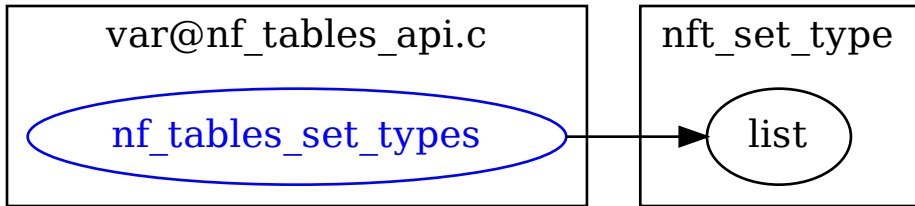
formats

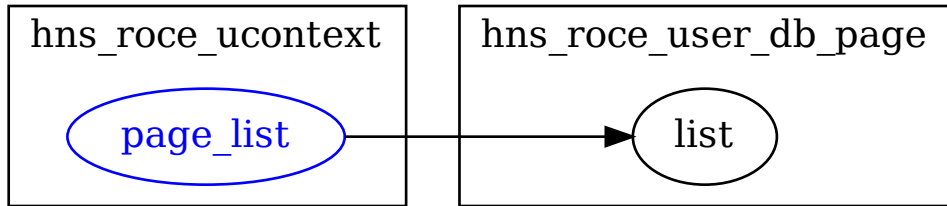
uvcg_format_ptr@uvc_configfs.c

entry

```
graph LR; subgraph Box1 [uvcg_streaming_header@uvc_configfs.c]; formats; end; subgraph Box2 [uvcg_format_ptr@uvc_configfs.c]; entry; end; formats --> entry;
```

The diagram consists of two rectangular boxes. The left box is titled 'uvcg_streaming_header@uvc_configfs.c' and contains an oval labeled 'formats' in blue text. The right box is titled 'uvcg_format_ptr@uvc_configfs.c' and contains an oval labeled 'entry' in black text. A horizontal arrow points from the 'formats' oval in the first box to the 'entry' oval in the second box.





wcd_slim_codec_dai_data@wcd934x.c

slim_ch_list

wcd934x_slim_ch@wcd934x.c

list

```
graph LR; A[wcd_slim_codec_dai_data@wcd934x.c] --> B((list)); B --- C[wcd934x_slim_ch@wcd934x.c];
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'wcd_slim_codec_dai_data@wcd934x.c' contains a blue oval labeled 'slim_ch_list'. On the right, a box labeled 'wcd934x_slim_ch@wcd934x.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'slim_ch_list' oval to the 'list' oval, indicating that the first module points to the second. The arrow is positioned between the two boxes, with its tail at the right side of the first box and its head at the left side of the second box.

var@armv8_deprecated.c

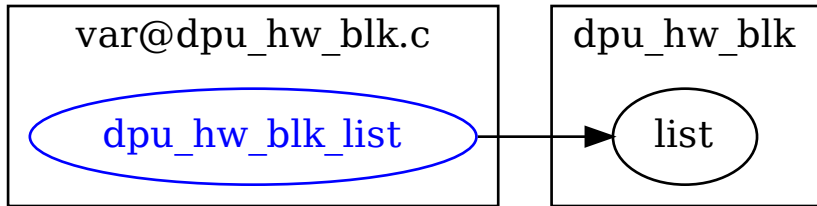
insn_emulation

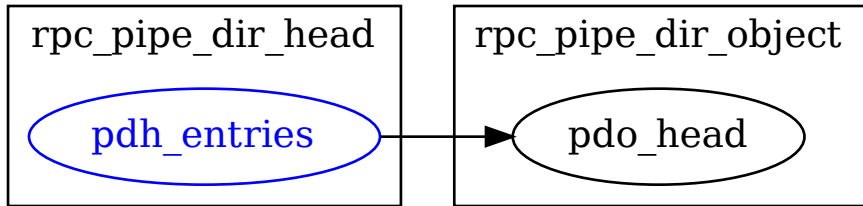
insn_emulation@armv8_deprecated.c

node

```
graph LR; subgraph Box1 [var@armv8_deprecated.c]; direction TB; A([insn_emulation]); end; subgraph Box2 [insn_emulation@armv8_deprecated.c]; direction TB; B([node]); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables defined in the same source file, `armv8_deprecated.c`. On the left, a box labeled `var@armv8_deprecated.c` contains a variable `insn_emulation`, which is highlighted with a blue oval. On the right, a box labeled `insn_emulation@armv8_deprecated.c` contains a variable `node`, which is enclosed in a black oval. A horizontal arrow points from the `insn_emulation` variable in the first box to the `node` variable in the second box, indicating that `insn_emulation` points to `node`.



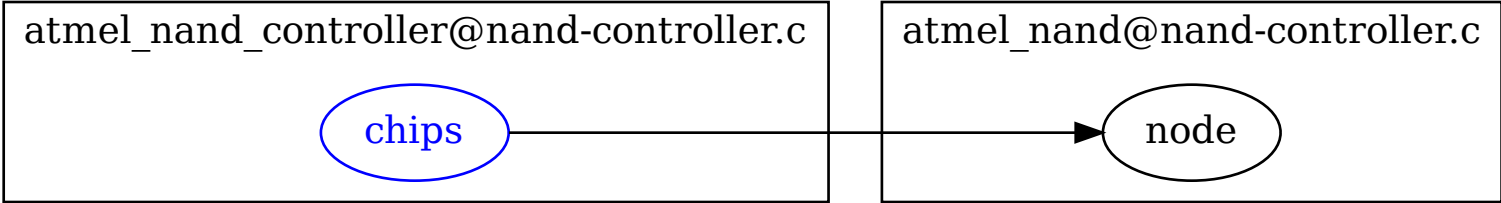


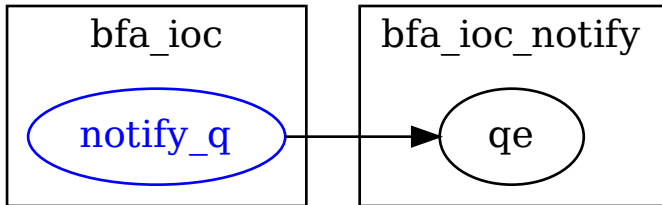
atmel_nand_controller@nand-controller.c

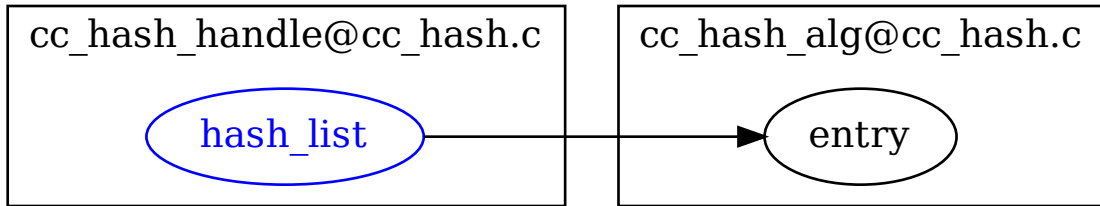
chips

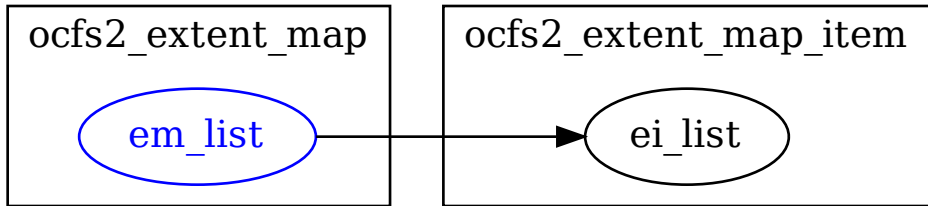
atmel_nand@nand-controller.c

node









var@aer_inject.c

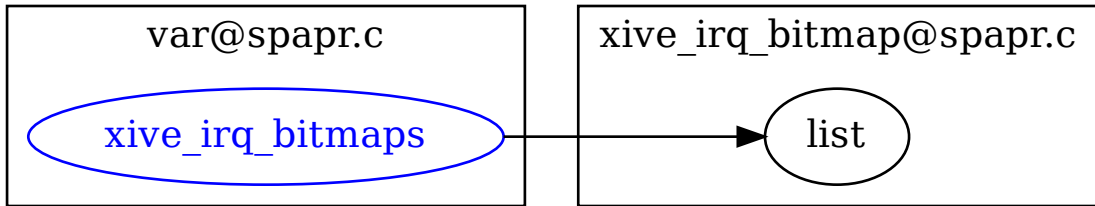
einjected

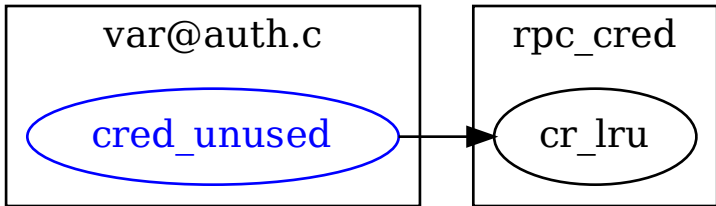
aer_error@aer_inject.c

list

```
graph LR; subgraph Box1 [var@aer_inject.c]; einjected; end; subgraph Box2 [aer_error@aer_inject.c]; list; end; einjected --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@aer_inject.c' and contains a blue oval labeled 'einjected'. The right box is labeled 'aer_error@aer_inject.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'einjected' oval to the left side of the 'list' oval.





var@hid-steam.c

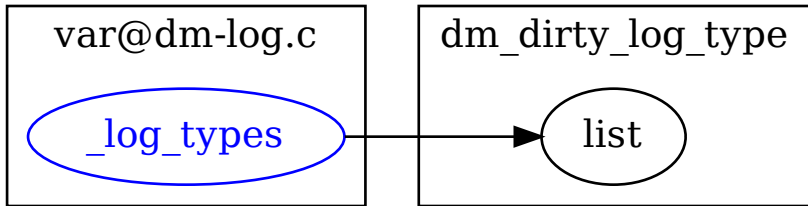
steam_devices

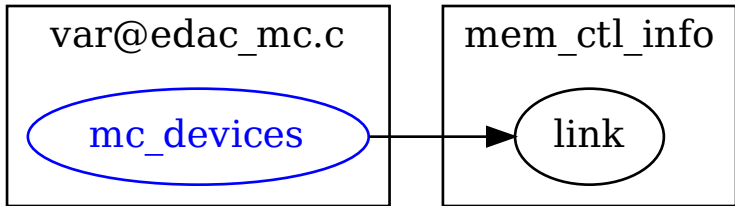
steam_device@hid-steam.c

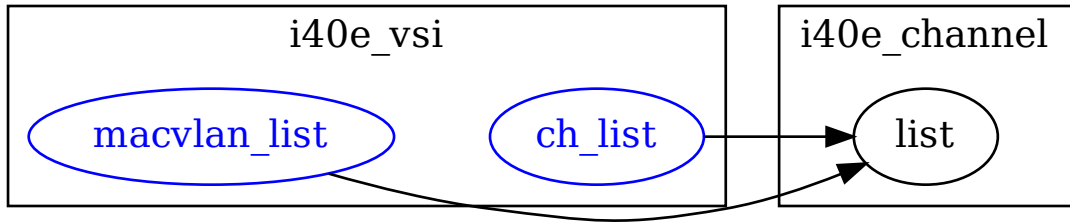
list

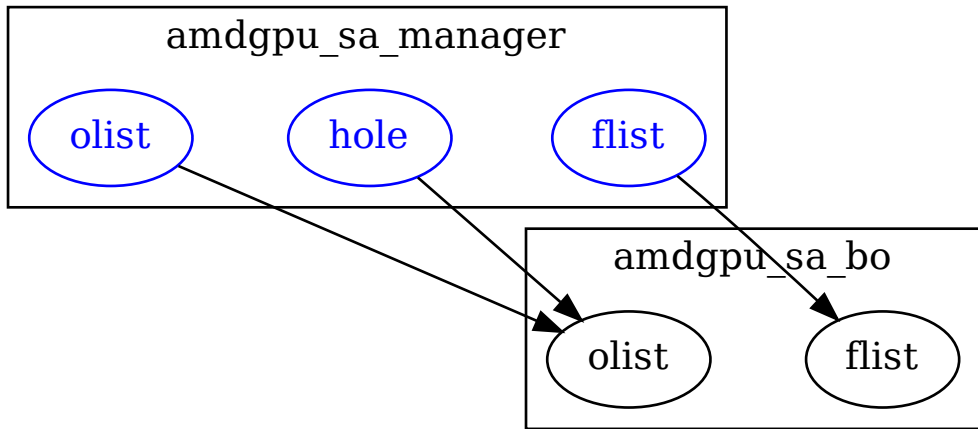
```
graph LR; subgraph Box1 [var@hid-steam.c]; SD([steam_devices]); end; subgraph Box2 [steam_device@hid-steam.c]; L([list]); end; SD --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@hid-steam.c' and contains a blue oval labeled 'steam_devices'. The right box is labeled 'steam_device@hid-steam.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'steam_devices' oval to the left side of the 'list' oval.









var@drivetemp.c

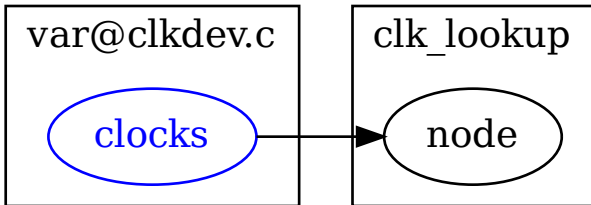
drivetemp_devlist

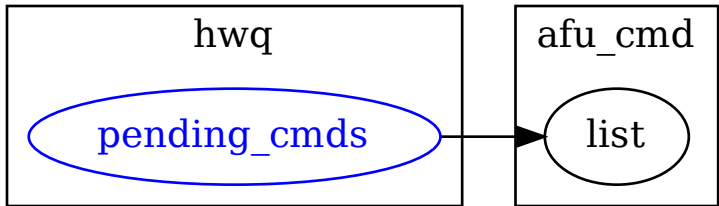
drivetemp_data@drivetemp.c

list

```
graph LR; subgraph "var@drivetemp.c"; devlist(drivetemp_devlist); end; subgraph "drivetemp_data@drivetemp.c"; list((list)); end; devlist --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@drivetemp.c'. Inside this box, the variable 'drivetemp_devlist' is enclosed in a blue oval. On the right, another rectangular box represents the source file 'drivetemp_data@drivetemp.c'. Inside this box, the variable 'list' is enclosed in a black oval. A horizontal arrow points from the right side of the 'drivetemp_devlist' oval to the left side of the 'list' oval, indicating that 'drivetemp_devlist' points to 'list'.





nfp_tun_offloaded_mac@tunnel_conf.c

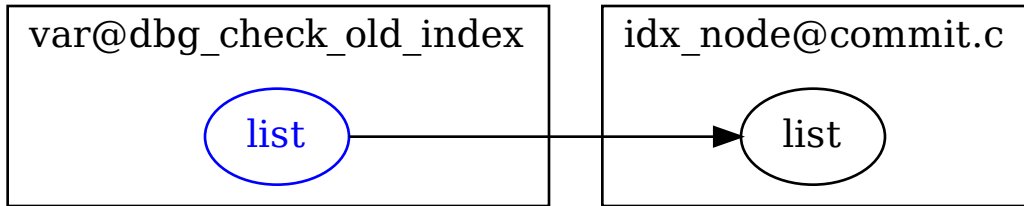
repr_list

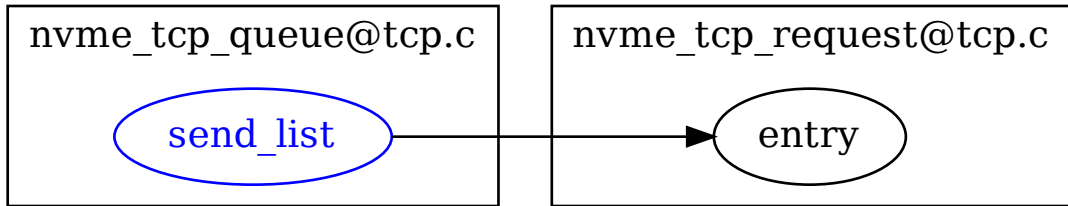
nfp_flower_repr_priv

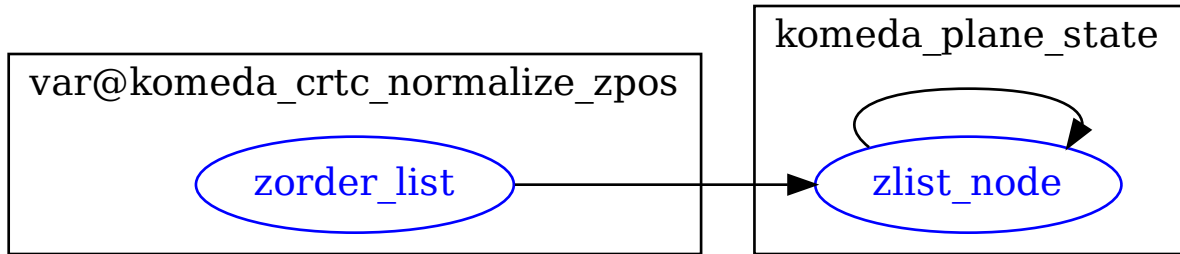
mac_list

```
graph LR; repr_list([repr_list]) --> mac_list([mac_list]);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'nfp_tun_offloaded_mac@tunnel_conf.c' contains an oval labeled 'repr_list' with a blue border. On the right, a box labeled 'nfp_flower_repr_priv' contains an oval labeled 'mac_list' with a black border. A horizontal arrow points from the 'repr_list' oval to the 'mac_list' oval, indicating that 'repr_list' points to 'mac_list'.







var@qcom_icc_set

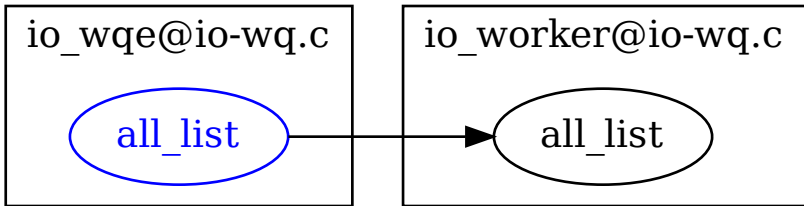
commit_list

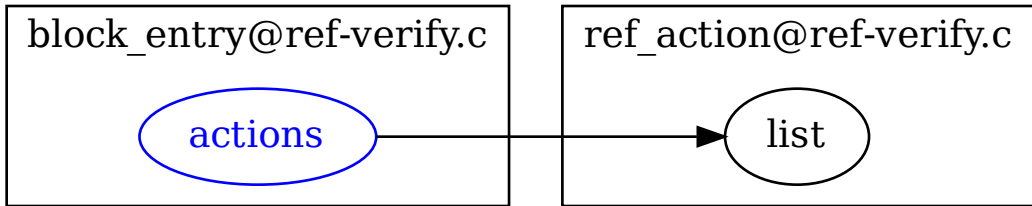
qcom_icc_bcm@sdm845.c

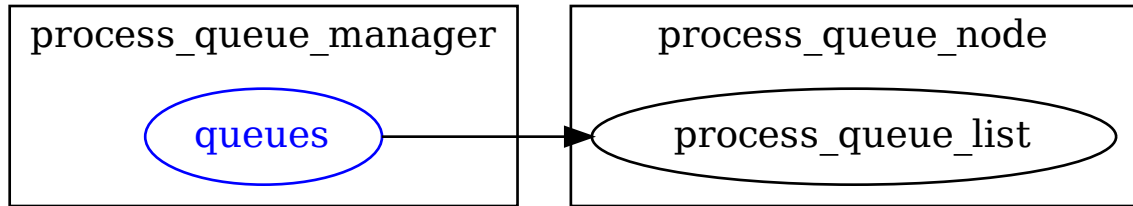
list

```
graph LR; subgraph Box1 [var@qcom_icc_set]; CL([commit_list]); end; subgraph Box2 [qcom_icc_bcm@sdm845.c]; L([list]); end; CL --> L;
```

The diagram illustrates a pointer relationship between two code blocks. The left block, titled 'var@qcom_icc_set', contains a blue oval labeled 'commit_list'. The right block, titled 'qcom_icc_bcm@sdm845.c', contains a black oval labeled 'list'. A horizontal arrow points from the 'commit_list' oval to the 'list' oval, indicating that the 'commit_list' variable points to the 'list' structure.







var@grant-table.c

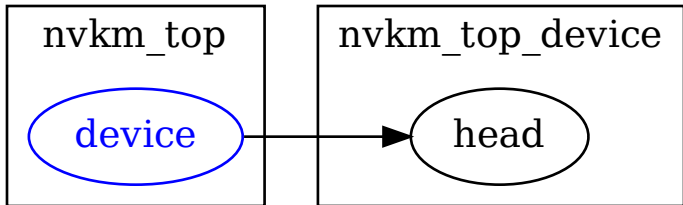
deferred_list

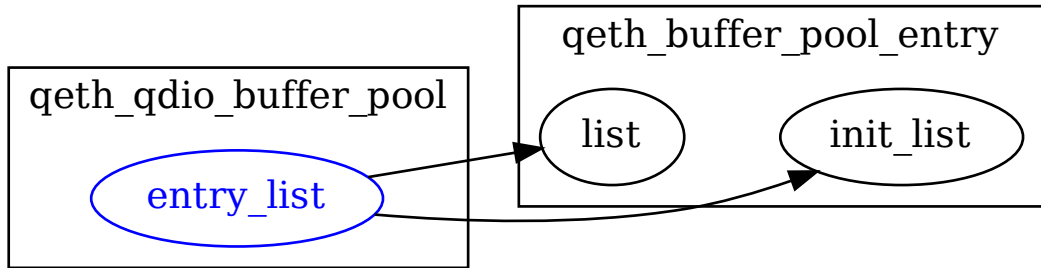
deferred_entry@grant-table.c

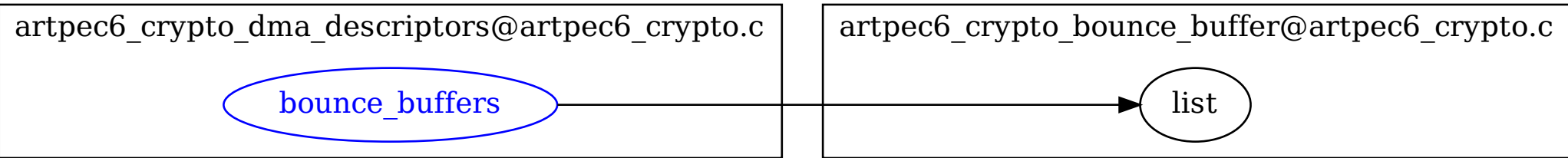
list

```
graph LR; A(var@grant-table.c) --> B(deferred_entry@grant-table.c); subgraph A [var@grant-table.c]; AL(deferred_list); end; subgraph B [deferred_entry@grant-table.c]; BL(list); end; AL --> BL;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@grant-table.c' contains an oval labeled 'deferred_list' (highlighted in blue). On the right, a box labeled 'deferred_entry@grant-table.c' contains an oval labeled 'list'. A horizontal arrow points from the 'deferred_list' oval to the 'list' oval, indicating that 'deferred_list' points to 'list'.







var@nfs4layouts.c

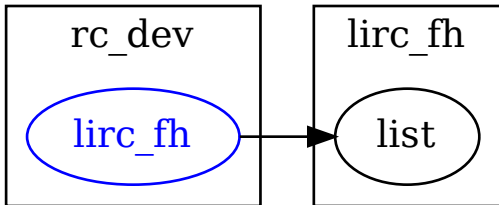
nfsd_devid_hash

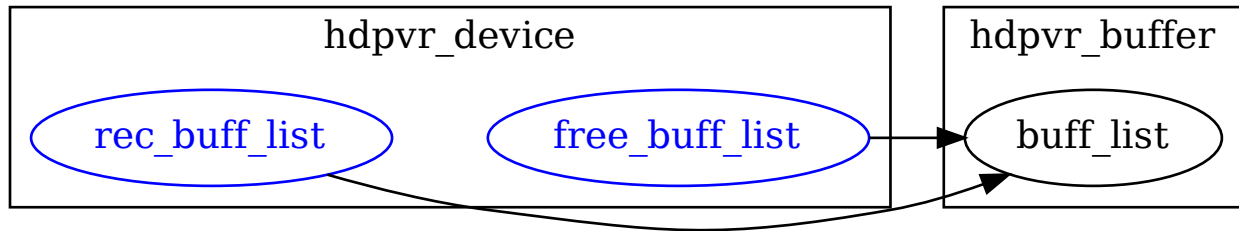
nfsd4_devid_map

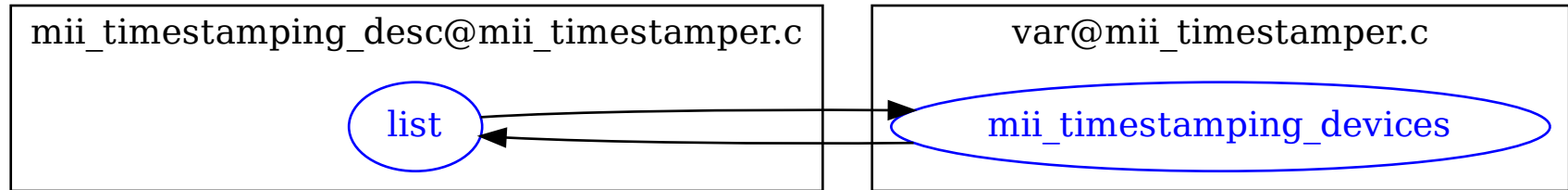
hash

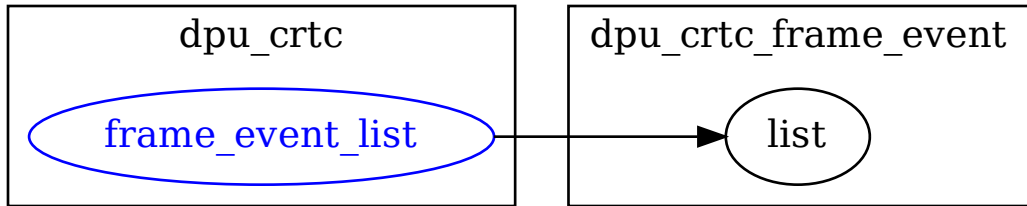
```
graph LR; A[nfsd_devid_hash] --> B[hash];
```

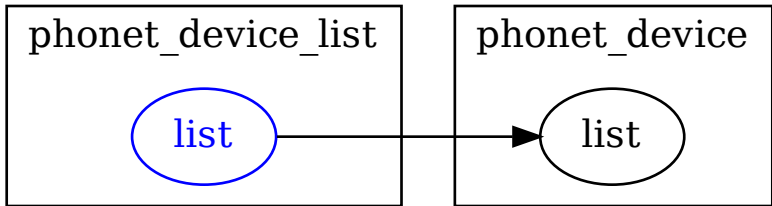
The diagram consists of two rectangular boxes. The left box contains the text 'var@nfs4layouts.c' at the top and 'nfsd_devid_hash' in the center, which is enclosed in a blue oval. The right box contains the text 'nfsd4_devid_map' at the top and 'hash' in the center, which is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.











var@ipmi_devintf.c

reg_list

ipmi_reg_list@ipmi_devintf.c

link

```
graph LR; subgraph File1 [var@ipmi_devintf.c]; reg_list; end; subgraph File2 [ipmi_reg_list@ipmi_devintf.c]; link; end; reg_list --> link;
```

The diagram illustrates a pointer relationship between two variables defined in the same source file, ipmi_devintf.c. On the left, a box labeled 'var@ipmi_devintf.c' contains a variable 'reg_list' highlighted with a blue oval. On the right, a box labeled 'ipmi_reg_list@ipmi_devintf.c' contains a variable 'link' highlighted with a black oval. A horizontal arrow points from the 'reg_list' oval to the 'link' oval, indicating that 'reg_list' points to 'link'.

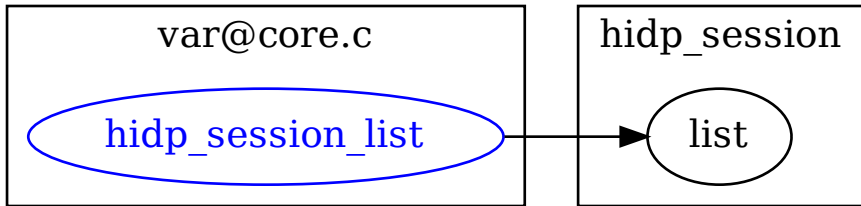
amdtee_shm_context

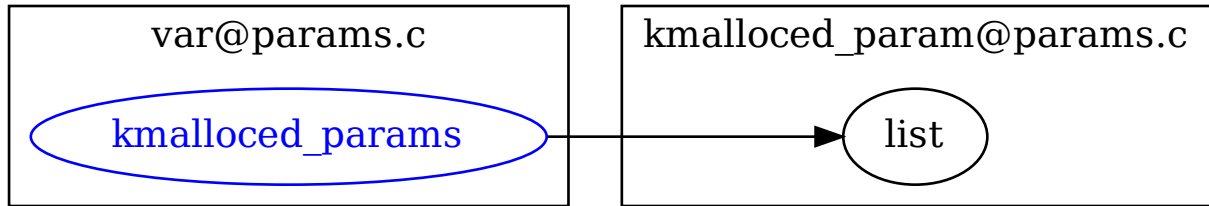
shmdata_list

amdtee_shm_data

shm_node







var@torture.c

shuffle_task_list

shuffle_task@torture.c

st_1

```
graph LR; A[shuffle_task_list] --> B((st_1))
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@torture.c' contains an oval labeled 'shuffle_task_list' with a blue border. On the right, a rectangular box labeled 'shuffle_task@torture.c' contains an oval labeled 'st_1' with a black border. A horizontal arrow points from the right side of the 'shuffle_task_list' oval to the left side of the 'st_1' oval, indicating that 'shuffle_task_list' points to 'st_1'.

var@domain.c

of_genpd_providers

of_genpd_provider@domain.c

link



```
graph LR; A[of_genpd_providers] --> B((link)); B --- C[of_genpd_provider@domain.c];
```

The diagram illustrates a relationship between two components. On the left, a rectangular box labeled 'var@domain.c' contains an oval labeled 'of_genpd_providers' with a blue border. On the right, a rectangular box labeled 'of_genpd_provider@domain.c' contains an oval labeled 'link' with a black border. A horizontal arrow points from the 'of_genpd_providers' oval to the 'link' oval, indicating a connection or pointer. The 'link' oval is positioned within the 'of_genpd_provider@domain.c' box.

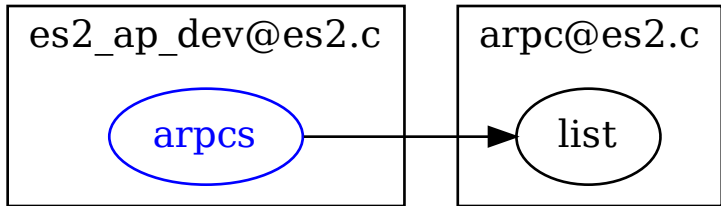
var@uprobe_mmap

tmp_list

uprobe@uprobes.c

pending_list





res_qp@resource_tracker.c

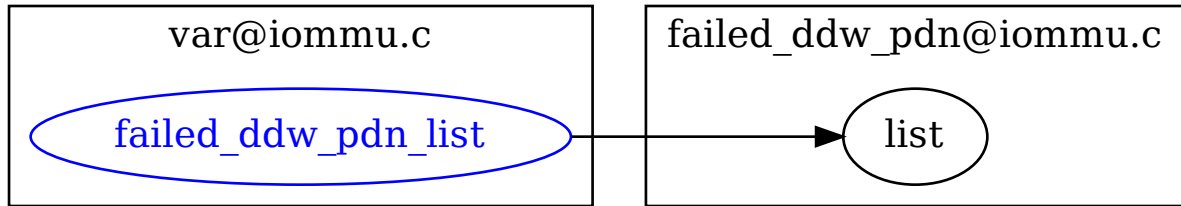
mcg_list

res_gid@resource_tracker.c

list

```
graph LR; A[res_qp@resource_tracker.c] --- B((mcg_list)); B --> C((list)); C --- D[res_gid@resource_tracker.c];
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'res_qp@resource_tracker.c' contains an oval labeled 'mcg_list'. On the right, a box labeled 'res_gid@resource_tracker.c' contains an oval labeled 'list'. A horizontal arrow points from the 'mcg_list' oval to the 'list' oval, indicating that 'mcg_list' points to 'list'.



var@cifs_push_posix_locks

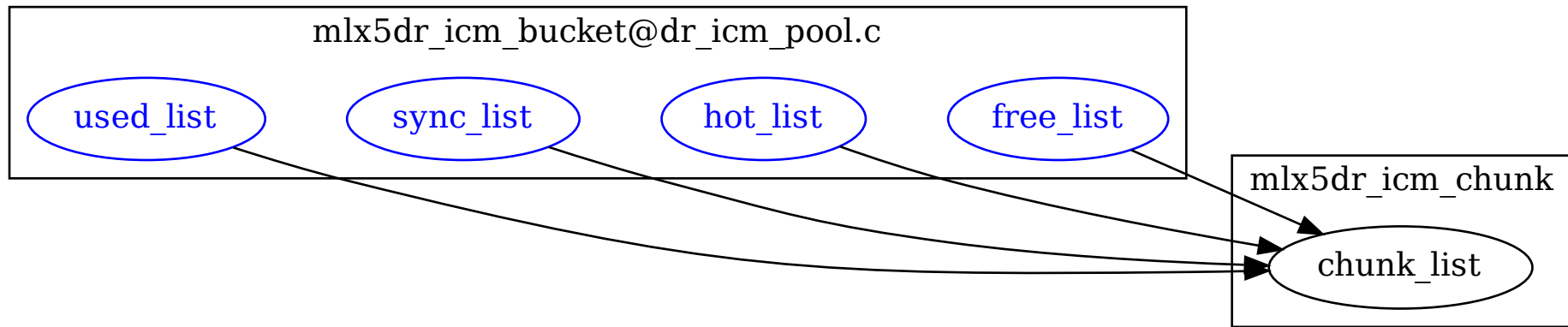
locks_to_send

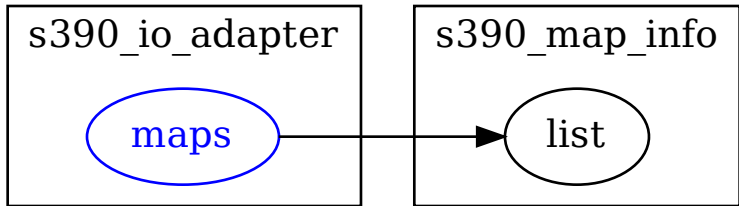
lock_to_push@file.c

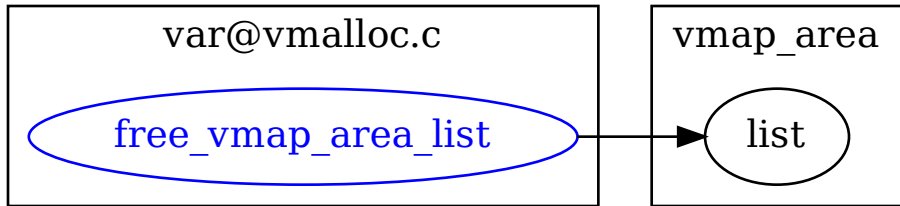
llist

```
graph LR; A[locks_to_send] --> B[llist]
```

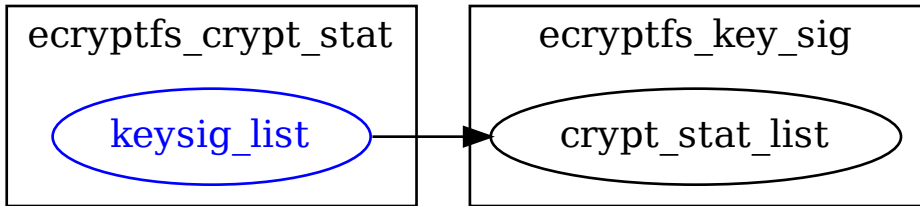
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@cifs_push_posix_locks'. Inside this box, the text 'locks_to_send' is enclosed in a blue oval. On the right, another rectangular box contains the text 'lock_to_push@file.c'. Inside this box, the text 'llist' is enclosed in a black oval. A horizontal arrow points from the right side of the 'locks_to_send' oval to the left side of the 'llist' oval, indicating that 'locks_to_send' points to 'llist'.

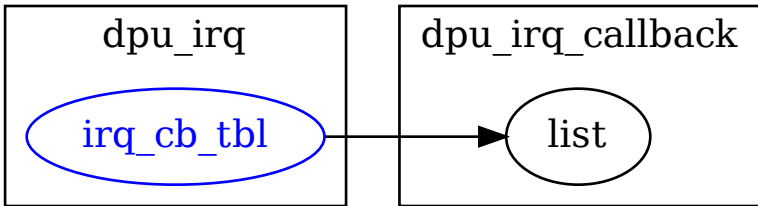


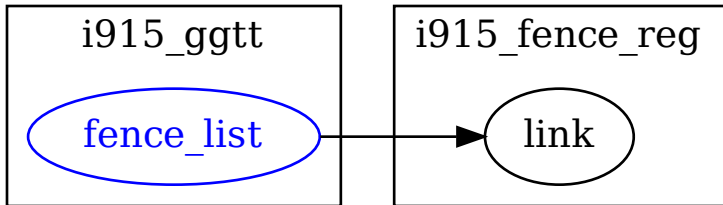


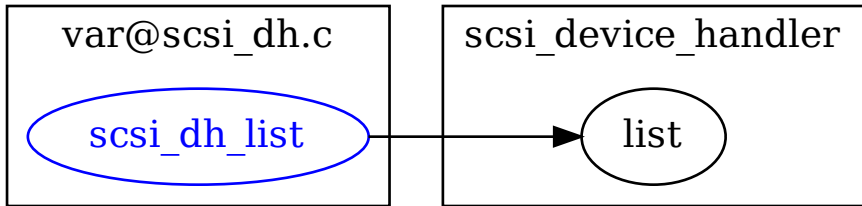












var@cache-uniphier.c

uniphier_cache_list

uniphier_cache_data@cache-uniphier.c

list

```
graph LR; A[uniphier_cache_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@cache-uniphier.c' at the top and 'uniphier_cache_list' below it. The 'uniphier_cache_list' text is enclosed in a blue oval. On the right, another rectangular box contains the text 'uniphier_cache_data@cache-uniphier.c' at the top. Inside this box, there is a circle containing the text 'list'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the circle in the second box, indicating that 'uniphier_cache_list' points to 'list'.

var@start_delalloc_inodes

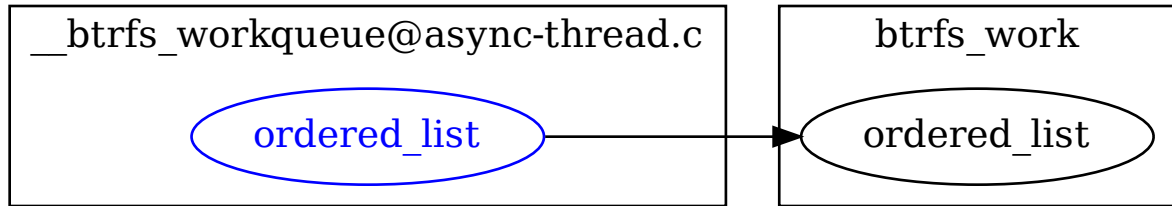
works

btrfs_delalloc_work@inode.c

list

```
graph LR; subgraph Box1 [var@start_delalloc_inodes]; works((works)); end; subgraph Box2 [btrfs_delalloc_work@inode.c]; list((list)); end; works --> list;
```

The diagram consists of two rectangular boxes. The left box is titled 'var@start_delalloc_inodes' and contains an oval labeled 'works' in blue text. The right box is titled 'btrfs_delalloc_work@inode.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'works' oval in the first box to the 'list' oval in the second box.



var@ccp-crypto-main.c

skcipher_algs

ccp_crypto_skcipher_alg

entry

```
graph LR; A[skcipher_algs] --> B(entry);
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@ccp-crypto-main.c' at the top and 'skcipher_algs' in a blue oval below it. The right box contains the text 'ccp_crypto_skcipher_alg' at the top and 'entry' in a black oval below it. A horizontal arrow points from the right side of the 'skcipher_algs' oval to the left side of the 'entry' oval.

atmel_tdes_drv@atmel-tdes.c

dev_list

atmel_tdes_dev@atmel-tdes.c

list

```
graph LR; subgraph "atmel_tdes_drv@atmel-tdes.c"; dev_list; end; subgraph "atmel_tdes_dev@atmel-tdes.c"; list; end; dev_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'atmel_tdes_drv@atmel-tdes.c' contains a variable 'dev_list' which is highlighted with a blue oval. On the right, a box labeled 'atmel_tdes_dev@atmel-tdes.c' contains a variable 'list' which is enclosed in a black oval. A horizontal arrow points from the 'dev_list' oval to the 'list' oval, indicating that 'dev_list' points to 'list'.

var@xen-scsiback.c

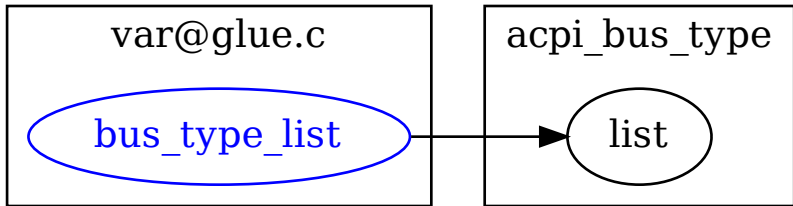
scsiback_list

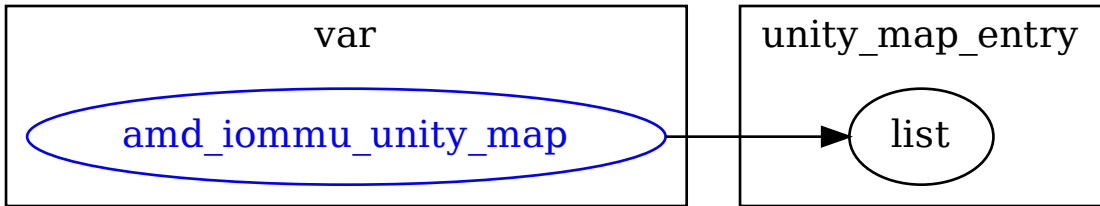
scsiback_tpg@xen-scsiback.c

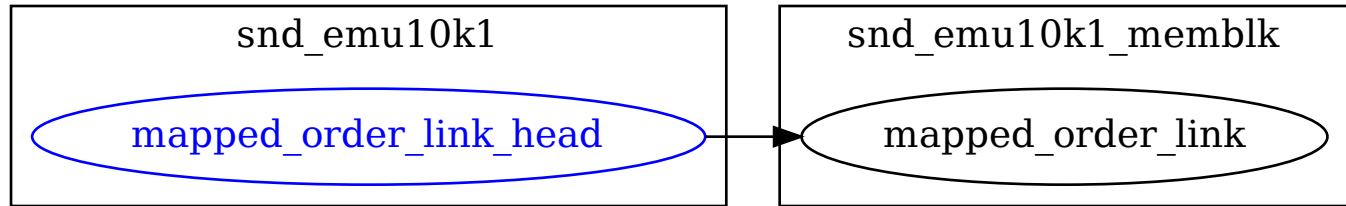
tv_tpg_list

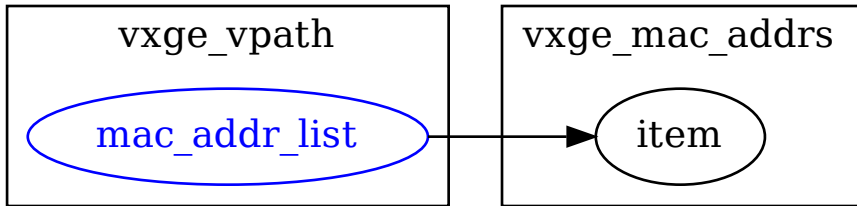
```
graph LR; A[scsiback_list] --> B(tv_tpg_list)
```

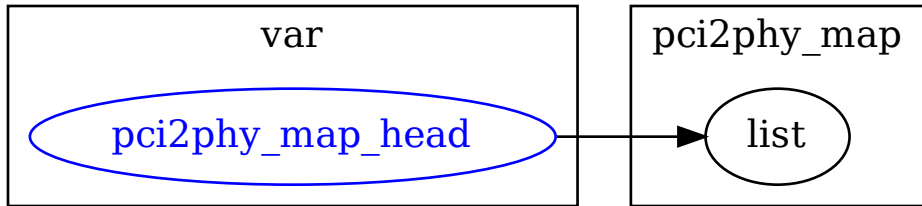
The diagram consists of two rectangular boxes. The left box contains the text 'var@xen-scsiback.c' at the top and an oval containing 'scsiback_list' in blue text below it. The right box contains the text 'scsiback_tpg@xen-scsiback.c' at the top and an oval containing 'tv_tpg_list' below it. A horizontal arrow points from the right side of the 'scsiback_list' oval to the left side of the 'tv_tpg_list' oval.

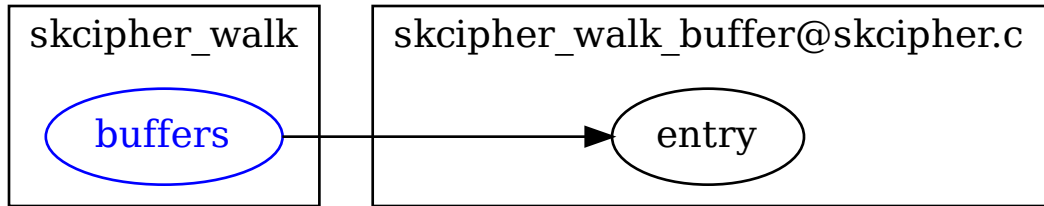














var@dm-cache-metadata.c

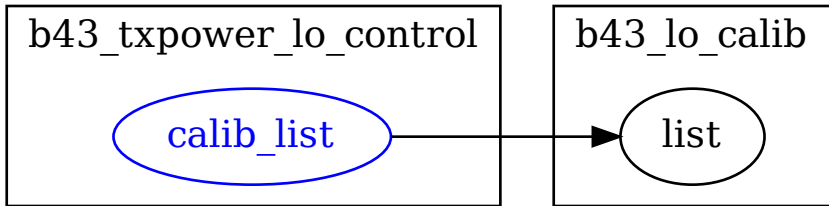
table

dm_cache_metadata@dm-cache-metadata.c

list

```
graph LR; subgraph Box1 [var@dm-cache-metadata.c]; table((table)); end; subgraph Box2 [dm_cache_metadata@dm-cache-metadata.c]; list((list)); end; table --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@dm-cache-metadata.c' and contains a blue oval labeled 'table'. The right box is labeled 'dm_cache_metadata@dm-cache-metadata.c' and contains a black oval labeled 'list'. A horizontal arrow points from the 'table' oval in the first box to the 'list' oval in the second box.



var@core.c

regulator_supply_alias_list

regulator_supply_alias@core.c

list

```
graph LR; A[regulator_supply_alias_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@core.c`. Inside this box, the variable `regulator_supply_alias_list` is highlighted with a blue oval. On the right, another rectangular box represents the source file `regulator_supply_alias@core.c`. Inside this box, there is a variable `list` enclosed in a black oval. A horizontal arrow points from the blue oval in the first box to the black oval in the second box, indicating that `regulator_supply_alias_list` points to `list`.

gk20a_instmem@gk20a.c

vaddr_lru

gk20a_instobj_iommu@gk20a.c

vaddr_node

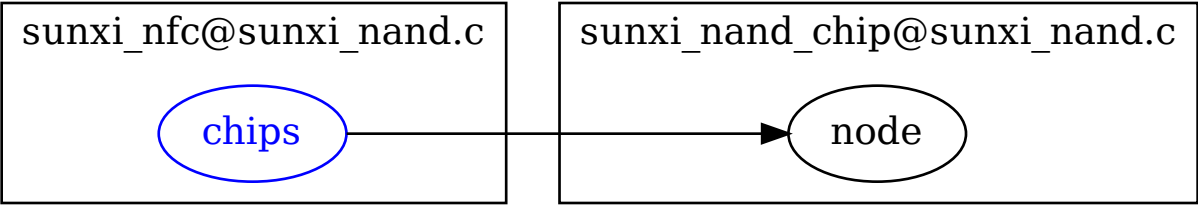


sunxi_nfc@sunxi_nand.c

chips

sunxi_nand_chip@sunxi_nand.c

node



```
graph LR; subgraph Box1 [ ]; direction TB; A[sunxi_nfc@sunxi_nand.c]; B(chips); end; subgraph Box2 [ ]; direction TB; C[sunxi_nand_chip@sunxi_nand.c]; D(node); end; B --> D;
```


efx_mcdi_iface

async_list

efx_mcdi_async_param@mcdi.c

list

```
graph LR; subgraph Box1 [efx_mcdi_iface]; A(async_list); end; subgraph Box2 [efx_mcdi_async_param@mcdi.c]; B(list); end; A --> B;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'efx_mcdi_iface' contains an oval labeled 'async_list' with a blue border. On the right, a rectangular box labeled 'efx_mcdi_async_param@mcdi.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'async_list' oval to the left side of the 'list' oval, indicating that 'async_list' points to 'list'.

connection@lowcomms.c

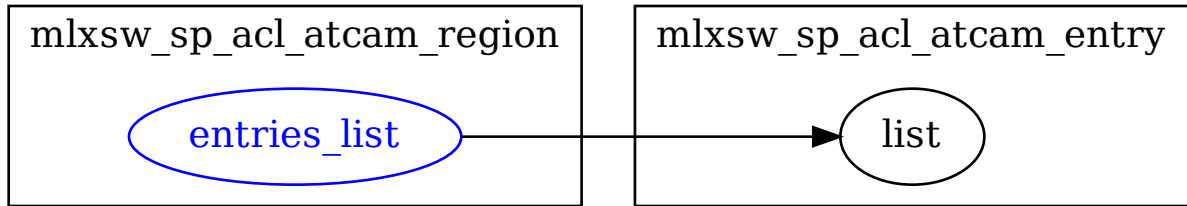
writequeue

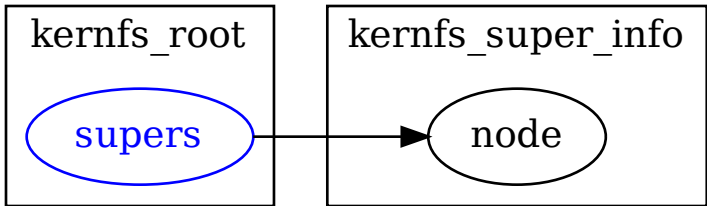
writequeue_entry@lowcomms.c

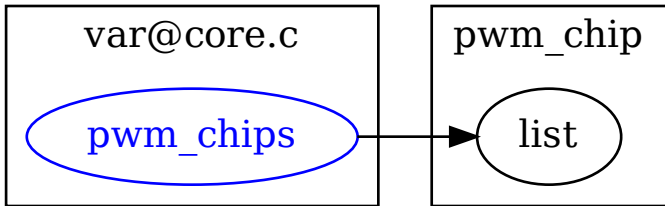
list

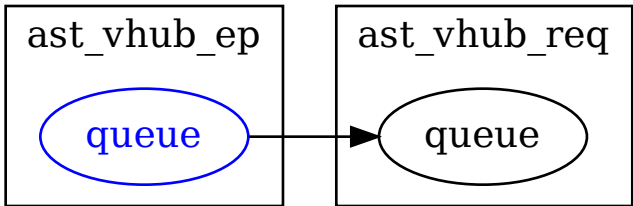
```
graph LR; subgraph "connection@lowcomms.c"; WQ(writequeue); end; subgraph "writequeue_entry@lowcomms.c"; L((list)); end; WQ --> L;
```

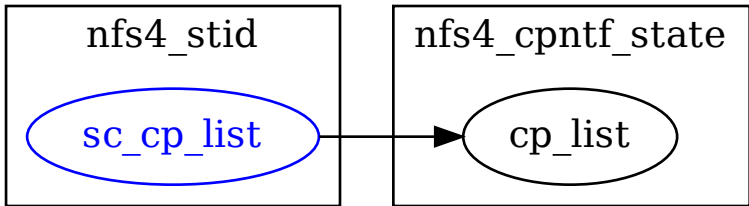
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'connection@lowcomms.c' contains an oval labeled 'writequeue'. On the right, a rectangular box labeled 'writequeue_entry@lowcomms.c' contains an oval labeled 'list'. A horizontal arrow points from the 'writequeue' oval to the 'list' oval, indicating that the 'writequeue' variable points to the 'list' structure.











var@irq-gic-v2m.c

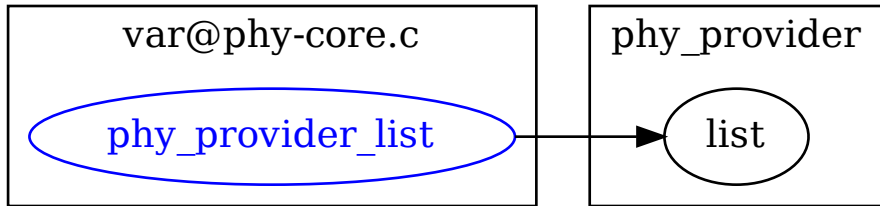
v2m_nodes

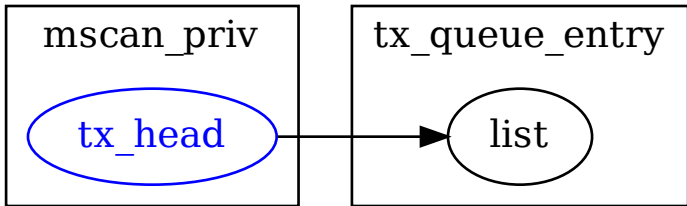
v2m_data@irq-gic-v2m.c

entry

```
graph LR; subgraph var ["var@irq-gic-v2m.c"]; v2m_nodes; end; subgraph v2m_data ["v2m_data@irq-gic-v2m.c"]; entry; end; v2m_nodes --> entry;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@irq-gic-v2m.c' contains an oval labeled 'v2m_nodes' in blue text. On the right, a box labeled 'v2m_data@irq-gic-v2m.c' contains an oval labeled 'entry'. A horizontal arrow points from the 'v2m_nodes' oval to the 'entry' oval, indicating that 'v2m_nodes' points to 'entry'.





var@dma-mapping.c

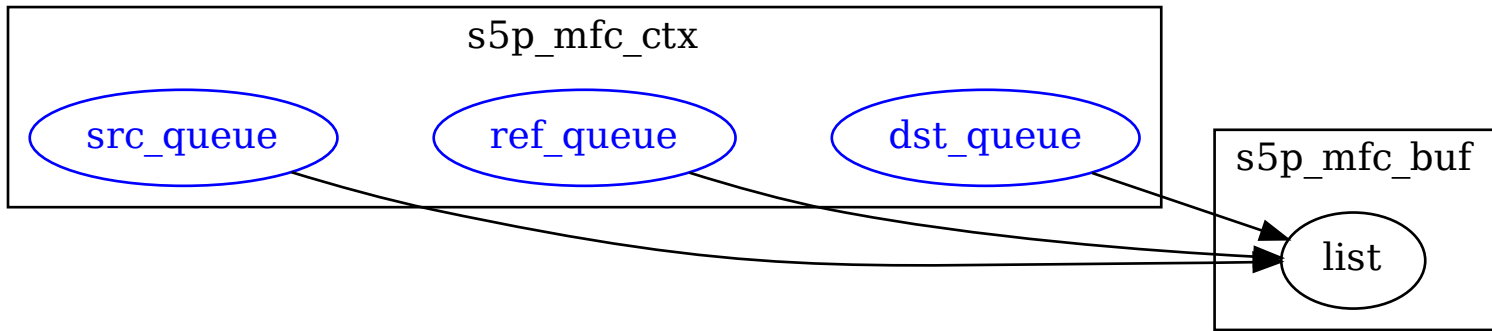
arm_dma_bufs

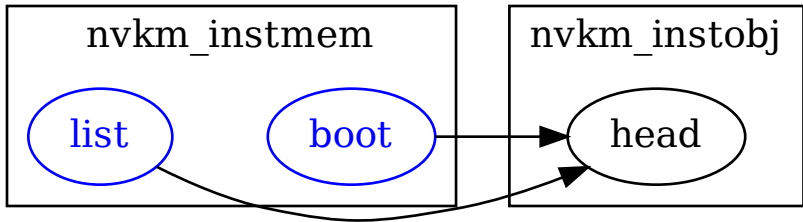
arm_dma_buffer@dma-mapping.c

list

```
graph LR; subgraph File1 [var@dma-mapping.c]; A([arm_dma_bufs]); end; subgraph File2 [arm_dma_buffer@dma-mapping.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file `var@dma-mapping.c`, containing a variable `arm_dma_bufs` highlighted by a blue oval. On the right, a box represents the source file `arm_dma_buffer@dma-mapping.c`, containing a variable `list` enclosed in a black oval. A horizontal arrow points from the `arm_dma_bufs` oval to the `list` oval, indicating that `arm_dma_bufs` points to `list`.





var@industrialio-trigger.c

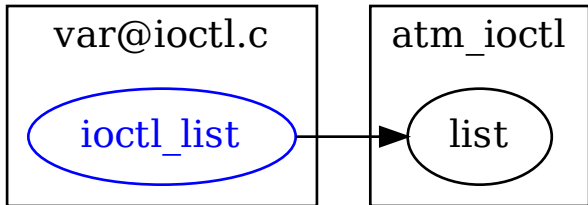
iio_trigger_list

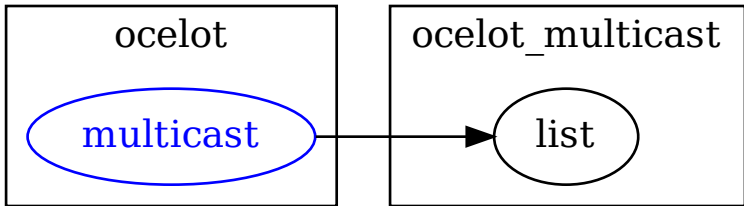
iio_trigger

list

```
graph LR; A[iio_trigger_list] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@industrialio-trigger.c' contains an oval labeled 'iio_trigger_list' in blue text. On the right, a box labeled 'iio_trigger' contains an oval labeled 'list'. A horizontal arrow points from the 'iio_trigger_list' oval to the 'list' oval, indicating that 'iio_trigger_list' points to 'list'.





var@soc_camera.c

hosts

soc_camera_host

list

```
graph LR; subgraph Box1 [var@soc_camera.c]; H([hosts]); end; subgraph Box2 [soc_camera_host]; L([list]); end; H --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@soc_camera.c' and contains a blue oval labeled 'hosts'. The right box is labeled 'soc_camera_host' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'hosts' oval to the left side of the 'list' oval, indicating a pointer or reference relationship.

var@ppp_generic.c

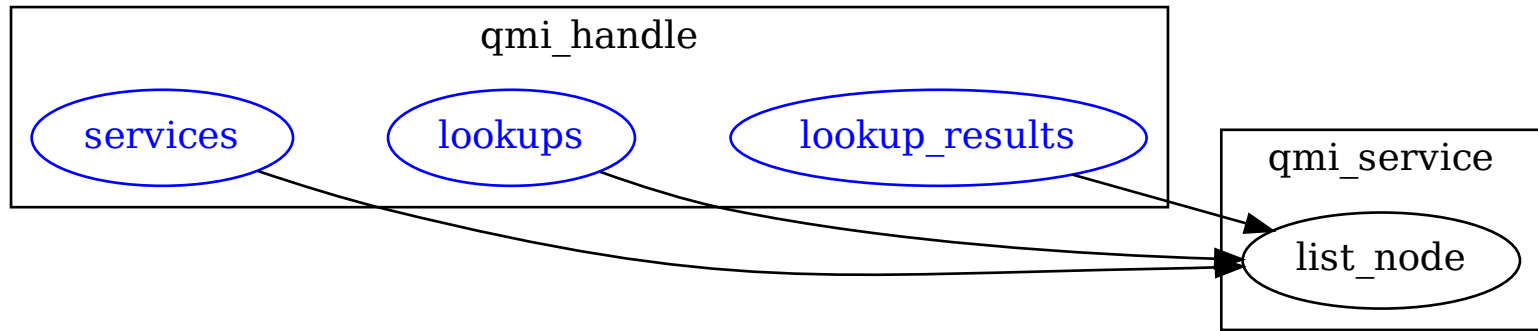
compressor_list

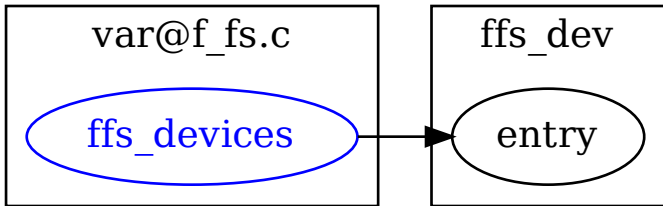
compressor_entry@ppp_generic.c

list

```
graph LR; subgraph File1 [var@ppp_generic.c]; CL([compressor_list]); end; subgraph File2 [compressor_entry@ppp_generic.c]; L((list)); end; CL --> L;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@ppp_generic.c'. Inside this box, the variable 'compressor_list' is enclosed in a blue oval. On the right, another rectangular box represents the source file 'compressor_entry@ppp_generic.c'. Inside this box, the variable 'list' is enclosed in a black oval. A horizontal arrow points from the blue oval 'compressor_list' to the black oval 'list', indicating that 'compressor_list' points to 'list'.





var@gpio-mxc.c

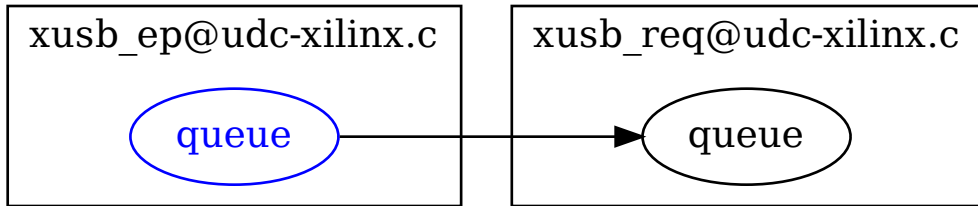
mxc_gpio_ports

mxc_gpio_port@gpio-mxc.c

node

```
graph LR; subgraph Box1 [var@gpio-mxc.c]; direction TB; A([mxc_gpio_ports]); end; subgraph Box2 [mxc_gpio_port@gpio-mxc.c]; direction TB; B([node]); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@gpio-mxc.c' contains an oval labeled 'mxc_gpio_ports' which is highlighted with a blue border. On the right, another rectangular box labeled 'mxc_gpio_port@gpio-mxc.c' contains an oval labeled 'node'. A horizontal arrow points from the 'mxc_gpio_ports' oval to the 'node' oval, indicating that the first variable points to the second.



var@smsgiucv.c

smsg_list

smsg_callback@smsgiucv.c

list

```
graph LR; subgraph Box1 [var@smsgiucv.c]; smsg_list((smsg_list)); end; subgraph Box2 [smsg_callback@smsgiucv.c]; list((list)); end; smsg_list --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@smsgiucv.c' and contains an oval labeled 'smsg_list' in blue text. The right box is labeled 'smsg_callback@smsgiucv.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the right side of the 'smsg_list' oval to the left side of the 'list' oval.

var@thinkpad_acpi.c

tpacpi_all_drivers

ibm_struct@thinkpad_acpi.c

all_drivers

```
graph LR; A[tpacpi_all_drivers] --> B[all_drivers]
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@thinkpad_acpi.c' and contains an oval labeled 'tpacpi_all_drivers' with a blue border. The right box is labeled 'ibm_struct@thinkpad_acpi.c' and contains an oval labeled 'all_drivers' with a black border. A horizontal arrow points from the right side of the 'tpacpi_all_drivers' oval to the left side of the 'all_drivers' oval.

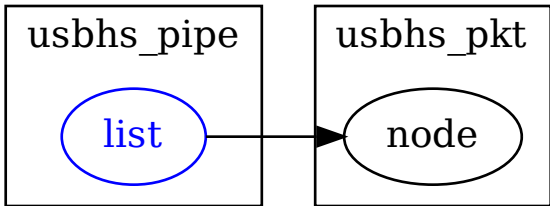
var@dasd_devmap.c

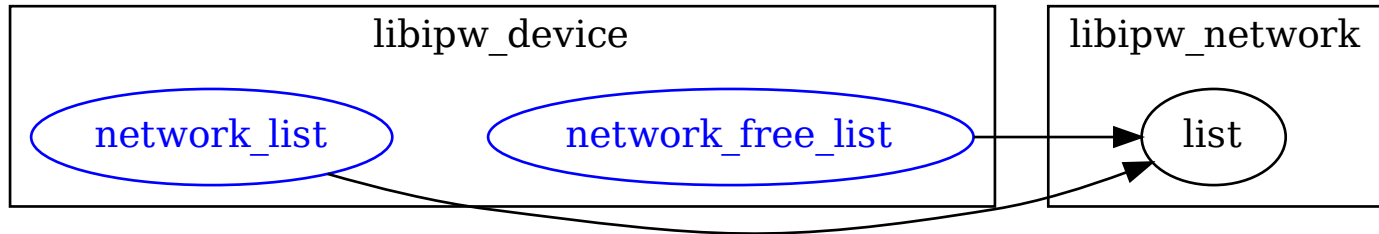
dasd_hashlists

dasd_devmap@dasd_devmap.c

list







cfcnfg@cfcnfg.c

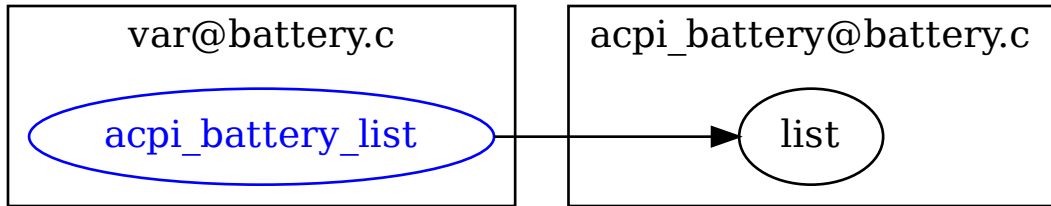
phys

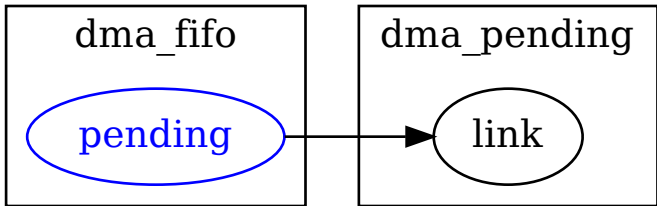
cfcnfg_phyinfo@cfcnfg.c

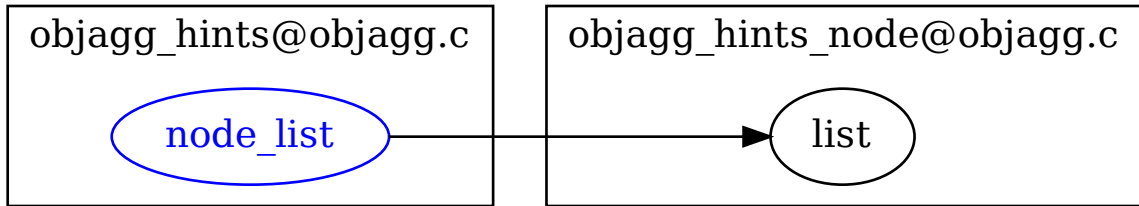
node

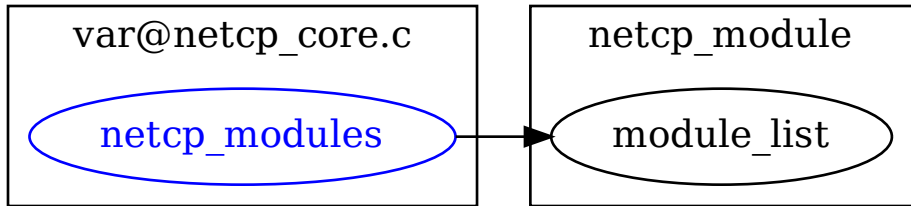
```
graph LR; subgraph "cfcnfg@cfcnfg.c"; phys((phys)); end; subgraph "cfcnfg_phyinfo@cfcnfg.c"; node((node)); end; phys --> node;
```

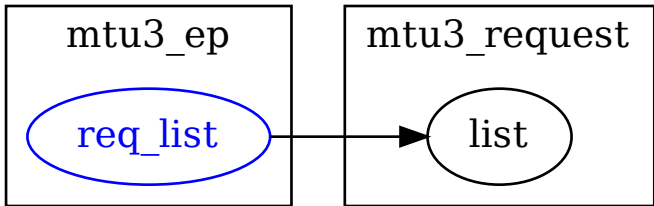
The diagram illustrates a mapping between two configuration domains. On the left, a box labeled 'cfcnfg@cfcnfg.c' contains an oval labeled 'phys'. On the right, a box labeled 'cfcnfg_phyinfo@cfcnfg.c' contains an oval labeled 'node'. A horizontal arrow points from the 'phys' oval to the 'node' oval, indicating a relationship or mapping between the two.











var@sram.c

sram_bank_list

sram_bank_info@sram.c

node

```
graph LR; subgraph var_sram_c [var@sram.c]; sram_bank_list; end; subgraph sram_bank_info_sram_c [sram_bank_info@sram.c]; node; end; sram_bank_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@sram.c' contains an oval labeled 'sram_bank_list'. On the right, a box labeled 'sram_bank_info@sram.c' contains an oval labeled 'node'. A horizontal arrow points from the 'sram_bank_list' oval to the 'node' oval, indicating that 'sram_bank_list' points to 'node'.

cpu_stopper@stop_machine.c

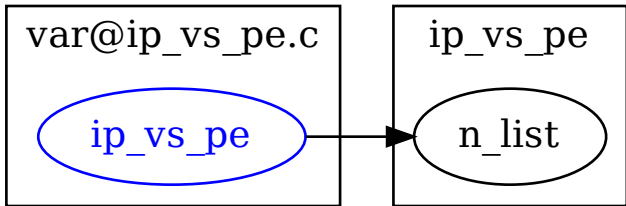
works

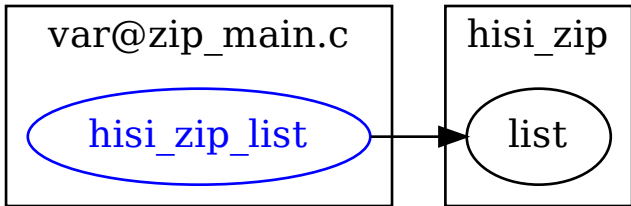
cpu_stop_work

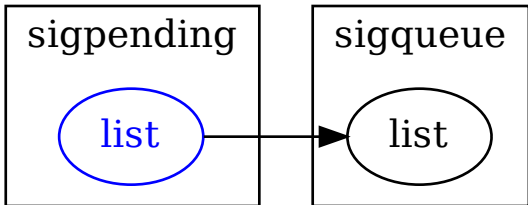
list

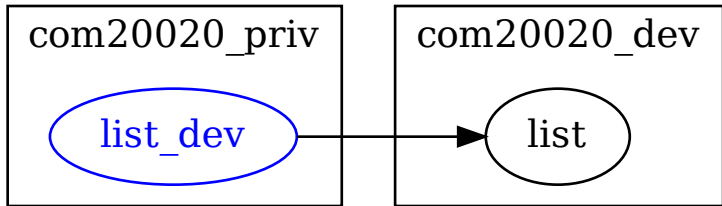
```
graph LR; subgraph Box1 [cpu_stopper@stop_machine.c]; works((works)); end; subgraph Box2 [cpu_stop_work]; list((list)); end; works --> list;
```

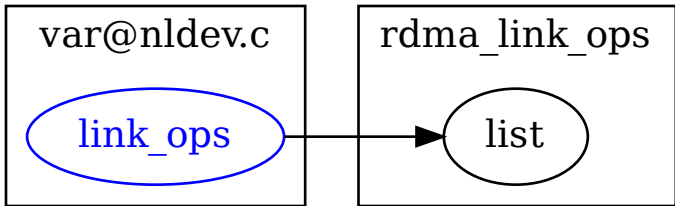
The diagram consists of two rectangular boxes. The left box is labeled 'cpu_stopper@stop_machine.c' and contains an oval labeled 'works' with a blue border. The right box is labeled 'cpu_stop_work' and contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'works' oval to the left side of the 'list' oval, indicating a pointer relationship.

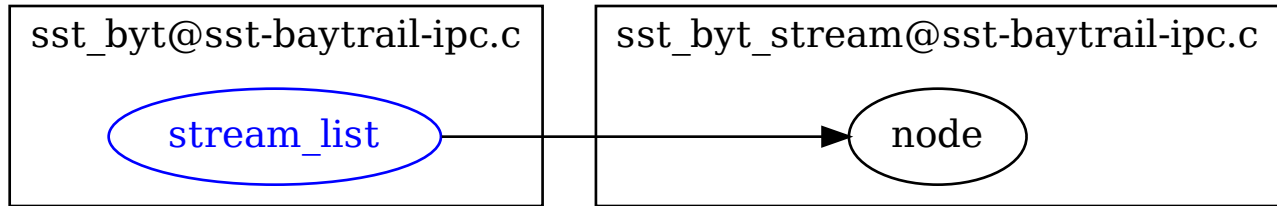


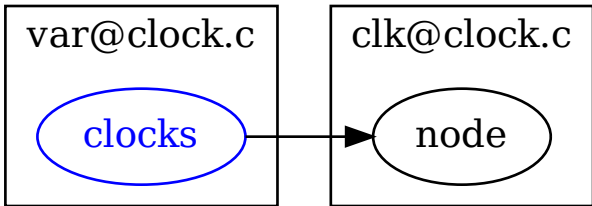












var@kpc_dma_driver.c

kpc_dma_list

kpc_dma_device

list

```
graph LR; A[kpc_dma_list] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@kpc_dma_driver.c' contains a blue oval labeled 'kpc_dma_list'. On the right, a box labeled 'kpc_dma_device' contains a black oval labeled 'list'. A horizontal arrow points from the 'kpc_dma_list' oval to the 'list' oval, indicating that 'kpc_dma_list' points to 'list'.

var@hid-quirks.c

dquirks_list

quirks_list_struct@hid-quirks.c

node

```
graph LR; subgraph Box1 [var@hid-quirks.c]; dquirks_list; end; subgraph Box2 [quirks_list_struct@hid-quirks.c]; node; end; dquirks_list --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@hid-quirks.c' and contains an oval labeled 'dquirks_list' in blue text. The right box is labeled 'quirks_list_struct@hid-quirks.c' and contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'dquirks_list' oval to the left side of the 'node' oval.

gsm_mux@n_gsm.c

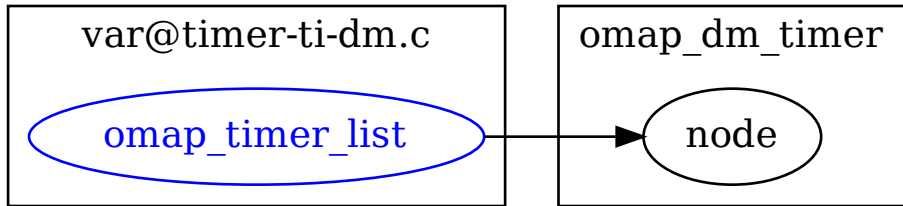
tx_list

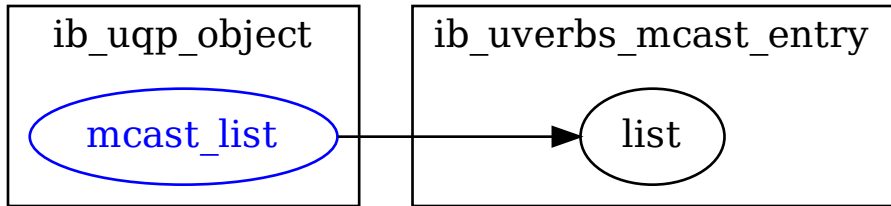
gsm_msg@n_gsm.c

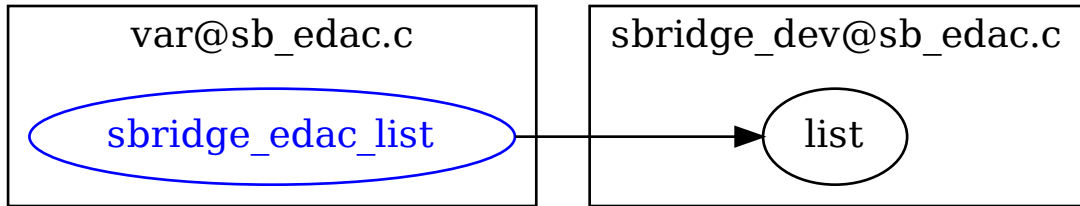
list

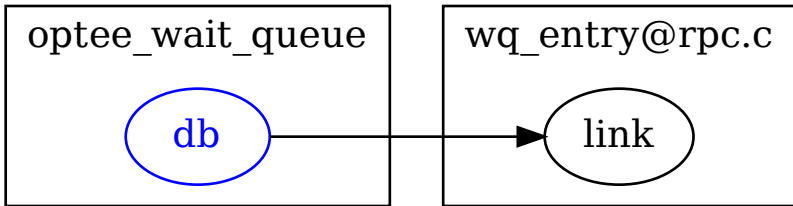
```
graph LR; subgraph "gsm_mux@n_gsm.c"; tx_list; end; subgraph "gsm_msg@n_gsm.c"; list; end; tx_list --> list;
```

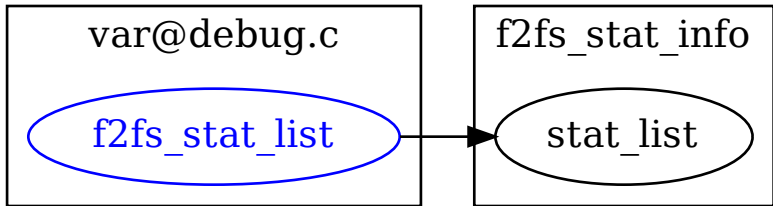
The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'gsm_mux@n_gsm.c' contains an oval labeled 'tx_list'. On the right, a box labeled 'gsm_msg@n_gsm.c' contains an oval labeled 'list'. A horizontal arrow points from the 'tx_list' oval to the 'list' oval, indicating that 'tx_list' points to 'list'.

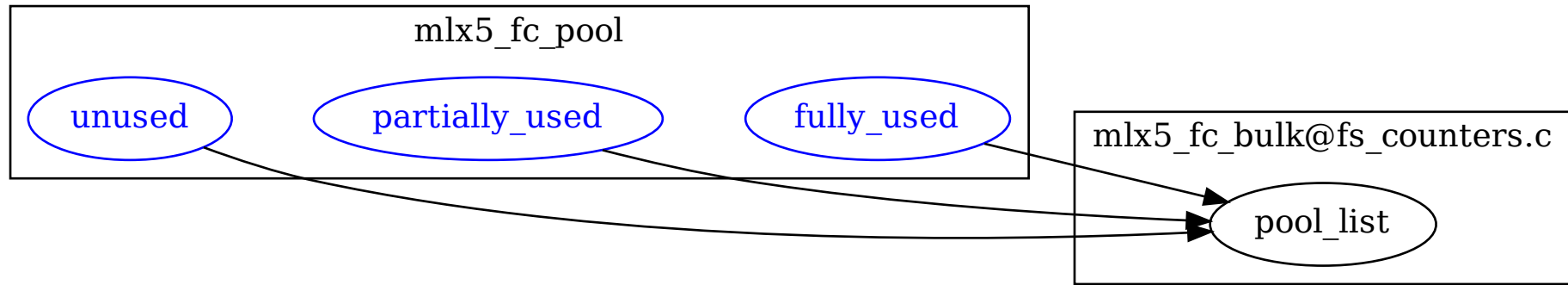


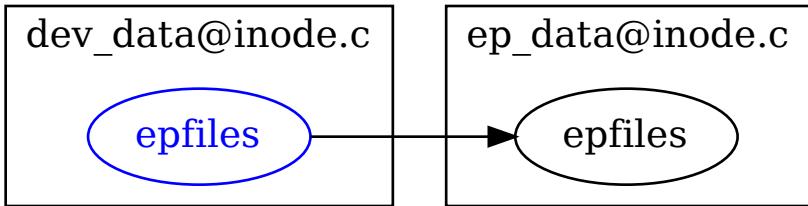


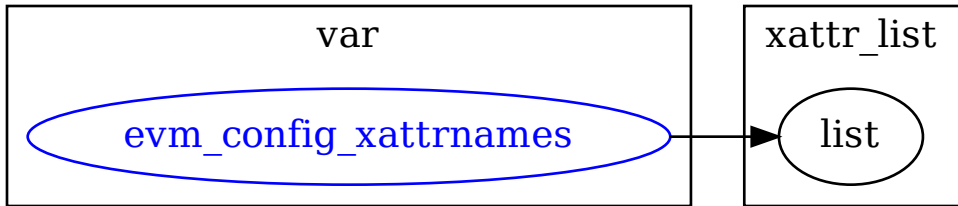


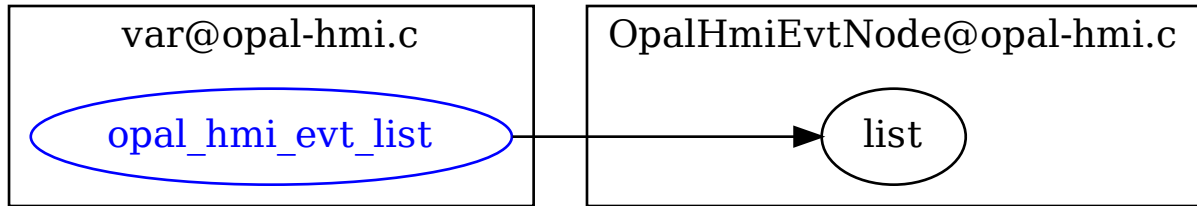


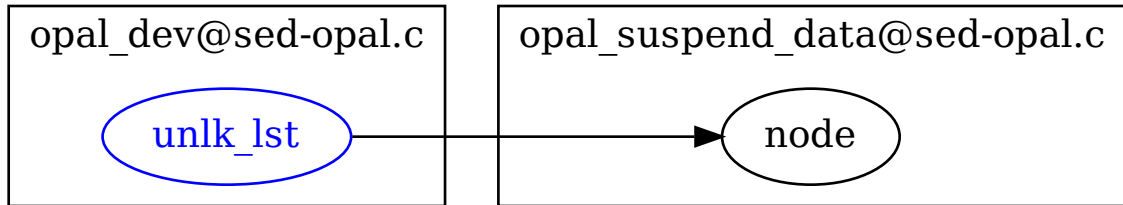












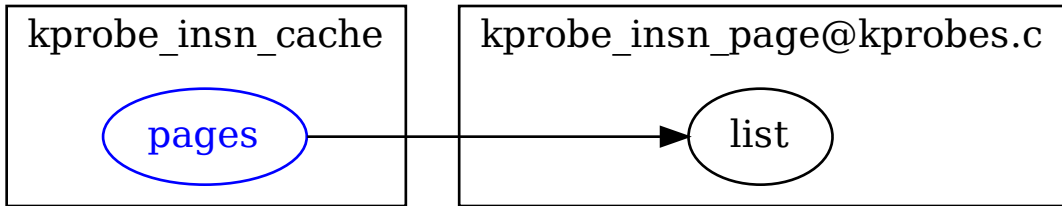
vexpress_syscfg@vexpress-syscfg.c

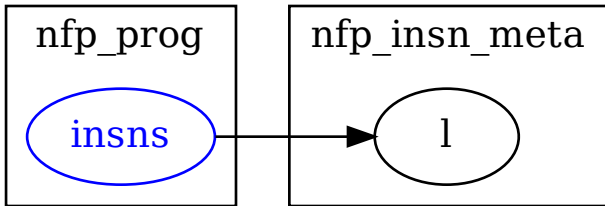
funcs

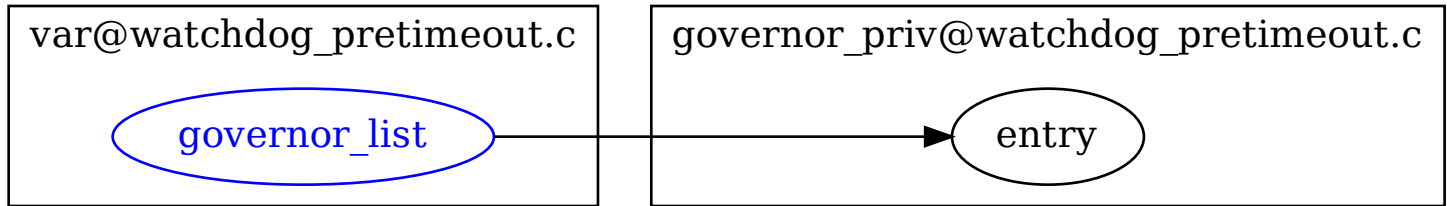
vexpress_syscfg_func@vexpress-syscfg.c

list









gvt_sched_data@sched_policy.c

lru_runq_head

vgpu_sched_data@sched_policy.c

lru_list



ring_buffer_per_cpu@ring_buffer.c

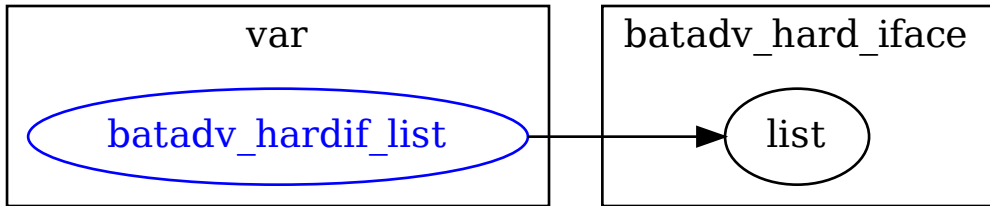
new_pages

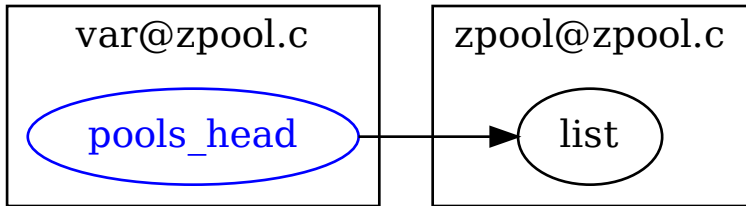
buffer_page@ring_buffer.c

list

```
graph LR; subgraph A [ring_buffer_per_cpu@ring_buffer.c]; NP([new_pages]); end; subgraph B [buffer_page@ring_buffer.c]; L([list]); end; NP --> L;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the scope 'ring_buffer_per_cpu@ring_buffer.c', containing an oval labeled 'new_pages' in blue text. On the right, another rectangular box represents the scope 'buffer_page@ring_buffer.c', containing an oval labeled 'list'. A horizontal arrow points from the 'new_pages' oval to the 'list' oval, indicating that 'new_pages' points to 'list'.





qcom_smd_edge@qcom_smd.c

channels

qcom_smd_channel@qcom_smd.c

list

```
graph LR; A[qcom_smd_edge@qcom_smd.c] --> B[qcom_smd_channel@qcom_smd.c]; subgraph A; C([channels]); end; subgraph B; D([list]); end; C --> D;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'qcom_smd_edge@qcom_smd.c' contains a blue oval labeled 'channels'. On the right, a box labeled 'qcom_smd_channel@qcom_smd.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'channels' oval to the 'list' oval, indicating that the 'channels' variable points to the 'list' structure.

var@amdgpu_pmu.c

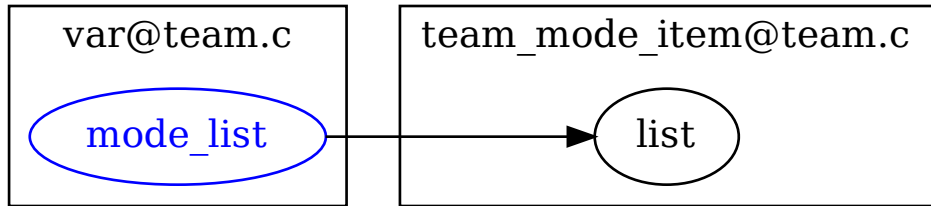
amdgpu_pmu_list

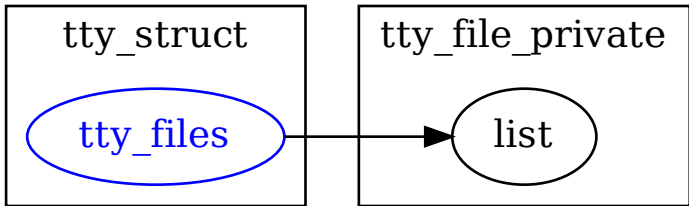
amdgpu_pmu_entry@amdgpu_pmu.c

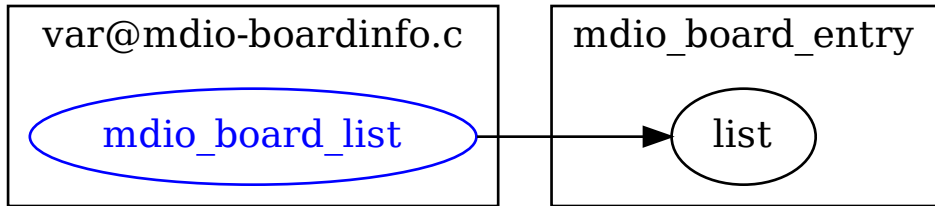
entry

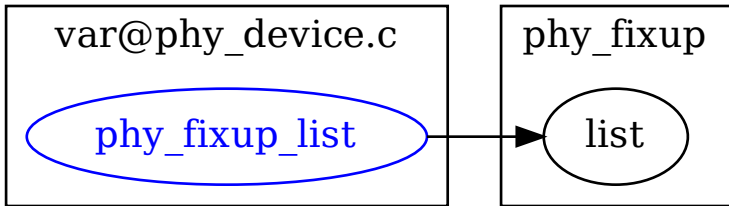
```
graph LR; A[amdgpu_pmu_list] --> B[entry]
```

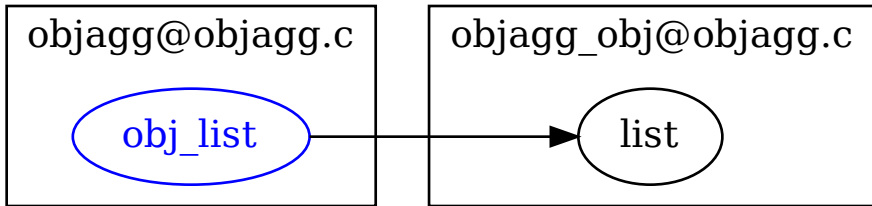
The diagram consists of two rectangular boxes. The left box contains the text 'var@amdgpu_pmu.c' at the top and 'amdgpu_pmu_list' in a blue oval below it. The right box contains the text 'amdgpu_pmu_entry@amdgpu_pmu.c' at the top and 'entry' in a black oval below it. A horizontal arrow points from the right side of the 'amdgpu_pmu_list' oval to the left side of the 'entry' oval.











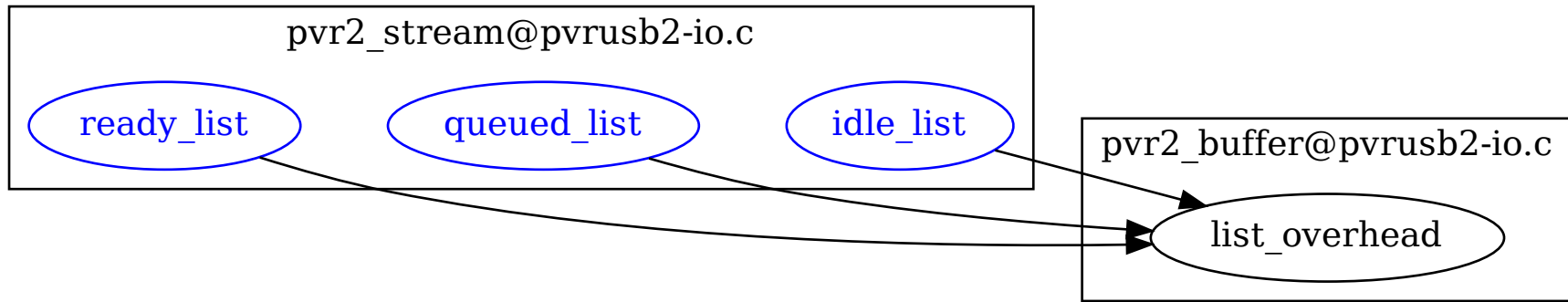
ecryptfs_kthread_ctl@kthread.c

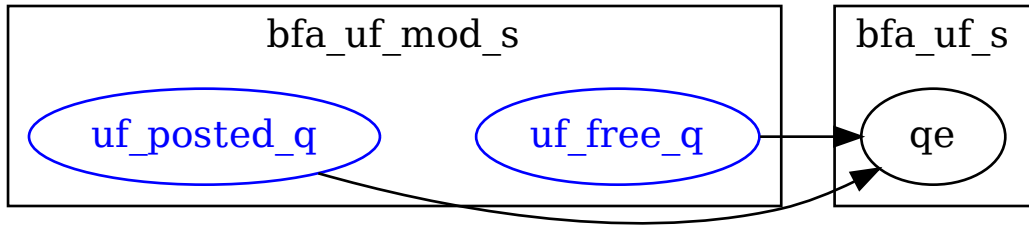
req_list

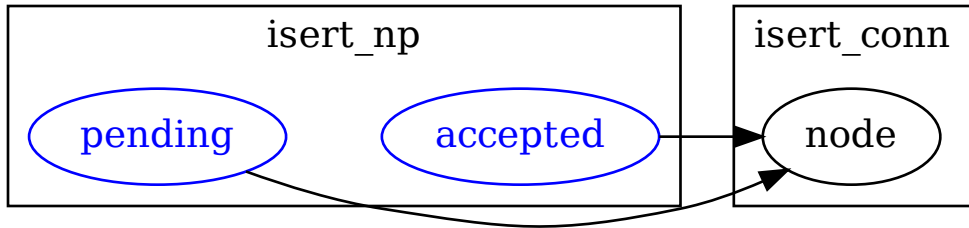
ecryptfs_open_req@kthread.c

kthread_ctl_list









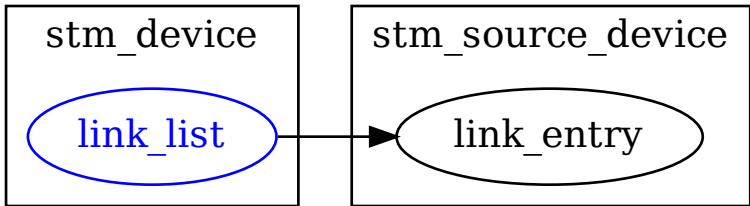
ecryptfs_mount_crypt_stat

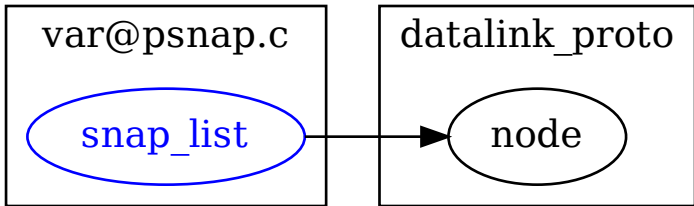
global_auth_tok_list

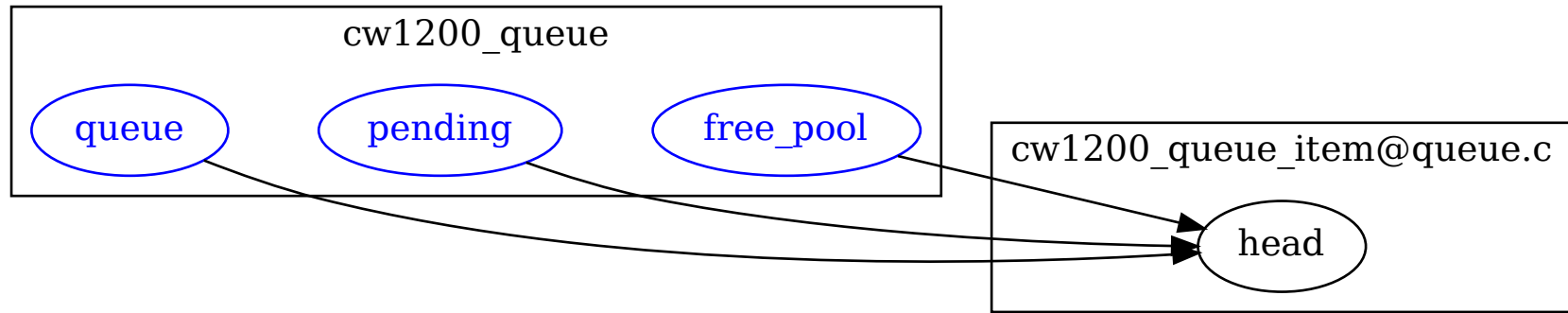
ecryptfs_global_auth_tok

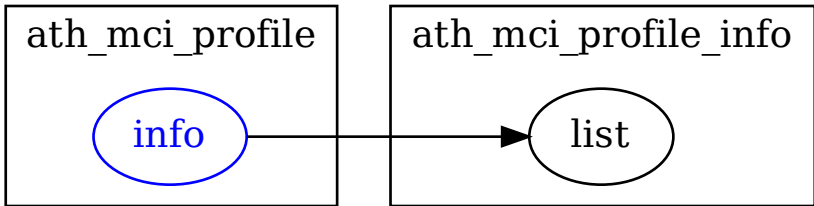
mount_crypt_stat_list











optee_supp

reqs

optee_supp_req@supp.c

link

A horizontal arrow points from the 'reqs' oval in the 'optee_supp' box to the 'link' oval in the 'optee_supp_req@supp.c' box.

var@console.c

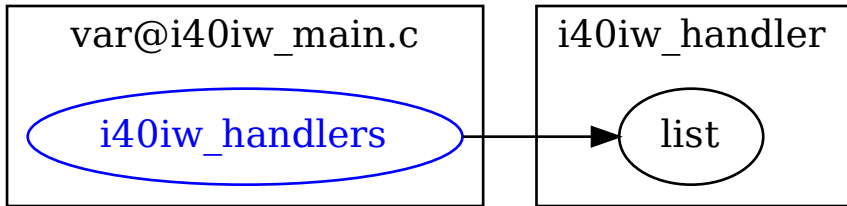
pm_vt_switch_list

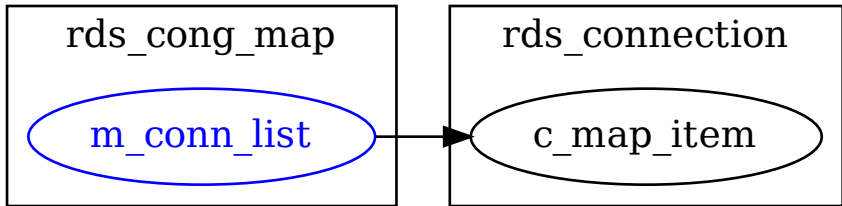
pm_vt_switch@console.c

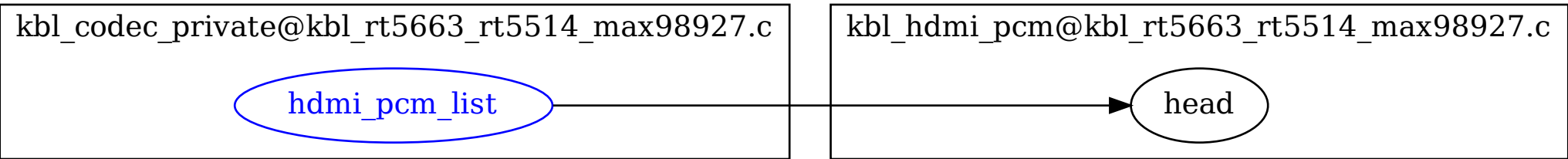
head

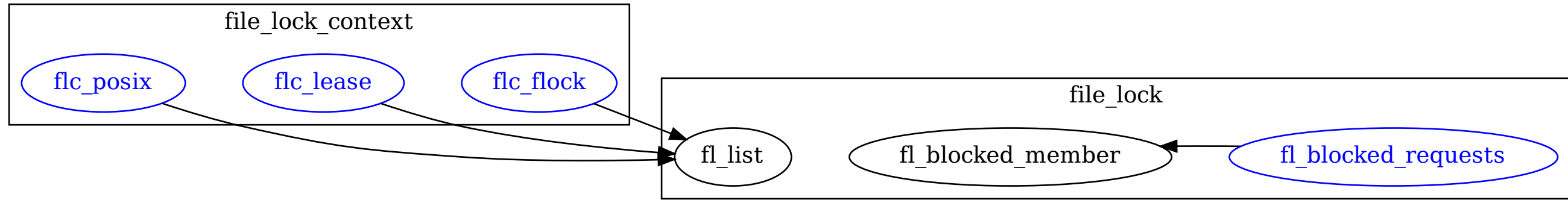
```
graph LR; A[pm_vt_switch_list] --> B((head))
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@console.c' contains an oval labeled 'pm_vt_switch_list' in blue text. On the right, a rectangular box labeled 'pm_vt_switch@console.c' contains an oval labeled 'head'. A horizontal arrow points from the right side of the 'pm_vt_switch_list' oval to the left side of the 'head' oval, indicating that 'pm_vt_switch_list' points to 'head'.









var@loop.c

nvme_loop_ctrl_list

nvme_loop_ctrl@loop.c

list

```
graph LR; A[nvme_loop_ctrl_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@loop.c' contains an oval labeled 'nvme_loop_ctrl_list' with a blue border. On the right, a rectangular box labeled 'nvme_loop_ctrl@loop.c' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'nvme_loop_ctrl_list' oval to the left side of the 'list' oval, indicating that the variable 'nvme_loop_ctrl_list' points to the variable 'list'.

var@sclp_cmd.c

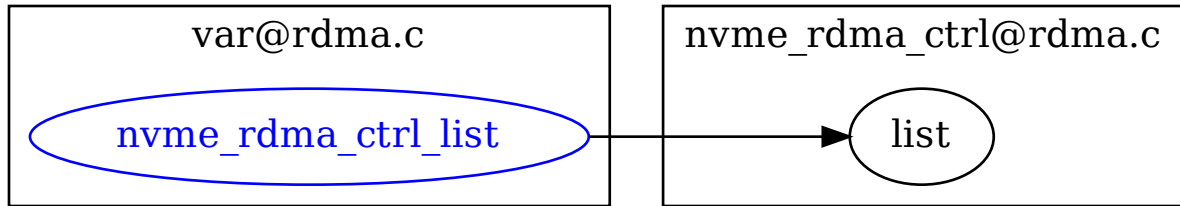
sclp_mem_list

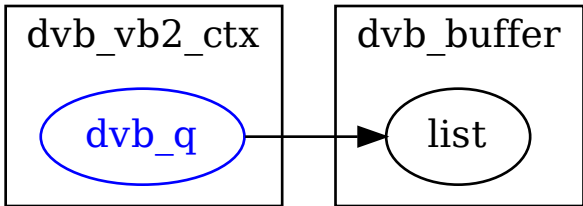
memory_increment@sclp_cmd.c

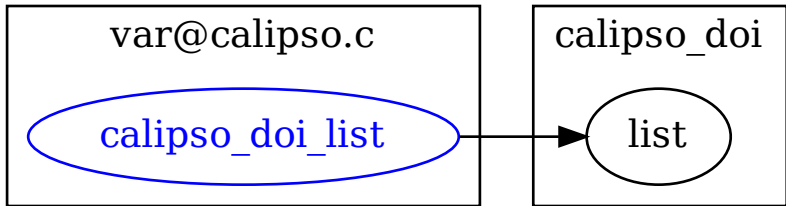
list

```
graph LR; subgraph Box1 [var@sclp_cmd.c]; A(sclp_mem_list); end; subgraph Box2 [memory_increment@sclp_cmd.c]; B((list)); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is titled 'var@sclp_cmd.c' and contains an oval labeled 'sclp_mem_list' in blue text. The right box is titled 'memory_increment@sclp_cmd.c' and contains an oval labeled 'list' in black text. A horizontal arrow points from the right side of the 'sclp_mem_list' oval to the left side of the 'list' oval.







aem_driver_data@ibmaem.c

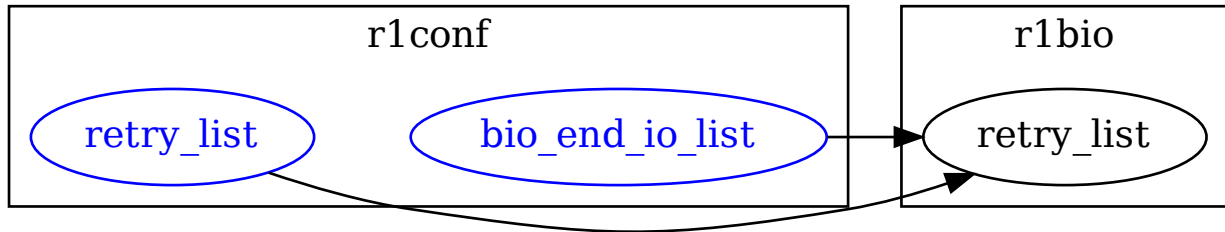
aem_devices

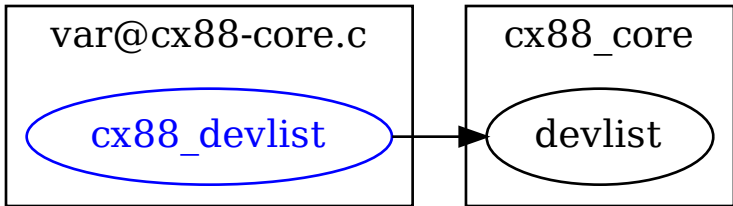
aem_data@ibmaem.c

list

```
graph LR; subgraph Box1 [aem_driver_data@ibmaem.c]; aem_devices; end; subgraph Box2 [aem_data@ibmaem.c]; list; end; aem_devices --> list;
```

The diagram consists of two rectangular boxes. The left box is titled 'aem_driver_data@ibmaem.c' and contains an oval labeled 'aem_devices' in blue text. The right box is titled 'aem_data@ibmaem.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'aem_devices' oval to the left side of the 'list' oval, indicating a pointer relationship.





a6xx_gpu_state@a6xx_gpu_state.c

objs

a6xx_state_memobj@a6xx_gpu_state.c

node

```
graph LR; A["a6xx_gpu_state@a6xx_gpu_state.c  
objs"] --> B["a6xx_state_memobj@a6xx_gpu_state.c  
node"]
```

The diagram consists of two rectangular boxes connected by a horizontal arrow. The left box contains the text 'a6xx_gpu_state@a6xx_gpu_state.c' at the top and a blue oval containing the text 'objs' below it. The right box contains the text 'a6xx_state_memobj@a6xx_gpu_state.c' at the top and a black oval containing the text 'node' below it. A horizontal arrow points from the 'objs' oval in the left box to the 'node' oval in the right box.

btrfs_free_space_ctl

trimming_ranges

btrfs_trim_range@free-space-cache.c

list

```
graph LR; A[btrfs_free_space_ctl] --> B((list));
```

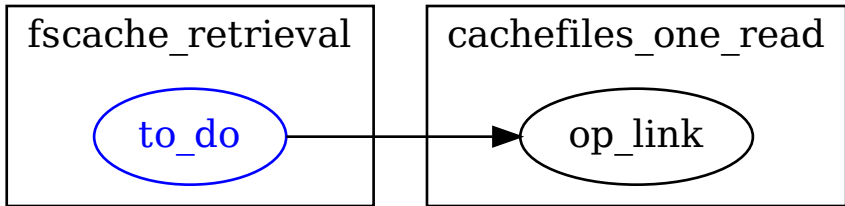
The diagram illustrates a pointer relationship. On the left, a rectangular box labeled 'btrfs_free_space_ctl' contains an oval labeled 'trimming_ranges' in blue text. A horizontal arrow points from the right side of the 'trimming_ranges' oval to a circular node labeled 'list' inside a larger rectangular box on the right. The larger box is labeled 'btrfs_trim_range@free-space-cache.c' at the top.

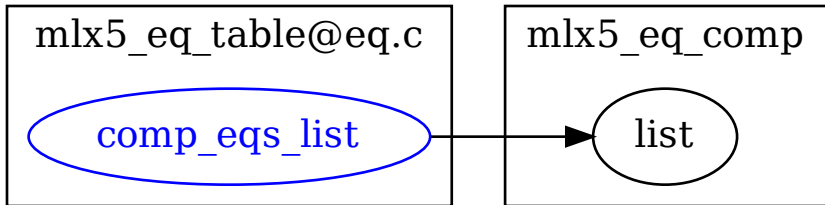
fscache_retrieval

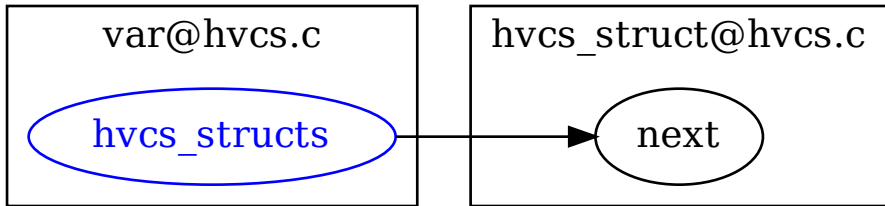
to_do

cachefiles_one_read

op_link







var@nf_flow_table_core.c

flowtables

nf_flowtable

list

```
graph LR; subgraph Box1 [var@nf_flow_table_core.c]; direction TB; FT(flowtables); end; subgraph Box2 [nf_flowtable]; direction TB; L(list); end; FT --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@nf_flow_table_core.c' and contains an oval labeled 'flowtables' in blue text. The right box is labeled 'nf_flowtable' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'flowtables' oval to the left side of the 'list' oval.

mtk_pcie@pcie-mediatek.c

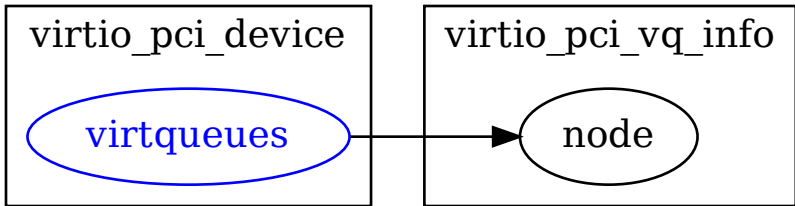
ports

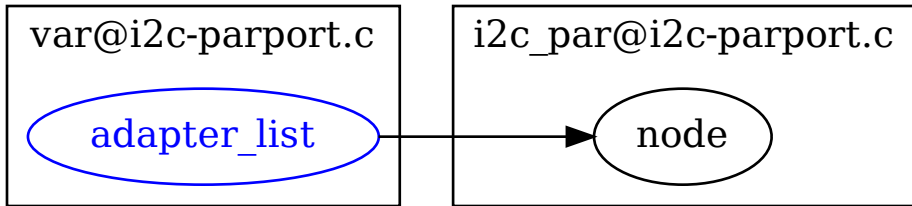
mtk_pcie_port@pcie-mediatek.c

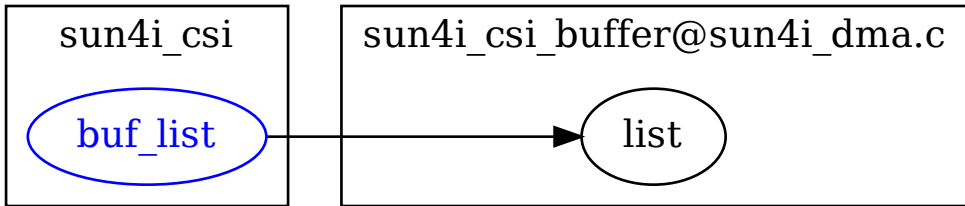
list

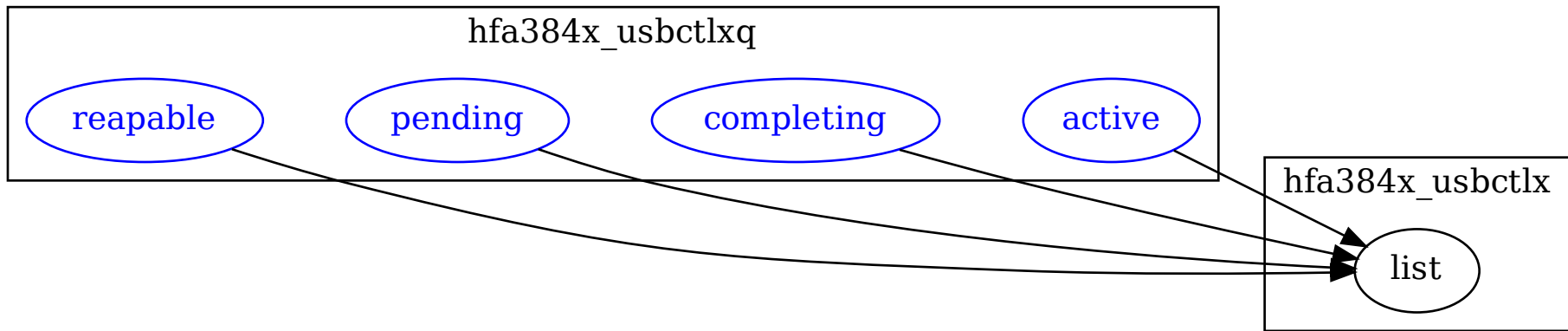
```
graph LR; A[mtk_pcie@pcie-mediatek.c] --> B[mtk_pcie_port@pcie-mediatek.c]; subgraph A [mtk_pcie@pcie-mediatek.c]; P([ports]); end; subgraph B [mtk_pcie_port@pcie-mediatek.c]; L([list]); end; P --> L;
```

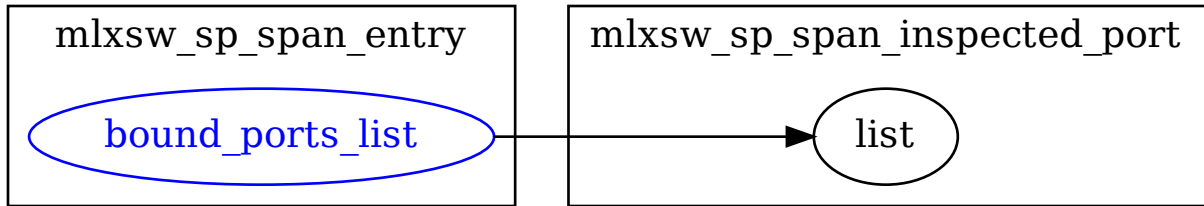
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'mtk_pcie@pcie-mediatek.c' contains an oval labeled 'ports' with a blue border. On the right, a box labeled 'mtk_pcie_port@pcie-mediatek.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'ports' oval to the 'list' oval, indicating that the 'ports' variable in the first module points to the 'list' variable in the second module.

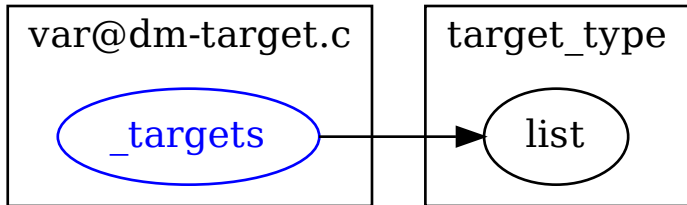










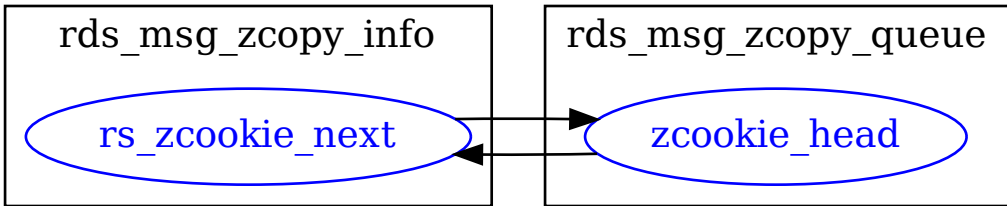


rds_msg_zcopy_info

rs_zcookie_next

rds_msg_zcopy_queue

zcookie_head



aspeed_video@aspeed-video.c

buffers

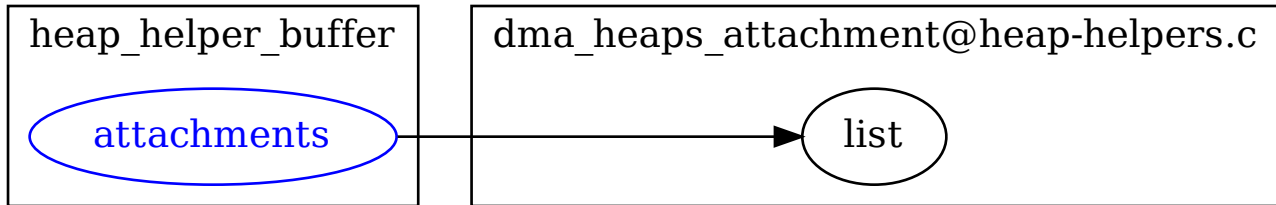
aspeed_video_buffer@aspeed-video.c

link



```
graph LR; subgraph LeftBox [aspeed_video@aspeed-video.c]; buffers; end; subgraph RightBox [aspeed_video_buffer@aspeed-video.c]; link; end; buffers --> link;
```

The diagram consists of two rectangular boxes. The left box contains the text 'aspeed_video@aspeed-video.c' at the top and an oval labeled 'buffers' in the center. The right box contains the text 'aspeed_video_buffer@aspeed-video.c' at the top and an oval labeled 'link' in the center. A horizontal arrow points from the 'buffers' oval in the left box to the 'link' oval in the right box.



var@smem_state.c

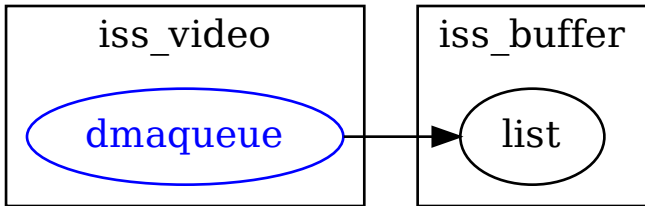
smem_states

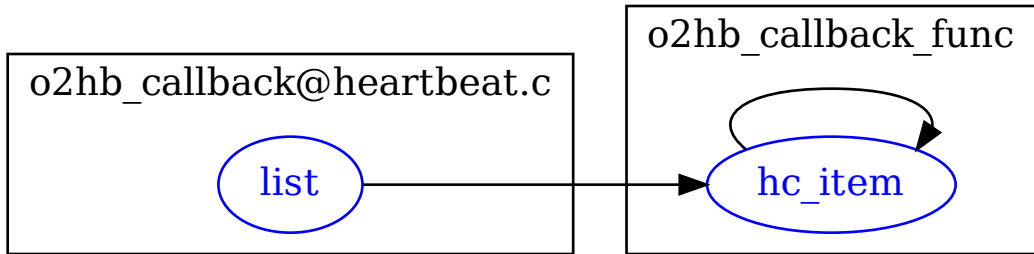
qcom_smem_state@smem_state.c

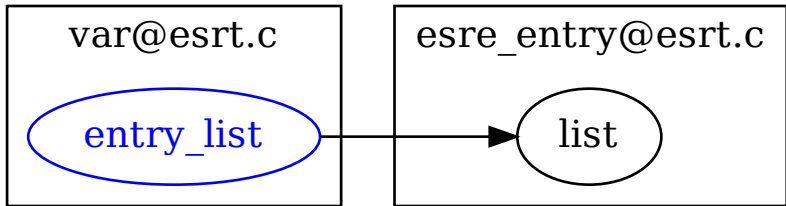
list

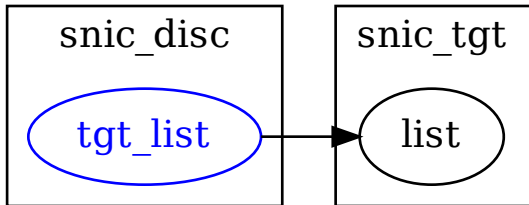
```
graph LR; subgraph Box1 [var@smem_state.c]; S(smem_states); end; subgraph Box2 [qcom_smem_state@smem_state.c]; L((list)); end; S --> L;
```

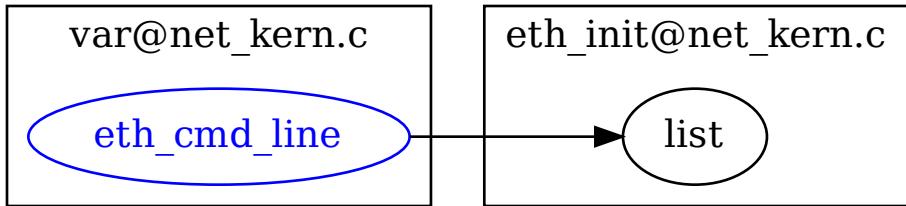
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@smem_state.c' contains an oval labeled 'smem_states' in blue text. On the right, a rectangular box labeled 'qcom_smem_state@smem_state.c' contains an oval labeled 'list'. A horizontal arrow points from the 'smem_states' oval to the 'list' oval, indicating that 'smem_states' points to 'list'.

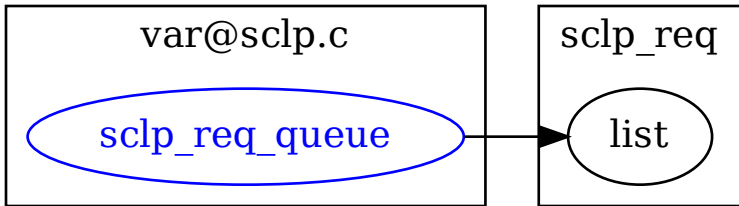












var@kthread.c

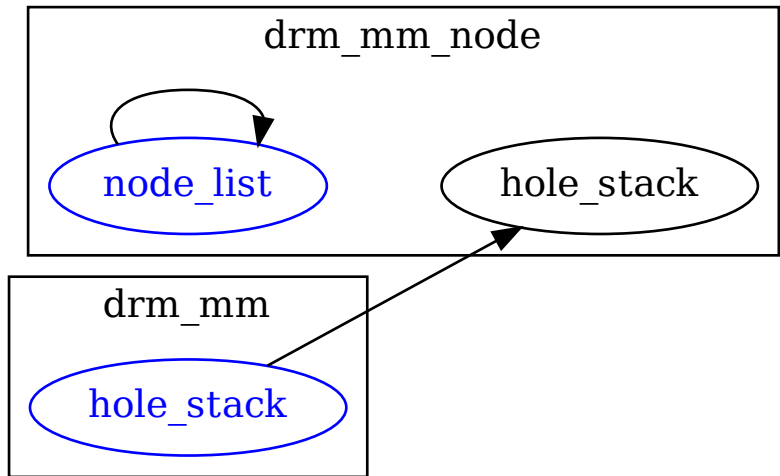
kthread_create_list

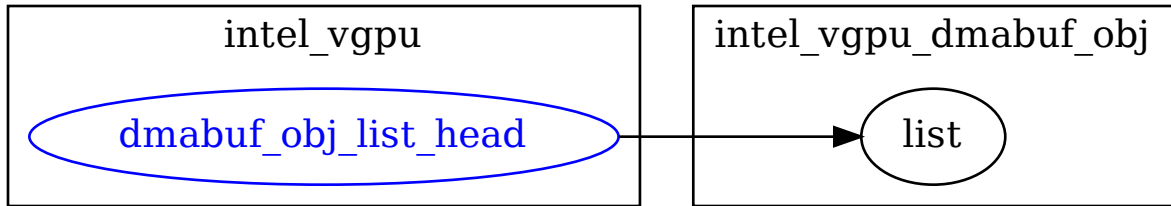
kthread_create_info@kthread.c

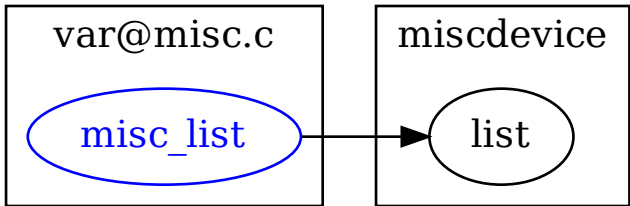
list

```
graph LR; A["var@kthread.c  
kthread_create_list"] --> B["kthread_create_info@kthread.c  
list"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@kthread.c' at the top and 'kthread_create_list' at the bottom, which is enclosed in a blue oval. The right box contains the text 'kthread_create_info@kthread.c' at the top and 'list' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.







var@qe_common.c

muram_block_list

muram_block@qe_common.c

head

```
graph LR; subgraph File1 [var@qe_common.c]; A([muram_block_list]); end; subgraph File2 [muram_block@qe_common.c]; B([head]); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@qe_common.c'. Inside this box, the variable 'muram_block_list' is enclosed in a blue oval. On the right, another rectangular box represents the source file 'muram_block@qe_common.c'. Inside this box, the variable 'head' is enclosed in a black oval. A horizontal arrow points from the right side of the 'muram_block_list' oval to the left side of the 'head' oval, indicating that 'muram_block_list' points to 'head'.

var@mediatek-cpufreq.c

dvfs_info_list

mtk_cpu_dvfs_info@mediatek-cpufreq.c

list_head

```
graph LR; A([dvfs_info_list]) --> B([list_head]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@mediatek-cpufreq.c' contains an oval labeled 'dvfs_info_list' with a blue border. On the right, a box labeled 'mtk_cpu_dvfs_info@mediatek-cpufreq.c' contains an oval labeled 'list_head' with a black border. A horizontal arrow points from the 'dvfs_info_list' oval to the 'list_head' oval, indicating that 'dvfs_info_list' points to 'list_head'.

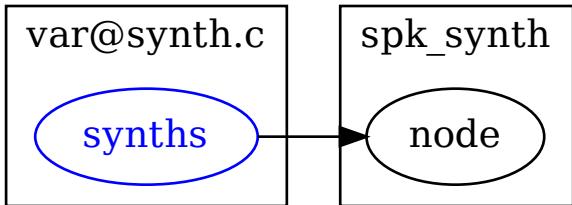
caif_device_entry_list@caif_dev.c

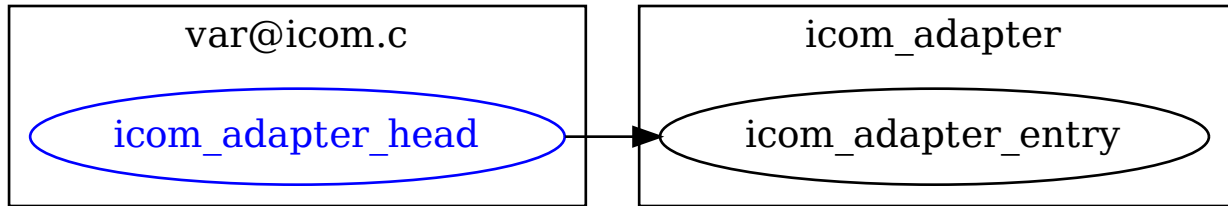
list

caif_device_entry@caif_dev.c

list







s3c24xx_txd@s3c24xx-dma.c

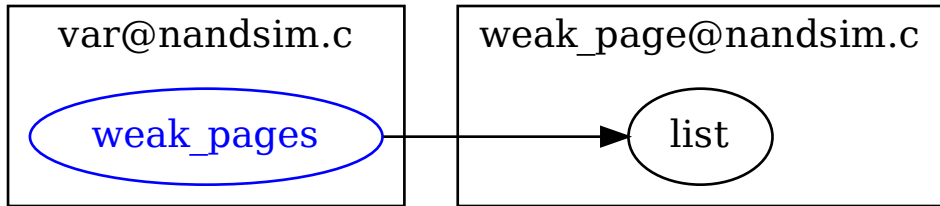
at

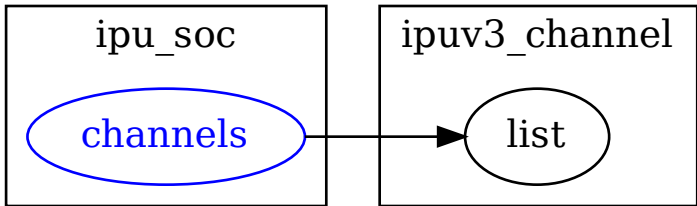
dsg_list

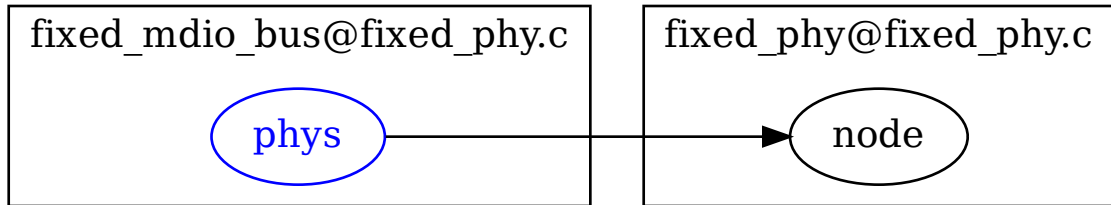
s3c24xx_sg@s3c24xx-dma.c

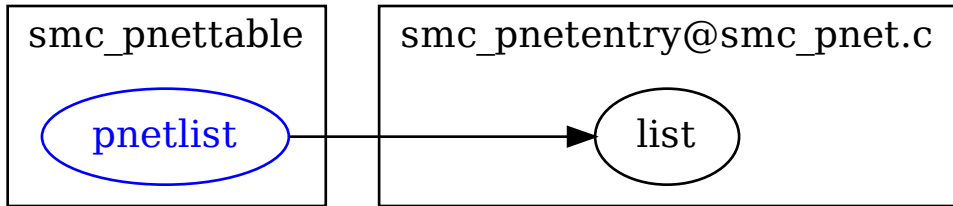
node

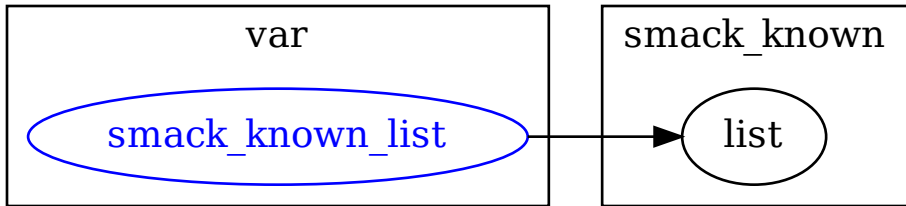












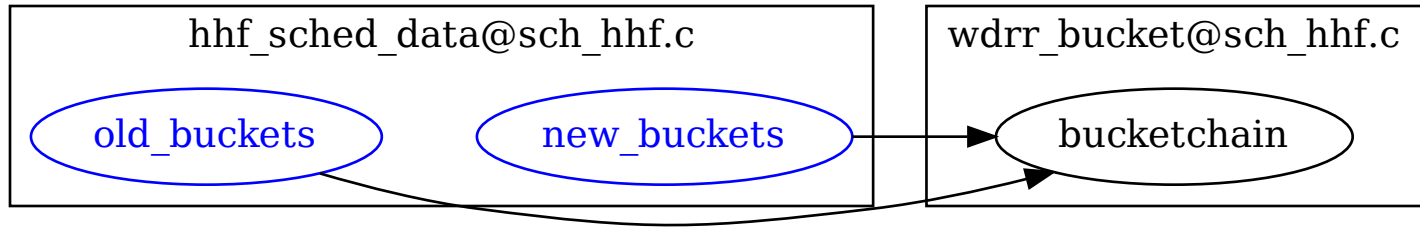
var@sch_teq1.c

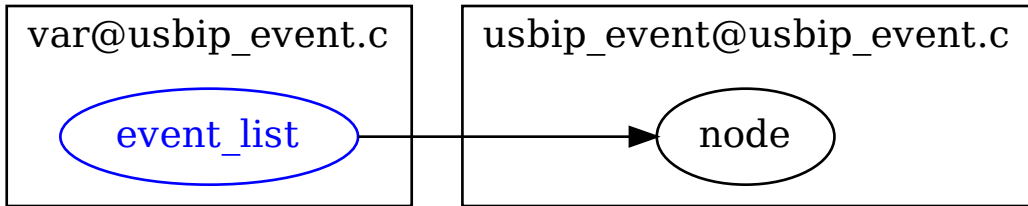
master_dev_list

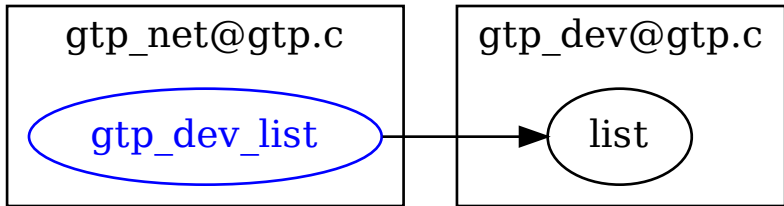
teql_master@sch_teq1.c

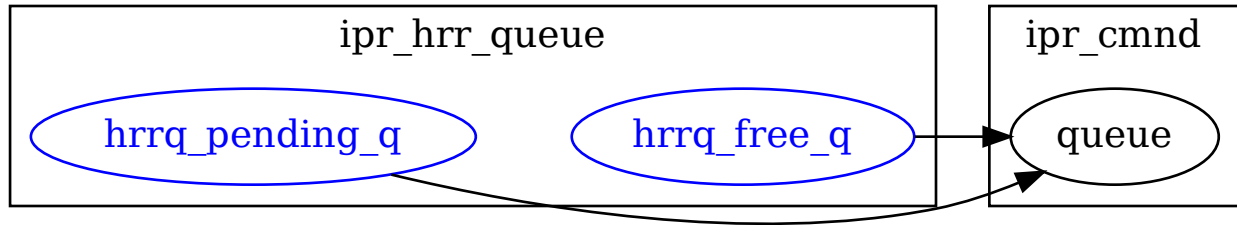
master_list

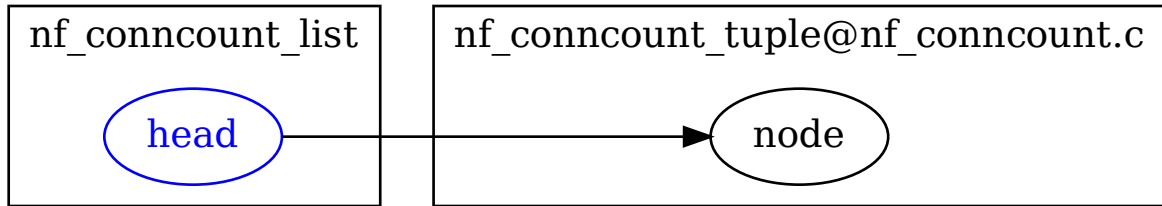


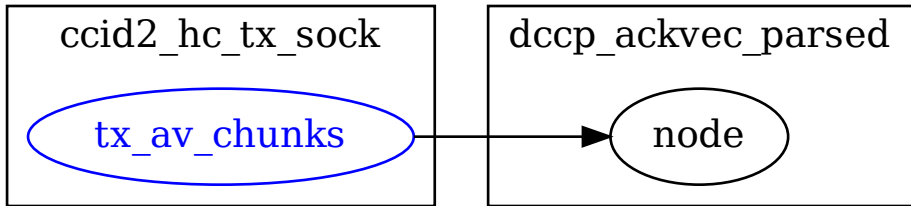


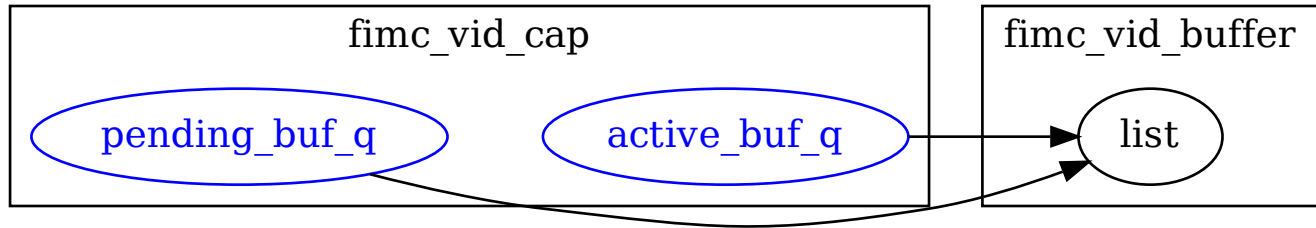


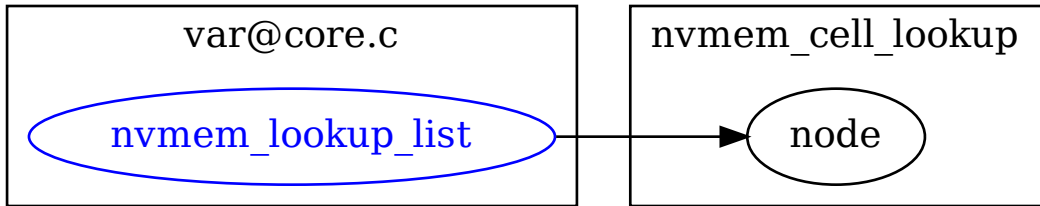












cbe_iommu@iommu.c

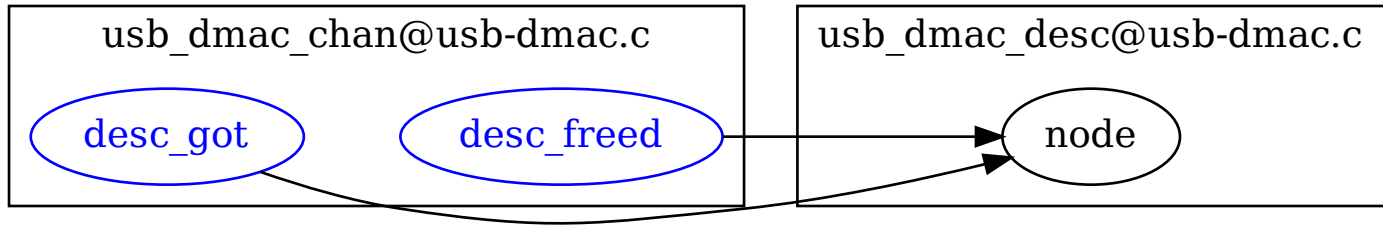
windows

iommu_window@iommu.c

list

```
graph LR; subgraph cbe_iommu [cbe_iommu@iommu.c]; windows; end; subgraph iommu_window [iommu_window@iommu.c]; list; end; windows --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'cbe_iommu@iommu.c' contains a blue oval labeled 'windows'. On the right, a box labeled 'iommu_window@iommu.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'windows' oval to the 'list' oval, indicating that the 'windows' variable in the first module points to the 'list' variable in the second module.



var@target_core_user.c

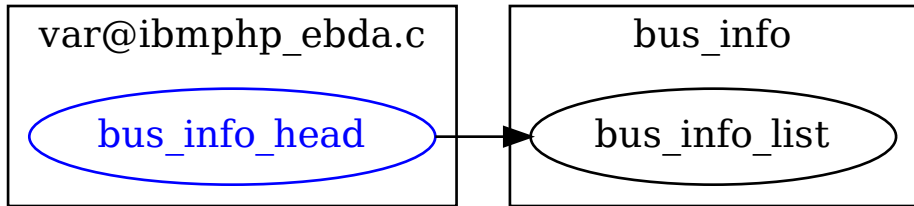
tcmu_nl_cmd_list

tcmu_nl_cmd@target_core_user.c

nl_list

```
graph LR; A["var@target_core_user.c  
tcmu_nl_cmd_list"] --> B["tcmu_nl_cmd@target_core_user.c  
nl_list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the file `var@target_core_user.c`, containing the variable `tcmu_nl_cmd_list`, which is highlighted by a blue oval. On the right, a box represents the file `tcmu_nl_cmd@target_core_user.c`, containing the variable `nl_list`, which is enclosed in a black oval. A horizontal arrow points from the `tcmu_nl_cmd_list` oval to the `nl_list` oval, indicating that the first variable points to the second.



var@intel-iommu.c

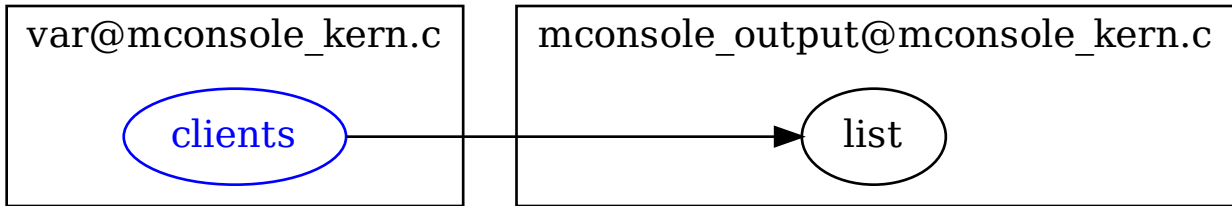
dm_{ar}_at_{sr}_units

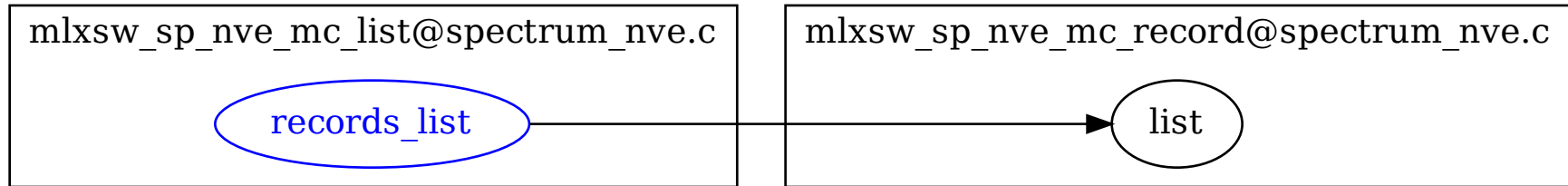
dm_{ar}_at_{sr}_unit@intel-iommu.c

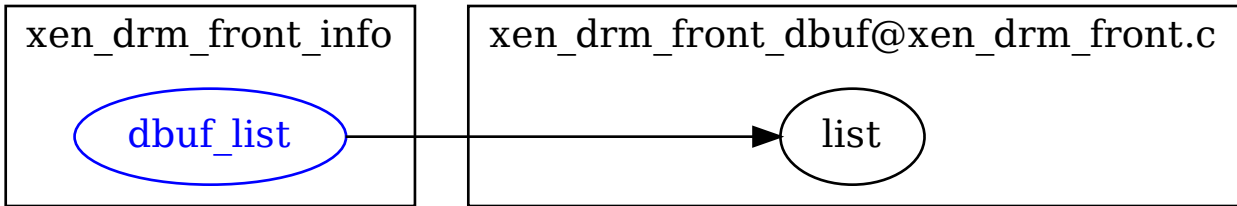
list

```
graph LR; subgraph File1 [var@intel-iommu.c]; A(dmar_atsr_units); end; subgraph File2 [dmar_atsr_unit@intel-iommu.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@intel-iommu.c`. Inside this box, the variable `dmar_atsr_units` is shown and highlighted with a blue oval. On the right, another rectangular box represents the source file `dmar_atsr_unit@intel-iommu.c`. Inside this box, there is an oval labeled `list`. A horizontal arrow points from the right side of the `dmar_atsr_units` oval in the first box to the left side of the `list` oval in the second box, indicating that the first variable points to the second.







preempted_vcore_list@book3s_hv.c

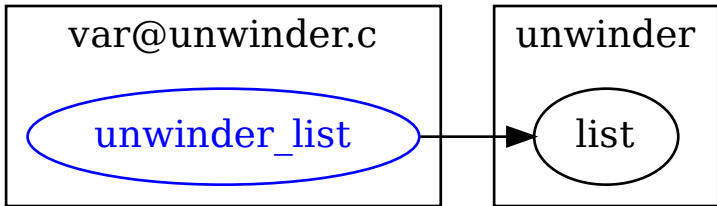
list

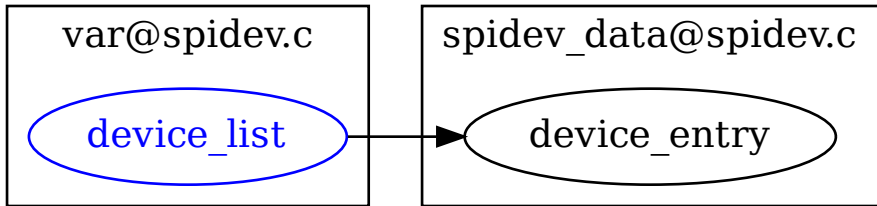
kvmppc_vcore

preempt_list

```
graph LR; A[preempted_vcore_list@book3s_hv.c] --> B[kvmppc_vcore.preempt_list];
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'preempted_vcore_list@book3s_hv.c'. Inside this box, there is a blue-outlined oval containing the word 'list'. A horizontal arrow points from the right side of this oval to the left side of another oval. This second oval is located inside a rectangular box on the right, which also contains the text 'kvmppc_vcore' above it. The oval on the right contains the text 'preempt_list'.





mlx5_fpga_tls

pending_cmds

mlx5_fpga_tls_command_context@tls.c

list

```
graph LR; A[mlx5_fpga_tls] --> B((list)); subgraph C [mlx5_fpga_tls_command_context@tls.c]; B; end
```

The diagram illustrates a pointer relationship. On the left, a rectangular box labeled 'mlx5_fpga_tls' contains an oval labeled 'pending_cmds'. A horizontal arrow points from the right side of the 'pending_cmds' oval to a circular node labeled 'list'. This 'list' node is located inside a larger rectangular box on the right labeled 'mlx5_fpga_tls_command_context@tls.c'.

var@sch_taprio.c

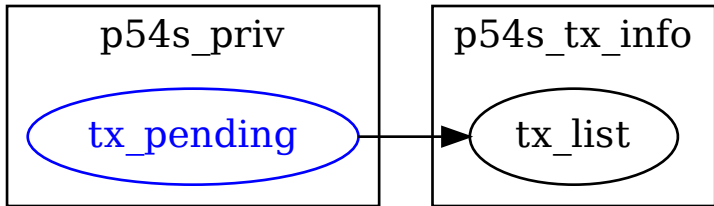
taprio_list

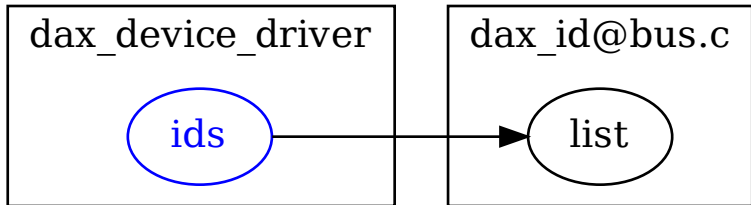
taprio_sched@sch_taprio.c

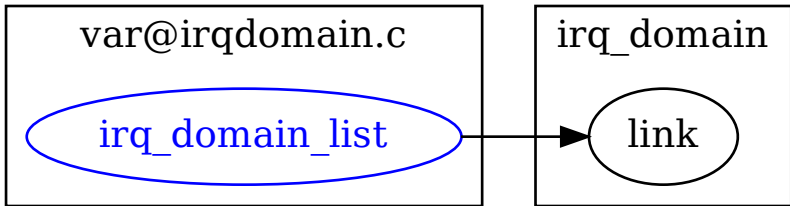
taprio_list

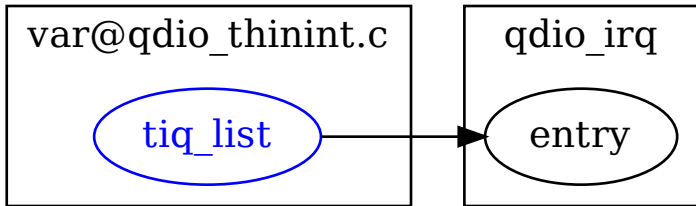
```
graph LR; A[var@sch_taprio.c] --> B[taprio_sched@sch_taprio.c]; subgraph A [var@sch_taprio.c]; A1([taprio_list]); end; subgraph B [taprio_sched@sch_taprio.c]; B1([taprio_list]); end; A1 --> B1;
```

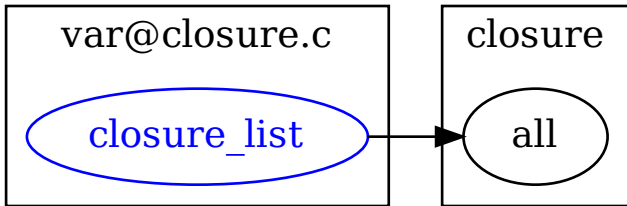
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@sch_taprio.c' contains an oval labeled 'taprio_list' with a blue border. On the right, a box labeled 'taprio_sched@sch_taprio.c' contains an oval labeled 'taprio_list' with a black border. A horizontal arrow points from the 'taprio_list' oval in the left box to the 'taprio_list' oval in the right box, indicating that the variable 'taprio_list' in 'var@sch_taprio.c' points to the 'taprio_list' variable in 'taprio_sched@sch_taprio.c'.

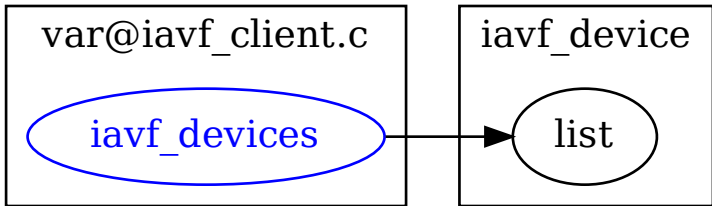


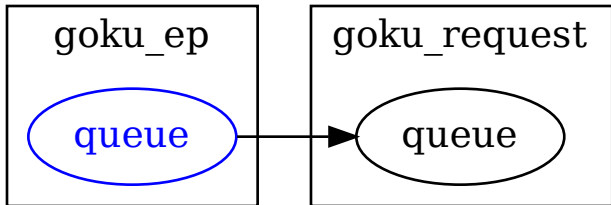


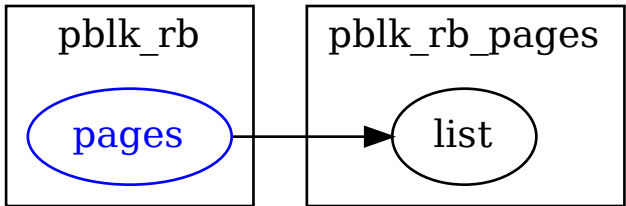


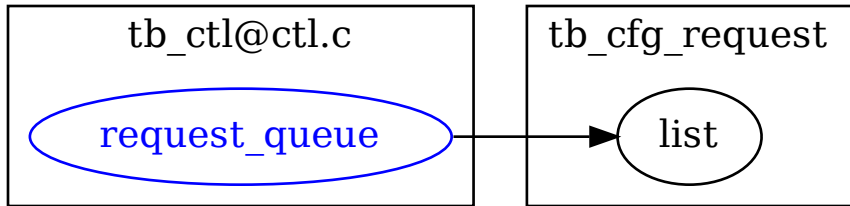












var@trans_xen.c

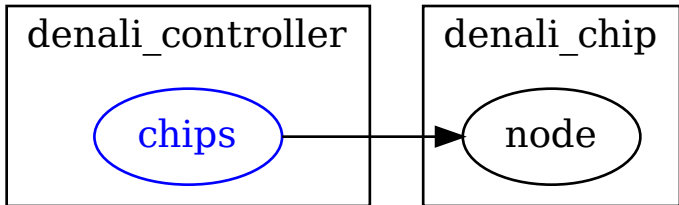
xen_9pfs_devs

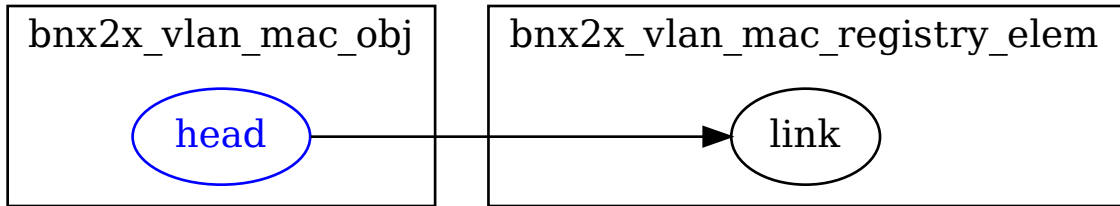
xen_9pfs_front_priv@trans_xen.c

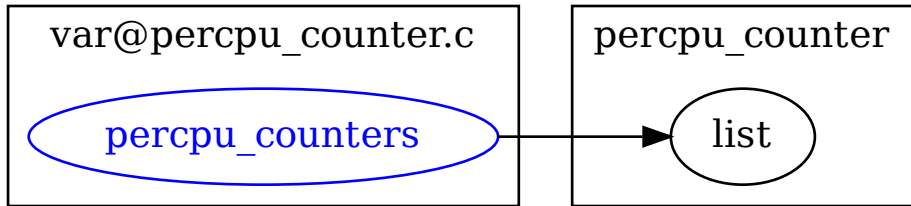
list

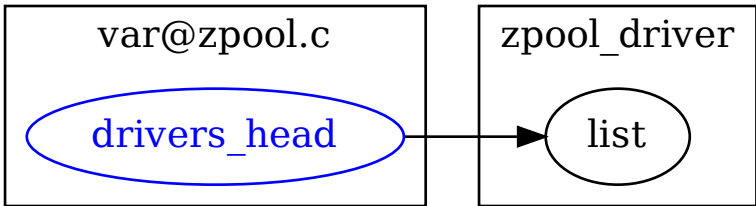
```
graph LR; subgraph File1 [var@trans_xen.c]; dev(xen_9pfs_devs); end; subgraph File2 [xen_9pfs_front_priv@trans_xen.c]; list((list)); end; dev --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file `var@trans_xen.c`, containing the variable `xen_9pfs_devs` which is highlighted by a blue oval. On the right, a box represents the source file `xen_9pfs_front_priv@trans_xen.c`, containing a circular node labeled `list`. A horizontal arrow points from the `xen_9pfs_devs` variable to the `list` node, indicating that `xen_9pfs_devs` points to `list`.









var@vfio_platform_common.c

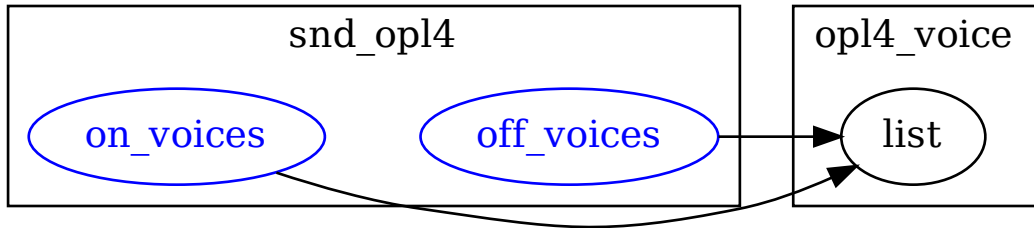
reset_list

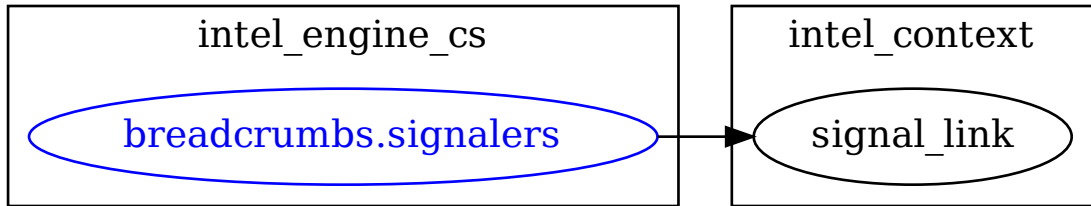
vfio_platform_reset_node

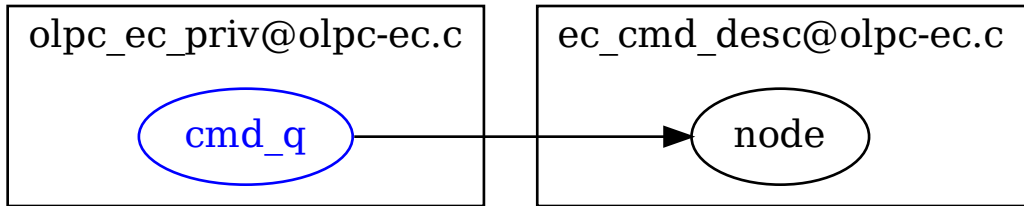
link

```
graph LR; A[reset_list] --> B(link);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@vfio_platform_common.c' contains an oval labeled 'reset_list' in blue text. On the right, a box labeled 'vfio_platform_reset_node' contains an oval labeled 'link'. A horizontal arrow points from the 'reset_list' oval to the 'link' oval, indicating that 'reset_list' points to 'link'.







bxt_card_private@bxt_da7219_max98357a.c

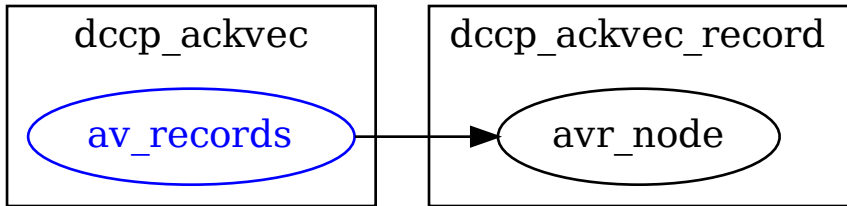
hdmi_pcm_list

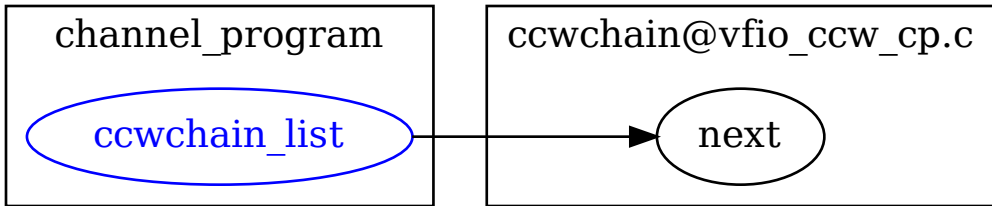
bxt_hdmi_pcm@bxt_da7219_max98357a.c

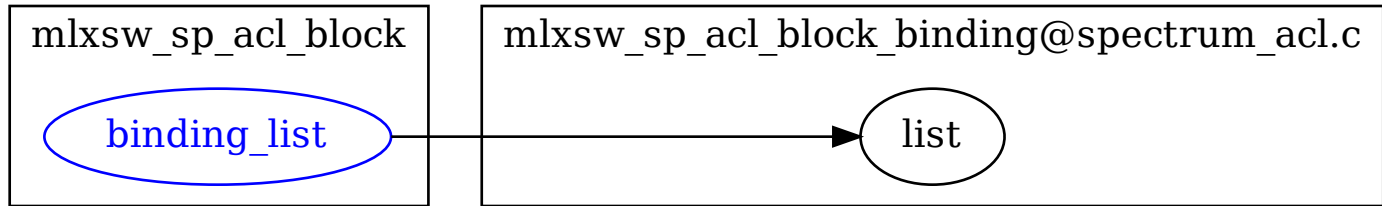
head

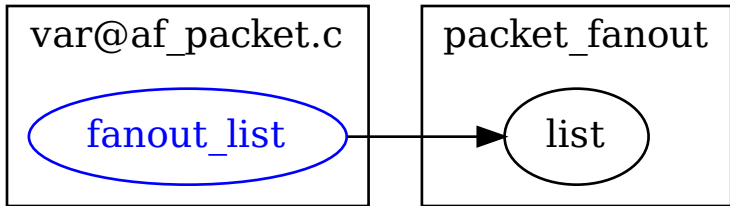
```
graph LR; A[bxt_card_private@bxt_da7219_max98357a.c] --> B[bxt_hdmi_pcm@bxt_da7219_max98357a.c]; B --> C((head));
```

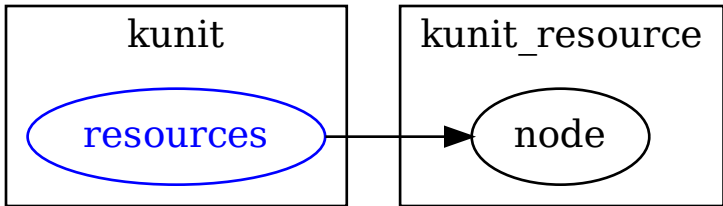
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box contains the identifier 'bxt_card_private@bxt_da7219_max98357a.c'. Inside this box, the variable 'hdmi_pcm_list' is highlighted with a blue oval. A horizontal line extends from the right side of this box to the left side of a second rectangular box on the right. This second box contains the identifier 'bxt_hdmi_pcm@bxt_da7219_max98357a.c'. Inside this box, the variable 'head' is enclosed in a black oval. A horizontal arrow points from the right side of the first box to the left side of the second box, indicating that 'hdmi_pcm_list' points to 'head'.

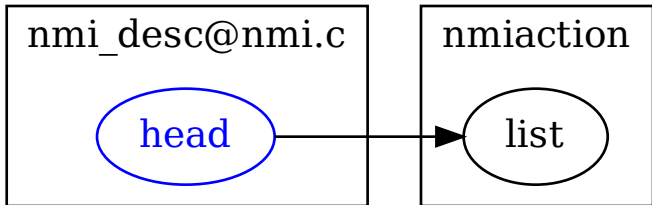


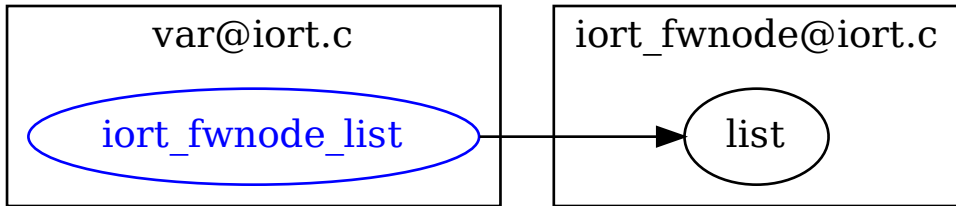


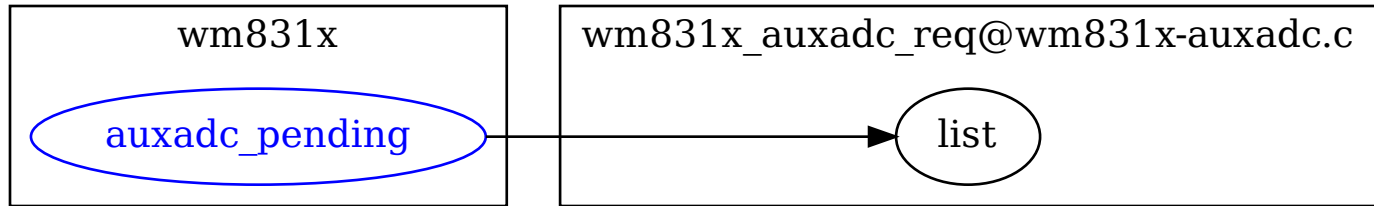


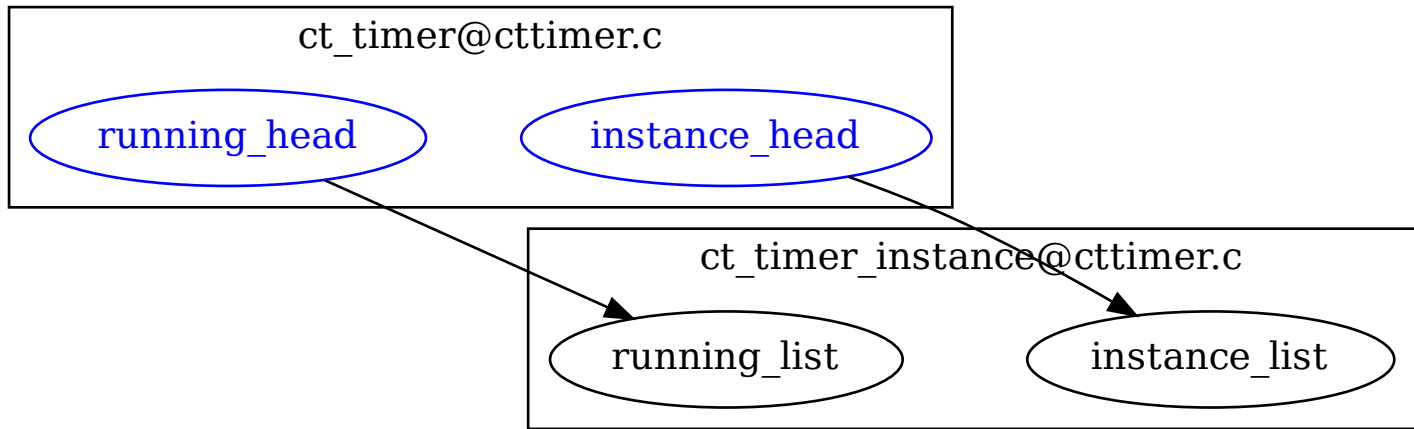


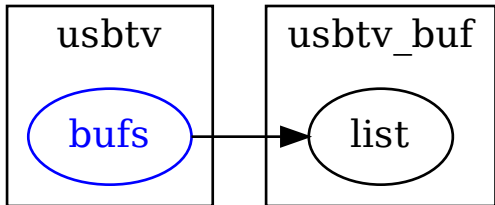


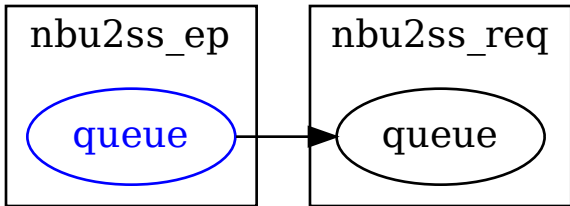


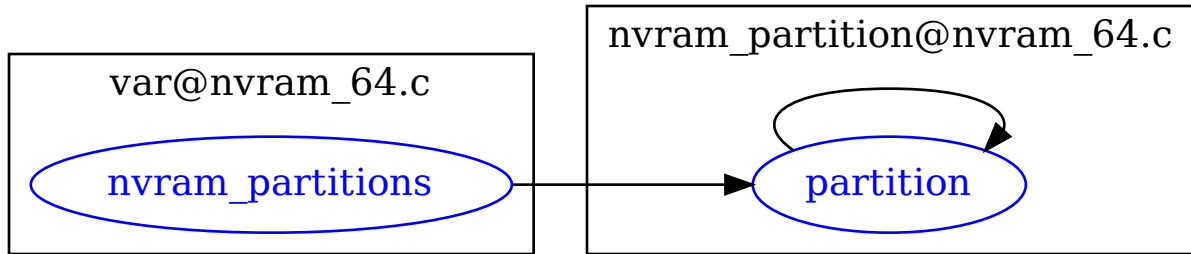


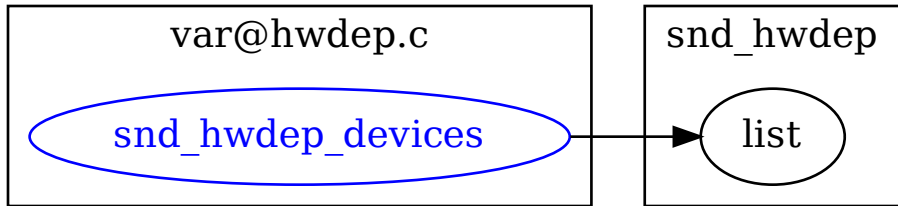


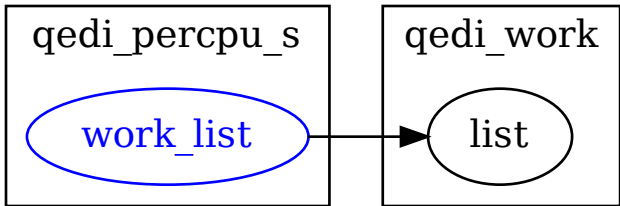












var@devfreq-event.c

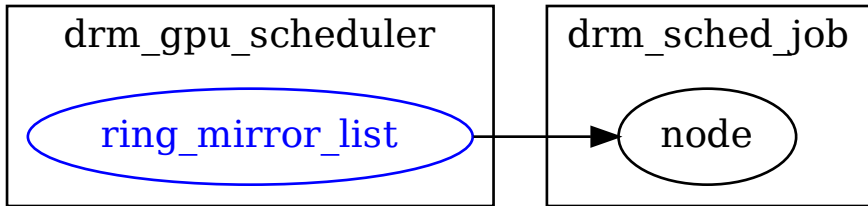
devfreq_event_list

devfreq_event_dev

node

```
graph LR; A[devfreq_event_list] --> B((node));
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'var@devfreq-event.c' contains an oval labeled 'devfreq_event_list' with a blue border. On the right, a rectangular box labeled 'devfreq_event_dev' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'devfreq_event_list' oval to the left side of the 'node' oval, indicating that 'devfreq_event_list' points to 'node'.



pcifront_device@xen-pcifront.c

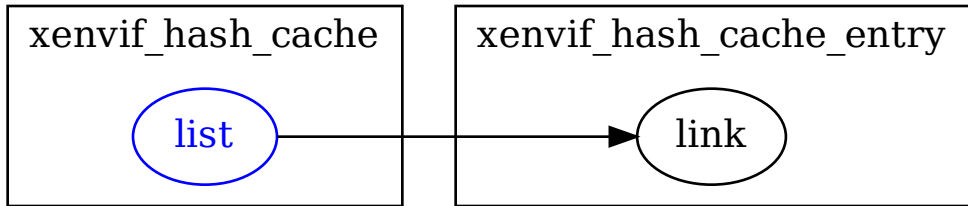
root_buses

pci_bus_entry@xen-pcifront.c

list

```
graph LR; subgraph PCIFrontDevice [pcifront_device@xen-pcifront.c]; root_buses; end; subgraph PCIBusEntry [pci_bus_entry@xen-pcifront.c]; list; end; root_buses --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'pcifront_device@xen-pcifront.c' contains an oval labeled 'root_buses'. On the right, a box labeled 'pci_bus_entry@xen-pcifront.c' contains an oval labeled 'list'. A horizontal arrow points from the 'root_buses' oval to the 'list' oval, indicating that 'root_buses' points to the 'list' structure.



var@iio-trig-sysfs.c

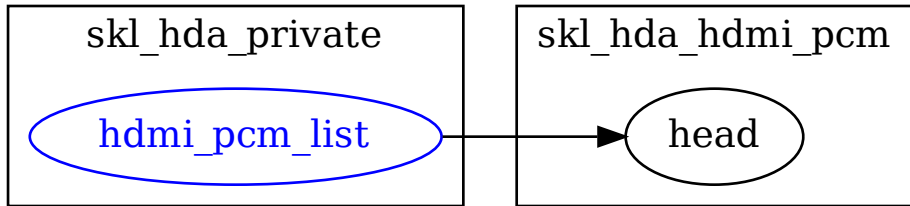
iio_sysfs_trig_list

iio_sysfs_trig@iio-trig-sysfs.c

1

```
graph LR; A["var@iio-trig-sysfs.c  
iio_sysfs_trig_list"] --> B["iio_sysfs_trig@iio-trig-sysfs.c  
1"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file `var@iio-trig-sysfs.c`, containing the variable `iio_sysfs_trig_list`, which is highlighted by a blue oval. On the right, a box represents the source file `iio_sysfs_trig@iio-trig-sysfs.c`, containing the variable `1`, which is enclosed in a black oval. A horizontal arrow points from the `iio_sysfs_trig_list` variable to the `1` variable, indicating that the first variable points to the second.



var@timerfd.c

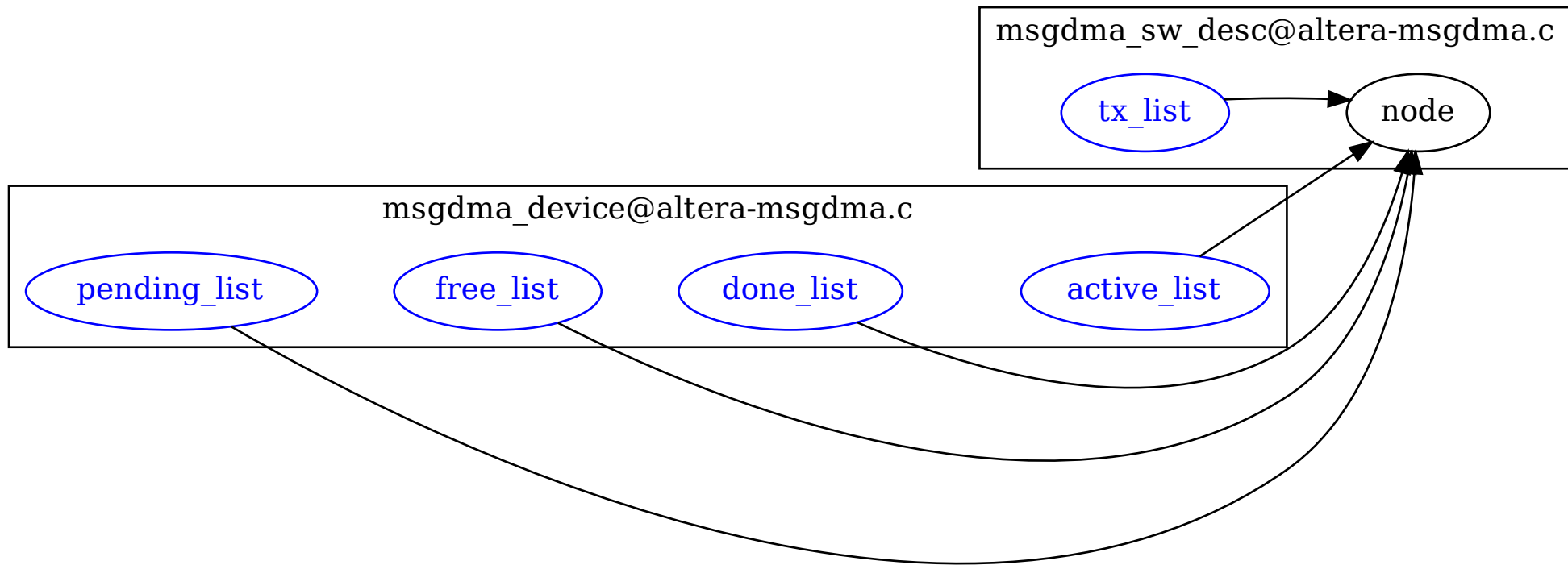
cancel_list

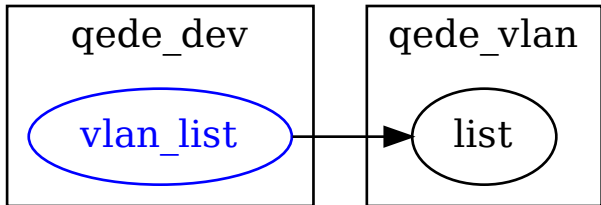
timerfd_ctx@timerfd.c

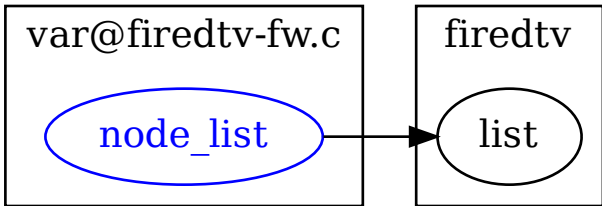
clist

```
graph LR; subgraph var_timerfd_c [var@timerfd.c]; cancel_list; end; subgraph timerfd_ctx_timerfd_c [timerfd_ctx@timerfd.c]; clist; end; cancel_list --> clist;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@timerfd.c' contains an oval labeled 'cancel_list'. On the right, a box labeled 'timerfd_ctx@timerfd.c' contains an oval labeled 'clist'. A horizontal arrow points from the 'cancel_list' oval to the 'clist' oval, indicating that 'cancel_list' points to 'clist'.







vpd_section@vpd.c

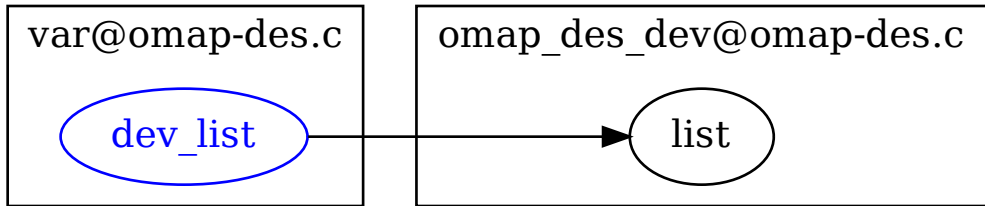
attribs

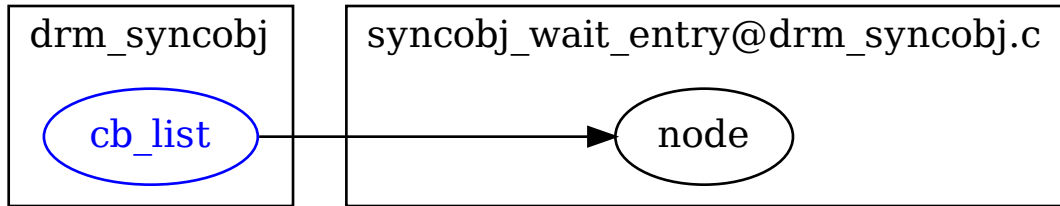
vpd_attr_info@vpd.c

list

```
graph LR; subgraph vpd_section [vpd_section@vpd.c]; attribs; end; subgraph vpd_attr_info [vpd_attr_info@vpd.c]; list; end; attribs --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'vpd_section@vpd.c' contains an oval labeled 'attribs'. On the right, a box labeled 'vpd_attr_info@vpd.c' contains an oval labeled 'list'. A horizontal arrow points from the 'attribs' oval to the 'list' oval, indicating that the 'attribs' field in the 'vpd_section' structure points to the 'list' field in the 'vpd_attr_info' structure.





var@test_klp_shadow_vars.c

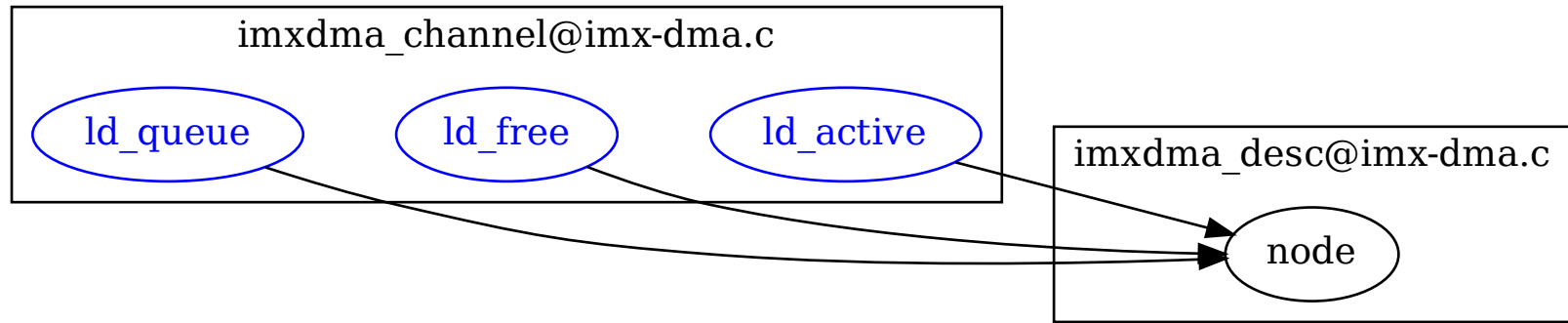
ptr_list

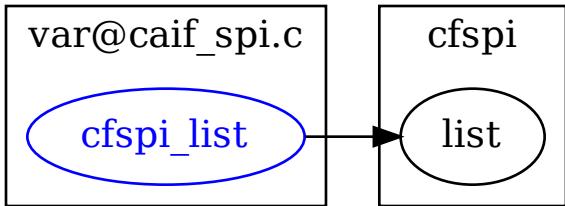
shadow_ptr@test_klp_shadow_vars.c

list

```
graph LR; ptr_list([ptr_list]) --> list([list]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@test_klp_shadow_vars.c' contains an oval labeled 'ptr_list' in blue text. On the right, a box labeled 'shadow_ptr@test_klp_shadow_vars.c' contains an oval labeled 'list' in black text. A horizontal arrow points from the 'ptr_list' oval to the 'list' oval, indicating that 'ptr_list' points to 'list'.





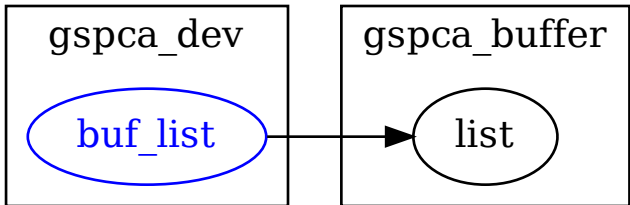
var@sch_cbs.c

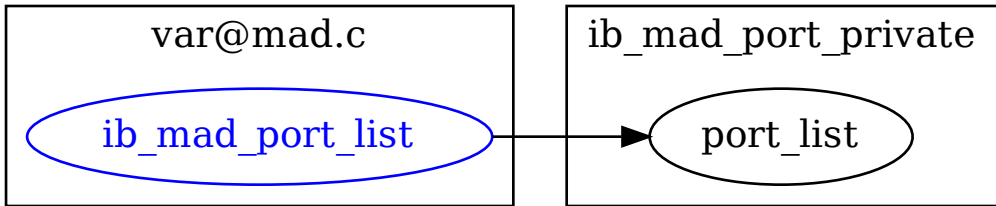
cbs_list

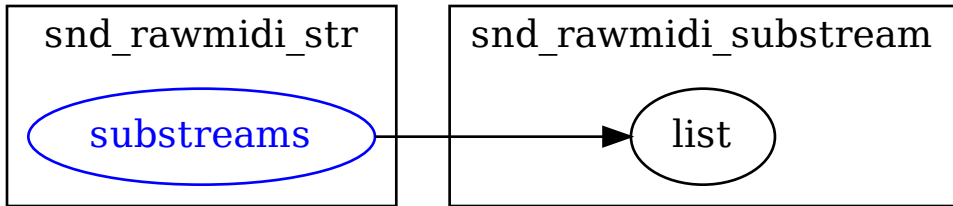
cbs_sched_data@sch_cbs.c

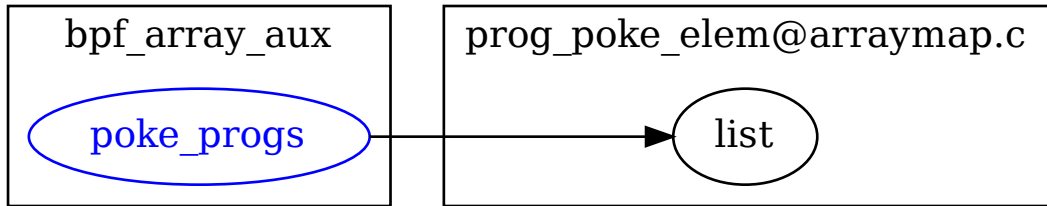
cbs_list

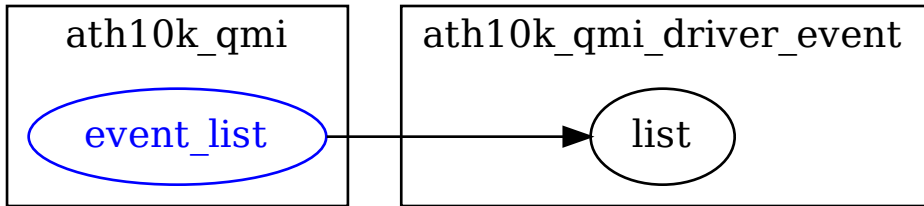












mtk_rpmsg_rproc_subdev@mtk_rpmsg.c

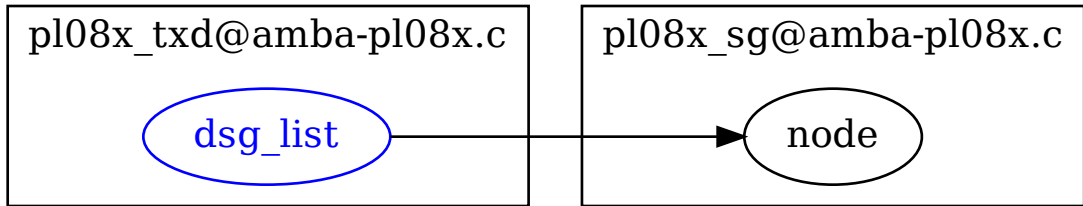
channels

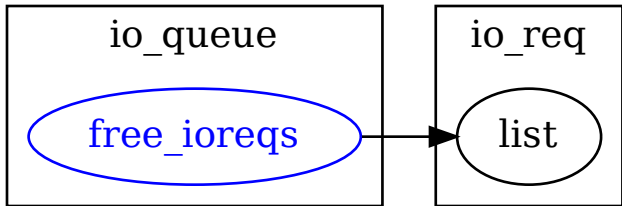
mtk_rpmsg_channel_info@mtk_rpmsg.c

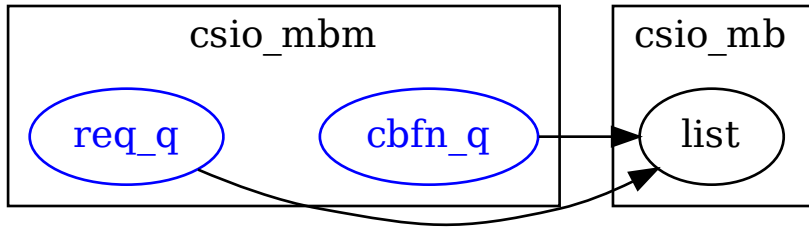
list

```
graph LR; subgraph Box1 [mtk_rpmsg_rproc_subdev@mtk_rpmsg.c]; channels; end; subgraph Box2 [mtk_rpmsg_channel_info@mtk_rpmsg.c]; list; end; channels --> list;
```

The diagram consists of two rectangular boxes. The left box is titled 'mtk_rpmsg_rproc_subdev@mtk_rpmsg.c' and contains an oval labeled 'channels'. The right box is titled 'mtk_rpmsg_channel_info@mtk_rpmsg.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'channels' oval in the left box to the 'list' oval in the right box.







var@warp.c

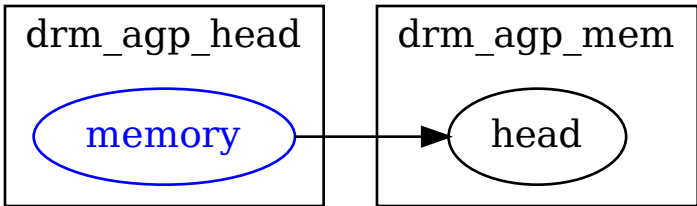
dtm_shutdown_list

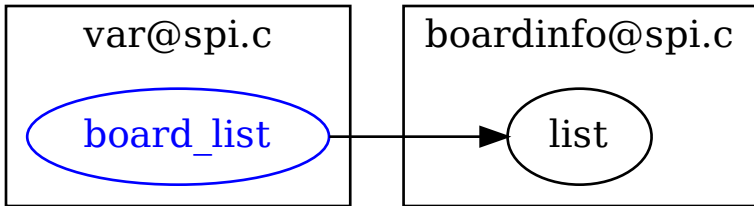
dtm_shutdown@warp.c

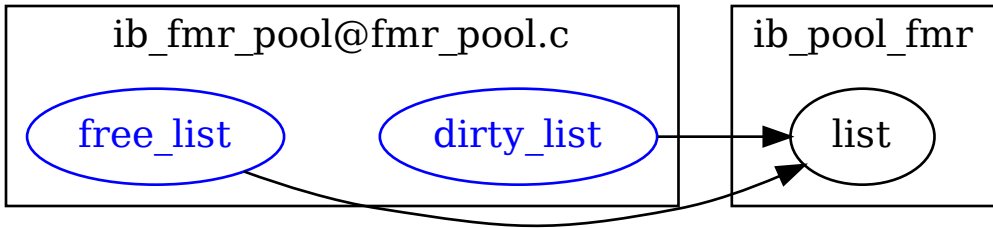
list

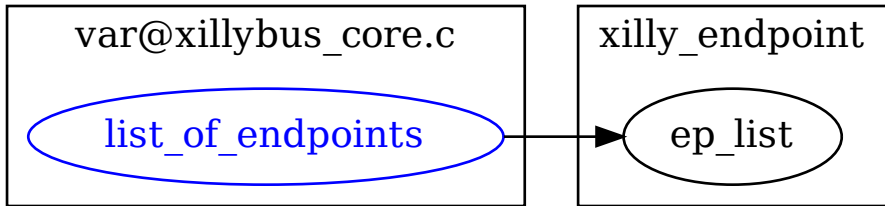
```
graph LR; subgraph var_warp_c [var@warp.c]; dtm_shutdown_list; end; subgraph dtm_shutdown_warp_c [dtm_shutdown@warp.c]; list; end; dtm_shutdown_list --> list;
```

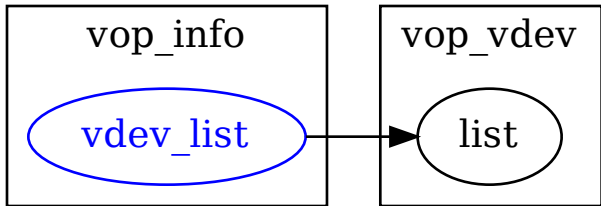
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@warp.c' contains an oval labeled 'dtm_shutdown_list'. On the right, a box labeled 'dtm_shutdown@warp.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dtm_shutdown_list' oval to the 'list' oval, indicating that the variable 'dtm_shutdown_list' in 'var@warp.c' points to the 'list' variable in 'dtm_shutdown@warp.c'.












cx23885_dmaqueue

active

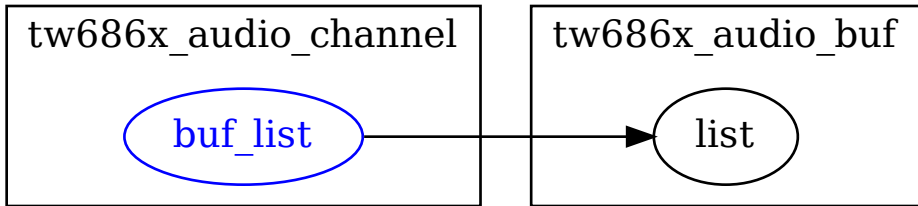
cx23885_buffer

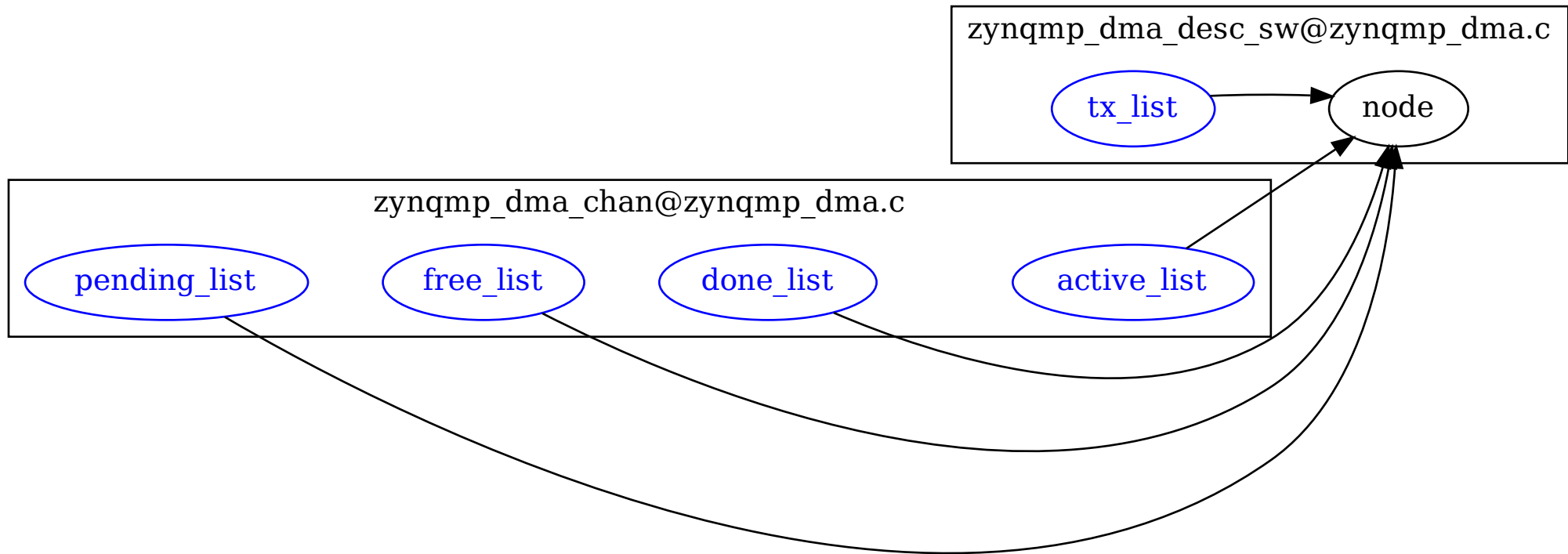
queue



```
graph LR; subgraph cx23885_dmaqueue; active; end; subgraph cx23885_buffer; queue; end; active --> queue;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'cx23885_dmaqueue' contains an oval labeled 'active' with a blue border. On the right, a rectangular box labeled 'cx23885_buffer' contains an oval labeled 'queue' with a black border. A horizontal arrow points from the 'active' oval to the 'queue' oval, indicating that the 'active' pointer refers to the 'queue' structure within the 'cx23885_buffer'.





flow_indr_block_dev@flow_offload.c

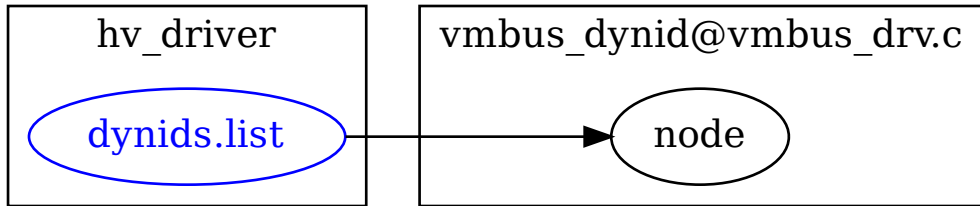
cb_list

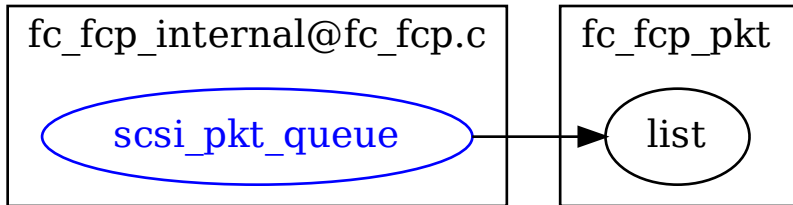
flow_indr_block_cb@flow_offload.c

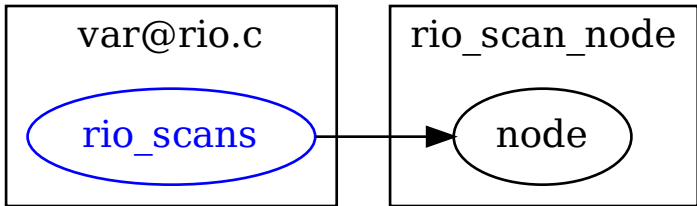
list

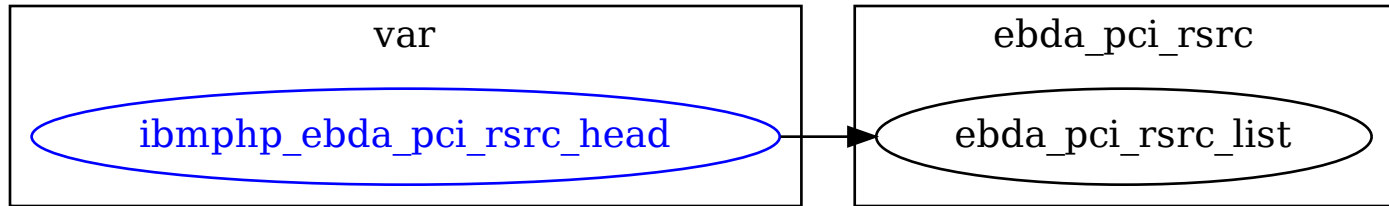


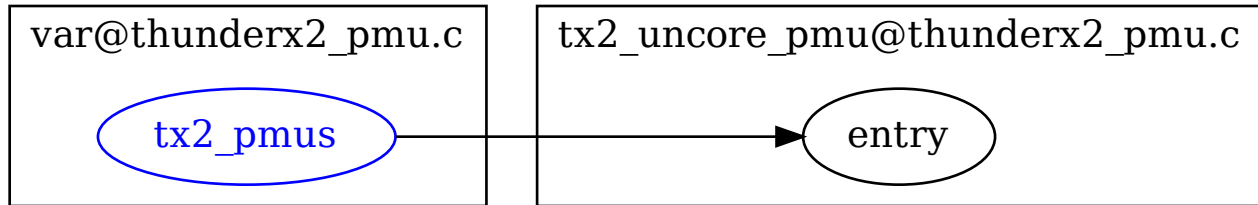
```
graph LR; subgraph Module1 [flow_indr_block_dev@flow_offload.c]; cb_list((cb_list)); end; subgraph Module2 [flow_indr_block_cb@flow_offload.c]; list((list)); end; cb_list --> list;
```











var@hv_utils_transport.c

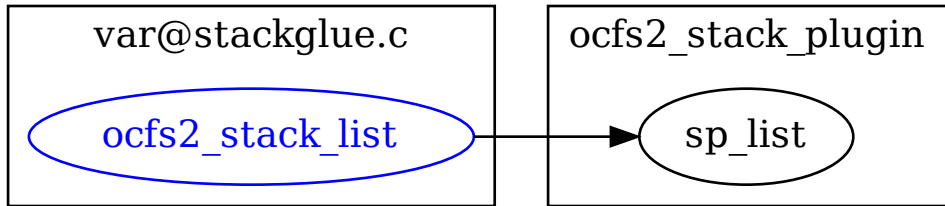
hvt_list

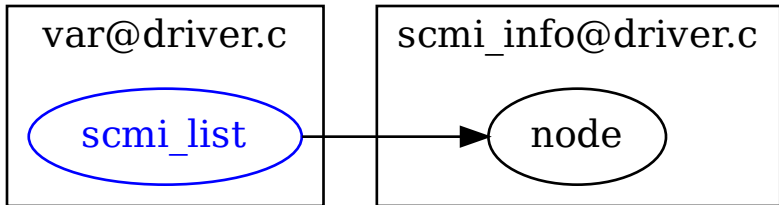
hvutil_transport

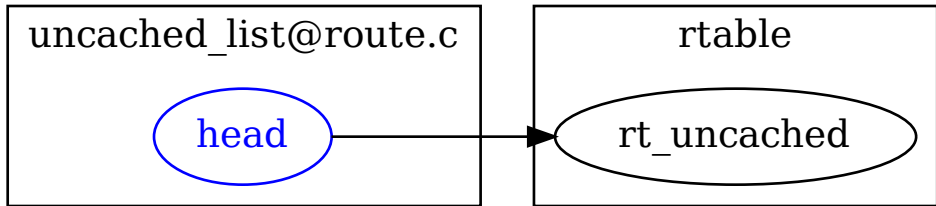
list

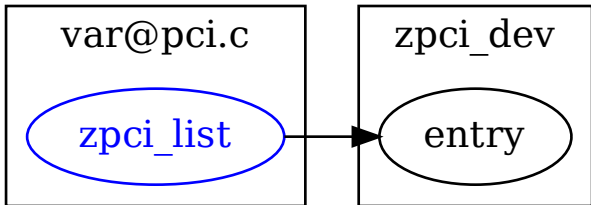
```
graph LR; subgraph Box1 [var@hv_utils_transport.c]; hvt_list((hvt_list)); end; subgraph Box2 [hvutil_transport]; list((list)); end; hvt_list --> list;
```

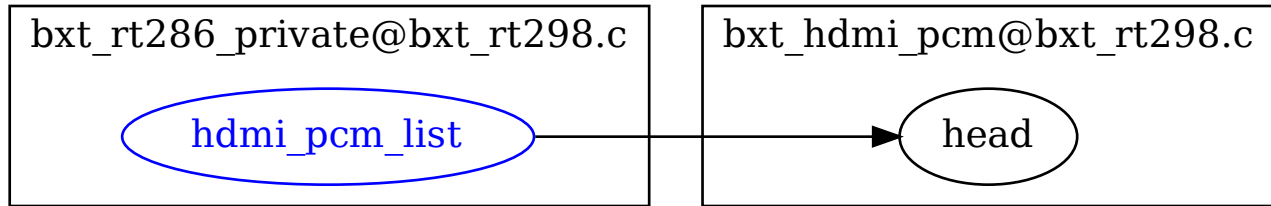
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@hv_utils_transport.c' contains an oval labeled 'hvt_list' with a blue border. On the right, a box labeled 'hvutil_transport' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'hvt_list' oval to the left side of the 'list' oval, indicating that 'hvt_list' points to 'list'.

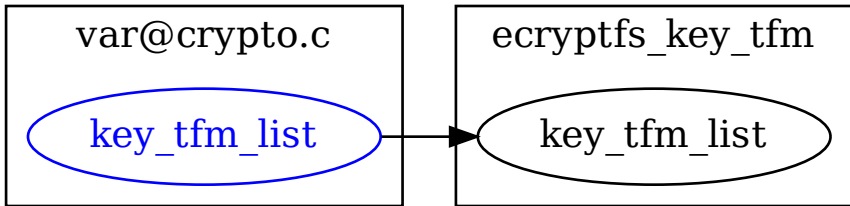


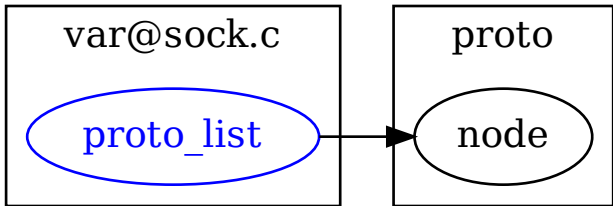


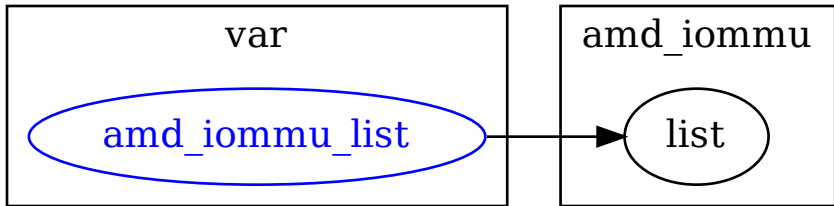


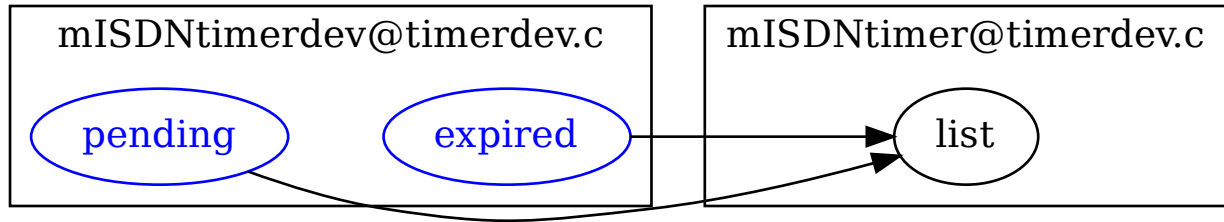


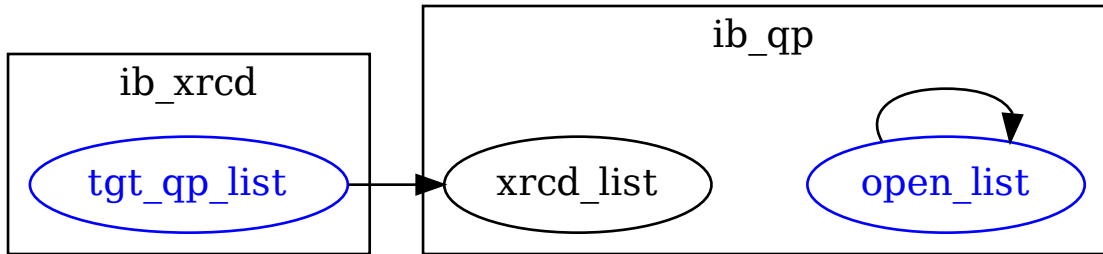


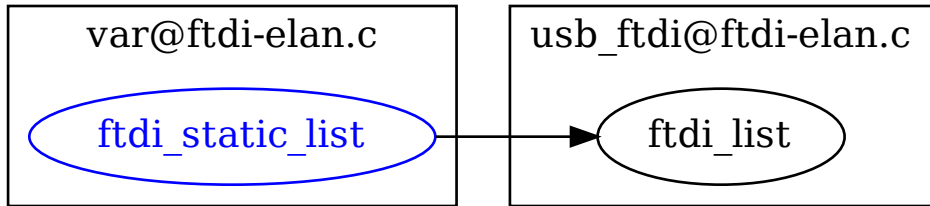












viommu_dev@virtio-iommu.c

requests

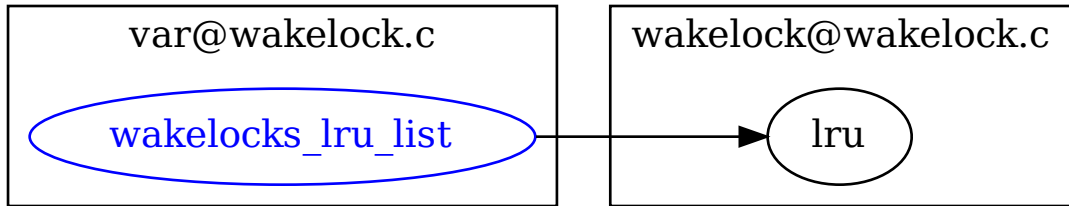
viommu_request@virtio-iommu.c

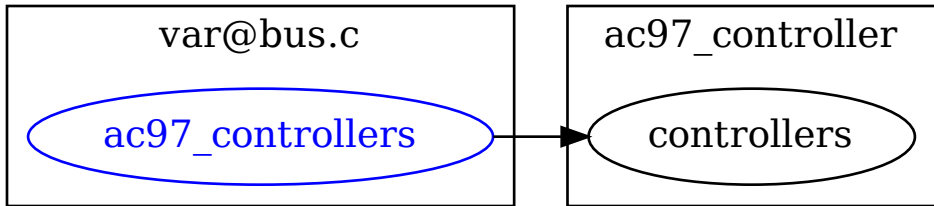
list

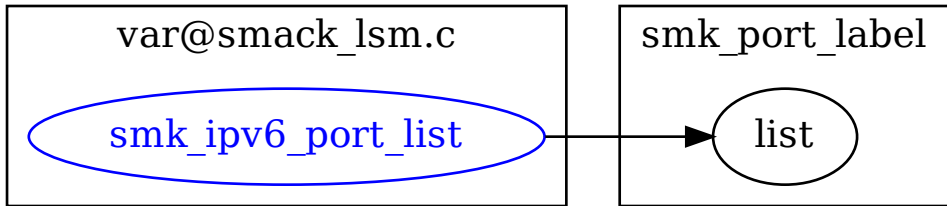


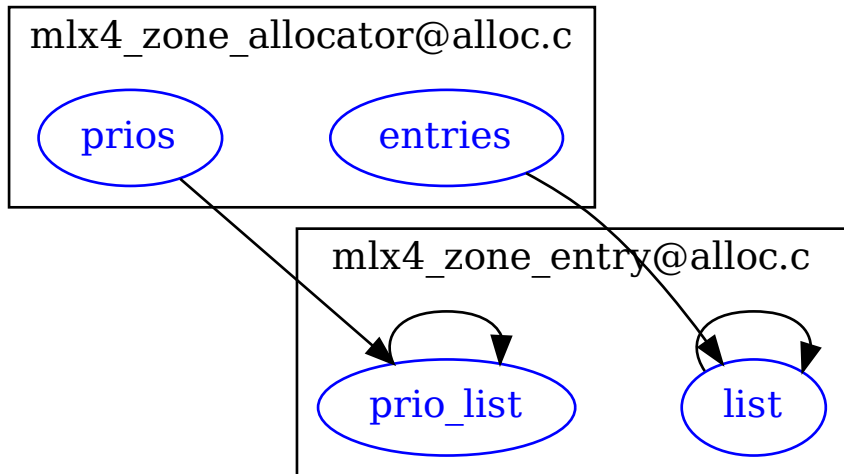
```
graph LR; A[requests] --> B(list);
```

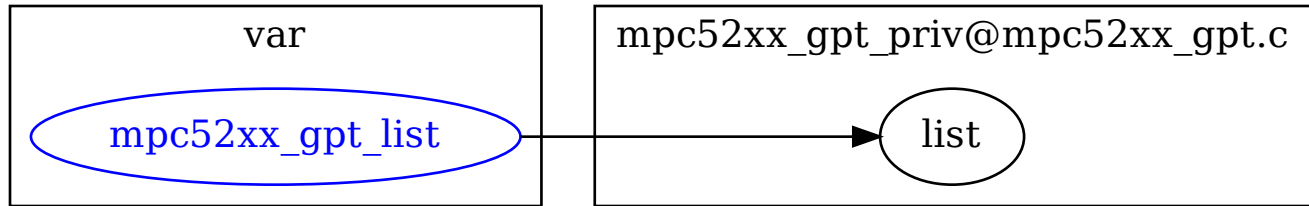
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'viommu_dev@virtio-iommu.c' contains an oval labeled 'requests' with a blue border. On the right, a rectangular box labeled 'viommu_request@virtio-iommu.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'requests' oval to the 'list' oval, indicating that the 'requests' variable points to the 'list' structure.

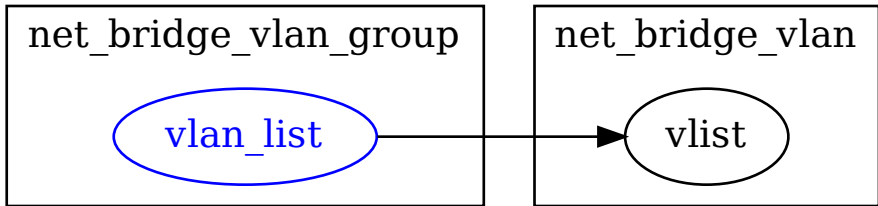


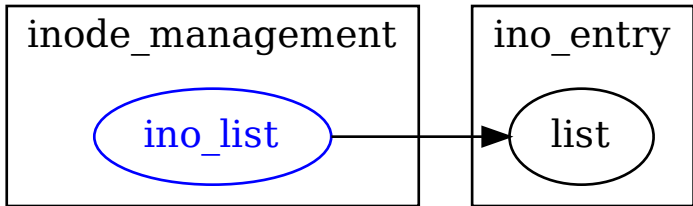












squashfs_stream@decompressor_multi.c

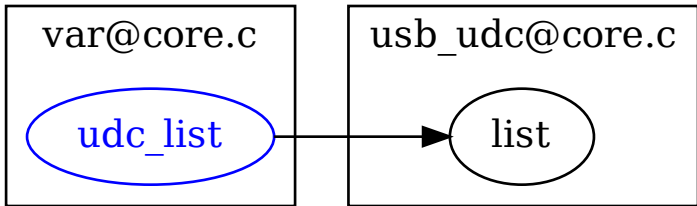
strm_list

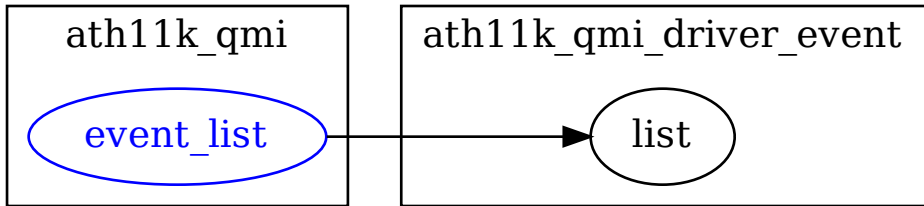
decomp_stream@decompressor_multi.c

list

```
graph LR; A[squashfs_stream@decompressor_multi.c] --> B[decomp_stream@decompressor_multi.c]; B --> C((list));
```

The diagram illustrates a pointer relationship between two modules. A horizontal line connects the two module boxes. An arrow points from the 'strm_list' variable in the first module to the 'list' variable in the second module.





var@industrialio-sw-device.c

iio_device_types_list

iio_sw_device_type

list

```
graph LR; A[iio_device_types_list] --> B((list));
```

The diagram illustrates a pointer relationship. On the left, a rectangular box contains the text 'var@industrialio-sw-device.c' and a blue oval containing 'iio_device_types_list'. On the right, another rectangular box contains the text 'iio_sw_device_type' and a black oval containing 'list'. A horizontal arrow points from the right side of the blue oval to the left side of the black oval, indicating that 'iio_device_types_list' points to 'list'.

s2255_vc@s2255drv.c

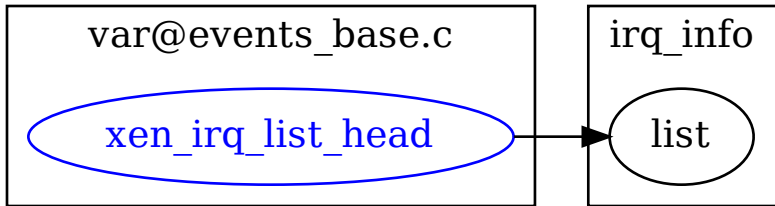
buf_list

s2255_buffer@s2255drv.c

list

```
graph LR; subgraph Box1 [s2255_vc@s2255drv.c]; BL(buf_list); end; subgraph Box2 [s2255_buffer@s2255drv.c]; L(list); end; BL --> L;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 's2255_vc@s2255drv.c' contains an oval labeled 'buf_list' in blue text. On the right, a rectangular box labeled 's2255_buffer@s2255drv.c' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'buf_list' oval to the left side of the 'list' oval, indicating that 'buf_list' points to 'list'.



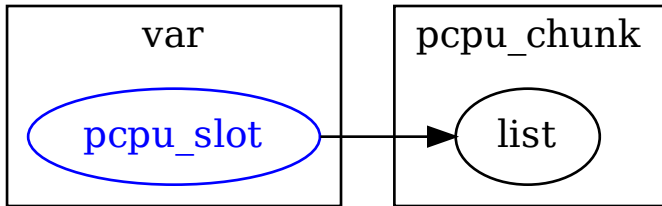
var@acpi-dma.c

acpi_dma_list

acpi_dma

dma_controllers





var@ccp-crypto-main.c

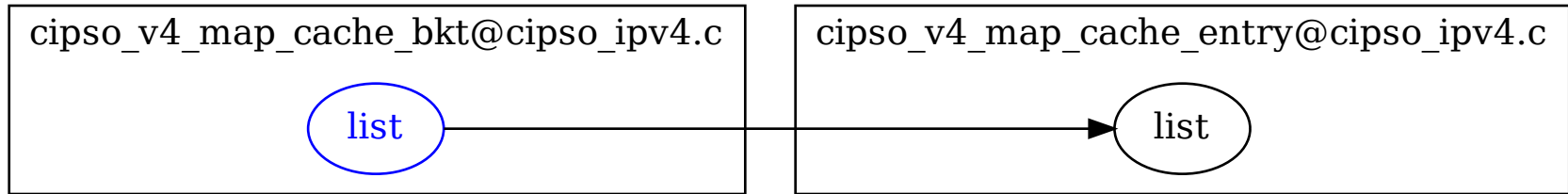
akcipher_algs

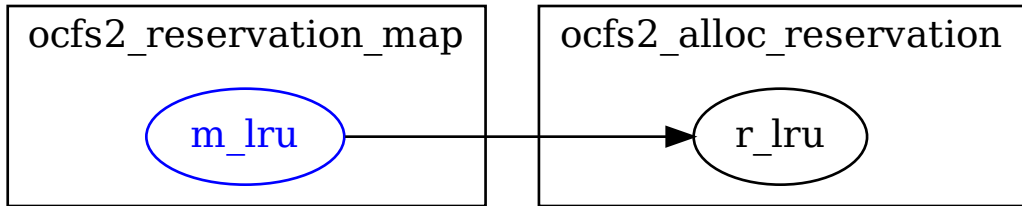
ccp_crypto_akcipher_alg

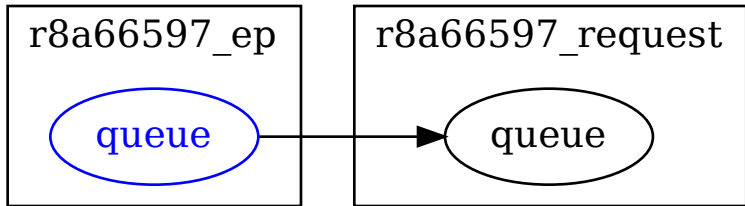
entry

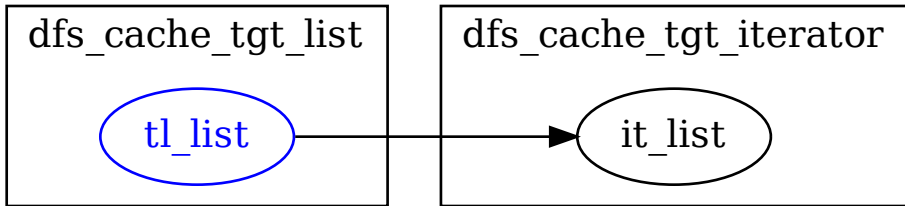
```
graph LR; A[akcipher_algs] --> B[entry];
```

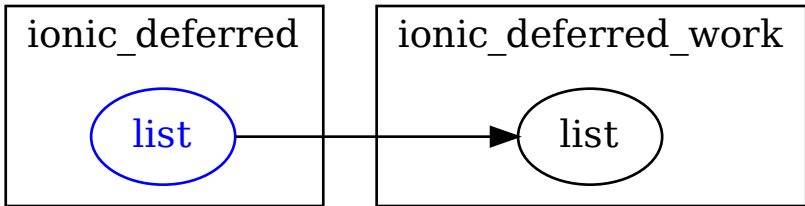
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@ccp-crypto-main.c' at the top and 'akcipher_algs' in a blue oval below it. On the right, another rectangular box contains the text 'ccp_crypto_akcipher_alg' at the top and 'entry' in a black oval below it. A horizontal arrow points from the right side of the 'akcipher_algs' oval to the left side of the 'entry' oval, indicating that 'akcipher_algs' points to 'entry'.











var@virtio_console.c

pending_free_dma_bufs

port_buffer@virtio_console.c

list

```
graph LR; subgraph Box1 [ ]; var1[var@virtio_console.c]; pfdma(pending_free_dma_bufs); end; subgraph Box2 [ ]; port_buffer[port_buffer@virtio_console.c]; list(list); end; pfdma --> list;
```

elf_note_info@binfmt_elf.c

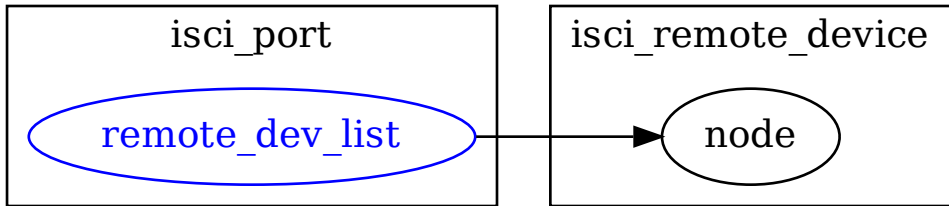
thread_list

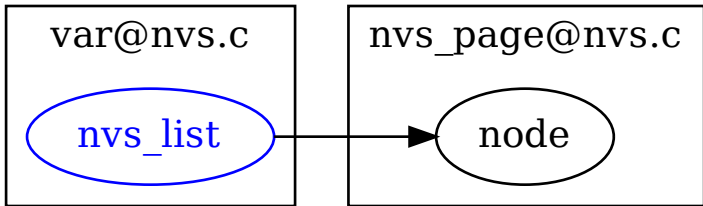
elf_thread_status@binfmt_elf.c

list

```
graph LR; A[elf_note_info@binfmt_elf.c] --- B[elf_thread_status@binfmt_elf.c]; subgraph A; C((thread_list)); end; subgraph B; D((list)); end; C --> D;
```

The diagram consists of two rectangular boxes. The left box contains the text 'elf_note_info@binfmt_elf.c' at the top and an oval containing 'thread_list' in blue text below it. The right box contains the text 'elf_thread_status@binfmt_elf.c' at the top and an oval containing 'list' in black text below it. A horizontal line connects the right side of the left box to the left side of the right box. An arrow points from the 'thread_list' oval in the left box to the 'list' oval in the right box.





var@net_namespace.c

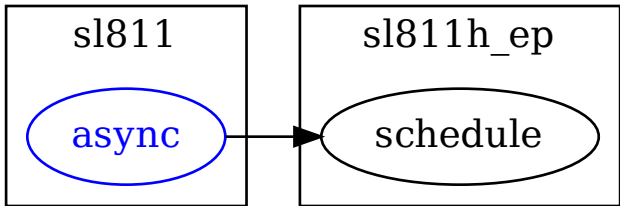
pernet_list

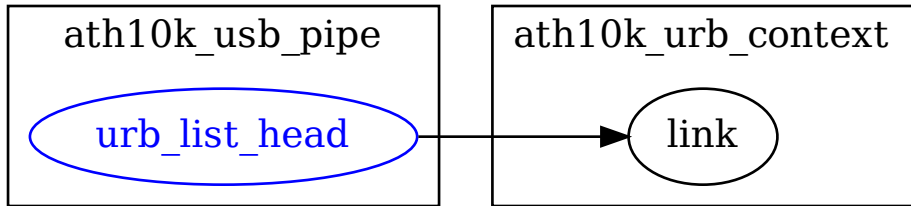
pernet_operations

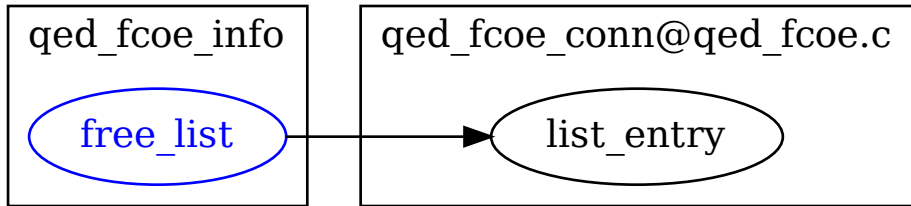
list

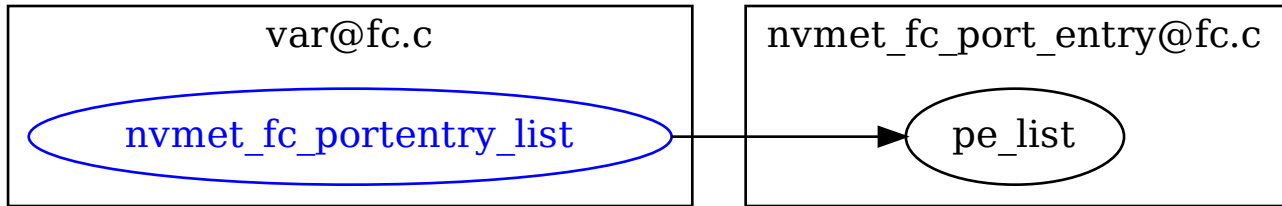
```
graph LR; subgraph A [var@net_namespace.c]; direction TB; A1(pernet_list); end; subgraph B [pernet_operations]; direction TB; B1(list); end; A1 --> B1;
```

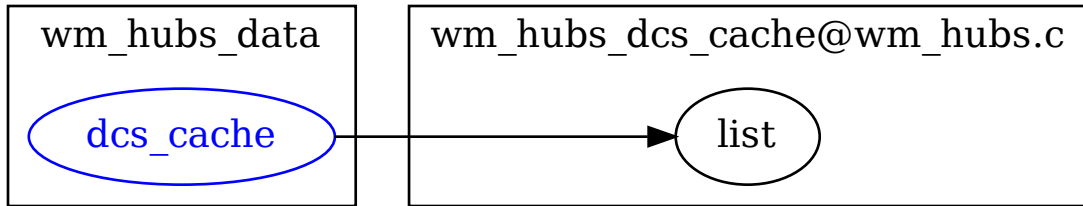
The diagram consists of two rectangular boxes. The left box is titled 'var@net_namespace.c' and contains an oval labeled 'pernet_list' in blue text. The right box is titled 'pernet_operations' and contains an oval labeled 'list' in black text. A horizontal arrow points from the right side of the 'pernet_list' oval to the left side of the 'list' oval, indicating a pointer relationship.











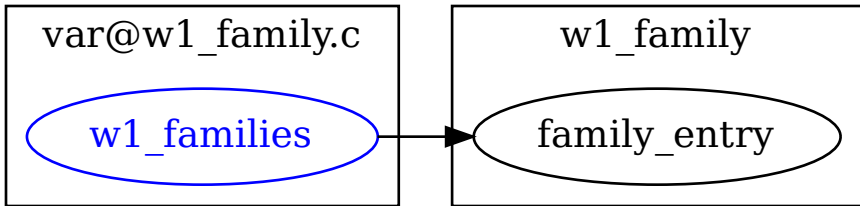
irq_info@8250_core.c

head

uart_8250_port

list





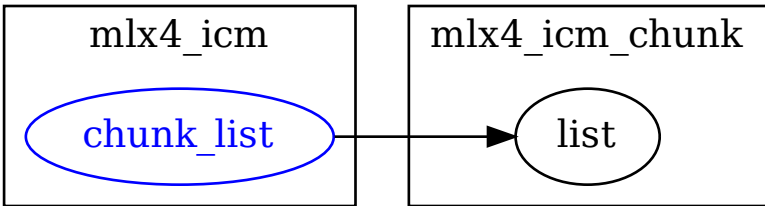
vmw_legacy_display@vmwgfx_ldu.c

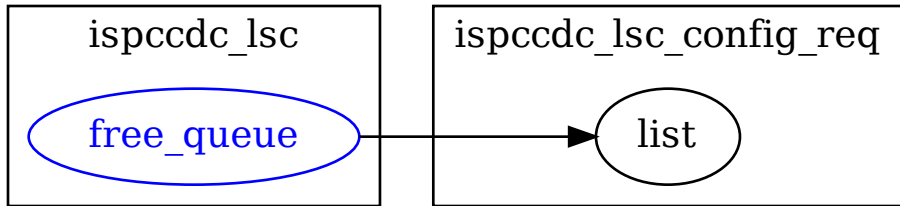
active

vmw_legacy_display_unit@vmwgfx_ldu.c

active









var@xt_LED.c

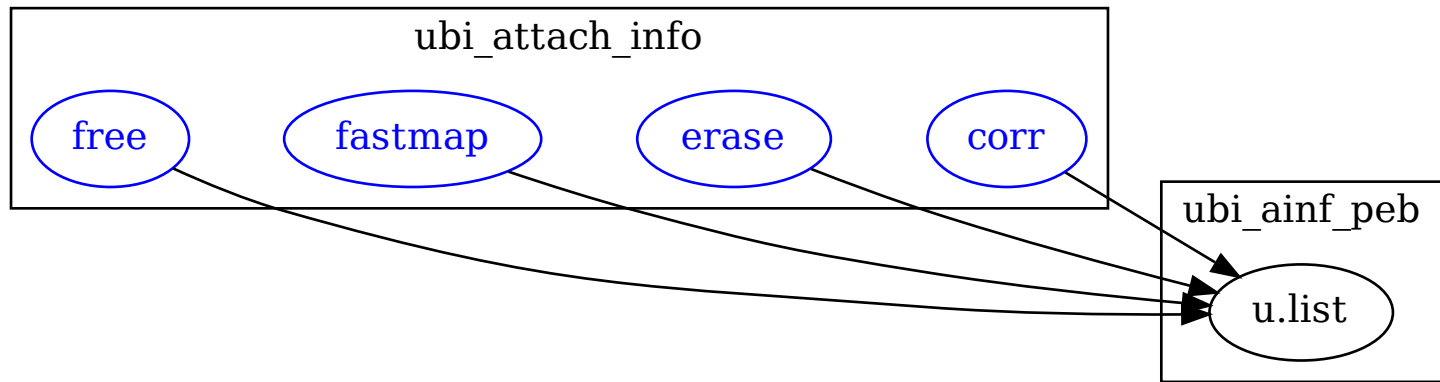
xt_led_triggers

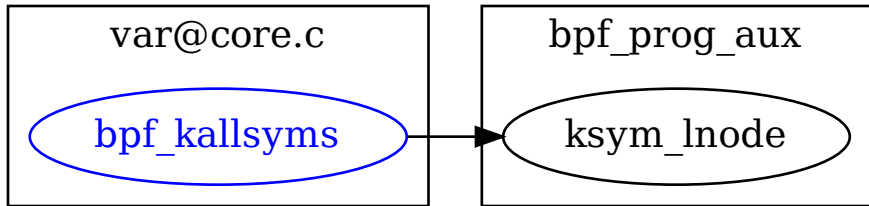
xt_led_info_internal@xt_LED.c

list

```
graph LR; subgraph File1 [var@xt_LED.c]; A([xt_led_triggers]); end; subgraph File2 [xt_led_info_internal@xt_LED.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@xt_LED.c`. Inside this box, the variable `xt_led_triggers` is enclosed in a blue oval. On the right, another rectangular box represents the source file `xt_led_info_internal@xt_LED.c`. Inside this box, the variable `list` is enclosed in a black oval. A horizontal arrow points from the right side of the `xt_led_triggers` oval to the left side of the `list` oval, indicating that `xt_led_triggers` points to `list`.





rio_channel@rio_cm.c

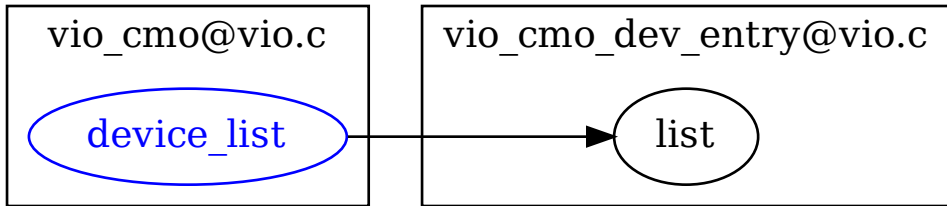
accept_queue

conn_req@rio_cm.c

node

```
graph LR; subgraph "rio_channel@rio_cm.c"; accept_queue; end; subgraph "conn_req@rio_cm.c"; node; end; accept_queue --> node;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'rio_channel@rio_cm.c' contains an oval labeled 'accept_queue' with a blue border. On the right, a rectangular box labeled 'conn_req@rio_cm.c' contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'accept_queue' oval to the left side of the 'node' oval, indicating that 'accept_queue' points to 'node'.



iio_hw_consumer@industrialio-hw-consumer.c

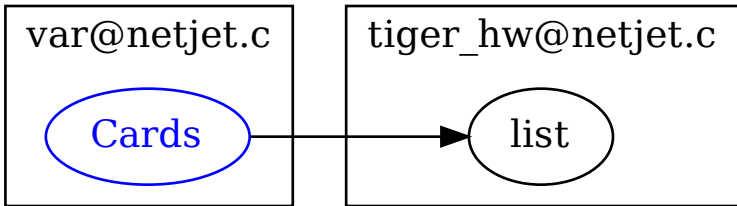
buffers

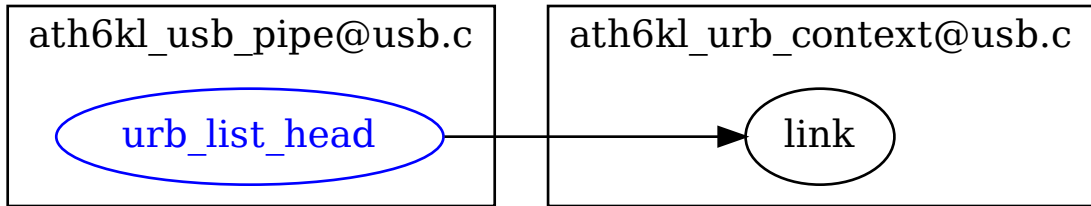
hw_consumer_buffer@industrialio-hw-consumer.c

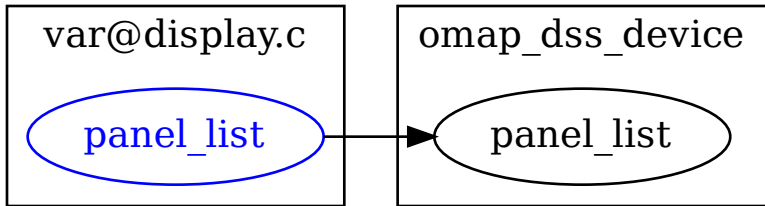
head

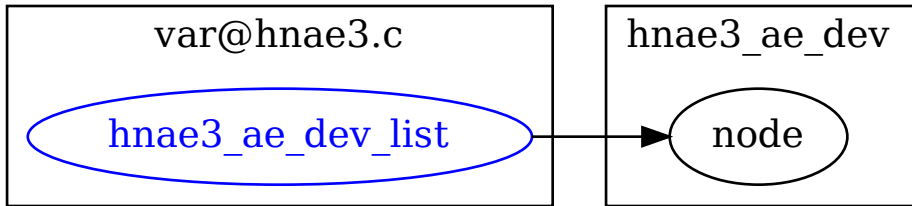


```
graph LR; subgraph Module1 [iio_hw_consumer@industrialio-hw-consumer.c]; buffers; end; subgraph Module2 [hw_consumer_buffer@industrialio-hw-consumer.c]; head; end; buffers --> head;
```









var@intel_menlow.c

intel_menlow_attr_list

intel_menlow_attribute@intel_menlow.c

node

```
graph LR; A["var@intel_menlow.c  
intel_menlow_attr_list"] --> B["intel_menlow_attribute@intel_menlow.c  
node"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@intel_menlow.c' at the top and 'intel_menlow_attr_list' at the bottom, which is enclosed in a blue oval. The right box contains the text 'intel_menlow_attribute@intel_menlow.c' at the top and 'node' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.

airspy@airspy.c

queued_bufs

airspy_frame_buf@airspy.c

list



```
graph LR; subgraph "airspy@airspy.c"; QB([queued_bufs]); end; subgraph "airspy_frame_buf@airspy.c"; L([list]); end; QB --> L;
```

The diagram illustrates a pointer relationship between two memory spaces. On the left, a rectangular box labeled 'airspy@airspy.c' contains an oval labeled 'queued_bufs' in blue text. On the right, a rectangular box labeled 'airspy_frame_buf@airspy.c' contains an oval labeled 'list'. A horizontal arrow points from the 'queued_bufs' oval to the 'list' oval, indicating that 'queued_bufs' points to the 'list' structure.

mlxsw_sp_nexthop_group@spectrum_router.c

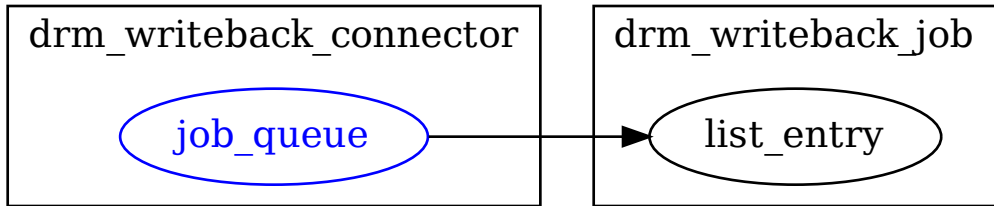
fib_list

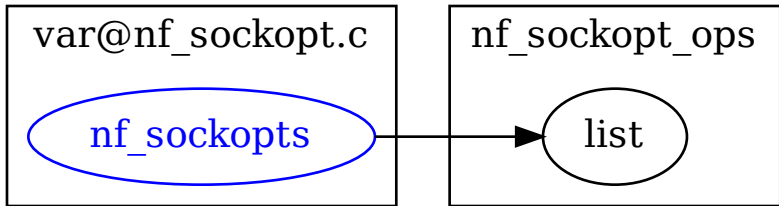
mlxsw_sp_fib_entry@spectrum_router.c

nexthop_group_node

```
graph LR; A[mlxsw_sp_nexthop_group@spectrum_router.c] --> B(mlxsw_sp_fib_entry@spectrum_router.c); B --> C(nexthop_group_node);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box represents the `mlxsw_sp_nexthop_group@spectrum_router.c` structure, which contains a blue oval labeled `fib_list`. On the right, another rectangular box represents the `mlxsw_sp_fib_entry@spectrum_router.c` structure, which contains a black oval labeled `nexthop_group_node`. A horizontal arrow points from the `fib_list` oval to the `nexthop_group_node` oval, indicating that `fib_list` points to the `nexthop_group_node`.





mwl8k_priv@mwl8k.c

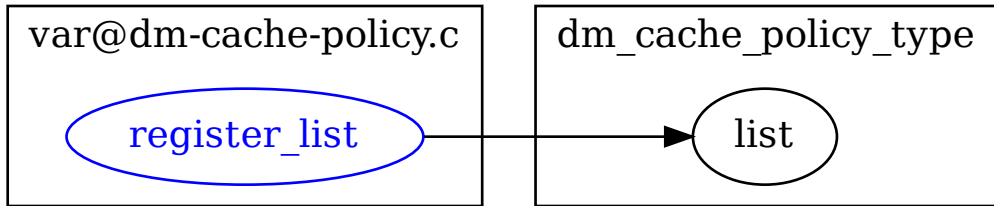
vif_list

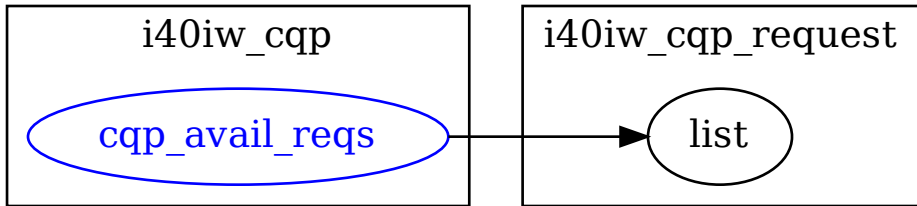
mwl8k_vif@mwl8k.c

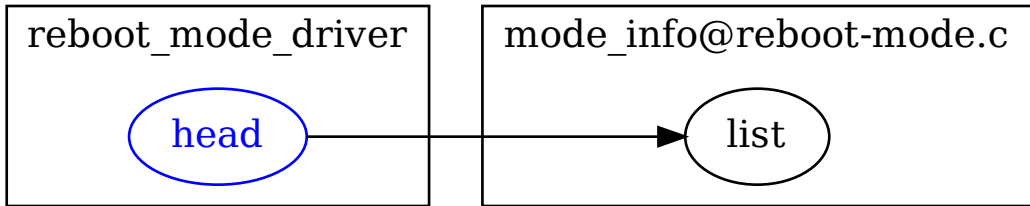
list

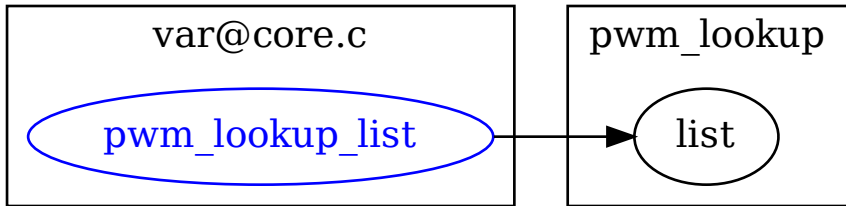
```
graph LR; subgraph mwl8k_priv [mwl8k_priv@mwl8k.c]; vif_list((vif_list)); end; subgraph mwl8k_vif [mwl8k_vif@mwl8k.c]; list((list)); end; vif_list --> list;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'mwl8k_priv@mwl8k.c' contains an oval labeled 'vif_list'. On the right, a box labeled 'mwl8k_vif@mwl8k.c' contains an oval labeled 'list'. A horizontal arrow points from the 'vif_list' oval to the 'list' oval, indicating that 'vif_list' points to 'list'.









vgasr_priv@vga_switcheroo.c

clients

vga_switcheroo_client@vga_switcheroo.c

list

```
graph LR; subgraph Box1 [vgasr_priv@vga_switcheroo.c]; clients; end; subgraph Box2 [vga_switcheroo_client@vga_switcheroo.c]; list; end; clients --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'vgasr_priv@vga_switcheroo.c' contains a blue oval labeled 'clients'. On the right, a box labeled 'vga_switcheroo_client@vga_switcheroo.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'clients' oval to the 'list' oval, indicating that the 'clients' variable in the first module points to the 'list' variable in the second module.

var@mtk_iommu.c

m4ulist

mtk_iommu_data

list

```
graph LR; subgraph Box1 [var@mtk_iommu.c]; m4ulist; end; subgraph Box2 [mtk_iommu_data]; list; end; m4ulist --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@mtk_iommu.c' and contains an oval labeled 'm4ulist' in blue text. The right box is labeled 'mtk_iommu_data' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'm4ulist' oval to the left side of the 'list' oval, indicating a pointer relationship.

var@rcar-fcp.c

fcp_devices

rcar_fcp_device@rcar-fcp.c

list

```
graph LR; subgraph Box1 [var@rcar-fcp.c]; fcp_devices; end; subgraph Box2 [rcar_fcp_device@rcar-fcp.c]; list; end; fcp_devices --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@rcar-fcp.c' and contains an oval labeled 'fcp_devices' in blue text. The right box is labeled 'rcar_fcp_device@rcar-fcp.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'fcp_devices' oval to the 'list' oval, indicating a pointer relationship.

var@qcom_sysmon.c

sysmon_list

qcom_sysmon@qcom_sysmon.c

node

```
graph LR; subgraph var ["var@qcom_sysmon.c"]; sysmon_list; end; subgraph qcom_sysmon ["qcom_sysmon@qcom_sysmon.c"]; node; end; sysmon_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@qcom_sysmon.c' contains an oval labeled 'sysmon_list'. On the right, a box labeled 'qcom_sysmon@qcom_sysmon.c' contains an oval labeled 'node'. A horizontal arrow points from the 'sysmon_list' oval to the 'node' oval, indicating that 'sysmon_list' points to 'node'.

var@qdio_debug.c

qdio_dbf_list

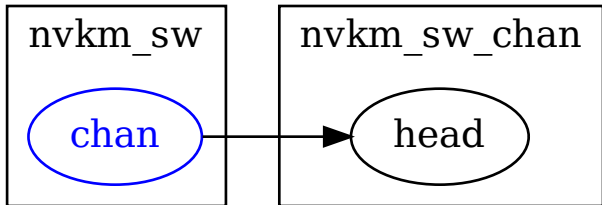
qdio_dbf_entry@qdio_debug.c

dbf_list

```
graph LR; A["var@qdio_debug.c  
qdio_dbf_list"] --> B["qdio_dbf_entry@qdio_debug.c  
dbf_list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@qdio_debug.c' contains an oval labeled 'qdio_dbf_list' with a blue border. On the right, a box labeled 'qdio_dbf_entry@qdio_debug.c' contains an oval labeled 'dbf_list' with a black border. A horizontal arrow points from the 'qdio_dbf_list' oval to the 'dbf_list' oval, indicating that the variable 'qdio_dbf_list' points to the variable 'dbf_list'.





var@mmp-driver.c

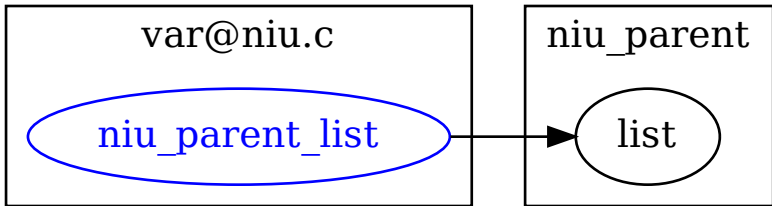
mmpcam_devices

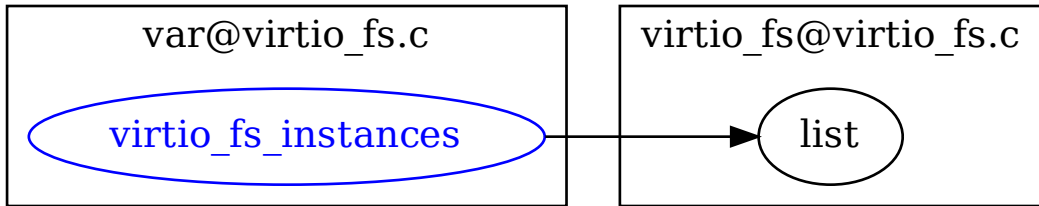
mmp_camera@mmp-driver.c

devlist

```
graph LR; subgraph Box1 [var@mmp-driver.c]; mmpcam_devices; end; subgraph Box2 [mmp_camera@mmp-driver.c]; devlist; end; mmpcam_devices --> devlist;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@mmp-driver.c' and contains an oval labeled 'mmpcam_devices' with a blue border. The right box is labeled 'mmp_camera@mmp-driver.c' and contains an oval labeled 'devlist' with a black border. A horizontal arrow points from the right side of the 'mmpcam_devices' oval to the left side of the 'devlist' oval, indicating a pointer or reference relationship.





var@hci_intel.c

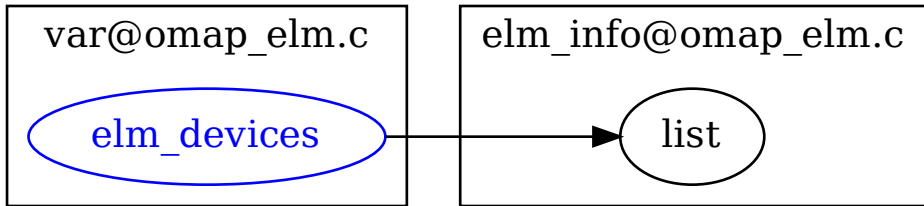
intel_device_list

intel_device@hci_intel.c

list

```
graph LR; A["var@hci_intel.c  
intel_device_list"] --> B["intel_device@hci_intel.c  
list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@hci_intel.c' and a blue oval containing 'intel_device_list'. On the right, another rectangular box contains the text 'intel_device@hci_intel.c' and a black oval containing 'list'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that 'intel_device_list' points to 'list'.



var@irq-bcm7038-l1.c

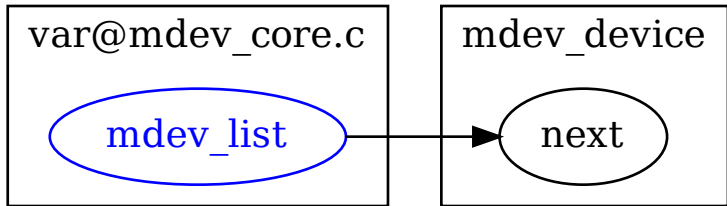
bcm7038_l1_intcs_list

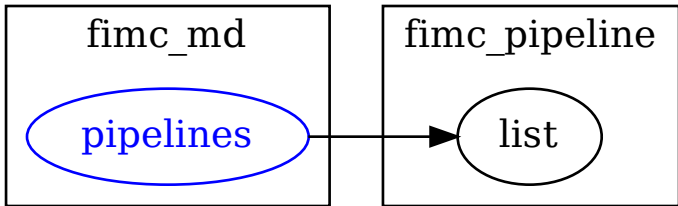
bcm7038_l1_chip@irq-bcm7038-l1.c

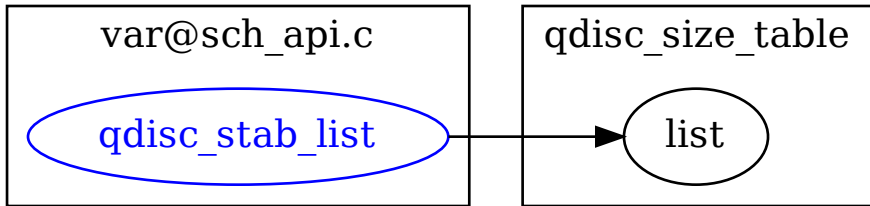
list

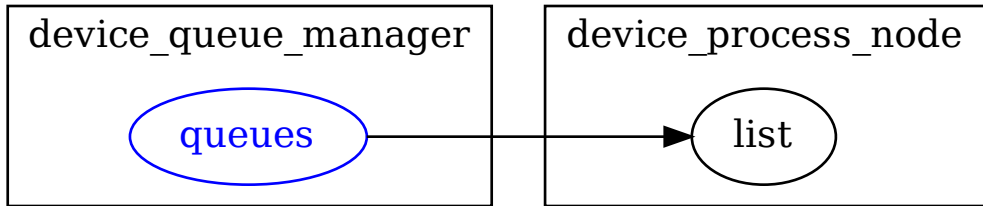
```
graph LR; A["var@irq-bcm7038-l1.c  
bcm7038_l1_intcs_list"] --> B["bcm7038_l1_chip@irq-bcm7038-l1.c  
list"]
```

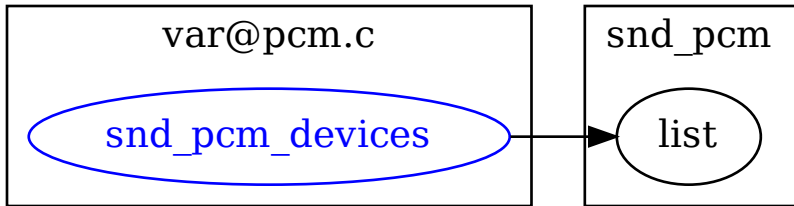
The diagram consists of two rectangular boxes. The left box contains the text 'var@irq-bcm7038-l1.c' at the top and 'bcm7038_l1_intcs_list' at the bottom, which is enclosed in a blue oval. The right box contains the text 'bcm7038_l1_chip@irq-bcm7038-l1.c' at the top and 'list' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the blue oval in the left box to the black oval in the right box.

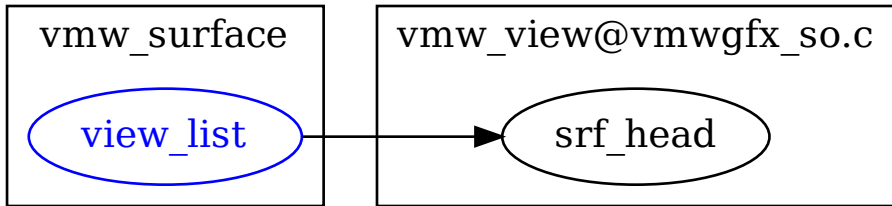












var@ecryptfs_parse_packet_set

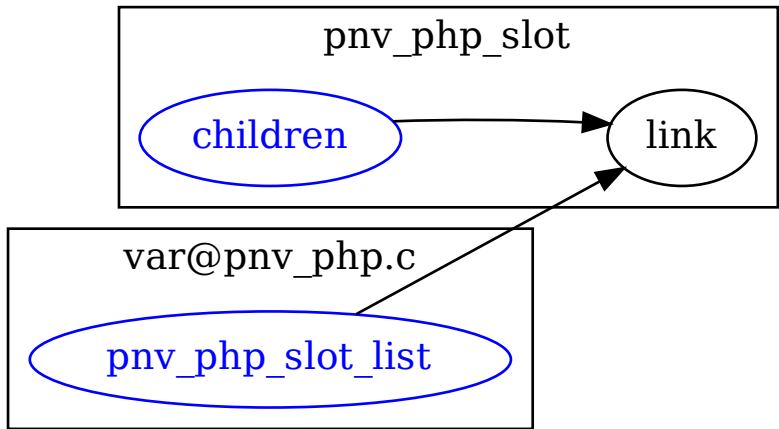
auth_tok_list

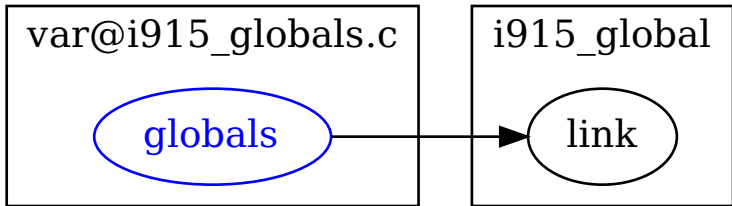
ecryptfs_auth_tok_list_item

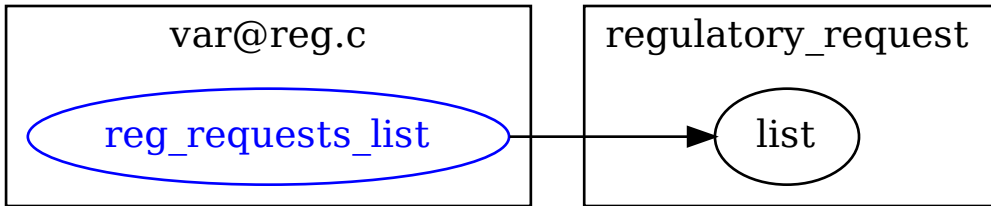
list

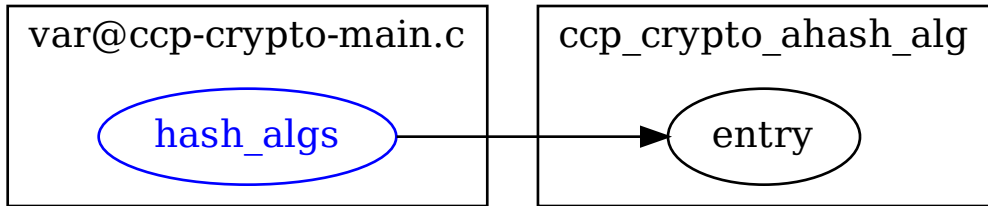
```
graph LR; A(auth_tok_list) --> B(list);
```

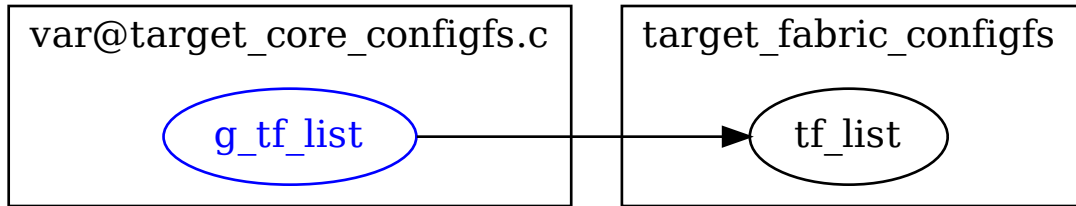
The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'var@ecryptfs_parse_packet_set' contains an oval labeled 'auth_tok_list' with a blue border. On the right, a rectangular box labeled 'ecryptfs_auth_tok_list_item' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'auth_tok_list' oval to the left side of the 'list' oval, indicating that 'auth_tok_list' points to the 'list' member of 'ecryptfs_auth_tok_list_item'.

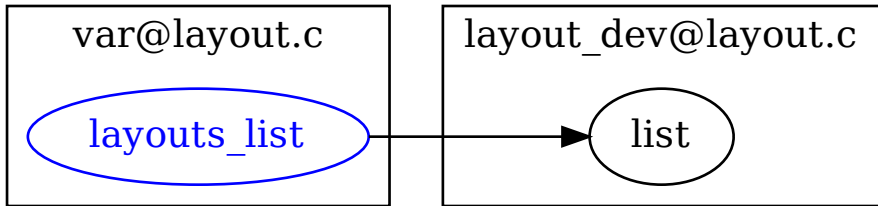


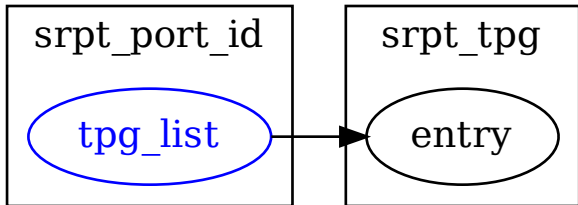












netlink_tap_net@af_netlink.c

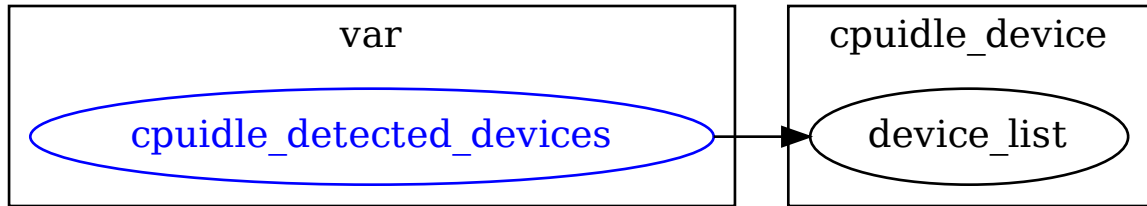
netlink_tap_all

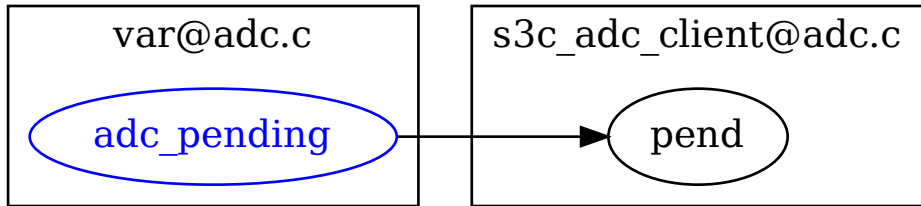
netlink_tap

list

```
graph LR; A[netlink_tap_all] --> B((list));
```

The diagram consists of two rectangular boxes. The left box contains the text 'netlink_tap_net@af_netlink.c' at the top and an oval containing 'netlink_tap_all' in blue text below it. The right box contains the text 'netlink_tap' at the top and an oval containing 'list' below it. A horizontal arrow points from the right side of the 'netlink_tap_all' oval to the left side of the 'list' oval.





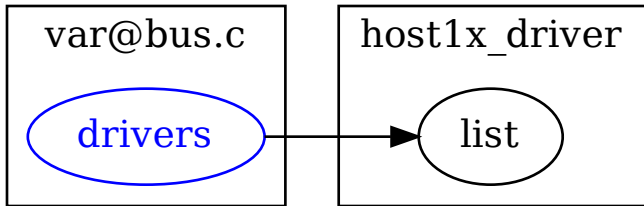
card_private@sof_da7219_max98373.c

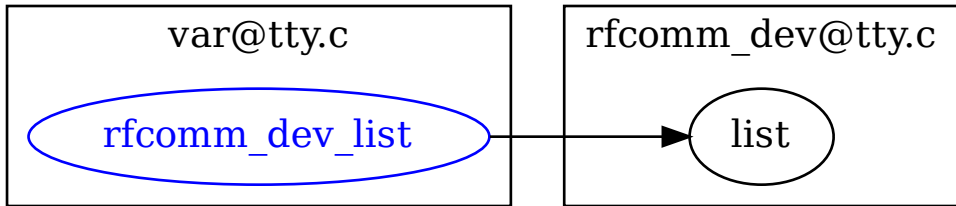
hdmi_pcm_list

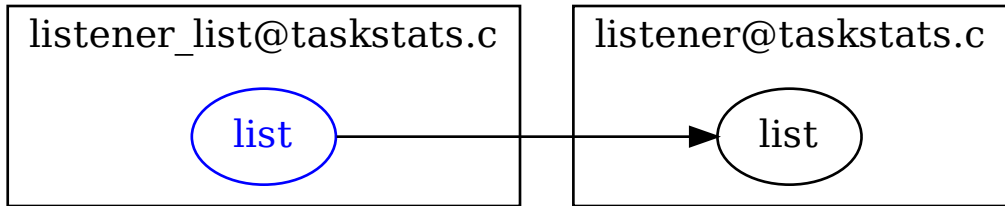
hdmi_pcm@sof_da7219_max98373.c

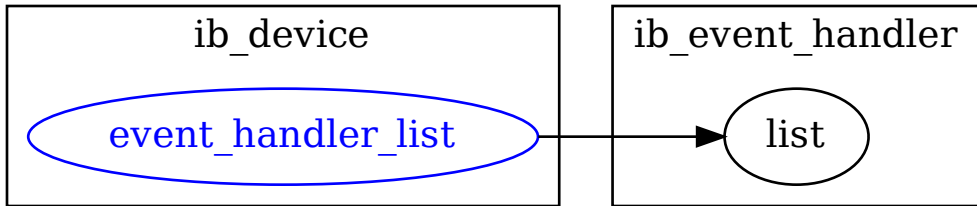
head

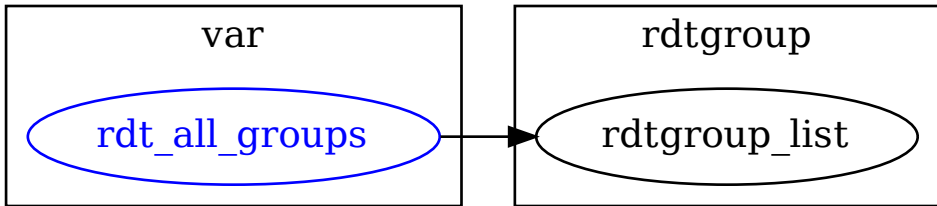


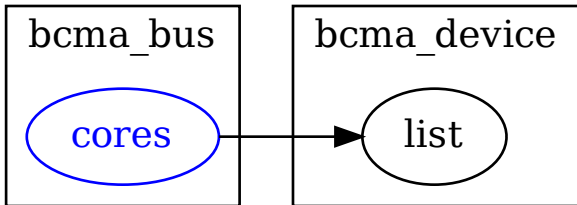


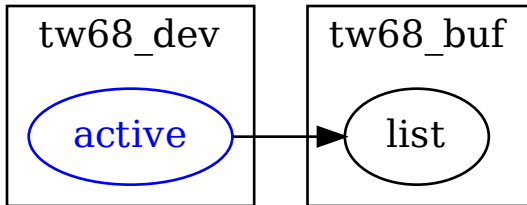












nsim_fib6_rt@fib.c

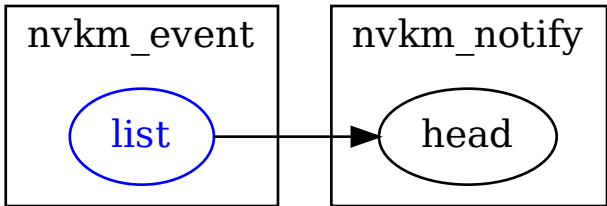
nh_list

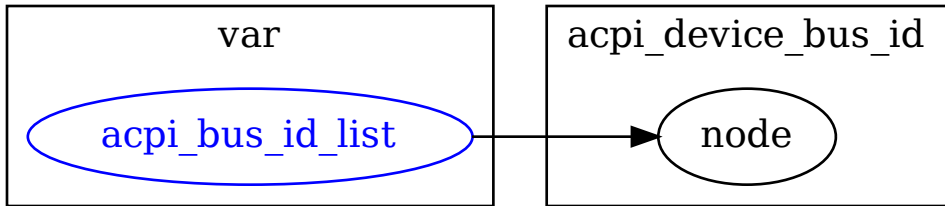
nsim_fib6_rt_nh@fib.c

list

```
graph LR; A[nsim_fib6_rt@fib.c] --> B[nsim_fib6_rt_nh@fib.c]; subgraph A; C([nh_list]); end; subgraph B; D([list]); end; C --> D;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'nsim_fib6_rt@fib.c' contains an oval labeled 'nh_list'. On the right, a box labeled 'nsim_fib6_rt_nh@fib.c' contains an oval labeled 'list'. A horizontal arrow points from the 'nh_list' oval to the 'list' oval, indicating that the 'nh_list' variable in the first module points to the 'list' variable in the second module.





var@stack_user.c

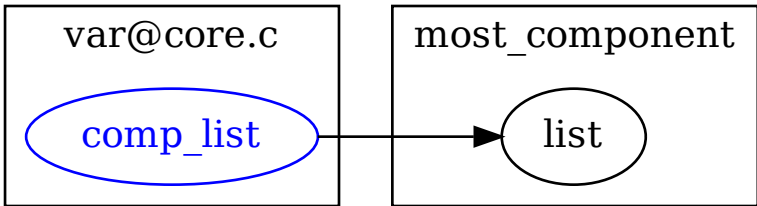
ocfs2_control_private_list

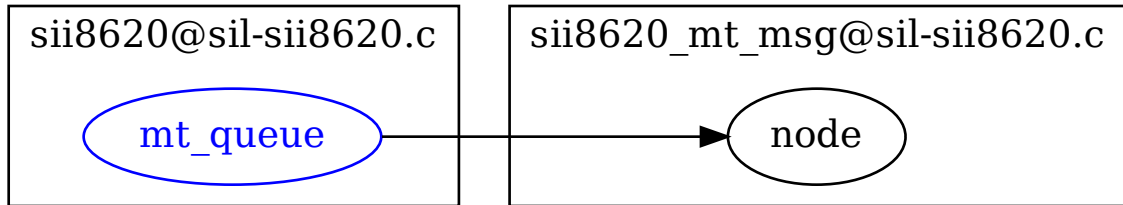
ocfs2_control_private@stack_user.c

op_list

```
graph LR; A[ocfs2_control_private_list] --> B((op_list))
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@stack_user.c'. Inside this box, the variable 'ocfs2_control_private_list' is highlighted with a blue oval. On the right, another rectangular box represents the source file 'ocfs2_control_private@stack_user.c'. Inside this box, the variable 'op_list' is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that 'ocfs2_control_private_list' points to 'op_list'.





var@devcom.c

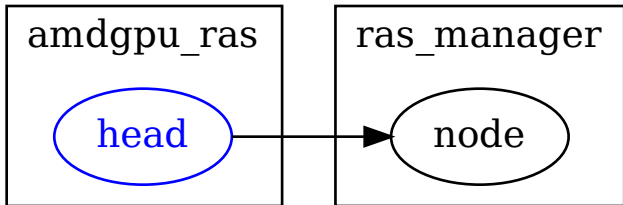
devcom_list

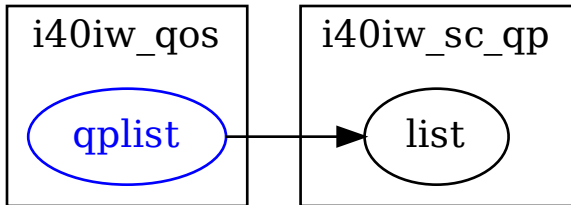
mlx5_devcom_list@devcom.c

list

```
graph LR; subgraph Module1 [var@devcom.c]; devcom_list; end; subgraph Module2 [mlx5_devcom_list@devcom.c]; list; end; devcom_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@devcom.c' contains an oval labeled 'devcom_list'. On the right, a box labeled 'mlx5_devcom_list@devcom.c' contains an oval labeled 'list'. A horizontal arrow points from the 'devcom_list' oval to the 'list' oval, indicating that the 'devcom_list' variable points to the 'list' structure.





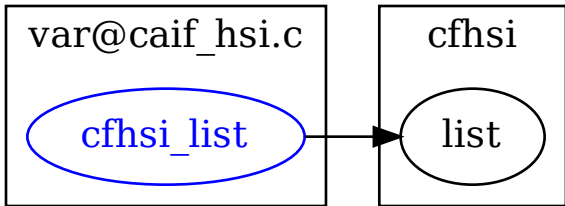
match_list_head@fs_core.c

list

match_list@fs_core.c

list





var@hci_bcm.c

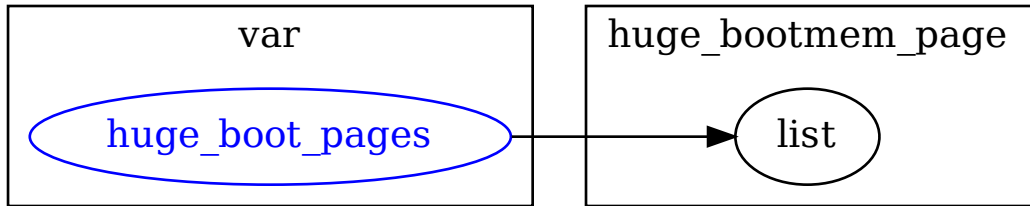
bcm_device_list

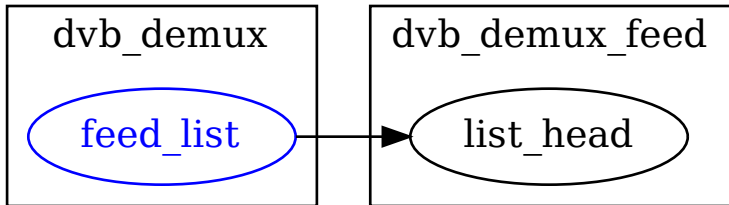
bcm_device@hci_bcm.c

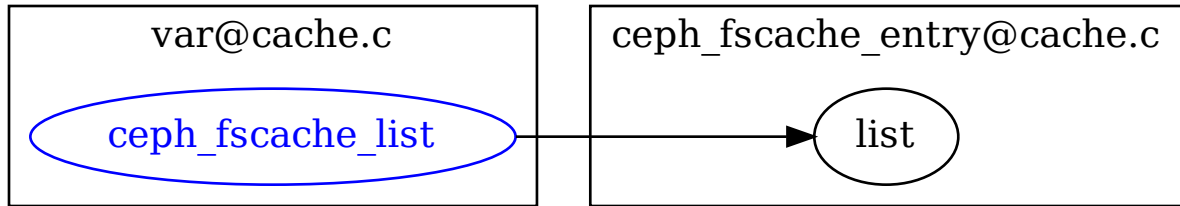
list

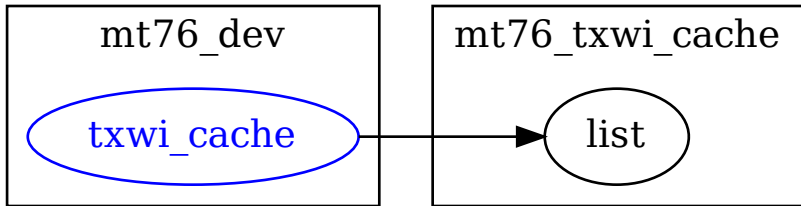
```
graph LR; subgraph Scope1 [var@hci_bcm.c]; A(bcm_device_list); end; subgraph Scope2 [bcm_device@hci_bcm.c]; B((list)); end; A --> B;
```

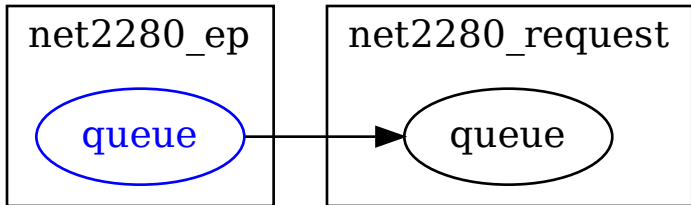
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the scope `var@hci_bcm.c`. Inside this box, the variable `bcm_device_list` is shown and highlighted with a blue oval. On the right, another rectangular box represents the scope `bcm_device@hci_bcm.c`. Inside this box, there is an oval labeled `list`. A horizontal arrow points from the right side of the `bcm_device_list` oval to the left side of the `list` oval, indicating that `bcm_device_list` points to the `list` variable.











var@probe.c

pci_domain_busn_res_list

pci_domain_busn_res@probe.c

list

```
graph LR; subgraph File1 [var@probe.c]; A([pci_domain_busn_res_list]); end; subgraph File2 [pci_domain_busn_res@probe.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file 'var@probe.c'. Inside this box, the variable 'pci_domain_busn_res_list' is enclosed in a blue oval. On the right, another rectangular box represents the source file 'pci_domain_busn_res@probe.c'. Inside this box, the variable 'list' is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that 'pci_domain_busn_res_list' points to 'list'.

var@qeth_core_main.c

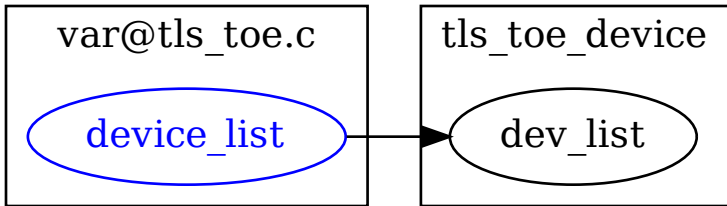
qeth_dbf_list

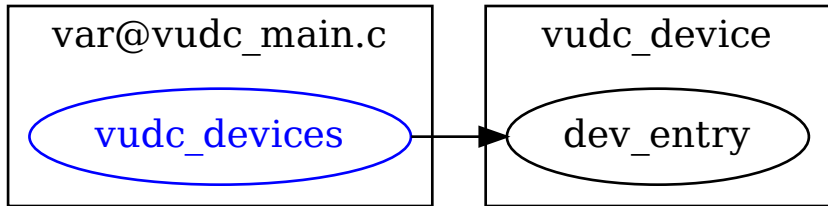
qeth_dbf_entry@qeth_core_main.c

dbf_list

```
graph LR; subgraph File1 [var@qeth_core_main.c]; A([qeth_dbf_list]); end; subgraph File2 [qeth_dbf_entry@qeth_core_main.c]; B([dbf_list]); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables defined in the same source file, `qeth_core_main.c`. On the left, the variable `var@qeth_core_main.c` contains a pointer to `qeth_dbf_list`. On the right, the variable `qeth_dbf_entry@qeth_core_main.c` contains a pointer to `dbf_list`. A horizontal arrow points from the `qeth_dbf_list` oval in the first box to the `dbf_list` oval in the second box, indicating that `qeth_dbf_list` points to the same memory location as `dbf_list`.





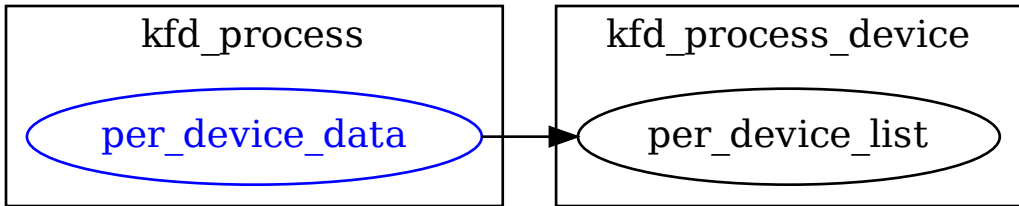
var@tfc_conf.c

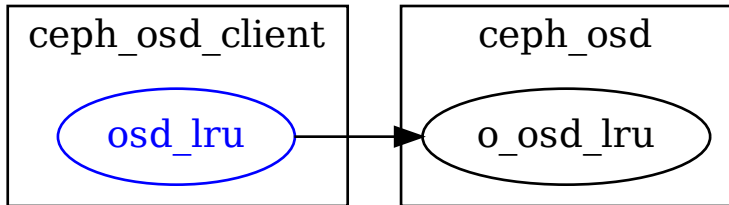
ft_wwn_list

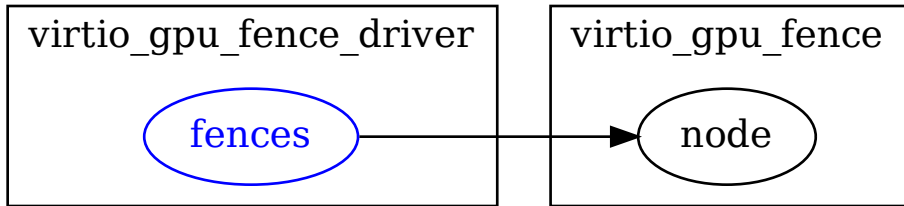
ft_lport_wwn

ft_wwn_node









scrub_parity@scrub.c

spages

scrub_page@scrub.c

list



```
graph LR; subgraph "scrub_parity@scrub.c"; spages; end; subgraph "scrub_page@scrub.c"; list; end; spages --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'scrub_parity@scrub.c' and contains a blue oval labeled 'spages'. The right box is labeled 'scrub_page@scrub.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'spages' oval to the left side of the 'list' oval.

var@dell-wmi-descriptor.c

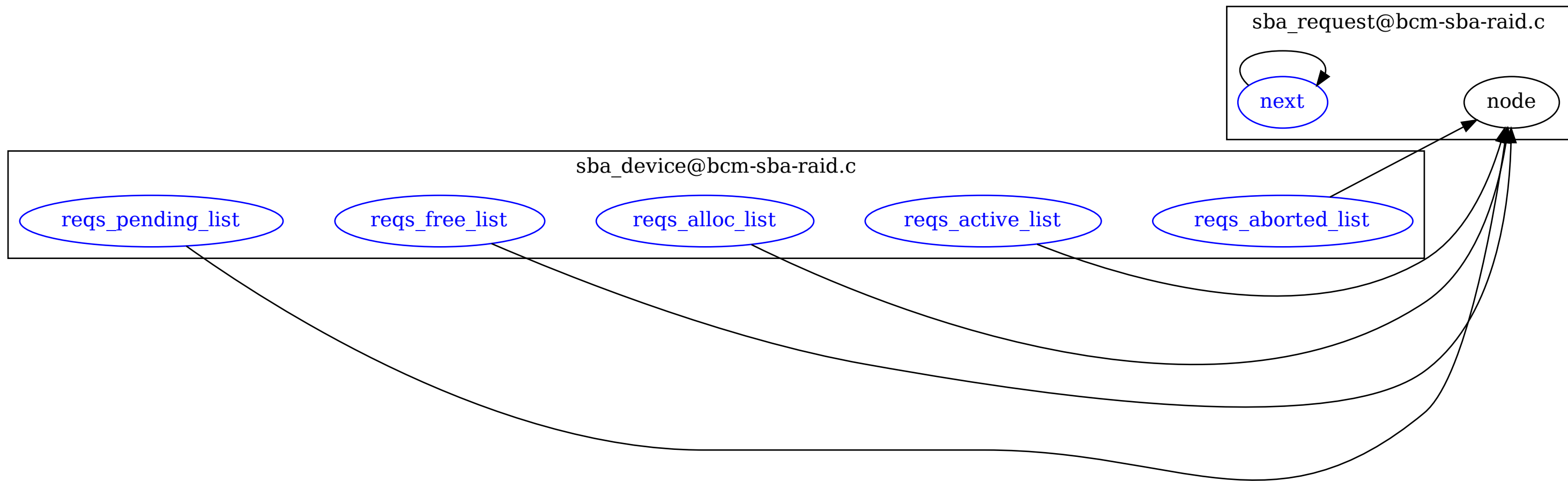
wmi_list

descriptor_priv@dell-wmi-descriptor.c

list

```
graph LR; wmi_list((wmi_list)) --> list((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@dell-wmi-descriptor.c' contains a blue oval labeled 'wmi_list'. On the right, a box labeled 'descriptor_priv@dell-wmi-descriptor.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'wmi_list' oval to the 'list' oval, indicating that 'wmi_list' points to 'list'.



ocfs2_filecheck

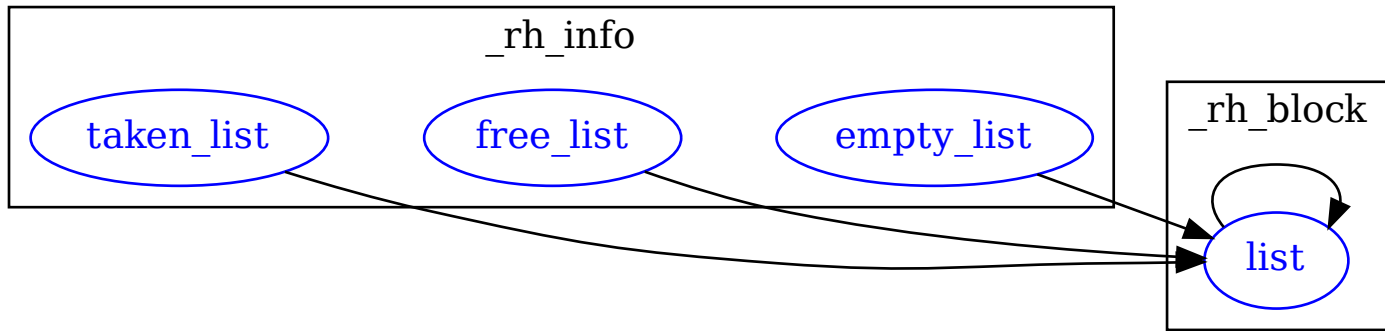
fc_head

ocfs2_filecheck_entry@filecheck.c

fe_list

```
graph LR; fc_head((fc_head)) --> fe_list((fe_list));
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'ocfs2_filecheck' contains an oval labeled 'fc_head'. On the right, a box labeled 'ocfs2_filecheck_entry@filecheck.c' contains an oval labeled 'fe_list'. A horizontal arrow points from the 'fc_head' oval to the 'fe_list' oval, indicating that 'fc_head' points to 'fe_list'.



var@mc-dev-allocator.c

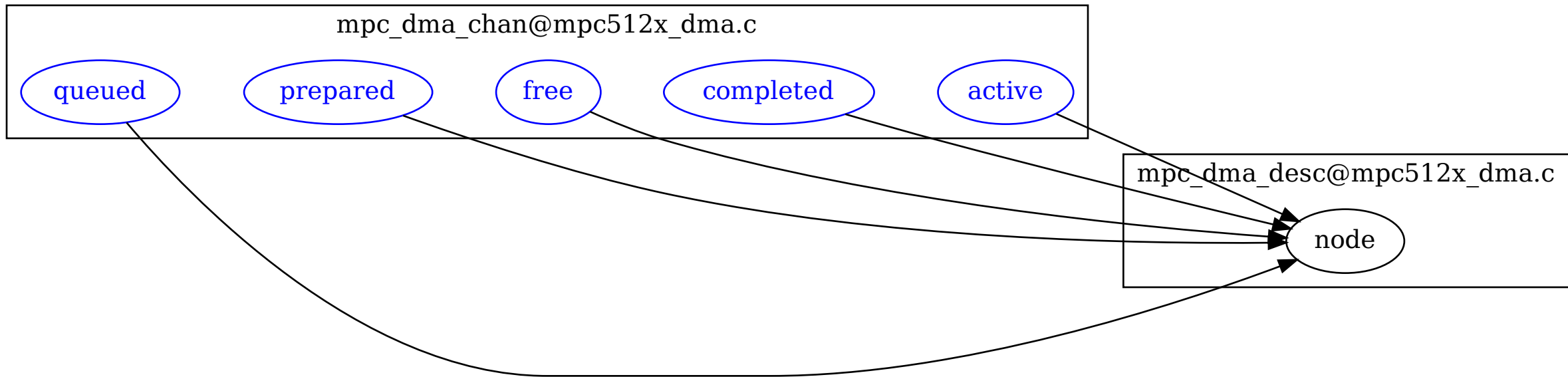
media_device_list

media_device_instance@mc-dev-allocator.c

list

```
graph LR; A["var@mc-dev-allocator.c  
media_device_list"] --> B["media_device_instance@mc-dev-allocator.c  
list"]
```

The diagram illustrates a pointer relationship between two variables. On the left, a box represents the source file `var@mc-dev-allocator.c`, containing the variable `media_device_list` which is highlighted by a blue oval. On the right, a box represents the source file `media_device_instance@mc-dev-allocator.c`, containing the variable `list` which is enclosed in an oval. A horizontal arrow points from the `media_device_list` oval to the `list` oval, indicating that the variable in the first file points to the variable in the second file.



zx_dma_dev@zx_dma.c

chan_pending

zx_dma_chan@zx_dma.c

node

```
graph LR; subgraph zx_dma_dev [zx_dma_dev@zx_dma.c]; cp([chan_pending]); end; subgraph zx_dma_chan [zx_dma_chan@zx_dma.c]; n([node]); end; cp --> n;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'zx_dma_dev@zx_dma.c' contains an oval labeled 'chan_pending'. On the right, a box labeled 'zx_dma_chan@zx_dma.c' contains an oval labeled 'node'. A horizontal arrow points from the 'chan_pending' oval to the 'node' oval, indicating that 'chan_pending' points to 'node'.

virtio_mmio_device@virtio_mmio.c

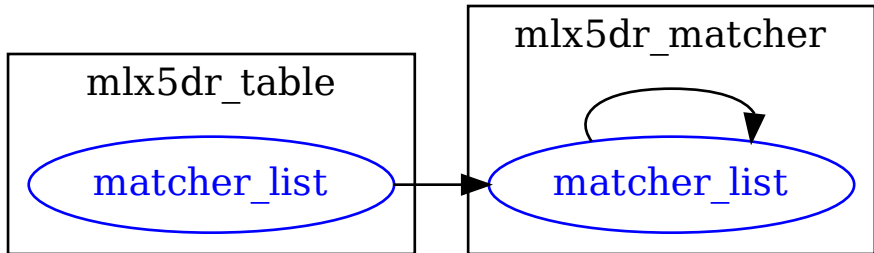
virtqueues

virtio_mmio_vq_info@virtio_mmio.c

node

```
graph LR; subgraph Box1 [virtio_mmio_device@virtio_mmio.c]; VQ(virtqueues); end; subgraph Box2 [virtio_mmio_vq_info@virtio_mmio.c]; N((node)); end; VQ --> N;
```

The diagram consists of two rectangular boxes. The left box is labeled 'virtio_mmio_device@virtio_mmio.c' and contains an oval labeled 'virtqueues' in blue text. The right box is labeled 'virtio_mmio_vq_info@virtio_mmio.c' and contains an oval labeled 'node' in black text. A horizontal arrow points from the right side of the 'virtqueues' oval to the left side of the 'node' oval.



var@trans_virtio.c

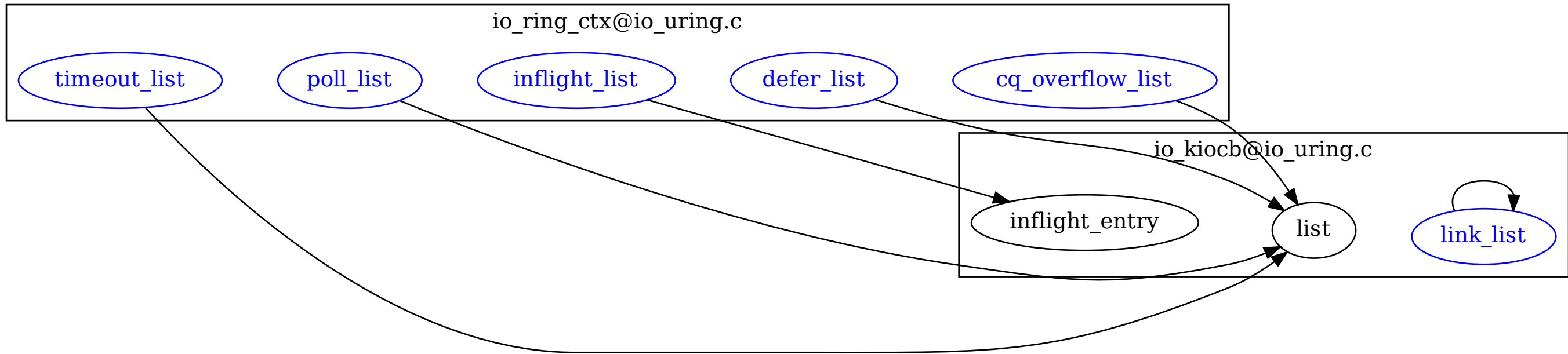
virtio_chan_list

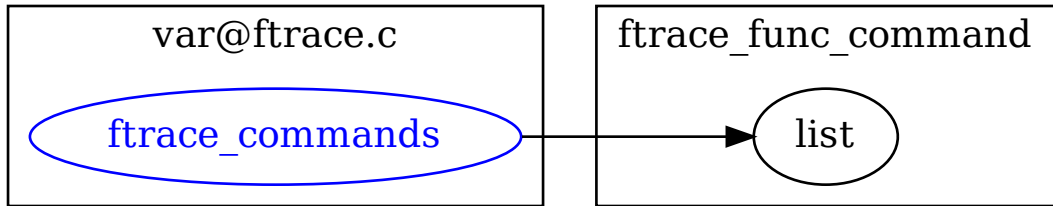
virtio_chan@trans_virtio.c

chan_list

```
graph LR; A[var@trans_virtio.c] --> B[virtio_chan@trans_virtio.c]; A --> C(chan_list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@trans_virtio.c' contains an oval labeled 'virtio_chan_list' with a blue border. On the right, a box labeled 'virtio_chan@trans_virtio.c' contains an oval labeled 'chan_list' with a black border. A horizontal arrow points from the right side of the 'virtio_chan_list' oval to the left side of the 'chan_list' oval, indicating that 'var@trans_virtio.c' points to 'chan_list'.





var@watchdog_core.c

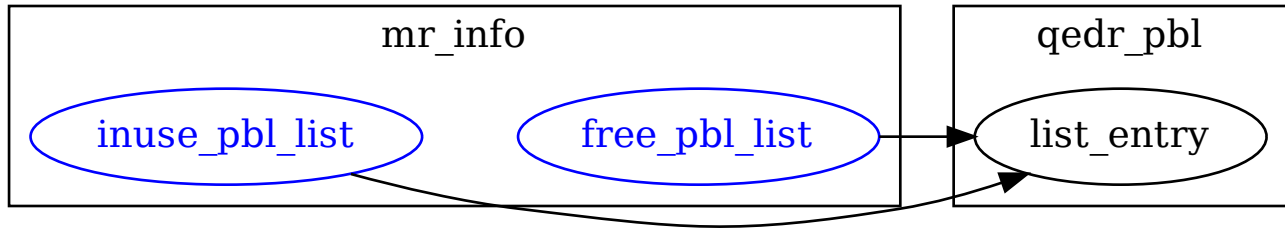
wtd_deferred_reg_list

watchdog_device

deferred

```
graph LR; A[wtd_deferred_reg_list] --> B[deferred];
```

The diagram illustrates a pointer relationship. On the left, a rectangular box labeled 'var@watchdog_core.c' contains an oval labeled 'wtd_deferred_reg_list' with a blue border. On the right, a rectangular box labeled 'watchdog_device' contains an oval labeled 'deferred' with a black border. A black arrow points from the right side of the 'wtd_deferred_reg_list' oval to the left side of the 'deferred' oval, indicating that 'wtd_deferred_reg_list' points to 'deferred'.



var@gameport.c

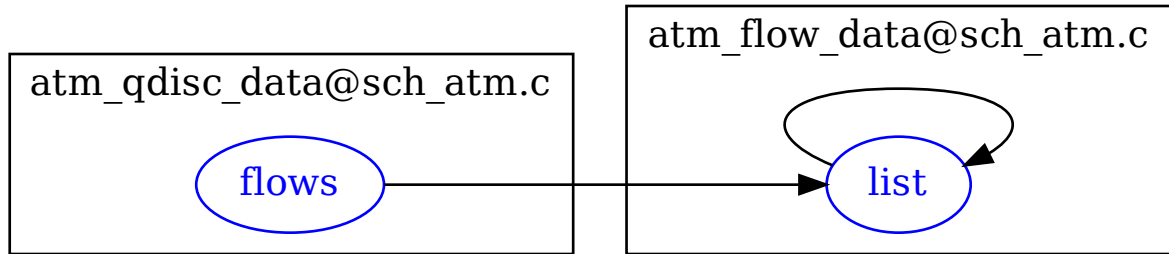
gameport_event_list

gameport_event@gameport.c

node

```
graph LR; A[gameport_event_list] --> B((node));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@gameport.c' contains an oval labeled 'gameport_event_list' with a blue border. On the right, a rectangular box labeled 'gameport_event@gameport.c' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'gameport_event_list' oval to the left side of the 'node' oval, indicating that 'gameport_event_list' points to 'node'.



ichxrom_window@ichxrom.c

maps

ichxrom_map_info@ichxrom.c

list

```
graph LR; maps([maps]) --> list([list]);
```

The diagram consists of two rectangular boxes. The left box contains the text 'ichxrom_window@ichxrom.c' at the top and an oval labeled 'maps' in blue text below it. The right box contains the text 'ichxrom_map_info@ichxrom.c' at the top and an oval labeled 'list' in black text below it. A horizontal arrow points from the right side of the 'maps' oval to the left side of the 'list' oval.

kbl_codec_private@kbl_da7219_max98357a.c

hdmi_pcm_list

kbl_hdmi_pcm@kbl_da7219_max98357a.c

head

```
graph LR; A[kbl_codec_private@kbl_da7219_max98357a.c] --> B(kbl_hdmi_pcm@kbl_da7219_max98357a.c); B --> C(head);
```

The diagram illustrates a pointer relationship between two kernel structures. On the left, a box represents the `kbl_codec_private@kbl_da7219_max98357a.c` structure, which contains a blue oval labeled `hdmi_pcm_list`. On the right, a box represents the `kbl_hdmi_pcm@kbl_da7219_max98357a.c` structure, which contains a black oval labeled `head`. A horizontal arrow points from the `hdmi_pcm_list` oval to the `head` oval, indicating that the `hdmi_pcm_list` pointer in the first structure points to the `head` pointer in the second structure.

atmel_sha_drv@atmel-sha.c

dev_list

atmel_sha_dev@atmel-sha.c

list

```
graph LR; subgraph Box1 [atmel_sha_drv@atmel-sha.c]; dev_list; end; subgraph Box2 [atmel_sha_dev@atmel-sha.c]; list; end; dev_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'atmel_sha_drv@atmel-sha.c' contains an oval labeled 'dev_list' with a blue border. On the right, a rectangular box labeled 'atmel_sha_dev@atmel-sha.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'dev_list' oval to the left side of the 'list' oval, indicating that 'dev_list' points to 'list'.

var@hid-cougar.c

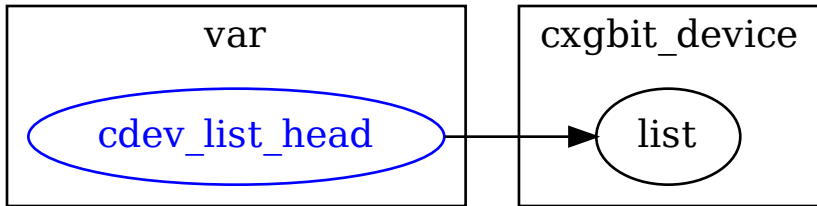
cougar_udev_list

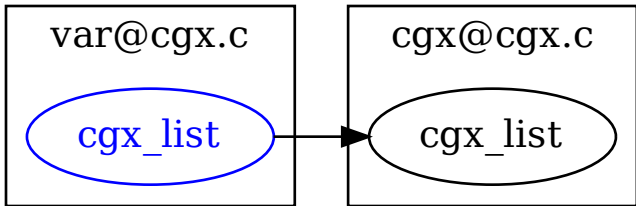
cougar_shared@hid-cougar.c

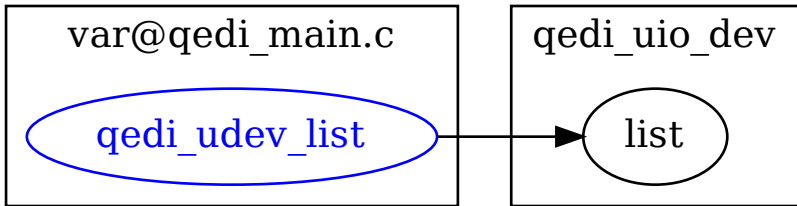
list

```
graph LR; A[cougar_udev_list] --> B((list));
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@hid-cougar.c' and contains an oval labeled 'cougar_udev_list' with a blue border. The right box is labeled 'cougar_shared@hid-cougar.c' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'cougar_udev_list' oval to the left side of the 'list' oval.







acpi_memory_device@acpi_memhotplug.c

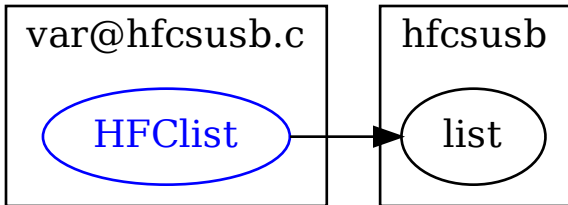
res_list

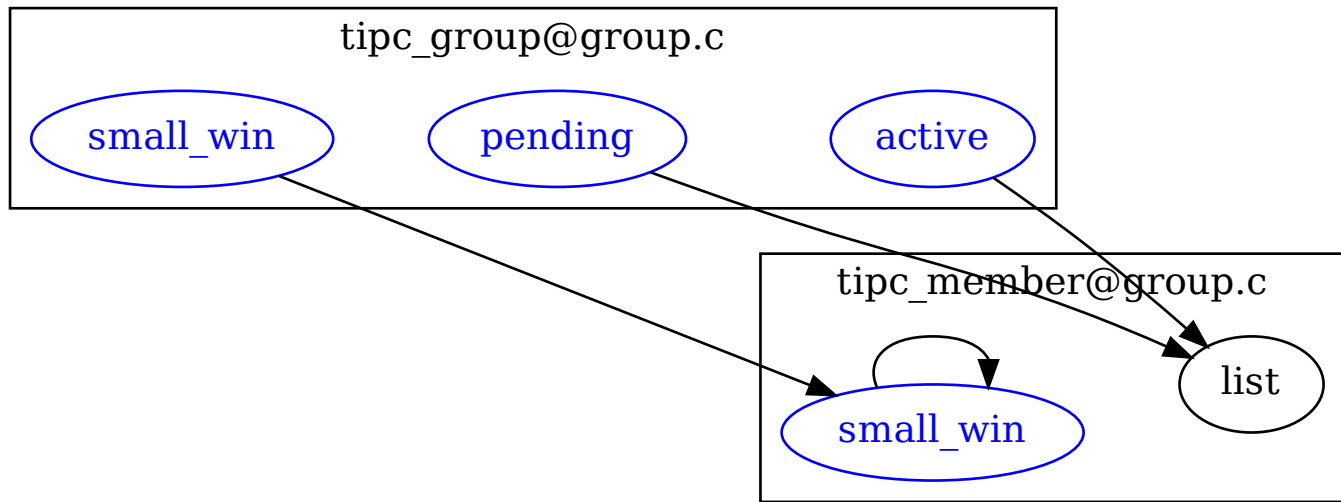
acpi_memory_info@acpi_memhotplug.c

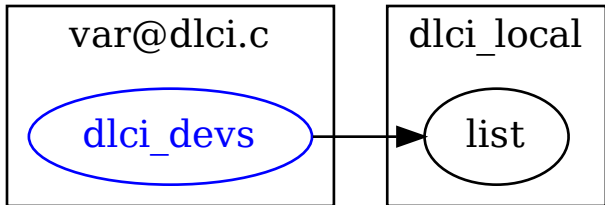
list

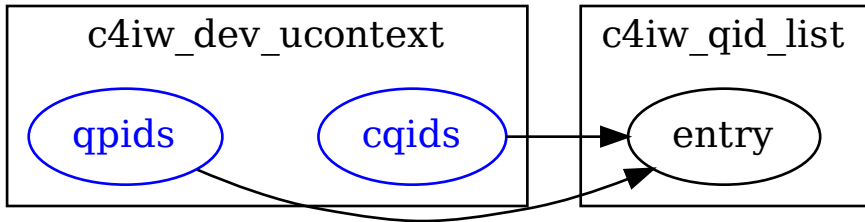
```
graph LR; A[acpi_memory_device@acpi_memhotplug.c] --> B[acpi_memory_info@acpi_memhotplug.c]; subgraph A; R((res_list)); end; subgraph B; L((list)); end; R --> L;
```

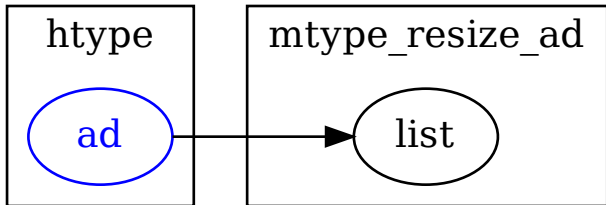
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'acpi_memory_device@acpi_memhotplug.c' contains a blue oval labeled 'res_list'. On the right, a box labeled 'acpi_memory_info@acpi_memhotplug.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'res_list' oval to the 'list' oval, indicating that 'res_list' points to 'list'.











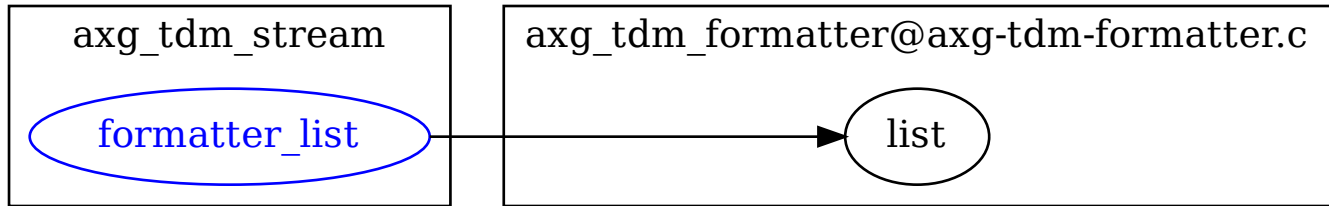
fc_exch_pool@fc_exch.c

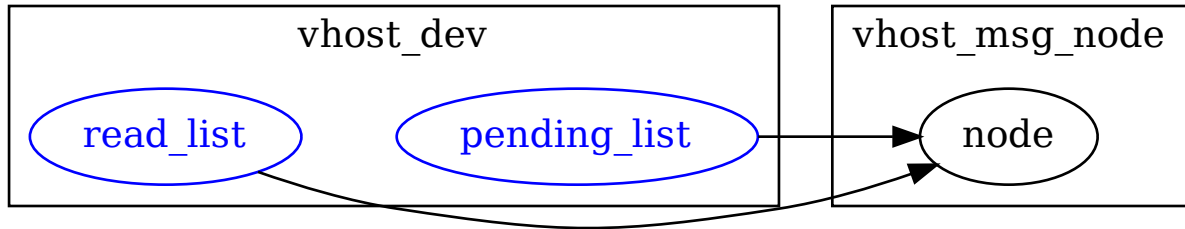
ex_list

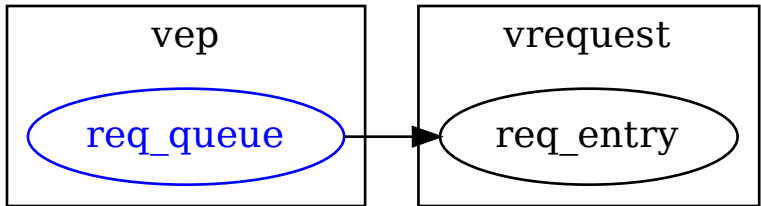
fc_exch

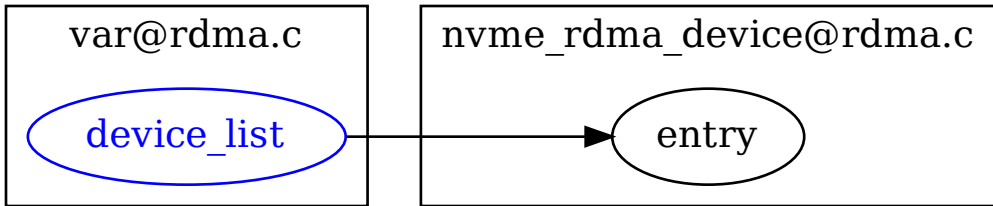
ex_list

The diagram consists of two rectangular boxes. The left box is labeled 'fc_exch_pool@fc_exch.c' and contains a blue oval labeled 'ex_list'. The right box is labeled 'fc_exch' and contains a black oval labeled 'ex_list'. A horizontal arrow points from the right side of the 'ex_list' oval in the left box to the left side of the 'ex_list' oval in the right box, indicating a pointer or reference relationship.









ines_clock@ptp_ines.c

list

var@ptp_ines.c

ines_clocks

```
graph LR; subgraph "ines_clock@ptp_ines.c"; list((list)); end; subgraph "var@ptp_ines.c"; ines_clocks((ines_clocks)); end; list --> ines_clocks; ines_clocks --> list;
```

The diagram illustrates a bidirectional relationship between two variables. On the left, a box labeled 'ines_clock@ptp_ines.c' contains a blue oval labeled 'list'. On the right, a box labeled 'var@ptp_ines.c' contains a blue oval labeled 'ines_clocks'. Two black arrows connect the ovals: one pointing from 'list' to 'ines_clocks' and another pointing from 'ines_clocks' back to 'list'.

mlxsw_sp_mr_tcam_erif_list@spectrum_mr_tcam.c

erif_sublists

mlxsw_sp_mr_erif_sublist@spectrum_mr_tcam.c

list

```
graph LR; A(erif_sublists) --> B(list)
```

var@autoidle.c

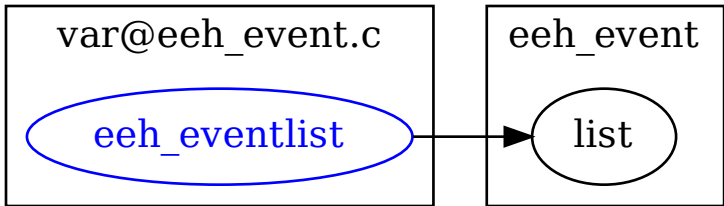
autoidle_clks

clk_ti_autoidle@autoidle.c

node

```
graph LR; subgraph Box1 [var@autoidle.c]; A([autoidle_clks]); end; subgraph Box2 [clk_ti_autoidle@autoidle.c]; B([node]); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@autoidle.c' and contains an oval labeled 'autoidle_clks' in blue text. The right box is labeled 'clk_ti_autoidle@autoidle.c' and contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'autoidle_clks' oval to the left side of the 'node' oval.



var@adummy.c

adummy_devs

adummy_dev@adummy.c

entry

```
graph LR; A[adummy_devs] --> B[entry];
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@adummy.c' at the top and 'adummy_devs' at the bottom. The 'adummy_devs' text is enclosed in a blue oval. The right box contains the text 'adummy_dev@adummy.c' at the top and 'entry' at the bottom. The 'entry' text is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.

var@industrialio-sw-trigger.c

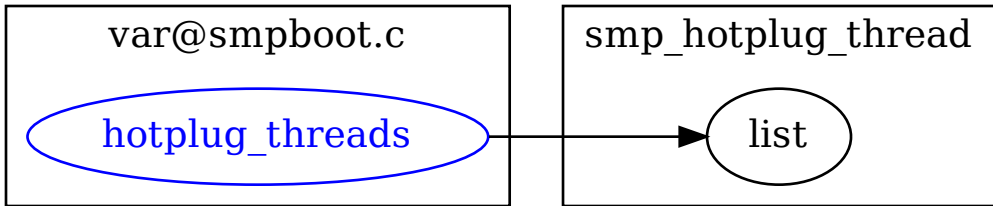
iio_trigger_types_list

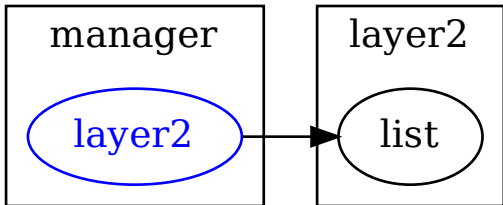
iio_sw_trigger_type

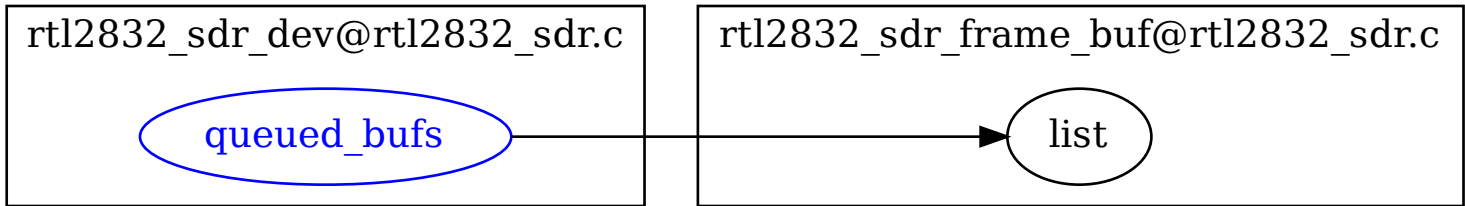
list

```
graph LR; A[iio_trigger_types_list] --> B((list));
```

The diagram illustrates a pointer relationship. On the left, a rectangular box contains the text 'var@industrialio-sw-trigger.c' and 'iio_trigger_types_list'. The 'iio_trigger_types_list' is enclosed in a blue oval. A horizontal arrow points from this oval to a black oval containing the text 'list' on the right. The right side is also enclosed in a rectangular box, which also contains the text 'iio_sw_trigger_type'.







var@yama_lsm.c

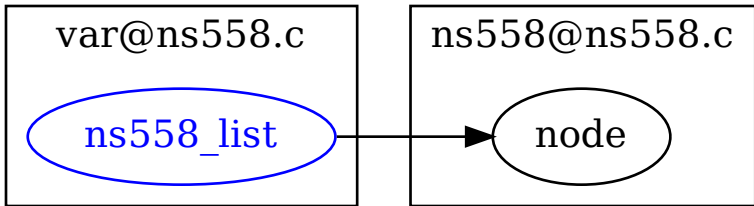
ptracer_relations

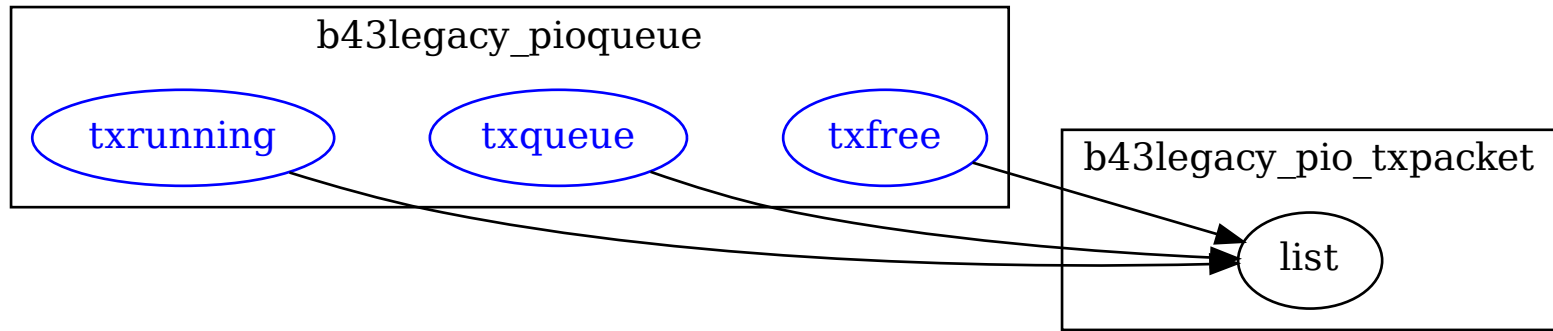
ptrace_relation@yama_lsm.c

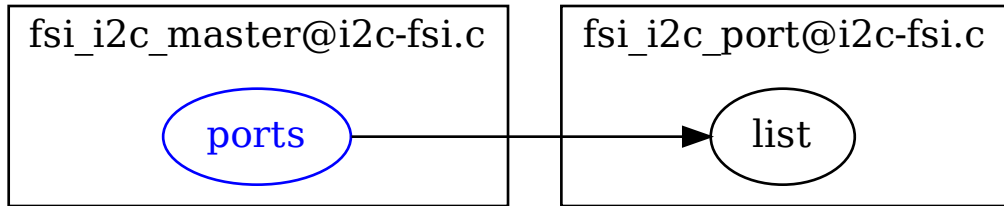
node

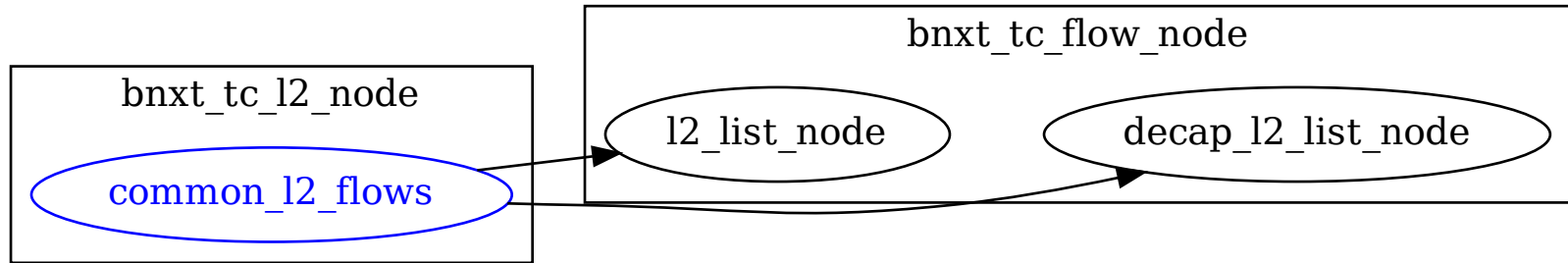
```
graph LR; subgraph var_yama_lsm_c [var@yama_lsm.c]; ptracer_relations; end; subgraph ptrace_relation_yama_lsm_c [ptrace_relation@yama_lsm.c]; node; end; ptracer_relations --> node;
```

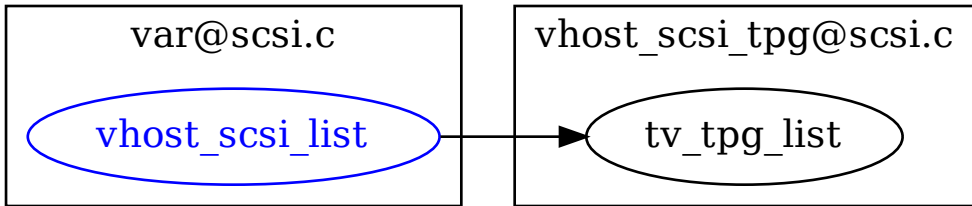
The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@yama_lsm.c' contains an oval labeled 'ptracer_relations'. On the right, a box labeled 'ptrace_relation@yama_lsm.c' contains an oval labeled 'node'. A horizontal arrow points from the 'ptracer_relations' oval to the 'node' oval, indicating that 'ptracer_relations' points to 'node'.











var@parport_pc.c

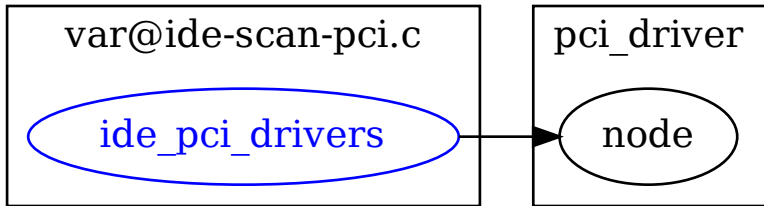
ports_list

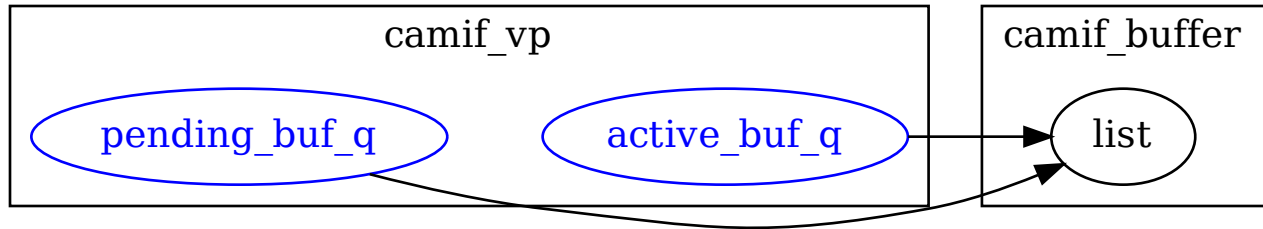
parport_pc_private

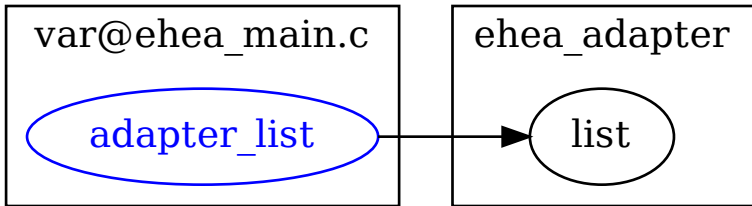
list

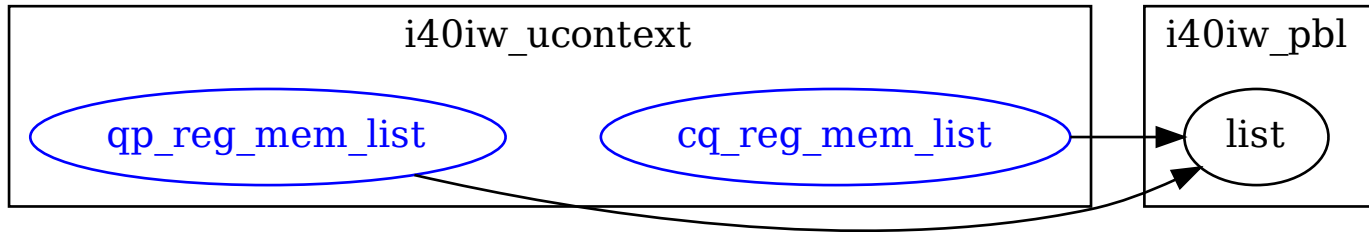
```
graph LR; subgraph Box1 [var@parport_pc.c]; ports_list([ports_list]); end; subgraph Box2 [parport_pc_private]; list([list]); end; ports_list --> list;
```

The diagram illustrates a pointer relationship between two structures. On the left, a box labeled 'var@parport_pc.c' contains an oval labeled 'ports_list' in blue text. On the right, a box labeled 'parport_pc_private' contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'ports_list' oval to the left side of the 'list' oval, indicating that 'ports_list' points to 'list'.

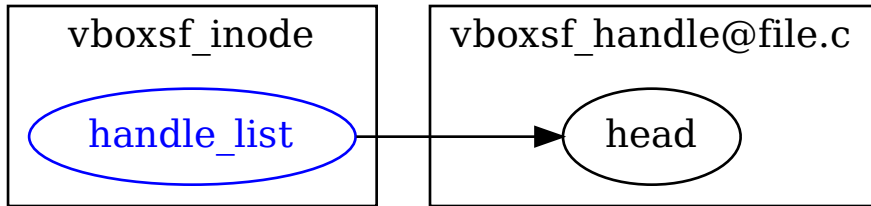


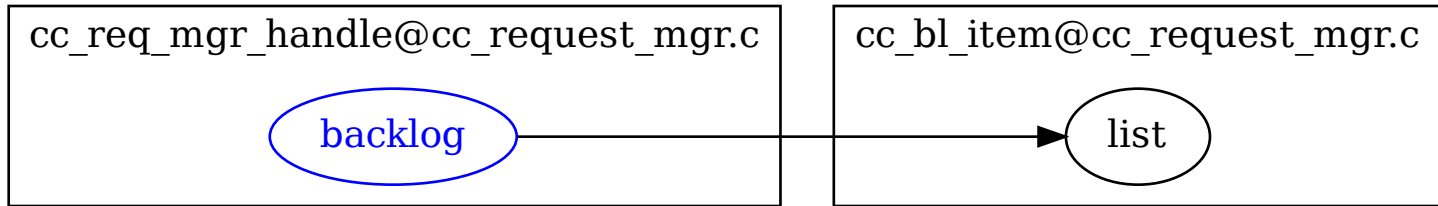


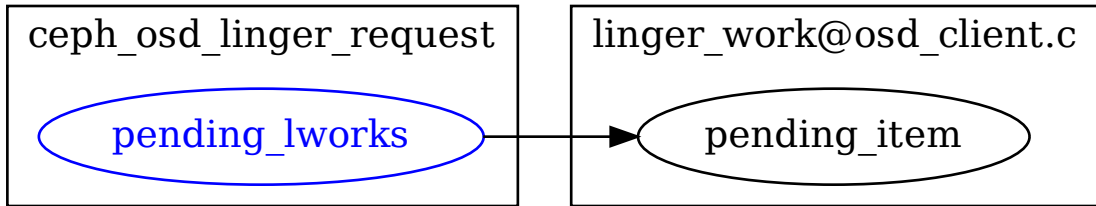


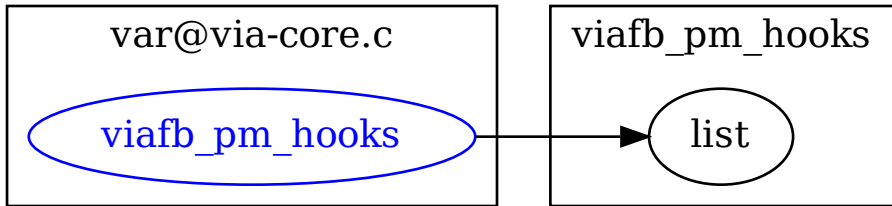


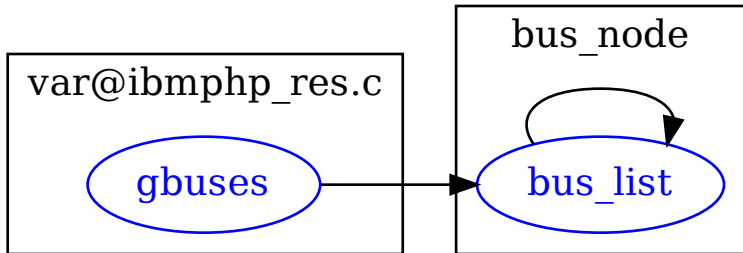


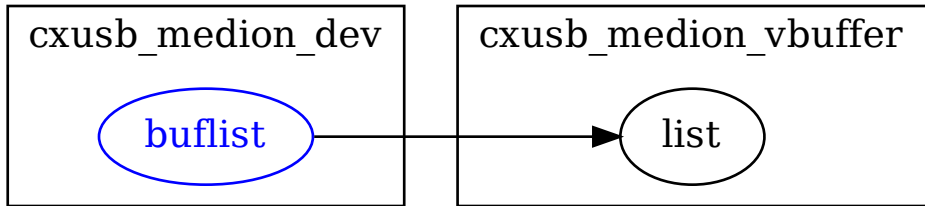












var@sta2x11-mfd.c

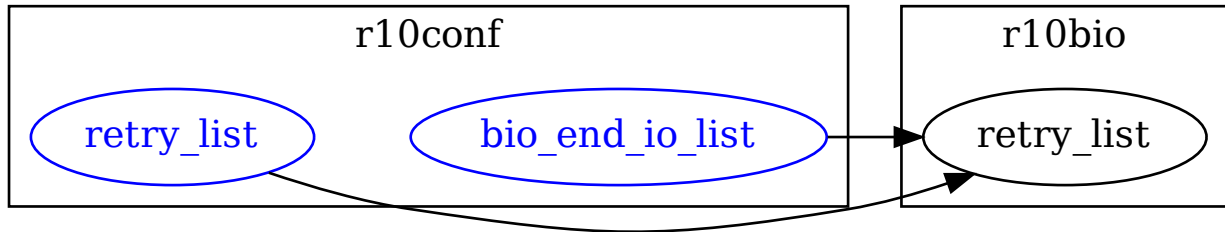
sta2x11_mfd_list

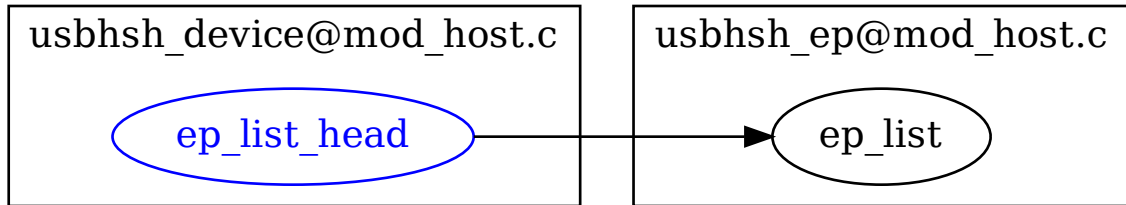
sta2x11_mfd@sta2x11-mfd.c

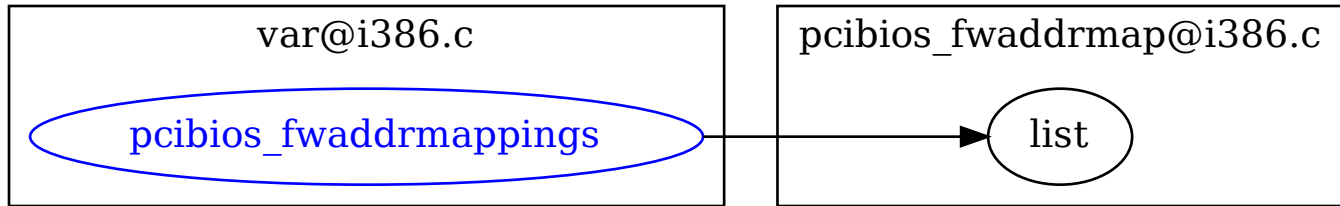
list

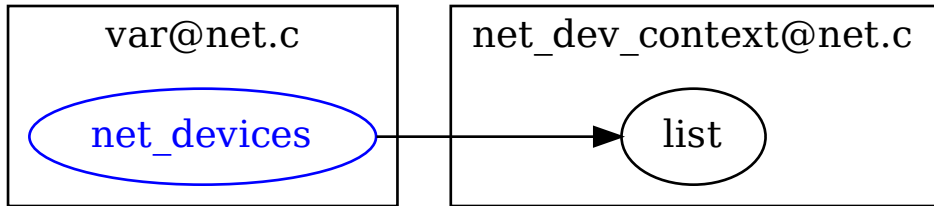
```
graph LR; A[sta2x11_mfd_list] --> B((list));
```

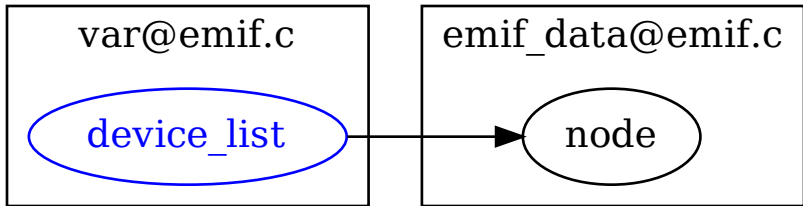
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@sta2x11-mfd.c' at the top and 'sta2x11_mfd_list' below it. The 'sta2x11_mfd_list' text is enclosed in a blue oval. On the right, another rectangular box contains the text 'sta2x11_mfd@sta2x11-mfd.c' at the top and 'list' below it. The 'list' text is enclosed in a black oval. A horizontal arrow points from the blue oval on the left to the black oval on the right, indicating that 'sta2x11_mfd_list' points to 'list'.

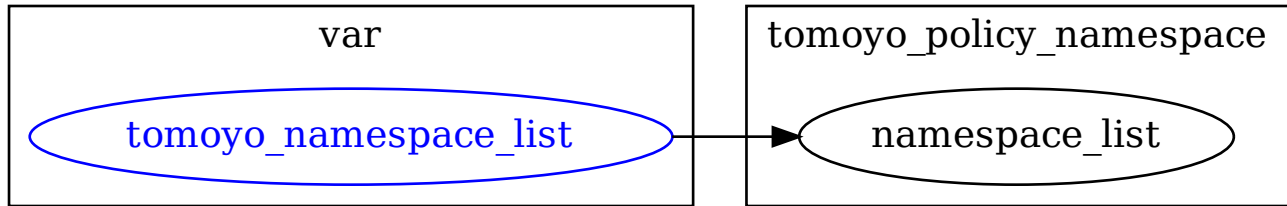












var@sram_reserve_regions

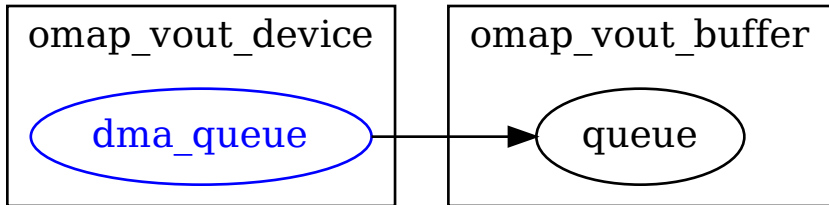
reserve_list

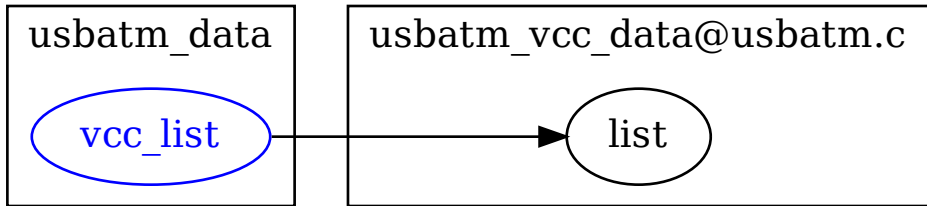
sram_reserve

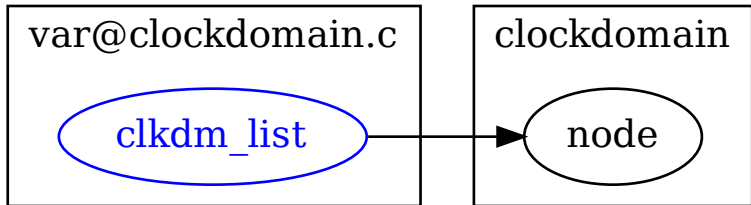
list

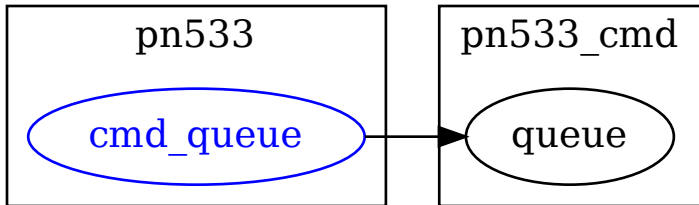
```
graph LR; A[reserve_list] --> B(list);
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@sram_reserve_regions' at the top and an oval containing 'reserve_list' in blue text below it. The right box contains the text 'sram_reserve' at the top and an oval containing 'list' below it. A horizontal arrow points from the right side of the 'reserve_list' oval to the left side of the 'list' oval.









fw_download@fw-download.c

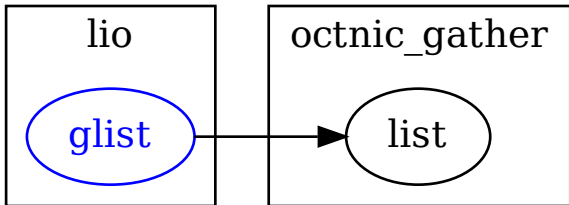
fw_requests

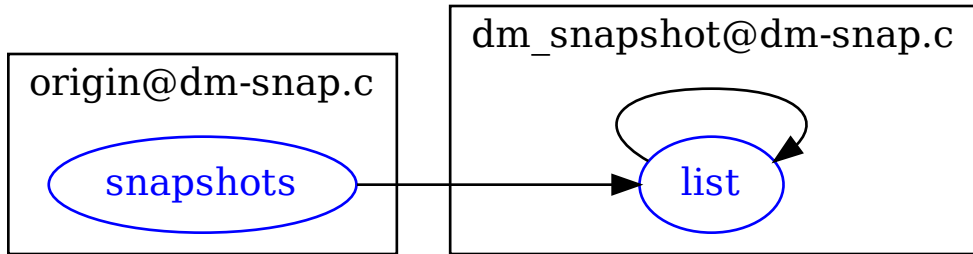
fw_request@fw-download.c

node

```
graph LR; subgraph Box1 [fw_download@fw-download.c]; fw_requests; end; subgraph Box2 [fw_request@fw-download.c]; node; end; fw_requests --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'fw_download@fw-download.c' and contains an oval labeled 'fw_requests' in blue text. The right box is labeled 'fw_request@fw-download.c' and contains an oval labeled 'node'. A horizontal arrow points from the 'fw_requests' oval to the 'node' oval, crossing the boundary between the two boxes.





atmel_ebi@atmel-ebi.c

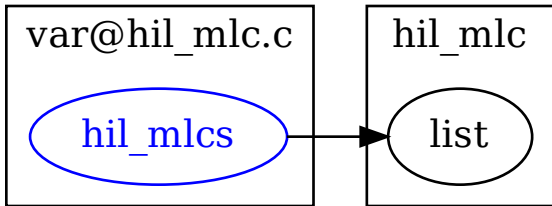
devs

atmel_ebi_dev@atmel-ebi.c

node

```
graph LR; A[atmel_ebi@atmel-ebi.c] --> B[atmel_ebi_dev@atmel-ebi.c]; subgraph A; A1([devs]); end; subgraph B; B1([node]); end; A1 --> B1;
```

The diagram consists of two rectangular boxes. The left box contains the text 'atmel_ebi@atmel-ebi.c' and a blue oval labeled 'devs'. The right box contains the text 'atmel_ebi_dev@atmel-ebi.c' and a black oval labeled 'node'. A horizontal arrow points from the 'devs' oval to the 'node' oval.



vmpressure

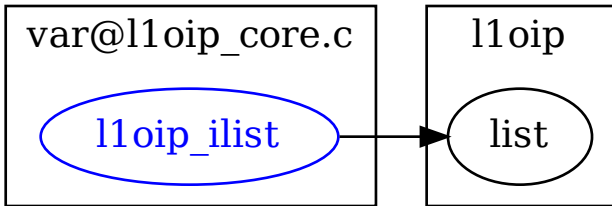
events

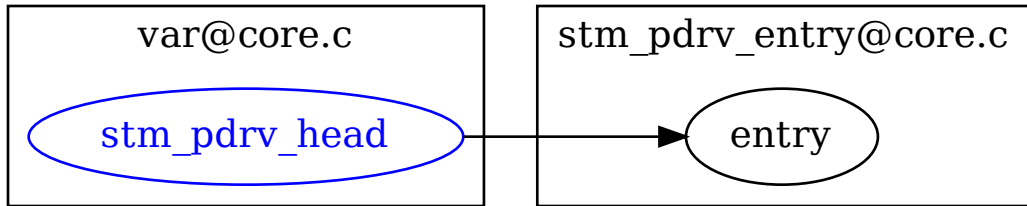
vmpressure_event@vmpressure.c

node

```
graph LR; subgraph vmpressure; events; end; subgraph vmpressure_event_at_vmpressure_c [vmpressure_event@vmpressure.c]; node; end; events --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'vmpressure' and contains an oval labeled 'events' with a blue border. The right box is labeled 'vmpressure_event@vmpressure.c' and contains an oval labeled 'node' with a black border. A horizontal arrow points from the right side of the 'events' oval to the left side of the 'node' oval.





var@iort.c

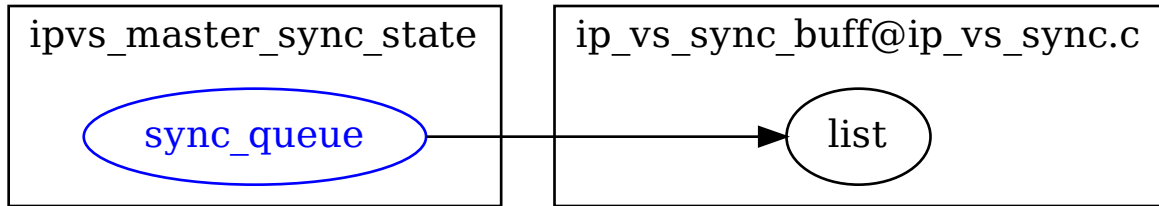
iort_msi_chip_list

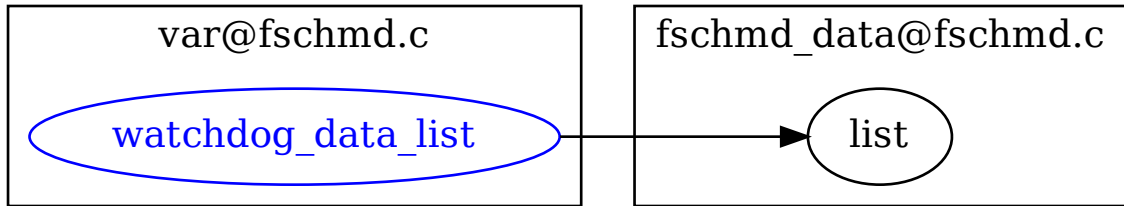
iort_its_msi_chip@iort.c

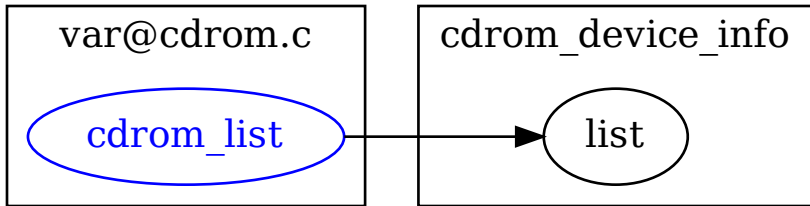
list

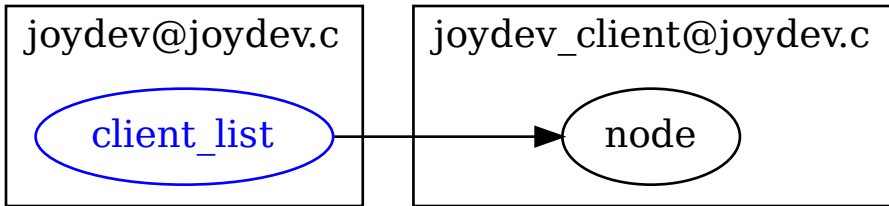
```
graph LR; subgraph var_at_iort_c [var@iort.c]; iort_msi_chip_list; end; subgraph iort_its_msi_chip_at_iort_c [iort_its_msi_chip@iort.c]; list; end; iort_msi_chip_list --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'var@iort.c' contains an oval labeled 'iort_msi_chip_list' with a blue border. On the right, a box labeled 'iort_its_msi_chip@iort.c' contains an oval labeled 'list'. A horizontal arrow points from the 'iort_msi_chip_list' oval to the 'list' oval, indicating that the variable 'iort_msi_chip_list' points to the 'list' structure.









page_address_slot@highmem.c

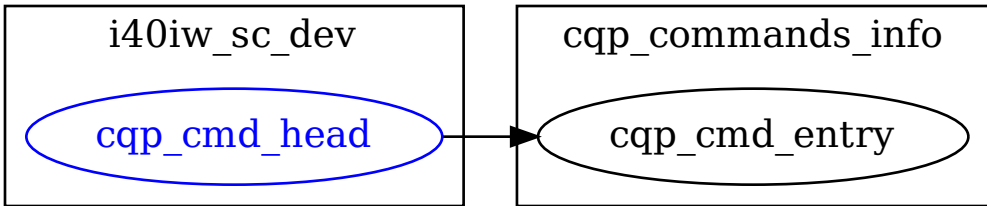
lh

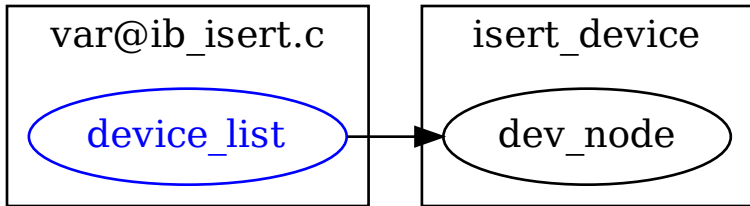
page_address_map@highmem.c

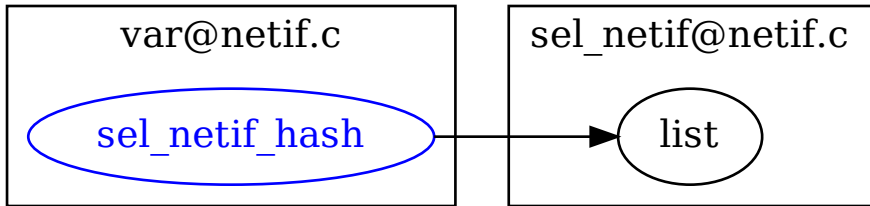
list

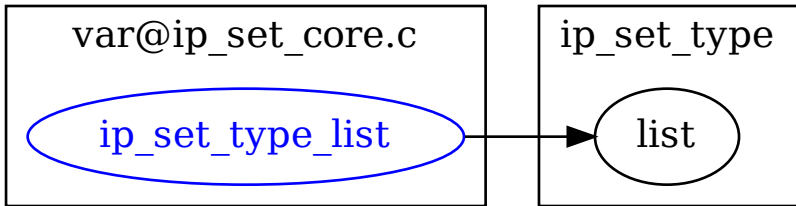


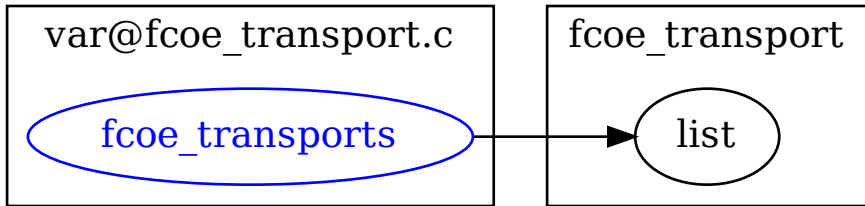
```
graph LR; subgraph Box1 [page_address_slot@highmem.c]; lh((lh)); end; subgraph Box2 [page_address_map@highmem.c]; list((list)); end; lh --> list;
```









var@mmc_test.c

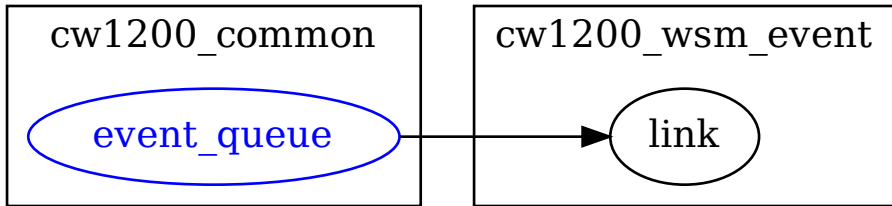
mmc_test_file_test

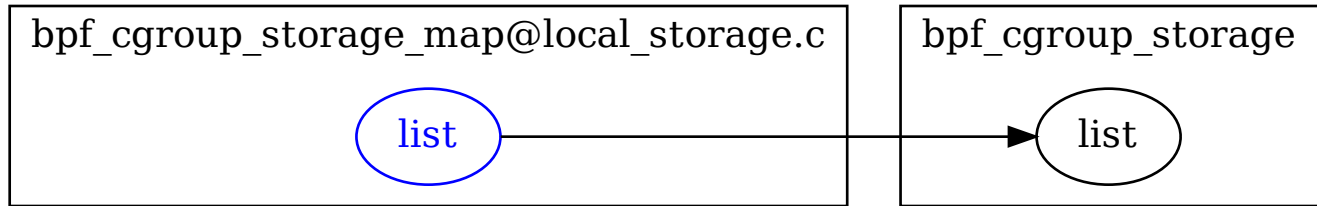
mmc_test_dbgfs_file@mmc_test.c

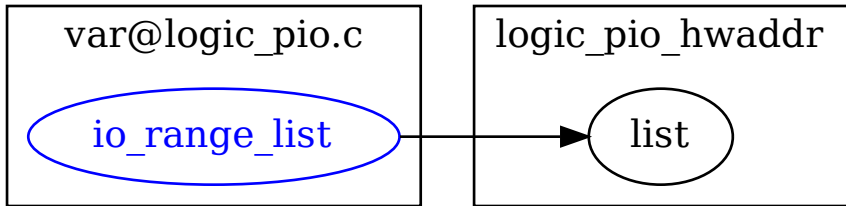
link

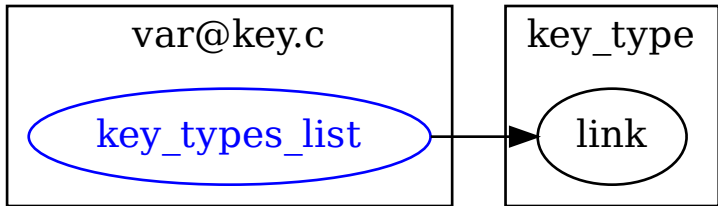
```
graph LR; subgraph Module1 [var@mmc_test.c]; direction TB; A(mmc_test_file_test); end; subgraph Module2 [mmc_test_dbgfs_file@mmc_test.c]; direction TB; B((link)); end; A --> B;
```

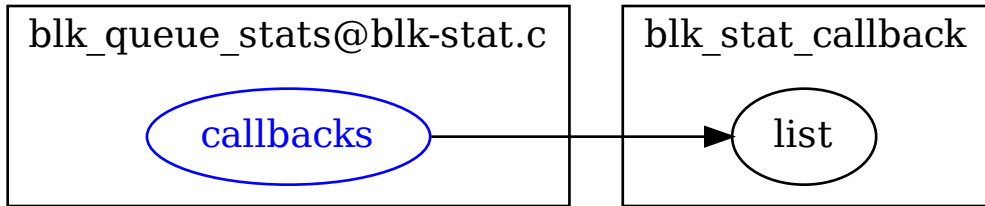
The diagram illustrates a link between two modules. On the left, a rectangular box represents the module 'var@mmc_test.c'. Inside this box, the text 'mmc_test_file_test' is enclosed in a blue oval. On the right, another rectangular box represents the module 'mmc_test_dbgfs_file@mmc_test.c'. Inside this box, the text 'link' is enclosed in a black oval. A horizontal arrow points from the blue oval in the first box to the black oval in the second box, indicating a dependency or link between the two components.

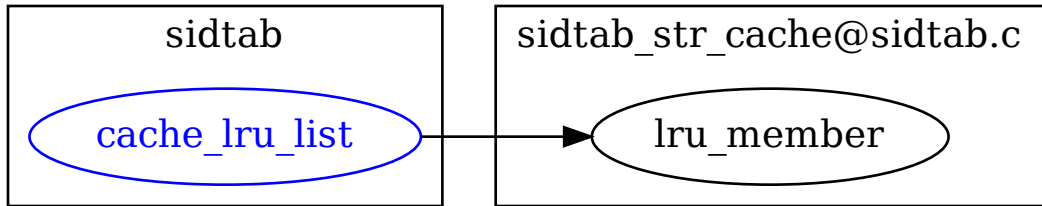


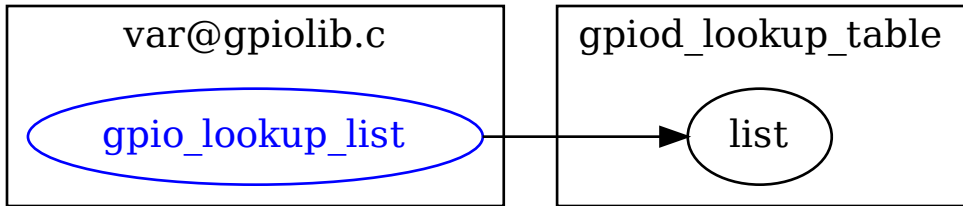


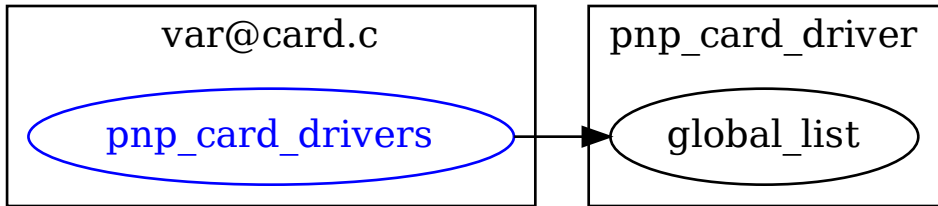


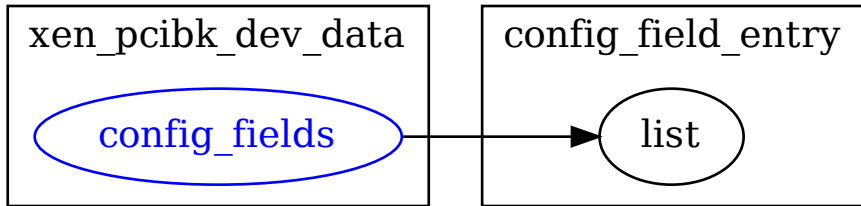


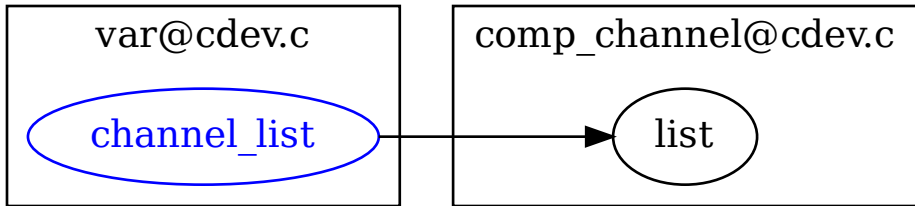


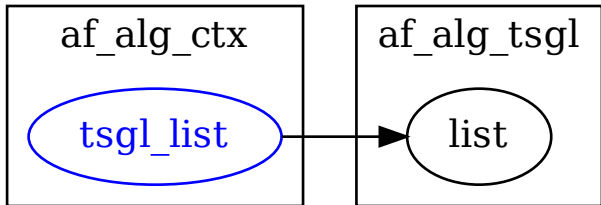












var@visorchipset.c

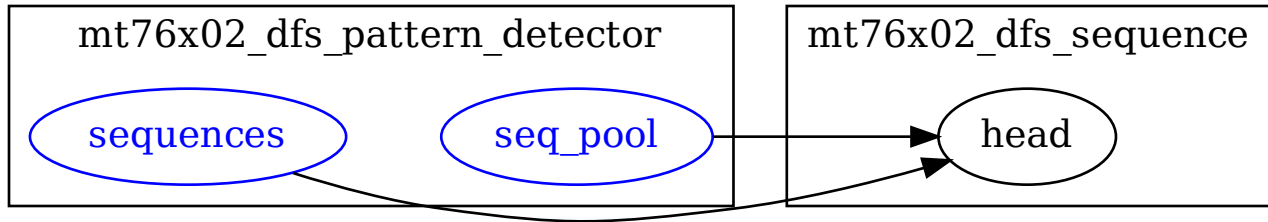
parahotplug_request_list

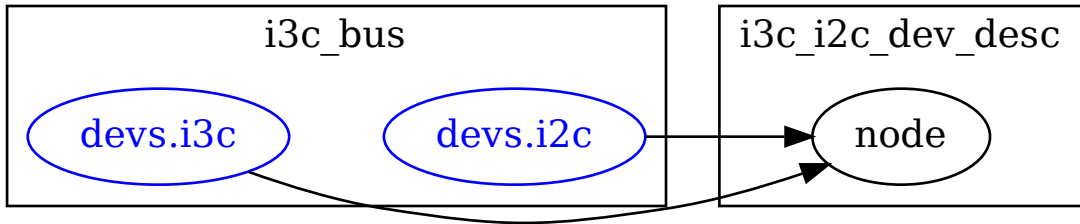
parahotplug_request@visorchipset.c

list

```
graph LR; A[parahotplug_request_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@visorchipset.c`. Inside this box, the variable `parahotplug_request_list` is highlighted with a blue oval. On the right, another rectangular box represents the source file `parahotplug_request@visorchipset.c`. Inside this box, there is an oval labeled `list`. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the `list` oval in the second box, indicating that `parahotplug_request_list` points to the `list` variable.





allegro_dev@allegro-core.c

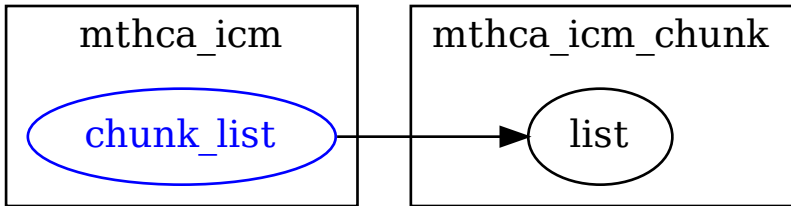
channels

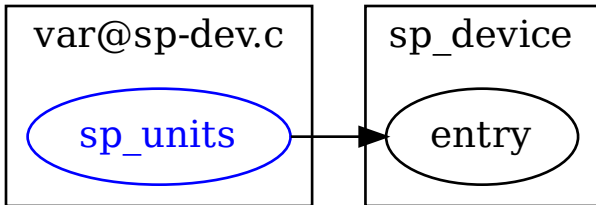
allegro_channel@allegro-core.c

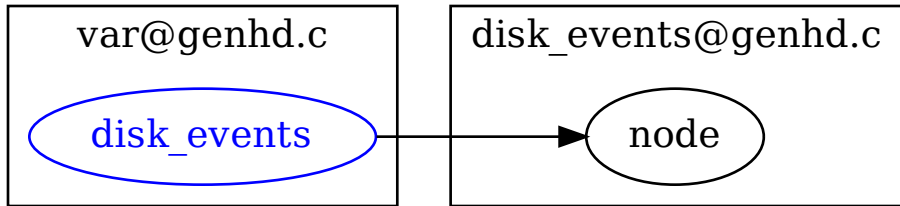
list

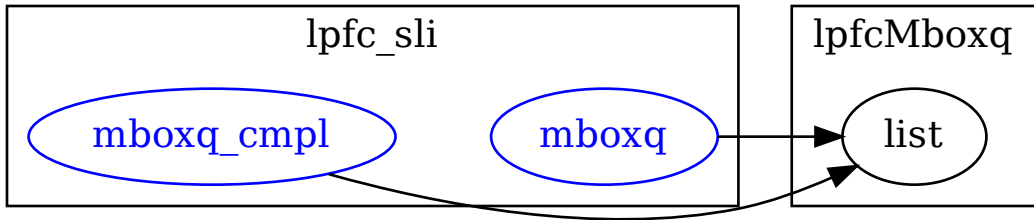
```
graph LR; A[allegro_dev@allegro-core.c] --> B[allegro_channel@allegro-core.c]; subgraph A [allegro_dev@allegro-core.c]; C((channels)); end; subgraph B [allegro_channel@allegro-core.c]; D((list)); end; C --> D;
```

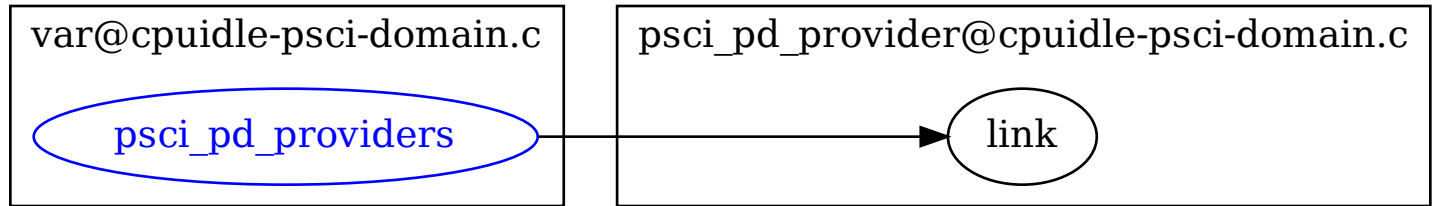
The diagram consists of two rectangular boxes. The left box contains the email address 'allegro_dev@allegro-core.c' and a blue oval labeled 'channels'. The right box contains the email address 'allegro_channel@allegro-core.c' and a black oval labeled 'list'. A horizontal arrow points from the 'channels' oval in the left box to the 'list' oval in the right box.

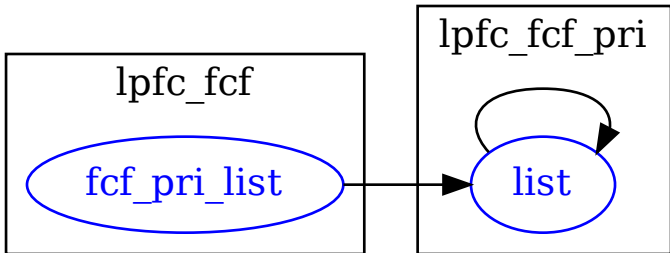


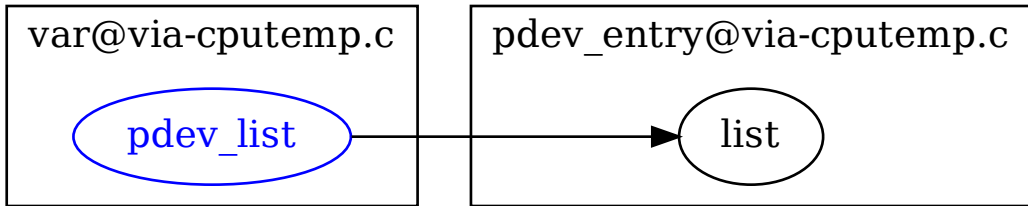












var@smscoreapi.c

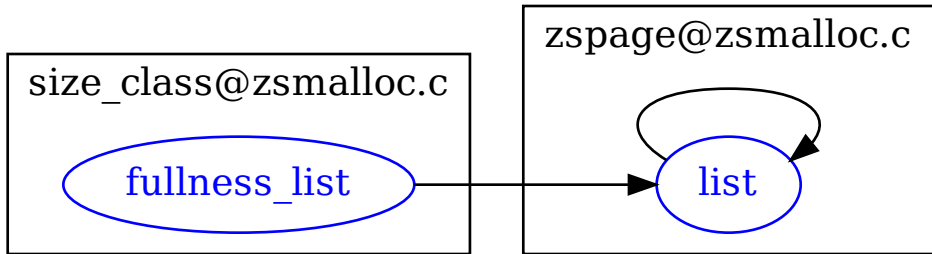
g_smscore_notifyees

smscore_device_notifyee_t@smscoreapi.c

entry

```
graph LR; subgraph var ["var@smscoreapi.c"]; g_smscore_notifyees; end; subgraph smscore_device_notifyee_t ["smscore_device_notifyee_t@smscoreapi.c"]; entry; end; g_smscore_notifyees --> entry;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'var@smscoreapi.c' contains an oval labeled 'g_smscore_notifyees'. On the right, a box labeled 'smscore_device_notifyee_t@smscoreapi.c' contains an oval labeled 'entry'. A horizontal arrow points from the 'g_smscore_notifyees' oval to the 'entry' oval, indicating that 'g_smscore_notifyees' points to the 'entry' within the 'smscore_device_notifyee_t' structure.



var@cx231xx-core.c

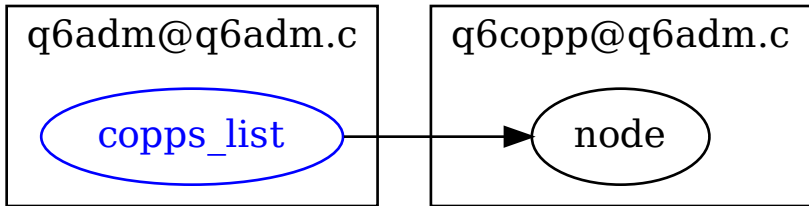
cx231xx_extension_devlist

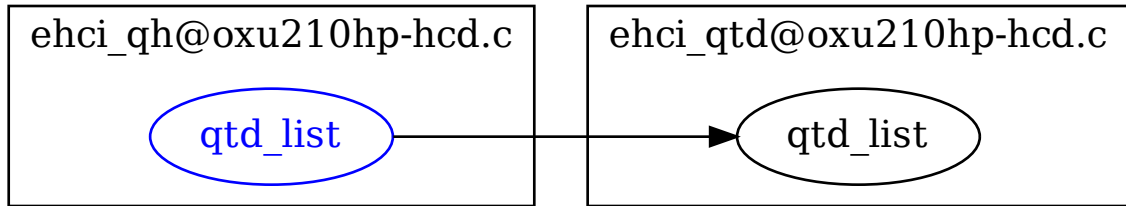
cx231xx_ops

next

```
graph LR; A["var@cx231xx-core.c  
cx231xx_extension_devlist"] --> B["cx231xx_ops  
next"]
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box contains the text 'var@cx231xx-core.c' and a blue oval containing 'cx231xx_extension_devlist'. On the right, another rectangular box contains the text 'cx231xx_ops' and an oval containing 'next'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the 'next' oval in the second box, indicating that 'cx231xx_extension_devlist' points to 'next'.





cld_net@nfs4recover.c

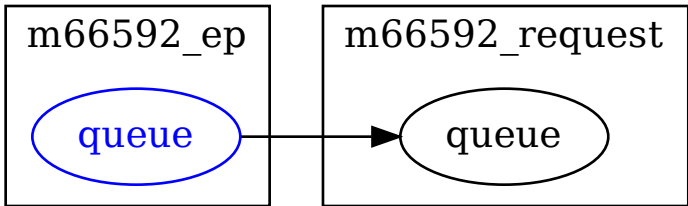
cn_list

cld_upcall@nfs4recover.c

cu_list

```
graph LR; subgraph Box1 [cld_net@nfs4recover.c]; cn_list((cn_list)); end; subgraph Box2 [cld_upcall@nfs4recover.c]; cu_list((cu_list)); end; cn_list --> cu_list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'cld_net@nfs4recover.c' contains an oval labeled 'cn_list' in blue text. On the right, a rectangular box labeled 'cld_upcall@nfs4recover.c' contains an oval labeled 'cu_list' in black text. A horizontal arrow points from the right side of the 'cn_list' oval to the left side of the 'cu_list' oval, indicating that 'cn_list' points to 'cu_list'.



var@trace_events_trigger.c

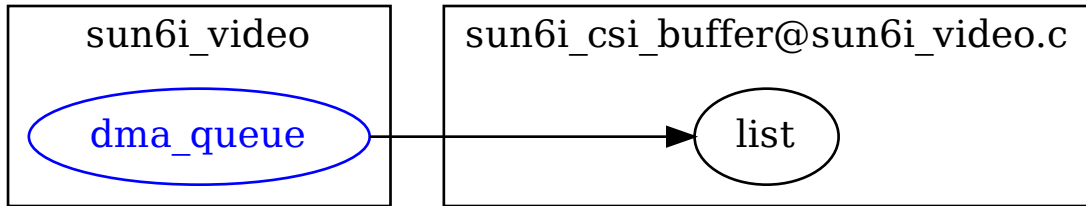
trigger_commands

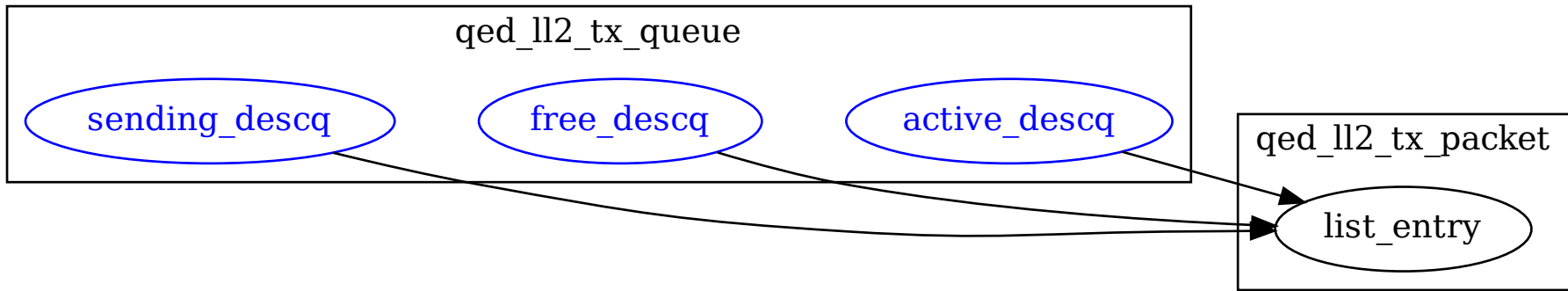
event_command

list

```
graph LR; subgraph Box1 [var@trace_events_trigger.c]; TC(trigger_commands); end; subgraph Box2 [event_command]; L((list)); end; TC --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@trace_events_trigger.c' and contains a blue oval labeled 'trigger_commands'. The right box is labeled 'event_command' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'trigger_commands' oval to the left side of the 'list' oval.





var@arm_sdei.c

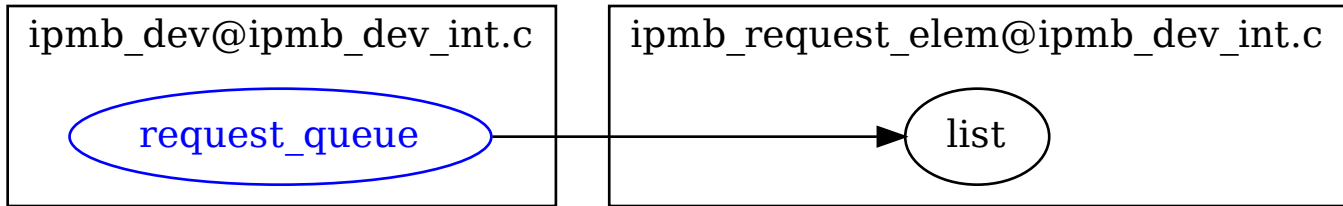
sdei_list

sdei_event@arm_sdei.c

list

```
graph LR; subgraph var_arm_sdei_c [var@arm_sdei.c]; sdei_list((sdei_list)); end; subgraph sdei_event_arm_sdei_c [sdei_event@arm_sdei.c]; list((list)); end; sdei_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@arm_sdei.c' contains an oval labeled 'sdei_list'. On the right, a box labeled 'sdei_event@arm_sdei.c' contains an oval labeled 'list'. A horizontal arrow points from the 'sdei_list' oval to the 'list' oval, indicating that 'sdei_list' points to 'list'.



coh901318_chan@coh901318.c

queue

free

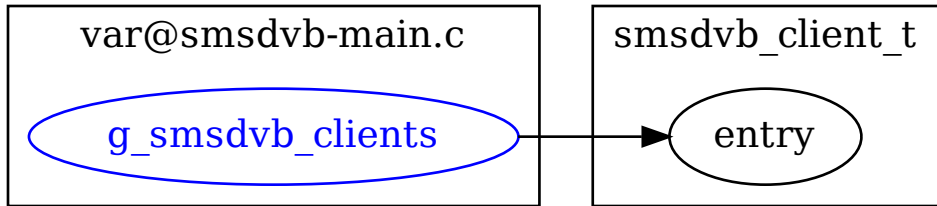
active

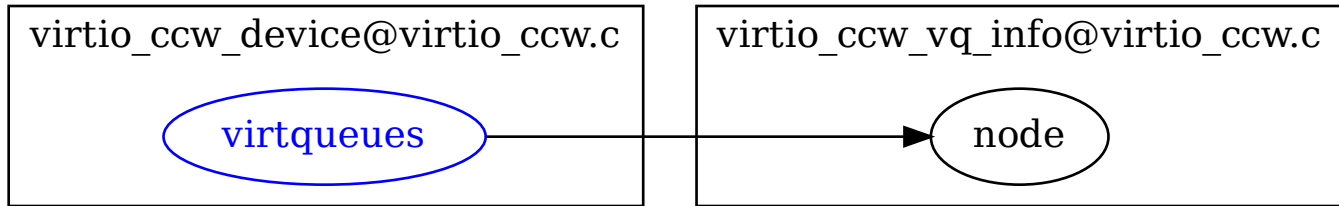
coh901318_desc@coh901318.c

node

```
graph LR; subgraph Box1 [coh901318_chan@coh901318.c]; queue; free; active; end; subgraph Box2 [coh901318_desc@coh901318.c]; node; end; queue --> node; free --> node; active --> node;
```

The diagram illustrates a memory layout with two rectangular containers. The left container, labeled 'coh901318_chan@coh901318.c', contains three blue-outlined ovals labeled 'queue', 'free', and 'active'. The right container, labeled 'coh901318_desc@coh901318.c', contains a single black-outlined oval labeled 'node'. Three curved arrows originate from the 'queue', 'free', and 'active' ovals and point to the 'node' oval, indicating that these three pointers reference the same memory location.





var@smsgiucv_app.c

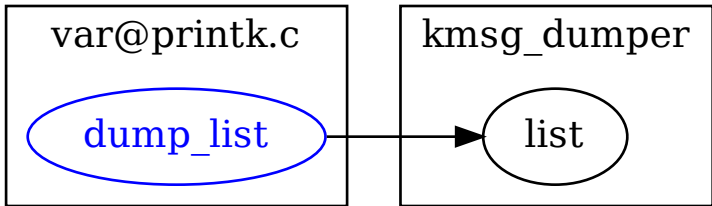
smsg_event_queue

smsg_app_event@smsgiucv_app.c

list

```
graph LR; subgraph File1 [var@smsgiucv_app.c]; direction TB; V1[smsg_event_queue]; end; subgraph File2 [smsg_app_event@smsgiucv_app.c]; direction TB; V2((list)); end; V1 --> V2;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@smsgiucv_app.c`. Inside this box, the variable `smsg_event_queue` is enclosed in a blue oval. On the right, another rectangular box represents the source file `smsg_app_event@smsgiucv_app.c`. Inside this box, the variable `list` is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that `smsg_event_queue` points to `list`.

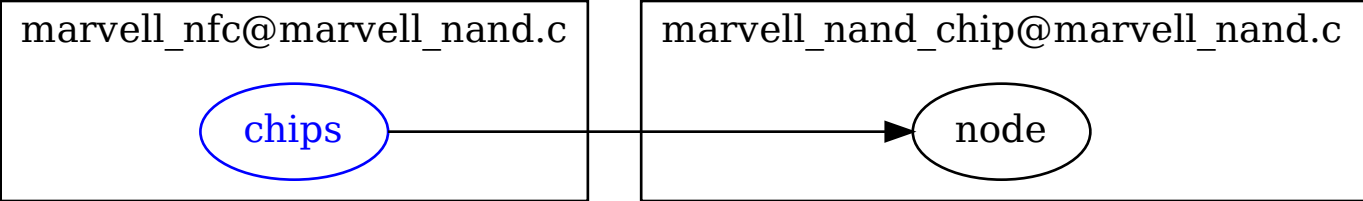


marvell_nfc@marvell_nand.c

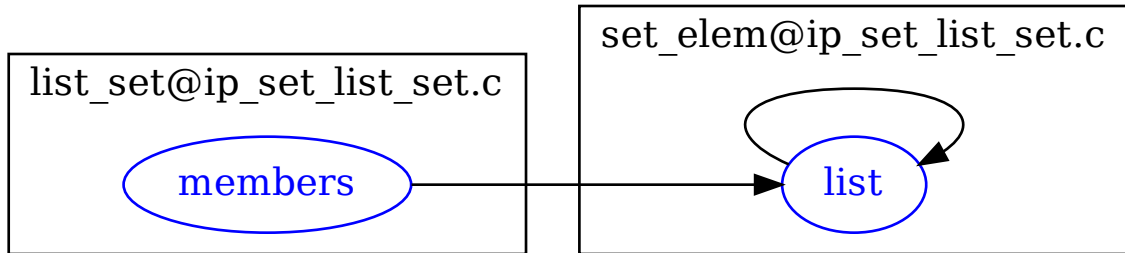
chips

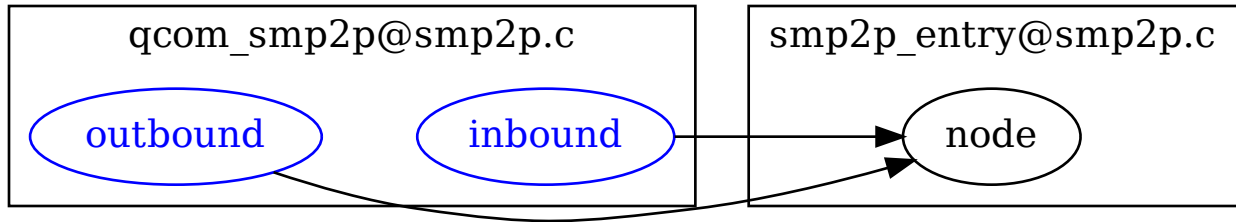
marvell_nand_chip@marvell_nand.c

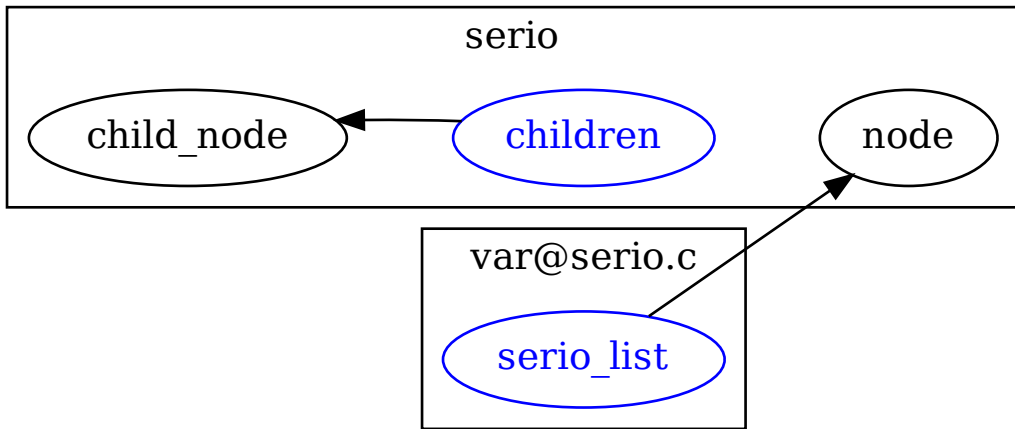
node

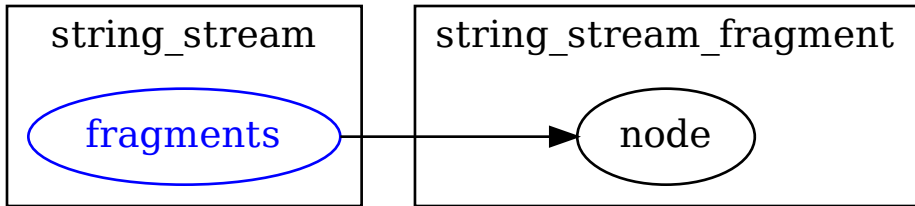


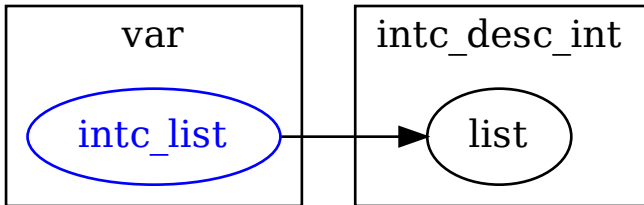
```
graph LR; subgraph LeftBox [marvell_nfc@marvell_nand.c]; chips((chips)); end; subgraph RightBox [marvell_nand_chip@marvell_nand.c]; node((node)); end; chips --> node;
```

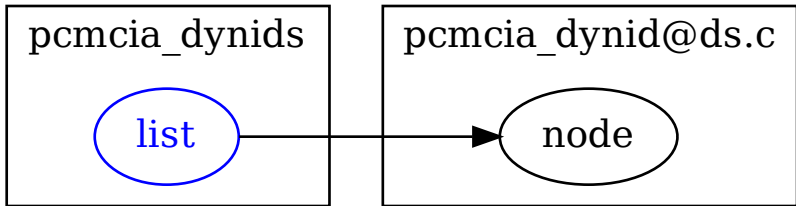












table@mkregtable.c

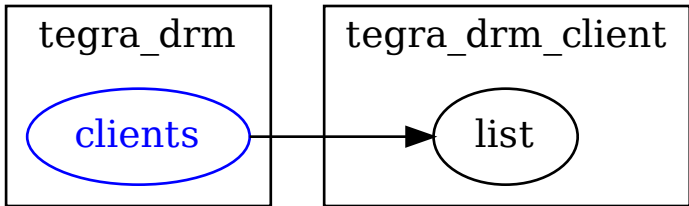
offsets

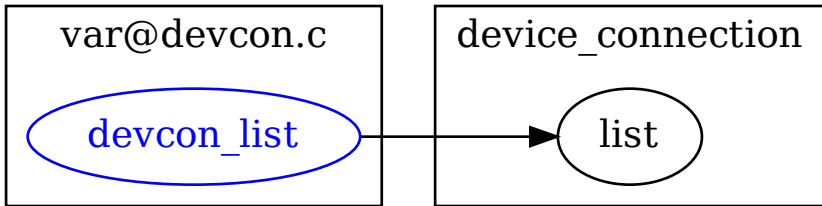
offset@mkregtable.c

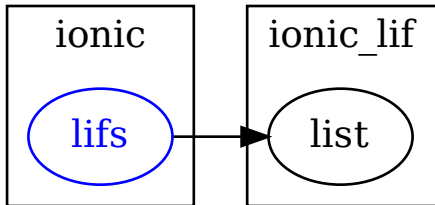
list

```
graph LR; A["table@mkregtable.c  
offsets"] --> B["offset@mkregtable.c  
list"]
```

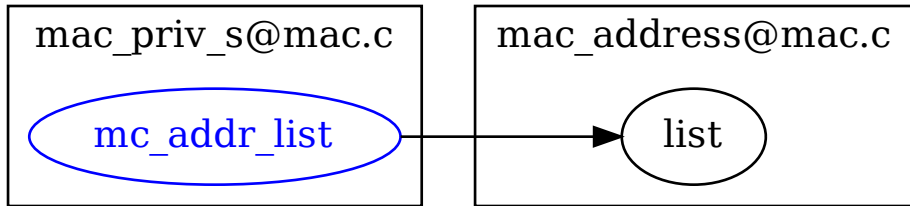
The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'table@mkregtable.c' contains an oval labeled 'offsets' with a blue border. On the right, a box labeled 'offset@mkregtable.c' contains an oval labeled 'list' with a black border. A horizontal arrow points from the 'offsets' oval to the 'list' oval, indicating that the 'offsets' variable in the first module points to the 'list' variable in the second module.



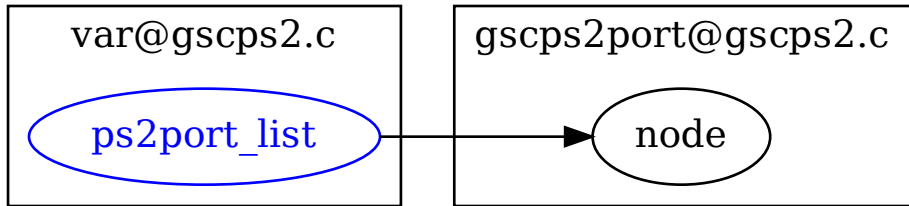


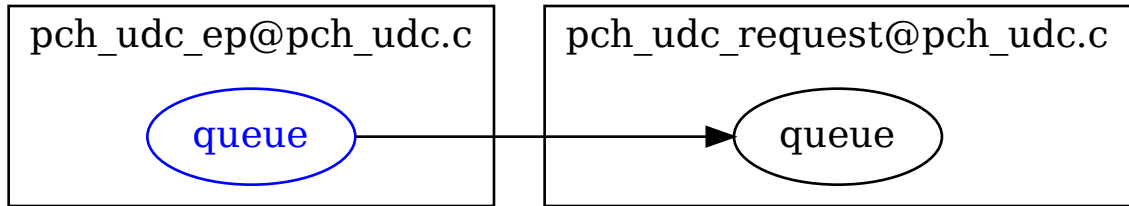




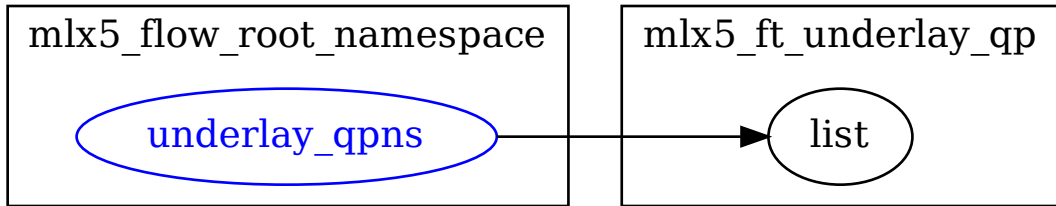


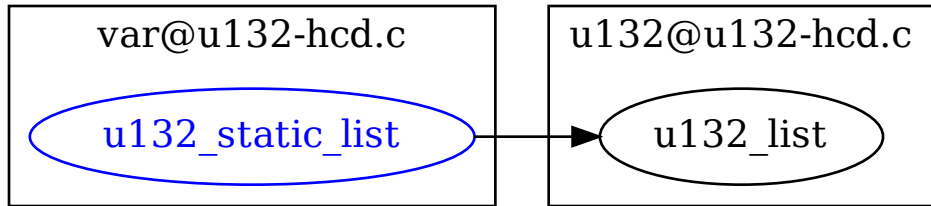


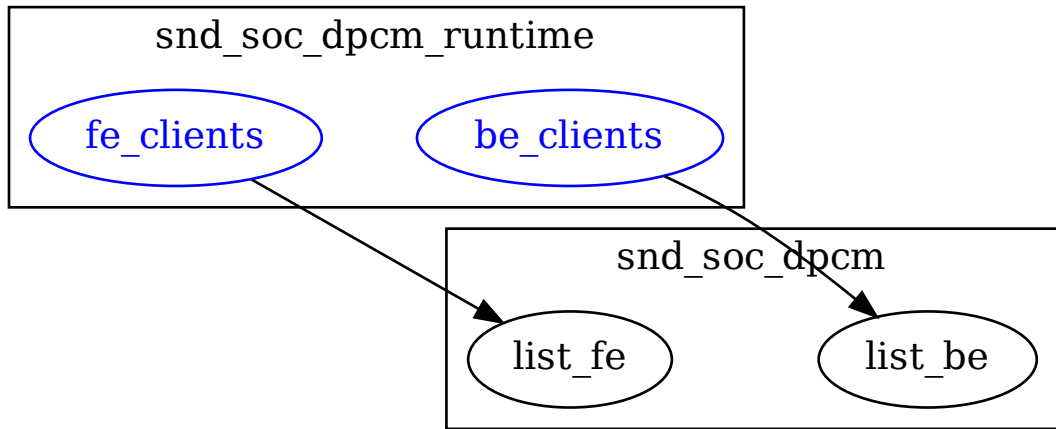


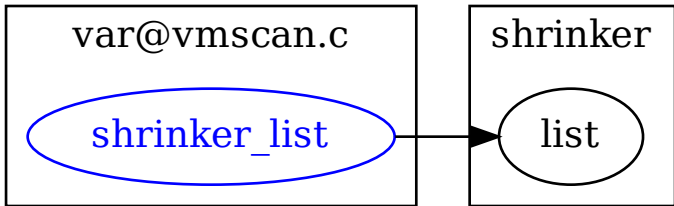












wf_fcu_priv@windfarm_fcu_controls.c

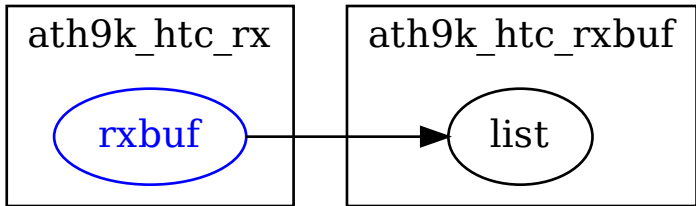
fan_list

wf_fcu_fan@windfarm_fcu_controls.c

link

```
graph LR; fan_list((fan_list)) --> link((link));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'wf_fcu_priv@windfarm_fcu_controls.c' contains a blue oval labeled 'fan_list'. On the right, a box labeled 'wf_fcu_fan@windfarm_fcu_controls.c' contains a black oval labeled 'link'. A horizontal arrow points from the 'fan_list' oval to the 'link' oval, indicating that 'fan_list' points to 'link'.



var@mac80211_hwsim.c

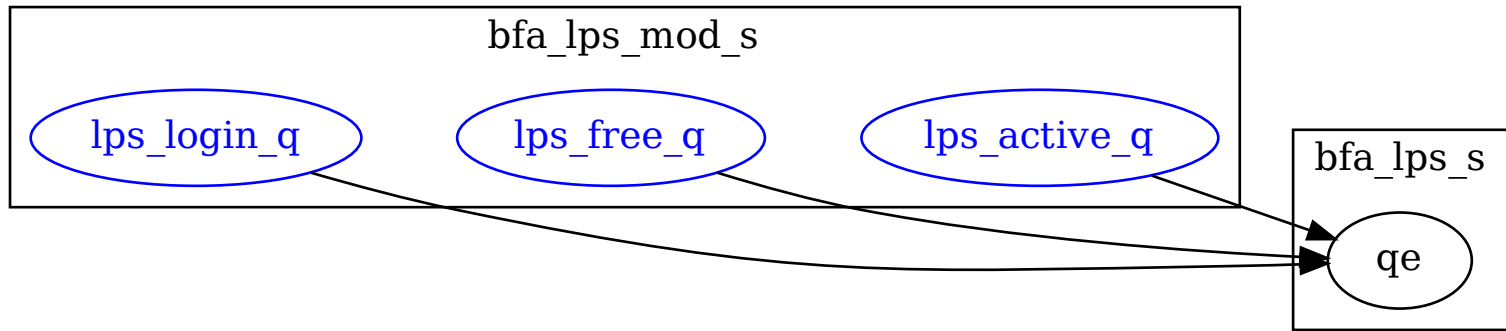
hwsim_radios

mac80211_hwsim_data@mac80211_hwsim.c

list

```
graph LR; A[var@mac80211_hwsim.c] --- B[mac80211_hwsim_data@mac80211_hwsim.c]; B --> C((list));
```

The diagram illustrates a pointer relationship between two code locations. On the left, a box labeled 'var@mac80211_hwsim.c' contains the variable 'hwsim_radios', which is highlighted with a blue oval. On the right, a box labeled 'mac80211_hwsim_data@mac80211_hwsim.c' contains a pointer 'list' inside a black oval. A horizontal arrow points from the 'hwsim_radios' oval to the 'list' oval, indicating that the variable points to the list structure.



var@omapdss-boot-init.c

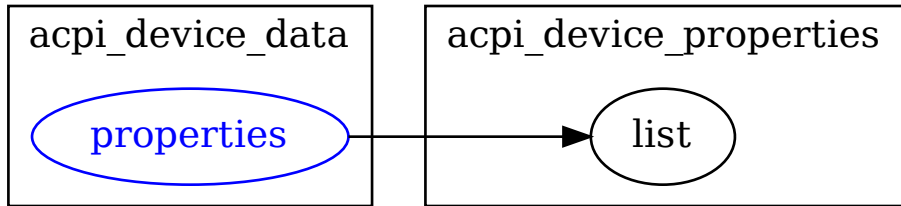
dss_conv_list

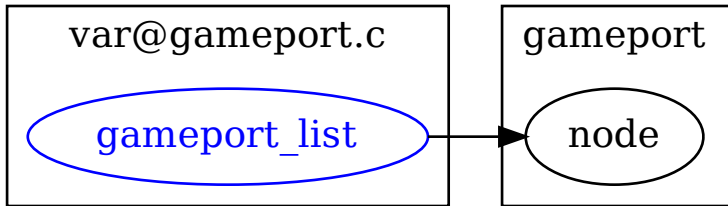
dss_conv_node@omapdss-boot-init.c

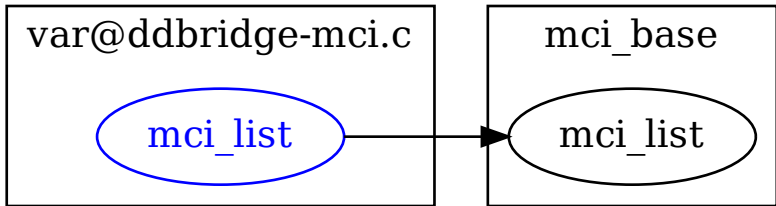
list

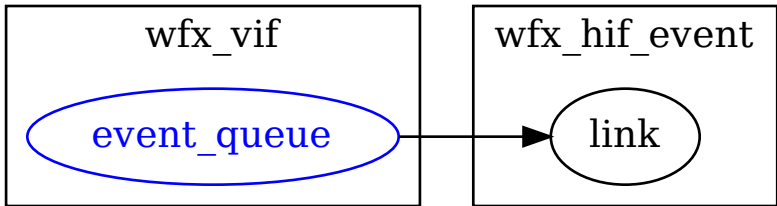
```
graph LR; A([dss_conv_list]) --> B([list]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@omapdss-boot-init.c' contains a blue oval labeled 'dss_conv_list'. On the right, a box labeled 'dss_conv_node@omapdss-boot-init.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'dss_conv_list' oval to the 'list' oval, indicating that 'dss_conv_list' points to 'list'.









var@aha152x.c

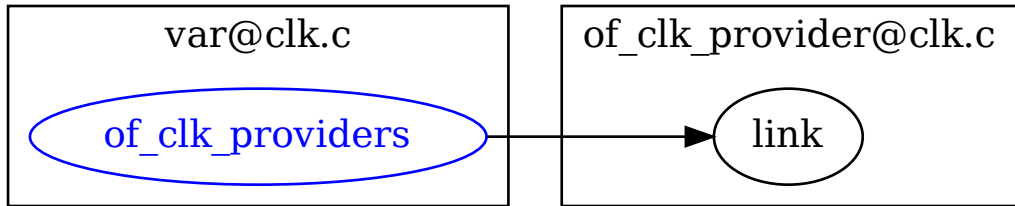
aha152x_host_list

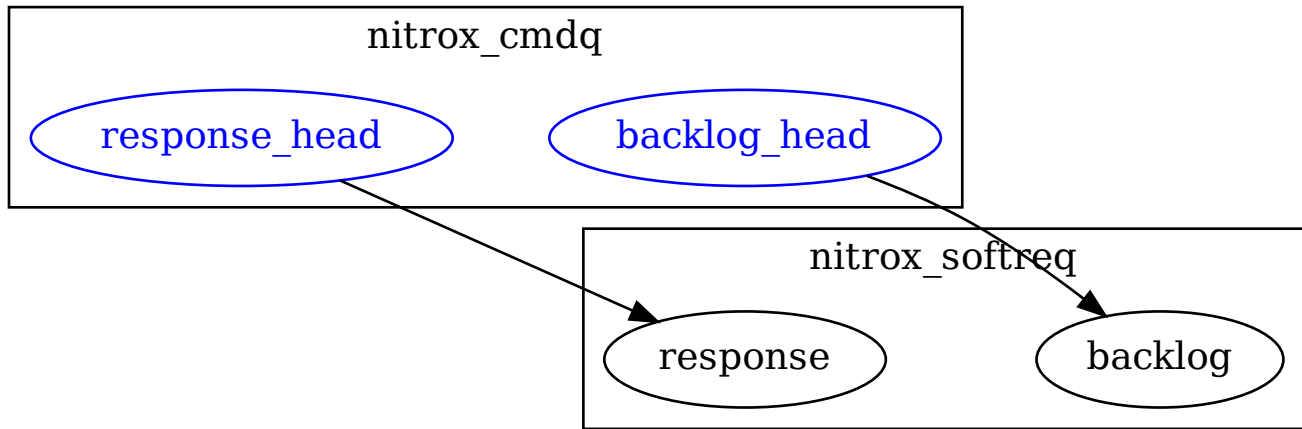
aha152x_hostdata@aha152x.c

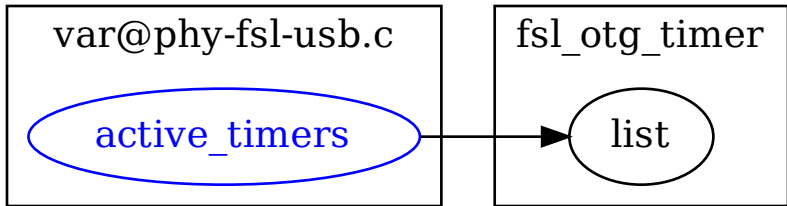
host_list

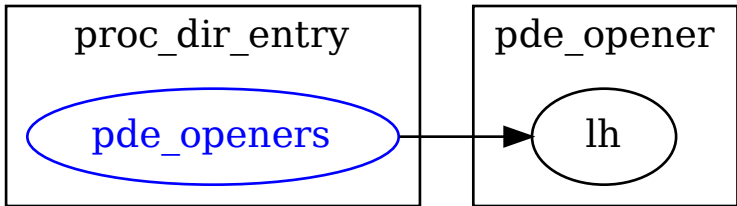
```
graph LR; A[aha152x_host_list] --> B[host_list]
```

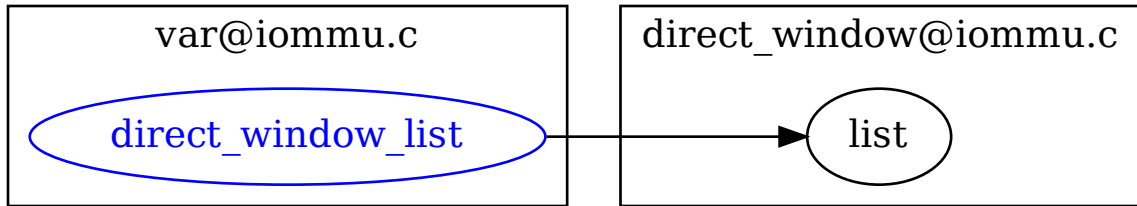
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@aha152x.c' at the top and 'aha152x_host_list' at the bottom. The 'aha152x_host_list' text is enclosed in a blue oval. On the right, another rectangular box contains the text 'aha152x_hostdata@aha152x.c' at the top and 'host_list' at the bottom. The 'host_list' text is enclosed in a black oval. A horizontal arrow points from the blue oval on the left to the black oval on the right, indicating that 'aha152x_host_list' points to 'host_list'.











var@heartbeat.c

o2hb_node_events

o2hb_node_event@heartbeat.c

hn_item

```
graph LR; subgraph Box1 [var@heartbeat.c]; direction TB; A(o2hb_node_events); end; subgraph Box2 [o2hb_node_event@heartbeat.c]; direction TB; B(hn_item); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@heartbeat.c' contains an oval labeled 'o2hb_node_events' with a blue border. On the right, a rectangular box labeled 'o2hb_node_event@heartbeat.c' contains an oval labeled 'hn_item' with a black border. A horizontal arrow points from the right side of the 'o2hb_node_events' oval to the left side of the 'hn_item' oval, indicating that 'o2hb_node_events' points to 'hn_item'.

var@psmouse-smbus.c

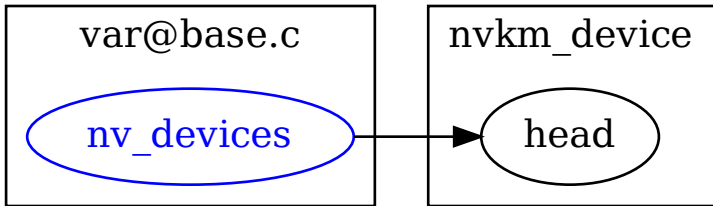
psmouse_smbus_list

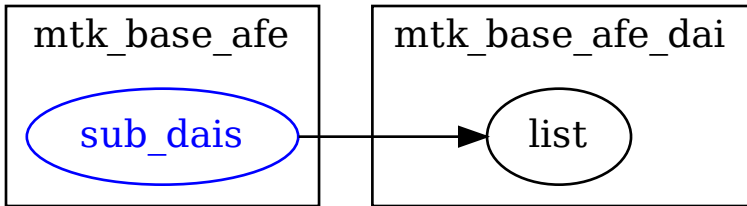
psmouse_smbus_dev@psmouse-smbus.c

node

```
graph LR; subgraph Box1 [var@psmouse-smbus.c]; direction TB; A([psmouse_smbus_list]); end; subgraph Box2 [psmouse_smbus_dev@psmouse-smbus.c]; direction TB; B([node]); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@psmouse-smbus.c' and contains an oval labeled 'psmouse_smbus_list' in blue text. The right box is labeled 'psmouse_smbus_dev@psmouse-smbus.c' and contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'psmouse_smbus_list' oval to the left side of the 'node' oval.





var@abx500-core.c

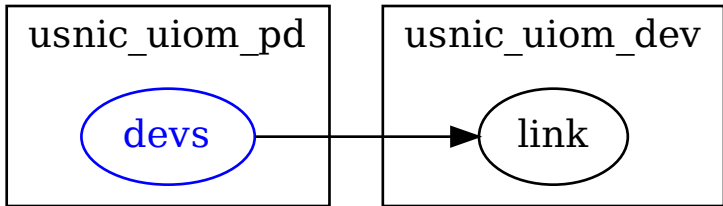
abx500_list

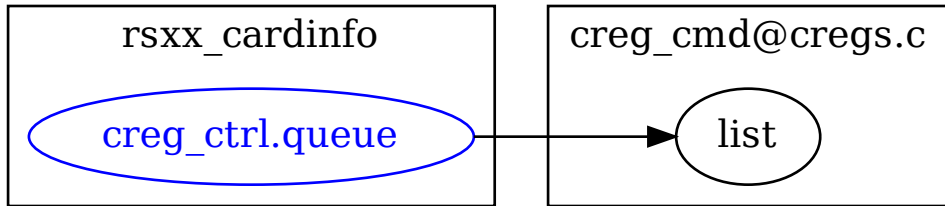
abx500_device_entry@abx500-core.c

list

```
graph LR; A[abx500_list] --> B((list));
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@abx500-core.c' at the top and 'abx500_list' in a blue oval below it. The right box contains the text 'abx500_device_entry@abx500-core.c' at the top and a black oval containing the text 'list' below it. A horizontal arrow points from the right side of the 'abx500_list' oval to the left side of the 'list' oval.





var@of_reserved_mem.c

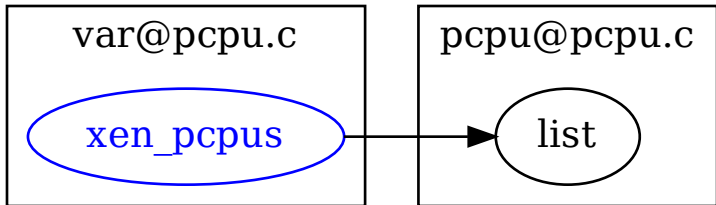
of_rmem_assigned_device_list

rmem_assigned_device@of_reserved_mem.c

list

```
graph LR; A[of_rmem_assigned_device_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@of_reserved_mem.c`. Inside this box, the variable `of_rmem_assigned_device_list` is highlighted with a blue oval. On the right, another rectangular box represents the source file `rmem_assigned_device@of_reserved_mem.c`. Inside this box, there is an oval labeled `list`. A horizontal arrow points from the blue oval in the first box to the oval in the second box, indicating that `of_rmem_assigned_device_list` points to `list`.



flow_head@cls_flow.c

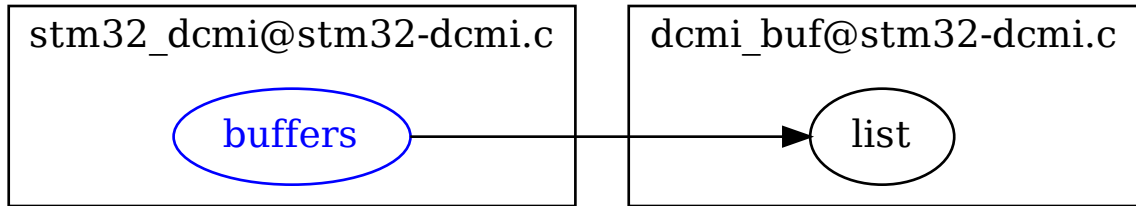
filters

flow_filter@cls_flow.c

list

```
graph LR; filters([filters]) --> list([list]);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'flow_head@cls_flow.c' contains an oval labeled 'filters'. On the right, a box labeled 'flow_filter@cls_flow.c' contains an oval labeled 'list'. A horizontal arrow points from the 'filters' oval to the 'list' oval, indicating that the 'filters' variable points to the 'list' structure.



vscsibk_info@xen-scsiback.c

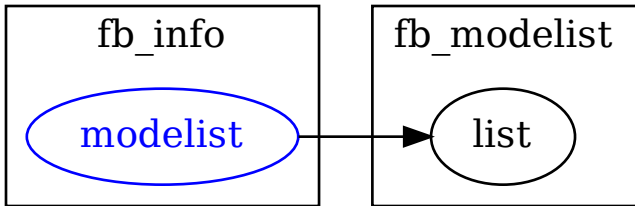
v2p_entry_lists

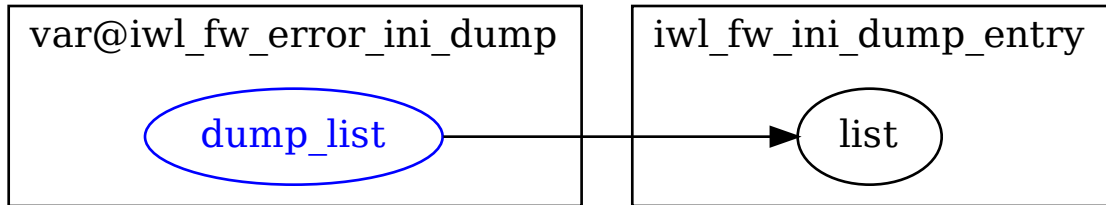
v2p_entry@xen-scsiback.c

1

```
graph LR; A["vscsibk_info@xen-scsiback.c"] --> B["v2p_entry@xen-scsiback.c"]; B --> C((1));
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'vscsibk_info@xen-scsiback.c' contains an oval labeled 'v2p_entry_lists' in blue. On the right, a box labeled 'v2p_entry@xen-scsiback.c' contains an oval labeled '1'. A horizontal arrow points from the 'v2p_entry_lists' oval to the '1' oval, indicating that the 'v2p_entry_lists' variable points to the memory location '1' within the 'v2p_entry@xen-scsiback.c' module.





var@edac_device.c

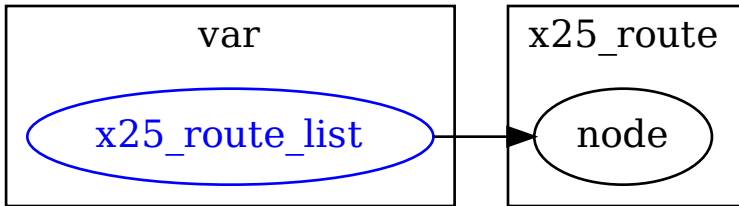
edac_device_list

edac_device_ctl_info

link

```
graph LR; A[edac_device_list] --> B(link);
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@edac_device.c' at the top and 'edac_device_list' in the center, which is enclosed in a blue oval. The right box contains the text 'edac_device_ctl_info' at the top and 'link' in the center, which is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.



var@omap_hwmod.c

clkctrl_providers

clkctrl_provider@omap_hwmod.c

link

```
graph LR; A[var@omap_hwmod.c] --> B((link)); B --- C[clkctrl_provider@omap_hwmod.c];
```

The diagram illustrates a relationship between two modules. On the left, a box labeled 'var@omap_hwmod.c' contains an oval labeled 'clkctrl_providers' in blue text. A horizontal arrow points from this oval to an oval labeled 'link' inside a box on the right labeled 'clkctrl_provider@omap_hwmod.c'.

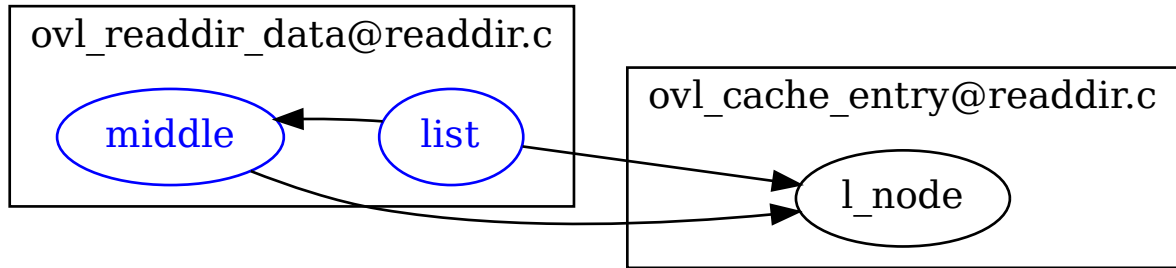
omap2_mcspi_regs@spi-omap2-mcspi.c

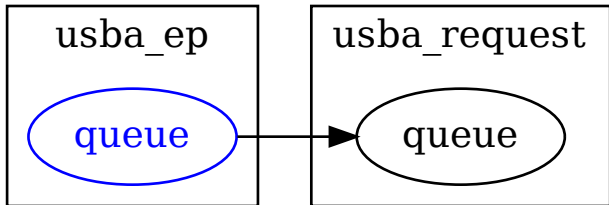
cs

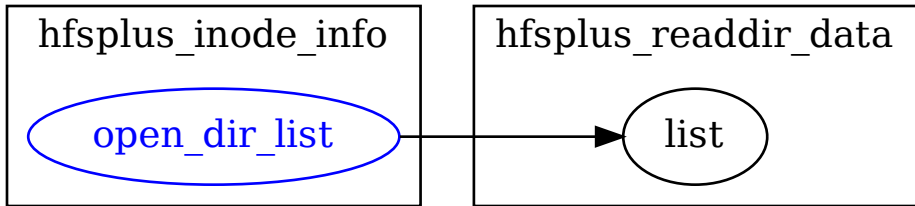
omap2_mcspi_cs@spi-omap2-mcspi.c

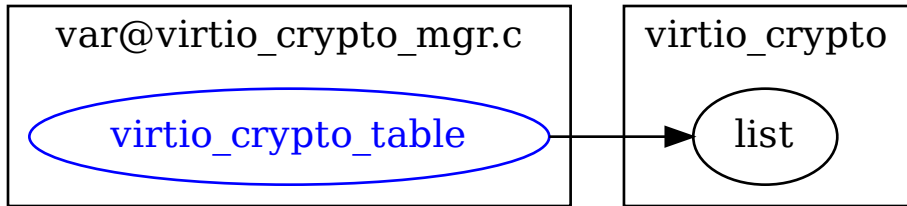
node











var@netconsole.c

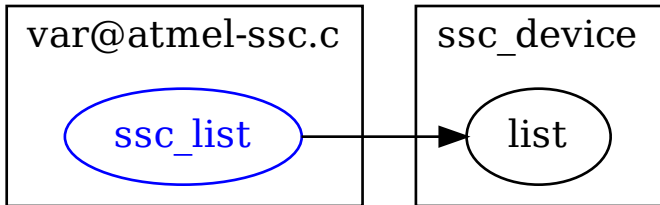
target_list

netconsole_target@netconsole.c

list

```
graph LR; A["var@netconsole.c  
target_list"] --> B["netconsole_target@netconsole.c  
list"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@netconsole.c' and a blue oval containing the text 'target_list'. The right box contains the text 'netconsole_target@netconsole.c' and a black oval containing the text 'list'. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box.



img_hash_drv@img-hash.c

dev_list

img_hash_dev@img-hash.c

list

```
graph LR; A[img_hash_drv@img-hash.c] --> B[img_hash_dev@img-hash.c]; subgraph A; C((dev_list)); end; subgraph B; D((list)); end; C --> D;
```

The diagram consists of two rectangular boxes. The left box contains the text 'img_hash_drv@img-hash.c' at the top and a blue oval containing 'dev_list' below it. The right box contains the text 'img_hash_dev@img-hash.c' at the top and a black oval containing 'list' below it. A horizontal arrow points from the right side of the left box to the left side of the right box. A second horizontal arrow points from the blue oval 'dev_list' to the black oval 'list'.

var@mpic_timer.c

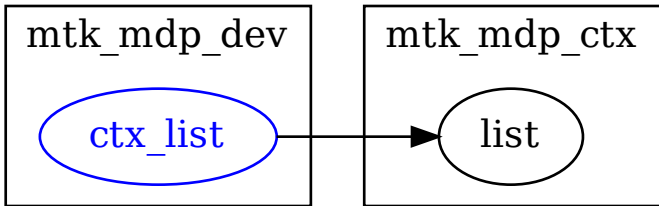
timer_group_list

timer_group_priv@mpic_timer.c

node

```
graph LR; subgraph var_mplic_timer_c [var@mpic_timer.c]; timer_group_list; end; subgraph timer_group_priv_mplic_timer_c [timer_group_priv@mpic_timer.c]; node; end; timer_group_list --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@mpic_timer.c' contains an oval labeled 'timer_group_list'. On the right, a box labeled 'timer_group_priv@mpic_timer.c' contains an oval labeled 'node'. A horizontal arrow points from the 'timer_group_list' oval to the 'node' oval, indicating that 'timer_group_list' points to 'node'.



ingenic_nfc@ingenic_nand_drv.c

chips

ingenic_nand@ingenic_nand_drv.c

chip_list

```
graph LR; A[ingenic_nfc@ingenic_nand_drv.c] --- B[ingenic_nand@ingenic_nand_drv.c]; subgraph A; C((chips)); end; subgraph B; D((chip_list)); end; C --> D;
```

The diagram consists of two rectangular boxes. The left box contains the text 'ingenic_nfc@ingenic_nand_drv.c' and an oval labeled 'chips'. The right box contains the text 'ingenic_nand@ingenic_nand_drv.c' and an oval labeled 'chip_list'. A horizontal arrow points from the 'chips' oval in the left box to the 'chip_list' oval in the right box.

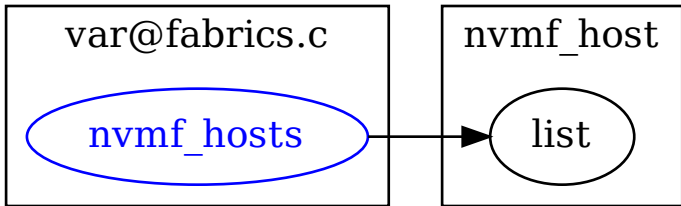
var@regulator-quirk-rcar-gen2.c

quirk_list

regulator_quirk@regulator-quirk-rcar-gen2.c

list





solos_card@solos-pci.c

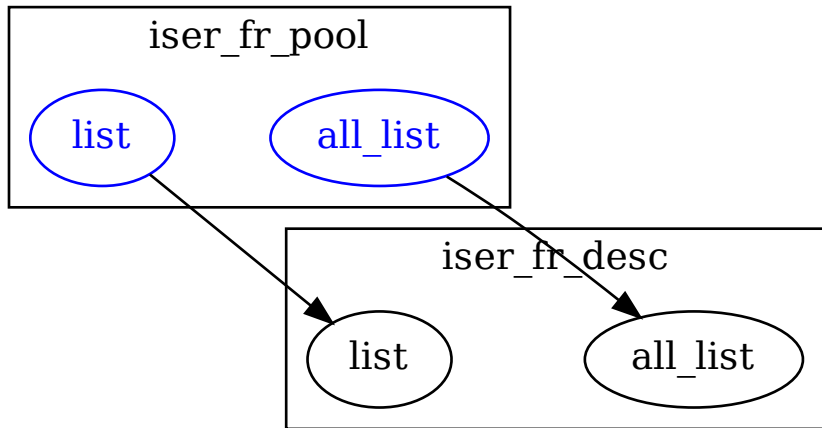
param_queue

solos_param@solos-pci.c

list

```
graph LR; subgraph solos_card [solos_card@solos-pci.c]; param_queue; end; subgraph solos_param [solos_param@solos-pci.c]; list; end; param_queue --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'solos_card@solos-pci.c' contains an oval labeled 'param_queue'. On the right, a box labeled 'solos_param@solos-pci.c' contains an oval labeled 'list'. A horizontal arrow points from the 'param_queue' oval to the 'list' oval, indicating that 'param_queue' points to 'list'.



ext4_getfsmap_info@fsmap.c

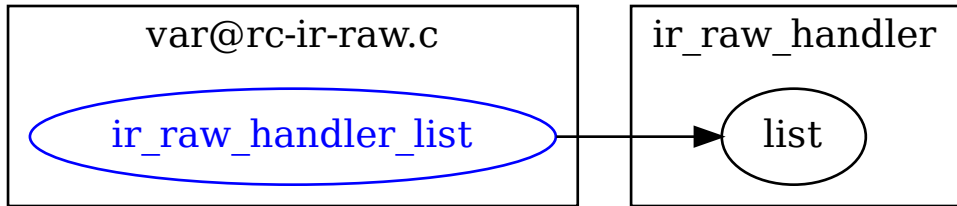
gfi_meta_list

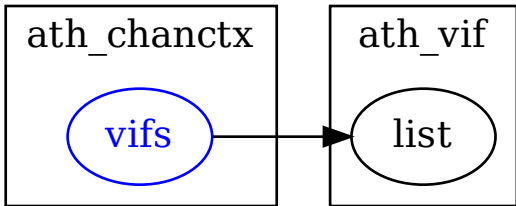
ext4_fsmap

fmr_list

```
graph LR; A[ext4_getfsmap_info@fsmap.c] --> B[ext4_fsmap]; subgraph A [ext4_getfsmap_info@fsmap.c]; C([gfi_meta_list]); end; subgraph B [ext4_fsmap]; D([fmr_list]); end; C --> D;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'ext4_getfsmap_info@fsmap.c' contains an oval labeled 'gfi_meta_list'. On the right, a box labeled 'ext4_fsmap' contains an oval labeled 'fmr_list'. A horizontal arrow points from the 'gfi_meta_list' oval to the 'fmr_list' oval, indicating that 'gfi_meta_list' points to 'fmr_list'.





var@bpqether.c

bpq_devices

bpqdev@bpqether.c

bpq_list

```
graph LR; subgraph var_bpqether_c [var@bpqether.c]; bpq_devices; end; subgraph bpqdev_bpqether_c [bpqdev@bpqether.c]; bpq_list; end; bpq_devices --> bpq_list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@bpqether.c' contains an oval labeled 'bpq_devices' with a blue border. On the right, a box labeled 'bpqdev@bpqether.c' contains an oval labeled 'bpq_list' with a black border. A horizontal arrow points from the right side of the 'bpq_devices' oval to the left side of the 'bpq_list' oval, indicating that 'bpq_devices' points to 'bpq_list'.

var@soc_camera.c

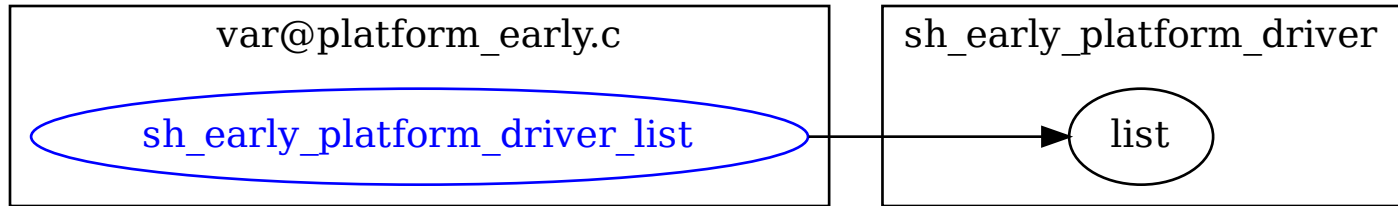
devices

soc_camera_device

list

```
graph LR; subgraph var_at_soc_camera_c [var@soc_camera.c]; devices; end; subgraph soc_camera_device_struct [soc_camera_device]; list; end; devices --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@soc_camera.c' contains an oval labeled 'devices'. On the right, a box labeled 'soc_camera_device' contains an oval labeled 'list'. A horizontal arrow points from the 'devices' oval to the 'list' oval, indicating that the 'devices' variable points to the 'list' member of the 'soc_camera_device' structure.



var@drop_monitor.c

hw_stats_list

dm_hw_stat_delta@drop_monitor.c

list

```
graph LR; A[hw_stats_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@drop_monitor.c' and a blue oval containing 'hw_stats_list'. On the right, another rectangular box contains the text 'dm_hw_stat_delta@drop_monitor.c' and a black oval containing 'list'. A horizontal arrow points from the blue oval 'hw_stats_list' to the black oval 'list', indicating that 'hw_stats_list' points to 'list'.

mlx5_fpga_ipsec@ipsec.c

pending_cmds

mlx5_fpga_ipsec_cmd_context@ipsec.c

list

```
graph LR; A[mlx5_fpga_ipsec@ipsec.c] --> B((pending_cmds)); B --> C[mlx5_fpga_ipsec_cmd_context@ipsec.c]; C --> D((list));
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'mlx5_fpga_ipsec@ipsec.c' contains an oval labeled 'pending_cmds'. On the right, a rectangular box labeled 'mlx5_fpga_ipsec_cmd_context@ipsec.c' contains an oval labeled 'list'. A horizontal arrow points from the 'pending_cmds' oval to the 'list' oval, indicating that the 'pending_cmds' variable points to the 'list' structure.

var@heartbeat.c

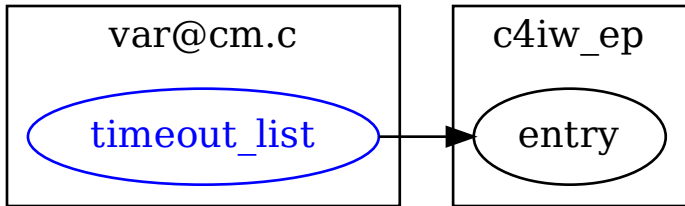
o2hb_live_slots

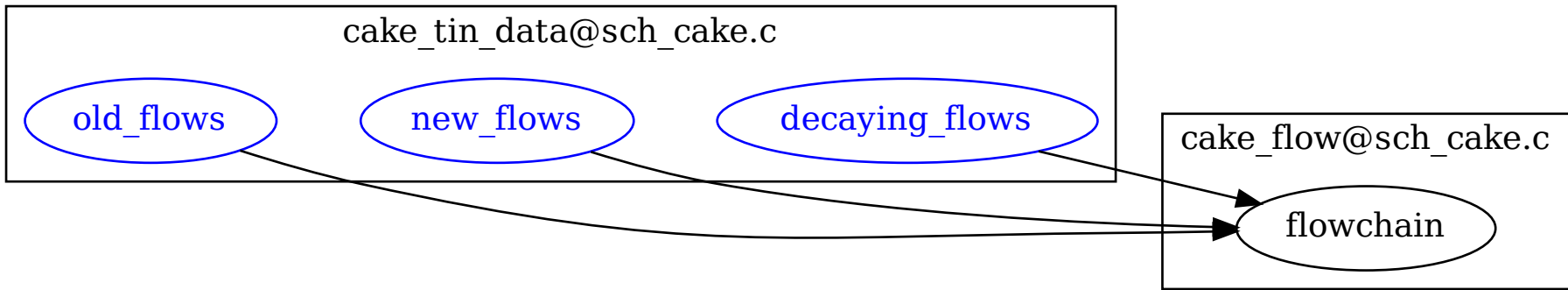
o2hb_disk_slot@heartbeat.c

ds_live_item

```
graph LR; A([o2hb_live_slots]) --> B([ds_live_item]);
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@heartbeat.c' contains an oval labeled 'o2hb_live_slots' with a blue border. On the right, a rectangular box labeled 'o2hb_disk_slot@heartbeat.c' contains an oval labeled 'ds_live_item' with a black border. A horizontal arrow points from the right side of the 'o2hb_live_slots' oval to the left side of the 'ds_live_item' oval, indicating that 'o2hb_live_slots' points to 'ds_live_item'.





var@n2_core.c

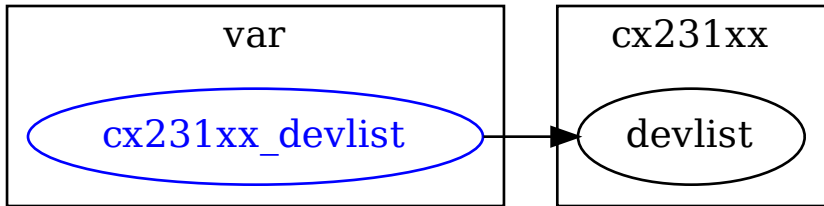
skcipher_algs

n2_skcipher_alg@n2_core.c

entry

```
graph LR; A[skcipher_algs] --> B(entry);
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@n2_core.c' at the top and an oval containing 'skcipher_algs' in blue text. The right box contains the text 'n2_skcipher_alg@n2_core.c' at the top and an oval containing 'entry'. A horizontal arrow points from the right side of the 'skcipher_algs' oval to the left side of the 'entry' oval.



var@fsl_hypervisor.c

db_list

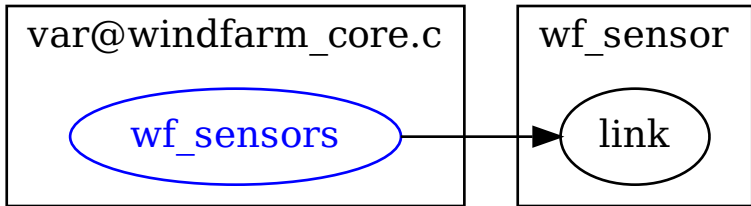
doorbell_queue@fsl_hypervisor.c

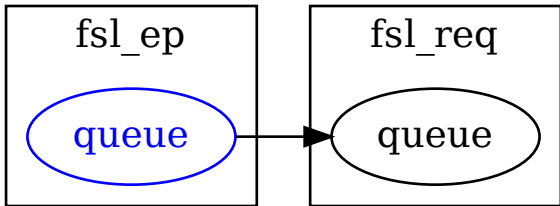
list

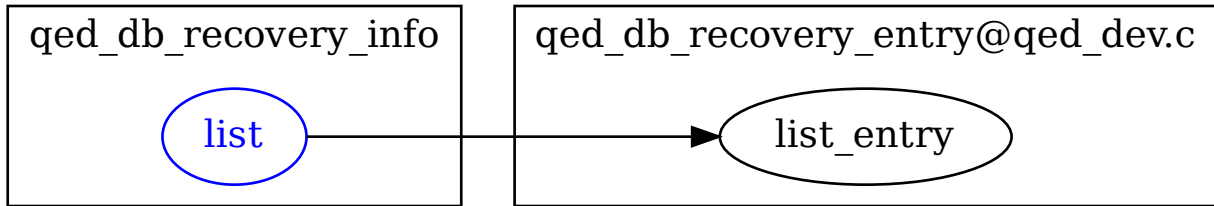


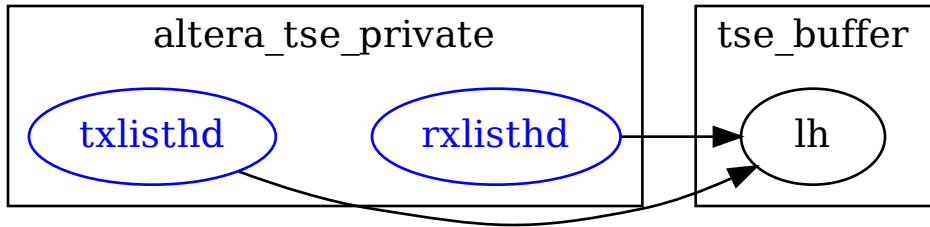
```
graph LR; subgraph var ["var@fsl_hypervisor.c"]; db_list((db_list)); end; subgraph doorbell_queue ["doorbell_queue@fsl_hypervisor.c"]; list((list)); end; db_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@fsl_hypervisor.c' contains an oval labeled 'db_list'. On the right, a box labeled 'doorbell_queue@fsl_hypervisor.c' contains an oval labeled 'list'. A horizontal arrow points from the 'db_list' oval to the 'list' oval, indicating that 'db_list' points to 'list'.









l2cache_pmu@qcom_l2_pmu.c

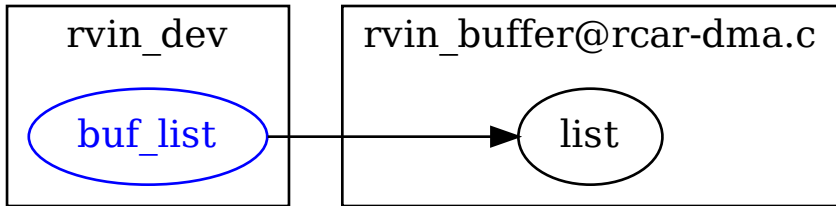
clusters

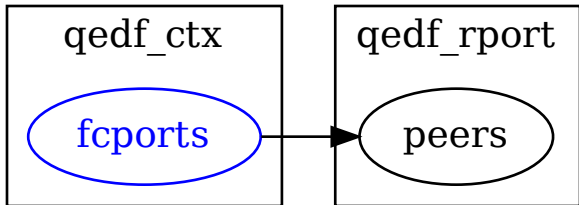
cluster_pmu@qcom_l2_pmu.c

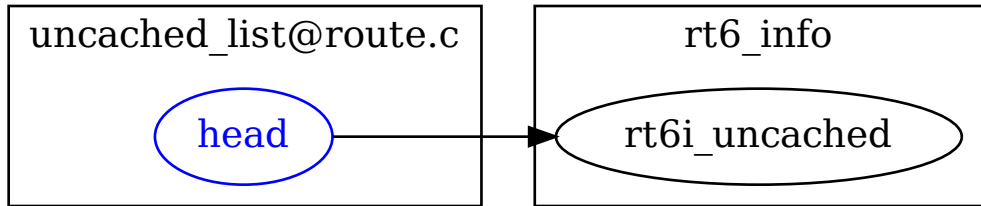
next

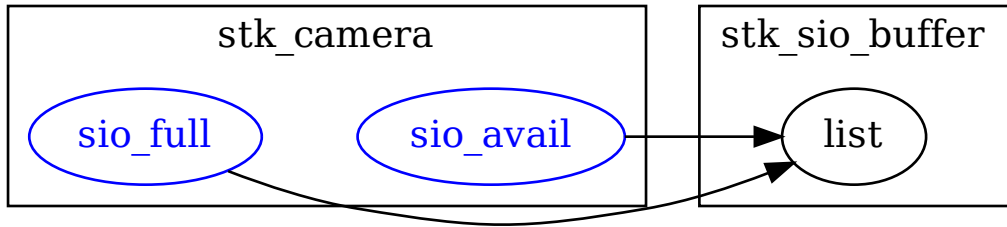
```
graph LR; subgraph Module1 [l2cache_pmu@qcom_l2_pmu.c]; direction TB; C([clusters]); end; subgraph Module2 [cluster_pmu@qcom_l2_pmu.c]; direction TB; N([next]); end; C --> N;
```

The diagram illustrates a pointer relationship between two PMU modules. On the left, a box labeled 'l2cache_pmu@qcom_l2_pmu.c' contains a blue oval labeled 'clusters'. On the right, a box labeled 'cluster_pmu@qcom_l2_pmu.c' contains a black oval labeled 'next'. A horizontal arrow points from the 'clusters' oval to the 'next' oval, indicating that the 'clusters' variable in the first module points to the 'next' variable in the second module.









var@nf_flow_table_offload.c

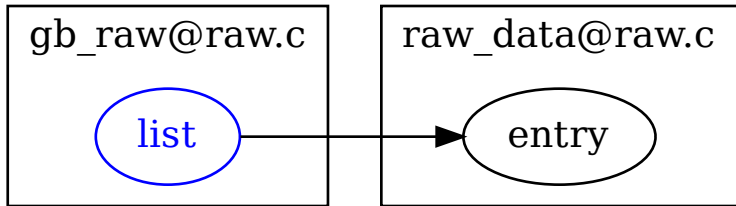
flow_offload_pending_list

flow_offload_work@nf_flow_table_offload.c

list

```
graph LR; A[flow_offload_pending_list] --> B((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box contains the text 'var@nf_flow_table_offload.c'. Inside this box, the variable 'flow_offload_pending_list' is highlighted with a blue oval. A horizontal arrow points from the right side of this oval to a black oval containing the text 'list'. This 'list' oval is located within a larger rectangular box on the right, which also contains the text 'flow_offload_work@nf_flow_table_offload.c' at the top.



var@trace.c

postponed_selftests

trace_selftests@trace.c

list

```
graph LR; subgraph var_trace_c [var@trace.c]; postponed_selftests; end; subgraph trace_selftests_trace_c [trace_selftests@trace.c]; list; end; postponed_selftests --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@trace.c' contains an oval labeled 'postponed_selftests'. On the right, a rectangular box labeled 'trace_selftests@trace.c' contains an oval labeled 'list'. A horizontal arrow points from the 'postponed_selftests' oval to the 'list' oval, indicating that 'postponed_selftests' points to 'list'.

var@irqbypass.c

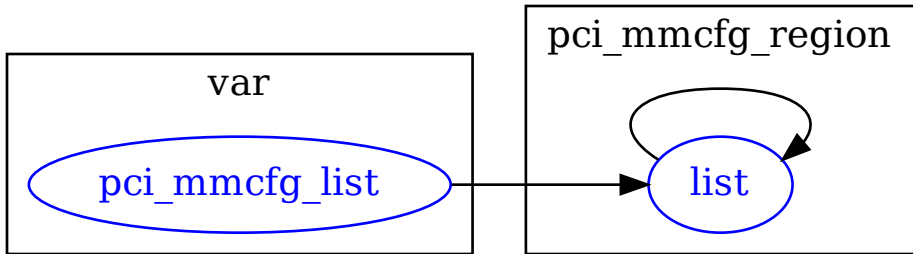
producers

irq_bypass_producer

node

```
graph LR; subgraph var_at_irqbypass_c [var@irqbypass.c]; producers; end; subgraph irq_bypass_producer; node; end; producers --> node;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@irqbypass.c' contains an oval labeled 'producers'. On the right, a box labeled 'irq_bypass_producer' contains an oval labeled 'node'. A horizontal arrow points from the 'producers' oval to the 'node' oval, indicating that the 'producers' variable points to the 'node' object.



var@target_core_hba.c

backend_list

target_backend

list

```
graph LR; subgraph var_target_core_hba_c [var@target_core_hba.c]; backend_list; end; subgraph target_backend [target_backend]; list; end; backend_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@target_core_hba.c' contains an oval labeled 'backend_list'. On the right, a box labeled 'target_backend' contains an oval labeled 'list'. A horizontal arrow points from the 'backend_list' oval to the 'list' oval, indicating that 'backend_list' points to 'list'.

var@sa_query.c

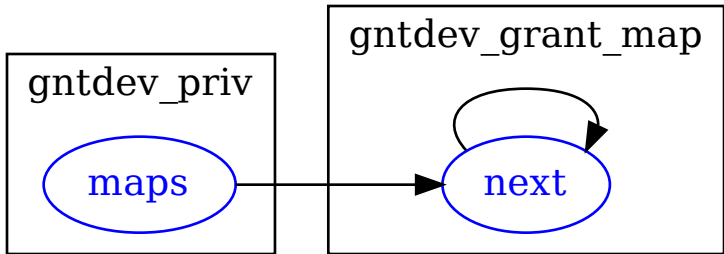
ib_nl_request_list

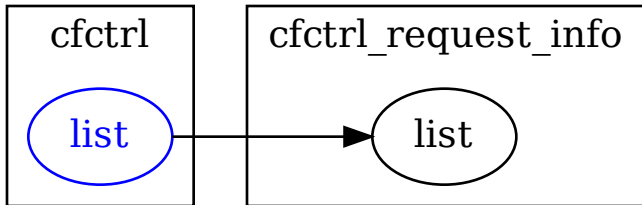
ib_sa_query@sa_query.c

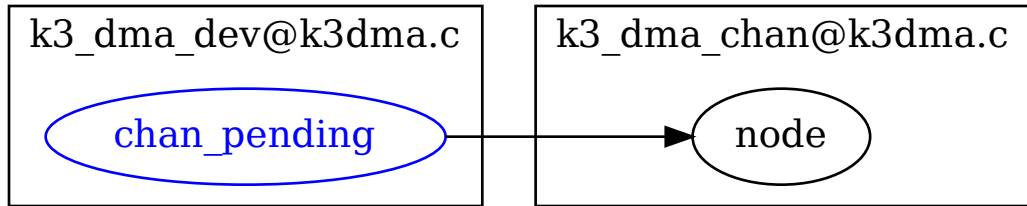
list

```
graph LR; subgraph var_sa_query_c [var@sa_query.c]; ib_nl_request_list; end; subgraph ib_sa_query_sa_query_c [ib_sa_query@sa_query.c]; list; end; ib_nl_request_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@sa_query.c' contains the variable 'ib_nl_request_list', which is highlighted with a blue oval. On the right, a box labeled 'ib_sa_query@sa_query.c' contains a variable 'list' inside an oval. A horizontal arrow points from the 'ib_nl_request_list' oval to the 'list' oval, indicating that the first variable points to the second.







var@alternative.c

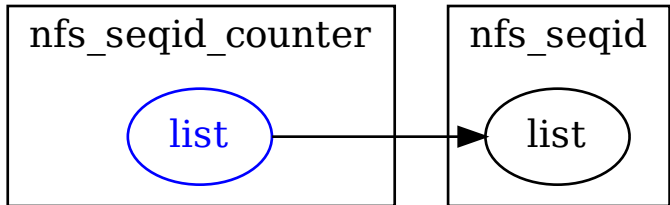
smp_alt_modules

smp_alt_module@alternative.c

next

```
graph LR; subgraph Box1 [var@alternative.c]; A(smp_alt_modules); end; subgraph Box2 [smp_alt_module@alternative.c]; B((next)); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@alternative.c' and contains an oval labeled 'smp_alt_modules' in blue text. The right box is labeled 'smp_alt_module@alternative.c' and contains an oval labeled 'next'. A horizontal arrow points from the right side of the 'smp_alt_modules' oval to the left side of the 'next' oval.



fdb_chain@eswitch_offloads_chains.c

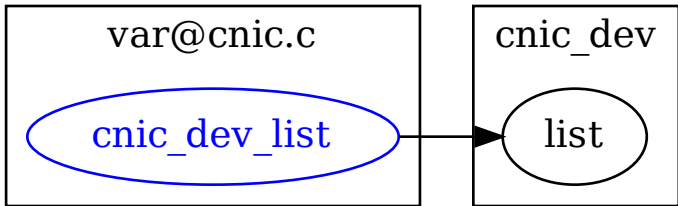
prios_list

fdb_prio@eswitch_offloads_chains.c

list

```
graph LR; prios_list((prios_list)) --> list((list));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'fdb_chain@eswitch_offloads_chains.c' contains an oval labeled 'prios_list'. On the right, a box labeled 'fdb_prio@eswitch_offloads_chains.c' contains an oval labeled 'list'. A horizontal arrow points from the 'prios_list' oval to the 'list' oval, indicating that 'prios_list' points to 'list'.



var@dsp_pipeline.c

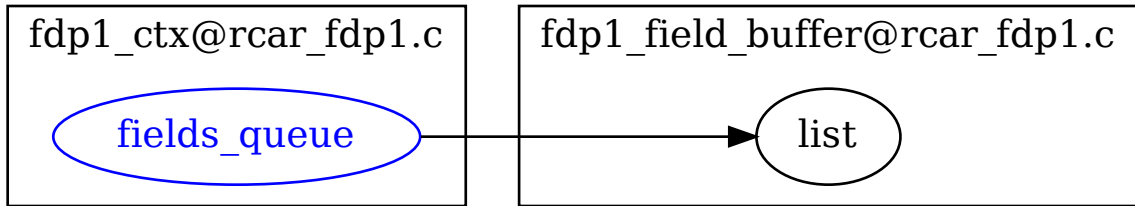
dsp_elements

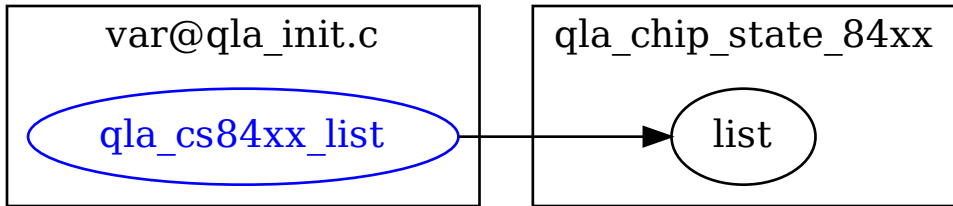
dsp_element_entry@dsp_pipeline.c

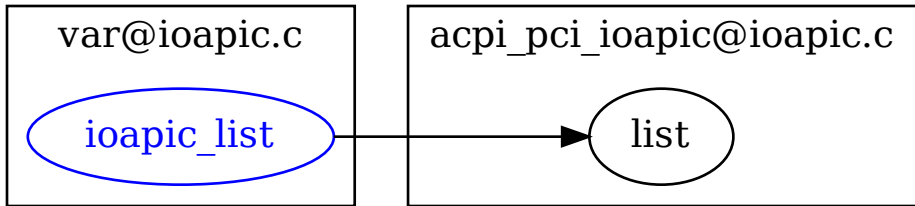
list

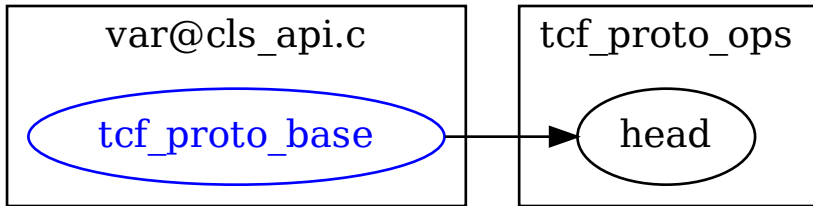
```
graph LR; subgraph var [var@dsp_pipeline.c]; de([dsp_elements]); end; subgraph dsp_element_entry [dsp_element_entry@dsp_pipeline.c]; list((list)); end; de --> list;
```

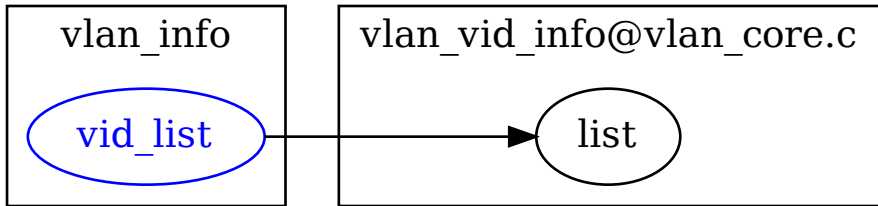
The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@dsp_pipeline.c' contains an oval labeled 'dsp_elements' in blue text. On the right, a rectangular box labeled 'dsp_element_entry@dsp_pipeline.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dsp_elements' oval to the 'list' oval, indicating that 'dsp_elements' points to 'list'.











garmin_data@garmin_gps.c

pktlist

garmin_packet@garmin_gps.c

list

```
graph LR; subgraph Module1 [garmin_data@garmin_gps.c]; pktlist; end; subgraph Module2 [garmin_packet@garmin_gps.c]; list; end; pktlist --> list;
```

The diagram illustrates a pointer relationship between two modules. On the left, a box labeled 'garmin_data@garmin_gps.c' contains a blue oval labeled 'pktlist'. On the right, a box labeled 'garmin_packet@garmin_gps.c' contains a black oval labeled 'list'. A horizontal arrow points from the 'pktlist' oval to the 'list' oval, indicating that the 'pktlist' variable in the first module points to the 'list' variable in the second module.

skeleton@v4l2-pci-skeleton.c

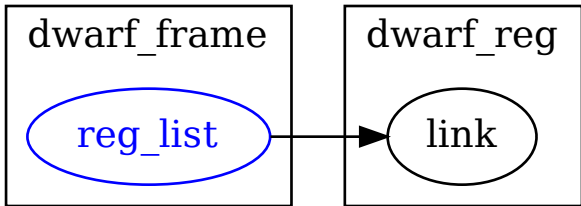
buf_list

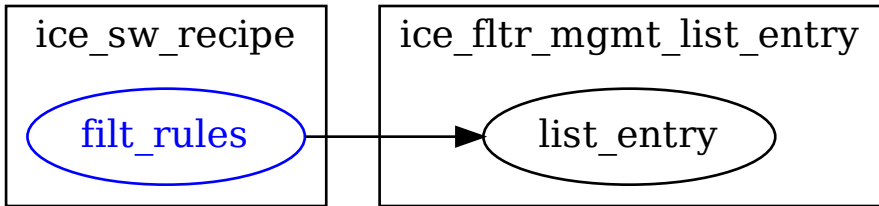
skel_buffer@v4l2-pci-skeleton.c

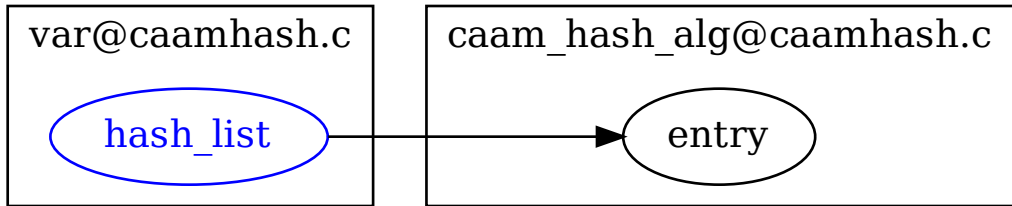
list

```
graph LR; subgraph "skeleton@v4l2-pci-skeleton.c"; buf_list; end; subgraph "skel_buffer@v4l2-pci-skeleton.c"; list; end; buf_list --> list;
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'skeleton@v4l2-pci-skeleton.c' contains a variable 'buf_list' enclosed in a blue oval. On the right, a box labeled 'skel_buffer@v4l2-pci-skeleton.c' contains a variable 'list' enclosed in a black oval. A horizontal arrow points from the 'buf_list' oval to the 'list' oval, indicating that 'buf_list' points to 'list'.







vmd_irq_list@vmd.c

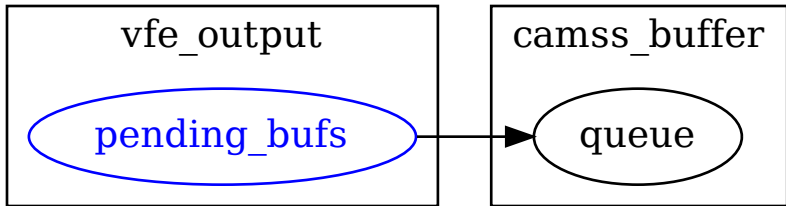
irq_list

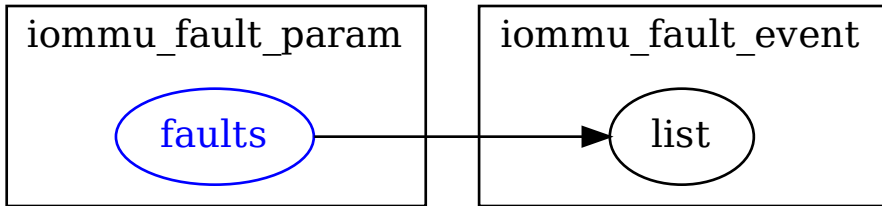
vmd_irq@vmd.c

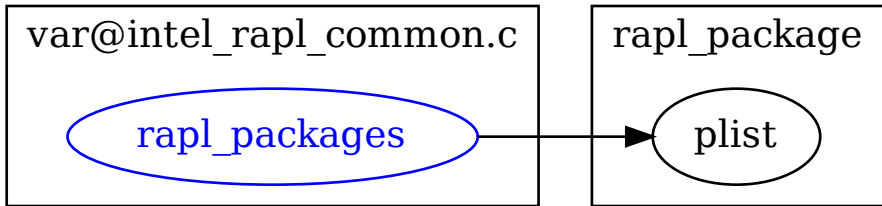
node

```
graph LR; A["vmd_irq_list@vmd.c"] --> B["vmd_irq@vmd.c"]; subgraph A; A1(["irq_list"]); end; subgraph B; B1(["node"]); end; A1 --> B1;
```

The diagram consists of two rectangular boxes. The left box is labeled 'vmd_irq_list@vmd.c' and contains a blue oval labeled 'irq_list'. The right box is labeled 'vmd_irq@vmd.c' and contains a black oval labeled 'node'. A horizontal arrow points from the right side of the 'irq_list' oval to the left side of the 'node' oval.







var@rename_volumes

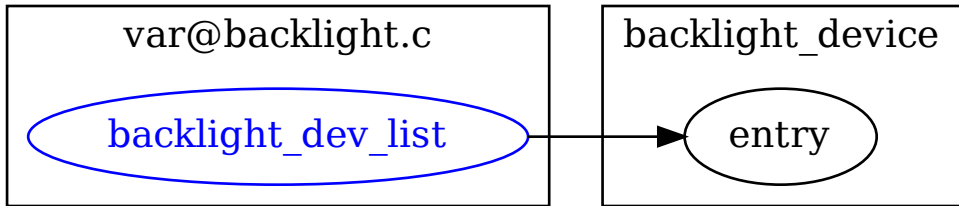
rename_list

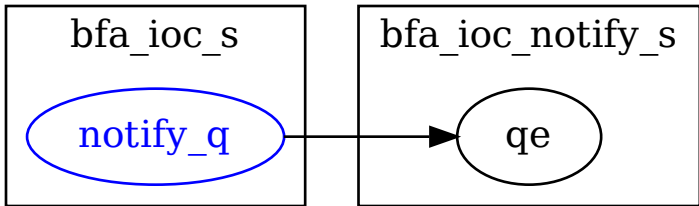
ubi_rename_entry

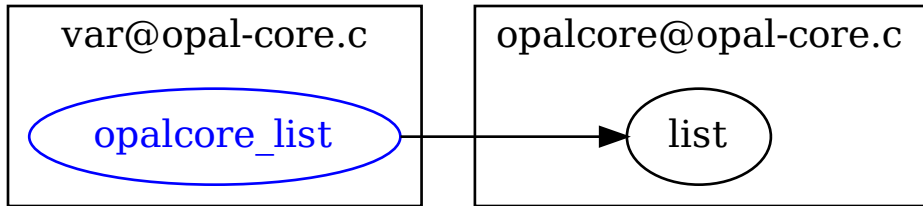
list

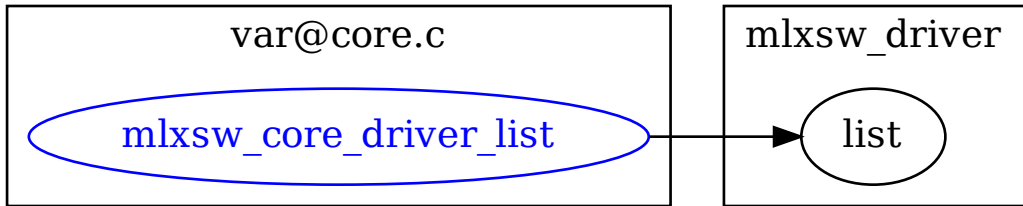
```
graph LR; A[rename_list] --> B(list);
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'var@rename_volumes' contains an oval labeled 'rename_list' with a blue border. On the right, a rectangular box labeled 'ubi_rename_entry' contains an oval labeled 'list' with a black border. A horizontal arrow points from the right side of the 'rename_list' oval to the left side of the 'list' oval, indicating that 'rename_list' points to 'list'.









var@mconsole_kern.c

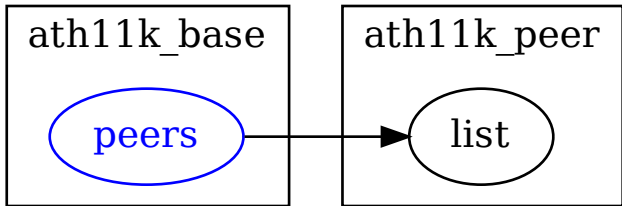
unplugged_pages

unplugged_pages@mconsole_kern.c

list

```
graph LR; subgraph Box1 [var@mconsole_kern.c]; U1(unplugged_pages); end; subgraph Box2 [unplugged_pages@mconsole_kern.c]; L((list)); end; U1 --> L;
```

The diagram illustrates a pointer relationship between two variables in kernel code. It consists of two rectangular boxes. The left box, labeled 'var@mconsole_kern.c', contains a blue oval labeled 'unplugged_pages'. The right box, labeled 'unplugged_pages@mconsole_kern.c', contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'unplugged_pages' oval in the first box to the left side of the 'list' oval in the second box, indicating that the 'unplugged_pages' variable points to the 'list' structure.

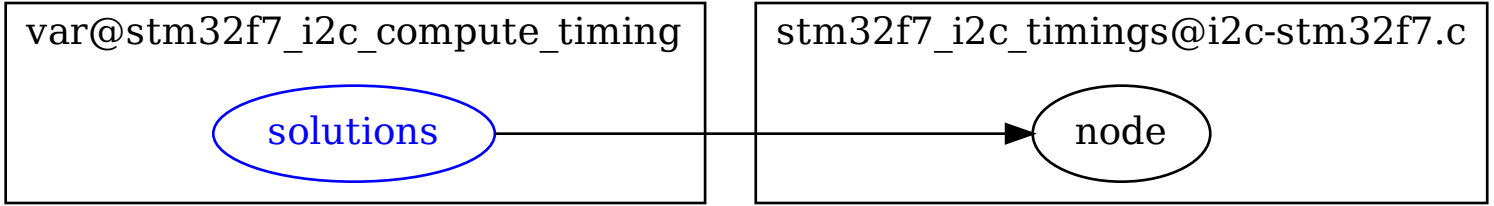


var@stm32f7_i2c_compute_timing

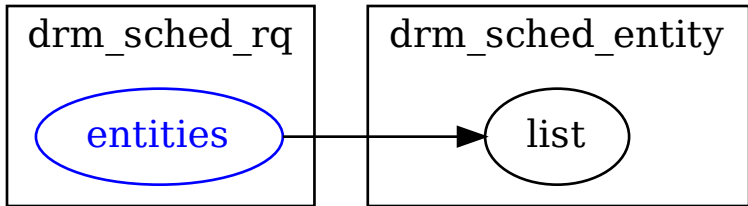
solutions

stm32f7_i2c_timings@i2c-stm32f7.c

node



```
graph LR; subgraph Box1 [var@stm32f7_i2c_compute_timing]; solutions((solutions)); end; subgraph Box2 [stm32f7_i2c_timings@i2c-stm32f7.c]; node((node)); end; solutions --> node;
```



iosapic_intr_info@iosapic.c

rtes

iosapic_rte_info@iosapic.c

rte_list

```
graph LR; subgraph Box1 [iosapic_intr_info@iosapic.c]; rtes((rtes)); end; subgraph Box2 [iosapic_rte_info@iosapic.c]; rte_list((rte_list)); end; Box1 --> Box2;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'iosapic_intr_info@iosapic.c' contains an oval labeled 'rtes' with a blue border. On the right, a rectangular box labeled 'iosapic_rte_info@iosapic.c' contains an oval labeled 'rte_list' with a black border. A horizontal arrow points from the 'rtes' oval to the 'rte_list' oval, indicating that the 'rtes' variable points to the 'rte_list' structure.

usbtmc_device_data@usbtmc.c

file_list

usbtmc_file_data@usbtmc.c

file_elem



var@pci_stub.c

pcistub_device_ids

pcistub_device_id@pci_stub.c

slot_list

```
graph LR; A[pcistub_device_ids] --> B(slot_list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@pci_stub.c' contains an oval labeled 'pcistub_device_ids' with a blue border. On the right, a rectangular box labeled 'pcistub_device_id@pci_stub.c' contains an oval labeled 'slot_list' with a black border. A horizontal arrow points from the right side of the 'pcistub_device_ids' oval to the left side of the 'slot_list' oval, indicating that 'pcistub_device_ids' points to 'slot_list'.

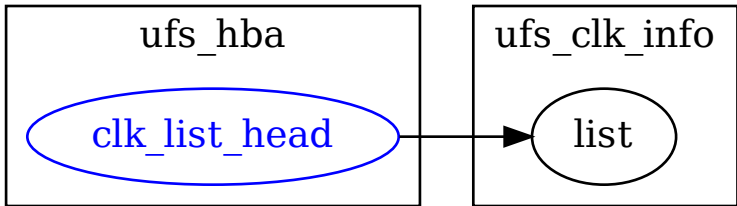
var@ibmphp_ebda.c

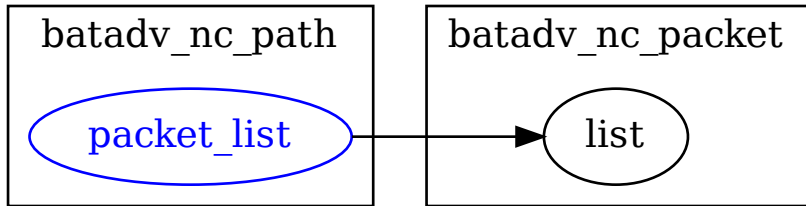
opt_lo_head

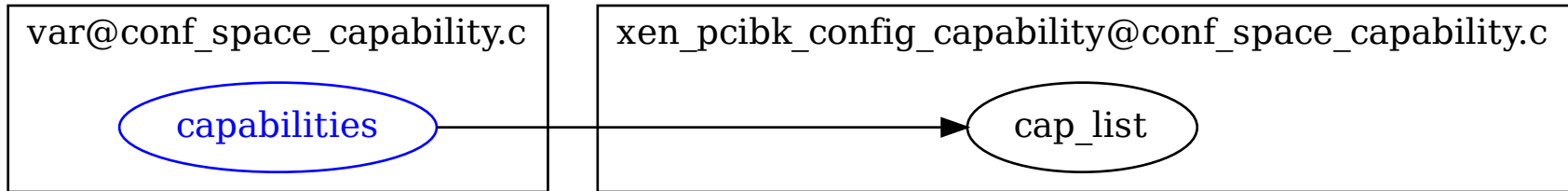
opt_rio_lo

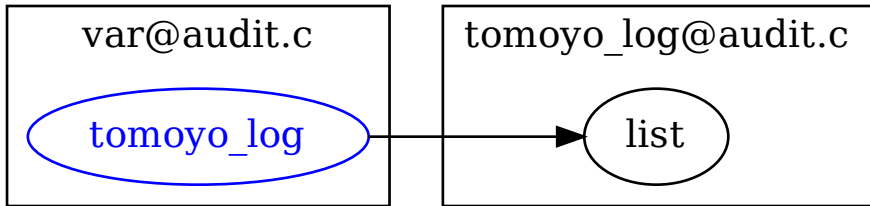
opt_rio_lo_list

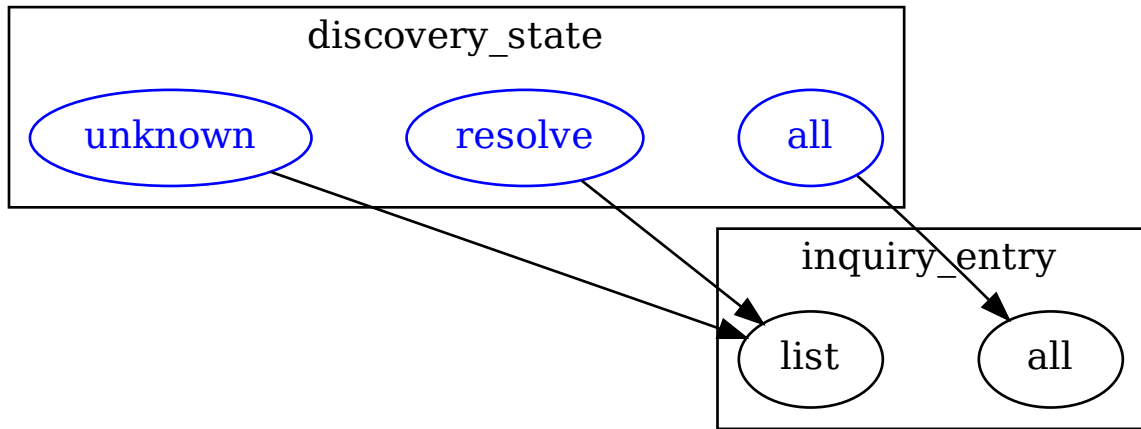


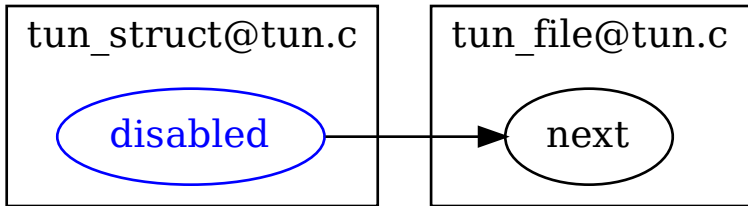












var@scsi_scan.c

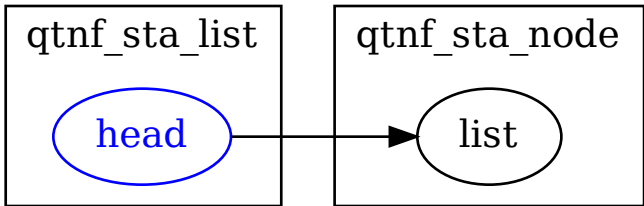
scanning_hosts

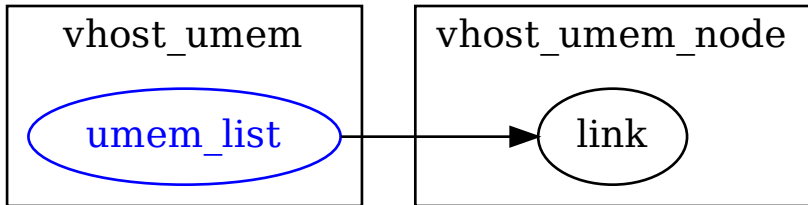
async_scan_data@scsi_scan.c

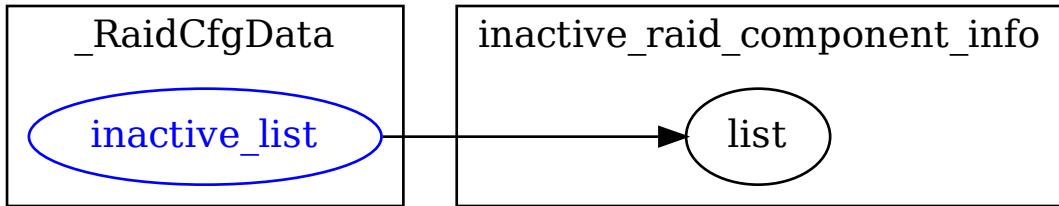
list

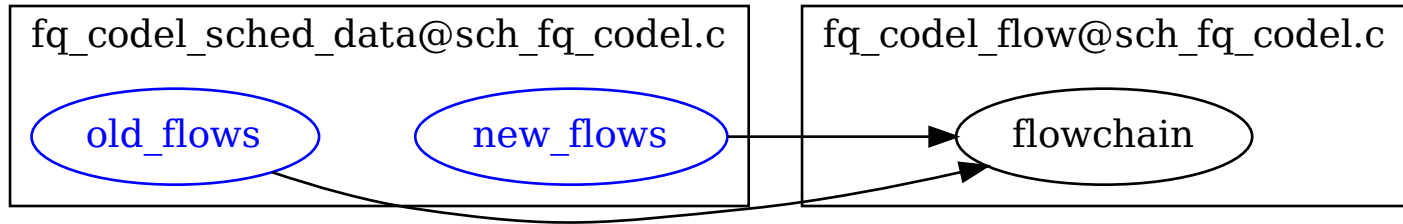
```
graph LR; subgraph Box1 [var@scsi_scan.c]; S([scanning_hosts]); end; subgraph Box2 [async_scan_data@scsi_scan.c]; L([list]); end; S --> L;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@scsi_scan.c' and contains a blue oval labeled 'scanning_hosts'. The right box is labeled 'async_scan_data@scsi_scan.c' and contains a black oval labeled 'list'. A horizontal arrow points from the right side of the 'scanning_hosts' oval to the left side of the 'list' oval.









dm_pool_metadata@dm-thin-metadata.c

thin_devices

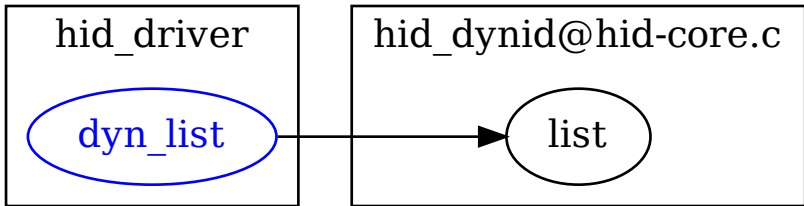
dm_thin_device@dm-thin-metadata.c

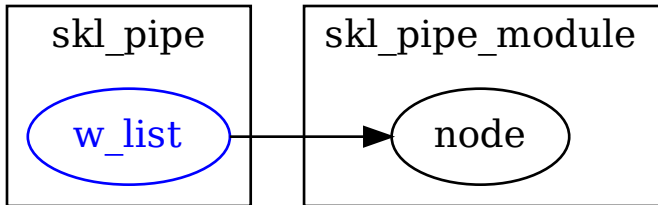
list

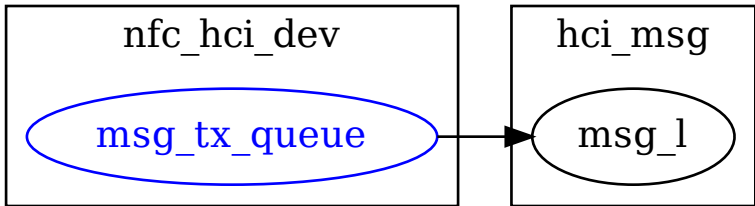


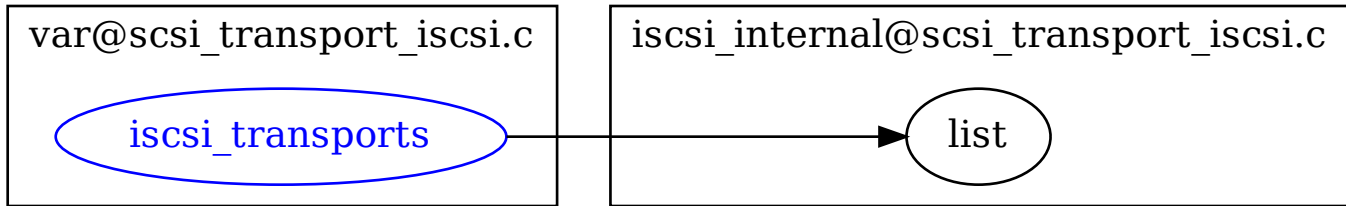
```
graph LR; A[dm_pool_metadata@dm-thin-metadata.c] --> B[dm_thin_device@dm-thin-metadata.c]; B --> C((list));
```

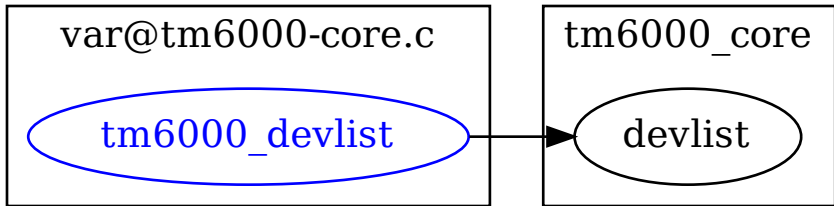
The diagram consists of two rectangular boxes. The left box contains the text 'dm_pool_metadata@dm-thin-metadata.c' at the top and 'thin_devices' in a blue oval below it. The right box contains the text 'dm_thin_device@dm-thin-metadata.c' at the top and 'list' in a black oval below it. A horizontal arrow points from the 'thin_devices' oval to the 'list' oval.

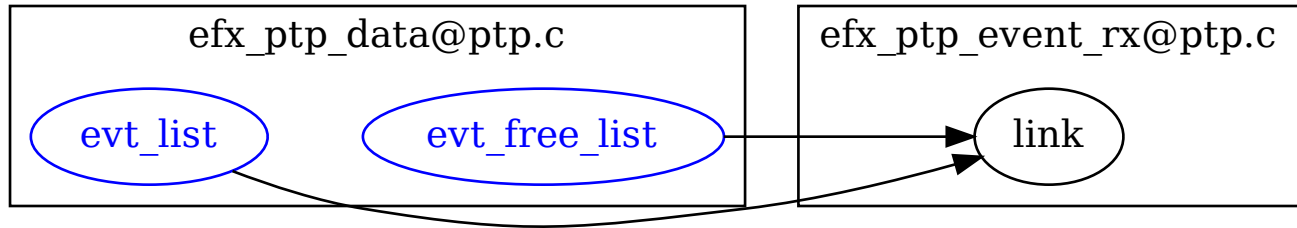


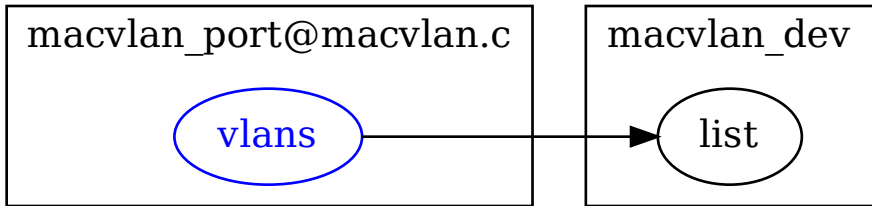


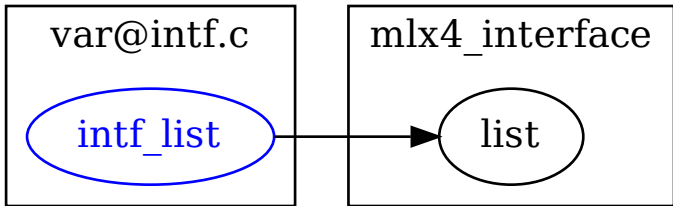


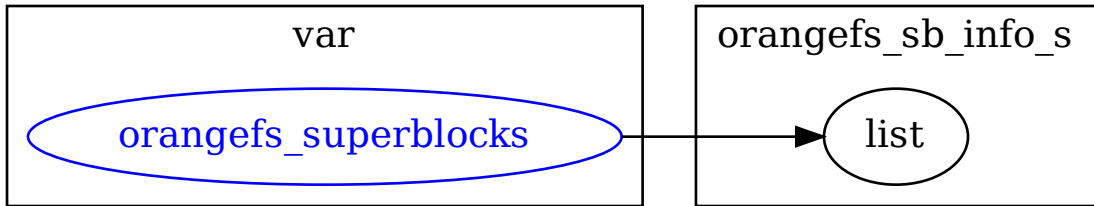


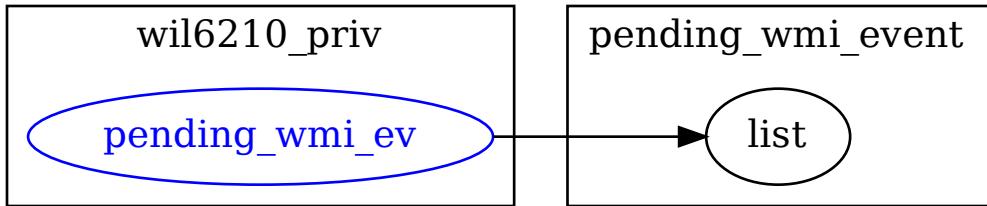


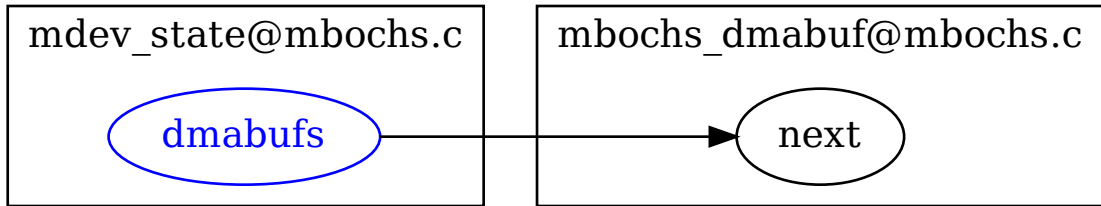


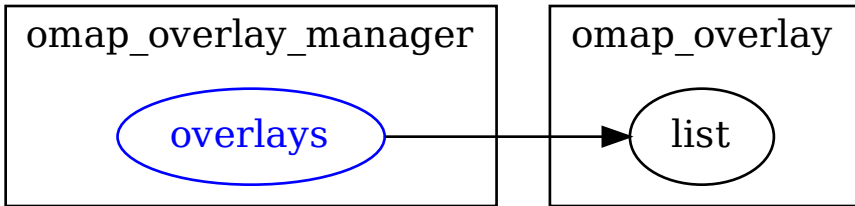


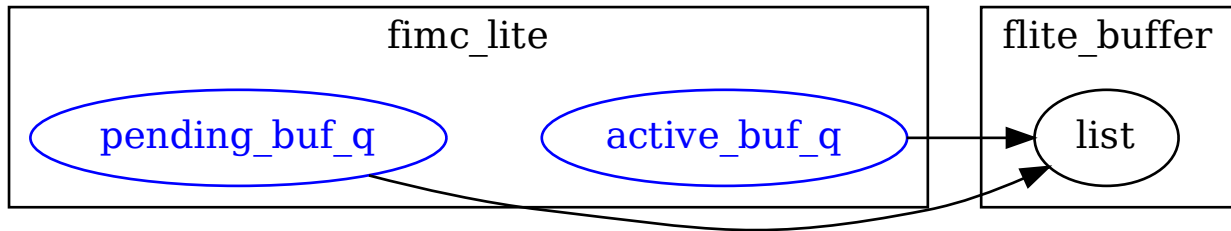


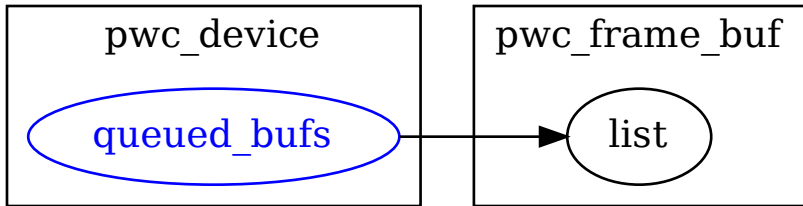


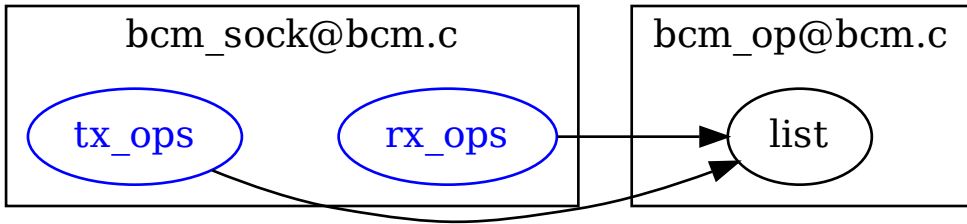


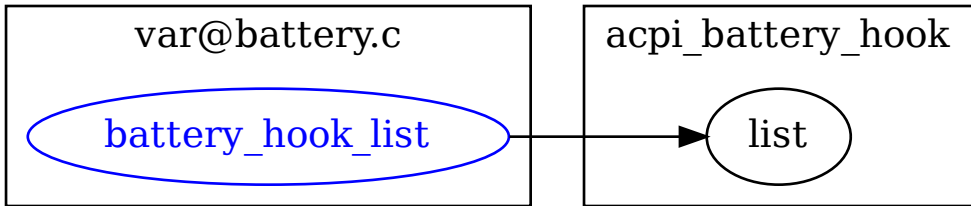


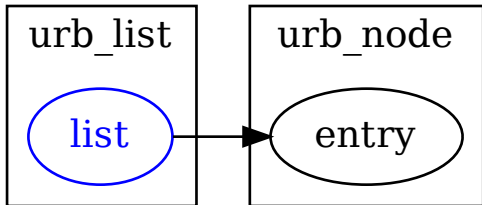


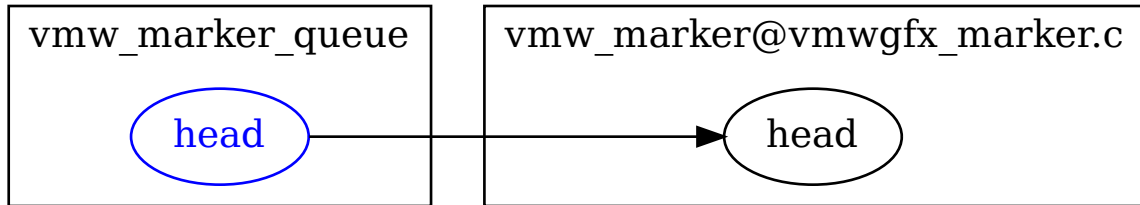


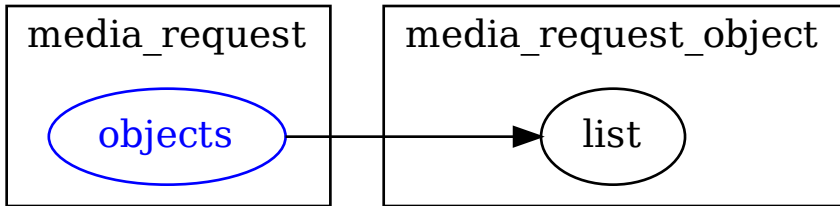


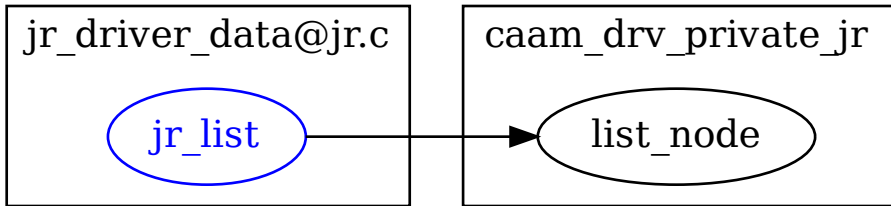












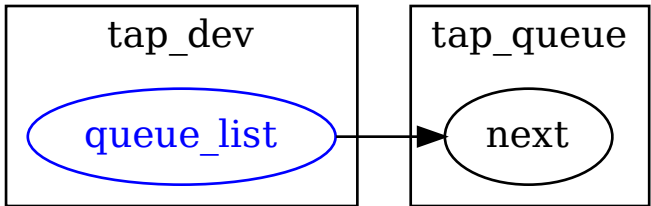
var@functions.c

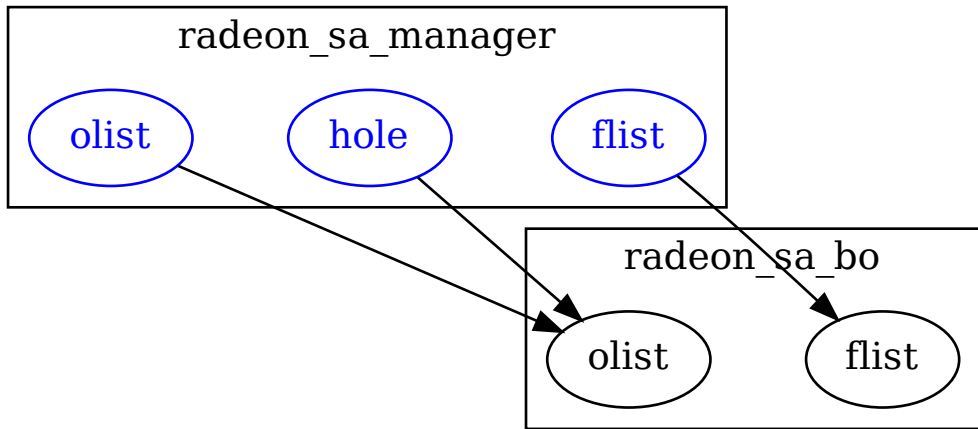
func_list

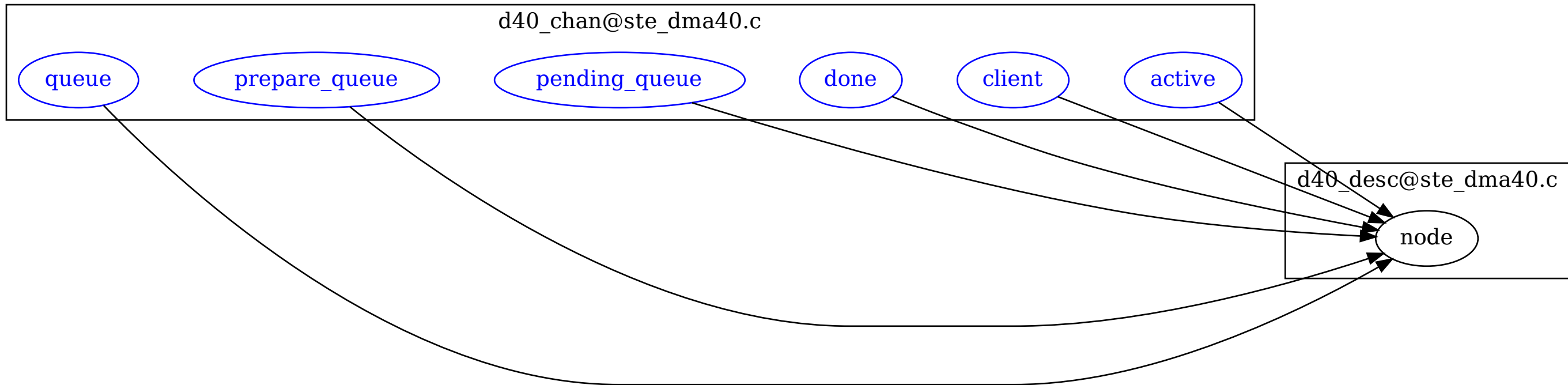
usb_function_driver

list

The diagram consists of two rectangular boxes. The left box is labeled 'var@functions.c' and contains an oval labeled 'func_list' in blue text. The right box is labeled 'usb_function_driver' and contains an oval labeled 'list'. A horizontal arrow points from the right side of the 'func_list' oval to the left side of the 'list' oval, indicating a pointer relationship.







brcmstb_gpio_priv@gpio-brcmstb.c

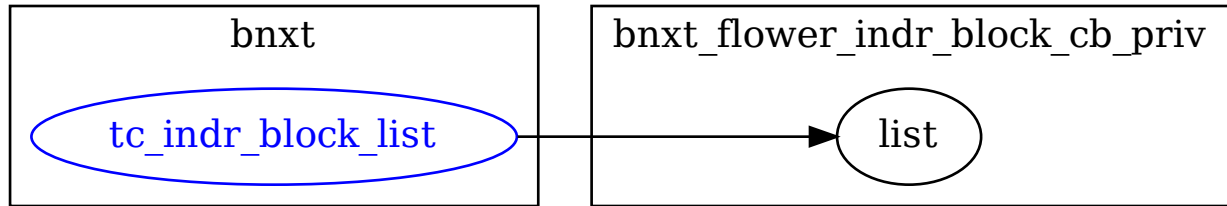
bank_list

brcmstb_gpio_bank@gpio-brcmstb.c

node

```
graph LR; A[brcmstb_gpio_priv@gpio-brcmstb.c] --> B[brcmstb_gpio_bank@gpio-brcmstb.c]; subgraph A [brcmstb_gpio_priv@gpio-brcmstb.c]; C((bank_list)); end; subgraph B [brcmstb_gpio_bank@gpio-brcmstb.c]; D((node)); end; C --> D;
```

The diagram illustrates a pointer relationship between two structures. On the left, a box represents the `brcmstb_gpio_priv@gpio-brcmstb.c` structure, which contains a variable named `bank_list` (highlighted with a blue oval). On the right, a box represents the `brcmstb_gpio_bank@gpio-brcmstb.c` structure, which contains a variable named `node` (highlighted with a black oval). A horizontal arrow points from the `bank_list` variable in the first structure to the `node` variable in the second structure, indicating that `bank_list` points to `node`.



drr_sched@sch_drr.c

active

drr_class@sch_drr.c

alist



nft_bitmap@nft_set_bitmap.c

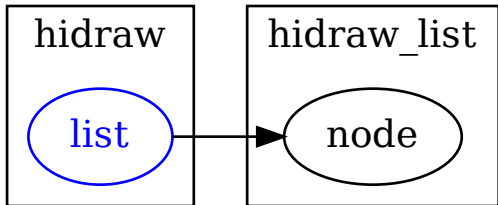
list

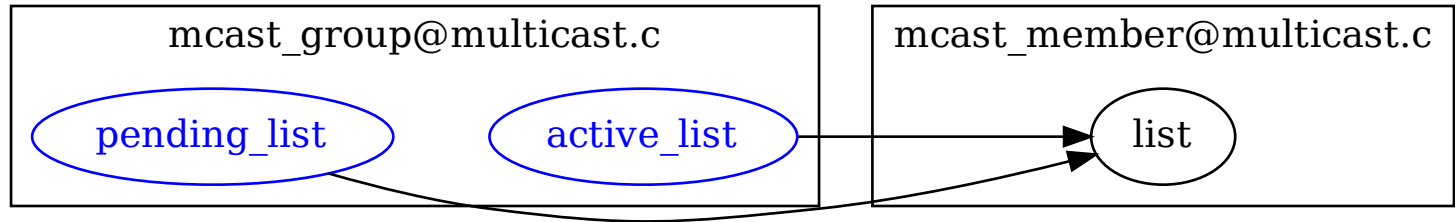
nft_bitmap_elem@nft_set_bitmap.c

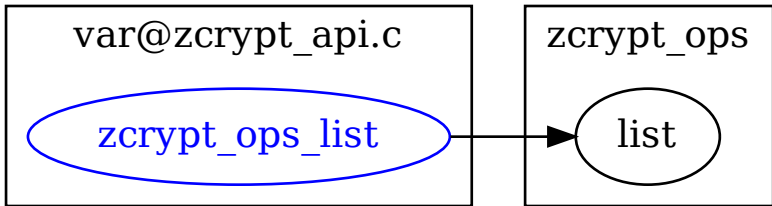
head

```
graph LR; list((list)) --> head((head));
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'nft_bitmap@nft_set_bitmap.c' contains a blue oval labeled 'list'. On the right, a box labeled 'nft_bitmap_elem@nft_set_bitmap.c' contains a black oval labeled 'head'. A horizontal arrow points from the 'list' oval to the 'head' oval, indicating that the 'list' variable points to the 'head' variable.







var@cec-notifier.c

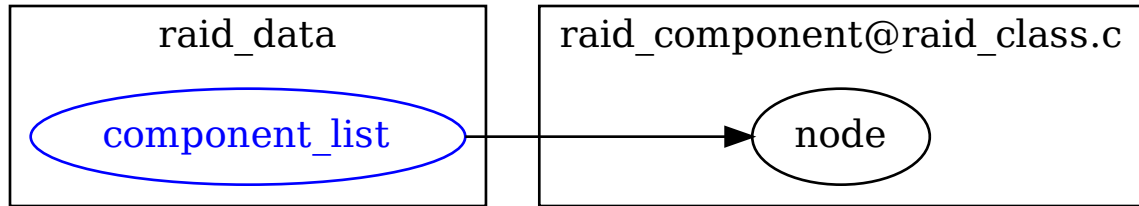
cec_notifiers

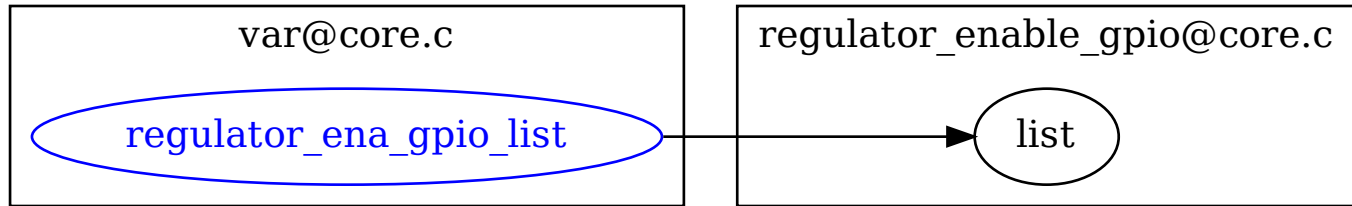
cec_notifier@cec-notifier.c

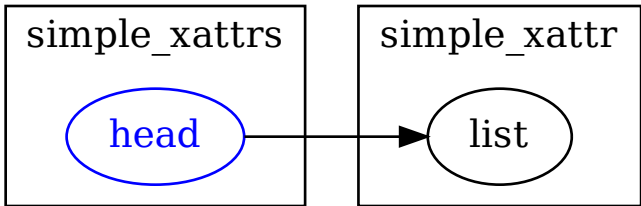
head

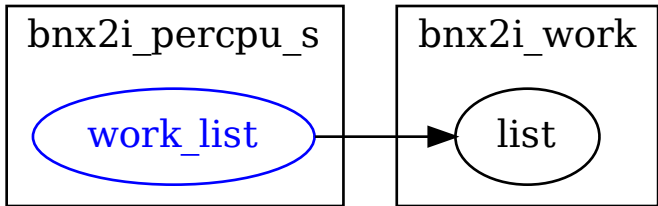
```
graph LR; A[cec_notifiers] --> B[head]
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@cec-notifier.c' and contains a blue oval labeled 'cec_notifiers'. The right box is labeled 'cec_notifier@cec-notifier.c' and contains a black oval labeled 'head'. A horizontal arrow points from the right side of the 'cec_notifiers' oval to the left side of the 'head' oval.









mlxsw_sp_fib6_entry@spectrum_router.c

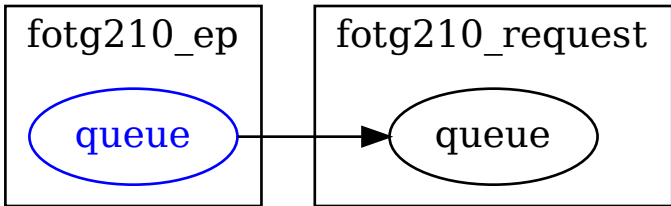
rt6_list

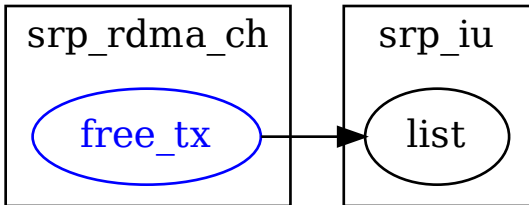
mlxsw_sp_rt6@spectrum_router.c

list

```
graph LR; A[mlxsw_sp_fib6_entry@spectrum_router.c] --> B((list)); B --- C[mlxsw_sp_rt6@spectrum_router.c];
```

The diagram illustrates a pointer relationship between two code locations. On the left, a box labeled 'mlxsw_sp_fib6_entry@spectrum_router.c' contains an oval labeled 'rt6_list'. On the right, a box labeled 'mlxsw_sp_rt6@spectrum_router.c' contains an oval labeled 'list'. A horizontal arrow points from the 'rt6_list' oval to the 'list' oval, indicating that 'rt6_list' points to 'list'. Additionally, a horizontal line connects the two boxes, passing through the arrow.





var@ccp-crypto-main.c

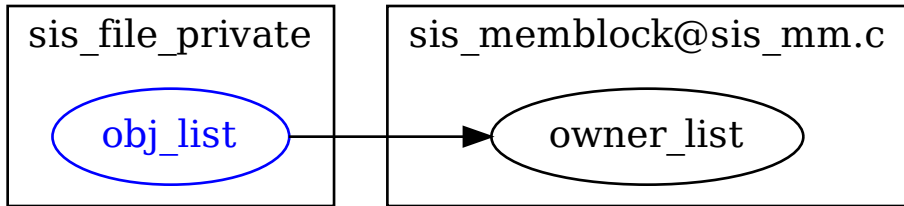
aead_algs

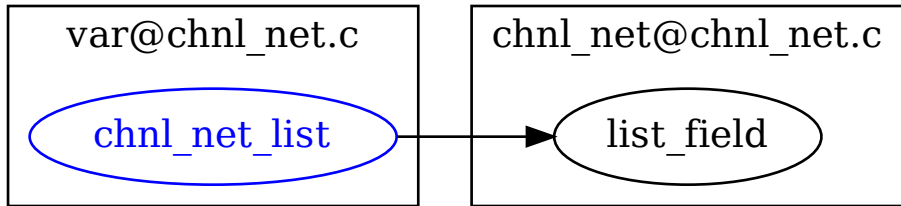
ccp_crypto_aead

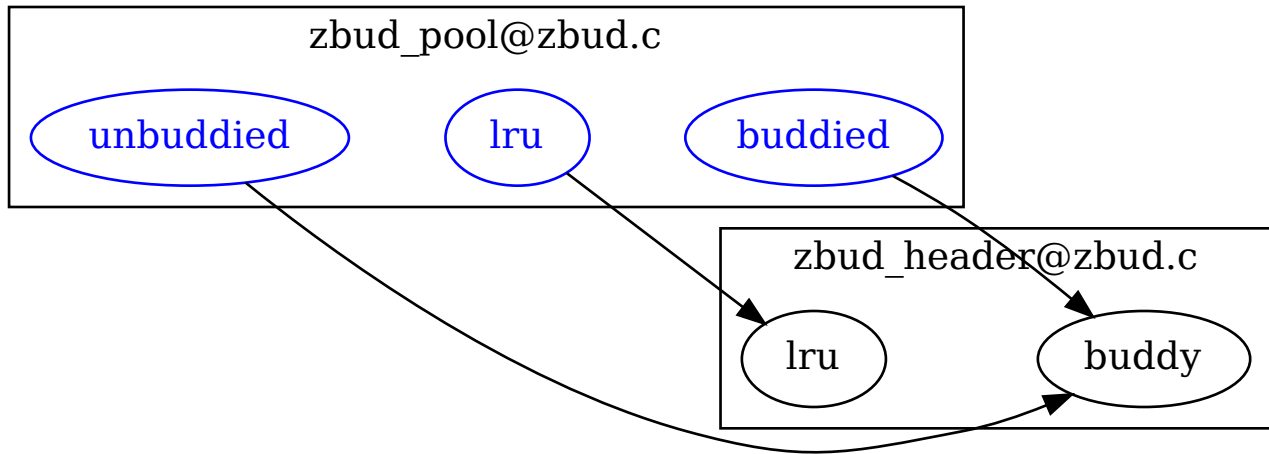
entry

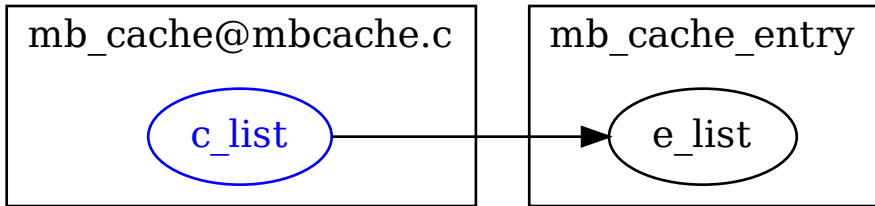
```
graph LR; subgraph var_ccp_crypto_main_c [var@ccp-crypto-main.c]; aead_algs; end; subgraph ccp_crypto_aead [ccp_crypto_aead]; entry; end; aead_algs --> entry;
```

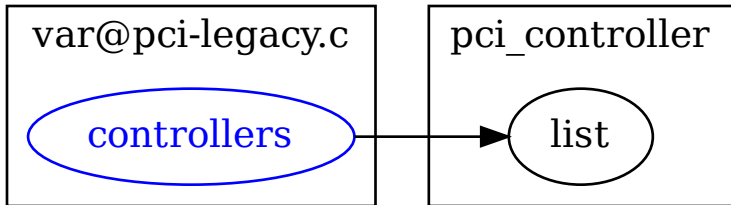
The diagram illustrates a pointer relationship between two code blocks. On the left, a box labeled 'var@ccp-crypto-main.c' contains an oval labeled 'aead_algs' with a blue border. On the right, a box labeled 'ccp_crypto_aead' contains an oval labeled 'entry'. A horizontal arrow points from the 'aead_algs' oval to the 'entry' oval, indicating that 'aead_algs' points to 'entry'.

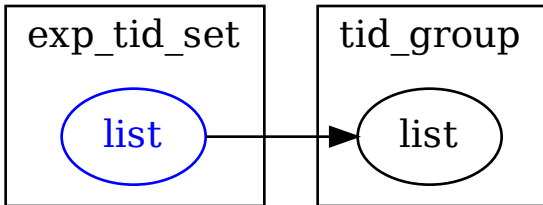


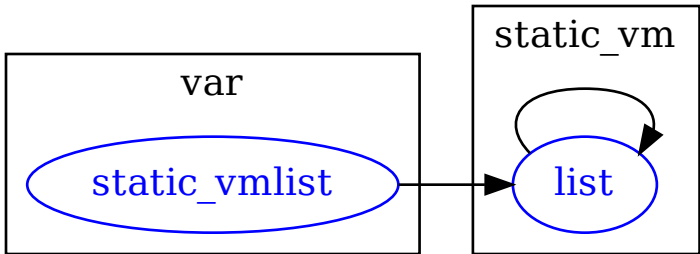












var@hid-logitech-dj.c

dj_hdev_list

dj_receiver_dev@hid-logitech-dj.c

list

```
graph LR; A[dj_hdev_list] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@hid-logitech-dj.c' contains the variable 'dj_hdev_list', which is highlighted with a blue oval. On the right, a box labeled 'dj_receiver_dev@hid-logitech-dj.c' contains an oval labeled 'list'. A horizontal arrow points from the 'dj_hdev_list' oval to the 'list' oval, indicating that 'dj_hdev_list' points to 'list'.

var@component.c

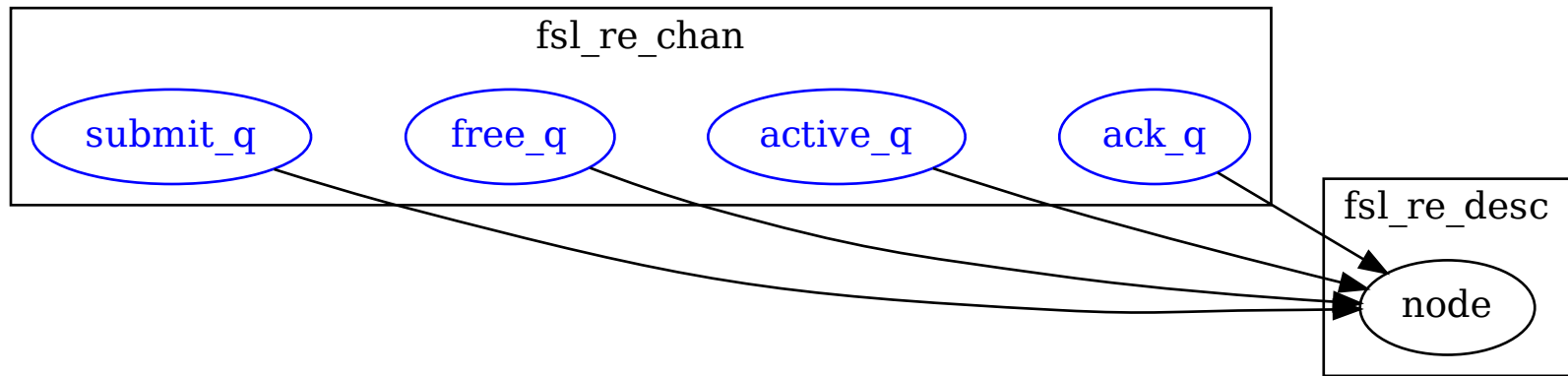
masters

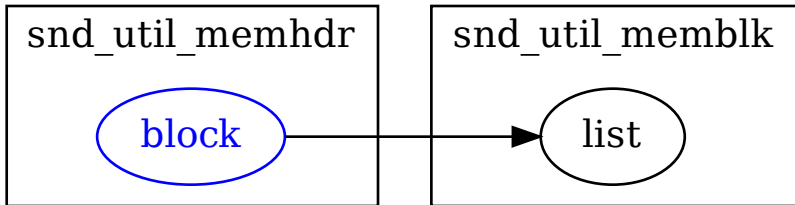
master@component.c

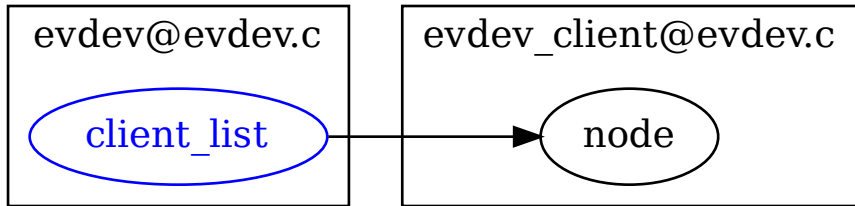
node

```
graph LR; subgraph var [var@component.c]; masters; end; subgraph master [master@component.c]; node; end; masters --> node;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@component.c' and contains an oval labeled 'masters' with a blue border. The right box is labeled 'master@component.c' and contains an oval labeled 'node'. A horizontal arrow points from the 'masters' oval to the 'node' oval, indicating a pointer or reference relationship.







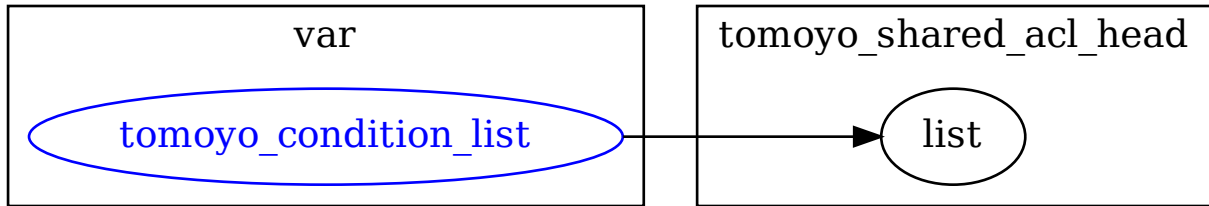
privcmd_buf_private@privcmd-buf.c

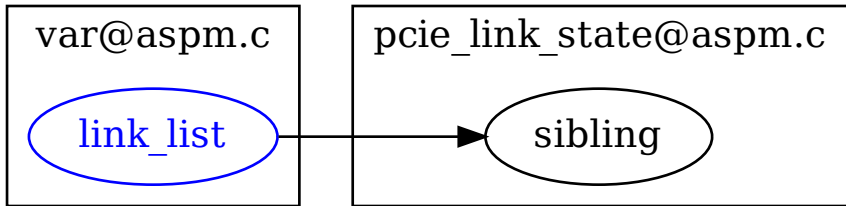
list

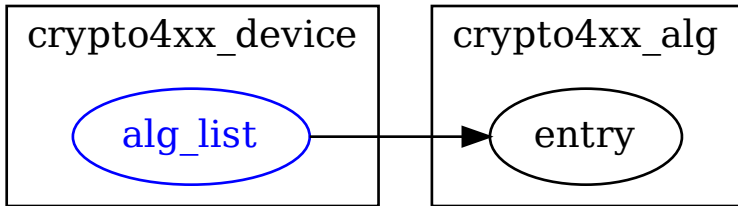
privcmd_buf_vma_private@privcmd-buf.c

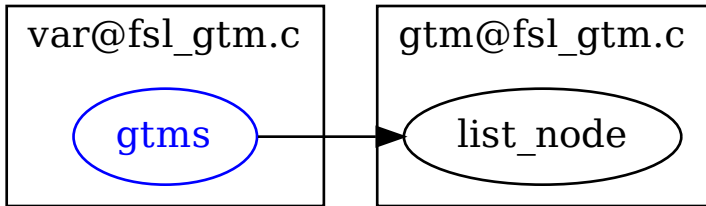
list











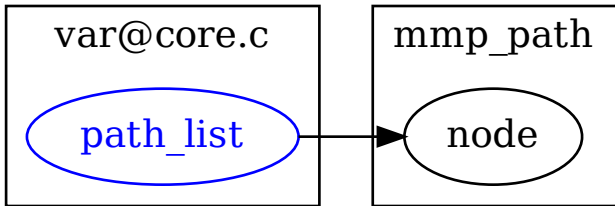
var@cxgb3_offload.c

ofld_dev_list

t3cdev

ofld_dev_list

The diagram consists of two rectangular boxes. The left box is labeled 'var@cxgb3_offload.c' and contains a blue oval labeled 'ofld_dev_list'. The right box is labeled 't3cdev' and contains a black oval labeled 'ofld_dev_list'. A horizontal arrow points from the right side of the blue oval in the left box to the left side of the black oval in the right box, indicating a pointer relationship.



var@audio_manager.c

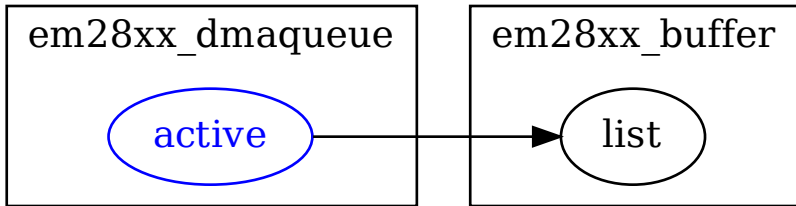
modules_list

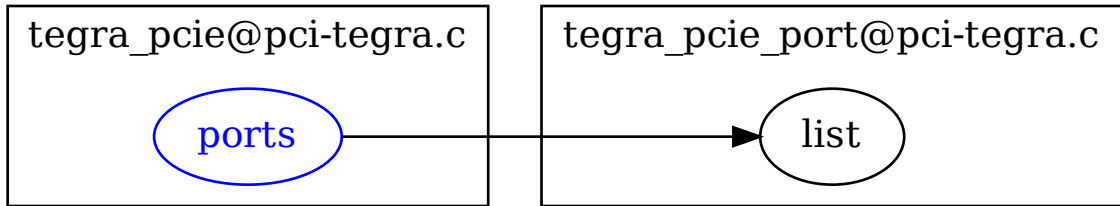
gb_audio_manager_module

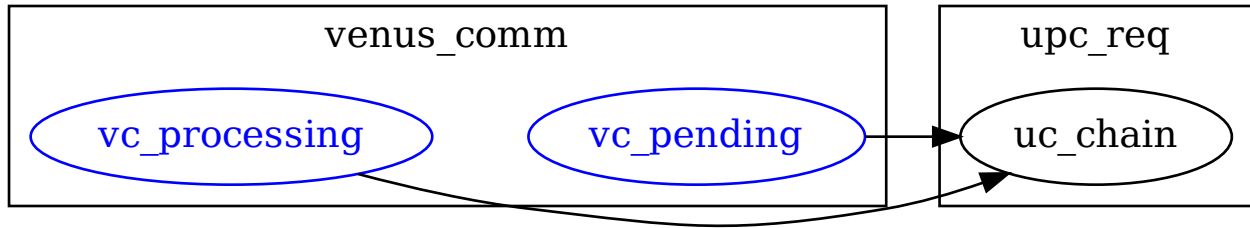
list

```
graph LR; A[modules_list] --> B(list);
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@audio_manager.c' contains an oval labeled 'modules_list' in blue text. On the right, a box labeled 'gb_audio_manager_module' contains an oval labeled 'list'. A horizontal arrow points from the 'modules_list' oval to the 'list' oval, indicating that 'modules_list' points to 'list'.







mlxsw_afa_block@core_acl_flex_actions.c

resource_list

mlxsw_afa_resource@core_acl_flex_actions.c

list

```
graph LR; A[resource_list] --> B(list);
```

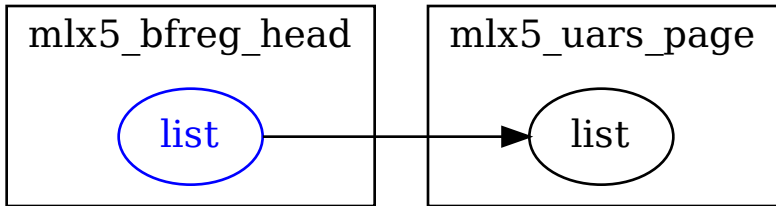
vmw_cmdbuf_res_manager@vmwgfx_cmdbuf_res.c

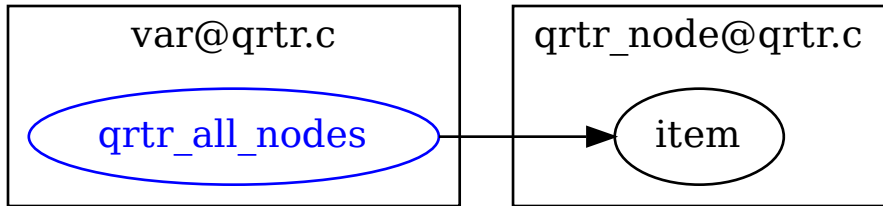
list

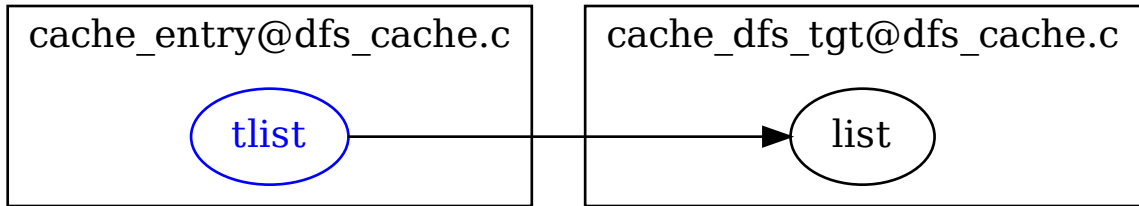
vmw_cmdbuf_res@vmwgfx_cmdbuf_res.c

head









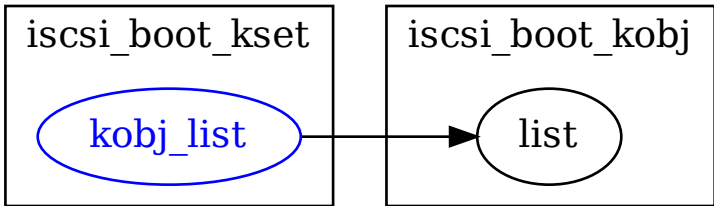
ccp_crypto_queue@ccp-crypto-main.c

cmds

ccp_crypto_cmd@ccp-crypto-main.c

entry





mv_xor_v2_device@mv_xor_v2.c

free_sw_desc

mv_xor_v2_sw_desc@mv_xor_v2.c

free_list




```
graph LR; subgraph Block1 [mv_xor_v2_device@mv_xor_v2.c]; free_sw_desc; end; subgraph Block2 [mv_xor_v2_sw_desc@mv_xor_v2.c]; free_list; end; free_sw_desc --> free_list;
```

qed_ptt_pool@qed_hw.c

free_list

qed_ptt@qed_hw.c

list_entry



```
graph LR; subgraph "qed_ptt_pool@qed_hw.c"; free_list; end; subgraph "qed_ptt@qed_hw.c"; list_entry; end; free_list --> list_entry;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'qed_ptt_pool@qed_hw.c' contains an oval labeled 'free_list' with a blue border. On the right, a box labeled 'qed_ptt@qed_hw.c' contains an oval labeled 'list_entry' with a black border. A horizontal arrow points from the right side of the 'free_list' oval to the left side of the 'list_entry' oval, indicating that 'free_list' points to 'list_entry'.

var@block2mtd.c

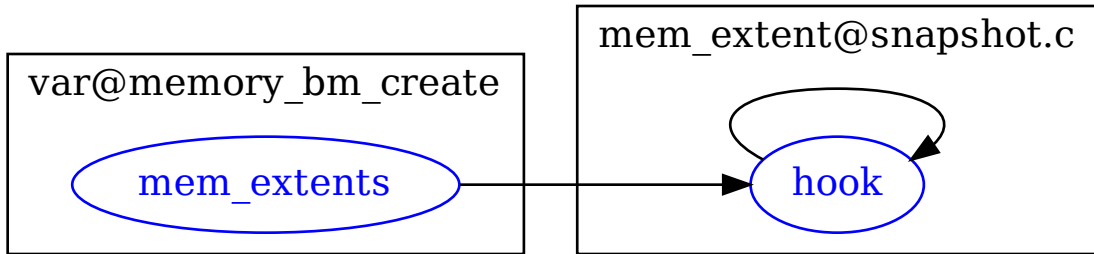
blkmtdev_device_list

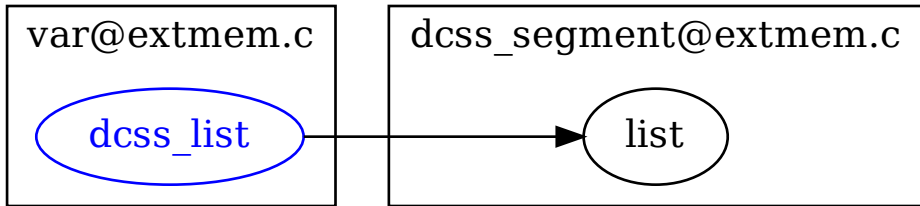
block2mtd_dev@block2mtd.c

list

```
graph LR; subgraph File1 [var@block2mtd.c]; A(blkmtdev_device_list); end; subgraph File2 [block2mtd_dev@block2mtd.c]; B((list)); end; A --> B;
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box represents the source file `var@block2mtd.c`. Inside this box, the variable `blkmtdev_device_list` is enclosed in a blue oval. On the right, another rectangular box represents the source file `block2mtd_dev@block2mtd.c`. Inside this box, the variable `list` is enclosed in a black oval. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the black oval in the second box, indicating that `blkmtdev_device_list` points to `list`.





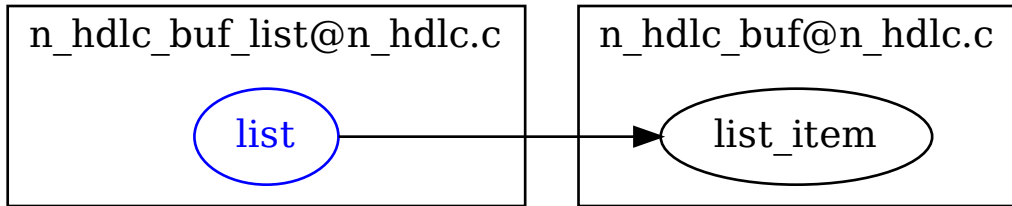
var@trace_printk.c

trace_bprintf_fmt_list

trace_bprintf_fmt@trace_printk.c

list

```
graph LR; subgraph var_trace_printk_c [var@trace_printk.c]; list1([trace_bprintf_fmt_list]); end; subgraph trace_bprintf_fmt_trace_printk_c [trace_bprintf_fmt@trace_printk.c]; list2((list)); end; list1 --> list2;
```



n2_request_context@n2_core.c

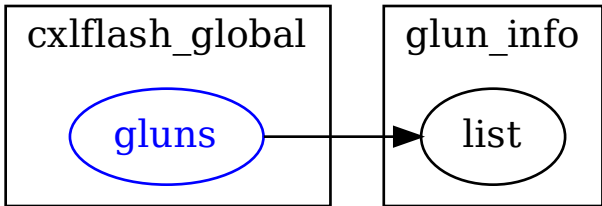
chunk_list

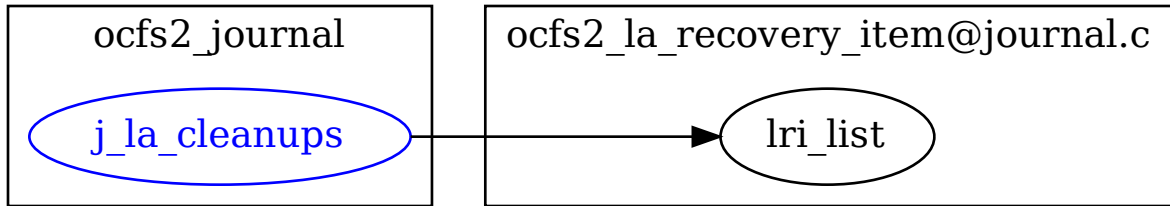
n2_crypto_chunk@n2_core.c

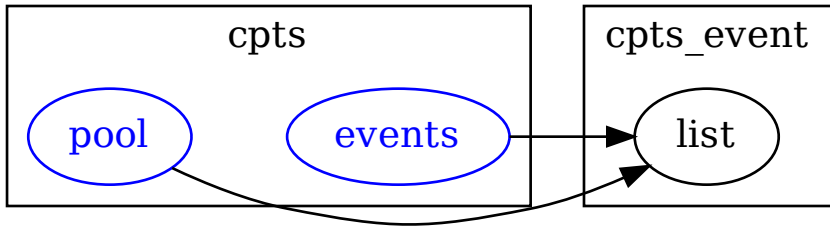
entry

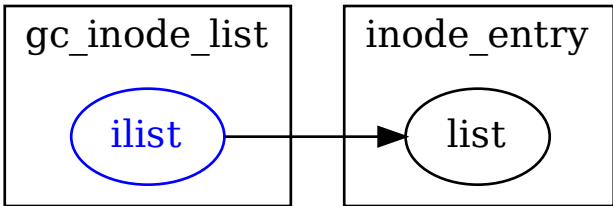
```
graph LR; A[n2_request_context@n2_core.c] --> B[n2_crypto_chunk@n2_core.c]; subgraph A; C((chunk_list)); end; subgraph B; D((entry)); end; C --> D;
```

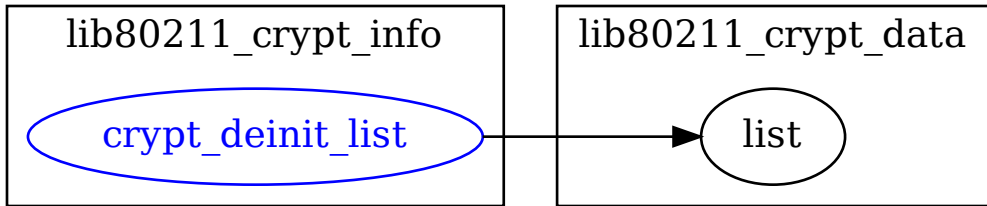
The diagram consists of two rectangular boxes. The left box is labeled 'n2_request_context@n2_core.c' and contains an oval labeled 'chunk_list' in blue text. The right box is labeled 'n2_crypto_chunk@n2_core.c' and contains an oval labeled 'entry'. A horizontal arrow points from the 'chunk_list' oval to the 'entry' oval, passing through the gap between the two boxes.

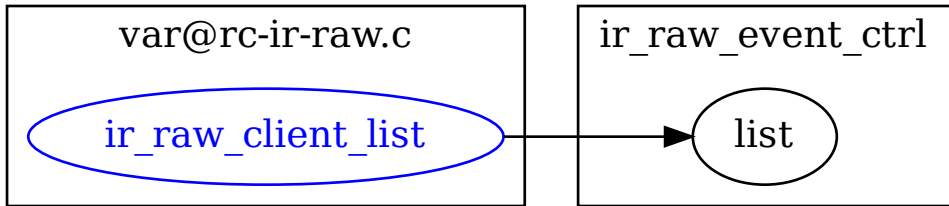


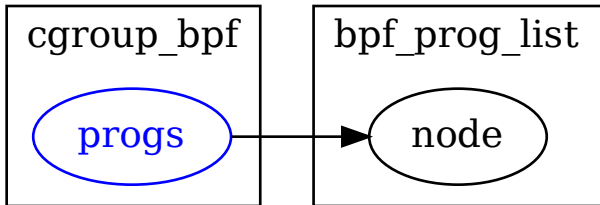


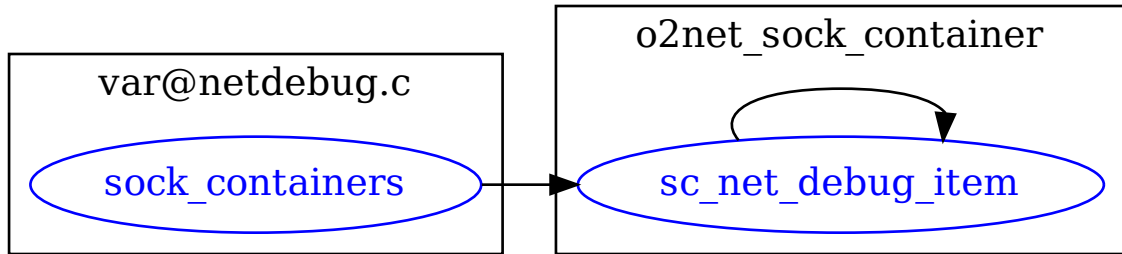


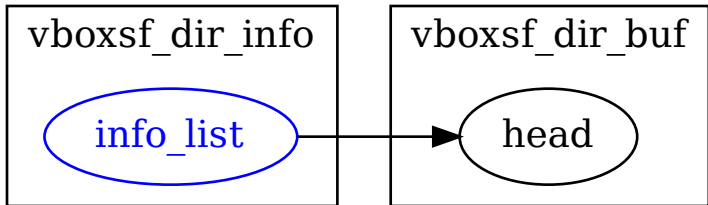


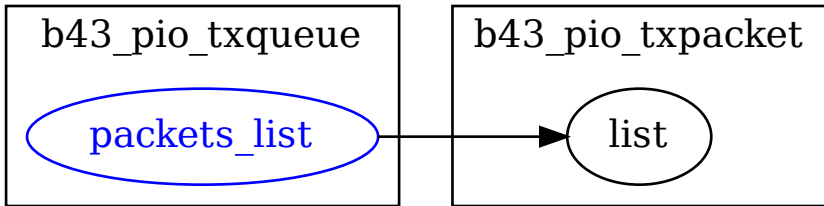












var@powercap_sys.c

powercap_cntrl_list

powercap_control_type

node

```
graph LR; A["var@powercap_sys.c  
powercap_cntrl_list"] --> B["powercap_control_type  
node"]
```

The diagram consists of two rectangular boxes. The left box contains the text 'var@powercap_sys.c' at the top and 'powercap_cntrl_list' at the bottom, which is enclosed in a blue oval. The right box contains the text 'powercap_control_type' at the top and 'node' at the bottom, which is enclosed in a black oval. A horizontal arrow points from the right side of the left box to the left side of the right box, specifically pointing towards the 'node' oval.

var@watchdog_pretimeout.c

pretimeout_list

watchdog_pretimeout@watchdog_pretimeout.c

entry

```
graph LR; A[pretimeout_list] --> B[entry];
```

The diagram illustrates a pointer relationship between two variables. On the left, a box labeled 'var@watchdog_pretimeout.c' contains a blue oval labeled 'pretimeout_list'. On the right, a box labeled 'watchdog_pretimeout@watchdog_pretimeout.c' contains a black oval labeled 'entry'. A horizontal arrow points from the 'pretimeout_list' oval to the 'entry' oval, indicating that 'pretimeout_list' points to 'entry'.

mlx4_sriov_alias_guid_port_rec_det

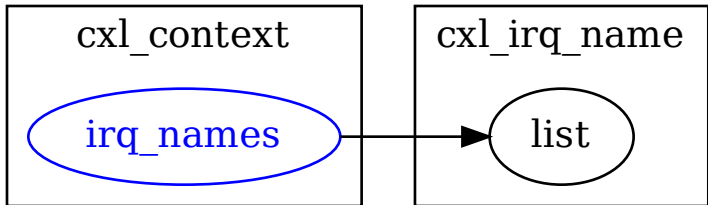
cb_list

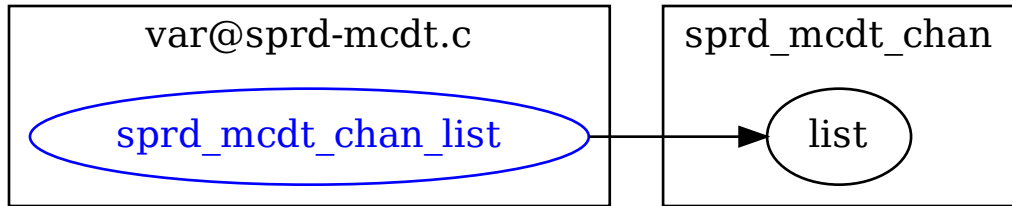
mlx4_alias_guid_work_context@alias_GUID.c

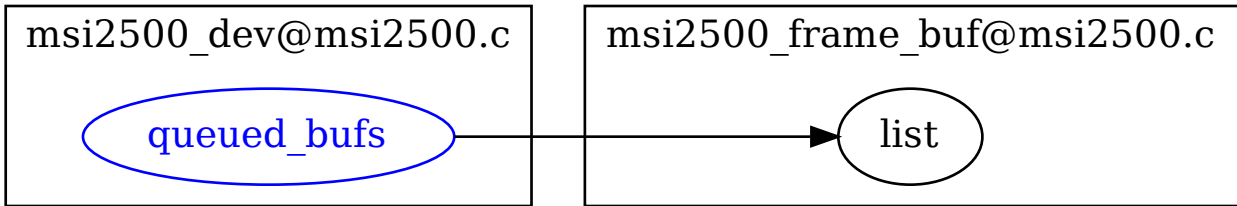
list

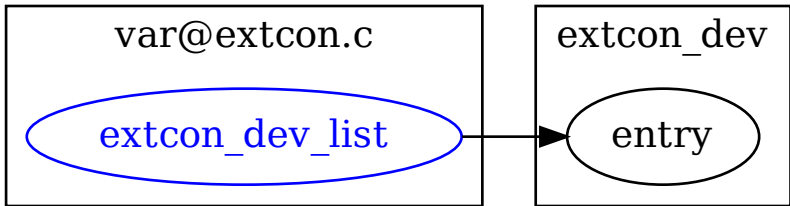
```
graph LR; A[mlx4_sriov_alias_guid_port_rec_det] --> B[mlx4_alias_guid_work_context@alias_GUID.c]; B --> C((list));
```

The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'mlx4_sriov_alias_guid_port_rec_det' contains a blue oval labeled 'cb_list'. On the right, a box labeled 'mlx4_alias_guid_work_context@alias_GUID.c' contains an oval labeled 'list'. A horizontal arrow points from the 'cb_list' oval to the 'list' oval, indicating that 'cb_list' points to 'list'.









nfs4_dir_ctx@nfs4recover.c

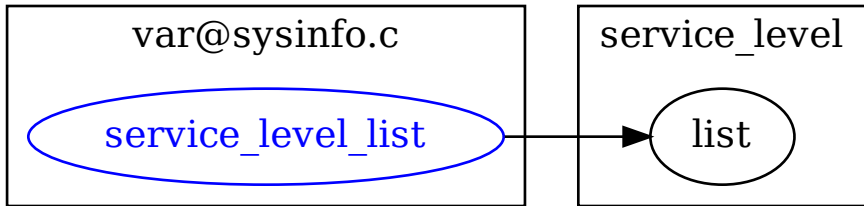
names

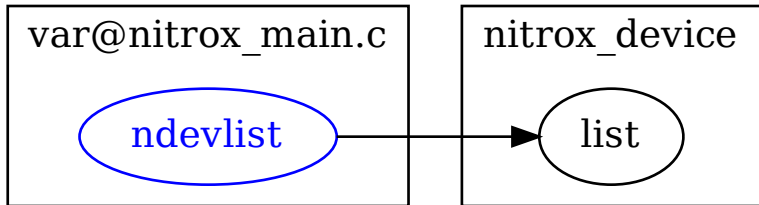
name_list@nfs4recover.c

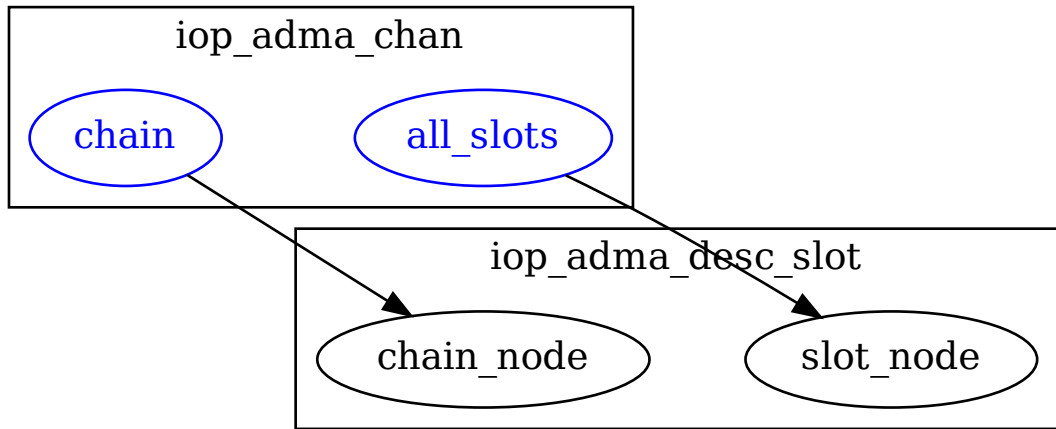
list

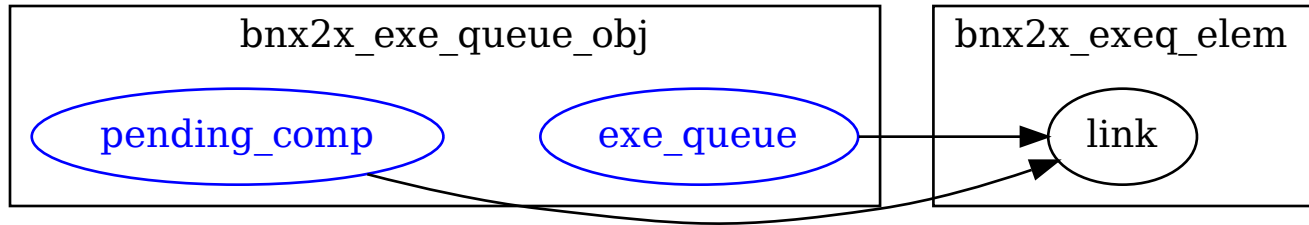
```
graph LR; subgraph Box1 [nfs4_dir_ctx@nfs4recover.c]; names; end; subgraph Box2 [name_list@nfs4recover.c]; list; end; names --> list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'nfs4_dir_ctx@nfs4recover.c' and contains an oval labeled 'names' in blue text. The right box is labeled 'name_list@nfs4recover.c' and contains an oval labeled 'list'. A horizontal arrow points from the 'names' oval in the first box to the 'list' oval in the second box.







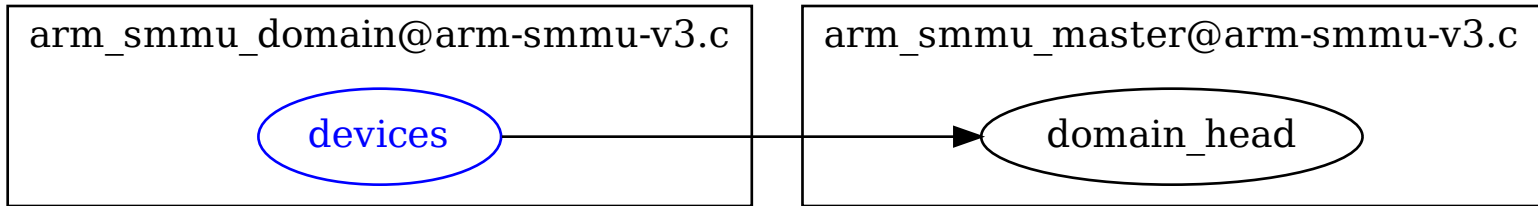


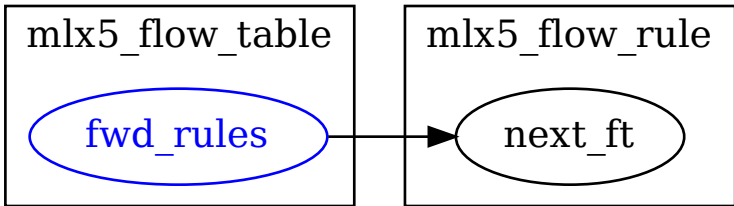
arm_smmu_domain@arm-smmu-v3.c

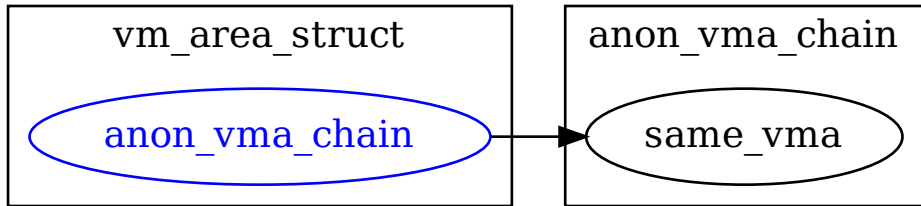
devices

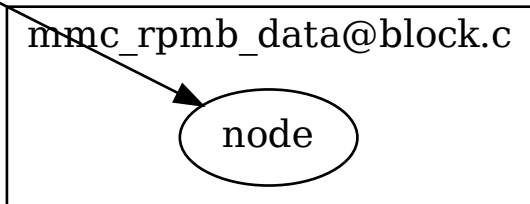
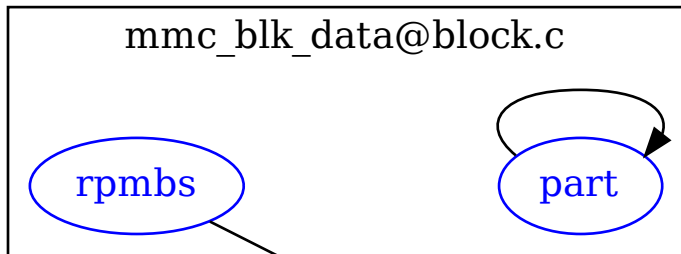
arm_smmu_master@arm-smmu-v3.c

domain_head









var@nf_conntrack_helper.c

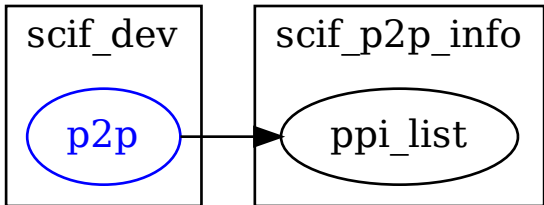
nf_ct_helper_expectfn_list

nf_ct_helper_expectfn

head

```
graph LR; A[nf_ct_helper_expectfn_list] --> B((head));
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box contains the text 'var@nf_conntrack_helper.c' and a blue oval containing 'nf_ct_helper_expectfn_list'. On the right, another rectangular box contains the text 'nf_ct_helper_expectfn' and an oval containing 'head'. A horizontal arrow points from the right side of the blue oval in the first box to the left side of the 'head' oval in the second box.



ck804xrom_window@ck804xrom.c

maps

ck804xrom_map_info@ck804xrom.c

list

```
graph LR; maps([maps]) --> list([list]);
```


The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box contains the text 'ck804xrom_window@ck804xrom.c' and a blue oval labeled 'maps'. On the right, another rectangular box contains the text 'ck804xrom_map_info@ck804xrom.c' and a black oval labeled 'list'. A horizontal arrow points from the 'maps' oval to the 'list' oval, indicating that the 'maps' variable points to the 'list' structure.

cx25821_dmaqueue

active

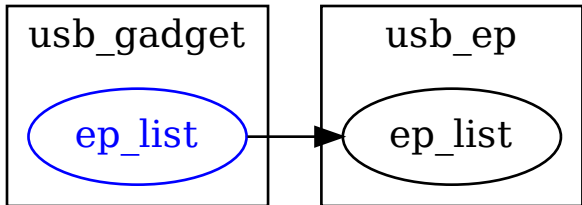
cx25821_buffer

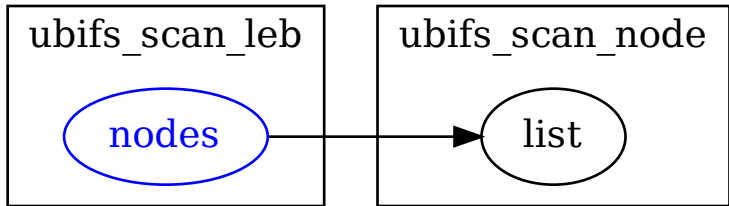
queue

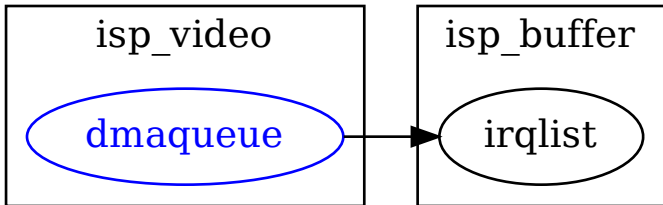


```
graph LR; subgraph cx25821_dmaqueue; active; end; subgraph cx25821_buffer; queue; end; active --> queue;
```

The diagram illustrates a pointer relationship between two data structures. On the left, a rectangular box labeled 'cx25821_dmaqueue' contains an oval labeled 'active' with a blue border. On the right, a rectangular box labeled 'cx25821_buffer' contains an oval labeled 'queue' with a black border. A horizontal arrow points from the 'active' oval to the 'queue' oval, indicating that the 'active' pointer refers to the 'queue' structure within the 'cx25821_buffer'.







vmap_block_queue@vmalloc.c

free

vmap_block@vmalloc.c

free_list

```
graph LR; subgraph Box1 [vmap_block_queue@vmalloc.c]; free((free)); end; subgraph Box2 [vmap_block@vmalloc.c]; free_list((free_list)); end; free --> free_list;
```

The diagram consists of two rectangular boxes. The left box is labeled 'vmap_block_queue@vmalloc.c' and contains a blue oval labeled 'free'. The right box is labeled 'vmap_block@vmalloc.c' and contains a black oval labeled 'free_list'. A horizontal arrow points from the 'free' oval in the left box to the 'free_list' oval in the right box.

dw_i3c_master@dw-i3c-master.c

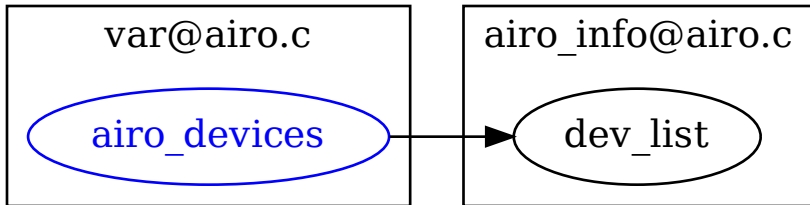
xferqueue.list

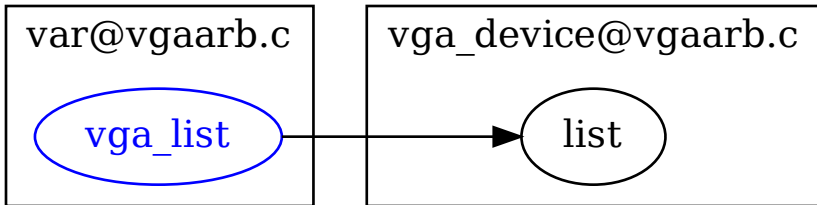
dw_i3c_xfer@dw-i3c-master.c

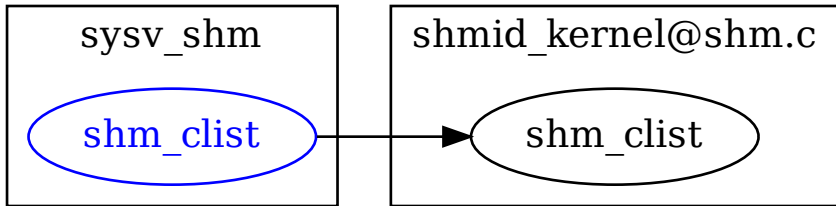
node

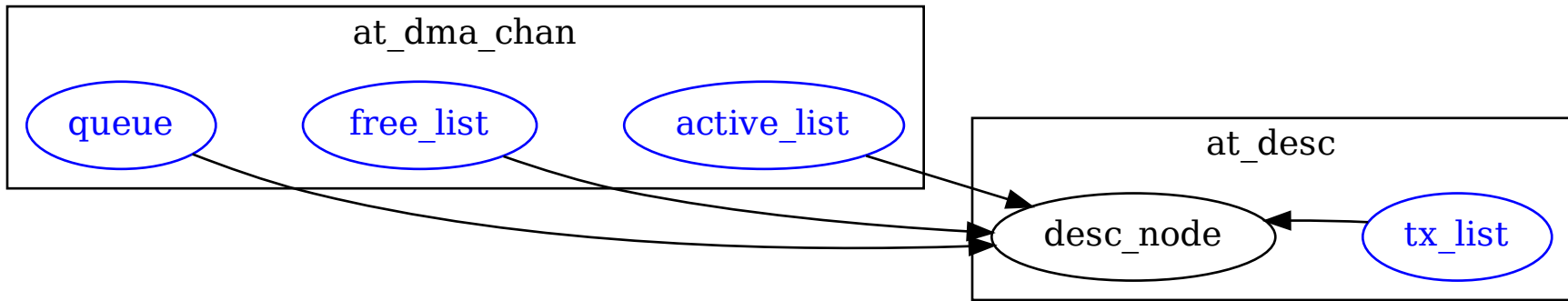
```
graph LR; subgraph Box1 [dw_i3c_master@dw-i3c-master.c]; xferqueue.list; end; subgraph Box2 [dw_i3c_xfer@dw-i3c-master.c]; node; end; xferqueue.list --> node;
```

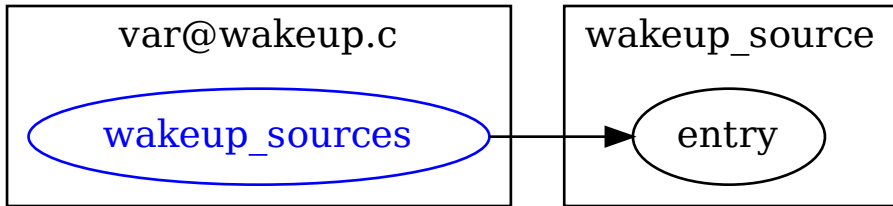
The diagram illustrates a pointer relationship between two data structures. On the left, a box labeled 'dw_i3c_master@dw-i3c-master.c' contains an oval labeled 'xferqueue.list' in blue text. On the right, a box labeled 'dw_i3c_xfer@dw-i3c-master.c' contains an oval labeled 'node'. A horizontal arrow points from the right side of the 'xferqueue.list' oval to the left side of the 'node' oval, indicating that 'xferqueue.list' points to 'node'.











var@dcookies.c

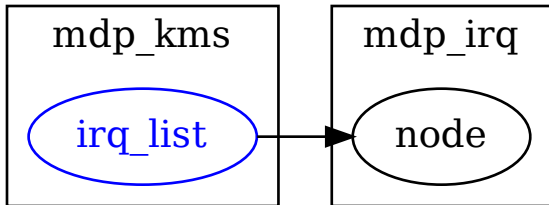
dcookie_users

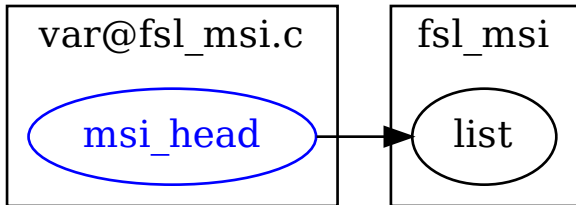
dcookie_user@dcookies.c

next

```
graph LR; subgraph Box1 [var@dcookies.c]; A(dcookie_users); end; subgraph Box2 [dcookie_user@dcookies.c]; B((next)); end; A --> B;
```

The diagram consists of two rectangular boxes. The left box is labeled 'var@dcookies.c' and contains an oval labeled 'dcookie_users' in blue text. The right box is labeled 'dcookie_user@dcookies.c' and contains an oval labeled 'next'. A horizontal arrow points from the right side of the 'dcookie_users' oval to the left side of the 'next' oval.





var@nfnetlink_cthelper.c

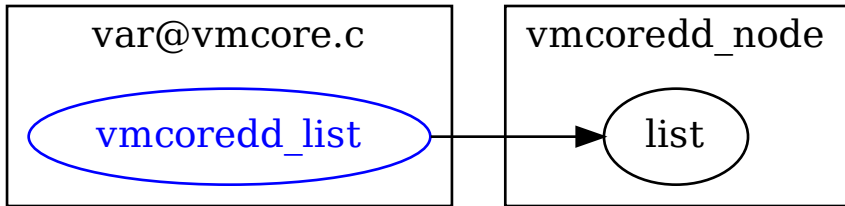
nfnl_cthelper_list

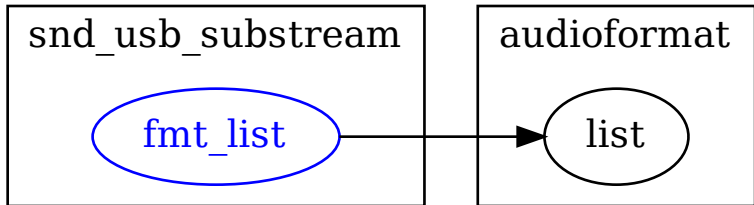
nfnl_cthelper@nfnetlink_cthelper.c

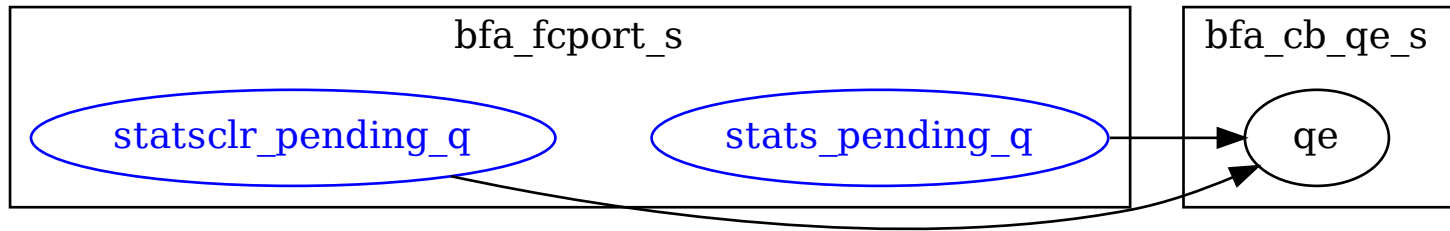
list



```
graph LR; subgraph LeftBox [var@nfnetlink_cthelper.c]; direction TB; A((nfnl_cthelper_list)); end; subgraph RightBox [nfnl_cthelper@nfnetlink_cthelper.c]; direction TB; B((list)); end; A --> B;
```





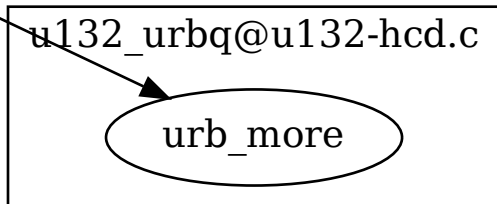
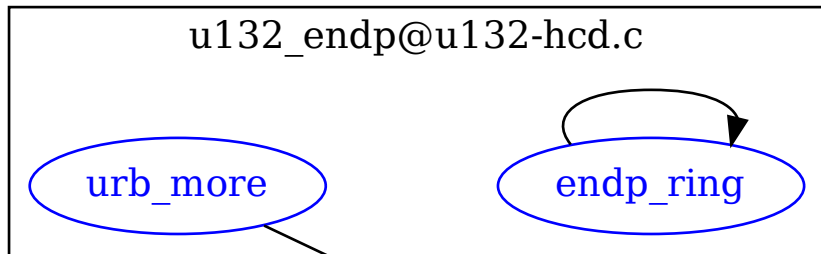
var@composite.c

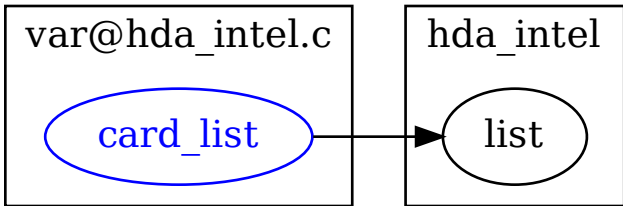
component_clks

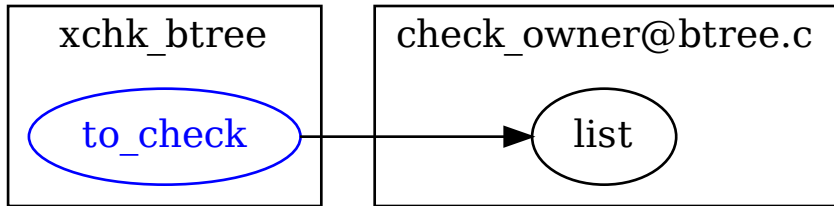
component_clk@composite.c

link

The diagram consists of two rectangular boxes. The left box is labeled 'var@composite.c' and contains an oval labeled 'component_clks' with a blue border. The right box is labeled 'component_clk@composite.c' and contains an oval labeled 'link'. A horizontal arrow points from the right side of the 'component_clks' oval to the left side of the 'link' oval.







var@trace_stat.c

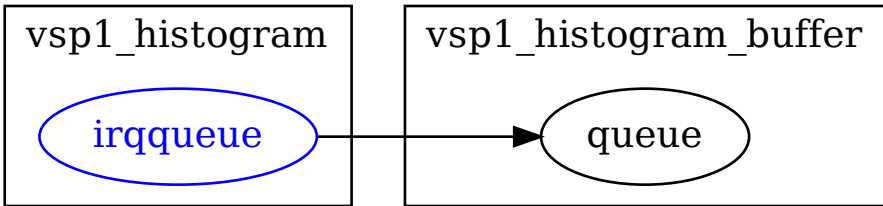
all_stat_sessions

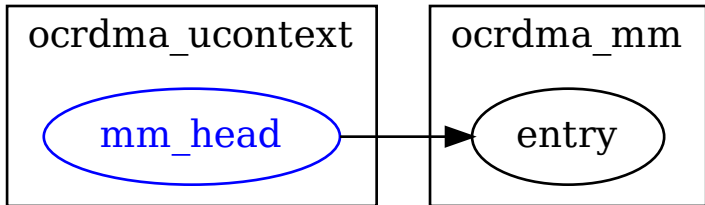
stat_session@trace_stat.c

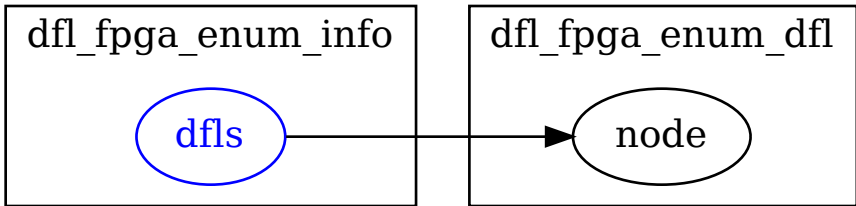
session_list

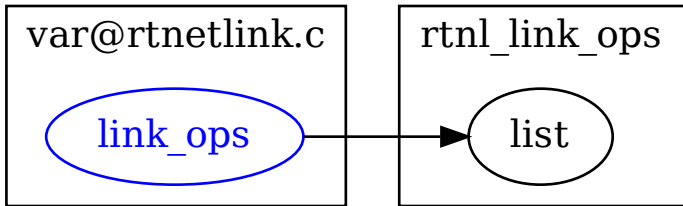
```
graph LR; A[all_stat_sessions] --> B[session_list]
```

The diagram illustrates a pointer relationship between two variables. On the left, a rectangular box labeled 'var@trace_stat.c' contains an oval labeled 'all_stat_sessions' with a blue border. On the right, a rectangular box labeled 'stat_session@trace_stat.c' contains an oval labeled 'session_list' with a black border. A horizontal arrow points from the right side of the 'all_stat_sessions' oval to the left side of the 'session_list' oval, indicating that 'all_stat_sessions' points to 'session_list'.









rionet_net@rionet.c

peers

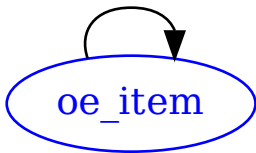
rionet_peer@rionet.c

node

```
graph LR; subgraph rionet_net [rionet_net@rionet.c]; peers; end; subgraph rionet_peer [rionet_peer@rionet.c]; node; end; peers --> node;
```

The diagram illustrates a relationship between two modules. The left module, labeled 'rionet_net@rionet.c', contains an oval labeled 'peers'. The right module, labeled 'rionet_peer@rionet.c', contains an oval labeled 'node'. A horizontal arrow points from the 'peers' oval to the 'node' oval, indicating a mapping or data flow from the network peers to a specific node in the peer module.

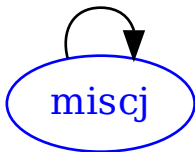
ceph_object_extent



efx_rss_context



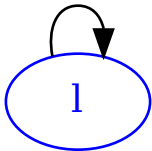
threshold_block



dw_edma_chunk



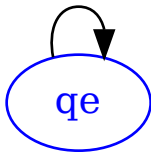
vf_mac_filter



udp_replicast@udp_media.c



bfa_mem_dma_s



dm_clone_region_hydration@dm-clone-target.c



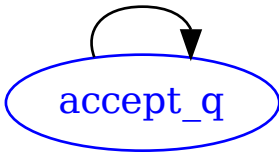
ishttp_cl_rb



drbd_epoch

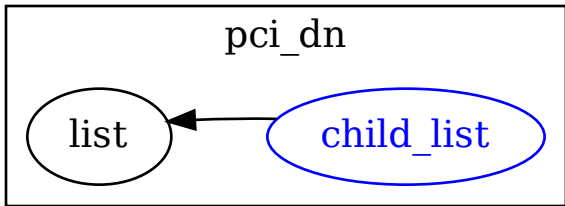


smc_sock



tipc_tfm@crypto.c





hfsc_class@sch_hfsc.c

siblings

children

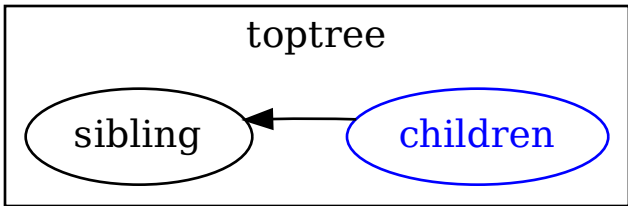


```
graph LR; children --> siblings
```

The diagram consists of a rectangular frame containing two ovals. The oval on the left is labeled 'siblings' and has a black border. The oval on the right is labeled 'children' and has a blue border. A black arrow points from the 'children' oval to the 'siblings' oval.

nfs_lock_context





ib_qp_security



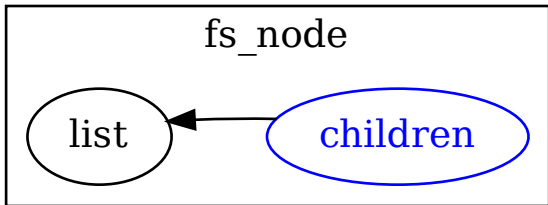
shared_qp_list

configs_dirent

s_sibling

s_children





mesh_preq_queue



dw_edma_burst



sit_entry_set



rxq_entry_t



swnode@swnode.c

entry

children



```
graph LR; children --> entry
```

The diagram consists of a rectangular frame containing two ovals. The oval on the left is labeled 'entry' and has a black border. The oval on the right is labeled 'children' and has a blue border. A black arrow points from the right side of the 'children' oval to the left side of the 'entry' oval.

iucv_sock



accept_q

mlx5_ib_flow_handler



cfg80211_bss



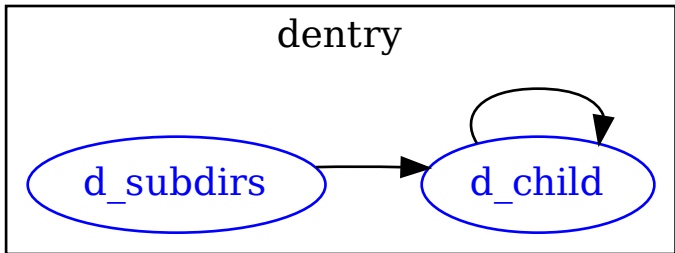
A diagram within a black rectangular border. At the top is the text 'cfg80211_bss'. Below it is a blue oval containing the text 'nontrans_list'. A curved black arrow points from the left side of the oval up to the text 'cfg80211_bss'.

nontrans_list

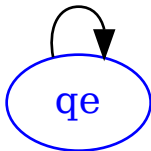
dentry

d_subdirs

d_child



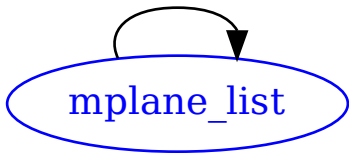
bfa_mem_kva_s



packet_data@dell_rbu.c



dpu_plane@dpu_plane.c



ehea_mc_list



nvkm_object

head

tree



```
graph LR; tree((tree)) --> head((head));
```

The diagram illustrates the internal structure of an `nvkm_object`. It is represented as a rectangular box containing two ovals. The oval on the left is labeled `head` and has a black border. The oval on the right is labeled `tree` and has a blue border. A black arrow points from the `tree` oval to the `head` oval, indicating a pointer relationship.

txq_entry_t



tipc_peer@monitor.c



bt_sock



accept_q

netdev_name_node



mbox_list



nfc_llcp_sock



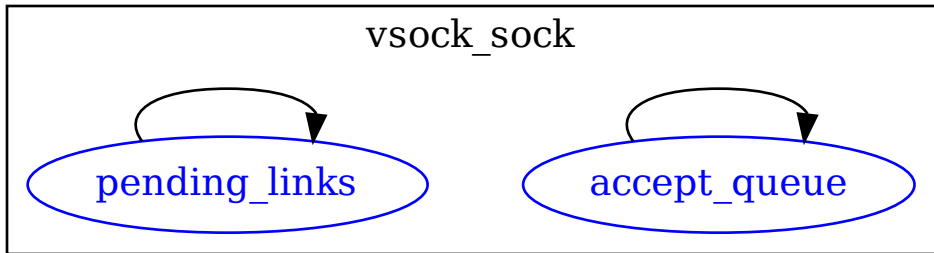
accept_queue

kprobe



axi_dma_desc





sock_msg_q



pci_dev_resource@setup-bus.c



usb_phy_roothub@phy.c



ext_wait_queue@mqueue.c



snd_info_entry

list

children



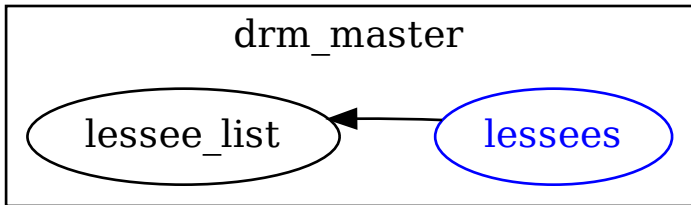
```
graph LR; children(children) --> list(list);
```

The diagram illustrates the structure of an `snd_info_entry`. It is enclosed in a rectangular box. At the top, the text `snd_info_entry` is centered. Below it, there are two ovals. The left oval is labeled `list` and has a black border. The right oval is labeled `children` and has a blue border. A black arrow points from the `children` oval to the `list` oval, indicating a pointer relationship.

mongroup



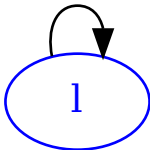
crdtgrp_list



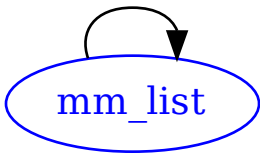
ishttp_cl_tx_ring



vf_macvlans



mm_slot@ksm.c



octeon_dispatch

