



HAL
open science

Development of a STEP-NC – Enabled Interoperable and Open Architecture CNC Controller

Julien Bechtold, Christophe Danjou, Walid Jomaa

► To cite this version:

Julien Bechtold, Christophe Danjou, Walid Jomaa. Development of a STEP-NC – Enabled Interoperable and Open Architecture CNC Controller. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2022, Gyeongju, South Korea. pp.252-259, <10.1007/978-3-031-16407-1_30>. <hal-05386064>

HAL Id: hal-05386064

<https://inria.hal.science/hal-05386064v1>

Submitted on 27 Nov 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.



Development of a STEP-NC – Enabled Interoperable and Open Architecture CNC Controller

Julien Bechtold^(✉), Christophe Danjou, and Walid Jomaa

Polytechnique Montréal, Montréal, QC H3T 1J4, Canada
bechtold.julien@gmail.com

Abstract. Most of modern computer numerical control (CNC) machine tools (CNCMTs) use standard G-code (ISO 6983) – based controllers. However, during the generation of G-codes by Computer-Aided Manufacturing (CAM) software, several machining information could be lost, which hinder the development of smart interoperable CNCMTs required for modern manufacturing environment. The STEP-NC (STEP compliant Numerical Control), which is a machine-tool control language governed by the ISO 14649 standard, was developed to overcome the G-code programming limitations. Over the last two decades, several research studies attempted to develop a new generation of CNCMTs controllers taking advantage of STEP-NC capabilities. Nevertheless, these controllers are customized for specific applications, complex to replicate, costly, and not yet accepted in the manufacturing industry. In this paper, an easy to replicate, noninvasive, and low-cost interpreted open architecture STEP-NC controller is developed. This controller uses a Single Board Computer which can be directly connected to a commercial lathe. The controller software is developed based on C++ and can interpret a ISO14649 STEP-NC file, generate a toolpath, simulate the toolpath, and communicate with the CNCMT. The developed STEP-NC controller was successfully tested by simulating the toolpath during the machining of a test part available in the ISO standard.

Keywords: STEP-NC · Open architecture controller · CNC machine tools · Machining

1 Introduction

Since the invention of numerically controlled machine-tools in the 1950s, their uses have steadily increased in the manufacturing industry until the beginning of the 1980s, were the first CNCMTs appeared. As of that date, the capabilities and performance of CNCMTs have evolved significantly. From its creation in the 1960s and normalization as ISO 6983 [1], the so-called G-code language is mainly used for programming CNCMTs. However, G-code is a low-level programming language containing basic information related to the toolpath and commands the machine must execute during the machining. Besides, G-codes allow only one-way communication between the controller and the CNCMT, that

is it does not allow any systematic feedback from the CNCMT to the Computer Aided Manufacturing (CAM) and/or the Computer Aided Design (CAD) levels. This limitation pushed CNCMTs manufacturers to add specific functions and commands requiring the use of post-processors to generate G-code programs compatible with their CNCMTs. It is worth noting that, current commercial controllers are totally closed and cannot be modified by the customer, they act like a black box.

STEP-NC, as a new standardized machine-tool control language, was developed to resolve many of the G-code and current controllers' limitations. It was standardized as the ISO 14649 [2] which is an extension of STEP (Standard for the Exchange of Product model data), standardized as ISO 10303 [3]. With STEP-NC it is possible to use a single standardized language throughout the numerical manufacturing chain, from the CAD to the CNC controller. In addition, STEP-NC can ensure a bidirectional communication transfer without any loss of information. Figure 1 shows the current manufacturing chain and the concept of STEP-NC-enabled interoperable manufacturing chain.

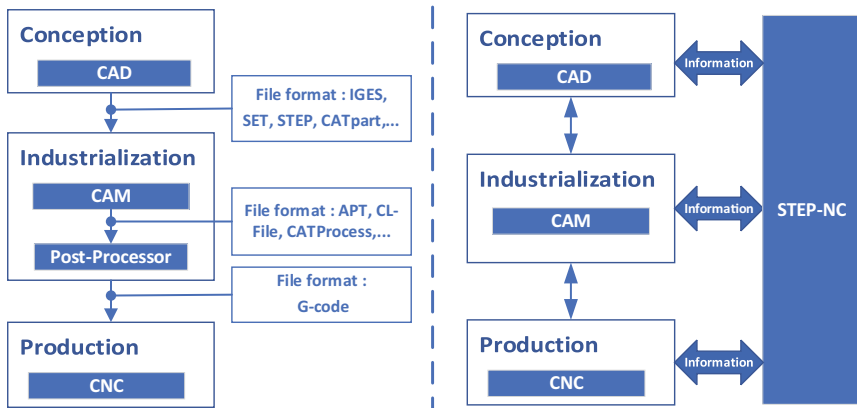


Fig. 1. Current manufacturing chain (left), manufacturing chain with STEP-NC (right)

With the STEP-NC standard, only one file is used along the numerical manufacturing chain. It is a high-level language which, in addition to how to make information (such as the tool path) as for the G-code, can store additional relevant information about what-to-make (features, operations, machining strategies, tool geometry, ...) [4]. Today, there is no commercial CNC controller working based on STEP-NC program. In the new era of industry 4.0, the scientific community seeks to develop STEP-NC – based open architecture controllers (OAC) to meet the economical and performance needs of modern manufacturing systems. To this end, the main contribution of the present article is the development of a new STEP-NC controller, allowing intelligent monitoring of CNCMT and easier integration of STEP-NC in a production environment.

In the next section, a state of the art of STEP-NC-based controllers will be introduced. Then, in Sect. 3, the developed software and hardware of the STEP-NC controller will be described and discussed. Finally, Sect. 4 will be dedicated to the validation of the proposed solution.

2 State of the Art

The first STEP-NC controller prototypes have been developed earlier in 2003 and their evolution has continued until today. Three types of concepts were documented in the open literature and are summarized in Fig. 2 [4].

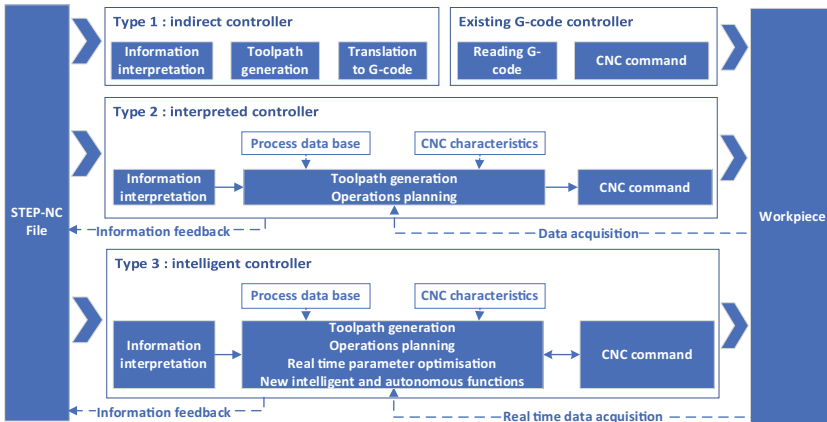


Fig. 2. Implementation of STEP-NC controllers, adapted from [4]

2.1 Type 1: Indirect STEP-NC Controllers

The first prototypes developed can be classified as type 1 or indirect controllers. These prototypes convert STEP-NC data into a G-code, so it is executable on a standard controller. In 1999, in collaboration with Siemens group, the European consortium ESPRIT [5] were among the first organizations able to run a single STEP-NC file on multiple types of CNCMTs using the Sinumerik 840D controller from Siemens Inc. Even so, this approach still suffers some limitations such as the use of G-code and the resulting loss of information after generating the toolpath. Furthermore, there is still a need for the post processor and for proprietary closed controllers. It is worth noting that this approach was useful to prove the feasibility of STEP-NC controllers at the beginning, but it cannot so far be a viable solution.

2.2 Types 2 and 3: Interpreted and Intelligent STEP-NC Controllers

Type 2 or interpreted controllers are capable of interpreting and executing STEP-NC files without using any G-code. The most relevant prototype of type 2 STEP-NC controller present in the literature are summarized in Table 1.

Presently, researchers are working on developing controllers with intelligent function, such as CLM feature [6] or automatic real time parameter correction [7]. However, a fully smart controller (Type 3) does not yet exist. It is the final goal of the STEP-NC

Table 1. Most relevant type 2 STEP-NC prototype controllers

Reference	Language	Hardware	Limitation
[4, 8]	C++	2 PC	Complex and costly architecture
[9]	C++	PC	Complex and costly architecture, no toolpath simulation or visualization
[10]	C	Microcontroller	More difficult to code, less user-friendly
[11]	C++	PC	Complex architecture with external control cards and acquisition cards
[12]	LabVIEW	PC	Custom control cards, proprietary software
[13]	C++	SBC	Lack of toolpath visualization, not tested on a commercial CNCMT

community, it should be able to optimize machining parameters in real time, to be a web-based and collaborative between machines and manufacture.

Conclusively, STEP-NC controllers are currently: (1) Complex and hard to replicate: most prototypes use a PC with additional (and mostly custom) controller and acquisition cards to communicate with the CNCMT. (2) High cost: a PC and custom controller cards are costly (estimated at more than 5000 \$). (3) Cumbersome: most prototypes use a hardware that is external to the body of the CNCMT. Dharmawardhana, et al. [13] proposed a relevant solution based on a Single Board Computer (SBC) and proves that it is possible to make economical and compact STEP-NC controllers. The lack of toolpath visualization and simulation are however missing, and the facts that they use an Arduino Mega in addition to a RPi induce additional challenges during the integration of the controller.

To this end, the present research work aims to the development of an open architecture interpreted STEP-NC controller for a CNCMT with an emphasis on the miniaturization, integration, reproducibility, and cost effectiveness.

3 Development

As we have seen in the last chapter, there is a need to develop an open architecture interpreted STEP-NC controller. The emphasis will be put on developing a cheap and compact controller that is easy to replicate. The developed STEP-NC controller is devoted to control turning operations on a CNC mini-lathe.

3.1 Design of the Architecture

The software of the controller is developed based on the open-source C++ programming language which is compatible with a wide range of hardware. The C++ is an object-oriented and low-level programming language making the translation of STEP-NC information more accurate and intuitive since STEP-NC is an object-oriented language too. Moreover, compared to other programming languages, the C++ enable higher

computation speed and more performant programming capabilities. Once the software language is chosen, it is necessary to focus on the hardware architecture.

The SBC is used as a hardware support for the developed controller. One of the well-known is the RPi. SBC which consists of a single electronic board including all PC components. The RPi SBC is not expensive, compact, and especially it has input/output ports called General Purpose Input/Output (GPIO) which allows communication directly with a CNCMT and other sensors without any additional control board. The RPi 4B model is selected for the hardware support. It has a quad-core ARM processor clocked at 1.5 GHz, 8 GB of RAM, and a 40-pin GPIO. This SBC meets the requirements of the controller and the objective of the project, that it is available for about 100 CA\$ and supports several languages, including C++ and Python. Thanks to its GPIO, it can communicate with external sensors and/or actuators. Furthermore, it offers many connectivity possibilities thanks to its Ethernet, camera, USB and HDMI Port and Wi-Fi connection.

To be able to communicate with the driver, the controller must send 5 V pulses. But the GPIO of the RPi works only with 3 V. So, a shifter card is used to upgrade the RPi from 3 V to 5 V. To protect the RPi and to facilitate the connection, a Breakout Board (BOB) is used, allowing the interface between the RPi and the driver. A BOB for 5 axis MOCN with optical couplers is available for less than 20 CA\$. The prototype controller is illustrated in Fig. 3.

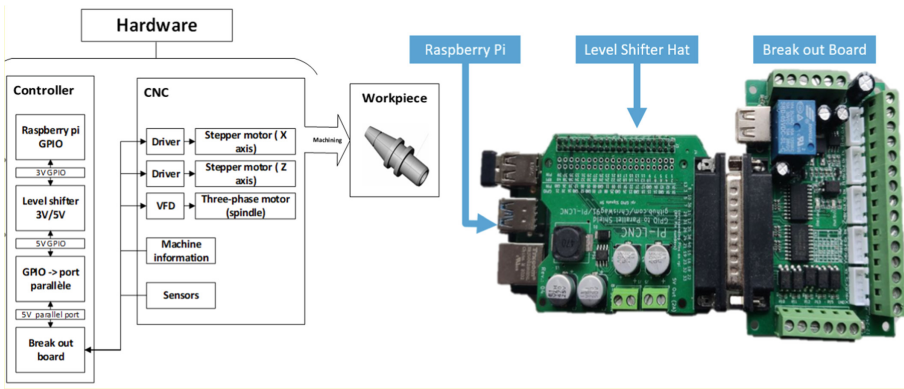


Fig. 3. Designed hardware of the developed STEP-NC controller.

Since the BOB as well as the GPIO can both send and receive signals, sensors or other actuators can therefore be connected to the controller in future development.

The signals sent by the RPi to the drivers must be sent in real time. Thus, the operating system (OS) must deal with real time signals. Contrary to a Windows system, which requires a paid and closed module to be transformed into a real time operating system (RTX of Venture COM is often used), the RPi OS has an open and free real time patch that can be installed.

3.2 Development of the Controller

The structure of the controller is as follows: First the STEP-NC file interpreter read all information that is stored in the STEP-NC file and instantiates the corresponding C++ objects. Then, the toolpath is generated as well as all the actions necessary to produce the workpiece by using the interpreted data and the characteristics of the machine-tool. The toolpath is then sent to the low-level controller. This is the part of the software that must work in real time since it oversees sending the information to the CNCMT. It is composed of two modules: The Numerical Control Kernel (NCK) and the Programmable Logic Control (PLC). The NCK interprets the information of the tool path and generates the signals that allow the movement of the axes of the CNCMT. To generate those signals, the NCK interpolate the toolpath and calculate the speed of the tool, considering the maximum acceleration allowable by the CNCMT. The general structure of the NCK and the algorithms used are given in Theory and Design of CNC [14]. The Programmable Logic Control (PLC) manages all the auxiliary actuators such as the spindle speed, the tool magazine, or the activation of the coolant system. All these instructions are sent to the motor drivers of the CNCMT and will allow the desired part geometry to be produced.

The functionalities of the software are as follows: The “Interpretation” function allows the reading and translation of the information included in the STEP-NC file. It, then, generates the toolpath and all the information necessary for moving the axes and managing the machine-tool’s functions. The “Automatic” function allows the execution of the STEP-NC file and the machining of the part by sending the corresponding operational signals to the CNCMT actuators. The “Manual” function allows the manual movement of the CNCMT axes via the keyboard connected to the RPi. The “Visualization” function displays the visualization of the part, the toolpath, and the interpolated points. The “Animation” function generates a simulation video of the movement of the axes in real time. The “Edit” function allows modifying some defined parameters, for example the dimensions of the stock part or the start position of the tool. The “Information” function allows displaying additional information included in the STEP-NC file, such as the name of the workingsteps or the workpiece parameters.

4 Case Study

The integration of the new controller with the EMCO PC TURN 55 CNC mini-lathe is in progress. Hence, in this paper, we merely present a simulation of the machining of a test part available in the ISO standard [14].

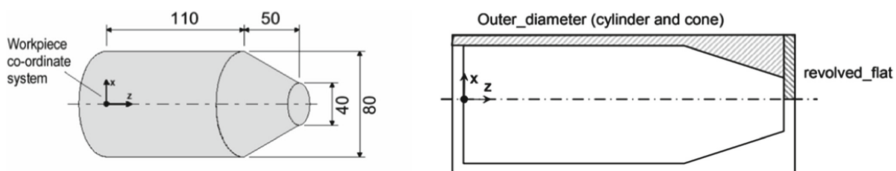


Fig. 4. Turning test piece from Annex D ISO14649 part 12 [15]

The test part that will be used is illustrated in Fig. 4. It is composed of three features: OUTER_DIAMETER for the cone, OUTER_DIAMETER for the cylinder and a REVOLVED_FLAT for the end face. To machine these features, several operations are defined. First the end face is machined with a rough facing, then the two OUTER_DIAMETER features are machined using a rough contouring operation. Finally, a finishing path is performed.

Once the STEP-NC file is imported, the information is read and is interpreted, and the toolpath is generated in approximately 500 ms. Figure 5 shows the generated toolpath.

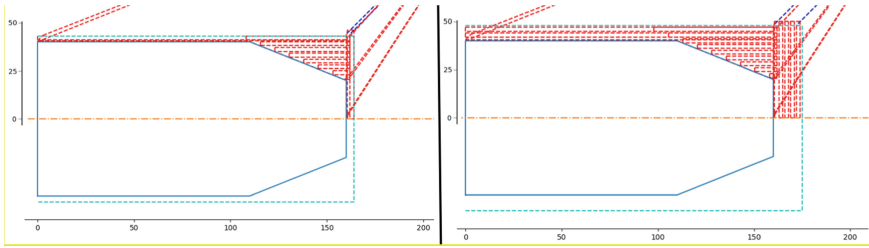


Fig. 5. Toolpath visualization of the test part. Stock dimension: left (length 165 mm, diameter 88 mm), right (length 175 mm, diameter 96 mm)

It is noteworthy to mention that, with the developed controller, it is possible to modify the size of the stock before the machining, using the edit function. The toolpath can be automatically recalculated without needing to manually machine the stock or modifying the CAD. The right image on Fig. 5 shows the toolpath for a cylinder stock different from the recommended stock (Fig. 5 - left image). In the future it is possible to add automatic probing to measure the size of the stock to adapt the toolpath.

5 Conclusion

A new STEP-NC OAC controller is developed, allowing to interpret STEP-NC files, to generate the toolpath and to communicate with a CNC machine-tool. The open-source C++ code and the proposed hardware make this controller more accessible and simpler to integrate compared to previous controllers which use home – made electronic cards and proprietary software. Moreover, with its cost less than 200\$, it is among the less expensive developed STEP-NC controllers as it cost less than a tenth of the price of other controllers.

Besides, the prototype confirms the feasibility of a controller with a simple and low-cost architecture. It is a step towards making STEP-NC more accessible and acceptable in the manufacturing industry. Future tests will be done to machine the test piece on a commercial lathe allowing the test and validation of new functionalities. Indeed, the controller allows adding external sensors and/or video camera. From a software point of view, many modules can be easily developed under C++ or Python directly on the RPi. These modules will allow improving productivity, safety, and interoperability.

References

1. ISO 6983-1:1982 Commande numérique des machines — Format de programme et définition des mots adresses —, ISO (1982)
2. ISO 14649-1: Overview and fundamental principles, ISO (2003)
3. ISO 10303 Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits, ISO (1994)
4. Suh, S.H., Lee, B.E., Chung, D.H., Cheon, S.U.: Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Comput. Aided Des.* **35**(12), 1069–1083 (2003). [https://doi.org/10.1016/s0010-4485\(02\)00179-3](https://doi.org/10.1016/s0010-4485(02)00179-3)
5. Mueller, P., Hyu, Y.T.: ESPRIT Project EP 29708 STEP-Compliant Data Interface for Numerical Controls (STEP-NC), STEP-NC consortium (2001)
6. Brecher, C., Verl, A., Lechler, A., Servos, M.: Open control systems: state of the art. *Prod. Eng. Res. Devel.* **4**(2–3), 247–254 (2010). <https://doi.org/10.1007/s11740-010-0218-5>
7. Zhao, Y.F., Habeeb, S., Xu, X.: Research into integrated design and manufacturing based on STEP. *Int. J. Adv. Manuf. Technol.* **44**(5–6), 606–624 (2008). <https://doi.org/10.1007/s00170-008-1841-6>
8. Choi, I., Suh, S.H., Kim, K., Song, M., Jang, M., Lee, B.E.: Development process and data management of TurnSTEP: a STEP-compliant CNC system for turning (in English). *Int. J. Comput. Integr. Manuf.* **19**(6), 546–558 (2006). <https://doi.org/10.1080/09511920600622072>
9. Lee, W., Bang, Y.B., Ryou, M.S., Kwon, W.H., Jee, H.S.: Development of a PC-based milling machine operated by STEP-NC in XML format. *Int. J. Comput. Integr. Manuf.* **19**(6), 593–602 (2006). <https://doi.org/10.1080/09511920600623674>
10. Calabrese, F., Celentano, G.: Design and realization of a STEP-NC compliant CNC embedded controller. In: 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007), 25–28 September 2007, pp. 1010–1017 (2007). <https://doi.org/10.1109/EFTA.2007.4416894>
11. Po, H., Hongya, F., Zhenyu, H., Dedong, H.: A closed-loop and self-learning STEP-NC machining system. In: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 8–11 July 2014, pp. 1598–1603 (2014). <https://doi.org/10.1109/AIM.2014.6878312>
12. Latif, K., Yusof, Y., Nassehi, A., Alias Imran Latif, Q.B.: Development of a feature-based open soft-CNC system. *Int. J. Adv. Manuf. Technol.* **89**(1–4), 1013–1024 (2016). <https://doi.org/10.1007/s00170-016-9124-0>
13. Dharmawardhana, M., Ratnaweera, A., Oancea, G.: STEP-NC compliant intelligent CNC milling machine with an open architecture controller. *Appl. Sci.* **11**(13) (2021). <https://doi.org/10.3390/app11136223>
14. Suh, S.-H., Kang, S., Chung, D.-H., Stroud, I.: *Theory and Design of CNC Systems* (2008)
15. ISO 14649-12: Process data for turning, ISO (2003)