



**HAL**  
open science

# Decoding Methods in LED-to-Smartphone Bidirectional Communication for the IoT

Alexis Duque, Razvan Stanica, Hervé Rivano, Adrien Desportes

► **To cite this version:**

Alexis Duque, Razvan Stanica, Hervé Rivano, Adrien Desportes. Decoding Methods in LED-to-Smartphone Bidirectional Communication for the IoT. Global LiFi Congress - 1st edition, Feb 2018, Paris, France. pp.1-6, 10.23919/GLC.2018.8319118 . hal-01683629

**HAL Id: hal-01683629**

**<https://inria.hal.science/hal-01683629v1>**

Submitted on 14 Jan 2018

**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.

# Decoding Methods in LED-to-Smartphone Bidirectional Communication for the IoT

Alexis Duque<sup>\*†</sup>, Razvan Stanica<sup>\*</sup>, Herve Rivano<sup>\*</sup>, Adrien Desportes<sup>†</sup>

<sup>\*</sup> Univ Lyon, INSA Lyon, Inria, CITI

F-69621, Villeurbanne, France

<sup>†</sup>Rtone, Lyon, France

alexis.duque@insa-lyon.fr

**Abstract**—The use of visible light for bidirectional communication between regular smartphones and the small LEDs integrated in most consumer electronics nowadays raises new challenges. In this paper, we enhance the state of the art with an efficient image processing algorithm to accurately detect the LEDs and decode their signal in real time. To enable flash-to-LED communication on non-rooted smartphones, we propose encoding and decoding methods that cope with the inaccurate flashlight clock rate.

**Index Terms**—Visible Light Communication, Internet of Things, Optical Camera Communication, LED-to-Camera Communication, Flash-to-LED, Smartphone, Image processing

## I. INTRODUCTION

With the exponential growth of the Internet of Things (IoT), people now expect every household appliance to be smart and connected. At the same time, smartphones have become ubiquitous in our daily life. Their continuous performance improvement and their compatibility with a broad range of radio protocols as WiFi, Bluetooth Low Energy (BLE) or Near Field Communication (NFC) make them the most convenient way to interact with these smart objects. However, providing wireless connectivity with BLE or NFC means adding an extra chipset and an antenna, increasing the object size and price.

This kind of hardware modification is not without impact for the manufacturers: even if the radio chip cost is negligible for a single unit, it may become huge when millions of products are sold.

Besides, many of these products already have a micro-controller and several light emitting diodes (LED), which are the only requirement to enable bidirectional visible light communication (VLC) [10]. On the other hand, previous works demonstrate the possibility of receiving information through visible light using an unmodified smartphone thanks to its camera [1].

In this paper, we propose to combine these two approaches, evaluating a bidirectional VLC transmission system between a low-cost, low-power colored LED that is part of an IoT device and an unmodified and non-rooted smartphone [4]. The IoT device is thus able to send and receive information at the same time through its LED, while the smartphone uses its flashlight and camera respectively to send and receive data. The additional costs to make a traditional device part of the IoT are therefore minimal.

Achieving this requires us to cope with several issues unsolved so far. Our first implementation based on state of the art

solutions gave us the following engineering insights: **a)** image processing with computer vision algorithms and filters to detect the LED position on the picture was too slow to achieve real-time processing; **b)** since the LED on the picture is small, the system becomes very sensitive to the user motion as the LED position highly varies between consecutive frames. Thus, performing LED detection only once to fasten the process cannot be envisaged [15]; **c)** smartphone manufacturers do not allow controlling the flashlight at the kernel level without root access, enforcing the use of their high-level API; **d)** the flashlight clock rate varies between smartphones, and even on the same device, depending on the system load; **e)** the flashlight API is not time accurate, which adds latency and heavy jitter to the signal.

This paper presents the design and the implementation of decoding processes that resolve these concerns. Our main contributions are two-fold: **1)** we deal with the LED-to-smartphone issues (a,b) and introduces an original decoding algorithm that decodes every frame in real-time. By bypassing computer vision techniques, our algorithm processes every 1920x1080 picture in 18.4 ms on average on a Nexus 5 with a Java implementation. Using native C code, the whole processing takes only 3.6 ms. These results outperform previous works [15] and add robustness against user motion. **2)** we propose smart mechanisms and protocols that cope with the flashlight signal jitter and detect its clock rate whatever the smartphone (c,d,e). We designed a modulation scheme based on a variable pulse width modulation (VPWM) that we adapted abetted by our experimental observations of the smartphones flashlight behavior. Our frame format includes synchronization patterns that let the device detect the beginning of a transmission, acquire the clock rate, correct both the clock jitter and bias by adapting the signal sampling itself at runtime. These key features harden the flash-to-LED channel and make it suitable for feedback, wake-up or communication purposes.

By solving these issues, we build a reliable and bidirectional VLC system for off-the-shelf smartphones and low-end IoT devices at little expense, opening the doors to a wide range of applications.

The rest of this paper is organized as follow. We describe our bidirectional LED-smartphone VLC system and the experimental platform in Sec. II. Sec. III focuses on the LED-to-smartphone communication link and introduces our

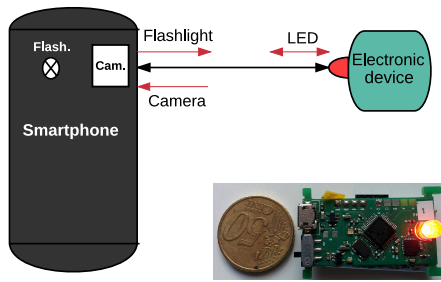


Fig. 1. On the left, our bidirectional VLC system. On the right, our smart VLC device prototype.

original decoding algorithm that is the first contribution of this paper. The second contribution that enables smartphone-to-LED communication is then presented in Sec. IV before the concluding remarks in Sec. V.

## II. SYSTEM DESCRIPTION

### A. Smart VLC device

We designed a 40x20 mm printed circuit board (PCB) shown in Figure 1 that supports a red 5 mm LED, a micro controller unit (MCU), and a surface mount device (SMD) RGB LED whose color can be easily changed.

The MCU is a low cost and low power Cortex M0+ STM32L051, an MCU similar to those already integrated in most household appliances. We configured the clock speed at just 8MHz using the high speed external crystal oscillator. The board is powered by a 3.7 V polymer lithium battery, rechargeable through USB. Following the recommendations of previous studies [1], [8], a simple OOK modulation scheme is used. We consider a symbol rate varying from 6 kHz to 10 kHz, which is a suitable transmission rate for receivers using the rolling shutter effect [1]. To avoid any flickering effect, we use the Manchester coding proposed in the literature [1], [8].

### B. Smartphone side

On the smartphone side, we use an LG Nexus 5 smartphone running Android Marshmallow version number 6.0.1. It has a Qualcomm Snapdragon 800 quad-core 2,26 GHz CPU and 2 Go RAM. Its 8 megapixels 1080p 1/3.2" CMOS sensor with 1.4  $\mu\text{m}$  pixel size can capture up to to 30 frames per second and supports advanced imaging application provided by the Camera2 API. We have developed an Android application that sets up the camera parameters to observe the rolling shutter effect produced by the modulating LED. For that, based on the work by Camps-Mur *et al.* [8], we set a very short exposure time and an increased sensor sensitivity, to 100  $\mu\text{s}$  and ISO 10000, respectively. As soon as a new frame is available, the application creates and starts a new thread to process and decode the picture on the background as further detailed in Sec. III

## III. ENABLING LED-TO-CAMERA COMMUNICATION

The communication between an LED and a smartphone uses the rolling shutter effect of the smartphone camera [1]. In LED-to-camera VLC, the LED is modulating the light it

emits using an on-off keying (OOK) scheme, which results in a dimmed light intensity, while also encoding information. On the smartphone side, the camera is used to take a picture of the transmitter LED, or of a surface illuminated by it. As a consequence of the rolling shutter effect, the OOK modulation used by the LED will create a series of dark and illuminated stripes on the picture. These stripes encode the information transmitted by the LED, much alike a bar code. However, unlike a bar code, the information is structured in packets and it changes from one picture to another, resulting in a wireless transmission of information.

Previous studies [8], [16] demonstrate the performance of LED-to-camera communications, by using commercial lighting LED bulbs. Instead, our focus is on small color LEDs, such as those integrated in most daily usage objects. For instance, *RollingLight* [15] uses a 500 mA, 1340 lm LED (Bridgelux bxra-56c1100 LED), while the LEDs we use in our study are CREE C503B-RAN, with 20 mA and 1.2 lm. While many concepts from the literature can be reused, regardless the type of the LEDs, the reduced size and power creates some new challenges. Our focus is on a novel decoding solution, detailed in this section, which allows the detection of small region of interests (ROI) on the picture frames captured by the smartphone. By developing a prototype testbed, described in Sec. II, we evaluate our LED-to-camera communication framework in a series of extensive tests in Sec. III-B. These experiments demonstrate that our algorithm not only decodes the frames in real time, i.e. below the camera frame rate, but also performs ROI detection in every frame and enables MIMO-like communication when several emitters are used.

### A. Decoding algorithm

The current state of the art solution, *RollingLight* [15], functions on ROIs of at least 600 pixels, achieving a throughput of 12 bytes/s. In *RollingLight*, the picture frame processing time is around 18 ms. However, this only counts the time required to extract the embedded information, and it does not take into account the ROI detection step, which is time consuming and performed only once, when the first frame is processed. Without skipping the ROI detection step, a feature which makes it deeply sensitive to motion, *RollingLight* would not be able to decode the transmitted information in real-time, i.e. it would not be able to decode information fast enough to receive the 30 frames/s allowed by most modern smartphone cameras. As a matter of fact, all the previous LED-to-camera solutions [12], [14], [15] rely on computer vision (CV) filters to detect the ROIs, which results in a frame processing time much higher than 30 ms (the time required to process 30 frames per second).

Instead, our method is specifically designed for small colored LEDs, allowing to process ROIs as small as 160 pixels and integrating a real-time decoding algorithm able to determine the ROI position in each frame. The frame processing algorithm is decomposed in five steps:

**Step 1. Image acquisition (S1):** handling incoming captures in the YUV color space and extracting only the luma

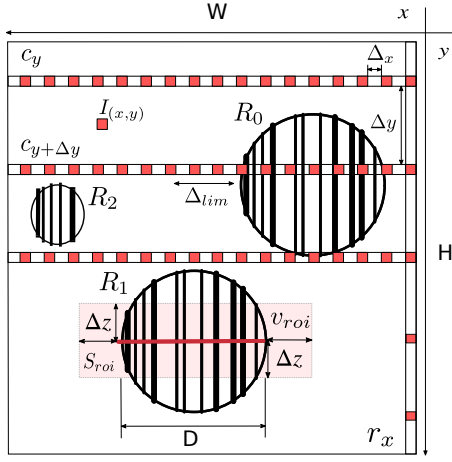


Fig. 2. A picture frame containing three ROIs. The smaller ROI on the left will be skipped by the decoding algorithm, as it does not contain any valid PHY-SDU.

component Y, which allows to distinguish between dark and illuminated stripes.

**Step 2. ROI detection (S2):** finding the LED position on the picture, excluding LEDs that are not transmitting and reducing the length of the Y component vector.

**Step 3. Signal equalization (S3):** enhancing the received signal to deal with inter-symbol interferences (ISI) and background noise, which allows to recover the signal clock rate.

**Step 4. Signal quantization (S4):** thresholding and binarization of the signal to recover the digital signal.

**Step 5. Decoding and error checking (S5):** decapsulating the packets, decoding the Manchester chip and checking data integrity with vertical redundancy check (VRC).

In order to detail S2 and S3, which represent advancements with respect to the state of the art, let us first introduce a series of notations, summarized in Fig. 2. The smartphone camera is configured to record frames of  $W$  per  $H$  pixels in the YUV color space. The transmitted information is embedded only in the luma component Y, represented by a matrix  $I$ , where  $I(x, y)$  is an 8-bit integer coding the pixel light intensity. Matrix  $I$  can also be represented using row and column vectors  $r_x$  and  $c_y$ , i.e.  $I(x, y) = r_x(y) = c_y(x)$ .

An LED captured by the camera will result in a ROI  $R_i$  in the frame. Multiple ROIs can be present in a picture frame, but some of these ROIs might be produced by regular LEDs, not carrying any information, while others, under a certain size, are too small to carry any useful information (e.g.  $R_2$  in Fig. 2). As further explained in [1], rolling shutter cameras sequentially expose their rows of pixels with a time duration  $\delta_r$ . In this way,  $c_y$  is a record of the light intensity signal during a time interval of  $W \cdot \delta_r$ , where each OOK symbol has a width of  $S$  pixels (depending on the transmission symbol rate). Therefore, if the number of symbols included in a physical layer service data unit (PHY-SDU)<sup>1</sup> is  $N_s$ , only ROIs with a

<sup>1</sup>The physical layer service data unit is generally known as a *frame*. However, in order to avoid any confusions with the picture frames, we denote it as *PHY-SDU* throughout the paper.

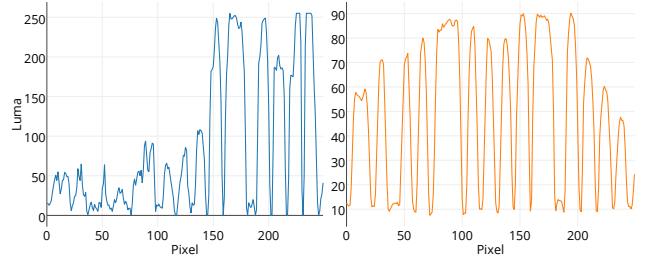


Fig. 3. Comparison of the signal before S3 (left) and after S3 (right).

diameter  $D > S \cdot N_s$  should be considered for decoding, since smaller ROIs can not embed a full PHY-SDU.

In order to detect the beginning of a transmission, a part of the  $N_s$  symbols of a PHY-SDU are used to encode a synchronization sequence known by the receiver, which we denote as SYNC. This results in a SYNC sequence encoded over a set of  $\Delta_{lim}$  pixels. The ROI detection algorithm consists in looking for a SYNC symbol in  $I$  by processing only the relevant pixels in order to keep the process duration as low as possible.

Let us denote by  $I_{max}$  the pixel with the highest luma intensity. The ROI detection algorithm starts scanning a row  $r_x$ , looking for the brightest pixels in  $I$ . However, to speed up the process, only  $1/\Delta_y$  pixels in  $r_x$  are considered, i.e. the red squares in Fig 2, with  $\Delta_y < D/2$  in order not to miss any meaningful ROI. More precisely, the algorithm checks whether  $r_x(y) < \gamma \cdot I_{max}$ , where  $\gamma$  is a predefined decoding threshold. Whenever this condition is true, the algorithm starts scanning the column  $c_y$  for  $\Delta_{lim}$  pixels, looking for a sequence of symbols that matches the predefined SYNC pattern. In the same way as for the row scan, only  $1/\Delta_x$  of the  $\Delta_{lim}$  pixels are considered, where  $\Delta_x < S$  ensures that each symbol is detected. If a SYNC pattern is found, the position is marked as ROI start. We note that a ROI will contain as many SYNC sequences as PHY-SDUs, which allows us to estimate the end of the ROI as well. Finally, by adding a margin  $\Delta_z$  to the detected ROI, we extract the signal  $v_{roi}$ , the red line in Fig. 2, to be further processed in S3.

Indeed, because of the reduced LED luminance, corroborated with the blooming effect and possibly low SNR conditions, the  $v_{roi}$  signal is difficult to quantize directly. An example is depicted on the left side of Fig. 3, where a sequence of falling and rising fronts can be noticed in the signal, but the luma intensity highly varies throughout  $v_{roi}$ .

To equalize the signal and prepare it for decoding, we use an averaging technique with a window  $\Delta_z$ . More precisely, we compute the output of S3,  $v'_{roi}$  on the right side of Fig. 3, as:

$$v'_{roi}(x) = \frac{1}{2\Delta_z} \sum_{i=y-\Delta_z}^{y+\Delta_z} I(x, i)$$

Since  $S$ , the duration of a symbol, slightly varies depending on the CMOS sensor [3], our algorithm then recovers the symbol rate by scaling the SYNC falling and rising fronts in  $v'_{roi}(x)$ . Finally, straightforward steps S4 and S5 end the decoding.



## B. Experimental Results

Fig. 4 summarizes the average execution time of the processing thread functions on the Nexus 5 smartphone, with a Java implementation.

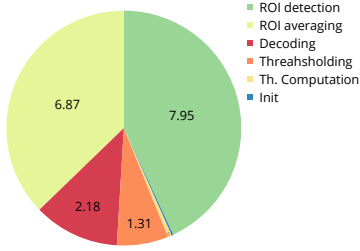


Fig. 4. Steps duration repartition of the optimized algorithm.

Fig. 4 show that our algorithm can recover the data information in 18.4 ms on average, in various illumination conditions. Moreover, when we implement the same algorithm in a C shared library, which is then called from our Android application, the whole processing takes only 3.6 ms. These results outperform previous works, especially when considering that, unlike the state of the art solutions, we compute the LED position in each frame (about 30 times per second), adding robustness against user motion.

We present a thorough evaluation of the LED-to-camera communication in [5].

## IV. ENABLING SMARTPHONE-TO-LED COMMUNICATION

A feed-back channel, from the smartphone to the object, is required if we want to transform the LED from a simple *beacon*, continuously broadcasting the same information, in a real network device, capable of transmitting different types of information in unicast communication. Moreover, this smartphone-to-LED communication might prove useful not only for control purposes, but even for some applications, e.g. modifying the default settings of a device using the user smartphone.

In this section, we present the main design principles of the communication link from the smartphone, more precisely from the flashlight, to the LED. Photo-diodes are often considered as receivers for VLC, as they are the appropriate and most efficient electronic components for light sensing, by converting the incoming light into electrical current. However, to be efficiently exploited, photo-diodes require a fitted piece of hardware, based on a trans-impedance amplifier (TIA) and analogical filters, making this solution not suitable for our system, which aims to be as unobtrusive and cheap as possible. Therefore, in our design, we exploit the fact that a regular LED can also be used as a photo-receptor, by reverse-biasing it. We describe the principles of the flash-to-LED communication in Sec. IV-A, and we evaluate its performance in our implementation in Sec. IV-D.

### A. Communication principles

Although they possess a low light sensitivity, LEDs have already been used as receiver in VLC, as detailed in [2], [10]:

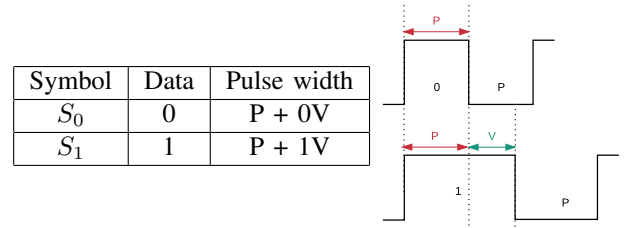


Fig. 5. Symbol data mapping and diagram.

by charging a reversed biased LED, and measuring its electric tension after a short time interval following this charging, the variations in the incoming light intensity can be detected. This allows us to easily detect the on/off periods of an incoming OOK modulated beam of light. However, previous works [2], [10] only investigated LED-to-LED communication; our implementation of smartphone-to-LED communication is therefore a first in the literature.

On the smartphone side, two possibilities exist: using the LCD screen of the smartphone by controlling its back-light intensity [17], or triggering the flash built into most current mobile phones [13]. Since using the screen makes it impossible to use the back camera as a receiver at the same time, we adopt the flashlight as VLC transmitter in the proposed system.

### B. Modulation characteristics

Controlling hardware components, such as an LED generating a signal, is generally achieved by a real time system, such as an MCU or a field-programmable gate array (FPGA), which uses hardware oscillators and timers to achieve time synchronization and turn on or off the light at precise time intervals. However, this precision is not possible for application-based implementations on a mobile operating system, e.g. an application controlling the smartphone flashlight, where these tasks are performed by the central processing unit (CPU). To add robustness against the clock jitter, we propose a modulation scheme based on a variable pulse width modulation (VPWM), where a pulse is a period during which the flash is turned ON. In this scheme, the data bits are mapped on VPWM symbols, as shown in Fig. 5, where each symbol is built using an initial pulse duration  $P$ , and a variable offset of the flash-ON duration,  $V$ .

The two modulation parameters,  $P$  and  $V$ , should be determined based on some properties of the smartphone in order to alleviate the clock jitter problem. First of all, the smartphone must be able to turn on and off the flashlight in a time interval  $P$ . Ideally, the smartphone needs to be able to correctly measure the period  $P$  and the execution of the switch on/off command should take a constant time. The clock jitter is produced by imprecisions on all these points, hence the use of parameter  $V$ , which should be larger than the worst case jitter. Since the optimal values of  $P$  and  $V$  depend on the smartphone, the LED needs to estimate their duration at the beginning of the reception. We cope with the jitter of the flash signal and with heterogeneous clock rates between devices with a smart synchronization and clock recovery method that we detail in Sec. IV-C.

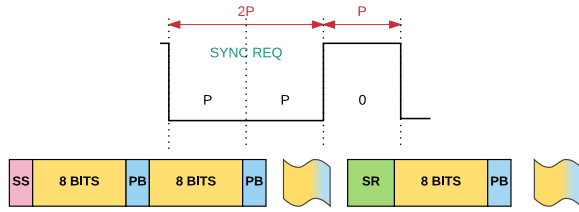


Fig. 6. Synchronization process over a transmission using the SR symbol.

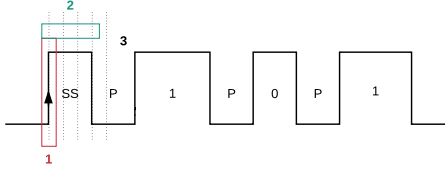


Fig. 7. Receiver synchronization process.

### C. Receiver synchronization

As explained before, the optimal values of the modulation parameters vary depending on the smartphone and the timing inaccuracy of the mobile OS, producing signal pulses of inconsistent width during a transmission. To address this point, the receiver must synchronize with the transmitter and retrieve the signal characteristics,  $P$  and  $V$ , to correctly demodulate the signal. For that purpose, the smartphone emits two kind of synchronization symbols as shown in Fig. 6: 1) a start sequence (SS) at the beginning of each transmission and 2) a SYNC REQ symbol (SR) periodically sent to recalibrate the smartphone symbols table. This is the key point to recover from clock jitter and harden the demodulation.

This mechanism is depicted in Fig. 7: the receiver detects a rising edge (1), that signals the start of a transmission; it then measures the pulse width (2) by sampling the signal until a falling edge is detected. Thus, the receiver recovers the pulse width period  $P$  and it can continue receiving data symbols to retrieve  $V$  on the fly, by measuring the longest next pulse. To compensate for a potential loss of synchronization and recover from a link loss, an SR control symbol is inserted after the parity bit, if the next symbol is  $S_0$ , as shown in Fig. 6. Thus, if a *light off* pulse of duration  $2P$  occurs, the receiver knows

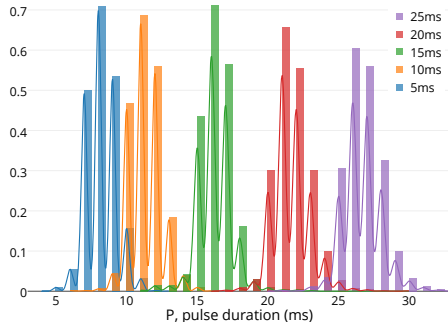


Fig. 8. Distribution of effective pulse duration for different configured pulse duration values.

that the next *light on* pulse represents the beginning of a byte, and it can recalibrate.

### D. Experimental evaluation

To evaluate the flash-to-LED communication system, we implement a smartphone application that randomly generates series of 1000 bytes and transmits them to the LED using the flashlight. We maintain the smartphone in a fixed position, while controlling the angle it makes with the LED axis. On the smartphone side, we record the transmitted symbols and timing information, while on the VLC module we log the LED discharge tension and the received symbols, as computed by the MCU.

We begin by evaluating the smartphone ability to transmit symbols at a fixed frequency, in order to deduce the optimal modulation parameters,  $P$  and  $V$ . Fig. 8 shows the distribution of the effective pulse duration for several configured values, from 5 ms to 25 ms.

We note that the system can not achieve a pulse duration as low as 5 ms; this can be noticed from the blue curve, where the effective pulse duration is around 8 ms. Another important observation is that the distribution of the effective pulse duration is rather wide and the results obtained for two configured values separated by less than 10 ms are overlapping. Based on these results, we consider  $V = 1.5P$  safe to avoid ISI in the worst case scenario and let the receiver synchronization algorithm, described in Sec. IV-C, retrieve the correct clock rate whatever the smartphone.

Next, we evaluate the communication range of the flash-to-LED technology and its potential limitations, especially the ambient light interference and the LOS condition. For this, in Fig. 9, we show the signal-to-noise ratio (SNR), measured at the LED receiver, for different communication distances and angles. Regarding the communication throughput, we notice an on/off behavior: for an SNR above 1 dB, a throughput of 30 bit/s is achieved, regardless the quality of the channel, while the throughput quickly drops to zero for an SNR under 1 dB. This 1 dB SNR threshold is represented by the red line in Fig. 9. We observe in these results a maximum communication range of 110 cm and a very constrained communication angle between the LED and the smartphone, lower than  $15^\circ$ . These results can not be compared with previous works on flashlight usage [7], [9], as these studies use a photo-diode as a receiver, and not a small LED as we do in our work.

### E. Flash-to-LED use case

Although the throughput achieved for the flash-to-LED communication is very low, this channel can still be useful. The first example is the use of flash-to-LED as a wake-up channel. Indeed, the use of a low-energy wake-up side channels has gained a lot of popularity in energy-constrained RF networks [11]. This allows switching off the main radio transceiver, with a significant impact on the overall power consumption.

We notice that this wake-up feature is innate in our system, since the LED does not need to be powered in order to

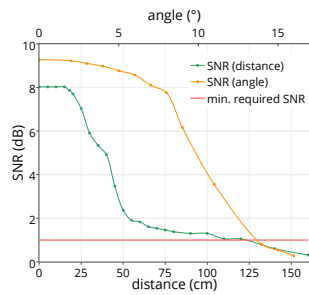


Fig. 9. SNR when varying the flashlight distance (green curve) and the smartphone-LED angle (orange curve). The red line represents the minimum required SNR for correct decoding.

detect an incoming transmission from the smartphone. Coming back to the LED-to-camera communication in Sec. III, this means that the LED does not need to act like an RF beacon, continuously transmitting the information in a loop. Instead, by sending a short light message, the smartphone can wake up the object from stand-by mode and initiate a unicast communication. This mode of functioning has important advantages, not only from an energy point of view, but also for security reasons: the object will transmit the information only when it knows that the smartphone is in its proximity.

A second usage of the flash-to-LED system is its integration in a bidirectional communication technology between a small colored LED and an unmodified smartphone. However, our results indicate that the flash-to-LED average throughput is 30 bit/s, while it can be about 2 kbit/s for the LED-to-camera link. With such an asymmetry between the channels, acknowledging every packet reception on the smartphone side does not seem realistic. Instead, the flash can be efficiently used as a feedback channel, by signaling only the missed packets or just acknowledging the reception of the whole message. Thus, in combination with an erasure code on the emitter side as random linear coding, the flash-to-LED channel would significantly improve the system performance and robustness [6].

## V. CONCLUSION

This paper introduces the key components that are necessary to build a robust bidirectional communication system between an LED and an unmodified smartphone.

We propose an efficient decoding algorithm, which can detect the LED position, process and decode the signal on average in 18.4 ms, for each frame, on a Nexus 5 unrooted smartphone. Thus, this implementation is convenient for low latency indoor localization or real-time transmission with a moving receiver. Also, as the ROI detection is the most complex step of the algorithm, scenarios with several transmitters can be envisaged, enabling MIMO-like transmissions.

We also present smart mechanisms and protocols to build a robust flash-to-LED communications channel using off-the-shelf smartphones and small LEDs. Our experimental evaluation shows a throughput of 30 bit/s, which is suitable for feedback, wake-up or even some limited communication purposes.

We believe that such bidirectional VLC communication system will be a great opportunity for smart and connected consumer electronic products, providing bidirectional smartphone-to-device communication at lower cost.

## REFERENCES

- [1] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas. Using a CMOS camera sensor for visible light communication. In *2012 IEEE Globecom Workshops*, pages 1244–1248. IEEE, 2012.
- [2] P. Dietz, W. Yezazunis, and D. Leigh. Very Low-Cost Sensing and Communication Using Bidirectional LEDs. In *UbiComp 2003: Ubiquitous Computing*, pages 175–191. Springer Berlin Heidelberg, 2003.
- [3] T.-h. Do and M. Too. Analysis on Visible Light Communication using Rolling Shutter CMOS Sensor. In *International Conference on Information and Communication Technology Convergence (ICTC)*, number 1, pages 755–757. IEEE, 2015.
- [4] A. Duque and H. Rivano. Demo : Off-the-shelf Bi-directional Visible Light Communication Module for IoT Devices and Smartphones. *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, pages 238–239, 2017.
- [5] A. Duque, R. Stanica, H. Rivano, and A. Desportes. Unleashing the power of LED-to-camera communications for IoT devices. In *Proceedings of the 3rd Workshop on Visible Light Communication Systems - VLCS '16*, pages 55–60. ACM Press, 2016.
- [6] A. Duque, R. Stanica, H. Rivano, and A. Desportes. SeedLight. In *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems - VLCS '17*, pages 15–20. ACM Press, 2017.
- [7] L. Fan, Q. Liu, C. Jiang, H. Xu, J. Hu, D. Luo, Z. He, and Q. Huang. Visible light communication using the flash light LED of the smart phone as a light source and its application in the access control system. In *2016 IEEE MTT-S International Wireless Symposium (IWS)*, pages 1–4. IEEE, 2016.
- [8] J. Ferrandiz-Lahuerta, D. Camps-Mur, and J. Paradells-Aspas. A Reliable Asynchronous Protocol for VLC Communications Based on the Rolling Shutter Effect. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.
- [9] M. M. Galal, H. A. Fayed, A. A. E. Aziz, and M. H. Aly. Smartphones for Payments and Withdrawals Utilizing Embedded LED Flashlight for High Speed Data Transmission. In *2013 Fifth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 63–66. IEEE, 2013.
- [10] D. Giustiniano, N. O. Tippenhauer, and S. Mangold. Low-complexity Visible Light Networking with LED-to-LED communication. In *2012 IFIP Wireless Days*, pages 1–8. IEEE, 2012.
- [11] L. Gu and J. A. Stankovic. Radio-Triggered Wake-Up for Wireless Sensor Networks. *Real Time Systems Journal*, 29(2):157–182, 2005.
- [12] J. Hao, Y. Yang, and J. Luo. CeilingCast: Energy efficient and location-bound broadcast through LED-camera communication. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
- [13] T. Hesselmann, N. Henze, and S. Boll. FlashLight - Optical Communication Between Mobile Phones and Interactive Tabletops. pages 135–138. ACM Press, 2010.
- [14] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose. In *Proceedings of the 20th annual international conference on Mobile computing and networking - MobiCom '14*, pages 447–458. ACM Press, 2014.
- [15] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, and K. C.-J. Lin. RollingLight. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '15*, pages 167–180. ACM Press, 2015.
- [16] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, number 21, pages 249–260. IEEE, 2014.
- [17] A. D. Wilson and R. Sarin. BlueTable: Connecting Wireless Mobile Devices on Interactive Surfaces Using Vision-Based Handshaking. pages 119–125. ACM Press, 2007.