From Sound to Sight: Using Audio Processing to enable Visible Light Communication

Stefan Schmid*†, Daniel Schwyn†, Kaan Akşit*, Giorgio Corbellini*, Thomas R. Gross†, Stefan Mangold*

*Disney Research
8006 Zurich, Switzerland

†Dep. of Computer Science
ETH Zurich, Switzerland

***Abstract*—Mobile phones can use their cameras and flashlight Light Emitting Diodes (LEDs) to exchange messages with low-complex Visible Light Communication (VLC) networks, but these interfaces impose serious restrictions when used in a VLC network. In this paper we discuss how to extend mobile phones or tablets with a small peripheral device that is battery-free, uses only passive components, and offers VLC capabilities at the required data rate (kilobit per second). This device plugs into the audio jack; on-board audio signal processing generates the outgoing light signals as well as decodes the incoming light signals. The device is powered from the phone's audio jack output signal, no additional battery is required. The audio signals directly modulate light emissions of an LED. Incoming light is detected by a photodiode and the generated electrical signals are fed into the microphone input. This simple device enables use of existing communication protocols and therefore makes it possible to integrate mobile phones or tablets into existing VLC LED-to-LED networks.**

I. Introduction

Visible Light Communication (VLC) with Light Emitting Diodes (LEDs) as transceivers enables short range low-bandwidth networking of consumer devices like toys, smartphones, or tablets. We refer to this use as LED-to-LED VLC networking [1], [2]. Devices use LEDs to emit light and transmit data encoded into on-off patterns. The encoding creates a slotted light pattern that is too fast to be perceived by human users (no flickering, typically at 1 kHz or more). During the off periods, the LEDs can be used as photodiodes to sense incoming light, and as a result the LEDs receive data symbols from other LEDs. Earlier papers described the software-based synchronization, the data encoding with flicker avoidance, and the required communication protocols [2].

In this paper we report on the design, implementation and evaluation of a miniature low-cost passive device that can be plugged into an audio jack connection of a mobile phone to enable two-way VLC communication. Figure 1 illustrates how a mobile phone exchanges data with a toy. The device uses an LED and a photodiode to transmit resp. receive signals. The use of a photodiode instead of a receiving LED is discussed in Section II-B. When connecting such a device with a phone's headset audio jack, the phone can exchange data through light at a data rate in the order of a kilobit per second in both directions. The miniature VLC communication device uses the audio output of the phone to drive an LED (transmission), and the microphone input to receive from a photodiode (reception).

The audio jack device is battery-free and operates without the involvement of a microcontroller. The audio signals needed to modulate the LED are generated by software running on the phone. The light sensed by the photodiode is also decoded on the phone. The application software running on the phone can generate data packets in real-time; this capability makes it possible to run dynamic communication protocols directly on the phone and enable transmission and reception of data with a peripheral device that uses the LED-to-LED VLC networking protocols referred to above.

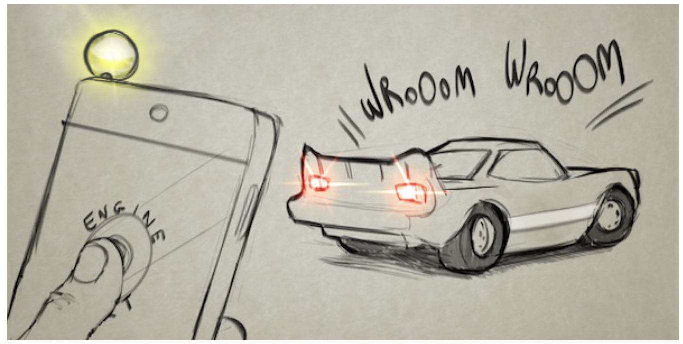

Fig. 1. (© Disney) VLC enabled toy remotely controlled by a smartphone extension device plugged into the audio jack.

A. Contributions

In this paper we report the following contributions:

- Hardware design of a VLC peripheral extension device for smartphones using the audio jack as interface. The device is battery-free and only powered and operated through audio signals generated by the mobile phone. Further, the device is equipped with a photodiode to feed incoming modulated light as electrical signals into the microphone input (Section II-B).
- Smartphone application software operating the audio jack peripheral device through audio signal processing only; there is no need for an additional microcontroller. Microphone input data is analyzed and decoded in software and arbitrary data packets can be generated in real-time using the peripheral's LED as a communication front-end (Sections II-C and II-A).
- Evaluation of the designed hardware together with application software running on iOS devices; results for different host devices are reported (Section III).

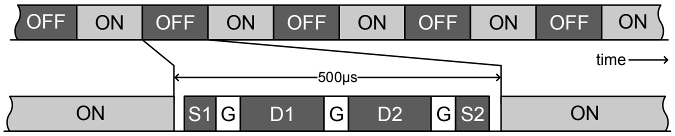

Fig. 2. Slotting structure of the LED-to-LED physical layer network protocol. ON and OFF slots are alternating to provide illumination and communication at the same time.

B. Related Work

Application scenarios and use cases for VLC-enabled consumer devices have been demonstrated on several occasions [3]–[5]. Smartphones communicating in LED-to-LED networks without the aid of additional devices (using built-in camera and flash light) have also been investigated [6], but the placement of these building blocks may limit their use in some scenarios. In this paper we extend the smartphone with a peripheral device to increase flexibility, performance, and stability for links between mobile devices and VLC-enabled consumer devices. Extending smartphones with light sensors through the audio jack interface has also been reported [7]. This system, however, allows only a one-way communication (by light) whereas the system described here enables a full-duplex two-way communication. Using a smartphone's audio jack as communication interface and hardware source has been extensively investigated by various projects at the University of Michigan [8], [9] which provided the foundation for commercial hardware developments [10]. The hardware system presented in this paper is based in part on this approach.

II. System

This paper focuses on how to extend smartphones and tablets with VLC functionality. Although these devices are already equipped with a flash light (light emitter) and camera (light receiver), which can be used for communication [6], [11], they do not provide enough flexibility to work well together with other VLC-enabled devices. Smartphone operating systems cannot support real-time scheduling, and control over the flash light and camera is often restricted. These constraints limit performance and stability of a VLC link. The system described in the following sections is based on a passive peripheral device that plugs into a smartphone's audio interface and can emit and receive light by using the phone's audio system. The peripheral device is battery-free and powered only through the audio signals, yet the communication protocols are handled without additional micro- or signal- processors – light is directly modulated through audio signals generated in real-time by the smartphone, and incoming signals are converted by the microphone and analyzed by application software. The system described can interact with existing VLC-enabled toys or other consumer devices that implement the VLC protocols developed for LED-to-LED networks, which are summarized in the following section.

A. LED-to-LED communication protocols

The protocols consist of a physical (PHY) layer and Medium Access Control (MAC) layer [2] based on a software-only implementation for microcontrollers and uses only LEDs as sender and receiver. The PHY layer is not only responsible for communication but also for illumination. Since an LED should always appear (to a human) as switched on, a slotting structure, illustrated in Figure 2, is introduced. ON and OFF slots are alternating with a period of 500 µs. During an ON slot light is emitted so that the LED appears as always on. The OFF slots are used to sense incoming light by reverse- biasing the LED [12]. These slots are further divided into smaller intervals (as shown in Figure 2) with the following functions: The synchronization slots (S1,S2) in the beginning and at the end are responsible for synchronization to the ON and OFF slots of a communication partner. The guard intervals (G1,G2,G3) prevent light leakage into the data intervals (D1,D2) in case of small phase offsets. A zero symbol is encoded as light emission during D1, and a one symbol is represented by light emission during D2. The MAC protocol is based on CSMA/CA as applied in the IEEE 802.11 standards and enables contention based medium access for a network of VLC devices.

B. Peripheral Device Hardware

Audio signals are AC-coupled, hence it is not possible to directly generate an on-off pattern to drive an LED. Further, even with the loudest audio settings, the amplitude of the audio output signal is still in the millivolts range (around 100 mV to 200 mV, depending on the device) and therefore not large enough to emit light with reasonable intensity through a standard LED. Our device uses a hardware design that is based on University of Michigan's Hijack project [8], [9]. The schematics of the peripheral device's Printed Circuit Board (PCB) are shown in Figure 3, and the electronic parts are listed in Table I. The schematics show a low-complex system with only a handful of components, without the need to include a microcontroller. The audio signals of the left and right channel are joined together to increase the available current and therefore also power. This can be seen as two batteries in parallel but phase-synchronized AC coupled. The signals amplitude is increased by the coupled inductors (T, 1:25 turns ratio). These inductors are a passive component that increases the voltage at the same electric power and therefore also decreases the current. The transformed signal's amplitude provides high enough voltage to drive an LED. Already with this raw signal (e.g., shown in Figure 6) the LED emits light, albeit with low intensity. To increase the light emission further, the signal is rectified (Q1-Q4). This step makes it possible to also use the negative parts of the sine waves to emit light instead of reverse biasing the LED. The rectifier is built using MOSFETs instead of diodes, since diode-based rectifiers suffer from power loss (voltage loss of about 0.7 V per diode in path). Finally, a capacitor is used to smooth and stabilize the LED's input voltage.

Instead off using exclusively LEDs (i.e., the LED is also employed for reception as described earlier) the device uses a photodiode to convert modulated light back to electrical signals. Using the same LED to send and receive is not trivial for this setup. There is a bias voltage of 2 V applied to the microphone input and therefore, to switch between emitting and receiving light, the LED needs to be attached and detached from and to the microphone line. Such an arrangement introduces a difficult problem (if to be solved without microcontroller). As we aim to keep the hardware as simple as possible, we focus on how to modulate the light with the help of audio

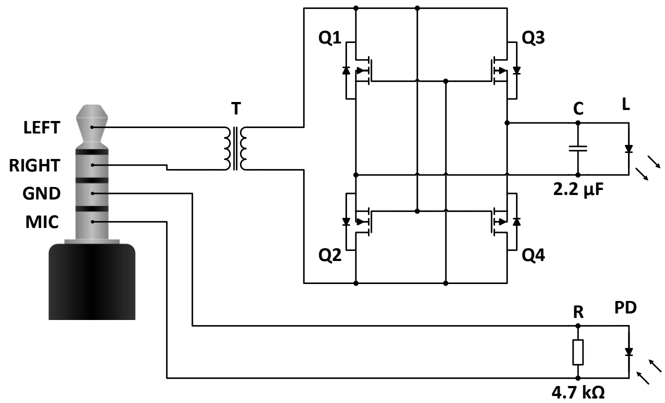

Fig. 3. Circuit diagram of the implemented peripheral device using only a handful electronic components, without microcontroller and the help of additional power sources.

TABLE I. ELECTRONIC PARTS KEY

Symbol	Name
T	Coupled inductors, 1:25
Q1,Q2	MOSFET, n-channel
Q3,Q4	MOSFET, p-channel
L	LED
PD	Photodiode

signals and to allow bidirectional communication with a low-complex circuit. The peripheral device uses the microphone as analog-to-digital converter (ADC) to measure the voltage over the resistor (R). Incident light generates a photocurrent in the photodiode (PD). The current is proportional to the light intensity and can be measured as voltage drop over R. Figure 4 shows the assembled PCB with and without casing. The device is still small, fits on a board of 1.7 by 2.7 cm, and requires no battery; it can be built with only a handful of inexpensive electronic parts.

C. Smartphone Software

The software part of the VLC system implemented for iOS is responsible for generating the waveforms needed to modulate the peripheral device's LED to conform with the protocol described in Section II-A. It also analyzes the incoming signal from the photodiode and decodes it. The structure of the system is depicted in Figure 5. There are three principal modules: the main module, the sender module, and the receiver module. Because sender and receiver use the same interface to the hardware, they are partly implemented in a common transceiver module. Data passed from the application to the VLC main module is encapsulated in VLC frames and added to a message queue. From there the frames are passed to the sender in FIFO order. Packets that must be acknowledged are retransmitted if a timeout occurs. After a packet is acknowledged or the maximum retransmission count is reached, the packet leaves the sender buffer and the application is notified via a callback. Incoming messages are processed by a decoding pipeline and finally delivered to the main module, which sends an acknowledgement if necessary and delivers the data to the application.

1) Sender: The core of the sender is a callback function that is invoked by the hardware whenever it needs to output

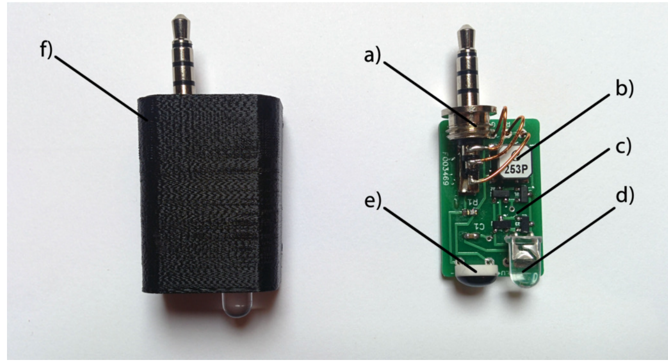

Fig. 4. Custom designed PCB: a) 3-channel audio plug, b) coupled inductors (T), c) rectifier (Q1-Q4), d) standard LED, e) photodiode, f) 3D-printed casing

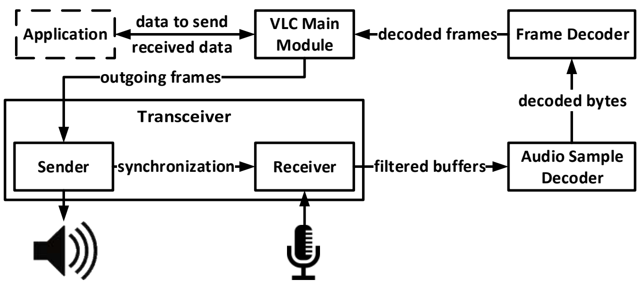

Fig. 5. Software system diagram. Received signals are decoded by the Audio Sample decoder and forwarded to the Frame Decoder where the payload is identified. The VLC main module is in charge of assembling new data frames that are handed over to the Sender module that generates the audio signals to modulate the LED.

sound buffers. To ensure that the callback function returns before a buffer underrun occurs, the templates for the three types of data slots that occur in the VLC protocol ('idle', 'bit 0' and 'bit 1') are pre-built at start-up and only copied into the target buffers. A template for a data slot always includes the following ON slot. This is necessary as after transmission slots ('bit 1' or 'bit 0') the LED needs to be switched off for a short time in the next on slot for the brightness to appear constant. iOS devices have a built-in sound processor that smooths quick on-off patterns. It was found that with a sample rate of 48 kHz and a signal frequency of 10 kHz this effect could be reduced to an acceptable level while still delivering enough power to the LED. Examples for generated audio signals based on the calculated wave forms are shown in Figures 6 and 7.

2) Receiver: The receiver is also callback-based. Whenever the hardware has input buffers ready it invokes a function that pre-processes these buffers and copies them to user memory. The buffers are then passed to the decoding pipeline. The first stage of the pipeline is the physical decoding stage. It compares the audio frames to a threshold with alternating sign. A value below the negative threshold value is considered to originate from incoming light, while a value above the positive threshold means there was no light detected. A change between light and no light is called a flip. The decoder then calculates the number of frames between the flips. These run lengths correspond to the on-off pattern of the sending LED. By analyzing these patterns, individual bits can be decoded. The decoded bits are accumulated to bytes and passed to the next stage. Due

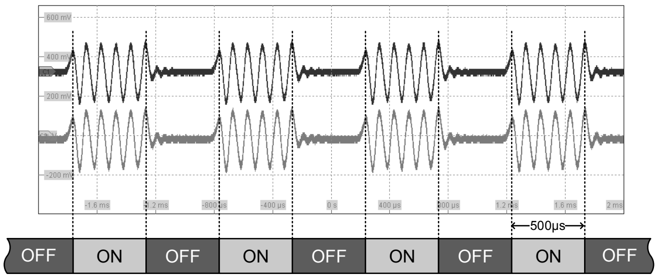

Fig. 6. Stereo audio signal generated by the smartphone's audio system to produce an ON-OFF emission pattern with the LED.

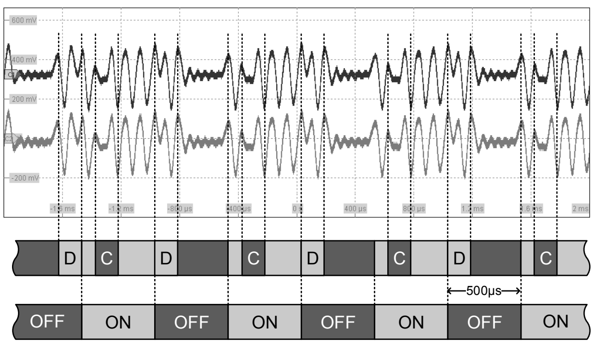

Fig. 7. Stereo audio signal during the transmission of a data packet. The signal is switched on earlier or stays on longer to provide light for a data slot at the beginning or in the end of an OFF slot (D). To compensate the additional light output to avoid flickering, the audio signal is switched OFF during the following ON slot (C). To ease reading the figure, an ON-OFF legend is shown.

TABLE II. DECODING PATTERNS

Interval	# frames	idle	bit 0	bit 1
GUARD1	5	OFF	OFF	OFF
DATA1	6	OFF	ON	OFF
GUARD2	2	OFF	OFF	OFF
DATA2	6	OFF	OFF	ON
GUARD3	5	OFF	OFF	OFF

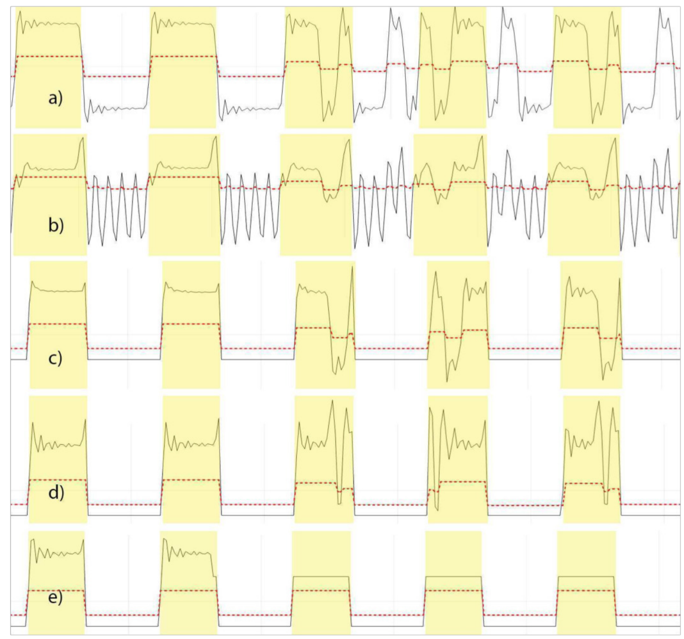

Fig. 8. Signals as received by the Audio Sample Decoder. The dashed (red) line indicates the run length pattern found. a) Receiving with sender off; b) Receiving with sender on; c) Receiving with ON slots filtered; d) Sending – only ON slots filtered; e) Sending – ON and OFF slots filtered.

to a high level of noise in the signal (see Section II-C3), short intervals cannot be reliably detected with a resolution of only 24 frames per slot (due to the maximum sample rate of iOS devices at 48 kHz). Thus the VLC decoding scheme from Figure 2 needs to be simplified in the following way (see also Table II): If there is no flip during the OFF slot, no bit is detected. If there is a flip, to distinguish between bits, the decoder detects in which half of the OFF slot a run length above the positive threshold exists for about 13 frames and assumes the spike to encode the symbol appears in the other half. Furthermore, the inability to detect short intervals prevents the system from being able to synchronize to another VLC device (as it cannot detect the synchronization intervals illustrated in Figure 2). As long as only one smartphone is in a network, this limitation does not pose a problem as the other VLC devices can synchronize to the smartphone's on-off pattern. The second pipeline stage decodes the protocol frames. The decoding process works the same as for LED-to-LED networks [2]: After a Start Frame Delimiter (SFD) is detected, the headers are decoded and the CRC is calculated. The decoded frames are then passed to the main module.

3) Signal feedback: Because of the simple circuitry that is included in the peripheral device, the device suffers from signal leaks into the receiver. This poses two problems: First, there is a 10 kHz feedback signal during ON slots. It was found that this makes it impossible to reliably detect the end of ON slots as the wavelength of this signal is inside the tolerance interval for GUARD1. Second, if the sender is active, the spikes in the data slots that encode the symbols are also fed back to the receiver. This also leads to the decoder losing synchronization with the slot pattern. To prevent both of the problems the buffers are preprocessed directly in the hardware callback: By keeping track of the OFF slot sample times in the sender callback, the corresponding incoming buffers are set to a value below the negative threshold to mark them as ON. Similarly the buffers corresponding to data slots where the sender is active are set to a value above the positive threshold to mark them as OFF. As OFF slots do not carry information and an incoming message in the same slots as an outgoing would lead to a collision anyway we lose no information this way. Figure 8 shows the signals received by the Audio Sample Decoder.

4) Multithreading: To enable asynchronism and increase performance, the system uses multi-threading. With Grand Central Dispatch (GCD) iOS offers an easy to use framework: Blocks of code can be dispatched to serial queues (serial means the queue operates in a FIFO manner). The system then takes care of the mapping of these queues to an suitable number of worker threads. In the main module there are two serial queues, one that handles incoming messages and one that dispatches packets from the message queue to the sender. Furthermore each stage of the decoding pipeline runs on its own serial queue.

III. Evaluation

The evaluation is conducted with a testbed consisting of an iPhone 5S and an iPad mini (both equipped with Apple's A7 processor, running iOS 7) and a VLC device running the protocols (on an Atmel microcontroller [13]) described in Section II-A that uses only an LED as transceiver. The audio jack peripheral device is platform-independent. Other smartphones or tablets, even laptops or desktop computers (independent of operating systems), could be used for this evaluation as long as they provide a 3-channel audio jack plug with a matching pinout. We focus on Apple's iOS devices because of the well documented audio API. All experiments are conducted in a standard office space without special shielding from artificial light and sunlight.

A. ACK timeout

The VLC device and smartphone both run a MAC layer capable for data frame acknowledgments and retransmissions. After transmitting a data frame, the transmitter waits for the ACK timeout. If no ACK from the data frame's destination is received within this interval, the frame counts as lost and the transmitter retransmits. This procedure is repeated until an ACK reaches the sender or a fixed number of retransmissions has happened. The firmware running on the VLC device's microcontroller is real-time and the ACK timeout can be kept small since a receiving device is fast in processing a received data frame and transmitting an ACK (within few milliseconds). A short ACK timeout guarantees higher throughput performance for a network consisting of only a few devices, as the communication channel can be used more efficiently.

A smartphone operating system is not a system with real-time guarantees and the main processor is used for several different tasks at the same time. Also, it may take some time to analyze incoming data from the audio jack and decode the data packet. Further the audio signals needed to transmit an ACK are generated on demand specifically for the received data frame; this step takes additional time. Therefore the ACK timeout of the VLC device must be adjusted to enable a successful and optimized data exchange with a smartphone using the audio jack extension device.

To find a proper value for the ACK timeout the following experiment is conducted: the VLC device is generating data packets (saturation) with the smartphone as destination. The smartphone needs to send an ACK back to the transmitter. If the ACK does not arrive on time, the data frame is retransmitted, resulting in a drop in throughput. To find an optimal value, the timeout is increased step by step. The same experiment is also repeated for different packet sizes so see if the processing time (on the smartphone) has any impact on the delay. The results are shown in Figure 9. For timeouts of 45 ms to 50 ms throughput is stable but not close to the theoretically reachable maximum, meaning that the ACK arrives too late and packets are always retransmitted. Between 50 ms and 70 ms throughput is increasing slowly, but the plot also shows higher error bars, leading to the conclusion that the ACK reaches the destination sometime within the time windows and with higher probability towards higher timeouts. For 71 ms and longer, the error bars are disappearing again and throughput stays stable. Measurements for higher timeouts are omitted since the throughput does not increase anymore. Also, the packet length and therefore the decoding time on the smartphone seems not to have any impact. In summary, the measurements show that a delay of around 70 ms is optimal to maximize throughput for a single communication link.

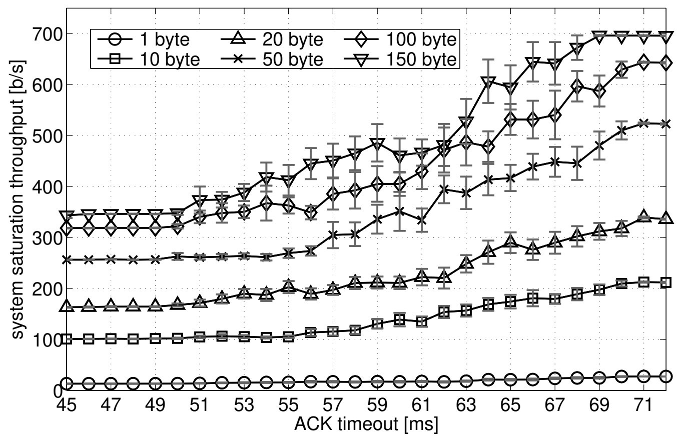

Fig. 9. Measurement results for a communication channel from a VLC device to iPhone 5S. With an ACK timeout of 70 ms no packets are retransmitted due to slow ACKs and throughput is about maximized.

B. Distance measurements

To be useful for some use cases, the peripheral device must be able to cover a certain communication distance and achieve a stable and reasonable data throughput. Figures 10 and 11 show throughput measurements for data payload only, for and iPhone respectively an iPad, for different packet sizes and distances. Here, the VLC device acts again as saturation packet generator. For the iPhone, the throughput stays stable up to 25 cm at a maximum of 700 bit/s for a packet size of 150 byte. With the MAC protocols retransmission scheme in place, it is also possible to achieve reliable communication up to a distance of 35 cm, but with losses in throughput. The iPad measurements show an increased communication range. This is most probably due to a stronger audio amplifier included in the iPad which increases the intensity of the light emissions. With an iPad it is possible to achieve stable throughput in the same order as for the iPhone at a communication distance of up to 50 cm. These measurements show that the mobile device's audio system has also an impact on the possible communication ranges and additional measurements with smartphone and tablets of different brands should also be considered. In conclusion, the possible communication distance and the achieved data throughput are sufficient for the indented use cases (e.g., remotely controlled toy car).

C. Power consumption

Highest light emissions are achieved by using the loudest audio output settings. This setup puts additional stress on the device's battery. Also, the computational power needed to decode and create data packets cannot be neglected (although checking the system monitor during measurements always shows CPU utilization below 10 percent). Figure 12 shows the battery level for both iPhone and iPad over time while transmitting and receiving. The measurements show that the

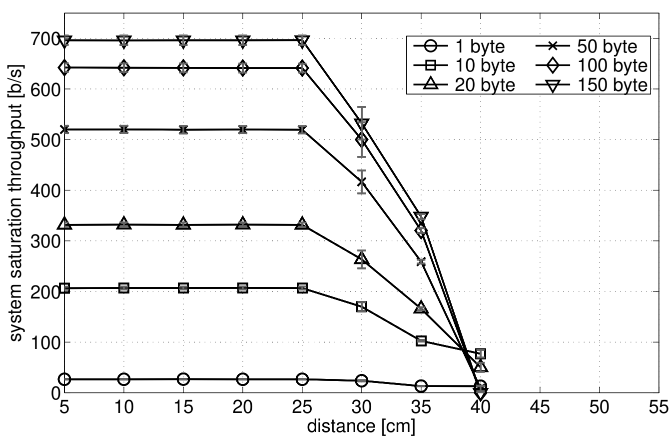

Fig. 10. Measurement results for a communication channel from a VLC device to iPhone 5S. Larger packet sizes result in higher throughput due to smaller protocol overhead. The system operates reliably up to 25 cm.

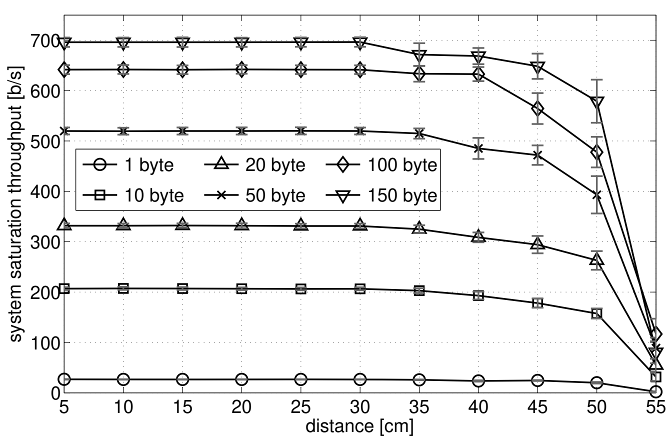

Fig. 11. Measurement results for a communication channel from a VLC device to iPad mini. The useful distance is increased by 20 cm compared to the iPhone 5S.

battery lifetime is more than four hours for the iPhone, and more than five hours for the iPad (due to higher battery capacity). If we assume that the audio jack peripheral device is not used more than 10% of the overall usage time, then the peripheral device does not impact battery lifetime significantly.

IV. CONCLUSION

This paper reports on the design, implementation and evaluation of a smartphone VLC extension device. It uses the phone's audio jack as interface and is operated by audio signal processing. The key design constraints were low-complexity, low-cost, battery-free operation, and interoperability with existing VLC systems. A simple and passive plug-in device is presented based on only a handful of electronic components powered buy audio signals. Its LED is modulated without the help of an additional microcontroller, directly by audio signals generated in real-time by the application. Evaluation results demonstrate that the VLC protocols implemented in software on the smartphone or tablet provide stable and reliable communication over distances from 25 cm to 50 cm depending on the device used (and its audio system). These results show

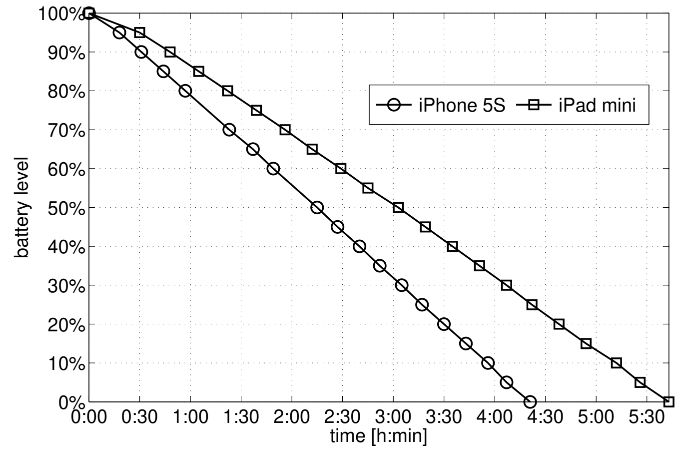

Fig. 12. Battery lifetime for iPhone and iPad mini. Sender idle, screen turned on. Battery life does not pose a problem for common use cases.

that smartphones or tablet can now be integrated into existing VLC networks by the addition of this small passive component.

REFERENCES

[1] S. Schmid, G. Corbellini, S. Mangold, and T. Gross, "An LED-to-LED Visible Light Communication System with Software-based Synchronization," in *Optical Wireless Communication. Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec. 2012, pp. 1264–1268.

[2] ——, "LED-to-LED Visible Light Communication Networks," in *MobiHoc, 2013 ACM*, Aug. 2013.

[3] S. Schmid, M. Gorlatova, D. Giustiniano, V. Vukadinovic, and S. Mangold, "Networking Smart Toys with Wireless ToyBridge and ToyTalk," in *Poster Session, Infocom*, 2011.

[4] G. Corbellini, S. Schmid, S. Mangold, T. R. Gross, and A. Mkrtchyan, "LED-to-LED Visible Light Communication for Mobile Applications," in *Demo at ACM SIGGRAPH Mobile 2012*, Aug. 2012.

[5] S. Schmid, J. Ziegler, T. R. Gross, M. Hitz, A. Psarra, G. Corbellini, and S. Mangold, "(In)visible light communication: Combining illumination and communication," in *ACM SIGGRAPH 2014 Emerging Technologies*, ser. SIGGRAPH '14, 2014.

[6] G. Corbellini, K. Akşit, S. Schmid, S. Mangold, and T. R. Gross, "Connecting networks of toys and smartphones with visible light communication," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 72–78, 2014.

[7] Luicom. (2014) Luciom, presentation at International CES in Las Vegas. 10-July-2014. [Online]. Available: http://www.luciom.com/fr/actus/ces-las-vegas-nevada-us-7-10-janvier-2014

[8] Michigan University. (2014) Project Hijack. 10-July-2014. [Online]. Available: http://web.eecs.umich.edu/~prabal/projects/hijack/

[9] Y.-S. Kuo, S. Verma, T. Schmid, and P. Dutta, "Hijacking power and bandwidth from the mobile phone's audio interface," in *Proceedings of the First ACM Symposium on Computing for Development*, ser. ACM DEV '10. New York, NY, USA: ACM, 2010, pp. 24:1–24:10. [Online]. Available: http://doi.acm.org/10.1145/1926180.1926210

[10] NXP Semiconductors. (2014) Smartphone Quick-Jack. 10-July-2014. [Online]. Available: http://www.lpcware.com/content/project/smartphone-quick-jack-solution

[11] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas, "Using a cmos camera sensor for visible light communication," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec 2012, pp. 1244–1248.

[12] P. Dietz, W. Yerazunis, and D. Leigh, "Very Low-Cost Sensing and Communication Using Bidirectional LEDs," in *UbiComp 2003: Ubiquitous Computing*, vol. 2864, 2003, Book Section, pp. 175–191. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39653-6_14

[13] Atmel, "8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash," www.atmel.com, 2012.