Stereo Audio Streaming via Visible Light

Dulanja Samudika, Lahiru Jayasinghe, Kasun E. Gunathilaka, Y. Rumesh, Ruwan Weerasuriya, Dileeka Dias
Department of Electronic & Telecommunication Engineering
University of Moratuwa
Sri Lanka
dulanja.samudika@gmail.com, lahiruaruna@gmail.com, kegunathilaka@gmail.com, yasintharumeshcb@gmail.com, ruwan@ent.mrt.ac.lk, dileeka@ent.mrt.ac.lk

***Abstract*—. Visible Light Communication (VLC) is emerging as a next generation data transmission method for short-range communication applications. In this paper we implement and characterize two prototype stereo audio streaming methods utilizing VLC. A software architecture is developed to process and stream data. The software architecture is bridged with a hardware section, which facilitates free-space VLC channel, over a Universal Serial Bus (USB) to serial interface. A particularly attractive feature of our system is that it uses commonly available, low-cost components which enables its implementation in everyday applications.**

Keywords—visible light communication, audio streaming, , on-off keying

I. INTRODUCTION

Wi-Fi and Bluetooth are two prominent short range wireless technologies used by a variety of wireless applications today. However these methods utilize scarce radio frequency spectrum. Other drawbacks of these methods inlcude requirement of special equipment, high power consumption and high cost. Visible Light Communication (VLC) is emerging as an alternative next generation wireless communication method. VLC utilizes a light source as the transmitter and a light detector as the receiver to exchange data. Data is encoded using on-off keying (OOK). This is generally done at a frequency greater than 50Hz to make the flickering of the transmitter imperceptible to the human eye.

VLC provides the foundation to many interesting applications and does not utilize already exhausted radio spectrum. It can be used as an alternative in places where radio communication is prohibitiv or undesirable. e.g., airplanes, hospitals, etc. [1] Another interesting application is home entertainment systems using existing house lighting. Integration of VLC to mobile phones is a research area of high interest. We mayexpect VLC compatibility as a mobile phone feature in the near future [2]. VLC is already proven to be capable of achieving higher speeds than Wi-Fi in laboratory environments [3].

Today most VLC research is being carried out using specialized, expensive transmitters and receivers such as laser diodes [4] and avalanche photodiodes [5].

In this paper, the implementation and analysis of two stereo audio transmitting methods referred to as *Pulse Code Modulation streaming* (PCM streaming) and *MP3 streaming* over VLC are presented. Audio streaming is a core component in home entertainment.

The communication architecture implemented is composed of software and hardware subsections. Hardware components are implemented using off the shelf, low cost components Light Emitting Diodes (LEDs) as the transmitter, photodiodes as the receiver [6] and a universal serial bus (USB) module as the interface between the hardware and software subsystems. The software subsystem produces a transmission-ready binary data stream from audio files and the hardware subsection transmits the binary stream over a free space VLC link [7][8].

PCM audio files, the most common type among Resource Interchange File Format (RIFF) audio file types, contain Windows Wave file (WAV) header in conjunction with stereo PCM data [9]. In PCM streaming, 8 bit samples from an audio file are extracted and transmitted over the VLC link.

Generally, media files contain different data streams (e.g. video and audio) [10]. In the MP3 streaming method, MP3 audio stream identification and streaming over a VLC link has being facilitated by the FFmpeg library [11]. The SDL2 library facilitates buffered audio playback in the client application [12].

When implementing VLC audio applications, it is important to choose the best-suited streaming method according to the application requirement. In this paper, the performance of the two streaming methods is compared and results are presented.

II. SYSTEM ARCHITECTURE

In this section, we describe the system architecture of our prototype system for streaming audio over visible light communication. As previously mentioned, the architecture is composed of a software subsystem and a hardware subsystem. Two alternative software subsystems for PCM streaming and MP3 streaming have been implemented. The hardware architecture for both types of streaming remained common.

978-1-5090-0645-8/16/$31.00 ©2016 IEEE

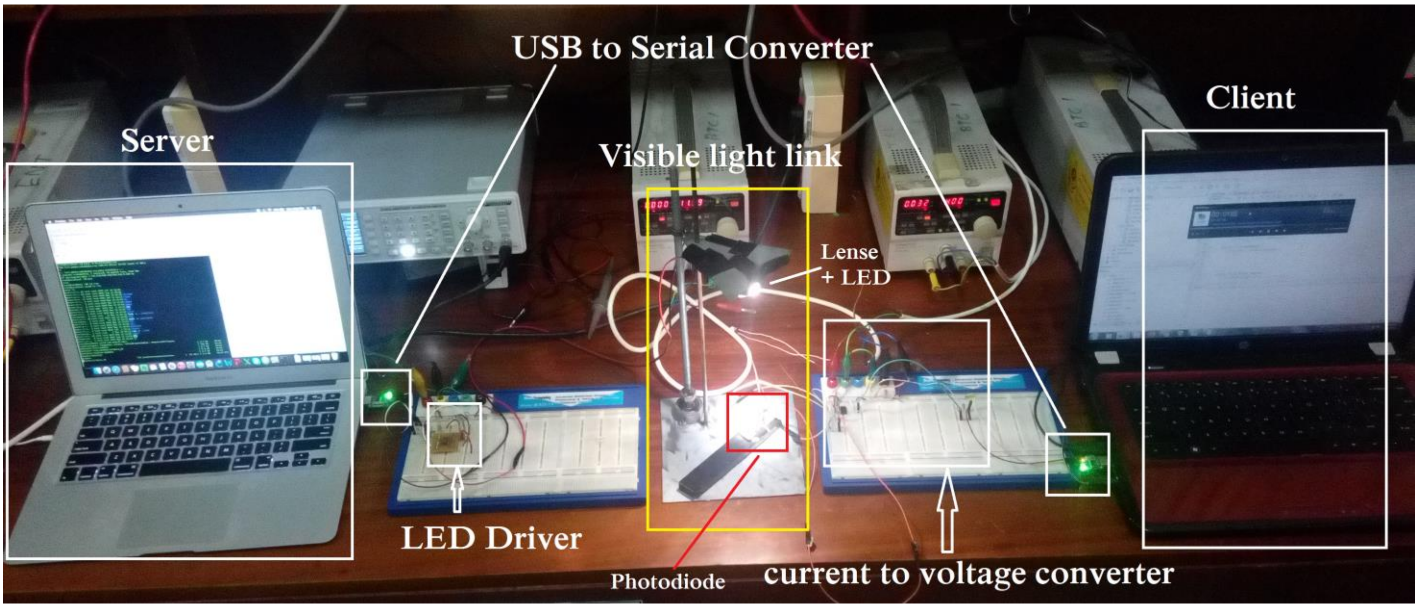

Fig. 1. Implemented VLC system

A. Software subsystem architecture

In order to keep the complexity of the VLC communication to a minimum, the system uses a simplex communication channel. Thus, the software subsystem must be robust. The following sections will describe the software subsystems in detail, while highlighting the critical characteristics of each.

A. 1. PCM streaming software architecture

The basic software architecture for PCM streaming is illustrated in Fig. 2. PCM software streaming application inherently (without any source code modification) supports PCM audio with 8-bit sample, with a sampling rate of 22050 sample per second.

The server instance reads a .wav file consisting stereo PCM data. Since the .wav file contains left and right channel PCM data sequentially, 8-bit PCM values of each channel are read at once and then written to the serial communication port. The USB to serial conversion module, which receives the data being written to the USB port, convey data to the client through the serial port and over the VLC channel.

PCM parameters (PCM sample size, sampling frequency and number of channels) are hardcoded in the client application. This is in order to avoid transmission of these parameters over an error-prone VLC channel, and any resulting mismatch of parameters on the two sides.

The client software configures its audio playback according to the hardcoded PCM parameter upon receiving PCM frames over serial to USB interface. The stereo audio buffer is queued with the received PCM samples. Concurrently, the client thread instance accesses the audio buffer and extracts queued PCM audio samples for playback.

Since we have used 8-bit PCM samples and configured serial interface with 8 -bits per sample, we can ensure accurate error detection of a PCM sample. Erroneous samples are dropped in the client side hardware itself. Error detection in PCM samples is important to achieve robust audio streaming. Sample alignment and enhanced error detection (with PCM samples having more than 8-bits) schemes are possible for further enhancement to achieve better fault tolerance in the software architecture.

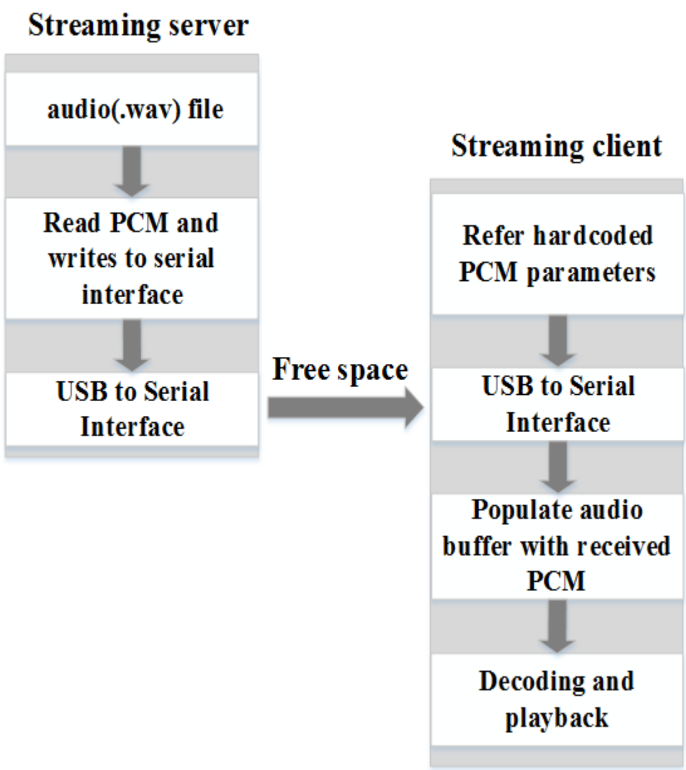

Fig. 2. PCM streaming software architecture

A. 2. MP3 streaming software architecture

The basic architecture of the developed software application for MP3 streaming is shown in Fig. 3. The server reads the media file, containing an audio stream and identifies the audio stream index. Audio packets are read from the audio stream and packetized before constructing the transmit frame. The compiled frame is then passed to the USB to serial interface and transmitted to the client over the VLC channel.

The Client receives the bit stream and checks for alignment and CRC check fields in the frame before de-serialization. De-serialized packets are queued. Clients audio playback thread accesses the queue and decodes audio packets before they are played back. Similar to PCM, the decoder parameters are hardcoded in the client.

Contrary to 8-bit PCM data, MP3 packets are substantial in size (around 480 bytes) and require error free submission to decoder for processing. Thus, we have used a robust frame structure illustrated in the Fig. 4.

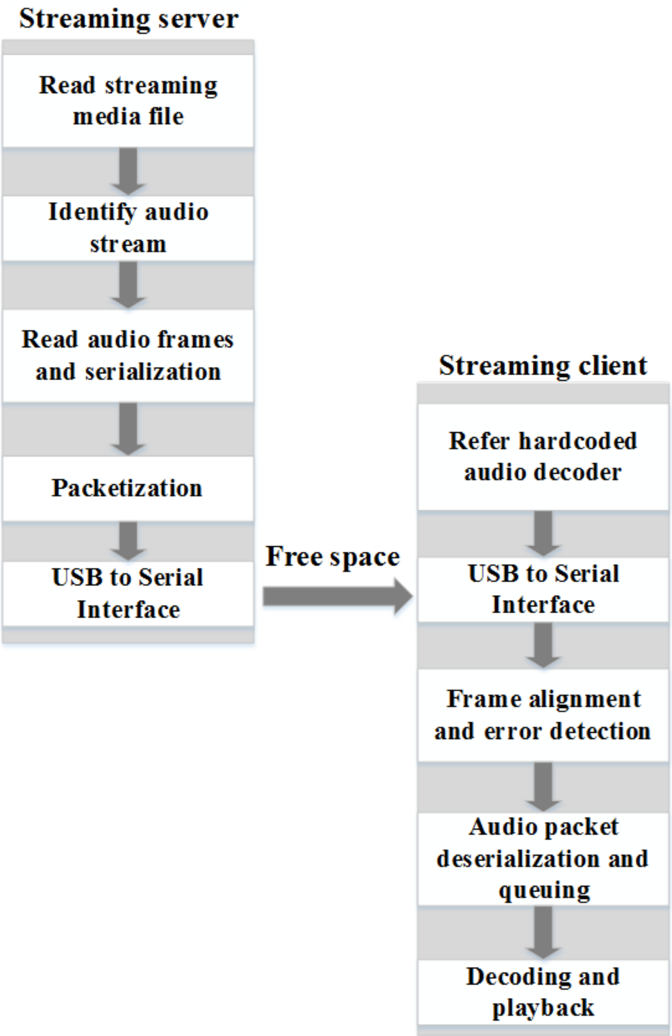

Fig. 3. MP3 streaming software architecture

SFD (4-bytes)	Data (480-bytes)	CRC (4-bytes)

Fig. 4. MP3 streaming frame structure.

MP3 frame structure in FFmpeg implementation, have a maximum size of 480 bytes and minimum size of 470 bytes. A fixed length frame structure is chosen considering the negligible data size variation and to reduce the receiver complexity. The 4-byte start frame delimiter (SFD) informs the receiver about the upcoming packet, thus receiver could dynamically identify the start of MP3 packet from a received bit stream.MP3 data structure(FFmpeg implementation) consist with memory pointers. Change of pointer size due to channel errors could cause client application to crash.CRC-32(cyclic redundancy check), 4-byte value ensures error detection and prevent client application from processing erroneous packets, ensures application continuation with channel errors.

B. Hardware subsystem architecture

The hardware architecture facilitates end to end data delivery from server to client over free space visible light communication. Hardware architecture can be broadly divided into server side and client side. Fig. 5 illustrates the basic composition of either side.

Data written to the serial port (standard USB to serial converter being used as plugin device) by the server application input to the voltage follower (transmitter hardware) which act as isolator between the high bright LED emitting white light (HB LED) and the computer. On-off keying (OOK) is used as the modulation scheme. Output of the voltage follower (LED driver) is then fed into the HB LED and an optical lens arrangement. Optical lens arrangement is being used to converge the LED light to a spot light beam.

A reversed biased photodiode is used to capture the received light. The reverse voltage supplied to photodiode ensures lower junction capacitance, and hence ensures enhanced response of the photodiode. This facilitates higher output current variation for the incoming bit stream through the VLC channel. The current generated by the photodiode is fed to the trans-impedance amplifier which converts it into a corresponding voltage. The trans-impedance amplifier output is impedance matched before feeding into the serial to USB device.

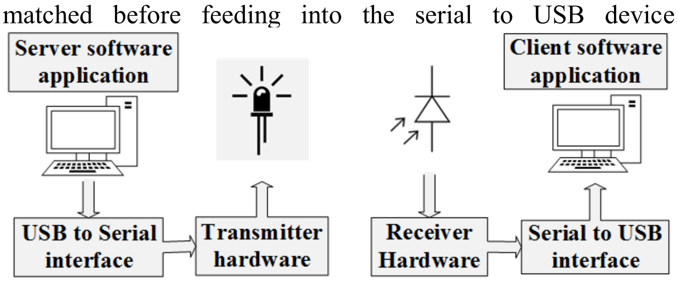

Fig. 5. Hardware architecture overview.

III. PERFORMANCE EVALUATIONS

In the evaluation process, a constant volume of data equivalent to 499712 PCM audio packets and 1024 Mp3 packets (488x1024 = 499712 Bytes) was streamed. Then the packet loss and audio quality for different data rates as well as for different channel lengths were observed. Observations has being carried out on two occasions with different environment conditions (ambient light levels), and obtained couple of readings for each instance. Out of the four experimental values per instance, mean value has being taken as the final observation.

A. *PCM performance evaluation*

Fig. 6 shows received packet percentage of PCM data vs. channel length for different data rates. When increasing the channel length, the PCM packet drop remains negligible up to 19-20cm., The packet drop increases beyond this. This is due to the optical signal to noise ratio (OSNR) degradation of the free space VLC link, causing the received power to fall below the sensitivity of the photodiode.

Fig. 7 shows the audio quality vs. channel length of PCM for different data rates. "Good" perceptible audible quality PCM streaming occurs at data rates above 500 kbps. Although the required minimum data rate for PCM streaming is 352800 bits per second, minimum data rate increases upto 441000 bits per second with the additional bits introduces by the hardware for channel synchronization. Thus, it explains the observerd quality degrade for PCM streaming via lower data rate VLC channels.

With higher quality PCM stereo (with higher sample rate and 16-bit PCM) may expect to reach "very-good" quality audio, the cost of increased streaming data-rate. With higher data rate, experienced audible quality will increase for all channel lengths.

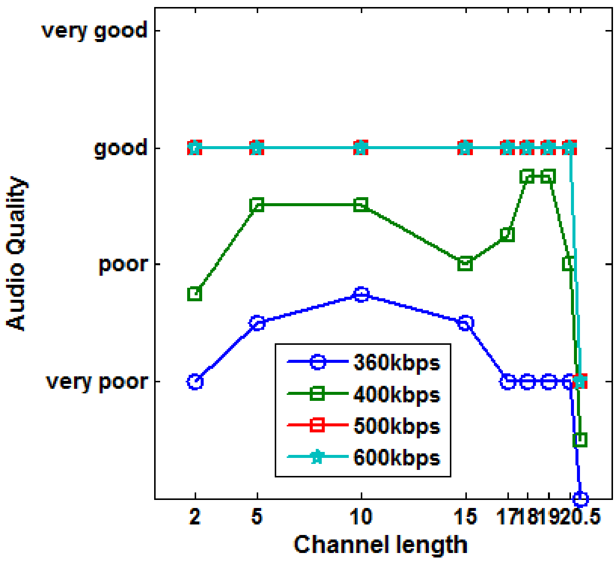

Fig. 7. PCM audio quality for different data rates with varying channel length

B. *MP3 performance evaluation*

Fig. 8 shows received packet percentage of MP3 data vs. channel length for different data rates. Substantial packet drop can be observed in MP3 streaming over lower data rate (150 kbps, 200 kbps) VLC channels.

Fig. 9 shows the audio quality vs. Channel length of MP3. "Good" perceptible audible quality can be experienced by streaming at 200 kbps data rate. "Very good" audio quality can be experienced with minimum data rate of 300 kbps in MP3.

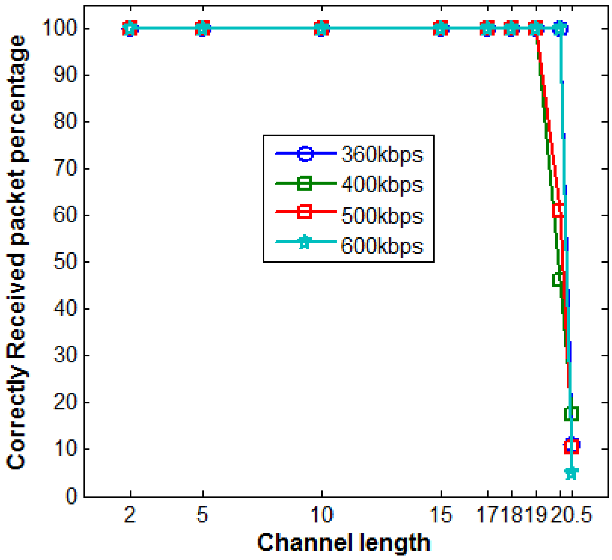

Fig. 6. Percentage of PCM packets received for different data rates with varying channel length.

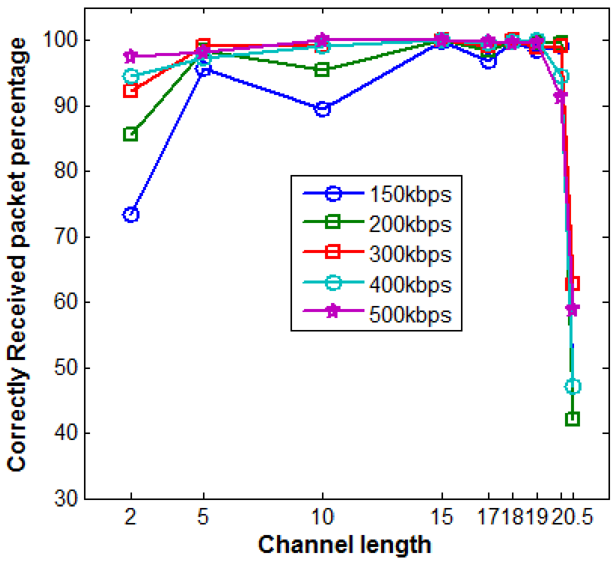

Fig. 8. Percentage of MP3 frame received for different data rates with varying channel length

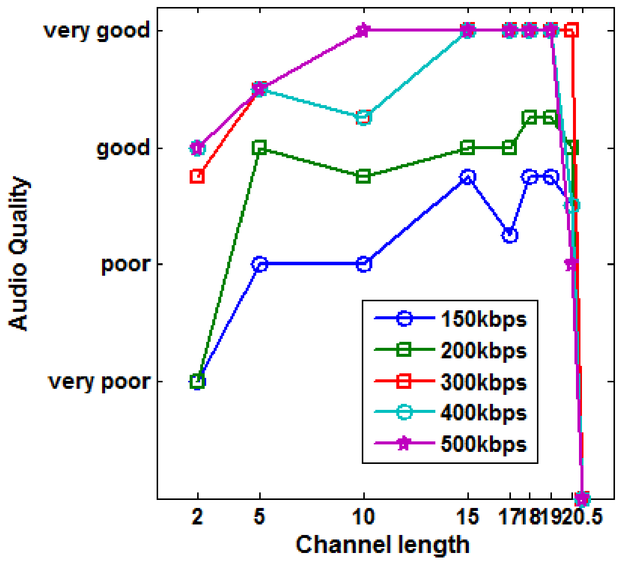

Fig. 9. MP3 audio quality for different data rates with varying channel length

A comparison between MP3 and PCM is given in TABLE 1. MP3 stream can be synchronized effectively if there was a link failure. But MP3 packets will be drop if there was a single error in data packets. PCM data can be stream with some error probability. Because of this the system complexity of PCM is low when compared to MP3.

TABLE I. COMPARISON BETWEEN PCM & MP3

		MP3
Required minimum channel data rate for good audible quality	400 kbps	200 kbps
Bit error influence to audio quality	Moderate	High
Audio synchronization with channel errors	Low	High
System Complexity	Low	High

IV. CONCLUSION

This research paper experimentally evaluate two architectures for both PCM and MP3 streaming over VLC channel. We achieved 20 cm error free channel without implementing any error correction hardware and only using one single optical lens. Beyond 20 cm it will generate errors. This is because OSNR will decrease when increasing the length. For higher data rates the audio quality will get high for both PCM and MP3. However, due to excess bandwidth the client side buffering quantity will increase. From the observations for different architectures, we can deduce that when the data rate is increased the audio quality reaches to an optimum level. Likewise, when channel length is increased, quality suddenly drops to a minimum level due to OSNR degradation. We can observer an increment in channel error percentage when channel length is around 2cm. The reason for this degradation is mainly due to photodiode saturation. Since the separation between the LED and photodiode is low, the free space loss is low, so a much higher power will receive at the photodiode causing a high photocurrent and thus the photodiode output get saturated.

REFERENCES

[1] J.Singh, Vikash,"A New Era in Wireless Technology using Light-Fidelity", International Journal of Recent Development in Engineering & Technology, vol. 2, no. 6, pp. 46-49, 2014.

[2] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. MobiCom, USA, pages 447–458,, 2014.

[3] S. Paul, S. sharma, "Future Of Telecommunication Technologies: Wi-Fi Vs. Wi-Max Vs. Li-Fi Vs. Gi-Fi ISTP Journal of Research in Electrical and Electronics Engineering (ISTP-JREEE), pp: 128-136, 2014.

[4] HU Guo-yong, C. Chang-ying, C. Zhen-qiang, "Free-Space Optical communication using visible light", Journal od Zhejiang University Science A, China, Vol 8(2), pp. 186-191, 2007

[5] C. H. Yeh, C. W. Chow, H. Y. Chen, J. Chen, and Y. L. Liu, "Adaptive 84.44 - 190 Mbit/s phosphor-LED wireless communication utilizing no blue filter at practical transmission distance," Opt. Express 22(8), 9783–9788 (2014).

[6] L. Ding, F. Liu, Y. He, H. Zhu and Y. Wang, "Design of Wireless Optical Access System using LED", Optics and Photonics Journal, vol. 03, no. 02, pp. 148-152, 2013. .

[7] D. Giustiniano, N. Tippenhauer and S. Mangold, "Low-complexity Visible Light Networking with LED-to-LED communication", IFIP Wireless Days, 2012.

[8] S. Schmid, G. Corbellini, S. Mangold and T. Gross, "LED-to-LED visible light communication networks", Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing - MobiHoc, 2013. .

[9] Soundfile.sapp.org, "Microsoft WAVE soundfile format". [Online]. Available: http://soundfile.sapp.org/doc/WaveFormat/.

[10] Dranger.com, "ffmpeg tutorial". [Online]. Available: http://dranger.com/ffmpeg/tutorial01.html.

[11] Ffmpeg.org, "FFmpeg". [Online]. Available: https://www.ffmpeg.org/.

[12] Libsdl.org, "Simple DirectMedia Layer - SDL version 2.0.4 (stable)". [Online]. Available: https://www.libsdl.org/download-2.0.php.