Full duplex Visible Light Communication Platform based on GNU Radio and USRP 2920

Antonio Costanzo ANTONIO.COSTANZO@INRIA.FR
Inria Lille Nord Europe, 40 Avenue Halley, 59650, Villeneuve d'Ascq, France

Valeria Loscri VALERIA.LOSCRI@INRIA.FR
Inria Lille Nord Europe, 40 Avenue Halley, 59650, Villeneuve d'Ascq, France

Abstract

An implementation of full duplex Visible Light Communication, based on USRP platform and GNU Radio environment, is proposed in this work. A main downlink communication between a VLC transmitter and a VLC receiver has been implemented, exploiting a set of Pulse Position Modulation (PPM) with different index. A feedback optical channel, integrating two Arduino 1, have been used for the exchange of control messages. Differing from most of experimental works on VLC, this implementation could potentially allows adaptive communication, advanced synchronization techniques and noise mitigation.

1. Introduction

Visible Light Communication paradigm reuses common LED lamps for both illumination purposes and data communication. The reuse of light power is particularly attractive in terms of energy consumption, since LED lamps are, nowadays, everywhere, and they are permanently turned on in most of indoor environments. Another advantage is due to the possibility of allowing a ubiquitous connection even in those environments, like aircraft, hospitals and chemical factories, where the use of radio-frequencies is forbidden or strictly limited. Differing from other optical technologies, like LASER and optical fiber, VLC does not need a sophisticated hardware, but only common LED lamps and commercial photo-detector. This paradigm potentially represents a low cost solution for high data rate, due to the exploitation of the huge optical spectrum. However, since LED power is normally really low, especially if compared with the one produced by a laser, and the communication does not take place in a controlled environment, the main problem due to the use of VLC in real scenarios is due to the very low signal to noise ratio, especially caused by sunlight interference. This aspect significantly complicates possible applications in real scenarios, so adaptive algorithms, noise cancellation techniques and other effective mechanisms have to be implemented in order to allow the system to properly working also in poor SNR conditions. In order to be implemented in real scenarios, the most of these techniques need a high flexible system, as well as an architecture properly supporting VLC feedback messages (A. Costanzo, 2019), (A. Costanzo, 2021). In order to allow this kind of operations, we implemented a flexible architecture, following the software defined paradigm and using low cost optical transmitters and receiving front-ends. Data communication is based on GNU Radio and NI USRP for the main communication and Arduino 1 for control and feedback operation. A simple exchange of coordination messages, through serial port, managed by a proper python module is established between the USRP and Arduino, both at the transmitter and at the receiver stage.

2. System Design

2.1. Hardware

The main optical communication is set up between two Universal Software Radio Peripheral (USRP) 2922, a universal platform, provided by National Instruments, for software defined operations. However, USRP front-ends have been replaced by receiving and transmitting low frequency daughter-boards, both provided by Ettus, in order to allow modulation and demodulation in the range [0-30 MHz] . Original motherboards have been maintained in the system. Gigabit Ethernet allows communication between USRP and PCs, while internal connection between motherboards and daughter-boards and the external connection between the daughter-boards and the optical circuits are performed by SMA connections. A bias tee and a single stage amplifier are used for allowing Led driving in the proper range [0V-5V], since transmitting USRP can only manage signals in the range [-1V, 1V]. A 12V power supply is used for feeding the LED array. Since the receiving USRP channel

Proceedings of the 3^{rd} European GNU Radio Days, Copyright 2021 by the author(s).

properly works in the range ([-0.9V - 0.9V]), an amplif ing and conditioning network has been designed in ord to allow the receiving signal to be correctly detected in th dynamic range of the receiving USRP. The first amplifi network allows the photodiode to be correctly polarize and the signal to be in the correct range for the secon amplifying network, which performs the high gain tran impedance amplification. An additive DC block capacito mitigates the high DC component due to amplifiers, in o der to avoid saturation of the USRP receiver. A 12V pow supply feeds both the amplifiers. In the uplink commun cation, the optical part of the circuit (LEDs and photod odes) is identical to the one in the main communicatio but analog to digital conversion and digital signal proces ing are performed using Arduino Uno. Since the dynam allowed by Arduino card ([0V-5V]) is larger than the on of USRP, bias tee and amplification are not needed in the feedback transmitting stage. Uplink receiving front-end is simply composed by a photodiode and a resistance, since the bandwidth needed for feedback channel is quite low, if compared with the one of main communication, and no particular issues in terms of dynamics could complicate signal reception by the Arduino. In both receiving and transmitting stages, a serial connection (allowing a standard 115200 baud rate) is used for communication between the Arduino and the corresponding PC. A paper box with black inner walls has been placed around the receivers, in order to avoid, as well as possible, optical interference between the two couples of front-ends. Since uplink channel does not need a very high data rate, in order to further reduce crosstalk between uplink and downlink communication, a low frequency OOK has been considered for communicating feedback messages. In addition, Manchester encoding has been used in order to reduce the problem of DC components, without using further block capacitance. Since PPM communication works at much higher blinking frequencies (up to some MHz), the interference between the main communication and the feedback communication is negligible. Feedback receiver circuit and main VLC transmitter shares a 12V feeding power supply, while the power supply of the main receiver is also used for feeding uplink transmitting LED. In both cases, some shunt capacitance have been used in order to reduce, as far as possible, high frequency noise components due to the feeding network. A picture of the corresponding setup is provided in Fig. 1, while a block diagram of system architecture is provided in Fig. 2.

2.2. Software

In order to reduce hardware costs and improve system flexibility, most of signal processing operation (i.e. signal generation, filtering, modulation, demodulation, time recovering and data evaluation) are performed via software, using GNU Radio for the main communication and the Arduino IDE for feedback communication, following, in both cases, a software defined approach. The main high data rate communication, involving the two USRP platforms, is completely managed using GNU Radio. In this work we have implemented Pulse Position Modulation (PPM), with different possible indexes (2,4,8 and 16) as modulation scheme for data transmission. Furthermore, proper modules have been designed for data elaboration, real time performance evaluation, generation of control message. A proper python block, integrated in the GNU Radio diagram, allows the exchange of control messages between the Arduino and USRP. Uplink communication is completely managed using the Arduino integrated development environment (IDE). For maintaining a proper reactivity of the device, speed up the access to the registers, and consequently, improve data rate communication of uplink stage, the code is directly flashed in Atmel 8-bit AVR micro-controller integrated in the Arduino, and standard IDE commands for read and write operations have been replaced with appropriate low level instruction. Main communication and uplink communication work at different data rate, so, in order to preserve real time operations and avoid to waste system performance, GNU Radio subroutines and the code flashed in the Arduino are executed in parallel. The exchange of control messages between GNU Radio and Arduino takes place through a serial communication, managed by a proper python Block. In order to preserve a multi-thread logic, a concurrent dedicated subroutine is provided in both transmitting and receiving stages for managing serial communication without stopping signal processing. Once performance related to a portion of received signal are evaluated, a new control message is im-

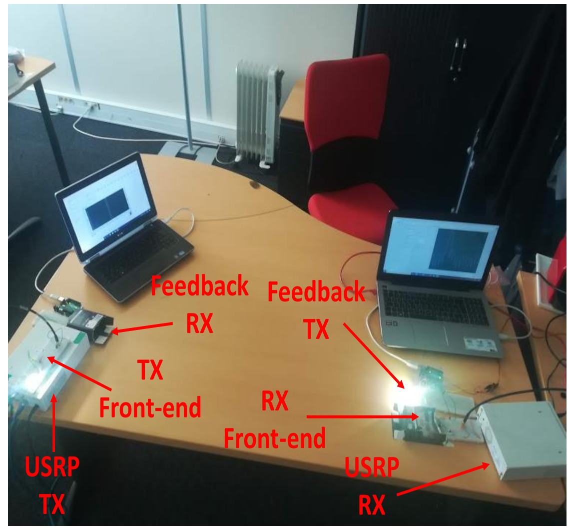

Figure 1. Main Setup

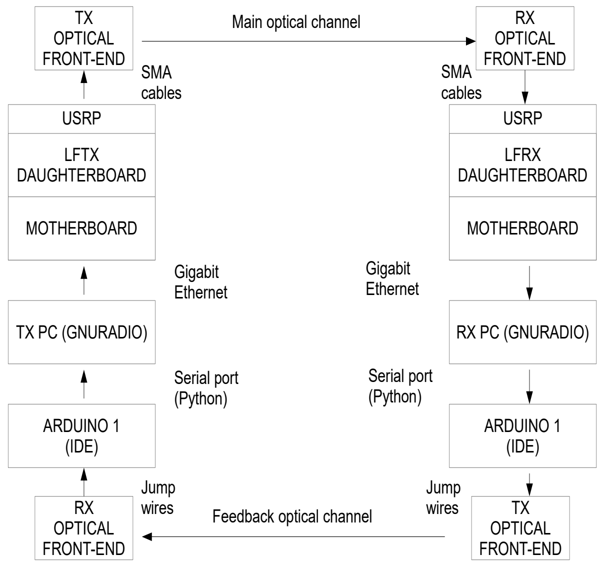

Figure 2. System Architecture

mediately generated via GNU Radio and transmitted to the serial port. An Arduino routine reads the stream incoming from the serial port, encodes and transmits feedback message. On the other side, the uplink RX Arduino receives and demodulates the feedback message. Thereafter, the control message is transmitted to the serial port, and a parallel subroutine, executed in GNU Radio, continuously listens for new messages and periodically communicates with the main subroutine. Feedback mechanisms could be transparently applied to different possible applications, so, in current implementation we just communicate on serial port one printable ASCII character per time, which represents a generic exchange of information between the transmitter and the receiver. For example, a set of 4 different characters could represent the next modulation index in an adaptive system which dynamically switches between 2PPM, 4PPM, 8PPM and 16PPM, depending on Signal to Noise Ratio Condition. In this case, since a restricted subset of characters is used for codifying control messages, if a feedback message is lost, or if it is corrupted, the main communication can be easily performed using previous parameters and there is no lack of communication even if the feedback mechanism fails.

3. Conclusions

In this work we have designed and implemented a complete VLC bi-directional system based on USRP and commercial LEDs and photodiodes, able to receive and send back on a dedicated back channel. In particular, a software defined approach, based on GNU Radio, has been developed in order to exchange generic information back to the transmitter. This architecture can be useful for implementing noise mitigation techniques, effective synchronization, and many other operations needing a feedback control in a real scenario.

References

A. Costanzo, V. Loscri, M. Biagi. A noise mitigation approach for vlc systems. In *Global LiFi Congress, Paris, France, 2019*, 2019.

A. Costanzo, V. Loscri, M. Biagi. Adaptive modulation control for visible light communication systems. *Journal of Lightwave Technology, doi:10.1109/JLT.2021.3056177*, 2021.