

# Purple VLC: Accelerating Visible Light Communication in Room Area through PRU Offloading

Shengrong Yin, Nour Smaoui, Milad Heydariaan and Omprakash Gnawali  
Department of Computer Science  
University of Houston  
{syn, nour, milad, gnawali}@cs.uh.edu

## Abstract

Embedded VLC (Visible Light Communication) has attracted significant research attention in recent years. A reliable and robust VLC system can become one of the IoT communication technologies for indoor environment. VLC could become a wireless technology complementary to existing RF-based technology but with no RF interference. However, existing low cost LED based VLC platforms have limited throughput and reliability. In this work, we introduce Purple VLC: a new embedded VLC platform that can achieve 100 kbps aggregate throughput at a distance of 6 meters, which is 6-7x improvement over state-of-the-art. Our design combines I/O offloading in computation, concurrent communication with polarized light, and full-duplexing to offer more than 99% link reliability at a distance of 6 meters.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*wireless communication*;  
C.3 [Computer Systems Organization]: Special-Purpose and Application-Based System—*real-time and embedded systems*

## General Terms

Design, Experimentation, Architecture

## Keywords

Visible Light Communication, Internet of Things, Wireless Platform

## 1 Introduction

Wireless Communication for IoT covers various technologies from the radio spectrum to light spectrum. The introduction of the 802.15.7 standard further lays the foundation of practical visible light communication systems [9]. Current low-end and low-cost VLC systems are limited to less than one meter communication range [29][18]. This

makes VLC inapplicable for room area IoT networking compared to other wireless technologies. Meanwhile, high end VLC systems are too costly for general use in IoT devices although they achieve impressive data rates. Researchers have proposed low-end VLC platforms that can be built using off-the-shelf components [32], led-to-led communication [24] and Multi-hop VLC [19]. However, these platforms suffer from limited communication range in room area as well as limited data rate for low power and often also suffer from poor reliability and robustness. Thus, the vision to combine illumination and communication in a systematic and low-cost manner is a long way from being realized.

It is important to improve the reliability and speed of low end VLC in room area so that we can use them in emerging IoT applications. Making visible light communication reliable in room area can provide an alternative option for networking the IoT devices in smart homes. Augmenting LEDs at homes with communication technology and providing reliable and robust wireless communication simultaneously will enable interesting design in illumination and networking infrastructure for the homes. In addition, this technology will provide alternative communication option for devices and gadgets when existing RF-based wireless faces challenge or is not available. Thus, the research community is actively trying to improve the low-end embedded VLC technology for IoT.

It is challenging to achieve high data rates in embedded VLC because of hardware limitations and prevalent software approaches. Highly sensitive photodetectors would improve the reliability but would be too costly and depending on the control interface could introduce system complexity. US-RPs allow fast modulation and demodulation but could be too costly for inexpensive IoT applications. On standard IoT platforms, software-based modulation and demodulation may be too slow. Thus, resource constraints pose tremendous challenge when we try to make low-end VLC systems fast and robust.

There are two main reasons why previous approaches have limited performance in low end VLC in terms of throughput and communication range. First, writing low-level software for MCU to toggle GPIO pins will not be fast enough for high data rate communication. MCU-based platforms in Klaver et al.[19] and Schmid et al. [24] are handling both clock signal generation and data processing task in a single-core. With this approach, the software cannot gener-

ate tens of kHz GPIO toggling. Second, it is also popular to use kernel-based GPIO toggling in a single board computer IoT platform. This approach requires external real-time library to generate the clock signal in the kernel of the OS. Meanwhile, the fastest GPIO toggling is also limited to tens of kHz [4]. Experimental evaluation indicates faster GPIO toggling from kernel module may trigger kernel oops or even freeze the OS in [16].

We design and implement Purple : a new embedded VLC platform based on the same single board computer. Purple has techniques to significantly boost embedded VLC performance in a reliable and robust manner by offloading computation from the kernel to the coprocessor. We utilize the coprocessor resources that are already present in a widely used single board computer platform called Beaglebone, which is cheap, reliable and provides rich peripherals in a single SoC for better prototyping in the IoT domain.

Purple is an embedded VLC platform built on top of Beaglebone. Purple can achieve 100 kbps data rate for room-area networking in IoT. At a high level, the approach utilizes a multi-processor architecture to speed up embedded VLC. Purple utilizes hardware resources that were not fully exploited by previous VLC platforms and uses appropriate software architecture to exploit the hardware. Specifically, it combines I/O offloading, adaptive transmission, simultaneous dual link communication and full duplexing in a low-cost and systematic manner.

The full hardware schematics and software for Purple VLC have been released and are available on github at <https://www.github.com/gnawali/purple>.

We make these contributions in this paper:

- We present Purple , a fully open-source embedded VLC platform that achieves 50 kbps effective throughput at a distance of up to 6 meters, 99% link reliability under normal ambient light conditions and 100 kbps aggregate throughput under full duplexing. This is at least 6-7x improvement over the state-of-the-art.
- We design, implement, and evaluate three techniques to improve embedded VLC that utilize modern single board IoT platforms such as Beaglebone. The techniques are VLC RX/TX IO offloading to PRUs, utilizing multiple processing/IO elements to drive full duplex communication channels, and prevent concurrent VLC channels from interfering with each other by channel isolation using light polarizers.
- We evaluate the system in both physical and link layer and provide a full Linux driver/firmware implementation of our approach. We open source the hardware and software of Purple platform.

## 2 Related work

We briefly review work related to reliable and robust visible light communication.

**Low-cost embedded VLC platforms:** Shine [19] is a low cost VLC platform that can provide a data rate of 1kbps with a communication range of around 1 meter. LED-to-LED communication system [24], [25] have been known to achieve a data rate of 800bps with an operating distance of

2 meters. More recently, OpenVLC1.0 was proposed as a low-cost embedded VLC platform [32], [30]. Heydariaan et al. investigated its performance under various experimental settings [16] and found that the system achieves a maximum data rate of 12kbps. These platforms do not provide data rate for IoT devices comparable to Bluetooth or low power wireless, making visible light communication far less appealing as a medium used in room-area networking. ModBulb [15] took the VLC performance one step further using a FPGA-based control. Their system can generate 1mbps base band signal, however their work's focus is on the transmitter side and hence it is unclear what a low-cost receiver design would look like for such a system. Although FPGA can be low-cost, it does require at least modest complexity hardware design compared to providing mostly software-based solution by fully utilizing resources on low-cost and popular IoT platforms such as Raspberry Pi and Beaglebone. In our work, we address both transmitter and receiver design to build a complete end-to-end system. Our transmitter routinely achieves more than 1 Mbps but the receiver becomes the performance bottleneck. Most Recently, Philips Lighting Research proposed a two way communication using one single RGBW LED in [20] and [21]. Their prototype can transmit and receive light signals in parallel using only one LED rather than photodetectors and achieve several kbps data rate at a distance of tens of centimeters. Yin et al. in [36] proposed an adaptive ambient light cancellation VLC platform robust to dynamic ambient light. The operating distance for the platform is also limited to tens of centimeters. In this work, Purple VLC can achieve in the order of hundred of kbps data rate at a distance up to 6 meters.

**Multi-processor architecture.** Researchers have utilized I/O offloading in multi-processor architecture. Zhai et al. in [37] identified the problem that various I/O devices can slow down the system performance and they reduced the overhead significantly by offloading I/O performance to a slower processor while keeping the main processor running normal OS scheduling tasks. Islam et al. presented a system with multi-processor architecture that can offload the audio data sampling from main processor to a weak processor [26]. Inspired by these work, we offload VLC related TX/RX IO tasks to the PRU on the Beaglebone platform and achieve the same speedup benefits that were realized in other domains. Thus, our work show that the offloading architecture is useful for VLC design and should be considered in future platform designs.

**Light-based sensing and communication system:** Light based sensing applications are active research areas in the past few years, especially using light for indoor localization. Yang et al. proposed an indoor positioning system using polarized visible light [35]. Their system has a data rate of 50bps data rate which is enough to send location beacons to the camera. LiTell et al. in [38] introduced a visible light based localization system that can sense the unique oscillating frequency for fluorescent light in buildings by using a portable light sensing dongle with TIAs (Transimpedance Amplifier) and Picoscope. Varshney et al. in [28] introduced a new visible light sensing system that utilized solar panel as the light receiver to achieve ultra low power sens-

**Table 1. Performance for state-of-the-art embedded VLC.**

System	Dietz et al. [13]	Schmid et al. [24]	Klaver et al.[19]	Wang et al. [31]	Hewage et al. [15]	Li et al. [21]	Our Work
<b>Data Rate</b>	250 bps	800 bps	1 kbps	16 kbps	1 mbps	1-10 kbps	100 kbps
<b>Distance</b>	~10cm	~2m	~1m	~5m	NA	~20cm	6m
<b>Multi-hop</b>	No	No	Yes	No	No	No	Yes
<b>Full-Duplex</b>	No	No	No	No	No	No	Yes
<b>Parallel Channels</b>	No	No	No	No	No	No	Yes
<b>Implementation</b>	MCU	MCU	MCU	ARM	FPGA+MCU	MCU	ARM + PRU
<b>Antenna</b>	LED-to-LED	LED-to-LED	LED-to-PD	LED-to-LED/PD	LED-to-PD	RGB-to-RGB	RGB/LED-to-LED/PD

ing. They leveraged radio backscattering for the communication part. This adds system complexity since it involves both light sensing and backscatter platforms. However, the proposed design in our system focus on simple and dedicated light communication performance.

### 3 Limitations of Existing VLC Platforms

We first give an overview of state-of-the-art embedded VLC platforms and the tradeoffs they make in their design. Table 1 summarize current embedded VLC platforms and compares them based on system performance, networking architecture and system flexibility. Among all these platforms, OpenVLC1.0 demonstrated the highest performance with a data rate of 16 kbps and a communication distance up to 5 meters using LED-to-PD antenna pair. To achieve that performance, it used a 1W LED-array as the transmitter and a relatively expensive photodiode with on-chip amplifier. The Purple platform has a 6-7x performance gain with similar electronic parts, but the transmitting LEDs consumes much less power. Another important aspect for embedded VLC networking is support for multi-hop communication. Shine is the first platform that supports this feature and demonstrated potential for future embedded VLC networking but it is limited to half-duplex multi-hop. Our design of Purple achieves multi-channel and full duplex networking and consequently high data rates. We next discuss the main performance and flexibility bottlenecks of state-of-the-art VLC platforms.

#### 3.1 Limited Clock Rate on Microcontroller-based VLC

Microcontroller-based VLC such as Arduino-based VLC system has up to 1kbps data rate [24]. The same group also designed a SoC-based Linux bulb that can achieve less than 1kbps data rate in [23]. The bottleneck for their design is due to the limited GPIO toggling frequency since the Microcontroller clock runs at 8-64 MHz. Supporting higher data rates will require faster GPIO toggling. The other issue for Microcontroller-based VLC is due to the limited GPIO pins and ability to control them concurrently to support concurrent VLC channels. Limited performance for Shine platform [19] was mainly due to the controller selection. As a result, on those platforms with a single MCU, the system cannot both transmit and receive at the same time and be limited to half-duplex communication. A full-duplex feature would not only increase point-to-point data rate but also minimize forwarding latency in a multi-hop setting.

#### 3.2 Kernel Overhead with I/O scheduling

Platforms such as OpenVLC [32] are built on top of more resourceful single-board computer platforms such as Beaglebone which can run Linux. OpenVLC implements a full VLC PHY/MAC layer as a kernel module. Implementing VLC RX/TX IO as part of kernel module can potentially make the software largely portable to other platforms that run Linux but the flexibility comes at a significant performance penalty because the kernel handled busy GPIO toggling tasks. In addition, because of that approach, to maintain a low CPU utilization, we need to also limit the toggling rate. As a result, their platform caps the ADC sampling rate used in reception: 75 kilo samples per second with 3.0v power supply and 200 kilo samples per second with 5.0v power supply. Offloading the RX/TX related IO tasks to another processing unit (PRU in case of Beaglebone) would free the processor from the overhead while achieving much faster IO performance leading to higher VLC data rate.

#### 3.3 FPGA-based Transmission and USRP-based Reception

FPGA has been used to achieve faster IO, for example in modBulb [15]. USRP could also achieve similarly high performance [27]. These approaches can easily generate toggling rate in the MHz range but they introduce system complexity and cost. Our work is motivated by the observation that the low-cost single-board IoT platforms already have resources (e.g., the additional processors called PRUs in Beaglebone) which can be utilized to achieve similar performance but without additional hardware or cost. Beaglebone-based VLC system can generate toggles at more than MHz without adding any USRP or FPGA hardware to the platform and the associated complexities.

### 4 System Design

Our design of Purple , a new embedded VLC platform, can achieve high data rate without using FPGA or USRP while allowing low-power platforms to keep their CPU utilization low. The platform integrates RX/TX I/O offloading with the capability to change the number of LEDs for communication performance objectives, multiple concurrent communication channels and full-duplexing to achieve high data rates. Next we describe the design of Purple .

#### 4.1 Transceiver Architecture

Fig. 1 shows the transceiver architecture. The transceiver architecture consists of TX (transmitter) and RX (receiver) modules. The transceiver supports LED to LED communication or LED to photodiode communication. It can use single color LED or RGB LED as the transmitter. Fig. 2

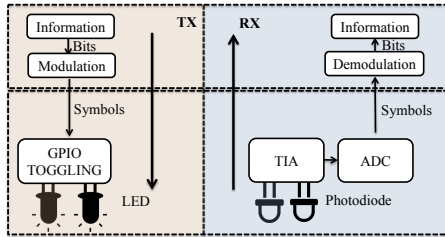


Figure 1. Bidirectional transceiver

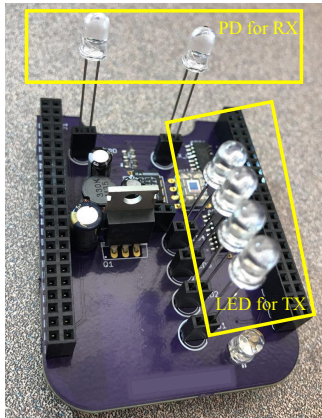


Figure 2. The Purple board showing four TX LEDs plugged in and two elements plugged in for RX chain.

shows the implementation of our design. We have populated four sockets (each with 2 pin header) for transmitter on the printed circuit board (PCB) so that users can use multiple or different types of LEDs and enable different combination of them depending on their communication objectives. Thus the platform can support multi-transmitter VLC with up to 4 LEDs. We also populated two sockets for the receiver. The users can insert either photodiode or LED as the reception antenna. Rest of the components and the software is designed so that the board can use any combination of LED or Photodiode from any socket. Concurrent channels can be established with this board since the TIA and the ADC can support 2-channel data acquisition simultaneously.

#### 4.1.1 I/O offloading

Fundamentally, VLC transmitter requires toggling or controlling the LEDs to encode the bits using different coding schemes. Previous approaches used a kernel based driver to control the IO [32] and thus could not achieve high data rates due to lower GPIO toggling rates: first high rates are difficult to achieve with that approach and second the rates had to be kept lower than what is possible so as to keep the CPU utilization low. In our design of Purple, we offload the control of IO operations from the main processor to the PRU, which is an auxiliary processor on the Beaglebone platform. Our use of the Programmable Real-Time Unit (PRU) to toggle GPIOs to encode the bits frees up the main CPU from this responsibility while achieving a high frequency and flexible GPIO control [14], [17]. Reports indicate PRU-based GPIO toggling is ~40 times faster than CPU (main processor) based

GPIO toggling [10], [1]. CPU will have longer processing delay over GPIO toggling compared to PRU since it needs to run OS and manage all the available onboard resources. Thus, I/O offloading from CPU to PRU can provide the system the flexibility on resource management while offloading the timing critical tasks related to VLC RX/TX to a specific coprocessor that can handle the transmission or reception in a fast and reliable manner. Further the software architecture is portable and runs on the BeagleBone board [5] with a standard Debian image.

PRU is a coprocessor with a 200 MHz clock frequency. Its deterministic characteristic can be utilized for real-time tasks and time critical responses. Considering this feature, PRU is an ideal processor to run RX/TX software for embedded VLC and achieve high performance. Fig. 4(a) shows Purple generating approximately 1 MHz signal to blink the LEDs in the transceiver without using additional hardware such as FPGA or USRPs. Speeds higher than 1 MHz can also be achieved but may degrade in performance due to high speed signal issues, not due to processing bottlenecks. One MHz toggling has been reported in VLC literature [15] and it requires FPGA but Purple does not add new hardware, simply utilizes the resources readily available in Beaglebone platform. Thus one of our contributions is to educate the community about this under-utilized resource that could dramatically improve the performance of the IoT platforms they are building.

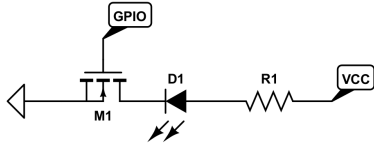
#### 4.1.2 Using Different Number of LEDs for Transmission Depending on Performance Objectives

Depending on the goals, the system has to control those LEDs *concurrently* or *independently*, both modes supported by Purple. We next describe the design issues and tradeoffs to support those two modes of multi-LED control.

The capability to change the number of LEDs in the VLC transmitter enables Purple to support different scenarios and performance objectives. If the VLC transmitter uses more LEDs, generally more light is generated. Then the photodetectors can detect more light, making the system perform better compared to single LED approach, especially for longer distances. More LEDs can achieve a larger field of view (FoV) increasing deployment flexibility: the photodetector does not need to be aligned with the transmitter in a precise manner. Using more LEDs also consumes more energy compared to using a single LED. On the other hand, using fewer LEDs will decrease the amount of light and also decrease the FoV. Thus, Purple's ability to support different number of LEDs is essential in a flexible VLC platform because different scenarios have different constraints and objectives.

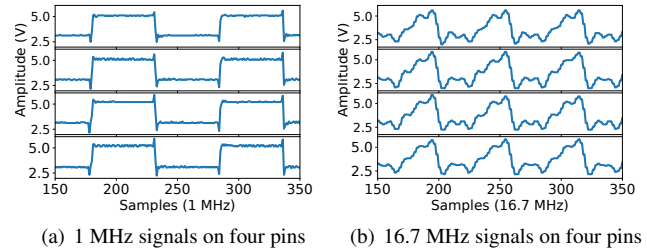
Fig. 3 shows the circuit for controlling one LED. For off-the-shelf 5 mm LED, the driving current is approximately 20mA. Purple has four copies of such circuit. Further, each LED driver circuit can be enabled and controlled independently. Controlling such a large number of GPIOs and hence LEDs independently at high speed using a single core architecture of the previous systems from the literature is not possible and hence requires the IO offloading technique we explore in this work.

**LED Synchronization Issues:** If we use multiple LEDs to simply extend the range and improve the reliability, we

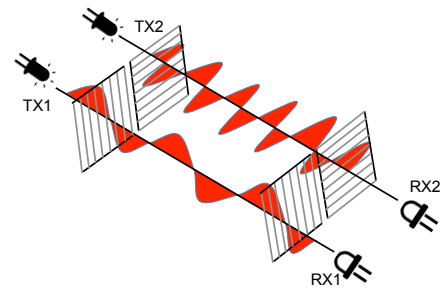


**Figure 3. GPIO controlled LED.** We duplicate the driver circuit to drive 4 LEDs with the GPIOs on board.

need to make sure that all the LEDs are controlled synchronously, i.e., all the LEDs are turned on and off at exactly the same time to transmit the encoded information. If extending the range and improving the reliability was the only goal for using multiple LEDs, we would simply hard wire multiple LEDs on the board or use enable/disable switch on the circuit for each LED and still control them from a single GPIO. That way all the LEDs are precisely synchronously controlled. In our case, we also want to be able to control the LEDs independently (to support concurrent channels) thus hard wiring or just enable/disable approach is not adequate as each LED requires independent GPIO control. Thus there is a need to synchronize GPIO toggling for data transmission using multiple LEDs. In our design, PRU can control multiple GPIOs in concurrent manner. In order to evaluate the performance of the GPIO toggling, PRU allows us to control the GPIOs independently using an IO register (32-bits) within one clock cycle. Thus rather than control one pin at a time, we can control a larger number of pins within a clock cycle. We can generate various toggling speed for the LEDs by modifying the waiting period. We use an oscilloscope to measure the generated clock signal while toggling the LEDs with the driving circuit shown in Fig. 3. The oscilloscope can sample 4 channels at 250 MHz concurrently. Fig. 4 shows the waveform at various toggling frequency. We find that the signals across the GPIO channels are synchronized due to the precise control from the PRUs. At 1 MHz, the four channels has a minimum offset of 0.5 ns and a maximum offset of 1.5 ns; and at 16.7 MHz, we observe the same minimum offset and maximum offset. Although the resolution of this instrumentation is 5ns, we used manual cursor on the oscilloscope to measure these sub-5ns offsets. These offsets are mainly due to the imprecise response time of the LEDs we used in our experiment. Typically, these LEDs can respond to toggling within 2 ns. However, at 16.7 MHz, waveform becomes distorted, making it more challenging to represent symbols using simple encoding schemes. Purple uses a symbol rate of 100 kHz, thus the 0.5-1.5ns offset is negligible at this rate. With the resolutions of less than 5ns synchronization across the pins, Purple is able to generate a stable symbol rate at 1 MHz for multiple LEDs. This level of synchronization would be difficult to achieve from on-CPU approach, with low utilization, using a Linux driver as in the previous platforms. In addition, users can decide the number of GPIOs to be controlled by the coprocessor and how to control them. For example, the user can configure 3 LEDs to provide illumination (The LED is always turned on) and one LED is used to provide communication. Thus, using co-



**Figure 4. 1 MHz and 16.7 MHz signals captured by the oscilloscope on the four pins with current driving circuit.** 16.7 MHz signals are significantly distorted making them unsuitable for VLC transmission. 1 MHz signals are clean and tightly synchronized demonstrating the level of control needed for independent and synchronous operation.



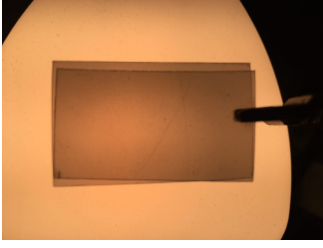
**Figure 5. The links will be isolated due to the use of polarizers such that data can be send concurrently without interfering with each other.**

processors such as PRUs not only enable us to control multiple LEDs when they need to transmit synchronously but also enable flexible design without the use of hardware thought to be needed for such precise and synchronous control (e.g., FPGAs or USRPs).

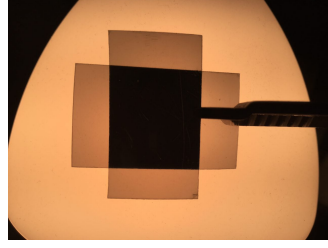
**Transmission Power Control:** Increasing the transmit power generally extends the range or improves reliability. Using the Purple 's capability to synchronously toggle the LEDs and select the number of LEDs, the user can change the effective VLC transmit power. The other way to provide transmit power control capability would have been using LEDs that emit the amount of light that is highly responsive to the current provided but that would have required building circuit that can provide variable current and more complex software. Hence we opt for the simple approach that just changes the number of LEDs to provide the capability to adjust transmit power control.

#### 4.1.3 Isolating Concurrent Channels

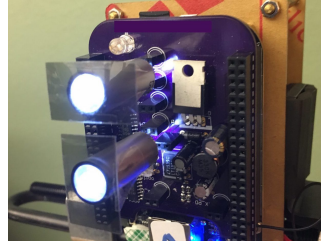
Now we describe how we achieve concurrent channels in embedded VLC. We present the overview of the concurrent design in Fig. 10. It shows concurrent data flow from TX to RX in two channels. For the software design, we configure one node to run the transmission program, the other node to run the reception program. We use two off-the-shelf



**Figure 6. Polarizers in parallel orientation**



**Figure 7. Polarizers in orthogonal orientation**



**Figure 8. TX: vertical and horizontal polarizers.**



**Figure 9. RX's view of the transmitter**

photodiode to detect the light signals with a double channel transimpedance amplifier (TIA). The amplified signal will then be sampled using an ADC with simultaneous channel sampling.

In RF based communication system, mutual interference across the two channels is the main problem in concurrent communication in the same frequency. However, in our design, we use polarizers to cancel the mutual interference such that each link can be independent. The principle of polarizer has been explained in [35]. The idea is to isolate the two concurrent transmissions such that the receivers can receive the corresponding data on each link. Fig. 5 shows the concept behind our approach. When the polarization angle matches between the receiver and the transmitter there is effective VLC TX/RX communication (see Fig. 6) otherwise not enough light passes through to the receiver resulting in poor communication. (see Fig. 7). Marus' law tells us how much light passes through for different angles:

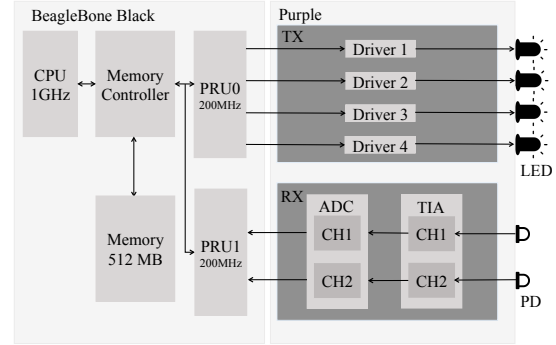
$$I = I_0 \cos^2 \theta \quad (1)$$

$I$  represents the light intensity after the two polarizers.  $I_0$  is the light intensity between the two polarizers. Marus' law applies if these two polarizers form an angle  $\theta$ . For example,  $\theta = 0$  in Fig. 6,  $\theta = 90^\circ$  in Fig. 7.

Fig. 8 shows how we implement the design in practical systems. We create a cylindrical LED housing/case with polarizers on the front and insert the LED inside this housing. Thus the light emitted by the LED has to go through the polarizer in the front. Further the housing can be physically rotated to achieve the desired angle of polarization. Thus, light from two LEDs will be two light beams with different polarization. The receivers LED or PD also have the cylindrical polarizer case on them. We turned on two LEDs in one node. Fig. 9 represents the PD's view after light passing through the polarizers. We observed only one LED illuminating from each photodiode demonstrating isolation between the two concurrent channels.

## 4.2 Full-Duplex VLC

Existing embedded VLC platform are either simplex or half-duplex, i.e., they can only transmit or receive at a time even though they support bi-directional communication. In a single MCU-based design, half-duplex communication is naturally supported: the MCU can run the transmit logic or receive logic at a time. However, in single board computer based design, such as OpenVLC1.0, there are sufficient on-



**Figure 10. System architecture to enable concurrent communication.**

board resources to provide full-duplex communication. This can potentially increase communication throughput and reduce latency. Full duplex communication has been explored in RF communication for decades. The problem for RF full duplex is self-interference, which means, the TX will interfere with RX on the same node. In RF domain, complex cancellation circuit is needed to cancel self-interference, particularly in the omnidirectional RF work.

Light transmitted by LEDs is directional and hence selection of LEDs, careful placement and orientation of the transmitting elements is sufficient to prevent self-interference at the physical layer. However, one also needs an appropriate hardware and software architecture to allow concurrent execution of transmission and reception logic and IO operation. Thus, we need two capabilities in full-duplex VLC platform: (1) prevention of self interference and (2) concurrent TX and RX processing elements. Purple provides both capabilities by utilizing resources available on Beaglebone.

In Purple architecture (Fig. 10), two PRUs (coprocessor) can operate independently supporting both transmission and reception at the same time. We configure the platform to connect one LED to PRU as the TX and one PD as the RX. The onboard TX and RX operate in parallel to achieve full duplex point-to-point communication.

**Self-interference in full-duplex VLC.** We now study the level of self-interference that can occur depending on the way we place the LEDs on the VLC board. More specifically, we run experiments to determine the level of self-interference as a function of antenna distance on the board.

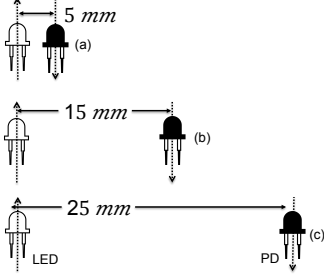


Figure 11. Three configurations for antenna distance.

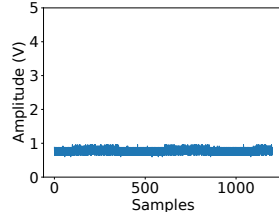


Figure 12. Received signal with antenna distance of 25mm.

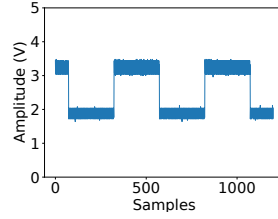


Figure 13. Received signal with antenna distance of 15mm.

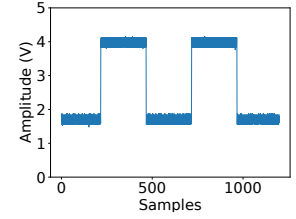


Figure 14. Received signal with antenna distance of 5mm.

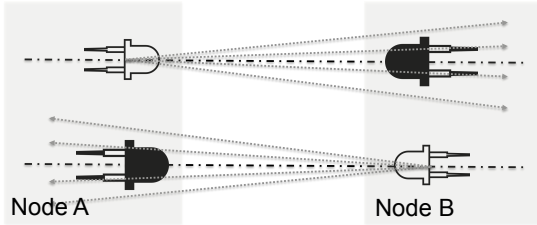


Figure 15. Configurations to enable full duplex communication.

Fig. 11 show the configurations we use for the antenna distance. We keep LED and PD at 5mm, 15mm, and 25mm. The LED is highly directional, so we place LED and PD with transmitting and receiving beam angle of  $180^\circ$  at each distance. We configure the LED to send continuous periodic signals and observe the received signal from the PD using an oscilloscope with passive sampling. We run the experiment in a lab environment where the ambient light is always on. We expect the received signal to be constant as we assume the PD is only detecting the ambient light intensity.

We observed the received signal had minimum interference from LED when the antenna distance is 25mm as shown in Fig. 12. We find that this level of interference has negligible impact on full-duplex VLC communication. With 5mm distance between the LED and PD, there was a strong interference from the LED to the PD as shown in Fig. 14. When LED and PD are side by side at a distance of 5mm, theoretically, the PD should not receive any signal from LED since it is out of LED's FoV. However, the directional LED we are using also emit photons outside of its FoV. The imperfect manufacturing process might cause its unpredictable FoV. For the antenna distance of 15mm, the received signal has weaker interference from the LED (Fig. 13) compared to 5mm. For each distance, we also slightly rotated the PD to form a different beam angle between LED and PD ranging from  $150^\circ$  to  $180^\circ$ . The rotating result for each antenna distance is similar, but the pattern is the same. The interference from LED becomes weaker at larger antenna distance. The evaluation section will present detailed results.

Table 2. Data Transmission Frame Format

Preamble	SYNC	SFD	Data	EFD
0xAAAAAA	0xD5	0x02	N bytes	0x03

## 4.3 Encoding and Decoding

### 4.3.1 Transmission

Our transmission mechanism consists of two main parts: (1) framing and (2) modulation. Similar to any other asynchronous communication system, we need a frame format for data transmission to ensure that the receiver is able to detect and receive the packet. We use the frame format shown in Table 2. By including a preamble, we make sure that the receiver is able to adapt its threshold to distinguish between 0 and 1 symbols. To further improve the reception, preamble can be sent continuously when no data is available to send. Start of frame delimiter (SFD) indicates where the receiver should start reading the packet. End of frame delimiter (EFD) indicates that the packet has ended and is a constant and weak form of error checking compared to Cyclic Redundancy Check (CRC).

We modulate every byte with on-off keying (OOK) modulation with Manchester coding. Symbol sequence of 10 represents bit 0 and symbol sequence of 01 represents bit 1. Further to keep LED in light emitting state, we add a 0 start bit (10 symbol) and a 1 end bit (01 symbol) to each byte. A delay is added between transmission of each symbol which defines the symbol duration and consequently the data rate.

For a dual link communication the transmitter sends packets in two channels at the same time, with a delay of 1 instruction cycle (5 ns) for the second packet. Our implementation runs on the PRU and is based on a polling-base approach instead of interrupt-based approach.

### 4.3.2 Reception

The decoding process is specific to the encoding schedule used in our design. We need to first extract the raw samples from the ADC. After that, we run the decoding algorithm to recover the packets.

**Bit Detection.** Our decoding algorithm is based on sampling-based Manchester decoding [3]. We identify the rising and falling edges as bit 1 and bit 0 using an adaptive thresholding from preamble symbols. We first decode the

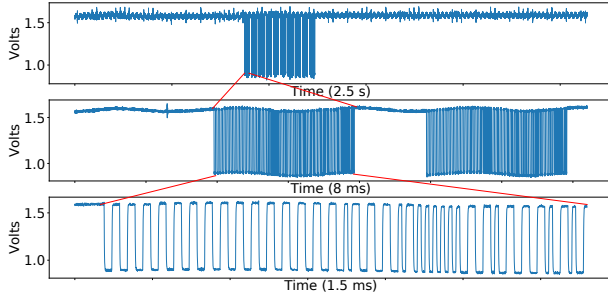


Figure 16. Raw data collected from onboard ADC.

Table 3. Propagation Delay Characterization

System Module	PRU Modulation	LED	PD	TIA	ADC	PRU Demodulation
Propagation Delay	20 $\mu$ s	5ns	5ns	10 $\mu$ s	2 $\mu$ s	20 $\mu$ s

raw samples into bit sequence. Then we use the predefined preamble to synchronize the packet to find the starting bit. Fig. 16 shows the general reception process. The top subfigure plots the reception of 10 packets. The middle subfigure shows the zoomed in version for one packets. The bottom subfigure presents details of the symbols for one packet.

#### 4.3.3 Communication Latency Analysis

We characterize the communication latency in our system from TX to RX by taking into account delays in different modules in the design. Fig. 1 shows the data flow from TX to RX based on half-duplex's setting. We send information from Linux Userspace to PRU through a character device. PRU receives the information and runs the encoding algorithm by toggling LEDs in a deterministic manner. Let's take an example of a symbol rate of 100 kHz. It takes one clock cycle (5ns) to change the LED status, it takes 10  $\mu$ s to transmit one symbol. We need two symbols to represent one bit. Thus we need 20  $\mu$ s to transmit one bit. Meanwhile, LED takes 2ns as a response for changing its status. Thus the processing delay on PRU modulation for one bit takes about 20  $\mu$ . On RX side, the symbols will be detected by the photodiode first, the response time for the photodiode takes 5ns [7]. The TIA will then amplify the detected symbols. The bandwidth of the transimpedance amplifier determines the upper bound of the transmission speed. In our design, we select a 2MHz bandwidth operational amplifier[8] which is adequate for the symbol rate we consider in our work. Once the signal is amplified by the TIA, it goes into ADC to be converted into digital samples. Based on the observation, we select ADS7783 as the ADC for our circuit since it has a sampling rate of 3 msp/s and previous approach has used the chip to sample signals with 1 msp/s using PRU [22] Thus, we suggest the symbol/clock rate should not be larger than 100 kHz. We summarize the delay at each module in table 3.

## 5 Evaluation

Now we describe various aspects of the system performance.

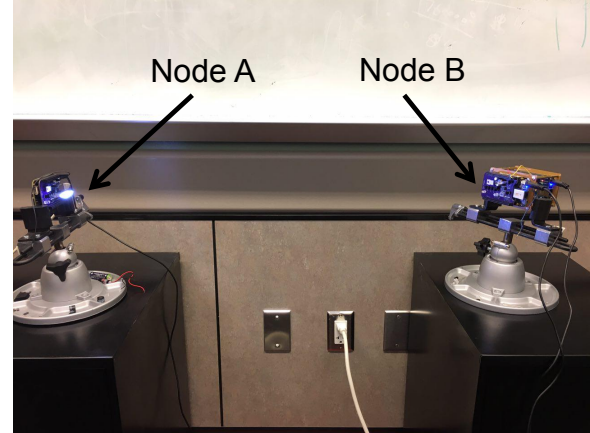


Figure 17. Experimental Setup

## 5.1 System Implementation

We connect Purple board to BeagleBone Black (BBB), which is a single-board computer, in order to assemble the full transceiver system. We implemented the firmware running on the PRU to send and receive packets. We load the firmware into the PRU core using Remoteproc framework on top of Debian Linux running on BBB. We use PRUDAQ, a high-speed ADC sampling board [6] to passively collect data from Purple board for performance analysis. Fig. 17 presents our experiment setup for the system evaluation. We use an assistant tool to set up the platform for experimentation. We implement the firmware running on PRU and make the data communication controllable in the embedded Linux environment running on the BBB. We then dump the data into a file for offline analysis. We evaluate the system in both office room and corridor with constant ambient light.

## 5.2 Metrics

We use the following three metrics to evaluate the system performance with the experimental data gathered from our testbeds.

**Packet Loss Ratio:** This metric (PLR) represents the failure rate for the link layer packets transmitted to the receiver. We use this metric to assess the quality of established visible light links.

**Bit Error Rate:** This metric (BER) represent the failure rate for the bits transmitted through the physical layer. It is used to evaluate the channel quality in the physical layer.

**Effective Throughput:** This metric represents the amount of the successfully received bits per second. We use it to quantify the PHY layer performance as this is a widely used experiment metric in previous studies [11, 34]. It is usually utilized as a metric at the level of the transport layer of the TCP/IP network model. However we use effective throughput to quantify the correctly received bits in the detected packets.

## 5.3 Single Link Throughput

We evaluate the performance using different number of LEDs on our platform for single link scenario.

*Experimental Setup:* We configure TX and RX to operate with the symbol rate of 100kHz. We first place TX and



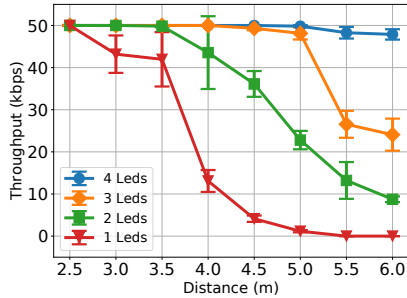


Figure 18. Throughput under various distances with different number of LEDs.

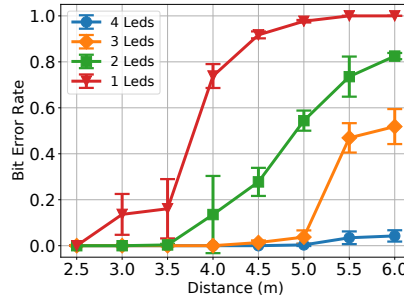


Figure 19. Bit Error Rate vs Communication Range using a symbol rate of 100kHz.

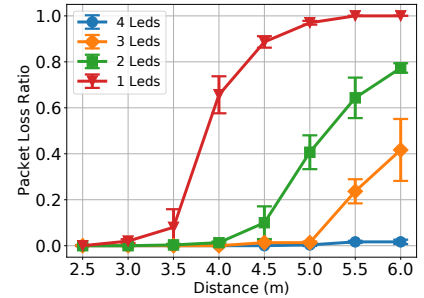


Figure 20. Packet Loss Rate vs Communication Range using a symbol rate of 100kHz.

RX nodes at a distance of 2.5 meters and run the experiment with 100 packets. Each packet consists of 1000 bytes. Then we increase the distance at 0.5 meter steps and repeat the experiment. At each distance, we repeat the experiment with different number of LEDs.

*Result:* Fig. 18 shows the communication performance using different number of LEDs. It is clear from the figure that more LEDs result in higher throughput under the same distance. With larger number of LEDs the generated illuminance is higher, thus more light signals can be detected by the photodetector, leading to a higher SNR. This observation indicates that adaptive transmission power is feasible using our system. Meanwhile, It is shown in Fig. 20 that we can achieve 99% reliability (PRR) when we are using 4 LEDs in our system with all distances up to 6 meters. For 3 LEDs, the reliability starts to drop at 5 meters. These can be verified through bit error rate in Fig. 19. The BER for our system with 4 LEDs is close to 0, while the BER with 1 LED is 100% with the communication range of 6 meters.

Fig. 19 shows that the system can have a working distance up to 6 meters with less than 1% BER. This is sufficient to validate the effective throughput is valid for indoor scenarios, especially for room area wireless communication. Meanwhile, it also suggests that the design can be scalable for the transmitter since more LEDs can be added to current design to increase the communication distance. It also suggests that multiple LEDs controlled by a cheap single-board computer are deployable immediately to replace current home light bulbs for both communication and illumination purpose. Our design can be tuned to contain both PDs and LEDs in a single package as the front-end, like the flashlight, making it a feasible solution to design smart bulbs that can send and receive data at the same time.

Fig. 20 shows the link quality for multiple transmitter evaluation. The packet loss ratio is less than 4% with a distance up to 6 meters. This is far better than state-of-the-art OpenVLC link quality demonstrated in [16], where the packet loss rate goes beyond 20% within 2 meters. With the achieved performance, it is now feasible to achieve room area networking. This demonstrates a highly reliable visible light link for data communication at a symbol rate of 100 kHz. This level of reliability at this symbol rate has not been

Table 4. Throughput and BER with different number of links.

Metrics	Single Link	Dual Link
Throughput (kbps)	49	95
BER	0.001	0.025
Packet Loss Ratio	0.0135	0.025

reported in embedded VLC literature so far. The link quality for embedded visible light communication can be as reliable as possible in a static scenario where both TX and RX does not require mobility.

#### 5.4 Concurrent Communication Performance

Now we study the communication performance when we use two concurrent channels.

*Experimental Setup:* We configure the transmitter to transmit packets on two links concurrently. The packets transmitted on each link are different. We use the polarizing setup described earlier (LED housing with polarizer at one end of the cylinder) to isolate the two channels. We then rotate the polarizing case so the two channels have opposite polarization (90 degrees). We place the two transmitters side-by-side (Fig. 8). We place TX and RX one meter apart and run the experiment with 100 packets for each link at a symbol rate of 100 kHz for each link. The system support simultaneous ADC sampling from different channels which enables us to dump the data from these two links simultaneously. We repeat the experiment with the same setup for single link to perform the performance comparison. The mutual interference between these two links can be canceled using the polarizers with crossed orientation. So we expect the effective throughput for concurrent channels to be twice the throughput for single link. Bit Error Rate for dual link and single link should be similar.

*Result:* Table 4 presents the system performance over different number of links. It shows that concurrent communication can achieve nearly twice the throughput of the single link indicating the effectiveness of the polarizer setup to isolate the channels. The BER for dual link is comparatively higher than with single link likely due to imperfect polarizer

and some cross-talk light that escapes the LED case we designed. Thus, the link layer for concurrent transmission has a higher packet loss ratio compared to single link. However, the overall link reliability for concurrent communication is 97.5%, which still validates that our design is efficient to cancel the mutual interference for concurrent communication. We also tried other approaches to cancel mutual interference and construct concurrent channels. For example, we used the red and blue LEDs as the dual link concurrent transmitter and the red and blue LEDs as the dual link concurrent LED receiver. Since the red and blue LEDs have different operating light spectrum, we expect them to be interference-free. But the transmitted signals can not be recognized by the proposed receiver design at this point. This validated that low cost indicator-type LED does not have the perfect matching in terms of responses to light bands. For example, we can receive blue light signals using red LED as the receiver, but we can't receive red light signals using blue LED receiver.

### 5.5 Full-duplex Throughput

*Experimental Setup:* In this section, we evaluate the full-duplex communication capability of Purple. We focus on the aggregate throughput gain compared to half-duplex system. We configure two nodes to transmit and receive packets simultaneously. We also ensure the packet transmission for the two nodes is overlapping. There will be two active flows at one time compared to half duplex. We run our experiment in a lab environment with normal lighting conditions. We configure the distance between the two node as 2.5 meters. We use aggregate effective throughput introduced in [12] as the metric here to evaluate the performance. For half duplex, we calculate the aggregate effective throughput by averaging the two single directional links. For full duplex, we calculate the aggregate effective throughput by adding the throughput for each direction when both links are active. Like evaluating full duplex in RF scenario [33], we ask ourself whether full duplex in visible light communication can double the throughput.

*Result:* We configure various distance between LED and PD to observe the impact on the visible light link reliability for both full duplex and half duplex. The result is shown in Fig. 21.

For half-duplex, the photodiode will receive the light signals transmitted from the other node, roughly at 50 kbps. This suggest the aggregate throughput with half-duplex design will not be affected by the distance between the LED and PD on the same board. However, it is interesting to see the aggregate throughput for full duplex design. In Fig. 21, the aggregate throughput is 0kbps when the LED and PD was placed 5mm. The onboard ADC will only accept the voltage between 0-2v. At this distance, the PD received too much light signals causing the amplified signal larger than the threshold the ADC can detect. The offline processing script will not detect any signals, causing the aggregate throughput to be 0. At 25mm, there is minimal interference on PD from the LED. At 15mm, the received signal for the photodiode will be including both the signal from the other transmitting LED and the onboard transmitting LED. As we

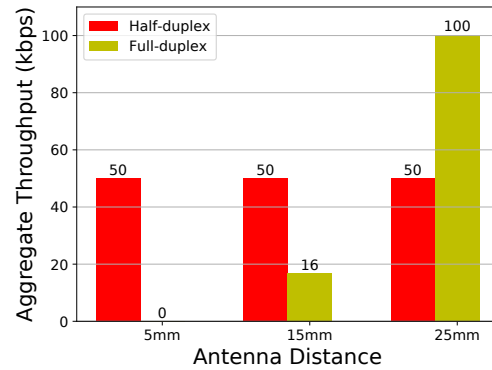


Figure 21. Aggregate throughput over various antenna distance.

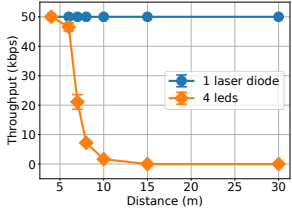
Table 5. Power consumption for one laser diode and one LED.

	Current (mA)	Voltage (V)	Power (mW)	Cost (\$)
Laser Diode	25	2.5	6.25	5.96
LED	21	2.9	6.19	0.44

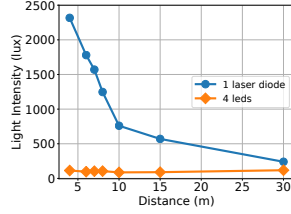
can see from Fig. 21, the aggregate throughput is 16kbps. The link is highly unreliable since most packet were dropped during the communication. We found that in average 92% and 62% packet loss ratio are for the two active links. We suggest board designers to place the LED and PD at least 25 mm apart to achieve the best full duplex performance in embedded visible light communication. We can further demonstrate quadruple performance boost with current design since we can configure the platform to enable full duplex for dual link communication. Then the potential gain can be 4 times over half duplex.

### 5.6 Comparison between Low Cost LED and Laser Diode

Next we study the VLC communication performance with low-power laser diodes vs one or multiple LEDs. Our platform also support off-the-shelf low power laser diode, such as 5mW or 1mW laser diode. We know the laser diode is energy efficient and more reliable with focused light beams, longer communication range compared to one single LED. Here is a table summarizing the power consumption between one LED and one laser diode when they are turned on. We configure Purple to use four LEDs and ran the experiment in a long corridor in an academic building. We repeat the experiment with low-power laser diode. Fig. 22 show the throughput between 4 meters and 30 meters for laser diode and 4 LEDs. It is expected that laser diode maintains the same reliability and effective throughput in all distances between 4 meters and 30 meters. However, the throughput drops significantly from 6 meters to 8 meters using 4 LEDs. In order to understand the impact from the light intensity received by the photodiode, we ran another experiment to collect the LUX value at these distances using a light sensor called TSL2561. We plot the LUX value in Fig. 23. It shows



**Figure 22. Throughput under various distances for laser diode and 4 leds.**



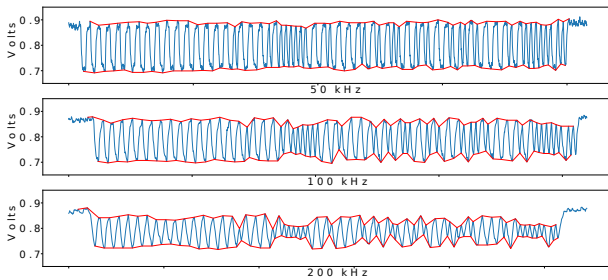
**Figure 23. Light Intensity measured under various distance.**

light intensity emitted from laser diode is higher compared to LEDs in shorter distance, such as 4 meters. It then goes down as distance increases. Light emitted from LEDs is less focused, leading to the receiver registering constant light intensity due to the ambient light in the corridor. The laser diode has a safety classification of IIIa, which is the same classification used in laser pointers for presentation purpose. It is safe, but it still requires careful handling[2]. Thus, we demonstrate the possibility of further increasing the performance of low-cost embedded VLC using low-power laser diodes while still remaining within the energy budget. These diodes require more careful handling than standard LEDs but are feasible to use in indoor spaces with people and useful to have as option to allow performance, power, and deployment constraint tradeoff for different applications.

### 5.7 Maximum Symbol Rate

We now study Purple’s performance bottleneck, particularly the stability of reception at high symbol rate. Symbol rate represents the clock rate used in our communication system. Since we are using OOK with Manchester coding, the bit rate for our system is half the symbol rate.

*Experimental Setup:* We configure TX and RX to be within a distance of 5 meters, We configure the TX to transmit packets at a symbol rate of 50kHz and then repeat the experiment for a symbol rate of 100kHz and 200kHz.



**Figure 24. Impacts of Symbol Rate on Reception.**

*Result:* Fig. 24 plots the received signal (one packet) sampled by the ADC when the system is configured using different symbol rate. It shows that the detected high and low symbols are stable (red line on the top subfigure) over a symbol rate of 50 kHz. The detected high and low symbols become

**Table 6. Standard Deviation for Local Peaks and Local Valleys.**

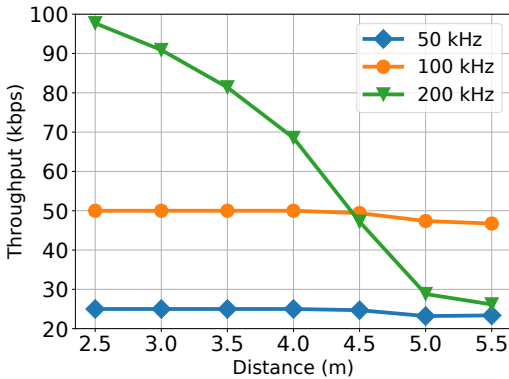
	50 kHz	100 kHz	200 kHz
<b>Std. for Peaks</b>	0.758%	1.265%	2.104 %
<b>Std. for Valleys</b>	0.887%	1.561%	2.213 %

more fluctuated when the symbol rate goes higher. This is due to the limitation of the TIA design. We selected a bandwidth of 2.2MHz operational amplifier (two dollars each) to serve as the TIA. The amplified signal will be distorted when the symbol rate goes from 100kHz to 200kHz. For 200kHz, the distortion is the worst. Thus it is extremely difficult to detect the fluctuated signal. While for 100kHz, it is operating as expected, which makes it an ideal symbol rate for long range experiments regarding to the performance. This also explains why the designed system performs well in terms of various symbol rate in Fig. 25. Effective throughput is expected to be the maximum at symbol rate of 50 kHz and 100 kHz. However, it drops significantly for longer distance with a symbol rate of 200 kHz. We quantify the standard deviation of the local peaks and local valleys to show the fluctuation in Table.6. It indicates that the amplified signal with 200 kHz symbol rate is mostly unstable. We can further improve current TIA design with high speed operational amplifier. However, that also brings the trade-off consideration in terms of high speed circuit design and system complexity.

## 6 Discussion

Interacting with low-level interfaces and processing data in a real-time manner is challenging when it comes to multitasking. Researchers and system designers have developed different techniques to maintain co-existence of time-critical and non-time-critical tasks in a single-processor system. A common technique is to offload I/O and data transfer tasks from the main processor to specially designed interfaces/processors so the tasks can be accomplished independently. Low-level digital communication interfaces such as I2C and SPI can be utilized to send bits to be transferred by IO pins. DMA (Direct Memory Access) allows such interfaces to access to main memory without involving main processor in the actual data transfer. Further, in some systems there are co-processors (like PRU in AM335x) allows a completely independent process to access the memory and peripherals in real time, which is what we used in this paper.

In order to compare these two options, we have implemented a version of the platform that uses DMA to drive the pins that power the LEDs. We have found that this software architecture to be more complex and less performant compared to the PRU-based approach. The complexity comes from having to switch between RX and TX even while using DMA. On the receiver side, it is necessary for VLC system to do error correction etc, so it is not efficient to copy all the digital bits from LED directly to the memory using DMA. DMA-based approach can be made to work: we can add some hardware near the VLC frontend so the cleaned bits can be directly copied to the memory using DMA. Although DMA can move data efficiently, it can not handle any data



**Figure 25. Throughput under various distances with different symbol rates for 4 LEDs.**

processing related tasks, such as modulation and demodulation. On the other hand, using PRU-based approach, we are able to provide such low-level processing before sending the data to the main processor. Hence, it significantly reduces CPU overhead to process the data. The platform architecture with one main processor and a small number of IO-processors is also general to other IoT platforms so the work has the potential to be applicable on other platforms. PRU provides both processing and data transfer functionalities. On more traditional platforms without PRU, we may be able to build Purple-like systems by combining MCUs (processing) and DMA (data transfer).

## 7 Conclusion

We designed and implemented Purple VLC: a novel embedded VLC platform that integrated I/O offloading, concurrent channels with polarized light and full-duplexing to significantly improve the performance for low cost embedded VLC platform. The results indicate that we can achieve an aggregate throughput up to 100 kbps given an operating distance for up to 6 meters. Our design combines I/O offloading, concurrent channels and full-duplexing to offer more than 99% link reliability. We expect more reliable and useful applications to be developed based on our platform. We believe this can be the complimentary technology other than RF for wireless connectivity in room area.

## 8 References

- [1] Am335x technical reference manual. <http://www.ti.com/lit/ug/spruh730/spruh730.pdf>.
- [2] Laser diode 5mm 650nm red. <https://www.adafruit.com/product/1054>.
- [3] Manchester coding basics application note. <http://www.atmel.com/images/atmel-9164-manchestercodingbasicsapplication-note.pdf>.
- [4] Openvnc. <http://www.openvnc.org>.
- [5] Pru software supported package. <http://www.ti.com/tool/pru-swpkg>.
- [6] Prudaq repository. <https://github.com/google/prudaq/wiki>.
- [7] Sfh213 datasheet. <http://www.farnell.com/datasheets/1672048.pdf>.
- [8] Tlc2274 datasheet. <http://www.ti.com/lit/ds/symlink/tlc2274.pdf>.
- [9] I. S. Association et al. Ieee std. for local and metropolitan area networks-part 15.7: Short-rang wireless optical communication using visible light. *IEEE Computer Society*, 2011.
- [10] R. Birkett and L. L. Need hard real time. Enhancing real-time capabilities with the pru. 2014.

- [11] C.-L. Chan, H.-M. Tsai, and K. C.-J. Lin. Poli: Long-range visible light communications using polarized light intensity modulation. In *MobiSys '17*. ACM, 2017.
- [12] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *MobiCom '10*. ACM, 2010.
- [13] P. Dietz, W. Yezazunis, and D. Leigh. Very low-cost sensing and communication using bidirectional leds. In *Ubicomp'03*. Springer, 2003.
- [14] W. Du, J. C. Liando, and M. Li. Softlight: Adaptive visible light communication over screen-camera links. In *INFOCOM '16*. IEEE, 2016.
- [15] K. Hewage, A. Varshney, A. Hilmia, and T. Voigt. modbulb: A modular light bulb for visible light communication. In *VLCS '16*. ACM, 2016.
- [16] M. Heydariaan, S. Yin, O. Gnawali, D. Puccinelli, and D. Giustiniano. Embedded Visible Light Communication: Link Measurements and Interpretation. In *MadCom '16*. ACM, 2016.
- [17] Y.-H. H. James Davis and H.-C. Lee. Humans perceive flicker artifacts at 500hz. In *Scientific Report*, Feb. 2015.
- [18] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban. Led based indoor visible light communications: State of the art. *IEEE Communications Surveys Tutorials*, 17(3):1649–1678, 2015.
- [19] L. Klaver and M. Zuniga. Shine: A Step Towards Distributed Multi-Hop Visible Light Communication. In *MASS '15*. IEEE, 2015.
- [20] S. Li, A. Pandharipande, and F. M. J. Willems. Unidirectional visible light communication and illumination with leds. *IEEE Sensors Journal*, 16(23):8617–8626, Dec 2016.
- [21] S. Li, A. Pandharipande, and F. M. J. Willems. Two-way visible light communication and illumination with leds. *IEEE Transactions on Communications*, 65(2):740–750, Feb 2017.
- [22] D. Molloy. *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Wiley, 2014.
- [23] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross. Linux light bulbs: Enabling internet protocol connectivity for light bulb networks. In *VLCS '15*. ACM, 2015.
- [24] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. Led-to-led visible light communication networks. In *MobiHoc '13*. ACM, 2013.
- [25] S. M. Schmid. *Software-Defined Low-Complex Visible Light Communication Networks*. PhD thesis, ETH Zurich, 2016.
- [26] I. Tamzeed, I. Bashima, and N. Shahriar. Soundsifter: Mitigating over-hearing of continuous listening devices. In *MobiSys '17*. ACM, 2017.
- [27] Z. Tian, K. Wright, and X. Zhou. The darklight rises: Visible light communication in the dark. In *MobiCom '16*. ACM, 2016.
- [28] A. Varshney, A. Soleiman, L. Mottola, and T. Voigt. Battery-free visible light sensing. In *VLCS '17*. ACM, 2017.
- [29] J. Vucic and K.-D. Langer. High-speed visible light communications: State-of-the-art. In *Optical Fiber Communication Conference*. Optical Society of America, 2012.
- [30] Q. Wang. *Visible light and device-to-device communications: system analysis and implementation*. PhD thesis, 2016.
- [31] Q. Wang, D. Giustiniano, and O. Gnawali. Low-cost, flexible and open platform for visible light communication networks. In *HotWireless '15*. ACM, 2015.
- [32] Q. Wang, S. Yin, O. Gnawali, and D. Giustiniano. Demo: Open-VLC1.0 Platform for Research in Visible Light Communication Networks. In *MobiCom '15*. ACM, 2015.
- [33] X. Xie and X. Zhang. Does full-duplex double the capacity of wireless networks? In *INFOCOM 2014*. IEEE, 2014.
- [34] Y. Yang, J. Hao, and J. Luo. Ceilingtalk: Lightweight indoor broadcast through led-camera communication. *IEEE Transactions on Mobile Computing*, 2017.
- [35] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *MobiSys '15*. ACM, 2015.
- [36] S. Yin and O. Gnawali. Towards embedded visible light communication robust to dynamic ambient light. In *GlobeCom '16*. IEEE, 2016.
- [37] S. Zhai, L. Guo, X. Li, and F. X. Lin. Decelerating suspend and resume in operating systems. In *HotMobile '17*. ACM, 2017.
- [38] C. Zhang and X. Zhang. Litell: Robust indoor localization using unmodified light fixtures. In *MobiCom '16*. ACM, 2016.