Delft University of Technology
Master's Thesis in Embedded Systems

Co-design of Smart Lighting and Communication for Visible Light Networks

Hongjia Wu

embedded software

TU Delft Delft University of Technology

Co-design of Smart Lighting and Communication for Visible Light Networks

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Hongjia Wu
H.Wu-7@student.tudelft.nl

21st May 2017

Author
Hongjia Wu (H.Wu-7@student.tudelft.nl)
Title
Co-design of Smart Lighting and Communication for Visible Light Networks
MSc presentation
31st May 2017

Graduation Committee

Prof. dr. K.G. Langendoen (Chair)	Delft University of Technology
dr. Marco Zuniga (Daily Supervisor)	Delft University of Technology
dr. Claudia Hauff	Delft University of Technology

Abstract

Visible Light Communication has been intriguing both the industry and the academia for over a decade. However, most of the research efforts in this area only consider scenarios where the light emitted from the transmitters is constant. This is *not* true when **Smart Lighting**, a technology designed for more comfortable and energy efficient buildings by adjusting the illumination of fixtures based on ambient light availability, is involved. Under the smart lighting, the changes of fixtures' illumination affect the communication links significantly. Therefore, new methods must be proposed for a revolutionary co-design of smart lighting and communication for the visible light networks.

This project proposes **SmartVLC**, a system that can optimize the throughput (benefit communication) while maintaining the LEDs' primary illumination function (benefit smart lighting). A new modulation scheme Adaptive Multiple Pulse Position Modulation (AMPPM) is proposed to support fine-grained dimming levels and optimize the throughput under each dimming level via pattern multiplexing. An algorithm is also designed to adjust the LEDs' dimming level smoothly. The proposed AMPPM and the LEDs' illumination adaptation algorithm are implemented into a low-cost platform. The performance of SmartVLC is evaluated under both static and dynamic scenarios. The current version of SmartVLC supports a communication distance of **3.6 m** and a throughput of **107 Kbps** in an indoor environment. Compared to state-of-the-art solutions, the performance gain of SmartVLC can reach up to **190%**.

Simplicity is the ultimate sophistication

Leonardo da Vinci

天下難事, 必作於易; 天下大事, 必作於細

老子

Preface

This thesis is submitted for the degree of Master of Science at the Delft University of Technology. This work is to the best of my knowledge original, except where references are made to the previous work.

Time flies. It has been roughly eight months since I have started this project. The completion of this project is approaching. Now it is the time to express my thanks for those who have helped and guided me through the progress of this project.

Foremost, I would like to express my sincere gratitude to my daily advisor, Dr. Qing Wang, for his patience, enthusiasm, and immense knowledge. His guidance helped me in all the time of research, writing of this thesis and preparation of the presentation. I could not have imagined having a better advisor and mentor for my thesis. I would also like to thank my direct advisor Dr. Marco Zuniga for introducing me into this fascinating VLC project. I appreciate him for giving us solid and unfailing feedbacks in both research problems and presentation skills. This project could not be finished without his generous support. I also want to thank Prof. Koen Langendeon for admitting me into this amazing Embedded Software group and insightful comments during our VLC group meetings. Special thanks go to Dr. Claudia Hauff for being a committee member of my thesis defense. I would also like to thank Eric Wang and Koen Schaper for their support on the technical side, especially on soldering electronic instruments. Last but not least, I would specially like to thank my parents and my girlfriend Mengdi Liu. Their encouragements, love, and support have fueled me with confidence. I would also like to thank all my friends in the Netherlands. We had a lot of good times together.

Hongjia Wu

Delft, The Netherlands
21st May 2017

Contents

Chapter 1

Introduction

Nowadays, 19% of the global energy consumption is for lighting. Electricity consumption in lighting is also projected to increase by 60% over the next 25 years, which is equivalent to a power consumption of over 4250 TWh [46]. Consequently, the necessity for energy saving in lighting is vital. This can be obtained via **smart lighting**, a technology that is designed for more productive, comfortable, and energy efficient buildings by adjusting the illumination of fixtures based on circumstances such as occupancy or daylight availability [14]. *In smart lighting enabled systems, light emitted from the illumination fixtures is expected to change with ambient light conditions, to keep the total illumination constant within an area-of-interest.* For instance, a shiny day offers sufficient ambient light to satisfy the basic illumination requirement. Therefore, less artificial light is demanded. In a cloudy day, however, more artificial light is required. Through smart lighting, the energy consumption of lighting can be reduced significantly [13, 27]. Besides, a fixture with illumination adjusting function produces less heat, which extends its lifespan and reduces HVAC[1] cooling loads.

Meanwhile, **Visible Light Communication (VLC)** has been intriguing both the industry and academia for the past decade, mainly due to the huge license-free spectrum of visible light. VLC refers to a type of short-range optical wireless communication adopting the visible light spectrum from 380 to 780 nm that is ten-thousand times wider than the Radio Frequency (RF) spectrum. In VLC, data is transmitted by modulating optical sources, such as Light Emitting Diodes (LEDs) [15, 20]. Since these optical sources can be turned on and off rapidly (in the order of million times per second), it has no impact on the illumination light perceived by human's eyes, but can be sensed by light sensors, as illustrated in Fig. 1.1. In this way, *VLC adds data communication as an additional functionality to illumination fix-*

[1]HVAC (Heating, Ventilation, and Air Conditioning) is the technology that provides thermal comfort in indoor and vehicular environments.

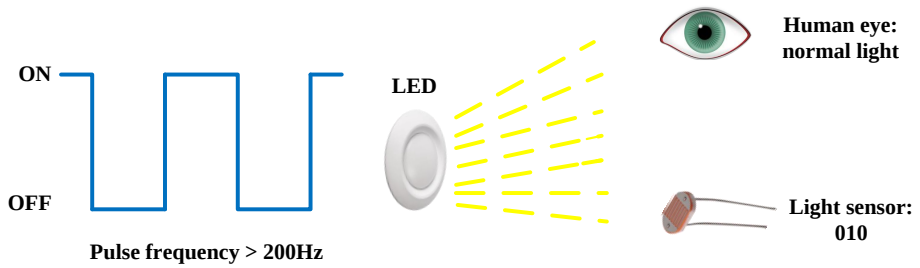


Figure 1.1: Human eyes cannot perceive the flickering of LED lights if a LED is modulated at frequencies higher than 200 Hz [12]. However, the light changes can be sensed by photosensors. VLC exploits this phenomenon for wireless communication beyond the illumination purpose of LED lights.

tures and introduces minimal deployment overhead (only attaching low-cost modulation units to existing LED bulbs). Besides, due to the physical characteristic of visible light, it cannot penetrate walls and hence guarantees the security of the wireless communication link. Another appealing advantage of VLC is that it can be established on the ubiquitous and dense LED lighting infrastructure in our life, which supports highly dense concurrent VLC links because of LED bulbs' directionality [21, 42, 48].

1.1 Problem statement

Most of the state-of-the-art VLC systems consider scenarios where the light emitted from transmitters (namely, LEDs) is constant [17, 22, 34]. This is *not* true in systems where smart lighting is involved. To support smart lighting, several modulation schemes with dimming support have been proposed, e.g., On-Off-Keying (OOK) modulation with compensation time [7, 16, 31], Multiple Pulse Position Modulation (MPPM) [23, 24, 35, 37], Overlapping PPM [10, 11, 38], etc. However, with smart lighting enabled, the changes in light intensity affect the VLC link quality significantly under these modulation schemes. For example, Fig. 1.2 shows the performance degradation of VLC links under the OOK modulation with compensation time (obtained from our experiments). Therefore, new schemes must be proposed for a better co-design of smart lighting and communication for visible light networks.

Motivated by the above demand, this thesis targets at solving the following *research problem:*

> **In a joint smart lighting and VLC system, how to maximize the system throughput while maintaining the LEDs' primary illumination function (i.e., LEDs are free to change their illumination and are flickering-free for users)?**

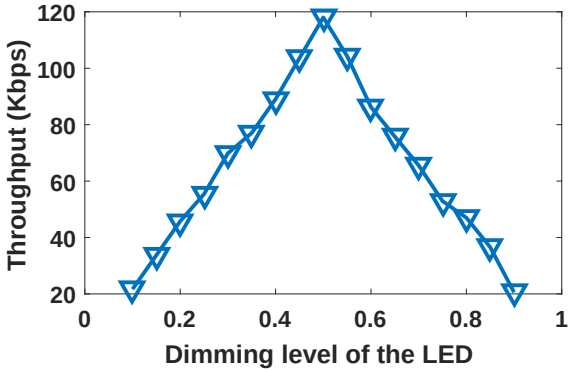


Figure 1.2: The system throughput under the OOK modulation with compensation time to support dimming

The above research problem can be mapped to the following requirements:

- The LED should support enough dimming levels so the human beings cannot perceive a strong flickering when the LED desires to modify its current illumination smoothly.
- The LED's emitted light should be changed according to ambient light changes. The sum of the ambient light and LED light should be kept constant within the area-of-interest.
- The throughput should be maximized under each dimming level of the LED. By doing this, the degradation of the system throughput can be minimized when the LED changes its illumination.

1.2 Contributions

This thesis proposes **SmartVLC**, a system that co-designs **Smart** lighting and **VLC** for visible light networks. The mechanisms designed in SmartVLC can solve the research problem presented in Section 1.1 perfectly. The contributions of this thesis are summarized as follows:

- A new modulation scheme Adaptive Multiple Pulse Position Modulation (AMPPM) is proposed. The AMPPM enables the system to possess fine-grained dimming levels and maximize the system throughput under each dimming level via pattern multiplexing and slope-based pattern selection.
- An algorithm is proposed to adjust the LED's dimming level according to ambient light changes and by considering the nonlinear reaction of human's eyes to different illumination.

- Front-end optical transceivers are built with off-the-shelf LEDs, photodiodes and other electronic devices. The proposed AMPPM and the LED's illumination adaptation algorithm are implemented using the low-cost platform BeagleBone Black (BBB). High computation tasks (e.g., sampling) are implemented using the BBB's built-in microcontrollers.

- The performance of SmartVLC is evaluated under both static and dynamic scenarios. The current implementation of SmartVLC supports a communication distance of **3.6 m** and a throughput of **107 Kbps** in an indoor environment. Compared to state-of-the-art solutions, the performance gain of SmartVLC can reach up to **190%**.

1.3 Organization

The rest of this thesis is organized as follows: background information and related work are provided in Chapter 2. The proposed SmartVLC is presented in Chapter 3. Details of the system implementation are given in Chapter 4, followed by the experimental evaluations presented in Chapter 5. Finally, conclusions and future work are summarized in Chapter 6.

Chapter 2

Background

This chapter first presents some background knowledge of VLC and then introduces the state-of-the-art modulation schemes that support dimming for smart lighting. Advantages and disadvantages of these modulation schemes are summarized and compared briefly to the one proposed in this thesis.

2.1 Background on VLC

During the past decade, VLC has been exploited in many application scenarios, such as high-speed communication [17, 30], indoor localization [21, 25, 43], human sensing [26, 47], screen-to-camera communication [18, 29], vehicle-to-vehicle communication [41, 44, 45], and so on.

Independent from different application scenarios, a typical VLC system mainly consists of two components: a transmitter and a receiver. The *transmitter* normally modulates the light emitted from LEDs to send data wirelessly. The *receiver* usually chooses one from the following three types of photosensors: *1) Photodiode*, a semiconductor device that converts light intensity into current. Off-the-shelf photodiodes can be operated to sample signals at rates of tens of MHz. *2) Imaging sensor or a camera*, which can be used to receive modulated light via its rolling shutter effect. Since it is available in most of the today's mobile devices, it reduces the cost of building new VLC receivers. However, it only supports limited throughput (around few Kbps) due to its low sampling rate. It also consumes much more energy than a photodiode. *3) LED*, that can act as a photosensor if it is operated in reverse bias mode. However, an LED is not a good receiver when compared to a photodiode, because it is normally less sensitive and also has a limited field of view due to its shape and size.

The block diagram of a typical VLC system is given in Fig. 2.1, where only the single-direction link from the transmitter to receiver is presented,

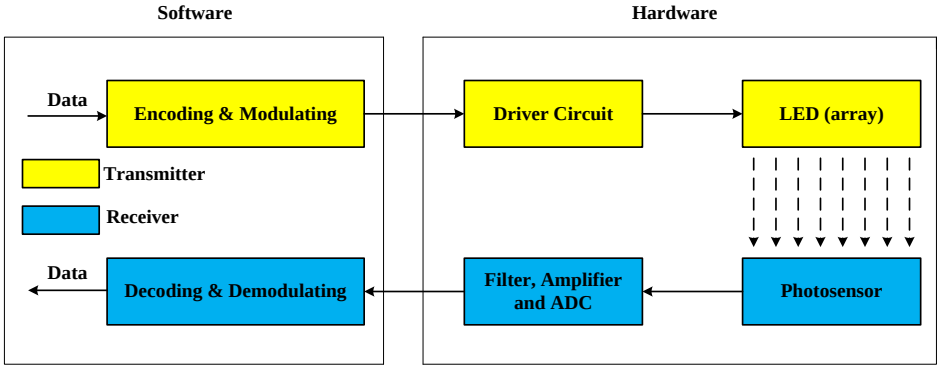


Figure 2.1: The block diagram of a typical VLC system

for simplicity. Both the transmitter and receiver are composed of software and hardware. The software of a transmitter is mainly responsible for data encoding and hardware control. After modulation, the driver circuit in the hardware controls the LEDs (e.g., by turning them on and off rapidly) to transmit the modulated signal. At the receiver, the light signals sensed by the photosensor are filtered, amplified, and converted from analog signals to digital signals. The converted signals are then decoded by the decoder in the software part of the receiver.

State-of-the-art implementations of the VLC systems can be classified into two categories: *1) low-end* testbeds with commodity hardware, and *2) high-end* testbeds with software defined radios. The low-end testbeds [19, 34, 40] allow rapid development using off-the-shelf low-cost hardware, e.g., BeagleBone Black [2], Raspberry Pi [3], or Arduino [1]. However, the performance of these low-end testbeds is limited by the computation capabilities of the hardware, where only simple modulation schemes such as OOK can be implemented. This makes them more suitable for applications that do not require high data rate. The high-end testbeds are suitable for data intensive applications where advanced modulation schemes are adopted, such as ACO-OFDM (Asymmetrically Clipped Optical Orthogonal Frequency Division Multiplexing) and DCO-OFDM (DC-biased Optical Orthogonal Frequency Division Multiplexing). The representatives are the WARP platform [5] and the combination of USRP and GNU Radio [4, 6]. The disadvantage of high-end testbeds is that the hardware cost is relatively high, usually in the order of thousands of US dollars.

Overview on the modulation schemes in VLC

The most striking difference between VLC and RF is that in VLC, the data can only be encoded in the domain of light intensity, due to the incoherent output of LEDs [39]. Therefore, a VLC system usually uses Intensity

Modulation and Direct Detection (IM/DD), in contrast with the complex valued and bi-polar signals in RF systems. Another requirement in VLC is that it should not cause human-perceivable fluctuations in the lightness from the LEDs (namely, *flickering*). It was shown that flickering can cause serious detrimental physiological changes in human eyes [12]. For this reason, the IEEE 802.15.7 standard [7] suggests that the change in light intensity should be performed at frequencies higher than 200 Hz.

Conjunct with the exclusive requirements of VLC system, the modulation schemes can be classified into two categories: *single carrier modulation* and *multi-carrier modulation*. The single carrier modulation schemes usually encode bits into pulse presence (e.g., OOK [34, 40]), pulse width (e.g., Pulse Width Modulation, PWM [36]), pulse position (e.g., Variable Pulse Position Modulation, VPPM [7]), etc. The multi-carrier modulation schemes, e.g., Orthogonal Frequency Division Multiplexing (OFDM) [8, 9], divide the channel into multiple orthogonal subcarriers and send the modulated data in parallel subcarriers so that the system can achieve higher data rate. This type of modulation requires more complicated hardware due to its demand on high computation capability.

2.2 Related modulation schemes

To support smart lighting that requires changing the lightness of LEDs in a wide range, some modulation schemes are enhanced/proposed to enable the dimming control of LEDs in VLC. State-of-the-art dimming methods can be categorized as two types: *analog dimming* and *digital dimming* [16, 46]. The difference between them is shown in Fig. 2.2. The analog dimming increases/decreases the LEDs' lightness directly by adjusting the forward current through the LEDs, which is simple to implement. However, directly changing the LEDs' forward current causes corresponding change in CCT[1], consequently, causing color shift [7, 46]. Digital dimming is usually known as PWM based lightness adjusting where the duty cycle is changed to achieve the target lightness of LEDs. Next, I introduce four state-of-the-art digital dimming methods that are tightly related to the work of this thesis:

OOK with time compensation: In the OOK modulation, data bits 1 and 0 are represented via turning the LED on and off, respectively. The advantage of OOK is its ease of implementation. It is always incorporated with Manchester coding to avoid maximum run length limitation and synchronization issue. Manchester coding embeds data into the clock by denoting a bit '0' as "OFF-ON" and a bit '1' as "ON-OFF", thus is DC-balanced.

[1]The Correlated Color Temperature (CCT) is a specification of the color appearance of the light emitted by a lamp, relating its color to the color of light from a reference source when heated to a particular temperature, measured in degrees Kelvin (K).

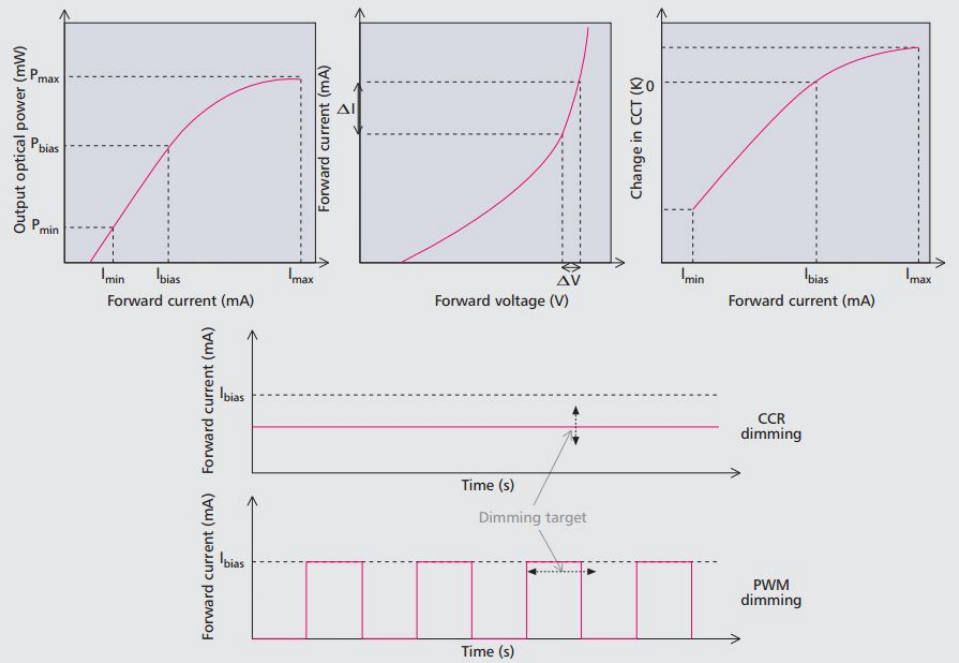
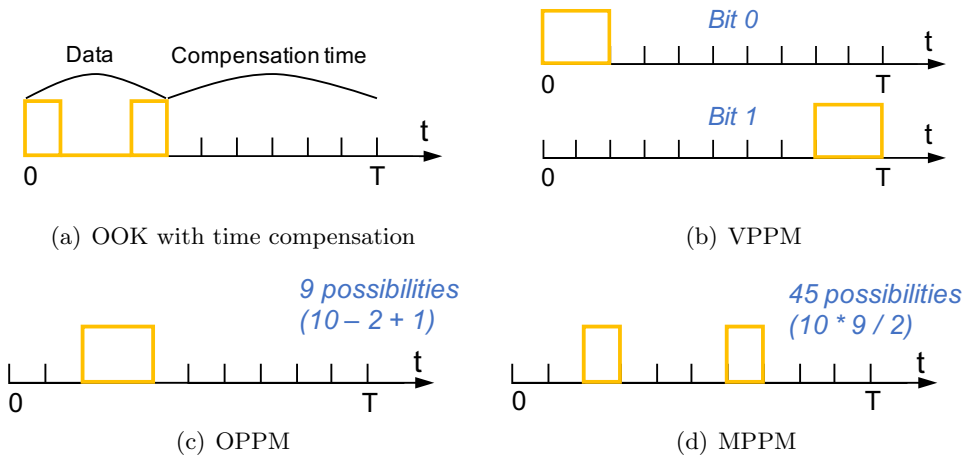


Figure 2.2: The difference between analog dimming and digital dimming. The analog dimming changes the LED lightness via modifying the forward current through LED, which may causes changes in correlated color temperature (CCT). The digital dimming changes the LED lightness via varying the duty cycle [46].

To support dimming for smart lighting, modulated data is appended with compensation time [7, 16, 31], as shown in Fig. 2.3(a). The modulated data from OOK has its own duty cycle (equivalent to the corresponding lightness) depending on the specific content of the data. If OOK is combined with Manchester coding to avoid flickering, then the lightness is fixed to 50%. To provide more dimming levels, compensation time can be added to the modulated data. During the compensation time, the "dummy data" is consecutive "ON" or "OFF", depending on the dimming target. After the compensation time, a sync point is usually inserted to avoid the loss of synchronization. In this way, the system can reach any dimming level simply by varying the length of compensation time. This is the *advantage* of providing dimming support through OOK with compensation time. The *disadvantage* is that the system loses its throughput dramatically if the target dimming level[2] is far away from 0.5, as shown in Fig. 1.2.

VPPM (Variable Pulse Position Modulation): In VPPM [7, 23, 31],

[2]In this thesis, **the range of the dimming level is set to** $[0, 1]$**.** The dimming levels of 0 and 1 mean the LED is completely turned off and turned on, respectively. A dimming level of 0.5 means the duty cycle of the LED is set to 50%.



(a) OOK with time compensation

(b) VPPM

(c) OPPM

(d) MPPM

Figure 2.3: State-of-the-art modulation schemes that support dimming

the data bits 1 and 0 are represented by the light pulse leaning to either left or right side of each *symbol slot* T. It is a hybrid of PPM [15] and PWM [36]. The pulse width depends on the required dimming level. When the dimming level is 0.5, then VPPM is similar to 2-PPM. Since within a symbol slot T, the pulse is only used to represent a single bit (either 1 or 0) independent of the pulse width, the throughput achieved by VPPM is always constant under different dimming levels. The light pulse width can be adjusted to satisfy different dimming level requirements. For example, Fig. 2.3(b) shows the waveform of how VPPM attains a 0.2 dimming level requirement, where the pulse widths of both bit 0 and bit 1 are 20% of the symbol slot T.

However, the supported dimming levels are constricted by the number of slots in each symbol slot T. Thus the *disadvantage* of VPPM is that under discrete dimming levels, the smart lighting system may not be smooth enough to eliminate user's perception of flickering. Meanwhile, the throughput is relatively low, because, in each symbol slot T, it can only represent one binary bit under any dimming levels.

OPPM (Overlapping Pulse Position Modulation): In OPPM [10, 11, 38], data is represented by consecutive pulses in a symbol slot T, as illustrated in Fig. 2.3(c). For example, if each symbol slot T has ten slots and the current required dimming level is 0.2, then two consecutive time slots will be occupied. Therefore, each symbol slot T can represent nine different possibilities, which can be mapped to three binary bits (i.e., $\lfloor \log_2 9 \rfloor$). Compared to VPPM, OPPM can achieve higher throughput but the supported number of dimming levels is also restricted by the number of time slots in each symbol slot T.

MPPM (Multiple Pulse Position Modulation): In MPPM [23, 24,

Table 2.1: Comparison of the state-of-the-art modulation schemes that support dimming

	High throughput	**Low throughput**
High resolution (more dimming levels)	***This thesis:*** AMPPM (Adaptive Multiple PPM)	OOK with compensation time
Low resolution (less dimming levels)	MPPM / VR-MPPM	OPPM; VPPM

35, 37], light pulses do not have to be consecutive compared to the OPPM. Therefore, under a fixed dimming level, MPPM can represent more possibilities than OPPM, as shown in Fig. 2.3(d). *This means MPPM can achieve higher throughput.* Since MPPM is also pulse position based, the supported dimming levels are also restricted by the number slots in each symbol slot T, as in OPPM and VPPM. Recently, the authors in [35] rename MPPM as Variable Rate MPPM (VR-MPPM) because of its variable data rate under different dimming levels. They also analyze the tradeoff between data rate and supported dimming levels. The details of MPPM (VR-MPPM) will be elaborated in the next chapter.

To support smart lighting, a VLC system should be able to provide enough dimming levels (*high resolution of dimming levels*), so the switching between different levels will not cause the users to observe flickering. A system with fine-grained dimming level control can also make the system more energy efficient and easier to cooperate with the ambient light changes. Besides, the *achievable throughput* under each dimming level should be maximized to reduce the performance degradation. Considering these two metrics (resolution of dimming levels and achievable throughput), none of the state-of-the-art solutions satisfy the requirement of high resolution and high throughput at the same time, as concluded in Table 2.1.

To solve this problem, AMPPM is proposed in SmartVLC in this thesis. AMPPM can achieve high resolution and high throughput concurrently. The details will be presented in the following chapters.

Chapter 3

SmartVLC

This chapter presents the proposed **SmartVLC**, a system that co-designs **Smart** lighting and **V**isible **L**ight **C**ommunication.

3.1 System overview

The system architecture of SmartVLC is shown in Fig. 3.1. As in a typical VLC system, SmartVLC is mainly composed of two types of devices: access point and mobile nodes. In the current design of SmartVLC, only the single communication link from the access point to the mobile node is considered. Therefore, in the rest of this thesis, the access point and the mobile node are simply referred as the *transmitter* and the *receiver*, respectively.

Recall that SmartVLC co-designs smart lighting and communication. *For the smart lighting*, SmartVLC aims at providing a constant illumination condition within the area-of-interest. This is done by adjusting the light emitted from LEDs according to ambient light changes. To sense the ambient light, the transmitter is equipped with a light sensor, as presented in Fig. 3.1. *For the communication*, SmartVLC targets at optimizing the throughput of the system under different dimming levels, to improve the whole system's performance of providing data connections. To achieve this goal, SmartVLC adopts the new modulation scheme AMPPM that is proposed in this thesis.

In SmartVLC, the transmitter works as follows:

1. Upon receiving the data from upper layers, the transmitter first reads ambient light from the light sensor. Based on the illumination requirement of the area-of-interest, the transmitter calculates the dimming level of the LED, i.e., the brightness of the light that should be emitted from the LED. The *sum* of the ambient light and the light emitted from the LED should be maintained constant, which is the key requirement of smart lighting.

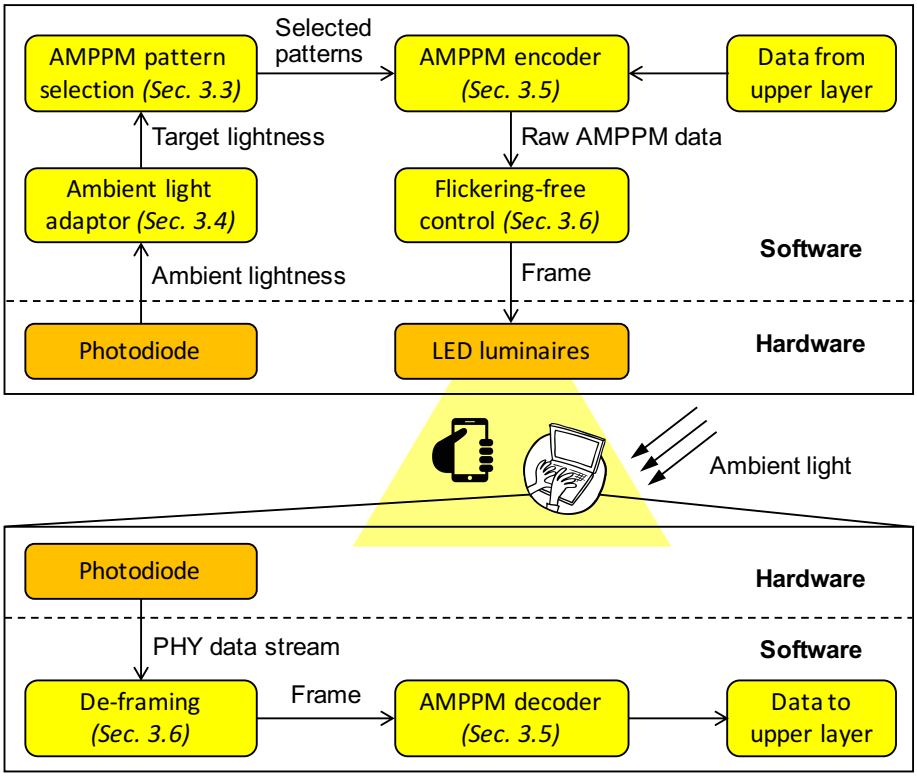


Figure 3.1: The system architecture of SmartVLC

2. After fixing the dimming level, the data from upper layers is modulated using the proposed AMPPM modulation scheme.
3. The modulated data is inserted into the payload of frames. To avoid flickering to users, each frame header is adjusted based on the targeted dimming level of the LED.
4. The frames are transmitted via visible light by modulating the LED.

At the receiver, light signals are detected by the photodiode. The receiver first decodes the header of each frame to obtain the current configuration of the AMPPM scheme (which is adaptive to ambient light changes). Based on that, the receiver decodes the payload to extract the data and then submit the data to upper layers.

The *main properties* of SmartVLC can be listed as follows (the details are presented in the rest of this chapter):

- A multiplexing approach is proposed to provide fine-grained dimming levels. (Section 3.2)
- The novel modulation scheme AMPPM is proposed to optimize the throughput at any given dimming level. (Section 3.3)

- The transmitter adapts smoothly its emitted light based on the changes in ambient light. The smooth adaptation is done by an algorithm that considers the nonlinear brightness response of the human being's eyes. (Section 3.4)
- A communication protocol and the frame format are designed to enable the flickering-free communication between of transmitter and receiver. (Section 3.6)

3.2 Support fine-grained dimming levels

Before presenting how to support fine-grained dimming levels in SmartVLC, I first give several definitions (some of them are illustrated in Fig. 3.2):

- **Time slot**: denoted as t_{slot}. It depends on the system implementation. For a specific system, t_{slot} is *fixed*. It is the *meta* time duration at which the system can control the hardware to perform tasks, such as turning on/off the LED to transmit data, or do the sampling.
- **ON/OFF state**: obtained via turning on/off the LED for each time slot t_{slot}. In SmartVLC, the states `ON` and `OFF` are the two *meta* states.
- **Pulse width**: the time when the LED is consecutively `ON`. Depending on the modulation, there can be one or several consecutive `ON` state(s) in a pulse width. For example, the pulse width equals to t_{slot} under the modulation schemes OOK and MPPM, as illustrated in Fig. 2.3(a) and (d); and equals to $2 \times t_{\text{slot}}$ under VPPM and OPPM when the duty circle of LED is 20%, as illustrated in Fig. 2.3(b) and (c).
- **Symbol**: a *group* of `ON` and `OFF` states that *together* represents one or several *data bits*. For example, under OOK with Manchester coding as shown in Fig. 2.3(a), a symbol is either a group of "`OFF-ON`" that represents the bit '0', or a group of "`ON-OFF`" that represents the bit '1'. Under MPPM as shown in Fig. 2.3(d), a symbol is a group of ten `ON/OFF` states that can represent $\lfloor \log_2 45 \rfloor = 5$ *data bits*.
- **Symbol slot**: the duration of a symbol, denoted as T. It is consisted of N time slots (recall that a symbol is a group of N "`ON-OFF`" states) and we have $T = N \cdot t_{\text{slot}}$. Note that both the N and T depend on the used modulation scheme and both of them ***are variable***.
- **Dimming level**: denoted as l. It is similar to the duty circle of LED. Mathematically, l can be expressed as follows

$$l = \frac{\text{the number of state "ON" in a symbol}}{N} \tag{3.1}$$

  In this work, the range of l is set to $[0, 1]$. For example, l=0 and l=1 mean that the LED is always turned off and on, respectively, and l=0.5 implies that the duty cycle of the LED is 50%.

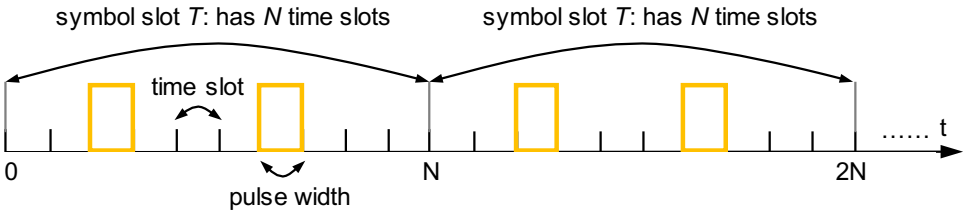


Figure 3.2: Illustration of the definition of the symbol, time slot, pulse width, and so on. In this illustrated example, we have N=10, l=0.2, and the pulse width equals to the time slot t_{slot}.

- **Resolution of dimming level**: the difference between two dimming levels. For example, for the set of dimming levels $[0.1, 0.2, 0.3, ..., 0.8, 0.9]$, its resolution is 0.1.
- **Symbol pattern**: denoted as $S(N, l)$, where N is the number of time slots in a symbol and l is the dimming level of that symbol. Note that a symbol pattern *does not* refer to the distribution of the states `ON` and `OFF` (namely, distribution of the light pulses) in a symbol.

3.2.1 Why MPPM does not work?

MPPM allows multiple `ON`s in a symbol slot and uses combinatorial symbol pattern to modulate the bits to be sent. If a symbol slot T is consisted of N time slots and among them, K time slots are occupied (i.e., are `ON`s), then in theory the achievable data rate R is given by

$$R = \frac{\left\lfloor \log_2 \binom{N}{K} \right\rfloor}{T} = \frac{\left\lfloor \log_2 \binom{N}{K} \right\rfloor}{N \cdot t_{\text{slot}}} \quad \text{bit/s} \tag{3.2}$$

According to combinatorial theory, the maximal R is achieved when $K \to \frac{N}{2}$. Moreover, if K is fixed, then $\binom{N}{K}$ increases consistently with N. Therefore, the upper bound of the data rate, $R_{\max}$, is given as

$$R_{\max} = \lim_{\substack{K \to \frac{N}{2} \\ N \to \infty}} R = \lim_{\substack{K \to \frac{N}{2} \\ N \to \infty}} \frac{\left\lfloor \log_2 \binom{N}{K} \right\rfloor}{N \cdot t_{\text{slot}}} \quad \text{bit/s} \tag{3.3}$$

Eq. (3.3) means that increasing N and letting $\frac{K}{N} \to \frac{1}{2}$ will benefit the data rate R of the system. In a VLC system with smarting lighting enabled, the dimming level of the LED should be fixed if the ambient light and the users' preference of lighting condition do not change. Therefore, $\frac{K}{N}$ is fixed and we can not set $\frac{K}{N} \to \frac{1}{2}$. By increasing N, the resolution of the dimming level is increased, and from Eq. (3.3), this also benefits R.

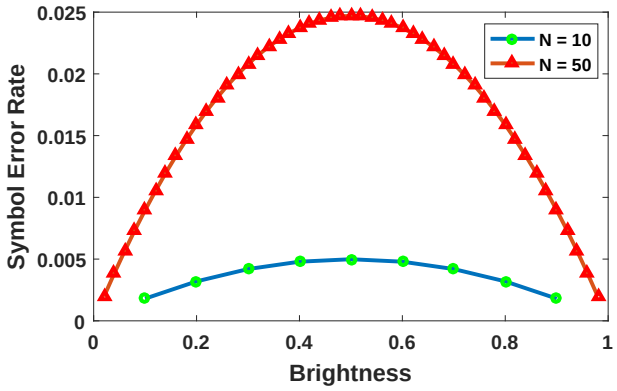


Figure 3.3: The MPPM symbol error rate as a function of dimming level

However, increasing N brings high Symbol Error Rate (SER). Let's model the detection behavior of the photodiode as photon counting process following Poisson distribution [37]. The probabilities of decoding an `OFF` (no pulse) and an `ON` (with pulse) incorrectly are denoted as P_1 and P_2, respectively. To decode a whole symbol correctly in MPPM, all the `ON`s and `OFF`s need to be detected correctly. Otherwise, the symbol will be decoded wrongly. The symbol error rate P_{SER} can be written as follows:

$$P_{\text{SER}} = 1 - (1 - P_1)^{N-K}(1 - P_2)^K \tag{3.4}$$

Moreover, in reality, Eq. (3.4) should be adjusted as follows [35]:

$$P_{\text{SER}} = 1 - \left(1 - \frac{K}{N} \cdot P_1\right)^{N-K} \left(1 - \left(1 - \frac{K}{N}\right) \cdot P_2\right)^K , \tag{3.5}$$

where the extra fixing factors of P_1 and P_2 are added. Eq. (3.5) implies the existence of trade-off between the symbol slot (which affects the resolution of dimming level) and the symbol error rate. Fig. 3.3 illustrates the relationship between N and the symbol error rate, where P_1 and P_2 are set to 0.001 and 0.002, respectively, without the loss of generality. We can observe that the symbol error rate reaches its peak value when the dimming level is 0.5. This can be explained by Fig. 3.4, where the LED's artifact is presented. Ideally, the received signal at the photodiode should be square waveform. Due to the slow charging/discharging of LED under high operation speed,

Figure 3.4: The signal distortion of LED waveform

the waveform is distorted and thus becomes triangle. When the dimming level is around 0.5, the switching between `ON`s and `OFF`s is more frequent, making the received light signal go up and down. Thus, the system is more prone to decode the symbols wrongly, compared to the pattern of consecutive `ON`s and `OFF`s when the dimming level is relatively low or high. Taking P_{SER} into account, the data rate can be further written as

$$\begin{aligned} R &= \frac{\left\lfloor \log_2 \binom{N}{K} \right\rfloor}{N \cdot t_{\text{slot}}} \cdot (1 - P_{\text{SER}}) \\ &= \frac{\left\lfloor \log_2 \binom{N}{K} \right\rfloor}{N \cdot t_{\text{slot}}} \cdot \left(1 - \left(1 - \frac{K}{N} \cdot P_1\right)^{N-K} \cdot \left(1 - \frac{N-K}{N} \cdot P_2\right)^{K}\right) \end{aligned} \tag{3.6}$$

From Eq. (3.6) and the results in Fig. 3.3, the following insight can be drawn:

Insight 1: MPPM can provide fine-grained dimming level by simply increasing the N, but this could reduce the system's other performance greatly, e.g., symbol error rate. Therefore, we should not simply use a large N to support fine-grained dimming levels.

3.2.2 Increase dimming levels via *multiplexing*

From the above analysis, we know that the resolution of the dimming level cannot be increased simply by increasing N, namely, increasing the symbol duration T. To solve this problem and provide fine-grained dimming levels, this thesis proposes to use the *multiplexing* of different dimming levels. The intuition behind the multiplexing is that if we combine *equally* the symbols with dimming level l_1 with the symbols with dimming level l_2, then we can obtain a *super-symbol* where the new dimming level is $\frac{l_1+l_2}{2}$.

For example, when N=10, the number of dimming levels supported by the system is nine, *i.e.*, $0.1, 0.2, ..., 0.8, 0.9$. Thus, the resolution is 0.1, which is low. Via multiplexing, we can append a symbol with a dimming level of 0.2 to a symbol with a dimming level of 0.1, which generates a super-symbol that has a dimming level of 0.15, as shown in Fig. 3.5. The number of slots

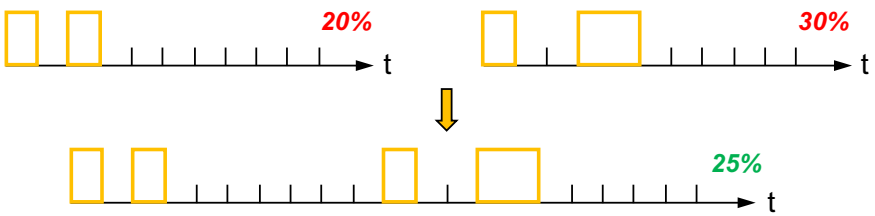
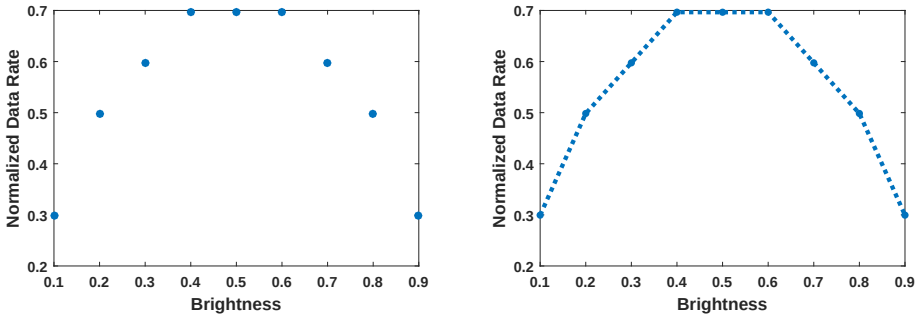


Figure 3.5: Using the multiplexing of existing patterns to achieve a higher resolution of dimming levels ($N = 10$)



(a) Before multiplexing

(b) After multiplexing

Figure 3.6: Supported dimming levels before/after applying the multiplexing approach proposed in this thesis

in the super-symbol is now 20 (i.e., N'=20, which is doubled). This *does not* increase the symbol error rate because each symbol in the super-symbol will be decoded separately. Note that the resolution after the above multiplexing now becomes 0.05 instead of 0.1.

Furthermore, we can have more fine-grained resolution. For example, for a dimming level of 0.175, we can use multiplexing by appending three symbols with the dimming level of 0.2 to a symbol with the dimming level of 0.1. And the new resolution becomes $0.2 - 0.175 = 0.025$.

The achieved fine-grained resolution after multiplexing is visually presented in Fig. 3.6, where x-axis is the dimming level (N=10) and y-axis is the normalized data rate. In Fig. 3.6(left), only nine discrete dimming levels are available. After applying the proposed multiplexing approach, the dimming levels become 'continuous', as shown in Fig. 3.6(right). The new dimming levels are the linear combination of the initial nine levels. Therefore, under the new dimming levels that are obtained by multiplexing two neighboring initial levels, the achieved data rate is a linear function of the new dimming level.

3.3 Adaptive multiple pulse position modulation

The previous section presents the proposed multiplexing approach. The advantages of multiplexing are that it can support fine-grained dimming levels and at the same time does not increase the symbol error rate. However, these are not enough. If we look back at Fig. 3.6, we can easily observe that *the data rate drops dramatically* from when the dimming level is 0.5 to when the dimming level is either low (e.g., 0.1) or high (e.g., 0.9). This is not what we want. In this section, I propose a novel modulation scheme for SmartVLC

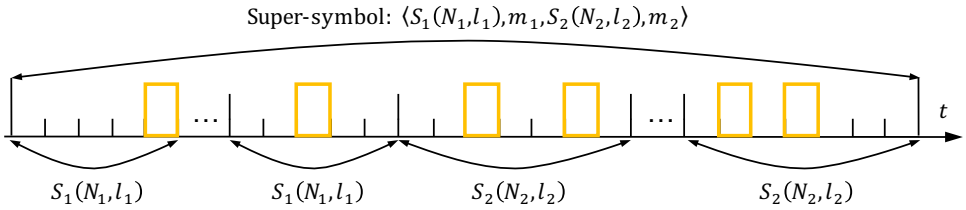


Figure 3.7: Illustration of a super-symbol

–**Adaptive Multiple Pulse Position Modulation (AMPPM)**– to alleviate the above problem.

The *key idea* in AMPPM is that **use different Ns and multiplex them in time domain to compose *super-symbols*. These super-symbols can satisfy the desired resolution of dimming level, at the same time can optimize the achievable data rate for each dimming level.**

In this thesis, a super-symbol is defined as a composition of *two* symbol patterns $S_1(N_1, l_1)$ and $S_2(N_2, l_2)$. A super-symbol is generated by concatenating a number of $S_1(N_1, l_1)$ and $S_2(N_2, l_2)$, as illustrated in Fig. 3.7. Furthermore, let S_{super} denote a super-symbol, then S_{super} can be represented by a tuple

$$S_{\text{super}}: \ < S_1(N_1, l_1),\ m_1,\ S_2(N_2, l_2),\ m_2 > ,$$

where m_i is the number of the symbol $S_i(N_i, l_i)$ in S_{super}, $i \in \{1, 2\}$. The detail composition of a super-symbol (i.e., the above tuple) is referred as the **pattern** of that super-symbol. Let N_{super} denote the number of slots in the super-symbol S_{super}, then we have

$$N_{\text{super}} = m_1 \cdot N_1 + m_2 \cdot N_2 \ .$$

Besides, let l_{super} be the dimming level supported by S_{super}, then we get

$$l_{\text{super}} = \frac{l_1 \cdot m_1 \cdot N_1 + l_2 \cdot m_2 \cdot N_2}{N_{\text{super}}} \ .$$

In AMPPM, *the key challenge is to find the **patterns of super-symbols** for any given dimming level*, i.e., choose the best N_1, l_1, m_1, and N_2, l_2, m_2 that can optimize the data rate for a required dimming level l_{super}. To find the best super-symbol patterns, AMPPM adopts the following four steps:

- **Step 1:** Calculate the upper bound of N and N_{super} that can support non-flickering. The upper bound is denoted as $N_{\max}$.
- **Step 2:** Given the restrictions of $N_{\max}$ and an upper bound of acceptable symbol error rate, find all the available Ns, i.e., a set $\mathcal{N} = \{N_1, N_2, N_3, ...\}$, and the available dimming levels $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, ...\}$ where $\mathcal{L}_i = \{l_i^1, l_i^2, ...\}$ is the set of available dimming levels for N_i. Given $\mathcal{N}$ and $\mathcal{L}$, we can get all the available symbol patterns $\mathcal{S}_{\text{step2}}$.

- **Step 3:** Calculate the data rate for all the symbol patterns in $\mathcal{S}_{\text{step2}}$.
- **Step 4:** For each dimming level, choose the symbol pattern that leads to the maximum data rate. Then, use the multiplexing approach to generate super-symbols (i.e., find the patterns) that support more fine-grained dimming levels.

The details of these steps are presented below:

Step 1: Calculate the upper bound of N and N_{super}

We know that the non-flickering VLC requires that the light intensity is fixed for every definite time interval. In AMPPM, since it is involved with using different Ns with different dimming levels, it is necessary to know that the upper bound of N or N_{super} so that we can use the patterns which will not cause flickering. On one hand, for each single pattern of a symbol, it should not cause flickering. On the other hand, the combined pattern of a super-symbol after multiplexing should also not cause flickering. Suppose the transmitter can operate the LED at a frequency of f_{tx}. Let f_{th} denote the minimal frequency of turning on/off the LEDs so that it will not cause flickering. Let N_{max} denote the maximal number of consecutive time slots which will not cause flickering. Then N_{max} can be calculated as follows

$$N_{\text{max}} = \frac{f_{\text{tx}}}{f_{\text{th}}} \tag{3.7}$$

For example, if f_{tx} is 125 KHz and the minimum frequency f_{th} for human not perceiving flickering is 200 Hz. Then the upper bound for not causing flickering is N_{max}=125000/200=625. This indicates that the number of slots in a symbol or a super-symbol should be less than or equal to 625, i.e., $N \leq$625 and $N_{\text{max}} \leq$625. The average of the light intensity in each 625 (or less) consecutive slots should be the same, so it will not cause flickering from the view of human's eyes. Note that with more advanced hardware, the upper bound N_{max} should be much higher than 625 and will not become a constraint in AMPPM.

Step 2: Calculate the available Ns and dimming levels under the constraint of symbol error rate

The VLC system suffers from higher symbol error rate with a large N. Therefore, the Ns satisfying the requirement in Step 1 cannot be alone taken for further calculation. To calculate a more precise range of Ns, here, I first define the Symbol Reception Rate (SRR) as follows:

$$P_{\text{SRR}} = 1 - P_{\text{SER}} = \left(1 - \frac{K}{N} \cdot P_1\right)^{N-K} \left(1 - \left(1 - \frac{K}{N}\right) \cdot P_2\right)^{K} \tag{3.8}$$

Then we can draw the curve where the symbol reception rate is a function of the dimming level for all the possible Ns obtained in Step 1, as shown in

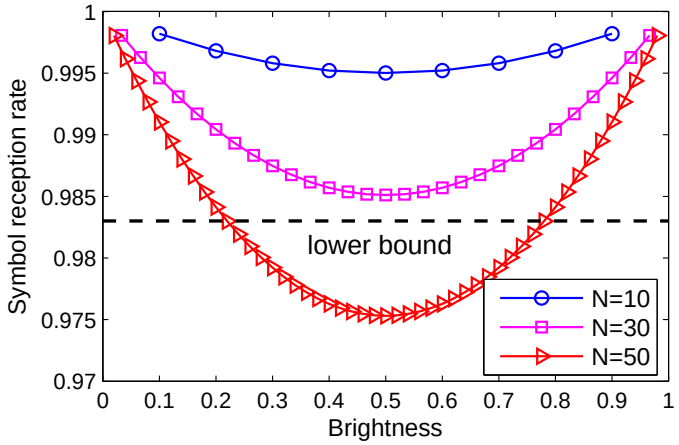


Figure 3.8: Available patterns above the lower bound

Fig. 3.8. In order to meet the reliable communication system requirements, a lowest bound of the symbol reception rate should be given according to the specific requirements. All the symbol patterns $S(N, l)$ (combinations of N and the dimming level l) that are below the lower bound of the symbol reception rate would be abandoned, e.g., $S(N{=}50, l{=}0.3)$ or $S(N{=}50, l{=}0.6)$ in Fig. 3.8. The other symbol patterns (denoted as $\mathcal{S}_{\text{step2}}$ for simplicity) are saved for the next-step calculation.

Step 3: Calculate the data rate for all the symbol patterns in $\mathcal{S}_{\text{step2}}$

In this step, the data rates under all the possible symbol patterns obtained finally in Step 2 are calculated according to Eq. (3.6). As an example, the calculated data rates under specific settings are given in Fig. 3.9. There, each color represents a different N, and each maker '*' represents a symbol pattern. Note that these data rates have been normalized.

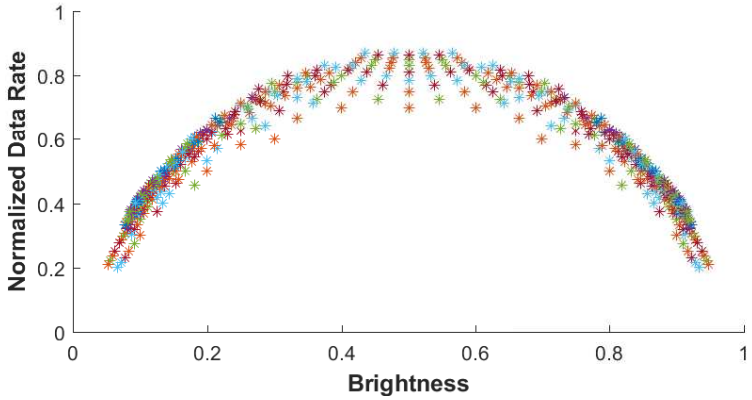


Figure 3.9: The normalized data rate for all the symbol patterns in $\mathcal{S}_{\text{step2}}$

Step 4: Obtain the patterns of super-symbols via multiplexing

The last step is to pick up the best symbol patterns for the multiplexing and then generate the super-patterns. Before multiplexing, we only have symbol patterns with discrete dimming levels. If we only pick the pattern with highest data rate under each dimming level, the resolution is discrete and the global throughput is not guaranteed to the most significant value. This is illustrated in Fig. 3.10 by the red dash line.

Through the analysis presented in the last section, we knew that the system can achieve continuous dimming levels through multiplexing. And the targeted data rate is determined by the two discrete patterns. The proposed AMPPM scheme exploits it not only to achieve fine-grained resolution of the dimming levels, but also to optimize the data rate under each dimming level.

First of all, the system will choose the symbol patterns with absolutely largest date rate. In Fig. 3.9, they are represented by the four blue points in the middle. And we connect them together. Now we can get all the dimming levels on this line(i.e. from dimming level 0.42 to 0.58), as long as the multiplexing patterns in the super-symbol do not exceed the flickering threshold. Next step is to find the optimal symbol patterns on the two sides. Since the graph is symmetrical, we can just focus on one side. We take the right side as an example as shown in Fig. 3.10. The symbol pattern at dimming level 0.58 can connect to all the other symbol patterns on its right

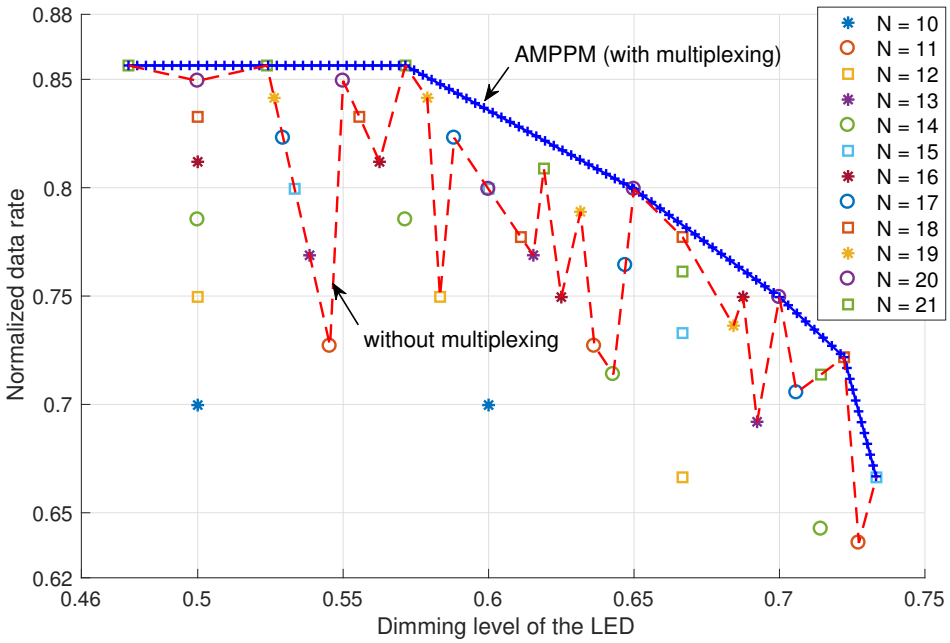


Figure 3.10: Best pattern selection based on slope

side to get the super-symbol with the dimming level between them. In order to select the optimal combination in terms of the throughput, we calculate all the slope of the line determined by the symbol pattern at 0.58 and the symbol patterns at other dimming levels and select the symbol pattern which possesses the smallest slope value. In Fig. 3.10, it is the symbol pattern at dimming level 0.65. From this newly selected symbol pattern, we again search for the next symbol pattern based on the slope value. The iteration is done until all the desired dimming range is covered. The optimal patterns from dimming level 0.47 to 0.73 are illustrated by the blue line in Fig. 3.10. It not only possesses larger throughput but also the higher resolution in terms of the dimming level than the red line.

3.4 Adaptation to dynamic ambient light

The adaptation of SmartVLC to dynamic ambient light is involved with the cooperation between the LED light and ambient light. There are two goals that must be achieved:

Goal 1: the sum of *ambient* and *LED* light intensity should be "constant" within the area-of-interest, as expressed by the following equation

$$I_{\text{sum}} = I_{\text{led}} + I_{\text{amb}} \tag{3.9}$$

where I_{sum} is the targeted light intensity within the area-of-interest and is constant (only depends on users' preference). I_{led} and I_{amb} are the intensities of LED light and ambient light, respectively.

Goal 2: the adaptation process of LED light should not be observed by users, namely, flickering-free to users.

The solution to achieve **Goal 1** is straightforward. Let assume at time t_1, the intensities of LED light and ambient light are I^1_{led} and I^1_{amb}, respectively. At time t_2, the intensity of ambient light changes to I^2_{amb}. Then we just need to increase the lightness of the LED by $\triangle I_{\text{led}}$ that is given as follows:

$$I^1_{\text{led}} + I^1_{\text{amb}} = I^2_{\text{led}} + I^2_{\text{amb}} \implies \triangle I_{\text{led}} = I^2_{\text{led}} - I^1_{\text{led}} = I^1_{\text{amb}} - I^2_{\text{amb}} \tag{3.10}$$

Whenever the system detects a change in ambient light, the LED's lightness should be changed accordingly, by modifying its dimming level. The ambient light changes 'continuous' in reality. Therefore, the lightness of the LED should not be changed 'suddenly', i.e., increased by $\triangle I_{\text{led}}$ in one step. Otherwise, it will cause flickering.

The **Goal 2** aims at avoiding the flickering problem during the process of the LED's adaptation to ambient light changes. One solution is to increase I^1_{led} *gradually and evenly*, e.g., each step by τ, until I^1_{led} reaches to $I^1_{\text{led}}+\triangle I_{\text{led}}$.

The number of steps taken is $\lceil \triangle I_{\text{led}}/\tau \rceil$. Note that here the τ is *constant* and its value is constrained by the requirement that a change of the LED's lightness by τ will not cause noticeable flickering for the users.

In SmartVLC, I propose a different method to achieve **Goal 2**. The main idea is to adopt a *variable* τ that can reduce the number of steps taken to reach the target changes $\triangle I_{\text{led}}$. The motivation behind this idea is that *the response of human's eyes to visible light changes is* ***nonlinear***. Specifically, the relationship between the perceived lightness I_p by human's eyes and the measured lightness I_m by a light meter is given as follows [32]

$$I_{\text{p}} = 100 \times \sqrt{\frac{I_{\text{m}}}{100}} \tag{3.11}$$

This relationship is also shown in Fig. 3.11(a). The reason behind this is the human pupil's biological reaction to different lightness. In dark environment, people enlarge their eyes and therefore induce more light. In the rest of this thesis, I use *measurement domain* and *perception domain* to represent the lightness measured by light meters and perceived by humans, respectively.

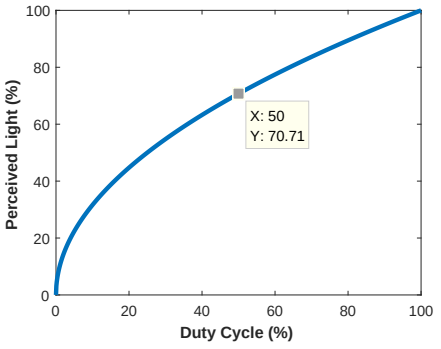
The proposed method in SmartVLC to achieve **Goal 2** works as follows:

1. Given the lightness I^1_{led} and I^2_{led} in *measurement domain*, convert them to $I^1_{\text{led-p}}$ and $I^2_{\text{led-p}}$ in *perception domain* based on Eq. (3.11):

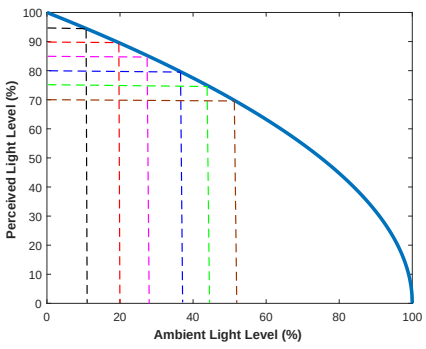
$$I^1_{\text{led-p}} = 100 \times \sqrt{\frac{I^1_{\text{led}}}{100}}\,, \qquad I^2_{\text{led-p}} = 100 \times \sqrt{\frac{I^2_{\text{led}}}{100}} \tag{3.12}$$

2. Let $\triangle I_{\text{led-p}}$ be the difference between $I^1_{\text{led-p}}$ and $I^2_{\text{led-p}}$, then

$$\triangle I_{\text{led-p}} = I^2_{\text{led-p}} - I^1_{\text{led-p}} \tag{3.13}$$



(a) Nonlinear brightness response of human's eyes



(b) The relationship between ambient light and perceived light

Figure 3.11: Adaptation to dynamic ambient light

3. Increase $I^1_{\text{led-p}}$ *gradually* by a step of τ_p, until it reaches to $I^2_{\text{led-p}}$. The number of steps taken is $\lceil \triangle I_{\text{led-p}}/\tau_p \rceil$. Here the τ_p is *constant* and a change of τ_p from the perception of human's eyes will not cause noticeable flickering. I will carry out experiments to obtain τ_p in Chapter 5. This method is illustrated in Fig. 3.11(b). **Note that in perception domain, with the *constant* step of τ_p, the lightness changes *linearly*; while in measurement domain, the lightness changes *non-linearly* due to the *variable* step τ.**

3.5 Coding scheme

Classical coding schemes of pulse position based modulation schemes can be categorized as two main approaches: tabulation approach [33] and constellation approach [28]. In the first approach, the mapping between bit stream and the codeword is recorded in the tabulation. In the second approach, the mapping relationship is recorded in the constellation graph.

However, both of these two approaches are based on the method of exhaustion searching and all the items are recorded in the memory space. With the increment of the number of N in one-time symbol T, the number of available patterns increases exponentially, which inevitably brings the burden on the hardware. For example, when $N = 50$ and $K = 25$, the different combinations are $\binom{N}{K} = \binom{50}{25} \approx 1.26\text{e+}14$. If each mapping relationship item occupies 4 Bytes, it results in 126 TB memory space. Therefore, it is quite necessary to come up with the direct mapping algorithm between the bit stream and the codeword.

To this end, I propose the coding scheme *combinatorial dichotomy*. When $N = 5$ and $K = 2$, the corresponding mapping relationship under the algorithm is shown in Tab. 3.1. The key idea is to search the mapping rela-

Table 3.1: Mapping example ($N = 5$, $K = 2$)

Binary Input	**Codeword Output**
000	00011
001	00101
010	01001
011	10001
100	00110
101	01010
110	10010
111	01100

tionship via combinatorial factors. Each time after updating the new combinatorial factor, the system can always calculate the value of one definite bit as either 0 or 1.

To start encoding, The encoding algorithm decides the codeword from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). If the LSB is set to the value "1", the remaining bits in code word can represent $\binom{N-1}{K-1}$ different types of binary inputs. Therefore, if the value of the binary input is smaller than the value of $\binom{N-1}{K-1}$, the LSB is set as "1". Otherwise, the LSB is set as "0" and the binary input is subtracted by $\binom{N-1}{K-1}$. In this example, with the incoming 3 bits binary stream, the LSB of the mapped codeword could be either 0 or 1. Therefore, the mapping in terms of this bit is divided into two groups and our mission is to find which one is the group of our target number. Based on the formula, e.g., for 100, its absolute value is not smaller than $\binom{5-1}{2-1}$ and it is in the second group where the LSB is 0. Meanwhile, 100 gets subtracted by $\binom{N-1}{K-1}$ and becomes 000.

Then, the algorithm calculates the value of second bit originating from the LSB. The algorithm compares the updated value of binary input with the value of patterns comprised of the remaining bits in the code word. Following our example, the second bit could be either 0 or 1. We use our remaining 000 to be compared $\binom{5-2}{2-1}$ and it is smaller than that. Then the second bit is arranged as 1. The algorithm stops either the total iteration is processed N times or K times "1"s or $(N-r)$ times "0"s are used up. For the next iteration of the algorithm, the 000 is smaller than $\binom{5-3}{2-2}$ and the third bit originating from LSB is assigned as 1. At this moment, all the "1" are used up and the assignation is over.

For the decoding algorithm, the decoding rule is done in the exactly opposite way of the encoding algorithm. Likewise, the receiver should also know the lightness pattern information before decoding (namely, the value of N and K) from the header of the frame. The specific decoding algorithm and encoding algorithm are shown in the Procedure 1 and Procedure 2 in Appendix A respectively.

3.6 Communication protocol

To enable the communication between the transmitter and receiver, a frame format must be designed. In SmartVLC, I design the frame format as shown

Table 3.2: The frame format in SmartVLC

Preamble	**Length**	**Pattern**	**Compensation**	**ShortSync**	**Payload**	**CRC**
3 Bytes	2B	4B	xB	1B	0-MAX B	2B

in Table 3.2. Each frame starts with a three-byte *Preamble* indicating the beginning of a new frame. The frame *header* comes after the preamble, and it includes the fields *Length* and *Pattern*. The *Length* represents the number of bytes in the payload. The *Pattern* occupies four bytes and carries the details about the super-frame, i.e., the tuple $< N_1, m_1, N_2, m_2 >$ that contains part of the information about the two frames S_1 and S_2 that are selected to compose the super-frame through multiplexing. Note that there is no need to include the dimming levels of the two symbols into the filed *Pattern* of the frame, because the receiver does not need them to decode data. The fields *Compensation* and *ShortSync* are related to flickering control, and will be explained later. The *Payload* and *CRC* (cyclic redundancy check) are placed to the end of each frame.

From the analysis presented in the previous sections, we know that the payload of each frame has been guaranteed to satisfy the non-flickering requirement by adopting the proposed AMPPM scheme. However, this is not enough. **We must guarantee that the whole transmission process is flickering-free, as illustrated in the high-level view in Fig. 3.13.** To achieve this goal, the flickering inside a frame and between different frames must be avoided, i.e., avoid *intra-frame flickering* and *inter-frame flickering*.

Avoid intra-frame flickering: the preamble and frame header are modulated with OOK, for simplicity. To align the lightness of them with that of the payload, *compensation time* must be appended to the frame header, as shown in Table 3.2. The compensation can be consecutive `ON`s or `OFF`s, depending on the dimming level of the payload. After the compensation time (could be 'very long'), the system is at the risk of losing synchronization. Therefore, the *ShortSync* field, which is a rise or falling edge, is appended to achieve synchronization again. When calculating the duration of the compensation time, the lightness of ShortSync is also taken into account. If the transmitter's operation frequency is very low, the system may suffer from flickering earlier in the header than expected. Under this case, the compensation time can be appended at the end of each single part, namely preamble, length and pattern part. Likewise, the sync part should be followed with each compensation time. In this thesis, the operation frequency is high enough to avoid this problem.

Avoid inter-frame flickering: With the solution described above, a frame

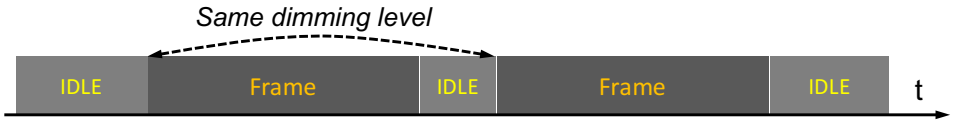


Figure 3.12: High-level view of the requirement of non-flickering

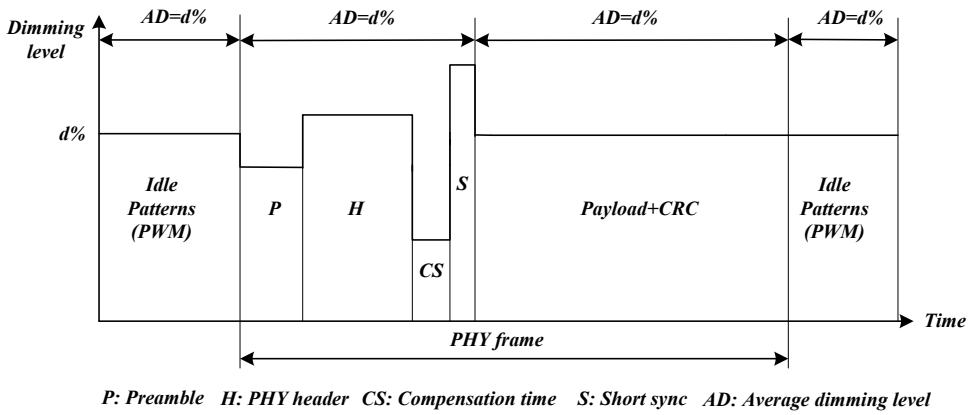


Figure 3.13: Details of the overall transmission process

is guaranteed to have even lightness. To avoid the inter-frame flickering, the idle pattern is used in SmartVLC. The lightness of the idle patterns are also adapted to the ambient light changes, similar to the adaptation of frames. In this thesis, I use the PWM to adjust the lightness of idle patterns.

Finally, the overall transmission process is shown in Fig. 3.13.

Chapter 4

Implementation

This chapter presents the implementation of the SmartVLC system, including both the hardware and the software implementations. An overview of the system implementation is given first, followed by the details on the electronic components selection, design, and prototype of the transmitter and the receiver.

4.1 Overview

The high-level overview of the implementation of the SmartVLC system is shown in Fig. 4.1. The Embedded System (ES) boards are responsible for all the tasks related to the software, including data generation, modulation/demodulation, encoding/decoding, etc., as introduced in Chapter 3. The front-end of SmartVLC cooperates with the ES boards to perform the LED driving, ambient light sensing, and sampling.

The ES board used in this thesis is the BeagleBone Black (BBB) board [2], as shown in Fig. 4.2. BBB is a small, inexpensive and relatively powerful Single Board Computer (SBC). The reasons why BBB is selected as the development platform for this project are listed below:

1. BBB has 65 General-Purpose Input/Output (GPIO) pins, which are

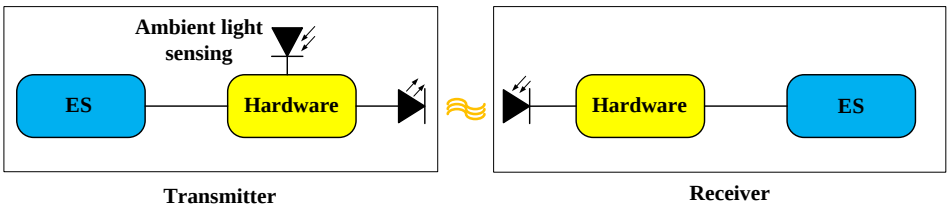


Figure 4.1: A high-level overview of the implementation of SmartVLC

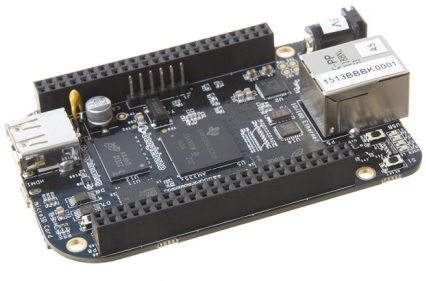


Figure 4.2: The BeagleBone Black embedded platform

quite convenient for prototyping.

2. Except for the main ARM processor, BBB also has two on-board 200 MHz Programmable Real-Time Units (PRU) that can work like independent micro-controllers. These PRUs can be used for time-sensitive tasks, such as sampling. Through the collaboration between the PRUs and the main ARM processor, the performance of a system can be improved greatly.

3. For time-sensitive tasks such as sampling, the BBB (≈56 USD) is much cheaper compared to other high-end platforms such as USRP (≈5000 USD) or WARP (≈4900 USD) that are introduced in Chapter 2.

4.2 Hardware

Next, I present the hardware implementation of the transmitter and receiver.

Transmitter

The block diagram, schematic, and prototype of the transmitter are given in Fig. 4.3, Fig. 4.4(a-b), respectively. There are mainly four parts: BBB,

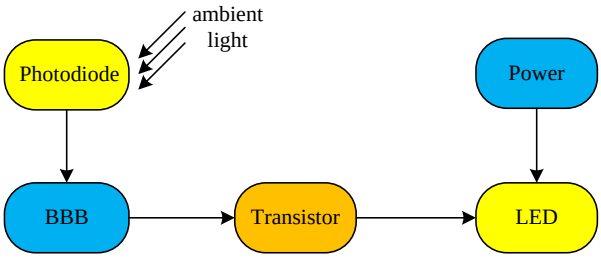
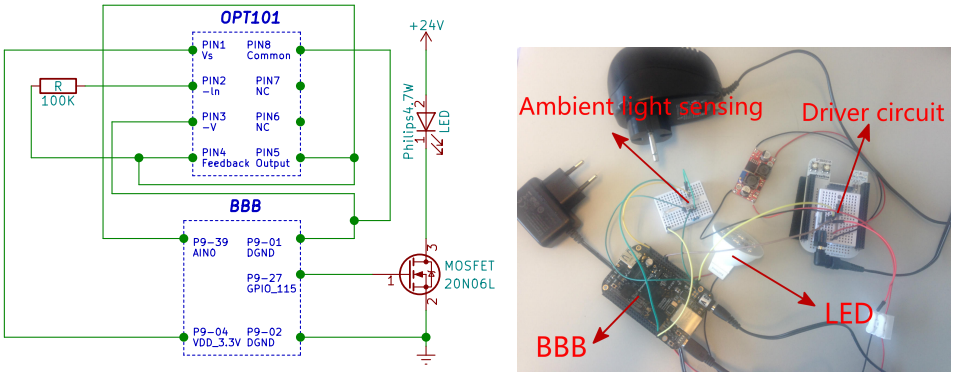


Figure 4.3: The block diagram of the transmitter



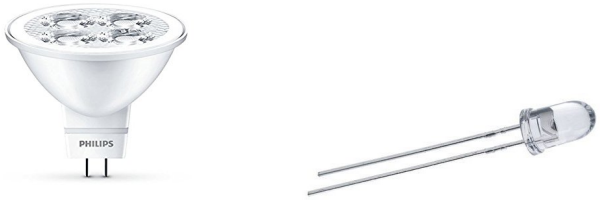
(a) Schematic of the transmitter (b) Prototype

Figure 4.4: Final design of the transmitter

LED, transistor (to drive the LED), and the photodiode. The GPIO of BBB outputs control signal to trigger the transistor at high speed to modulate the LED light. The LED is powered by a 24V DC voltage that is boosted up from a 12V DC power source through a DC-to-DC converter. Sensing the ambient light is done by sampling the photodiode that is driven directly by the on-board 3.3V DC voltage from the BBB's GPIO.

Ideally, in VLC, it is a good idea to choose the LED whose lightness is a linear function of the forwarding current as the transmitter. In this project, I explore the possibility of both high-power and low-power off-the-shelf LEDs, as shown in Fig. 4.5. The low-power LED can be driven directly by the GPIOs of the BBB, while a high-power LED needs to be driven by an external power source (24V). The high-power LED I choose in this project is the Philips 4.7W LED, and the low-power LED is HLMP-EG08-YZ000. The high-power LED is further disassembled by removing the AC-DC converter that can slow down the transition speed. In my tests within an indoor space with the existence of sunlight and ceiling lights, the low-power LED can support a communication distance up to 0.75 meters. The communication distance supported by the high-power LED is longer, which is up to 3.65 meters. Therefore, I use the high-power LED in most of the experiments presented in the next chapter. Another advantage of using high-power LEDs is that they are normally used indoor for the illumination purpose, thus can prove to some extent the practical capability of this project. Nevertheless, for the low-power LED, I will use it in some tests regarding flickering, for the purpose of comparison.

The output voltage of the BBB's GPIOs is only 1.8V, which is not sufficient to drive the high power LED. Therefore, a transistor and an additional power supply are needed. In this work, I use the Metal Oxide Semiconductor Field Effect Transistor (MOSFET) which is suitable for our requirement. It



(a) High-power LED (b) Low power LED

Figure 4.5: The high-power and the low-power LEDs used in this project: (a) Philips 4.7W LED; (b) HLMP-EG08-YZ000

(a) SFH203P (b) SFH206K (c) OPT101

Figure 4.6: The different photodiodes that are tested in this project

can be operated using a small positive voltage on the gate, and the GPIOs of the BBB can be adopted to do this task. The model of the used MOSFET is N-channel 20N06L from Texas Instrument. This transistor is also designed with ultra low gate resistance and capacitance, making it quite suitable for the low power output board such as the BBB.

The transmitter senses ambient lights using a photodiode. In this project, I have tested three types of photodiodes, as shown in Fig. 4.6. Eventually, the OPT101 is selected due to its sensitive response to ambient light and its intrinsic operational amplifier based circuit. The BBB cannot be connected to an incoming signal higher than 3.3V, otherwise it can be damaged. Therefore, a carefully selected feedback resistor (25 KOhm) is connected to OPT101 to adjust the range of the output voltage accordingly.

Receiver

The block diagram of the receiver is shown in Fig. 4.7. There are mainly four parts: BBB, ADC, amplifier, and photodiode. Incoming light signals are first sensed by the photodiode and then amplified by the amplifier. Analog signals from the amplifier are converted to digital signals by the ADC and then read by software in the BBB board for further process.

The photodiode used at the receiver converts incoming light signal to electrical signal. To support long communication distance, a sensitive pho-

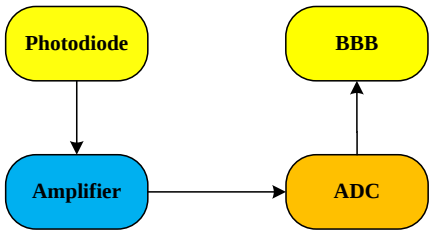


Figure 4.7: Receiver's block diagram

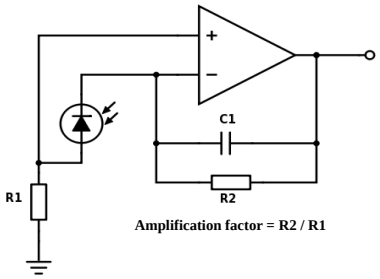


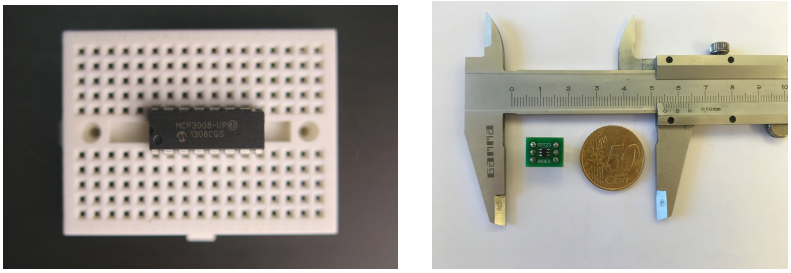
Figure 4.8: Transimpedance amplifier

todiode is required at the receiver. I again test several popular photodiodes as listed in Fig. 4.6. OPT101 is sensitive enough, but its response (rise/fall) time is too slow to satisfy the requirement of high-speed communication. SFH203P, on the other hand, possesses faster response time. However, it lacks in sensitivity. Eventually, I select SFH206K because it meets both the requirements of response time and sensitivity.

The output signal from SFH206K can be simply amplified via a resistor. However, due to the internal capacitance of the photodiode, large RC delay will be introduced. The larger the value of R, the larger the amplifying time, so the larger the RC delay. For the long distance communication, the amplifying gain of the signal requires being large. To solve this trade-off, the transimpedance amplifier is adopted. Its diagram is shown in Fig. 4.8. I use the amplifier TLC237 in this work. The gain can be determined directly by the feedback resistor and the system is free from the delay disturbance. On the other side, oscillations can occur if the phase margin is not sufficient because the transimpedance amplifier has a closed-loop amplifier. This can be compensated via the feedback capacitance. Hence, the experiments regarding the feedback capacitance value are conducted until the oscillation has disappeared. More specifically, a 2-stage amplifying circuit is used in the receiver instead of 1-stage, because the large gain in 1-stage brings heavy distortion to the amplified signal. Normally, the 1-stage amplify gain should not exceed 3 dB.

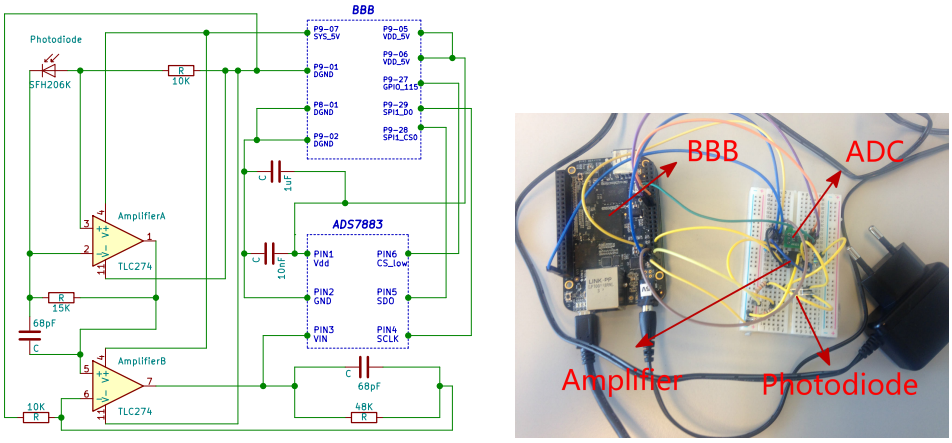
The amplifier is connected to the ADC. I have tested two popular ADCs: MCP3008 and ADS7883, as shown in Fig. 4.9. The ADC's sampling rate is determined by both the ES board's computation capability and the intrinsic speed of the ADC. The MCP3008 can be placed into a breadboard directly, but it can only reach up to 200 Kbps. The ADS7883 is a relatively smaller IC chip and cannot be placed into a breadboard directly. I solder the ADS7883 on a SOT23 package and it is proven to reach 1.5 Mbps sampling rate in the experiment. Due to its high sampling rate, the ADS7883 is finally used in the receiver.

The receiver's final design (schematic and prototype) is shown in Fig. 4.10.



(a) MCP3008 (b) ADS7883

Figure 4.9: The ADC chips tested in this project



(a) Schematic of the receiver (b) Prototype

Figure 4.10: Final design of the receiver

The receiver can be fully powered by the BBB, which enables the possibility of unifying the receiver to the BBB for convenient testing.

4.3 Software

The implementation of the software of SmartVLC is classified as two parts: hardware control and upper layer processing. The hardware control represents the process of controlling input/output pins of BBB. The upper layer processing represents the data generation, coding, and modulation, etc.

Hardware control

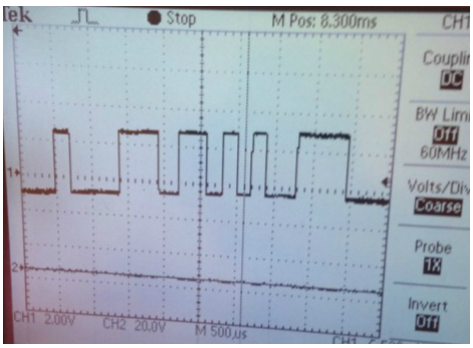
Writing/reading to/from the GPIO pins is a common method to control the front-end transceiver. For example, writing a '1' to a GPIO pin means the output of the pin is set to 1.8V, and writing a '0' means the output is set to

0V. These outputs can be used to modulate the LED lights.

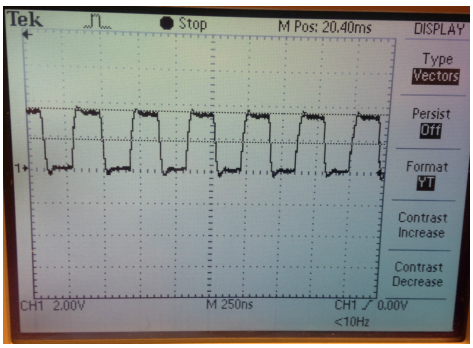
The BBB board runs the Linux operating system. It offers an easy way to write/read the GPIO pins through a number of files linked to the GPIOs. Nevertheless, this approach brings additional overhead when open/close the files. Another approach to interface with the GPIO pins is through memory. Since the GPIOs are connected to the CPU, they are assigned with physical memory addresses. Reading and writing these memory addresses can directly control the GPIOs and are much faster than the first approach.

To trigger the MOSFET at the transmitter or perform the sampling at the receiver, accurate timing is required. However, the above two approaches do not work well. For example, if we modulate the LED lights using the first approach, i.e., by writing to the files associated with the GPIOs, then the generated waveform is shown in Fig. 4.11(a). Apparently, the waveform suffers greatly from jitters. The jitters are mainly caused by the inaccurate timer function provided in either the Linux user space or the kernel space when the timing period is too short, e.g., shorter than 0.5 ms (i.e., the frequency of triggering the MOSFET is higher than 2 KHz). Even if the MOSFET is triggered by writing directly to the memory address of the GPIOs, the generated waveform still suffers from jitters. The reason behind these is that Linux is not an operating system designed for handling real-time tasks.

There are two ways to solve the above problem without adding any hardware such as a FPGA. The first solution is adding the Xenomai patch to the Linux kernel. Xenomai can improve the real-time performance of the original Linux kernel to some extent. However, even with Xenomai, the receiver's ADC can only be driven at a speed up to 50 K, which is still a limitation for the whole system (the maximal sampling rate of the ADC chip is 1.5 Mbps).



(a) By writing to the GPIO associated files: suffering from serious jitters (2 KHz signal)



(b) By adopting the BBB's PRU: jitter-free (5 MHz signal)

Figure 4.11: The shape of the generated signals using different methods

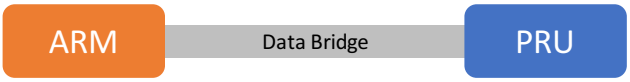


Figure 4.12: Collaboration between the ARM processor and the PRU

The other approach is what I adopt in this project: use the existing PRUs of the BBB. I combine the ARM processor with the PRUs, which gives us a heterogeneous processing solution, as shown in Fig. 4.12. The method is to cooperate the ARM processor with the PRU via a shared memory, so the PRU is responsible for all the hardware control and the ARM processor can abundantly exploit its resource to only deal with upper layer processing. Through this way, the system can easily meet the real-time requirements, as shown in Fig. 4.11(b). The PRU is programmed in assembly where each instruction takes 5 ns, and the ARM is programmed in C.

Upper layer processing

The upper layer processing includes the data generation, encoding/decoding, modulation/demodulation, etc.

The software of the transmitter in SmartVLC has three threads to perform different tasks: *(i)* the thread that is used to sense the ambient light using the timer function provided by the Linux kernel space; *(ii)* the thread that is used to generate data (dummy data in the current implementation of SmartVLC); *(iii)* the main thread that is used to perform the tasks shown in Fig. 4.13. First, it waits for the data generated by the second thread. Then it calculates the desired dimming level based on the current ambient light. After that, it encapsulates the frames by inserting the data into their payloads and by filling up the frame headers. Then the frames are sent to the PRU through the data bridge. After receiving the acknowledgment from the PRU, it returns to the *initial* state.

The PRU at the transmitter keeps checking the status of the data bridge. If new frames come, it sends out the frames by modulating the LED at the given frequency and dimming level, and then sends an acknowledgment to the ARM processor. If no frames received, the PRU just sends idle patterns with the required dimming level.

The state machine of the receiver is shown in Fig. 4.14. The PRU keeps capturing the data regardless the status of the main ARM processor. This is because the ADC's sampling should be continuous, otherwise would cause errors. The PRU keeps updating its pointer in the memory pool. If the pointer reaches the end of the data pool, it will update the pointer to the initial position and start a new round of data updating. The main thread of the ARM processor at the receiver firstly checks the availability of the data based on the position of a pointer in case of speed discrepancy. If the

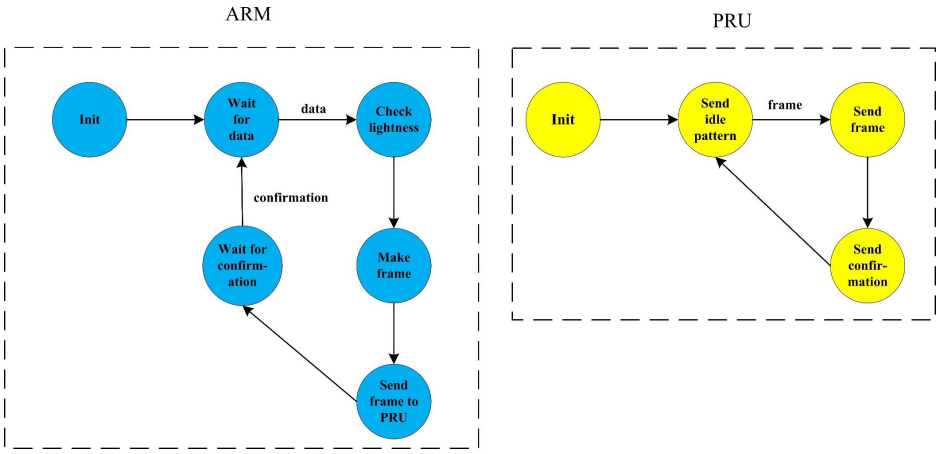


Figure 4.13: State machine of the transmitter

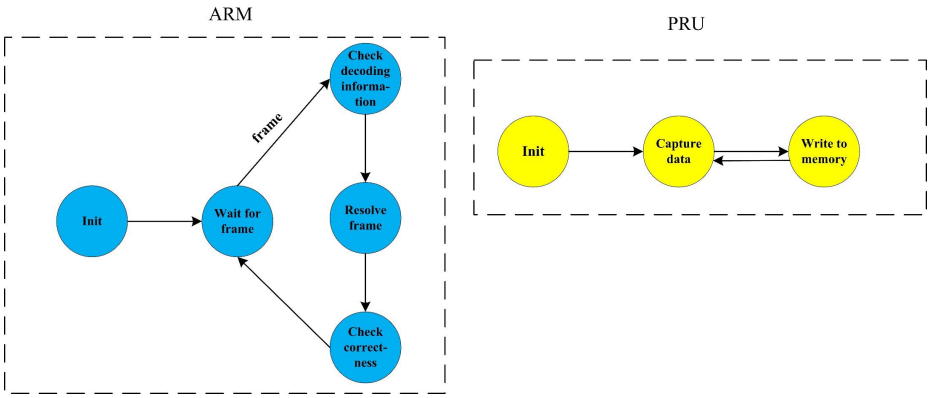


Figure 4.14: State machine of the receiver

amount of data is sufficient to capture the preamble and verified to be true, it subsequently captures the whole frame. Based on the decoded configuration of the AMPPM from the header, the system then decodes the whole frame. After that, the system checks the CRC of the frame to see if it is correctly resolved. If not, the system reports an error.

Chapter 5

Evaluation

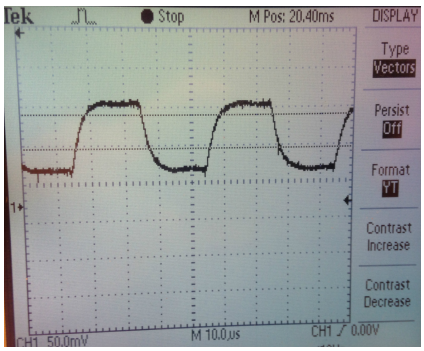
This chapter presents the performance evaluation of the SmartVLC via experiments. Several key system parameters are determined first, followed by the evaluations under static and dynamic scenarios.

5.1 Key parameters determination

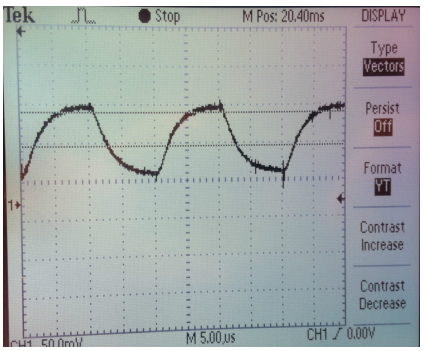
PHY rate of the transmitter and sampling rate of the receiver

As presented in Chapter 4.2, the ADC in SmartVLC is ADS7883, which supports a sampling rate up to 1.5 Mbps together with the usage of PRU for sampling. Currently, the main bottleneck of SmartVLC is the bandwidth of the Philip LED. When the LED is switched on/off at different frequencies f_{tx}, the received signals at the photodiode are shown in Fig. 5.1. We can observe that with the increase of f_{tx}, the light signal gradually gets distorted from square waveform to triangle waveform. The photodiode SFH 206K used in SmartVLC has a relatively short response time (20 ns). Therefore, the signal distortion is mainly due to the limited bandwidth of the LED, which is also demonstrated by Fig. 5.2 where the oscilloscope snapshot is obtained by testing the signal at the transmitter.

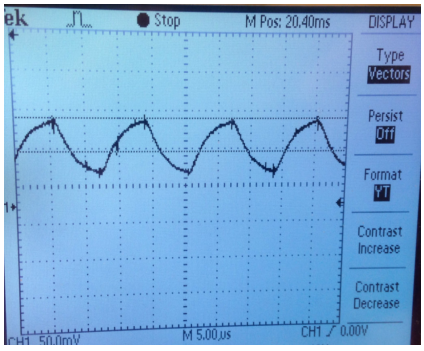
Since the gap between `ON` and `OFF` becomes narrower and the rising and falling times are longer under large f_{tx}, the system's performance in terms of data rate and communication distance will be reduced. The distorted triangle waveform is also more prone to cause decoding errors. To maintain the system's communication distance and accuracy, and also operate the LED at a frequency as high as possible, I eventually choose to set the operation frequency f_{tx} (i.e., PHY rate) of the LED to **125 KHz**. This also means t_{slot} is 8 μs. At the receiver, I set the sampling rate to **500 KHz**, i.e., four times of f_{tx}.



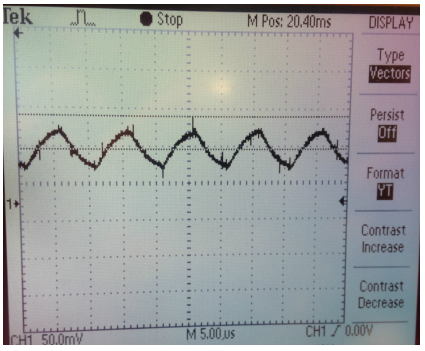
(a) $f_{\text{tx}} = 50$ KHz



(b) $f_{\text{tx}} = 100$ KHz

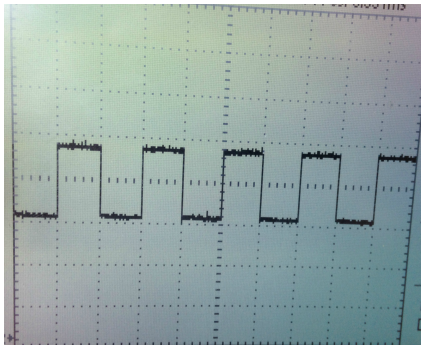


(c) $f_{\text{tx}} = 150$ KHz

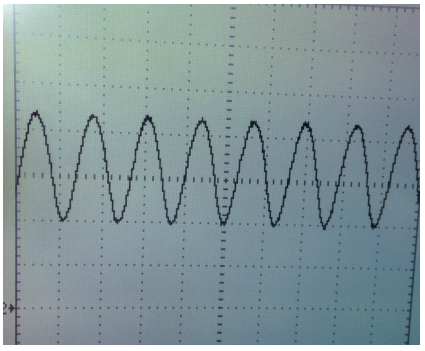


(d) $f_{\text{tx}} = 200$ KHz

Figure 5.1: The oscilloscope snapshot of the received signals at the receiver



(a) $f_{\text{tx}} = 50$ KHz

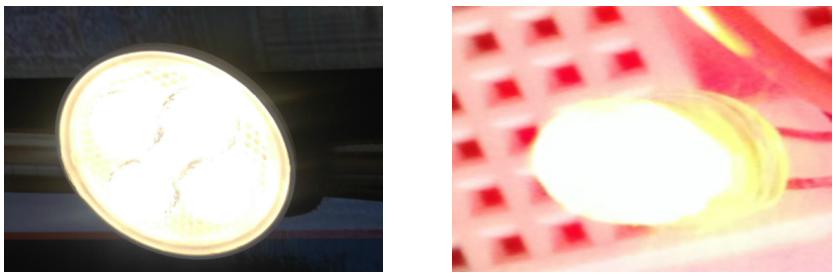


(b) $f_{\text{tx}} = 200$ KHz

Figure 5.2: The snapshot of the transmitted signals at the transmitter

Flickering test

According to the IEEE 802.15.7 standard [31], the minimum frequency to turn on/off an LED is 200 Hz, under which people will not observe flickering. However, according to my findings in the real experiments, the minimum



(a) High power (b) Low power

Figure 5.3: Two LEDs with different lux are used in the experiment

Table 5.1: Users' perception of flickering under different operation frequency f_{tx}. The percentage is the likelihood that the volunteers perceive flickering

Frequency f_{tx}	**Low power LED**	**High power LED**
50 Hz	100%	100%
100 Hz	10%	100%
150 Hz	5%	100%
175 Hz	0%	70%
200 Hz	0%	10%
225 Hz	0%	5%
250 Hz	0%	0%

frequency heavily depends on the type of LEDs. That is, when operating several different types of LEDs under the same frequency, they can bring different degrees of flickering to human beings. Based on this observation, I invite 20 volunteers (10 male and 10 female) between 19 to 41 years old to perform the experiment of users' perception of flickering. The experiments are done with two different types of LEDs, as shown in Fig. 5.3. Under both the scenarios, the LEDs are put horizontally on the table, emitting lights towards the white wall. And the volunteers, who directly view the LEDs, report whether or not they feel flickering when the LEDs are operated at various frequencies. The results are shown in Table 5.1.

Based on the experimental results of the users' perception of flickering, in this thesis, the flickering threshold f_{th} is set to 250 Hz. Since the frequency f_{tx} of turning on/off the LED is set to 125 KHz as presented above, then according to Eq. (3.7), we have

$$N_{\max} = \frac{f_{\text{tx}}}{f_{\text{th}}} = \frac{125000}{250} = 500 \tag{5.1}$$

which means that we can have up to 500 consecutive slots for a super-symbol when performing the multiplexing as presented in Chapter 3.3.

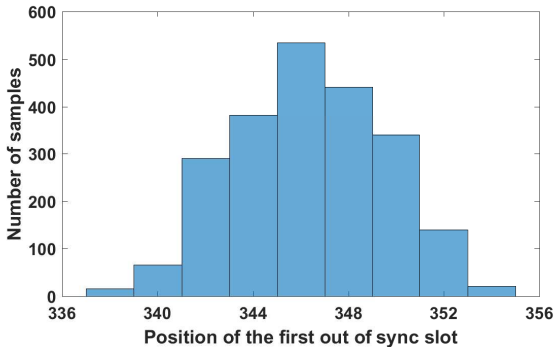


Figure 5.4: Statistics of the position of the first 'out-of-synchronization' slot

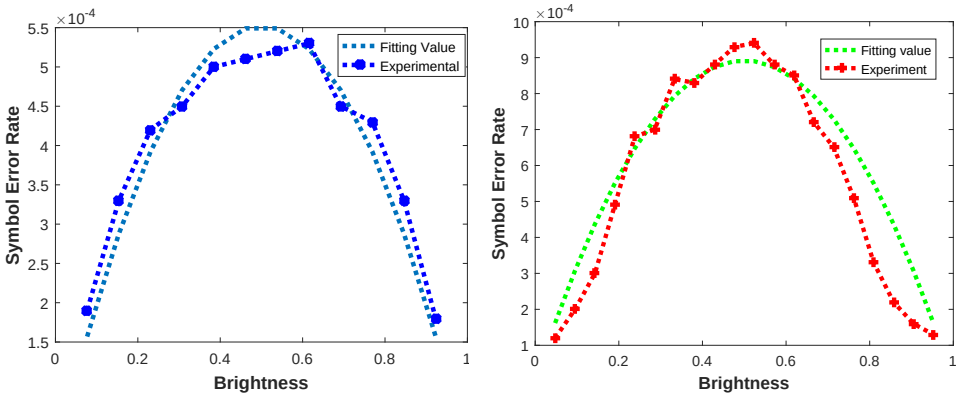
Synchronization test

The PRUs' clocks at the transmitter and the receiver are hardly perfectly synchronized due to the hardware artifact. One clock can tick slightly faster/slower than the other, which raises the synchronization problem between the transmitter and receiver, and consequently, causes accumulating decoding errors. To solve the problem, SmartVLC adjusts the sampling interval at rising/falling edges that is commonly used in the literature [31, 38].

Under the AMPPM modulation scheme, the pattern of consecutive `ON`s or `OFF`s exists. Therefore, the system may be in the risk of losing synchronization before encountering a rising/falling edge. Thus, it is necessary to figure out the position of the first 'out-of-synchronization' slot. In other words, it is equivalent to the maximum running length before losing synchronization. I carry out experiments to test it. The result is shown in Fig. 5.4. It can be observed that the position of the first 'out-of-synchronization' slot is basically subject to normal distribution. Under the worst case, the testbed may start to lose synchronization after consecutive 337 slots, which means the system cannot have 337 consecutive slots of `ON`s or `OFF`s in one symbol (not a super-symbol) or between two symbols. This bound is far from reachable in this thesis (because the number of slots N in one symbol is below 100, from the empirical tests I did in this thesis). Thus, SmartVLC is free from decoding errors caused by clock drift if the approach of adjusting the sampling interval at rising/falling edges is adopted. In subsequent experiments, I will not analyze the decoding errors caused by clock drift.

Symbol error rate and pattern selection

The AMPPM scheme proposed in this thesis depends on the surrounding noise level which affects the symbol error rate P_{SER}. To select the optimal patterns for the super-frame under different dimming levels, we need to determine the flickering threshold f_{th}, the lowest bound of the symbol re-



(a) N=13 (the number of slots in a symbol) (b) N=21 (the number of slots in a symbol)

Figure 5.5: Experimental evaluation of symbol error rate

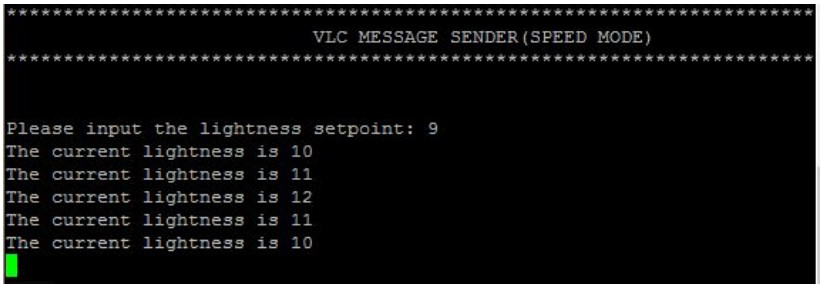
ception rate P_{SRR}, P_1 and P_2, as presented in Eq. (3.5). For the first two parameters, I set f_{th} to 250 Hz according to the flickering test presented above, and P_{SRR} to 0.9991 based on the normal requirement of a reliable communication.

The determination of P_1 and P_2 is based on the following empirical experiment. The receiver is placed at a perpendicular distance of 3.6 m from the transmitter and the ambient noise level is high (ceiling lights are on and window blinds are completely off as shown in Fig. 5.7). Experimental results of the relationship between P_{SER} and brightness under different Ns (N =13 and N = 21) are shown in Fig. 5.5. The results are fitted according to Eq. (3.5) with different values of P_1 and P_2. They are chosen based on the most suitable values related to the actual experimental results for both N = 13 and N = 21. Based on the experimental results, P_1 and P_2 are set to 0.00009 and 0.00008, respectively, for the pattern selections in AMPPM for the rest parts of this chapter.

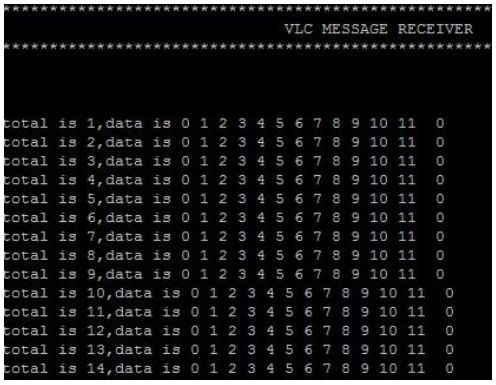
5.2 User interface

Both the user interfaces of the transmitter and the receiver are implemented in the Linux command line for the convenient testing purpose.

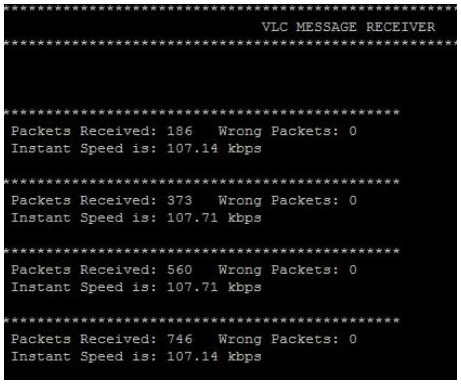
Transmitter: The user interface at the transmitter is presented in Fig. 5.6(a). Initially, the user can choose the target LED lightness (setpoint), scaled from 1 to Max (the specific number of Max depends on experiment requirements and scales from 12 to 300 in this work). Level 1 represents the weakest lightness and level Max represents the strongest one. Consequently, the system adjusts the dimming level of the LED based on the initial lightness and instantaneous ambient light. The sum of the surrounding light and the LED



(a) Transmitter



(b) Receiver: print all the received data



(c) Receiver: only print the summary

Figure 5.6: The user interfaces that are implemented in this project

light is always kept constant, equal to the initially configured value.

Receiver: Two user interfaces are implemented at the receiver, as shown in Fig. 5.6(b) and (c). Depending on the inter-frame interval, one of the two interfaces can be used. If the inter-frame interval is large, the receiver then has the capability to print out the details of the data it receives, as shown in Fig. 5.6(b). If the inter-frame interval is set to be short, then the receiver will only have time to print the summary of the received data, namely, how many frames have been received and what are the instantaneous data rates, as shown in Fig. 5.6(c). The reason behind this is that the `print` function in Linux is quite time-consuming. The receiver will not have enough time to print the details of the received data if the inter-frame interval is too short.

5.3 Static scenario

The system performance of SmartVLC is first evaluated under static scenarios, i.e., fixed ambient light, as shown in Fig. 5.7. More specifically, the window blind and the room ceiling lights are kept on. The relative positions

Figure 5.7: Setup of the static scenario

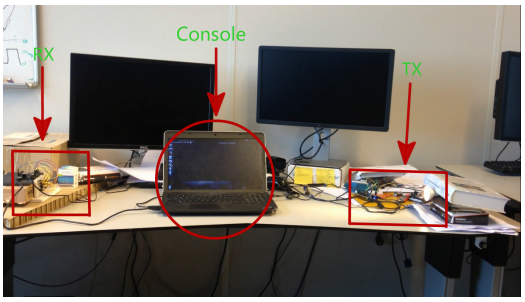


Figure 5.8: Deployment of the transmitter and the receiver

of the transmitter and the receiver are shown in Fig. 5.8.

Throughput versus distance: The throughput of SmartVLC is tested by varying the distance between the transmitter and the receiver. The photodiode and the LED face directly to each other, free from the impact of incidence and irradiation angles. For each plot, the experiment lasts for 30 seconds. SmartVLC reports the instantaneous throughput every second. Three dimming levels (0.18, 0.5, and 0.7) are considered in the experiments. The average throughput from the experiments is plotted in Fig. 5.9. We can observe that SmartVLC maintains its peak throughput at each dimming level at a distance up to 3.6 m. After this distance, the throughput drops dramatically because the received signal strength and the ambient noise level are not sufficient for the receiver to decode the signal. Considering that the distance between a VLC receiver and the LED on the ceiling is usually around 2.2 m in the office, the 3.6 m communication distance supported by SmartVLC is sufficient in reality. Besides, it is easy to conclude that the dimming level of the LED does not affect the communication distance. This is because the brightness of the LED is varied via duty cycle instead of amplitude, thus does not affect the received signal strength under a fixed distance.

Furthermore, Fig. 5.10 shows the box plot of the obtained throughput when the distance is between 0.6 m and 3.6 m, and the dimming level is fixed to 0.5. We can observe small variations in the throughput under each communication distance. There are two reasons behind this observation: *(i)* the instantaneous throughput is calculated every second, thus the number of received frames in each second may be slightly different; *(ii)* some of the frames may wrongly be decoded due to symbol errors. From 3.6 m onward, the throughput starts to drop greatly, because it almost reaches its maximal communication distance.

Throughput versus incidence angle: The performance of SmartVLC is also measured while varying the incidence angle from LED to the photodi-

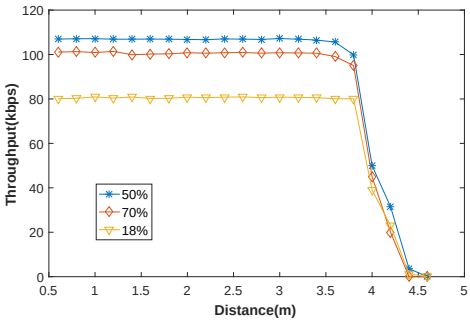


Figure 5.9: Throughput versus communication distance

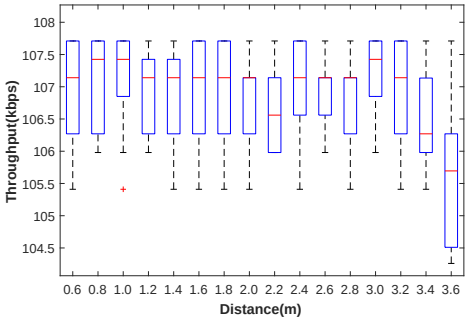


Figure 5.10: Box plot of throughput versus distance (d=0.5)

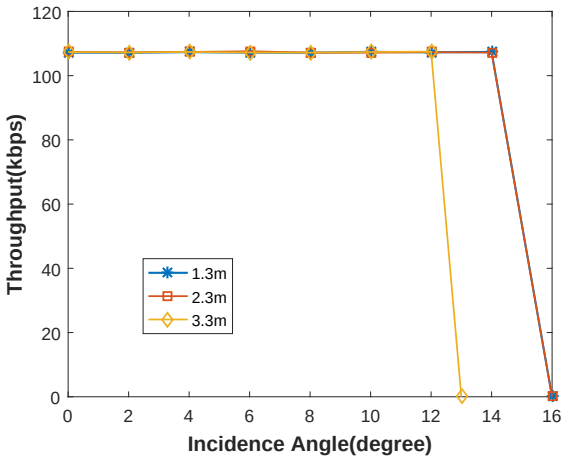


Figure 5.11: Viewing angle at different distance

ode. The perpendicular distance from LED to the photodiode is constant during the variation of incidence angle. I conduct three experiments with different perpendicular distance (1.3 m, 2.3 m, 3.3 m). The results are shown in Fig. 5.11. From these result, we can observe that the throughput drops dramatically after definite degree offset. This is limited by the field of view of the LED. Another observation is that long perpendicular distance possesses shorter cut-off incidence angle because the system already reaches its distance limitation before the incidence angle restricts the system.

Comparison with other schemes: I continue to compare the performance of AMPPM with VOOK and MPPM. The OPPM and VPPM are not considered here for their intrinsic low throughput. In the experiment, 20 discrete dimming levels are considered, ranging from 0.1 to 0.9. The communication distance is fixed to 3 m. For MPPM, N is set to 20, which can satisfy this requirement. For VOOK, the lightness depends on the random

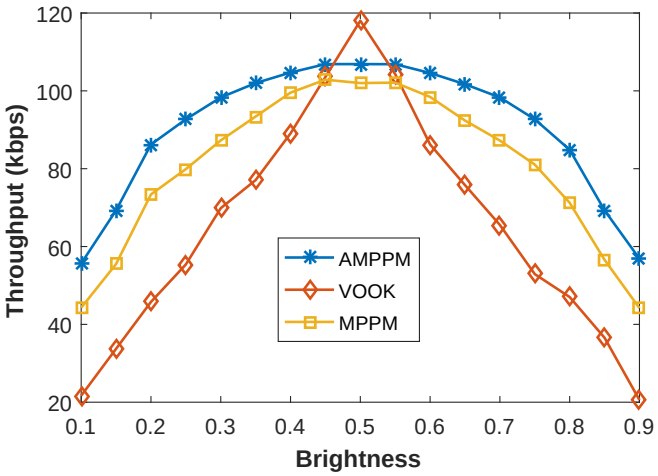


Figure 5.12: Comparison with VOOK and MPPM

input data of the payload. In this section, the probabilities of the bit 0 and 1 are assumed to be equivalent in the payload.

The evaluation results are shown in Fig. 5.12. First, we can observe that AMPPM is the most superior dimming approach. In a narrow dimming level range between 0.47 to 0.53, VOOK is more superior than the other two. That is because VOOK does not bring any overhead when the dimming level is 0.5, meaning that each pulse can represent one bit. However, it loses the advantage quickly under other dimming levels because of the compensation time, and the loss is almost linear. For example, when the dimming level is 0.1 or 0.9, the throughput under VOOK and AMPPM are 20 Kbps and 58 Kbps, which means AMPPM can improve the performance of VOOK by up to 190% when the dimming level is either very low or very high.

For MPPM, it has the similar shape to AMPPM but its throughput under each dimming level is smaller than that of AMPPM, and its slope is also steeper. This is because AMPPM only selects the optimal pattern in each discrete lightness. Overall, it is without a doubt to conclude that the AMPPM in SmartVLC outperforms both VOOK and MPPM.

5.4 Dynamic scenario

This section presents the system performance of SmartVLC under dynamic scenario, i.e., by varying the ambient light through window blind, as shown in Fig. 5.13. The intensity of the ambient light is associated with the position of the window blind.

5.4.1 Perception of light changes

I first conduct the study of users' perception of light to test whether SmartVLC is flickering-free to users or not. In this experiment, I place the LED towards the test users who are instructed to view the LED in two different manners as shown in Fig. 5.14: *(i)* direct viewing, where the users directly look at the LED; *(ii)* indirect viewing, where the users indirectly look at the LED and judge the LED from its illuminance on the wall. I change the **perceived light percentage** as mentioned in Section 3.4. The users are not aware of the dimming level resolution in advance and they are asked to tell me whether they can perceive the lightness variation or not. I invite 20 volunteers (10 male and 10 female) between 19 to 41 years old into the users' perception experiment, the same group of volunteers for the flickering study. The experiments are conducted in three different ambient light conditions: *(1)* a sunny day with indoor artificial light on (8900–9760 lux); *(2)* a sunny day without any indoor artificial light on (7960–8200 lux); *(3)* the window blind is on and without any indoor artificial light (12–21 lux). The experimental results are shown in Table 5.2 and 5.3.

According to the results, we observe that under the same viewing manner, weaker ambient light cause users incline more to observe the LED light changes. This is because human beings enlarge their pupil in dark conditions. On the other hand, under the direct viewing, users are easier to feel the LED's flickering when it changes its dimming level. Under the worst case, the resolution of the dimming level should be maintained around 0.003.

5.4.2 Adaptation to ambient light

Maximum communication distance: The ambient light condition affects the maximal communication distance. I change the window blind to two extreme positions. In one scenario, the window blind is *at the top* as in

Figure 5.13: Setup of the dynamic scenario

Table 5.2: User perception of light changes under indirect viewing

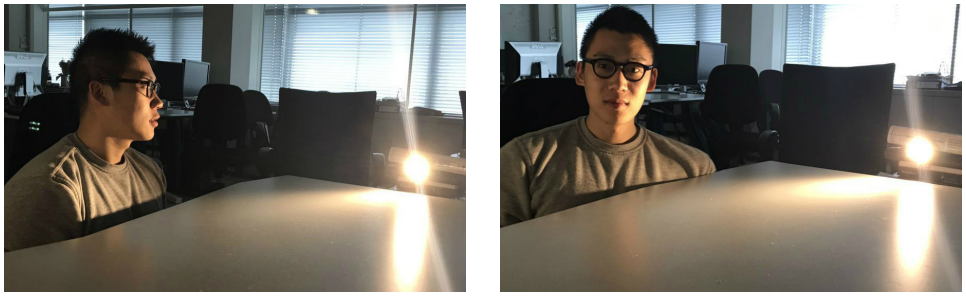
Resolution	Light level 1	Light level 2	Light level 3
4%	0%	0%	0%
5%	0%	15%	20%
6%	30%	50%	90%
7%	100%	100%	100%
8%	100%	100%	100%

Table 5.3: User perception of light changes under direct viewing

Resolution	Light level 1	Light level 2	Light level 3
0.3%	0%	0%	0%
0.4%	0%	0%	15%
0.5%	5%	30%	50%
0.6%	40%	75%	100%
0.7%	100%	100%	100%
0.8%	100%	100%	100%

the previous section. In the other one, the window blind is *at the bottom* but with the ceiling light on. The LED dimming level is fixed to 0.5 in each test. The position between the LED and the receiver is changed, and the average throughput is measured at each spot. The results are shown in Fig. 5.15. We can observe that SmartVLC can reach a longer communication distance when the window blind is *at the bottom*, due to the higher SNR.

Dynamic throughput: I calibrate the dimming level of SmartVLC to the lowest (0.1) when the blind is *at the top* and to the highest (0.9) when the blind is *at the bottom*. Nine positions of the window blind are used, resulting nine different ambient light levels (0–8, from the weakest to the strongest).



(a) Direct viewing (b) Indirect viewing

Figure 5.14: Users' perception of light changes

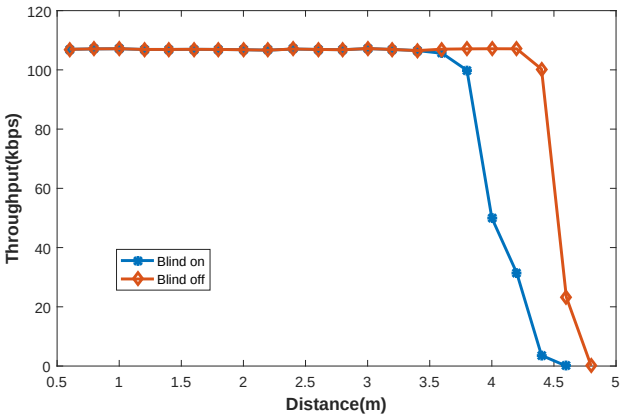


Figure 5.15: Impact of the ambient light on communication distance

Specifically, different weather conditions are also taken into account, which bring different ambient light variation range. Three scenarios are considered: *(1)* cloudy day (320–2130 lux variation); *(2)* sunny afternoon (450–2990 lux variation); *(3)* sunny morning (the testing room is faced to the east, thus this is the strongest ambient light moment, with 660–9780 lux variation). The distance between the transmitter and the receiver is fixed to 3 m.

The relationship between the position of the window blind (ambient light level) and throughput is shown in Fig. 5.16. The shapes of the first two scenarios are basically subject to the theoretical shapes of the AMPPM as shown in Chapter 3.3. However, we observed that the third scenario lacks in throughput when the position of the window blind is at the identical position compared to the first two scenarios. That is because the third scenario possesses the lower SNR, resulting in higher symbol error rate. On the other hand, the overall curve of the third scenario is not symmetrical

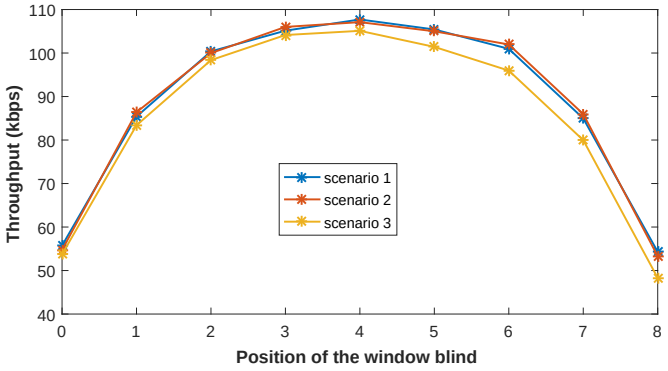


Figure 5.16: Throughput under different ambient light conditions

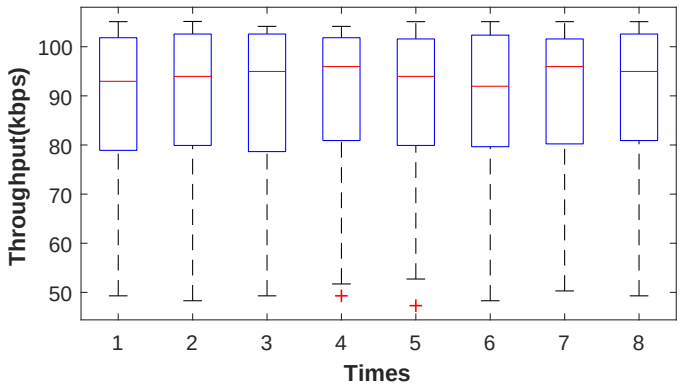


Figure 5.17: Evaluation results under the dynamic scenario

as the first two scenarios, which is also because of the difference of SNR. In the first scenario, though the symmetrical positions (e.g., 2 and 6) do not possess the same SNR, the gap is not significant. For the third scenario, since the SNR gap at the symmetrical position is quite large, the difference of throughput is consequently large.

Finally, the performance of SmartVLC is evaluated in a completely dynamic scenario, where the window blind continuously goes down from the *top* to the *bottom* in a duration of one minute, stimulating the changes of ambient light changes in a day. During this one-minute experiment, the system records an average of 60 samples of the instantaneous system throughput. The one-minute experiment is repeated for eight times. Box plot of the results is shown in Fig. 5.17. We can observe from the figure that although there is a large gap among the achieved throughput due to SmartVLC's adaptation to ambient light changes, the system can achieve a high through during most of the time of the experiment. However, if the users are willing to obtain throughput rather than precise lightness from the smart lighting, users can sacrifice the lightness variation range a bit (only keep the dimming level near 0.5, e.g., from 0.25 to 0.75) and maintain high throughput.

Chapter 6

Conclusion

The co-design of smarting lighting and visible light communication is an important topic but still in its immature stage, especially missing the research efforts in the system level. This thesis proposed the system SmartVLC that could optimize the data rate and at the same time adjust LEDs' illumination freely and flickering-free. Its key compositions included *i)* the *multiplexing* approach to provide fine-grained dimming levels; *ii)* the *AMPPM* scheme to optimize the date rate under each dimming level; and *iii)* the *adaptation* algorithm to change LEDs' illumination smoothly based on the surrounding ambient light. SmartVLC has been implemented with off-the-shelf electronic devices and the low-cost but powerful platform BeagleBone Black. Experimental results have demonstrated SmartVLC's superior performance over the state-of-art solutions such as MPPM and OOK with compensation time. Going forward, I believe SmartVLC's design principles could intrigue more system-level research efforts on co-designing smart lighting and communication for the visible light networks. The implementations of SmartVLC using the PRUs of BBB would also benefit other low-cost solutions for visible light communication networks, such as the OpenVLC project.

The proof-of-concept of SmartVLC has been presented and validated in this thesis. For the *future work*, I plan to extend its functionalities from two directions: *i) advanced front-end.* The current bottleneck of SmartVLC is the limited bandwidth of the LED used at the transmitter. I plan to change it to a specialized LED with a higher bandwidth. Besides, the current front-end is built on the breadboard. It will be improved by the PCB design to increase the system's stability; *ii) uplink.* To improve users' experience on both illumination and communication, real-time monitoring of the system's performance and the surrounding environment changes at the mobile nodes is important. I plan to add the uplink to the SmartVLC system using radio-frequency based techniques, such as the BLE, to send the monitoring results to the transmitter (access point).

Appendix A

Coding algorithms

Procedure 1 Decoding algorithm

Input: 1) N, the number of slots in one time symbol;
2) K, the number of slots with light pulses in one symbol;
3) IP, the input code word in AMPPM form;

Output: OP, the output code word in binary form.

```
OP = 0;
t = 1;
i = 1;
while i <= N do
  if IP[i] == 0 then
    OP = OP + C(N-i, K-t)
  else
    t = t + 1
  end if
  if t > K then
    break;
  end if
  if i + 1 + (K - t) == N then
    break;
  end if
  i = i + 1
end while
```

Procedure 2 Encoding algorithm

Input: 1) N, the number of slots in one time symbol;
2) K, the number of slots with light pulses in one symbol;
3) IP, the input code word in binary form;

Output: OP, the output code word in AMPPM binary form.

```
tmp = IP;
t = 1;
i = 1;
while i <= N do
  if tmp >= C(N-i, K-t) then
    OP[i] = 0
    tmp = tmp - C(N-i, K-t)
  else
    OP[i] = 1
    t = t + 1
  end if
  if t > K then
    for x = i + 1 to N do
      OP[x] = 0
    end for
    break;
  end if
  if i + 1 + (K - t) == N then
    for x = i + 1 to N do
      OP[x] = 1
    end for
    break;
  end if
  i = i + 1
end while
```

Appendix B

Demo video

The demo video of this project is available here: `https://goo.gl/S8fn5f`. For a better viewing experience, please download it and then watch it offline.

Bibliography

[1] Arduino. *https://www.arduino.cc/*.

[2] Beaglebone black. *http://beagleboard.org/*.

[3] Raspberry pi. *https://www.raspberrypi.org/*.

[4] Universal software radio peripheral. *https://www.ettus.com/*.

[5] Warp project. *http://warpproject.org*.

[6] Wiki start gnu radio. *http://gnuradio.org/redmine/projects/gnuradio/wiki*.

[7] Ieee standard for local and metropolitan area networks–part 15.7: Short-range wireless optical communication using visible light. *IEEE Std 802.15.7-2011*, pages 1–309, Sept 2011.

[8] Mostafa Z Afgani, Harald Haas, Hany Elgala, and Dietmar Knipp. Visible light communication using ofdm. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, pages 6–pp. IEEE, 2006.

[9] Jean Armstrong and AJ Lowery. Power efficient optical ofdm. *Electronics Letters*, 42(6):370–372, 2006.

[10] Bo Bai, Zhengyuan Xu, and Yangyu Fan. Joint led dimming and high capacity visible light communication by overlapping ppm. In *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*, pages 1–5. IEEE, 2010.

[11] Israel Bar-David and Gideon Kaplan. Information rates of photon-limited overlapping pulse position modulation channels. *IEEE Transactions on Information Theory*, 30(3):455–464, 1984.

[12] Samuel M Berman, Daniel S Greenhouse, Ian L Bailey, Robert D Clear, and Thomas W Raasch. Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources. *Optometry & Vision Science*, 68(8):645–662, 1991.

[13] Lucio Ciabattoni, Alessandro Freddi, Gianluca Ippoliti, Maurizio Marcantonio, Davide Marchei, Andrea Monteriu, and Matteo Pirro. A smart lighting system for industrial and domestic use. In *Mechatronics (ICM), 2013 IEEE International Conference on*, pages 126–131. IEEE, 2013.

[14] Craig DiLouie. *Advanced lighting controls: energy savings, productivity, technology and applications*. The Fairmont Press, Inc., 2006.

[15] Svilen Dimitrov and Harald Haas. *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge University Press, 2015.

[16] John Gancarz, Hany Elgala, and Thomas DC Little. Impact of lighting requirements on vlc systems. *IEEE Communications Magazine*, 51(12):34–41, 2013.

[17] Liane Grobe, Anagnostis Paraskevopoulos, Jonas Hilt, Dominic Schulz, Friedrich Lassak, Florian Hartlieb, Christoph Kottke, Volker Jungnickel, and Klaus-Dieter Langer. High-speed visible light communication systems. *IEEE Communications Magazine*, 51(12):60–66, 2013.

[18] Tian Hao, Ruogu Zhou, and Guoliang Xing. Cobra: color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 85–98. ACM, 2012.

[19] Lennart Klaver and Marco Zuniga. Shine: A step towards distributed multi-hop visible light communication. In *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*, pages 235–243. IEEE, 2015.

[20] Toshihiko Komine and Masao Nakagawa. Fundamental analysis for visible-light communication system using led lights. *IEEE transactions on Consumer Electronics*, 50(1):100–107, 2004.

[21] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458. ACM, 2014.

[22] Hoa Le Minh, Dominic O'Brien, Grahame Faulkner, Lubin Zeng, Kyungwoo Lee, Daekwang Jung, and YunJe Oh. High-speed visible light communications using multiple-resonant equalization. *IEEE Photonics Technology Letters*, 20(14):1243–1245, 2008.

[23] Kwonhyung Lee and Hyuncheol Park. Modulations for visible light communications with dimming control. *IEEE photonics technology letters*, 23(16):1136–1138, 2011.

[24] Rimhwan Lee, Kyungsu Yun, Jong-Ho Yoo, Sung-Yoon Jung, and Jae Kyun Kwon. Performance analysis of m-ary ppm in dimmable visible light communications. In *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pages 380–383. IEEE, 2013.

[25] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. Epsilon: A visible light based positioning system. In *NSDI*, pages 331–343, 2014.

[26] Tianxing Li, Chuankai An, Zhao Tian, Andrew T Campbell, and Xia Zhou. Human sensing using visible light communication. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 331–344. ACM, 2015.

[27] Luigi Martirano. A smart lighting control to save energy. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, volume 1, pages 132–138. IEEE, 2011.

[28] Hyuncheol Park and John R Barry. Trellis-coded multiple-pulse-position modulation for wireless infrared communications. *IEEE transactions on communications*, 52(4):643–651, 2004.

[29] Samuel David Perli, Nabeel Ahmed, and Dina Katabi. Pixnet: interference-free wireless links using lcd-camera pairs. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 137–148. ACM, 2010.

[30] M Rahaim, A Miravakili, T Borogovac, TDC Little, and V Joyner. Demonstration of a software defined visible light communication system. In *the 17th Annual International Conference on Mobile Computing and Networking, Mobicom2011*, 2011.

[31] Sridhar Rajagopal, Richard D Roberts, and Sang-Kyu Lim. Ieee 802.15. 7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3), 2012.

[32] Mark S Rea. Illuminating engineering society of north america. *The IESNA lighting handbook*, 2000.

[33] K. Sato, T. Ohtsuki, I. Sasase, and S. Mori. Performance analysis of (m, 2) mppm with imperfect slot synchronization. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, pages 765–768 vol.2, 1993.

[34] Stefan Schmid, Giorgio Corbellini, Stefan Mangold, and Thomas R Gross. Led-to-led visible light communication networks. In *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, pages 1–10. ACM, 2013.

[35] Abu Bakar Siddique and Muhammad Tahir. Joint rate-brightness control using variable rate mppm for led based visible light communication systems. *IEEE Transactions on Wireless Communications*, 12(9):4604–4611, 2013.

[36] Hidemitsu Sugiyama, Shinichiro Haruyama, and Masao Nakagawa. Brightness control methods for illumination and visible-light communication systems. In *Wireless and Mobile Communications, 2007. ICWMC'07. Third International Conference on*, pages 78–78. IEEE, 2007.

[37] Hisayoshi Sugiyama and Kiyoshi Nosu. Mppm: A method for improving the band-utilization efficiency in optical ppm. *Journal of Lightwave Technology*, 7(3):465–472, 1989.

[38] Zhao Tian, Kevin Wright, and Xia Zhou. The darklight rises: Visible light communication in the dark. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 2–15. ACM, 2016.

[39] Dobroslav Tsonev, Stefan Videv, and Harald Haas. Light fidelity (li-fi): towards all-optical networking. In *SPIE OPTO*, pages 900702–900702. International Society for Optics and Photonics, 2013.

[40] Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. Openvlc: Software-defined visible light embedded networks. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 15–20. ACM, 2014.

[41] Li-Che Wu and Hsin-Mu Tsai. Modeling vehicle-to-vehicle visible light communication link duration with empirical data. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 1103–1109. IEEE, 2013.

[42] Shaoen Wu, Honggang Wang, and Chan-Hyun Youn. Visible light communications for 5g wireless networking systems: from fixed to mobile communications. *IEEE Network*, 28(6):41–45, 2014.

[43] Bo Xie, Guang Tan, and Tian He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 211–223. ACM, 2015.

[44] Shun-Hsiang You, Shih-Hao Chang, Hao-Min Lin, and Hsin-Mu Tsai. Visible light communications for scooter safety. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 509–510. ACM, 2013.

[45] Shun-Hsiang Yu, Oliver Shih, Hsin-Mu Tsai, Nawaporn Wisitpongphan, and Richard D Roberts. Smart automotive lighting for vehicle safety. *IEEE Communications Magazine*, 51(12):50–59, 2013.

[46] Fahad Zafar, Dilukshan Karunatilaka, and Rajendran Parthiban. Dimming schemes for visible light communication: the state of research. *IEEE Wireless Communications*, 22(2):29–35, 2015.

[47] Chi Zhang, Josh Tabor, Jialiang Zhang, and Xinyu Zhang. Extending mobile interaction through near-field visible light sensing. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 345–357. ACM, 2015.

[48] Xia Zhou and Andrew T Campbell. Visible light networking and sensing. In *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pages 55–60. ACM, 2014.