Article | Received 9 June 2025; Revised 14 November 2025; Accepted 1 December 2025; Published 15 December 2025
https://doi.org/10.55092/esp20250006

A study of SerDes logic for visible light communication using 8B13B code

Tokio Yukiya*, Nobuo Nishimiya and Takayuki Uchida

Graduate School of Engineering, Tokyo Polytechnic University, Atsugi, Japan

* Correspondence author; E-mail: yukiya@eng.t-kougei.ac.jp.

Highlights:

- FPGA-based SerDes with 8B13B coding achieved 3.48 Mbit/s in a VLC system.
- A simple FPGA–Raspberry Pi design enables a low-cost VLC system.
- Stable VLC over 3 m demonstrated under strong ambient light conditions.

Abstract: Visible light communication (VLC) has been increasingly implemented in data transmission to overcome the limitations faced by radio wave communication. However, obtaining specialized equipment, particularly serializers and deserializers, remains a significant challenge for the realization of the VLC systems. In this study, we developed an 8B13B coding scheme for VLC that enables reliable synchronization and effectively addresses pulse-width variations. The proposed serializer and deserializer (SerDes) logic was implemented in Verilog hardware description language (Verilog HDL) and deployed on a field-programmable gate array (FPGA), which interfaces with Raspberry Pi via the serial peripheral interface (SPI), forming a simple yet effective communication system. Although the overall communication speed relies on the data transfer frequency between the FPGA and Raspberry Pi, the bit rate was 3.48 Mbit/sec. We evaluated the communication quality of the system in environments with ambient light interference and achieved stable communication over a distance of approximately 3 m between the light emitting diode (LED) light source and receiver. The ability to use the VLC with the widely popular and commonly used Raspberry Pi is expected to promote the advancement of research and development of applications utilizing this communication system.

Keywords: 8B13B; FPGA; deserializer; Raspberry Pi; SerDes; serializer; serial peripheral interface; visible light communication; wireless communication; VLC; DDPWS

1. Introduction

Extensive research is being conducted on VLC, and is expected to be applied across a wide range of fields [1–4]. Figure 1 concisely depicts the history of VLC presented by Gupta *et al.* [3]. VLC is typically applied for the transmission of information to indoor devices using the light emitted from LED lamps. These applications include outdoor scenarios such as building-to-building communication, intelligent transport systems (ITS), and communication between ships or even underwater.

Copyright©2025 by the authors. Published by ELSP. This work is licensed under Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

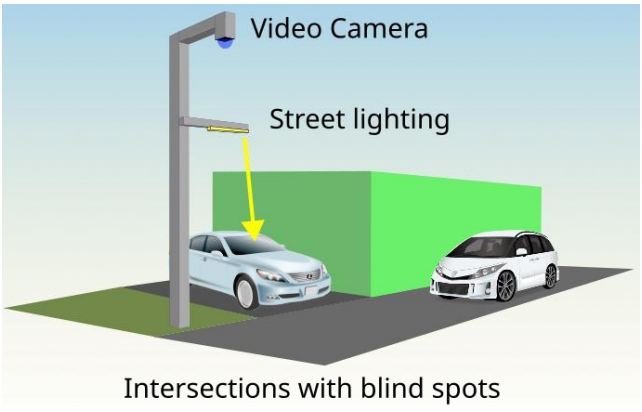

Figure 1. Schematic of the proposed VLC system for intersections with blind spots.

In this study, we focused on the application of VLC in ITS. The typical use cases of VLC in ITS include the transmission of road information from traffic signals to vehicles, vehicle-to-vehicle communication, and communication between streetlights and vehicles [5,6]. Several studies have been conducted on these applications since the year 2000 [7,8].

Vieira *et al.* recently proposed a multi-intersection traffic control system that employed VLC with streetlights, headlights, and traffic signals to improve the efficiency of urban intersections [9–11].

In the application of VLC to ITS, the distance between the traffic lights or street lamps and vehicles can extend to several meters. Consequently, LED drivers must be employed to ensure high output power, while the receivers must exhibit high sensitivity. Additionally, a narrow-beam approach such as using a spotlight is unsuitable since the information must be transmitted simultaneously to multiple vehicles. Furthermore, stable communication must be maintained even under strong ambient sunlight.

El-Garhy *et al.* evaluated the signal-to-noise ratio (SNR) while considering the distance between the traffic lights and vehicles [12]. Seminara *et al.* developed a VLC-enabled traffic light and receiver, and experimentally evaluated their performance [13]. Additionally, they enhanced the gain of the receiver using a lens and discussed the off-axis characteristics of the optical setup.

Various modulation methods were developed for VLC. Among them, orthogonal frequency division multiplexing (OFDM) is currently being standardized and is considered a highly efficient communication method. An example of ITS application was also reported, where 45–55 Mbit/sec communication was achieved using DC-biased optical orthogonal frequency division multiplexing (DCO-OFDM) comprising a 4 × 5 LED array and a camera-type receiver configuration [14]. VLC using OFDM, also known as Li-Fi, was standardized as IEEE 802.11bb as of 2025 [15,16]. In IEEE 802.11bb, the minimum and maximum throughput of LiFi is 10 Mbit/sec and 9.6 Gbit/sec, respectively [17]. OFDM is considered ideal for indoor applications owing to its high spectral efficiency and effective utilization of available frequencies.

However, ITS applications require longer communication distances, along with high-power light sources and high-sensitivity receivers. OFDM requires high linearity and wide dynamic range for both the transmitters and receivers since the signals are divided and superimposed across several subcarriers. The high peak-to-average power ratio (PAPR) of OFDM also increases the complexity of the power supply design [18–20].

On-off keying (OOK), standardized in IEEE 802.15.7 and summarized by Rajagopal *et al.*, has also gained considerable attention as a simpler modulation scheme. OOK employed simple switching control,

making it more suitable for LEDs with low linearity and enabling a simpler circuit than OFDM [21]. OOK can achieve longer transmission distances than OFDM since the rated current flows when the LED is turned on. Furthermore, it exhibits better performance in noisy environments, making it suitable for outdoor usage.

Pulse position modulation (PPM) was standardized and used since the early stages of VLC. In this method, the position of a pulse within a 4-bit segment was used to encode information, transmitting 2 bits per 4-bit segment, presenting a coding efficiency of 0.5. Mietzner *et al.* determined whether simultaneous communication can be achieved using nested PPM, which achieves higher transmission efficiency than conventional PPM, with both fast and slow response receivers [22]. They reported that the simplicity of the system can facilitate the commercialization of VLC.

Nawaz *et al.* proposed a Manchester code-based system for infrastructure-to-vehicle-to-vehicle (I2V2V) configuration [23]. This system achieved a maximum transmission distance of 30 m under a transmission rate of 115.2 kBd and a packet error rate of 1×10^{-5}. Ricci *et al.* considered IEEE 802.15.7 and implemented a system in FPGA that also used Manchester codes [24,25]. They employed a 12-bit AD converter as the interface of the receiver, thereby achieving synchronization by calculating the phase and enabling frequency shift and jitter up to 1%. In the same research group, Caputo *et al.* analyzed the difference in the operating characteristics between full-duplex and half-duplex communication in VLC compliant with the IEEE 802.15.7 standard [26]. They implemented this system in FPGA, obtained experimental data, and evaluated its performance, including the reflected light.

Run-length-limited coding schemes such as 4B6B and 8B10B are essential in VLC. Mambou *et al.* redesigned the IEEE 802.15.7-defined 4B6B code to improve the error correction capabilities. Their proposed scheme outperforms the conventional 4B6B, extended Miller code, Manchester, 5B10B, and (0,4) 2/3 RLL [27]. Dhannoon *et al.* proposed a 3B6B scheme that maintains a brightness of 50% and achieves high free distance [28].

Choudhury proposed a system that incorporated a high-pass filter on the receiver side to mitigate the noise from fluorescent and AC-driven LEDs [29]. This system employed 8B10B modulation over a 10 Mbit/sec VLC link, and maintained a BER of 10^{-5} up to noise frequencies of 500 kHz when subjected to a noise tone of 100 kHz.

Chen *et al.* considered semantic communication for transmitting images by using VLC to prevent rapid image deterioration due to poor communication quality [30]. To this end, they employed wavelength division multiplexing with four LEDs. Each LED was modulated with 8B10B and information is transmitted to the receiver at four different wavelengths. They have demonstrated that optical signals can be transmitted through a Fresnel lens up to 3 meters at 1 Gbit/sec.

The coding efficiencies of 4B6B and 8B10B are 0.67 and 0.8, respectively. The run-length control ensures DC balance and suppresses flickering in VLC, which requires stable light intensity. Furthermore, it is required for maintaining synchronization.

Residual carriers can be eliminated to operate LEDs using methods such as OOK at high switching speeds. They can be easily eliminated by applying a reverse bias; however, this increases the complexity of the circuit. Yan *et al.* proposed an LED driver with an additional circuit to eliminate the residual carriers [31]. This LED driver employed a commercial-based phosphorescent white-light LED and comprises a transistor that eliminates residual carriers apart from the switching FET. They reported that this extends the cutoff frequency from 3 MHz to 80 MHz.

In VLC using "Non-Return-to-Zero-type OOK", the pulse width may change due to the effect of the rise time/fall time at high switching rates and weak signal strength, and the bits may change. This is known as data dependent pulse width shrinkage (DDPWS). Although removing residual carriers from LEDs using reverse biasing enables high-speed switching, some commercially available LEDs comprise built-in protection circuits (e.g., Zener diodes), which may not be applicable to all LEDs [32].

In this study, we propose an 8B13B coding scheme to adapt VLC for ITS applications. Figure 1 depicts the proposed VLC system, which transmits video from a camera installed above an intersection with limited visibility to vehicles, thereby reducing blind spots. The blind spots are expected to remain an issue even in autonomous driving. In the future, this system could function as a communication method for localized traffic control at intersections.

Currently, we are considering video transmission of the intersection information. The transmission rate is estimated to be several hundred kilobits per second with a resolution of 640 × 480, 30 Frames Per Second (FPS), and H.264 compression. A streetlight is used as the light source, and the communication distance is approximately 5–5.5 m, based on the height of the traffic light. The system must maintain stable operation under strong sunlight. Wide-area illumination is required since the relative position of the vehicle and light source may vary.

Although extensive research has been conducted on VLC applications for ITS for over 20 years, large-scale practical deployment has yet to be achieved. This challenge is attributed to the fact that the information receiver is only set up after the light source responsible for transmitting data has been installed. The widespread adoption of VLC systems requires both the transmitters and receivers to be installed at the users' discretion. Therefore, we consider affordability to be a crucial factor in facilitating their adoption.

The proposed 8B13B code does not achieve transmission rates comparable to OFDM, but presents a maximum speed of 3 Mbit/sec. This code follows the RZ format, and is considered to be robust against the pulse distortion caused by DDPWS and other factors since it depends only on the rising edge of the pulse. The VLC system utilizing this code comprises the readily available FPGA and Raspberry Pi components. A Raspberry Pi can help in constructing this system at a low cost owing to its compact size and ability to integrate a camera. Furthermore, on the receiving side, the Raspberry Pi supports video display and augmented reality, making it a viable platform for analyzing various presentation methods. In this study, we propose the 8B13B code, implement a serializer/deserializer utilizing this code in FPGA, and validate its functionality. Additionally, to promote further research in VLC, we aim to release the serializer/deserializer code written in Verilog HDL as open source.

2. Methods

Figure 2 depicts the block diagram of a VLC system, which integrates an FPGA and a single-board computer. Two pairs, each consisting of an FPGA board (DE0-Nano-SoC, manufactured by Terasic) and a single-board computer (Raspberry Pi 4), are used for transmission and reception. By installing a camera on the transmitting Raspberry Pi and a display on the receiving Raspberry Pi, video transmission and reception can be performed as shown in Figure 1.

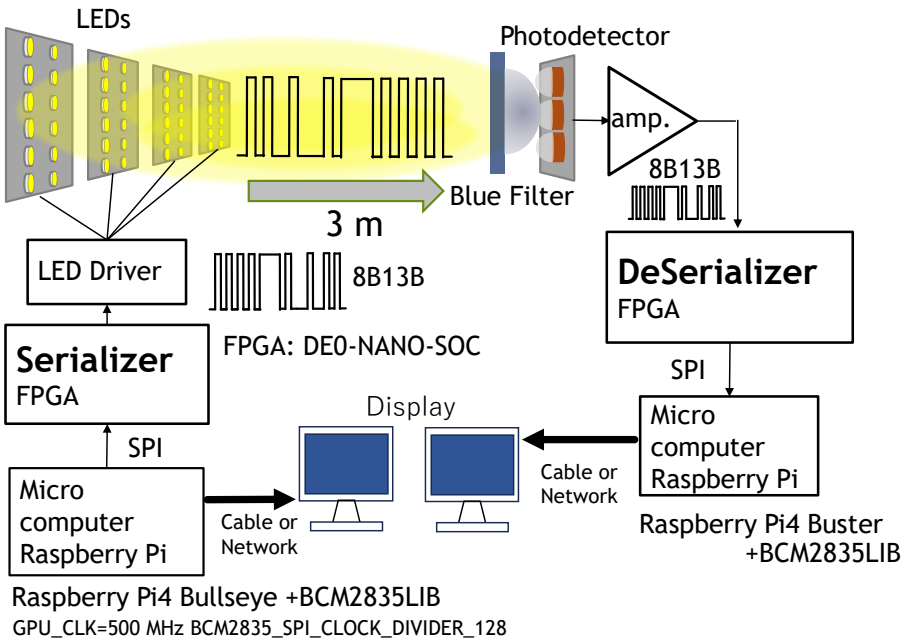

Figure 2. Block diagram of a visible light communication system using Raspberry Pi and FPGA.

The transmitter serializer converts the transmitted data from Raspberry Pi into an 8B13B code, and the LED driver converts these signals into an optical output. An optical bandpass filter (BrightLine 452/45) is installed in the optical detector to eliminate the fluorescence generated by the white LED. Subsequently, the signal is decoded using the Deserializer module and transmitted to the Raspberry Pi via SPI.

The Raspberry Pi operates on a 64-bit version of the Bullseye OS, and the BCM2835 library is used for SPI communication [33].

The data transfer between the FPGA and Raspberry Pi is achieved via the SPI. To ensure precise timing, the FPGA transmits a data request signal to the Raspberry Pi to facilitate SPI communication.

The Serial Clock (SCLK) for SPI communication in Raspberry Pi 3 and 4 depends on the Graphics Processing Unit (GPU) clock. Therefore, if the GPU clock frequency changes, the SCLK also changes. To avoid this, the GPU clock of the Raspberry Pi is fixed at 500 MHz, and the SPI clock is derived by dividing this clock by 128. The GPU clock must be considered when using Raspberry Pi 3. The maximum GPU clock for Raspberry Pi 3 is 400 MHz, which produces a longer SCLK cycle when compared with Raspberry Pi 4.

2.1. LED driver

The output of the LED driver must be increased to transmit light signals over a wide area and a range of several meters. High-power white LEDs comprise a series connection. This is not used in this study because the response characteristics do not satisfy the desired conditions. In this study, we aimed to increase the speed by installing commercial-based multiple single phosphorescent white-light LEDs in parallel. Conversely, connecting many LEDs in parallel requires a large circuit area. We considered a system in which the serial signals are divided by a signal distributor and then input into the LED switching circuits units.

Figure 3 depicts the block diagram of an LED driver comprising a distribution circuit and LED switching circuits units. The distribution circuit employs Not AND (NAND) logic to divide the serial signal

into four parts, which are input to the low-voltage differential signaling (LVDS) dual line driver with dual line receivers (Texas Instruments DS90LV049). To minimize variations in the characteristics of the transmission line, the distribution circuit and the LED switching circuit unit are connected using a repurposed 2-meter Category 6 twisted-pair LAN cable.

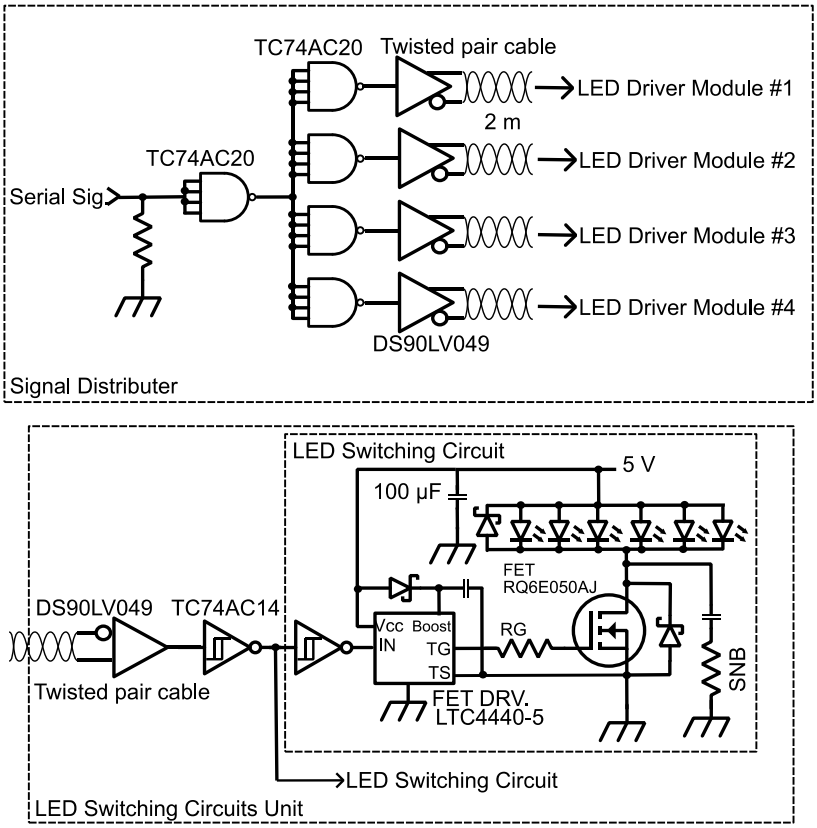

Figure 3. Block diagram of a LED driver.

LED switching circuit unit is equipped with six 1 W-class white LEDs (LENOO THEM-CLWX). An N-channel MOSFET (Rohm RQ6E050AJ) switches six LEDs arranged in a parallel configuration; this unit operates at 5 V. The gates of each transistor are driven by an FET driver (Analog Devices LTC4440-5). Two of these sets are placed on a single board to drive 12 LEDs as one module. This circuit did not have a mechanism to eliminate the residual carriers [31]. Four of these modules are used to control a total of 48 LEDs. However, in this study, there are no lenses attached to the LED driver to increase directivity; this corresponds to the radiation characteristics of the LEDs.

2.2. Optical detector

Figure 4 depicts a block diagram of the receiver, which is a crucial component of a VLC system. A p-type, intrinsic, n-type (PIN) photodiode with a large reception area can be used to improve the sensitivity. However, increasing the area presents higher capacitance, which reduces the response speed. Since we consider blue light emitted by blue LEDs, the sensitivity of the PIN photodiode is lower when compared with the near-infrared region. Therefore, a receiver with a low noise, high gain, and fast response is desirable for VLC.

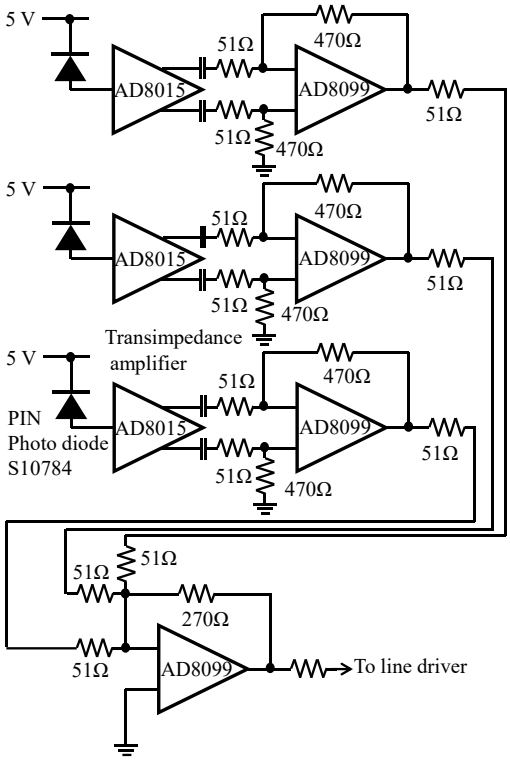

Figure 4. Block diagram of a receiver using three PIN photodiodes.

In this study, we used three S10784 PIN photodiodes (Hamamatsu Photonics) to enhance the reception sensitivity. The signals from the PIN photodiodes, reverse-biased at 5 V, are amplified using a transimpedance amplifier (Analog Devices AD8015). After the DC component is cut, the signals are further amplified using a subtraction circuit comprising an operational amplifier (Analog Devices AD8099). The signal-to-noise ratio (S/N) can be improved by summing these signals.

Grooves were formed on a copper-clad board to create a circuit pattern for implementation. Twisted-pair wires were used between each transimpedance amplifier and the subtraction circuit to avoid external noise. The twisted pairs were adjusted to approximately 5 cm to ensure that they were equal in length. The signal from the addition circuit was transmitted via a coaxial cable to a binarization circuit located near the FPGA by passing through a line driver.

2.3. Serialize/Deserializer on FPGA

Figure 5 presents a block diagram of the SerDes logic circuit implemented on the FPGA. The SerDes logic circuit, written in Verilog HDL, is presented as DE0NanoSoc_TXRX.v in Appendix (i) and as spi_slave.v in Appendix (ii). The keywords are written as comments in the appropriate section of the program to help identify the location of the code.

This SerDes logic circuit enables switching between the transmission and reception modes via a switch. The transmission timing is controlled using a counter circuit implemented on the FPGA. Since the Raspberry Pi operates only as an SPI master, it communicates with the FPGA by requesting data using the data_req signal connected to the “data_req” register defined in the DE0NanoSoc_TXRX as the top-level module in Appendix (i), thereby prompting SPI communication. The data received from the Raspberry Pi via SPI are transmitted to the serializer module. In the serializer module, a packet is generated with 8B13B conversion, the cyclic redundancy check (CRC) code is added, and transmission

begins when the "start" register defined in the top-level module is set to "1" in the timing controller. The CRC module in this system begins operation when the port name, "start," is set to 1, and the port name, "strobe," is set to 1 following the completion of the operation (CRC module in Appendix (i)). The "in" and "out" signals connected to the CRC, which are depicted in Figure 5, represent the port names, "data_in" and "result," respectively, of the CRC module. The serialized signal is then transmitted to the LED driver, where it is output as an optical signal.

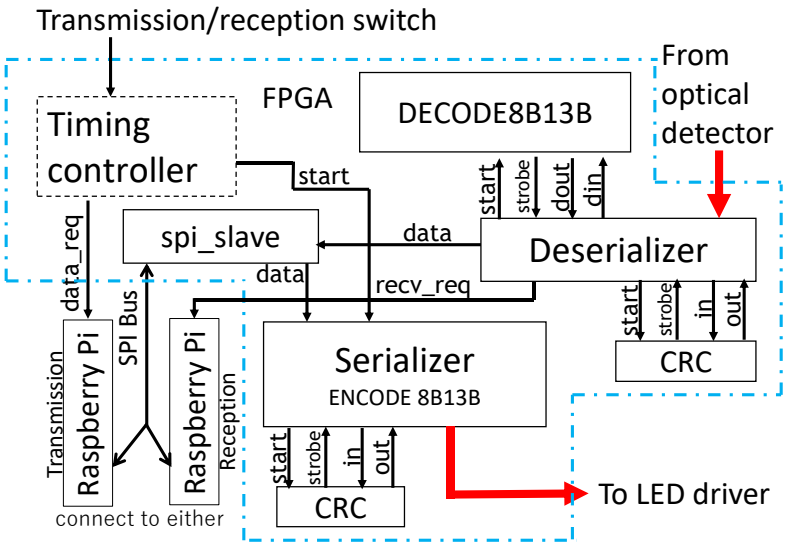

Figure 5. Block diagram of SerDes logic circuit implemented on FPGA.

The signal converted from light to electricity is transmitted to the deserializer module. In the deserializer module, a logic that operates similarly to a monostable multivibrator is used to shape pulses with a pulse width of 130 ns, triggered by the positive edge of the input signal (see also monostable multivibrator in Appendix (i)). The signal is clocked using the delay-locked loop implemented in the deserializer module, and the bit sequence is then read. The bit string is decoded by the DECODE8B13B module following error detection by the CRC module. The DECODE8B13B module in this system starts operation when the port named "start" is set to "1" and "strobe" is set to "1" following the completion of the operation.

The "din," "dout," "start," and "strobe" signals connected to DECODE8B13B (Figure 5) are the port names of "data448_in," "data_out," "start," and "strobe," respectively, of the DECODE8B13B module (DECODE8B13B module in Appendix (i)). Following the successful decryption, the "recv_req" signal, which is depicted in Figure 5 and defined by the same name in the top-level module, is generated, and the information is transmitted from the Raspberry Pi side via SPI communication.

The timing of the SPI communication between the FPGA and Raspberry Pi is crucial. In this study, when the program running on the Raspberry Pi used interrupts, it was necessary to include a communication interval of approximately 25 ms. When using the polling method, the SPI communication interval needed to be approximately 7 ms.

The timing is controlled using the "counter" register defined in the top-level module. This counter increments every 640 ns, counts to 38600, and returns to zero. When the counter reaches "1," it prompts the data request signal to be transmitted to the Raspberry Pi. If the SPI communication is successful, the

data is set to “1” in the “active_r” register defined in the top-level module, indicating that the data is being held. If the SPI communication fails, the “active_r” register is set to “0.”

When transmitting packets from the serializer to the LED driver, if the “active_r” register is set to 1, the start bits are set to 1111 in the packet described below. If it is set to 0, the start bits are set to 0000. When the start bit is 0000 on the receiving side, the data in the packet will be discarded. This prevents flickering by keeping the LED lit even when no information is being transmitted.

When implementing the proposed VLC system on other FPGAs or single-board computers, it is crucial to set the timing of the “start” register defined in the top-level module to 1. In Appendix (i), this is set to 11000, which is approximately 7 ms. A polling method can be used on the Raspberry Pi to reduce this to 5.5 ms. This value can be further reduced by using a single-board computer with faster response times or a real-time operating system.

2.4. 8B13B code

Table 1 lists some of the 8B13B codes, with the complete table provided in Appendix (iii) as the Verilog HDL code (8B13B.h).

Table 1. 8B13B code for visible light communication.

8B13B code		
Value	Transmission	Reception
0	**0**0010101**1**0101(6)	**0**0010101**0**0101(5)
1	**0**0010101**1**011**0**(6)	**0**0010101**0**0100(4)
2	**0**0010101**11**001(6)	**0**0010101**0**0001(4)
3	**0**0010101**11**010(6)	**0**0010101**0**0010(4)
4	**0**0010101**111**00(6)	**0**0010101**0**0000(3)
5	**0**00101**1**010101(6)	**0**0010100010101(5)
.	.	.

Parenthesis in number of bits 1.

The newly proposed 8B13B code uses the return-to-zero (RZ) method. For instance, consider the number, “3”. The first bit is converted into the 13-bit sequence, 000101010111010, as shown in Table 1. The first bit is always set to 0 to ensure the RZ format. Set the number of “1” bits to 6 to minimize fluctuations in light intensity during transmission and maintain DC balance. Upon reception, only the positive edges are considered as 1. Therefore, in this example, the received code is converted to 000101010100010. Using positive edges helps in reducing the influence of fluorescence decay on the falling edges of the white LEDs.

Although the 8B13B code requires additional bits, it is expected to be more resilient to waveform distortion (e.g. DDPWS) since the RZ format is preserved. A similar code, 4B7B, can achieve the same functionality; however, 8B13B exhibits greater efficiency. The coding rates for 4B7B and 8B13B are 0.571 and 0.615, respectively.

The transmitted bit sequence, which is encoded using 8B13B, is assigned a CRC code (CRC-16-CCITT) to the packet after the calculation. This CRC code is used as an error-correction code during reception. The error-correction code section employs a modified Manchester code that encodes 1 as 010 and 0 as 001 (see also 3-bit Manchester in Appendix (i)).

The optical signal transmitted from the LED driver is converted into an electrical signal by the receiver and then binarized. The consecutive "1" of two or more bits are converted using a monostable multivibrator implemented in the FPGA, as shown on the right side of Table 1 (see also monostable multivibrator in Appendix (i)).

2.5. Delay locked loop on FPGA

Figure 6 depicts the packet format. We employed a delay-locked loop (DLL) in this process, which is described later in the proposed VLC system. The preamble synchronizes and sets the DC bias to an appropriate level. The start bit comprises a continuous 4-bit sequence of "1." The CRC section is 48 bits, which is encoded using 1B3B conversion (see also serializer in Appendix (i)).

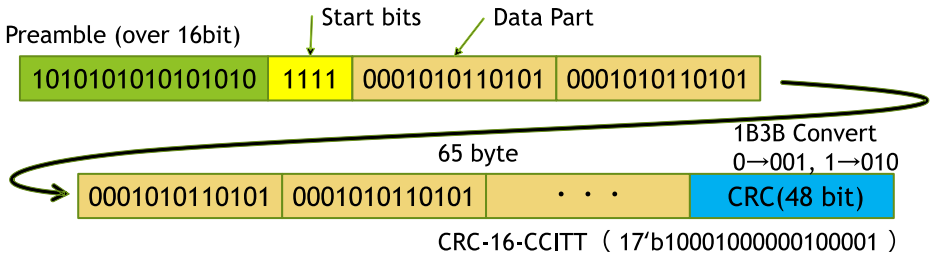

Figure 6. Packet format of VLC using 8B13B encoding.

If the preamble is 36 bits transmitted at a rate of 160 ns/bit, the duration would be 149.28 µs/packet. The communication speed is 65 bytes × 8 / 148.28 µs, which is approximately 3.48 Mbit/sec. However, the communication frequency between the Raspberry Pi and FPGA reduces the communication speed.

Figure 7a depicts the synchronization method. The input signal from the photodetector is fed into a 32-bit shift register after being passed through a waveform-shaping circuit, which functions as a monostable multivibrator implemented in the FPGA. Edge detection is performed by extracting the bit pattern stored within the shift register using a case statement (DelaySelector32 in Appendix (i)). In the figure, bits 31 through 22 are set to "0," and bits 21 through 7 are consecutively set to "1."

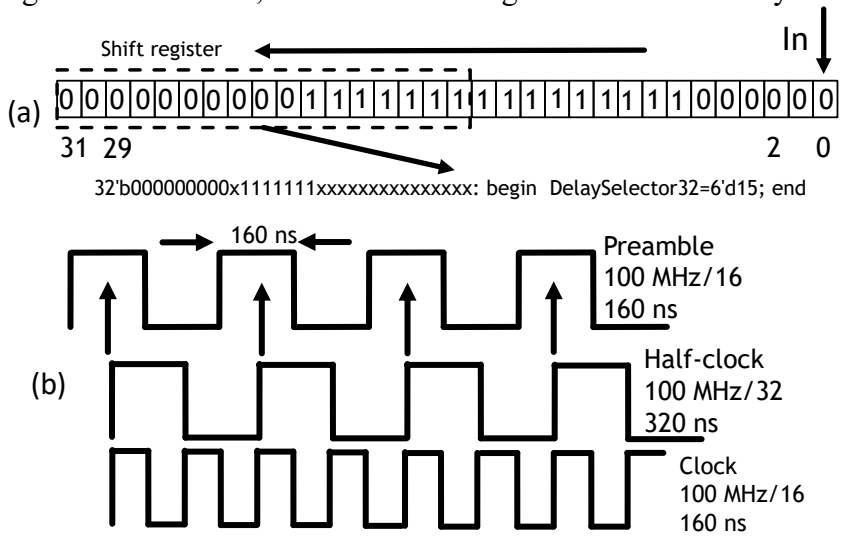

Figure 7. Delay locked loop fabricated on FPGA.

Figure 7b depicts the synchronization method included in the preamble section. Since “1” and “0” are repeated alternately in the preamble section, synchronization with the input signal is performed using a clock with a period that is twice that of the clock to be played. In this case, the correct position is determined using the bit pattern stored in the shift register, as explained earlier.

For the data section, the phase detected at each positive edge of the serial data was used to adjust the reading position of the next bit.

Figure 8 depicts the phase difference detection between the serial data and the recovered clock, along with its correction. The difference between the serial data and recovered clock is detected by the bit pattern of the received signal in the shift register. The transmitter and receiver clocks are not necessarily in perfect alignment. When the bit pattern, as described in the case statement, indicates the edge of the shift register, the phase is adjusted by increasing or decreasing the counter value of the 1/16th division circuit at 100 MHz (DelaySelector32 and Clock Shift in Appendix (i)). Communication is possible despite a frequency difference of 2.3 kHz between the 50 MHz clocks of the two FPGAs used in this study. We intentionally changed only the 50 MHz system clock of the transmitting FPGA to verify that synchronization is maintained without problems as long as the frequency difference from the receiving FPGA’s system clock is up to ± 50 kHz.

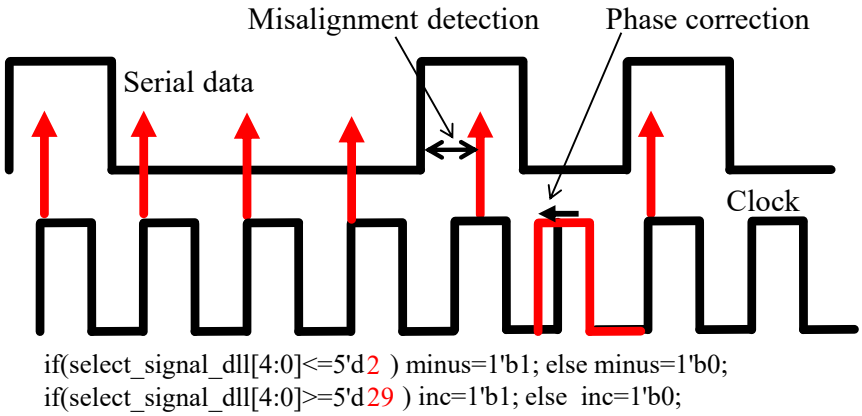

Figure 8. Misalignment detection and phase correction.

3. Results

Firstly, we evaluate the noise level of the optical detector. Figure 9a depicts a histogram of noise intensity during 70 μs for optical detector, in the dark room (5 lx) and under a fluorescent lamp (980 lx). Measurements were taken 70000 times at intervals of 1 ns per measurement. The root mean square (RMS) of the noise in the dark room and under fluorescent lamp were 0.034 V and 0.041 V, respectively, which we estimated to be the N_{RMS} of this photodetector. When compared with the darkroom, the noise increases only slightly, by approximately 18%, under fluorescent lighting. This is attributed to the fact that the bandpass filter, which transmits blue light, is tuned to the excitation wavelength of the white LED and effectively blocks other wavelengths.

Figure 9b depicts the signal to noise ratio *versus* distance under a 980 lx fluorescent lighting environment. The S/N of each distance in Figure 9b were estimated using the peak-to-peak of the first bit after the preamble and N_{RMS}. Typically, the SNR for stable communication is assumed to be 25 to 30 dB or more. Therefore, the communication distance that can be expected with the current combination of LED driver and receiver is approximately 5 m.

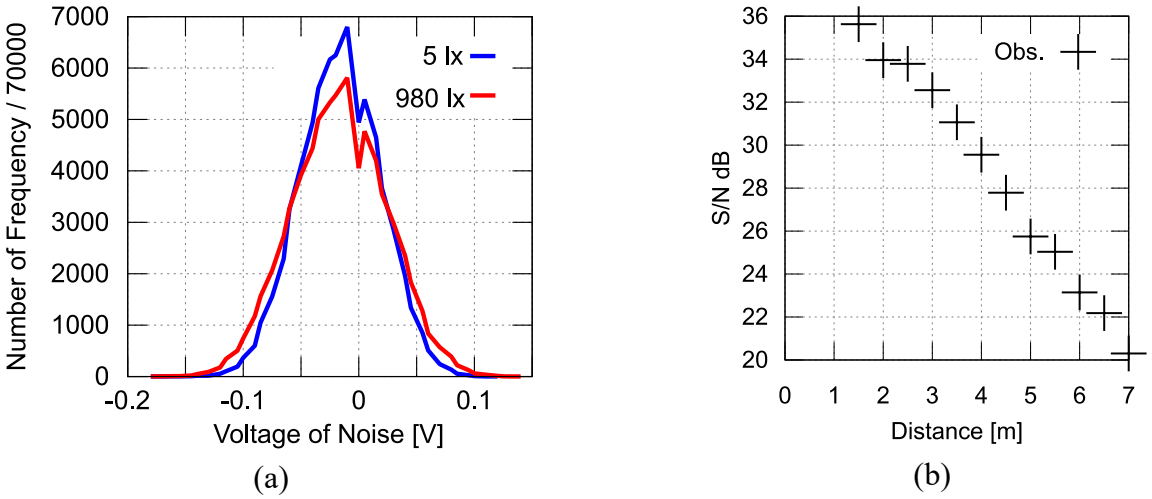

Figure 9. Histogram of noise intensity for our optical detector and signal-to-noise ratio versus distance under 980 lx fluorescent lighting environment.

Figure 10 depicts the evaluation conducted under sunlight considering the outdoor applications of this system. The evaluation focused on the distance between the LED driver and receiver, intensity of sunlight as a disturbance light, and the corresponding error rate. The experimental setup is identical to that depicted in Figure 2. The intensity of the ambient light irradiated in front of the receiver was measured using an ANA-F11 illuminance meter (Tokyo Garasu Kikai Co., Ltd.).

Figure 10 depicts the output signal from the photodetector receiving light from the LED driver, along with the signal after passing through the binarization circuit. Figure 10d depicts the experimental environment in this evaluation. The upper part in Figure 10a,b depicts the output of the photodetector, which is an analog signal, whereas the lower part shows the digital signal after conversion using a binarization circuit. The RMS of noise was 0.034–0.041 V; however, the threshold of the binarization circuit was set to 0.54 V, considering external electromagnetic waves. The distance between the receiver and LED was set to 3 m and 4 m, respectively, and the background sunlight was 24,600 lx. When the distance between the receiver and the LED driver was 3 m, the signal-to-noise ratio was 31 dB in Figure 10a. The light-receiving element did not saturate even under strong sunlight owing to the blue bandpass filter on the photodetector. The signal-to-noise ratio decreased to 27 dB at a distance of 4 m, indicating that the bits of the signal below the threshold were being lost, as shown in Figure 10b. This combination of the LED driver and receiver limited the communication range to approximately 3.5 m. In this experiment, the transmission line between the analog signal and the binarization circuit was a coaxial cable; however, it was expected that the communication distance could be extended slightly by adjusting the transmission line and threshold value.

Figure 10c shows the pulse signal of “0001010101101010 1”, meaning 0, received from the LED driver at a distance of 350 cm from the receiver. The SNR at this distance was around 31 dB, and there were almost no communication errors. The pulse width of “1” is reduced to 67 ns at the threshold, while the pulse width of “0” is expanded to 240 ns in some areas. “11” part is 240 ns, showing how the pulse width varies depending on the bit pattern. Since our newly proposed 8B13B code focuses only on the rising edge of the bit, it is tolerant of pulse width variations such as DDPWS.

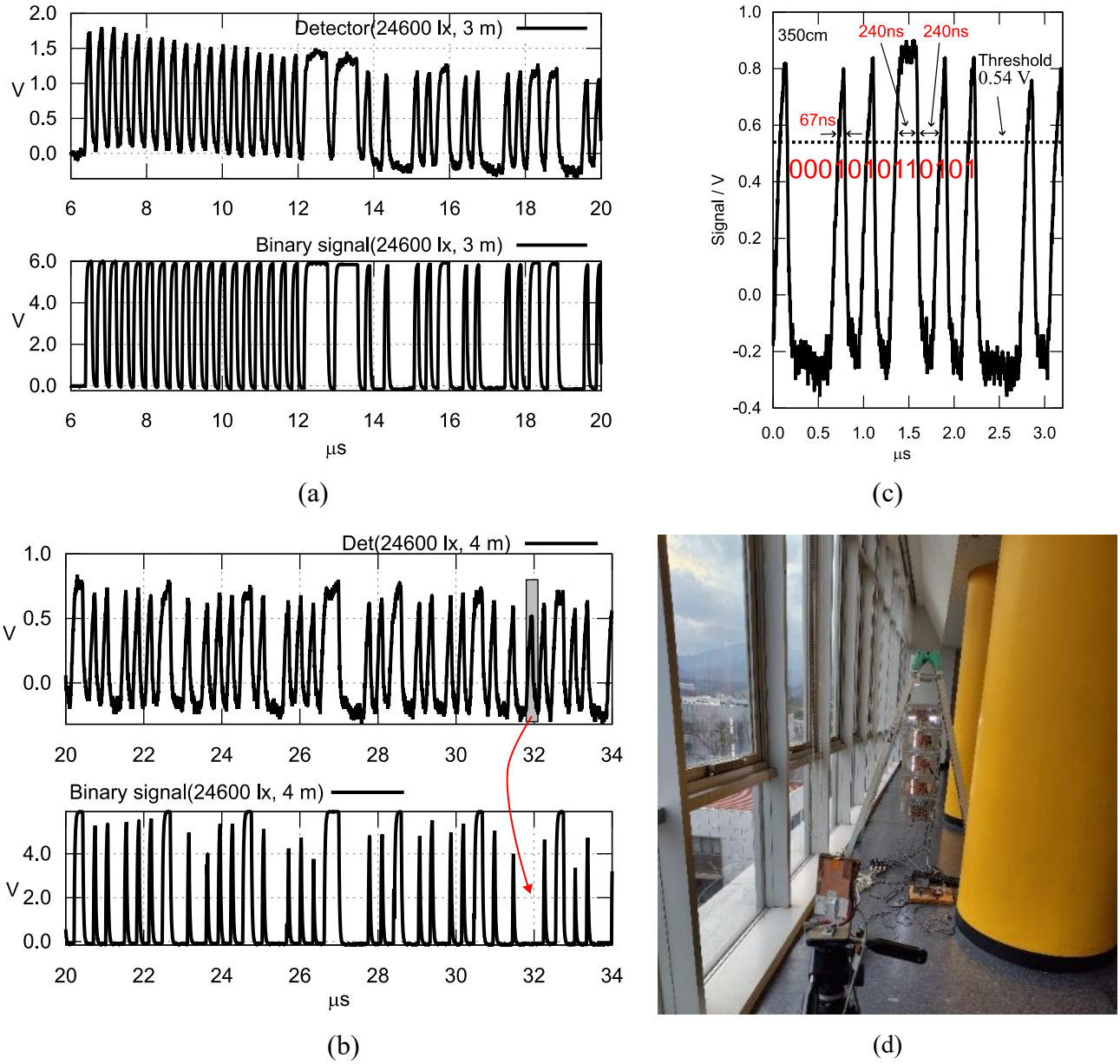

Figure 10. Output signals of receiver and binarization circuit and measurement environment. **(a)** 3 m distance between LED driver and detector; **(b)** 4 m distance between LED driver and receiver; **(c)** Pulse width of the analog signal; **(d)** Experimental Environment.

Figure 11 depicts the relationship between the distance and packet loss. The horizontal axis represents the distance between the LED light source and the photodetector. The optical axis of the photodetector shifts when the distance between the photodetector and the LED driver lies between 100 and 180 cm owing to the intense illumination from the LED light source, as shown in Figure 11a,b.

Figure 11a depicts the illuminance of the LED light source *versus* distance, along with the signal voltage received by the photodetector.

Figure 11b depicts the number of packet losses, as well as the numbers of successful and failed error corrections, plotted against the distance for 20,001 packet transmissions. In this experiment, we included the illuminance of the sunlight in front of the receiver ranged from 91 to 777 lx and from 518 to 1080 lx when light from the fluorescent lamp.

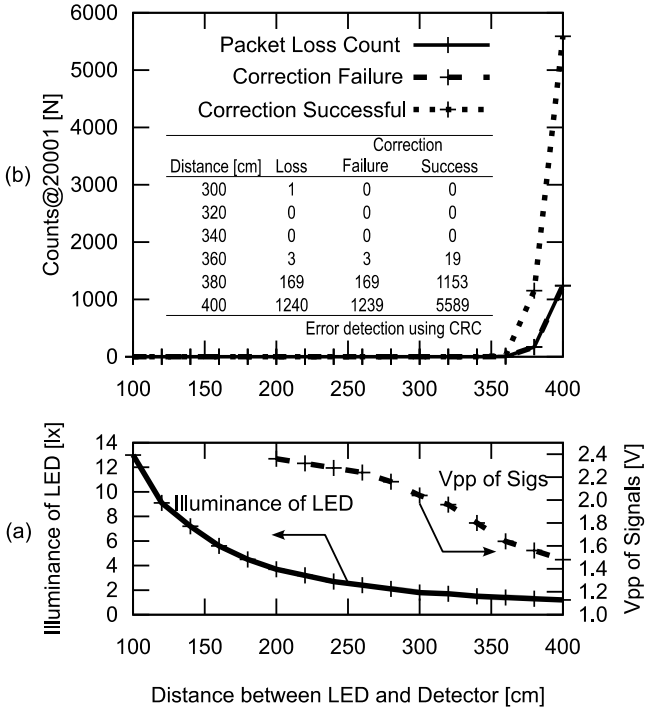

Figure 11. Relationship between the distance and packet loss.

The proposed deserializer achieves error detection and correction capabilities by using the CRC module. An external switch is operated to record the numbers of detected errors and corrected packets at the end of the received packet (see also Amount Error in Appendix (i)). Each packet is assigned a serial number, enabling the detection of synchronization failure based on its continuity. If synchronization is successful but an error occurs, it is counted as an error in the register (num_error) defined in the deserializer module. If the error is successfully corrected, the register (num_corr) of this module is incremented separately. This enables the detection of synchronization failures, error occurrences, and error corrections.

One packet was lost at 300 cm, indicating a failure of synchronization, as neither successful nor failed error correction was recorded. The SNR at 340 cm was approximately 31 dB, no errors were detected, and the bit error rate was estimated to be less than 5.6×10^{-8} since each packet contained 893 bits. The error corrections were performed at distances of 360 cm and above; however, the failure rate also increased rapidly. The close correlation between the error correction failures and packet loss values suggests that synchronization was successful. However, the increased distance likely caused multibit errors that could not be corrected. In the experiment, the analog signal is digitized by a binarization circuit with a threshold voltage of 0.54 V. When the voltage drops below this threshold voltage, the bits disappear as shown in Figure 10b. The peak-to-peak Vpp of the serial signal is approximately 1.6 V, as shown in Figure 11a. However, the bit pattern does not provide sufficient amplitude, and the bits were assumed to disappear, as shown in Figure 10b.

We tested this system under strong sunlight environment (94,000 lx). Figure 12 presents a photograph of the experimental environment. The distance between the LED light source and receiver was 320 cm. When 100,001 packets were transmitted to the receiver, representing 89.3 Mbit, only one packet was lost. This transmission was concluded to be a synchronization failure since no CRC errors were observed. Therefore, the packet loss rate owing to the synchronization failure was estimated to be approximately 10^{-4}–10^{-5} at a communication distance of 3 m in this system.

Figure 12. Photograph of the experimental environment.

4. Conclusion

In this study, we developed the SerDes logic and its implementation on an FPGA to achieve VLC. A simple transmitter/receiver system was realized by connecting the FPGA and a Raspberry Pi via SPI communication. The newly proposed 8B13B code, which helps in suppressing the flickering during transmission by balancing the number of "1 s" and "0 s", was devised for VLC. The RZ code can be easily synchronized and adjusted, even when the pulse width varies.

The fabricated LED driver utilizes commercially available LEDs and incorporates a signal distribution circuit that facilitates easy expansion of the number of LEDs used. Additionally, the signal distribution and LED switching circuits were connected via a twisted pair cable with a length of 2 m, thereby ensuring flexibility during installation.

Three PIN photodiodes were used in the photodetector to enhance the light-receiving sensitivity while avoiding issues with the response time. Additionally, an optical narrow-bandwidth bandpass filter was employed to ensure stable transmission and reception, even under sunlight. The transmission time was 160 ns/bit. The SNR at 340 cm was approximately 31 dB, no errors were detected, and the bit error rate was estimated to be less than 5.6×10^{-8}.

We believe that there is still scope for improvement in both the LED driver and receiver, and that the communication distance can be further extended.

The details on the SerDes logic written in Verilog HDL are provided as supplementary data. This code can be easily ported to other FPGAs since it was created with minimal use of IP cores. These codes were then released as open source [33].

A low-cost VLC system can be established by using the proposed method. Leveraging Raspberry Pi enables the development of VLC systems that incorporate image recognition or AR technologies. Although SPI communication between an FPGA and a Raspberry Pi is relatively simple to implement, it imposes limitations on the communication interval, resulting in an overall data transfer rate that is not particularly high. Increasing the packet size in SPI communication is expected to enhance the communication speed by up to 1 Mbit/sec.

The VLC system proposed in this study encompasses a wide range of technologies, including analog and digital circuits, along with software. The Verilog HDL code for the SerDes logic is publicly available, and the system is highly reproducible owing to the use of readily obtainable components. We believe the proposed VLC system is well-suited as an educational tool for communication systems, as it requires a broad range of electronic engineering knowledge to design and develop electronic circuits and software.

Supplementary data

The supplementary data associated with this article include Appendix (i) DE0NanoSoc_TXRX.v, (ii) spi_slave.v, and (iii) 8B13B.h, which are the source codes of SerDes logic written in Verilog HDL. The complete source code, including Verilog HDL for the FPGA and the Raspberry Pi software, can be downloaded from github.com (https://github.com/atlasmaker/8B13B_65B). On this site, you can find the FPGA/DE0NanoSoc_TXRX.qpf file and build it using Quartus 23.1.

Acknowledgments

This work was supported by the regular institutional budget of Tokyo Polytechnic University.

Authors' contribution

Conceptualization, Tokio Yukiya; methodology, Tokio Yukiya; software, Tokio Yukiya; validation, Tokio Yukiya; formal analysis, Tokio Yukiya; investigation, Tokio Yukiya; resources, Tokio Yukiya; data curation, Tokio Yukiya; writing—original draft preparation, Tokio Yukiya; writing—review and editing, Nobuo Nishimiya and Takayuki Uchida; visualization, Tokio Yukiya; supervision, Tokio Yukiya; project administration, Tokio Yukiya; funding acquisition, Tokio Yukiya. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

The authors declare no conflict of interest.

References

[1] Rehman SU, Ullah S, Chong PHJ, Yongchareon S, Komosny D. Visible light communication: a system perspective—overview and challenges. *Sensors* 2019, 19(5):1153.

[2] Patle N, Raj A, Joseph C, Sharma N. Review of fibreless optical communication technology: history, evolution, and emerging trends. *J. Opt. Commun*. 2024, 45(3):679–702.

[3] Gupta S, Roy D, Bose S, Dixit V, Kumar A. Illuminating the future: a comprehensive review of visible light communication applications. *Opt. Laser Technol.* 2024, 177:111182.

[4] Liang C, Li J, Liu S, Yang F, Dong Y, *et al*. Integrated sensing, lighting, and communication based on visible light communication: a review. *Digital Signal Process.* 2024, 145:104340.

[5] Shaaban K, Shamim MHM, Abdur-Rouf K. Visible light communication for intelligent transportation systems: a review of the latest technologies. *J. Traffic Transp. Eng.* 2021, 8(4):483–492.

[6] Memedi A, Dressler F. Vehicular visible light communications: a survey. *IEEE Commun. Surv. Tutor.* 2021, 23(1):161–181.

[7] Pang G, Kwan T, Chan CH, Liu H. LED traffic light as a communications device. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No. 99TH8383)*, Tokyo, Japan, October 5–8, 1999, pp. 788–793.

[8] Akanegawa M, Tanaka Y, Nakagawa M. Basic study on traffic information system using LED traffic lights. *IEEE Trans. Intell. Transp. Syst.* 2001, 2(4):197–203.

[9] Vieira M, Vieira MA, Galvão G, Louro P, Véstias M, *et al*. Enhancing urban intersection efficiency: utilizing visible light communication and learning-driven control for improved traffic signal performance. *Vehicles* 2024, 6(2):666–692.

[10] Vieira MA, Galvão G, Vieira M, Louro P, Vestias M, *et al*. Enhancing urban intersection efficiency: visible light communication and learning-based control for traffic signal optimization and vehicle management. *Symmetry* 2024, 16(2):240.

[11] Vieira MA, Vieira M, Louro P, Vieira P, Fantoni A. Vehicular visible light communication for intersection management. *Signals* 2023, 4(2):457–477.

[12] El-Garhy SM, Khalaf AAM, Abaza M, Aly MH. Intelligent transportation system using wireless optical communication: a comparative study. *Opt. Quantum Electron.* 2023, 56(2):247.

[13] Seminara M, Nawaz T, Caputo S, Mucchi L, Catani J. Characterization of field of view in visible light communication systems for intelligent transportation systems. *IEEE Photonics J.* 2020, 12(4):1–16.

[14] Goto Y, Takai I, Yamazato T, Okada H, Fujii T, *et al*. A new automotive VLC system using optical communication image sensor. *IEEE Photonics J.* 2016, 8(3):1–17.

[15] Haas H, Yin L, Wang Y, Chen C. What is LiFi? *J. Lightwave Technol*. 2016, 34(6):1533–1544.

[16] Purwita AA, Haas H. Studies of flatness of LiFi channel for IEEE 802.11bb. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Republic of Korea, May 25–28, 2020, pp. 1–6.

[17] IEEE Standards Association (IEEE-SA). IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Light Communications (IEEE Std 802.11bb-2023). 2023, pp. 1–37. Available: https://doi.org/10.1109/IEEESTD.2024.10315104 (accessed on 6 November 2025).

[18] Ge P, Wang J, Ling X, Liang X, Tian Y, *et al*. Achievable rate analysis for post- and pre-equalization in DCO-OFDM VLC with limited dynamic range. *Opt. Commun*. 2020, 476:126277.

[19] Alrakah HT, Gutema TZ, Sinanovic S, Popoola WO. PAPR reduction in DCO-OFDM based WDM VLC. *J. Lightwave Technol*. 2022, 40(19):6359–6365.

[20] Mapfumo I, Shongwe T. PAPR reduction using selective mapping of an ACO-OFDM hybrid PLC/VLC system for IoT applications. *IEEE Access* 2024, 12:190809–190820.

[21] Rajagopal S, Roberts RD, Lim SK. IEEE 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Commun. Mag*. 2012, 50(3):72–82.

[22] Mietzner J, Lampe L. Nested PPM for visible light communication with heterogeneous optical receivers. *IEEE Photonics J*. 2024, 16(4):1–12.

[23] Nawaz T, Seminara M, Caputo S, Mucchi L, Cataliotti FS, *et al*. IEEE 802.15.7-compliant ultra-low latency relaying VLC system for safety-critical ITS. *IEEE Trans. Veh. Technol*. 2019, 68(12):12040–12051.

[24] Ricci S, Caputo S, Mucchi L. FPGA-based visible light communications instrument for implementation and testing of ultralow latency applications. *IEEE Trans. Instrum. Meas*. 2023, 72:1–11.

[25] Ricci S, Caputo S, Mucchi L. FPGA-based Manchester decoder for IEEE 802.15.7 visible light communications. *Electronics* 2025, 14(1):96.

[26] Caputo S, Ricci S, Mucchi L. IEEE 802.15.7-compliant full duplex visible light communication: Interference analysis and experimentation. *IEEE Open J. Veh. Technol.* 2024, 5:1242–1255.

[27] Mambou EN, Tonnellier T, Gross WJ. Improved DC-free run-length limited 4B6B codes for concatenated schemes. *IEEE Access* 2022, 10:21847–21852.

[28] Dhannoon Z, Fischer RFH. Performance of run-length-limited codes in visible-light communications. In *SCC 2019; 12th International ITG Conference on Systems, Communications and Coding*, Rostock, Germany, February 11–14, 2019, pp. 1–6.

[29] Choudhury PK. Enhanced tolerance against optical background noise in VLC link by using line coded signal with receiver filter. *Results Opt*. 2020, 1:100020.

[30] Chen W, Ren T, Yuan T, Han D, Yang C, *et al*. Semantic-empowered visible light communication for image transmission based on deep convolutional generative adversarial network. *Opt. Express* 2024, 32(19):32564–32584.

[31] Yan D, Mao X, Xie S, Cong J, Chen H. Design fully integrated driver circuit for phosphorescent white light-emitting-diode high-speed real-time wireless communication. *IEEE Photonics J.* 2019, 11(2):1–10.

[32] Li X, Ghassemlooy Z, Zvanovec S, Alves LN. An equivalent circuit model of a commercial LED with an ESD protection component for VLC. *IEEE Photonics Technol. Lett.* 2021, 33(15):777–779.

[33] McCauley M. Available: https://www.airspayce.com/mikem/bcm2835/ (accessed on 2 June 2025).