Journal of Applied Science, Engineering, Technology and Management
Vol. 03, Issue 01, June 2025 pp. 33-36

Li-Fi Based Image Transmission

Parthiv A
Department of Electronics and Instrumentation
Federal Institute of Science and Technology
Ernakulam, Kerala, India
parthiv03804@gmail.com

Vishnu V B
Department of Electronics and Instrumentation
Federal Institute of Science and Technology
Ernakulam, Kerala, India
vishnubabuindot@gmail.com

Shahila T I
Department of Electronics and Instrumentation
Federal Institute of Science and Technology
Ernakulam, Kerala, India
shahilatharumannil@gmail.com

Loyd John J
Department of Electronics and Instrumentation
Federal Institute of Science and Technology
Ernakulam, Kerala, India
loydjohnj16@gmail.com

Anil Johny
Department of Electronics and Instrumentation
Federal Institute of Science and Technology
Ernakulam, Kerala, India
aniljohny@fisat.ac.in

***Abstract*— Communication is an essential part of modern life, powering everything from personal devices to industrial systems. Traditional wireless communication technologies primarily rely on radio frequency (RF) transmission, which faces challenges such as bandwidth limitations, interference, and restrictions in sensitive environments like hospitals and aircraft. This paper presents a prototype system for image transmission using Light Fidelity (Li-Fi), a wireless communication technology that utilizes visible light. We propose a method in which grayscale images are converted into binary data via Python, and transmitted using LED pulses controlled by an Arduino. The receiving end employs a photodiode and a comparator to detect signals, which are then decoded and reconstructed by a Raspberry Pi. The system transmits images at a fixed bit rate of 4 bps using a simple protocol with periodic start sequences. Our results demonstrate successful image reconstruction but also highlight challenges such as synchronization drift and limited throughput. This work showcases the feasibility of Li-Fi-based image communication in low-speed, RF-sensitive applications and provides a basis for future enhancement.**

Keywords - Li-Fi, Image Transmission, Arduino, Raspberry Pi, Visible Light Communication, Wireless Communication

I. INTRODUCTION

Li-Fi (Light Fidelity) is an emerging wireless communication technology that uses visible light for data transmission. Compared to conventional RF-based systems like Wi-Fi or Bluetooth, Li-Fi offers benefits including higher available bandwidth, reduced electromagnetic interference, and enhanced security. This work explores Li-Fi for image transmission, proposing a novel approach using simple embedded systems. Applications include RF-restricted environments such as hospitals, aircraft, and defense facilities.

The Li-Fi image transmission system is designed with simplicity in mind, incorporating easily accessible and compatible components. The transmitter consists of an Arduino Uno, an IRF7400 MOSFET, and a white LED, while the receiver setup uses a BPW34 photodiode and an LM339 comparator interfaced with a Raspberry Pi. The hardware connections are intuitive and minimize the risk of assembly errors. On the software side, Python scripts automate the entire data pipeline—converting the image into binary data, transmitting it via light pulses, and reconstructing the image at the receiving end—thereby reducing the need for user intervention. Additionally, the blinking LED serves as a visual indicator of ongoing data transmission, offering a simple yet effective way to monitor system activity.

Despite its accessible design, the system presents certain challenges that affect overall ease of use. Accurate synchronization between the transmitter and receiver is essential, necessitating meticulous calibration. Even slight misalignment or inconsistent light intensity can disrupt communication, making setup adjustments both sensitive and time-consuming. Furthermore, the current implementation lacks error detection or correction mechanisms. Any data loss or misinterpretation during transmission must be manually identified and resolved, which limits the system's robustness and user-friendliness in practical scenarios.

II. LITERATURE SURVEY

Recent research has shown a growing interest in using Li-Fi technology for secure and efficient wireless data transmission, particularly in image communication. Kumar and Lee proposed an optical wireless communication system using a white LED controlled by an Arduino and a laptop running Python, with a BPW34 photodiode and an LM339 comparator as the receiver, interfaced with a Raspberry Pi [1]. They highlighted the need for periodic re-synchronization due to the asynchronous nature of the devices. Building on this, Rodriguez et al. explored high-speed optical data transmission using MOSFET-driven LEDs and photodiodes, emphasizing the importance of synchronization techniques to reduce bit drift [2].

eISSN 2584-0371

Received for review: 30-05-2025
Accepted for publication: 01-07-2025
Published: 11-07-2025

In another prototype, successful wireless image transmission via Li-Fi was demonstrated by converting images into binary data on a laptop and transmitting them through an Arduino-controlled LED, with reception and reconstruction handled by a Raspberry Pi [3]. To improve synchronization, Smith studied fixed-delay and periodic re-synchronization methods, which were essential for maintaining data alignment [4]. Gupta and Zhao focused on optimizing the hardware interface between Arduino and Raspberry Pi, while also addressing software-level synchronization challenges to ensure robust data delivery [5].

Further enhancements were proposed through wireless clock synchronization using RF modules and optical sync pulses [6]. Error detection mechanisms and periodic start sequences were also introduced to improve data integrity in noisy environments [7]. A broader review highlighted various optical wireless communication systems, showcasing the use of low-cost platforms like Arduino and Raspberry Pi in IoT applications, and emphasizing their potential in low-power, high-security environments such as hospitals and aircraft [8].

III. Methodology

This section details the design and implementation of the Li-Fi-based image transmission system. The methodology encompasses the components used, transmitter and receiver configurations, image conversion techniques, and synchronization strategies employed during data transfer.

A. FE. Required Components

The system utilizes the following hardware and software components:

- Arduino Uno (ATmega328P)
- N7000 N-Channel MOSFET
- White LED
- BPW34 Photodiode
- LM339 Comparator
- Voltage Divider Circuit
- Raspberry Pi 3 Model B+
- Python 3.11 with OpenCV, NumPy, and GPIO libraries

B. Transmitter Circuit

The transmitter setup includes a Python-based image processing script running on a laptop, which converts a grayscale image into a binary stream. This stream is transmitted via USB serial communication to the Arduino Uno. The Arduino controls an N7000 MOSFET, which in turn drives a high-brightness white LED, which is shown in Fig. 1.

- Binary '1': LED is ON
- Binary '0': LED is OFF

The transmitter sends a start control byte 'S' followed by an 8-bit start sequence (10101010) every 50 bits to assist with synchronization. An end byte 'E' marks the transmission conclusion.

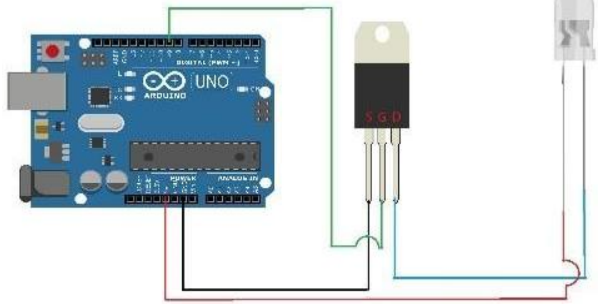

Fig. 1 Transmitter circuit, including the Arduino, MOSFET driver, and LED configuration.

C. Receiver Circuit

At the receiver, a BPW34 photodiode converts the LED light pulses into voltage signals, which is shown at Figure 2. These are fed to an LM339 comparator, which uses a voltage divider reference (≈0.16 V) to convert the analog signal into clean digital logic (0V or 3.1V). This digital signal is read by GPIO pin 17 of the Raspberry Pi.

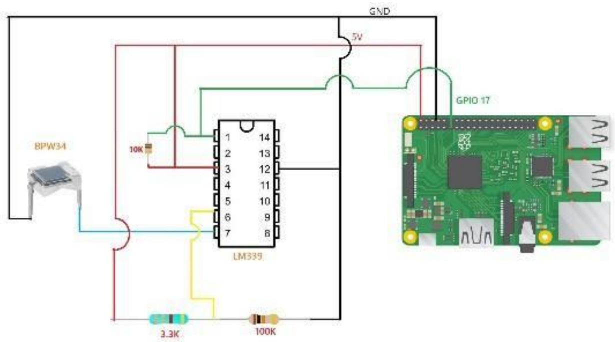

Fig. 2 Receiver circuit diagram including the photodiode, comparator, and GPIO interface

D. Flow Diagram of the System

The overall system flow for Li-Fi based image transmission is illustrated in Figure 3. It encompasses two main stages: the transmitter and the receiver.

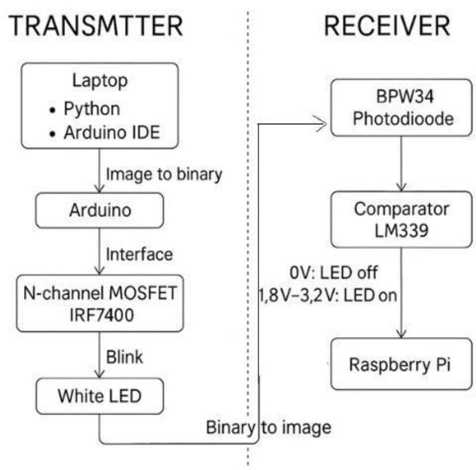

Fig. 3 Block Diagram of the Transmitter and Receiver setup for Li-Fi based image transmission.

On the transmitter side, a laptop running Python and the Arduino IDE initiates the process by converting a grayscale image into binary data using image thresholding techniques. This binary stream is then sent to an Arduino UNO, which interfaces with an N-channel MOSFET IRF7400. The MOSFET acts as a switch, driving a white LED to emit visible light pulses representing binary '1' (LED ON) and '0' (LED OFF).

Fig. 3 represents the Block Diagram of the Transmitter and Receiver setup for Li-Fi based image transmission. The system converts image data into binary using a Python script on a laptop and transmits it via a white LED controlled by an Arduino and IRF7400 MOSFET. On the receiver side, a BPW34 photodiode detects the light pulses, which are converted to digital signals by an LM339 comparator and processed by a Raspberry Pi for image reconstruction.

The overall system flow for Li-Fi based image transmission encompasses two main stages: the transmitter and the receiver. On the transmitter side, a laptop running Python and the Arduino IDE initiates the process by converting a grayscale image into binary data using image thresholding techniques. This binary stream is then sent to an Arduino UNO, which interfaces with an N-channel MOSFET IRF7400. The MOSFET acts as a switch, driving a white LED to emit visible light pulses representing binary '1' (LED ON) and '0' (LED OFF).

These modulated light pulses propagate to the receiver side, where a BPW34 photodiode detects the incoming light intensity. The photodiode outputs analog voltages, approximately 0 V for no light (LED OFF) and 1.8 V to 3.2 V for light detection (LED ON). This analog signal is fed into a Comparator (LM339), which transforms it into a clean digital signal based on a threshold voltage provided by a voltage divider circuit.

The digital output is read by the Raspberry Pi via GPIO pin 17. A Python script on the Pi detects the start sequence, captures the incoming binary stream, and reconstructs the image by reshaping the bits into a 2D pixel matrix. This process ensures a full binary-to-image conversion at the receiver end, mirroring the original image sent from the transmitter.

E. Image Reconstruction and Synchronization

On the Raspberry Pi, a Python script continuously monitors GPIO pin 17. The script waits for the 8-bit start sequence to begin capturing binary data. Each bit is sampled with a 0.25-second delay, matching the transmission rate of 4 bits per second. The reconstructed binary stream is reshaped into the original image matrix (e.g., 10×10 or 32×32) and saved as a grayscale image using OpenCV.

To minimize drift and alignment issues:

- A start sequence is re-inserted every 50 bits.
- The receiver checks for and re-aligns based on the detected start pattern.
- Basic checksum logic can be optionally integrated (not implemented in this version) to detect corrupted packets.

F. Packet Structure and Data Rate

The data is transmitted in the following format:
[S] [Start Sequence] [Data Bits...] [E]

- Start Byte: ASCII 'S'
- Start Sequence: 8-bit 10101010
- Image Data: Flattened binary matrix
- End Byte: ASCII 'E'

Transmission rate = 4 bits/sec
Typical packet size for a 10×10 image = 100 bits + framing bits
Total transmission time ≈ 25–30 seconds

IV. RESULTS

The Li-Fi image transmission system successfully demonstrates the end-to-end process of wireless optical data communication. The setup as shown in Figure 4 includes a well-coordinated arrangement of the transmitter and receiver circuits. The transmitter encodes a grayscale image into binary using a Python script and sends it through LED light pulses controlled by an Arduino. The receiver, composed of a BPW34 photodiode and LM339 comparator connected to a Raspberry Pi, decodes the optical signals back into binary and reconstructs the image.

Fig.4 Device Setup

The system effectively performs the intended functionality: image conversion, transmission, and reconstruction. The output image displayed on the receiver side is shown in Figure 5, which closely resembles the original image, validating the core functionality of the proposed system. Minor distortions observed in the output are attributed primarily to synchronization mismatches between the transmitter and receiver. These occur due to the absence of a shared clock, resulting in occasional misalignment of bits during data reception.

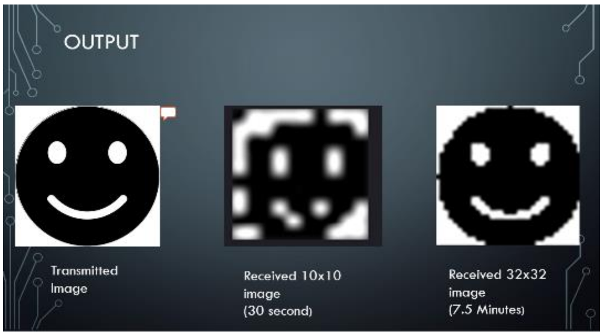

Fig 5. Output image displayed on the receiver

Furthermore, the current system operates at a modest data rate of 4 bits per second. While sufficient for proof-of-concept demonstrations, this low speed limits its practicality for high-resolution or real-time image transmission. Nonetheless, the reliable delivery of a complete image, despite such constraints, is a testament to the robustness and viability of the approach.

With enhancements in clock alignment techniques and bit rate optimization, the system is poised for significant performance improvements, making it a strong candidate for applications in environments where RF communication is restricted or undesirable.

V. Conclusion

The Li-Fi image transmission system showcases the exciting potential of visible light as a medium for wireless data communication. This novel implementation successfully achieves the conversion, transmission, and reconstruction of image data through optical means. While the current system encounters challenges such as synchronization mismatches between the transmitter and receiver, and a limited transmission speed of 4 bits per second, these are seen as opportunities for continued innovation. With the improvements underway, the system is poised to evolve into a more practical, robust, and high-performance solution for future wireless communication applications.

References

[1] S. Kumar and H. Lee, "Design and implementation of a Li-Fi data communication system," in Proc. IEEE Conf. on Wireless Optical Communications, 2018.

[2] L. Rodriguez, J. Kim, and M. Patel, "High-speed optical communication using LED and photodiode," Optoelectronics Letters, vol. 15, no. 2, pp. 110–115, 2019.

[3] X. Chen and N. Patel, "Wireless image transmission via Li-Fi: A prototype system, Int. Journal of Optical Networking, vol. 12, no. 1, pp. 34–40, 2020.

[4] D. Smith, "Adaptive synchronization techniques in Li-Fi networks," IEEE Trans. on Lightwave Tech., vol. 39, no. 5, pp. 1883–1891, 2021.

[5] R. Gupta and Y. Zhao, "Optical wireless communication with Arduino and Raspberry Pi," in Proc. Int. Conf. on Embedded Systems, pp. 58–63, 2021.

[6] B. Davis, L. Thomas, and K. Shah, "Robust clock synchronization in Li-Fi systems using wireless methods," J. Commun. Tech. and Electronics, vol. 67, no. 7, pp. 210–218, 2022.

[7] M. Thompson, "Enhancing data integrity in Li-Fi communication with error correction," Optical Comm. Research, vol. 23, no. 4, pp. 99–106, 2022.

[8] H. Nguyen and C. Alippi, "A review of optical wireless communication technologies in IoT applications," ACM Computing Surveys, vol. 52, no. 3, pp. 1–27, 2020.