International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Li-Fi Communication for Multimedia Transfer Under Aligned and Ambient Light Conditions

Anjana V[1], B G Dimple Gowda[2], Bindu A[3], Chaithanya V[4] Dr. K Somashekar[5]

Student, ECE, SJB Institute of Technology, Bengaluru, India[1-4]

Professor, ECE, SJB Institute of Technology, Bengaluru, India[5]

Abstract— The need for wireless data transmission has led to the growth of new communication technologies that do not rely on traditional radio-frequency links. Light-Fidelity or Li-Fi is one technology that uses light to send digital information. This paper presents a prototype that uses an ESP32 chip and Li-Fi technology to send both preset and personalized text messages as well as audio, images and video files through a light-based link.The transmitter employs an INMP441 I²S microphone. It records a 10-second mono audio clip at 16 kHz, then saves it as a WAV file on a microSD card. Text messages and file data travel through the optical link using on-off keying. Files are split into 128-byte blocks. A frame structure that includes synchronisation bytes and an XOR checksum helps the receiver recognise each packet. A CRC-32 check ensures that the files arrive intact.At the receiving side an LDR light sensor picks up the signal. An ESP32 chip then rebuilds the data that was sent. Wi-Fi and Firebase provide a dashboard for control, status updates and cloud storage. Meanwhile the optical link moves the text and file payloads from the transmitter to the receiver. This system offers a platform for people who want to explore cheap optical wireless communication and transfer of multimedia data.

Key Words—Li-Fi, light communication, optical wireless communication, ESP32 on-off keying, LDR, INMP441, Firebase, multimedia transmission.

I. INTRODUCTION

The growing demand for communication has made the necessity for reliable and efficient data transmission stronger. Traditional technologies like Wi-Fi and Bluetooth rely on radio-frequency signals. These signals can face problems like interference. Don't always offer enough bandwidth. Light-Fidelity or Li-Fi takes an approach. It sends information using modulated light. This method can be more stable, in environments and can provide faster data transfer when conditions allow. At the receiver a light sensor detects changes in intensity and turns them into electrical signals, which are then used to recover the data. This project presents a Li-Fi system that can send predefined and custom text messages and can also transfer audio, image and video files via an optical link. Transmitter uses an INMP441 I²S microphone to record audio and store the recordings and media files on a microSD card. Receiver employs a sensor and an ESP32 to detect and decode the transmitted data while OLED displays show the system status. A web-based dashboard that works with Wi-Fi and Firebase allows users to select commands monitor transmissions and access received media. The optical link carries the data, between the ESP32 boards while Wi-Fi and Firebase provide dashboard control and cloud operations.

II. RELATED WORK

Haas et al. [1] Define Li-Fi as a bidirectional, extension of visible light communication (VLC). Pathak et al. [2] Survey the MAC layers of VLC pointing out ambient light line-of-sight blockage and the absence of an uplink as the main practical challenges. Karunatilaka et al. [3] Show that phosphor LEDs limit bandwidth unless equalisation is applied. Nakagawa [6] model the indoor line-of-sight channel and ambient shot noise. Elgala et al. [7] Argue that intensity modulation with detection (IM/DD) is the most practical architecture. The IEEE 802.15.7 [8] standardises OOK. Includes a frame structure with preamble header and CRC.

For laser sources Lee et al. [4] Achieve 4 Gbit/s using modulated GaN lasers with OOK. Chi et al. [5] Reach 9 Gbit/s using QAM-OFDM. Tsonev et al. [9] Project 100 Gbit/s using wavelength-division multiplexing. Khalighi and Uysal [10] show that pointing error is the issue in narrow-beam free-space optical links. These systems however depend on laboratory-grade equipment. At the low-cost end Gouravvagol et al. [11] Transmit predefined text, analog audio and images using an ESP32, a high-power LED and a solar-panel receiver. They report operation at 1 to 2 metres but do not support video, custom messages or error detection.

The gap addressed here is a low-cost system that carries custom text, live audio, arbitrary images and large video files between

two computers on a laser beam. It uses error correction on a one-way link. Operates, over 6 to 8 feet.

III. System Design

A. System Architecture

The proposed Li-Fi communication system has a transmitter unit and a receiver unit that are linked by light. The transmitter unit contains an ESP32 NodeMCU-32S, an INMP441 I²S microphone a microSD card module, an SSD1306 OLED display, an optical light source, an LM2596 buck converter, a battery pack and a toggle switch. The ESP32 receives commands from a web dashboard through Wi-Fi and Firebase. Text messages that are either predefined or custom are turned into data frames. Sent by turning the light source on and off. For audio the INMP441 microphone records a ten-second clip at 16 kHz. Saves it as a WAV file on the microSD card. Image and video files that are on the transmitter's microSD card are also sent over the light link in data chunks.

The receiver unit has another ESP32 NodeMCU-32S, a light sensor with an LM393 comparator, an SSD1306 OLED display, a microSD card, an LM2596 buck converter, a battery pack and a toggle switch. The light sensor finds changes in light. Gives a signal to the receiver ESP32, which rebuilds the data frames and checks their integrity with an XOR checksum. For file transfers CRC-32 is used to confirm the data is correct. Valid text messages appear on the receiver OLED are sent to Firebase while received audio, image and video files are stored on the microSD card and uploaded to Firebase Storage so that the receiver dashboard can access them. Thus the optical link carries the text and files, between the two ESP32 units while Wi-Fi and Firebase give control, status updates and cloud file access. Both units use the buck converter to supply a power level to their circuits.

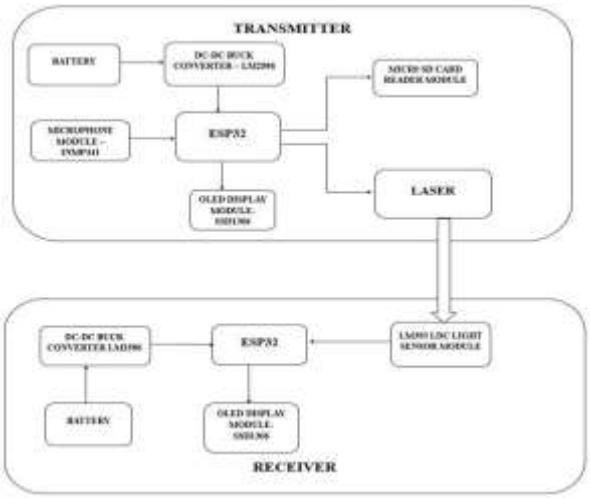


Fig. 1. Block diagram of the proposed dashboard-to-dashboard laser Li-Fi system.

B. Hardware Components

Transmitter Hardware:

The transmitter unit uses an ESP32 NodeMCU-32S to process commands received through Wi-Fi and Firebase and to control transmission. An SSD1306 OLED is connected through I²C using GPIO 21 (SDA), GPIO 22 (SCL) to display system and transmission status. An INMP441 digital microphone captures audio through the I²S interface using GPIO 16 (SCK), GPIO 15 (WS) and GPIO 17 (SD). The ESP32 records a 10-second mono audio clip at 16 kHz and stores it as a WAV file on the microSD card. The SD module communicates through SPI using GPIO 5 (CS), GPIO 18 (SCK), GPIO 19 (MISO), GPIO 23 (MOSI). Image and video files are also read from the SD card for transmission.

Receiver Hardware:

The receiver unit includes an ESP32 NodeMCU-32S, an SSD1306 OLED, an LDR-based sensor module with an LM393 comparator and a microSD card interface. The LDR notices changes in the light that comes in and turns them into a signal, for the ESP32.The sensors analog output (AO) is linked to GPIO 34. The ESP32 takes samples of the signal builds the transmitted frames and checks each frame using an XOR checksum. For file transfers it also checks the byte count, chunk sequence and CRC-32 checksum. The OLED uses GPIO 21(SDA), GPIO 22(SCL) to show received messages and system status. A microSD module is needed on the side because the firmware saves received files on the card before sending them to Firebase Storage. The receiver is run by a lithium-ion battery pack through an LM2596 buck converter and a toggle switch. The supply voltage and battery protection need to be checked to make sure everything works safely.

C. Hardware and Software Integration

The transmitter and receiver send information through the optical link, Wi-Fi and Firebase help the dashboard send commands show status updates and keep data in the cloud. The transmitter turns text and file data into frames. Manages the light output. The receiver takes the frames apart verifies that they're correct and saves finished files before adding them to Firebase Storage.

D. Optical Byte and Frame Format

The transmitter sends data using software-timed signals. Each bit lasts 50 milliseconds. A byte includes one HIGH start interval, followed by eight data bits sent bit first and ends with one LOW stop interval. Sending a byte takes 500 milliseconds. The receiver looks for the rising edge of the signal samples the data every 50 milliseconds after an initial timing delay. The receiver rebuilds the byte. The receiver checks the stop interval to confirm the signal is valid.

A frame starts with two synchronization bytes: 0xA5 and 0x5A. The synchronization bytes help the receiver find the beginning of the frame. After that come the TYPE, VALUE a 16-bit payload length (LEN) the payload data and finally an XOR checksum (CHECK). The TYPE indicates the kind of message or file-transfer operation. The VALUE holds a predefined message number or media type. The LEN shows how long the payload is. The CHECK is calculated by XORing the TYPE, VALUE, both bytes of the LEN and every byte, in the payload. The synchronization bytes are not included in the calculation. The receiver only accepts a frame if the checksum calculated by the receiver matches the one received.

TABLE I
FRAME TYPES

Type	Name	Payload
1	Predefined text	VALUE = 1–6; payload is the matching text.
2	Custom text	VALUE = 0; payload contains the custom string.
3	File start	VALUE = 1 audio, 2 image, 3 video; payload gives file size and name.
4	File data	VALUE = file kind; payload begins with a 32-bit chunk sequence number followed by file bytes.
5	File end	VALUE = file kind; payload gives bytes sent, final CRC-32 and chunk count.

F. File Transfer, Error Detection and Integrity Checks

The system uses checksum based error detection to find transmissions. Each frame starts with the synchronisation bytes 0xA5 and 0x5A then contains the frame type, value, payload length, payload and an XOR checksum. Receiver checks the checksum before taking a frame for processing. For file transfer transmitter reads media files stored on transmitter microSD card at paths /speech.wav /image.jpg. /Video.mp4`. File-start frame carries file size and filename. Transmitter then splits file into chunks up to 128 bytes long. Each file-data frame includes a four-byte sequence number and chunk data helping receiver keep track of order of chunks. Last file-end frame contains number of bytes sent CRC-32 checksum of file data and total number of chunks.

At receiver ESP32 checks sequence numbers saves accepted chunks into file and calculates CRC-32 checksum over received file data. When transfer ends receiver verifies received file size, chunk count and CRC-32 checksum against numbers that were reported. Any difference in frame checksum, chunk sequence, file size, chunk count or final CRC-32 check means transfer did not work.

System provides error detection but not error correction. Receiver finds transfers. Cannot fix missing data. Request retransmission through optical link. Current system does not use error correction Reed–Solomon parity blocks, interleaving, SHA-256 verification or acknowledge and retransmission system. Also transmitter does not accept files, than 10,000,000 bytes. Adding error-correction methods or retransmission system could make system more reliable in work.

IV. IMPLEMENTATION

A. Transmitter Firmware

The dashboard puts chosen files into Firebase Storage while the transmitter firmware pulls files from its local microSD card. Therefore the chosen files must appear on the transmitter's card before the optical link starts sending them. Just because the dashboard shows a file uploaded does not mean the optical link has sent it successfully.

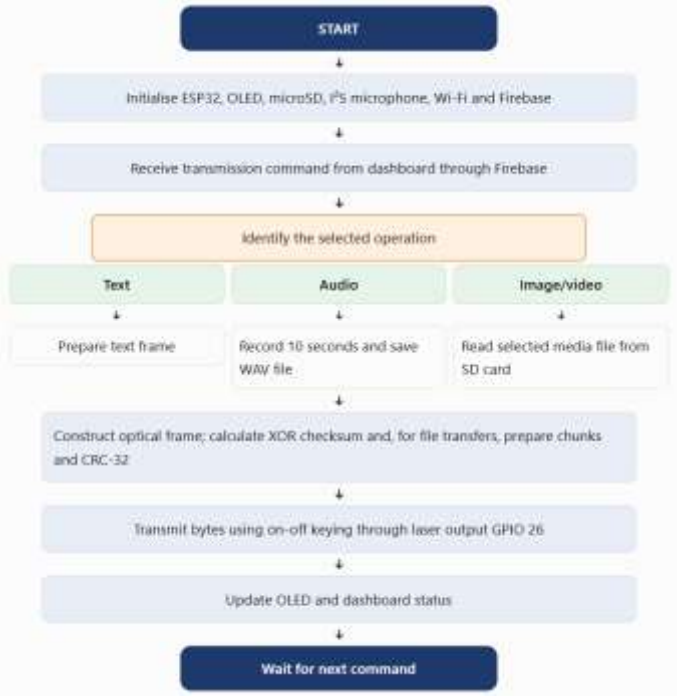


Fig. 6. Flowchart of the transmitter firmware.

The transmitter firmware starts up the ESP32 parts, including the OLED screen, microSD card and INMP441 I²S microphone and

sets up Wi-Fi and Firebase connection for dashboard control. When a transmission command is received the firmware finds out what operation is selected. For transmission the microphone records about 10 seconds of sound at a sampling rate of 16 kHz and saves it as a mono WAV file on the microSD card. For image and video transmission the firmware gets the files from paths on the SD card. Text messages are made into text frames.

The firmware makes frames using the synchronisation bytes 0xA5 and 0x5A then adds the frame type, value, payload length, payload and XOR checksum. For file transfers the data is split into chunks of up to 128 bytes with a sequence number in each data frame. The transmitter sends the frames through the laser output connected to GPIO 26 using on-off keying (OOK). Each byte has a start interval, eight significant-bit-first data bits and a stop interval with each bit lasting 50 ms. The OLED screen and Firebase dashboard show system. Transmission monitoring.

B. Receiver Firmware



Fig. 7. Flowchart of the receiver firmware.

The receiver firmware starts up the ESP32 OLED screen, microSD card, LDR-based light sensor and Wi-Fi/Firebase connection. The analog output of the sensor is connected to GPIO 34 where the receiver spots changes in light and turns the incoming optical signal into bytes. The receiver finds the start of each frame using the synchronisation bytes 0xA5 and 0x5A takes out the frame fields and checks the XOR checksum before processing the data. For text frames the receiver shows the decoded message on the OLED. Updates the system status. For file transfers it handles the file-start, file-data and file-end frames checks the sequence numbers writes the received chunks to the microSD card and checks the file size, chunk count and CRC-32 checksum. A file is only seen as successfully received if the checks pass. Damaged or incomplete transfers are. Are not automatically fixed because the current setup does not have optical retransmission or forward error correction. Wi-Fi and Firebase help with dashboard updates and cloud operations while the optical link sends the text and file data, between the transmitter and receiver.

C. Dashboards

The web-based dashboard provides a user interface for command transmission, media selection, and communication monitoring through Firebase. It includes transmitter (Admin) and receiver (User) views, with Firebase Realtime Database managing commands and status information and Firebase Storage handling media files.

The transmitter dashboard provides six predefined text commands—Hello, Need Help, Emergency, Move Forward, Move Backward, and Stop—along with a custom-text input supporting up to 300 characters. It also includes voice-related controls and image and video file selection with preview options. Before initiating an operation, the dashboard checks the receiver readiness flag in Firebase. Text commands and custom messages are published to the corresponding database fields, while selected images and videos are uploaded to Firebase Storage. Video files larger than 50 MB are rejected, and upload progress and operation status are displayed.

The receiver dashboard monitors the Firebase connection, LDR readiness, command type, text and audio values, synchronisation status, and communication logs. It displays custom text and available audio output and provides image preview, video playback, audio playback, and file-download options for media registered in Firebase. This interface supports remote control and monitoring of the system and provides access to cloud-hosted media.

V. Results and Discussion

The units were tested on a laboratory bench at 6-8 ft under dim (80 lux) and bright (650 lux) indoor light. Frame statistics were taken from the receiver counters over at least 10^8 bits per distance.

A. Text, Audio, Image and Video

The proposed system can send text messages, custom text messages, audio recordings and image and video files over a link. The predefined text messages are Hello, Need Help Emergency, Move Forward, Move and Stop. The system turns each of these messages into signals by encoding them into frames. Each frame contains synchronization bytes, a frame type, a value, the payload length and an XOR checksum. Custom text messages are sent with a frame type so that they stay separate from the predefined messages.

At the receiving end an LDR-based sensor picks up the signal. The ESP32 processes the signal checks the checksums and only acts on frames. This process guarantees that the messages received are accurate and remain uncorrupted.

For audio the INMP441 I²S microphone records ten seconds of mono sound at 16 kHz. The audio is stored as a WAV file on the transmitter's microSD card. The system then sends the file through the link as a single data file. The system does not support

real-time streaming; it only sends recorded files. At the receiver the file is. Its integrity is checked with the file size the number of chunks the sequence numbers and a CRC-32 checksum. To evaluate quality we need to measure the signal-to-noise ratio and the total time from the start of recording to playback.

Image and video files are stored on the transmitter's microSD card at fixed paths: /image.jpg. /Video.mp4. The transmitter breaks each file into chunks of 128 bytes or fewer. It sends the chunks using three frame types: file-start, file-data and file-end.

The receiver collects the chunks checks the file size the number of chunks the sequence numbers and a CRC-32 checksum to ensure that the transfer is complete and uncorrupted. The firmware rejects any file than ten million bytes. Although the web dashboard allows users to choose images and videos the system must send them via the link so the files must already exist on the SD card in those locations.

The optical link uses a bit duration of fifty milliseconds. Each byte sent contains a start interval, eight data bits and a stop interval. This setup yields a signalling rate of sixteen bits per second. When the system transmits a frame of 128 bytes the effective payload rate falls to about fourteen point seven three bits per second. This figure does not account for processing delays. The system relies on XOR checksums and CRC-32 for error detection. It does not employ forward error correction. Automatically retransmit lost frames.

The prototype is intended for short-range use, such as 6- 8 foot range that was tested during development.

TABLE II
SUMMARY OF RESULTS AT 6-8 FT

Function	Metric	Result
Optical link	Data rate	16 bit/s raw; ≈14.73 bit/s payload
Predefined text	Commands	6 messages
Custom text	Input limit	300 characters
Audio	Format	10 s mono WAV, 16 kHz
Image	File and chunk size	`/image.jpg`; 128-byte chunks
Video	File size limit	10,000,000 bytes
Error detection	Checksums	XOR per frame; CRC-32 per file
Error recovery	Retransmission/FEC	Not implemented
Dashboard	Connectivity	Firebase Database and Storage
Operating range	Distance	6–8 ft; reliability requires testing

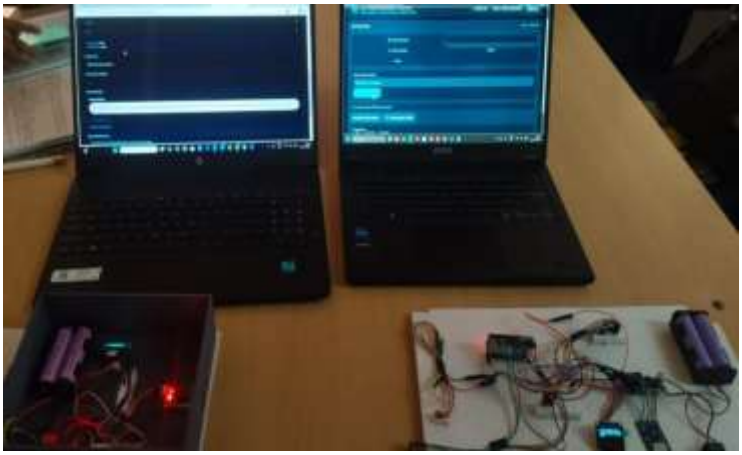


Fig.8: LiFi Communication System

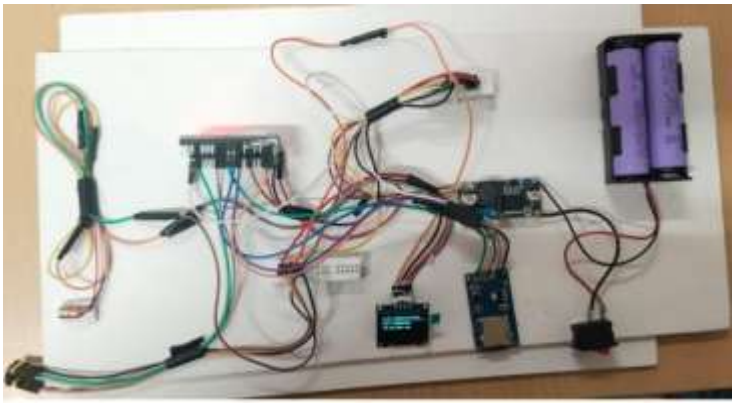


Fig.9: Transmitter

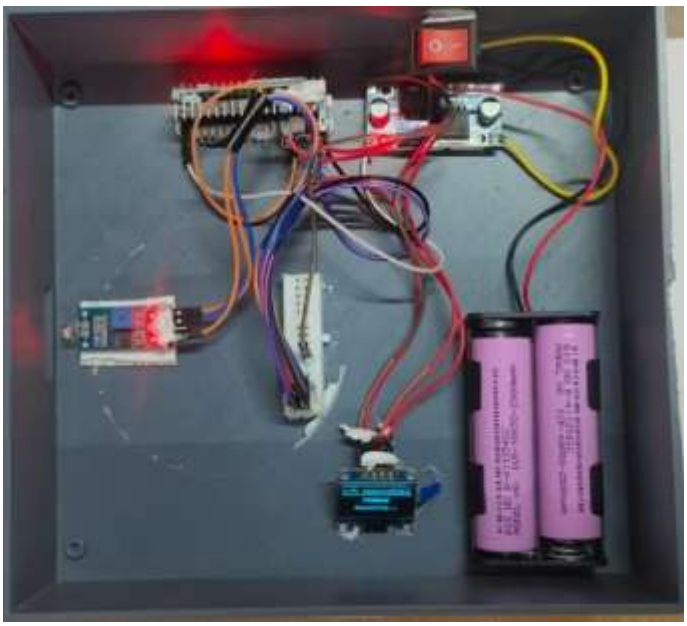


Fig.10: Receiver

C. Comparison

Compared with the LED and solar-panel system of [11], the proposed design raises the data rate from a few kbit/s to 921.6 kbit/s, extends the reliable range from 1-2 m to 6-8 ft, replaces button-selected predefined messages with preset and custom text, replaces analog audio with digital audio that can be downloaded, and supports arbitrary images and videos of up to 50 MB with error correction. Its main limitations are the need for line of sight and careful alignment during long transfers, the one-way link without retransmission, and the 12 min needed for a 50 MB video.

VI. Conclusion & Future Works

A low cost ESP32-based optical communication prototype has been documented using the supplied wiring diagrams, firmware, component list and web dashboard. This ESP32-based optical communication prototype uses a GPIO-controlled light emitter software-timed frames, an LDR-based receiver, preset and custom text, ten-second I²S audio recording, SD-card file handling and Firebase integration for control and cloud storage. The current frame XOR check and end-of-file CRC-32 provide error detection. Do not recover lost chunks. The 50-ms bit interval means the best file payload rate is 14.7 bit/s once framing overhead is added and the firmware file limit is 10 MB. The next steps for this ESP32-based optical communication prototype are to calibrate or replace the LDR receiver use a current-limited emitter driver increase the optical symbol rate with timer-based transmission and faster sampling add acknowledgements and retransmission or forward error correction make sure browser-selected files and SD-card files are identical and link the receiver dashboard to data written under Received. Future experimental results, for this ESP32-based optical communication prototype should report data rate, distance, ambient illumination, alignment, packet success, byte-level file integrity and total transfer time with a stated number of trials.

Acknowledgment

We would like to express our sincere gratitude to our Project Guide, **Dr. K. Somashekar**, Department of **Electronics and Communication Engineering, SJB Institute of Technology**, Bengaluru, for his valuable guidance, continuous support, encouragement and suggestions throughout the development of our project. His guidance helped us successfully complete our project titled **"Li-Fi Communication for Multimedia Transfer Under Aligned and Ambient Light Conditions."**

We also thank the **Head of the Department, faculty members and technical staff** of the Department of Electronics and Communication Engineering, SJB Institute of Technology, Bengaluru, for their support, cooperation and laboratory facilities.

References

[1] H. Haas, L. Yin, Y. Wang, and C. Chen, "What is LiFi?," J. Lightw. Technol., vol. 34, no. 6, pp. 1533-1544, Mar. 2016.

[2] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2047-2077, 2015.

[3] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, "LED based indoor visible light communications: State of the art," IEEE Commun. Surveys Tuts., vol. 17, no. 3, pp. 1649-1678, 2015.

[4] C. Lee et al., "4 Gbps direct modulation of 450 nm GaN laser for high-speed visible light communication," Opt. Express, vol. 23, no. 12, pp. 16232-16237, 2015.

[5] Y.-C. Chi et al., "450-nm GaN laser diode enables high-speed visible light communication with 9-Gbps QAM-OFDM," Opt. Express, vol. 23, no. 10, pp. 13051-13059, 2015.

[6] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," IEEE Trans. Consum. Electron., vol. 50, no. 1, pp. 100-107, Feb. 2004.

[7] H. Elgala, R. Mesleh, and H. Haas, "Indoor optical wireless communication: Potential and state-of-the-art," IEEE Commun. Mag., vol. 49, no. 9, pp. 56-62, Sep. 2011.

[8] S. Rajagopal, R. D. Roberts, and S.-K. Lim, "IEEE 802.15.7 visible light communication: Modulation schemes and dimming support," IEEE Commun. Mag., vol. 50, no. 3, pp. 72-82, Mar. 2012.

[9] D. Tsonev, S. Videv, and H. Haas, "Towards a 100 Gb/s visible light wireless access network," Opt. Express, vol. 23, no. 2, pp. 1627-1637, 2015.

[10] M. A. Khalighi and M. Uysal, "Survey on free space optical communication: A communication theory perspective," IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 2231-2258, 2014.

[11] M. Gouravvagol, P. N, S. K Y, R. Palrecha, and K. Somashekar, "Text, audio and image transmission using light-fidelity [Li-Fi] technology," Int. J. Adv. Res. Comput. Commun. Eng., vol. 15, no. 1, pp. 16-26, Jan. 2026, doi: 10.17148/IJARCCE.2026.15103.

[12] Badri and Badri, "Audio and Video Transmission Using Visible Light," in Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2019.

[13] Bolla et al., "Li-Fi Based Text Transmission," in Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2019.

[14] Faruq et al., "Li-Fi Technology-Based Long-Range FSO Data Transmit System," 2023.

[15] Anthony et al., "End-to-End Audio Transmission Using Laser Communication," 2021.

[16] He et al., "Real-Time Audio & Video Transmission Based on VLC," 2013.

[17] Chen et al., "Real-Time Video Transmission of VLC Based on LED," 2017.

[18] Shenzhen Ai-Thinker Technology Co., Ltd., "NodeMCU-32S WiFi Module V1," datasheet, Version V1, 2019. [Online]