Project 1

Odin Augustine Erik Lundstam,
Rajvir Singh, James Guentert

BME 60A | Dr. Daryl Preece | 10am

Background Research: LiFi transmission

Electromagnetic (EM) communication is the most widely adopted form of wireless transmission, utilizing frequencies of 2.4 GHz, 5 GHz, and 6 GHz in technologies such as Wi-Fi, Bluetooth, and cellular networks. Hospitals and medical facilities rely on Wi-Fi to transmit sensitive information and coordinate operations, while growing consumer demand for high-speed data places increasing strain on existing wireless infrastructure. Despite its effectiveness, EM communication faces critical limitations in environments with high device density and security concerns. Li-Fi (Light Fidelity) presents a viable alternative by addressing interference, data security risks, and bandwidth constraints.

The widespread deployment of GHz-bandwidth communication in medical settings introduces significant security risks, as public Wi-Fi networks remain vulnerable to a growing number of sophisticated cyber threats [1]. Wi-Fi routers broadcast signals omnidirectionally, making unauthorized interception possible for any device within range of the transmission. Li-Fi mitigates this risk by confining data transmission to the visible light spectrum, which cannot penetrate opaque barriers, ensuring that data remains inaccessible outside the physical space of transmission.

Beyond security, Wi-Fi performance in hospital environments is further degraded by signal interference. In hospitals, thick concrete walls contribute to low-frequency signal reflection, and numerous electronic devices—such as laptops, anesthesia gas machines, medical imaging equipment, and life-support systems—compete within the same radio frequency spectrum [2]. In such high-density environments, where multiple devices must operate with minimal latency and maximum reliability, the interference inherent to Wi-Fi networks poses a serious limitation.

Li-Fi offers a more stable wireless alternative since visible light mitigates interference issues inherent to Wi-Fi. Unlike Wi-Fi, where most devices are calibrated to specific shared frequency bands for full network compatibility, Li-Fi can utilize the broader visible light spectrum, enabling transmission on distinct wavelengths with minimal interference. Baseline noise from ambient lighting can be mitigated through thresholding techniques that distinguish data signals from general illumination, while communication paths can be further isolated using opaque barriers to block unwanted signals or optimized with phase cancellation outside the transmission path to suppress interference. Additionally, since LED lighting primarily emits three distinct wavelengths—red, green, and blue—unused portions of the visible spectrum could be allocated for data transmission, reducing spectral overlap and enhancing signal clarity while preserving standard illumination. In environments requiring strict control over signal integrity, integrating fiber optics with Li-Fi infrastructure could further enhance communication stability by providing high-speed, interference-free transmission channels. While some of these solutions remain theoretical in their application to Li-Fi, their feasibility is grounded in established optical and signal processing technologies, making Li-Fi a compelling alternative for secure, high-speed,

and interference-resistant wireless communication in environments where network reliability is critical.

For long-distance optical communication, Li-Fi is constrained by atmospheric effects such as spectral absorption and more so from group velocity dispersion. Spectral absorption, primarily due to atmospheric moisture, attenuates specific wavelengths, weakening signal intensity over distance. Group velocity dispersion, caused by variations in the refractive index of the atmosphere across different wavelengths, results in pulse broadening and signal distortion, ultimately limiting data rates and transmission range. Addressing these challenges requires high-power optical transmitters, adaptive optics, and advanced attenuation techniques to maintain signal integrity over extended distances [3]. While Li-Fi is not yet a universal replacement for Wi-Fi, its potential provides significant advantages where data security, network stability, and high-speed communication are critical. As research advances in mitigating dispersion and absorption effects, Li-Fi may become a more practical alternative to traditional EM-based communication.

Implementation: Lifi Communication

Our information transmission method employs character-to-binary 8-bit conversion (ASCII), the standard encoding system used by computers to interpret text. ASCII assigns the first 97 values to control functions, symbols, and capital letters; however, to streamline transmission speeds, our Li-Fi implementation focuses solely on lowercase letters, excluding these additional characters.

Using Arduino, a light source is programmed to flash a number of times equal to the character's position in the lowercase alphabet (ASCII value minus 97), ensuring that 'a' corresponds to one flash and 'z' to 26 flashes. On the receiving end, a photoresistor detects and counts the flashes, adds 97, and retrieves the corresponding character from the ASCII table. This approach provides a straightforward and effective method for wirelessly transmitting simple messages between computers using visible light.

Synchronization is not a concern in this system, as the receiver continuously reads incoming signals while the transmitter operates on a fixed timing schedule. Each flash is separated by a 50ms delay, ensuring the receiver correctly counts individual pulses. To prevent ambiguity between consecutive characters, a three-second pause is introduced after each character. For example, since 'a' is represented by one pulse and 'b' by two pulses, without a distinct pause, the receiver might misinterpret them as a single three-pulse signal corresponding to 'c'. The three-second wait between characters ensures that individual flashes are correctly interpreted, preventing consecutive pulses from being blended into an unintended character. While this flash-based ASCII transmission method is functional, it is inefficient for larger-scale implementations. Expanding the system to accommodate all ASCII characters and improve transmission speed necessitates a shift to a binary-based encoding scheme.

Proposed Improvement: Redundancy Packet Encoding

To integrate this system into actual hardware, full ASCII character compatibility is necessary, and while the current method of counting flashes is functional, binary encoding provides a more efficient means of communication. On average, transmitting characters in decimal format requires 128 pulses per character, whereas binary encoding requires only eight pulses, significantly reducing transmission time. However, this introduces a fundamental challenge: distinguishing sequences of consecutive 0s or 1s, as the absence of transitions makes it difficult to determine where one character ends and another begins. This issue is particularly problematic with runs such as 000000 or 1111111, where a receiver must infer the number of bits based solely on pulse duration, which can lead to severe synchronization errors. A robust synchronization strategy is required—one that eliminates reliance on timing intervals and ensures the receiver can reconstruct the message unambiguously.

Inspired by DNA replication, noncoding redundancy packets can be implemented to eliminate ambiguity caused by consecutive voltages. These packets act as structured delimiters, appearing between characters and at the end of the message to ensure that sequences of repeated 0s or 1s do not interfere with decoding. The encoding process works by doubling every bit, expanding an 8-bit sequence (10101010) into 16 bits (1100110011001100), ensuring that every transition remains distinct. Between these augmented sequences, a predefined 1010 redundancy packet is inserted, and the message concludes with another 1010 to mark the end. This structure ensures that 1010 sequences cannot naturally occur within the coding data, making them impossible to misinterpret. Unlike traditional encoding approaches, this method eliminates the need for the receiver to measure pulse durations to infer the number of bits, as the redundancy inherently encodes separation between coding symbols rather than relying on timing constraints.

On the receiving end, the system first detects and removes all 1010 sequences, systematically stripping away the noncoding packets. What remains is the augmented bit sequence, which is then restored to its original form by removing every second bit (2n), reconstructing the initial data exactly as it was transmitted. This method guarantees that symbol boundaries remain intact, eliminating the need for timing-based synchronization mechanisms, which are highly susceptible to drift. By ensuring that redundancy packets cannot be replicated within the coding data, this approach prevents false synchronization errors, making decoding a straightforward process of pattern matching and reduction rather than requiring complex real-time timing adjustments. See Figures 1 & 2 for a visual demonstration of the decoding process.

A traditional Forward Error Correction (FEC) approach would likely rely on methods such as Hamming codes or Reed-Solomon codes, which are designed primarily for bit-flip correction—that is, recovering from noise-induced changes in individual bits. However, these methods assume that the length of the transmitted data remains intact and do not effectively handle deletion and insertion errors, which are the primary concern in this system. More relevant to this problem, convolutional codes and turbo codes rely on probabilistic reconstruction techniques such as maximum-likelihood estimation, which infer missing bits based on statistical

models. While these methods are effective for large datasets with continuous streams of information, they are less practical for small-scale transmissions like this, where explicit separation between coding symbols provides a more deterministic solution. That said, convolutional error correction could be a valuable addition for larger datasets, particularly if this system were to scale into high-speed Li-Fi communication with longer messages or real-time data streams. The use of systematic redundancy rather than parity checks eliminates the need for iterative decoding algorithms, making this a computationally simple and robust solution for ensuring unambiguous data transmission.

While this method increases transmission sizes by a factor of six, this level of redundancy is the minimum required to ensure that coding bits remain distinct and unambiguous. The goal of noncoding redundancy packets is to introduce separation between coding bits that cannot be mistaken for one another, eliminating the need for pulse duration-based synchronization, which is prone to drift. Future improvements could focus on introducing a noncoding start sequence, which would instruct the receiver when to start recording rather than relying on it to be active when the message begins. This would be particularly beneficial for higher refresh rate systems, where synchronization drift is more pronounced. However, this does not entirely eliminate synchronization concerns, as successful reception is still dependent on the relative phase of each program's refresh cycle. In theory, matching the clock speeds of the transmitter and receiver would mitigate synchronization issues further, but this has not been tested in the current system. Ultimately, this redundancy-based approach offers a deterministic and computationally simple solution to deletion and insertion errors, ensuring self-delimiting symbol transmission without requiring complex reconstruction algorithms. While bandwidth efficiency and transmission speeds are sacrificed, the tradeoff is justified for unambiguous, reliable, and easy-to-decode Li-Fi transmission.

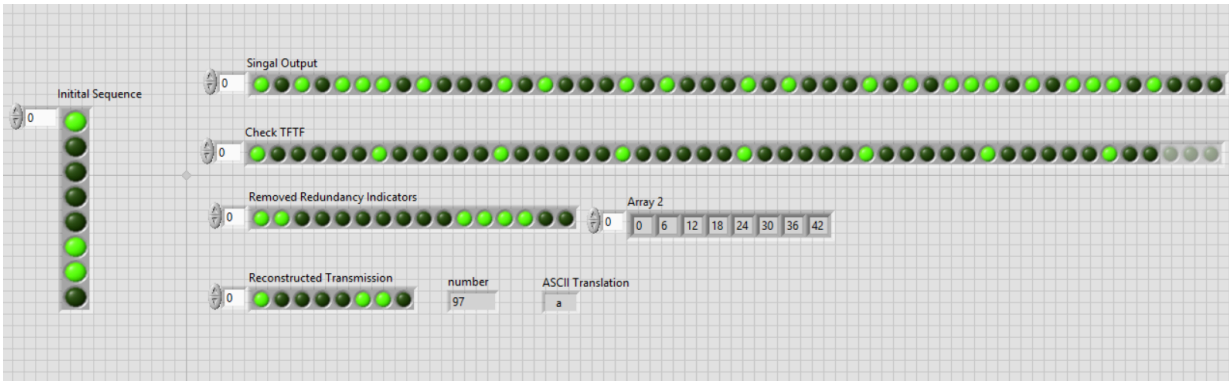

Figure 1: The **Initial Sequence** (left panel) shows the binary input **01100001** before encoding, representing the **original 8-bit ASCII character**. The **Signal Output** (top row) displays the received signal after **ADC**, where bit-doubling and noncoding redundancy packets are visible. Below, the **Check TFTF** step detects occurrences of the **1010 redundancy packet**, marking their start indices in **Array 2** for later removal. The **Removed Redundancy Indicators** row presents the **augmented message** after extracting noncoding packets, revealing the **doubled bit version of the original transmission**. Finally, the **Reconstructed Transmission** restores the original **8-bit sequence** by removing every second bit (2n), allowing for conversion into a **decimal value (97) and its corresponding ASCII character ('a')**.

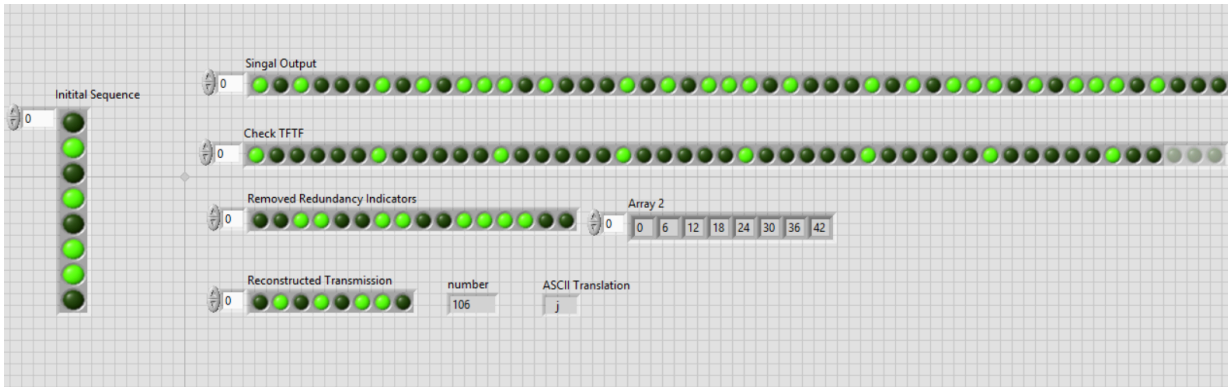

Figure 2: This figure follows the same decoding process as Figure 1, but for a different input character. The **Initial Sequence** (left panel) represents the binary input **01101010**, corresponding to ASCII **106 ('j')**. The **Signal Output** again contains **redundancy markers and bit-doubling**, which are detected in **Check TFTF** and logged in **Array 2**. The **Removed Redundancy Indicators** row strips these markers, revealing the **doubled bit transmission**, which is then **reduced back to its original 8-bit form** in **Reconstructed Transmission**, confirming the final output as **'j'**. These figures demonstrate the **effectiveness of redundancy packets in preventing synchronization errors**, ensuring **clear character reconstruction** from Li-Fi signal data.

References

[1] Aura, "The Dangers of Public Wi-Fi: How Hackers Can Steal Your Data & How to Stay Safe," *Aura.com*, [Online]. Available: https://www.aura.com/learn/dangers-of-public-wi-fi.

[2] 7signal, "WLAN Woes: Why Wi-Fi Fails in Hospitals," *7signal.com*, [Online]. Available: http://7signal.com/news/blog/wlan-woes-why-wi-fi-fails-in-hospitals#:~:text=Challenging%20physical%20environment.,to%20its%20standard%20computer%20system.

[3] A. Singh and P. Kumar, "Atmospheric Propagation Limitations for Free-Space Optical Communications," *arXiv*, Apr. 2021. [Online]. Available: https://arxiv.org/pdf/2104.00611.