Li-Fi - Visible Light Communication System

Yiren Qu

Lyndon Institute

Project Advisor: John McKinnon

Table of Contents

Abstract

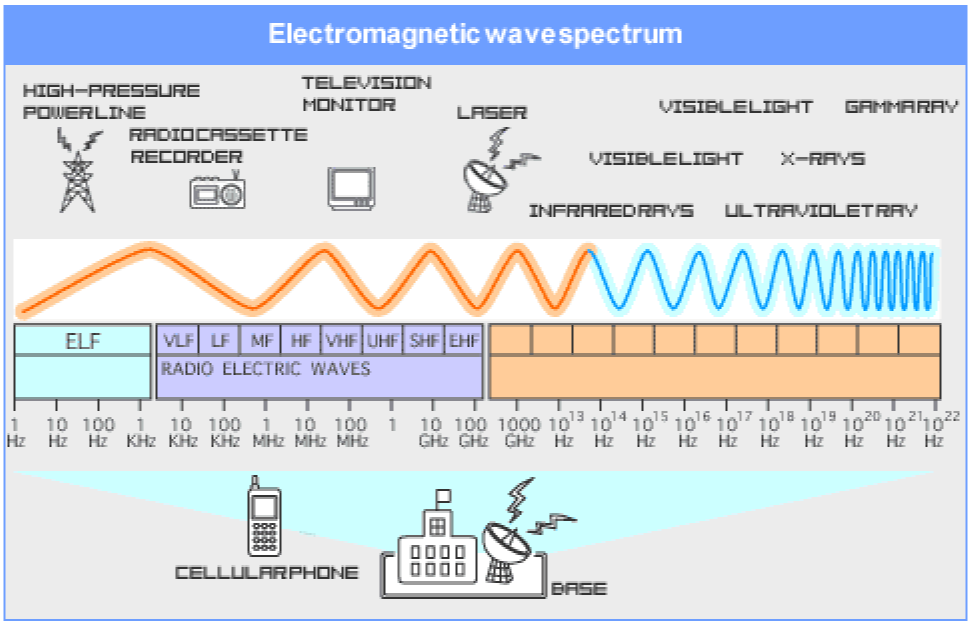
Wireless communications, such as Wi-Fi and Bluetooth, are commonly used in the daily life. However, while more and more wireless devices occupy the available frequencies, the frequency spectrum is reaching a maximum capacity.

Thus, Li-Fi, communicating using light waves between two devices, becomes a future solution to transmit data. LED devices with a data wave generator may keep the lighting functionality of a LED while sending data in the background. With the full range of the visible and infrared light spectrum, we may transmit data day and night. With Li-Fi, data communication happens indoors with low latency and high speed and is hard to hijack from outside the room. On the receiver side, a small solar panel both receives the signal and recycles the power to charge the battery for future use. It thus saves energy lost in standard wireless data transmission. The system consists of two Arduino (microcontroller) devices as the emitter and receiver. The user interface, based on the Python programming language, is used to control sending and receiving data. Infrared LED signals the beginning and end of the data transmission During the transmission, LEDs switching on and off signal binary 0's and 1's.

This system can transmit text and images successfully. However, with the limitation of the consumer-grade solar panel's response time, it is hard to reach high speeds in data transmission. With high-end and low-response-time components, it will be able to transmit live videos at high speeds. The Li-Fi system not only provides the basic lighting function but also sends the data and power to the receiver side.

1 Background & Introduction

Wireless communication techniques are commonly used in modern life. It is defined as the communication between two points without direct physical connection or electrical conductor between them. In the 2000s, wi-fi, Bluetooth, radio, were the standard types of the wireless communication. Wi-Fi transmits data via electromagnetic waves between devices. Most other common forms of wireless communication also embed information in the electromagnetic wave. They all fall in certain ranges of the radio frequency spectrum.



Graph 1.1 radio frequency spectrum

In the radio frequency spectrum or electromagnetic spectrum, Wi-Fi is 2.4 GHz and 5GHz, and visible light is between 430 to 770 THz.

The base unit for networking connection is a bit, which is 1 or 0. The bottom layer of the networking is a physical layer which transmits raw data bits. The device converts the information to binary and transmits to the other side. The other end of the system will convert back to the same encryption method to get the information. The speed of transmitting bits to the other end is called data transmission rate, which is also known as baud rate or bit rate. In wireless connections, it depends on bandwidth (max bit rate), power and ambient noise. Between Wi-Fi and visible light, visible lights have higher frequency so that they have greater bandwidth than Wi-Fi.

2 Problem Formulation

2.1 Wireless connection problems

Wireless data transmissions are around us all the time. However, household machines like microwaves, cordless phones, and others use the same frequency as wi-fi and may interfere the wi-fi connection. Also, other factors like the physical space and certain materials may decrease the quality of the connection. Further, wireless hackers can use a fake Wi-Fi signal to attract people to connect through their network and retrieve the information from their devices. Although wi-fi can be protected by passwords, they are sometimes prone to hacking. This project's initiative is to find another way to increase the security of the data and increase the stability and the speed of the data transmission.

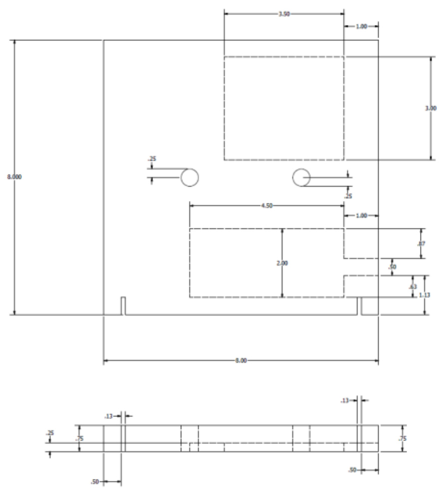
2.2 Energy problems

Wi-Fi communication needs a modem to emit electromagnetic waves to devices, and people usually turn the modem on all the time. Although the energy loss from the modem is not much compared to other electronic equipment, we should still consider other alternative ways to reuse the energy being transmitted. Also, in some emergency situation, if the devices are low in power, connecting to wi-fi would not be an ideal choice as the remaining power will be consumed quickly. In the rural places, they not only need data access for the information but may also need a supply of power to run electronic devices.

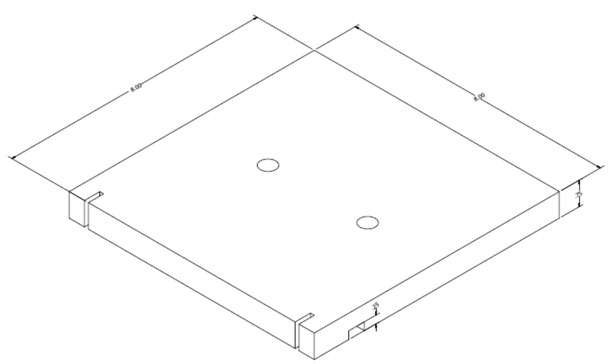
3 System Design

3.1 Emitter design

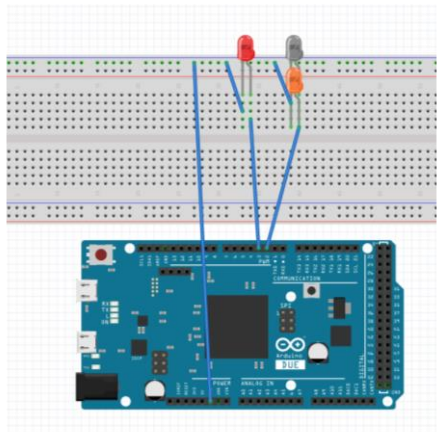
The Emitter side of the Li-Fi system has a LED, and an infrared LED for data transmission purpose. The emitter panel design is shown below, and it is printed with the reusable 3D material.



Graph 3.1 Emitter Panel Blueprint (1/2)



Graph 3.2 Emitter Panel Blueprint (2/2)



Graph 3.3 Emitter Circuit and Wiring Diagram

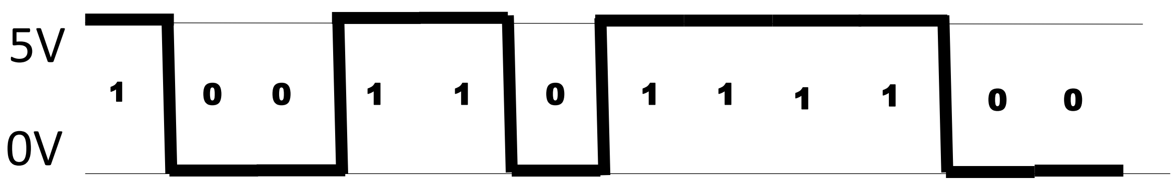
3.2 Emitting Data

The Arduino transmits the binary data by manipulating the LED. A custom designed encoding method is the protocol used to emit a bit.

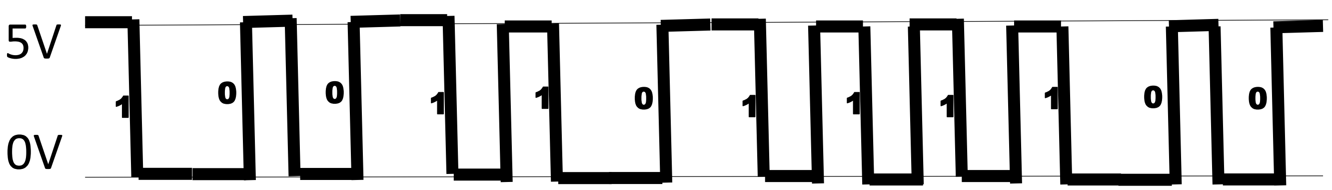
I. Emit in one single channel: (If we set the bit time, the delay time for one bit, is t seconds):

a. The Naïve Solution, which is easiest to complete and does not think about the error correction and other interferences. Turning on the LED for t seconds means 1 and turning off for t seconds means 0. If the raw bit data is 100110111100, the LED on and off diagram is shown below.

Graph 3.4 Single Channel & Easy Approach

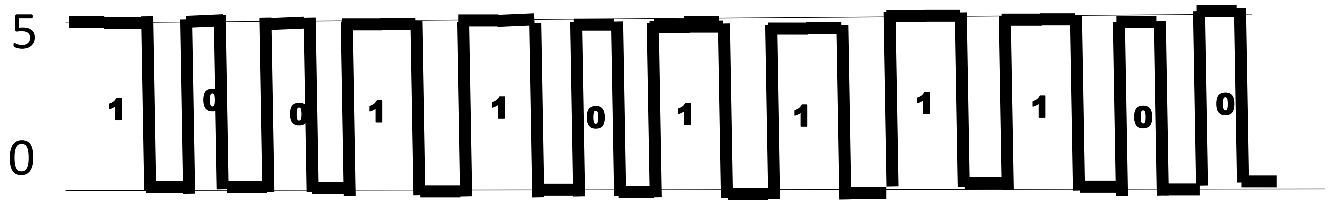


b. The Manchester encoding method. When using the naive solution, the receiver side may false read the 1 and 0, and bit shifting may happen. The Manchester encoding method can be used to prevent this type of error. In this approach, turning on the LED for t/2 seconds and off for t/2 seconds means 1. The diagram is shown below.



Graph 3.5 Single Channel & Manchester Encoding

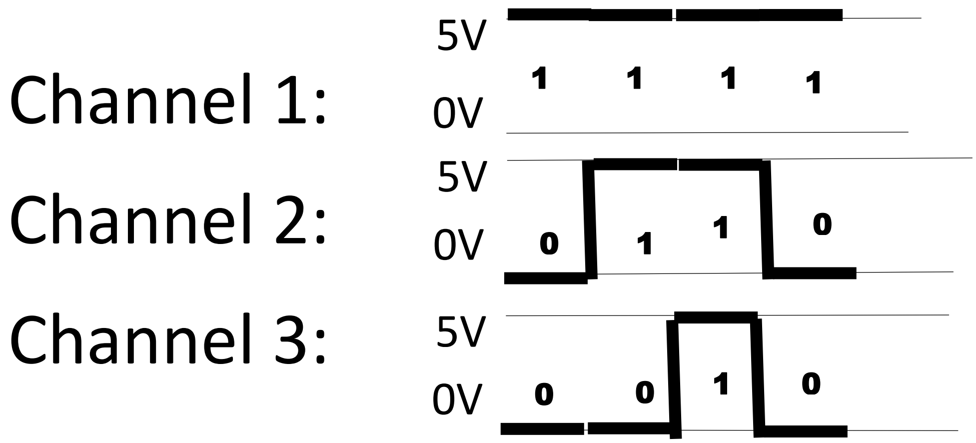
c. The Time Difference Method. The two approaches previously discussed use LED's intensity for bit data. However, if the transmitting LED is too far away from the receiver, the difference of the voltage received when LED on and off is small which makes it harder for the receiver to identify the corresponding bit information. The time difference method measures the time difference between the last off and current off. If the time delay is t seconds, it means 1, and if the time delay is t/2 seconds, it means 0. The diagram is shown below.



Graph 3.6 Single Channel & Time difference method

II. Multi-channel:

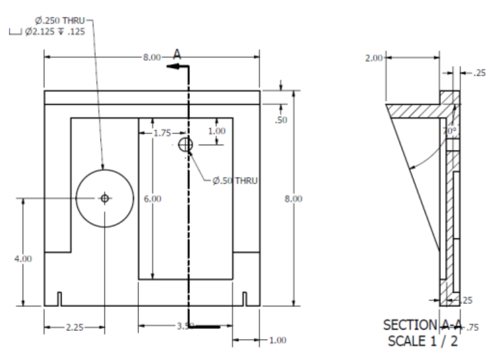
When using a white LED to transmit data, it is only one channel transmission. In theory, an n-channel communication may increase the bit rate 2^n times. In this case, we may use RGB (red, green, blue) to express white light and transmit information with three channels. The diagram is shown below.



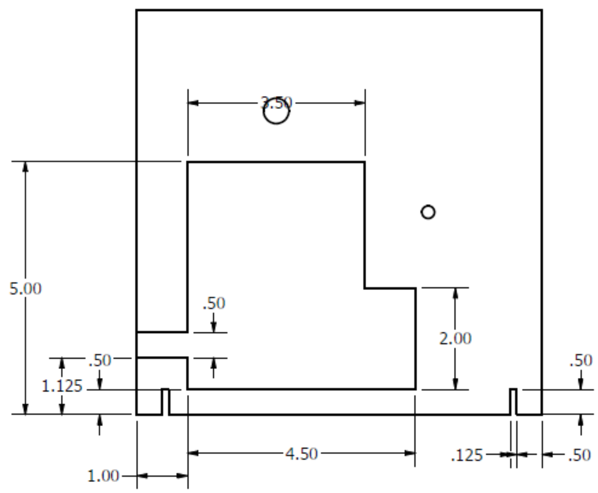
Graph 3.7 Multi-Channel & Easy Approach

3.3 Receiver design

Graph 3.8 Receiver Panel 3D model

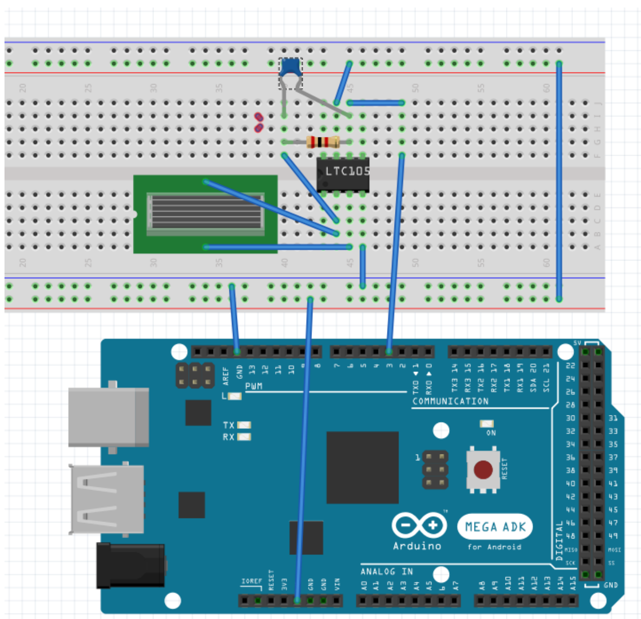


Graph 3.9 Receiver Panel Blueprint (1/2)



Graph 3.10 Receiver Panel Blueprint (2/2)

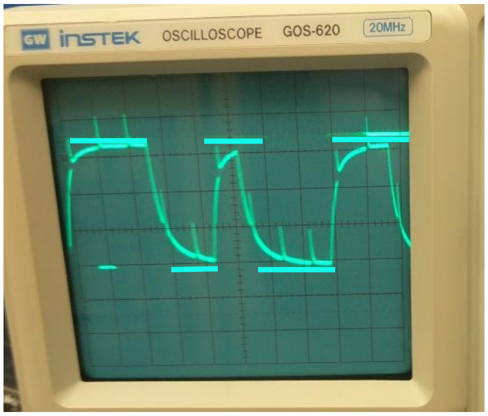
The circuit used on the receiver is a photodiode-amplifier circuit, which amplifies the electric wave generated by the solar panel. The potential difference when the light is on and off is bigger than the difference directly received. It lights another visible LED to signal whether the infrared LED is on or off.



Graph 3.11 Solar Panel Amplifier Circuit

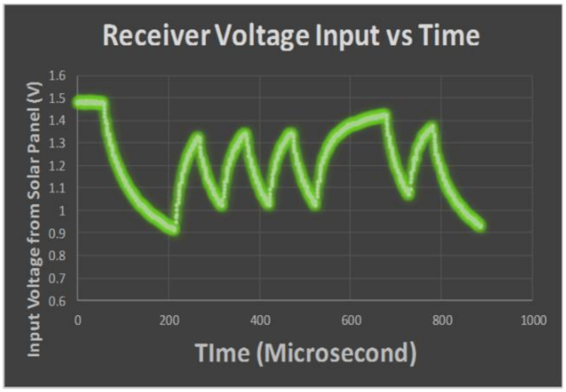
3.4 Receiving data

When setting up the receiver in a new environment, it needs to re-calibrate the threshold, which is the average of the peak and valley potential difference received from ambient light. When detecting the potential difference, the Arduino will compare it with the threshold value. If the value received is higher than the threshold value, it is considered as a High/Peak, vice versa. The Arduino converts the analog data to digital data with this method. The image shown below is taken from an oscilloscope, connected to the solar panel.



Graph 3.12 Receiver potential wave – oscilloscope

Moreover, the received analog value from the solar panel is also captured by the Arduino. The graph shown below demonstrates the charge and discharge period of the solar panel. The limit of the charging period affects the limit of the bandwidth of the transmission. And from the experiment, one millisecond is the average response time of the solar panel in the system to charge or discharge 3 volts fully.



Graph 3.13 Receiver voltage received vs. Time Graph

3.5 Synchronization

The receiver and emitter need to be synchronized for every transmission. Otherwise, the bit may be shifted and the received data may not be accurate. A protocol is required to control the start and end of the data transmission. The protocol uses the infrared LED to signal the beginning and end of the data transmission. As such, both sides have the same bit time, which is the time delay for a bit. Moreover, the receiver will record the sensor data at the middle point of a bit's peak or valley.

3.6 Data protocol design

The protocol of this system is similar to UDP (User Datagram Protocol) which transmits data one way from the server side to the client side. The data packet (the smallest unit of a transmission) is 15 bits to prevent false readings.

One complete data transmission contains a header and multiple data packets. The header gives the number of the packets in one series. Both header and packet are using binary encoding.

3.7 Python serial communication

In the project, the computer is both the server and the client. On the server side, it connects to the emitter panel. The limit for a computer to send binary to the Arduino serial monitor one-time is 540 bits. Moreover, after the Arduino receives the raw bits, it will regroup them into 15 bits/packet and manipulate the LED based on the raw bits.

On the other side, the receiver always sends the raw bits back to the computer, and the computer will regroup the raw bits and convert back to useful information.

The computer uses Python to communicate with the serial monitor of the Arduino.

3.8 Transmit texts

The computer uses Python to read the text, and in the programming, they are a character array. For every character, Python converts it to binary based on ASCII encoding and pads 0s on the left to make every character 8 bits long. The raw bits are regrouped to 540 bits/group and sent to the Arduino. When the Arduino receives 540 bits, it regroups them to 15 bits/packet. The Arduino manipulates the LED based on the raw bits.

The Receiver side receives the data header first and converts it to a number, which indicates the number of packets needs to be received. The Python receives the same amount number of the data packet. After receiving all raw bits, the program regroups them to 8 bits/group and converts them back to characters.

3.9 Transmit images

When using Python to read a picture, the images can be read as a base64 file, which is a long character array. The transmission uses the same method to transmit this array as a long text to the other side. The receiver converts back base64 file from the string.

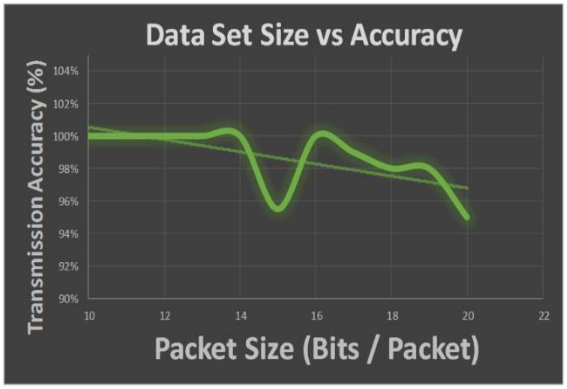
4 Improvement & Optimization

4.1 Receiver transmission rate limit

The transmission rate is hard to be upgraded. On the emitter side, we can easily control the LED blink rate up to 1MHz rate. However, on the receiver side, based on the results from the experiments, the discharge needs 1 ms to complete. A better solar diode which has higher response time may solve this problem.

4.2 Transmission data protocol improvement

When transmitting the data, the accuracy of the transmission (packet loss rate) is a factor to describe the quality of the connection. In this case, the packet size should be related to the transmission quality, because the more variation of the wave received on the receiver side, the harder it is to set a clean threshold value, in another word, it is more difficult to determine whether it is on a high peak or a low valley. The Arduino transmitted binary data size from 10 bits/packet to 20 bits/packet, 200 times for one size. The packet loss rate increases from 0% (size:10 bits) to 5% (size:20 bits). The quality of the connection became unstable when greater than 18 bits. However, the trend is decreasing. So, it shows the relationship between size and the packet loss rate. The graph shown below demonstrates the relationship between packet size and the accuracy of the transmission.



Graph 4.1 Packet Size vs. Transmission Accuracy

5 Conclusion and Future Goals

5.1 Conclusion

The purpose of this research is to develop a potential alternative wireless connection for Wi-Fi. In the process of this project, a two-sided transmission system can transmit texts and pictures successfully with Li-Fi technique. The system with a naive solution can reach 1kbit/s baud rate with a single channel. Li-Fi can not only transmit information and keep its lighting function but can also send energy to the other side of the system.

5.2 Future Goals

- Increase the transmission rate: Using high-end solar diodes which have fast reaction rate of light.
- Broadband/multi-channeling: When using multiple channels, the problem is that each electric curve is different and should be configured by itself. However, the Arduino can only process one thread at a time. For Li-Fi multi-channeling, it may use different colors of the light (different wavelength) to transmit data to eliminate the interference among them. Also, using other types of the microcontroller which has the function to processing multiple channels at the same time.

Bibliography

“DIY Science: Measuring Light with a Photodiode II.” Outside Science, 18 Apr. 2013, outsidescience.wordpress.com/2012/11/03/diy-science-measuring-light-with-a-photodiode-ii/. Accessed 26 Mar. 2017.

Schmid, Stefan, et al. “An LED-to-LED Visible Light Communication System with Software-Based Synchronization.” 2012 IEEE Globecom Workshops, 2012, doi:10.1109/glocomw.2012.6477763.

Khan, Fazlullah, et al. “Applications, Limitations, and Improvements in Visible Light Communication Systems.” 2015 International Conference on Connected Vehicles and Expo (ICCVE), 2015, doi:10.1109/iccve.2015.46.

Khan, Talha A., et al. “Visible Light Communication Using Wavelength Division Multiplexing for Smart Spaces.” 2012 IEEE Consumer Communications and Networking Conference (CCNC), 2012, doi:10.1109/ccnc.2012.6181092.

Grobe, Liane, Anagnostis Paraskevopoulos, Jonas Hilt, Dominic Schulz, Friedrich Lassak, Florian Hartlieb, Christoph Kottke, Volker Jungnickel, and Klaus-Dieter Langer. "High-speed visible light communication systems." IEEE Communications Magazine 51.12 (2013): 60-66. Web.

Թագուհի Սաղաթելյան. Electromagnetic Waves. Digital image. Wikimedia. N.p., May 2015. Web.