Building a *Bidirectional* Visible Light Communication Link: Challenges and Contributions

Pranav Harathi

Electrical and Computer Engineering
University of Texas at Austin
Honors Thesis

Visible Light Communication is a new information transmission method that involves sending data using light bulbs flashing imperceptibly fast. It has strong applications in improving security for IoT (Internet of Things) devices. This paper describes a hardware-first approach to building a visible light communication (VLC) link. A VLC link was designed by choosing the simplest possible circuit and software and then incrementally improving it as challenges such as ambient lighting noise and data rate limitations were encountered. This link was used with two main communication protocols: On-off keying (OOK), and Frequency-Shift Keying. The paper describes a design for a fast, robust system that allows for both protocols and an adjustable data rate. Because many issues were encountered along the way, the paper presents several possible sources of noise and data rate limitations and suggestions for how to remove them. Finally, the paper also describes extensions to the design to make it bidirectional, more robust, and faster.

Contents

1 Introduction

The recent increase in the use of LED lighting has prompted many in industry and academia to consider alternative use cases for LEDs. One major use case is Visible Light Communication. The core concept behind VLC is modulating a light source faster than human perception. This allows for information transmission without disrupting existing lighting levels. However, this type of communication link requires a line of sight and can easily be disrupted by noise from other light sources. This means that applications are usually limited to short range communications. For short range communications, however, VLC is both secure, because it is not susceptible to leakage through walls (a WiFi vulnerability), and provides an alternative communication pathway for Internet of Things devices.

This paper describes several VLC configurations, issues associated with using each, and takeaways from solving these issues. The configurations can be categorized by the type of optical hardware involved. The overall goal of this process was to work towards a stable hardware and software configuration that would work in any lighting scenario. This configuration could then be packaged into any use relevant case, such as a game where each transceiver would be used in a “wand” and the communication link would be used to send “spells” back and forth. This interactive game represents a proof of concept that shows the ability of the communication link to work at high bit rates, be limited to a specific direction, and allow for full duplex communication.

I took a hardware first approach to building each configuration. This means that I prioritized building the physical circuit and simply modified my software abstraction to include any drivers necessary for each circuit. The goal was to create the simplest configuration needed for a robust link. Each configuration contained a transmitter and a receiver circuit. The receiver circuits can be categorized into four groups based on the optical receiver part used: LED receiver, photo-resistor receiver, photo-transistor receiver, and photo-diode receiver. Each circuit included the main optical receiver part and any needed amplification, filtering, and conditioning circuitry. Additionally, the circuit used to transmit signals was always the same: one bright white LED and a resistor.

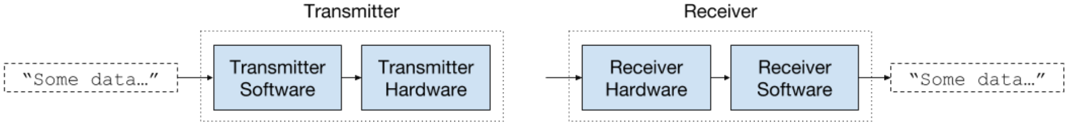


Figure 1: Overall Block Diagram

The VLC software consisted of two parts, driver software and protocol software. Driver software was used to directly interface with a specific hardware and convert the electrical signals into a readable format. The protocol software was used to transmit data and synchronize the link. In the block diagram (fig. 1), both are grouped into one block because they both interact very closely with the hardware and data.

The main contributions of this paper involve building a good design for a VLC link and providing solutions to two key VLC obstacles: ambient lighting noise and bit rate limitations. This paper presents a hardware receiver design that handles both obstacles, works with multiple protocols, and can handle high data rates. The main components of this design are an amplifier, a noise filter, band pass filters, and conditioning circuits. This paper also recommends three solutions to reducing ambient light interference: limit the distance of the link, ignore all light modulation frequencies except the transmitter modulation frequency, and use frequency shift keying. To reduce bit rate limitations, this paper shows that using photo-diodes, op amps with high gain bandwidth products, and hardware filters leads to the best results.

2 Existing Work

The increasing ubiquity of communications technology has rendered current Radio Frequency (RF) based devices insufficient to meet new bandwidth needs and speed concerns. Visible Light Communication (VLC) technology was pioneered to fill the gaps, because it can take existing light fixtures and repurpose them to serve as new points of communication. VLC is becoming increasingly important as a part of the fifth generation of wireless networks (5G) and many organizations have been created to research applications from LED-based localization to vehicular communication [1].

Modern VLC systems use LEDs for both lighting and communication [1] and therefore require that protocols be built around both needs. Given this consideration, the main mechanism of the transmitter portion in a VLC system is to flash or modulate an LED faster than a human can perceive [2]. To actually send data, many protocols have been developed that have this modulation at their core. In fact, the IEEE 802.15.7 standard has clearly defined three physical layer protocols and a medium access control layer protocol that satisfies these requirements [3].

2.1 Physical Layer Protocols

The focus of this paper is on the physical layer, because VLC is a physical technology created to improve upon current communications systems. In the Open Systems Interconnection (OSI) model, the physical layer is an abstraction that represents the hardware and software needed to transmit bit streams [4]. The overarching protocol for the physical layer defines an approach to using LEDs to transmit electrical signals and converting those signals to usable bit streams.

There are two main types of such protocols: time-domain based modulation and frequency or intensity-domain based modulation [2]. The first type of protocol uses sequences of two levels of brightness, ON and OFF, to encode symbols and maintain illumination. The second type of protocol varies light intensity and/or modulation frequency to accomplish these tasks.

2.1.1 Time Domain Protocols

The simplest time domain protocol is on-off keying (OOK). This protocol encodes a high voltage signal (ON) as a “one” symbol and a low voltage signal (OFF) as a “zero” symbol. To compensate for the lighting loss caused by sending many uninterrupted zeros, some have added a compensation pulse[1] or used a slightly more complex encoding like Manchester encoding [5].

Fig. 2 depicts the three time domain encoding methods mentioned: OOK, OOK with compensation, and Manchester encoding. OOK with compensation adds a high pulse after sending a data pulse to maintain the brightness of the light even if many low data pulses are sent. Manchester encoding uses a high-low pulse to represent a 1 and a low-high pulse to represent a 0. This means that 50% of the pulses are high, so illumination will be maintained. Although these more complex approaches preserve lighting levels, they require more of the hardware and software. For example, the transmitter modulation must either double in frequency or take twice as much time to transmit the same signal with both Manchester encoding and OOK with compensation.

To handle multiple transceivers, these time domain approaches have been used in combination techniques such as frequency domain multiplexing [2]. This allows for building larger routing protocols where devices transmit signals on unique frequencies.

2.1.2 Frequency and Intensity Domain Protocols

Frequency domain protocols such as frequency shift keying (FSK) represent symbols as specific modulation frequencies. Intensity domain protocols change the light intensity of the signal and often also use frequency keying to encode symbols [2].

Fig. 3 shows an example of simple frequency shift keying. Note that although the symbols

---

[1] A high pulse after every symbol [2]

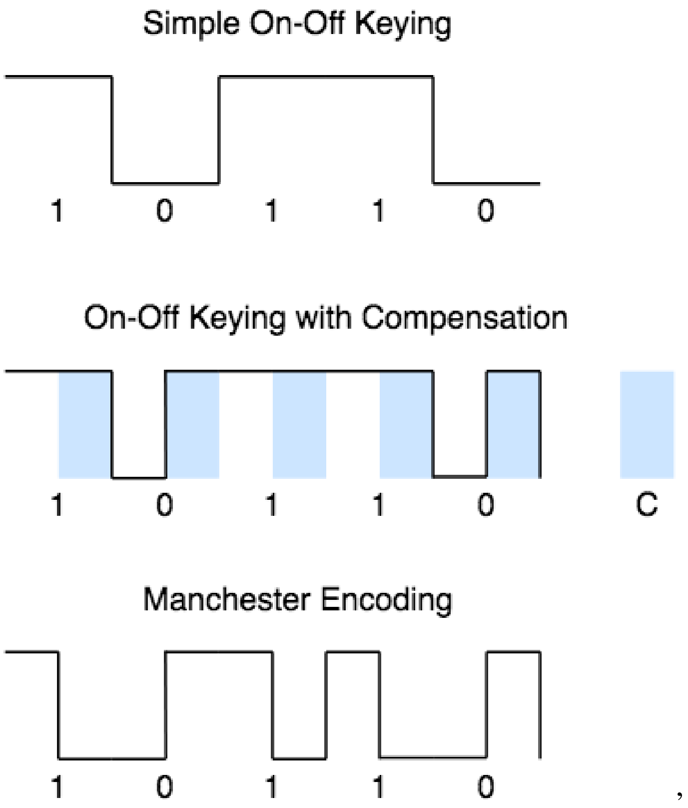


Figure 2: Time Domain Approaches

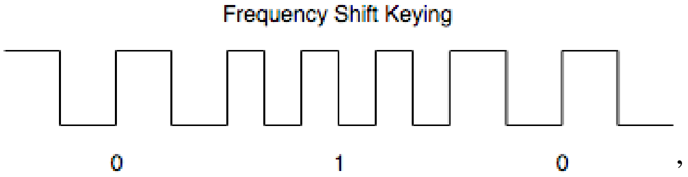


Figure 3: A Frequency Domain Protocol

are the same length, the digital signal used for each is of a different frequency. Additionally, there is no dimming concern because the light always maintains a constant duty cycle (percentage of pulses that are high). In this example, the light will be on 50% of the time.

However, a concern with frequency domain protocols is that they can require complex signal processing and limit data rates. This is because the data rate cannot exceed a multiple of the lowest frequency symbol. Additionally, filter hardware for identifying symbol frequencies is required. One research team used an audio jack to transmit signal information to a mobile device because of the high sampling rate requirement [6]. This example helps demonstrate the limitations of existing hardware in dealing with VLC technology.

An alternate transmission protocol is Pulse Width Modulation (PWM). This protocol is often used to power motors, and involves changing the duty cycle of a digital signal to encode information. A research team achieved a 960 bps data rate using this encoding to transmit information without affecting lighting [7]. However, given that PWM is often used to control dimming, this might cause issues with adjustable dimming in lighting. Additionally, PWM is also susceptible to the same noise and ambient lighting issues that On Off Keying is susceptible to.

3 Hardware Prototypes

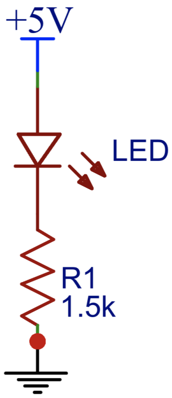
The main goal of this research was to work towards a visible light communication link that was both bidirectional and robust. I took a "hardware-first" approach, meaning I built a specific hardware configuration and then used software drivers to apply transmission and receiving protocols.

The hardware for a VLC configuration can be split into two parts: the transmitter and the receiver. The transmitter circuit receives a bit signal and uses a protocol to transmit said signal through an LED. The receiver then receives the analog signal and converts it into a readable format.

Throughout the hardware, I considered two main protocols: On-Off Keying (OOK) and Frequency Shift Keying (FSK). These protocols are easy to implement and easy to extend. They form the basis for many more complex protocols and are therefore good indicators of how more complex protocols would perform in worst case scenarios.

3.1 Transmitter Circuits

A transmitter circuit is required to read a bit stream and use a given protocol to transmit it via an LED. For both On-Off Keying and Frequency Shift Keying, the main requirement was to modulate the LED at a specific frequency and with a specific pattern. Because both of these parameters can be controlled from software, I used a basic LED and resistor pair for the transmitter circuit. Additionally, the goal of this paper was to create a VLC link with a single LED to test how VLC works in embedded devices, so an amplifier was not required to increase the light intensity of the LED.



,

Figure 4: LED Transmitter Circuit

3.2 Receiver Circuit

A receiver circuit is required to receive an optical signal and convert it to a digital signal. This means that the most important component of this circuit is the optical receiver part. Several electrical components can fulfill this role, but I focused on four possible components: the LED itself, the photo-resistor, the photo-transistor, and the photo-diode. I eventually decided that the photo-diode was the most robust part because it can respond very quickly to changes in light and can be used in many different types of circuits.

3.2.1 LED Receiver

The LED receiver uses certain properties of an LED to make it act as a weak optical receiver instead of a light source. This was attractive because it meant that the transmitter, if set up correctly could also be used as a receiver [8].

A light emitting diode or an LED is an electrical component that can serve as a light source. Like any other diode, an LED has a "p-n junction". This means that the "p" side of the diode has fewer electrons and is positively charged, and the "n" side of the diode has more electrons and is negatively charged. These properties are artificially applied by choosing certain metals and using a process called "doping" when creating the LED. When an electrical current is applied to an LED, electrons jump from the "n" side to the "p" side and emit energy in the form of light. Fig. 5 illustrates this.

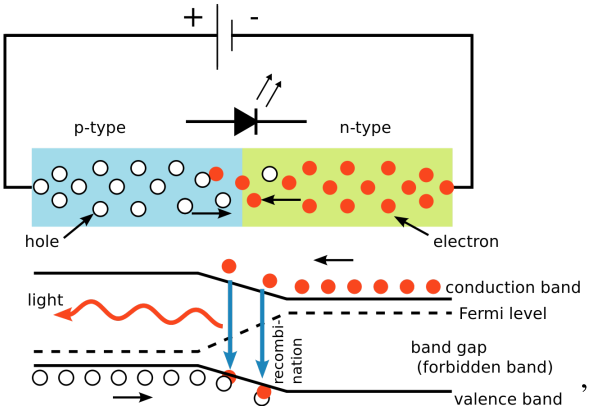
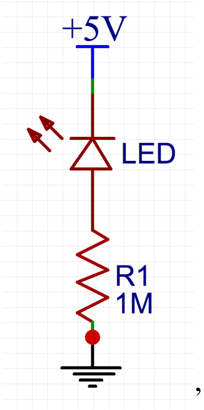


Figure 5: P-N Junction

These properties also make the LED useful as a receiver or "photo-detector". This is because, when the LED is flipped, or put in "reverse bias", the opposite happens. Light energy is absorbed by the LED and causes current to flow from the p-n junction. This current is called a “photo-current” and can be used to measure fluctuations in light intensity.



,

Figure 6: Bad LED Receiver Circuit

A simple circuit (fig. 6) to read light fluctuations would be an LED in reverse bias with a resistor. However, there are two problems with this type of circuit. First, LEDs output very low current [9], so a very large resistor would be required (V = I*R), which can increase noise (e.g. Johnson-Nyquist noise). Second, this is a “passive” circuit, so reading this voltage with a micro-controller or using this circuit as part of a larger circuit would cause unpredictable results. This is any circuit used to read the voltage would add resistance, often changing, so the voltage value would be unpredictable.

Therefore, I used an “active” amplifier circuit (fig. 7) to convert the LED photo-current into a readable voltage. However, the circuit still required a very large resistor and the LED was not very sensitive to changes in light or very fast [10]. I concluded that a better solution would be to use a part specifically made for acting as a light detector.

3.2.2 Photo-resistor Receiver

A photo-resistor or photocell is a very simple resistor built with a high resistance semiconductor. The semiconductor material responds to light by decreasing its resistance. Additionally, building a receiver circuit with a photo-resistor is very simple (see fig. 8).

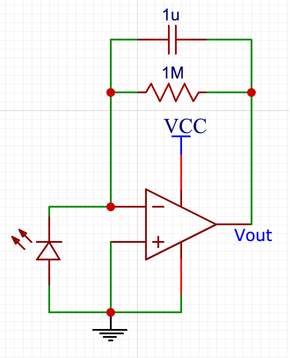


Figure 7: Active LED Amplifier Circuit

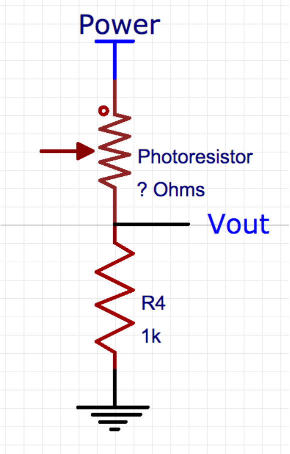


Figure 8: Photoresistor Circuit

This circuit is very predictable with room temperature conditions. Additionally, the photo-resistor is cheap and uses very little power. However, the main drawback is switching speed, or how fast the circuit can respond to changes in light. The photo-resistor was very slow to respond (on the order of ms) when an LED was flashed on and off about a meter away. The circuit responded more quickly when there was low ambient lighting, but these conditions are not realistic. This made the circuit good for testing basic software transmission but not feasible for a useful data link.

3.2.3 Photo-transistor Receiver

A photo-transistor is a modified version of another semiconductor component, the bipolar junction transistor. The difference between this component and diode is that there can be two “p” sections (PNP) or two “n” junctions (NPN). This means that there are three pins: base (middle), collector, and emitter. This allows for many configurations, but the relevant configuration I used is pictured in fig. 9.

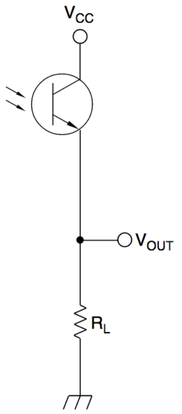


Figure 9: Photo-transistor Circuit

In this configuration, light enters the base junction of the photo-transistor and causes current to flow. This is because the energy of the photons frees electrons and causes them to flow towards higher voltage. This causes current to flow from the higher voltage collector (the junction connected to the Vcc node) through the emitter. Then, voltage can be measured across the resistor.

This circuit was moderately fast and reliable, requiring little setup or amplification and still receiving the digital signal without much distortion. However, the maximum switching speed of the photo-transistor I used was about four microseconds. This limits the highest frequency signal that could be transmitted to a 250KHz sine wave. Accounting for time required to transmit information, the data rate will be much lower. Although these data rates would be sufficient for a moderate speed data link, the hard limitations were too constraining to accommodate every use case.

3.2.4 Photo-diode Receiver

A photo-diode is very similar to an LED in reverse bias, but it is a component specifically created for use as a photo-detector. This means that it is more sensitive than an LED and can receive all visible light frequencies. Additionally, photo-diodes are a much faster component than photo-transistors. I used a modified photo-diode known as a PIN diode. This type of diode includes a semiconductor metal between the "p" and "n" regions, which helps increase the size of the electric field caused by applying a voltage and improves switching.

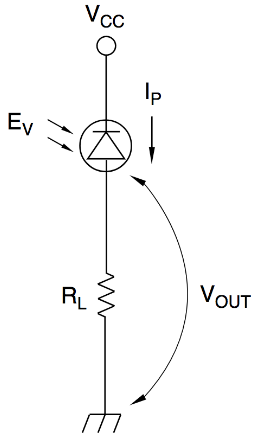


Figure 10: Basic Photo-diode Circuit

Fig. 10 shows a basic photo-diode configuration. This configuration is the standard configuration used earlier to test the other optical receivers. The photo-diode circuit turned out to be very fast at switching and sensitive enough for the data link. However, like the other receivers, the amount of current generated was very low (on the order of micro-amps), so an amplifier circuit was needed.

There are other base photo-diode configurations as well. The two main configurations or "modes" the photo-diode can be put in are the "photo-conductive" mode and the "photo-voltaic" mode. The photo-voltaic mode is when the photo-diode is connected to zero voltage ("zero bias") and the photo-voltaic effect generates a voltage across the diode. This effect occurs when light excites electrons and causes them to move to higher energy levels. The photo-voltaic mode is often used when the photo-diode is acting as a solar cell. The photo-conductive mode is very similar, but the voltage the diode is connected to helps increase the size of the region where electrons have moved from the "n" section to the "p" section. This increases the switching speed of the diode. Therefore, the circuit I used took advantage of the photo-conductive mode.

3.2.5 Photo-diode Receiver with Amplifier

The next step to improving the photo-diode circuit was to build an amplifier. A simple model of the photo-diode is a current source that depends on the amount of light it is exposed to at any instant. A circuit that converts such a current signal to a readable voltage signal is called a trans-impedance amplifier. In general, it is easiest to implement this circuit with an operational amplifier and a feedback resistor.

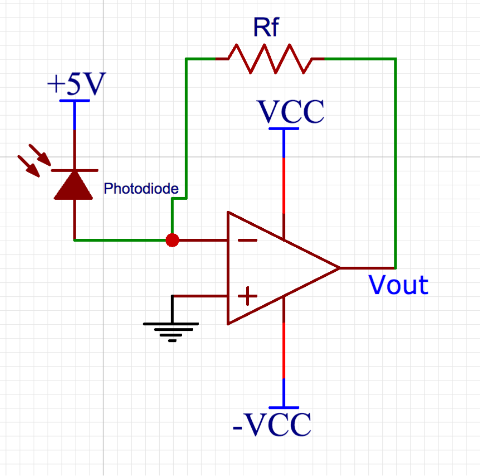


Figure 11: One Stage Trans-impedance Amplifier

Fig. 11 shows this circuit configuration. However, given the very low current output of a photo-diode, a very large resistor would be needed to make the voltage readable by a computer. However, at very high gain, noise concerns and oscillation (changes in the output caused by the op amp) become an issue. Therefore, I added a second stage to the amplifier.

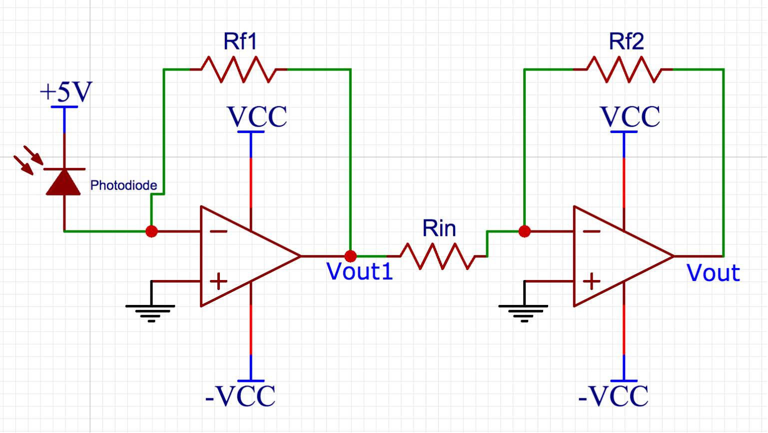


Figure 12: Two Stage Trans-impedance Amplifier

This circuit then works as an amplifier that converts the photo-diode current to a readable voltage. As an example, if the photo-diode current on average is 50 μA, and we set Rf1 = 5k Ohms, Rin = 1k Ohms, and Rf2 = 20k Ohms, then the first stage will output a 250 mV signal and the second stage will output a 5 V signal.

3.2.6 Receiver Circuit Filtering and Conditioning

Although the photo-diode amplifier circuit successfully converts the photo-diode current to a reasonable voltage, most micro-controllers can only handle a voltage range from 0V to 3.3V. To handle this concern, a simple solution is to tune the resistors of the amplifier such that the highest output voltage would be 3.3 V and use a voltage buffer (an op amp with a gain of 1) with a positive power source of +3.3 V to clip off voltages higher than 3.3 V.

Another issue is converting the signal to a digital value. One solution is to use an Analog to Digital Converter (which is described in the software section). An alternative solution is to reduce the level of detail in the circuit output by using a comparator. A comparator compares

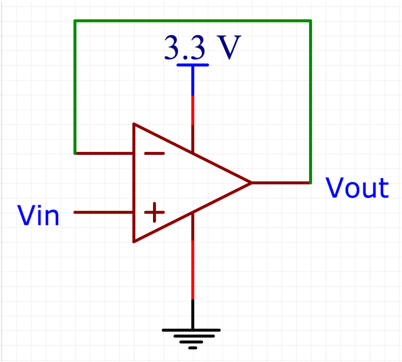


Figure 13: Voltage Buffer

the input voltage to a given voltage and outputs 3.3 V if the input voltage is higher and 0 V if the input voltage is lower (these voltages are determined by the power sources to the op amp). This means that the output of the circuit could be used directly with a digital input pin.

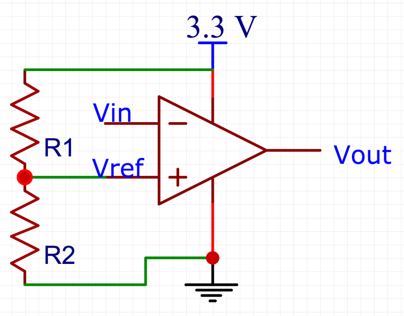


Figure 14: Comparator

Currently, the amplifier and comparator circuit work well for a time domain encoding method. However, to shift some of software burden of implementing frequency encoding to the hardware, I used an active filter. An active filter is an op amp circuit similar to an amplifier, but using capacitors to change how the circuit responds with different frequency signals. For a frequency shift keying (FSK) circuit, each symbol needs a band pass filter. This filter only allows a signal of a specific frequency to pass through the circuit. This means that if the LED is modulated at a certain frequency, then the received signal will pass through the associated band pass filter circuit.

Additionally, to make the signal easily readable by the digital input on a micro-controller, I decided to use a rectifier. A rectifier circuit turns a waveform signal into a DC or a single voltage signal by using a specific diode arrangement or "topology". Adding this circuit at the end of each band pass filter means that the output of each rectifier will be a high voltage if that particular symbol was detected and a low voltage if the symbol was not detected.

Finally, to further reduce the burden of the micro-controller to decode what symbol was received, I can also use digital logic. As a simple example, I have provided the digital circuit for an FSK decoder with two symbols, '0' and '1'. The logic below uses a multiplexer to select either the '1' symbol if that symbol was received or the other symbol. Then the XOR gate is used to ensure that only one symbol is received at a time. The micro-controller can then detect which symbol was sent and whether the sent symbol was valid.

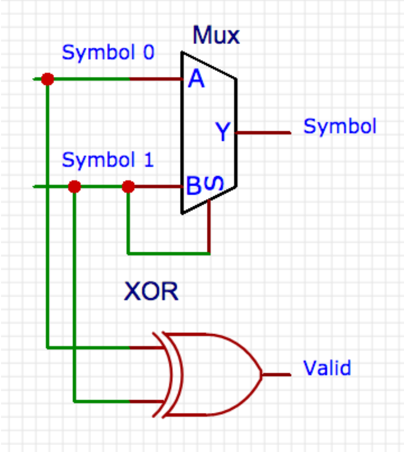


Figure 15: Digital Logic for Two Symbols

3.3 Hardware Overview

In this section, I have provided two block diagrams that review the transmitter and receiver hardware. The transmitter hardware is simple, an LED and resistor connected to the output pin of a micro-controller. The receiver hardware is much more complex, but ultimately creates a simple interface with the micro-controller.

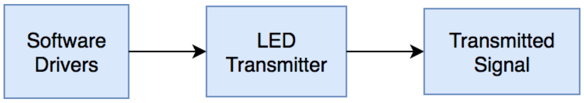


Figure 16: Transmitter Block Diagram

The transmitter block diagram could become more complex if multiple LEDs are used or a different encoding scheme such as color shift keying is used. This paper omits these for simplicity.

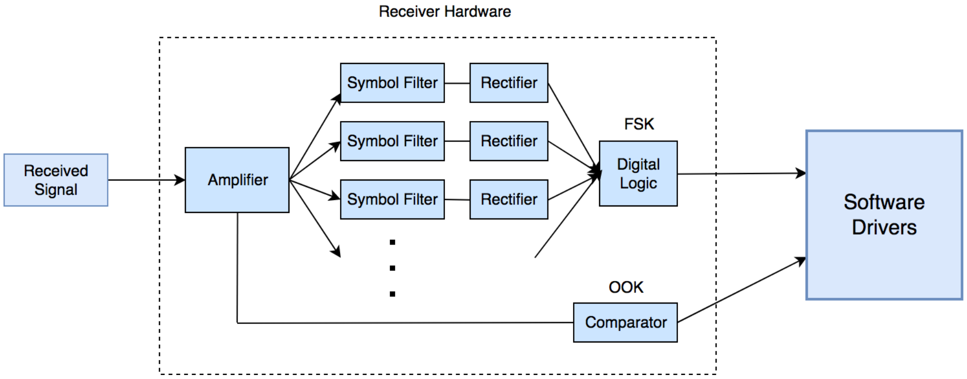


Figure 17: Receiver Block Diagram

The receiver block diagram shows that both on-off keying (OOK) and frequency shift keying are accommodated by the hardware circuit. Although it is omitted, more filtering could be used in the OOK circuit to remove ambient light noise, etc.

4 Software Protocol Design

Although the hardware system successfully transmits a signal, receives a signal from a transmitter, and outputs a indicator of which symbol was received, it does not actually interpret the signal or create a working communication link. This is the responsibility of the software system.

As shown in the block diagrams (fig. 16 and fig. 17), each hardware circuit requires software drivers. These drivers accomplish two main goals: encoding and decoding the signal and syncing the transmitter and receiver.

Signal encoding and decoding for OOK and FSK at the bare minimum involved implementing the parts of the protocol that were in-feasible to accomplish in hardware. For OOK, this simply meant either reading the 1s and 0s and then sending the data to a function that used a given decoding procedure to convert the signal into ASCII or some other format (or vice versa for encoding). For FSK, the process was similar, because the hardware eventually output a 1 or a 0 when the appropriate symbol was received. With intermediate hardware stages, however, software was used to approximate the hardware processing.

Transmitter and Receiver syncing is a difficult problem, but in this paper we use a simple protocol to set up a communication link. This protocol involves sending a certain bit sequence to indicate the beginning of a transmission, meta information, then the transmission data.

4.1 Simple Software Protocol

As previously mentioned, the software protocol used had two components: a syncing protocol and a decoding protocol.

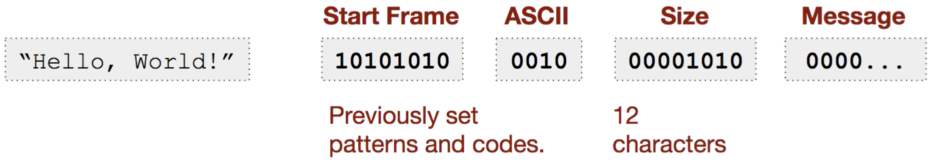


Figure 18: Simple Software Protocol Example

The above example (fig. 18), shows the encoding for transmitting the string “Hello, World!”. The start frame is a byte of a specific pattern of 1s and 0s agreed upon beforehand that signals the receiver that a message is about to be transmitted. Starting with a 1 makes the signal easier to detect when using on-off keying and the light is initially off. The sequence can be flipped if the light is initially on, so that it is again easy to detect. Alternatively, the receiver can simply reject a transmission with an incomplete start frame and wait for a re-transmission.

The next part of the encoding is a four bit sequence that indicates the encoding of the data, in this case ASCII. The four bit sequence is another code that is agreed upon beforehand as part of the protocol. After this, the size of the encoding is sent, so the receiver know how many bytes to read. Finally, the data is sent in bytes.

This simple protocol works for primitive data transfer in a two device transmitter-receiver set up and works well for testing. The protocol was used successfully for testing a one way communication link.

4.1.1 Protocol Extensions

To extend the protocol for a bidirectional link, I propose two methods. The first method is for both devices to use different carrier frequencies to send information. For the OOK protocol, this can easily be implemented by adding band pass filters to the receivers. For FSK, this can be implemented by choosing different frequencies for the symbols being sent by the LEDs. The second method is to use time slots for the two devices. This protocol would be much more complicated, but could be implemented entirely in software without significant hardware changes. However, this method, unlike the first method is not full duplex.

The carrier frequency method is commonly used for other communication protocols such as in Orthogonal Frequency Division Multiplexing or OFDM. A carrier frequency in the context of this paper is the base frequency at which symbols are transmitted. For OOK, the carrier frequency is the frequency of the digital signal. For FSK, the carrier frequency is the frequency of one of the symbols, with the other symbol frequencies being some real number times the carrier frequency. OFDM assigns different non-overlapping carrier frequencies and bands to different signals. This technique could be applied to each device attempting to send a signal, and like an FM radio, a receiver on one device could be tuned to a specific carrier frequency based on the device it desires to communicate with.

The time slot protocol is more complicated and not full duplex, but as Schmid et al. [13] showed, can be implemented mostly in software. Schmid et al. used time slots for illumination and communication. Each illumination slot accounted for the needed brightness level of the light and the communication slot used time slots for each device. Each slot alternated and was a fixed length of 1ms, which meant that the bit rate could easily be calculated based on the modulation speed of the transmitter. A variation of that method could be implemented for the OOK protocol and the FSK protocol talked about in this paper.

4.2 Software Alternatives to Hardware

The only necessary component in the hardware receiver and transmitter were the LEDs and photo-diodes. Each other component: the amplifier, the filters, and the signal conditioning components had software alternatives that I explored and attempted to use. As mentioned previously, the software alternative for each had some merits, such as being generally low cost, but also some major disadvantages, such as performing significantly worse than their hardware counterparts, occasionally being more expensive, and often being much slower.

For both the OOK and FSK protocols, a trans impedance amplifier was needed to convert the photo-diode signal to a readable value. However, for the photo-diode and photo-transistor, the values were actually readable by the analog-to-digital converter (ADC) on a micro-controller. The software alternative I tried then was to read values from the ADC and then use software to detect the changes in amplitude and assign 0s or 1s to values above or below certain thresholds. However, this method was highly susceptible to ambient lighting conditions because the threshold had to be recomputed each time these conditions changed. Additionally, the hard speed limitation was the speed of the ADC conversion, which on most micro-controllers can take up to 1ms. Given the complexity of the software required to account for changing light conditions and other light noise, this meant that the actual data rate was much lower than the ADC conversion speed. This was not an acceptable setup for creating a flexible link, so I decided that using a hardware amplifier would be best.

After adding the amplifier, it was much easier to compute thresholds for on-off keying, but the circuit was still heavily speed limited because of the ADC and the software. Therefore, I then added the comparator, the signal conditioning component needed to convert the analog signal into a digital signal by using a hardware threshold comparison. This certainly reduced the complexity of the software and now created a digital signal that could be read at the clock speed of the micro-controller, which was much less limiting condition. However, the circuit no longer adapted to changing lighting conditions, which greatly increased the amount of error in the transmission.

Therefore, the last signal conditioning component need for the circuit (for the OOK protocol) was a high pass filter that removed the noise from the receiver. This meant that the threshold no longer had to be adaptive and the software could reliably expect a digital signal and only had to implement the protocol for converting the signal into information.

Although these components also significantly increased the robustness of the FSK setup, the software still had to detect whether a certain light modulation frequency was sent by the transmitter. To do this, I used the Fast Fourier Transform (FFT) algorithm to compute the frequency spectrum of the signal and then identified the dominant frequency based on a signal power threshold. However, there were three problems with this approach: the FFT is an O(nlogn) algorithm, which means it would take a lot of software processing to read one symbol, the FFT's accuracy reduces greatly when using a small number of samples, and using a signal power threshold can be very inaccurate when the signal-to-noise ratio is low. These problems meant that the FFT added another bottleneck[2] to the data rate and introduced new sources of noise. Given these disadvantages, I decided that using a band pass filter and rectifier would the best approach, despite the relative complexity of that circuit in comparison to the circuits.

5 Conclusion and Suggestions for Future Work

The hardware designs for both an OOK protocol and an FSK protocol create a visible light communication link that effectively modulates and demodulates a signal, such that the software simply outputs and receives a digital signal. The amplifier component make the signal readable, the high pass filter removes ambient light noise, the comparator and rectifier condition the signal for the micro-controller specifications, and the band pass filter effectively identifies frequency symbols. These hardware components also allow for data rates up to the MB/s, depending on the integrated circuits used.

The software protocol for testing is a simple protocol that demonstrates how well the physical layer can be abstracted so that any information encoding and decoding protocol can be used. Finally, the software alternative analysis shows that each hardware component is necessary to implement a robust and fast physical layer.

In the future, I recommend that this design be extended to include techniques such as frequency division multiplexing, which allow communication between multiple devices. Additionally, to increase bandwidth, the design should also be extended with multiple light colors and color filters. Finally, the frequency shift keying approach should also be replicated with a digital signal processing chip, because although this complicates the hardware, it has the potential to greatly increase bandwidth even in a set up with low speed hardware.

---

[2] A Digital Signal Processing Chip can be used to speed up the processing, but this greatly complicates the software.

References and Notes

1. R. Zhang, “Localisation, Communication and Networking with VLC: Challenges and Opportunities,” arXiv:1709.01899 [cs.IT], Sep. 2017.

2. S. H. Lee, S. Y. Jung and J. K. Kwon, “Modulation and coding for dimmable visible light communication,” in IEEE Communications Magazine, vol. 53, no. 2, pp. 136-143, Feb. 2015.

3. “IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light,” in IEEE Std 802.15.7-2011 , vol., no., pp.1-309, Sept. 6 2011

4. H. Zimmermann, “OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection,” in IEEE Transactions on Communications, vol. 28, no. 4, pp. 425-432, April 1980.

5. N. Rajagopal, P. Lazik, and A. Rowe. “Hybrid visible light communication for cameras and low-power embedded devices,” In Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems (VLCS ’14), 2014, pp. 33-38.

6. Y. Ashida, K. Okuda and W. Uemura, “A VLC receiving devise using audio jacks with a folding noise,” 2012 International Symposium on Information Theory and its Applications, Honolulu, HI, 2012, pp. 475-479.

7. A. Pradana, N. Ahmadi and T. Adionos, “Design and implementation of visible light communication system using pulse width modulation,” 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, 2015, pp. 25-30.

8. S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. “LED-to-LED visible light communication networks,” In Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing (MobiHoc ’13). New York, 2013, pp. 1-10.
9. Kingbright, “Solid State Lamp,” WP7113SRD/D datasheet [Revised May 2007].
10. P. Dietz, W. Yerazunis, D. Leigh “Very Low-Cost Sensing and Communication Using Bidirectional LEDs,” In UbiComp 2003: Ubiquitous Computing, 2003, pp. 175-191.
11. Vishay Semiconductors, “Silicon NPN Phototransistor ,” BPW85 datasheet, [Revised Aug. 2014].
12. Sharp, “Photodiode/Phototransistor Application Circuit,” Optoelectronics Application Note, 1999.
13. S. Schmid, B. Deschwanden, S. Mangold, and T. R. Gross. “Adaptive Software-Defined Visible Light Communication Networks,” In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation (IoTDI ’17), New York, pp. 109-120.