

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Shivank Sharma IVEM156425

DESIGN & IMPLEMENTATION OF VISIBLE LIGHT COMMUNICATION SYSTEM

Master's Thesis

Supervisor: Dr. Paul Annus

PhD
Senior Research
Scientist

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Shivank Sharma IVEM156425

**NÄHTAVA VALGUSE
KOMMUNIKATSIOONI SÜSTEEMI DISAIN
JA RAKENDAMINE**

Magistritöö

Juhendaja: Dr. Paul Annus

PhD
Vanemteadur

Tallinn 2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Shivank Sharma

06.05.2017

Abstract

Wireless communication using visible light is an exciting prospect which has started gaining a lot of attention recently. This is because this branch of communication has several advantages over using RF, such as the fact that the visible light spectrum is still unlicensed, which can solve the problem of present status of congested RF spectrum, secondly the infrastructure already exists, all that needs to be done is to replace the current light bulbs with intelligent and efficient bulbs capable of data transmission.

A lot of research and development is being done in this hot field of wireless communication, but most of which is focused on increasing its throughput/data rate and very less has been concentrated on utilizing it for short range indoor applications such as using it as a Wireless Sensor Node.

This thesis work explores the designing and easy implementation of a low cost visible light communication system through different prototype implementations and evaluating its applicability as a Wireless Sensor Node in terms of scalability.

Two different techniques were implemented in the prototypes. One being On Off Keying Modulation without encoding and other being On Off Keying Modulation with Manchester encoding. Both of them were implemented on software level and algorithms for each of them are clearly explained in this thesis. Furthermore, to see the scalability of the system, an additional transceiver node was added and was evaluated for both of the above-mentioned methods. The prototypes were designed to sense the ambient temperature readings and transmit them via visible light.

In closing, the thesis presents suggestions for further work to increase system performance and capabilities.

Annotatsioon

Nähtava valguse kommunikatsiooni süsteemi disain ja rakendamine

Nähtavat valgust saab kasutada juhtmevabaks kommunikatsiooniks. See väljavaade on hiljuti kogunud rohkem tähelepanu. See on sellepärast, et sellel kommunikatsiooni harul on mitmeid eeliseid võrreldes RFga, näiteks nähtav valgus on ikka veel litsenseerimata, mis võib lahendada umbes RF spektri side kanalite mahutatavuse. Teiseks on taristu selleks juba olemas ning selleks oleks vaja lihtsalt välja vahetada praegused elektripirnid tarkade pirnidega, mis on võimelised infot edastama.

Juhtmevaba kommunikatsiooni uuritakse ja arendatakse väga palju, aga enamus sellest keskendub andmemahu läbilaskevõime suurendamisele. Väga vähe keskendatakse väikese vahemaa siseruumide rakendustele nagu kasutamaks seda juhtmevaba andurina.

Lõputöös käsitletakse erinevate nähtava valguse kommunikatsiooni süsteemide disainimist ja kerget rakendamist prototüüpides ja hinnates seda juhtmevaba andurina võrgu skaleeritavuses.

Prototüüpides rakendati kahte erinevat meetodikat. Üks meetoodika oli sisse välja lülitamis modulatsioon kodeerimiseta ning üks oli sisse välja lülitamise modulatsioon, koos manchester kodeerimisega. Mõlemaid teostati tarkvara tasandil ja aloritmid lahenduseks on selgesti lahti seletatud. Nägemaks süsteemi skaleeritavust lisati üks saatja süsteemi juurde ning hinnati mõlemat meetodit, mis eespüool kirjeldati. Prototüüpe disainiti lugemaks toatemperatuuri ning edastamiseks seda kasutades nähtavat valgust.

Lõpeduseks kirjeldati olemasolevaid soovitusi, et parandada süsteemi töökindlust ja võimekust.

List of abbreviations and terms

ADC	<i>Analog to Digital Converter</i>
C _j	<i>Junction Capacitance</i>
COM	<i>Communication Port</i>
FM	<i>Frequency Modulation</i>
ID	<i>Identity</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IM/DD	<i>Intensity Modulation/Direct Detection</i>
IP	<i>Internet Protocol</i>
LED	<i>Light Emitting Diode</i>
LSB	<i>Least Significant Bit</i>
MSB	<i>Most Significant Bit</i>
OC	<i>Optical Communication</i>
OOK	<i>On Off Keying</i>
OWC	<i>Optical Wireless Communication</i>
PC	<i>Personal Computer</i>
PPM	<i>Pulse Position Modulation</i>
PWM	<i>Pulse Width Modulation</i>
R	<i>Resistance</i>
RF	<i>Radio Frequency</i>
RX	<i>Receiver</i>
TTU	<i>Tallinn University of Technology</i>
TX	<i>Transmitter</i>
USB	<i>Universal Serial Bus</i>
V2V	<i>Vehicle to Vehicle</i>
VLC	<i>Visible Light Communication</i>
VPPM	<i>Variable Pulse Position Modulation</i>
Wi-Fi	<i>Wireless Fidelity</i>
WSN	<i>Wireless Sensor Node/Wireless Sensor Network</i>

Table of contents

Author's declaration of originality	3
Abstract.....	4
Annotatsioon Nähtava valguse kommunikatsiooni süsteemi disain ja rakendamine	5
List of abbreviations and terms	6
Table of contents	7
List of figures	9
1 INTRODUCTION	11
1.1 Motivation	11
1.2 Problem Statement.....	12
1.3 Objectives and Goals	13
1.3.1 Primary Objectives	13
1.4 Significance of the work.....	14
1.5 Background of Visible Light Communication	14
2 BACKGROUND THEORY	15
2.1 Light Emitting Diode (LED)	15
2.2 Photodiodes	16
2.2.1 PiN Photodiode.....	16
2.3 Operation Modes of Photodiode.....	17
2.3.1 PHOTOVOLTAIC MODE	17
2.3.2 PHOTCONDUCTIVE MODE	17
2.4 Arduino UNO	17
2.5 ON-OFF KEYING (OOK)	18
2.6 Manchester Coding.....	19
2.7 Wireless Sensor Network (WSN) and Visible Light Communication (VLC)	19
3 LITERATURE SURVEY OF VISIBLE LIGHT COMMUNICATION AND WIRELESS SENSOR NETWORKS	20
3.1 VISIBLE LIGHT COMMUNICATION	20

3.1.1	VISIBLE LIGHT COMMUNICATION FOR INTELLIGENT TRANSPORT SYSTEM(ITIS)	20
3.1.2	VLC SYSTEM FOR DEEP SEA COMMUNICATION	21
3.1.3	FRAUNHOFER INSTITUTE OF TELECOMMUNICATION	22
4	DESIGN REQUIREMENTS.....	23
4.1	Development Platform.....	23
4.2	Hardware	23
4.2.1	Transmitter	23
4.2.2	Receiver	24
4.3	Software.....	24
5	DESIGN AND IMPLEMENTATION OF VLC SYSTEM.....	26
5.1	Design of Prototype 1 (On Off Keying Modulation)	26
5.1.1	TRANSMITTER	27
5.1.2	RECEIVER.....	30
5.1.3	Transceiver	33
5.2	Design of Prototype 2 (On Off Keying Modulation with Manchester Coding) ...	34
5.2.1	Manchester Encoding	34
5.2.2	Manchester Decoding	36
6	EXPERIMENTAL SETUP AND RESULTS	38
6.1	Two Nodes with and without Manchester coding.....	39
6.2	Three Nodes with and without Manchester coding	42
7	DISCUSSION AND SUMMARY	45
7.1	General Discussion and Conclusion	45
7.2	Scope of improvement.....	46
8	References	48
	Appendix 1 – Source Code.....	50
	Appendix 2- Bill of Material	65

List of figures

Figure 1 Shows the Frequency allocation in United States as of Jan. 2016 [30]	12
Figure 2 Shows the electron-hole combination principle for LED electroluminescence	15
Figure 3 Shows the PiN photodiode and the movement of minority carriers resulting in photocurrent generation [8]	16
Figure 4 Shows the technical specification of Arduino Uno prototyping board [10]	18
Figure 5 Shows the comparison of inverse 4-ppm scheme and VPPM scheme [12].....	20
Figure 6 Shows the vehicle to vehicle (V2V) VLC demonstration [12].....	21
Figure 7 Shows the transmitter block diagram for the deep-sea communication system [14]	22
Figure 8 Shows the receiver block diagram for the deep-sea communication system [14]	22
Figure 9 Shows the prototypes from left: Transmitter, Transceiver and a Receiver.....	26
Figure 10 Shows the block diagram of Prototype 1	26
Figure 11 Shows the schematic of Transmitter for prototype 1	28
Figure 12 Shows the simulation for the transmitter circuit using Autodesk Circuits	28
Figure 13 Shows the algorithm followed for the transmission of 8-bit data	29
Figure 14 Shows the schematic of the Receiver module.....	30
Figure 15 Shows the algorithm for the receiver module	31
Figure 16 Shows the algorithm used for Receiver calibration and Threshold calculation	32
Figure 17 Shows the layout of the Prototype 1.2	33
Figure 18 Shows the schematic for the Transceiver circuit.....	34
Figure 19 Shows the encoding of bits sing Manchester coding	35
Figure 20 Shows the algorithm for implementing Manchester encoding on the microcontroller	36
Figure 21 Shows the algorithm for implementing Manchester Decoding on the microcontroller	37
Figure 22 Shows the transmitted and Received data on the terminal.....	38

Figure 23 Shows the graph for Data received (bits per second) VS Distance for VLC system without Manchester Coding	39
Figure 24 Shows the graph for Data received (bits per second) VS Distance for VLC system without Manchester Coding	40
Figure 25 Shows the Data Loss Vs Data Rate graph for VLC system without Manchester encoding.....	40
Figure 26 Shows the Data Loss Vs Data Rate graph for VLC system with Manchester encoding	41
Figure 27 Shows the outputs of Tx and Rx on the oscilloscope for the system without Manchester Coding.....	41
Figure 28 Shows the outputs of Tx and Rx on the oscilloscope for VLC system with Manchester Coding.....	42
Figure 29 Shows the experimental setup for the three Nodes	42
Figure 30 Shows the graph for Data Loss Vs Data Rate for system without Manchester coding	43
Figure 31 Shows the graph for Data Loss Vs Data Rate for system with Manchester coding	43
Figure 32 Shows the output at each TX and RX in the three nodes.....	44
Figure 33 Shows the Output at all TX and RX on the three nodes in prototype 2. The left and right images represent different interval of time.	44
Figure 34 Shows the schematic mechanism of VPPM dimming as described in the IEEE standard [11].....	47

1 INTRODUCTION

1.1 Motivation

Since Wi-Fi's accidental invention in 1992 by an Australian astronomer through an unsuccessful experiment to observe mini black holes Wi-Fi has revolutionized the digital communication.

Wi-Fi presently contributes to 60% of the web traffic globally. However, Wi-Fi still has some problems like it is not entirely secure because these signals can travel through walls and be simply picked up by some other person, it uses radio waves that is restricted, expensive and there's a certain limitation.

So how do we improve this technology and solve these problems? We've to use something that's low cost, safe, in abundance and sturdier for data transfer. The solution is light; light is ideal because it's a part of the electromagnetic spectrum just like radio waves however at a much higher frequency. The frequency range of light is ten thousand times bigger than radio waves i.e. 750 terahertz as compared to a maximum of 300 GHz for radio waves, which suggests that light is capable to transmit more pulses of data in much less time as compared to radio waves. The spectrum of light is 10,000 times bigger than the spectrum of Radio waves and this is still unutilized and can be used.

Figure 1 shows the present radio frequency allocation in United States of America as of year, 2016, where we can clearly see the overcrowded spectrum. And according to Cisco's Global IP Traffic forecast the number of networked devices are going to be increased exponentially in coming years.

1.3 Objectives and Goals

The VLC technology uses LED as transmitter and photodiode as a receiver for communication. It is a sub-branch of Optical Wireless Communication (OWC) which includes infrared, visible light and ultraviolet waves. With the already available infrastructure for illumination additional services of data transmission can be provided.

Visible light communication (VLC) discourses the limitations of RF communication. Optical communication (OC) systems are less power hungry, smaller, lighter and do not interfere with RF signals [1].

This work will give a clear overview and understanding on design and implementation of a VLC system and will also act as a great starting point for anyone who wants to continue further Research and Development in this field of optical wireless communication.

1.3.1 Primary Objectives

The primary objective is to design and implement a VLC system which should be capable of:

- Sending data via visible light.
- Receiving the data over visible light.
- Sending and receiving the data over at least 1 meter range.
- Can be scalable via adding additional nodes.
- Should be easy to implement and low cost.
- Able to work in different indoor ambient light situations.

1.4 Significance of the work

Wireless sensor networks (WSNs) has grown tremendously due to the great variety and capability of their modern and real-world applications in areas like smart home applications, environment applications, traffic management, tracking object and e-health [2]. Typically, sensor nodes use the RF technology for transmission, which has some issues. The fact the light is powered and unregulated is the key for enabling this new technology to support the existing RF infrastructure in driving the future communication needs.

1.5 Background of Visible Light Communication

Use of light as a medium of communication is known to mankind for many years. Starting from the use of fire as signal and invention of the electric light bulb by Thomas Edison in the 19th century [3] which then led to the invention of the Signal Lamp, which is an optical communication system invented by Arthur C. W. Aldis used for visual signalling [4].

The first demonstration of transmission of information through visible light communication was performed by Alexander Graham Bell in year 1880 with his invention of Photophone. This device was used to transmit voice signal over beam of light. It was based on working principle of modulating the sunlight focused on mirror by vibrations and the vibrating beam was picked up by the detector at the receiving end and the voice signal was obtained after decoding, similar to the mechanism as the phone did with electrical signals. [5]

The invention of LED (Light Emitting Diode) has again paved in the way for new opportunities in the area of visible light communication. It works on the principal of Morse Code i.e. flashing the light on and off. But LED being semiconductor has a property that its frequency can be modulated rapidly at such a high speed that the turning On and OFF is undetectable by the human eyes. This OF and OFF are then interpreted by the photodiode at the receiving end as Logic 1 and logic 0.

2 BACKGROUND THEORY

This chapter will provide some background information about certain topics which are needed for this thesis, i.e. LED (Light Emitting Diodes), Photodiodes, Arduino Uno Development Board, OOK (On Off Keying) and Manchester Coding

2.1 Light Emitting Diode (LED)

The VLC technology is concentrated on exploiting the property of semiconductor i.e. electroluminescence. Light Emitting Diode or LED are made by P-type and N-type semiconductor materials. When the PN junction of the semiconductor is in forward biased it release energy in form of photons which is known as Electroluminescence.

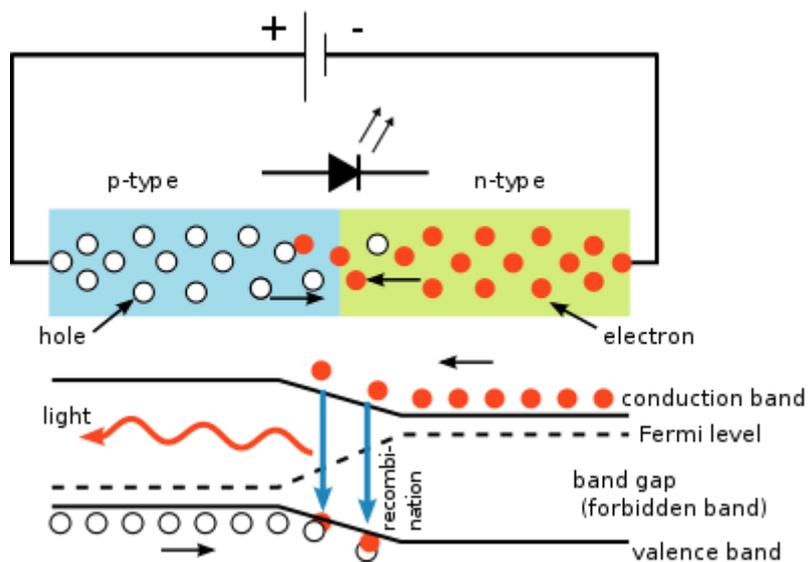


Figure 2 Shows the electron-hole combination principle for LED electroluminescence

When in forward bias, recombination of charge carriers takes place at the P-N junction. The electrons from N-region travels towards the P-region which contains holes and recombines with them which in turn releases energy in form of Light and heat as shown in Figure 2.

2.2 Photodiodes

A Photodiode is a semiconductor device which changes the incident light into electric current or voltage. These photodiodes are usually designed as PIN junction rather than PN junction, reason being the introduction of an intrinsic region results in increase of speed of response of photodiode. A photodiode operates in reverse bias [6]. Vishay BPW24R [7] silicon PiN photodiode is used for this thesis work.

2.2.1 PiN Photodiode

The PiN photodiode has an intrinsic region i.e. lightly doped region in between the P-type semiconductor and the N-type semiconductor. This lightly doped region results in increase in capture area for the photons.

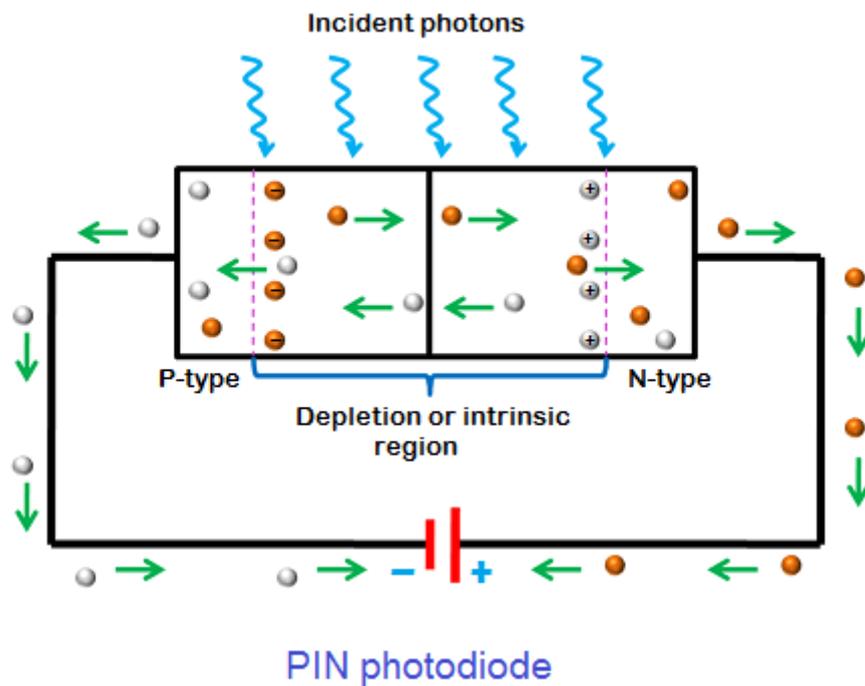


Figure 3 Shows the PiN photodiode and the movement of minority carriers resulting in photocurrent generation [8]

When the photons are incident on the intrinsic region, they get absorbed resulting in formation of charge carriers i.e. electron-hole pairs which are then drifted to the P & N region resulting in the generation of the photocurrent. Due to the reverse bias, this drift process is accelerated resulting in stronger photocurrent generation. The current generated is directly proportional to the number of incident photons.

2.3 Operation Modes of Photodiode

The photodiode operates in two modes i.e. Photovoltaic Mode and Photoconductive Mode. This mode depends upon the biasing. The selection of operation mode is based on the application i.e. speed requirement.

2.3.1 PHOTOVOLTAIC MODE

In Photovoltaic Mode, there is no external biasing i.e. no external voltage is applied to the photodiode. This mode is used when a low response speed application is needed or for detecting low light levels. There is very low dark current (leakage current flowing in photodiode in absence of light) generated in this mode. Therefore, it is very less conductive.

2.3.2 PHOTOCONDUCTIVE MODE

In Photoconductive mode, an external reverse biasing applied to the photodiode, due to which the width of depletion region is increased making it more conductive. We know from the formula that junction capacitance C_j is inversely proportional to the separation distance (W) therefore, on increasing the depletion region the junction capacitance C_j decreases which increases the frequency response of the photodiode making it faster and responsive.

In the formula below C_j = Junction Capacitance, ϵ = permittivity of semiconductor, A = Junction Area, W = width of depletion region, f = frequency, R = Resistance

$$C_j = \frac{\epsilon A}{W}, f = \frac{1}{2\pi R C_j} [9]$$

This mode is used for high speed operations because of the external electric field the new photons generated carriers are immediately swept apart.

2.4 Arduino UNO

Arduino Uno is an open-source microcontroller (ATmega328P) based prototyping development board having 14 Digital input/output pins, 16 analog input pins for interfacing with the microcontroller. Apart from having I/O pins it also supports a USB connection which is used for connecting the board to a computer and programming it. [10] Figure 4 shows the technical specifications of Arduino UNO.

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Figure 4 Shows the technical specification of Arduino Uno prototyping board [10]

Arduino comes with an IDE software for writing the program and uploading it to the Arduino board.

2.5 ON-OFF KEYING (OOK)

On-Off Keying or OOK is a digital modulation technique. It represents the digital data in presence and absence of carrier signal. So, when the carrier is transmitted during the transmission then it is represented as logic 1 and when no carrier is transmitted during the transmission then it is represented as logic 0.

As carrier signal is in two logic states i.e. logic 1 and logic 0, this is the reason it is known as ON OFF Modulation or ON OFF Keying.

2.6 Manchester Coding

The IEEE 802.5.7 [11] standard for short range visible light communication suggests to use Manchester coding as one of the data coding methods with OOK modulation [11]. Manchester coding technique states that utilizes the mid-point of the data bit frame as a transition of the message signal state. A Logic “1” is represented by a mid-point transition of a signal from low to high and a Logic “0” is represented by a mid-point transition from high to low. A more detailed explanation on encoding and decoding the data will be shown in section 4.3.

2.7 Wireless Sensor Network (WSN) and Visible Light Communication (VLC)

Currently all the research in VLC field is dominated towards achieving high data rates for communication. The fact that lighting infrastructure is centrally located and can provide a valuable opportunity of incorporating the sensing and computing means in the indoor space.

VLC technology with slow data transmission i.e. below 100 kbps can be a prime candidate in places where RF is not a viable solution or for short range communication where a fixed quantity of data is being transmitted or received such as a wireless sensor network.

Visible light communication systems supporting the wireless sensor networks can be used in many indoor applications, such as:

- **Smart homes:** These systems can be used to observe and account parameters in the home environment such as temperature data, humidity etc.
- **Indoor Localization:** Different interconnected nodes can relay their location information to the receiver which can improve the indoor navigation and can be an aid for navigation inside of building.
- **Promotional information:** People in retail store can receive promotional information such as coupons and sale on the receiving terminal, which can be mounted on the shopping cart.

3 LITERATURE SURVEY OF VISIBLE LIGHT COMMUNICATION AND WIRELESS SENSOR NETWORKS

This chapter gives an overview on current research and literature work on Visible Light Communication and Wireless Sensor Networks. It will discuss about the ideas that supplemented this thesis work and will discuss about the previous works which have been done in this field.

3.1 VISIBLE LIGHT COMMUNICATION

3.1.1 VISIBLE LIGHT COMMUNICATION FOR INTELLIGENT TRANSPORT SYSTEM(ITIS)

Recently researchers from Korea have demonstrated the use of Visible Light for Vehicle to Vehicle communication via LED headlamps. They developed a system which showed the capability of VLC with 10kbps data rate at a distance of 30m in daylight condition. [12]

This system used the inverse 4-PPM [13] modulation scheme i.e. inverse pulse position modulation scheme which has higher data rate than VPPM [11] and provides higher percent of PWM dimming.

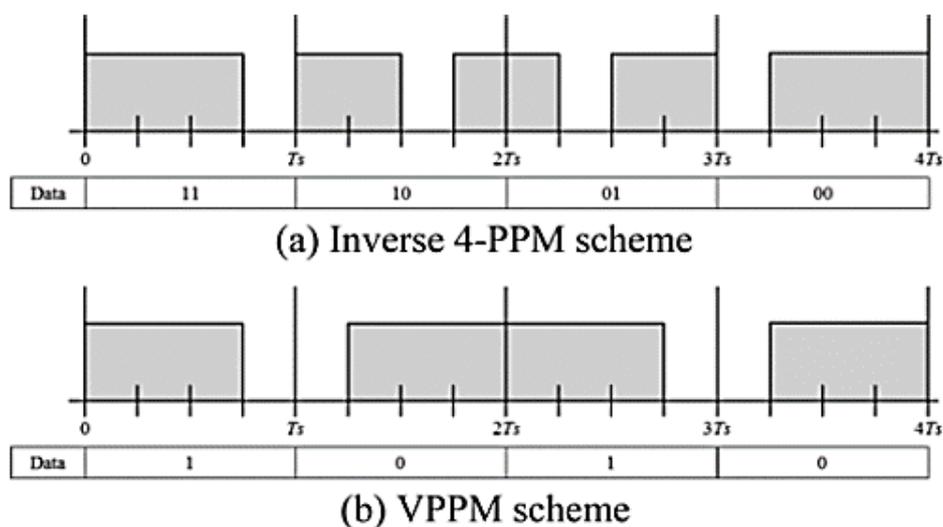


Figure 5 Shows the comparison of inverse 4-ppm scheme and VPPM scheme [12]

Figure 6 below shows the Vehicle to Vehicle (V2V) VLC demonstration at 10kbps data rate. [12]

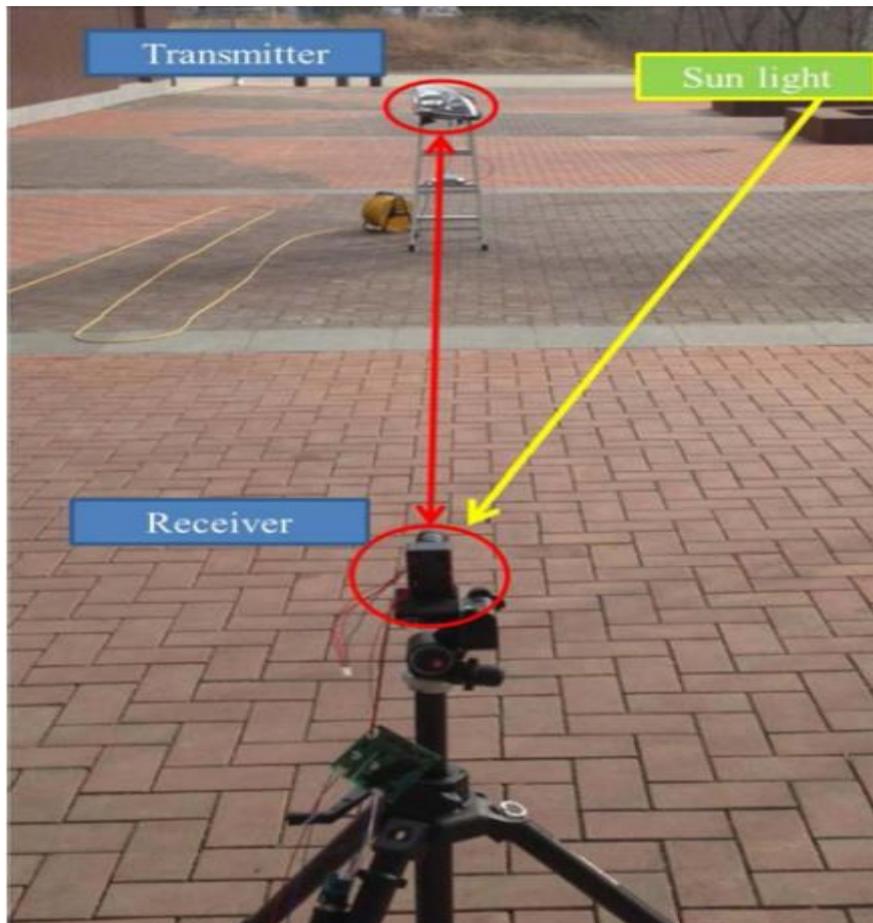


Figure 6 Shows the vehicle to vehicle (V2V) VLC demonstration [12]

3.1.2 VLC SYSTEM FOR DEEP SEA COMMUNICATION

Recently researchers from Coventry University, UK have developed a VLC based system for deep sea communication based on frequency modulation. The system enables the deep-sea divers to communicate. [14]

The Figure 7 & Figure 8 below shows the transmitter and receiver block diagram for this system.

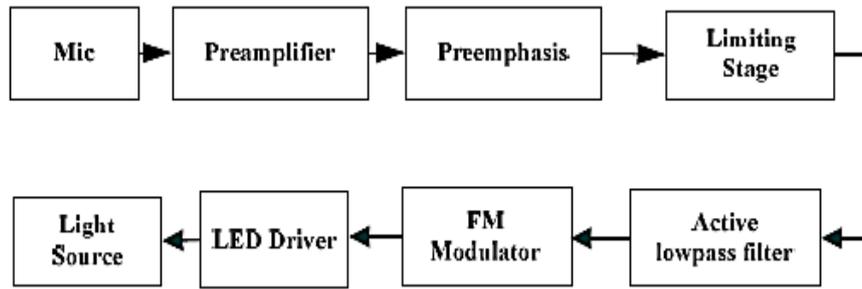


Figure 7 Shows the transmitter block diagram for the deep-sea communication system [14]

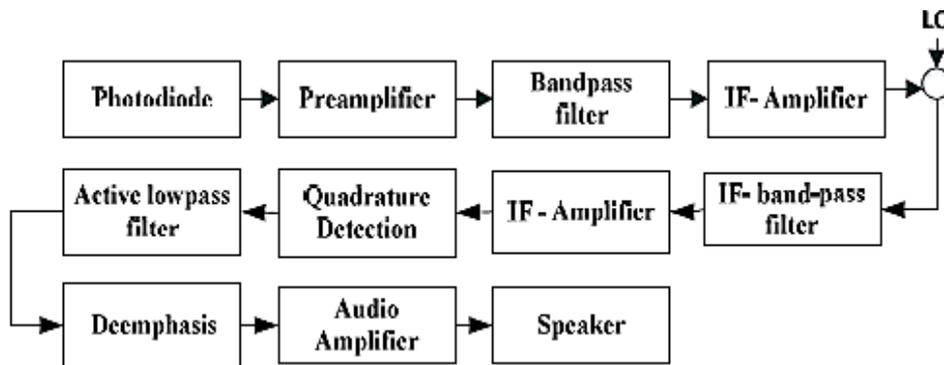


Figure 8 Shows the receiver block diagram for the deep-sea communication system [14]

They developed a full duplex FM based underwater VLC system for voice communication with a distance of 2.7 meter under harsh water conditions

3.1.3 FRAUNHOFER INSTITUTE OF TELECOMMUNICATION

Researchers at the Fraunhofer institute of Telecommunication demonstrated a bidirectional visible light communication prototype achieving speed up to 1 Gb/s in lab. The prototype supports the adaptation according to lightening conditions and combines both lightening and fast wireless data communication. [15]

4 DESIGN REQUIREMENTS

This section will discuss about the requirements for development and implementation the VLC system. It includes both hardware and software implementation requirements. It will an overview about the components required for this system.

4.1 Development Platform

As already discussed in Section 2.4, an Arduino Uno was used as a development board. It is a small, inexpensive, powerful, cross-platform and easy to program development board.

Variety of sensors can be used as an input with which Arduino can sense the environment. It has an ATmega328P [16] microcontroller on board which can be easily programmed using the IDE provided by Arduino. For complete description of the development platform see [10].

4.2 Hardware

The physical layer is implemented by using the hardware i.e. the electrical components used between the Arduino boards. For the transmitter, LED was used to transmit the data via visible light and at the receiver a photodiode was used to convert the light back to the electrical signal which was then processed by the Arduino board to interpret is as data.

A temperature sensor was used at the transmitter for sending the temperature data and other active components like transistor and resistors were used for further optimization of the circuit.

A complete list of components used is available in Appendix 1 with references to complete datasheets of the components.

4.2.1 Transmitter

The main purpose of the transmitter is to transmit the digital data by converting it into visible light, for this this purpose an LED was used because of its linearity between the current and light intensity [17]. The main idea was to modulate the intensity of light corresponding to the transmitted symbol.

LED

As already discussed in section 2.1 the basic principle of LED operation. As the Arduino board has an operational voltage of 5V and current output at its I/O ports is 20mA, to meet this criterion the LED chosen accordingly. *Adafruit 5mm Bright LED* has a forward voltage from 1.8v-2.2V at 20mA current. The LED has its peak wavelength of 640nm.

Temperature Sensor

Temperature was used as the data being sent and received. For this purpose, *TMP36* [18] Temperature Sensor was used. It operates at a voltage of 2.7 – 5.5V and has a 10mV/°C scale factor. It can read temperature from -40°C to 125°C with an accuracy of $\pm 2^\circ\text{C}$.

4.2.2 Receiver

The main purpose of the receiver is to convert the received light into electrical signal using a photodiode. The photodiode is connected to one of the analog pin of the board to take the input, which then converts the analog signal to digital signal using is onboard 6 channel analog to digital converter (ADC).

The photodiode is connected in photoconductive mode as discussed in section 2.3.2 earlier.

Photodiode

As discussed earlier, a photodiode is a semiconductor sensor which converts light into electrical current. *VISHAY BPW24R* photodiode was used in this project as it has quick response time and a large radiant sensitivity area allowing more light capture area [7].

4.3 Software

The major or the main part of this project was software implantation, since it consists of:

- Implementation of Modulation and Demodulation scheme
- Data packaging
- Manchester Encoding

- Manchester Decoding
- Transmitter and Receiver synchronization.
- Receiver calibration for ambient light compensation.

The algorithms used to cover the above-mentioned points are described in Chapter 5 and the source codes for each of them is provided in Appendix 1

Arduino IDE

The ATmega328p microcontroller is programmed with the help of an open-source integrated development environment(IDE) provided by Arduino. The programming is done with C and C++ [19].

Autodesk Eagle

Autodesk Eagle is a PCB design software which was used to draw the schematics for this project [20].

Autodesk Circuits

Autodesk Circuits is an online circuit design and simulation software which provides the functionality of simulating circuits based on Arduino development board [21]. This tool was used to simulate the transmitter circuit.

5 DESIGN AND IMPLEMENTATION OF VLC SYSTEM

This section will explain about the steps involved in designing the prototypes, algorithms used for applying different modulation and coding technique and the results obtained.

For the analysis two approaches were used, one was to create a VLC system with OOK modulation and other was to create a VLC system with OOK modulation and Manchester coding technique.

Figure 9 shows all the prototypes designed and used for this thesis work

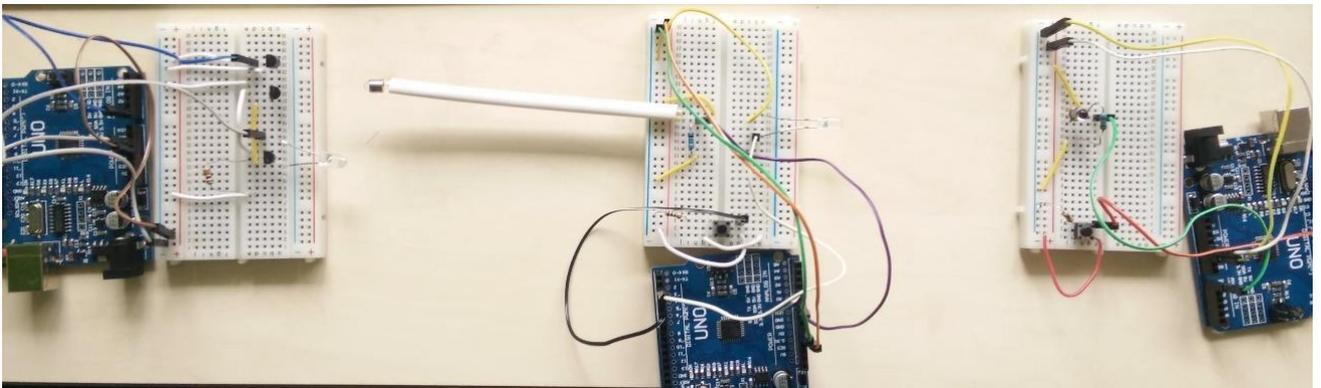


Figure 9 Shows the prototypes from left: Transmitter, Transceiver and a Receiver

5.1 Design of Prototype 1 (On Off Keying Modulation)

This prototype was designed to show the communication between two Nodes, sending and receiving temperature data via visible light as a mode of communication. The block diagram can be seen in Figure 10.

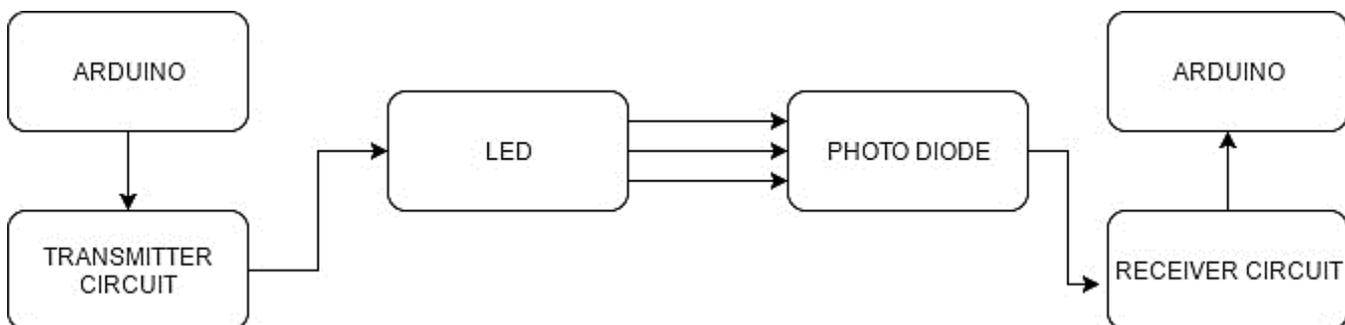


Figure 10 Shows the block diagram of Prototype 1

The Arduino development board is connected with a PC via a USB which provides power and also a user interface for monitoring the data transmitted and received. The transmitter circuit contains a temperature sensor and a LED driving circuit connected to the ATmega328P microcontroller which is mounted on the Arduino board.

The temperature of the environment is sensed by the temperature sensor and is converted into 8-bit binary number. This 8-bit binary data is then transmitted serially via the LED. The LED provides the visible light as a carrier to the signal. The modulation scheme used here is on-off keying (OOK) i.e. modulating the intensity of light source and demodulating it at the receiver. This modulation scheme is implemented on software level. This means that the information bits are encoded into the number of photons which are then transmitted and received by the receiver i.e. a photodiode in this case.

The photodiode then receives the incoming photons and converts it into the 8-bit number which is then displayed on the PC2 as the received temperature data.

5.1.1 TRANSMITTER

The transmitter is designed to transmit the temperature data which is an 8-bit long numeric value. The transmitter module for this prototype includes a temperature sensor TMP36 and LED driving circuit which are connected to the Arduino development board. The temperature sensor TMP36 is a three-terminal sensor which calculates the temperature value by using the voltage drop across its terminals, as the voltage increases its temperature also increases [22]. The sensor has a sensitivity of 10mV/°C scale factor [23].

The temperature data is calculated by the microcontroller which then delivers it to the driver circuit comprising of two transistors in Darlington pair [24] for fast switching of LED. The data received from microcontroller is then modulated using OOK modulation [25].

The schematic for the transmitter, designed using Autodesk Eagle [26] software can be seen in Figure 11 below.

5.1.1.1 Algorithm for Acquiring data and transmission

This section will explain about the working of the transmitter on software level, how the data is acquired, modulated and transmitted.

The temperature data (voltage drop) is read from the sensor TMP36. This value along with the transmitter ID which are 8-bit decimal value are then packaged into a 16-bit frame where the 8-bit ID value acts as a preamble. The frame is then transmitted starting from the MSB to the LSB by the means of LED, which is turned ON for every binary '1' and turned OFF for every binary '0'. Here visible light acts as a carrier for the signal data bits. Figure 13 below shows the algorithm followed for this process.

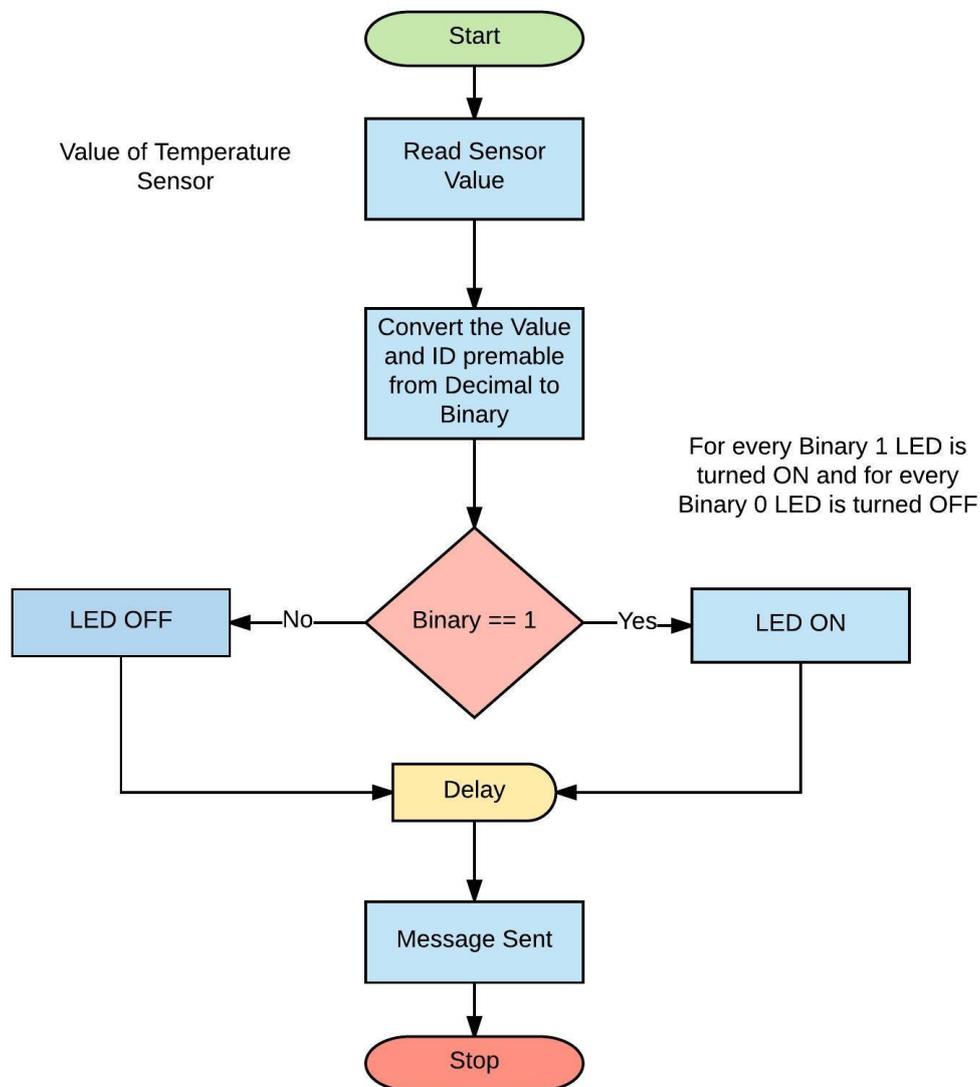


Figure 13 Shows the algorithm followed for the transmission of 8-bit data

5.1.2 RECEIVER

The receiver module comprises of a Vishay BPW24R PiN photodiode, a push button for recalibrating the photodiode according to the ambient light and setting the threshold value for the photodiode, ATmega328P microcontroller mounted on Arduino board for analog to digital conversion (ADC) & demodulation and a PC for monitoring the result.

The schematic for the receiver circuit designed using Eagle [26] software can be seen in Figure 14 below.

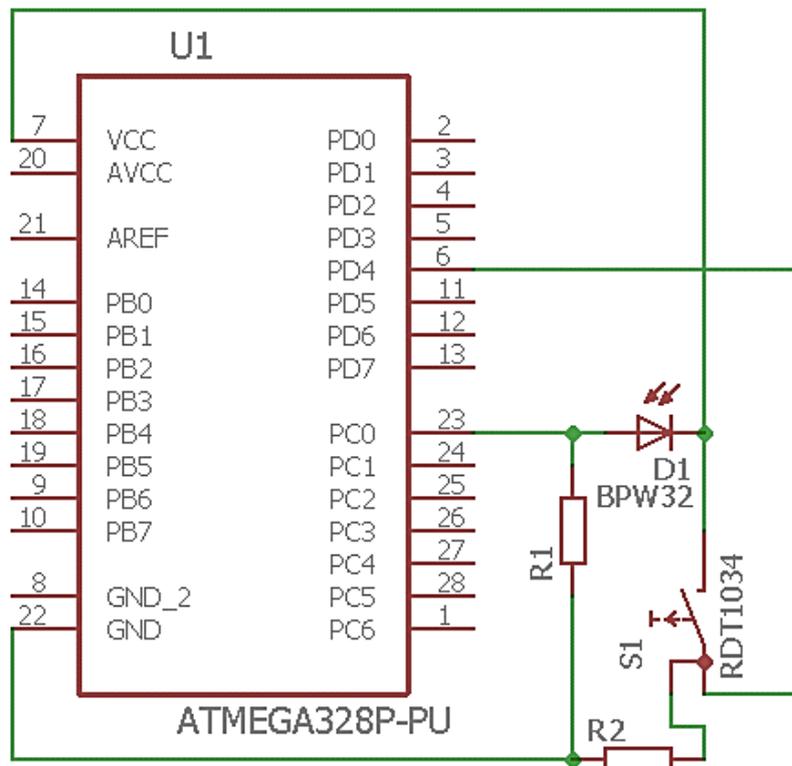


Figure 14 Shows the schematic of the Receiver module

The BPW24R photodiode detects the incoming optical signal from the LED and converts it into electrical signal. Initially a threshold value is set by the photodiode according to the ambient light present. This threshold value determines whether the received binary signal is representing logic '1' or logic '0'. After that the analog to digital conversion is done by the microcontroller.

5.1.2.1 ALGORITHM FOLLOWED BY THE RECEIVER

This section will explain about the working of the receiver on software level, how the receiver is calibrated for compensation of ambient light, data is acquisition and demodulated.

Figure 15 below shows the algorithm followed for this process.

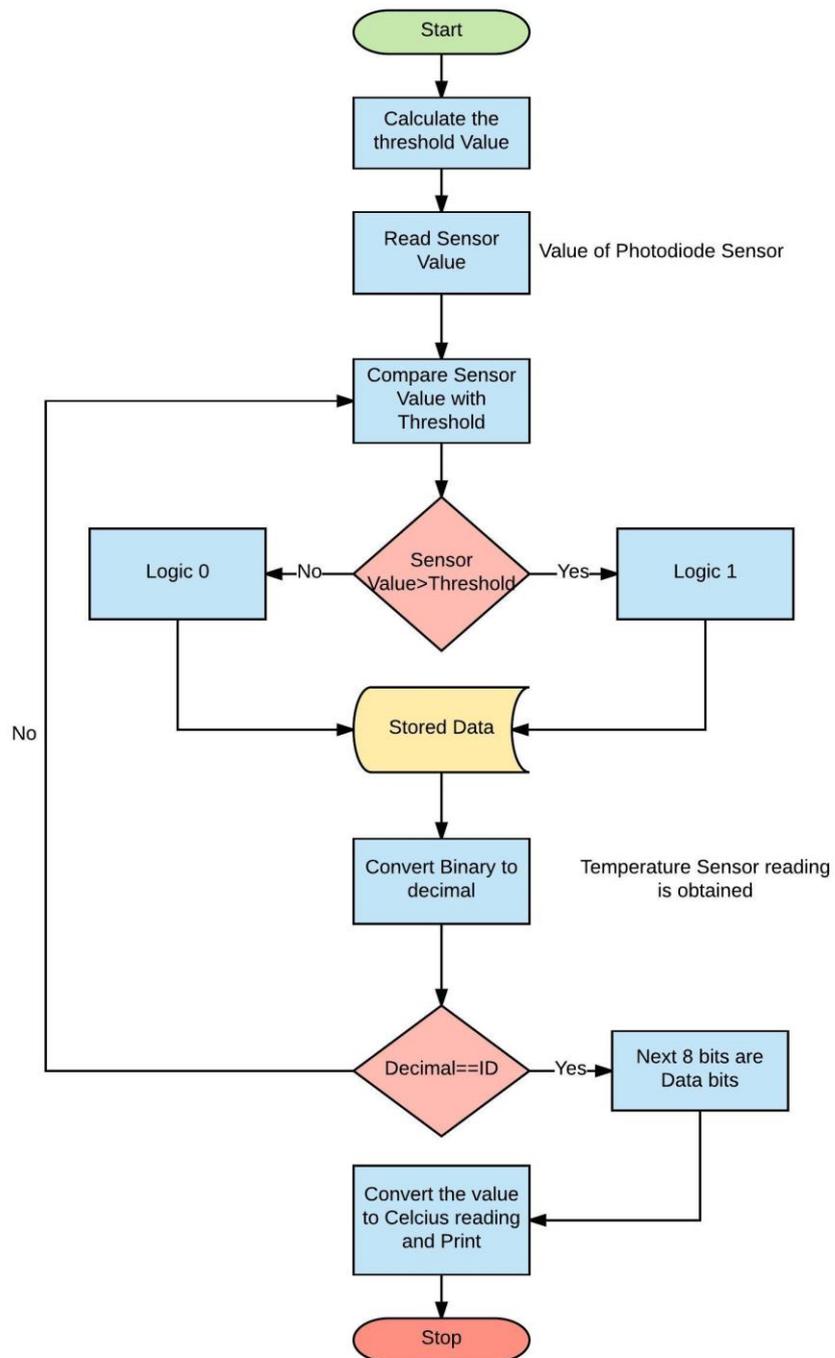


Figure 15 Shows the algorithm for the receiver module

On turning ON the receiver, a threshold value for distinguishing between the logic ‘1’ and logic ‘0’ is calculated, by setting this threshold value the receiver is also calibrated according to the ambient light.

Since the data received is in binary form, it is converted into decimal value. The 8-bit information is read from MSB to LSB by comparing the sensor value with the threshold value. For making sure that the correct data being received is from the correct transmitter, an ID is sent by the transmitter which is first matched with the ID of the receiver. If the ID of both Transmitter and Receiver match, then only further reception of signal occurs which is the data bits (temperature reading).

The Figure 16 below explains the algorithm used for calibrating the receiver according to the ambient light and for calculating the threshold value.

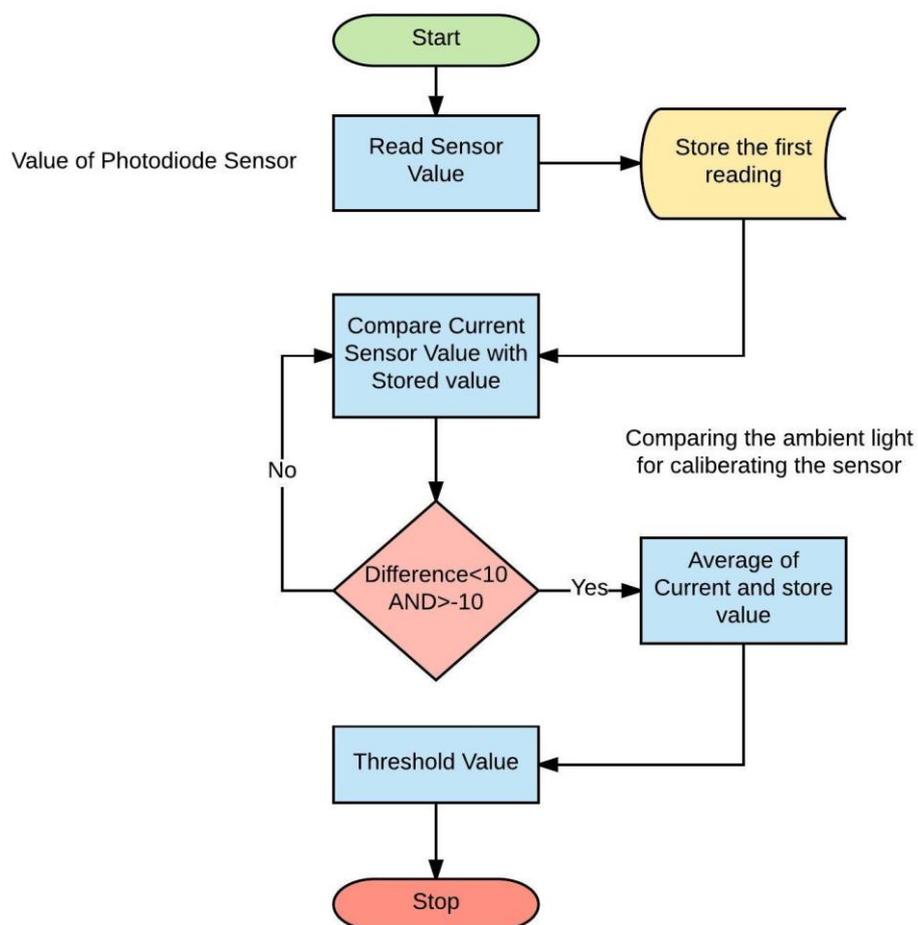


Figure 16 Shows the algorithm used for Receiver calibration and Threshold calculation

For setting this threshold value the photodiode sensor takes one initial reading and stores it, then this initial reading is compared with several consequent readings until the

difference between the initial reading and concurrent readings is high enough to be in limit of -10 and 10. Then an average of the current value and the initial reading is taken which determines the threshold value.

This threshold value helps the receiver for differentiating between Logic 1 and Logic 0.

5.1.3 Transceiver

For testing the scalability of this system, a third independent node was introduced in the system. This additional node will act as a transceiver node i.e. transmitting and receiving the data together. The photodiode receives the binary data from Node A , the microcontroller does the data processing of converting it into decimal value to display the received information and then this data is again processed for retransmission and is transmitted by the LED acting as a transmitter.

Figure 17 shows the block diagram for the whole system. The transceiver Node B will receive data, process it and again transmit it.

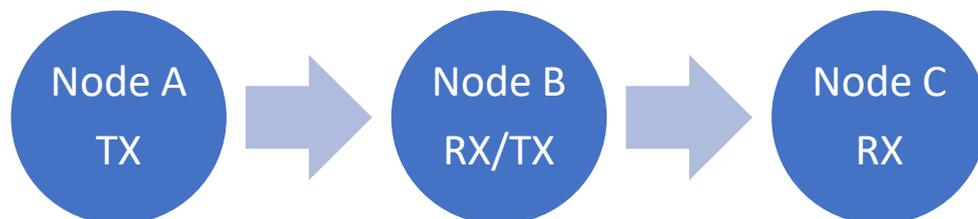


Figure 17 Shows the layout of the Prototype 1.2

Figure 18 shows the schematic for the Receiver/Transmitter circuit. From the schematic, we can see that the transmitter LED and receiver photodiode are both implemented on same circuit.

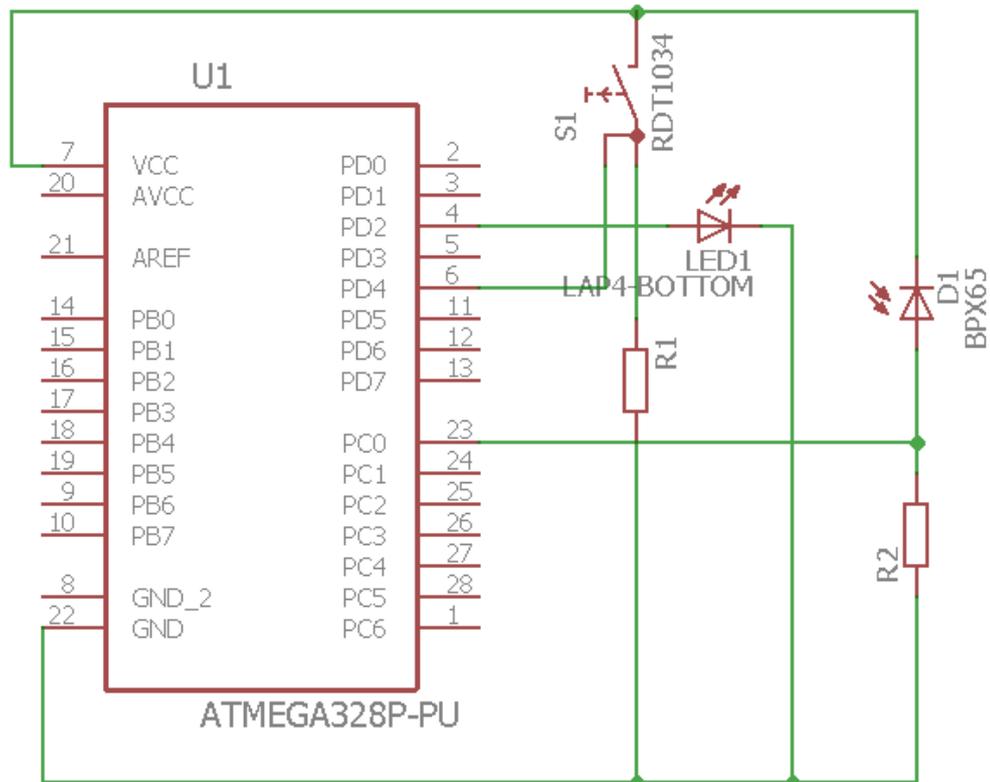


Figure 18 Shows the schematic for the Transceiver circuit

The algorithm works on the same principle as it was for the transmitter and receiver shown in Section 5.1, the only difference is that both of the principles are implemented together on same microcontroller.

5.2 Design of Prototype 2 (On Off Keying Modulation with Manchester Coding)

This section will explain about the implementation of the second prototype i.e. VLC system with Manchester coding. On hardware level this system is same as that of Prototype 1 but all the changes are implemented on software level which are explained in subsequent sections.

5.2.1 Manchester Encoding

Manchester encoding is a coding technique used in many communication systems in conjunction with OOK. This method encodes a zero into the sequence 01 and the one into the sequence 10. This has an advantage in a VLC systems like to avoid flickering. Even

at high frequencies a long sequence of zeros followed by a long sequence of ones will be perceived as annoying flicker of the LED. Manchester encoding solves this by always sending an equal number of ones and zeroes i.e. each bit is sent as a transition from the low state to the high state being encoded as “1” and a transition from high state to the low state being a “0”.

The disadvantage of Manchester encoding is that every logical bit is sent using two physical bits, and as a result the transfer speed is halved compared to OOK without Manchester encoding. How the bits are encoded is displayed in Figure 19 below.

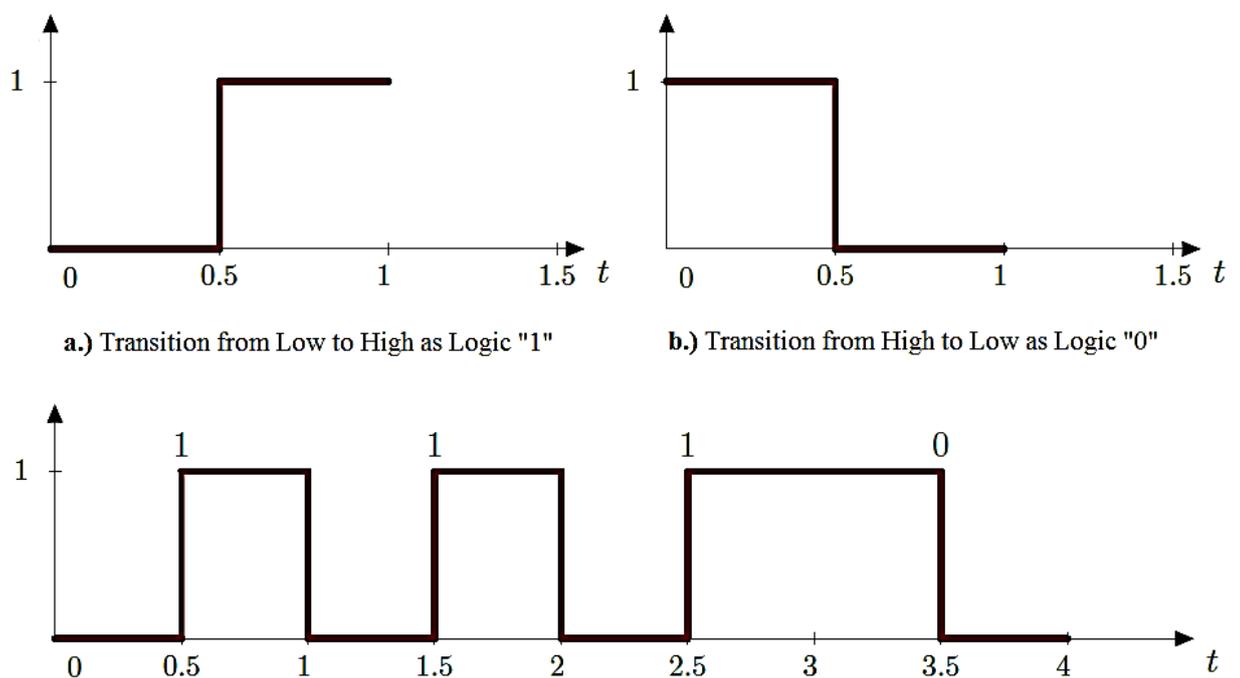


Figure 19 Shows the encoding of bits sing Manchester coding

From the above figure, we can see that at each mid-point of a signal the state changes. It shows the representation of a binary sequence 1110 in Manchester encoding.

Figure 20 below describes the algorithm used in this project for Manchester encoding.

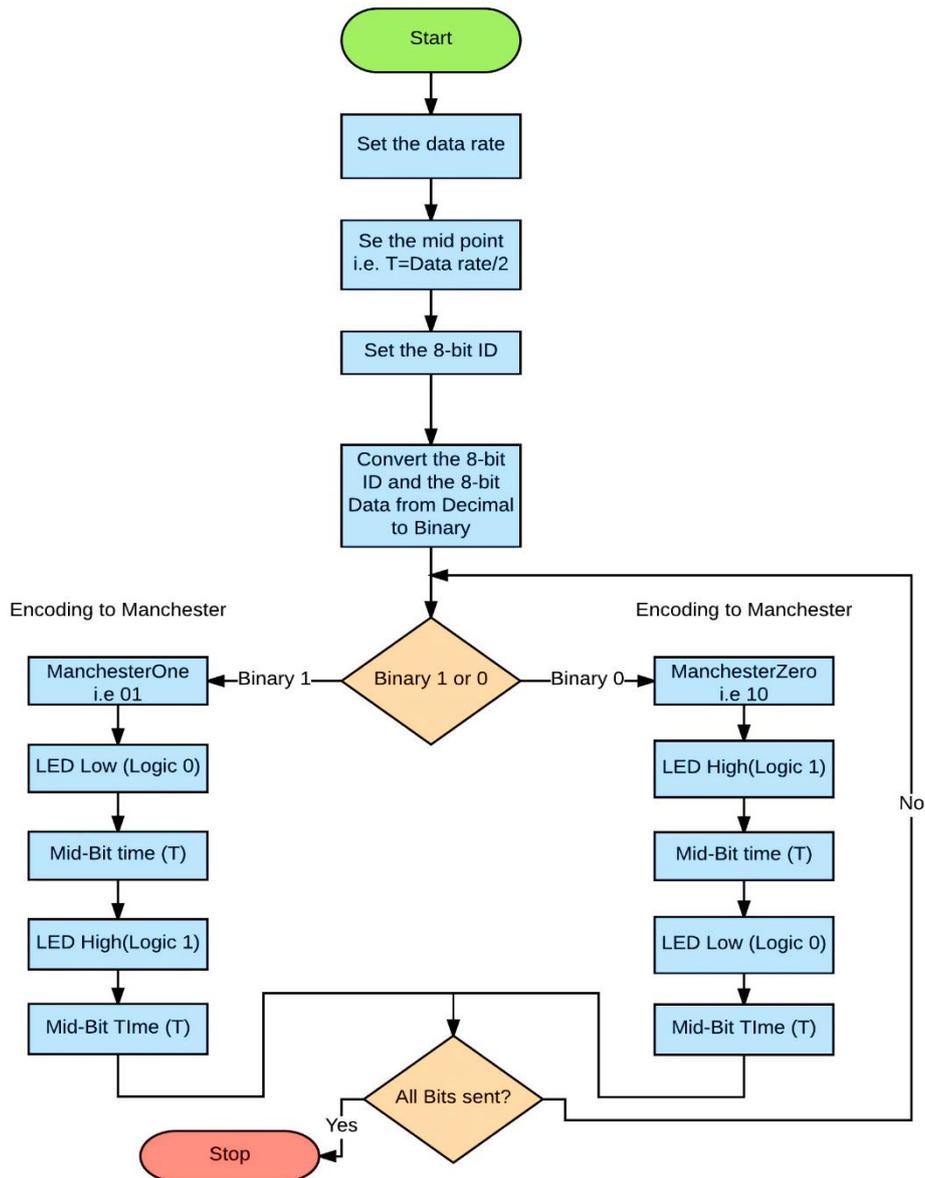


Figure 20 Shows the algorithm for implementing Manchester encoding on the microcontroller

5.2.2 Manchester Decoding

Manchester decoding is the part where most of the problems arise and have to be done carefully. The most important and complex part in Manchester decoding is synchronization, i.e. making the receiver able to detect the transition point or the edge coming from the demodulation circuit. Atmel corporation has explained the process of Manchester encoding and decoding very clearly in its documentation [27].

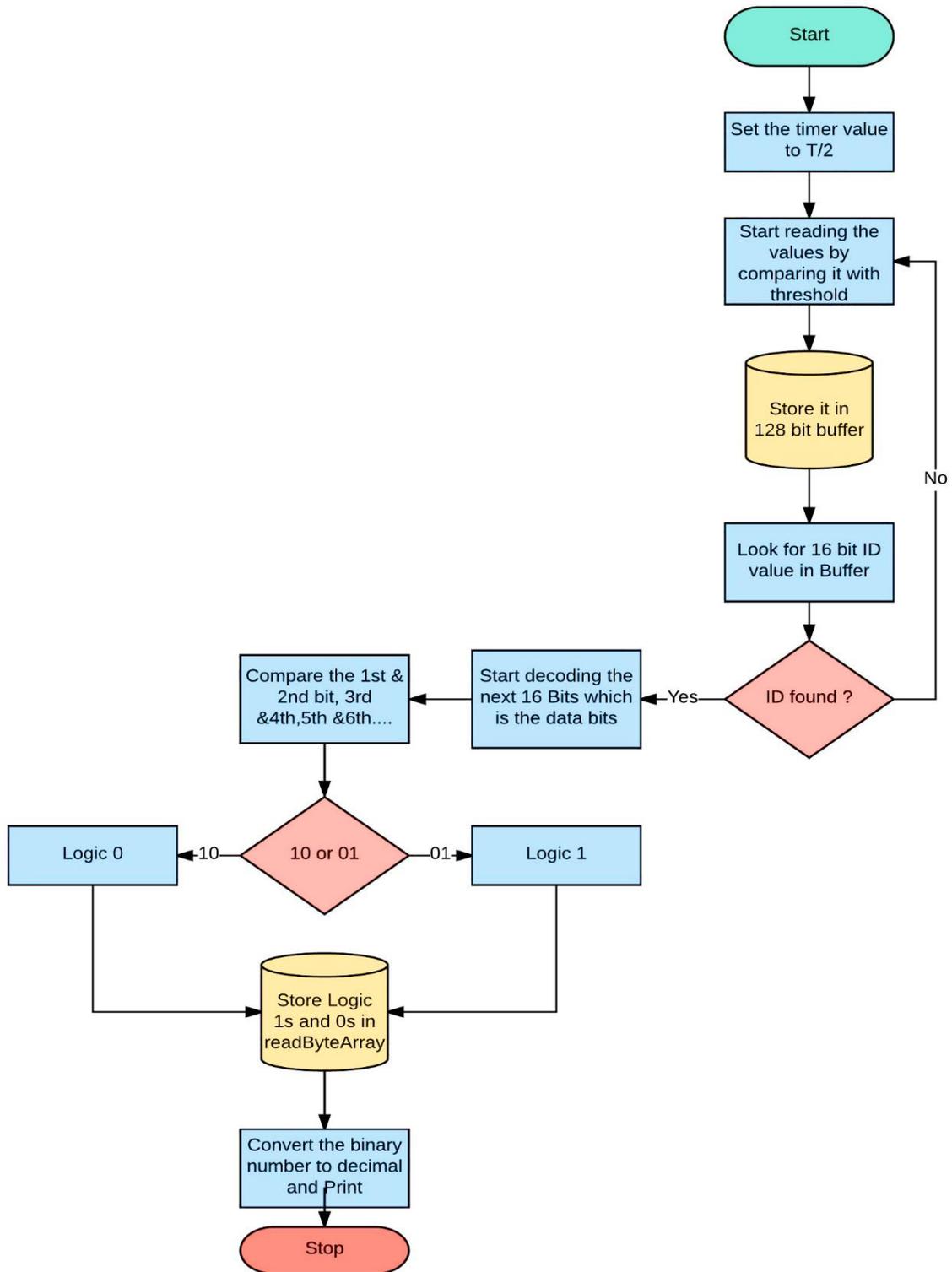


Figure 21 Shows the algorithm for implementing Manchester Decoding on the microcontroller

The Figure 21 above shows the algorithm followed for implementing Manchester Decoding on the receiver in this project.

6 EXPERIMENTAL SETUP AND RESULTS

This section shows and discusses about the results of the tests conducted for the two VLC systems. It discusses about the amount of data lost during the transmission and reception of data for the two approaches i.e. VLC with and without Manchester coding, secondly also presents the results on addition of a third node for the same.

The readings were noted 20 seconds for each setup at 10 different distances between the nodes in an environment of ambient light of 2000 lux.

The Figure 22 shows the transmitted received temperature data on the PC terminal. We can see that the COM 7 port shows the transmitted data in form of Binary value. Each data value is preceded by an ID value preamble, which lets the receiver know that the next value is the data value.

COM7 (Transmitted)	COM5 (Received)
1 : The temperature is 16.89 °C	1 : The temperature is 16.00 °C
10110100	2 : The temperature is 17.00 °C
00010000	3 : The temperature is 16.00 °C
2 : The temperature is 17.38 °C	4 : The temperature is 16.00 °C
10110100	5 : The temperature is 17.00 °C
00010001	6 : The temperature is 16.00 °C
3 : The temperature is 16.89 °C	7 : The temperature is 17.00 °C
10110100	8 : The temperature is 16.00 °C
00010000	9 : The temperature is 17.00 °C
4 : The temperature is 16.89 °C	10 : The temperature is 17.00 °C
10110100	11 : The temperature is 17.00 °C
00010000	12 : The temperature is 16.00 °C
5 : The temperature is 17.38 °C	13 : The temperature is 17.00 °C
10110100	14 : The temperature is 17.00 °C
00010001	15 : The temperature is 16.00 °C
6 : The temperature is 16.89 °C	16 : The temperature is 22.00 °C
10110100	17 : The temperature is 17.00 °C
00010000	18 : The temperature is 16.00 °C
7 : The temperature is 17.38 °C	19 : The temperature is 17.00 °C
10110100	20 : The temperature is 17.00 °C
00010001	21 : The temperature is 17.00 °C
8 : The temperature is 16.41 °C	22 : The temperature is 17.00 °C
10110100	23 : The temperature is 17.00 °C
00010000	24 : The temperature is 16.00 °C
9 : The temperature is 17.38 °C	25 : The temperature is 16.00 °C
10110100	26 : The temperature is 16.00 °C
00010001	27 : The temperature is 17.00 °C
10 : The temperature is 17.38 °C	28 : The temperature is 17.00 °C
10110100	29 : The temperature is 17.00 °C
00010001	30 : The temperature is 21.00 °C
11 : The temperature is 17.38 °C	31 : The temperature is 17.00 °C
10110100	32 : The temperature is 18.00 °C
00010001	33 : The temperature is 17.00 °C
12 : The temperature is 16.89 °C	34 : The temperature is 17.00 °C
10110100	35 : The temperature is 16.00 °C
00010000	36 : The temperature is 16.00 °C
13 : The temperature is 17.38 °C	37 : The temperature is 17.00 °C
10110100	38 : The temperature is 22.00 °C
	39 : The temperature is 17.00 °C

Figure 22 Shows the transmitted and Received data on the terminal

6.1 Two Nodes with and without Manchester coding

Both systems were tested for different data rates and at different distances upto 1.5 meters. After 1.5 meters the receiver was not able to receive any data. For the system without Manchester coding, the highest speed at which the receiver was able to receive data was 500bits per second or 62.5 Bytes per second. For the system with Manchester coding, the highest speed at which the receiver was able to receive data was 100bits per second, this is because of the fact that for Manchester encoding 1 bit is divided into two equal halves which are then transmitted separately.

We can see from the graph shown in Figures 23 & Figure 24 below, that the number of data bits received do not change on changing the distance between the transmitter and the receiver. This shows that the VLC system is capable of receiving the same amount of data till the receiver is able to detect the variation in signals from the transmitter. In this case the distance till it can detect is 1.5 meters. After 1.5 meters the receiver is not able to interpret the incoming signals and no data is received.

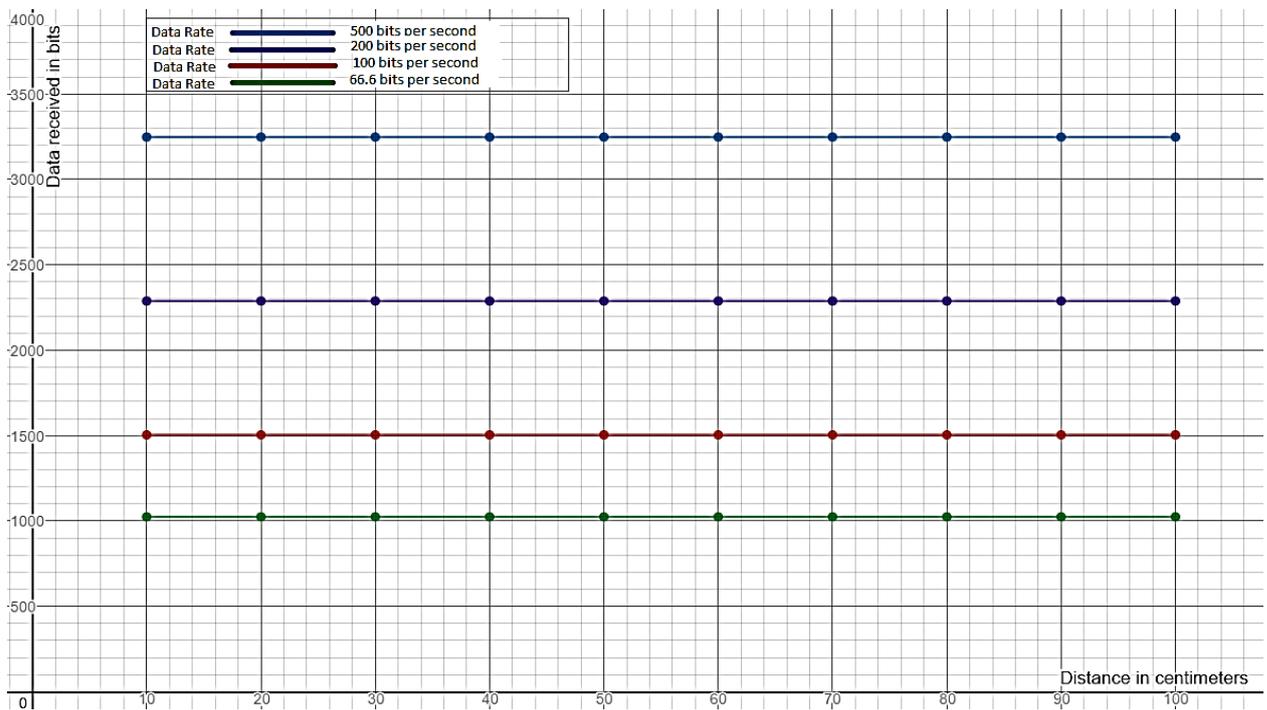


Figure 23 Shows the graph for Data received (bits per second) VS Distance for VLC system without Manchester Coding

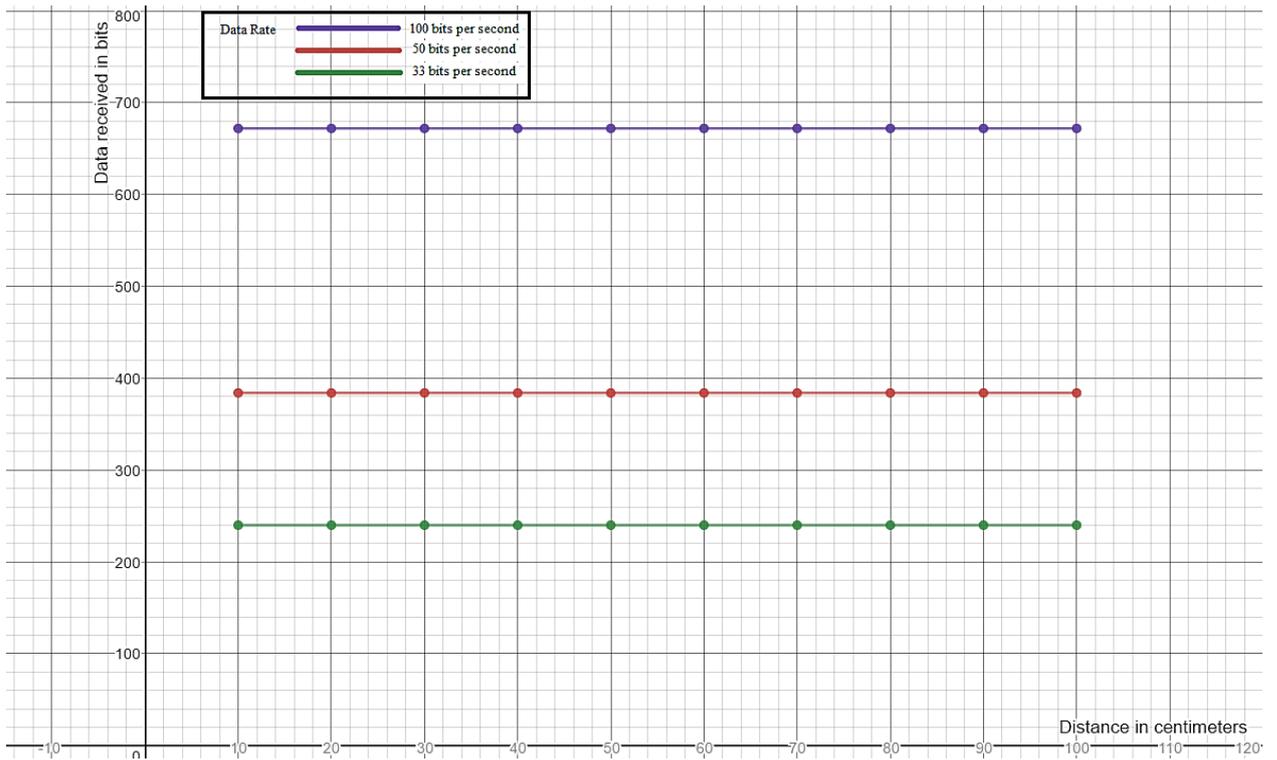


Figure 24 Shows the graph for Data received (bits per second) VS Distance for VLC system without Manchester Coding

Figure 25 & Figure 26 below shows the Data Loss in percent Vs Data Rate graph for VLC system with and without Manchester respectively. The measurements were taken at different distances ranging from 10 cm to 100 cm for different data rates. The results show an increment in the Data Lost during the transmission is increased as the data rate is increased. The system is not acceptable for a data rate higher than 500 bits/seconds.

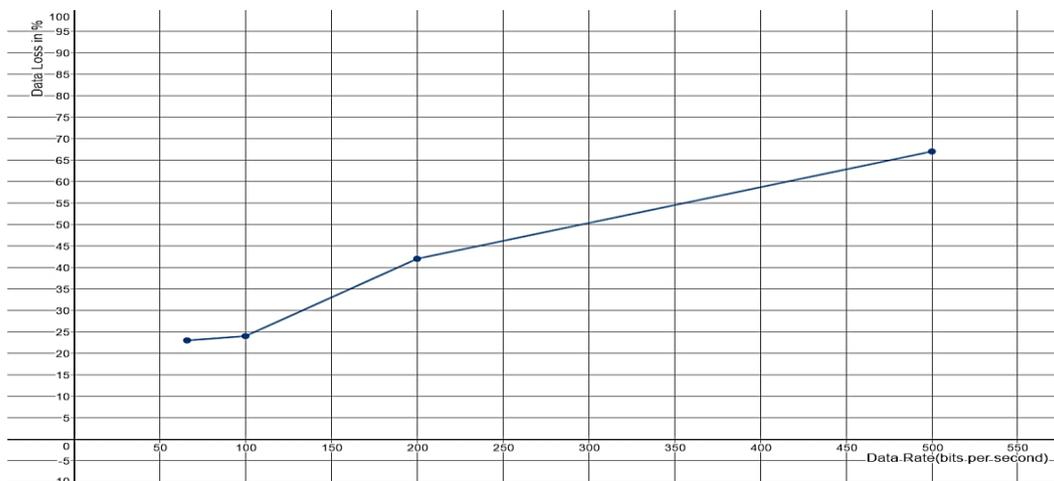


Figure 25 Shows the Data Loss Vs Data Rate graph for VLC system without Manchester encoding

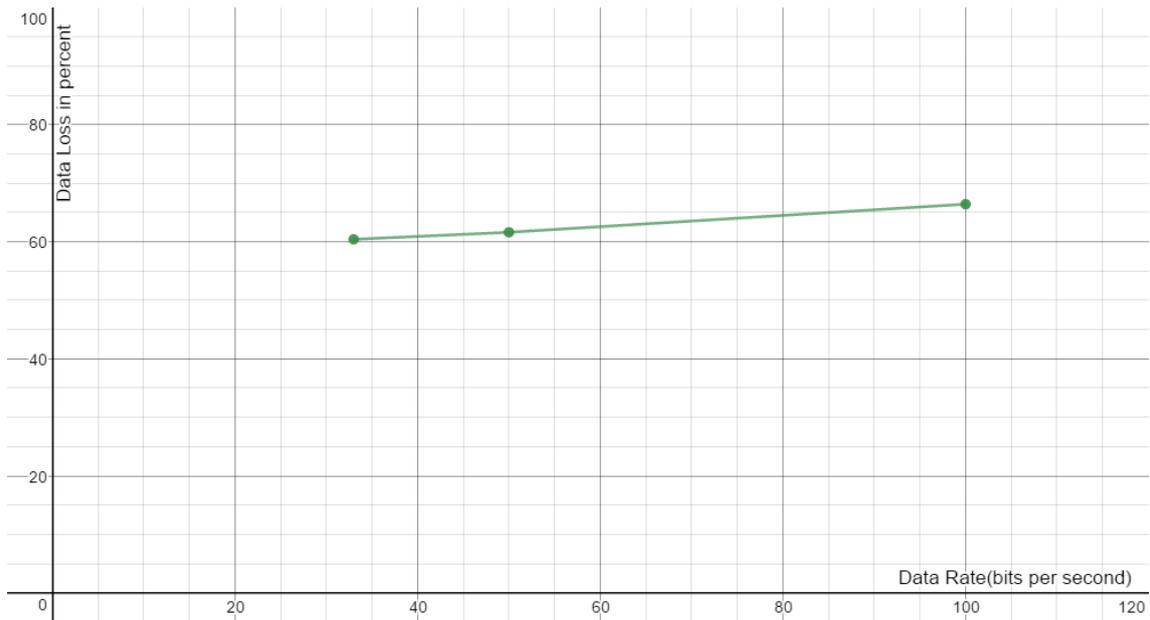


Figure 26 Shows the Data Loss Vs Data Rate graph for VLC system with Manchester encoding. The Figure 26 above also shows similar pattern of Data Loss for different data rates as shown earlier for Prototype 1 i.e. VLC system without Manchester coding in Figure 25. The percentage of data lost while transmission is increased as the data rate is increased, which is between 60 – 70 %.

Figure 27 & Figure 28 below shows the output of the transmitter and the receiver on the oscilloscope for both the systems respectively. From the Figure 27, we can see that the both (Green for Transmitter and Yellow for Receiver) the outputs are similar.



Figure 27 Shows the outputs of Tx and Rx on the oscilloscope for the system without Manchester Coding.

From Figure 28 below, we can see the output (TX Yellow and RX Green) for the VLC system with Manchester coding.

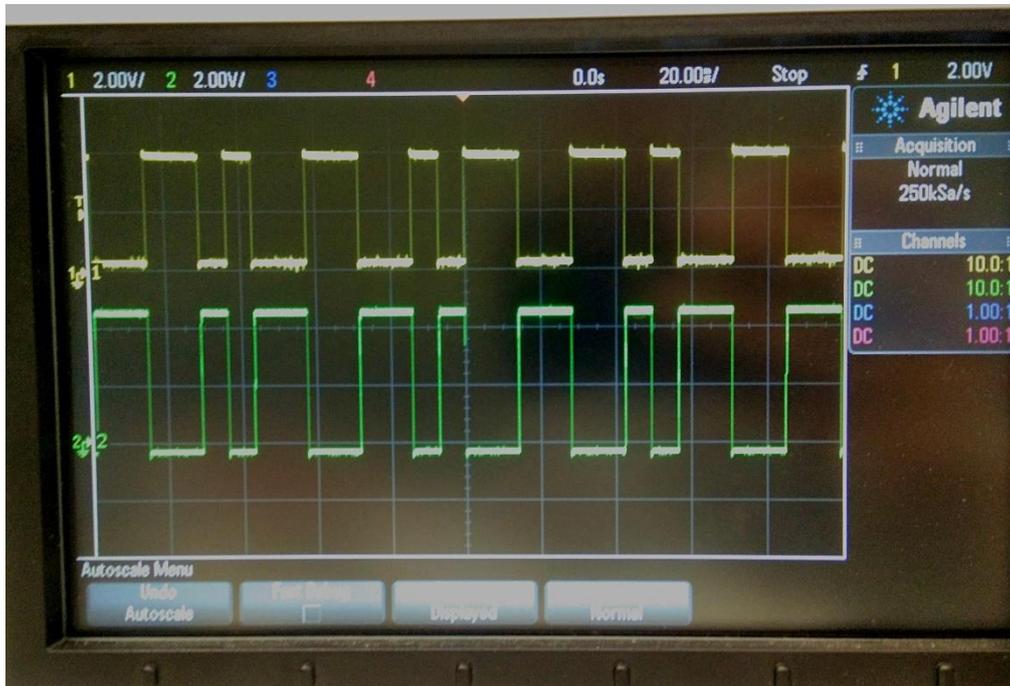


Figure 28 Shows the outputs of Tx and Rx on the oscilloscope for VLC system with Manchester Coding.

6.2 Three Nodes with and without Manchester coding

Here the results obtained on addition of a third node is discussed. The total distance from Node A to Node C is 2.5 meters. Figure 29 shows the experimental setup for the three Nodes.



Figure 29 Shows the experimental setup for the three Nodes

As can be seen from the graph in the Figure 30 & Figure 31 below, the data lost during the transmission is increased as the data rate is increased. But one point to be noted here

is that the amount of data lost while using Manchester coding scheme is higher than that of without using it. This is because of delay caused by microcontroller while processing the data which causes loss of synchronization resulting in more data loss.

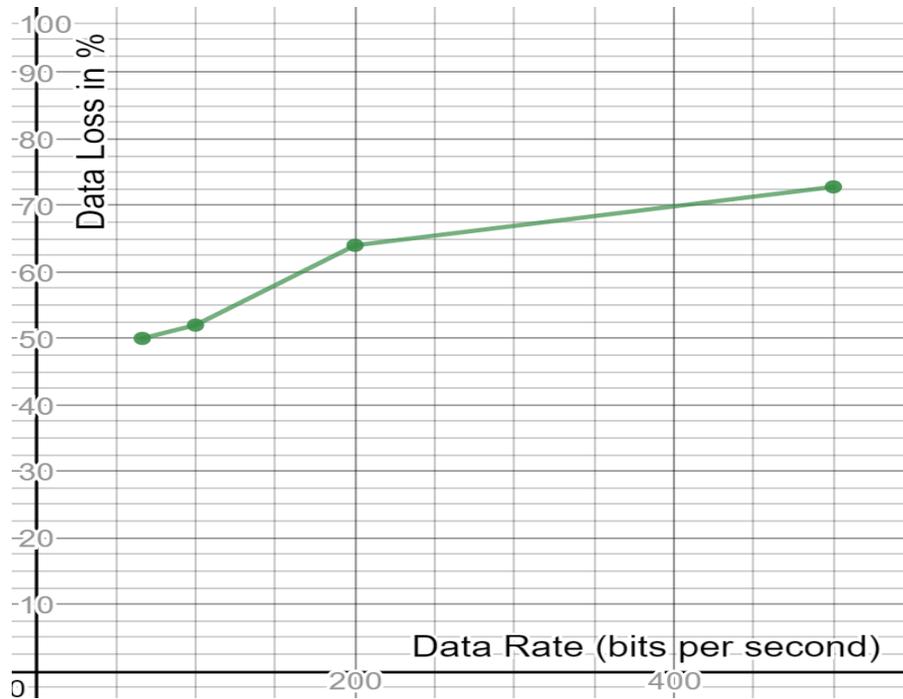


Figure 30 Shows the graph for Data Loss Vs Data Rate for system without Manchester coding

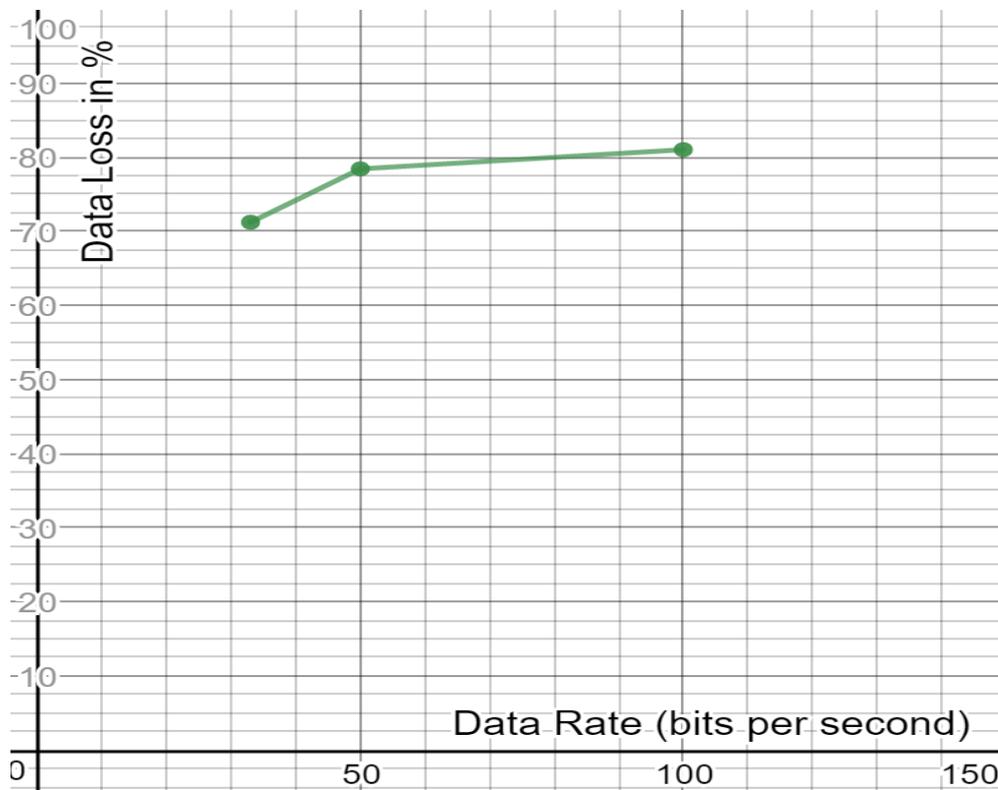


Figure 31 Shows the graph for Data Loss Vs Data Rate for system with Manchester coding

The Figure 32 & Figure 33 below shows the readings taken from the oscilloscope for both setups. Here the four channels represent the output from the two transmitters and the two receivers. The pink channel is for TX1, Green – RX1, Blue - TX2 and Yellow – RX2.

From Figure 32 We can see that the outputs of TX1 & RX1 are almost identical and also the outputs of TX2 & RX2 are identical. But the outputs of all the four are not identical. This is because when data is received by RX1 the microcontroller again processes the data for retransmission and in that process some data is lost, scope of improvement is discussed in Section 7.1



Figure 32 Shows the output at each TX and RX in the three nodes

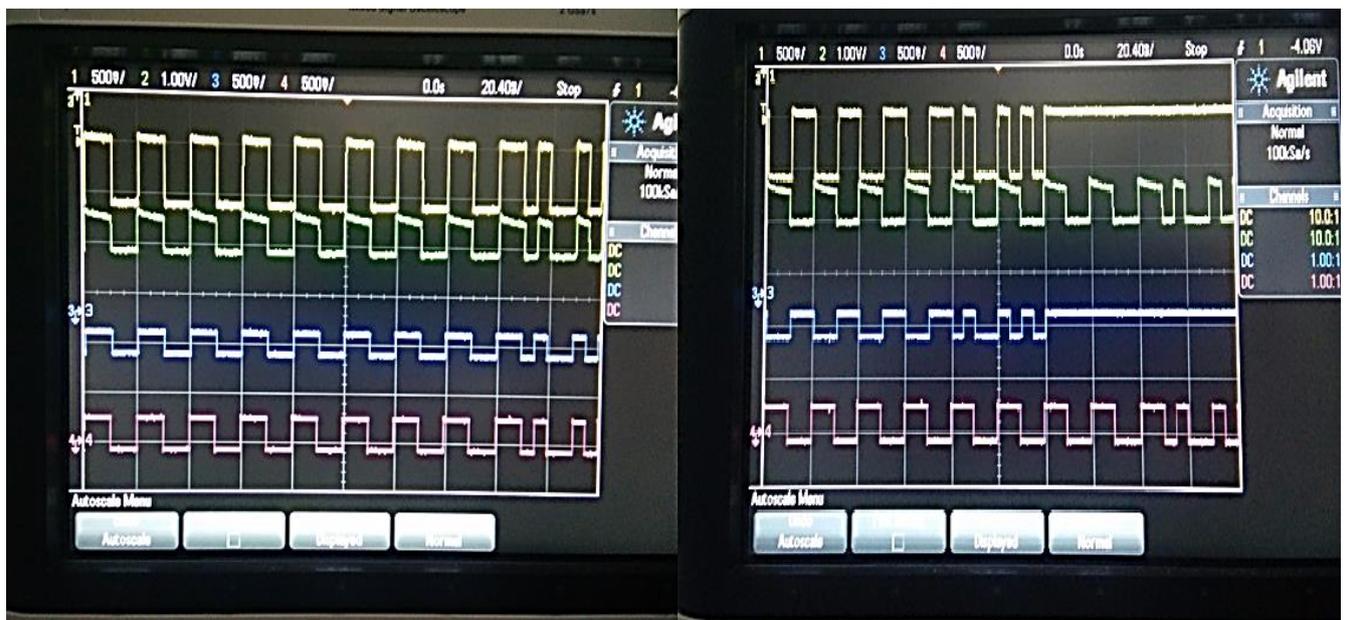


Figure 33 Shows the Output at all TX and RX on the three nodes in prototype 2. The left and right images represent different interval of time.

Figure 33 above shows the oscilloscope output for second prototype with three nodes. We can see in the left image that the outputs of TX1, RX1, TX2 & RX2 are almost identical for short duration, but the outputs of all the four are not identical at some stages in between. This is because of the loss of synchronization and data loss as mentioned in previous paragraph. Due to this a lot of data is lost while using Manchester coding scheme.

7 DISCUSSION AND SUMMARY

This project has achieved its goal of designing and implementing a low cost (45/- BOM in Appendix 2), short range indoor VLC system (stated in Section 1.3), though there are still many areas of improvements which are discussed in Section 7.1, but the system works and demonstrates the application of Visible light as a mode of communication.

This final section will discuss about this project, compare its results with the goals set at the beginning and will discuss about further scope of improvement.

7.1 General Discussion and Conclusion

Visible Light Communication is the rapidly growing segment in the field of optical wireless communication. There are many advantages to this rapidly growing field but there are many challenges also. It is still an emerging field but improvements are being made rapidly.

This thesis work was focused on design, implementation and demonstration of this technology. This was done by two different methods for implementing Visible Light Communication i.e. (OOK with Manchester Encoding and OOK without Manchester Encoding). Furthermore, the system was extended by adding additional transceiver node for evaluating its scope as a wireless sensor node.

From the results obtained it can be concluded that implementing the system by using just OOK modulation have certain advantages like higher data rate, easy synchronization and less error but this modulation scheme alone cannot solve the purpose of incorporating this

system in the present infrastructure and using it as a light source because it cannot solve the issue of flickering of light as discussed in previous section. While using this modulation scheme along with Manchester encoding scheme which is the second method, solves the issue of flickering. But it needs a better synchronization mechanism which is discussed in next section.

Secondly it is also evident from the results that the amount of data received is not fully dependent on the distance between the transmitter and receiver i.e. a constant amount of data is received as long as the receiver is able to detect the incoming light source.

So, a VLC system was designed and implemented, which is capable of sending and receiving data via visible light over the range of 1.5 meters in a full lit room, can calibrate the receiver according to ambient light conditions and is scalable also. It covers all the objectives mentioned in the section 1.3.1.

It is intended that this thesis will provide a solid foundation for further research and development on improving this technology and making its way in Wireless Sensor Networks.

7.2 Scope of improvement

Though thesis work successfully implemented and demonstrated a Visible Light Communication System, but there is still a scope of improvement in this system which can improve its efficiency further more. In this section, we will discuss about the same.

Improving the synchronization in Manchester coding:

In this project, a pre-defined timer was used for the purpose of finding the mid-point of the signal and for synchronizing the transmitter and the receiver. Though by using this method the system works fine but it is susceptible to loss of synchronization which causes loss of data and adds some error values. This can be improved by using interrupts, triggers, flags and Interrupt Service Routine in the microcontroller. Using interrupts will not only improve the synchronization, it will also make the system more power efficient.

Error Control:

An error control mechanism can be added to the system by using an error handling mechanism which will help to detect and correct errors that occur during the transmission.

Dimming Support:

Dimming Support can be added to this system, which means the transmitter light can be dimmed while still transmitting the data. And this was also seen in the results that receiver receives the same amount data till it is able to detect the light source.

For achieving this IEEE proposed a new modulation technique known as Variable Pulse Position Modulation (VPPM) in its 802.15.7 standard for short range VLC. [11]

This modulation scheme incorporates two modulations schemes i.e. PPM (Pulse Position Modulation) and PWM (Pulse Width Modulation) working independently.

The amplitude of pulse is always constant in VPPM and pulse width controls the dimming, not amplitude. Within a unit period the bits '1' and '0' have same pulse width and are distinguished by the pulse position within the same period.

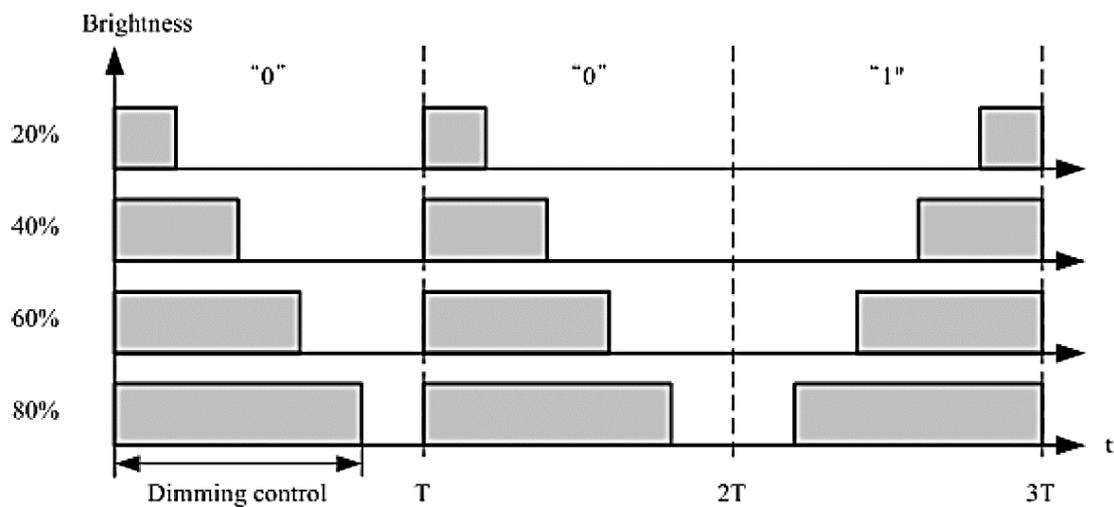


Figure 34 Shows the schematic mechanism of VPPM dimming as described in the IEEE standard [11]. The Figure 34 above shows the mechanism for controlling the brightness or dimming control mechanism by VPPM. The pulse width can be adjusted based on the dimming requirement.

8 References

- [1] M. A. K. a. M. Uysal, "Survey on free space optical communication: Acommunication theory perspective," *IEEE Communication Surveys Tuts*, vol. 16, pp. 2231-2258, 2014.
- [2] H. Hematkhah and Y. Kavian, "DCPVP: Distributed Clustering Protocol Using Voting and Priority for Wireless Sensor Networks," 2015.
- [3] T. A. Edison, "Improvement in electric lights". Patent US Patent 214,636, 22 April 1879.
- [4] Wikipedia, "Signal Lamp," Wikimedia Foundation, Inc., [Online]. Available: http://en.wikipedia.org/wiki/Signal_lamp.
- [5] "America's Library," The Library of Congress, [Online]. Available: http://www.americaslibrary.gov/jb/gilded/jb_gilded_bell_3.html.
- [6] J. F. Cox, "Photodiodes," in *Fundamentals of linear electronics: integrated and discrete*, Cengage Learning, 2001, p. 91.
- [7] V. Semiconductor, "BPW24R Silicon PiN Photodiode Data Sheet," [Online]. Available: <http://www.vishay.com/docs/81520/bpw24r.pdf>.
- [8] "Physics and Radio-Electronics," [Online]. Available: <http://www.physics-and-radio-electronics.com/electronic-devices-and-circuits/semiconductor-diodes/photodiodesymboltypes.html>.
- [9] O. Optoelectronics, "Photodiode Characteristics and Applications," [Online]. Available: <http://www.osioptoelectronics.com/application-notes/an-photodiode-parameters-characteristics.pdf>.
- [10] "Arduino," [Online]. Available: <https://www.arduino.cc/en/main/ArduinoBoardUno>.
- [11] "IEEE Standard for short range wireless," *IEEE Std 802.15.7*, p. 14, 2011.
- [12] J.-S. J. J. H.-C. D.-W. S.-Y. J. J.-H.YOO, "DEMONSTRATION OF VEHICULAR VISIBLE LIGHT," *International Journal of Automotive Technology*, vol. 17, no. 2, 2016.
- [13] H.-D. a. J. S.-Y. Moon, "Multi-coded variable PPM for high data rate visible light communications," *J.Optical Society of Korea*, pp. 107-114, 2012.
- [14] P. G. O. S. a. S. R. Zahir Ahmad, "Design of a Visible Light Communication System for Deep Sea Divers Based on Analogue Frequency Modulation," *International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, no. 10, 2016.
- [15] F. H. H. Institute, "Visible Light Communication System up to 1 Gbit/s – Fraunhofer Heinrich Hertz Institute," [Online]. Available: <https://www.hhi.fraunhofer.de/en/departments/pn/technologies-and-solutions/hardware-products/vlc-system-1-gbits.html>.
- [16] M. T. Inc, "ATmega328P," [Online]. Available: <http://www.microchip.com/wwwproducts/en/ATmega328p>.
- [17] E. F. Schubert, "Light-Emitting Diodes," Cambridge University Press, 2006, p. 391.

- [18] Analog Devices, “Temperature Sensor -TMP36,” [Online]. Available: http://cdn.sparkfun.com/datasheets/Sensors/Temp/TMP35_36_37.pdf.
- [19] Arduino, “Arduino IDE Software,” [Online]. Available: <https://www.arduino.cc/en/main/software>.
- [20] Autodesk Inc, “Autodesk Eagle,” 2017. [Online]. Available: <https://www.autodesk.com/products/eagle/overview>.
- [21] “Autodesk Circuits,” Autodesk, [Online]. Available: <https://circuits.io>.
- [22] T. Aggarwal, “Types of Temperature sensores & their working principles,” ELPROCUS, [Online]. Available: <https://www.elprocus.com/temperature-sensors-types-working-operation/>.
- [23] N. Semiconductor, *LM35 Precision Centigrade Temperature Sensors*, Nation Semiconductors, 2000.
- [24] “Electronics Tutorials,” [Online]. Available: <http://www.electronics-tutorials.ws/transistor/darlington-transistor.html>.
- [25] R. R. a. S.-K. L. S. Rajagopal, “IEEE 802.15.7 visible light communication: modulation schemes and dimming support,” *IEEE Communications Magazine*, vol. 50, pp. 72-82, 2012.
- [26] “Autodesk Eagle,” Autodesk Inc., [Online]. Available: <http://www.autodesk.com/products/eagle/overview>.
- [27] A. Corporation, *Manchester Codin Basics*, San Joe: Atmel Corporation, 2015.
- [28] “Serial Communication,” Wikimedia Foundation, Inc, [Online]. Available: https://en.wikipedia.org/wiki/Serial_communication.
- [29] L. S. Sterling, *The Art of Agent-Oriented Modeling*, London: The MIT Press, 2009.
- [30] U.S. Department of Commerce, Office of Spectrum Management, “United States Frequency Allocation Chart | NTIA,” [Online]. Available: <https://www.ntia.doc.gov/page/2011/united-states-frequency-allocation-chart>.
- [31] E. P. R. P.-J. a. J. R. F. J. Lopez Hernandez, “Low-Cost Diffuse Wireless Optical Communication System based on White LED,” *IEEE International Symposium on Consumer Electronics*, p. 1, 2006.

Appendix 1 – Source Code

Temp TX

```
#define DELAY 15
#define LED 2 //Output pin where LED is connected
#define ID 180 //ID of Receiver
#define BITLENGTH 8 //1 Byte
#define FIRSTBIT pow(2, (BITLENGTH-1))//MSB Value
#define TRUE 1
#define TEMPERATURE A0
int i, j;
int count=0;
long Timer = millis();
float voltage;
void writeByte(int);
int counter=0;
void setup() {
    //Setting the LED as Output
    pinMode(LED,OUTPUT);
    Serial.begin(115200);
}

void loop()
{
    if ((millis()-Timer)<20000)
    {
        i=analogRead(TEMPERATURE); //Reading the temperature(i.e. the voltage drop on
        the sensor)
        writeByte(ID); //Sending ID of Receiver
        writeByte(i);//sending the temperature to the receiver
        float temperature = (float)i*5/1024;
        temperature=(temperature -0.5)*100;

        Serial.print(++count); // Sending the value of temperature
        Serial.print (" : The temperature is ");
        Serial.print(temperature); // Sending the value of temperature
        Serial.println(" 'C");
    }
    if (count >256)
    {
        count=0;
    }
}
//Sending 1byte number
void writeByte(int decimal)
{
```

```

    //converting the decimal value to binary and sending 8 bit information from
    MSB to LSB
    int i,binary;
    for(i=0;i<BITLENGTH;i++)
    {
        binary = (int)decimal/FIRSTBIT; //Getting the first binary bit value
        decimal= (decimal & ((int)FIRSTBIT -1)); //Setting the first bit to zero
        decimal=decimal<<1; //Shift all bits by one to left
        if(binary==TRUE)
        {
            digitalWrite(LED,HIGH);
            //Serial.print("1");
        }else
        {
            digitalWrite(LED,LOW);
            //Serial.print("0");
        }
        delay(DELAY);
    }
    //Serial.println();
    digitalWrite(LED,LOW);
}

```

Temp RX

```

#define SETTING 5
#define DELAY 2
#define BITLENGTH 8
#define DIFFERENCE 10
#define S1 4
#define ID 180
#define sensorPin A0
int sensorValue = 0;
int middleValue;
int inputByte;
void checkMiddle();
int readByte();
int j;
float temperature;
long counter=millis();
int count=0;
void setup() {
    Serial.begin(115200);
    checkMiddle();//Setting the value between Logic 0 and Logic 1
}

void loop() {
    do
    {
        if(digitalRead(S1)==HIGH)
        {
            checkMiddle();
            Serial.println("Sensor is calibrated ");
        }
    }
}

```

```

    sensorValue = analogRead(sensorPin);
}while(sensorValue<=middleValue);

if((inputByte=readByte())==ID)
{
    inputByte=readByte(); //Reading 1 Byte
    //Calculating temperature
    temperature = (float)inputByte*5/1024;
    temperature=(temperature -0.5)*100;

    Serial.print(++count); // Sending the value of temperature
    Serial.print (" : The temperature is ");
    Serial.print(temperature); // Sending the value of temperature
    Serial.println(" 'C");
    if (count >256)
    {
        count=0;
    }
}
}

void checkMiddle()
{
    int difference;
    middleValue = analogRead(sensorPin); //Cehcking initial voltage on the
sensor

    do{
        sensorValue=analogRead(sensorPin);//Checking second voltage on sensor
        difference= sensorValue - middleValue;//Comparing the values of the
voltage
        delay(SETTING);
    }while(difference<DIFFERENCE && difference>-DIFFERENCE);
    //Calculating the middle value between Logic 1 and Logic 0
    middleValue = (sensorValue + middleValue)/2;
}

int readByte()
{
    int i,input=0;
    for(i=0;i<BITLENGTH;i++)
    {//Repeat for each bit
        sensorValue=analogRead(sensorPin);//Read the sensor Value
        if(sensorValue>=middleValue)
        {
            input++;
        }
        //Binary shift to store another bit
        input=input<<1;
        delay(DELAY);
    }
    //Return the value if input
    input=input>>1;
    return input;
}

```

RXTX CODE

```

#include "SendRecieve.h"

void setup() {
  //Setting the LED as Output
  pinMode(LED,OUTPUT);
  pinMode(S1,INPUT_PULLUP);
  checkMiddle();//Setting the value between Logic 0 and Logic 1
  Serial.begin(115200);
}

void loop() {
  do
  {
    if(digitalRead(S1)==HIGH)
    {
      checkMiddle();
      //Serial.println("Sensor is calibrated ");
    }
    checkMiddle();
    sensorValue = analogRead(sensorPin);
  }while(sensorValue<=middleValue);

  if((inputByte=readByte())==ID)
  {
    inputByte=readByte(); //Reading 1 Byte
    //Calculating temperature
    temperature = (float)inputByte*5/1024;
    temperature=(temperature -0.5)*100;
    //++counter;
    //Serial.print (counter);
    Serial.print (" : The tempereature is ");
    Serial.print(temperature); // Sending the value of temperature
    Serial.println(" 'C");
  }

  writeByte(ID); //Sending ID of Receiver
  i=temperature; //Reading the temperature(i.e. the voltage drop on the
sensor)
  writeByte(i);//sending the temperature to the receiver

}

```

SendReceive.h

```

#define DELAY 15
#define SETTING 5
#define DELAY 15
#define DIFFERENCE 10
#define S1 4
#define ID 180
#define sensorPin A0
#define LED 2 //Output pin where LED is connected
#define ID 180 //ID of Receiver
#define BITLENGTH 8 //1 Byte

```

```

#define FIRSTBIT pow(2, (BITLENGTH-1))//MSB Value
#define TRUE 1
#define TEMPERATURE A0
int i, j;
float voltage;
void writeByte(int);
int counter=0;

int sensorValue = 0;
int middleValue;
int inputByte;
void checkMiddle();
int readByte();
float temperature;
//Sending 1byte number
void writeByte(int decimal)
{
    //converting the decimal value to binary and sending 8 bit information from
    MSB to LSB
    int i,binary;
    for(i=0;i<BITLENGTH;i++)
    {
        binary = (int)decimal/FIRSTBIT; //Getting the first binary bit value
        decimal= (decimal & ((int)FIRSTBIT -1)); //Setting the first bit to zero
        decimal=decimal<<1; //Shift all bits by one to left
        if(binary==TRUE)
        {
            digitalWrite(LED,HIGH);
            //Serial.print("1");
        }else
        {
            digitalWrite(LED,LOW);
            //Serial.print("0");
        }
        delay(DELAY);
    }
    Serial.println();
    digitalWrite(LED,LOW);
}

void checkMiddle()
{
    int difference;
    middleValue = analogRead(sensorPin); //Cehcking initial voltage on the
    sensor

    do{
        sensorValue=analogRead(sensorPin); //Checking second voltage on sensor
        difference= sensorValue - middleValue; //Comparing the values of the
        voltage
        delay(SETTING);
    }while(difference<DIFFERENCE && difference>-DIFFERENCE);
    //Calculating the middle value between Logic 1 and Logic 0
    middleValue = (sensorValue + middleValue)/2;
}

```

```

int readByte()
{
    int i,input=0;
    for(i=0;i<BITLENGTH;i++)
    { //Repeat for each bit
        sensorValue=analogRead(sensorPin); //Read the sensor Value
        if(sensorValue>=middleValue)
        {
            input++;
        }
        //Binary shift to store another bit
        input=input<<1;
        delay(DELAY);
    }
    //Return the value if input
    input=input>>1;
    return input;
}

```

TXTemp_Manchester

```

#define DELAY 30
#define LED 2 //Output pin where LED is connected
#define ID 170 //ID of Receiver
#define BITLENGTH 8 //1 Byte
#define FIRSTBIT pow(2, (BITLENGTH-1)) //MSB Value
#define TRUE 1
#define TEMPERATURE A0

int i, j;
int count=0;
long Timer = millis();
void writeByte(int);

void setup() {
    //Setting the LED as Output
    pinMode(LED,OUTPUT);
    Serial.begin(115200);
}

void loop() {
    if ((millis()-Timer)<20000)
    {
        writeByte(ID); //Sending ID of Receiver
        //writeByte(ID);
        i = analogRead(TEMPERATURE);
        float temperature = (float)i*5/1024;
        temperature=(temperature -0.5)*100;
        Serial.print(++count); // Sending the value of temperature
        Serial.print (" : The temperature is ");
        Serial.print(temperature);
        Serial.println(" 'C");
        if (count >256)
        {

```

```

        count=0;
    }
    writeByte(i); //Reading and sending the temperature(i.e. the voltage drop
on the sensor)
}
}
void manchesterOne(int T)
{
    digitalWrite(LED,LOW);
    //Serial.print("0");
    delay(T/2);
    digitalWrite(LED,HIGH);
    //Serial.print("1");
    delay(T/2);
}

void manchesterZero(int T)
{
    digitalWrite(LED,HIGH);
    //Serial.print("1");
    delay(T/2);
    digitalWrite(LED,LOW);
    //Serial.print("0");
    delay(T/2);
}

//Sending 1byte number
void writeByte(int decimal)
{
    //converting the decimal value to binary and sending 8 bit information from
MSB to LSB
    int i,binary;
    for(i=0;i<BITLENGTH;i++)
    {
        binary = (int)decimal/FIRSTBIT; //Getting the first binary bit value
        decimal= (decimal & ((int)FIRSTBIT -1)); //Setting the first bit to zero
        decimal=decimal<<1; //Shift all bits by one to left
        if(binary==TRUE)
        {
            manchesterOne(DELAY);
            //Serial.print("1");
        }else
        {
            manchesterZero(DELAY);
        }
        //delay(DELAY/2);
    }
    //Serial.println();
    //digitalWrite(LED,LOW);
    //manchesterOne(DELAY);
    //manchesterZero(DELAY);
}

```

RXTEMP_Manchester

```

#include "functions.h"
#define SETTING 1
#define DELAY 15
#define BITLENGTH 128
#define DIFFERENCE 10
#define S1 4
#define sensorPin A0
int sensorValue = 0;
int middleValue;
long inputByte;
void checkMiddle();
void readByte();
int Buffer[128];
int ID=0;
int count=0;
void setup() {
  Serial.begin(115200);
  pinMode(S1,INPUT_PULLUP);
  checkMiddle();//Setting the value between Logic 0 and Logic 1
}

void loop() {
  do
  {
    if(digitalRead(S1)==HIGH)
    {
      checkMiddle();
      Serial.println("Sensor is calibrated ");
    }
    sensorValue = analogRead(sensorPin);
  }while(sensorValue<=middleValue);

  readByte();
  ID = findID(Buffer);
  int IDGot=Byte(Buffer, (ID));
  // Serial.println("ID received :");
  // //Printing the ecoded ID recieved
  // for (int k=ID; k<(ID+16);++k)
  // {
  //   Serial.print(Buffer[k]);
  // }
  // Serial.println();
  // Serial.println("ID received decoded:");
  // Serial.println(Byte(Buffer,ID));//Printing the Decoded ID value in
  decimal

  // Printing the Encoded Data received
  if (IDGot==170)
  {
    Serial.println("Data received :");
    for (int k=(ID+16); k<(ID+32);++k)
    {

      Serial.print(Buffer[k]);
    }
    Serial.println();
    Serial.println("Data received decoded:");
  }
}

```

```

Serial.println(Byte(Buffer, (ID+16)));
int AnalogTemperature = Byte(Buffer, (ID+16));
float temperature = (float)AnalogTemperature*5.0/1024.0;
temperature = (temperature-0.5)*100.0;
Serial.print(++count); // Sending the value of temperature
Serial.print(" :Temperature is:");
Serial.print(temperature); // Printing the decode data in decimal. Here ID
acts as a pointer for starting of frame. So from 16th position it will start
reading the data
Serial.println(" 'C");
if (count >256)
{
    count=0;
}
}

// Serial.println("Data received :"); // for monitoring
// for (int k=(ID+16); k<(ID+32);++k)
// {
//     Serial.print(Buffer[k]);
// }
// Serial.println();

}

//setting the threshold for the receiver sensor i.e adjusting to ambient
light
void checkMiddle()
{
    int difference;
    middleValue = analogRead(sensorPin); //Checking initial voltage on the
sensor

    do{
        sensorValue=analogRead(sensorPin); //Checking second voltage on sensor
        difference= sensorValue - middleValue; //Comparing the values of the
voltage
        delay(SETTING);
    }while(difference<DIFFERENCE && difference>-DIFFERENCE);
    //Calculating the middle value between Logic 1 and Logic 0
    middleValue = (sensorValue + middleValue)/2;
}
//Storing the encoded data into buffer
void readByte()
{
    int i;
    long input=0;
    for(i=0;i<BITLENGTH;i++)
    { //Repeat for each bit
        sensorValue=analogRead(sensorPin); //Read the sensor Value
        if(sensorValue>=middleValue)
        {
            Buffer[i]=1;
        }
        else
        {

```

```

    Buffer[i]=0;
}

delay(DELAY);
}
}

```

Functions.h

```

//binary to decimal conversion
int readByteArray(int Array[])
{
    int i,input=0;
    for(i=0; i<8; i++)
    {
//Binary shift to store another bit
        if (Array[i]>1)
        {
            input=input<<1;
            input|=0;
        }
        else
        {
            input=input<<1;
            input|=Array[i];
        }
    }
    return input;
}
//Decoding the received data
//comparing the consequent bits to find 1 bit of data
int Compare(int bit1, int bit2)
{
    if (bit1>bit2)
    {
        return 0;
    }
    else if (bit1<bit2)
    {
        return 1;
    }
}

//decoding 1 byte of data i.e. extracting 8 bits from 16 bits
int Byte(int buf[], int pointer)
{
    int h=0;
    int nr=0;
    int Decoded[8];
    for (int currentbit =0;currentbit<16;++currentbit)
    {
        //filtering out 8 comparisions i.e. not comparing the 2nd and 3rd bit

```

```

    if(currentbit%2==0) // 8 comparisons
    {
        Decoded[h]=Compare(buf[currentbit+pointer], buf[1+currentbit+pointer]);
        ++h;
    }
    else if (currentbit%2==0 && pointer%2!=0)
    {
        Decoded[h]=Compare(buf[currentbit+pointer], buf[1+currentbit+pointer]);
        ++h;
    }
}
nr = readByteArray(Decoded);
return nr;
}

//Finding the ID
int findID(int buf[])
{
    int i=0;
    for (i;i<(128-16);++i)
    {
        long input =0;
        for (int j=0;j<16;++j)
        {
            //Binary shift to store another bit
            input=input<<1;
            input|=buf[i+j];//coverting it to decimal
        }
        if (input==26214)// Encoded ID value i.e. 170
        {
            break;
        }
    }
    return i;
}
}

```

RXTX Manchester (NODE B – Transceiver)

```

#include "functions.h"
#include "Transmit.h"
#define SETTING 1
#define DELAY2 5
#define BITLENGTHBUFFER 128
#define DIFFERENCE 10
#define S1 4
#define sensorPin A0
int sensorValue = 0;
int middleValue;
long inputByte;
void checkMiddle();
void readByte();
int Buffer[BITLENGTHBUFFER];
int ID=0;
int count=0;
int AnalogTemperature=0;
void setup() {

```

```

Serial.begin(115200);
pinMode(S1,INPUT_PULLUP);
pinMode(LED,OUTPUT);
checkMiddle();//Setting the value between Logic 0 and Logic 1
}

void loop() {
  do
  {
    if(digitalRead(S1)==HIGH)
    {
      checkMiddle();
      Serial.println("Sensor is calibrated ");
    }
    sensorValue = analogRead(sensorPin);
  }while(sensorValue<=middleValue);

  readByte();
  ID = findID(Buffer);
  int IDGot=Byte(Buffer,(ID+16));

  if (IDGot==170)
  {
    AnalogTemperature = Byte(Buffer,(ID+32));
    float temperature = (float)AnalogTemperature*5.0/1024.0;
    temperature = (temperature-0.5)*100.0;
    //Serial.print(" :ID Pointer:");
    //Serial.println(ID);// Printing the decode data in decimal. Here ID acts
as a pointer for starting of frame. So from 16th position it will start
reading the data
    Serial.print(++count); // Sending the value of temperature
    Serial.print(" :Temperature is:");
    Serial.print(temperature);// Printing the decode data in decimal. Here ID
acts as a pointer for starting of frame. So from 16th position it will start
reading the data
    Serial.println(" 'C");
    if (count >256)
    {
      count=0;
    }
    writeByte(170); //Sending ID of Receiver
    writeByte(170);
    writeByte(AnalogTemperature); //Reading and sending the temperature(i.e.
the voltage drop on the sensor)

  }

}

//setting the threshold for the receiver sensor i.e adjusting to ambient
light
void checkMiddle()
{
  int difference;
  middleValue = analogRead(sensorPin); //Cehcking initial voltage on the
sensor

```

```

do{
    sensorValue=analogRead(sensorPin); //Checking second voltage on sensor
    difference= sensorValue - middleValue; //Comparing the values of the
voltage
    delay(SETTING);
}while(difference<DIFFERENCE && difference>-DIFFERENCE);
//Calculating the middle value between Logic 1 and Logic 0
middleValue = (sensorValue + middleValue)/2;
}
//Storing the encoded data into buffer
void readByte()
{
    int i;
    long input=0;
    for(i=0;i<BITLENGTHBUFFER;i++)
    { //Repeat for each bit
        sensorValue=analogRead(sensorPin); //Read the sensor Value
        if(sensorValue>=middleValue)
        {
            Buffer[i]=1;
        }
        else
        {
            Buffer[i]=0;
        }

        delay(DELAY2);
    }
}

```

Functions.h

```

//binary to decimal conversion
int readByteArray(int Array[])
{
    int i,input=0;
    for(i=0; i<8; i++)
    {
        //Binary shift to store another bit
        if (Array[i]>1)
        {
            input=input<<1;
            input|=0;
        }
        else
        {
            input=input<<1;
            input|=Array[i];
        }
    }
    return input;
}
//Decoding the received data

```

```

//comparing the consequent bits to find 1 bit of data
int Compare(int bit1, int bit2)
{
    if (bit1>bit2)
    {
        return 0;
    }
    else if (bit1<bit2)
    {
        return 1;
    }
}

//decoding 1 byte of data i.e. extracting 8 bits from 16 bits
int Byte(int buf[], int pointer)
{
    int h=0;
    int nr=0;
    int Decoded[8];
    for (int currentbit =0;currentbit<16;++currentbit)
    {
        //filtering out 8 comparisions i.e. not comparing the 2nd and 3rd bit
        if(currentbit%2==0 && pointer%2==0) // 8 comparisions
        {
            Decoded[h]=Compare(buf[currentbit+pointer], buf[1+currentbit+pointer]);
            ++h;
        }
        else if (currentbit%2==0 && pointer%2!=0)
        {
            Decoded[h]=Compare(buf[currentbit+pointer], buf[1+currentbit+pointer]);
            ++h;
        }
    }
    nr = readByteArray(Decoded); ///binary to decimal
    return nr;
}

//Finding the ID
int findID(int buf[])
{
    int i=0;
    for (i;i<(128-32);++i)
    {
        long input =0;
        for (int j=0;j<32;++j)
        {
            //Binary shift to store another bit
            input=input<<1;
            input|=buf[i+j];
        }
        if (input==1717986918)// Encoded ID value i.e. 170
        {
            break;
        }
    }
}

```

```

    return i;
}

```

Transmit.h

```

#define DELAY 10
#define LED 2 //Output pin where LED is connected
#define BITLENGTH 8 //1 Byte
#define FIRSTBIT pow(2, (BITLENGTH-1))//MSB Value
#define TRUE 1
#define TEMPERATURE A0

void manchesterOne(int T)
{
    digitalWrite(LED,LOW);
    //Serial.print("0");
    delay(T/2);
    digitalWrite(LED,HIGH);
    //Serial.print("1");
    delay(T/2);
}

void manchesterZero(int T)
{
    digitalWrite(LED,HIGH);
    //Serial.print("1");
    delay(T/2);
    digitalWrite(LED,LOW);
    //Serial.print("0");
    delay(T/2);
}

void writeByte(int decimal)
{
    //converting the decimal value to binary and sending 8 bit information from
    MSB to LSB
    int i,binary;
    for(i=0;i<BITLENGTH;i++)
    {
        binary = (int)decimal/FIRSTBIT; //Getting the first binary bit value
        decimal= (decimal & ((int)FIRSTBIT -1)); //Setting the first bit to zero
        decimal=decimal<<1; //Shift all bits by one to left
        if(binary==TRUE)
        {
            manchesterOne(DELAY);
        }
        }else
        {
            manchesterZero(DELAY);
        }
    }
}
//Serial.println();

```

}

Appendix 2- Bill of Material

The overall price for this project came to be around 45EUR, but thing to be noted here is that it includes the cost of prototyping board which is 18EUR.

Part	Qty and Price(EUR)
Development Board UNO	3 x 4/- (For Chinese, original costs 20/-)
VISHAY BPW24R Photodiode	2 x 5/-
TMP36 Temperature Sensor	1 x 1.80/-
LEDs and Resistors	1/-
Transistor BC548	2 x 0.10/-
Prototyping BreadBoard	3 x 6/-
Jumper Wires	2.22/-
Total	45.22/-