IEEE 802.15.7 Visible Light Communication: Modulation Schemes and Dimming Support

Agastya Seth — Achal Kasturia — Saumya Puri—Aditya Jain

May 2019

1 Introduction

The need to be always connected to the network and to have access to ever larger volumes of data has led to the improvement of the existing transmission technologies and the creation of new transmission techniques. One of these new techniques is the Visible Light Communication (VLC). It is based on the transmission of light pulses that belong to the visible spectrum. VLC employs a part of the electromagnetic spectrum that is currently unused for transmission purposes, this provides a huge bandwidth (390 THz). This technology is based on LEDs and photodiodes, devices which, although present for decades in the global market, are seeing only recently improved performance and wide spread. These considerations are leading many researchers to approach the VLC.

The simulation involve the On-Off Keying and the Pulse Position modulation. For this two modulations the BER (Bit Error Ratio) curves were derived. Finally, to test and get feedback with simulated and analyzed data we prepared prototype using Arduino. We built a simple model, which transmits data using a OOK modulation

2 Algorithm

2.1 OOK (On-Off Keying)

OOK is the simplest form of modulation suitable to transmit a VLC signal. Turning on and off the light is the simplest way to transmit information. OOK has carrier wave with two amplitude levels that represent the bits 0 and 1 of the modulating signal.

This modulation is simple to implement and is little affected by LED non linearity because has only two levels of amplitude. The flickering of OOK is in general low because the On-Off frequency of LED are very big (at least 200Khz). The only OOK drawback is low $\frac{1bit}{s}Hz$ spectral efficiency his because the OOK is only a one-dimensional modulation scheme. There are some different methods to support dimming in this modulation scheme, for example: redefining the “on” or “off” levels of the OOK to have a lower or higher luminous intensity

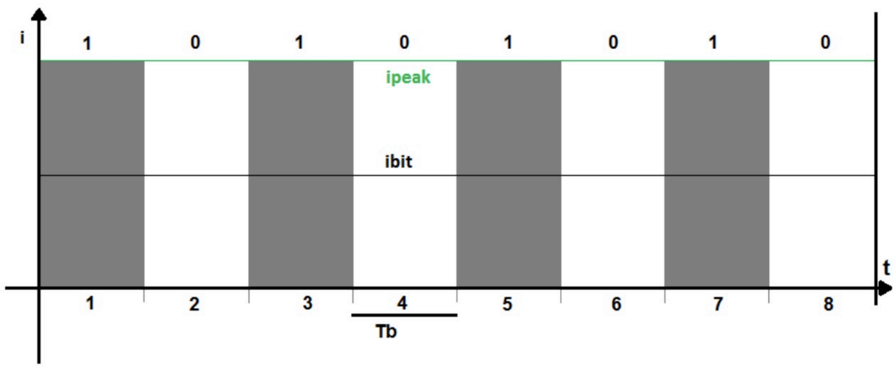


Figure 1: OOK received wave envelope. ipeak is the generated electrical current after the receiver (y axis), t is the time (x axis), ipeak is the peak wave current energy, ibit is the bit current

The electrical power and the peak and bit energy for one period is as follows:

$$E_{peak} = I_{peak}^2 T_b = 2E_b$$

The electric peak current is proportional to the luminous power received and the luminous power received is proportional to the square root of the electric wave power:

$$I_{peak} = 2RP_{avg-OOK} = 2R\sqrt{\frac{R_b E_b}{2R^2}} = \sqrt{\frac{2T_b}{E_b}}$$

$$P_{avg-OOK} = \sqrt{\frac{N_0 R_b SNR}{2R^2}} = \sqrt{\frac{R_b E_b}{2R^2}}$$

2.2 Theoretical OOK Error Probability

To evaluate the simulated BER curve one can compare the simulated BER curve with a theoretical BER curve.

The OOK plus WGN noise bit error probability is given by:

$$P_{error} = p(0)\int_{i_{th}}^{+\infty} p(i/0)di + p(1)\int_{0}^{i_{th}} p(i/0)di$$

The $p(0)$ and $p(1)$ represent the probability of obtaining respectively 0 or 1, $p(i/0)$ and $p(i/1)$ are the tail density probability of obtaining 0 instead 1 and vice versa, i_{th} represent the threshold level. The integral gives the distribution function. If one consider white Gaussian noise the tail density probability for Gaussian noise are:

$$p(1/0) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-i^2}{2\sigma^2}\right)$$

$$p(1/0) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(i-I_p)^2}{2\sigma^2}\right)$$

$$I_p = 2I_{th}$$

If symbols are equiprobable the probability of detecting one or zero are the same $p(0) = p(1) = \frac{1}{2}$ and the optimum threshold point is With some mathematical passage one can obtain:

$$P_{error} = \frac{1}{2}\int_{i_{th}}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-i^2}{2\sigma^2}\right) di + \frac{1}{2}\int_{0}^{i_{th}} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(i-I_p)^2}{2\sigma^2}\right) di =$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2}}\left(\int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di + \int_{0}^{i_{th}} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(\frac{-(i-I_p)^2}{2\sigma^2}\right) di\right) =$$

$$= \frac{1}{2\sqrt{2\pi\sigma^2}}\left(2 * \int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{i_{th}}^{+\infty} exp\left(\frac{-i^2}{2\sigma^2}\right) di\right)$$

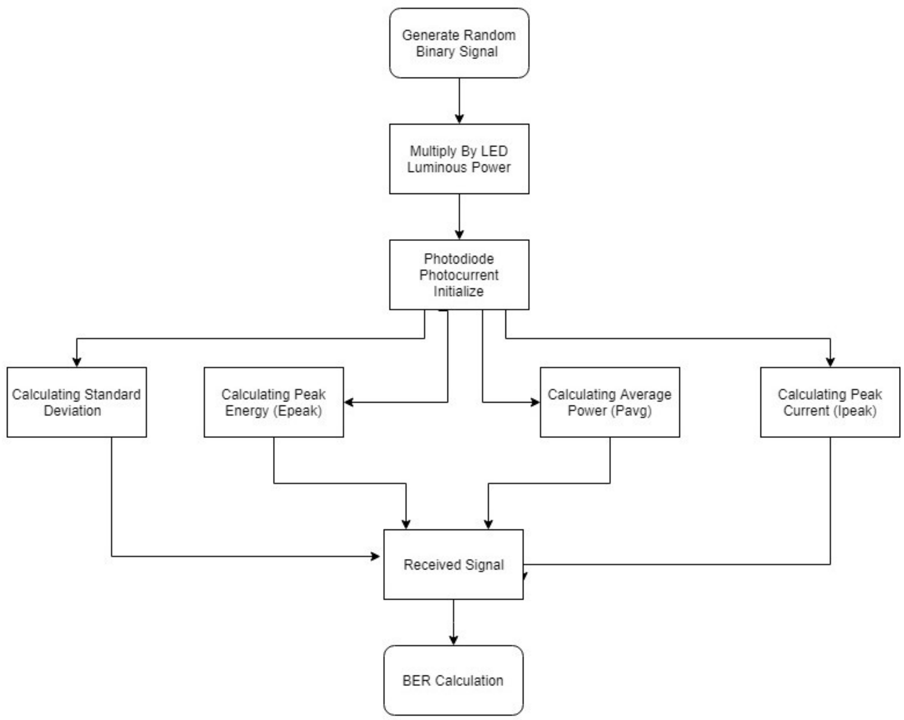
$P_{error} = Q_{func}(\frac{i_{th}}{\sigma})$

$Q_{func} = \frac{1}{\sqrt{2\pi}} \int_{x}^{+\infty} e^{\frac{-\alpha^2}{2}} d\alpha$

2.3 PPM(Pulse Position Modulation)

In L-PPM (pulse position modulation) the bit is encoded on symbol that is dived in L time slots, for each symbol one pulse is transmitted that changes its time position according to the bit that itshould represent. L-PPM has in general (except for the 2-PPM) better BER(SNR) performance respect OOK but at the cost of increased bandwidth occupation and greater system complexity. REFER FIG3

2.4 CSK



ber.jpg

Figure 2: OOK Code FlowChart

$$BER = P_{bit-error} = \frac{\frac{L}{2}}{L-1} * P_{sym-error}$$

Figure 3: L-PPM Theoretical BER

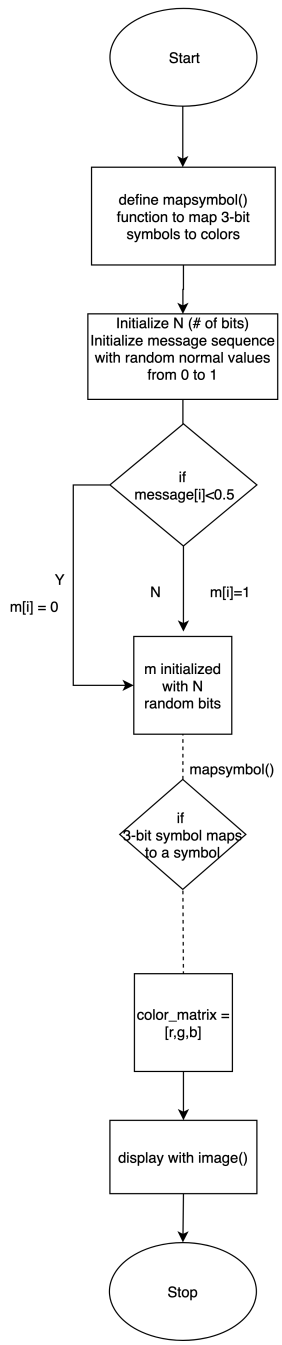


Figure 4: Flow Diagram for CSK

3 Programs

3.1 MATLAB Simulations

We simulated the Bit-Error Ratio (BER) curves of the modulation schemes provided in the paper, including On-Off Keying (OOK), Variable Pulse Position Modulation (VPPM). We also focused on simulating the Color Shift Keying (CSK) modulation scheme.

3.1.1 BER for OOK

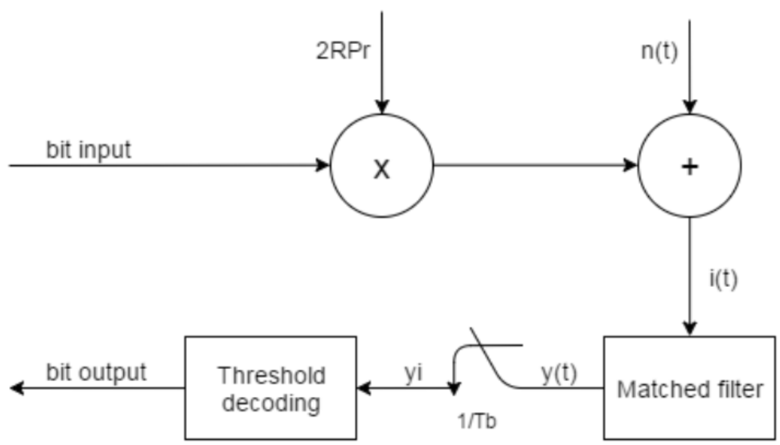


Figure 5: OOK Modulation Block Diagram

```
%% MATLAB CODE FOR BER SIMULATION OF OOK %%
%% SIGNAL GENERATED VS ORIGINAL %%
%number of bits to generate
nSignal = 1000; %Signal length
ele = 1.6e-19; %Charge of Electron
Iback = 202e-6; %Background Noise Current
%N0 = 2*ele*Iback; %Noise Spectral Density, 2*electron*
    backgcurrent
N0 = 1;
Resp = 1; %Receiver responsivity
bitRate = 10^6 %BitRate
Tbit = 1/bitRate; %one bitTime
SNR_db = 1:16; %db SNR
SNR = 10.^(SNR_db./10); % linear SNR
```

```
randombinary = rand (1,nSignal)> 0.5;   % Random Binary Signal
randombinary = randombinary *1; %transform logical input in double
for i=1:length(SNR_db) %SNR_db cycle
    Pavg(i) = sqrt((N0*bitRate*SNR(i))/(2*Resp^2)); % Luminous power
    Ipeak(i) = 2*Resp*Pavg(i); %Photodiode Current
    Epeak(i) = Ipeak(i)^2 * Tbit; %Peak current energy
    sigma(i)=sqrt(N0*Epeak(i)/2); %standard deviation after receiver
    threshold=0.5*Epeak(i); %threshold level
    for j=1 : nSignal;
        % n = normrnd(0,sigma(i));
        receivedSignal(j) = randombinary(j)*Epeak(i)+normrnd(0,sigma(i)); %matched filter output
    % bitsignal * Energy for one bit + normal distribuited noise
    end
    % receivedSignal = awgn(randombinary*Epeak(i),SNR_db(i)+3,'measured');
    % same of above cycle
    Rx = zeros(1,nSignal); %received signal inizialization
    Rx(find(receivedSignal>threshold)) = 1; %threshold detection
    [No_of_Error(i) simuBER(i)]=biterr(randombinary,Rx); %matlab function
end
theorBER = qfunc(sqrt(SNR)); %theorical formula of OOK BER
semilogy(SNR_db,theorBER,'red'); %theoretical BER graph
grid on
ylabel('BER'); xlabel('SNR (dB)');title('Bit Error Rate for Binary ');
%graph definition
hold on
semilogy(SNR_db,simuBER,'blue'); %simulation BER graph
```

3.1.2 BER for VPPM

```
%% this program calculate SLER of PPM
clc;
```

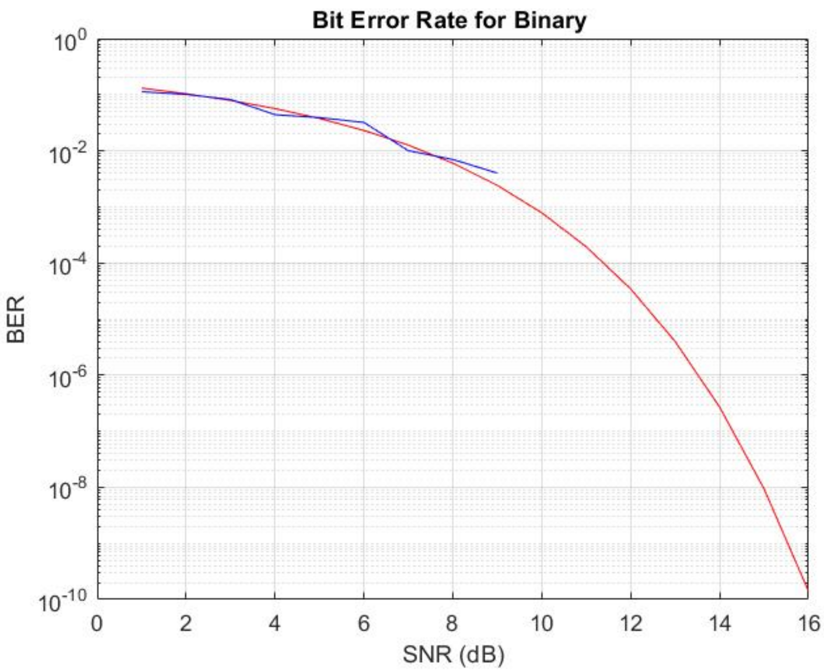


Figure 6: BER Curve for OOK

```
clear all;
%%
M=3;      %bit order
N0=1;     %noise
L=2^M;    %symbol length
nsym=200; %number of PPM symbols
Lsig=nsym*L; %total length of PPM slots
R=1;          %photoreceiver responsivity
Rsymb=1e6; %slot rate symbol
Rb = (Rsymb*M); %bitRate
Tb=1/Rb;
SNR_db= -5:0.5:16; % Energy per bit db
EsN0=SNR_db+10*log10(M); % Energy per symbol db
SNR=10.^(SNR_db./10); %Energy per bit Eb/N0
PPM=generate_PPM(M,nsym); %function to generate PPM
    signal
PPM = PPM*1; %Matlab logic signal in double
    for i=1:length(SNR_db)
            Pavg(i)=(1/L)*sqrt((((2*M)*N0*Rsymb*SNR(i))
                /(2*R^2))); %Luminous power factor (2M/L)
```

```
            Ipeak(i) = L*R*Pavg(i); %Photodiode Current
            Epeak(i) = L*M*Ipeak(i)^2 * Tb; %Peak current energy
            sigma(i)=sqrt(N0*Epeak(i)/(2)); %standard deviation after receiver
            threshold=0.5*Epeak(i); %threshold level
            for j=1 :Lsig;
                % n = normrnd(0,sigma(i));
                MF_out(j) = (PPM(j)*Epeak(i))+ normrnd(0,sigma(i)); %matched filter output
                % bitsignal * Energy for one bit + normal distribuited noise
            end
    received_PPM=zeros(1,Lsig); %generating empty PPM vector
    received_PPM(find(MF_out> threshold))=1;   %generating the received signal
    [No_of_Error(i) ser_hdd(i)]= biterr(received_PPM,PPM);
    %Matlab function to caluclate the SER
    end
semilogy(SNR_db,ser_hdd,'magenta'); %simulation BER graph
ylabel('SLER'); xlabel('SNR (dB)');
title([num2str(L),'-PPM SlotErrorRate']);
grid on
grid minor
hold on;
% theoretical calculation
Pse_ppm_theor=qfunc(sqrt(M*2^M*0.5*SNR)); %transform SLER to SER
semilogy(SNR_db,Pse_ppm_theor,'green','linewidth',0.1);
%theoretical BER graph
```

3.1.3 Function to generate PPM

```
function [PPM]=generate_PPM(M,nsym)
    % function to generate PPM% 'M' bit order
    %'nsym': number of PPM symbol to generate
    PPM=[]; %PPM array empty inizialization
    for i= 1:nsym %cycle from 1 to number of symbol,every cycle generate one symbol
        bitSig=rand(1,M)> 0.5; % random binary number
        dec_value=bi2de(bitSig,'left-msb');   %converting bit to decimal value
```

```
        tempPPM=zeros(1,2^M); %zero sequence of length 2^M
        tempPPM(dec_value+1)=1; %placing a pulse accoring to decimal value,
        %matlab index start from 1 and not from 0, so need to add 1;
        PPM=[PPM tempPPM];
        %put tempPPM in array queue
        %PPM symbolend
        %close for cycle
    end
```

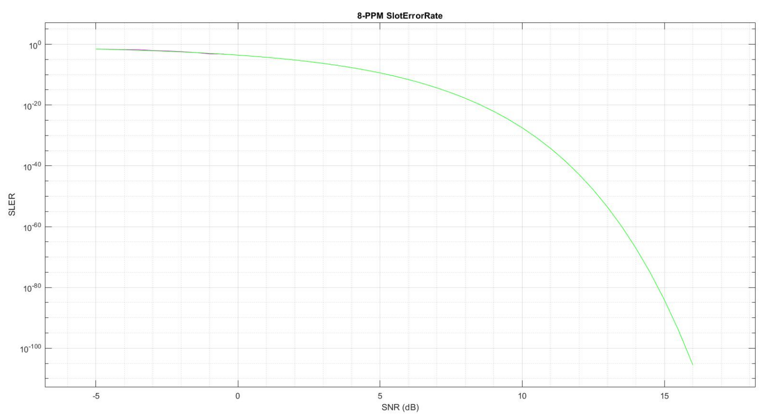


Figure 7: BER Curve for 8-VPPM

3.1.4 CSK Modulation

```
%% Clearing Workspace

clc;
clear all;
close all;

N=300;  % Number of Bits
cnt = 1;
m=rand(1,N); % Message Signal

```

```
%% Creating a random sequence of N-bits
for i=1:N
 t=[t1:(Tb/100):t2];
 if m(i)>0.5
m(i)=1;
 m_s=ones(1,length(t));
 invm_s=zeros(1,length(t));
 else
m(i)=0;
 m_s=zeros(1,length(t));
 invm_s=ones(1,length(t));
 end

end

%% Plotting the CSK Colors using the Image() function
m_symbol = [];
for i=1:3:length(m)
    m_symbol = [m_symbol;m(i:i+2)];
end

colors = zeros(length(m_symbol),3);
for i=1:size(m_symbol,1)
    [r,g,b] = mapsymbol(m_symbol(i,:));
    colors(i,:) = [r g b];
end

colors_norm = colors./255;
map = zeros(length(colors_norm),1,3);
map(:,1,:) = colors_norm(:,:);
image(map);

%% Spectrum test - Plotting all the symbol mappings
test_message = [[0 0 0];[0 0 1];[0 1 0];[0 1 1];[1 0 0];[1 0 1];[1 1 0];[1 1 1];[0 0 0]];
colors = zeros(length(test_message),3);
for i=1:size(test_message,1)
    [r,g,b] = mapsymbol(test_message(i,:));
    colors(i,:) = [r g b];
end

colors_norm = colors./255;
map = zeros(length(colors_norm),1,3);
map(:,1,:) = colors_norm(:,:);
image(map);
```

Frequency band	Spectral width (nm)	Color	Proposed Code
380-450	70	pB	000
450-510	60	B,BG	001
510-560	50	G	010
560-600	40	yG,gY,Y,yO,O	011
600-650	50	rO	100
650-710	60	R	101
710-780	70	R	110
		Reserved	111

Figure 8: Freq Division

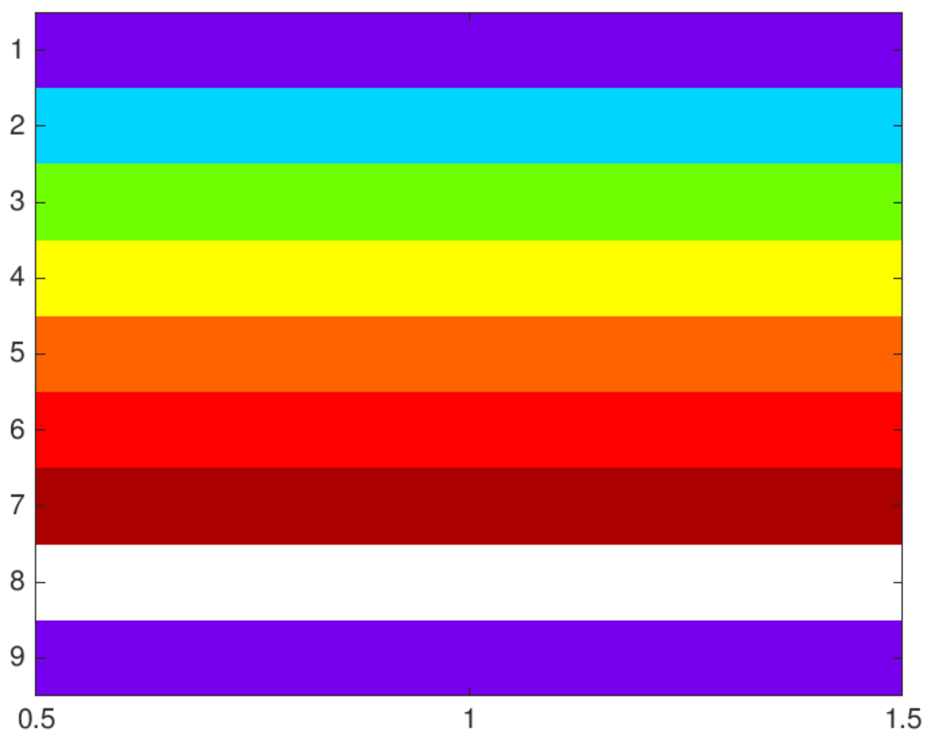


Figure 9: Spectrum of all the symbol mappings.

[b]0.2

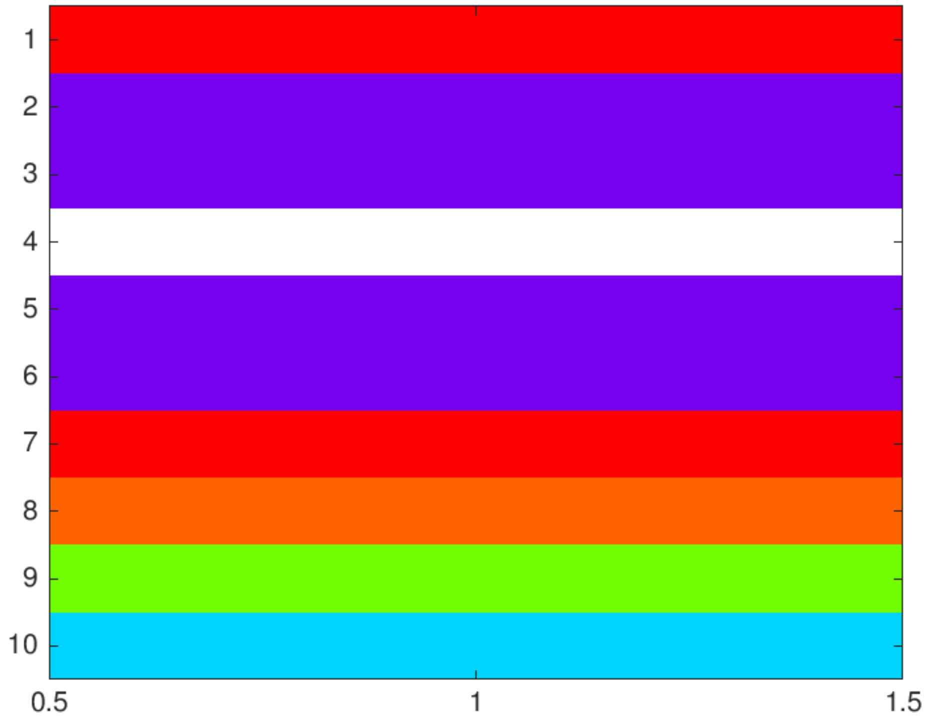


Figure 10: Simulation for N=30

[b]0.2

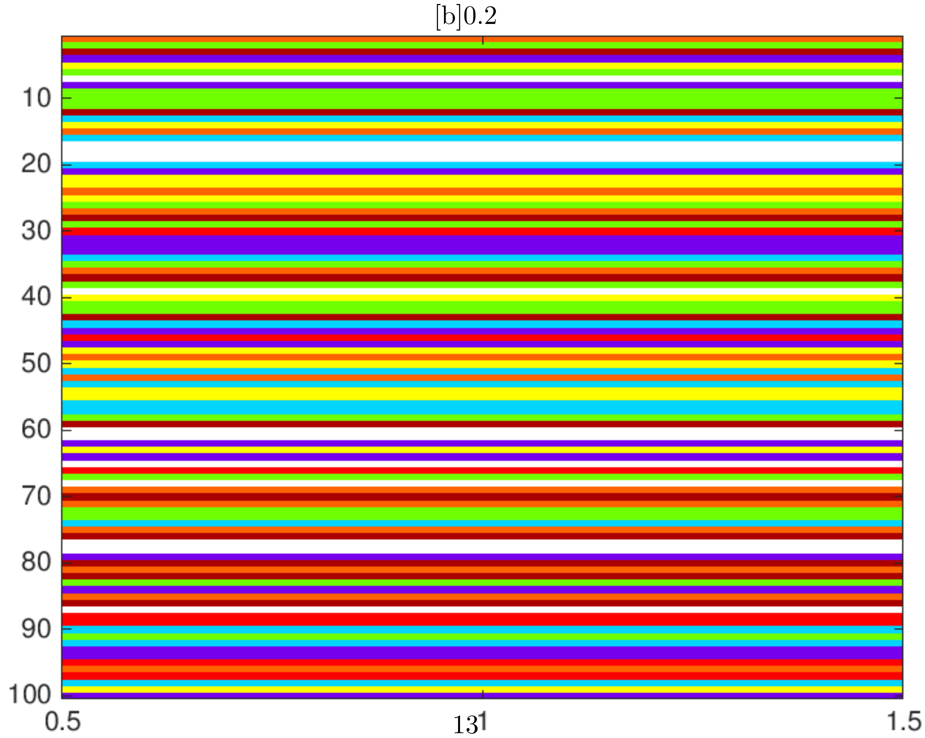


Figure 11: Simulation for N=300

Figure 12: Simulating CSK

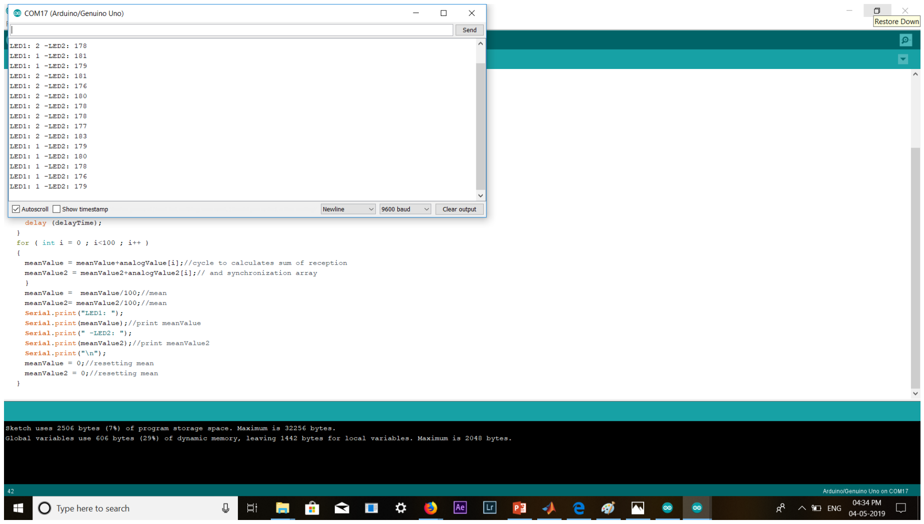


Figure 13: LED Threshold O/P

3.2 Arduino Simulations

3.2.1 Threshold Calculation Transmitter

The threshold values of the receiver photodiode are calculated to distinguish between logical 1 and 0.

```
int ledPin = 9;  //definition of transmission LED pin
int ledPin2 =10;
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(ledPin,OUTPUT); //LED transmission pin as output
pinMode(ledPin2,OUTPUT); //LED transmission pin as output
digitalWrite(ledPin,HIGH); //LED transmission on (this
    value can be changed to calculates other threshold
    conditions)
digitalWrite(ledPin2,HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

3.2.2 Threshold Calculations Receiver

```
int analogValue[100];//vector to calculate photoreceiver reception mean
int analogValue2[100];//vector to calculate photoreceiver synchronization mean
int photoPin = A0;//defining pin of reception pd
int photoPin2 = A1;//defining pin of synchronization pd
int delayTime = 10;//delaytime between reception
int meanValue2 = 0;//mean of synchronization pd
int meanValue = 0;//mean of reception pd

void setup() {
  Serial.begin(9600);
  pinMode(photoPin,INPUT);//pin of pd reception defining input
  pinMode(photoPin2,INPUT);//pin of pd synchronization defining input

}

void loop() {
  for(int i = 0;i<100;i++)
  {
    analogValue[i] = analogRead(photoPin);//cycle to read 100 values of pd reception
    delay (delayTime);
  }
  for(int i = 0;i<100;i++)
  {
    analogValue2[i] = analogRead(photoPin2);//cycle to read 100 values of pd synchronization
    delay (delayTime);
  }
  for ( int i = 0 ; i<100 ; i++ )
  {
    meanValue = meanValue+analogValue[i];//cycle to calculates sum of reception
    meanValue2 = meanValue2+analogValue2[i];// and synchronization array
    }
    meanValue =  meanValue/100;//mean
    meanValue2= meanValue2/100;//mean
    Serial.print("LED1: ");
    Serial.print(meanValue);//print meanValue
```

```
    Serial.print(" -LED2: ");
    Serial.print(meanValue2);//print meanValue2
    Serial.print("\n");
    meanValue = 0;//resetting mean
    meanValue2 = 0;//resetting mean
  }
```

3.2.3 1-Bit Transmitter Program

A program which transmits a bit through VLC using OOK Modulation scheme is written and executed.

```
int ledPin = 9;   //Led pin
int delayTime = 100;  //delay time
char value;    //handy value to insert one bit
String values;   //bit string to transmit
int length;    //handy value to memorize the length of the transmission string
boolean shouldSend;
void setup() {
  // put your setup code here, to run once:
pinMode(ledPin, OUTPUT);   //define the LED pin as output
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
 while (Serial.available())  //if the string is available start the while cycle
 {
  //here the user should insert a binary string
  value = Serial.read();   //put in value one char from serial monitor
  // Build string
  values = values + String(value);  //create a string composed by each char values
  // Loop
  delay(1);    //wait 1ms
  }
  if (values != "")   //if the values string isn t empty
  {
    Serial.println(values);  //print the bits string
    length = values.length();   //acquire the length of the string
    shouldSend = true;  //Set the sent value true
    }
```

```
    if (shouldSend)    //if shouldSend true
    {
      //start LED ON-OFF cycle
      for (int i = 0; i < length; i++)  //cycle the String character
      {
        if (values.charAt(i) == '0') //if the character is 0
        {
          digitalWrite(ledPin, LOW); //turn OFF led
          delay(delayTime);    //wait delaytime
          }
          else if (values.charAt(i) == '1') //if character is 1
          {
            digitalWrite(ledPin, HIGH);  //turn ON led
            delay(delayTime);  //wait delaytime
            }
            }
            }
            values = "";  //after the string transmission reset the string
            shouldSend = false;  //set the shouldSend false, in that way the program wait until other bits string is ready     }
        }
```

3.2.4 1-Bit Receiver Program

A program which transmits a bit through VLC using OOK Modulation scheme is written and executed.

```
int photoPin = A0;// photodiode receiver pin
int delayTime = 100; // delay time in ms
int analogValue;//handy variable to memorize the receiver value
int lowThreshold = 60;//threshold levelcalculated in C.1
char value;//handy value to memorize on bit
String values;//handy value to memorize bits
void setup() {
  Serial.begin(9600);
  pinMode(photoPin, INPUT); //define photopin as input
  }
```

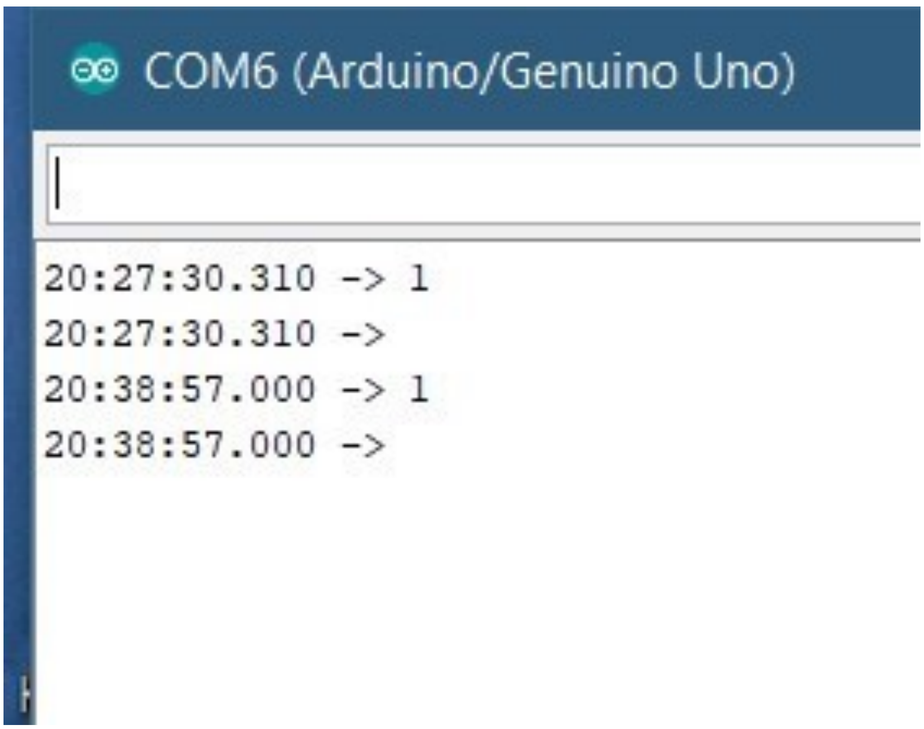


Figure 14: 1-bit Transmitter Input

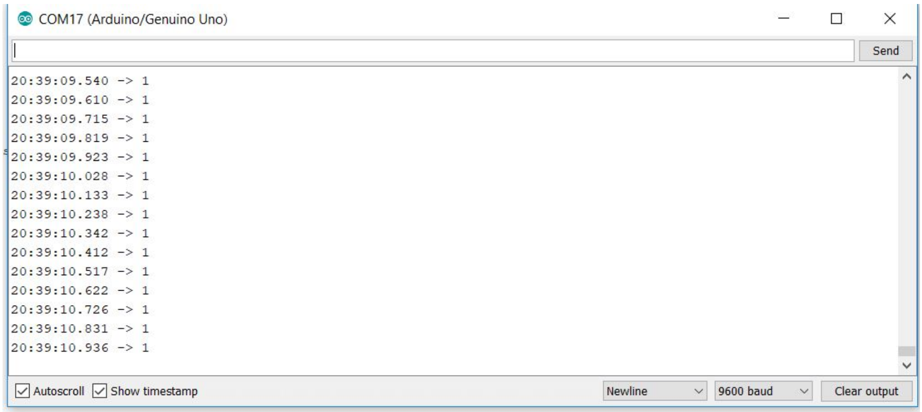


Figure 15: 1-bit Receiver Output

```
void loop() {
     analogValue = analogRead(photoPin);//acquire analog photoPin value
     if (analogValue < lowThreshold) // if photoPin value < lowthreshold set 0
     {value = '0'; }
     else     //else set 1
     {
     value = '1';
     }
     values = String(value);  //save binary value in values string
     if (values != "")          //if the string values is not empty
     {                          // if there is transmission data
       Serial.print(values); // print on serial
       Serial.print("\n");
       }
       delay(delayTime);//attend delay Time after every acquisition
     }
```

3.2.5 RS-FEC RLL Encoding

```
%%
```

```
clear all;
close all;
clc;

%% Reed Solomon Codes

% The page describes RS Encoder based on following polynomials.
% Code generator polynomial:
% g(x) = (x + ?^0)(x + ?^1)(x + ?^2)...(x + ?^(2T−1)), ? = 02(hex)
% Field Generator polynomial:
% p(x) = x^8 + x^4 + x^3 + x^2 +1

%% RS(8,4,t)
input_data=([1 0 1 1]); %binary input data
k=length(input_data); %no. of data bytes before encoding
n=k+4; %no. of data bytes after encoding
t=(n−k)/2; %no. of data bytes which can be corrected by RS encoder and decoder combination


g_f=0;
g_f(2)=1;
for i=1:7
g_f(i+2)=bitshift(g_f(i+2−1),1); %creates an array by multiplying 2 to the previous value
% i.e. creates [0 1 2 4 8 16 32 64 128]
end
x=i+2; %total number of elements in g_f
x1=2;x2=4;x3=5;x4=6;
%%
for i=1:247
g_f(x+i)=bitxor(bitxor(g_f(x1),g_f(x2)),bitxor(g_f(x3),g_f(x4))); % performing bitwise
% XOR for various inputs
x1=x1+1;
x2=x2+1;
x3=x3+1;
x4=x4+1;
end

%code generator polynomial
r = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
for j=2:17
for i=1:256
if(r(j)==g_f(i))
```

```
indx(j-1)=i-2;
break
end
end
end

lfsr_op(1:16)=0;

for dc=1:k
lfsr_input=bitxor(lfsr_op(1),input_data(dc));

%get index of data with respect to Galois Field
for i=1:256
if(lfsr_input==g_f(i))
lfsr_input=i-2;
break
end
end
temp1=0;
reg_in=2+mod((indx+lfsr_input),255);
for j=16:-1:1
temp=lfsr_op(j);
if(lfsr_input==-1)
lfsr_op(j)=bitxor(0,temp1); %lfsr_op contains the redundant bits being
% added to the original data
else
lfsr_op(j)=bitxor(g_f(reg_in(j)),temp1);
end
temp1=temp;
end
end

encoded_data_output=[lfsr_op(1:(n-k)) input_data]; % concatenating the redundant bits with the
% input data to be corrected later

% Manchester Encoding
out = manchester(encoded_data_output,2); %manchester(bits,bitrate)
display(out); %manchester encoded output

%Plotting the Manchester code for the Reed Solomon-
%Forward Error Corrected/Encoded output
figure;
[b,s] = manchester(encoded_data_output,2);
```

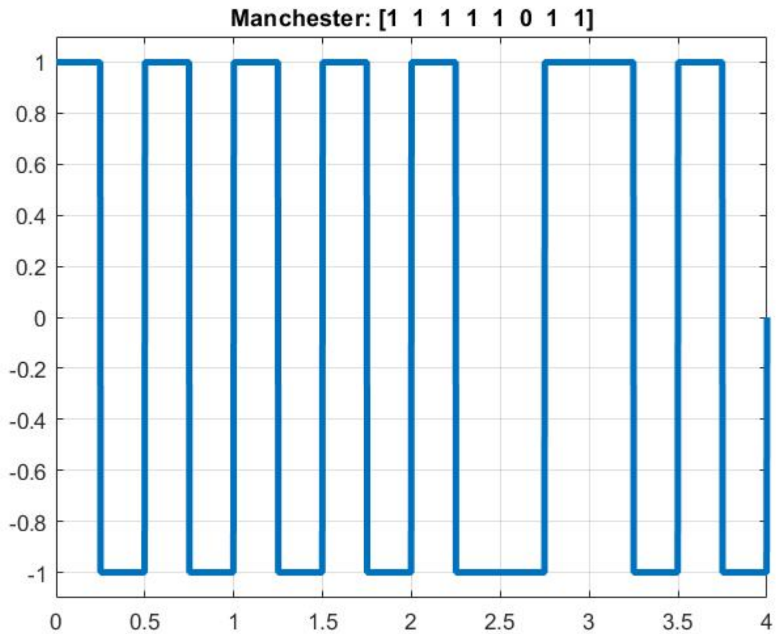


Figure 16: RS-FEC,Manchester Coding

```
plot(b,s,'LineWidth',3);
axis([0 b(end) -1.1 1.1])
grid on;
title(['Manchester: [' num2str(encoded_data_output) ']'])
    ;
```

3.2.6 Manchester Coding

```
function [t,x] = manchester(bits, bitrate)
% MANCHESTER Encode bit string using Manchester code.
% [T, X] = MANCHESTER(BITS, BITRATE) encodes BITS array using Manchester
% code with given BITRATE. Outputs are time T and encoded signal values X.

T = length(bits)/bitrate; % full time of bit sequence
n = 200;
N = n*length(bits);
dt = T/N;
t = 0:dt:T;
x = zeros(1,length(t)); % output signal

for i = 0:length(bits)-1
  if bits(i+1) == 1
    x(i*n+1:(i+0.5)*n) = 1;
    x((i+0.5)*n+1:(i+1)*n) = -1;
  else
    x(i*n+1:(i+0.5)*n) = -1;
    x((i+0.5)*n+1:(i+1)*n) = 1;
  end
end
```

4 Learning Outcomes

4.1 Saumya Puri:

Our research paper describes the IEEE 802.15.7 standards for Visible light communication (VLC). Visible light communication is a branch of optical wireless communication that involves electromagnetic waves in the visible spectrum. We try to make use of such electromagnetic waves to stay connected to the network and to have access to more data that is useful for us. All transmission systems use waves to communicate. In particular radio and optical communication use electromagnetic waves to transport information. The electromagnetic spectrum is very broad and is categorized by wavelength or by frequency. The visible part of the electromagnetic spectrum is currently not in much use. However, it has a wide scope and is continuously spreading. In general VLC uses commercial LEDs, photodiodes etc to communicate, which have recently improved performance. For indoor systems, the most commonly used modulation for VLC is currently the ON-OFF Keying (OOK) and the Pulse Position Modulation (PPM), in particular, Variable Pulse Position Modulation (VPPM). Colour

Shift Keying (CSK) can also be used. These three modulation schemes discussed in the paper have their own advantages and disadvantages. ON-OFF Keying (OOK) is the simplest form of modulation suitable to transmit a VLC signal. Turning the light on and off is the simplest way to transmit information. OOK has a carrier wave with two amplitude levels that represent the bits 0 and 1 of the modulating signal. Variable Pulse Position Modulation (VPPM) just like OOK, it is one of the simplest and intuitive form of modulation to transmit a signal in VLC. PPM, Pulse Position Modulation is based on the position of the pulse inside the symbol. Colour Shift Keying (CSK) is similar to FSK (Frequency Shift Keying) for radio transmission because changing the frequency means changing the wavelength. The base band signal is modulated with carrier waves of many wavelengths. The colour, in the name of this modulation, is referred to the wavelength of the carrier. Changing the wavelength means changing the colour of the carriers. We have implemented the following in our project:

Arduino Prototype: Using an Arduino setup, we were able to implement OOK using LEDs for transmission and photodiodes for reception. The threshold for the LEDs had been calculated and using those threshold values, the photodiode was made to distinguish between the different bits i.e. 0 and 1. The bits were successfully transferred from one system to another. However, we experienced some delay in the communication. This was mainly because the delays of the diodes were unknown and there was a lack of proper synchronisation. For such types of communication, we need proper hardware i.e. proper LEDs (with known and lesser delays) and photodiodes to obtain correct results. As, the hardware components were limited, we could not arrange for a better working setup. We also tried to implement the transmission of bits consecutively to obtain ASCII values using 8-bit binary data. However, we were not successful in doing so because of the same reason i.e. lack of synchronisation amongst the components and unexpected, large delays. BER calculation: We tried to implement the modulation schemes by calculating the BER (Bit-Error Ratio) for OOK and VPPM, which is the ratio of the number of bits wrongly transmitted to the total number of bits. The bit stream usually gets altered due to noise, interference or distortion. The BER should be low for successful data transmission. We were able to obtain the ideal BER curve, which was calculated using the theoretical proof of OOK bit error rate. A random signal was generated and received whose BER was compared to the ideal BER curve. According to our observations from the implementation, PPM has in general (except 2-PPM), a better BER (SNR) performance with respect to OOK. There is however, a trade-off between the cost of increased bandwidth occupation and greater system complexity.

CSK Modulation Scheme Implementation: CSK modulation was also implemented on MATLAB to understand its' working. Different CSK symbols were mapped to their particular RGB values according to wavelengths determined by frequency bands. The CSK symbols were then plotted by taking various random signals to understand the concept behind CSK waveforms and also to know how the colours of the LEDs are decided. However, we could not assign intensity variations.

VPPM Transmitter Block Coding with Reed Solomon Forward Error Correction (RS FEC Encoding): The input bits are taken in by the VPPM transmitter block and then RS FEC encoded. We also applied Manchester encoding to minimize the error rate. Reed Solomon (RS) codes are basically used for providing FEC (Forward Error Correction) to the received erroneous blocks. RS encoder is used at the transmit end by which redundancy bytes are added to the input data to be transmitted. RS decoder is used at the receiver to correct the errors occurred on the way by use of redundant information. RS codes are systematic linear block codes. It is a block code because the code is put together by splitting the original message into fixed length blocks. A Reed-Solomon code is specified as RS (n, k) with s-bit symbols. This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are n-k parity symbols of s bits each. A Reed-Solomon decoder can correct up to t symbols that contain errors in a codeword, where 2t = n-k.

4.2 Achal Kasturia

Our research paper describes the IEEE 802.15.7 standards for Visible light communication (VLC) .

VLC refers to short-range optical wireless communication using the visible light spectrum from 380 to 780 nm. VLC transmits data by intensity modulation optical sources, such as light emitting diodes (LEDs) which were also used by us to implement a hardware solution of one of the modulation schemes - OOK as follows;

ARDUINO PROTOTYPE: We calculated the threshold of the LEDs using codes 3.2.1 3.2.2 and then used those threshold values to help the photodiode connected to the receiver to distinguish between logical 1 and 0.We had transmitted 1 bit values using the setup stated. We also attempted to transmit ASCII values. PROBLEMS FACED: For implementing hardware solutions, we required proper components for Visible Light Communication , specific LEDs which performed this function best and Photodiodes which had a lesser delay. We couldn't obtain the hardware required due to some factors. We had to implement the consecutive transmission of bits to transmit and receive ASCII values using 8-bit binary data.Which could not be completed due to the unknown delay of the hardware components and also the lack of synchronization components present with us (After a lot of running around , we were able to find 2 photodiodes to implement the hardware part, we couldn't find other components from the labs)

BER CALCULATION: The BER was calculated for OOK AND LPPM The number of bit errors is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion or bit synchronization errors. The bit error rate (BER) is the number of bit errors per unit time. The ideal BER curve is calculated using the theoretical proof of OOK bit error rate and the a random signal was generated and received whose BER was compared to the PPM has in general (except for the 2-PPM) better BER(SNR) performance respect OOK but at the cost of increased bandwidth occupation and greater system complexity.

CSK IMPLEMENTATION: We mapped the CSK symbols to particular RGB values according to the wavelengths which were determined by the frequency bands and plotted them by taking different random signals to visualize how a CSK waveform is made and how the colour of the LEDs is decided.

PROBLEMS FACED: 1.We didn't have the hardware required to implement CSK.As a proper photodiode which could detect different colours and their intensity at reception as that is required for the correct implementation of CSK.

VPPM Transmitter Block Coding: The transmitter block of the VPPM takes the input data bits and uses the RS-FEC encoding and RLL encoding for which we have chosen Manchester Encoding. The chief advantage of Manchester encoding is the fact that the signal synchronizes itself. This minimizes the error rate and optimizes reliability Forward error correction (FEC) is a digital signal processing technique used to enhance data reliability. It does this by introducing redundant data, called error correcting code, prior to data transmission or storage. This allows broadcasting data to be sent to multiple destinations from a single source. A Reed-Solomon code is specified as RS(n,k) with s-bit symbols. This means that the encoder takes k data symbols of s bits each and adds parity symbols to make an n symbol codeword. There are n-k parity symbols of s bits each. A Reed-Solomon decoder can correct up to t symbols that contain errors in a codeword, where 2t = n-k.

4.3 Aditya Jain

Our project consists of three modulation schemes named OOK, VPPM and CSK for making visible light communication viable by eliminating Flicker Mitigation and by providing Diming Support. –For OOK we made an Arduino Prototype with the help of an LED and a Photodiode. We were able to send bits using LED and receive the same value at receiver end. We used two arduinos for this purpose, one as transmitter and one as receiver. We uploaded the Tx and Rx codes in both Arduino respectively. With LED on we were getting "1" at the receiver and with LED off we were getting "0" at the receiver (as observed in Serial Monitor output). Problems while implementing Arduino prototype- We were supposed to use another set of LED and photodiode for synchronization purpose , but it was not working at all, the output was generating some random characters due to lack of synchronization. Sometimes the normal bits were received wrong , due to environment interference. –We implemented Reed-Solomon error correction , the Reed-Solomon encoder takes a block of digital data and adds extra "redundant" bits. Errors occur during transmission or storage for a number of reasons (for example noise or interference). The Reed-Solomon decoder processes each block and attempts to correct errors and recover the original data. The number and type of errors that can be corrected depends on the characteristics of the Reed-Solomon code. Even if we end up losing some of the data bits Reed Solomon makes sure that we can still recover the data using the parity bits we added previously. –Color-shift keying (CSK) is a visible light communication intensity modulation scheme, that transmits data imperceptibly through the variation of the color emitted by red, green, and blue light emitting diodes. An advantage of CSK is that the power envelope of the transmitted signal is fixed; therefore, CSK reduces the potential for human health complications related to fluctuations in light intensity. Here we implemented CSK transmitter on MATLAB, we took 8 different colors to represent 8 different symbols. We represented 100 symbols (each of 3 bit) and code assigns one out of 8 colors we defined in the code earlier.

4.4 Agastya Seth

The research paper we chose described the IEEE 802.15.7 standard for Visible Light Communication. Visible Light Communication has been a topic of research for some years now, since given the exhaustive nature of radio wave bandwidth, the need for an alternate communication medium is inevitable.

Our main inclination to pick this topic was thus to explore the world of visible light communication, and get some hands-on work experience with it. There were mainly 3 modulation schemes described in the paper: OOK, VPPM and CSK. For our midterm review, we visited briefly the concept of Bit-Error-Ratio (BER), specifically for the OOK modulation. This helped us realize the limitations and nature of the channel we were dealing with. For most of our software simulations, since the topic is still under research, we were unable to model the channel.

This urged us to find avenues in hardware implementations. Since one of the biggest strength of VLC is the minimal hardware support required. We were actually able to build a proof-of-concept model for the simplest modulation scheme OOK, by using as little as an Arduino, a few IR LEDs and some photodiodes. Granted, IR LEDs weren't the most realistic option for this, but since we required a special visible light photodiode, which we weren't able to procure, this was probably the next best thing.

One of my main contributions to the project involved writing the MATLAB simulation program for Color Shift Keying (CSK), which is one of the modulation schemes discussed in the paper. The use of such multi-color LEDs forms the principle behind CSK modulation. Color shift keying modulation is similar to frequency shift keying in that the bit patterns are encoded to color (wavelength) combinations. For example, for 4-CSK (two bits per symbol) the light source is wavelength keyed such that one of four possible wavelengths (colors) is transmitted per bit pair combination. In order to define various colors for communication, the IEEE 802.15.7 standard breaks the spectrum into 7 color bands in order to provide support for multiple LED color choices for communication. The CSK signal is generated by using three color light sources out of the seven color bands. The three vertices of the CSK constellation triangle are decided by the center wavelength of the three color bands on xy color coordinates. CSK has the following advantages:

1. The final output color (e.g., white) is guaranteed by the color coordinates. CSK channels are determined by mixed colors that are allocated in the color coordinates plane.
2. The total power of all CSK light sources is constant, although each light source may have a different instantaneous output power. CSK dimming ensures that the average optical power

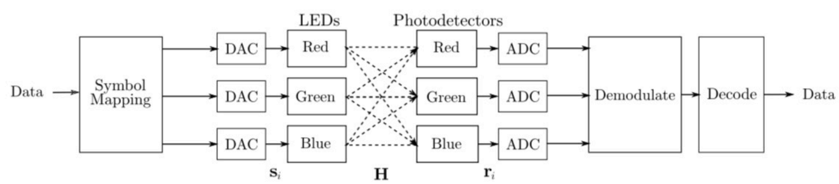


Figure 17: Functional block diagram of the VLC channel that supports the CSK modulation format.

As far as the future of the technology is concerned, several technical challenges must be addressed to realize the full potential of VLC technology. First, channel models for VLC are not well under- stood, especially for outdoor non-line-of-sight (NLOS) environments, and there is an active area of research for channel models and platforms for VLC. Also, the networking of the light sources and upgrading current infrastructures to support communication is another challenge, which requires support from the lighting industry. With continued growth in LED based light sources and the need for multi-Gb/s data distribution, VLC, being developed as a global industry standard in IEEE 802.15.7, promises to be a very attractive candidate as a future high data rate and power-efficient technology. The research paper we chose described the IEEE 802.15.7 standard for Visible Light Communication. Visible Light Communication has been a topic of research for some years now, since given the exhaustive nature of radio wave bandwidth, the need for an alternate communication medium is inevitable.

Our main inclination to pick this topic was thus to explore the world of visible light communication, and get some hands-on work experience with it. There were mainly 3 modulation schemes described in the paper: OOK, VPPM and CSK. For our midterm review, we visited briefly the concept of Bit-Error-Ratio (BER), specifically for the OOK modulation. This helped us realize the limitations and nature of the channel we were dealing with. For most of our software simulations, since the topic is still under research, we were unable to model the channel.

This urged us to find avenues in hardware implementations. Since one of the biggest strength of VLC is the minimal hardware support required. We were actually able to build a proof-of-concept model for the simplest modulation scheme OOK, by using as little as an Arduino, a few IR LEDs and some photodiodes. Granted, IR LEDs weren't the most realistic option for this, but since we required a special visible light photodiode, which we weren't able to procure, this was probably the next best thing.

One of my main contributions to the project involved writing the MATLAB simulation program for Color Shift Keying (CSK), which is one of the modulation schemes discussed in the paper. The use of such multi-color LEDs forms the principle behind CSK modulation. Color shift keying modulation is similar to frequency shift keying in that the bit patterns are encoded to color (wavelength) combinations. For example, for 4-CSK (two bits per symbol) the light source is wavelength keyed such that one of four possible wavelengths (colors) is transmitted per bit pair combination. In order to define various colors for communication, the IEEE 802.15.7 standard breaks the spectrum into 7 color bands in order to provide support for multiple LED color choices for communication. The CSK signal is generated by using three color light sources out of the seven color bands. The three vertices of the CSK constellation triangle are decided by the center wavelength of the three color bands on xy color coordinates. CSK has the following advantages:

1. The final output color (e.g., white) is guaranteed by the color coordinates. CSK channels are determined by mixed colors that are allocated in the color coordinates plane.
2. The total power of all CSK light sources is constant, although each light source may have a different instantaneous output power. CSK dimming ensures that the average optical power

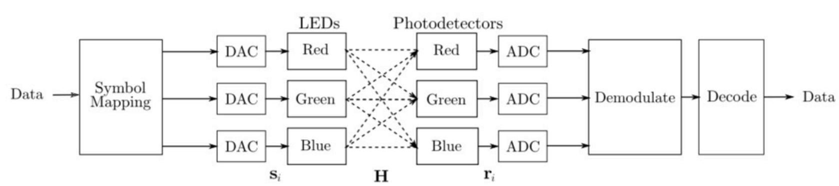


Figure 18: Functional block diagram of the VLC channel that supports the CSK modulation format.

As far as the future of the technology is concerned, several technical challenges must be addressed to realize the full potential of VLC technology. First, channel models for VLC are not well under- stood, especially for outdoor non-line-of-sight (NLOS) environments, and there is an active area of research for channel models and platforms for VLC. Also, the networking of the light sources and upgrading current infrastructures to support communication is another challenge, which requires support from the lighting industry. With continued growth in LED based light sources and the need for multi-Gb/s data distribution, VLC, being developed as a global industry standard in IEEE 802.15.7, promises to be a very attractive candidate as a future high data rate and power-efficient technology.

5 Conclusion

5.1 CSK Modulation

We simulated the CSK Modulation Scheme on MATLAB, with the intention of observing the different colors the LED would output. Using the chromaticity mapping given in the paper, we created a function $'mapsymbol()'$ to map the various symbols to their color wavelength in 8-CSK modulation. The MATLAB program generates a random N-bit sequence, which is then formatted in symbols of 3-bits each. This is then mapped to their respective color maps and then finally displayed using the in-built $'Image()'$ function.

The takeaway from this experiment was that we were able to visualize the working of CSK Modulation. This code can be used as the modulation block given in the paper. This could also take various color spectrum mid-points (of the 7 given in the paper) to simulate different base frequency (color).

5.2 Arduino Prototype

A good cheap and easy solution to implement a VLC prototype is presented. This solution consists in the use of two Arduino micro controllers: one to implement the LED transmission and second to implement the photo-receiver reception.The modulation used to transmit/receive signal is the OOK.For these experiments we calculated the threshold level using the two Arduino.We were able to transmit the first bits but we faced problems in transmitting continuous bits for transmitting ASCII values.The reason for these problems was mainly synchronisation of the Transmitter and Receiver which we weren't able to do due to lack of proper hardware(Photodiodes and VL LEDs) also, the delay of the LEDs and the photodiodes were unknown.

5.3 BER Simulations

The modulation block consist taking the bit value and multiply that by the led luminous power. On the reception block the photodiode generates a photocurrent proportional to the transmitted luminous power plus white Gaussian noise n(t), both proportional to R the responsivity of photodiode. The ideal device used for reception of OOK signal with WGN noise is matched filter followed by threshold detector, the matched filter performs a convolution integration to maximize the SNR.

As the models are derived and knowing the theoretical BER curves, we were able to test the OOK and PPM model using this parameter.

5.4 RS-FEC RLL Blocks Coding

The transmitter block displayed in which the input bits were first RS-FEC encoded and then RLL encoded was simulated and the output waveform was displayed.

6 Future Applications

6.1 Hybrid system (Radio + VLC)

The thought of replacing all radio communications with VLC is impracticable, just think the shadowing problem that affects light transmission. However VLC has many advantages with respecttoradio transmission (chapter 1),in particular for indoor environments. The most popular solution is to support the radio Wi-Fi with VLC: if VLC signal isn't present the radio connection can replace the VLC and viceversa.The system must be able to select the best solution at the right moment.

6.2 Sites for VLC (Hospital, Airplane, underwater)

VLC can be used safely in hospital where the radio transmission can interfere with the hospital equipment. For the same reason VLC can be used on airplanes. Underwater the radio waves do not propagate far this because the high attenuation coefficient. The light, instead, propagates very easily through the water, for that, VLC could be a great technology to transmit underwater signals.

6.3 High density wireless

High density wireless is a wireless with high energy density to transmit more fast and more amount of information. A good example is university classroom: if every students using the wireless at same time the network should probably collapse. With VLC this issue can be avoided.High density wireless is difficult to realize with radio waves because if one increases the radio waves energy this can becomes dangerous for human safety, furthermore as we say before the cheapness and simplicity of VLC system make them more attractive for built an high density wireless.

6.4 Smart drive/shop/city.

The car, the semaphore, the publicillumination and private shop signs are all sources of lights, even more in most case that lights are generated from LEDs sources. A car, a semaphore, a public illumination system and private shop signs are all sources of lights: moreover in most case that lights are generated from LEDs sources. With that premise one can consider to use that sources to illuminate and transmit information. In that way many solutions are possible to imagine.

1 Agastya Seth

The research paper we chose described the IEEE 802.15.7 standard for Visible Light Communication. Visible Light Communication has been a topic of research for some years now, since given the exhaustive nature of radio wave bandwidth, the need for an alternate communication medium is inevitable.

Our main inclination to pick this topic was thus to explore the world of visible light communication, and get some hands-on work experience with it. There were mainly 3 modulation schemes described in the paper: OOK, VPPM and CSK. For our midterm review, we visited briefly the concept of Bit-Error-Ratio (BER), specifically for the OOK modulation. This helped us realize the limitations and nature of the channel we were dealing with. For most of our software simulations, since the topic is still under research, we were unable to model the channel.

This urged us to find avenues in hardware implementations. Since one of the biggest strength of VLC is the minimal hardware support required. We were actually able to build a proof-of-concept model for the simplest modulation scheme OOK, by using as little as an Arduino, a few IR LEDs and some photodiodes. Granted, IR LEDs weren't the most realistic option for this, but since we required a special visible light photodiode, which we weren't able to procure, this was probably the next best thing.

One of my main contributions to the project involved writing the MATLAB simulation program for Color Shift Keying (CSK), which is one of the modulation schemes discussed in the paper. The use of such multi-color LEDs forms the principle behind CSK modulation. Color shift keying modulation is similar to frequency shift keying in that the bit patterns are encoded to color (wavelength) combinations. For example, for 4-CSK (two bits per symbol) the light source is wavelength keyed such that one of four possible wavelengths (colors) is transmitted per bit pair combination. In order to define various colors for communication, the IEEE 802.15.7 standard breaks the spectrum into 7 color bands in order to provide support for multiple LED color choices for communication. The CSK signal is generated by using three color light sources out of the seven color bands. The three vertices of the CSK constellation triangle are decided by the center wavelength of the three color bands on xy color coordinates. CSK has the following advantages:

1. The final output color (e.g., white) is guaranteed by the color coordinates. CSK channels are determined by mixed colors that are allocated in the color coordinates plane.
2. The total power of all CSK light sources is constant, although each light source may have a different instantaneous output power. CSK dimming ensures that the average optical power

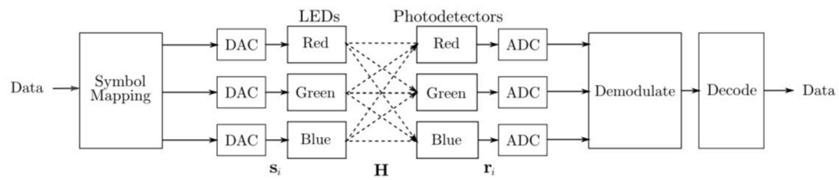


Figure 1: Functional block diagram of the VLC channel that supports the CSK modulation format.

As far as the future of the technology is concerned, several technical challenges must be addressed to realize the full potential of VLC technology. First, channel models for VLC are not well under- stood, especially for outdoor non-line-of-sight (NLOS) environments, and there is an active area of research for channel models and platforms for VLC. Also, the networking of the light sources and upgrading current infrastructures to support communication is another challenge, which requires support from the lighting industry. With continued growth in LED based light sources and the need for multi-Gb/s data distribution, VLC, being developed as a global industry standard in IEEE 802.15.7, promises to be a very attractive candidate as a future high data rate and power-efficient technology.