MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
EE101: INTRODUCTION TO ELECTRICAL & ELECTRONICS ENGINEERING

Project 4
Arduino-Based Visible Light Communication

ABSTRACT

This document has been written to guide freshman students for their Arduino project aiming to lay the foundations of visible light communication (VLC) by the construction of a simple system that is able to sense visible light and determine the dominant frequency of the absorbed light. The project aims to use the constructed VLC system to communicate ASCII-coded text.

INTRODUCTION

Visible light communication (VLC) essentially refers to the data transfer between systems realized by using a specific type of electromagnetic radiation which can be detected by the human eye. It is a quite new way of optical wireless communication (OWC) that uses 'visible' light as the transmission medium. The term OWC corresponds to a free-space optical link, where the transmitter and receiver are not necessarily aligned to each other[1].

In present fast paced life, there is a strong urgency for the improvement in the means of communication. VLC is a newly emerging trend that can easily pave the way for a comfortable wire-free future. The usage of light as a source of communication is an innovative technology[2] that is recently commercialized. A subset of VLC technology, known as Li-Fi, is currently commercially available[3].

Li-Fi systems are light communication systems that are capable of transmitting data at high speeds in a similar manner to Wi-Fi. But while Wi-Fi utilizes radio waves, Li-Fi uses visible, ultraviolet, and infrared light to transmit data. Currently, only light-emitting diodes (LEDs) can be used for the transmission of data in visible light. Light signals emitted by LEDs are captured by a photodiode device attached to the receiving device to access the transmitted data, which can be images, videos, documents, or the internet. LiFi is viewed as a complementary technology to WiFi, which can provide relief to the already congested radio spectrum for providing internet access to the general public.

---

[1] *Grobe, Liane, et al. "High-speed visible light communication systems." Communications Magazine, IEEE 51.12 (2013): 60-66.*

[2] *Sagotra, Rajan, and Reena Aggarwal. "Visible light communication." International Journal of Computer Trends and Technology (IJCTT) volume4 Issue4 (2013): 906-910.*

[3] *"LiFi Companies," LiFi: Wireless data from every light bulb, https://lifi.co/lifi-companies/ (accessed May 19, 2023).*

In VLC, LEDs that are used for illumination purposes are simultaneously used for wireless data transmission. It offers numerous advantages such as high data rates, unlicensed large bandwidth and better data security leading to smart spaces. Moreover, some of the notable advantages of VLC over RF (radio frequency) and IR (infrared) based systems are[4]:

• There are no regulations regarding the use of the visible EM (electromagnetic) spectrum.

• Unlike IR communication schemes, there are no health regulations to restrict the transmit power.

• Optical communication provides higher security than RF communication schemes; it is very difficult for an intruder to pick up the signal from outside the room.

Considering the above virtues, there has been recently an increased interest in VLC systems. The research is motivated by an increasing need for indoor communication systems and the improvements of light emitting diode technologies (LEDs). High brightness LEDs are already used for several applications and it is foreseen that they will also replace conventional lighting sources in the next decade. This widespread use provides the necessary infrastructure and hence removes one of the major hurdles faced by new communication schemes; thus making the technology particularly appealing. Consequently, the VLC system is expected to be the indoor communication system of the next generation.

If you are interested, you can check the articles mentioning recent achievements and trends in high-speed indoor VLC and LiFi research[5] [6] [7]. You can also go over the up-to-date website about LiFi which is mostly kept by non-academic wording[8].

In this project, you are asked to analyze a simple VLC system that consists of a transmitter utilizing a LED and a receiver utilizing a light-dependent resistor (LDR) as illustrated in Fig. 1. We will mostly focus on the receiver side. For the receiver, a simple voltage divider circuit is constructed using an LDR and a fixed resistance (where the selection of the resistance value is important for effectiveness of the VLC system). The voltage related to the light intensity on LDR is measured using an Arduino board and then this information is sent to MATLAB through serial communication. You will write MATLAB programs to analyze this information (mostly in the frequency domain). The necessary details of the design procedure and working principle of the VLC system are presented in the rest of this document.

---

[4] *Afgani, Mostafa Z., et al. "Visible light communication using OFDM." Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on. IEEE, 2006.*

[5] Haas, Harald, et al. "What is lifi?." *Journal of lightwave technology* 34.6 (2015): 1533-1544.

[6] *Wu, Xiping, et al. "Hybrid LiFi and WiFi networks: A survey." IEEE Communications Surveys & Tutorials 23.2 (2021): 1398-1420.*

[7] *Grobe, Liane, et al. "High-speed visible light communication systems." Communications Magazine, IEEE 51.12 (2013): 60-66.*

[8] *http://lifi.co/*

LIST OF EQUIPMENT

- A light-dependent resistor (LDR) for light detection
- An LED for light emission
- 2 Arduino UNO boards—one for the receiver and one for the transmitter
- Resistors
- Jumper cables for making connections
- A breadboard (protoboard) for constructing the circuits
- A MATLAB running computer for data acquisition and analysis

SCHEMATIC OF THE VLC SYSTEM

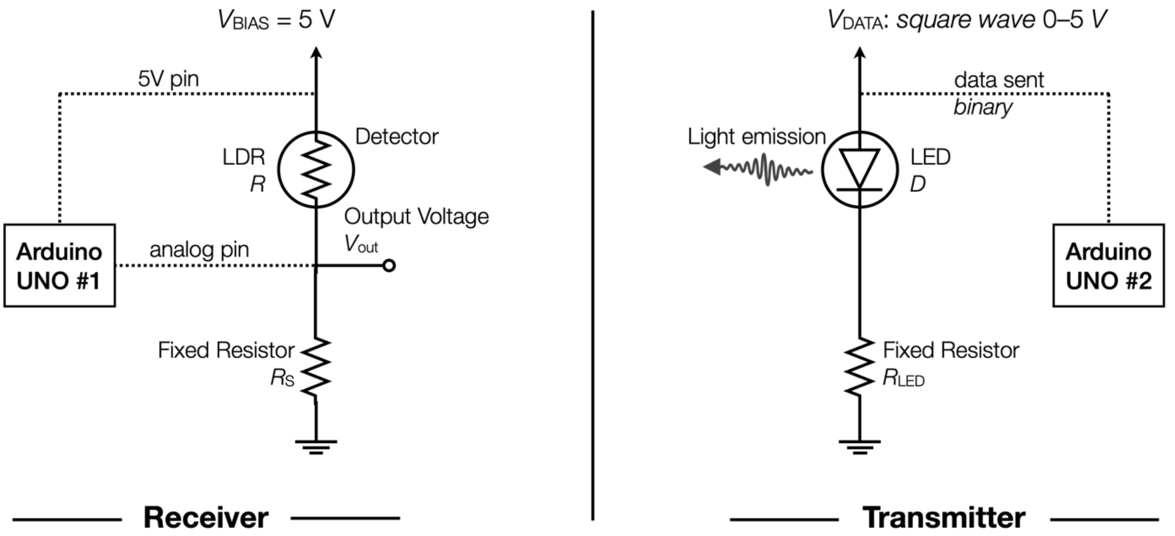

Figure 1. The schematic of the VLC system

On the transmitter side, a red LED is used to transmit data. To the LED, we applied a square-wave (V_{DATA}) oscillating between 0 and 5 V. The period of the square-wave is changed at certain instants to encode the transmitted data. The task on the receiving end is to determine the dominant frequency of the encoded light at each duration by analyzing its frequency characteristics and then decode the transmitted message in ASCII form. The distance of the LED to the LDR is varied to analyze the detection performance. Note that the biasing resistor R_{LED} is selected as 330 Ω on the transmitter side and V_{BIAS} = 5 V on the receiver side.

The resistance of LDR changes with the amount of exposed light. The voltage across the LDR is measured using an Arduino board. Throughout this project, the data supplied to you is obtained by Arduino UNO #1 shown on the receiver side of Figure 1. This data is uniformly sampled with a sampling frequency of f_S = 1.96 kHz—meaning that each sampled point is separated in time by $T_S = 1/f_S$ = 0.5102 ms.

DETAILS OF THE HARDWARE AND THE SOFTWARE

Read this section thoroughly before attempting to work on the project.

LDRs. Light-dependent resistors (LDR), also known as photo resistors, are light sensitive devices which are mostly employed to indicate the presence or absence of light, or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity. LDR is typically a nonlinear device whose resistance-illumination characteristic is illustrated in Figure 2.

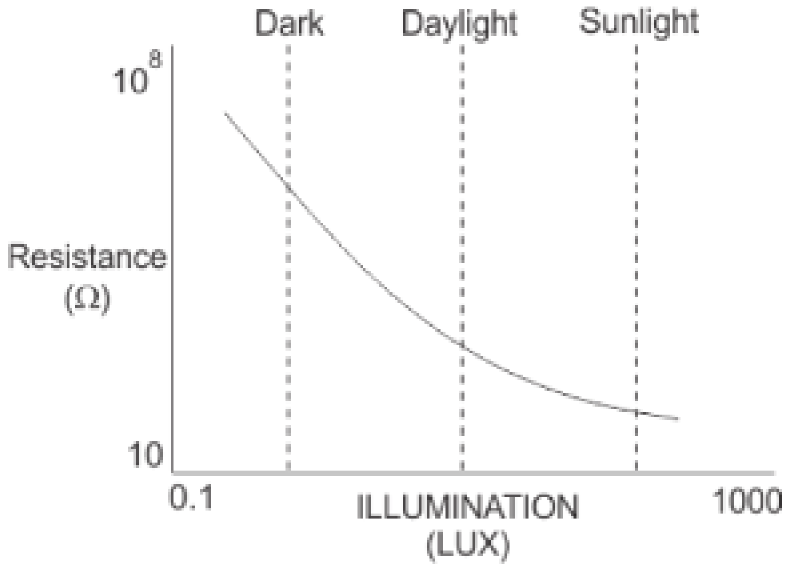

Figure 2. Resistance-illumination characteristic of a light-dependent resistor (LDR).

Voltage Dividers. Voltage division approach using two resistors is widely utilized in simple sensor configurations. Here we build a voltage divider circuit on the receiver side as illustrated in Figure 1. Since the same current passes through the LDR (R) and the fixed resistor (R_S), one can express the output voltage (V_{out}) as follows:

$$V_{out} = \frac{R_S}{R_S+R} V_{BIAS} \tag{1}$$

In the limit of very high illumination ($R \to 0$, or $R \ll R_S$), V_{out} approaches V_{BIAS} = 5 V. Conversely, in the dark ($R \to \infty$, or $R \gg R_S$), V_{out} approaches 0 V. Therefore, higher the illumination incident on LDR, higher the output voltage V_{out}. However, note that V_{out} and illumination level are not linearly related due to the nonlinear characteristic of LDR as illustrated in Figure 2.

Note that we should select an appropriate value for the fixed resistance (R_S) so that the change appearing in the output voltage is maximized. By doing so, we can increase the chance of accurate detection of the frequency content of the absorbed light.

Transmitter. On the transmitter side illustrated in Figure 1, the current passing through the light-emitting diode (LED) is controlled by changing the voltage V_{DATA} through a digital pin on Arduino UNO #2. This voltage (V_{DATA}) either takes the value 0 V or 5 V, in the form of a square wave. A current limiting resistor R_{LED} = 330 Ω is used to prevent LED burnout. The circuit built on the breadboard and connections on the Arduino board are shown in Figure 3.

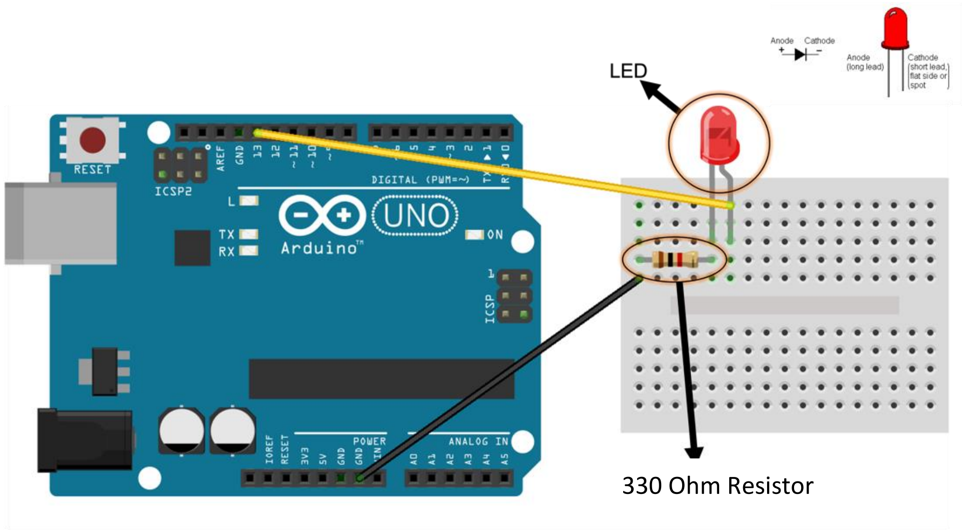

Figure 3. Connections on Arduino for the transmitter part.

As an example, a square wave switching at a frequency of 1Hz has been generated with the Arduino board (similar to the Blink example covered in the tutorial) and applied to the LED. The resulting waveform for V_{DATA} is presented in Figure 4. Note that the voltage waveform takes values 0 V and 5 V for the same amount of duration—hence the name square wave. In this example, the switching frequency is 1 Hz (i.e. the period is 1 second).

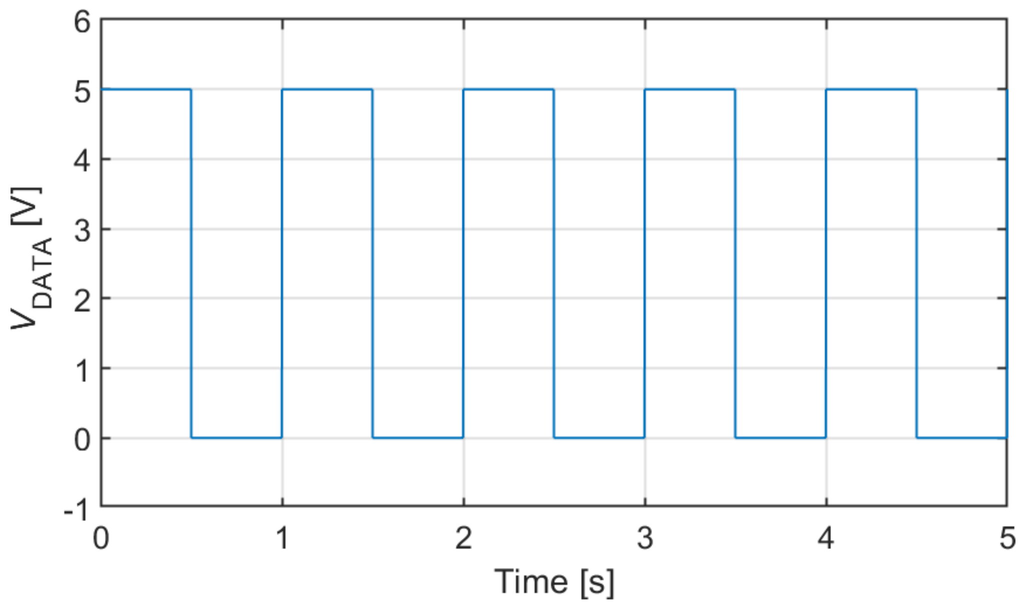

Figure 4. Voltage waveform example (1 Hz square wave) used on the transmitter side.

Receiver. On the receiver side illustrated in Figure 1, we measure the output voltage V_{out} that is related to the light intensity on the LDR and analyze its frequency characteristics. The illumination on LDR is sensed using a voltage divider circuit and 5 V output of Arduino (V_{BIAS} = 5 V). As the light intensity increases, LDR resistance (R) drops, and the output voltage (V_{out}) increases based on Equation 1. Therefore, voltage readout made via an analog input pin of Arduino UNO #1 can be considered as a measure of light flux incident on our LDR. The circuit built on the breadboard and connections on the Arduino board are shown in Figure 5.

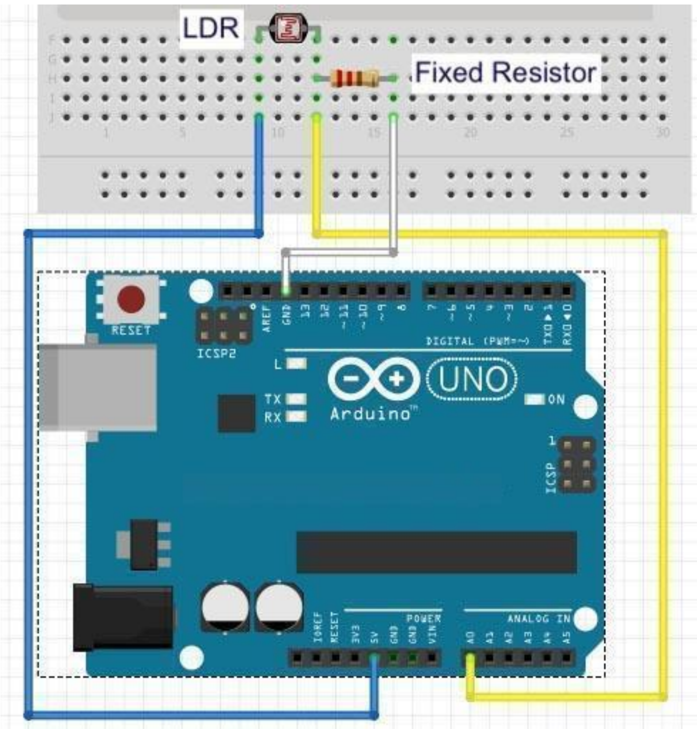

Figure 5. Connections on Arduino for the receiver part.

Arduino end of the receiver. An Arduino program is run to measure the output voltage via an analog channel and to send this information immediately over a serial communication channel to MATLAB (without storing it). The Arduino code used at the receiver side is given in Figure 6. Here, a serial communication channel between MATLAB and Arduino is opened with baud rate of 115200 kbits/s. The voltage readout is made by the analog pin A0 on the Arduino UNO #1. Since readout is made by an analog-to-digital converter of 10-bits, we divide and scale the readout by a factor of 5 V / (2^{10}–1) bits. You should pay attention to the comments given in Figure 6 to understand the order of operations made by the Arduino end of the receiver.

```
//Reads an analog input on pin A0, converts it to voltage, and prints the result to the Serial Monitor.

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 115200 bits per second:
  Serial.begin(115200);
}

// the loop routine runs over and over again forever:
void loop() {

  // read the input on analog pin 0 and,
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  // Force Arduino to store the value in floating number for variable 'voltage'
  float voltage = analogRead(A0) * (5.0 / 1023.0);
  // print out the value you read (will be accessed using MATLAB):
  Serial.println(voltage);

}
```

Figure 6. Arduino code for the receiver.

MATLAB end of the receiver. In the MATLAB end of this project, we use a Matlab program to communicate with the Arduino UNO #1 board via a serial channel, store and analyze the incoming information over this channel.

The MATLAB code for receiving the data sent by Arduino UNO #1 over the serial port is given in Figure 7. You should pay attention to the comments so that you understand the overall operation. To receive the data over the serial channel, we open a serial port at 115200 kbits/s, define the start and stop bits, and configure the callback function (which stops the readout when desired number of samples are reached). Note that the sampling frequency of this configuration corresponds to f_S = 1.96 kHz.

```
clear; clc;
% We open a serial port (115200 bits/s) and then configure the terminators
s = serialport('COM4',115200); % The address of Arduino is 'COM4'
configureTerminator(s,"CR/LF");

% We allocate a struct to store the incoming data
s.UserData = struct("Data",[],"Count",1);

% We clear the serial port device buffers
flush(s);

% Configure the callback function readData, see the function below. When
% condition of this function satisfied, it returns the data to be analyzed.
configureCallback(s,"terminator",@readData);

% Callback function
function readData(src,~)

recordTime = 5; % Recording time [sec]
samplingFrequency = 1960; % [Hz, or number of samples per second]
nSamples = recordTime*samplingFrequency; % Number of samples to be recorded
data = readline(src); % Read the ASCII data from the serialport object.

% Convert the string data to numeric type and save it in the UserData
% property of the serialport object.
src.UserData.Data(end+1) = str2double(data(1));

% Update the Count value of the serialport object.
src.UserData.Count = src.UserData.Count + 1;

% If nSampling data points have been collected from the Arduino, display
% 'Data acquired.'
if src.UserData.Count > nSamples
    configureCallback(src, "off"); disp('Data acquired.');
end

end
```

Figure 7. MATLAB readout code for retrieving data from Arduino serial channel

The data obtained with MATLAB is stored in the variable `'s.UserData.Data'`. The code in Figure 8 shows how to plot the data and take its Fourier transform using `fft` function of MATLAB to analyze its frequency content.

```
outputVoltage = s.UserData.Data; % Extract the output voltage data
samplingFrequency = 1960; % Sampling frequency in Hz
recordTime = length(outputVoltage)/samplingFrequency; % Total length of signal in time [s]
time = 0:1/samplingFrequency:recordTime; % Array of time [s]
time = time(1:end-1); % Discard last element since time starts at 0 s.

fftLength = length(outputVoltage); % length of the signal
FV = fft(outputVoltage - mean(outputVoltage)); % FFT of outputVoltage, mean subtracted
amplitudeSpectrum = abs(FV/fftLength); % Double-sided spectrum
amplitudeSpectrum = amplitudeSpectrum(1:fftLength/2+1); % Make Single-sided
amplitudeSpectrum(2:end-1) = 2*amplitudeSpectrum(2:end-1); % Correction of amplitude
freqs = samplingFrequency*(0:(fftLength/2))/fftLength; % Frequencies of FFT

figure(1); % Open new figure
plot(time,outputVoltage); % Plot against linearly spaced time array
xlabel('Time [s]'); ylabel('Output Voltage V_{out} [V]');
grid on;

figure(2); % Open new figure
semilogx(freqs,amplitudeSpectrum); % Plot against logarithmically spaced frequencies
xlabel("Frequency [Hz]"); ylabel("Amplitude Spectrum of V_{out} [V]");
grid on;
```

Figure 8. A sample MATLAB analysis code

When we transmit the sample square wave switching at a frequency of 1Hz (given in Figure 4) through the LED, the voltage V_{out} on the receiver side is measured as shown in Figure 9. Note that the switching frequency is 1 Hz, and the voltage waveform changes its value from a low voltage readout to a higher value—in conjunction with the applied waveform presented in Figure 4. Here we define voltage change (ΔV) as the difference between the maximum and minimum values in the output voltage signal V_{out}. Note that the larger the voltage change, the easier to detect the signal.

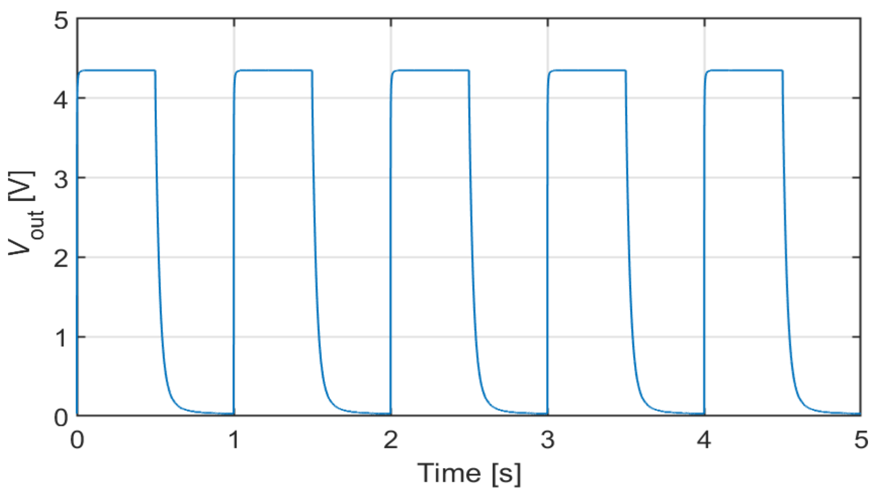

Figure 9. The output voltage waveform obtained by the receiver corresponding to 1 Hz square wave presented in Figure 4.

The amplitude spectrum of the output voltage signal V_{out} is illustrated in Figure 10 for the provided example. Note that the fundamental frequency is 1 Hz (i.e., the dominant frequency in the FFT) and its magnitude has value 2.72. Since the transformed signal is close to a square wave (distorted due to a finite rise and fall time of LDR response), we have higher order harmonics of smaller magnitudes in integer multiples of 1 Hz, as expected.

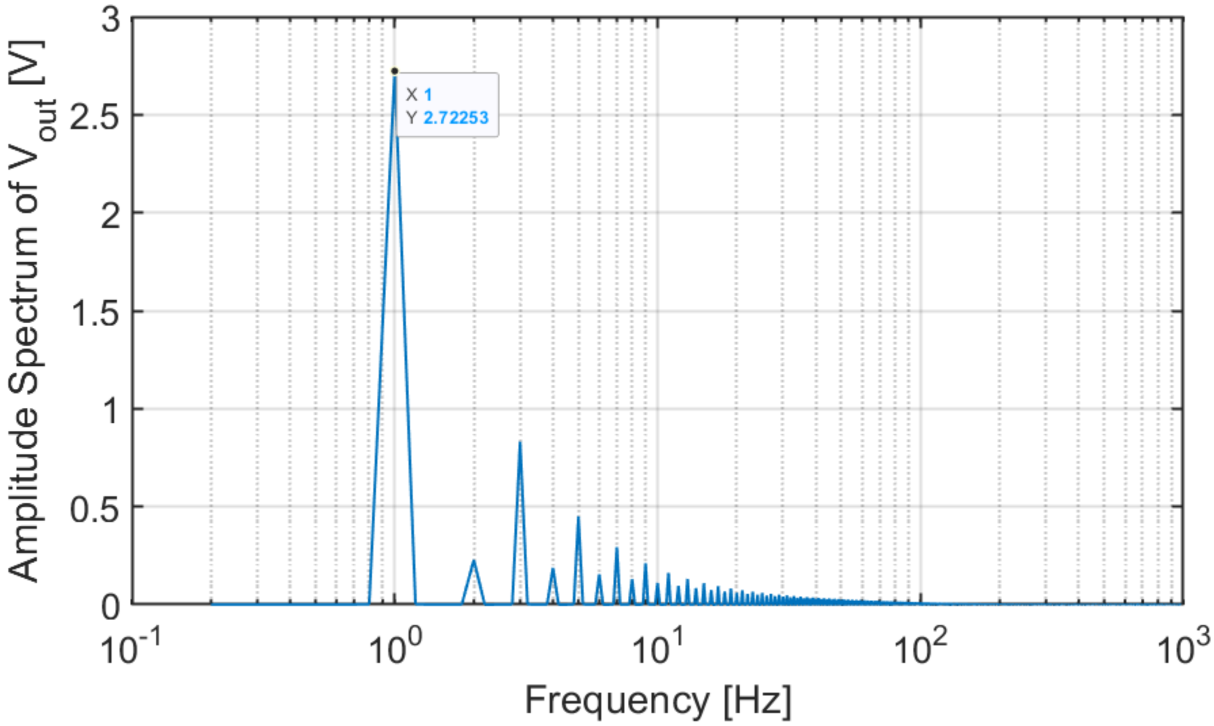

Figure 10. The amplitude spectrum of acquired signal as presented in Figure 9.

Project Work To Be Done

All the data you will analyze are stored in a single `mat` file named as `'vars_arduino_project.mat'` and are provided in ODTUClass. You can double click this file to load variables into your MATLAB workspace. The variables that will be used are indicated under each part below. You can benefit from the MATLAB codes presented in the project description above to help you guide on your work. You should modify these codes as needed. For example, you can change the first line of Figure 8 to

```
outputVoltage = c1;
```

to work on the case #1 of Part 1.

Part 1. Main Characteristics of VLC System Composed of LDR and LED

Here, you will work on some basic aspects of our VLC system by analyzing the data provided.

On the receiver side, 3 different values of biasing resistor (R_S) were used in the voltage divider circuit as shown in the table below. An opaque tube was used to block interference from the environment. The recorded voltage data (V_{out}) for these three different cases are named as MATLAB variables "`c1`", "`c2`", and "`c3`", for the cases 1, 2, and 3, respectively.

Case	**1**	**2**	**3**
R_S [kΩ]	100	8.2	0.56
Variable name	`c1`	`c2`	`c3`

P1.1. Plot the corresponding $V_{out}(t)$ for each case on the same plot. Which case is optimum for the VLC system considering the range of the output voltage signal, i.e., output voltage change? Justify your choice.

P1.2. For case #2, determine the maximum resistance value of LDR (i.e. dark case corresponding to $V_{DATA} = 0$ V) and minimum resistance value of LDR (i.e. bright light case corresponding to $V_{DATA} = 5$ V).

P1.3[9]. Plot the amplitude spectrum (i.e. magnitude of the Fourier transform obtained using `fft` in MATLAB) for each case on the same plot. Determine and show the fundamental (dominant) frequency for each case. Why is the frequency at which FFT takes its maximum value considered so as to determine the frequency of the signal?

PART 2. ENVIRONMENTAL (AMBIENT) EFFECTS ON VLC OPERATION

In this part, you will analyze the effects of the environment on the measurements made by our receiver. We have 3 different cases summarized in the table below:

Case	**4**	**5**	**6**
Ambient condition	Dark environment	Dark environment	Fluorescent light open
Receiver distance [cm]	4	7	4
Variable name	`c4`	`c5`	`c6`

P2.1. Plot the corresponding $V_{out}(t)$ for the case #4 and #5 on the same graph, which correspond to cases with different receiver to transmitter distance (x). Calculate the voltage change ΔV for each case (i.e. the maximum voltage value minus the minimum voltage value). Suppose that we can model the relation between the voltage change ΔV and the distance x between the transmitter and the receiver as follows by assuming that ΔV is inversely proportional to the squared distance x^2:

$$\Delta V = \frac{c}{(x + x_0)^2} \tag{2}$$

Using this model, determine[10] the parameters x_0 and c for the case #4 and #5. Assuming that the minimum detection limit for our system is $\Delta V \geq 25\ \mathrm{mV}$, determine the maximum distance x for a reliable operation using the found parameters x_0 and c.

---

[9] For this part, subtract the mean value of the entire data from itself since we are not interested in the dc part of the signal. You can use *mean* function in MATLAB in the form of x - mean(x).

[10] You may think that parameter c is the gain and x_0 is a dispersion coefficient due to non-idealities in the point-source approximation of LED.

P2.2. To analyze the interference effect of a conventional fluorescent lighting on the VLC system, plot the frequency content (by subtracting the mean like P1.3) for the case #4 and case #6 on the same graph. Can you see anything unexpected in this spectrum? If it is so, what can be the origin of this change? When fluorescent lighting is present in the environment, what limitation does it impose on the selection of the operating frequency of our VLC system?

PART 3. A FULLY OPERATIONAL SIMPLE VLC SYSTEM FOR SENDING ASCII-CODED TEXT

In this part, your task is to encode the ASCII-coded text transmitted via the LED. The receiver readout is stored as the variable "`msg`". The transmitted ASCII message is coded in binary form that is composed of 7-bits. The message is initiated with a constant illumination, and then it is followed by 7-bits, each of which is either 0 or 1. The bits 0 and 1 is coded with two distinct switching frequency of the LED. The lower switching frequency corresponds to bit 0 and the higher switching frequency corresponds to bit 1. Each bit is transmitted for 100 ms (equivalently 196 sampling points on the receiver side). Once all the 7-bits in the ASCII-coded text are transmitted, the entire message is repeated again.

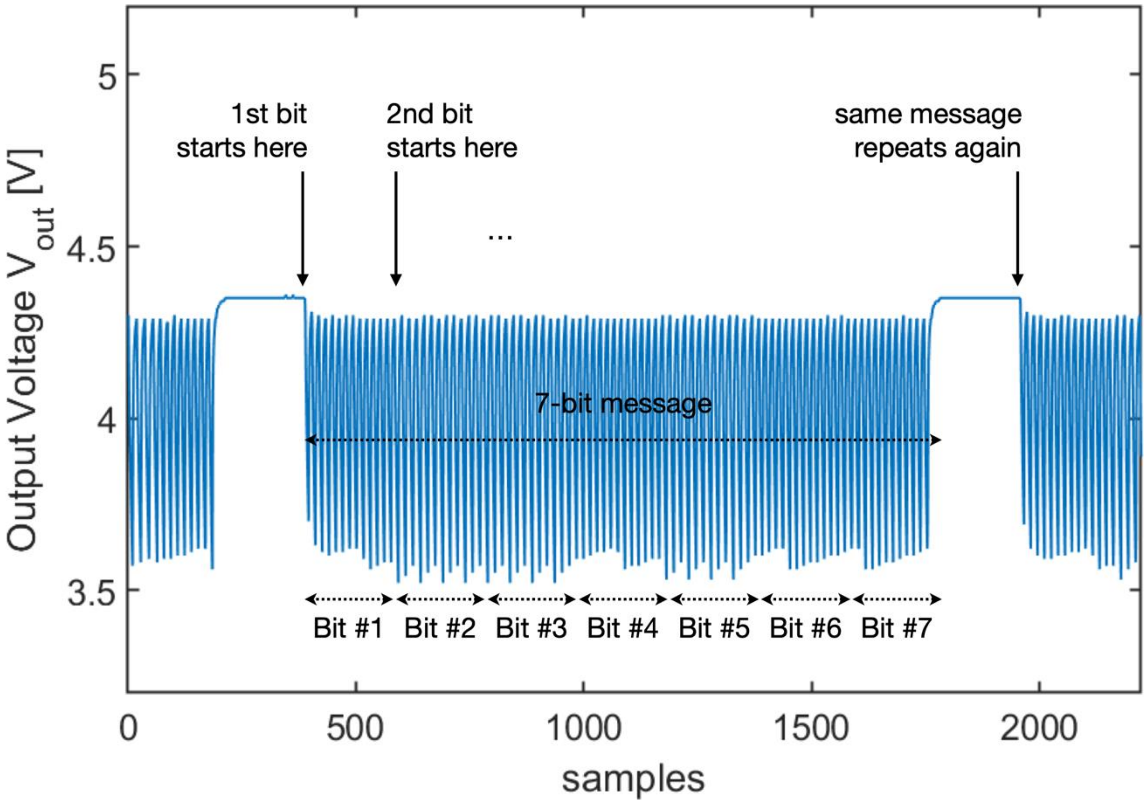

Figure 11. A sample plot showing the starting indices for each bit.

A sample output voltage signal measured on the receiver side is given in Figure 11. The starting points of the first 2 bits are marked on the plot of this received data. Noting that each bit is separated by 196 points, you can determine the start and end indices of each bit. Let `i1` and `i2` denote the start and end points of a bit. Then to find the switching frequency of the LED in this duration, you need to take the Fourier transform of the received data in this duration from `i1` to `i2`. (Hint: You can benefit from the code in Figure 8 by modifying it properly). With this information, you can understand the corresponding bit transmitted in this duration. You can repeat this process for each of the 7-bits.

P3.1. Plot $V_{out}(t)$ for this message and show the starting point of each bit on the plot. Write down the start and end indices of each bit. Note that the sampling frequency of the Arduino is f_S = 1.96 kHz. Hence, each transmitted bit corresponds to 196 samples since each bit is transmitted for a duration of 100 ms. What is the bitrate[11] of our VLC system? Is this simple VLC system capable of transmitting data at high speeds? Comment. (Complicated modulation and coding schemes are used in practice to enable fast communication speeds. You can learn such techniques in the communication courses of our department.)

P3.2. Determine the switching frequencies of the LED corresponding to bits 0 and 1. Plot the amplitude spectrum[12] of each bit segment and use data tips to find the dominant frequency for each bit. Write down the transmitted message in binary form. Then convert this binary message into decimal, and determine the corresponding character using an ASCII chart (note that it is case-sensitive). In your report include the plots of the two amplitude spectrum corresponding to bits 0 and 1 (you may use any portion of the received signal that corresponds to bit 0 and 1).

---

[11] The bitrate is defined as the number of bits send in 1 second. Note that it is not the samples per seconds (this is the sampling frequency, not bitrate).

[12] You need to segment your data (see Array Indexing in MATLAB support), and take successive Fourier transform of each segment to determine the frequency. Like P1.3 and P2.2, subtract the mean of each segment to get rid of the DC part.