Data transmission prototypes through wireless optical communication link using Arduino microcontroller

TAREK A. ELDALLAL[a,b,*], HEBA A. FAYED[a,b], MOUSTAFA H. ALY[a,b,c]
[a]*Electronics and Communications Engineering Department, College of Engineering and Technology,*
[b]*Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt*
[c]*OSA member, mosaly@aast.edu*

In this paper, Free Space Optical communication link prototypes have been practically experimented. Audio and text data transmission models are implemented using a programmed microcontroller "Arduino Uno". Using a 650 nm laser source, a digital audio signal is transmitted with a data rate of 506.3 kbps in addition to a text data transmitted with a bit rate of 125 kbps. Both signals are received using a TSL250R photodiode. These prototypes create an FSO communication link designed for audio and text transmission. The use of Arduino microcontroller offers a more accurate transmission of higher quality. The concept discussed can be generalized for real optical communication systems instead of lab work only.

(Received December 11, 2018; accepted August 20, 2019)

Keywords: Optical communication, Wireless communication, Free space optics, Data transmission, Audio signal, Text data, Arduino microcontroller

1. Introduction

The use of wireless optical communication (WOC) technology has extensively increased as an alternative to the existing radio frequency (RF) technology for many applications [1]. Free space optics (FSO) is a telecommunication technology that transmits data via the atmosphere using light signal as a carrier.

This technology does not require a license to use and it is capable of handling high data rates with lower power consumption and low-cost deployment. In addition, it does not suffer any interference from electronic systems and has an excellent security system where any interception can be immediately detected [2]. WOC can be performed in the infrared, the visible light, and the ultraviolet spectrums [3].

The first experiment in FSO included demonstrations conducted by Alexander Graham Bell in the late 19th century. Bell used light beams to transmit voice conversations through air, and he called it the *Photophone* [4]. The technology of Bell's photophone was not able to guarantee the required quality of service at that time, however it is considered the first practical manifestation of the free-space optical link [4]. In the last few years, there were various experiments testing the transmission of data through the wireless optical links, for both indoor and outdoor applications. Recently, after the use of microcontrollers rather than electronic circuits only, similar experiments were implemented for the same purpose and more. Researchers continued developing systems that not only transmit audio but also text, images, and videos [5], as well as providing internet access [6].

Optical wireless systems can be broadly classified as outdoor and indoor systems. The outdoor systems are also known as FSO systems. FSO links are point-to-point systems that transmit a modulated beam of visible or infrared light through the atmosphere. For a long distance, laser is used as a light source. Light Fidelity (Li-Fi) is a category of WOC that includes both infrared and visible light communications. Uniquely, Li-Fi can use the same light energy of illumination for communication. Li-Fi technology is essential for indoor applications.

Light emitting diode (LED) can be switched on and off very fast that the human eye cannot detect as its operating speed is less than one microsecond. This invisible on-off activity enables data transmission using binary codes. If the LED is on, a digital '1' is transmitted, and a digital '0' is transmitted when the LED is off. Li-Fi allows data transmission by modulating light intensity, which is then received by a photo-sensitive detector. The light signal is then demodulated into its electronic form.

In this paper, two prototypes are implemented to study and attain the concept of transmitting and receiving both audio and text data using the FSO link with the aid of an Arduino microcontroller for outdoor applications. The use of the microcontroller offers a more accurate transmission of higher quality. Each prototype is designed to transmit different types of data from the main building to other buildings on the campus representing a direct communication link between various faculty members. This connection can be generalized to include an indoor design using the Li-Fi technology by applying some manageable adjustments to achieve an internal communication between the dean and his students. The experiments, in this paper, represent a practical demonstration of FSO application according to the courses studied in the students' undergraduate level, including optical communications and optical devices.

2. Experimental setup

This section introduces the design and implementation of the two prototypes used to transmit and receive audio signals and text, respectively.

2.1. Transmitting an audio signal

The purpose of this experiment is to build a prototype for transmitting and receiving audio data through a wireless optical link, which can work for long distance applications. The system configuration is shown in Fig. 1. The channel for such an FSO application could be either atmosphere (air) or vacuum for the satellite to satellite and satellite to ground station communications. To execute this experiment, a Line Of Sight (LOS) between both the transmitter and receiver has to be maintained.

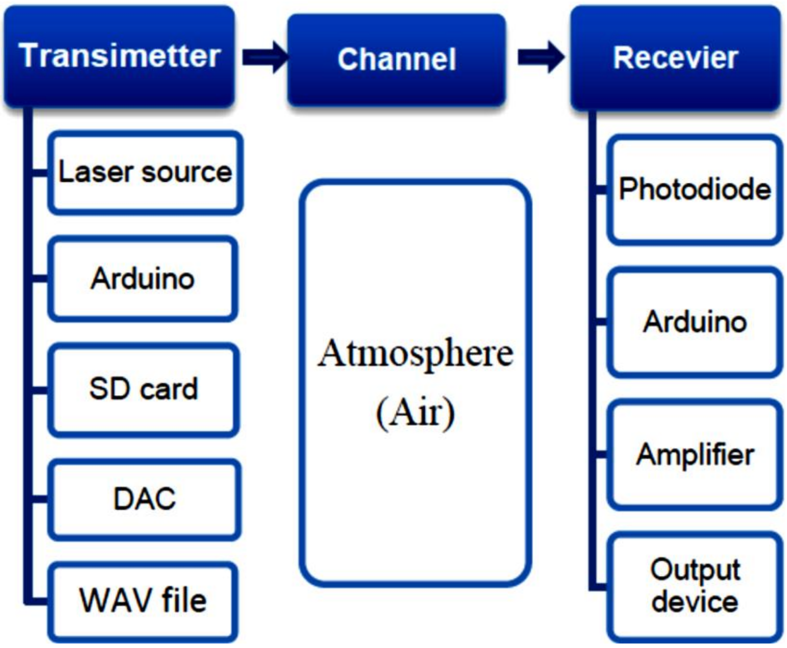

Fig. 1 Audio system configuration

2.1.1. Transmitter

The transmitter circuit in Fig. 2 incorporates a mono SD card module that reads the waveform audio file format. The default content of a WAV file is uncompressed, pulse code modulated (PCM) digital samples are derived from the analog source [7], and has higher quality than the MPEG-2 audio layer III (mp3). The audio file is converted to a digital form and then stored in the memory card. A Micro SD card reader module is connected to the Arduino board to read the data from the memory card and pass it to the laser driving circuit. The laser source used in this experiment is 650 nm, operates at 5 V and is associated with Arduino. The transmitter circuit has a switch connected to the Arduino board and the ground to control whether to play or stop the audio. Fig. 3 represents the transmitted signal as it appears in the oscilloscope.

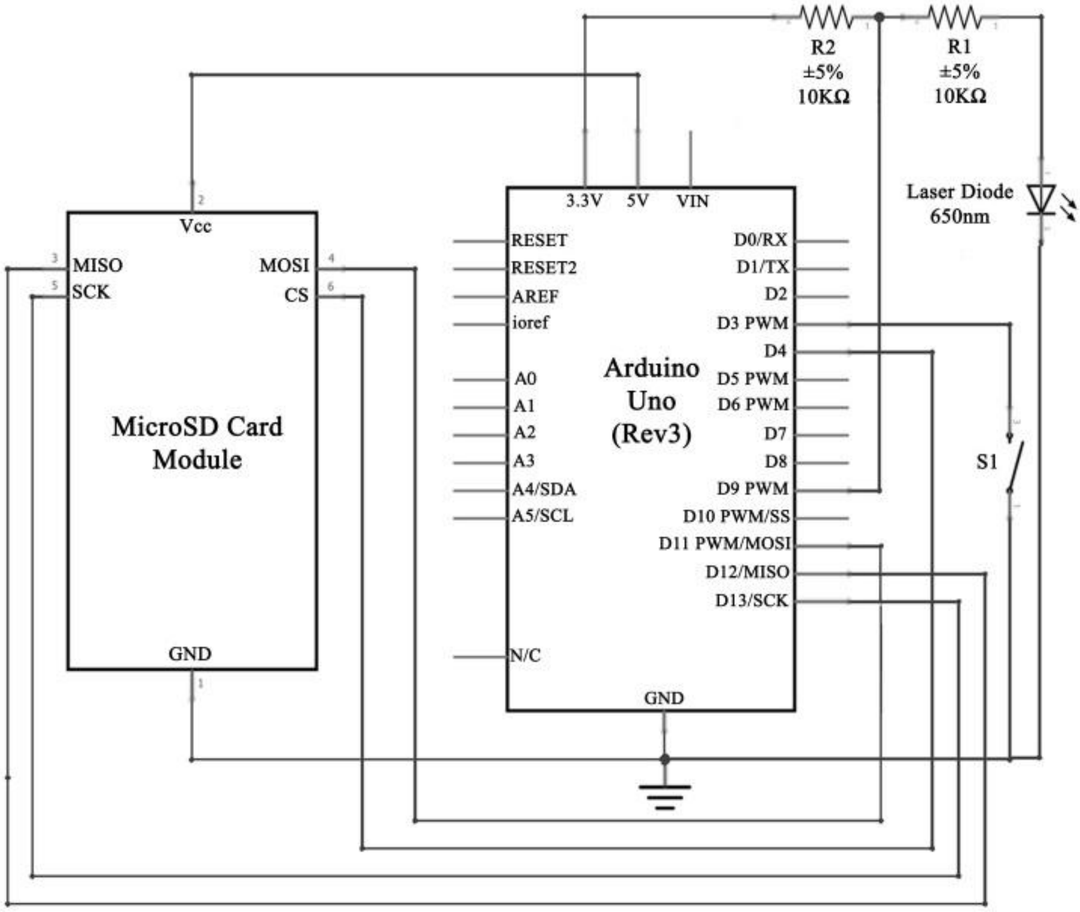

Fig. 2. Transmitter circuit design

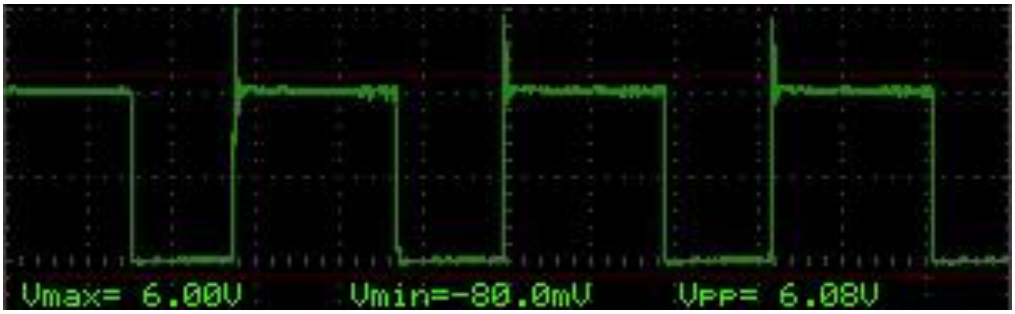

Fig. 3. Transmitted audio signal

2.1.2. Receiver

The proposed model of the receiving circuit is shown in Fig. 4. It consists of a photodiode driving circuit, Arduino board, audio amplifier circuit and a high-quality speaker as an output device.

Photodetector circuit

The TSL250R is a light-to-voltage optical sensor IC containing a photodiode responsible for converting the received light signal that carries the audio signal into an electrical one. The transimpedance amplifier (TIA) converts the low-level photodiode current signal into a usable voltage output. As shown in Fig. 5, the operational amplifier is used to avoid slow response, low gain, and large time constant by feeding the photodiode output current directly into the summing point of the TIA. Consequently, a faster response time, a larger gain and an improvement of the signal to noise ratio are successfully achieved. The TSL250R sensor output voltage is directly proportional to the light intensity (irradiance) on the photodiode. This is used for improving the amplifier offset-voltage stability with lower power consumption [8]. The TSL250R optical sensor has high irradiance responsivity; typically 137 mV/W.cm^{-2} at a wavelength of 635 nm [8] and a relative responsivity of 1.04 A/W at a wavelength of 655 nm as shown in Fig. 6.

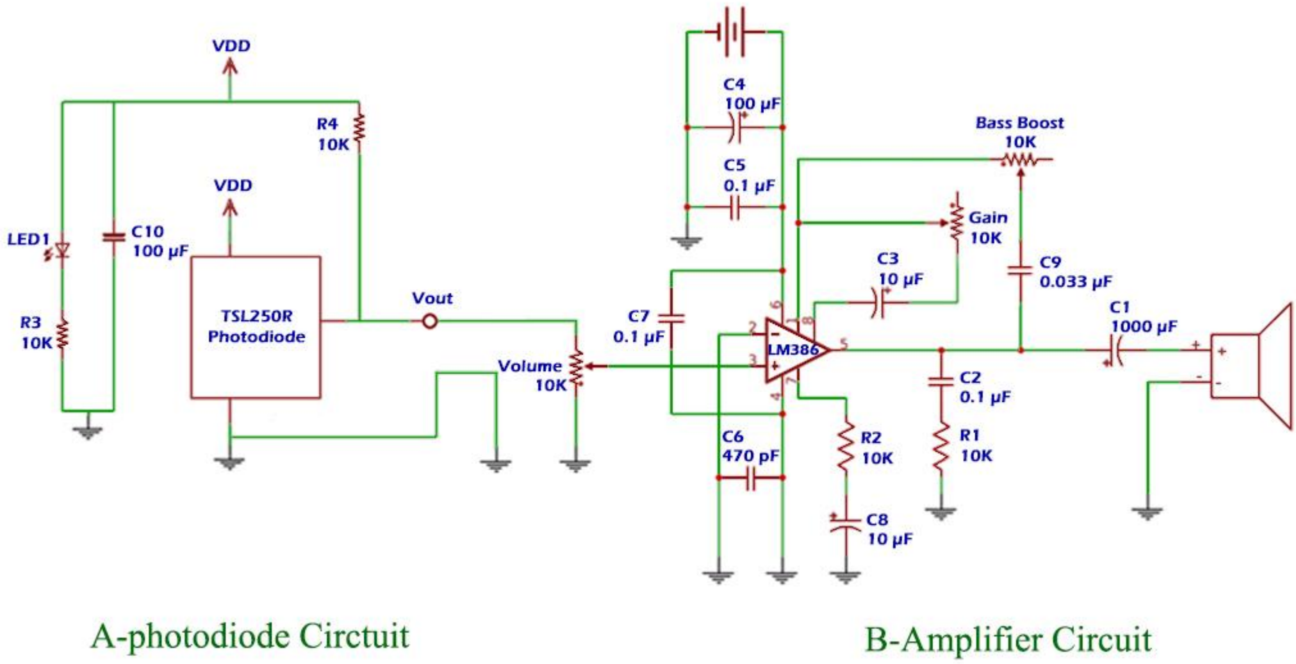

Fig. 4. Received circuit design (photodiode circuit plus an audio amplifier circuit)

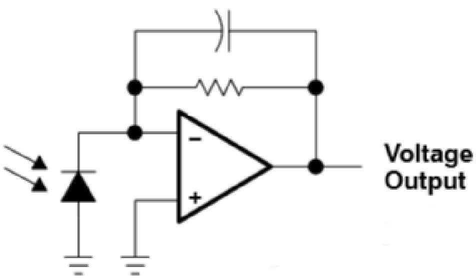

Fig. 5. TSL250R functional block diagram [8]

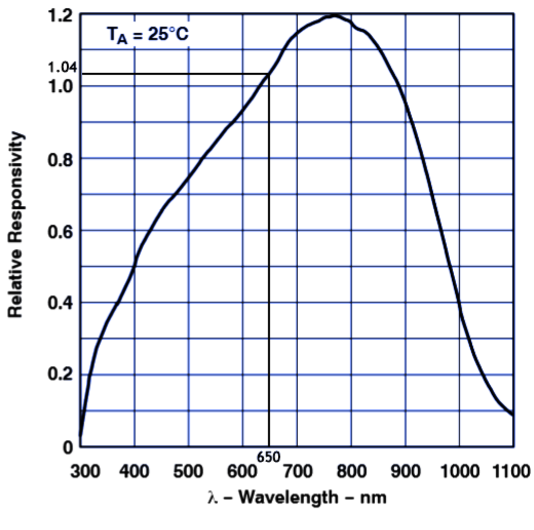

Fig. 6. TSL250R photodiode spectral responsivity [8]

The photodiode circuit in Fig. 4 shows the connection of the TSL250R optical sensor with an LED for power connection indication, a 100 nF capacitor for noise elimination and a load resistor for the output voltage installation. Fig. 7 shows the received audio signal before being amplified by the LM386 amplifier circuit.

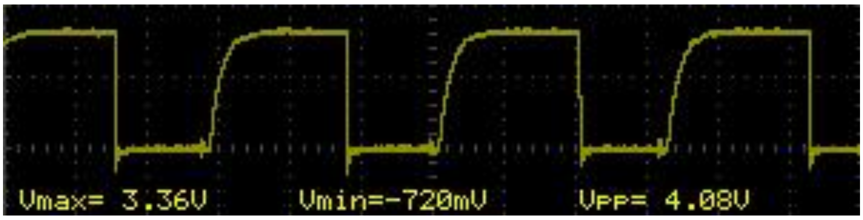

Fig. 7. The received signal after the photodiode circuit

Amplifier circuit

The LM386 low voltage audio power amplifier receives the output signal from the Arduino board to increase the amplitude (strength) of the received audio signal, and then passes it through the speaker. The amplifier shown in Fig. 4-B has three potentiometers to control the gain, volume, and the bass boost. Using a 10 μF capacitor in series with a 10 kΩ potentiometer between pins 1 and 8, the gain can be set to any value between 26 to 46 dB. A 0.033 μF capacitor in series with a 10 kΩ potentiometer between pins 1 and 5 controls the low pass filter (bass boost) which eliminates most of the noise that was not removed by the decoupling capacitors. A 100 μF and 0.1 μF capacitors are used to filter both low and high frequency noise. Power supply decoupling capacitors 100 μF and 0.1 μF are used between the positive and negative power rails, and a 0.1 μF capacitor is used between pins 4 and 6 for further decoupling of the power supply and the IC.

Fig. 8 displays a 1 kHz square wave produced by a function generator and the same signal after being amplified by the designed LM386 audio amplifier circuit, via a DSO1052B digital oscilloscope to study the effect of the amplifier circuit before attaching it to the receiver model.

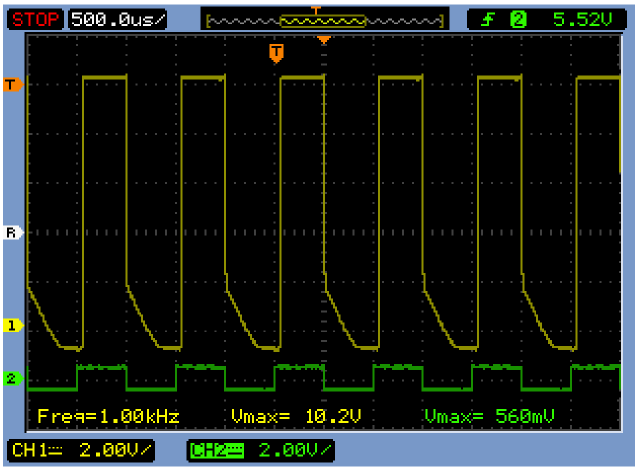

Fig. 8. The 1 kHz signal before and after amplification

To complete the receiver system, the amplifier circuit (B) is connected to the photodiode circuit (A) in Fig. 4. The output signal shown in Fig. 9 suffers a distortion and has lost the uniform shape of the square wave due to the change in the rise and fall times. In order to overcome this distortion, the time constant (τ = RC) of the capacitor must be modified to a lower value without affecting the gain by reducing the capacitor value. Adding a 470 pF capacitor between the positive input signal and ground filters the interference accumulated throughout the audio input wires.

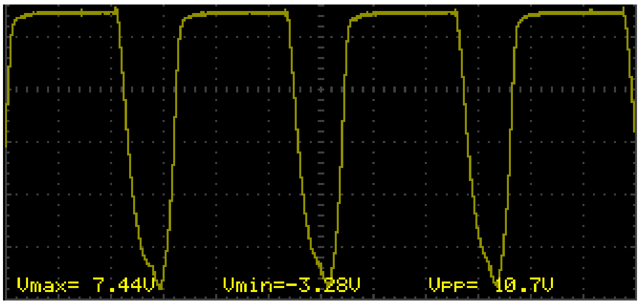

Fig. 9. Received audio signal after being amplified

2.1.3. Results and discussion

To accomplish the prototype of the WOC link for a long distance; a digital audio file was transmitted with a data rate of 506.3kbps using a laser source of 650 nm wavelength (in the visible light region), as it was the available source. The wavelengths 1310 and 1550 nm are more preferable to use in a definitive model, through the atmospheric channel (air) leading to high-quality received audio data using a TSL250R optical sensor. Then, a speaker converts the audio signal into sound.

In order to attain a high-quality transmission and an accurate lossless easy to process and manipulate high-quality audio file, the WAV format is used. An 8-bit mono unsigned value audio file with a sample rate of 64 kbps is used and stored in the SD card memory for more flexibility. The Arduino board is utilized in this model because it is an open-source electronics platform based on easy-to-use hardware and software. The Arduino Software IDE, based on Processing, runs on Windows, Macintosh OSX, and Linux operating systems [9]. Using the Arduino facilitates carrying the digital data and reading it from the SD card reader module through the file system.

As for the transmitted signal shown in Fig. 10, there is noise added from the transmitter circuit to the signal before being dispatched. The noise increases to the maximum and peak-to-peak voltage from 4 to 6 V and affects the quality of the audio significantly if a speaker is directly connected to the transmitted signal.

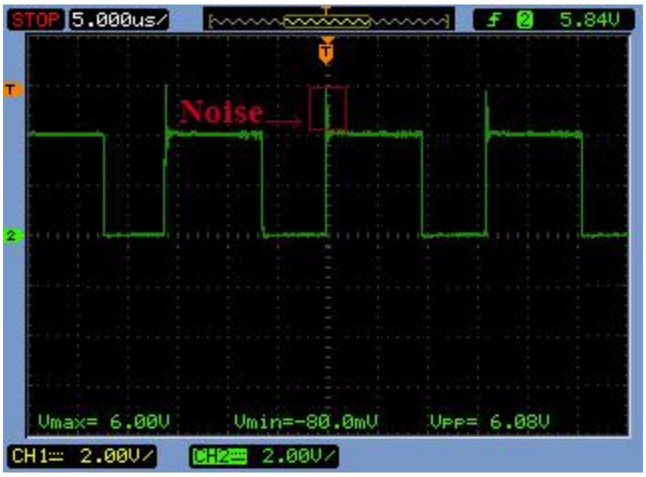

Fig. 10. Transmitted signal and effect of the system internal noise

Furthermore, the experiment substantiated that connecting an Arduino with a pulse width modulation (PWM) code to the photodiode circuit (Fig. 4-A) in order to control the amplitude of the received signal (second signal in Fig. 11), will eventually overcome the blemish of the high rise time. Connecting the output of the photodiode circuit to the Arduino directly instead of the amplifier circuit (Fig. 4-B) will enhance the received signal, reduce the rise time from 1.9 µs to less than 1 µs, increase the peak to peak voltage from 4.08 to 5.6 V and improve the positive half cycle width from 9 to 10.4 µs. Then, the Arduino output will be correlated to the audio amplifier circuit input.

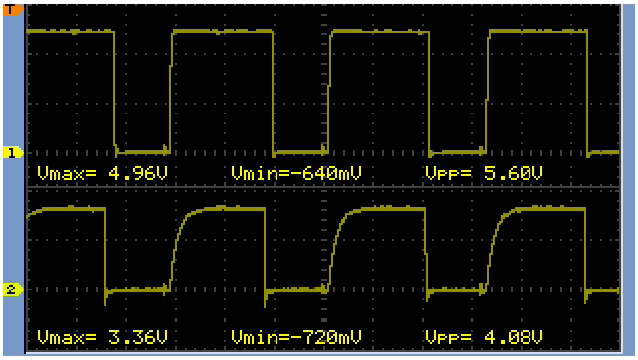

Fig. 11. Comparison between the received signals using the Arduino (1) vs. not using the Arduino (2)

The experiment shows that there are two approaches to consider after receiving the signal using the photodiode circuit. The first model uses the LM387 audio amplifier in Fig. 4-B before displaying the output signal on a speaker. However, this is still inadequate due to the slanted rising and falling edges shown in Fig. 9 caused by the capacitors' insufficiency. The second reveals that using the Arduino after the photodiode circuit instead of the electronic amplifier circuit is more suitable. Empirically, the second approach shows an improvement of the signal disfigurement and results in a more satisfactory sound quality. Fig. 12 shows a comparison between the two strategies in the maximum and minimum voltage and the rise and fall times of both output signals.

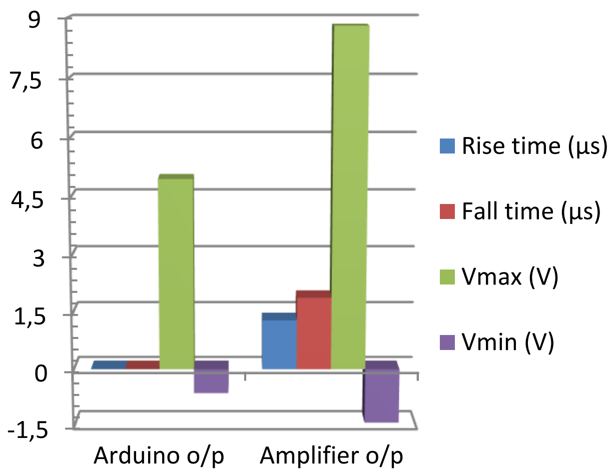

Fig. 12. A comparison between using the Arduino and the LM386 amplifier circuit after receiving the signal

Conclusively, Fig. 13 shows a comparison between the measurements of the peak-to-peak voltage, the positive and negative width of the signals, where signal A is the transmitted signal, signal B is the received signal, signal C is the received signal after using the Arduino and signal D is the received signal after using the amplifier circuit.

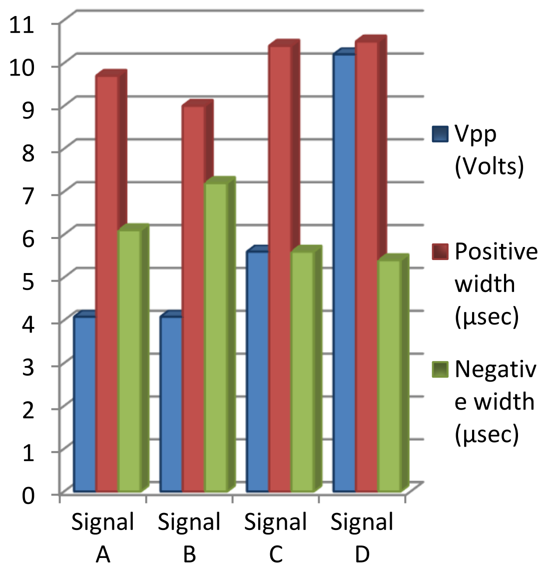

Fig. 13. A comparison between the transmitted and received signals

The results in Figs. 12 and 13 show that signal C is preferable to use, as it has the lowest rise and fall times and a more satisfactory peak to peak voltage compared to signal D. Using an Arduino microcontroller without supplementing the amplifier circuit improves the received signal, overcomes the transmitted signal noise problem and yields a very delicate sound. Hence, the Arduino countervails the side effects of the amplifier system and completely adjusts the received signal. Fig. 14 displays the transmitted and received signals using the microcontroller Arduino.

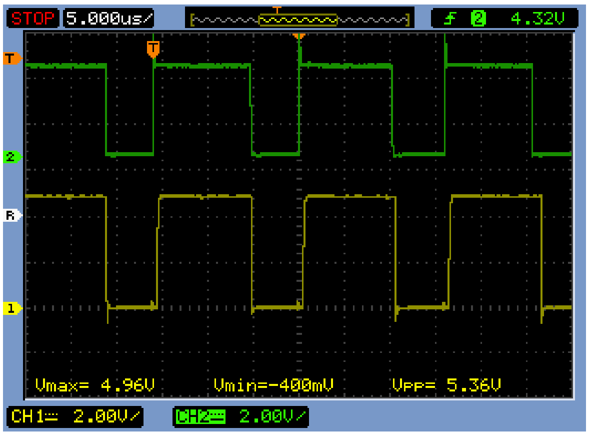

Fig. 14. Transmitted and received signals using Arduino board

2.2. Text transmission model

The aim of this experiment is creating a one-way text data communication between two computers through an FSO link using Arduino boards in both the transmitter and receiver. Like the first experiment, a 650 nm laser source is used to carry information through the atmospheric channel and a TSL250R photodiode to receive it. Moreover, an LOS between the laser source and the photodiode is required. However, in this prototype, as shown in Fig. 15, the Arduino serial monitor considers the interface between the program and the user. The first user input uses the transmitter Arduino serial monitor. Then, the program will process the data and upload it to the laser driving circuit. After the data receival using the photodetector circuit, the Arduino program will process this received data and display it onto the second computer serial monitor. Despite the similarity of the transmitter and receiver circuits in the two experiments, the Arduino is basically programmed to achieve data processing and error detection, accomplish free error transmission and make the application straightforward to use.

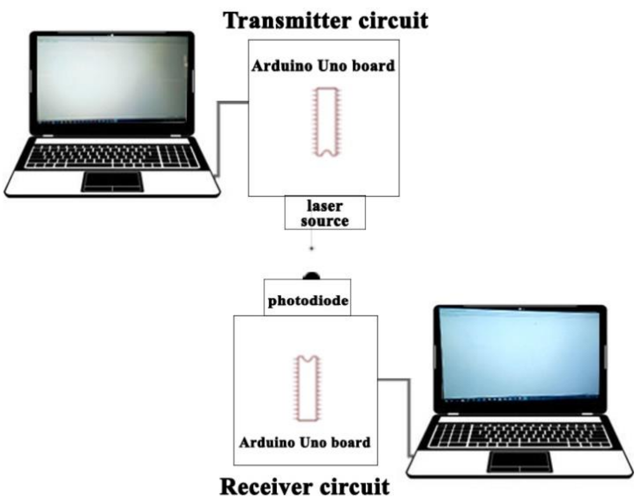

Fig.15. Text transmission implemented model diagram

2.2.1. Transmitter

The transmitter circuit in Fig. 16 illustrates the design and components of the laser driving circuit connected to the Arduino board. It is then connected to the input screen (the Arduino serial monitor) via the COM ports of the computer to encode the typed text and upload it to the laser circuit.

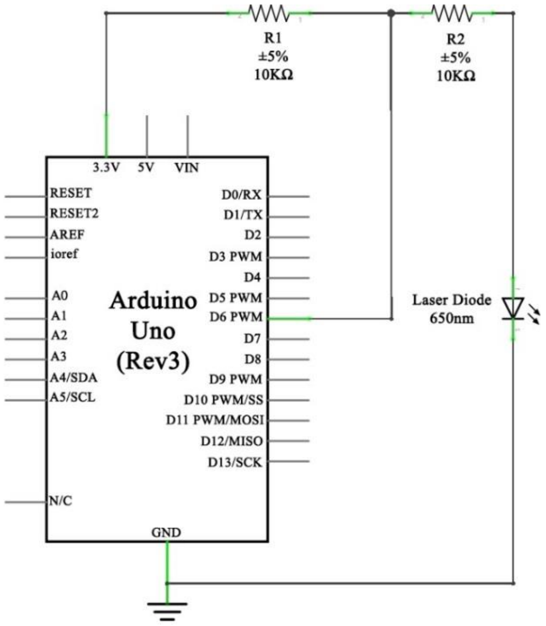

Fig. 16. Transmitter circuit diagram

2.2.2. Receiver

Compared to the first experiment, Sec. 2.1.2, the receiver of this model is considerably similar to the design of the photodetector circuit, as represented in Fig. 17. However, the role of the Arduino in the experiment is absolutely different. The Arduino program is responsible for reading the received data from the photodiode circuit. Afterwards, the received signal is demodulated and decoded. The text is then displayed on the serial window of the receiving computer.

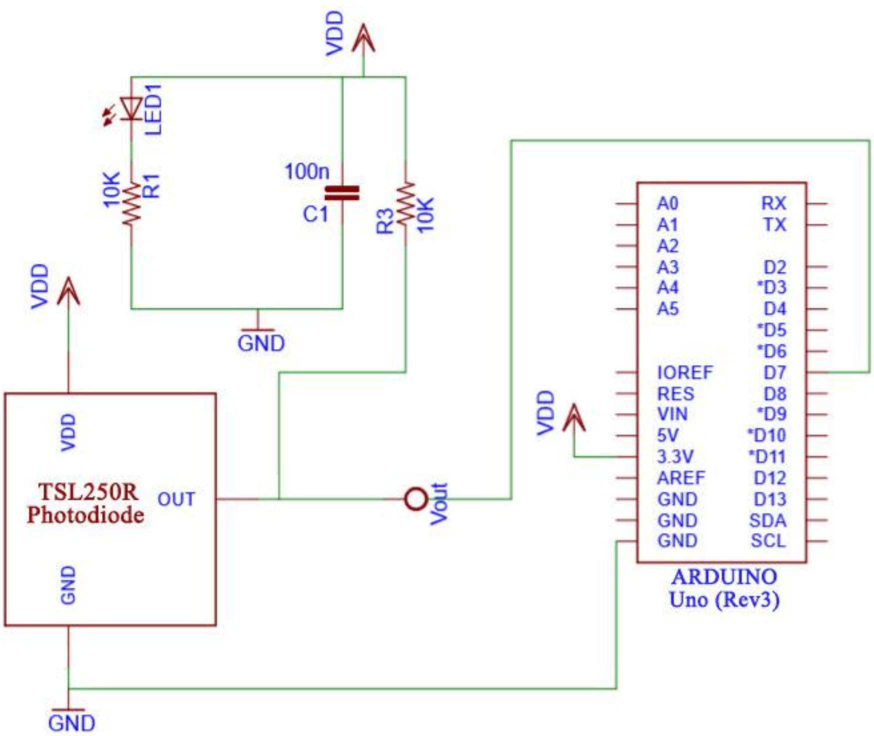

Fig. 17. The receiver circuit design diagram

2.2.3. Experiment operation and discussion

Regardless of the design and components of the implemented prototype discussed earlier, the operation of the experiment and the work of the microcontroller code must be explicitly explained. This creates a better understanding of how the model operates and highlights the impact of the Arduino and shows how smooth is using it in many applications. Fig. 18 illustrates the main steps of this experimental operation, including the function of the microcontroller.

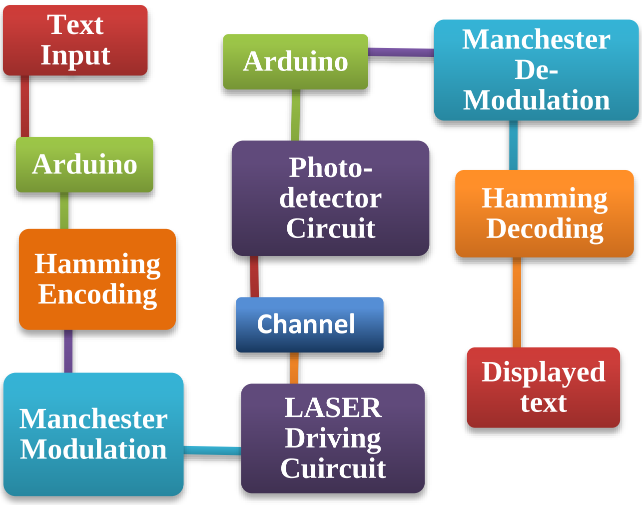

Fig. 18. Model operational steps diagram

After entering the text characters through the transmitting computer into the Arduino serial monitor window, the Arduino program converts each character according to its ASCII code [11] to its binary representation in one byte. Then, each byte is splitted into two nibbles, the most significant bit “MSB” and the least significant bit “LSB”. The following tables are indicative examples for the program process, considering the transmitted and received data is the character (A). Table 1 shows the representative ASCII, binary codes of the character (A) and the value of each digital byte.

Table 1. Example of data entering

Character (A)	
ASCII Code	65
Binary Code	01000001
MSB (First Nibble)	LSB (Second Nibble)
D7 D6 D5 D4	D3 D2 D1 D0
0100	0001

Encoding

The first and second nibbles are encoded using Hamming encoding, producing an unsigned 8-bit integer for each nibble. In each nibble, the extra 4 (P0, H0, H1 & H2) bits include the initial nibble data to help in error detection and initial bits recovery. The two encoded nibbles are concatenated to provide a 16-bit integer.

Tables 2 and 3, show the encoding process of sending the character (A), the original 8 bits of the character (A), binary code and the 16 bits generated from the encoding phase.

Here, we have the equations that produce the extra 4 bits

H0 = D0 + D1 + D3

H1 = D0 + D2 + D3

H2 = D1 + D2 + D3

where the parity bit P0 is the XOR sum of the other bits. P0 = D0 + D1 + D2 + D3 + P0 + P1 + P2

Table 2. First nibble syndrome matrix of character (A) example

D3	D2	D1	D0	H2	H1	H0	P0	Input:
1	1	1	1	1	1	1	1	0100
1	1	1	0	1	0	0	0	Output:
1	1	0	1	0	1	0	0	01001110
1	0	1	1	0	0	1	0	

Table 3. Second nibble syndrome matrix of character (A) example

D3	D2	D1	D0	H2	H1	H0	P0	Input:
1	1	1	1	1	1	1	1	0001
1	1	1	0	1	0	0	0	Output:
1	1	0	1	0	1	0	0	00011011
1	0	1	1	0	0	1	0	

Modulation

The 16-bits are splitted into two bytes; LSB and MSB. Then, each byte is modulated using Manchester modulation by adding two start bits and one stop bit, to indicate the actual data to the receiver, providing 11 bits represented in two parts. The first is the initial data X-ORed with 1 and the second is X-ORed with 0 instead. This adds a clock pulse to the signal and delivers 22 bits for each byte. Each nibble (22 bits each) concatenated the LSBs first giving 44 bits of data to be transmitted. The IEEE 802 convention states that Manchester encodes a “0” with a transition from a high voltage level to a low voltage level “10”, in the middle of an interval. While a “1” is encoded with a transition from a low to a high voltage level ”01”, whether or not there are transitions in the middle of each interval, as shown in Fig. 19.

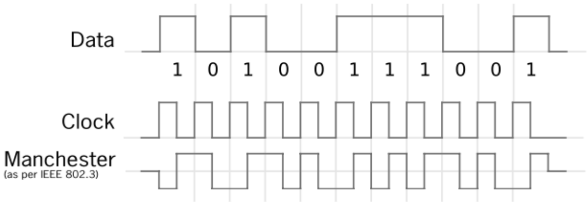

Fig. 19. Manchester modulation representation [10]

The 44- bits data is carried by the 650 nm laser through the free space (atmospheric channel), and is then detected by the TSL250R photodiode and Manchester demodulation reconstructs the original 16-bits encoded signal. Later, the 16-bits demodulated signal is decoded to reproduce the original byte. Eventually, the data will be converted to its ASCII representation and the text is printed on the Arduino serial monitor window.

Table 4 represents the modulation process in case of sending the character (A), the original 8 digital bits, the 16 bits after Hamming encoding, adding two start bits and one stop bit. This is done for the receiver to distinguish the actual data and the 44 bits produced from Manchester modulation.

Table 4. Sequence of Manchester modulation character (A) example

	LSB	MSB
Original Input	0100	0001
Hamming Encoding Output	01001110	00011011
Adding start and stop bits	00010011100	00000110110
Manchester Output	XOR with 1	XOR with 0
Add a clock pulse to the signal		
Manchester modulation	1010100110100101 011010	1010101010010110010110

2.2.4. Results

In this experiment, an error-free text transmitting and receiving model prototype is achieved and experimented repeatedly, with a data rate of 125 kbps. Both transmitted and received signals are displayed on a DSO1052B digital oscilloscope as shown in Figs. 20 and 21. Both figures show the transmitted and received text signals, with different time base, 200μs/div and 500μs/div, respectively.

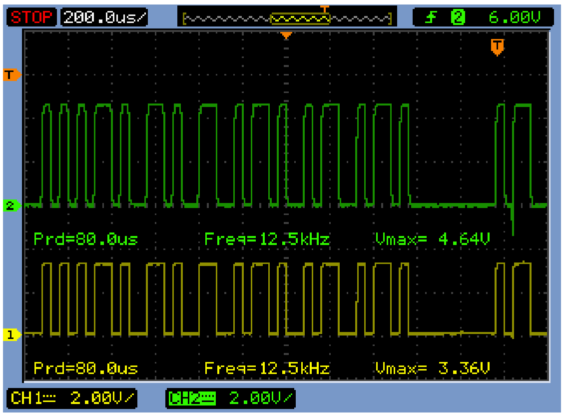

Fig. 20. Transmitted and received text signals

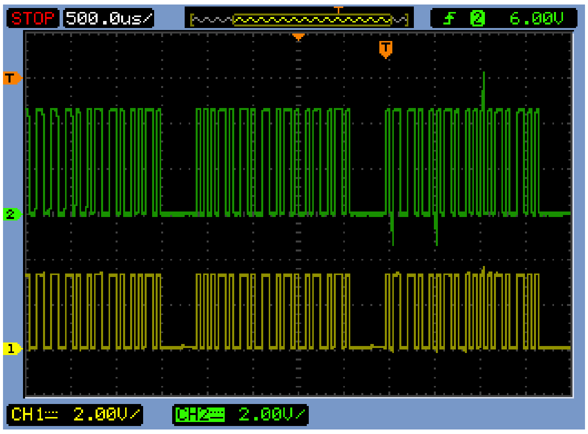

Fig. 21. Transmitted and received text signals

The first signal represents the received signal and the second signal represents the transmitted signal. However, this model can be generalized and integrated with that of the audio signal transmission. The transmitted text could be converted and displayed as an audio signal, with a different light source (LED) and some manageable adjustments on the driving circuit utilizing a LiFi model with the same principles.

3. Conclusion

In this paper, a real-time audio and text transmission prototypes using a laser source are practically implemented. The audio data transmission model achieved high-quality data transmission with a data rate of 506.3 kpbs, using a coded Arduino in the transmitter to fetch the data from the SD card module and deliver it to the laser driving circuit. The experiment reveals that adding a programmed Arduino microcontroller board to the receiving system improves the quality of received data.

The one-way data communication experimental model achieved a free error data transmission with the 125 kpbs data rate, using an Arduino attached to the transmitter and receiver circuits. The Arduino boards are programmed to accept data from the input serial window and process it using Hamming encoding and Manchester modulation. It is then sent via the laser circuit to be received by the TSL250R photodiode. Later, the data is processed by the receiver Arduino and is displayed on the Arduino serial window. These prototypes create an FSO communication link designed for audio and text transmission. The concept discussed can be generalized for real optical communication systems rather than lab work only.

References

[1] M. Aminikashani, Optical Wireless Communications: Theory and Applications, Ph.D. Thesis, The Pennsylvania State University, 2016.
[2] R. Hou, Y. Chen, J. Wu, H. Zhang, A Brief Survey of Optical Wireless Communication, The 13th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2015), Sydney, Australia, p. 41, 2015.
[3] Z. Ghassemlooy, W. Popoola, S. Rajbhandari, Optical Wireless Communications: System and Channel Modelling with MATLAB, CRC Press, ISBN 9781138074804, 2017.
[4] O. Wilfert, H. Henniger, Z. Kolka, Optical communication in free space, The 16th Polish-Slovak-Czech Optical Conference on Wave and Quantum Aspects of Contemporary Optics, Proc. SPIE **7141**, 714102 (2008).
[5] A. Phatak, M. Suthar, P. Virulkar, N. Yadav, Image and video transmission using LED, IJCA National Conference on Role of Engineers in Nation Building NCRENB, p. 16, 2015.
[6] N. Serafimovski, LiFi - How every LED light could connect you to the Internet | Nikola Serafimovski | TEDxBucharest http://iph.heliohost.org/cgi-bin/videoclip.pl?key=O-ncPjD1IeY.
[7] J. A. Rios Chavez, C. Aquino Ruiz, S. A. Garcia Sanchez, Audio watermarking of WAV files by echo modulation, The 22nd International Conference on Electrical Communications and Computers CONIELECOMP 2012, Cholula, Puebla, p. 350, 2012.
[8] TSL250R light-to-voltage optical sensors datasheet https://www.distrelec.de/Web/Downloads/_t/ds/tsl250r-2r_eng_tds.pdf.
[9] Arduino—Homepage. http://www.arduino.cc.
[10] A. M. Elzain, A. J. Alzubaidi, International Organization of Scientific Research 20, P a g e Design of Manchester Digital to Digital Encoding Data Transmission. IOSR Journal of Engineering **5**, 20 (2017).
[11] Tommy, R. Siregar, I. Lubis, A. Marwan, A. Mahmud, M. Harahap, A Simple Compression Scheme Based on ASCII Value Differencing. Journal of Physics: Conference Series **1007**, 012022 (2018).

*Corresponding author: tarek.el-dallal@aast.edu