# MOBILITY CANE

**A PROJECT REPORT**

***Submitted by***



| | |
|---|---|
| **Nivetha R** | **113220051018** |
| **Tamizharasi R** | **113220051031** |
| **Pugazhendhi R** | **113220051021** |



**BACHELOR OF ENGINEERING**

***in***

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**VELAMMAL ENGINEERING COLLEGE, CHENNAI**

**ANNA UNIVERSITY: CHENNAI 600 025**

**JUNE 2022**

# BONAFIDE CERTIFICATE

Certified that this project report "**MOBILITY CANE**" is the bonafide work of **"NIVETHA.R(**113220051018**), TAMIZHARASI.R(**113220051031**), PUGAZHENDHI.R(**113220051021**)",** who carried out the project work under my supervision .

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| **Dr.R.UMAMAHESWARI** | **Mrs.K.VIDHYA** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| | **Assistant professor-III** |
| Electrical and Electronics Engineering | Electrical and Electronics Engineering |
| Velammal Engineering College | Velammal Engineering College |
| Chennai-600 066. | Chennai - 600 066. |

## CERTIFICATION FOR EVALUATION

**College Name : 1132 Velammal Engineering College**

**Branch & Semester : 105 Electrical and Electronics Engineering /VIII**

| Names of the students | Title of the project | Name of the Supervisor with Designation |
|---|---|---|
| NIVETHA R<br>TAMIZHARASI R<br>PUGAZHENDHI R | MOBILITY CANE | Mrs. K. VIDHYA<br>Assistant professor-III |

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Engineering degree in **ELECTRICAL AND ELECTRONICS ENGINEERING** of Anna University was confirmed to be report of the work done by the above students and then evaluated**.**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere thanks to **Shri.M.V.Muthuramalingam,** Chairman and **Mr.M.V.M.Velmurugan,** Chief Executive Officer of Velammal Engineering College for their support during our project work.

We thank **Dr.S.Satishkumar, Principal**,Velammal Engineering College for his motivation and support during the course of the project.

We are very much grateful to **Dr.R.Umamaheswari**, **Professor and Head**, Department of Electrical and Electronics Engineering, Velammal Engineering College for the encouragement and useful suggestions during this work.

We are very much indebted to our Internal Guide ,**Mrs.K.Vidhya,Assistant Professor**, Department of Electrical and Electronics Engineering, Velammal Engineering College for her admirable guidance and constant encouragement throughout the project work.

We would like to extend our fullest gratitude to our Project Coordinator, **Dr.S.Premalatha** and project review committee for their timely advice and successful grooming.

We take this opportunity to thank our parents, all faculty and staff of our department for their kind help during the course of work.

# ABSTRACT

The visually impaired people usually face many difficulties when trying to interact with their environment. Physical movement is a big challenge for them because it can become tricky to distinguish where they are and how to move from one place to another.

A variety of assistive tools are available to improve the mobility of the visually impaired people, there is still much scope for improvement. They find difficulty in doing their everyday tasks and detecting objects in front of them that can be harmful to them. These people experiencing special risks and hazards in road traffic. As a solution to this problem, we introduce a Mobility Cane for assisting visually impaired people.

The Mobility Cane comes as a proposed solution to help visually impaired people to detect obstacles and investigate dangers in the world around while walking and also to cross the traffic signal without any accident. The system is designed to act like an artificial vision and alarm unit.

It consists of ultrasonic, LED, TCS 230Colour detecting sensor and ISD1820 Voice recorder module in addition to the microcontroller (Arduino Uno) to receive the sensors signals and process them to short pulses to the Arduino pins where buzzer, TCS 230, ISD1820 Voice recorder module and LED are connected. We seek in our project to provide a helpful service with affordable price.

The stick activates an alert system in case of loss. Several test cases prove that the functionalities introduced with the stick are performing correctly.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF ABBREVIATION

| S.NO | ABBREVIATION | EXPANSION |
|---|---|---|
| 1 | LED | Light Emitting Diode |
| 2 | USB | Universal Serial Bus |
| 3 | IDE | Integrated Development Environment |
| 4 | ICSP | In Circuit Serial Programming |
| 5 | AC | Alternative Current |
| 6 | DC | Direct Current |
| 7 | PWM | Pulse Width Modulation |
| 8 | TQFP | Thin Quad Flat Pack |
| 9 | QFN | Quad Flat No-Lead |
| 10 | SPI | Serial Peripheral Interface |

## LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 PROBLEM STATEMENT

According to World Health Organization statistics, there are around 285 million people who are visually impaired globally, with 39 million blind and about 246 million have low vision.

The visually impaired people need help mostly to carry out their daily lives routine, especially navigation. When visually impaired subjects are in new or unknown places, the most important thing for them is to know the road surfaces, the position of the obstacle and other interference to enable a secure and safe navigation.

Meanwhile, some of these subjects use guidance dogs or humans as a guide. However, this method is expensive and very difficult to maintain. An alternative to dogs and human guides is the use of a stick or cane to detect an object without any prior information about the environmental situation.

## 1.2 EXISTING SYSTEM

In the existing system, the smart blind stick with ultrasonic sensor detects the object in front of the person and give response to the user by alarm from the buzzer. So, the person can walk without any fear. The another LDR sensor are used in the stick to identify the day and night for the blind people. The microcontroller (Arduino Uno R3) to receive the sensor signals and process them to short pulses to the Arduino  pins where buzzers are connected. The another water sensor are used in the stick to detect water. The system comes with a wireless RF remote control that uniquely sounds buzzier when pressed, that helps in locating the stick in case it gets misplaced. This device will be best solution to overcome their difficulties and help them to live the better life.

## DISADVANTAGES

1. Pits and bumps of the road cannot be detected using this device.
2. Smart stick is unfoldable.

## 1.3 PROPOSED SYSTEM

The use of Arduino technology in the construction of a Mobility cane that can be used by people suffering from blindness and help them to overcome in the drawbacks of this condition. In this project, we used the power of Arduino along with sensors to construct a simple Mobility cane that can detect obstacles (solid bodies), water, and light, and use a sound alarm (buzzer) as outputs to warn the users.

Ultrasonic sensor can detect forward obstacles at around 1-meter-high above ground level. We suggest a solution to design obstacle detection unit with multiple ultrasonic sensors in which can sense obstacles in different heights and angles.

TCS230 sense the light of traffic signal and gives an alert to help people suffering from Nyctalopia (Night Blindness).

In addition, ISD1820 Voice recorder module gives the recorded audio when it senses the traffic signal and it alerts the users. This cane can improve the quality of the visually impaired people's lives, save efforts and time.

### 1.3.1 PROPOSED SYSTEM ADVANTAGES

1. Auto detection and alarming system.
2. Helps the blind to cross the traffic signal.
3. Cost efficient and effective.
4. Helps in future for any immediate causality help.
5. Speech message will help the blind and physically disables to recognize the destination.

# CHAPTER 2

# PROJECT DESCRIPTION

## 2.1 INTRODUCTION

Visually impaired people are the people who can't identify smallest detail with healthy eyes. Those who have the visual acuity of 6/60 or the horizontal extent of the visual field with both eyes open less than or equal to 20 degrees, these people are considered blind. Such people are in need of aiding devices for blindness related disabilities. As described in 10% of blind have no usable eyesight at all to help them move around independently and safely. The electronic aiding devices are designed to solve such issue.

To record information about the obstacles presence in a road, active or passive sensors can be used. In case of a passive sensor, the sensor just receives a signal. It detects the reflected, emitted or transmitted electro-magnetic radiation provided by natural energy sources. In case of using an active sensor, the sensor emits a signal and receives a distorted version of the reflected signal. It detects reflected responses from objects irradiated with artificially generated energy sources. These kind of active sensors are capable of sensing and detecting far and near obstacles. In addition, it determines an accurate measurement of the distance between the blind and the obstacle.

In this project, we used the power of Arduino along with sensors to construct a simple Mobility cane that can detect obstacles (solid bodies), water, and light, and use a sound alarm (buzzer) as outputs to warn the users.

Ultrasonic sensor can detect forward obstacles at around 1-meter-high above ground level. TCS230 sense the light of traffic signal. ISD1820 Voice recorder module gives the recorded audio when it senses the traffic signal and it alerts the users.

## 2.2 BLOCK DIAGRAM

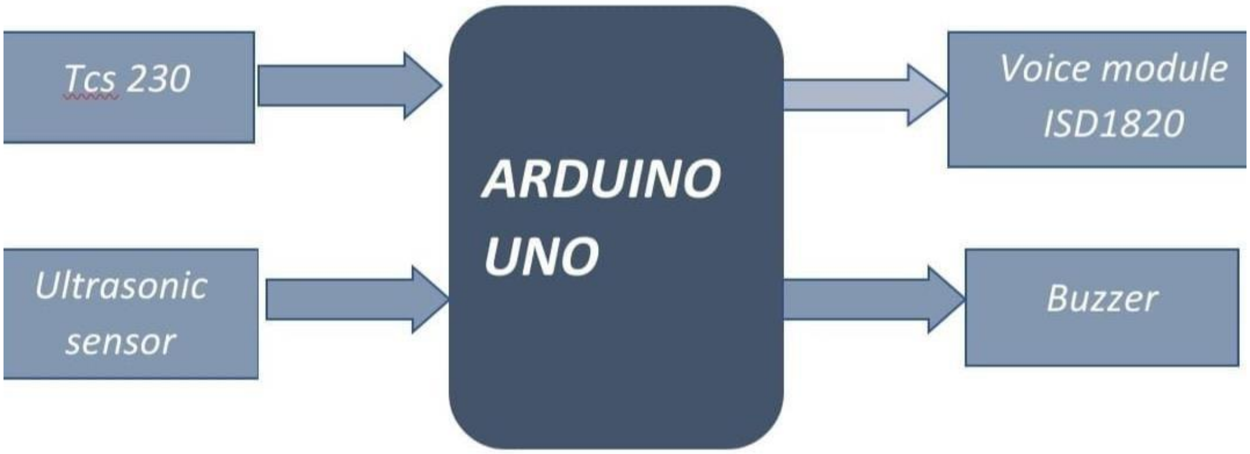



**Figure 2.1 BLOCK DIAGRAM**

# CHAPTER 3

# SYSTEM CONFIGURATION

## 3.1 HARDWARE DESCRIPTION

### 3.1.1 ARDUINO UNO

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures Single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet).It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

**Figure 3.1 ARDUINO UNO**

Arduino is an open-source project that created microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices. The project is based on microcontroller board.

These systems provide sets designs, produced by several vendors, using various microcontrollers of digital and analog input/output (I/O) pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers.

For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on a programming language named Processing, which also supports the languages C and C++.

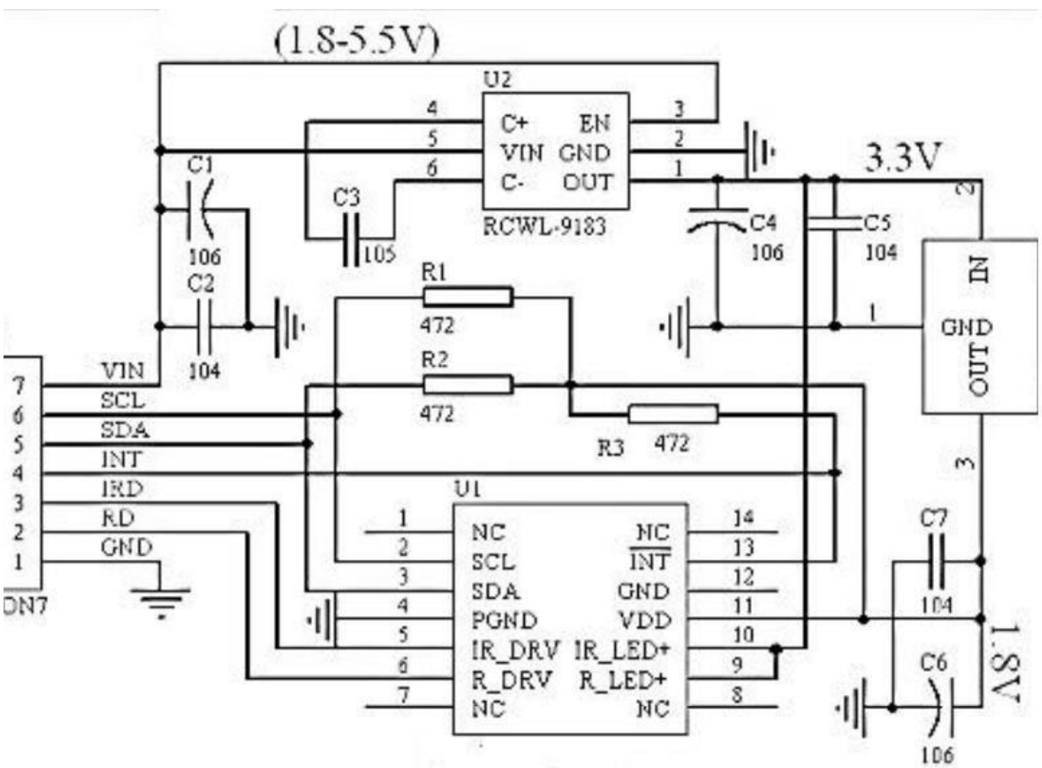



**Figure 3.2 INTERFACE OF ARDUINO UNO**

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter. Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers.

## ARDUINO UNO SPECIFICATION

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 68.6 mm
- Width: 58.4 mm
- Weight: 25 g

## FEATURES

- Microcontroller: ATmega328P
- Operating voltage: 5V
- Input voltage: 7-12V
- Flash memory: 32KB

- SRAM: 2KB
- EEPROM: 1KB

### 3.1.2 ATMEGA328 IC

The ATmega328 isa single-chip microcontroller created by Atmel in the mega AVR family. The Atmel 8-bit AVR RISC-based microcontroller combines 32kB ISP flash memory with read-while-write capabilities, 1kB EEPROM, 2kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port,6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes.

The device operates between 1.8-5.5 volts. The device achieves through put approaching 1 MIPS per MHz.

**Figure 3.3 ATMEGA328P IC**

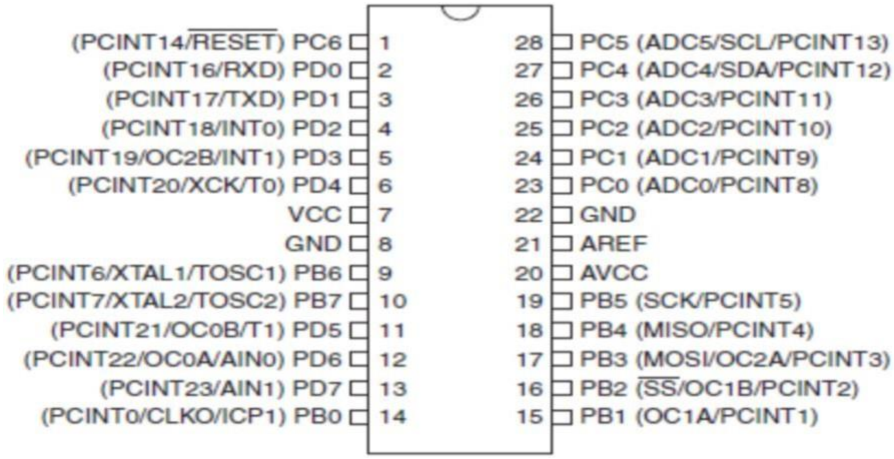



**Figure 3.4 ATMEGA328P IC PIN DIAGRAM**

- High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family
- Advanced RISC Architecture
- 131 Powerful Instructions
- Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 20 MIPS Throughput at 20MHz
- On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
- 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
- 256/512/512/1KBytes EEPROM
- 512/1K/1K/2KBytes Internal SRAM
- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM C(1)°C/100 years at25º– Data retention: 20 years at 85
- Optional Boot Code Section with Independent Lock Bits In-SystemProgramming by On-chip Boot Program

- True Read-While-Write Operation
- Programming Lock for Software Security Atmel® QTouch® library support
- Capacitive touch buttons, sliders and wheels– QTouch and QMatrix®acquisition
- Up to 64 sense channels
- Peripheral Features
- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Six PWM Channels
- 8-channel 10-bit ADC in TQFP and QFN/MLF package
- Temperature Measurement
- 6-channel 10-bit ADC in PDIP Package
- Temperature Measurement Programmable Serial USART–Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

### 3.1.3 TCS230 COLOUR DETECTING SENSOR

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance).

The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters.

The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance.


**Figure 3.5 TCS230 COLOR DETECTING SENSOR**

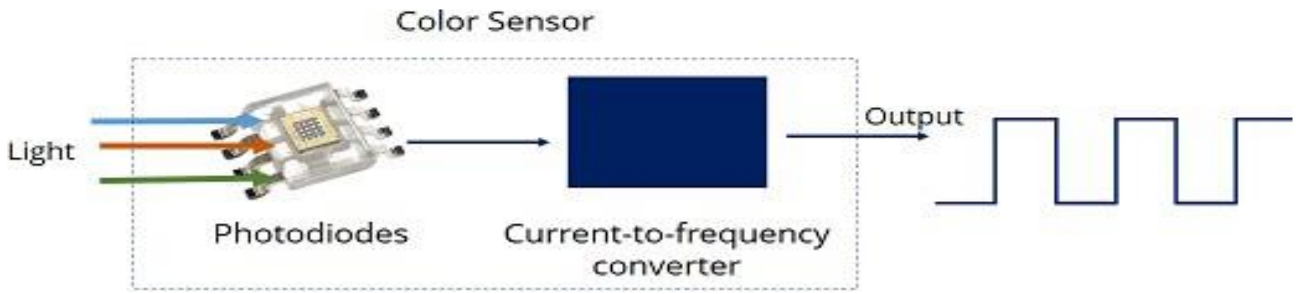



**Figure 3.6 FUNCTIONAL BLOCK DIAGRAM OF TCS230**

**PIN CONFIGURATION**

The TCS3200 has an array of photodiodes with 4 different filters. A photodiode is simply a semiconductor device that converts light into current. The sensor has

- 16 photodiodes with red filter – sensitive to red wavelength
- 16 photodiodes with green filter – sensitive to green wavelength
- 16 photodiodes with blue filter – sensitive to blue wavelength
- 16 photodiodes without filter

The sensor has a current-to-frequency converter that converts the photodiodes' readings into a square wave with a frequency that is proportional to the light intensity of the chosen color. This frequency is then, read by the Arduino.

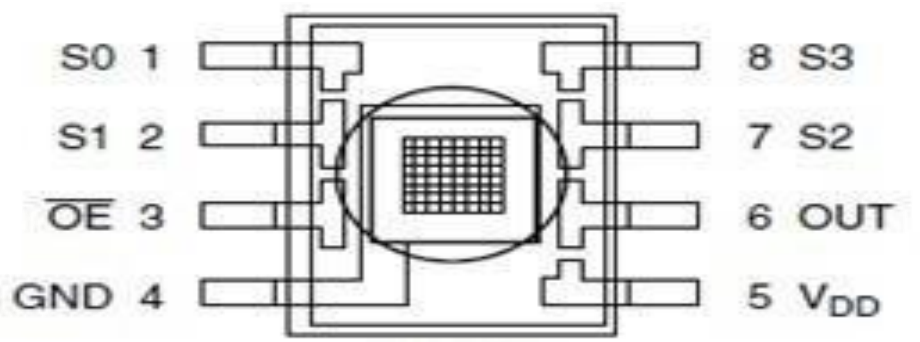



**Figure 3.7 PIN DIAGRAM OF TCS230 COLOR DETECTING SENSOR**

| S.NO | PIN NAME | I/O | DESCRIPTION |
|---|---|---|---|
| 1. | GND (4) | | Power supply ground |
| 2. | OE(3) | I | Enable for output frequency |
| 3. | OUT(6) | O | Output frequency |
| 4. | S0,S1(1,2) | I | Output frequency scalling |
| 5. | S2,S3(7,8) | I | Phptpdiode type selection inputs |
| 6. | VDD(5) | | Voltage supply |

**Table 3.1 Pin Description**

**SPECIFICATION**

- Power: 2.7V to 5.5V
- Size: 28.4 x 28.4mm (1.12 x 1.12″)
- Interface: digital TTL
- High-resolution conversion of light intensity to frequency
- Programmable color and full-scale output frequency
- Communicates directly to microcontroller

### 3.1.4 ISD1820 VOICE RECORDER MODULE

Voice Record Module is base on ISD1820, which a multiple-message record/playback device. It can offers true single-chip voice recording, no-volatile storage, and playback capability for 8 to 20 seconds. The sample is 3.2k and the total 20s for the Recorder.

This module use is very easy which you could direct control by push button on board or by Microcontroller such as Arduino, STM32, Chip Kit etc. From these, you can easy control record, playback and repeat and so on.

**FEATURES**

1. Push-button interface, playback can be edge or level activated.
2. Automatic power-down mode.
3. On-chip 8Ω speaker driver.
4. Signal 3V Power Supply.
5. Can be controlled both manually or by MC.
6. Sample rate and duration changable by replacing a single resistor.
7. Record up to 20 seconds of audio.
8. Dimensions: 37 x 54 mm.

**APPLICATION**

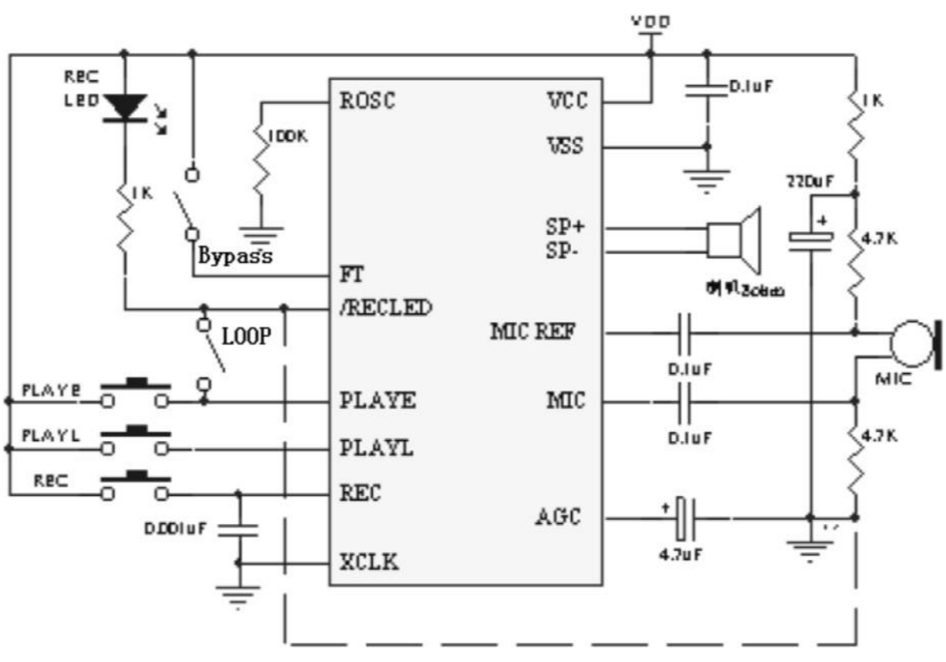



**Figure 3.8 Interface of ISD1820 Voice Recorder Module**

If you want change record duration, an external resistor is necessary to select the record duration and sampling frequency, which can range from 8 – 20 seconds (4-12kHz sampling frequency). The Voice Record Module of our provide default connect 100k resistor by short cap. So the default record duration is 10s.

| ROSC | Duration | Sample Rate | Bandwidth |
|---|---|---|---|
| 80K Ω | 8 secs | 8. 0KHz | 3. 4KHz |
| 100K Ω | 10 secs | 6. 4KHz | 2. 6KHz |
| 120K Ω | 12 secs | 5. 3KHz | 2. 3KHz |
| 160K Ω | 16 secs | 4. 0KHz | 1. 7KHz |
| 200K Ω | 20 secs | 3. 2KHz | 1. 3KHz |

**TABLE 3.2 RECORDING**

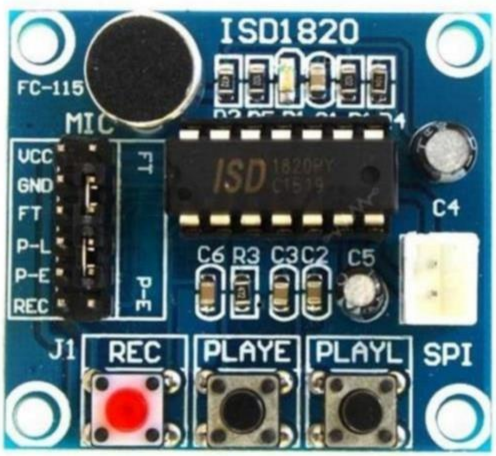



**Figure 3.9 ISD1820 Voice Recorder Module**

1. ***VCC***– 3.3V power supply
2. ***GND***– Power ground

3. **REC** – The REC input is an active-HIGH record signal. The module starts recording whenever REC is HIGH. This pin must remain HIGH for the duration of the recording. REC takes precedence over either playback(PLAYL or PLAYE) signal.
4. ***PLAYE*** – Playback, Edge-activated: When a HIGH-going transition is detected on continues until an End-of-Message (EOM) marker is encountered or the end of the memory space is reached.
5. ***PLAYL*** – Playback, Level-activated, when this input pin level transits for LOW to HIGH, a playback cycle is initiated.
6. ***Speaker Outputs*** – The SP+ and SP- pins provide direct drive for loudspeakers with impedances as low as 8Ω.
7. ***MIC*** – Microphone Input, the microphone input transfers its signals to the on-chip preamplifier.
8. ***FT*** – Feed Through: This mode enable the Microphone to drive the speaker directly.
9. ***P-E*** – Play the records endlessly.

## RECORD OPERATE GUIDE

1. Push REC button then the RECLED will light and keep push until record end.
2. Release the REC button
3. Select Playback mode: PLAYE, just need push one time, and will playback all of the record or power down ; PLAYL, you need always push this button until you want to stop playback record or end ; When short P-E jumper the record will playback time a time until jumper off or power down.
4. FT mode, when short FT jumper, that means all of you speak to MIC will direct playback to speaker.

**POWER AMPLIFIER CIRCUIT**

If you want extern power amplifier circuit to Speakers, you can use LM386, D2283, D2322, TA7368, MC34119 etc amplifier IC. Note, SP+ or SP- is you do not want to use, must vacant, do not connect to GND. Used LM386 power amplifier circuit as below.

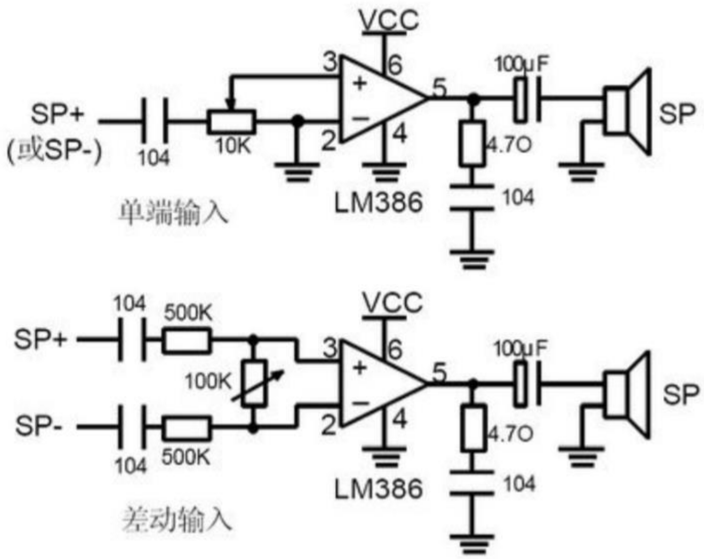



**Figure 3.10 Power Amplifier Circuit of ISD1820**

### 3.1.5 ULTRASONIC SENSOR HC-SR04

An HC-SR04 ultrasonic distance sensor actually consists of two **ultrasonic transducers.** One acts as a transmitter that converts the electrical signal into 40 KHz ultrasonic sound pulses. The other acts as a receiver and listens for the transmitted pulses.

When the receiver receives these pulses, it produces an output pulse whose width is proportional to the distance of the object in front.

This sensor provides excellent non-contact range detection between 2 cm to 400 cm (~13 feet) with an accuracy of 3 mm. Since it operates on 5 volts, it can be connected directly to an Arduino or any other 5V logic microcontroller.

**TECHNICAL SPECIFICATIONS**

| | |
|---|---|
| **1. Operating Voltage** | DC 5V |
| **2. Operating Current** | 15mA |
| **3. Operating Frequency** | 40KHz |
| **4. Max Range** | 4m |
| **5. Min Range** | 2cm |
| **6. Ranging Accuracy** | 3mm |
| **7. Measuring Angle** | 15 degree |
| **8. Trigger Input Signal** | 10 µS TTL pulse |
| **9. Dimension** | 45 x 20 x 15mm |

**HC-SR04 ULTRASONIC SENSOR PINOUT**


**Figure 3.10 HC-SR04 ULTRASONIC SENSOR**

- ***VCC*** supplies power to the HC-SR04 ultrasonic sensor. You can connect it to the 5V output from your Arduino.
- ***Trig (Trigger)*** pin is used to trigger ultrasonic sound pulses. By setting this pin to HIGH for 10µs, the sensor initiates an ultrasonic burst.
- ***Echo*** pin goes high when the ultrasonic burst is transmitted and remains high until the sensor receives an echo, after which it goes low. By measuring the time the Echo pin stays high, the distance can be calculated.
- ***GND*** is the ground pin. Connect it to the ground of the Arduino.

## WORKING OF HC-SR04 ULTRASONIC DISTANCE SENSOR

- It all starts when the trigger pin is set HIGH for 10µs. In response, the sensor transmits an ultrasonic burst of eight pulses at 40 kHz.
- This 8-pulse pattern is specially designed so that the receiver can distinguish the transmitted pulses from ambient ultrasonic noise.
- These eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile the echo pin goes HIGH to initiate the echo-back signal.
- If those pulses are not reflected back, the echo signal times out and goes low after 38ms (38 milliseconds). Thus a pulse of 38ms indicates no obstruction within the range of the sensor.
- If those pulses are reflected back, the echo pin goes low as soon as the signal is received. This generates a pulse on the echo pin whose width varies from 150 µs to 25 ms depending on the time taken to receive the signal.

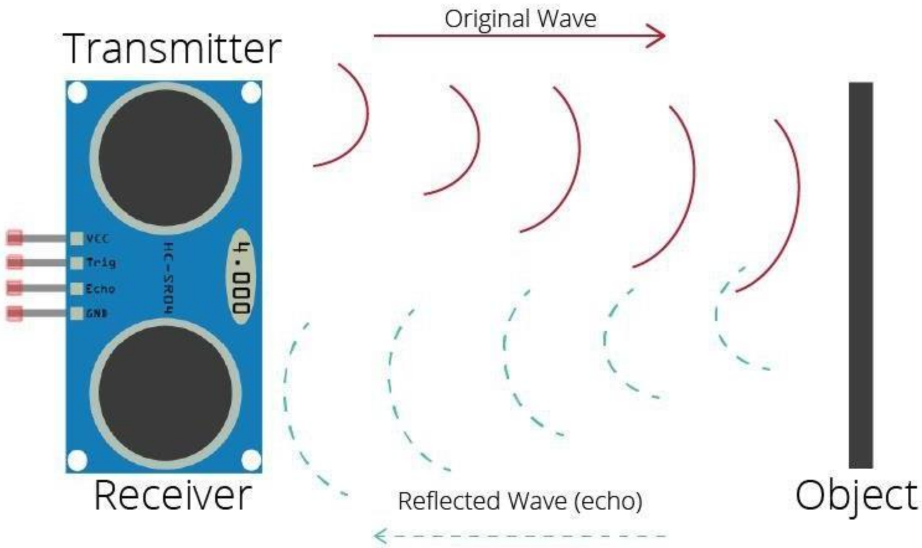
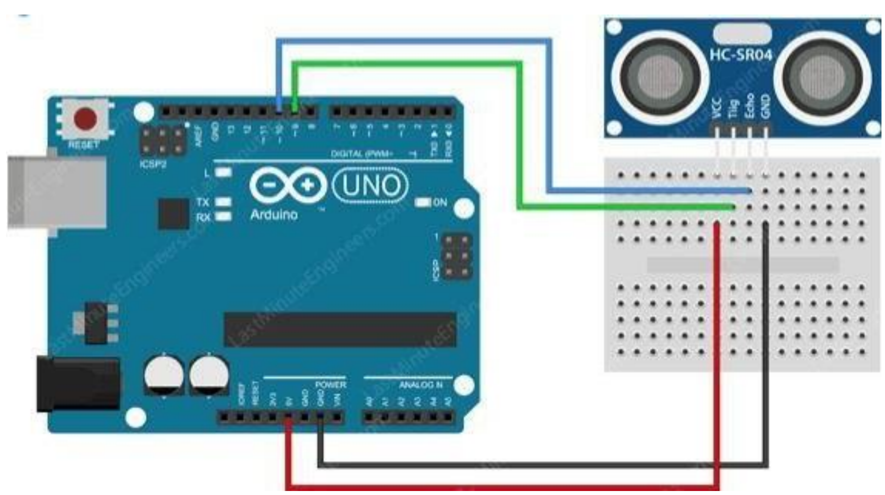



**Figure 3.12  WORKING OF ULTRASONIC SENSOR**

## WIRING AN HC-SR04 SENSOR TO AN ARDUINO

- Connecting the HC-SR04 to Arduino is very easy.
- Start by placing the sensor on your breadboard.
- Connect the VCC pin to the 5V pin on the Arduino and the GND pin to the ground pin. Now connect the trig and echo pins to digital pins #9 and #10 respectively.

**Figure 3.13 WIRING WITH ARDUINO UNO**



## 3.2 SOFTWARE DESCRIPTION

### 3.2.1 ARDUINO IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors.

The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Before uploading your sketch, you need to select the correct items from the **Tools> Board** and **Tools** > Port menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1** (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial devices in the ports section of the Windows Device Manager. On Linux, it should be /**dev/ttyACM , /dev/ttyUSBx** or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

# CHAPTER 4

## HARDWARE IMPLEMENTATION

Implementation of the circuit on a printed circuit board is not going to be fruitful, as most of the circuit parts would be implemented on a wooden chassis/cardboard/plastic mounting. We can't hardwire the sensors on PCB as they would be in a vertical position; and more so placed in a circular /semi-circular fashion. The final assembled device is placed about midway on the stick.

After that, make a cardboard box or a plastic box with 8 circular holes in it. 6 of the 8 holes should be of the size of transmitter/receiver of the ultrasonic sensors,tcs230 sensor,ISD1820 sensor & they should be punched on the wall of the box. Remaining 2 holes should be of the size greater than the diameter of the wooden stick you choose & they should be punched on the top & bottom of the box, also they should be collinear to each other. The top of the box should be removable if in case we need the access to circuit.

Once the circuit is assembled, put the Arduino & sensor along with the connecting wires inside the cardboard box. Through the six holes punched on the surface take the transmitter & receiver set of each sensor out of the box. The significance of this assembly is to avoid false triggers as the box itself would behave as an obstacle to the sensors. Proper placing and alignment is very important. Ensure proper mounting by optimally placing the buzzer, vibration motor, switch & battery after removing it from the box. Test the assembly now against variable obstacles at variable distances.

**Figure 4.1 OUTPUT OF THE PROJECT**

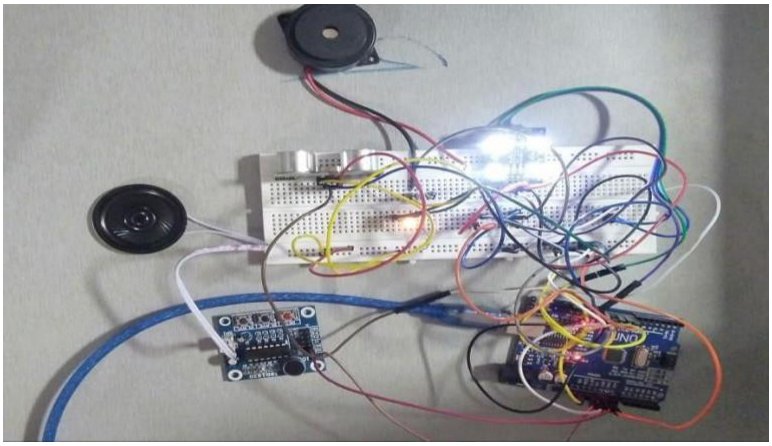


**Figure 4.2 OUTPUT COMPONENT CONNECTION**

# CHAPTER 5

# CONCLUSION

Blind people face lot of difficulties while travelling from one place to another. With the intention to help the blind, the smart blind stick is proposed**.**

It is worth mentioning at this point that the aim of the of this project is design and implementation of a smart walking stick for the blind has been fully achieved The smart stick as a basic platform for the coming generation of more adding devices to help the visually impaired to navigate safely both indoor and outdoor. It leads the good result in detecting the obstacles on the path of the user in a range, sensing the color of the traffic signal.

This project offer low cost, reliable, portable, low power consumption and robust technology for navigation with obvious short response time .In this project, different types of sensors and other component are used with the light weight.

The proposed  system takes the blind person to reach the destination without any struggle in their path. The buzzer were used to make alarm to warn them.

The system not only make them more free, but also liberate their minds and throw away many worries and doubts. However, in some specific open environment, such as on the road, the blind still need someone accompany them if they have to take a long trip. The effect of the system will reduced the dependency of other.

***FUTURE SCOPE:***

- The certain modification of sensor and programing we can detect the small pits and dumps of road.
- Water resistant can be eliminated.

# CHAPTER 6

## REFERENCES:


1. R. R. Bourne et al., "Magnitude temporal trends and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis", Lancet Glob Health, vol. 5, no. 9, pp. e888-e897, Sep. 2017.

2. J. A. Dowling, Mobility enhancement using simulated artificial human vision, 2007.

3. N. A. Giudice and G. E. Legge, "Blind navigation and the role of technology", The Engineering Handbook of Smart Technology for Aging Disability and Independence, pp. 479-500, 2008.

4. E. Milios, B. Kapralos, A. Kopinska and S. Stergiopoulos, "Sonification of range information for 3-D space perception", IEEE Trans Neural Syst Rehabil Eng, vol. 11, no. 4, pp. 416-421, 2003.

5. J. Brabyn, "Developments in electronic aids for the blind and visually impaired", IEEE Eng Med Biol Mag, vol. 4, no. 4, pp. 33-37, Dec. 1985.

6. J. Villanueva and R. Farcy, "Optical device indicating a safe free path to blind people", IEEE Trans Instrum Meas, vol. 61, no. 1, pp. 170-177, Aug. 2011.

7. R. Farcy, R. Leroux, A. Jucha, R. Damaschini, C. Grégoire and A. Zogaghi, "Electronic travel aids and electronic orientation aids for blind people: Technical rehabilitation and everyday life points of view", Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Technology for Inclusion, vol. 12, Jul. 2006.

8. H. Takizawa, K. Orita, M. Aoyagi, N. Ezaki and S. Mizuno, "A spot reminder system for the visually impaired based on a smartphone camera", Sensors, vol. 17, no. 2, pp. 291, Feb. 2017.

9. M. A. Hersh and M. A. Johnson, Assistive Technology for Visually Impaired and Blind People, London, UK:Springer, 2010.

10. Ulrich and J. Borenstein, "The GuideCane-applying mobile robot technologies to assist the visually impaired", IEEE Trans Syst Man Cybern A Syst Hum, vol. 31, no. 2, pp. 131-136, Mar. 2001.

11. J. A. Brabyn, "New developments in mobility and orientation aids for the blind", IEEE Trans Biomed Eng, vol. 4, pp. 285-289, Apr. 1982.

12. K. Ito et al., "CyARM: an alternative aid device for blind persons", CHI’05 Extended Abstracts on Human Factors in Computing Systems, pp. 1483-1488, Apr. 2005.

13. GDP Research Australia, Oct. 2019, [online] Available: http://www.gdp-research.com.au/.

14. S. Bhatlawande, M. Mahadevappa, J. Mukherjee, M. Biswas, D. Das and S. Gupta, "Design development and clinical evaluation of the electronic mobility cane for vision rehabilitation", IEEE Trans Neural Syst Rehabil Eng, vol. 22, no. 6, pp. 1148-1159, May 2014.

15. HENRY'S BENCH, "Arduino Rain Sensor Module Guide and Tutorial", HENRY'S BENCH a place for the electronic hobbyist, 2015.

16. Lady Ada, "Adafruit FONA 808 Cellular + GPS Shield for

Arduino", "Adafruit Co., 2017.

17. J. Banks; J. Carson; B. Nelson; D. Nicol, "Discrete-Event System Simulation". Prentice Hall. p. 3. , 2001, ISBN 0-13-088702-1.

18. EDAC, "About the EDA Industry", Electronic Design Automation Consortium, Archived from the original on August 2, 2015.

19. McRoberts, Michael, "Beginning Arduino", APress. pp. xx. 2010, ISBN 978-1-4302-3240-7.

20. Rubin, Steven M., "An Integrated Aid for Top-Down Electrical Design", Proceedings, VLSI '83, Anceau and Aas, eds., North Holland, Amsterdam,1983.

## APPENDIX I – SOURCE CODE

## ARDUINO CODE

```
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
int const trigPin = 10;
int const echoPin = 9;
int const buzzPin = 2;
void setup()
{
 pinMode(trigPin, OUTPUT);
 // trig pin will have pulses output
 pinMode(echoPin, INPUT);
 // echo pin should be input to get pulse width
 pinMode(buzzPin, OUTPUT);
 // buzz pin is output to control buzzering
 }
 void loop()
 {
  // Duration will be the input pulse width and distance will be the distance to the obstacle in centimeters
  int duration, distance;
```

```
// Output pulse with 1ms width on trigPin
digitalWrite(trigPin, HIGH);
delay(1);
digitalWrite(trigPin, LOW);
// Measure the pulse input in echo pin
duration = pulseIn(echoPin, HIGH);
// Distance is half the duration devided by 29.1 (from datasheet)
distance = (duration/2) / 29.1;
// if distance less than 0.5 meter and more than 0 (0 or less means over range)
if (distance <= 50 && distance >= 0)
{
 // Buzz digitalWrite(buzzPin, HIGH);
 }
 else
 {
  // Don't buzz
  digitalWrite(buzzPin, LOW);
  }
  // Waiting 60 ms won't hurt any one
  delay(60);
  }
// TCS230 or TCS3200 pins wiring to Arduino
```

```
void setup() {
  // Setting the outputs
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);

  // Setting the sensorOut as an input
  pinMode(sensorOut, INPUT);

  // Setting frequency scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);

   // Begins serial communication
  Serial.begin(9600);
}
void loop() {
  // Setting RED (R) filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
```

```
// Reading the output frequency
redFrequency = pulseIn(sensorOut, LOW);

 // Printing the RED (R) value
Serial.print("R = ");
Serial.print(redFrequency);
delay(100);

// Setting GREEN (G) filtered photodiodes to be read
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);

// Reading the output frequency
greenFrequency = pulseIn(sensorOut, LOW);

// Printing the GREEN (G) value
Serial.print(" G = ");
Serial.print(greenFrequency);
delay(100);

// Setting BLUE (B) filtered photodiodes to be read
digitalWrite(S2,LOW);
```

```
  digitalWrite(S3,HIGH);


  // Reading the output frequency
  blueFrequency = pulseIn(sensorOut, LOW);


  // Printing the BLUE (B) value
  Serial.print(" B = ");
  Serial.println(blueFrequency);
  delay(100);
}
```

# APPENDIX II – SCREENSHOTS