ME3902 O/E Report

Class: ME3902

Instructor: Ahmet Can Sabuncu

Name: Hansen Ding

Date: 12/11/2020

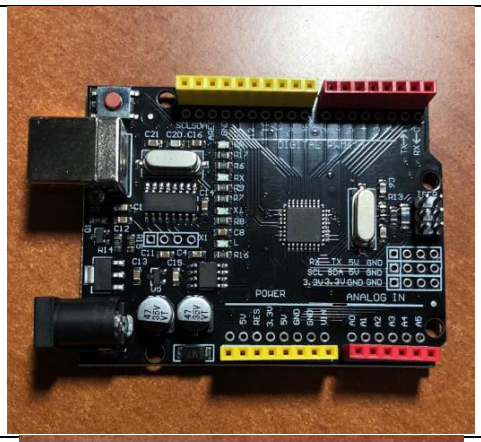
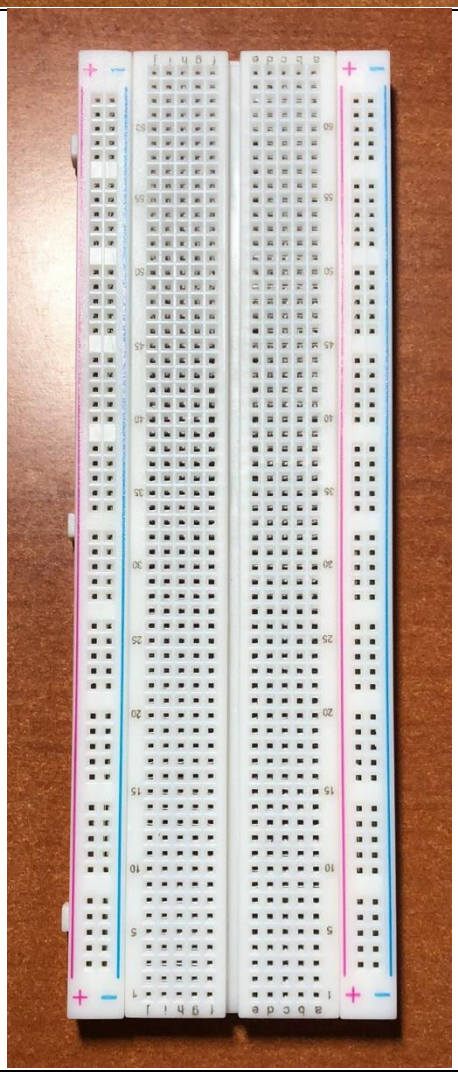
Introduction

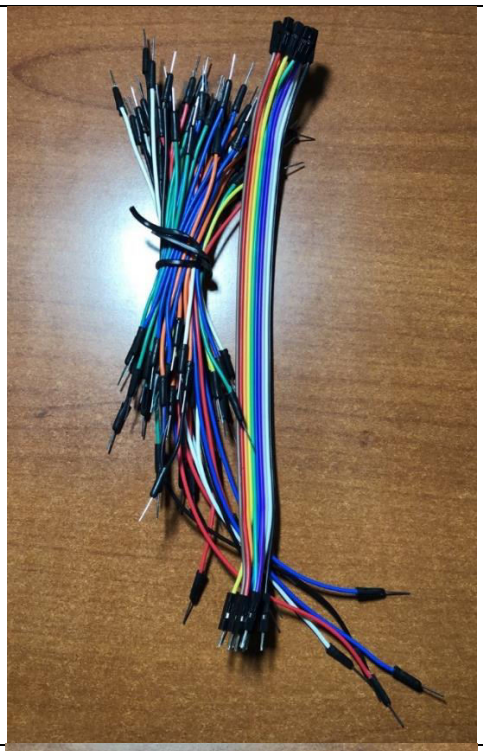
The option I chose was to build a complex experimental setup which used at least three different types of sensors/controllers/motors/etc. My project is to develop a simple audio player. Just like other audio players, there are some key functions that my personal audio player must contain: I should be able to turn the player on and off, I should be able to pause or unpause the song that is playing, and I should be able to skip the current song to the next one. What's different is that this could be applied in places like karaoke – there is a requirement that only the singer could control all the button functions. To achieve all functioning goals, there are several parts that are required. Arduino board, breadboard, and wires since everything would be built based on those. A speaker, so that after everything is set up, it can reach the main purpose of playing a music. For storing lists of songs that will be played by the player, an SD card is needed, but in order for the card to be used in the Arduino board, an SD card adapter is something that must-have as well. An IR sensor acting as a lock only when the singer activates it, all those buttons can function properly. Or otherwise, none of those mentioned functions will work. LEDs can also be included just to show which action the user has made.

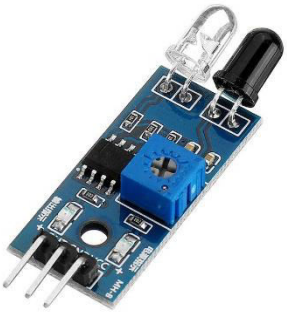
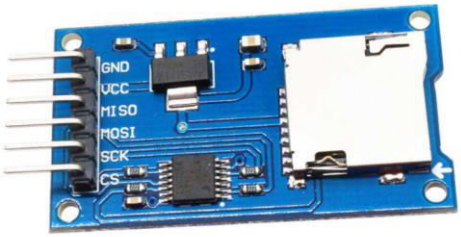
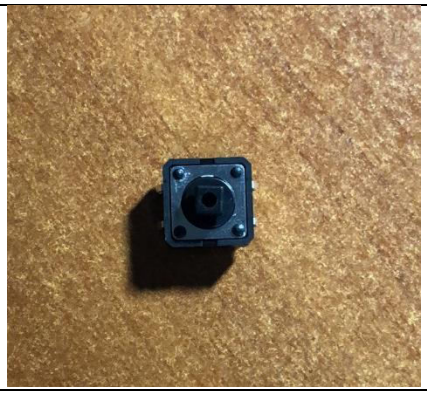
Methodology

Component List & Price

Component Name	Image	Price (USD)

Arduino Uno Board		6.79
Breadboard		2.27

Wires		0.29
Speaker		2.72

Geekcreit® IR Infrared Obstacle Avoidance Sensor		0.42
SD Card		5.89
SD Card Adapter		0.36
Button		0.18

KY-004 Pin Button Key Tactile Switch Sensor		0.16
LED		0.33
		Total: 19.41

Schematic Diagram

The system schematic diagram is shown below in Figure 1. The system includes a speaker on the top right, an ultrasonic sensor attached on the breadboard, and one SD card adapter that is not shown in the graph. The current procedure is to test all three parts functioning well.

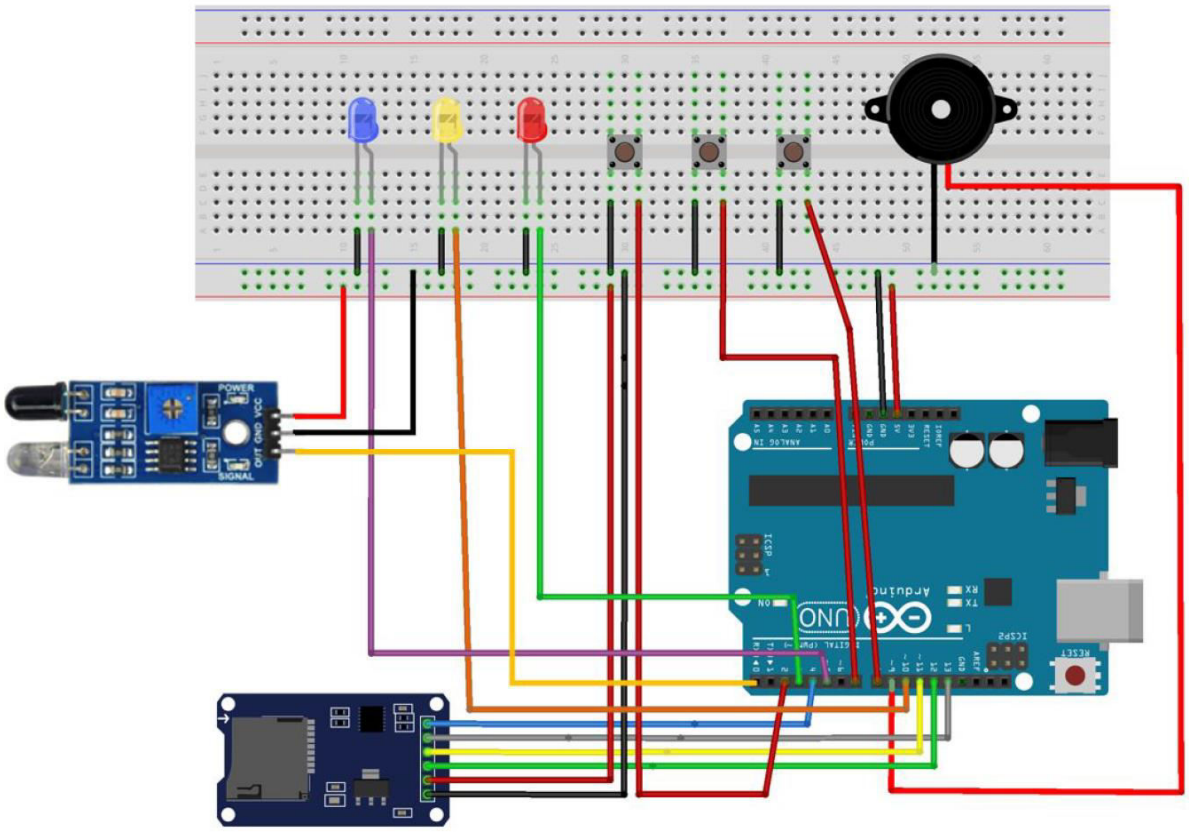


Figure 1 Schematic Circuit

Testing

Speaker

The first thing to test was the small speaker. It was also the key part of the entire system; without it, even the system run perfectly, we wouldn't hear any sound as results. To test the speaker, I put a 10K ohm resistor in the series as Figure 2 shows. Figure 3 shows the code of testing the speaker. In order to generate different sounds, just change the frequency in the "tone" line. The speaker generated 1KHz of sound every one second and stopped for one second, showing the part worked fine.

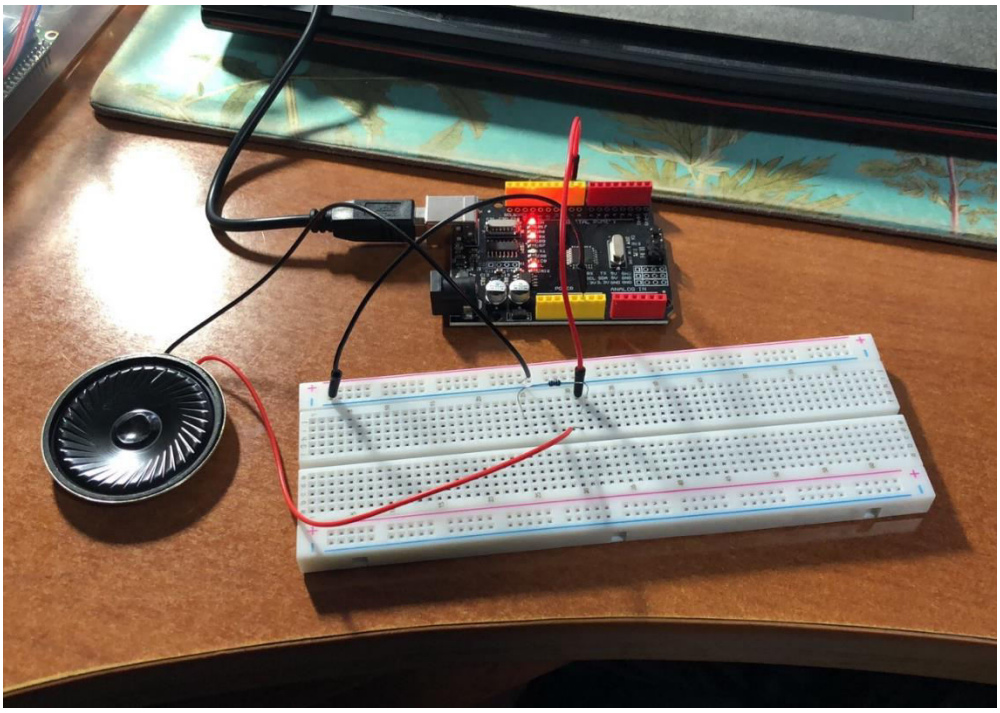


Figure 2 Speaker Testing Circuit

Speaker

```
const int buzzer = 9;

void setup(){
  pinMode(buzzer, OUTPUT); //set buzzer as an output
}

void loop(){
  tone(buzzer,1000);   //send 1KHz sound signal to the buzzer
  delay(1000);         //buzz for 1s
  noTone(buzzer);      //stop buzzing
  delay(1000);         //stop for 1s
}
```

Figure 3 Source Code for Testing the Speaker

Obstacle Avoidance Sensor

To test the IR sensor, I applied what we had done in out Module 2 and repeat the steps to this project. The circuit and testing object are as shown in Figure 4 in below. Code for this step was also from the source code of Module 2 (Figure 5). The serial monitor showed the distance of the object cube as it moved close and far, which meant the ultrasonic sensor was functioning properly.

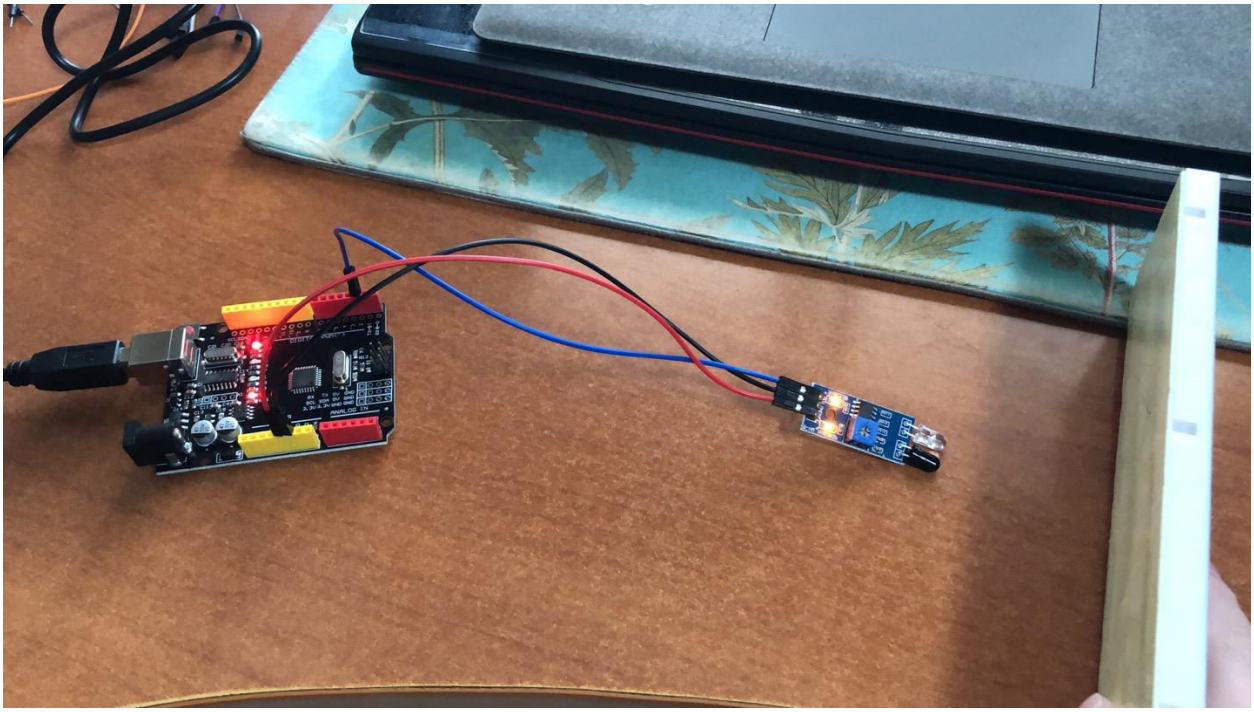


Figure 4 Ultrasonic Testing Circuit

Sensor

```
int LED = 13;
int SensorOutputPin = 2;
int Detect = HIGH;

void setup(){
  pinMode(LED, OUTPUT);
  pinMode(SensorOutputPin, INPUT);
  digitalWrite(LED, LOW);
  delay(1000);
}

void loop(){
  Detect = digitalRead(SensorOutputPin);
  if (Detect == LOW){
    digitalWrite(LED, HIGH);
  }
  else{
    digitalWrite(LED, LOW);
  }
  delay(10);
}
```

Figure 5 Source Code for Testing Obstacle Avoidance Sensor

SD Card & Adapter

Download the testing audio and convert it into applicable file type, which was 8-bit, 16000Hz, Mono audio channel. Then copy it using USB to the SD card. And next was to connect the SD card with the Arduino board. Here I used micro-SD card adapter to hold the wav. Showing in Figure 6 was the testing set up. As for coding, I needed to download "SD",

"TMRpcm", and "SPI" libraries in order to implement the code into my playing logic which was showed in Figure 7.

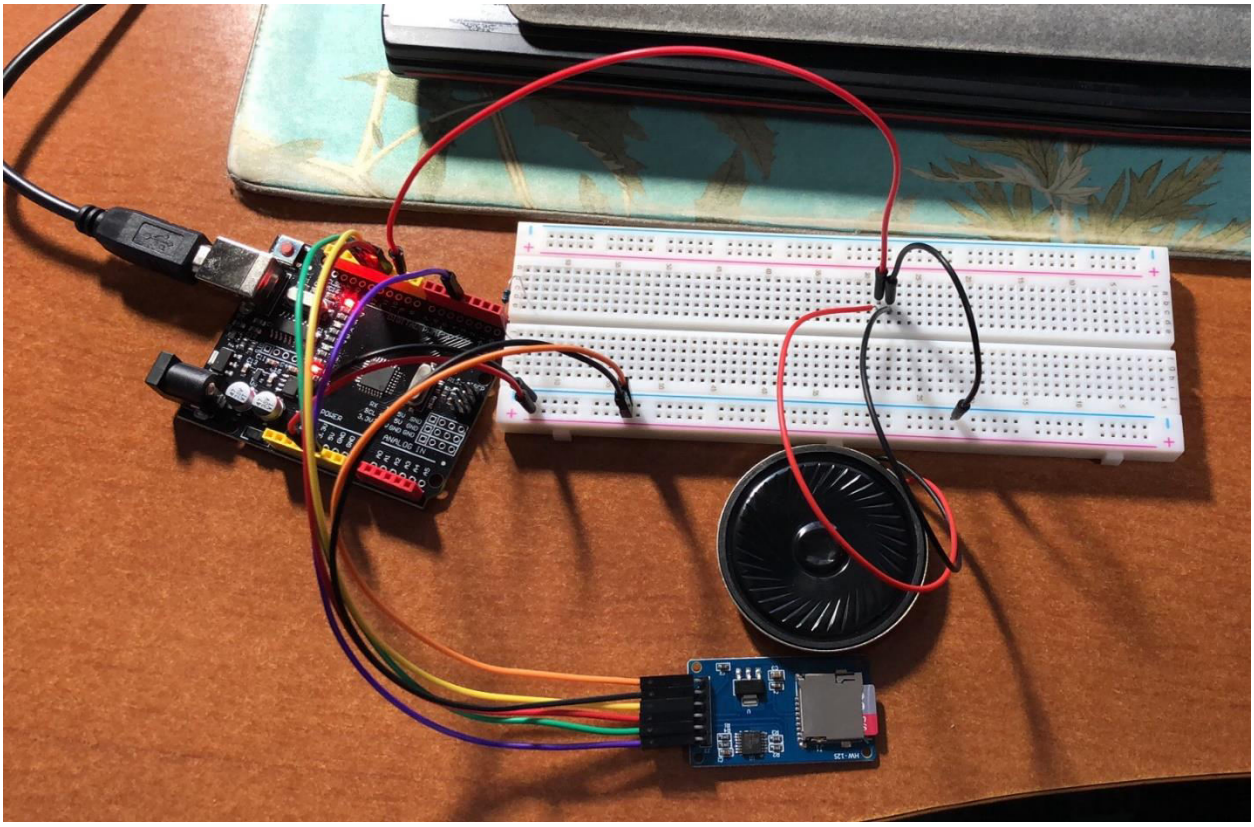


Figure 6 Micro SD Card Reading Circuit Setup

SD_Card

```
#include "SD.h"
#include SD_ChipSelectPin 4
#include "TMRpcm.h"
#include "SPI.h"

TMRpcm tmrpcm;

void setup(){
  tmrpcm.speakerPin = 9;
  Serial.begin(9600);
  if(!SD.begin(SD_ChipSelectPin)){
    Serial.println("SD fail");
    return;
  }

  tmrpcm.setVolume(6);
  tmrpcm.play("Test.wave");
}

void loop(){

}
```

Figure 7 Source Code for Testing Reading SD Card

Final Prototype

The final prototype was the combination of all tested components. The experimental layout was as showed in figure 8. The final product included an IR obstacle avoidance sensor, a micro SD card and its adapter, three push buttons, three LEDs, and a micro speaker. The push buttons were used for "pause", "pass", and "previous", respectively. The yellow, blue, and red were corresponding to "pause", "pass", and "previous", respectively. When pressing a button, its corresponding LED would light for one second; when there is no button being pushed down, the red LED would blink along with the beats of the playing song. All functions would work only after the IR sensor was activated. The hard part of building it was firstly the wires which might be overwhelming, and secondly the code – how would I combine separated code together without. There was another part didn't get tested on which was the light bulbs. The final code would be as shown in Figure 9, 10, 11.

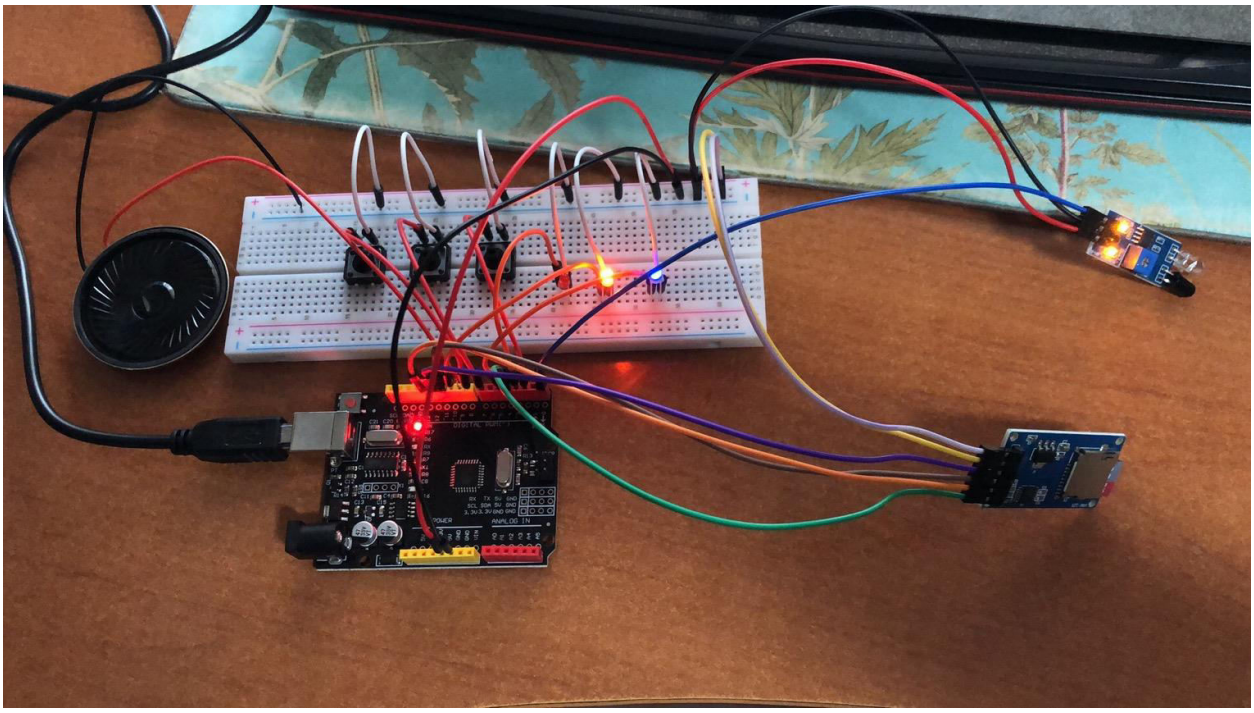


Figure 8 Final Prototype Circuit Overview

Final

```
#include "SD.h"
#define SD_ChipSelectPin 4
#include "TMRpcm.h"
#include "SPI.h"

TMRpcm tmrpcm;
int temp = 1;
int Pause = 5;
int Next = 6;
int Previous = 7;

int LEDB = 3;
int LEDY = 8;
int LEDR = 10;

int SensorOutputPin = 2;
int Detect = HIGH;

void setup(){
  pinMode(Pase, INPUT_PULLUP);
  pinMode(Next, INPUT_PULLUP);
  pinMode(Previous, INPUT_PULLUP);
  pinMode(LEDB, OUTPUT);
  pinMode(LEDY, OUTPUT);
  pinMode(LEDR, OUTPUT);

  tmrpcm.speakerPin = 9;
  Serial.begin(9600);
  if (!SD.begin(SD_ChipSelectPin)){
    Serial.println("SD fail");
    return;
  }

  tmrpcm.setVolume(6);
  tmrpcm.play("song1.wav");

  pinMode(SensorOutputPin, INPUT);

  delay(1000);
}

void song(void){
  if (temp == 1){
```

Figure 9 Final Prototype Code #1

Final

```
    tmrpcm.play("song1.wav");
  }
  else if (temp == 2){
    tmrpcm.play("song2.wav")
  }
  else if (temp == 3){
    tmrpcm.play("song3.wav")
  }
  else if (temp == 4){
    tmrpcm.play("song4.wav")
  }
}

void songPlay(void){
  while (digitalRead(Pause)==0 || digitalRead(Next)==0 || digitalRead(Previous)==0){
    if (digitalRead(Pase)==0){
      tmrpcm.pause();
      digitalWrite(LEDY,HIGH);
      delay(1000);
      digitalWrite(LEDY,LOW);
      while (digitalRead(Pause)==0);
      delay(200);
    }
    else if (digitalRead(Next)==0){
      if (tem<4)
      temp=temp+1;
      digitalWrite(LEDB,HIGH);
      delay(1000);
      digitalWrite(LEDB,LOW);
      while (digitalRead(Next)==0);
      delay(200);
      song();
    }
    else if (digitalRead(Previous)==0){
      if (temp>1)
      temp=temp-1;
      digitalWrite(LEDR,HIGH);
      delay(1000);
      digitalWrite(LEDR,LOW);
      while (digitalRead(Previous)==0);
      delay(200);
      song();
    }
```

Figure 10 Final Prototype Code #2

```
  }
}

void loop(){
  Detect = digitalRead(SensorOutputPin);
  if (Detect == LOW){
    songPlay();
  }
}
```

Figure 11 Final Prototype Code #3

Result & Discussion

From this project, I've learned some basic mechanisms of IR obstacle avoidance sensor, mini speaker, and other parts. I've also learned how to build a basic music player using Arduino, IR obstacle avoidance sensor, mini speaker, SD card adapter, and other parts, combining a simple circuit both in codes and setups. **All proposed objectives were accomplished.**