M5Stack Ultrasonic Music Box

Com Prog Lab 2190151 2563/2

Chawin Sungkhapong Chulalongkorn University

Demonstration video: https://youtu.be/yFCGvHrMZDY

Table of contents

Section 1 - Executive Summary

Background

In 2016, M5Stack was founded with a goal to help makers create high-quality prototypes for product development at an affordable cost. Today, M5Stack is one of the leading product development toolkits in the world. It offers many different applications such as to make IoT devices, automated systems, or customized gadgets for both product developers and tech hobbyists. In this project, one interesting application of an M5Stack is explored.

Objectives

1. Create a music box that plays different sounds depending on the distance between itself and an object in front.
2. Be able to apply the principles of programming in C, working with Arduino, and using M5Stack to solve the problem.
3. Understand the numerous potentials of microcontrollers and sensors.

Progress

This project uses an M5Stack core basic module with an esp32 development board inside. A small laptop is connected to the M5Stack with a USB type C cable, then, the M5Stack is wired to 2 potentiometers and one ultrasonic sensor by using jumper wires. After researching how the ultrasonic sensor works, a source code to control it along with the potentiometers is then written in the Arduino IDE and compiled onto the board. Finally, after testing and refining the code many times, each component is assembled into one compact device.

Outcomes

The finished product is a rectangular box with the transducers sticking out from the right side, the LCD display and the knobs located on top, and the power cable coming out from the left side. The LCD display features a logo of the product as well as the measuring distance in cm unit. The controls include a mute/unmute button (M5Stack Button A), a 3-step pitch control (top knob), and a tone pulsation frequency control (bottom knob). The 3 levels of the top knob are x1 (base octave), x2 (increases the pitch by 1 octave), and x4 (increases the pitch by 2 octaves). The device measures up to 400 cm where each note is separated by 5 cm.

Next Step

Despite having many errors corrected, the device still has much room for improvement. Each component be replaced with better models for more stability or range. A temperature and humidity sensor can also be included for better measurement accuracy. And finally, better speakers can be used instead of the M5Stack built-in speaker.

Section 2 - Introduction

Importance

Devices that deal with distance and sound are often related to safety. For example, the beeping sound when a car goes reverse, the alarm when a person is too close to a hazardous spot, etc. The core principles of the Ultrasonic Music Box are not different. For this reason, the Ultrasonic Music Box can be utilized as a safety device and measuring device, not just for entertainment. Additionally, this project lets the builder practice and understand the basics of working with a microcontroller and sensors. The knowledge acquired can be used to develop useful inventions in the future.

Applications

For social distancing purposes, the Ultrasonic Music box can be used to alert when another person is too close (requires modification). In addition, it could possibly function as a learning tool for young children (help relating short and long distances with higher and lower pitch).

In this report

This report includes 6 sections in order: Executive summary, Introduction, Background information, Project details, Summary, and Conclusions. At the end of the report are the references followed by the source code in the appendix. The executive summary is a short overview of the entire project. The background information section discusses the hardware specifications, sources used, and related projects with their shortcomings. The project details section talks about the approach used to build the Ultrasonic Music Box, the summarized equipment cost, and the results. The summary wraps up everything, and finally, the conclusions states what was accomplished and suggests what improvements could be done next time.

Section 3 – **Background Information**

Hardware specifications

1. M5Stack core basic development kit (https://docs.m5stack.com/en/core/basic)

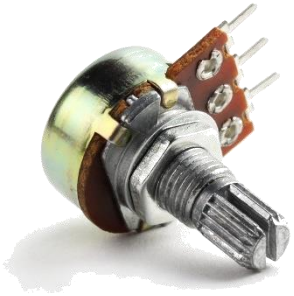
- ESP32-based
- Built-in Speaker, Buttons,Color LCD, Power/Reset button
- TF card slot (16G Maximum size)
- Magnetic suction at back
- Extendable Pins & Holes
- M-Bus Socket & Pins
- Program Platform: UIFlow, MicroPython, Arduino

Resources	Parameter
ESP32	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi, dual mode Bluetooth
Flash Memory	16MB
Power Input	5V @ 500mA
Port	TypeC x 1, GROVE(I2C+I/0+UART) x 1
Core Bottom Port	PIN (G1，G2，G3，G16, G17, G18, G19, G21, G22, G23, G25, G26, G35, G36)
IPS Screen	2 inch, 320x240 Colorful TFT LCD, ILI9342C, max brightness 853nit
Button	Custom button x 3
Speaker	1W-0928
Battery	110mAh @ 3.7V
Antenna	2.4G 3D Antenna
Operating Temperature	32°F to 104°F (0°C to 40°C)
Net weight	47.2g
Gross weight	93g
Product Size	54 x 54 x 18mm
Package Size	95 x 65 x 25mm
Case Material	Plastic (PC)

2. B10K potentiometer (https://www.amazon.co.uk/Adjustable-Potentiometer-Reverse-Linear-Rotary/dp/B0095S9H66)

10K Ohm Linear Taper Rotary Potentiometer Panel B10K Pot
- Resistance:10K Ohm
- Adjustment Type:Top Adjustment
- Maximum Power:0.5W
- Maximum Voltage:200V
- Type:Linear
- Type B ¡¡

Package Includes:
- 10 X 10K Ohms potentiometer



3. HC-SR04 Ultrasonic distance sensor (https://www.electroschematics.com/hc-sr04-datasheet/)

- Working Voltage: DC 5V
- Working Current: 15mA
- Working Frequency: 40Hz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angle: 15 degree
- Trigger Input Signal: 10µS TTL pulse
- Echo Output Signal Input TTL lever signal and the range in proportion
- Dimension 45 * 20 * 15mm

Important sources

The notable sources used in this project are

1. Esp32 data sheet (ESP32 Resources | Espressif Systems)
2. M5Stack official webpage (m5-docs (m5stack.com))

These two sources provide in-depth information that is needed to work correctly with each component.

3. Esp32 with ultrasonic sensor tutorial (ESP32: Pocket Size Distance Measuring and Logger : 4 Steps (with Pictures) - Instructables)

This webpage instructs how to initialize the connection between the esp32 and the HC-SR04 and provides a set of codes that triggers the sensor. At the beginning, the source code of this project is an adaptation of the codes from this source.

4. How HC-SR04 Ultrasonic Sensor works (How HC-SR04 Ultrasonic Sensor Works & How to Interface It With Arduino (lastminuteengineers.com))

This webpage explains the working principle of an HC-SR04. This information is needed to understand the sensor and construct the right code.

Related projects

This project is related to a project from the Bachelor of Creative University, AUT (Villiers). Their ultrasonic music box features 4 different sound controls: coarse pitch adjustment, fine pitch adjustment, volume, and frequency of the tone pulsation. It also has 1 mute switch, an LED indicator, and dual 0.5W speakers. The product is powered by batteries or via a USB cable.

Although their work is well-made, it still has some shortcomings which are the lack of on-screen display for more human-friendly interface and not utilizing the measurement values.

For this reason, 2 additional functions have been added to the M5Stack Ultrasonic Music Box in this project to improve on the prototype by AUT. One is an on-screen display that shows an “on” status, as well as mute/unmute buttons. Another is displaying the measured distance on the screen. This would allow the user to produce certain melodies by keeping track of distances.

Section 4 – **Project Details**

Approach and course of action

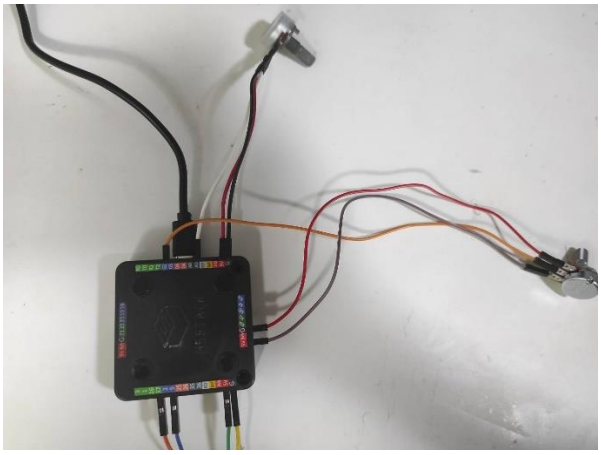
At the beginning, the approach is to see how the sensor should be connected to M5. As stated in the previous section, an online source explains how the HC-SR04 sensor works, and another tutorial provided useful information in terms of wiring. It also showed how to trigger the sensor and for what duration should the trigger pulse last (10 ms).

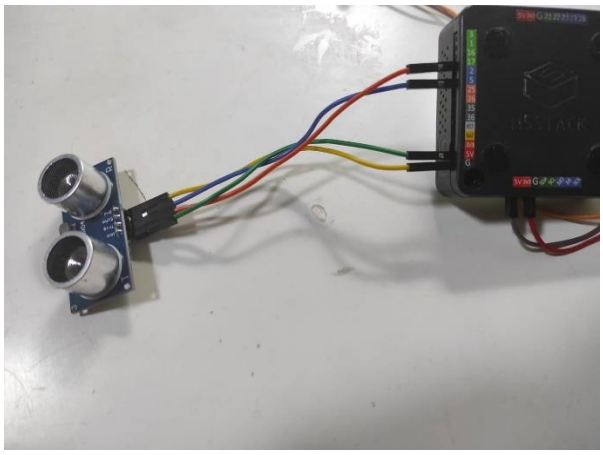
After firing the sensor, the sound wave return duration is obtained and used to calculate the distance. We get the distance to object by multiplying the duration with the speed of sound (in cm) and divided by 2 (to get one-way trip), since distance = speed * time.

Next is to create a display to indicate an “on” status as well as mute/unmute control on the M5 BtnA. A product logo is added on screen for more friendly appearance by using basic M5 LCD graphic text and shapes. The measuring distance is also displayed next to the mute button.

The “mute” button signifies that the speaker is not muted and pressing the button would mute it. If the speaker is currently muted, the button would change to “unmute” in red color.

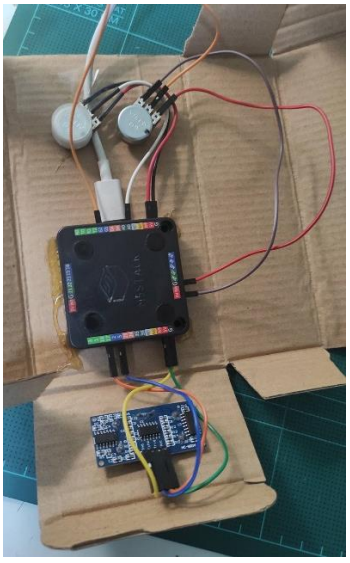
The next approach is to add the wiring and coding for using potentiometers. These are obtained from previous projects. The values sent from the potentiometers are obtained from the serial monitor. This allows testing in order to modify the code.





1st potentiometer:	2nd potentiometer:	HC-SR04
GND	GND	Trigger to pin 17
Pin 36	Pin 2	Echo to pin 5
5V	5V	VCC to 5V
		Gnd to GND

Each component is then fitted into one box for better appearance. They are attached using hot glue on non-electronic surfaces.



Finally, a 3-step pitch adjustment feature is added into the code and displayed on screen (x1, x2, and x4). The tone pulsation frequency is not displayed on screen since it does not have a defined level, but rather changes gradually.

Equipment and cost

Item	Unit Price (THB)	Quantity	Price (THB)
M5Stack core basic kit (w/ jumper wires)	1206.91	1	1206.91
HC-SR04 ultrasonic sensor	35	1	35
B10K potentiometer	10	2	20
Small cardboard box	21	1	21
USB-C cable	29	1	29
Total cost	THB 1311.91		

*The sources are specified in the References section.

Results

The music box functions as expected. When switched on, the ultrasonic sensor is triggered and sends out a signal that would reflect from the first object it touches. It then sends the measured duration to M5Stack, which performs calculations and plays a tone according to the potentiometer settings.

It is observed that increasing the distance by 5 cm does increment the note by one and turning the pitch knob by one step increases the sounds produced by 1 octave, with a maximum of 3 steps. The first note (at 5 cm distance) is C. The sensor measures up to 400 cm, and when the distance is longer, the onscreen display shows "> 400".

The distances are accurate down to about ±1cm. However, the music box seems to become more unstable at longer distances. This could result from the ultrasonic waves travelling in larger areas ($4\pi r^2$) and reflecting off from multiple objects. In addition, the potentiometers also give unstable signals at times, which are most likely due to the limitations in hardware quality.

Section 5 – Summary

This project was inspired by a project from the Bachelor of Creative University, AUT. The goal was to use an ultrasonic distance sensor along with a microcontroller to build a music box that plays different sounds when it measures different distances.

The music box from AUT has four sound controls functions (coarse pitch adjustment, fine pitch adjustment, volume, and frequency of the tone pulsation) and dual speakers. They used Arduino Pro Mini with HC-SR04. This project, using M5Stack with HC-SR04, has two sound control functions (pitch adjustment, frequency of tone pulsation) and one speaker. However, the music box in this project has an LCD display so functions such as displaying measured distance and displaying pitch increase level are added.

The completed music box functions well. It can be muted/unmuted by pressing the M5Stack button A. The overall pitch can be adjusted into a total of 3 octaves by turning a knob labeled “Pitch”. The frequency of tone pulsation can be adjusted gradually by turning a knob labeled “Pulse freq.” In addition, the M5Stack onscreen display shows the currently measured distance up to 400cm.

Section 6 – Conclusion

This lab has provided an opportunity to work microcontrollers. It has taught how to connect sensors to M5Stack, how to send and receive data from them, and how to code functions to control their actions. This project is an important achievement because an understanding in these mechanisms is needed to develop devices or systems in the future.

Another important take away from this project is that it has shown that sensors and microcontrollers can be used to create custom-made gadgets that could be very useful, or fun, in our daily life. Furthermore, they can be used to model actual products that sell. The possibilities are endless.

If this project is to be remake, changes would be to use better casing. The cardboard box is suitable for testing, but not durable enough to be a device that can be used frequently. Also, making the connections more neat, for example, using superglue instead of hot glue would make the music box look nicer.

Possible improvements for future projects include: 1)Using the M5 button B and C to control volume up/down. 2) Add a screen off or turn off feature by using long press on M5 button A. 3) Connect better speakers to produce more soothing sounds.

References

Important sources

https://vimeo.com/126472704

https://www.espressif.com/en/products/socs/esp32/resources

https://docs.m5stack.com/en/core/basic

https://www.instructables.com/Pocket-Size-Ultrasonic-Measuring-Tool-With-ESP32/

https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/

Equipment specifications

https://docs.m5stack.com/en/core/basic

https://components101.com/resistors/potentiometer

https://www.amazon.co.uk/Adjustable-Potentiometer-Reverse-Linear-Rotary/dp/B0095S9H66

https://www.electroschematics.com/hc-sr04-datasheet/

Equipment prices

https://sea.banggood.com/th/M5Stack-ESP32-Basic-Core-Development-Kit-Extensible-Micro-Control-WiFi-BLE-IoT-Prototype-Board-p-1236069.html?cur_warehouse=CN&rmmds=buy

https://shopee.co.th/%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%AA%E0%B9%88%E0%B8%87-HC-SR04-Ultrasonic-Sensor-Module-i.5401692.1751788055

https://shopee.co.th/ตัวต้านทานปรับค่าได้-WH148-B1K-B2K-B5K-B10K-B20K-B50K-B100K-B250K-B500K-i.12584257.2575437715

https://shopee.co.th/Kuulaa-สายเคเบิ้ล-Usb-Type-C-To-Usb-Type-60-W-Pd-Qc-4.0-สำหรับ-Samsung-Galaxy-S10-S9-Xiaomi-i.219770014.7315593553

Appendix - source code

```
#include <M5Stack.h>


const int trigPin = 17;

const int echoPin = 5;

const int pitchknob = 36;

const int periodknob = 2;


long duration;

int distance;

int pitch;

int period;

int pitchfactor;


boolean mute = false;


//--------------------------------------------------------------


void screen() {

  M5.Lcd.fillScreen(0xffff);


  //Logo

  M5.Lcd.setTextSize(3);

  M5.Lcd.fillRect(0, 70, 320, 80, 0xd7ff);

  M5.Lcd.setCursor(50, 85);

  M5.Lcd.setTextColor(0x0000);

  M5.Lcd.print("Ultrasonic");
```

```
  M5.Lcd.setCursor(50, 115);
  M5.Lcd.print("Music Box");
}

void mutebutton(){
  M5.Lcd.fillRect(0, 200, 120, 60, 0xffff);
  M5.Lcd.setCursor(45, 200);
  M5.Lcd.setTextColor(0x7bef);
  M5.Lcd.setTextSize(2);
  M5.Lcd.print("mute");
  M5.Lcd.fillTriangle(55,220,65,230,75,220,0x7bef);
}

void unmutebutton(){
  //UN-Mute
  M5.Lcd.fillRect(0, 200, 120, 60, 0xffff);
  M5.Lcd.setCursor(30, 200);
  M5.Lcd.setTextColor(0xf800);
  M5.Lcd.setTextSize(2);
  M5.Lcd.print("unmute");
  M5.Lcd.fillTriangle(55,220,65,230,75,220,0xf800);
}

void notelogo(){
  M5.Lcd.fillCircle(250,125,5, 0x0000);
  M5.Lcd.fillRect(253, 90, 3, 35, 0x0000);

  M5.Lcd.fillCircle(270,125,5, 0x0000);
```

```
  M5.Lcd.fillRect(273, 90, 3, 35, 0x0000);


  M5.Lcd.fillRect(253, 90, 20, 10, 0x0000);
}

void showdistance(){
  //Show on the screen
  M5.Lcd.fillRect(130, 200, 200, 20, 0xffff);
  M5.Lcd.setCursor(130, 200);
  M5.Lcd.setTextColor(0x03ef);
  M5.Lcd.setTextSize(2);
  M5.Lcd.print("Distance: ");
  if(distance > 400){
    M5.Lcd.print(">400");
  }else{
  M5.Lcd.print(distance);
  }
}

void pitchmult(){
  //Show on the screen above distance
  M5.Lcd.fillRect(130, 170, 200, 20, 0xffff);
  M5.Lcd.setCursor(130, 170);
  M5.Lcd.setTextColor(0x03ef);
  M5.Lcd.setTextSize(2);
  M5.Lcd.print("Pitch: x");
  M5.Lcd.print(pitchfactor);
}
```

```
//---------------------------------------------------------------------

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
  M5.begin();
  M5.Speaker.setVolume(1);
  M5.Speaker.update();
  screen();
  notelogo();
}

//---------------------------------------------------------------------

void loop() {

  // Initialize the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Fire trigPin for 10 ms
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the echoPin, returns the sound wave travel time (ms)
  duration = pulseIn(echoPin, HIGH);
```

```
// Calculate the distance
distance = duration*0.034/2;

// Read the input knob values
pitch = analogRead(pitchknob);
period = analogRead(periodknob);

// kepp track of reading values on Serial Monitor
Serial.println("\npitch = ");
Serial.println(pitch);
Serial.println("\nperiod = ");
Serial.println(period);
//Serial.print("Distance: ");
//Serial.println(distance);
//delay(500);

//adjust pitch factor
if (pitch <= 1000){
  pitchfactor = 1;
}else if (pitch >1000 && pitch <=2000){
  pitchfactor = 2;
}else{
  pitchfactor = 4;
}

if (!mute){
  M5.Speaker.tone((distance*6 + 230)*pitchfactor/2);
```

```
  //delay(50);
  //M5.Speaker.mute();

  //show Mute buttun status
  mutebutton();

 }else if (mute){
  M5.Speaker.mute();

  //show unmute button status
  unmutebutton();
 }

 if (M5.BtnA.wasPressed()) {
  mute = !mute;
 }

 //Display distance and pitch mulitplier
 showdistance();
 pitchmult();
 delay(100);
 M5.Speaker.mute();
 delay(period/50);
 M5.update();
}
```