

Girls Conference 2023

Workshop on rapid prototyping of a custom electronic musical instrument



THE UNIVERSITY OF
SYDNEY



Code for the Ultrasonic Musical Instrument:

The code to run the ultrasonic musical instrument is available at:

https://github.com/mitchbryson/ultrasonic_music

Program Code:

```
/*
ultra_music: Code for an Arduino-based, customisable electronic musical instrument.
The code runs on an Arduino Uno connected to an ultrasonic range sensor and a potentiometer.
Signals from these sensors are sent via USB, which can be used to control different musical tones.
*/

// Pins on the Arduino connected to the ultrasonic range sensor
#define TRIGGERPIN 10
#define ECHOPIN 9

// Minimum and maximum distance readings (in mm) for controlling sounds
#define MIN_DISTANCE 30.0
#define MAX_DISTANCE 500.0

// This function is called once at the start of the program
void setup() {
  pinMode(TRIGGERPIN, OUTPUT); // Pulses will be sent out of the Arduino
  pinMode(ECHOPIN, INPUT); // Echo pulses will be received into the Arduino
  Serial.begin(115200);
}

// This function is called repeatedly by the program
void loop() {

  // Read current potentiometer value
  int sensor_value = analogRead(A0);

  // Send an output pulse to the sonar sensor (1ms wide)
  digitalWrite(TRIGGERPIN, HIGH);
  delay(1); // wait 1ms
  digitalWrite(TRIGGERPIN, LOW);

  // Measure the time taken to receive the pulse back // Will be "0" or "-1" if no pulse is received
  int duration = pulseIn(ECHOPIN, HIGH);

  // Formula for the distance is half the duration, then divide by 2.910
  // to get distance in mm
  int distance = (duration/2) / 2.91;

  // Convert this into a control signal between 0 and 1023
  int distance_value;
  if ( (distance < MIN_DISTANCE) || (distance > MAX_DISTANCE) ) {
    distance_value = -1;
  }
  else {
    float df = (distance-MIN_DISTANCE)/(MAX_DISTANCE-MIN_DISTANCE);
    distance_value = df*1024; // "distance_value" is between 0 (MIN) and 1023 (MAX)
  }

  // Send message across USB
  Serial.print(distance_value);
  Serial.print(' ');
  Serial.print(sensor_value);
  Serial.println("");
  Serial.flush();

  // wait 50 milliseconds between messages sent
  delay(50);
}
```

Coding Instructions:

- **Setting up the Arduino IDE:**
 - Go to the web address above and click on the link to the code file called “ultrasonic_music.ino”
 - This contains the code for the device: we will compile this code and upload it to your Arduino using the Arduino software on your laptop/computer and the USB cable
 - Open up the software/program “Arduino” on your laptop/computer (sometimes called the “Arduino IDE”):
 - If you don’t have this installed already, you can download it for free from <https://www.arduino.cc/en/software>
 - If you are using version 1 of the Arduino IDE, you may need to setup the board that you are using:
 - Inside the Arduino IDE software, click on the menu “Tools” then “Boards” and select “Arduino Uno”: this will tell the software the type of Arduino we are programming
 - If you are using Version 2 or the Arduino software, this will usually be setup for you automatically.
- **Compiling and running the code:**
 - Go to the menu “File” then “New” to start a new Arduino sketch
 - Delete the contents of the new file then copy and paste the code into this file
 - Click on the “Verify/Compile” button (the big tick inside a small circle, located up the top left of the Arduino IDE window) to compile the code.
 - If there are any code errors, and error message in orange writing will appear at the bottom of the program window
 - Connect the Arduino Uno to your laptop/computer using the USB cable
 - The click on the “Upload” button (the right-facing arrow in a circle at the top left of the Arduino IDE) to upload your code to your Arduino and run your program
- **Connecting your device to start playing music:**
 - Once your code is uploaded, you can close the Arduino IDE application.
 - Go to the web address:
 - https://mitchbryson.github.io/arduino_musical_instrument/music.html
 - Make sure your Arduino circuit is plugged into the USB and follow the instructions to start playing!

Potential modifications and updates to the code

- Lines 12 and 13 of the code specify the minimum and maximum ranges for the ultrasonic sensors: you can modify these numbers to set the distance over which the instrument is controlled: change the numbers (make sure to put a decimal place!)
 - `#define MIN_DISTANCE 30.0`
 - `#define MAX_DISTANCE 500.0`