Mobile maker spaces for promoting maker education in schools

democratizing STEAM education and innovation development for all the learners

---

IO2

Guidelines for teachers

Theremin “The sound creation”

Difficulty level: low

Reference age: from 16 to 18 years

Cost: currently quite low

Lead partner: IIS Leonardo da Vinci

Authors: Antonio Caserta & Renato Gatti

Erasmus+

GRANT NUMBER: 2020-1-PL01-KA201-081698

Declaration

This file has been prepared in the context of the STEAM2GO project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Guidelines for teachers

Theremin “The sound creation”

Contents

Introductory notes

About the project

The students will implement the Theremin musical instrument using Arduino, two ultrasonic proximity sensor and one or two speakers. It will also be possible to create supports in cardboard or with a 3D printer if available, to keep the sensors distanced from the table surface.
Entire system will be programmed with Arduino IDE (Wiring).

Learning objectives

The students involved in the project will have the opportunities to:

- Consolidate concepts of physics on electromagnetic fields.
- Understand an electronic circuit in detail
- Create circuits using Arduino UNO and electronic components
- Learn the fundamentals of basic programming to implement a proposed model
- Analyze and identify the parameters and functions to be controlled in a complex system
- Interface, through the Arduino UNO, the I/O signals coming from the devices
- Develop, through the Arduino IDE (Wiring) software, a complex control system
- Test the prototype
- Optimize and define the optimal work solution by sharing among the participants

Connection to other subject areas and/or specific STEM concepts

- Mathematics
- Technology
- Electronics
- Information Technology
- Realization of artifacts related to a technical-scientific project
- Science: electrical circuit making

Hardware and Software that you may need

Hardware	Software	Crafting material
- 1 or 2 breadboard - Jumper wires kit - Wiring cables kit - 1 x Arduino UNO rev.3 - 2 x Ultrasonic Sensor HC-SR04 - 1 or 2 x Speakers - 1 x led RGB	- TinkerCAD - Arduino IDE - Access to Autodesk Tinkercad	- Some sheets of A4 size white paper or paperboard - 1 x Cutter - 1 x Scissor - 1 x Glue - 1 x Double-sided tape - 1 x Pencil - 1 x Pencil sharpener - Some fine permanent markers of various colours - Stationery equipment like pencils, pens, paper sheets

The working phases

1. The initial phase will be the identification of the correct functioning of a Theremin as musical instrument. Particular attention must be paid to study the physical principle that governs the instrument. This can be done through a quick Internet search guided by the teacher.
2. Make all the necessary interconnections between the ARDUINO UNO board and the various external components
3. Create and transfer the program from the PC to the Arduino board and replace the generic commands and I/O, used for programming the system on the PC
4. Test the correct functioning of the system

The philosophy of the project

The Theremin is an electronic musical instrument controlled with no physical contact with the Thereminist (performer). It was invented about 90 years ago.

Original instrument's controlling section work in high frequency and it usually consists of two metal antennas that detect the relative positions of the Thereminist's hands and, based on the distance detected, they control the oscillators: one antenna controls the frequency and the other controls the amplitude (volume) of the output audio signal.

The output electric signals from the theremin are amplified and sent to a loudspeaker.

Our project will be little different. We will use HC-SR04 ultrasonic distance sensor to measure hand position and will translate the distance detected by the sensors into a sound that will have an amplitude and a frequency proportional to the distance detected by the two sensors. Also, sound will turn off when hand it too far from the sensors.

Introducing the Arduino Uno board

What is Arduino?

Designed to allow even simple enthusiasts to program and build electronic circuits in a simple way and at a reduced cost, Arduino has become, in a short time, a real open-source ecosystem, acquiring the status of "success story made in Italy ".

An ecosystem, in fact. Defining it as a hardware is limiting, since it is an entire family of "cards", different from each other in a number of aspects: from size to processing capacity, from the type of connectivity available to the software / firmware equipment. It is an open source hardware board accompanied by software, with reduced dimensions (similar to that of a credit card), easy to use and with great potential, which allows the simplified creation of prototypes for interactive projects, especially in the field of Internet of Things, where the use of various types of sensors is envisaged. These are hardware that are able to interact with the physical world.

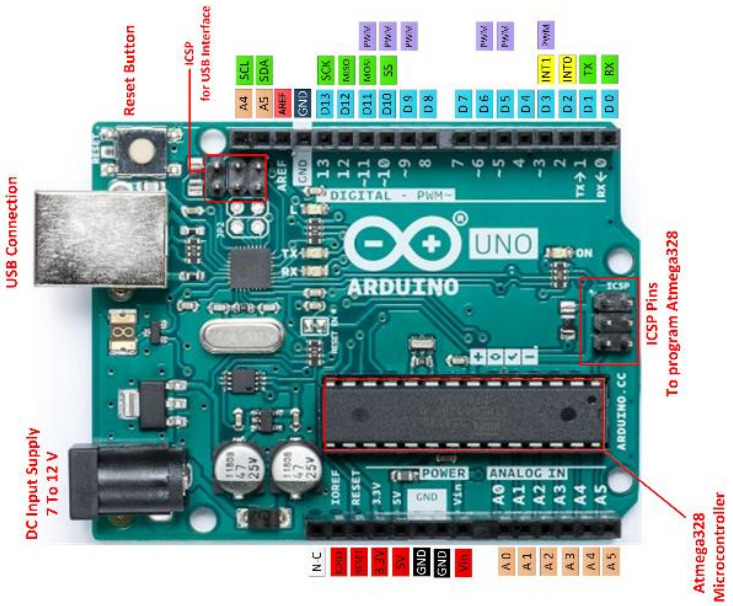
The main features of Arduino:

- it easily interfaces with sensors of various types, both analogue and digital
- it is easily programmable
- it is very compact
- it has a very low energy consumption

- it is very popular, so there are many discussion and planning groups

Layout of Arduino Uno Rev.3 has shown in ***Figure 1***

Figure 1 - Arduino Uno Rev 3

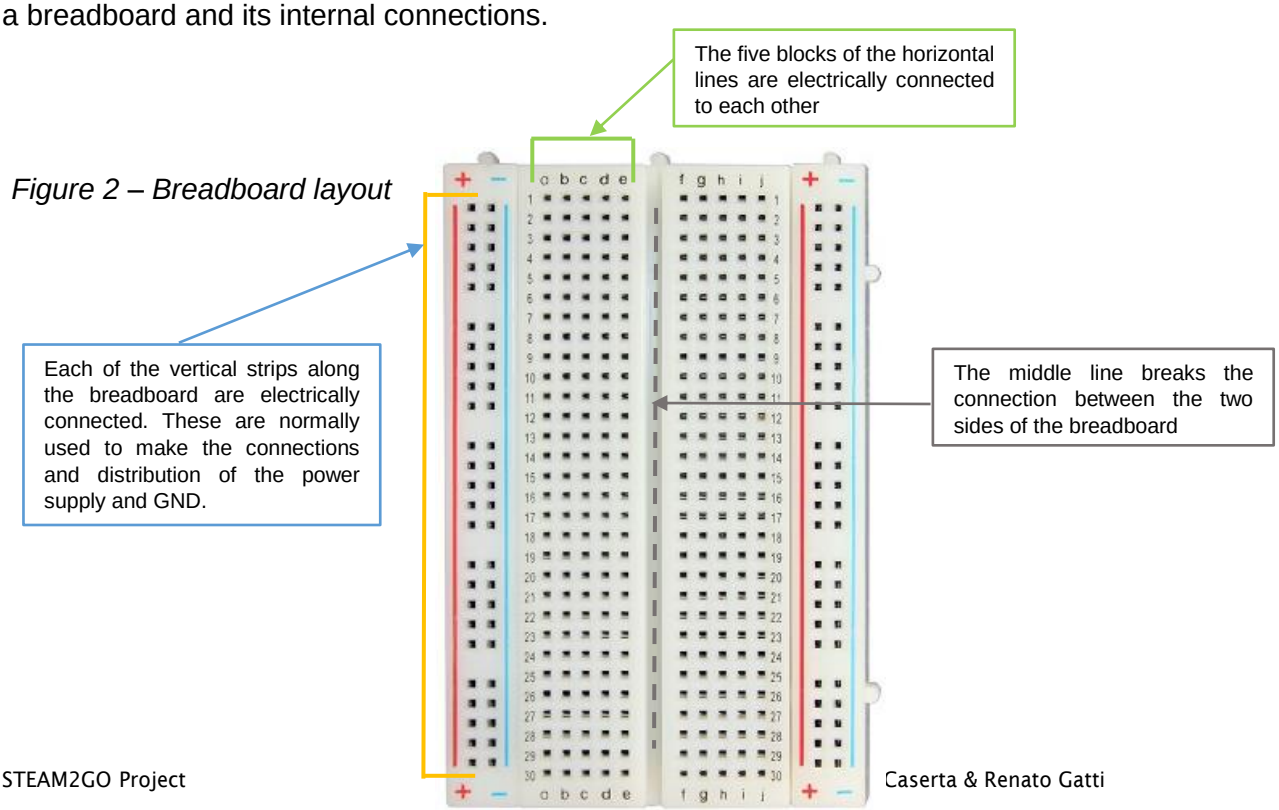


The breadboard

Breadboard is a way of constructing electronic devices without having to use a soldering iron. Components are pushed into the sockets (holes) on the breadboard and then extra 'jumper' wires are used to make the connections between the compomemts.

Breadboards are made for prototyping or quick experiments. They are not suitable for keeping circuits running for a long time. The connections, not being permanent, are easily removable in case you make a mistake or simply just start over and do a new project. In ***Figure 2*** the layout of a breadboard and its internal connections.

Figure 2 – Breadboard layout



In ***Figure 3*** an example of how to connect a LED with a battery and a button

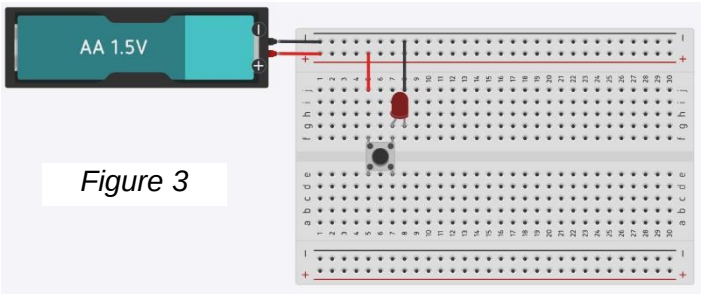


Figure 3

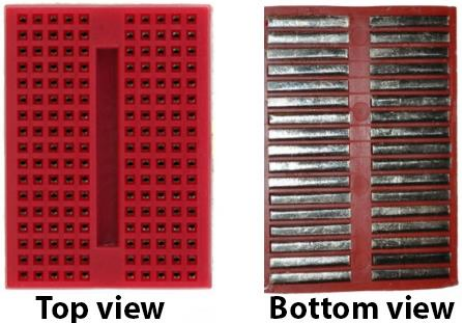


Figure 4

In ***Figure 4*** another type of breadboard, simpler and smaller, which does not have the vertical strips for the connections and distribution of the power supply and GND

Programming Arduino

First you need to download and install the IDE for Arduino (Integrated Development Environment). An IDE is software that allows you to write code directly, compile it and download it to the Arduino board to make it work. Often it also provides other features (such as debugger, help, keyword highlighting, etc.).

You can download the Arduino IDE 2.0.3 here:
https://www.arduino.cc/en/software

Connecting the Arduino Board to external devices

The first step is to connect the Arduino Board to the external devices, in our case two ultrasonic distance sensors, a loudspeaker and a LED RGB. To do this it is necessary to use the breadboard and the jumper wires, included in the Mobile Maker Space.

It is convenient to make a spreadsheet with the interconnections between the Arduino GPIOs and the various devices present in the respective breadboards. Example in ***Figure 5***

	Arduino	Ultrasonic Sensor 1
GPIO	PIN	Pin Name
+5V	5V via Breadbord	Vcc
D12	12	Trig
D11	11	Echo
GND	GND or 14	GND

	Arduino	Loudspeaker 1
GPIO	PIN	Cable/PIN Colour
D8	8	RED
GND	GND via Breadbord	BLACK

	Arduino	Ultrasonic Sensor 2
GPIO	PIN	Pin Name
+5V	5V	Vcc
D7	7	Trig
D6	6	Echo
GND	GND via Breadbord	GND

	Arduino	Loudspeaker 2 (optional)
GPIO	PIN	Cable/PIN Colour
D9	9	RED
GND	GND via Breadbord	BLACK

Figure 5 - Interconnections between the Arduino GPIOs and the external devices

By clicking **here** you can download the interconnections table in Excel format.

By clicking **here** you can download the interconnections table in JPG format.

The digital ports of the Arduino boards are shown in the following ***Figure 6***

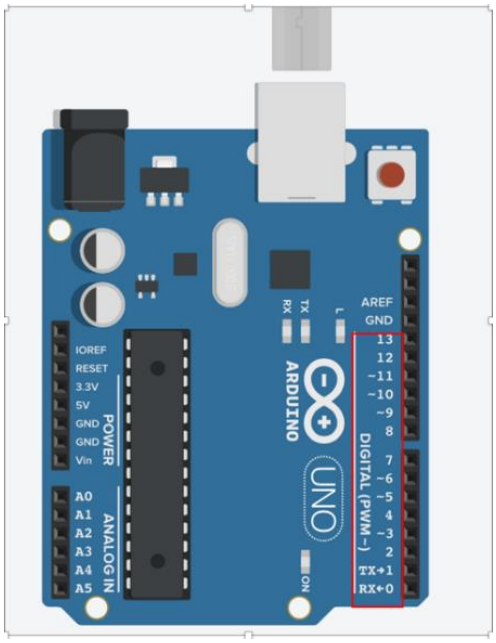


Figure 6 - Arduino board digital ports

We will use two HC-SR04 ultrasonic distance sensors **(*Figure 7)*** to measure the position of the hands and that will be converted into the frequency and amplitude of the sound coming out of the speaker.

Figure 7 - HC-SR04 ultrasonic distance sensors

Ultrasonic sensors do not directly measure the distance between the object and the sensor itself, but measure the time taken by the signal they emit to reach the object and return back to the sensor, i.e. they measure the echo, also called "flight time".

The HC-SR04 sensor has 4 pins: Vcc (+ 5V), Trigger, Echo, GND. To activate the sensor, a “high level” pulse signal is sent on the Trigger pin for at least 10 microseconds: at this point the sensor will send an ultrasonic wave, called “ping”, and wait for the return of the reflected waves; the sensor will respond on the Echo pin with a high pulse of the duration corresponding to the “travel” time of the sound waves. The duration of this return-back signal is used by the program to define the related output sound parameter.

Circuit and program to measure the distance with TinkerCAD

TinkerCad is an application developed by Autodesk that allows you to carry out numerous simulations with various electronic devices, including Arduino.
It is necessary to register and create an account on the following website:

https://www.tinkercad.com

The circuit is in ***Figure 8***. You can download the program to measure the distance using HC-SR04 in MS Word format by clicking **HERE** or in PDF format by clicking **HERE**.

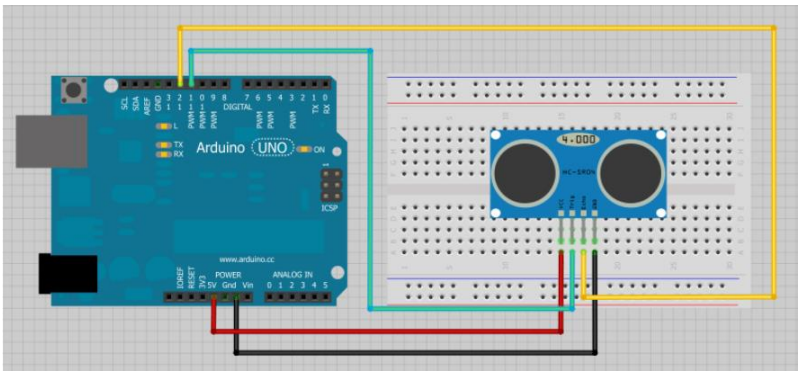


Figure 8 - Circuit to measure the distance with TinkerCAD

Theremin with two HC-SR04 ultrasonic distance sensors and RGB LED

To implement the "Theremin" system we connect the two ultrasonic sensors as in ***Figure 9.*** Then we connect the RGB LED as shown in the figure 9 respecting the pinout, with the shorter pin (cathode) connected to ground, finally we connect a loudspeaker to pin 8.

We will also use a multicolour LED (RGB) which will change colour, based on the frequency emitted by the Theremin speaker

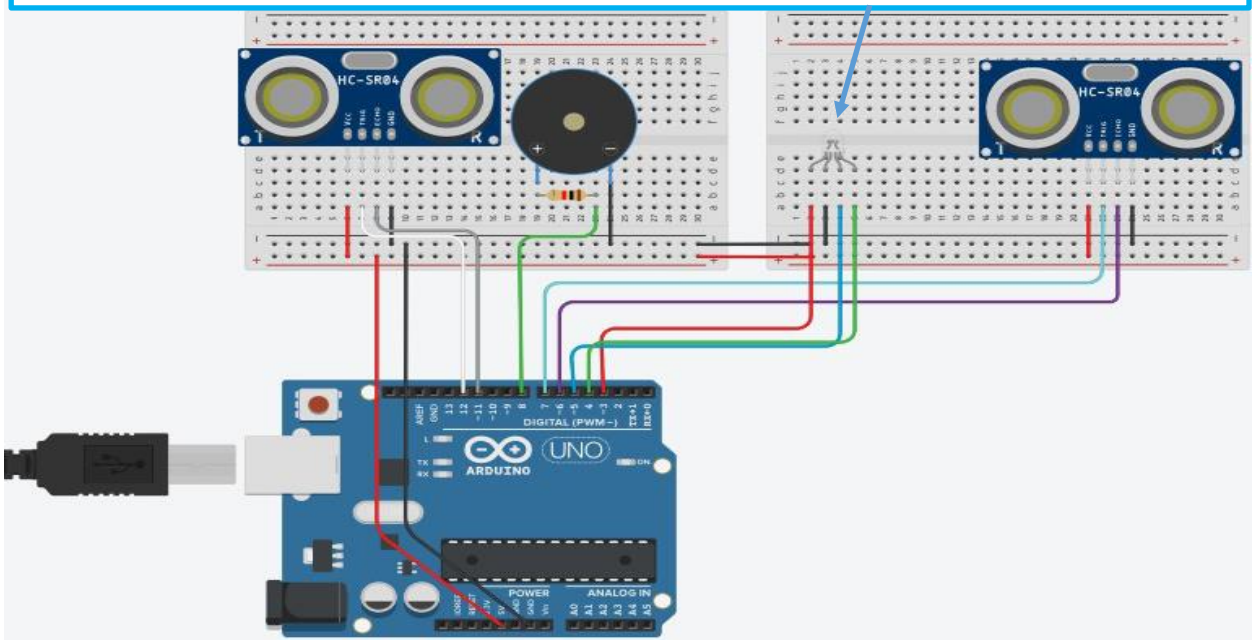


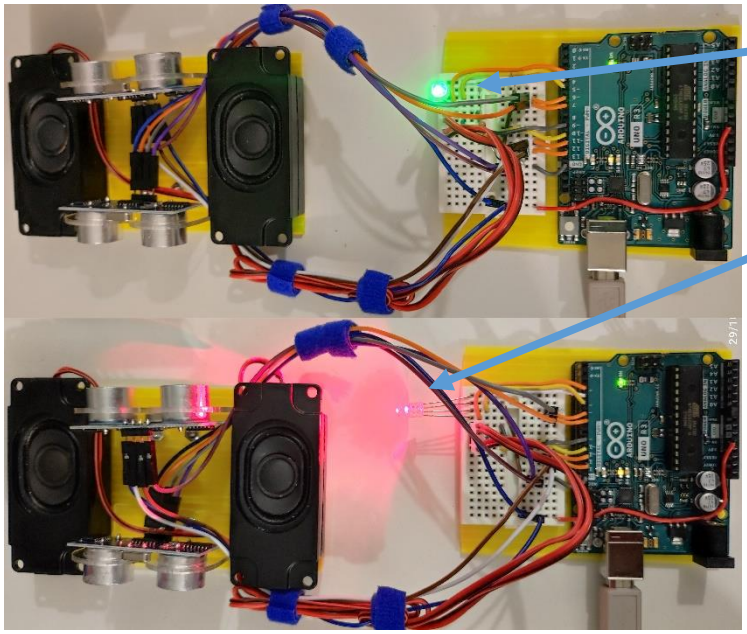
Figure 9 - Our Theremin with two HC-SR04 ultrasonic distance sensors and an RGB LED

The RGB LED, that we have included in the project, will change colour in a directly proportional way to the frequency of the sound produced. This will make it possible to the teacher to explain the correlation between the audio frequency spectrum and the electromagnetic frequency spectrum in general. For example, the visible spectrum could be introduced, which is between red (lowest visible frequency) and violet (highest visible frequency), observing the changes in the colour of the LED as the frequency of the emitted sound changes.

Figure 10 and ***Figure 11*** show the Theremin prototype on a single breadboard and speaker and on two breadboards and two speakers respectively.

We will also use a multicolour LED (RGB) which will change color, based on the frequency emitted by the Theremin

Figure 10 - Theremin assembled using only one breadboard and only one loudspeaker



Here you can see the RGB LED changing color based on the output audio frequency

Figure 11 - Theremin assembled using only two breadboards and two loudspeakers

Build the program using Arduino IDE on the PC

In this phase the code to implement the Theremin system is written using the Arduino IDE platform.

You can download the Theremin “Wiring Code program” by clicking **HERE**.

In the ***Figure 12*** is shown the complete program code of Theremin made by Arduino IDE.

Theremin.ino

```
#include <NewPing.h>
#include <toneAC.h>

#define DEBUG          false // Set to true to enable Serial debug
#define TONE_PIN       8
//#define TONE_VOLUME   10// 1-20
#define TRIGGER_PIN   12   // Board pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN      11   // Board pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE  200  // Maximum distance we want to ping for (in centimeters). Maximum sensor distance is rated at 400-500cm.
int portarossa=3;
int portaverde=4;
int portablu=5;
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
NewPing sonar1(7,6,200);

void colore (unsigned char rosso, unsigned char verde, unsigned char blu)
{
 analogWrite(portarossa, rosso); //activates the red LED with the intensity defined in the rosso variable
 analogWrite(portablu, blu); //activates the blue LED with the intensity defined in the blu variable
 analogWrite(portaverde, verde); //activates the green LED with the intensity defined in the verde variable
}

void setup() {
    Serial.begin(9600);
    //Serial.println("Theremin starting");
    pinMode(portarossa, OUTPUT); // declares port 3 as the output port
    pinMode(portaverde, OUTPUT); // declares port 4 as the output port
    pinMode(portablu, OUTPUT); // declares port 5 as the output port
}

 void loop() {
   delay(30); // Wait 30ms between pings (about 33 pings/sec). 29ms should be the shortest delay between pings.
   unsigned long uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
   unsigned long uS1= sonar1.ping();
   int distance=sonar1.ping_cm();
   int TONE_VOLUME=distance;
   //Serial.println(uS);
   //Serial.println(uS1);
   //Serial.println(distance);
   if (uS > 2000) { // Range is about 0-30 cm from sensor
     toneAC(0); // Turn sound off when not in range
    //if (DEBUG)
    //Serial.println("No tone");
   } else {
     int freq = 2000 - uS / 1.5; // Get sound frequency
     toneAC(freq, TONE_VOLUME); // Play it!
     //if (DEBUG)
     Serial.println(freq);
     if (freq>675 && freq<=875)
  {colore(255, 0, 0); // launches the color routine, with the red parameter at 255, the green at 0
  // and blue at 0 "lights up" red
  delay(10);}
  if (freq>876 && freq<=996)
  {colore(237,109,0); // "lights up" orange (237 of red and 109 of green)
  delay(10);}
  if (freq>997 && freq <=1117)
  {colore(255,215,0); // "lights up" yellow (255 of red and 215 of green)
  delay(10);}
  if (freq>1118 && freq<=1398)
  {colore(0,255, 0); // launches yhe color routine and "lights up" green
  delay(10);}
  if (freq>1399 && freq<=1639)
  {colore(0, 0, 255); // "lights up" blue
  delay(10);}
  //if (freq>1100 && freq<=1200)
  //{colore(0,46,90); // "lights up" indigo (46 of green and 90 of blue)
  //delay(10);}
  if (freq>1639 && freq<=2000)
  {colore(128,0,128); // "lights up" violet (128 of red and 128 of blue)
  delay(10);}
   }
 }
```

Figure 12 - The complete program code of Theremin made by Arduino IDE

Once the programming is complete we can move onto the test phase, using the prototype with one breadboard and one speaker (***Figure 13)*** or two breadboards and two speakers (***Figure 14***).

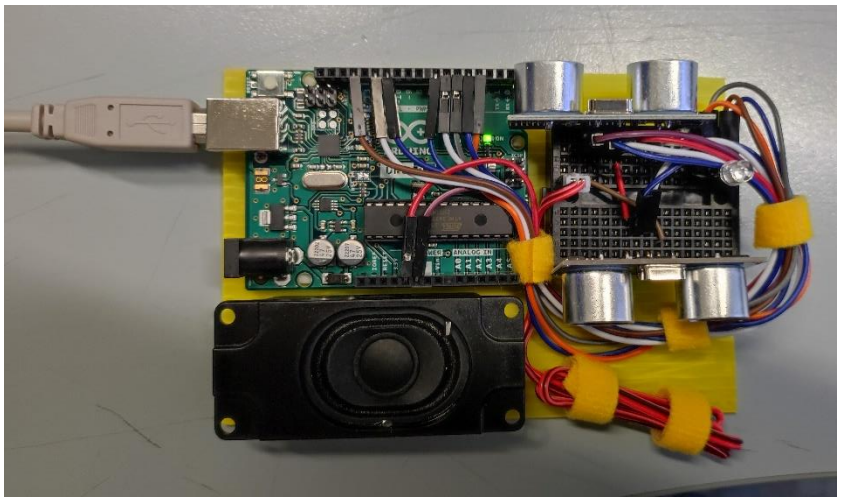


Figure 13 - The prototype made with only one breadboard

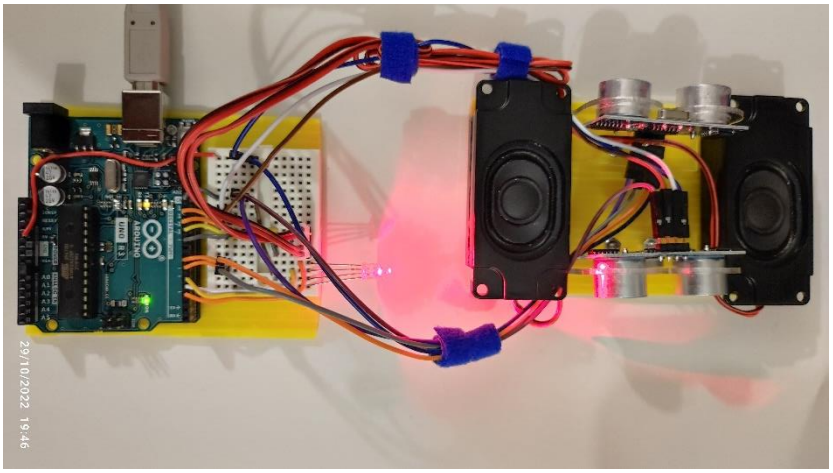


Figure 14 - The prototype made with two breadboards

You can find the video of the prototype made with a single breadboard and a single speaker by clicking **HERE**.

Here you will find the video of the prototype made with two breadboards and two speakers by clicking **HERE**.

IMPORTANT NOTE: Pay attention to the ultrasonic sensors that MUST NOT be placed directly on a surface but positioned at a height of at least 30/40 cm. from the table surface to achieve better functioning.

Supporting files

Worksheet for students

The worksheet for student

Programs

The folder with the programs

Figures

The folder with all the figures

Videos

The folder with all the videos

Arduino guides

Getting started with Arduino:

https://docs.arduino.cc/hardware/uno-rev3

https://docs.arduino.cc/learn/starting-guide/getting-started-arduino

Arduino Software (IDE):

https://www.arduino.cc/en/software

Tutorials Arduino:

https://docs.arduino.cc/tutorials

https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics

Other documentation:

https://store.arduino.cc/products/arduino-uno-rev3