PROGRAMMING RESOURCES - ACTIVITY SHEET 8

MAKE A THEREMIN

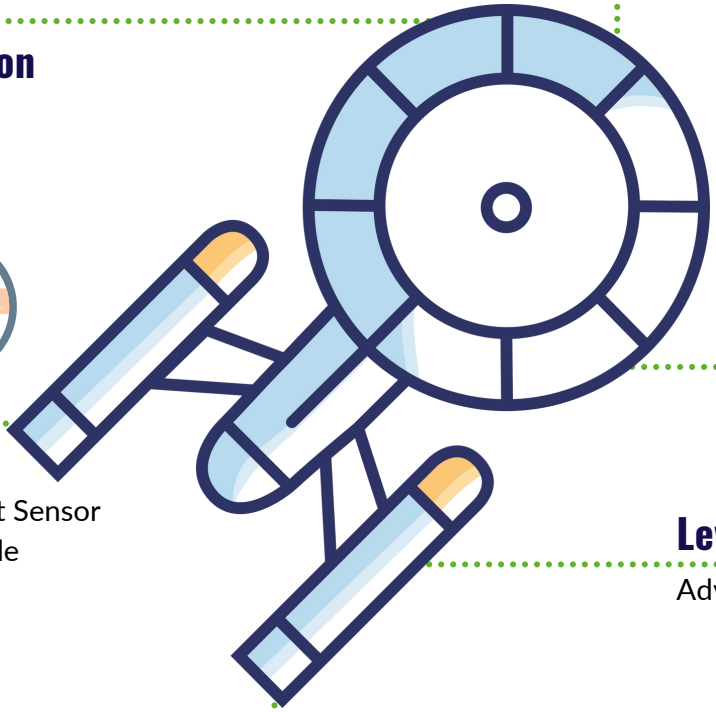
WITH THE DISTANCE SENSOR

#R1AS08

Available on

What is it?

A theremin is an electronic musical instrument, that can be played without touching it. The original concept is based on using two antennas to detect the hands' position. One antenna is used for the volume, and the other for the pitch of the tone.



Duration

30 minutes

Pre-requisites

- R1AS04 - Basic Light Sensor
- R1AS06 - Morse code

Level of difficulty

Advanced

Material

- 1 Programming board "***STM32 IoT Node Board***"
- Micro-B USB Cable
- 1 piezo buzzer or a speaker
- 1 Breadboard
- Jumper wires

LEARNING OBJECTIVES

- Use a distance sensor and understand how it works
- Make music with a really strange instrument
- Use the map function to transform a number from one range to another

This activity sheet is part of the Let's STEAM project funded with support from the European Commission through the Erasmus + Strategic Partnership programme. Its content 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.

Co-funded by the Erasmus+ Programme of the European Union

MAKE A THEREMIN WITH THE DISTANCE SENSOR

The theremin is an electronic musical instrument controlled without physical contact by the thereminist (performer). It is named after its inventor, Leon Theremin, who patented the device in 1928. The instrument's controlling section usually consists of two metal antennas that sense the relative position of the thereminist's hands and control oscillators for frequency with one hand and amplitude (volume) with the other. The electric signals from the theremin are amplified and sent to a loudspeaker.
Our version will be more simple, we will only control the pitch of the tone, with the distance sensor, the volume will be predetermined. **Let's make music!**
Resources: https://en.wikipedia.org/wiki/Theremin, https://youtu.be/x0NVb25p1oU

STEP 1 - MAKE IT

Wire buzzer/speaker

1

In theory, a buzzer or a speaker is not polarized (it means that there is no "+" nor "-"), but you often have a pair of wires black/red or signs ("+" and/or "-") on the device. If you are in this configuration, attach the lead on the "+" side of the buzzer to **D3** and the other one to **GND**.

If there is no colour or indication, just plug one wire on **D3** and the other one on **GND**.

Connect the board to the computer

2

With your USB Cable, connect the board to your computer by using the **micro-USB ST-LINK connector** (on the right corner of the board). If everything is going well you should see a new drive on your computer called **DIS_L4IOT**. This drive is used to program the board just by copying a binary file.

Open MakeCode

3

Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.
Resource: makecode.lets-steam.eu

Program your board

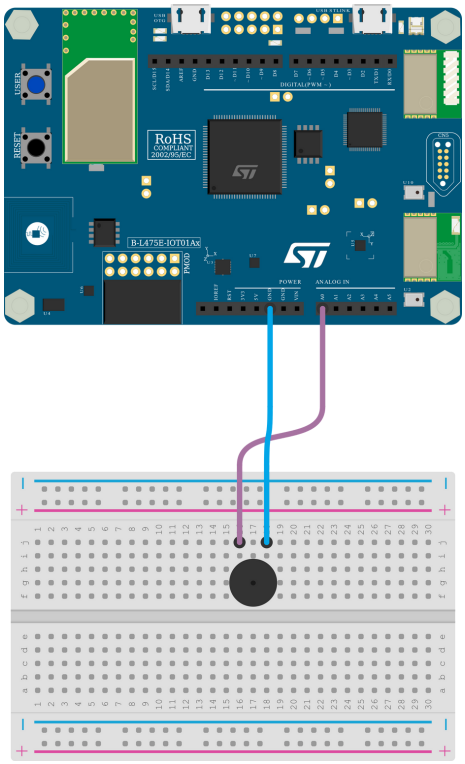
4

Inside the MakeCode Javascript Editor, copy/paste the code available in the **Code It Section** below. If not already done, think of giving a name to your project and click on the "**Download**" button. Copy the Binary file on the drive **DIS_L4IOT**, wait until the board finishes blinking and your program is ready!

Run, modify, play

5

Your program will automatically run each time you save it or reset your board (push the button labelled RESET). Try to understand the example and start modifying it.



Wiring the buzzer/speaker

MAKE A THEREMIN WITH THE DISTANCE SENSOR

STEP 2 - CODE IT

```
let distance = 0
forever(function () {
   // Get distance
   distance = input.distance(DistanceUnit.Millimeter)

   if (distance > 500) {
      // Convert the distance into frequency
      let note = Math.map(distance, 0, 500, 440, 830)
      music.ringTone(note)
   } else {
      music.stopAllSounds()
   }
})
```

Variables

In this program, there are 2 variables. The first, distance, is used to keep the same distance across the condition and for the tone to play. Then, you will find note, which is not technically necessary/mandatory but helps to introduce a greater understanding of each step of the program. It contains the transformation of the distance into tone frequency.

Getting distance

Using a variable to keep the distance is great, but knowing how to get the distance is better! Once again, there is no difficulty. We need to call the input.distance(DistanceUnit.Millimeter) function. The parameter DistanceUnit.Millimeter specifies to the function that we want the result in millimetres (1 meter = 1,000 millimetres).

Condition

The condition, if (distance > 500) { ... }, gives the information that we only play a sound if the measured distance is lower or equal to 500 millimetres.

Convert the distance into frequency

The most important part is the **conversion**. To make it, we use a mathematical function named map. This function remaps a value from a range to another. In this case, the value is remapped from **distance range** to **frequency range**. As you can see in the code above, this function takes five parameters, namely: map(value, in_min, in_max, out_min, out_max). Let's get a closer look at each of them:

- **value**: the value to re-map
- **in_min**: The minimum value of input range (distance)
- **in_max** : the maximum value of input range (distance)
- **out_min** : the minimum value of output range (frequency)
- **out_max**: the maximum value of output range (frequency)

So, we can understand, what this line does i.e., remapping the distance (with a range of 0 mm to 500 mm) to frequency (with a range from 440 Hz to 830 Hz).

The chosen frequencies are not random, the range of frequency from 440Hz to 830Hz represents an octave. It means you can find all the notes: A B C D E F G

Now we have a frequency. It's time to play it, simply using the music.ringTone(note).

MAKE A THEREMIN WITH THE DISTANCE SENSOR

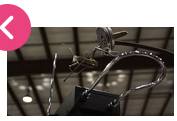
STEP 3 - IMPROVE IT

Change the map value to add octaves and/or distance to enhance your song.

Try to add a **potentiometer** to control the volume.

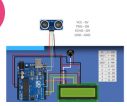
GOING FURTHER

Theremin - Learn more about the history, operating principles and uses of the theremin. *https://en.wikipedia.org/wiki/Theremin*



LED Ring Distance Sensor - Discover a fun project, which will end up in an alternative parking sensor. *https://www.instructables.com/LED-Ring-Distance-Sensor/*

Water Level Detector - Discover ultrasonic sensors converting electrical energy into acoustic waves. *https://www.instructables.com/Water-Level-Detector-2/*



Cat Feeder - Use an ultrasonic sensor to build an automatic cat feeder. *https://www.instructables.com/Cat-Feeder/*

Explore other activity sheets

R1AS05 - Potentiometer