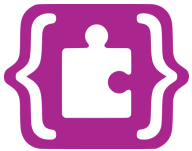


MUSIC

LET'S CREATE A MELODY

#R1AS07



Available on



Pre-requisites

- R1AS02 - Breadboarding: Make your first circuit!
- R1AS06 - Morse code

Material

- 1 Programming board "**STM32 IoT Node Board**"
- Micro-B USB Cable
- 1 set of LEDs
- 1 set of resistors
- 1 Breadboard
- Jumper wires

What is it?

Let's create a melody pleasant to our ears inspired by 8-bit consoles.

Duration

30 minutes

Level of difficulty

Advanced

LEARNING OBJECTIVES

- Play music with a programming board





While we are making a lot of noises using buzzers and speakers in diverse activity sheets such as making a theremin with the distance sensor or the buttons and LEDs buzzer quiz game, let's see what can be done for creating a more pleasant for the ears melody. We will learn how to play some notes and tones using a program to play a well-known melody. To stay in the electronic sound atmosphere, we will start with music inspired by 8-bit consoles.

Chiptune, also known as chip music or 8-bit music, is a style of synthesized electronic music made using the programmable sound generator (PSG) sound chips or synthesizers in vintage arcade machines, computers and video game consoles.

Resource: <https://en.wikipedia.org/wiki/Chiptune>



STEP 1 - MAKE IT



Wire buzzer/speaker

In theory, a speaker, or a buzzer, is not polarized (it means that there no "+" nor "-"), but often you've got a pair of wires **black/red** or **signs** ("+" and/or "-") on the device.

If you are in this configuration plus the **red** (or "+" wire side) on **D3**, and the **black** (or "-" wire side) on **GND**.

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

Connect the board to the computer

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

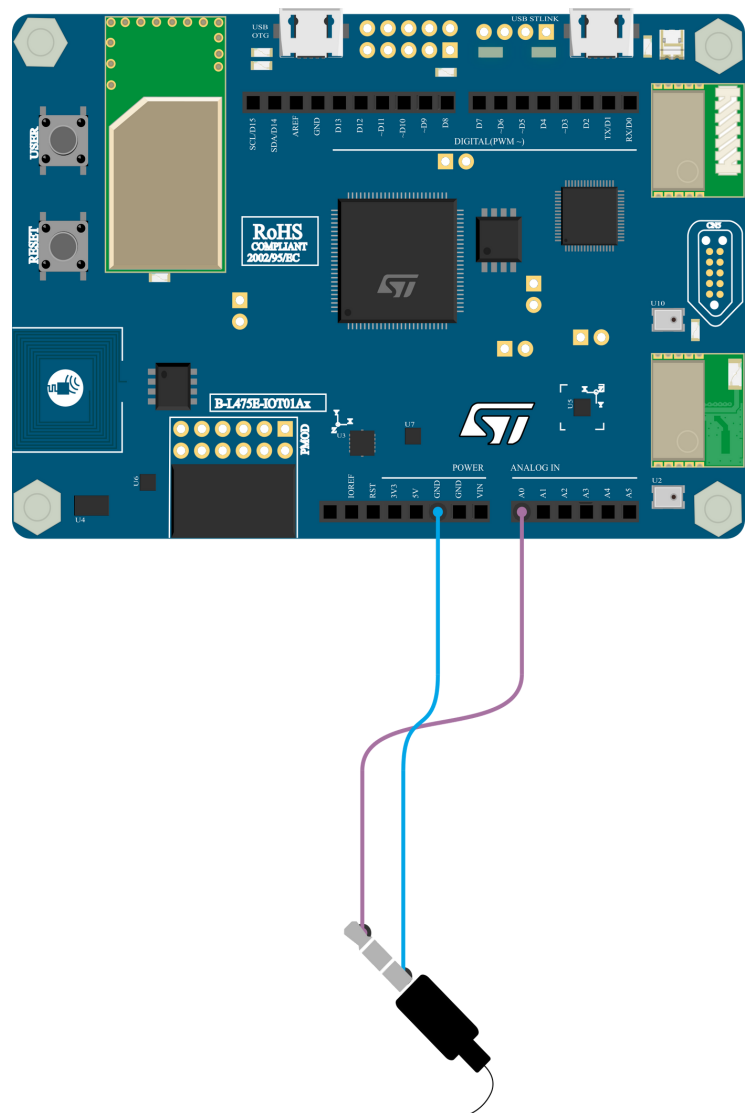
Go to the **Let's STEAM MakeCode editor**. On the home page, create a new project by clicking on the "New Project" button. Give a name to your project more expressive than "Untitled" and launch your editor.

Resource: makecode.lets-steam.eu

1

2

3



MakeCode simulator



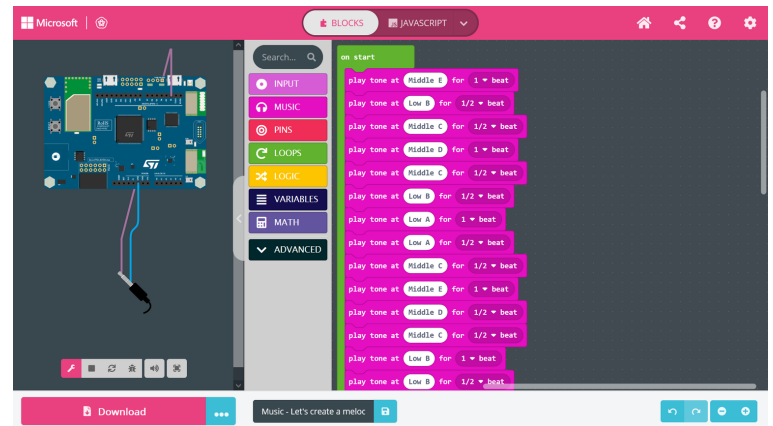
STEP 1 - MAKE IT



Install extension

After creating your new project, you will get the default "ready to go" screen shown here and will need to install an extension.

4



MakeCode editor with the Music Extension

Extensions in MakeCode are groups of code blocks that are not directly included in the basic code blocks found in MakeCode. Extensions, like the name implies, add blocks for specific functionalities. There are extensions for a wide array of very useful features, adding gamepad, keyboard, mouse, servo and robotics capabilities and much more.

See the black **ADVANCED** button at the bottom of the column of different block groups. Clicking **ADVANCED** will show additional block groups. At the bottom is a grey box named **EXTENSIONS**.

Click on that button. In the list of extensions available, you can easily find the **Music extension** that will be used for this activity. If not directly available on your screen, you can search it using the searching tool. Click on the extension you want to use and a new block group will appear on the main screen.

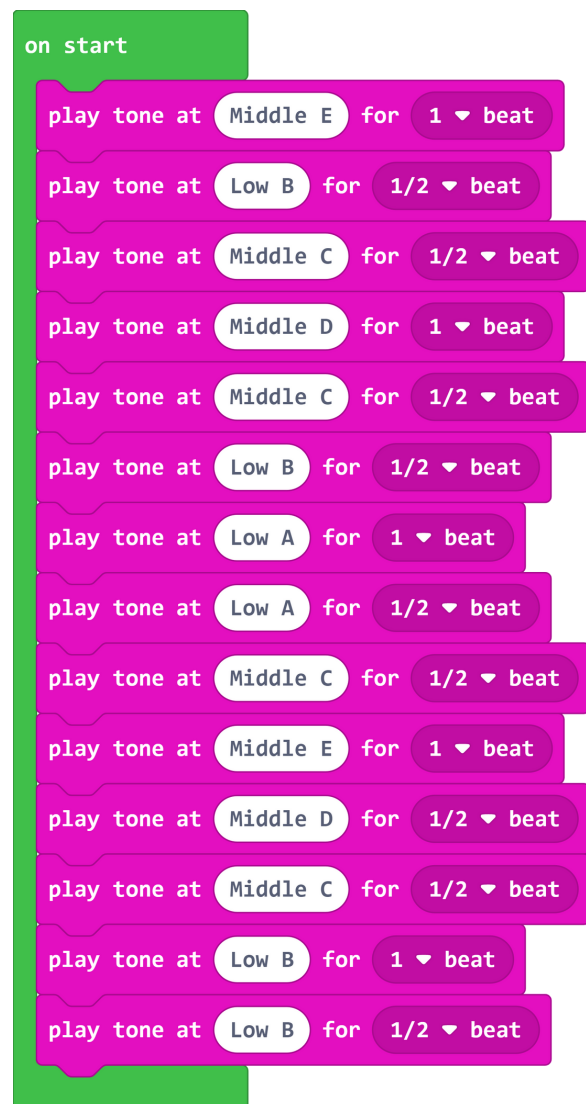
Program your board

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

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 by changing the period between two notes.

5



6

Full blocks enabling the program to run



STEP 2 - CODE IT



```
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(392, music.beat(BeatFraction.Half))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
```



STEP 2 - CODE IT



How does it work?

This program represents a sequence of notes with timing. The understanding of this activity is more related to music than programming.

The built-in music library in MakeCode allows us to play music on our board. To play a note we use the following command:

play tone at **Middle C** for **1/2 ▼ beat**

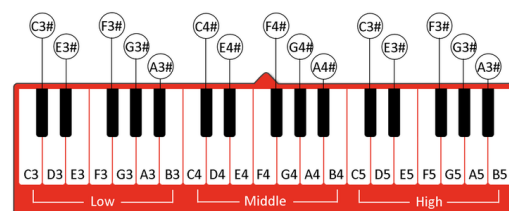
Where Middle C = note and 1 beat = duration.

Transcribe songs from sheet music

If we want to re-create our favourite songs, we first need a basic understanding of sheet music. Here's a reminder of the most common notes used in a musical score:



To choose the right note on MakeCode, you can click on the name of the note and make appear the virtual piano. Each key is a specific note:



Duration of the note

If we look again at the notes in a musical score, you will notice that they have different shapes and colours. These different shapes and colours indicate different durations called note values and expressed in number of beats.

Notes	Name	Value	Code
	Semibreve Whole note	4 beat	4 ▼ beat
	Minim Half note	2 beat	2 ▼ beat
	Crotchet Quarter note	1 beat	1 ▼ beat
	Quaver Eighth note	1/2 beat	1/2 ▼ beat
	Semiquaver Sixteenth note	1/4 beat	1/4 ▼ beat

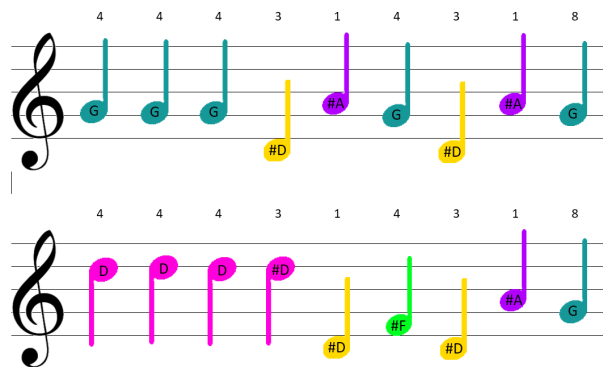


STEP 3 - IMPROVE IT

Write a program that **plays the following sound**:



Try to make the **Darth Vader's theme** with this partition:



By using the **distance sensor** as a presence detector, make a program that **plays the music of your choice each time it detect something**.

GOING FURTHER

233 music projects using Arduino.
<https://create.arduino.cc/projecthub/projects/tags/music>

How to make music with micro:bits - Using alligator clips, you can connect all sorts of things to your micro:bit, including a speaker.
<https://www.youtube.com/watch?v=bm7MGKspk0o>

Coding with micro:bit - Part 4 - Making Music - Look at sound and audio from the micro:bit and test out a variety of different buzzers and speakers.
https://www.youtube.com/watch?v=6hxxvLZSM_pM

Making music with micro:bit - Using the built-in music library in Make Code to play music on our micro:bit. <https://www.teachwithict.com/microbit-music.html>



Explore other activity sheets

R1AS12 - Motion Detection Alarm



R1AS08 - Make a theremin with the distance sensor

