

# SHALL WE DANCE



## STEMJAM Teaching Guide

Developing make spaces to promote creativity  
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

[www.stemjam.eu](http://www.stemjam.eu)



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

# SHALL WE DANCE

## ABSTRACT

The idea is to promote the learning of music through gamification and robotics.

The activity will consist of creating and coding a "choreography" for the music of the film "flash dance". It will be played by a mBot.

In first place, it will be necessary to understand some contents of music: types of notes, their duration and the meaning of some musical symbols.

Then, once the song has been chosen, every single note will have to be coded so that the mBot will play the song as if it was a musician.

On the other hand, taking into account the duration of the different notes that form the song, a dance will be devised, where each movement will be designed according to the notes. The dance will also be accompanied by lighting effects.

For the second version, "Lets Play Music", we need two arduino cards for musicplayer. The ultrasonic sensor connects to the card switches between music. We will decrease or increase the volume of the music with potentiometer. In addition, we will show on the screen the play / stop, the decrease / increase of the volume.

Important Note: To give basic electronic information to students; cables, pins, cards; Arduino card has been used in our project to ensure that circuits are built. It is targeted to use Raspberry Pi in later stages of the project.

- ❖ Phase 1: The process of working with mBot and mBlock.
- ❖ Phase 2: Arduino and mblock-S4a work process
- ❖ Phase 3: Raspberry Pi working process (Project students must be at least 8th grade)

## DIDACTIC OBJECTIVES

### GENERAL ISSUES:

- ❖ Know how to use the free sensor as motion sensor.
- ❖ Knowing to add music from outside to S4.
- ❖ Knowing how to control over-arduino card.

### TECHNOLOGY:

- ❖ Introduction to computational thinking.
- ❖ Assimilation, creation and programming of algorithms, to decompose complex problems into ordered sequences of simple instructions, which solve it.



## ENGINEERING and SCIENCE:

- ❖ Design of structures:
  - Stability: concepts of inertia and center of gravity.
  - Choice of the most suitable structure for the assembly of the chosen components and sensors.
  - Assembly of the structure.

## OTHERS:

- ❖ To acquire motivation for the study and understanding of the contents of other areas than STEM areas, such as MUSIC or DANCE.
- ❖ Learning to work and cooperate in a group.

STEM Subject:    Science                     Technology                     Engineering                     Mathematics

Education Level:                    12-14 years                     14-16 years

## PROBLEM STATEMENT

The mBot robot has to be programmed to work altogether with the laptop, by creating the operations' algorithms and the resultant effects. This way, when the students calculate the answer to one equation, there is a specific effect on the robot and so there is in the race.

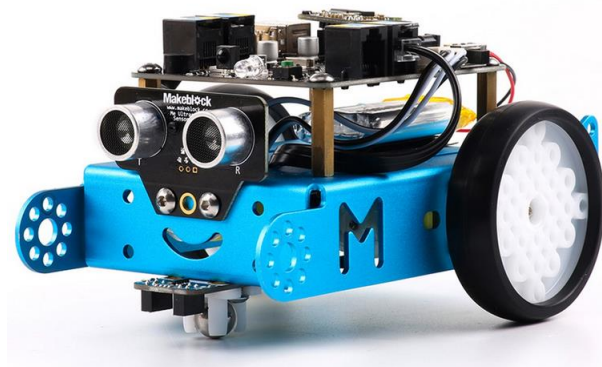
To build a robot to make a competition between two teams. Thinking about difficulties of maths operations. The basic one is for children at the beginning of primary school. Next you can make the problem more complex - add division with rational result. At the end – solving linear equation.

On reviewed version:

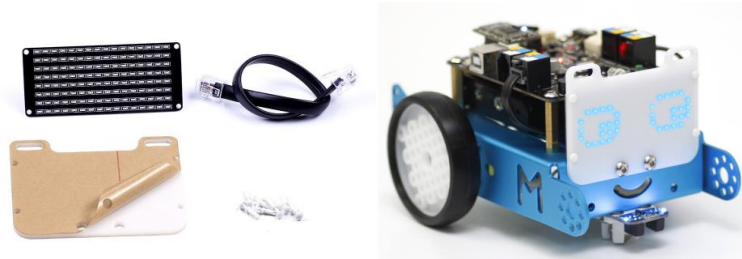
1. You need a remote to change music
2. Need to be close to music player to change music

## BOM (Bill of Materials Needed)

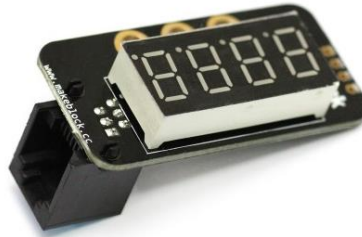
- (x2) mBot => Ref. 90054



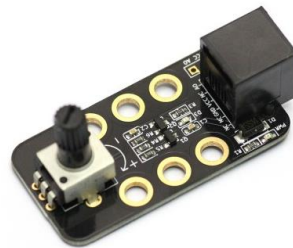
❖ Me LED Matrix 8 × 16:



❖ Me 7-Segment Serial Display - Red:



❖ Me Potentiometer:



❖ Me Ultrasonic Sensor:



❖ (x5) Cables RJ25.

❖ (x2) Arduino shields.

❖ Music and visual files to be loaded into the application.

❖ Rest of Attrezzo (not essential)

## First version

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl		
Mbot Robot 2'4G			1																
Motor 1	W*																W*		
Motor 2	W*																	W*	
Me 7-Segment serial display	B		1							B									
Me Led Matrix 8x16	B		1		B														
RJ25 cables			2																
Structures and beams			Several																
Laptops			1																
Attrezzo (not essential)			X																

## Second version

ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl		
Mbot Robot 2'4G			1																
Motor 1	W*																W*		
Motor 2	W*																	W*	
Me Led Matrix 8x16	B		3		B			B						B					
RJ25 cables			3																
Structures and beams			Several																
Laptops			1																
Attrezzo (not essential)			X																

## ACTIVITY DESCRIPTION

### First Version

This activity consists of coordinating the music a mBot plays, with the dance other mBot performs.

To achieve this goal, students will have to decide how to split the sheet music in small pieces. They will have to design, in a coordinated way, what both mBots (musician and dancer) do for every single cut, for every single beat.

Together with the music teacher, we studied the original sheet music and we decided to shorten it so that it was not too long. In the image below, you can see it shortened. This is the one we have coded.

### Flash Dance BSO

The musical score is written on a single treble clef staff in 4/4 time. It consists of four lines of music. The first line contains beats 1 through 8. The second line contains beats 11 through 27. The third line contains beats 28 through 29R, 30, 23R, 24R, 25R, 26R, 27R, 28R, and 29R. The fourth line contains beats 30R, 46, 47, and 48, ending with a double bar line and the word 'Fine'. Blue numbers are placed above each beat to indicate its position in the sequence.

The same numbers written above the beats, are the ones written at the right of the lines of the code.

Before starting to code its beat, it must be translated into the system of notes by letters, (the one used by MBlock):

Do	→	C
Re	→	D
Mi	→	E
Fa	→	F
Sol	→	G
La	→	A
Si	→	B

For all the programming, we will use MBlock software. We can download it at MBlock's page: <http://www.mblock.cc/download/>.



Once the software is installed, it can be paired with MBot by using the 2.4G Wireless Serial Port. But in this activity is rather better to upload the code to the mCore board (both codes, musician and dancer).

After all these technical tasks, we start with the PROGRAMMING.

### MUSICIAN CODE:

#### 1. INTRODUCTION PART OF THE PROGRAM:

As the program is uploaded in the mCore board, it will not start unless we press the on board button. As soon as we press it, the musicians will greet and show their instruments (by using the Led Matrix). After, the timer will reset.

```
mBot Program
wait until on board button pressed
set led on board all red 100 green 0 blue 0
show drawing Port1 x: 0 y: 0 draw: [Guitar]
show drawing Port2 x: 0 y: 0 draw: [Drum]
show drawing Port4 x: 0 y: 0 draw: [Trumpet]
set led on board all red 51 green 255 blue 255
wait 5 secs
show drawing Port1 x: 0 y: 0 draw: [Guitar]
show drawing Port2 x: 0 y: 0 draw: [Drum]
show drawing Port4 x: 0 y: 0 draw: [Trumpet]
wait 1 secs
reset timer
```

#### 2. CODING THE MUSIC (from 1-8 beats):

```
reset timer
play tone on note E5 beat Whole ▶ 1
play tone on note D5 beat Quarter
play tone on note C5 beat Quarter
play tone on note D5 beat Whole ▶ 2
play tone on note D5 beat Quarter
play tone on note E5 beat Quarter
play tone on note F5 beat Whole ▶ 3
play tone on note E5 beat Quarter
play tone on note D5 beat Quarter ▶ 4
play tone on note E5 beat Quarter
play tone on note D5 beat Quarter
play tone on note C5 beat Half
play tone on note D5 beat Quarter
play tone on note C5 beat Quarter
```

```
play tone on note A5 beat Whole ▶ 5
play tone on note G5 beat Quarter
play tone on note F5 beat Quarter
play tone on note G5 beat Whole ▶ 6
play tone on note F5 beat Half
play tone on note F5 beat Quarter ▶ 7
play tone on note E5 beat Quarter
play tone on note D5 beat Half
play tone on note C5 beat Half
play tone on note D5 beat Whole ▶ 8
play tone on note C5 beat Quarter
play tone on note D5 beat Quarter
```

### 3. CODING THE MUSIC (from 16-30 beats):

```

play tone on note E5 beat Whole ▶ 16
play tone on note D5 beat Quarter
play tone on note C5 beat Quarter
play tone on note D5 beat Whole ▶ 17
play tone on note D5 beat Quarter
play tone on note E5 beat Quarter
play tone on note F5 beat Whole ▶ 18
play tone on note E5 beat Quarter
play tone on note D5 beat Quarter
play tone on note E5 beat Quarter ▶ 19
play tone on note D5 beat Quarter
play tone on note C5 beat Half
play tone on note C5 beat Half
    
```

```

play tone on note A5 beat Whole ▶ 20
play tone on note C6 beat Quarter
play tone on note A5 beat Quarter
play tone on note G5 beat Whole ▶ 21
play tone on note F5 beat Half
play tone on note F5 beat Quarter ▶ 22
play tone on note E5 beat Quarter
play tone on note D5 beat Whole
wait 1 secs ▶ 23
play tone on note G5 beat Quarter
play tone on note B5 beat Quarter
    
```

```

play tone on note C6 beat Half ▶ 24
play tone on note C6 beat Half
wait 0.5 secs
wait 1 secs ▶ 25
play tone on note G5 beat Quarter
play tone on note B5 beat Quarter
play tone on note D6 beat Half ▶ 26
play tone on note C6 beat Half
wait 0.5 secs
play tone on note G5 beat Quarter ▶ 27
play tone on note B5 beat Quarter
play tone on note C6 beat Half
play tone on note C6 beat Half
    
```

```

play tone on note D6 beat Half ▶ 28
play tone on note C6 beat Quarter
play tone on note B5 beat Quarter
play tone on note C6 beat Half
play tone on note C6 beat Half ▶ 29
play tone on note D6 beat Half
play tone on note G5 beat Quarter
play tone on note G5 beat Quarter
play tone on note G5 beat Whole ▶ 30
wait 0.5 secs
    
```

### 4. CODING THE MUSIC (from 23R-30R beats):

```

wait 1 secs ▶ 23r
play tone on note G5 beat Quarter
play tone on note B5 beat Quarter
play tone on note C6 beat Half ▶ 24R
play tone on note C6 beat Half
wait 0.5 secs
wait 1 secs ▶ 25R
play tone on note G5 beat Quarter
play tone on note B5 beat Quarter
play tone on note D6 beat Half ▶ 26R
play tone on note C6 beat Half
wait 0.5 secs
    
```

```

play tone on note G5 beat Quarter ▶ 27R
play tone on note B5 beat Quarter
play tone on note C6 beat Half
play tone on note C6 beat Half
play tone on note D6 beat Half ▶ 28R
play tone on note C6 beat Quarter
play tone on note B5 beat Quarter
play tone on note C6 beat Half
play tone on note C6 beat Half ▶ 29R
play tone on note D6 beat Half
play tone on note G5 beat Quarter
play tone on note G5 beat Quarter
play tone on note G5 beat Whole ▶ 30R
wait 0.5 secs
    
```



5. CODING THE MUSIC (from 46-48 beats):

```
wait 1 secs ▶ 46
play tone on note G5 beat Quarter
play tone on note B5 beat Quarter
play tone on note C6 beat Half ▶ 47
play tone on note C6 beat Half
play tone on note C6 beat Quarter
play tone on note C6 beat Quarter
play tone on note C6 beat Double ▶ 48
GOODBYE
```

6. GOODBYE Routine:

Once the music is finished, the leds on board turn on red and the musicians say goodbye with a wink.

```
play tone on note C6 beat Double ▶ 48
GOODBYE
```

```
define GOODBYE
show drawing Port1 x: 0 y: 0 draw: [ ]
show drawing Port2 x: 0 y: 0 draw: [ ]
show drawing Port4 x: 0 y: 0 draw: [ ]
set led on board all red 255 green 0 blue 0
wait 2 secs
show drawing Port1 x: 0 y: 0 draw: [ ]
show drawing Port2 x: 0 y: 0 draw: [ ]
show drawing Port4 x: 0 y: 0 draw: [ ]
wait 2 secs
show drawing Port1 x: 0 y: 0 draw: [ ]
show drawing Port2 x: 0 y: 0 draw: [ ]
show drawing Port4 x: 0 y: 0 draw: [ ]
```

**DANCER CODE:**

1. INTRODUCTION PART OF THE PROGRAM:

As the program is uploaded in the mCore board, it will not start unless we press the on board button. As soon as we press it, the dancer will greet and the timer will reset. After that, the dancing will take place in three stages: slow dance – fast dance – final dance.



```

mBot Program
wait until on board button pressed
set led on board all red 0 green 0 blue 0
show drawing Port1 x: 0 y: 0 draw: 
set 7-segments display Port3 number 0
wait 5 secs
show drawing Port1 x: 0 y: 0 draw: 
wait 1 secs
reset timer
set Time to round timer * 10 / 10
SLOW DANCE
FAST DANCE
FINAL DANCE
GOODBYE

```

2. DANCING PART – SLOW DANCE:

```

define SLOW DANCE
repeat until Time = 23.8
set Time to round timer * 10 / 10
set 7-segments display Port3 number Time
if not Time > 2 then
show drawing Port1 x: 0 y: 0 draw: 
set led on board all red 102 green 255 blue 255
run forward at speed 150
else
if not Time > 4 then
set led on board all red 51 green 153 blue 255
run backward at speed 150
else
if not Time > 6 then
show drawing Port1 x: 0 y: 0 draw: 
set led on board all red 0 green 204 blue 204
run forward at speed 150
else
if not Time > 8 then
set led on board all red 51 green 51 blue 255
run backward at speed 150
else
if not Time > 8.5 then
show drawing Port1 x: 0 y: 0 draw: 
set led on board all red 0 green 76 blue 153
turn right at speed 150
else

```

```

else
  if not Time > 11 then
    set led on board all red 0 green 0 blue 102
    run forward at speed 150
  else
    if not Time > 13.5 then
      show drawing Port1 x: 0 y: 0 draw: 000
      set led on board all red 102 green 255 blue 255
      run backward at speed 150
    else
      if not Time > 14 then
        set led on board all red 51 green 153 blue 255
        turn right at speed -150
      else
        if not Time > 16 then
          show drawing Port1 x: 0 y: 0 draw: 000
          set led on board all red 0 green 204 blue 204
          run forward at speed 150
        else
          if not Time > 18 then
            set led on board all red 51 green 51 blue 255
            run backward at speed 150
          else
            if not Time > 18.5 then
              show drawing Port1 x: 0 y: 0 draw: 000
              set led on board all red 0 green 76 blue 153
              turn left at speed 150
            else

```

```

else
  if not Time > 21 then
    set led on board all red 0 green 0 blue 102
    run forward at speed 150
  else
    if not Time > 23 then
      show drawing Port1 x: 0 y: 0 draw: 000
      set led on board all red 102 green 255 blue 255
      run backward at speed 150
    else
      if not Time > 23.5 then
        show drawing Port1 x: 0 y: 0 draw: 
        set led on board all red 51 green 153 blue 255
        turn left at speed -180
      else
        run forward at speed 0

```

### 3. DANCING PART – FAST DANCE:

```
define FAST DANCE
repeat until Time = 48.6
set Time to round timer * 10 / 10
set 7-segment display Port3 number Time
if not Time > 24.6 then
set led on board all red 255 green 0 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 255
else
if not Time > 25.6 then
set led on board all red 0 green 0 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run backward at speed 255
else
if not Time > 26.6 then
set led on board all red 0 green 255 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 0
else
if not Time > 27.6 then
set led on board all red 0 green 255 blue 0
show drawing Port1 x: 0 y: 0 draw: 000
run backward at speed 255
else
```

```
if not Time > 28.6 then
set led on board all red 255 green 255 blue 0
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 255
else
if not Time > 29.6 then
set led on board all red 255 green 60 blue 0
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 0
else
if not Time > 30.6 then
set led on board all red 255 green 0 blue 0
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 200
else
if not Time > 31.6 then
set led on board all red 255 green 0 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run backward at speed 200
else
if not Time > 32.6 then
set led on board all red 0 green 0 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 0
else
if not Time > 33.6 then
set led on board all red 0 green 255 blue 255
show drawing Port1 x: 0 y: 0 draw: 000
run backward at speed 200
else
```

```

else
  if not Time > 34.8 then
    set led on board all red 0 green 255 blue 0
    show drawing Port1 x: 0 y: 0 draw: 100
    run forward at speed 200
  else
    if not Time > 36.8 then
      set led on board all red 255 green 255 blue 0
      show drawing Port1 x: 0 y: 0 draw: 200
      run forward at speed 0
    else
      if not Time > 37.8 then
        set led on board all red 255 green 60 blue 0
        show drawing Port1 x: 0 y: 0 draw: 300
        run forward at speed 255
      else
        if not Time > 38.8 then
          set led on board all red 255 green 0 blue 0
          show drawing Port1 x: 0 y: 0 draw: 400
          run backward at speed 255
        else
          if not Time > 39.7 then
            set led on board all red 255 green 0 blue 255
            show drawing Port1 x: 0 y: 0 draw: 500
            run forward at speed 0
          else
            if not Time > 40.7 then
              set led on board all red 0 green 0 blue 255
              show drawing Port1 x: 0 y: 0 draw: 600
              run backward at speed 255
            else

```

```

else
  if not Time > 41.7 then
    set led on board all red 0 green 255 blue 255
    show drawing Port1 x: 0 y: 0 draw: 700
    run forward at speed 255
  else
    if not Time > 42.6 then
      set led on board all red 0 green 255 blue 0
      show drawing Port1 x: 0 y: 0 draw: 800
      run forward at speed 0
    else
      if not Time > 43.6 then
        set led on board all red 255 green 255 blue 0
        show drawing Port1 x: 0 y: 0 draw: 900
        run forward at speed 200
      else
        if not Time > 44.4 then
          set led on board all red 255 green 60 blue 0
          show drawing Port1 x: 0 y: 0 draw: 1000
          run backward at speed 200
        else

```

```
else
  if not Time > 45.0 then
    set led on board all red 255 green 0 blue 0
    show drawing Part x: 0 y: 0 draw: 00
    run forward at speed 0
  else
    if not Time > 46.0 then
      run backward at speed 200
    else
      if not Time > 47.0 then
        run forward at speed 200
      else
        run forward at speed 0
```

#### 4. DANCING PART – FINAL DANCE:

```
define FINAL DANCE
repeat until Time = 54.1
set Time to round timer * 10 / 10
set 7-segmentsdisplay Port3 number Time
set led on board led left red 255 green 0 blue 0
set led on board led right red 0 green 0 blue 0
set led on board led left red 0 green 0 blue 0
set led on board led right red 255 green 255 blue 0
set led on board led right red 0 green 0 blue 0
set led on board led left red 255 green 0 blue 0
set led on board led left red 0 green 0 blue 0
set led on board led right red 255 green 255 blue 0
set led on board led left red 255 green 255 blue 0
set led on board led right red 255 green 0 blue 0
if not Time > 49.1 then
show drawing Port1 x: 0 y: 0 draw: 000
turn right at speed 150
else
if not Time > 50.1 then
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 0
else
if not Time > 51.1 then
show drawing Port1 x: 0 y: 0 draw: 000
turn right at speed 200
else
```

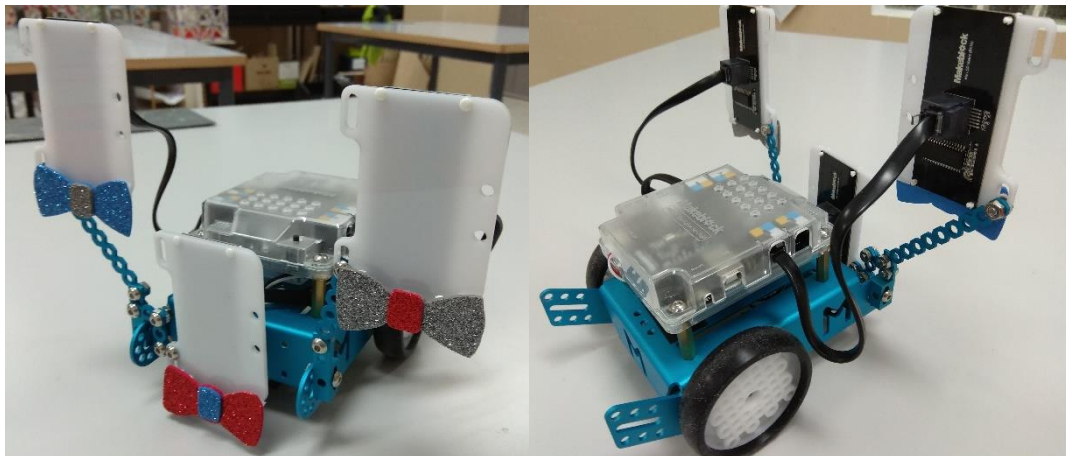
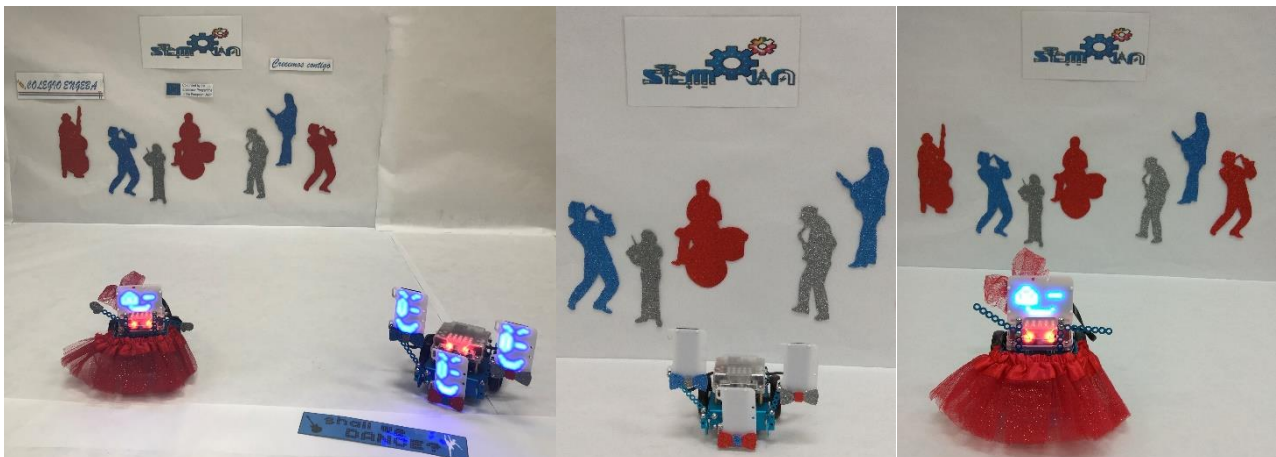
```
else
if not Time > 51.6 then
show drawing Port1 x: 0 y: 0 draw: 000
run forward at speed 0
else
if not Time > 53.6 then
show drawing Port1 x: 0 y: 0 draw: 000
turn right at speed 255
else
run forward at speed 0
```

## 5. GOODBYE Routine:

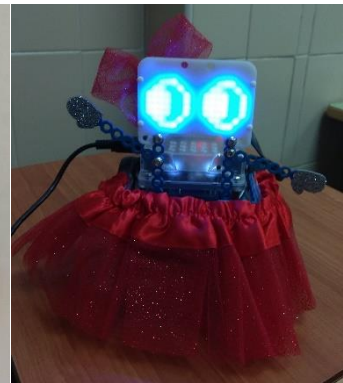
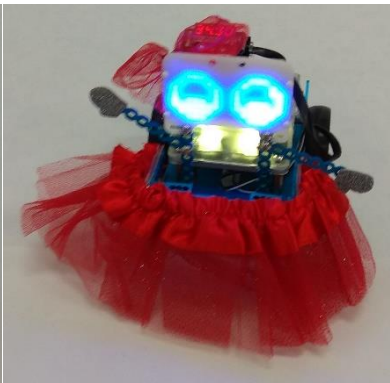
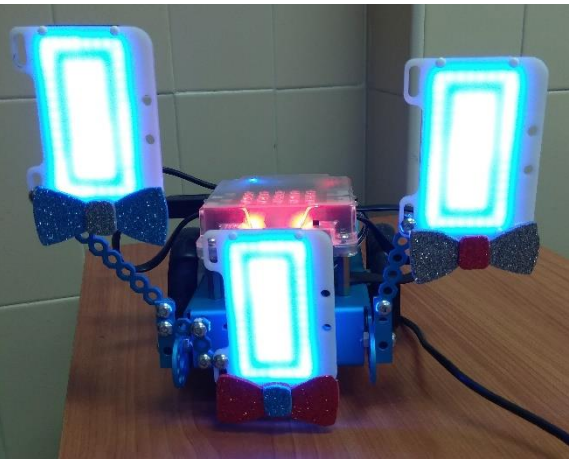
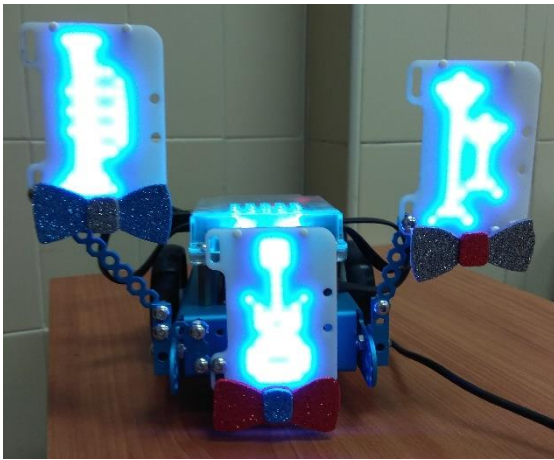
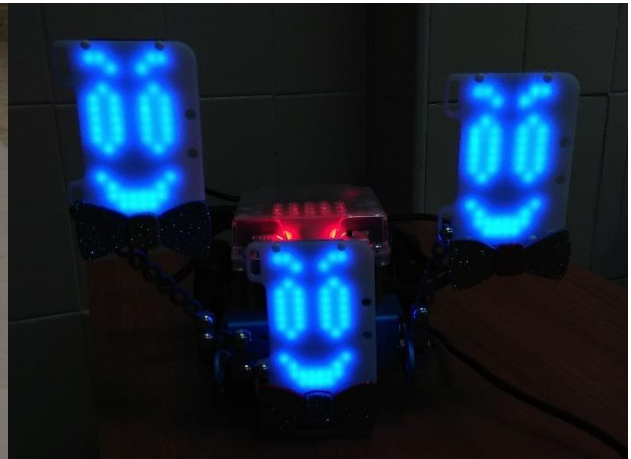
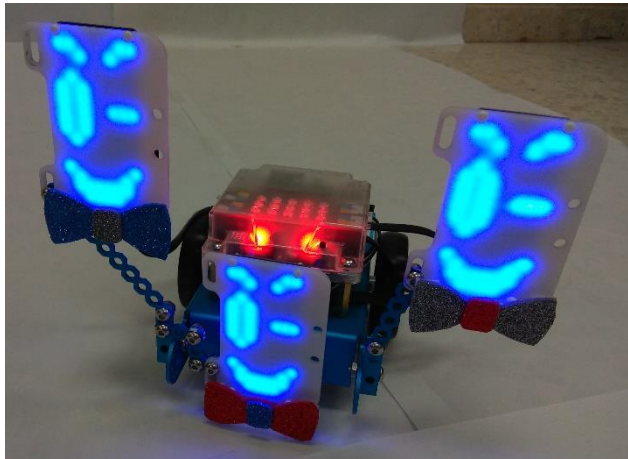
Once the music is finished, the mBot stops dancing and say goodbye with a wink.

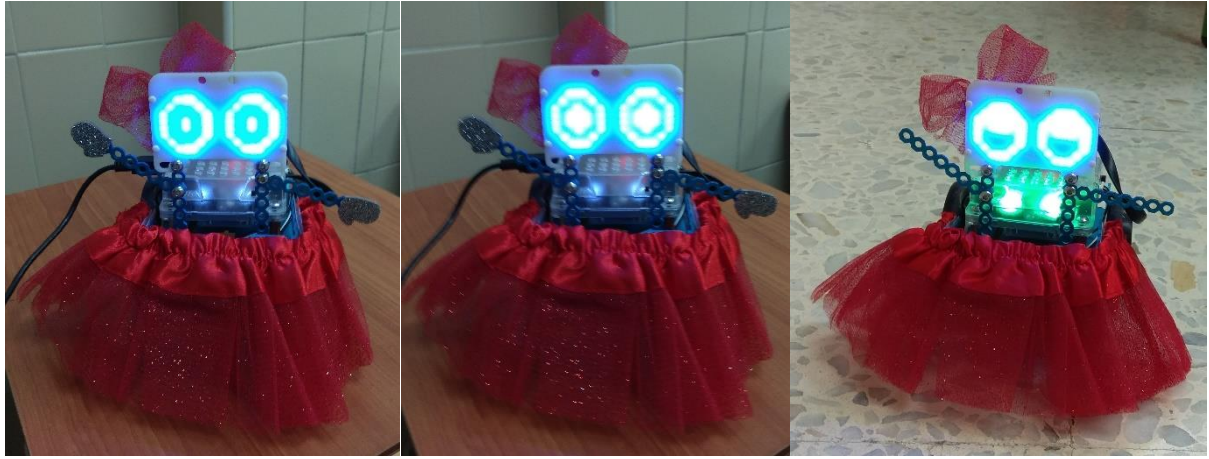
```
define GOODBYE
  set 7-segmentdisplay Port3 number Time
  show drawing Port1 x: 0 y: 0 draw: [drawing]
  set led on board all red 0 green 0 blue 0
  wait 2 secs
  set led on board all red 255 green 0 blue 0
  show drawing Port1 x: 0 y: 0 draw: [drawing]
  wait 2 secs
  show drawing Port1 x: 0 y: 0 draw: [drawing]
  wait 0.15 secs
  show drawing Port1 x: 0 y: 0 draw: [drawing]
```

Once, the programming is finished, we start building up THE STRUCTURE, where all the mechanical elements will be set. Also the electronic elements.









## Second version

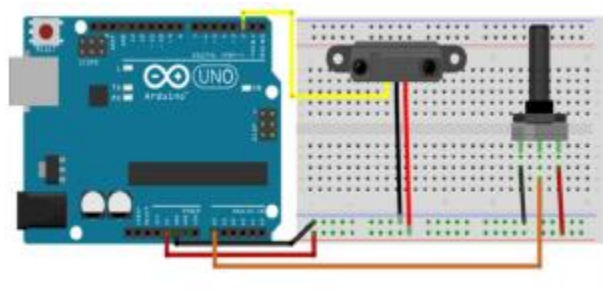
Step 1: Circuit diagram created by Fritzing. (You can also use ThinkCorD to ask)

The potentiometer and ultrasonic sensor are supplied from the common cathode lead to the power generated.

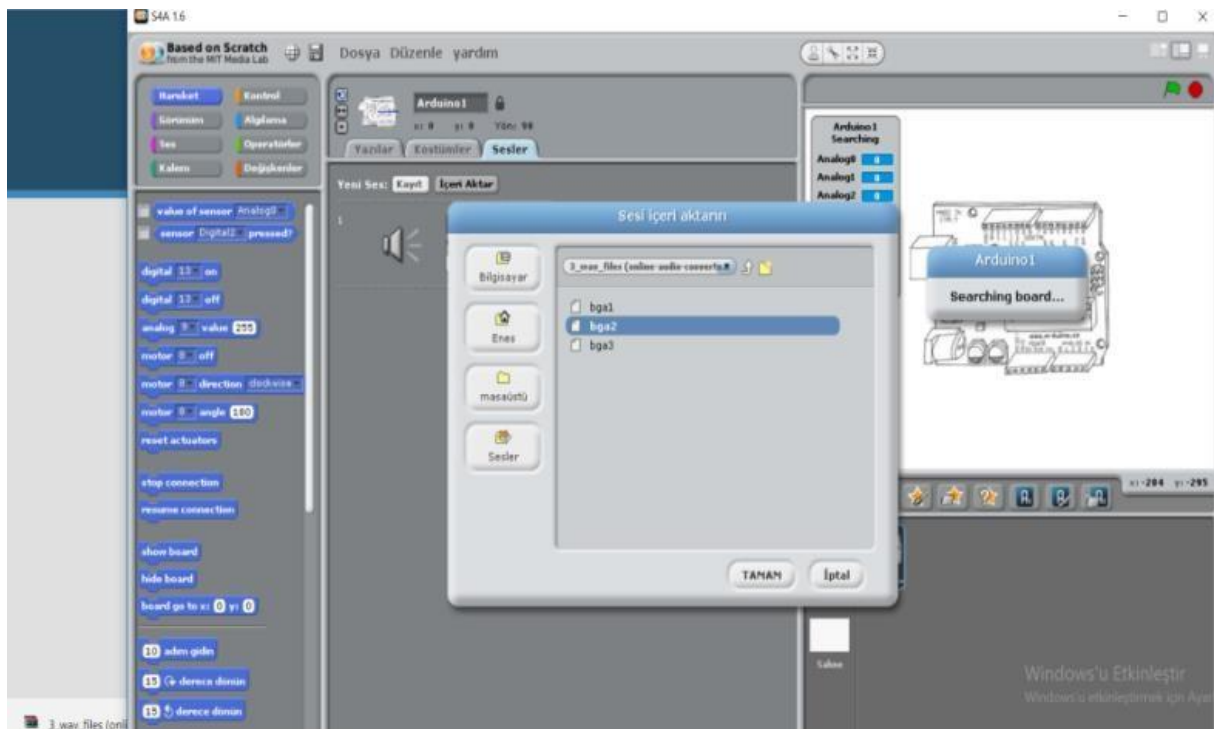
The potentiometer and the ultrasonic sensor are connected to the common anode via the gnd. To measure the potentiometer value, the analogue must be connected to any pine.

The ultrasonic sensor is connected to digital number 2 pine.

Cardboard can be powered by either a 9 volt or a pcbusb cable.



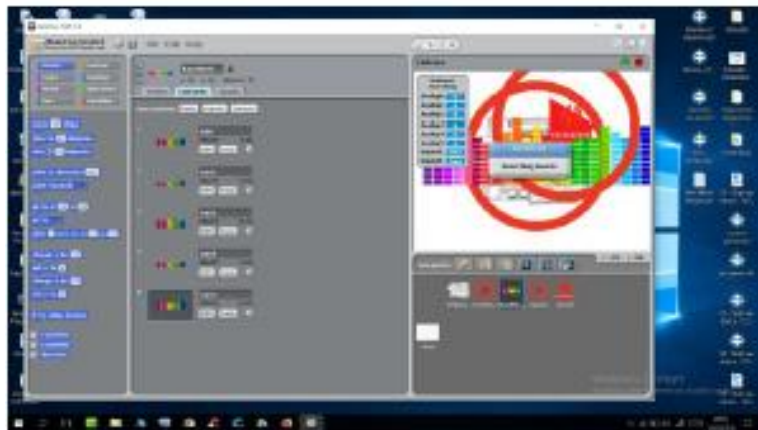
Step 2: We are loading music to program. The format of the music must be Mp3 or Wav. Sounds/Import/Files.



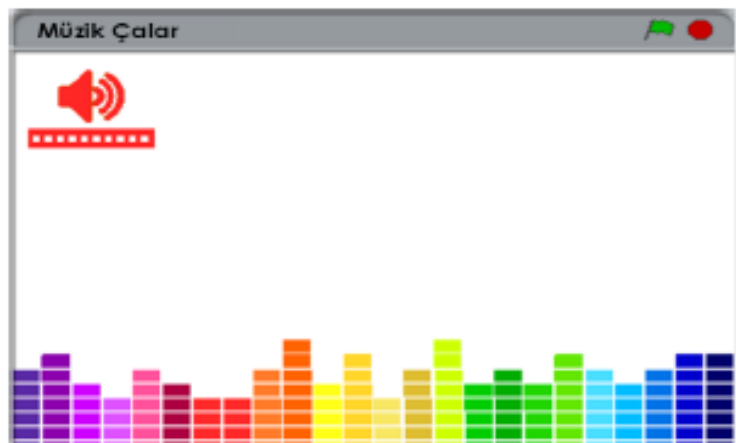
We upload images that we will use in the system. This can be done by verifying the picture or by loading the sprite.



It is added as the first character from the images to be loaded for the melody, and the others are added as costumes.



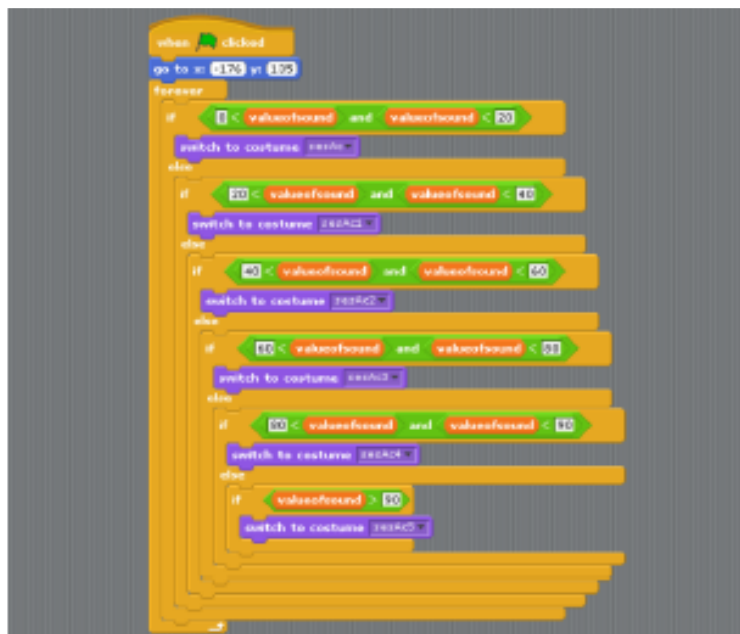
The final screen layout should look like this. Images added as characters will appear directly on the screen. It can be hidden with code to not show them in the first stage.



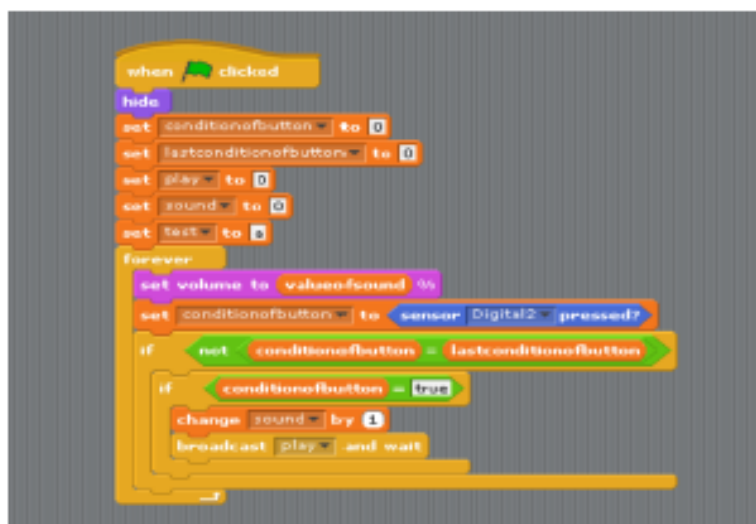
Step 3: We create the variables to be used for coding.



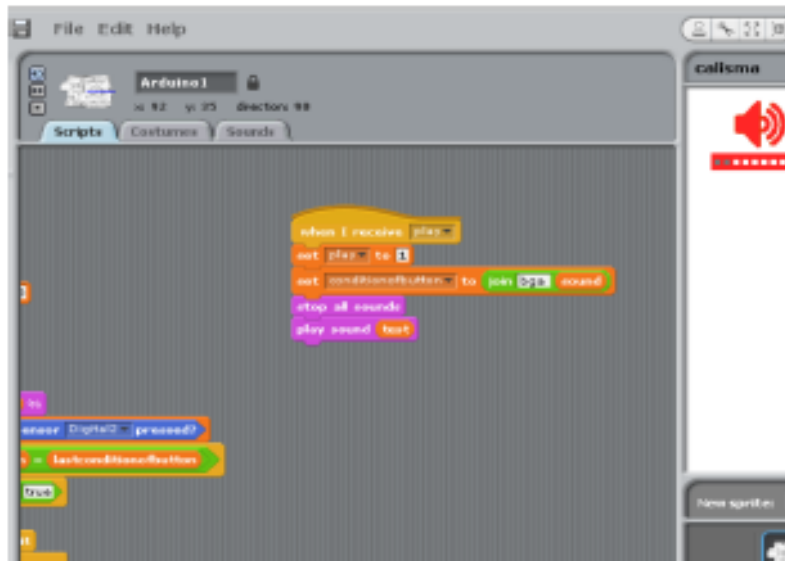
Our volume-editing code blocks are visible. The program codes will be loaded into Drive along with the mBlock codes.



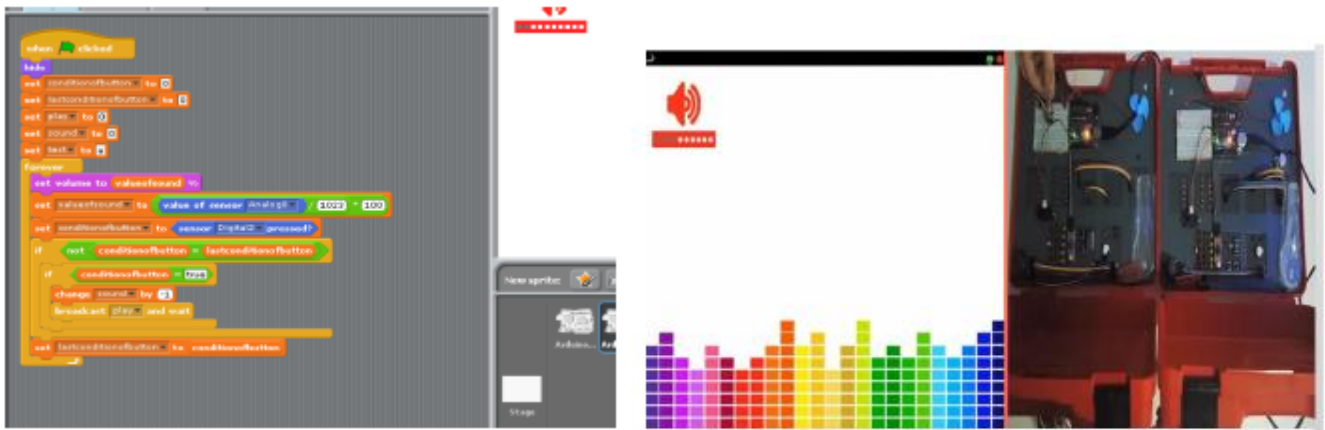
Our other code blocks:



Required code blocks for the second Arduino card



The final image of our system. Our music players are now ready.



## Last Step: mBlock codes for mBot

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Bağlantıyı kes - Kaydedilmedi

Dosya Düzenle Bağlan Kartlar Uzantılar Lisan Yardım

Sardana

Hareket Görünüm Ses Kalem Van&Blok

Olaylar Kontrol Algılama İşlemler Robotlar

10 adım git  
15 derece dön  
15 derece dön  
90° yönüne dön  
ye doğru dön  
x: 11 y: 24 noktasına git  
fare oku 'na git  
sn.de x: 11 y: 24 a süzül  
x1 10 arttır  
x 0 olsun  
y yi 10 arttır  
y 0 olsun  
kenara geldiyse sek

Kuklalar Yeni kukla: Yeni dekor

Sahne 1 dekor  
Yeni dekor

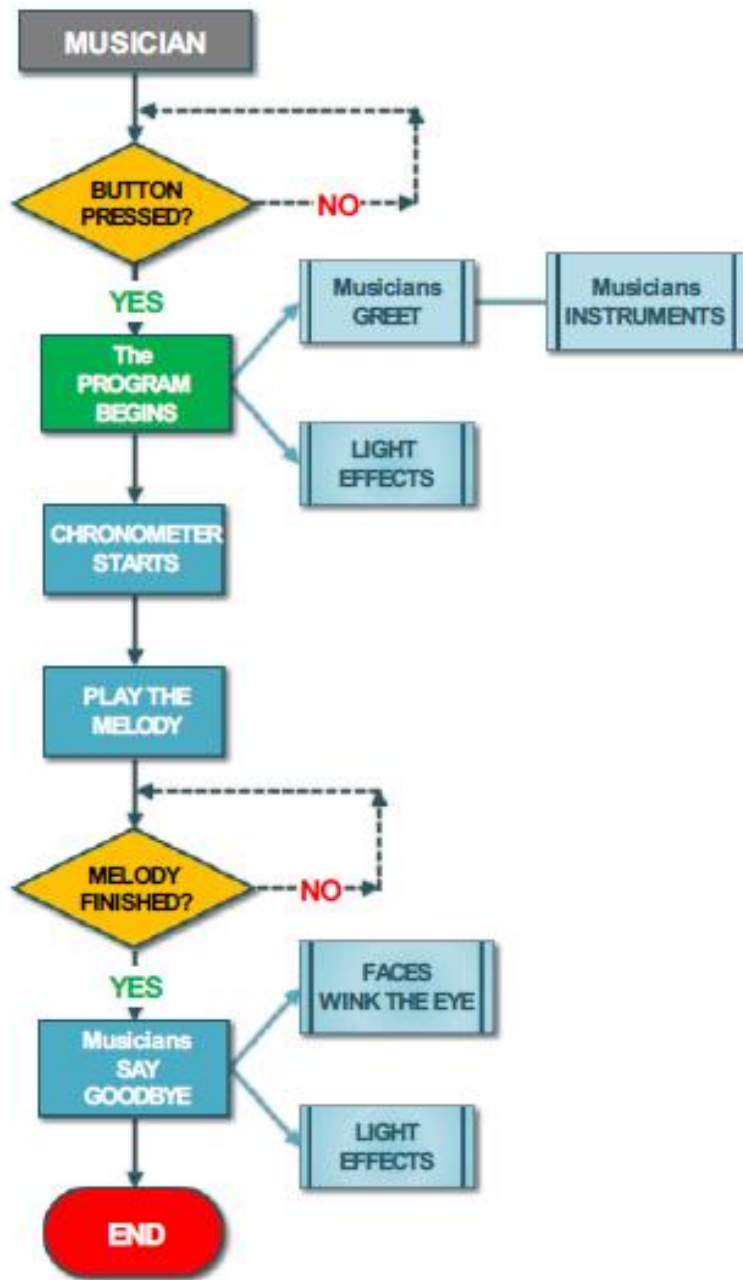
mBot Programı

```
kenara geldiyse sek
M1 motorun hızı 100 yap
M2 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 0 yap
M2 motorun hızı 100 yap
M1 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 100 yap
M2 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 0 yap
M2 motorun hızı 100 yap
M1 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 100 yap
M2 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 0 yap
M2 motorun hızı 100 yap
M1 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 100 yap
M2 motorun hızı 0 yap
1 saniye bekle
M1 motorun hızı 0 yap
M2 motorun hızı 100 yap
M1 motorun hızı 0 yap
1 saniye bekle
```

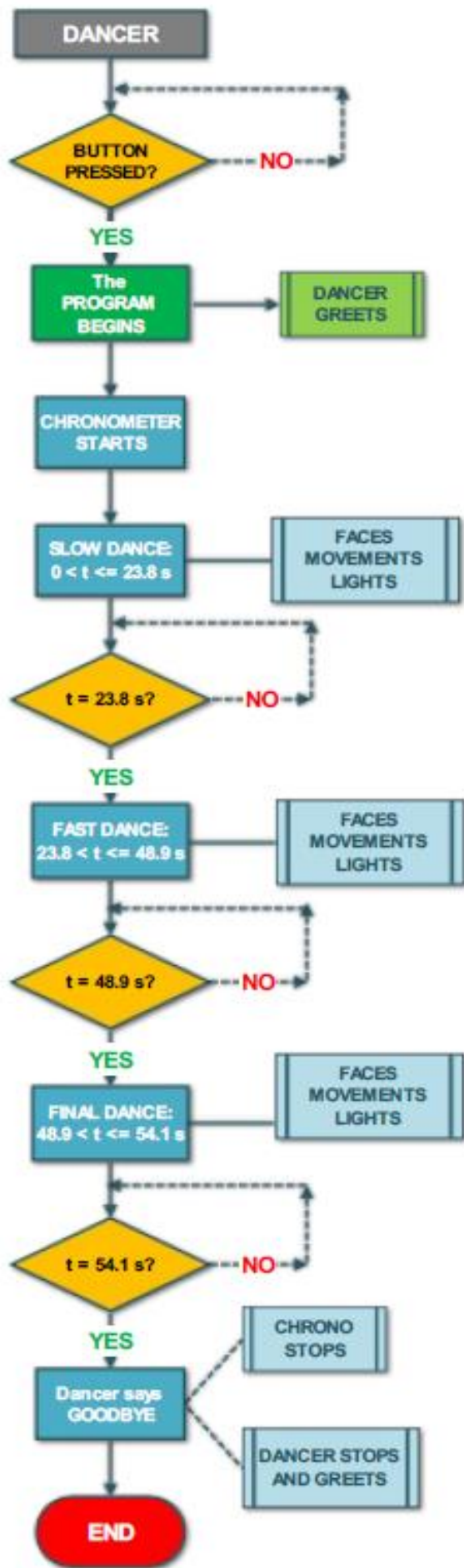
x: 11 y: 24  
11:40  
14.05.2018

# FLOW CHART

## First version







## STUDENT'S EVALUATION

For the evaluation of the students in this activity, use the Evaluation Rubric designed for this project.

## BIBLIOGRAPHY

“Guía de Scratch”. CEIP de Cella, (Teruel). Pdf.

“Scratch. Guía didáctica para profesores”. (Pdf). ([www.isuriarte.com](http://www.isuriarte.com)).

“Jugando con MBlock”. Makeblock España.

“Divirtiéndome con MBot”. Susana Oubiña.

Comunidad de Makeblock en español. (<http://www.makeblock.es/foro/>).

“Curso de Scratch + Arduino”. J. Javier Esquiva Mira.

## MORE INFORMATION

### DIFFICULTIES:

- RESET THE CHRONO: within a loop, when the chrono starts, if we enter waiting times, the activity is disorganized.