

# MENTALIST MBOT



## 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

# MENTALIST MBOT

## ABSTRACT

The activity consists in asking “yes” or “no” questions to the mBot, and this will answer us. This activity is born before the need to introduce the programming in the students.

## DIDACTIC OBJECTIVES

### TECHNOLOGY

- ❖ Create a programming base for students.

STEM Subject:    Science             Technology             Engineering             Mathematics

Education Level:            12-14 years             14-16 years

## PROBLEM STATEMENT

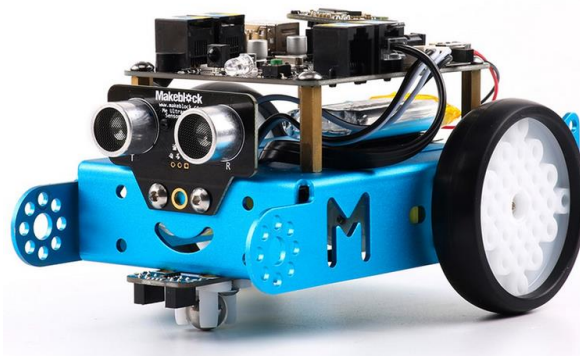
Need to create a programming base for students.

The students who study computer science generally have the problem of not knowing programming, so to do dynamic activities, simple that gradually increase the difficulty, awaken the curiosity and the desire to learn students' programming.

In this way, it will create a knowledge base in programming that will avoid starting a university career without having any knowledge.

## BOM (Bill of Materials Needed)

- mBot => Ref. 90054



## ACTIVITY DESCRIPTION

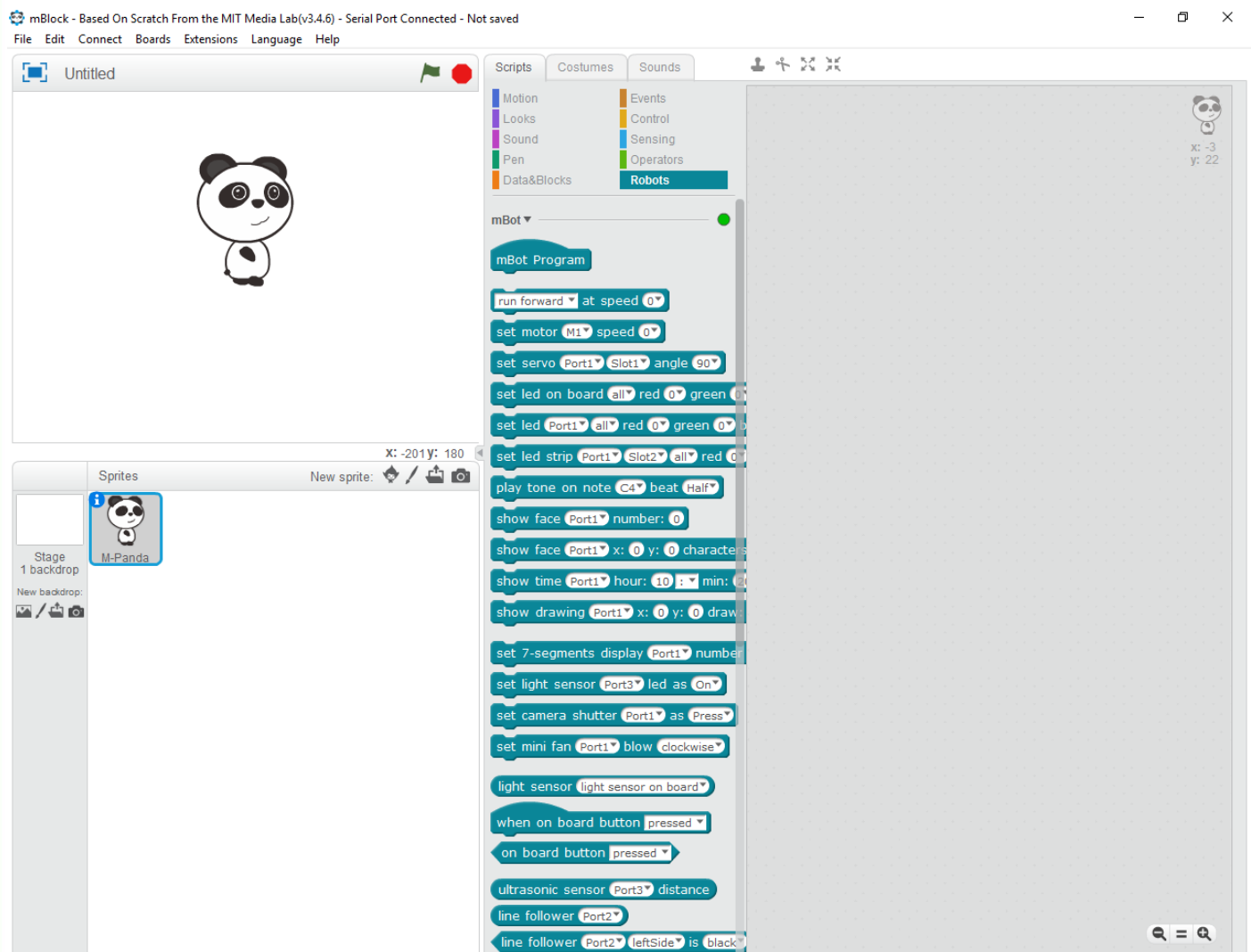
This activity consists of asking “yes” or “no” questions to mBot, and this answer us. mBot will emit two colours, red and green, and when we approach our hand or other object during one of the two colours, MBOT will save the answer.

If we pass the hand when the red colour appears, mBot save “no”, and if we pass it when the green colour is active, mBot will save “yes”.

In this way, we can fool people saying that the mBot can read the mind or tell the truth, or rather, that we want it to say :D

First of all, we will have to program the instructions to mBot.

For do this, we use mBlock software (<http://www.mblock.cc/download/>)

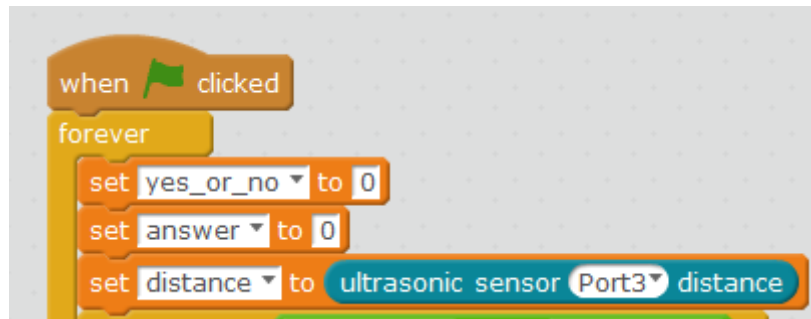


The screenshot displays the mBlock software interface. The main workspace shows a panda sprite on a stage. The script editor is open, showing a sequence of blocks for the mBot program. The blocks include: 'run forward at speed 0', 'set motor M1 speed 0', 'set servo Port1 Slot1 angle 90', 'set led on board all red 0 green 0', 'set led Port1 all red 0 green 0', 'set led strip Port1 Slot2 all red 0', 'play tone on note C4 beat Half', 'show face Port1 number: 0', 'show face Port1 x: 0 y: 0 characters', 'show time Port1 hour: 10 min: 0', 'show drawing Port1 x: 0 y: 0 draw', 'set 7-segments display Port1 number', 'set light sensor Port3 led as On', 'set camera shutter Port1 as Press', 'set mini fan Port1 blow clockwise', 'light sensor light sensor on board', 'when on board button pressed', 'on board button pressed', 'ultrasonic sensor Port3 distance', 'line follower Port2', and 'line follower Port2 leftSide is black'.



First, we paired the MBLOCK software with MBOT. For do this, we use the 2.4G Wireless<sup>1</sup> Serial Port. Then we proceed to perform the programming:

1. Create the variables:

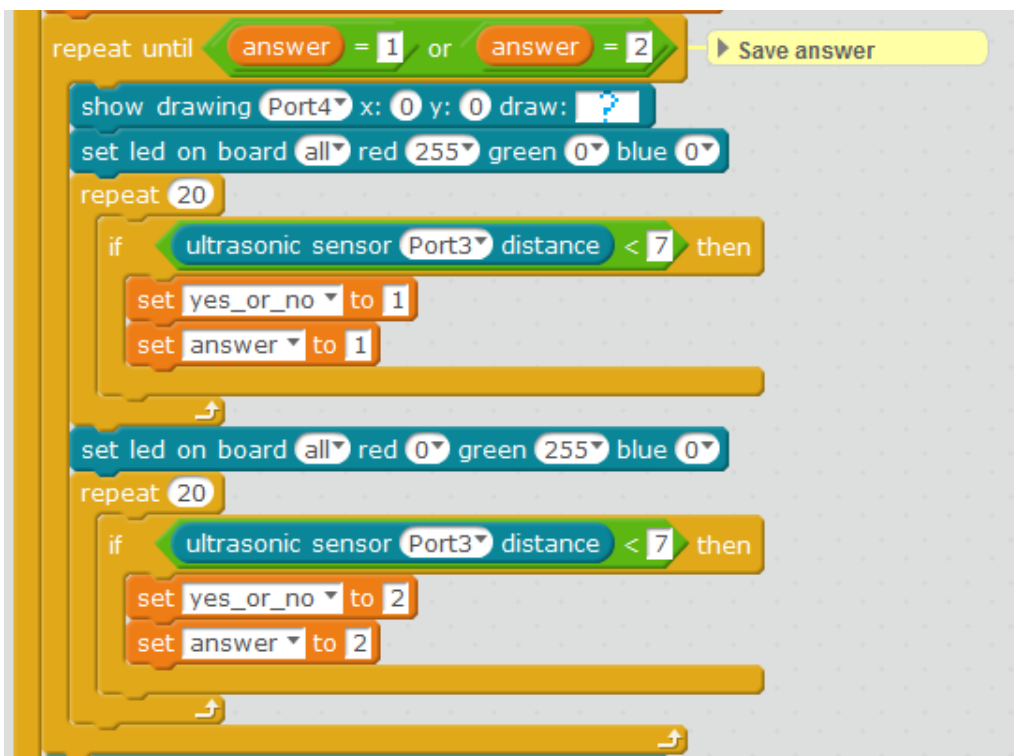


The variable “yes\_or\_no” save the value of the answer that the MBOT has read.

The variable “answer” will serve to know if the MBOT has answered.

And the variable “distance” will serve to know to the distance that we pass the hand by the ultrasonic sensor.

2. Save the answer:

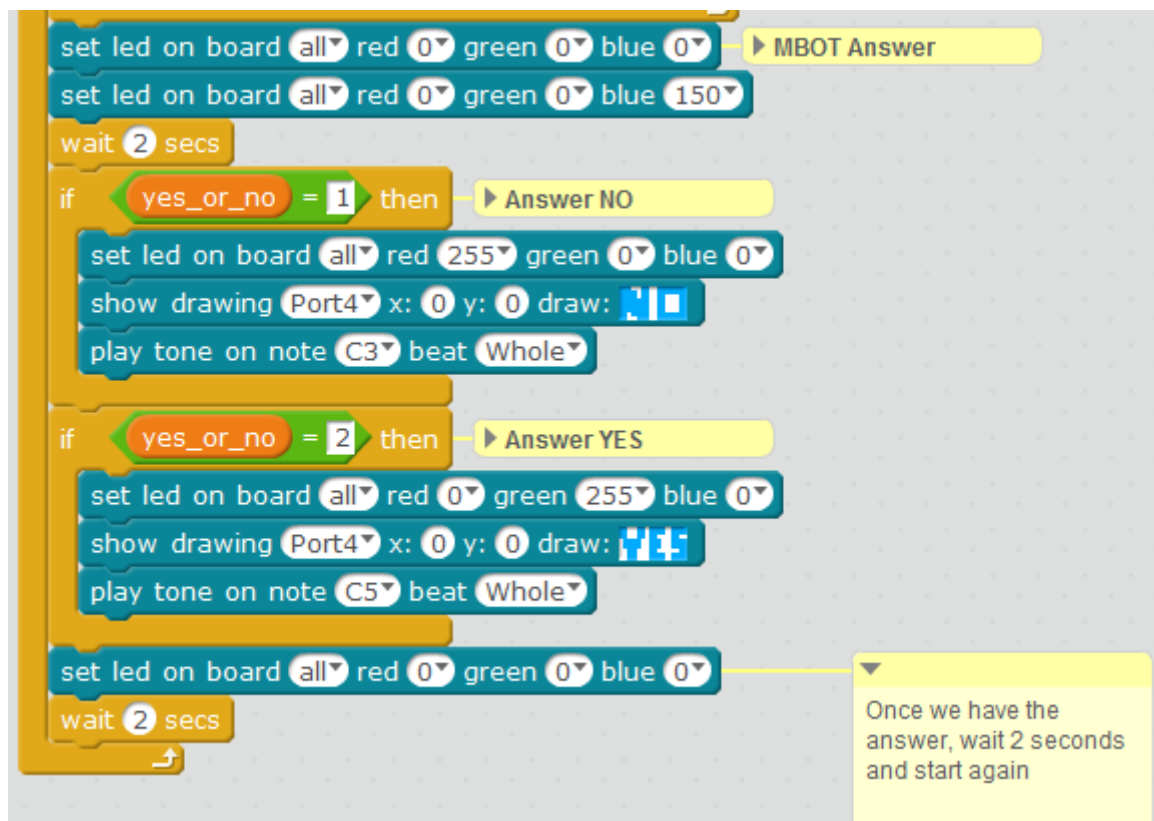


This block consists of saving the answer we tell the MBOT, so the loop will exit when the MBOT has received a response.

With the LED matrix we draw an interrogation to make the activity more graphical.

Every 2 seconds the colours will be exchanged, and the moment we pass the hand to a distance of less than 7cm in one of the two colours, the MBOT will record the answer.

### 3. MBOT reproduce the answer:



```
set led on board all red 0 green 0 blue 0 → MBOT Answer
set led on board all red 0 green 0 blue 150
wait 2 secs
if yes_or_no = 1 then → Answer NO
  set led on board all red 255 green 0 blue 0
  show drawing Port4 x: 0 y: 0 draw: [Interrogation]
  play tone on note C3 beat Whole
if yes_or_no = 2 then → Answer YES
  set led on board all red 0 green 255 blue 0
  show drawing Port4 x: 0 y: 0 draw: [Yes]
  play tone on note C5 beat Whole
set led on board all red 0 green 0 blue 0
wait 2 secs
```

Once we have the answer, wait 2 seconds and start again

Finally, we will put the LEDs in blue to indicate that the MBOT is going to make the answer, and depending on the answer that it has registered, MBOT emit a sound, a colour and the LED matrix will draw “yes” or “no”.

Once the response is issued, the same process will be repeated after 2 seconds.



4. That would be our final code:

```
when green flag clicked
  forever loop
    set yes_or_no to 0
    set answer to 0
    set distance to ultrasonic sensor Port3 distance
    repeat until (answer = 1 or answer = 2)
      show drawing Port4 x: 0 y: 0 draw: ?
      set led on board all red 255 green 0 blue 0
      repeat 20
        if ultrasonic sensor Port3 distance < 7 then
          set yes_or_no to 1
          set answer to 1
      end repeat
      set led on board all red 0 green 255 blue 0
      repeat 20
        if ultrasonic sensor Port3 distance < 7 then
          set yes_or_no to 2
          set answer to 2
        end if
      end repeat
    end repeat
    set led on board all red 0 green 0 blue 0
    set led on board all red 0 green 0 blue 150
    wait 2 secs
    if yes_or_no = 1 then
      set led on board all red 255 green 0 blue 0
      show drawing Port4 x: 0 y: 0 draw: NO
      play tone on note C3 beat Whole
    end if
    if yes_or_no = 2 then
      set led on board all red 0 green 255 blue 0
      show drawing Port4 x: 0 y: 0 draw: YES
      play tone on note C5 beat Whole
    end if
    set led on board all red 0 green 0 blue 0
  end forever
```

The code is a Scratch script that runs in a forever loop. It starts by setting variables 'yes\_or\_no' and 'answer' to 0, and 'distance' to the value from an ultrasonic sensor at Port3. A 'repeat until' loop (containing a 'Save answer' comment) ensures that 'answer' is either 1 or 2. Inside this loop, there are two 'repeat' blocks of 20 iterations each. The first 'repeat' block checks if the distance is less than 7; if so, it sets 'yes\_or\_no' and 'answer' to 1, and turns the red LED on. The second 'repeat' block checks if the distance is less than 7; if so, it sets 'yes\_or\_no' and 'answer' to 2, and turns the green LED on. After the 'repeat until' loop, the code turns off all LEDs, turns the blue LED on (150 intensity), and waits 2 seconds. It then checks 'yes\_or\_no': if 1, it turns the red LED on, shows a drawing of a 'NO' sign, and plays a C3 note. If 2, it turns the green LED on, shows a drawing of a 'YES' sign, and plays a C5 note. Finally, it turns off all LEDs and waits 2 seconds before the loop repeats.

We left some example images:

