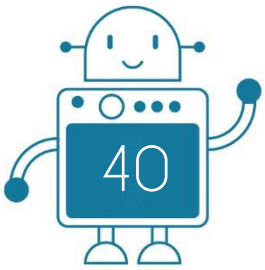


SPEED RADAR

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

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

SPEED RADAR

ABSTRACT

The activity consists of developing a speed radar.

It will be able to calculate the speed, (cm/s), to which an object moves.

In first version, it used an Arduino Uno board.

In second version, it is created exclusively with the mBot hardware. It also proposes the conversion of cm/s to km/h, since it is a unit of measure more recognizable by the students.

DIDACTIC OBJECTIVES

- Learning the calculation of speed relating distance and time.
- Learning the conversion of physical units of speed.
- Learning how to use a 7 segment display.
- Learning about using a distance sensor.

TECHNOLOGY

- Develop all the Arduino System and programming the code.

ENGINEERING

- Create a structure of radar.

MATHEMATICS

- Calculate the speed.

STEM Subject: Science☐ Technology ☒ Engineering☒ Mathematics☒

Education Level: 12-14 years☐ 14-16 years☒

PROBLEM STATEMENT

The mBot will act as a speed radar. It will detect a vehicle circulating ahead of it and it will measure its speed. For this purpose the mBot will use a distance sensor.

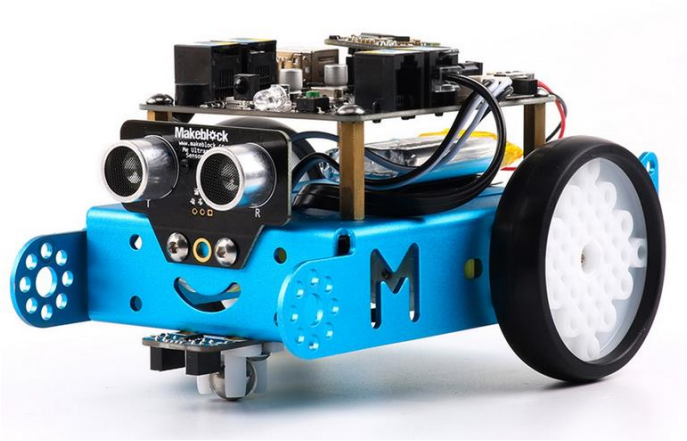
BOM (Bill of Materials Needed)

First version (with Arduino):

1. mBlock Software.
2. Arduino IDE Software.
3. Protoboard.
4. Ultrasonic Sensor.
5. LCD Display.

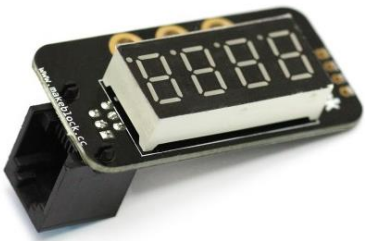
Second version (without Arduino):

- mBot => Ref. 90054



- Me **Ultrasonic Sensor**:

- Me **7-Segment Serial Display - Red**:

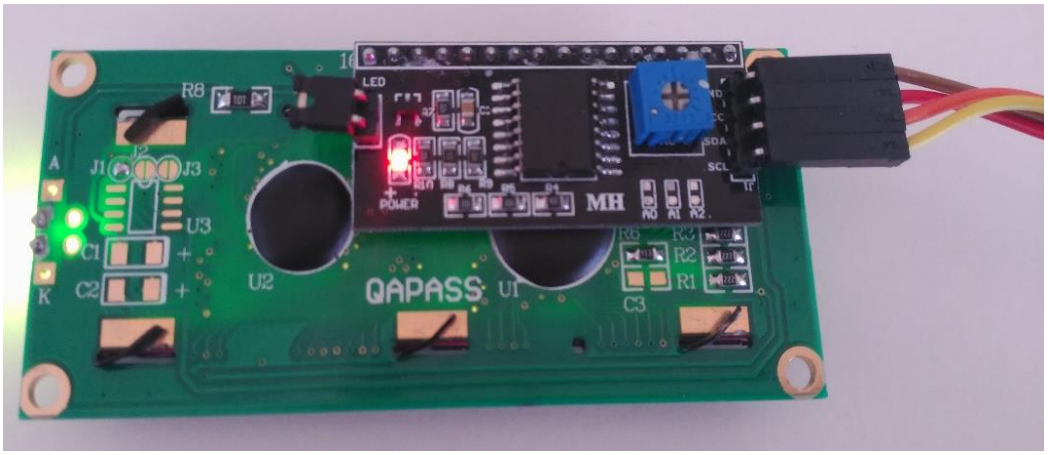


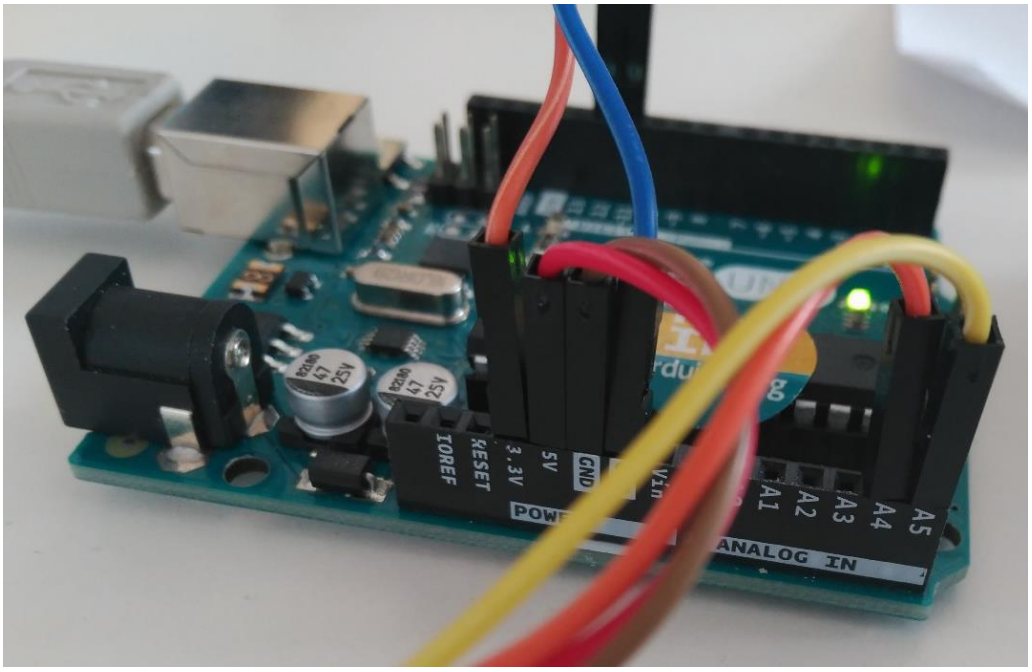
First version (with Arduino):

The speed radar will calculate the speed, in cm/sec, at which an object moves. In this case it will be a toy car. The speed will be calculated through the Arduino ultrasonic sensor.

First of all, we create the Arduino structure and connect the sensors.

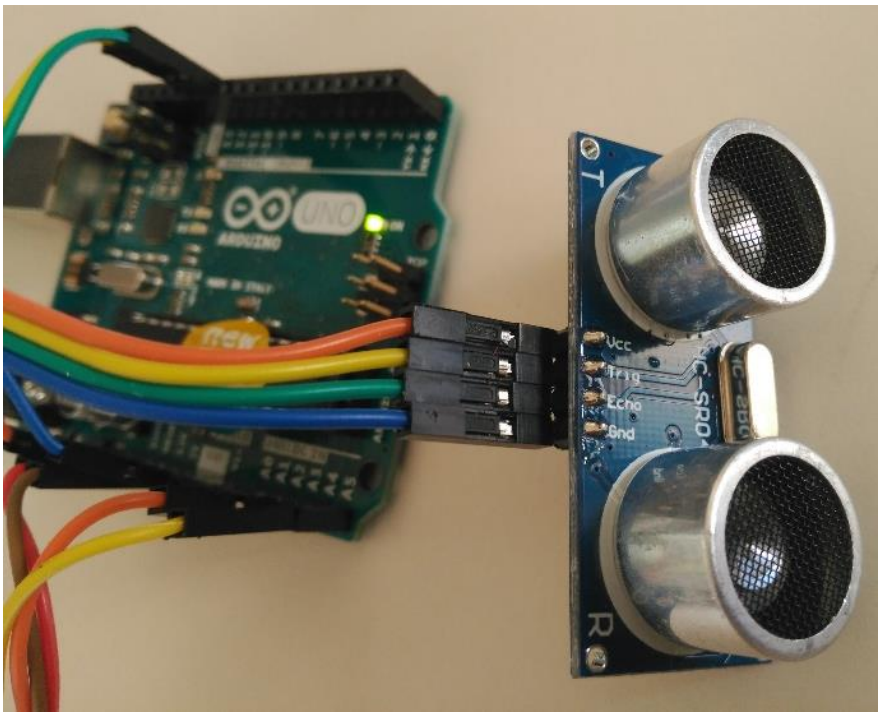
1. Connect the LCD Display to Arduino Shield:

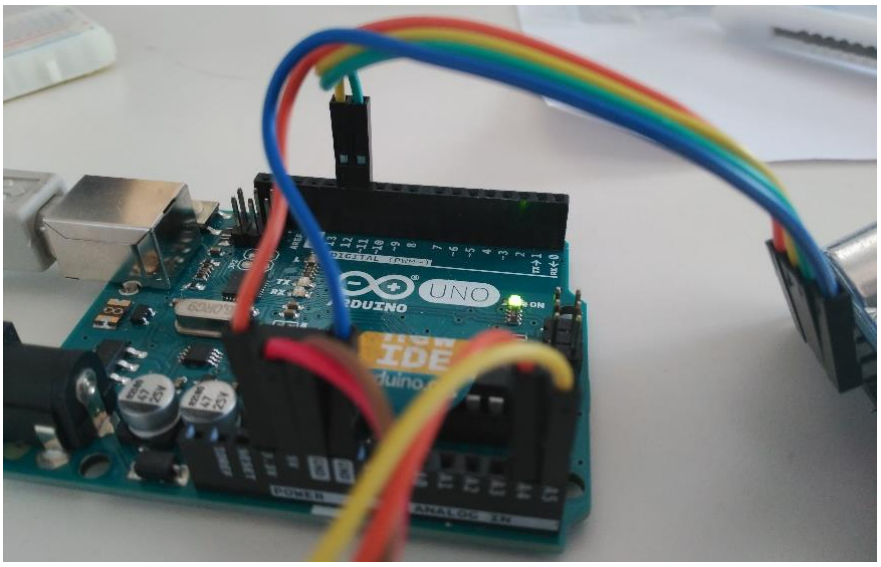




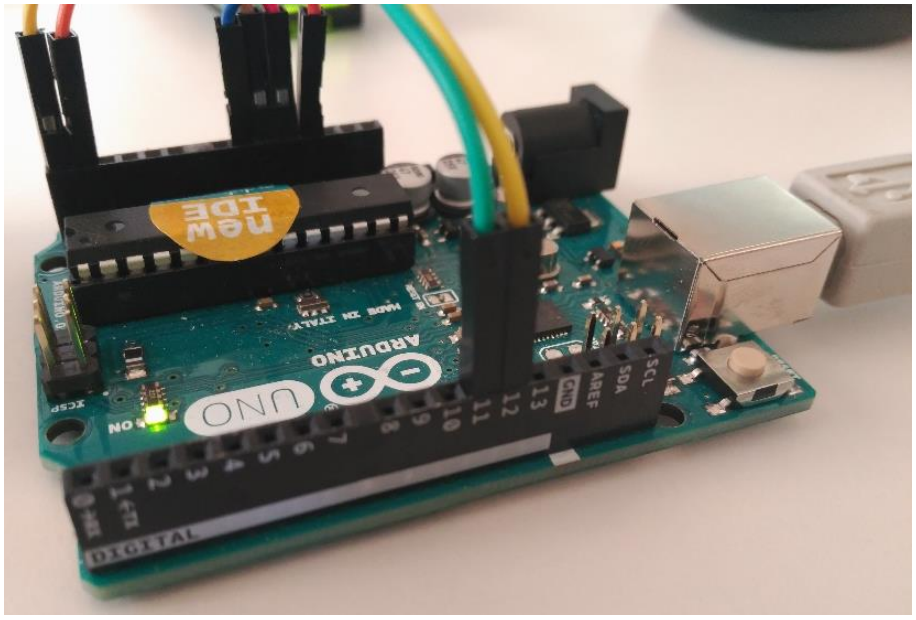
- **GND Pin** => [Brown cable] is connected in GND port of the Arduino Uno Shield.
- **VCC Pin** => [Red cable] is connected in 5V port of the Arduino Uno Shield, or you have a protoboard, you must be connect in the "+" red section. And with another cable, connect the "+" red section with 5V port to Arduino Uno Shield.
- **SDA Pin** => [Orange cable] is connected in Analog 4 port of the Arduino Uno Shield.
- **SCL Pin** => [Yellow cable] is connected in Analog 5 port of the Arduino Uno Shield.

2. Connect the Ultrasonic Sensor to Arduino Shield:



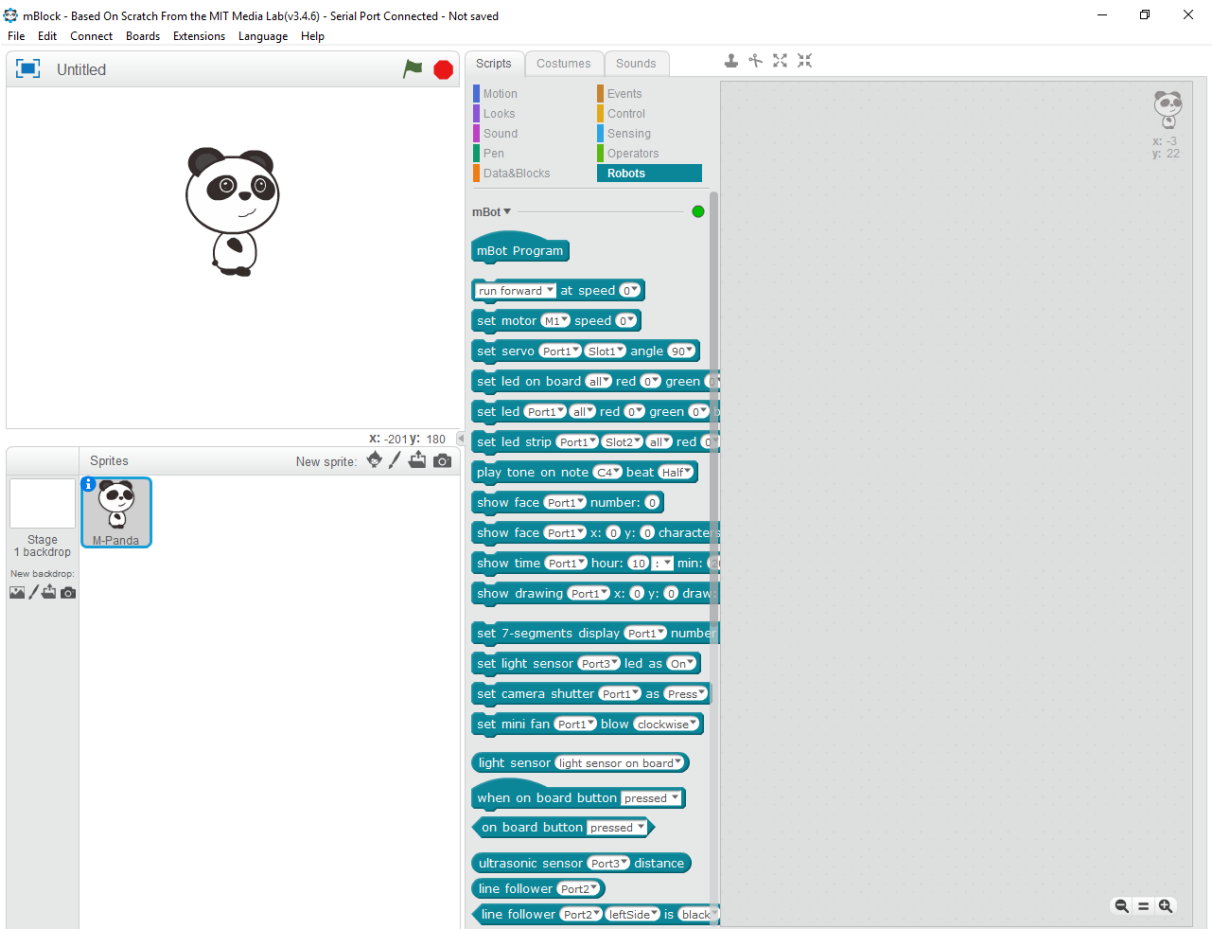


- **GND Pin** => [Blue cable] is connected in GND port of the Arduino Uno Shield.
- **VCC Pin** => [Orange cable] is connected in 5V port of the Arduino Uno Shield, or you have a protoboard, you must be connect in the "+" red section. And with another cable, connect the "+" red section with 5V port to Arduino Uno Shield.
- **Trig Pin** => [Yellow cable] is connected in Digital 12 port of the Arduino Uno Shield.
- **SCL Pin** => [Green cable] is connected in Digital 11 port of the Arduino Uno Shield.

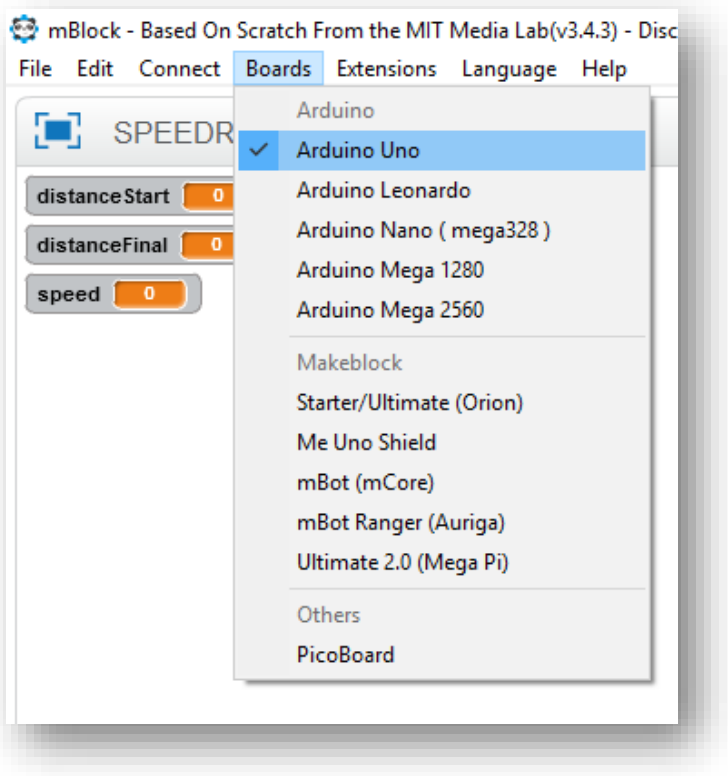


3. Create the Radar Structure:

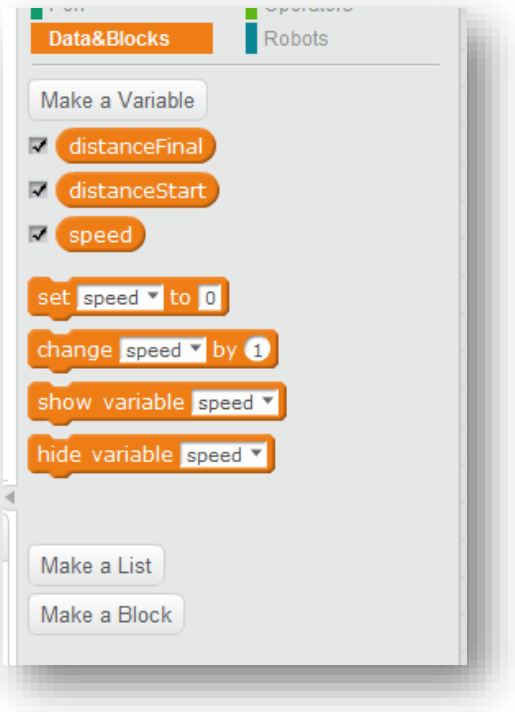
Now, we will develop the instructions to Arduino Uno Shield in mBlock Software. (http://www.mblock.cc/download/)



First, we paired the Arduino UNO Shield with software with MBOT:



1. Create the variables:

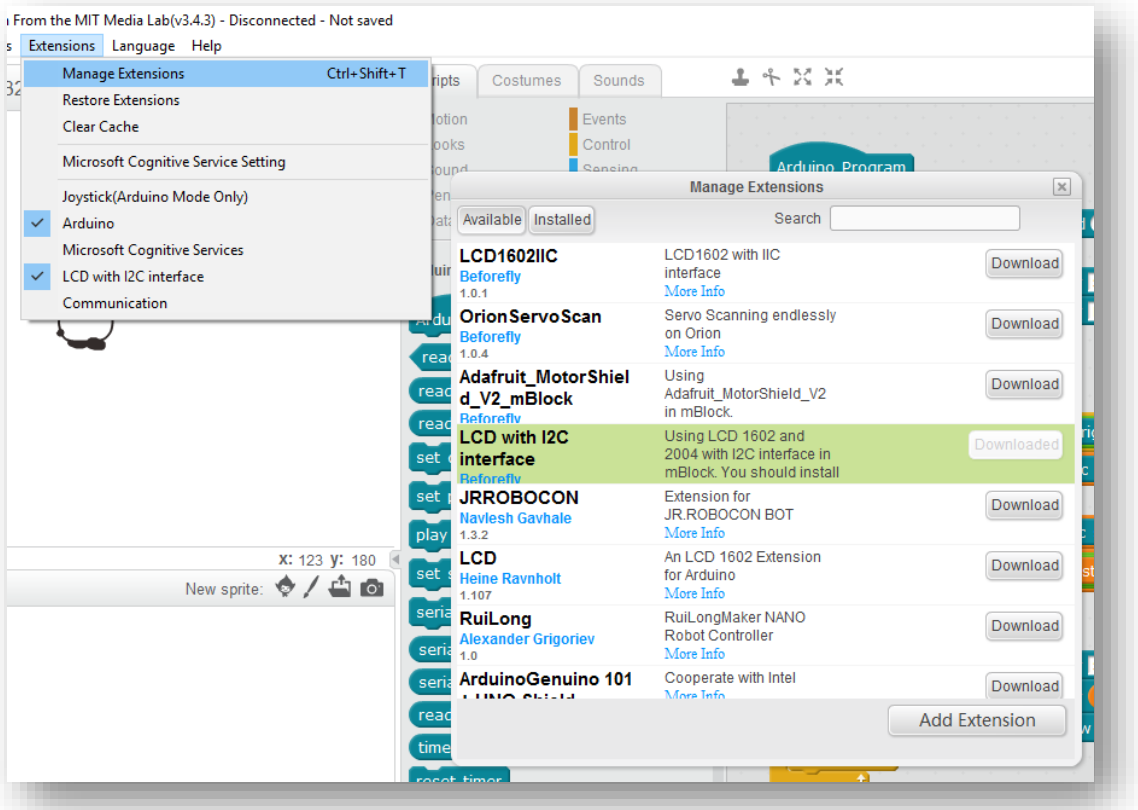


The variable “distanceStart” save the value of first distance that ultrasonic sensor has read.

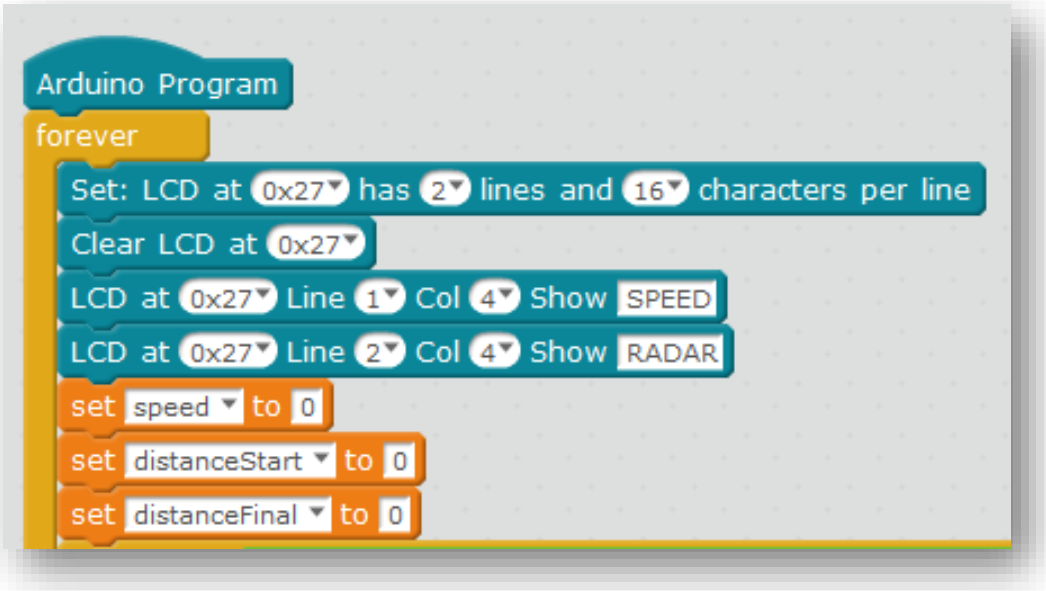
The variable “distanceFinal” save the value of the second distance that ultrasonic sensor has read.

And the variable “speed” will be the result after calculate the speed.

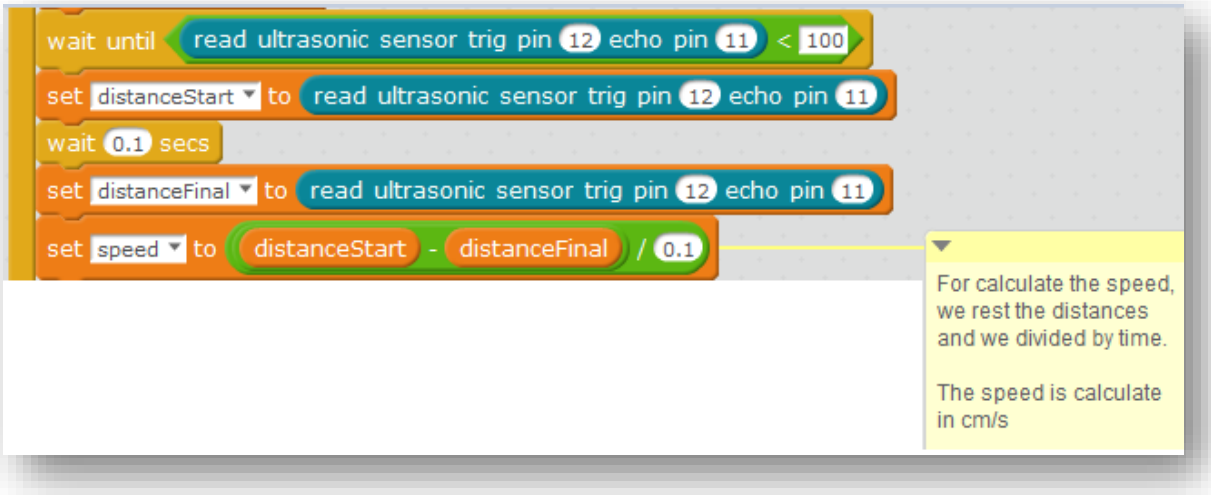
2. Import the “LCD with I2C Interface” extension:



3. Set the parameters of the LCD Display and initialize the variables:

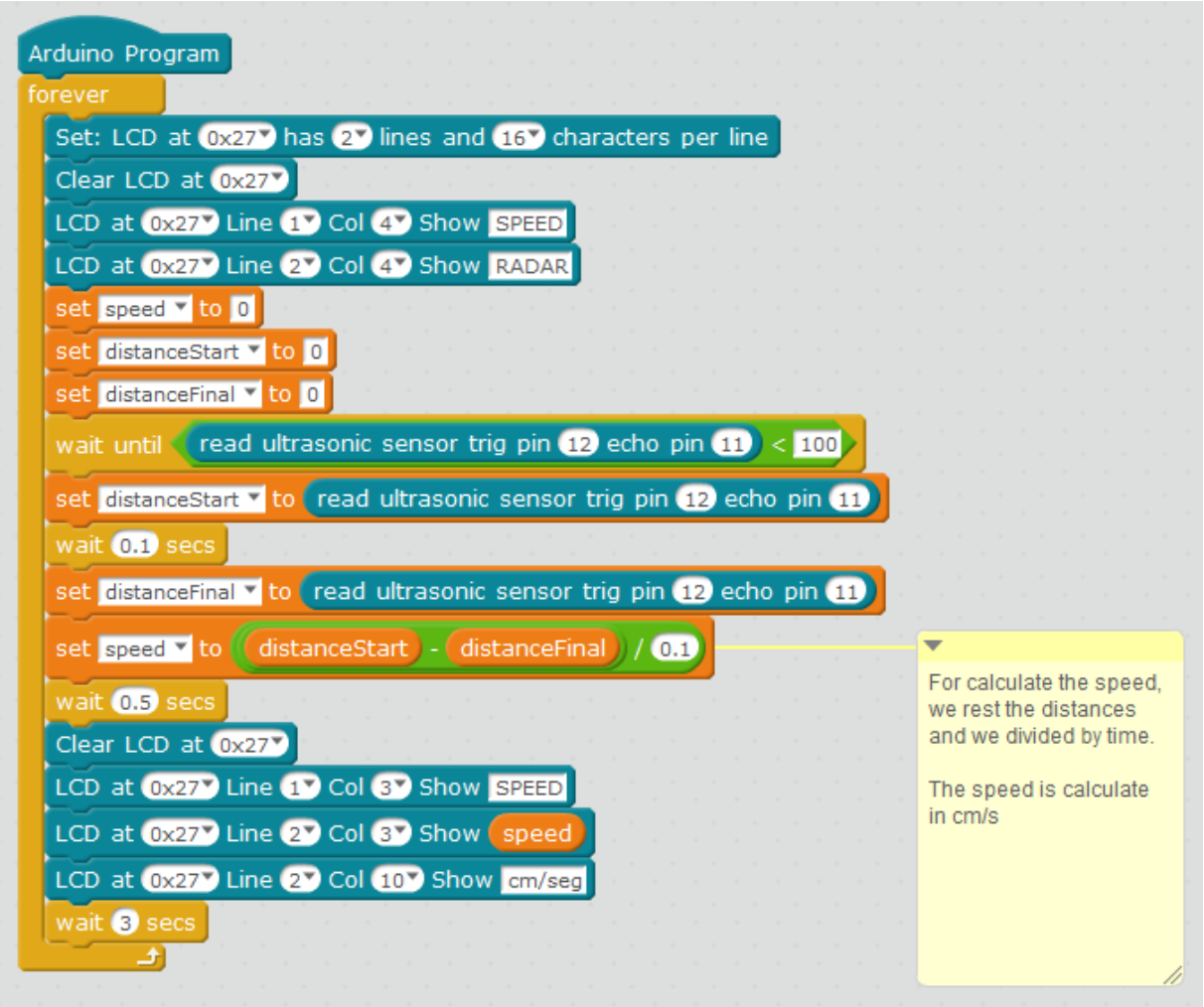


4. Programming the instructions that calculate the distance:

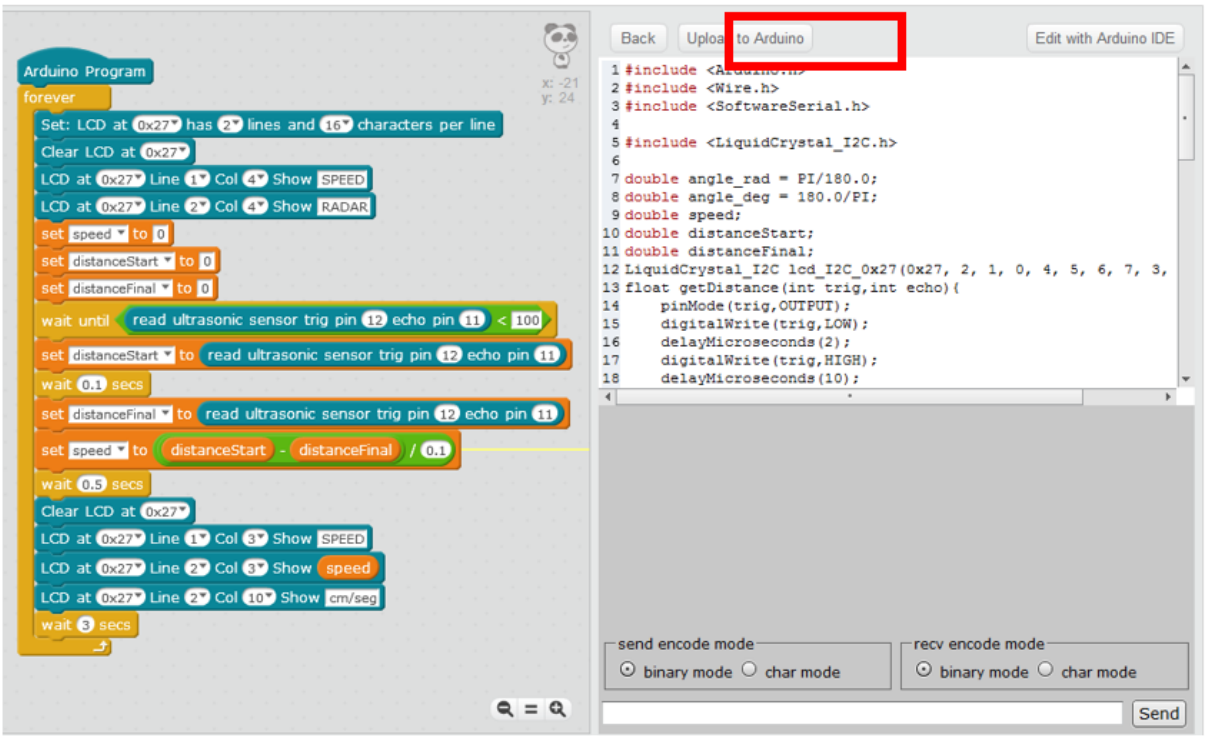


5. Finally, we show the results on LCD Display:

6. That would be our final code:



7. For upload the code to Arduino Uno Shield, we clicked in “Arduino Program” with left mouse button and another window appears. In the new window, we click in “Upload to Arduino”:



We left some example images:


SCX
SPEED
115.24 cm/sec
SCALEXTRIC


BOXES

Second Version (without Arduino):

The students' work is to know the mathematical formula relating speed, space and time and with the teacher's help, create the flowchart and develop the code that allows the mBot work as a speed radar.

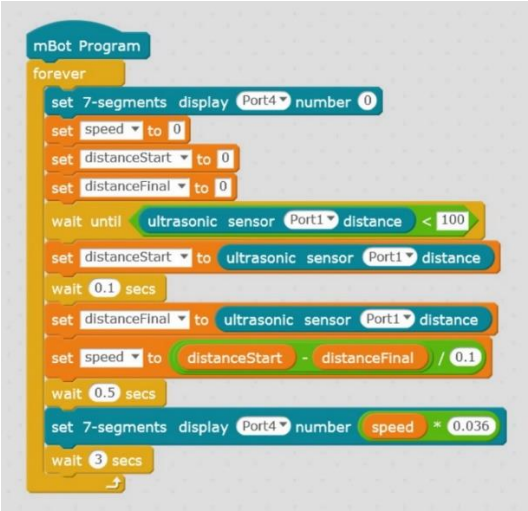
The variable "space" will be obtained by the difference in between two measured distances. The variable "time", will be the gap of time between those two measurements.

The speed radar will remain in standby mode, until a "vehicle" passes in front of it. At that time, it will calculate the speed of the vehicle and display it on the 7-segment display. (It will be calculated in km/h).

Code of the activity**:**

The code is very similar to the original AIJU's code. We have only made a translation of the Arduino commands to scratch commands and we have added the operation to convert cm/s to km/h.

The code is very short and simple, but powerful.



We will record the code in the arduino board of the mBot. In this way, the mBot will work independently of the computer.

How to load a program on the arduino mBot board using mBlock:

In order to load a program on the board using mBlock:

1. Choose mBot in the Board tab of the mBlock menu.

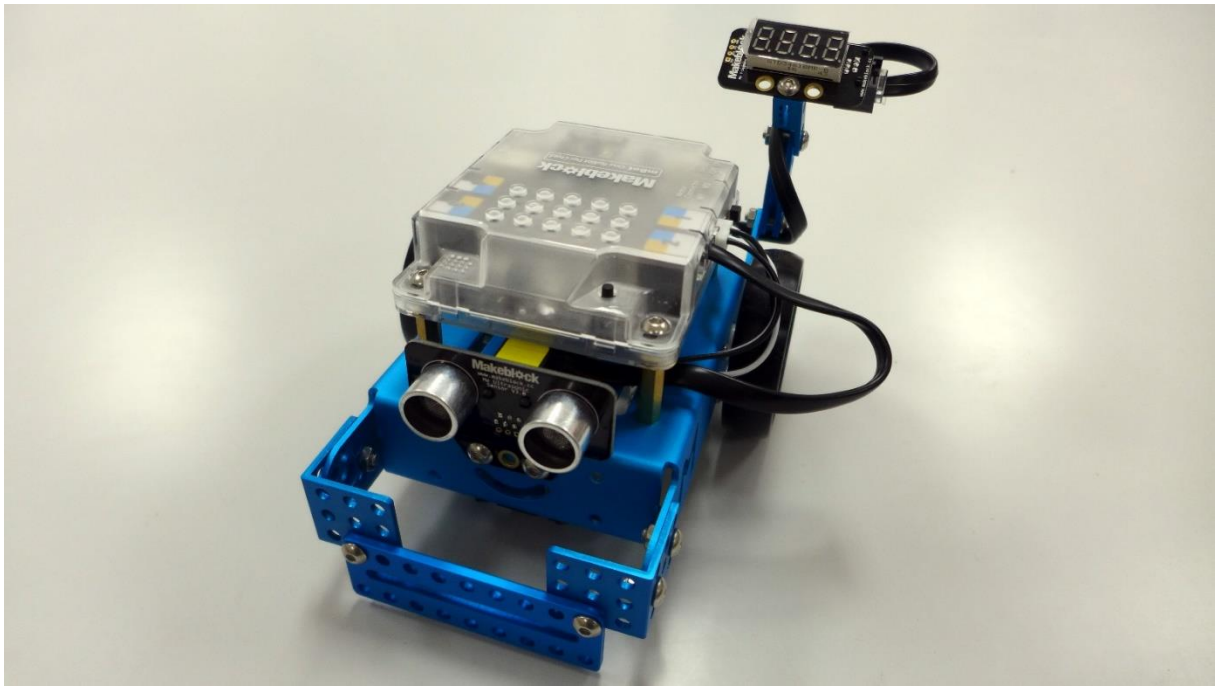
2. Connect the USB and choose "Serial Port" in the connect tab.

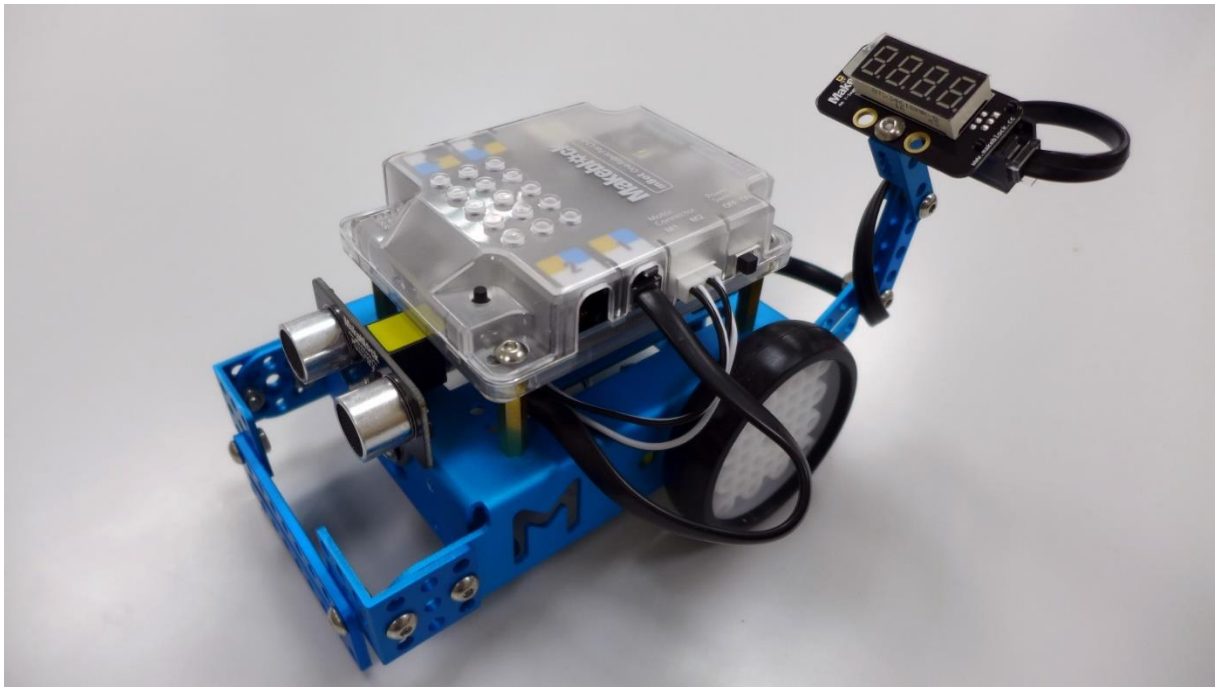
3. In the tab edit, choose "Arduino Mode" (In the program that we are going to load, instead of the green flag, we will put the blue command "mBot program").

4. A window with the code will open to record it on the Arduino board of mBot. You can, if you want, modify your program. Finally, click on Upload to Arduino.

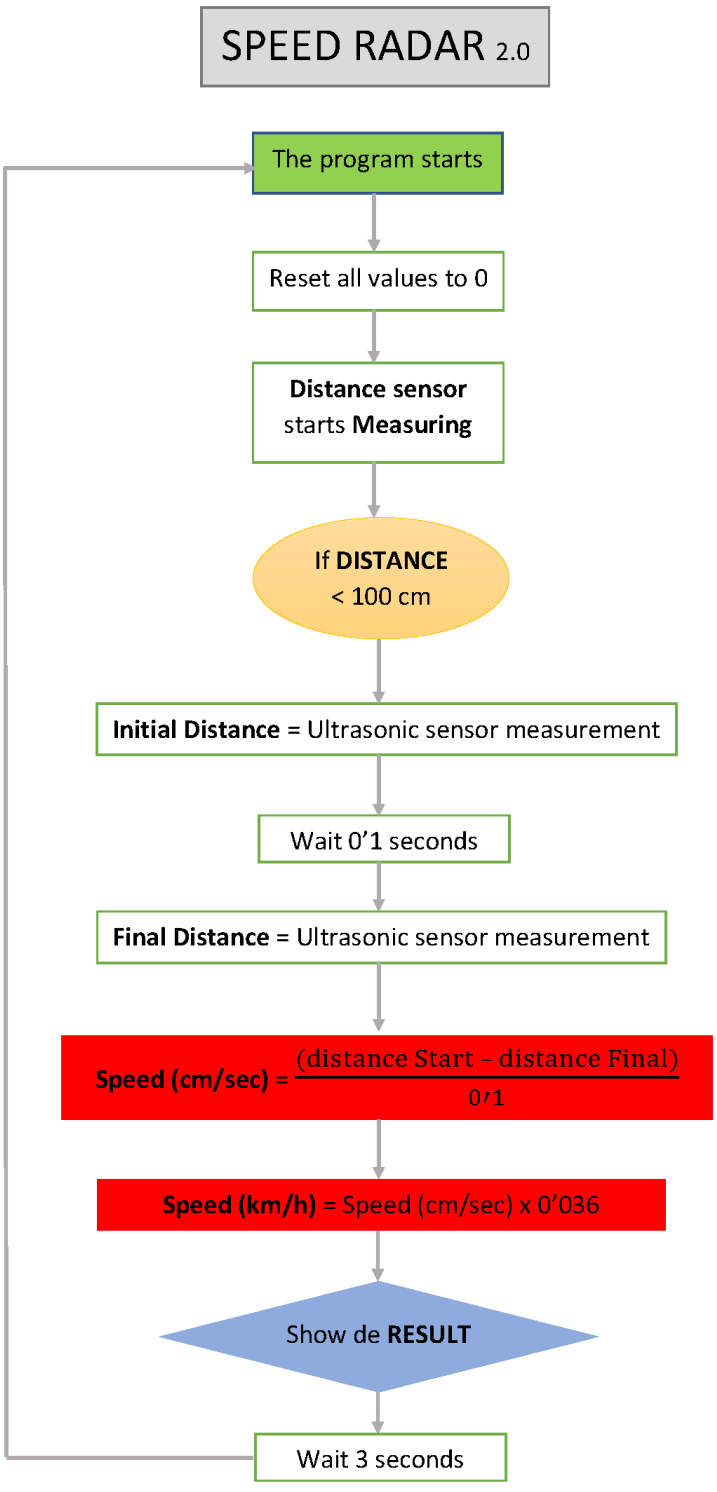
5. If there have been no errors, a message will be sent informing that the program has been recorded correctly. At this moment you will be able to start enjoying the program introduced in the robot, without the computer turned on. For doing this, you must disconnect the USB cable and connect the batteries (or lithium battery) of the robot. You will see that your mBot works independently.

Structural composition:





FLOW CHART



SCALABILITY

This activity is complicated because there may be problems with libraries and their versions. In addition, the ultrasound sensor is very sensitive and difficult to configure.

Every time you move it, you have to configure it to be as accurate as possible.