

# RADAR DE VELOCIDAD



## 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

# RADAR DE VELOCIDAD

## RESUMEN

La actividad consiste en desarrollar un radar de velocidad.

Podrá calcular la velocidad (cm/s) a la que se mueve un objeto.

En la primera versión, se utilizó una placa Arduino Uno.

En la segunda versión, se crea exclusivamente con el hardware mBot. También propone la conversión de cm/s a km/h, ya que es una unidad de medida más reconocible por los estudiantes.

## OBJETIVOS DIDÁCTICOS

- ❖ Aprendizaje del cálculo de velocidad relacionando distancia y tiempo.
- ❖ Aprender la conversión de unidades físicas de velocidad.
- ❖ Aprender a usar una pantalla de 7 segmentos.
- ❖ Aprender a usar un sensor de distancia.

## TECNOLOGÍA

- ❖ Desarrollar todo el sistema Arduino y programar el código.

## INGENIERÍA

- ❖ Crear una estructura para el radar.

## MATEMÁTICAS

- ❖ Calcular la velocidad.

Materia STEM:      Ciencias       Tecnología       Ingeniería       Matemáticas

Nivel educativo:      12-14 años       14-16 años

## PLANTEAMIENTO DEL PROBLEMA

El mBot actuará como un radar de velocidad. Detectará un vehículo que circula por delante y medirá su velocidad. Para ello, el mBot utilizará un sensor de distancia.



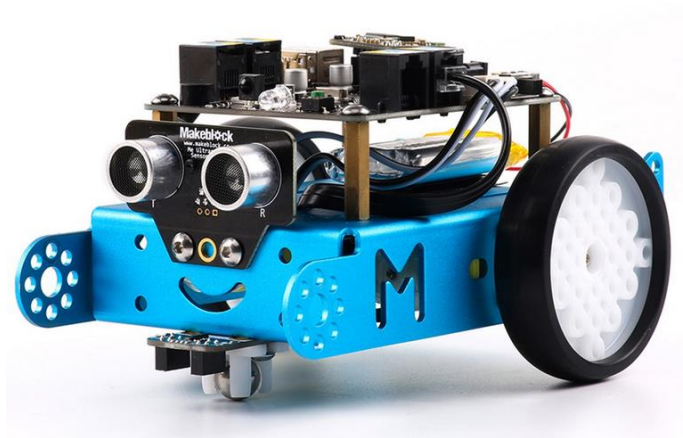
## LISTADO DE MATERIALES

### Primera versión (con Arduino):

1. mBlock Software.
2. Arduino IDE Software.
3. Protoboard.
4. Sensor de Ultrasonido.
5. Pantalla LCD.

### Segunda versión (sin Arduino):

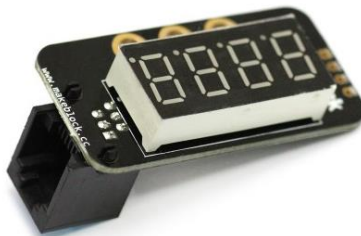
- mBot => Ref. 90054



### ❖ Me Sensor Ultrasonidos:



### ❖ Me Display 7 segmentos (4 dígitos. Rojo):



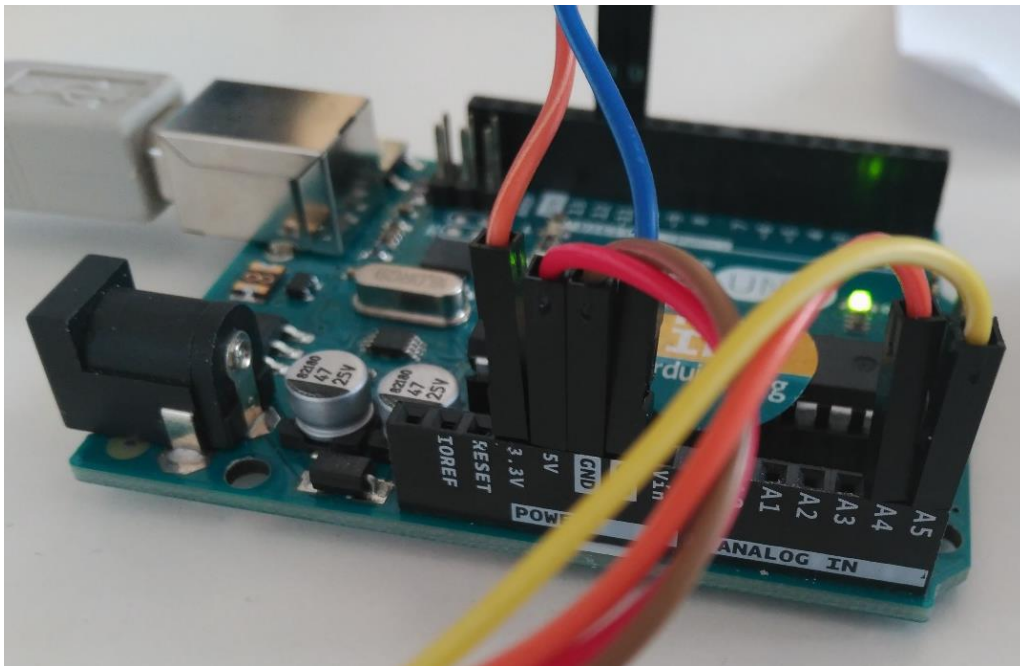
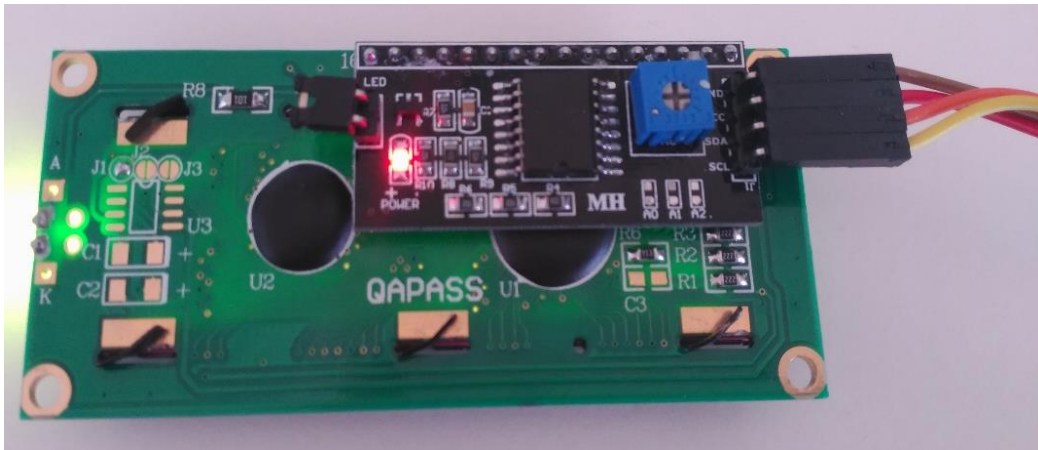
## DESCRIPCIÓN DE LA ACTIVIDAD

### Primera versión (con Arduino):

El radar de velocidad calculará la velocidad, en cm/s, a la que se mueve un objeto. En este caso será un coche de juguete. La velocidad se calculará a través del sensor de ultrasonidos de Arduino.

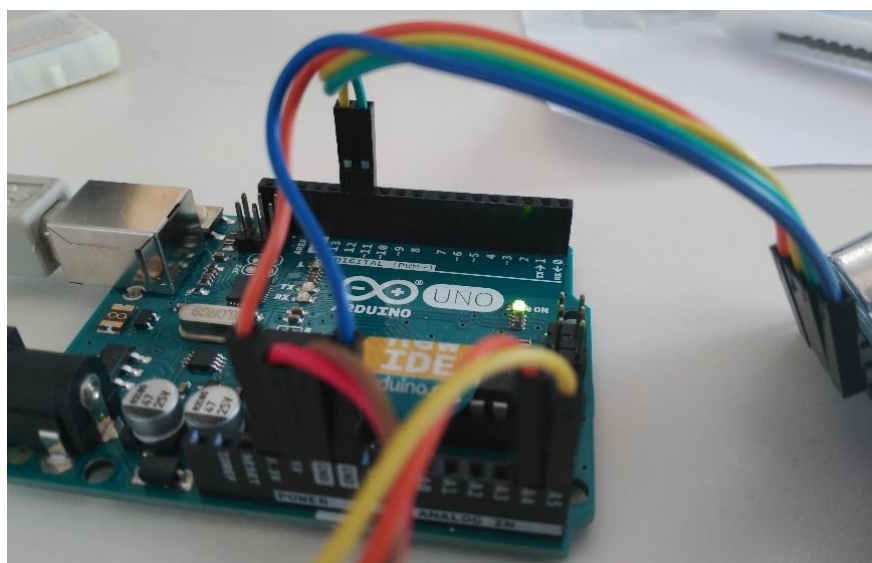
En primer lugar, creamos la estructura Arduino y conectamos los sensores.

1. Conectamos la pantalla LCD a la placa Arduino:

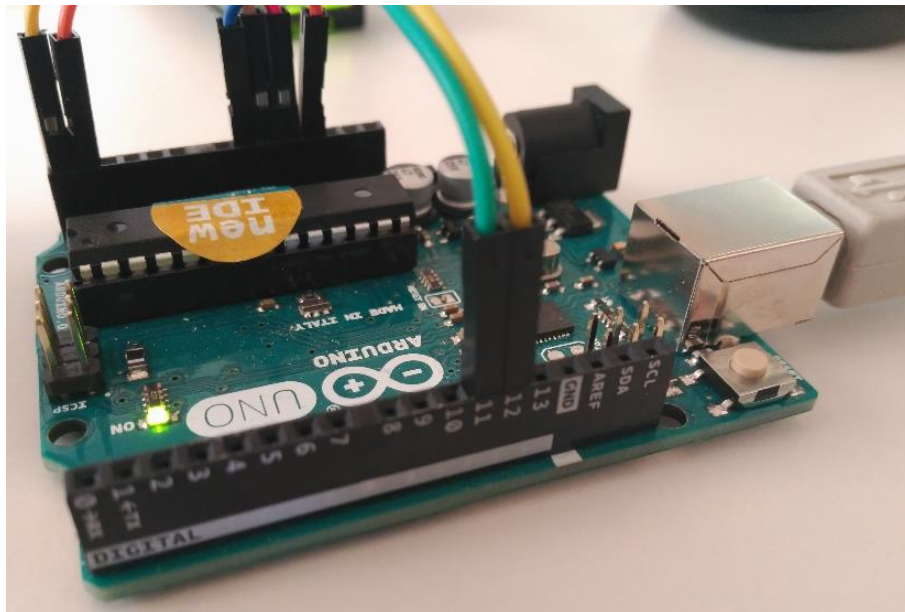


- ❖ **GND Pin** => [Cable Marrón] se conecta al puerto GND de la placa Arduino Uno.
- ❖ **VCC Pin** => [Cable Rojo] está conectado en el puerto 5V del Arduino Uno, o si tiene un protoboard, debe conectarse en la sección roja "+". Y con otro cable, conecte la sección roja "+" con el puerto de 5 V al Arduino Uno.
- ❖ **SDA Pin** => [Cable Naranja] está conectado en el puerto analógico 4 del Arduino Uno.
- ❖ **SCL Pin** => [Cable Amarillo] está conectado en el puerto analógico 5 del Arduino Uno.

## 2. Conecte el sensor de ultrasonidos al Arduino:



- ❖ **GND Pin** => [Cable Azul] se conecta al puerto GND de la placa Arduino Uno.
- ❖ **VCC Pin** => [Cable Naranja] está conectado en el puerto 5V del Arduino Uno, o si tiene un protoboard, debe conectarse en la sección roja "+". Y con otro cable, conecte la sección roja "+" con el puerto de 5 V al Arduino Uno.
- ❖ **Trig Pin** => [Cable Amarillo] está conectado en el puerto digital 12 del Arduino Uno.
- ❖ **SCL Pin** => [Cable Verde] está conectado en el puerto digital 11 del Arduino Uno.

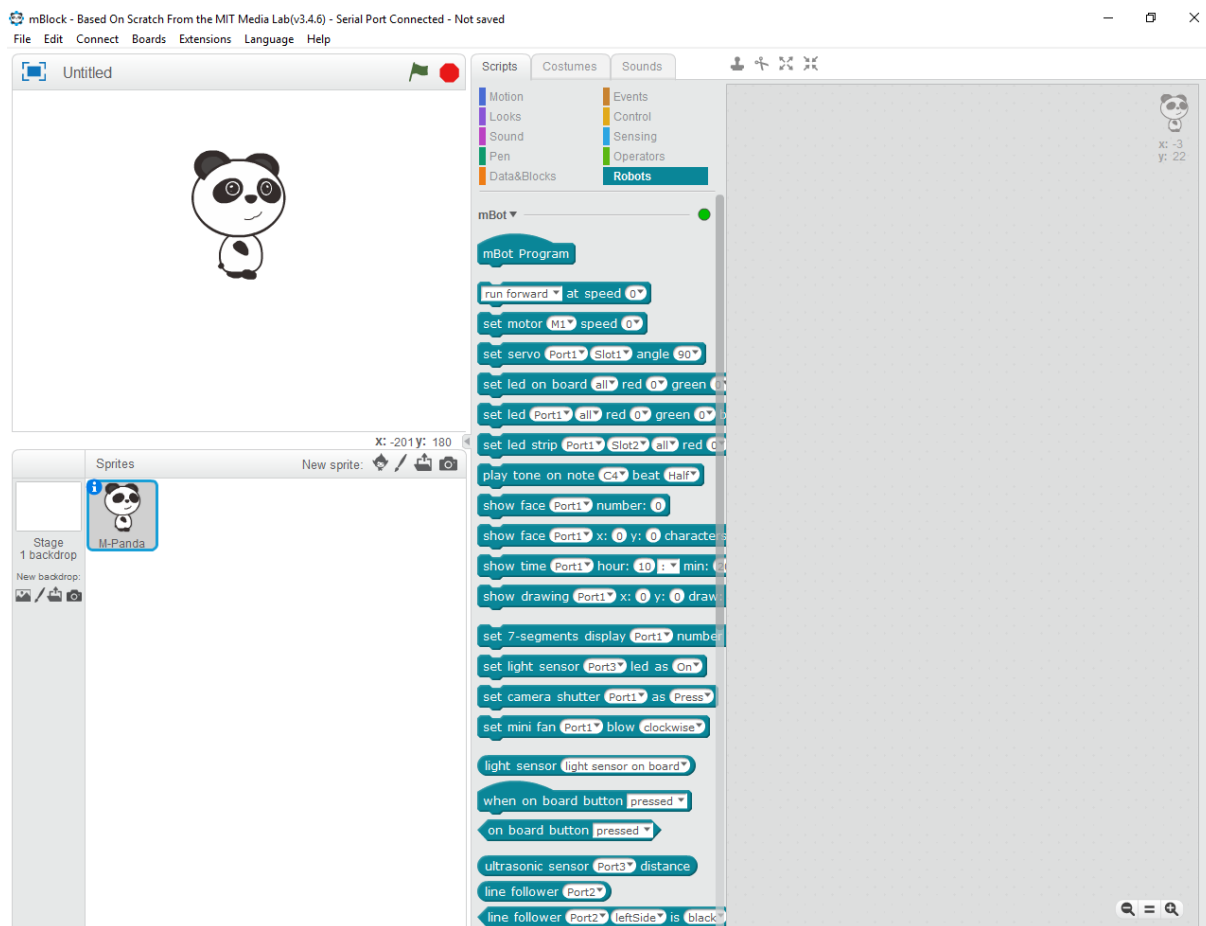


3. Creamos la estructura del radar:

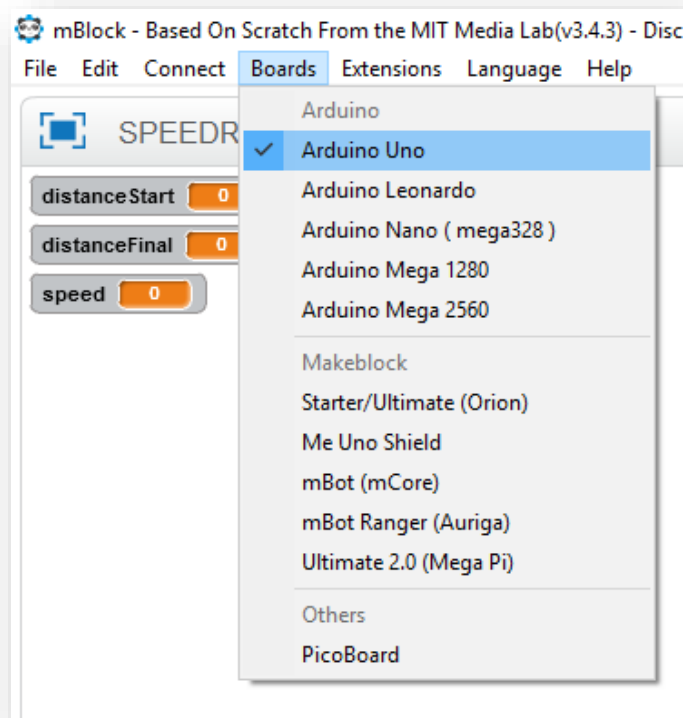


Ahora, desarrollaremos las instrucciones para el Arduino Uno en el software mBlock.

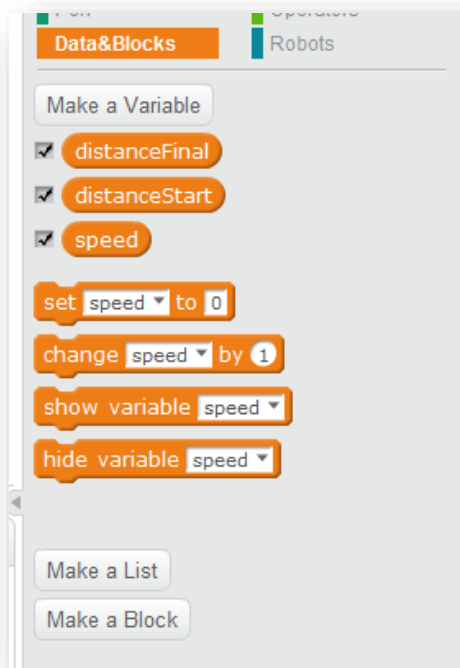
(<http://www.mblock.cc/download/>)



Primero, emparejamos la placa Arduino UNO con el software:



## 1. Creamos las variables:

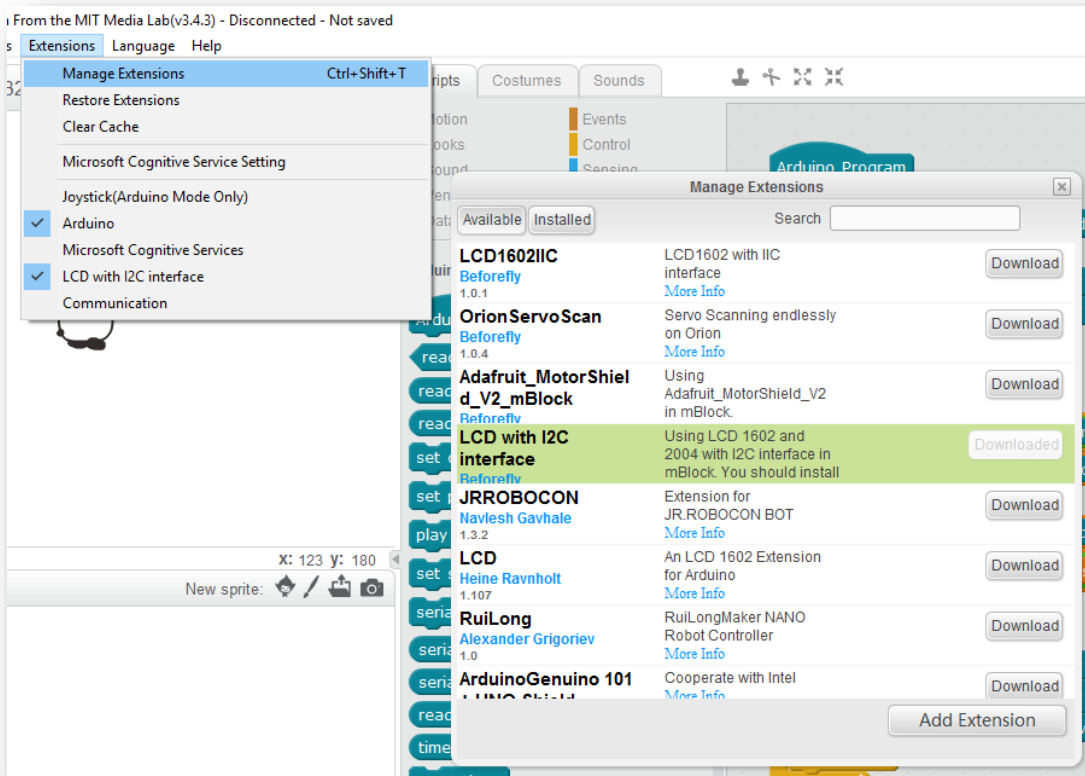


La variable “distanceStart” guarda el valor de la primera distancia que el sensor de ultrasonido lee.

La variable “distanceFinal” guarda el valor de la segunda distancia que el sensor de ultrasonido lee.

Y la variable “speed” será el resultado de calcular la velocidad.

## 2. Importamos la extensión “LCD with I2C Interface”:





3. Establecemos los parámetros para la pantalla LCD e iniciamos las variables:

```
Arduino Program
forever
  Set: LCD at 0x27 has 2 lines and 16 characters per line
  Clear LCD at 0x27
  LCD at 0x27 Line 1 Col 4 Show SPEED
  LCD at 0x27 Line 2 Col 4 Show RADAR
  set speed to 0
  set distanceStart to 0
  set distanceFinal to 0
```

4. Programamos las instrucciones para calcular la velocidad:

```
wait until read ultrasonic sensor trig pin 12 echo pin 11 < 100
set distanceStart to read ultrasonic sensor trig pin 12 echo pin 11
wait 0.1 secs
set distanceFinal to read ultrasonic sensor trig pin 12 echo pin 11
set speed to distanceStart - distanceFinal / 0.1
```

For calculate the speed, we rest the distances and we divided by time.  
The speed is calculate in cm/s

5. Finalmente, mostramos los resultados en la pantalla LCD:

```
Clear LCD at 0x27
LCD at 0x27 Line 1 Col 3 Show SPEED
LCD at 0x27 Line 2 Col 3 Show speed
LCD at 0x27 Line 2 Col 10 Show cm/seg
wait 3 secs
```

6. Este sería el código final del radar:

```
Arduino Program
forever
  Set: LCD at 0x27 has 2 lines and 16 characters per line
  Clear LCD at 0x27
  LCD at 0x27 Line 1 Col 4 Show SPEED
  LCD at 0x27 Line 2 Col 4 Show RADAR
  set speed to 0
  set distanceStart to 0
  set distanceFinal to 0
  wait until read ultrasonic sensor trig pin 12 echo pin 11 < 100
  set distanceStart to read ultrasonic sensor trig pin 12 echo pin 11
  wait 0.1 secs
  set distanceFinal to read ultrasonic sensor trig pin 12 echo pin 11
  set speed to (distanceStart - distanceFinal) / 0.1
  wait 0.5 secs
  Clear LCD at 0x27
  LCD at 0x27 Line 1 Col 3 Show SPEED
  LCD at 0x27 Line 2 Col 3 Show speed
  LCD at 0x27 Line 2 Col 10 Show cm/seg
  wait 3 secs
```

For calculate the speed, we rest the distances and we divided by time.

The speed is calculate in cm/s

7. Para subir el código a la placa Arduino Uno, hacemos click en “Arduino Program” con el botón izquierdo del ratón y se mostrará otra ventana. En esta nueva ventana, hacemos click en “Upload to Arduino”:

```
Arduino Program
forever
  Set: LCD at 0x27 has 2 lines and 16 characters per line
  Clear LCD at 0x27
  LCD at 0x27 Line 1 Col 4 Show SPEED
  LCD at 0x27 Line 2 Col 4 Show RADAR
  set speed to 0
  set distanceStart to 0
  set distanceFinal to 0
  wait until read ultrasonic sensor trig pin 12 echo pin 11 < 100
  set distanceStart to read ultrasonic sensor trig pin 12 echo pin 11
  wait 0.1 secs
  set distanceFinal to read ultrasonic sensor trig pin 12 echo pin 11
  set speed to (distanceStart - distanceFinal) / 0.1
  wait 0.5 secs
  Clear LCD at 0x27
  LCD at 0x27 Line 1 Col 3 Show SPEED
  LCD at 0x27 Line 2 Col 3 Show speed
  LCD at 0x27 Line 2 Col 10 Show cm/seg
  wait 3 secs
```

```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 #include <LiquidCrystal_I2C.h>
6
7 double angle_rad = PI/180.0;
8 double angle_deg = 180.0/PI;
9 double speed;
10 double distanceStart;
11 double distanceFinal;
12 LiquidCrystal_I2C lcd_I2C_0x27(0x27, 2, 1, 0, 4, 5, 6, 7, 3,
13 float getDistance(int trig,int echo) {
14   pinMode(trig,OUTPUT);
15   digitalWrite(trig,LOW);
16   delayMicroseconds(2);
17   digitalWrite(trig,HIGH);
18   delayMicroseconds(10);
```

Back Upload to Arduino Edit with Arduino IDE

send encode mode binary mode char mode

recv encode mode binary mode char mode

Send

A continuación, mostramos algunas imágenes del radar de velocidad:





## Segunda versión (sin Arduino):

El trabajo de los estudiantes es conocer la fórmula matemática que relaciona la velocidad, el espacio y el tiempo y con la ayuda del profesor, crear el diagrama de flujo y desarrollar el código que permite que el mBot funcione como un radar de velocidad.

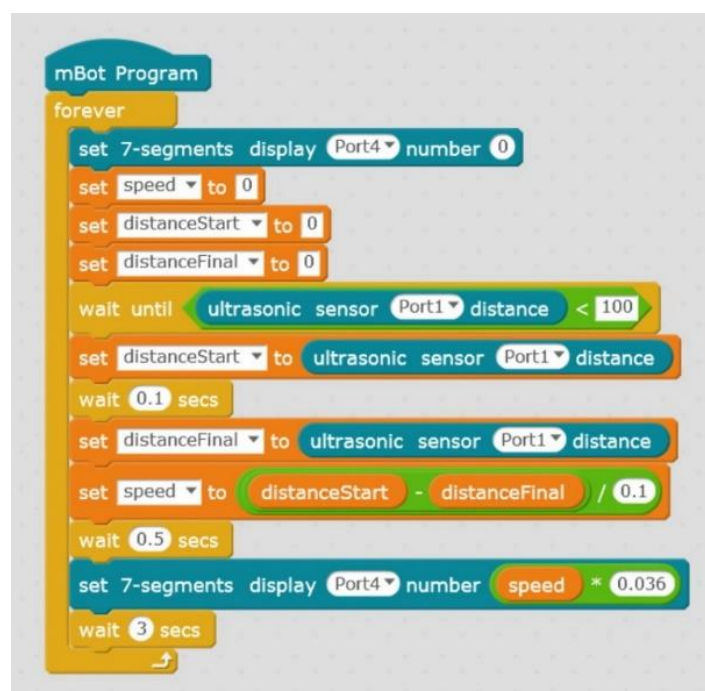
La variable "espacio" se obtendrá por la diferencia entre dos distancias medidas. La variable "tiempo", será el intervalo de tiempo entre esas dos mediciones.

El radar de velocidad permanecerá en modo de espera hasta que pase un "vehículo" delante de él. En ese momento, calculará la velocidad del vehículo y lo mostrará en la pantalla de 7 segmentos. (Se calculará en km/h).

## Código de la actividad:

El código es muy similar al código de la primera versión. En este caso, se ha realizado una traducción de los comandos de Arduino a comandos de scratch y hemos agregado la operación para convertir de cm/s a km/h.

El código es muy corto y simple, pero potente.



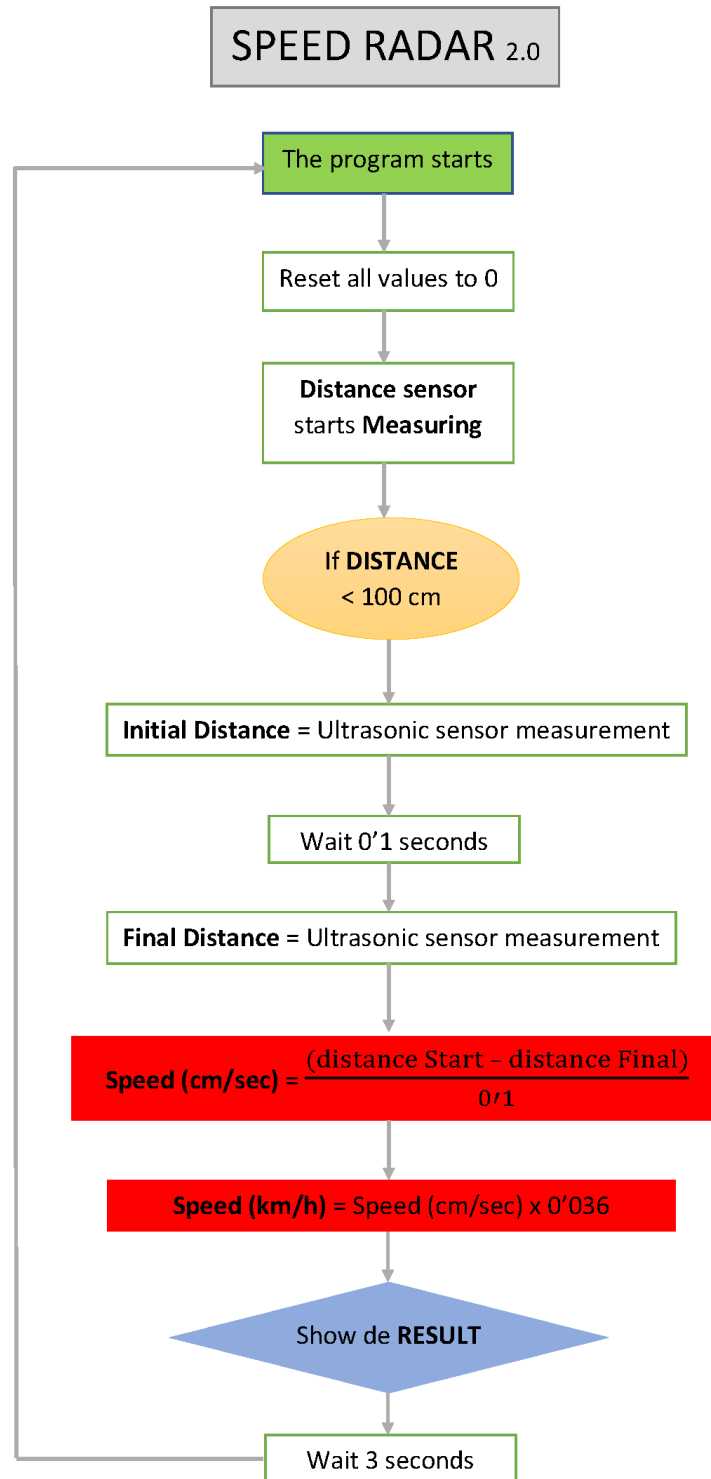
```
mBot Program
forever
  set 7-segments display Port4 number 0
  set speed to 0
  set distanceStart to 0
  set distanceFinal to 0
  wait until ultrasonic sensor Port1 distance < 100
  set distanceStart to ultrasonic sensor Port1 distance
  wait 0.1 secs
  set distanceFinal to ultrasonic sensor Port1 distance
  set speed to (distanceStart - distanceFinal) / 0.1
  wait 0.5 secs
  set 7-segments display Port4 number (speed * 0.036)
  wait 3 secs
```

Grabaremos el código en la placa Arduino del mBot. De esta manera, el mBot funcionará independientemente de la computadora.

Composición de la estructura:



## DIAGRAMA DE FLUJO



## ESCALABILIDAD

Esta actividad es complicada porque puede haber problemas con las bibliotecas y sus versiones. Además, el sensor de ultrasonido es muy sensible y difícil de configurar.

Cada vez que lo mueves, debes configurarlo para que sea lo más preciso posible.