

# Robotics 1d

---

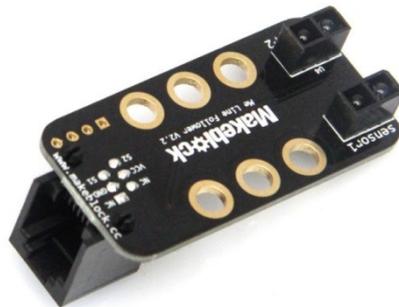
## *mBot Sensors*

A robot will have a tough time if it cannot sense the world around it. The mBot has a number of sensors already onboard which help it respond to things it might encounter as it moves around.

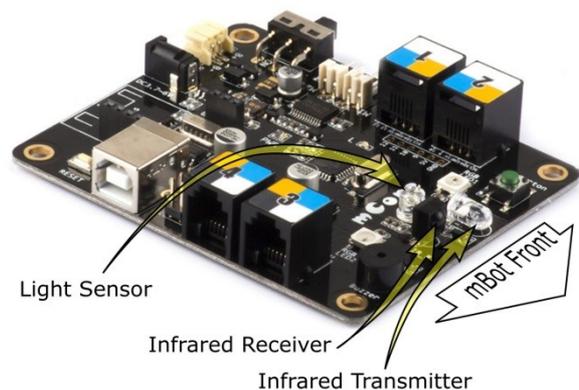
- Ultrasonic Sensor



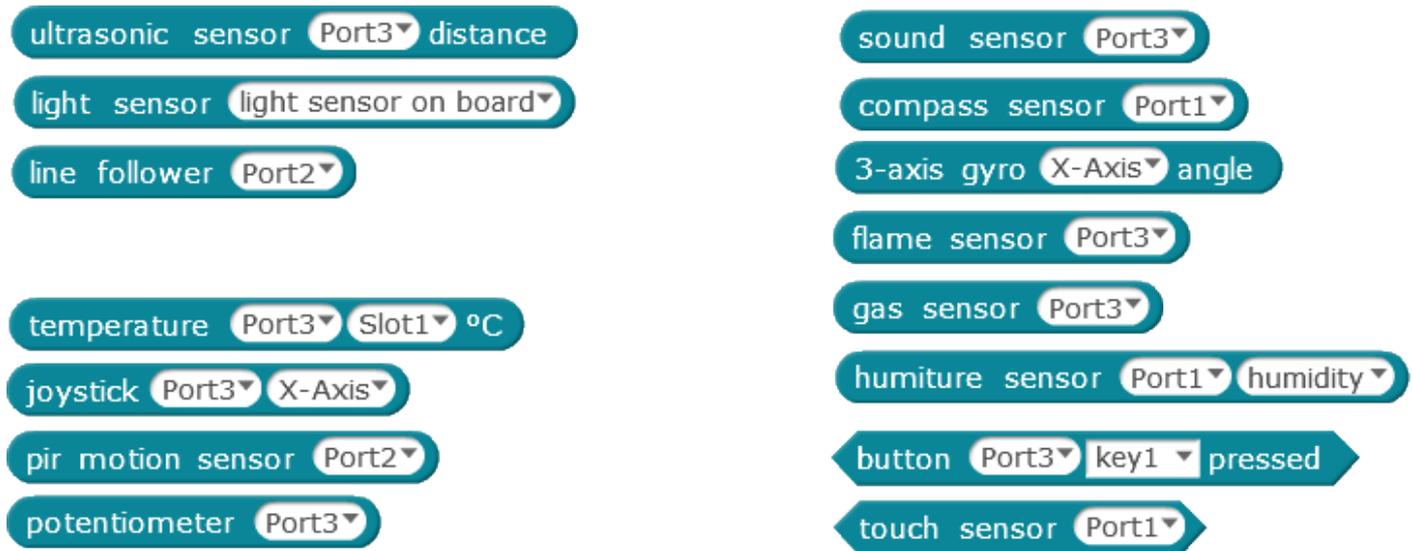
- Line Following Sensor



- Light Sensors
- Infrared Receiver (Communication between mBots)
- Infrared Transmitter (Communication between mBots)



In the mBlock interface there are code blocks that match each type of sensor in the standard mBot kit as well as extra sensors and devices that can be added.



You will notice that these blocks are all designed to fit where you would put a number in the *Scratch operators* and *control* blocks. This means they can be used directly to control program loops, motor speeds, and anywhere else where you might use a number.

## Test the Ultrasonic Sensor

We're going to focus on the Ultrasonic range finder.

### How does it work?

The ultrasonic range finder works by sending out pulses of sound at a very high frequency. The sound bounces off objects and returns to the sensor as an echo. The sensor measures the time between sending the pulse and hearing the echo. Because the sensor knows the speed of sound is 340m/s it can calculate the distance to the object that a sound wave reflected off and report that to the mBot. It is using echolocation in much the same way a bat does.

The sound emitted by the sensor is 40kHz. This is too high for us to hear. Our hearing range is typically between 20Hz and 20kHz. A dog could hear it because a dog can hear up to 45kHz. A bat could also hear it because they can hear up to 200kHz. Would it confuse a bat?

The **Manufacturers Specifications** state that the sensor can do the following;

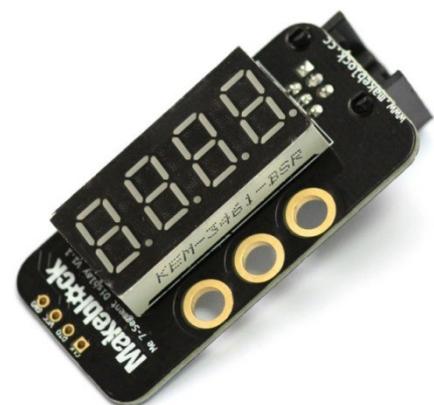
- It has a measurement resolution of 10mm.
- It has an angle over which it can detect things of 15°
- And it can measure distances between 2cm and 4m

The reading from the sensor is in centimetres.

### Setting up a test

To test it we will make use of an extra device we can add to our mBot. We have some nifty 4-digit displays. If you don't have a 4-digit display, use the onboard LEDs to light up if they detect something at a particular distance.

We'll connect the 4-digit display to **Port 1** on the mBot.



Amongst the mBlock code blocks we can find the following block to interact with the 4-digit display.

```
set 7-segments display Port1 number 100
```

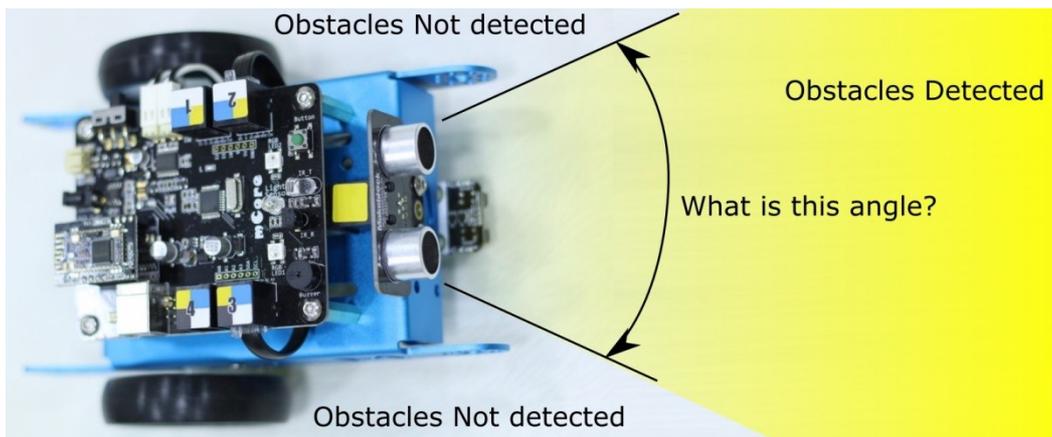
Create a simple program to read the Ultrasonic sensor and display the distance it is measuring. Upload this and see that it responds to your hand in front of it. Note: We have created the variable called **Distance** for holding the reading from the sensor. You can put the distance sensor block directly into the 7-segment display block if you wish. To do this, just drag and drop the Ultrasonic sensor block into the place occupied by the number “100” in the image above.

```
mBot Program
forever
  set Distance to ultrasonic sensor Port3 distance
  set 7-segments display Port1 number Distance
  wait 1 secs
```

### Let's test its sensitivity.

Here are some simple tests to try with this arrangement. The 4-digit display will show the distance in cm to the object the mBot is detecting.

- How far away and how close does it detect distances? You may need to point it at a wall to measure longer distances. How does that match the manufacturer's specifications?
- What is the *measurement resolution*? Measure how much the sensor reading changes if you move something towards it by 10cm, 20mm, 10mm, and 5mm?
- How big does the object need to be before it can be detected?
- Are there shapes it has trouble with?
- What is the angle of view over which the mBot can detect things?

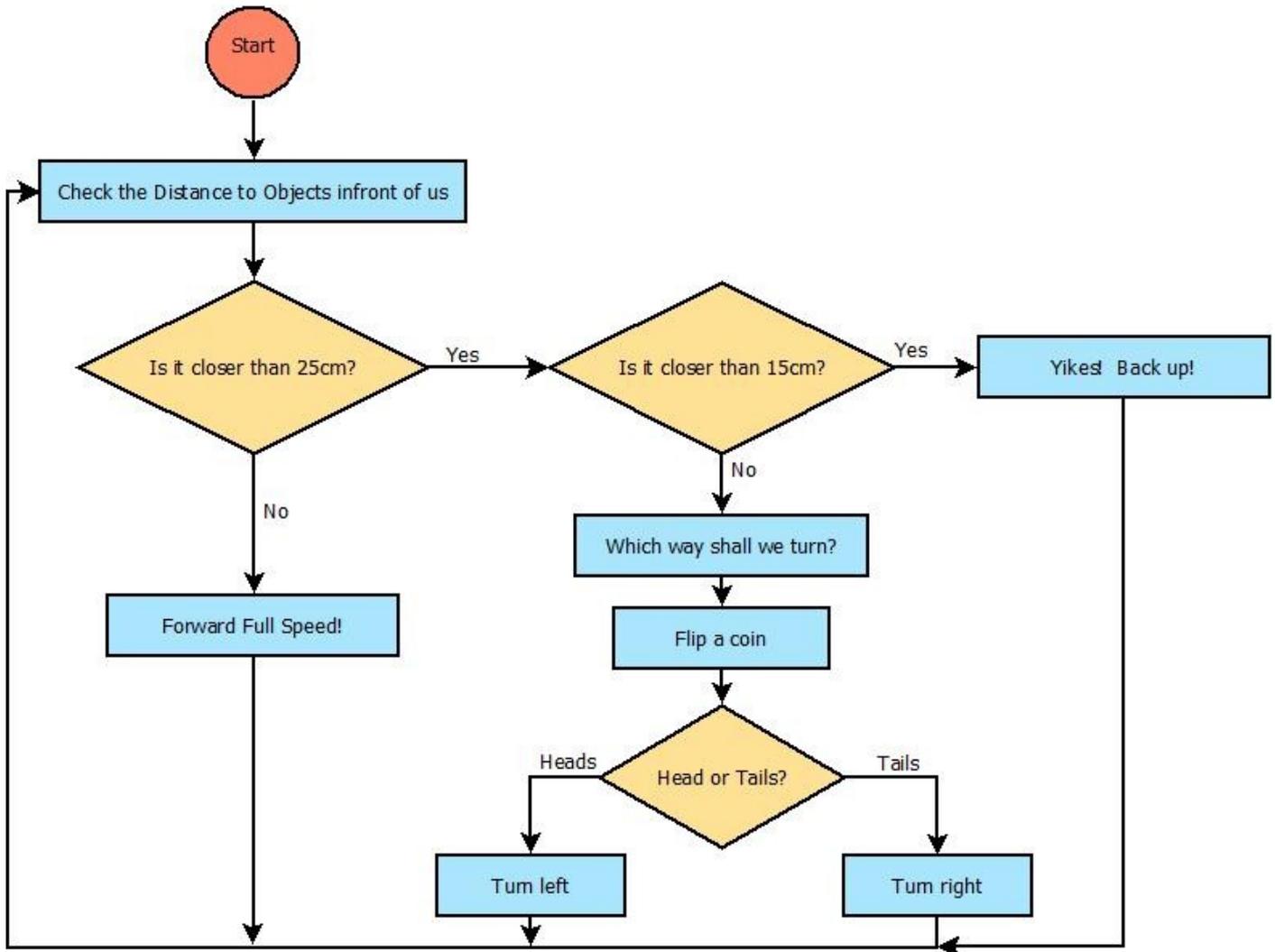


Do we know more about the limitations of this sensor?

## Obstacle Avoidance

Let's recreate the obstacle avoidance behaviour of the default program. The program requires the following measurements and decisions to be made by the mBot.

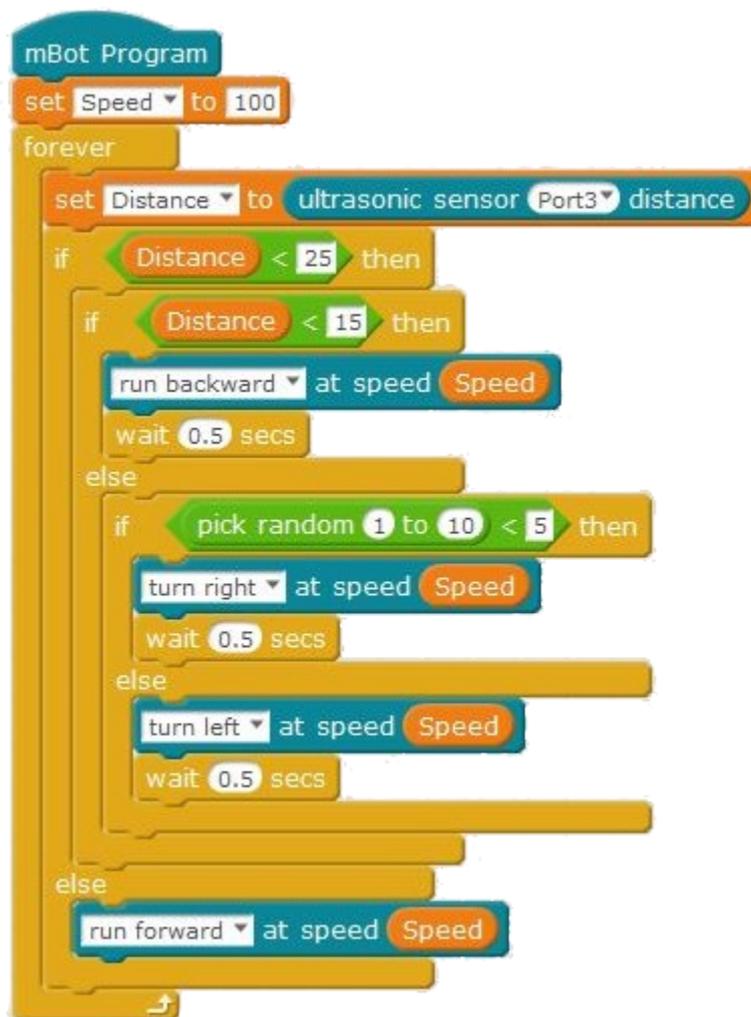
### Logic for Obstacle Avoidance



Have a go at using these decisions yourself as you walk around the room.

Now try to build an mBot sketch that will do the same. Remember to add in the **Wait** command to ensure you have sufficient time for the motors to move enough.

Here is one solution to the problem.



## Your Challenge

Is your mBot afraid of the dark? No? Well it - should be. There's monsters out there.

Use the on-board light sensor to make your mBot speed up to get out of dark places quickly.

## We're done for this Session

Next time. We'll look at the on-board line-following sensor.

## Challenge Sketch

Here is a suggested sketch to adapt the sketch above to make it run faster if the light levels are low.

```
mBot Program
forever
  set Scared to light sensor light sensor on board
  if Scared < 120 then
    set Speed to 254
  else
    set Speed to 80
  set Distance to ultrasonic sensor Port3 distance
  if Distance < 25 then
    if Distance < 15 then
      run backward at speed Speed
      wait 0.5 secs
    else
      if pick random 1 to 10 < 5 then
        turn right at speed Speed
        wait 0.5 secs
      else
        turn left at speed Speed
        wait 0.5 secs
    else
      run forward at speed Speed
```



This document "Robotics1d – mBot Sensors" by Hamish Trolove is provided under a creative commons license - Attribution, Share Alike.

<http://creativecommons.org/licenses/by-sa/4.0/>

[www.techmonkeybusiness.com](http://www.techmonkeybusiness.com)

