Ultrasonic Sensor with Makecode Mindstorms for LEGO EV3

This is the fifth in a series of lessons that are designed to prepare students for Robotics competitions such as RoboCup and First Lego League. The target is students in year 5-8.

1 Lesson

References: [https://makecode.mindstorms.com/blocks/logic/if](https://makecode.mindstorms.com/blocks/logic/if)

<table>
<thead>
<tr>
<th>Design</th>
<th>Digital</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Image 225x613 to 371x662]</td>
<td>[Image 40x551 to 94x604]</td>
</tr>
<tr>
<td>[Image 40x468 to 94x521]</td>
<td>[Image 226x286 to 370x335]</td>
</tr>
<tr>
<td>[Image 40x394 to 94x447]</td>
<td>[Image 36x789]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design</th>
<th>Digital</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5-6 explanation</strong> of how the features of technologies impact on designed solutions for each of the prescribed technologies contexts</td>
<td><strong>5-6 definition</strong> of problems in terms of data and functional requirements</td>
</tr>
<tr>
<td><strong>7-8 explanation</strong> of how the features of technologies impact on designed solutions and influence design decisions for each of the prescribed technologies contexts</td>
<td><strong>7-8 definition and decomposition</strong> of problems in terms of functional requirements and constraints</td>
</tr>
<tr>
<td><strong>5-6 production</strong> of designed solutions by selecting and using appropriate technologies and techniques correctly and safely</td>
<td><strong>5-6 implementation</strong> of digital solutions, including a visual program</td>
</tr>
<tr>
<td><strong>7-8 production</strong> of effective designed solutions for the intended purpose independently and safely</td>
<td><strong>7-8 testing, modification and implementation</strong> of digital solutions</td>
</tr>
<tr>
<td><strong>5-6 communication</strong> of design ideas to audiences using: • graphical representation techniques • technical terms</td>
<td><strong>5-6 design</strong> of solutions by: • developing algorithms to address defined problems  • incorporating decision making, repetition (iteration) and user interface design</td>
</tr>
<tr>
<td><strong>7-8 communication</strong> to different audiences using: • appropriate technical terms • a range of technologies and graphical representation techniques</td>
<td><strong>7-8 design</strong> of user experiences and algorithms incorporating branching and iterations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design</th>
<th>Digital</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Image 36x774]</td>
<td>[Image 36x759]</td>
</tr>
<tr>
<td>[Image 36x712]</td>
<td>[Image 192x675]</td>
</tr>
<tr>
<td>[Image 36x675]</td>
<td>[Image 422x675]</td>
</tr>
</tbody>
</table>

- **Explain how to use the ultrasonic sensor**
- **Stop when an object is detected**
- **Communicate solution**
- **Write the Algorithm to Stop when an object is detected**
The Ultrasonic Sensor can measure the distance to an object in front of it. It does this by sending out sound waves and measuring how long it takes the sound to reflect back to the sensor. The sound frequency is too high for you to hear ("ultrasonic").

You can measure the distance to an object in centimetres. You could use this, for example, to make your robot stop a certain distance from a wall.

The Ultrasonic Sensor can give the following data:

<table>
<thead>
<tr>
<th>Data</th>
<th>Type</th>
<th>Range</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance in Centimetres</td>
<td>Numeric</td>
<td>0 to 255</td>
<td>Distance to object in centimetres.</td>
</tr>
<tr>
<td>Distance in Inches</td>
<td>Numeric</td>
<td>0 to 100</td>
<td>Distance to object in inches.</td>
</tr>
<tr>
<td>Ultrasound Detected</td>
<td>Logic</td>
<td>True/False</td>
<td>True if another ultrasonic sensor is detected.</td>
</tr>
</tbody>
</table>

**distance**

The distance of an object detected by the ultrasonic sensor.

```javascript
ultrasonic 4 \ distance

distance(): number;
```

The distance value returned is the number of centimeters to the object that the sensor is currently detecting.

**Returns**

- a number that is the distance of the object detected by the ultrasonic sensor in centimeters.
Instructions
Predict and explain what these blocks do and how it works [Add this to your Report]

Video link
https://youtu.be/8rPftn916YQ

Predict and explain what these blocks do and how it works [Add this to your Report]
Video link
https://youtu.be/8rPftn916YQ

Algorithm

Forever
Go straight
If ultrasonic sensor detects object within 10 cm
  Stop
Elseif ultrasonic sensor detects object within 20 cm
  Go slow

Recreate the program shown, then download and run to test.

Challenge 1

Try this alternative using Events
Doesn't work for 20cm though...

Test your code. If it doesn’t work as expected, alter the parameters and try again until it does.

Challenge 2

Cause your robot to do the following for 10 seconds:
· Whenever there is an object that is less than 20 cm from the robot’s ultrasonic sensor, the robot should move backwards, away from the object.
· Otherwise, the robot should move forwards
After 10 seconds have elapsed, your robot should stop moving.
You can view a video of a working example of this program at http://youtu.be/TQycE0lbSiw

Test your code. If it doesn’t work as expected, alter the parameters and try again until it does.

Notes and hints:
- If there is nothing in front of the robot, your robot should just drive forward smoothly for 10 seconds. If your robot does not drive smoothly in this situation, that’s probably because you have set a fixed duration (in terms of either time or motor rotations) on your move blocks. Consider whether that is really necessary.
- If there is a wall more than 20 cm in front of your robot, your robot should drive forward smoothly until it is roughly 20 cm away from the wall, and then move jerkily forward and backward – when it recognizes the distance is less than 20 cm it moves forward, but as soon as that is no longer the case it retreats.
- The ultrasonic sensor works best with large flat objects that have a matt surface like a piece of cardboard or a pad of paper.
- You may find it easier to do this task if you reduce the power to the motors on your move blocks.
- During the 10 second period, the robot should never stop moving. It should always be moving either forwards or backwards.
Detect objects and react

On the road, when a driver sees an object, they slow their car down before coming to a full stop.

Write a program with the same behaviour:

If the Ultrasonic sensor:

1. Detects an object less than 10 cm away, make the robot stop
2. Detects an object between 10 and 20 cm away, make the robot slow down
3. Does not detect any object, continue to move at full speed.

Test your code. If it doesn’t work as expected, alter the parameters and try again until it does.

Hints
Extension Challenge 2

Challenge 1: Simple Wall Follow

**Challenge:** Can you write a program to have a robot follow the wall (always staying 15cm away from the wall) using an ultrasonic sensor?

**STEP 1:** In a Switch Block, turn Left or Right based on whether the robot is too close to the wall or too far from the wall.

**STEP 2:** Repeat everything in a loop that runs forever (you can change the exit condition of the loop if you wish).

Play the video to see how the robot should move.
Challenge 2: Optimizing the Code

The code for the Simple Ultrasonic Wall Follow Challenge was slow and the robot wiggles back and forth a lot.

**Challenge:** For this next challenge, think about how you can improve the program so that the wall follower is smoother.

**Hint:** Change the angle of the turns

Play the video to see how the robot should move

Do you notice any differences?
Challenge: Right-hand rule path

Write a program to line follow your way through the following maze.

Test your code. If it doesn’t work as expected, alter the parameters and try again until it does.