MANE 4490/ECSE 4090 MECHATRONICS
Ultrasonic Distance Sensor

Part 1: Basic Sensor Setup and MATLAB Usage

Objective

- Install the MATLAB files for the HC-SR04, connect the sensor module, and test the basic functionality of the distance sensor.

Background Information

The HC-SR04 and other ultrasonic modules operate using the principles of sonar. An acoustic signal is emitted by the unit, while a microphone listens for the returning echo. The distance is calculated by comparing the echo return time to the known speed of sound in the medium (air).

This sonic range-finding is the mechanism by which dolphins, bats, and submarines navigate. By increasing the number of receiving sensors, an object's precise location can be triangulated.

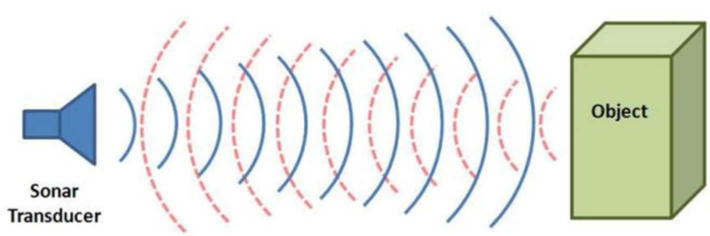


Image courtesy PICAXE Hobbyzine

The HC-SR04 operates by sending out a pulse at 40 kHz, nearly twice the threshold of human hearing [although if you hold the sensor close to your ear while it is operating, you may hear faint pulses]. One of the transducers is dedicated to emitting the pulse, and the other to receiving the echo. The 'Trig' and 'Echo' pins also independently handle data I/O for the respective triggering and receiving transducers.

Figure 1:HC-SR04 Module and pins

The data capture and calculation used to determine distance is handled by a separate microcontroller. You can also view the base code by opening the file “NewPing.cpp”.

Sensor Hardware

The HC-SR04 sensor can be used in any breadboard for prototyping. MinSegMega V2 can be wired using pins 7 for the trigger and 8 for the echo.

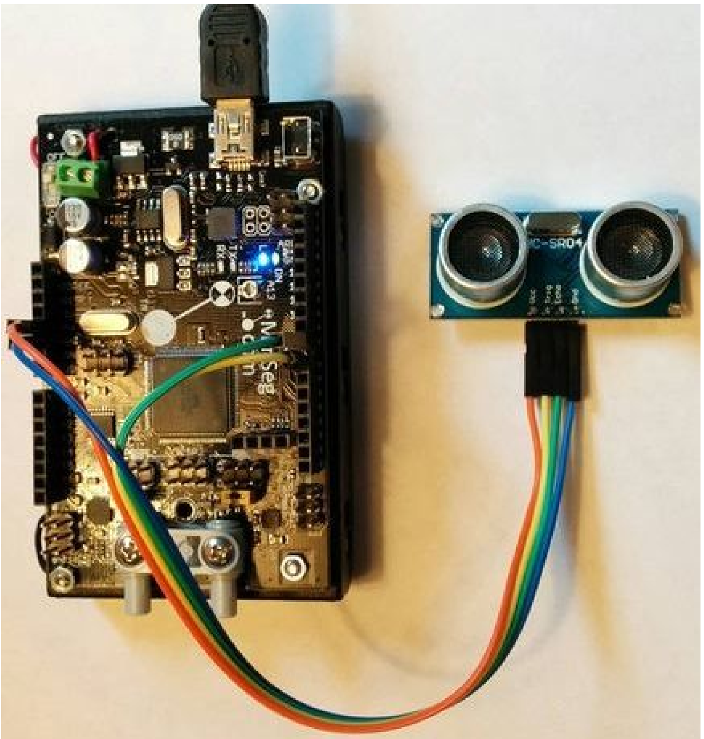


Figure 2: Wired to 7 and 8 for MinSegMega

The MinSegShield boards can use the pre-wired header socket.

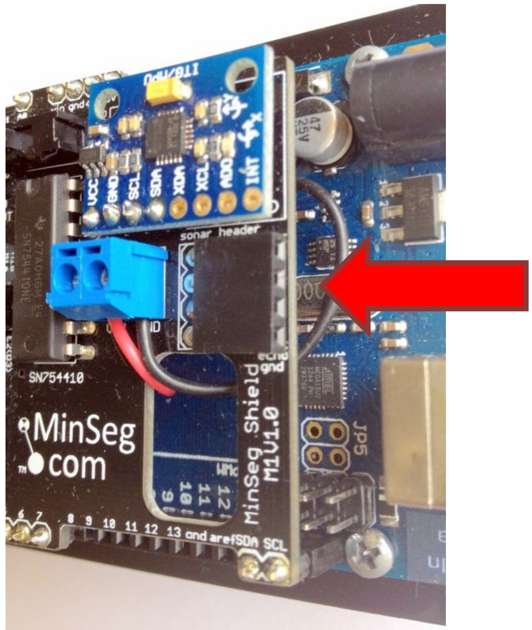


Figure 3 Sensor header on MinSegShield

Insert the ultrasonic sensor with the transducers face up.

MATLAB files

- Build the following Simulink diagram from the files provided or from the RASPLib files:

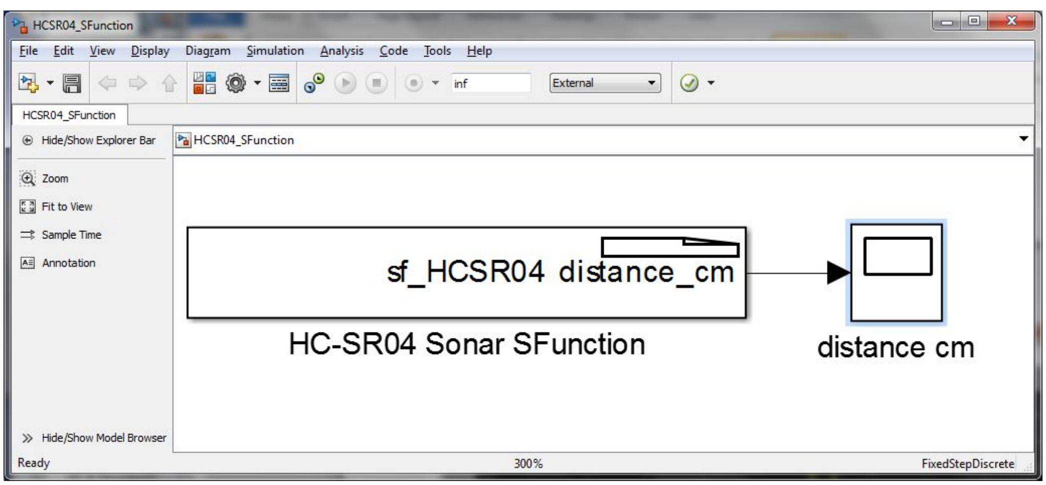


Figure 4 General Distance sensor diagram

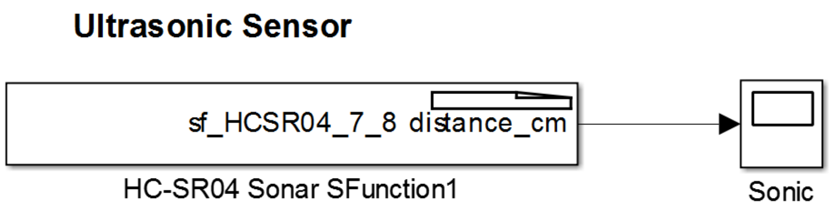


Figure 5 Distance sensor diagram for MinSegShield or MinSegMega

Configure the model parameters as follows:

- Under ‘Run on Target Hardware’, ensure that the “Enable External mode” box is checked.
- Under ‘Solver’, make sure the model is set to a fixed step size of .03, and a stop time of ‘inf’.
- Run the simulation, and double-click scope to observe the results.

Question: Try placing your hand over the sensor. How high can you raise it before you encounter an overly noisy signal? Now try using a textbook or folder. Do you notice a difference in the reading? Finally, try a piece of soft fabric. How might you explain your observations?

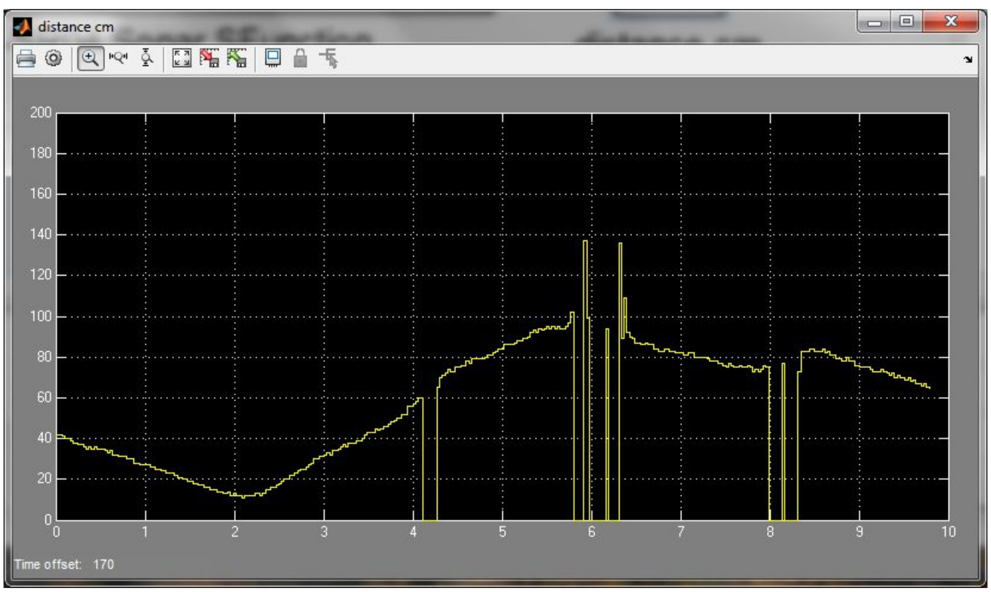


Figure 6 Sample distance reading with noise spikes

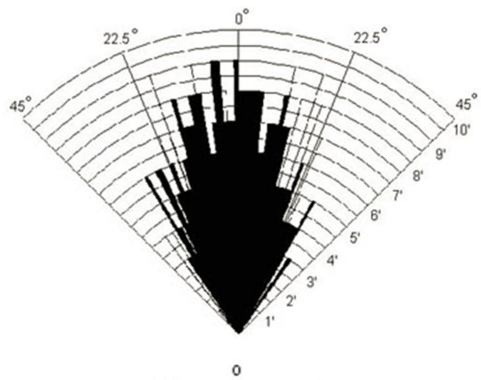
Part 2: Sensor Detection limits

Objective

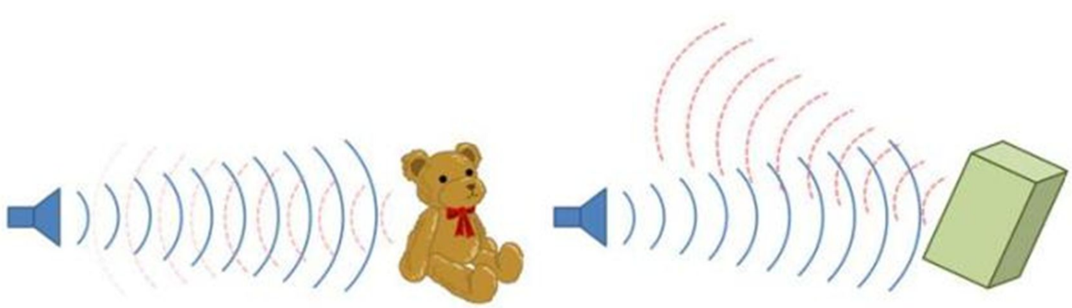
- Determine the distance limits for reasonable sensor readings
- Measure the edge of the detection area at different heights
- Explore the effects of various object shapes and materials

Background Information

You will notice that the detection region of the sensor expands as an object moves away from the sensor. If an object is close to the module, but off to a side, the emitted signal will not bounce off the object. The object is essentially in a ‘blind spot’. Sample readings can indicate a sensor’s “detection cone”, as in the example plot below.



You may also have noticed a difference in readings due to various object shapes, sizes, and materials. These affect the reflection of the sound waves, and the reading obtained by the sensor. The amount of noise in the signal response will be dependent on the object and the distance away from the distance sensor.



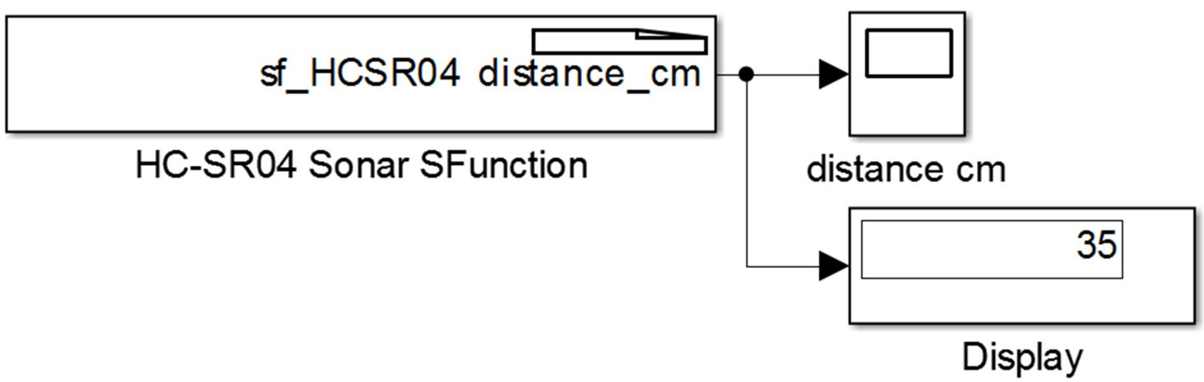
Some materials absorb more of the acoustic signal, and the shape or position may reflect waves away from sensor

(*Image courtesy PICAXE Hobbyzine*)

Height detection accuracy

- Place the MinSeg unit on a flat table in front of you. Find a large flat rigid object, such as a piece of cardboard or plastic. Also, find a ruler or measuring stick with metric denominations.
- Run the Simulink model, and hold the object above the sensor. Take a distance reading using the scope.
- Using the measuring stick, measure the actual distance of the object above the sensor. Be sure to subtract any distance which the sensor sits above your measurement baseline (floor, table, etc.).

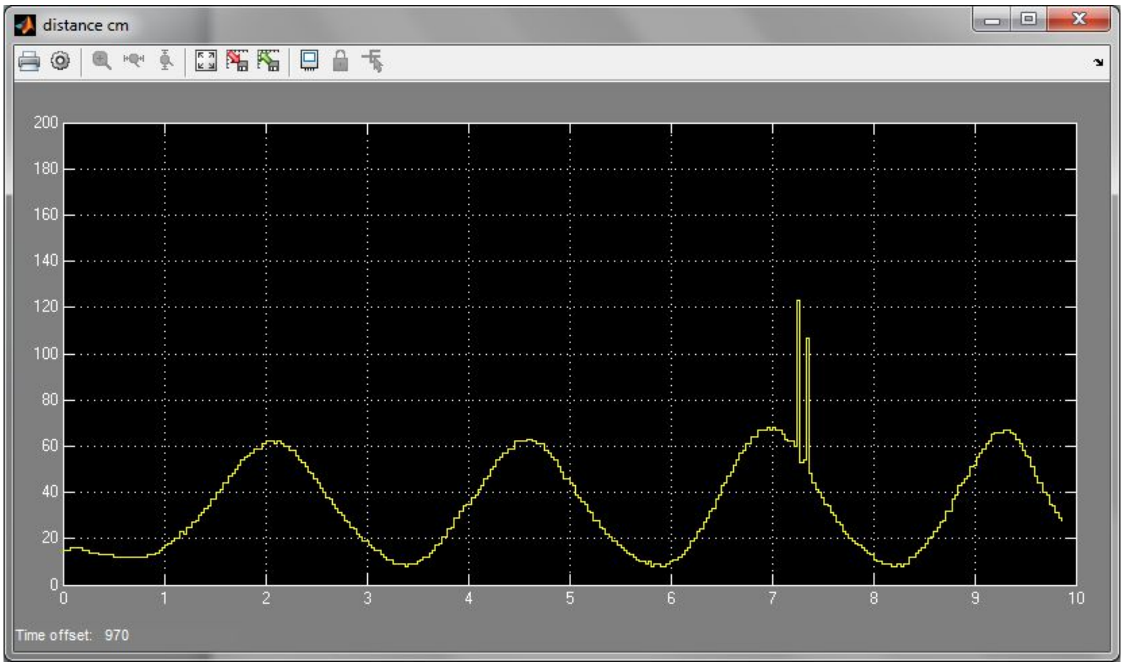
Note: You can edit the model to include a 'Display' element that will show the current reading value, as seen below. This block is found under Simulink Library -> 'Sinks'.



Question: How do the distance readings on the scope compare to those you measured using the ruler? What is an approximate lower limit for accurate detection distance?

Height Detection Upper Limit

- Gather several sample objects to test. Try to find a flat object such as a book, a cylindrical object like a water bottle, a fabric material such as a fleece or soft towel, and some aluminum foil.
- Run the distance sensor code, and bring up the scope. Hold the flat object a few centimeters above the sensor, and raise the object about 40-50cm. Lower the object again, and repeat this cycle a few times (try to create a sinusoidal pattern).
- Before the scope screen clears and repeats, take a screenshot of the window, as seen below. Copy the image into a document, and record which object that created the data.



- Repeat the previous steps, increasing the maximum height to which you raise the object. Fewer cycles are okay.
- Repeat this test using the various objects. Wrap the flat object in the soft fabric for one trial, and slightly crumple the aluminum foil into irregular object for another trial. Copy and record the scope data.

Checkpoint: Compare the responses of each material at the different heights

Edge Detection

Note: For this exercise, accept the distance reading (the height) in order to simplify measurement.

- With the MinSeg flat on a table or floor, place a metric ruler or measuring stick underneath the hardware unit. Line up the ruler so that the center measurement is lined up with the center of the ultrasonic unit.
- Select an object for testing. Run the Simulink model, and open the scope.
- Hold your selected object above the sensor, and raise or lower it until it the scope reading indicates a chosen height. Now, move the object to the side of the sensor (in line with the transducers) until the object is no longer detected. Slowly move the object back until it is just detected, and read the approximate position of the object's edge against the ruler. Record this value, along with the height measurement.

[**Note**: For additional accuracy, you can use a thin string with a weighted object at the end to create a makeshift plumb bob. Hang this from the edge of the object for easier readings.]

- Repeat when approaching the sensor from the opposite side. Make sure you are always recording the position of the object edge closest to the sensor.

Checkpoint: Repeat this procedure for several heights (at least four), and then create a graph of your findings. You can generate such a plot using MATLAB: Input the X data points as a matrix, and their corresponding height values in another matrix. Plot the data using the following code snippet.

```
%Example data plotting using MATLAB

x = [-2 2 -4 4 -6 6 -8 8];
y = [10 10 20 20 30 30 40 40];
scatter (x, y)              % Creates the scatterplot of matrix data
axis([-10 10 0 50]);        % Adjust Axes in format ([xmin xmax ymin ymax])
title('Distance Sensor Cone of Detection')
xlabel('X-axis position (cm)')
ylabel('Y-axis position (cm)')
```

This will generate a plot such as the one below.

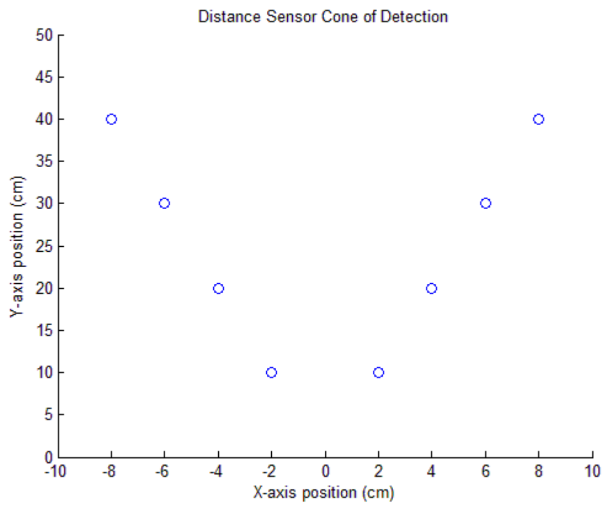


Figure 4: Example detection plot using MATLAB (fictional data)

Questions: What approximate angle do your data points form for a detection field? Are the readings for both sides symmetrical? What might explain your findings?

Now, repeat the basic edge detection procedure with the different objects from earlier tests:

- Bring the object into the window, then remove it from the field once it registers on the scope. Try to move the object at a constant speed.
- Increase the object height, then repeat.
- You do not need to manually log data for these. Save a screenshot for each item.

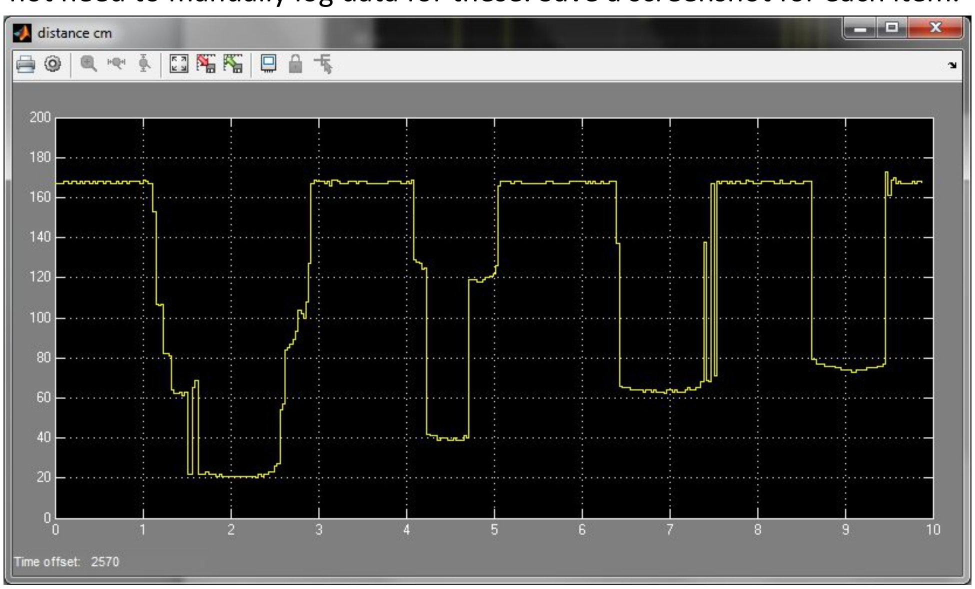


Figure 5: Edge detection with object 1

Questions: How does the edge detection vary with the objects? Why might this information be useful when considering applications for the distance sensor?