USING THE (I2C) LCD AND ULTRASONIC SENSOR

INTRODUCTION

We revisit the concept of a smart sensor in this lab and also investigate a smart output device. While a microcontroller is a powerful engine having robust, human readable output is a big benefit.

In this lab you process distance information from a smart ultrasonic ranging sensor and use the I2C communications protocol to talk to display information on an LCD device.

You will:
- Interface your ultrasonic sensor to your RoboRed microcomputer given a schematic.
- Interface your LCD (Liquid Crystal Display) module to your RoboRed microcomputer given a schematic.
- Learn about the I2C communications protocol.
- Continually display distance from the Ultrasonic sensor to object in millimeters or inches on LCD. Units are selectable via serial monitor input.

MATERIALS

- Arduino Programming book (Blum)
- Makerspace Kit
    - HC-SR04 Ultrasonic sensor, also called a “ping” sensor
    - LCD-Blue-I2C

Optional
- O-scope
- Power Supply.

BACKGROUND

The ultrasonic sensor used two transducers: One to generate and the other to detect sound pulses. The time of flight (TOF) for the pulse to be generated and the echo to be heard is what is the output. Because it is a smart sensor, it has associated electronics and even a dedicated microcontroller to interpret the raw data for us and to make it easier to use. Figure 1 shows a basic idea of what’s happening:

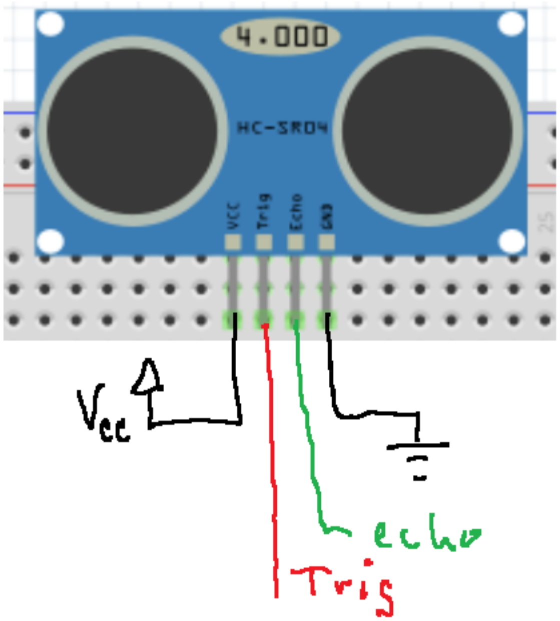
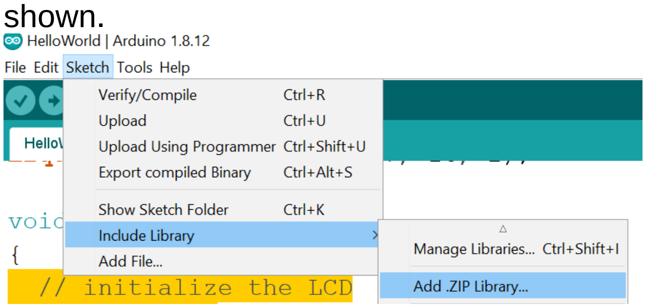


Figure 1. operation of a 4-pin ping sensor. The Arduino initiates the pulse with a short 10us pulse from a digital output pin. Then, another pin measure the time for the echo to return. The distance to the object is D = TOF/2c where c is the speed of sound approx. 340m/s. The echo pin is configured as input to the Arduino.

The LCD uses i2c communication to connect to it. We'll be using commands that print numbers and characters and strings.

EXERCISES

Downloads: Download from Bb the two zipped libraries for the LCD and the ping sensor. Use the sketch/include library/add .zip to make the libraries available, as shown.



NewPing-1.9.1.zip
Arduino-LiquidCrystal-I2C-library-master.zip

Part I. The ultrasonic sensor

The goal here will be to demonstrate the function of your ping ultrasonic sensor alone using the serial monitor. You will show how the distance to an object can be displayed on the serial monitor and control the units of distance from the serial monitor input.
Enter ‘i’ for inches
Enter ‘m’ for cm

Procedure

- Use the schematic/pictorial in figure 2 to construct your ultrasonic transceiver hardware interface.

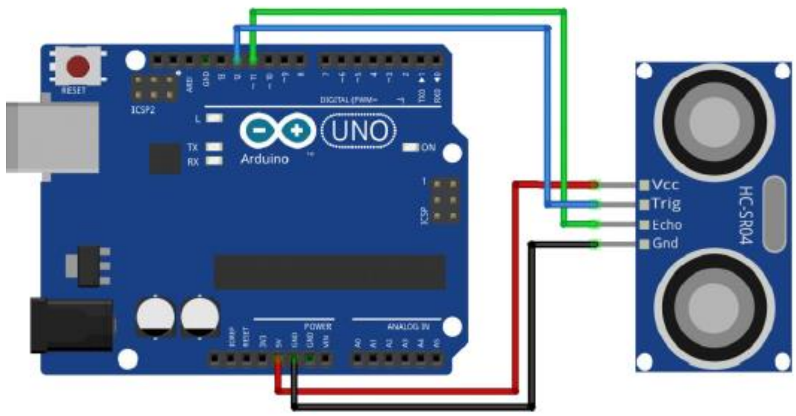


Figure 2. Layout of your ultrasonic sensor / Arduino

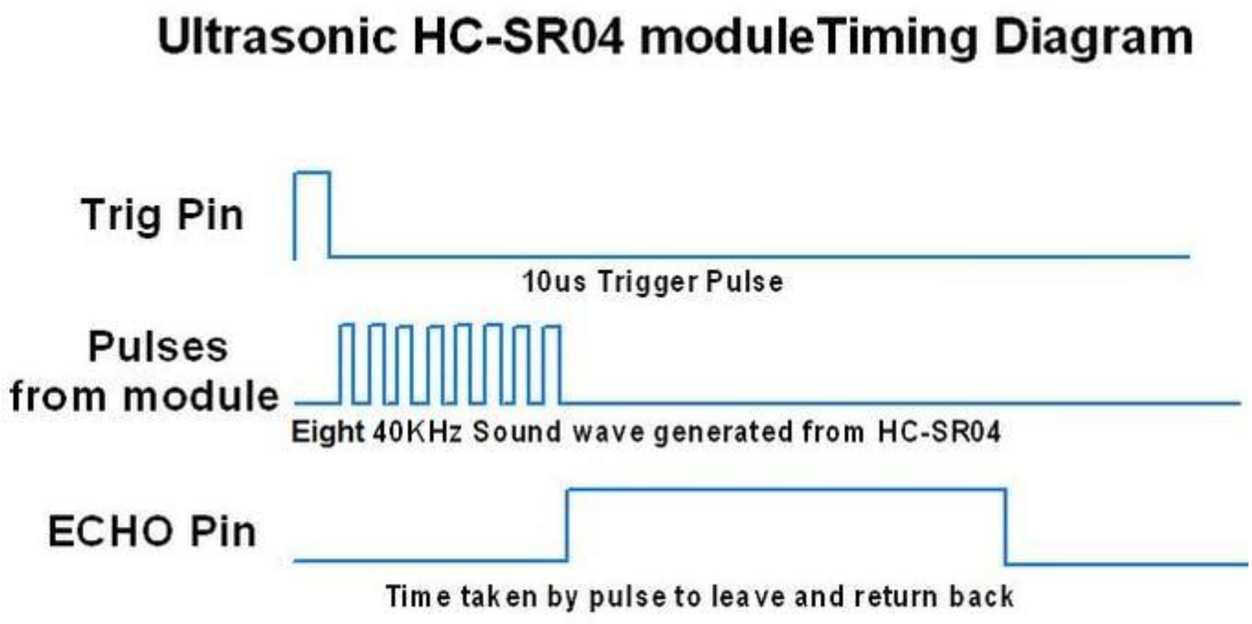
- Write your code to display sensor to object distance on the serial monitor.
- Create a command to display distance in either inches or millimeters. Consider using the Arduino “NewPing” library to help construct your software.

Helen—

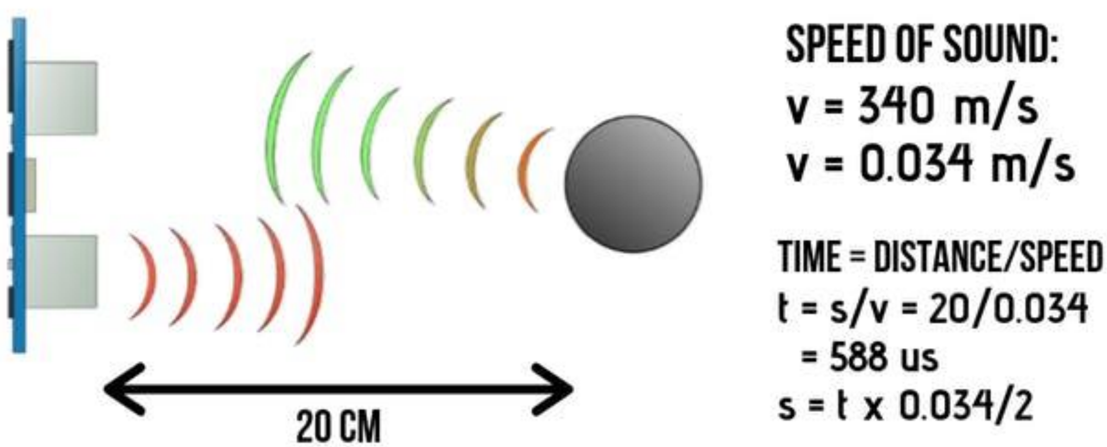
Ultrasonic Sensor HC-SR04 is a sensor that can measure **distance**. It emits an **ultrasound** at **40 000 Hz (40kHz)** which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The **supply voltage** of VCC is **+5V** and you can attach TRIG and ECHO pin to any Digital I/O in your Arduino Board.

In order to generate the ultrasound we need to set the **Trigger Pin** on a **High** State for **10 µs**. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo Pin. The Echo Pin will **output** the **time** in microseconds the sound wave traveled.



For example, if the object is 20 cm away from the sensor, and the speed of the sound is **340 m/s** or 0.034 cm/µs the sound wave will need to travel about 588 microseconds. But what you will get from the Echo pin will be **double** that number because the sound wave needs to **travel forward** and **bounce backward**. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.



For the programming code, first we need to define the Trigger Pin and Echo Pin that connected to Arduino board. In this project EchoPin is attached to **D2** and TrigPin to **D3.** Then define variables for the distance (int) and duration (long).

In the loop first you have to make sure that the trigPin is clear so we have to set that pin on a **LOW State** for just **2 µs**. Now for generating the **ultrasound** wave we have to set the **trigPin** on **HIGH State** for **10 µs**. Using the ***pulseIn()***function you have to read the travel time and put that value into the variable "duration". This function has 2 parameters, the first one is the name of the echo pin and for the second one you can write either HIGH or LOW. In this case, HIGH means that the ***pulseIn()*** function will wait for the pin to go HIGH caused by the bounced sound wave and it will start timing, then it will wait for the pin to go LOW when the sound wave will end which will stop the timing. At the end the function will return the length of the pulse in microseconds. For getting the distance we will multiply the duration by 0.034 and divide it by 2 as we explained this equation previously. At the end we will print the value of the distance on the Serial Monitor.

Steps :

1. First do the wiring as shown in the picture

2. Open Arduino IDE Software and write down your code, or download the code below and open it

3. Choose your own Arduino board (in this case Arduino Uno), by selecting **Tools** > **Board** > **Arduino/Geniuno Uno**

4. Choose your COM Port (usually it appears only one existing port), **Tools** > **Port** > **COM..** (If there are more than one ports, try it one by one)

5. Upload your code by pressing **Ctrl + U** or **Sketch** > **Upload**

6. To display the measurement data you can use Serial Monitor by pressing **Ctrl + Shift + M** (make sure that the baudrate speed is 9600)

Part II Integrate the LCD

Iin this part your will integrate the I2C-based LCD to your RoboRed microcontroller. First you'll simply demonstrate a "Hello World" output to the LCD then demonstrate the output obtained from the ultrasonic sensor. LCD output must be switchable between inches and millimeters upon command from the serial monitor.

Procedure

- Assemble the layout shown in Figure 3. (SDA goes to A4, SCL goes to A5)

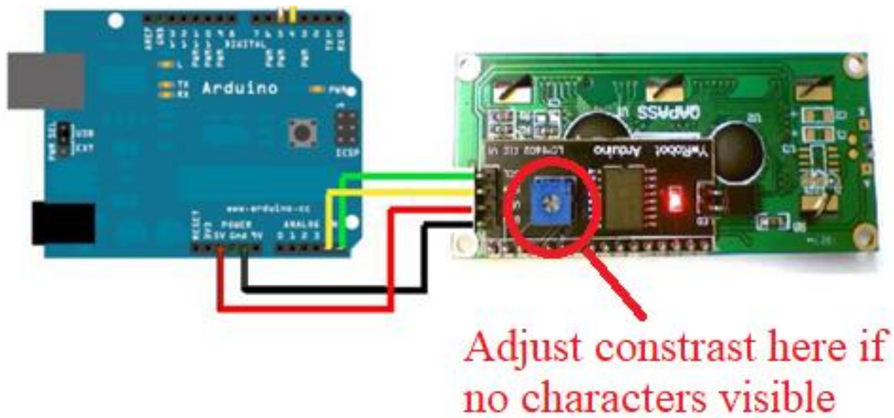


Figure 3. Layout of the LCD. You may need to adjust the screen bias to see the actual characters.

- Integrate into your development environment.
- Build the code to interface to your LCD.
- Output a test "Hello World" message to the LCD.
- Some notes:
  - You can substitute serial print messages with lcd
    Serial.print(sonar.ping_cm());   can become
    lcd.print(sonar.ping_cm());

  - You made need some other commands to get the screen in a readable format, for instance clear and set cursor:
    lcd.clear();
    lcd.setCursor(5,0);

- Lastly, output the distance value from your Ultrasonic sensor to the LCD. If you are really good you will append units to the numerical value.
- Incorporate the display unit change ability into the LCD readout.

Write up a description of the lab. Append the report with your commented code as an appendix.