
cs281: Introduction to Computer Systems
Lab04 – Sensors and Actuators

The purpose of this lab exercise is to explore ways the Arduino communicates with the external environment. We will incorporate one *sensor* – a device which gathers information from the external environment for the Arduino, and one *actuator* – a device in which the Arduino affects physical change to the external environment. We will add more sensors and actuators in future labs as we expand the ways to use the Arduino.

This lab is also structured differently. There are two *exploratory exercises* intended to allow us to learn the basic operation of a particular sensor or actuator and to build a very simple working example. There is then a *design exercise* in which your lab team designs a solution to a project, in this case using the sensors and actuators. Your lab report will be different as well. It will focus solely on the design exercise. You are to describe the design and operation of your solution to the presented design challenge.

Exploration: Buzzer

In our first exploratory exercise we implement a buzzer, a simple actuator. The buzzer affects physical change in the external environment by introducing a sound which is controlled by the Arduino.

The buzzer has two input pins labeled + and -. There is a metallic membrane inside the buzzer. In its rest state when no voltage is applied across the input pins, the membrane is contracted (bubble in). When a +5 voltage difference is applied across the input pins, the metallic membrane enters an excited state (bubble out). The transition between rest state to excited state or vice versa produces an audible click. By toggling the voltage very quickly, we can cause the buzzer to produce a tone. The frequency of vibration determines the frequency (pitch) of the tone.

Test 1

1. Plug the buzzer onto the breadboard so that the two pins span two distinct rows. Make sure the + pin is in the higher row (up) and the - pin is in the lower row (down).
2. Use a wire to connect the - pin to ground. Plug a longer test wire in the + pin.
3. Now take the other end of the + wire and alternately touch it and release it to any +5v connection. You should hear an audible "click" each time you

touch or release the +5v. You can also plug it into the pushbutton on your breadboard to transition back and forth.

Test 2

1. Plug the + end of the buzzer wire into the breadboard function generator. Make sure you plug it into the TTL input!
2. Play with the frequency knobs, both the discrete selector and the continuous selector. What do you observe about the buzzer? Record some observations here as this "data" will be helpful later on.

Test 3

1. Plug the + end of the buzzer wire into Arduino pin 13.
2. Wire the Arduino +5v and Ground connections.
3. Write and upload the following program. What do you observe?
4. What happens if you alter the value for p? Try values between 1 and 5.

```
#define BUZZER 13
int once = 0;

void setup ()
{
  pinMode(BUZZER, OUTPUT);
  Serial.begin(9600);
  Serial.println("Start");
}

void loop ()
{
  int p = 3;
  int i;
  if ( !once )
  {
    Serial.println("Starting buzzer");
    for ( i = 0; i < 500; i++ )
    {
      digitalWrite(BUZZER, HIGH);
      delay(p);
    }
  }
}
```

```
        digitalWrite(BUZZER, LOW);  
        delay(p);  
    }  
    once = 1;  
    Serial.println("Stopping buzzer");  
}  
}
```

Exploration: Ultrasonic Sensor

In this second exploratory section, we learn to use the ultrasonic sensor, aka sonar device. This device measures distance by using sonar; it sends out a ping (ultrasonic – you can't hear it) and records the time the ping echoes back. The time interval, along with the speed of sound, tells you the distance to the object upon which the sound waves echoed.

Obtain a Parallax Ultrasonic Sensor, which has two "microphone" looking things and four pins emitting from the bottom. You will also need a Arduino daughterboard and a battery power cable.

Test 1

1. First set up your daughter board. If it is still in its original packaging, you will need to open the package and mount the breadboard in the center using the adhesive on the back of the breadboard. Just center it somewhere in the middle. It helps to leave the styrofoam on the bottom of the daughterboard so you can press the breadboard on.
2. Now carefully mount the daughterboard onto the top of the Arduino. Be sure the pins line up exactly on both sides before you press. You will fill the rows of pins from the top (away from the Arduino plug) but will notice that a couple of pins remain open near the bottom edge (near the USB plug). Use the labels on common on both boards to make sure of the alignment – you can ruin the device if you get it wrong.
3. Plug your sonar sensor into the daughterboard (see the picture). We want it on the daughterboard instead of the main breadboard because the proximity of other stuff on the breadboard will give you false echo readings. By mounting it on the auxiliary board, you can "aim" it toward an object and be clear of other breadboard structures. Your four pins should span four rows of the breadboard. I chose four pins that were on the Vin and GND side, but away from these connections so that I can access them. See the picture below.

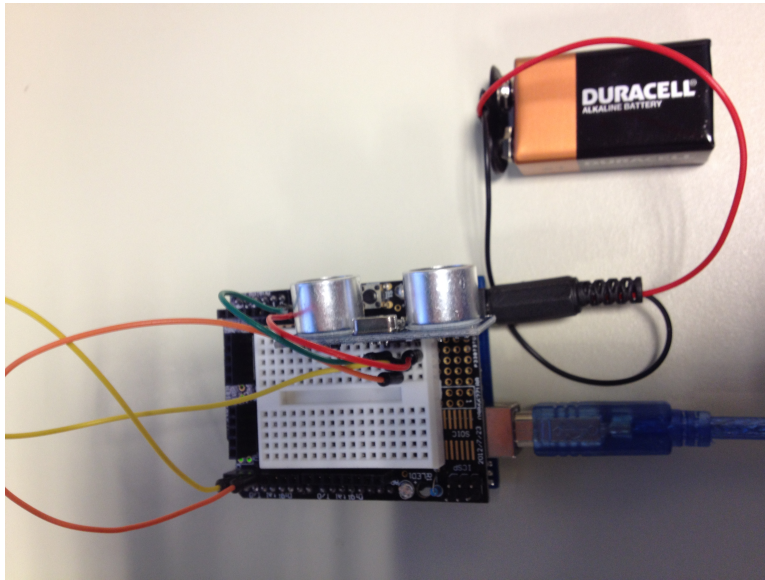


Figure 1: Mounting the Sonar Device

4. On the sonar sensor, connect the + pin (top pin, shown in red above) to +5 (+5 on the daughterboard, NOT the Vin pin) and the - pin (bottom pin, green above) to GND. You will power your Arduino with a battery so you do not need these pins to be wired to the +5V and Ground connections on the main breadboard as we usually do.
5. Connect the Trig pin (yellow) to pin 2 on the Arduino. This will be the pin which turns on the sonic device (an output for your Arduino, an input for your sonar device).
6. Connect the Echo pin (orange) to pin 3 on the Arduino. This will be the pin which reads the echo (input on Arduino, output on sonar device).
7. Connect your 9v battery to the power cable and plug it into the Arduino. Now connect the USB cable from the Arduino to the laptop.
8. Write, upload and run the following program. Open the serial window to see the results. Test the device by aiming the sensor at different objects or by placing objects (a book, a person, etc) in front of it at different distances. How accurate is it? How consistent is it – how stable are the readings? Can you "tune" the distance to be more accurate by playing with a program parameter?

```

#define TRIG 2
#define ECHO 3

void setup ()
{
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
  Serial.println("Start");
}

void loop ()
{
  Serial.println("Initiating Reading");
  digitalWrite(TRIG, HIGH);
  delay(10);
  digitalWrite(TRIG, LOW);
  int distance = pulseIn(ECHO, HIGH) / 2;

  distance = distance / 29; // tuning parameter
  Serial.print("Distance in cm is ");
  Serial.println(distance);
  delay(2000); // wait 2 seconds before next reading
}

```

Design Challenge: Audible Distance Indicator

The goal of this part of the lab is for you to design a solution (hardware and software) to solve a particular design challenge. You are to build a robotic distance detector that emits a sound whose pitch correlates with the distance to a nearby object.

1. Your device should be sensitive to (respond to) distances that span from very close (1cm or less) to a distance of 300cm (3 meters).
2. Your device should emit a sound tone. Use a low sound tone to indicate a far distance (3m or more). Use a high pitched tone to indicate a near distance (1cm or less). Have your device respond appropriately to intermediate distances.

3. Make your device fully portable. Yes, you may tether to a computer to download the program, but after that you should be able to operate the device while walking around without tethers to either the laptop or the main breadboard.
4. Do you want a linearly proportional relationship between distance and tone? Are there alternatives to consider?
5. Demonstrate the operation of your device to your professor or lab assistant.
6. Write a lab report in which you fully explain this design challenge and your design solution. Your lab report should stand alone from this lab handout. Fully articulate the design problem and list all your solution steps in your design. Discuss areas where you made design decisions and explain why you opted for the choice you made instead of other alternative designs. Complete one report per group. Report is due on Friday at the start of class.