LAB 6: IMPLEMENTING SENSORS

Objective

Interface a microcontroller board with analog and digital infrared (IR) range sensors.

Prelab Assignment

1. Review an example of switch debouncing code (see http://arduino.cc/en/Tutorial/Debounce) and determine how it debounces a switch using the `millis()` function. Then create your own code for debouncing a switch using the `delay()` function. As you construct your program, include the capacity for bypassing your debug code, such that an output pin on your Arduino board turns on and off based solely on the voltage you read at the switch output.
2. Draw a circuit diagram for connecting the digital IR sensor to your microcontroller board. Make sure the digital sensor output is connected to a digital input pin.
3. Write a simple program to read the digital IR sensor output and turn ON an LED when the sensor detects an object.
4. Draw a circuit diagram for interfacing the analog IR sensor with your microcontroller board. Make sure that the output from the analog sensor is connected to an analog input pin. For the analog sensor, include a 10 μF (or larger) capacitor between V_{cc} and GND.
5. Write a simple program that reads the analog IR sensor and outputs an appropriate integer code (0 to 1023) to the serial monitor.

Debouncing

When in normal operation, most people think that a switch is only in two states, on or off. However, there is another state where the switch is not connected to anything. This state occurs through the transition between high and low, and usually provides an unknown output. To prevent this reading, several different implementations have been adapted for "debouncing" a switch. The most common method when working with a microcontroller is to incorporate an internal timing delay after a pushbutton switch changes state.

IR Range Sensors

In this lab, you will be using Sharp IR distance sensors: either the GP2Y0A02 or GP2D12 analog sensor and the GP2D05 digital sensor. The analog sensors produce a voltage signal that is dependent upon the distance between the sensor and an object. The digital sensor output is LOW when no object is detected and HIGH when an object is detected. All such sensors have a limited distance range and may produce erroneous results when being used on a distance outside of their range. Make sure you review the specification sheets before you begin wiring.

Procedure

1. **Debouncing a Switch**

   Wire a switch to your Arduino board, using either a pull-up or pull-down resistor, as shown in Figure 1. Implement your code for debouncing a switch as developed in the Prelab Assignment.

   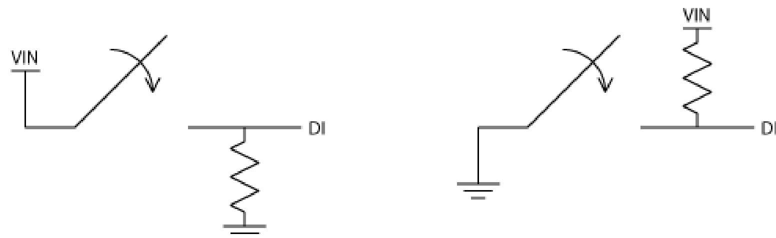

   Figure 1: Switch wiring.

   (a) Use an oscilloscope to examine the switch output, as well as the microcontroller output you activate with your debouncing code. If necessary, modify the time delay in your code to ensure that it works properly.

   (b) What is the purpose of debouncing a switch?

   (c) Does your code work as expected?

   (d) Include oscilloscope screenshots in your report showing the raw switch output, as well as the Arduino output voltage. from your debouncing routine.

2. **Using the Digital IR Sensor**

   Construct the digital IR sensor circuit you designed in the Prelab. Implement your digital IR sensor code, also from the Prelab Assignment.

   (a) Does your code work as expected? If necessary, replace the digital IR sensor with a push-button to test your program.

   (b) According to the specification sheet, at what range of distances will the digital sensor output a LOW signal?

   (c) Using a yard stick (or tape measure), record the range of distances for which the digital sensor will output a LOW signal.

   (d) Does your range match with the values given in the specification sheet?

3. **Using the Analog IR Sensor**

   Construct the analog IR sensor circuit you designed in the Prelab. Implement your analog IR sensor code, also from the Prelab Assigment.

   (a) Does your code work as expected? If necessary, test your program by replacing the analog IR sensor with a potentiometer.

   (b) Using a yard stick (or tape measure) to measure distance, and a multimeter to measure voltage, create a graph of the relationship between output voltage and object distance.

   (c) What is the maximum output voltage from your analog IR sensor, and at what distance does it occur?

   (d) Does your data match with the values given in the graph on the specification sheet? If not, why do you think this is so?

   (e) Extend your Prelab code so that it outputs the voltage from your analog IR sensor (rounded to a single digit; i.e. 3.14159 V $\Longrightarrow$ 3) to a seven-segment display. (See Lab 5 for a pin out of a seven-segment display. Remember to use appropriate current limiting resistors.)

4. **Bonus: Controlling a Motor Based on Distance to Object**

   Build a finite state machine to read an analog IR sensor and then output motor commands (go/stop, forward/backward, slow/fast) based on the distance to an object. Use the logic shown below. The "Emergency" input may be a push button, optical interrupter or other device to indicate a crisis condition. Use LEDs to indicate the outputs of your finite state machine.

Sensing	Motor Response
No Object	Fast Forward
Object Visible	Slow Forward
Object Close	Slow Backward
Emergency	Stop

Table 1: Motor control logic

   Does your code work as expected? Demonstrate your device for the instructor. Include a state transition diagram in your report.