

## LAB 5: ANALOG INPUT AND OUTPUT

### Objective

1. Gain microcontroller programming experience.
2. Practice interfacing with analog I/O.

### Prelab Assignment

1. Review the Arduino's `analogRead()` command (see <http://arduino.cc/en/Reference/analogRead>). What is the input voltage range? What is the output integer range? What is the sampling rate?
2. Review the Arduino's `analogWrite()` command (see <http://arduino.cc/en/Reference/analogWrite>). Familiarize yourself pulse width modulation (PWM) output.
3. Describe the `analogWrite()` function.
4. Do PWM output pins need to be declared in the `setup` function?
5. What is the duty cycle of output pin 11 with the following code: `analogWrite(11, 127)`?
6. Write Arduino code to read an analog voltage, display the value using the Serial Monitor, and output the same voltage using PWM.

### Analog I/O

The Arduino microcontroller can read analog inputs between 0 and 5 V and generate corresponding integer values between 0 and 1023. This is accomplished using the `analogRead()` function. Look at the following link for assistance on understanding analog inputs: <http://arduino.cc/en/Tutorial/AnalogInput>

Pulse width modulation (PWM) is a means for generating an analog signal from a digital output. Look at the following link for assistance on implementing a PWM output: <http://arduino.cc/en/Tutorial/AnalogInOutSerial>

A PWM output generates a repeating pattern of high and low outputs at varying duty cycles. When the frequency of the on/off duty cycle of a PWM output is fast enough, certain electrical equipment will read the signal as the average voltage rather than turning on and off quickly. The plot below shows varying duty cycles generated with the `analogWrite()` command.

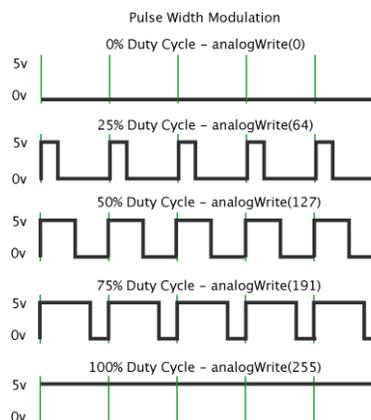


Figure 1: Sample duty cycles generated by `analogWrite()`

## IR Range Sensors

In this lab, you will be using a Sharp analog infrared (IR) distance sensors: either the GP2Y0A02 or GP2D12. These analog sensors produce a voltage signal that is dependent on the distance between the sensor and an object. All such sensors have a limited operating range and may produce erroneous results when used at distances outside of their specified range. Make sure you review the specification sheets before you begin wiring.

## Seven-segment Displays

If you attempt the bonus portion of this lab, you will use a seven-segment display (see Figure 2). Seven-segment displays are available in either a *common-anode* or *common-cathode* package. A common pinout for a common-anode display is shown in Figure 3. Schematics for the common-anode and common-cathode packages are shown below. Note that the digital output (HIGH or LOW) required to illuminate an LED segment depends upon the type of seven-segment display you are using.

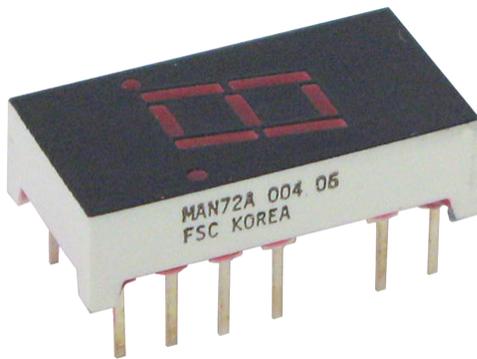


Figure 2: Seven segment LED display

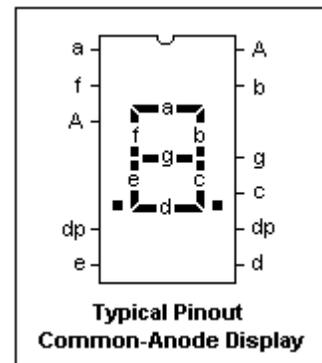


Figure 3: Seven segment LED display pinout

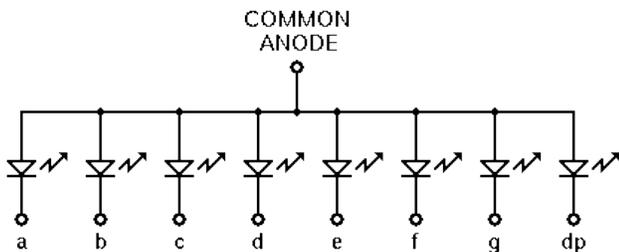


Figure 4: Common-anode schematic

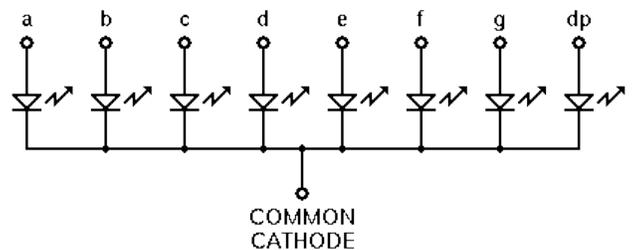
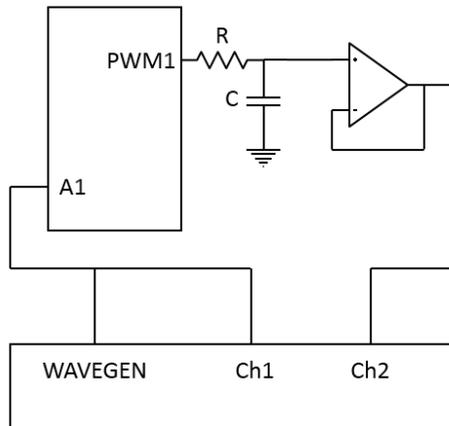


Figure 5: Common-cathode schematic

## Procedure

### 1. Analog Input and Output

- (a) Build the circuit shown below using your Arduino, an oscilloscope, and an RC filter with an op-amp follower. Choose  $R$  and  $C$  to have a break frequency of 100 Hz. Load your Prelab code into the Arduino.



- (b) Apply a sinusoidal input voltage from 0 to 5 V at 10 Hz to the Arduino analog input.
- (c) Observe the Serial Monitor. What do the numbers represent?
- (d) Calculate the Arduino ADC quantization interval (amount of input voltage change over which the converted integer count remains constant). Why is this value important?
- (e) Observe the analog input and output of the Arduino. How do they compare? Does it matter if the Serial Monitor is active?
- (f) Can you find a voltage and frequency combination where the analog input and output no longer match? Are you able to explain why the input and output might differ?

### 2. Analog Sensor Reading

- (a) Replace the oscilloscope function generator with an analog IR sensor.
- (b) Using the oscilloscope, observe the noise on the analog IR sensor output. What voltage range does the broadband noise on the signal span? Include an oscilloscope screenshot in your report.
- (c) Using the Arduino quantization interval, how many digital codes will be spanned by the measured broadband noise voltage range? Verify your calculation using the Serial Monitor.
- (d) The analog IR sensor output includes a periodic spike of noise. Identify the frequency and maximum amplitude of the noise spike. Include an oscilloscope screenshot in your report.
- (e) Using the Arduino quantization interval, how many digital codes will be spanned by the measured noise spike amplitude? Verify your calculation using the Serial Monitor.
- (f) Filter the noise using an RC filter on the IR sensor output. Does your filter work? Include a before and after comparison of oscilloscope screenshots in your report.
- (g) Filter the noise using an averaging filter in the Arduino code. Does your filter work? Include a before and after comparison of oscilloscope screenshots in your report.
- (h) Identify two advantages and disadvantages for the hardware (RC filter) implementation.
- (i) Identify two advantages and disadvantages for the software (averaging filter) implementation.
- (j) How might you use these results in your final project?

### 3. Bonus: 4-bit Binary Counter with Speed Control and Display

- (a) Design and implement a 4-bit digital up counter as a state machine. Use a potentiometer as a voltage divider to provide an analog input that varies the speed of counting. Use the Serial Monitor to validate your design.
- (b) Write Arduino code that displays the output of your counter on a seven-segment display. Build a circuit to test your code. Include a circuit diagram and the display code in your report. Does your code work as expected?
- (c) For several different potentiometer settings, measure the voltages being applied to the analog input pin, and compare this with the corresponding counting speeds. Do these results match your expectations?
- (d) What is the effect of using a single current-limiting resistor at the common node of the seven-segment display, as compared to using separate resistors at the individual nodes?