## Documentation for Workshop 3 - Circuits and Microcontrollers

Yixuan Xu

Manqiu Wu

Xinghao Li

Part 1: Microcontrollers, Arduino Uno, and Outputs

* To choose the resistors in this circuit, we noted that the voltage is 5 V and that the LEDs have a current rating of 12-20 mA. This means that we should select resistors so that the current is controlled to be below 12 mA to avoid being burned. Using the equation V = IR, we found that the resistance should be above 500 ohms to control the current at 10 mA. Given that the internal resistance of LED is 13 ohms (according to Google), we recognize that a 500 ohms resistor should suffice.
* However, the resistors we are given have large resistances (10k and 21k). To reduce the resistance, we set two 10k resistors in parallel, giving 3387 ohms of equivalent resistance.
* When confirming the voltage on pin, multimeter gives a reading of 5.10V from output pin to ground, and 2.10Vacross LED. If we don’t have resistor in place, the current will be very high, at 0.38A, which will damage the LED.
* After building the script and circuit to make the LED blink, we measure the output voltage, and verified that there is PWM signals. This is because the voltage is varying between 0V and 5V over time.

Part 2: Inputs

* To confirm whether the digital readings match the analog input the sensors are producing, we adjusted the analog voltage and compare it with converted digital output on the microcontroller, and recorded the values in the following table:

|  |  |  |
| --- | --- | --- |
| Analog values | Calculated Digital Signal | Measured Digital Signal |
| 1.25V | 255 | 250 |
| 1.75V | 357 | 348 |
| 3.3V | 674 | 663 |
| 4.2V | 859 | 846 |
| 4.8V | 982 | 963 |

* To convert from analog input to output, we scaled with the following equation (input+1)/4-1=output

Part 3: Sensors

* After setting up the wiring for IR proximity sensor, we tested the relationship between reading, analog voltage, and distance of objects. The analog voltage was calculated by direct conversion using the scale from the table above. Note that the reading value fluctuates constantly, and the average of three measurements was used as the integer reading. We believe that the operating range is between 5 cm and 35 cm. Because at below 5 cm, the analog voltage decreases. The peak voltage is at 5-6 cm, meaning that the sensor is able to start detecting the object. When the distance is above 35 cm, the reading becomes inconsistent and fluctuates greatly, which we believe is due to the distance being out of range.

|  |  |  |
| --- | --- | --- |
| Integer Reading | Analog Voltage (y = 201.64x - 3.0255) | Distance |
| 360 | 1.8 | 1 cm |
| 300 | 1.5 | 2 cm |
| 465 | 2.3 | 3 cm |
| 561 | 2.8 | 4 cm |
| 620 | 3.1 | 5 cm |
| 625 | 3.1 | 6 cm |
| 601 | 3.0 | 7 cm |
| 552 | 2.8 | 8 cm |
| 500 | 2.5 | 9 cm |
| 432 | 2.2 | 10 cm |
| 300 | 1.5 | 15 cm |
| 261 | 1.3 | 20 cm |
| 207 | 1.0 | 25 cm |
| 199 | 1.0 | 30 cm |
| 184 | 0.9 | 35 cm |
| 170 | 0.9 | 40 cm |

* After setting up the wiring for ultrasonic sensor, we tested the relationship between reading, analog voltage, and distance of objects. To convert from ping time to distance, we use (velocity of sound X ping time / 2) = distance
* The operating range seems to be above 2 cm, and since the computed distance is different from the actual distance, we need to calibrate the sensor. The computed distance needs to be adjusted by adding approximately 0.5 cm to obtain a more accurate reading.

|  |  |  |
| --- | --- | --- |
| Ping Time (micrometer) | Computed Distance (cm) | Distance |
| -430 | -8 | 0 cm |
| 236 | 4 | 1 cm |
| 103 | 1.78 | 2 cm |
| 148 | 2.56 | 3 cm |
| 203 | 3.51 | 4 cm |
| 251 | 4.34 | 5 cm |
| 302 | 5.22 | 6 cm |
| 365 | 6.31 | 7 cm |
| 417 | 7.21 | 8 cm |
| 468 | 8.10 | 9 cm |
| 564 | 9.76 | 10 cm |
| 835 | 14.45 | 15 cm |
| 1112 | 19.24 | 20 cm |
| 1420 | 24.57 | 25 cm |
| 1676 | 28.99 | 30 cm |
| 1986 | 34.36 | 35 cm |
| 2260 | 39.10 | 40 cm |

* After we built the circuit of distance controlled LED, we found that the sensors are fairly accurate in controlling the brightness of light when the distance is between 2 and 40 cm for the ultrasonic sensor. When the sensor is facing the surface at an angle, the accuracy is not affected significantly. However, when the angle reaches a “critical value” (approximately 40 degrees), the reading starts to change uncontrollably.
* When counting the number of times, we found that ultrasonic sensor always glitched and counted once at the beginning first. Therefore, to remedy this, we subtracted one from the sum.

Topics: digital input/output handling and pulse width modulation

Tasks: blinking, dimmable, distance-controlled

Arduino:

* 20 GPIO pins, (Mega has 54 pins), different functions (digital i/o 0-13, PWM ~ for simulate analog output, serial communication, analog pin for reading analog inputs)
  + Max current 40mA, else will damage microcontroller
* Atmel ATmega328 microcontroller to interact with external and store program (brain)
* Voltage regulators to supply 3.3V and 5V voltage sources
* Plugins (shields) to give more functionalities
* Aref: set and control external reference voltage

I/O

* Digital output: HIGH (5V) or LOW (0V)
* Digital input: HIGH above threshold, LOW below
  + When pins have nothing connected to them will report seemingly random changes in pin state, picking up electrical noise or capacitive coupling
  + To solve, set default value to input port, setup externally or built-in Arduino fcn
  + When initial default to LOW, use pull-up resistor
  + When initial default to HIGH, use pull-down resistor
* Analog output: between 0V and 5V, achieved with pulse width modulation (PWM), cycling signal between high and low in duty cycle, set between 0, 63, 127, 255 steps (0 to 100% duty cycle)
* Analog input: take readings between 0V and 5V, achieved with 10-bit Analog to Digital Converters (A0 to A5), output reading values between 0 to 1023
* Sensors: IR/Ultrasonic range sensors
  + IR depends on colour, lighter means more radiation back
  + Ultrasonic distance: time take X speed of sound / 2
* LED: light-emitting diode, with current rating around 12-20mA, longer lead is positive terminal

Computer-to-microcontroller serial communication: display values to serial window to debug

Potentiometer: continuously variable resistor

Codes

* + int redLEDPin=13; //initialize variable redLEDPin for later use
* setup(): executed once, used to set pin modes, begin serial communications, initialize variables
  + pinMode(redLEDPin,output); //set digital pin 13 as output
  + pinMode(pin,INPUT\_PULLUP); //set pin to input with pull up enabled
  + Serial.begin(9600); //initiate use of computer-to-microcontroller serial communication
* loop(): executed repeatedly
  + digitalWrite (redLEDPin,HIGH); //activate pin 13 and give 5V
  + delay(1000); //delay by 1000 milliseconds
  + digitalWrite (redLEDPin,LOW); // turn off pin 13 and give 0V
  + for (int j=1;j<=10;j=j+1){ //code} // execute code 10 times
  + Serial.println(redLEDPin); //display 13 onto serial window
  + analogWrite(redLEDPin,127); //set LED to 2.5V
  + readValue = digitalRead(2); // read voltage on pin 2 and assign to readValue
  + readValue = analogRead(pin); // read voltage on pin and assign to readValue
  + digitalWrite(trigPin,LOW); delayMicroseconds(2000); digitalWrite(trigPin,HIGH); delayMicroseconds(10); digitalWrite(trigPin,LOW); pingTime=PulseIn(echoPin,HIGH); // For ultrasonic sensor, trigPin is trigger pin