Lab Two Memo

Abhishek Adhikari

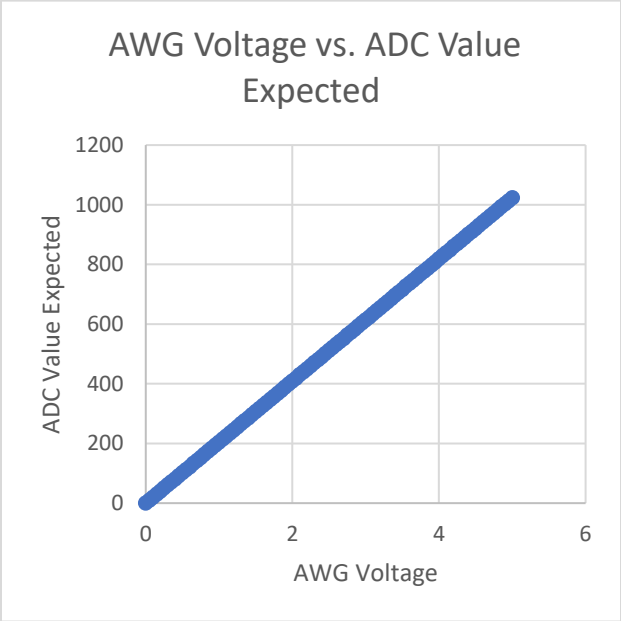
Arduino Code

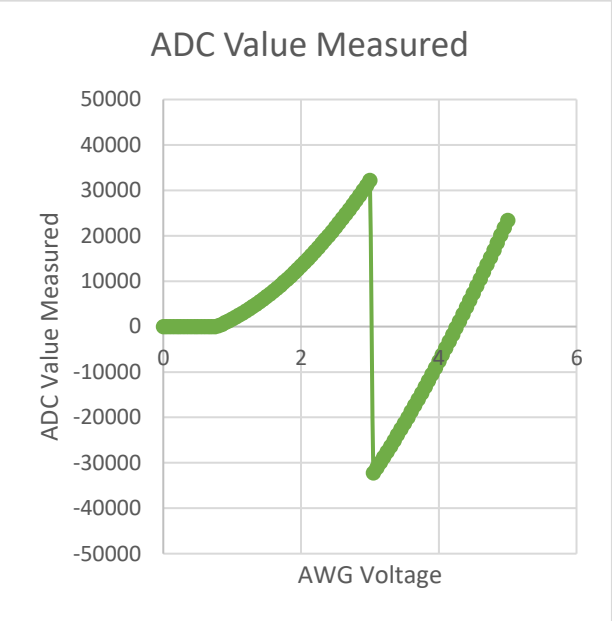
```
//const int input = A0; // This is where the input is fed.
int pulse = 0; // Variable for saving pulses count.
int var = 0;
int digitalPin = 20;
char IncomingByte;

int val=0;
const int pin_in=A0;
int pin_out=10;
void setup() {
  // put your setup code here, to run once:
  TCCR2B = (TCCR2B & 0xF8) | 0x03;  //pg. 57
  //TCCR1B = (TCCR1B & 0xF8) | 0x02;  //pg. 57
  pinMode(pin_in, INPUT);
  attachInterrupt (digitalPinToInterrupt(digitalPin),pulse_counter,HIGH);
  Serial.begin(9600);
  //Serial.println(F("No pulses yet...")); // Message to send initially (no pulses detected yet).
}

void loop() {
  // put your main code here, to run repeatedly:

  val=analogRead(pin_in);
  val << 2;
  analogWrite(pin_out,val/4);

  //TCCR2B = TCCR2B & ~B00110000; //switch off output B
  //TCCR2B |= B00110000;  //switch on the B output with inverted output
  //analogWrite(pin_out,1023);
  if (Serial.available() > 0){
      IncomingByte=Serial.read();
      Serial.println(pulse);

  }
     |
}

void pulse_counter()
  {

    pulse++;
   // while(Serial.available()) {
    //Serial.println(float(pulse),0);

    }
```

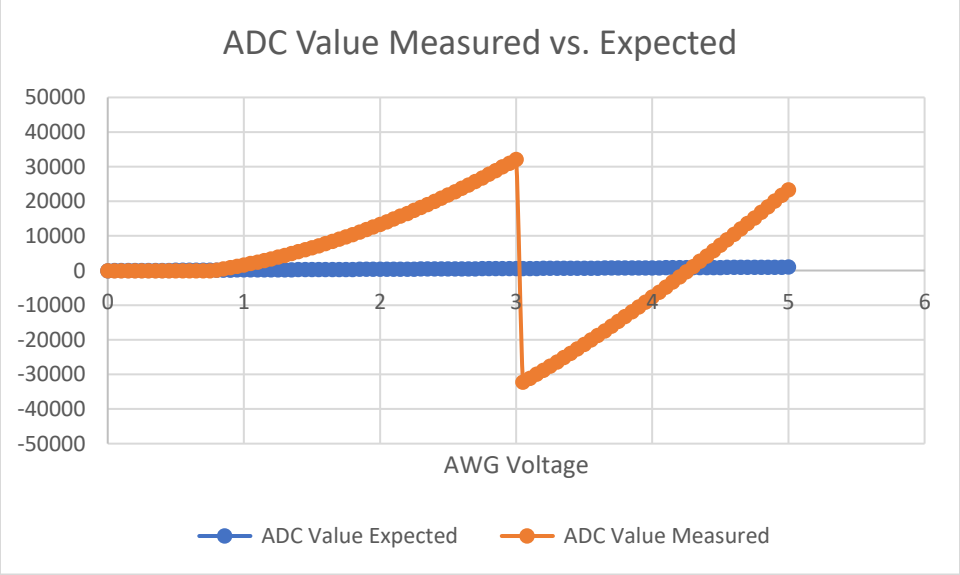
Python Code

```
# ADC Experiment
import visa
import numpy as np
import time
import serial

# Input v al u e s
com_port = 'COM3'
baud_rate = 9600
awg_address = 'USB0 : : 0 x0957 : : 0 x0407 : : MY44043483 : : 0 : : INSTR' # AWG
Address
awg_volts = np.linspace(0,5,101) # Duty c y c l e v e c t o r (%)
num_points=len(awg_volts)

# Initialize important matrices
ADC_measured=np.zeros(len(awg_volts))
ADC_expected=[int(x/5*1024) for x in awg_volts]

# Initiate communications with and open instruments
rm = visa.ResourceManager()
awg = rm.open_resource(awg_address)

# Place waveform generator in High-Z
awg.write("OUTP:LOAD INF")
# Initialize and open serial communications
arduino = serial.Serial(com_port, baud_rate)
arduino.reset_input_buffer()

# Per form measurements
count=0
for V in awg_volts:
    print("Analyzing Voltage %f" %V)
    awg.write("APPL:DC DEF,DEF,"+str(V))  # AWG output signal
    arduino.reset_input_buffer()
    time.sleep(2)  # Pauses to guarantee a good RPM count
    # Get ADC value from Arduino
    arduino.write(b'1\n')
    #arduino.reset_output_buffer()
    time.sleep(2)
    ADC_measured[count] = float(arduino.readline())
    count += 1

# Close communications
savename=experiment_name+'_data.csv'
arduino.close()
awg.close()
with open(savename,'w') as f:
    f.write( "AWG Voltage, ADC Value Expected, ADC Value Measured\n")
    for val in range (len(awg_volts)):
        f.write( "{vol}, {ADCe}, {ADCm}\n".format (vol=awg_volts[val] ,
                                                   ADCe=ADC_expected [val] ,
                                                   ADCm=ADC_measured [val]))
```

Data


AWG Voltage vs. ADC Value Expected
ADC Value Expected
1200
1000
800
600
400
200
0
0
2
4
6
AWG Voltage


ADC Value Measured
ADC Value Measured
50000
40000
30000
20000
10000
0
-10000
-20000
-30000
-40000
-50000
0
2
4
6
AWG Voltage


ADC Value Measured vs. Expected
50000
40000
30000
20000
10000
0
-10000
-20000
-30000
-40000
-50000
0
1
2
3
4
5
6
AWG Voltage
ADC Value Expected
ADC Value Measured