/\* Written by Blaine Harris

\* MXET 375 - Fall 2019

\* This Arduino Sketch (code) controls the instrumentation of the Lab created for the final project.

\*

\* Circuit Components:

\* - Arduino Nano

\* - INA219 Current Sensor

\* - IRF520N N-Channel MOSFET (Or DC Motor H-Bridge)

\* - Uxcell 12V DC Motor with Encoder

\* - (1x) 5K Potentiometer

\* - (1x) 100 Ohm Resistor

\* - (1x) 10K Resistor

\* - Breadboard, Prototyping Board, or PCB to assemble circuit

\*

\* NOTE: This sketch was created to provide PWM to the DC motor (Controlled w/ Potentiometer)

\* However, it was decided for simplicity sake to just modify the voltage of the variable power supply.

\* KEEP THE POTENTIOMETER AT 100% (100% Duty Cycle) FOR ALL EXPERIMENTS

\* - PWM causes additional effects to the output of the motor, that are more difficult to model

\* IF USING MOSFET: An additional 4.8V (supplied from Arduino Digital Output) is added to the Vcc for the motor

\* this means if 12V is being supplied the voltage accross the motor is 16.8 volts.

\*/

#include <Adafruit\_INA219.h>

#include <Wire.h>

Adafruit\_INA219 ina219;

#define encoder1 11 // Yellow Wire from Motor

#define encoder2 10 // Green Wire from Motor

#define motorPWM 3 // PWM Ouput to Motor

#define potPin A3 // Pot for PWM Control

unsigned long freq, rpm;

float currentTime;

float potVal, percentVal, voltageVal, current\_mA;

float seconds;

int i = 0;

void setup() {

uint32\_t currentFrequency;

pinMode(encoder1, INPUT); // Setup GPIO 11 as Input

pinMode(encoder2, INPUT); // Setup GPIO 10 as Input

pinMode(potPin, INPUT); // Setup GPIO 10 as Input

pinMode(motorPWM, OUTPUT); // Setup GPIO 3 as OUTPUT

pinMode(2, OUTPUT); // Setup GPIO 1 as OUTPUT

ina219.begin();

Serial.begin(9600); // Baud Rate ENSURE this is the same in the Python Script

}

void loop() {

digitalWrite(2,HIGH);

current\_mA = 0;

currentTime = millis();

seconds = currentTime/1000;

// initialization delay (wait for step input)

if(currentTime < 250){

analogWrite(motorPWM, 0);

Serial.print("t");

Serial.println(seconds);

Serial.print("e");

Serial.println(0);

Serial.print("i");

Serial.println(0);

Serial.print("w");

Serial.println(0);

delay(10);

//i = i + 1;

} else {

potVal = analogRead(potPin); // Read Pot Value

percentVal = (potVal/1023); // Percentage of Pot Rotation

voltageVal = (16.8 \* percentVal); // Find Equivalent Voltage

current\_mA = ina219.getCurrent\_mA();// Record the Current

analogWrite(motorPWM, (255 \* percentVal)); // Output PWM Signal

if (voltageVal > 3){ // Is voltage high enough to power Motor?

freq = 100000/checkRPM(); // Provides Frequency in Hz

rpm = (freq \* 60); // Convert to RPM

} else {

rpm = 0;

}

publishData();

}

}

// Checking motor RPM

// Removed using both hall effect sensors, as using both was redundant

int checkRPM(){

int highPulse1, lowPulse1;

int pulseTotal1;

highPulse1 = pulseIn(encoder1, HIGH);

pulseTotal1 = highPulse1\*2;

return pulseTotal1;

}

// Write the recorded data over serial

int publishData(){

Serial.print("t");

Serial.println(seconds);

Serial.print("e");

Serial.println(voltageVal);

Serial.print("i");

Serial.println(current\_mA);

Serial.print("w");

Serial.println(rpm);

}