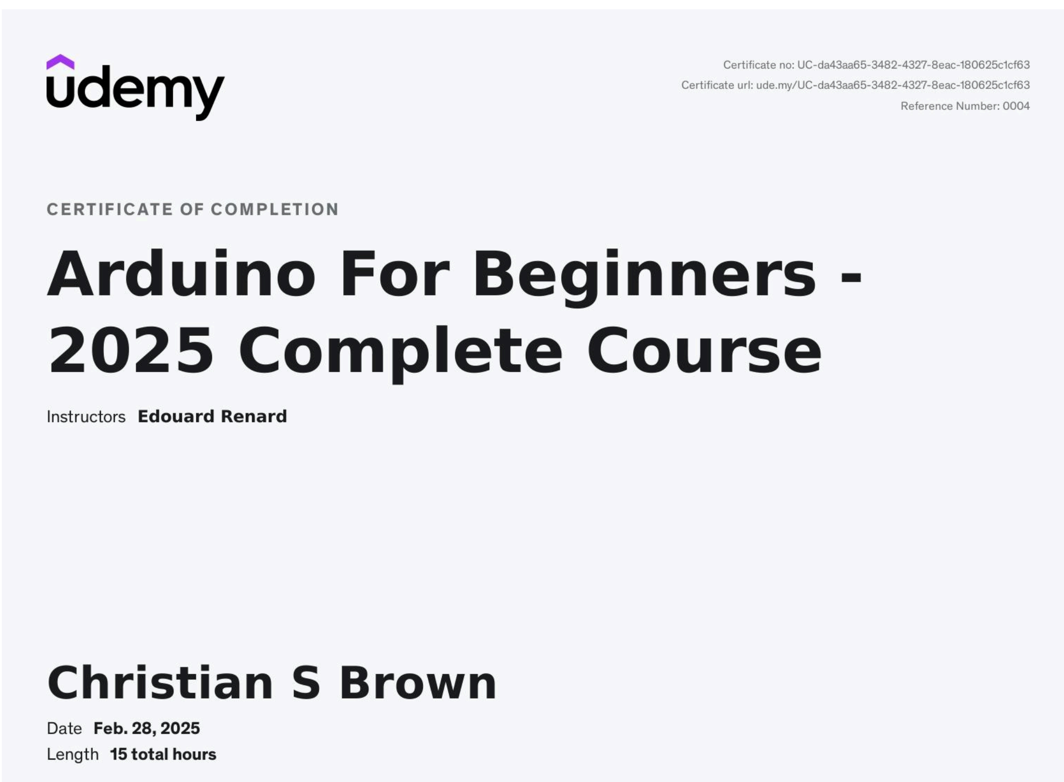
Christian Brown
ME 476C
Self Learning Assignment
Due: 2/28/2025

Self Learning Assignment

A New Skill

For this self learning assignment I began to learn how to set up and code an Arduino. I have never coded in C++, and the limited amount of coding I have done is with MATLAB for previous and current classes. I have never been good at coding because of the specifics that go into the notation, grammar, and wording of coding languages. I chose to learn coding an Arduino because our test bench, the dynamometer, runs on an Arduino. This way I can contribute to building both the circuitry and code of the Arduino to read the specific outputs of the generators we are going to test. These outputs include voltage, current, rpm, and torque of the generator. The dynamometer's Arduino setup is irreparable, so everyone on the team needs to contribute to creating and coding an Arduino circuit from scratch.

I used an Udemy course to help with my self learning. I took the "Arduino For Beginners - 2025 Complete Course" to help my fundamental understanding of the Arduino, the code it runs on, and the different applications the Arduino can be used for.


udemy

Certificate no: UC-da43aa65-3482-4327-8eac-180625c1cf63
Certificate url: ude.my/UC-da43aa65-3482-4327-8eac-180625c1cf63
Reference Number: 0004

CERTIFICATE OF COMPLETION

Arduino For Beginners - 2025 Complete Course

Instructors **Edouard Renard**

Christian S Brown

Date **Feb. 28, 2025**
Length **15 total hours**

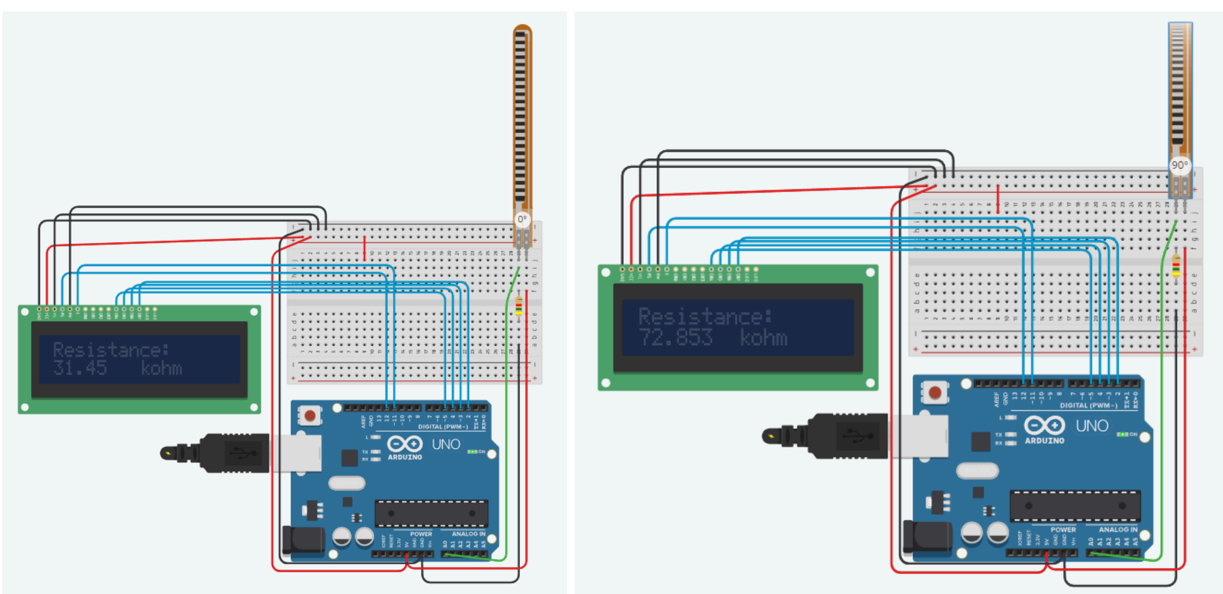
Figure 1: Udemy Course Certificate

Above in Figure 1 is the certificate for completing the online course. Fortunately the videos could be sped up so it did not take the full 15 hours as advertised. I did struggle on many of the in-lesson assignments, so that time nearly made up what was saved watching them at 1.5x speed.

Application

Applying this new skill to the dynamometer, I virtually built and coded an Arduino to output a resistance based on a flex sensor. The torque reader of the dynamometer is a torque load cell that reads the strain caused by the torque in the rotating shaft. The strain sensors output different resistances based on their length, similar to the flex sensor. This torque is important to find as it determines how much force is needed to turn the generator. This is an important factor in the small scale wind generators we are working to develop. The cut in wind speed, the lowest wind speed for the generator to start spinning, is directly related to the cogging torque. The lower the torque needed, the lower wind speeds and the better the generator will perform.

The program I used, TinkerCAD, did not have the strain sensors that are used in the load cell. So, the flex sensor was used to simulate a similar case of change in resistance due to change in physical geometry. Along with the Udemy course, I used another online guide outlining the use and integration of the flex sensor with an Arduino. To make the application more relevant to the dynamometer, I decided to add an LCD screen because values of the generators tested will be displayed on an LCD screen of our own.



Figures 2 & 3: Resistance of flex sensor at 0 degrees (left), Resistance of flex sensor at 90 degrees (right)

Above is the completed Arduino circuit with the flex sensor, a resistor of known resistance (4.7 kohms), and an LCD screen to show the active resistance of the flex sensor at different angles. When flexed, the flex sensor outputs a value called ADC that has to be altered to find the equivalent voltage due to the angle it is at. This flex voltage is then converted into resistance in kilo-ohms that is displayed on the LCD screen. The code used for this application is copied into the Appendix below. This simulation of a material outputting a signal due to a physical change is equivalent to the load cell, helping my understanding of how the load cell will be circuited and coded into the Arduino the dynamometer will use. For the load cell, the resistance change is converted into an equivalent torque due to the measurable strain of the piezoelectric material that is changing dimension. This relation is linear so calculating the equivalent resistance is easy once it is properly calibrated.

This self learning assignment helped me understand the Arduino we will be working on to complete the dynamometer. Finishing the dynamometer is our first priority as it will be our main way of testing generators to get important data from them. My skills in coding and the different coding languages grew as this was my first experience with C++. I will now be able to work with my team on completing the Arduino for the dynamometer as I was tasked with researching and working on the torque load cell on the dyno.

References:

[1]“Login - Tinkercad,” *Tinkercad*, 2025. https://www.tinkercad.com/things/bznAhwFtUzl-flexor-resistance/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits (accessed Mar. 01, 2025).

[2]“Login - CAS – Central Authentication Service,” *Udemy.com*, 2025. https://nau.udemy.com/course/arduino-for-beginners-complete-course/learn/lecture/27298892#overview (accessed Mar. 01, 2025).

[3]“Flex Sensor Hookup Guide - learn.sparkfun.com,” *Sparkfun.com*, 2016. https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all

Appendix:

```
#include<LiquidCrystal.h>
const int ADCpin = A0; //Choosing which pin the flexor is in
const float V = 5; // Plugged into the 5V pin on the Arduino
const float resistor = 4700.0; // The chosen resistance of the resistor
int ADCvalue;
// Defining which LCD ports are connected to which Arduino pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
 lcd.begin(16, 2); // Defines size of LCD
}

void loop()
{
  // Find value from bending, called ADC
  ADCvalue = analogRead(ADCpin);
  Serial.println(ADCvalue);
  // Convert ADC value to voltage to be used in resistance formula
  float voltage = ADCvalue * V / 1023;
  // Calculates resistance in kohms due to bending angle and displays it
  float resistance = (resistor * (V / voltage - 1))/1000;
  lcd.setCursor(0,0);
  lcd.print("Resistance: ");
  lcd.setCursor(0,1);
  lcd.print(resistance);
  lcd.setCursor(8,1);
  lcd.print("kohm");
}
```