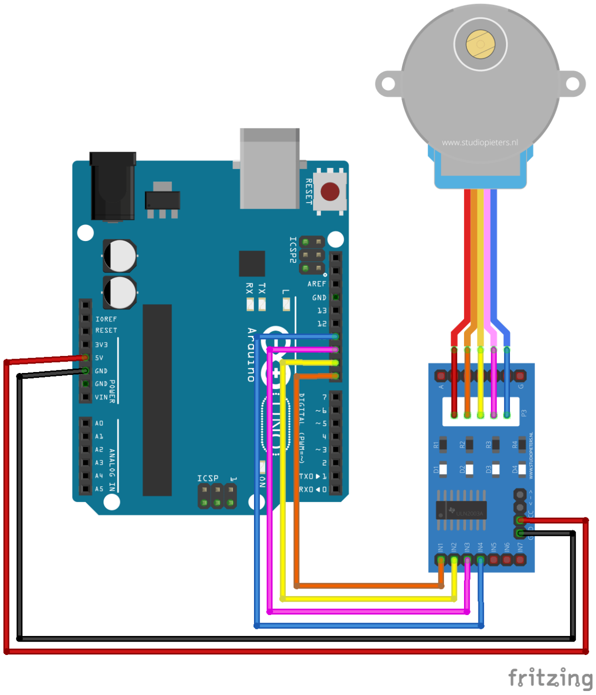
Will Stief

ECE387

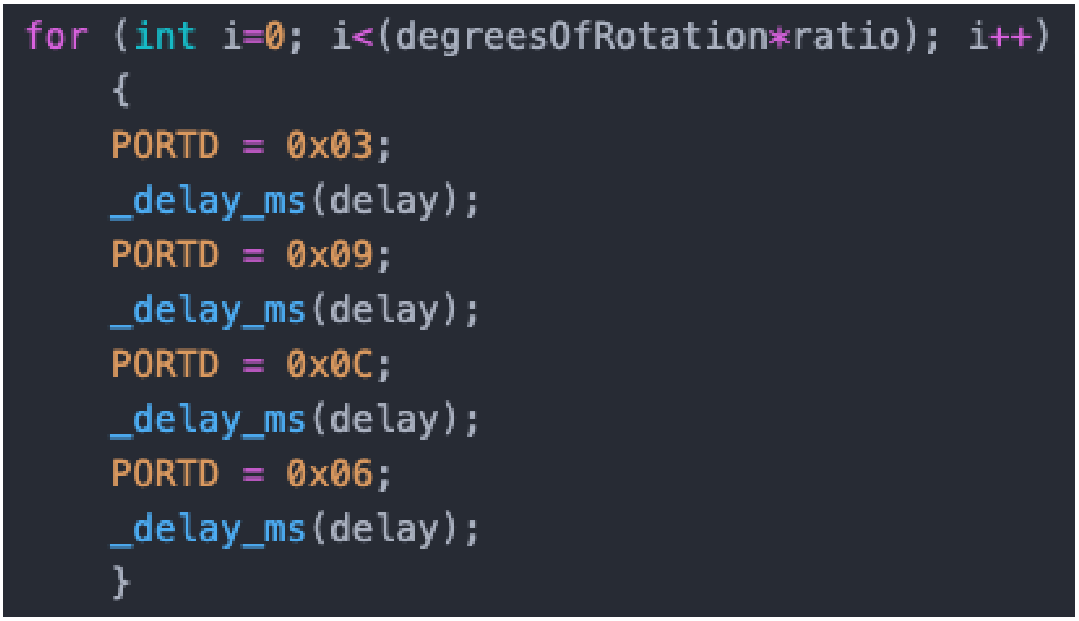
Stepper Motor and Driver: Panoramic Camera Attachment

For the basic chip project, I interfaced with the 28BYJ48 stepper motor along with the ULN2003 driver board. For the project, I decided I wanted to make an electronic camera attachment which can be used for panoramic videos and photos. These devices are sold on Amazon, running anywhere from $30-$100. Using the Arduino Uno, I was able to make my own version of this device for less than $30, including the price of the Arduino itself.



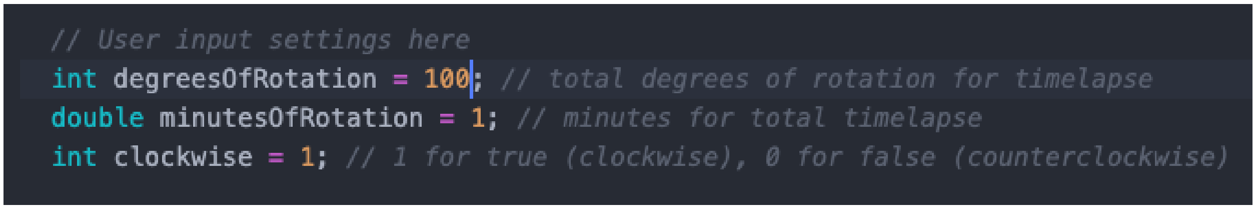
Wiring the stepper motor and driver to the Arduino Uno was quite simple. As shown in the picture on the left, there are four wires running from the output pins to the driver board, and two wires which are running from power and ground to the board. The stepper motor is then just plugged directly into the driver board. My setup is the same as the configuration shown in the picture, except I used pins 0 through 3 instead of 8 through 11 for the output pins.

Writing the code for the stepper motor in the C programming language was the most challenging part of this project. I began with figuring out how to rotate the stepper motor in one direction, which I did by sending a series of hexadecimal numbers to the port controlling the output pins. For example, the hex value “0x03” is equivalent to “1001” in binary, so this value would send power to pins 0 and 3. By cycling through specific hex values, we can rotate the motor with incremental “steps”.



```
for (int i=0; i<(degreesOfRotation*ratio); i++)
    {
    PORTD = 0x03;
    _delay_ms(delay);
    PORTD = 0x09;
    _delay_ms(delay);
    PORTD = 0x0C;
    _delay_ms(delay);
    PORTD = 0x06;
    _delay_ms(delay);
    }
```

Next, I did calculations to figure out how many steps were needed to rotate a set number of degrees, and more calculations to find the correct delay to make the rotation happen over the correct time period. I also created a variable that allows the user to decide between clockwise or counterclockwise rotation. In the end, I had created a program that allowed the user to decide the degrees of rotation, time of the rotation in minutes, and the direction of the rotation for their panorama.



```
// User input settings here
int degreesOfRotation = 100; // total degrees of rotation for timelapse
double minutesOfRotation = 1; // minutes for total timelapse
int clockwise = 1; // 1 for true (clockwise), 0 for false (counterclockwise)
```

This project was daunting at first but once I fully understood how the stepper motor was operated, it was straight forward from there. After some debugging I was left with a final product that exceeded the goals I had set for myself.