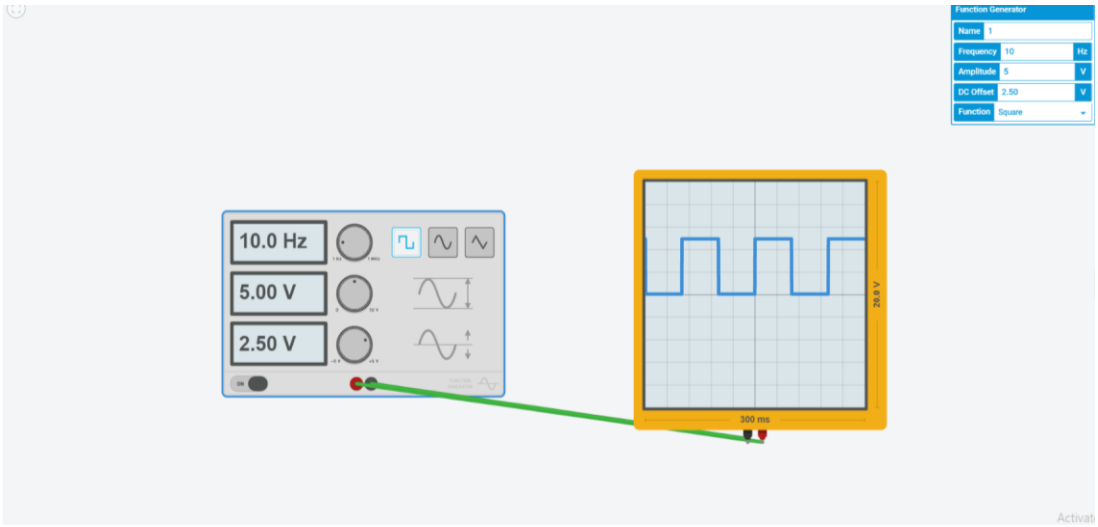
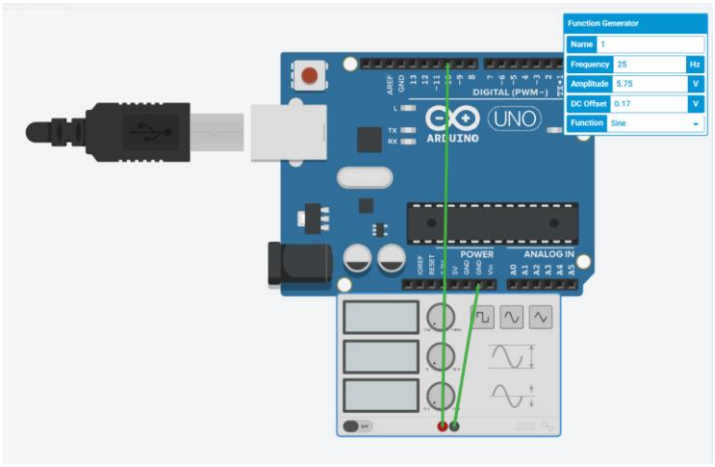
LAB REPORT #8

(Names) Kenilkumar Patel, Austin Guiney
(Date) 4/22/2021
Screenshot and components





Function Generator: This has three main settings that are changeable: Frequency, Amplitude, and DC Offset. When connected to an oscilloscope, the visual of the wave made by the three settings of the function generator would show.The function generator also provides three outputs of the wave visuals in rectangular, smooth, or triangular. The amplitude is measured in volt.

Oscilloscope: This is used to display waves visuals affected by the function generator settings.

Summary:

For part 1, I set up the required components, the function generator and the oscilloscope, and connected through the right positive and negative parts. Then, I tried changing the setting of the

generator as I run the simulation. I tried changing around the setting of both components so to find the relation between period and frequency as shown on the oscilloscope.
For part 2, I first copied over the code. Then I connected the arduino to the function generator. I will calculate the frequency and period of the electrical signal, in which frequency is changed by the function generator. When the wave is still high, the input bit of the pin is 1. The wave is low when the input bit is 0. The total length of the period is one cycle of high wave and low wave. The period can be measured by measuring the time after the one cycle minus the time at the start of the cycle. Since the function micros() measures time in microseconds, I had to convert the period to seconds by dividing a million. Frequency is calculated by 1/period, in which I will print frequency and period out with 5 decimal places.
Result: The frequency in the serial monitor was extremely close including the first decimal place, to what was shown on the function generator. This limit could be due to the time function having lag or not accurate enough. I also changed the frequency on the generator, as the frequency increases, the period decreases as per reciprocal function.
Conclusion:
I can now confidently know the reciprocal relation between frequency and period through the use of a function generator and oscilloscope. I also learned the amplitude measured by the generator is actually voltage. Also, since the program runs very slow, printing out the hello line character by character, I decided to make the line into a comment instead. This sped up the program by only printing out the frequency and period. The program also lagged sometimes and stopped working, but deleting the hello line decreased the chance of the program not working. I checked again, there was no need for the double width, the period = time2/1000000 gave a closer answer since the time2 here is from beginning of the cycle to the end of the cycle.

```
/*
  Program to measure the pulse duration of a pulse generator.
 */


// Define function for Pins: int wave = 10;/*
pulse generator input pin */

// the setup routine runs once when you press reset:
void setup() {
        // initialize the digital pins
        pinMode(wave, INPUT);

        // Add a line printed to the serial monitor
        Serial.begin(9600);
        //Serial.println("Hello world.  Watch me measure the frequency.");
}

// the loop routine runs over and over again forever:
void loop() {
```

```
            while (digitalRead(wave)) {} //skip the first positive level
            while (!digitalRead(wave)){} //wait until the next positive
            level long count = 0;  //count how many times it polls
            //Note: this variable is just to see how many times we poll, it shouldn't be used in calculations

            unsigned long start_time = micros();  //measure the current time in usec

/* Your code should look something like this:
            *
            *           while (wave_still_high)
            *           increment count
            *           read_time
            *           while (wave_still_low)
            *           increment count
            *           read_time_again
            *
            *           calculate the period in seconds
            *           calculate the frequency in seconds
            */
  double time1, time2, period, frequency, width, width2;
    while (digitalRead(wave)) {

    }
            time1 = micros() - start_time;
            while (!digitalRead(wave)){}
            time2 = micros() - start_time;
  //width2 = (time2 - time1); width = (time2
          - time1)/ 1000000;
    period = time2/1000000;
            //period = (time2 + width) / 1000000;
            frequency = 1.00/period;

             Serial.println(frequency, 5); //the second argument shows more decimal places
            //Serial.println(period, 5);
  //Serial.println(start_time, 5);
  //Serial.println(time1, 5);
  //Serial.println(time2, 5);
  //Serial.println(width2, 5); delay(5000); //wait 5 seconds
          and do the test again

}
```