1) The process function $V(n)$ of a PID controller maps the control variable $n$ to the process variable $V$. That is $V=V(n)$. You will make a PID controller to control the average brightness of an LED as measured by a photoresistor. In this system the process variable $V$ will be the measured average brightness of the LED in counts 0-1023, and the control variable $n$ will be the value sent to `analogWrite()` to change the brightness of the LED. To begin you will measure the process function $V(n)$. From lab 3 and 4 we know that the LED driven by `analogWrite()` is pulse-width-modulated (pwm). That means that it is actually turning on and off, but at a high enough rate to not be noticed by our eyes. In lab you found that the frequency is $\sim 490\text{Hz}$. Use the function `getPhoto()` (below) to measure the average brightness. `getPhoto(na,dta)` takes 2 arguments. `na` is the number of averages and `dta` is the sampling time. The exact period of the pwm output is 2040us so we take 15 samples at 136us each to give 2040us total. By averaging over exactly one cycle we eliminate some noise. Briefly explain why?

a. Use `getPhoto()` to measure and plot $V=V(n)$ for every value of $n$ [0,255], where $n$ is the input to `analogWrite(n)` and $V$ is the value of `getPhoto()`. You should pause about 100ms between changing `analogWrite()` and measuring the brightness using `getPhoto()`.

```c
float getPhoto(int na = 15, int dta = 136) {
    int n;
    unsigned long dt;
    float vS = 0;

    for (n = 0, dt = micros(); n < na; n++) {
        while (micros() - dt < dta);
        dt = micros();
        vS += analogRead(inPin);
    }
    return (vS / na);
}
```

b. From the plot of the process function $V(n)$ estimate the maximum value of the variable $P$ in a proportional controller given by the equation $n=P*e$, where $e=V_{set}-V$ is the error and $V_{set}$ is the control set point. Use $V_{set}=V(10)$ for the estimate.