1. Finish the calculation of a PID controller design based on the following table, by fill in the blanks in this table.

	t=0	t=1	t=2	t=3	t=4	
e(t)	1.0	1.3	0.7	0.2	-0.1	
Derivative of e(t)						
Integration of e(t) squared						

Note: use central difference for the derivative calculation.

2. Write Python (or C/C++) program to perform the computation of the above, assume:
(2.1) the error is stored in a text file, error.txt, then your program prompt the user for the input;
(2.2) then perform computation, print out the time index and then derivative of error and integral of error for each time time, such as

Time = 3, Derivative of E = xxx, Integration of E = xxx

(2.3) run the Python or C/C++ code to test GPIO, and capture the screen.

3. Draw a PID Controller with LSM 303 Sensor in the loop, and
(3.1) with motor drive of your choice in the loop and
(3.2) stepper motor in the loop.

4. Modify a menuconfig script to add an item for your device drive with your name, CMPE242 class, and the date, as shown in Figure 1 (bottom of the Figure with Redmark), then Run $make menuconfig and capture the screen similar as in the Figure.
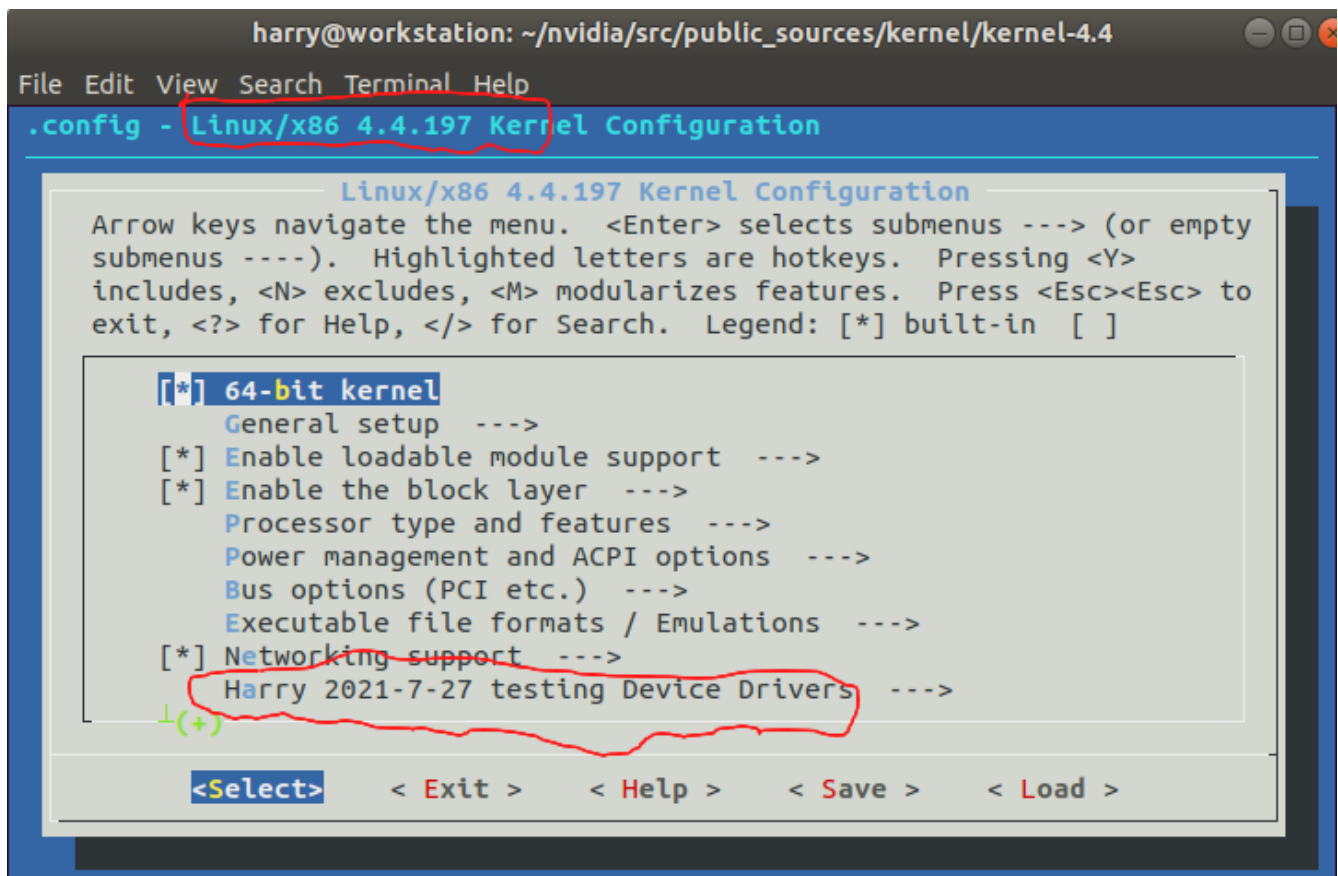
Figure 1.

5. Based on the class material, write a Python or C/C++ to generate PWM output,
(5.1) with frequency of 1000 Hz, duty cycle 20%;
(5.2) run your program, capture the computer screen;
(5.3) Option: use Oscilloscope to capture the PWM waveform like in Figure 2.

```
cd pwm0
echo 5000000 > period
echo 2500000 > duty_cycle
echo 1 > enable
```

Define as in Hz

Output high
defined as in Hz

y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
y@harry-deskt
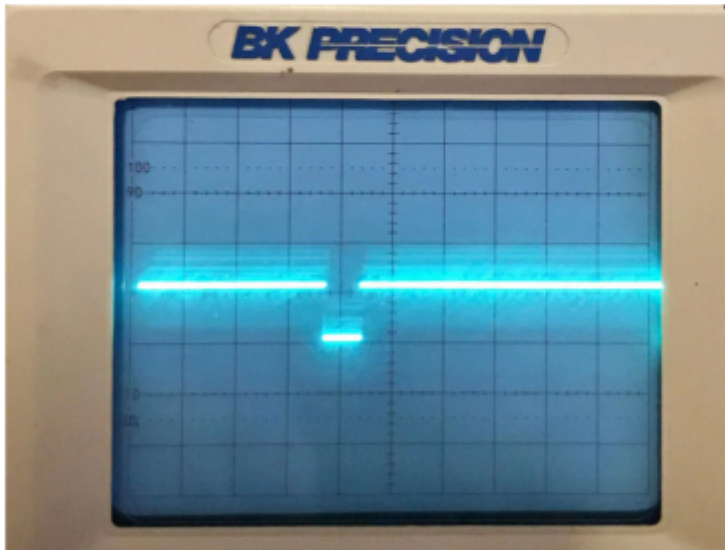y@harry-deskt
y@harry-deskt

BK PRECISION

Figure 2.

6. Combine the above 1-4 into one PDF file, then zip it (except the video clip, please make it a stand lone file, then zip it together with the pdf file to form one zip file). Use the following file naming convention:
firstName_lastName_SID(last-4-digits)_hw_pid_pwm.pdf.
Submit it to the class canvas.

(END)