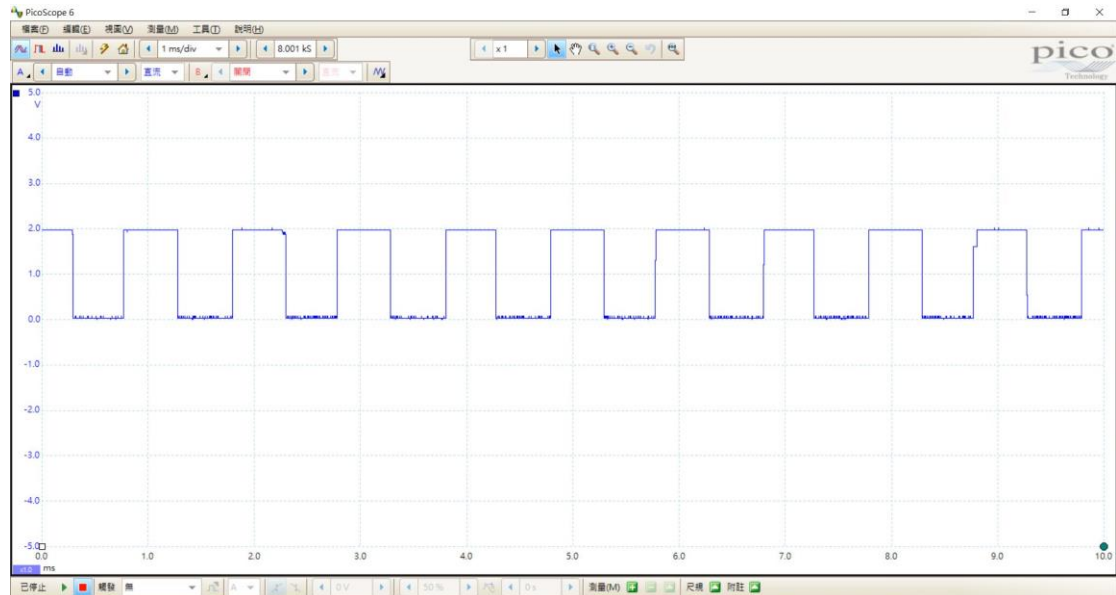


In this lab, I learned ADC work in mbed system, how PWM output effect LED, frequency analyze by FFT, sample rate & Nyquist theorem.

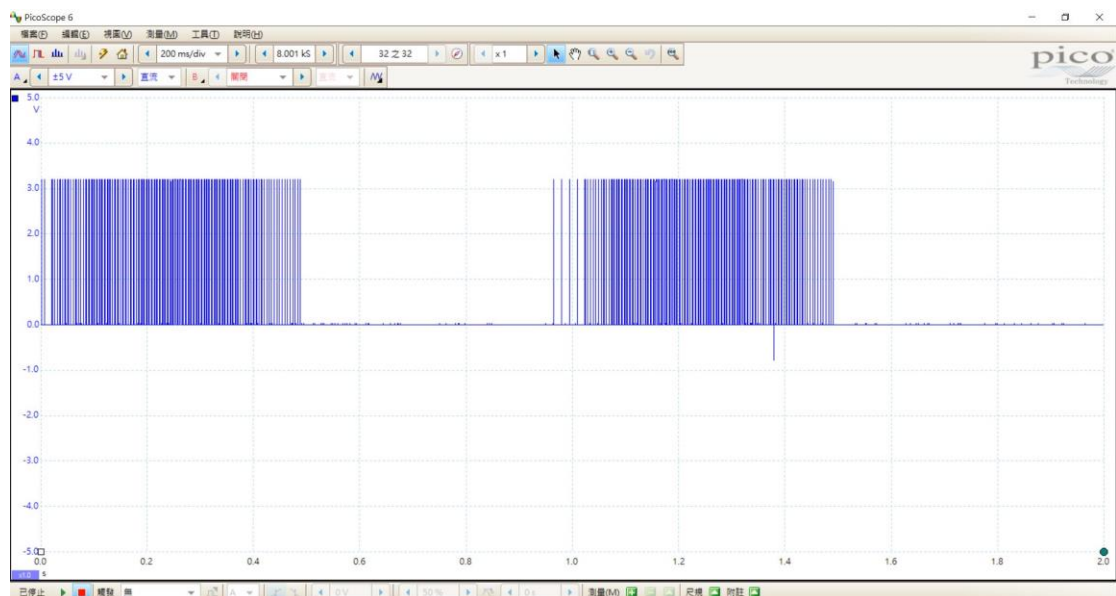
(1) Lab4 1 LED Analog

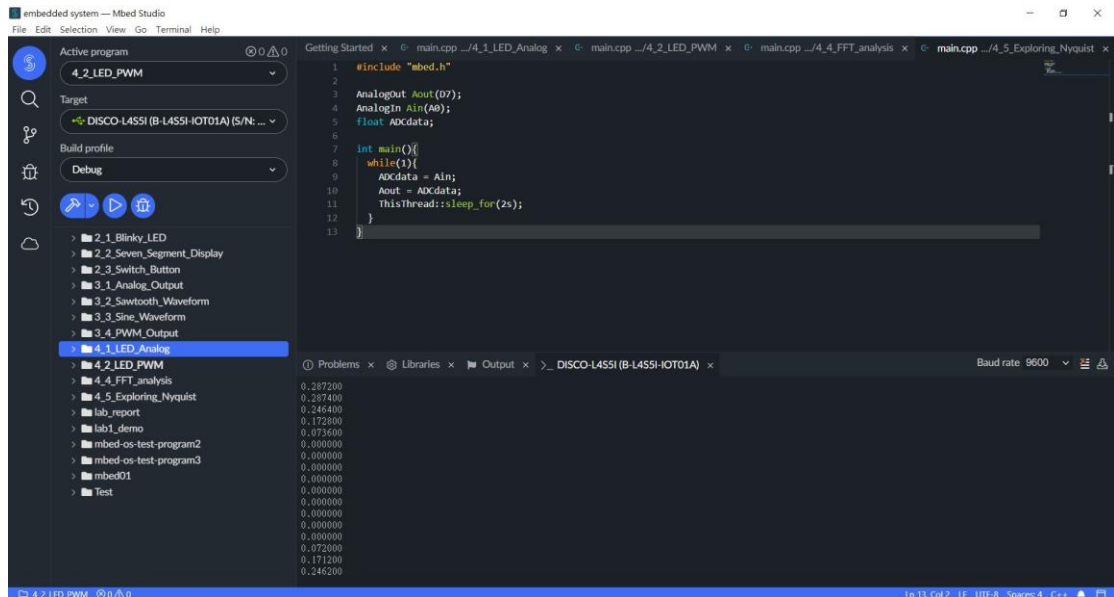
In this lab we use ADC read in analog signal created by Picoscope and created analog signal by DAC to trigger LED.



(2) Lab4 2 LED PWM

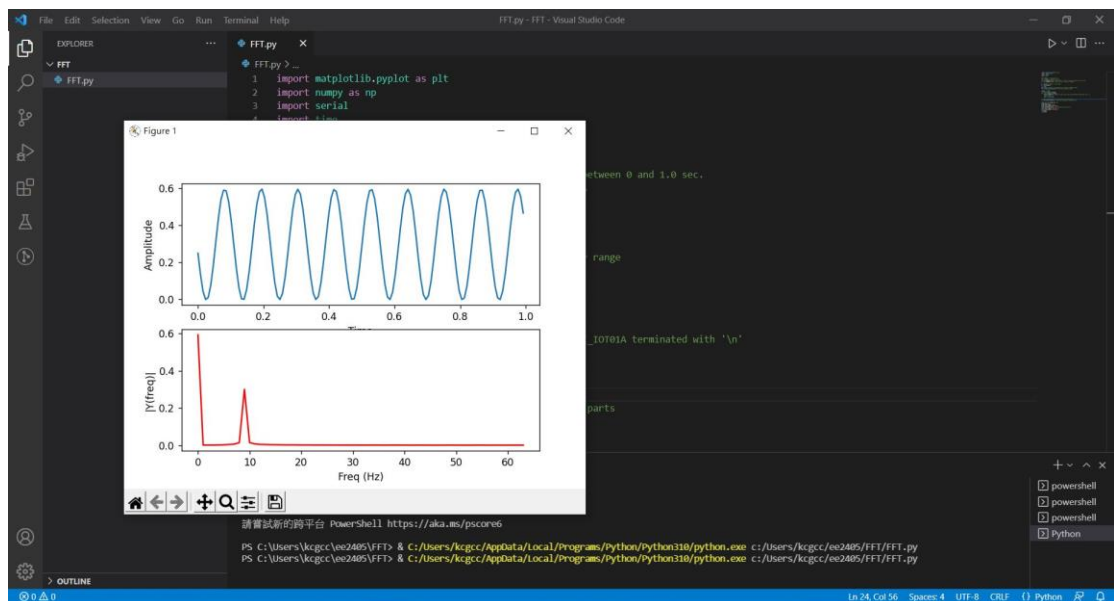
In this lab, we see how PWM affect LED, with different waveform length we can make a breathing LED.

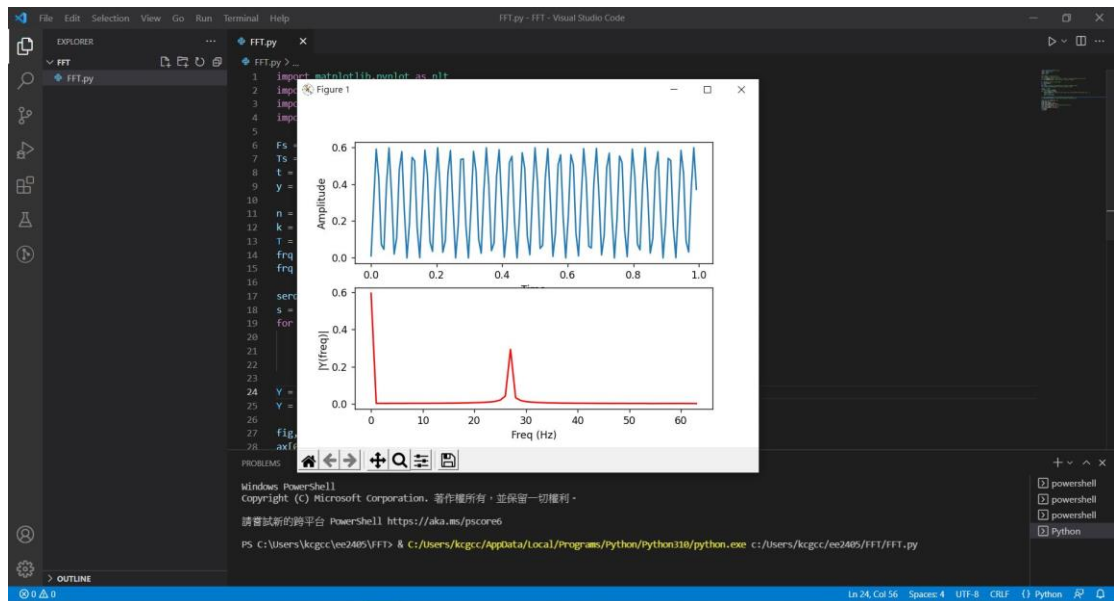




(3) Lab4 4 FFT analysis

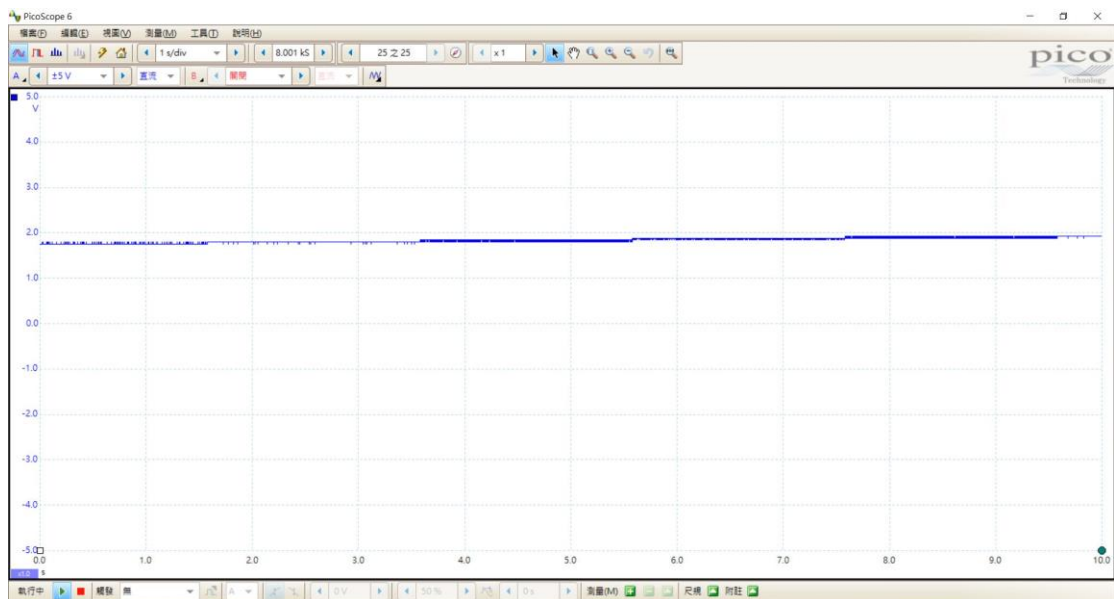
In this lab we use python with FFT function to analysis signal we read in, analyze the frequency of the signal, and by applying different frequency signal created by picoscope, we can see different figure on signal domain.

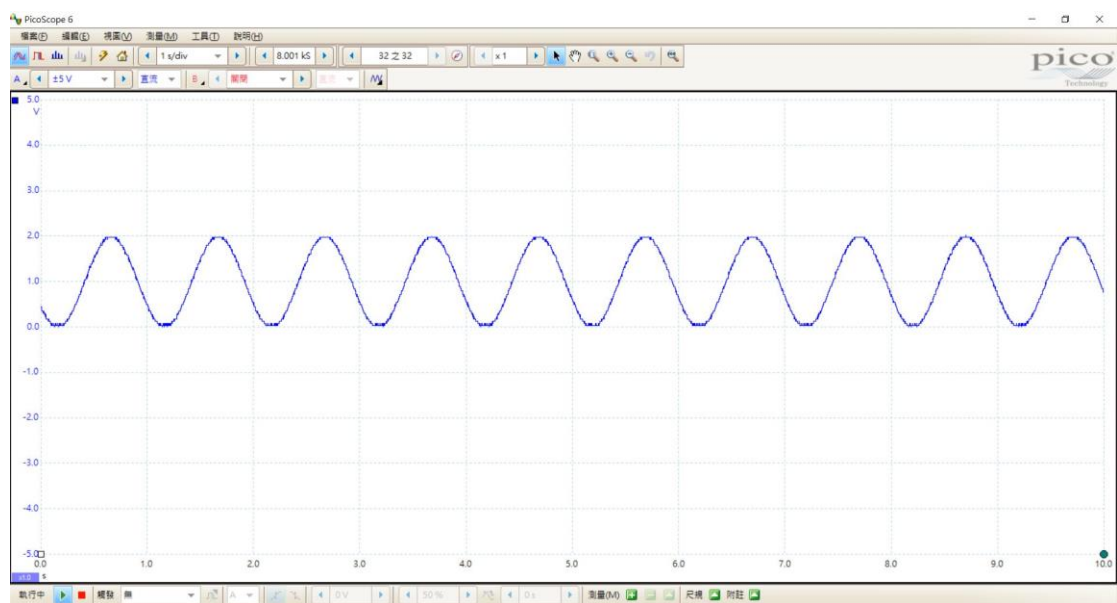
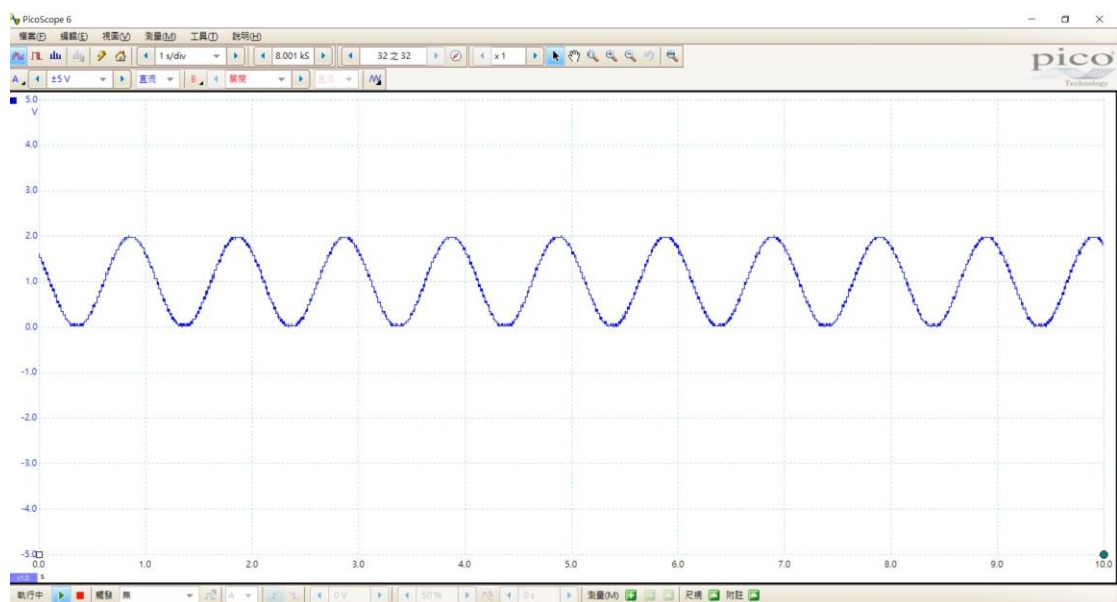
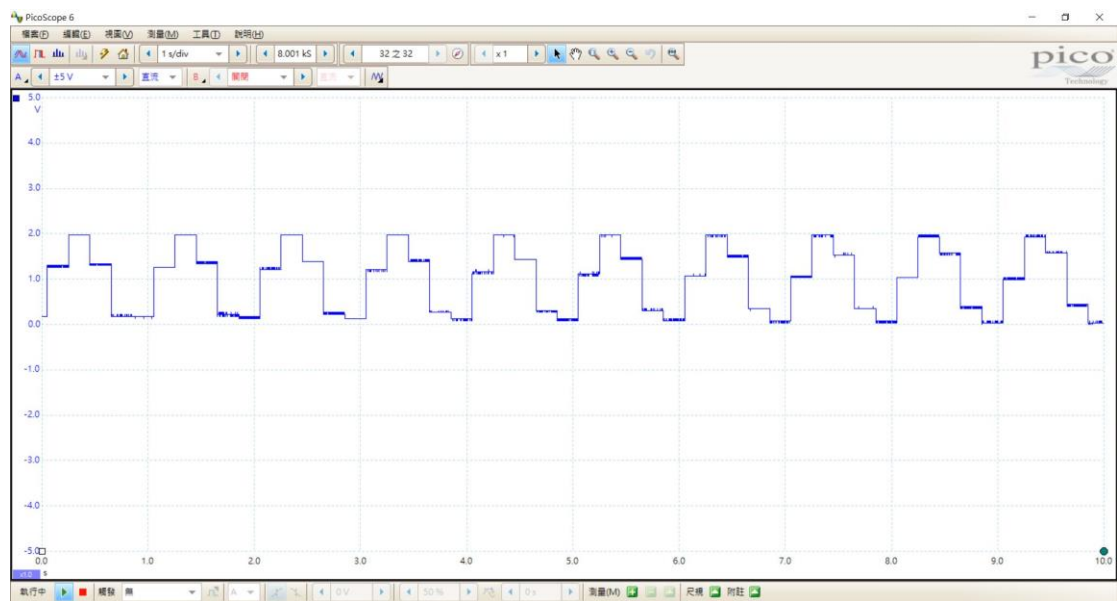




(4) Lab4 5 Exploring Nyquist

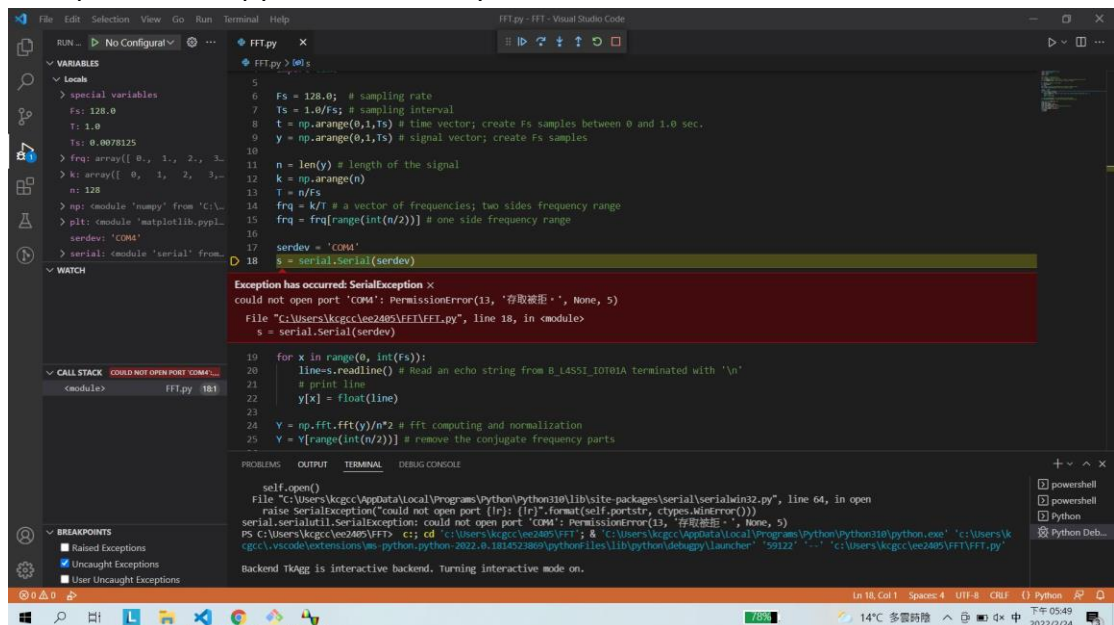
In this lab we see how sample rate can affect the signal, with higher sample rate we can have more precise waveform.





(5) Discussion

In lab 4_4 FFT analysis, we need to press reset button to reset the mbed system, or Python will stuck and won't work, I think it's because after we boot program into mbed, it will instantly read in signal and output signal, which is too fast for us to implement FFT.py and successfully read in data which will lead to a stuck.



```
5
6 Fs = 128.0; # sampling rate
7 Ts = 1.0/Fs; # sampling interval
8 t = np.arange(0,1,Ts) # time vector; create Fs samples between 0 and 1.0 sec.
9 y = np.arange(0,1,Ts) # signal vector; create Fs samples
10
11 n = len(y) # length of the signal
12 k = np.arange(n)
13 T = n/Fs
14 frq = k/T # a vector of frequencies; two sides frequency range
15 frq = frq[range(int(n/2))] # one side frequency range
16
17 serdev = 'COM4'
18 s = serial.Serial(serdev)
19
20 for x in range(0, int(Fs)):
21     line=s.readline() # Read an echo string from B_L455I_IOT01A terminated with '\n'
22     # print line
23     y[x] = float(line)
24
25 Y = np.fft.fft(y)/n*2 # fft computing and normalization
26 Y = Y[range(int(n/2))] # remove the conjugate frequency parts
```

Exception has occurred: SerialException x
could not open port 'COM4': PermissionError(13, '存取被拒', None, 5)
File "C:\Users\kgccc\ee2405\FFT\FFT.py", line 18, in <module>
s = serial.Serial(serdev)

self.open()
File "C:\Users\kgccc\AppData\Local\Programs\Python\Python310\lib\site-packages\serial\serialwin32.py", line 64, in open
raise SerialException("could not open port (%r): (%r)" % (self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM4': PermissionError(13, '存取被拒', None, 5)
PS C:\Users\kgccc\ee2405\FFT> cd 'c:\Users\kgccc\ee2405\FFT'; & 'c:\Users\kgccc\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\kgccc\vscode\extensions\ms-python.python-2022.0.181452\pythonfiles\lib\python\debugpy\launcher' '59122' '-.' 'c:\Users\kgccc\ee2405\FFT\FFT.py'

Backend TRAgg is interactive backend. Turning interactive mode on.

Besides that, I had confronted some Port issue and I use coolterm to check if the Port is working properly or not.