

## DSP Lab 1: Introduction to Arduino Platform

電機 18 林楷平 103061109

1.

### a. Specification:

See the Arduino training manual and finish the labs till class 7。

自己學習並熟悉 arduino 的使用到 class 7。

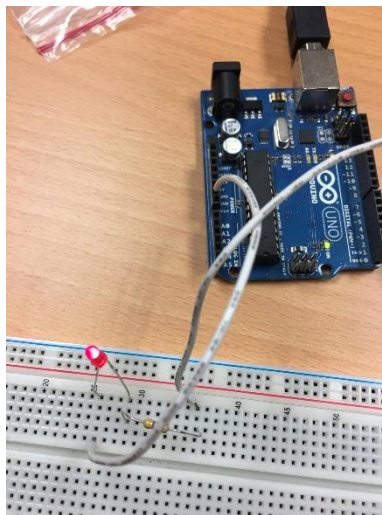
### b. Implementation:

class 1: 認識 arduino 板子與基本操作，認識一些規格及元件，並教一個簡單的程式。

Class 2: 認識 arduino 基本的程式語言，他的語言與 C 語言類似，所以很好上手，其中要注意的就是一些資料傳輸的指令，如 **Serial**。

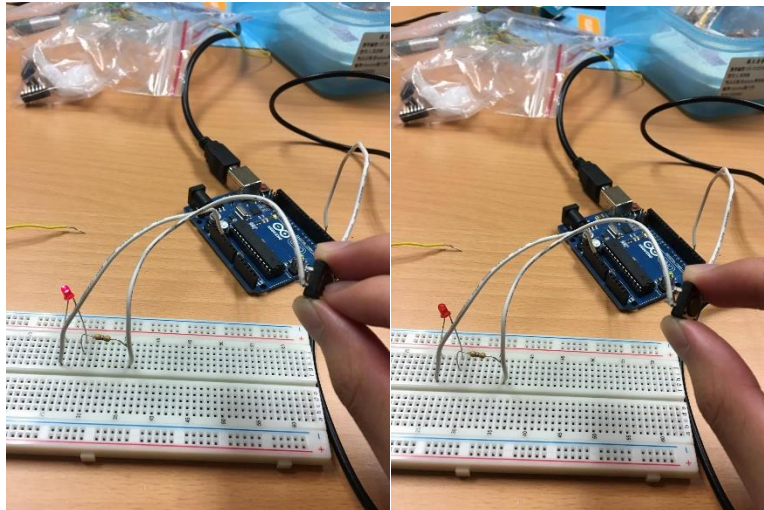
Class 3: 這個 class 就是 class 2 的延伸，使用的一些語法也與 C 語言相同，而這個 class 還有介紹基本 LED 的操作。

Class 4: 利用 arduino 以及 LED、麵包板、電阻等幾個簡單的元件，做出一些 LED 的功能，而我們自己有組個簡單的 LED 的電路實驗。



利用板子控制 LED 閃爍的頻率。

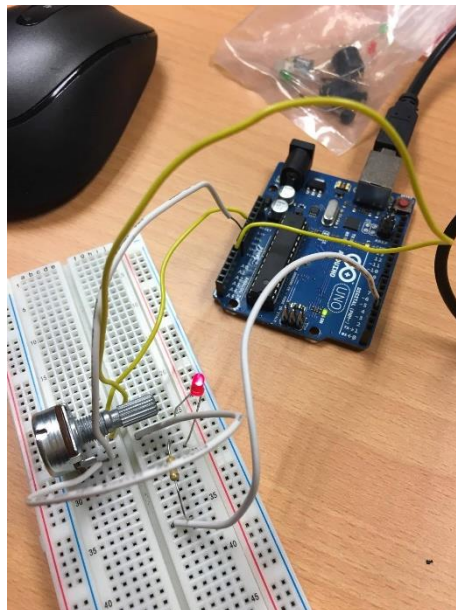
Class 5: 利用按鍵開關來決定該線路是否導通，並介紹按鍵電路是如何操作，我們也實際操作這個小實驗。



可以看到左圖按下開關時，LED 亮起，右圖沒按按鍵，LED 就熄滅。

**Class 6:** 這個 class 介紹數位以及類比的訊號，並且也有範例利用類比訊號(使用 PWM)的方式來控制 LED 的亮度。

**Class 7:** 這個 Class 是教我們使用可變電阻來調整光，剛開始我們認識這個可變電阻的每個 PIN 腳，後來我們也有利用簡單的電路來觀察是否可行。



### **c. Discussion:**

這題算是最入門的題目，沒有什麼壓力跟難度，也沒有遇到什麼問題，本子介紹的之前在 822 實驗室也都碰過了，所以簡單接幾個電路操作也很快就上手了，遇到比較大的應該是之後用實驗室燒成是到板子遇到很多問題，USB 恐接觸不良之類的。

2.

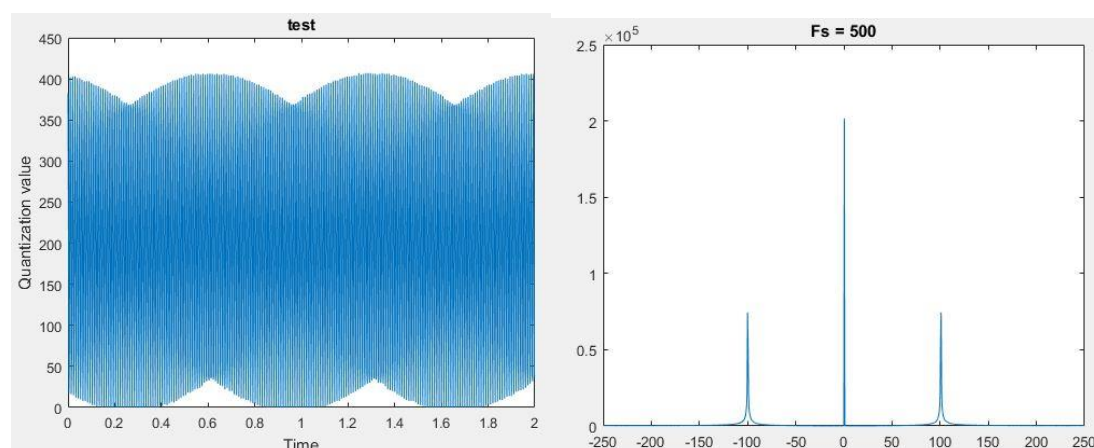
### a. Specification:

(Sampling and Aliasing) Connect the function generator (Picoscope, see PICOSCOPE.pdf) to the ADC of the Arduino board. Set sine wave as the ADC input waveform with frequency of 100 Hz. Use the following frequencies (500 Hz, 200 Hz, 100 Hz, 80 Hz) to sample the input waveform.

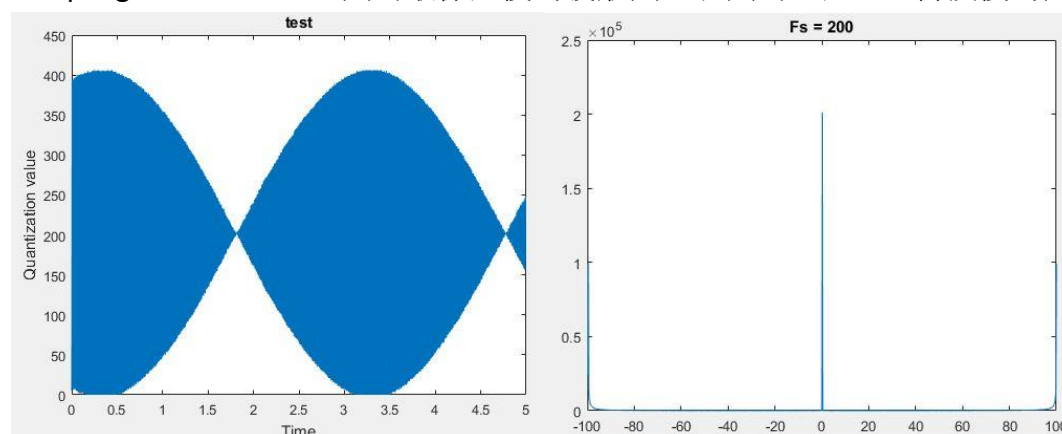
這題是要 Picoscope 產生一個 100Hz 的正弦波，經過 Arduino 上的 ADC 轉換，再利用 Matlab 的不同取樣頻率做取樣，觀察不同取樣頻率所產生的現象為何、FFT 後的圖形為何，並且觀察是否有混疊現象(aliasing)產生。

### b. Implementation:

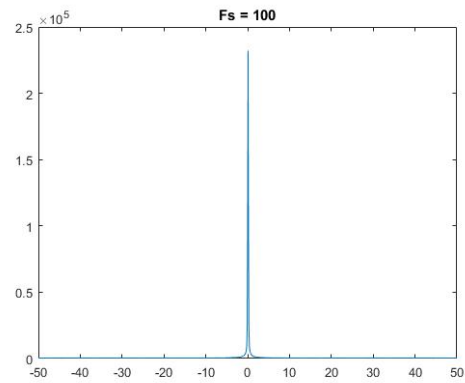
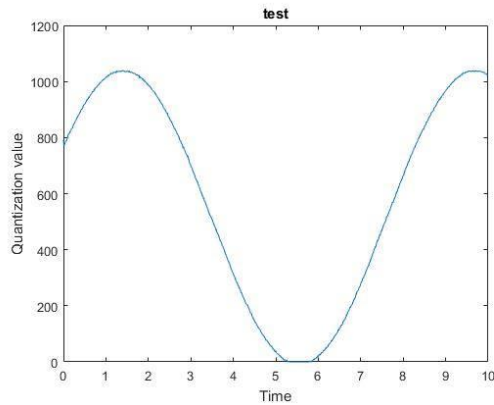
Sampling Rate: 500Hz 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



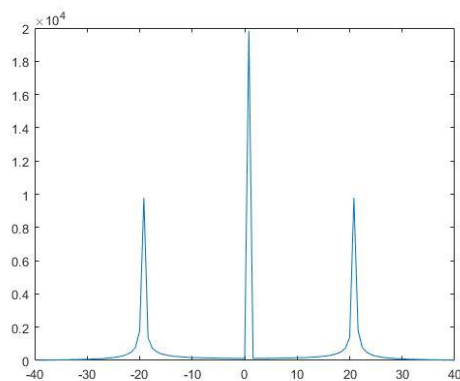
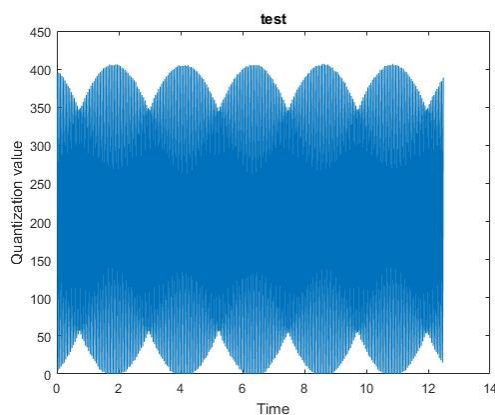
Sampling Rate: 200Hz 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



Sampling Rate: 100Hz 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



Sampling Rate: 80Hz 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖

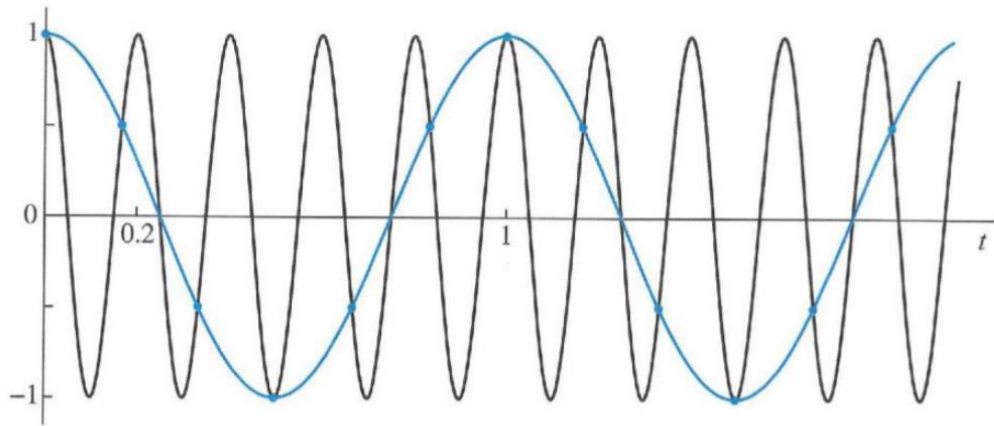


### c. Discussion:

(1) 從上面幾張波型圖還有我觀察 Sampling Rate=500Hz 的 FFT 圖，可以發現整個 sine 波不是落在完美 100Hz，這是因為我們所使用的 Picoscope 還有 adurino 板子，並不會打出完美的 100Hz 訊號，雖然我們可以設定出 100Hz 的 sine 波，不過可能出來的波會有誤差，而 adurino 板也可能會有這個問題，不能發出完美的 100Hz 或 sine 波，所以從 FFT 中可以直接看出，並不是完美的直線。

(2) 從上圖每個 FFT 圖可以看到，在頻率為 0 的地方都會有 value，所以我跟我的 partner 討論後，覺得是因為我們在 Picoscope 6 軟體的介面中，有設定 sine 波有一個 offset 1V，所以這個 offset 經過 Matlab FFT 的轉換，會在頻率 0 的地方出現值。

(3)



**Figure 6.12** Demonstration of aliasing in sinusoidal signals. The signals  $x(t) = \cos(2\pi F_1 t)$ ,  $F_1 = 1$  Hz, and  $x_2(t) = \cos(2\pi F_2 t)$ ,  $F_2 = 5$  Hz, sampled at a rate  $F_s = 6$  Hz generate the same set of samples. The ideal DAC reconstructs the sinusoid whose frequency is in the fundamental range.

$$F_1 = F_s - F_2 = 6 - 5 = 1 \text{ Hz}$$

(圖片來自李夢麟老師 DSP 概論講義)

從上圖可以看出，若取樣頻率不夠高時，可能會產生混疊現象(aliasing)，藍色的現就是由於取樣頻率太低，使頻率為 6Hz 的波經過轉換會被看成 1Hz 的波。所以從取樣定理(sampling theorem)來看，取樣頻率要大於等於兩倍的原本樣本的頻率才不 aliasing，所以從上圖幾張 FFT 來討論，我已經把頻率軸的部分換成  $-0.5F_s \sim 0.5F_s$  了以便觀察，100Hz 的 sine 波被取樣，要大於等於 200Hz 才不會產生 aliasing，上圖 Sampling Rate = 500Hz，就可以看出沒有產生 aliasing 的情況發生，原本的波都落在頻率為 0 還有 100Hz 的地方，也就是上述所提的 offset 還有 sine 波的頻率；而 Sampling Rate = 200Hz，剛好是取樣定理的臨界值，可以看到 FFT 圖旁邊兩側有值，也就是 sine 波的頻率；Sampling Rate = 100Hz，已經產生 aliasing 了，所以可以看到所有的頻率都落回頻率 0 的地方，而這樣也合理，因為在 100Hz 的波上用 100Hz 的取樣頻率，只會取到相同的值，不會有變化，頻率也就等於 0；最後 Sampling Rate = 80Hz，可以看到也會產生 aliasing 的現象，所以超過取樣頻率的部分就會折回 fundamental frequency 區間，由上圖可以看到就是  $100 - 80 = 20$ ，就會落在 20Hz 上。

3.

#### a. Specification:

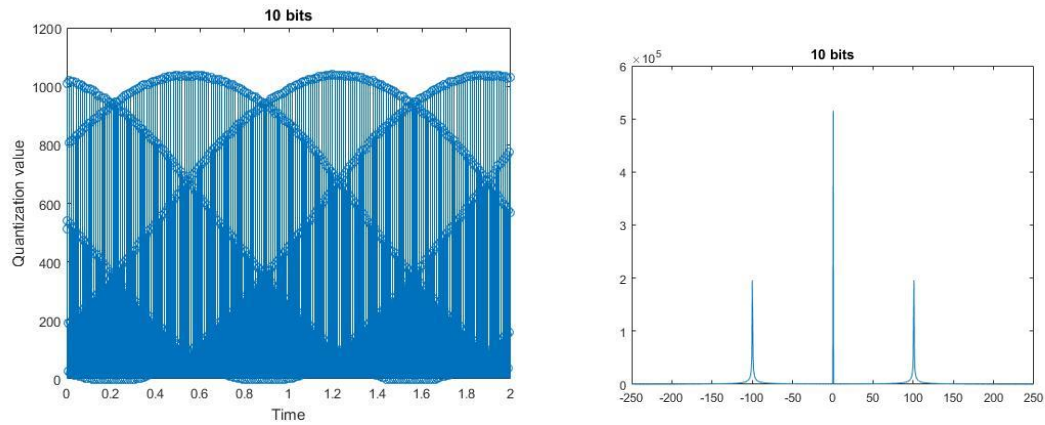
Use 10 bits, 8 bits, 5 bits, and 3 bits to sample the oscillator waveform. 1) Plot the waveforms with the provided Matlab codes. 2) Use FFT to show the spectra of the saved samples

用 10、8、5、3 個 bits 來表達每個波的數值，再利用 matlab 的範例 code 來表達波的形狀，並且印出 FFT 的轉換圖

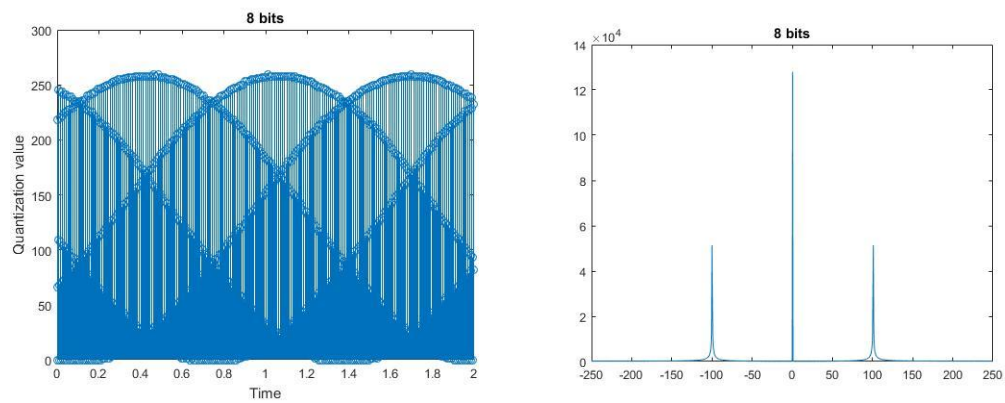


## b. Implementation:

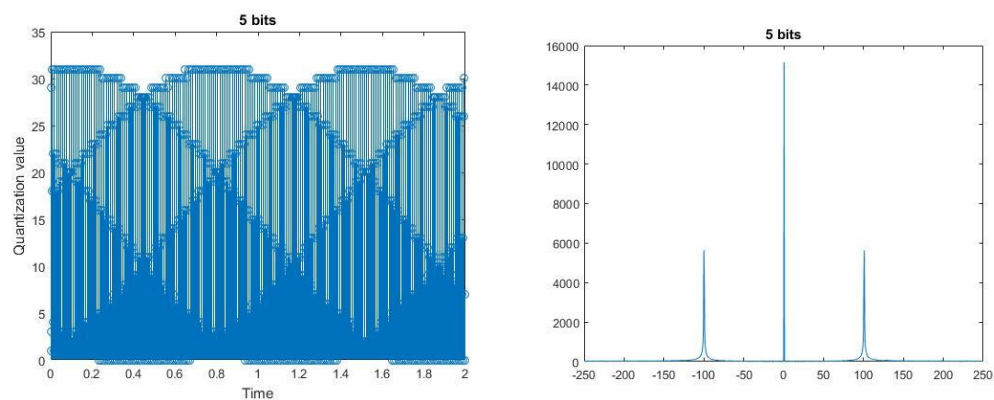
10 bits sample 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



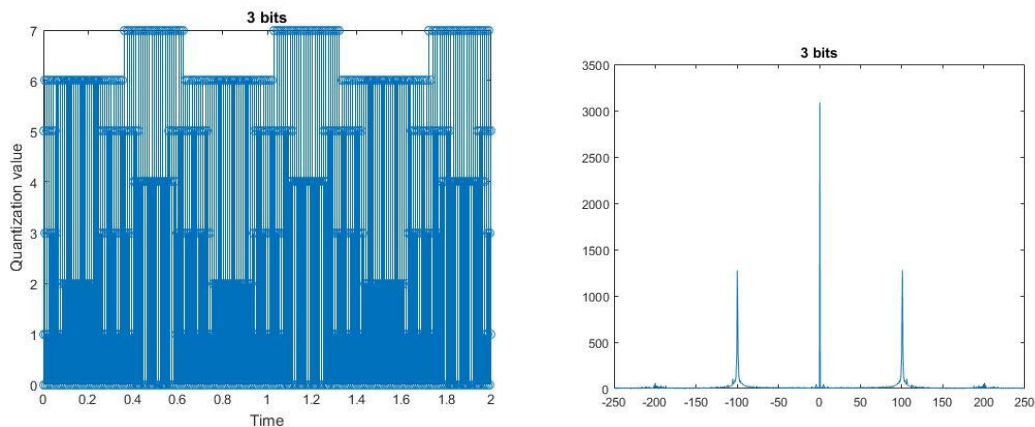
8 bits sample 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



5 bits sample 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



3 bits sample 左圖為取樣過後的波形圖，右圖為經過 FFT 轉換後的圖



### c. Discussion:

(1)雖然 arduino 板子可以供給電壓是 0 到 5V，但 Picoscope 只能供給 0 到 2V，所以在 code 的部分把 0 到 1023 的部分，切成 0 到大約 400，這樣 Picoscope 就可以與 arduino 接上。然後再來是 bit 數的問題，若要 3bit 的話就需要 8 個 level(2 的 3 次方等於 8)，5 bits 就需要 32 個 level，以此類推，我就將 0 到 400 切割成 8 個份，也就是 0 到 7，這樣就可以把波型做 Quantization。

(2)從上圖 3 bits sample 的 FFT 中可以看出，FFT 的波型圖有些毛躁彎曲，我認為這是因為波行被切成 8 個 level，也就是被 Quantization 了，所以數值的變化也變成了有限的排列組合，也就剩下幾個頻率有值而已，在 FFT 的圖中，鋸齒狀的現象也變得比較明顯了。

## 4. Conclusion

這個 LAB 算是比較輕鬆的 LAB，因為之前就有用過這個板子做實驗了，所以對板子不會陌生。而第二題最主要就是要我們用 matlab 來表達訊號的波型，這個我們之前在 DSP 導論也做過類似的作業了，所以也很簡單就上手，並且改了頻率軸上的大小，讓他變成 fundamental frequency；最後一題比較好玩，是要將波型做 Quantization，第一次將訊號轉成數位的模式，也可以清楚看到用比較多 bit 可以比較清出還原原本的波型。

## 5.Reference

- DSP 導論講義、作業
- 認識 Arduino 的書
- 講義上的教學還有提供的 code