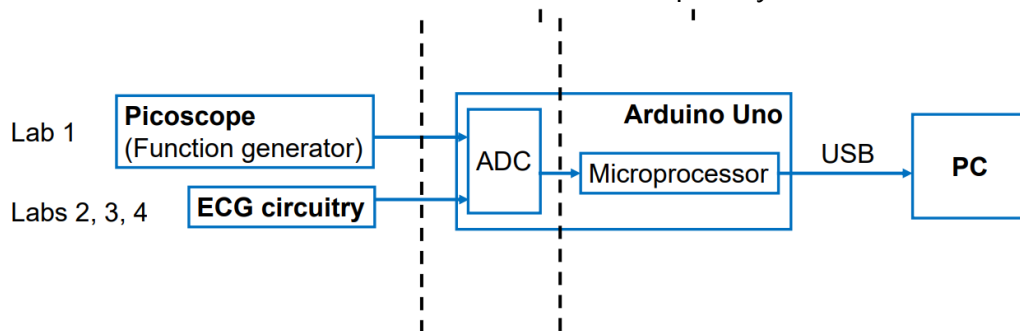


- **Lab Objective**

運用 Arduino 實做一個數位示波器，嘗試各種 sampling rate、Quantization bit，並比較其差異然後討論造成原因。

- **Design Implementation**

訊號傳遞的過程如下圖。首先，我們透過 Picoscope 產生 100Hz 的正弦波，並透過 Arduino 內部的 analogRead 去做取樣，再透過 serial communication 將取樣到的資料傳入電腦中，運用 MATLAB 將量測到的波形畫出來以及進行 FFT 將其 frequency domain 畫出來。



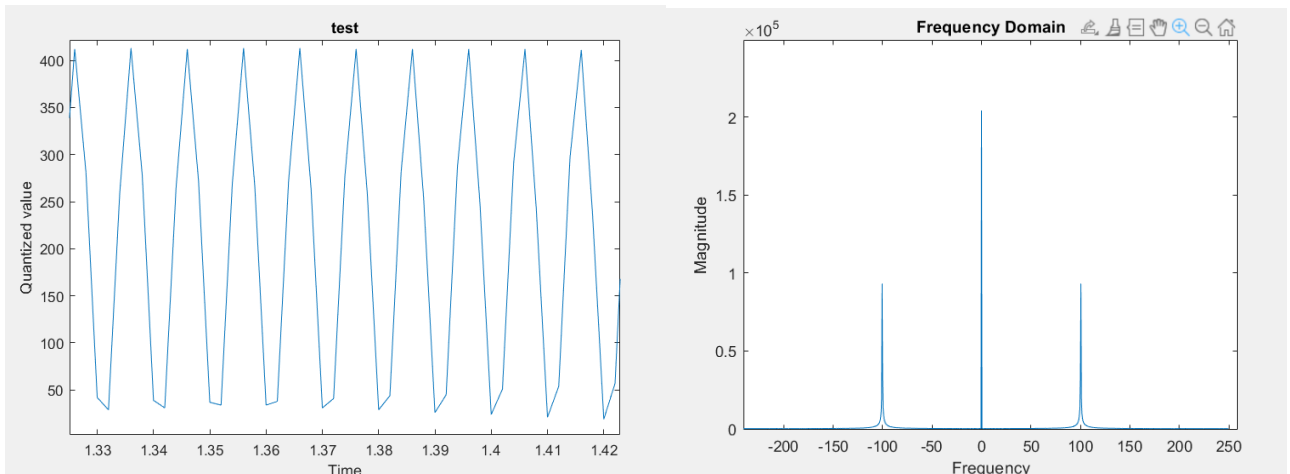
接著我們展示在各種不同的 sampling rate( $f_s$ )以及不同 Quantization Bit 下的量測結果：

### Part 1. Different Sampling Rate ( $f_s$ )

我們可以透過調整 Arduino 程式碼中 Loop 的週期來調整 sampling rate( $f_s$ )。根據 **Sampling Theorem**，取樣後要不產生 aliasing  $\leftrightarrow 2 * f_{max} < f_s$ ，因此理論上來說，這個部份我們應該要在  $f_s < 2 * 100 = 200\text{Hz}$  的 case 中觀察到 aliasing。量測結果如下：

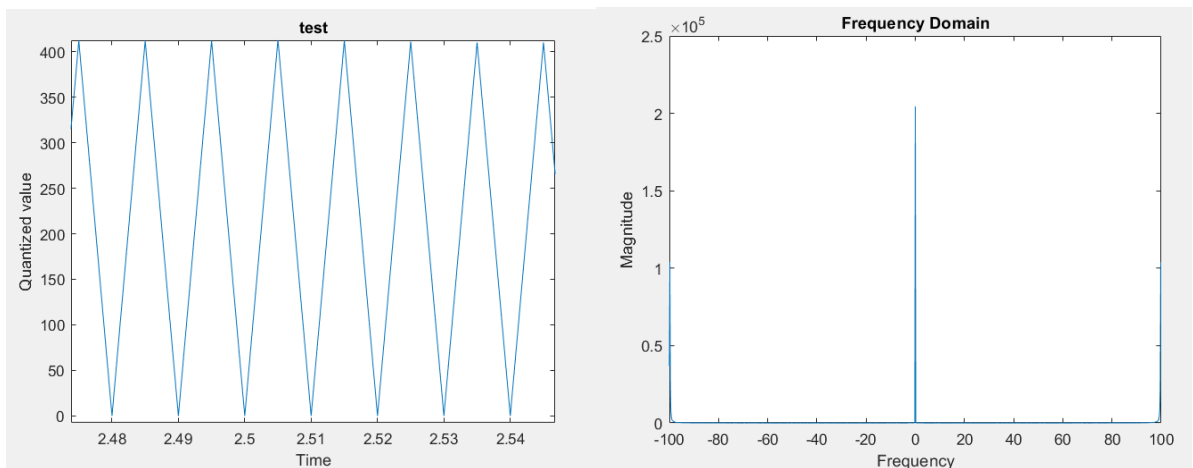
### A. $f_s = 500\text{Hz}$

從 frequency domain 我們可以很清楚的看出他是 100Hz 的正弦波(低頻多一個 impulse 的原因會在下一個部分做討論，這邊先忽略)。另一方面，從波形圖也可以看出他是 100Hz，我們可以很清楚的看到每個波峰之間間隔 0.01s  $\rightarrow$  100Hz



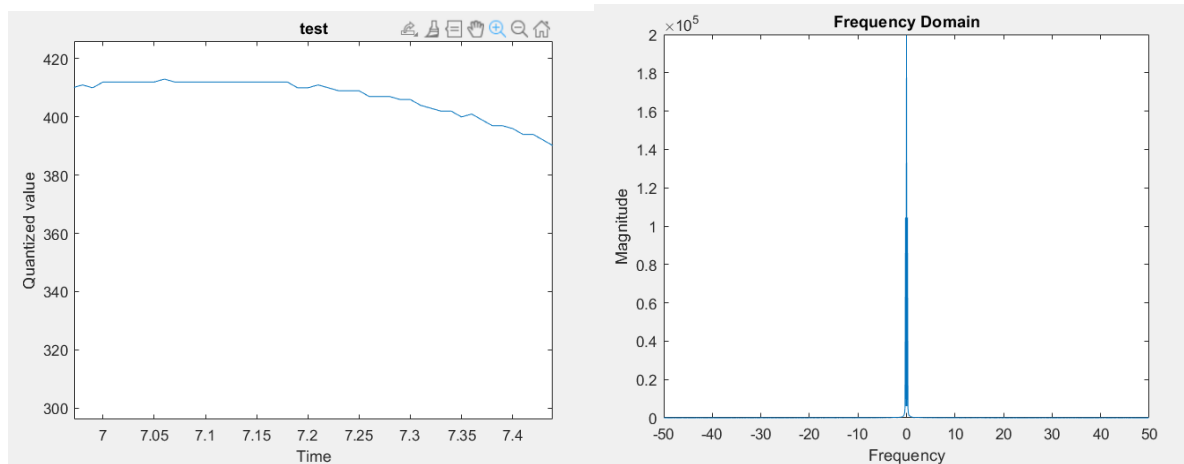
### B. $f_s = 200\text{Hz}$

這個 case 也沒有發生 aliasing，但也可以發先這是一個十分極限的 case，我們取樣到的點基本上就是正弦波的波峰以及波谷。從 frequency domain 也可以看出我們取樣的正弦波 100Hz。

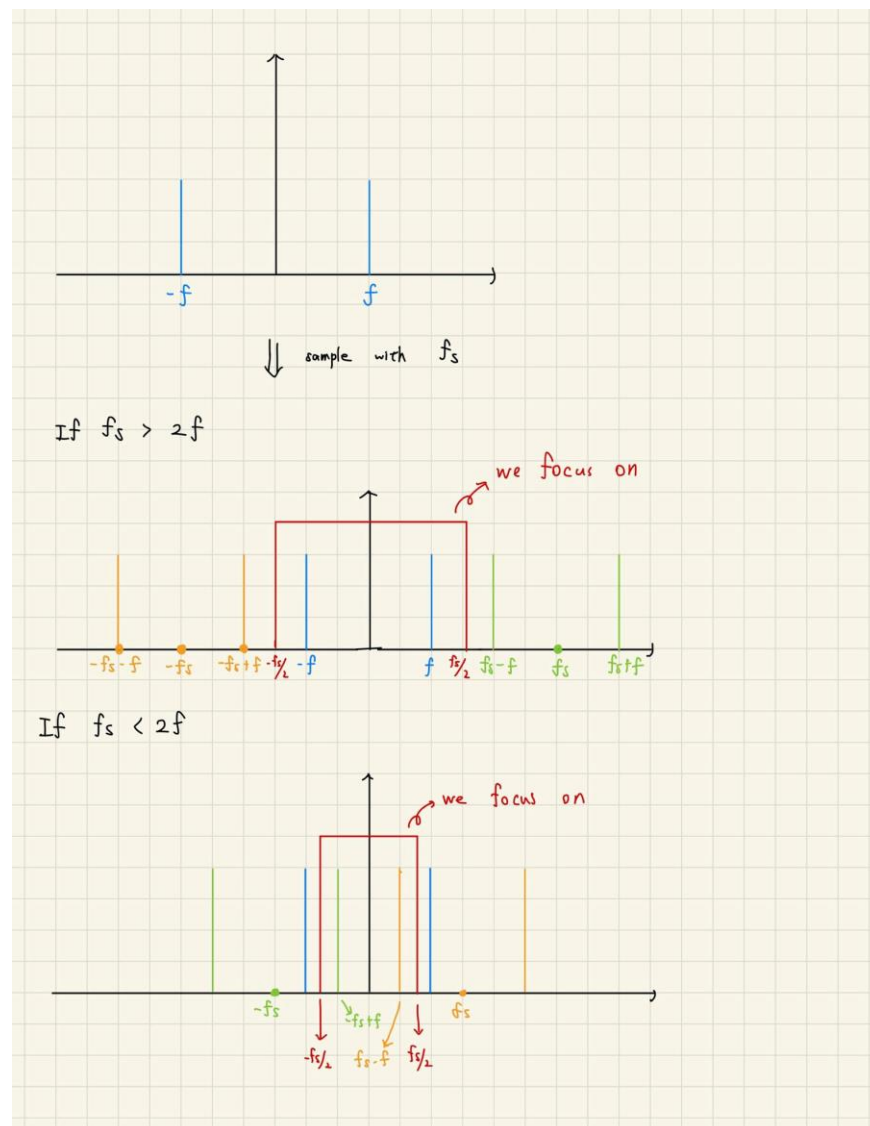


### C. $f_s = 100\text{Hz}$

這個 case 就發生了 aliasing，而且相當嚴重，從波形圖和頻譜可以看出幾乎變成 DC 訊號。

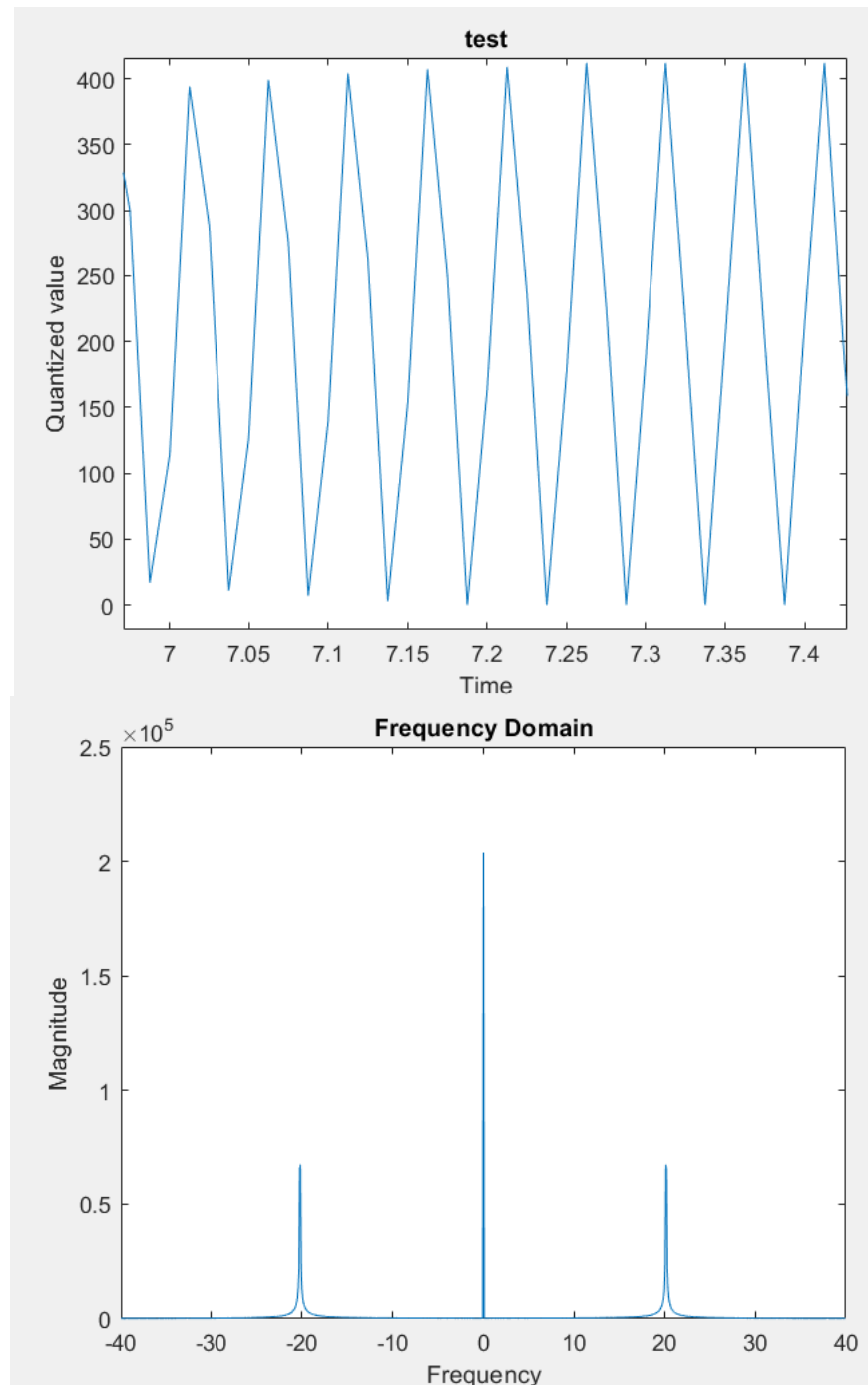


根據下圖，在取樣率為  $100\text{Hz}$  的情況下，原先在頻譜上  $100\text{Hz}$  和  $-100\text{Hz}$  處的 impulse 會被移到  $0\text{Hz}$  附近，因此才會產生取樣後類似 DC 訊號的情形。



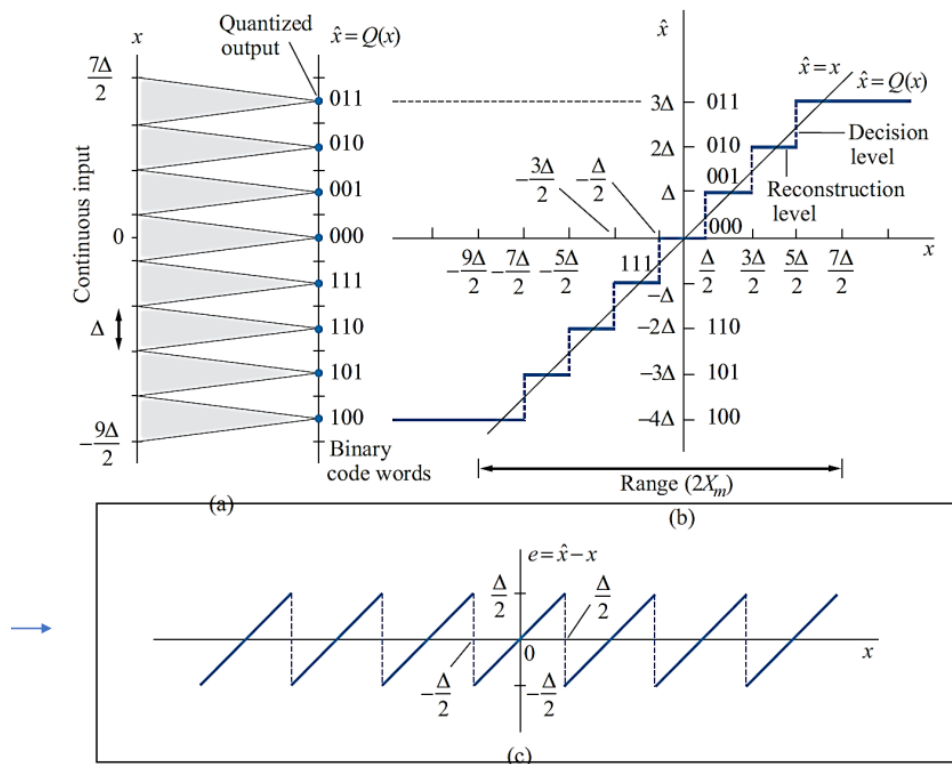
**D.  $f_s = 80\text{Hz}$** 

根據我們前面的推導，這個 case 應該也會發生 aliasing，而且他應該會變成一個  $-80 + 100 = 20\text{Hz}$  的正弦波。而量測結果也如我們預期，頻譜上的 impulse 在 20 和  $-20\text{Hz}$ ，從啾啾行也可以看出波峰間約間隔  $0.05\text{s}$ 。



## Part 2. Different Quantization Bit

接著我們觀察不同數量的 Quantization Bit 在  $f_s = 500\text{Hz}$  (no aliasing case) 下造成的影響。我們先解釋一下 Quantization 會造成的影響。如下圖所示，Quantization 就是將我們可以容忍的區域進行切割，以我們用的 Arduino 為例，AnalogRead 的電壓範圍介於  $0\text{V} \sim 5\text{V}$ ， $N$  bit quantization 就是將  $0 \sim 5\text{V}$  切割成  $2^N$  個區間，將我們量測到的電壓歸類在最接近的區間。很容易可以想到 Quantization 會產生 error，其 error 如(c)所示。

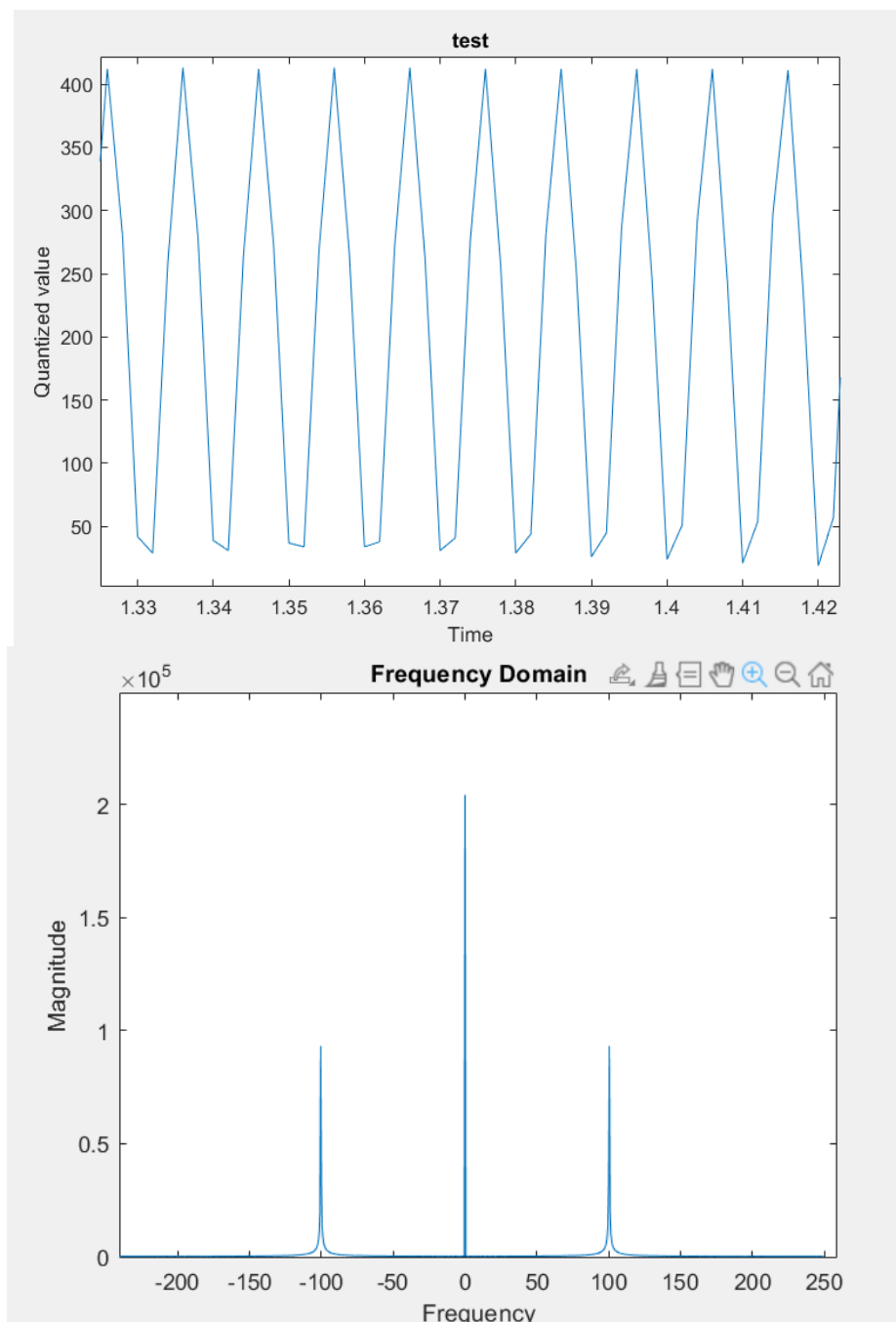


**Figure 6.24** The quantization operation allocates intervals to a number of discrete levels, that is, quantization is a many-to-one mapping. (a) Allocation of levels in a 3-bit quantizer which rounds  $x/\Delta$  to the closest integer. Input-output (b) and quantization error (c) transfer function of a uniform rounding quantizer.

Quantization Noise 就來自這些 error，我們可以將原始的訊號經取樣後得到的  $Y[n]$  想像成  $Y_Q[n] + Y_e[n]$ ，分別代表 quantization 後的訊號以及 quantization noise，也就是說我們傳送的訊號  $Y_Q[n] = Y[n] - Y_e[n]$ ，再經過 FFT 後，頻譜上會不僅有原始訊號的頻率，也會有 Quantization Noise 的頻率。不同的 Quantization 方式會造成  $Y[n]$  和  $Y_e[n]$  之間有不同的 SQNR (Signal-to-quantization-noise-ratio)，如果以上面那張圖的方式為例，經過計算每多一個 Quantization bit 我們可以多大約 6dB 的 SQNR。因此，當 Quantization Bit 越少，Quantization Noise 就越明顯，實際情況可以從以下的兩側結果看出來：

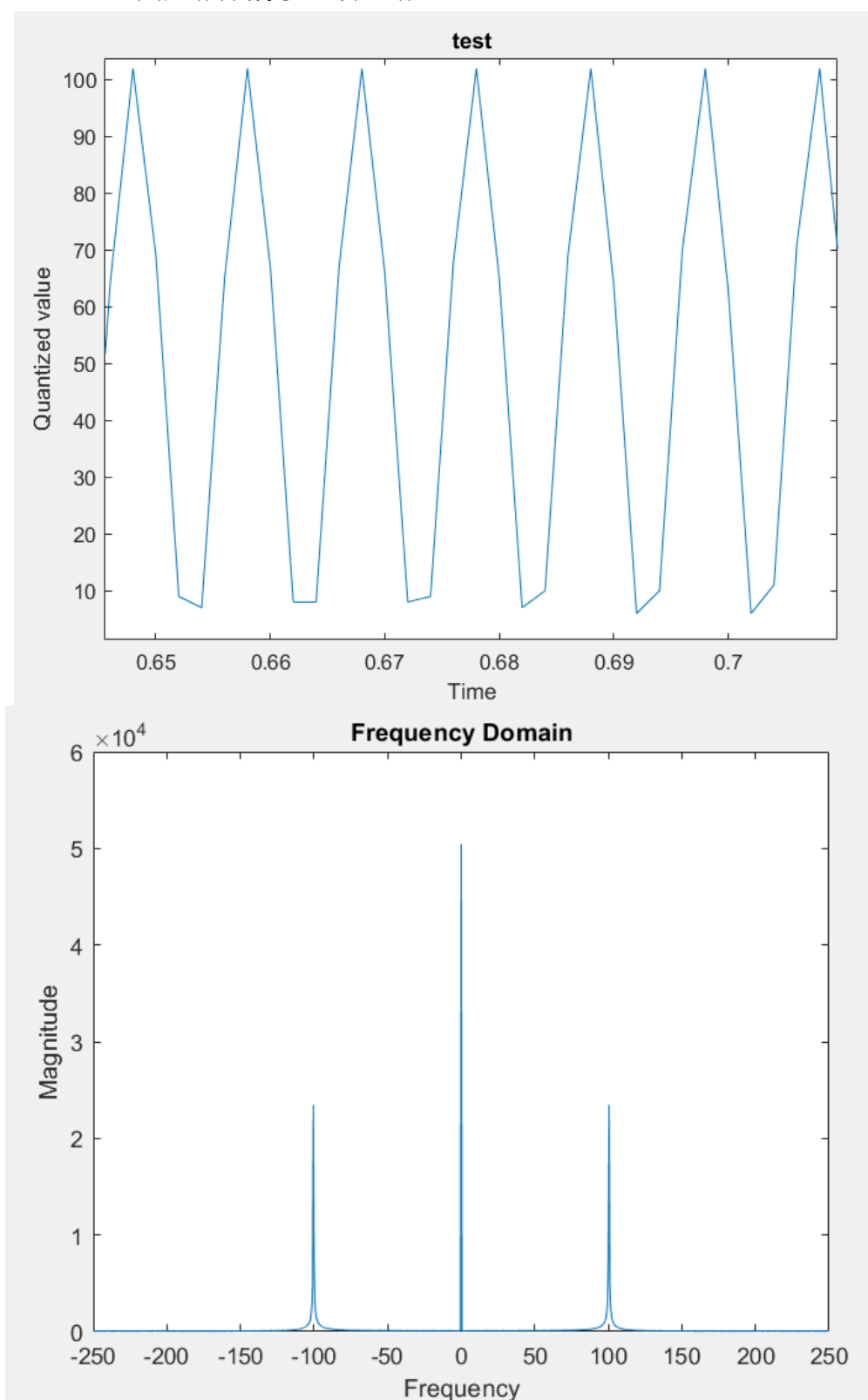
## A. 10 Bits

頻譜看起來十分平滑，noise 相當小。從我們的推倒也可看出 10 Bit 提供 60dB SNQR，代表訊號是雜訊的 1000000 倍，因此雜訊幾乎可以忽略。



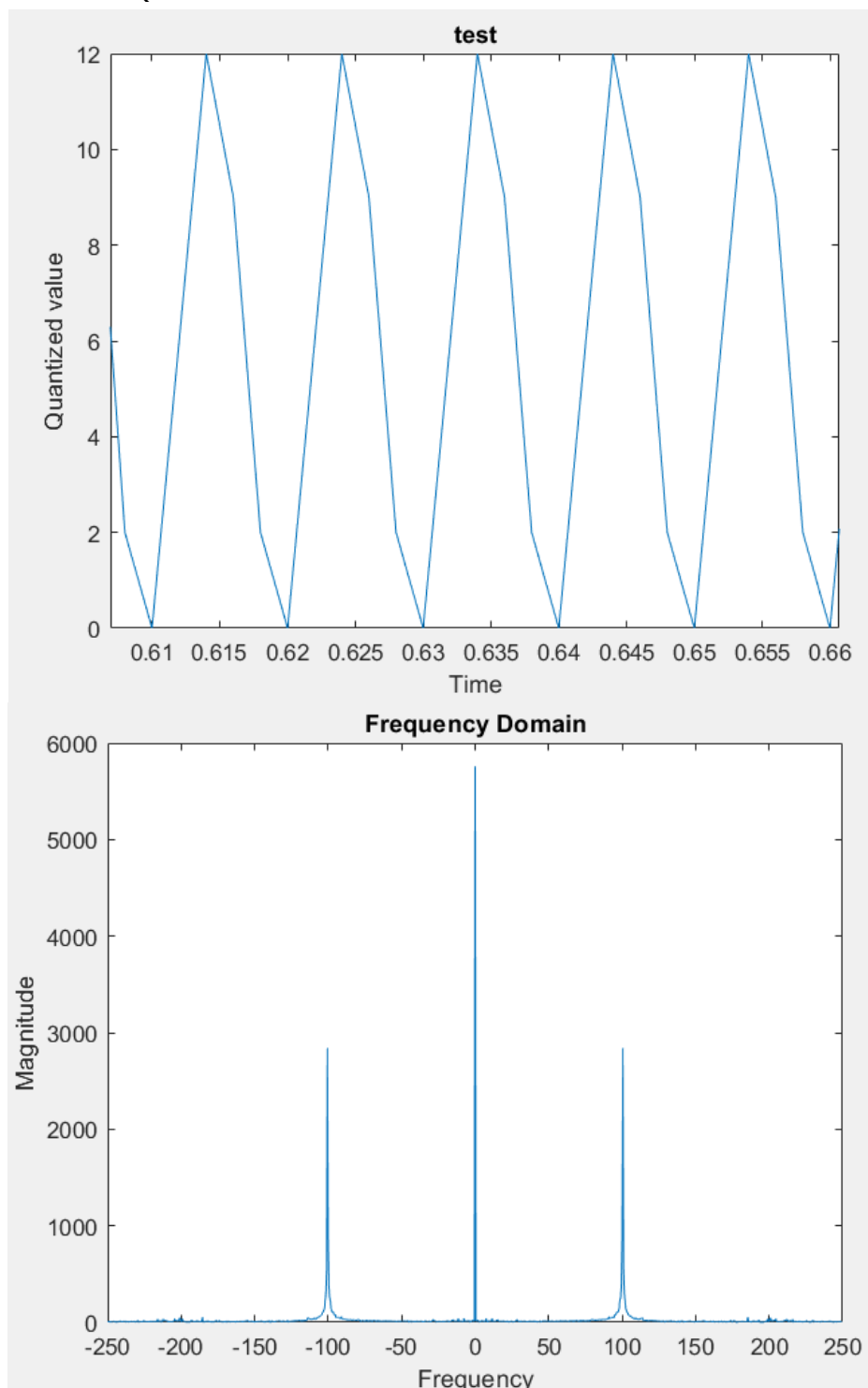
## B. 8 Bits

仍舊看不出有雜訊。從我們的推導也可看出 10 Bit 提供 48dB SNQR，代表訊號是雜訊的約 100000 倍，因此雜訊幾乎可以忽略。



## C. 5 Bits

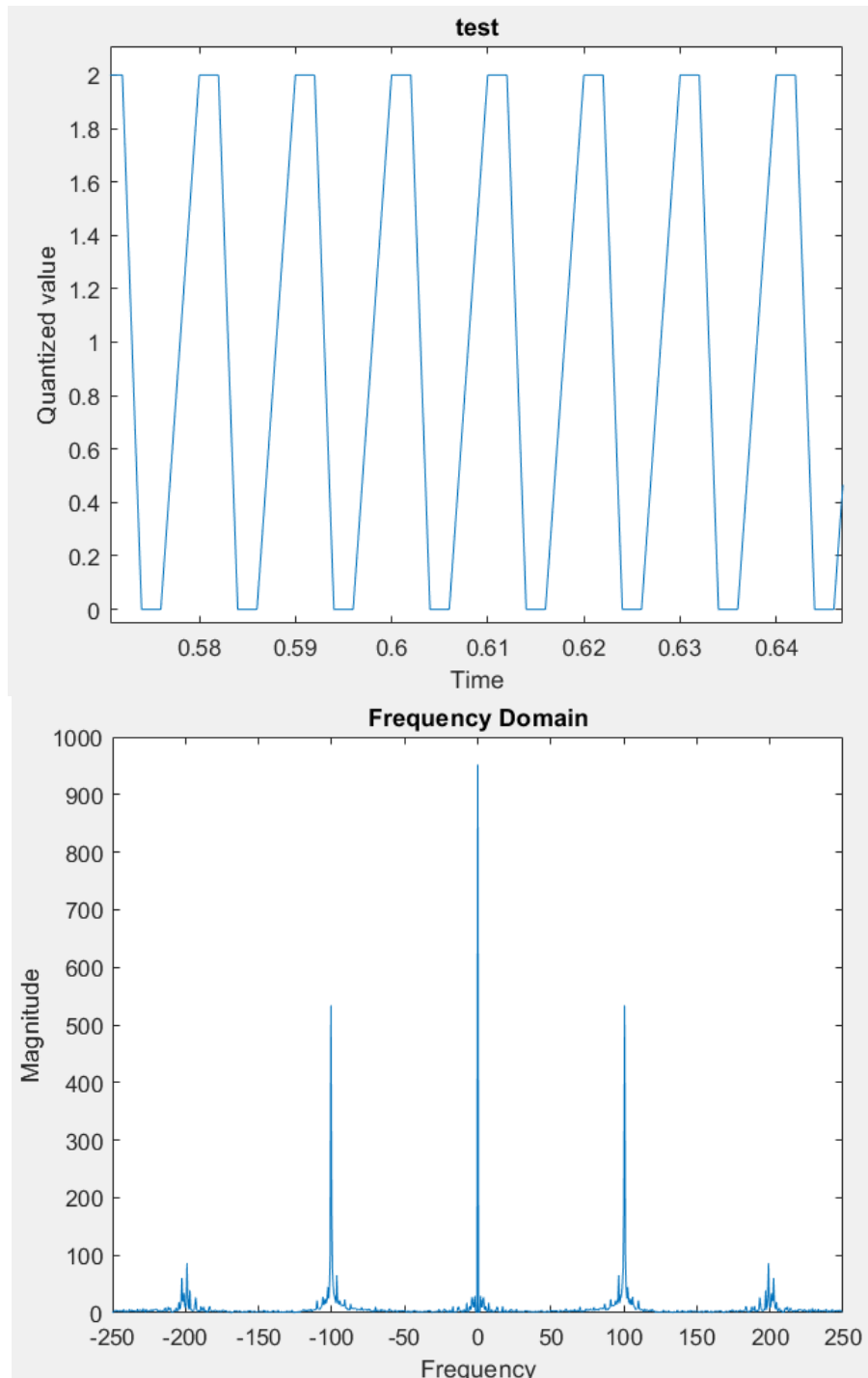
已經可以看出有一點雜訊成分出現在頻譜上了，再 200Hz, -200Hz 以及 0Hz 處有一些小凸起。這時的 SNQR 為 30dB，訊號是雜訊的 1000 倍。





## D. 3 Bits

從頻譜可以很明顯的看出有雜訊存在於 0, 200 以及 -200Hz 處。原因在於這時的 SNQR 為 18，代表訊號能量約雜訊能量的 60 倍左右而已。

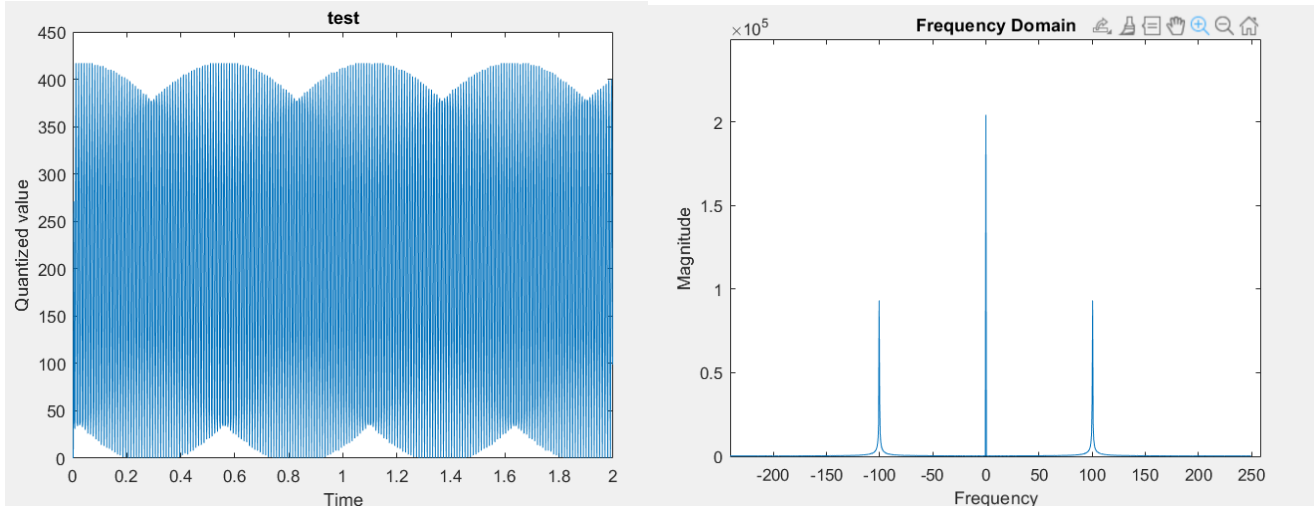


### ● Discussion

前一個部分基本上已經滿清楚解釋，不同 sampling rate 和 quantization bit 對量測的影響。這部分我們討論一些實驗中遇到的特殊情形，並嘗試做出解釋。

**Q1 )為什麼取樣後的波形看起來像是一個低頻的訊號乘上一個 100Hz 的載波(類似於 AM)**

我們在 MATLAB 中得到的波形圖如下，如果沒有進行 zoom in 可以發現他與我們預期的波形圖(100Hz 正弦波)有些不同，反而比較像一個經過 Amplitude Modulation，並取 100Hz 作為 carrier frequency 的訊號。從頻域也可以看出，這個訊號中有一個 DC 訊號。

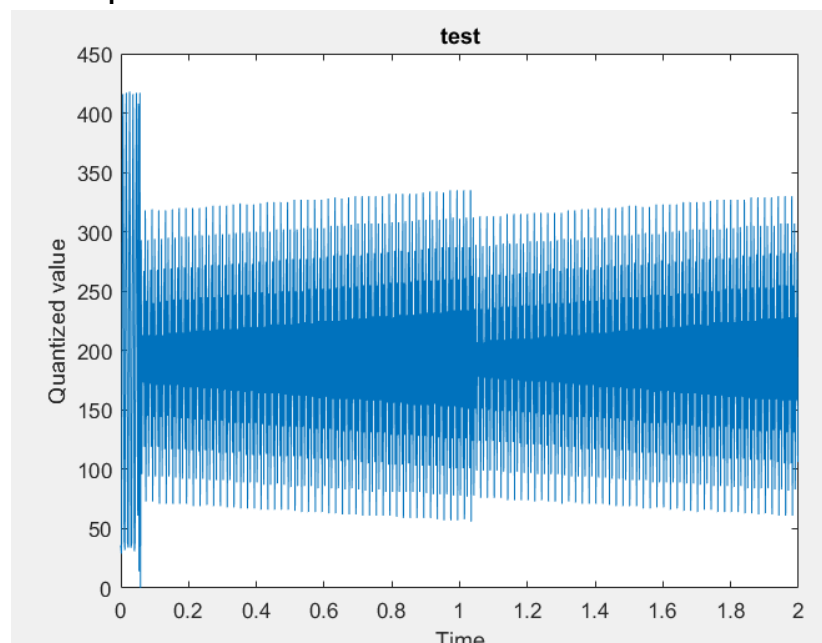


這個 DC 訊號的來源是我們對 Picoscope 的設定。因為 Arduino 中 AnalogRead 對於電壓的可讀取範圍是 0V~5V，為了要輸出一個可被 Arduino 讀取的正弦波，我們必須給 Picoscope 輸出的正弦波一個 offset(不然會出現負電壓，導致 Arduino 讀出錯的電壓值)。這個 offset 本身就是一個 DC 訊號，這才導致我們在示波器上的訊號有一個 DC 訊號，看起來像經過 Amplitude Modulation 的訊號。

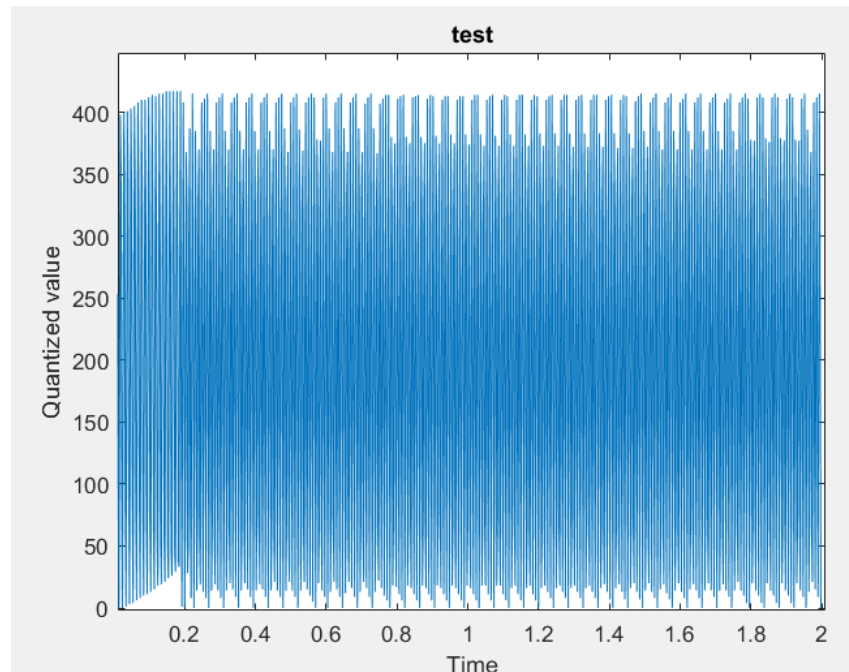
## Q2 )BaudRate 的影響是什麼？

BaudRate 是資料傳輸的速率，以我們的 code 為例，Baudrate = 115200 表示每秒傳輸 115200 bit。因此 Baudrate 會影響我們接收到的值。假設我們以  $f_s$  作為 sampling rate，Arduino 的 analogRead 所回傳的值是一個 integer 佔 4byte，也就是 32bit。換句話說，我們每秒會取得  $\text{Data\_Amount} = 32 * f_s$  這麼多 bit 的資料。如果我們採用的 BaudRate  $< \text{Data\_Amount}$ ，那就會發生後半部分的波形無法完整呈現。下面我們簡單做一個實驗，以  $f_s = 500\text{Hz}$  為例，Baudrate 需大於  $500 * 32 = 16000$  避免上述的事情發生，我們嚐試 3 種 Baudrate，分別是 9600, 19200, 25000，結果如下：

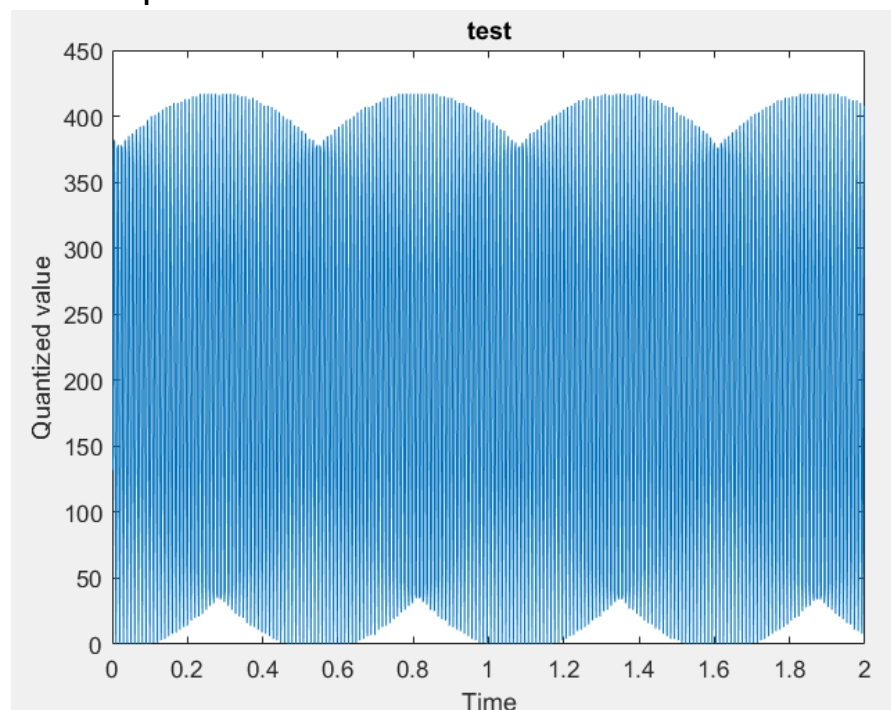
### 1. Baudrate= 9600bps



### 2. Baudrate = 19200bps



### 3. Baudrate = 25000bps



可以發現當 Baudrate = 9600 以及 19200 時的波形都出現了問題，跟我們前未量測到的完全不同，而在 Baudrate = 25000 時就不會有這個問題了。其原因就如我們面所述，資料傳輸太慢會導致我們沒辦法做一個 real-time 的示波器。另外，雖然 19200 大於我們所推導的 16000bps 但仍然在波形上出現問題，我認為這與 Arduino 與 PC 端都溝通方式有關，除了傳送資料外，Arduino 可能還要傳輸額外的資料來建立起整個 communication channel 才會出現這樣的差異。

#### Q3) Quantization 造成的影響為何？

詳細的說明可以參考前一部分所述。總體而言，Quantization 所造成的 error 對於我們分析訊號就如同 noise，會在頻譜以及波形上造成影響。Resolution 越高(Quantization bit 越多)，雜訊的影響越小(SQNR 越大，每多一個 bit 多 6dB)。

## ● Conclusion

在這個 Lab 中，我們運用 Arduino 實際體驗 sampling 和 quantization 的過程。其中，也可以看到不同 sampling rate 和 quantization bit 對結果造成的差異。整個過程讓我簡單的複習在訊號與系統與 DSP 導論中學到的知識，也遇到一些在實作上才會出現的問題(ex. offset 造成的 DC 訊號、Quantization noise 以及 Baudrate 造成的影響等)，十分有趣。

## ● References

- 教授與助教的講義
- 李祈均教授的DSP講義
- Arduino Reference