

1 Try to be familiar with Arduino. See the Arduino training manual and finish the labs till class 7. No demo is required for this part.

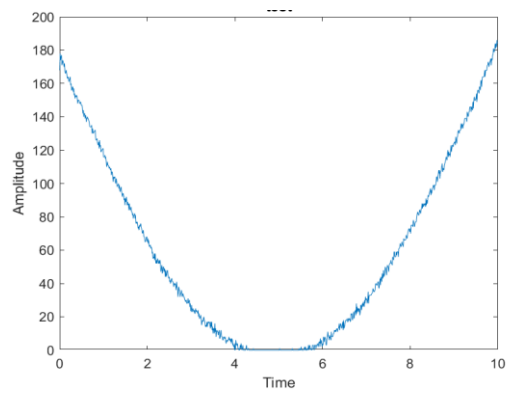
在這些簡單的嘗試中並沒有遇到明顯的問題，主要是必須習慣 Arduino 的語法和電路的接法，Arduino 必須要在 `setup` 部分設定好 `pin` 所接的位置，而且必須要分 `input` 和 `output`，可以將訊息傳至 Arduino 的，例如 `button` 就是作為 `input`，而 Arduino 輸出訊息所控制的如 `led` 燈就是 `output`，而 `loop` 部分則是要 Arduino 不斷循環執行的程式，例如控制 `led` 燈閃爍的頻率或定期讀取電阻大小等，而比較可能遇到的問題主要在於有些讀取或寫入資料的值有大小範圍限制，例如 `analogRead` 和 `analogWrite` 等，在嘗試很容易因為大小範圍限制或單位的問題沒注意而導致 Arduino 運作不正常，不過這部分只要多加注意，很容易就可以從網路找到相關資料修正正確的範圍或單位。

2 (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.

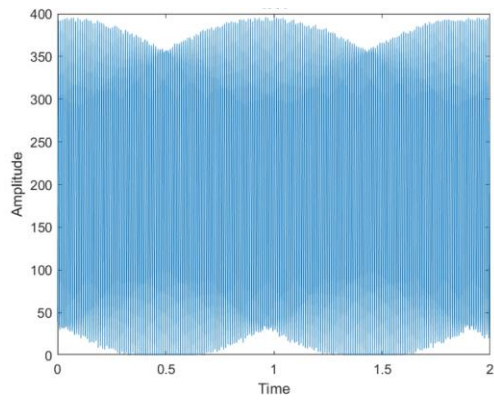
1) Plot the waveforms with the provided MATLAB codes.

500Hz

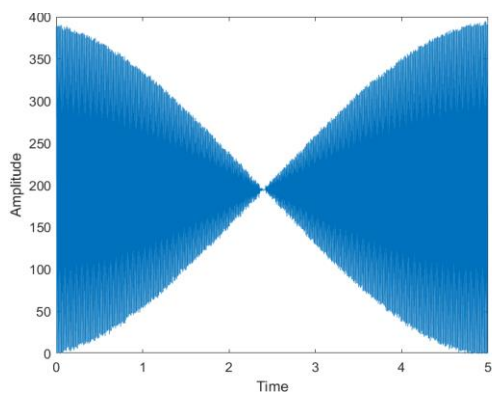
200Hz



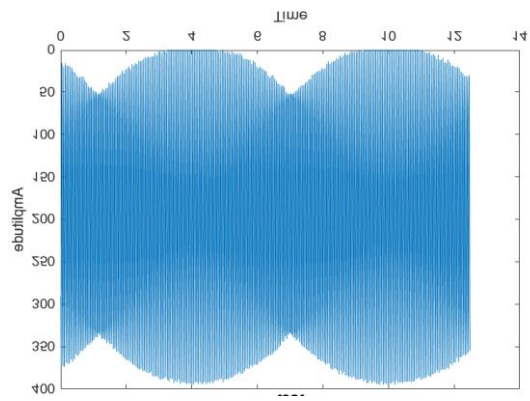
100Hz



80Hz

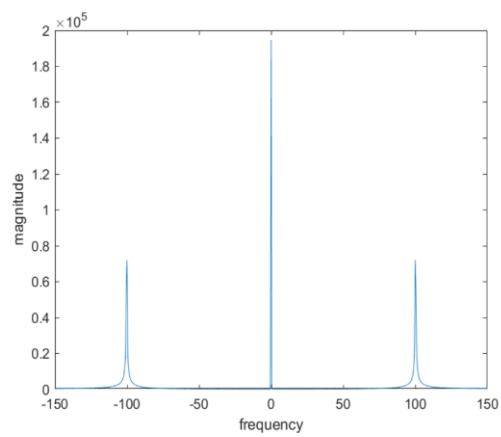


500Hz

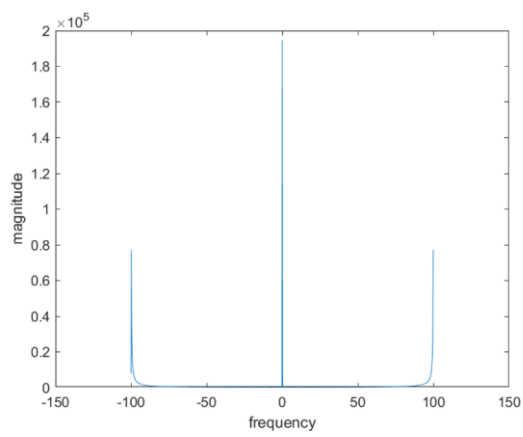


200Hz

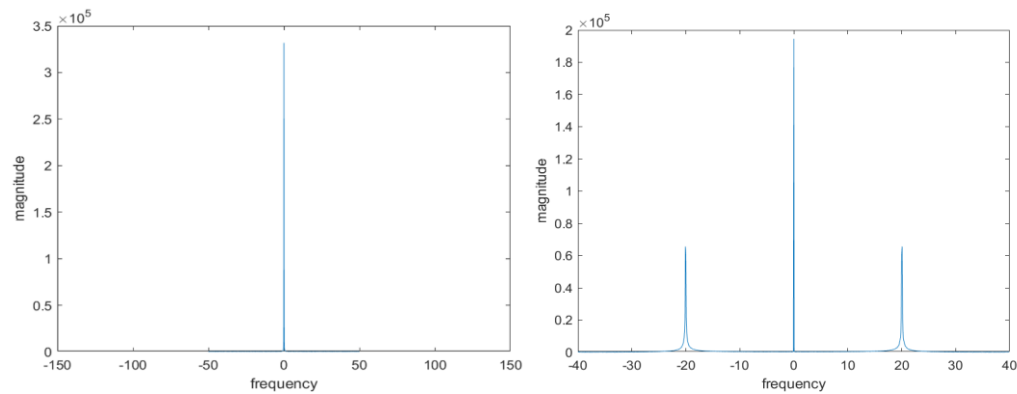
2) Use FFT to show the spectra of the saved samples.



100Hz



80Hz

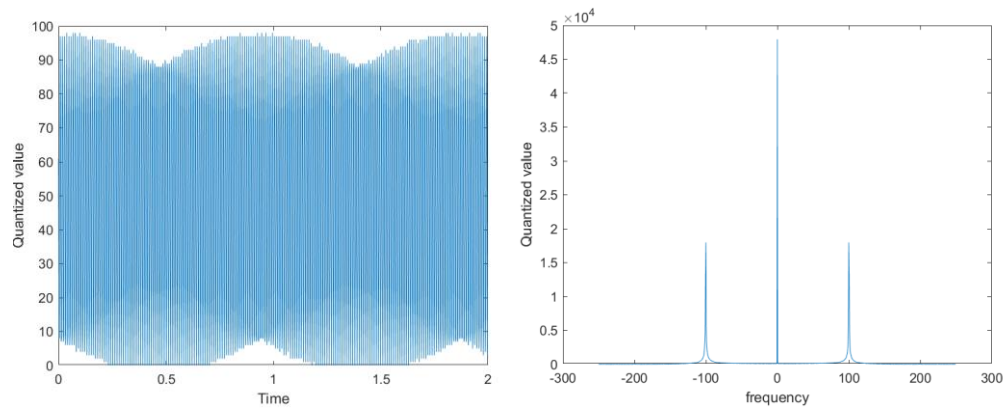


### 3) Discuss about the aliasing issues.

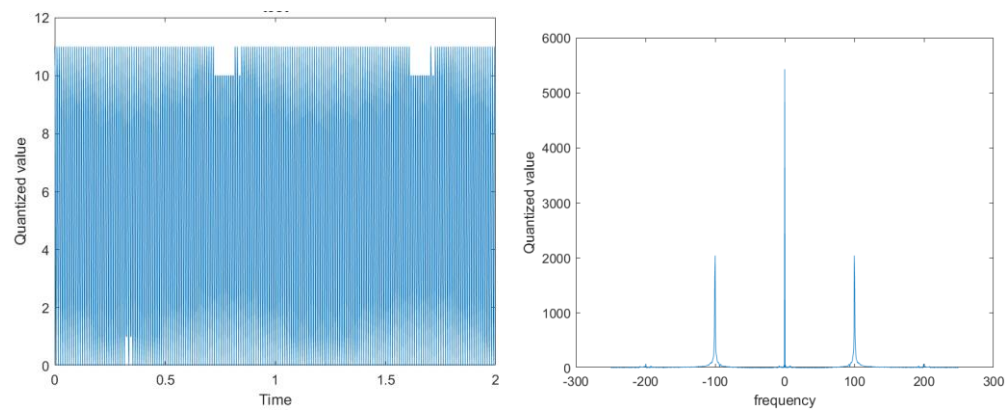
如果取樣低於兩倍 Nyquist frequency 就會有 aliasing 產生，進行取樣，會使原本訊號的頻譜振幅改變，並且在新的頻譜以週期性出現，如果取樣頻率不適當就會導致這些週期性出現的頻譜有所重疊，因為我們所產生的 sine wave 的週期為 100Hz，本實驗中的 Nyquist frequency 是 100Hz，故而要取樣頻率超過 200Hz 才不會產生 aliasing，而由圖中的結果可以看出，500Hz 和 200Hz 因為取樣頻率夠高，頻譜的組成正確地出現在正負 100Hz，而在 100Hz 取樣頻率的圖中，因為頻譜是以 100Hz 重複出現，而每個週期的頻譜寬度是 200Hz，所以會分別以正負 100Hz 為中心向上和向下 100Hz 出現頻譜的組成，而正 100Hz 向下和負 100Hz 向上的頻譜剛好就落在 0 的位置，因此才會看到能量只落在 0Hz，而 80Hz 的取樣頻率時，以正 80Hz 為中心向下 100Hz 為負 20Hz，以負 80Hz 為中心向上 100Hz 為 20Hz，所以才會有能量落在正負 20Hz，而 aliasing 會導致頻譜的組成出現在不正確的位置，因此要 reconstruction 時無法重建原本的訊號，要避免這樣的現象，以本實驗來說就是要取 Nyquist frequency(100Hz)兩倍以上的取樣頻率，如此一來週期性出現的頻譜就不會互相干擾。

## 3 (Quantization) Use 10 bits, 8 bits, 5 bits, and 3 bits to sample the oscillator waveform

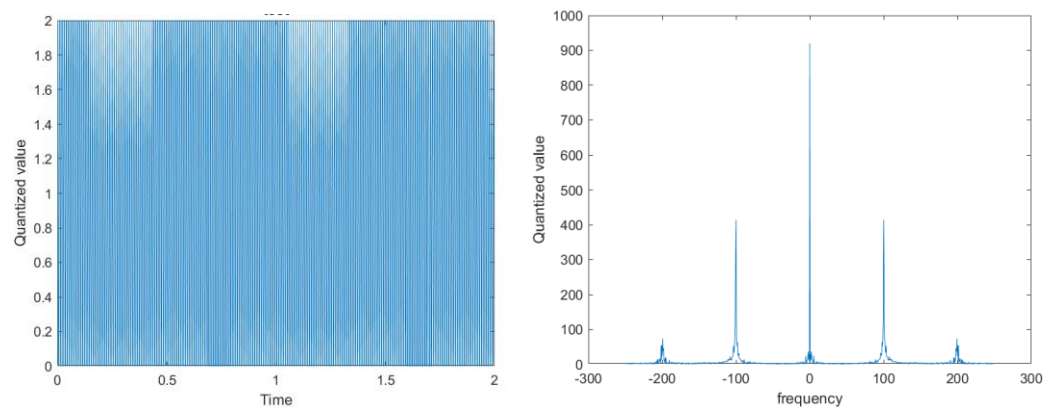
500Hz quantized 8bits



500Hz quantized 5bits



500Hz quantized 3bits



## 其他問題

(1). Why does the sampled sine signal ( $f_s = 500$  Hz or  $200$  Hz) look like an amplitude-modulated (AM) sine wave (i.e.,  $100$  Hz sine carrier with amplitude modulated envelope)? not like an expected sine wave? If you try to interpolate the sampled sine

signal, you will see the expected sine wave. Why? (You can try using simulated sine signal)

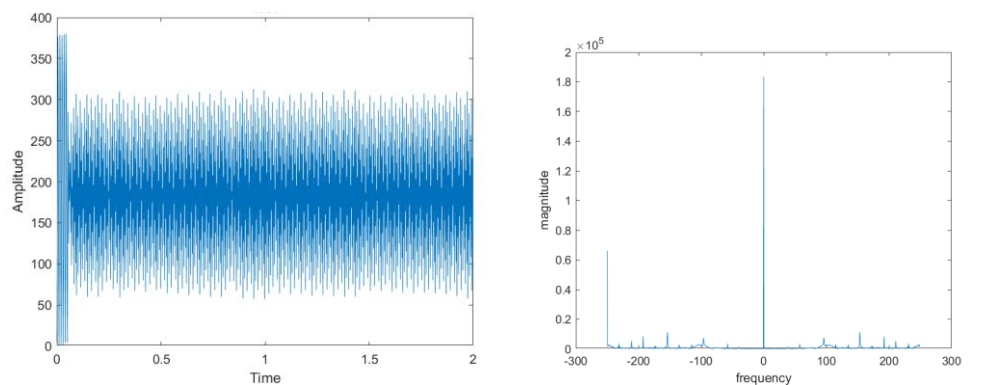
如果對於時域的 signal 作取樣，因為取樣只保留少數的點，所以無法直接表現出 sine wave，取樣後的 signal 的頻譜會等同於原本 signal 的頻譜以每  $2\pi/T$  的週期重複出現，並且振幅為原本的  $1/T$  ( $T$  為週期)，也就是原本的 signal 的頻譜和

$\frac{1}{T} \sum_{-\infty}^{+\infty} \delta(j(\omega - \frac{2\pi}{T} k))$  作 convolution 就相當於取樣後的頻譜，而  $\frac{1}{T} \sum_{-\infty}^{+\infty} \delta(j(\omega - \frac{2\pi}{T} k))$  除了頻率和振幅的差異外，和 sinusoid wave 的頻譜有非常接近的形式，

所以在頻域和  $\frac{1}{T} \sum_{-\infty}^{+\infty} \delta(j(\omega - \frac{2\pi}{T} k))$  作 convolution 類似於和 cosine wave 的頻譜作 convolution，而在頻域和 cosine wave 的頻譜作 convolution，也就等同在時域乘上 cosine wave，和 AM 調變的將 sinusoid wave 作為載波的形式類似。

(2). In the experiment, you're using the highest baud rate (i.e., 115200 bps) of the serial port. What would happen if you set a lower baud rate, e.g., 9600 bps or 19200 bps?

baud rate 就是訊號的傳送速度，因為每個取樣點都要用 10bits 來表示，因此 sampling rate 乘上 10 不宜太過於接近 baud rate，如果 sampling rate 很高而 baud rate 不大時，有可能因為傳送速度不足，導致有部分訊息無法完整傳送，在本題當中當 sampling rate 200 以下時，即使 9600bps 也可以很表達出取樣的 signal，但是當 sampling rate 高於 500(如下圖)後，取樣的 signal 會非常明顯的不穩定，頻譜也嚴重的雜訊，應該就是因為傳送速度不夠快導致部分訊息沒有順利取得，而提高到 19200bps 仍不太足夠，在 500Hz 的 sampled signal 的頻譜仍有很大的雜訊。



(3). Please remember to **elaborate** the spectral difference between the 10-bit and 3-bit sampled sine wave, and what causes the difference.

可以由看出 3-bit 的 signal 很明顯的改變，10bit 時對於 signal 細節的高低起伏可

以很清楚表現，但是 3bit 時 signal 變化很少，也無法呈現細節的高低起伏，並且頻譜嚴重失真，在正負 200 處有 power，和原本 10bit 時只有正負 100 有 power 不同，原因應該是因為在 quantization 的過程當中，我們只能將訊號表達成最接近的可表達的值，而 3bit 表示我們只用 8 個值來表示所有訊號的 amplitude，10bit 則是用 1024 個值來表示，越多的值表示能夠呈現更多的細節，而且能夠利用和真正的 amplitude 相差較小的值來表示，因此導致了 10bit 比 3bit 更能良好呈現細節，並且頻譜也更接近理想狀態。