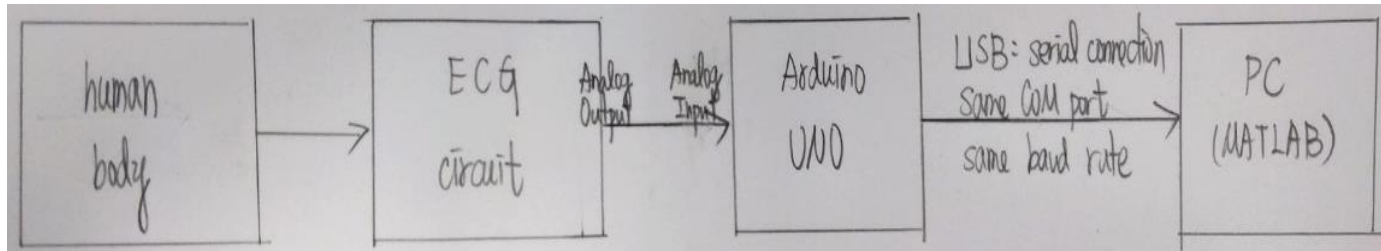


## 105061254 林士平 數位訊號處理實驗報告 Lab3

### 1.

#### (1) Design Specification :



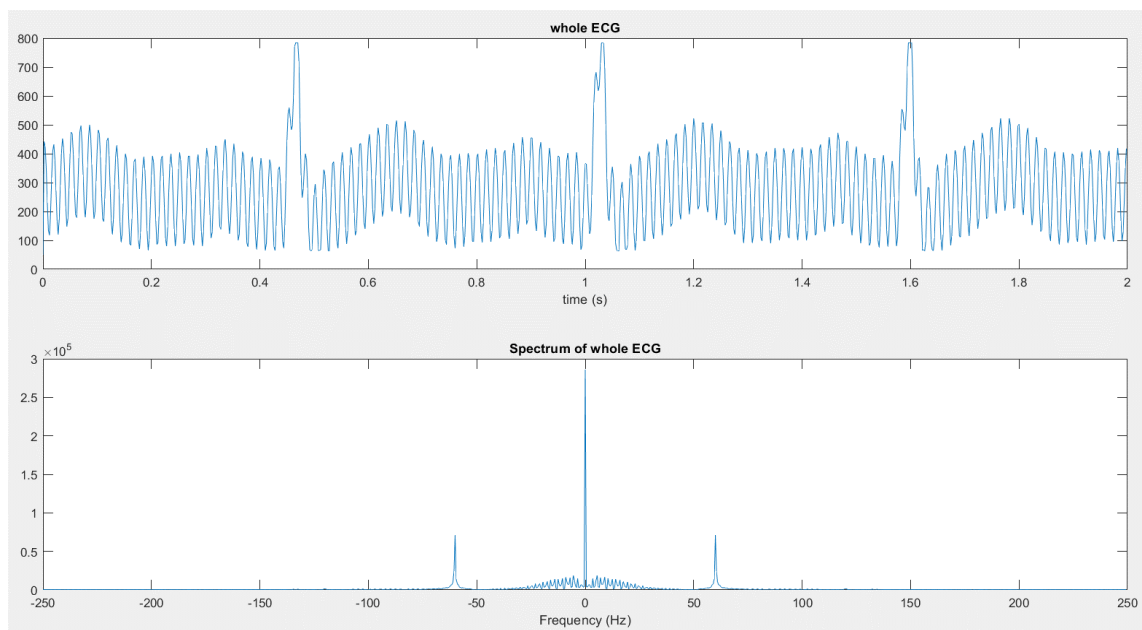
↑ 本實驗的 block diagram，其中 ECG 電路沒有使用 analog notch filter

#### (2) Experiment Objective :

- 使用 digital notch filter 取代 analog notch filter，濾除 60-Hz power line noise。
- 使用 FIR, IIR notch filter 和 moving average filter，並比較三者的結果。

#### (3) Results and analysis :

##### a. ECG 訊號雜訊分析：



↑ 尚未濾波前的心電訊號：1. 從 time domain 訊號可以很明顯發現有 **60Hz power line noise**，由 frequency domain 也可看到在正負 60Hz 的地方有兩根 impulse，因為 notch filter 為 bandstop filter，可以將 stopband 設在 60Hz 附近把雜訊濾掉，因此接下來我會利用 **FIR 和 IIR 的 notch filter**，除此之外也會使用 **moving average filter**，因為移動平均濾波器的頻譜類似 sinc function，大小=0 的地方有週期性，如果讓這個週期=60Hz，便能把 60Hz、120Hz、180Hz.....等雜訊濾掉。

2. 從 frequency domain 訊號可發現在 0Hz 附近有很大的雜訊，這個雜訊來自於 **DC offset** 和 **baseline wander**，所以必須設計 **high pass filter**(我利用 **FIR filter design**)把他們濾掉(見 2.)。

## b. 三種濾波器的設計流程：

```
% fir1(n, [f1 f2], 'stop')
% f1 and f2 are normalized to the Nyquist frequency
wo = 60/(Fs/2);
bw = 0.04; %wo/35;
a = 1;
b = fir1(100, [wo-bw/2 wo+bw/2], 'stop');
```

←**FIR notch filter** 設計流程：Fs = 500(取樣頻率)，stopband 中心頻率 = 60。首先計算 stopband 中心頻率的 normalized frequency，也就是中心頻率/Nyquist frequency，再來選擇 stopband 的 bandwidth，然後代入 fir1(階數, [stopband], 'stop')。(在此我選擇階數=100)。

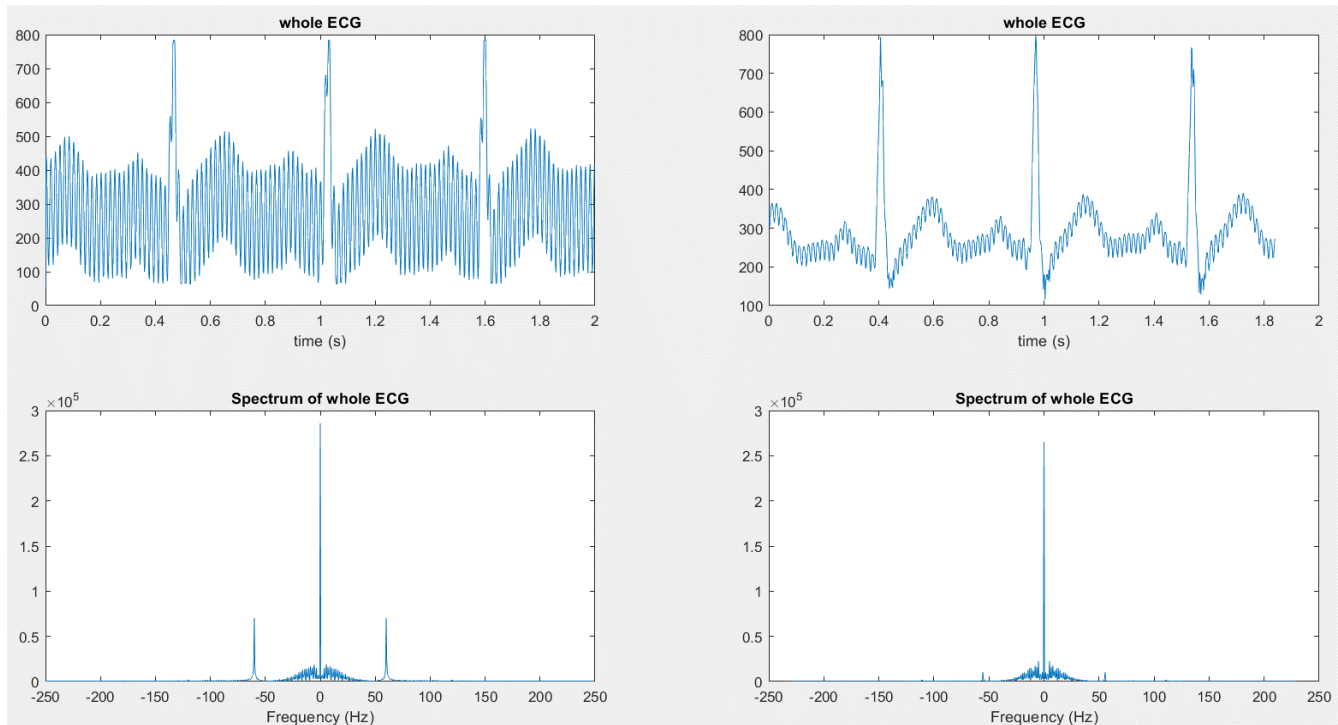
```
% [num, den] = iirnotch(w0, bw)
wo = 60/(Fs/2);
bw = wo/35; %let q factor = 35
[b, a] = iirnotch(wo, bw);
```

←**IIR notch filter(二階)**設計流程：Fs = 500(取樣頻率)，stopband 中心頻率 = 60。首先計算 stopband 中心頻率的 normalized frequency，也就是中心頻率/Nyquist frequency，再來選擇 quality factor(我選擇 35)，接著利用 quality factor = 中心頻率/bandwidth 得到 bandwidth，然後代入 iirnotch(中心頻率,bandwidth)。

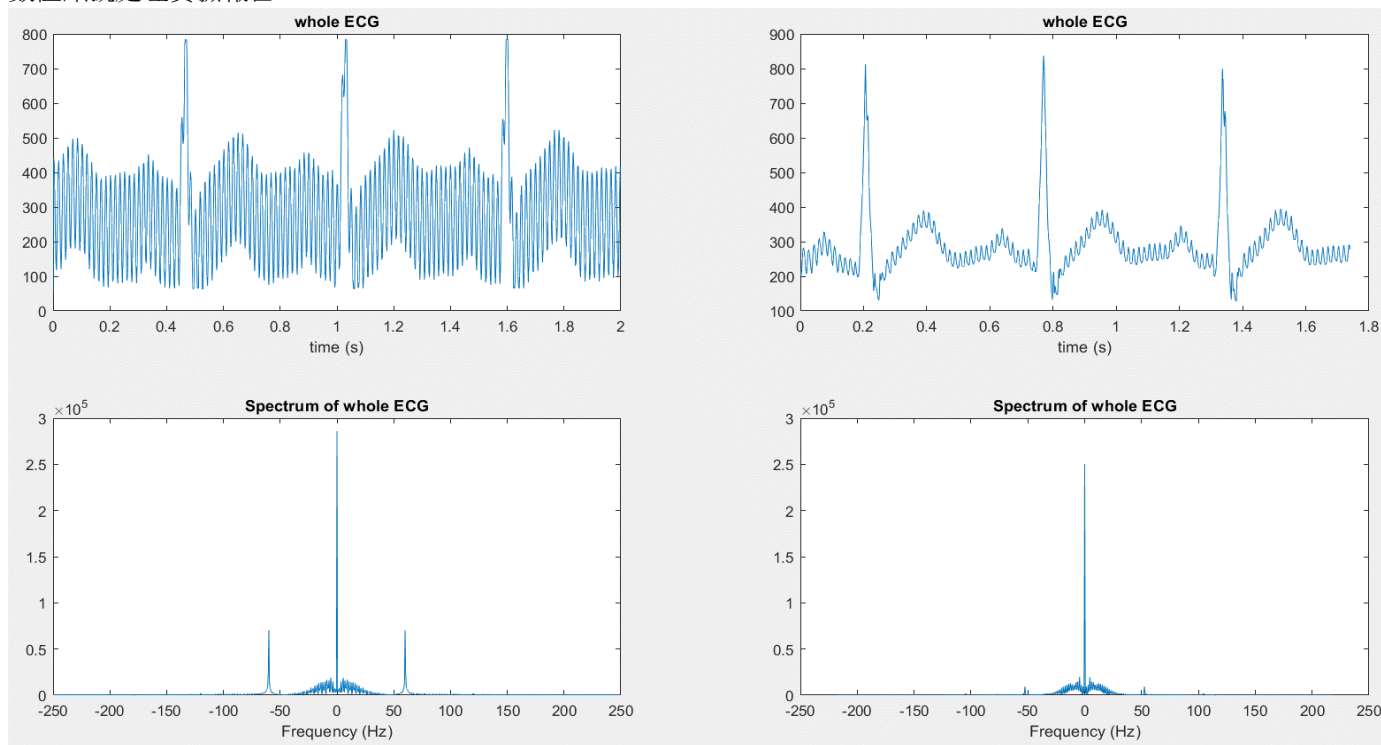
```
% moving average filter
pt = 9;
b = ones(1,pt)/pt;
a = 1;
```

←**N-pt Moving average filter** 設計流程：利用 freqz，找到讓 frequency response = 0 的點落在 60Hz 與其共振頻率時的 N 值，經過 try and error 發現在 N = 9 時最適合。

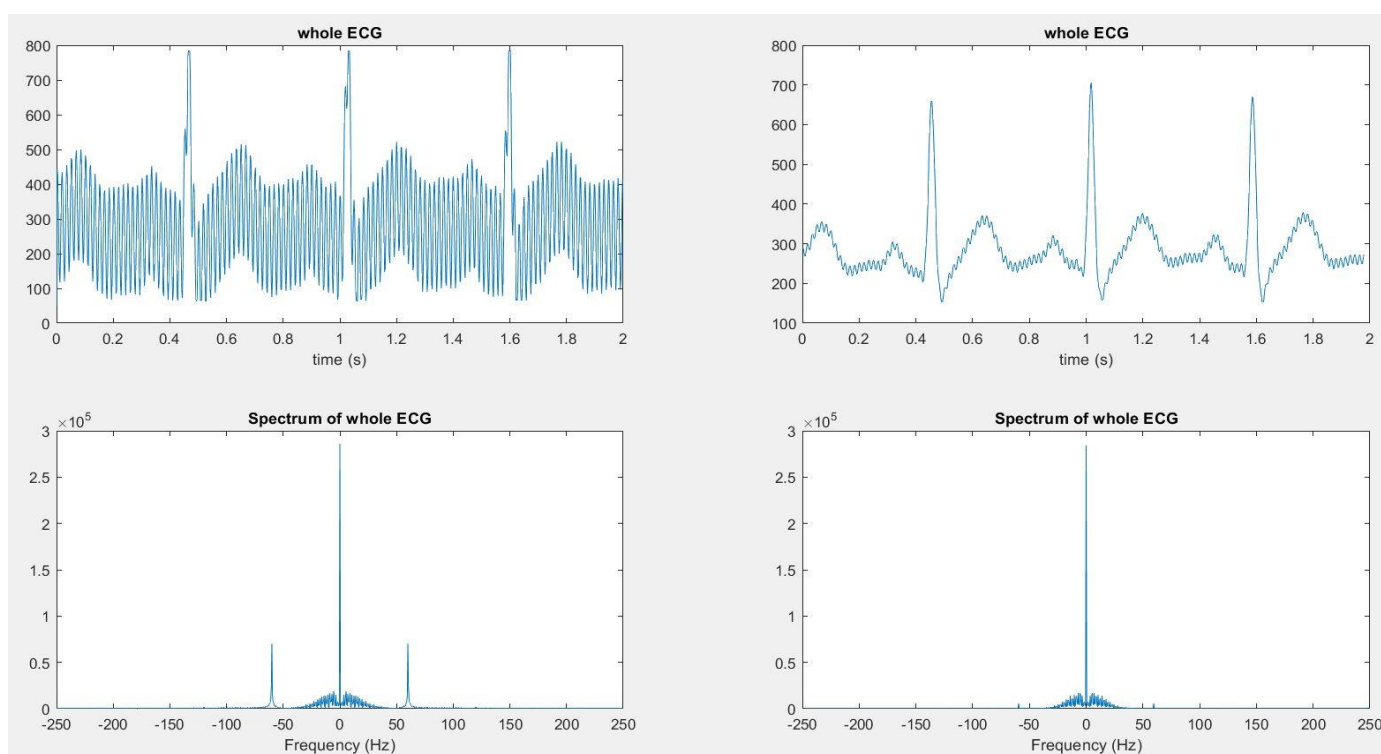
## c. 濾波前和濾波後的結果截圖比較：



↑ 使用 **FIR notch filter**(MATLAB function：fir1())結果。左上圖為尚未濾波前的訊號，左下圖為尚未濾波前的頻譜，右上圖為濾波後的訊號，右下圖為濾波後的頻譜。可以發現 time domain 在濾波後 PQRST 點變得清楚，而且 frequency domain 在正負 60Hz 的 impulse 變得非常小。

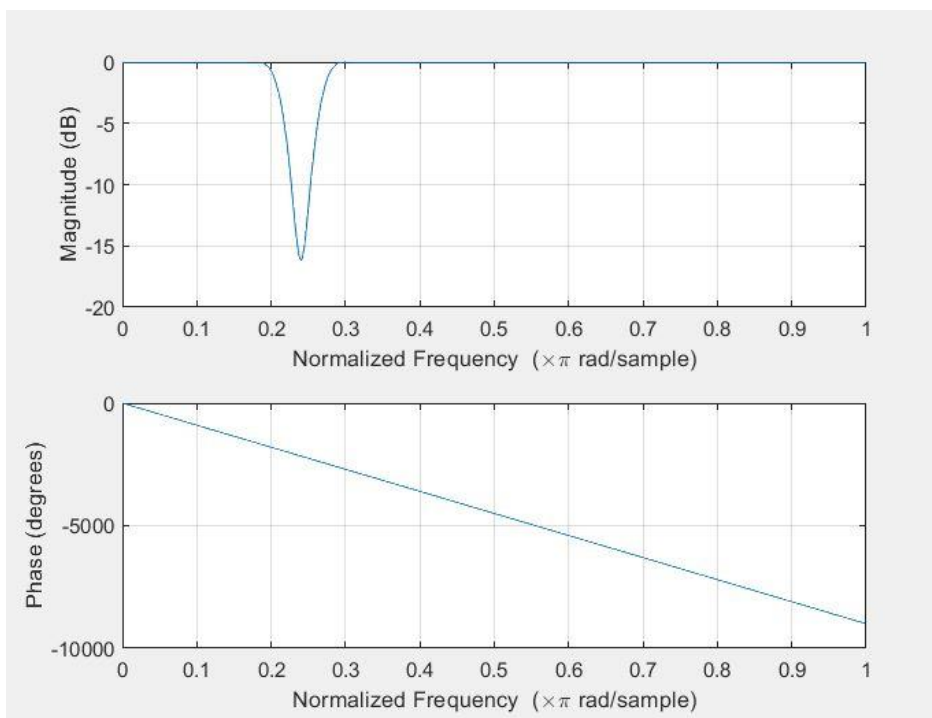


↑ 使用 **IIR notch filter**(MATLAB function : `irrnotch()`)結果。左上圖為尚未濾波前的訊號，左下圖為尚未濾波前的頻譜，右上圖為濾波後的訊號，右下圖為濾波後的頻譜。可以發現 **time domain** 在濾波後 PQRST 點變得清楚，而且 **frequency domain** 在正負 60Hz 的 impulse 變得非常小。

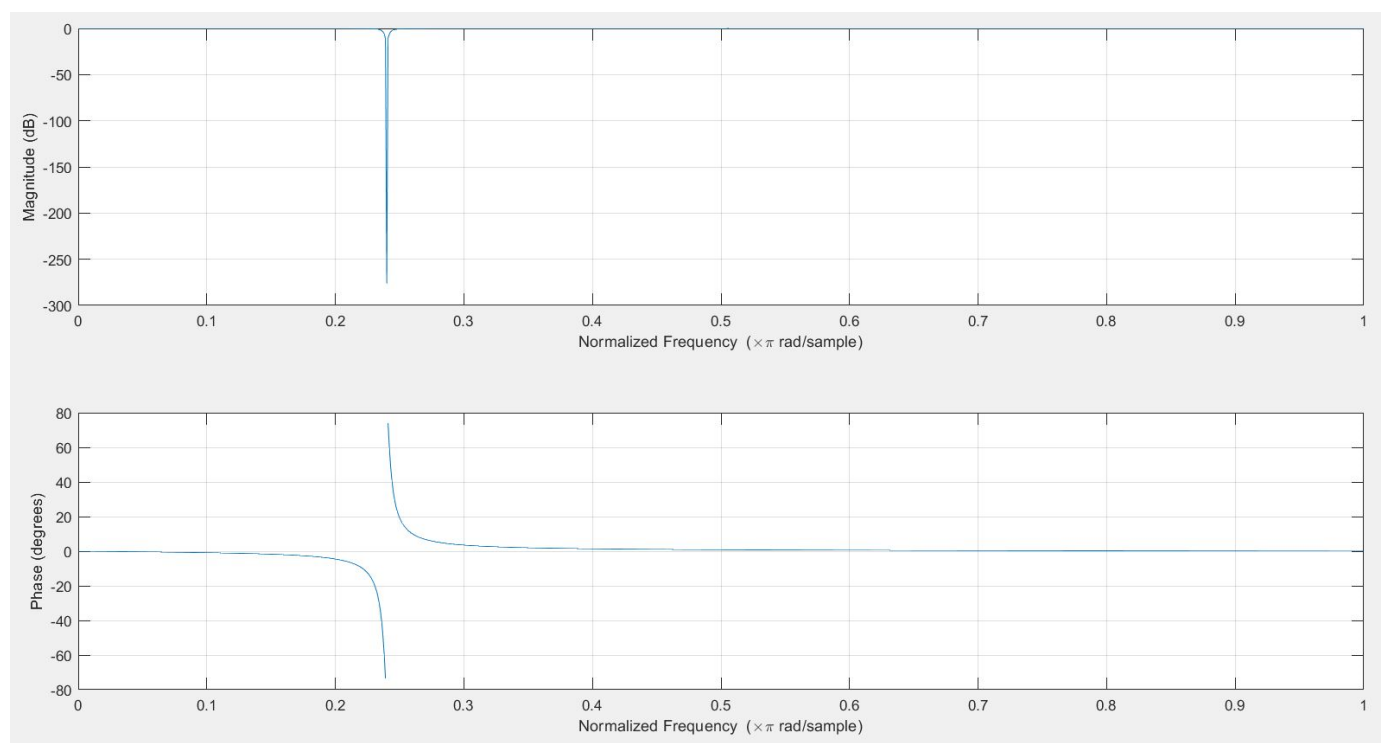


↑ 使用 **Moving average filter** 結果。左上圖為尚未濾波前的訊號，左下圖為尚未濾波前的頻譜，右上圖為濾波後的訊號，右下圖為濾波後的頻譜。可以發現 **time domain** 在濾波後 PQRST 點變得清楚，而且 **frequency domain** 在正負 60Hz 的 impulse 變得非常小。

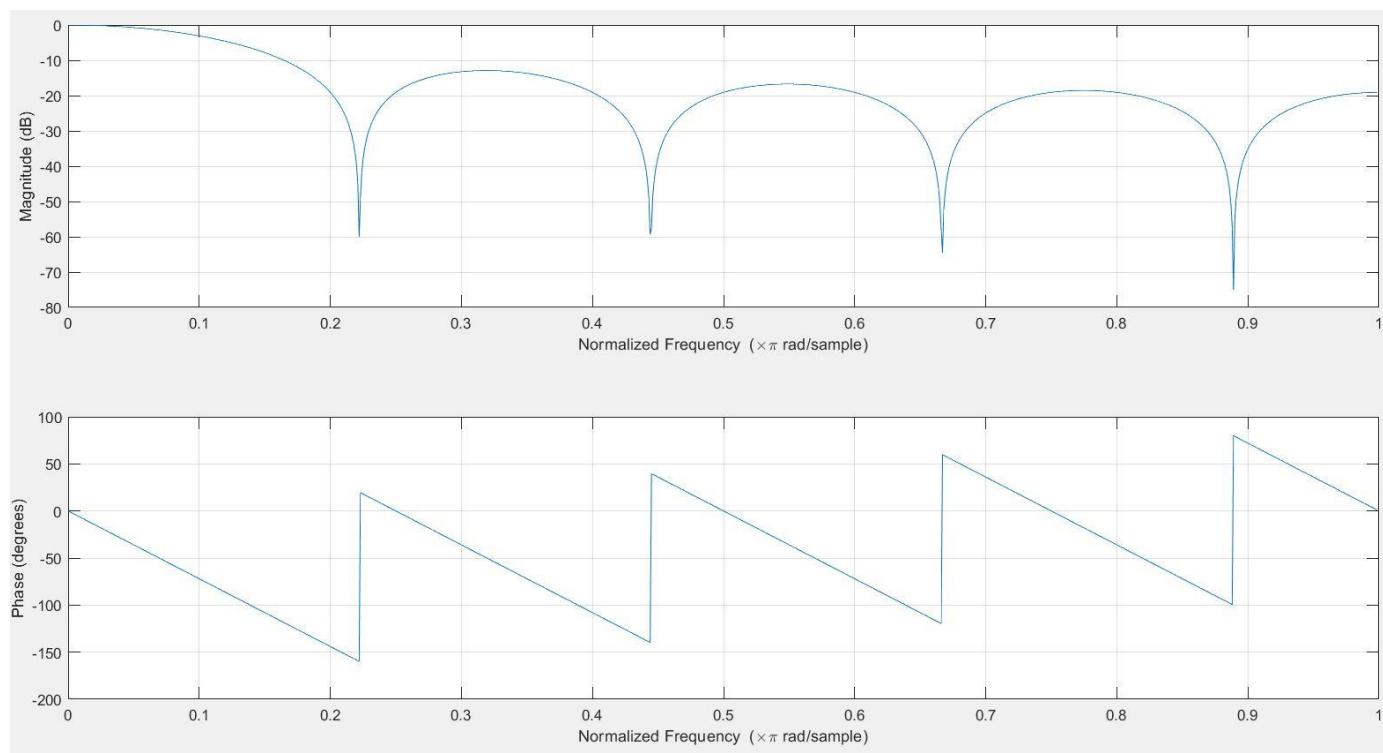
c. 三種濾波器的 frequency response



↑FIR notch filter 的 frequency response，上圖為 magnitude response，下圖為 phase response。



↑IIR notch filter 的 frequency response，上圖為 magnitude response，下圖為 phase response。

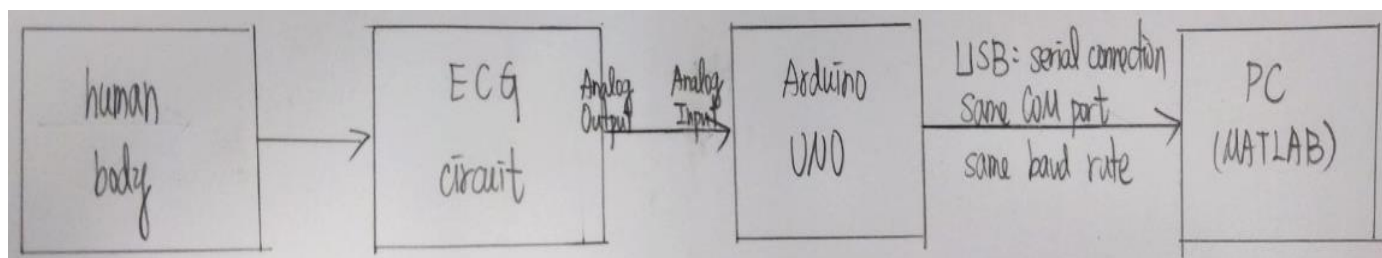


↑moving average filter 的 frequency response，上圖為 magnitude response，下圖為 phase response。

d. 使用三種濾波器的結果比較與討論見 **discussion**。

## 2.

(1) Design Specification :



↑ 本實驗的 block diagram，其中 ECG 電路沒有使用 analog notch filter

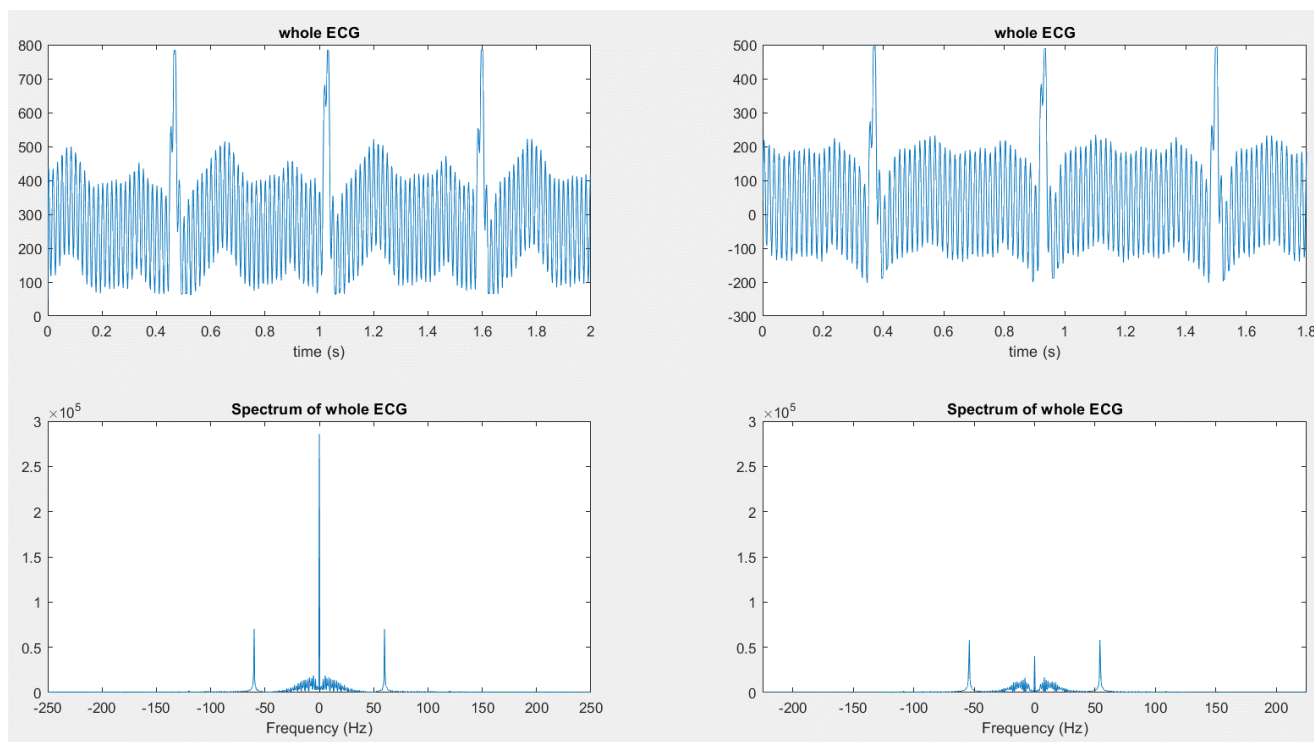
(2) Experiment Objective :

a. 使用數位高通濾波器將 baseline wander 等低頻雜訊濾除。



### (3) Results and analysis :

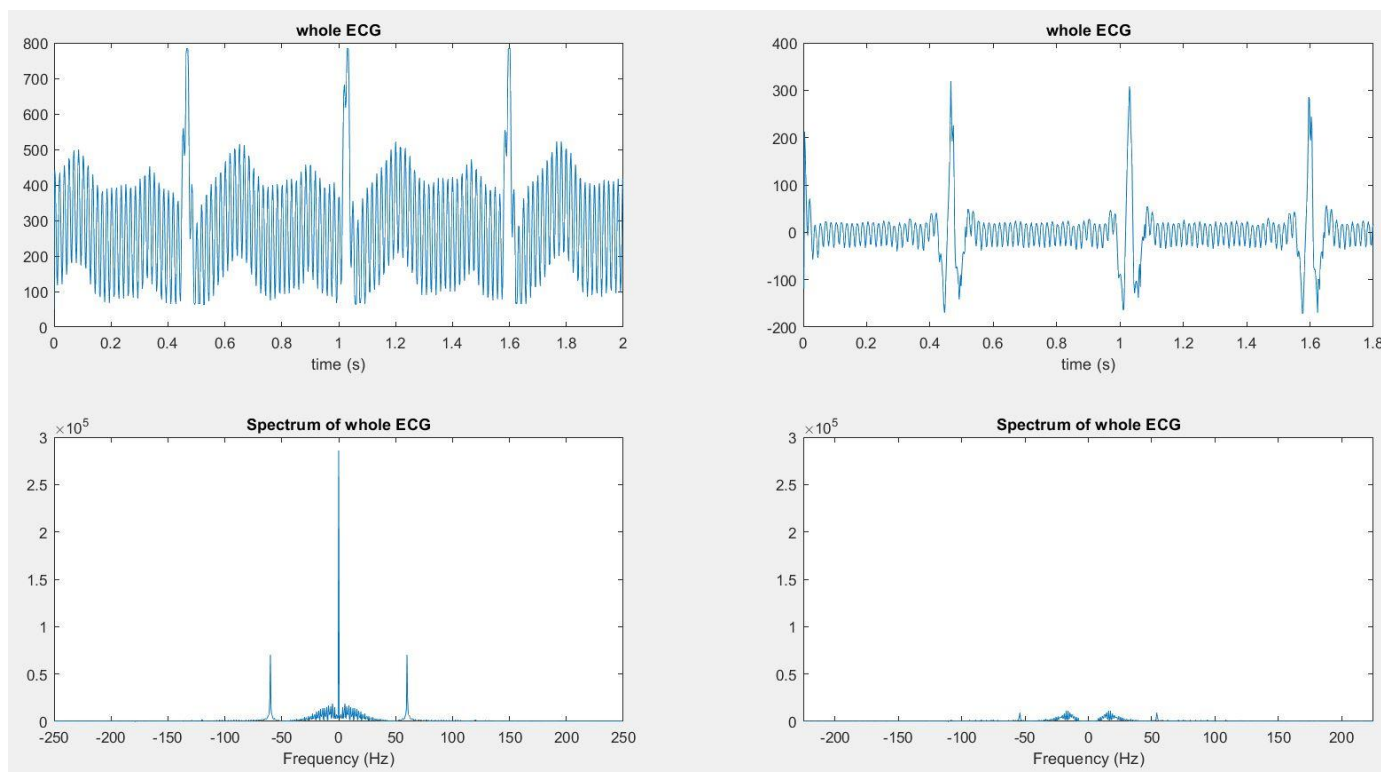
#### a. 濾波後的結果



↑使用 **fir high pass filter** 濾波結果(未經 **notch filter** 濾波)。左上圖為尚未濾波前的訊號，左下圖為尚未濾波前的頻譜，右上圖為濾波後的訊號，右下圖為濾波後的頻譜。可以發現 time domain 在濾波後 **DC offset** 消失了，ECG 訊號的平衡位置由大約 300 移動到 0，而且 frequency domain 在 0Hz 的 impulse 變小。(正負 60Hz 的 impulse 不變，因為沒有使用 **notch filter**)

#### b. 設計流程

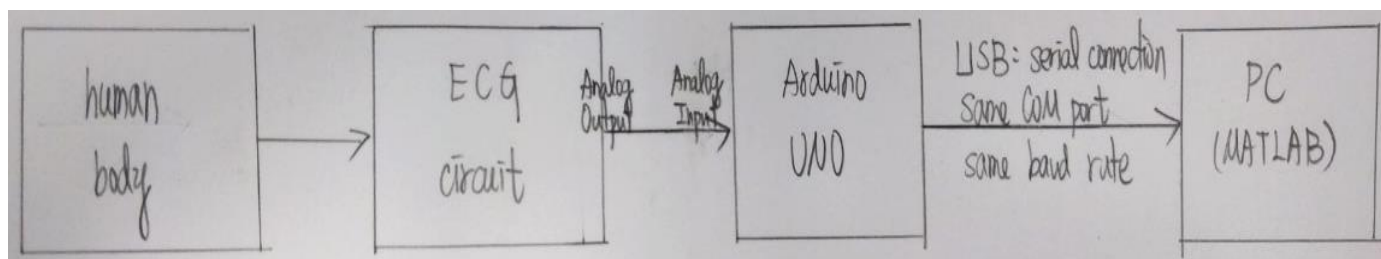
同樣使用 **fir1**，只是這次輸入參數改為：**fir1(階數, cutoff frequency, 'high')**，我的階數選擇 100，cutoff frequency 選擇 0.02。其中 cutoff frequency 便是要設計的地方，由於 baseline wander 是相當低頻的雜訊，所以 cutoff frequency 不用也不能設太大，如果設太大的話可能會把 ECG 訊號也給濾掉，下面是一個例子：



↑使用 **fir high pass filter** 濾波結果(有先經過 **notch filter** 濾波再送入 **high pass filter**)。左上圖為尚未濾波前的訊號，左下圖為尚未濾波前的頻譜，右上圖為濾波後的訊號，右下圖為濾波後的頻譜。可以發現 **time domain** 的心電訊號和 P.2、P.3 的結果比起來相當不清楚，也就是 **PQRST** 點變的不明顯，原因是 **high pass filter** 的 **cutoff frequency** 太高導致心電訊號也被濾掉。

### 3.

#### (1) Design Specification :



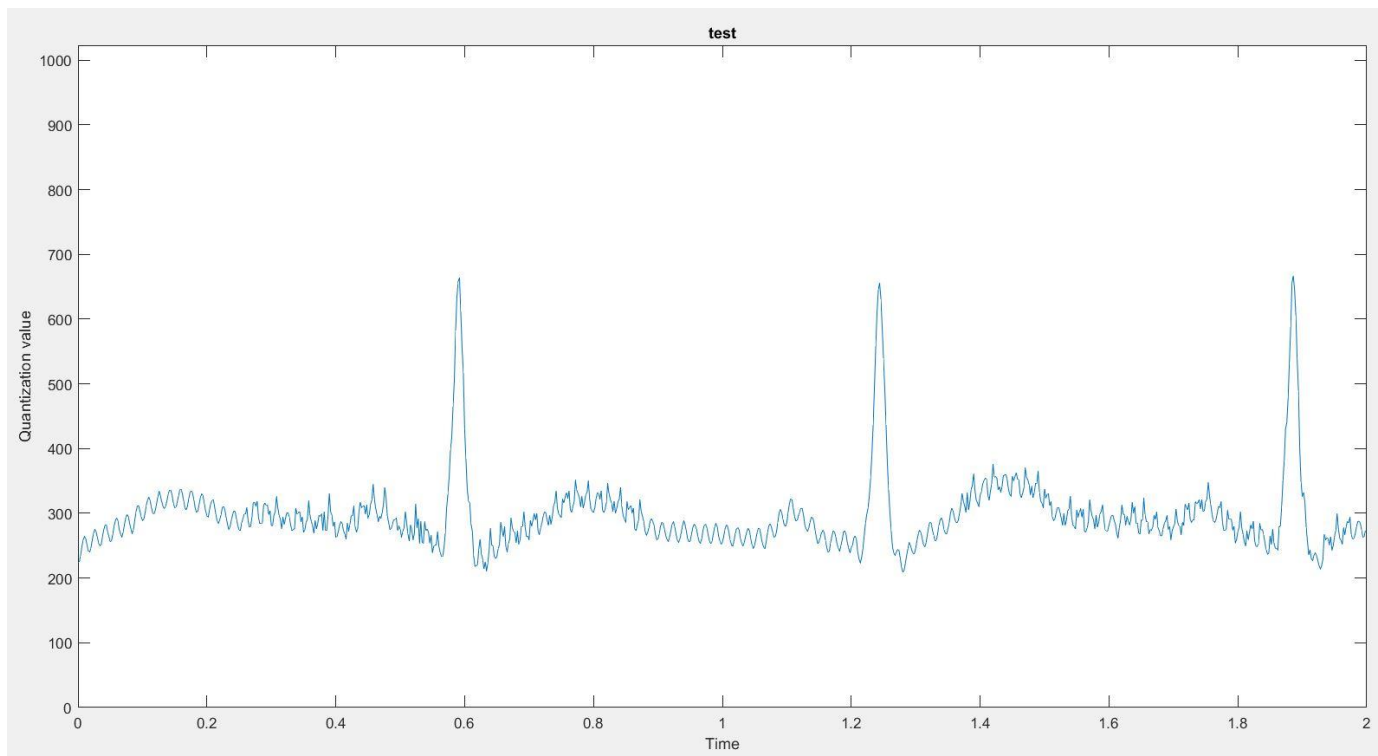
↑ 本實驗的 **block diagram**，其中 **ECG 電路** 沒有使用 **analog notch filter**

#### (2) Experiment Objective :

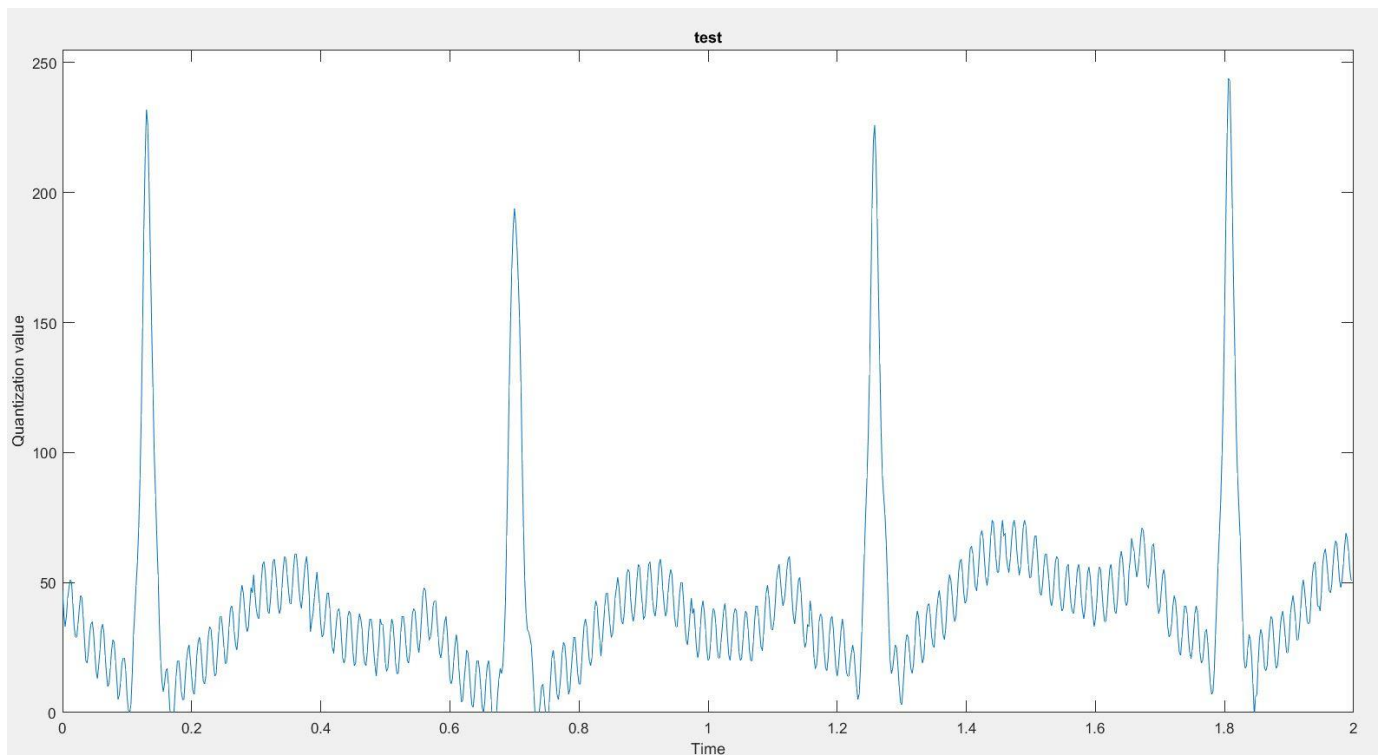
- (a) 設計 **Arduino** 程式，使 **ECG** 訊號從原本的 **10bit** 轉為 **8bit**，並能保留完整的波型，目的是讓一個 **sample** 的傳輸量下降，如此一來可以增加傳送速率，並能減少儲存空間的浪費。

### (3) Results and analysis :

a. 結果截圖如下：



↑還未處理前的原始訊號(10-bit)(沒有經過濾波)



↑處理後的訊號(8-bit)(沒有經過濾波)，由縱座標可以發現範圍從原本的 0~1023 轉為 0~255。



b. 設計流程

```
int low_bond = 200;
int high_bond = 700;
```

←首先設定兩個全域變數，分別代表原始心電訊號的下界 (low\_bond)和上界(high\_bond)。觀察 P.8 的原始訊號圖可知範圍大概在 200 ~ 700。

```
val = analogRead(pot);    //read analog input

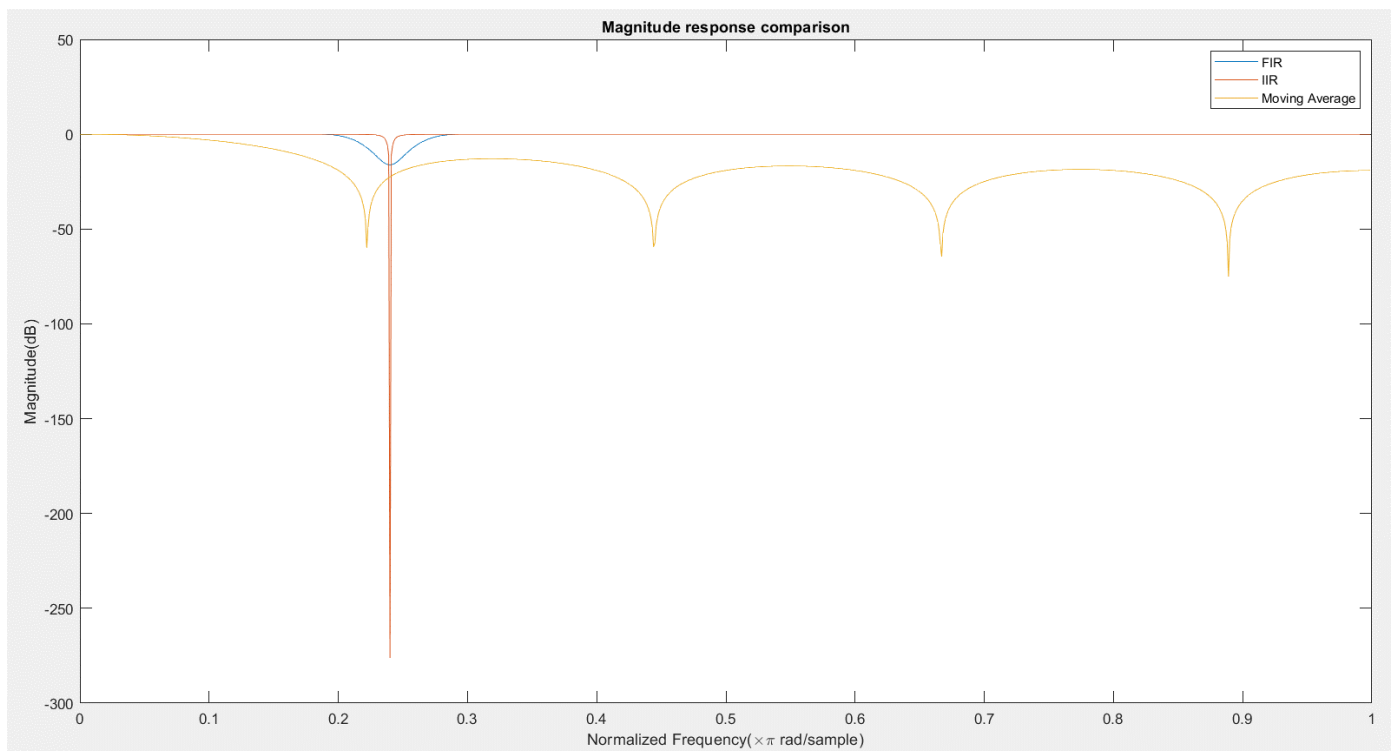
val = val - low_bond;
val = map(val, 0, high_bond - low_bond, 0, 255);

Serial.println(val,DEC);  //print on Matlab
```

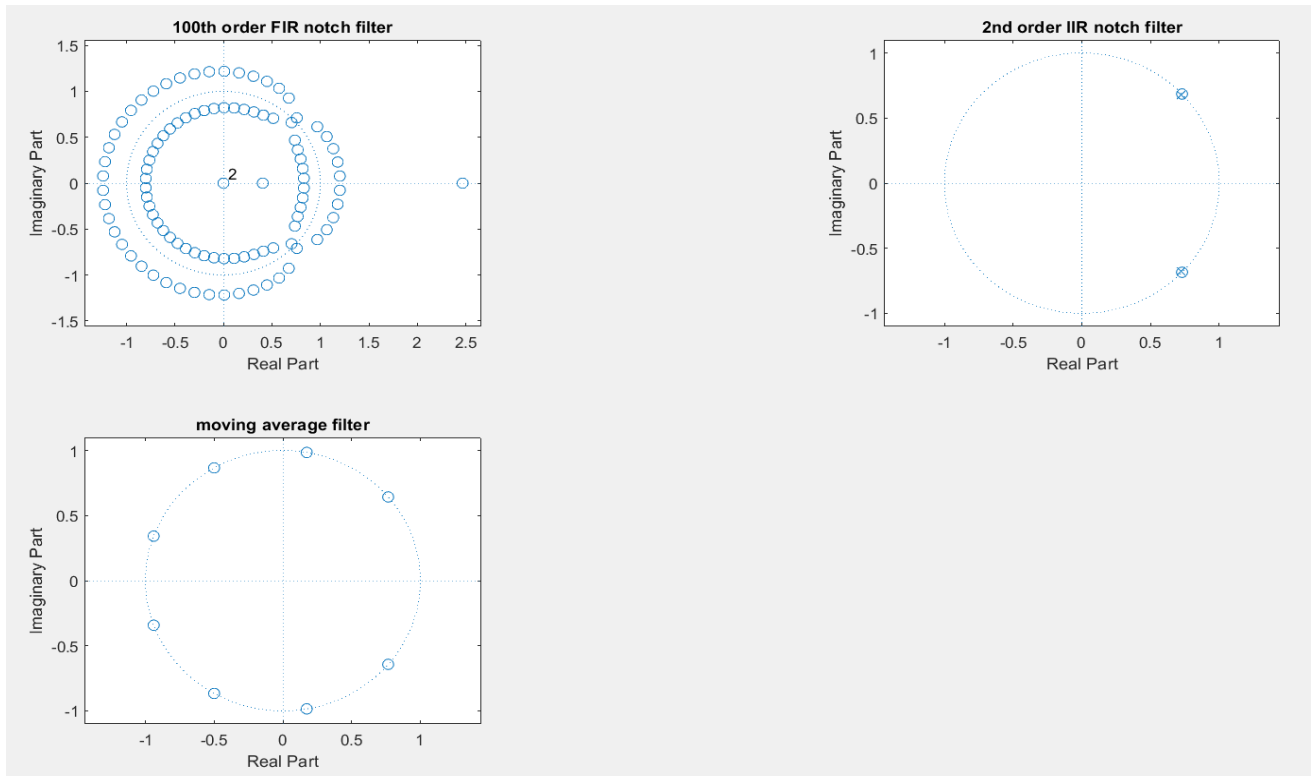
←再來將 ADC 收到的訊號先減掉下界，如此一來訊號範圍就變成 0 ~ (上界 - 下界)，然後再把 0 ~ (上界 - 下界) map 到 0~255，便能得到 8-bit 的 val，而且能保留心電訊號完整波型，並可以讓訊號塞滿 0~255 這些值。

4. Discussion :

Please compare and comment the results from your designed FIR (MATLAB function: fir1()), IIR (MATLAB function: iirnotch) notch filters and moving average filter.



↑三種濾波器 magnitude response 比較，藍線為 FIR notch filter(100 階)、紅線為 IIR notch filter(2 階)、橘線為 N-point moving average filter(N = 9)。



↑三種濾波器 pole-zero plot 比較，左上圖為 FIR notch filter(100 階)、右上圖為 IIR notch filter(2 階)、左下圖為 N-point moving average filter( $N = 9$ )。

從上頁的 magnitude response 比較可以發現，IIR notch filter 最接近理想的濾波器，而且階數最低，才 2 階而已，相對的 FIR notch filter 固然有 100 階，但 bandwidth 較大，可能把不想要濾掉的頻率訊號也給濾除，當然也可以讓 FIR notch filter 的階數再更高一些，使得 FIR notch filter 更接近理想狀況，不過這樣會造成 transient response 增加，也就是說不能用的資料量會增加，而且濾波器也會變得更複雜。

不過固然 IIR notch filter 在 magnitude response 的表現比較好，在階數上也較少，設計上會比較簡潔，但是它的 phase response 是 non-linear 的，相對而言，moving average filter(也是一種 FIR filter)和 FIR notch filter 的 phase response 是 linear phase 的，所以如果今天使用者需要考慮到 phase 的問題，就必須使用 FIR filter。

由 pole-zero plot 可以發現，IIR notch filter 之所以可以讓階數變得那麼小卻又有那麼小的 bandwidth，是因為有 2 個 pole，這兩個 pole 放置在單位圓內和兩個 zero 同個角度且相當接近兩個 zero 的地方，這樣可以使 notch filter 更加理想，因為越接近 pole，理論上 magnitude response 會越接近無窮大，但是 pole 和 zero 很近，所以 zero 又把 magnitude response 壓下來，兩者作用之下便造成上頁 magnitude response 的結果。

那 moving average filter 呢?它的好處是 magnitude response = 0 的地方是具有週期性的，所以不只有 60Hz、120Hz、180Hz、240Hz.....等共振頻率雜訊也可以濾掉，但缺點是由上頁的圖可以觀察到它 magnitude response 的 gain 並非 = 1，FIR notch filter 和 IIR notch filter 的 gain 均為 1，所以 moving average filter 可能造成濾完後的訊號有 magnitude distortion。

### 5. Extra stuff :

我覺得這次實驗其實可以直接使用現成的數據，因為這樣我們就不用再處理 ECG 電路的問題，除此之外，每個人收到的 ECG 訊號裡頭包含的雜訊不盡相同，像我收到的訊號 baseline wander 就很不明顯，如此一來本實驗第二題便比較難看到 high pass filter 對濾除 baseline wander 的效果。

### 6. Conclusion :

經過這次的實驗讓我看到了 FIR notch filter、IIR notch filter 和 moving average filter 的實際應用，並且了解三種設計方式的差別與各自的考量。這次的實驗更加深了我對 DSP 理論的了解，讓我體會到數位訊號處理實際會遇到的困難與選擇。

除此之外，我也學到在資料傳輸上應盡量用最少的空間來儲存資料，如此一來不但能避免浪費記憶體，更重要的是可以讓資料的傳送速率加快。

### 7. Reference :

(1) Discrete-Time Signal Processing, Alan V. Oppenheim, Ronald W. Schaffer

(2) DSP: Using an FIR filter to remove 50/60Hz from an ECG

<https://www.youtube.com/watch?v=r7ypfE5TQK0>

(3) Overview of FIR and IIR Filters

<https://www.youtube.com/watch?v=9yNQBWKRss4>