

FINAL PROJECT REPORT

期末專題解說及配分

<i>Final Project</i>	<i>Lab(Video)+ Exp. Analysis Report</i>
(%)	20%+90%

期末專題分兩部份：*Video Demo 及 Analysis Report (20%+90%)*

ECG 實驗及分析：*(Including: simulation, emulation and realization)*

共有四個實驗：

實驗一：將心律訊號以 1Hz sine wave 灌進去實體電路去做 Emulation，然後再與 Lab10 LTspice exp. 3 中的 Simulation 結果去比較差異，利用時域波形與頻域波形分別去比較結果並分析差異。(比較 Emulation & Simulation 有何差異?)

實驗二：與實驗一的實驗程序相同，其中差異是將 1Hz sine wave 換成人體 heartbeat wave 仿真訊號，分別去做電路的 Emulation(利用 AD2 將 ECG.csv 波形仿真並灌至電路中)及 Simulation(參考 Lab08 助教上課資料 Lab08 ECG sim.rar, 將人體 heartbeat wave 仿真訊號在 LTspice 電路中跑模擬)。

實驗三：由電極貼片拮取人體實際 ECG 訊號並直接實現在電路上，在這個實驗中要留意電極貼片及導線接點問題，可以鉻鐵加強焊點減少接觸電阻提高訊號品質。因此這個實驗你可以量出自己心律波形，這種體驗是很難得的。將量到的心律波形拍成影片上傳至 YT，並貼上連結檔，分享你的成就。

實驗四：此實驗是 Arduino 整合實驗，Lab09 已有相當訓練了，所以只要上傳影片結果即可，不需附程式碼。

最後有項 Task，就是把材料及設備歸還。視疫情狀況、安排歸還時間，公佈在 LINE 羣組上。(因這次所領材料單價高，若未歸還材料及設備，將通報系辦及學校處理)

1. Finger heart rate sensor, LTH1550
2. Analog temperature sensor, LM35
3. Ultrasonic distance sensor, HC SR04
4. AD2

再提醒幾點：

1. 按規定時間上傳期末專題結報。
2. Demo 改以拍攝影片，並將 video 上傳至 YouTube，在結報內貼上影片連結。

1. Video Demo (20%)

將你的實驗結果拍成影片，並在影片中說明實驗內容及展示實驗的結果；最後將影片上傳至 YouTube.
Make a video of your experiment results, explain the experiment content and show the results of the experiment in the video. Finally upload the video to YouTube.

實驗的遠距 Demo 注意事項：

1. demo 影片權限不可設成私人，若無法播放，該項 demo 成績以零分計。
2. 結報的表格填錯 demo 題，該 demo 以零分計。
3. demo 要能看得到實驗器材(包括麵包板/電子元件/Arduino/AD2)，若只有 show 結報的圖形，該項 demo 成績以零分計。
4. demo 影片一律上傳 YT，別用其它連結或.odt 檔，若無法觀看，該項 demo 成績以零分計。

<i>Exp.1: Simulation and Emulation by sine waveform</i>	
YT URL Link:	https://www.youtube.com/shorts/IMaNk2kbhzg

<i>Exp.2: Simulation and Emulation by heartbeat wave.</i>	
YT URL Link:	https://www.youtube.com/watch?v=98BEsHz0XbY

<i>Exp.3: Measure the ECG waveform result</i>	
YT URL Link:	https://www.youtube.com/shorts/yfQMppPMGko

<i>Exp.4 : Write down the Youtube link of your experiment 4</i>	
<i>(Just show the video demo)</i>	
YT URL Link:	https://www.youtube.com/watch?v=8j_qB50o_CI

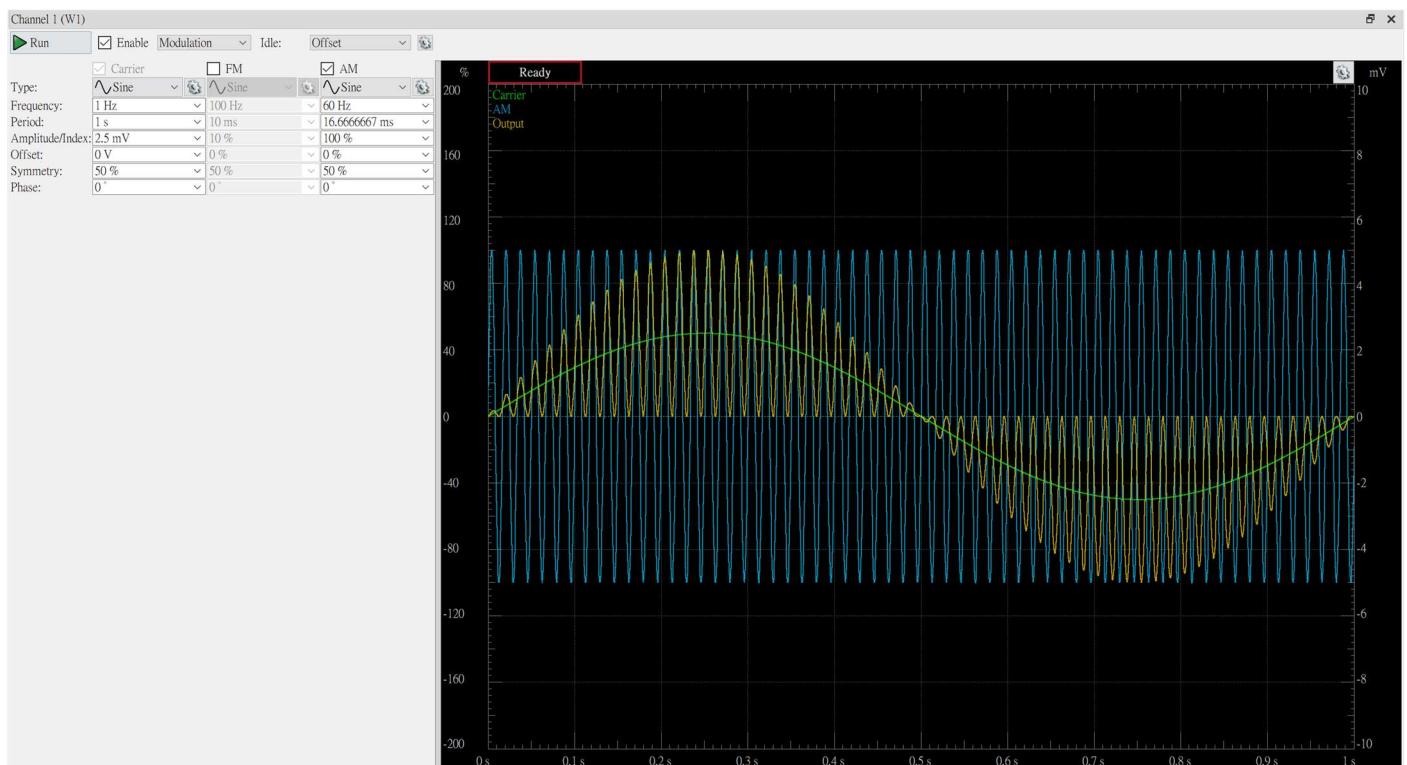
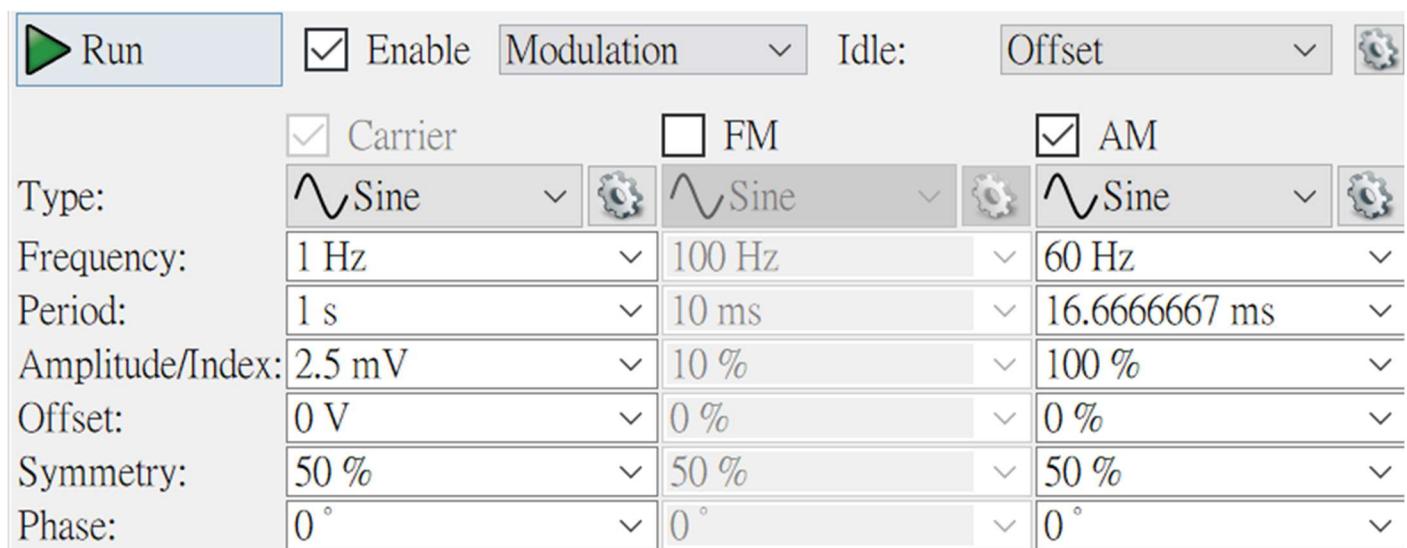
2. Analysis Report (90%)

A. 圖表(5%)及數據(5%)

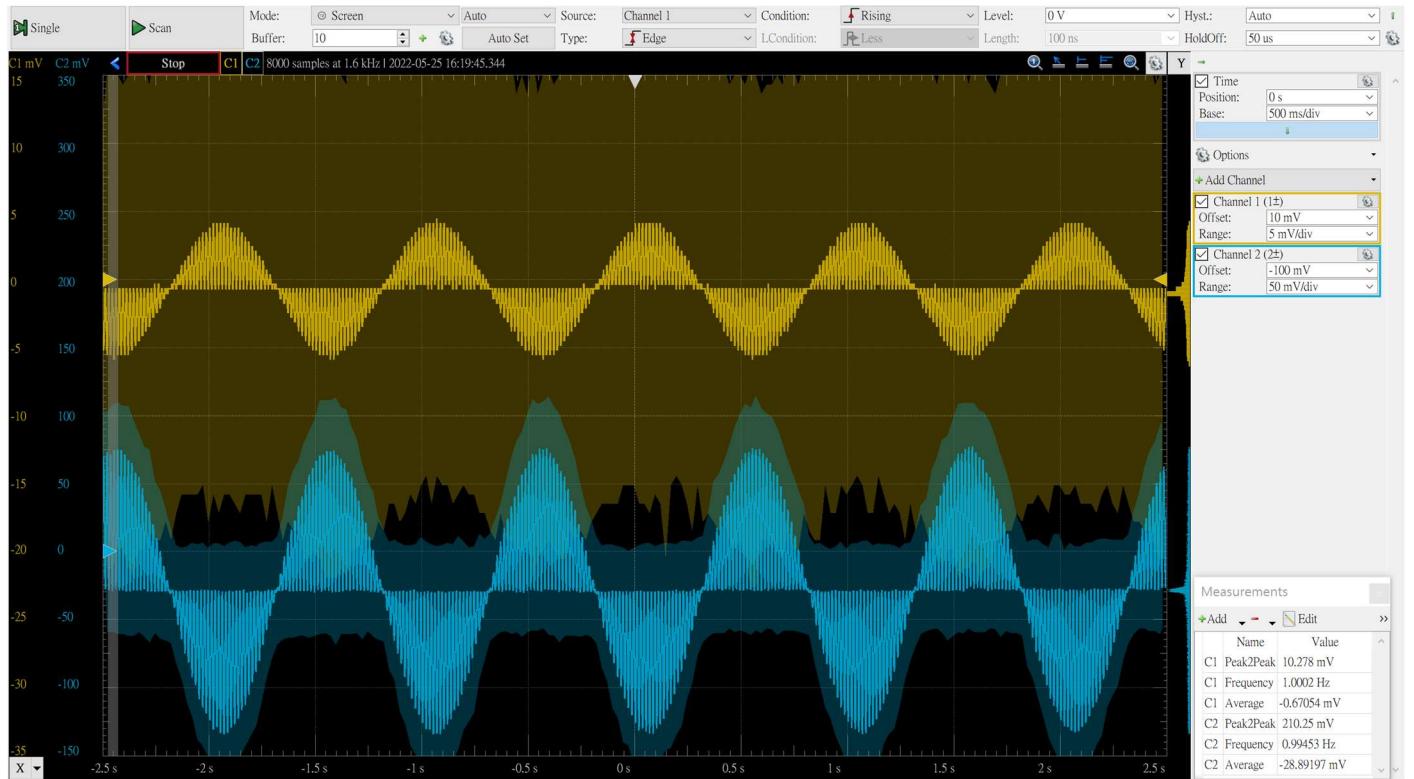
Exp. I: Simulation and Emulation by *sine waveform*

1. Paste the waveform of node: V1, Vx, Vo by AD2 scope.
2. Paste the waveform of node: V1, Vx, Vo by AD2 spectrum.
3. Compare the current experimental results with Simulation and Emulation.

輸入訊號設定如下(Wavegen)

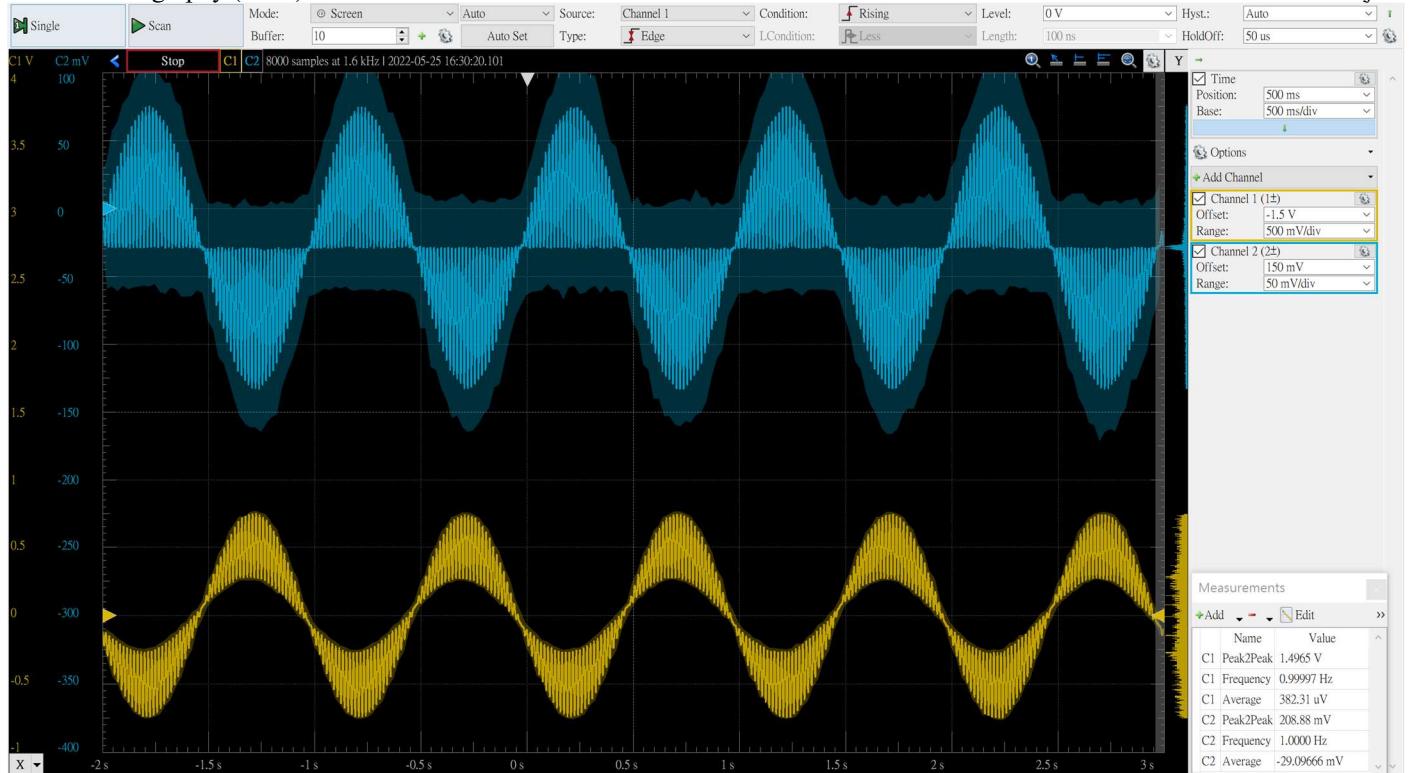


1. V_1 、 V_x 、 V_o 波型



	V_1	V_x
f	1 Hz	0.994 Hz
V_{pp}	10.278 mV	210.25 mV
V_{avg}	- 0.67 mV	- 28.89 mV
輸入輸出變化		
A_v	20.456	V/V
Phase shift	180	°

Electrocardiography (ECG)

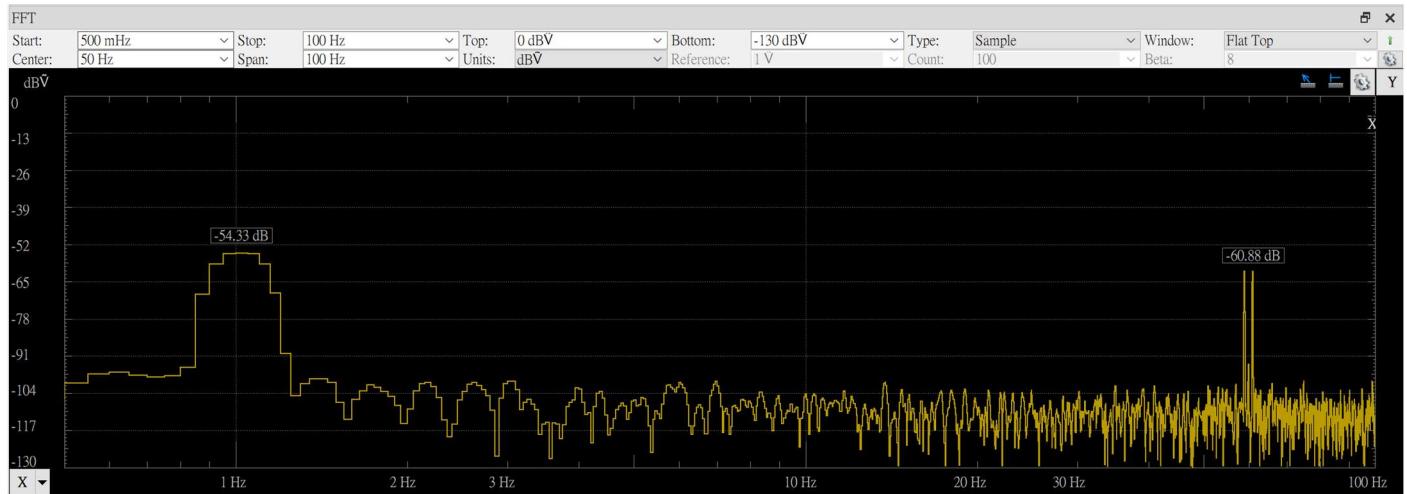


Final Project

	V_x	V_o
f	0.994 Hz	1 Hz
V_{pp}	0.209 V	1.5 V
V_{avg}	-29.1 mV	0.38 mV
輸入輸出變化		
A_v	7.177	V/V
Phase shift	180	°

2. V_1 、 V_x 、 V_o 頻譜

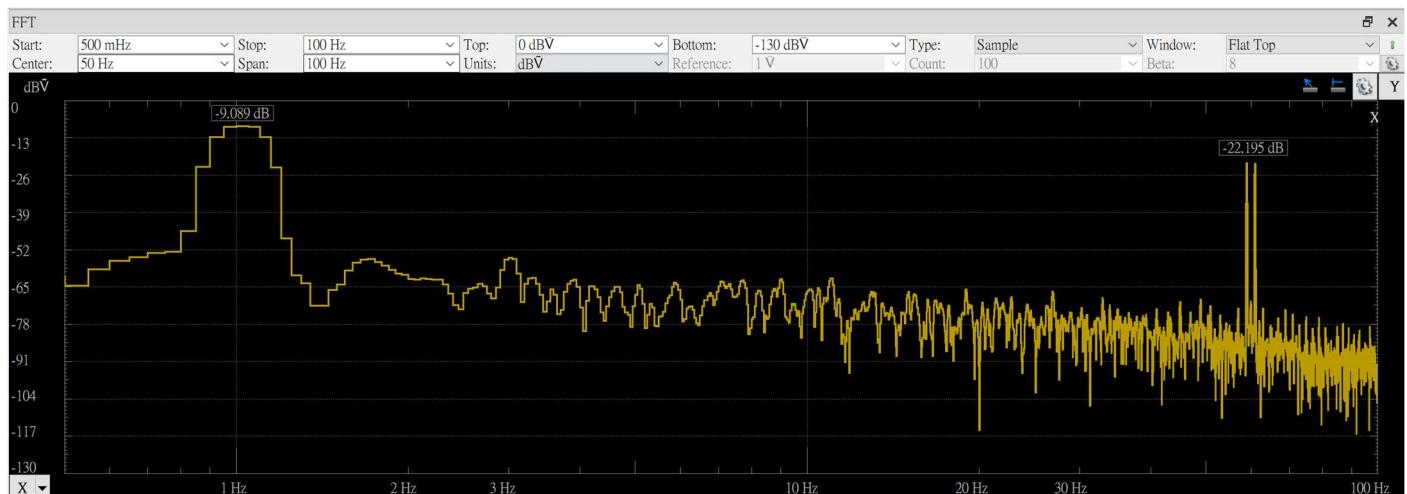
V_1



V_x



V_o



3. 模擬(simulation)與仿真(emulation)實驗結果比較

單位 (dBV)	目標訊號		雜訊	
	1Hz		60Hz	
	Simulation	Emulation	Simulation	Emulation
V ₁	-49.06	-54.33	-59.87	-60.88
V _x	-22.62	-28.75	-33.44	-35.30
V _o	-3.08	-9.09	-20.28	-22.2

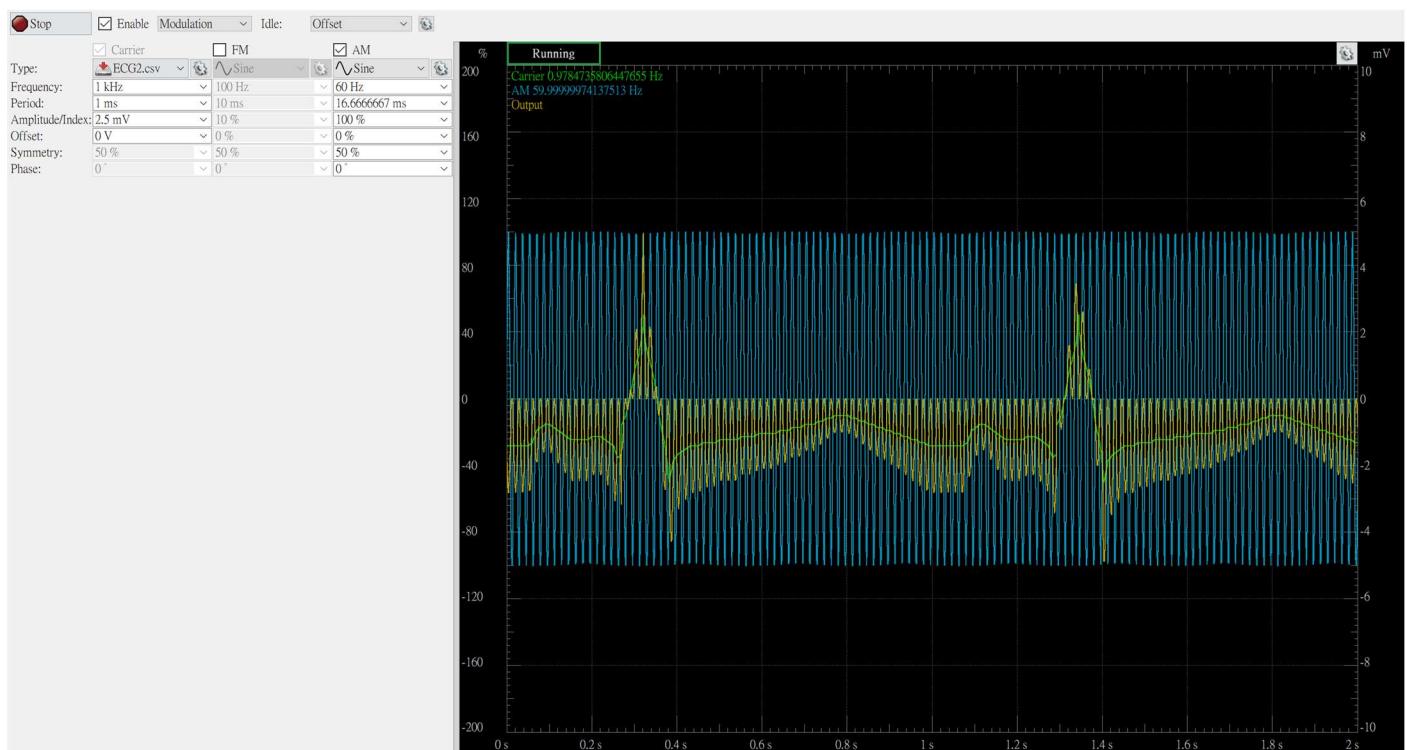
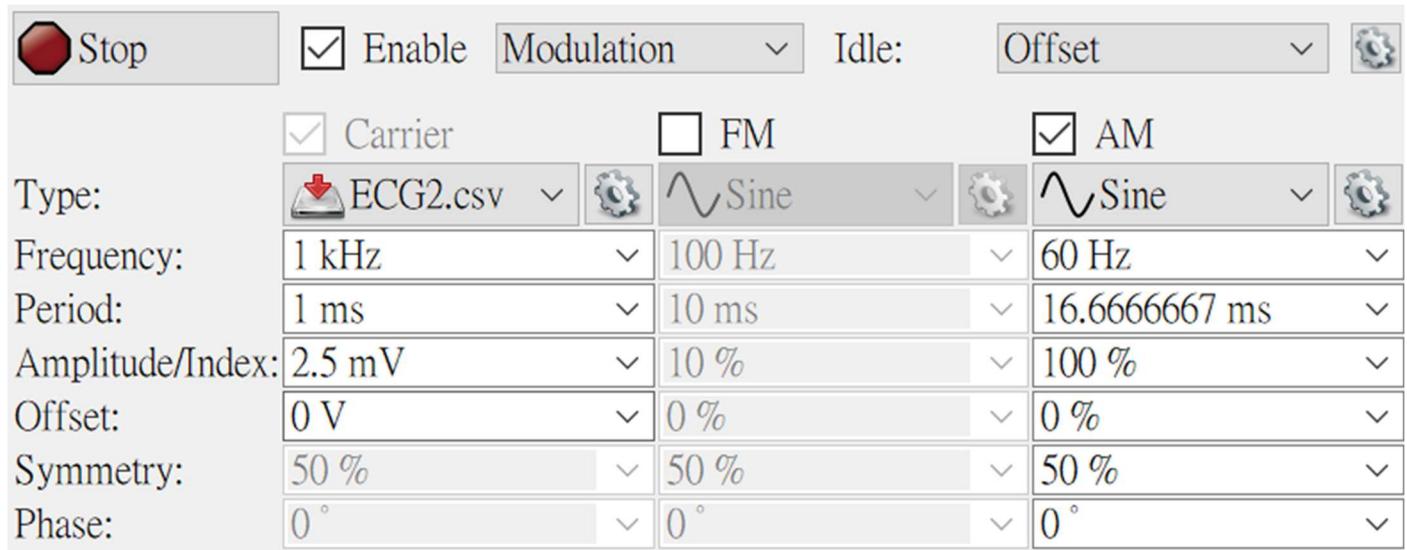
電路	訊號增益 (V/V)		相位差 (°)	
	Simulation	Emulation	Simulation	Emulation
儀表放大器	21	20.456	180	180
濾波器	7	7.177	180	180
總和	147	146.81	0	0

除 1Hz 的目標訊號在 Emulation 時皆比 Simulation 少大約 6dB 以外，其餘數值都非常相近，誤差很小。因此得出仿真實作的結果與模擬結果相符的結論。

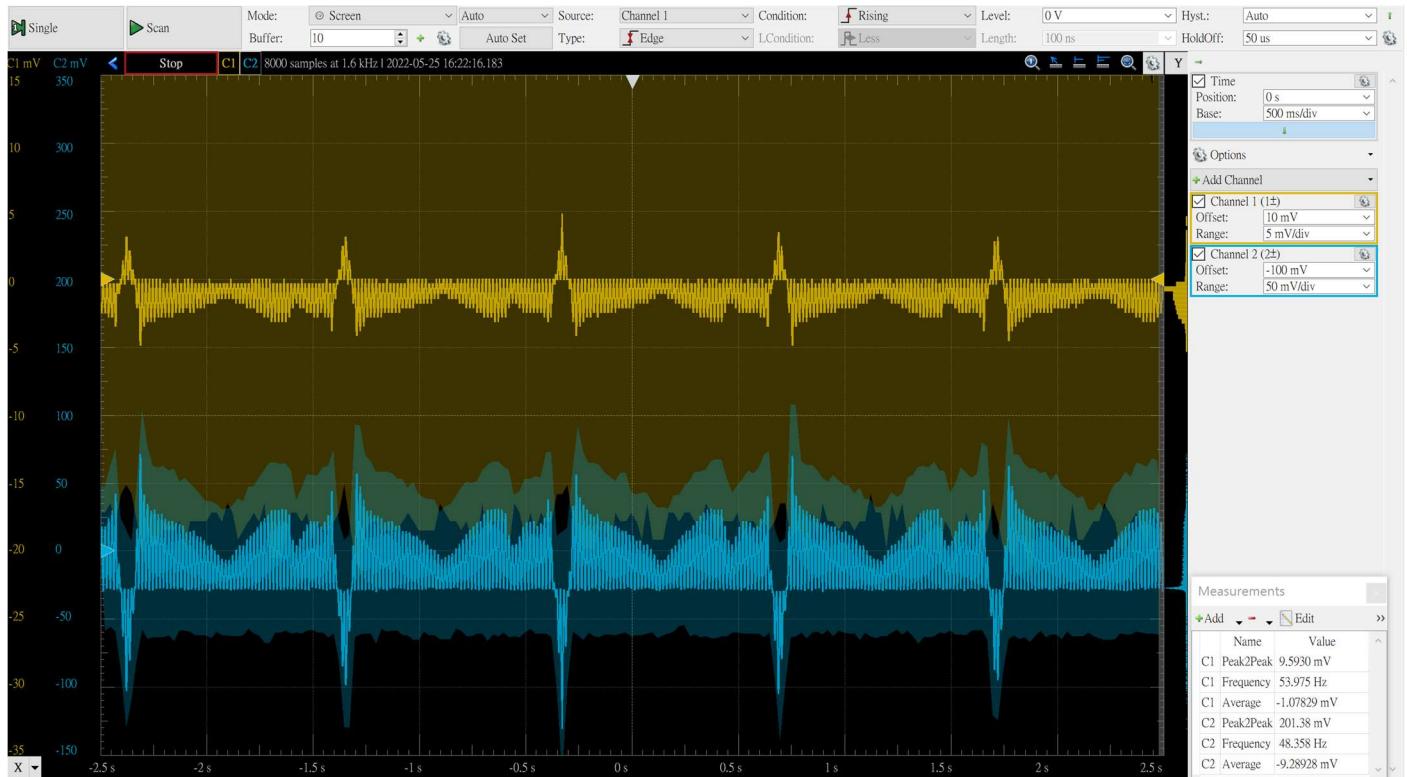
Exp.2: Simulation and Emulation by heartbeat wave.

1. Paste the waveform of node: V_1, V_x, V_o by AD2 scope.
2. Paste the waveform of node: V_1, V_x, V_o by AD2 spectrum.
3. Compare the current experimental results with Simulation and Emulation.

輸入訊號設定如下(Wavegen)



1. V_1 、 V_x 、 V_o 波型



	V_1	V_x
f	1 Hz	1 Hz
V_{pp}	9.593 mV	201.38 mV
V_{avg}	-1.078 mV	-9.289 mV
輸入輸出變化		
A_v	20.992	V/V
Phase shift	180	°

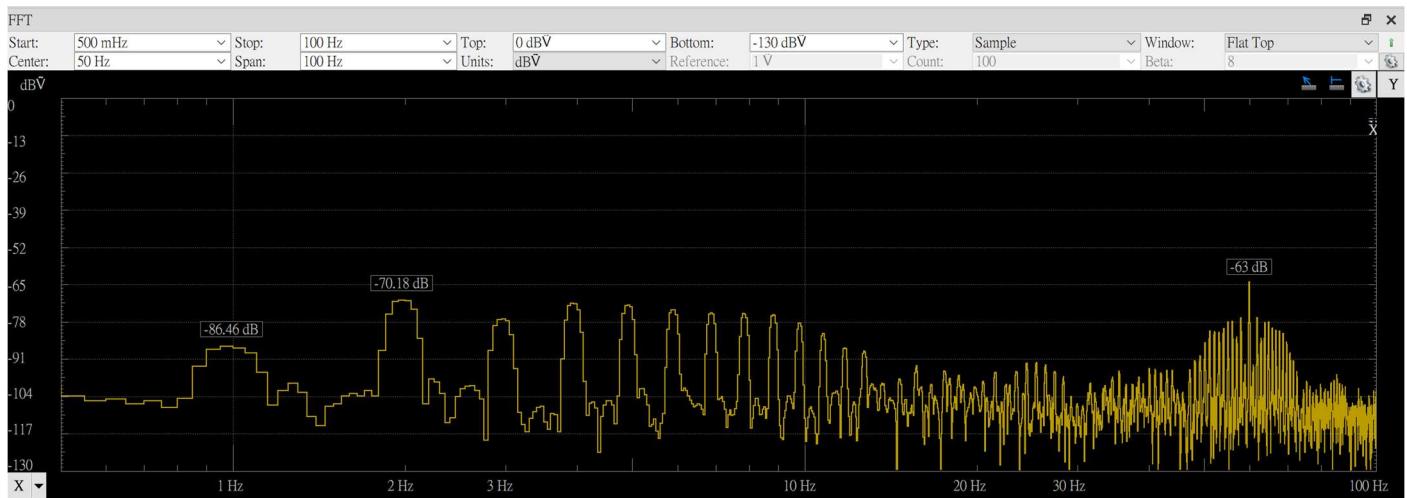
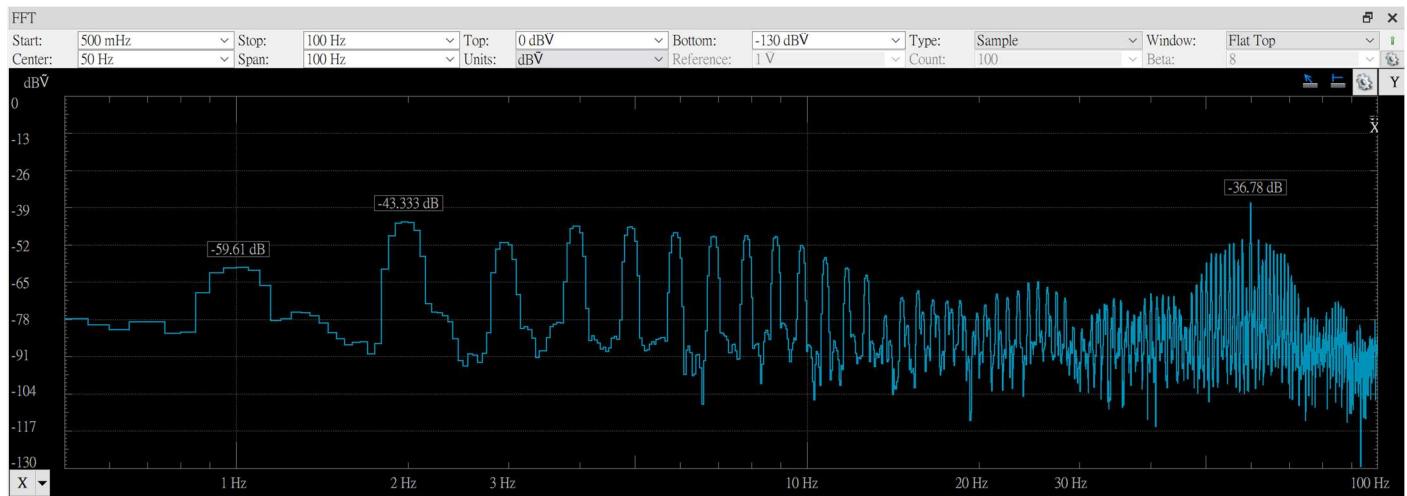
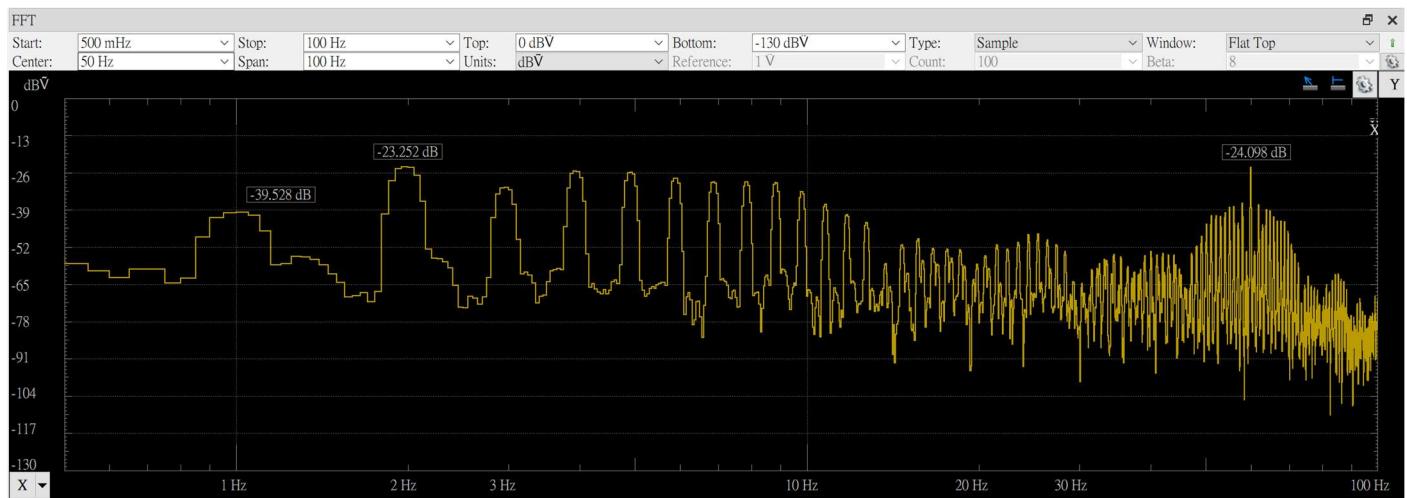
*此處紀錄的頻率為心跳波型的 1Hz，但儀器測量得出的頻率是雜訊 60Hz。

Electrocardiography (ECG)



	V_x	V_o
f	1 Hz	1 Hz
V_{pp}	0.201 V	1.292 V
V_{avg}	- 9.39 mV	-1.65 mV
輸入輸出變化		
A_v	6.43	V/V
Phase shift	180	°

2. V_1 、 V_x 、 V_o 頻譜

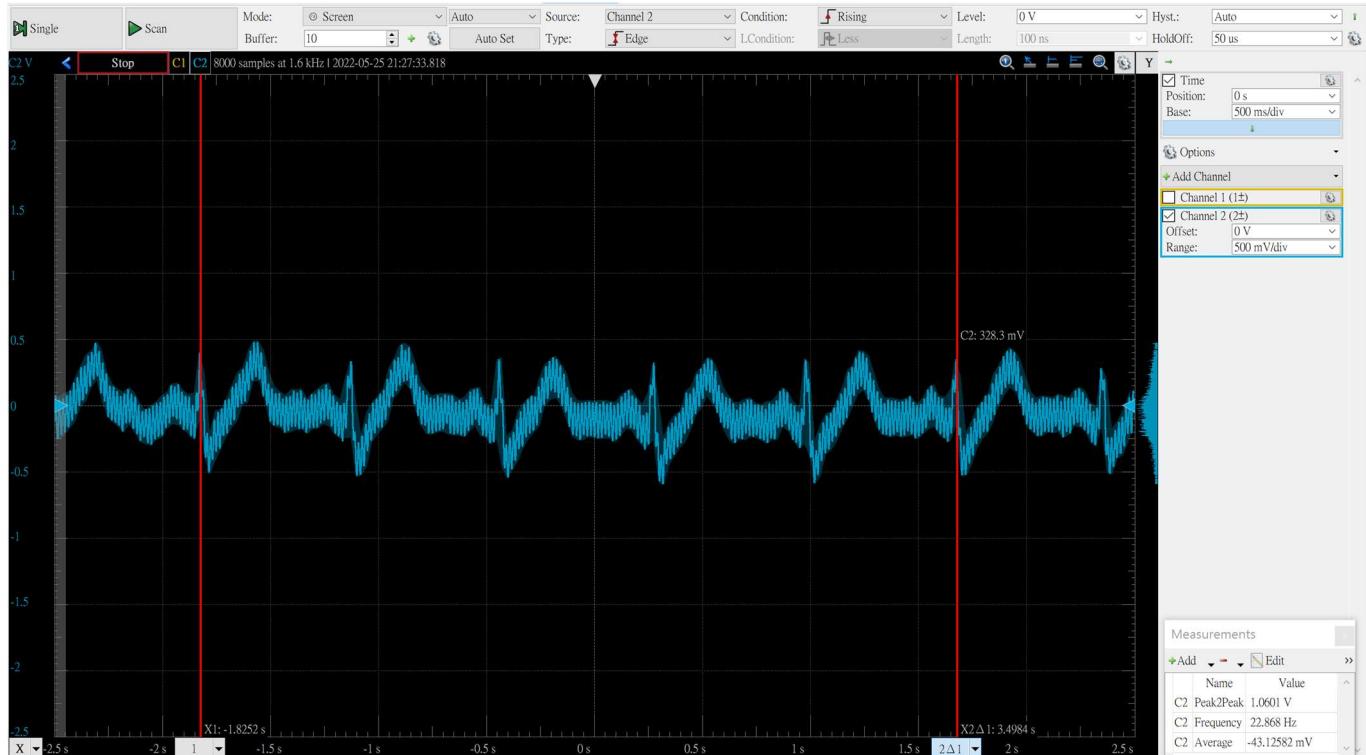
 V_1  **V_x**  **V_o** 

3. 模擬(simulation)與仿真(emulation)實驗結果比較

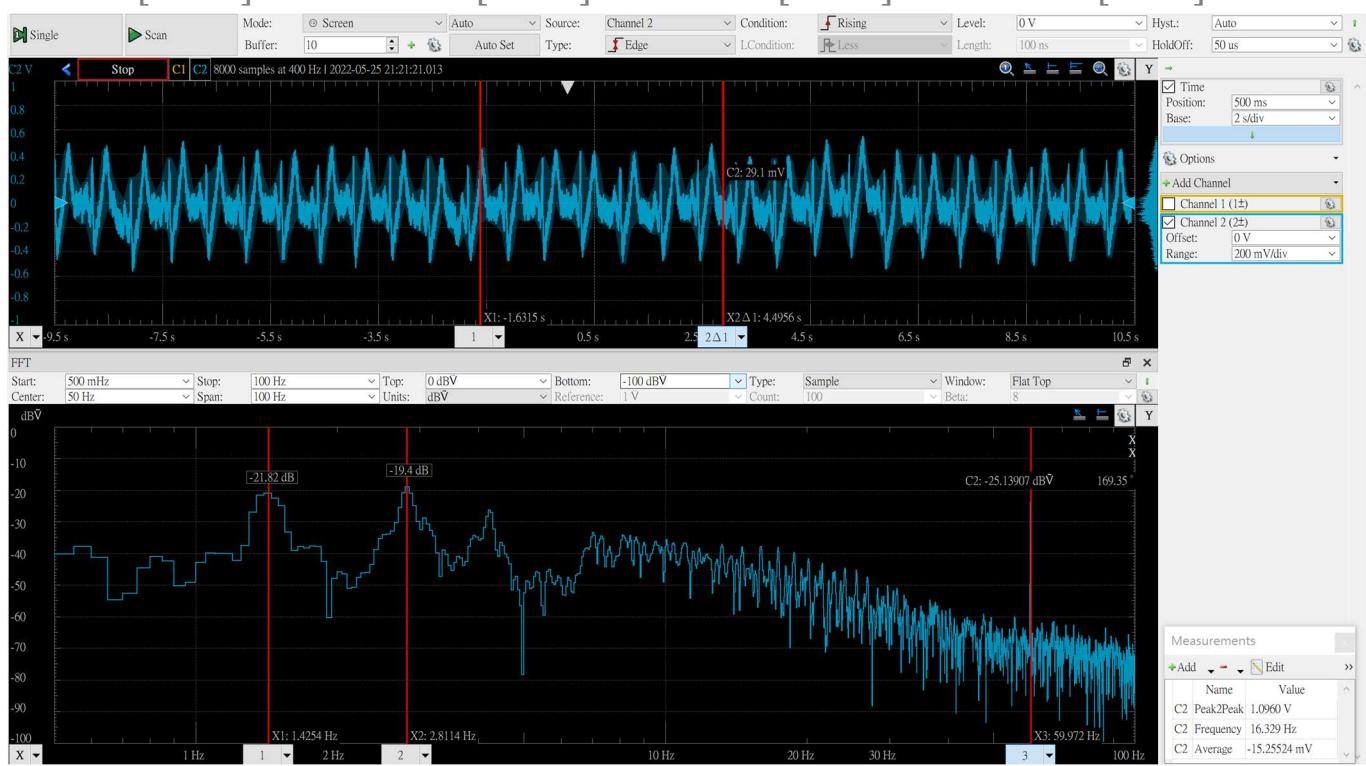
單位 (dBV)	目標訊號				雜訊	
	1Hz		2Hz		60Hz	
	Simulation	Emulation	Simulation	Emulation	Simulation	Emulation
V_1	-87.42	-86.46	-80.78	-70.18	-80.31	-63.01
V_x	-61.41	-59.61	-54.10	-43.33	-56.29	-36.78
V_o	-40.86	-39.53	-34.28	-23.25	-54.78	-24.11

電路	訊號增益 (V/V)		相位差 (°)	
	Simulation	Emulation	Simulation	Emulation
儀表放大器	21	21	180	180
濾波器	7.22	6.43	180	180
總和	151.71	135	0	0

模擬時，雜訊在 60Hz 有震幅較小且頻率有些微差異的雜訊，實際仿真電路時則可明顯觀察到較大且單一頻率的 60Hz 的雜訊，因此記錄到的 Emulation 雜訊都比 Simulation 雜訊多約 20dB。心跳的訊號在 1Hz 的大小非常接近，但是模擬時，頻譜上 3Hz 的訊號比 2Hz 的大一些，實際仿真操作時則是 2Hz 的訊號較大。訊號增益的部分濾波器實際操作時有比模擬再小一些。但仍在可接受的誤差範圍內，也可稍微調整電阻 R_9 來微調增益。

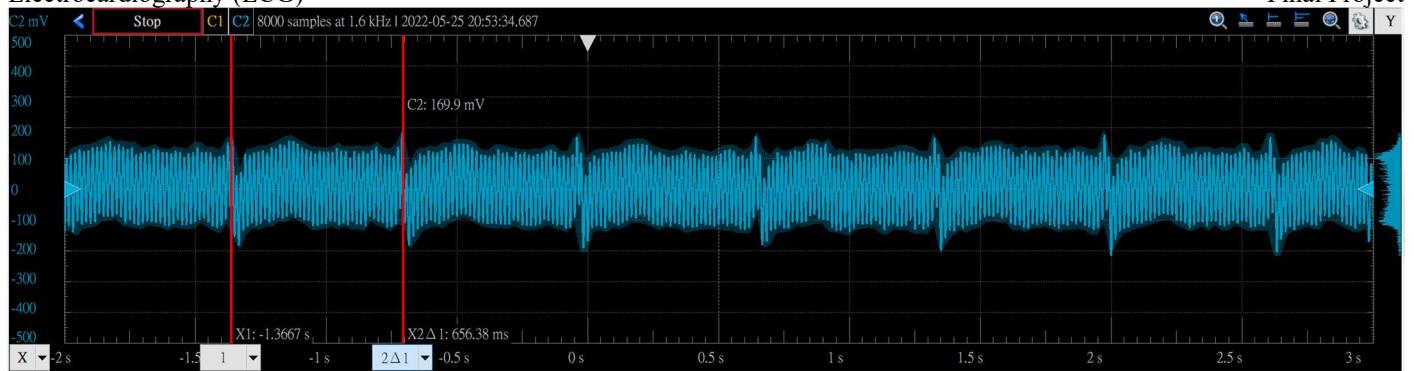
Exp.3: Measure the ECG waveform result**1. Paste your ECG waveform and measure the heartbeat frequency.****心跳 85.75BPM = 1.43Hz***為呈現最清楚的心跳波型，此處數據於量測時有調整 R₇ 與 R₉ 大小

FFT [1.4Hz]: -21.82dB [2.8Hz]: -19.4dB [4.2Hz]: -23.1dB [60Hz]: -25.1dB



下圖為未經濾波時觀測到的訊號(V_x 經過電容後的節點)，雜訊明顯大很多。

Electrocardiography (ECG) Final Project



B. 發現問題:(10%)

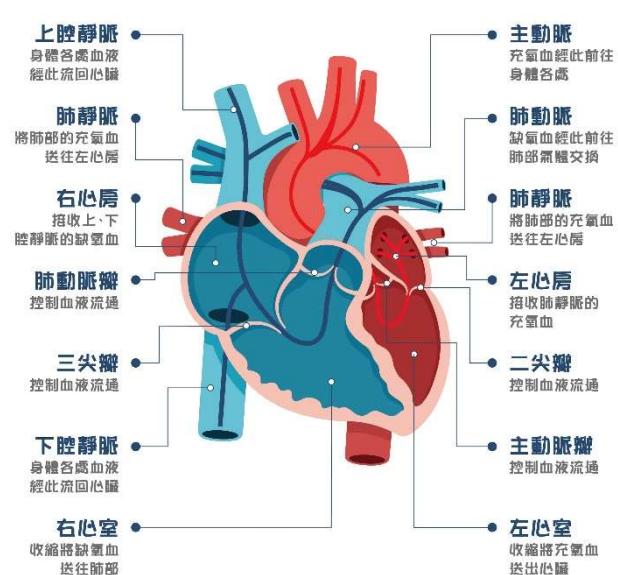
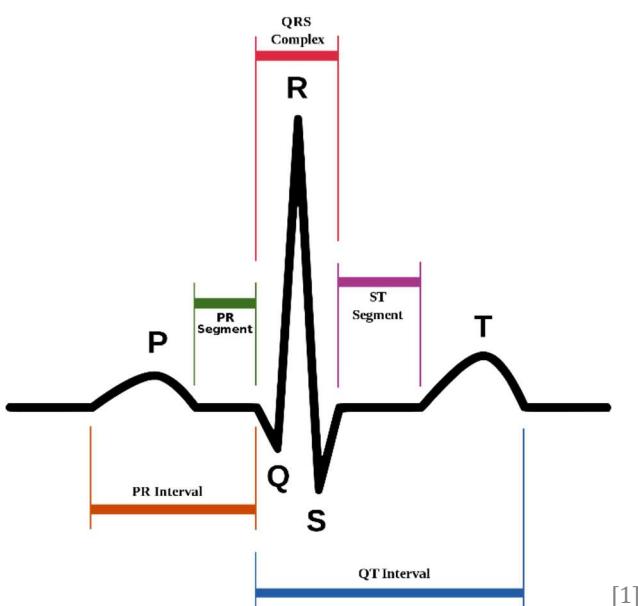
EXP1~3. ECG

Q1. 何謂 ECG ?

心電圖 (Electrocardiography 、簡稱 ECG) 是透過捕捉皮膚表面由心臟律動所造成的微小電位差(mV 等級) , 並將其波型記錄下來的診療技術。由於訊號微小 , 容易被雜訊影響 , 因此需要能較精準處理訊號的電路與放大訊號的電路。

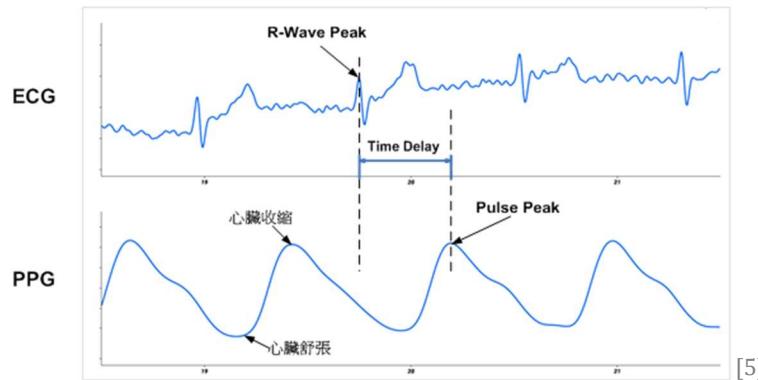
心電圖波型在一個週期中可分成 PQRST 五個波 , 範例圖如左下。實驗最容易觀測到的是 R 波與 T 波 , 尖尖的是 R 波 , R 波右邊山型的是 T 波。這也能解釋為何實驗圖表數據中 , 會有心律的整數倍的訊號出現。

血液在心臟中的循環是 靜脈 → 心房 → 心室 → 動脈 , 心電圖中的 P 波對應到心房收縮 , 此時血液由心房流向心室 , QRS 對應心室收縮 , 血液流入動脈。T 波則是心室舒張造成的訊號。



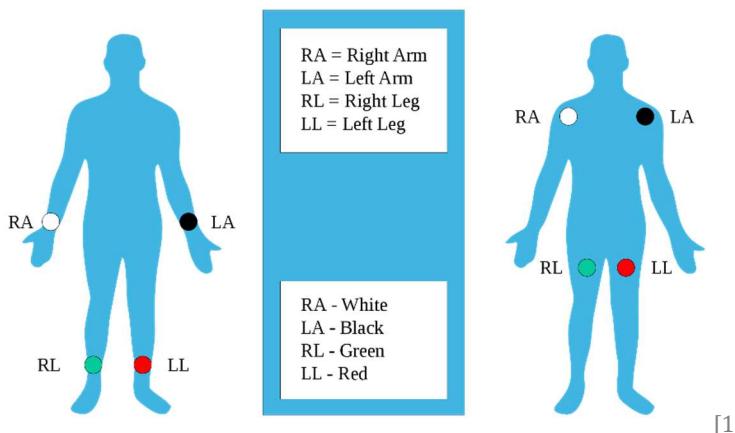
Q2. PPG 與 ECG 的差異 ?

在 Lab 8 使用 LTH-1550 量測心律所使用的技術叫做光體積變化描記圖法 (Photoplethysmography , 簡稱 PPG) , 與本次 ECG 不同。PPG 量測的是由心搏造成的血液容積變化 ; ECG 則是量測神經訊號傳遞造成的電位差。因此波型長得不同。另外 , PPG 只能量測出心律(HR) 且啟動後需要等待一段時間才能量到訊號 , 而 ECG 可以追蹤較細節的資訊 , 也是目前醫療所使用的方法。



Q3. 量測用的貼片要貼在哪？使用注意事項為何？

標準的 12 導程心電圖需要放置 10 個電極貼片，本次實驗僅使用 3 個電極貼片，分別黏在左右手臂以及右腳上，參考位置如下圖。須注意所有貼片皆須避免貼在骨頭突出的位置，且左右邊的貼片需對稱黏貼。電極貼片通常由導電凝膠、自黏合板及電纜組成，撕開後凝膠接觸到空氣會逐漸乾掉，因此要等實驗時再撕開保護膜，且不要更換太多次貼片位置，看好再貼。



Q4. 如何改善電路的濾波效果？

適當地調整 R_7 與 R_9 即可。降低 R_7 會使儀表放大器增益變大，增加 R_9 會使 f_{H3dB} 降低，同時增益也會變大。在調整時須注意波型有無超出電路最高或最低電壓範圍，將 f_{H3dB} 降太低也有可能造成波型失真。詳細的說明於電路分析[4.調整操作變因]。

Q5. 貼上貼片後站立時量不到訊號時該怎麼辦？

實驗過程觀察到量測到的訊號品質與大小與貼片位置及人體姿勢有關。採取坐姿且腳不要碰地較容易量出訊號，且雜訊較小。若量不太到訊號可嘗試稍微移動貼片位置，或改變姿勢。

EXP4. Arduino

Q6. 為何改變程式中代表模式的變數後需要 delay 一段時間才較不會有 Bug ?

避免硬體(實際電路)完成反向前就執行到下一步的程式，導致文字內容更新失敗，卡在溫度模式。

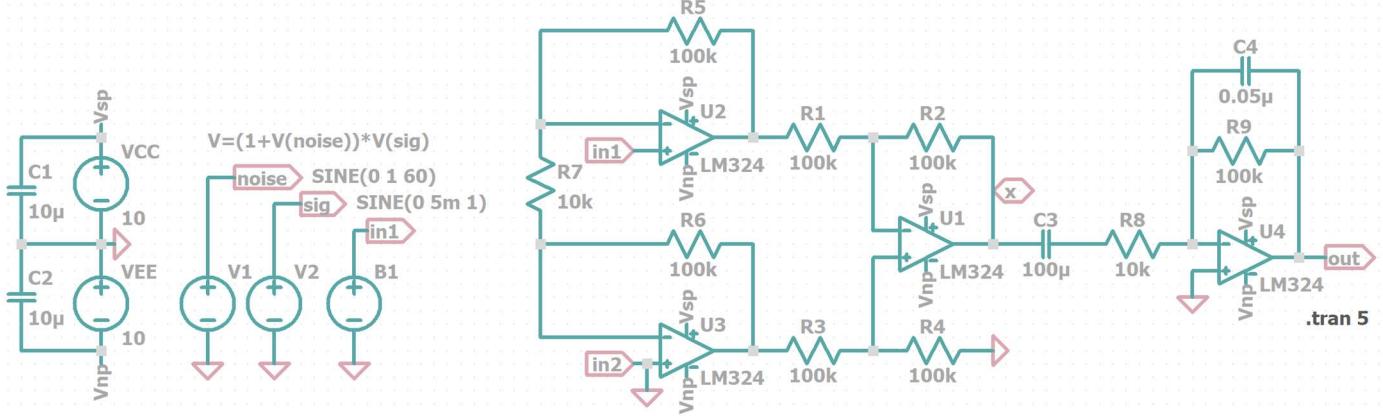
Q7. 為何燒錄程式時若 Pin 0,1 未維持開路可能導致燒錄失敗？

Arduino UNO 板在燒錄程式時會使用到 TX/RX 腳位，若連接的電路影響燒錄的訊號傳送，就會導致燒錄失敗。

Q8. 為何用 Pin 0,1 控制七段顯示器 VCC 時使用 Serial print 會導致七段顯示器無法正常顯示？

除了 digitalWrite，使用 Serial.print 等串列傳輸的指令也會用到 TX/RX 腳位，因此當兩者功能相衝突時就會出錯。

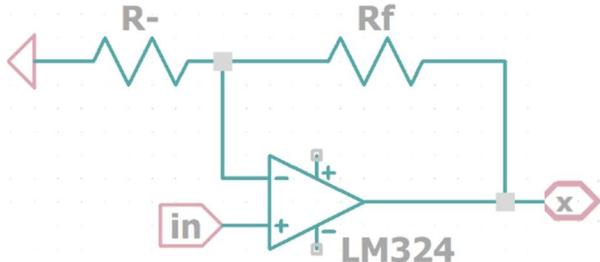
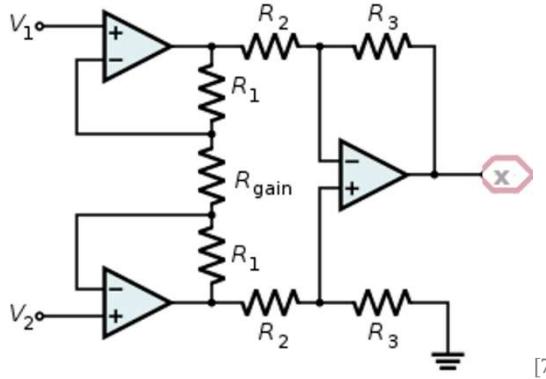
C. 電路分析+結合LTspice分析(40%)



電路分析與模擬

此電路為 Lab10 EXP3 的電路，由儀表放大器與帶通濾波器組成。為求版面精簡整潔，以下圖例將省略電源的部分，被省略的 VCC 為 5V，VEE 為 -5V，實作時，電源由 AD2 的 V+、V-供電。

1. 儀表放大器 Instrumentation Amplifier



上圖(左)為一儀表放大器，利用半電路簡化分析。上圖(右)為儀表放大器第一級的半電路，與左圖的對應為 $R_- = \frac{R_{gain}}{2}$, $R_f = R_1$ 。運用疊加原理、分壓定律及放大器輸入兩端為虛短路的概念分析可得

$$V_{in} = V_+ = V_- = \frac{R_-}{R_- + R_f} \times V_{out1} \Rightarrow V_{out1} = \left(1 + \frac{R_f}{R_-}\right) V_{in} = \left(1 + \frac{2R_1}{R_{gain}}\right) V_{in}$$

接著考慮第二級放大，並把半電路整合得

$$V_x = \frac{R_3}{R_2} \left(1 + \frac{2R_1}{R_{gain}}\right) (V_2 - V_1)$$

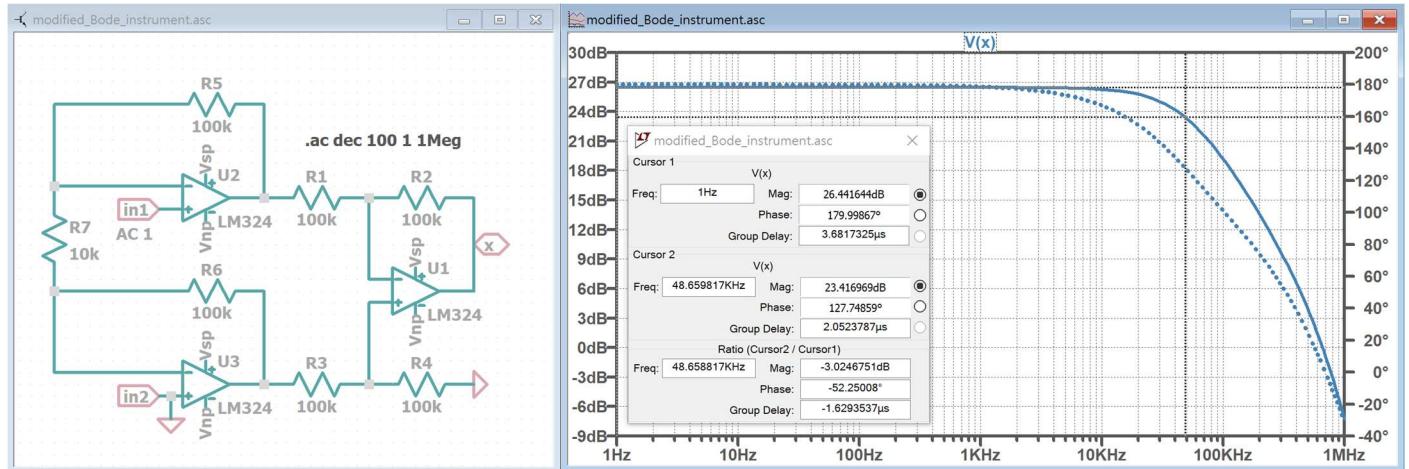
代入本題實驗指定數值，除了電阻 $R_{gain} = 10k$ 以外，其餘電阻均為 100k 因此

$$V_x = 21(V_2 - V_1)$$

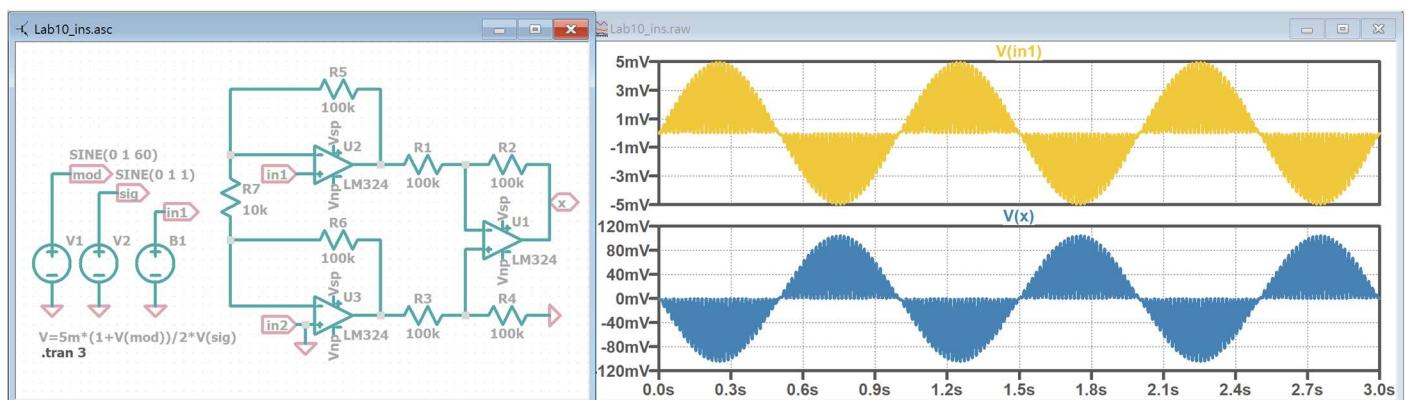
若將輸入正極接 V_1 、負極接 V_2 ，可由上式推得經過此電路後相位差為 180 度。

接著使用 LTSpice 模擬並驗證此推測。

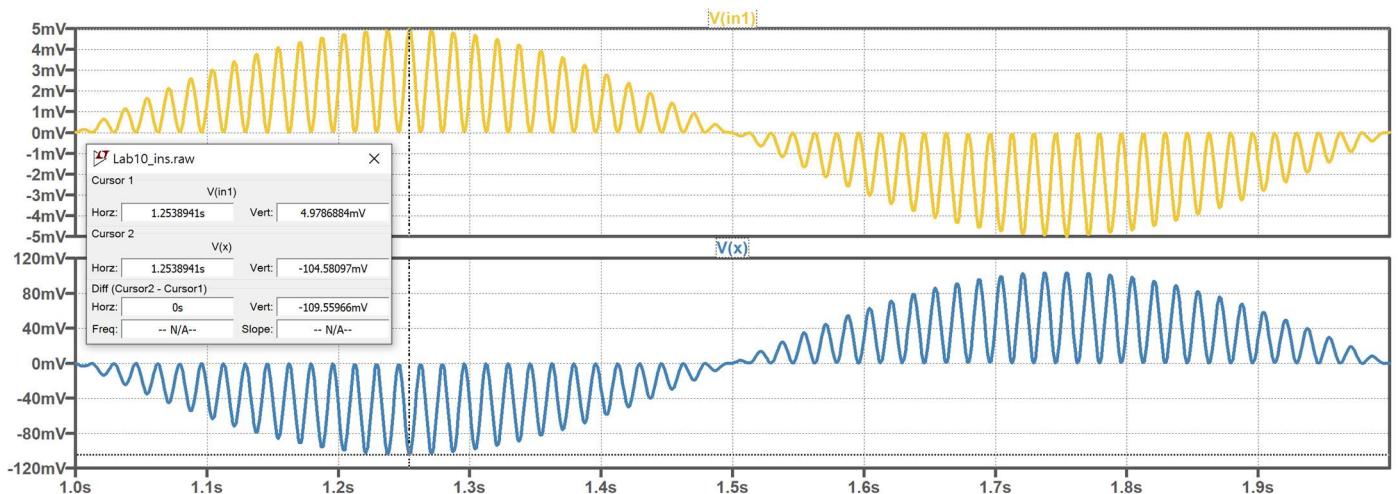
透過在節點 **in1** 輸入訊號 **AC 1** 模擬出 Bode plot，使用指令為 **.ac dec 100 1 1Meg**。此儀表放大器的半功率點(f_{3dB})約為 50kHz，而這次實驗著重的頻帶在 1~60Hz 附近，因此儀表放大器電路的頻率響應在我們感興趣的頻帶範圍內幾乎沒有影響。



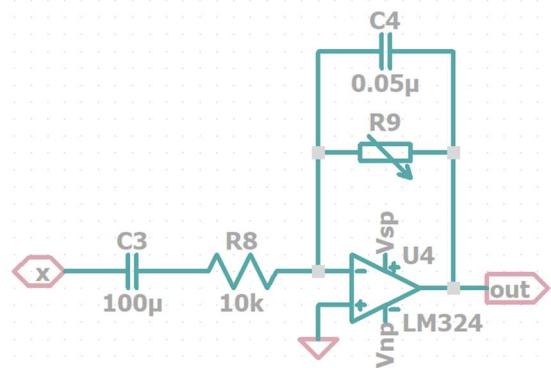
接著一樣在 **in1** 輸入訊號，指令如圖，這次輸入的是調變訊號，目的為觀察此電路在時域的行為。



黃色為 **in1**，藍色為 **x**。 $V_{in1,pp} = 10mV$ ， $V_{x,pp} = 210mV$ 。如同預期，經過此電路震幅被放大 21 倍，且相位差為 180 度。



2. 帶通濾波器 Bandpass Filter



Zeros

Z_1	0
-------	---

Poles

p_1	$\frac{1}{R_8 C_1}$	0.16 Hz
p_2	$\frac{1}{R_9 C_2}$	$\frac{10}{\pi R_9}$ MHz

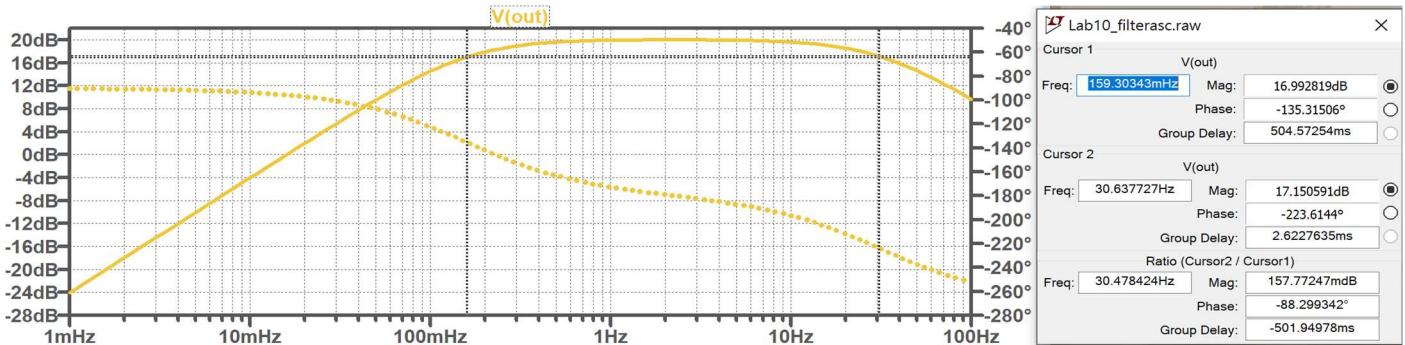
極點與零點位置可由以下轉移函式推得

$$H(s) = \frac{Z_{C_2} || Z_{R_9}}{Z_{C_1} + Z_{R_8}} = \frac{\frac{1}{R_9} + sC_2}{\frac{1}{sC_1} + R_8} = \frac{sR_9C_1}{1 + s(R_8C_1 + R_9C_2) + s^2R_8R_9C_1C_2} = \frac{sR_9C_1}{(1 + sR_8C_1)(1 + sR_9C_2)}$$

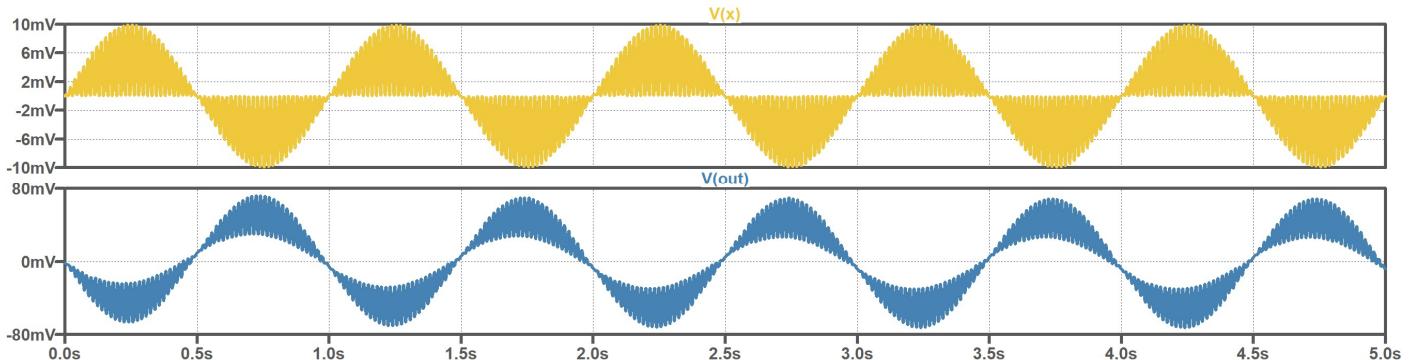
因為實作時 R_9 會放 100k 以上的電阻，導致 p_2 大於 p_1 ，因此 f_{H3dB} 主要由 p_2 決定， f_{L3dB} 則主要由 p_1 決定。此濾波器在頻寬內電壓放大倍率可以快速估算，其值大約為 $\frac{R_9}{R_8}$ 。此電路的操作變因為 R_9 ，透過調整 R_9 ，濾波器的增益與頻寬都會受影響。

先使用 Lab 10. EXP 3 的電路模擬，將 R_9 以普通電阻取代並設為 100k。 $p_2 \approx 31.8\text{Hz}$ 。

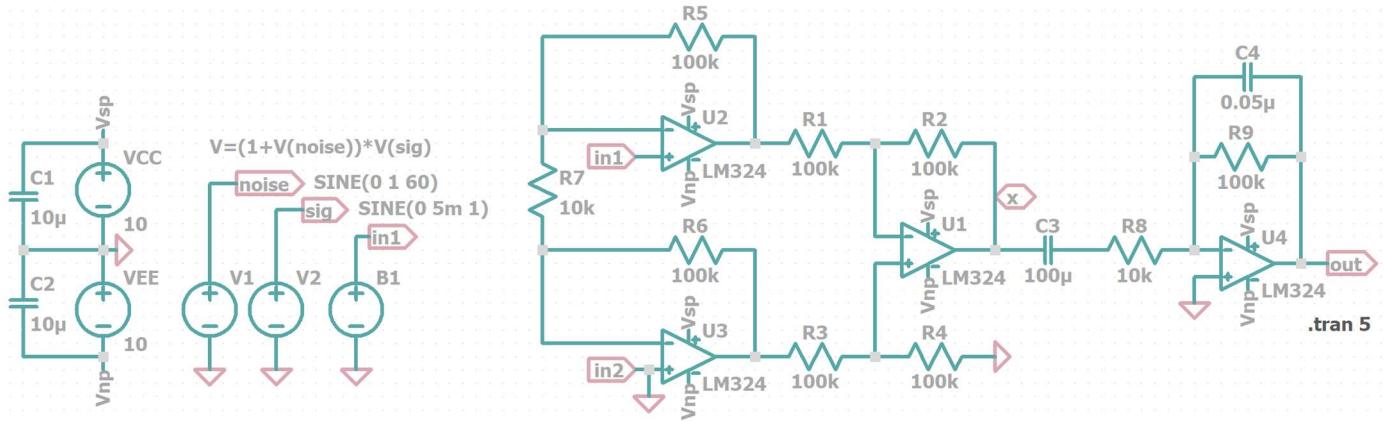
下圖中，可得知 $f_{L3dB} = 0.16\text{Hz}$ 、 $f_{H3dB} = 30\text{Hz}$ ，符合預測。且 1Hz 處的相位差約為 180 度。



雖然 1Hz 在頻帶內，60Hz 則否，但 1Hz 處的增益約為 20dB，60Hz 處的增益約為 13.4dB，其實只差 $10^{\frac{20-13.4}{20}} = 2.1$ 倍左右，因此在時域的輸出仍可明顯觀察到 60Hz 的部分，如下圖。



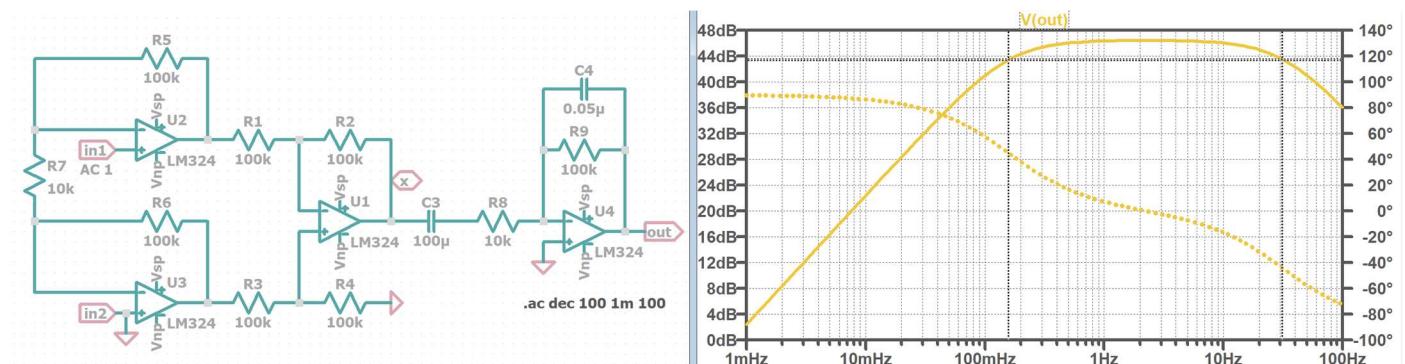
3. 整合



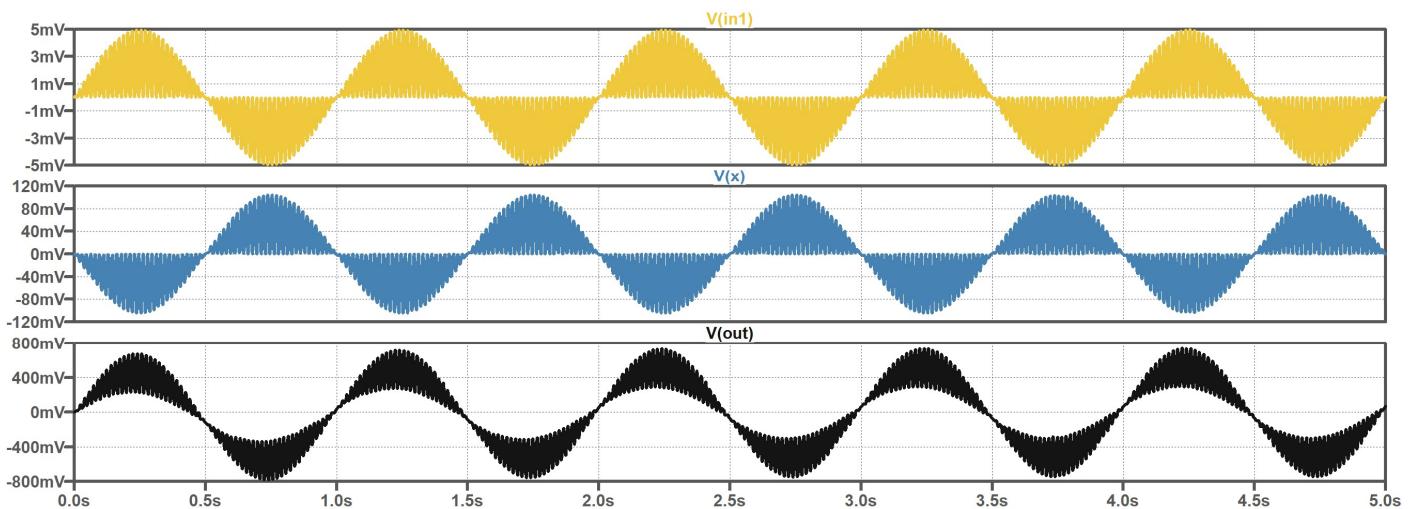
上圖為 Lab10. EXP 3 的完整電路，轉移函數為前兩個電路轉移函數的乘積，頻帶主要由濾波器決定。

$$H(s) = \frac{-s^2 21 R_9 C_1}{(1 + s R_8 C_1)(1 + s R_9 C_2)}$$

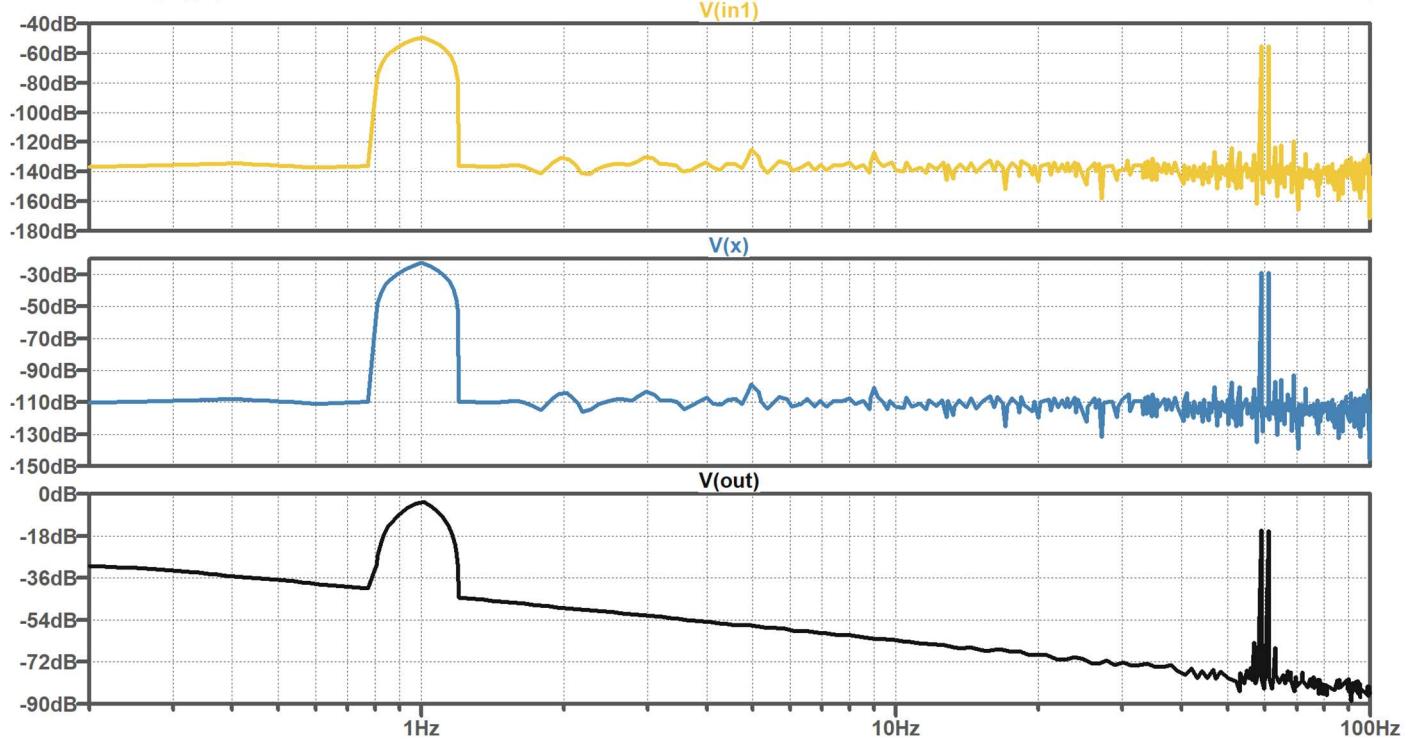
下圖為經由 LTSpice 模擬出的 Bode plot。 $f_{L3dB} = 157\text{mHz}$ 、 $f_{H3dB} = 31.1\text{Hz}$ ，很接近濾波器的半功率點。



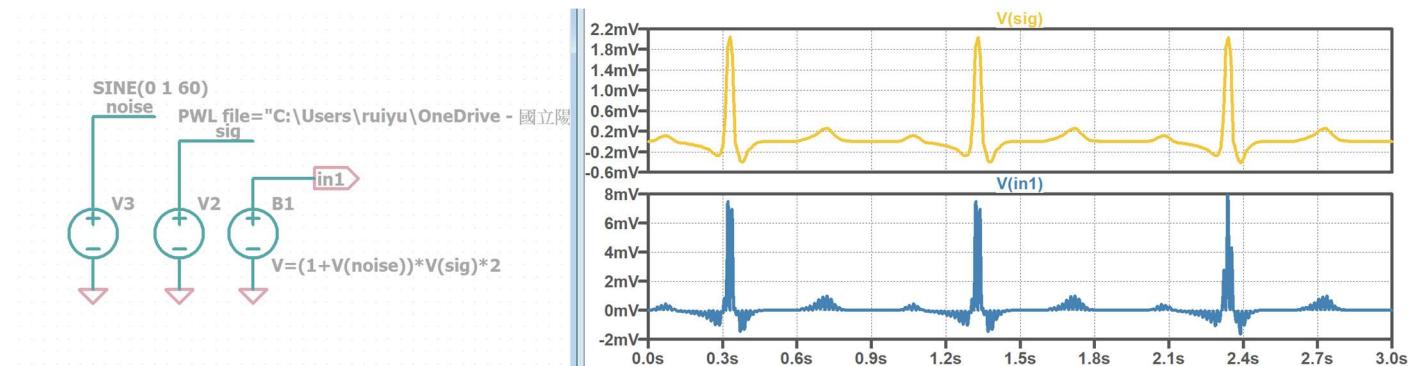
1Hz 訊號經過儀表放大器後，放大 21 倍，反向一次；經過濾波器後，放大 20dB，也就是 10 倍，再反向一次。因此可估算出電壓大約會被放大 210 倍，相位差為 0，如下圖。



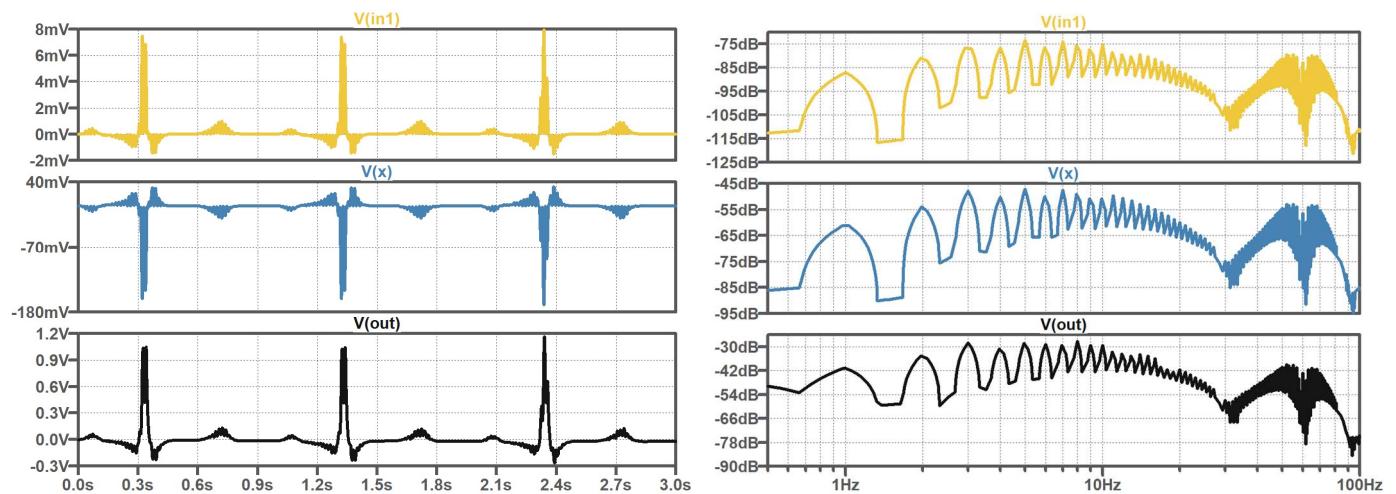
$V_{in1,pp} = 10\text{mV}$ 、 $V_{in1,pp} = 210\text{mV}$ 、 $V_{in1,pp} = 1.47\text{V}$ 。儀表放大器電壓增益為 21 倍，濾波器增益為 7 倍。進行 FFT，得到下圖。此結果仍然較難看出 60Hz 雜訊有被濾除的跡象。



接著模擬有雜訊的心跳訊號，使用 **PWL file = “path”** 開啟儲存模擬心跳訊號的.txt 檔，再運用震幅調變的方式與雜訊混合，右下波型圖黃色為.txt 檔儲存的波型，藍色為經過調變的結果。



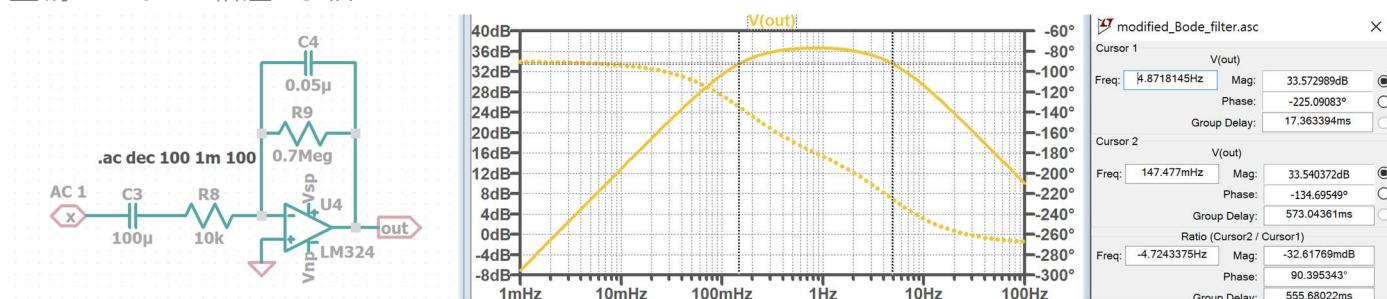
以下為輸入模擬心跳後對應的輸出。 $V_{\text{in,pp}} = 9.36\text{mV}$ 、 $V_{x,\text{pp}} = 196.55\text{mV}$ 、 $V_{\text{out,pp}} = 1.42\text{V}$ 。



4. 調整操作變因

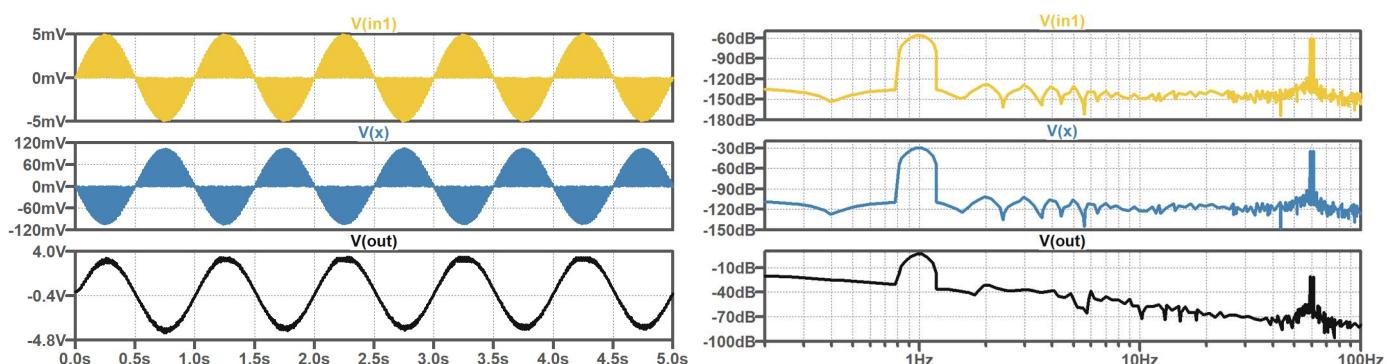
以上使用 Lab 10. EXP3 電路模擬的結果可觀察出 60Hz 的雜訊在經過濾波器之後，確實有被過濾掉一些，但濾波效果不太好，尤其在頻域更難看出來。因此有必要針對電路的頻帶與增益進行調整。這次實驗的電路主要有兩個操作變因，分別是 R_7 與 R_9 。 R_7 相當於前面電路分析的 R_{gain} ，由前面推導出的公式可知， R_7 越大，儀表放大器的增益越小。 R_9 則是會同時影響濾波器的 f_{H3dB} 與增益。 R_9 越大， f_{H3dB} 越小，增益越大。因此透過適當增加 R_9 的電阻值，可以更好地抑制 60Hz 的雜訊。調整 R_9 時，除了要注意 f_{H3dB} 有沒有降太低以致影響心律波型以外，還要注意放大後的訊號會不會超過 5V 造成波型被截斷失真，這個部分可以透過選用較大的 R_7 來達到減少儀表放大器增益的效果。

經過多次嘗試後，總結出將 R_9 調整成 700k，整個電路在濾波及放大訊號兩個部分皆有較佳的表現。另外，藉由模擬得知調整過的濾波器 f_{H3dB} 大約在 5Hz 左右，1Hz 增益約為 36.6dB，60Hz 的增益為 14.4dB，相差 13 倍。

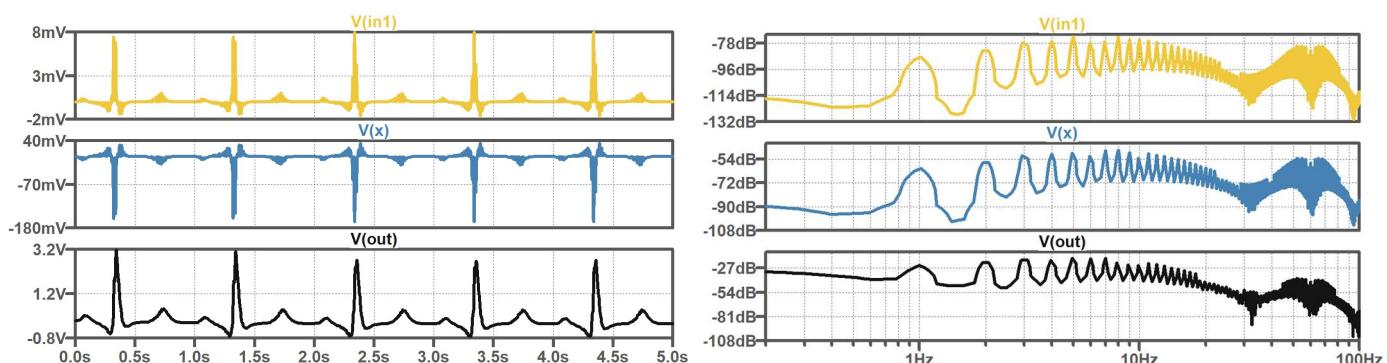


改良後電路模擬結果如下，左側為時域，右側為頻域。

輸入調變後的弦波



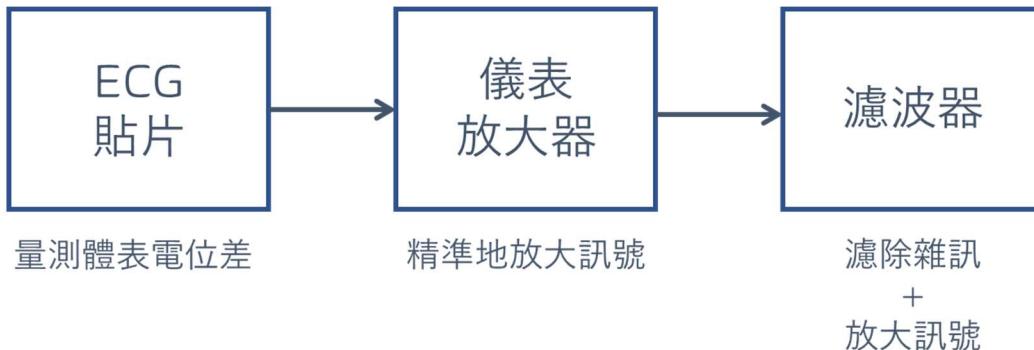
輸入調變後的心跳



無論時域、頻域、輸入的波，濾波效果皆有得到顯著改善，且波型保持完整。

D. 心得+結論(20%)

本次實驗架構如下圖：



這次的實驗很有趣，可以看到自己的心電圖(好像是在醫院一樣??)，也讓我體會到差動電路與儀表放大器的重要性。雖然雜訊時大時小，很難完全消除有點可惜，但也以此為契機去查了許多跟心電圖相關的知識，知道圖形中對應心臟怎麼動。實驗過程中只要不小心移動到，量測到的波型就會大亂，而且我需要整個人坐在椅子上，腳也放上去才量得到訊號，可能家裡地板上有會影響量測的因素存在，但我還沒找到答案。有時候還會不小心被電到，稍微點麻麻的(做實驗前我有先查過，這次實驗的電壓被電到也不會出事，所以就放心地被電??)。

在後來寫結報查資料的時候我也發現一開始不管怎麼樣都量不到訊號可能是因為我把貼片貼得離骨頭太近，跟 PPG 不同的是，不是脈搏比較大的地方訊號就比較強，一開始被這個觀念誤導了，幸好後來貼其他位置後有量到訊號，虛驚一場。實驗過程中做了不少嘗試，跟寫程式一樣，有時候有些 Bug 都是藏在枝微末節的地方，雖然基礎但是需要花時間慢慢除錯。例如放大倍率調錯導致波型被截，因為被截掉的部分還不到 5V，所以我一開始沒往電壓上限去想，但後來除錯出來還真的是因為倍率調太大。

Arduino 的部分也因為必須使用平常不常用的腳位而出現了一些問題，還有字串處理也不能用平常 C/C++的一些函式，必須自己手刻。至於中斷程式裡面改變模式要 Delay 也是想很久之後靈光乍現，之前有聽過別人遇到類似的問題，雖然是之前微算機實驗遇到的，但幸好有想起來，有些問題不是親身經歷印象就沒那麼深刻，現在自己遇到了應該之後想忘也忘不了。

因為是遠距做實驗，其實有遇到一些問題無法在課堂中當場處理有點麻煩，還要學習 AD2 怎麼使用，熟悉器材也花了點時間。現在回想起來，這也是最後一個電子實驗了，感覺時間過得很快，有點不真實。做電子實驗以及寫結報的過程中，常會發現有許多課堂上沒弄清楚的觀念，或是遇到沒碰過的問題，自己努力把問題解決掉會很有成就感。雖然遇到問題的當下會頭很大，但解決之後學到的事情跟經驗是無法替代的。這應該就是所謂的不經一事，不長一智吧。

E. Reference (10%)

- [1] [維基百科] 心電圖
<https://zh.wikipedia.org/zh-tw/%E5%BF%83%E7%94%B5%E5%9B%BE>
- [2] 心電圖原理與病症資料
<http://rportal.lib.ntnu.edu.tw:8080/server/api/core/bitstreams/1cf4b9ac-3ff2-4794-a06f-c58b6092344c/content>
- [3] [Hello 醫師] 維持心臟健康：心臟保健與功能介紹
<https://helloyishi.com.tw/heart-health/things-you-need-to-know-about-heart-health/>
- [4] ECG vs PPG for Heart Rate Monitoring: Which is Best?
<http://neurosky.com/2015/01/ecg-vs-ppg-for-heart-rate-monitoring-which-is-best/>
- [5] [RICHTEK] ECG/PPG 量測解決方案
https://www.richtek.com/Design%20Support/Technical%20Document/AN057?sc_lang=zh-TW
- [6] Arduino : stk500_getsync(): not in sync
<http://yehnan.blogspot.com/2014/10/arduinostk500getsync-not-in-sync.html>
- [7] [維基百科] 儀表放大器
<https://zh.m.wikipedia.org/zh-tw/%E5%84%80%E8%A1%A8%E6%94%BE%E5%A4%A7%E5%99%A8>

Arduino 紀錄

1. 腳位

Digital

0	1	2	3	4	5	6	7	8	9	10	11	12
[七段顯示器]	開關	[七段顯示器] (共腳位)									[測距器]	
VCC (L)	VCC (R)	輸入	E	D	C	P	G	F	A	B	Trig	Echo

Analog

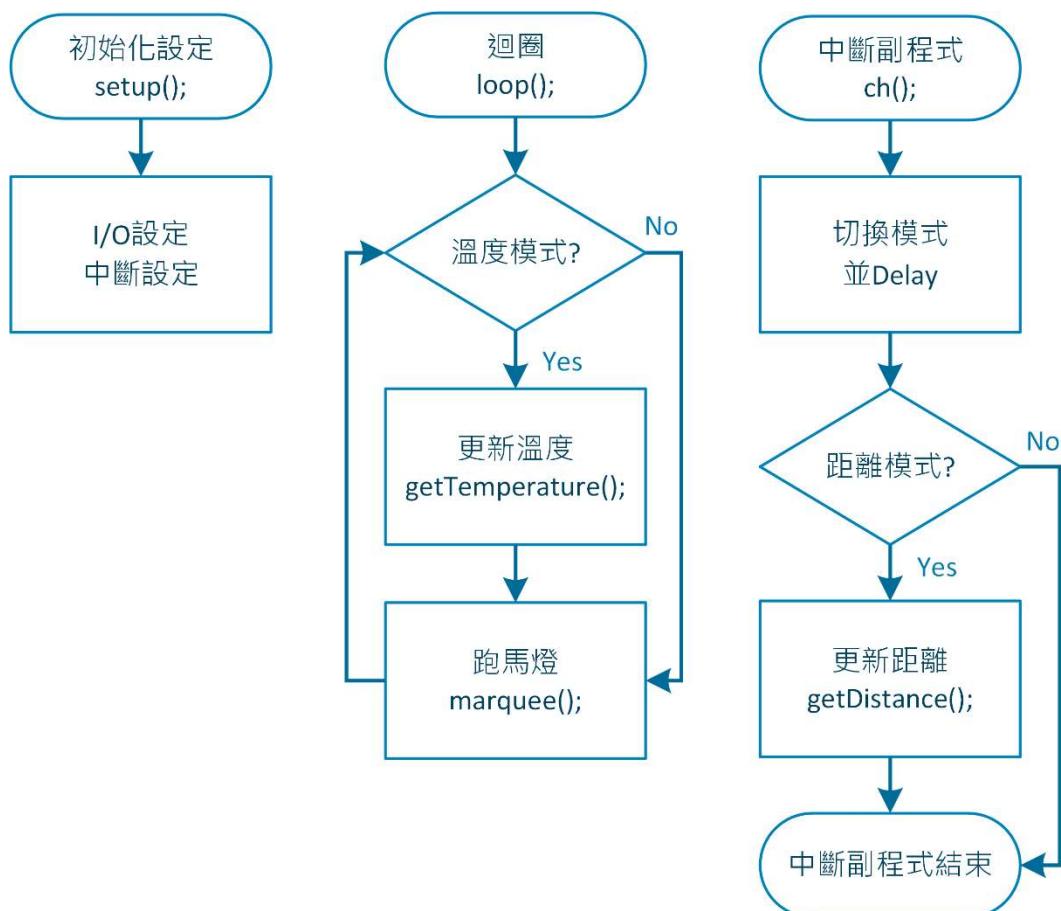
A0: [溫度感測器] S

Power

VCC: 溫度、距離感測器 VCC

GND: 溫度、距離感測器 GND

2. 程式架構



3. 程式碼

```
int      button = 2; //change mode
bool     mode = 0;    //0:temperature 1:distance
char    content[10];//Display string
int      len;        //strlen
int      dt = 5;     //duration per frame (ms)
int      repeat = 50;//5ms x 2 x 50 = 500ms/frame (marque)
//-----[pin][func]-----
//7-seg:display
int      pinE=3, pinD=4, pinC=5, pinP=6, pinG=7, pinF=8, pinA=9, pinB=10;
int      digits[2]={0,1};
void    showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P);
void    showChar(char c);
void    showFrame(char a,char b); //left char, right char
void    marquee(char str[]);
//HC-SR04:distance
int      Trig = 11;
int      Echo = 12;
double   distance;
void    getDistance();
//LM35:temperature
//      pinS = A0
int      MAX = 66;
int      i;
int      x,a;
double   temperature;
char    fstr[10];
void    getTemperature();
//-----[interrupt]-----
bool    en = 0;
void    itr();
//-----
void setup()
{
    pinMode(button, INPUT);
    //7-seg
    pinMode(digits[0],OUTPUT); pinMode(digits[1], OUTPUT);
    pinMode(pinA, OUTPUT); pinMode(pinB, OUTPUT); pinMode(pinC, OUTPUT);
    pinMode(pinD, OUTPUT); pinMode(pinE, OUTPUT); pinMode(pinF, OUTPUT);
    pinMode(pinG, OUTPUT); pinMode(pinP, OUTPUT);
    //HC-SR04
```

```
pinMode(Trig, OUTPUT);
pinMode(Echo, INPUT);
//interrupt
attachInterrupt(digitalPinToInterruption(button), ch, RISING);
}

void loop()
{
    if(!mode){getTemperature();} //Temperature mode: update data every cycle
    marquee(content);
}

void ch()
{
    if(en)
    {
        en = 0;
        mode=!mode;
        delay(50); //preventing from mode hasn't inverted
        if(mode){getDistance();} //Distance mode: update data once
    }
}

void getDistance()
{
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    distance = pulseIn(Echo, HIGH);
    distance *= 0.017;
    x=(int)distance;
    for(int i=2;i>=0;i--)
    {a=x%10;fstr[i]='0'+a;x/=10;}
    fstr[3]=0;
    sprintf(content, " %scm ",fstr);
}

void getTemperature()
{
    x=0;
    for(i=0;i<MAX;i++)
}
```

```
{  
    x += analogRead(A0);  
    delay(5);  
}  
x/=MAX;  
temperature=((float)x)*500.0/1024.0;  
dtostrf(temperature,3,1,fstr);  
sprintf(content," %soC ",fstr);  
}  
  
void showSevenSeg(byte A, byte B, byte C, byte D, byte E, byte F, byte G, byte P)  
{  
    digitalWrite(pinA, A); digitalWrite(pinB, B); digitalWrite(pinC, C);  
    digitalWrite(pinD, D); digitalWrite(pinE, E); digitalWrite(pinF, F);  
    digitalWrite(pinG, G); digitalWrite(pinP, P);  
}  
  
void showChar(char c)  
{  
    switch(c)  
    {  
        case '0':  
            showSevenSeg(0,0,0,0,0,0,1,1);  
            break;  
        case '1':  
            showSevenSeg(1,0,0,1,1,1,1,1);  
            break;  
        case '2':  
            showSevenSeg(0,0,1,0,0,1,0,1);  
            break;  
        case '3':  
            showSevenSeg(0,0,0,0,1,1,0,1);  
            break;  
        case '4':  
            showSevenSeg(1,0,0,1,1,0,0,1);  
            break;  
        case '5':  
            showSevenSeg(0,1,0,0,1,0,0,1);  
            break;  
        case '6':  
            showSevenSeg(0,1,0,0,0,0,0,1);  
    }  
}
```

```
        break;
    case '7':
        showSevenSeg(0,0,0,1,1,1,1,1);
        break;
    case '8':
        showSevenSeg(0,0,0,0,0,0,0,1);
        break;
    case '9':
        showSevenSeg(0,0,0,0,1,0,0,1);
        break;
    case ' '://space
        showSevenSeg(1,1,1,1,1,1,1,1);
        break;
    case 'c'://XXXcm
        showSevenSeg(0,1,1,0,0,0,1,1);
        break;
    case 'm'://XXXcm
        showSevenSeg(1,1,0,1,0,1,0,1);
        break;
    case '.'://XX.X°C
        showSevenSeg(1,1,1,1,1,1,1,0);
        break;
    case 'o'://XX.X°C
        showSevenSeg(0,0,1,1,1,0,0,1);
        break;
    case 'C'://XX.X°C
        showSevenSeg(0,1,1,0,0,0,1,1);
        break;
    default://off
        showSevenSeg(1,1,1,1,1,1,1,1);
        break;
}
```

```
void showFrame(char a,char b)
{
    int count = repeat;
    while(count--)
    {
        //digit 0
        digitalWrite(digits[1],LOW);
```

```
showChar(a);
digitalWrite(digits[0],HIGH);
delay(dt);
//digit 1
digitalWrite(digits[0],LOW);
showChar(b);
digitalWrite(digits[1],HIGH);
delay(dt);
}

}

void marquee(char str[])
{
for(int i=1;i<strlen(str);i++)
{showFrame(str[i-1],str[i]);}
if(!en){en=1;}
}
```