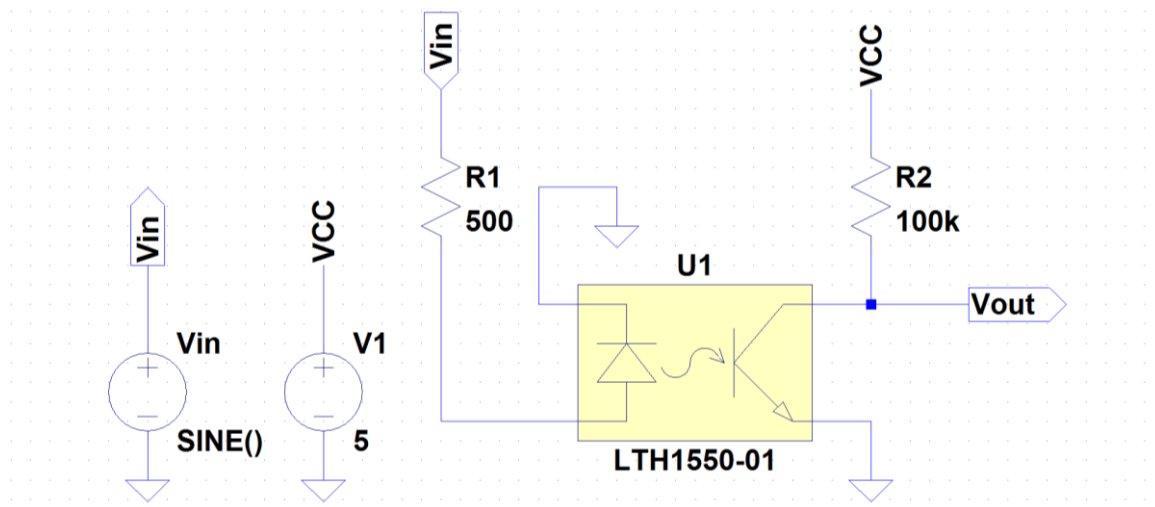


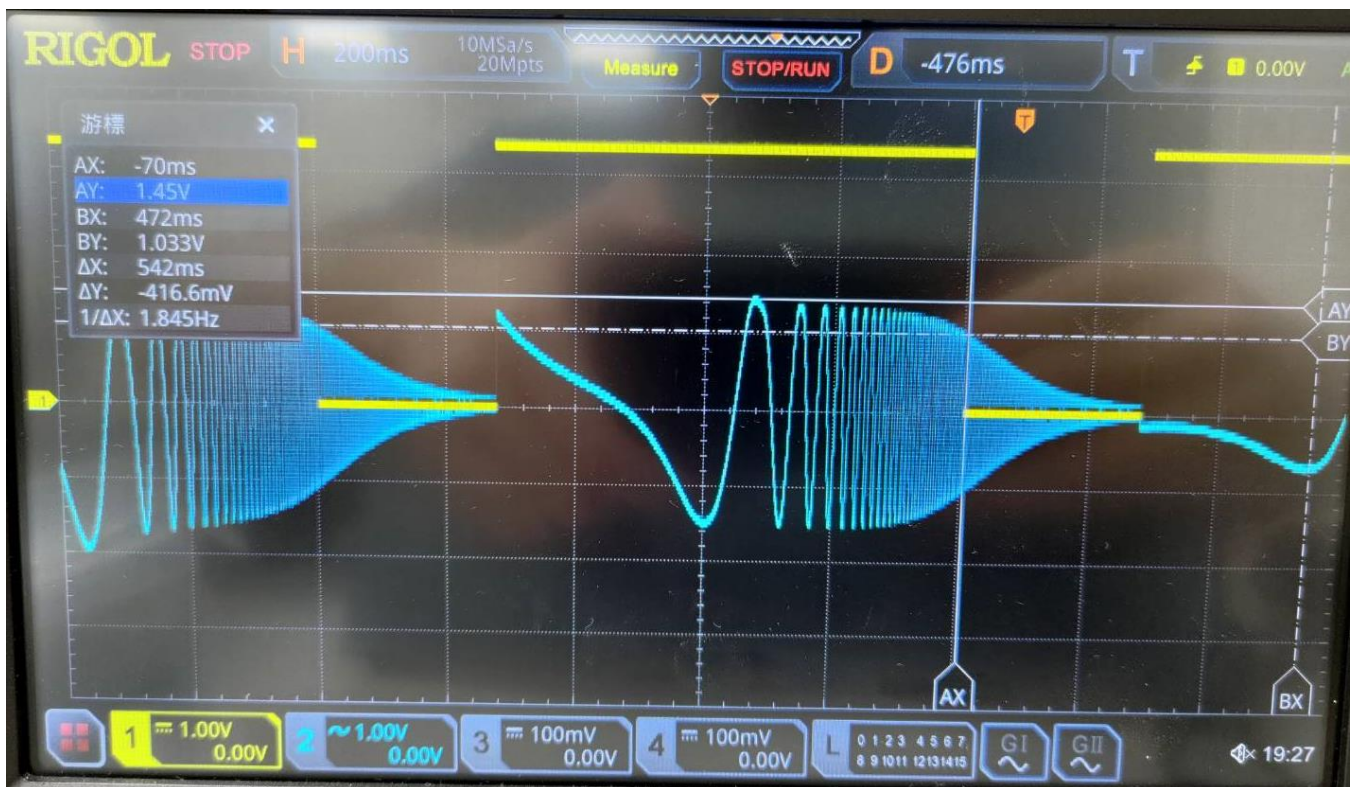
REPORT

Experiment 1: IR Driver and Sensor



2. AC SWEEP and Bias

(AC couple in channel 2)對 channel 2 做交流耦合才能夠比較輕易的量測到 3dB 頻率為多少。



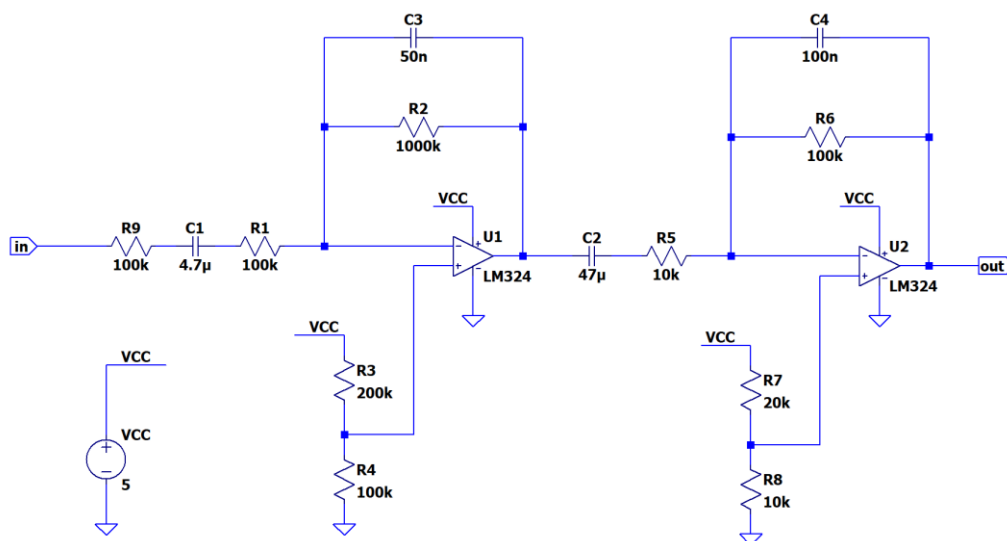
(DC couple)

但是倘若要量測 V_{out} 的平均電壓可不能忽略了直流的成分，因此要切回直流耦合。



$f_{3dB,H}$ (Hz)	V_{out} average voltage (V)
425Hz	3.23

Experiment 2: Filter Stage



Vin : 50mVpp or appropriate value that Vo is not distorted

OSC : DC coupling

2. DC Bias

U1,V+	U1,Vout	U2,V+	U2,Vout
(V)	(V)	(V)	(V)
1.63	1.66	1.63	1.60

3. AC SWEEP waveform

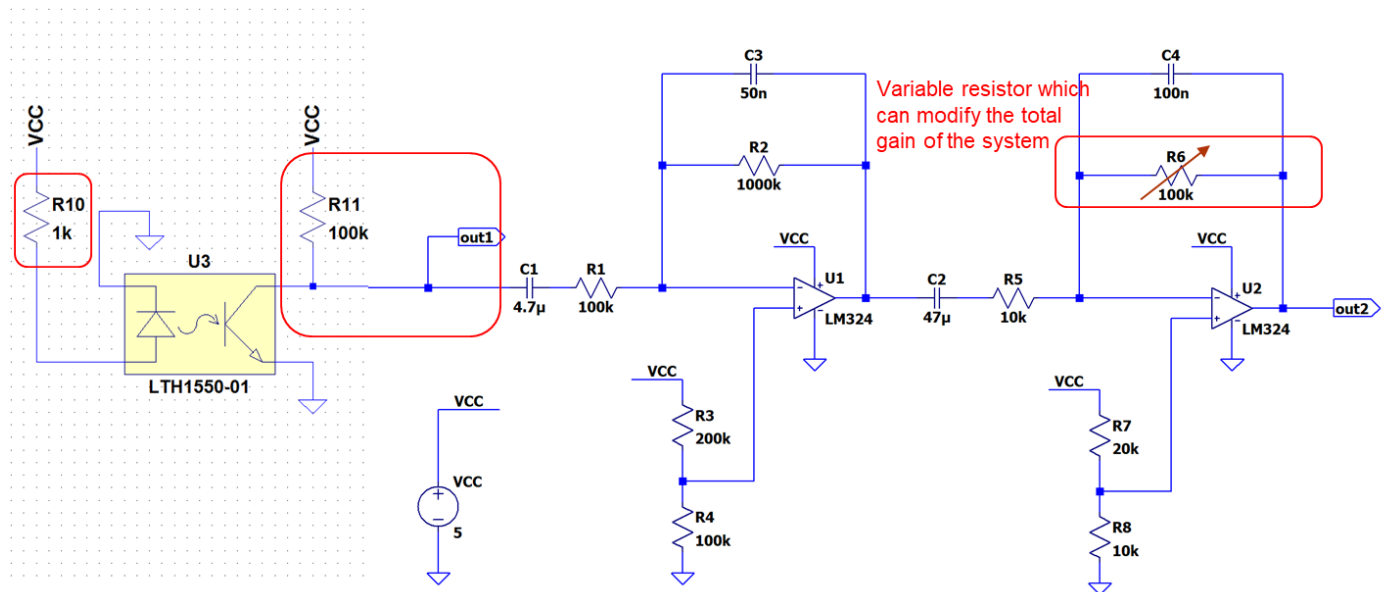
$f_{3dB,H}$ (Hz) _____ 4.5 _____



在實驗二的電路中，有一個值得注意的小陷阱:電路圖中的 C1 電容的極性要注意不要接錯。

因為 C1 左邊看到的是 FG 所產生的 50mVpp 的 sine wave，而右邊看到的則是大約 2~3V 的電壓，因此電容的正確接法應該是左邊為負腳而右邊為正腳。我一開始就接顛倒了，因此我 U1,Vout 量測到的電壓為大約 2.2V，經過助教的提醒我便發現原來我是電容正負接錯所致，謝謝助教！

Experiment 3: Heart Rate Monitor



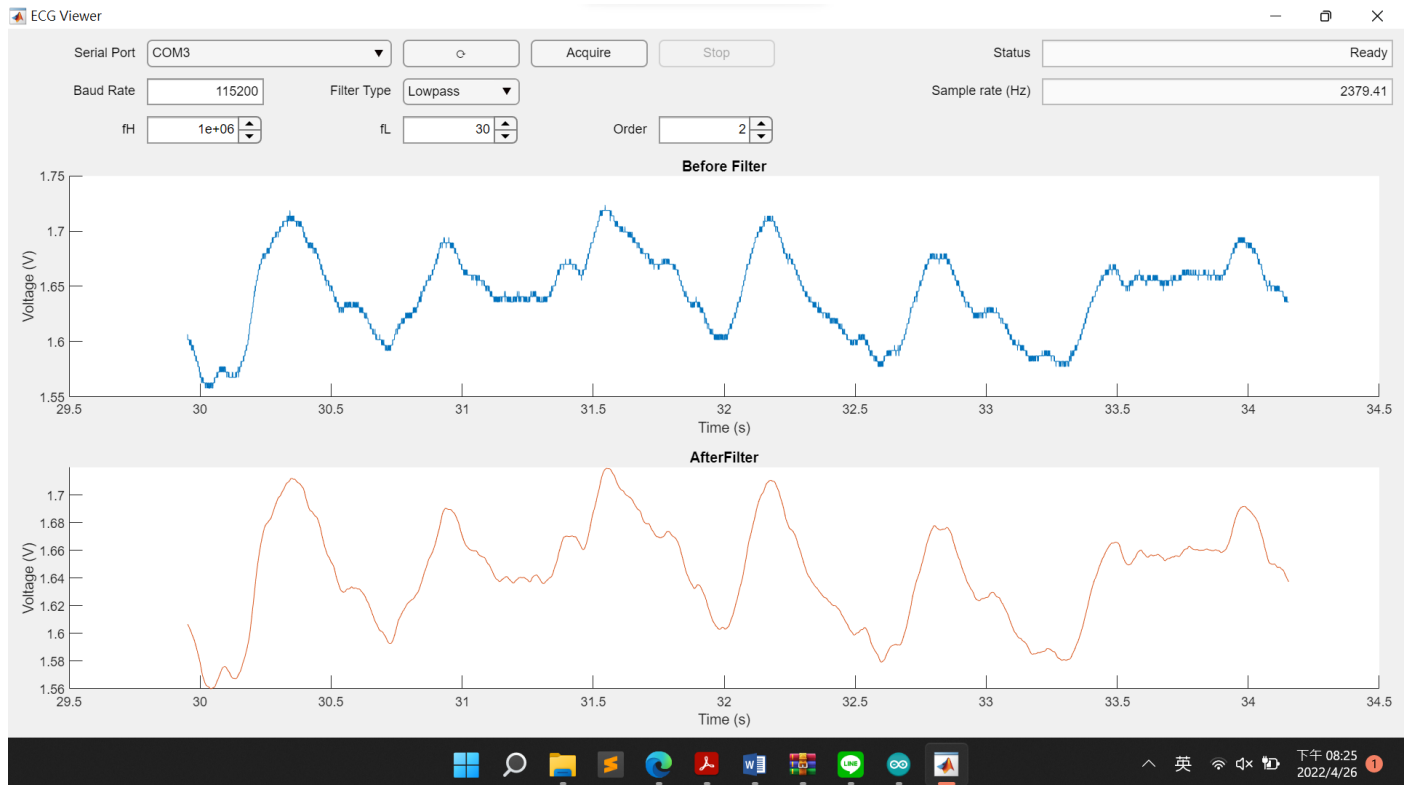
2. Vout1 and Vout2 waveform (in time domain)



Channel 1 為透過 LTH1550 將我的手指脈搏轉換成電訊號，而這個訊號是非常小的，可以透過左下角的刻度可以看出他的振幅是非常的小，而後經過後面兩個主動帶通濾波器的放大及訊號處理後，可以得到較大的 Channel 2 的訊號，time scale 為 500ms，因此可以推測出我當時做實驗的心跳大約是 120 beats per minute，可以說是非常的緊張，深怕實驗做不出來。

3. Record you real time heart rate = 120 BPM (beat per minute)

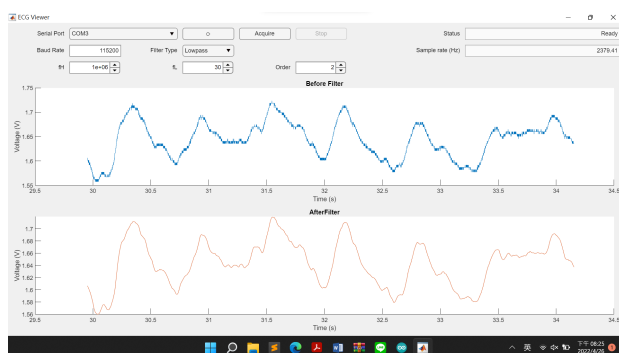
Take a screenshot after reading the stable heart rate.



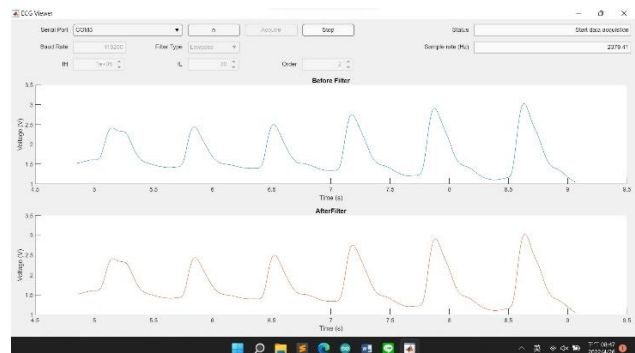
最後，將通過兩次放大電路的訊號透過助教寫好的 Matlab GUI 進行訊號處理並且視覺化。將 channel2 的訊號傳入 Arduino，而後通過助教所撰寫的訊號處理的過程，將我們的訊號變得更加平滑，因為是選用 low pass filter，因此可以看到在 Before Filter 的上圖中有一些高頻的震動項，到下圖的 After Filter 後大致上都被濾除。目測這個執行檔所做的工作應該有兩個部分。

1. 將原始訊號與一個低通的 System 進行 convolution，以此將高頻部分濾掉。
2. 針對經過 Low Pass Filter 之後的訊號使用 moving average，將其變得更為平滑，易讀。

另外，在國小的健康檢查中，我便被診斷出有心臟病。一開始的檢查只是醫師用聽筒聽聽我的心跳，之後他便眉頭深鎖，請我躺好，使用超音波更進一步的檢查我的心臟，而後便因為心律不整及心雜音等地症狀檢測出了我心臟出了一點小毛病，從上圖可以發現我的心跳與其他同學的心跳相比起來較沒有規律(可以看下圖比較好比較)。



(這是我的心臟。)

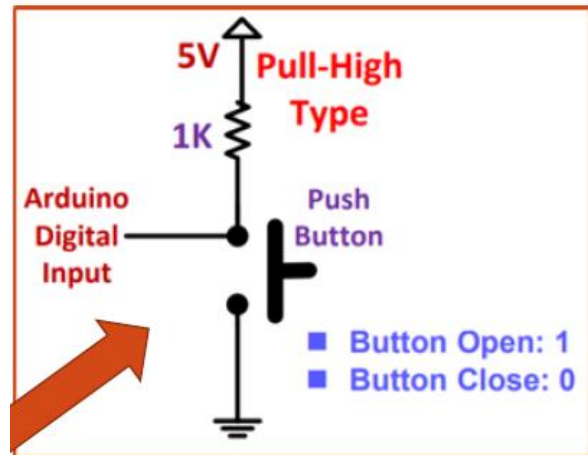
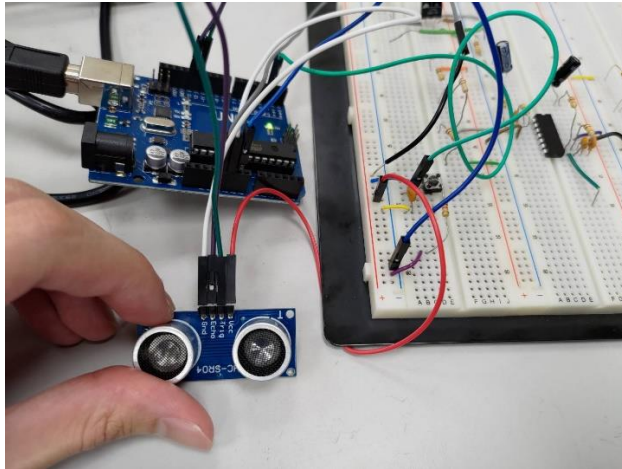


(這是我的同桌 17 號林瑾茹的心臟。)

都是用我的電腦測因為這個 ECG_viewer 沒有開發出 for MAC 的軟體，已徵求本人同意。

Experiment 4: Ultrasonic Sensor

The circuit diagram of your design: (label every port clearly)



因為之前上數位電路實驗寫 FPGA 時有提到 Button 會有 Bounce 的問題，因為不想透過 coding 來解決要判斷 posedge 跟 negedge 好難，所以我在 inter_pin 那邊加了一個 503 的電容以避免因為 button bouncing 的問題。

The sketch of your design: (copy from the Arduino IDE window and paste here)

```

1  int inter_pin = 2;
2  int trigger_pin = 12;
3  int echo_pin = 13;
4  long duration, cm;
5
6  void setup() {
7      pinMode(echo_pin, INPUT);
8      pinMode(trigger_pin, OUTPUT);
9      attachInterrupt(digitalPinToInterrupt(inter_pin), ultrasonic, RISING);
10     Serial.begin(9600);
11 }
12
13 void loop(){
14 }
15
16 void ultrasonic() {
17     pinMode(trigger_pin, OUTPUT);
18     digitalWrite(trigger_pin, LOW);
19     delayMicroseconds(2);
20     digitalWrite(trigger_pin, HIGH);
21     delayMicroseconds(10);
22     digitalWrite(trigger_pin, LOW);
23     pinMode(echo_pin, INPUT);
24     duration = pulseIn(echo_pin, HIGH);
25     cm = duration / 29 / 2;
26     Serial.print("Distance : ");
27     Serial.print(cm);
28     Serial.println(" cm");
29     delay(500);
30 }

```

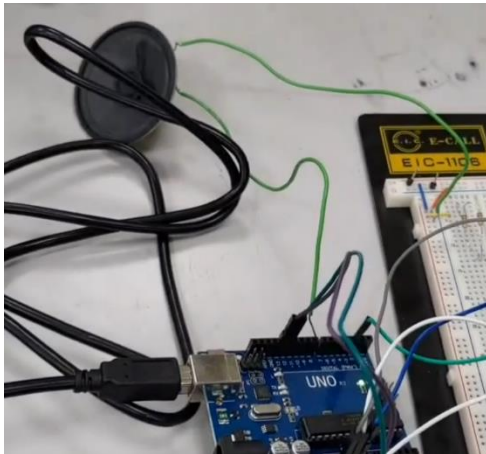
因為是使用 Push Button 來讓這個 Ultrasonic Sensor 來測距，根據 Pull-High type 的接法當按鈕沒有被按時會是高電位，而當按鈕被按下去之後會變成低電位，因此我們選擇要 Interrupt 的時機應該是從高電位轉到低電位的時候，或是被放開的過程，低電位變成高電位。若是單單以 HIGH 或是 LOW 來判斷會因為按鈕的 bouncing 而觸發測距多次，因此我選用 RISING edge 的時候當成 Sensor 會去測量的時機。如此一來便可以成功透過 Arduino 完成測量距離的功能。

The screen capture of the serial monitor: (show the distance value on the window)

```
Distance : 2 cm
Distance : 11 cm
Distance : 12 cm
Distance : 21 cm
Distance : 28 cm
Distance : 31 cm
Distance : 37 cm
Distance : 46 cm
```

Experiment 5: Melody Generator

The circuit diagram of your design: (label every port clearly)



單純的將喇叭的正極接到 Arduino 的 pin8，將負腳接到 GND。

The sketch of your design: (copy from the Arduino IDE window and paste here)

```
1  #define NOTE_CS4 277
2  #define NOTE_D4 294
3  #define NOTE_E4 330
4  #define NOTE_FS4 370
5  #define NOTE_GS4 415
6  #define NOTE_A4 440
7  #define NOTE_B4 494
8  #define NOTE_CS5 554
9  #define NOTE_D5 587
10 #define NOTE_E5 659
11
12 int tempo = 160;
13
14 int melody[] = {
15     NOTE_E5, NOTE_D5, NOTE_FS4, NOTE_GS4,
16     NOTE_CS5, NOTE_B4, NOTE_D4, NOTE_E4,
17     NOTE_B4, NOTE_A4, NOTE_CS4, NOTE_E4,
18     NOTE_A4
19 };
20
21 int durations[] = {
22     8, 8, 4, 4, 8, 8, 4, 4, 8, 4, 4, 2
23 };
24
25 int notes = sizeof(melody) / sizeof(int);
26 int wholenote = (60000 * 4) / tempo;
27 int noteDuration = 0;
28
29 void setup() {
30     for (int i = 0; i < notes; i++) {
31         noteDuration = (wholenote) / durations[i];
32         tone(8, melody[i], noteDuration*0.9);
33         delay(noteDuration);
34         noTone(8);
35     }
}
```

Your demo video (play the complete melody) link: <https://youtu.be/dCIXEbDhEME>