結報評分標準

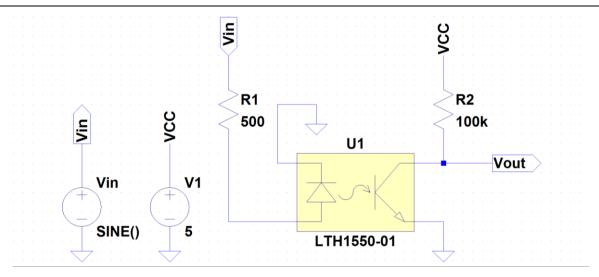
圖表	數據	發現問題	電路分析	心得+結論	Reference
15%	15%	10%	30%	20%	10%

請假後補交結報的規定

- 1. 請假需依規定提出假單申請,並安排時間補做實驗並將核準過的假單截圖貼於下方,助教才會進行 結報的批改。
- 2. 以請假日計算;需在一星期內完成補做驗,二個星期內補交結報(將結報交至 Delay 區)。逾時不進行結報批改。例如:3/1 請假,需在 3/8 前完成補做實驗,3/15 前補交結報。

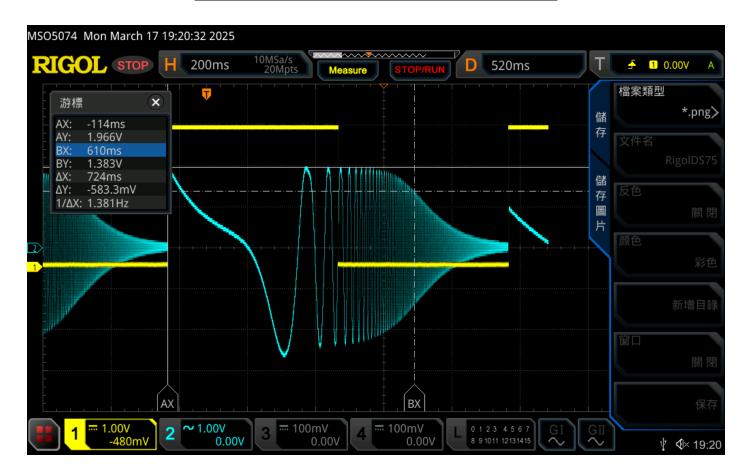
REPORT

Experiment 1: IR Driver and Sensor

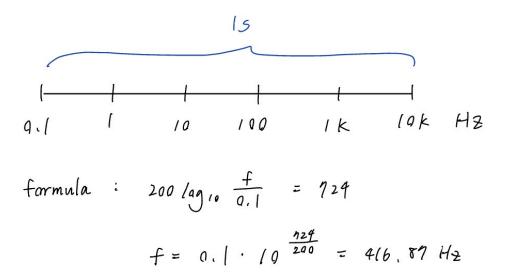


2. AC SWEEP and Bias

f _{3dB,H} (Hz)	Vout average voltage (V)	
416.87	2.67	



Heart Rate Monitor **3dB** 頻率計算:

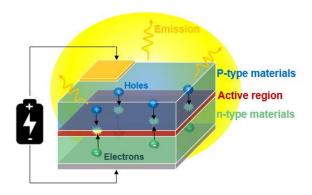


Explain the working principle of the LTH1550-01 sensor.

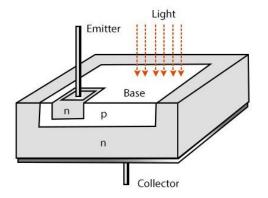


由右圖我們可以看到兩顆圓圓又光滑的東西,左邊那顆是 LED(emitter)、右邊那顆是光敏 BJT(receiver),透過左邊那顆 LED 發射出紅外線再透過光敏 BJT 感受光的波動轉而輸出電流,這時電 流在某些程度上會受到接收光的影響,因此我們就能利用這特性偵 測脈搏的變化。

LED & photodiode

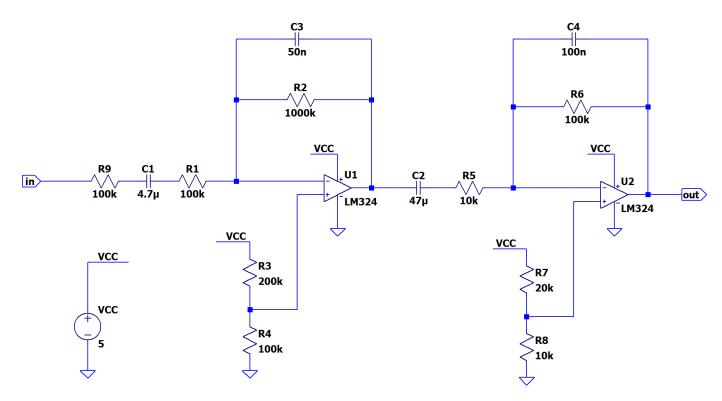


LED (發光二極管)是一種半導體元件,通電時能發出可 見光或紅外光。具有高效、壽命長、體積小等優點,廣泛 用於照明、顯示器和電子設備指示燈。



光電晶體管是一種利用光線控制電流的三極管。當光照射到基 極時,會產生光電流,放大後輸出。

Experiment 2: Filter Stage



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

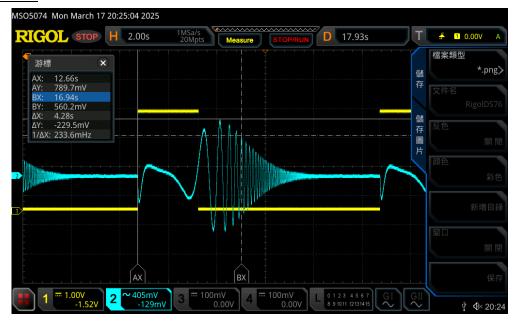
OSC: DC coupling

2. DC Bias

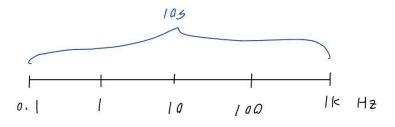
U1,V+	U1,Vout	U2,V+	U2,Vout
(V)	(V)	(V)	(V)
1.66	1.65	1.67	1.68

3. AC SWEEP waveform

 $f_{3dB,H}$ (Hz) 5.15



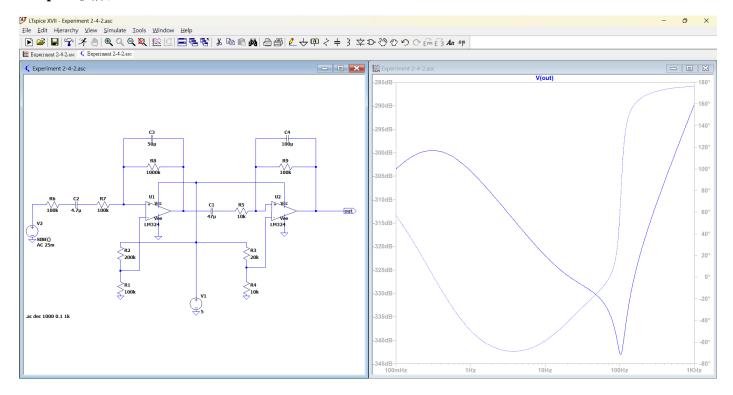
3dB 頻率計算:



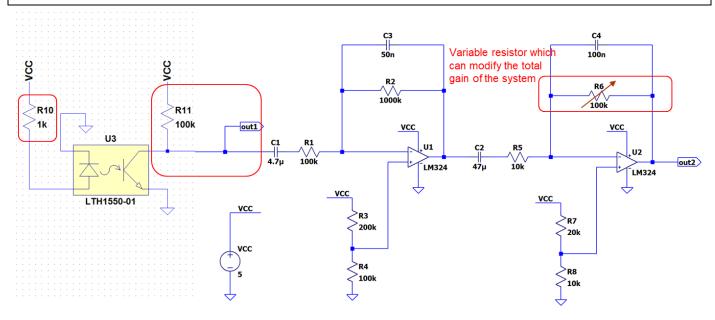
formula: 2.5 log10
$$\frac{f}{0.1} = 4.28$$

$$f = 0.1 \times 10^{\frac{4.28}{2.5}} = 5.15 \text{ Hz}$$

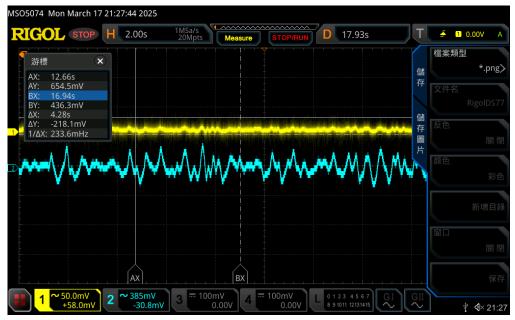
LTspice 模擬:



Experiment 3: Heart Rate Monitor

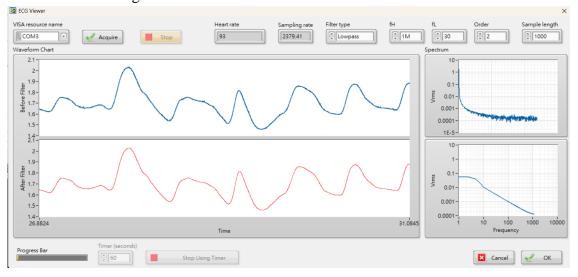


2. Vout1 and Vout2 waveform

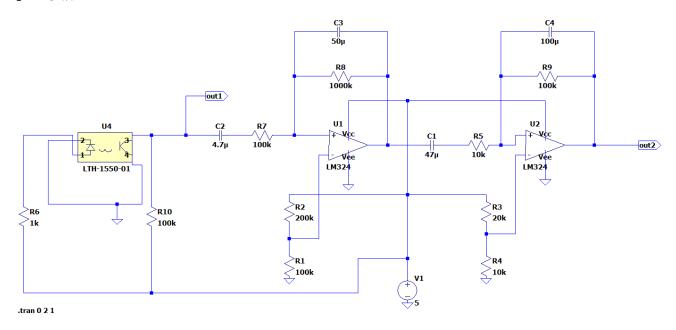


3. Record you real time heart rate = _____93 ____ BPM (beat per minute)

Take a screenshot after reading the stable heart rate.

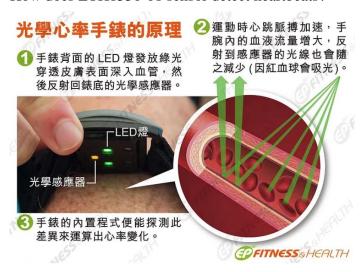


LTspice 模擬



LTspice 模擬的部分因為我在網路上沒看到LTH1550-01模型的關係,所以我靠Grok Ai 的幫忙之下基於LTH1550-01的 datasheet寫了一部份lib檔,但試錯好幾次error之後就放棄了,因為有bug一直沒辦法修好,上方的圖示我自己用線拉出來的,但是由於我對Carve不太熟所以有點歪請見諒。

How does LTH1550-01 sensor detect heartbeats?

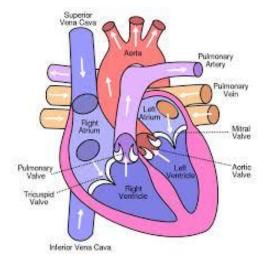


這時我們就要來討論下血液裡的成分,我們 在高中就有學到說血液的成分大致上有:水、紅血 球、

白血球、凝血因子、拜謝廢物等等的其他雜質, 然後其中紅血球會吸光,所以我們可以利用這個 特性去監測脈搏,原理如下:

心臟在收縮時會打出許多紅血球,這時候光線相對就會被吸收很多部分,這時候輸出的電流相對較小,這時候 Vout 讀到的電壓反之相對較高。

心臟在舒張時紅血球會相對變少許多,這時候光線相對沒被吸收,這時候電流就相對較大,不過 Vout 輸出的電壓反之越小。

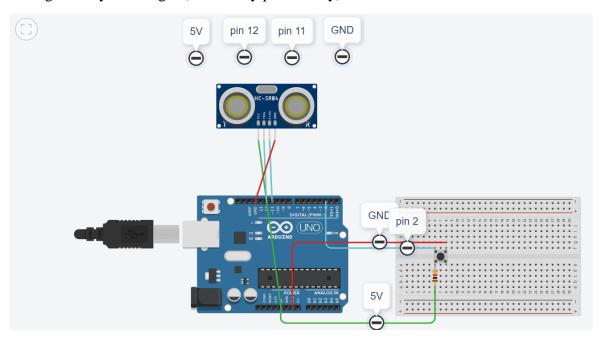


Why do we need adjustable resister?

我覺得主要原因是在 transfer function 上,雖然我沒有計算 band-pass filter 的 transfer function 但 我覺得調整可變電阻值能夠控制 transfer function 的值,這樣就可以間接改變 cut-off frequency,即可以自由地調整可視頻域大小,直到看到心跳的頻率為止。

Experiment 4: Ultrasonic Sensor

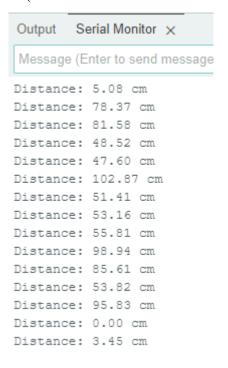
The circuit diagram of your design: (label every port clearly)



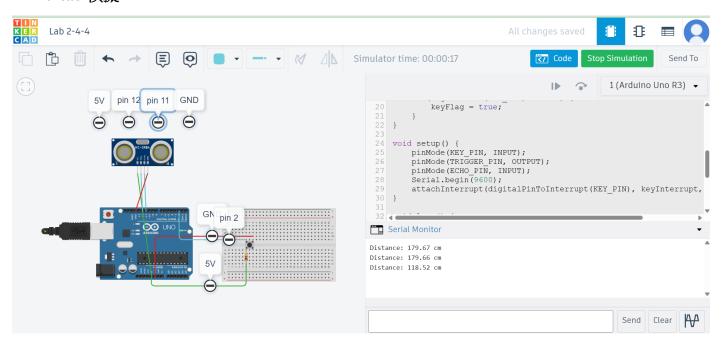
The sketch of your design: (copy from the Arduino IDE window and paste here)

```
#define KEY_PIN 2
     #define TRIGGER_PIN 12
 3
     #define ECHO_PIN 11
 4
 5
     bool keyFlag = false;
 6
     float getDistance(int trigPin, int echoPin) {
 8
         digitalWrite(trigPin, LOW);
9
         delayMicroseconds(2);
10
         digitalWrite(trigPin, HIGH);
         delayMicroseconds(10);
11
12
         digitalWrite(trigPin, LOW);
13
14
         long duration = pulseIn(echoPin, HIGH, 20000); // 20ms timeout
         return duration * 0.034 / 2; // Convert duration to distance in centimeters
15
16
17
18
     void keyInterrupt() {
19
         if (digitalRead(KEY_PIN) == LOW) {
20
             keyFlag = true;
21
22
23
24
     void setup() {
         pinMode(KEY PIN, INPUT);
25
         pinMode(TRIGGER_PIN, OUTPUT);
26
27
         pinMode(ECHO_PIN, INPUT);
28
         Serial.begin(9600);
29
         attachInterrupt(digitalPinToInterrupt(KEY_PIN), keyInterrupt, FALLING);
30
31
     void loop() {
32
33
         if (keyFlag) {
             delay(90); // Debounce
34
35
             if (digitalRead(KEY_PIN) == LOW) {
                 float distance = getDistance(TRIGGER_PIN, ECHO_PIN);
36
37
                 Serial.print("Distance: ");
38
                 Serial.print(distance);
39
                 Serial.println(" cm");
40
                 delay(1000);
41
                 keyFlag = false;
42
43
```

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



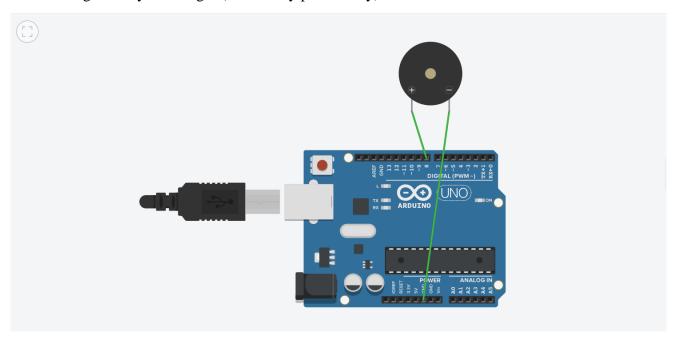
Tinkercad 模擬



在這部分遇到個有趣的問題,我原本在模擬 tinkercad 時 serial monitor 一直不跳訊息,這讓我非常困惑,檢查電路檢查了個 3 次都沒看出問題所在,所以我就重寫程式碼,但是還是沒發現問題,直到我不經意間去零件搜尋的地方找看看 button 是否還有其他的款式,無意間發現還有一款叫做 "pushbutton" 的按鍵,抱著好奇心再次做實驗,結果還真的被我做出來了。

Experiment 5: Melody Generator

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



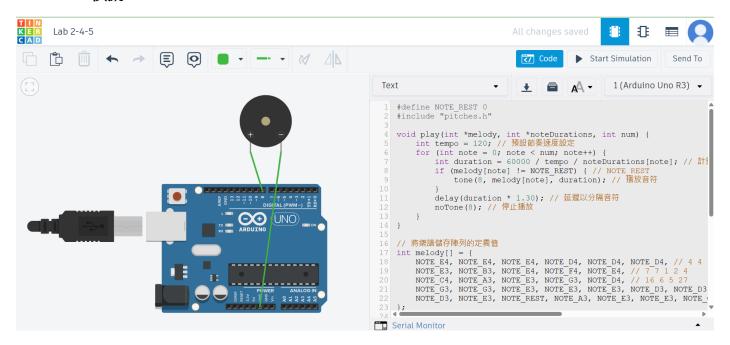
The sketch of your design: (copy from the Arduino IDE window and paste here)

```
toneMelody.ino pitches.h
   1
       #define NOTE_REST 0
   2
       #include "pitches.h"
       void play(int *melody, int *noteDurations, int num) {
   4
   5
          int tempo = 120; // 根據樂譜速度設置
   6
           for (int note = 0; note < num; note++) {</pre>
   7
               int duration = 60000 / tempo / noteDurations[note]; // 計算每個音符的持續時間
   8
               if (melody[note] != NOTE_REST) {
   9
                tone(8, melody[note], duration); // 播放音符
  10
  11
              delay(duration * 1.30); // 延遲以區分音符
  12
              noTone(8); // 停止播放
  13
  14
  15
       // 根據簡譜轉換的旋律
  16
  17
       int melody[] = {
  18
           NOTE_E4, NOTE_E4, NOTE_D4, NOTE_D4, NOTE_D4, // 4 4 4 3 3 3
  19
           NOTE_B3, NOTE_B3, NOTE_E4, NOTE_FS4, NOTE_E4, // 7 7 1 2 4
  20
           NOTE_C4, NOTE_A3, NOTE_A3, NOTE_G3, NOTE_D4, NOTE_D4, // 16 6 5 27
           NOTE_G3, NOTE_G3, NOTE_E3, NOTE_E3, NOTE_E3, NOTE_D3, NOTE_D3, NOTE_D3, // 55 3 3 3 2 2 2
  21
  22
           NOTE_D3, NOTE_E3, NOTE_REST, NOTE_A3, NOTE_E3, NOTE_E3, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_C4, NOTE_E3 // 2 i 3 0 6 31 31 15 15
  23
       };
  24
       // 音符持續時間
  25
       int noteDurations[] = {
          4, 4, 4, 4, 4, // 4 4 4 3 3 3
  27
          4, 4, 4, 4, 4, // 7 7 1 2 4
  28
          4, 4, 4, 4, 4, 4, // 16 6 5 27
  29
  30
          4, 4, 4, 4, 4, 4, // 55 3 3 3 2 2 2
  31
           4, 4, 4, 4, 4, 4, 4, 4, 4 // 2 i 3 0 6 31 31 15 15
  32
  33
  34
       void setup() {
       // 初始化設置(無需特別設置)
  35
  36
  37
  38
       void loop() {
  39
           play(melody, noteDurations, sizeof(melody) / sizeof(int)); // 播放旋律
  40
           delay(5000); // 播放完後等待 5 秒
  41
```

Your demo video (play the complete melody) link:

https://youtube.com/shorts/-zCzZec4wI8?si=DpUMG2_v1As5d4jP

Tinkercad 模擬



說明: 這首歌是 <APT.> 的某段旋律,因為我看不太懂音譜所以就上網找了最簡單簡譜時做

♣ 結論

實驗一主要是要讓我們理解 LTH1550-01 的工作原理以及實做,透過實際將手指放到裝置上面的方式觀測在 AC sweep 下的結果。

實驗二則是透過兩個 op amp 實做出一個 band-pass filter。

實驗三則是實驗二和實驗三的結合體,但也有做一些改變,像是加入了可變電阻調整 transfer function 等等的。

實驗四是帶我們認識超音波感測器,透過 Arduino 控制模組製作出最簡單的聲波雷達。

實驗五用到上學期的揚聲器,除此之外我們還用 Arduino 控制模組透過程式調控的方式輸出特定頻率的聲音,除此之我們還可以自己些頻率、時間上的 delay 實際寫出喜歡的歌、旋律。

♣ 心得

之後我們做完實驗三就回去休息了,不過因為有點睡不著,所以就先開始做實驗四和實驗五。不得不說,這兩個實驗真的很有趣!實驗四的監測雷達非常有意思,居然可以透過程式控制啟動機制和單位等參數。我還拿著雷達到處檢測距離,不過發現它有時候會出現讀不到值的情況,不知道這是什麼原因?這讓我感到非常好奇。至於實驗五,我們需要用 Arduino 製作一段旋律。基於個人喜好,我選擇了《APT.》作為旋律的主體,但沒想到聲音效果有點四不像,不知道是我找的樂譜有問題,還是零件放得太久了......

♣ 引用

- 心律電路檢測原理(LTH1550)
 - https://www.cnblogs.com/zjutlitao/p/5133489.html
- 智慧手錶的檢測原理(LTH1550 相似模組) https://hk.running.biji.co/index.php?q=news&act=info&id=2185
- LTH1550-01 data-sheet

https://www.mouser.com/datasheet/2/239/H1550-01new-1141667.pdf?srsltid=AfmBOopRnK 75RyLmh5RHS3oTQ08jBODpsLXGl7-ekxx-p2XwhyglBpvu

- LED
 - $\underline{https://dop.nycu.edu.tw/ch/field_ii.html?aID=13}$
- Photodiode

https://zh.wikipedia.org/zh-tw/%E5%85%89%E7%94%B5%E4%BA%8C%E6%9E%81 %E7%AE%A1