

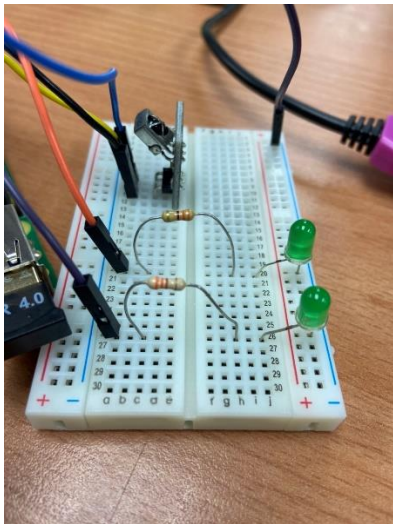
# Communication Networks Lab

## Topic Sensor Network-Lab4

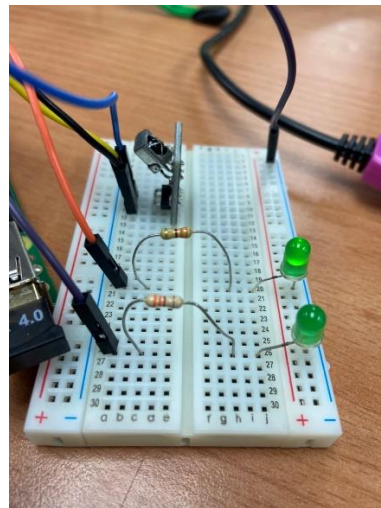
- Q1: 請錄製四個按鈕以控制 2 LED 的亮暗
  - LED\_1\_ON
  - LED\_2\_ON
  - LED\_1\_OFF
  - LED\_2\_OFF
- 接收遙控器發送的訊號，並根據相對應的按鈕使 LED 採取相應的行為(亮、暗)

Result:

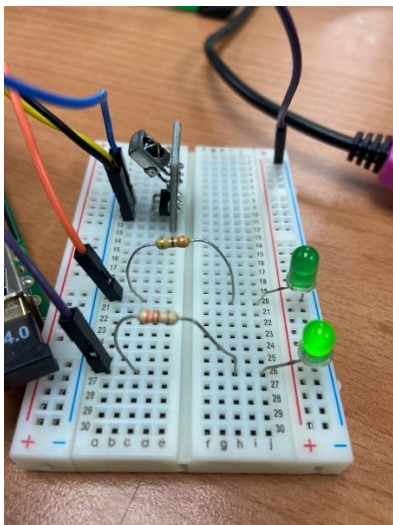
LED1 off LED2 off:



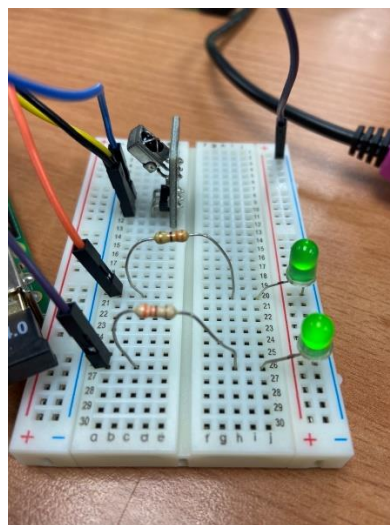
LED1 on LED2 off:



LED1 off LED2 on:



LED1 on LED2 on:



Using try except to ensure lirc.deinit() and gpio.cleanup() will be executed after the user press Ctrl + C causing keyboard interrupt errors.

main code:

```

18 try:
19     while True:
20         inf = lirc.nextcode()
21
22         if len(inf) == 0:
23             continue
24
25         if inf[0] == 'on_1':
26             GPIO.output(PIN_1, GPIO.HIGH)
27             print("LED_1_ON")
28
29         elif inf[0] == 'off_1':
30             print("LED_1_OFF")
31             GPIO.output(PIN_1, GPIO.LOW)
32
33         elif inf[0] == 'on_2':
34             print("LED_2_ON")
35             GPIO.output(PIN_2, GPIO.HIGH)
36
37         elif inf[0] == 'off_2':
38             print("LED_2_OFF")
39             GPIO.output(PIN_2, GPIO.LOW)
40
41 except KeyboardInterrupt:
42     print('stop')
43
44 finally:
45     GPIO.cleanup()
46     lirc.deinit()

```

.lircrc:

```

1 begin
2     prog = LED
3     button = KEY_1
4     config = on_1
5 end
6
7 begin
8     prog = LED
9     button = KEY_2
10    config = off_1
11 end
12
13 begin
14    prog = LED
15    button = KEY_3
16    config = on_2
17 end
18
19 begin
20    prog = LED
21    button = KEY_4
22    config = off_2
23 end

```

prog: program name

button: button name

config: message deliver to program when pressing buttons

It's really important to keep our GPIO pinMode in IN mode so that it will not be dangerous to our circuit or device. After installing the wiringPi package, we can type gpio readall to monitor all of the pinMode to better understand whether each pin is set to IN mode after the program is keyboard interrupted.

BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v		1	1	2		5v		
3	9	SDA.1	IN	1	3	4		5v		
4	7	SCL.1	IN	1	5	6		0v		
		GPIO. 7	IN	1	7	8	0	TxD	15	14
		0v			9	10	1	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v		
22	3	GPIO. 3	IN	0	15	16	0	GPIO. 4	4	23
		3.3v			17	18	0	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v		
9	13	MISO	IN	0	21	22	0	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	CE0	10	8
		0v			25	26	1	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	0	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v		
19	24	GPIO.24	IN	0	35	36	0	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	GPIO.28	28	20
		0v			39	40	0	GPIO.29	29	21

□ 1. 在使用紅外線技術傳輸時，可能受到哪些因素影響而無法正常運作？

在使用紅外線技術傳輸時，可能會因為發射端與接收端中間受到屏蔽，而使得接收端沒有成功收到訊號。此外，發射端與接收端的距離也需要很短，否則接收端也無法順利接收到訊號。在以空氣介質傳輸下，也可能受到大氣中的水氣影響，進而衰減訊號，使得接收端沒辦法接收到完整的訊號。

□ 2. 為避免受環境中相同波長的電磁波干擾，一般會在紅外線傳輸訊號時加上載波，試問市面上常見的載波頻率範圍為何？

調變:將特定頻率的訊號混入想要傳輸的訊號，以避免其他相同性質的訊號互相干擾。在台灣市面常見的家電載波的頻率大多都是約定俗成的 38kHz。

□ 3. 紅外線技術與藍芽技術有什麼差別？各自有什麼優缺點？(越詳細越高分)

紅外線：

優點:紅外線元件的價格相較射頻元件便宜許多，在結構上也比較簡單，因此紅外線傳輸的應用普及率很高。

缺點:傳輸資料時，發射端及接收端必須正對，並且中間不能有屏蔽。傳輸速率與距離成反比。在安全性上的控制略顯不足。同時最多只能兩個設備進行連線資料傳輸。

藍芽：

優點:傳輸時能夠穿越障礙物，不需要正對就能夠進行資料傳輸。能夠同時連接多個設備，共用頻寬。安全性的控制做得比紅外線裝置要好。與同樣為短距離傳輸的紅外線相比，傳輸距離較長一些。

缺點:因為藍芽無線電波發射在 2.4GHz 上，因此容易受到其他系統的無線電波干擾。傳輸互聯的設備都需要配備特定的 RF IC 晶片，價格稍微高一點。

## □ 4. 本次實驗心得，你學到了什麼東西？

這次的實驗學到了紅外線的傳輸技術，在上個學期的電子實驗中，也有透過紅外線傳輸技術實現無線的傳輸音樂訊號。當時發射端與接收端必須要正對的剛剛好才能夠順利地接收到訊號。這次做的是遙控器來傳輸訊號，就跟電視遙控一樣，非常新奇，比較接近日常生活會碰到的東西，也比較有共鳴。