

前 言

研究此專題的目的是為了瞭解紅外線控制的原理，探索光電科技，進而將紅外線技術，應用於實務上。

目前大部分機械都是用操作面板上的按鍵或者按鈕，做為控制介面，而對於移動性或遠距離的機械則無法以機身上的按鍵或者按鈕進行控制，此時，無線遙控的操作方式，將成為唯一選擇，例如：遙控飛機、電視機，音響器材等。

此專題的重點在於紅外線訊號的發射與接收，基本原理就像船艦在夜間以燈號傳送訊息一樣，利用燈亮時間的長短，代表 0 與 1 的訊號，對資料進行編碼/解碼，接收特定的編碼資料後，即進行對應動作。

在製作的過程中我們了解到紅外線的接收距離，接收器是否面對著遙控器也影響訊號強度，程式設計更是此專題得已完成的關鍵。

目 錄

前 言	1
第一章 紅外線控制的材料與介紹	
1-1 關於硬體與軟體	3
1-2 紅外線應用原理	4
1-3 紅外線遙控應用	5
1-4 BASIC Stamp 2	6
1-5 伺服機的校正	7
1-6 紅外線電路的設計	8
1-7 紅外線感測器分析	11
第二章 成果與程式	
2-1 成果	13
2-2 程式	14
參考文獻	19

第一章 紅外線控制的材料與介紹

1-1 關於硬體與軟體

我們參考書籍上的資料，剛開始學習microcontroller就跟接觸新品牌的電腦一樣。大部分的人一拿到電腦就會把它從盒子中取出來、接線，再來就是安裝並測試一些軟體，或著甚至開始自己寫程式。如果第一次使用BASIC Stamp 2 模組的話，你會進行如上面所說的一樣的事情。我們依照以下步驟指引來執行BASIC Stamp 2 (BS2)的程式。

- 執行程式編輯軟體(由原廠網站下載安裝)。
- 接通BASIC Stamp 2 模組電源。
- 連接BASIC Stamp 2 模組到電腦 RS232 Port 以供程式傳輸。
- 編寫PBASIC 程式並傳至BS2。
- 完成時拆除通訊線，BS2 可獨立運作。

1-2 紅外線應用原理

地球上佈滿不同的射線，例如：X 射線 (X-ray)，微波 (Microwaves)，紫外線 (Ultra-Violet) 等，他們的分別之處為大家都有不同的波長 (Wavelength) 及頻率 (Frequency)。而不同射線的波長範圍則組成電磁光譜 (Electromagnetic Spectrum)。

紅外線 (Infrared) 的波長介於可見光及微波之間。其波長比可見光長，比微波短。紅外線主要來自熱力。任何溫度比絕對零度 (Absolute Zero，大約是攝氏負 273 度) 都會發出紅外線。紅外線波長在 1 毫米到 770 奈米之間，在光譜上位於紅色光外側。具有很強熱效應，並易於被物體吸收，通常被作為熱源。透過雲霧能力比可見光強。在通訊、探測、醫療等方面有廣泛的用途。俗稱紅外光。

而在眾多光線中為何選擇紅外線，原因就是它適合在通訊方面輔助我們達成紅外線遙控。在別的光線中例如紫外線，它適合運用在醫療方面做放射治療，而不適合通訊方面。

1-3 紅外線遙控應用

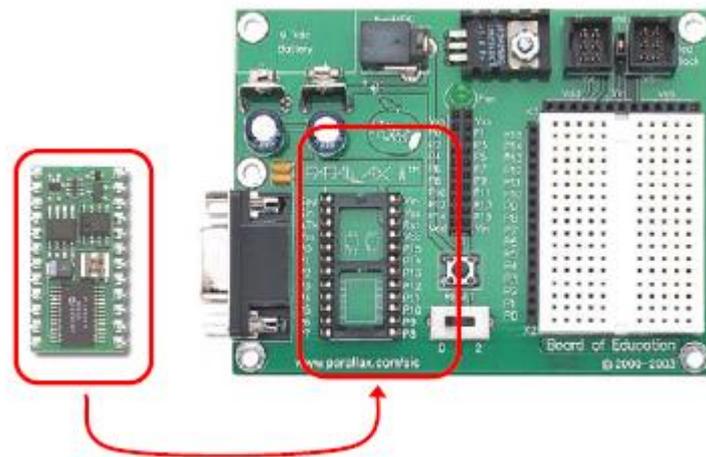
紅外線遙控是目前使用最廣泛的一種通信和遙控手段。由於紅外線遙控裝置具有體積小、功耗低、功能強、成本低等特點，因而，繼彩電之後，錄像機、音響設備、空調機以及玩具等其它小型裝置上也紛紛採用紅外線遙控。工業設備中，在高壓、輻射、有毒氣體、粉塵等環境下，採用紅外線遙控不僅完全可靠而且能有效地隔離電氣干擾。

通用紅外遙控系統由發射和接收兩大部分組成，應用編/解碼專用集成電路芯片來進行控制操作。發射部分包括鍵盤矩陣、編碼調製、LED紅外發送器；接收部分包括光、電轉換放大器、解調、解碼電路。

1-4 BASIC Stamp 2 Microcontroller

作為本專題的訊號擷取與解碼元件，利用類似Basic語言編寫程式，是一個容易上手的微控制器。

下圖為BASIC Stamp 2 模組和實驗電路板的圖片。BASIC Stamp模組像是一個很小的電腦，具有16支I/O pin。這個很小的電腦會插在電路板上。在程式編寫過程的除錯階段，或程式碼的下載時，BASIC Stamp 2透過RS232線與PC連接，方便它與PC間的資料傳遞，但當程式下載完畢，它也可以脫離PC而獨立作業。



1-5 伺服機的校正與控制

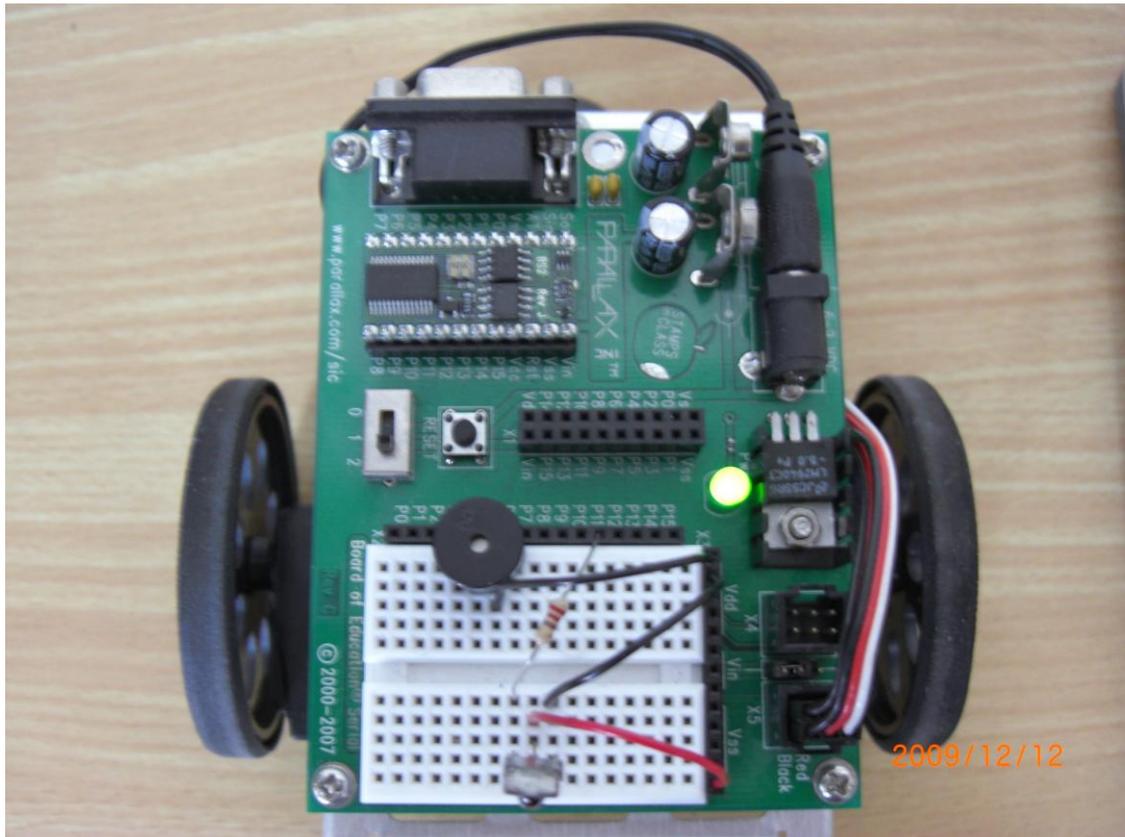
要執行一個程式，這個程式會傳送訊號到伺服機，要求伺服機保持靜止的狀態。

因為剛買來的伺服機並沒有在工廠事先調整過，因此剛買的伺服機收到這個訊號時並不會保持靜止。所以需要以螺絲起子調整伺服機校正螺釘，這樣一來當它收到這個訊號時就會保持停止。這叫做伺服機的校正。在經過調整後，測試一下伺服機功能是否正常。

伺服機運轉，是以50Hz脈波控制(脈波週期相當於20ms)。每一脈波週期中，on的時間為1.5ms則伺服機靜止，1.3ms則順時針全速運轉，1.7ms則逆時針全速運轉，介於這些值之間則可調整轉速。

1-6 紅外線與線路版的接合

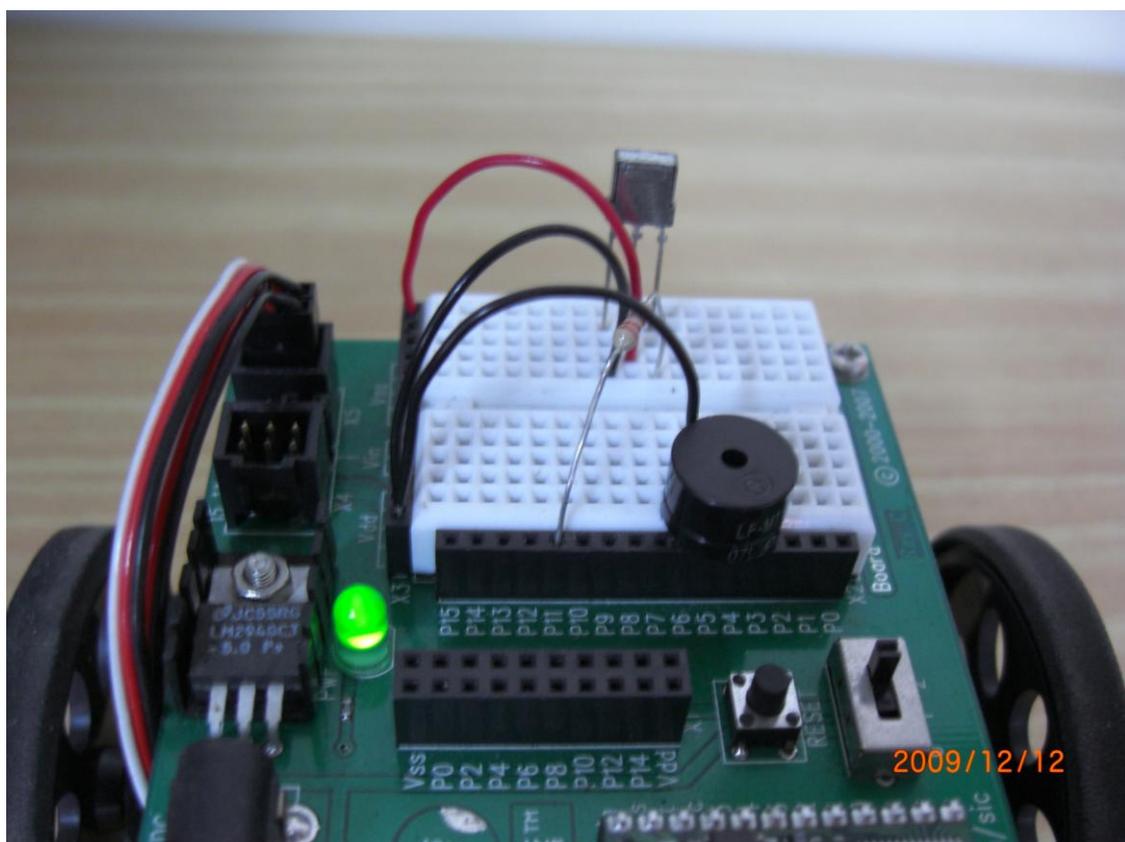
下圖是我們的電路板與紅外線感測器的連接線路，包括遙控器的控制方向按鈕。



(遙控車全圖)



(市售通用型遙控器)

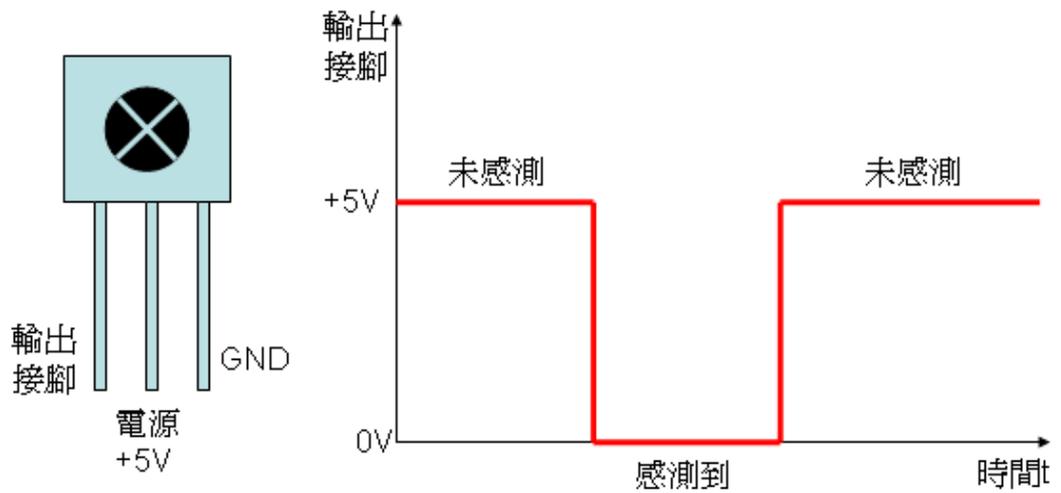


(紅外線感測器接線)

1-7 紅外線感測器分析

紅外線感測器工作時, 感測到紅外線則輸出接腳為0, 未感測到則輸出接腳為1(5V), 工作電壓5V. 將輸出接腳串聯電阻再接至Basic Commander I/O pin0~pin15任一接腳, 則可讀取此訊號.

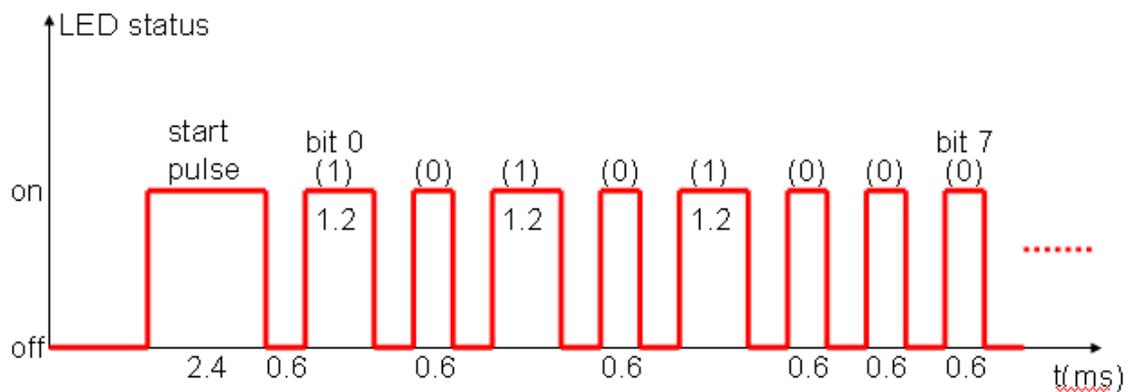
(下圖為分析圖)



(下圖為紅外線遙控器編碼表格, SONY電視格式)

按鍵	1	2	3	4	5	6	7	8	9	0	enter	CH+	CH-	VL+	VL-	mute	power
內碼	0	1	2	3	4	5	6	7	8	9	11	16	17	18	19	20	21

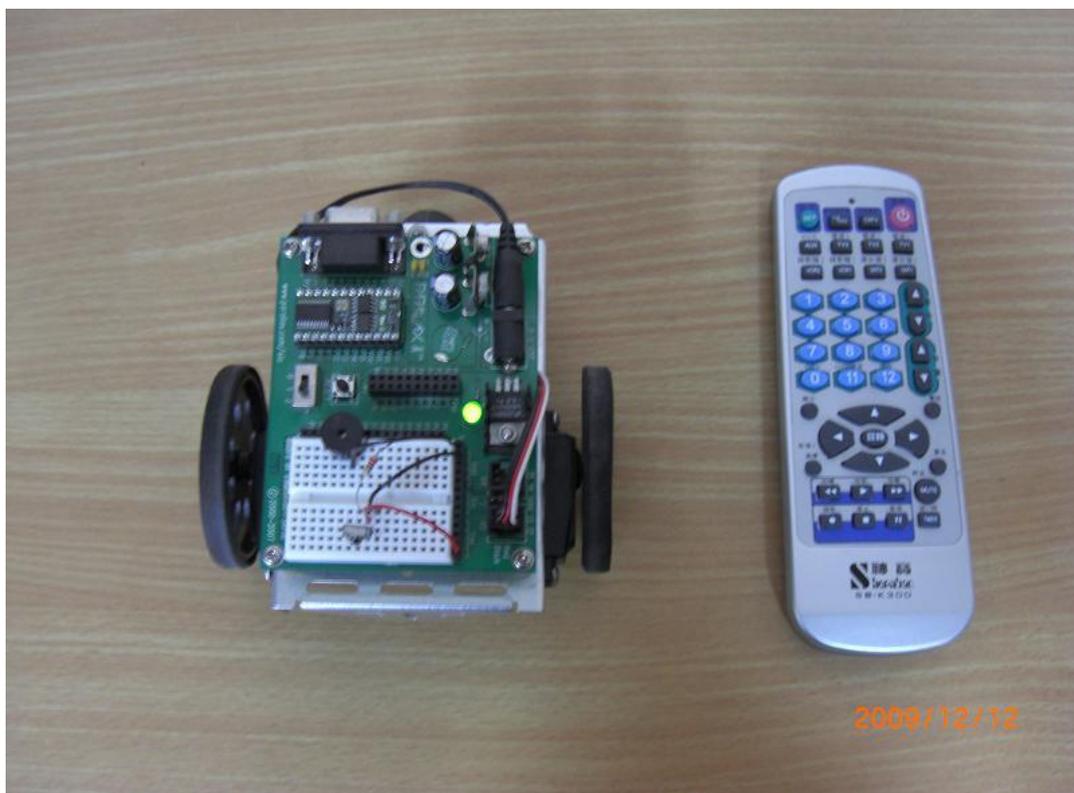
當按下遙控器按鍵，特定的內碼二進位值將藉由紅外線 LED閃滅的方式送出。LED on 0.6 ms 為“0”；LED on 1.2 ms 為“1”；間隔0.6ms LED off。以“power”鍵為例，內碼為21=00010101, LED on/off，動作如下圖：



以上的一組訊號，稱為一個“封包”(packet)，所需時間視0與1的數目而定，最長約24ms(不含間隔20~30ms)。

第二章 成果與程式

2-1 成果



組裝完成的車子跟操控的遙控器

2-2 程式

```
' IR Control Demo Car

' *****

' {$STAMP BS2}

' {$PBASIC 2.5}

ir_det PIN 9      ' Ir_RC sensor(left) input pin

spk PIN 4         ' speaker

ch_up CON 94      ' button constant

ch_dn CON 92      ' button constant

vol_up CON 116    ' button constant

vol_dn CON 117    ' button constant

power CON 21      ' button constant

freq CON 3000     ' beep

p_s CON 10        ' pause time

i VAR Byte

ir_p VAR Word     ' Ir pulse width detected

code VAR Byte     ' pressed button's code
```

```
p12 VAR Word    ' left servo pulse width  
p13 VAR Word    ' right servo pulse width
```

```
FREQOUT spk, 300, freq
```

```
DO
```

```
    GOSUB get_code
```

```
    GOSUB move
```

```
LOOP
```

```
get_code:
```

```
    code=0      ' reset button code
```

```
DO
```

```
    RCTIME ir_det, 1, ir_p
```

```
    LOOP UNTIL ir_p>1000    ' confirm pulse down edge of
```

```
resting time
```

```
RCTIME ir_det, 0, ir_p      ' confirm the 2.4 ms start pulse  
IF ir_p>1125 OR ir_p<675 THEN GOTO get_code      ' this  
instructuon last 0.3 ms into next pulse
```

```
RCTIME ir_det, 0, ir_p      ' don't use FOR i=0 TO 6...NEXT  
or code.LOWBIT(i) to get pulse (take too much time)
```

```
IF ir_p>300 THEN code.BIT0=1      ' this instructuon last  
0.3 ms into next pulse
```

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT1=1      ' this instructuon last  
0.3 ms into next pulse
```

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT2=1      ' this instructuon last  
0.3 ms into next pulse
```

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT3=1      ' this instructuon last  
0.3 ms into next pulse
```

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT4=1      ' this instructuon last
```

0.3 ms into next pulse

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT5=1 ' this instructuon last
```

0.3 ms into next pulse

```
RCTIME ir_det, 0, ir_p
```

```
IF ir_p>300 THEN code.BIT6=1
```

```
RETURN
```

move:

```
SELECT code
```

```
CASE ch_up', 94
```

```
p12=700 : p13=800
```

```
CASE ch_dn
```

```
p12=780 : p13=720
```

```
CASE vol_up
```

```
p12=780 : p13=780
```

```
CASE vol_dn
```

```
p12=720 : p13=720
```

```
CASE power
```

END

CASE ELSE 'undefined code

p12=750:p13=750

FREQOUT spk, 100, freq

ENDSELECT

PAUSE p_s

PULSOUT 12, p12

PULSOUT 13, p13

RETURN

參考文獻

1. BoeBot技術手冊
2. What is microcontroller技術手冊
3. IR remote control技術手冊