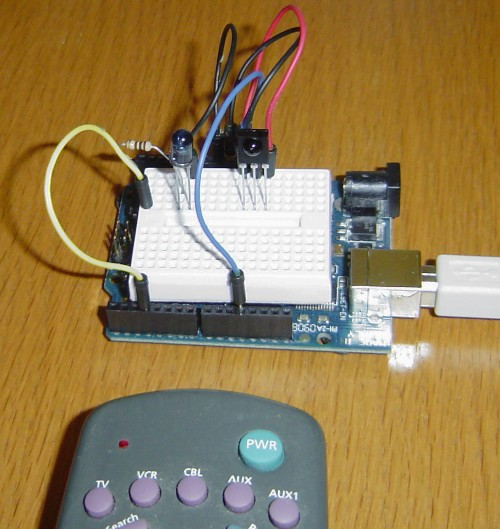
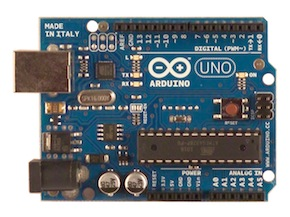
**[IRRemote 紅外線遙控教學](http://coopermaa2nd.blogspot.co.uk/2011/03/irremote.html)**

**IRRemote 函式庫簡介**

你想用 Arduino 做紅外線遙控嗎? 你想用 Arduino 控制你的電視、音響、冷氣機、DVD、MP3 Player、玩具遙控車或其它設備嗎? 用 Ken Shirriff 寫的 IRRemote 這個函式庫，會讓你的 Arduino 既可發射又可以接收紅外線訊號，它支援許多種紅外線協定，包括 NEC, Sony SIRC, Philips RC5, Philips RC6 和原始資料 (raw) 等紅外線協定。

這一系列的教學，將教你使用 IRRemote 函式庫來接收與發射紅外線訊號。 [](http://lh3.ggpht.com/_Ap9EEd9K-fg/TZL_rDzwCdI/AAAAAAAAEEg/JnPs1HoPJLE/s1600-h/image5.png)  
▲ 用 Arduino 與 IRRemote 函式庫的紅外線遙控電路。(圖片來源: [Ken Shirriff’s blog](http://www.arcfn.com/))

**你需要準備什麼器材?**

***1)*** 首先，你需要一張 Arduino 板子，一條 USB 傳輸線，以及 [Arduino IDE](http://arduino.cc/en/Main/Software) 軟體開發環境:   
  
▲ Arduino UNO 與 USB 傳輸線

  
▲ Arduino IDE

如果你沒有接觸過 Arduino，建議先看「[Arduino 筆記 – 認識 Arduino](http://coopermaa2nd.blogspot.com/2010/12/arduino-arduino.html)」和「[Arduino 筆記 - 準備開發環境](http://coopermaa2nd.blogspot.com/2010/12/arduino.html)」，從這兩篇開始。

***2)*** 一顆 38 kHz 的紅外線接收器 (Infrared Receiver)，例如 [TSOP2438](http://www.bitsbox.co.uk/data/TSOP2438.pdf),[TSOP4438](http://www.vishay.com/docs/82090/tsop48xx.pdf), [PNA4602](http://www.ladyada.net/media/sensors/PNA4602.pdf), [FM-8038TM2-5DN](http://www.icsourse.com/uploads/soft/100727/FM-8038TM2-5DN.pdf) 等。   
[](http://lh6.ggpht.com/_Ap9EEd9K-fg/TZL_wDP_DLI/AAAAAAAAEEo/0TYBUl1L9lI/s1600-h/image4.png)    
▲ 紅外線接收器(Infrared Receiver)

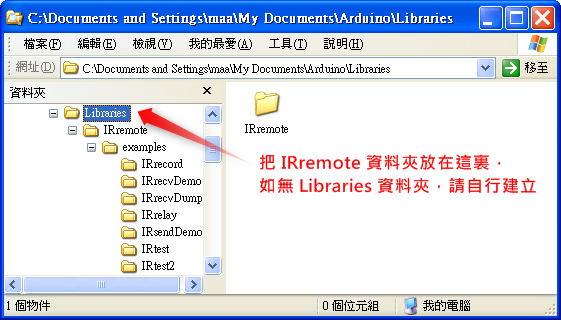
***3)*** 一顆發射波長為 940nm (nanometer, 奈米) 的紅外線發射器 (Infrared LED 或稱 Infrared Emitter)   
 [](http://lh3.ggpht.com/_Ap9EEd9K-fg/TZL_yZZLOPI/AAAAAAAAEEw/aqNcVIuGEbQ/s1600-h/irLEDblue4.jpg)   
▲ 紅外線發射器(Infrared LED)

紅外線接收器和發射器單價都很便宜，幾十塊台幣就可以搞定。

***4)*** 最後，你必須安裝 IRRemote 函式庫，見後文說明。

**安裝 IRRemote 函式庫**

到 [Ken Shirriff 的部落格](http://www.arcfn.com/2009/08/multi-protocol-infrared-remote-library.html)或[到這裏](https://github.com/coopermaa/Arduino-IRremote)下載 IRRemote 函式庫，然後把檔案解壓縮放到 Arduino **速寫簿 (Sketchbook)** 資料夾的 Libraries 資料夾底下即可。Arduino **速寫簿 (Sketchbook)** 資料夾的預設位置是在「**我的文件 > Arduino**」這個地方:

[](http://lh6.ggpht.com/_Ap9EEd9K-fg/TZL_0CdD0nI/AAAAAAAAEE4/jLSIZebpTAs/s1600-h/image111.png)  
▲ 把 IRRemote 資料夾放在「我的文件 > Arduino > Libraries」底下即可

注意！如果你用的是 Arduino 1.0，請接著參考「[如何讓 IRRemote 相容於 Arduino 1.0 ?](http://coopermaa2nd.blogspot.com/2011/12/irremote-arduino-10.html" \t "_blank)」一文設定環境。

**IRRemote 教學目錄**

***1)*** 接收紅外線:

***1.1)*** [接收紅外線訊號](http://coopermaa2nd.blogspot.com/2011/03/11.html)

***1.2)*** [顯示紅外線協定種類](http://coopermaa2nd.blogspot.com/2011/03/12.html)

***1.3)*** [增加紅外線訊號指示燈](http://coopermaa2nd.blogspot.com/2011/03/13.html)   
***1.4)*** [自製 LCD 介面的紅外線解碼器 (IR Decoder)](http://coopermaa2nd.blogspot.com/2011/03/14-lcd-ir-decoder.html)   
***1.5)*** [兩個取得遙控器紅外線資料的方法](http://coopermaa2nd.blogspot.com/2011/03/15.html)

***2)*** 發射紅外線:

***2.1)*** [發射紅外線訊號](http://coopermaa2nd.blogspot.com/2011/03/21.html)   
***2.1)*** [發射紅外線訊號 part 2](http://coopermaa2nd.blogspot.tw/2012/08/21-part-2.html)   
***2.2)*** [自製電視紅外線遙控器 (使用 Keypad)](http://coopermaa2nd.blogspot.com/2011/04/22-keypad.html)   
***2.3)*** [用光敏電阻切換電視頻道](http://coopermaa2nd.blogspot.com/2011/04/23.html)

***3)*** 應用

***3.1)*** [使用 Arduino 遙控家電](http://coopermaa2nd.blogspot.tw/2012/12/arduino.html)

**你需要知道紅外線協定嗎?**

有了 IRRemote 函式庫，就算你不知道紅外線協定原理，也可以做紅外線遙控。不過，既知其然，如果能夠進一步知其所以然，會是更好的一件事。這系列的教學沒有打算講解紅外線協定原理，如果你想知道紅外線遙控原理以及各種紅外線協定，可以到底下這個網站:

<http://www.sbprojects.com/knowledge/ir/index.php>

**延伸閱讀**

* [紅外線遙控原理與 NEC IR Protocol](http://coopermaa2nd.blogspot.com/2010/01/nec-ir-protocol.html)
* [IR Remote Control Theory](http://www.sbprojects.com/knowledge/ir/ir.htm)

**實驗目的**

練習用 IRRemote 函式庫接收紅外線訊號。在這個實驗裏，你將學習用 IRrecv 物件接收紅外線訊號。

**前提**

在做這個實驗前，我假設:

1. 你已經看過「[IRRemote 紅外線遙控教學](http://coopermaa2nd.blogspot.com/2011/03/irremote.html)」一文，而且
2. 也已經把 IRRemote 函式庫安裝起來了

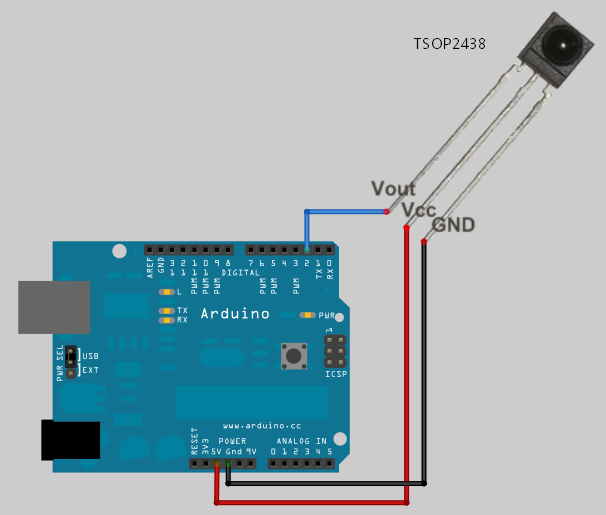
**材料**

* 麵包板 x 1
* Arduino 主板 x 1
* 38 kHz 紅外線接收器 (如 [TSOP2438](http://www.bitsbox.co.uk/data/TSOP2438.pdf),[TSOP4438](http://www.vishay.com/docs/82090/tsop48xx.pdf), [PNA4602](http://www.ladyada.net/media/sensors/PNA4602.pdf), [FM-8038TM2-5DN](http://www.icsourse.com/uploads/soft/100727/FM-8038TM2-5DN.pdf)) x 1
* 單心線 x N

**接線**

* 照下表把紅外線接收器跟 Arduino 連接起來:

|  |  |
| --- | --- |
| **紅外線接收器接腳** | **Arduino 接腳** |
| GND (-) | 接到 GND |
| Vcc (+ 或 V+) | 接到 +5V 電源 |
| Vout (或 OUT) | 接到 pin 2。你可以接到其它 Digital pin，但程式要配合修改 |

[](http://lh6.ggpht.com/_Ap9EEd9K-fg/TYFHO0-RAzI/AAAAAAAAD_M/C-t5ERxC5pE/s1600-h/image%5B5%5D.png)    
▲ 有些紅外線接收器 GND 和 Vcc 是相反的，接線時請查閱你的 datasheet

**程式**

底下這支程式 ([Example1.1.pde](https://docs.google.com/uc?id=0B4GOwiN2Qm96OTA1OGIzYTgtM2Q0ZC00ZDBlLWIwZTUtZGM4YjU5YTg3M2U4&export=download&hl=en)) 示範如何用 IRrecv 物件接收紅外線訊號:

|  |  |
| --- | --- |
| 01 | /\* |
| 02 | \* IRRemote 紅外線遙控教學 | |

|  |  |  |
| --- | --- | --- |
| 03 | \*   範例 1.1: 接收紅外線 | |
| 04 | \*/ |

|  |  |  |
| --- | --- | --- |
| 05 | #include <IRremote.h>                    // 引用 IRRemote 函式庫 | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | const int irReceiverPin = 2;             // 紅外線接收器 OUTPUT 訊號接在 pin 2 | |
| 08 |  |

|  |  |
| --- | --- |
| 09 | IRrecv irrecv(irReceiverPin);            // 定義 IRrecv 物件來接收紅外線訊號 |
| 10 | decode\_results results;                  // 解碼結果將放在 decode\_results 結構的 result 變數裏 |

|  |  |
| --- | --- |
| 11 |  |
| 12 | void setup() | |

|  |  |
| --- | --- |
| 13 | { |
| 14 | Serial.begin(9600);                     // 開啟 Serial port, 通訊速率為 9600 bps | |

|  |  |  |
| --- | --- | --- |
| 15 | irrecv.enableIRIn();                   // 啟動紅外線解碼 | |
| 16 | } |

|  |  |
| --- | --- |
| 17 |  |
| 18 | void loop() | |

|  |  |
| --- | --- |
| 19 | { |
| 20 | if (irrecv.decode(&results)) {         // 解碼成功，收到一組紅外線訊號 | |

|  |  |
| --- | --- |
| 21 | // 印到 Serial port |
| 22 | Serial.print("irCode: "); | |

|  |  |  |
| --- | --- | --- |
| 23 | Serial.print(results.value, HEX);    // 紅外線編碼 | |
| 24 | Serial.print(",  bits: "); |

|  |  |
| --- | --- |
| 25 | Serial.println(results.bits);        // 紅外線編碼位元數 |
| 26 | irrecv.resume();                    // 繼續收下一組紅外線訊號 | |

|  |  |  |
| --- | --- | --- |
| 27 | } | |
| 28 | } |

在收到紅外線訊號的時候，程式會順便把收到的紅外線編碼以及編碼長度 (位元數) 列印到 Serial Port 上，所以在跑程式的時候就可以用 Serial Monitor 來觀察解碼結果。

**如何用 IRrecv 物件接收紅外線訊號**

要接收紅外線訊號，有四個步驟要做:

***1)*** 引用 IRRemote 函式庫:

|  |  |
| --- | --- |
| 1 | #include <IRremote.h> |

***2)*** 建立一個 IRrecv 的物件以及一個用來存放解碼結果的 decode\_results 結構的變數:

|  |  |  |
| --- | --- | --- |
| 1 | const int irReceiverPin = 2;             // 紅外線接收器 OUTPUT 訊號接在 pin 2 | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | IRrecv irrecv(irReceiverPin);            // 定義 IRrecv 物件來接收紅外線訊號 |
| 4 | decode\_results results;                  // 解碼結果將放在 decode\_results 結構的 result 變數裏 |

***3)*** 呼叫 enableIRIn() 啟動紅外線解碼:

|  |  |
| --- | --- |
| 1 | irrecv.enableIRIn();                   // 啟動紅外線解碼 |

***4)*** 這時候 IRrecv 物件會在背景收紅外線訊號。接著，就可以在 loop() 函式裏用 decode() 函式持續的檢查，看看有沒有解碼成功:

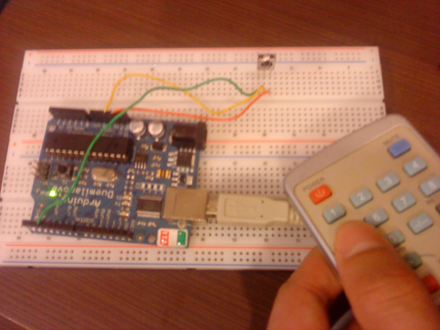
|  |  |
| --- | --- |
| 1 | irrecv.decode(&results) |

假如有成功解碼，decode() 函式會回傳 true，並且把解碼結果放在 result 變數裏。

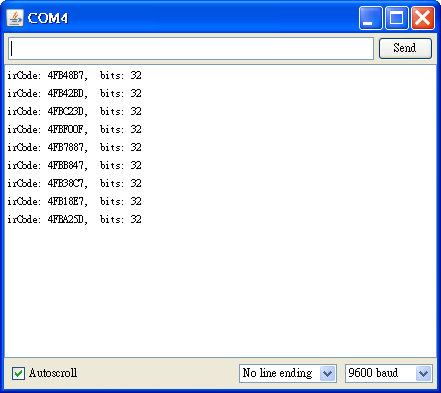
在解到一個紅外線編碼後，記得要呼叫 resume() 函式，這樣才能繼續收下一組訊號。

**示範照片**

隨手拿起一個電視遙控器按幾個鈕測試：

[](http://lh5.ggpht.com/_Ap9EEd9K-fg/TZMFLWqI2uI/AAAAAAAAEFA/r58jffCEQEY/s1600-h/image1.png)  
▲ 接受紅外線訊號

打開 Serial Monitor 觀察，我得到的解碼結果如下:

[](http://lh6.ggpht.com/_Ap9EEd9K-fg/TZMFNxtn0vI/AAAAAAAAEFI/eVsaOHW_UVM/s1600-h/image4.png)   
▲ 上圖中，紅外線編碼是以 16 進位格式顯示，我遙控器的編碼長度皆為 32 位元

**動動腦**

1. 請找一個紅外線遙控器，電視、音響、手機、相機、投影機或是冷氣機的遙控器都可以，用這個實驗的範例程式測試，看看 IRRemote 能否為你的遙控器解碼。
2. 閱讀「[Arduino 筆記 – Lab6 控制蜂鳴器發聲](http://coopermaa2nd.blogspot.com/2010/12/arduino-lab6.html)」一文，試連接一顆蜂鳴器到 Arduino 板子上並修改範例程式，讓程式在收到紅外線時發個聲響。

##### decode\_results 結構

從前一篇中，你已經知道 decode\_results 是用來存放紅外線訊號解碼結果的結構。decode\_results 結構它的定義如下:

|  |  |  |
| --- | --- | --- |
| 1 | class decode\_results { | |
| 2 | public: |

|  |  |  |
| --- | --- | --- |
| 3 | int decode\_type;      // 協定種類: NEC, SONY, RC5, RC6, DISH, SHARP, UNKNOWN | |
| 4 | unsigned long value;  // 解到的編碼數值 |

|  |  |  |
| --- | --- | --- |
| 5 | int bits;             // 編碼的位元總數(Number of bits in decoded value) | |
| 6 | volatile unsigned int \*rawbuf; // 紅外線訊號原始波形資料 |

|  |  |  |
| --- | --- | --- |
| 7 | int rawlen;                    // rawbuf 的記錄總筆數(Number of records in rawbuf) | |
| 8 | }; |

下表是 decode\_results 結構各個欄位的說明:

|  |  |
| --- | --- |
| **decode\_results 欄位名稱** | **說明** |
| decode\_type | 協定種類，包括 NEC, SONY, RC5, RC6, DISH 以及 SHARP 等紅外線協定，若解析不出則為 UNKNOWN。 |
| value | 解到的編碼數值 |
| bits | 編碼的位元數(即資料長度) |
| rawbuf | 紅外線訊號原始波形資料 |
| rawlen | rawbuf 的記錄總筆數 |

我們已經看過 value 和 bits 欄位。rawbuf, rawlen 是紅外線訊號的原始波形資料與記錄總筆數，這兩個欄位是比較深入的東西，一般而言，這兩個欄位的機會非常少，所以目前你可以先暫時忽略不管。

這次實驗的目的是判斷並顯示紅外線協定的種類，因此程式將會用到 decode\_type 欄位。

##### 程式

底下的程式([Example1.2.pde](https://docs.google.com/uc?id=0B4GOwiN2Qm96NDU3Y2FjNGYtYzE2My00ZGY5LTljOTAtMGYxMWI3NzU3ZmVj&export=download&hl=en))示範如何用 decode\_type 欄位來判斷並顯示紅外線協定種類:

|  |  |
| --- | --- |
| 01 | /\* |
| 02 | \* IRRemote 紅外線遙控教學 | |

|  |  |  |
| --- | --- | --- |
| 03 | \*   範例 1.2: 顯示紅外線協定種類，如 NEC, Sony SIRC, Philips RC5, Philips RC6 等協定 | |
| 04 | \*/ |

|  |  |  |
| --- | --- | --- |
| 05 | #include <IRremote.h>                    // 引用 IRRemote 函式庫 | |
| 06 |  |

|  |  |  |
| --- | --- | --- |
| 07 | const int irReceiverPin = 2;             // 紅外線接收器 OUTPUT 訊號接在 pin 2 | |
| 08 |  |

|  |  |
| --- | --- |
| 09 | IRrecv irrecv(irReceiverPin);            // 定義 IRrecv 物件來接收紅外線訊號 |
| 10 | decode\_results results;                  // 解碼結果將放在 decode\_results 結構的 result 變數裏 |

|  |  |
| --- | --- |
| 11 |  |
| 12 | void setup() | |

|  |  |
| --- | --- |
| 13 | { |
| 14 | Serial.begin(9600);                     // 開啟 Serial port, 通訊速率為 9600 bps | |

|  |  |  |
| --- | --- | --- |
| 15 | irrecv.enableIRIn();                   // 啟動紅外線解碼 | |
| 16 | } |

|  |  |
| --- | --- |
| 17 |  |
| 18 | // 顯示紅外線協定種類 | |

|  |  |  |
| --- | --- | --- |
| 19 | void showIRProtocol(decode\_results \*results) | |
| 20 | { |

|  |  |  |
| --- | --- | --- |
| 21 | Serial.print("Protocol: "); | |
| 22 |  |

|  |  |
| --- | --- |
| 23 | // 判斷紅外線協定種類 |
| 24 | switch(results->decode\_type) { | |

|  |  |
| --- | --- |
| 25 | case NEC: |
| 26 | Serial.print("NEC"); | |

|  |  |
| --- | --- |
| 27 | break; |
| 28 | case SONY: | |

|  |  |  |
| --- | --- | --- |
| 29 | Serial.print("SONY"); | |
| 30 | break; |

|  |  |
| --- | --- |
| 31 | case RC5: |
| 32 | Serial.print("RC5"); | |

|  |  |
| --- | --- |
| 33 | break; |
| 34 | case RC6: | |

|  |  |  |
| --- | --- | --- |
| 35 | Serial.print("RC6"); | |
| 36 | break; |

|  |  |
| --- | --- |
| 37 | default: |
| 38 | Serial.print("Unknown encoding"); | |

|  |  |  |
| --- | --- | --- |
| 39 | } | |
| 40 |  |

|  |  |
| --- | --- |
| 41 | // 把紅外線編碼印到 Serial port |
| 42 | Serial.print(", irCode: "); | |

|  |  |  |
| --- | --- | --- |
| 43 | Serial.print(results->value, HEX);    // 紅外線編碼 | |
| 44 | Serial.print(",  bits: "); |

|  |  |  |
| --- | --- | --- |
| 45 | Serial.println(results->bits);        // 紅外線編碼位元數 | |
| 46 | } |

|  |  |
| --- | --- |
| 47 |  |
| 48 | void loop() | |

|  |  |
| --- | --- |
| 49 | { |
| 50 | if (irrecv.decode(&results)) {         // 解碼成功，收到一組紅外線訊號 | |

|  |  |
| --- | --- |
| 51 | showIRProtocol(&results);            // 顯示紅外線協定種類 |
| 52 | irrecv.resume();                     // 繼續收下一組紅外線訊號 | |

|  |  |  |
| --- | --- | --- |
| 53 | } | |
| 54 | } |

程式碼沒有太大的變動，只是增加一個 showIRProtocol() 函式來顯示紅外線協定的種類而已，相信你一定看得懂。

如果你用 Arduino 內建的 LED 當作指示燈或是把 LED 指示燈接在 pin 13 上，IRrecv 也有提供一個 blink13() 的函式，你只要呼叫 blink13(true) 並傳入 true 參數，接著當收到紅外線訊號時，指示燈就會自動閃爍幾下。底下的程式([Example1.3v2.pde](https://docs.google.com/uc?id=0B4GOwiN2Qm96MDM1MzgyYjctZTcyYy00MmQyLWEwM2YtNTkzMGIwZTA0YjRj&export=download&hl=en))示範 blink13() 函式的用法:

|  |  |  |  |
| --- | --- | --- | --- |
|  | void setup() | | |
| 13 | | { |

|  |  |
| --- | --- |
| 14 | Serial.begin(9600);                     // 開啟 Serial port, 通訊速率為 9600 bps |
| 15 | irrecv.blink13(true);                  // 收到紅外線時, 讓板子上 pin 13 的指示燈閃爍幾下 |

|  |  |  |
| --- | --- | --- |
| 16 | irrecv.enableIRIn();                   // 啟動紅外線解碼 | |
| 17 | } |

[**2.1 發射紅外線訊號**](http://coopermaa2nd.blogspot.co.uk/2011/03/21.html)

**實驗目的**

練習用 IRRemote 函式庫發射紅外線訊號。在這個實驗裏，你將學習用 IRsend 物件發射紅外線訊號。

**前提**

在做這個實驗前，我假設:

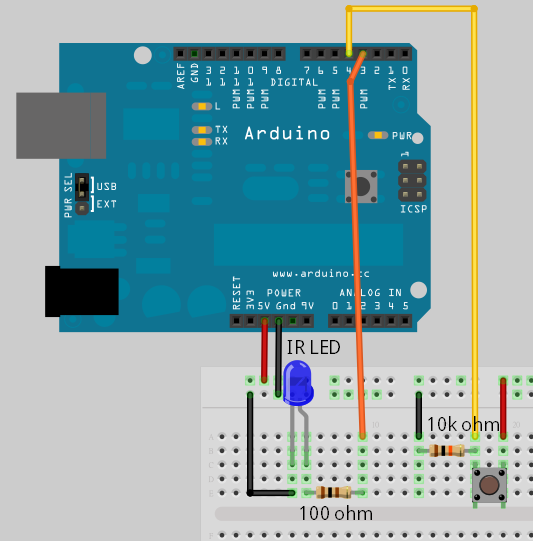
1. 你已經看過「[IRRemote 紅外線遙控教學](http://coopermaa2nd.blogspot.com/2011/03/irremote.html)」一文，而且
2. 也已經把 IRRemote 函式庫安裝起來了

**材料**

* 麵包板 x 1
* Arduino 主板 x 1
* 發射波長為 940nm 的紅外線發射器 x 1
* 按鈕 (pushbutton) x 1
* 10k 歐姆電阻 x 1
* 100 歐姆電阻 x 1
* 單心線 x N

**接線**

* 把紅外線發射器 (IR LED) 接到 pin 3，長腳 (陽極) 串接一顆 100 ohm 電阻接到 pin 3，短腳 (陰極) 接到 GND
* 按鈕的一支腳接到 +5V，另一支腳接到 pin 4 同時接一個 10K 電阻連到 GND

[](http://lh5.ggpht.com/_Ap9EEd9K-fg/TZRpHMLEfII/AAAAAAAAEGY/ILpFtSu6WQY/s1600-h/image4.png)

**注意！**因為 IRRemote 函式庫限定使用 pin 3，所以你只能把紅外線發射器接到 pin 3，不能接到其它腳位。

**程式**

底下這支程式 ([Example2.1.pde](https://docs.google.com/uc?id=0B4GOwiN2Qm96NmMzMDNjZmMtM2ExMy00ZGI4LThkMDUtNGQ4YWM5N2E0ZTIw&export=download&hl=en)) 示範如何用 IRsend 物件發射紅外線訊號:

|  |  |
| --- | --- |
| 01 | /\* |
| 02 | \* IRRemote 紅外線遙控教學 | |

|  |  |
| --- | --- |
| 03 | \*   範例 2.1: 發射紅外線訊號 |
| 04 | \*     注意! 紅外線發射器(IR LED)必須接到 pin 3, 不能接其它腳位 | |

|  |  |
| --- | --- |
| 05 | \*/ |
| 06 | #include <IRremote.h>                    // 引用 IRRemote 函式庫 | |

|  |  |
| --- | --- |
| 07 |  |
| 08 | const int buttonPin = 4;                 // 按鍵(pushbutton)接在 pin 4 | |

|  |  |  |
| --- | --- | --- |
| 09 | int buttonState = 0;                     // 按鍵狀態 | |
| 10 |  |

|  |  |  |
| --- | --- | --- |
| 11 | IRsend irsend;                           // 定義 IRsend 物件來發射紅外線訊號 | |
| 12 |  |

|  |  |  |
| --- | --- | --- |
| 13 | void setup() | |
| 14 | { |

|  |  |  |
| --- | --- | --- |
| 15 | pinMode(buttonPin, INPUT);             // 把 buttonPin 設置成 INPUT | |
| 16 | } |

|  |  |
| --- | --- |
| 17 |  |
| 18 | void loop() | |

|  |  |
| --- | --- |
| 19 | { |
| 20 | // 讀取按鍵狀態 | |

|  |  |  |
| --- | --- | --- |
| 21 | buttonState = digitalRead(buttonPin); | |
| 22 |  |

|  |  |
| --- | --- |
| 23 | // 檢查按鍵是否被按下(pressed) |
| 24 | // 有的話，buttonState 會是 HIGH | |

|  |  |  |
| --- | --- | --- |
| 25 | if (buttonState == HIGH) { | |
| 26 | // 發射紅外線訊號 |

|  |  |  |
| --- | --- | --- |
| 27 | irsend.sendNEC(0x4FB48B7, 32);   // 我電視 Power 鈕的紅外線編碼, 記得換成你的紅外線編碼 | |
| 28 | } |

|  |  |
| --- | --- |
| 29 | } |

在按下按鈕的時候，程式就會發射一個 NEC 協定(NEC IR Protocol)的紅外線訊號，紅外線編碼為 0x4FB48B7，資料長度為 32 位元。

0x4FB48B7 是我電視遙控器 Power 鈕的紅外線編碼，有關如何取得遙控器的紅外線資料，請見「[1.5 兩個取得遙控器紅外線資料的方法](http://coopermaa2nd.blogspot.com/2011/03/15.html)」一文。

**如何用 IRsend 發射紅外線訊號**

要發射紅外線訊號，只要三個步驟:

***1)*** 引用 IRRemote 函式庫:

|  |  |
| --- | --- |
| 1 | #include <IRremote.h> |

***2)*** 建立一個 IRsend 的物件:

|  |  |
| --- | --- |
| 1 | IRsend irsend;                           // 定義 IRsend 物件來發射紅外線訊號 |

***3)*** 呼叫對應紅外線協定的 send() 函式並傳入紅外線編碼與資料長度，就可以送出一個紅外線訊號:

|  |  |
| --- | --- |
| 1 | irsend.sendNEC(0x4FB48B7, 32);   // 我電視 Power 鈕的紅外線編碼, 記得換成你的紅外線編碼 |

IRRemote 函式庫支援 NEC, Sony SIRC, Philips RC5, Philips RC6, DISH, Sharp 這些紅外線協定，所以 IRsend 有下這幾個 send() 函式可用:

* sendNEC(data, nbits)
* sendSony(data, nbits)
* sendRC5(data, nbits)
* sendRC6(data, nbits)
* sendDISH(data, nbits)
* sendSharp(data, nbits), 及
* sendRaw(buf[], len, hz) - 進階的函式，可指定頻率傳送指定的波形 (waveform)。

**示範影片**

底下這個影片示範用 Arduino 控制關電視機的開關:

**用 Serial Port 控制紅外線的發射**

如果你手上沒有按鈕 (Pushbutton)，可以用底下這個版本的程式([Example2.1v2.pde](https://docs.google.com/uc?id=0B4GOwiN2Qm96ZjFkYTkxMmEtYzQyNC00MWNjLTkwYTUtMGI3NTc5NjQ4MTFl&export=download&hl=en))：

|  |  |
| --- | --- |
| 01 | /\* |
| 02 | \* IRRemote 紅外線遙控教學 | |

|  |  |
| --- | --- |
| 03 | \*   範例 2.1v2: 發射紅外線訊號 |
| 04 | \*     注意! 紅外線發射器(IR LED)必須接到 pin 3, 不能接其它腳位 | |

|  |  |
| --- | --- |
| 05 | \*/ |
| 06 | #include <IRremote.h>                    // 引用 IRRemote 函式庫 | |

|  |  |
| --- | --- |
| 07 |  |
| 08 | IRsend irsend;                           // 定義 IRsend 物件來發射紅外線訊號 | |

|  |  |
| --- | --- |
| 09 |  |
| 10 | void setup() | |

|  |  |
| --- | --- |
| 11 | { |
| 12 | Serial.begin(9600);                     // 開啟 Serial port, 通訊速率為 9600 bps | |

|  |  |  |
| --- | --- | --- |
| 13 | } | |
| 14 |  |

|  |  |  |
| --- | --- | --- |
| 15 | void loop() | |
| 16 | { |

|  |  |
| --- | --- |
| 17 | // 檢查 Serial Port 是否有資料 |
| 18 | // 有的話，Serial.read() 會回傳非 -1 的數值 | |

|  |  |  |
| --- | --- | --- |
| 19 | if (Serial.read() != -1) { | |
| 20 | // 發射紅外線訊號 |

|  |  |  |
| --- | --- | --- |
| 21 | irsend.sendNEC(0x4FB48B7, 32);   // 我電視 Power 鈕的紅外線編碼, 記得換成你的紅外線編碼 | |
| 22 | } |

|  |  |
| --- | --- |
| 23 | } |

這支程式會不斷地檢查 Serial Port，如果 Serial Port 上有資料進來，就會發射紅外線訊號，因此你只要打開 Serial Monitor，隨便輸入任何資料就可以測試紅外線的發射，。

**動動腦**

1. 請找一個紅外線遙控器，電視、音響、手機、相機、投影機或是冷氣機的遙控器都可以，先取得遙控器任意一個按鈕的紅外線資料，然後再修改這個實驗的範例程式，用 Arduino 來控制你的影音設備。
2. 閱讀「[Arduino 筆記 – Lab8 使用光敏電阻控制 LED 的開關](http://coopermaa2nd.blogspot.com/2010/12/arduino-lab8-led.html)」一文，然後把按鈕換成光敏電阻，用光敏電阻來控制電視機。你的程式邏輯可以這樣寫: 當光敏電阻讀值有大幅度的變化時，就送出 Power 鈕紅外線訊號。

但是萬一你 pin3 有其它用途，真的不能挪來接紅外線發射器，那要怎麼辦才好呢？還好，其實 IRRemote Library 有提供一個選項可以讓你選擇其它腳位，方法是修改 IRRemote Library 的 IRremoteInt.h 這一段:

|  |  |  |
| --- | --- | --- |
| 01 | // define which timer to use | |
| 02 | // |

|  |  |
| --- | --- |
| 03 | // Uncomment the timer you wish to use on your board.  If you |
| 04 | // are using another library which uses timer2, you have options | |

|  |  |  |
| --- | --- | --- |
| 05 | // to switch IRremote to use a different timer. | |
| 06 |  |

|  |  |
| --- | --- |
| 07 | // Arduino Mega |
| 08 | #if defined(\_\_AVR\_ATmega1280\_\_) || defined(\_\_AVR\_ATmega2560\_\_) | |

|  |  |  |
| --- | --- | --- |
| 09 | //#define IR\_USE\_TIMER1   // tx = pin 11 | |
| 10 | #define IR\_USE\_TIMER2     // tx = pin 9 |

|  |  |
| --- | --- |
| 11 | //#define IR\_USE\_TIMER3   // tx = pin 5 |
| 12 | //#define IR\_USE\_TIMER4   // tx = pin 6 |

|  |  |  |
| --- | --- | --- |
| 13 | //#define IR\_USE\_TIMER5   // tx = pin 46 | |
| 14 |  |

|  |  |
| --- | --- |
| 15 | // Teensy 1.0 |
| 16 | #elif defined(\_\_AVR\_AT90USB162\_\_) | |

|  |  |  |
| --- | --- | --- |
| 17 | #define IR\_USE\_TIMER1     // tx = pin 17 | |
| 18 |  |

|  |  |
| --- | --- |
| 19 | // Teensy 2.0 |
| 20 | #elif defined(\_\_AVR\_ATmega32U4\_\_) | |

|  |  |  |
| --- | --- | --- |
| 21 | //#define IR\_USE\_TIMER1   // tx = pin 14 | |
| 22 | //#define IR\_USE\_TIMER3   // tx = pin 9 |

|  |  |  |
| --- | --- | --- |
| 23 | #define IR\_USE\_TIMER4\_HS  // tx = pin 10 | |
| 24 |  |

|  |  |
| --- | --- |
| 25 | // Teensy++ 1.0 & 2.0 |
| 26 | #elif defined(\_\_AVR\_AT90USB646\_\_) || defined(\_\_AVR\_AT90USB1286\_\_) | |

|  |  |  |
| --- | --- | --- |
| 27 | //#define IR\_USE\_TIMER1   // tx = pin 25 | |
| 28 | #define IR\_USE\_TIMER2     // tx = pin 1 |

|  |  |  |
| --- | --- | --- |
| 29 | //#define IR\_USE\_TIMER3   // tx = pin 16 | |
| 30 |  |

|  |  |
| --- | --- |
| 31 | // Sanguino |
| 32 | #elif defined(\_\_AVR\_ATmega644P\_\_) || defined(\_\_AVR\_ATmega644\_\_) | |

|  |  |
| --- | --- |
| 33 | //#define IR\_USE\_TIMER1   // tx = pin 13 |
| 34 | #define IR\_USE\_TIMER2     // tx = pin 14 |

|  |  |
| --- | --- |
| 35 |  |
| 36 | // Arduino Duemilanove, Diecimila, LilyPad, Mini, Fio, etc | |

|  |  |
| --- | --- |
| 37 | #else |
| 38 | //#define IR\_USE\_TIMER1   // tx = pin 9 | |

|  |  |  |
| --- | --- | --- |
| 39 | #define IR\_USE\_TIMER2     // tx = pin 3 | |
| 40 | #endif |

這段程式碼主要是設定 Timer，Timer 的設定會決定紅外線發射器所用的腳位。

以 UNO, Duemilanove, Diecimila 等板子為例，預設是 Timer2，發射器要接 pin 3，假如你不想用 pin 3 當作紅外線發射的腳位，你有一個選擇，可以換成 Timer1，這樣就可以改用 pin 9 來接紅外線發射器：

// Arduino Duemilanove, Diecimila, LilyPad, Mini, Fio, etc   
#else   
  #define IR\_USE\_TIMER1   // tx = pin 9   
  // #define IR\_USE\_TIMER2     // tx = pin 3   
#endif

修改 IRremoteInt.h 之後，接線也要同步修改：

* 把紅外線發射器 (IR LED) 接到 pin 9，長腳 (陽極) 串接一顆 100 ohm 電阻接到 pin 9，短腳 (陰極) 接到 GND
* 按鈕的一支腳接到 +5V，另一支腳接到 pin 4 同時接一個 10K 電阻連到 GND