

Arduino程式控制

科學魔法車

40個實驗範例



超音波偵測



紅外線偵測

網購: www.letry.com.tw (或搜尋"科學魔法車")

請選型號D01科學魔法車(1800元)

可進行本講義前面38個實驗

作者 曹齊平 曹為翔 2020年3月2日

- | | |
|------------------|---------------------------------------|
| 實驗1 閃爍的燈 | 實驗21 紅外線閃障、循跡、不落桌下 |
| 實驗2 讓車子動起來 | 實驗22 紅外線閃避障礙(死巷退出)
(用軟體消除雜訊干擾) |
| While 和 if 有何不同? | 實驗23 紅外線閃避障礙(死巷退出)
(用軟體消除干擾) |
| 實驗3 開關和燈 | 實驗24 紅外線閃避障礙(黑暗停車) |
| 實驗4 黑暗中發光二極體自動亮 | 實驗25 紅外線閃避障礙(死巷退出)
(黑暗停車)(用電容消除干擾) |
| 實驗5 用手的影子讓車子動起來 | 實驗26 自由落體實驗時間量測 |
| 實驗6 黑暗中並且有人,燈就亮 | 實驗27 音樂(生日快樂) |
| 實驗7 for 迴圈 | 實驗28 超音波閃避障礙的魔法車 |
| 實驗8 定義閃爍函式 | 實驗29 超音波電子琴 |
| 實驗9 紅綠燈 | 實驗30 拍拍手,燈就亮 |
| 實驗10 PWM方法控制亮度 | 實驗31 偵測拍手次數、拍手時間 |
| 實驗11 類比輸入 | 實驗32 聲控車 |
| 實驗12 越黑暗,LED越亮 | 實驗33 聲控(拍1次前進、拍2次後退..) |
| 實驗13 越黑暗車子動越快 | 實驗34 檢視你家的遙控器內碼 |
| H型馬達驅動電路 | 實驗35 紅外線遙控器電子琴 |
| 實驗14 光強度顯示器 | 實驗36 紅外線遙控車 |
| 實驗15 人影偵測警報器 | 實驗37 用三條電線設定燈閃的次數 |
| 實驗16 會跳舞的魔法車 | 實驗38 8合1程式 |
| 實驗17 前進、後退的函式 | 實驗39 舵機(伺服馬達) |
| 實驗18 用手的影子控制車子 | 實驗40 步進馬達 |
| 實驗19 跟著光線旋轉的魔法車 | |
| 實驗20 越黑暗,車子跑越快 | |

在科學魔法車上做Arduino程式控制實驗

超音波感測器

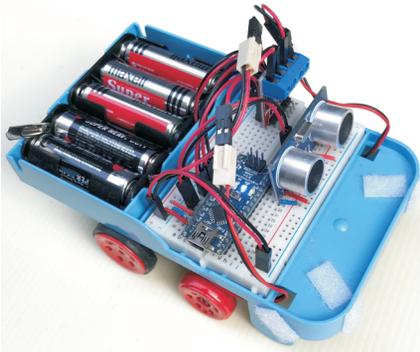


馬達驅動板



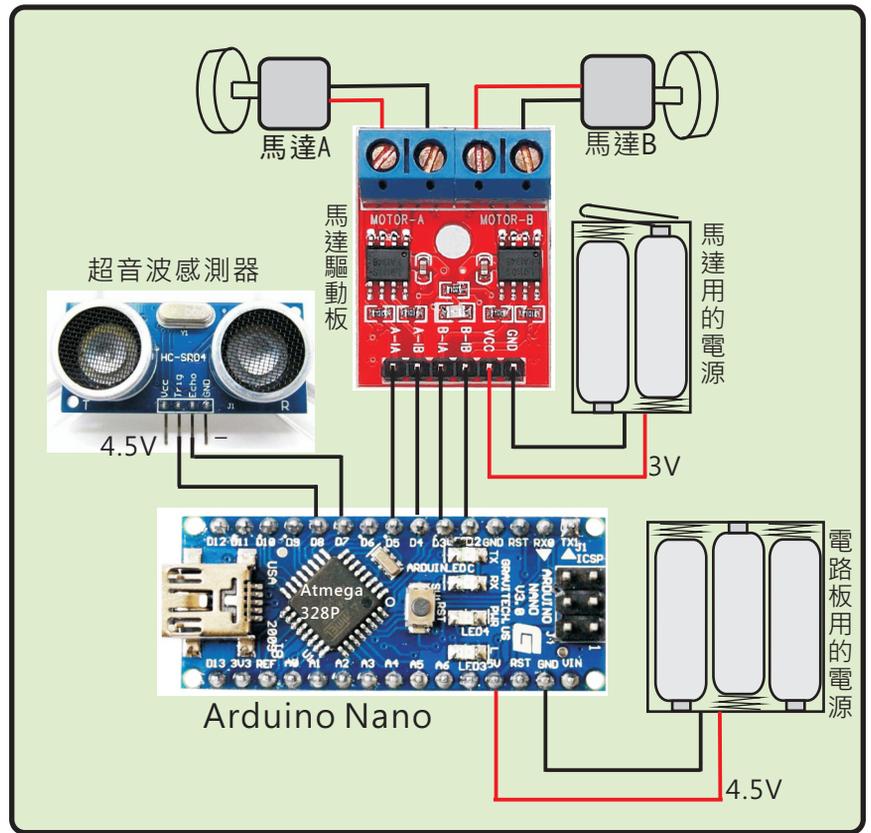
Arduino Nano

把這三個東西裝在魔法車上
再照右圖用電線接起來，
再上傳程式到晶片裡，
就完成了超音波偵測的魔法車。



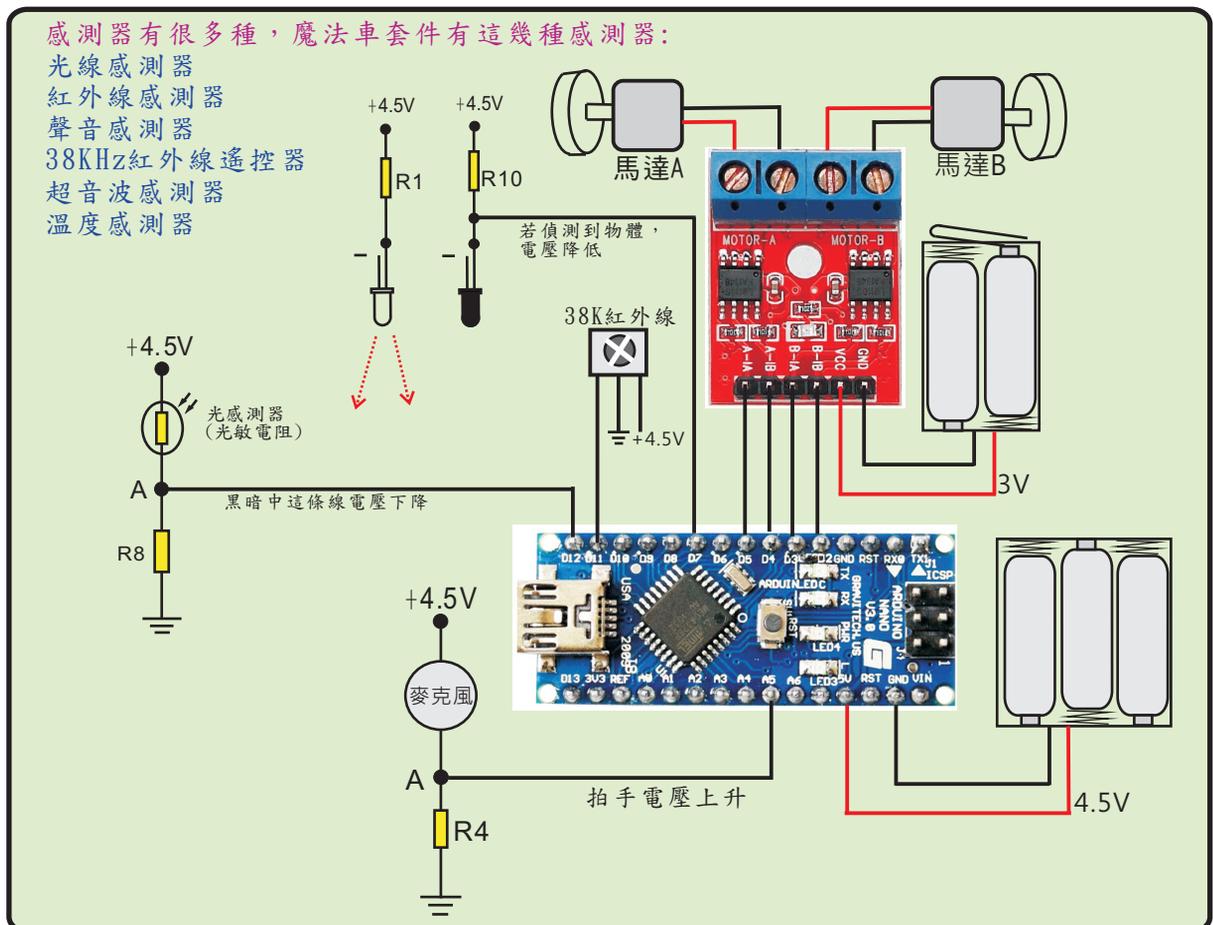
Arduino上的石英晶體震盪頻率不高
所以電源用4.5v沒問題。

超音波感測自走車



感測器有很多種，魔法車套件有這幾種感測器：

- 光線感測器
- 紅外線感測器
- 聲音感測器
- 38KHz紅外線遙控器
- 超音波感測器
- 溫度感測器



安裝Arduino軟體

先到Arduino官網下載安裝Arduino軟體，
執行後，出現下列畫面。
完成設定，就可以開始寫程式。
最重要的三個設定是下面三個：
開發板:"Arduino Nano"
處理器:"ATmega328P"
序列埠:"COMxx"

檔案 編輯 草稿碼 工具 說明

ATmega328P
ATmega328P(Old Bootloader)
ATmega168

Atmega168現在很少人用了
Bootloader 則有新的也有舊的
兩種設定都可試試看

當你把Arduino連到你的電腦，
電腦會給你的Arduino一個編號，
如果這次是COM4，
那麼下次換一片新的Arduino，
編號就是COM5。

自動格式化
封存草稿碼
修正編碼並重新載入
管理程式庫...
序列埠監控視窗
序列繪圖家
WiFi101/WIFININA
開發板:"Arduino Nano"
處理器:"ATmega328P"
序列埠:"COMxx"
取得開發板資訊
燒錄器
燒錄Bootloader

最重要的三個設定

科學魔法車採用Nano

"管理程式庫" 讓你可以很方便的
下載全世界有名的開源軟體
讓你寫程式更容易

檔案 編輯 草稿碼 工具 說明



編譯程式
若編譯成功就上傳到Arduino

編譯程式
看看文法有沒有問題

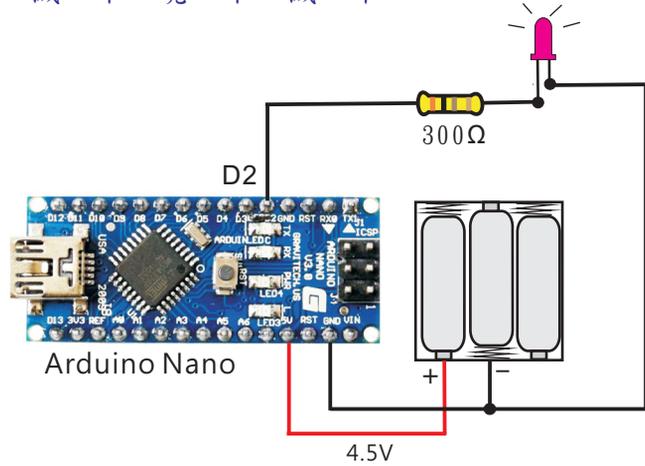
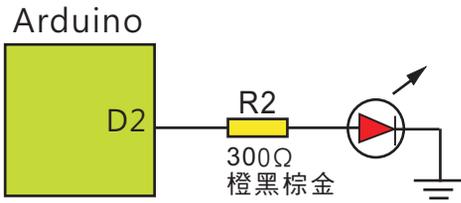
在電腦上寫好程式後
一定要上傳到Arduino才能執行

檔案 編輯 草稿碼 工具 說明

新增
開啟
開啟最近
草稿碼簿
範例
關閉
儲存 寫好程式要記得存檔
另存新檔
頁面設定
列印
偏好設定
離開

實驗1 閃爍的燈

亮一下、減一下、亮一下、減一下

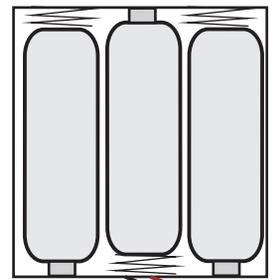


在電腦寫下列程式再上傳到Arduino內

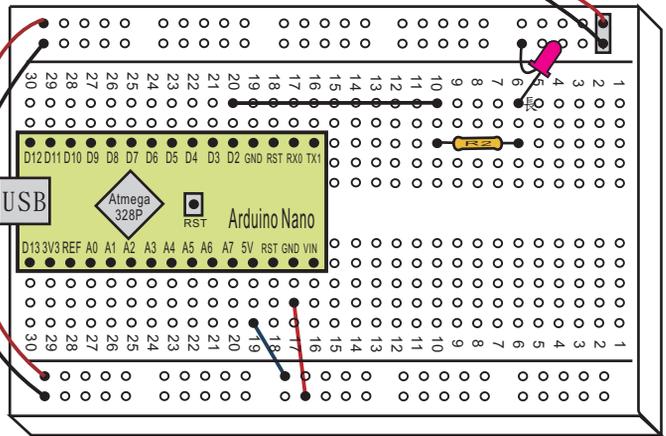
```
void setup() {
  pinMode(2,OUTPUT);
}

void loop() {
  digitalWrite(2,HIGH);
  delay(1000);
  digitalWrite(2,LOW);
  delay(1500);
}
```

在電腦上寫完程式之後，透過USB線上傳到Atmega328晶片裡面，發光二極體就開始閃滅閃滅了。USB線會提供5V電壓給arduino 程式上傳後可拔掉USB線，改由3顆電池供電。Arduino上的石英晶體震盪頻率不高，所以電源可用4.5v。



setup()底下的程式只會執行一次
loop()底下的程式會自動循環執行
本程式讓發光二極體：
亮1秒，減1.5秒，
亮1秒，減1.5秒，
亮1秒，減1.5秒，
.....



D2到D13這些腳，是當做輸入腳，還是當輸出腳，要在setup()裡做設定，用pinMode指令來設定。A0到A7是輸入類比電壓用的。

```
void loop()
{ 亮1秒;
  減1.5秒;}
```

```
void setup() {
  pinMode(2,OUTPUT); //設定D2為輸出
}

void loop() { //循環執行下列指令
  digitalWrite(2,HIGH); 亮 //輸出高電位到D2
  delay(1000); //延時1秒
  digitalWrite(2,LOW); 減 //輸出低電位到D2
  delay(1500); //延時1.5秒
}
```

檔案 編輯 草稿碼 工具 說明

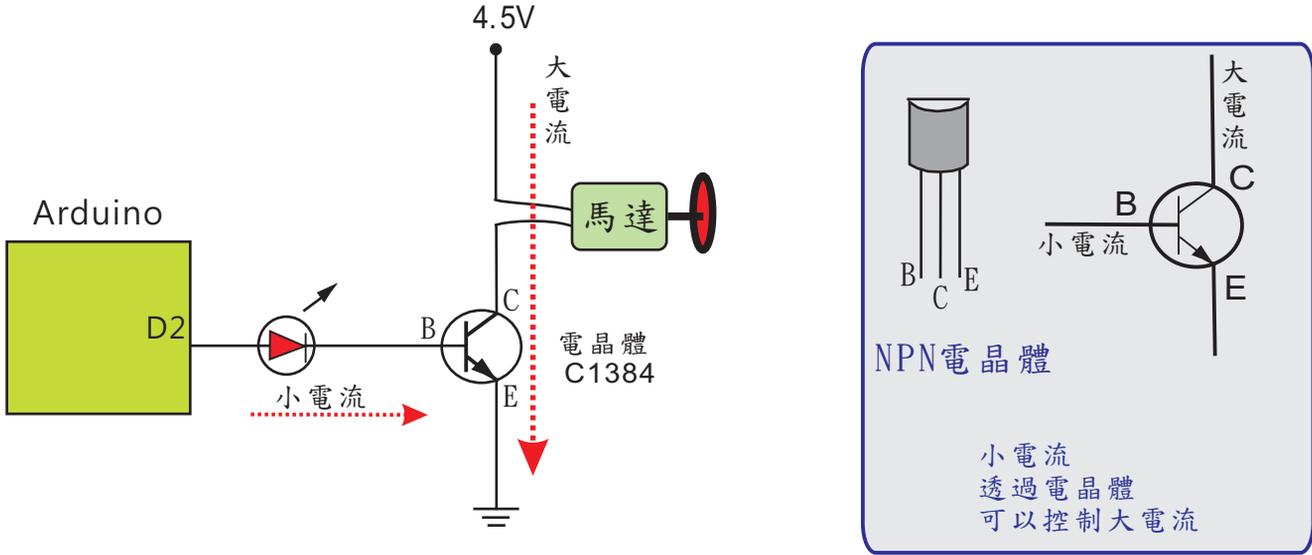
```
void setup()
{ 寫在這裡的指令
  只會執行一次 }

void loop()
{ 寫在這裡的指令
  會反覆執行 }
```

實驗2 用電晶體讓車子動起來

動一下、停一下、動一下、停一下

實驗1和實驗2的程式一模一樣

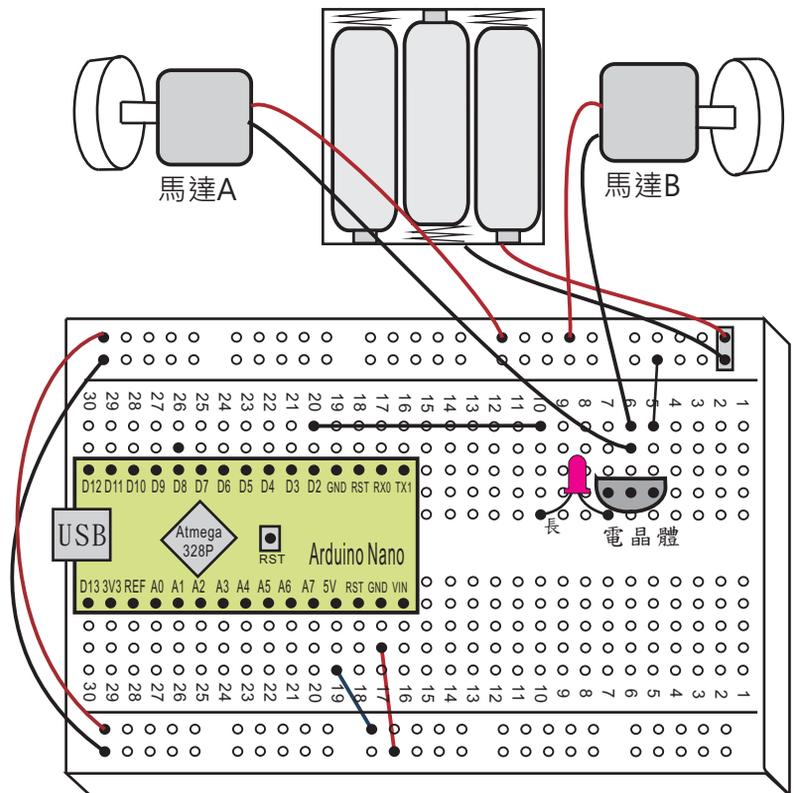


Arduino輸出的電流很小，
要透過電晶體去啟動大電流，
讓馬達動起來。

```
void setup() {  
  pinMode(2,OUTPUT);  
}  
  
void loop() {  
  digitalWrite(2,HIGH);  
  delay(1000);  
  digitalWrite(2,LOW);  
  delay(1500);  
  .....  
}
```

車子
動1秒
停1.5秒
動1秒
停1.5秒
.....

板子上面有一個小按鈕，
按一下，
程式就重新執行。



While 和 if 有何不同?

while

```

AAA;
BBB;
CCC;
while(晚上)
{DDD;
EEE;
FFF;}
GGG;
HHH;
    
```

執行結果

白天	晚上
AAA;	AAA;
BBB;	BBB;
CCC;	CCC;
GGG;	DDD;
HHH;	EEE;
	FFF;
	DDD;
	EEE;
	FFF;

在晚上DEF會一直重覆執行

while

```

void loop(){
綠燈亮;
while(晚上)
{紅燈亮;}
黃燈亮;
}
    
```

執行結果

白天	晚上
綠燈亮;	綠燈亮;
黃燈亮;	紅燈亮;
綠燈亮;	紅燈亮;
黃燈亮;	紅燈亮;
綠燈亮;	紅燈亮;
黃燈亮;	紅燈亮;

if

```

AAA;
BBB;
CCC;
if(晚上)
{DDD;
EEE;
FFF;}
GGG;
HHH;
    
```

執行結果

白天	晚上
AAA;	AAA;
BBB;	BBB;
CCC;	CCC;
GGG;	DDD;
HHH;	EEE;
	FFF;
	GGG;
	HHH;

在晚上DEF只會執行一次

if

```

void loop(){
綠燈亮;
if(晚上)
{紅燈亮;}
黃燈亮;
}
    
```

執行結果

白天	晚上
綠燈亮;	綠燈亮;
黃燈亮;	紅燈亮;
綠燈亮;	黃燈亮;
黃燈亮;	綠燈亮;
綠燈亮;	紅燈亮;
黃燈亮;	黃燈亮;
	綠燈亮;
	紅燈亮;
	黃燈亮;

if

```

void loop(){
綠燈亮;
if(晚上)
{紅燈亮;}
else
{藍燈亮;}
黃燈亮;
}
    
```

執行結果

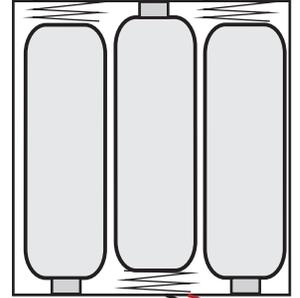
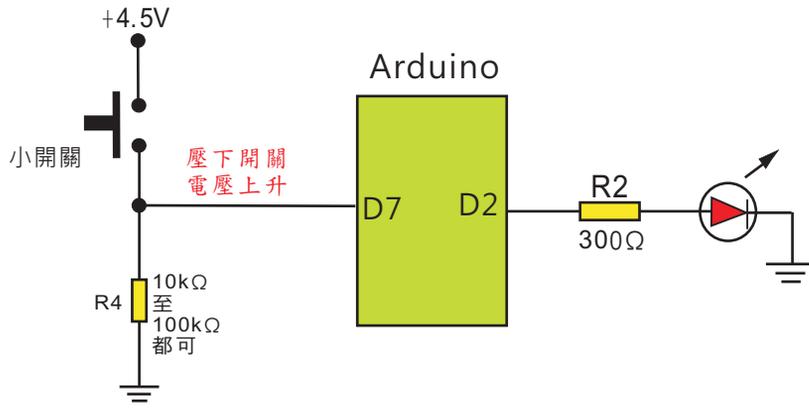
白天	晚上
綠燈亮;	綠燈亮;
藍燈亮;	紅燈亮;
黃燈亮;	黃燈亮;
綠燈亮;	綠燈亮;
藍燈亮;	紅燈亮;
黃燈亮;	黃燈亮;

while(A) 只要A成立
{D} 就不斷執行D
G 直到A 不成立
才執行G

if(A) 如果A成立
{D} 就執行D
else 否則
{G} 就執行G

if(A) 如果A成立
{D} 就執行D
else if(B) 否則如果B成立
{G} 就執行G
else if(C) 否則如果C成立
{H} 就執行H

實驗3 開關和燈



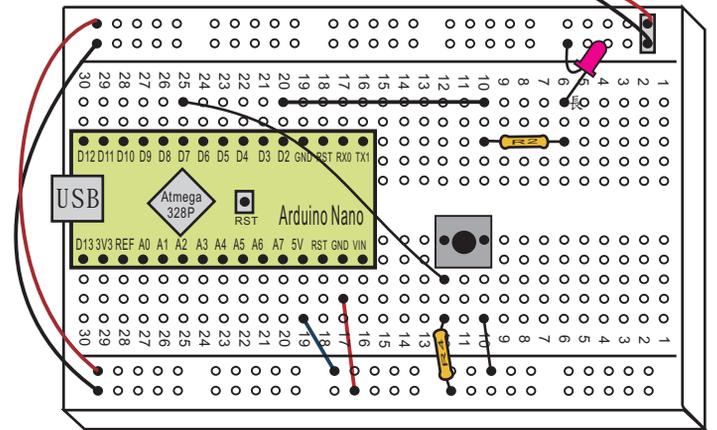
第1個程式

if(壓開關)
{燈亮2秒}

if()裡面的等於只是做邏輯判斷
不是要改變數值，
所以用兩個等號==

```
void setup() {
  pinMode(7,INPUT);
  pinMode(2,OUTPUT);}

void loop() {
  if (digitalRead(7)==HIGH)
  {digitalWrite(2,HIGH);
  delay(2000); 亮2秒
  digitalWrite(2,LOW);}
}
```



第2個程式

while (沒壓開關) {
燈亮2秒}

我們希望程式暫停，
等待某件事情發生，
再繼續執行下去，
就可以用這個句型：
while(某件事沒發生) { }

```
void setup() {
  pinMode(7,INPUT);
  pinMode(2,OUTPUT);}

void loop() {
  while (digitalRead(7)==LOW) {
    digitalWrite(2,HIGH);
    delay(2000); 亮2秒
    digitalWrite(2,LOW);
  }
```

while(A)
{D}
G

只要A成立
就不斷執行D
直到A 不成立
才執行G

第3個程式

if (壓開關)
{ 燈亮兩秒 }
else
{閃爍}

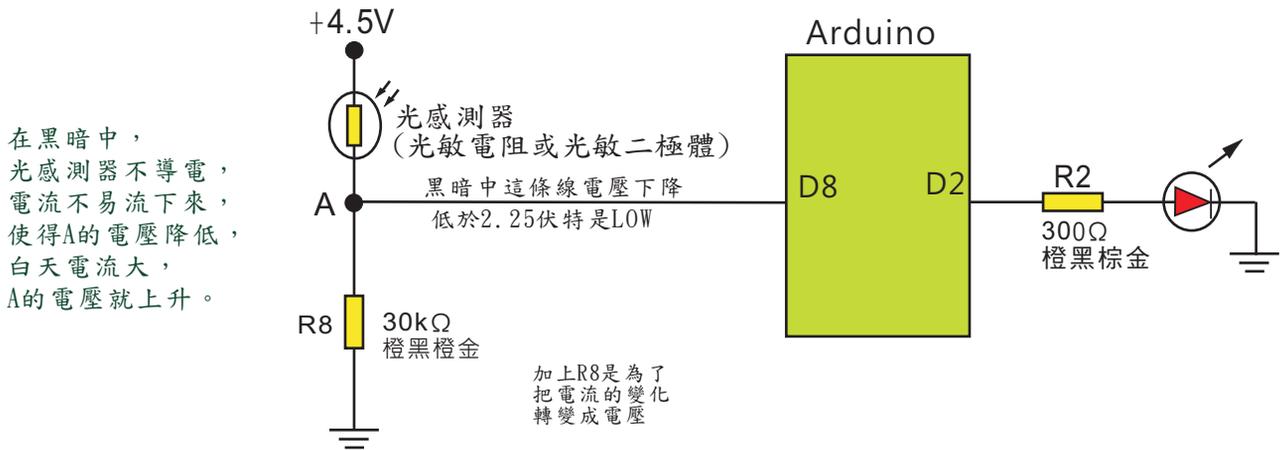
燈閃爍，壓一下開關，
持續亮2秒
然後繼續閃爍。

```
void setup() {
  pinMode(7,INPUT);
  pinMode(2,OUTPUT);}

void loop() {
  if (digitalRead(7)==HIGH)
  {digitalWrite(2,HIGH);
  delay(2000); 亮2秒
  digitalWrite(2,LOW);}
  else
  {digitalWrite(2,HIGH);
  delay(100);閃爍
  digitalWrite(2,LOW);
  delay(100); }
}
```

實驗4 黑暗中發光二極體自動亮

(用if 和 while 兩種方法)



第1個程式

```
if(D8是低電位)
就輸出高電位到D2
否則就輸出低電位到D2
```

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(8,INPUT); }

void loop() {
  if(digitalRead(8)==LOW)
  {digitalWrite(2,HIGH); }
  else { digitalWrite(2,LOW); }
}
```

第2個程式

```
while(D8是低電位)
{輸出高電位到D2}
輸出低電位到D2
```

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(8,INPUT); }

void loop() {
  while(digitalRead(8)==LOW)
  {digitalWrite(2,HIGH); }
  digitalWrite(2,LOW);
}
```

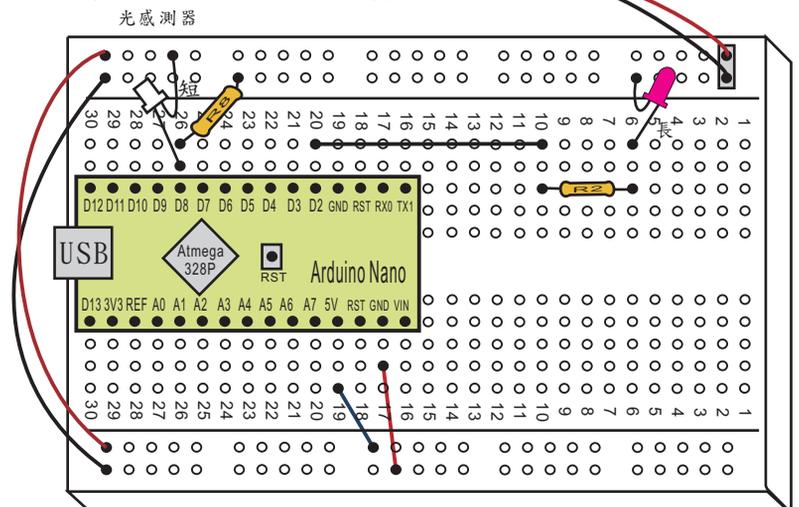
```
if (D8是低電位) {閃爍}
else {滅}
```

第3個程式

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(8,INPUT); }

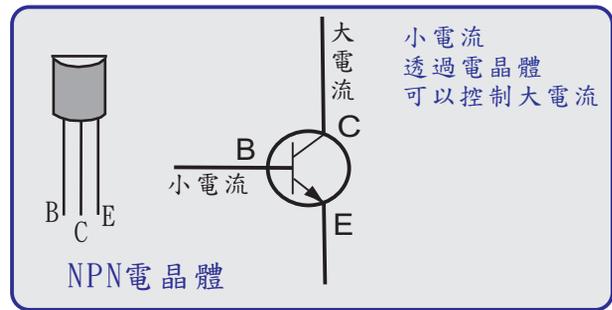
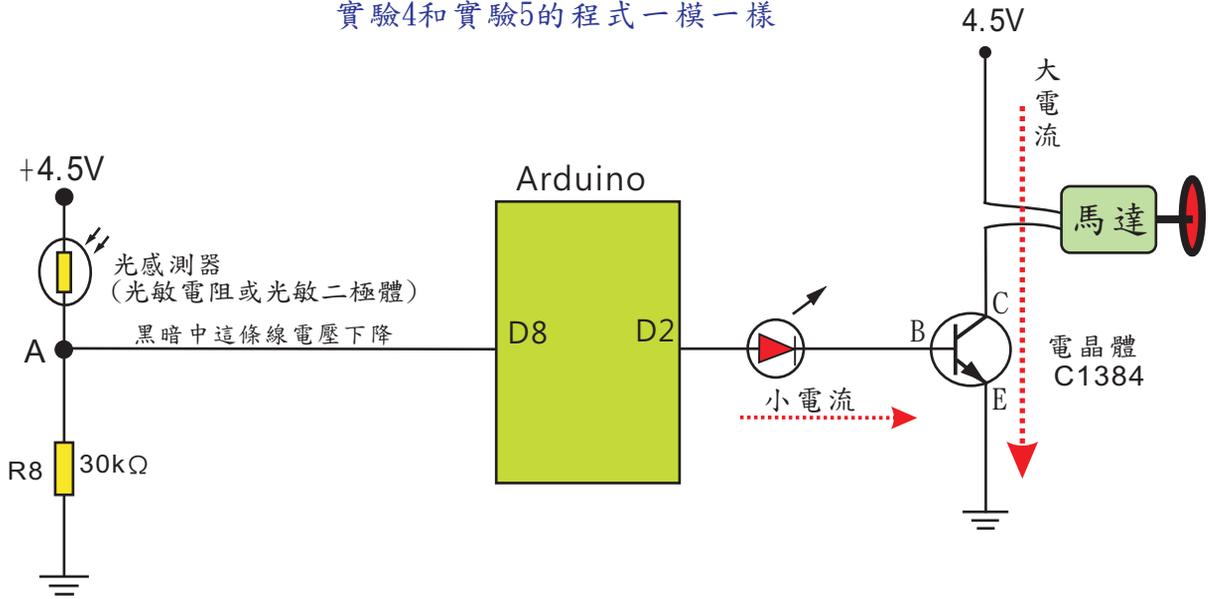
void loop() {
  if(digitalRead(8)==LOW)
  {digitalWrite(2,HIGH);
  delay(200);
  digitalWrite(2,LOW);
  delay(200);}
  else
  {digitalWrite(2,LOW);} }
}
```

注意:光敏二極體是短正長負
但有些光敏二極體是長正短負



實驗5 用手的影子讓車子動起來

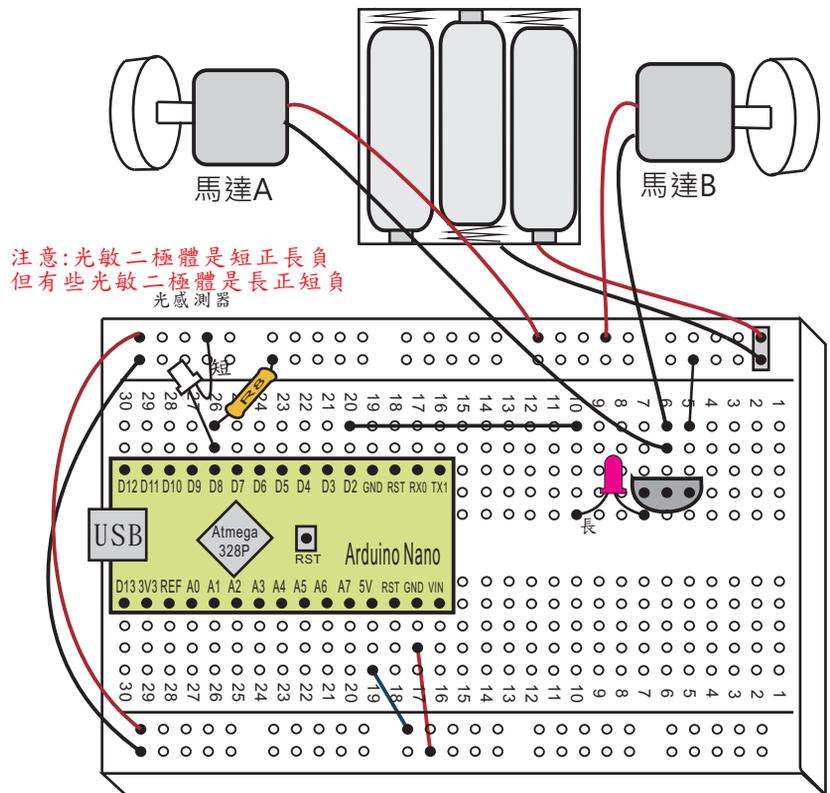
實驗4和實驗5的程式一模一樣



```
if(D8是低電位)
就輸出高電位到D2}
否則{就輸出低電位到D2}
```

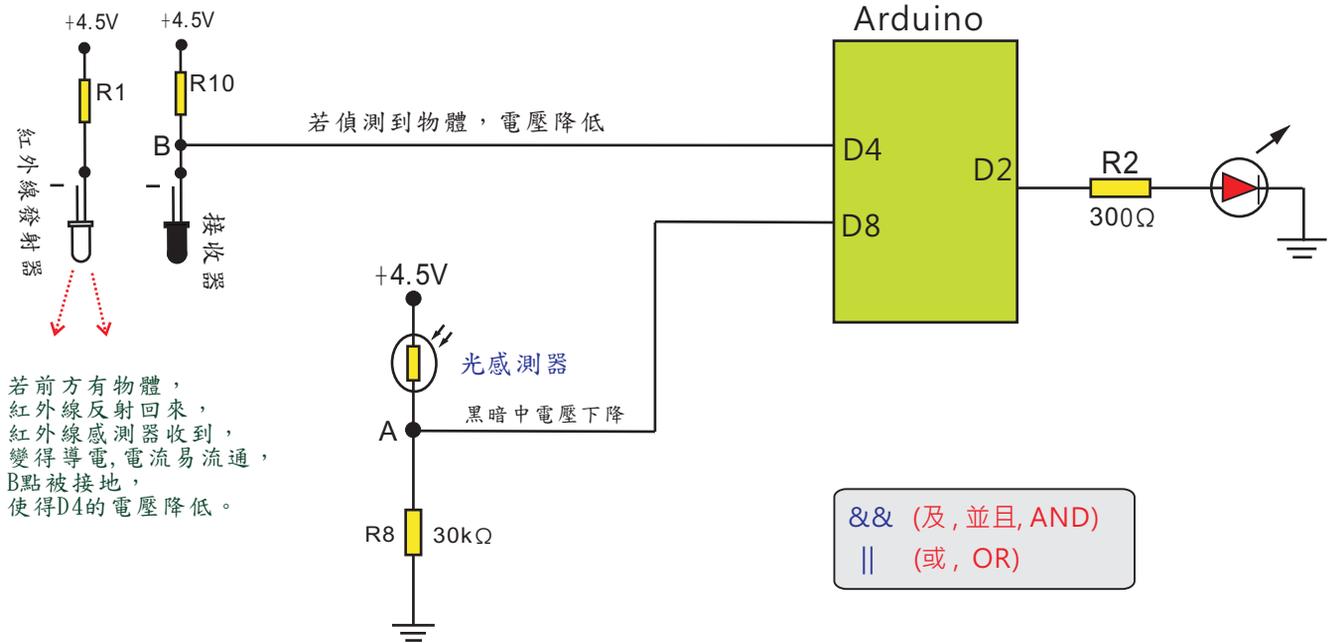
```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(8,INPUT); }

void loop() {
  if(digitalRead(8)==LOW)
  { digitalWrite(2,HIGH); }
  else
  { digitalWrite(2,LOW); }
}
```



一個電晶體只能控制馬達轉或不轉，若要控制馬達正轉或反轉，則需要4個電晶體。

實驗6 黑暗中 並且 紅外線偵測到人 發光二極體就自動亮



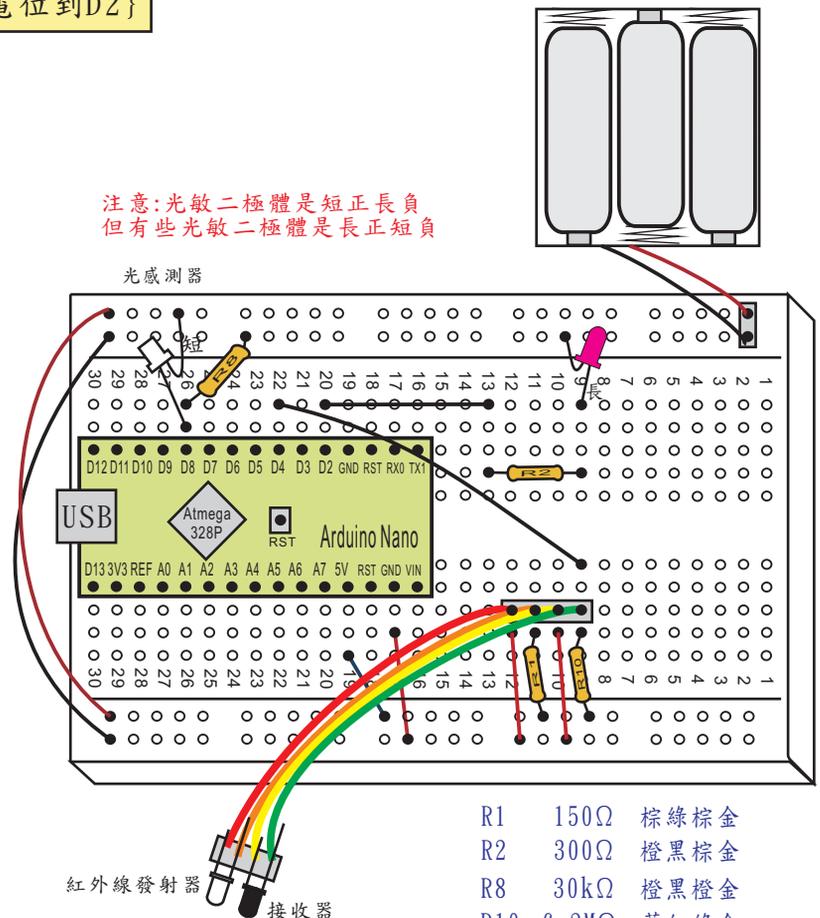
```
if(黑暗 並且 有人)
就 {燈亮}
否則 {燈滅}
```

```
if(D4和D8都是低電位)
{輸出高電位到D2}
else {就輸出低電位到D2}
```

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(4,INPUT);
  pinMode(8,INPUT);
}

void loop() {
  if(digitalRead(4)==LOW
  && digitalRead(8)==LOW) {
    digitalWrite(2,HIGH);
  }
  else {
    digitalWrite(2,LOW);
  }
}
```

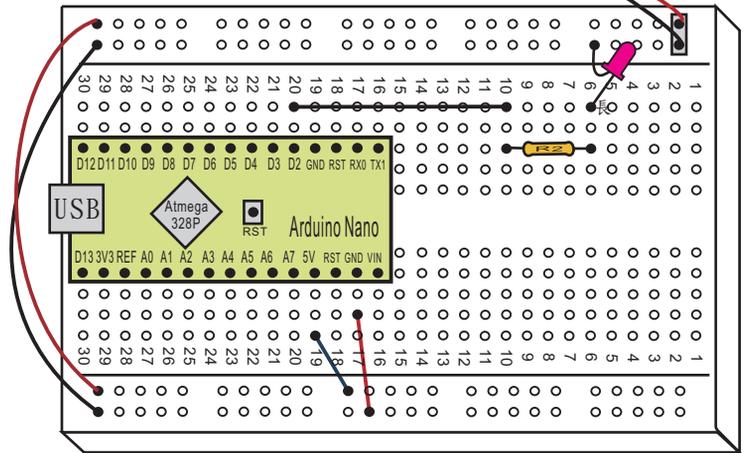
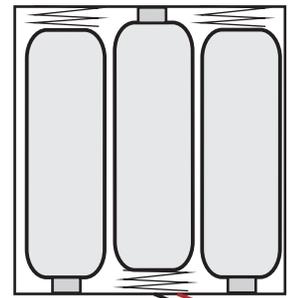
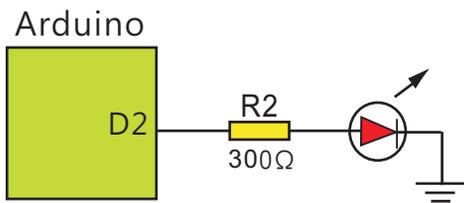
把 && 改成 ||
再執行看看有何不同



實驗7 for 迴圈和閃爍的燈

```
for (開始值; 結束值; 每次迴圈增值)
{ 程式
}
```

程式不斷反覆執行
直到結束值發生



第1個程式 閃爍10次

```
for (i=1;i<=10;i=i+1)
{ 亮0.5秒
  滅0.5秒 }
```

{ }裡面的程式會執行10次

第2個程式 閃爍8次 越閃越慢

```
for (n=100;n<=800;n=n+100)
{ 亮0.1*n秒
  滅0.1*n秒 }
```

{ }裡面的程式會執行8次

第1次n=100

第2次n=200

...

第8次n=800

```
int i;
void setup() {
  pinMode(2,OUTPUT); }

void loop(){
  for (i = 1; i <= 10; i=i+1){
    digitalWrite(2,HIGH);
    delay(500);
    digitalWrite(2,LOW);
    delay(500); }
  delay(2000);
}
```

整數的英文是integer
int i 宣告i是整數變數

閃爍10次

等待2秒

```
int i;
void setup() {
  pinMode(2,OUTPUT); }

void loop(){
  for (i = 100; i <= 800; i=i+100){
    digitalWrite(2,HIGH);
    delay(i);
    digitalWrite(2,LOW);
    delay(500); }
  delay(2000);
}
```

閃爍8次

第1次亮0.1秒

滅0.5秒後

第2次亮0.2秒

滅0.5秒後

第3次亮0.3秒

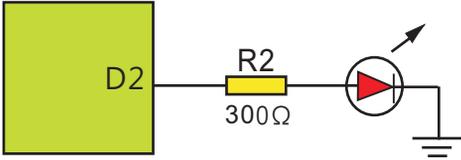
.....

第8次亮0.8秒

delay(2000); 等待2秒

實驗8 定義閃爍函式

Arduino

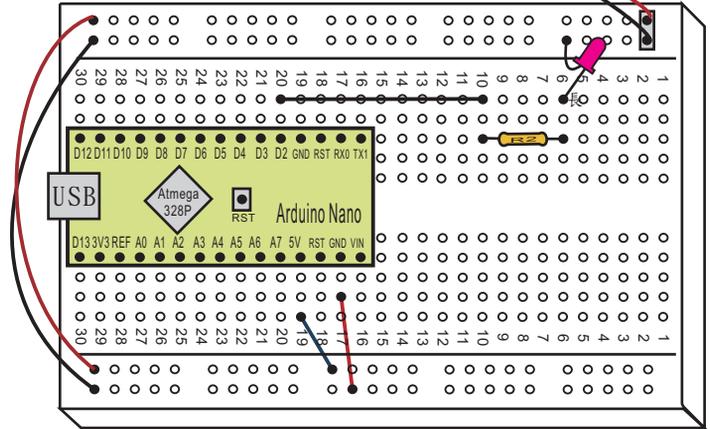
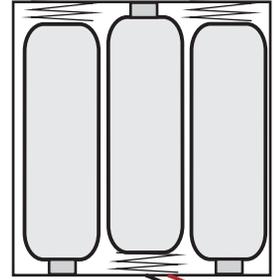


第1個程式 閃爍的燈

```
int i;
void setup() {
  pinMode(2,OUTPUT);}

void loop(){
  for (i = 1; i <= 10; i=i+1)
  { digitalWrite(2,HIGH);
    delay(100);
    digitalWrite(2,LOW);
    delay(100); }
  delay(1000);
}
```

閃爍10次



第2個程式

閃爍的燈(用閃爍10次的函式)

```
void setup() {
  pinMode(2,OUTPUT);
}

void loop() {
  flash(2); 閃爍10次
  delay(1000);
}

void flash(int n) {
  for (int i=1;i<=10;i=i+1)
  { digitalWrite(n,HIGH);
    delay(100);
    digitalWrite(n,LOW);
    delay(100); }
}
```

閃爍函式
閃爍10次

第3個程式

可改變閃爍次數的閃爍函式

```
void setup() {
  pinMode(2,OUTPUT);
}

void loop() {
  flash(2,5); 閃爍5次
  delay(1000);
  flash(2,15); 閃爍15次
  delay(2000);
}

void flash(int n,int m) {
  for (int i=1;i<=m;i=i+1)
  { digitalWrite(n,HIGH);
    delay(100);
    digitalWrite(n,LOW);
    delay(100); }
}
```

閃爍函式
可改次數

沒有return回傳數據的函式，前面要加上void。

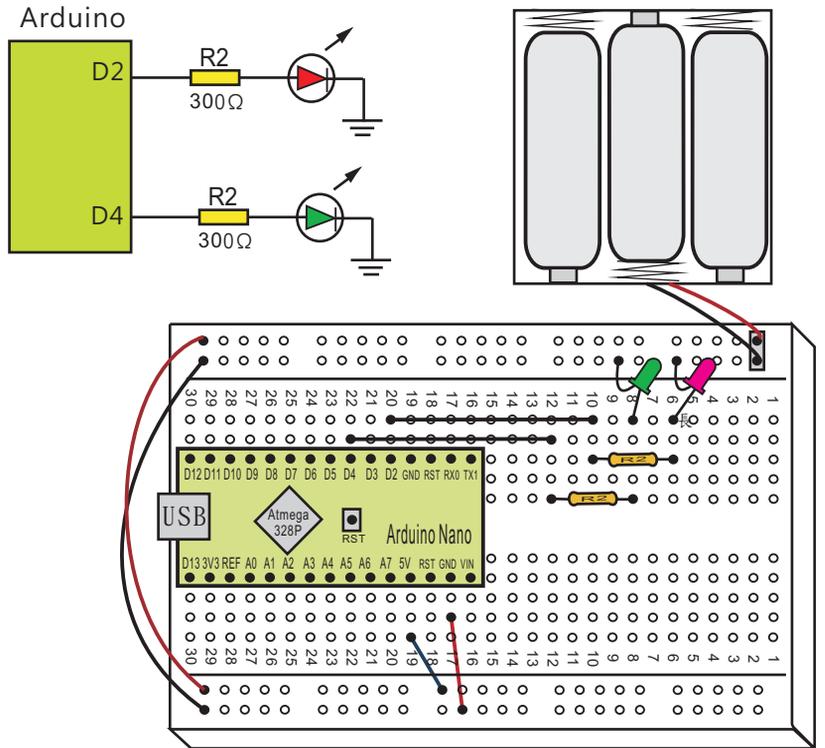
實驗9 紅綠燈 利用函式簡化程式

第1個程式
紅綠燈

紅燈亮1秒
綠燈亮1秒

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(4,OUTPUT);
}

void loop() {
  digitalWrite(2,HIGH); 紅燈
  digitalWrite(4,LOW);
  delay(1000);
  digitalWrite(2,LOW);
  digitalWrite(4,HIGH); 綠燈
  delay(1000);
}
```



寫程式多利用函式，可以讓主程式簡短，結構比較清楚有條理。

第2個程式

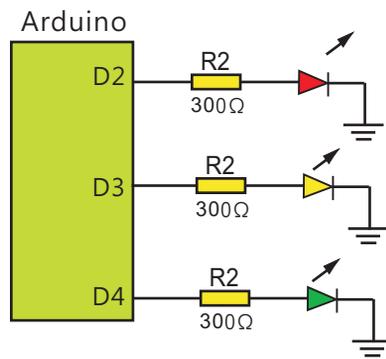
會閃的紅綠燈(用閃爍函式)

紅燈亮1秒
紅燈閃10次
綠燈亮1秒
綠燈閃10次

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(4,OUTPUT);
}

void loop() {
  digitalWrite(2,HIGH); 紅燈
  digitalWrite(4,LOW);
  delay(2000);
  flash(2); 閃紅燈
  digitalWrite(2,LOW);
  digitalWrite(4,HIGH); 綠燈
  delay(2000);
  flash(4); 閃綠燈
}

void flash(int n) {
  for (int i=1;i<=10;i=i+1)
  { digitalWrite(n,HIGH);
    delay(100); 閃爍函式
    digitalWrite(n,LOW);
    delay(100); } }
```



第3個程式 三個燈輪流亮

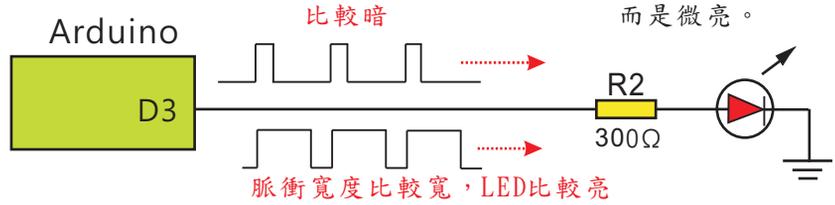
```
int i;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
}

void loop() {
  for (i=2;i<=4;i=i+1)
  { digitalWrite(i,HIGH);
    delay(2000);
    digitalWrite(i,LOW); }
}
```

實驗10 PWM方法控制亮度

analogWrite()

脈衝頻率若很快，就看不到閃爍，而是微亮。



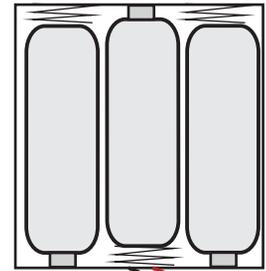
下面兩個程式
哪一個程式的LED比較亮？
為什麼？

第1個程式

```
void setup(){
  pinMode(3,OUTPUT);
}

void loop(){
  digitalWrite(3,HIGH);
  delay(2);
  digitalWrite(3,LOW);
  delay(8);
}
```

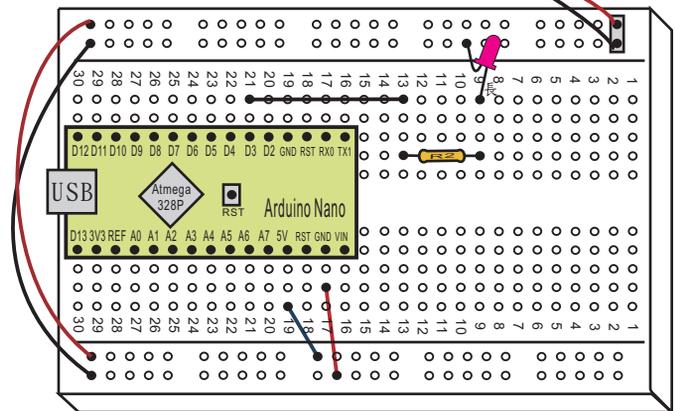
PWM
Pulse Width Modulation
脈衝寬度調變
用脈衝寬度來控制功率。
因為控制電流大小不容易，
控制通電或不通電，比較容易
PWM用通電時間的長短來控制功率。



第2個程式

```
void setup(){
  pinMode(3,OUTPUT);
}

void loop(){
  digitalWrite(3,HIGH);
  delay(8);
  digitalWrite(3,LOW);
  delay(2);
}
```



第3個程式

analogWrite(腳位,脈衝寬度)

analogWrite這個指令
會幫我們用PWM來輸出信號，
所以數位腳可以輸出類比信號，
只有3, 5, 6, 9, 10, 11這六支腳
可以用analogWrite輸出類比信號。
脈衝寬度在0與255之間，
若超過255，256則視同0，
258則視同2。

0就完全不亮，255就全亮。
analogWrite(3, 0) = digitalWrite(3, LOW)
analogWrite(3, 255) = digitalWrite(3, HIGH)

用analogWrite指令
來調控LED亮度。

```
int i;
void setup(){
  pinMode(3,OUTPUT);
}

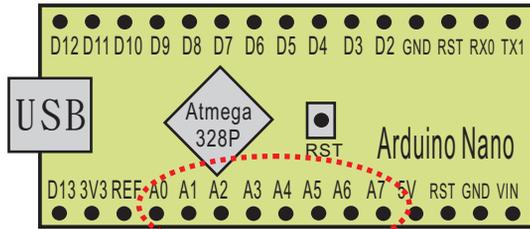
void loop(){
  analogWrite(3, 50); 微亮
  delay(500);
  analogWrite(3, 255); 強亮
  delay(500);
}
```

練習11 類比輸入 analogRead()

數位digital: 只有0和1，在電路上，高電位HIGH代表1，低電位LOW代表0。

類比analog: 除了0和1，還有其他的數值。

物理世界的資訊要用電壓(不是電流)的型態傳給電腦。



- D2...D13 數位信號的輸入或輸出 (用pinMode來設定)
- A0...A7 類比信號的輸入 (預設是輸入，不用設定)

A0到A7這八支腳被預設為類比電壓的輸入，所以可以不必用pinMode宣告，我們用三顆電池做電源，電壓是4.5伏特，類比輸入腳讀入的數值在0與1023之間，如果讀入的值是0，就是0伏特，如果讀入的值是1023，就是4.5伏特，如果讀入的值是512，就是2.25伏特，讀入值每增加1，電壓就增加4.5伏特/1023，4.5伏特/1023=0.0044伏特，讀入的值是n，電壓就是就是4.5伏特*n/1023，也就是0.0044伏特*n

請填這表

讀入值n	0	100	340	512	820	1023
電壓	0V					4.5V

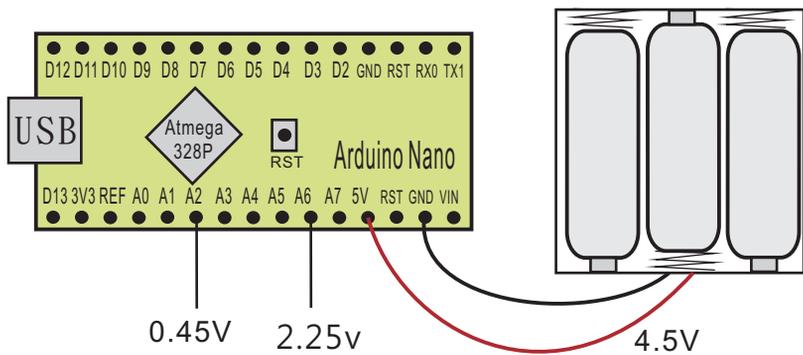
$$\text{電壓} = 4.5v * n / 1023$$

請問v2和v6分別是多少？

```
int v2,v6;
void loop(){
  v2 = analogRead(A2);
  v6 = analogRead(A6);
}
```

$$\text{電壓} = 4.5v * n / 1023$$

$$n = 1023 * \text{電壓} / 4.5$$



答案:

$$v2 = 1023 * 0.45 / 4.5 = 102$$

$$v6 = 1023 * 2.25 / 4.5 = 512$$

實驗12 越黑暗, LED越亮

類比輸入(analogRead)和類比輸出(analogWrite)



第1個程式
越黑暗, LED越亮

```
int aa ;
void setup(){
  pinMode(3,OUTPUT);
}

void loop(){
  aa = analogRead(5);
  analogWrite(3,255-aa/4);
}
```

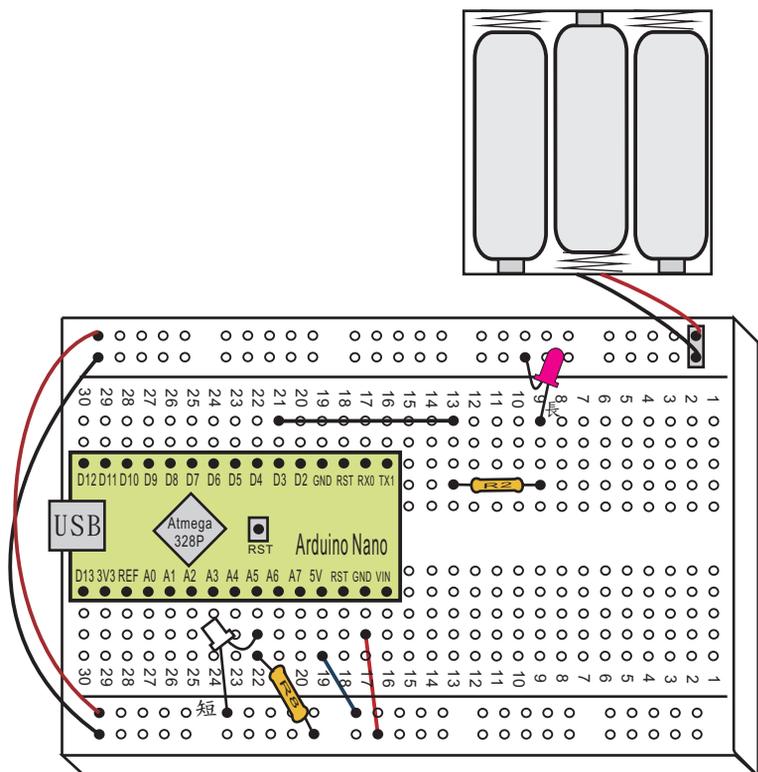
黑暗中, A5電壓會下降, A0到A7這八支腳被預設為類比電壓的輸入, 所以可以不必用pinMode宣告, analogRead讀入的類比電壓在0與1023之間, 0是0伏特, 1023是4.5伏特, 512是2.25伏特, 依此類推。analogWrite類比輸出的值, 則在0與255之間, 只有3, 5, 6, 9, 10, 11這六支腳可以輸出類比電壓。讀入的電壓(0到1023)要再除以4, 因為類比輸出為0到255。黑暗中, 第五點電壓aa會降低, 所以用255減去aa/4 再類比輸出到第3腳。

若麵包板孔接觸不良, 可取出麵包板, 用手把背面金屬條用力壓回去。

第2個程式
越黑暗, LED閃爍越快

```
int aa ;
void setup(){
  pinMode(3,OUTPUT);
}

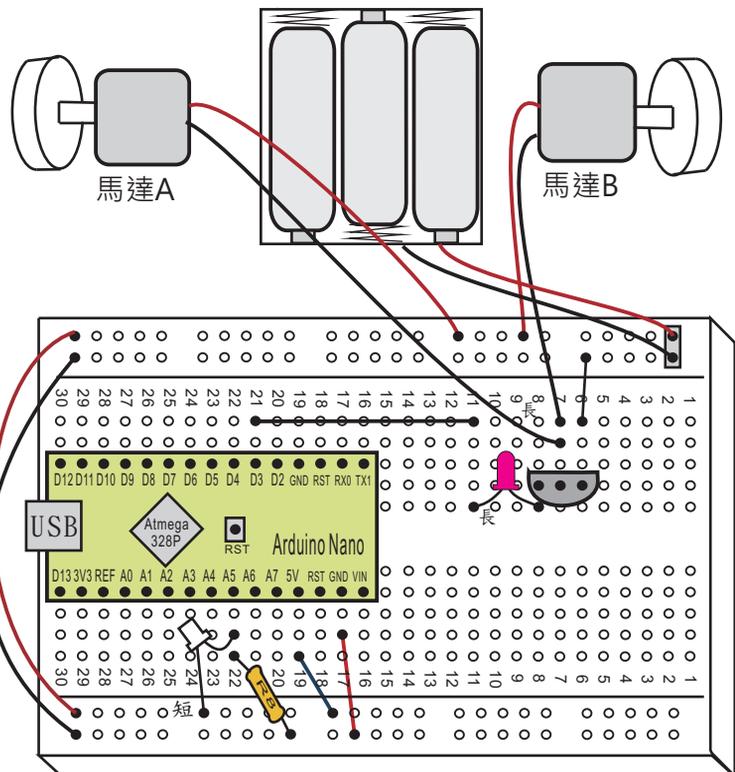
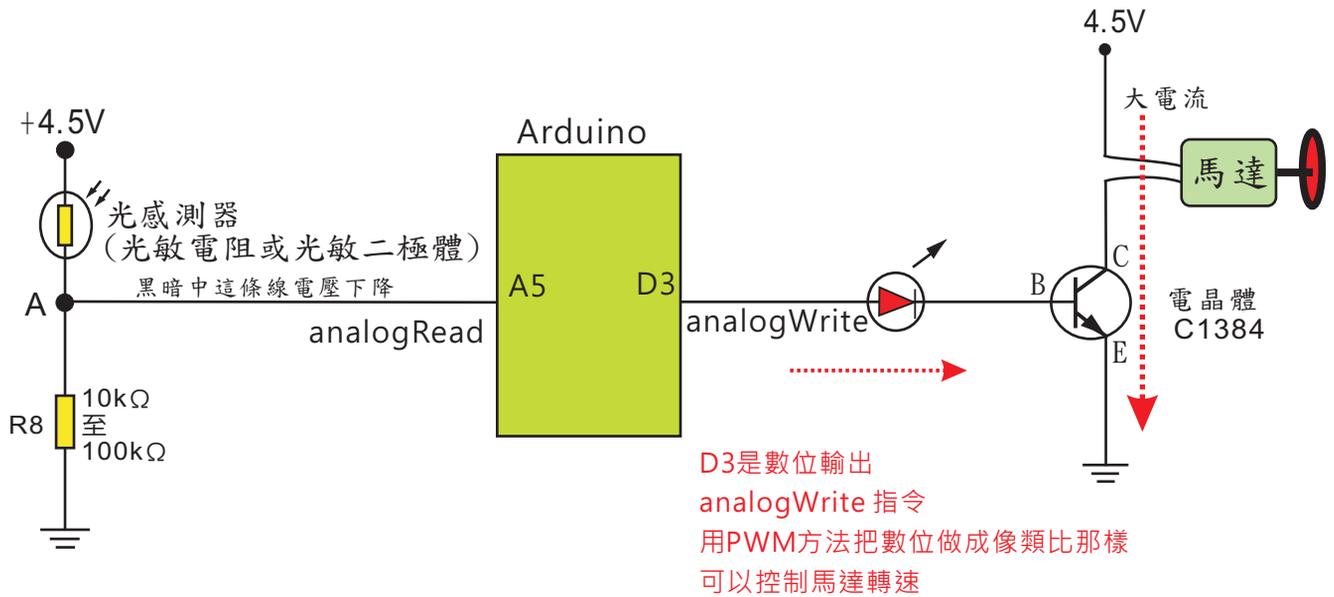
void loop(){
  aa = analogRead(5);
  digitalWrite(3,HIGH);
  delay(aa);
  digitalWrite(3,LOW);
  delay(aa);
}
```



光感測器 注意:光敏二極體是短正長負
但有些光敏二極體是長正短負

實驗13 越黑暗車子動越快

類比輸入(analogRead)和類比輸出(analogWrite)



```
int aa = 0;
void setup(){
  pinMode(3,OUTPUT);
}

void loop(){
  aa = analogRead(5);
  analogWrite(3,255-aa/4);
}
```

注意:光敏二極體是短正長負
但有些光敏二極體是長正短負

實驗14 光強度顯示器

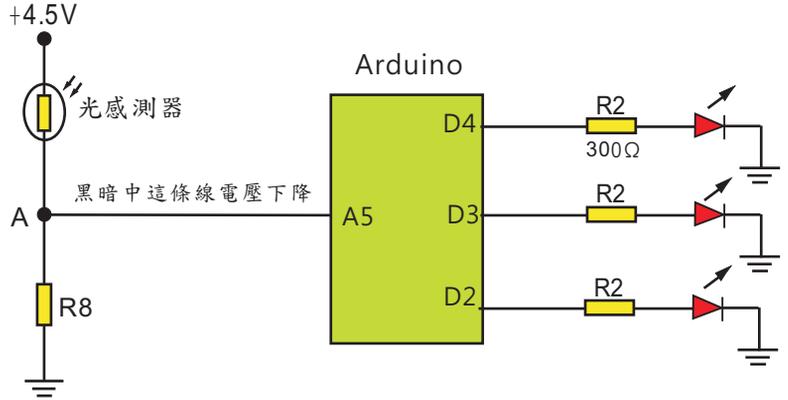
類比輸入(analogRead)

```

void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
}

void loop() {
  int aa = analogRead(A5);
  int bb = map(aa, 0, 1000, 0,7);
  switch (bb) {
    case 0:
      digitalWrite(2,LOW);
      digitalWrite(3,LOW);
      digitalWrite(4,LOW);
      break;
    case 1:
      digitalWrite(2,HIGH);
      digitalWrite(3,LOW);
      digitalWrite(4,LOW);
      break;
    case 2:
      digitalWrite(2,LOW);
      digitalWrite(3,HIGH);
      digitalWrite(4,LOW);
      break;
    case 3:
      digitalWrite(2,HIGH);
      digitalWrite(3,HIGH);
      digitalWrite(4,LOW);
      break;
    case 4:
      digitalWrite(2,LOW);
      digitalWrite(3,LOW);
      digitalWrite(4,HIGH);
      break;
    case 5:
      digitalWrite(2,HIGH);
      digitalWrite(3,LOW);
      digitalWrite(4,HIGH);
      break;
    case 6:
      digitalWrite(2,LOW);
      digitalWrite(3,HIGH);
      digitalWrite(4,HIGH);
      break;
    case 7:
      digitalWrite(2,HIGH);
      digitalWrite(3,HIGH);
      digitalWrite(4,HIGH);
      break;
  }
}

```



由A5讀入光線的強弱，
再利用map函數，
把0到1023
分成0,1,2..7 八個等級，
1000以上是7
用bb來代表
若bb=0 就顯示000
若bb=1 就顯示001
若bb=2 就顯示010

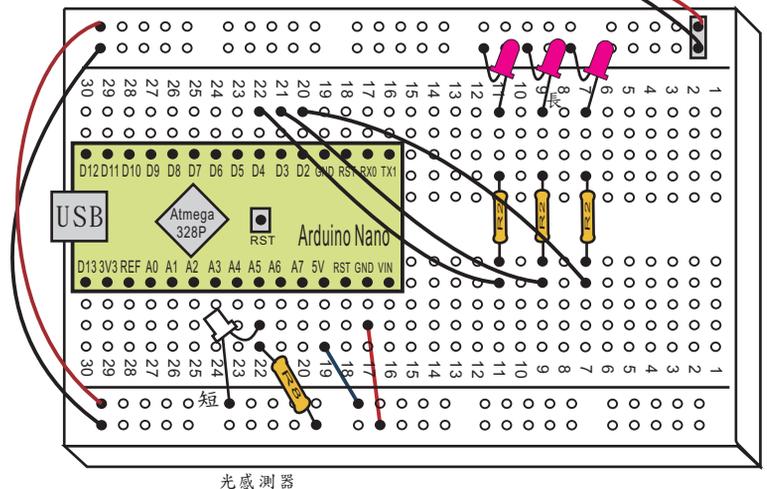
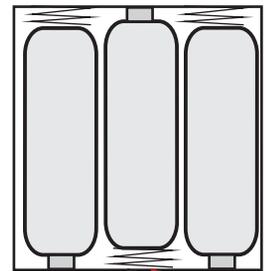
讀入A5電壓
把A5電壓分成8個等級
switch (電壓等級)

```

{ case 0: 000
  case 1: 001
  case 2: 010
  case 3: 011
  case 4: 100
  case 5: 101
  case 6: 110
  case 7: 111
}

```

條件判斷與選擇
switch(n)
case
case
case



實驗15 人影偵測警報器

類比輸入(analogRead)

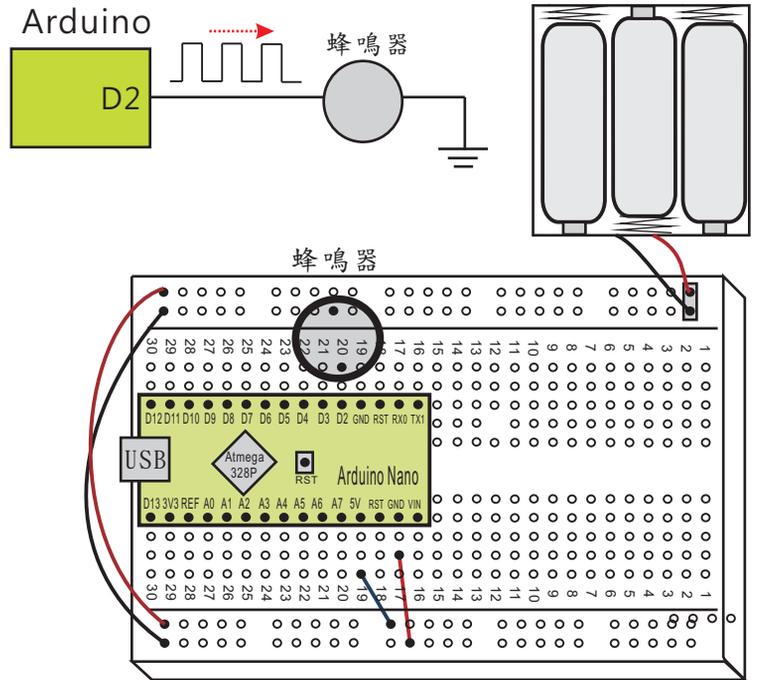
第1個程式

警報聲

```
void setup() {
  pinMode(2,OUTPUT); }

void loop() {
  for (float i=0; i<=1000;i=i+10)
  { digitalWrite(2,HIGH);
    delayMicroseconds(i);
    digitalWrite(2,LOW);
    delayMicroseconds(i); }
}
```

改變參數再聽聽看!



第2個程式

光影晃動就發出警報

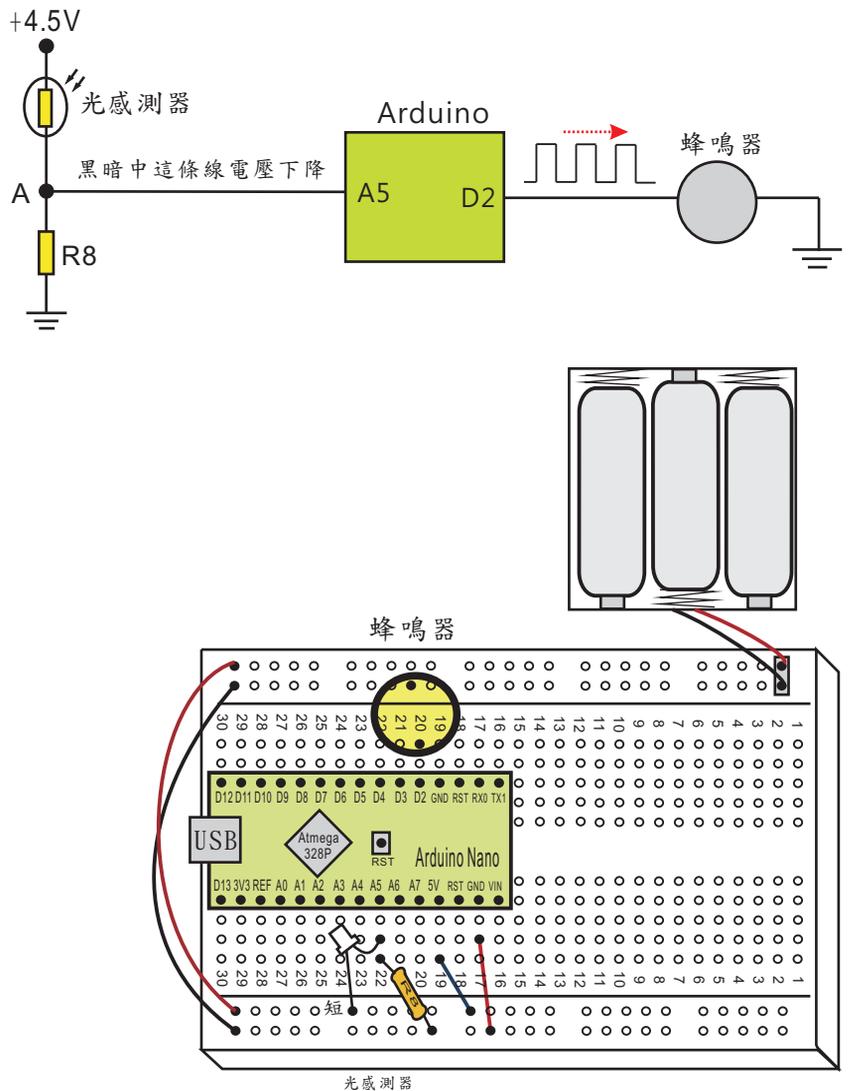
讀取A5值；
隔0.1秒再讀一次
如果(兩者相差10以上)
{ 警報聲三次 }

```
void setup() {
  pinMode(2,OUTPUT); }

void loop()
{ v1=analogRead(A5);
  delay(100);
  v2=analogRead(A5);
  if (abs(v2-v1) > 10)
    { buzz();buzz();buzz(); }
}
```

```
void buzz(){
  for (float i=0; i<=1000;i=i+10)
  { digitalWrite(2,HIGH);
    delayMicroseconds(i);
    digitalWrite(2,LOW);
    delayMicroseconds(i);}}
```

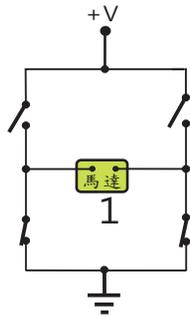
警報聲函式



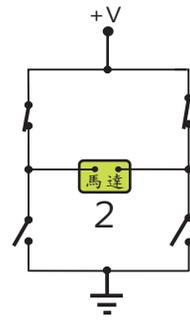
用H型電路來驅動馬達

馬達的正反轉控制(H型)

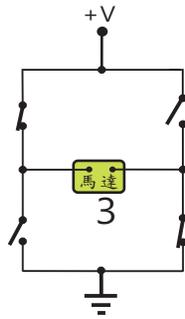
哪幾個馬達會轉動?
轉動方式有何不同?



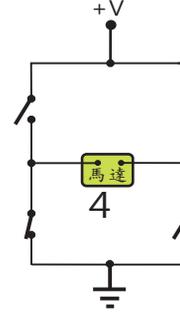
不轉



不轉



轉動

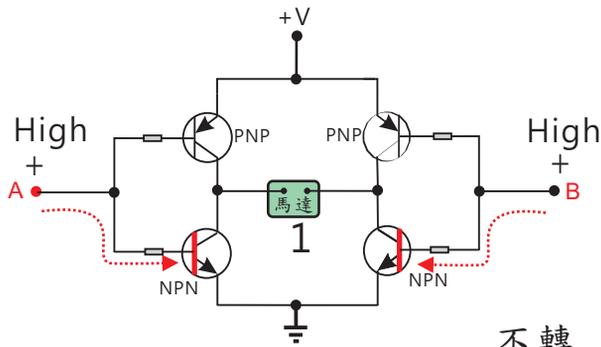


轉動

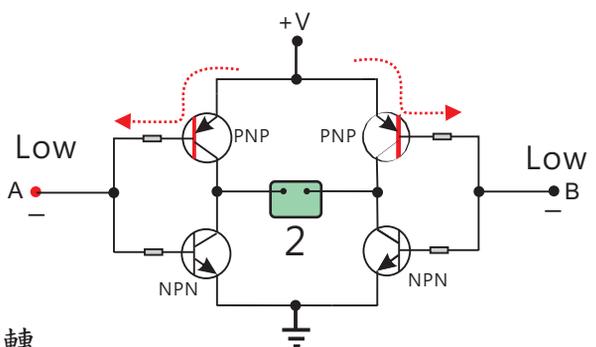
用四個電晶體取代四個開關，
PNP電晶體是Low電位啟動，NPN電晶體是High電位啟動。

馬達的正反轉控制(H型)

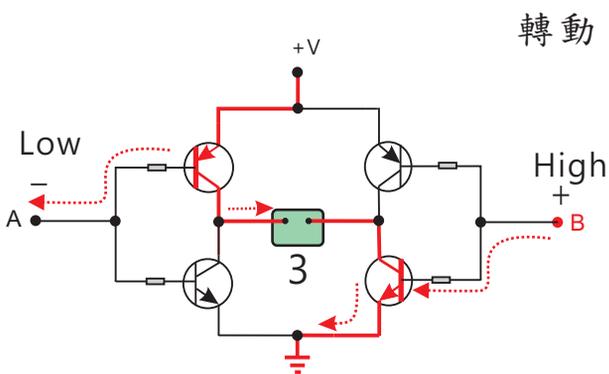
哪幾個馬達會轉動?
轉動方式有何不同?



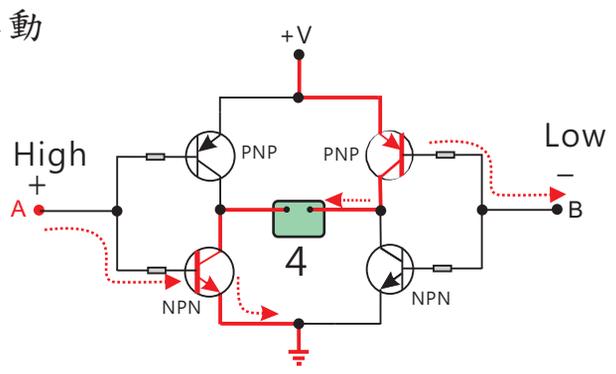
不轉



不轉



轉動



轉動

H型馬達驅動IC-L9110

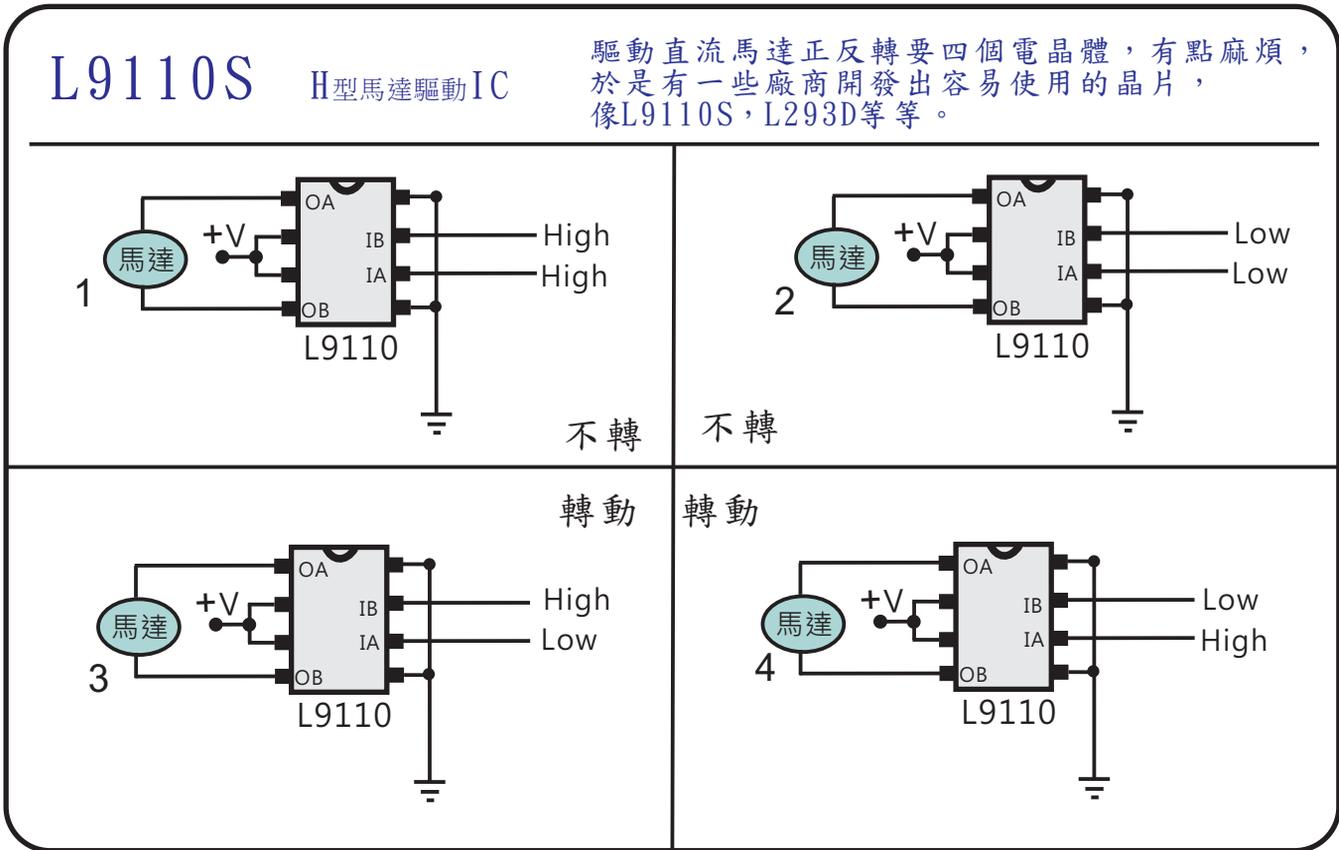
兩個L9110



哪幾個馬達會轉動？ 轉動方式有何不同？

L9110S H型馬達驅動IC

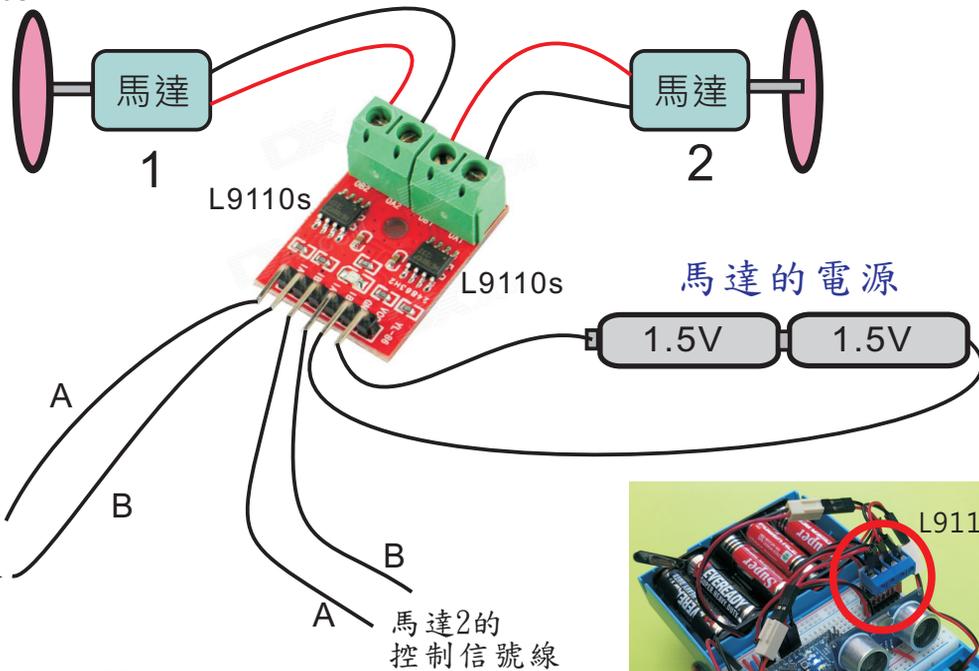
驅動直流馬達正反轉要四個電晶體，有點麻煩，於是有一些廠商開發出容易使用的晶片，像L9110S，L293D等等。



這電路板有兩個L9110s



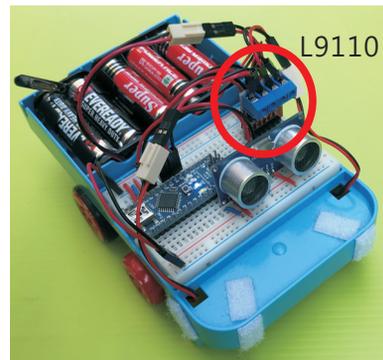
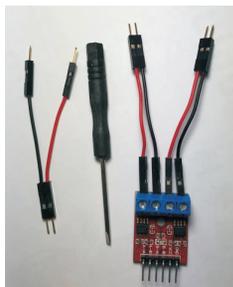
A	B	馬達
High	High	停
Low	Low	停
Low	High	正轉
High	Low	反轉



馬達1的控制信號線

馬達2的控制信號線

請用小螺絲起子把兩條端子線鎖上去



這一塊驅動馬達的電路板有兩個L9110S晶片，可驅動兩個直流馬達，可取代兩顆繼電器，網路上買得到，幾十元一片。

這台Arduino 程式控制的科學魔法車就是採用L9110來驅動馬達

實驗16 會跳舞的魔法車

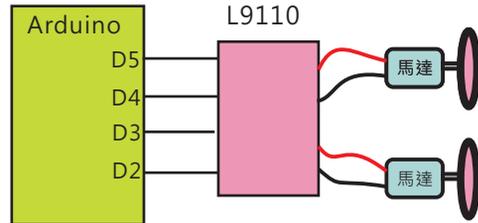
第1個程式

一次只轉一個輪子，
左輪先前進 0.5秒，
然後換右輪前進0.5秒，
然後換左輪前進0.5秒，
交替前進

馬達用久了會變慢，
若一快一慢，車子走不直，
可更換馬達。

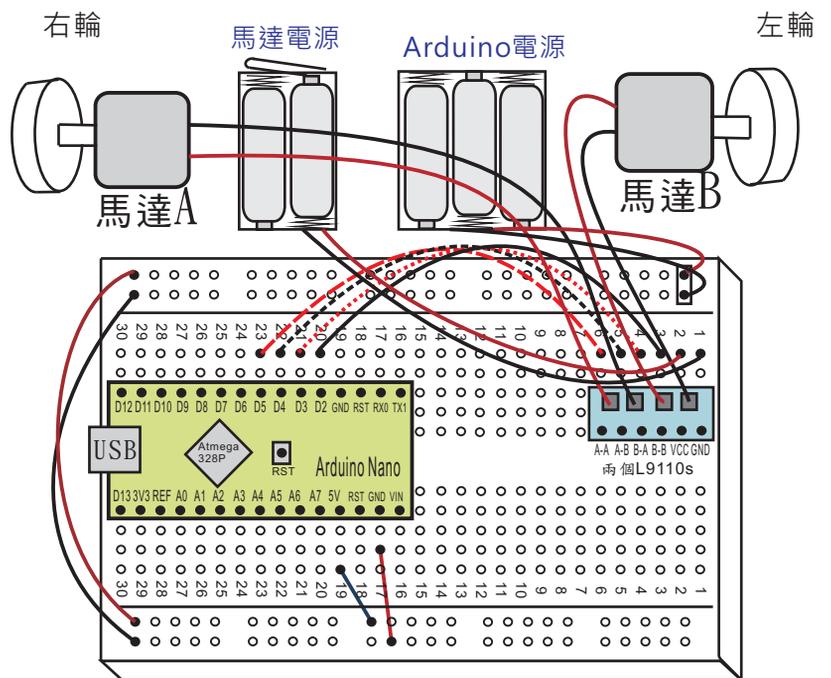
```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}
```

```
void loop() {
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH); // A輪前進
  digitalWrite(5,LOW);
  delay(500);
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW); // B輪前進
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  delay(500);
}
```



驅動馬達的電路板

在這裡我們用兩顆L9110驅動馬達，
每一顆L9110可以驅動一顆馬達，
做正轉，反轉或停止的動作。
馬達的電力由兩顆電池3V提供。



第2個程式

```
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}
```

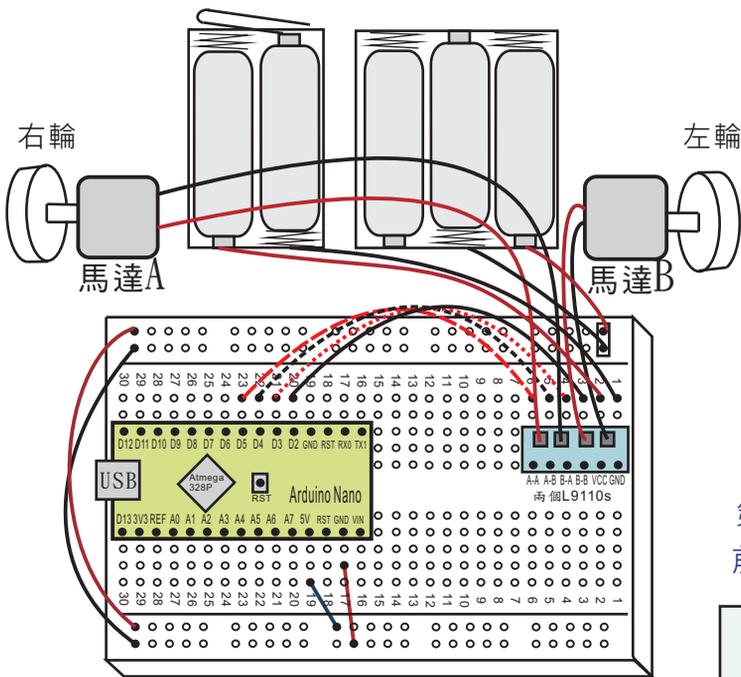
這個程式
車子會如何動作?

```
void loop() {
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
  delay(1000);
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
  delay(1000);
}
```

右 左
↓
前進

5	4	馬達A	3	2	馬達B
Low	Low	停	Low	Low	停
Low	High	前進	Low	High	前進
High	Low	後退	High	Low	後退

實驗17 前進、後退的函式



右			左		
5	4	馬達A	3	2	馬達B
Low	Low	停	Low	Low	停
Low	High	前進	Low	High	前進
High	Low	後退	High	Low	後退

若麵包板孔接觸不良，可取出麵包板，用手把背面金屬條用力壓回去。

第2個程式
前進、後退、前進、後退...(採用函式)

第1個程式
前進、後退、前進、後退...



```

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  delay(1000);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  delay(1000);
}
    
```

前進

後退

```

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  forward(); 前進
  delay(1000); 延時1秒
  backward(); 後退
  delay(1000); 延時1秒
}

void forward() {
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
}

void backward() {
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
}
    
```

主程式

前進函式

後退函式

實驗18 用手的影子控制車子轉彎

用手的影子讓車子轉彎

```

int a;
void setup()
{ pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT); }
void loop() {
  a=analogRead(A5);
  if (a <= 700)
    {leftward();}
  else
    {forward();} }

void forward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);}

void leftward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);}
    
```

在暗的地方
700要改小
否則手還沒遮
車子就轉彎。

前進

左轉

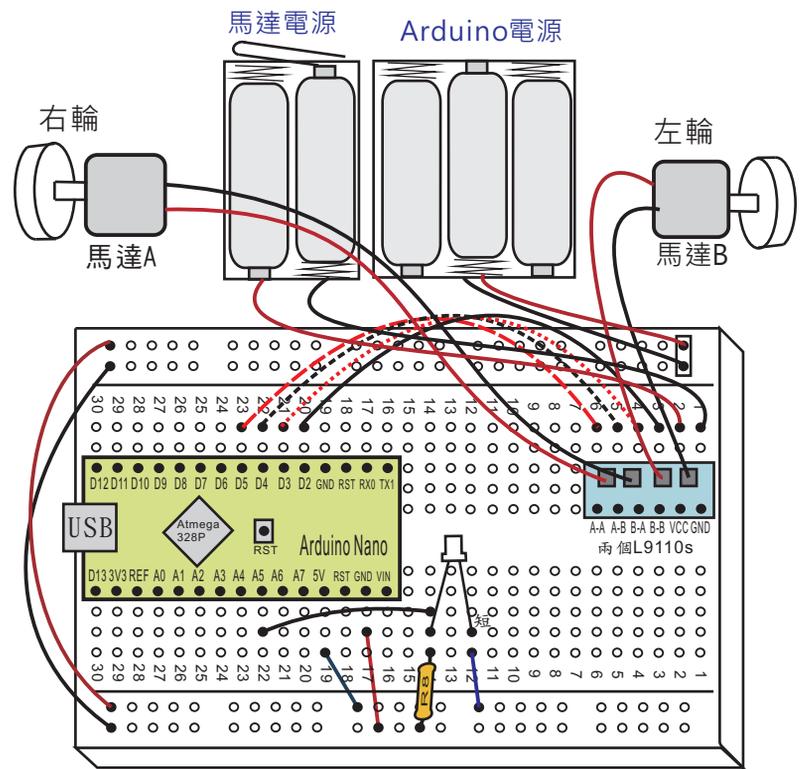
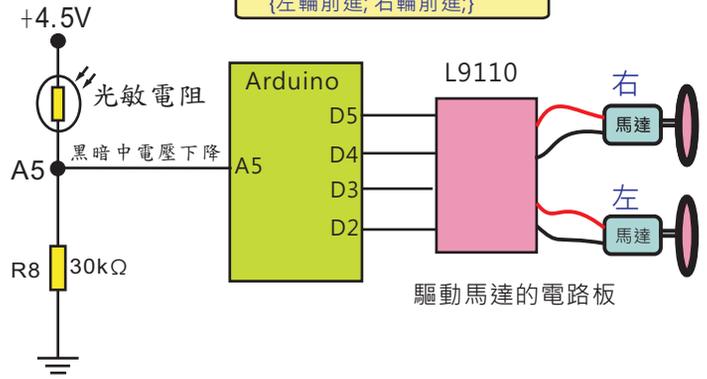
也可以追手電筒光源
(要改700)

寫程式要多利用函式
讓主程式小一點

在這裡我們用兩顆L9110驅動馬達，
每一顆L9110可以驅動一顆馬達，
做正轉，反轉或停止的動作。
馬達的電力由兩顆電池3V提供。

```

if(analogRead(A5)<=700)
{左輪後退; 右輪前進;}
else
{左輪前進; 右輪前進;}
    
```



R8 30KΩ 橙黑橙金

注意:光敏二極體是短正長負
但有些光敏二極體是長正短負



右		左	
5	4	3	2
馬達A	馬達B	馬達A	馬達B
Low	Low	Low	Low
High	High	Low	High
High	Low	High	Low
停	停	前進	後退
前進	後退	後退	前進

實驗19 跟著光線旋轉的魔法車

車頭自動朝向手電筒光線

```

void setup()
{ pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()
{
  while (analogRead(A5)-analogRead(A7)>10)
    { rightward(); } 右轉
  while (analogRead(A7)-analogRead(A5)>10)
    { leftward(); } 左轉
  stopward(); 停止
}

void leftward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
}

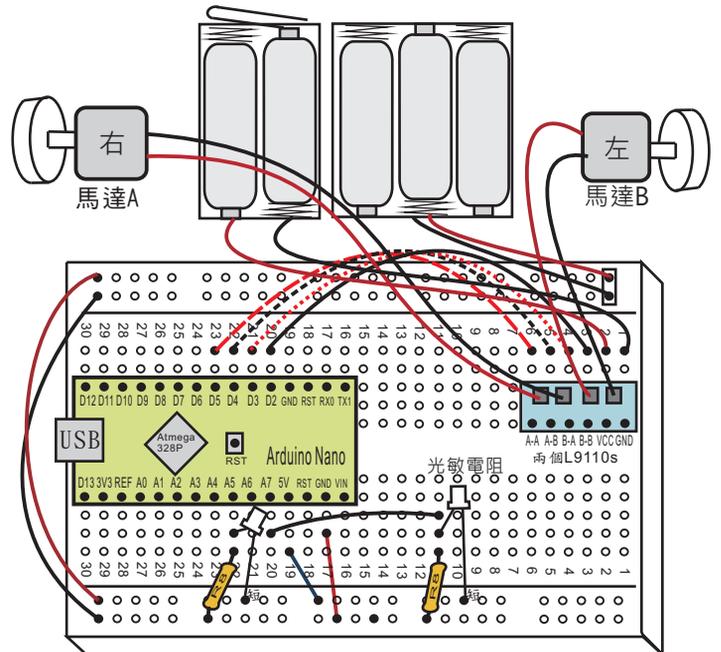
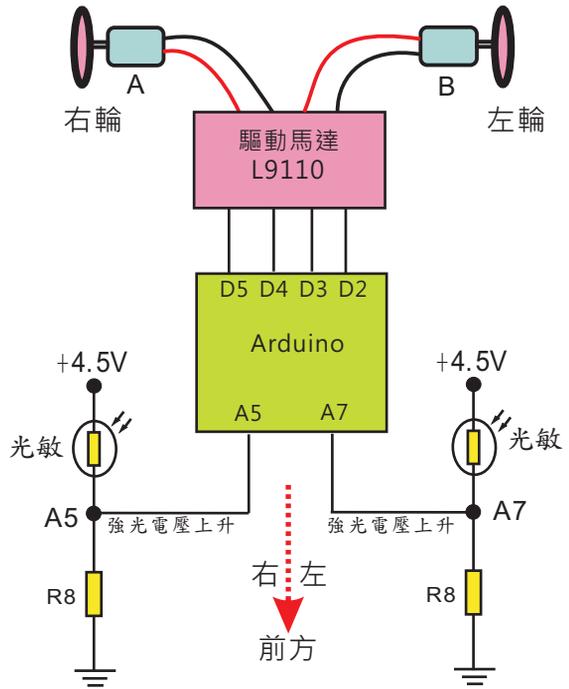
void rightward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
}

void stopward(){
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
}
    
```

```

while (A5 -A7>10)
{ 右轉 }
while (A7 -A5>10)
{ 左轉 }
停止
    
```

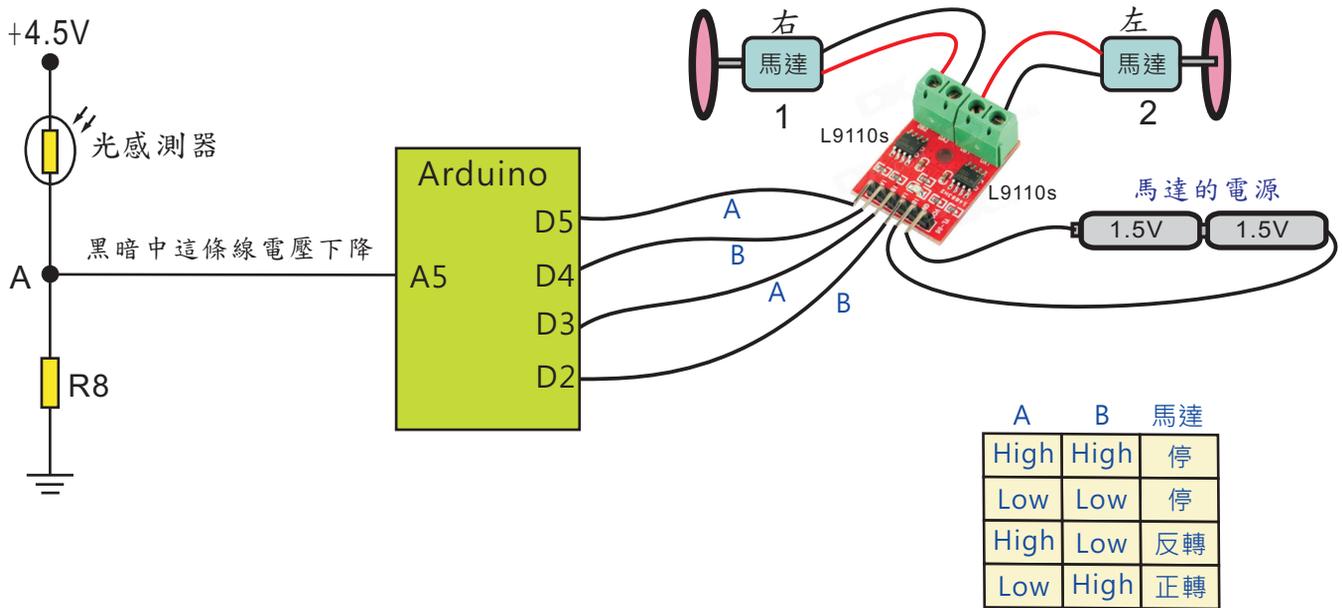
若轉動不穩定
10改2、5或20
再試試看



注意:光敏二極體是短正長負 但有些光敏二極體是長正短負 R8 30KΩ 橙黑橙金

實驗20 越黑暗,車子跑越快

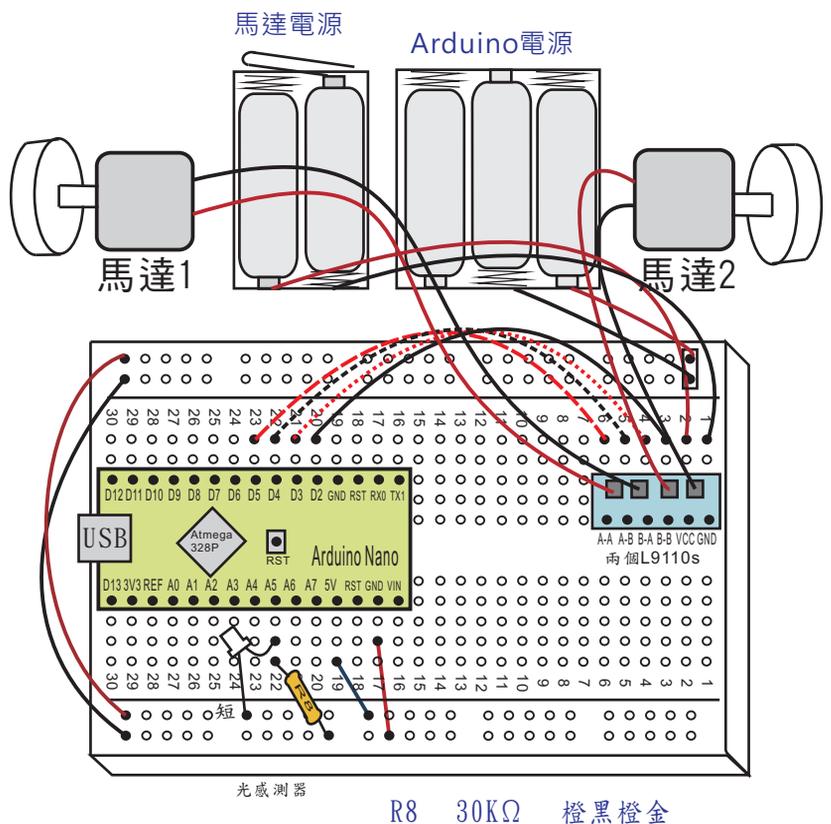
類比輸入(analogRead)和類比輸出(analogWrite)



影子越黑, A5電壓越低, 車速越快

```
int speed=0;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop() {
  speed=(analogRead(5)+200)/5;
  digitalWrite(2,HIGH);
  analogWrite(3,speed);
  digitalWrite(4,HIGH);
  analogWrite(5,speed);
}
```



注意:光敏二極體是短正長負
但有些光敏二極體是長正短負

實驗21 紅外線閃障、循跡、不落桌下

```

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
}

void loop()
{
  if (digitalRead(6) == LOW)
  {
    digitalWrite(5, HIGH); // 左輪反轉
    digitalWrite(4, LOW);
  }
  else
  {
    digitalWrite(5, LOW); // 左輪正轉
    digitalWrite(4, HIGH);
  }

  if (digitalRead(7) == LOW)
  {
    digitalWrite(3, HIGH); // 右輪反轉
    digitalWrite(2, LOW);
  }
  else
  {
    digitalWrite(3, LOW); // 右輪正轉
    digitalWrite(2, HIGH);
  }
}

```

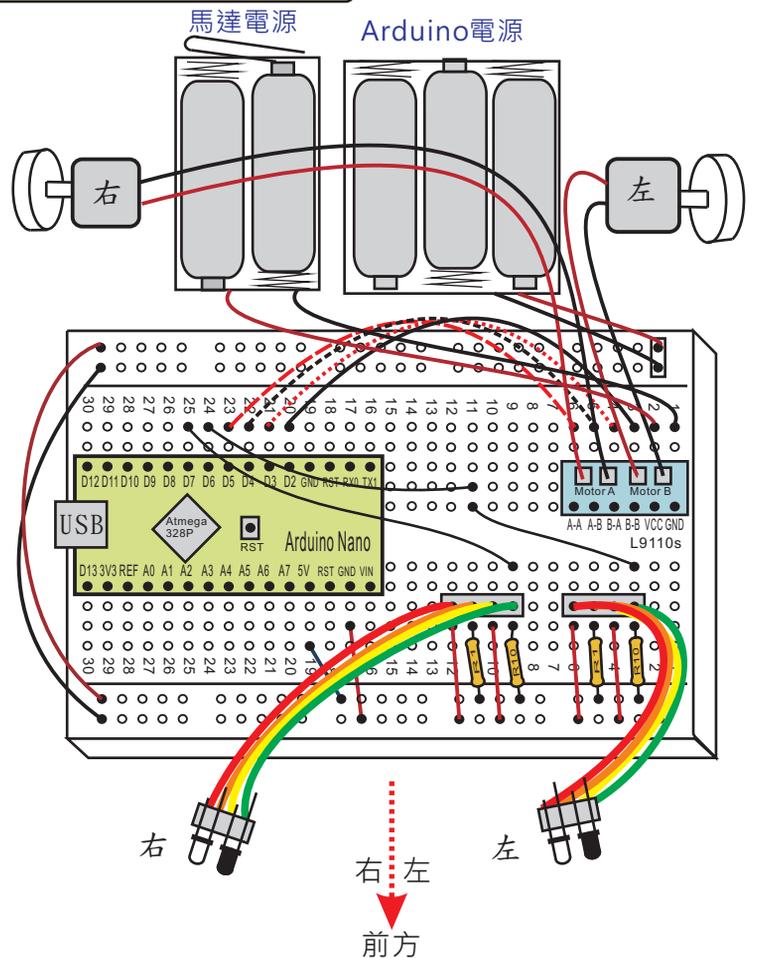
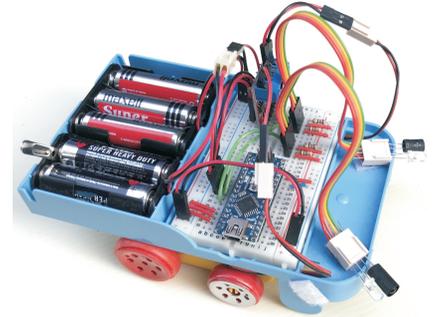
紅外線偵測前方，閃避障礙
左前方有物體，右輪就反轉
右前方有物體，左輪就反轉

不需改程式，
把馬達紅黑線交換，
再調整紅外線位置，
就可循黑線軌跡、
不落下桌子。

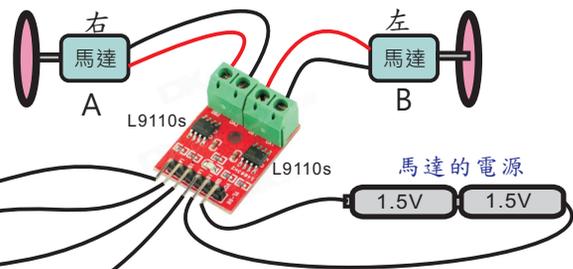
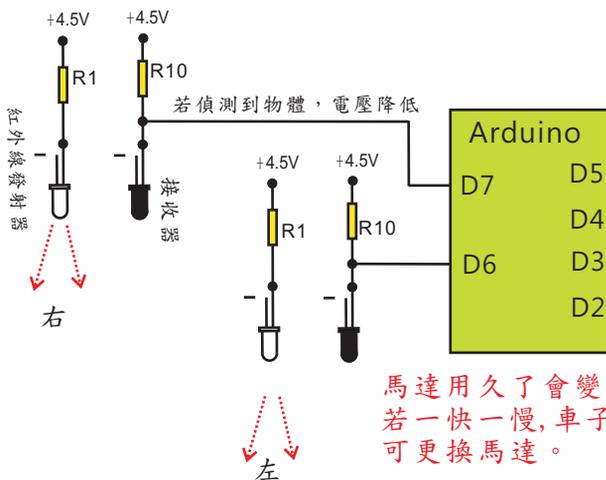
程式架構

如果D6=低
D5高，D4低(左輪後退)
否則
D5低，D4高(左輪前進)

如果D7=低
D3高，D2低(右輪後退)
否則
D3低，D2高(右輪前進)



R1 150Ω 棕綠棕金
R10 6.2MΩ 藍紅綠金



右			左		
5	4	馬達A	3	2	馬達B
Low	Low	停	Low	Low	停
Low	High	前進	Low	High	前進
High	Low	後退	High	Low	後退

馬達用久了會變慢，
若一快一慢，車子走不直，
可更換馬達。

實驗22 紅外線閃避障礙(死巷子自動退出)

(用陶瓷電容消除雜訊干擾)

```
float time1=0;
float time2=100;
float totaltime=100;
```

```
void setup()
{ pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT); }
```

```
void loop()
{
  if (digitalRead(6) == LOW)
  { digitalWrite(4, LOW);      反轉
    digitalWrite(5, HIGH);
    time1=millis()/1000; }
  else
  { digitalWrite(4, HIGH);    正轉
    digitalWrite(5, LOW); }
  if (digitalRead(7) == LOW)
  { digitalWrite(2, LOW);    反轉
    digitalWrite(3, HIGH);
    time2=millis()/1000; }
  else
  { digitalWrite(2, HIGH);   正轉
    digitalWrite(3, LOW); }
```

```
totaltime=time2-time1;
if (abs(totaltime)<=0.5){
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  delay(800);
  time1=0;
  time2=100;
}
```

兩組紅外線偵測到障礙物的時間若小於0.5秒，車子就後退再迴轉出來。

如果D6=低
D5高，D4低(左輪後退)
time1=時間

否則
D5低，D4高(左輪前進)

如果D7=低
D3高，D2低(右輪後退)
time2=時間

否則
D3低，D2高(右輪前進)

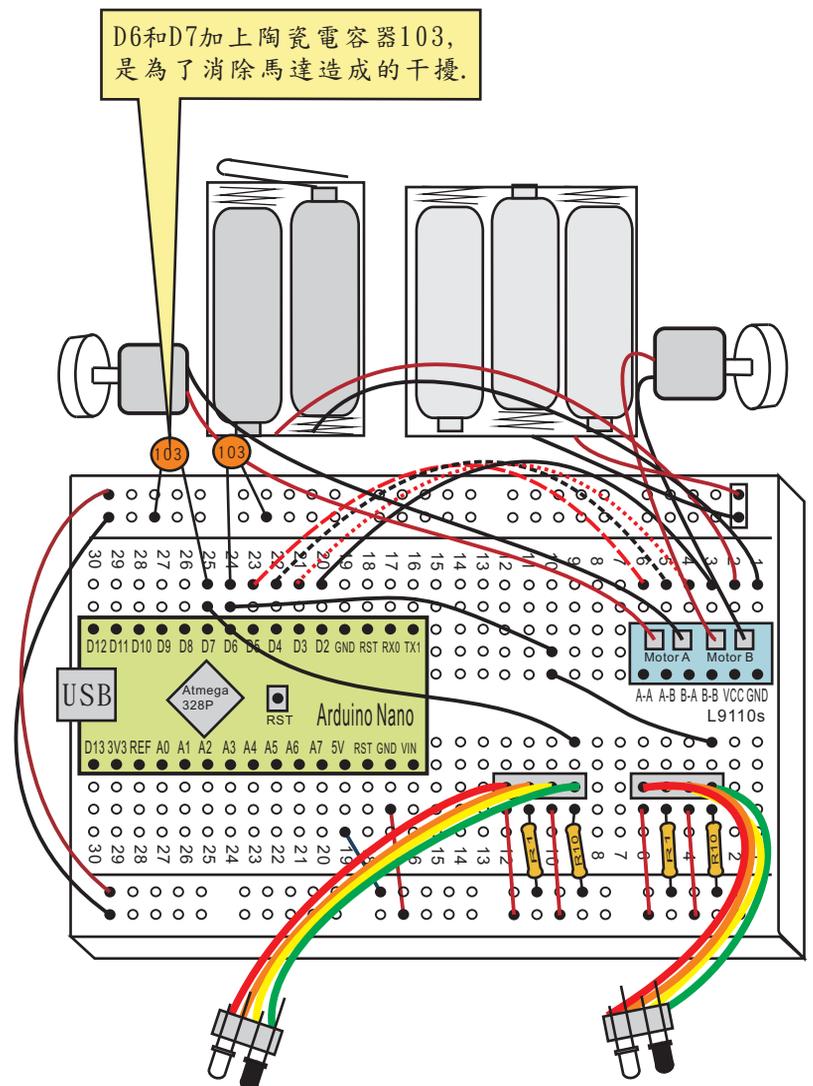
如果 time1-time1 < 0.5秒
後退0.5秒
轉彎0.8秒

自動閃避前方障礙物的魔法車走到死巷子，會自動後退出來(用陶瓷電容消除雜訊干擾)

紅外線偵測前方，閃避障礙左前方有物體，右輪就反轉右前方有物體，左輪就反轉

兩組紅外線偵測到障礙物的時間若小於0.5秒，車子就後退再迴轉出來。D6和D7加上陶瓷電容器103，是為了消除馬達的干擾。

若麵包板孔接觸不良，可取出麵包板，用手把背面金屬條用力壓回去。



實驗23 紅外線閃避障礙(死巷子自動退出)

(用軟體消除雜訊干擾)

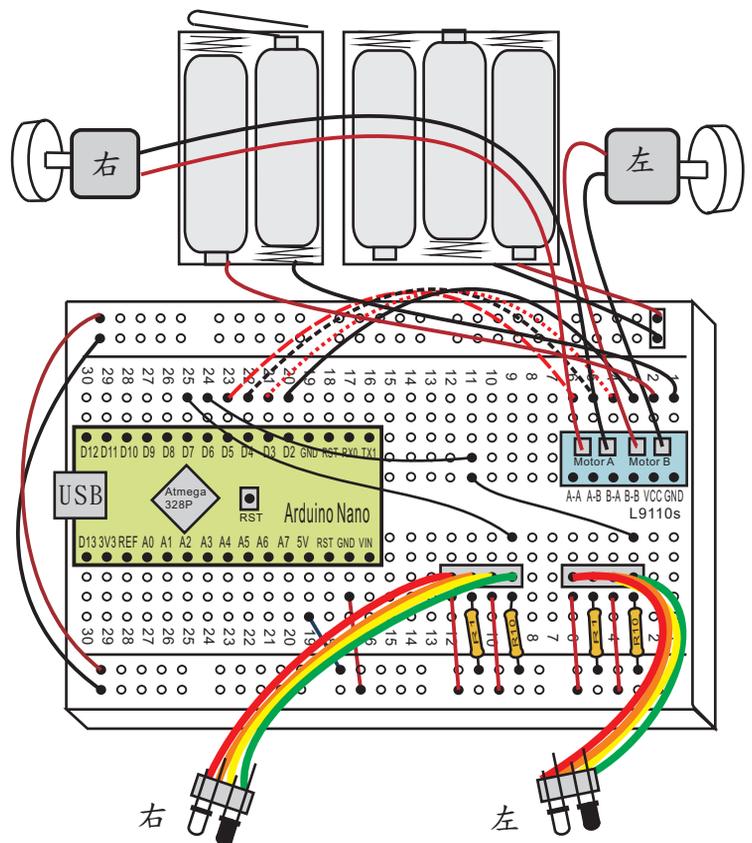
```

float time1=0;
float time2=100;
float totaltime=100;
void setup()
{ pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT); }
void loop()
{ boolean a,b;
  a=digitalRead(6);
  delay(10); 連續讀兩次信號
  b=digitalRead(6);
  if (a==LOW && b==LOW)
  { digitalWrite(4, LOW); D6隔0.01秒再讀一次,
    digitalWrite(5, HIGH); 兩次都有紅外線信號,
    time1=millis()/1000; } 才認可,
  else 以避免誤判雜訊.
  { digitalWrite(4, HIGH);
    digitalWrite(5, LOW); }
  a=digitalRead(7);
  delay(10); 連續讀兩次信號
  b=digitalRead(7);
  if (a==LOW && b==LOW)
  { digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    time2=millis()/1000; }
  else
  { digitalWrite(2, HIGH);
    digitalWrite(3, LOW); }
  totaltime=time2-time1;
  if (abs(totaltime)<=0.5)
  { digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW); 兩組紅外線偵測
    digitalWrite(5, HIGH); 到障礙物的時間
    delay(500); 若小於0.5秒,
    digitalWrite(2, LOW); 車子就後退
    digitalWrite(3, HIGH); 再迴轉出來.
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    delay(800);
    time1=0;
    time2=100; } }
  
```

如果D6=低, 0.01秒後還是低
D5高, D4低(左輪後退)
time1=時間
否則
D5低, D4高(左輪前進)

如果D7=低, 0.01秒後還是低
D3高, D2低(右輪後退)
time2=時間
否則
D3低, D2高(右輪前進)

如果 abs(time1-time2)<0.5秒
後退0.5秒
轉彎0.8秒



實驗24 紅外線閃避障礙(黑暗停車)

```

while (digitalRead(8) == LOW) { 停車 }
if (digitalRead(7) == LOW)
  { 左轉 }
else
  { 直行 }
if (digitalRead(6) == LOW)
  { 右轉 }
else
  { 直行 }
    
```

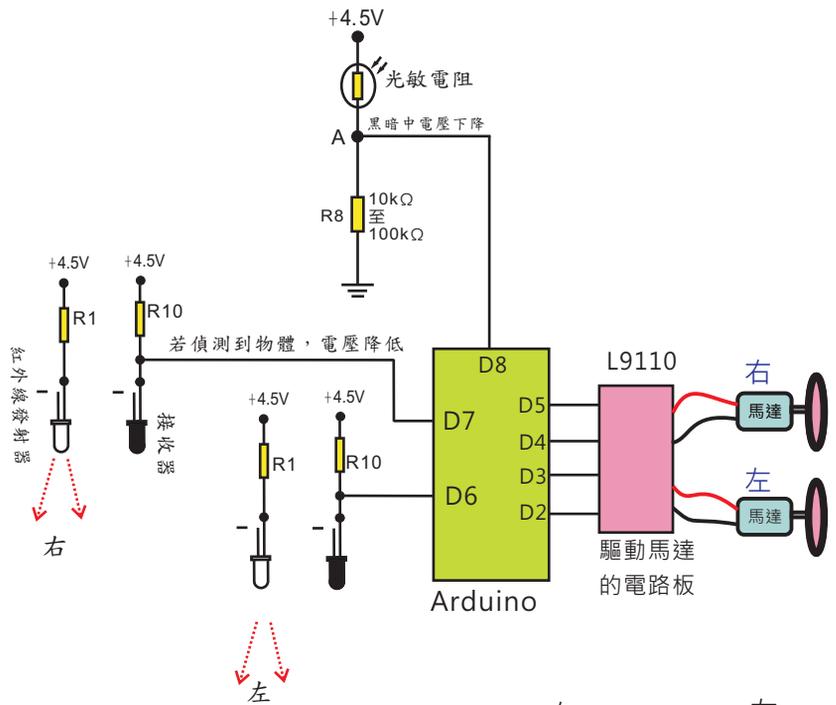
```

void setup()
{ pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT); }

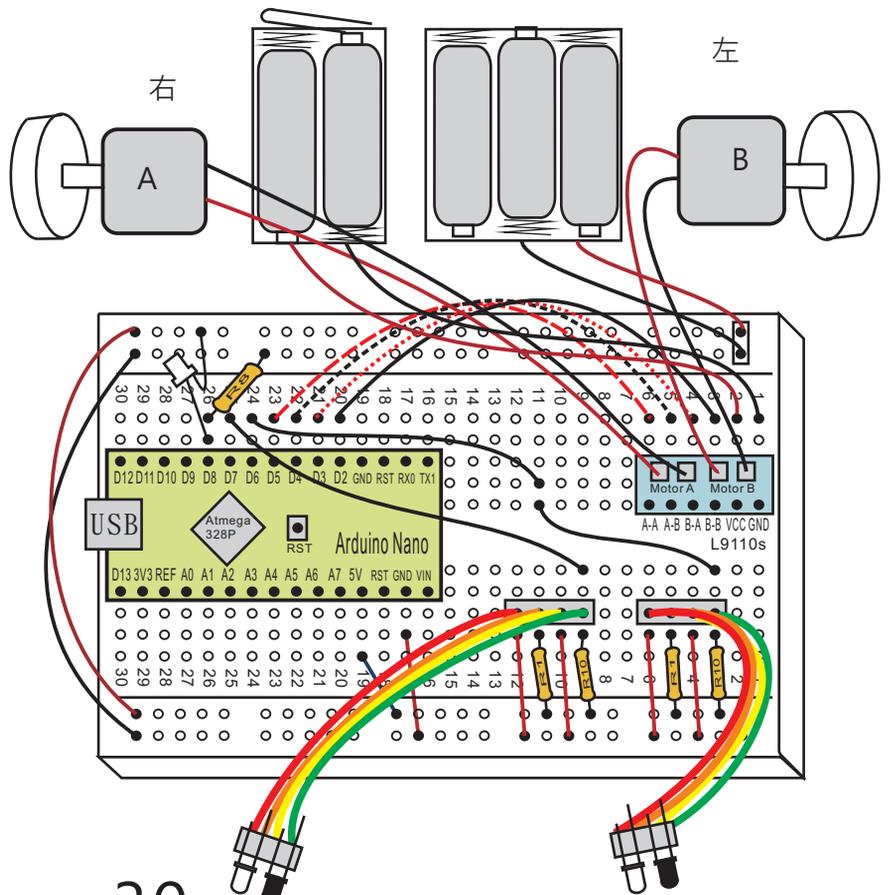
void loop() {
  while (digitalRead(8) == LOW) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW); 停
    digitalWrite(4, LOW);
    digitalWrite(5, LOW); }

  if (digitalRead(7) == LOW)
    { digitalWrite(2, LOW);
      digitalWrite(3, HIGH); } 左退
  else
    { digitalWrite(2, HIGH);
      digitalWrite(3, LOW); } 左進

  if (digitalRead(6) == LOW)
    { digitalWrite(4, LOW);
      digitalWrite(5, HIGH); } 右退
  else
    { digitalWrite(4, HIGH);
      digitalWrite(5, LOW); } 右進
}
    
```



右			左		
5	4	馬達A	3	2	馬達B
Low	Low	停	Low	Low	停
Low	High	前進	Low	High	前進
High	Low	後退	High	Low	後退



實驗25 紅外線閃避障礙(死巷子自動退出)

(黑暗停車)(用陶瓷電容消除雜訊干擾)

```
float time1=0;
float time2=100;
float totaltime=100;

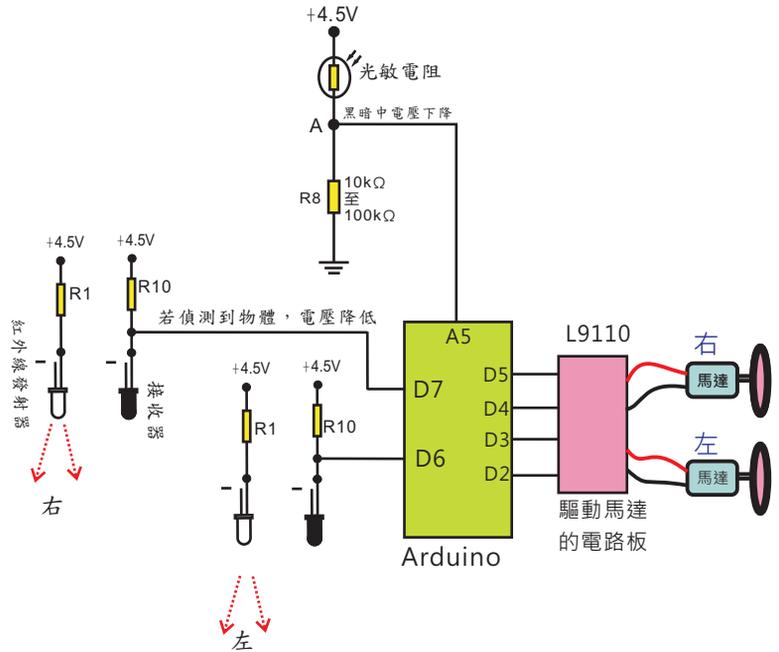
void setup()
{pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, INPUT);
pinMode(7, INPUT); }

void loop()
{ while (analogRead(A5) <= 300)
{digitalWrite(2, LOW);
digitalWrite(3, LOW); 黑暗就停車
digitalWrite(4, LOW);
digitalWrite(5, LOW); }
```

```
if (digitalRead(6) == LOW)
{ digitalWrite(4, LOW); 紅外線B偵測到
digitalWrite(5, HIGH); A輪後退
time1=millis()/1000; }
else
{ digitalWrite(4, HIGH);
digitalWrite(5, LOW); }
```

```
if (digitalRead(7) == LOW)
{ digitalWrite(2, LOW); 紅外線A偵測到
digitalWrite(3, HIGH); B輪後退
time2=millis()/1000; }
else
{ digitalWrite(2, HIGH);
digitalWrite(3, LOW); }
```

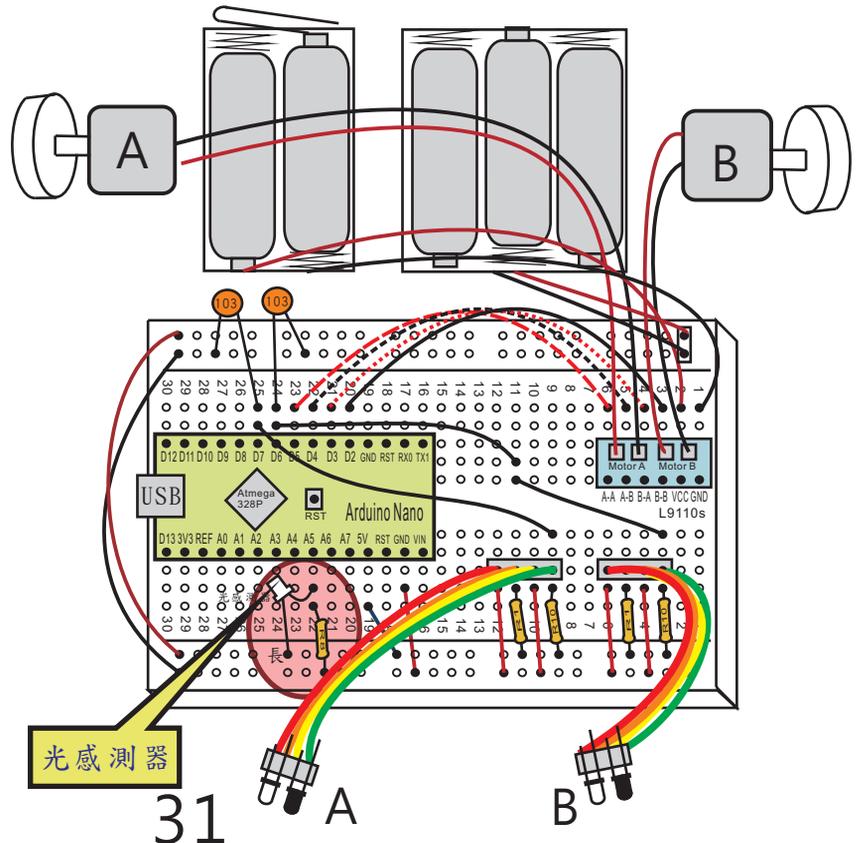
```
totaltime=time2-time1; 兩紅外線幾乎
if (abs(totaltime) <= 0.5){ 同時偵測到
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(4, LOW); 後退0.5秒
digitalWrite(5, HIGH);
delay(500);
digitalWrite(2, LOW);
digitalWrite(3, HIGH); 轉彎0.8秒
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
delay(800);
time1=0;
time2=100; }
```



```
while (analogRead(A5) <= 300) {停車 }
if (digitalRead(7) == LOW)
{ 左轉 }
if (digitalRead(6) == LOW)
{ 右轉 }
如果左右紅外線偵測間隔 < 0.5秒
後退0.5秒
轉彎0.8秒
```

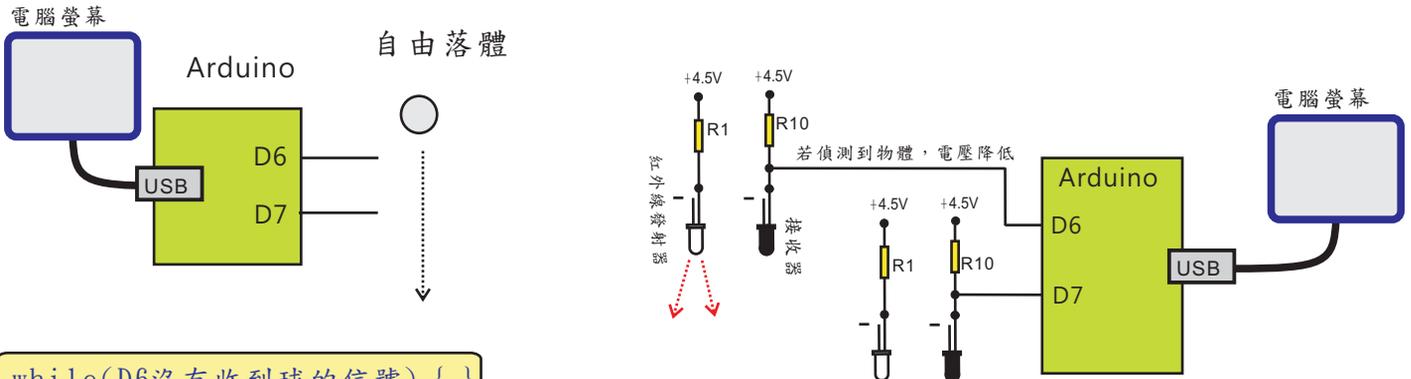
5	4	馬達A
Low	Low	停
Low	High	前進
High	Low	後退

3	2	馬達B
Low	Low	停
Low	High	前進
High	Low	後退



實驗26 自由落體實驗時間的記錄

球落下來，經過兩組紅外線感測器，
Arduino 計算經過第1點及第2點所花費的時間，
然後由電腦螢幕顯示時間。

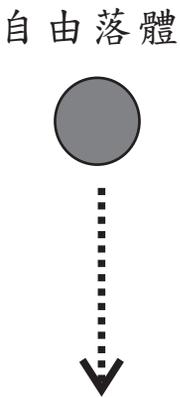


```
while(D6沒有收到球的信號) { }
time1=目前時間
while(D7沒有收到球的信號) { }
time2=目前時間
時間間隔=time2-time1
把時間間隔傳到電腦螢幕上
```

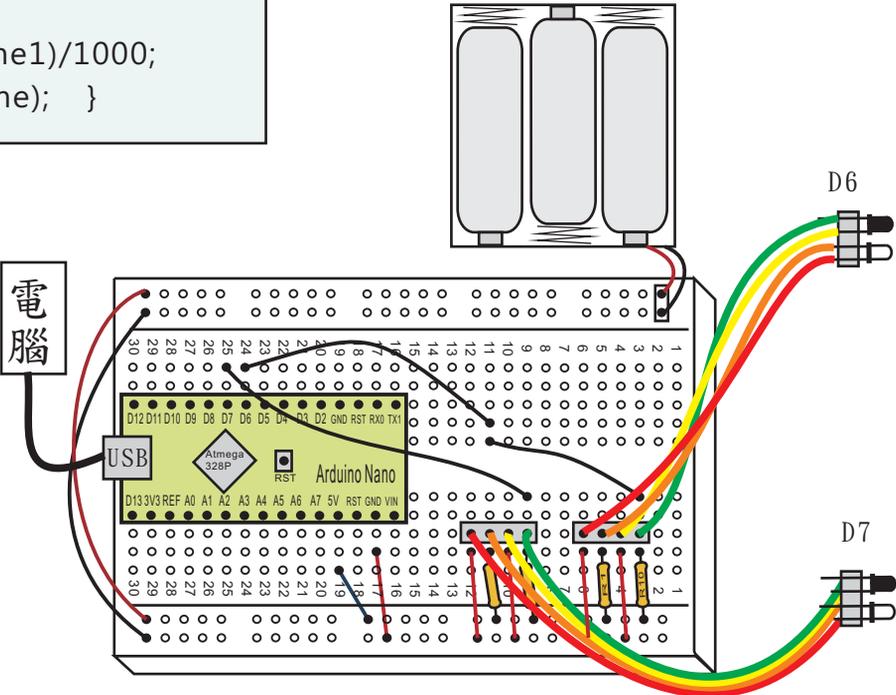
```
float time1,time2,totaltime;
void setup() {
  Serial.begin(9600);
  pinMode(6,INPUT);
  pinMode(7,INPUT); }

void loop() {
  while(digitalRead(6)==HIGH){ }
  time1=millis();
  while(digitalRead(7)==HIGH){ }
  time2=millis();
  totaltime=(time2-time1)/1000;
  Serial.println(totaltime); }
```

```
float 宣告帶小數點的浮點變數
Serial.begin(9600)
以每秒9600位元速度和電腦交換資訊
millis()
Arduino程式開始執行到現在的時間，
1000是1秒。
Serial.println 把資訊傳到電腦銀幕上
```



程式等待，
當D6偵測到球，就開始計時，
當D7偵測到球，就停止計時，
然後把時間傳回電腦。



上面的紅外線若偵測到球，D6會變成低電位
下面的紅外線若偵測到球，D7會變成低電位

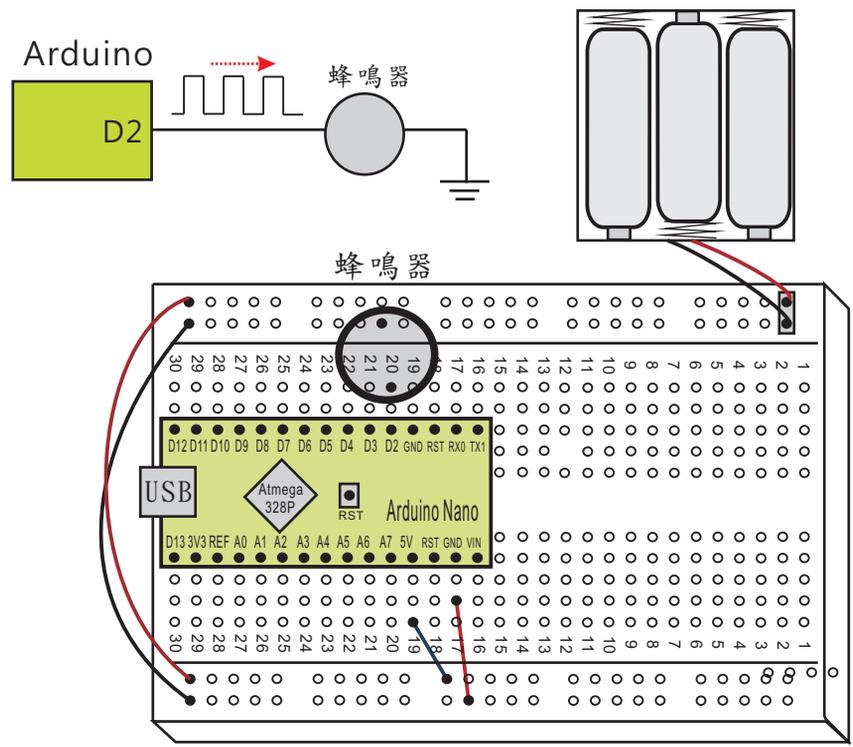
實驗27 音樂(生日快樂)

音樂頻率	Fa	So	La	Si	Do	Re	Mi	Fa	So	La	Si	Do	Re	Mi
	349	392	440	494	523	587	659	698	784	880	988	1046	1175	1318

第1個程式 生日快樂歌

```

void setup(){ }
void loop(){
  tone(2, 784);
  delay(200);
  noTone(2);
  delay(20);
  tone(2, 784);
  delay(200);
  tone(2, 880);
  delay(500);
  tone(2, 784);
  delay(500);
  tone(2, 1047);
  delay(500);
  tone(2, 988);
  delay(1300);
}
    
```



melody 和 duration 都是序列 · 各有6個整數元素

```
int melody[] = { so2, so2, la2,so2 ,do3, si2};
```

```
int duration[] = { 200,200, 500,500, 500 ,1300};
```

那麼

```
melody[0]=so2=784
```

```
melody[2]=la2=880
```

```
duration[5]=1300
```

tone(n,k)
對n腳輸出每秒k次的振盪頻率
noTone(n)
讓n腳停止振盪

第2個程式 用英文取代頻率

```

int so2 = 784;
int la2 = 880;
int si2 = 988;
int do3 = 1046;
void setup() { }
void loop() {
  tone(2,so2);
  delay(200);
  noTone(2);
  delay(20);
  tone(2,so2);
  delay(200);
  tone(2,la2);
  delay(500);
  tone(2,so2);
  delay(500);
  tone(2,do3);
  delay(500);
  tone(2,si2);
  delay(1300);
}
    
```

第3個程式

利用序列及 FOR 迴圈簡化程式

```

int so2 = 784;
int la2 = 880;
int si2 = 988;
int do3 = 1046;
void setup() { }
void loop() {
  int melody[] = { so2, so2, la2,so2 ,do3, si2};
  int duration[] = { 200,200, 500,500, 500 ,1300};
  for (int A = 0; A < 6; A=A+1) {
    tone(2, melody[A]);
    delay(duration[A]); }
}
    
```

實驗28 超音波閃避障礙的魔法車

蝙蝠也是用超音波偵測前方

```
a = Ping(7,8);
if(a < 20)
{ 轉彎 }
else
{ 前進 };
```

如果前面20公分內有障礙物，車子就轉彎

距離 = 音速 * 時間 / 2
音速 = 340公尺/秒



D8為HIGH 則觸發超音波
若接收到反彈超音波，D7由HIGH變LOW
pulseIn(7, HIGH)可計算D7HIGH的時間。

```
void setup()
{ pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(7,INPUT);
  pinMode(8,OUTPUT);
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
}
```

```
void loop()
{ long a;
  a = Ping(7,8);
  if(a < 20)
  { digitalWrite(4,LOW);
    digitalWrite(5,HIGH); } A輪反轉
  else
  { digitalWrite(4,HIGH);
    digitalWrite(5,LOW); } A輪正轉
}
```

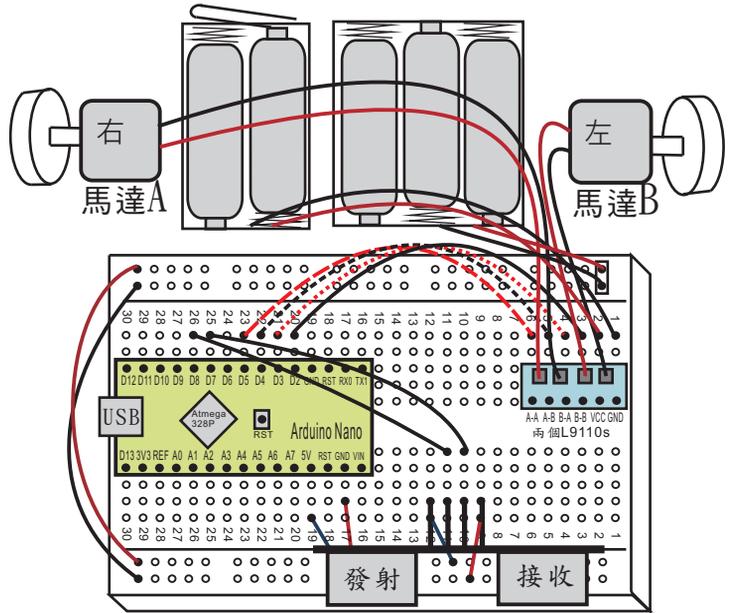
偵測距離的函式 Ping(接收腳·發射腳)

```
long Ping(int aa,int bb)
{ long duration, cm;
  digitalWrite(bb, HIGH);
  delayMicroseconds(10); 發射超音波 持續0.00001秒
  digitalWrite(bb, LOW);
  duration = pulseIn(aa, HIGH);
  return cm = 0.0340*duration / 2; }
```

距離 = 音速 * 時間 / 2

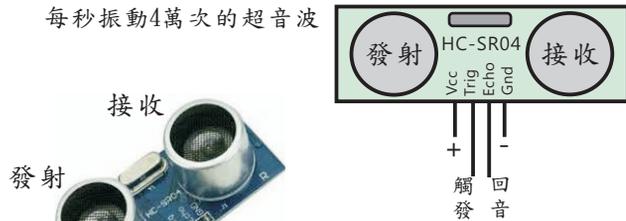
duration
發射出去到接收到回音的時間
單位是0.000001秒

函式若有return回傳，前面就不必加void，但是要加回傳數值的型態。



超音波感測器

每秒振動4萬次的超音波



馬達用久了會變慢，若一快一慢，車子走不直，可更換馬達。

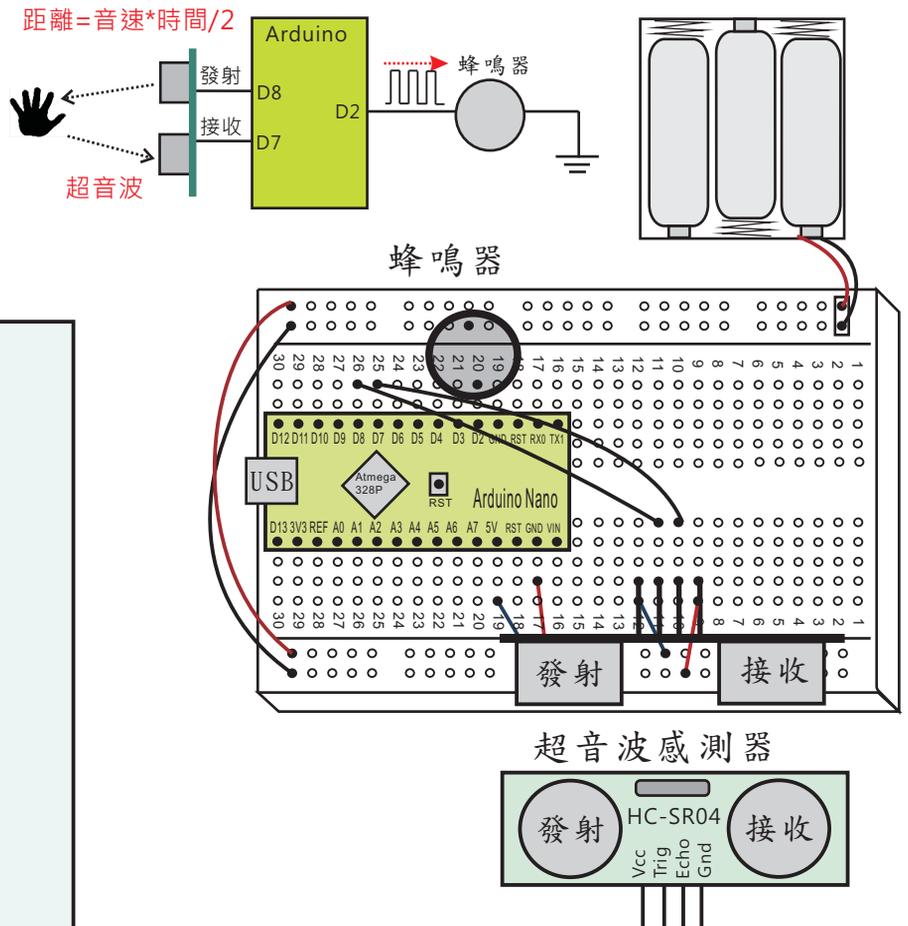


實驗29 超音波電子琴

手在超音波感測器前面揮動，彈出音樂

第1個程式

如果距離小於10公分
{發聲頻率523，持續0.1秒}
如果距離大於10公分，小於20公分
{發聲頻率587，持續0.1秒}
如果距離大於20公分，小於30公分
{發聲頻率659，持續0.1秒}
如果距離大於30公分
{發聲頻率698，持續0.1秒}



```
void setup()
{pinMode(2,OUTPUT);
pinMode(7,INPUT);
pinMode(8,OUTPUT);}

void loop() {
long aa;
aa= Ping(7,8); 偵測距離aa
```

```
if (aa < 10)          發出音樂
  {tone(2,523);
  delay(100);}
if (aa >= 10 && aa < 20)
  {tone(2,587);
  delay(100);}
if (aa >= 20 && aa < 30)
  {tone(2,659);
  delay(100);}
if (aa >= 30 && aa < 40)
  {tone(2,698);
  delay(100);}
if (aa >= 40)
  {noTone(2);}
}
```

超音波測距函式

```
long Ping(int bb,int cc)
{long duration, cm;
digitalWrite(cc, HIGH);
delayMicroseconds(10);
digitalWrite(cc, LOW);
duration = pulseIn(bb, HIGH);
return cm = duration* 0.034 / 2; }
```

距離=時間*音速/2

函式若有return回傳，前面就不必加void，但是要加回傳數值的型態。

第2個程式

聲音頻率設定為100*距離(公分)
距離越遠，頻率越高

```
void setup()
{pinMode(2,OUTPUT);
pinMode(7,INPUT);
pinMode(8,OUTPUT);
}

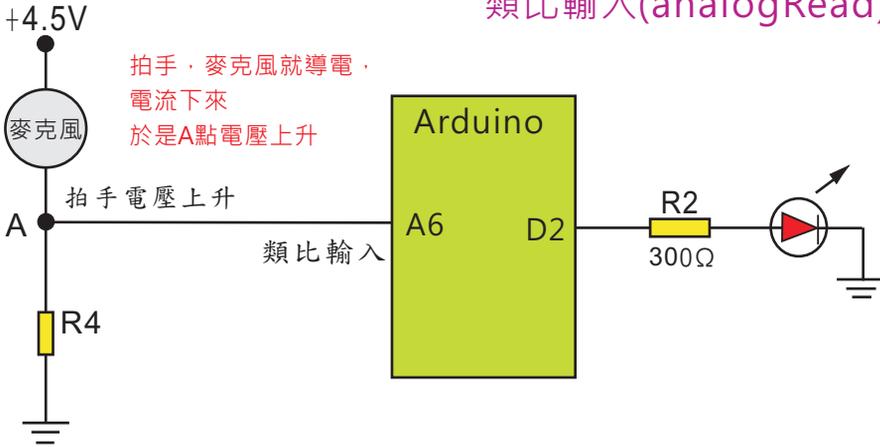
void loop()
{long aa;
aa= 100*Ping(7,8);
tone(2,aa);
delay(50); }
```

超音波測距函式

```
long Ping(int bb,int cc)
{long duration, cm;
digitalWrite(cc, HIGH);
delayMicroseconds(10);
digitalWrite(cc, LOW);
duration = pulseIn(bb, HIGH);
return cm = duration * 0.034 / 2; }
```

實驗30 拍拍手, 燈就亮

類比輸入(analogRead)



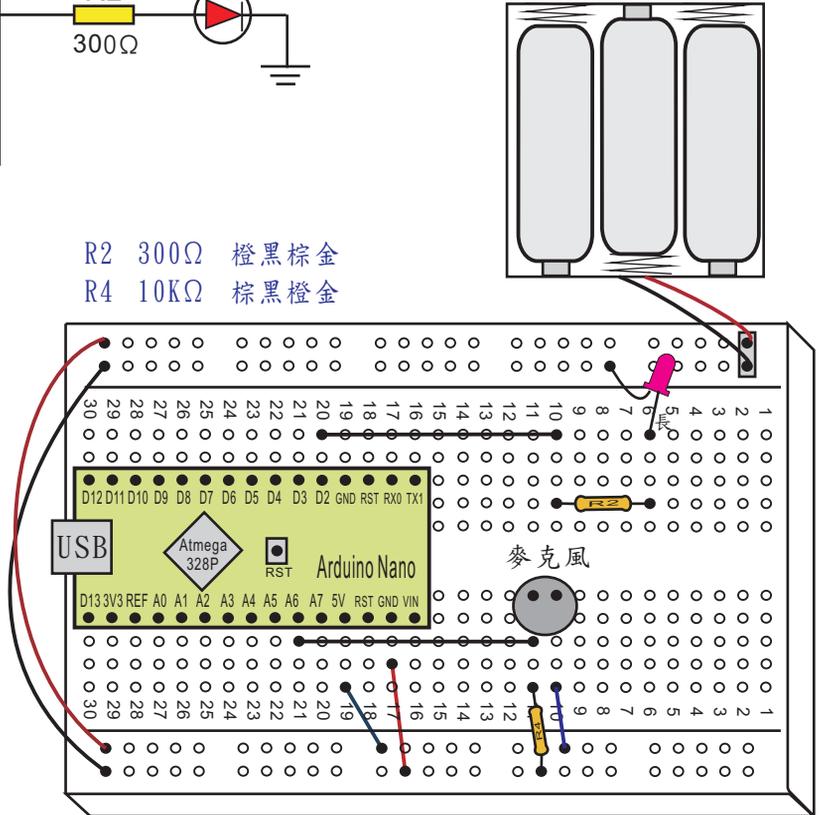
第1個程式

while(A6低於700){} 700若改800
燈亮2秒 就要拍更大聲

```
void setup() {
  pinMode(2,OUTPUT);}

void loop()
{ while(analogRead(A6)<700){}
  digitalWrite(2,HIGH);
  delay(2000);
  digitalWrite(2,LOW); }
```

R2 300Ω 橙黑棕金
R4 10KΩ 棕黑橙金



上下這兩程式功能一樣·要拍大聲燈才亮

第2個程式

讀取A6值;
如果(A6大於 700)
{ 燈亮2秒 }

若麵包板孔接觸不良,
可取出麵包板,
用手把背面金屬條
用力壓回去。

```
int v;
void setup() {
  pinMode(2,OUTPUT);
}

void loop()
{ v=analogRead(A6);
  if (v > 700)
  { digitalWrite(2,HIGH);
    delay(2000);
    digitalWrite(2,LOW); }
}
```

第3個程式 很靈敏·輕拍手就亮

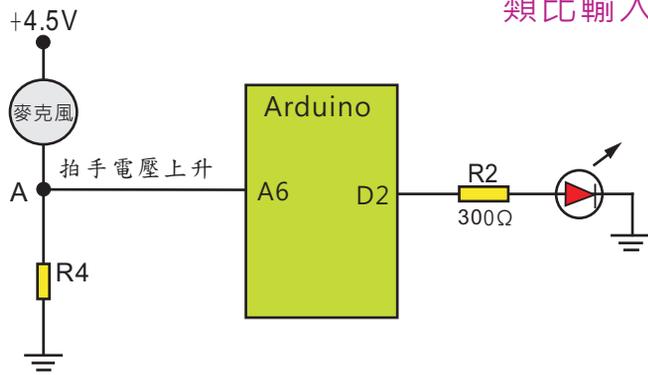
讀取A6值;
隔0.001秒再讀一次
如果(增加10以上)
{ 燈亮2秒 }

```
int v1,v2;
void setup() {
  pinMode(2,OUTPUT); }

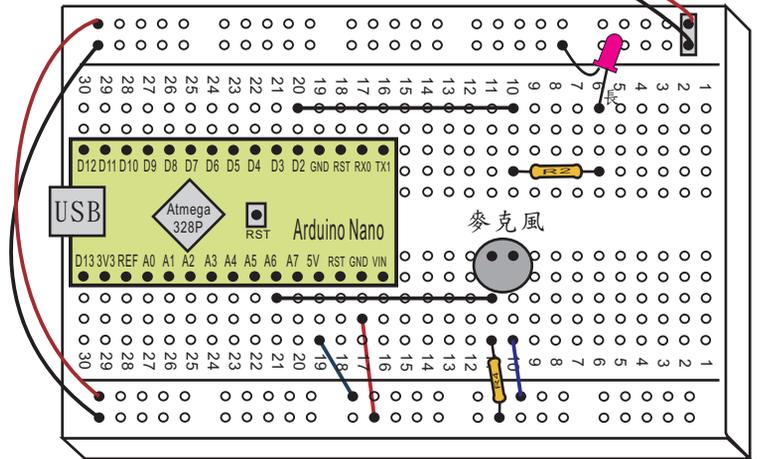
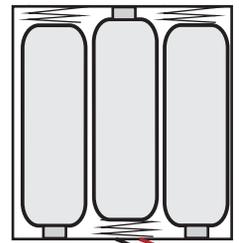
void loop()
{ v1=analogRead(A6);
  delay(1);
  v2=analogRead(A6);
  if ((v2-v1) > 10)
  { digitalWrite(2,HIGH);
    delay(2000);
    digitalWrite(2,LOW); }
}
```

10改小
會更靈敏

實驗31 偵測拍手次數、拍手時間



類比輸入(analogRead)



第1個程式

偵測2秒內拍幾次手，
燈就閃幾次

```
while(沒拍手){
time1=目前時間;
num=1;
while(目前時間-time1 < 2秒)
{ 如果(拍手){num=num+1} }
```

最後NUM就是兩秒內的拍手次數

700若改800
就要拍更大聲

第2個程式

偵測拍手後
隔了幾秒
再拍第二次手，
隔幾秒，燈就閃幾次

```
while(沒拍手){
time1=目前時間;
while(沒拍手){
time2=目前時間;
time1-time1 就是間隔時間
```

```
float time1; float 有小數點的變數
int num,i; int 整數變數
void setup() {
pinMode(2,OUTPUT); }
```

每次偵測到拍手聲
都讓程式暫停0.1秒
delay(100)
再進行下一次偵測
避免重覆偵測到同
一次拍手

偵測拍手幾次

```
while(analogRead(A6)<700){
time1=millis(); 第一次拍手的時間
num=1;
delay(100); 0.1秒內很難拍2次手
while(millis()-time1 < 2000){
if(analogRead(A6)>700)
{ delay(100); 偵測2秒內拍了幾次手
num=num+1; } }
```

```
for (i = 1; i <= num ; i=i+1)
{ digitalWrite(2,HIGH); LED閃num次
delay(500);
digitalWrite(2,LOW);
delay(500);
}
```

```
float time1,time2;
int num,i;
void setup() {
pinMode(2,OUTPUT); }
```

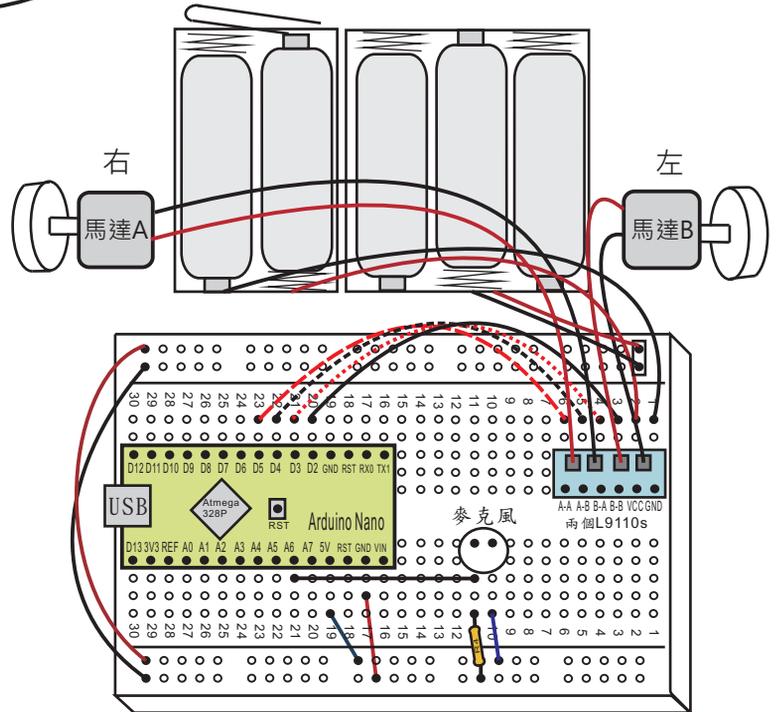
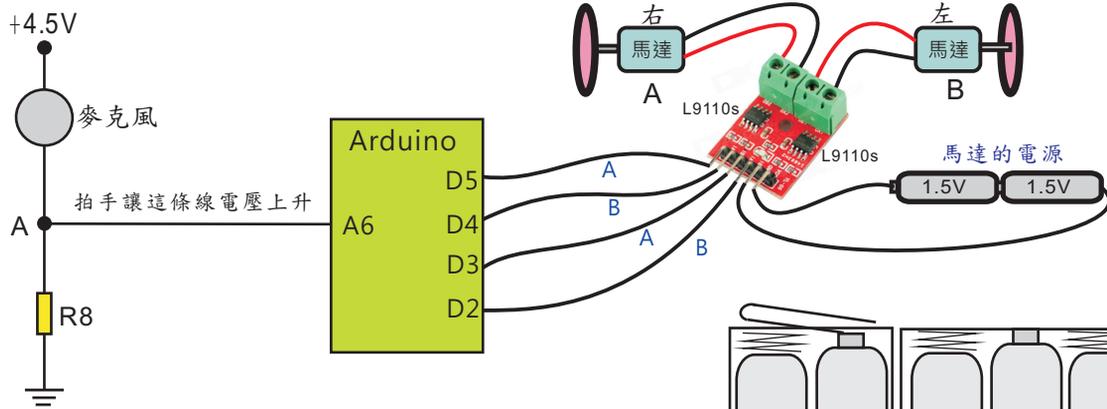
偵測拍手時間

```
while(analogRead(A6)<700){
time1=millis(); 第一次拍手的時間
digitalWrite(2,HIGH);
delay(100);
while(analogRead(A6)<700){
time2=millis(); 第二次拍手的時間
digitalWrite(2,LOW); 兩次拍手隔了num秒
num=(time2-time1)/1000;
delay(2000);
```

```
for (i = 1; i <= num ; i=i+1)
{ digitalWrite(2,HIGH);
delay(500); LED閃num次
digitalWrite(2,LOW);
delay(500); }
```

實驗32 聲控車

類比輸入(analogRead)



第1個程式

拍拍手，
車子就動兩秒

```

if (A6 > 800)
{ 左輪前進;
  右輪前進;
  delay(2000);
  左輪停;
  右輪停;
  delay(500); }
    
```

```

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}
// 2,3,4,5一開始都是LOW

void loop()
{if (analogRead(A6) > 800)
{
  digitalWrite(2, HIGH);
  digitalWrite(4, HIGH);
  delay(2000);
  digitalWrite(2, LOW);
  digitalWrite(4, LOW);
  delay(500);
}
}
    
```

車子停止半秒後
再偵測麥克風
以避開馬達干擾

第2個程式

拍拍手，
車子就轉1秒

```

if (A6 > 800)
{ 左輪後退;
  右輪前進;
  delay(2000);
  左輪前進;
  右輪前進;
  delay(500); }
    
```

拍拍手，車子就轉彎1秒

```

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(4, HIGH); }

void loop()
{if (analogRead(A6) > 800) {
  digitalWrite(2, LOW); // 後退
  digitalWrite(3, HIGH);
  delay(1000);
  digitalWrite(2, HIGH); // 前進
  digitalWrite(3, LOW);
  delay(500); } }
    
```

5	4	馬達A
Low	Low	停
Low	High	前進
High	Low	後退

3	2	馬達B
Low	Low	停
Low	High	前進
High	Low	後退

實驗33 拍手次數 (拍1次前進、拍2次後退、拍3次左轉、拍4次右轉)

```
float time1;
int num;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT); }

void loop() {
  while(analogRead(A6)<800){
    time1=millis();
    num=1;
    delay(100);
    while(millis()-time1 < 1000){
      if(analogRead(A6)>800) {
        delay(100);
        num=num+1; } }

    switch(num){
    case 1:
      forward();
      delay(2000);
      stopward();
      delay(500);
      break;
    case 2:
      backward();
      delay(2000);
      stopward();
      delay(500);
      break;
    case 3:
      leftward();
      delay(500);
      stopward();
      delay(500);
      break;
    case 4:
      rightward();
      delay(500);
      stopward();
      delay(500);
      break;
    }
  }
}
```

偵測一秒內拍了幾次手

前進2秒

後退2秒

左轉0.5秒

右轉0.5秒

拍手一次,車子就前進兩秒
 拍手兩次,車子就後退兩秒
 拍手三次,車子就左轉半秒
 拍手四次,車子就右轉半秒
 (連續拍手要在一秒之內完成)

```
switch(num) {
  case 1:
    AAAA;
    break;
  case 2:
    BBBB;
    break;
  case 3:
    CCCC;
    break; }
```

如果num=1 就執行AAAA
 如果num=2 就執行BBBB
 如果num=3 就執行CCCC

每次偵測到拍手聲
 都讓程式暫停0.1秒
 再進行下一次偵測
 避免重覆偵測到同一次拍手

每次馬達停止
 讓程式暫停0.5秒
 再繼續偵測拍手聲
 以避開馬達及車聲的干擾

這4個函式
 屬於左邊程式

```
void forward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW); }
```

前進

```
void backward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH); }
```

後退

```
void leftward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW); }
```

左轉彎

```
void rightward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH); }
```

右轉彎

```
void stopward(){
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW); }
```

停

實驗34 如何檢視你家的遙控器內碼

接收器採用38KHz紅外線接收器，
遙控器採用汽車音響用的遙控器，
或家裡的電視冷氣遙控器都可用，

用本頁的程式找出遙控器按鍵的內碼，
再用下一頁程式來讀入這些按鍵的內碼，控制車子的動作。

檔案 編輯 草稿碼 **工具** 說明

自動格式化
封存草稿碼
修正編碼並重新載入
管理程式庫...
序列埠監控視窗
序列繪圖家
WiFi101/WiFiNINA
開發板:"Arduino Nano"
處理器:"ATmega328P"

請先進入 "管理程式庫"

找到IRremote

從網站安裝IRremote.h這個程式庫到你電腦裡
你的程式裡的第一行 #include <IRremote.h>
會把IRremote.h這個檔案抓進你的程式
你才能使用檢查遙控器內碼的指令

這個程式可用來檢查遙控器各鍵的內碼
按遙控器按鍵
電腦螢幕會顯示這個鍵的內碼

```
#include <IRremote.h>
int receiver = 7;
IRrecv irrecv(receiver);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop() {
  if (irrecv.decode(&results)){
    Serial.println(results.value);
    delay(100);
    irrecv.resume();
  }
}
```

16753245

電腦螢幕

按下紅外線遙控器
電腦螢幕會顯示按鍵內碼

按這鍵

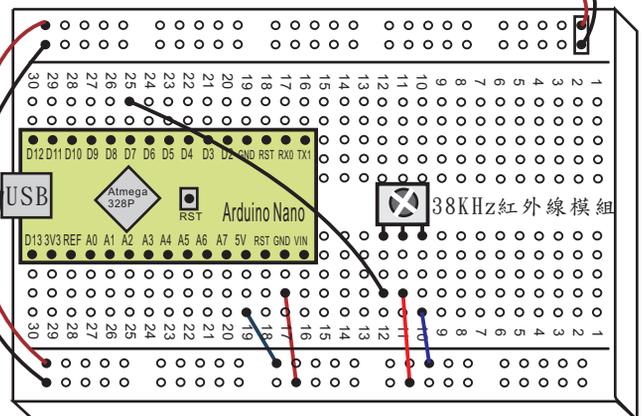
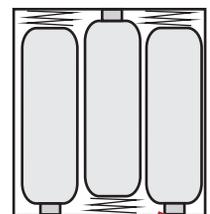


這兩個遙控器的內碼是一樣的
只是左邊遙控器少了四個按鍵



左圖的遙控器每個按鍵的內碼

16753245	16736925	16769565
16720605	16712445	16761405
16769055	16754775	16748655
16738455	16750695	16756815
16724175	16718055	16743045
16716015	16726215	16734885
16728765	16730805	16732845



USB線要連接

實驗35 紅外線遙控器電子琴

把遙控器變成電子琴
任何紅外線遙控器都可

```
#include <IRremote.h>
int receiver = 7;
IRrecv irrecv(receiver);
decode_results results;
void setup(){
  irrecv.enableIRIn();
  pinMode(2,OUTPUT); }
```

要先上網下載
IRremote.h這個檔案
再用#include這個指令
把這檔案變成
你程式的一部份

```
void loop() {
  if (irrecv.decode(&results)){
```

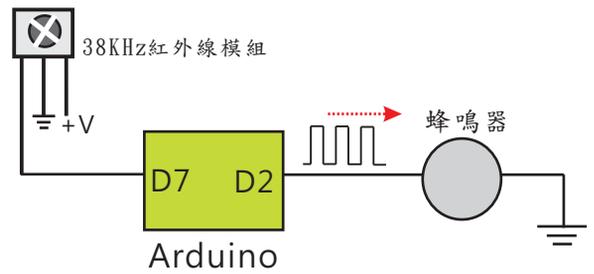
```
    switch(results.value) {
      case 16724175:      按鍵1
        newtone(2,523,200); Do
        break;
      case 16718055:      按鍵2
        newtone(2,587,200); Re
        break;
      case 16743045:      按鍵3
        newtone(2,659,200); Mi
        break;
      case 16716015:      按鍵4
        newtone(2,698,200); Fa
        break;
      case 16726215:      按鍵5
        newtone(2,784,200); So
        break;
      case 16734885:      按鍵6
        newtone(2,880,200); La
        break;
    }
```

```
    delay(10);
    irrecv.resume();
  } }
```

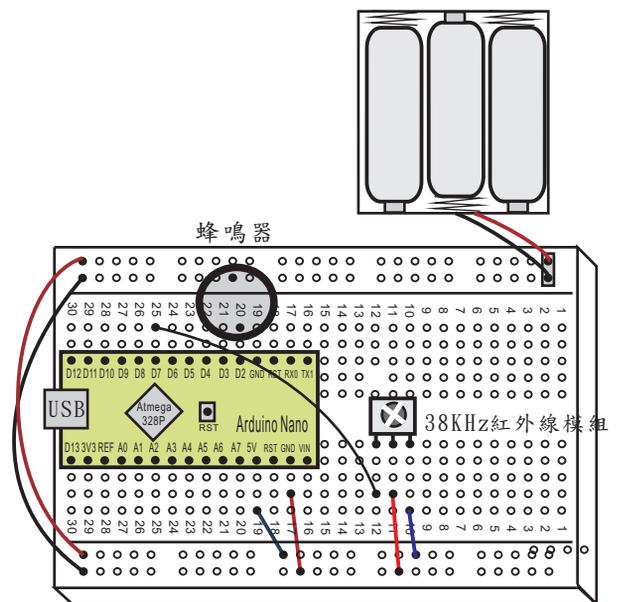
遙控器的函式庫IRremote.h
和發聲指令tone()會衝突。
可用這個函數newtone()
取代tone()這個指令。

```
void newtone(int t, int f, int d) {
  int p = 1000000L / f;
  int pulse = p / 2;
  for (long i = 0; i < d * 1000L; i += p) {
    digitalWrite(t, HIGH);
    delayMicroseconds(pulse);
    digitalWrite(t, LOW);
    delayMicroseconds(pulse); } }
```

newtone(輸出腳, 頻率, 時間)



按遙控器數字鍵1,2,3,4...
發出Do,Re,Mi,Fa...



左圖的遙控器每個按鍵的內碼



16753245	16736925	16769565
16720605	16712445	16761405
16769055	16754775	16748655
16738455	16750695	16756815
16724175	16718055	16743045
16716015	16726215	16734885
16728765	16730805	16732845

	Fa	So	La	Si
	349	392	440	494
Do	Re	Mi	Fa	So
523	587	659	698	784
Do	Re	Mi		
1046	1175	1318		

實驗36 紅外線遙控車

```
#include <IRremote.h>
int receiver = 7;
IRrecv irrecv(receiver);
decode_results results;
void setup(){
  irrecv.enableIRIn();
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)){
    switch(results.value) {
      case 16736925:
        forward(); 前進
        break;
      case 16754775:
        backward(); 後退
        break;
      case 16720605:
        leftward(); 左轉
        break;
      case 16761405:
        rightward(); 右轉
        break;
      case 16712445:
        stopward(); 停
        break;
    }
    delay(100);
    irrecv.resume();
  }
}
```

```
void forward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
}

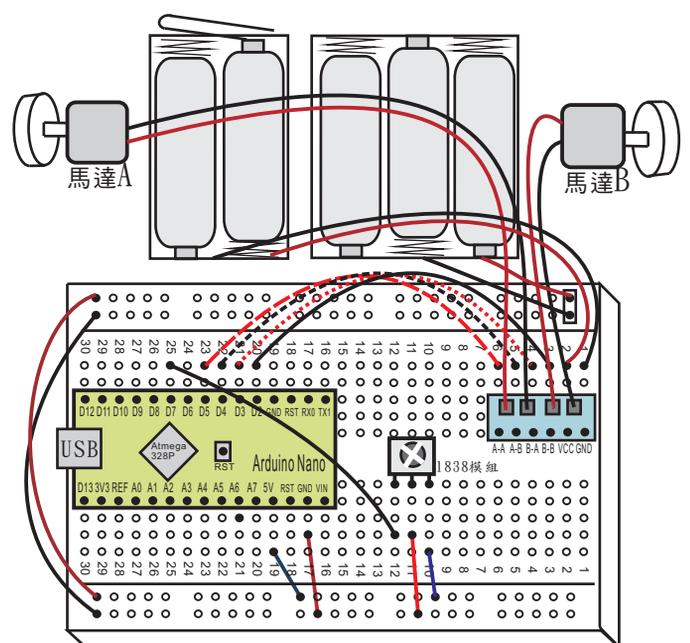
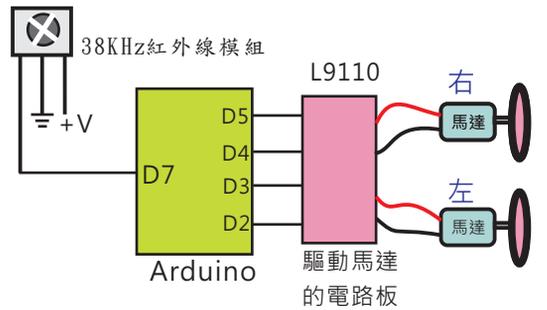
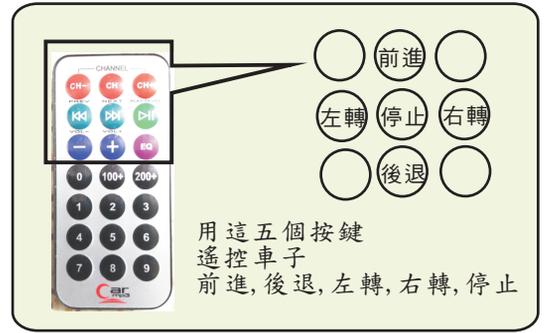
void backward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
}

void leftward(){
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW);
}

void rightward(){
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);
}

void stopward(){
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
}
```

```
if (按下鍵){
  switch(按鍵內碼)
  { case 16736925:
    前進
  case 16754775:
    後退
  case 16720605:
    左轉
  case 16761405:
    右轉
  case 16712445:
    停
  }
}
```



兩個人同時遙控，
如何避免互相干擾？

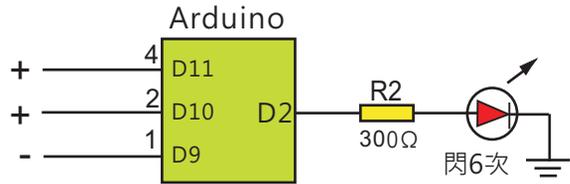
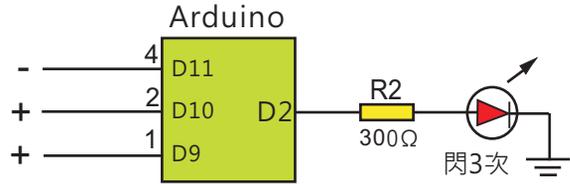
16736925改成16750695
16754775改成16726215
16720605改成16724175
16761405改成16743045
16712445改成16718055

用這五個按鍵
遙控車子
前進, 後退, 左轉, 右轉, 停止

實驗37 用三條電線設定燈閃的次數

```

if (D9 是高電位)
  {aa= 1;}
if (D10 是高電位)
  {bb = 1;}
if (D11是高電位)
  {cc = 1;}
dd= cc*4+bb*2+aa;}
    
```



```

void setup(){
  pinMode(2,OUTPUT);
  pinMode(9,INPUT);
  pinMode(10,INPUT);
  pinMode(11,INPUT);
  }
  
```

主程式

```

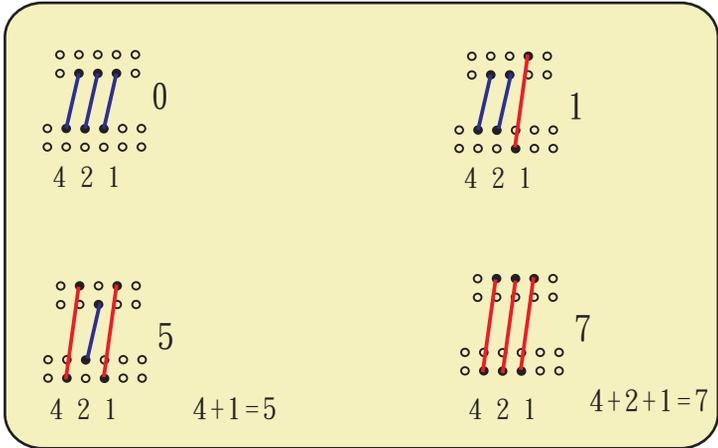
void loop() {
  int ee=checkMode();
  flash(ee);
  delay(2000); }
  
```

```

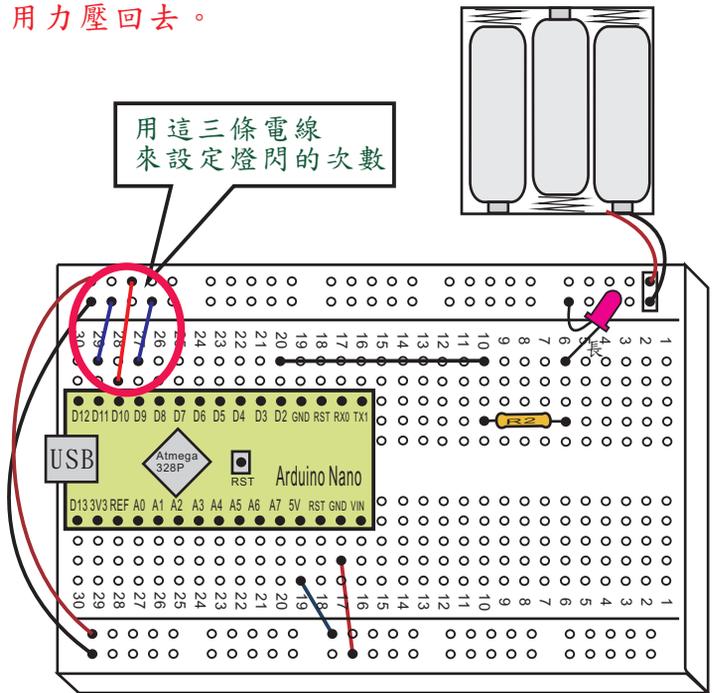
int checkMode(){ 檢查三條電線的函式
  int aa=0,bb=0,cc=0,dd;
  if (digitalRead(9) == HIGH)
    {aa= 1;}
  if (digitalRead(10) == HIGH)
    {bb = 1;}
  if (digitalRead(11) == HIGH)
    {cc = 1;}
  dd= cc*4+bb*2+aa;
  return dd; }
  
```

```

void flash(int n) { 閃n次
  for (int i=1;i<=n;i=i+1)
    { digitalWrite(2,HIGH);
      delay(300);
      digitalWrite(2,LOW);
      delay(300); } }
  
```



若麵包板孔接觸不良，
可取出麵包板，
用手把背面金屬條
用力壓回去。



flash函式裡沒有return回傳數值
所以前面要加上void

實驗38 如何把8個程式合併成一個大程式

程式架構

```
void setup(){ }
```

```
void loop(){ 主程式
  int ee=checkMode();
  switch (ee) {
    case 0:
      action0(); 程式0
      break;
    case 1:
      action1(); 程式1
      break;
    case 2:
      action2(); 程式2
      break;
    case 3:
      break;
    case 4:
      break;
    case 5:
      break;
    case 6:
      break;
    case 7:
      break;
  } }
```

```
int checkMode(){ 檢查三條電線
  int aa=0,bb=0,cc=0,dd;
  if (digitalRead(9) == HIGH)
    {aa= 1;}
  if (digitalRead(10) == HIGH)
    {bb = 1;}
  if (digitalRead(11) == HIGH)
    {cc = 1;}
  dd= cc*4+bb*2+aa;
  return dd; }
```

```
void action0() 八個小程式
{while(6>5){程式0 } }

void action1()
{ while(6>5){程式1 } }

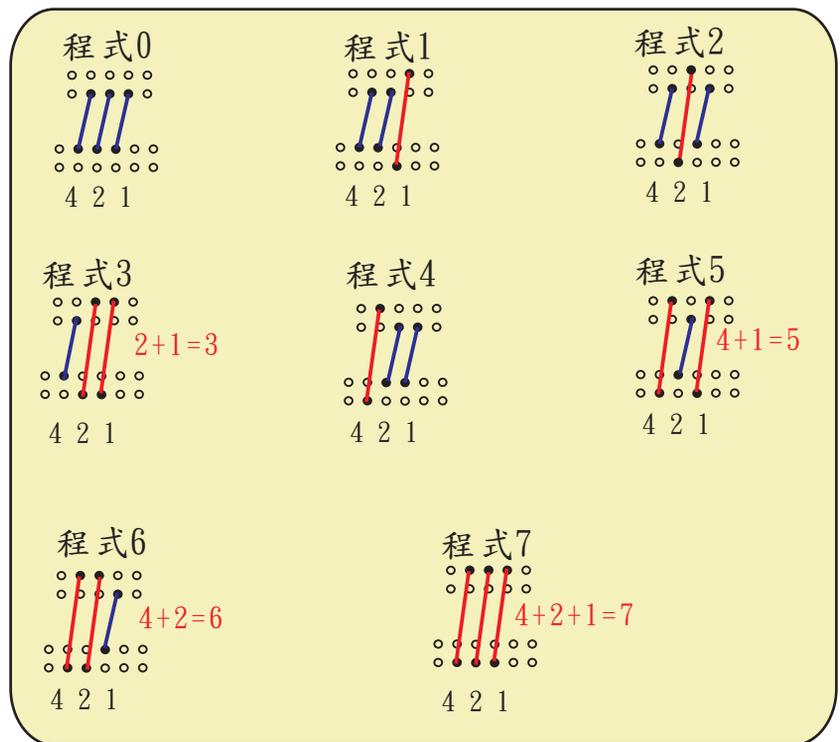
void action2()
{ while(6>5){程式2 } }
```

把聲控,光控,遙控器控制,超音波感測,紅外線感測...
等等多個程式整合成一個大程式,
同時寫入到arduino晶片中,
由左邊三條電線的接線方式,來選擇要執行哪一個程式,
三條線有八種排列組合,所以可以有八種玩法。

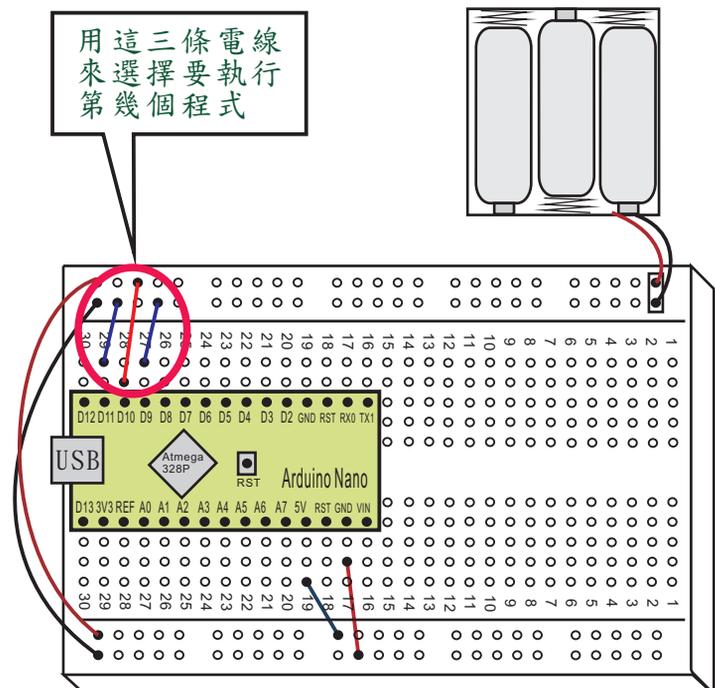
在此用while(6>5){程式} 取代 void loop(){程式}
讓程式反覆執行

8合1程式

用這三條電線選擇要執行哪一個程式



用這三條電線
來選擇要執行
第幾個程式



八個程式合成一個大程式 (8合1)

```
#include <IRremote.h>
float time1,time2,totaltime;
int num;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,INPUT);
  pinMode(7,INPUT);
  pinMode(8,OUTPUT);
  pinMode(9,INPUT);
  pinMode(10,INPUT);
  pinMode(11,INPUT); }
void loop() {
  int ee=checkMode();
  switch (ee) {
    case 0:
      action0();//實驗1、2 亮滅亮滅的燈、車
      break;
    case 1:
      action1();//實驗4、5 黑暗中自動亮的燈、車
      break;
    case 2:
      action2(); //實驗36 紅外線遙控車
      break;
    case 3:
      action3();//實驗36 遙控車(按鍵下移三行)
      break;
    case 4:
      action4();//實驗33 拍手聲控
      break;
    case 5:
      action5(); //實驗28 超音波偵測20公分
      break;
    case 6:
      action6();//實驗21 紅外線避障
      break;
    case 7:
      action7();//實驗23 紅外線避障(死巷退出)
      break;
  } }
```

主程式

```
void action0(){
  while(6>5){
    digitalWrite(2,HIGH);
    delay(1000);
    digitalWrite(2,LOW);
    delay(1500);
  } }
```

0

實驗1、2

```
void action1(){
  while(6>5){
    if(digitalRead(6)==LOW){
      digitalWrite(2,HIGH);
    }
    else {
      digitalWrite(2,LOW);
    }
  } }
```

1

實驗4、5

```
void action2() {
  int receiver=7;
  IRrecv irrecv(receiver);
  decode_results results;
  irrecv.enableIRIn();
  while(6>5) {
    if(irrecv.decode(&results)){
      switch(results.value){
        case 16736925 :
          forward();
          break;
        case 16754775 :
          backward();
          break;
        case 16720605:
          leftward();
          break;
        case 16761405 :
          rightward();
          break;
        case 16712445:
          stopward();
          break;
      }
      delay(100);
      irrecv.resume();
    }
  } }
```

2

實驗36

```
void action3() {
  int receiver=7;
  IRrecv irrecv(receiver);
  decode_results results;
  irrecv.enableIRIn();
  while(6>5) {
    if(irrecv.decode(&results)){
      switch(results.value){
        case 16750695 :
          forward();
          break;
        case 16726215 :
          backward();
          break;
        case 16724175:
          leftward();
          break;
        case 16743045 :
          rightward();
          break;
        case 16718055:
          stopward();
          break;
      }
      delay(100);
      irrecv.resume();
    }
  } }
```

3

實驗36

```
void action4() {
  while(6>5){
    while(analogRead(A6)<800){
      time1=millis();
      num=1;
      delay(100);
      while(millis()-time1 < 1000){
        if(analogRead(A6)>800) {
          delay(100);
          num=num+1;
        }
        switch(num){
          case 1:
            forward();
            delay(2000);
            stopward();
            delay(500);
            break;
          case 2:
            backward();
            delay(2000);
            stopward();
            delay(500);
            break;
          case 3:
            leftward();
            delay(500);
            stopward();
            delay(500);
            break;
          case 4:
            rightward();
            delay(500);
            stopward();
            delay(500);
            break; } }
  } }
```

4

實驗33

```
void action5()
{ long a;
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  while(6>5){
    a= Ping(7,8);
    if(a<20)
      {digitalWrite(4,LOW);
        digitalWrite(5,HIGH);
      }
    else
      {digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
      }
  }
  long Ping(int aa,int bb)
  {
    long duration, cm;
    digitalWrite(bb, HIGH);
    delayMicroseconds(10);
    digitalWrite(bb, LOW);
    duration = pulseIn(aa, HIGH);
    return cm = 0.0334*duration / 2;
  } }
```

5

實驗28

```
void action6(){
  while(6>5){
    if (digitalRead(6) == LOW)
      { digitalWrite(4, LOW);
        digitalWrite(5, HIGH);
      }
    else
      { digitalWrite(4, HIGH);
        digitalWrite(5, LOW);
      }
    if (digitalRead(7) == LOW)
      { digitalWrite(2, LOW);
        digitalWrite(3, HIGH);
      }
    else
      { digitalWrite(2,HIGH);
        digitalWrite(3,LOW);
      }
  } }
```

6

實驗21

```
void action7(){
  boolean a,b;
  time1=0;
  time2=100;
  while(6>5){
    a=digitalRead(6);
    delay(10);
    b=digitalRead(6);
    if (a==LOW && b==LOW)
      { digitalWrite(4, LOW);
        digitalWrite(5, HIGH);
        time1=millis()/1000;
      }
    else
      { digitalWrite(4, HIGH);
        digitalWrite(5, LOW);
      }
    a=digitalRead(7);
    delay(10);
    b=digitalRead(7);
    if (a==LOW && b==LOW)
      { digitalWrite(2, LOW);
        digitalWrite(3, HIGH);
        time2=millis()/1000;
      }
    else
      { digitalWrite(2,HIGH);
        digitalWrite(3,LOW);
      }
  }
  totaltime=time2-time1;
  if (abs(totaltime)<=0.5){
    backward();
    delay(2000);
    leftward();
    delay(1000);
    forward();
    time1=0;
    time2=100;
  }
}
```

7

實驗23

```
int checkMode(){
  int aa,bb,cc,dd;
  if (digitalRead(9) == HIGH)
    {aa= 1;}
  else{aa= 0;}
  if (digitalRead(10) == HIGH)
    {bb= 1;}
  else{bb= 0;}
  if (digitalRead(11) == HIGH)
    {cc= 1;}
  else{cc= 0;}
  dd= cc*4+bb*2+aa;
  return dd; }
```

檢查三條線

```
void forward() {
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW); }
void backward() {
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH); }
void leftward() {
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
  digitalWrite(5,LOW); }
void rightward() {
  digitalWrite(2,HIGH);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH); }
```

前進

後退

左轉

右轉

```
void stopward() {
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW); }
```

停

若改變三條電線，要重新啟動電源，或按RST鍵，讓程式重新啟動。

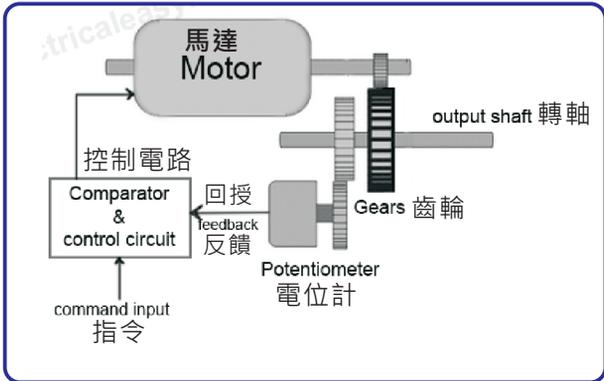
八合一程式(要用三條電線選擇)

- 0 實驗1、2 亮滅亮滅的燈、車
- 1 實驗4、5 黑暗中自動亮的燈、車
- 2 實驗36 紅外線遙控車
- 3 實驗36 遙控車(按鍵下移三行)
- 4 實驗33 拍手聲控
- 5 實驗28 超音波偵測20公分就轉彎
- 6 實驗21 紅外線避障
- 7 實驗23 紅外線避障(死巷退出)

實驗39 舵機(伺服馬達)

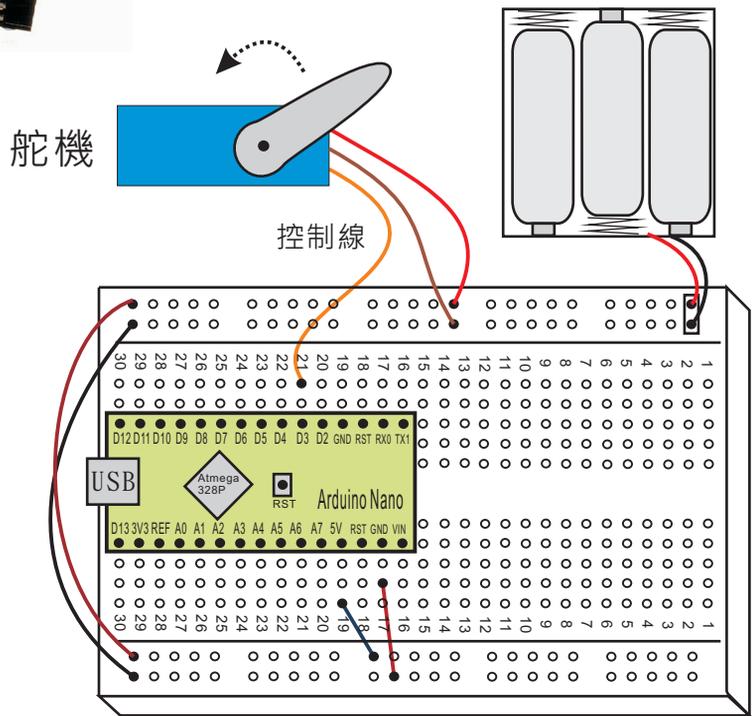


請先點選"工具" "管理程式庫"
安裝舵機的函式庫Servo.h



電位計
由電位來度量轉動角度
有時也叫可調電阻

舵機



write()這個函式
是在Servo.h裡定義的函式
用來設定舵機的角度

想想看，這兩個程式
舵機擺動方式有什麼不一樣？

第2個程式

角度在0和180之間

第1個程式

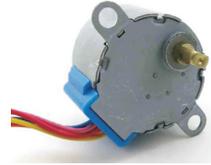
```
#include <Servo.h>
Servo s;
void setup()
{s.attach(3);}

void loop()
{ s.write(0);
  delay(2000);
  s.write(90);
  delay(2000);
  s.write(180);
  delay(2000);
}
```

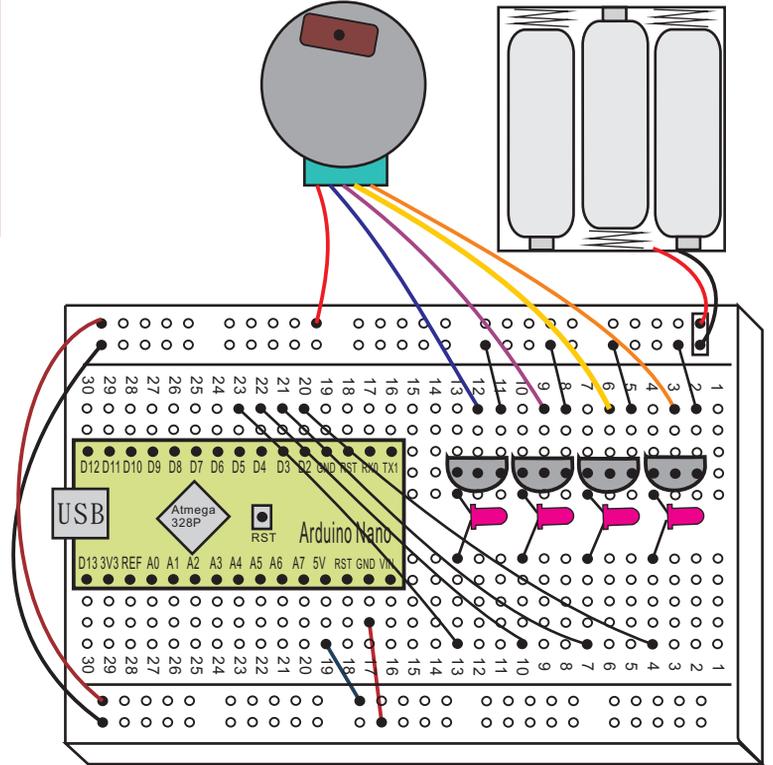
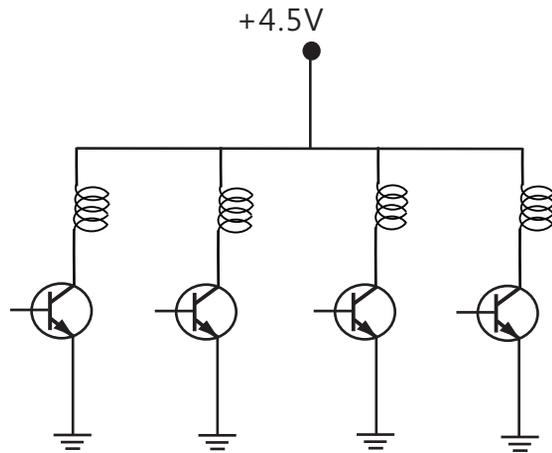
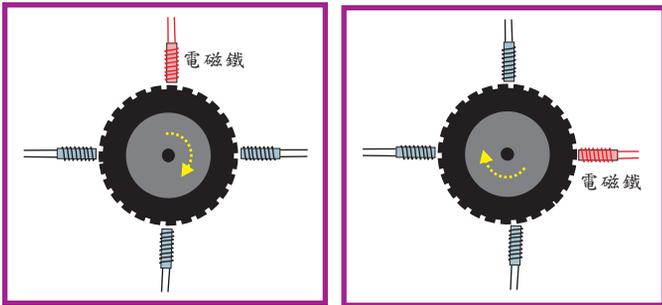
```
#include <Servo.h>
Servo s;
int p=0;
void setup()
{s.attach(3);}

void loop()
{for(p=0; p <= 180; p=p+2)
 { s.write(p);
   delay(100); }
  for(p=180; p >= 1; p=p-10)
 { s.write(p);
   delay(100); }
}
```

實驗40 步進馬達



這款步進馬達有減速齒輪
轉速很慢，但是扭力很大。



4個線圈輪流通電
順序要對
否則步進馬達只會抖動
不會轉動

3D列印機就是用步進馬達，
因為步進馬達可以準確的轉動。

第1個程式 逆時針轉

第2個程式 順時針轉



```
int i;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop() {
  for (i=2; i<=5; i=i+1)
  { digitalWrite(i,HIGH);
    delay(100);
    digitalWrite(i,LOW);}
}
```

delay(100)改成delay(10)
速度會變快

```
int i;
void setup() {
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
}

void loop() {
  for (i=5; i>=2; i=i-1)
  { digitalWrite(i,HIGH);
    delay(10);
    digitalWrite(i,LOW);}
}
```

步進馬達有四線、五線和六線，
六線的可把其中兩線並成一線，變成五線。
四線的就不一樣了，控制電路和程式都不一樣。

電阻

由電阻的顏色可辨認電阻值
 每種顏色代表一個數字
 黑0 棕1 紅2 橙3 黃4 綠5 藍6 紫7 灰8 白9

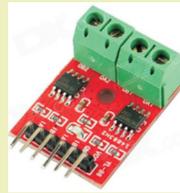
例如：
 橙黑棕 301，30後面再加1個0，就是300Ω
 棕紅橙 123，12後面再加3個0，就是12KΩ
 K代表1000，M代表百萬，Ω唸歐姆，
 第四色金色代表電阻的製造誤差為5%以內

R1	150Ω	棕綠棕金
R2	300Ω	橙黑棕金
R4	10KΩ	棕黑橙金
R7	20KΩ	紅黑橙金
R8	30KΩ	橙黑橙金
R10	6.2MΩ	藍紅綠金

超音波感測器



馬達驅動板



鎖緊端子線



Arduino Nano



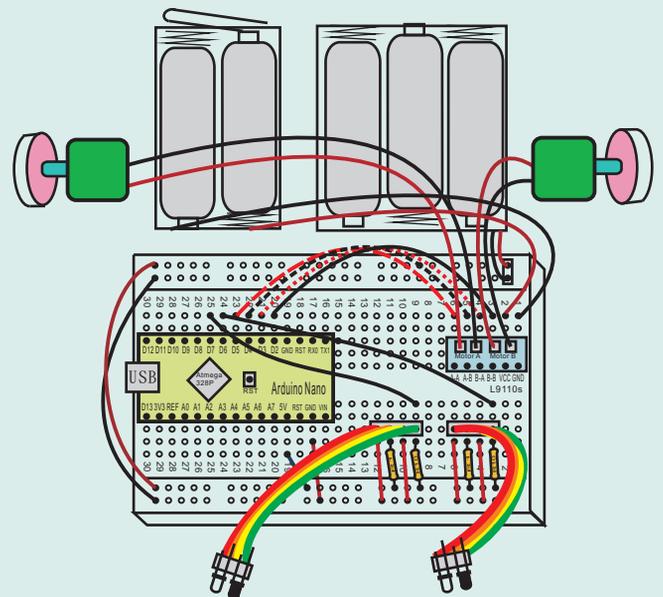
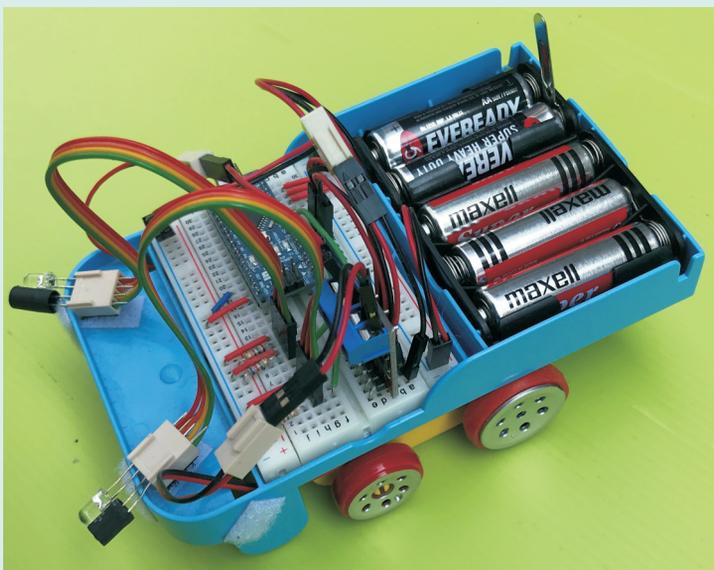
紅外線遙控器



38KHz
紅外線模組



科學魔法車D01型號，以科學魔法車為實驗平台，
 再加上面這些材料和本講義，可進行講義前38個實驗。



洛奇科技有限公司
 網購 www.letry.com.tw

學習順序

教課時數

- | | |
|--------------|------|
| 1 迷你魔法車 | 8小時 |
| 2 科學魔法車(初級版) | 12小時 |
| 3 科學魔法車(標準版) | 10小時 |
| 4 科學魔法車(程式版) | 20小時 |