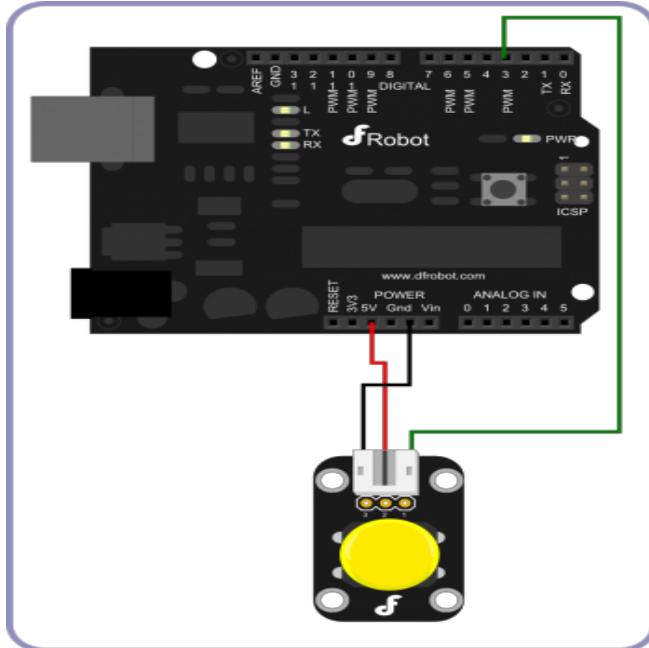


Lab3

Lab3.1 Digital Push Button

接線圖：



這一個實驗為arduino對於按鈕是否有按下的實驗，以下為範例程式，按鈕的input設為digital port 3

Sample Code

```
///Arduino Sample Code
///www.DFRobot.com
///Last modified on 16th September 2010

int ledPin = 13;           // choose the pin for the LED
int inputPin = 3;          // Connect sensor to input pin 3

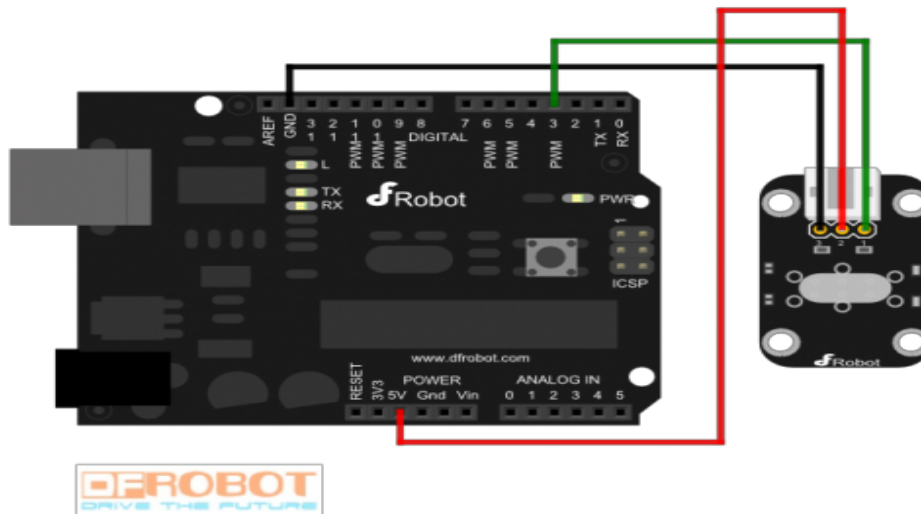
void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  int val = digitalRead(inputPin); // read input value
  if (val == HIGH) {              // check if the input is HIGH
    digitalWrite(ledPin, LOW);    // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH);   // turn LED ON
  }
}
```

按鈕在按下時板子內建的led燈會熄滅，放開時會亮起，實驗版上也有內建led燈可以讓大家來判斷按鈕是否有損壞！

Lab3.2 Digital Tilt Sensor

接線圖：



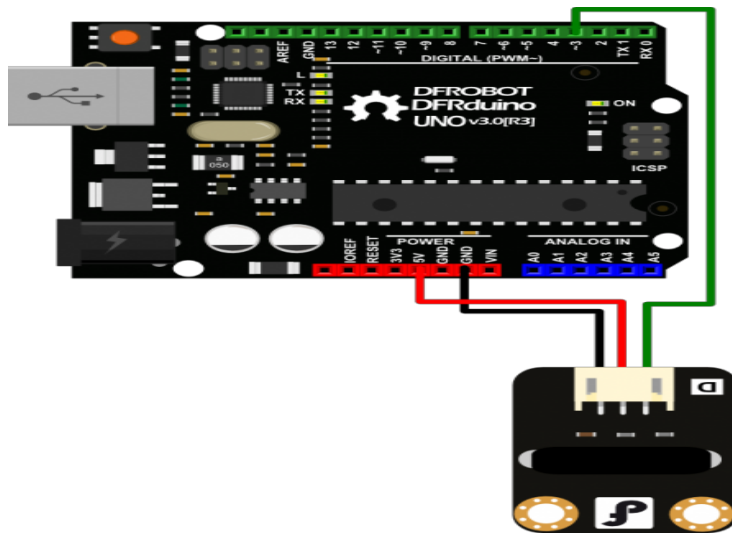
本次介紹的 sensor 為 Tilt Sensor，將下列 Sample Code 執行後，使用的 Port 為 Digital port 3，將板子傾斜，便可以觀察到他在某個角度會使 Led 開關變化！

Sample Code

```
1  int ledPin = 13;           // Connect LED to pin 13
2  int switcher = 3;         // Connect Tilt sensor to Pin3
3
4  void setup()
5  {
6      pinMode(ledPin, OUTPUT);    // Set digital pin 13 to output mode
7      pinMode(switcher, INPUT);   // Set digital pin 3 to input mode
8  }
9  void loop()
10 {
11
12     if(digitalRead(switcher)==HIGH) //Read sensor value
13     {
14         digitalWrite(ledPin, HIGH);    // Turn on LED when the sensor is tilted
15     }
16     else
17     {
18         digitalWrite(ledPin, LOW);     // Turn off LED when the sensor is not triggered
19     }
20 }
```

Lab3.3 Digital Vibration Sensor

接線圖：



本次實驗介紹的為Vibration Sensor，這一個Sensor與其他的Sensor比較不一樣的地方，這一個Sensor的input只能一直保持高電位，而只有在震動發生時會造成電位變化，而變化的時間很短，且變化完電位回到高電位，所以我們必須要使用另一種方式來達成我們的目的，也就是interrupt，如以下Sample Code：

Sample Code

```
1  #define SensorLED      13
2  #define SensorINPUT    3 //Connect the sensor to digital Pin 3 which is Interrupts 1.
3
4  unsigned char state = 0;
5
6  void setup()
7  {
8      pinMode(SensorLED, OUTPUT);
9      pinMode(SensorINPUT, INPUT);
10     attachInterrupt(1, blink, FALLING); // Trigger the blink function when the falling edge is detected
11 }
12
13 void loop()
14 {
15     if(state!=0)
16     {
17         state = 0;
18         digitalWrite(SensorLED,HIGH);
19         delay(500);
20     }
21     else
22         digitalWrite(SensorLED,LOW);
23 }
24
25
26 void blink()//Interrupts function
27 {
28     state++;
29 }
```

範例程式中有使用到一個為attachInterrupt 的function，而引三個引入的參數分別為interrupt的port，由於我們所使用的開發版為arduino uno r3，他所支援的只有digital的port 2 和port 3可以使用interrupt，而編號可以從參考資料中找到，而本次使用第三個digital port，編號為1，第二個為interrupt發生時要執行哪一個函式，而第三個發生什麼事件時要執行這一個function(trigger)，有五種模式可以設定，如下所示：

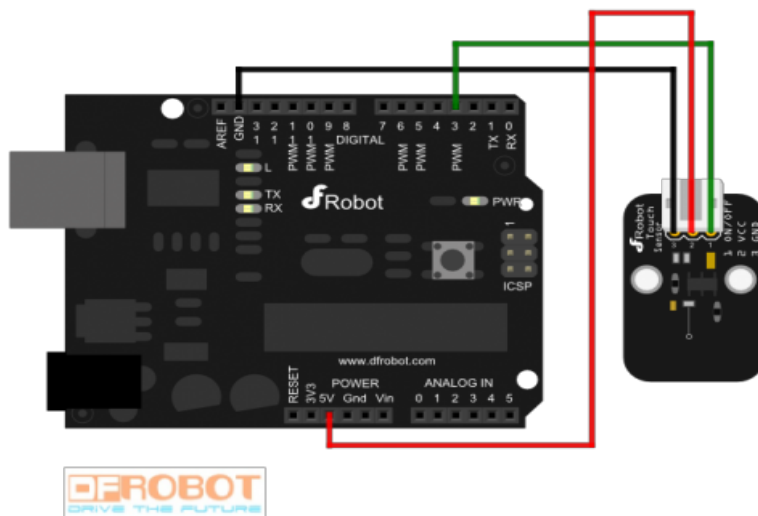
- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value
- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.
- **HIGH** to trigger the interrupt whenever the pin is high.(Duo only)

而我們將trigger設定為FALLING即為當電位降下時，arduino無論在哪裡，都會跳去執行我們所設定好的程式，而執行完便會跳回原來暫停的地方，所以範例程式執行結果為當去晃動ic板時便會造成Led燈亮，不晃時Led燈熄。

參考資料：<http://arduino.cc/en/Reference/attachInterrupt>

Lab3.4 Capacitive Touch Sensor

接線圖：



本次介紹的 sensor 為Capacitive Touch Sensor，將下列Sample Code執行後，使用的Port為Digital port 3，觸碰板子上的金屬部分，觀察到他在某個角度會使Led開關變化！

```

int ledPin = 13;           // Connect LED on pin 13, or use the onboard one
int KEY = 3;               // Connect Touch sensor on Digital Pin 3

void setup()
{
  pinMode(ledPin, OUTPUT); // Set ledPin to output mode
  pinMode(KEY, INPUT);     //Set touch sensor pin to input mode
}
void loop()
{
  if(digitalRead(KEY)==HIGH) //Read Touch sensor signal
  {
    digitalWrite(ledPin, HIGH); // if Touch sensor is HIGH, then turn on
  }
  else
  {
    digitalWrite(ledPin, LOW); // if Touch sensor is LOW, then turn off the led
  }
}

```

器材：Arduino Uno R3*1
 I/O Expansion Shield V7擴充板*1
 Digital Push Button*1
 Digital Vibration Sensor*1
 Touch Sensor*1
 Magnetic Sensor*1
 SG90 Servo Motor*1

Arduino 內建了伺服馬達的Library。我們可以輕鬆控制伺服馬達旋轉角度和方向。

伺服馬達參考資料：<http://arduino.cc/en/reference/servo>

注意：

伺服馬達只能從0至180度旋轉，請控制好角度，否則馬達可能會損壞

Arduino IDE也有內建example code 讓大家參考

```

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos <= 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15);           // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos >= 0; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15);           // waits 15ms for the servo to reach the position
  }
}

```

接線：

一般伺服馬達有三條線，電源（紅色）、接地（黑或棕色）、訊號線（白、黃、橘、藍，甚至是黑色）

Demo:

這次的Demo請大家使用magnetic sensor和vibrate sensor控制馬達旋轉。這兩個sensor能分別讓馬達以**不同的方向**一度一度的旋轉，當馬達到達了0度或180度必須讓他往回旋轉。另外touch sensor為開關，必須接觸touch sensor的同時，操作magnetic sensor或是vibrate sensor才能使馬達旋轉。

例如：馬達一開始不會轉，而當touch sensor和magnetic sensor觸發時變為逆時鐘轉，放開touch或magnetic sensor馬達會停止，而當touch sensor和vibrate sensor一起觸發時馬達順時鐘轉，也就是一定要touch sensor + (magnetic or vibrate) sensor才會發生旋轉事件，而馬達在0或180度時請自動往反方向轉，同時發生(三個sensor皆觸發的情況)的情況不考慮。

Bonus:

器材：B10K可變電阻*1

可變電阻經常使用在音響的音量控制或是電子設備的格式功率或準位的控制。簡單來說也就是旋轉控制電源大小。

接線：

可變電阻中間腳位接到類比輸入(Analog Input) pin，剩下的兩支腳位，一支接到 5V，另外一支接到 GND

我們可以利用可變電阻控制馬達旋轉，旋轉可變電阻的同時，伺服馬達也跟著旋轉。

請增加push button為切換模式，當按下push button將切成可變電阻控制馬達模式，再按下push button切回magnetic sensor控制馬達旋轉。如果一直按著push button不會有任何反應。另外touch sensor一樣為開關，必須接觸touch sensor的同時，操作可變電阻或是magnetic sensor才能使馬達旋轉。