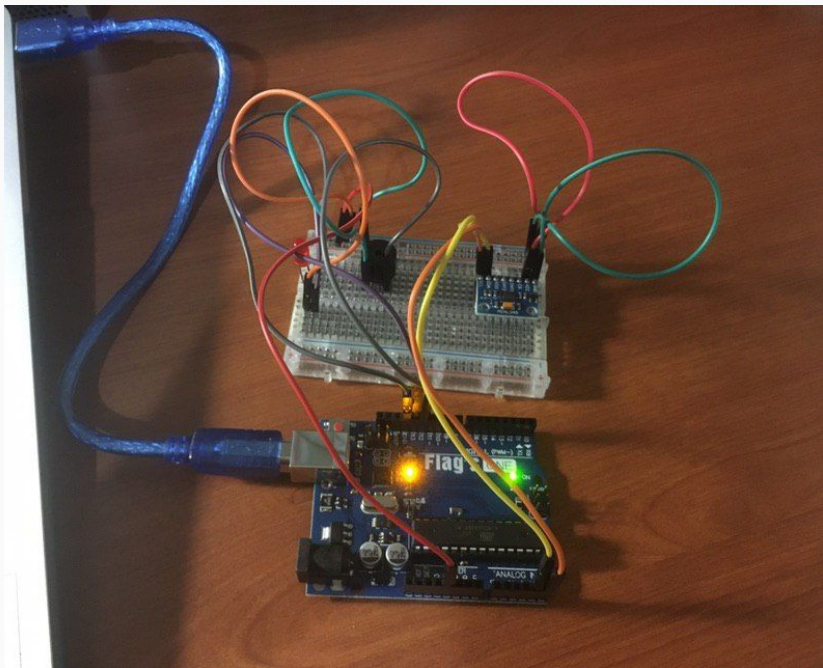


# 1072 數位學習期末專題

## 地震監測警報系統(ADXL345感測器)

資科三 U10516045 王冠中



## 地震感測器

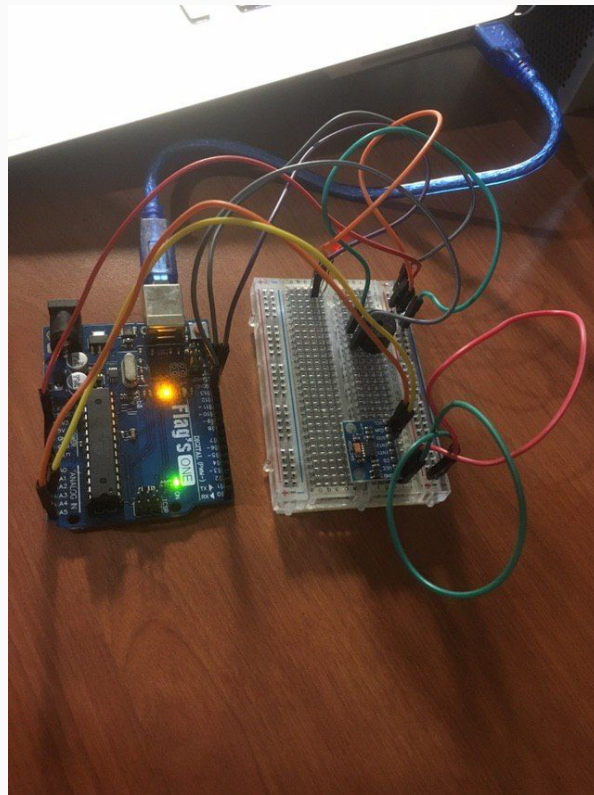
- 當ADXL345元件感測到晃動時，  
會讓LED燈發光並鳴笛
- 停止晃動後停止發光、鳴笛

## 邏輯

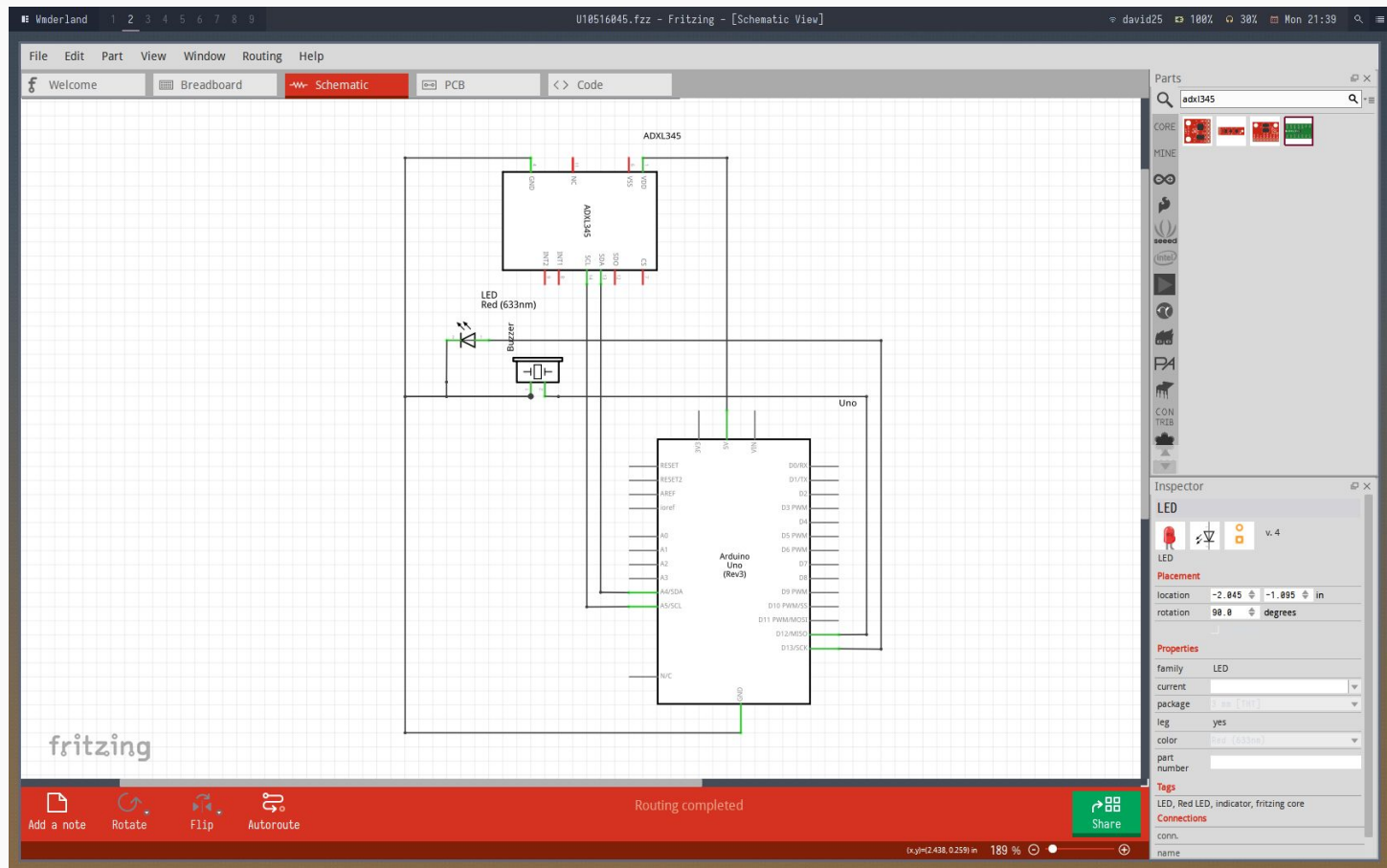
- `loop()` 中每次讀取sensor的x,y,z軸值，  
然後和前一次的x,y,z軸值比較(相減取絕對值)，  
如果大於一定的量就發亮鳴笛

# 元件清單

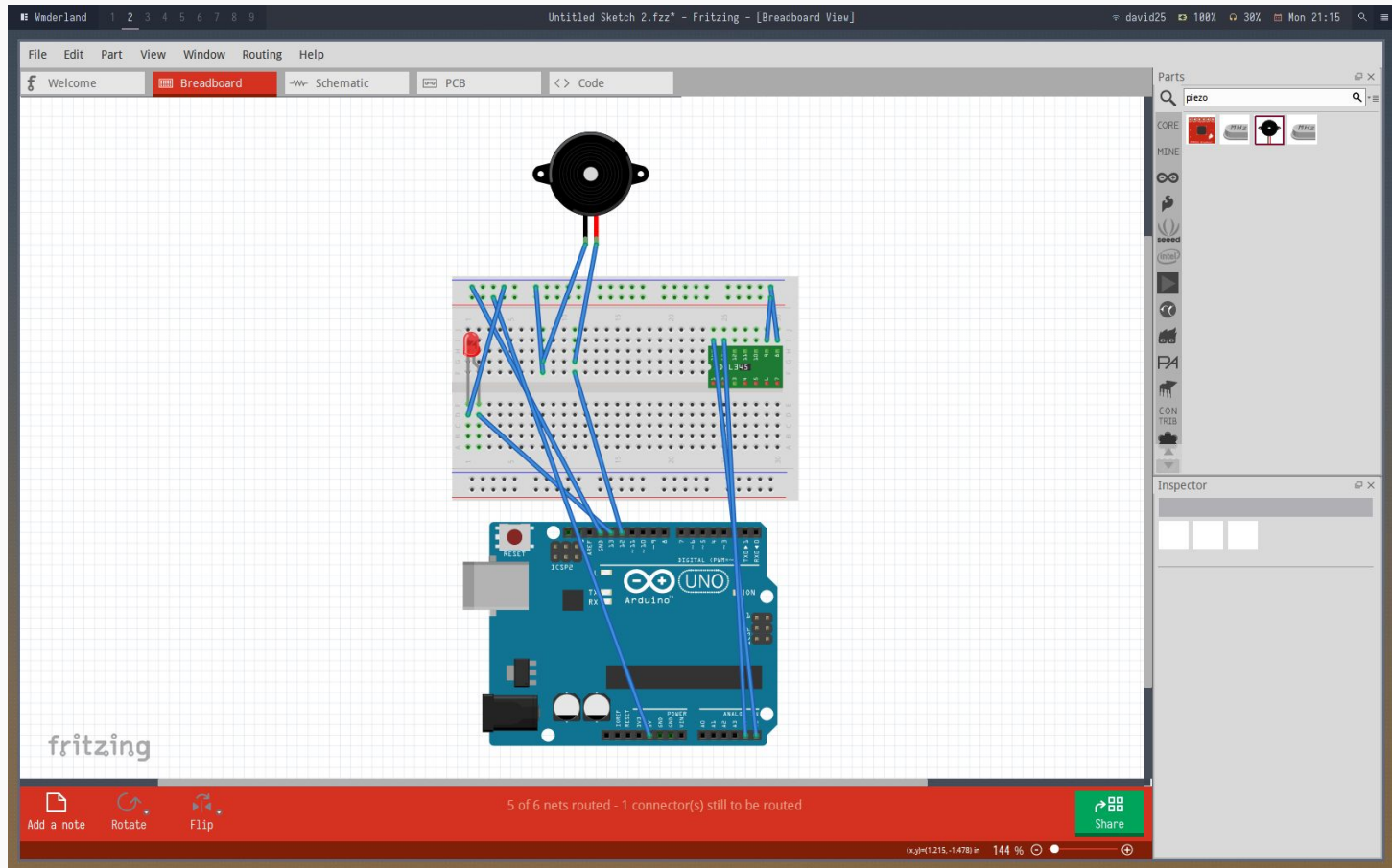
- Arduino Uno R3
- 面包板
- ADXL345
- LED(紅色)
- 蜂鳴器



# 電路圖



# 電路圖(麵包板接線圖)



# 實作

```
#include <Wire.h> // Wire library - used for I2C communication
```

```
#define ADXL345 0x53; // ADXL345 sensor I2C address
```

```
static float prevX = 0;  
static float prevY = 0;  
static float prevZ = 0;  
static float x = 0;  
static float y = 0;  
static float z = 0;
```

前一次的x,y,z  
和  
目前的x,y,z

```
static bool goingUp = false;  
static int currentTone = 440;
```

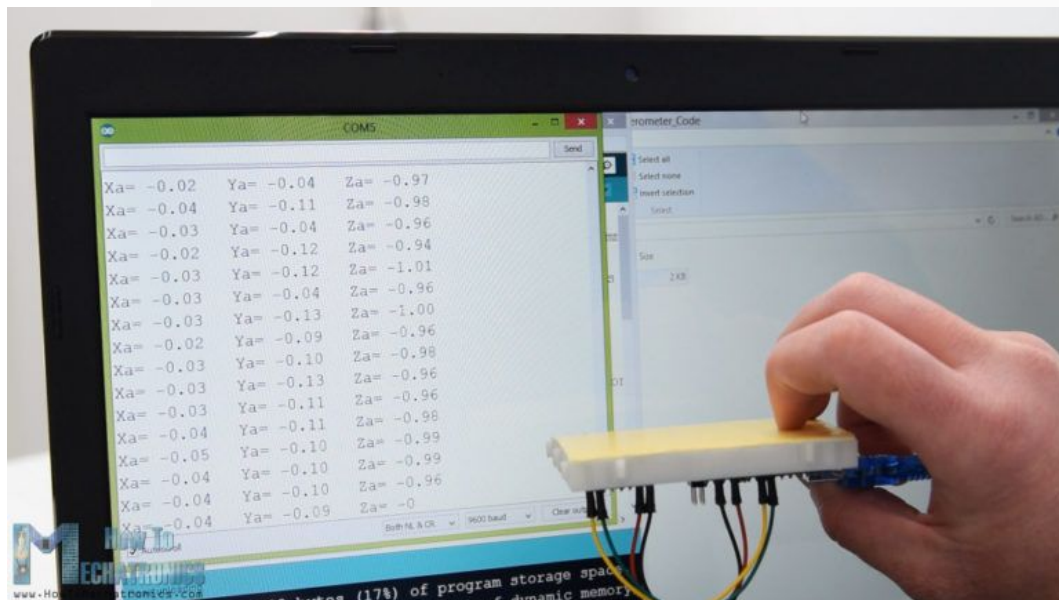
```
void setup() {
```

```
  // Initialize ADXL345.  
  Serial.begin(9600);  
  Wire.begin();  
  Wire.beginTransmission(ADXL345);  
  Wire.write(0x2d);  
  Wire.write(8);  
  Wire.endTransmission();  
  delay(10);
```

初始化ADXL345

```
  // Initialize buzzer.  
  pinMode(12, OUTPUT);  
  pinMode(13, OUTPUT);
```

```
}
```





# 實作

```
void loop() {  
  // Read accelerometer data.  
  Wire.beginTransmission(ADXL345);  
  Wire.write(0x32);  
  Wire.endTransmission(false);  
  Wire.requestFrom(ADXL345, 6, true);  
  
  prevX = x;  
  prevY = y;  
  prevZ = z;  
  
  x = (Wire.read() | Wire.read() << 8);  
  x /= 256;  
  y = (Wire.read() | Wire.read() << 8);  
  y /= 256;  
  z = (Wire.read() | Wire.read() << 8);  
  z /= 256;  
  
  if (abs(x - prevX) >= .2f || abs(y - prevY) >= .2f || abs(z - prevZ) >= .2f) {  
    digitalWrite(13, HIGH);  
    playAlarmSound(12);  
  } else {  
    digitalWrite(13, LOW);  
    stopAlarmSound(12);  
  }  
}
```

讀取新的x,y,z

# 實作

```
void loop() {  
  // Read accelerometer data.  
  Wire.beginTransmission(ADXL345);  
  Wire.write(0x32);  
  Wire.endTransmission(false);  
  Wire.requestFrom(ADXL345, 6, true);
```

```
  prevX = x;  
  prevY = y;  
  prevZ = z;
```

讀取前先把目前的 x, y, z 分別複製到 prevX, prevY, prevZ

```
  x = (Wire.read() | Wire.read() << 8);  
  x /= 256;  
  y = (Wire.read() | Wire.read() << 8);  
  y /= 256;  
  z = (Wire.read() | Wire.read() << 8);  
  z /= 256;
```

讀取新的x,y,z

```
  if (abs(x - prevX) >= .2f || abs(y - prevY) >= .2f || abs(z - prevZ) >= .2f) {  
    digitalWrite(13, HIGH);  
    playAlarmSound(12);  
  } else {  
    digitalWrite(13, LOW);  
    stopAlarmSound(12);  
  }  
}
```



# 實作

```
void loop() {  
  // Read accelerometer data.  
  Wire.beginTransmission(ADXL345);  
  Wire.write(0x32);  
  Wire.endTransmission(false);  
  Wire.requestFrom(ADXL345, 6, true);
```

```
  prevX = x;  
  prevY = y;  
  prevZ = z;
```

讀取前先把目前的 x, y, z 分別複製到 prevX, prevY, prevZ

```
  x = (Wire.read() | Wire.read() << 8);  
  x /= 256;  
  y = (Wire.read() | Wire.read() << 8);  
  y /= 256;  
  z = (Wire.read() | Wire.read() << 8);  
  z /= 256;
```

讀取新的x,y,z

```
  if (abs(x - prevX) >= .2f || abs(y - prevY) >= .2f || abs(z - prevZ) >= .2f) {  
    digitalWrite(13, HIGH);  
    playAlarmSound(12);  
  } else {  
    digitalWrite(13, LOW);  
    stopAlarmSound(12);  
  }
```

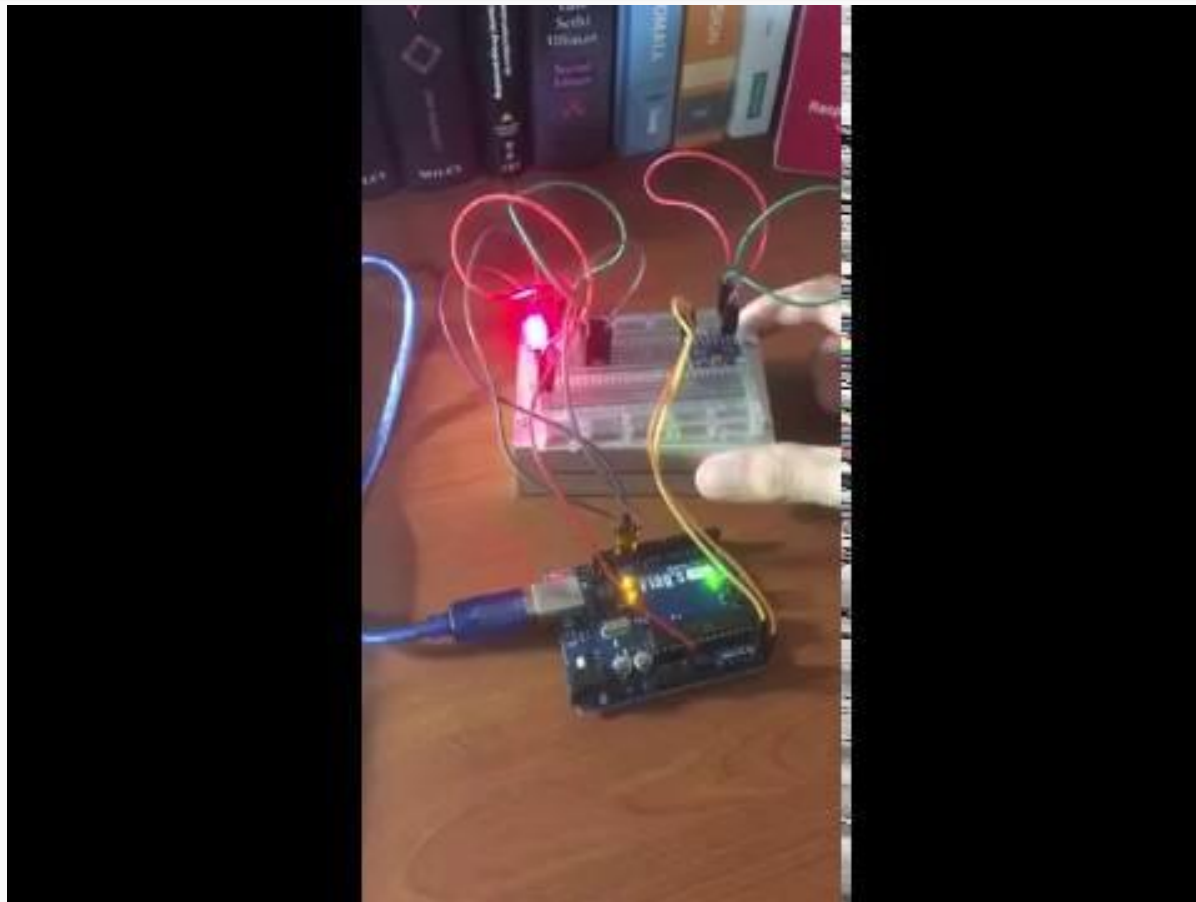
如果有晃動就讓LED發光並呼叫playAlarmSound();, 否則就熄燈並停止蜂鳴器

# 實作

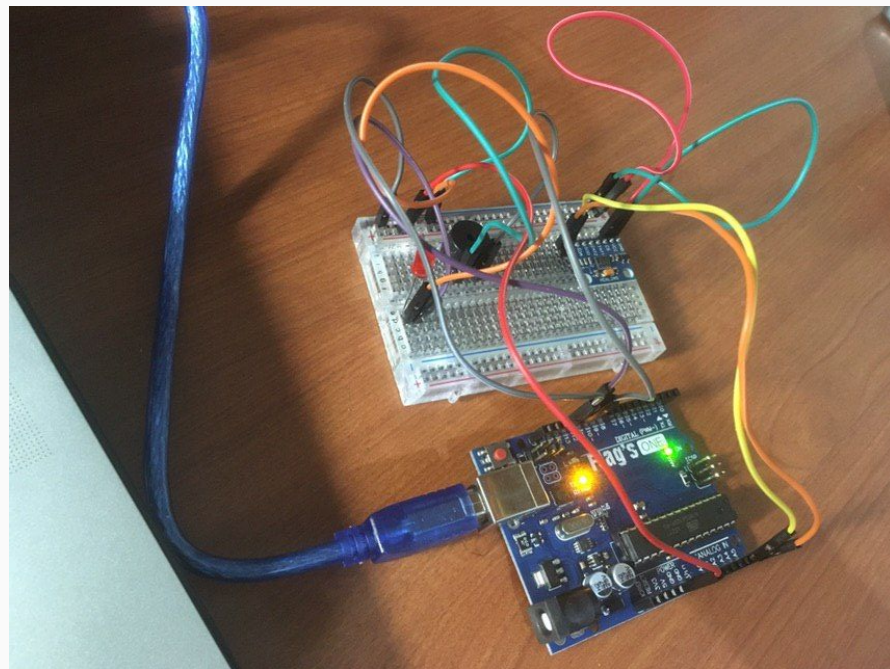
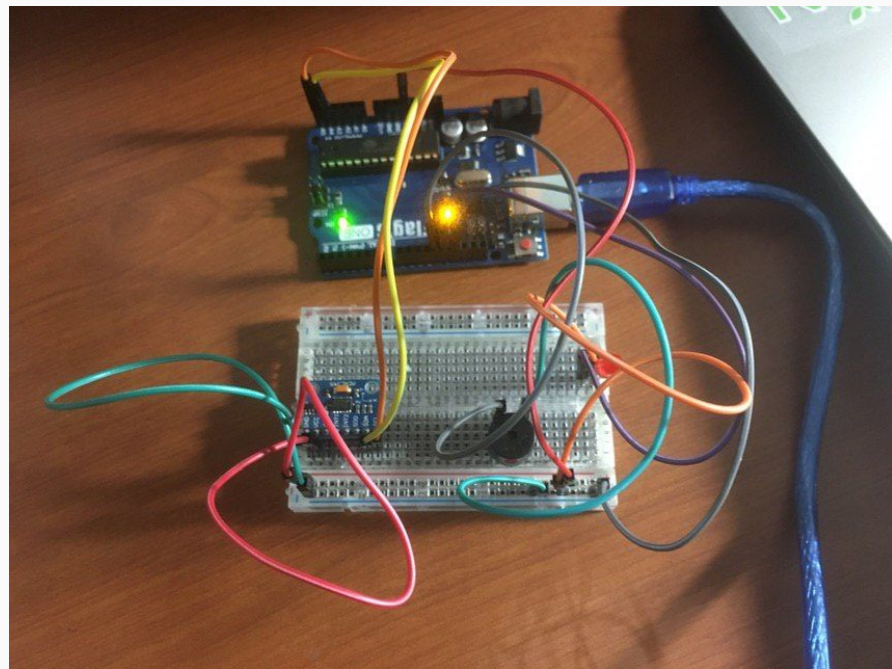
```
void playAlarmSound(int speakerPin) {  
    if (currentTone > 1000) {  
        goingUp = false;  
    } else if (currentTone < 440) {  
        goingUp = true;  
    }  
  
    if (goingUp) {  
        currentTone += 50;  
    } else {  
        currentTone -= 50;  
    }  
  
    tone(speakerPin, currentTone, 50);  
}  
  
void stopAlarmSound(int speakerPin) {  
    noTone(speakerPin);  
}
```

讓蜂鳴器的頻率在 440Hz~1000Hz間來回

# Demo



# 照片



# 結語

## 感想：

以前我沒接觸過 Arduino，也沒接觸過硬體的實作，但是上完這學期的課之後已對 Arduino 有初步的理解（線路接法、程式寫法）。做完這學期的專題覺得獲益良多，能獨立做出一個簡單的小東西還蠻有成就感的！

## 遭遇到的問題及解決辦法：

一開始完全沒有頭緒，不知道從哪裏下手。找到一篇教學之後，發現 ADXL345 感測器可以在 `loop()` 中不斷讀取 `x,y,z` 的值，這時就想到可以把他們和上一次的 值比較，如果  $\text{abs}(x1-x2) \geq k$  的話就讓他發光鳴笛。發光鳴笛的部分正好是之前作業常練習到的功能。

## 分工及努力程度：

實作+電路圖+測試+投影片，90分

報告完畢