**電通二甲微處理器實驗 實驗結報**

|  |  |  |  |
| --- | --- | --- | --- |
| **實驗名稱** | 中斷控制與超音波測距 | | |
| **組別** | **11** | **組員** | **蔣毓哲 尤竣賢** |

1. **實驗目的**

Arduino 於 當按下 Pin 2 外部中斷 0 時, 讀入超音波測距 之值並顯示於 PC上

1. 如何讀取超音波測距之值

2. 如何將超音波測距之值顯示於 LCD?

3. Arduino 如何規劃外部中斷 INT0?

4. 接一 SW, 當 SW 按下時暫停所有中斷, 實驗結果又如 何?

1. **實驗步驟**

1.讀取超音波測距之值並顯示在 LCD 上

2.修改 int0() 程式碼 ，於 int0 中執行超音波讀值，並將結果顯示在 LCD 上

3.觀察當 D5 按下時，按下 INT0 時之現象

按下:不接受中斷，INT0 不動作

放開: 重新接受中斷， INT0 有反應

1. **程式碼**

**1.**

#include <Ultrasonic.h>

#define TRIGGER\_PIN 12

#define ECHO\_PIN 13

Ultrasonic ultrasonic(TRIGGER\_PIN, ECHO\_PIN);

void setup()

{

Serial.begin(9600);

}

void loop()

{

float cmMsec, inMsec;

long microsec = ultrasonic.timing();

cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM); // 計算距離，單位: 公分

inMsec = ultrasonic.convert(microsec, Ultrasonic::IN); // 計算距離，單位: 英吋 Serial.print("MS: ");

Serial.print(microsec);

Serial.print(", CM: ");

Serial.print(cmMsec);

Serial.print(", IN: ");

Serial.println(inMsec);

delay(1000);

}

2.

const byte intPin=2; //interrupt pin

const byte ledPin=13; //built-in LED

volatile boolean state=LOW; //initial value of switch pin

void setup()

{

pinMode(ledPin, OUTPUT);

pinMode(intPin, INPUT\_PULLUP); //enable pull-up resistor of input pin digitalWrite(ledPin, ledState); //set LED OFF

attachInterrupt(0, int0, LOW); //assign int0

}

void loop()

{

if (state)

{

digitalWrite(ledPin, HIGH);

} //turn LED on

else

{

digitalWrite(ledPin, LOW);

} //turn LED off

}

void int0()

{

//interrupt handler state=!state; //reverse state

}

3.

void loop()

{

boolean a= digitalRead(7);

if(a==LOW)

noInterrupts();

else

interrupts();

}

1. **實驗結果及分析**

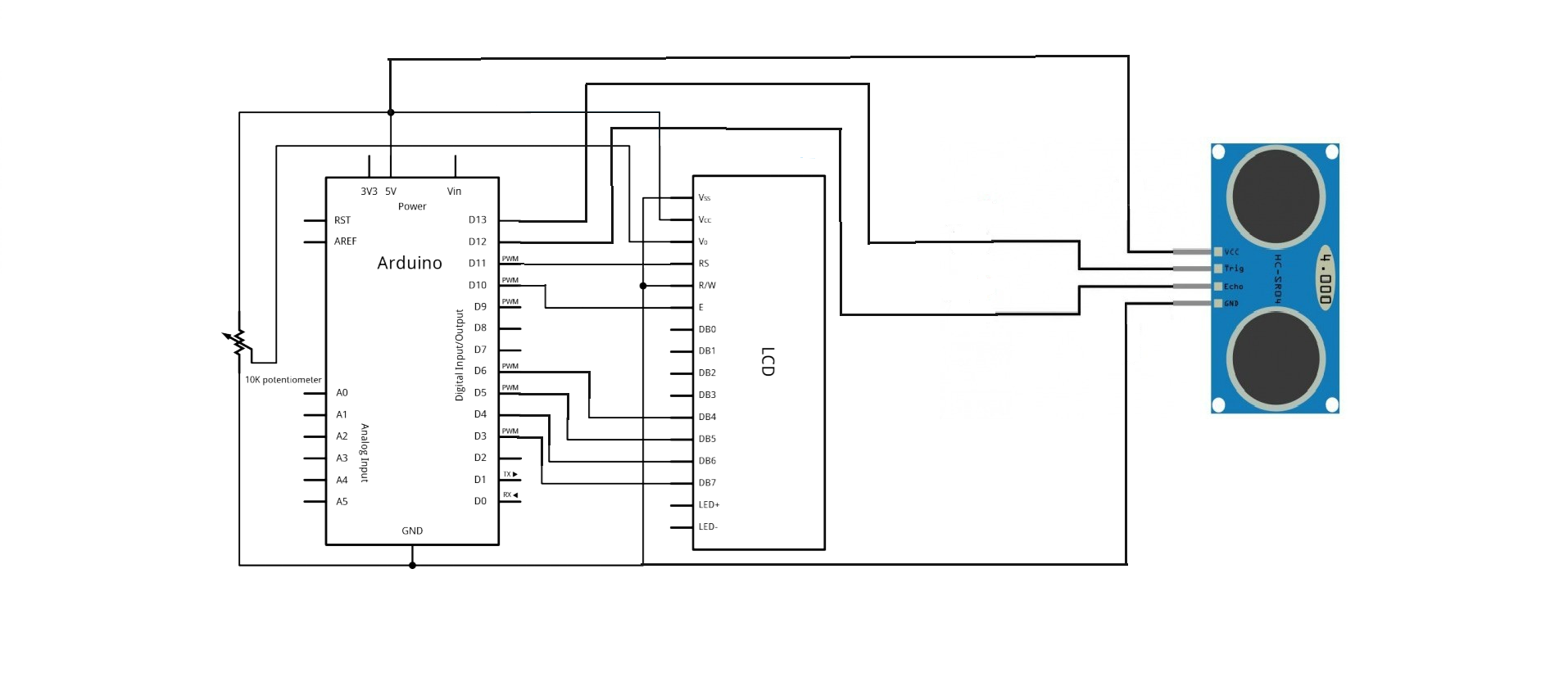
這次實驗結合LCD與超音波，測試發現太遠會測試不到，沒有數值

1. **心得討論**

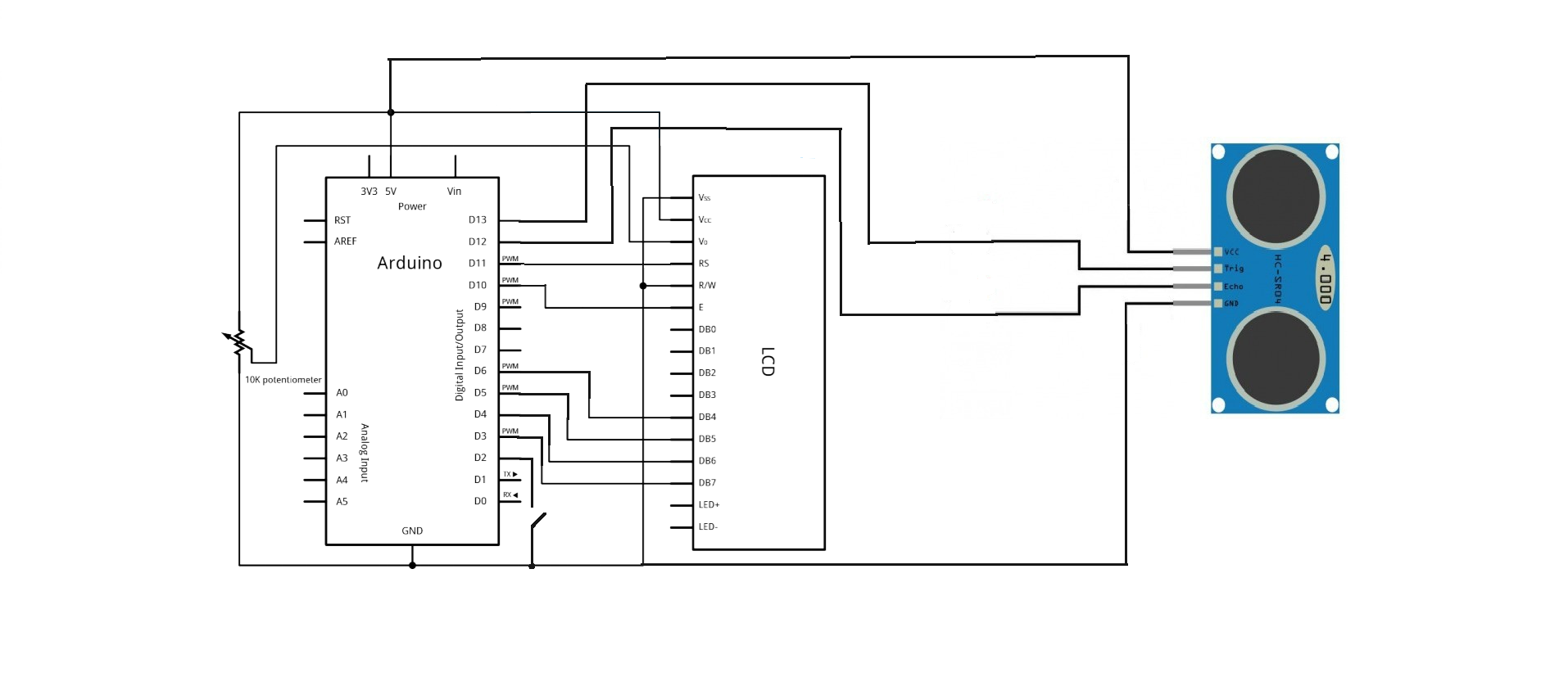
這次實驗沒有遇到太大困難，唯一有困難的還是接線，總是不確定是接線錯還是程式有錯

1. **修正電路圖**

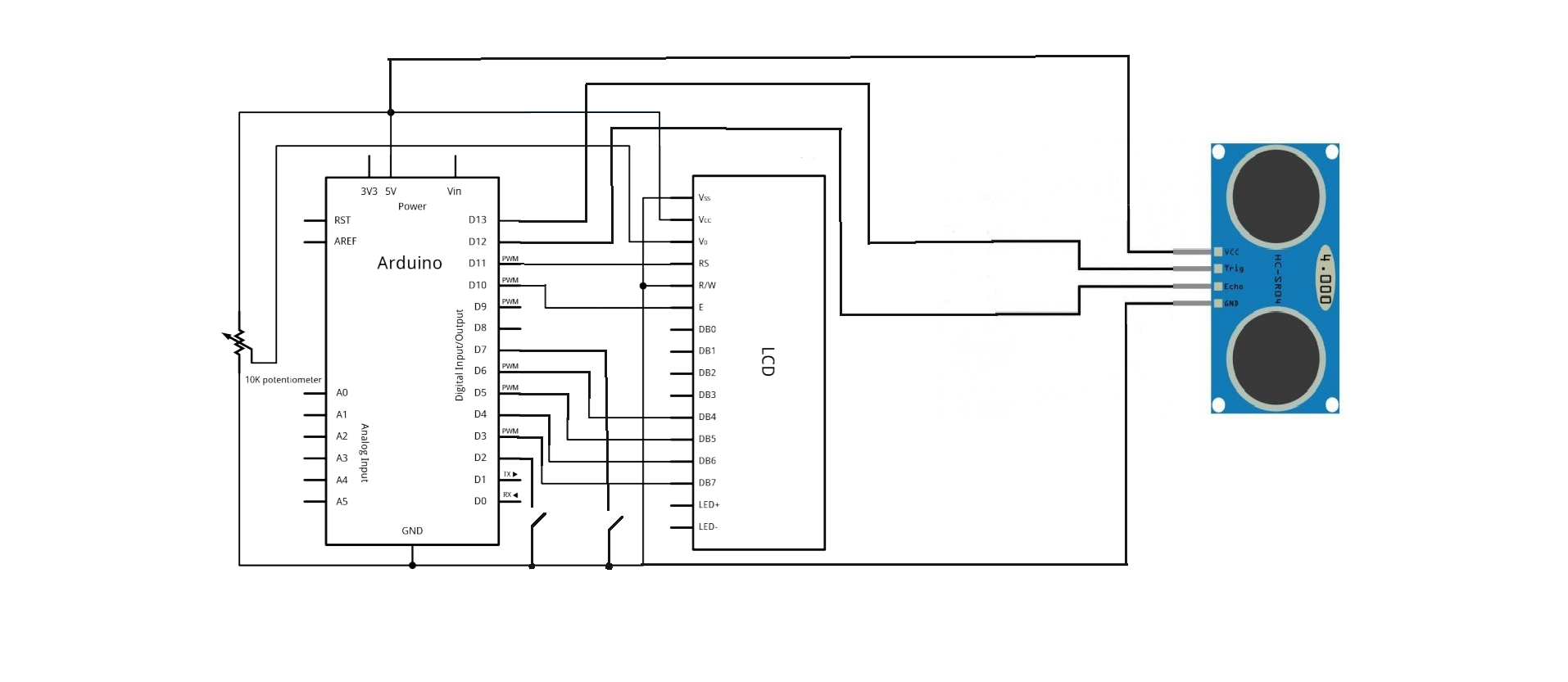
**1.**



**2.**



**3.**



1. **修正程式碼**

**1.**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(11, 10, 6, 5, 4, 3);**

**#include <Ultrasonic.h>**

**#define TRIGGER\_PIN 12**

**#define ECHO\_PIN 13**

**Ultrasonic ultrasonic(TRIGGER\_PIN, ECHO\_PIN);**

**void setup()**

**{**

**Serial.begin(9600);**

**lcd.begin(16, 2);**

**}**

**void loop()**

**{**

**String val;**

**lcd.setCursor(0,0);**

**float cmMsec, inMsec;**

**long microsec = ultrasonic.timing();**

**cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);**

**inMsec = ultrasonic.convert(microsec, Ultrasonic::IN);**

**lcd.print("MS: ");**

**lcd.print(microsec);**

**delay(1000);**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print(", CM: ");**

**lcd.print(cmMsec);**

**lcd.setCursor(0,1);**

**lcd.print(", IN: ");**

**lcd.print(inMsec);**

**delay(1000);**

**}**

**2.**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(11, 10, 6, 5, 4, 3); // initialize interface pins**

**#include <Ultrasonic.h>**

**#define TRIGGER\_PIN 12**

**#define ECHO\_PIN 13**

**const byte intPin=2; //interrupt pin**

**Ultrasonic ultrasonic(TRIGGER\_PIN, ECHO\_PIN);**

**void setup()**

**{**

**Serial.begin(9600);**

**lcd.begin(16, 2);**

**pinMode(intPin, INPUT\_PULLUP);**

**attachInterrupt(0, int0, LOW);**

**}**

**void loop()**

**{**

**}**

**void int0()**

**{**

**lcd.setCursor(0,0);**

**float cmMsec, inMsec;**

**long microsec = ultrasonic.timing();**

**cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);**

**inMsec = ultrasonic.convert(microsec, Ultrasonic::IN);**

**lcd.print("MS: ");**

**lcd.print(microsec);**

**delay(1000);**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print(", CM: ");**

**lcd.print(cmMsec);**

**lcd.setCursor(0,1);**

**lcd.print(", IN: ");**

**lcd.print(inMsec);**

**delay(1000);**

**}**

**3.**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(11, 10, 6, 5, 4, 3); // initialize interface pins**

**#include <Ultrasonic.h>**

**#define TRIGGER\_PIN 12**

**#define ECHO\_PIN 13**

**const byte intPin=2; //interrupt pin**

**const byte intPin2=7;**

**Ultrasonic ultrasonic(TRIGGER\_PIN, ECHO\_PIN);**

**void setup()**

**{**

**Serial.begin(9600);**

**lcd.begin(16, 2);**

**pinMode(intPin, INPUT\_PULLUP);**

**pinMode(intPin2, INPUT\_PULLUP);**

**attachInterrupt(0, int0, LOW);**

**}**

**void loop()**

**{**

**boolean a= digitalRead(7);**

**if(a==LOW)**

**noInterrupts();**

**else**

**interrupts();**

**}**

**void int0()**

**{**

**lcd.setCursor(0,0);**

**float cmMsec, inMsec;**

**long microsec = ultrasonic.timing();**

**cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);**

**inMsec = ultrasonic.convert(microsec, Ultrasonic::IN);**

**lcd.print("MS: ");**

**lcd.print(microsec);**

**delay(1000);**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print(", CM: ");**

**lcd.print(cmMsec);**

**lcd.setCursor(0,1);**

**lcd.print(", IN: ");**

**lcd.print(inMsec);**

**delay(1000);**

**Serial.print(microsec);**

**}**