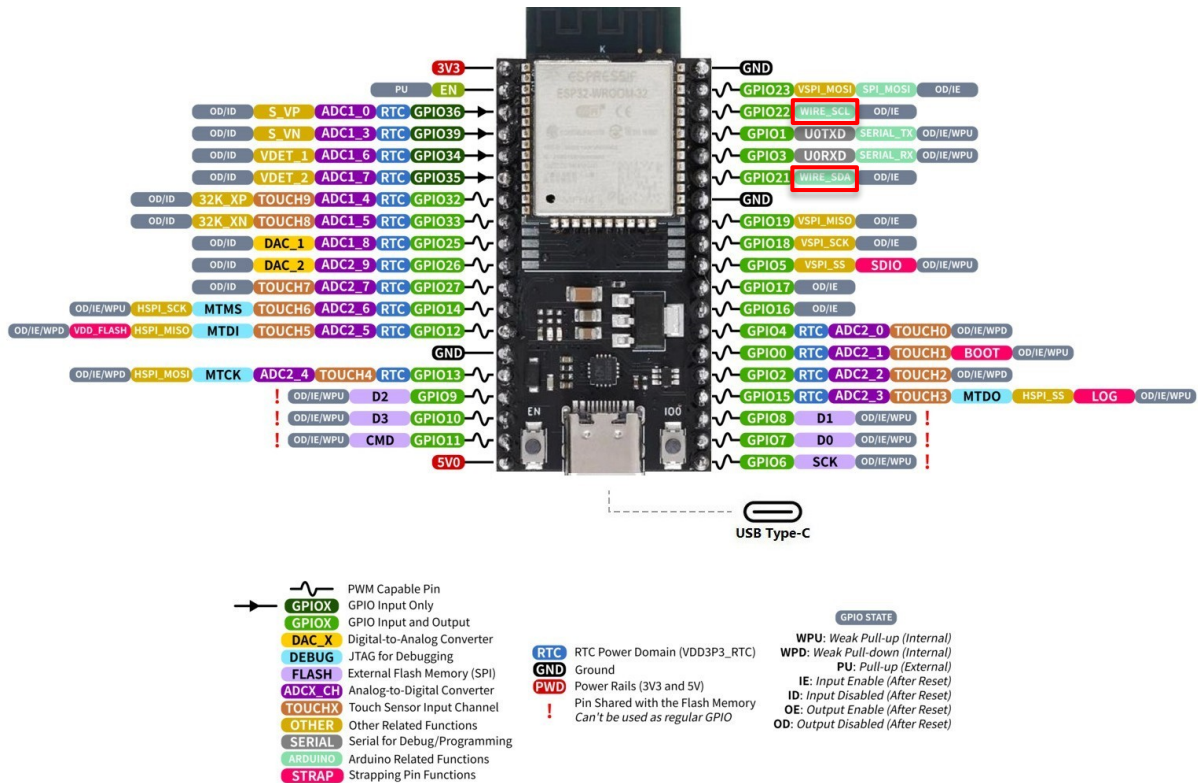


## 單元 3-1：LCD 顯示器 16X2 練習

### 學習目標：

- 了解 IIC (I<sup>2</sup>C) (Inter-Integrated Circuit) 通訊協定
- 了解 LiquidCrystal\_I2C 物件與指令運用(於範例註釋說明)

### ESP32 接腳圖：



### 材料需求：

ESP32 開發板	X1
LCD16X2	X1
電阻 220Ω	X2
LED	X2

### 設備需求：

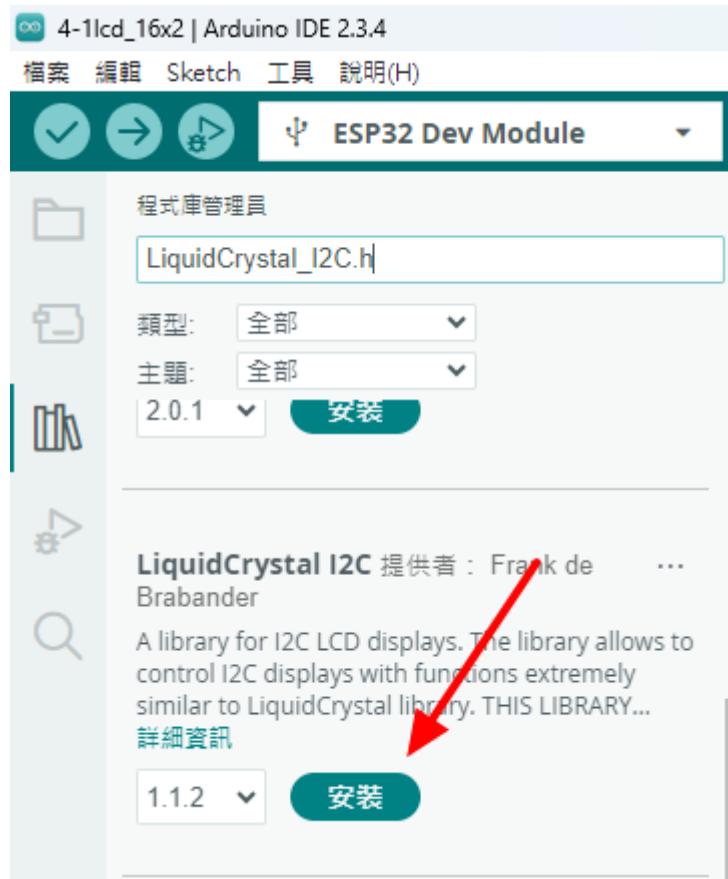
三用電表	X1
------	----

I2C，又稱 I<sup>2</sup>C(Inter-Integrated Circuit)，在 I2C 的通訊協定中，收發資料只單純靠兩條線就能完成，分別為 SCL(serial clock) 以及 SDA(serial data)，

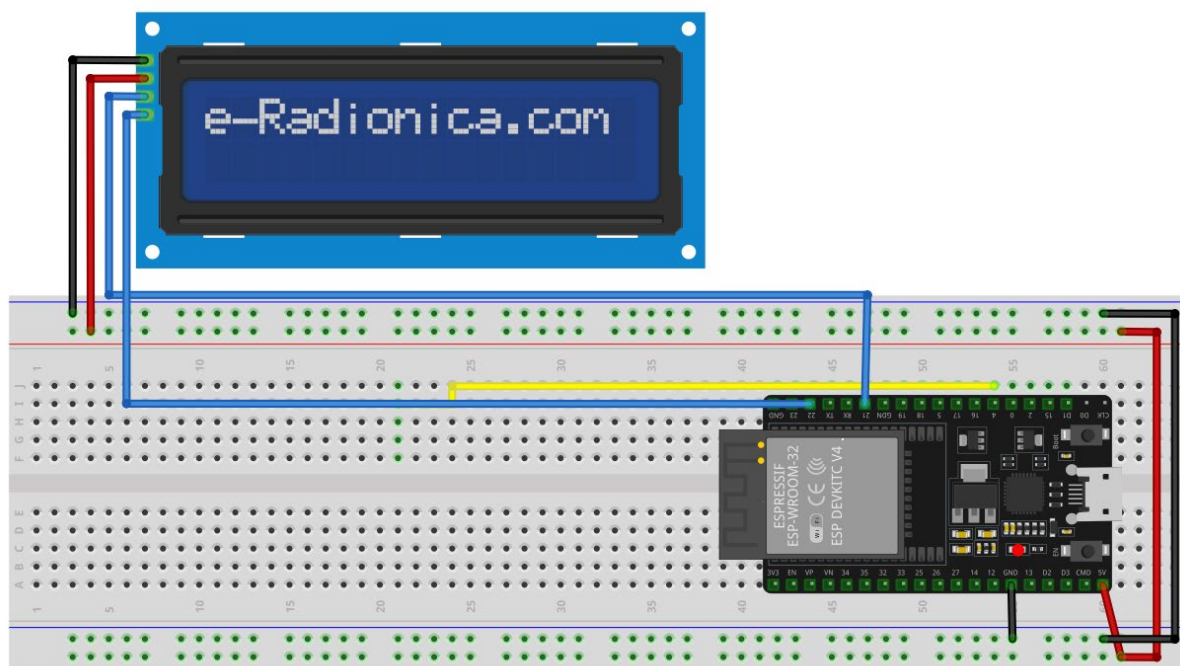
### 指令說明：

將隨著程式的編寫，逐步說明指令功能，請大家循序漸進學習。

至工具/管理程式庫/搜尋 LiquidCrystal\_I2C.h/下載安裝



接線圖：LCD：GND 接 -、Vcc 接 +、SDA 接 GPIO21、SCL 接 GPIO22



程式內容(一)LCD 16X2

```

#include <LiquidCrystal_I2C.h>      //匯入相關資料庫<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);
    //設定 LCD 位址為 0x27(利用背板 A0~A2 控制) 為 20X4 或 16X2 的字幕顯示

int i;

void setup()
{
    lcd.init();                    // 初始設定
    lcd.begin(16, 2);              // 初始化 LCD，一行 16 的字元，共 2 行，預設開啟背光

    for(int i = 0; i < 3; i++)
    {
        lcd.backlight(); // 開啟背光
        delay(250);
        lcd.noBacklight(); // 關閉背光
        delay(250);
    }
    lcd.backlight();              //設定背光

    lcd.setCursor(3,0);           //設定位置第(4,0)游標顯示(第 3 個字，第 1 行)
    lcd.print("Hello, world!");   //顯示字串"Hello, world!"
    lcd.setCursor(3,1);           //設定位置第(4,1)游標顯示(第 3 個字，第 2 行)
    lcd.print("I Love ESP32!");   //顯示字串"Show Arduino!"
    delay(2000);
}

void loop()
{
    lcd.clear();                  //清除螢幕
    lcd.print("Monday");          //無設定游標位置由(0,0)顯示
    delay(2000);
}

```

```
lcd.clear();
lcd.print("13:45");
delay(2000);

lcd.cursor();           //顯示底線游標接續文字後
delay(2000);
lcd.noCursor();        //不顯示底線游標
delay(2000);

lcd.display();
delay(2000);
lcd.noDisplay();       //不是 clear 只是螢幕不顯示
delay(2000);
lcd.display();         //螢幕顯示
delay(2000);

lcd.blink();           //實心長方形游標閃爍
delay(5000);
lcd.noBlink();
delay(2000);

lcd.scrollDisplayRight(); //右位移一個位置
delay(1500);
lcd.scrollDisplayLeft();  //右位移一個位置
delay(1500);
}
```

---

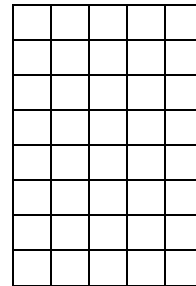
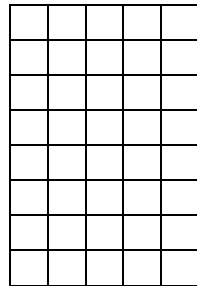
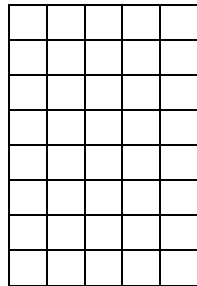
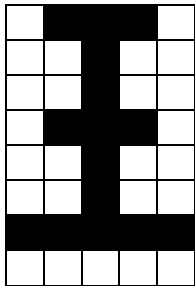
整合練習一：

1. 運用 for 迴圈讓你的英文名字，每秒右移一個位置移動五次後每秒左移一個位置移動五次。

2. 加入字元陣列顯示字元

```
char str[]={ 'T', 'e', 'm', 'p', '=', '18', 0xDF};  
lcd.print(str); //無設定游標位置由(0,0)顯示
```

3.參考下列程式製作自己的暱稱



---

```
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27,16,2); //lcd 為物件，LiquidCrystal_I2C 程式  
庫名稱
```

```
byte
```

```
yearChar[8]={B01000,B11111,B00010,B01111,B01010,B11111,B00010,B00000}; //設定  
yearChar[8]位元組陣列[0-7]
```

```
byte moonChar[8]={B01111,B01001,B01111,B01001,B01111,B01001,B11101,B00000};
```

```
byte dayChar[8]={B01111,B01001,B01001,B01111,B01001,B01001,B01111,B00000};
```

```
byte wang[8]={B01110,B00100,B00100,B01110,B00100,B00100,B11111,B00000};
```

```
int years=2021;
```

```
byte moons=1;
```

```
byte days=7;
```

```
void setup()
```

```
{
```

```
  lcd.init();
```

```
  lcd.backlight();
```

```

    lcd.createChar(0,yearChar); //創建字元 yearChar，放入 byte(0)
    lcd.createChar(1,moonChar); //創建字元 moonChar，放入 byte(1)，以下同上說
明
    lcd.createChar(2,dayChar);
    lcd.createChar(3,wang);
}
void loop()
{
    lcd.setCursor(0,0); //於顯示於(0,0)位置
    lcd.print(years); //顯示變數值
    lcd.write(byte(0)); //呈現 byte(0)的字元，以下同上說明
    lcd.print(moons);
    lcd.write(byte(1));
    lcd.print(days);
    lcd.write(byte(2));

    lcd.setCursor(3,1);
    lcd.print("name:");
    lcd.write(byte(3));
    lcd.print((char)0x3F); //利用(char)0x3F 呈現 ASCII 編碼'?'
    lcd.print((char)0x3E); //利用(char)0x3E 呈現 ASCII 編碼'>'
}

```

---

整合練習二：

運用 4-1、5-1 將溫度顯示於 LCD 上

