UNIT

10 超音波距離感測器實習

Arduino

10-1 實習目的

本單元實習目的，主要在介紹如何運用超音波距離感測器。實習內容是透過 Arduino 主板的序列埠 TX 與 RX 接腳，讀取超音波距離感測器偵測到的距離。並將之呈現在 Arduino IDE 監控視窗中，方便讀者觀察超音波距離感測器之實際效果。

在 Arduino 程式練習方面，本單元除將用到之前第 7 章已學過的 `Serial.begin()`、`Serial.print()` 與 `Serial.println()` 外，還將介紹序列埠傳輸的其他新函式，包括：`Serial.available()`、`Serial.read()` 以及 `Serial.write()`。

本單元影片請掃描本頁之 QR 碼，供讀者於實作時參考並確認實習結果是否正確。

本單元實習影片

10-2 材料介紹：超音波距離感測器

圖 10.1 為本實習主角－超音波距離感測器的實體照片，其型號為 US-100 Y4U1。超音波距離感測器是藉由發送超音波與接收到反射後的時間間距，計算出發射點與物體間之距離。圖 10.2 為其腳位與元件說明，其中：

- 1 號標籤為 GND 接腳。
- 2 號標籤亦為 GND 接腳。
- 3 號標籤 Echo/RX 為序列埠資料輸出接腳，需與 Arduino 主板 RX 接腳連接。
- 4 號標籤 Trig/TX 為序列埠資料輸入接腳。需與 Arduino 主板 TX 接腳連接。
- 5 號標籤 VCC 為本元件之電源接腳。
- 6 號標籤與 7 號標籤為超音波收發器，須對準欲測量距離之物體。

請注意，本元件之序列埠 RX、TX 接腳，與 Arduino 主板一致。換言之，不是傳統的元件 TX 對接主板的 RX，元件 RX 對接主板的 TX；而是元件 TX 對接主板的 TX，元件 RX 對接主板的 RX。

另外，序列埠傳輸是以位元組 (byte) 為單位，之前實習使用的 `Serial.print()` 或 `Serial.println()` 函式，以傳送字元或字串為主，其格式通常為標準的 ASCII 碼，因此電腦端的 Arduino IDE 可以認得，不需再行處理。但是當利用序列埠傳輸數值時，在數值格式未確定下，通常是以一個位元組之二進位形式來記錄數值。換言之，一個位元組含 8 個位元，因此該位元組之最小值為 0，最大只能是 255。若要傳輸大於 255 以上的數值，則需要用到兩個位元組。當接收端收到這些位元組後，需要自己重新組合成對應數值。

本單元使用的超音波距離感測器，其量測數值大於 255。因此感測器之序列埠傳輸，需用到兩個位元組，稱為高位元組 HighByte 以及低位元組 LowByte。接收端收到此兩位元組後，會先將 HighByte 與 LowByte 轉換成 0 到 255 間的數值，再透過以下公式計算出量測距離：

距離 = HighByte 數值 × 256 + LowByte 數值

上述計算必須在設計程式時正確實作，否則會誤判距離。我們會在 10-4 節中再行解說程式部分。

另外，本實習使用的超音波距離感測器之序列埠傳輸速率，應設為 9600 baud。同時在使用前，需傳輸指令方可激活，此指令為英文字母中的大寫 "U"。本型號之超音波距離感測器，可偵測之距離為 20mm 到 4500mm，而夾角範圍則約 15 度。

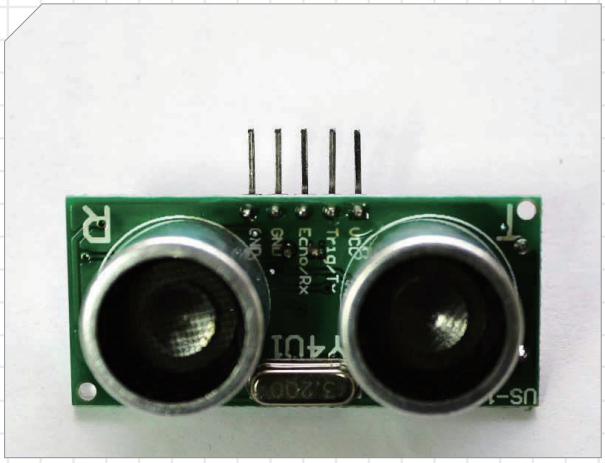

圖 10.1 超音波距離感測器實體照

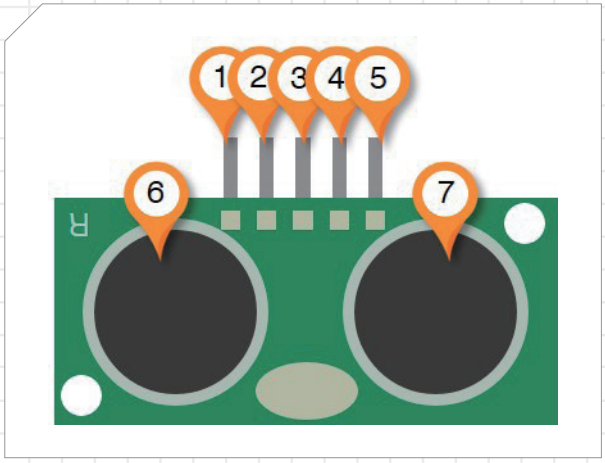

圖 10.2 超音波距離感測器元件與腳位介紹

10-3 電路設計與連線步驟

圖 10.3 為本次實習的電路設計步驟圖。其中，圖 (a) 與圖 (b) 分別是針對 Mega 主板與 UNO 主板的電路設計圖。圖中將電路連線步驟拆解為 7 個步驟，各步驟說明如下：

步驟 1： 將超音波距離感測器，插入麵包板 E 或 F 區之孔位，位置可任意選擇。

步驟 2： 利用單芯電線將超音波距離感測器 GND 接腳－即圖 10.2 中的 1 號或 2 號標籤，連接至麵包板 A 或 B 區之負極孔位。

步驟 3： 利用單芯電線將超音波距離感測器 Echo/RX 接腳－即圖 10.2 中的 3 號標籤，對應之直向孔位與主板 RX0 接腳連接。

步驟 4： 利用單芯電線將超音波距離感測器 Trig/TX 接腳－即圖 10.2 中的 4 號標籤，對應之直向孔位與主板 TX0 接腳連接。

步驟 5： 利用單芯電線將超音波距離感測器 VCC 接腳－即圖 10.2 中的 5 號標籤，連接至麵包板 C 或 D 區之正極孔位。

步驟 6： 如圖 10.3 所示，利用單芯電線將主板的 5V 接腳，連接至麵包板 C 或 D 區之正極孔位。

步驟 7： 如圖 10.3 所示，利用單芯電線將主板的 GND 接腳，連接至麵包板 A 或 B 區之負極孔位。

圖 10.2 中的 1 號標籤 GND 腳位於本實習中並未使用到，US-100 Y4U1 具有 2 組 GND 腳位，擇一使用即可。此外，當使用 Arduino IDE 經由 USB 上傳程式至 Arduino 主板時，是透過序列埠接腳傳輸資料。而超音波距離感測器，亦是透過序列埠傳輸資料，這將導致 Arduino IDE 開發環境無法上傳程式。因此，在完成與 IDE 連結上傳程式前，先不要連結步驟 3 與步驟 4 中的單芯電線，待程式上傳完成後再連接之。

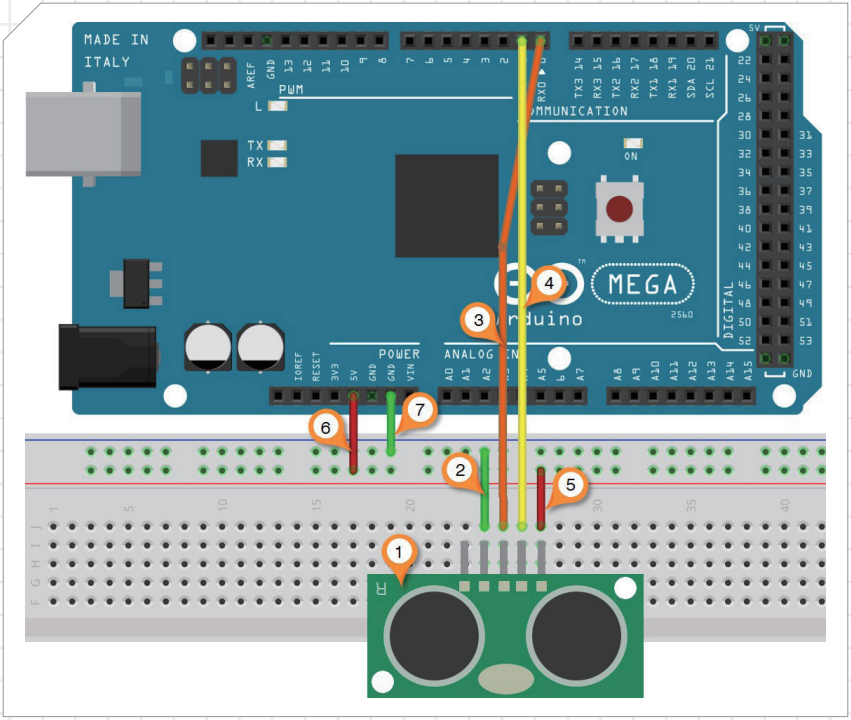

圖 10.3(a)
Mega 主板電路設計圖

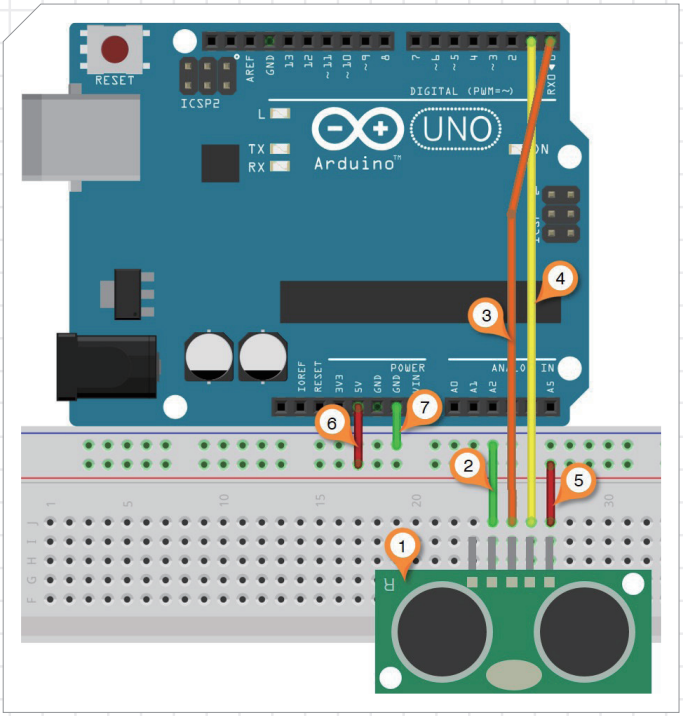

圖 10.3(b)
UNO 主板電路設計圖

10-4 使用函式說明

1 | int Serial.available()

此函式目的在確認序列埠收到資料。呼叫此函式後，將回傳目前於序列埠中的位元組個數，其資料型態為整數型態。

2 | Serial.read()

此函式目的在讀取序列埠已接收到的資料。每呼叫此函式一次，將讀取位於序列埠中的第 1 個位元組資料，同時此資料於讀取後，將被從序列埠中移除。換言之，此函式一次僅能讀取一個位元組資料，並且無法重複讀取該同一資料。

3 | Serial.write(value)

此函式目的在將 **value** 寫入序列埠，**value** 為一位元組。若此序列埠為電腦上的 Arduino IDE，則此位元組會以 ASCII 碼格式顯示於監控視窗上。與 **Serial.print** 最大不同處，在於其傳輸資料格式不同。**Serial.print()** 函式使用 ASCII 編碼，而 **Serial.write()** 函式，則是基本的二進制編碼。

10-5 程式碼解說

圖 10.4 為本實習的程式碼。首先於第 1 行與第 2 行程式，分別宣告 high_Byte 與 low_Byte 兩個整數型態的變數，用於儲存從序列埠接收到的兩個位元組。high_Byte 儲存從超音波距離感測器接收到高位元組資料；low_Byte 則用於儲存從超音波距離感測器接收到的低位元組資料。第 3 行程式另外宣告一個整數變數 distance，用於儲存經計算後所得的距離值。

第 4 至第 5 行程式為 setup() 函式。配合本單元使用之超音波距離感測器，應設定序列埠的傳輸速率為 9600 baud。同時本程式也透過同一序列埠傳送資料到 Arduino IDE 監控視窗，因此在監控視窗的右下方，也應選擇 9600 baud。

接下來是 loop() 函式。第 7 行程式目的，是為激活超音波距離感測器，乃透過 Serial.write() 從 Arduino 序列埠 TX 接腳寫入大寫 "U"。再於第 8 行程式中，以 delay() 函式使程式暫停 500 毫秒，等待回傳測量結果。

第 9 行程式，透過 if() 條件式，判斷超音波距離感測器回傳之資料是否為兩個位元組。若為 true，則於第 10 行以及 11 行程式，透過 Serial.read() 函式，依序讀取高位元組與低位元組資料，並分別儲存於 high_Byte 與 low_Byte 變數中。第 12 行程式則依照 10-2 節

之距離計算公式，計算出實際值並寫入變數 distance。最後之第 13 與 14 行程式，透過序列埠，將 distance 中儲存的數值，加上距離單位 mm 由 Arduino IDE 監控視窗顯示之。

此程式執行時，可在電腦上的 Arduino IDE 監控視窗內，不斷看到目前超音波距離感測器所偵測到的距離。上述實習結果請與本單元短片比較，確認是否一致。若一致，恭喜您順利完成本單元，並請前往下一單元。

```
1.   int high_Byte = 0;
2.   int low_Byte = 0;
3.   int distance = 0;
4.   void setup(){
5.     Serial.begin(9600); }
6.   void loop(){
7.     Serial.write("U");
8.     delay(500);
9.     if(Serial.available() == 2) {
10.      high_Byte = Serial.read();
11.      low_Byte = Serial.read();
12.      distance = high_Byte *256 + low_Byte;
13.      Serial.print(distance);
14.      Serial.println("mm");
15.    }
16.  }
```

圖 10.4 第 10 章程式碼