

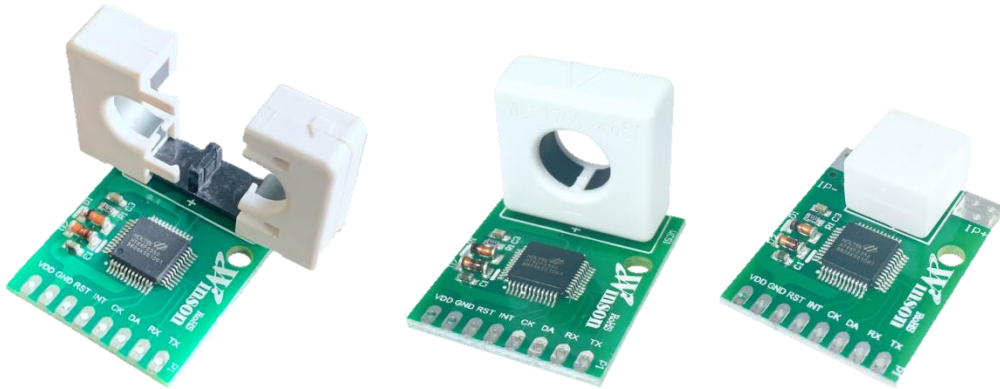
## WCM 電流感測模組使用說明 **2**

- 1. 通訊介面格式 **2**
- 2. 工作模式 **2**
- 3. 量測方式:(連續模式) **2**
- 4. 量測方式:(MODBUS-RTU) **3**
- 5. 應用電路 **5**
- 6. WCM-LCD4X32 **7**

## ARDUINO 範例程式 **9**

- 1. 使用 ARDUINO 平台須知 **9**
- 2. 連續模式 **10**
  - 接線示意圖 **10**
  - 接線圖 **11**
  - 軟體&程式 **12**

## WCM 電流感測模組使用說明



### 1. 通訊介面格式

通訊介面	UART TTL 接口 RS232 / RS485 (需搭配 RS232/RS485 信號轉換器)
通訊速率	9600 bps
通訊格式	Parity bit: None , Data bit: 8 , Stop bit: 1

### 2. 工作模式

(2.1)連續模式:輸出 8 個 byte,資料更新率 3 Data/sec，歸零重置時間 1 sec。

(2.2)Modbus-RTU 模式:採用主從傳送/響應方式通訊。

### 3. 量測方式:(連續模式)

(3.1) DC50C (直流量測) / 50C (交/直流量測)

因感測元件的鐵心殘磁會些微影響量測電流的精準度，**建議初次使用或量測電流方向改變時**，先通過一次欲量測的電流後，再於**零電流**的情況下手動歸零，可有效減少殘磁對量測電流的影響。

(3.2) AC50C (交流量測)

交流量測:上電在無電流通過感測器會自動歸零電流值，也可以透過重置接腳(RST)強制歸零電流值。

(3.3)歸零

在無電流通過時，透過重置接腳(RST)重新校準歸零電流值再進行量測，合理使用本功能，會使量測更加精準。在量測 DC 電流時，感測器會產生一定的剩磁量，若發現此現象造成小電流讀值不準請重新歸零。

Winson reserves the right to make changes to improve reliability or manufacturability.

## 4. 量測方式: (Modbus-RTU)

50M (Modbus-RTU): 上電透過"歸零重置指令"歸零電流值，依此基準量測電流。

(4.1) 讀取保持暫存器(功能碼:03H)

※廣播位址(0x00)無法執行讀取指令，且不會回傳異常碼。

(4.1.1) 電流

主機傳送: 01 03 00 02 00 02 65 CB

地址	命令	起始位址 (高位在前)	暫存器數 (高位在前)	校驗碼(CRC) (低位在前)
01H	03H	00H, 02H	00H, 02H	65H, CBH

從機響應: 01 03 04 00 00 04 D2 78 AE

地址	命令	資料長度	資料 (4 Bytes)	校驗碼(CRC) (低位在前)
01H	03H	04H	<b>00H, 00H, 04H, D2H</b>	78H, AEH

結果為: (01)感測器號碼為 1，(00 00 04 D2)電流=1234/1000 = 1.234A。

(4.1.2) 溫度

主機傳送: 01 03 00 04 00 02 85 CA

地址	命令	起始位址 (高位在前)	暫存器數 (高位在前)	校驗碼(CRC) (低位在前)
01H	03H	00H, 04H	00H, 02H	85H, CAH

從機響應: 01 03 04 00 00 01 2C FA 7E

地址	命令	資料長度	資料 (4 Bytes)	校驗碼(CRC) (低位在前)
01H	03H	04H	<b>00H, 00H, 01H, 2CH</b>	FAH, 7EH

結果為: (01)感測器號碼為 1，(00 00 01 2C)溫度=300/10 = 30.0°C。

(4.2) 寫入保持暫存器(功能碼:06H)

※廣播位址(0x00)可執行寫入指令，但不會響應。

(4.2.1) 歸零重置

主機傳送: 01 06 00 00 01 00 88 5A

從機響應: 01 06 00 00 01 00 88 5A

地址	命令	起始位址 (高位在前)	資料 (2 Bytes)	校驗碼(CRC) (低位在前)
01H	06H	00H, 00H	<b>01H, 00H</b>	88H, 5AH

結果為: (01)感測器號碼為 1，(01 00)寫入 256 歸零。

Winson reserves the right to make changes to improve reliability or manufacturability.

## (4.2.2) 寫入從機位址

主機傳送: 01 06 00 10 00 01 49 CF

從機響應: 01 06 00 10 00 01 49 CF

地址	命令	起始位址 (高位在前)	資料 (2 Bytes)	校驗碼(CRC) (低位在前)
01H	06H	00H, 10H	<b>00H, 01H</b>	49H, CFH

結果為: (01)感測器號碼為 1, 預設地址為 1, (00 01) **寫入位址 1**。

## (4.3) 異常碼

### (4.3.1) 功能碼異常

主機傳送: 01 01 00 00 00 00 3C 0A

地址	命令	起始位址 (高位在前)	暫存器數 (高位在前)	校驗碼(CRC) (低位在前)
01H	01H	00H, 00H	00H, 00H	3CH, 0AH

從機響應: 01 81 01 81 90

地址	命令	異常代碼	校驗碼(CRC) (低位在前)
01H	81H	<b>01H</b>	81H, 90H

結果為: (01)感測器號碼為 1, (81)=0X80(異常) + 0X01(功能碼), **(01)異常代碼**

### (4.3.2) 位址異常

主機傳送: 01 03 FF FF 00 04 44 2D

從機響應: 01 83 **02** C0 F1

結果為: (01)感測器號碼為 1, (83)=0X80(異常) + 0X03(功能碼), **(02)異常代碼**

### (4.3.3) 資料異常

主機傳送: 01 03 00 00 FF FF 44 7A

從機響應: 01 83 **03** 01 31

結果為: (01)感測器號碼為 1, (83)=0X80(異常) + 0X03(功能碼), **(03)異常代碼**

## ※位址重設

### (1) 廣播模式(0x00): 位址重設為 01

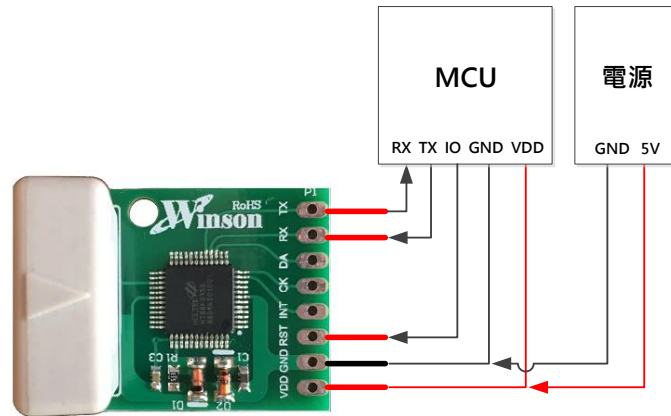
主機傳送: **00** 06 00 10 00 01 48 1E

從機響應: **無響應**

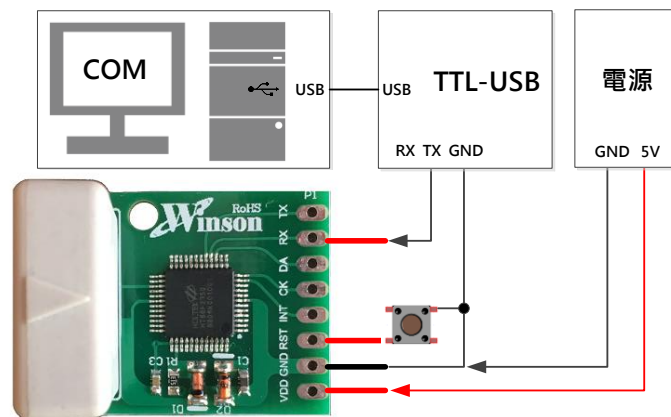
### (2) 外部接腳(INT)下拉到地: 位址重設為 01

## 5. 應用電路

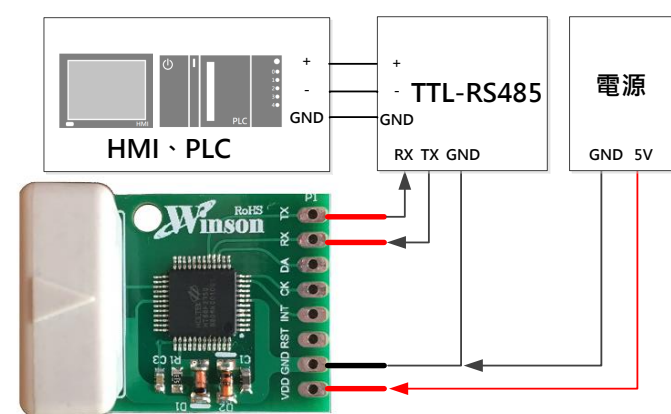
### (5.1) MCU 接線圖



### (5.2) TTL to USB 接線圖

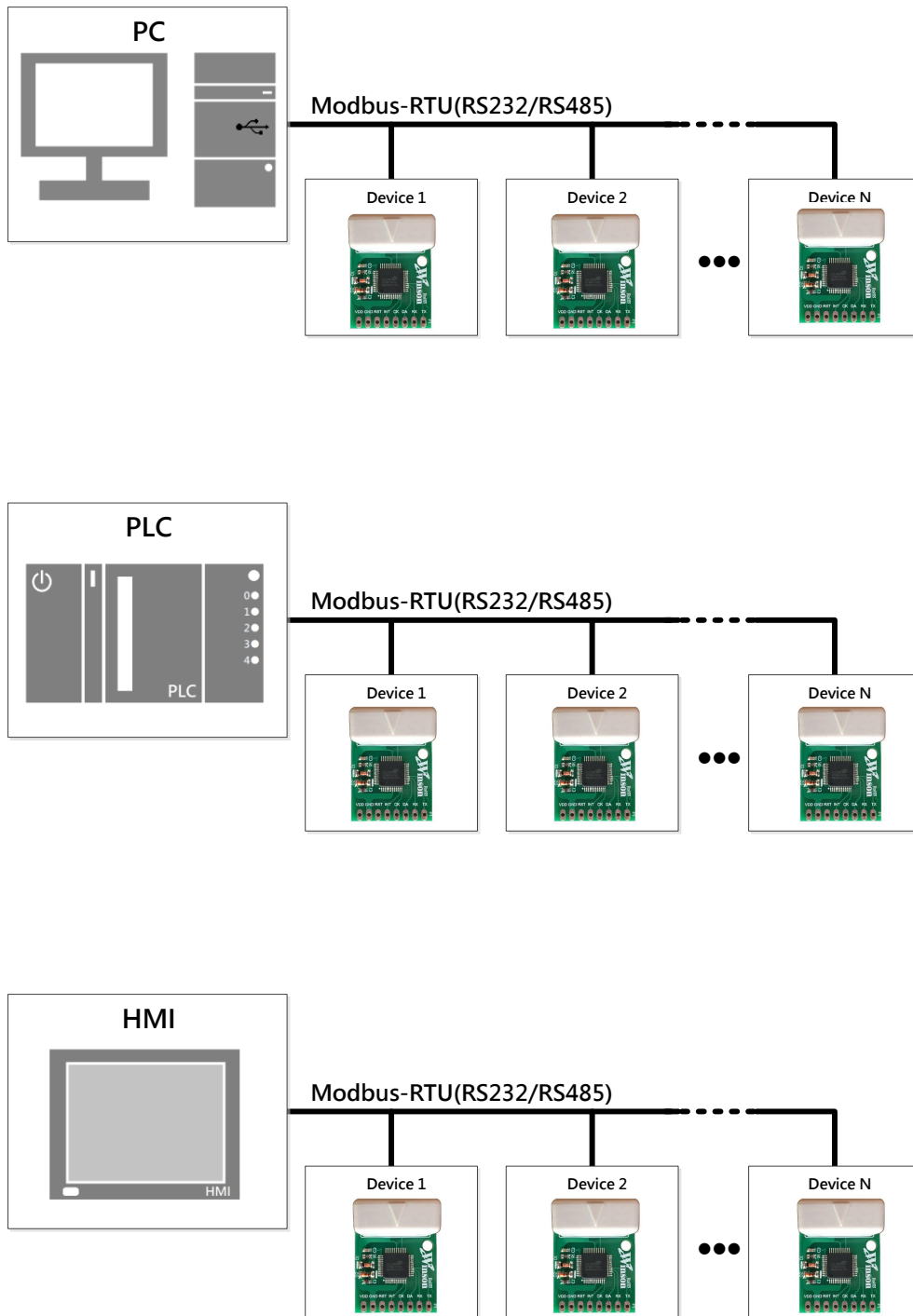


### (5.3) TTL to RS485 接線圖



Winson reserves the right to make changes to improve reliability or manufacturability.

## (5.4) Modbus-RTU 通訊架構圖



Winson reserves the right to make changes to improve reliability or manufacturability.

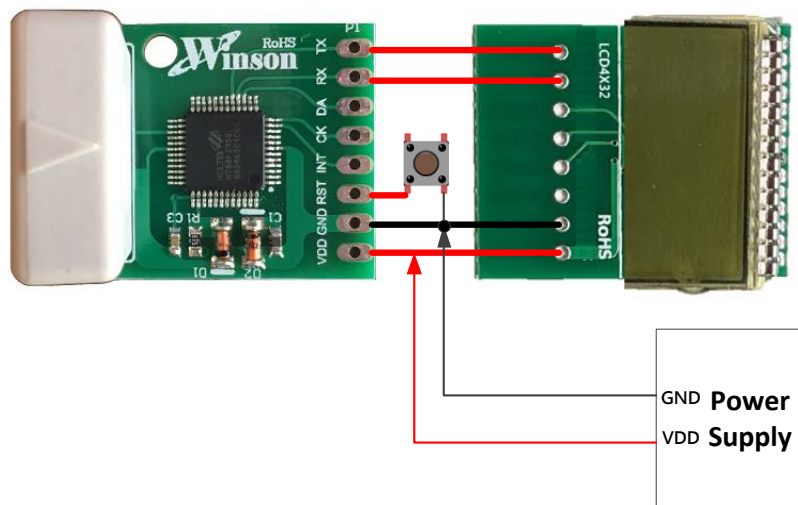
## 6. WCM-LCD4X32

為一 LCD 模組適於此電流感測模組

### (6.1) 腳位說明

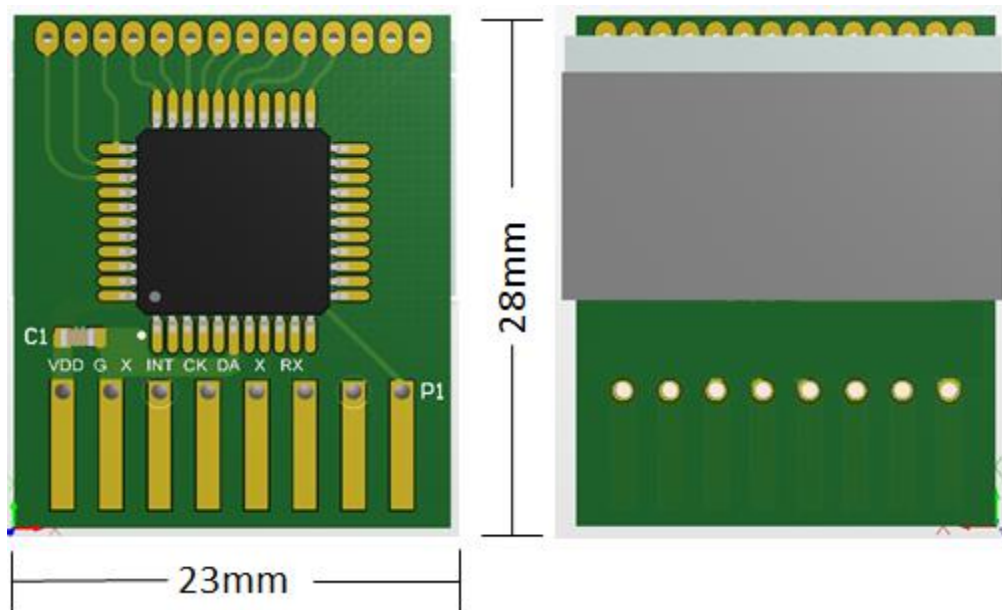
編號	名稱	I/O	說明
1	VDD	-	正電源輸入
2	GND	-	系統接地
3	x	-	預留腳位
4	x	-	預留腳位
5	CK	I/O	系統燒錄，預留腳位
6	DA	I/O	
7	x	-	預留腳位
8	RX	I	量測電流資料輸出，為 UART 通信，波特率為 9.6K bits/sec

### (6.2) LCM-LCD4X32 接線圖



Winson reserves the right to make changes to improve reliability or manufacturability.

## (6.3) 封裝資訊: (單位 mm)



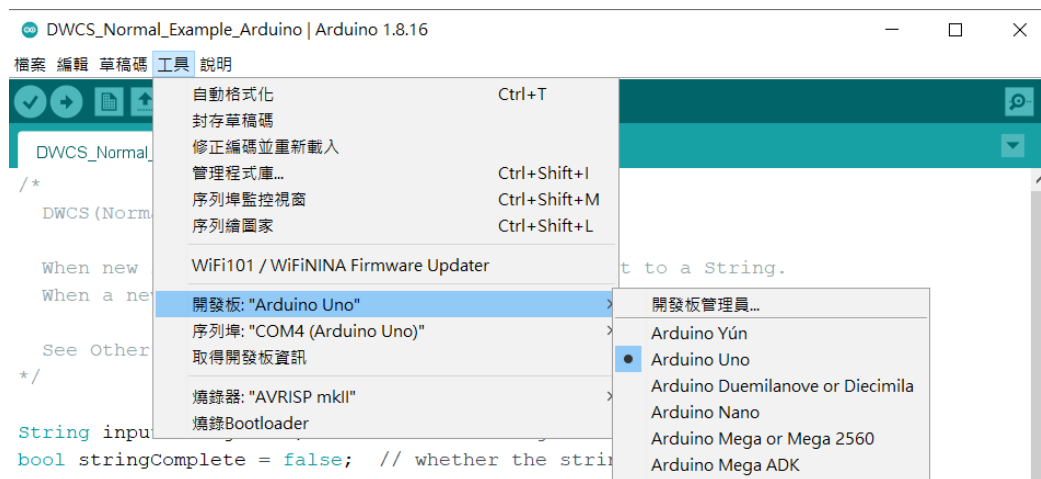
長 x 寬 x 高 = 28mm x 23mm x 8mm



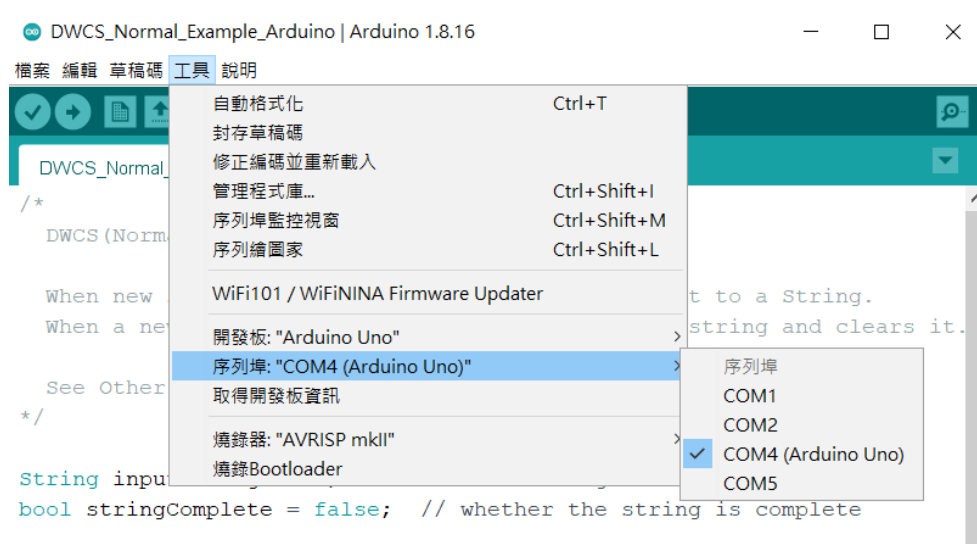
## Arduino 範例程式

### 1. 使用 Arduino 平台須知

#### (1.1)於 工具\_開發板: 選取 Arduino 的板別

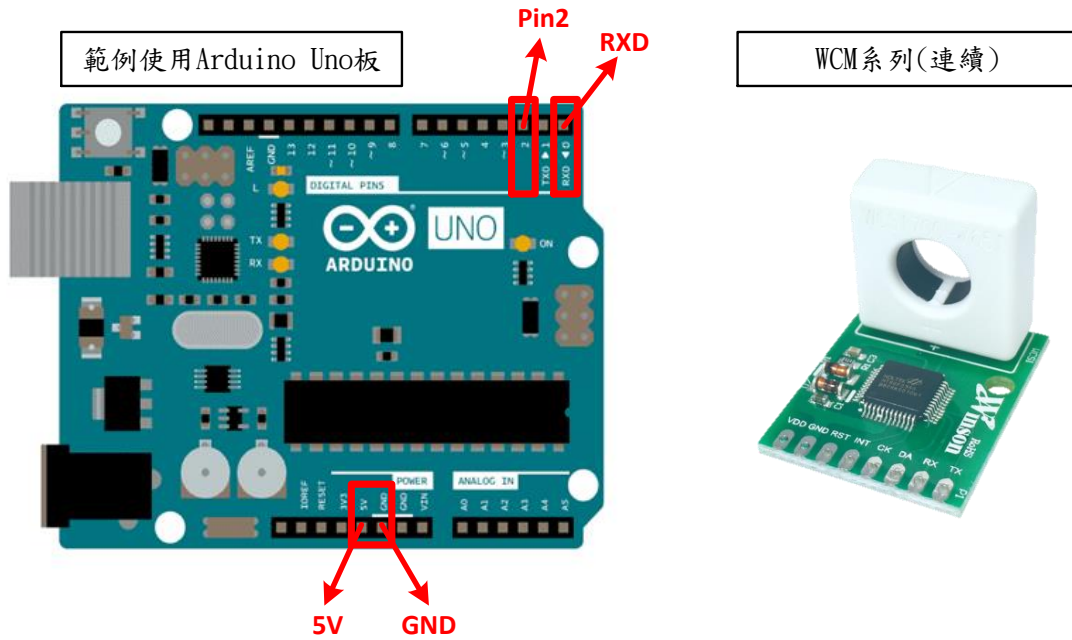


#### (1.2)於 工具\_序列埠: 選取 Arduino 的序列埠位置

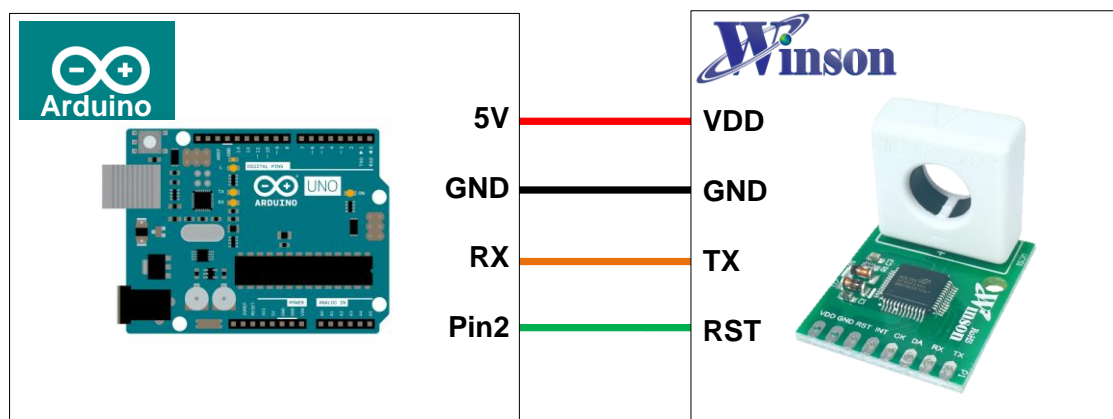


※Arduino 序列埠位置可使用 Windows 的裝置管理員查詢。

## 2. 連續模式

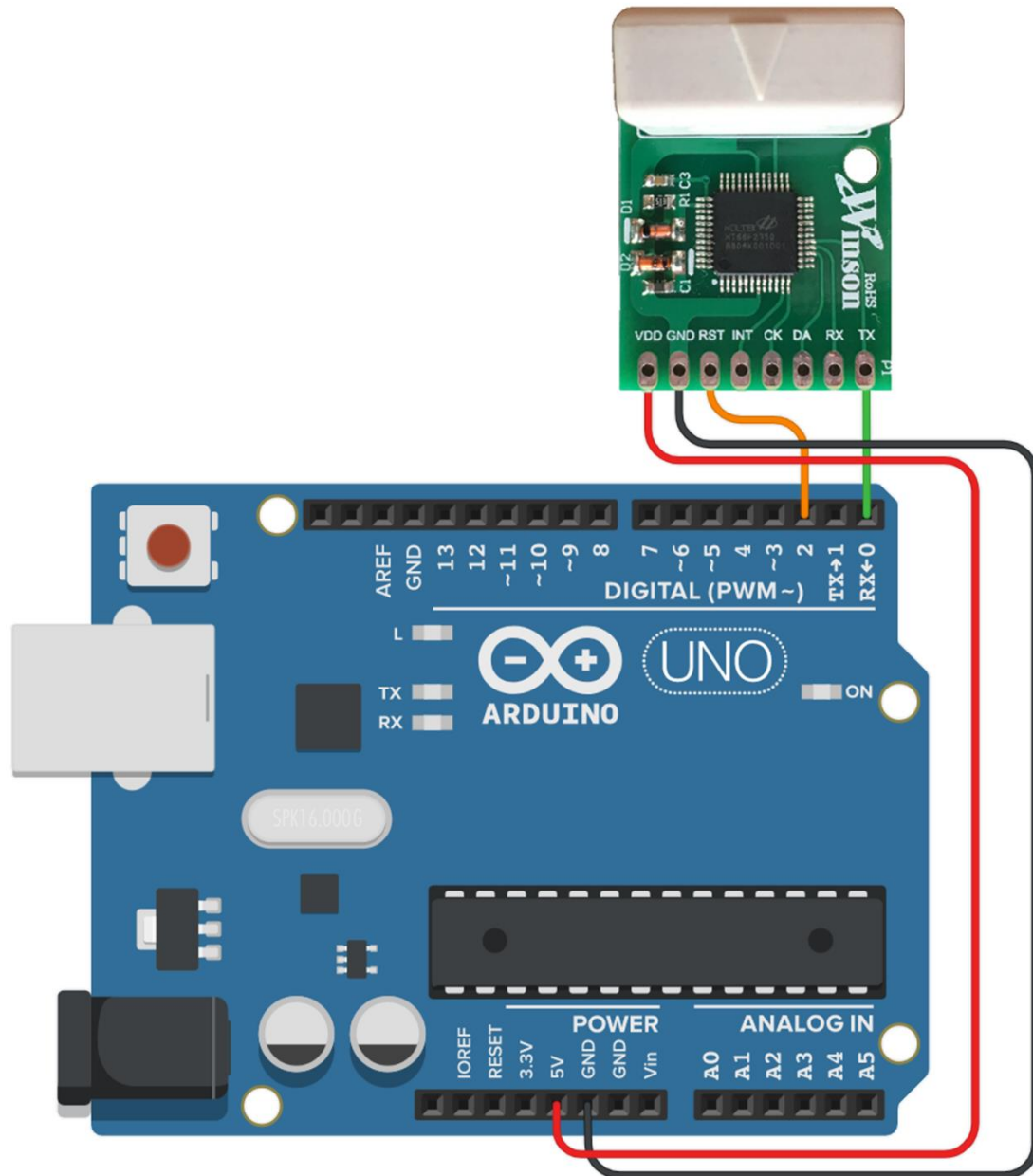


### (2.1)接線示意圖



Winson reserves the right to make changes to improve reliability or manufacturability.

## (2.2) 接線圖



Winson reserves the right to make changes to improve reliability or manufacturability.

## (2.3) 軟體&程式

(2.3.1) 範例程式於 Winson 官網載: <http://www.winson.com.tw/Product/155>

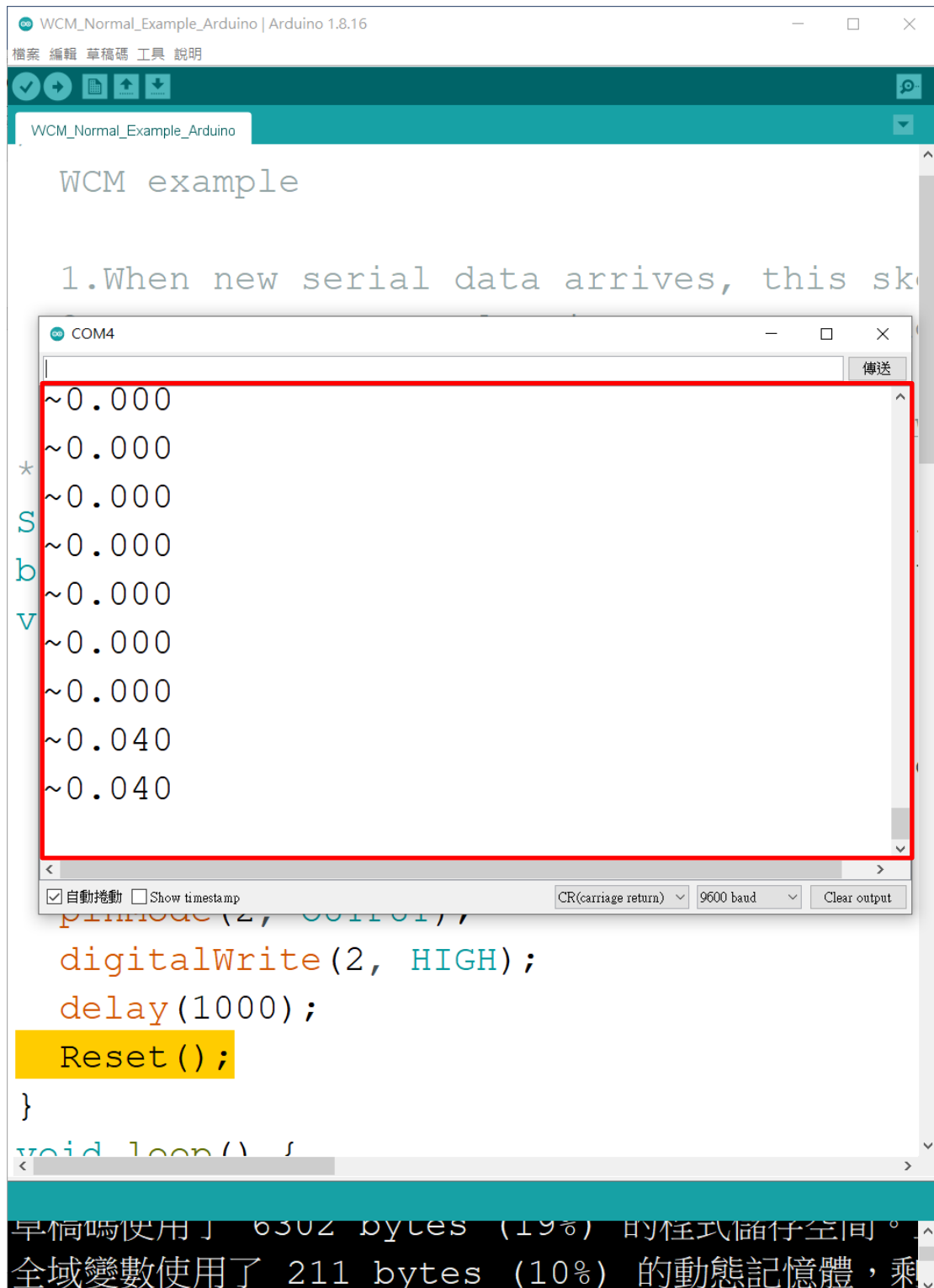


```
WCM_Normal_Example_Arduino | Arduino 1.8.16
檔案 編輯 草稿碼 工具 說明
WCM_Normal_Example_Arduino
1.When new serial data arrives, this sketch adds it to a String.
2.Convert Return Value into easy-to-read format and print it on Serial Monitor.
See Other Winson's Products on: http://www.winson.com.tw/
*/
String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
void setup() {
  // initialize serial:
  Serial.begin(9600);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
  //Reset
  pinMode(2, OUTPUT);
  digitalWrite(2, HIGH);
  delay(1000);
  Reset();
}
void loop() {
  // print the string when a newline arrives:
  if (stringComplete) {
    ConvertFormat(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
}
//Data Received Event
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag so the main loop can
    // do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
//Convert return value into Double and Display it.
double ConvertFormat(String WCM_ReturnValue){
  // String To Char Array
  char box[WCM_ReturnValue.length()];
  WCM_ReturnValue.toCharArray(box,WCM_ReturnValue.length());
  //Convert into double
  double value = (box[1] - '0')*10 + (box[2] - '0') + (box[3] - '0')*0.1 + (box[4] - '0')*0.01 + (box[5] - '0')*0.001;
  //Display on Serial Monitor
  if(box[0] == '-')value = -value;
  else Serial.print(box[0]);
  if(box[1] != '0') Serial.println(String(value,2));
  else Serial.println(String(value,3));
  //return value for other use.
  return value;
}
void Reset(){
  Serial.println("Reset");
  digitalWrite(2, LOW);
  delay(1000);
  digitalWrite(2, HIGH);
}
//Code End Here. |
```

※燒錄時注意!! 必免 Arduino 在燒錄程式時出現異常，請在燒錄後再接上 WCM。

Winson reserves the right to make changes to improve reliability or manufacturability.

(2.3.2) 驗證執行後，開啟工具\_序列埠監控視窗，可觀察量測電流值。



The screenshot shows the Arduino IDE interface. The main window displays the code for a WCM example. The serial monitor window, titled 'COM4', is open and shows the following output:

```
~0.000  
~0.000  
~0.000  
~0.000  
~0.000  
~0.000  
~0.000  
~0.000  
~0.040  
~0.040
```

The serial monitor settings are:  自動捲動,  Show timestamp, CR(carriage return), 9600 baud, and Clear output.

The code in the main window includes:

```
digitalWrite(2, HIGH);  
delay(1000);  
Reset();  
}  
void loop() {
```

At the bottom of the IDE, a status bar indicates: 早桐嗎使用了 6302 bytes (19%) 的程式儲存空間。全域變數使用了 211 bytes (10%) 的動態記憶體，乘

Winson reserves the right to make changes to improve reliability or manufacturability.