

---

## DWCS 數位輸出電流感應器使用說明

---

1. 通訊介面格式:	2
2. 工作模式:	2
3. 量測方式: (連續版別)	3
4. 量測方式: (Modbus-RTU)	3
5. 應用電路(連續版別):	6
6. 應用電路(AT 指令 & Modbus-RTU):	7

---

## Arduino 範例程式

---

1. 使用 Arduino 平台須知	10
2. 連續版別	11
3. AT 指令版本	15
4. Modbus-RTU 版別(修改地址&量測溫度)	19
5. Modbus-RTU 版別(一對多通訊)	23

## DWCS 數位輸出電流感應器使用說明

### 1. 通訊介面格式:

通訊介面	UART TTL 接口
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) 連續版別: 連續傳送電流數據，歸零需將 RST 腳下拉到 GND。

(2.2) AT 指令版別: 依指令取得電流數據及歸零，RX 接收指令，請參考指令。

項目	指令	範例	回傳值
歸零電流	AT+RST\r\n	"AT+RST\r\n"	"OK\r\n" <sup>(1)</sup>
0:直流/ 1:交流	AT+CURRE\r\n	"AT+CURRE,0\r\n" "AT+CURRE,1\r\n"	"OK\r\n" <sup>(1)</sup>
量測電流	AT+MEAS\r\n	"AT+MEAS\r\n"	<Current>
量測溫度	AT+TEMP\r\n	"AT+TEMP\r\n"	<Temperature>

1. 指令錯誤:回傳"Err\r\n"。

(2.3) Modbus-RTU 版別: 核對設備位址，並根據主機指令回傳數據資料。

項目	暫存器地址	字節	讀/寫	說明
歸零	0x0000	2	Write	輸入 256, 歸零
電流	0x0002	4	Read	十六進制有符號, 單位:0.001A 實際值=16 進制/1000 (A)
溫度	0x0004	4	Read	十六進制有符號, 單位:0.1°C 實際值=16 進制/10 (°C)
設備位址 <sup>(1)</sup>	0x0010	2	Write	默認值 0x0001, 默認地址為 01 輸入地址 1~247
直流/交流	0x0020	2	Write	0:直流/ 1:交流

1. 記憶設備位址。

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

### 3. 量測方式: (連續版別)

(3.1)交流量測:上電後在無電流通過感測器會自動歸零電流值，也可以手動歸零電流值。資料傳送:每次輸出 8 個 byte，資料更新率 2 Data/ s。

(3.2)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。資料傳送:每次輸出 8 個 byte，資料更新率 10 Data/ s。

(3.3)量測直流電時，正負電流交換時鐵心會有殘磁，請依照電流方向重新歸零。

(3.4)在無電流通過時，重新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。

### 4. 量測方式: (Modbus-RTU)

#### (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) 從機不會響應。

### (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。

### (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	00H, 01H	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。

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

### (4.2.3)寫入量測電流方式 (AC / DC)

主機傳送: 01 06 00 20 00 01 49 C0

從機響應: 01 06 00 20 00 01 49 C0

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 , 20H	00H, <b>01H</b>	49H, C0H

結果為: (01)感測器號碼為 1，寫入電流量測模式 AC (00 **01**) / DC(00 **00**)。

## (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	01H	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

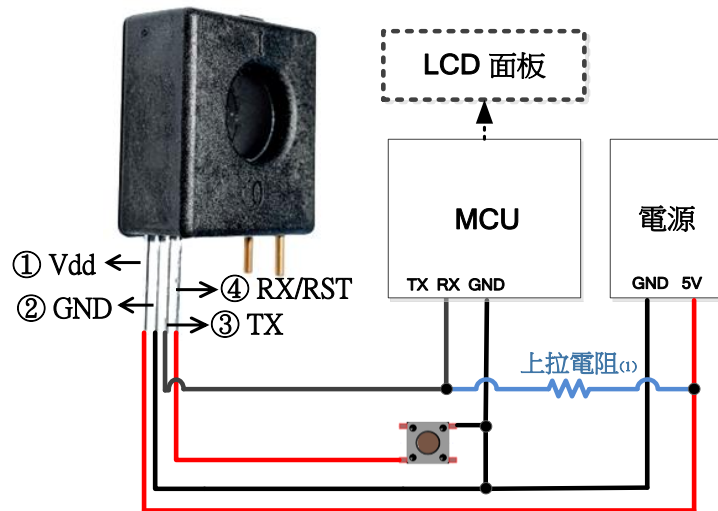
主機傳送: 00 06 00 10 00 01 48 1E

從機響應: **無響應**

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

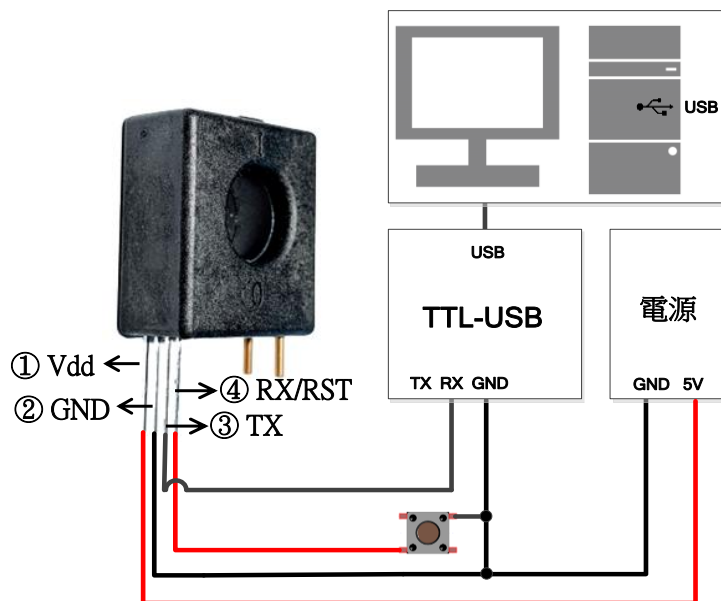
## 5. 應用電路(連續版別):

- MCU 接線圖



(1)DWCS 的 TX 輸出為 Open Drain 配置，須使用上拉電阻，如果 MCU RX 腳位已內部上拉，則此電阻可以不加。

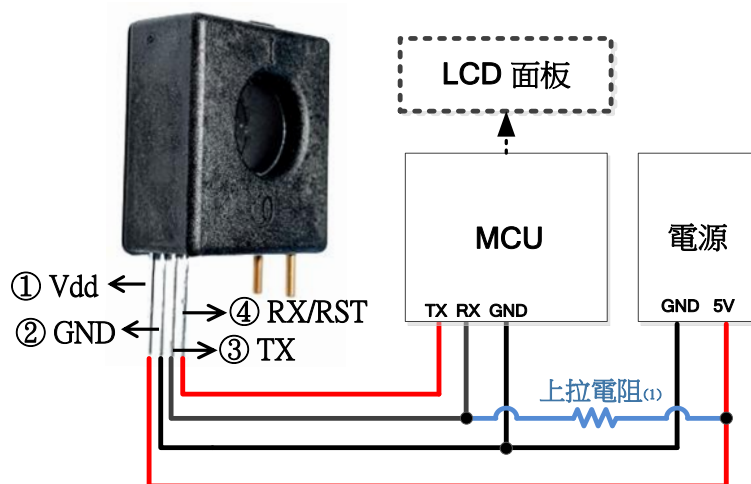
- TTL to USB 接線圖



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

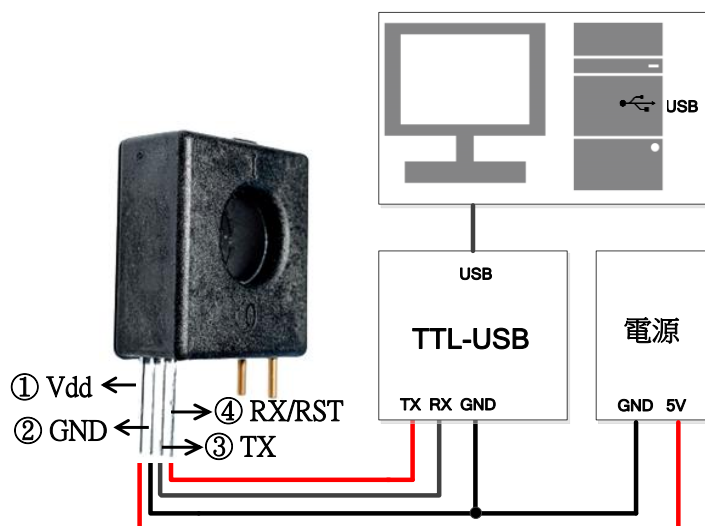
## 6. 應用電路(AT 指令 & Modbus-RTU):

- MCU 接線圖



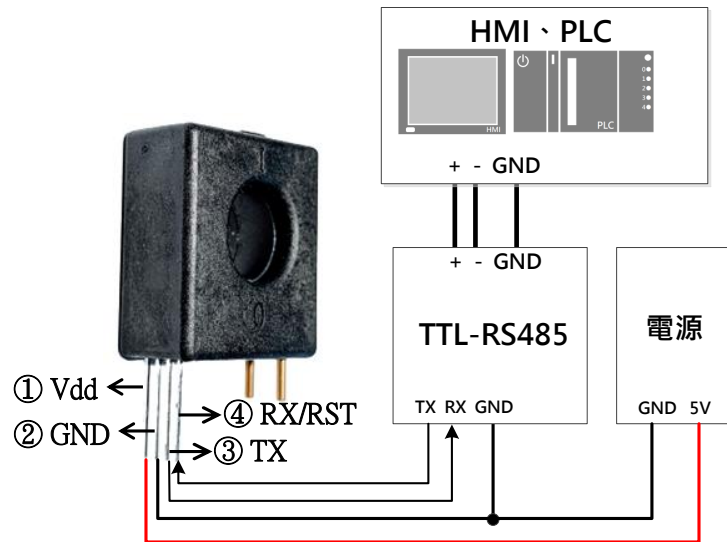
(1)DWCS 的 TX 輸出為 Open Drain 配置，須使用上拉電阻，如果 MCU RX 腳位已內部上拉，則此電阻可以不加。

- TTL to USB 接線圖



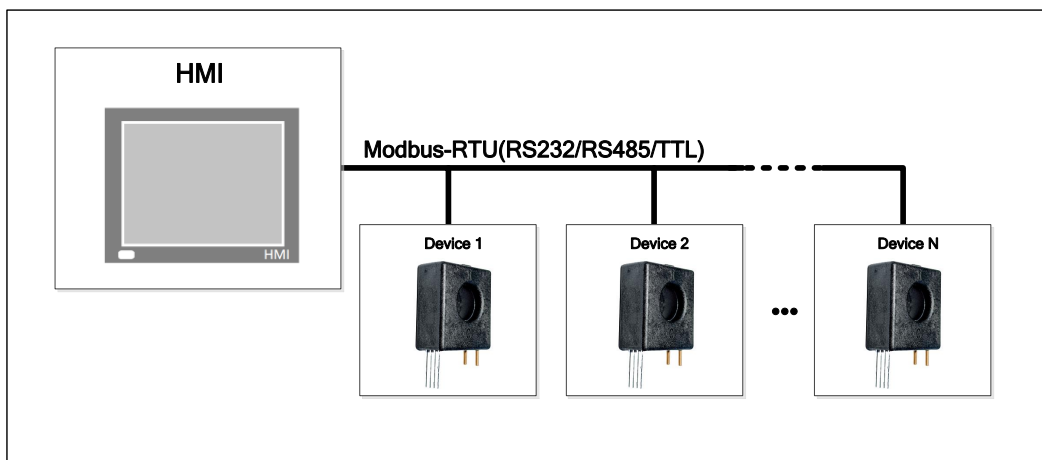
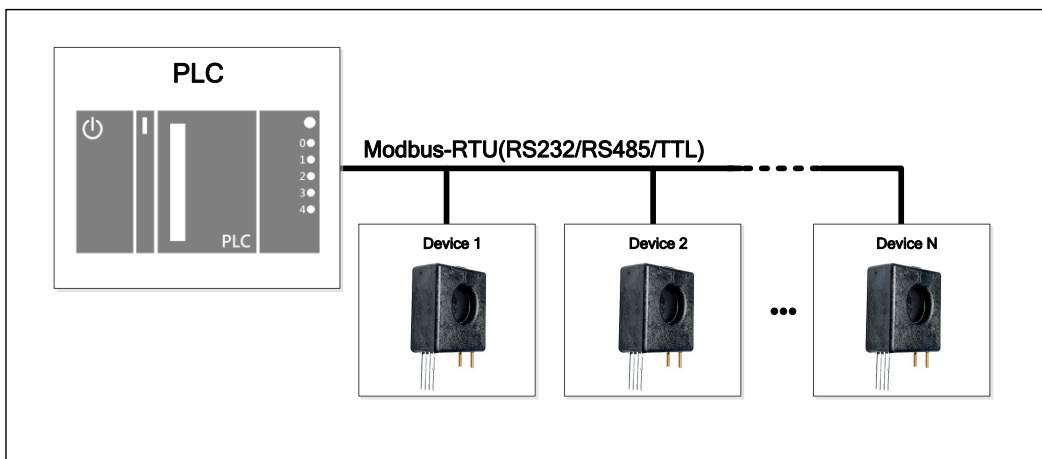
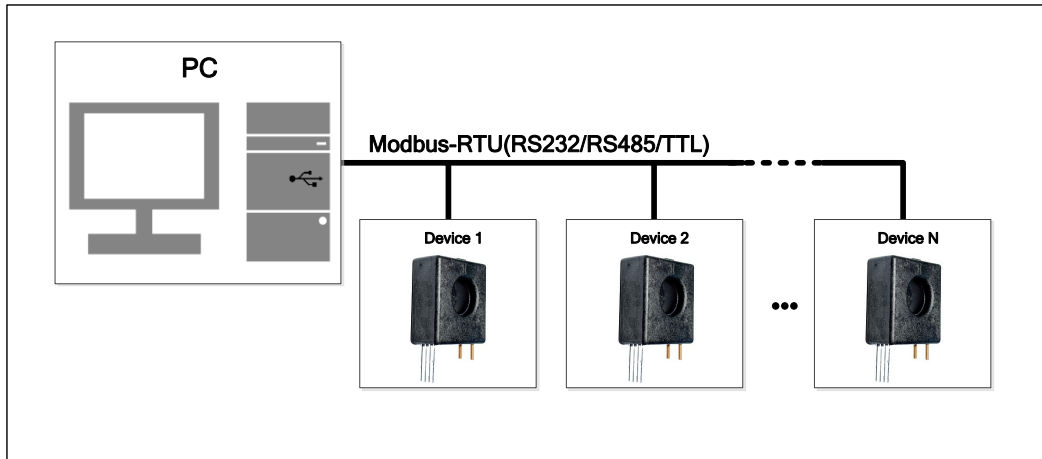
Winson reserves the right to make changes to improve reliability or manufacturability.

- TTL to RS485 接線圖





- **Modbus-RTU 架構圖：**

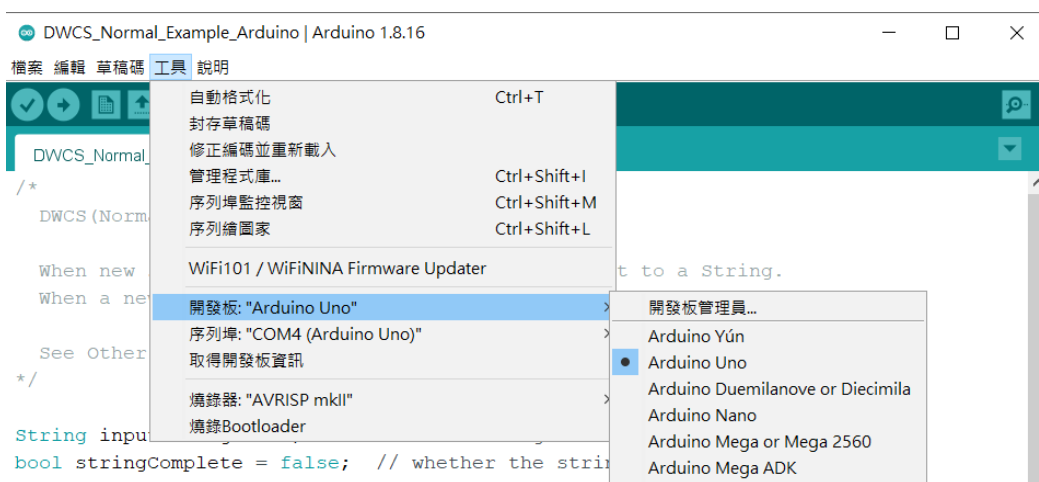


Winson reserves the right to make changes to improve reliability or manufacturability.

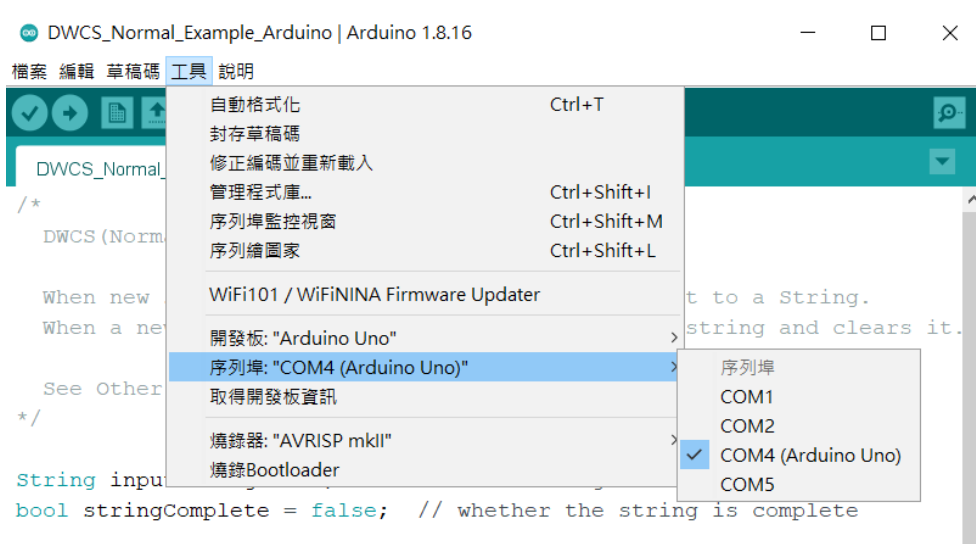
## Arduino 範例程式

### 1. 使用 Arduino 平台須知

- **工具\_開發板:** 選取 Arduino 的板別



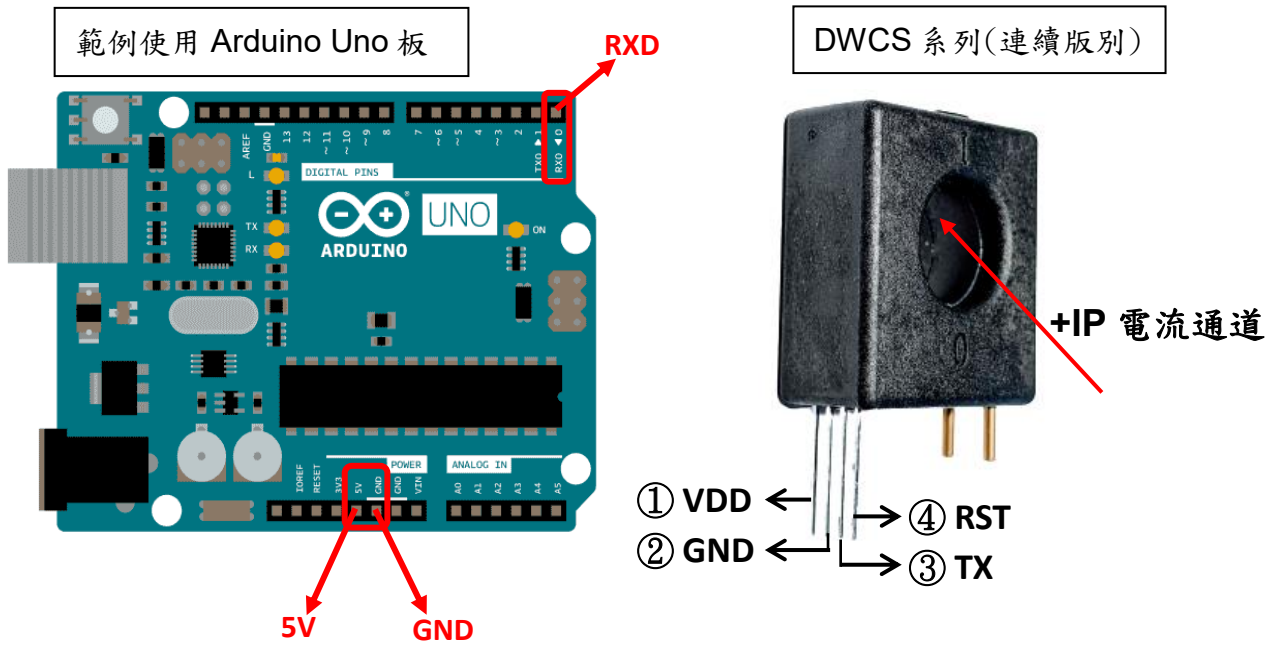
- **工具\_序列埠:** 選取 Arduino 的序列埠位置



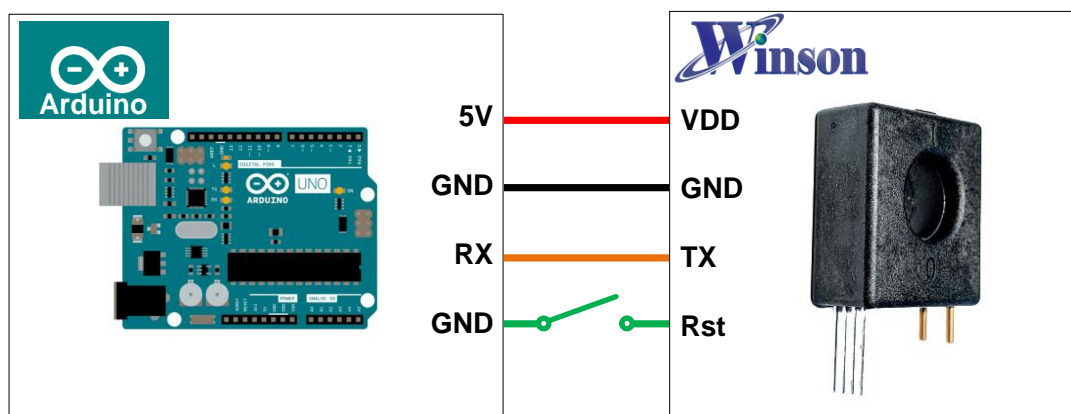
※Arduino 序列埠位置可使用 Windows 的裝置管理員查詢。

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

## 2. 連續版別

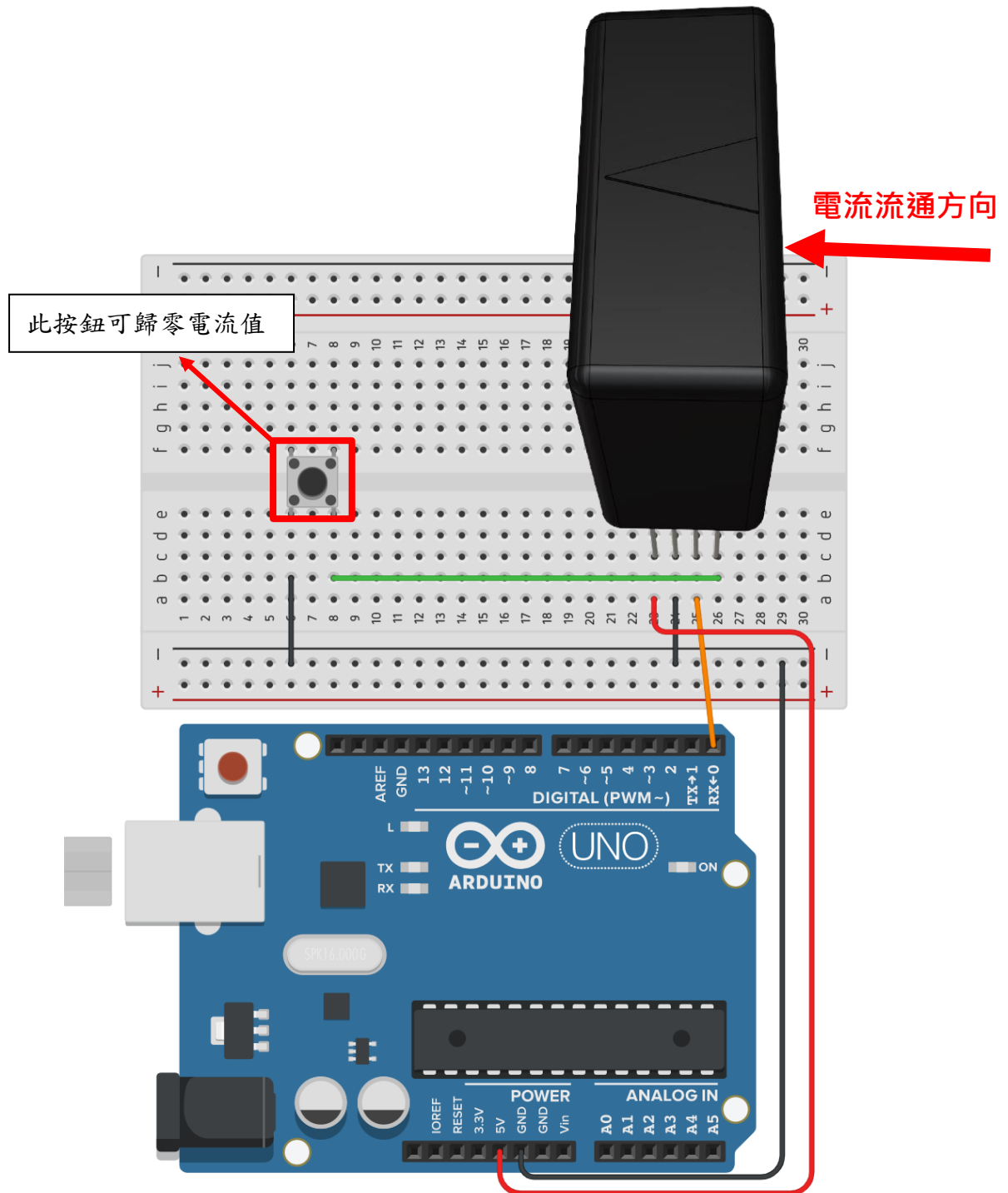


- 接線示意圖



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

- 接線圖



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

- 軟體&程式

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



```
SerialEvent | Arduino 1.8.16
檔案 編輯 草稿碼 工具 說明
SerialEvent$
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);
}

void loop() {
  // print the string when a newline arrives:
  if (stringComplete) {
    Serial.println(inputString);
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
}

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;
    }
  }
}
//Code End Here

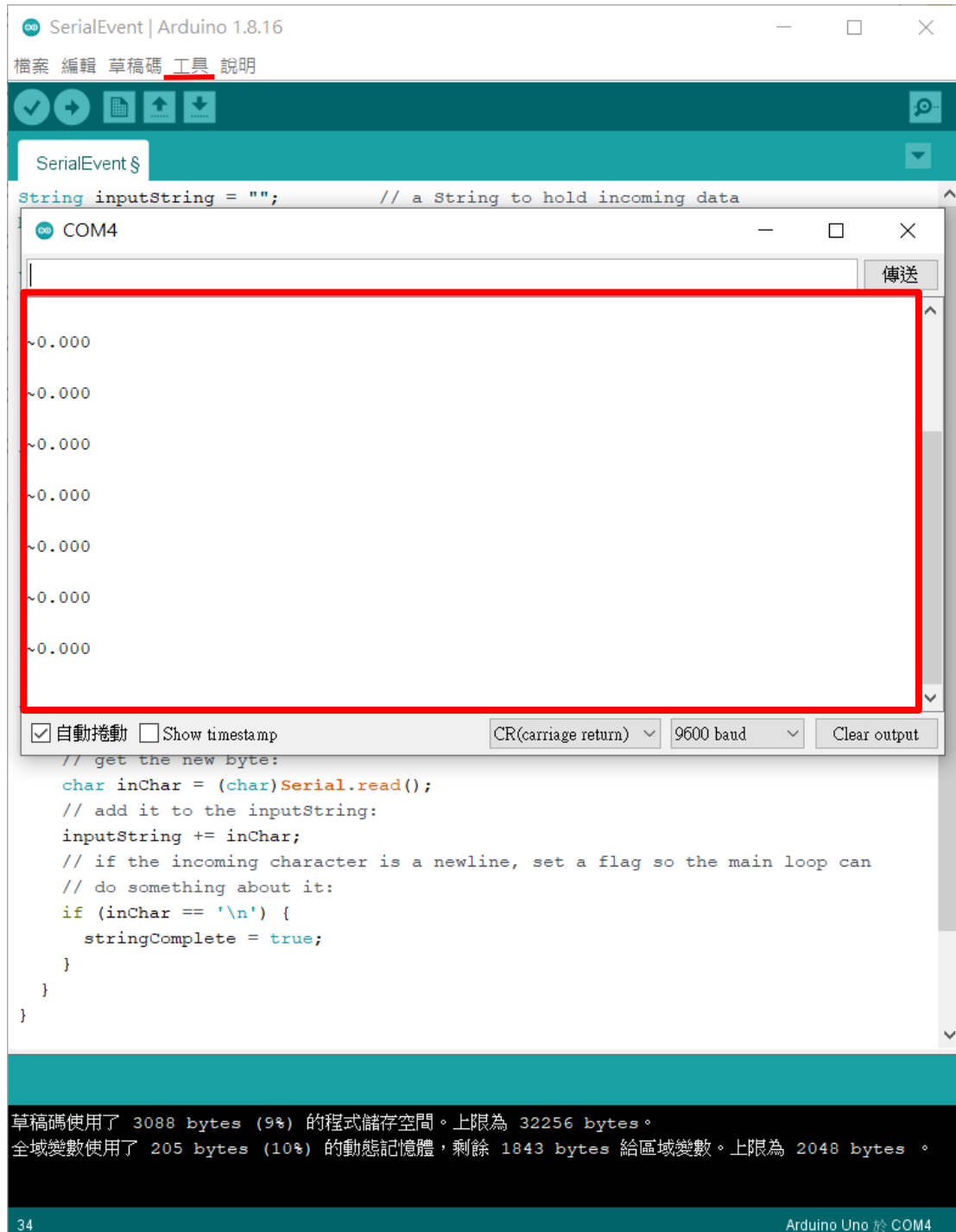
草稿碼使用了 3088 bytes (9%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 205 bytes (10%) 的動態記憶體，剩餘 1843 bytes 給區域變數。上限為 2048 bytes。

34 Arduino Uno 於 COM4
```

※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 DWCS。

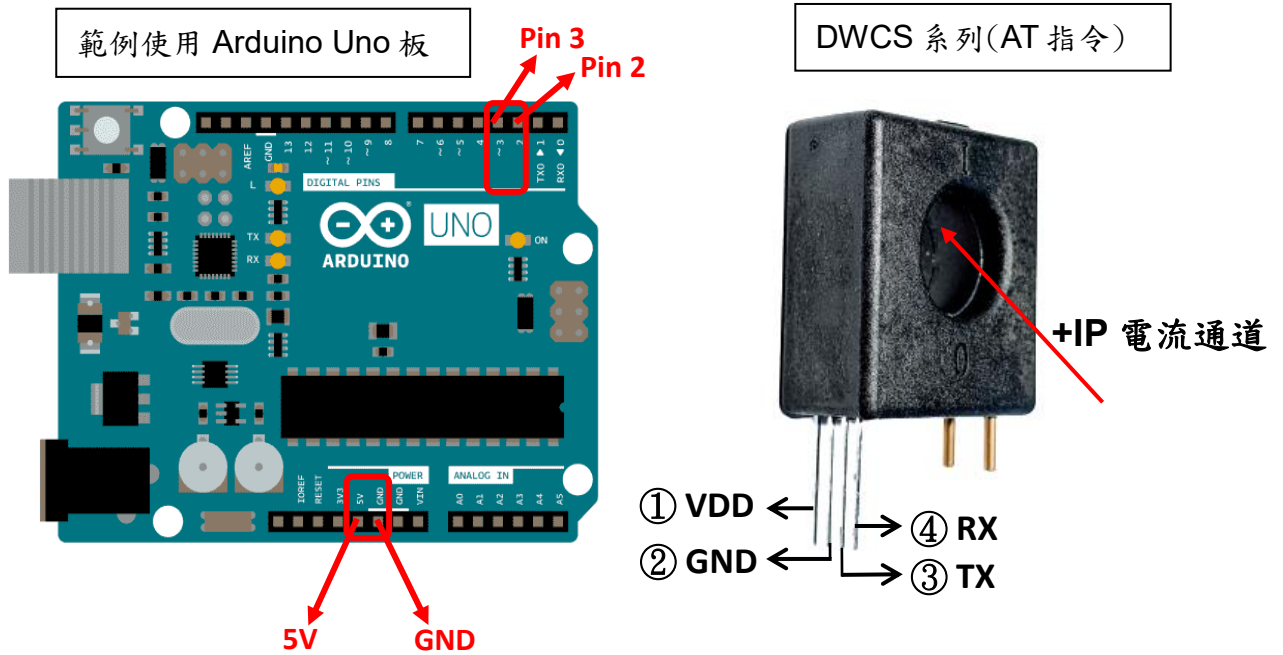
Winson reserves the right to make changes to improve reliability or manufacturability.

(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。

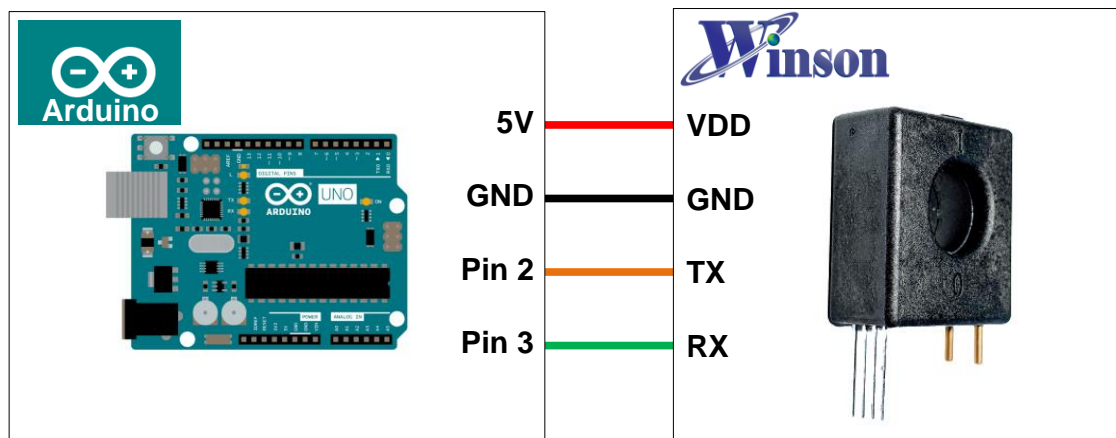


Winson reserves the right to make changes to improve reliability or manufacturability.

## 3. AT 指令版別

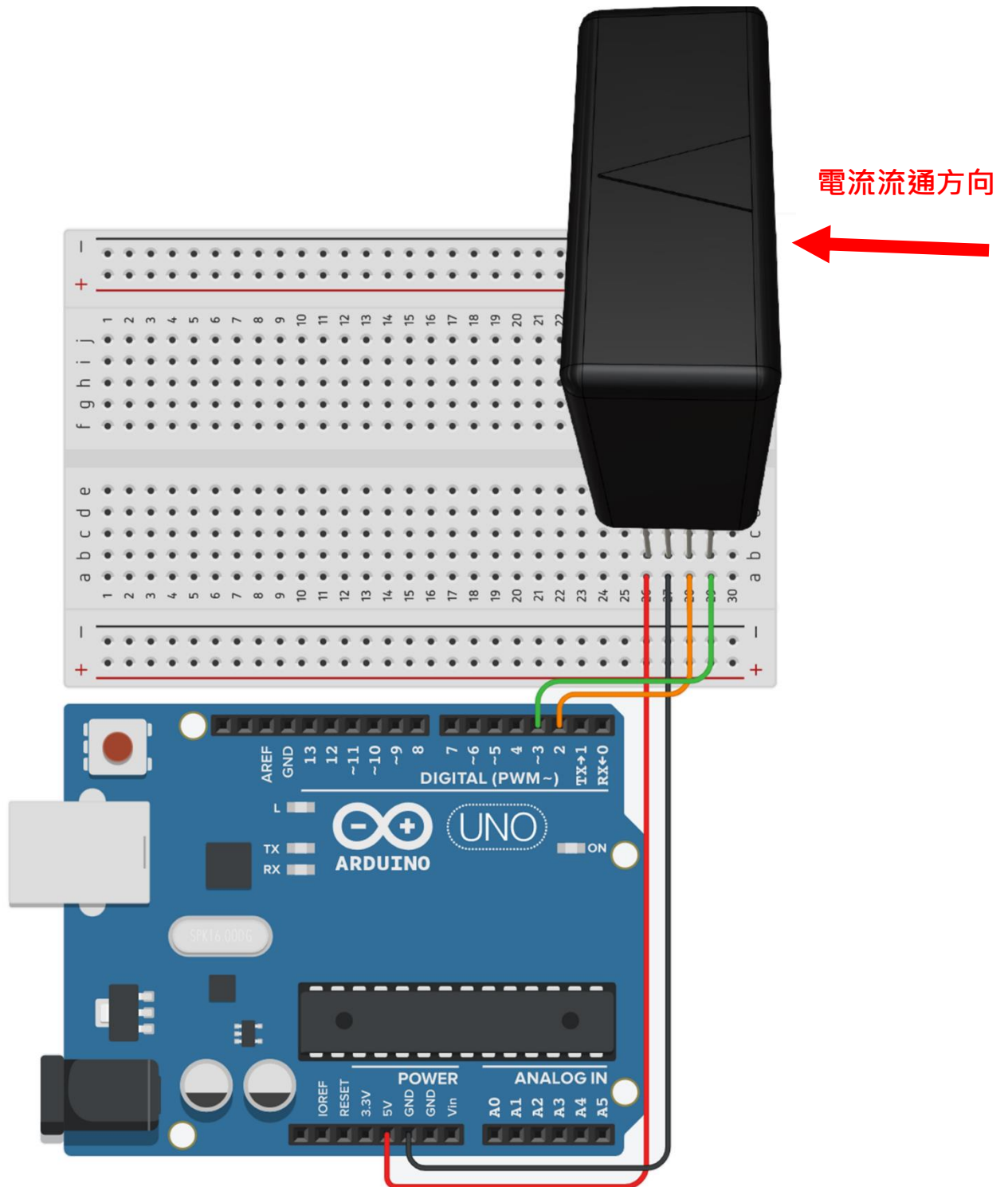


- 接線示意圖



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

- 接線圖



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



## ● 軟體&程式

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



```
DWCS_Professional_Example_Program_Arduino | Arduino 1.8.16
File Edit Sketch Tools Help
DWCS Professional Example Program Arduino
/*
DWCS (Professional Version) example
1. In this example, we use Serial Monitor as display panel.
2. Send a Command("AT+MEAS\r\n") to DWCS every 1 Seconds.
3. When DWCS response , this sketch adds the return value to a String.

See Other Winson's Products on: http://www.winson.com.tw/
*/
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2 , 3); // RX, TX for DWCS
String inputString = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
String recievedString = ""; //variable for save the return value

void setup() {
//reserve 200 bytes for the inputString:
inputString.reserve(200);
// initialize Display serial:
Serial.begin(9600);
//initialize DWCS serial:
mySerial.begin(9600);
delay(1000);
}

void loop() {
//Write Command to DWCS
recievedString = WriteCommand("AT+MEAS\r\n");
delay(1000);
}

void serialEvent() {
while (mySerial.available()) {
// get the new byte:
char inChar = (char)mySerial.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;
}
}
}

//Write Command to DWCS.
//this function will auto display the command and return value on Serial Monitor.
String WriteCommand(String stringData) {
int count = 0;
String box;
//Display Command on Monitor
Serial.print(stringData);

//Send Command to DWCS
mySerial.print(stringData);

//Wait until RX recieved data
serialEvent();
delay(200);
//if Rx Recieved Completed
if(stringComplete){
box = inputString;
// clear the string:
inputString = "";
stringComplete = false;
//Display Return value on Monitor
Serial.println(box);
return box;
}
}
//Codes 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 DWCS。

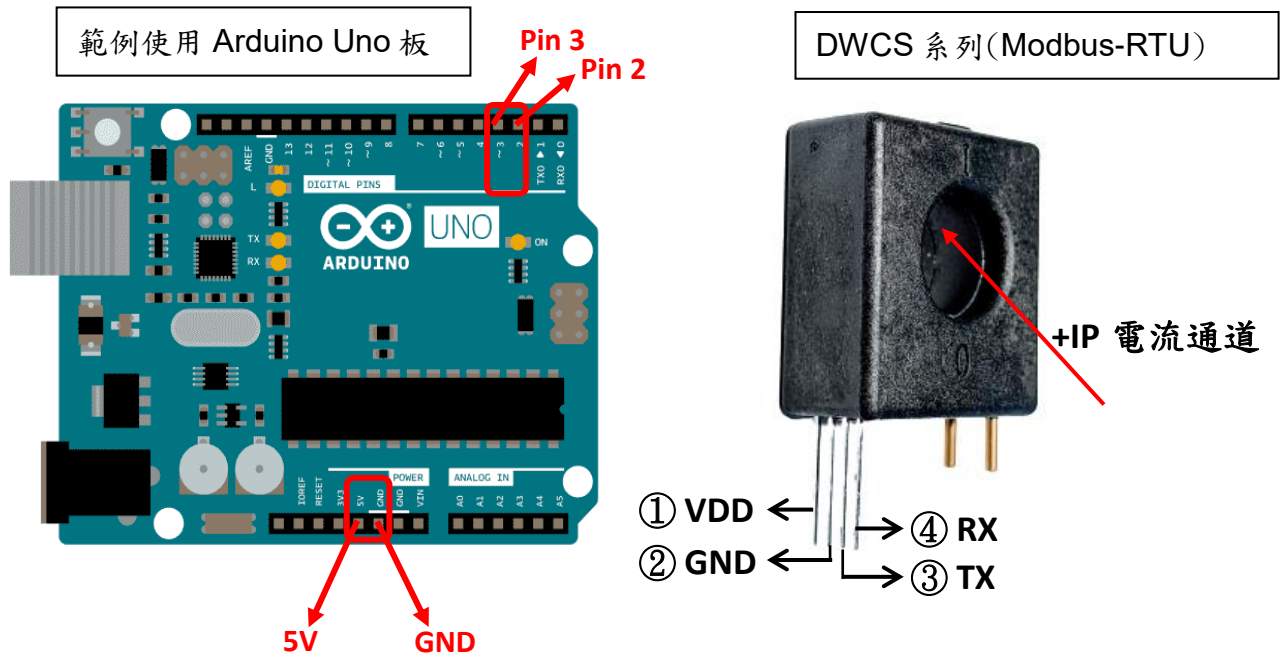
Winson reserves the right to make changes to improve reliability or manufacturability.

(2). 驗證執行後，開啟工具\_序列埠監控視窗，可觀察指令回傳值。

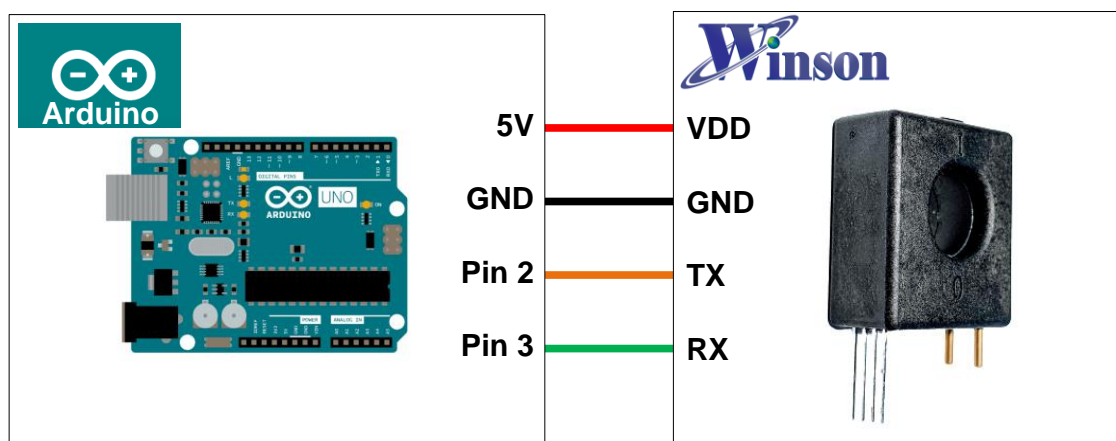


Winson reserves the right to make changes to improve reliability or manufacturability.

## 4. Modbus-RTU 版別(修改地址&量測溫度)

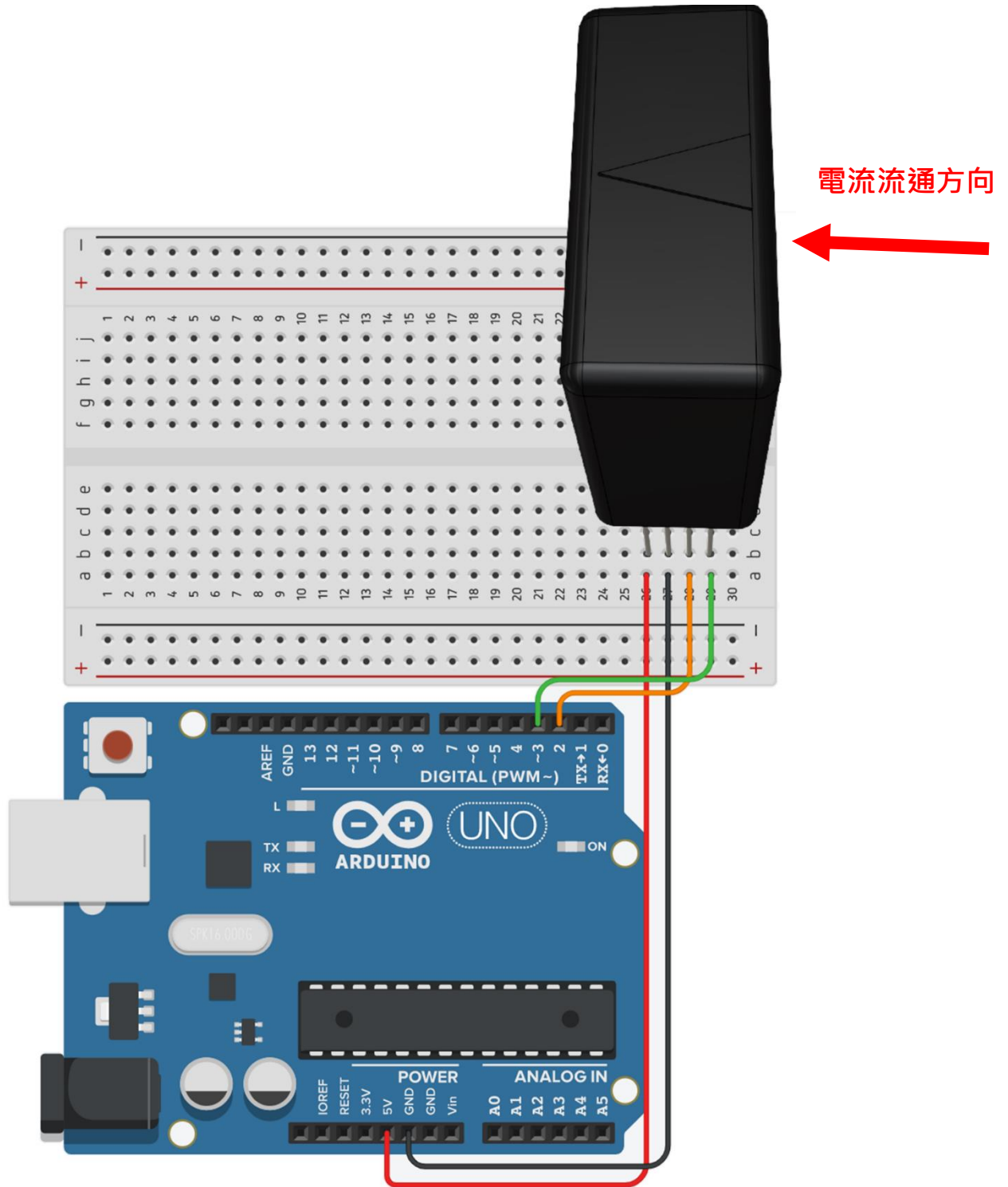


- 接線示意圖



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

- 接線圖



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

- 軟體&程式

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

```

SoftwareSerial mySerial(2 , 3); // RX, TX for DWCS
//=====
word NewAddress = 2; //The
//=====
byte RxBuff[100]; //Rx Recieved Buffer
int RxIndex = 0; //RxBuff Index

void setup() {
  // initialize Display serial:
  Serial.begin(9600);
  //initialize DWCS serial:
  mySerial.begin(9600);
  delay(1000);
  //Use Podcast Address to Change Every Slave Address to the Same Address.
  WriteCommand(0x00,0x06,0x0010,NewAddress); // Write Address Command
  delay(1000);
  //Use New Address to send Reset Command to DWCS
  WriteCommand(NewAddress,0x06,0x0000,0x0100); //Reset Command
  delay(1000);
}

void loop() {
  //Routinely send command to DWCS use New Address
  Serial.println("=====");
  WriteCommand(NewAddress,0x03,0x0004,0x0002); //Read Temperature Command
  delay(1000);
}
/*****
* Function : DataRecieved
  
```

在此輸入修改的新地址

使用廣播地址(0x00)將所有位置修改為新地址

WriteCommand(0x00,0x06,0x0010,NewAddress); // Write Address Command

使用新地址傳送歸零指令

WriteCommand(NewAddress,0x06,0x0000,0x0100); //Reset Command

讀取溫度

WriteCommand(NewAddress,0x03,0x0004,0x0002); //Read Temperature Command

### WriteCommand 指令:

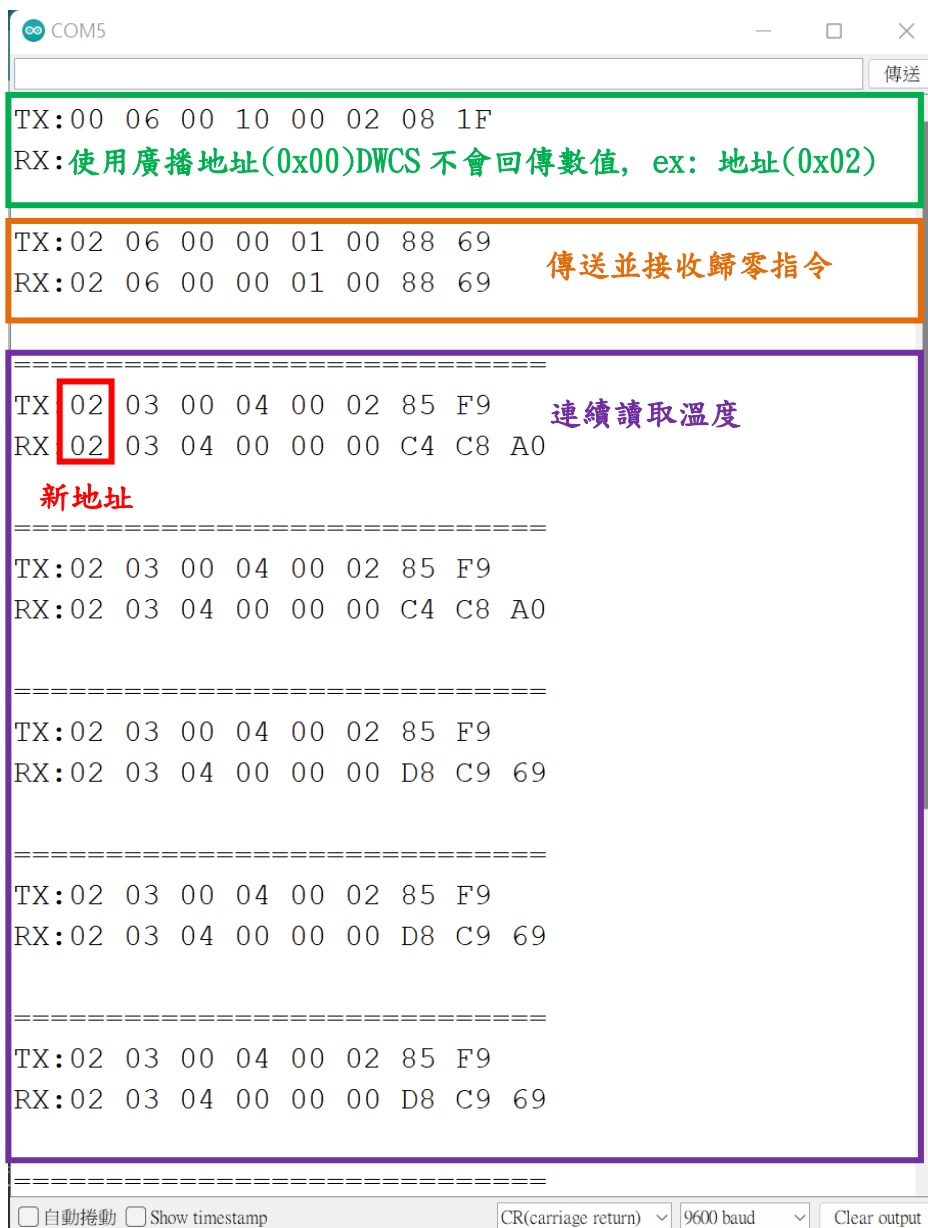
```

void WriteCommand(byte SlaveAddress,byte FunctionCode,word DeviceAddress,word RegisterNum)
                地址          讀(03H)/寫(06H)  暫存器地址  暫存器數
  
```

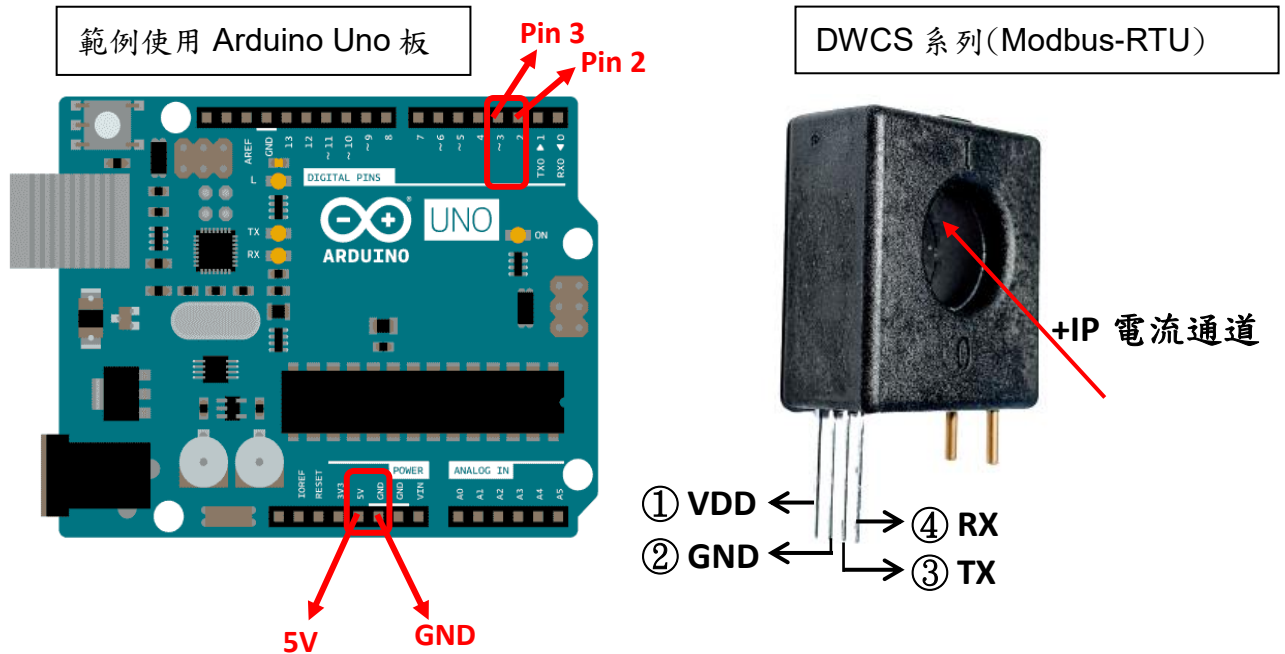
※燒錄時注意!! 必免 Arduino 在燒錄程式時出現異常，請在燒錄後再接上 DWCS。

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

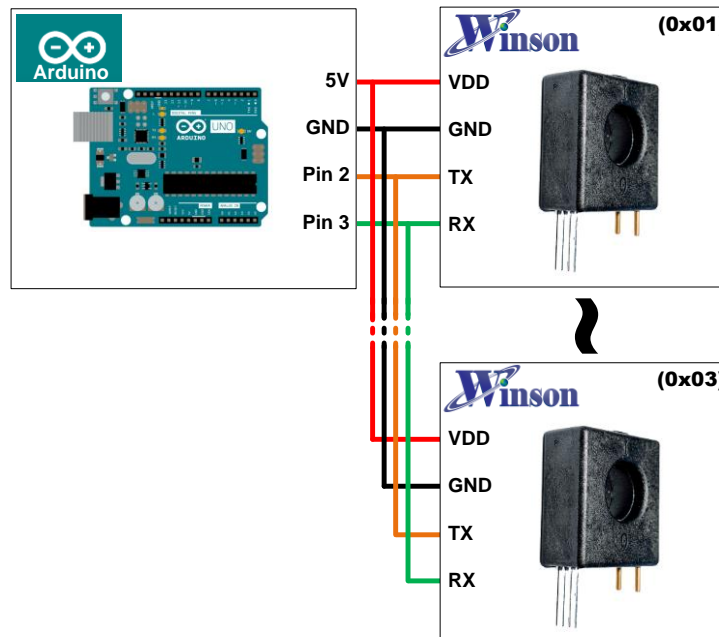
(2). 驗證執行後，開啟工具\_序列埠監控視窗，可觀察指令回傳值。



## 5. Modbus-RTU 版別(一對多通訊)

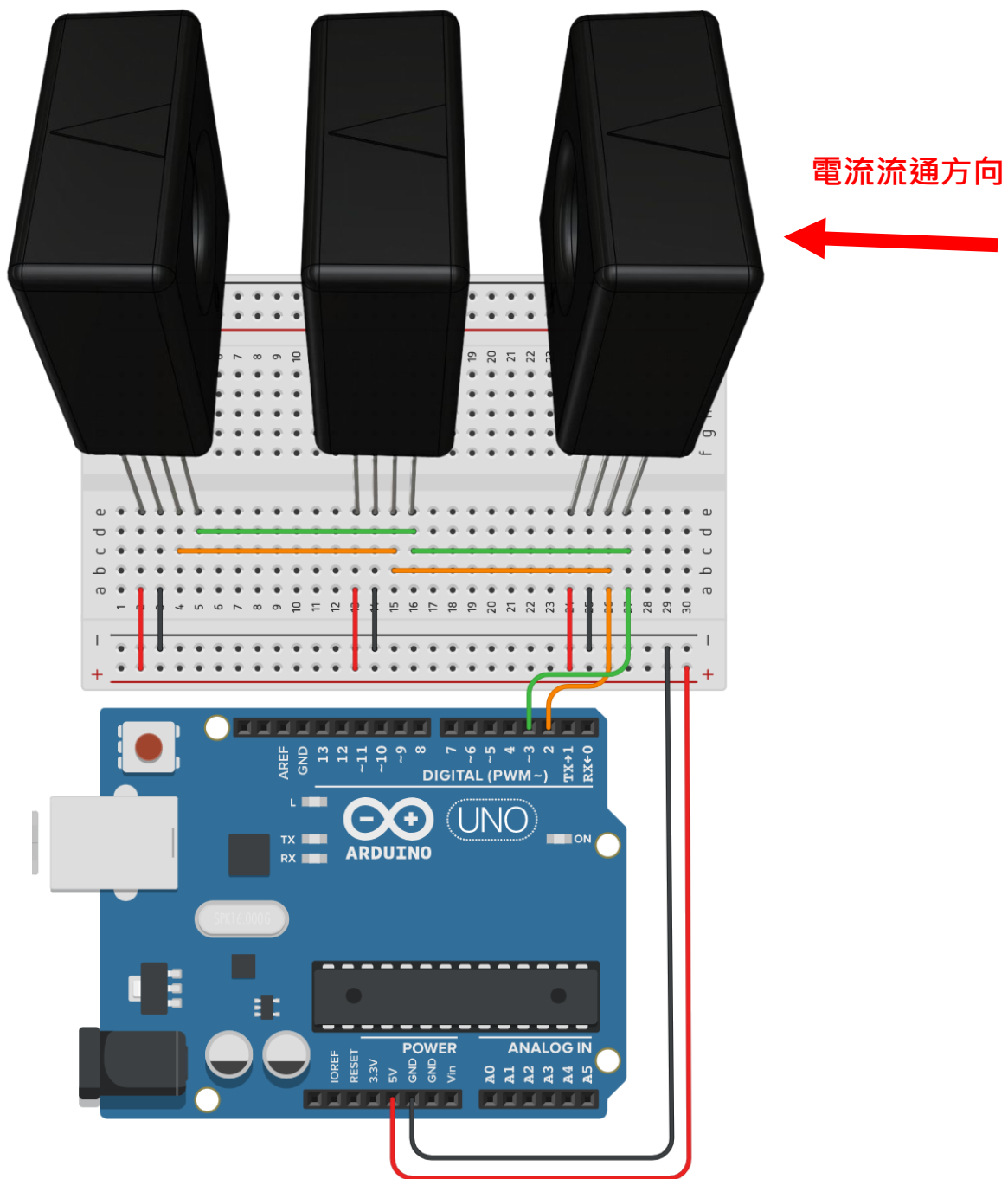


- 接線示意圖



※每個 DWCS 位址不可重複(更改位址請看[上例](#))

- 接線圖



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



- 軟體&程式

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

```
OneToManyCommunication
*/
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2 , 3); // RX, TX for DWCS
//=====
byte RxBuff[100]; //Rx Recieved Buffer
int RxIndex = 0; //RxBuff Index

void setup() {
  // initialize Display serial:
  Serial.begin(9600);
  //initialize DWCS serial:
  mySerial.begin(9600);
  delay(1000);
  //Use Podcast Address to send Reset Command to all DWCS at once.
  WriteCommand(0x00,0x06,0x0000,0x0100); //Reset Command
  delay(1000);
}

void loop() {
  //Routinely send command to DWCS in address 0x00,0x06,0x0000,0x0100
  Serial.println("=====\\r\\n");
  for(int i = 1 ;i<4;i++)
  {
    WriteCommand(i,0x03,0x0002,0x0002); //ReadCurrent Command
    delay(1000);
  }
}

/*****
 * Function : DataRecieved
 * Discription: serial Data Recieved Event.
 *****/
void DataRecieved() {
  //Reset RxIndex if RxBuff is full.
  if(RxIndex>(sizeof(RxBuff)- 1))RxIndex = 0;
}
```

使用廣播地址(0x00)一次歸零所有感測器

多個感測器循環讀取量測電流值

### WriteCommand 指令:

```
void WriteCommand(byte SlaveAddress,byte FunctionCode,word DeviceAddress,word RegisterNum)
```

地址                      讀(03H)/寫(06H)                      暫存器地址                      暫存器數

※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 DWCS。

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

(2). 驗證執行後，開啟工具\_序列埠監控視窗，可觀察指令回傳值。

