

---

## **DWCS 數位輸出電流感應器使用說明** **2**

---

1. 通訊介面格式:	2
2. 工作模式:	2
3. 量測方式:(連續版別 & AT 指令版別)	4
4. 量測方式:(MODBUS-RTU)	5
5. 量測方式:(I2C)	10
6. 應用電路(連續版別):	15
7. 應用電路(AT 指令 & MODBUS-RTU):	16
8. 應用電路(I2C):	18

---

## **Arduino 範例程式** **19**

---

1. 使用 ARDUINO 平台須知	19
2. 連續版別	20
3. AT 指令版本	24
4. MODBUS-RTU 版別(修改地址&量測溫度)	28
5. MODBUS-RTU 版別(一對多通訊)	32

## 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

通訊介面	I2C 接口
通訊速率	標準模式(100kHz)

### 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、溫度數據及歸零，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" <b>(1)</b>
0: 直流 / 1: 交流	AT+CURRE\r\n	"AT+CURRE,0\r\n" "AT+CURRE,1\r\n"	"OK\r\n" <b>(1)</b>
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>
量測內部工作電壓	AT+VDD\r\n	"AT+VDD\r\n"	< Voltage>
量測 120 筆資料	AT+BUF\r\n	"AT+BUF\r\n"	<I1,I2,I3,...\r\n>
更改波特率	AT+BR\r\n	"AT+BR,1\r\n" <b>(2)</b>	"OK\r\n" <b>(1)</b>

1. 指令錯誤: 回傳 "Err\r\n"。

2. Baud Rate ( 1: 9600, 2: 19200, 3: 38400, 4: 57600, 5: 125000 (bit/s) )

(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	寫入 0x0100, 歸零
量測旗標 資料有效	0x0001	2	Write/ Read	Write: 0x0002: 置位量測旗標 Read: 0x0000: 量測旗標復位, 資料旗標無效 0x0001: 量測旗標復位, 資料旗標有效 0x0002: 量測旗標置位, 資料旗標無效 0x0003: 量測旗標置位, 資料旗標有效
電流	0x0002	4	Read	32 位有符號, 單位: 0.001A 實際值=32 位有符號/1000 (A)
溫度	0x0004	4	Read	32 位有符號, 單位: 0.1°C 實際值=32 位有符號/10 (°C)
設備位址	0x0010	2	Write	默認值 0x0001, 默認地址為 01 輸入地址 1~247
波特率	0x0011	2	Write	默認值 0x0001, 0x0001: 波特率= 9600 bit/s 0x0002: 波特率= 19200 bit/s 0x0003: 波特率= 38400 bit/s 0x0004: 波特率= 57600 bit/s 0x0005: 波特率= 125000 bit/s
直流/交流	0x0020	2	Write	0: 直流/ 1: 交流

(2.4) I2C 版別: 核對設備位址，並根據主機指令回傳數據資料。

位址	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	功能	範圍
00h	0	0	0	0	0	0	0	Auto Mode	Control	0-1
01h	0	Address							Slave Address	-
02h	0	0	0	0	0	0	0	Reset	Current Reset	0-1
03h	0	0	0	0	0	0	Status	Valid	Status	0-3
04h~07h	Temperature								Temperature	-
08h~0Bh	+/-Current								DC Current	-
0Ch~0Fh	~Current								AC Current	-

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

### 3. 量測方式: (連續版別 & AT 指令版別)

(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 5 Data/ s。

(3.2)直流量測: 因感測元件的鐵芯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 5 Data/ s。

(3.3)量測直流電時，正負電流交換時鐵芯會有殘磁，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。

(3.5) 量測電流數據(連續版別 & AT 指令版別):

- 輸出交流電流值為 1.23A，輸出數據 '~', '1', '.', '2', '3', '0', '\r', '\n'，共 8 byte 以 ASCII 輸出。若測得數據為 10.45，則輸出數據為 '~', '1', '0', '.', '4', '5', '\r', '\n'，共 8 個 byte。
- 輸出正直流電流值為 1.23A，輸出數據 '+', '1', '.', '2', '3', '0', '\r', '\n'，共 8 個 byte 以 ASCII 輸出。輸出負直流電流值為 1.23A，輸出數據 '-', '1', '.', '2', '3', '0', '\r', '\n'，共 8 個 byte。

(3.6)量測溫度數據(AT 指令版別): 輸出溫度為 25.5°C，輸出數據 '2', '5', '.', '5', '\r', '\n'，共 6 個 byte 以 ASCII 輸出。若測得數據為 5.0°C，則輸出數據為 '5', '.', '0', '\r', '\n'，共 5 個 byte。若測得數據為 -10.0°C，則輸出數據為 '-', '1', '0', '.', '0', '\r', '\n'，共 7 個 byte。

(3.7)量測內部工作電壓數據(AT 指令版別): 輸出電壓為 5.002v，輸出數據 '5', '.', '0', '0', '2', '\r', '\n'，共 7 個 byte 以 ASCII 輸出。

(3.8)量測 120 筆電流數據(AT 指令版別): 輸出數據

"+1.234,+1.233,+10.23,+10.24,-1.234,-1.233.....\r\n"，共 120 筆資料以 ASCII 輸出。

## 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)指令旗標位&資料有效位

主機傳送: 01 03 00 01 00 01 D5 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 , 01H	00H , 01H	D5H, CAH

從機響應: 01 03 02 00 03 F8 45

地址	命令	資料長度	資料 (2 Bytes)	校驗碼(CRC) (低位在前)
<b>01H</b>	03H	02H	<b>00H , 03H</b>	F8H, 45H

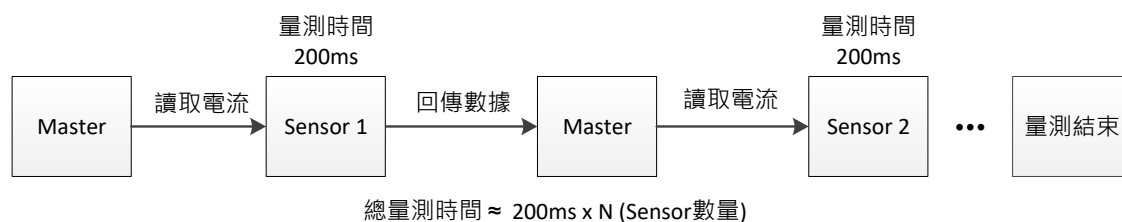
結果為: (01)感測器號碼為 1

(00 00):量測旗標復位,資料旗標無效，(00 01):量測旗標復位,資料旗標有效

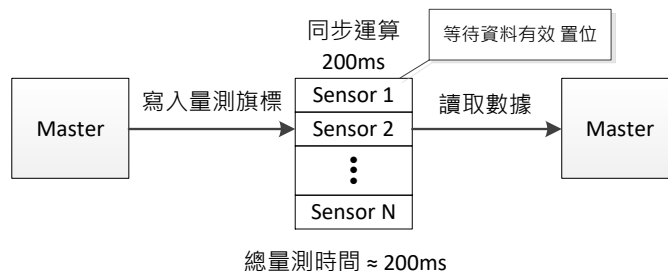
(00 02):量測旗標置位,資料旗標無效，(00 03):量測旗標置位,資料旗標有效

### 1. 量測方式

一般量測方式:



使用量測旗標方式:



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

### (4.1.2) 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)
<b>01H</b>	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。

### (4.1.3) 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)
<b>01H</b>	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)量測旗標&資料有效

主機傳送: 01 06 00 01 00 02 59 CB

從機響應: 01 06 00 01 00 02 59 CB

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 , 01H	<b>00H , 02H</b>	59H, CBH

結果為: (01)感測器號碼為 1, (**00 02**) 置位量測旗標。

### (4.2.3)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。

### (4.2.4) 寫入從機波特率 (Baud Rate)

主機傳送: 01 06 00 11 00 01 18 0F

從機響應: 01 06 00 11 00 01 18 0F

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 , 11H	<b>00H , 01H</b>	18H, 0FH

結果為: (01)感測器號碼為 1, 預設為 1, (**00 01**)寫入波特率 9600 bit/s。

(00 01): 9600, (00 02): 19200, (00 03): 38400, (00 04): 57600, (00 05): 125000 (bit/s)

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

### (4.2.5) 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)
<b>01H</b>	06H	00H, 20H	<b>00H, 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)
<b>01H</b>	<b>81H</b>	<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

### (4.3.4)從機忙碌

主機傳送: 01 03 00 01 00 01 D5 CA

從機響應: **01 83 06** C1 32

結果為: (01)感測器號碼為 1, (83)=0X80(異常) + 0X06(功能碼), (06)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**

## 5. 量測方式: (I2C)

### (5.1) 暫存器設置

- 暫存器初始值:

暫存器	上電復位值	暫存器	上電復位值
00h	0000 0001	08h	0000 0000
01h	0101 0011	09h	0000 0000
02h	0000 0000	0Ah	0000 0000
03h	0000 0000	0Bh	0000 0000
04h	0000 0000	0Ch	0000 0000
05h	0000 0000	0Dh	0000 0000
06h	0000 0000	0Eh	0000 0000
07h	0000 0000	0Fh	0000 0000

- Control 暫存器(00h)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	Auto Mode
R/W	-	-	-	-	-	-	-	R/W
POR	-	-	-	-	-	-	-	1

Bit 7~1 未定義，讀為”0”

Bit 0 Auto Mode: 自動量測溫度與電流

0: 手動，需在 Status 暫存器設置(參考 **Status 暫存器**)

1: 自動

- Slave Address 暫存器(01h)

Bit	7	6	5	4	3	2	1	0
Name	-	Slave Address						
R/W	-	R/W						
POR	-	1	0	1	0	0	1	1

Bit 7 未定義，讀為”0”

Bit 6~0 Slave Address: 初始位置 0x53

● Current Reset 暫存器(02h)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	Reset
R/W	-	-	-	-	-	-	-	R/W
POR	-	-	-	-	-	-	-	0

Bit 7~1 未定義，讀為”0”

Bit 0 Reset: 電流歸零旗標

0: 復位

1: 置位，歸零

歸零後會自動清零復位。

● Status 暫存器(03h)

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	Status	Valid
R/W	-	-	-	-	-	-	R/W	R
POR	-	-	-	-	-	-	0	0

Bit 7~2 未定義，讀為”0”

Bit 1 Status: 量測狀態功能

0: 除能

1: 始能，開始量測電流與溫度

Bit 0 Valid: 電流/溫度有效位

0: 未量測完成，數值無效

1: 已量測完成，數值有效

1. 手動始能 **Status** 位便會開始量測電流，需等待 **Valid** 位置位方可取值。

2. **Valid** 位會在讀取電流或溫度後清零。

● 量測資料暫存器

數值計算:每組數據由 4 個 Byte 組成，由高字節往低字節排列成一組 32 位元有符整數(int32)，再換算轉換成實際數值。

(1)Temperature 暫存器(04h~07h)

暫存器(04h)

Bit	31	30	29	28	27	26	25	24
Name	D31	D30	D29	D28	D27	D26	D25	D24

暫存器(05h)

Bit	23	22	21	20	19	18	17	16
Name	D23	D22	D21	D20	D19	D18	D17	D16

暫存器(06h)

Bit	15	14	13	12	11	10	9	8
Name	D15	D14	D13	D12	D11	D10	D9	D8

暫存器(07h)

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0

溫度: 實際值 =  $D[31:0] / 10$  (°C)

(2)DC Current 暫存器(08h~0Bh)

暫存器(08h)

Bit	31	30	29	28	27	26	25	24
Name	D31	D30	D29	D28	D27	D26	D25	D24

暫存器(09h)

Bit	23	22	21	20	19	18	17	16
Name	D23	D22	D21	D20	D19	D18	D17	D16

暫存器(0Ah)

Bit	15	14	13	12	11	10	9	8
Name	D15	D14	D13	D12	D11	D10	D9	D8

暫存器(0Bh)

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0

電流: 實際值 =  $D[31:0] / 1000$  (A)

### (3)AC Current 暫存器(0Ch~0Fh)

暫存器(0Ch)

Bit	31	30	29	28	27	26	25	24
Name	D31	D30	D29	D28	D27	D26	D25	D24

暫存器(0Dh)

Bit	23	22	21	20	19	18	17	16
Name	D23	D22	D21	D20	D19	D18	D17	D16

暫存器(0Eh)

Bit	15	14	13	12	11	10	9	8
Name	D15	D14	D13	D12	D11	D10	D9	D8

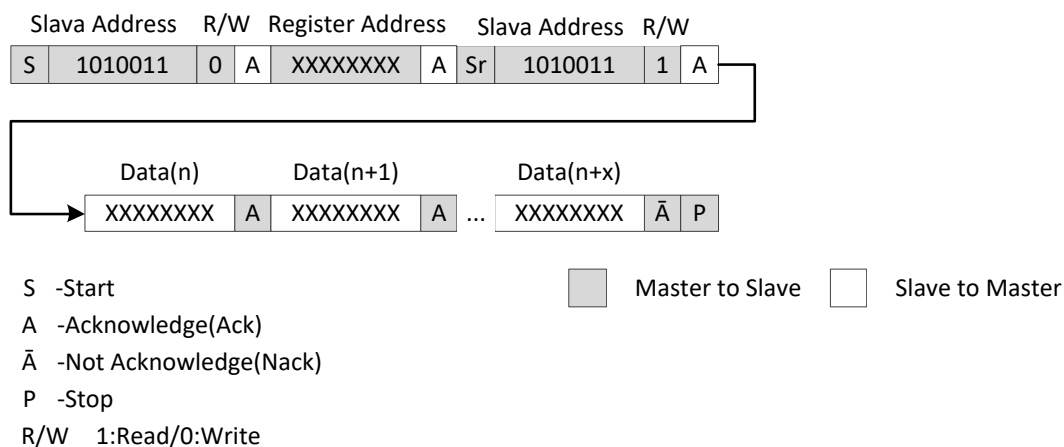
暫存器(0Fh)

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0

電流: 實際值 =  $D[31:0] / 1000$  (A)

## (5.2)發送讀取指令

※廣播位址(0x00)無法執行讀取指令。

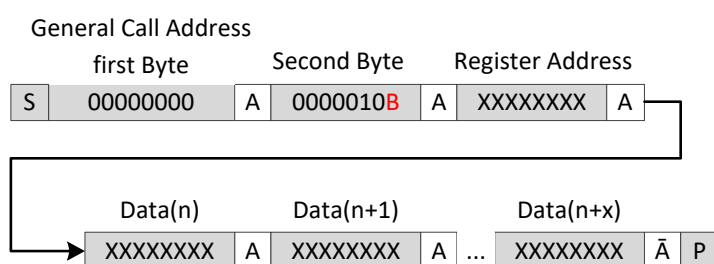


## (5.3)發送寫入指令

Slava Address	R/W	Register Address	Data(n)	Data(n+1)	Data(n+x)
S	1010011	0 A	XXXXXXXX A	XXXXXXXX A	... XXXXXXXX A P

S -Start  Master to Slave  Slave to Master  
 A -Acknowledge(Ack)  
 $\bar{A}$  -Not Acknowledge(Nack)  
 P -Stop  
 R/W 1:Read/0:Write

## (5.4)廣播模式(0x00)

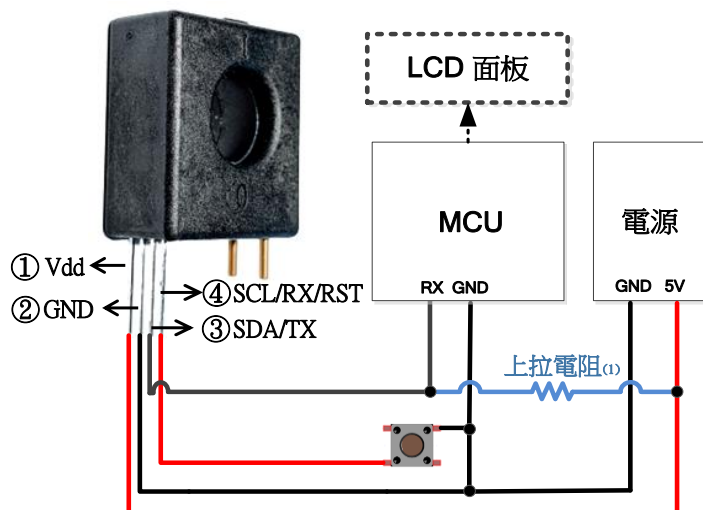


S -Start  Master to Slave  Slave to Master  
 A -Acknowledge(Ack)  
 $\bar{A}$  -Not Acknowledge(Nack)  
 P -Stop  
 R/W 1:Read/0:Write  
 First Byte : General Call Address  
 Second Byte(04h) : System does not reset when writing data

- 第二個字節最低位 B 為 0:  
 可通過第二個字節 04h 寫數據到從機內部，過程中不會復位硬體，並接收主機發送的數據。  
 可通過第二個字節 06h 寫數據到從機內部，過程中復位硬體，並接收主機發送的數據。(不使用)
- 第二個位元組最低位 B 為 1: 為硬體廣播(不使用)

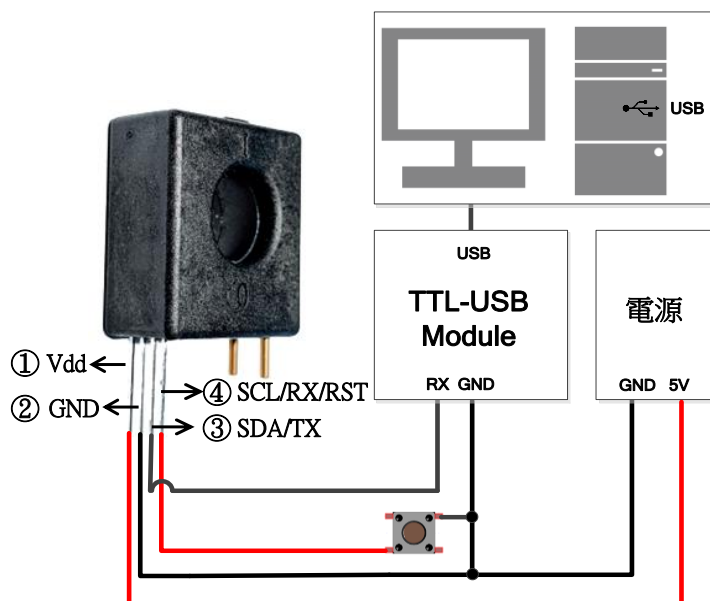
## 6. 應用電路(連續版別):

- 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。

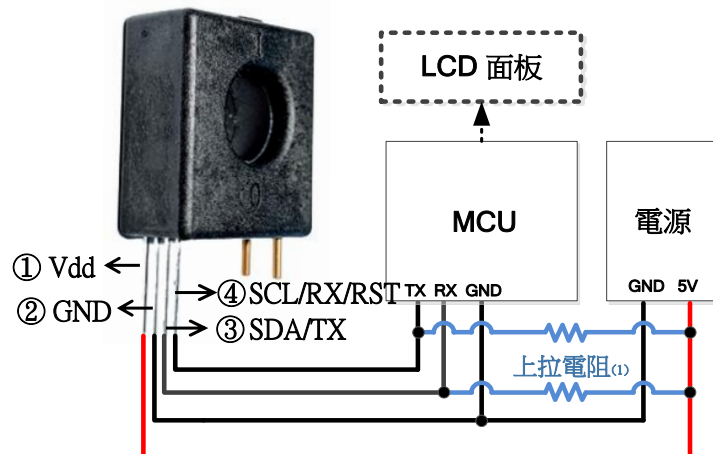
- TTL to USB Module 接線圖



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

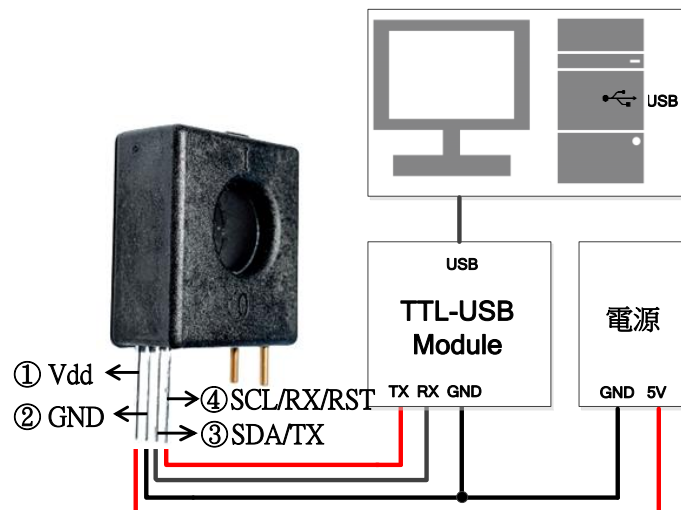
## 7. 應用電路(AT 指令 & Modbus-RTU):

- MCU 接線圖



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

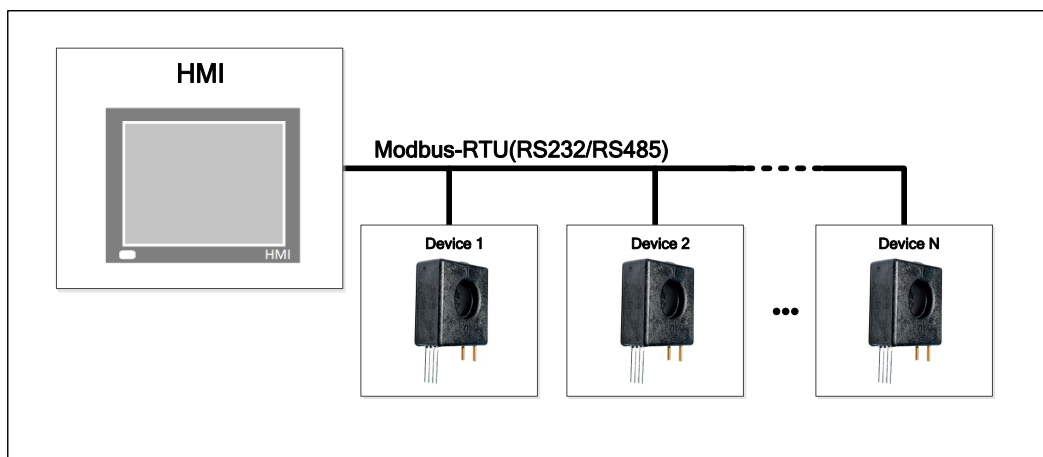
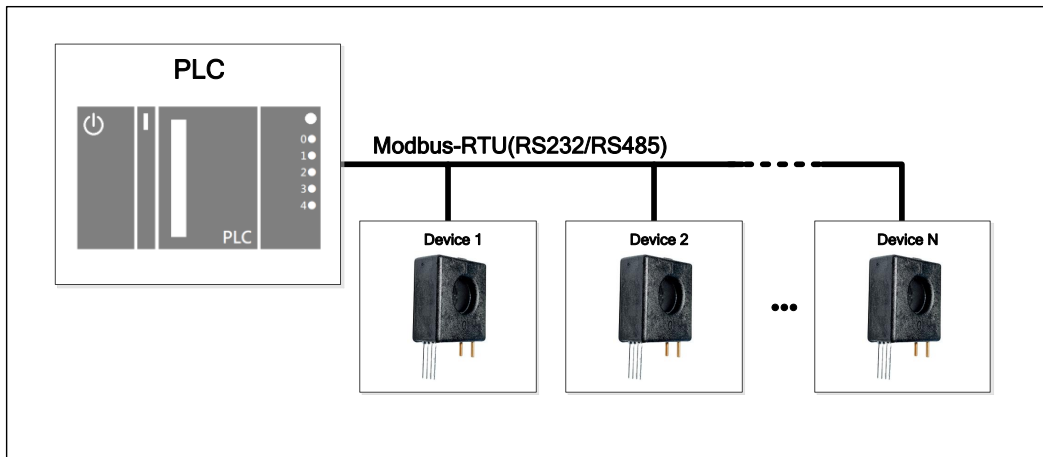
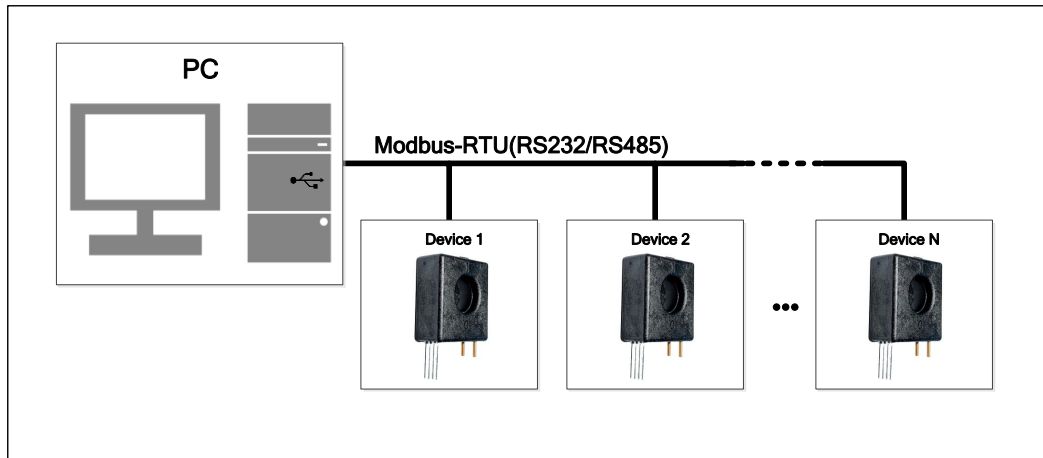
- TTL to USB Module 接線圖



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



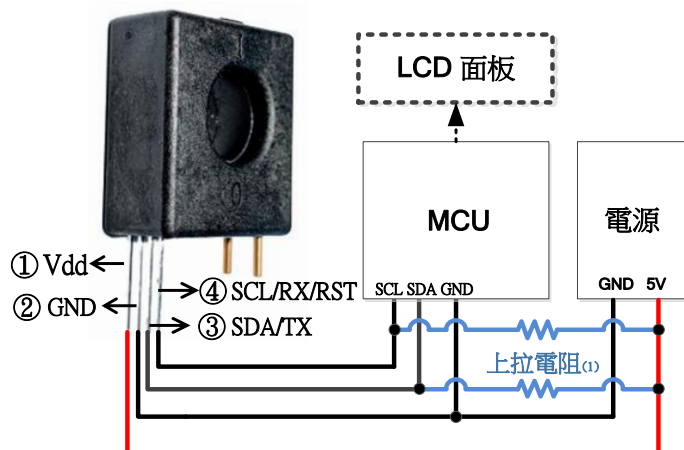
● Modbus-RTU 架構圖：



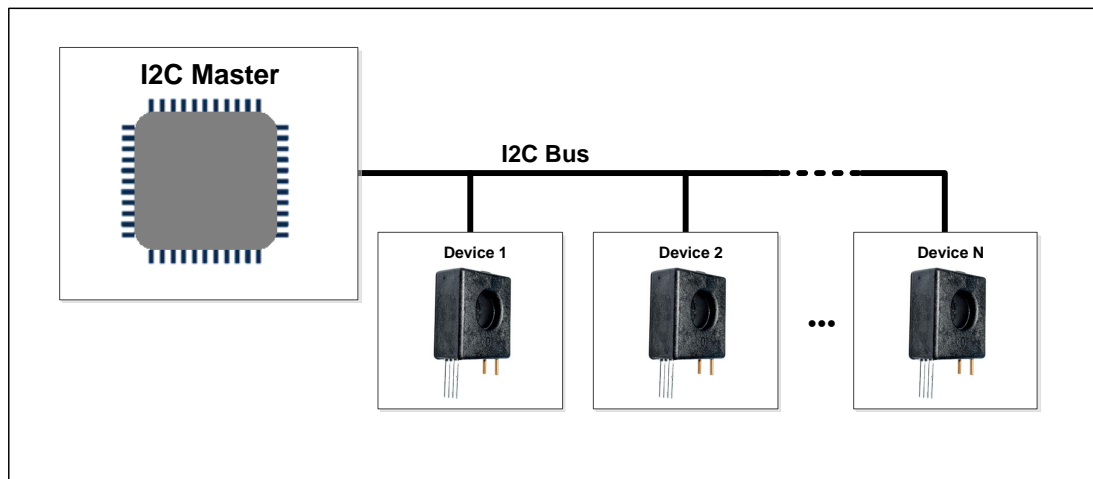
Winson reserves the right to make changes to improve reliability or manufacturability.

## 8. 應用電路(I2C):

- I2C 接線圖&架構圖:



(2) DWCS 的 SDA/SCL 輸出為 Open Drain 配置，須使用上拉電阻，如果 MCU Master 端 SDA/SCL 腳位已內部上拉，則此電阻可以不加。

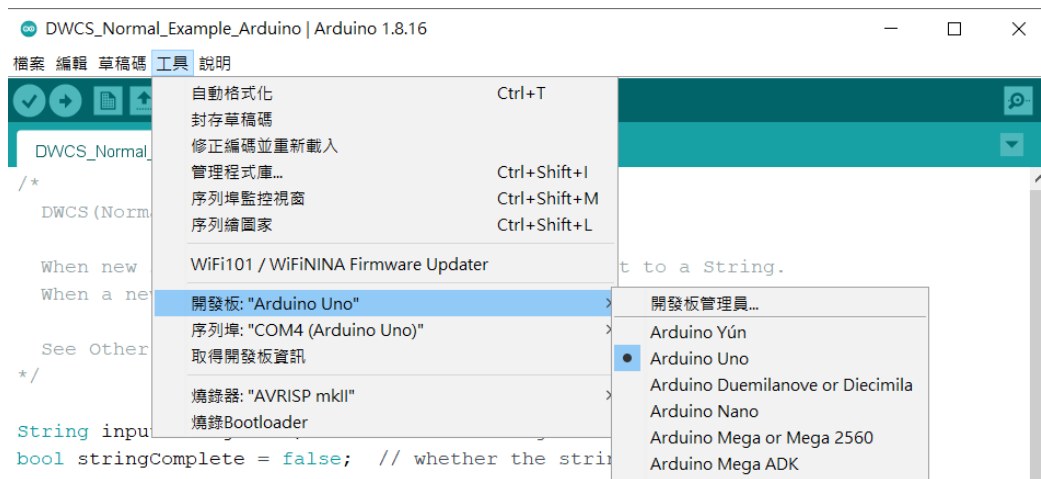


Winson reserves the right to make changes to improve reliability or manufacturability.

## Arduino 範例程式

### 1. 使用 Arduino 平台須知

- **工具\_開發板:** 選取 Arduino 的板別



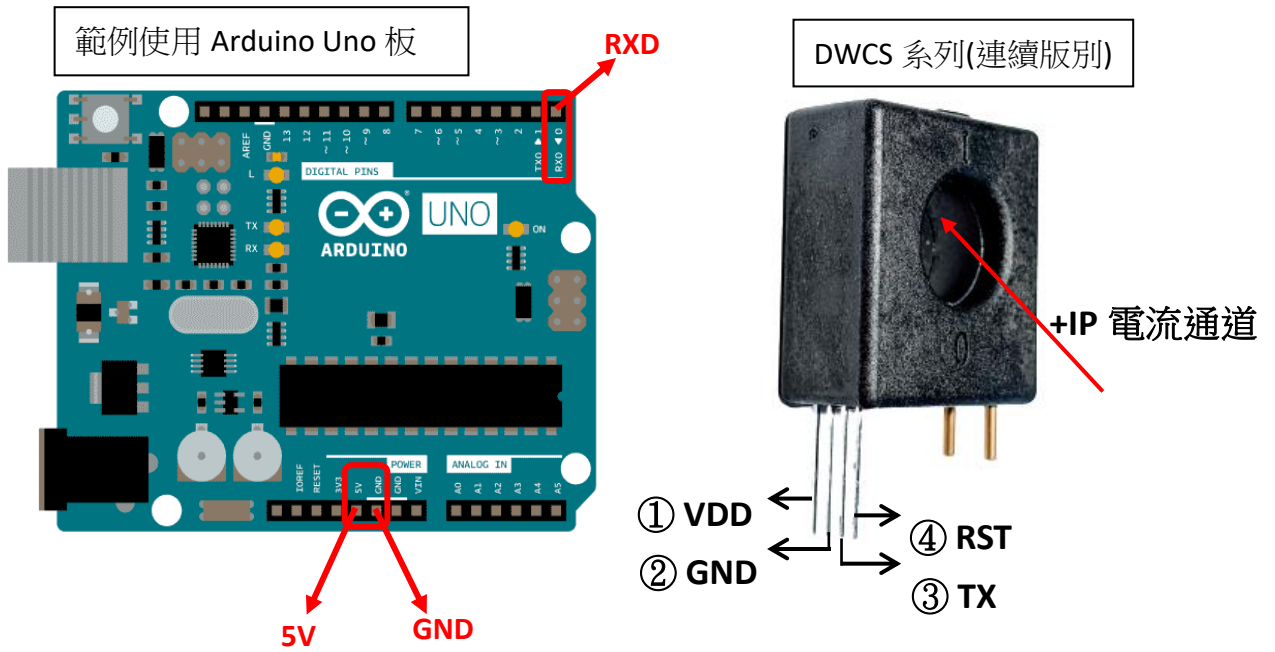
- **工具\_序列埠:** 選取 Arduino 的序列埠位置



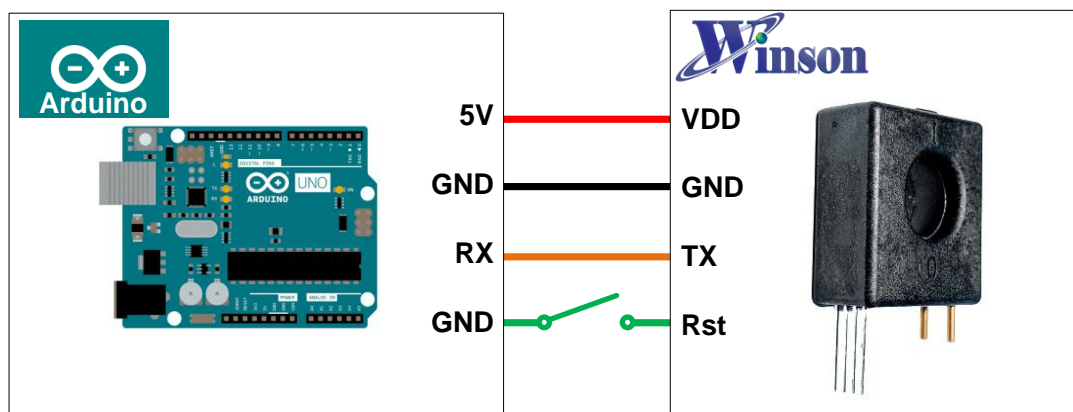
※Arduino 序列埠位置可使用 Windows 的裝置管理員查詢。

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

## 2. 連續版別

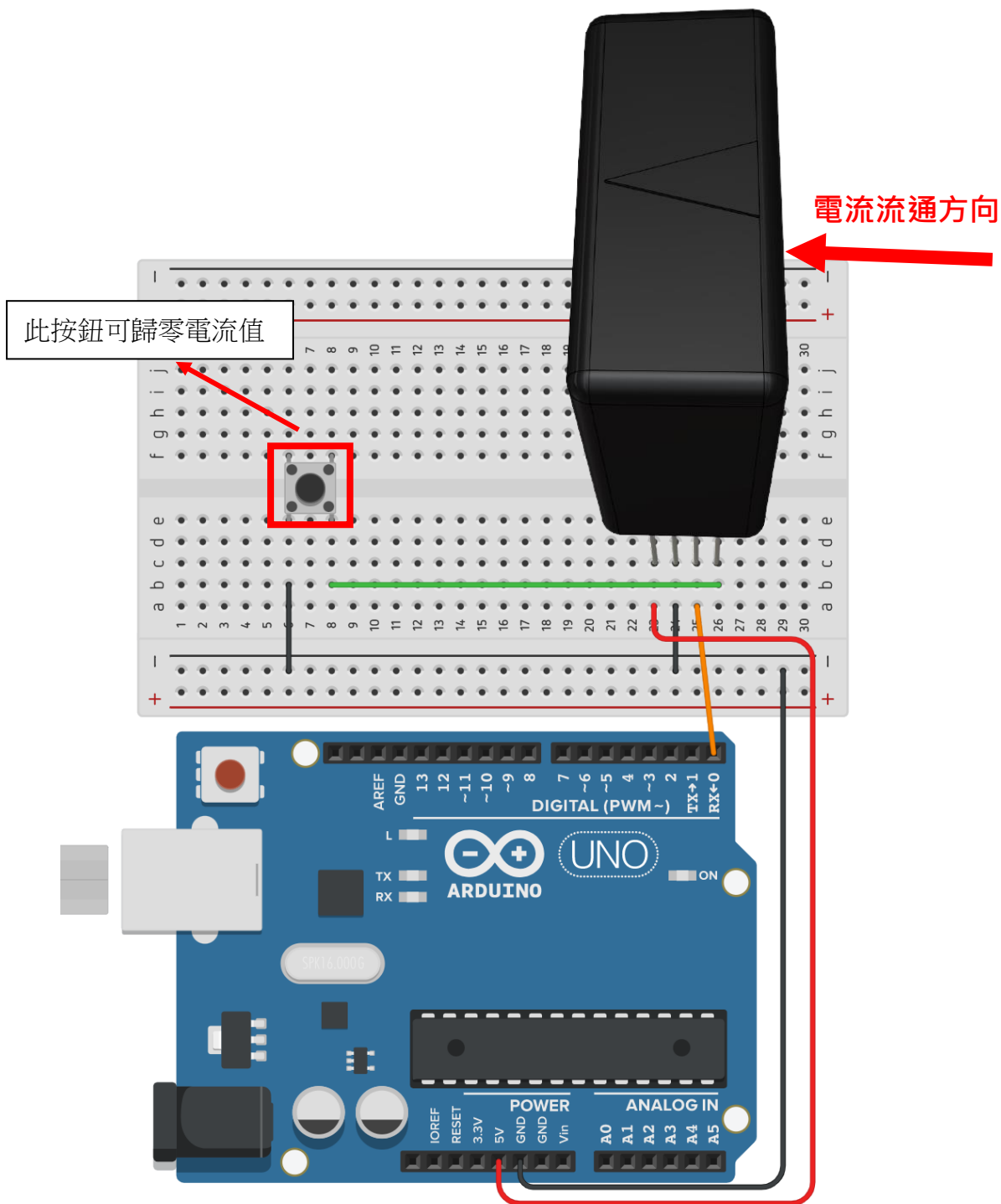


### ● 接線示意圖



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

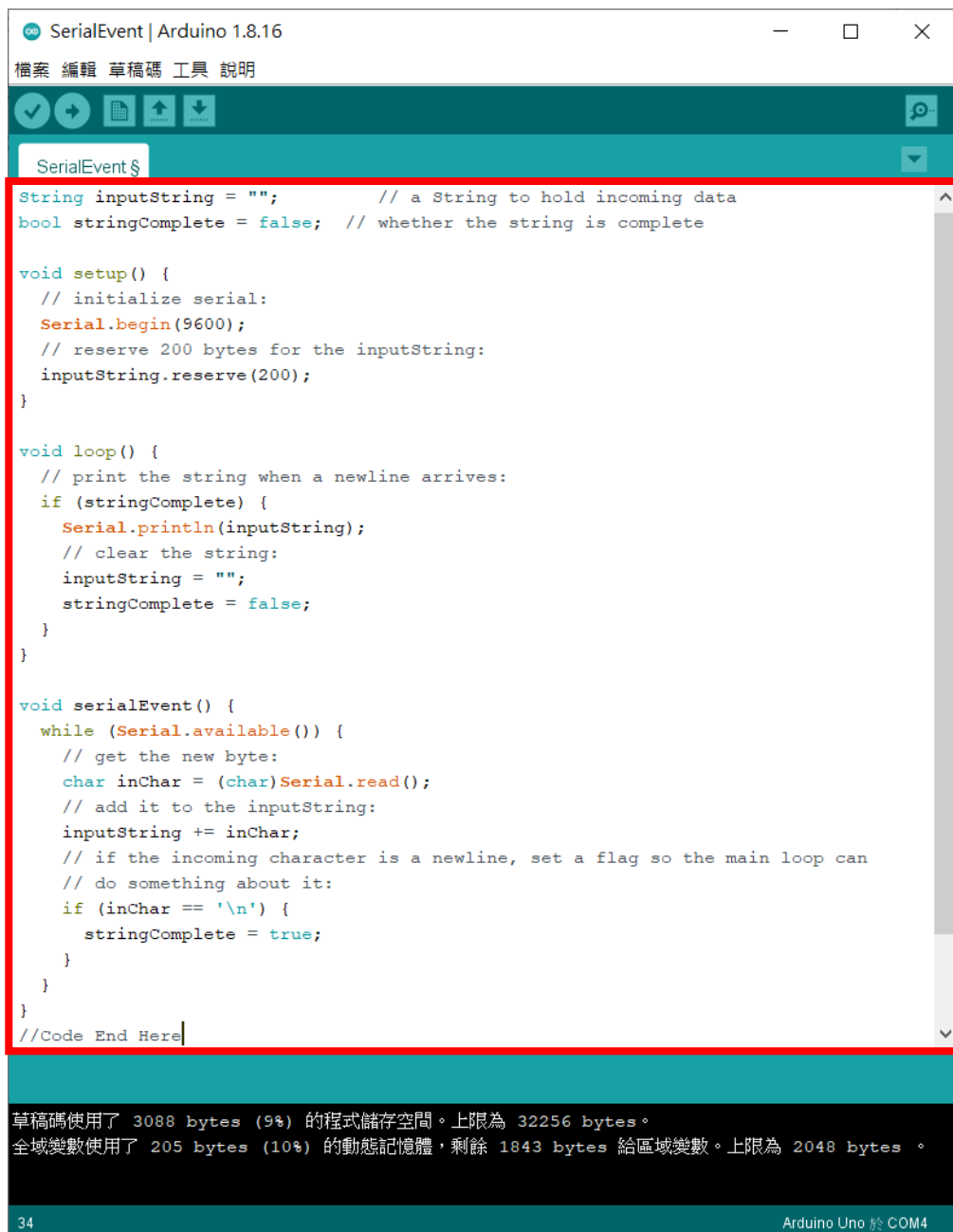
- 接線圖



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

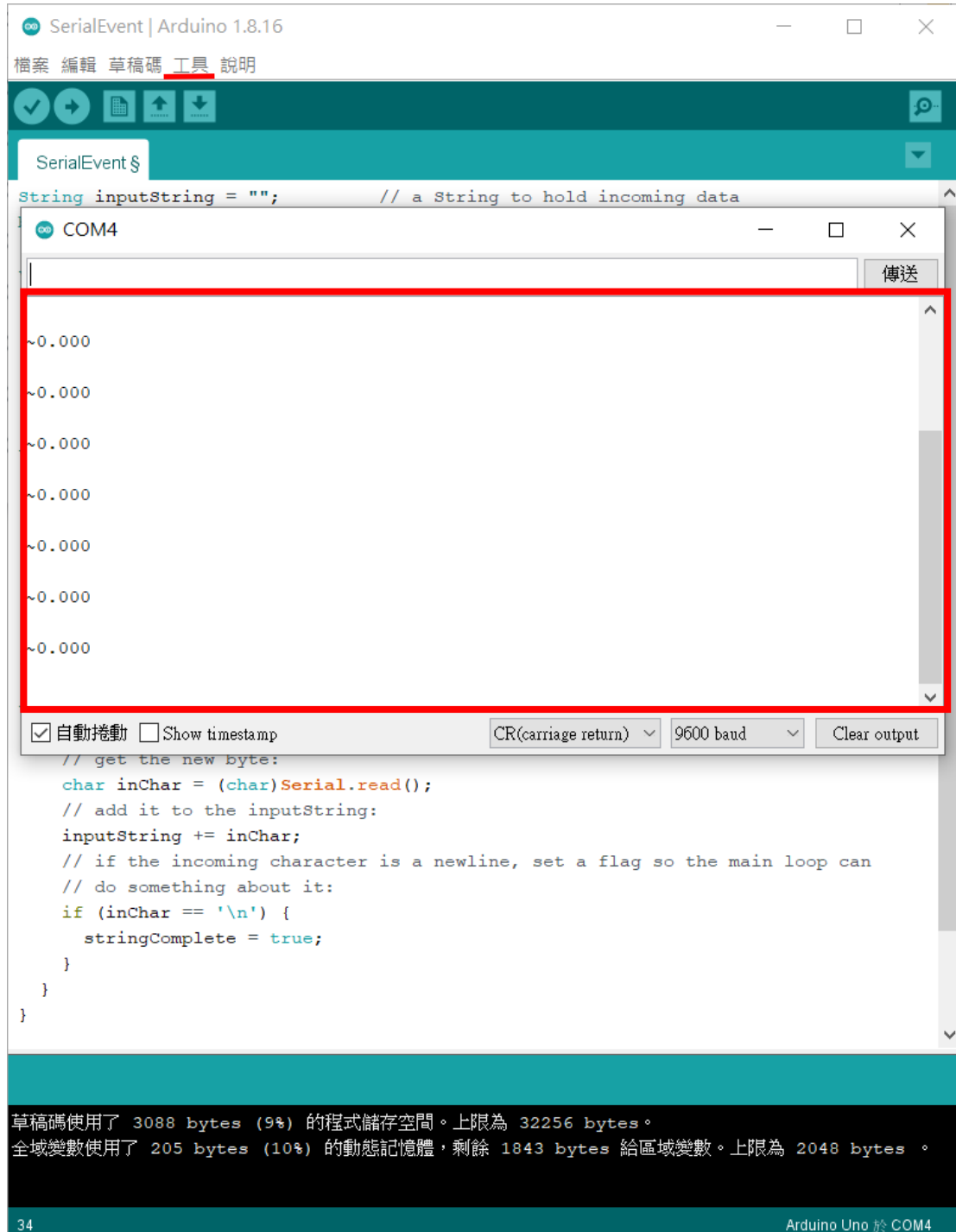
草稿碼使用了 3088 bytes (9%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 205 bytes (10%) 的動態記憶體，剩餘 1843 bytes 給區域變數。上限為 2048 bytes。

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。

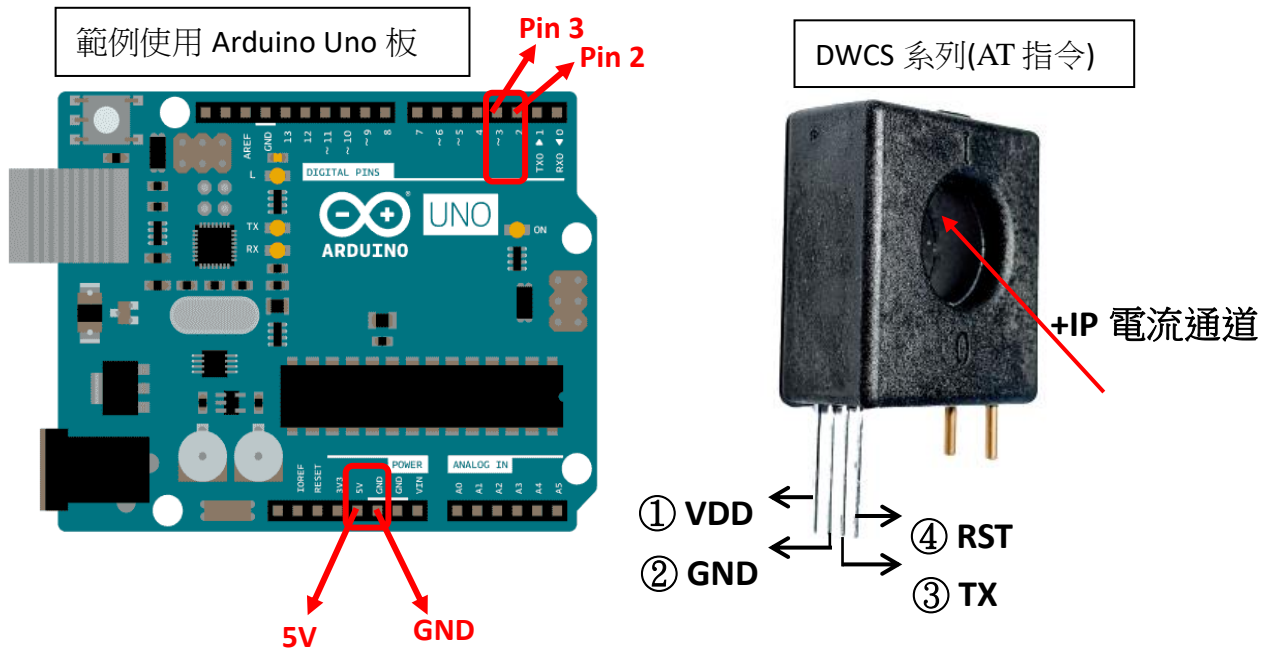
Winson reserves the right to make changes to improve reliability or manufacturability.

(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。

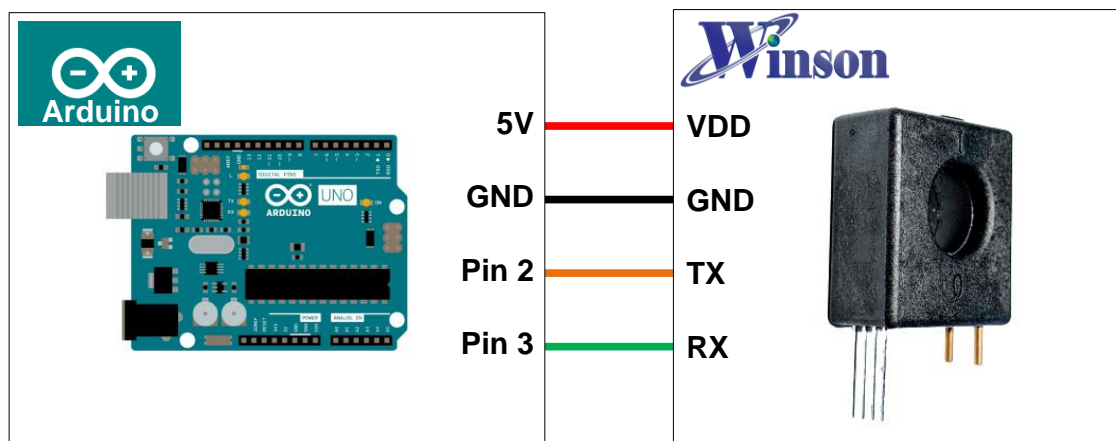


Winson reserves the right to make changes to improve reliability or manufacturability.

## 3. AT 指令版本



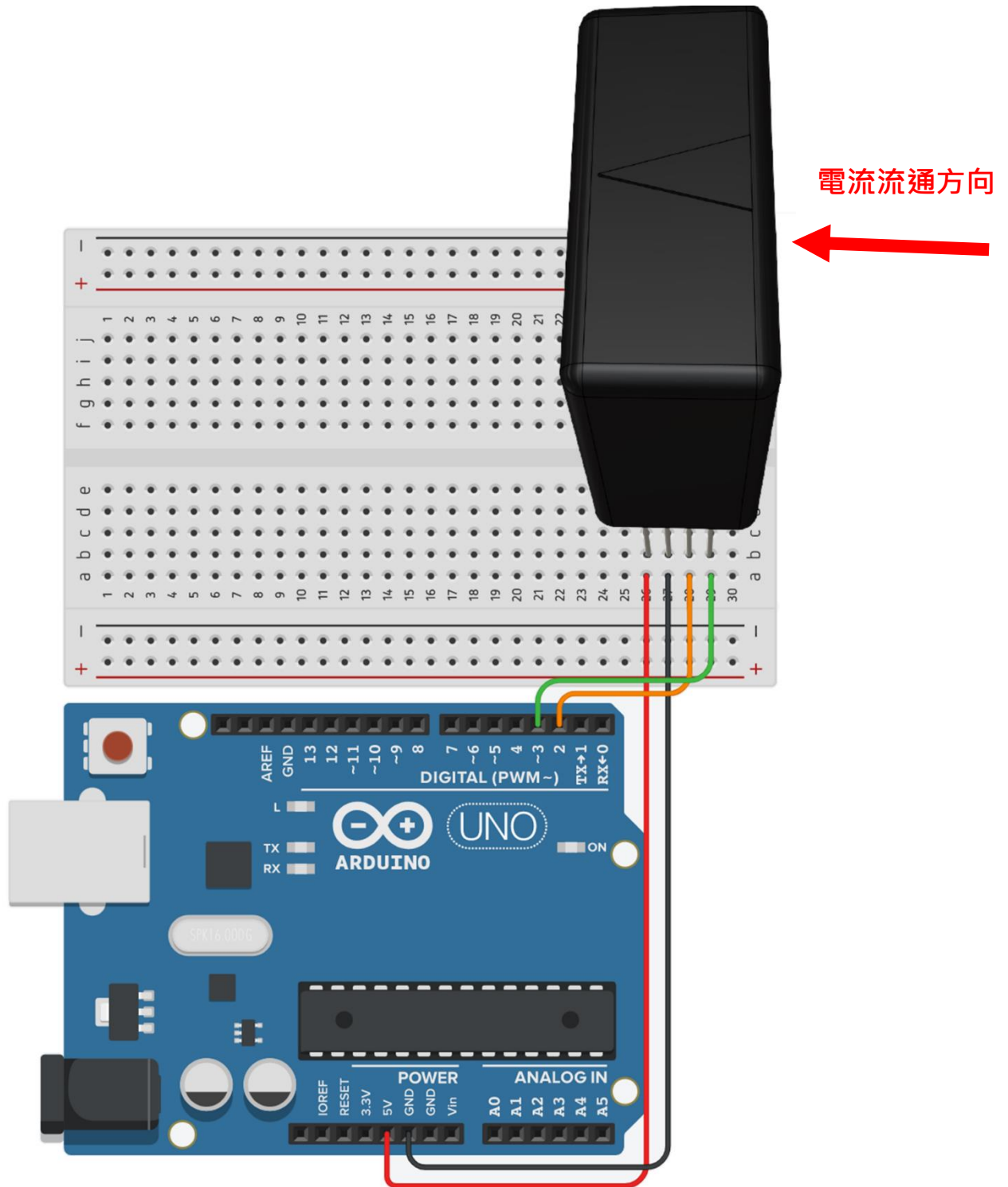
### ● 接線示意圖



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
```

輸入指令  
指令表於 Page.2

※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。

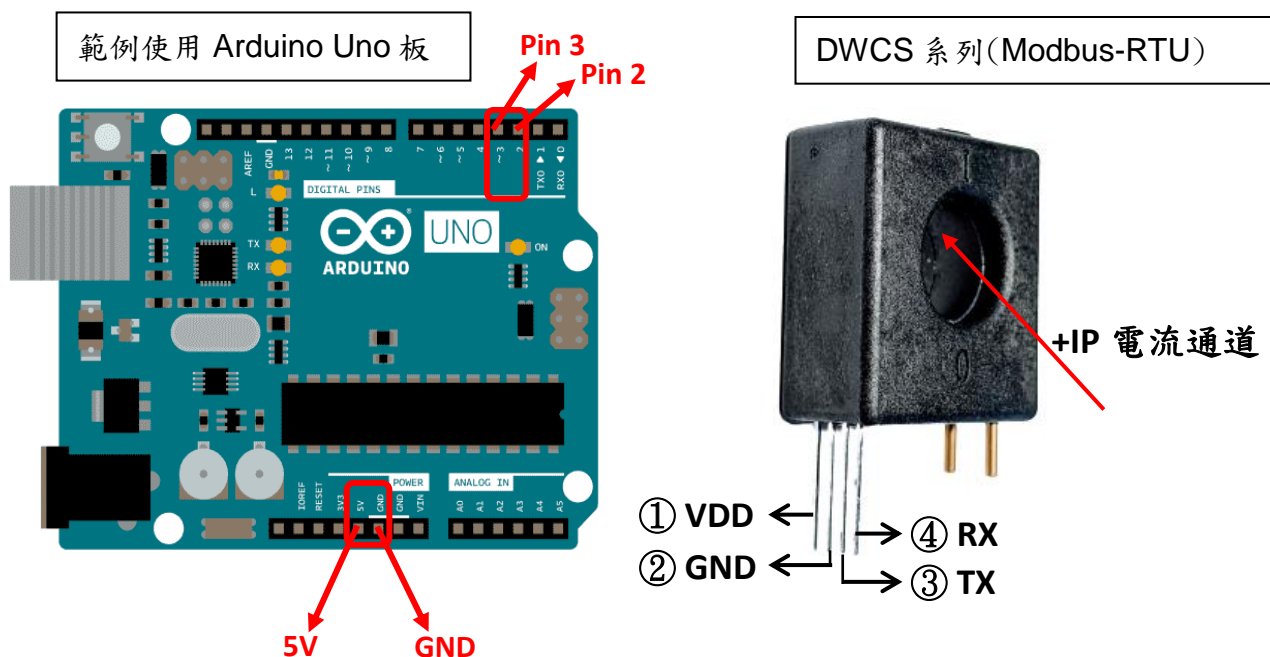
Winson reserves the right to make changes to improve reliability or manufacturability.

(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。

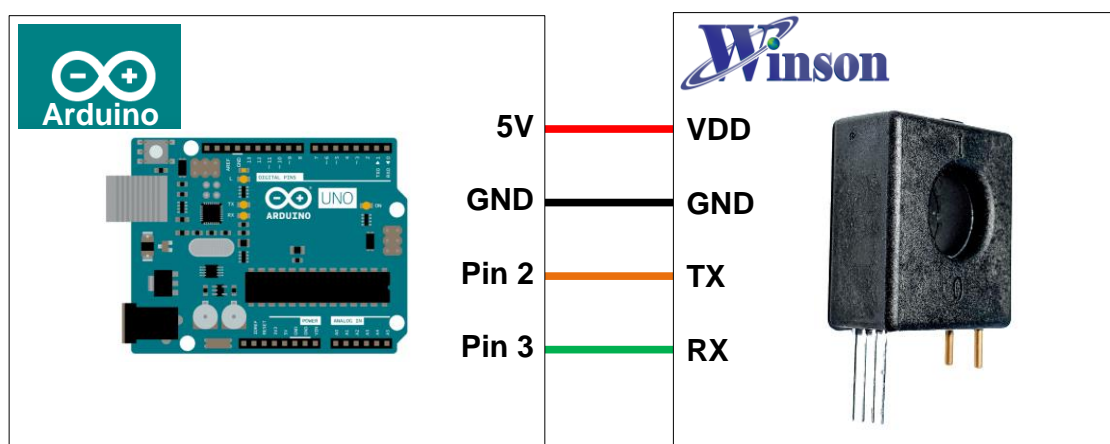


Winson reserves the right to make changes to improve reliability or manufacturability.

## 4. Modbus-RTU 版別(修改地址&量測溫度)

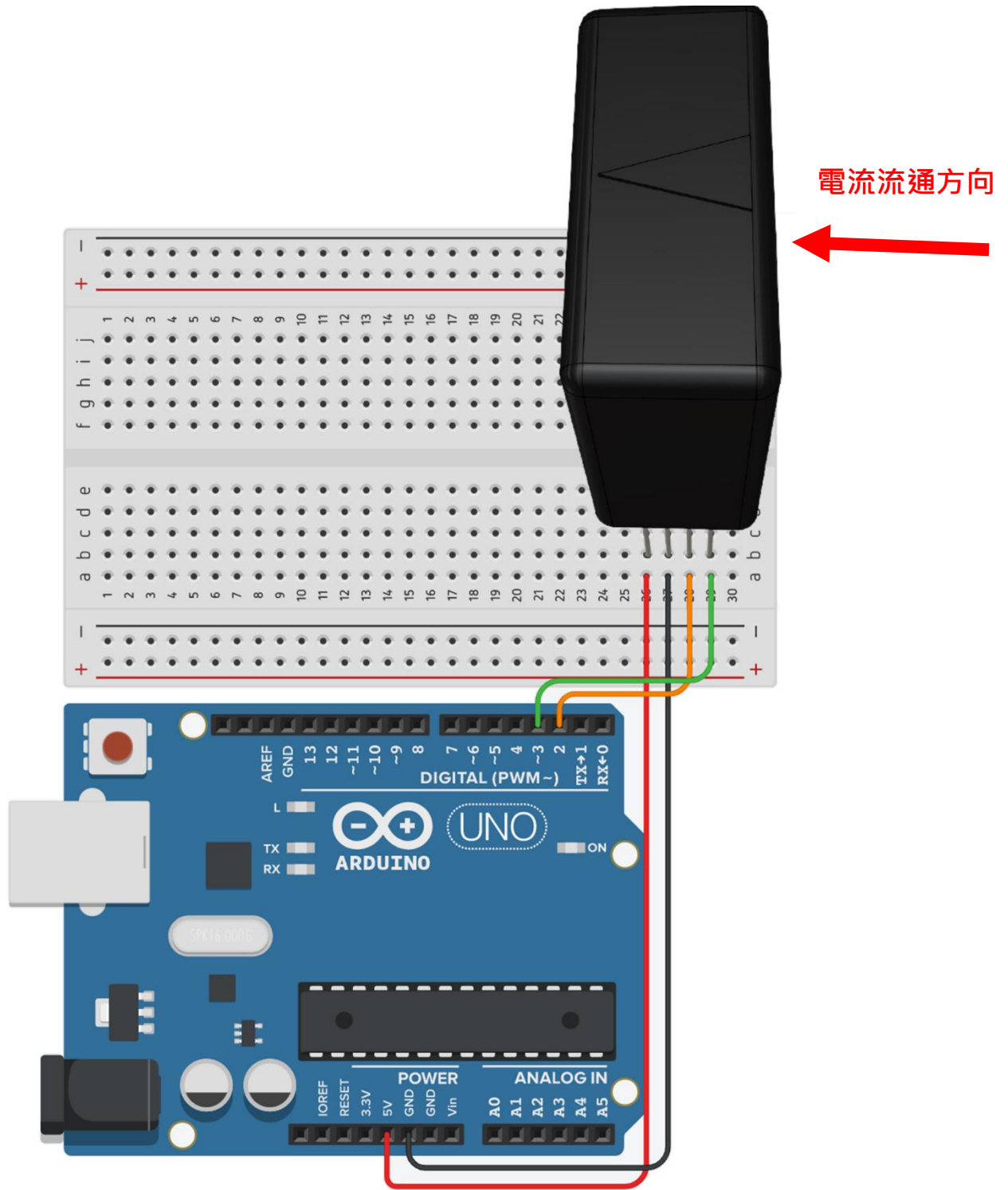


● 接線示意圖



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:

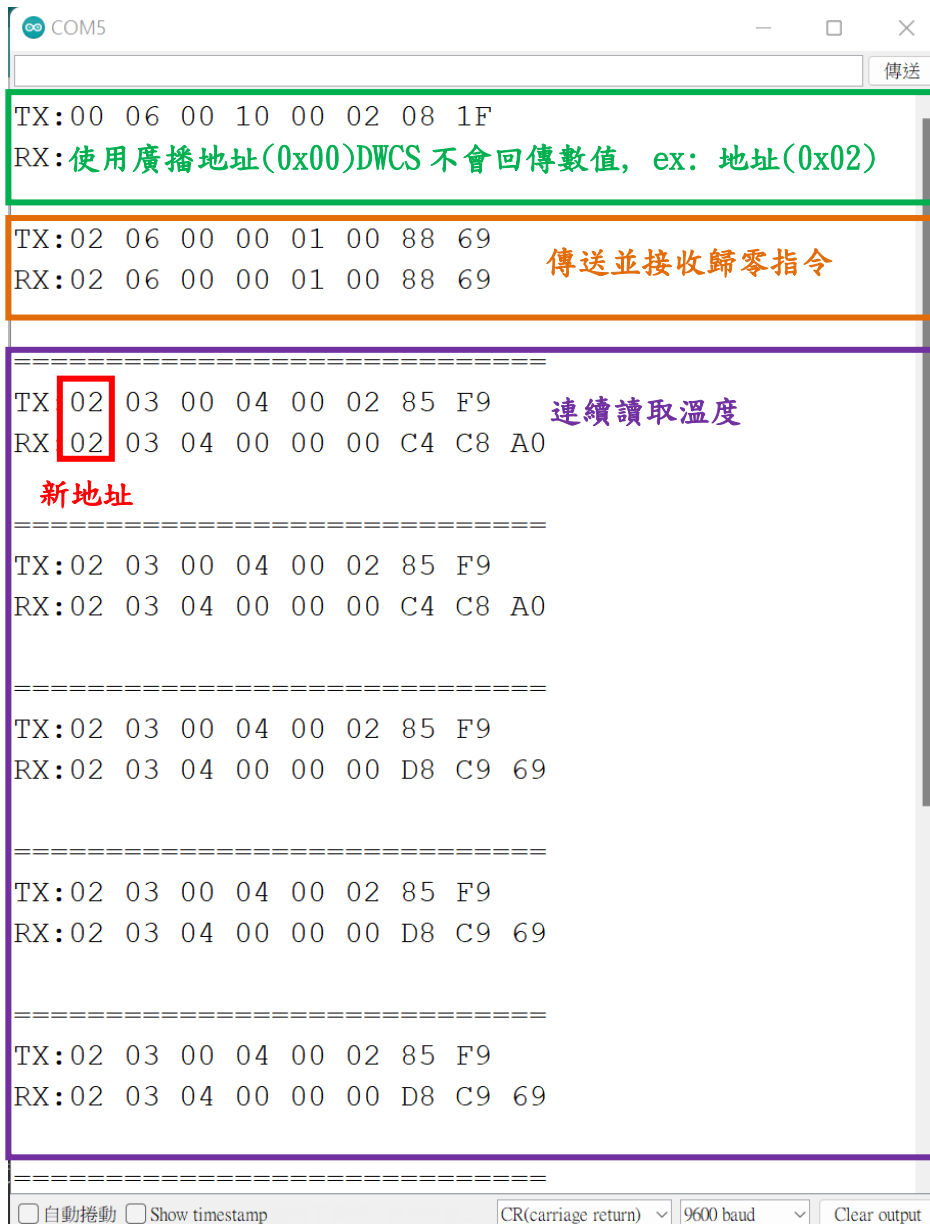
```

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。



The screenshot shows a serial port monitoring window titled 'COM5'. It displays several lines of TX and RX data. The first line shows TX:00 06 00 10 00 02 08 1F and RX: 使用廣播地址(0x00)DWCS 不會回傳數值, ex: 地址(0x02). The second line shows TX:02 06 00 00 01 00 88 69 and RX:02 06 00 00 01 00 88 69, with an annotation '傳送並接收歸零指令'. The third line shows TX:02 03 00 04 00 02 85 F9 and RX:02 03 04 00 00 00 C4 C8 A0, with an annotation '連續讀取溫度' and '新地址' below it. The window has a '傳送' button and a status bar at the bottom with options for '自動捲動', 'Show timestamp', 'CR(carriage return)', '9600 baud', and 'Clear output'.

```
COM5  
TX:00 06 00 10 00 02 08 1F  
RX: 使用廣播地址(0x00)DWCS 不會回傳數值, ex: 地址(0x02)  
TX:02 06 00 00 01 00 88 69  
RX:02 06 00 00 01 00 88 69 傳送並接收歸零指令  
=====
```

```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 C4 C8 A0  
新地址  
=====
```

```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 C4 C8 A0  
=====
```

```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 D8 C9 69  
=====
```

```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 D8 C9 69  
=====
```

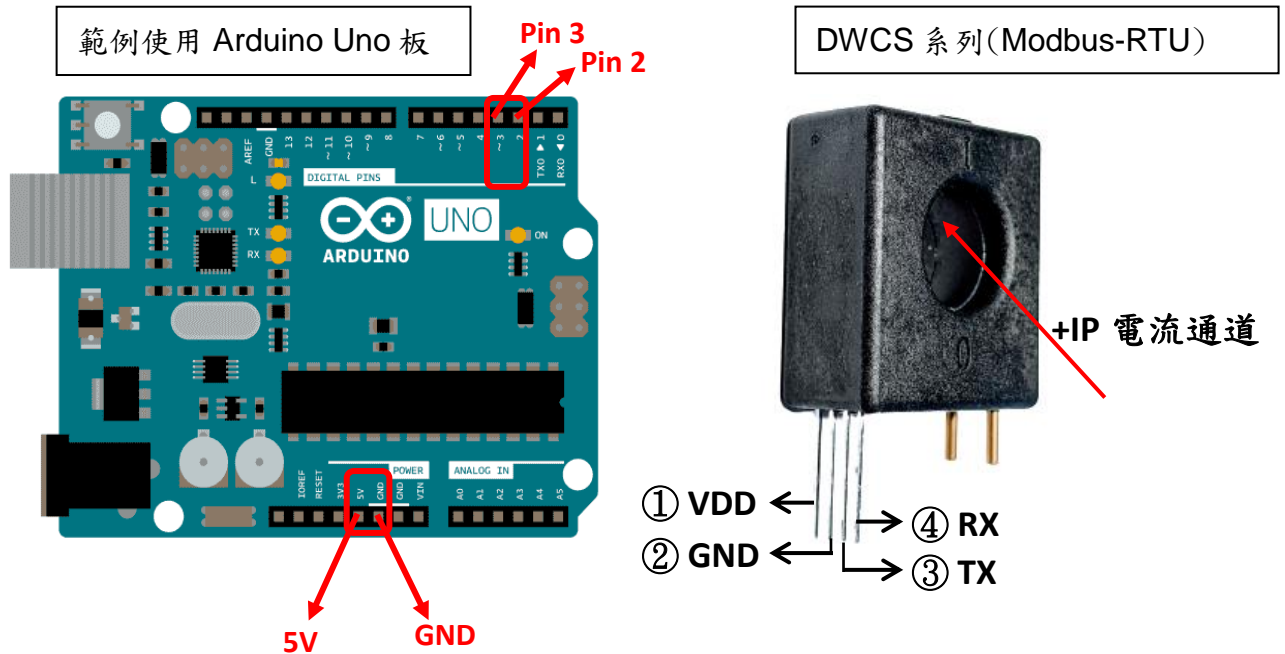
```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 D8 C9 69  
=====
```

```
TX:02 03 00 04 00 02 85 F9  
RX:02 03 04 00 00 00 D8 C9 69  
=====
```

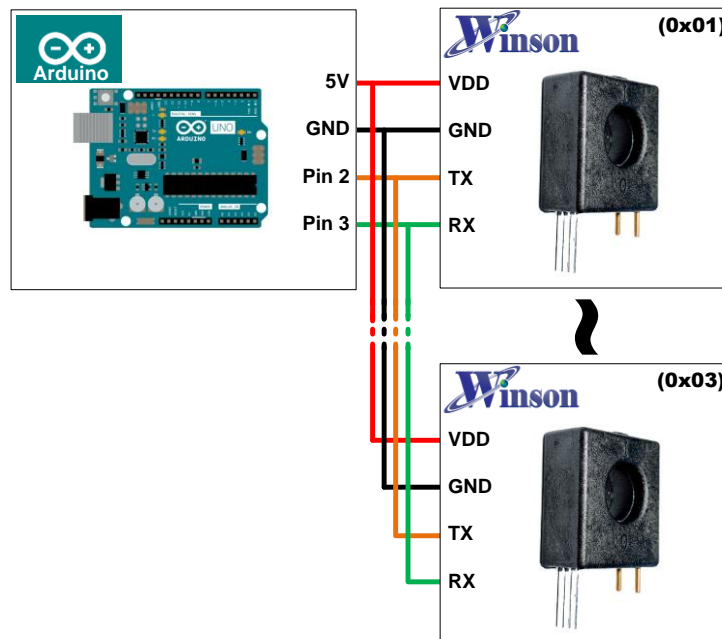
自動捲動  Show timestamp CR(carriage return) 9600 baud Clear output

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

## 5. Modbus-RTU 版別(一對多通訊)



### ● 接線示意圖

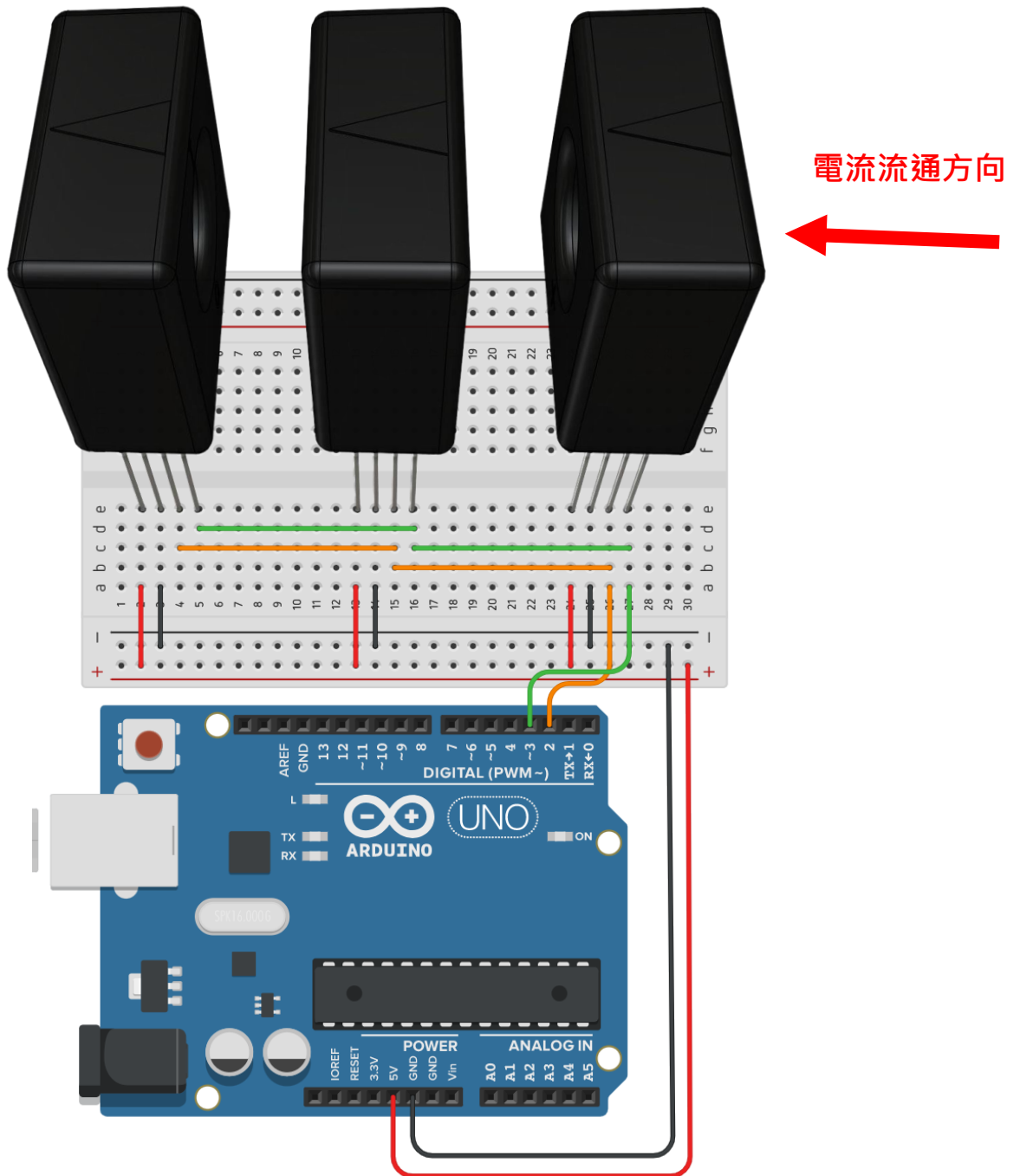


※每個 DWCS 位址不可重複(更改位址請看 [上例](#))

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>

```
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
  Serial.println("=====");
  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：

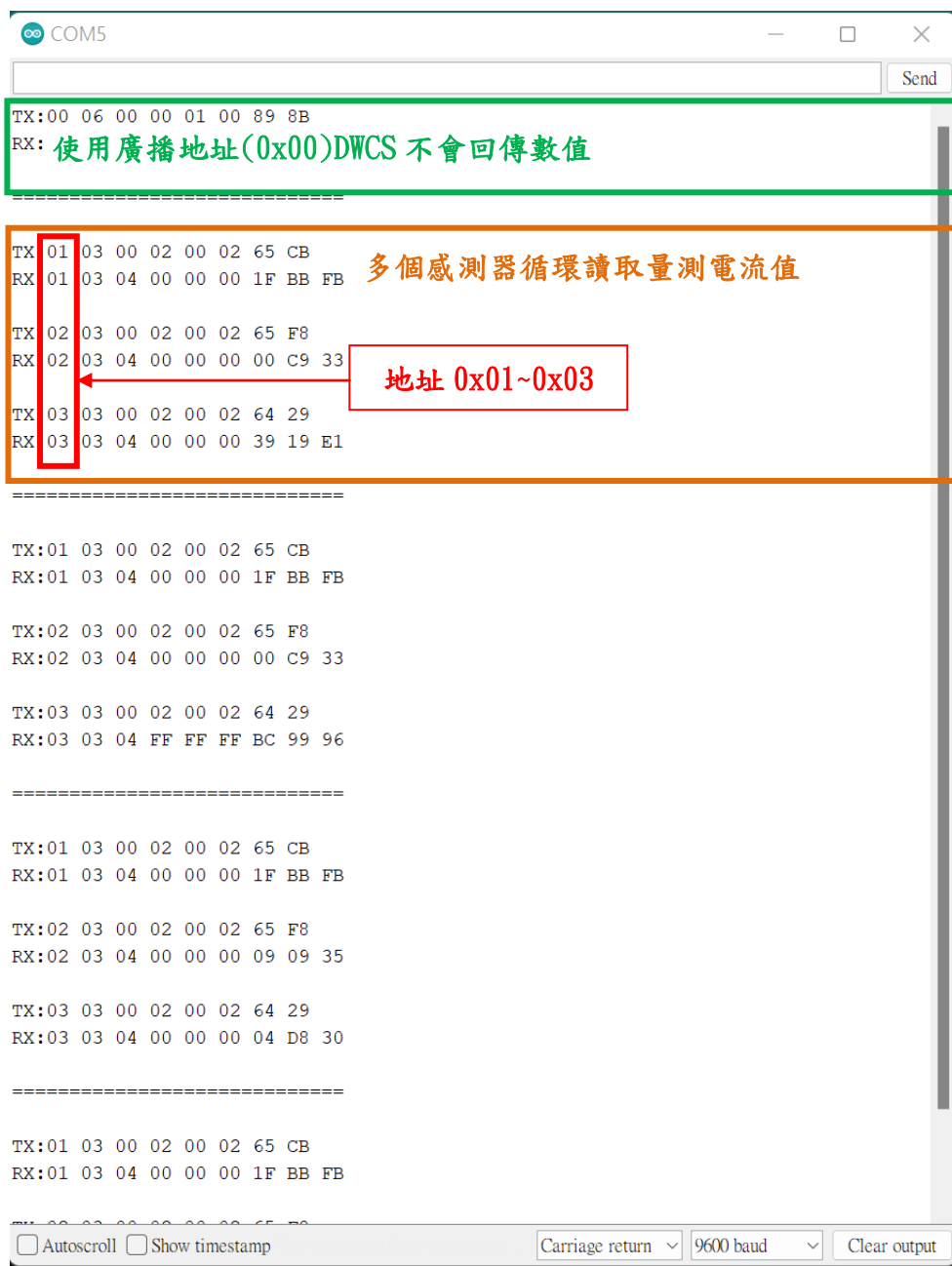
```
void WriteCommand(byte SlaveAddress,byte FunctionCode,word DeviceAddress,word RegisterNum)
```

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。



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