

目錄

1. Arduino 函式庫安裝步驟:	2
2. Arduino 執行程式步驟:	6
3. DWCS:	7
(3.1) CONTINUOUS MODE (連續模式):	7
(3.2) AT COMMAND MODE (指令模式):	8
(3.3) MODBUS-RTU MODE (一對一):	9
(3.4) MODBUS-RTU MODE (一對多):	9
4. WCM:	11
(4.1) CONTINUOUS MODE (連續模式):	11
(4.2) MODBUS-RTU MODE (一對一):	12
(4.3) MODBUS-RTU MODE (一對多):	13
5. WCS:	14
(5.1) SINGLE OUTPUT: DC (單輸出 DC 電流):	14
(5.2) DIFFERENTIAL OUTPUT: DC(差動輸出 DC 電流):	14
(5.3) SINGLE OUTPUT: AC(單輸出 AC 電流):	15
(5.4) DIFFERENTIAL OUTPUT: AC(差動輸出 AC 電流):	15
6. Hall IC:	16
(6.1) SINGLE OUTPUT: SWITCHING (單輸出開關):	16
(6.2) DUAL OUTPUT: SWITCHING (雙輸出開關):	16
7. 函式說明	17
(7.1) DWCS 類別	17
(7.2) WCM 類別	21
(7.3) WCS 類別	25

WinsonLib: Arduino Library 使用說明

1. Arduino 函式庫安裝步驟:

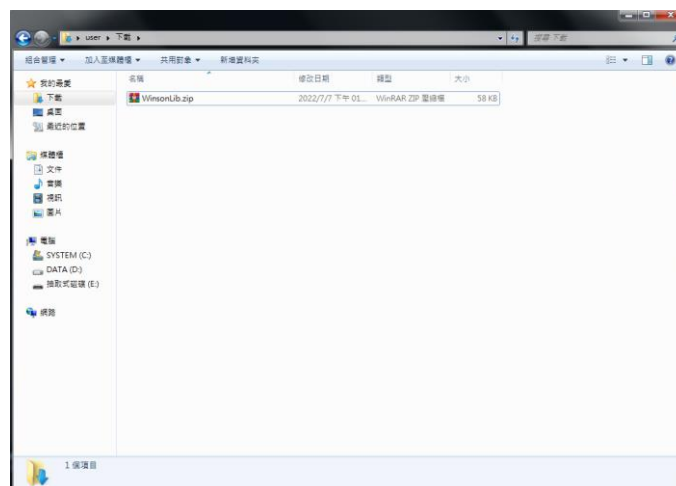
安裝方式有兩種: 1. Winson 官網下載安裝, 2. Arduino IDE 管理員程式庫安裝。

使用的 Arduino 函式庫安裝文件, 以 Winson 公司官網資料的函式庫為範例, 進行安裝。函式庫包含 Winson 公司所開發大部分產品, 如: WCS、WCM、DWCS、Hall IC 之函式供用戶開發使用。

如下圖所示, 本文使用 WinsonLib 函式庫來當範例, 請點選 Winson Library (WinsonLib)函式庫。

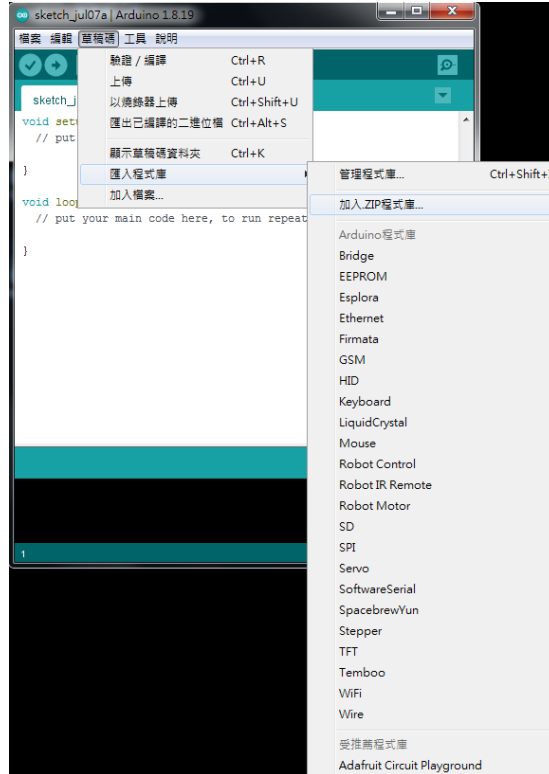


如下圖所示, 下載 WinsonLib 函式庫完成後, 一般而言, 都會在系統的下載目錄區內。

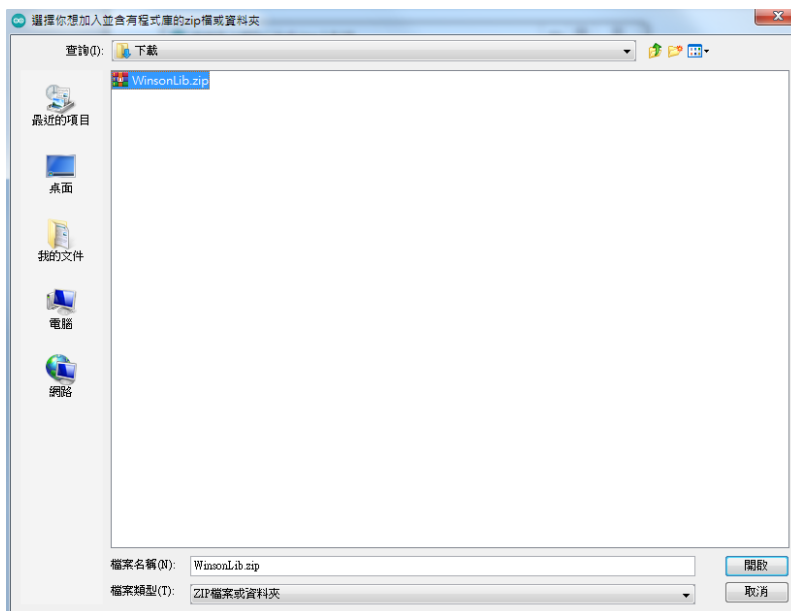


Winson reserves the right to make changes to improve reliability or manufacturability.

如下圖所示，進入 Arduino 開發版的開發工具：Sketch IDE 整合環境中選擇下載加入新的壓縮檔型的函式庫。

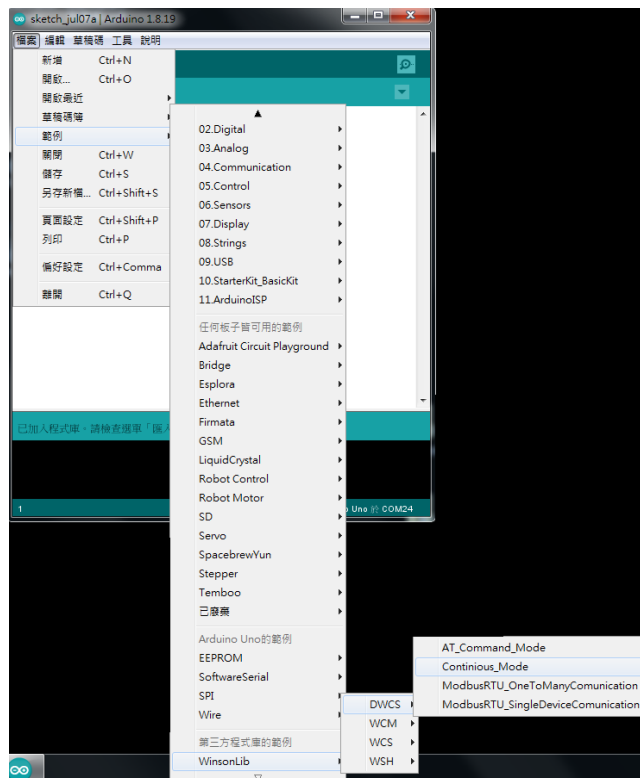


如下圖所示，選擇剛才下載函式庫壓縮檔的目錄，本範例為系統下載目錄。



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

如下圖所示，進行安裝 WinsonLib 是否成功安裝，我們使用安裝範例的方法測試。

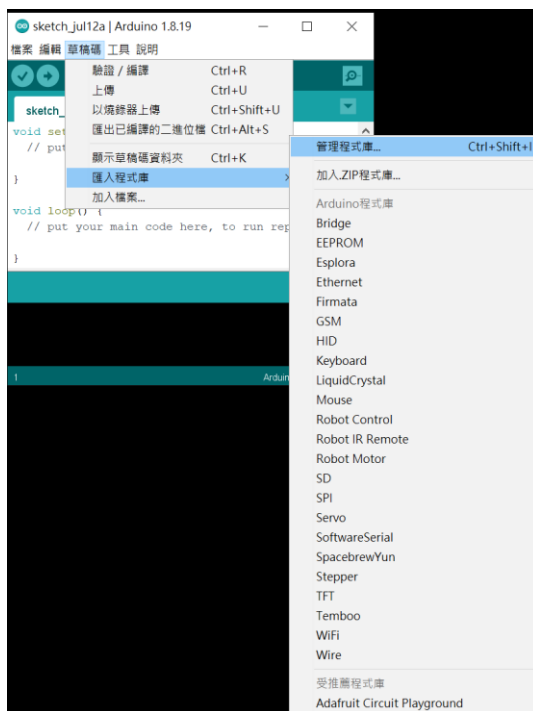


如下圖所示，如果可以正常使用 WinsonLib 範例程式，代表函式庫正確安裝。

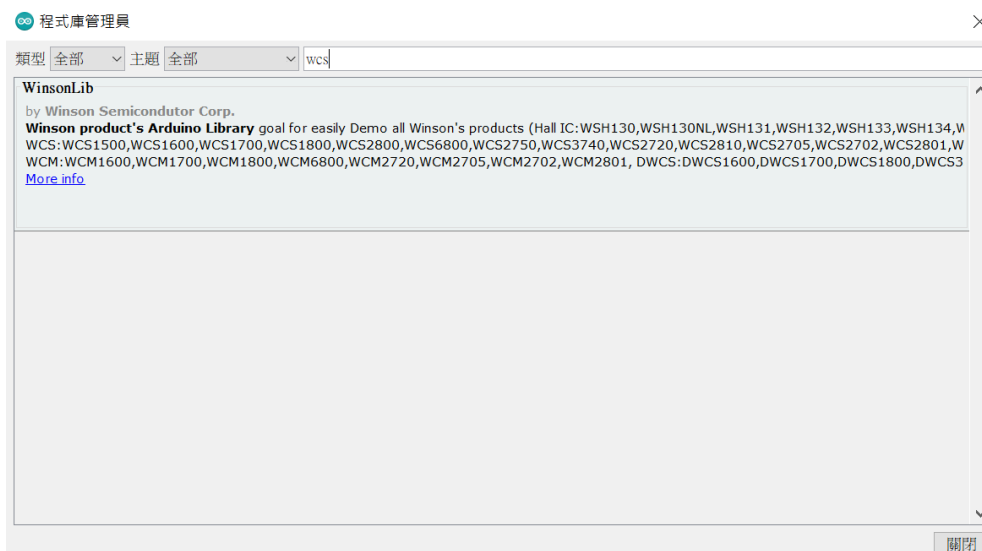


Winson reserves the right to make changes to improve reliability or manufacturability.

另外一種方式，以 Arduino IDE 直接安裝使用，本文使用 Arduino IDE 裡的程式庫管理員安裝來當範例。如下圖所示，進到 Arduino 開發版的開發工具選擇管理程式庫。




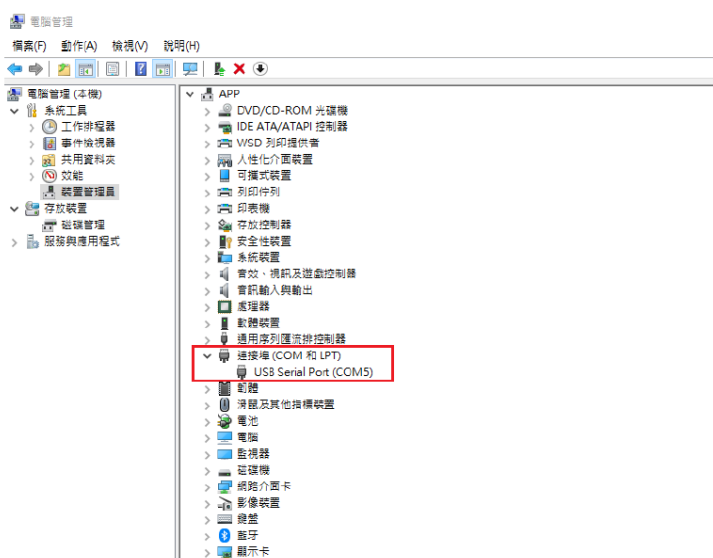
如下圖所示，進入程式庫管理員並搜尋: Winson、WCS、WCM 或 DWCS 等等都能找到 WinsonLib 程式庫。點選並執行安裝，使用上一頁安裝範例測試。




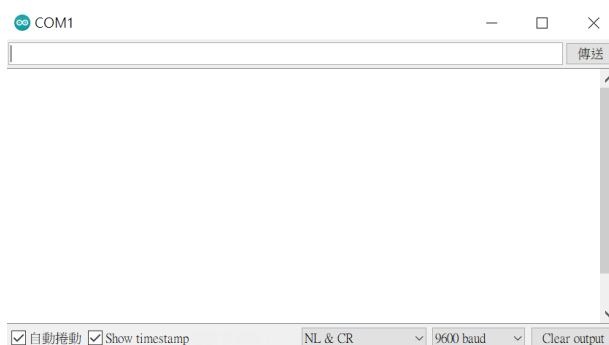
Winson reserves the right to make changes to improve reliability or manufacturability.

2. Arduino 執行程式步驟:

1. 開啟 Arduino IDE: 
2. 開啟程式: 檔案→範例→WinsonLib→DWCS→Continuous_Mode
3. 選擇板子: 工具→開發板:→Arduino Uno
4. 選擇 COM Port: 工具→序列埠:→COMn
5. 確定 電腦 裝置管理員→連接埠 是否有偵測到 USB Serial Port (COMn)



6. 上傳程式: 
7. 執行結果(監控視窗): 工具→序列埠監控視窗

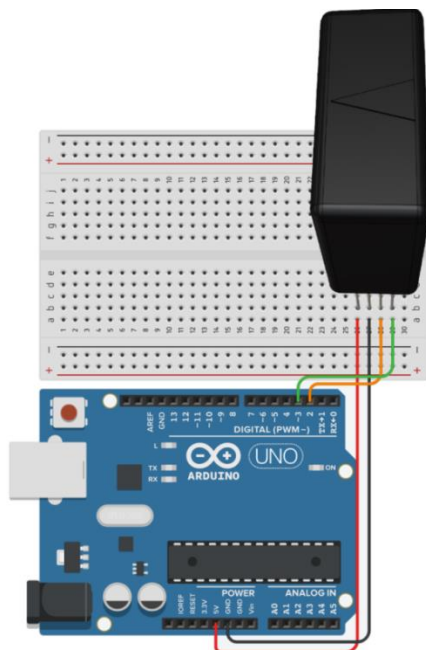


<註>Arduino 預設只有一個序列埠(系統序列埠)，本例子不使用系統序列埠(第 0 和 1 腳)，而是將透過 SoftwareSerial(軟體序列埠)程式庫，把其他接腳設為序列埠。Serial Monitor →Raud Rate : 9600 bps

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

3. DWCS:

連續模式、指令模式和一對一 Modbus-RTU 模式接線圖如下，把 DWCS 電流感測器接上 5V 電源，序列埠輸出(Tx)接在 Arduino 的第 2 腳，歸零(RX/RST)接在第 3 腳。



(3.1) Continuous Mode (連續模式):

點開 Continuous_Mode 範例，將程式燒進 Arduino 裡面，執行結果如下圖所示。
(感測器通過 DC 電流 10A 測試)

```
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
Current (mA) : 9903 mA, 9.903 A
```

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

(3.2) AT Command Mode (指令模式):

點開 AT_Command_Mode 範例，將程式燒進 Arduino 裡面，執行結果如下圖所示。(感測器通過 DC 電流 10A 測試)

```
DC Mode : Finish!!  
Reset : Finish!!  
Current (mA): -11 mA, -0.011 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF  
Current (mA): 9949 mA, 9.949 A, Temperature (oC): 34.9 oC, 94.8 oF
```

<註>欲量測 AC 電流，須更改部分程式(預設值為 DC 電流)
(#if 1→為量測 DC，#if 0→為量測 AC)

DC 直流電流	AC 交流電流
<pre>#if 1 // 1 為量測 DC Serial.print("DC Mode : "); Pass = DWCS.DC(); #else Serial.print("AC Mode : "); Pass = DWCS.AC(); #endif</pre>	<pre>#if 0 // 0 為量測 AC Serial.print("DC Mode : "); Pass = DWCS.DC(); #else Serial.print("AC Mode : "); Pass = DWCS.AC(); #endif</pre>

(3.3) Modbus-RTU Mode (一對一):

點開 ModbusRTU_SingleDeviceCommunication 範例，將程式燒進 Arduino 裡面，執行結果如下圖所示。(感測器通過 DC 電流 10A 測試)

```

Set Address(1): Finish!!
DC Mode : Finish!!
Reset: Finish!!
Current(mA): 0 mA, 0.000 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10105 mA, 10.105 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.106 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.106 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10107 mA, 10.105 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.109 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.108 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.106 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10106 mA, 10.105 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10105 mA, 10.106 A, Temperature(oC): 29.3 oC, 84.7 oF
Current(mA): 10105 mA, 10.105 A, Temperature(oC): 29.3 oC, 84.7 oF
  
```

<註 1>欲量測 AC 電流，須更改部分程式(預設值為 DC 電流)

(#if 1→為量測 DC，#if 0→為量測 AC)

DC 直流電流	AC 交流電流
<pre> #if 1 // 1 為量測 DC Serial.print("DC Mode : "); Pass = DWCS.DC(); // Pass = DWCS.DC(SlaveAddress); #else Serial.print("AC Mode : "); Pass = DWCS.AC(); // Pass = DWCS.AC(SlaveAddress); #endif </pre>	<pre> #if 0 // 0 為量測 AC Serial.print("DC Mode : "); Pass = DWCS.DC(); // Pass = DWCS.DC(SlaveAddress); #else Serial.print("AC Mode : "); Pass = DWCS.AC(); // Pass = DWCS.AC(SlaveAddress); #endif </pre>

<註 2>修改 Modbus-RTU 位址可以更改以下程式

```

#define SlaveAddress 0x01 // Key in SlaveAddress
    
```

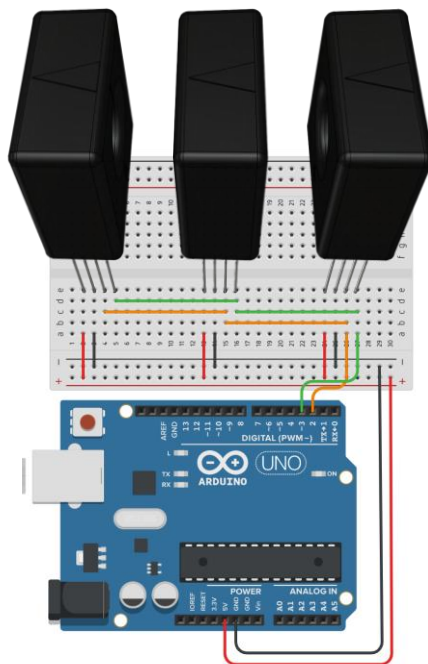
↓ 修改區域，並重新上傳

(3.4) Modbus-RTU Mode (一對多):

首先，分別設定 3 個 DWCS 電流感測器 Modbus-RTU 位址 (0x00, 0x01, 0x02)，設定方式如範例(3.3) [Modbus-RTU Mode \(一對一\)](#)，再點開

ModbusRTU_OneToManyCommunication 範例，將程式燒進 Arduino 裡面。把 3 個 DWCS 電流感測器接上 5V 電源，序列埠輸出(Tx)並接至 Arduino 的第 2 腳，序列埠輸入(Rx)並接至第 3 腳，如下圖所示，接線圖及執行結果。(感測器 1、2 通過 DC 電流 10A 測試，感測器 3 未接)

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



```

DC Mode: Finish!!
Reset: Finish!!
Address: 0x01, Current(mA): 0 mA, -0.007 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): 17 mA, 0.021 A, Temperature(oC): 28.3 oC, 82.0 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====
Address: 0x01, Current(mA): -10118 mA, -10.117 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): -9916 mA, -9.918 A, Temperature(oC): 27.4 oC, 81.7 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====
Address: 0x01, Current(mA): -10121 mA, -10.121 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): -9920 mA, -9.919 A, Temperature(oC): 27.6 oC, 81.3 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====

```

<註>欲量測 AC 電流，須更改部分程式(預設值為 DC 電流)

(#if 1→為量測 DC，#if 0→為量測 AC)

DC 直流電流	AC 交流電流
<pre> #if 1 // 1 為量測 DC Serial.print("DC Mode : "); // Pass = DWCS.DC(); Pass = DWCS.DC(PodCast_Address); #else Serial.print("AC Mode : "); // Pass = DWCS.AC(); Pass = DWCS.AC(PodCast_Address); #endif </pre>	<pre> #if 0 // 0 為量測 AC Serial.print("DC Mode : "); // Pass = DWCS.DC(); Pass = DWCS.DC(PodCast_Address); #else Serial.print("AC Mode : "); // Pass = DWCS.AC(); Pass = DWCS.AC(PodCast_Address); #endif </pre>

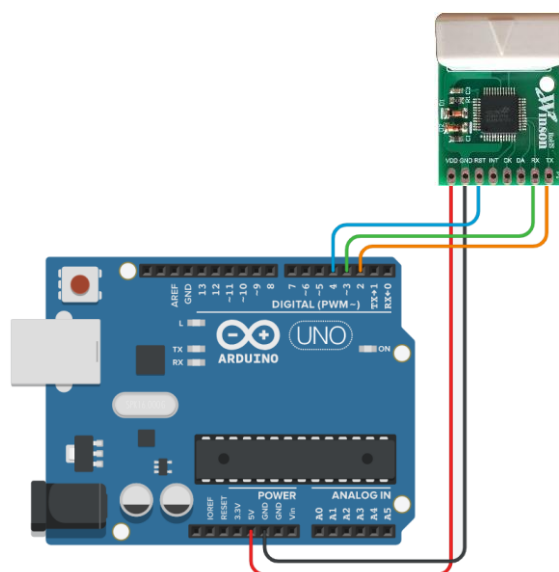
使用廣播位址 PodCast_Address 可以更改全部感測器欲量測 AC 或 DC 電流

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

4. WCM:

(4.1) Continuous Mode (連續模式):

點開 Continuous_Mode 範例，將程式燒進 Arduino 裡面。把 WCM 電流感測模組接上 5V 電源，序列埠輸出(Tx)接在 Arduino 的第 2 腳，序列埠輸入(Rx)接在第 3 腳，歸零(RST)接在第 4 腳，如下圖所示，接線圖及執行結果。(模組通過 DC 電流 5A 測試)

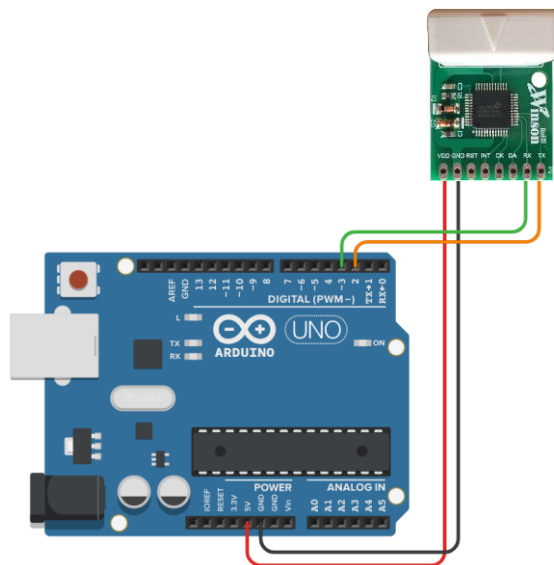


```
Current (mA) : 10 mA, 0.010 A
Current (mA) : 10 mA, 0.010 A
Current (mA) : 10 mA, 0.010 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.960 A
Current (mA) : 4960 mA, 4.960 A
Current (mA) : 4950 mA, 4.960 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4960 mA, 4.950 A
Current (mA) : 4950 mA, 4.960 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4960 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
Current (mA) : 4950 mA, 4.950 A
```

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

(4.2) Modbus-RTU Mode (一對一):

點開 ModbusRTU_SingleDeviceCommunication 範例，將程式燒進 Arduino 裡面。把 WCM 電流感測模組接上 5V 電源，序列埠輸出(Tx)接在 Arduino 的第 2 腳，序列埠輸入(Rx)接在第 3 腳，如下圖所示，接線圖及執行結果。(模組通過 DC 電流 5A 測試)



```
Finish!!
Reset: Finish!!
Current (mA): 12 mA, 0.012 A, Temperature (oC): 28.6 oC, 83.5 oF
Current (mA): 12 mA, 0.011 A, Temperature (oC): 28.4 oC, 83.5 oF
Current (mA): 5017 mA, 5.019 A, Temperature (oC): 28.6 oC, 83.8 oF
Current (mA): 5017 mA, 5.018 A, Temperature (oC): 28.6 oC, 83.1 oF
Current (mA): 5019 mA, 5.018 A, Temperature (oC): 28.8 oC, 83.1 oF
Current (mA): 5018 mA, 5.018 A, Temperature (oC): 28.6 oC, 83.5 oF
Current (mA): 5019 mA, 5.018 A, Temperature (oC): 28.4 oC, 83.1 oF
Current (mA): 5020 mA, 5.020 A, Temperature (oC): 28.4 oC, 83.1 oF
Current (mA): 5022 mA, 5.020 A, Temperature (oC): 28.8 oC, 83.5 oF
Current (mA): 5022 mA, 5.021 A, Temperature (oC): 28.8 oC, 83.5 oF
Current (mA): 5020 mA, 5.021 A, Temperature (oC): 28.4 oC, 83.5 oF
Current (mA): 5018 mA, 5.017 A, Temperature (oC): 28.2 oC, 83.1 oF
Current (mA): 5018 mA, 5.023 A, Temperature (oC): 28.4 oC, 84.2 oF
Current (mA): 5023 mA, 5.020 A, Temperature (oC): 28.4 oC, 83.1 oF
Current (mA): 5019 mA, 5.017 A, Temperature (oC): 28.6 oC, 83.8 oF
Current (mA): 5016 mA, 5.016 A, Temperature (oC): 28.4 oC, 83.1 oF
```

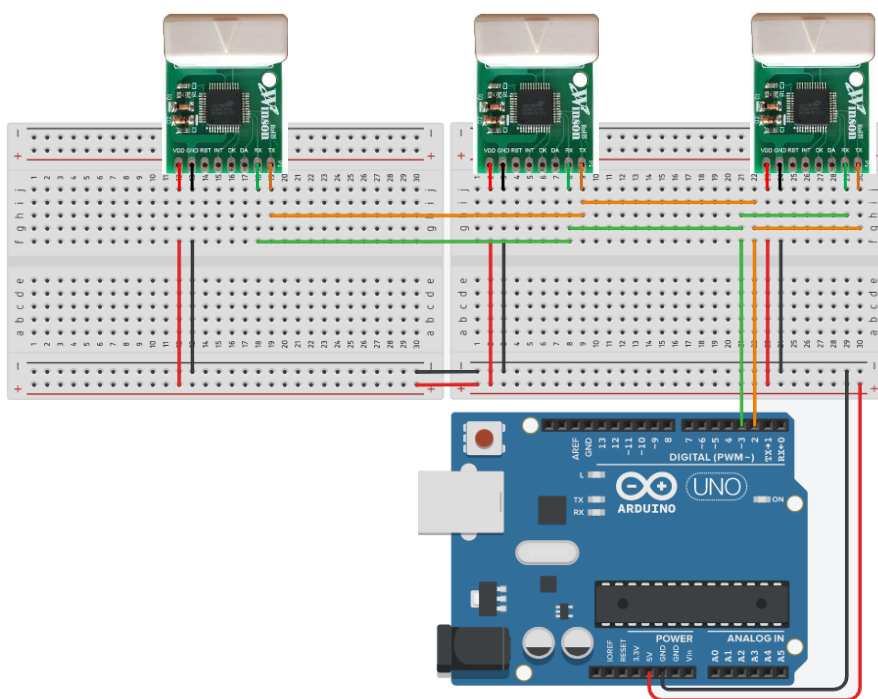
<註>本範例可以修改 Modbus-RTU 位址

```
#define SlaveAddress 0x01 // Key in SlaveAddress
    ↳ 修改區域，並重新上傳
```

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

(4.3) Modbus-RTU Mode (一對多):

首先，分別設定 3 個 WCM 電流感測模組 Modbus-RTU 位址 (0x00, 0x01, 0x02)，設定方式如範例[\(4.2\) Modbus-RTU Mode \(一對一\)](#)，再點開 ModbusRTU_OneToManyCommunication 範例，將程式燒進 Arduino 裡面。把 3 個 WCM 電流感測模組接上 5V 電源，序列埠輸出(Tx)並接至 Arduino 的第 2 腳，序列埠輸入(Rx)並接至第 3 腳，如下圖所示，接線圖及執行結果。(模組 1、2 通過 DC 電 5A 測試，模組 3 未接)



```

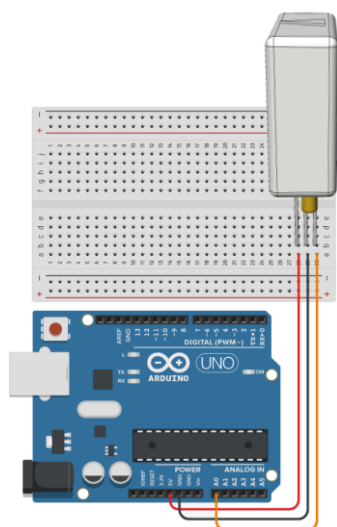
Reset: Finish!!
Address: 0x01, Current(mA): 0 mA, 0.000 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): 8 mA, 0.006 A, Temperature(oC): 29.1 oC, 84.0 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====
Address: 0x01, Current(mA): 5069 mA, 5.069 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): 5001 mA, 5.008 A, Temperature(oC): 28.9 oC, 84.0 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====
Address: 0x01, Current(mA): 5070 mA, 5.070 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA): 4997 mA, 4.996 A, Temperature(oC): 28.9 oC, 84.4 oF
Address: 0x03, Current(mA): 0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
=====
Address: 0x01, Current(mA): 5070 mA, 5.072 A, Temperature(oC): 29.3 oC, 84.7 oF
    
```

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

5. WCS:

(5.1) Single Output: DC (單輸出 DC 電流):

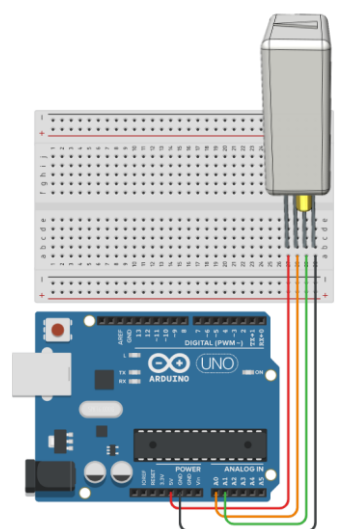
點開 SingleOutput_DC_Current 範例，將程式燒進 Arduino 裡面。把 WCS 電流感測元件接上 5V 電源，輸出(Vout)接在 Arduino 的 A0 腳，如下圖所示，接線圖及執行結果。



```
Reset
Current (A) : 0.000 A
Current (A) : 0.148 A
Current (A) : 0.000 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.588 A
Current (A) : 16.588 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
Current (A) : 16.440 A
```

(5.2) Differential Output: DC(差動輸出 DC 電流):

點開 DifferentialOutput_DC_Current 範例，將程式燒進 Arduino 裡面。把 WCS 電流感測元件接上 5V 電源，輸出 1(Vout1)接在 Arduino 的 A0 腳，輸出 2(Vout2)接在 Arduino 的 A1 腳，如下圖所示，接線圖及執行結果。

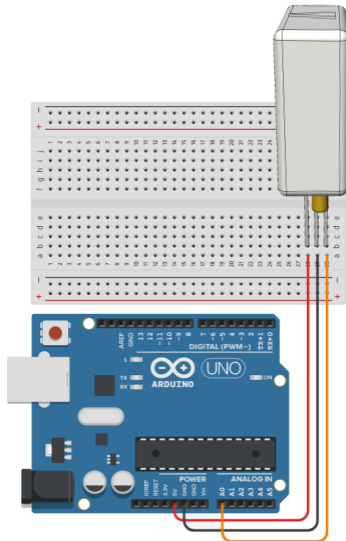


```
Reset
Current (A) : 0.000 A
Current (A) : 0.070 A
Current (A) : 0.035 A
Current (A) : 0.070 A
Current (A) : 0.070 A
Current (A) : 15.186 A
Current (A) : 15.256 A
Current (A) : 15.221 A
Current (A) : 15.117 A
Current (A) : 15.221 A
Current (A) : 15.221 A
Current (A) : 15.152 A
Current (A) : 15.186 A
```

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

(5.3) Single Output: AC(單輸出 AC 電流):

點開 SingleOutput_AC_Current 範例，將程式燒進 Arduino 裡面。把 WCS 電流感測元件接上 5V 電源，輸出(Vout)接在 Arduino 的 A0 腳，如下圖所示，接線圖及執行結果。

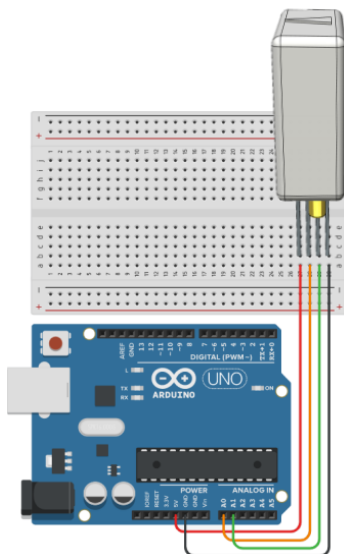


```

Reset
Current (A) : 0.137 A
Current (A) : 0.129 A
Current (A) : 0.125 A
Current (A) : 1.112 A
Current (A) : 1.074 A
Current (A) : 1.109 A
Current (A) : 1.109 A
Current (A) : 1.103 A
Current (A) : 1.107 A
Current (A) : 1.101 A
Current (A) : 1.102 A
Current (A) : 1.113 A
Current (A) : 1.094 A
    
```

(5.4) Differential Output: AC(差動輸出 AC 電流):

點開 DifferentialOutput_AC_Current 範例，將程式燒進 Arduino 裡面。把 WCS 電流感測元件接上 5V 電源，輸出 1(Vout1)接在 Arduino 的 A0 腳，輸出 2(Vout2)接在 Arduino 的 A1 腳，如下圖所示，接線圖及執行結果。



```

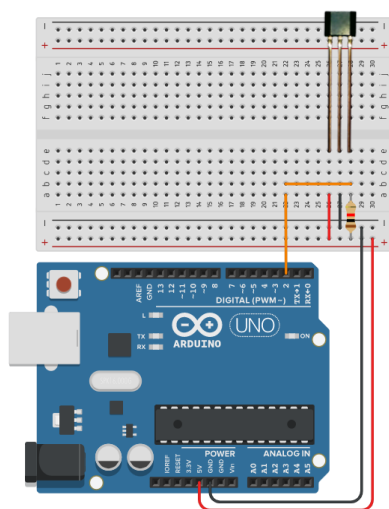
Reset
Current (A) : 0.049 A
Current (A) : 0.043 A
Current (A) : 0.186 A
Current (A) : 1.066 A
Current (A) : 1.061 A
Current (A) : 1.061 A
Current (A) : 1.057 A
Current (A) : 1.066 A
Current (A) : 1.060 A
Current (A) : 1.065 A
Current (A) : 1.051 A
Current (A) : 1.069 A
Current (A) : 1.062 A
    
```

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

6. Hall IC:

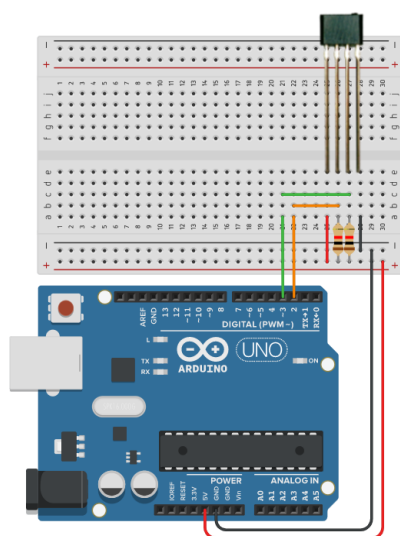
(6.1) Single Output: Switching (單輸出開關):

點開 SingleOutput_Switch_IC_Polling 範例，將程式燒進 Arduino 裡面。把 Hall switch IC 接上 5V 電源，輸出(Vout)接上外部上拉電阻(1KΩ)連接至 Arduino 的第 2 腳，如下圖所示，接線圖。



(6.2) Dual Output: Switching (雙輸出開關):

點開 DualOutput_Switch_IC_Polling 範例，將程式燒進 Arduino 裡面。把 Hall switch IC 接上 5V 電源，分別把 Hall switch IC 輸出腳接上外部上拉電阻(1KΩ)，輸出 1(Vout1)接在 Arduino 的第 2 腳，輸出 2(Vout2)接在 Arduino 的第 3 腳，如下圖所示，接線圖。



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

7. 函式說明

(7.1) DWCS 類別

1. DWCS(byte Tx, byte Rx, Wtype_t Mode)

說明:

DWCS : 初始化 DWCS 類別 (Continuous Mode, AT Command Mode)

Tx : DWCS TX Pin

Rx : DWCS RX Pin

Mode : **AC**, **DC**, **AT** Mode

語法 :

DWCS DWCS(2, 3, **AC**);

2. DWCS(byte Tx, byte Rx, Wtype_t Mode, byte SlaveAddress)

說明:

DWCS : 初始化 DWCS 類別 (Modbus-RTU Mode)

Tx : DWCS TX Pin

Rx : DWCS RX Pin

Mode : **Modbus** Mode

SlaveAddress : Modbus-RTU 位址

語法 :

DWCS DWCS(2, 3, **Modbus**, 0x01);

3. void Init()

說明:

初始化 DWCS

語法 :

DWCS.init();

4. double mA()

說明:

量測電流，單位:mA

語法 :

data = DWCS.mA();

5. double mA(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測電流，單位:mA

SlaveAddress : Modbus-RTU 位址

語法 :

```
data = DWCS.mA( 0x01);
```

6. double A()

說明:

量測電流，單位:A

語法 :

```
data = DWCS.A();
```

7. double A(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測電流，單位:A

SlaveAddress : Modbus-RTU 位址

語法 :

```
data = DWCS.A( 0x01);
```

8. double oC()

說明:

量測溫度，單位:°C

語法 :

```
data = DWCS.oC();
```

9. double oC(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測溫度，單位:°C

SlaveAddress : Modbus-RTU 位址

語法 :

```
data = DWCS.oC( 0x01);
```

10. double oF()

說明:

量測溫度，單位:°F

語法：

```
data = DWCS.oF();
```

11. double oF(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測溫度，單位:°F

SlaveAddress : Modbus-RTU 位址

語法：

```
data = DWCS.oF( 0x01);
```

12. bool Reset()

說明:

歸零電流

語法：

```
DWCS. Reset();
```

13. bool Reset (byte SlaveAddress)

說明:

根據 Modbus-RTU 位址歸零電流

SlaveAddress : Modbus-RTU 位址

語法：

```
DWCS.Reset( 0x01);
```

14. bool DC()

說明:

切換 DC 電流

語法：

```
DWCS. DC();
```

15.bool DC (byte SlaveAddress)

說明:

根據 Modbus-RTU 位址切換 DC 電流

SlaveAddress : Modbus-RTU 位址

語法 :

DWCS.DC(0x01);

16.bool AC()

說明:

切換 AC 電流

語法 :

DWCS.AC();

17.bool AC (byte SlaveAddress)

說明:

根據 Modbus-RTU 位址切換 AC 電流

SlaveAddress : Modbus-RTU 位址

語法 :

DWCS.AC(0x01);

18.bool SetAddress(byte SlaveAddress)

說明:

根據 Modbus-RTU 當前位址更換新位址

SlaveAddress : Modbus-RTU 位址

語法 :

DWCS.SetAddress(0x02);

19.bool SetAddress(byte OldAddress, byte NewAddress)

說明:

指定 Modbus-RTU 位址更換新位址

SlaveAddress : 舊 Modbus-RTU 位址

NewAddress : 新 Modbus-RTU 位址

語法 :

DWCS.SetAddress(0x00, 0x02);

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

20. bool FactoryReset()

說明:

Modbus-RTU 回復原廠設定位址 (0x01)

<註>此指令使用廣播位址(0x00)，建議使用單顆感測器進行設定。

語法:

```
DWCS. FactoryReset();
```

21. byte addr()

說明:

讀取 Modbus-RTU 當前位址

語法:

```
address = DWCS. addr();
```

(7.2) WCM 類別

1. WCM(byte Tx, byte Rx, Wtype_t Mode)

說明:

WCM : 初始化 WCM 類別 (Continuous Mode)

Tx : WCM TX Pin

Rx : WCM RX Pin

Mode : **AC**, **DC**, **ACDC** Mode

語法:

```
WCM WCM ( 2, 3, ACDC);
```

2. WCM(byte Tx, byte Rx, byte Rst, Wtype_t Mode, byte SlaveAddress)

說明:

WCM : 初始化 ACDC 類別 (Modbus-RTU Mode)

Tx : WCM TX Pin

Rx : WCM RX Pin

Rst : WCM RST Pin

Mode : **Modbus** Mode

SlaveAddress : Modbus-RTU 位址

語法:

```
WCM WCM ( 2, 3, 4, Modbus, 0x01);
```

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

3. void Init()

說明:

初始化 WCM

語法:

```
WCM.init();
```

4. double mA()

說明:

量測電流，單位:mA

語法:

```
data = WCM.mA();
```

5. double mA(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測電流，單位:mA

SlaveAddress : Modbus-RTU 位址

語法:

```
data = WCM.mA( 0x01);
```

6. double A()

說明:

量測電流，單位:A

語法:

```
data = WCM.A();
```

7. double A(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測電流，單位:A

SlaveAddress : Modbus-RTU 位址

語法:

```
data = WCM.A( 0x01);
```

8. double oC()

說明:

量測溫度，單位:°C

語法：

```
data = WCM.oC();
```

9. double oC(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測溫度，單位:°C

SlaveAddress : Modbus-RTU 位址

語法：

```
data = WCM.oC( 0x01);
```

10.double oF()

說明:

量測溫度，單位:°F

語法：

```
data = WCM.oF();
```

11.double oF(byte SlaveAddress)

說明:

根據 Modbus-RTU 位址量測溫度，單位:°F

SlaveAddress : Modbus-RTU 位址

語法：

```
data = WCM.oF( 0x01);
```

12.bool Reset()

說明:

歸零電流

語法：

```
WCM. Reset();
```

13.bool Reset (byte SlaveAddress)

說明:

根據 Modbus-RTU 位址歸零電流

SlaveAddress : Modbus-RTU 位址

語法:

```
WCM.Reset( 0x01);
```

14.bool SetAddress(byte SlaveAddress)

說明:

根據 Modbus-RTU 當前位址更換新位址

SlaveAddress : Modbus-RTU 位址

語法:

```
WCM.SetAddress( 0x02);
```

15.bool SetAddress(byte OldAddress, byte NewAddress)

說明:

指定 Modbus-RTU 位址更換新位址

SlaveAddress : 舊 Modbus-RTU 位址

NewAddress : 新 Modbus-RTU 位址

語法:

```
WCM.SetAddress( 0x00, 0x02);
```

16.bool FactoryReset()

說明:

Modbus-RTU 回復原廠設定位址 (0x01)

<註>此指令使用廣播位址(0x00)，建議使用單顆感測器進行設定。

語法:

```
WCM. FactoryReset();
```

17.byte addr()

說明:

讀取 Modbus-RTU 當前位址

語法:

```
address = WCM. addr();
```


(7.3) WCS 類別

1. WCS(uint8_t analogPin, uint16_t mVperA)

說明:

WCS : 初始化 WCS 類別 (單輸出)

analogPin : WCS 輸出 Pin

mVperA : WCS 靈敏度(如表格)

名稱	靈敏度(mV/A)	名稱	靈敏度(mV/A)
_WCS1500	11	_WCS3740	32
_WCS1600	22	_WCS2750	32
_WCS1700	33	_WCS2720	65
_WCS1800	66	_WCS2810	135
_WCS6800	65	_WCS2705	260
_WCS2800	70	_WCS2702	1000
_WCS2200	140	_WCS2801	2000
_WCS2210	280	_WCS37A50	3500
_WCS2202	1120	_WCS38A25	7000
_WCS2201	4200	其他，直接輸入	

<注>使用者也可以直接輸入數值

語法 :

WCS WCS (0, **_WCS1800**);

WCS WCS (0, 66);

2. WCS(uint8_t analogPin, uint8_t analogPin2, uint16_t mVperA)

說明:

WCS : 初始化 WCS 類別 (差動輸出)

analogPin : WCS 輸出 Pin 1

analogPin2 : WCS 輸出 Pin 2

mVperA : WCS 靈敏度(如表格)

語法 :

WCS WCS (0, 1, **_WCS2200**);

3. void Reset()

說明:

歸零電流

語法 :

WCS.Reset();

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

4. **double A_AC()**

說明:

量測交流電流

語法:

```
data = WCS.A_AC();
```

5. **double A_DC()**

說明:

量測直流電流

語法:

```
data = WCS.A_DC();
```