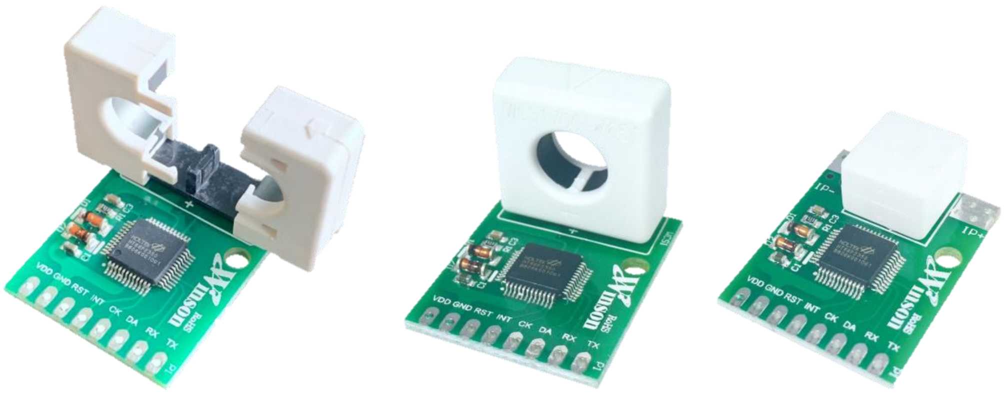
Current Sensing Module Application Note



1. Communication Interface Format

Interface	UART TTL RS232 / RS485 (Need to connect RS232/RS485 signal converter)
Rate	9600 bps
Format	Parity bit: None , Data bit: 8 , Stop bit: 1

2. Operating Mode

(2.1)Continuous Mode: 8 Data Byte, update rate: 3 data/s, reset time: 1s
(2.2)Modbus-RTU Mode: use master-slave request / response communication

3. Measuring Method: (Continuous Mode)

(3.1) DC50C (DC) / 50C (AC/DC): The residual magnetism of the sensor could affect the measurement accuracy. **When first use or switching the measurement direction, it is recommended to provide the test current first, and then reset the sensor when zero current pass.**

(3.2) AC50C (AC) : after power-on, the sensor will automatically reset the current value when zero current pass through the sensor. To measure the effective current, zero current value can also be forced to reset through the reset pin (RST).

(3.3)Zeroing : when there is no current flowing through the current sensor, you can use the RST pin to reset the zero value of current. The proper use of this function will make the measurement more accurate. When measuring DC current, the sensor will generate an amount of remanence. If this remanence cause reading error, please re-zero it.

4. Measuring Method: (Modbus-RTU)

50C (Modbus-RTU) : use "Reset Command" to reset current when no current passes through the sensor, after power-on.

(4.1) Read Holding Registers (Function code:03H)

※The broadcast address (0x00) cannot execute.

(4.1.1) Current

Master request: 01 03 00 02 00 02 65 CB

Slave Address	Function Code	Start Address	No. of Registers	Check Code (CRC)
01H	03H	00H , 02H	00H , 02H	65H, CBH

Slave response: 01 03 04 00 00 04 D2 78 AE

Slave Address	Function Code	Byte Count	Data	Check Code (CRC)
01H	03H	04H	**00H , 00H , 04H , D2H**	78H, AEH

Result: (01) sensor number 1, (00 00 04 D2) **current**=1234/1000 = 1.234A

(4.1.2) Temperature

Master request: 01 03 00 04 00 02 85 CA

Slave Address	Function Code	Start Address	No. of Registers	Check Code (CRC)
01H	03H	00H , 04H	00H , 02H	85H, CAH

Slave response: 01 03 04 00 00 01 2C FA 7E

Slave Address	Function Code	Byte Count	Data	Check Code (CRC)
01H	03H	04H	**00H , 00H , 01H , 2CH**	FAH, 7EH

Result: (01) sensor number 1, (00 00 01 2C) **temperature**=300/10 = 30.0°C

(4.2) Write Holding Registers (Function code:06H)

※The broadcast address (0x00) can execute, but will not respond.

(4.2.1) Reset

Master request: 01 06 00 00 01 00 88 5A

Slave: 01 06 00 00 01 00 88 5A

Slave Address	Function Code	Start Address	Data	Check Code (CRC)
01H	06H	00H , 00H	**01H , 00H**	88H, 5AH

Result: (01) sensor number 1, (01 00) write 256 to **reset**

(4.2.2) Write Address

Master request: 01 06 00 10 00 01 49 CF

Slave response: 01 06 00 10 00 01 49 CF

Slave Address	Function Code	Start Address	Data	Check Code (CRC)
01H	06H	00H , 10H	**00H, 01H**	49H, CFH

Result: (01) sensor number 1, default address 1,(00 01) write **address** 1

(4.3) Exception Code

(4.3.1) Function Code Exception

Master request: 01 01 00 00 00 00 3C 0A

Slave Address	Function Code	Start Address	No. of Registers	Check Code (CRC)
01H	01H	00H , 00H	00H , 00H	3CH, 0AH

Slave response: 01 81 01 81 90

Slave Address	Function Code	Exception Code	Check Code (CRC)
01H	81H	**01H**	81H, 90H

Result: (01) sensor number 1, (81)=0X80(exception) + 0X01(function code), **(01) Exception Code**

(4.3.2) Address Exception

Master request: 01 03 FF FF 00 04 44 2D

Slave response: 01 83 **02** C0 F1

Result: (01) sensor number 1,(83)=0X80(exception) + 0X03(function code), **(02) Exception Code**

(4.3.3) Data Exception

Master request: 01 03 00 00 FF FF 44 7A

Slave response: 01 83 **03** 01 31

Result: (01) sensor number 1, (83)=0X80(exception) + 0X03(function code), **(03) Exception Code**

※Restore Slave Address to Factory State (0x01)

(1) Broadcast **(0x00)**: Set Slave Address to 0x01

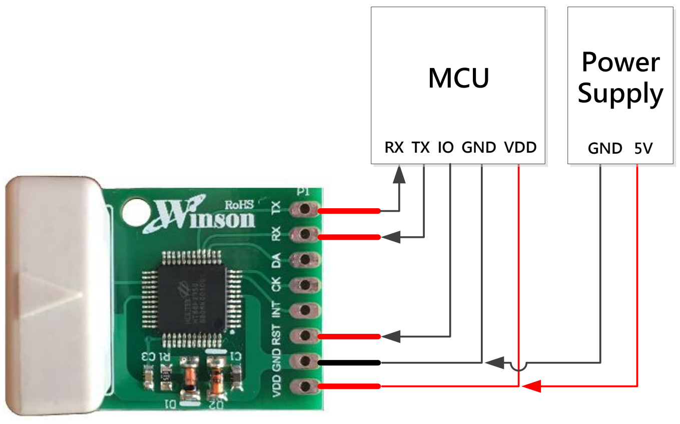
Master request: **00** 06 00 10 00 01 48 1E

Slave response: write only, not respond

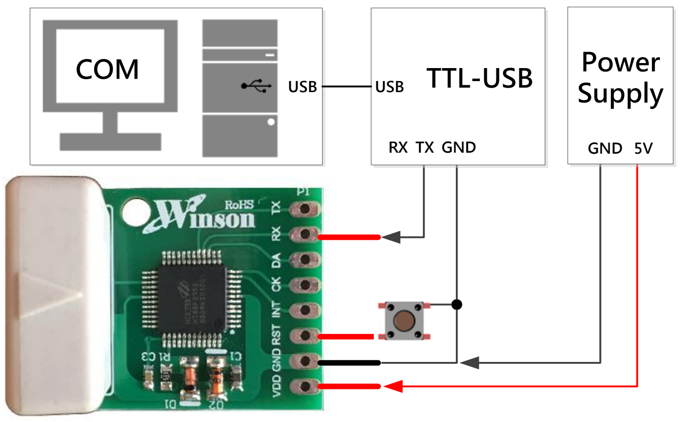
(2) Pin**(INT)** pull-low to reset the slave address (0x01)

5. Application Diagram:

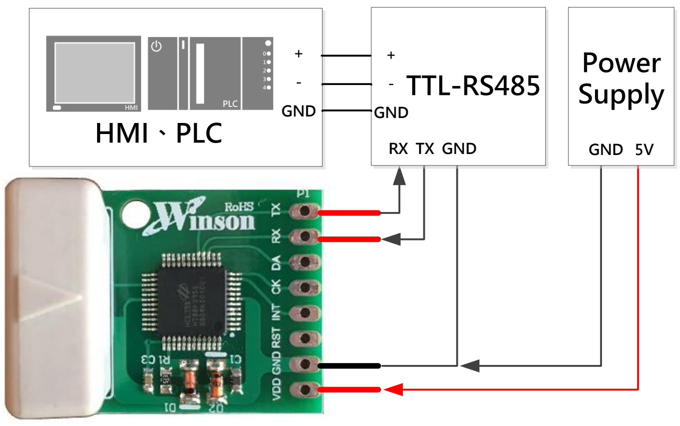
(5.1) MCU Connection Diagram



(5.2) TTL to USB Connection Diagram

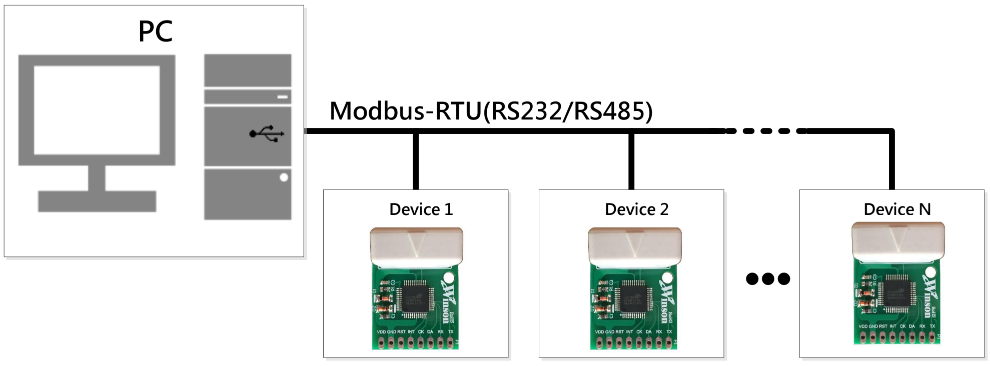


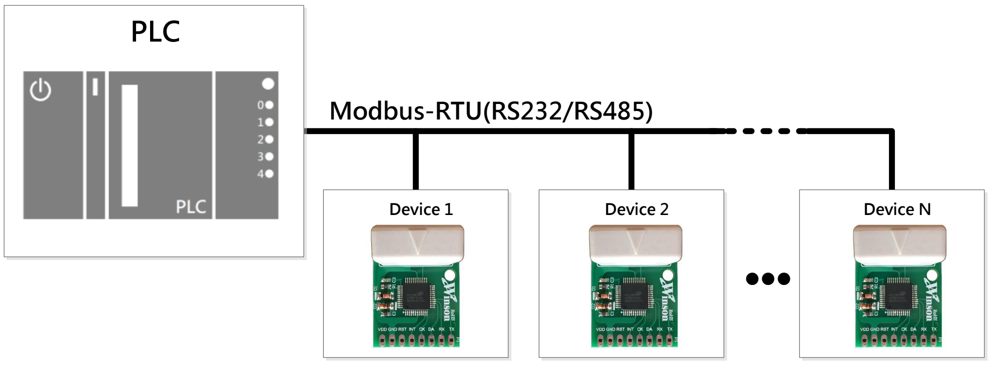
(5.3) TTL to RS485 Connection Diagram

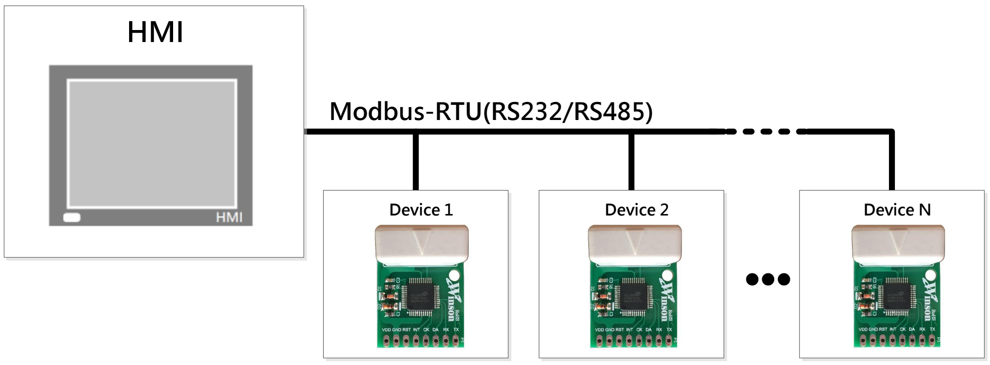


Winson reserves the right to make changes to improve reliability or manufacturability.

(5.4) Modbus-RTU Communication Diagram







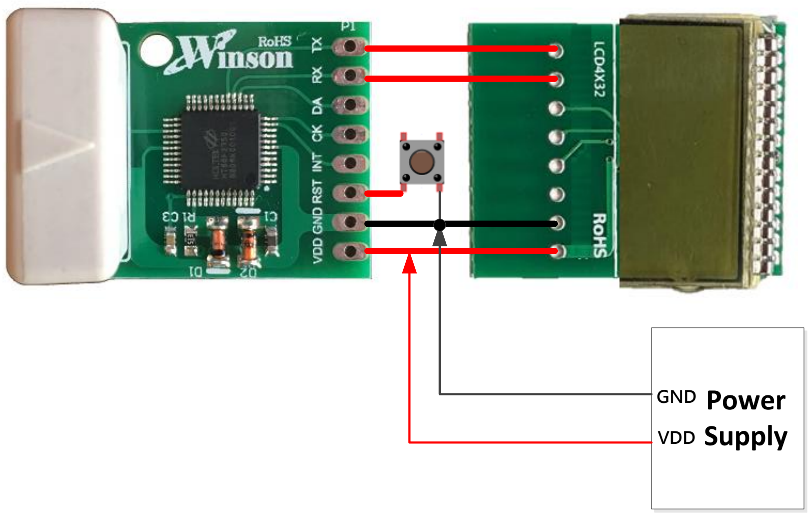
6. WCM-LCD4X32

This is a LCD module applied this current module

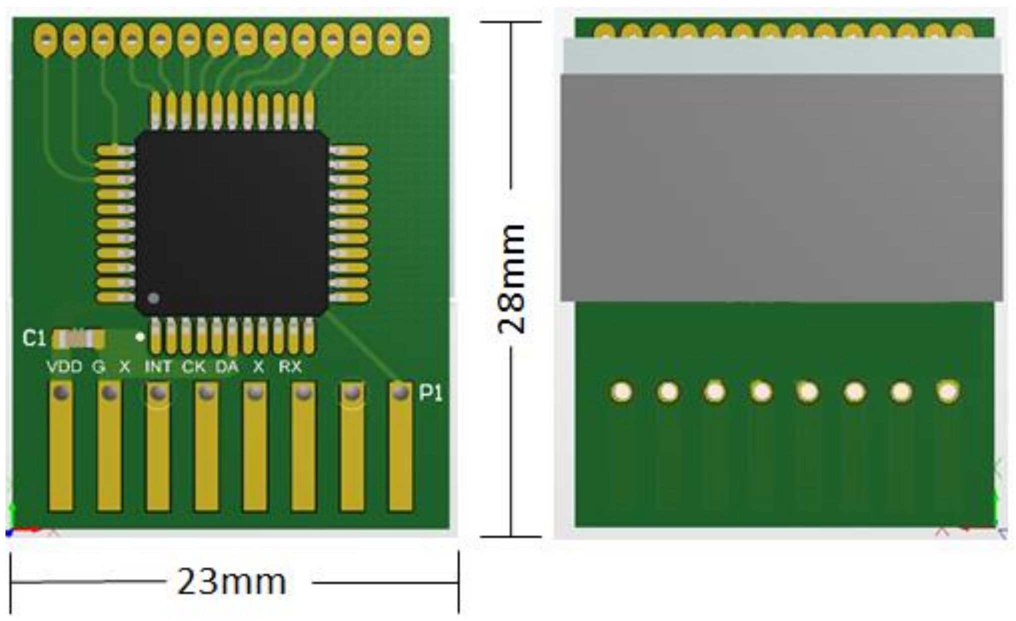
(6.1) Pad Description

Pad No	Pad Name	I/O	Description
1	VDD	-	The positive power input pin
2	GND	-	The system ground
3	x	-	Reserve
4	x	-	Reserve
5	CK	I/O	System programming, reserve
6	DA	I/O	
7	x	-	Reserve
8	RX	O	The data of measured current output. Its output is UART communication. The baud rate is 9.6K bits/sec.

(6.2) LCM-LCD4X32 Application Diagram



(6.3) Package: (Units: mm)

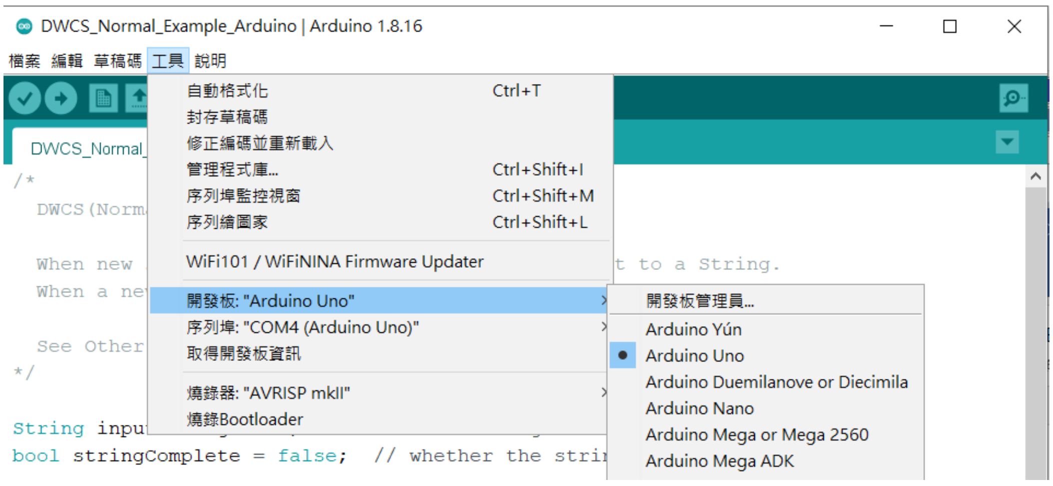


L x W x H = 23mm x 28mm x 8mm

Application Example on Arduino

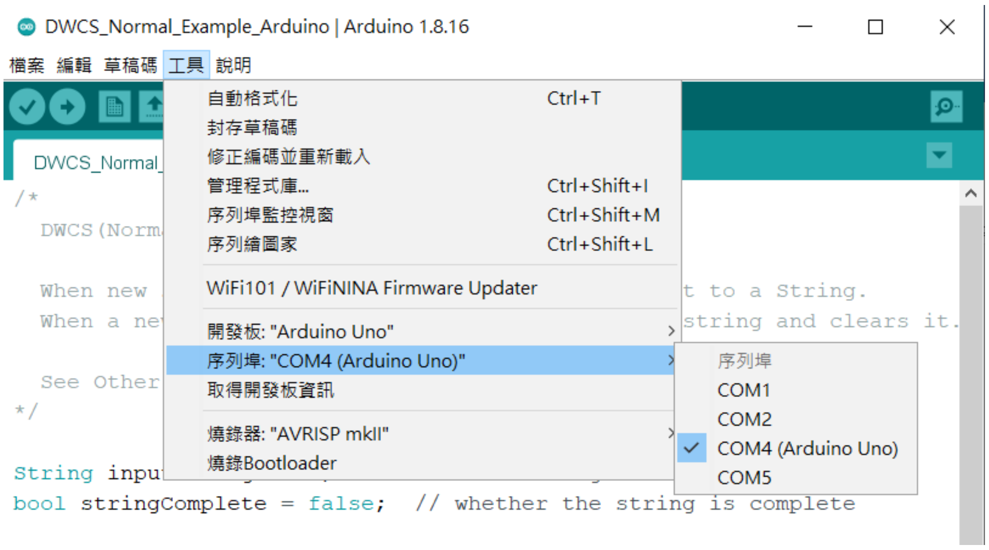
1. Instructions for Arduino

(1.1) Check the type of board is correct.


DWCS_Normal_Example_Arduino | Arduino 1.8.16
檔案 編輯 草稿碼 工具 說明
自動格式化 Ctrl+T
封存草稿碼
修正編碼並重新載入
管理程式庫... Ctrl+Shift+I
序列埠監控視窗 Ctrl+Shift+M
序列繪圖家 Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
開發板: "Arduino Uno"
序列埠: "COM4 (Arduino Uno)"
取得開發板資訊
燒錄器: "AVRISP mkII"
燒錄Bootloader
開發板管理員...
Arduino Yún
● Arduino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino Mega or Mega 2560
Arduino Mega ADK
DWCS_Normal_

```
/*
  DWCS(Norm

  When new                         t to a String.
  When a ne

  See Other
*/

String inpu
bool stringComplete = false;  // whether the stri
```

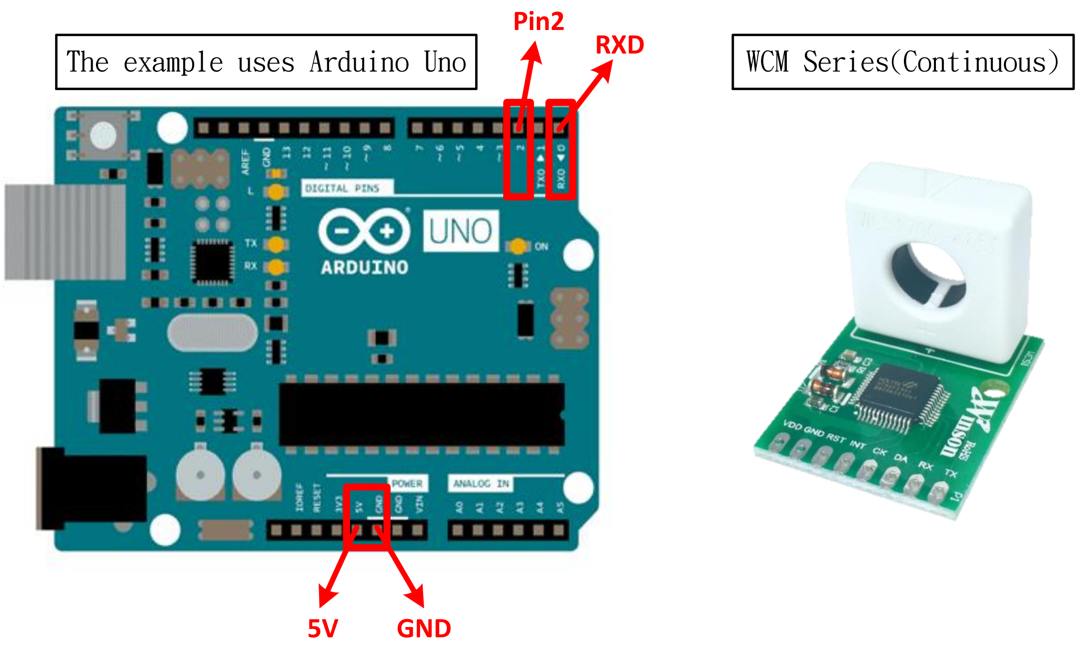
(1.2) Check the port of Arduino is connected and selected correctly.



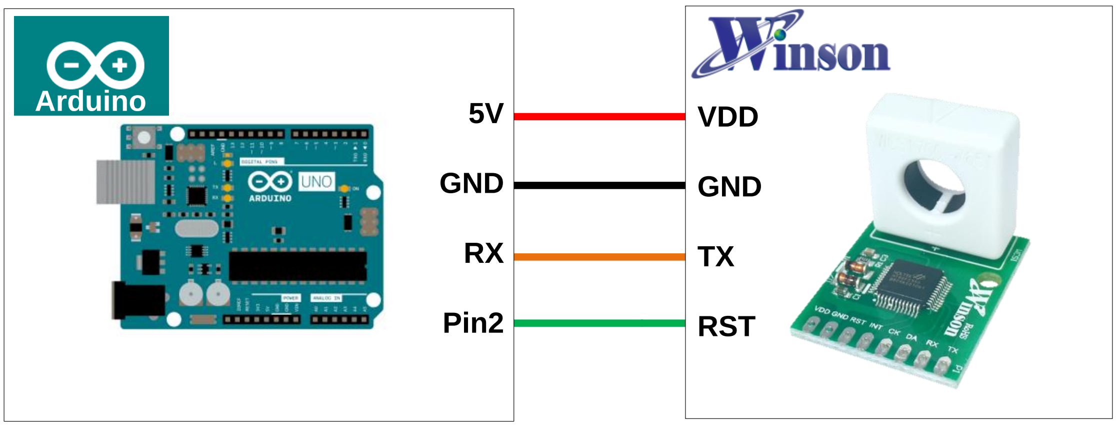
```
/*
  DWCS(Norm

  When new                         t to a String.
  When a ne                        string and clears it.

  See Other
*/

String inpu
bool stringComplete = false;  // whether the string is complete
```

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

2. Continuous Mode

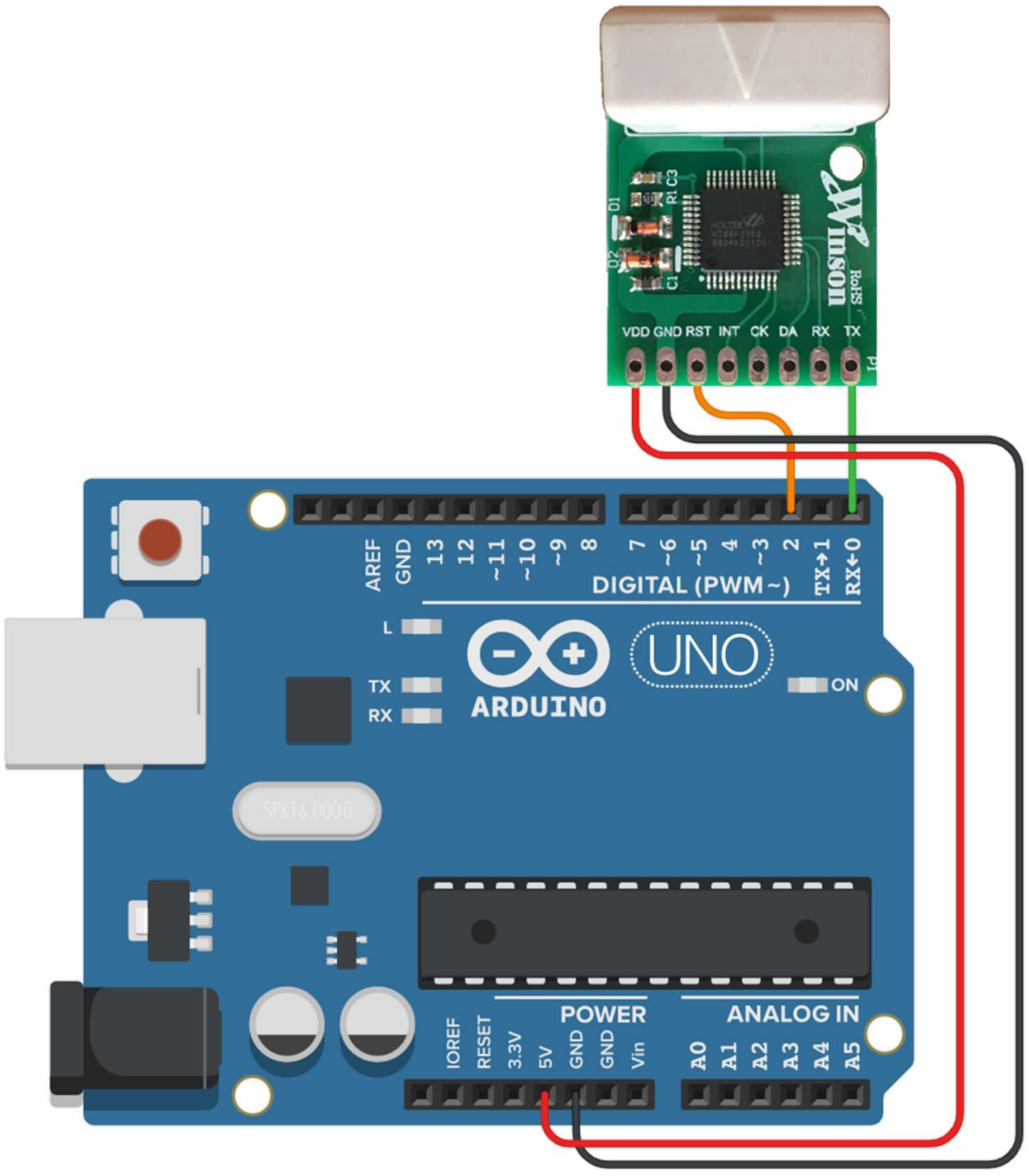


(2.1) Schematic Diagram



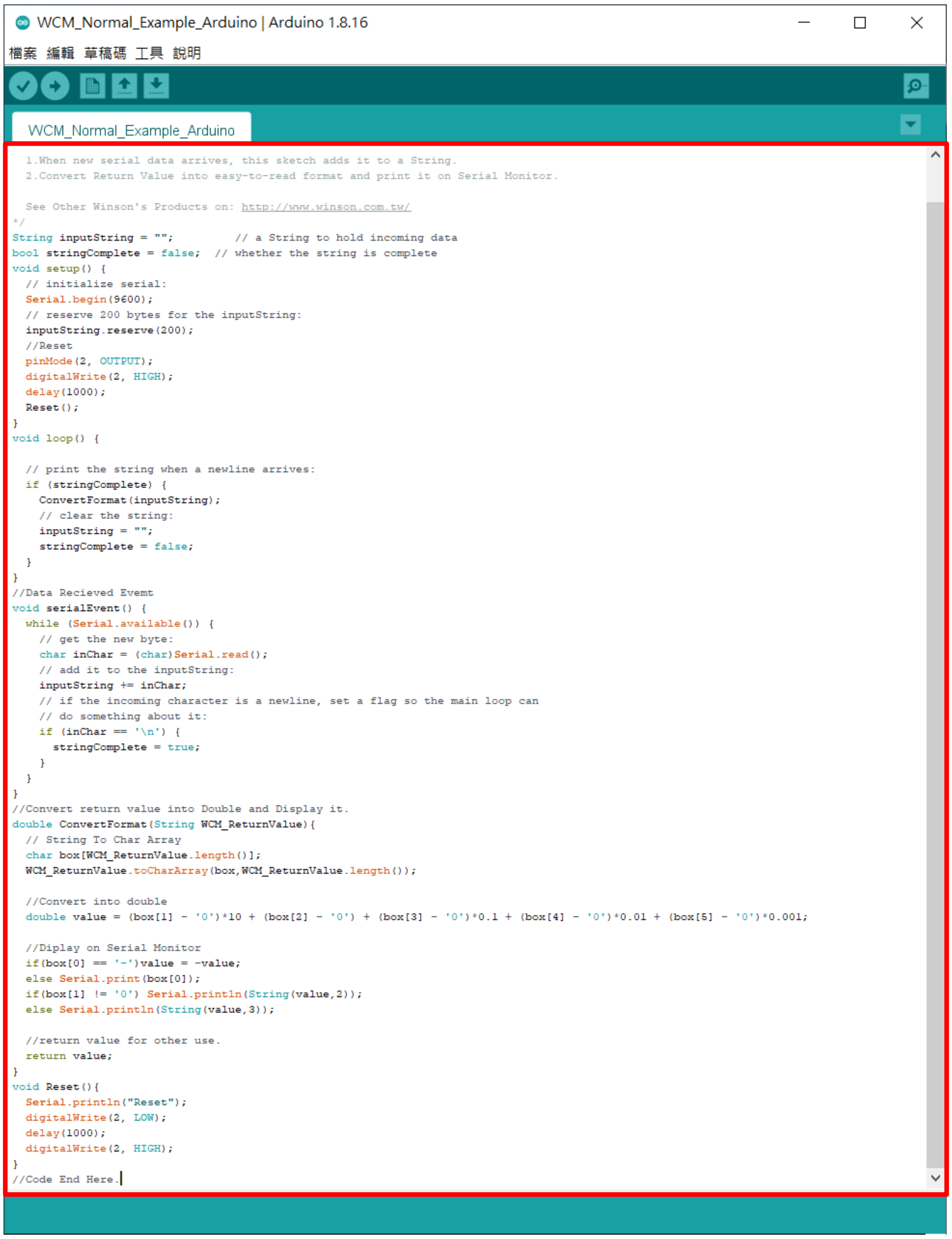
Winson reserves the right to make changes to improve reliability or manufacturability.

(2.2) Wiring Diagram



(2.3) Software & Program

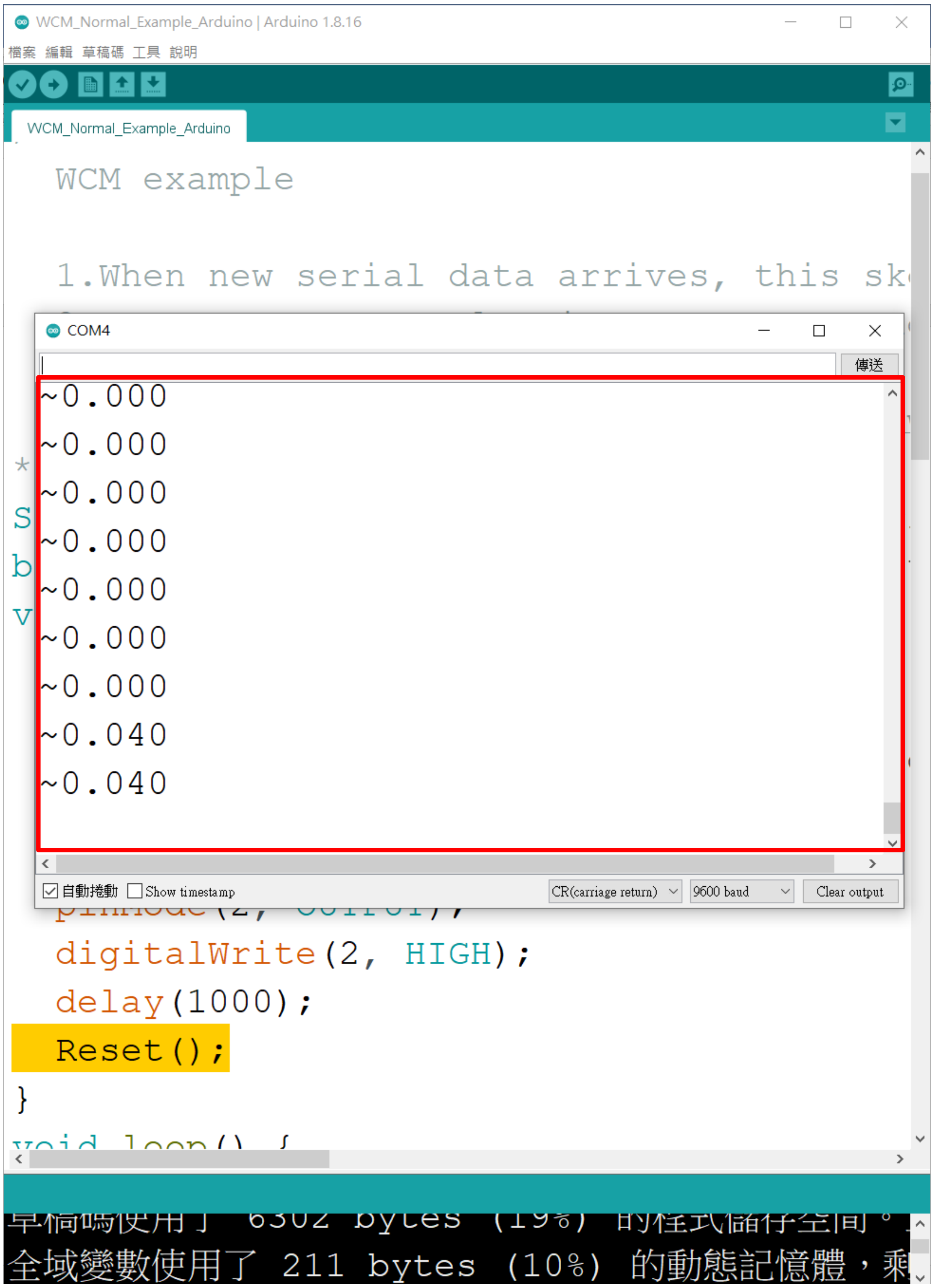
(2.3.1) Code can be download at: http:/www.winson.com.tw/Product/156


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 Recieved Evemt
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;

  //Diplay 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.
```

※CAUTION!! To prevent upload failure of Arduino, please insert WCM after upload process.

(2.3.2) Upload the example code and open Serial Monitor to display the measured current.


WCM_Normal_Example_Arduino | Arduino 1.8.16
檔案 編輯 草稿碼 工具 說明
WCM_Normal_Example_Arduino
COM4
傳送
自動捲動
Show timestamp
CR(carriage return)
9600 baud
Clear output

```
WCM example

1.When new serial data arrives, this sk
```

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

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

```
草稿碼使用了 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
```