Table of contents

1. Arduino library installation steps: 2

2. Arduino program execution steps: 6

3. DWCS: 7

(3.1) Continuous Mode: 7
(3.2) AT Command Mode: 8
(3.3) Modbus-RTU Mode (one on one): 9
(3.4) Modbus-RTU Mode (multipoint): 9

4. WCM: 11

(4.1) Continuous Mode: 11
(4.2) Modbus-RTU Mode (one on one): 12
(4.3) Modbus-RTU Mode (multipoint): 13

5. WCS: 14

(5.1) Single Output: DC : 14
(5.2) Differential Output: DC: 14
(5.3) Single Output: AC: 15
(5.4) Differential Output: AC: 15

6. Hall IC: 16

(6.1) Single Output: Switching: 16
(6.2) Dual Output: Switching: 16

7. Function Description 17

(7.1) DWCS Class 17
(7.2) WCM Class 21
(7.3) WCS Class 25

WinsonLib: Arduino Library Application Note

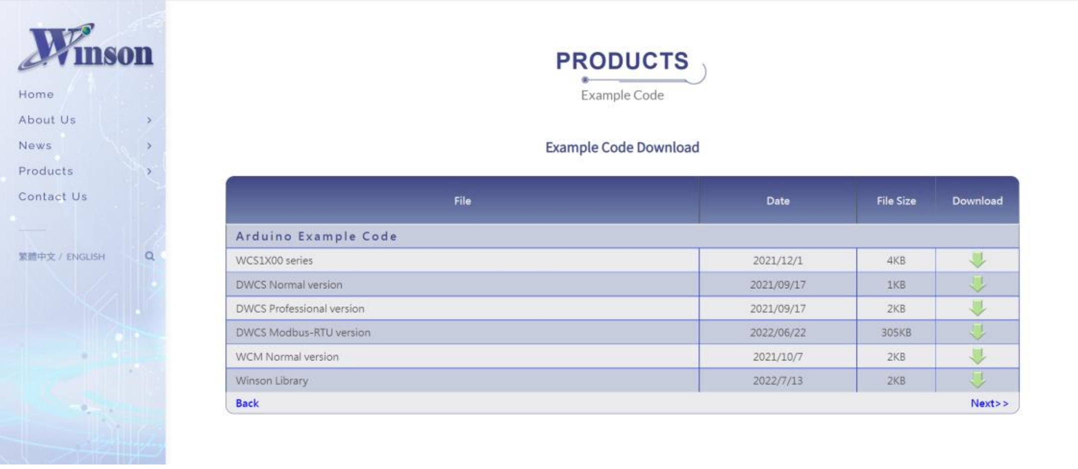
1. Arduino library installation steps:

There are two installation methods: 1. download and install on Winson official website, 2.use Arduino IDE Library Manager.

This installation example used the library on the official website of Winson company. The library contains the functions of most products developed by Winson company, such as WCS, WCM, DWCS and Hall IC, for user development.

Method: 1

Step 1 : download “WinsonLib” library on the website.

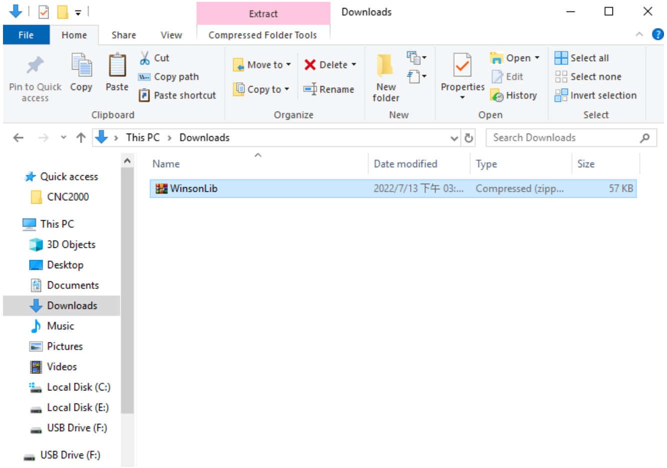


Example Code Download

File	Date	File Size	Download
Arduino Example Code			
WCS1X00 series	2021/12/1	4KB	⬇
DWCS Normal version	2021/09/17	1KB	⬇
DWCS Professional version	2021/09/17	2KB	⬇
DWCS Modbus-RTU version	2022/06/22	305KB	⬇
WCM Normal version	2021/10/7	2KB	⬇
Winson Library	2022/7/13	2KB	⬇

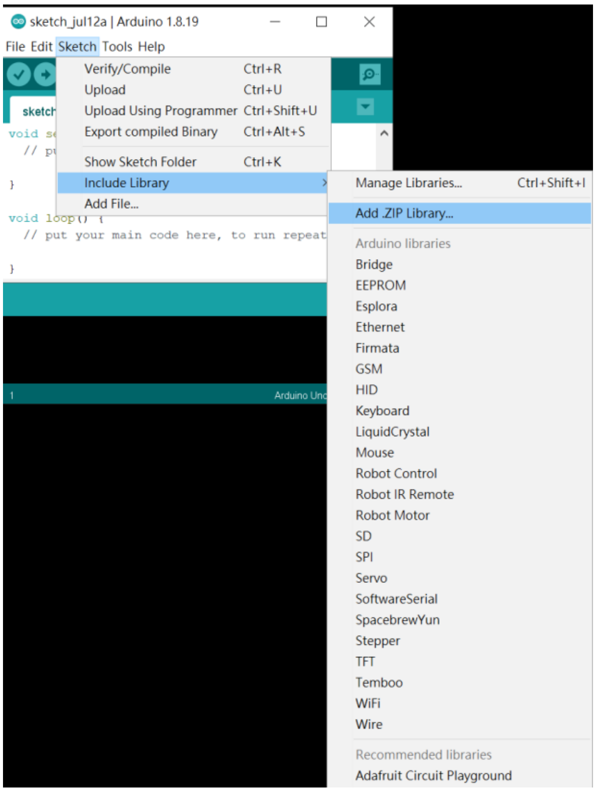
Back Next>>

Step 2: generally, the library file (.zip) will be in the default download folder of the system.

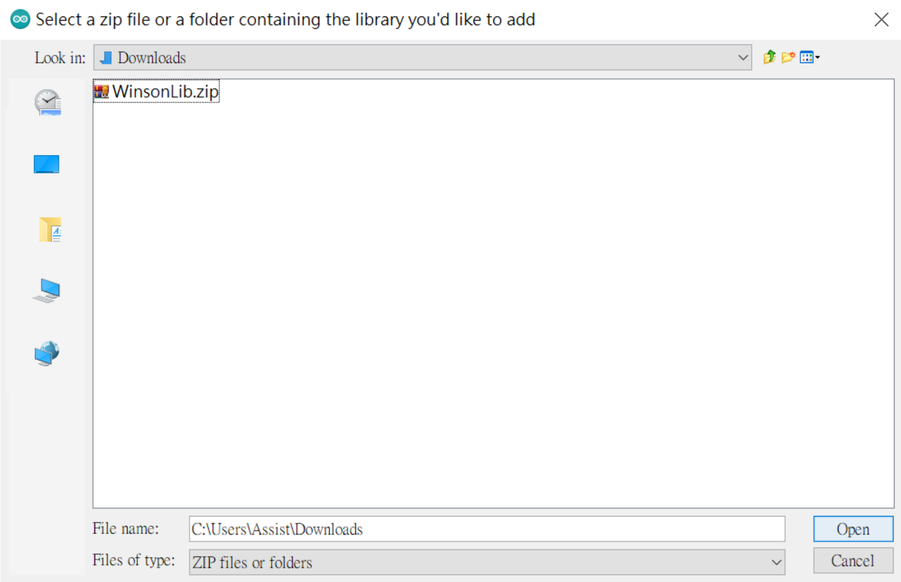


Name	Date modified	Type	Size
WinsonLib	2022/7/13 下午 03:...	Compressed (zipp...	57 KB

Step 3: enter Arduino IDE and choose Sketch→Include Library →Add .ZIP Library...



Step 4: select the library file (.zip) you just downloaded.



Select a zip file or a folder containing the library you'd like to add

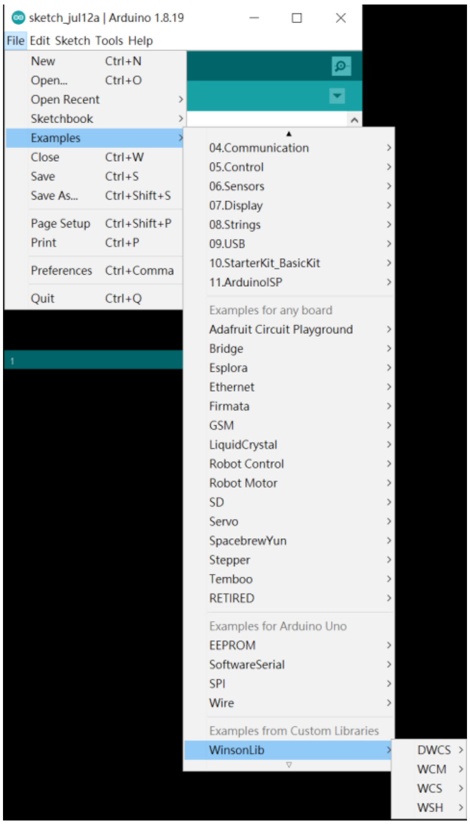
Look in: Downloads

WinsonLib.zip

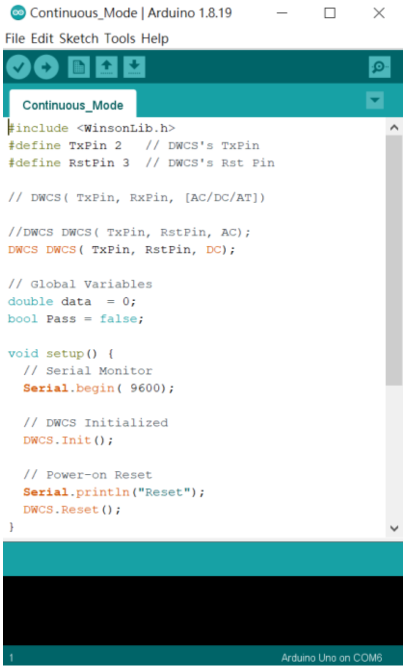
File name: C:\Users\Assist\Downloads

Files of type: ZIP files or folders

Step 5: if installation successfully, the example can be found in the Arduino IDE : File→Examples→"WinsonLib" (Examples from Custom Libraries).



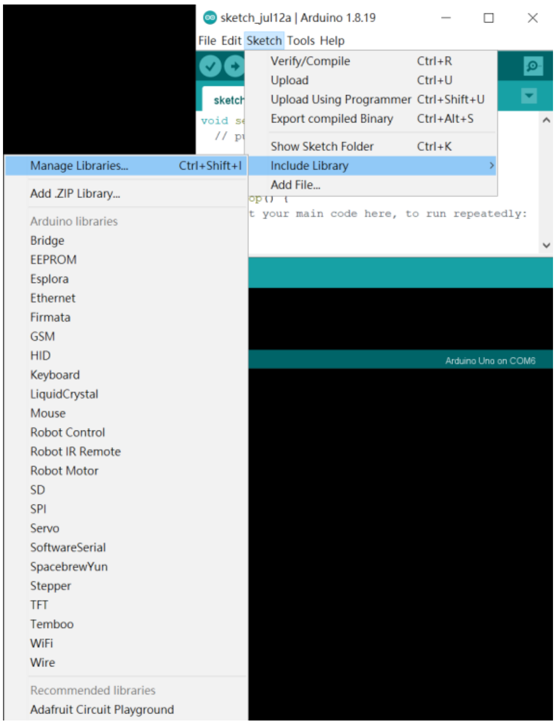
Step 6: if the example program in the Arduino IDE can be used normally, it means that the library is installed correctly.



```
#include <WinsonLib.h>
#define TxPin 2   // DWCS's TxPin
#define RstPin 3  // DWCS's Rst Pin

// DWCS( TxPin, RxPin, [AC/DC/AT])

//DWCS DWCS( TxPin, RstPin, AC);
DWCS DWCS( TxPin, RstPin, DC);

// Global Variables
double data  = 0;
bool Pass = false;

void setup() {
  // Serial Monitor
  Serial.begin( 9600);

  // DWCS Initialized
  DWCS.Init();

  // Power-on Reset
  Serial.println("Reset");
  DWCS.Reset();
}
```

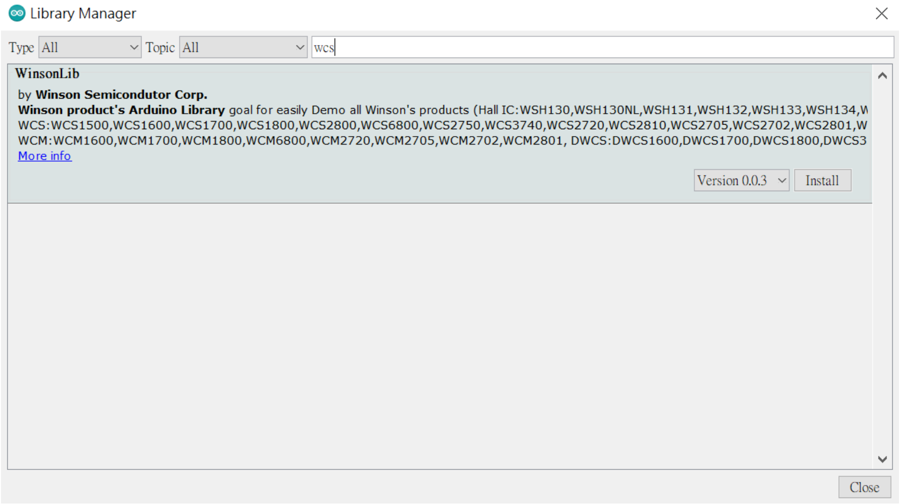
Method: 2

Step 1: open the Arduino Library Manager : Sketch→Include Library→ Manage Libraries...



Step 2: enter the Library Manager and search: Winson, WCS, WCM or DWCS, etc. to find the library "WinsonLib", and then Install.

Step 3: use the method to test on the previous page.



Type All Topic All wcs

WinsonLib

by **Winson Semicondutor Corp.**

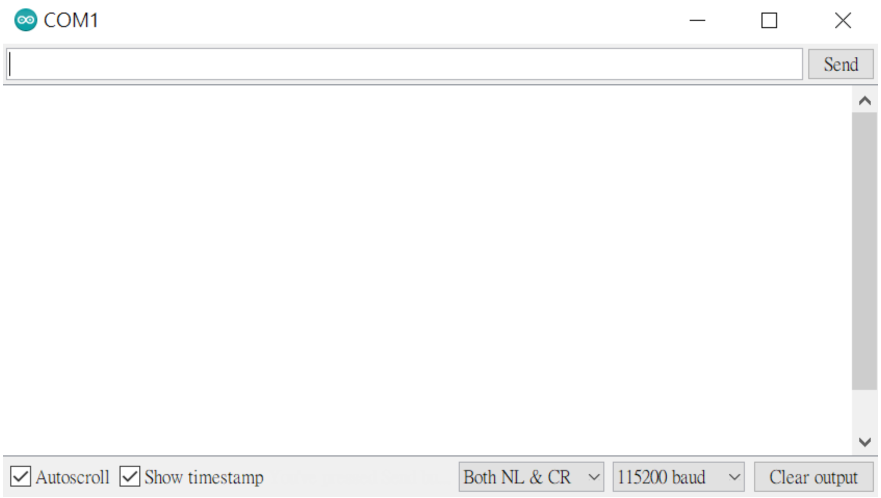
Winson product's Arduino Library goal for easily Demo all Winson's products (Hall IC:WSH130,WSH130NL,WSH131,WSH132,WSH133,WSH134,W
WCS:WCS1500,WCS1600,WCS1700,WCS1800,WCS2800,WCS6800,WCS2750,WCS3740,WCS2720,WCS2810,WCS2705,WCS2702,WCS2801,W
WCM:WCM1600,WCM1700,WCM1800,WCM6800,WCM2720,WCM2705,WCM2702,WCM2801, DWCS:DWCS1600,DWCS1700,DWCS1800,DWCS3

Version 0.0.3

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

2. Arduino program execution steps:

1. Open the Arduino IDE: ♾
2. Open the program: File→Examples→WinsonLib→DWCS→ Continuous_Mode
3. Choose the test board: Tools→Board:→Arduino Uno
4. Choose COM Port: Tools→Port:→COMn
5. Upload: ➜
6. Execution results (monitoring window): Tools→Serial Monitor

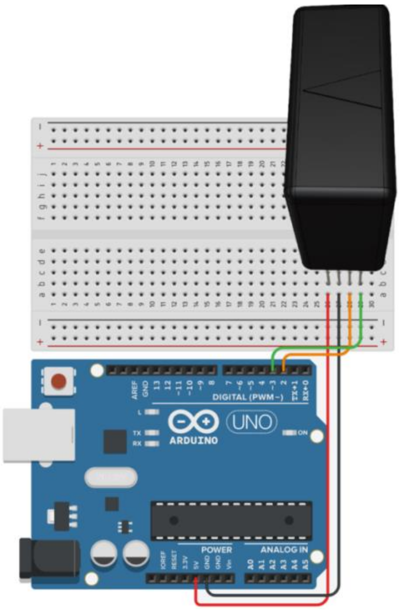


<Note> Arduino has only one default serial port (system serial port). This example does not use the system serial port (pins 0 and 1), but sets other pins as serial ports through the software serial library. Serial Monitor→ Raud Rate : 9600 bps

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

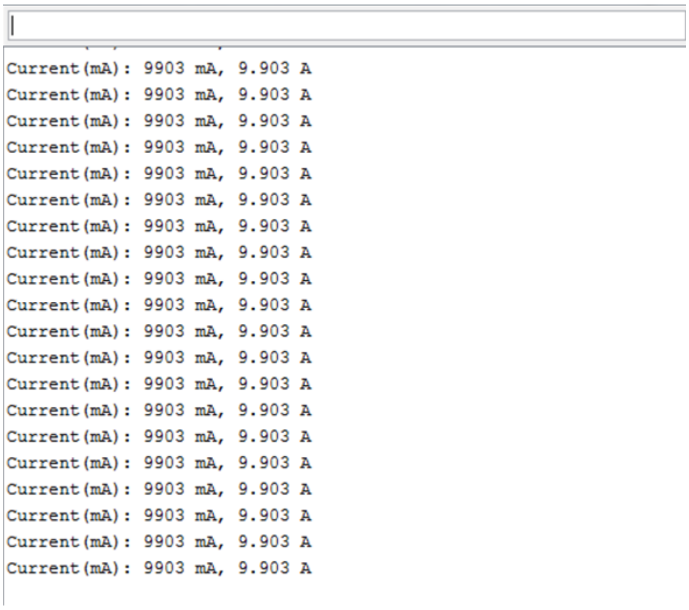
3. DWCS:

The wiring diagram of continuous mode, AT command and Modbus-RTU mode (one on one) is as following. Connect the DWCS current sensor to the 5V power supply. The serial port output pin (TX) of DWCS current sensor connects the pin 2 of Arduino, and the zero pin (RX/RST) to the pin 3.



(3.1) Continuous Mode:

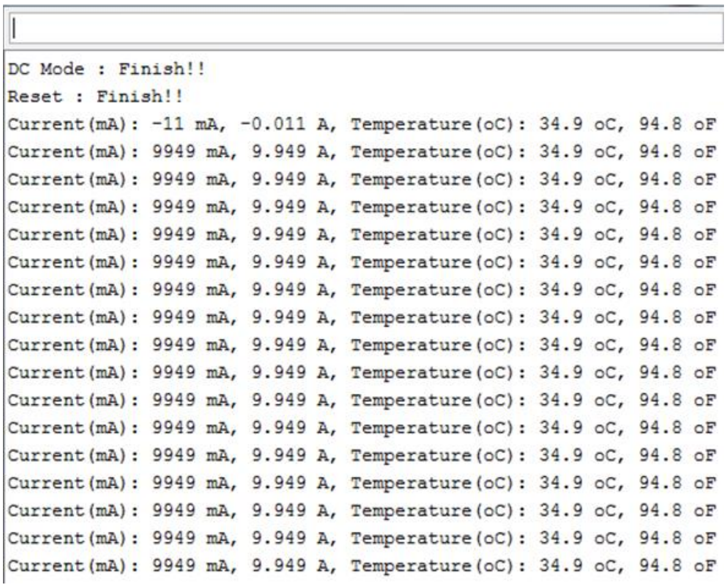
Click on “Continuous_ Mode” example, and upload the program into Arduino. The result is shown in the figure below. (DC current 10A)



```
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
```

(3.2) AT Command Mode:

Click on “AT_Command_Mode” example, and upload the program into Arduino. The result is shown in the figure below. (DC current 10A)



```
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
```

<Note> The code for measuring AC current must be changed. (Default is DC current) (#if 1→DC，#if 0→AC)

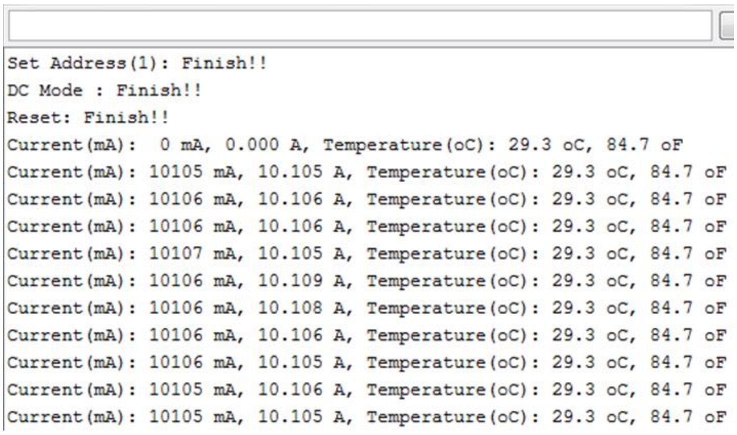
DC	AC

#if 1 // 1: DC
  Serial.print("DC Mode : ");
  Pass = DWCS.DC();
#else
  Serial.print("AC Mode : ");
  Pass = DWCS.AC();
#endif
	#if 0 // 0: AC
	  Serial.print("DC Mode : ");
	  Pass = DWCS.DC();
	#else
	  Serial.print("AC Mode : ");
	  Pass = DWCS.AC();
	#endif

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

(3.3) Modbus-RTU Mode (one on one):

Click on “ModbusRTU_SingleDeviceCommunication” example, and upload the program into Arduino. The result is shown in the figure below. (DC current 10A)



```
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
```

<Note>The code for measuring AC current must be changed. (Default is DC current) (#if 1→DC，#if 0→AC)

DC	AC

#if 1 // 1: 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
	#if 0 // 0: 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

<Note> Modifying the Modbus-RTU address need to change the following code.

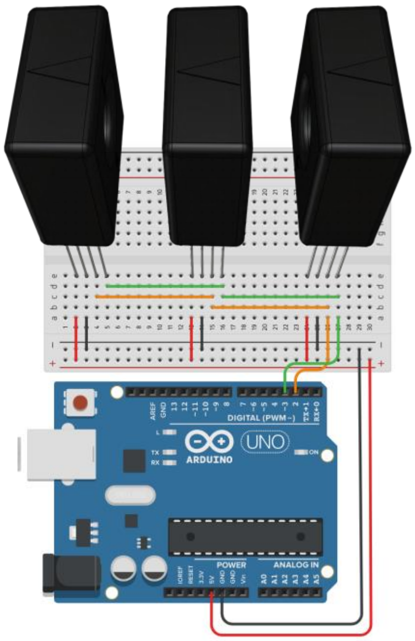
```
#define SlaveAddress 0x01 // Key in SlaveAddress
                    ↳ Modify area, and re-upload
```

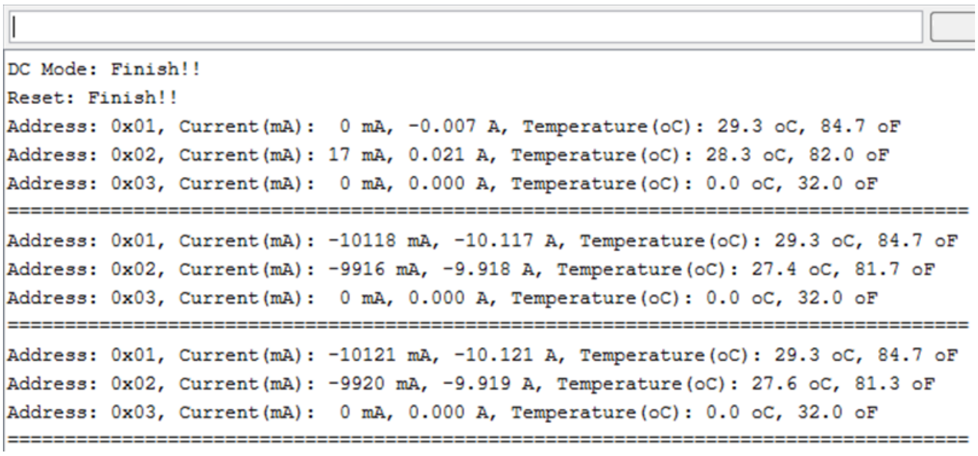
(3.4) Modbus-RTU Mode (multipoint):

First, set the Modbus-RTU Slave address (0x00, 0x01, 0x02) of the three DWCS current sensors respectively, as shown in (3.3) Modbus-RTU Mode (one on one), and then click on “ModbusRTU_OneToManyCommunication” example, and upload the program into Arduino. Connect the three DWCS current sensors to the 5V power supply. The serial port output pin (TX) of the three DWCS current sensors connect the pin 2 of Arduino, and serial port input pin (RX) to the pin 3, as shown in the following figure. (Sensors 1 and 2

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

pass the DC current 10A, and sensor 3 is not connected)





```
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
======================================================================================
```

<Note>The code for measuring AC current must be changed. (Default is DC current) (#if 1→DC，#if 0→AC)

DC	AC

#if 1 // 1: 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
	#if 0 // 0: 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

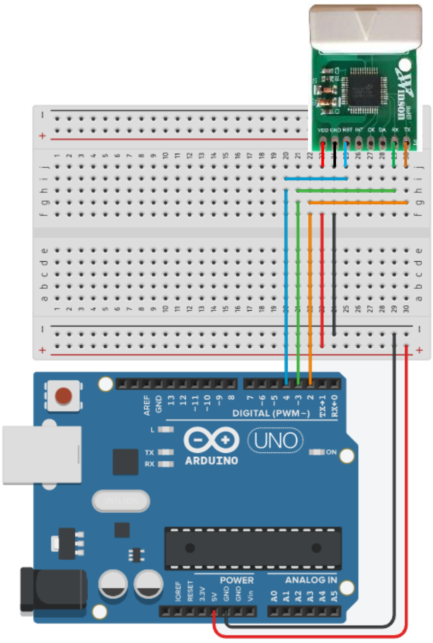
Use the broadcast address (PodCast_Address) to change all sensors to AC or DC current.

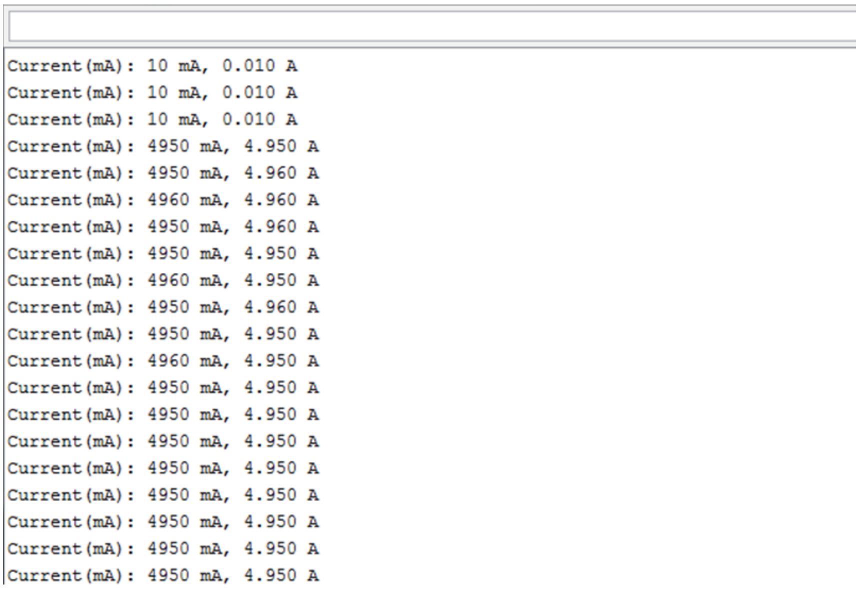
Winson reserves the right to make changes to improve reliability or manufacturability.

4. WCM:

(4.1) Continuous Mode:

Click on “Continuous_ Mode” example, and upload the program into Arduino. Connect the WCM current module to the 5V power supply. The serial port output pin (TX) of the WCM current module connects the pin 2 of Arduino, and the serial port input pin (RX) to the pin 3, and the zero pin (RST) to the pin 4, as shown in the following figure. (DC current 5A)

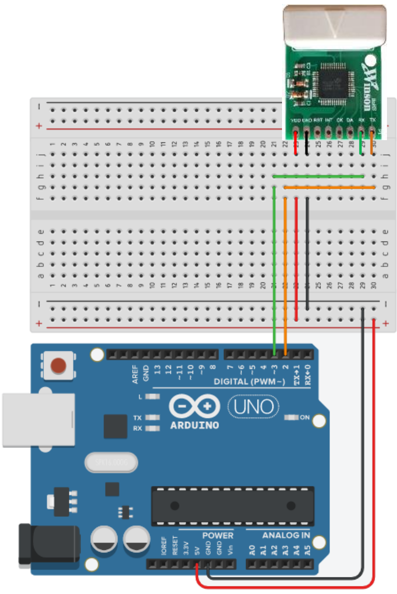


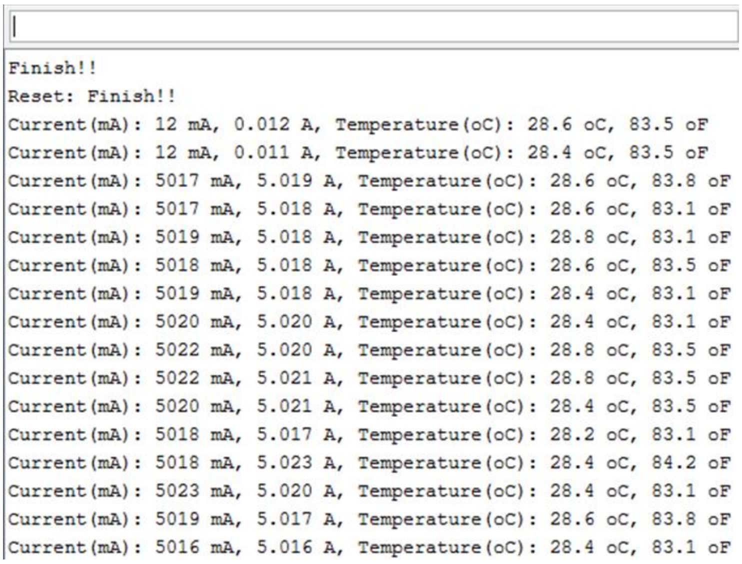


```
Current(mA): 10 mA, 0.010 A
Current(mA): 10 mA, 0.010 A
Current(mA): 10 mA, 0.010 A
Current(mA): 4950 mA, 4.950 A
Current(mA): 4950 mA, 4.960 A
Current(mA): 4960 mA, 4.960 A
Current(mA): 4950 mA, 4.960 A
Current(mA): 4950 mA, 4.950 A
Current(mA): 4960 mA, 4.950 A
Current(mA): 4950 mA, 4.960 A
Current(mA):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

(4.2) Modbus-RTU Mode (one on one):

Click on “ModbusRTU_SingleDeviceCommunication” example, and upload the program into Arduino. Connect the WCM current module to the 5V power supply. The serial port output pin (TX) of the WCM current module connects the pin 2 of Arduino, and serial port input pin (RX) to the pin 3, as shown in the following figure. (DC current 5A)





```
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
```

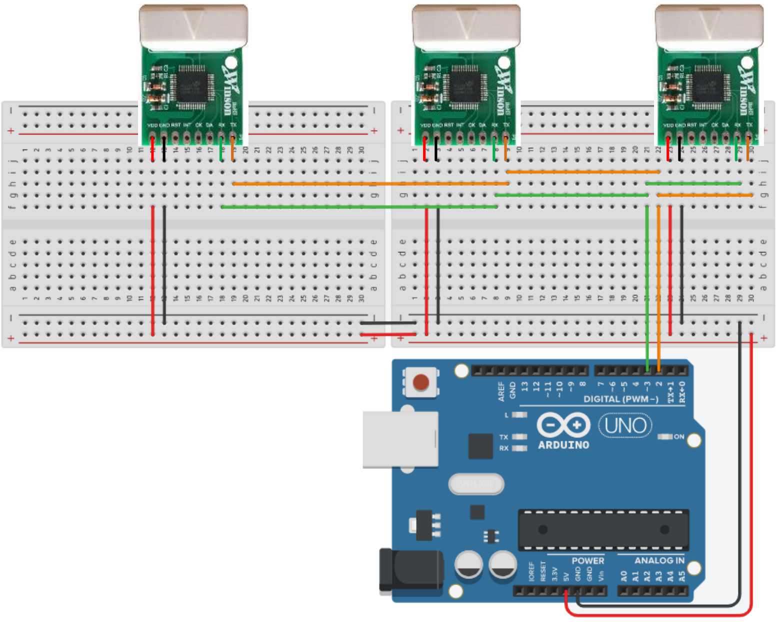
<Note> Modifying the Modbus-RTU address need to change the following code.

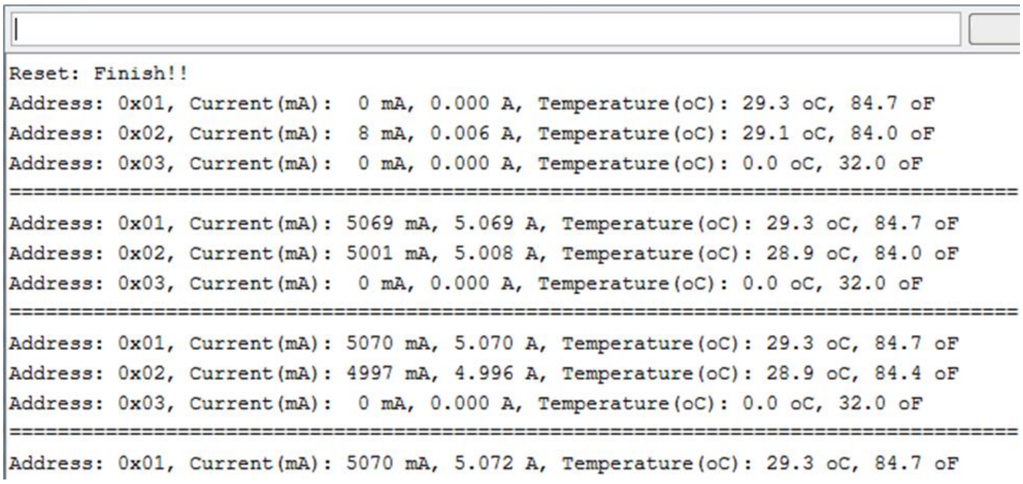
```
#define SlaveAddress 0x01 // Key in SlaveAddress
                         ↳ Modify area, and re-upload
```

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

(4.3) Modbus-RTU Mode (multipoint):

First, set the Modbus-RTU Slave address (0x00, 0x01, 0x02) of the three WCM current modules respectively, as shown in (4.2) Modbus-RTU Mode (one on one), and then click on “ModbusRTU_OneToManyCommunication” example, and upload the program into Arduino. Connect the three WCM current modules to the 5V power supply. The serial port output pin (TX) of the three WCM current modules connect the pin 2 of Arduino, and serial port input pin (RX) to the pin 3, as shown in the following figure. (Modules 1 and 2 pass the DC current 10A, and Module 3 is not connected)



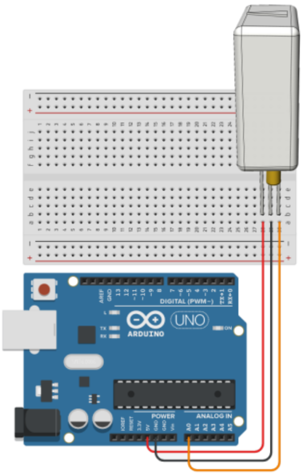

```
Reset: Finish!!
Address: 0x01, Current(mA):  0 mA, 0.000 A, Temperature(oC): 29.3 oC, 84.7 oF
Address: 0x02, Current(mA):  8 mA, 0.006 A, Temperature(oC): 29.1 oC, 84.0 oF
Address: 0x03, Current(mA):  0 mA, 0.000 A, Temperature(oC): 0.0 oC, 32.0 oF
==========================================================================
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
```

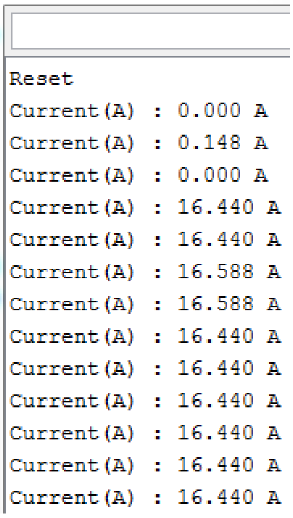
Winson reserves the right to make changes to improve reliability or manufacturability.

5. WCS:

(5.1) Single Output: DC:

Click on “SingleOutput_DC_Current” example, and upload the program into Arduino. Connect the WCS current sensor to the 5V power supply. The output pin (Vout) of the WCS current sensor connects the analog pin A0 of Arduino, as shown in the following figure.

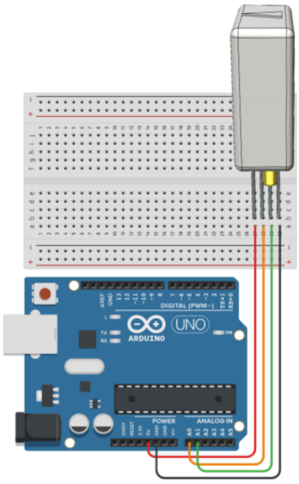


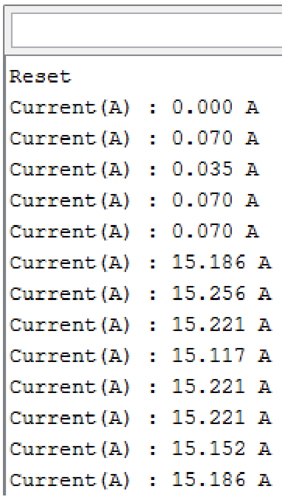


```
Reset
Current(A) : 0.000 A
Current(A) : 0.148 A
Current(A) : 0.000 A
Current(A) : 16.440 A
Current(A) : 16.440 A
Current(A) : 16.588 A
Current(A) : 16.588 A
Current(A) : 16.440 A
Current(A) : 16.440 A
Current(A) : 16.440 A
Current(A) : 16.440 A
Current(A) : 16.440 A
Current(A) : 16.440 A
```

(5.2) Differential Output: DC:

Click on “DifferentialOutput_DC_Current” example, and upload the program into Arduino. Connect the WCS current sensor to the 5V power supply. The output pin 1(Vout1) of the WCS current sensor connects the analog pin A0 of Arduino, and the output pin 2(Vout2) to A1, as shown in the following figure.

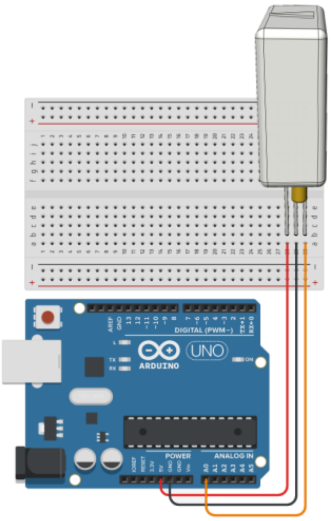


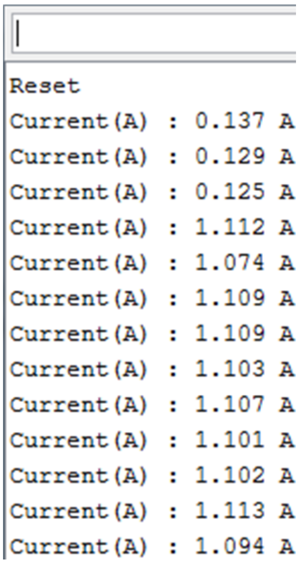


```
Reset
Current(A) : 0.000 A
Current(A) : 0.070 A
Current(A) : 0.035 A
Current(A) : 0.070 A
Current(A) : 0.070 A
Current(A) : 15.186 A
Current(A) : 15.256 A
Current(A) : 15.221 A
Current(A) : 15.117 A
Current(A) : 15.221 A
Current(A) : 15.221 A
Current(A) : 15.152 A
Current(A) : 15.186 A
```

(5.3) Single Output: AC:

Click on “SingleOutput_AC_Current” example, and upload the program into Arduino. Connect the WCS current sensor to the 5V power supply. The output pin (Vout) of the WCS current sensor connects the analog pin A0 of Arduino, as shown in the following figure.

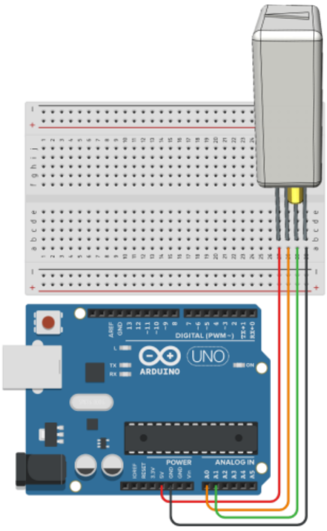


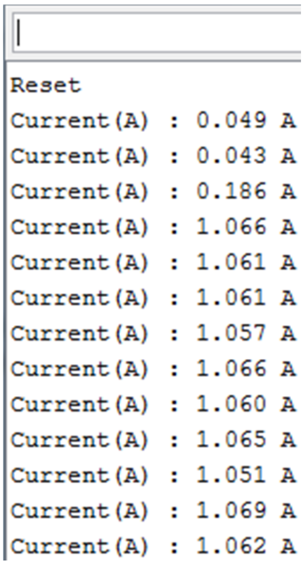


```
Reset
Current(A) : 0.137 A
Current(A) : 0.129 A
Current(A) : 0.125 A
Current(A) : 1.112 A
Current(A) : 1.074 A
Current(A) : 1.109 A
Current(A) : 1.109 A
Current(A) : 1.103 A
Current(A) : 1.107 A
Current(A) : 1.101 A
Current(A) : 1.102 A
Current(A) : 1.113 A
Current(A) : 1.094 A
```

(5.4) Differential Output: AC:

Click on “DifferentialOutput_AC_Current” example, and upload the program into Arduino. Connect the WCS current sensor to the 5V power supply. The output pin 1(Vout1) of the WCS current sensor connects the analog pin A0 of Arduino, and the output pin 2(Vout2) to A1, as shown in the following figure.



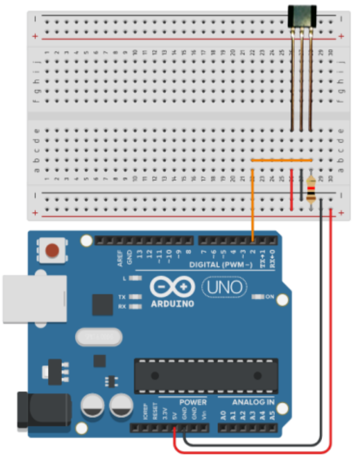


```
Reset
Current(A) : 0.049 A
Current(A) : 0.043 A
Current(A) : 0.186 A
Current(A) : 1.066 A
Current(A) : 1.061 A
Current(A) : 1.061 A
Current(A) : 1.057 A
Current(A) : 1.066 A
Current(A) : 1.060 A
Current(A) : 1.065 A
Current(A) : 1.051 A
Current(A) : 1.069 A
Current(A) : 1.062 A
```

6. Hall IC:

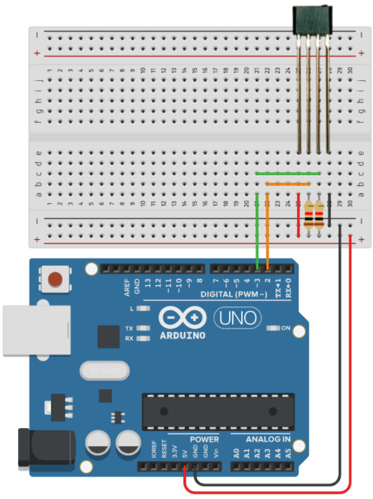
(6.1) Single Output: Switching:

Click on “SingleOutput_Switch_IC_Polling” example, and upload the program into Arduino. Connect the Hall switching IC to the 5V power supply. The output pin (Vout) of the Hall switching IC with external pull-up resistor (1KΩ) connects the pin 2 of Arduino, as shown in the following figure.



(6.2) Dual Output: Switching:

Click on “DualOutput_Switch_IC_Polling” example, and upload the program into Arduino. Connect the Hall switching IC to the 5V power supply. All the output pins of the Hall switching IC pull up resistor (1KΩ) respectively. The output pin 1(Vout1) of the Hall switching IC connects the pin 2 of Arduino, and the output pin 2(Vout2) to the pin 3, as shown in the following figure.



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

7. Function Description

(7.1) DWCS Class

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

Description:

DWCS: initialize DWCS Class (Continuous Mode, AT Command Mode)

Tx: DWCS TX Pin

Rx: DWCS RX Pin

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

Example:

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

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

Description:

DWCS: initialize DWCS Class (Modbus-RTU Mode)

Tx: DWCS TX Pin

Rx: DWCS RX Pin

Mode: **Modbus** Mode

SlaveAddress: Modbus-RTU slave address

Example:

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

3. void Init()

Description:

Initialize DWCS.

Example:

DWCS.init();

4. double mA()

Description:

Measure current, unit: mA.

Example:

data = DWCS.mA();

5. double mA(byte SlaveAddress)

Description:

Measure current according to Modbus-RTU slave address, unit: mA.

SlaveAddress: Modbus-RTU slave address

Example:

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

6. double A()

Description:

Measure current, unit: A.

Example:

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

7. double A(byte SlaveAddress)

Description:

Measure current according to Modbus-RTU slave address, unit: A.

SlaveAddress: Modbus-RTU slave address

Example:

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

8. double oC()

Description:

Measure temperature, unit: ˚C.

Example:

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

9. double oC(byte SlaveAddress)

Description:

Measure temperature according to Modbus-RTU slave address, unit: ˚C.

SlaveAddress: Modbus-RTU slave address

Example:

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

10. double oF()

Description:

Measure temperature, unit: ˚F.

Example:

data = DWCS.oF();

11. double oF(byte SlaveAddress)

Description:

Measure temperature according to Modbus-RTU slave address, unit: ˚F.

SlaveAddress: Modbus-RTU slave address

Example:

data = DWCS.oF(0x01);

12. bool Reset()

Description:

Reset current.

Example:

DWCS.Reset();

13. bool Reset (byte SlaveAddress)

Description:

Reset current according to Modbus-RTU slave address.

SlaveAddress: Modbus-RTU slave address

Example:

DWCS.Reset(0x01);

14. bool DC()

Description:

Switch DC current.

Example:

DWCS.DC();

15. bool DC (byte SlaveAddress)

Description:

Switch DC current according to Modbus-RTU slave address.

SlaveAddress: Modbus-RTU slave address

Example:

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

16. bool AC()

Description:

Switch AC current.

Example:

```
DWCS.AC();
```

17. bool AC (byte SlaveAddress)

Description:

Switch AC current according to Modbus-RTU slave address.

SlaveAddress: Modbus-RTU slave address

Example:

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

18. bool SetAddress(byte SlaveAddress)

Description:

Change the new Modbus-RTU slave address according to present slave address.

SlaveAddress: Modbus-RTU slave address

Example:

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

19. bool SetAddress(byte OldAddress, byte NewAddress)

Description:

Specify Modbus-RTU slave address and replace with a new slave address.

OldAddress: old Modbus-RTU slave address

NewAddress: new Modbus-RTU slave address

Example:

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

20. bool FactoryReset()

Description:

Restore factory Modbus-RTU slave address (0x01).

<Note> This command uses podcast address (0x00), it is recommended to use single sensor for setting.

Example:

DWCS. FactoryReset();

21. byte addr()

Description:

Read Modbus-RTU Slave Address.

Example :

address = DWCS. addr();

(7.2) WCM Class

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

Description:

WCM: initialize WCM Class (Continuous Mode)

Tx: WCM TX Pin

Rx: WCM RX Pin

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

Example:

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

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

Description:

WCM: initialize WCM Class (Modbus-RTU Mode)

Tx: WCM TX Pin

Rx: WCM RX Pin

Rst: WCM RST Pin

Mode: **Modbus** Mode

SlaveAddress: Modbus-RTU slave address

Example:

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

3. void Init()

Description:

Initialize WCM.

Example:

WCM.init();

4. double mA()

Description:

Measure current, unit: mA.

Example:

data = WCM.mA();

5. double mA(byte SlaveAddress)

Description:

Measure current according to Modbus-RTU slave address, unit: mA.

SlaveAddress: Modbus-RTU slave address

Example:

data = WCM.mA(0x01);

6. double A()

Description:

Measure current, unit: A.

Example:

data = WCM.A();

7. double A(byte SlaveAddress)

Description:

Measure current according to Modbus-RTU slave address, unit: A.

SlaveAddress: Modbus-RTU slave address

Example:

data = WCM.A(0x01);

8. double oC()

Description:

Measure temperature, unit: ˚C.

Example:

data = WCM.oC();

9. double oC(byte SlaveAddress)

Description:

Measure temperature according to Modbus-RTU slave address, unit: ˚C.

SlaveAddress: Modbus-RTU slave address

Example:

data = WCM.oC(0x01);

10. double oF()

Description:

Measure temperature, unit: ˚F.

Example:

data = WCM.oF();

11. double oF(byte SlaveAddress)

Description:

Measure temperature according to Modbus-RTU slave address, unit: ˚F.

SlaveAddress: Modbus-RTU slave address

Example:

data = WCM.oF(0x01);

12. bool Reset()

Description:

Reset current.

Example:

WCM. Reset();

13. bool Reset (byte SlaveAddress)

Description:

Reset Current according to Modbus-RTU slave address.

SlaveAddress: Modbus-RTU slave address

Example:

WCM.Reset(0x01);

14. bool SetAddress(byte SlaveAddress)

Description:

Change the new Modbus-RTU slave address according to present slave address.

Example:

WCM.SetAddress(0x02);

15. bool SetAddress(byte OldAddress, byte NewAddress)

Description:

Specify Modbus-RTU slave address and replace with a new slave address.

OldAddress: old Modbus-RTU slave address

NewAddress: new Modbus-RTU slave address

Example:

WCM.SetAddress(0x00, 0x02);

16. bool FactoryReset()

Description:

Restore factory Modbus-RTU slave address (0x01).

<Note> This command uses podcast address (0x00), it is recommended to use single sensor for setting.

Example:

WCM. FactoryReset();

17. byte addr()

Description:

Read Modbus-RTU slave address.

Example:

address = WCM. addr();

(7.3) WCS Class

1. WCS(uint8_t analogPin, uint16_t mVperA)

Description:

WCS: initialize WCS Class (Single Output Mode)

analogPin: WCS output pin

mVperA: WCS sensitivity (as shown in table)

mVperA	Sensitivity (mV/A)	mVperA	Sensitivity (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		

<Note> you can also enter the value directly.

Example:

WCS WCS (0, _**WCS1800**);

WCS WCS (0, 66);

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

Description:

WCS: initialize WCS Class (Differential Output)

analogPin: WCS output pin 1

analogPin2: WCS output Pin 2

mVperA: WCS sensitivity (as shown in table)

Example:

WCS WCS (0, 1, _**WCS2200**);

3. void Reset()

Description:

Reset current.

Example:

WCS.Reset();

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

4. double A_AC()

Description:

Measure AC current.

Example:

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

5. double A_DC()

Description:

Measure DC current.

Example:

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

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