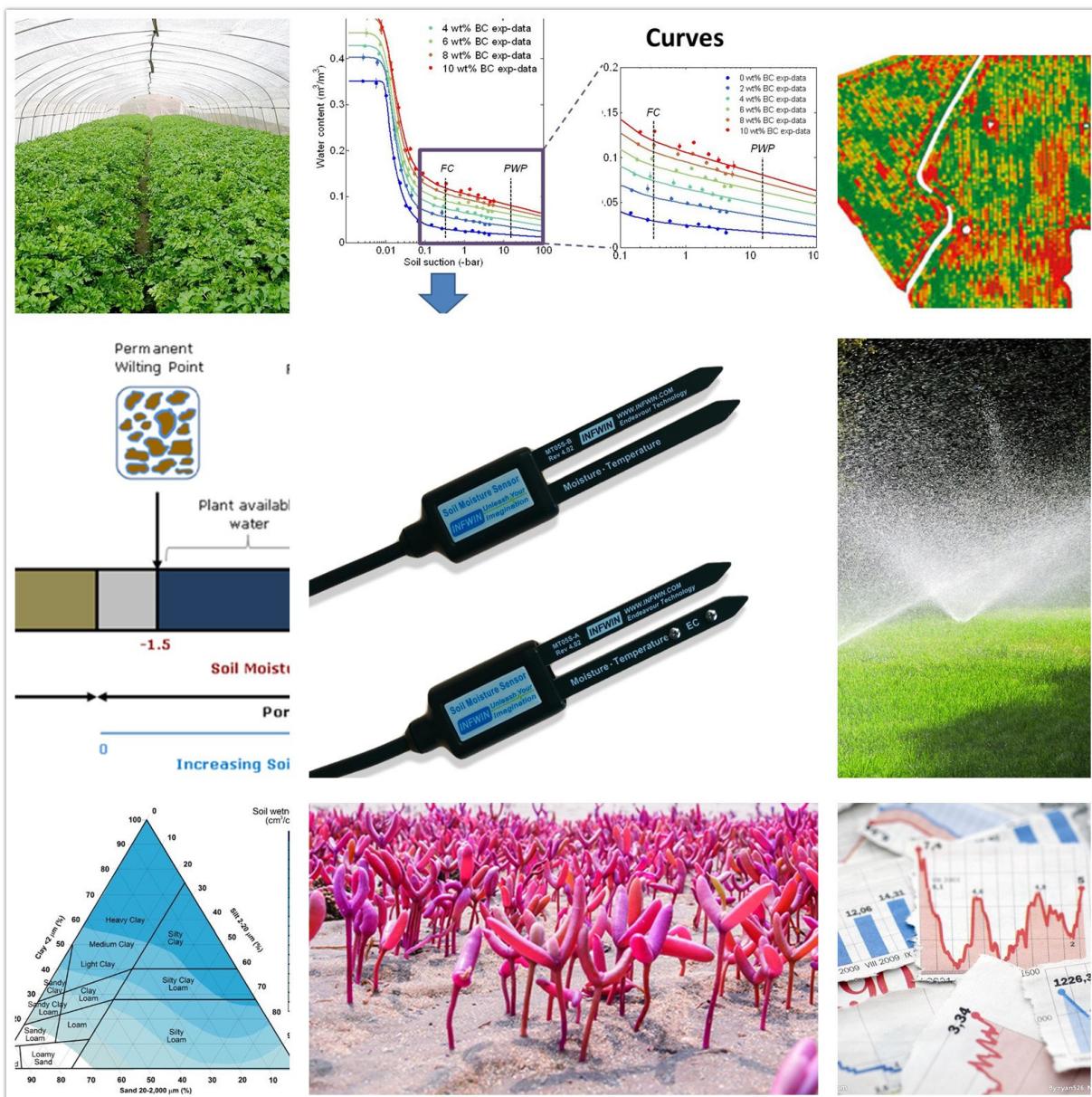


Application Note

MT05S 1-Wire Soil Moisture, EC, Temperature Sensor Demo and Testing



Index

1 Customer Support.....	3
2 MT05S Introduction	4
2.1 Key Features.....	4
2.2 Ordering Infomation	4
2.3 Wiring	5
3 Testing case and Preparation	5
3.1 Testing Case.....	5
3.2 Arduino IDE ESP32 Support Package	6
3.3 Visual Studio Code and Platform IO IDE extension	10
4 Demo on DOIT ESP32 DevKit V1	14
4.1 Preparation.....	14
4.2 Wiring and Connection	14
4.3 Platform IO.....	16
4.4 Arduino IDE	17
5 Demo on NodeMCU-32S	18
5.1 Preparation.....	18
5.2 Wiring and Connection	19
5.3 Platform IO.....	20
5.4 Arduino IDE	21
6 Demo on RaspberryPI 3 Model B	22
6.1 Preparation.....	22
6.2 Wiring and Connection	23
6.3 Program	24
7 Demo on Arduino UNO R3	27
7.1 Preparation.....	27
7.2 Wiring and Connection	27
7.3 Platform IO.....	29
7.4 Arduino IDE	30
Copyright and Trademark.....	31
Revision.....	31

1 Customer Support

Thank you very much for your order. Our success comes from the continuous faith in the excellence of our products and services, something we are committed to and would never sacrifice. Our customer service, especially in the after sales phase, guarantees the satisfaction of our clients. In line with this strategy, we appreciate that you can share with us your feedback at any time for our improvement, be it positive or negative, so if we can serve you better in anyway, please do inform us.

Website

<http://www.infwin.com>

E-Mail

infwin@163.com

Telephone

+86-411-66831953, +86-4000-511-521

Fax

+86-411-82388125

2 MT05S Introduction

2.1 Key Features

MT05S is 1-Wire soil moisture sensor, measuring soil moisture content, temperature and EC(Conductivity). It sealed with resin packaged plastic body with sensing rods which can be insert directly into the soil with long time stability. Sensor with relatively small size and can be used for pot culture and Seedling tray. The sensor is applicable for science research, irrigation, greenhouse, smart agriculture etc.

- MT05S-A for Soil Moisture, Temperature, EC measurement.
- MT05S-B for Soil Moisture, Temperature measurement.
- Power Supply 2.7-12V DC, Low Quiescent current Max. 30uA.
- Water proof to IP68 ratings and can be directly buried into soil.
- 1-Wire bus interface, Operation compatible with DS18B20.
- Connect to MCU, Arduino, Raspberry PI etc by ONE I/O pin with a PULL-UP resistor.
- Software library for DS18B20 can be reused to read/write the sensor.
- Small Dimension 20*11*95mm with high accuracy and excellent stability.
- Operating temperature -40~85°C.

2.2 Ordering Infomation

Parameters	Code	Comments
Code 1: Product Series	MT05S	MT05S series sensor with 1-Wire Interface
Code 2: Measuring Parameters	A B	Soil Moisture & Temperature & EC Soil Moisture & Temperature
Code 3: Power Supply	B	2.7-12V DC
Code 4: Connector	B C	Cold pressed terminal pre-tinning wire
Code 5: Cable Length	005	5 meters
Ordering Code Example: MT05S sensor with 1-Wire Interface, Measuring Parameters Soil Moisture & Temperature & EC, 2.7-12VDC Power supply, Cold pressed terminal, Cable Length 5 meters. Ordering Code is : MT05S-ABB005		

2.3 Wiring

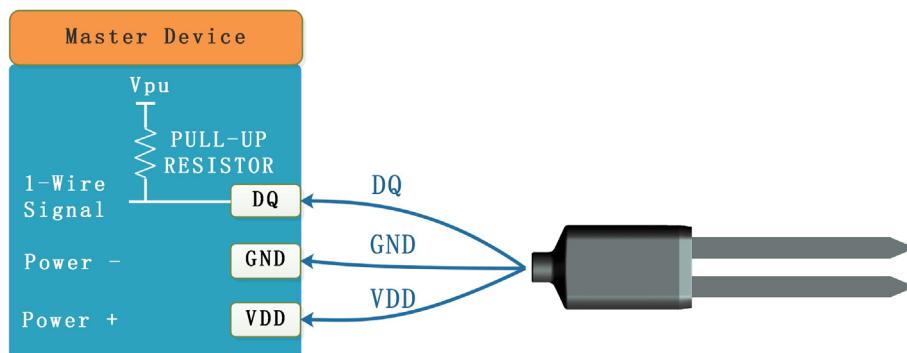
Cold pressed terminal



Tinned lead wires



Wiring Diagram



3 Testing case and Preparation

3.1 Testing Case

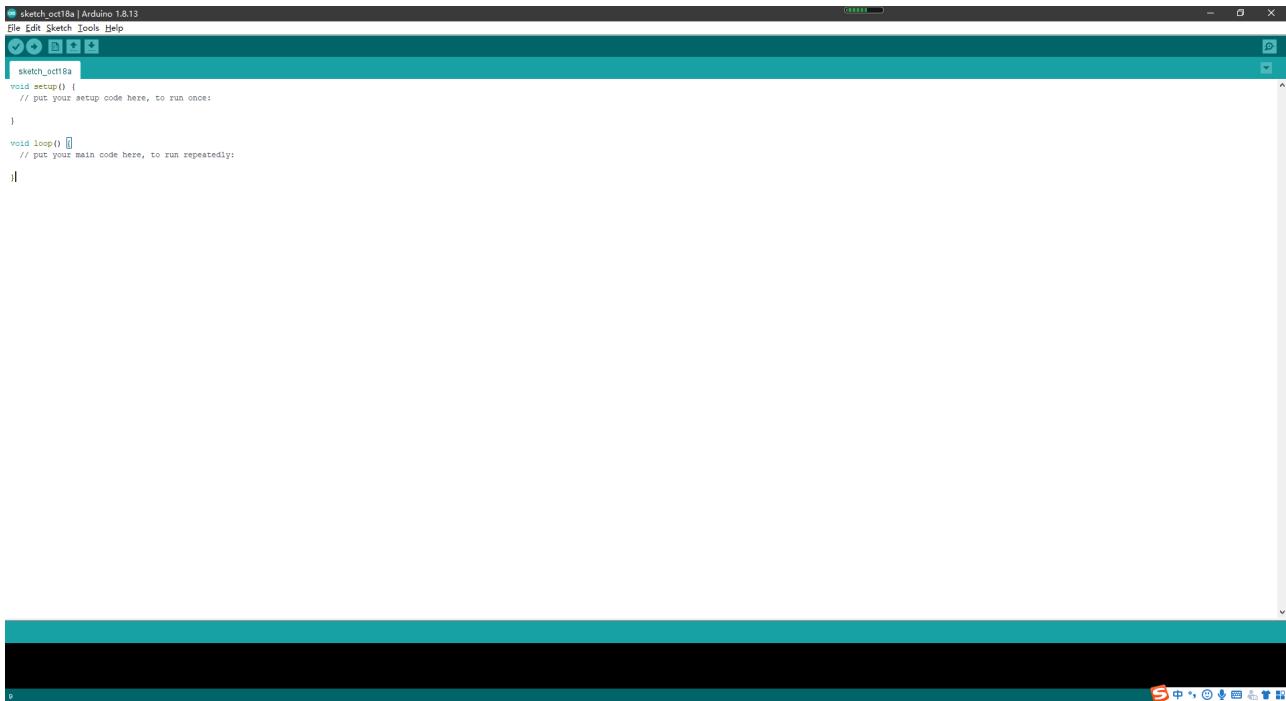
The testings are based on following platform and boards. It can be easily migrated to others.

Platform	Boards
Arduino IDE	ESP32 DOIT DevKit V1, NodeMCU 32S
Visual Studio Code	ESP32 DOIT DevKit V1, NodeMCU 32S
Raspberry PI	Raspberry PI 3 Model B

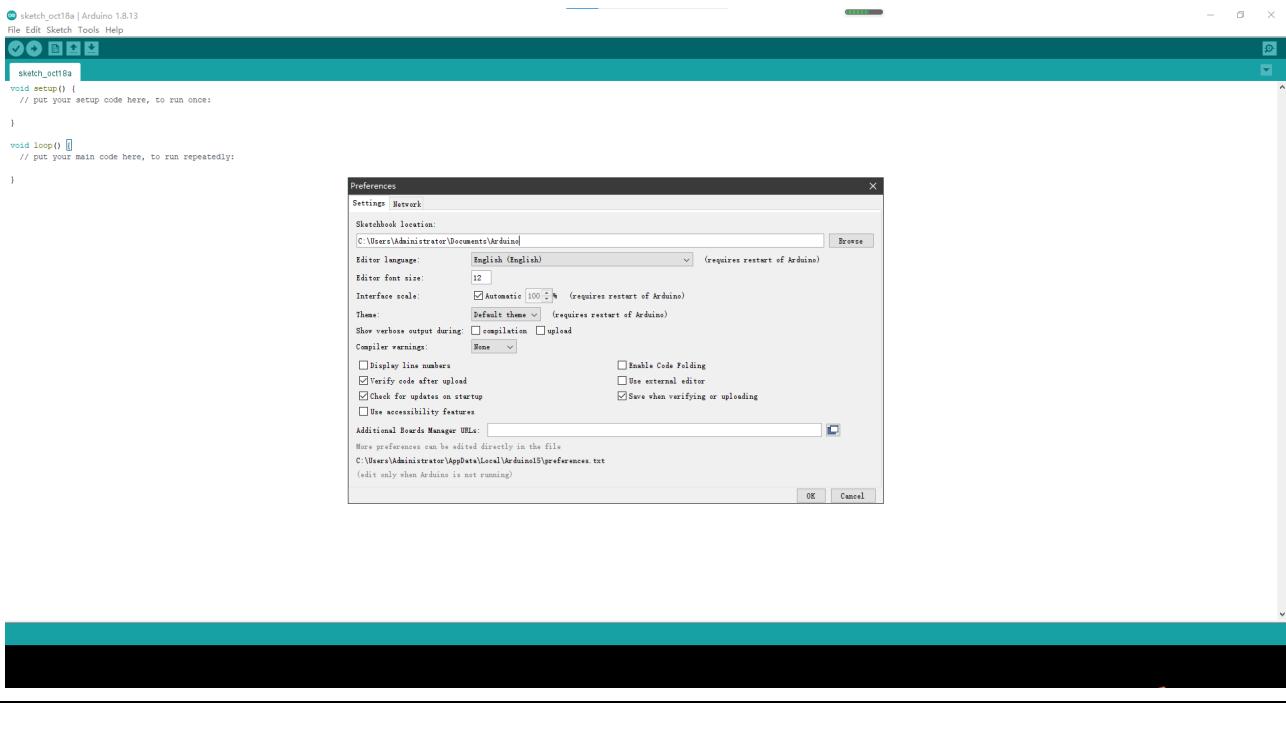
3.2 Arduino IDE ESP32 Support Package

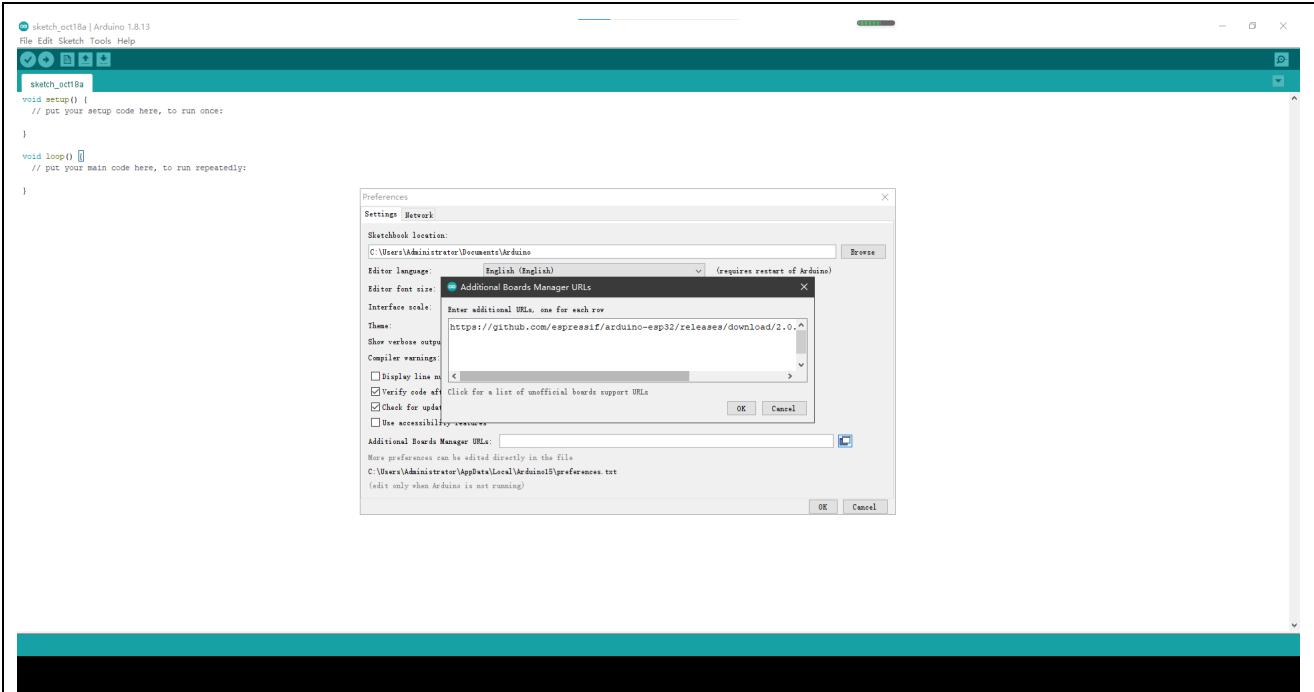
Please Install Arduino IDE first, and follow steps below to install support package or libraries.

● Startup Arduino IDE

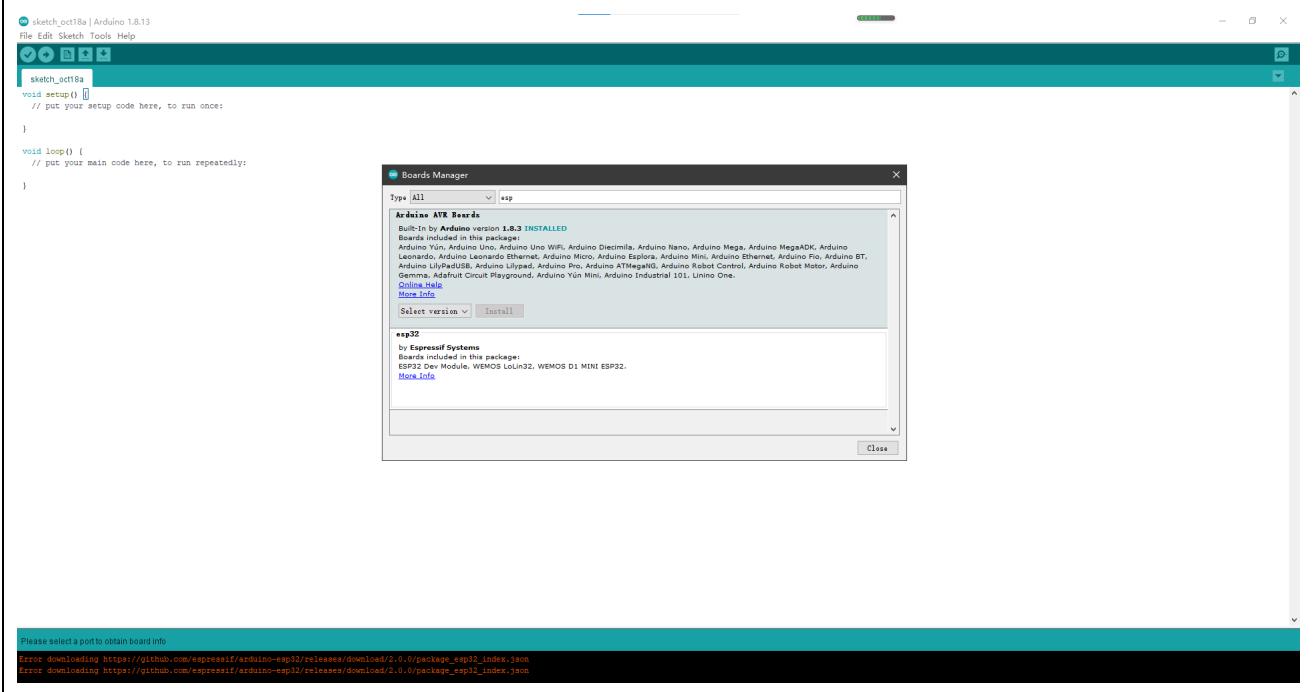


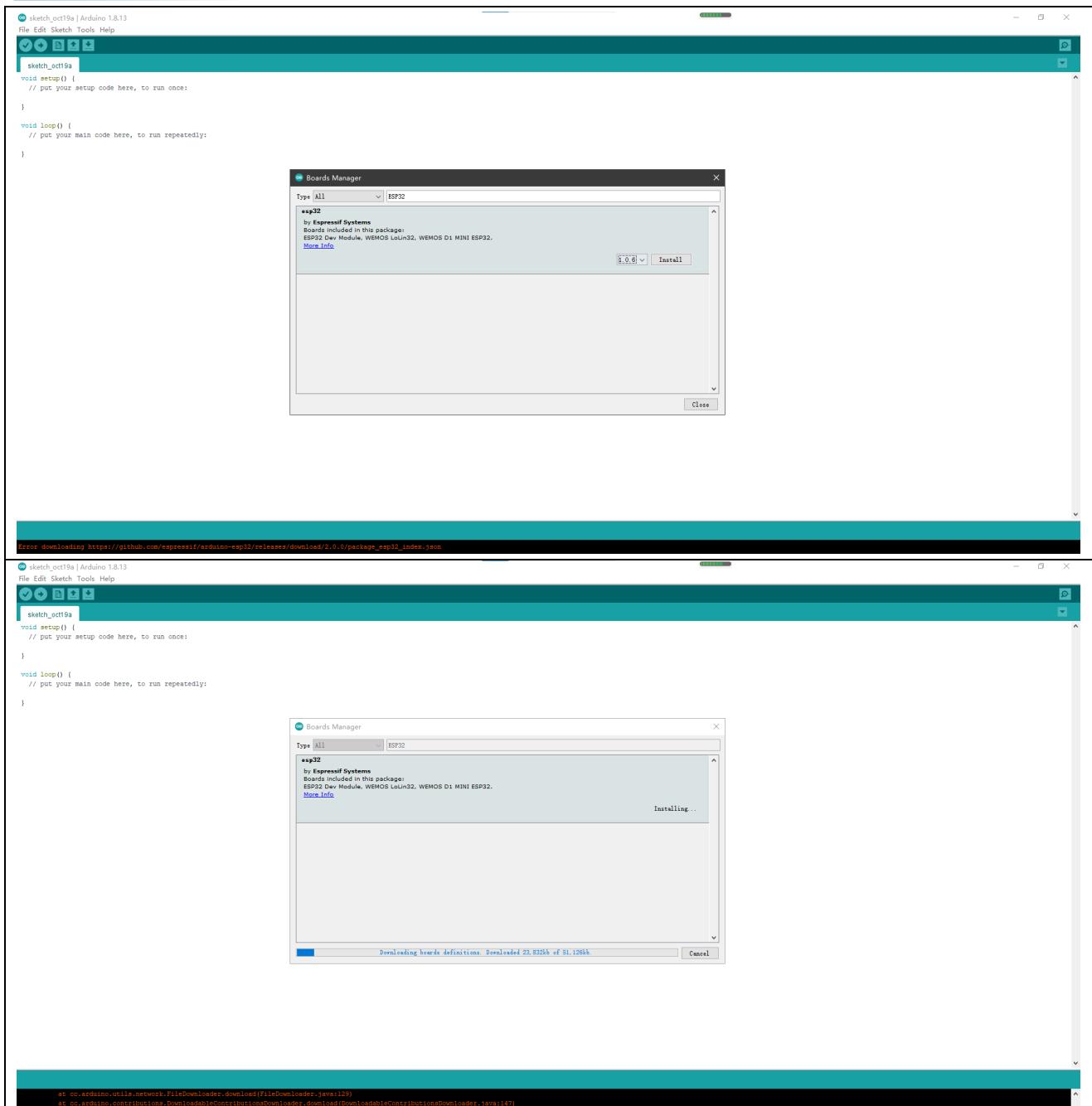
- Install “Additional Boards Manager URLs” in Menu>File>Preference, input https://github.com/espressif/arduino-esp32/releases/download/2.0.0/package_esp32_index.json

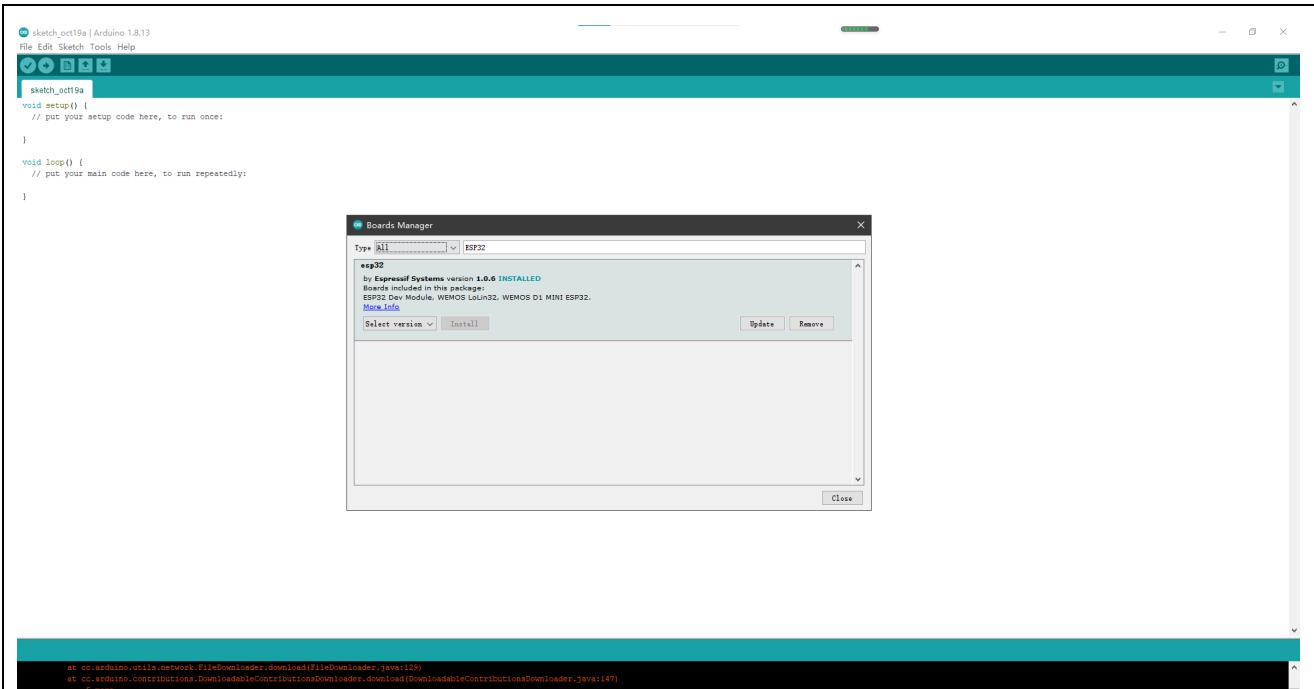




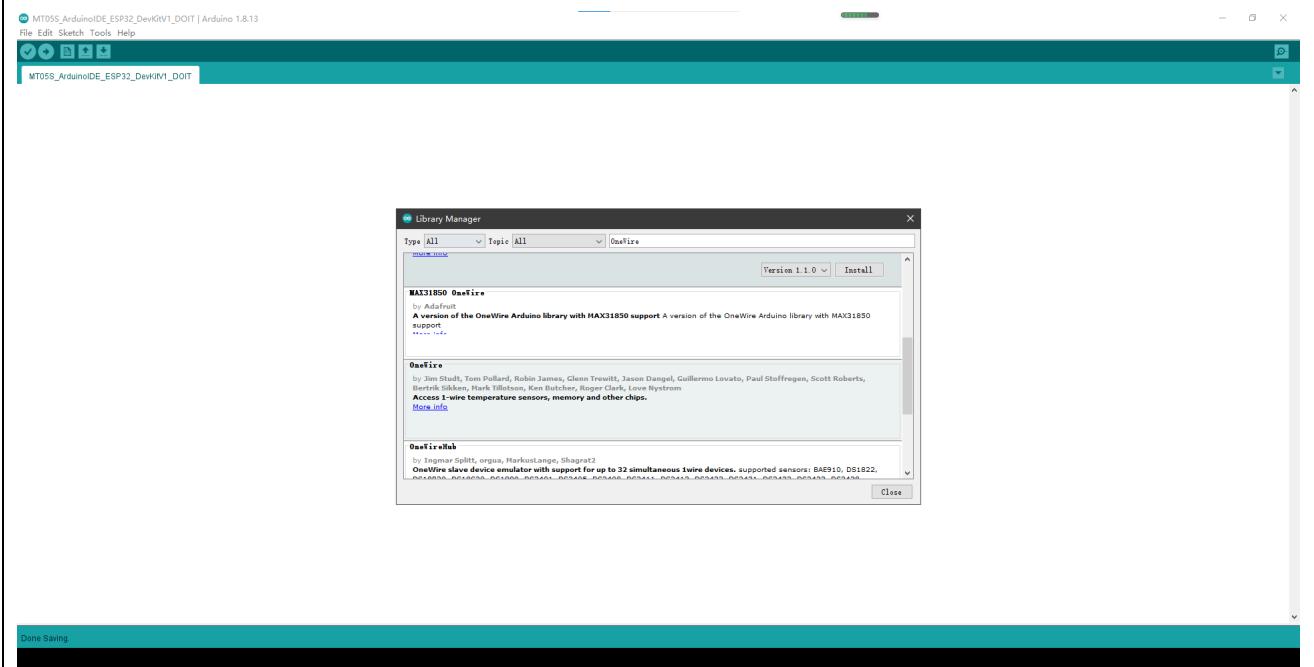
- Install Boards support package in Menu>Tools>Board>Boards Manager, then find ESP32 and select version “1.0.6” to install.

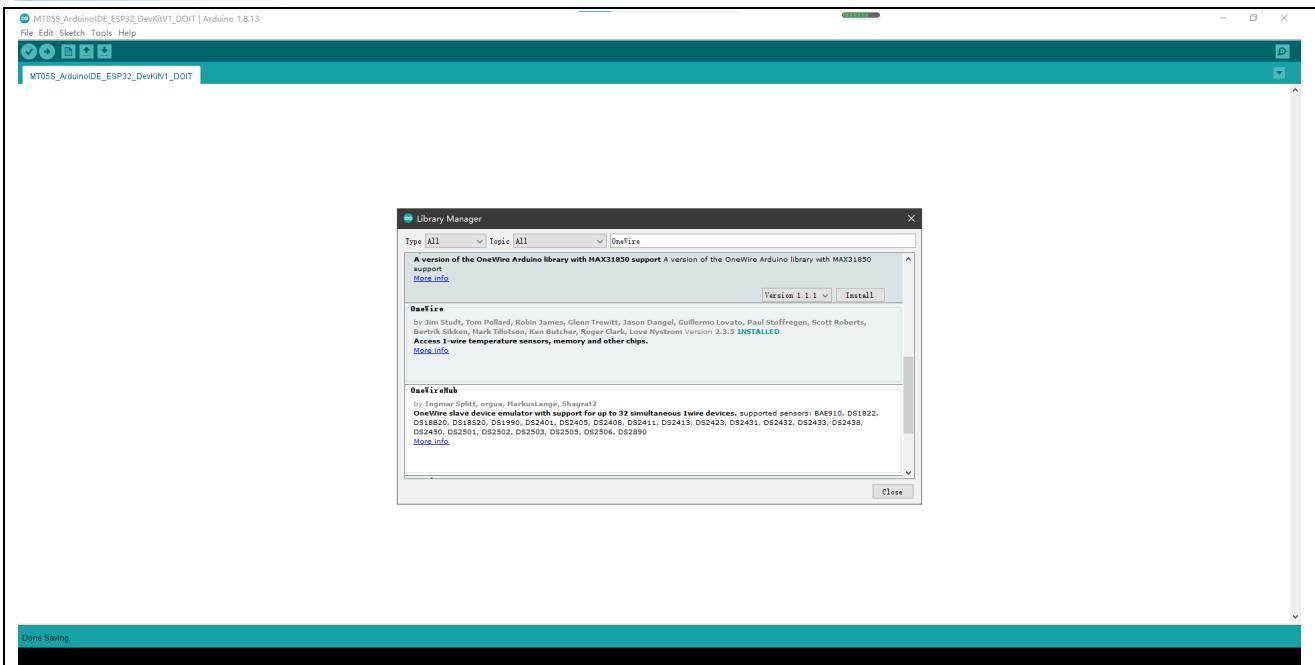






- Install OneWire support library in Menu>Tools>Manage Libaries, search OneWire library and then install.

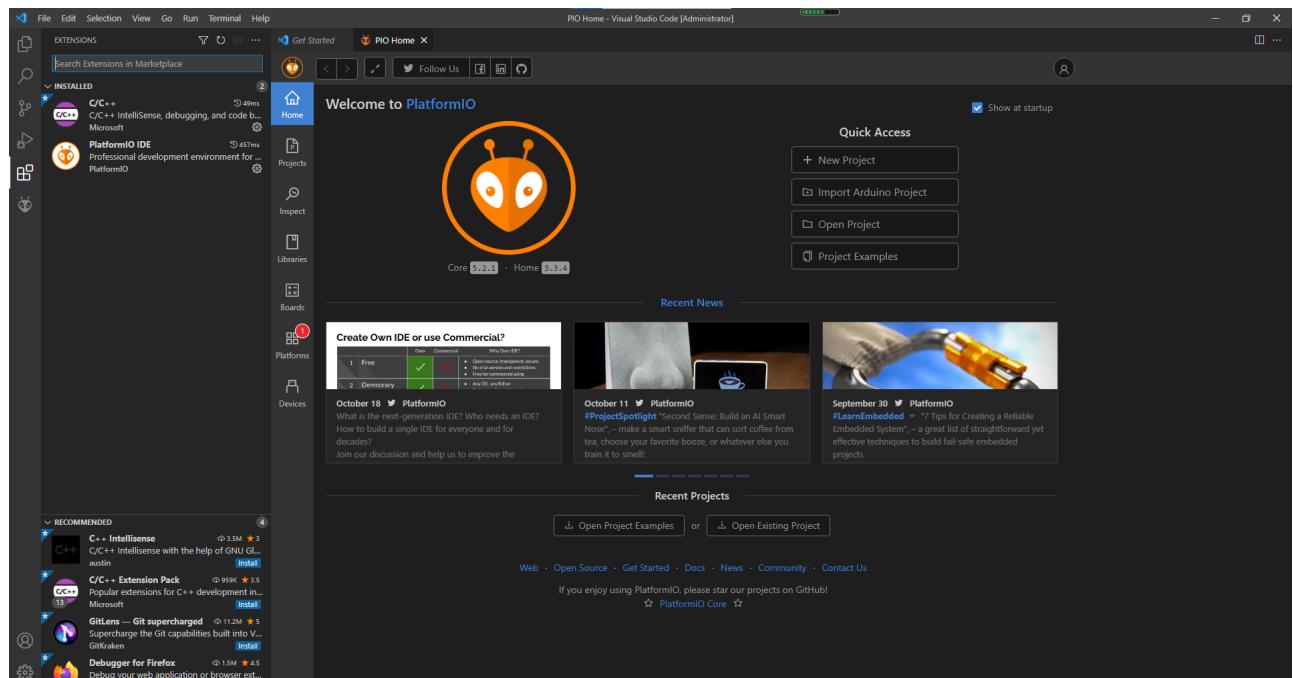




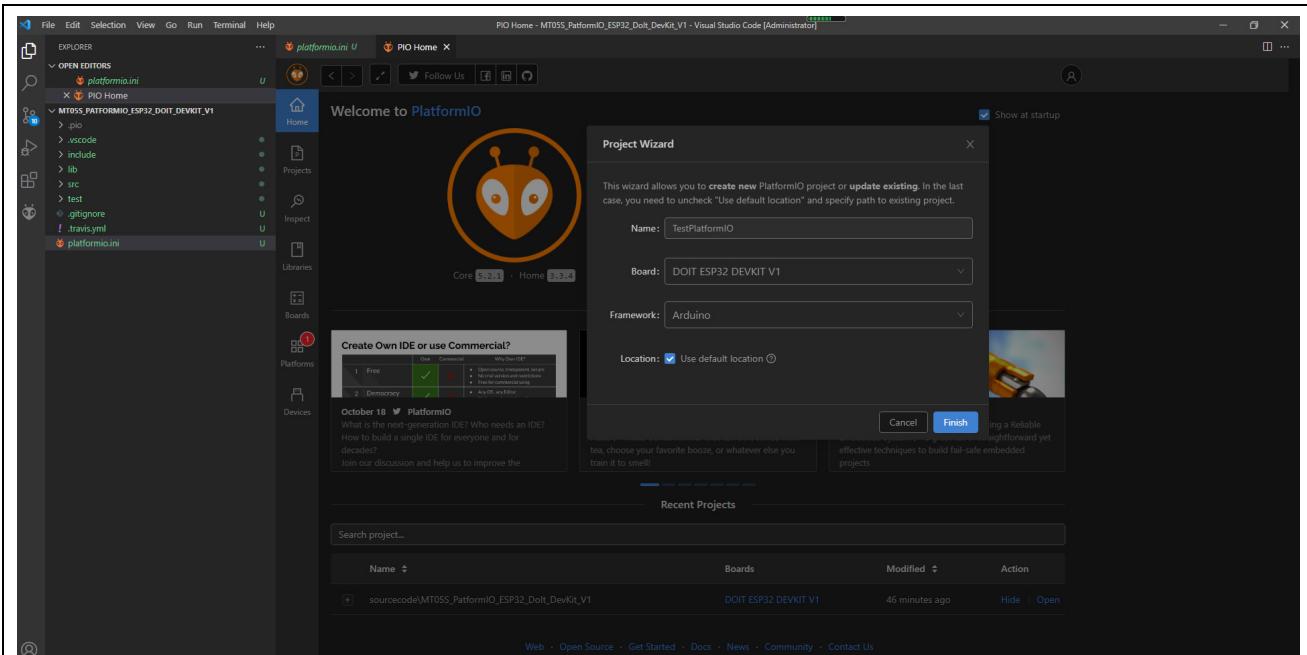
3.3 Visual Studio Code and Platform IO IDE extension

Please Install Visual Studio Code and Platform IO IDE extension.

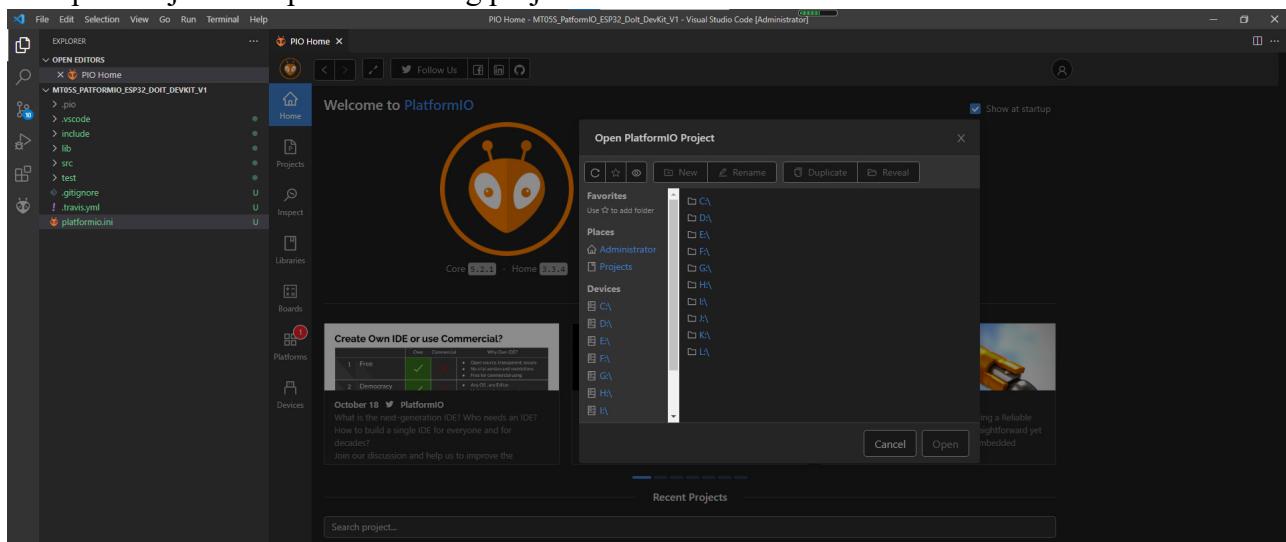
- Startup Visual Studio Source Code and Open Platform IO IDE home.



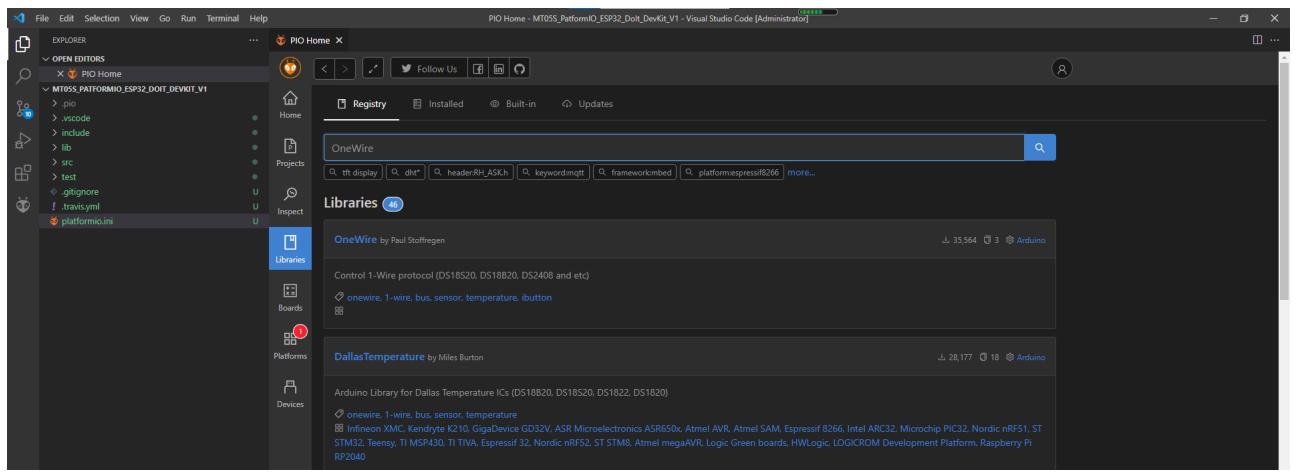
- “New Project” to create a new project and specify the board you use.

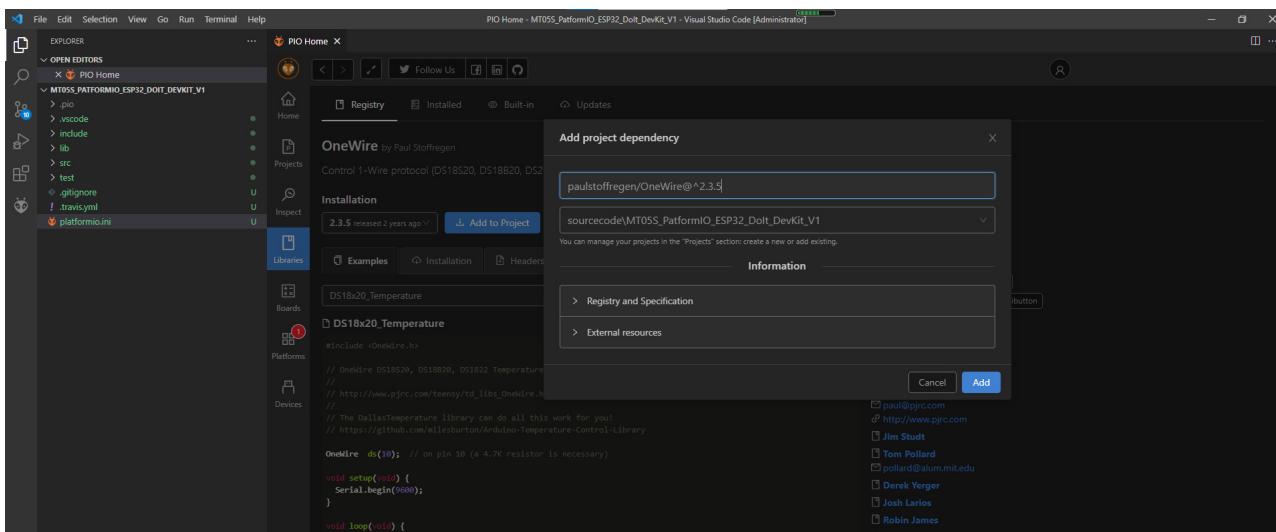
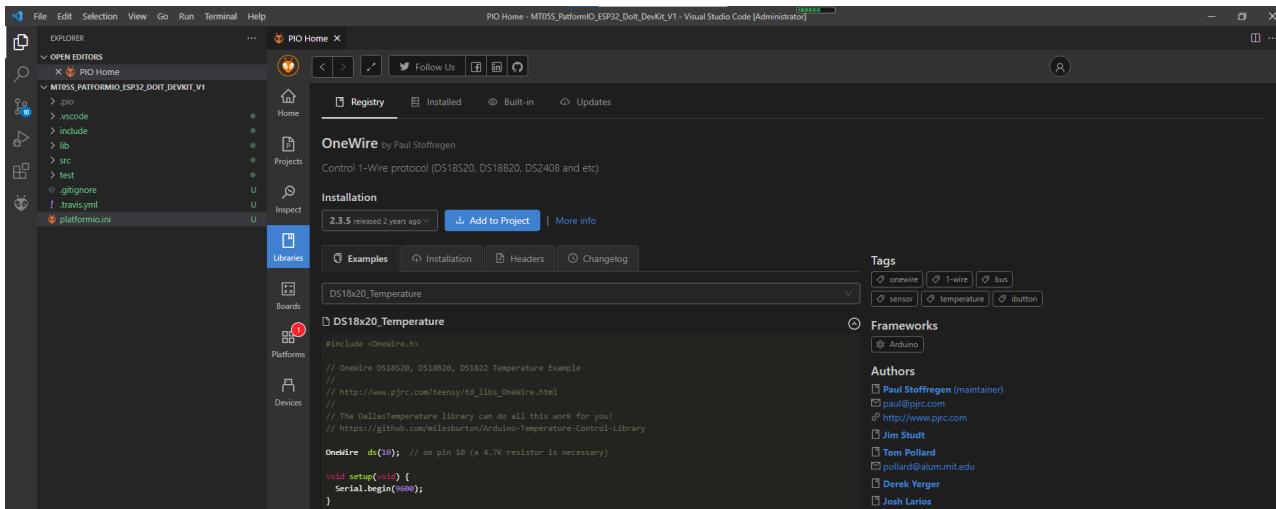


- “Open Project” to open an existing project.

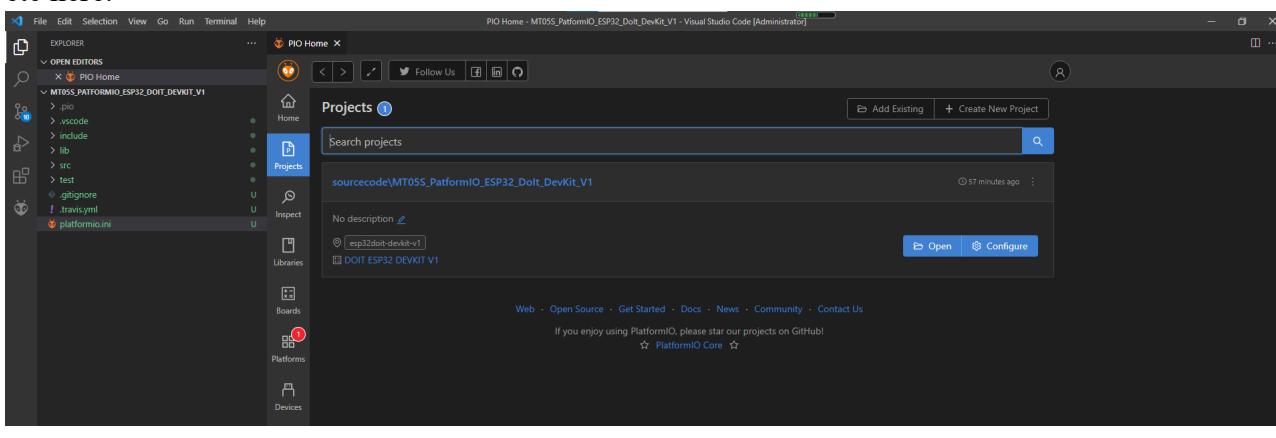


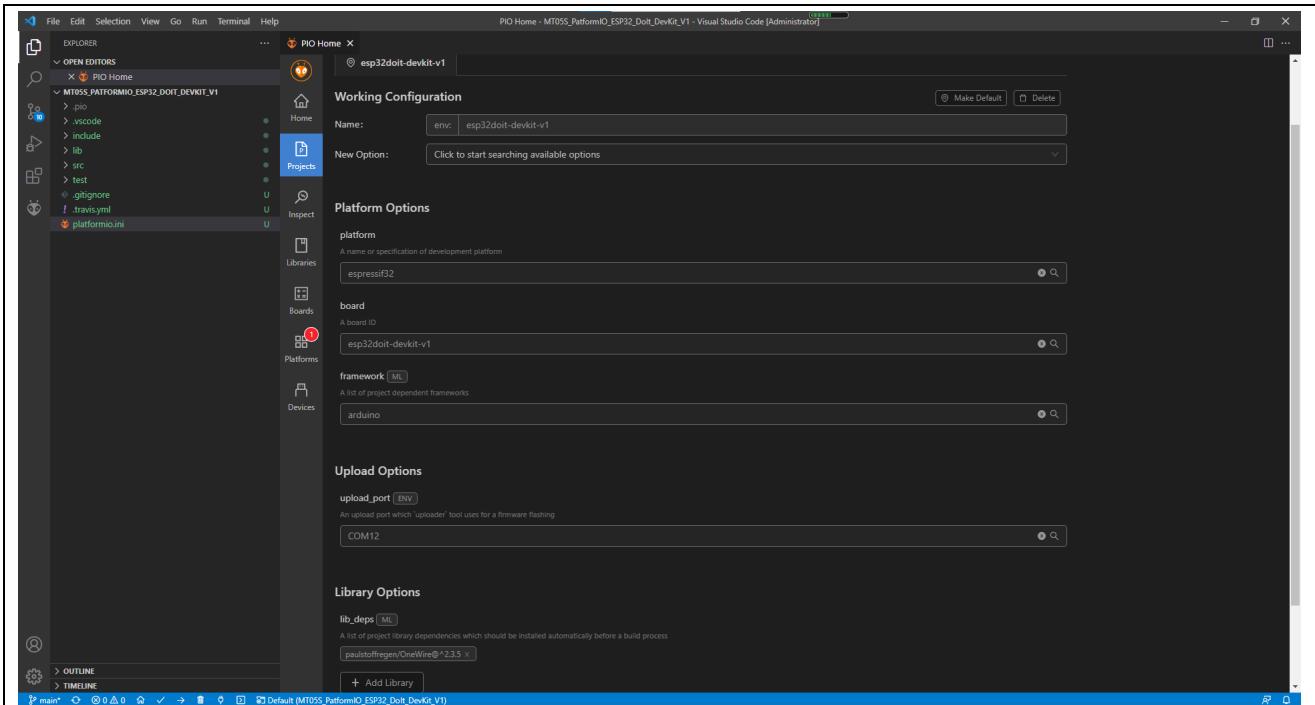
- Install libraries you need after opening a project.



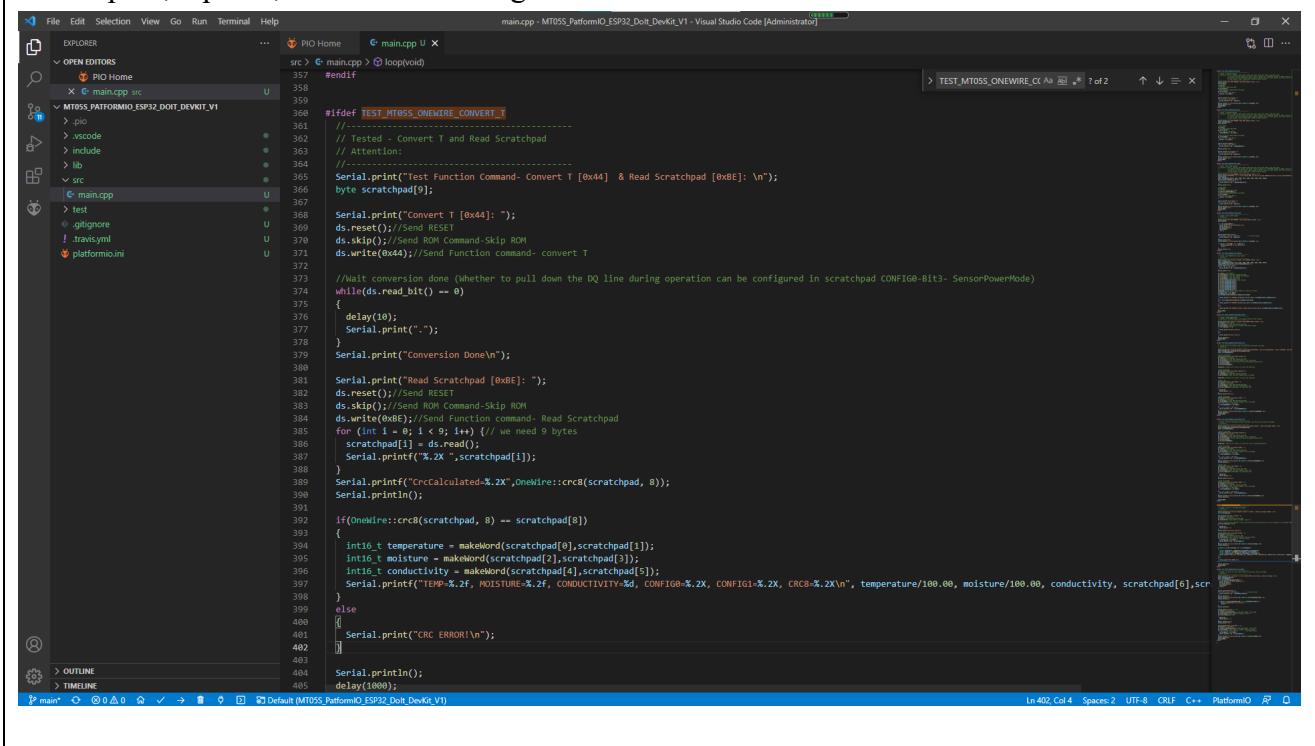


- Configure your project after opening a project. You can configure upload and monitor Port, Board etc here.





- Compile , Upload, and Monitor using icon button listed in bottom toolbox.

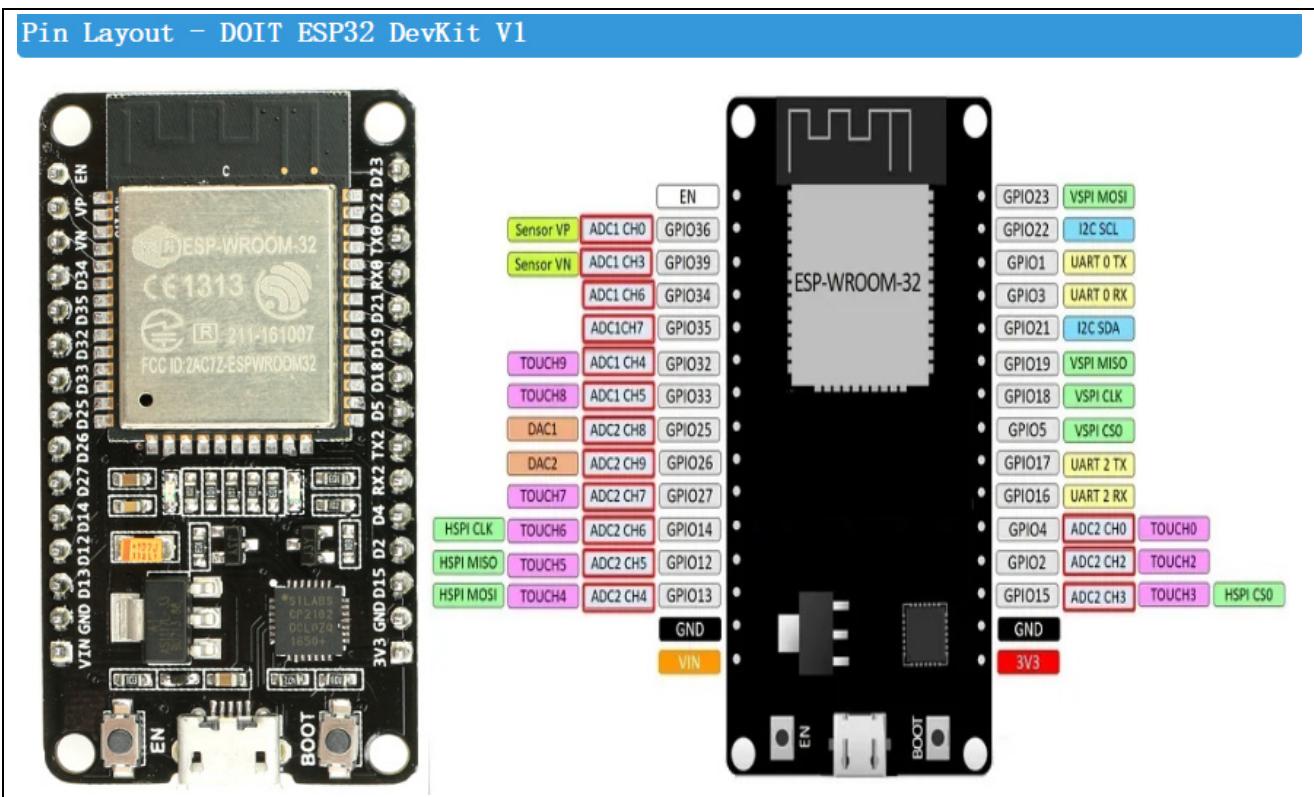


4 Demo on DOIT ESP32 DevKit V1

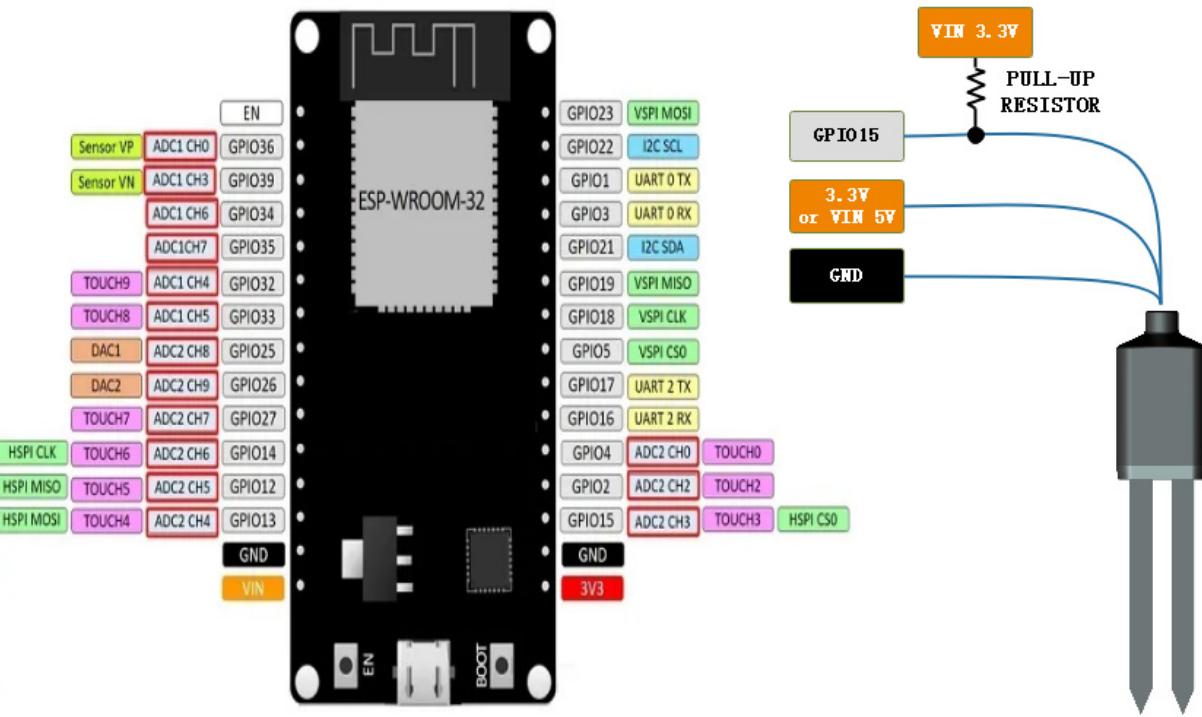
4.1 Preparation

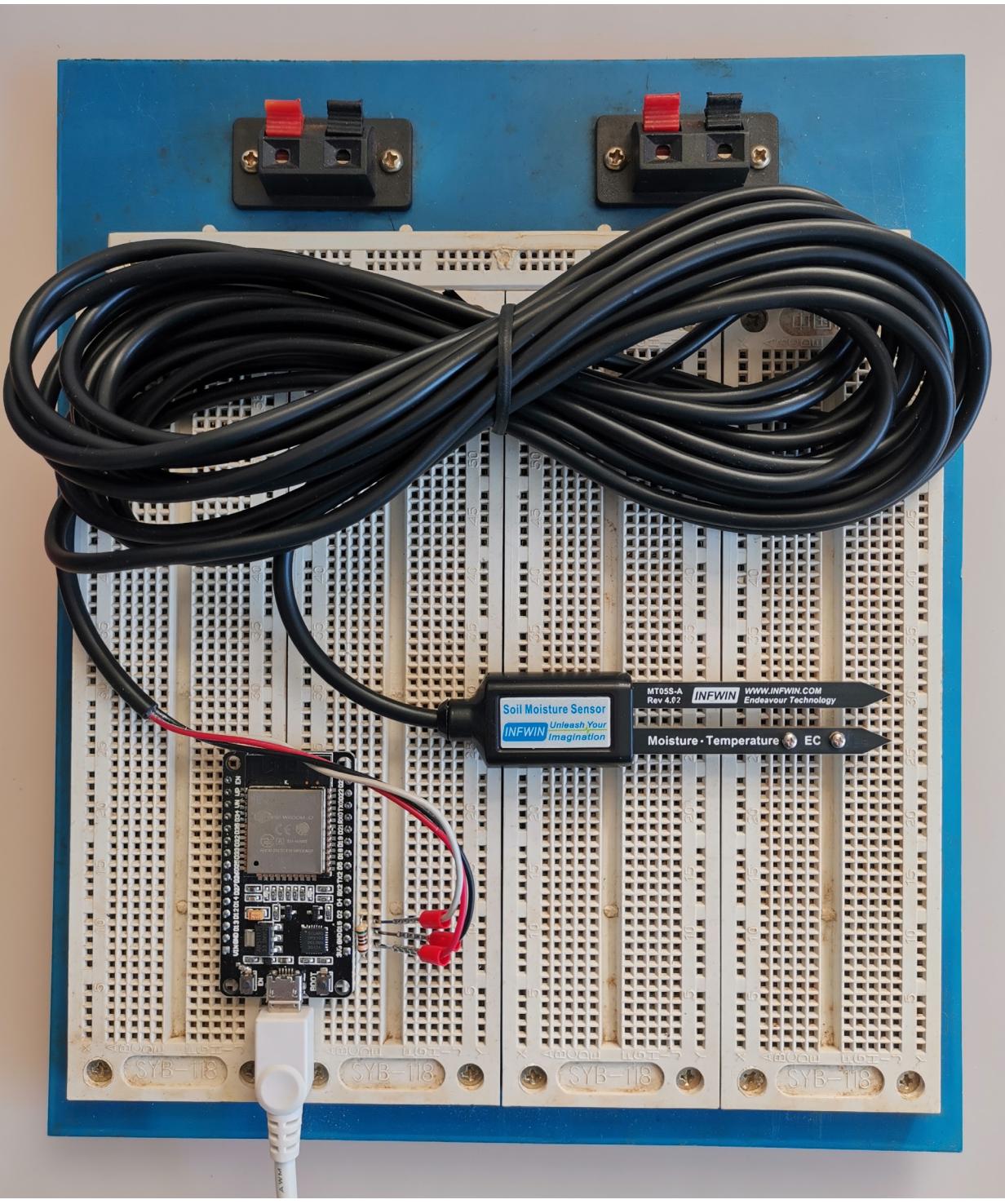
In this demo we use GPIO15 as 1-Wire signal I/O, Please pull up this IO by a resistor range from 2.0~5.1 Kohm to VIN3.3V.

4.2 Wiring and Connection



Wiring Diagram – DOIT ESP32 DevKit V1



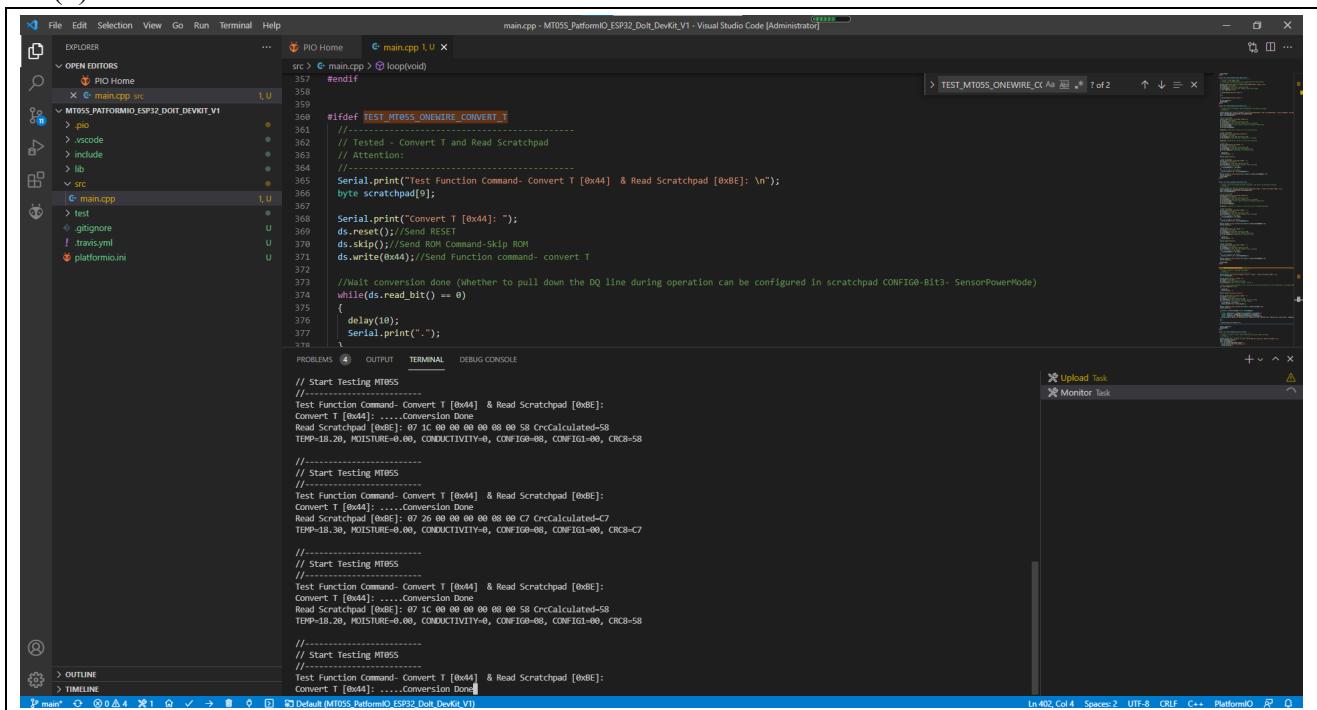


4.3 Platform IO

- Download Platform IO project at : <https://github.com/INFWIN/mt05s-demo>
- Open Platform IO project in sourcecode folder: MT05S_PlatformIO_DOIT_ESP32_DevKit_V1
 - (1) Select proper PORT, Board(DOIT ESP32 DevKit V1) in SideBar Projects>Configure.
 - (2) Build

(3) Upload

(4) Monitor



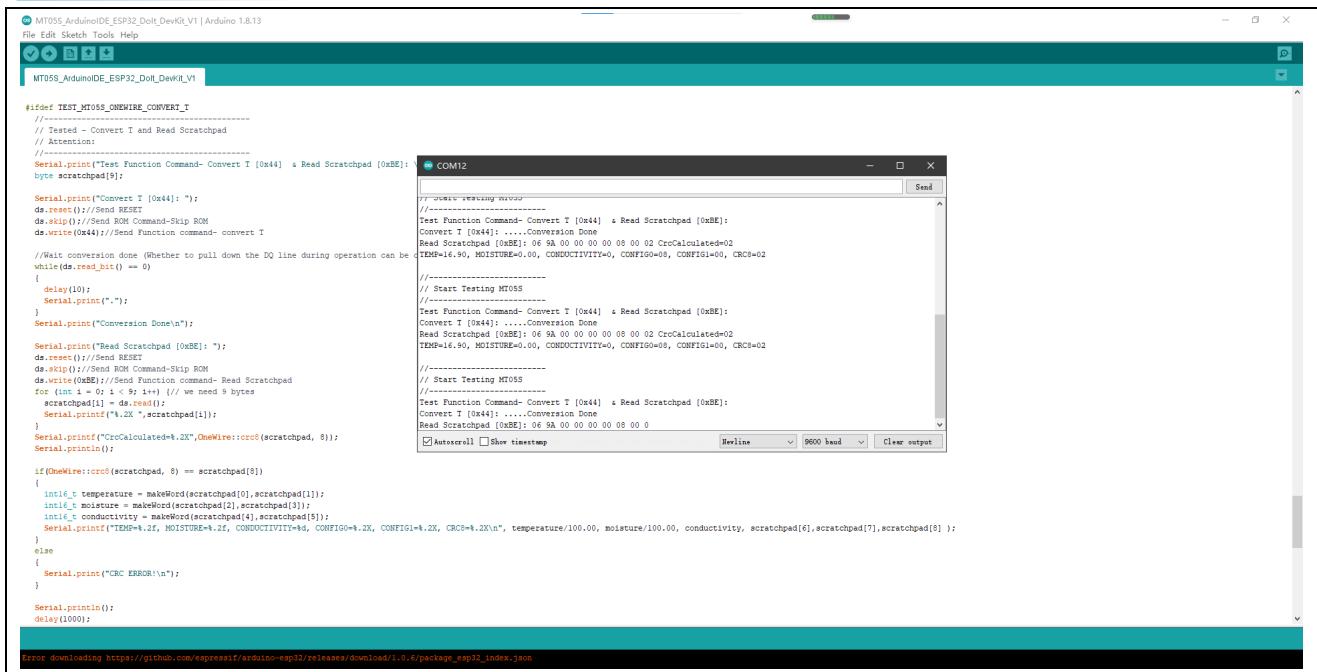
```

File Edit Selection View Go Run Terminal Help
... PIO Home main.cpp 1.U ...
src > main.cpp > loop()
357 #endif
358 //-----
359 #ifdef TEST_MT05S_ONewire_CONVERT_T
360 // Test - Convert T and Read Scratchpad
361 // Attention:
362 // -----
363 Serial.print("Test Function Command- Convert T [0x44] & Read Scratchpad [0x8E]: \n");
364 byte scratchpad[9];
365
366 Serial.print("Convert T [0x44]: ");
367 ds.reset(); //Send RESET
368 ds.skipROM(); //Send ROM Command-Skip ROM
369 ds.write(0x44); //Send Function command- convert T
370
371 //Wait conversion done (Whether to pull down the DQ line during operation can be configured in scratchpad CONFIG0-Bit3- SensorPowerMode)
372 while(ds.readBit() == 0)
373 {
374 delay(10);
375 Serial.print(".");
376 }
377
378 // Start Testing MT05S
379 //-----
380 // Test Function Command- Convert T [0x44] & Read Scratchpad [0x8E];
381 // Convert T [0x44]: .....Conversion Done
382 // Read Scratchpad [0x8E]: 07 1C 00 00 00 00 00 00 58 CrcCalculated-C7
383 // TEMP=-18.20, MOISTURE=0.00, CONDUCTIVITY=0, CONFIG0=00, CONFIG1=00, CRC8=58
384
385 // Start Testing MT05S
386 //-----
387 // Test Function Command- Convert T [0x44] & Read Scratchpad [0x8E];
388 // Convert T [0x44]: .....Conversion Done
389 // Read Scratchpad [0x8E]: 07 1C 00 00 00 00 00 00 58 CrcCalculated-S8
390 // TEMP=-18.20, MOISTURE=0.00, CONDUCTIVITY=0, CONFIG0=00, CONFIG1=00, CRC8=58
391
392 // Start Testing MT05S
393 //-----
394 // Test Function Command- Convert T [0x44] & Read Scratchpad [0x8E];
395 // Convert T [0x44]: .....Conversion Done
396

```

4.4 Arduino IDE

- Download Arduino IDE project: <https://github.com/INFWIN/mt05s-demo>
- Open Arduino IDE project in sourcecode folder:MT05S_ArduinoIDE_DOIT_ESP32_DevKit_V1
 - (1) Select proper PORT: Menu>Tools>Port
 - (2) Select Board: Menu>Tools>Board>Arduino ESP32>DOIT ESP32 DevKit V1
 - (3) Compile: Menu>Sketch>Verify/Compile
 - (4) Upload: Menu>Sketch>Upload
 - (5) Monitor: Menu>Tools>Serial Monitor



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, Help, and a toolbar with icons for file operations. The main window displays the code for the MT055 sensor, which includes comments and serial print statements for testing. The bottom status bar shows an error message about downloading the package index. On the right side, a serial monitor window titled 'COM12' is open, showing the raw data being received from the sensor. The data includes commands like 'Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE]' and responses such as 'Convert T [0x44]:Conversion Done' and 'Read Scratchpad [0xBE]: 0F 9A 00 00 00 00 00 02'. The serial monitor also has settings for 'Autoscroll', 'Show timestamp', 'Newline', and a baud rate of 9600.

```
#ifndef TEST_MT055_Onewire_CONVERT_T_
#define TEST_MT055_Onewire_CONVERT_T_
//-----//
// Tested - Convert T and Read Scratchpad
// Attention:
//-----//
Serial.print("Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE]:\n");
byte scratchpad[9];
Serial.print("Convert T [0x44]: ");
ds.reset(); //Send RESET
ds.skip(0); //Send ROM Command-Skip ROM
ds.write(0x44); //Send Function command- convert T
//Wait conversion done (Whether to pull down the DQ line during operation can be
while(ds.read_bit() == 0)
{
    delay(10);
    Serial.print(".");
}
Serial.print("Conversion Done\n");

Serial.print("Read Scratchpad [0xBE]: ");
ds.reset(); //Send RESET
ds.skip(0); //Send ROM Command-Skip ROM
ds.write(0x4E); //Send Function command- Read Scratchpad
for (int i = 0; i < 9; i++) // we need 9 bytes
{
    scratchpad[i] = ds.read();
    Serial.print("0x", 2, scratchpad[i]);
}
Serial.print("CrcCalculated=4.2X,OneWire::crc8(scratchpad, 8);\n");
Serial.println();

if(OneWire::crc8(scratchpad, 8) == scratchpad[8])
{
    int16_t temperature = makeWord(scratchpad[0],scratchpad[1]);
    int16_t moisture = makeWord(scratchpad[2],scratchpad[3]);
    int16_t conductivity = makeWord(scratchpad[4],scratchpad[5]);
    Serial.print("TENP=4.2f, MOISTURE=4.2f, CONDUCTIVITY=%4.2K, CONFIG0=%4.2K, CONFIG1=%4.2K, CRC8=%4.2X\n", temperature/100.0, moisture/100.0, conductivity, scratchpad[6],scratchpad[7],scratchpad[8] );
}
else
{
    Serial.print("CRC ERROR!\n");
}

Serial.println();
delay(1000);

```

Error downloading https://github.com/espressif/arduino-esp32/releases/download/1.0.4/package_esp32_index.json

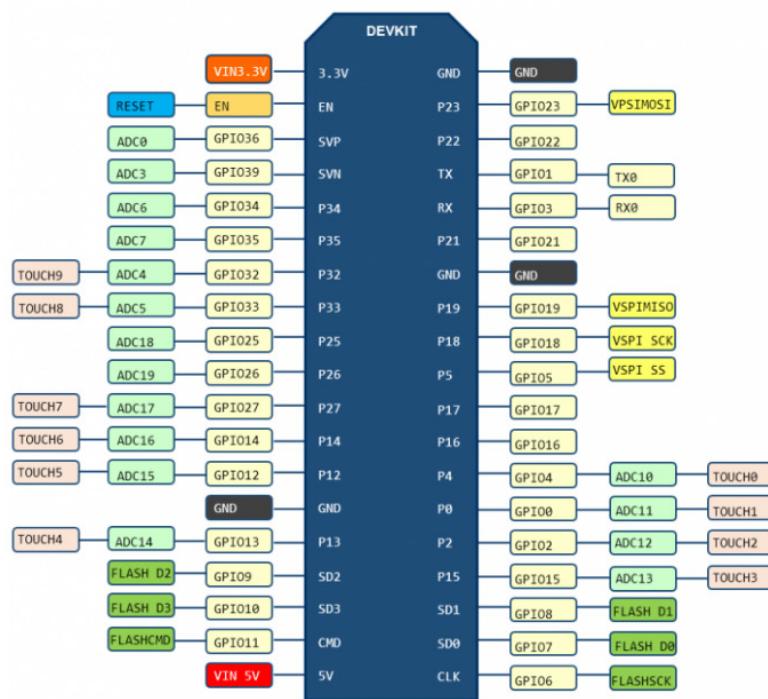
5 Demo on NodeMCU-32S

5.1 Preparation

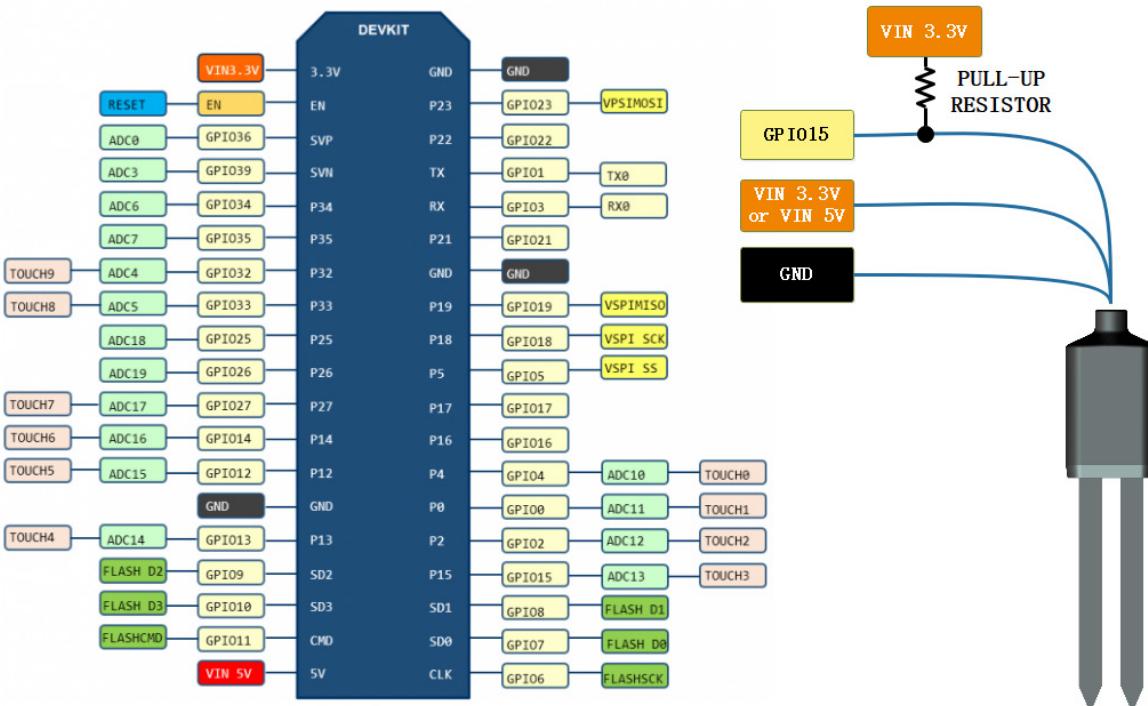
In this demo we use GPIO15 as 1-Wire signal I/O, Please pull up this IO by a resistor range from 2.0~5.1 Kohm to VIN3.3V.

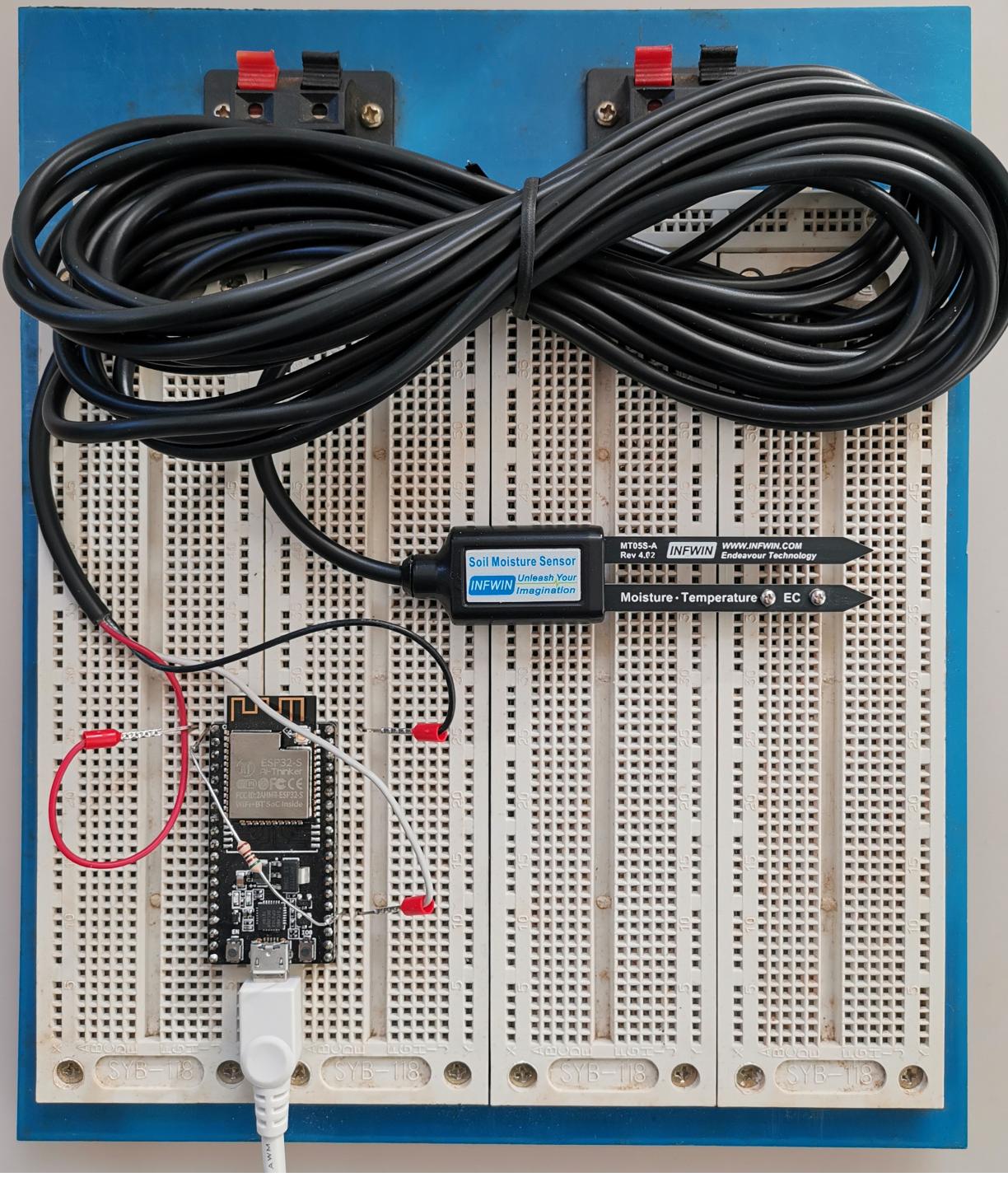
5.2 Wiring and Connection

Pin Layout – NodeMCU-32S



Wiring Diagram – NodeMCU-32S





5.3 Platform IO

- Download Platform IO project at : <https://github.com/INFWIN/mt05s-demo>
- Open Platform IO project in sourcecode folder: MT05S_PlatformIO_NodeMCU_32S
 - (1) Select proper PORT, Board(NodeMCU-32S) in SideBar Projects>Configure.
 - (2) Build

- (3) Upload
- (4) Monitor

```

main.cpp - MT05S_ParallelIO_NodeMCU_32S - Visual Studio Code [Administrator]

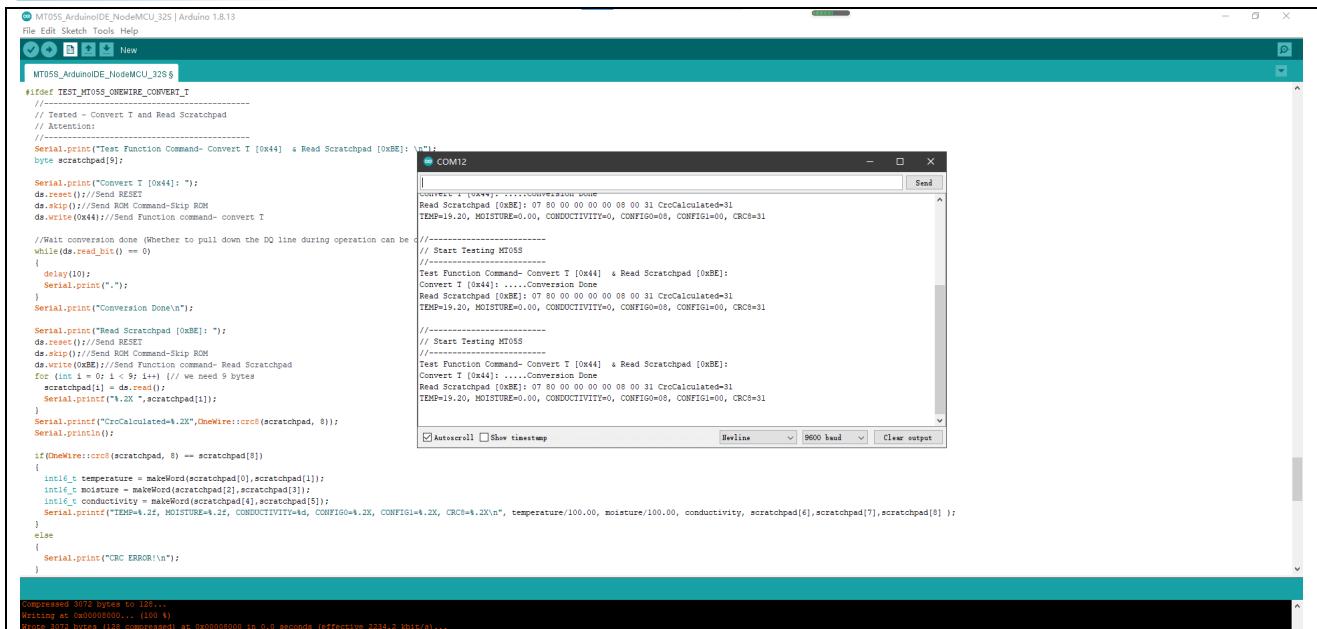
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
  main.cpp
  PIO Home
  MT05S_ParallelIO_NodeMCU_32S
    .vscode
      include
      lib
      src
        main.cpp
        test
        .gitignore
        .travis.yml
        platformio.ini

  PIO Home
  main.cpp (1.0)
  loop(c)
  #ifdef TEST_MT05S_ONEWIRE_CONVERT_T
  // -----
  // Tested - Convert T and Read Scratchpad
  // -----
  // Attention:
  // -----
  Serial.print("Test Function Command- Convert T [0x44] & Read Scratchpad [0xE8]: \n");
  byte scratchpad[9];
  Serial.print("Convert T [0x44]: ");
  ds.reset(); //Send RESET
  ds.skipROM(); //Send ROM Command-Skip ROM
  ds.write(0x44); //Send Function command- convert T
  //Wait conversion done (Whether to pull down the DQ line during operation can be configured in scratchpad CONFIG0-Bit3- SensorPowerMode)
  while(ds.read_bit() == 0)
  {
    delay(10);
    Serial.print(".");
  }
  Serial.print("Conversion Done\n");
  Serial.print("Read Scratchpad [0xE8]: ");
  // -----
  // Start Testing MT05S
  // -----
  Test Function Command- Convert T [0x44] & Read Scratchpad [0xE8]:
  Convert T [0x44]: .....Conversion Done
  Read Scratchpad [0xE8]: 07 88 00 00 00 08 00 31 CrcCalculated=31
  TEMP=-19.20, MOISTURE=0.00, CONDUCTIVITY=0, CONFIG0=00, CRC3=31
  // -----
  // Start Testing MT05B
  // -----
  Test Function Command- Convert T [0x44] & Read Scratchpad [0xE8]:
  Convert T [0x44]: .....Conversion Done
  Read Scratchpad [0xE8]: 07 88 00 00 00 08 00 31 CrcCalculated=31
  TEMP=-19.20, MOISTURE=0.00, CONDUCTIVITY=0, CONFIG0=00, CRC3=31
  // -----
  // Start Testing MT05S
  // -----
  Test Function Command- Convert T [0x44] & Read Scratchpad [0xE8]:
  Convert T [0x44]: .....Conversion Done
  Read Scratchpad [0xE8]: 07 88 00 00 00 08 00 31 CrcCalculated=31
  TEMP=-19.20, MOISTURE=0.00, CONDUCTIVITY=0, CONFIG0=00, CRC3=31
  
```

5.4 Arduino IDE

- Download Arduino IDE project: <https://github.com/INFWIN/mt05s-demo>
- Open Arduino IDE project in sourcecode folder: MT05S_ArduinoIDE_NodeMCU_32S
 - (1) Select proper PORT: Menu>Tools>Port
 - (2) Select Board: Menu>Tools>Board>Arduino ESP32>NodeMCU-32S
 - (3) Compile: Menu>Sketch>Verify/Compile
 - (4) Upload: Menu>Sketch>Upload
 - (5) Monitor: Menu>Tools>Serial Monitor



```

MT05S_ArduinoIDE_NodeMCU_32S | Arduino 1.8.13
File Edit Sketch Tools Help
MT05S_ArduinoIDE_NodeMCU_32S.h
#ifndef TEST_MT05S_1WIRE2_CONVERT_I_
#define TEST_MT05S_1WIRE2_CONVERT_I_
//-----//
// Tested - Convert T and Read Scratchpad
// Attention:
//-----//
Serial.print("Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE]: \n");
byte scratchpad[9];
Serial.print("Convert T [0x44]: ");
ds.reset(); //Send RESET
ds.skip(); //Send ROM Command-Skip ROM
ds.write(0x44); //Send Function command- convert T
//Wait conversation done (Whether to pull down the DQ line during operation can be
while(ds.read_bit() == 0)
{
    delay(10);
    Serial.print(".");
}
Serial.print("Conversion Done\n");
Serial.print("Read Scratchpad [0xBE]: ");
ds.reset(); //Send RESET
ds.skip(); //Send ROM Command-Skip ROM
ds.write(0xBE); //Send Function command- Read Scratchpad
for (int i = 0; i < 9; i++) // we need 9 bytes
{
    scratchpad[i] = ds.read();
    Serial.print("0x", hex, i);
}
Serial.print("Scratchpad[0] = ", scratchpad[0]);
Serial.print("Scratchpad[1] = ", scratchpad[1]);
Serial.print("CrcCalculated=4..2X,OneWire::crcf(scratchpad, 8);");
Serial.println();
if(OneWire::crcf(scratchpad, 8) == scratchpad[8])
{
    int16_t temperature = makeWord(scratchpad[0],scratchpad[1]);
    int16_t moisture = makeWord(scratchpad[2],scratchpad[3]);
    int16_t conductivity = makeWord(scratchpad[4],scratchpad[5]);
    Serial.print("TENV=4.2F, MOISTURE=4.2F, CONDUCTIVITY=4.2K, CONFIG1=4.2K, CRCE=4.2K\n");
    Serial.print("temperature/100.00, moisture/100.00, conductivity, scratchpad[6],scratchpad[7],scratchpad[8] );
}
else
{
    Serial.print("CRC ERROR!\n");
}
Compressed 3072 Bytes to 128...
Writing at 0x00000000... (100 %)
Write 3072 Bytes (125 compressed) at C:\d0\c0\00 in 0.8 seconds (effective 234.0 kbit/s)... .

```

6 Demo on RaspberryPI 3 Model B

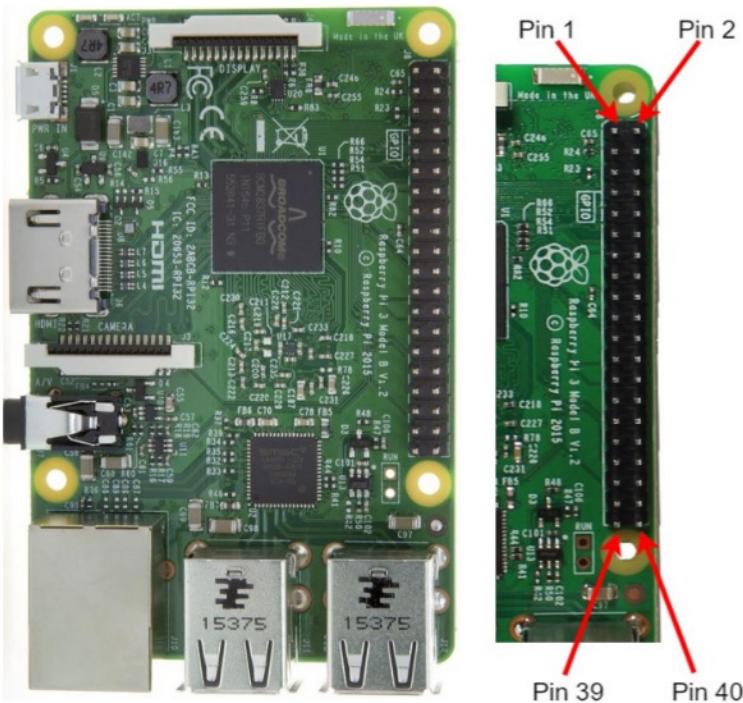
6.1 Preparation

In this demo we use GPIO4 as 1-Wire signal I/O , Please pull up this IO by a resistor range from 2.0~5.1 Kohm to 3.3V.

Note that in this demo two sensors are connected to the 1-Wire bus, “**28-000005a1cb5b**” is the DS18B20, “**28-060504030201**” is the MT05S.

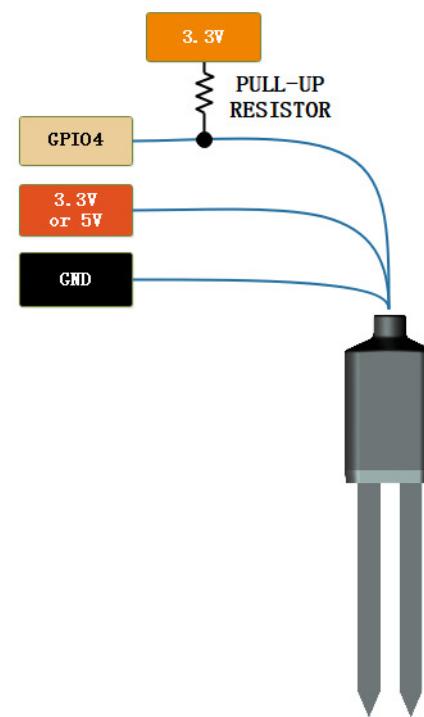
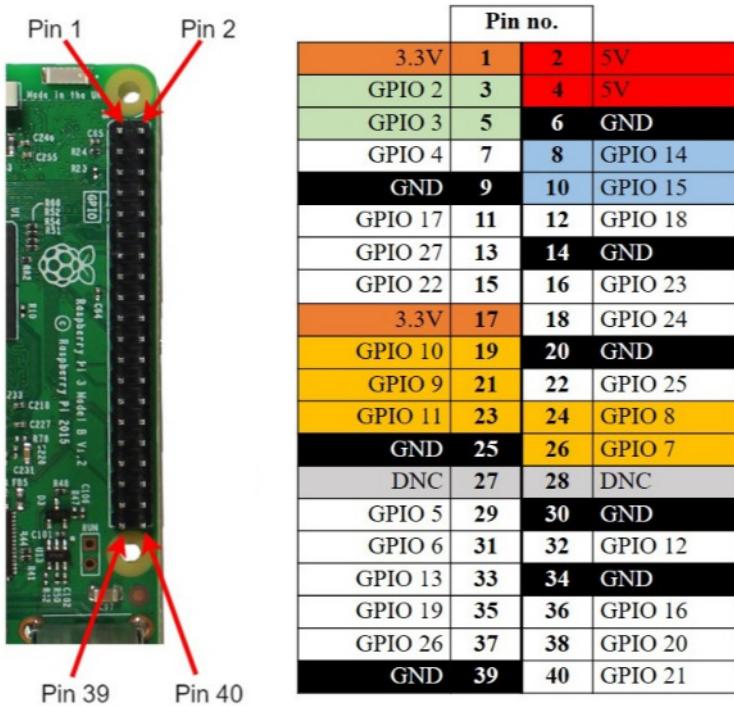
6.2 Wiring and Connection

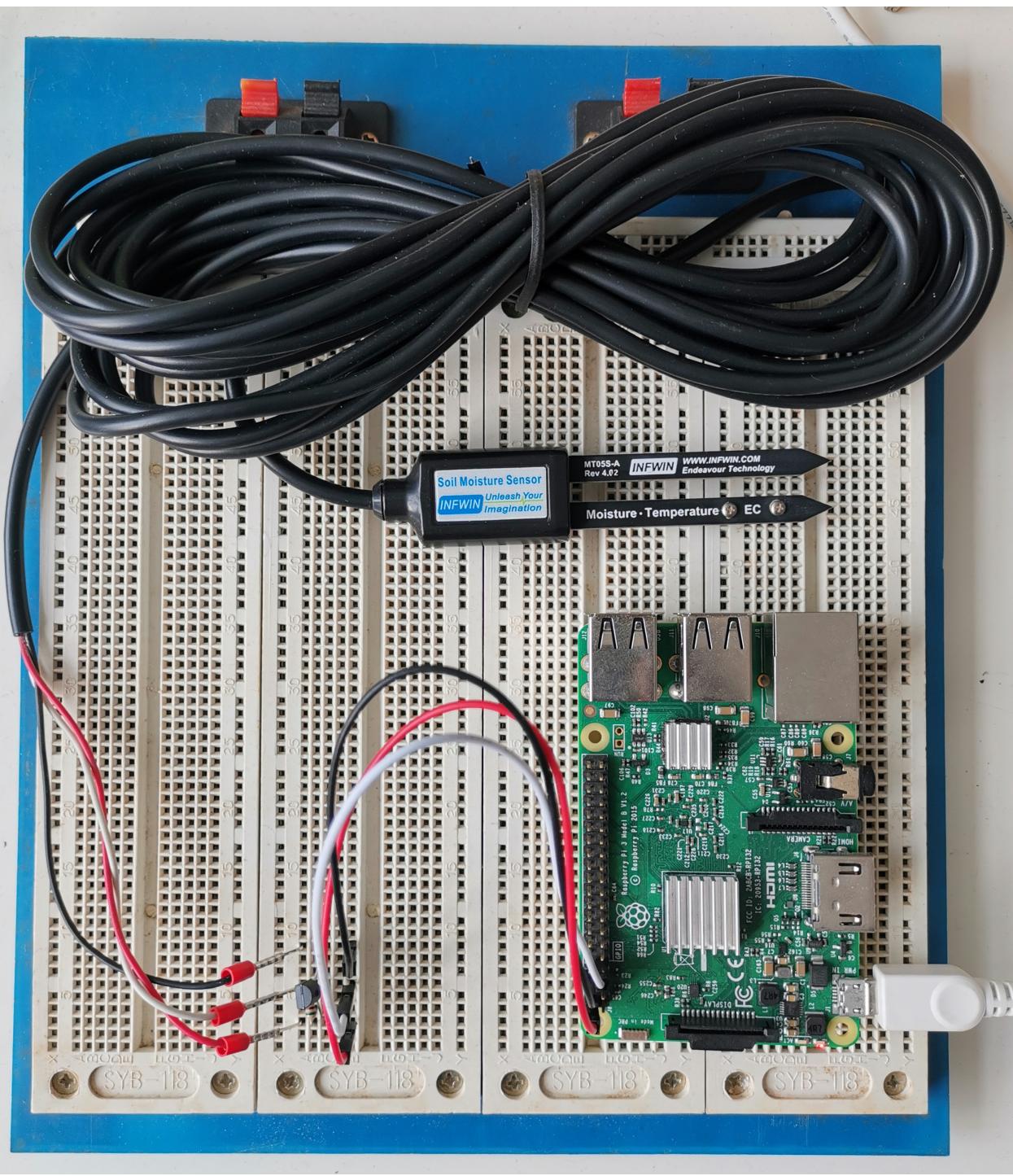
Pin Layout – Raspberry PI Model 3B



	Pin no.		
3.3V	1	2	5V
GPIO 2	3	4	5V
GPIO 3	5	6	GND
GPIO 4	7	8	GPIO 14
GND	9	10	GPIO 15
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10	19	20	GND
GPIO 9	21	22	GPIO 25
GPIO 11	23	24	GPIO 8
GND	25	26	GPIO 7
DNC	27	28	DNC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21

Wiring Diagram – Raspberry PI Model 3B



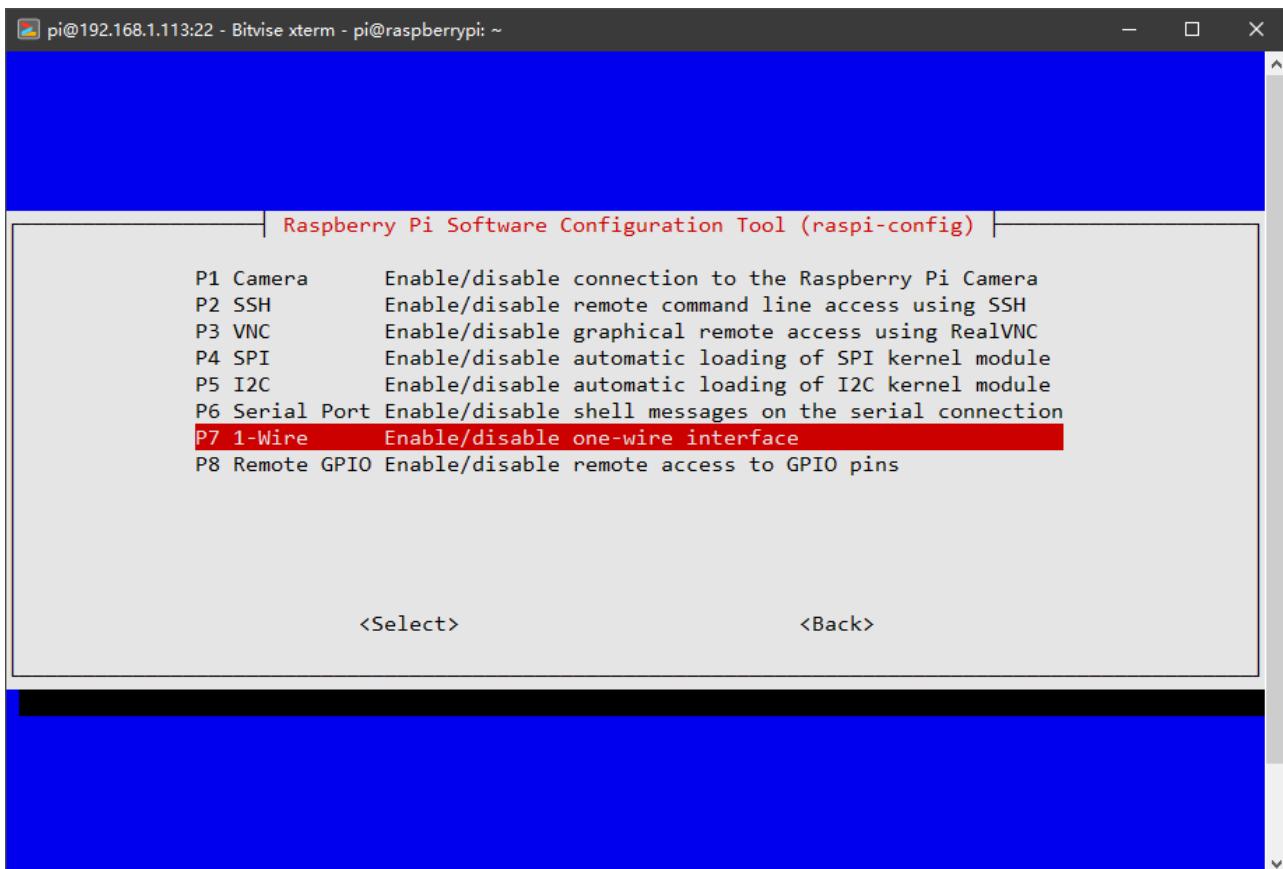
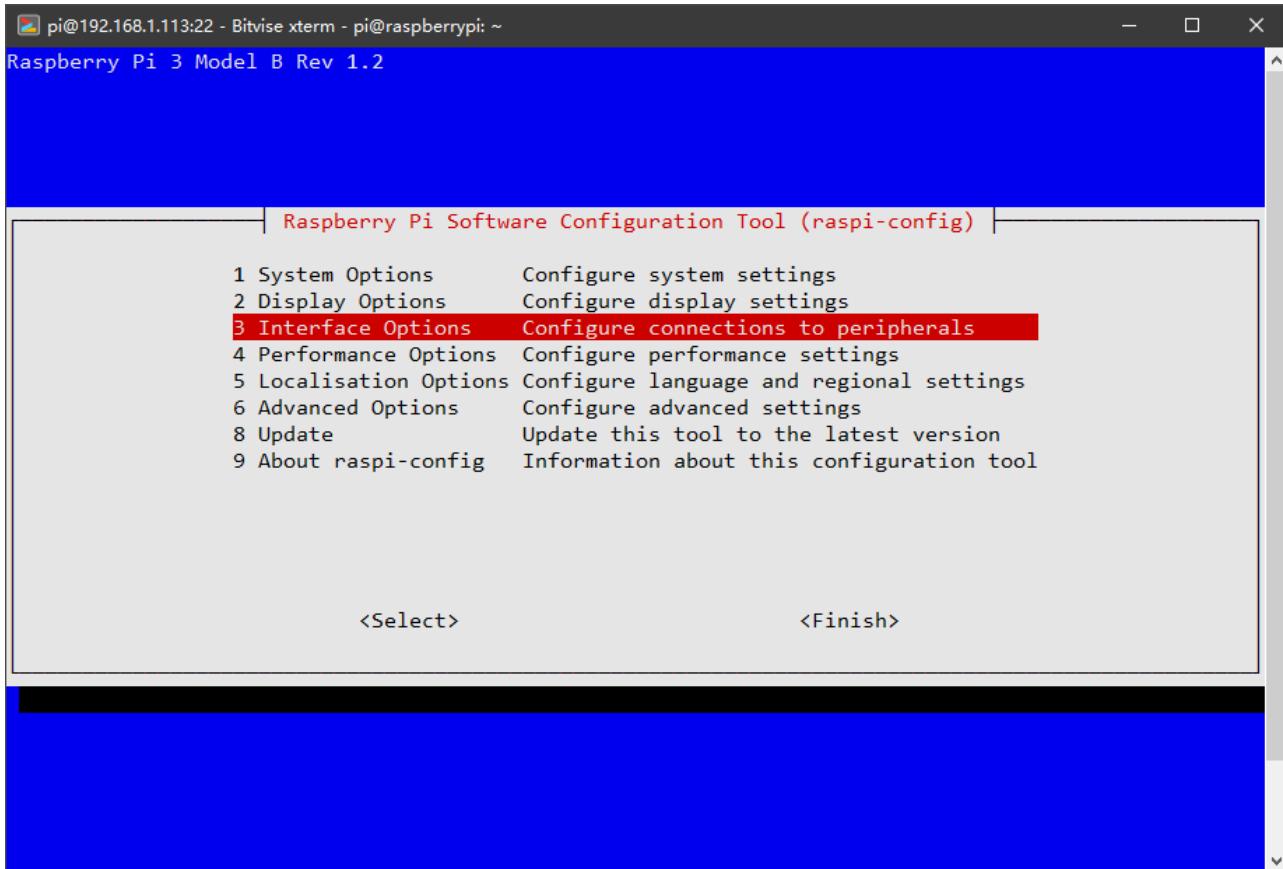


6.3 Program

- Download Arduino IDE project: <https://github.com/INFWIN/mt05s-demo>
- Open sourcecode folder: MT05S_RaspberryPi_3B
- Login to Raspberry PI
- Enable 1-Wire interface and check sensor:

```
pi@raspberrypi:~ $ sudo raspi-config
```

>> Bring up the configuration Tool and then enter into “Interface Options”, and then select “1-Wire” to enable it.



- Check 1-Wire devices:

```
pi@raspberrypi:~ $ cd /sys/bus/w1/devices/  
pi@raspberrypi:/sys/bus/w1/devices $ ls  
28-000005a1cb5b 28-060504030201 w1_bus_master1
```

Note that in this demo there are two sensors found, “28-000005a1cb5b” is the DS18B20, “28-060504030201” is the MT05S

- Check 1-Wire device data:

```
pi@raspberrypi:/sys/bus/w1/devices $ cd 28-060504030201  
pi@raspberrypi:/sys/bus/w1/devices/28-060504030201 $ ls  
alarms driver ext_power hwmon name resolution temperature w1_slave  
conv_time eeprom features id power subsystem uevent  
pi@raspberrypi:/sys/bus/w1/devices/28-060504030201 $ cat w1_slave  
07 9e 00 00 00 00 08 00 b5 : crc=b5 YES  
07 9e 00 00 00 00 08 00 b5 t=-1567562  
pi@raspberrypi:/sys/bus/w1/devices/28-060504030201 $
```

- Test the python demo code : MT05S_RaspberryPi_3B.py

```
pi@raspberrypi:~ $ python MT05S_RaspberryPi_3B.py
```

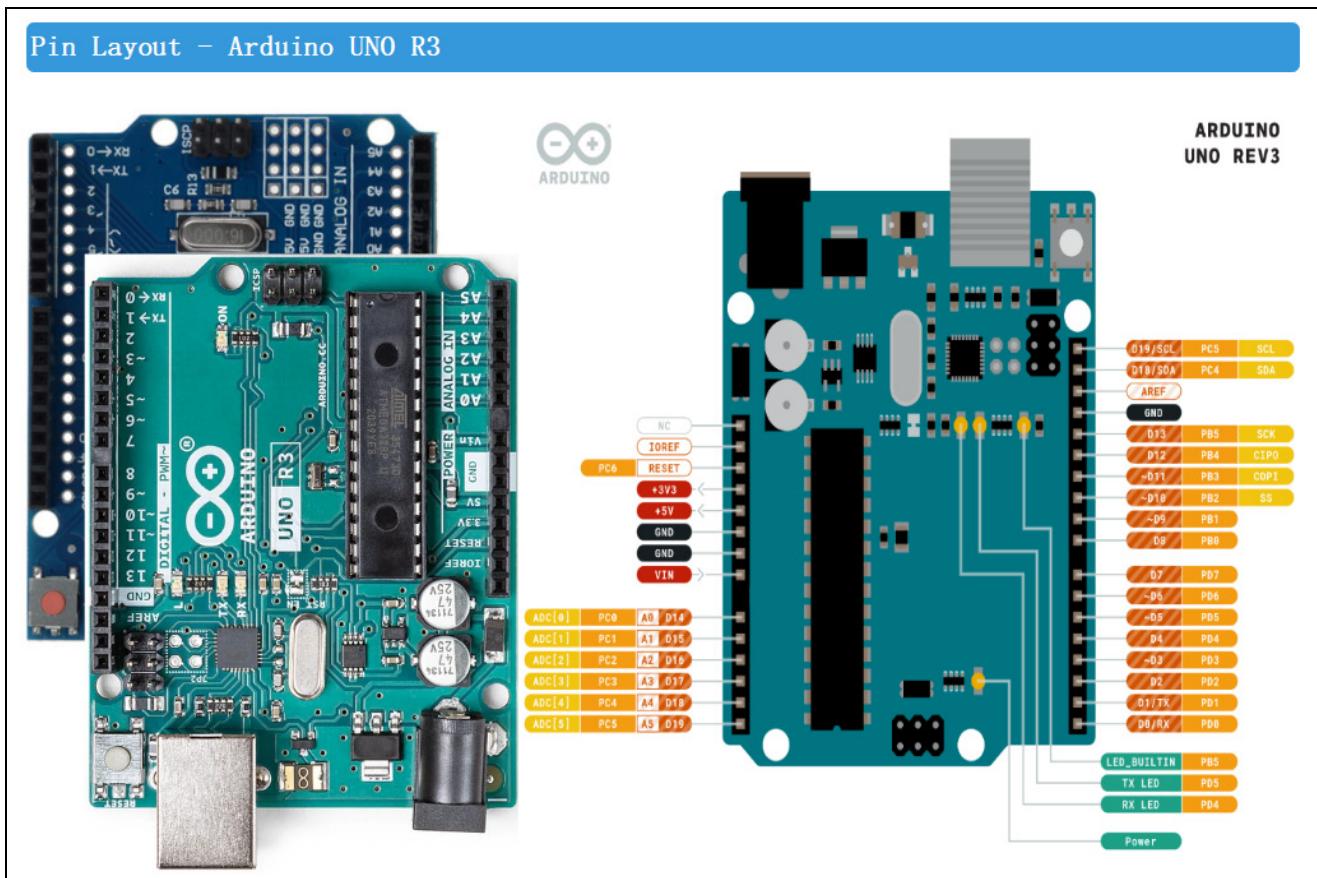
```
pi@192.168.1.113:22 - Bitvise xterm - pi@raspberrypi: ~  
pi@raspberrypi:~ $ python MT05S_RaspberryPi_3B.py  
//-----  
// Start Testing MT05S  
// Rom Code= 28-060504030201  
// Change ROM CODE FOR YOUR SENSOR !!!!!!  
//-----  
Soil Temp(C)= 19.50  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 1.80  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 13.20  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 18.30  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 19.60  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 17.90  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 17.10  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 19.90  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 15.60  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.50  Moisture(%)= 19.60  EC(ms/cm)= 0.010  
Soil Temp(C)= 19.50  Moisture(%)= 21.10  EC(ms/cm)= 0.020  
Soil Temp(C)= 19.50  Moisture(%)= 18.80  EC(ms/cm)= 0.020  
Soil Temp(C)= 19.50  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.60  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
Soil Temp(C)= 19.60  Moisture(%)= 0.00  EC(ms/cm)= 0.000  
^CTraceback (most recent call last):  
  File "MT05S_RaspberryPi_3B.py", line 42, in <module>  
    time.sleep(1)  
KeyboardInterrupt  
pi@raspberrypi:~ $
```

7 Demo on Arduino UNO R3

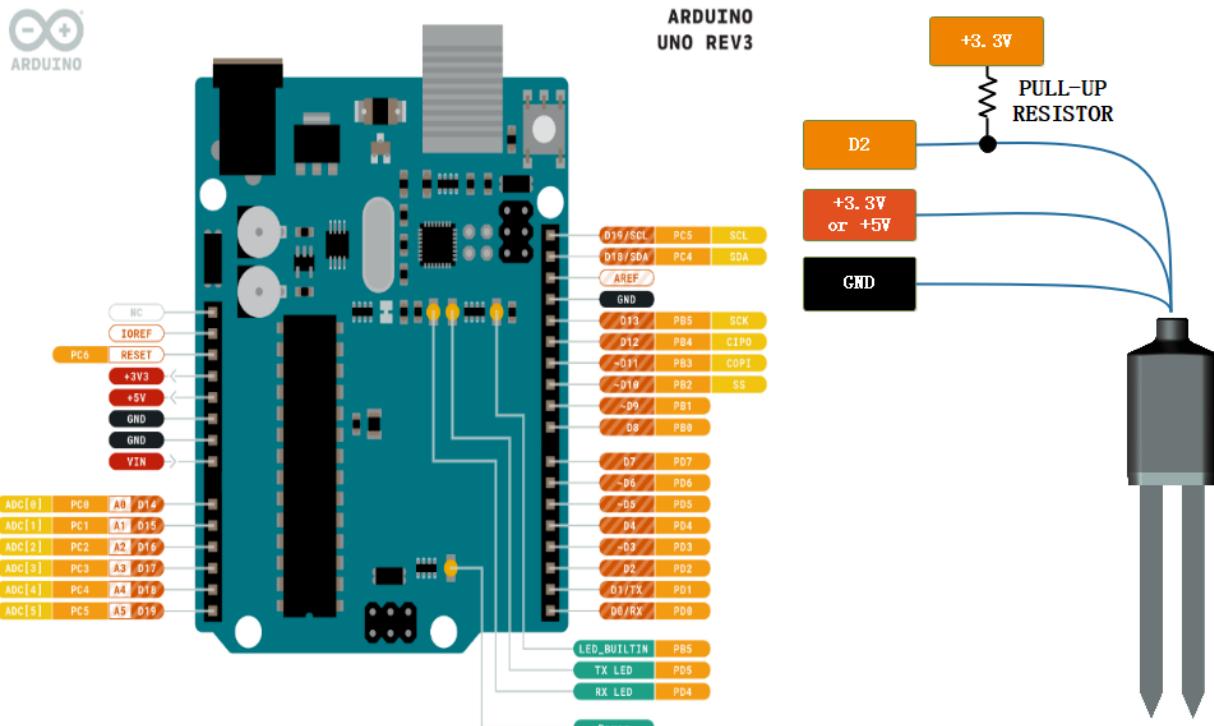
7.1 Preparation

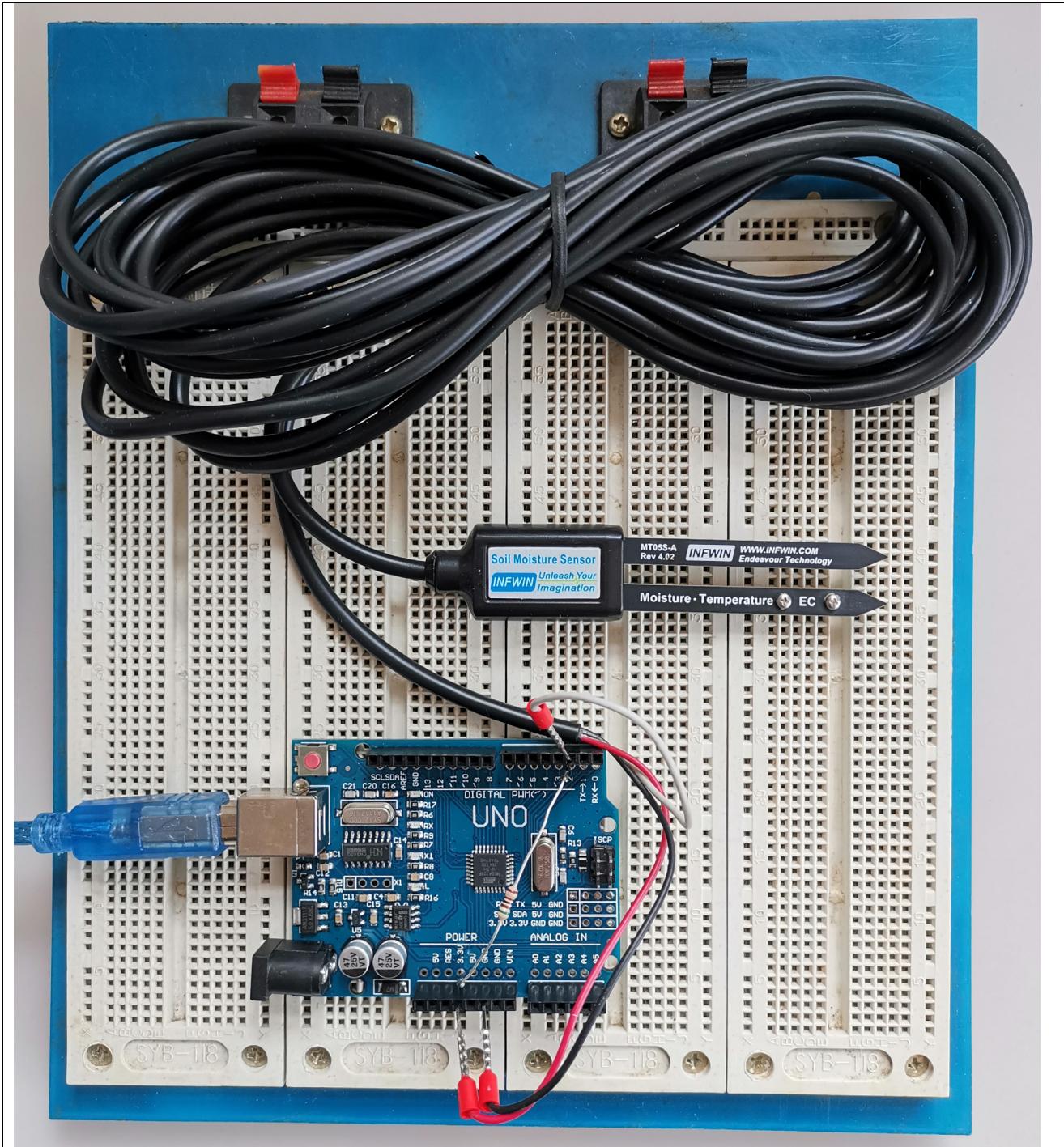
In this demo we use D2 as 1-Wire signal I/O, Please pull up this IO by a resistor range from 2.0~5.1 Kohm to +3.3V.

7.2 Wiring and Connection



Wiring Diagram – Arduino UNO R3





7.3 Platform IO

- Download Platform IO project at : <https://github.com/INFWIN/mt05s-demo>
- Open Platform IO project in sourcecode folder: MT05S_PlatformIO_ArduinoUnoR3
 - (1) Select proper PORT, Board(Arduino UNO) in SideBar Projects>Configure.
 - (2) Build
 - (3) Upload
 - (4) Monitor

```

main.cpp - MT05S_PlatformIO_ArduinoUnoR3 - Visual Studio Code [Administrator]

File Edit Selection View Go Run Terminal Help

EXPLORER OPEN EDITORS PIO Home main.cpp 2_U
src > main.cpp > ...
1 #include <OneWire.h>
2
3 #define TEST_MT05S_ONEWIRE_CONVERT_T 1
4
5 // Define a 1-Wire signal pin (a 2.0K-5.1K pull-up resistor is necessary)
6 // Change according to your wiring and connection.
7 OneWire ds(2);
8
9 void setup(void) {
10   Serial.begin(9600);
11 }
12
13 void loop(void) {
14   Serial.println("-----");
15   Serial.println(" Start Testing MT05S -----");
16   Serial.println("-----");
17   uint8_t i;
18
19 #ifdef TEST_MT05S_ONEWIRE_CONVERT_T
20   //-----
21   // Tested - Convert T and Read Scratchpad
22   // Attention:
23
TEMP(C) = 18.18 MOISTURE(%) = 0.00 CONDUCTIVITY(ms/cm) = 0.00 CONFIG0 = 8 CONFIG1 = 0 CRC8 = 27
//-----
// Start Testing MT05S
//-----
Test Function Command- Convert T [0x44] & Read Scratchpad [0xdE];
Convert T [0x44]: ....Conversion Done
Read Scratchpad [0x8E]: 7 12 0 0 0 0 8 0 27 CrcCalculated=27
TEMP(C) = 18.18 MOISTURE(%) = 0.00 CONDUCTIVITY(ms/cm) = 0.00 CONFIG0 = 8 CONFIG1 = 0 CRC8 = 27

//-----
// Start Testing MT05S
//-----
Test Function Command- Convert T [0x44] & Read Scratchpad [0xdE];
Convert T [0x44]: ....Conversion Done
Read Scratchpad [0x8E]: 7 12 0 0 0 0 8 0 27 CrcCalculated=27
TEMP(C) = 18.18 MOISTURE(%) = 0.00 CONDUCTIVITY(ms/cm) = 0.00 CONFIG0 = 8 CONFIG1 = 0 CRC8 = 27

//-----
// Start Testing MT05S
//-----
Test Function Command- Convert T [0x44] & Read Scratchpad [0xdE];
Convert T [0x44]: ....Conversion Done
Read Scratchpad [0x8E]: 7 1C 0 0 0 0 8 0 58 CrcCalculated=58
TEMP(C) = 18.20 MOISTURE(%) = 0.00 CONDUCTIVITY(ms/cm) = 0.00 CONFIG0 = 8 CONFIG1 = 0 CRC8 = 58

```

7.4 Arduino IDE

- Download Arduino IDE project: <https://github.com/INFWIN/mt05s-demo>
- Open Arduino IDE project in sourcecode folder: MT05S_ArduinoIDE_ArduinoUnoR3
 - (1) Select proper PORT: Menu>Tools>Port
 - (2) Select Board: Menu>Tools>Board>Arduino AVR Boards>Arduino UNO
 - (3) Compile: Menu>Sketch>Verify/Compile
 - (4) Upload: Menu>Sketch>Upload
 - (5) Monitor: Menu>Tools>Serial Monitor

```

MT655_ArduinoIDE_NodeMCU_32S | Arduino 1.8.13
File Edit Sketch Tools Help
MT655_ArduinoIDE_NodeMCU_32S
#include <OneWire.h>
#define TEST_MT655_ONewire_CONVERT_T 1
// Define a 1-Wire signal pin (a 2.0K-5.1K pull-up resistor is necessary)
// Change according to your wiring and connection.
OneWire ds(2);
void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  Serial.println("//-----");
  Serial.println("// Start Testing MT655");
  Serial.println("//-----");
  // Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE];
  // Convert T [0x44]; ....Conversion Done
  // Read Scratchpad [0xBE]: 4 F4 3 34 0 A 8 0 FD CrcCalculated=FD
  TEMP(C) = 17.80 MOISTURE() = 8.20 CONDUCTIVITY(ms/cm) = 0.01 CONFIG0 = 8 CONFIG1 = 0 CRC8 = FD
  Serial.println("//-----");
  // Start Testing MT655
  //-----;
  // Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE];
  // Convert T [0x44]; ....Conversion Done
  // Read Scratchpad [0xBE]: 4 F4 3 48 0 A 8 0 B1 CrcCalculated=B1
  TEMP(C) = 17.80 MOISTURE() = 8.40 CONDUCTIVITY(ms/cm) = 0.01 CONFIG0 = 8 CONFIG1 = 0 CRC8 = B1
  Serial.println("//-----");
  // Start Testing MT655
  //-----;
  // Test Function Command- Convert T [0x44] & Read Scratchpad [0xBE];
  // Convert T [0x44]; ....Conversion Done
  // Read Scratchpad [0xBE]: 4 F4 3 34 0 A 8 0 FD CrcCalculated=FD
  TEMP(C) = 17.80 MOISTURE() = 8.20 CONDUCTIVITY(ms/cm) = 0.01 CONFIG0 = 8 CONFIG1 = 0 CRC8 = FD
  Serial.println("//-----");
  // Wait conversion done (Whether to pull down the DQ line during operation can be configured)
  while(ds.read_bit() == 0)
  {
    delay(10);
    Serial.print(".");
  }
  Serial.print("Conversion Done\n");

  Serial.print("Read Scratchpad [0xBE]: ");
  ds.reset(); //Send RESET
  ds.skip(); //Send ROM Command-Skip ROM
  ds.write(0x47); //Send Function command- Read Scratchpad
  for (int i = 0; i < 9; i++) // we need 9 bytes
    scratchpad[i] = ds.read();
  for (int i = 0; i < 9; i++) // we need 9 bytes

Serial.download("https://github.com/esp8266/Arduino/releases/download/1.0.6/package_esp32_index.json"
Sketch uses 5010 bytes (1%) of program storage space. Maximum is 32256 bytes
Global variables use 520 bytes (2%) of dynamic memory, leaving 1528 bytes for local variables. Maximum is 2048 bytes.

```

Copyright and Trademark

This document is copyrighted, 2021, by Dalian Endeavour Technology Co., Ltd. All rights are reserved. Dalian Endeavour Technology Co., Ltd. Reserves the right to make improvements to the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission. Information provided in this manual is intended to be accurate and reliable. However, Dalian Endeavour Technology Co., Ltd. assumes no responsibility for its use, nor for any infringements upon the rights of third parties, which may result from its use.

INFWIN® is the trademark of Dalian Endeavour Technology Co., Ltd.

1-Wire® is a trademark of MAXIM.

Revision

Date	Version	Comment	Updated by
2021-10-25	V1.0	Initial Creation	jz51930