

Sunfounder Smart Home Kit for Arduino

SUNFOUNDER

Content

Sunfounder Smart Home Kit for Arduino	0
Content	1
Kit Overview.....	2
A Brief Introduction to DeviceBit Platform	3
Hardware Description	4
Photoresistor.....	4
Temperature & Humidity Sensor DHT11	4
Gas Sensor MQ-2	5
Human Body Pyroelectric Infrared Sensor (PIR)	5
RFID-RC522.....	6
Gas Pressure sensor BMP180	7
RF Transceiver Chip NRF24L01	7
W5100.....	8
Discrete Experiment.....	9
Experiment 1 How to Use Photoresistor.....	9
Experiment 2 How to Use Temperature & Humidity Sensor DHT11	10
Experiment 3 How to Use MQ-2 Gas Sensor	12
Experiment 4 How to Use PIR	14
Experiment 5 How to Use Gas Pressure Sensor BMP180	17
Experiment 6 How to Use RFID Module.....	18
Experiment 7 How to Use NRF24L01	21
DeviceBit Platform and Control Board Joint Test.....	24
How to Operate DeviceBit Platform.....	24
Register an Account on DeviceBit Platform.....	24
Add a New Device.....	25
Add a New Sensor or Controller	25
How to Operate Control Board.....	27
Acquisition board.....	28
Control Board.....	31
How to Operate Web app on Mobile Phone	38
Summary	41

Kit Overview

With this kit tutorial, you will learn how to DIY a simple Smart Home system. In this Smart Home system, we constantly upload temperature and humidity, light intensity, gas values, gas pressure, elevation, RFID status values, human pyroelectric infrared sensor status values and other data to DeviceBit platform at a certain time interval. With the controller options on DeviceBit platform, you will be able to control your remote relay in real time.

SUNFOUNDER

A Brief Introduction to DeviceBit Platform

The DeviceBit Platform (<http://www.devicebit.com/home/publicsensors>) is a real-time data brokerage platform for the Internet of Things (IOT), providing most of its functionality via its Application Programming Interface (API). It is quick and easy to add Devices and Applications to the DeviceBit platform. It provides real-time data storage and remote control at scale. The DeviceBit platform is not just an easy way to prototype new Internet-enabled Sensors; it's also a service that helps companies bring products to market at scale.

The DeviceBit platform provides basic data analysis tools for rapid data evaluation, as well as real-time alerts and notifications if Sensors report "abnormal" conditions.

You can realize your own ideas and develop your own devices relying on this platform. You can focus on hardware instead of software infrastructure.

The DeviceBit platform also communicates with existing social network, such as Twitter and Facebook, allowing you to share what you do with your friend, which might be very helpful to their research on similar field.

Hardware Description

Photoresistor

A photoresistor (see Figure 3.1) gets its resistance from the light. If it is dark, the sensor will read up in the millions of Ohms, but as light is shined at it, it will have a lower resistance. The sensor requires a voltage divider to get a correct signal.



Fig3.1. Photoresistor

Temperature & Humidity Sensor DHT11

The DHT11 sensor (see Figure 3.2) can detect temperature (C and F) & humidity. It has everything it requires built into it, so it will work very well with the SUNFOUNDER UNO. This sensor is used in conjunction with the DHT11 Library.

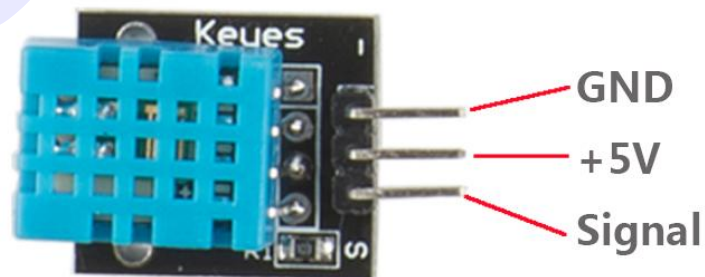


Fig3.2. DHT11

Gas Sensor MQ-2

MQ-2 gas sensor (see Figure 3.3) belongs to surface ion N type semiconductor, which is made of tin oxide semiconductor gas sensitive material. With the gas concentration rising, the sensor's conductivity increases and its output resistance decrease. MQ-2 gas sensor has high sensitivity to natural gas, LPG and other smoke. You can read out gas concentration value with SUNFOUNDER MEGA2560 built-in ADC module according to the principle of voltage division.



Fig3.3. MQ-2

Human Body Pyroelectric Infrared Sensor (PIR)

The PIR (see Figure 3.4) will detect changes in infrared radiation; thus, we can use it to detect motion, because the human body does not have a constant temperature. The PIR cannot sense movement unless there is this change in temperature. To use a PIR sensor, connect the signal of the PIR to a digital pin on the SUNFOUNDER MEGA2560. When the sensor detects people pass by, pin OUT will output 1, otherwise it will output 0.



Fig3.4. PIR

RFID-RC522

RFID (see Figure 3.5) is the abbreviation of radio frequency identification, called radio frequency technology. We use RC522 as the chip of RFID module in this kit. RC522 is a highly integrated contactless R/W chip (13.56 MHz). The delivery module is fully integrated into the contactless communication methods and protocols (13.56 MHz) by using the principle of modulation and demodulation. In this kit tutorial, just call RFID library files to easily use RFID module.



Fig3.5. RFID

Gas Pressure sensor BMP180

The BMP180 (see Figure 3.6) is designed to be connected directly to a microcontroller of a mobile device via the I2C bus. The pressure and temperature data has to be compensated by the calibration data of the E2PROM of the BMP180.

The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with E2PROM and a serial I2C interface. In this tutorial, we use Adafruit_BMP085_Unified and adafruit_sensor_maste libraries to read the value of BMP180.



Fig3.6. BMP180

RF Transceiver Chip NRF24L01

The NRF24L01 (see Figure 3.7) is a single-chip RF transceiver chip and works in 2.4 ~ 2.5 GHz ISM frequency band. The chip includes a fully integrated frequency synthesizer, power amplifier, crystal oscillator and the modulator. Operating parameters, such as transmitting power and working frequency can be done easily by three-line SPI port. With ultra low current consumption, it is only 10.5mA when the output power is -5dBm and only 18mA in receiving mode. It can easily achieve low power requirements in power off mode.

The NRF24L01 has four operating modes: transceiver mode, configuration mode, idle mode and power off mode. The operating mode of NRF24L01 is determined by three pins of PWR_UP, CE, and TX_EN.

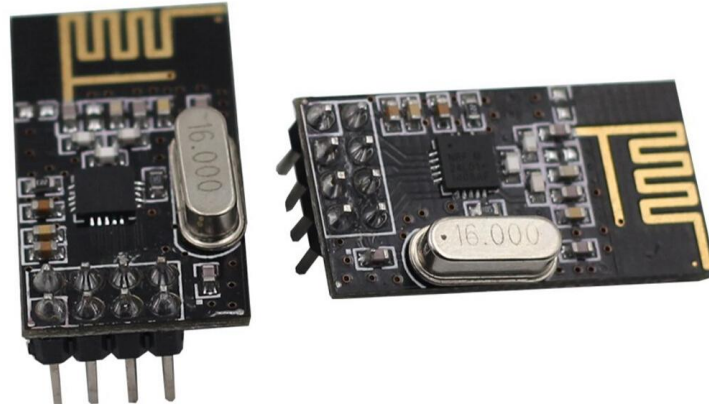


Fig3.7. NRF24L01

W5100

W5100 (see Figure 3.8) is a versatile single-chip network interface chip with 10/100 Ethernet controller integrated internally. It is mainly used in highly integrated, high stable, high performance and low cost embedded system. W5100 enables you to connect to Internet without operation system. W5100 is compatible with IEEE802.3 10BASE-T and 802.3u 100BASE-TX.

W5100 integrates whole hardware and years of market proven TCP/IP protocol stack inside, Ethernet medium transport layer (MAC) and physical layer (PHY). Hardware TCP/IP protocol stack supports the TCP, UDP, IPV4, ICMP, ARP, IGMP and PPoE. These protocols have been validated for years in many fields. In addition, W5100 also internally integrates 16KB memory for data transmission. For W5100, you don't need to consider the control of Ethernet, but only a simple port (Socket) programming.

W5100 has three interfaces, that is, direct parallel bus, indirect parallel bus and the SPI bus. W5100 is easy to connect with MCU, just like accessing external memory. Here we use Ethernet library to use W5100 more easily and conveniently.

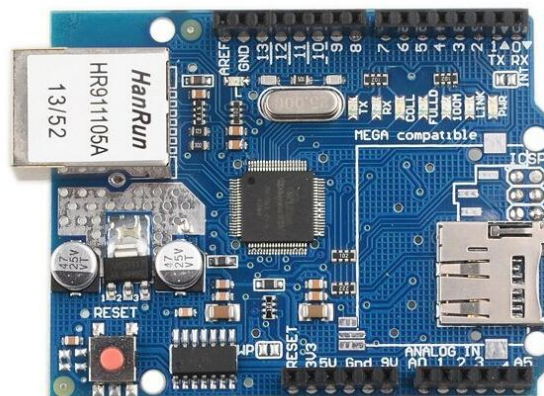


Fig3.8. W5100

Discrete Experiment

Experiment 1 How to Use Photoresistor

Overview:

In this experiment, we use the analog port A0 of SUNFOUNDER MEGA2560 to read the resistance of photoresistor after processing by the master control chip built-in ADC module, and display it on the serial monitor.

Components:

SUNFOUNDER MEGA2560 * 1

Photoresistor * 1

Resistor (10K Ohm) * 1

USB Cable * 1

Jumper wires

Steps:

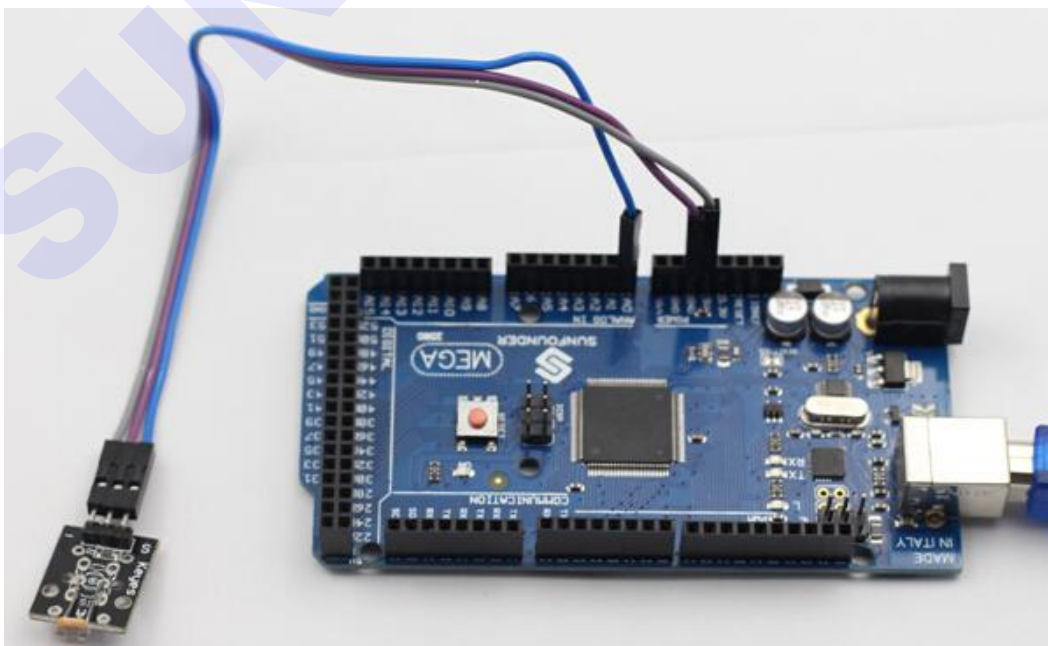
First, connect the circuit

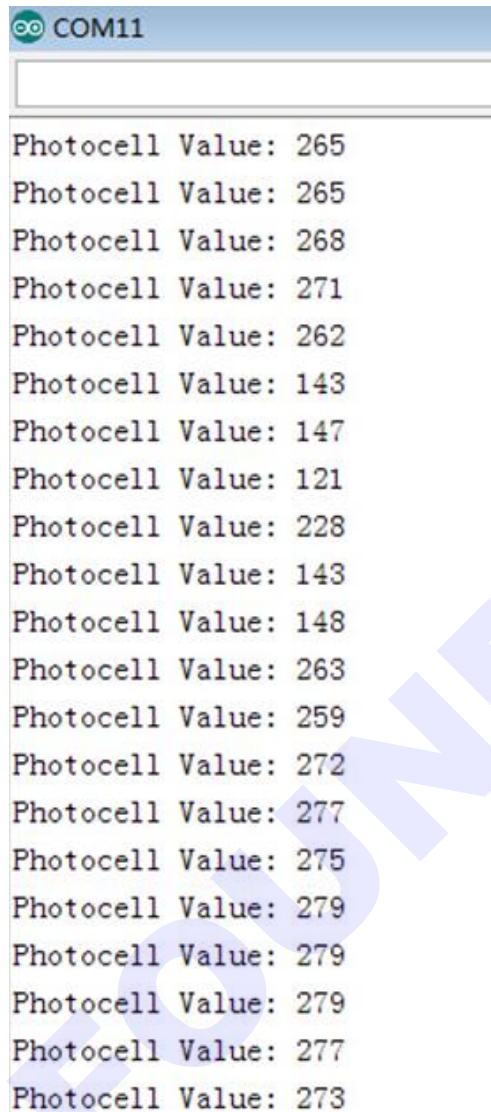
MEGA2560	Photoresistor
A0	S
5V	+
GND	" - "

Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board





```
COM11
Photocell Value: 265
Photocell Value: 265
Photocell Value: 268
Photocell Value: 271
Photocell Value: 262
Photocell Value: 143
Photocell Value: 147
Photocell Value: 121
Photocell Value: 228
Photocell Value: 143
Photocell Value: 148
Photocell Value: 263
Photocell Value: 259
Photocell Value: 272
Photocell Value: 277
Photocell Value: 275
Photocell Value: 279
Photocell Value: 279
Photocell Value: 279
Photocell Value: 277
Photocell Value: 273
```

Experiment 2 How to Use Temperature & Humidity Sensor DHT11

Overview:

In this experiment, we will learn how to use Temperature & Humidity Sensor DHT11. After you performed all the steps, you will see the temperature & humidity data collected by DHT11 display on the serial monitor.

Components:

SUNFOUNDER MEGA2560 * 1

DHT11 * 1

USB Cable * 1

Jumper wires

Steps:

First, connect the circuit

SUNFOUNDER MEGA2560

DHT11 Module

36 ----- S

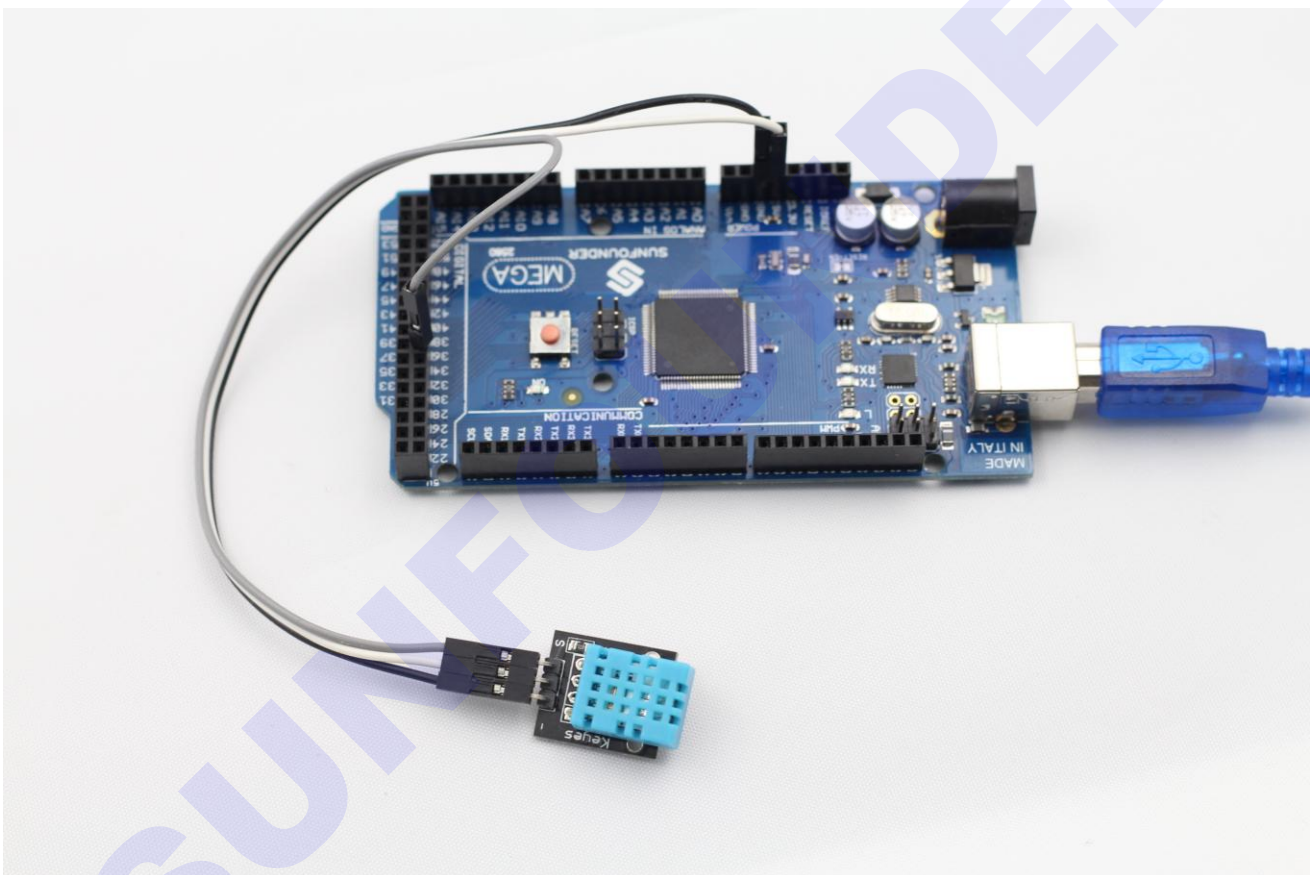
5V ----- +

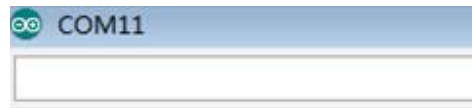
GND ----- "-"

Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board





```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

```
DHT11, OK,  
Tem: 30 C  
Hum: 37 %
```

Experiment 3 How to Use MQ-2 Gas Sensor

Overview:

In this experiment, we will learn how to use the MQ-2 gas sensor. With the value displaying on the serial monitor, we can intuitively know the concentration value of surrounding combustible gas. The higher the concentration of combustible gas is, the greater the value is; the lower the concentration is, the smaller the value is.

Components:

SUNFOUNDER MEGA2560 * 1

MQ-2 gas sensor * 1

USB Cable * 1

Jumper wires

Steps:

First, connect the circuit

SUNFOUNDER MEGA2560

MQ-2 Module

A1 ----- OUT

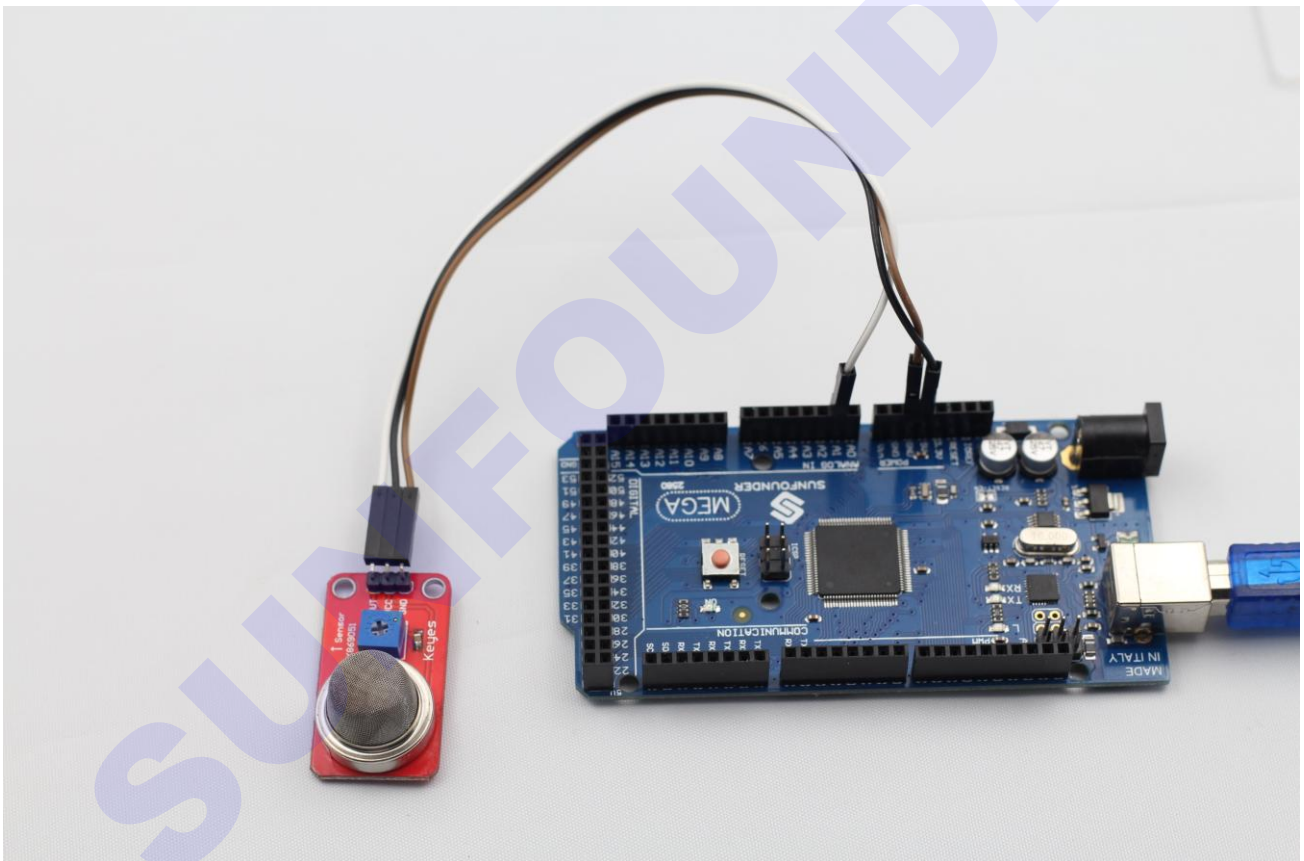
5V ----- VCC

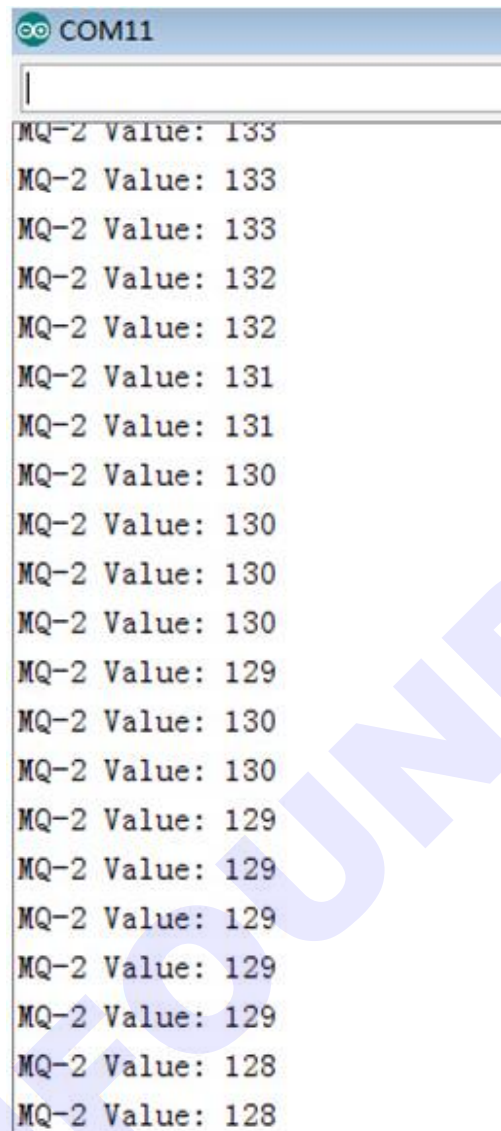
GND ----- GND

Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board





```
COM11
MQ-2 Value: 133
MQ-2 Value: 133
MQ-2 Value: 133
MQ-2 Value: 132
MQ-2 Value: 132
MQ-2 Value: 131
MQ-2 Value: 131
MQ-2 Value: 130
MQ-2 Value: 130
MQ-2 Value: 130
MQ-2 Value: 130
MQ-2 Value: 129
MQ-2 Value: 130
MQ-2 Value: 130
MQ-2 Value: 129
MQ-2 Value: 129
MQ-2 Value: 129
MQ-2 Value: 129
MQ-2 Value: 129
MQ-2 Value: 128
MQ-2 Value: 128
```

Experiment 4 How to Use PIR

Overview:

In this experiment, we will use an LED to indicate the state value of PIR. When someone goes by, the LED will light up; when no one goes by, the LED will go out.

Components:

SUNFOUNDER MEGA2560 * 1

PIR * 1

USB Cable * 1

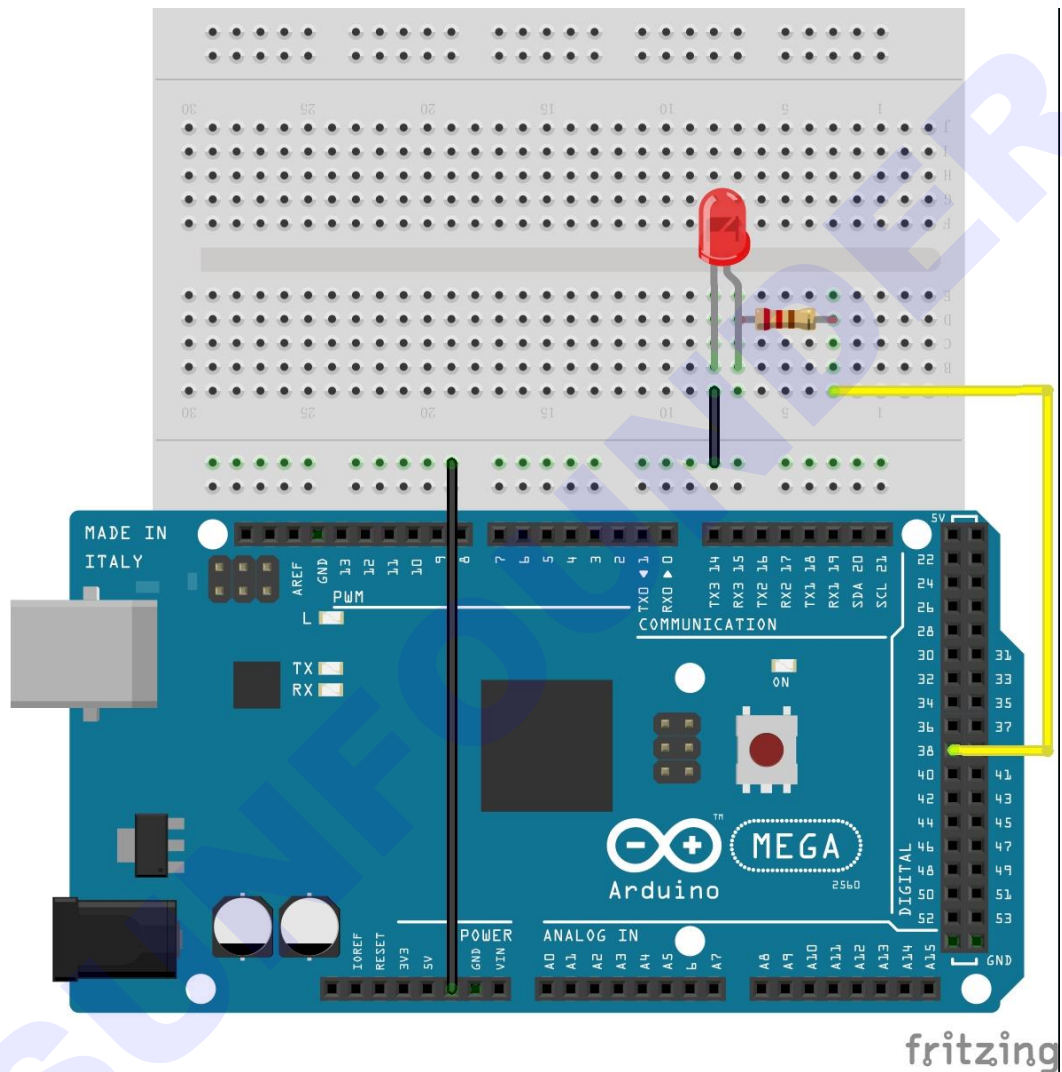
Jumper wires

Steps:

First, connect the circuit

SUNFOUNDER MEGA2560	PIR
36 -----	OUT
5V -----	VCC
GND-----	GND

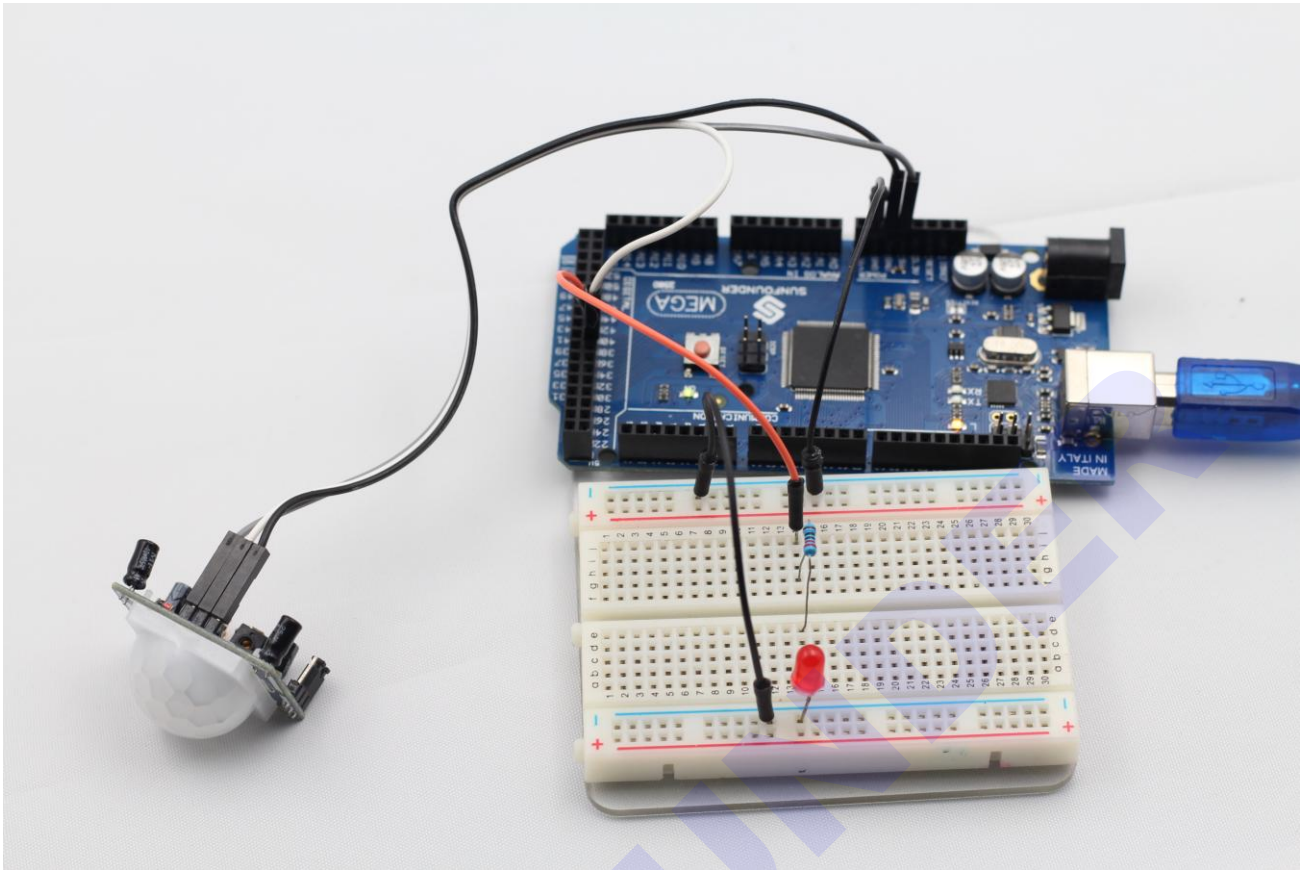
The connection between SUNFOUNDER MEGA2560 and the LED is shown below:



Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board



```
COM11
Pir State:
0
Pir State:
0
Pir State:
0
Pir State:
1
Pir State:
1
Pir State:
1
Pir State:
1
Pir State:
1
Pir State:
1
Pir State:
0
Pir State:
```

Experiment 5 How to Use Gas Pressure Sensor BMP180

Overview:

In this experiment, we will learn how to use gas pressure sensor. We use SUNFOUNDER MEGA2560 as master controller board to read the values from BMP180 and display them on the serial monitor.

Components:

SUNFOUNDER MEGA2560 * 1

BMP180 * 1

USB Cable * 1

Jumper wires

Steps:

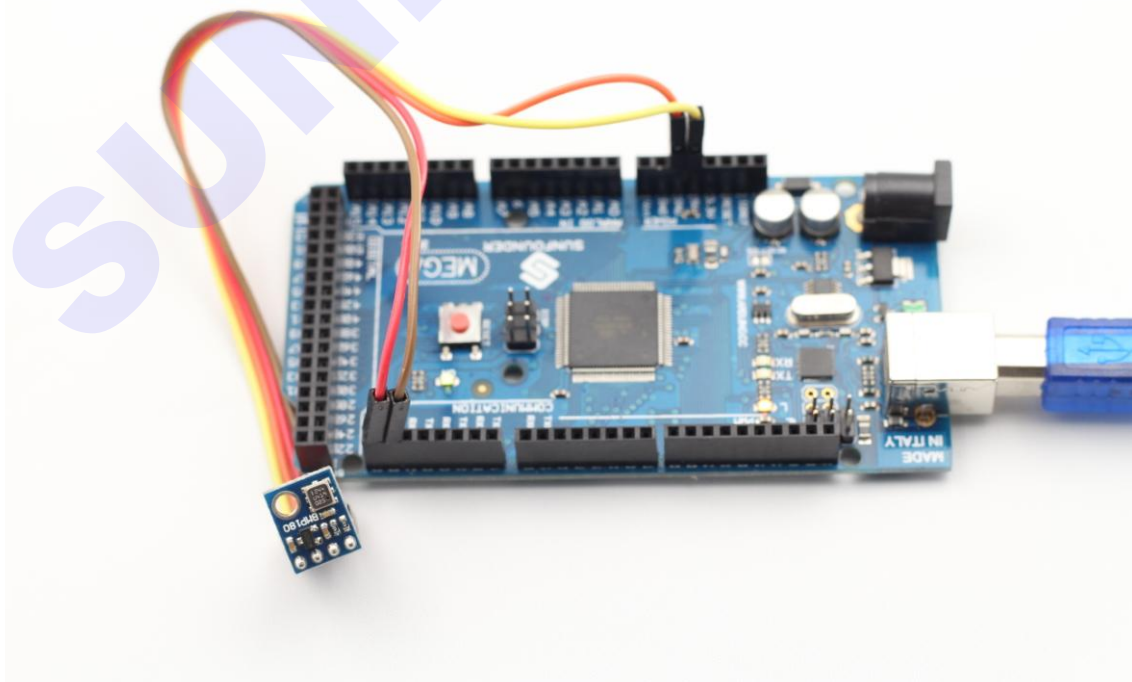
First, connect the circuit

SUNFOUNDER MEGA2560	BMP180
5V -----	VIN
GND -----	GND
SCL -----	SCL(21)
SDA -----	SDA(20)

Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board



```
COM11

Sensor:      BMP085
Driver Ver:  1
Unique ID:   10085
Max Value:   300.00 hPa
Min Value:   1100.00 hPa
Resolution:  0.01 hPa

-----

Pressure:    1006.87 hPa
Temperature: 29.30 C
Altitude:    55.95 m

Pressure:    1006.78 hPa
Temperature: 29.30 C
Altitude:    56.73 m

Pressure:    1006.84 hPa
Temperature: 29.30 C
Altitude:    56.21 m

Pressure:    1006.88 hPa
Temperature: 29.30 C
Altitude:    55.86 m
```

Experiment 6 How to Use RFID Module

Overview:

In this experiment, we will learn how to use RFID module. We use RFID library and simulate SPI with software to realize the communication between SUNFOUNDER MEGA2560 and RFID module. After you have finished the following experimental steps, if you make the matched magnetic card approach to the card reader, the serial monitor will display "Collect!".

Components:

SUNFOUNDER MEGA2560 * 1

RFID Module (including card reader and tag) * 1

USB Cable * 1

Jumper wires

Steps:

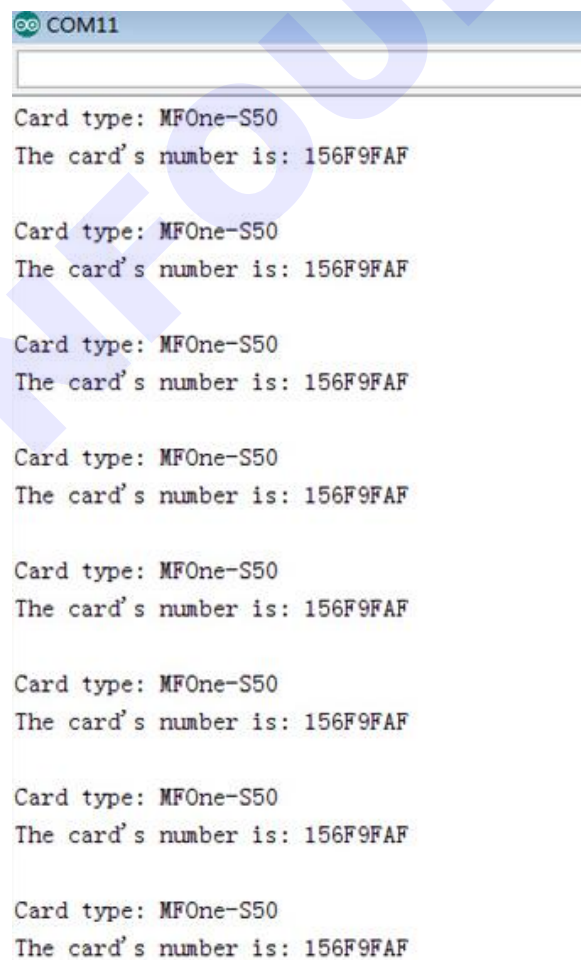
First, connect the circuit

SUNFOUNDER MEGA2560	RFID Module
2 -----	IRQ
3 -----	SCK
4 -----	MOSI
5 -----	MISO
6 -----	NSS
7 -----	RST
GND -----	GND
3.3V -----	VCC

Second, program

Open the getId file provided in our CD in Arduino IDE, compile the file and then burn it into SUNFOUNDER MEGA2560 board.

After burning getId, make the tag approach to the card reader, you will see the following values printed on the serial monitor:



```
COM11
Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF

Card type: MFOne-S50
The card's number is: 156F9FAF
```

Open your serial monitor and write down your magnetic card ID (e.g. my magnetic card show ID is 156F9FAF). Open the rfidTest file provided in our CD and replace the ID in the program with the ID you just wrote down (divide the ID into four parts and fill them according to the following format).

```
else if(id[0] == 0x15 && id[1] == 0x6F && id[2] == 0x9F && id[3] == 0xAF)
{
    Serial.println("Collect!");
}
```

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board



After burning rfidTest, you will see the following values printed on the serial monitor:

```
COM11  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!  
  
Card type: MF0ne-S50  
The card's number is: 156F9FAF  
Collect!
```

Experiment 7 How to Use NRF24L01

Overview:

This routine is divided into sending and receiving part. The sending terminal consists of SUNFOUNDER MEGA 2560, five keys, and NRF24L01 module. The receiving terminal consists of SUNFOUNDER NANO, 4-channel relay module, and NRF24L01 module. The five keys on the sending terminal are defined as sw1, sw2, sw3, sw4, sw5. The first four keys are corresponding to the four channels on the relay module of receiving terminal. The sw5 key is close key. When pressing any keys on sending terminal, the corresponding relay channel on receiving terminal will conduct. When pressing sw5, all the relays will be disconnected.

Components:

SUNFOUNDER MEGA2560 * 1

SUNFOUNDER NANO * 1

NRF24L01 * 2

4-channel relay module * 1

Breadboard * 1

USB Cable * 1

Jumper wires

Steps:

First, connect the circuit

Sending Terminal

SUNFOUNDER MEGA2560

NRF24L01

GND ----- GND
5V ----- VCC
22 ----- CE
24 ----- CSN
26 ----- SCK
28 ----- MOSI
30 ----- MISO
32 ----- IRQ

One end of the SW1 to SW5 is connected to pin 2 to 6 of SUNFOUNDER MEGA2560, the other end of them is connected to GND.

Receiving Terminal

SUNFOUNDER NANO

NRF24L01

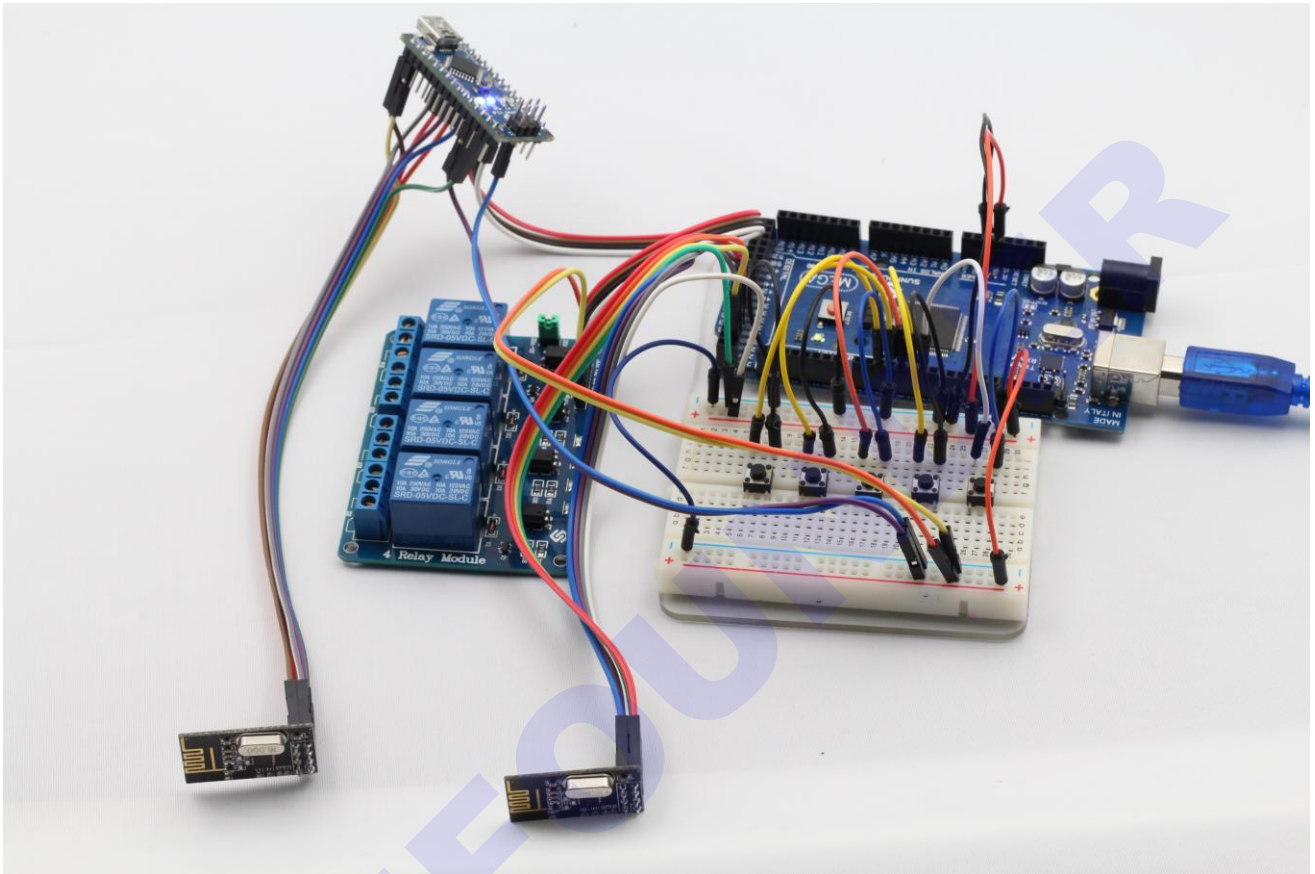
4-Channel Relay Module

GND ----- GND ----- GND
5V ----- VCC ----- VCC
D8 ----- CE
D9 ----- CSN
D10 ----- SCK
D11 ----- MOSI
D12 ----- MISO
D13 ----- IRQ
D3 ----- IN1
D4 ----- IN2
D5 ----- IN3
D6 ----- IN4

Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn nrfTest_send into SUNFOUNDER MEGA2560 and nrfTest_receive into SUNFOUNDER NANO, the physical connection is shown below:



DeviceBit Platform and Control Board Joint Test

How to Operate DeviceBit Platform

Register an Account on DeviceBit Platform

Login the website (<http://www.devicebit.com/home/publicsensors>) as shown in fig 5.1. Click **Sign Up** on the top right corner of the page.

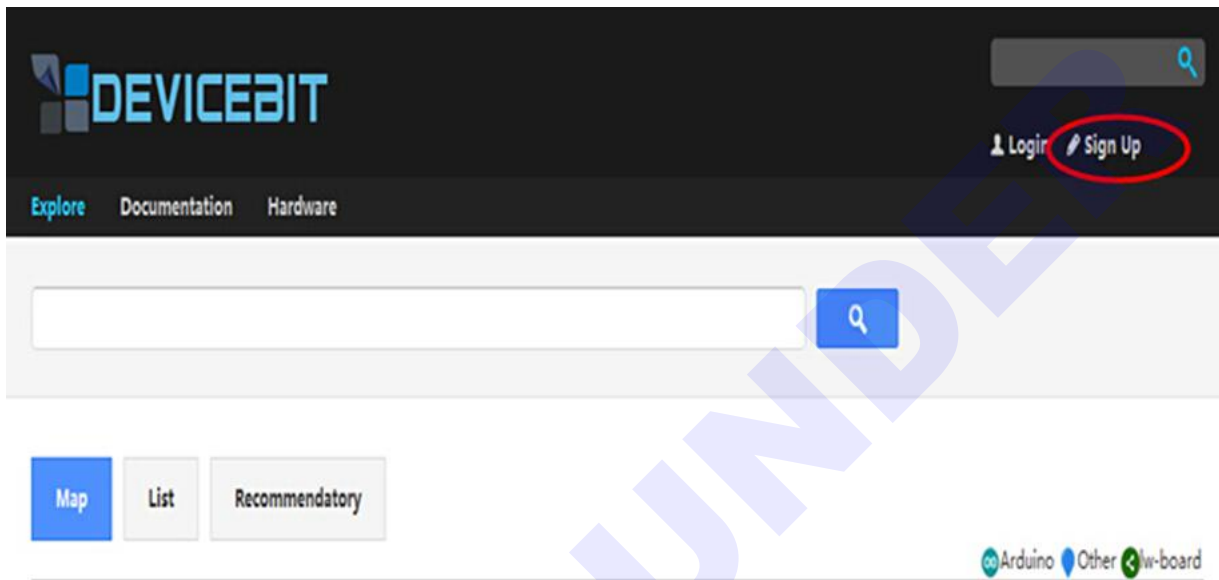


Fig 5.1

The following page will appear, as shown in fig 5.2. Type your username, password, and Email. Then log on to your mailbox to activate your account.

Fig 5.2

Add a New Device

Click to enter User Center, go to **My Devices** -> **Devices** -> **Add A device**, fill in corresponding information and then save.

Device List **Add A Device (Default)** Add A Device (by SN)

ID: 01

Type: Arduino

Template: Without Template

Name: My Device

Controllable: ☒ Yes

API Address: http://tcp.lewei50.com:9965/?method=send&gatewayNo=01

Internet Available: ☒ Yes

Internet IP: tcp.lewei50.com

Fig 5.3

ID: The ID of the device, allocated by system automatically. It is unique for each device.

Type: The type of the device.

Ø Arduino: Arduino Board.

Ø Lw-board: The board developed by DeviceBit.

Ø Other: Other board.

Name: The name of the device. For example, Smart Home

Description: Description of the device.

Public: If your devices are public, other users can also view them and their measurement.

Location: The location of your device. Click the location point on the Map.

Add a New Sensor or Controller

The next step is to add a new sensor or controller connected to your device.

1. Add a new sensor

As shown in fig 5.4, go to **My Devices** -> **Sensors&Controllers** -> **Sensors**, and then click **Add** button to create a new sensor.

Sensors Controllers

Device: All Type: All ID: Name: Query Add

ID	Device	Name	Last Value	Last Update Time	Type	Public	Sort No.	Operation
Humidity	My Device	Humidity	75%	2014-03-18 07:11	Humidity Monitoring	Yes		Alarm History Edit Delete
Temperature	My Device	Temperature	32°C	2014-03-18 07:11	Temperature Monitoring	Yes		Alarm History Edit Delete

[Total Recording:2 Total Pages:1]

Fig 5.4

Now, the following page will appear, as shown in Fig 5.5:

The screenshot shows the 'Add A Sensor' form in the DEVICEBIT application. The form is titled 'Add A Sensor' and 'Add A Controller'. It contains the following fields:

- ID: A text input field with the value 'Humidity'.
- Type: A dropdown menu with the value 'Humidity Monitoring(%)'.
- Unit: A text input field with the value '% '.
- Device: A dropdown menu with the value 'My Device'.
- Name: A text input field with the value 'Humidity'.
- Data Conversion: A section with 'Coefficient' and 'Offset' input fields. Below them is a note: 'The final saved value=uploaded value*coefficient+offset.(Please leave blank if you don't need to set'.
- Image: A section with a 'Choose File' button, 'No file chosen' text, and '150*150' dimensions.
- Public: A checkbox that is checked, with the label 'Yes'.

The 'Add A Sensor' tab is circled in red, and the 'Public' checkbox is also circled in red.

Fig 5.5

ID: The ID of the new sensor. It is unique for each sensor.

Type: The type of the new sensor.

Unit: The unit of the measurement.

Device: Select the device which the sensor is to be connected to.

Name: The name of the new sensor.

Data Conversion: To calibrate your sensor's measurement. You can fill in the Coefficient and offset if applicable.

Picture: Upload the picture of your sensor.

Public: If your sensors are public, other users can also view them and their measurement.

Normal range: The normal range of the sensor's measurement. You can set it according to your actual need.

Overrange alarm: Turn it on and you will receive the alert by SMS or email when the sensor's measurement is out of the range.

Data post period: The system indicates that the sensor is OFFLINE if it does not get the data posted by the sensor within every data post period set here.

Remark: Remark of the sensor.

2. Add a new controller

As shown in fig 5.6, go to **My Devices** -> **Sensors&Controllers** -> **Controllers**, and then click **Add** button to add a new controller.

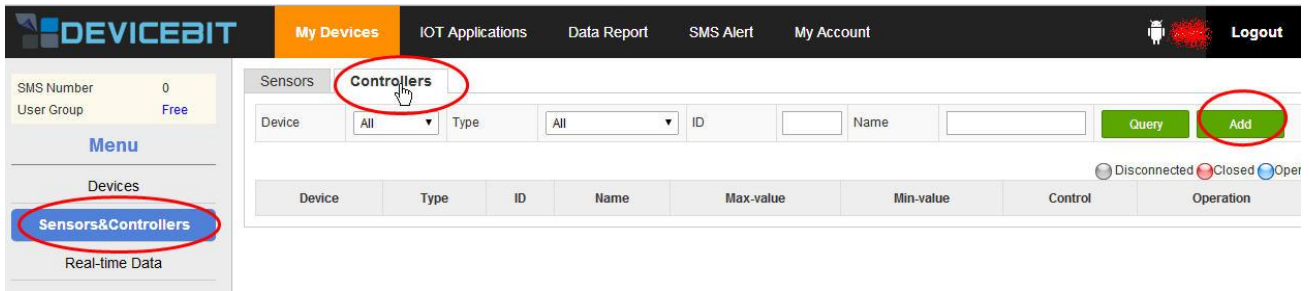


Fig 5.6

Now, the following page will appear, as shown in Fig 5.7:

Fig 5.7

Fill in the related information and then click Save.

The related parameters are explained as follows:

ID: The ID of the new controller. It is unique for each controller.

Name: The name of the new controller.

Device: To select the device which the controller is connected to.

Type: The type of the new controller, which can be switch control or numerical control. For switch type, 0 or 1 is used to control the switch. While for numerical type, the controller status is adjusted by different numerical values.

Max-value: The max-value for numerical type.

Min-value: The min-value for numerical type.

How to Operate Control Board

Control board is divided into acquisition board and control board. The acquisition board is mainly composed of SUNFOUNDER MEGA2560 and W5100, used to collect sensor data, upload the sensor data to DeviceBit Platform, receive control commands from DeviceBit Platform, and finally send these control commands to the control board via wireless RF module. The control board will

perform related operations after receiving these commands. That is the working process of control board terminal. For specific steps, please see below:

Acquisition board

Components:

SUNFOUNDER MEGA2560 * 1

W5100 * 1

DHT11 * 1

Photoresistor module * 1

Gas sensor MQ-2 * 1

RFID-RC522 * 1

PIR * 1

Gas pressure BMP180 * 1

NRF24L01 * 1

LED * 2

220 Ohm Resistor * 2

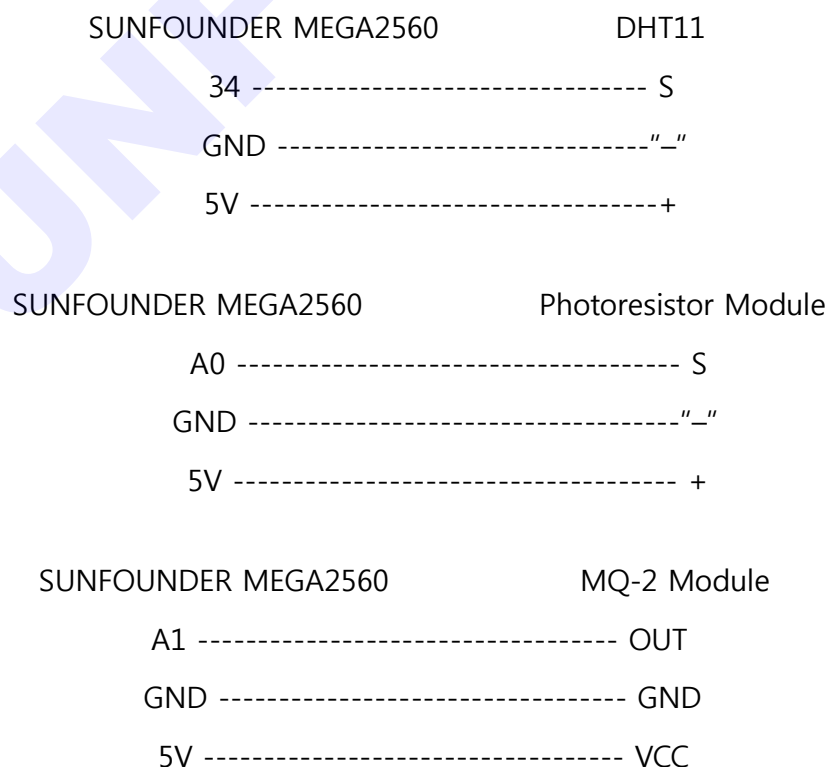
USB Cable * 1

Breadboard * 1

Jumper wires

Steps:

First, connect the circuit (connect W5100 and SUNFOUNDER MEGA2560 as shown below)



SUNFOUNDER MEGA2560	PIR Module
36 -----	OUT
GND -----	GND
5V -----	VCC

SUNFOUNDER MEGA2560	RFID Module
2 -----	IRQ
3 -----	SCK
4 -----	MOSI
5 -----	MISO
6 -----	NSS
7 -----	RST
GND -----	GND
3.3V -----	VCC

SUNFOUNDER MEGA2560	NRF24L01
22 -----	CS
24 -----	CSN
26 -----	SCK
28 -----	MOSI
30 -----	MISO
32 -----	IRQ
GND -----	GND
5V -----	VCC

SUNFOUNDER MEGA2560	BMP180
SCL(21) -----	SCL
SDA(20) -----	SDA
GND -----	"-"
5V -----	+

Connect a 220 Ohm resistor and an LED in series to pin 38 on SUNFOUNDER MEGA2560. Connect another 220 Ohm resistor and another LED in series to pin 40 on SUNFOUNDER MEGA2560. These



Second, program (Please refer to example code in CD we provide)

Program description: Since we need to detect the status of RFID and PIR in real time, so we upload the data to DeviceBit Platform, and also we use many delays. If we place the code that used to detect the status of RFID and PIR in main loop, it will seriously affect the real-time performance. As a result, we use timer 1 of master chip to regularly detect the status of RFID and PIR to ensure real time. The timer 1 is actually easy to use, that is, set several registers. The registers we use here are TCCR1A (T/C1 control register A), TCCR1B (T/C1 control register B), TCNT1 (16-bit counting register), TIMSK (interrupt mask register) and SREG (interrupt register).

TCCR1A: Since we only use the timing overflow interrupt of timer 1 here, so we set it to 0 to mask the output function of PWM.

TCCR1B: Here we only need to set the CS12, CS11, and CS10 bit of TCCR1B to 1, 0, and 1. That is, select 1024 frequency division.

TCNT1: This register is used to set the timing initial value of timer. We set it to $65536 - F_{CPU}/F_{DIV} * TIME$, F_{CPU} (main frequency of master control board) to 16000000 (16MHz), F_{DIV} (frequency dividing ratio) to 1024. For $TIME$ (timing length), we define it as 0.1. That is, it will go to the interrupt service program of timer 1 to perform corresponding tasks every 0.1 seconds. The task here is to query the status of PIR and RFID.

TIMSK: This register can be used to turn timers on and off. The timer 1 is opened as long as we set TOIE1 to 1.

SREG: As long as we set the seventh I to 1, we can open the master switch of interrupt. The TOIE1 setting of timer 1 can take effect only if the master switch of the interrupt is opened.

The interrupt service routine for Timer 1 is corresponding to the ISR (TIMER1_OVR_vect) function. Once entering the interrupt service routine, the 16-bit counting register will be reset. So we need to reload the initial value, that is, call " $TCNT1 = 65536 - F_{CPU}/F_{DIV} * TIME;$ " statement again in the interrupt service routine.

Third, compile the program

Fourth, burn the program into SUNFOUNDER MEGA2560 board

Control Board

Components:

SUNFOUNDER NANO * 1

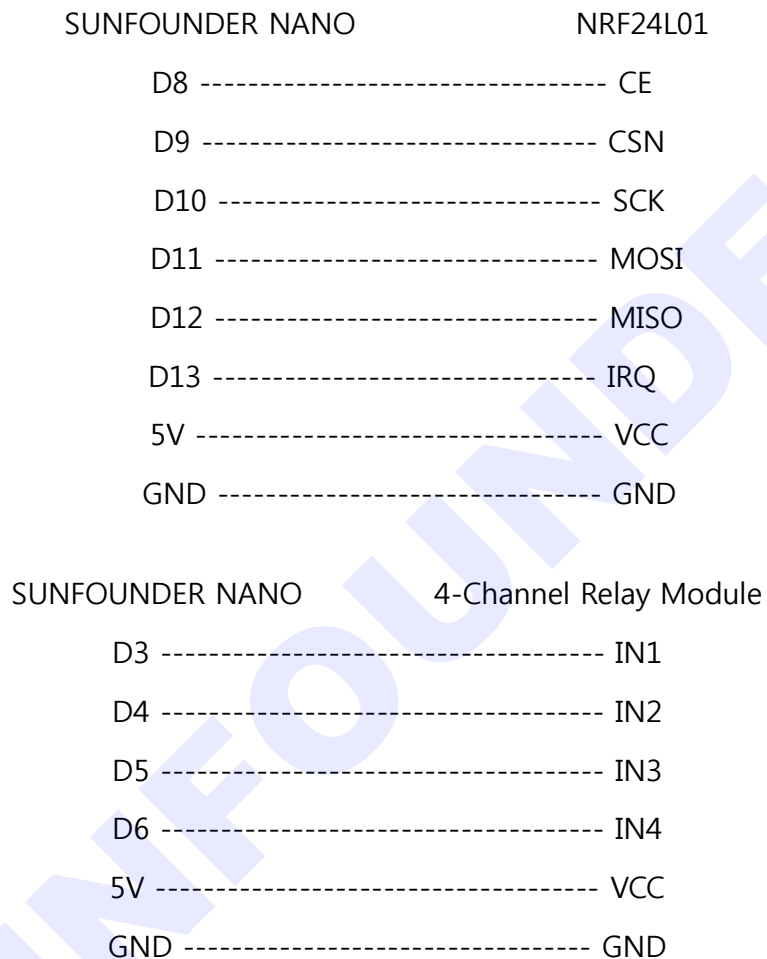
NRF24L01 * 1

4-channel Relay Module * 1

USB Cable * 1

Steps:

First, connect the circuit

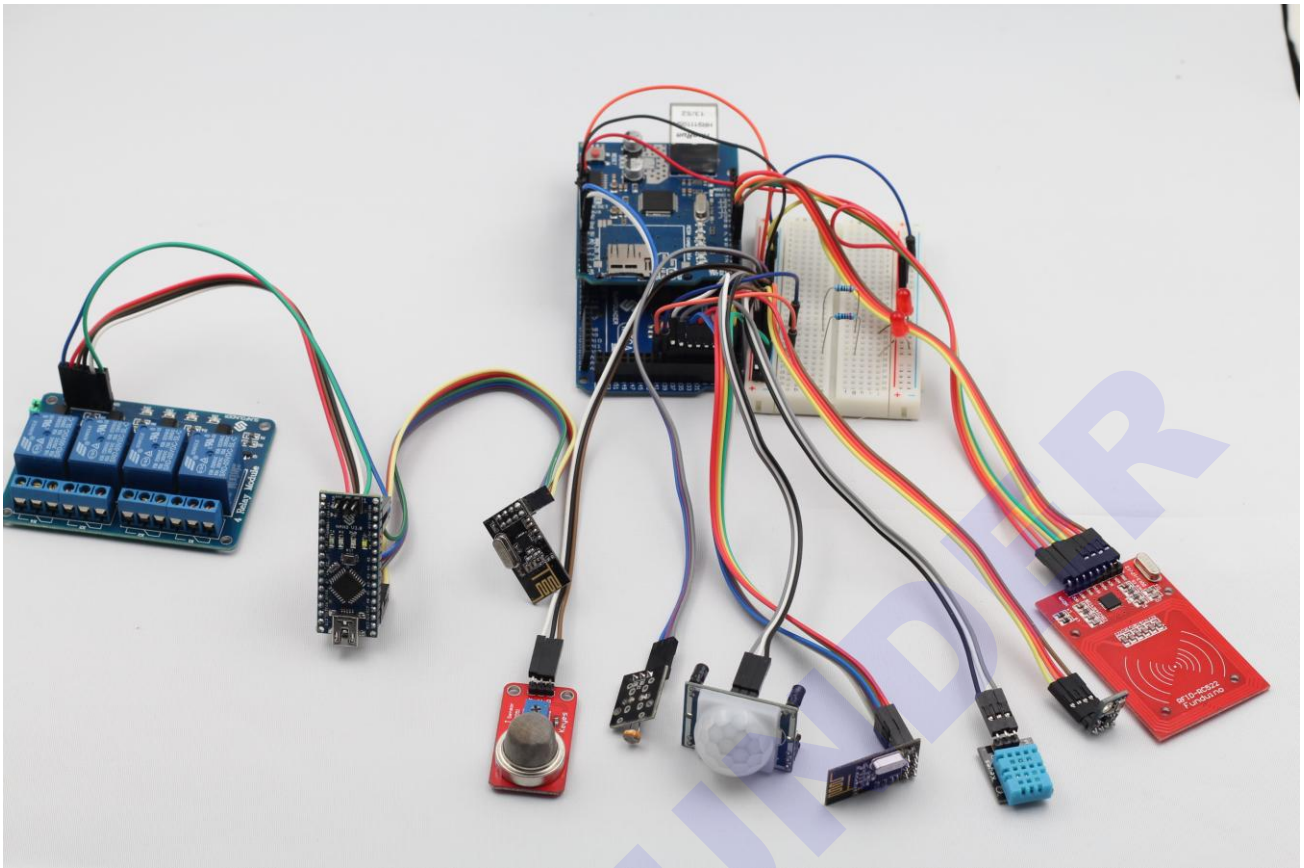


Second, program (Please refer to example code in CD we provide)

Third, compile the program

Fourth, burn the program into SUNFOUNDER NANO board

The physical connection diagram for joint debugging is shown below:



After all the above steps, visit <http://www.devicebit.com/home/publicsensors> to enter the DeviceBit Platform. Click **User Center** on the top right corner of the page (If you have already login), you will see the following interface as shown in Fig 5.8:

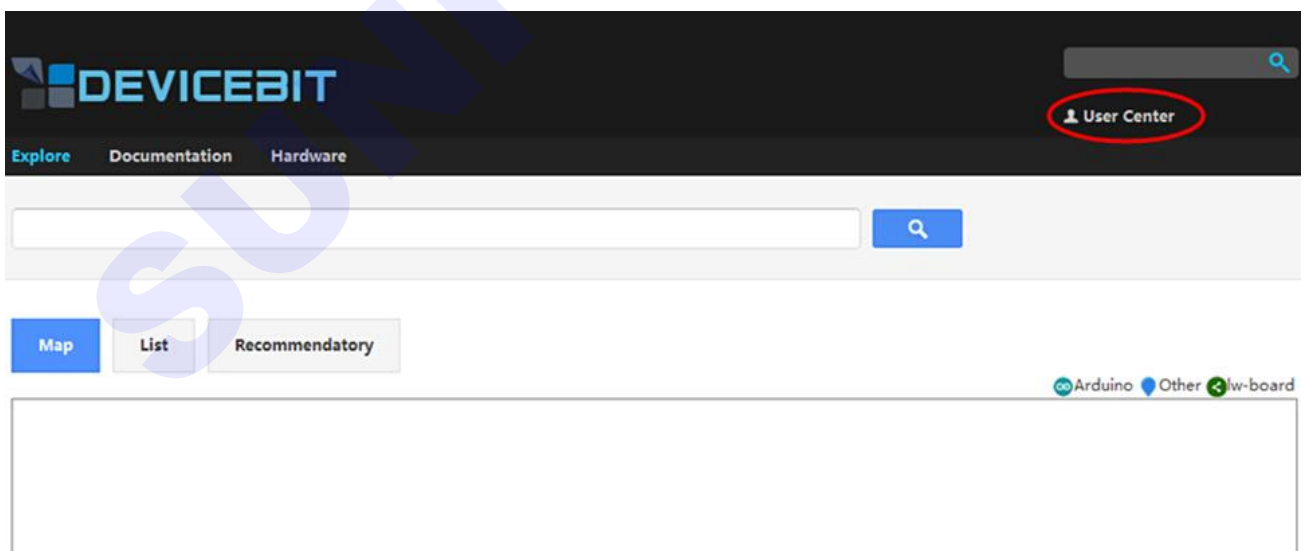


Fig 5.8

If you have not login, click **Login**, as shown in Fig 5.9:

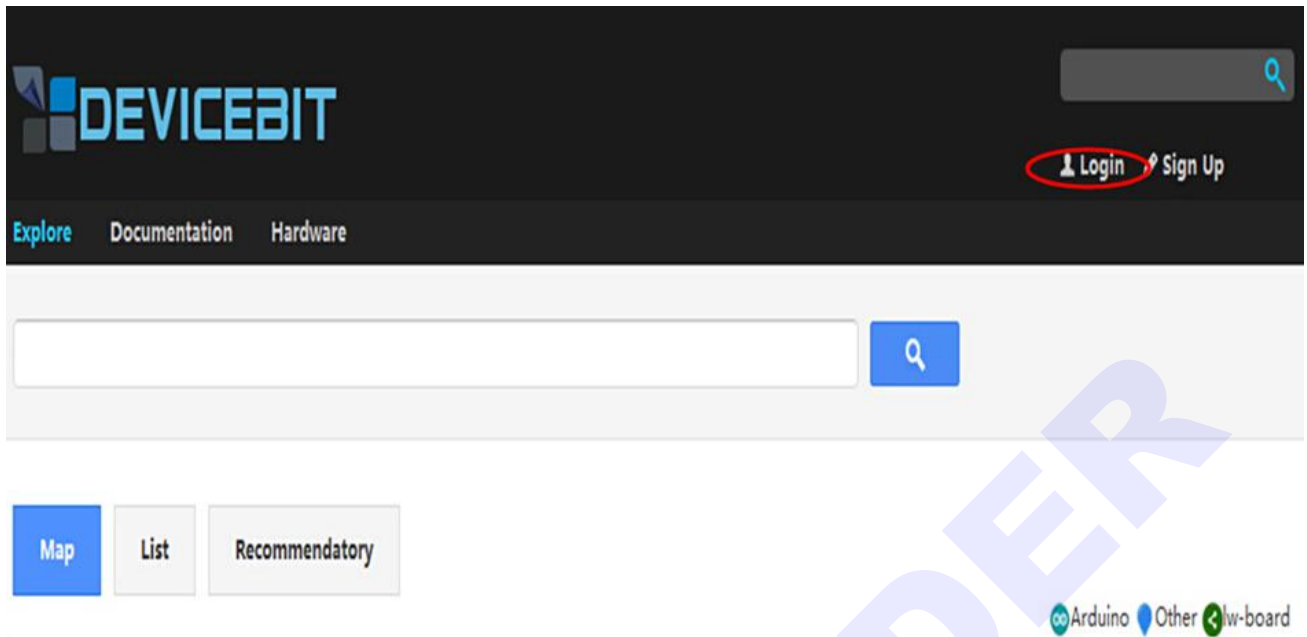


Fig 5.9

Fill in your Username, Password and Verification Code, as shown in Fig 5.10:

The image shows the 'User Login' form on the DeviceBit website. The form is titled 'User Login' and contains three input fields: 'Username', 'Password', and 'Verification Code'. Each of these three fields is circled in red. The 'Verification Code' field has a small image of a verification code '811381' next to it. Below the input fields is a checkbox labeled 'Remember Me' which is checked. At the bottom of the form are two buttons: 'Login' and 'Forgot Password'.

Fig 5.10

After entering into **User Center**, you will see the following interface as shown in Fig 5.11:

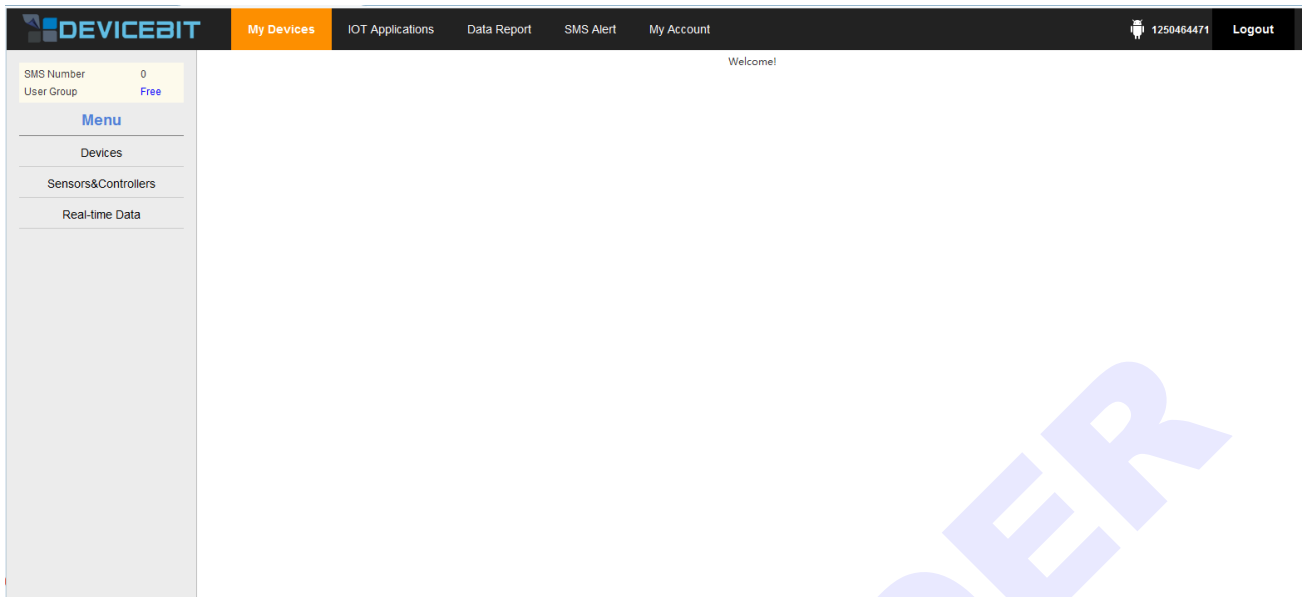


Fig 5.11

Click **Devices**, you will see the following interface as shown in Fig 5.12:

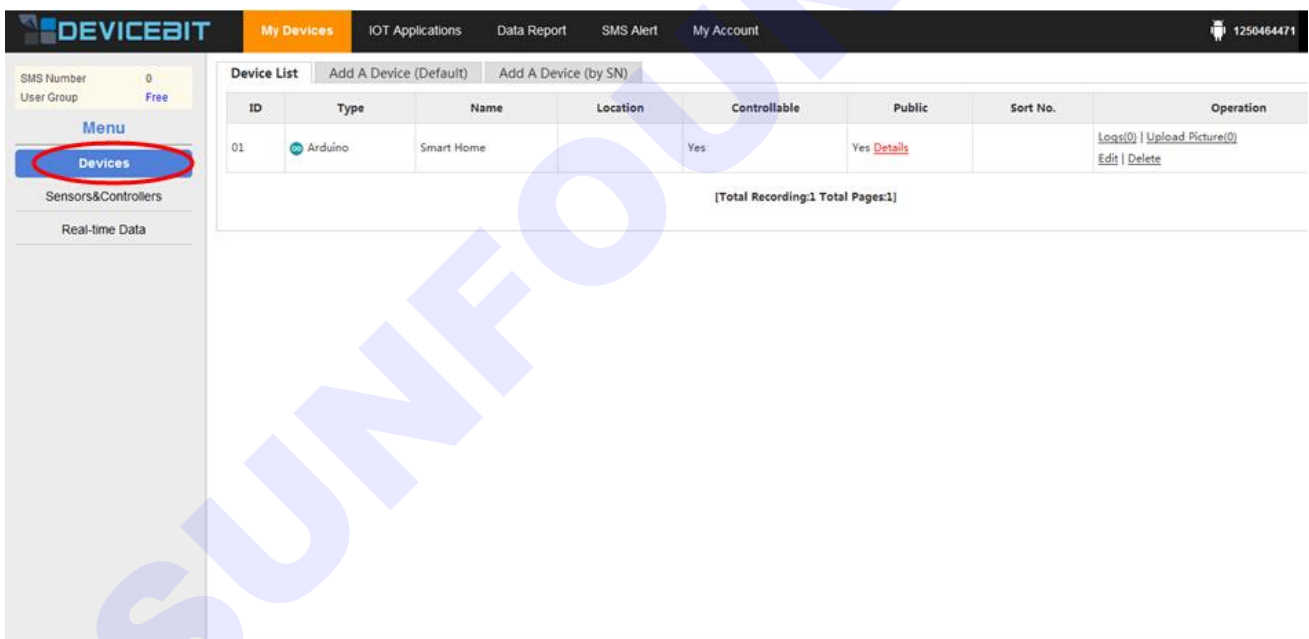


Fig 5.12

Click red "**Details**" to see your device details, as shown in Fig 5.13:

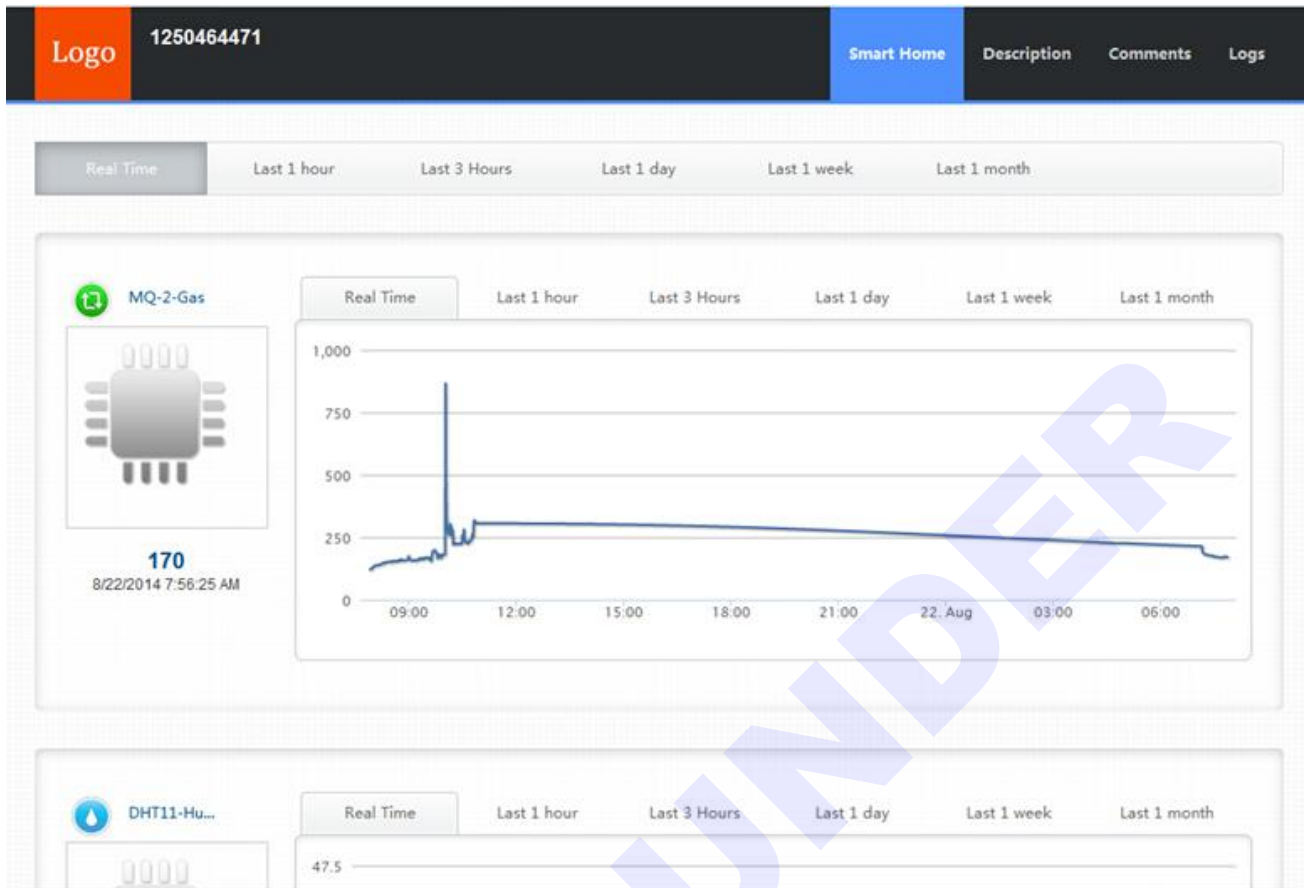


Fig 5.13

Return to the interface as shown in Fig 5.11, click **Sensors&Controllers** on the left side to see your sensors and controllers, as shown in Fig 5.14:

Figure 5.14 displays the Sensors and Controllers interface. The interface shows a list of sensors and controllers with columns for ID, Device, Name, Last Value, Last Update Time, Type, Public, Sort No., and Operation. The 'Sensors&Controllers' menu item is highlighted in the left sidebar.

ID	Device	Name	Last Value	Last Update Time	Type	Public	Sort No.	Operation
Gas	Smart Home	MQ-2-Gas	169	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete
Humidity	Smart Home	DHT11-Humidity	42%	2014-08-22 07:49	Humidity Monitoring	Yes		Alarm History Edit Delete
Light	Smart Home	Photocell-Light	230	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete
PIR	Smart Home	PIR	2	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete
Pressure	Smart Home	BMP180-Pressure	1006.6545Hpa	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete
RFID	Smart Home	RFID	0	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete
Temperature	Smart Home	DHT11-Temperature	29°C	2014-08-22 07:49	Temperature Monitoring	Yes		Alarm History Edit Delete
Weather	Smart Home	BMP180-Weather	-1	2014-08-22 07:49	Other Type	Yes		Alarm History Edit Delete

[Total Recording:8 Total Pages:1]

Fig 5.14

Click **Controllers** to see your controllers, as shown in Fig 5.15:

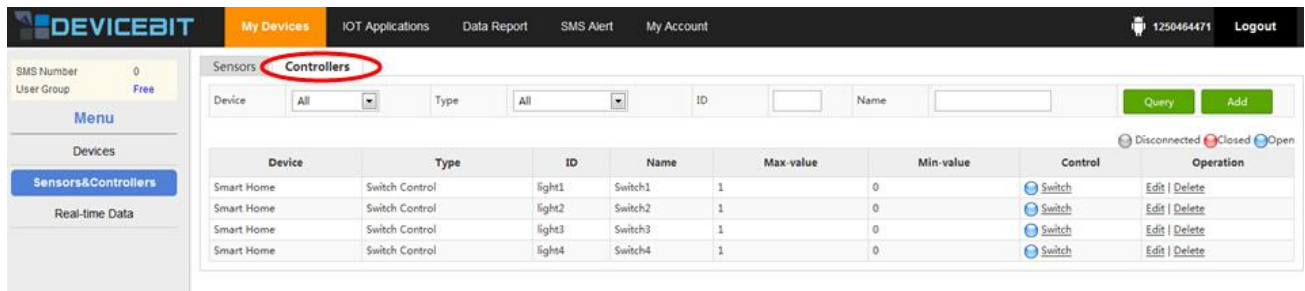


Fig 5.15

If everything goes well, the status icon of **Switch** should display as blue, which indicate that your master control board is connected to Devicebit Platform. Click **Switch**, if the icon color changes from blue to red, check the status of your relay at the receiving end, if normal, the corresponding relay should pull in, and the corresponding LED light should be lit. Click **Switch** again, the icon color change back from red to blue, check the relay status, if normal, the LED light just lit should go out, and the corresponding relay should be disconnected.

Back to the interface as shown in Fig 5.11, click **Real-time Data** on the left side to see the real-time data of your sensor, as shown in Fig 5.16:

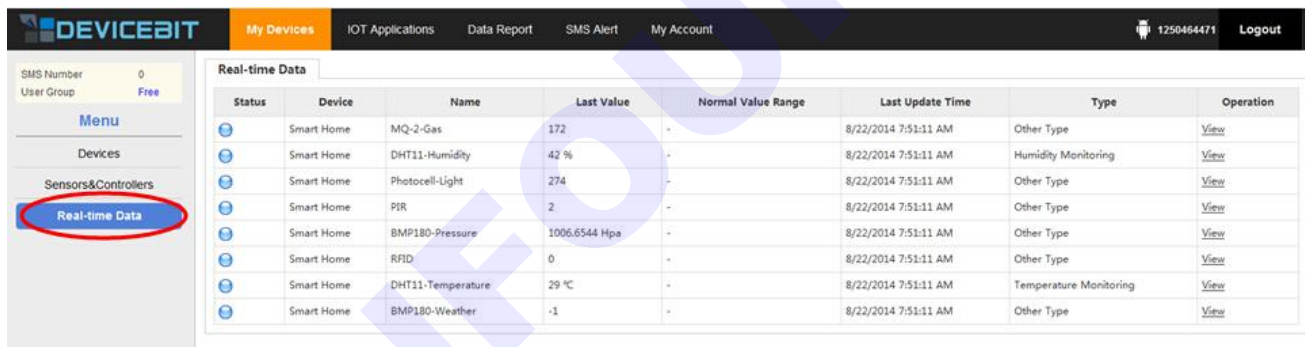


Fig 5.16

If your master control board is connected to Devicebit Platform and work as normal, the status indicator light of the corresponding sensor on the left side should display as blue, otherwise grey. Click **View** on the right side to see the corresponding curve of sensor, as shown in figure 5.17:

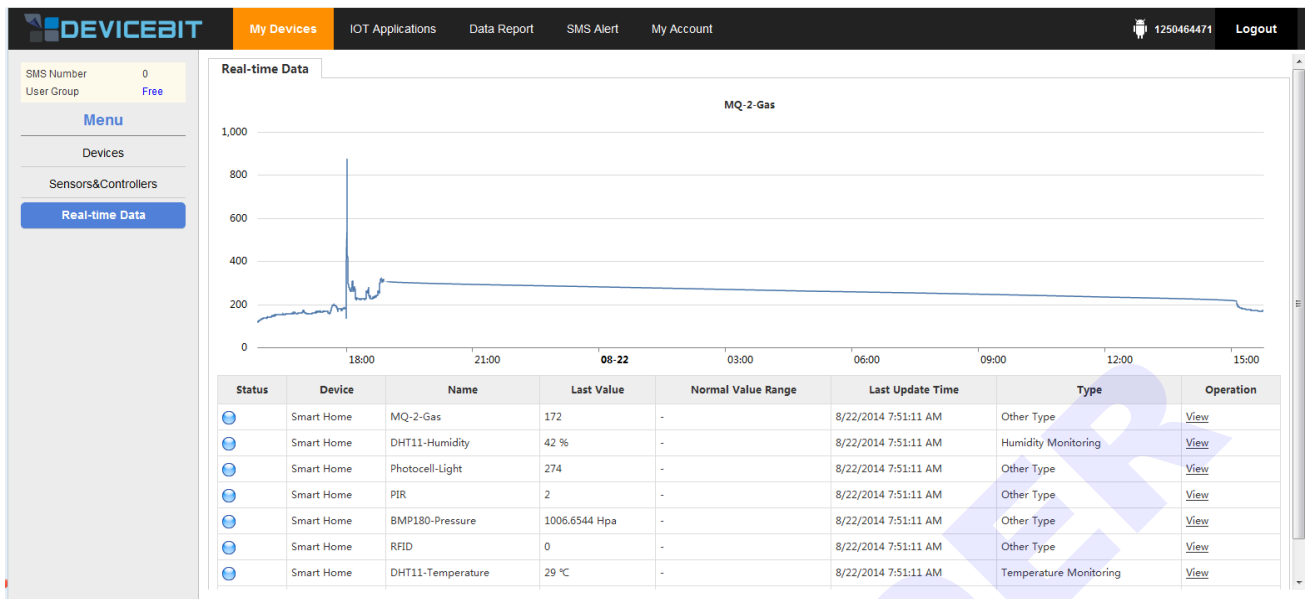


Fig 5.17

How to Operate Web app on Mobile Phone

Visit m.devicebit.com on your mobile phone to enter into Devicebit Platform.



Fig 5.18

Click the icon on the top left corner to select a language.

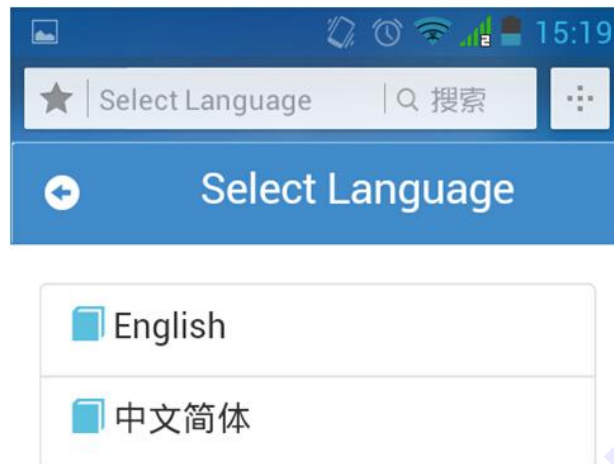


Fig 5.19

Select **English**.

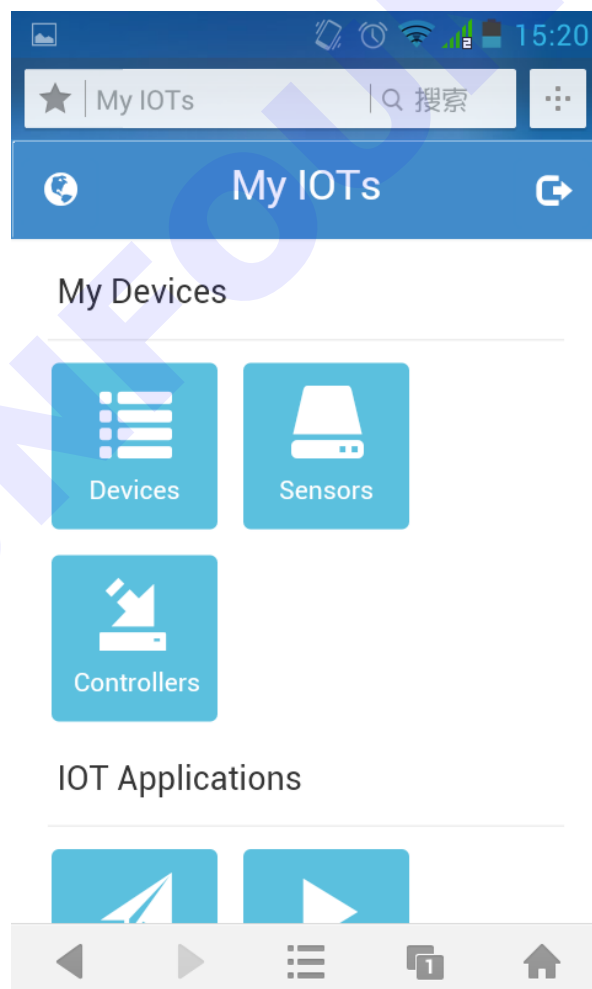


Fig 5.20

Click **Devices** icon.

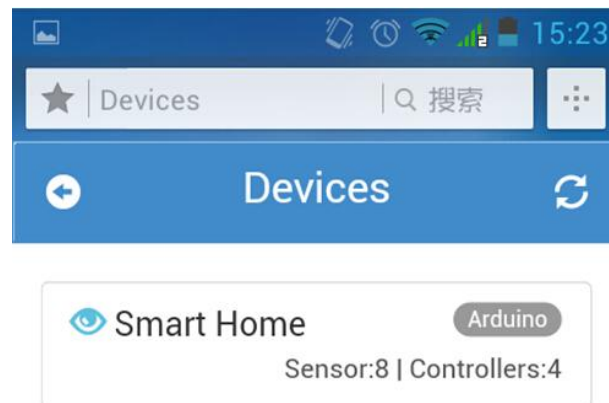


Fig 5.21

The figure above shows the number of sensors and controllers, click the back icon on the top left corner to return to the interface as shown in Fig 5.20, then click **Sensors** icon.

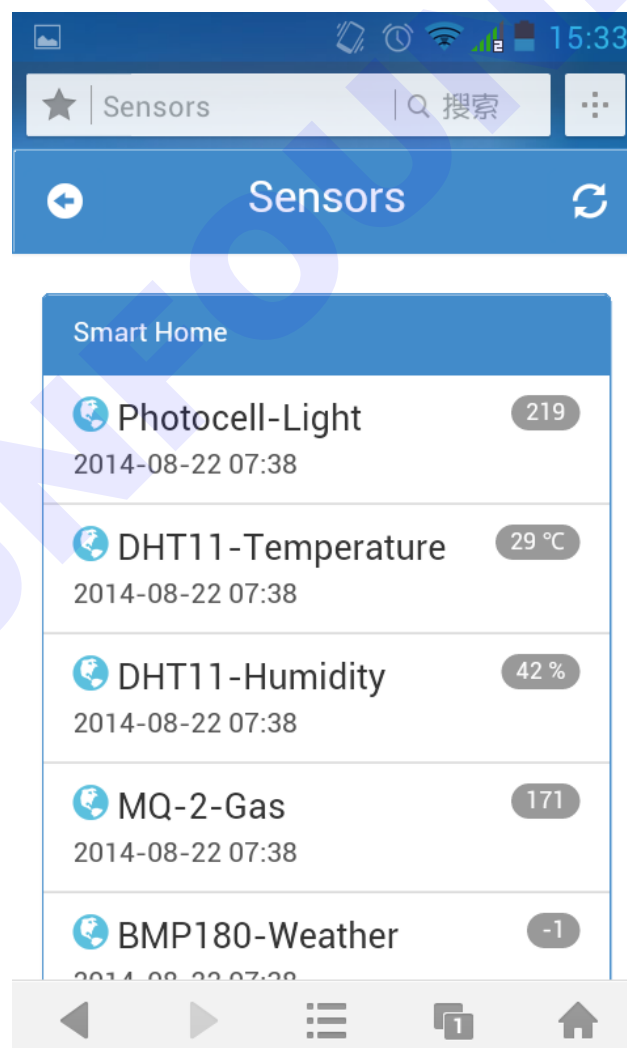


Fig 5.22

Click the back icon on the top left corner to return to the interface as shown in Fig 5.20, then click **Controllers** icon.

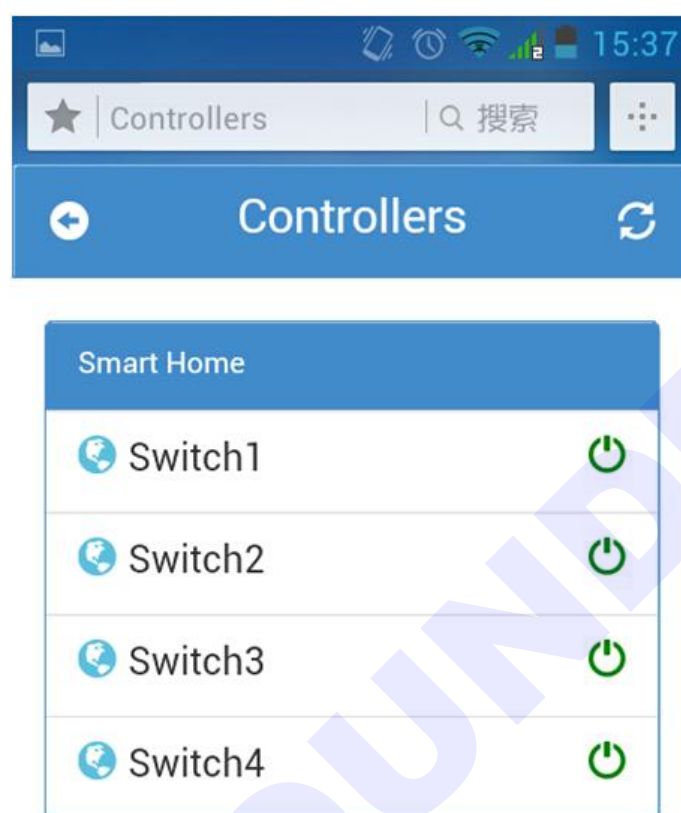


Fig 5.23

If the connection between your master control board and DeviceBit Platform works normally, switch icons will display as green, otherwise grey. Click any switch icon, you will see the icon color change from green to red. Check your receiving terminal, if everything goes well, the corresponding relay will conduct and the corresponding LED will be lit.

Summary

This paper details how to DIY a simple Smart Home system with SUNFOUNDER MEGA2560 and W5100. Through hands-on practice, you have had a more visual recognition on the Internet of Things. Of course this is only the first step you set your foot in the field of Internet of Things, you can make further exploration on the basis of this paper.