

Lesson 9 Smart Agriculture IoT System

The train of the 5G era has long departed, while the waves of artificial intelligence and big data are still surging. Nowadays, Internet of Things (IoT) technology is at the forefront of technological development. Compared to traditional hardware devices, IoT technology enables various hardware devices to be connected to the Internet through information transmission devices, facilitating the exchange of information and achieving intelligent identification and management, bringing convenience to people's daily lives.

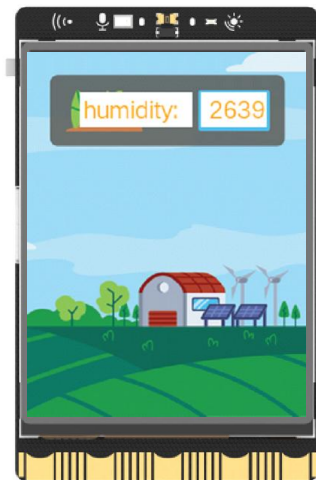
In the previous lesson, we designed a smart agriculture visualization system using the board, simulating real-time monitoring and improvement of crop growth environment in agriculture. However, the data collected could only be displayed on the screen of the board, requiring us to be physically close to the board for observation, which is inconvenient. Is there a way for us to stay at home and remotely access real-time environmental data?

In this lesson, let's explore the use of powerful IoT technology to simulate a smart agriculture IoT system and solve this problem!



Task Objectives

The humidity values detected by the soil moisture sensor are displayed on the screen. At the same time, the SIoT IoT system is activated to synchronize the data transmission to the IoT platform for viewing. When the observed soil humidity value is inadequate, the relay and water pump can be controlled for watering by entering the specified characters "on" and "off" on the web interface of the IoT platform.



Topic : SmartAgricultureIoTSystem/Soil_moisture_value

Send message

(Adding a -> prefix to the message means that the message is a command to the device. For example "->off")

Starting time End time 10

Hide/Show chart ☐ Automatically refresh messages ☐

Topic	value	time
SmartAgricultureIoTSystem/Soil_moisture_value	on	2023-05-17 16:17:58
SmartAgricultureIoTSystem/Soil_moisture_value	2547	2023-05-17 16:17:58
SmartAgricultureIoTSystem/Soil_moisture_value	2551	2023-05-17 16:17:57
SmartAgricultureIoTSystem/Soil_moisture_value	2557	2023-05-17 16:17:56
SmartAgricultureIoTSystem/Soil_moisture_value	2567	2023-05-17 16:17:55
SmartAgricultureIoTSystem/Soil_moisture_value	2567	2023-05-17 16:17:54
SmartAgricultureIoTSystem/Soil_moisture_value	2567	2023-05-17 16:17:53



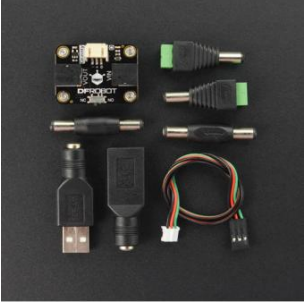




Send successfully

Knowledge points

1. Understand IoT, MQTT, and SLoT
2. Learn how to connect to an IoT platform using the SLoT library
3. Learn how to send messages to an IoT platform using the SLoT library
4. Learn how to receive messages from an IoT platform using the SLoT library

Material List

Hardware List:

		
UNIHiker x1	Soil Moisture Sensor x1	Easy Relay Module x1
		
Type-C & Micro Dual-Use USB Cable x1	Wall Adapter Power Supply 12VDC 1A x1	Immersible Pump & WaterTube x1
		
PH2.0-3P white silicone twisted wire at both ends x2		

Software Preparation: Mind+ Programming Software x1

Others:

1. Plant pot with a plant x1
2. Beaker with water x1
3. Cross/flathead screwdriver x1

Knowledge background

1. Internet of Things (IoT)

Internet of Things (IoT) is a network that allows everyday objects with independent functionality to be interconnected and communicate with each other through the Internet, traditional telecommunications networks, and other information carriers.

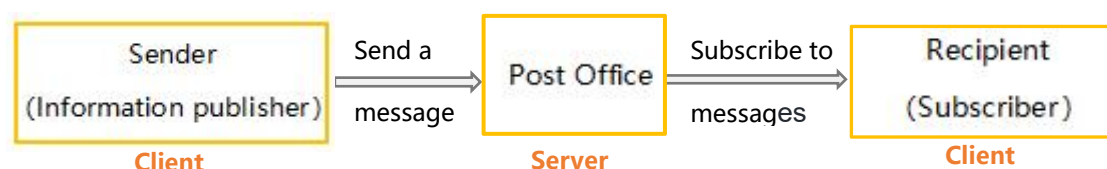
IoT is an extension of the Internet, where the endpoints of the Internet are computers (PCs, servers), while the endpoints of IoT are hardware devices (phones, computers, sensors, etc.). Through IoT, we can connect and manage various machines and devices using a central computer, enabling connectivity between physical objects. For example, by connecting household appliances such as lights, fans, and air conditioners to the IoT, we can control them using a smartphone.



2. MQTT

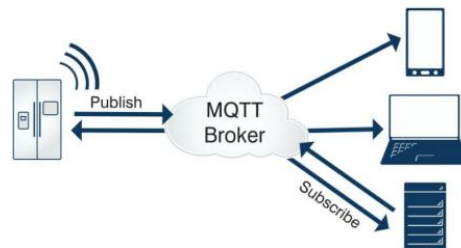
In order for computers to connect to the Internet, communication protocols need to be defined. It's similar to a group of people from different countries coming together, where they can only communicate if they speak the same language. Similarly, to achieve the interconnection of everything, a set of protocols is required. In fact, there are many protocols available for building an Internet of Things (IoT) system, including MQTT, HTTP, XMPP, CoAP, and more. In this case, we are using MQTT, which allows us to transmit messages in the IoT.

MQTT is a client-server-based publish/subscribe messaging protocol, and the server is the platform on which we build our IoT system. The clients can be message publishers or subscribers. Each client can connect to the server. Let's take the example of a post office that we are all familiar with. We can go to the post office to send and receive letters. When we send a letter, we are message publishers to the post office, and when we receive a letter, we become subscribers. At the same time, whether we are sending or receiving letters, we are customers. The post office, which serves the purpose of sending and receiving letters, is the server in this scenario, providing us with letter delivery services.



When a publisher has new data to distribute, it sends a control message to the server containing the data. The server then distributes the information to any client that has subscribed to that

topic. The publisher does not need any data about the number or location of subscribers, and the subscribers do not need to configure any data about the publisher.



In addition, the messages transmitted by MQTT can be divided into two parts: Topic and payload.

- (1) Topic can be understood as the type of message. Subscribers receive the message content (payload) of that topic once they subscribe to it.
- (2) Payload can be understood as the actual content of the message, which refers to the specific content that subscribers want to receive.

3. SIoT

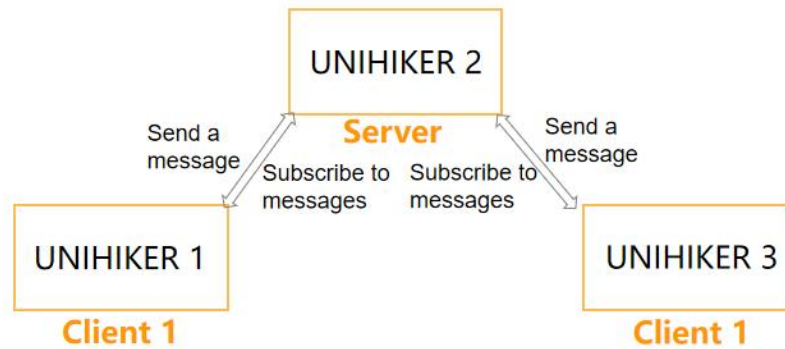
SIoT is a cross-platform open-source MQTT server program designed for education. The "S" stands for science and simplicity. SIoT supports operating systems such as Win10, Win7, Mac, Linux, and can be easily launched with just one click. Users do not need to register or configure the system to use it.

SIoT is also designed to help primary and secondary school students understand the principles of the Internet of Things and develop various creative applications based on IoT technology. It focuses on the collection and export of IoT data, making it an excellent choice for collecting scientific data.



In the case of the board used in this context, the SIoT application program is already built-in on the board. Therefore, when using it, we can directly enable the service by clicking on the application switch. Additionally, for ease of use, we have developed a Python library called "SIoT" with the aim of connecting to the MQTT server and facilitating message transmission through Python programming.

In practical usage, we can consider the board that enables the SIoT service as a server, and the board (device) connected to the sensors for data detection as the client.



If both enabling the SloT service and connecting sensors to detect data are done on the same UNIHAKER, then it functions as both a client and a server.



Tips: In this lesson, we will implement IoT functionality using a single UNIHAKER. In the next lesson, we will use three boards to simulate a multi-node IoT system.

4. Common Functions in the SloT Library

(1) `init()`, `connect()`, `loop()` functions for initialization and connection to the IoT platform

The `init()` function in the SloT library is used to initialize the connection to the IoT platform. Before using it, we need to import the library and set the required parameters for connecting to the IoT platform, including the server IP address, the username of the IoT platform account, and the corresponding password. After setting up the parameters, we can use the `init()` function to establish the connection and verify the credentials. Once the verification is successful, we can use the `connect()` and `loop()` functions to establish and maintain the connection with the platform.

```

import siot # Import siot library for SloT communication

SERVER = "10.1.2.3" # MQTT server IP, enter your actual IP
CLIENT_ID = "" # On the SloT platform, CLIENT_ID can be left empty
IOT_UserName = 'siot' # Username
IOT_PassWord = 'dfrobot' # Password
IOT_pubTopic = 'Smart Agriculture IoT System/Soil_moisture_value' # Moisture topic, "Project name/Device name"

siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_PassWord) # Initialize SloT with the provided credentials
  
```

```
siot.connect() # Connect to the SloT platform
siot.loop() # Enter the SloT loop
```

Here, SERVER refers to the IP address of the server used to connect to the SloT service; CLIENT_ID refers to the client identifier, which is a unique identifier for the client's connection to the server, and can be left blank in this case; user refers to the username of the IoT platform account; password refers to the corresponding password for the account.

5. The publish() function sends messages to the IoT platform

After connecting to the SloT IoT platform, the publish() function in the SloT library can be used to send messages to the platform. To use it, you need to specify the topic (destination) and the content of the message to be sent.

```
IOT_pubTopic = 'Smart Agriculture IoT System/Soil_moisture_value' # Moisture topic, "Project name/Device name"
    Soil_moisture_value = adc0.read_analog() # Read the analog value
    siot.publish(IOT_pubTopic, Soil_moisture_value) # # Subscribe to the topic
```

In this case, IOT_pubTopic refers to the topic, which includes the project ID and device name; Soil_moisture_value refers to the message containing the soil moisture data to be sent.

Tips: On the SloT platform, the topic is used to represent both the project ID and device name. Therefore, it is important to separate them using the "/" symbol in the topic. Otherwise, the IoT platform won't be able to recognize the project correctly.

(1) The subscribe() function is used to subscribe to messages

from the IoT platform. After connecting to the SloT IoT platform, you can use the subscribe() function from the SloT library to subscribe to messages sent by the platform. When using this function, you need to specify the topic (destination) to subscribe to and the actions to be performed upon receiving messages from the platform.

```
IOT_pubTopic = 'SmartAgricultureIoTSystem/Soil_moisture_value' # Moisture topic, "Project name/Device name"

def sub_relay(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode()
    '''Define actions when receiving commands'''
    print("\nTopic:" + topic +
        " Message:" + payload) # Print the received information
    if payload == 'on': # If received "on"
        img.config(w=240, h=320, image='img/water1.png')
        relay.write_digital(1) # Set relay to high level
    elif payload == 'off': # If received "off"
```



```
img.config(w=240, h=320, image='img/stop1.png')  
relay.write_digital(0) # Set relay to low level  
siot.subscribe(IOT_pubTopic, sub_relay) # Subscribe to the topic
```

IOT_pubTopic refers to the topic, which includes the project ID and device name. sub_relay is the defined function to be executed upon receiving messages from the platform. msg.topic refers to the topic of the message, while msg.payload refers to the content of the message. decode() is used to convert the message content into a string for easier subsequent condition checking.

Hands-on practice

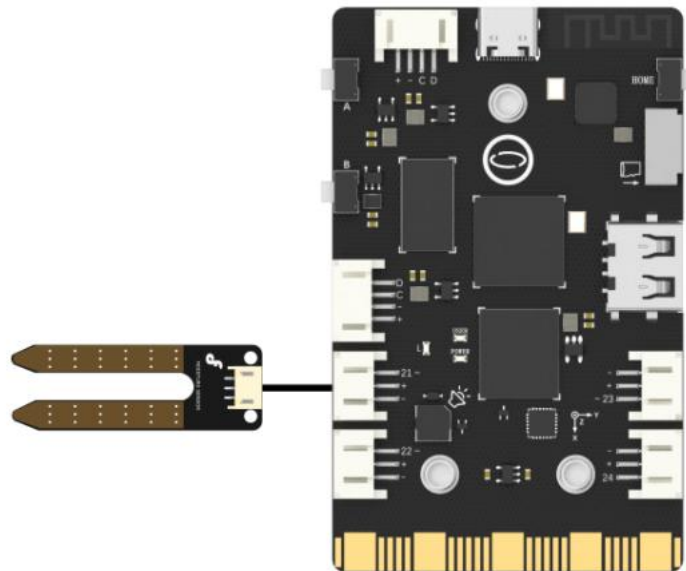
Task Description 1: Sending Humidity Data to SloT IoT Platform

Start the SloT IoT system and use an external soil moisture sensor to detect the humidity value. Display the data on the screen and simultaneously send it to the SloT IoT platform for viewing on the platform's web interface.

1. Hardware setup

STEP 1: Connect the UNIHAKER to the computer via a USB cable.

STEP 2: Connect the soil moisture sensor to pin P21 of the UNIHAKER.

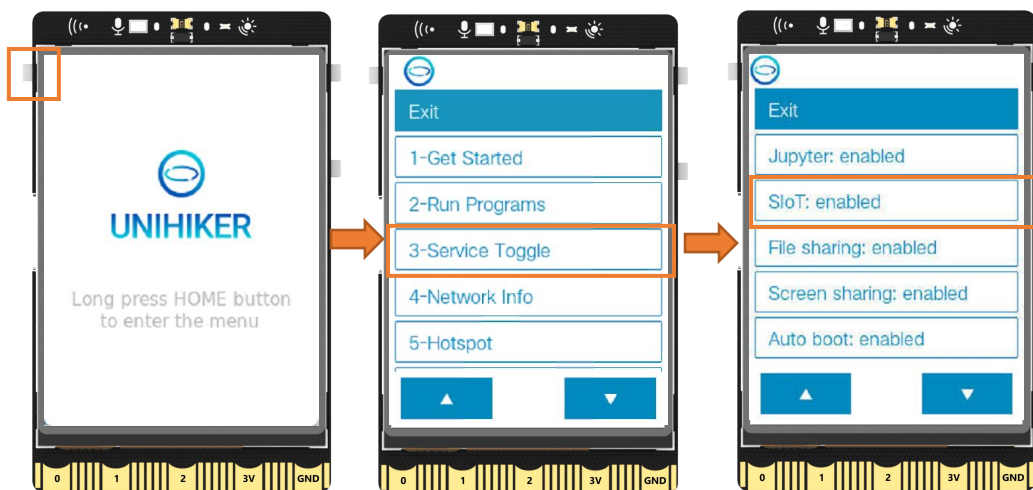


STEP 3: Insert the soil moisture sensor into the flower pot.



STEP 4: Start SloT service

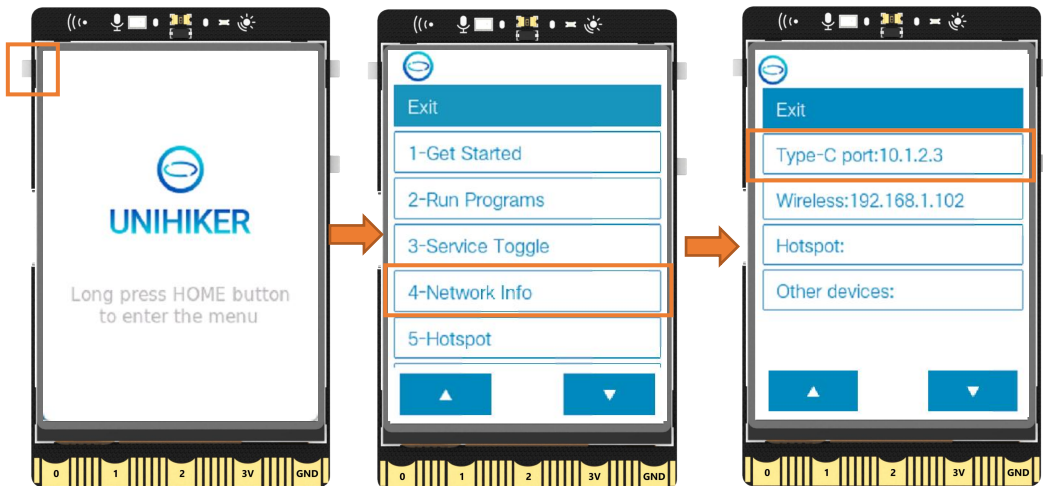
- (1) Press the HOME button to access the menu.
- (2) Click on "Service Toggle" to open the application switcher.
- (3) Locate the SloT and click on it to enable it.



STEP 5: View and record the IP address

- (1) Press the HOME button to access the menu.
- (2) Navigate to the menu option labeled "Network Info" .
- (3) Click on the "Network Info" option.

Tips: "10.1.2.3" is the fixed IP address assigned to the board when it is connected to the computer via a USB cable. It is recommended to make a note of this IP address as it will be useful for future steps or configurations.



2. program coding

STEP1: Creating and Saving Project Files

Launch Mind+ and save the project as "009 Smart Agriculture IoT System".

STEP2: Creating and Saving Python Files

Create a Python program file named "main1.py" and double-click to open it.

STEP3: Import image folder

Import the 'img' folder into the project directory.

STEP 4: Programming

(1) Import the required libraries

In this task, we need to use the Pinpong library to read values from the soil moisture sensor and display them on the screen. Therefore, we need to import the Pinpong library and the GUI module from the UNIHiker library. Additionally, to send the data to the SLoT IoT platform in real-time, we also need to import the SLoT library.

```
from unihiker import GUI # Import GUI module from unihiker library
from pinpong.board import Board, Pin # Import Board and Pin modules from pinpong library
import time # Import time library
import siot # Import siot library for SLoT communication
```

(2) Instantiate the GUI class and initialize the board and pins

Next, we create a GUI object and initialize the UNIHiker and pins to enable the use of screen functionality and control the sensor.

```
gui = GUI() # Instantiate the GUI object

Board().begin() # Initialize the UNIHiker board
```

```
adc0 = Pin(Pin.P21, Pin.ANALOG) # Initialize pin 21 as analog input mode
```

(3) Set parameters and connect to the SloT platform

Next, we will establish a connection with the SloT IoT platform. Before that, we need to set the required parameters for the connection, including the IP address, CLIENT_ID, username, password, and topic. After setting the parameters, we can establish the connection and keep it active.

Tips: There should be no spaces between the words in the project name.

```
SERVER = "10.1.2.3" # MQTT server IP, enter your actual IP
CLIENT_ID = "" # On the SloT platform, CLIENT_ID can be left empty
IOT_UserName = 'siot' # Username
IOT_PassWord = 'dfrobot' # Password
IOT_pubTopic = 'SmartAgricultureIoTSystem/Soil_moisture_value' # Moisture topic, "Project name/Device name"

siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_PassWord) # Initialize SloT with the provided credentials
siot.connect() # Connect to the SloT platform
siot.loop() # Enter the SloT loop
```

(4) Display background image, fill rectangle, and initial text

Next, we will display a background image on the screen of the board and add a filled rectangle and initial text. This will provide a foundation for updating and displaying data later on.

```
# Display background image
img = gui.draw_image(w=240, h=320, image='img/stop1.png')

gui.fill_rect(x=45, y=35, w=95, h=30, color="white") # Draw a rectangle for humidity
gui.fill_rect(x=148, y=35, w=55, h=30, color="white") # Draw a rectangle to display the humidity value

text_1 = gui.draw_text(x=48, y=36, color="orange", text='humidity:') # Display humidity
text_value = gui.draw_text(x=155, y=36, color="orange", text="") # Display the humidity value
```

(5) Detect soil moisture and display data on the screen while sending it to the SloT IoT platform

Finally, we will set up the code to read the sensor data for soil moisture, display the moisture value on the screen, and send it to the IoT platform. To continuously perform these operations every second, we will use a while loop.

```
while True:
    Soil_moisture_value = adc0.read_analog() # Read the analog value
```

```
print(Soil_moisture_value) # Print the humidity value
siot.publish(IOT_pubTopic, Soil_moisture_value) # Publish the information to the SloT platform
text_value.config(text=Soil_moisture_value) # Update the humidity value
time.sleep(1) # Delay for 1 second
```

Tips: The complete example program is as follows:

```
from unihiker import GUI # Import GUI module from unihiker library
from pinpong.board import Board, Pin # Import Board and Pin modules from pinpong library
import time # Import time library
import siot # Import siot library for SloT communication

gui = GUI() # Instantiate the GUI object

Board().begin() # Initialize the UniHiker board
adc0 = Pin(Pin.P21, Pin.ANALOG) # Initialize pin 21 as analog input mode

SERVER = "10.1.2.3" # MQTT server IP, enter your actual IP
CLIENT_ID = "" # On the SloT platform, CLIENT_ID can be left empty
IOT_UserName = 'siot' # Username
IOT_Password = 'dfrobot' # Password
IOT_pubTopic = 'Smart Agriculture IoT System/Soil_moisture_value' # Moisture topic, "Project name/Device name"

siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_Password) # Initialize SloT with the provided credentials
siot.connect() # Connect to the SloT platform
siot.loop() # Enter the SloT loop

# Display background image
```

```

img = gui.draw_image(w=240, h=320, image='img/stop1.png')

gui.fill_rect(x=45, y=35, w=95, h=30, color="white") # Draw a rectangle for humidity
gui.fill_rect(x=148, y=35, w=55, h=30, color="white") # Draw a rectangle to display the humidity value

text_1 = gui.draw_text(x=48, y=36, color="orange", text='humidity:') # Display humidity
text_value = gui.draw_text(x=155, y=36, color="orange", text="") # Display the humidity value

while True:

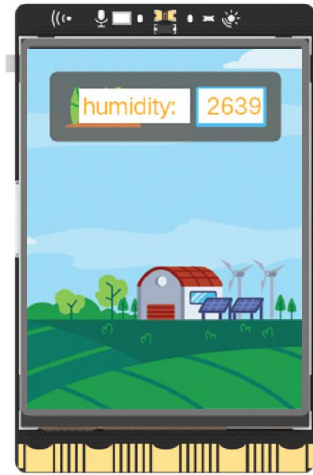
    Soil_moisture_value = adc0.read_analog() # Read the analog value
    print(Soil_moisture_value) # Print the humidity value
    siot.publish(IOT_pubTopic, Soil_moisture_value) # Publish the information to the SloT platform
    text_value.config(text=Soil_moisture_value) # Update the humidity value
    time.sleep(1) # Delay for 1 second

```

3. Running the Program

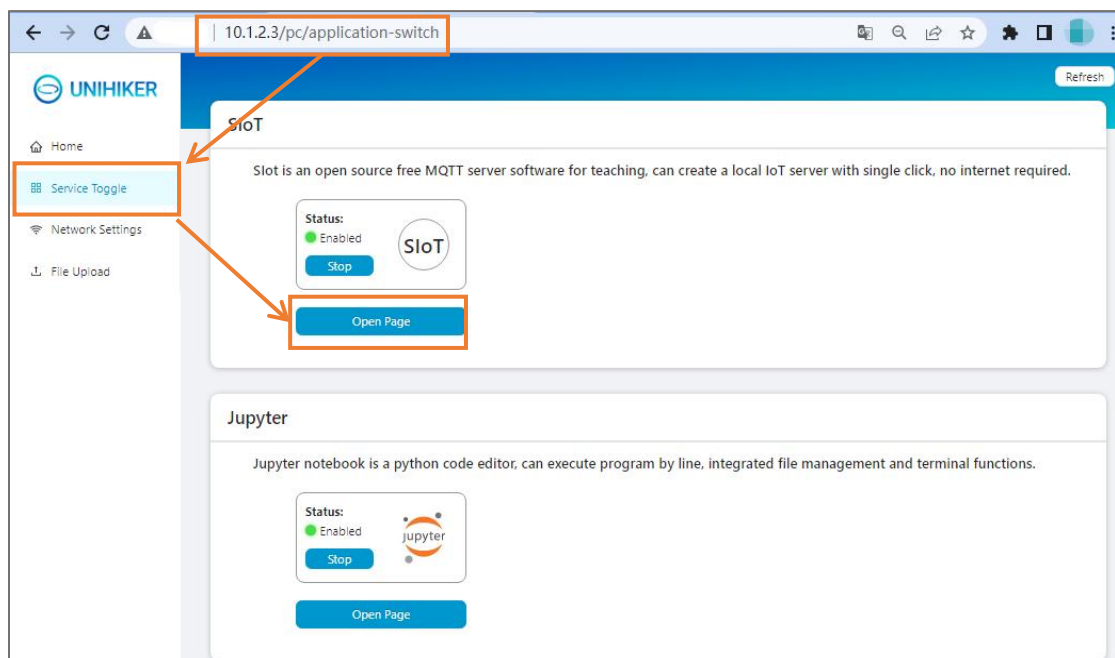
STEP 1: Remote Connect to the Rowboard, Run the Program, and Observe the Results

Observe the UNIHAKER, and you will see that on the background image of smart agriculture, the detected soil moisture values are displayed every second.



STEP 2: View Messages on the Platform Web Page

- (1) Open a web browser and enter "10.1.2.3", then press Enter to access the Rowboard's web menu. Find the Slot service under the application switch and click "Open Page".



- (2) Enter the default account "siot" and password "dfrobot", then click "Login".

SIoT

Project list

Device List

Send message

ID(default:siot)

siot

Password(default:dfrobot)

.....

Login

SIoT

Project list

Device List

Send message

Page after logging in.

siot Logout

Topic(ProjectID/DeviceName)

ProjectID

100 pieces

Search

ProjectID

Tips

Command

SmartAgricultureIoT System

View device list

Add tips

delete

(3) Click on "Device List" under the "Smart Agriculture IoT System" project.

SIoT

Project list

Device List

Send message

siot Logout

Topic(ProjectID/DeviceName)

ProjectID

100 pieces

Search

ProjectID

Tips

Command

SmartAgricultureIoT System

View device list

Add tips

delete

You can see that the "Project ID" and "Name" correspond to the information specified in the program as "IOT_pubTopic."

SIoT

Project List

Device list

Send message

siot Logout

SmartAgricultureIoTSystem's device

ProjectID

DeviceName

100 pieces

Search

ProjectID

Name

Tips

Command

SmartAgricultureIoTSystem

Soil_moisture_value

View message

Clear messages

Delete device

Add Tips

(4) Click on "View Messages."

SIoT

Project List

Device list

Send message

siot Logout

SmartAgricultureIoTSystem's device

ProjectID

DeviceName

100 pieces

Search

ProjectID	Name	Tips	Command
SmartAgricultureIoTSystem	Soil_moisture_value	View message	Clear messages Delete device Add Tips

Then we can view the MQTT message records for the topic "Smart Agriculture IoT System/Soil_moisture_value". The records will include the message content, as well as the time it was sent. This represents the soil moisture value detected at that moment. See the image below.

SIoT

Project List

Device list

Send message

Topic : SmartAgricultureIoTSystem/Soil_moisture_value

Send message

Message

Send

(Adding a -> prefix to the message means that the message is a pure instruction message and will not be stored in the database. For example "->off")

Starting time

End time

100 pieces

Search

Export Excel

Hide/Show chart

Automatically refresh messages ☐

Topic	message	time
SmartAgricultureIoTSystem/Soil_moisture_value	2733	2023-05-17 16:39:33
SmartAgricultureIoTSystem/Soil_moisture_value	2715	2023-05-17 16:39:32
SmartAgricultureIoTSystem/Soil_moisture_value	2711	2023-05-17 16:39:31
SmartAgricultureIoTSystem/Soil_moisture_value	2703	2023-05-17 16:39:30

(5) Click on "Automatically Refresh Messages"

SIoT

Project List

Device list

Send message

Topic : SmartAgricultureIoTSystem/Soil_moisture_value

Send message

Message

Send

(Adding a -> prefix to the message means that the message is a pure instruction message and will not be stored in the database. For example "->off")

Starting time

End time

100 pieces

Search

Export Excel

Hide/Show chart

Automatically refresh messages ☒

Topic	message	time
SmartAgricultureIoTSystem/Soil_moisture_value	2733	2023-05-17 16:39:33
SmartAgricultureIoTSystem/Soil_moisture_value	2715	2023-05-17 16:39:32
SmartAgricultureIoTSystem/Soil_moisture_value	2711	2023-05-17 16:39:31

By refreshing the messages, we can see that the soil moisture data is being updated every second, as shown in the following image.

SIoT	Project List	Device list	Send message
------	--------------	-------------	--------------

Topic : SmartAgricultureIoTSystem/Soil_moisture_value		
Send message	Message	Send

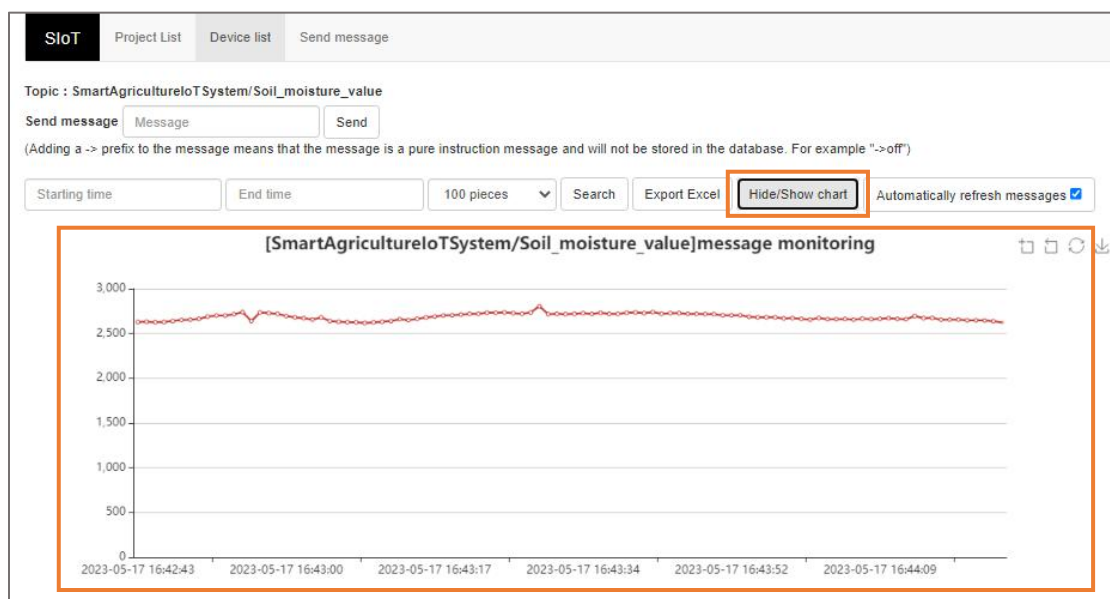
(Adding a -> prefix to the message means that the message is a pure instruction message and will not be stored in the database. For example "->off")

Starting time	End time	100 pieces	Search	Export Excel	Hide/Show chart	Automatically refresh messages <input checked="" type="checkbox"/>
---------------	----------	------------	--------	--------------	-----------------	--

Topic	message	time
SmartAgricultureIoTSystem/Soil_moisture_value	2733	2023-05-17 16:39:33
SmartAgricultureIoTSystem/Soil_moisture_value	2715	2023-05-17 16:39:32
SmartAgricultureIoTSystem/Soil_moisture_value	2711	2023-05-17 16:39:31

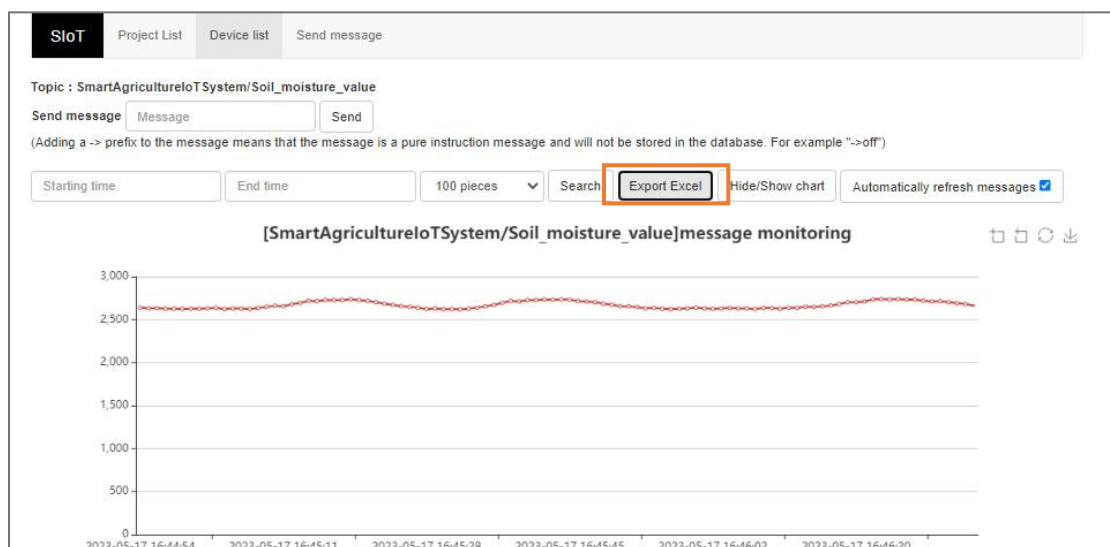
(6) By clicking on the "Show/Hide Chart" icon

we can see that the data is presented in the form of a chart, as shown in the following image.



(7) By clicking on the "Export Excel" button

we can download the data in tabular form for further analysis.

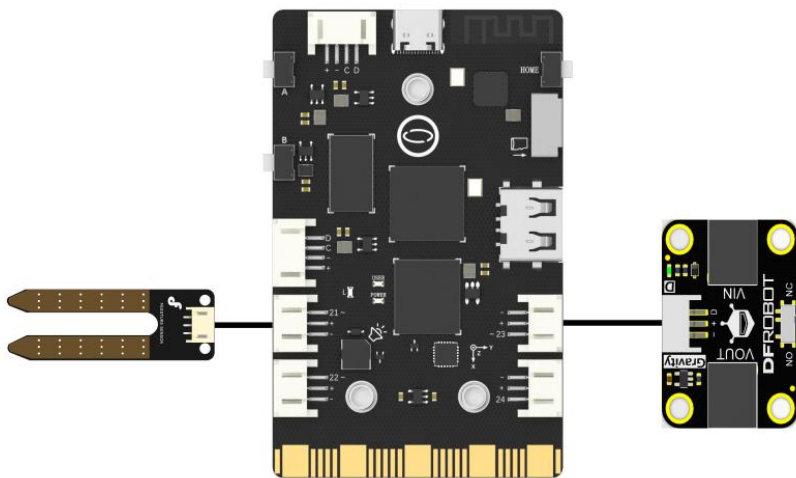


Task Description 2: Controlling Irrigation via the Platform Web Page

In the previous task, we successfully uploaded the soil moisture data from the sensor to the SloT IoT platform and monitored the humidity values in real-time on the platform's web page. Now, we will add the functionality of subscribing to platform messages on the board, allowing us to control irrigation in a timely manner through the SloT IoT platform's web page when the soil is dry.

1. Hardware setup

STEP 1: Connect the relay to Pin P23 on the UNIHAKER.



STEP 2: Use a screwdriver to connect the positive and negative wires of the water pump to the adapter. Follow the steps and illustration below:

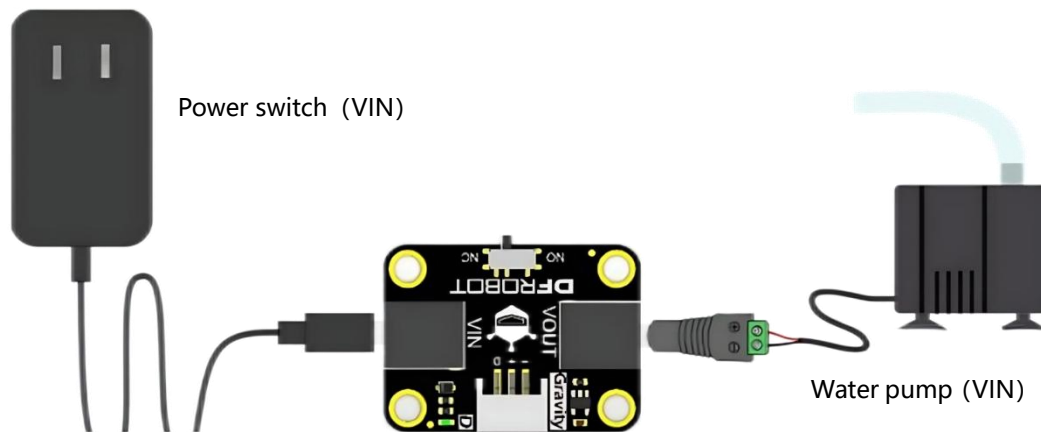
- (1) Loosen the screws on the adapter.
- (2) Insert the wires, paying attention to the polarity (brown/red - positive; blue/black - negative).
- (3) Tighten the screws.

The red/brown wire is connected to the positive terminal.

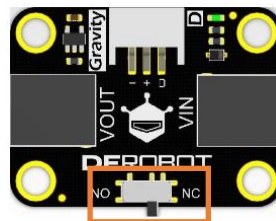


The blue/black wire is connected to the negative terminal.

STEP 3: Connect the 12V power switch to the water pump's adapter using a relay.



STEP 4: Set the relay switch to the NC (Normally Closed) terminal.



Switch to the NC (Normally Closed) terminal.

STEP 5: Secure the water pump in a filled beaker

Tips: The water pump should not run dry. Make sure to immerse the black pump head in water to prevent potential hardware damage.



STEP 6: Insert the water pipe and soil moisture sensor into the flowerpot.



2. program coding

STEP1: Creating and Saving Python Files

Create a Python program file named "main2.py" and double-click to open it.

STEP 2: Programming

(1) Initialize the Pins

Since we will be using a relay to control the water pump for irrigation, we need to add the initialization code for the relay pins. Set Pin 23 as a digital output pin.

```
relay = Pin(Pin.P23, Pin.OUT) # Initialize pin 23 as a digital output mode.
```

(2) Define the Operation upon Receiving Platform Messages

Since we will be receiving messages from the platform and controlling the watering based on the received messages, we need to define a callback function to specify the actions to be taken upon receiving a message. In this case, when the message is "on," we will activate the relay to turn on the water pump, and when the message is "off," we will deactivate the relay to stop the water pump.

```
def sub_relay(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode()
    """Define actions when receiving commands"""
    print("\nTopic:" + topic +
          " Message:" + payload) # Print the received information
    if payload == 'on': # If received "on"
        img.config(w=240, h=320, image='img/water1.png')
        relay.write_digital(1) # Set relay to high level
    elif payload == 'off': # If received "off"
        img.config(w=240, h=320, image='img/stop1.png')
```

```
relay.write_digital(0) # Set relay to low level
```

(3) Subscribe to Platform Messages

To receive messages from the IoT platform, we need to subscribe to the appropriate topic. After connecting to the platform, we can add the subscription code to ensure that we receive the desired messages.

```
siot.subscribe(IOT_pubTopic, sub_relay) # Subscribe to our newsletter
```

Tips: The complete example program is as follows:

```
from unihiker import GUI # Import GUI module from unihiker library

from pinpong.board import Board, Pin # Import Board and Pin modules from pinpong library

import time # Import time library

import siot # Import siot library for SloT communication


gui = GUI() # Instantiate the GUI object


Board().begin() # Initialize the UniHiker board

adc0 = Pin(Pin.P21, Pin.ANALOG) # Initialize pin 21 as analog input mode

relay = Pin(Pin.P23, Pin.OUT) # Initialize pin 23 as a digital output mode.


SERVER = "10.1.2.3" # MQTT server IP, enter your actual IP

CLIENT_ID = "" # On the SloT platform, CLIENT_ID can be left empty

IOT_UserName = 'siot' # Username

IOT_Password = 'dfrobot' # Password

IOT_pubTopic = 'SmartAgricultureIoTSystem/Soil_moisture_value' # Moisture topic, "Project name/Device name"


def sub_relay(client, userdata, msg):

    topic = msg.topic

    payload = msg.payload.decode()

    '''Define actions when receiving commands'''

    print("\nTopic:" + topic +

          " Message:" + payload) # Print the received information
```

```

if payload == 'on': # If received "on"

    img.config(w=240, h=320, image='img/water1.png')

    relay.write_digital(1) # Set relay to high level

elif payload == 'off': # If received "off"

    img.config(w=240, h=320, image='img/stop1.png')

    relay.write_digital(0) # Set relay to low level


siot.init(CLIENT_ID, SERVER, user=IOT_UserName, password=IOT_PassWord) # Initialize, make
sure the username and password are correct

siot.connect() # Connect to the SloT platform

siot.subscribe(IOT_pubTopic, sub_relay) # Subscribe to the topic

siot.loop() # Loop


# Display background image

img = gui.draw_image(w=240, h=320, image='img/stop1.png')


gui.fill_rect(x=45, y=35, w=95, h=30, color="white") # Draw a rectangle for humidity

gui.fill_rect(x=148, y=35, w=55, h=30, color="white") # Draw a rectangle to display the humidity value

text_1 = gui.draw_text(x=48, y=36, color="orange", text='humidity:') # Display humidity

text_value = gui.draw_text(x=155, y=36, color="orange", text="") # Display the humidity value


while True:

    Soil_moisture_value = adc0.read_analog() # Read the analog value

    #print(Soil_moisture_value) # Print the humidity value

    siot.publish(IOT_pubTopic, Soil_moisture_value) # Publish the information to the SloT platform

    text_value.config(text=Soil_moisture_value) # Update the humidity value

```



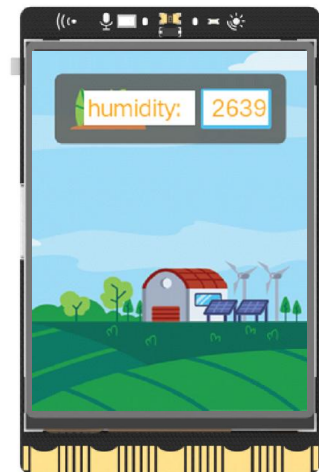
```
time.sleep(1) # Delay for 1 second
```

3. Running the Program

STEP 1: Plug the 12V power switch into a power outlet.

STEP 2: Remote Connect to the UNIHAKER, Run the Program, and Observe the Results

Observing the board, you can see that the temperature and humidity values are continuously updated and displayed on the screen in real-time.



STEP 3: Sending Messages from the Platform

Observing the detected humidity data, when the humidity value is too low, you can input "on" in the specified field on the web page and click on "Send". You will notice that the relay starts working, controlling the pump to continuously draw water from the beaker into the pot.

SIoT Project List Device list Send message

Topic : SmartAgricultureIoTSystem/Soil_moisture_value

Send message on Send

(Adding a -> prefix to the message means that the message is a pure instruction message and will not be stored in the database. For example "->off")

Starting time End time 100 pieces Search Export Excel Hide/Show chart Automatically refresh messages ☐

Topic	message	time
SmartAgricultureIoTSystem/Soil_moisture_value	on	2023-05-18 15:40:15
SmartAgricultureIoTSystem/Soil_moisture_value	1	2023-05-18 15:40:14
SmartAgricultureIoTSystem/Soil_moisture_value	0	2023-05-18 15:40:13
SmartAgricultureIoTSystem/Soil_moisture_value	0	2023-05-18 15:40:12

Send successfully



Afterwards, when the water level is appropriate, we can enter "off" in the message box and click send. We will then observe that the water pump stops.

SloT

Project List

Device list

Send message

Topic : SmartAgricultureIoTSystem/Soil_moisture_value

Send message:

(Adding a -> prefix to the message means that the message is a pure instruction message and will not be stored in the database. For example "->off")

☐ Automatically refresh messages

Topic	message	time
SmartAgricultureIoTSystem/Soil_moisture_value	off	2023-05-18 15:41:23
SmartAgricultureIoTSystem/Soil_moisture_value	1	2023-05-18 15:41:22
SmartAgricultureIoTSystem/Soil_moisture_value	0	2023-05-18 15:41:21
SmartAgricultureIoTSystem/Soil_moisture_value	0	2023-05-18 15:41:20
SmartAgricultureIoTSystem/Soil_moisture_value	0	2023-05-18 15:41:19

Send successfully



Challenge Yourself

Take a moment to think about what other factors can affect crop growth in an agricultural system. Are there any methods available to detect and monitor these factors? Take some time to explore and conduct research on your own, and try to analyze the impact of these factors on crops using remote methods, in conjunction with an IoT platform.