ACEBOTT

ESP32 Smart Weather Station Kit

Perface

Our Company

ACEBOTT STEM Education Tech Co.,Ltd
Founded in China's Silicon Valley in 2013, ACEBOTT is a STEM education solution leader We have a team of 150 individuals, including members from research and development, sales, and logistics. Our goal is to provide high-quality STEM education products and services to our customers. We are working together with STEM education experts and our business partners to produce successful STE products together Our self-own factory also provides CEM services for our clients including logo customization on product packaging and PCB.

Our Tutorial

This course and smart home learning kit is designed for 8+ children and teenagers to learn more about ESP32 board and meteorological knowledge, sensors and circuit components. If you like to learn ESP32 meteorological knowledge, this kit could provide you the knowledge and steps to build your own smart weather station together with Dr.Lumi.

Through this kit, you can:

1.Learn how to effectively use the ESP32 board, including uploading code, understanding its features, and coding with the ARDUINO IDE.
2.Gain a solid foundation in the basics of the C language, as the ESP32 utilizes a simplified C/C++ programming language for controlling circuits and sensors.
3.Explore various electronic components such as LEDs, sensors, and motors, and understand how they work together in real-world smart weather station projects.
4.Enhance your maker skills by building your own smart weather station using the ACEBOTT kit, following step-by-step tutorials.
5.Implement various weather measurements in the smart weather station project and master the usage of different sensors and other basic functions.

Overall, the ACEBOTT Smart Weather Station Learning Kit is specifically designed for beginners to learn weather knowledge based on the ESP32. Using this kit, users can gain a comprehensive understanding of the role of the control board and sensors in weather monitoring. By following the provided tutorials, students of different age groups can acquire valuable knowledge about meteorological detection and successfully build their own weather monitoring projects.

Customer service

ACEBOTT is a dynamic and fast-growing STEM education technology company that strives to offer excellent products and quality services that meet your expectations. We value your feedback and encourage you to drop us a line at support@acebott.com with any comments or suggestions you may have.

Our experienced engineers are dedicated to promptly addressing any problems or questions you may have about our products. We guarantee a response within 24 hours during business days.

Follow Us

Scan the QR codes to Follow Us for troubleshooting & the latest news.
We have a very large community that is very helpful for troubleshooting and we also have a support team at the ready to answer any questions.

ACEBOTT FB Group QR Code

YouTube QR Code

Contents

Prologue

Dr. Lumi is a wise and knowledgeable scientist, who is not only highly respected in the academic world but also has an insatiable passion for exploring the world. He loves to travel, his footprints cover the globe, delving into the mysteries of unique cultures and geographical environments. Dr. Lumi is not only passionate about exploration but also eager to use his knowledge and wisdom to help those in need. One day, Dr. Lumi and his students embarked on a new journey, arriving at a place called Future Village. This village is nestled in a remote and beautiful valley, like a paradise on earth.

The residents of Future Village have lived on this land for generations, filled with reverence and love for nature, and they also show an immense enthusiasm and hospitality to visitors. Dr. Lumi and his students were warmly welcomed by the villagers. To reciprocate the villagers' warmth, Dr. Lumi proactively asked if they had encountered any difficulties in their daily lives that he could help with to the best of his ability.

The villagers expressed that, although they rely on the gifts of nature for their livelihood, the variability of the weather often brings them trouble. Sudden heavy rains, droughts, or storms can have a severe impact on their crops and lives. After pondering for a moment, Dr. Lumi decided to take action. He planned to establish a smart weather station to help the villagers grasp the weather dynamics in advance, thus taking appropriate preventive measures to reduce the losses caused by natural disasters.

As one of Dr. Lumi's students, you will join him in embracing this challenge, assisting Dr. Lumi in the design, installation, and debugging of the weather station, ensuring that it can accurately monitor weather changes and provide real-time data.

Lesson 1: Setting up the Programming Environment

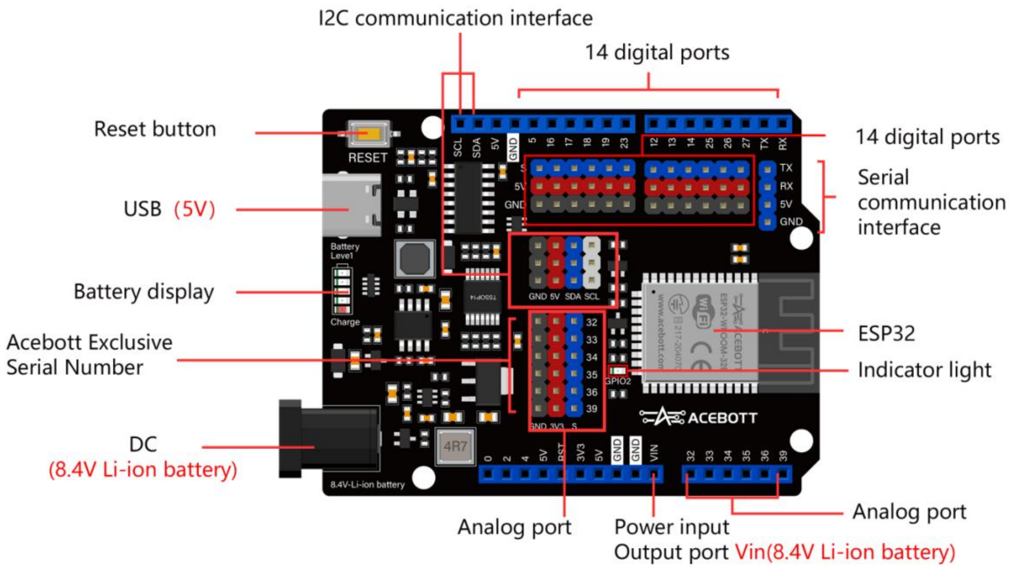
The operation of the smart weather station relies on the execution of its program. To better assist villagers in setting up the smart weather station, it is essential to first establish the programming environment and acquire some necessary foundational knowledge before starting the construction.

I. Understanding the ESP32 Controller Board

The core part of the weather station is the ESP32 controller board, which is the "brain" of the station, capable of processing various meteorological data collected by sensors and providing this data back to users.

The ESP32 controller board is a low-power, high-performance microcontroller that is well-suited for Internet of Things (IoT) development. It features a 240MHz dual-core processor, 520KB of RAM, and 4MB of flash memory. It has built-in WiFi and Bluetooth 4.2 modules for wireless communication and 34 GPIO pins for connecting and controlling various peripherals.

The ESP32 controller board used in this tutorial is configured with rechargeable capabilities. The mainboard has 5 LED modules to display the battery level. The first 4 LEDs correspond to the current battery charge. When the battery is fully charged, all 4 LEDs light up in blue, and as the charge decreases, the number of lit LEDs will also decrease accordingly. The 5th LED indicates whether the battery is charging. When it lights up in red, it means the battery is charging. When it is off, it means the battery is fully charged or not charging.



Note:

1. The controller board does not allow input of a voltage exceeding 12V, as it may cause damage to the controller board!

2. The use of dry cell batteries is not allowed, as it may cause the battery to explode! Only two 8.4V lithium batteries in series are allowed for power supply.

3. It is not recommended to touch the power components of the chip with your hands to prevent burns.

II. Setting Up the Programming Environment

For the smart weather station to operate, it is not only necessary to have essential intelligent hardware but also to write programs for the hardware to make it work and achieve various functions.

How to write programs?

The first step in writing programs is to install the programming platform. The programming platform used in this tutorial is ACECode.

1. Install Arduino IDE

Arduino IDE is a free, cross-platform programming tool used for writing, uploading, and debugging Arduino code. Arduino code is written in the C++ programming language and includes many advanced libraries and functions, which simplify hardware control and interaction, making it easier for beginners to learn and use.

Arduino IDE can run on Windows, Mac, and Linux operating systems. Here, we will focus on learning how to install the Arduino IDE on Windows and Mac.

Click here for the installation steps of Arduino IDE on Windows.

Click here for the installation steps of Arduino IDE on Mac.

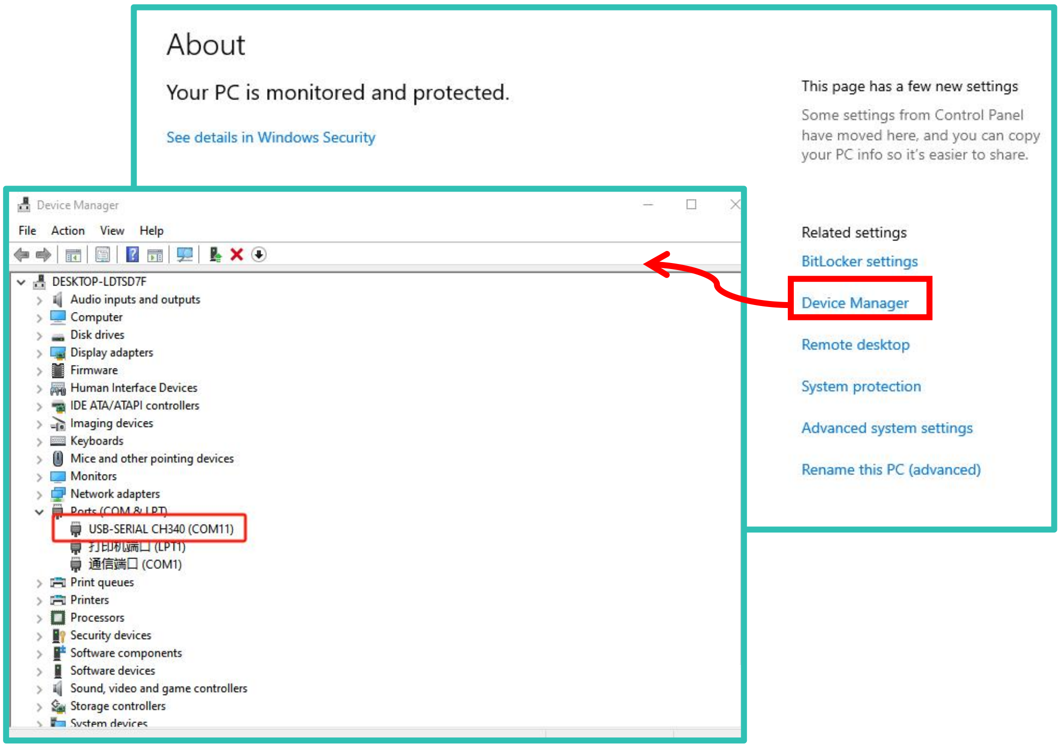
2. Install the ESP32 Development Environment

The native Arduino IDE does not include resources for ESP32. To develop for ESP32 using the Arduino IDE, you need to add an additional source to the Arduino IDE board manager and then install the ESP32 resources.

Click here for instructions on installing ESP32 resources.

3. Install the Driver

Connect the ESP32 controller to your computer using a data cable, then open Device Manager on your computer by following this path: “My Computer” -> “Properties” -> click on “Device Manager.” If the driver is installed correctly, you will find "USB-SERIAL CH340 (COM11)" under the "Ports" option. The content in parentheses indicates the port number, which may vary depending on your specific setup.



If your driver is not installed successfully, "Unknown Device" will appear under the "Other Devices" option. In this case, you will need to install the driver manually.

Click here to download the driver file.

4. Install Library Files

An Arduino library is a collection of pre-written programs that can be used to achieve specific functions by calling functions within the library. Arduino IDE comes with many built-in libraries, such as the Wire library, Servo library, etc. These libraries help us quickly develop various applications. However, sometimes the needed functions are not included in the built-in libraries, requiring you to install custom user libraries.

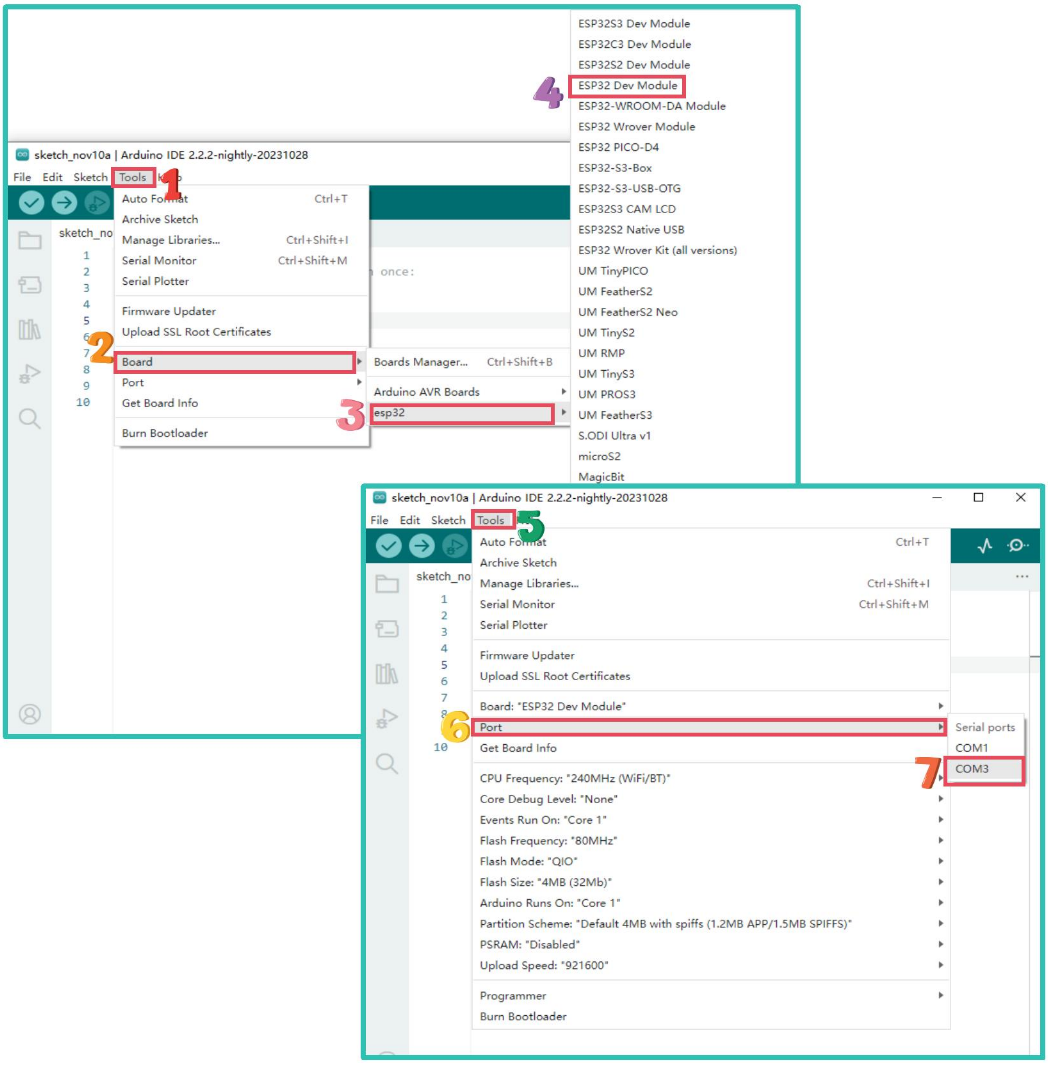
In the smart weather station project, many sensors are used, and some of them require additional library files. You need to install the necessary libraries for weather station development in advance.

Click here for instructions on installing library files.

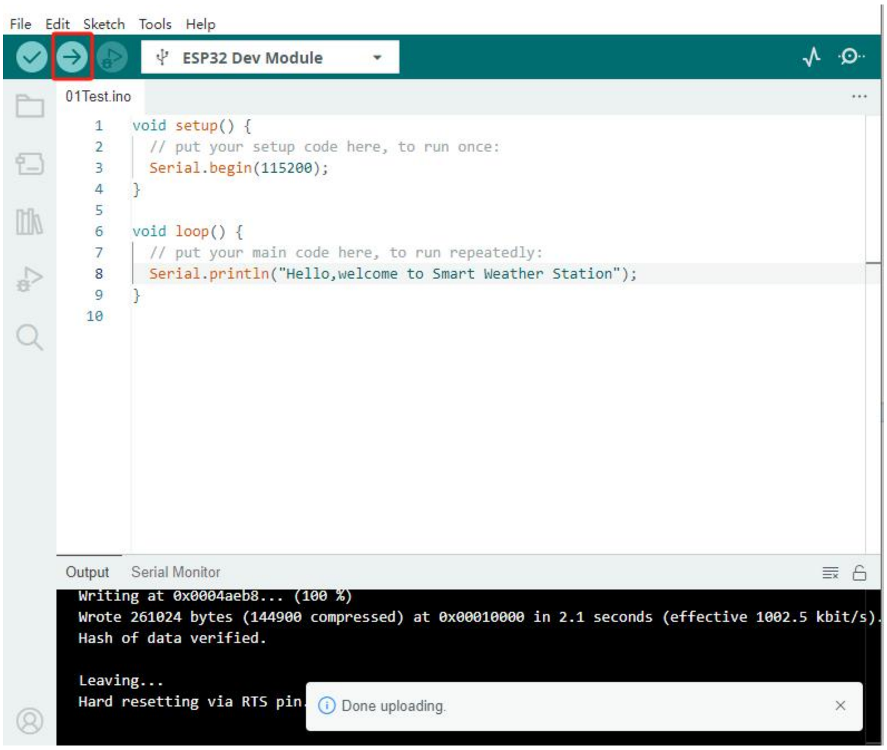
5. Program Testing

Upload a test program to the ESP32 using Arduino IDE to check if the program uploads and works correctly. Open Arduino IDE, select the corresponding ESP32 board model—ESP32 Dev Module—and choose the correct port number (which should match the ESP32 communication port number found in Device Manager).

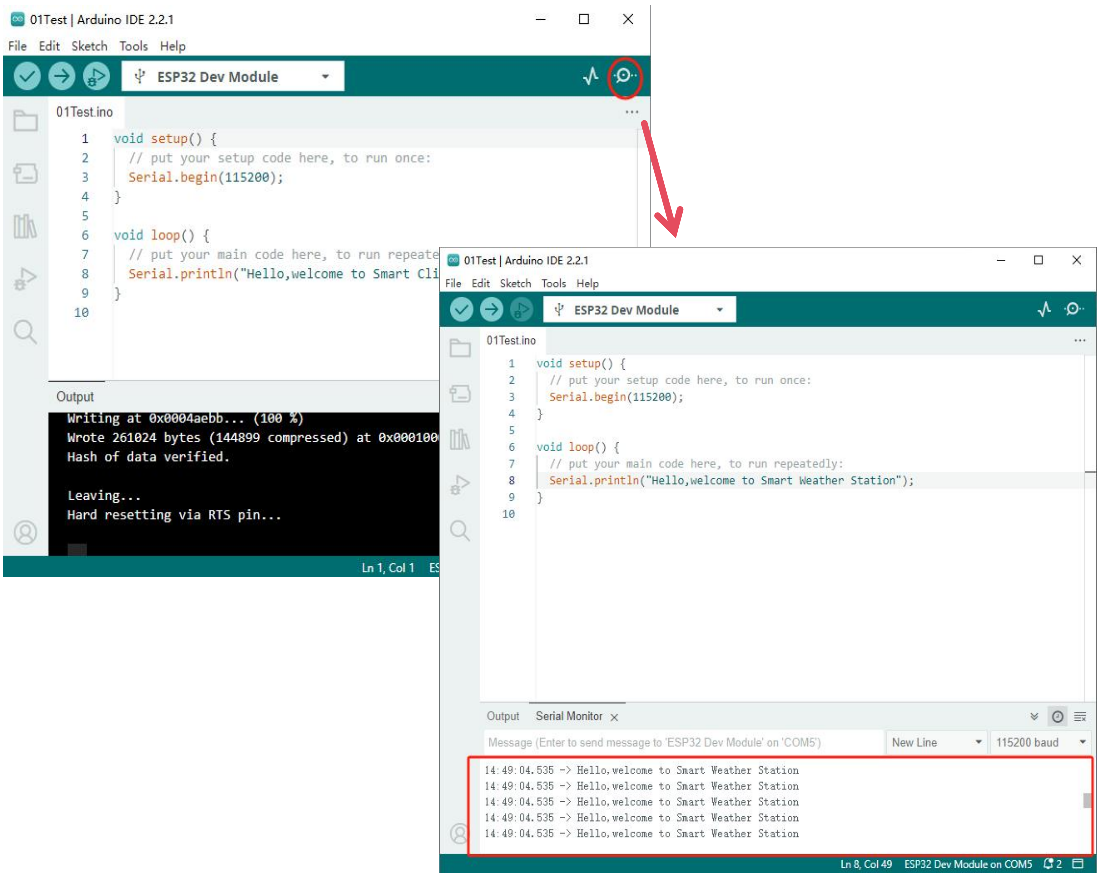
Click here to access the test program.



Click the "Upload" button to upload the program. If the upload is successful, "Done uploading" will be displayed.



Click on the Serial Monitor to observe the output.



Lesson 2: Building a Smart Weather Station

In order to assist the villagers in establishing a fully functional weather station, Dr. Lumi ventured deep into the village, visiting many residents household by household, and meticulously documenting the various problems and issues they faced. With this valuable feedback, Dr. Lumi compiled a detailed list of materials, providing the necessary preparations for building the weather station.

Next, you will need to guide the villagers step by step in building the smart weather station based on this list and the provided assembly diagram. This will be a systematic project that requires you to carefully check each item on the list and ensure that the construction follows the blueprints closely, in order to guarantee the accuracy and reliability of the weather station. Through your efforts, the villagers will have a smart weather station that can provide accurate meteorological information, enabling them to better cope with the various challenges posed by weather changes.

I. Steps for Building a Smart Weather Station

Note: If you want to watch the assembly video, please click the link below.

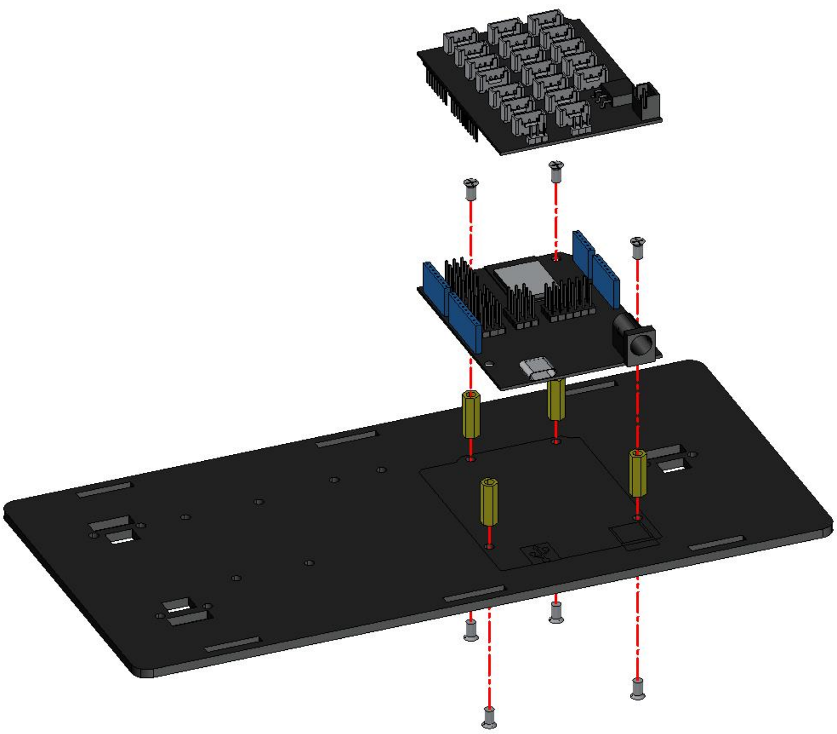
https://www.youtube.com/playlist?list=PLkW5fEtHNu6JbnSmm2qSQQ3nkMr6SF59Y

Or scan the QR code below.

1. Install the ESP32 controller board

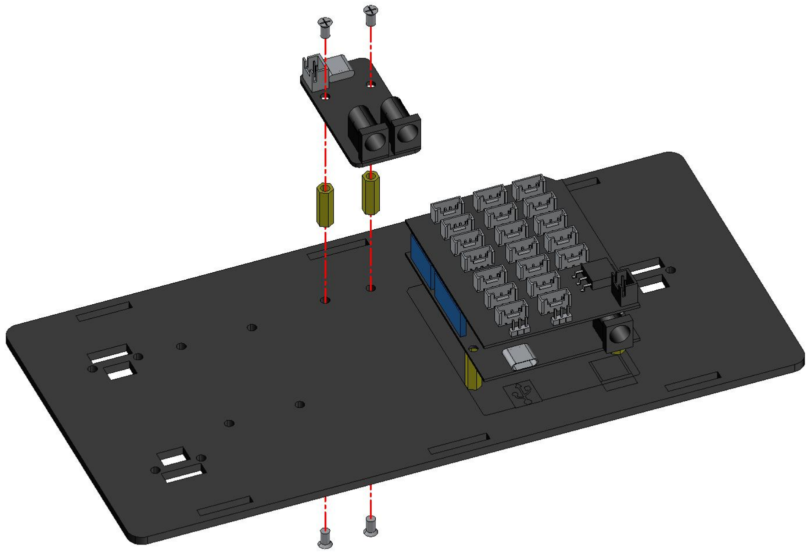
Parts List	
Name	**Quantity**
ESP32 Controller Board	1
ESP32-Shield Board	1
Smart Weather Station Baseboard	1
M3*14 MM Dual-Pass Copper Pillar	4
M3*8 MM Flat Head Screw	7

Note:Remove the protective film from all acrylic sheets before use.



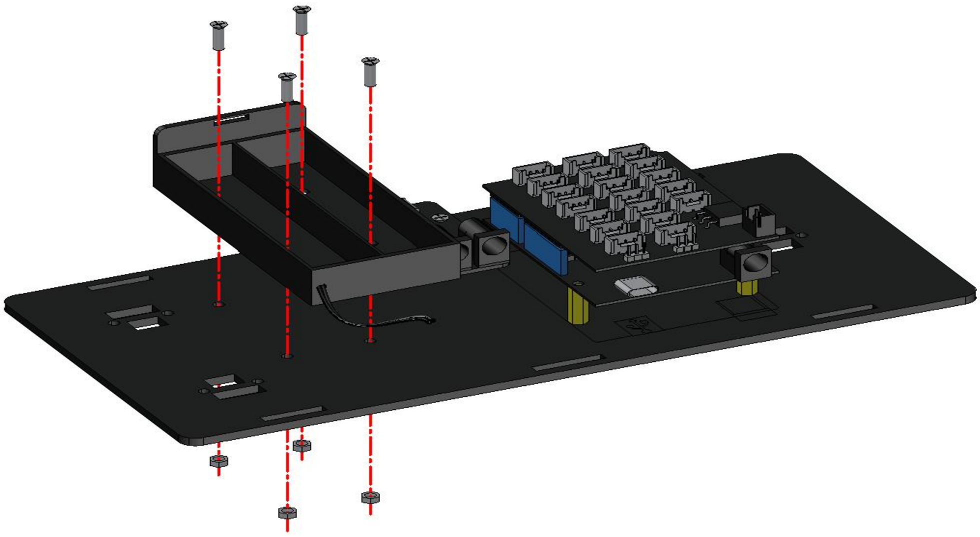
2.Install the charging module.

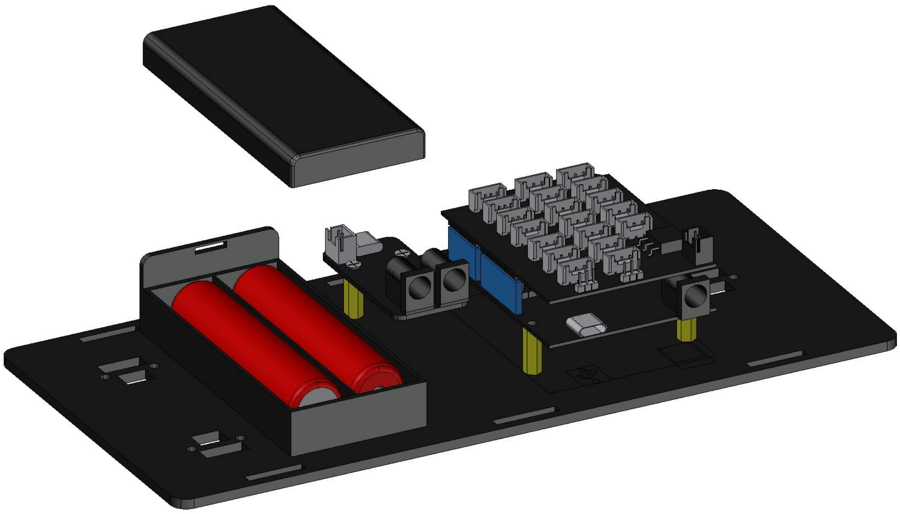
Parts List	
Name	**Quantity**
Charging Module	1
M3*14 MM Dual-Pass Copper Pillar	2
M3*8 MM Flat Head Screw	4



3.Install the battery holder.

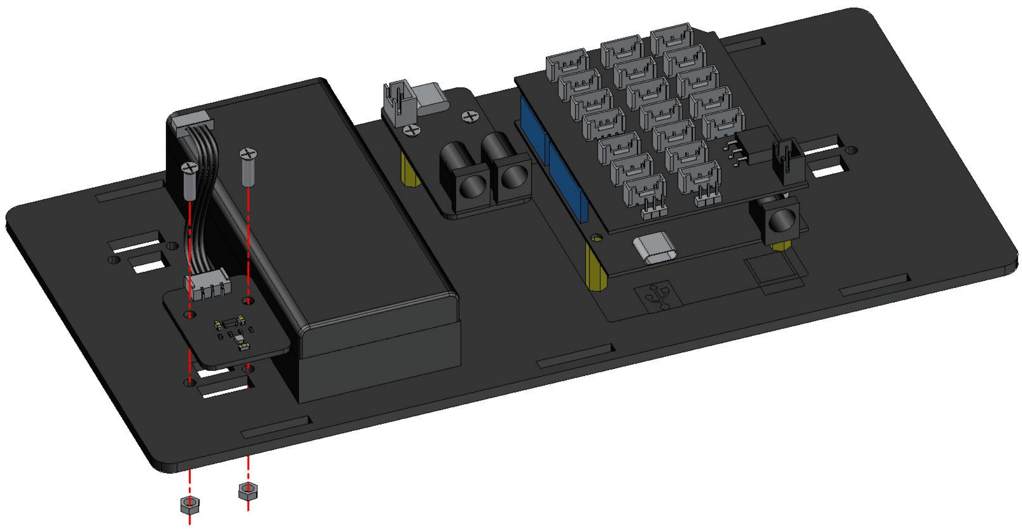
Parts List	
Name	**Quantity**
18650 Battery Holder	1
18650 Lithium Battery	2
M3*10 MM Flat Head Screw	4
M3 Nickel-Plated Nut	4





4.Install the atmospheric pressure sensor.

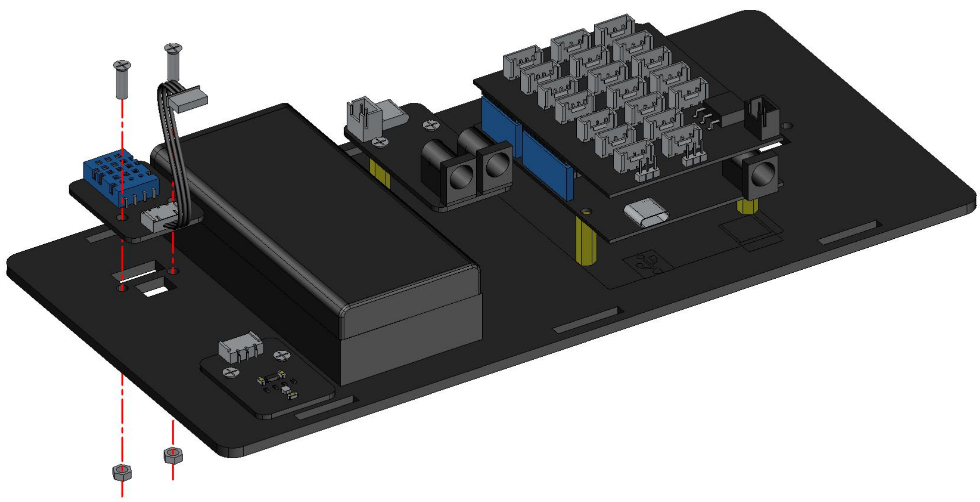
Parts List	
Name	**Quantity**
Barometric Pressure Sensor	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
44P Cable	1



Note: Before installation, connect the 4P cable to the barometric pressure sensor.

5. Install the DHT11 sensor.

Parts List	
Name	**Quantity**
DHT11 Sensor	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
3P Cable	1

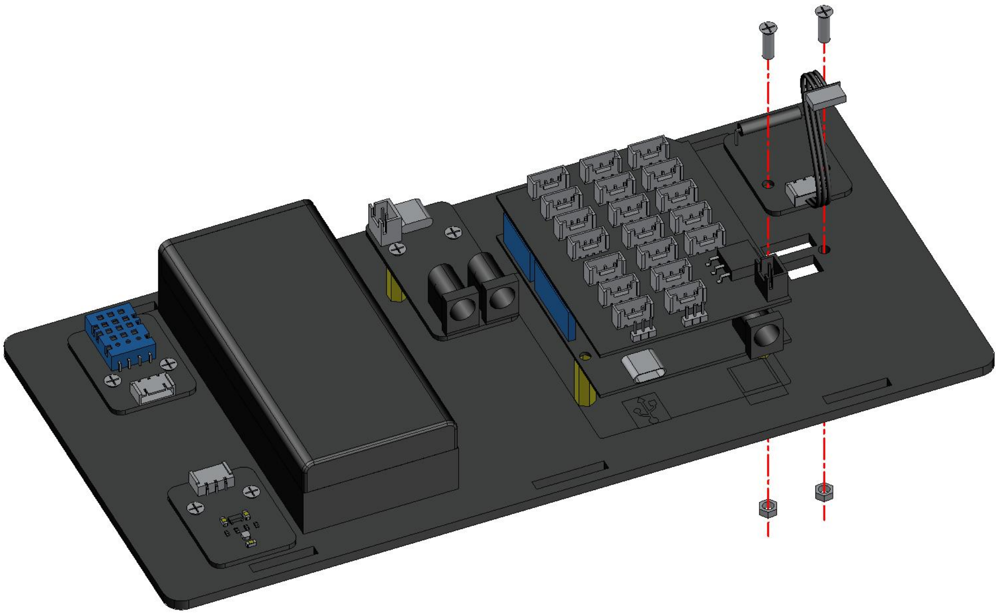


Note: Before installation, connect the 3P cable to the DHT11

sensor.

6. Install the vibration sensor.

Parts List	
Name	**Quantity**
Vibration Sensor	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
3P Cable	1



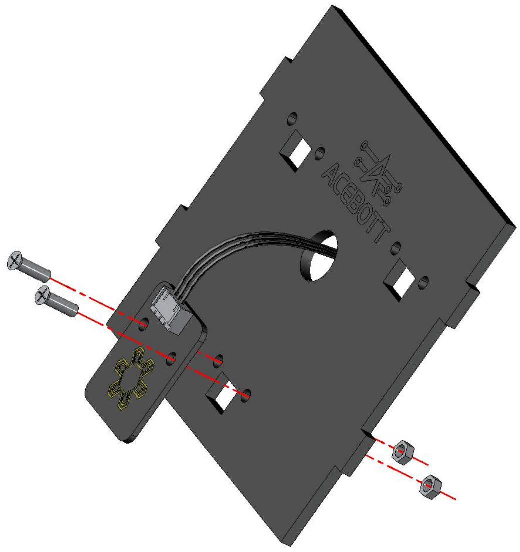
Note: Before installation, connect the 3P cable to the vibration sensor.

7. Install the raindrop sensor.

Parts List	
Name	**Quantity**
Raindrop Sensor	1
Rear Plate	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2

3P Cable	1

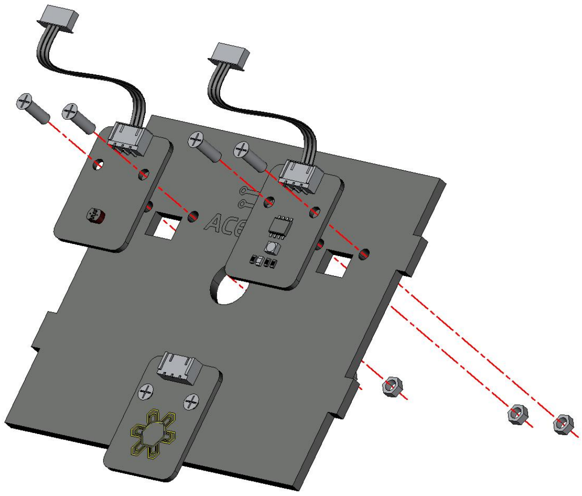
Note: Before installation, connect the 3P cable to the raindrop sensor.



8. Install the light sensor and UV sensor.

Parts List	
Name	**Quantity**
Light Sensor	1
UV Sensor	1
M3*10 MM Flat Head Screw	4
M3 Nickel-Plated Nut	4
3P Cable	2

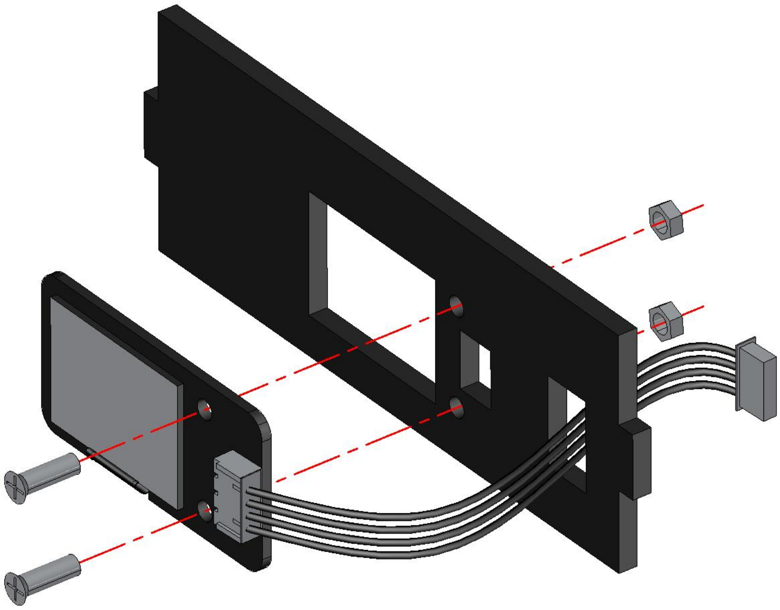
Note: Before installation, connect the 3P cable to the light sensor and the UV sensor.



9. Install the OLED module.

Parts List	
Name	**Quantity**
OLED Module	1
OLED Support Board	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
4P Cable	1

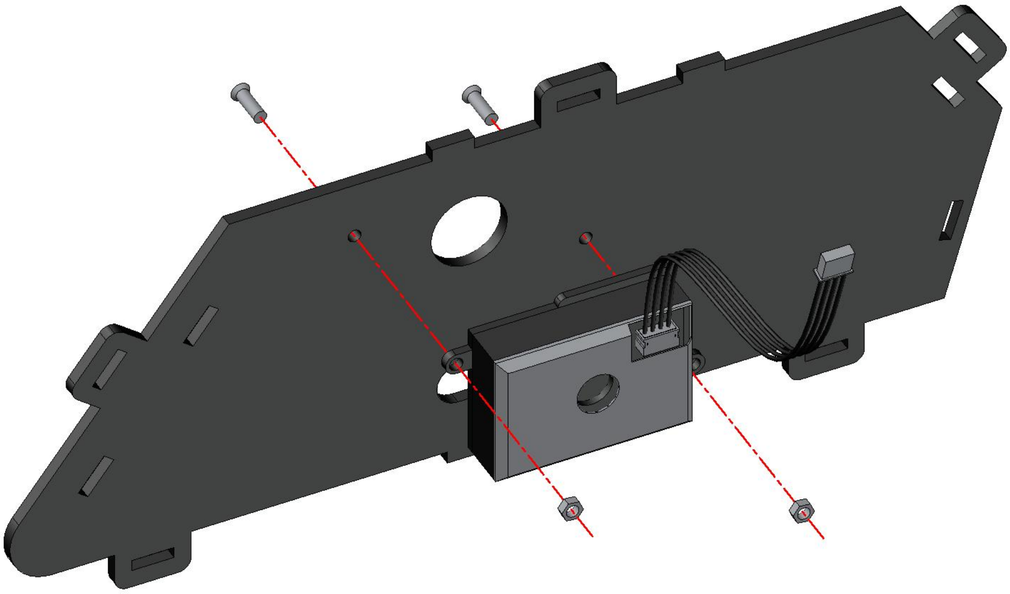
Note: Before installation, connect the 4P cable to the OLED module.



10.Install the PM2.5 sensor.

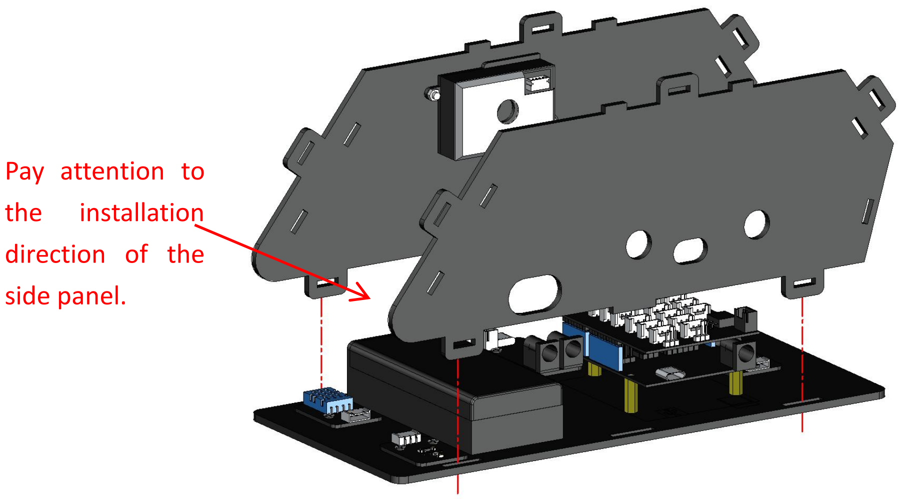
Parts List	
Name	**Quantity**
PM2.5 Sensor	1
Right Side Panel	1
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
4P Cable（One side is bigger and the other is smaller）	1

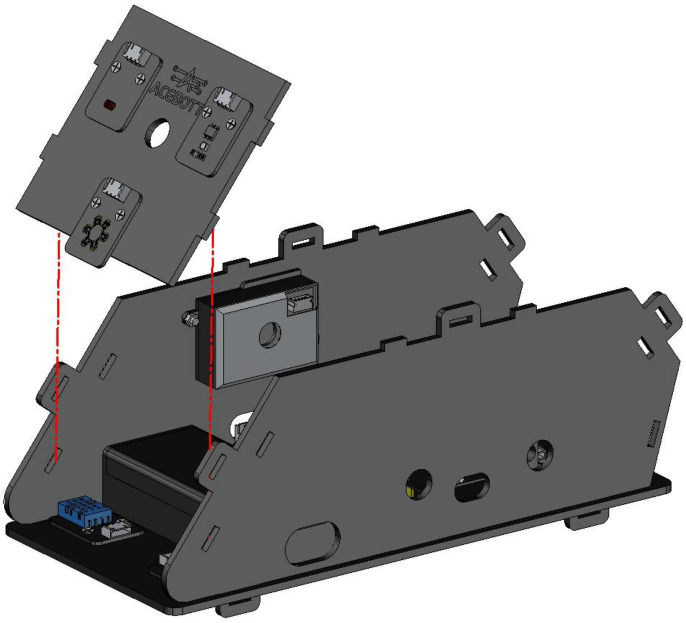
Note: Before installation, connect the 4P cable to the PM2.5 sensor.

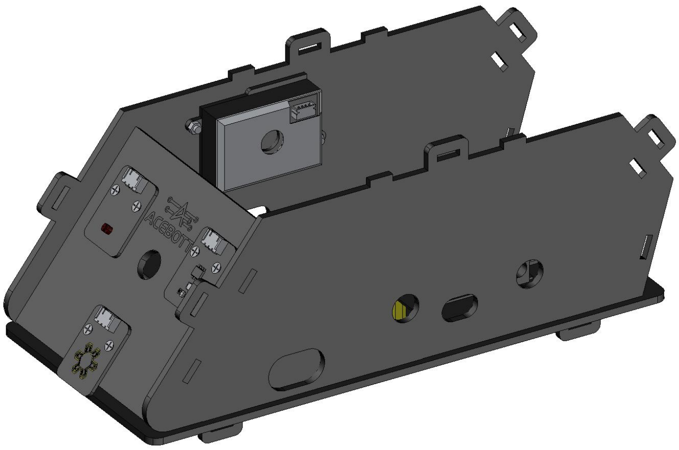


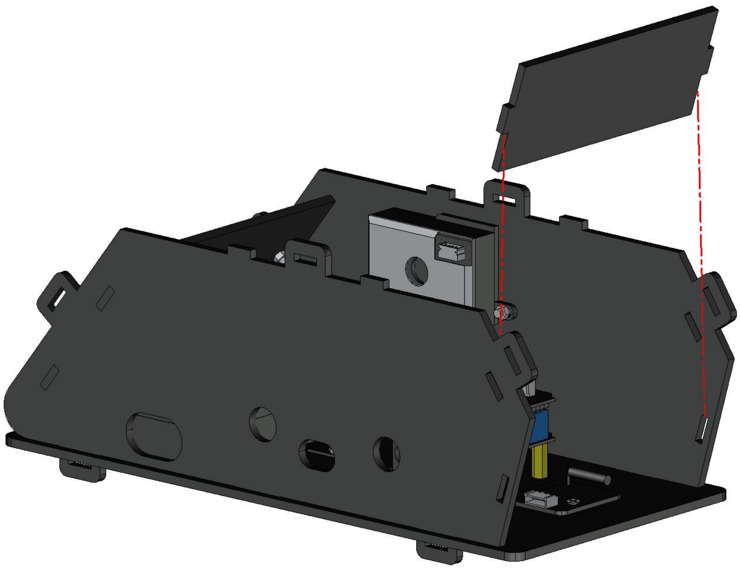
11. Assemble the side panels of the weather station.

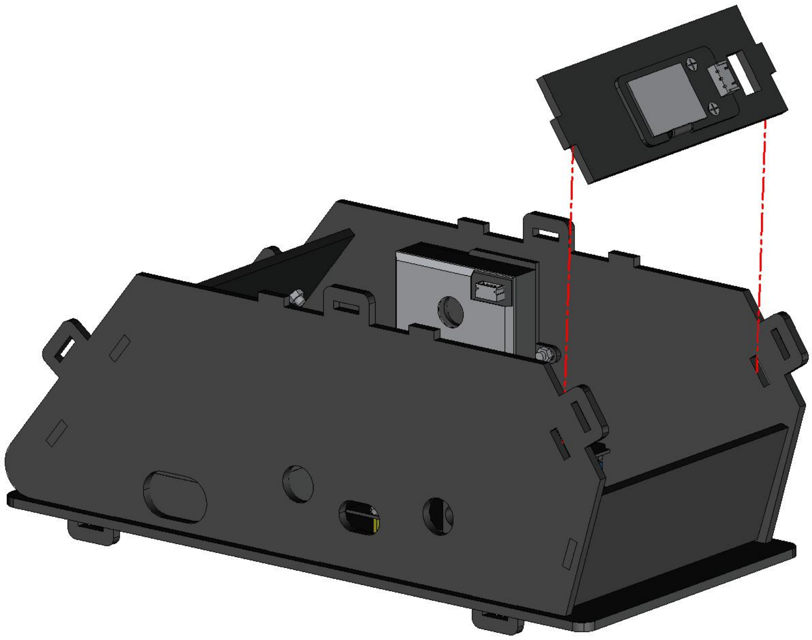
Parts List	
Name	**Quantity**
Lock	4

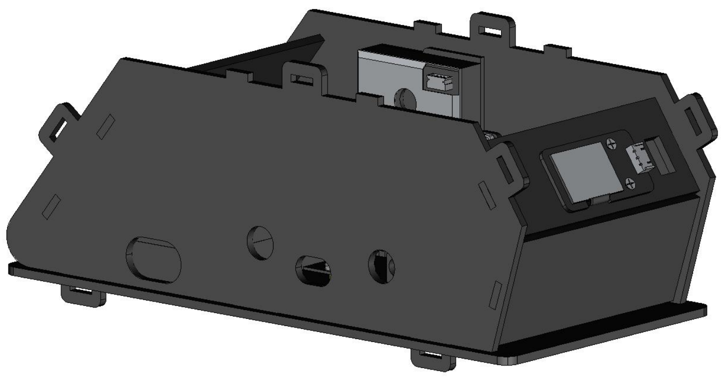


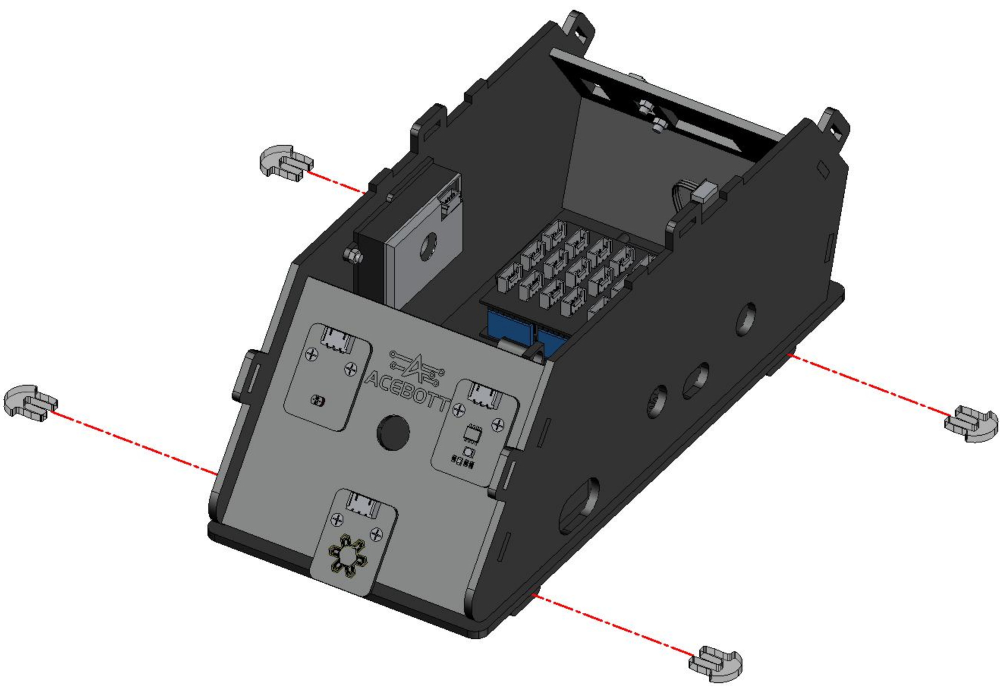






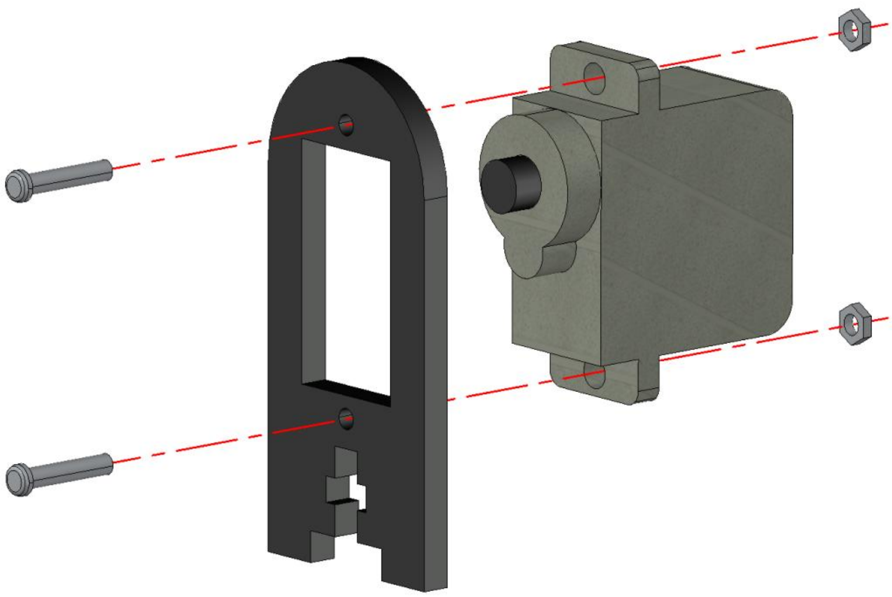


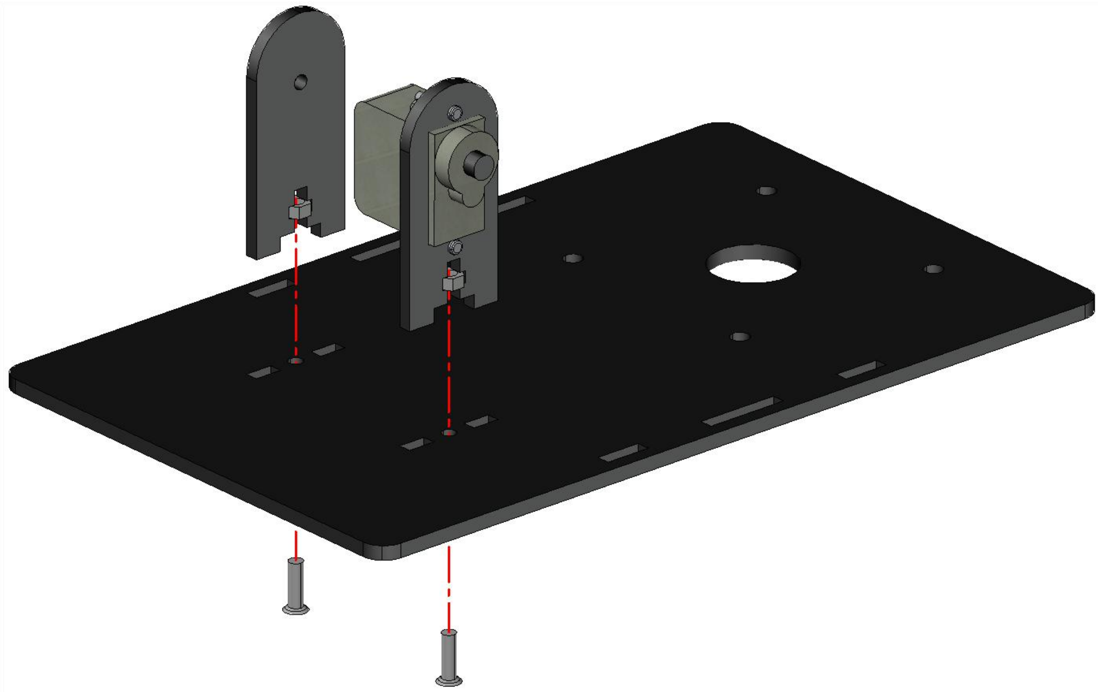




12. Install the servo MG90 9G.

Parts List	
Name	**Quantity**
Servo MG90 9G	1
Servo Bracket	2
Top Plate	1
M2*10 MM Round Head Screw	2
M2 Nickel-Plated Nut	2
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2

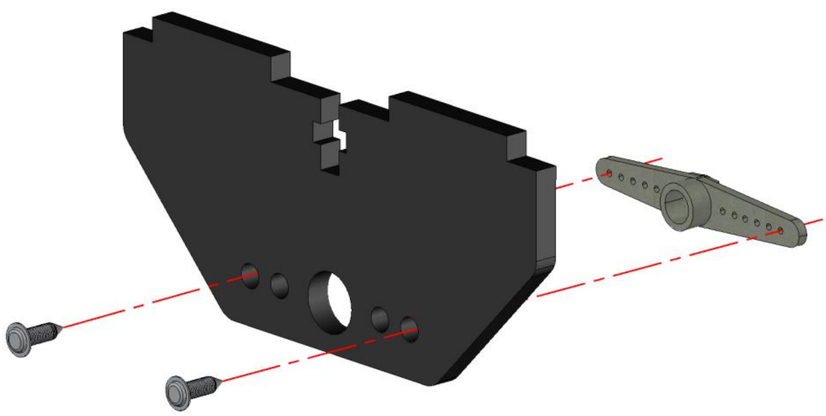




13.Install the servo horn.

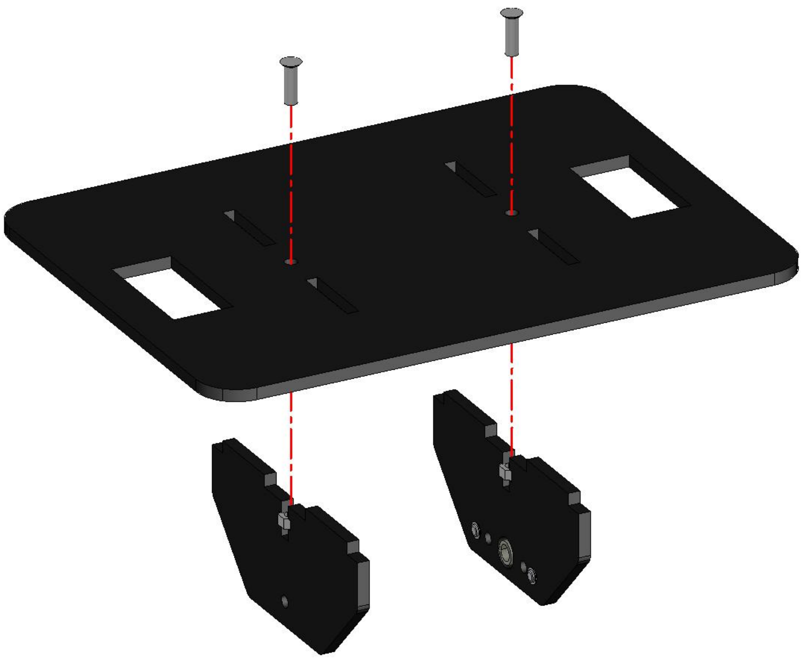
Parts List	
Name	**Quantity**
Single-axis Servo Horn	1
Solar Panel Mount Bracket	1
M1.7*6 Large Round Flat Head Tapping Screw	2

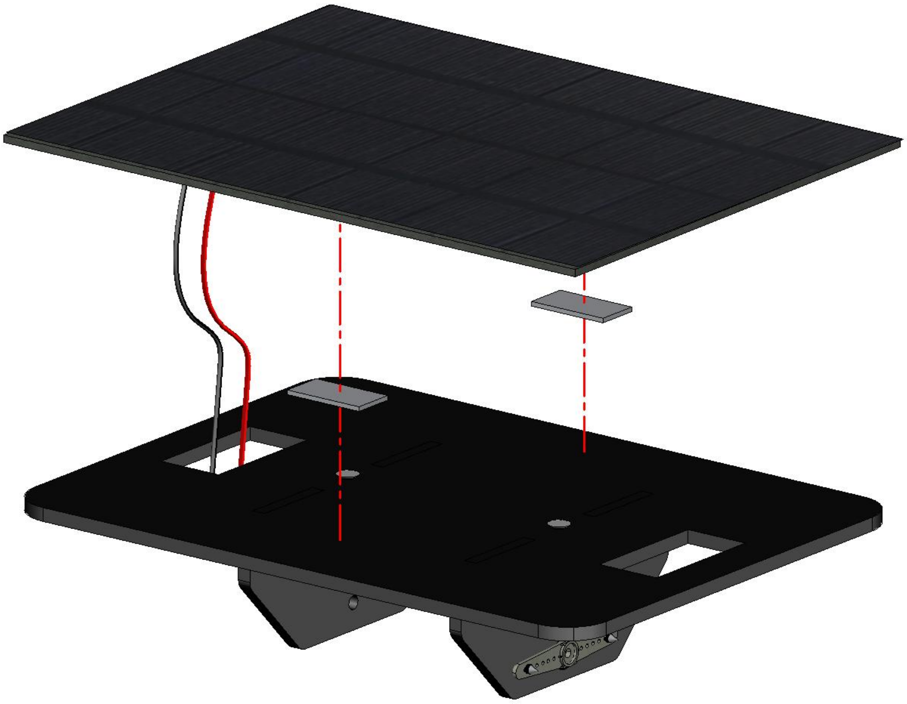
Note: The M1.7*6 large round flat head tapping screw mentioned here are not the ones included with the servo package.



14.Install the solar charging panel.

Parts List	
Name	**Quantity**
Solar Charging Panel	1
Solar Panel Tray	1
Solar Panel Mount Bracket	2
M3*10 MM Flat Head Screw	2
M3 Nickel-Plated Nut	2
Double Sides Tape	2

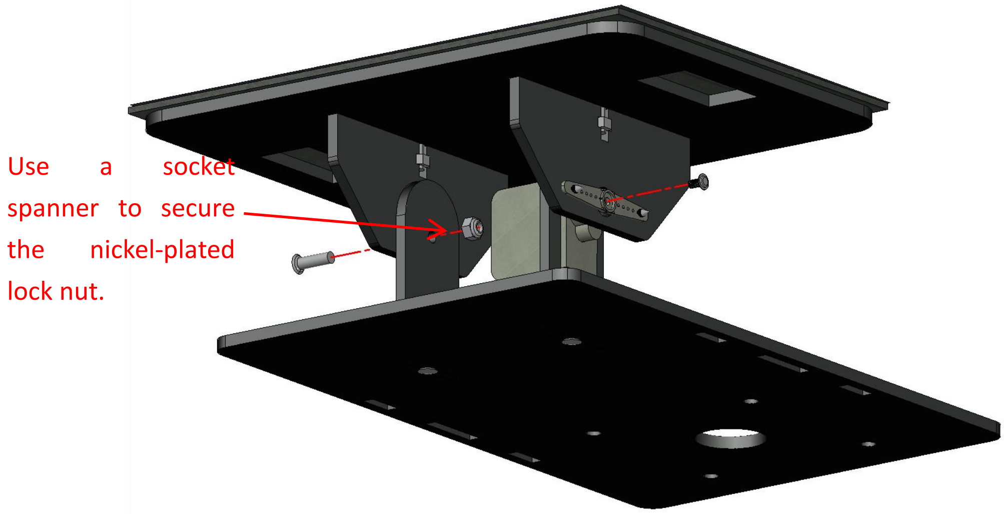




15.Install the solar panel mount bracket.

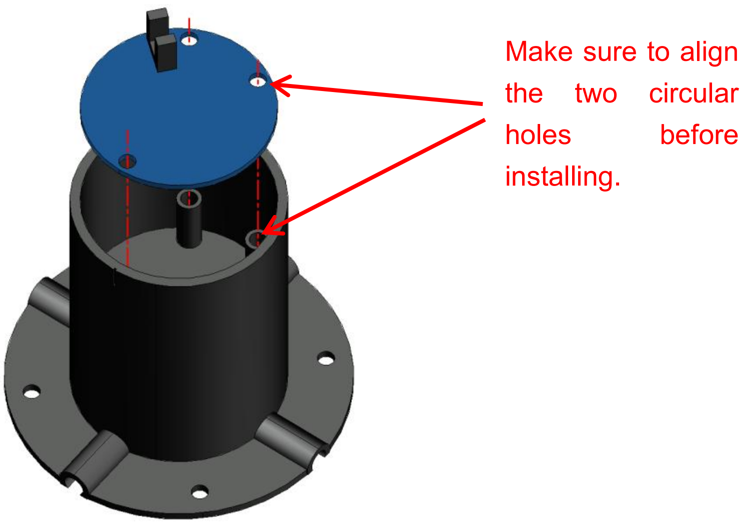
Parts List	
Name	**Quantity**
M3*10 MM Flat Head Screw	1
M3 Nickel-Plated Lock Nut	1
M2.5*4 Round Head Screw (Included with Servo Pack)	1

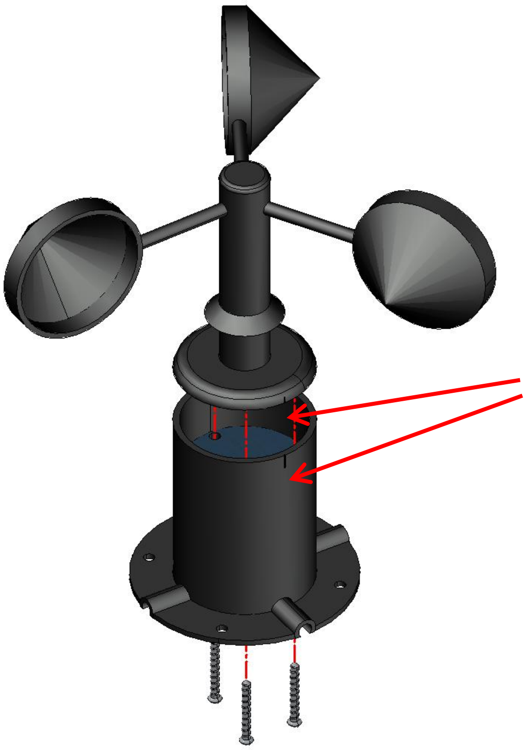
Before proceeding with this step, you should first initialize the servo by connecting the servo signal wire to pin 4 of the ESP32 controller board, power on the ESP32, click here to get the servo initialization program, and upload the program to the ESP32.

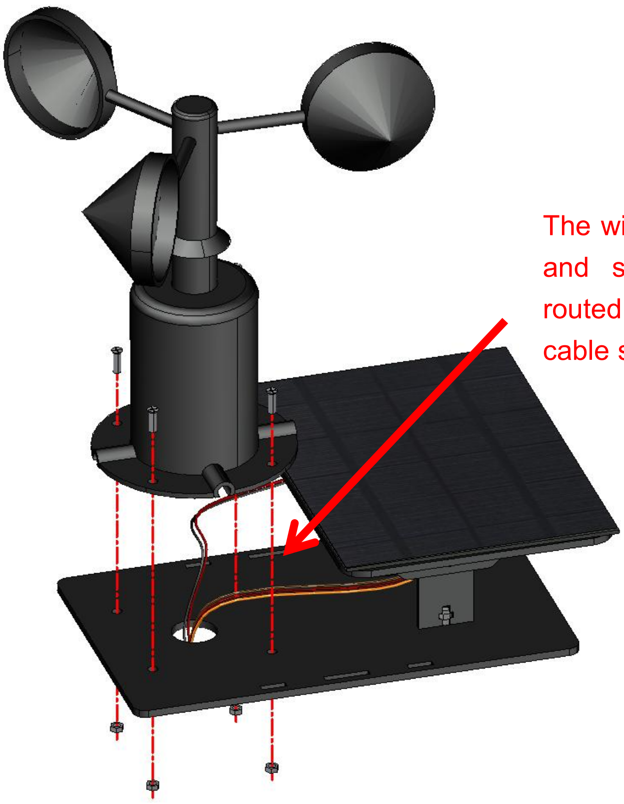


16.Install the anemometer.

Parts List	
Name	**Quantity**
Anemometer	1
M3*10 MM Flat Head Screw	4
M3 Nickel-Plated Nut	4
M3*25 Round Head Self-Tapping Screw	3

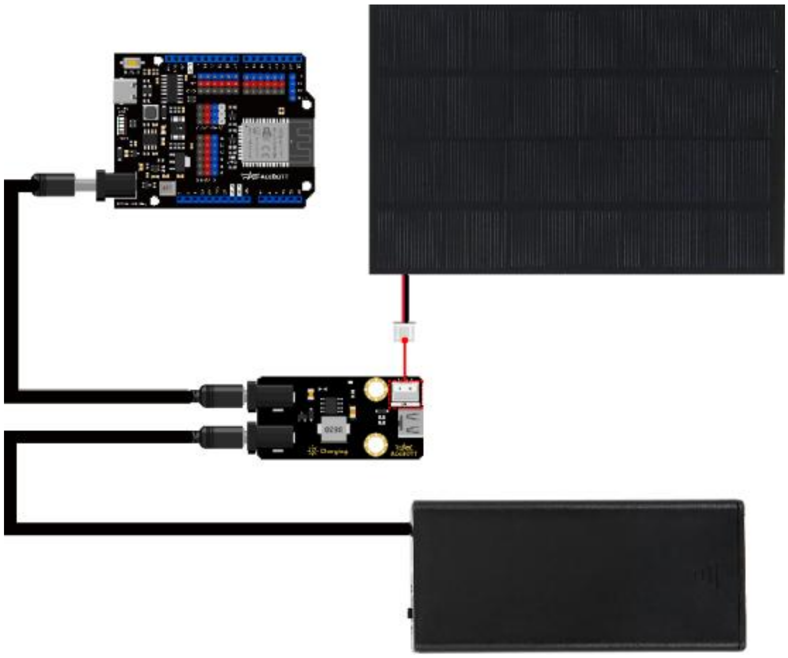




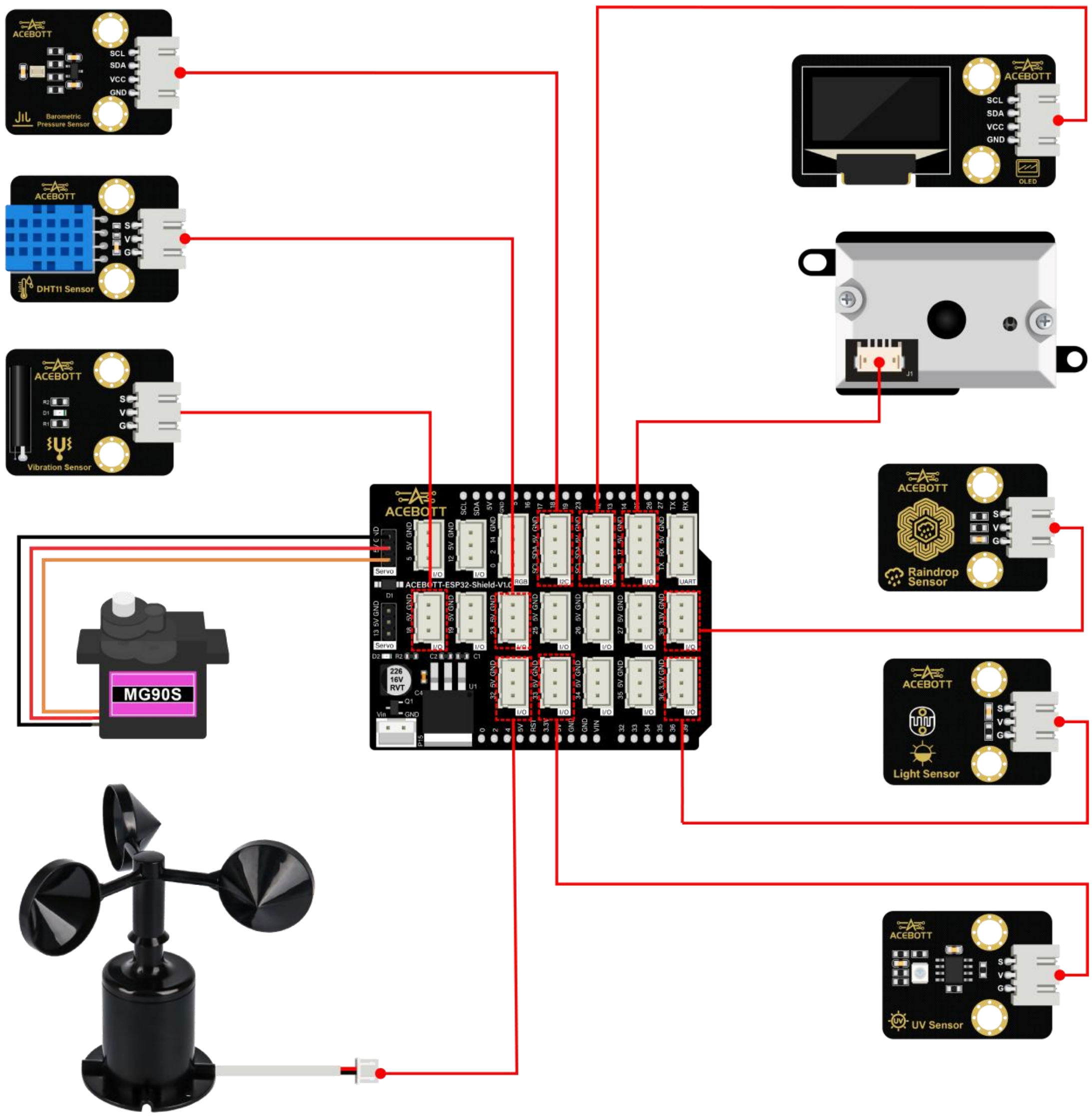


17.Wiring

(1) Solar Panel Wiring



(2) Wiring of Electronic Modules to the ESP32-Shield Board



Wiring Instructions:

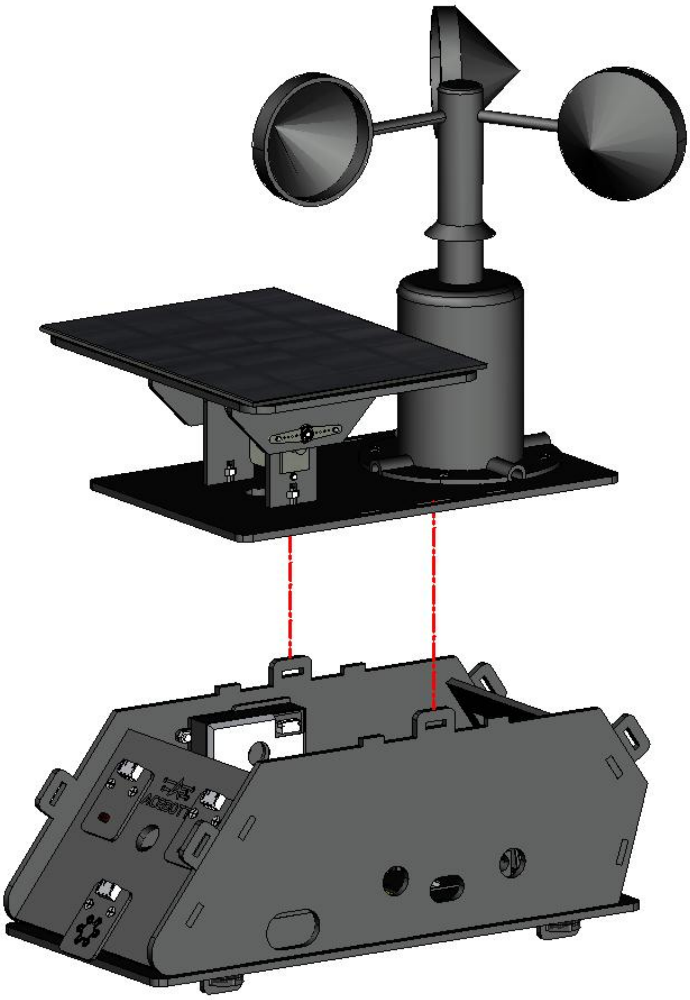
Module	ESP32 Pin
MG90S Servo Motor (Orange Wire)	PIN 4
Vibration Sensor	PIN 18
DHT11 Sensor	PIN 23
Anemometer	PIN 32
UV Sensor	PIN 33
Light Sensor	PIN 36
Raindrop Sensor	PIN39
PM2.5 Sensor	PIN16、PIN17

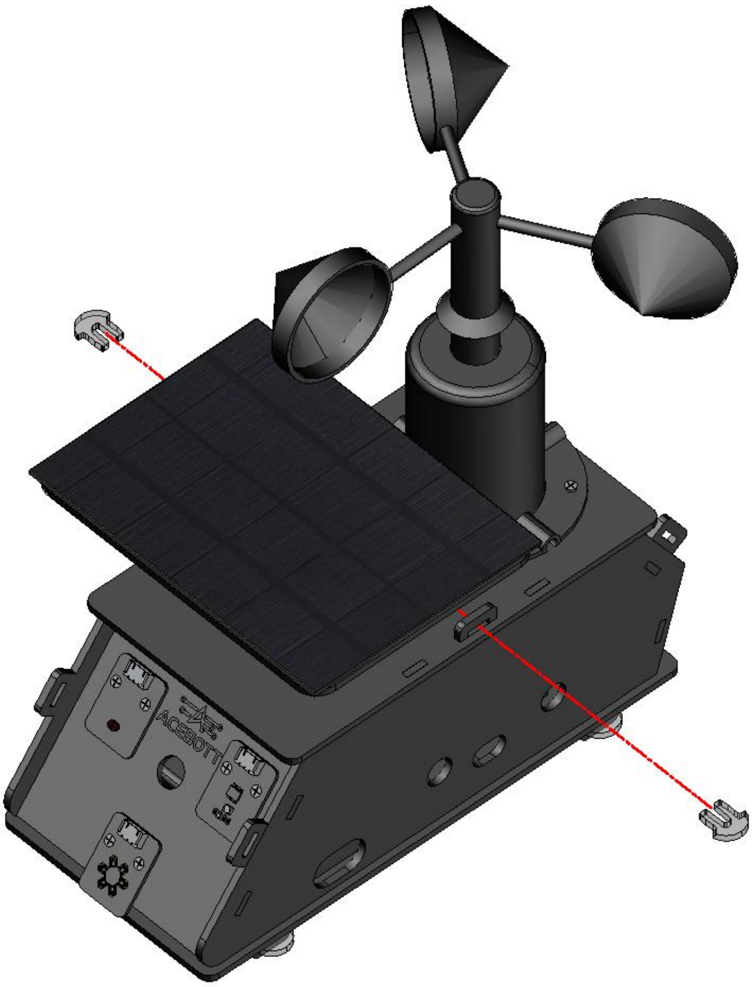
Atmospheric Pressure Sensor	I2C
OLED Module	I2C

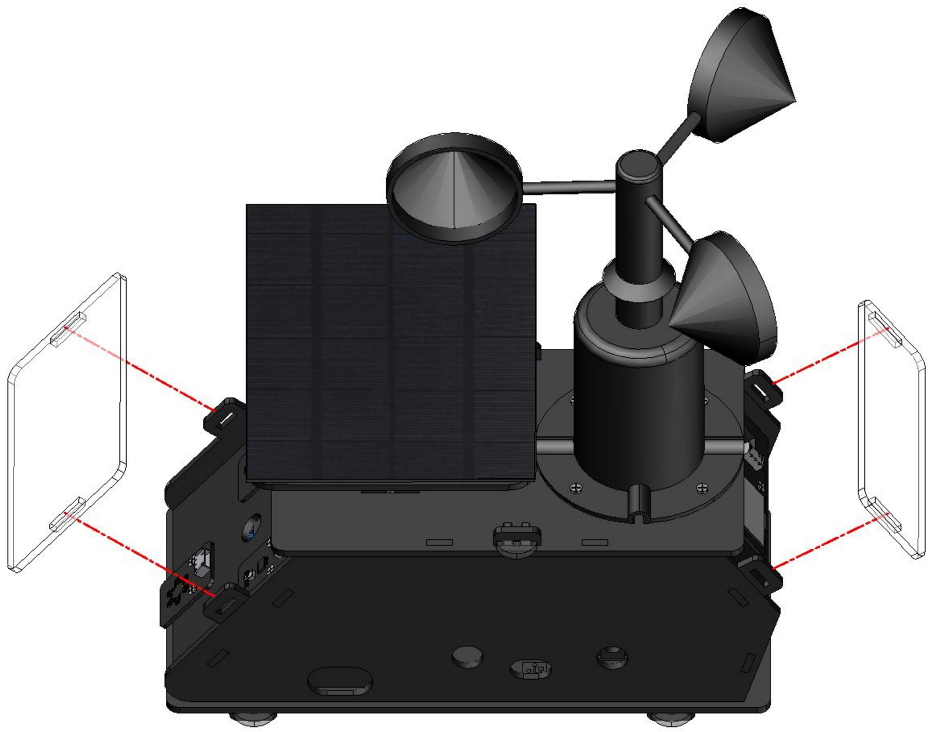
Note:Please make sure to strictly follow the wiring instructions when connecting the module to the ESP32 controller board. Incorrect wiring may cause a short circuit and damage the ESP32 controller board.

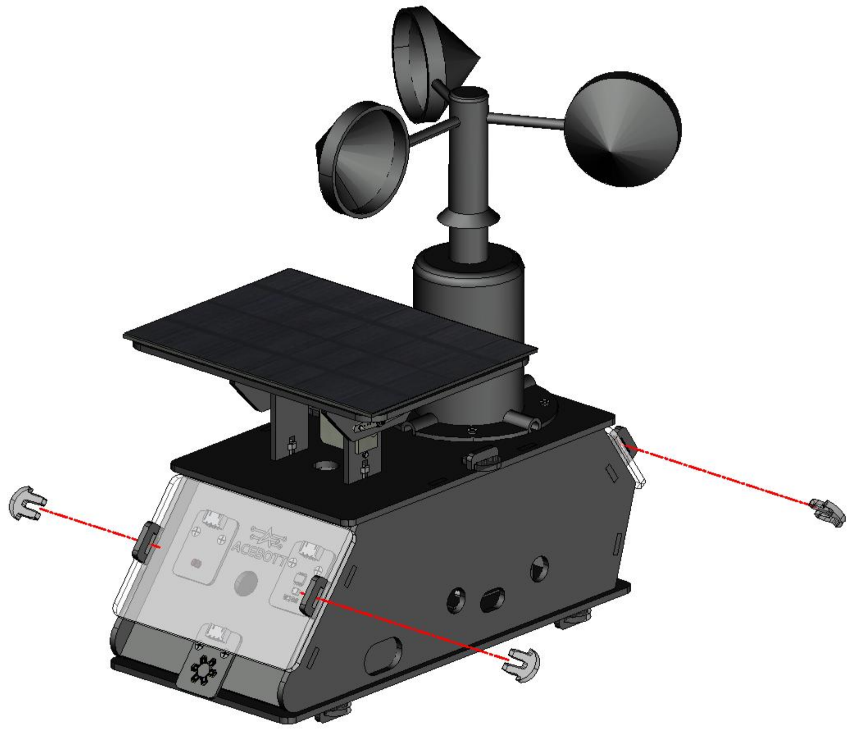
18.Assemble the complete weather station.

Parts List	
Name	**Quantity**
Transparent Acrylic Board	2
Lock	6




ACEBOTT





Lesson 3: Welcome to the Smart Weather Station

On the first day of the completed smart weather station, villagers gathered to watch, and Dr. Lumi introduced the various functions of the weather station to the villagers. At that moment, a villager asked, "How can we receive this meteorological data?" Dr. Lumi smiled and replied, "There is an OLED screen at the center of the weather station, where everyone can observe the meteorological data and various messages." To demonstrate the functionality of the OLED to the villagers, Dr. Lumi decided to assign you a special task - to display a welcome message for the villagers on the OLED screen.

Before completing the task given by Dr. Lumi, you need to understand the relevant content of OLED first.

I. Understanding the OLED Module

The OLED module (Organic Light Emitting Diode, organic light-emitting diode) is a self-luminous display module, widely used in various electronic projects, especially in small embedded systems, wearable devices, smart home devices, and other fields.

1. Features of OLED

Self-luminous: OLED does not require a backlight; the screen itself displays images through the luminescence of organic materials.

Ultra-thin design: Due to the lack of a backlight structure, OLED modules can be made very thin, making them suitable for devices with strict thickness requirements, such as smartwatches and mobile devices.

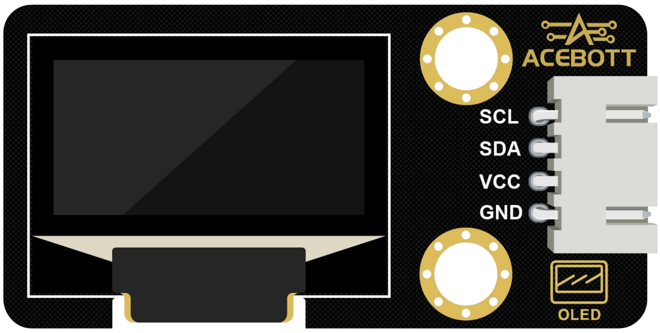
High contrast and brightness: OLED can display pure black (because pixels do not emit light at all when turned off), so its contrast is very high, and brightness can also be controlled very well.

Low power consumption: OLED modules consume less power when displaying dark or black colors, which is very important for projects powered by batteries.

Wide viewing angle: OLED maintains good image quality when viewed from different angles.

Fast response time: OLED has an extremely fast response time, making it suitable for displaying videos or dynamic images without any trailing issues.

2. Parameter Description



Resolution: 128*64, the higher the resolution, the clearer the displayed content.

Screen size: 0.96 inches.

Operating voltage: 3.3V/5V.

Color display: White.

Control method: I2C protocol.

3. Control Methods

Use commands from the "u8g2lib.h" library to control the OLED display content. Commonly used library functions include:

setI2CAddress(adr): Set the I2C address for the OLED screen, where "adr" is the specific address for I2C communication.

begin(): Initialize OLED Module.

enableUTF8Print(): Allow the OLED screen to display Unicode characters.

u8g2.setFont(font): Set the font format for displaying characters on the OLED screen, where "font" specifies the font format.

setCursor(x,y): Set the cursor position on the OLED screen.

print(str): Display text content on the OLED screen, where str is the specific content to be displayed.

clearBuffer(): Clear the content in the buffer.

sendBuffer(): Send the display content to the buffer to be shown on the screen.

II. OLED Screen Display Program

Control the OLED screen to display "Welcome to Smart Weather Station" through programming.

Click here to open the OLED screen display program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library

//creat an OLED object:
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);
void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.enableUTF8Print();//enable Unicode display
}

void loop() {
```

```
  u8g2.clearBuffer();//clear buffer of OLED
  u8g2.setFont(u8g2_font_timR10_tf);//set font of displaying text
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Hello,welcome to"   );//print text of displaying
  u8g2.setCursor(0,30);//set position of cursor
  u8g2.print("Smart Weather Station."   );//print text of displaying
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

After uploading the program, you will see the following effect on the OLED screen:

Lesson 4: Earthquake Alarm System

One early morning in Future Village, the villagers felt a slight shaking of the ground, and the water in their tea cups rippled with tiny waves. Although the earthquake was minor and caused no damage, it unsettled the villagers. Future Village is located in a mountainous area with frequent geological activity, and everyone is worried about whether a larger earthquake is coming. The villagers went to Dr. Lumi one after another, hoping that Dr. Lumi could help them sense the arrival of an earthquake in advance.

Dr. Lumi soothed the villagers' emotions and said, "Actually, before an earthquake comes, we can infer the possibility of an earthquake by observing the natural phenomena around us."

The villagers asked, "What natural phenomena indicate that an earthquake is coming?"

Dr. Lumi explained, "The main cause of an earthquake is that the Earth's surface is composed of several huge plates, which float and move slowly on the lithosphere. The interaction between the plates can cause huge stress. Once the stress accumulates to a certain extent, the rocks will suddenly break or slide, causing an earthquake."

"When an earthquake occurs, it releases a huge amount of energy.

The infrasound waves produced can cause changes in the Earth's magnetic field. These changes are imperceptible to us humans, but some animals can sense them, which may lead to abnormal behavior before an earthquake, such as pets like dogs and cats suddenly becoming restless or trying to flee their homes, fish swimming erratically in the water, jumping, or even dead fish phenomena."

"Earthquakes can also cause fractures or changes in the groundwater system, affecting water sources, resulting in phenomena such as sudden turbidity of well water or bubbles emerging."

"Although these natural phenomena have been observed before some earthquake events, scientists have not yet been able to accurately predict the time and place of earthquakes. Earthquake prediction remains a major challenge in Earth science research."

"There is a vibration sensor in the smart weather station. It can perceive subtle vibrations and is used to detect minor ground shaking, which can help villagers warn of earthquakes and ensure everyone's safety."

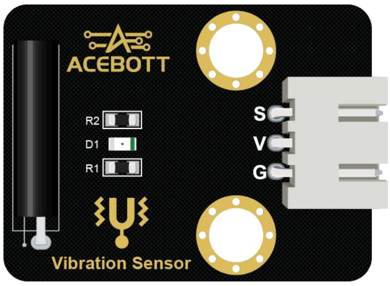
Next, Dr. Lumi asks you to help the villagers build an earthquake alarm system. Before completing this challenge, you need to understand the related content of the vibration sensor.

I. Vibration Sensor

A vibration sensor is a device used to detect and measure the vibration of objects. It can convert mechanical vibrations into electrical signals, which are then processed and analyzed by the circuit. Vibration sensors typically consist of a sensing element, a load resistor, and a supporting structure. The sensing element can be a piezoelectric ceramic, a resistive strain gauge, a capacitive sensor, or a spring switch, etc.

1. Sensor Description

The vibration sensor in this tutorial uses the SW-18010P high-sensitivity vibration switch, which is a spring-type non-directional vibration sensing device that can be triggered at any angle. When the module is stationary and undisturbed, the two internal terminals are disconnected, and the signal S terminal outputs a high level. When subjected to external force impact or severe vibration, the spring deforms and the center electrode contacts to make the two terminals momentarily conductive, and the signal S terminal outputs a low level. When the external force disappears, the circuit returns to the disconnected state.



2. Parameter Description

Operating Voltage: 3.3V/5V.

Operating Temperature: -20°C to +70°C.

Output Type: Digital signal.

II. Earthquake Alarm Program

Detect the occurrence of an earthquake through programming, and when an earthquake is detected, display a message on the OLED screen indicating that an earthquake has occurred.

Click here to open the earthquake alarm program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library

#define vibrationPin 18 //declare pin of vibration sensor

//creat OLED object：
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

void setup() {
  pinMode(vibrationPin,INPUT);//set the pin of vibration sensor as INPUT mode
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode display
  Serial.begin(115200);//set paud rate
```

```
}

int value;//creat variable of vibration sensor value
void loop() {
  value = digitalRead(vibrationPin);//read value of vibration sensor
  Serial.println(value);//serial print value of vibration sensor
  u8g2.clearBuffer();//clear buffer of OLED
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Vibration Signal:" + String(value) );//display value of vibration sensor
  u8g2.setCursor(0,30);//set position of cursor
  if(value){//if no earthquake is detected
    u8g2.print("Normal");//display"Normal"
    u8g2.sendBuffer();//send text to buffer
  }
  else{//If earthquake is detected
    u8g2.print("Earthquake");//display"Earthquake"
    u8g2.sendBuffer();//send text to buffer
    delay(5000);
  }
}
```

After uploading the program, you can see the following effect on the OLED screen: when no vibration is detected, it displays "Normal," and when vibration is detected, it displays "Earthquake."

Lesson 5: Preventing Ultraviolet Radiation

The summer climate of "Future Village" is highly variable, with periods of intense sunlight followed by continuous rain. Recently, as the sun's rays have become increasingly strong and temperatures have soared, many villagers have suffered from sunburn while working in the fields. Dr. Lumi pointed out: "Ultraviolet (UV) radiation in sunlight, when of high intensity, can cause damage to the skin and eyes." In order to raise the villagers' awareness of sun protection and safeguard their health, Dr. Lumi plans to use the ultraviolet sensor equipped with the smart weather station to develop a sun protection reminder system.

Now, you will assist Dr. Lumi in implementing the design work of the sun protection reminder system. To better complete the task, you need to understand the relevant knowledge about ultraviolet radiation and ultraviolet sensor.

I. Ultraviolet Index

Ultraviolet (UV) radiation is a type of electromagnetic radiation that falls between visible light and X-rays in terms of wavelength. Based on different wavelengths, ultraviolet radiation can be divided into three types: UVA, UVB, and UVC. UVA has a longer wavelength and strong penetrating power, capable of reaching the dermis layer of the skin, causing risks such as skin aging and skin cancer. UVB has a shorter wavelength and weaker penetrating power, affecting the outer layer of the skin and is the main cause of sunburn. UVC has the shortest wavelength and is generally absorbed by the Earth's ozone layer, not reaching the ground.

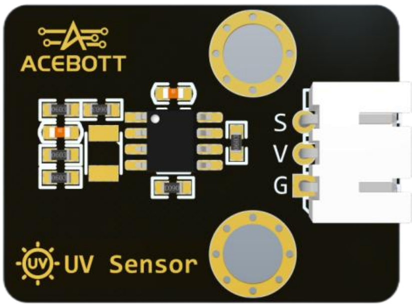
The Ultraviolet Index is a standard for measuring the intensity of ultraviolet radiation, used to guide the public in taking necessary protective measures during outdoor activities. The value of the Ultraviolet Index is usually divided into several levels, each corresponding to different health risks and protection recommendations. Below is the impact of the Ultraviolet Index on the human body:

Level	Impact	Preventive Measures
0-2	No Risk to Human Health	No Special Protection Needed.
3-5	Prolonged time outdoors may cause skin damage.	Wear sun-protective clothing, wear sunglasses.
6-7	The risk of damage to the skin and eyes increases.	Wear sun-protective clothing, wear sunglasses.
8-10	High risk, exposure to sunlight for a short period may cause skin injury.	Try to avoid sunlight as much as possible.
11+	Extremely high risk, exposure for a few minutes may lead to severe sunburn.	Try to avoid sunlight as much as possible.

II. UV Sensor

1. Sensor Description

The UV sensor is an optimal sensor for measuring the total amount of ultraviolet radiation. It does not require the use of wavelength filters and is sensitive only to ultraviolet light. It directly outputs a linear voltage corresponding to the ultraviolet index (UV INDEX), with an output voltage range of approximately 0 to 5V (corresponding to UV INDEX values of 0 to 11).



2. Parameter Description

Operating Voltage: 3.3V.

Output Type: Analog signal.

Response Wavelength: 200nm to 370nm.

III. Ultraviolet Detection Program

Implement UV sensor detection of ultraviolet radiation intensity in the environment through programming. When the ultraviolet index reaches a certain value, remind villagers to pay attention to sun protection.

Click here to open the ultraviolet reminder program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library
#include <ACB_Ultraviolet.h> //import UV library

#define UVPin 33 //declare pin of vibration sensor

ACB_Ultraviolet uv;//creat UV object
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);//creat OLED object

int uv_index;

void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode display
  Serial.begin(115200);//set paud rate
  uv.setpin(UVPin);
}

void loop() {
  uv_index = uv.read("level");
  delay(20);
  u8g2.clearBuffer();//clear buffer of OLED
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("UV Index:" + String(uv_index) );//display UV index
  if(uv_index > 6){
    u8g2.setCursor(0,30);//set position of cursor
    u8g2.print("UV level is dangerous.");//UV warning
  }
  else{
    u8g2.setCursor(0,30);//set position of cursor
    u8g2.print("UV level is safe.");
  }
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

OLED screen: when the UV level is less than 6, it displays "UV level is safe." When the UV level exceeds 6, it displays "UV level is dangerous."

Lesson 6: Light Detection

On a sunny morning, the villagers warmly invited Dr. Lumi and his students to the farm in Future Village to visit and study the growth conditions of the crops. During their careful observation, they noticed that some of the crop leaves showed signs of withering and brown spots, clearly unable to grow normally. Faced with the villagers' worries, Dr. Lumi patiently explained, "This is mainly due to the continuous impact of the recent high temperatures, with the crops exposed to intense sunlight for a long time, leading to damage to plant cell membranes and chlorophyll."

Dr. Lumi further educated the villagers on the key role of light intensity in plant growth: "Appropriate light conditions are an indispensable element for plant growth. They can promote photosynthesis, enhance the efficiency and quality of plant growth, and lay a solid foundation for a bountiful harvest. However, if the light is too weak, it will slow down the growth rate, cause the leaves to turn yellow, and even lead to plant death; on the contrary, if the light is too strong, it will directly damage the plant cell membranes, causing the leaves to wither."

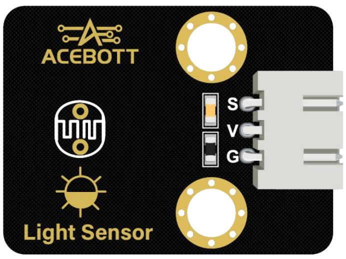
To more effectively solve the problems faced by the villagers, Dr. Lumi decided to use a smart weather station to monitor light intensity in real-time. It allows the villagers to more intuitively understand the growth environment of the crops, thereby taking more scientific and reasonable management measures to ensure the healthy growth of the crops.

Next, you will be tasked with implementing a light detection system. Before completing the task, you need to understand the knowledge of light sensors.

I. Light Sensor

1. Sensor Description

A light sensor, also known as a photoresistor, is a sensor that can detect the intensity of ambient light. Its working principle is based on the resistance characteristics of photosensitive materials, usually made from semiconductor materials. Light sensors have a lower resistance in strong light and a higher resistance in weak light.



2. Parameter Description

Operating Voltage: 3.3V/5V.

Operating Temperature: -10° C to +50° C.

Output Type: Analog signal.

II. Light Detection Program

Implement a light sensor to detect the intensity of ambient light through programming and display the light value on the OLED screen.

Click here to obtain the light detection program.

Reference program:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library

//creat an OLED object
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
}

void loop() {
  int lightness = analogRead(36);//read light value
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("lightness:"+String(lightness) );//display lightness value
  u8g2.setCursor(0,30);//set font
  if(lightness<1500){ //if light value less than 1500
    u8g2.print("Strong Light!");//display "Strong Light"
  }
  else if(lightness>1500 && lightness<2500){//if light value more than 1500 and less than 2500
    u8g2.print("Weak Light!");//display "Weak Light"
  }
  else{
    u8g2.print("Darkness!");//display "Darkness"
  }
```

```
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

After uploading the program, you can see the following effect on the OLED screen: it displays the intensity value of the light and divides the light into different levels.

Lesson 7: Rain Alert System

In the summer of Future Village, in addition to the scorching sun, there are also frequent periods of continuous rain. These long rainy days may quietly erode the foundation of the crops, causing them to fall into a dilemma of poor nutrition and frequent pests and diseases, ultimately affecting the harvest and prosperity of the entire village. To more effectively manage the growth of crops during the rainy season, we must monitor the rainfall in real-time to take timely preventive measures. To this end, the smart weather station is equipped with an advanced raindrop sensor, which can detect rainfall in real-time and issue early warning prompts to the villagers, allowing them to respond quickly and take necessary measures to protect the crops.

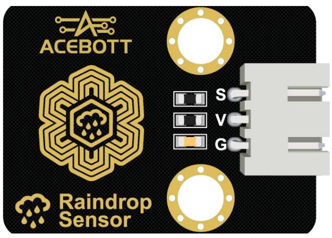
Next, you will be tasked with designing a rain alert system for the villagers. Before completing this task, you need to understand the relevant knowledge of raindrop sensor.

I. Raindrop Sensor

1. Sensor Description

A raindrop sensor is a device used to detect the presence of rainwater, widely used in meteorological monitoring, automated irrigation systems, car wiper control, and other fields. Raindrop sensors are divided into resistive raindrop sensors and capacitive raindrop sensor.

The project uses a resistive raindrop sensor, which is a sensor that detects the presence of rainwater by utilizing changes in resistance. Its working principle is mainly based on the fact that when raindrops come into contact with the sensor surface, they change the internal resistance value of the sensor, thus the presence of rainwater can be detected through changes in resistance. This type of sensor usually includes a sensitive surface with multiple conductive channels, which have a higher resistance when dry and a lower resistance when covered with rainwater.



2. Parameter Description

Operating Voltage: 3.3V/5V.

Operating Temperature: -25° C to +85° C.

Output Type: Analog signal.

II. Rain Alert System Program

Implement the function of a rain alert through programming, displaying rain information on the OLED screen when rain is detected.

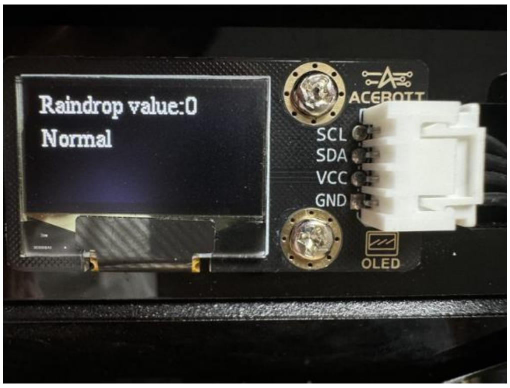
Click here to obtain the rain alert program.

Reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library

#define raindropPin 39 //declare raindrop sensor

//creat OLED object
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
}

int value;
void loop() {
  value = analogRead(raindropPin);//read value of raindrop sensor
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Raindrop value:" + String(value) );//display value of raindrop sensot
  u8g2.setCursor(0,30);//set position of cursor
  if(value < 1000){//if no rain is detected
    u8g2.print("Normal");//display"Normal"
    u8g2.sendBuffer();//send the text to buffer
    delay(500);
  }
  else{//if rain is detected
    u8g2.print("Raining");//display"Raining"
    u8g2.sendBuffer();//send the text to buffer
```

```
    delay(5000);
  }
}
```

After uploading the program, you can see the following effect on the OLED screen: it displays the value of the raindrop sensor and indicates whether it is raining.



Lesson 8: Temperature and Humidity Detection

During their time in Future Village, some students have suffered from heatstroke, while others have been troubled by skin diseases. After an in-depth analysis by Dr. Lumi, these health issues were attributed to the local climate characterized by high temperature and humidity. Further research revealed that this harsh climate not only affects the health of the students but is also a common challenge faced by the local villagers.

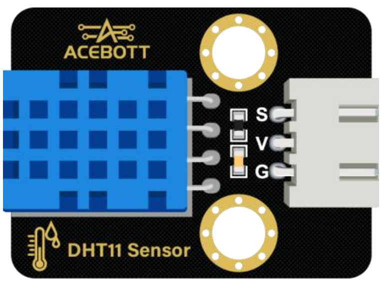
To effectively address and prevent the recurrence of such situations, Dr. Lumi decided to design and develop a temperature and humidity monitoring system. This system mainly helps villagers prepare in advance through real-time monitoring and early warning, thereby effectively resisting the invasion of high temperature and humidity environments and ensuring everyone's health and quality of life.

Next, you will design a temperature and humidity monitoring system through the temperature and humidity sensor of the smart weather station. Before completing this challenge, you need to understand the knowledge of temperature and humidity sensors.

I. Temperature and Humidity Sensor

1. Sensor Description

A temperature and humidity sensor is a device used to measure the temperature and humidity in the environment. The type of temperature and humidity sensor used in this project is the DHT11 sensor, which measures the temperature and humidity in the environment through internal electronic components. The temperature measurement uses a component called a thermistor, while the humidity measurement utilizes a humidity-sensitive capacitor. The DHT11 sensor outputs the measurement results digitally, which means it can be directly connected to the development board and transmit data through a digital interface. The DHT11 sensor is widely used in various applications such as indoor temperature and humidity monitoring, weather stations, greenhouse control, smart home systems, especially for scenarios that require real-time digital temperature and humidity data.



2. Parameter Description

Operating Voltage: 5V.

Operating Temperature: -20° C to +60° C.

Output Type: Digital signal.

3. Control Methods

In this project, commands from the "ACB_DHT11.h" library file will be used to obtain values from the temperature and humidity sensor.

get_Temperature_Data(): Get the temperature value detected by the DHT11 sensor.

get_Humidity_Data(): Get the humidity value detected by the DHT11 sensor.

II. Temperature and Humidity Detection Program

Implement temperature and humidity detection through programming and display the temperature and humidity values on the OLED screen.

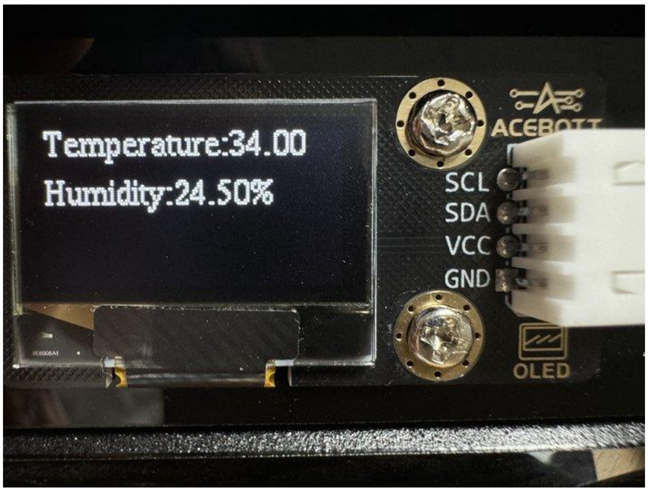
Click here to obtain the temperature and humidity program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library
#include <ACB_DHT11.h>//import DHT11 library

#define DHTPIN 23 //declare pin of DHT11

ACB_DHT11 dht11(DHTPIN);//creat a DHT11 object
```

```
//creat an OLED object:
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
  Serial.begin(115200);//set paud rate
}

void loop() {
  float h = dht11.get_Temperature_Data(); //read temperature value
  float t = dht11.get_Humidity_Data();   //read humidity value
  Serial.println(h);
  Serial.println(t);
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Temperature:" + String(t)+"℃" );//display value of temperature
  u8g2.setCursor(0,30);//set position of cursor
  u8g2.print("Humidity:" + String(h)+"%" );//display value of humidity
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

After uploading the program, you can see the following effect on the OLED screen, displaying the temperature and humidity values.



Lesson 9: Wind Speed Detection

One early morning, while Dr. Lumi was out for a walk, he stumbled upon the devastating state of the villagers' orchards: branches scattered all over the ground and fruits in disarray. After inquiring with the villagers, he learned that Future Village occasionally experiences windy weather, and last night was no exception with a strong wind that not only ravaged the fruit trees but also damaged some of the villagers' greenhouses.

Upon hearing this, Dr. Lumi became deeply concerned and immediately made a decision—to establish an efficient wind speed detection system for the village. This system would be able to monitor wind speed in real-time, ensuring that villagers could prepare in advance for strong winds by reinforcing their houses and greenhouses, avoiding unnecessary outings, and thus minimizing damage.

Dr. Lumi has decided to entrust you with the task of building the wind speed detection system. Before you embark on this task, you need to understand the relevant knowledge about Anemometers.

I. Anemometer

1. Sensor Description

Anemometer come in various types, such as mechanical, thermistor, and ultrasonic. For this project, we are using a three-cup mechanical Anemometer. It consists of three conical cups, an axis, and a bracket, with the cups evenly distributed around the axis. The sensor housing is made of polycarbonate composite material, which is corrosion and erosion-resistant, ensuring long-term use without rust. It is equipped with a smooth internal bearing system to ensure reliable data collection. It also integrates an optoelectronic conversion mechanism, an industrial microcomputer processor, a standard current generator, and a current driver.

When the wind blows towards the cups, they are propelled by the force of the wind, causing them to rotate. The rotation speed of the cups is proportional to the wind speed. The greater the wind speed, the faster the cups rotate. By measuring the rotational speed of the cups and combining it with the instrument's calibration curve (i.e., the relationship between wind speed and rotation speed), the actual wind speed can be calculated.

2. Parameter Description

Operating Voltage: 5V.

Output Type: Analog signal (0-5V voltage signal).

Measurement Range: 0-30 m/s.

Start-up Wind Speed: ⩽ 0.3 m/s.

The relationship between output voltage and wind speed: wind speed = 6 * output voltage.

II. Wind Speed Detection Program

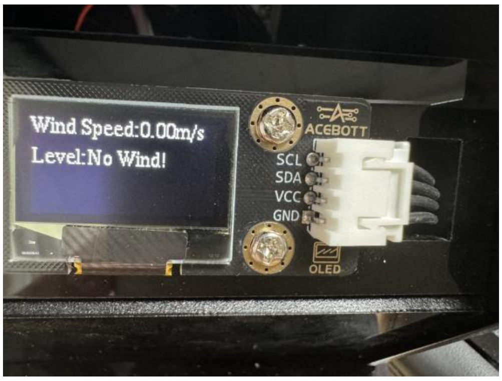
Implement wind speed detection functionality through programming and display the wind speed on the OLED screen.

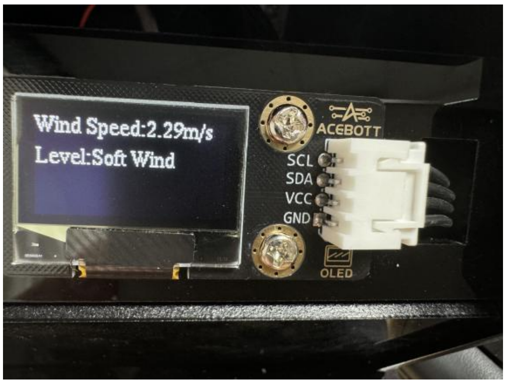
Click here to obtain the wind speed program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library

#define windPin 32//declare pin of wind speed sensor
//creat an OLED object
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

float voltage;
float speed;
String level;
void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
}

void loop() {
  int sensor_input = analogRead(windPin);//read value of wind speed sensor
  voltage = (sensor_input/4095.0)*5.0;//get output voltage of wind speed sensor
  speed =   6 * voltage;//calculate wind speed
  if(speed < 0.3){
    level = "No Wind!";
  }
  else if(speed >= 0.3 && speed < 5.4 ){
    level = "Soft Wind";
  }
  else{
    level = "Strong Wind";
  }
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Wind Speed:"+String(speed)+"m/s" );//display wind speed
  u8g2.setCursor(0,30);//set position of cursor
  u8g2.print("Level:"+String(level));//display wind speed level
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

After uploading the program, you can see the following effect on the OLED screen, displaying the wind speed and the wind level.





Lesson 10: Weather Forecasting

After a period of observation, Dr. Lumi found that the weather in Future Village was highly variable. When severe weather approached, villagers sometimes failed to take timely measures. To better assist the villagers in taking preventive actions, Dr. Lumi hoped to create a weather forecasting system using the Atmospheric Pressure Sensor from the weather station, allowing villagers to anticipate future weather conditions. To help villagers understand how to predict the weather, Dr. Lumi explained the relationship between atmospheric pressure and weather to them.

"Atmospheric pressure refers to the pressure exerted by the Earth's atmosphere on the ground, which is generated by the weight of the air. The magnitude of atmospheric pressure varies with factors such as altitude, temperature, and humidity. Changes in atmospheric pressure are closely related to the weather. When the pressure drops, it indicates deteriorating weather, suggesting possible rain or stormy conditions ahead. When the pressure rises, it indicates improving weather, suggesting possible clear and dry conditions ahead. So, when villagers notice a significant drop in pressure, they should take preventive measures."

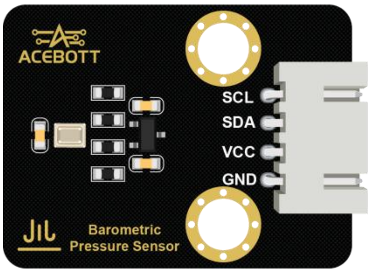
Next, you will take Dr. Lumi's place to build a weather forecasting system. Before completing this task, you also need to understand the relevant knowledge about Atmospheric Pressure Sensors.

I. Barometric Pressure Sensor

1. Product Description

The Barometric Pressure Sensor used in this project is the BMP280, a pressure and temperature sensor from Bosch, widely used in applications such as pressure detection, temperature detection, and indoor and outdoor navigation.

The BMP280 operates through MEMS (Micro-Electro-Mechanical Systems) technology, featuring a piezoresistive pressure sensor inside. When barometric pressure acts on the sensor's diaphragm, it causes a slight deformation, which in turn changes the internal resistance value. The sensor measures this resistance change to calculate the current barometric pressure. The BMP280 also integrates a temperature sensor inside for compensating environmental temperature to ensure the accuracy of the pressure data. It uses resistive temperature detection technology, calculating temperature based on the change of material resistance with temperature.



2. Parameter Description

Operating Voltage: 5V

Operating Temperature: -0° C to +50° C

Pressure Measurement Range: 300-1100 hPa

Pressure Accuracy: ±0.1 hPa

Temperature Measurement Range: 0° C to +65° C

Output Type: I2C protocol

II. Barometric Pressure Detection Program

Detect the barometric pressure in the air through programming, and predict weather conditions based on changes in barometric pressure.

Click here to obtain the barometric pressure detection program.

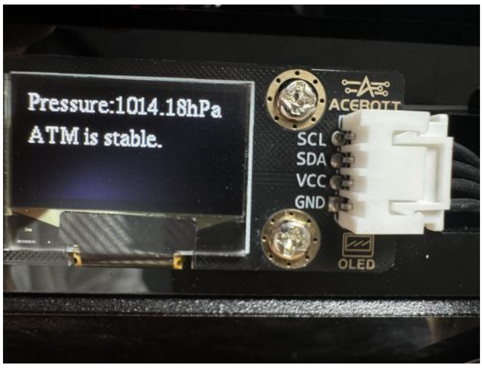
The reference program is as follows:

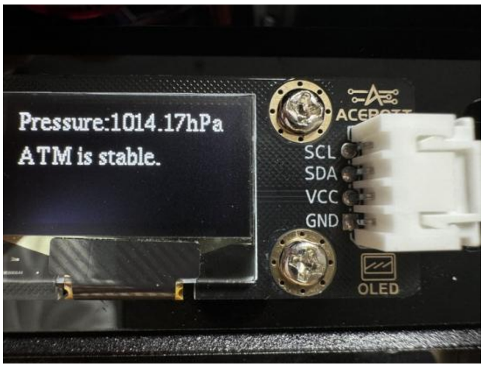
```
#include <ACB_Atmospheric.h>
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library
```

```
ACB_Atmospheric BMP;//creat BMP object
//creat an OLED object:
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

float previousPressure = 0;//creat variable to record previous pressure
float currentPressure = 0;//creat variable to record current pressure
int interval = 5000;
int previousMillis = 0;
String pressureState;
void setup() {
  Serial.begin(115200); //initial Serial port
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
  BMP.Atmospheric_init();//initial barometric pressure sensor
  currentPressure = BMP.read("Press");//read barometric pressure value
  previousPressure = currentPressure;
}

void loop() {
  int currentMillis = millis();//read current time value
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    currentPressure = BMP.read("Press");
    if(currentPressure > previousPressure + 1) {
      Serial.println("ATM is rising.");
      pressureState = "ATM is rising.";
    } else if(currentPressure < previousPressure - 1) {
      Serial.println("ATM is dropping.");
      pressureState = "ATM is dropping.";
    } else {
      Serial.println("ATM is stable.");
      pressureState = "ATM is stable.";
    }
    previousPressure = currentPressure;
  }
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("Pressure:"+String(currentPressure)+"hPa" );//display barpmetric pressure
  u8g2.setCursor(0,30);//set position of cursor
  u8g2.print(String(pressureState));//barpmetric pressure state
  u8g2.sendBuffer();//send text to buffer
  delay(50);
```

```
}
```

After uploading the program, you can see the following effect on the OLED screen, displaying the atmospheric pressure value and indicating the change status of the pressure.





Lesson 11: Air Quality Detection

On a sunny day, a village guide led Dr. Lumi and his students on an adventure around the future village. They unexpectedly discovered a towering volcano about 8 kilometers away from the future village. According to the village guide, the volcano erupts periodically, and each eruption covers the sky with thick black smoke. Dr. Lumi provided a scientific explanation for this phenomenon: "This area is located at the junction of the Earth's tectonic plates, where the interaction of the plates causes magma from the mantle to rise, forming a volcano. When the volcano erupts, it releases a large amount of black smoke, which is mainly composed of volcanic ash and various gases."

Dr. Lumi further emphasized the hazards of volcanic ash and gases, explaining in detail: "Volcanic ash consists of very fine glass fragments, mineral particles, and rock fragments that can be dispersed by the wind for hundreds or even thousands of kilometers, affecting the air quality over a wide area. Inhaling volcanic ash can cause serious damage to the respiratory system, leading to difficulty breathing, throat pain, and lung diseases. In addition, volcanic eruptions also release harmful gases such as sulfur dioxide, carbon dioxide, and hydrogen chloride, which can cause acid rain and air pollution."

Considering the potential threat of volcanic eruptions to air quality and the health of the villagers, Dr. Lumi decided to establish an air quality monitoring system. This system will help villagers monitor air quality in real-time and remind them to take appropriate preventive measures when necessary.

You will now help the villagers build an air quality monitoring system. Before completing this task, you need to understand knowledge related to air quality.

I. Air Quality Indicators

Air Quality Index (AQI) is an indicator used to measure the condition of air quality and alert the public to the degree of air pollution. AQI usually includes the concentrations of various pollutants, with common indicators including PM2.5 (particulate matter with a diameter less than or equal to 2.5 micrometers), PM10 (particulate matter with a diameter less than or equal to 10 micrometers), sulfur dioxide (SO_2), nitrogen oxides (NOx), carbon monoxide (CO), etc. Among them, PM2.5 is the most commonly used air quality indicator, and its unit is $\mu g/m^3$.

According to the U.S. Environmental Protection Agency (EPA) Air Quality Index (AQI) standards, PM2.5 concentration is divided into six levels:

PM2.5 Concentration ranges (in µg/m³)	AQI	Health Impact Descriptions
0-12	Good	The air quality is satisfactory, with little to no air pollution, posing no health risks.
12. 1-35. 4	Moderate	The air quality is acceptable, but a very small number of sensitive individuals may experience health effects.
35. 5 - 55. 4	Unhealthy for Sensitive Groups	Sensitive groups may experience health effects, but the impact on the general population is minimal.
55. 5 - 150. 4	Unhealthy	Everyone may begin to experience health effects, with more pronounced impacts on sensitive groups.
150. 5 - 250. 4	Very Unhealthy	Health alert: Everyone is at increased risk of experiencing significant health effects.
250. 5 - 500	Hazardous	Severe health alert: All groups in the area may be affected, potentially leading to a health emergency.

II. PM2.5 Sensor

1. Sensor Description

This project will use the D01 dust sensor to detect the PM2.5 levels in the air. This dust sensor product utilizes the principle of optical irradiation, measuring the dust concentration within the detection range through the conversion of the optical path and the circuit, and can sensitively detect dust particles with a diameter of more than 1 μm. It directly applies a microcontroller for UART communication. The product is small in size, high in precision, low in power consumption, wide in measurement range, short in response time, and convenient for application.

2. Parameter Description

Detection Range: PM0.3 to PM10.

Detection Range: 5 to 2500μg/m³ .

Operating Voltage: 5V.

Output Type: UART.

III. Air Quality Detection Program

Implement air quality detection through programming, and remind villagers to take protective measures when the air quality exceeds the target value.

Click here to obtain the air quality detection program.

The reference program is as follows:

```
#include <U8g2lib.h>//import OLED library
#include <Wire.h>//import IIC library
#include <ACB_PM25.h>

ACB_PM25 PM25;//creat a PM25 object
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

void setup() {
  u8g2.setI2CAddress(0x3C*2);//set IIC address
  u8g2.begin();//initial OLED
  u8g2.enableUTF8Print();//enable Unicode print
  Serial.begin(115200);
  PM25.setpin(17,16);//initial PM2.5 sensor
}

void loop() {
  float PM25Value = PM25.read();//read PM2.5 value
  String airQuality;
  if (PM25Value <= 35){
    airQuality = "Air quality is good.";
  }
  else {
    airQuality = "The air is polluted.";
  }
  u8g2.clearBuffer();//clear buffer
  u8g2.setFont(u8g2_font_timR10_tf);//set font
  u8g2.setCursor(0,10);//set position of cursor
  u8g2.print("PM2.5:"+String(PM25Value)+"μg/m³ " );//display PM2.5 value
  u8g2.setCursor(0,30);//set position of cursor
  u8g2.print(String(airQuality));//display air quality
```

```
  u8g2.sendBuffer();//send text to buffer
  delay(500);
}
```

After uploading the program, you can see the following effect on the OLED screen, displaying the air quality value and the air quality status.

Note: If the measured PM25 value is too high, remove the cover of the sensor, the lens inside the sensor, and let the lens clean.

Lesson 12: Solar Power Supply

Due to the remote location of Future Village, the supply of energy has always faced many inconveniences. The villagers have long been in a predicament of energy shortage, which not only affects their daily life but also restricts the development of the village. To alleviate this issue, Dr. Lumi specially considered the problem of energy supply when designing the smart weather station. He decided to install solar panels in the design of the weather station to provide the necessary electricity for the station using renewable energy.

Solar panels have high reliability and low maintenance costs, which can provide long-term and stable power support for the weather station. In this way, the villagers no longer have to worry about energy shortages affecting the normal operation of the weather station. This ensures that the weather station can continue to provide accurate meteorological information for the village, helping villagers better cope with weather changes and improve the quality of life.

Solar panels can convert sunlight into electrical energy, providing a continuous supply of green energy for the operation of the weather station. This not only reduces dependence on traditional energy sources, reduces energy consumption, but also reduces carbon emissions, which helps protect the environment.

Next, you will implement the solar power supply function for the weather station. Before completing this task, you need to understand the knowledge related to solar power sources.

I. Solar Charging Panel

1. Module Description

The basic working principle of solar panels is the photovoltaic effect. When sunlight shines on the silicon material of the panel, photons are absorbed, exciting the electrons of silicon atoms and causing them to detach from the atoms. This process generates free electrons and "holes" in the silicon material. The electric field inside the photovoltaic cell (formed by doped p-type and n-type silicon) pushes these free electrons to one side, while holes move to the other side, thus forming an electric current. This flowing current is transported to the load through wires.

2. Types of Solar Charging Panel

(1) Monocrystalline solar panels: Made from monocrystalline silicon, they have high efficiency and long service life, but the cost is relatively high.

(2) Polycrystalline solar panels: Made from polycrystalline silicon, they have lower costs, but the efficiency is slightly lower than monocrystalline silicon.

(3) Thin-film solar panels: Manufactured using thin-film technology, they are flexible and lightweight, suitable for different applications, but have lower efficiency.

The tutorial uses monocrystalline silicon solar panels, with a photoelectric conversion efficiency between 15% and 22%.

3. Parameter Description

Operating Voltage: 5.52V

Operating Current: 500mA

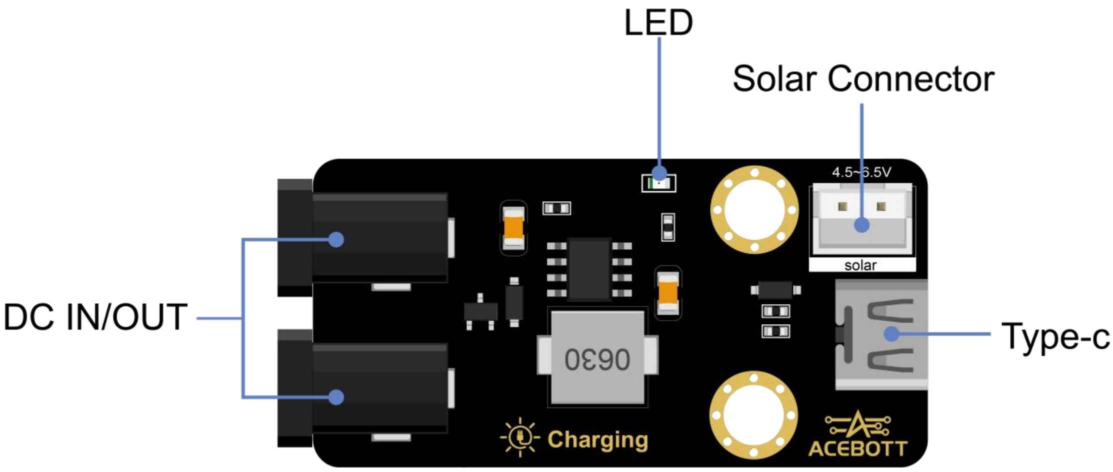
Product Dimensions: 110*162mm

II. Charging Module

1. Product Description

The charging module is equipped with solar charging capabilities, which allow it to harness solar energy to charge lithium batteries under specific lighting conditions. In addition, it features power protection functions, providing an efficient and reliable power management solution for various solar projects. Besides solar charging, this module also comes with a power adapter charging interface, enabling charging even when there is no sunlight exposure.

2. Interface Description



Solar Connector Interface: Power input interface, connects to the solar panel, inputs solar power, and can accept a power input of 4.5V to 6.5V.

Type-C Interface: Power input interface, connects to a power adapter, and can accept a power input of up to 5V.

DC IN/OUT Interface: Power input/output interface. When one of the interfaces is connected to a lithium battery, it can charge the lithium battery, and at the same time, the stored electrical energy of the lithium battery can be output and used through the other interface.

LED: Charging indicator light, which lights up during the charging process of the lithium battery and automatically goes out once the battery is fully charged.

3. Charging Instruction

The charging time of lithium batteries is mainly affected by factors such as battery capacity, charging current, battery voltage, charging method, and battery life. Therefore, the charging time parameter of lithium batteries is only a reference time under the conditions of this test environment, and the actual charging time may vary.

(1) Adapter Charging

Using a 5V/2A charger to charge two 18650 batteries (with a capacity of 4800mAh) to 75% power state takes approximately 1 hour and 42 minutes; to fully charge, it takes about 5 hours.

Note: The adapter specification must be less than 5V voltage.

(2) Solar Charging

On a sunny day, using solar energy to charge an 18650 battery (with a capacity of 4800mAh) to a 50% power state takes about 3.5

hours; to charge to an 85% power state, it takes about 12 hours.

Note: The Solar Charging Panel must use a model with a working voltage of 6V and not exceeding 3W.

(3) USB Charging

If using a computer's USB port to charge an 18650 battery (with a capacity of 4800mAh) to a 90% power state, it takes about 6 hours; to fully charge, it takes about 9 hours.

(4) Simultaneous Charging with Solar and Adapter

If using solar energy and a 5V/2A charger to charge an 18650 battery (with a capacity of 4800mAh) simultaneously under specific conditions, to charge to an 80% power state, it takes about 1 hour and 40 minutes; to fully charge, it takes about 4 hours and 50 minutes.

Lesson 13: Sun-Tracking Solar Panel

After a period of use, the villagers keenly observed that the solar panels charge the fastest at noon, while the charging speed is slower in the morning and afternoon. In response to this phenomenon, Dr. Lumi provided the following explanation: "The intensity of sunlight in the morning and afternoon is not as strong as at noon, hence the converted energy is relatively less. Additionally, a key factor is that the solar panels' ability to absorb sunlight is closely related to the angle of incidence of the light. When the panels are perpendicular to the sunlight, they can absorb the maximum amount of light energy, thus achieving the highest power generation efficiency. At noon, the sunlight is almost perpendicular to the panels, so they absorb the maximum amount of energy, and naturally, the charging efficiency is the fastest."

Dr. Lumi then offered a plan to optimize the use of solar panels, saying: "To keep the solar panels absorbing solar energy at maximum efficiency, it is necessary to maintain a perpendicular state with the sunlight, which requires adjusting the tilt angle of the panels according to the season or the position of the sun during the day to maximize sunlight absorption. In the smart weather station, the solar panels are mounted on a servo, and we can adjust the servo's degrees to change the tilt angle of the panels."

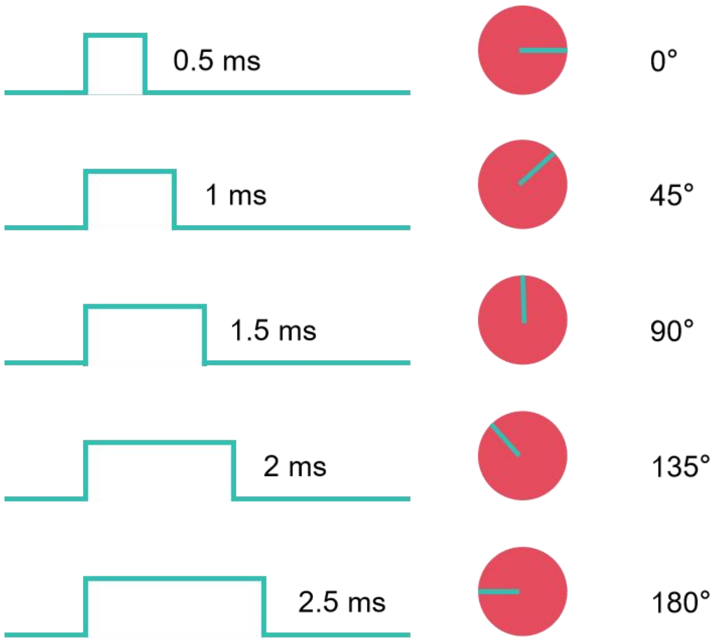
Next, you will implement the function of adjusting the tilt angle of the solar panels. Before taking on this challenge, you need to understand the knowledge related to servos.

I. Servo

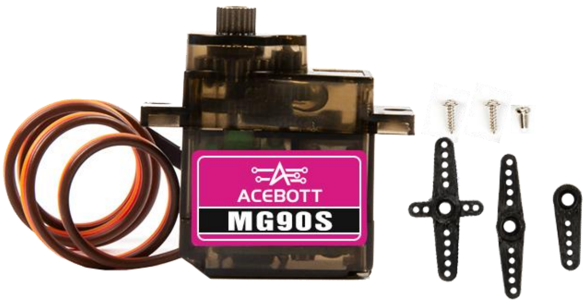
1. Product Introduction

A servo is a type of actuator that can precisely control position, speed, and acceleration, and is widely used in fields such as robotics, modeling, and automated control systems. A servo typically consists of an electric motor, a gear set, a control circuit, and a position sensor (usually a potentiometer).

Servos are usually controlled by PWM (Pulse Width Modulation) signals. PWM is a type of pulse signal, and its duty cycle (the proportion of the high-level time of the pulse to the total cycle time) determines the target position of the servo. A common PWM signal frequency is 50Hz (a period of 20ms), with the pulse width varying between 0.5ms and 2ms corresponding to a rotation range of 0 to 180 degrees.



In this project, the servo model used in this tutorial is the MG90 9G, which has a servo axis movement range of 0° to 180°.



2. Parameter Description

Operating Voltage: 5V.

Movement Range: 0°-180°.

Control Method: PWM.

3. Control Methods

In this project, commands from the "ESP32Servo.h" library will be used to control the movement of the servo motor.

attach(PIN): Connect the servo motor to the specified pin.

write(angle): Control the servo motor to rotate to the specified angle.

II. Solar Panel Angle Control Program

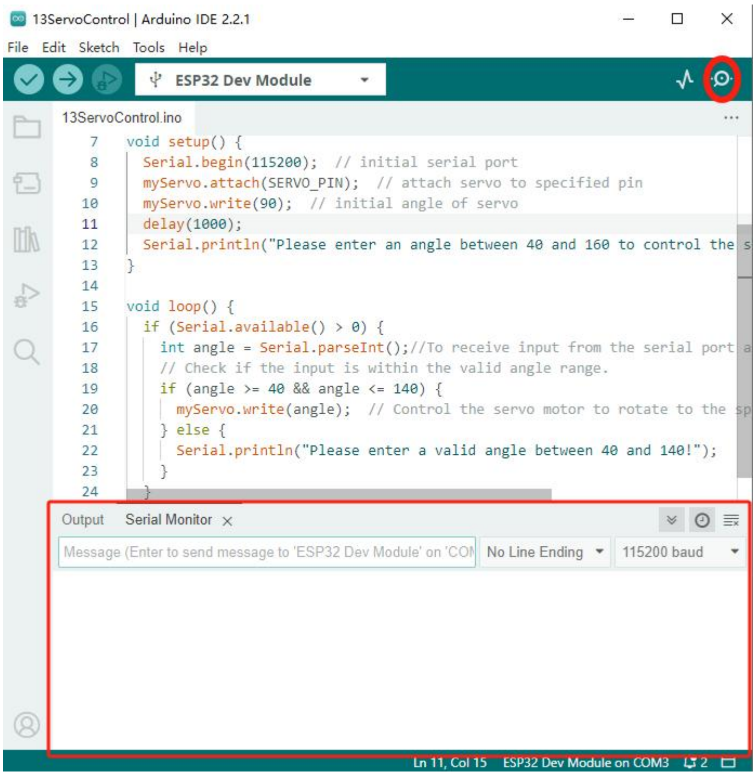
Implement the adjustment of the solar panel's tilt angle through programming, input an angle value from the keyboard, and control the servo to rotate to that angle.

Click here to obtain the solar servo control program.

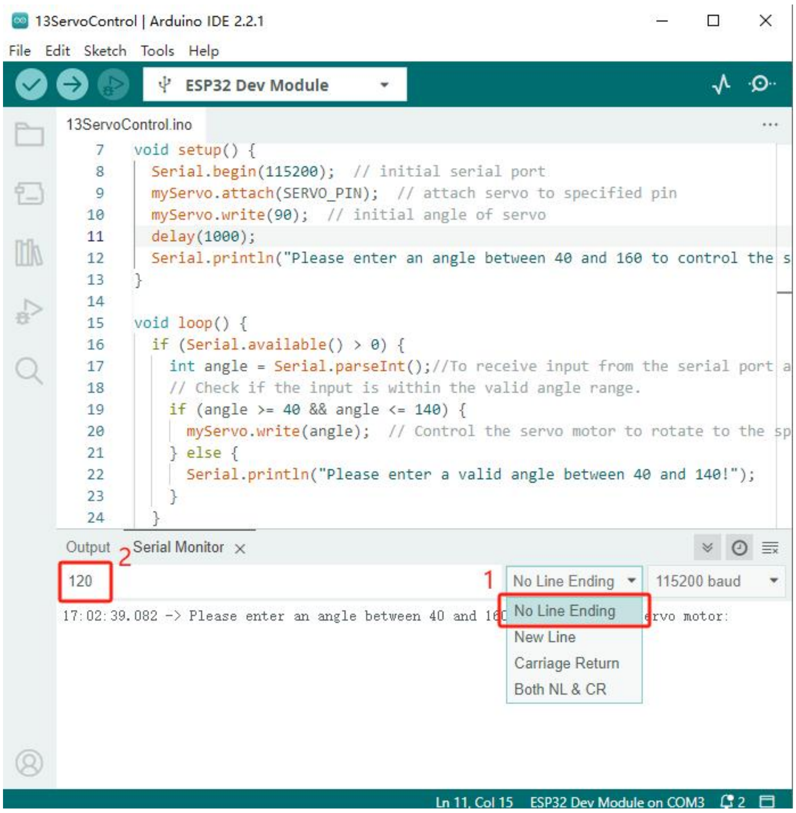
The reference program is as follows:

```
#include <ESP32Servo.h>//import servo library

#define SERVO_PIN 4 //define pin of servo

Servo myServo;   // creat Servo object

void setup() {
  Serial.begin(115200);   // initial serial port
  myServo.attach(SERVO_PIN);   // attach servo to specified pin
  myServo.write(90);   // initial angle of servo
  Serial.println("Please enter an angle between 60 and 140 to control the servo motor:");
}

void loop() {
  if (Serial.available() > 0) {
    int angle = Serial.parseInt();//To receive input from the serial port and convert it to an integer
    // Check if the input is within the valid angle range:
    if (angle >= 60 && angle <= 140) {
      myServo.write(angle);   // Control the servo motor to rotate to the specified angle.
    } else {
      Serial.println("Please enter a valid angle between 60 and 140!");
    }
  }
}
```

After uploading the program, first click the button in the upper right corner of the Arduino IDE to enter the Serial Debugging interface.



Then, select "No Line Ending" from the dropdown menu on the right side of the input box. Enter the angle value for the servo in the serial input box and press the Enter key; the servo will move to the specified angle position.


13ServoControl | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
13ServoControl.ino
void setup() {
Serial.begin(115200); // initial serial port
myServo.attach(SERVO_PIN); // attach servo to specified pin
myServo.write(90); // initial angle of servo
delay(1000);
Serial.println("Please enter an angle between 40 and 160 to control the s
}
void loop() {
if (Serial.available() > 0) {
int angle = Serial.parseInt();//To receive input from the serial port a
// Check if the input is within the valid angle range.
if (angle >= 40 && angle <= 140) {
myServo.write(angle); // Control the servo motor to rotate to the sp
} else {
Serial.println("Please enter a valid angle between 40 and 140!");
}
}
Output
2
Serial Monitor
120
1
No Line Ending
115200 baud
17:02:39.082 -> Please enter an angle between 40 and 16
ervo motor:
No Line Ending
New Line
Carriage Return
Both NL & CR
Ln 11, Col 15 ESP32 Dev Module on COM3

Lesson 14: Comprehensive Display of Meteorological Data

After continuous functional development, the smart weather station's capabilities have gradually been perfected, enabling the monitoring of various meteorological data. This assists the villagers in timely responding to adverse weather and natural phenomena, reducing potential harm to personal safety and property. However, in previous projects, the smart weather station could only display the monitoring results of a single meteorological data, and could not present other related meteorological information simultaneously, which limited the villagers' ability to obtain comprehensive meteorological data at the same time. To address this issue, Dr. Lumi decided to take optimization measures to display all meteorological data simultaneously on the OLED screen of the smart weather station.

At this point, a student raised a question: "Given the small size of the OLED screen, will there be enough space to display all the data?"

Dr. Lumi responded approvingly, "Good question! If the OLED screen cannot display all data at once, we can adopt a multi-page display method."

Dr. Lumi further explained, "The so-called OLED multi-page display refers to displaying different content on the same OLED screen by switching between different pages or interfaces. Each page can independently display a set of specific information or images."

Next, you will implement the task of comprehensive display of meteorological data.

Click here to obtain the program for comprehensive display of meteorological data.

The reference program is as follows:

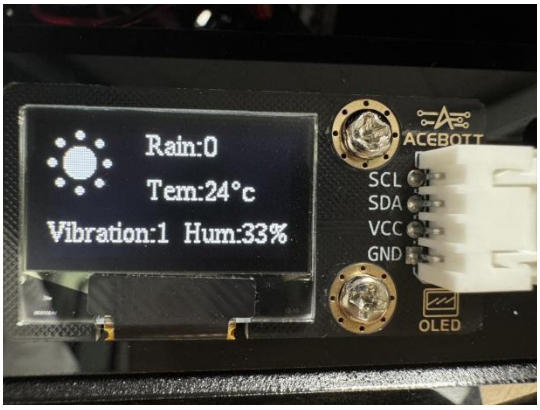
```
#include <Arduino.h>
#include <Wire.h>
#include <U8g2lib.h>
#include <ACB_DHT11.h>
#include <ACB_Ultraviolet.h>
#include <ACB_WindCup.h>
#include <ACB_PM25.h>
#include <ACB_Atmospheric.h>

#define raindropPin 39
#define vibrationPin 18
#define DHT11PIN 23
#define UVPin 33
#define lightPin 36
#define WindPin 32
#define PM25RX 17
#define PM25TX 16

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);
ACB_DHT11 dht11(DHT11PIN);
ACB_Ultraviolet uv;
ACB_Atmospheric BMP;
ACB_WindCup wind;
ACB_PM25 PM25;

int raindrop;
```

```
int vibration;
int tem;
int hum;
int uv_level;
float barometric_pressure;
int light;
int windSpeed;
int PM25_value;
void setup() {
  Serial.begin(115200);
  u8g2.setI2CAddress(0x3C*2);
  u8g2.begin();
  u8g2.enableUTF8Print();
  uv.setpin(UVPin);
  PM25.setpin(PM25RX,PM25TX);
  wind.setpin(WindPin);
  BMP.Atmospheric_init();
}

void loop() {
  raindrop = analogRead(raindropPin);
  vibration = digitalRead(vibrationPin);
  tem = dht11.get_Temperature_Data();
  hum = dht11.get_Humidity_Data();
  uv_level = uv.read("level");
  barometric_pressure = BMP.read("Press");
  light = analogRead(lightPin);
  windSpeed = wind.read();
  PM25_value = PM25.read();
  Page1();
  Page2();
}
void Page1() {//define the content of page 1
  u8g2.clearBuffer();//clear buffer
  if(raindrop>1000){
    u8g2.setFontPosBottom();
    u8g2.setFont(u8g2_font_open_iconic_all_4x_t);
    u8g2.drawGlyph(0,0+4*8,241);
  }
  else{
    u8g2.setFontPosBottom();
    u8g2.setFont(u8g2_font_open_iconic_all_4x_t);
    u8g2.drawGlyph(0,0+4*8,259);
  }
```

```
  u8g2.setFont(u8g2_font_timR10_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(50,0);
  u8g2.print(String("Rain:")+String(raindrop)   );

  u8g2.setFont(u8g2_font_timR10_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(50,23);
  u8g2.print(String("Tem:")+String(tem)+String("° c")   );

  u8g2.setFont(u8g2_font_timR10_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(70,45);
  u8g2.print(String("Hum:")+String(hum)+String("%")   );

  u8g2.setFont(u8g2_font_timR10_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,45);
  u8g2.print(String("Vibration:")+String(vibration));

  u8g2.sendBuffer();
  delay(2000);
}

void Page2() {//define the content of page 2
  u8g2.clearBuffer();
  u8g2.setFont(u8g2_font_timR08_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,0);
  u8g2.print(String("UV Level:")+String(uv_level));

  u8g2.setFont(u8g2_font_timR08_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,12);
  u8g2.print(String("Wind Speed:")+String(windSpeed)+String("m/s"));

  u8g2.setFont(u8g2_font_timR08_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,24);
  u8g2.print(String("Lightness:")+String(light));

  u8g2.setFont(u8g2_font_timR08_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,36);
```

```
  u8g2.print(String("PM2.5:")+String(PM25_value)+String("μg/m³ "));

  u8g2.setFont(u8g2_font_timR08_tf);
  u8g2.setFontPosTop();
  u8g2.setCursor(0,48);
  u8g2.print(String("ATM:")+String(barometric_pressure)+String("hPa"));

  u8g2.sendBuffer();
  delay(2000);
}
```

After uploading the program, you can see the following effect on the OLED screen, where all the data is displayed on the OLED, and these data are displayed by switching through different pages.



Lesson 15: Weather Station Website

In the latest feedback session with the villagers, they expressed their desire to access weather data remotely, so they can obtain this information anytime and anywhere without frequently visiting the weather station. To meet the villagers' needs and make it more convenient for them to obtain meteorological information, Dr. Lumi decided to use Internet of Things (IoT) technology to build a weather website. This way, villagers can query meteorological data through the website at any time.

Next, you will take Dr. Lumi's place to build a weather website. Before completing this challenge, you need to understand the relevant knowledge about websites.

I. Internet of Things (IoT)

The Internet of Things (IoT) refers to the interconnection of various physical devices (such as sensors, home appliances, vehicles, industrial equipment, etc.) through the internet to achieve data collection, sharing, and remote control. These devices communicate and collaborate with each other through built-in sensors, software, and other technologies, forming an intelligent network system.

II. WiFi Technology

WiFi communication technology is a wireless local area network (WLAN) technology that allows electronic devices such as smartphones, tablets, laptops, etc., to connect to the internet or a local area network wirelessly. WiFi communication technology connects devices to the same network through a wireless router or access point (AP), allowing devices to receive and send data to each other.

III. Steps to Access the Weather Website

1. Upload the weather website program

Before logging into the website, we need to connect the ESP32 controller board to a WiFi hotspot, create a server for the weather website through the WiFi network, and upload the meteorological data to the server. Before uploading the program, set the name and password of the WiFi hotspot you want to connect to, ensuring that the phone or other mobile devices are connected to the same WiFi network as the smart weather station. Open the weather website program and find the following instruction; you can modify this instruction to set the WiFi name and password.

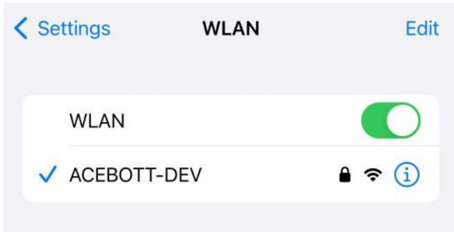
```
// set wifi name
#define WIFI_SSID "ACEBOTT"
// set wifi password
#define WIFI_PASSWORD "12345678"
```

Click here to obtain the program, upload it to the ESP32 controller board, and turn on the power for the weather station.

Note: There are two ways to connect the ESP32 to the same network as your phone or computer.

1. Connect your phone or computer and the ESP32 to the same WiFi network.

```
WebServer server(80);
// WiFi user name and password
const char* ssid = "ACEBOTT-DEV";
const char* password = "12345678";
```

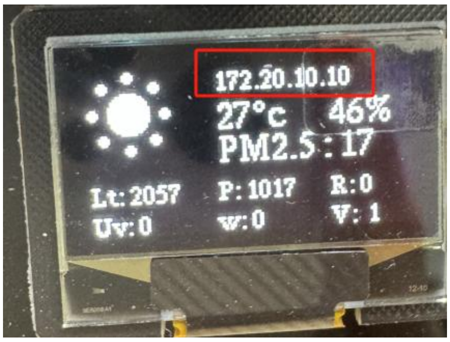


2. Enable and set up a mobile hotspot on your phone, change the ssid and password in the ESP32 program to match the name and password of the mobile hotspot, upload the ESP32 program, and have the ESP32 connect to the hotspot on your phone.

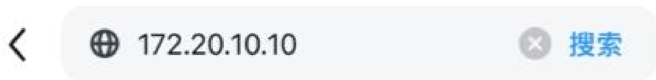
```
#include <ESPmDNS.h>
#include <WiFiClient.h>
String item = "0";
const char* ssid = "ACEBOTT";//char
const char* password = "12345678";
```

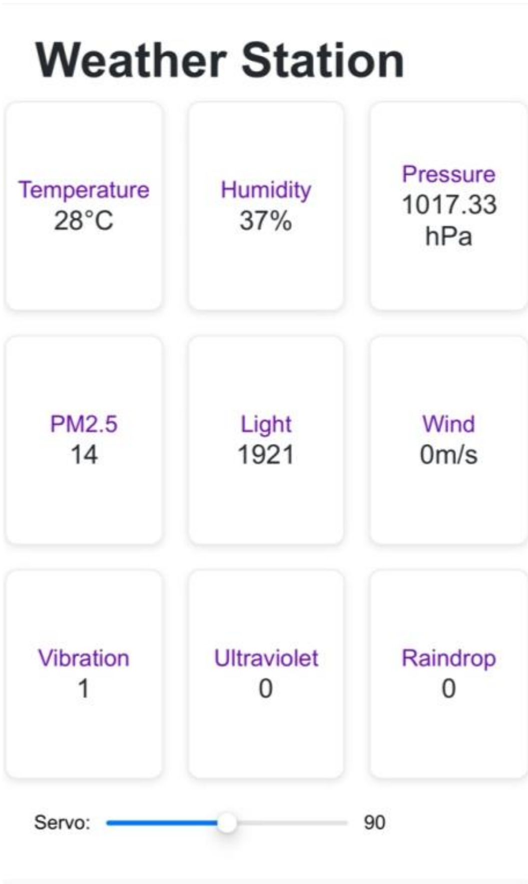
2. Access the Weather Website

After connecting your phone to the same network as the smart weather station, locate the IP address of the smart weather station on the OLED screen.



Open your mobile browser and enter the IP address of the smart weather station in the address bar to access the webpage. On the webpage, you can obtain various meteorological data and control the movement of the servo.




Weather Station
Temperature
28°C
Humidity
37%
Pressure
1017.33
hPa
PM2.5
14
Light
1921
Wind
0m/s
Vibration
1
Ultraviolet
0
Raindrop
0
Servo:
90

Lesson 16: Weather Station APP

In order to further enhance the villagers' experience with the weather station services, Dr. Lumi decided to develop a dedicated meteorological APP. This way, the villagers would be able to more conveniently and intuitively obtain the meteorological data they need. After several days of intense and orderly development work, Dr. Lumi finally announced to everyone that the meteorological application has been successfully developed.

The next task falls on your shoulders; you need to be responsible for demonstrating and explaining the specific usage methods of this meteorological APP to the villagers. Ensure that every villager can master how to use this application to obtain the meteorological information they need, so as to better arrange their daily life and agricultural activities.

I. Download the APP

If you have an iOS system phone, you need to search for the keyword: ACEBOTT in the APP Store and then download it; if you have an Android system phone, you need to search for the keyword: ACEBOTT in Google Play Store and then download it. The icon is shown in the figure below.

Note:

1. This tutorial is applicable to ACEBOTT APP version 2.0 and above. You can click the settings button in the top left corner of the APP to view the software version number. Please ensure that the software version you are using meets the requirements;

2. If you need to update the ACEBOTT software version, you can follow the instructions in this manual to download the latest APP version.

II. Connect to WiFi

Upload the meteorological station APP program and ensure that the smart weather station and the phone are connected to the same WiFi network (see the previous lesson for specific methods).

Click here to obtain the program for the smart weather station APP.

III. Using the Weather Station APP

1. Open the ACEBOTT APP, scroll through the page, and find

"Weather Station," as shown in the figure below.

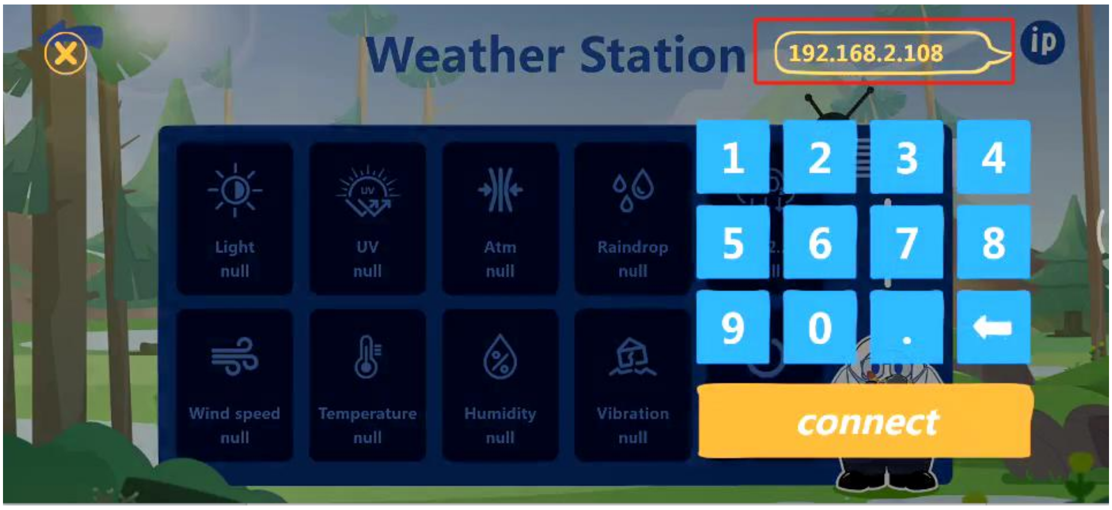
Note: If you need to watch the APP operation video, please click the link below.

https://youtu.be/OMD2zjySJ8Q

After clicking, select "Control" to enter the control page.

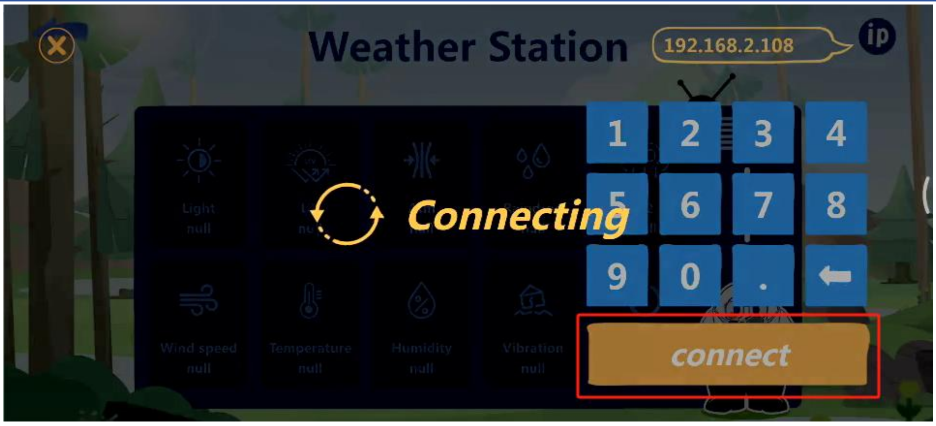
Note: You can click the build button on the right to view the assembly video of this project.

2. Tap on the "Weather Station" image to enter the smart weather station page, then tap on the IP icon in the top right corner and enter the IP address of the smart weather station.

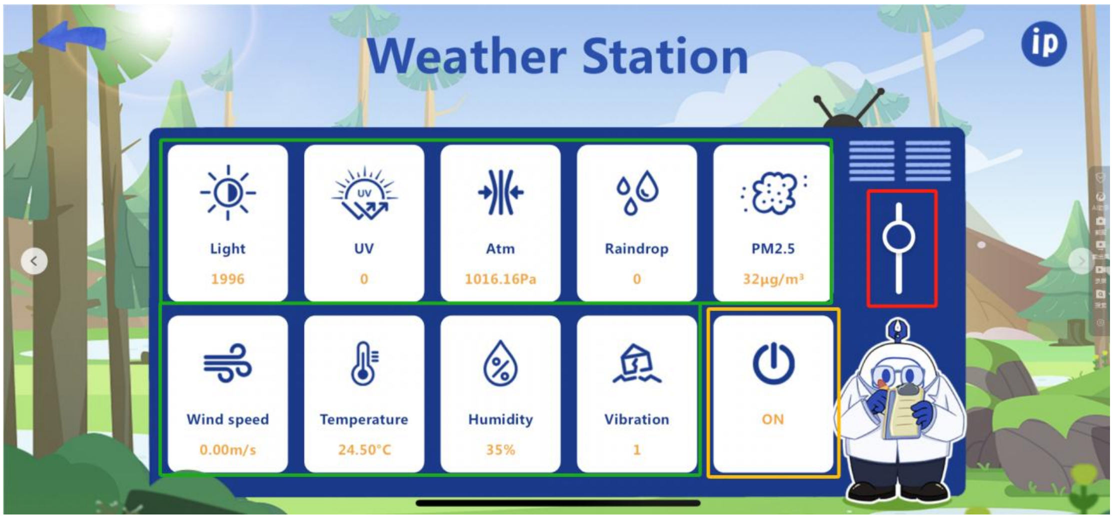


Once the smart weather station successfully connects to WiFi, you can obtain the IP address of the smart weather station through the OLED screen.

3. After entering the IP address of the smart weather station, click the "connect" button to connect to the server of the smart weather station.



4. After the connection is established, you can click on various icons to retrieve the corresponding meteorological data.



Clicking on the icons within the green box allows you to independently turn on and off the display of each sensor's data, including light intensity, UV level, atmospheric pressure, raindrop sensor, PM2.5, wind speed, temperature, humidity, and vibration sensor values.

Clicking the icon in the yellow box enables you to turn on or off the display of all data with one click.

Sliding the slider within the red box controls the movement of the servo for the solar panel, with a movement range of 60° to 140°.