

1. Course Description
2. Download supporting resources
3. Development Kit Introduction
  - 3.1. Introduction
  - 3.2. Hardware composition & technical parameters
  - 3.3. Circuit Schematic & PCB Diagram - Standard Board
  - 3.4. Circuit Schematic & PCB Diagram - MiNi Board
  - 3.5. CC2530F256 module specifications
  - 3.6. WiFi module specifications
  - 3.7. Communication distance test
4. Part 1: Preparation
  - 4.1. Introduction to ZigBee 3.0 that even a novice can understand
  - 4.2. IAR EW for 8051 Introduction and Installation
  - 4.3. TI Z-Stack 3.0 Introduction and Installation
  - 4.4. SmartRF Flash Programmer Download and Installation
  - 4.5. Serial Port Assistant Introduction and Installation
  - 4.6. SmartRF04EB Driver
  - 4.7. USB to Serial Port Driver
  - 4.8. Other software installation (optional)
    - 4.8.1. Xshell 7 Introduction and Installation Guide
    - 4.8.2. PuTTY Introduction and Installation
5. Part 2: Introduction to 51 MCU - Based on CC2530
  - 5.1. Chapter 1: CC2530 Development Basic Experiment
    - 5.1.1. Create a new workspace and project
    - 5.1.2. Source code writing and compilation
    - 5.1.3. Program download and simulation
    - 5.1.4. Firmware Burning
  - 5.2. Chapter 2: GPIO Experiment
    - 5.2.1. Multi-project management foundation
    - 5.2.2. GPIO output experiment - LED control
    - 5.2.3. GPIO input experiment - mechanical buttons
    - 5.2.4. GPIO input and output general configuration experiment
    - 5.2.5. GPIO external interrupt experiment
  - 5.3. Chapter 3: Timer Experiment
    - 5.3.1. Project Overview
    - 5.3.2. Timer T1 Experiment - Query Trigger
    - 5.3.3. Timer T3 Experiment - Interrupt Trigger
    - 5.3.4. Watchdog Timer Experiment
    - 5.3.5. Low Power Timer Experiment
  - 5.4. Chapter 4: Serial Communication Experiment
  - 5.5. Chapter 5: ADC Experiment - Using the Light Sensor
  - 5.6. Chapter 6: OLED Display Experiment
  - 5.7. Chapter 7: Peripheral Experiments
    - 5.7.1. DHT11 Temperature and Humidity Sensor
    - 5.7.2. NorFLASH reading and writing experiment
    - 5.7.3. Relay Control Experiment
6. Part 3: Z-Stack 3.0 Detailed Explanation
  - 6.1. Chapter 1: Detailed Explanation of Z-Stack 3.0 Architecture
    - 6.1.1. Z-Stack 3.0.1 File Organization
    - 6.1.2. Z-Stack 3.0.1 Project Framework
  - 6.2. Chapter 2: Task Scheduling Principles of Operating Systems
  - 6.3. Chapter 3: OSAL Detailed Explanation
    - 6.3.1. OSAL Task Scheduling Principle
    - 6.3.2. Task initialization and event processing
    - 6.3.3. Application of Z-Stack Events
    - 6.3.4. Using Dynamic Memory
  - 6.4. Chapter 4: Hardware Adaptation Layer Application - LED
    - 6.4.1. HAL file structure and project structure
    - 6.4.2. HAL Architecture Overview
    - 6.4.3. Introduction to LED API
    - 6.4.4. LED Experiment
  - 6.5. Chapter 5: Hardware Adaptation Layer Application - Buttons
    - 6.5.1 Keystroke Experiment
    - 6.5.2 Detailed Explanation of HAL Key Framework (Optional)
  - 6.6. Chapter 6: Hardware Adaptation Layer Application - Serial Port

## **6.7. Chapter 7: Hardware Adaptation Layer Application - Display Screen**

## **6.8. Chapter 8: Hardware Adaptation Layer Application - ADC**

## **7. Part 4: ZigBee 3.0 Network Programming**

### **7.2. Chapter 1: ZigBee 3.0 Network Principles**

7.1.1. Protocol Hierarchy

7.1.2. IEEE 802.15.4 Protocol

7.1.3. Network Layer

### **7.2. Chapter 2: ZigBee 3.0 BDB**

7.2.1. Introduction to BDB

7.2.2. BDB Commissioning Modes

7.2.3. ZigBee 3.0 Networking Experiment

### **7.3. Chapter 3: Data Communication Based on AF**

7.3.1. Simple Descriptor

7.3.2. Communication Principle

7.3.3. Introduction to Data Sending API

7.3.4. ZigBee 3.0 Communication Experiment

### **7.4. Chapter 4: ZCL Basic Principles**

7.4.1. Introduction to ZCL

7.4.2. Detailed explanation of ZCL content

### **7.5. Chapter 5: Switch command transmission and reception based on ZCL**

7.5.1. Application layer calls ZCL API

7.5.2. ZCL switch command sending and receiving API

7.5.3. ZCL switch command sending and receiving experiment

### **7.6. Chapter 6: Attribute reading and writing based on ZCL**

7.6.1. ZCL attribute read and write API

7.6.2. ZCL attribute reading and writing experiment

### **7.7. Chapter 7: Attribute reporting experiment based on ZCL**

7.7.1. Overview

7.7.2. Terminal Equipment Development

7.7.3. Coordinator Device Development

7.7.4. Simulation Debugging

## **8. Extracurricular: Project Practice**

### **8.1. ZigBee 3.0 Environmental Information Collection (1)**

8.1.1. System Introduction

8.1.2. Source code download

8.1.3. Hardware List

8.1.4. System Construction

8.1.5. Source code description

### **8.2. ZigBee 3.0 Environmental Information Collection (2)**

8.2.1. System Introduction

8.2.2. Source code download

8.2.3. Hardware List

8.2.4. Using Tencent Cloud

8.2.5. Download and install MQTT.fx

8.2.6. MQTT.fx connects to Tencent Cloud

8.2.7. MQTT.fx publishes messages

8.2.8. Configuring the Development Board

8.2.9. Burn the coordinator firmware

8.2.10. Burning Router Firmware

8.2.11. Source code description

### **8.3. ZigBee Temperature and Humidity Monitoring & Sound and Light Alarm System (1)**

8.3.1. System Introduction

8.3.2. Source code download

8.3.3. Hardware List

8.3.4. System Construction

8.3.5. Source code description

### **8.4. ZigBee automatic light on - based on light intensity & human body monitoring (1)**

8.4.1. System Introduction

8.4.2. Source code download

8.4.3. Hardware List

8.4.4. System Construction

8.4.5. Source code description

### **8.5. File transmission system based on ZigBee**

### **8.6. ZigBee-based automatic curtain opening and closing**

### **8.7. ZigBee-based temperature, humidity & light intensity acquisition system**

### **8.8. Agricultural environmental information collection based on ZigBee**

8.8.1. System Introduction

8.8.2. Source code download

8.8.3. Hardware List

8.8.4. Using a private cloud server

**8.8.5. Configuring the Development Board**

**8.8.6. Networking and WiFi Connection**

**8.8.7. Using Android APP**

**8.8.8. Source code description**

**8.9. Common problems in project construction**

**9. Extracurricular: Advanced Electives**

**9.1. Description of "Extracurricular: Advanced Elective"**

**9.2. Chapter 1: Serial Communication Protocol Design**

**9.2.1. Design basis**

**9.2.2. Protocol Format**

**9.3. Chapter 2: Optimizing the Coordinator Project Structure**

**9.3.1. Project Structure**

**9.3.2. Detailed explanation of application framework**

**9.3.2.1. Framework Description**

**9.3.2.2. zbmsg**

**9.3.2.3. zbcategory**

**9.4. Chapter 3: Coordinator host computer debugging**

**9.4.1. Host computer description**

**9.4.2. Debugging Instructions**

**9.5. Chapter 4: Dynamic Modification of Channels and PanId**

**9.5.1. Serial Port Protocol**

**9.5.2. Important interface description**

**9.5.2.1. NIB**

**9.5.2.2. NLME\_UpdateNV**

**9.5.3. Architecture Adjustment**

**9.5.4. Application**

**9.5.4.1. zbnwk interface implementation**

**9.5.4.2. Serial communication analysis**

**9.5.4.3. Burning and debugging**

**9.6. Chapter 5: Obtaining Network Short Address and MAC Address**

**9.6.1. Interface Description**

**9.6.1.1. Description**

**9.6.1.2. Calling Process**

**9.6.1.3. Asynchronous Data**

**9.6.2. Debugging**

**9.7. Chapter 6: Network Access Control and Whitelist**

**9.7.1. Basic content**

**9.7.1.1. Network access control**

**9.7.1.2. Whitelist**

**9.7.2. Function Encapsulation**

**9.7.3. Program Debugging**

**9.8. Chapter 7: Coordinator Partition Storage Management**

**9.8.1. Software Framework**

**9.8.2. Application**

**9.8.3. Debugging**

**9.9. ZigBee 2 WiFi - Based on ESP8266**

**9.9.1. Use cloud server**

**9.9.2. Source code description and testing**

**9.10. Capturing and analyzing ZigBee wireless messages**

**9.11. Connect to Xiaomi Aqara smart socket and temperature and humidity sensor**

**9.12. NV Application of Z-Stack**

**9.12.1. Introduction to NV**

**9.12.2. NV reading and writing**

**9.13. HAL-based external FLASH application**

**9.14. TFT display experiment (optional)**

**9.15. Lighting project source code analysis**

**9.15.1. ZHA Lighting Project**

**9.15.2. ZHA Lighting source code analysis**

**9.15.3. Lighting brightness adjustment experiment**

**9.16. TemperatureSensor project source code analysis**

**9.16.1. ZHA TemperatureSensor Project**

**9.16.2. ZHA TemperatureSensor Source Code Analysis**

**9.17. Other extracurricular projects**

**9.17.1. Temperature and harmful gas SMS alarm system based on ZigBee**

**9.17.2. Multi-sensor detection and lighting alarm system based on ZigBee**

**9.17.3. Temperature, humidity, human infrared and sound and light alarm system based on ZigBee**

**9.17.4. ZigBee 3.0 multi-node networking practice**

**9.17.5. Temperature, humidity & signal strength detection system based on ZigBee**

**9.18. IAR EW for 8051 FAQ & Solutions**

# 8. Extracurricular: Project Practice

## 1. Required Reading

This course **is an extracurricular course**, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise **please stop studying!**

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530
- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

## 2. Contents

- 8.1. ZigBee 3.0 Environmental Information Collection (1)
- 8.2. ZigBee 3.0 Environmental Information Collection (2)
- 8.3. ZigBee Temperature and Humidity Monitoring & Sound and Light Alarm System (1)
- 8.4. ZigBee automatic light on - based on light intensity & human body monitoring (1)
- 8.5. File transmission system based on ZigBee
- 8.6. ZigBee-based automatic curtain opening and closing
- 8.7. ZigBee-based temperature, humidity & light intensity acquisition system
- 8.8. Agricultural environmental information collection based on ZigBee
- 8.9. Common problems in project construction

## 8.1. ZigBee 3.0 Environmental Information Collection (1)

### Video Explanation

<https://www.bilibili.com/video/BV1k34y1D7Vz?p=40>

## 1. Required Reading

This course **is an extracurricular course**, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise **please stop studying!**

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530
- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

## 2. Contents

- 8.1.1. System Introduction
- 8.1.2. Source code download
- 8.1.3. Hardware List
- 8.1.4. System Construction
- 8.1.5. Source code description

## 8.1.1. System Introduction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Video Explanation

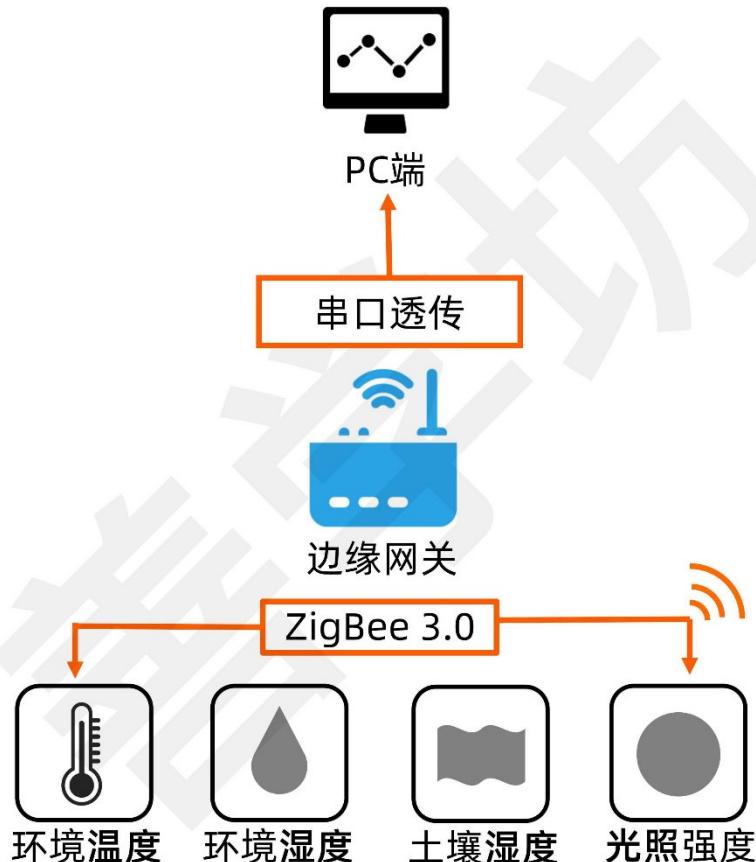
<https://www.bilibili.com/video/BV1k34y1D7Vz?p=40>

### 1. Field of focus

This solution is aimed at the functional verification Demo field with general requirements on communication stability, response speed and communication distance.

### 2. System Architecture

(1) The system architecture is shown in the figure.



- (2) This system consists of an environmental information collection device, an edge gateway and a PC program. The relevant technical principles are as follows:
- (2A) The environmental information collection device is equipped with a temperature and humidity sensor that can sense the ambient temperature;
- (2B) The environmental information collection device is equipped with a soil moisture sensor that can sense soil moisture;
- (2C) The environmental information collection device is equipped with a light intensity sensor that can sense light intensity;
- (2D) The environmental information collection device and the edge computing gateway integrate the ZigBee3.0 cloud module. The environmental information collection device can send the collected environmental information to the edge computing gateway through ZigBee wireless communication;
- (2E) The edge computing gateway receives the environmental information data from each sub-device on the one hand, and sends it to the PC through serial port transparent transmission on the other hand, so that the PC displays the data measurement results;
- (2F) The PC supports saving data in Excel tables for subsequent data analysis and display.

### 3. System Demonstration

The screenshot of the PC host computer effect is shown in the figure.



#### 8.1.2. Source code download

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

#### Source code download

- (1) Users who have purchased the product can download the additional materials from the "Supplementary Materials Download" section in the catalog.
- (2) This item is located in the additional materials as shown in the figure.

<input type="checkbox"/> 文件名	<input type="checkbox"/> 大小
<input type="checkbox"/> 其他项目 1	-
<input type="checkbox"/> 其他项目 2	-
<input checked="" type="checkbox"/> ZigBee 3.0 环境信息采集 (1) .zip	64.58 MB
<input type="checkbox"/> ZigBee 3.0 环境信息采集 (1) (多节点版) .zip	281.51 MB
<input type="checkbox"/> ZigBee 3.0 环境信息采集 (2) .zip	56.09 MB
<input type="checkbox"/> 基于ZigBee的光照自动开关窗帘.zip	38.80 MB
<input type="checkbox"/> 基于ZigBee的农业环境信息采集.zip	69.76 MB
<input type="checkbox"/> 基于Zigbee的温湿度 & 光照度采集系统.zip	33.58 MB
<input type="checkbox"/> 基于ZigBee的温湿度 & 信号强度探测系统.zip	33.51 MB
<input type="checkbox"/> 基于ZigBee的文件传送系统.zip	27.13 MB

## 8.1.3. Hardware List

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Hardware List

- (1) ZigBee 3.0 standard board × 1: <https://item.taobao.com/item.htm?id=683089996879>
- 2) ZigBee 3.0 Mini板 × 1 : <https://item.taobao.com/item.htm?id=683089996879>
- (3) 0.96 OLED12864 display × 2: <https://item.taobao.com/item.htm?id=683089996879>
- (4) SmartRF04EB emulator × 1: <https://item.taobao.com/item.htm?id=683089996879>
- (5) Micro USB data cable × 2: Free when purchasing the development board
- (6) Dry cell battery box × 2 (optional): <https://item.taobao.com/item.htm?id=683115420758>
- (7) Soil moisture sensor & DuPont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

### Buying Guide

**Tip:** You can purchase all the hardware required for this project by purchasing as follows.

(1) ZigBee development kit version selection is shown in the figure.



(2) Select the battery box as shown in the figure.



(3) The soil moisture sensor selection is shown in the figure.



## 8.1.4. System Construction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

## Video Explanation

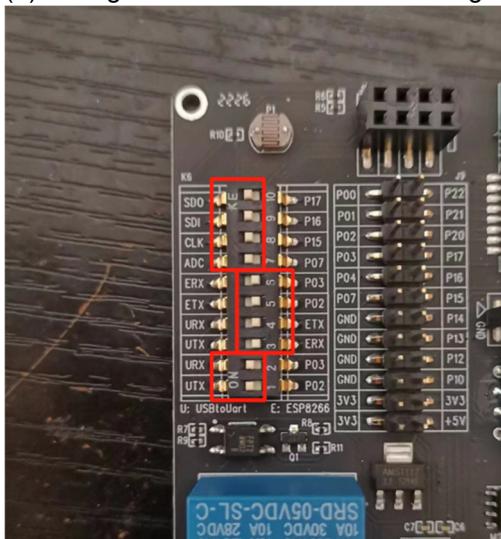
<https://www.bilibili.com/video/BV1k34y1D7Vz?p=40>

## 1. Build a development environment

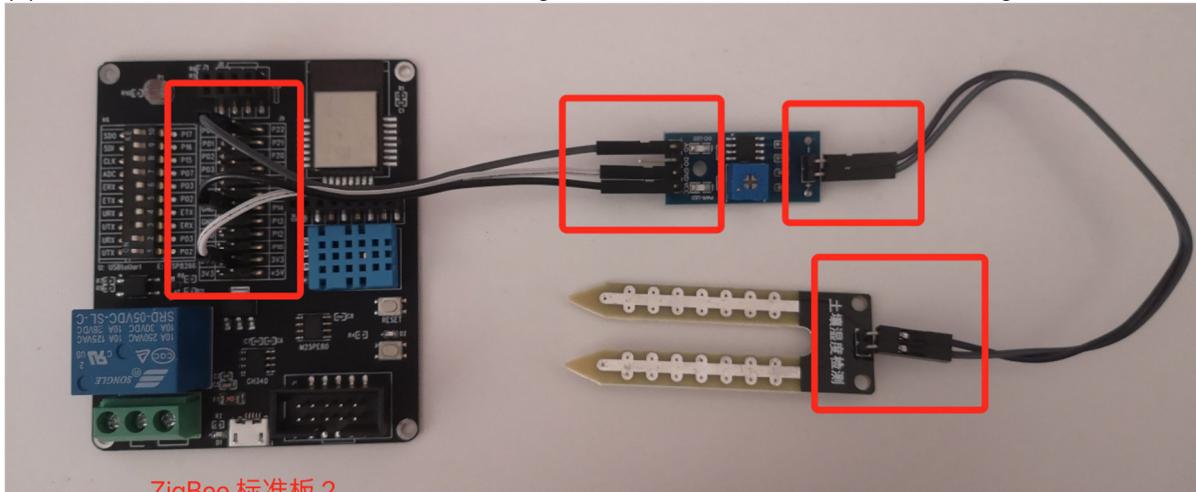
Install the development software according to Part 1: Preparation of this tutorial.

## 2. Configure ZigBee 3.0 standard board

(1) Configure the DIP switches in the ZigBee 3.0 standard board as shown in the figure.



(2) Connect the soil moisture sensor to the ZigBee standard board as shown in the figure.

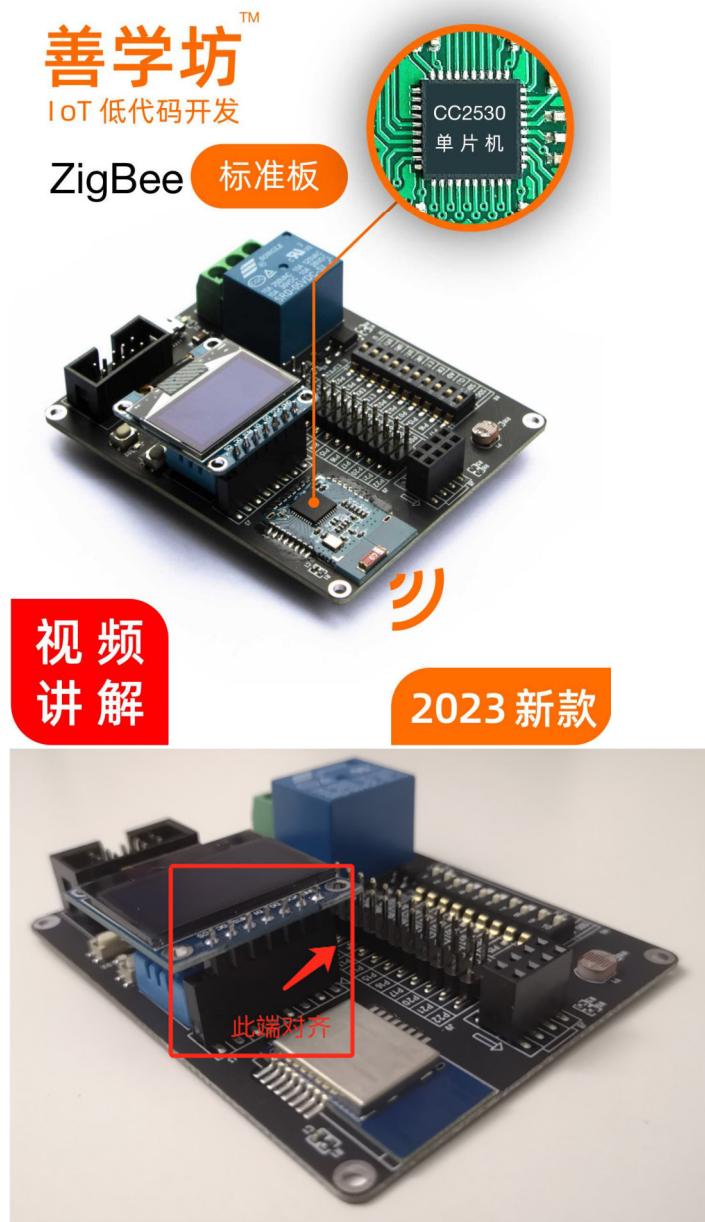


(3) The pin connection instructions are as follows:

- (3A) The soil moisture probe and the sensor main board can be connected with two Dupont wires, and there is no requirement for pin docking;  
(3B) The VCC pin of the sensor main board is connected to any 3v3 pin of the ZigBee standard board;  
(3C) The GND pin of the sensor main board is connected to any GND pin of the ZigBee standard board;  
(3D) The AO pin of the sensor main board is connected to the P0\_0 pin of the ZigBee standard board;  
(3E) The DO pin of the sensor main board is left blank.

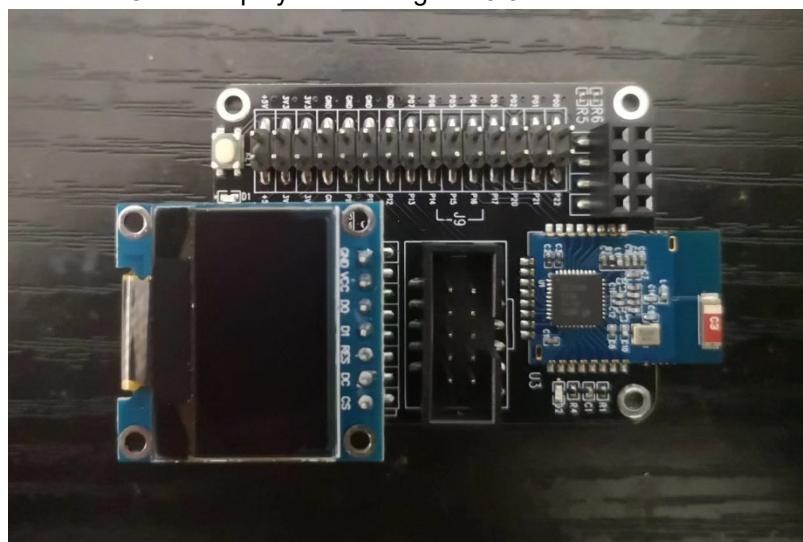
**Tip:** Why does the AO pin above need to be connected to the P0\_0 pin? This is determined by the code written. Readers can also modify the code to use other pins.

(4) Insert the OLED display into the ZigBee 3.0 standard board, as shown in the figure.



### 3. Configure ZigBee 3.0 Mini Board

Insert the OLED display into the ZigBee 3.0 Min board as shown in the picture.



### 4. Burn the firmware

(1) After downloading the source code of this project, find the firmware, as shown in the figure.



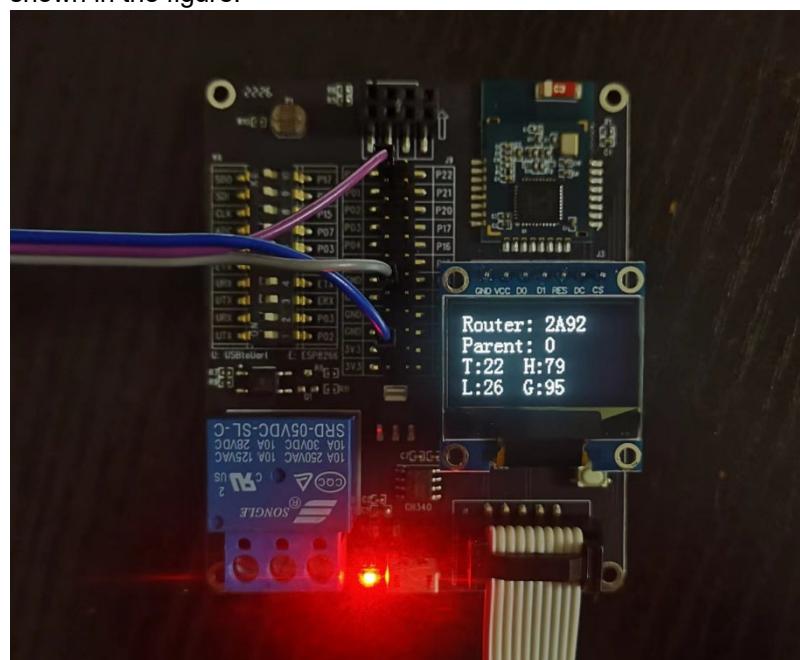
(2) Burn the Coordinator firmware into the ZigBee 3.0 Mini board. **After burning, connect the Mini board to the computer using a Micro USB cable.**

(3) Burn the Router firmware into the ZigBee 3.0 standard board.

**Tip:** For detailed operation steps, please refer to "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: Basic Experiment of CC2530 Development" → "Firmware Burning".

(4) After burning the firmware, use the battery box to power the ZigBee 3.0 standard board.

(5) The ZigBee 3.0 standard board will automatically collect ambient temperature, humidity, light intensity and soil moisture, as shown in the figure.



**Note:** Please follow the above steps! In particular, you must power on the Mini board first, and then burn the ZigBee 3.0 standard board firmware, otherwise the standard board may fail to join the network.

## 5. Networking and serial port communication

(1) The ZigBee 3.0 standard board will automatically join the ZigBee network created by the Mini board and send data to the Mini board.

(2) Find and open the host computer program in the project data, as shown in the figure.



(3) Click the "Path" button to select the storage location for the collected data, as shown in the figure.



(4) After configuring the serial port information as shown in the figure, click the "Link" button.



**Note:** By default, the program will automatically select the correct Mini board port. If it fails to identify correctly, please manually select the correct Mini board port.

(5) If the above operation is correct, after the ZigBee standard board and Mini board are successfully networked, the ambient temperature and humidity, light intensity (reference value) and soil moisture values can be collected and saved in the specified Excel table.

## 8.1.5. Source code description

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions!
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

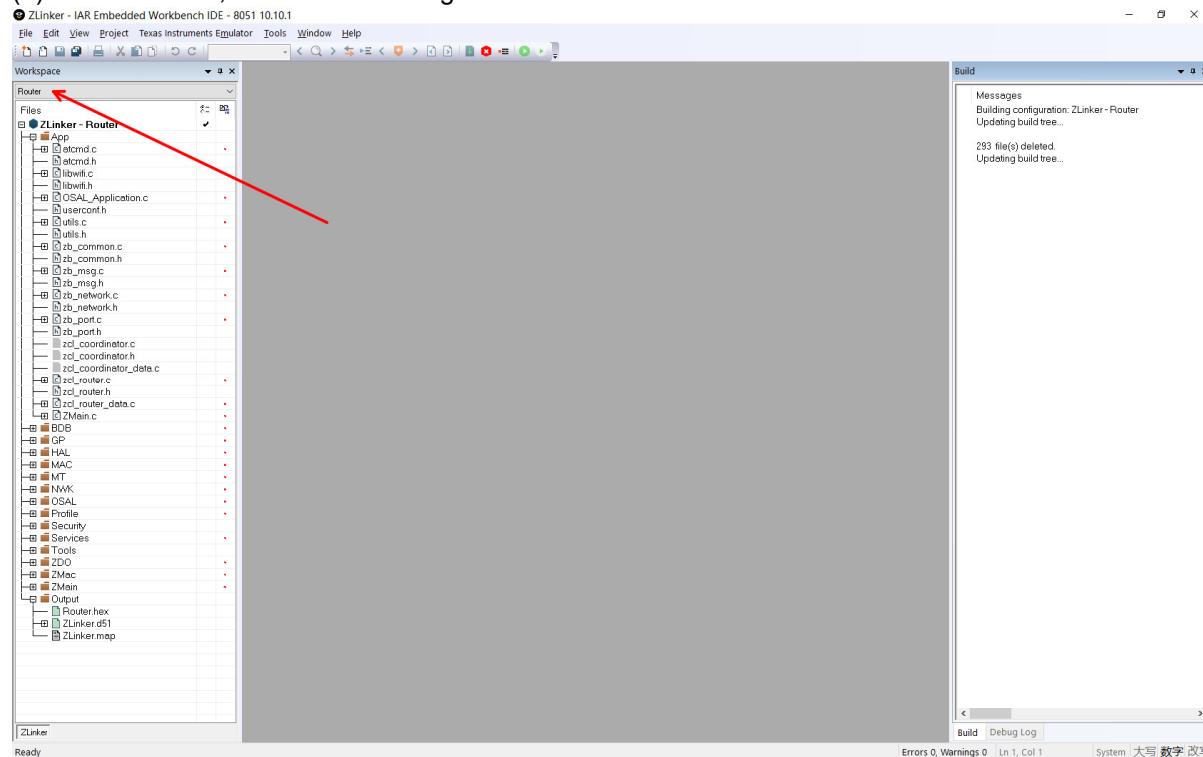
## 1. Compile the source code

(1) Install the development software according to "Part 1: Preparation" of this tutorial.

(2) Use IAR EW for 8051 to open the project file, as shown in the figure.

名称	修改日期	类型	大小
Category	2023/3/18 22:37	文件夹	
Settings	2023/3/18 22:37	文件夹	
Tools	2023/3/18 22:37	文件夹	
BuildLog.log	2021/3/27 15:38	文本文档	1 KB
ZLinker.dep	2023/4/19 13:46	DEP 文件	376 KB
ZLinker.ewd	2021/3/29 0:47	EWD 文件	75 KB
ZLinker.evp	2023/3/19 11:21	EWP 文件	160 KB
ZLinker.ewt	2023/3/18 11:29	EWT 文件	265 KB
<b>ZLinker.eww</b>	2021/3/27 14:39	IAR IDE Works...	1 KB

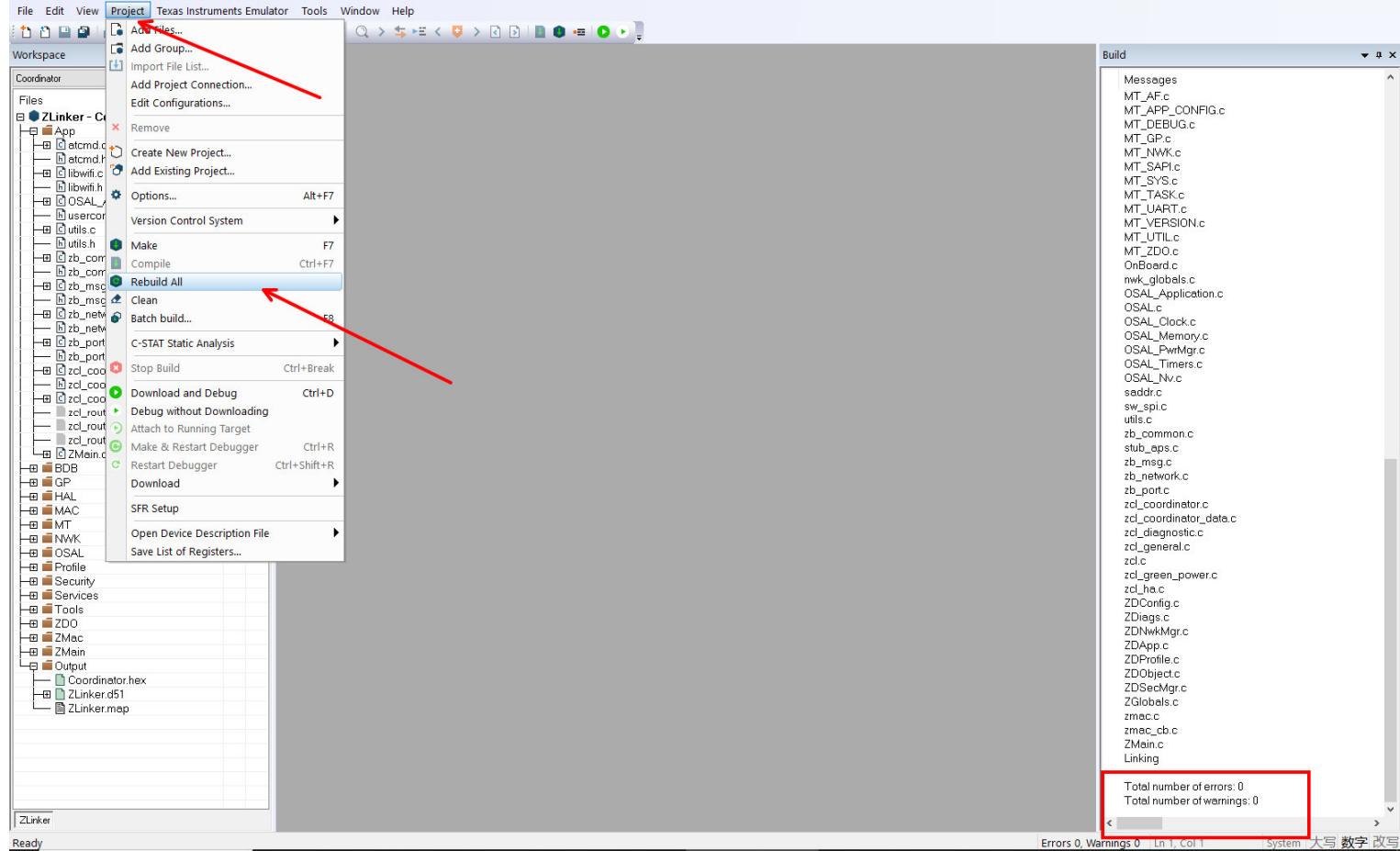
(3) Select Router, as shown in the figure.



Note: You need to select the Router role, and you cannot select EndDevice for the time being.

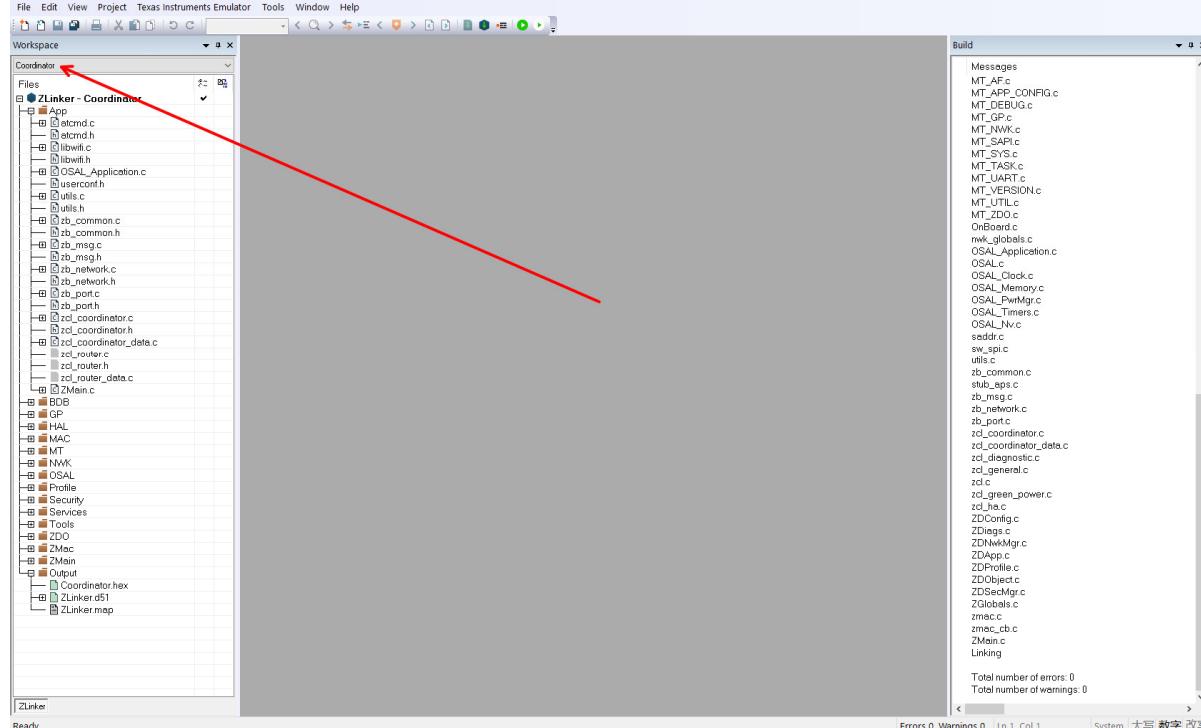
(4) The compilation and connection are successful, and the node firmware is generated, as shown in the figure.

ZLinker - IAR Embedded Workbench IDE - 8051 10.10.1

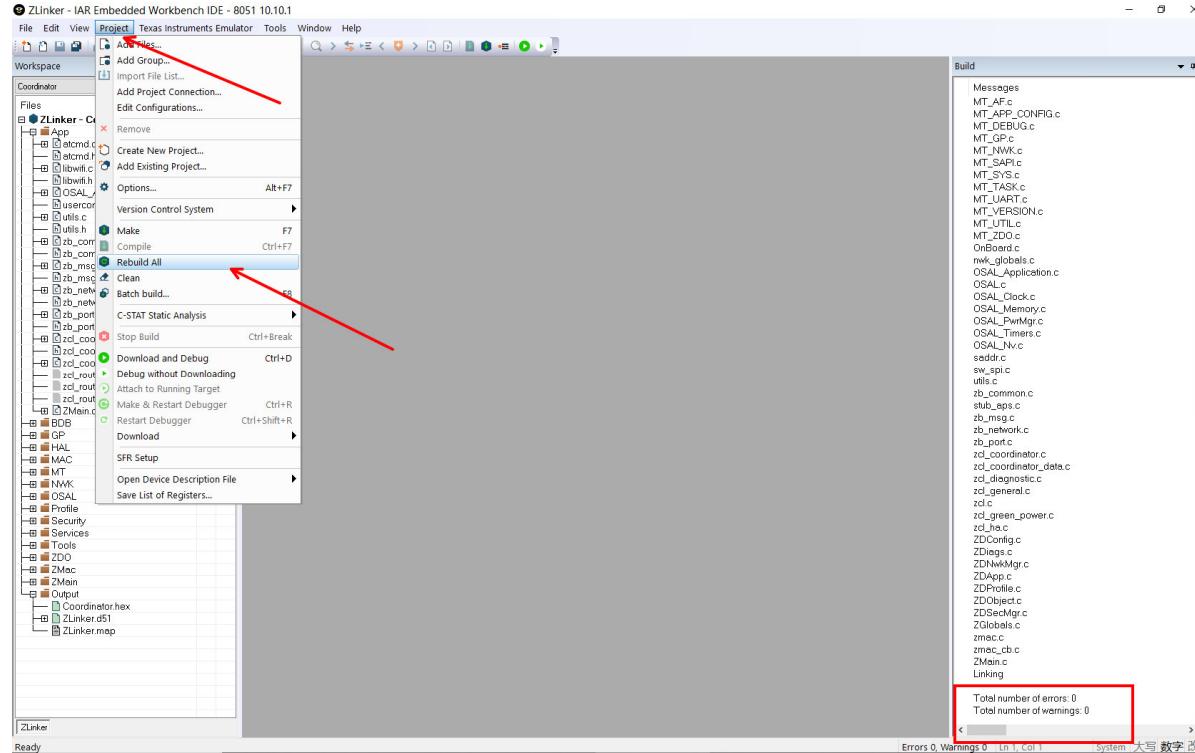


(5) Select Coordinator, as shown in the figure.

ZLinker - IAR Embedded Workbench IDE - 8051 10.10.1



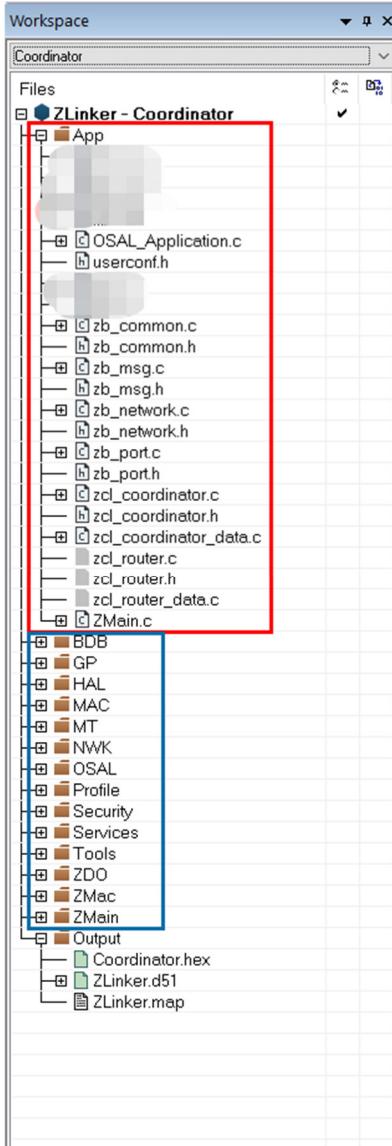
(4) The compilation and linking are successful, and the coordinator firmware is generated, as shown in the figure.



## 2. Source code file structure

**Note:** Since the first to fourth parts have explained each part of the code in detail, they will not be explained in detail here again. Please be sure to fully understand the first to fourth parts before reading this project code.

(1) The source code of this project is shown in the figure.



(2) The code in the blue box is mainly the code of Z-Stack 3.0 itself, provided by Texas Instruments. For detailed explanation of Z-Stack 3.0, please refer to "Part 3: Z-Stack 3.0 Detailed Explanation" and "Part 4: ZigBee 3.0 Network Programming" of this tutorial.

(3) The code in the red box is mainly written by us for this project. The brief introduction of each file is as follows:

- (A) OSAL\_Application.c: OSAL application layer code. For detailed description, please refer to "Part 3: Z-Stack 3.0 Detailed Explanation" → "Chapter 3: OSAL Detailed Explanation"
- (B) zb\_common.h/c: stores common definition content.
- (C) zb\_msg.h/c: Files related to Uart functions.
- (D) zb\_network.h/c: Files related to ZigBee networking.
- (E) zb\_port.h/c: stores files related to general logic processing.
- (F) zcl\_coordinator.h/c: Files related to coordinator functions.
- (G) zcl\_coordinator\_data.c: File related to coordinator data.
- (H) zcl\_router.h/c: Files related to routers.
- (I) zcl\_router\_data.c: Files related to router data.
- (J) ZMain.c: The file where the main() function is located.

(4) **zb\_port.h/c** is the file most relevant to the function of this project, and if the function of this project needs to be modified, it is generally started from **zb\_port.h/c**. Therefore, the following article will focus on **zb\_port.h/c**.

### 3. Detailed explanation of zb\_port.h/c files

(1) A zb\_port component is defined in zb\_port.h, and the content is as follows:

```
#ifndef __ZB_PORT_H__
#define __ZB_PORT_H__

#include "zb_common.h"

#ifndef __cplusplus
extern "C"
{

#endif

/*
 * zb_port 组件定义
 */
typedef struct {
    void (*init)(void);
    void (*on_click)(void);
    void (*on_priv_msg)(uint16_t nwkaddr, uint16_t cluster, const uint8_t *data, uint16_t len);
    void (*on_timer)(uint16_t ms);
} zb_port_t;

extern const zb_port_t *zbport_;


#endif /* __ZB_PORT_H__ */
```

(2) In the coordinator role, when the user presses the onboard User button, zb\_port\_on\_click() is automatically called. The code is as follows.

```
/*
 * 按钮事件处理函数
 */
void zb_port_on_click(void)
{
```

(3) The zb\_port\_on\_timer() function will be automatically executed every 1 second. You can add code that needs to be executed repeatedly.

```
/*
* 每隔1秒自动执行1次本函数
*/
void zb_port_on_timer(uint16_t ms)
{
    //省略
}
```

(4) When a ZigBee message is received, the zb\_port\_on\_priv\_msg() function will be called automatically. You can add message processing code in it.

```
/*
* 接收到ZigBee消息时，自动调用此函数
*
* @param nwkaddr 发送者的网络地址
* @param cluster Cluster
* @param data 数据内容
* @param len 数据长度
*/
void zb_port_on_priv_msg(uint16_t nwkaddr, uint16_t cluster, const uint8_t *data, uint16_t len)
{
    //省略
}
```

## 8.2. ZigBee 3.0 Environmental Information Collection (2)

### 1. Required Reading

This course is an **extracurricular course**, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise **please stop studying!**

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530
- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

### 2. Contents

- 8.2.1. System Introduction
- 8.2.2. Source code download
- 8.2.3. Hardware List
- 8.2.4. Using Tencent Cloud
- 8.2.5. Download and install MQTT.fx

- 8.2.6. MQTT.fx connects to Tencent Cloud
- 8.2.7. MQTT.fx publishes messages
- 8.2.8. Configuring the Development Board
- 8.2.9. Burn the coordinator firmware
- 8.2.10. Burning Router Firmware
- 8.2.11. Source code description

## 8.2.1. System Introduction

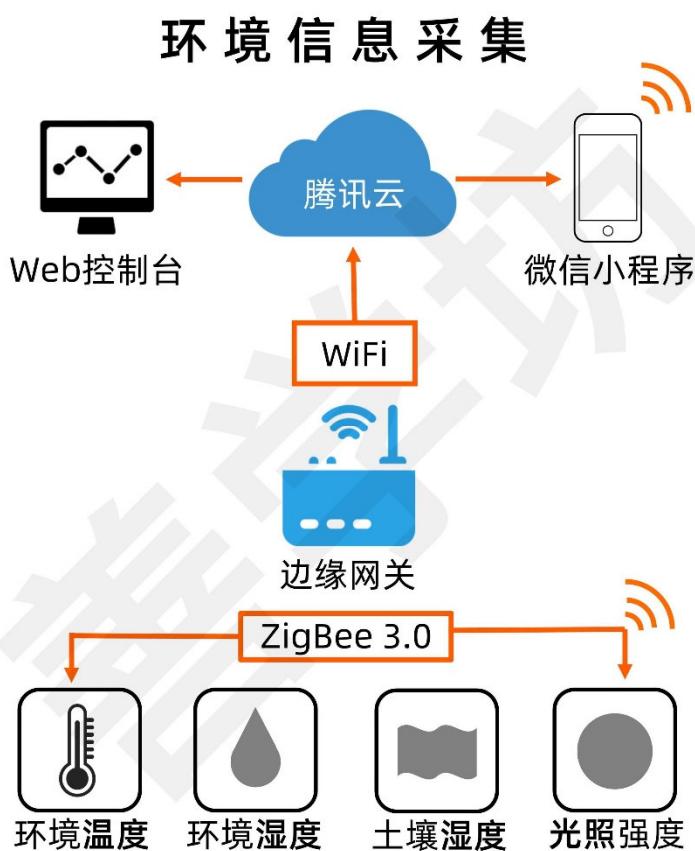
This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### System Introduction

(1) This system can collect ambient temperature, humidity, light intensity, and soil moisture, and upload them to **Tencent Cloud** and the Tencent Lianlian **WeChat applet**.

(2) The system architecture is shown in the figure.



(3) A screenshot of the Tencent LianLian WeChat applet is shown in the figure.



## 8.2.2. Source code download

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Source code download

- (1) Users who have purchased the product can download the additional materials from the "Supplementary Materials Download" section in the catalog.
- (2) This item is located in the additional materials as shown in the figure.

[↑ 上传](#)[新建文件夹](#) | [新建在线文档](#) | [离线下载](#)

返回上一级 | 全部文件 &gt; ... &gt; ZigBee &gt; 《ZigBee3.0开发指... &gt; 课例篇：项目实战

文件名	修改时间	大小
其他项目 1	2022-06-29 10:59	-
其他项目 2	2022-06-29 10:59	-
ZigBee 3.0 环境信息采集 (2) .zip	2023-03-31 12:53	56.1M
ZigBee 3.0 环境信息采集 (1) (多节点版) .zip	2023-03-30 16:28	281.5M
ZigBee 3.0 环境信息采集 (1) .zip	2023-03-30 16:28	64.6M
基于Zigbee的温湿度 & 光照度采集系统.zip	2022-06-29 10:59	33.6M
基于ZigBee的文件传送系统.zip	2022-06-29 10:59	27.1M
基于ZigBee的光照自动开关窗帘.zip	2022-06-29 10:59	38.8M
基于ZigBee的农业环境信息采集.zip	2022-06-29 10:59	69.8M
基于ZigBee的温湿度 & 信号强度探测系统.zip	2022-06-29 10:59	33.5M

## 8.2.3. Hardware List

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Hardware List

(1) ZigBee 3.0 standard board × 1: <https://item.taobao.com/item.htm?id=683089996879>

(2) ZigBee 3.0 Mini board × 1: <https://item.taobao.com/item.htm?id=683089996879>

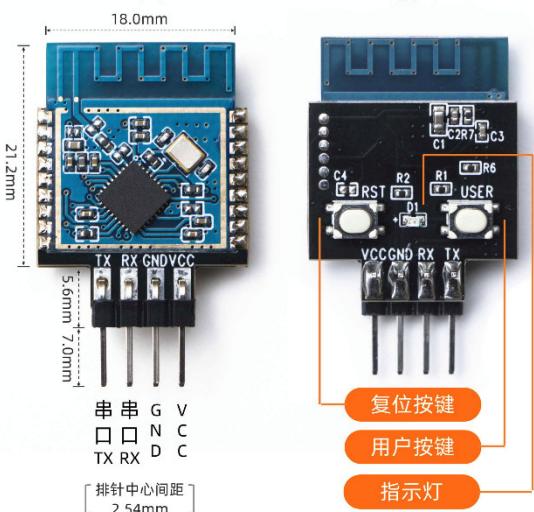
(3) 0.96 OLED12864 display × 2: <https://item.taobao.com/item.htm?id=683089996879>

(4) SmartRF04EB emulator × 1: <https://item.taobao.com/item.htm?id=683089996879>

(5) Micro USB data cable × 2: Free with the purchase of the development board.

(6) Dry cell battery box × 2 (optional): <https://item.taobao.com/item.htm?id=683115420758>

(7) 32C3-1 WiFi module × 1: <https://item.taobao.com/item.htm?id=695499826327>



(8) DuPont cable (female to female, 2.54 mm wide) × 1: <https://detail.tmall.com/item.htm?id=14466195609>  
(This DuPont cable is used to connect the 32C3-1 WiFi module to the development board)

(9) Soil moisture sensor & DuPont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

## Buying Guide

**Tip:** You can purchase all the hardware required for this project by purchasing according to the above method.

(1) ZigBee development kit version selection is shown in the figure.

zigbee开发板cc2530模块 IoT物联网无线控制套件单片机3.0m  
价格 ¥ 238.00 200+  
月销量  
淘宝价 ¥ 237.98 热卖促销  
优惠 淘金券可抵7.13元  
店铺优惠券 30元店铺优惠券, 满1500元可用 领取  
店铺优惠券 20元店铺优惠券, 满1000元可用 领取  
配送 广东广州至 广东广州番禺区 快递 免运费  
现货, 付款后24小时内发货  
网关类型 WiFi模块+USB串口 USB串口  
液晶屏幕 2个 1个 0个  
接收器 SmartRF04EB 接收器 不带 (无法接收)  
版本 ZigBee 标准版+Mini板 ZigBee 标准版 ZigBee Mini板  
【自由搭配】找客服  
收藏宝贝 (1384人气)

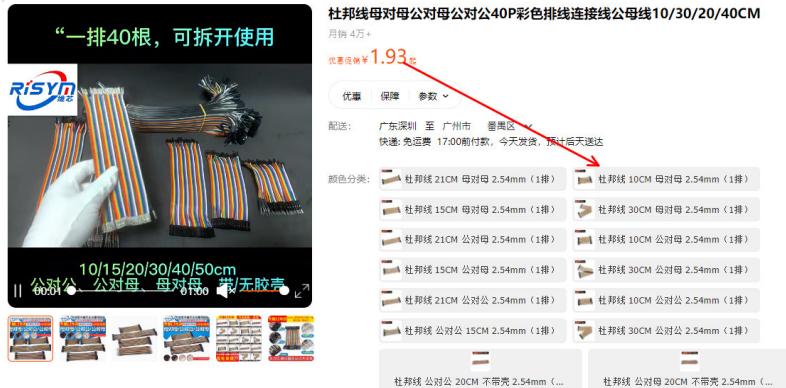
(2) Select the battery box as shown in the figure.

干电池盒  
• 可给各类开发板供电  
• 4.5v 输出  
USB 接口  
善学坊  
价格 ¥ 9.80  
优惠 店铺优惠券 30元店铺优惠券, 满1500元可用 领取  
店铺优惠券 20元店铺优惠券, 满1000元可用 领取  
配送 广东广州至 广东广州番禺区 快递 免运费  
现货, 付款后24小时内发货  
颜色分类  
数量 - 2 + 件  
立即购买 加入购物车  
承诺 7天无理由 运费险

(3) 32C3-1 WiFi module selection is shown in the figure.

型号: 32C3-1  
芯片: ESP32-C3FN4  
类型: 串口透传  
WiFi: 支持  
蓝牙5.0智能配网: 支持  
发射功率: 17dBm(典型)  
通信距离: 100m(无干扰)  
WiFi速率: 最高150Mbps  
供电: 3.0~3.6v; ≥500mA  
频率: 2.4GHz  
温度: -40~85°C  
SPI Flash: 4MB  
WIFI模块串口透传 Matter模组 esp32c3 无线物联网iot蓝牙ble射频ASR  
价格 ¥ 20.00 81  
月销量  
淘宝价 ¥ 11.58 热卖促销  
优惠 淘金券可抵0.34元  
店铺优惠券 30元店铺优惠券, 满1500元可用 领取  
店铺优惠券 20元店铺优惠券, 满1000元可用 领取  
配送 广东广州至 广东广州番禺区 快递 免运费  
现货, 付款后24小时内发货  
颜色分类 32C3-1 (推荐)  
数量 - 1 + 件  
立即购买 加入购物车  
承诺 7天无理由 运费险  
收藏宝贝 (132人气)

(4) The DuPont line version selection is shown in the figure.



(5) The soil moisture sensor selection is shown in the figure.



## 8.2.4. Using Tencent Cloud

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### 1. Register a Tencent Cloud account

(1) Log in to Tencent Cloud's official website <https://cloud.tencent.com/> and click the "Register" button, as shown in the figure.



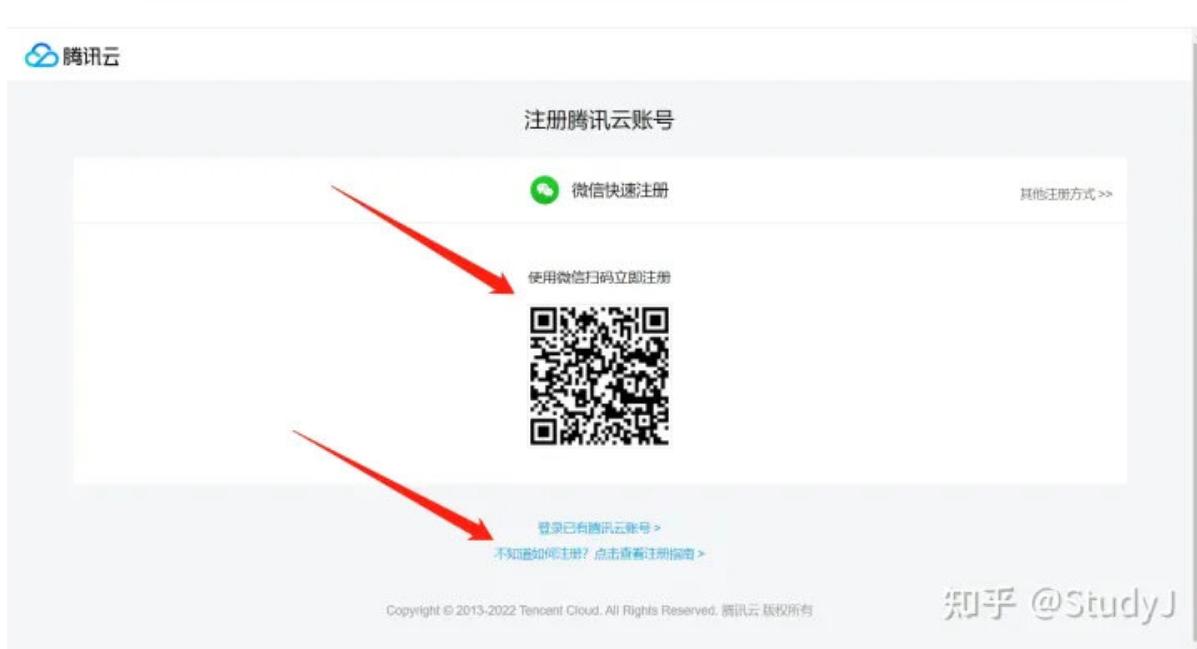
性能强大、安全、稳定的云产品

腾讯多年技术沉淀，300+ 款产品共筑腾讯云产品矩阵 [查看全部产品 >](#)

知乎 @StudyJ

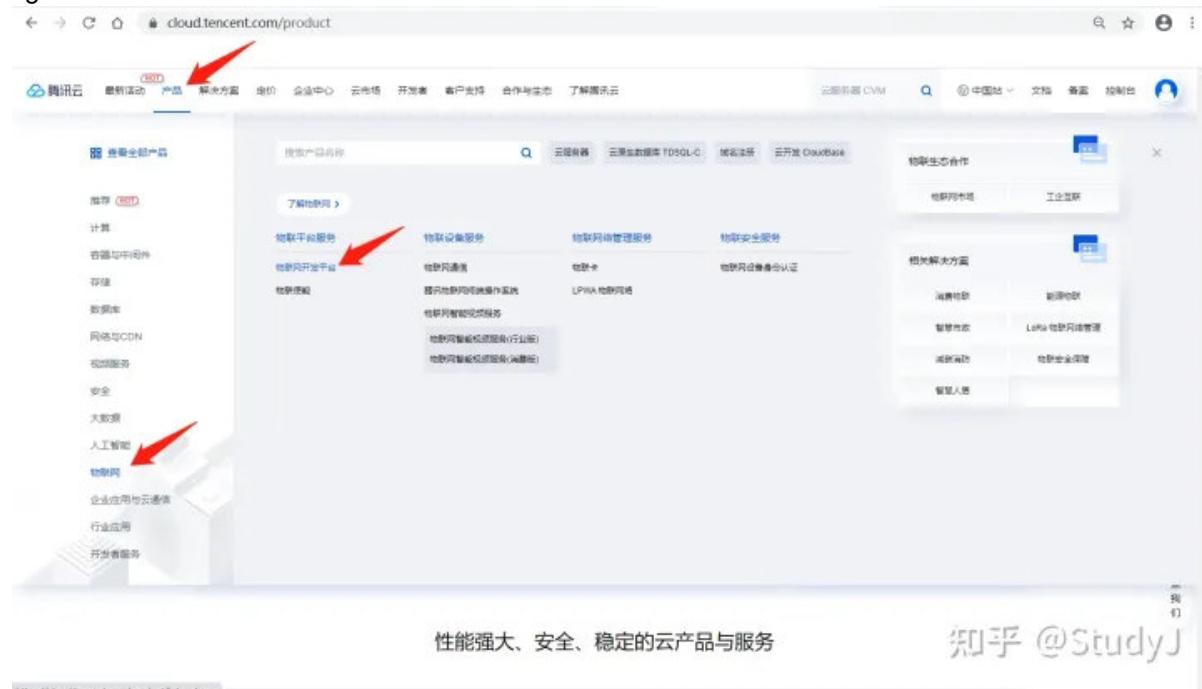
2024年6月6日

(2) Scan the QR code in the picture. If you have any questions during the registration process, you can also click the "Don't know how to register..." link at the bottom of the page.



## 2. New Project

(1) After completing the account registration as prompted, return to the official website homepage, then move the mouse over the "Products" and "Internet of Things" links in turn, and then click "Internet of Things Development Platform", as shown in the figure.



**Tip:** You can also directly click this link to enter the IoT development platform <https://cloud.tencent.com/product/iotexplorer>

(2) Click the “Management Console” link to enter the console, as shown in the figure.

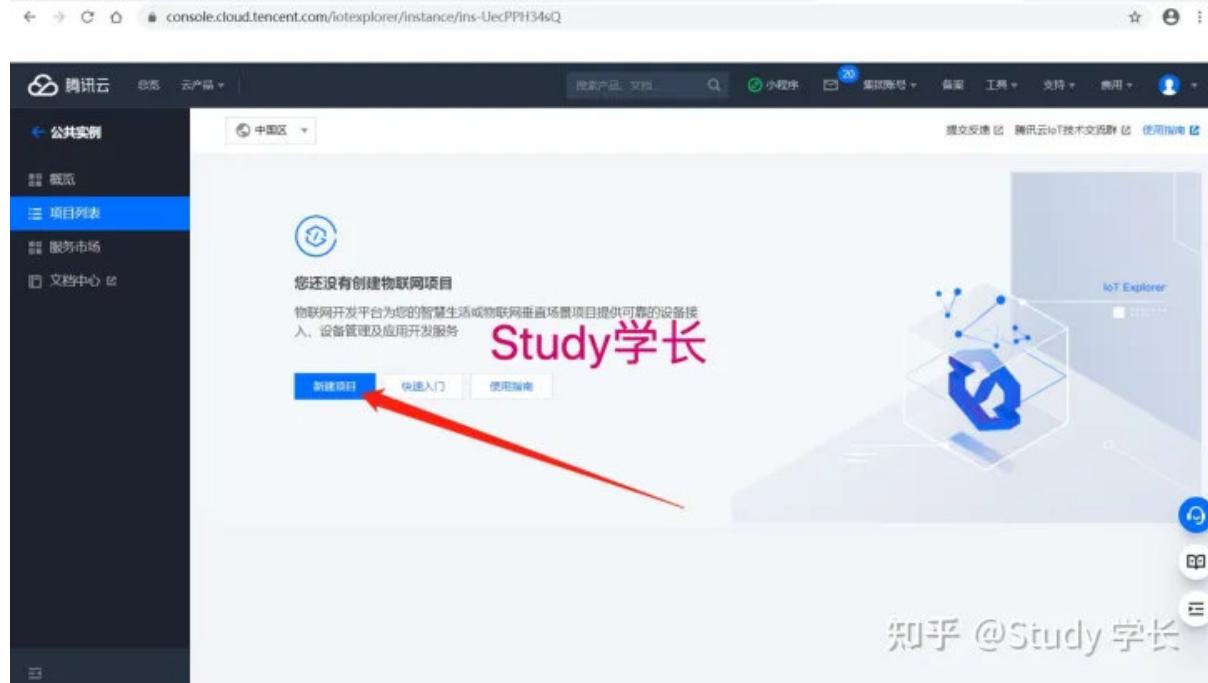
The screenshot shows the IoT Explorer product page. At the top, there's a navigation bar with links like '首页' (Home), '产品' (Products), '解决方案' (Solution), '定价' (Pricing), '企业中心' (Enterprise Center), '云市场' (Cloud Market), '开发者' (Developer), '客户支持' (Customer Support), '合作与生态' (Partnership and Ecosystem), and '了解腾讯云' (Learn about Tencent Cloud). Below the navigation is a large central image of a blue circular device with a hexagonal logo in the center. To the left of the image are three buttons: '立即体验' (Try Now), '管理控制台' (Management Console), and '产品文档' (Product Documentation). A red arrow points to the '管理控制台' button. Below these buttons is a section titled '产品特性' (Product Features) with several items listed, each with an icon and a brief description. On the right side of the page, there's a vertical sidebar with icons for '帮助中心' (Help Center), '联系客服' (Contact Customer Service), and '我的订单' (My Orders). At the bottom right, there's a watermark for '知乎 @StudyJ'.

Tip: You can also directly click this link to enter the console <https://console.cloud.tencent.com/iotexplorer>

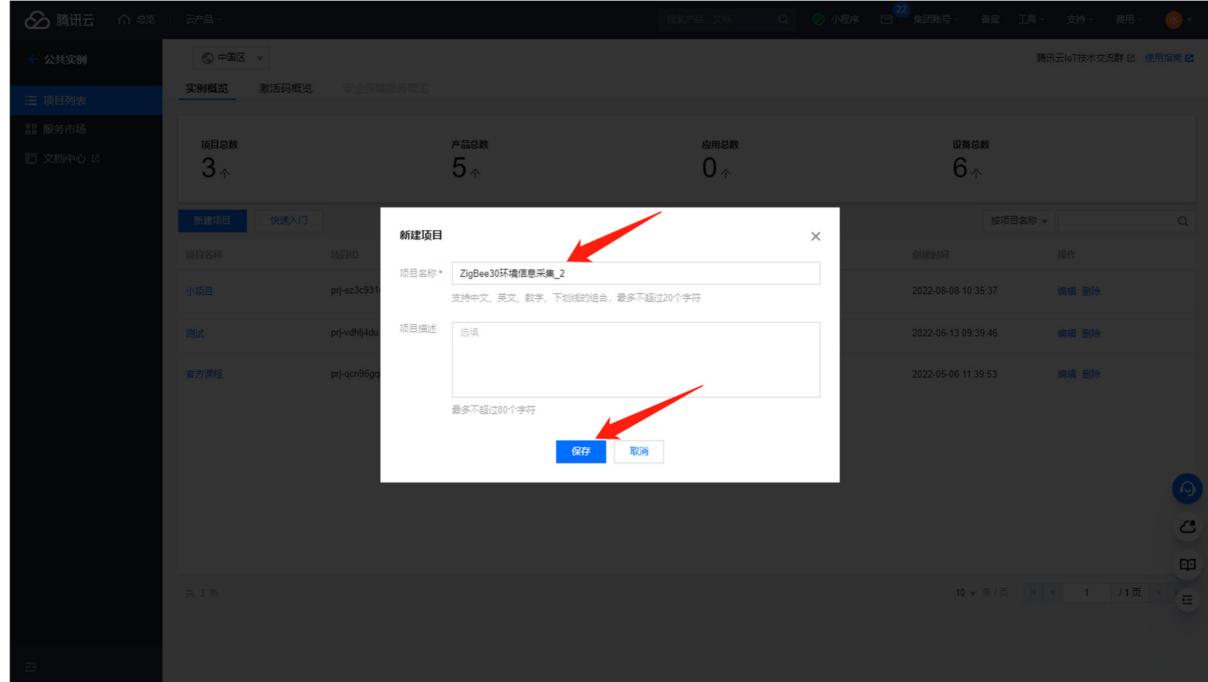
(3) Click the “Use Now” button in the public instance, as shown in the figure.

The screenshot shows the '实例管理' (Instance Management) page under the '物联网开发平台' (IoT Development Platform). The left sidebar has sections for '开通中心', 'EI 实例管理' (selected), '相关产品', 'IoT Hub', and '身份认证 TID'. The main area shows '资源概况' (Resource Overview) with a total of 1 instance and 0 enterprise instances. Below this is a '公共实例' (Public Instance) card for instance ID 'ins-UecPPH34sQ', showing 0 project datasets, 0 products, 0 devices, and a creation time of '2022-05-08 11:32:05'. A red arrow points to the '立即使用' (Use Now) button in this card. To the right is a '常用功能' (Common Functions) panel with buttons for '购买企业实例' (Buy Enterprise Instance), '下载设备SDK' (Download Device SDK), '使用指南' (User Guide), and '设备迁移' (Device Migration). There's also a '产品动态' (Product Dynamics) section with news items like '新增人脸识别增值服务' (New added face recognition value-added service).

(4) Click the “New Project” link, as shown in the figure.



(5) Enter the project name in the pop-up prompt box, and then click the "Save" link, as shown in the figure.



### 3. Create a new product

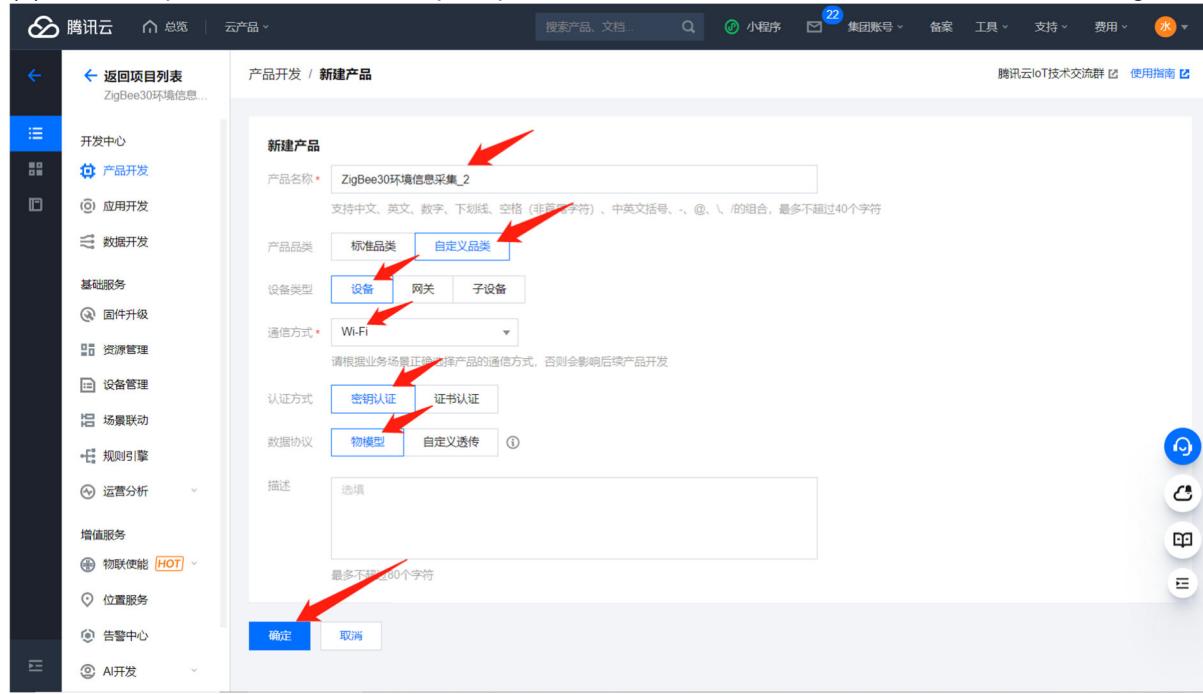
(1) After saving the project, you will automatically enter the project list page. Click the project you just created on this page, as shown in the figure.

The screenshot shows the Tencent Cloud Project List interface. On the left sidebar, under '项目列表', there is a '公共实例' section. The main area displays project statistics: 4个 项目总数, 5个 产品总数, 0个 应用总数, and 6个 设备总数. Below this is a table listing projects. The first row, 'ZigBee30环境信息采集\_2', is highlighted with a red box. This row contains columns for '项目名称' (ZigBee30环境信息采集\_2), '项目ID' (prj-vv22c7my), '产品/设备数量' (0 / 0), '应用数量' (0), '物联网能力' (未开通), and '创建时间' (2023-03-30 19:13:34). There are '编辑' and '删除' buttons in the '操作' column. To the right of the table is a vertical toolbar with icons for refresh, cloud, data, and more. At the bottom, there is a pagination bar showing '共 4 条' and a page number '1 / 1 页'.

(2) Click the "New Product" link on the product development page, as shown in the figure.

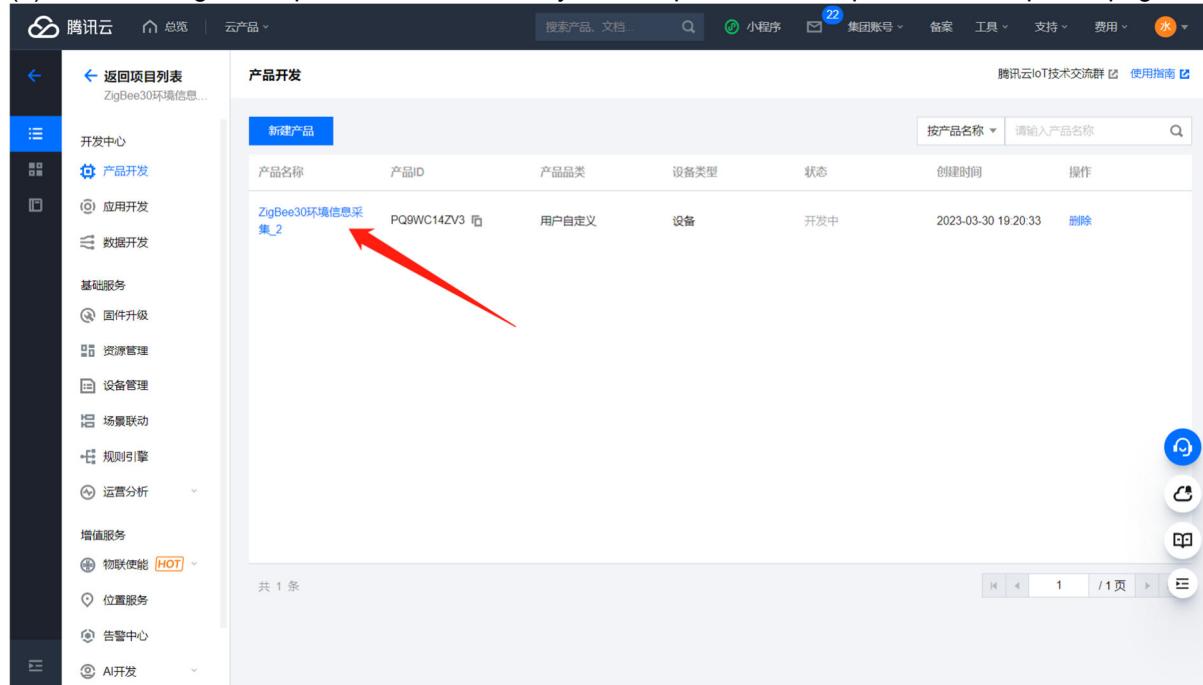
The screenshot shows the Tencent Cloud Product Development page. On the left sidebar, under '开发中心', there is a '产品开发' section. A red arrow points to the '新建产品' button, which is highlighted with a blue box. The main area has a header '产品开发' and a search bar with '按产品名称' and '请输入产品名称'. Below the header is a table with columns: '产品名称', '产品ID', '产品品类', '设备类型', '状态', '创建时间', and '操作'. A message at the bottom of the table says '当前项目下产品列表为空, 您可以点击新建产品'. To the right of the table is a vertical toolbar with icons for refresh, cloud, data, and more. At the bottom, there is a pagination bar showing '共 0 条' and a page number '1 / 1 页'.

(3) Enter the product information as prompted and then click the "OK" button, as shown in the figure.



## 4. Physical Model Configuration

(1) After creating a new product, click the newly created product on the product development page, as shown in the figure.



(2) Find and click the "New Custom Function" link in the object model, as shown in the figure.

The screenshot shows the 'Object Model' step in the 'ZigBee30环境信息采集\_2' project. On the left sidebar, '产品开发' is selected. The main area displays two sections: 'Standard Function' and 'Custom Function'. A red box highlights the '1 物模型' step. A red arrow points from the '新建自定义功能' button in the 'Custom Function' section to the right panel, which shows product details like Product ID: PQ9WC14ZV3 and Product Category: User-defined Device.

(3) Enter the temperature function as prompted, and then click the "Save" button, as shown in the figure. Note that the settings must be exactly as shown in the figure, especially the identifier must be set to temp.

The screenshot shows the 'Add New Custom Function' dialog. The 'Properties' tab is selected. The configuration includes:

- Function Type: 事件 (Event)
- Function Name: 环境温度 (Environmental Temperature)
- Identifier: temp
- Data Type: 整数型 (Integer)
- Read Type: 只读 (Read-only)
- Value Range: -50 to 100
- Initial Value: 0
- Step: 1
- Unit: °C

A red arrow points from the 'temp' identifier field to the right panel, which shows the same product details as the previous screenshot. Another red arrow points from the '保存' (Save) button at the bottom right of the dialog to the right panel.

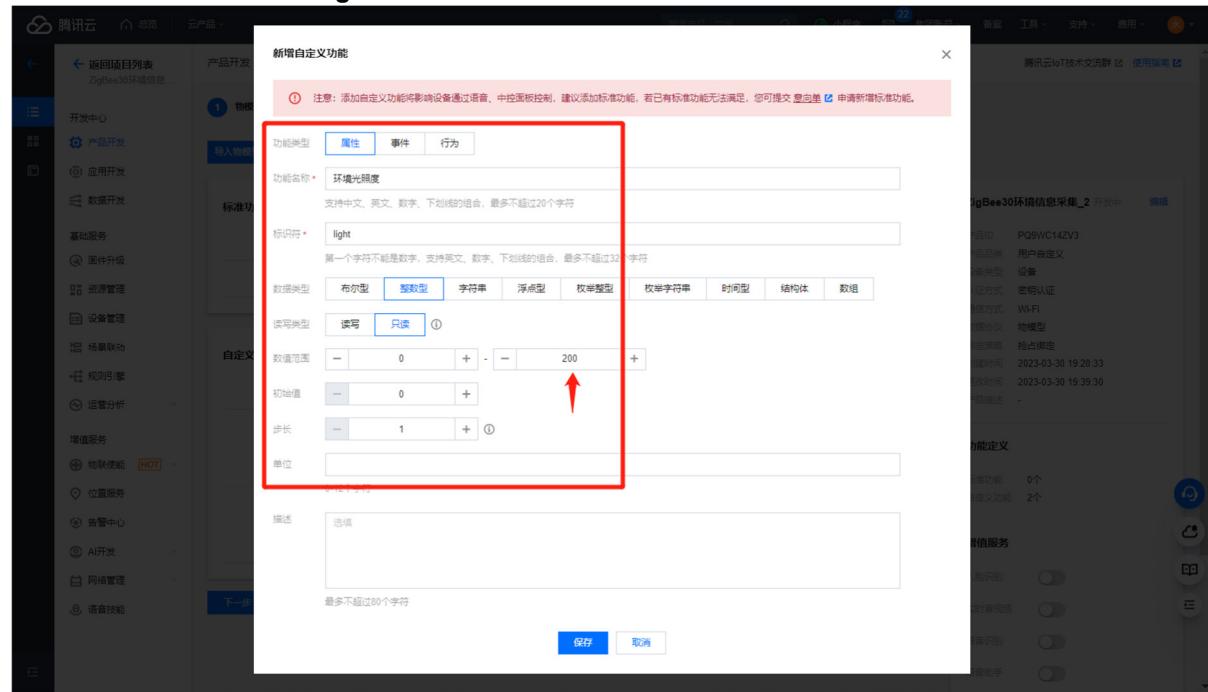
(4) The temperature acquisition function is created, as shown in the figure.

The screenshot shows the Tencent Cloud IoT platform interface for creating a ZigBee30 environment information collection project. On the left sidebar, under '物模型' (Thing Model), there is a red box highlighting the '自定义功能' (Custom Function) section. A new custom function named '环境温度' (Environment Temperature) has been created, with its details visible in the main panel: type '属性' (Attribute), name 'temp', identifier 'temp', data type '整数型' (Integer), read/write type '只读' (Read-only), range '-50-100', initial value '0', step '1', and unit '°C'. The right side of the screen displays the product's basic information, including ID 'PQ9WC14ZV3', name 'ZigBee30环境信息采集\_2', and creation time '2023-03-30 19:20:33'.

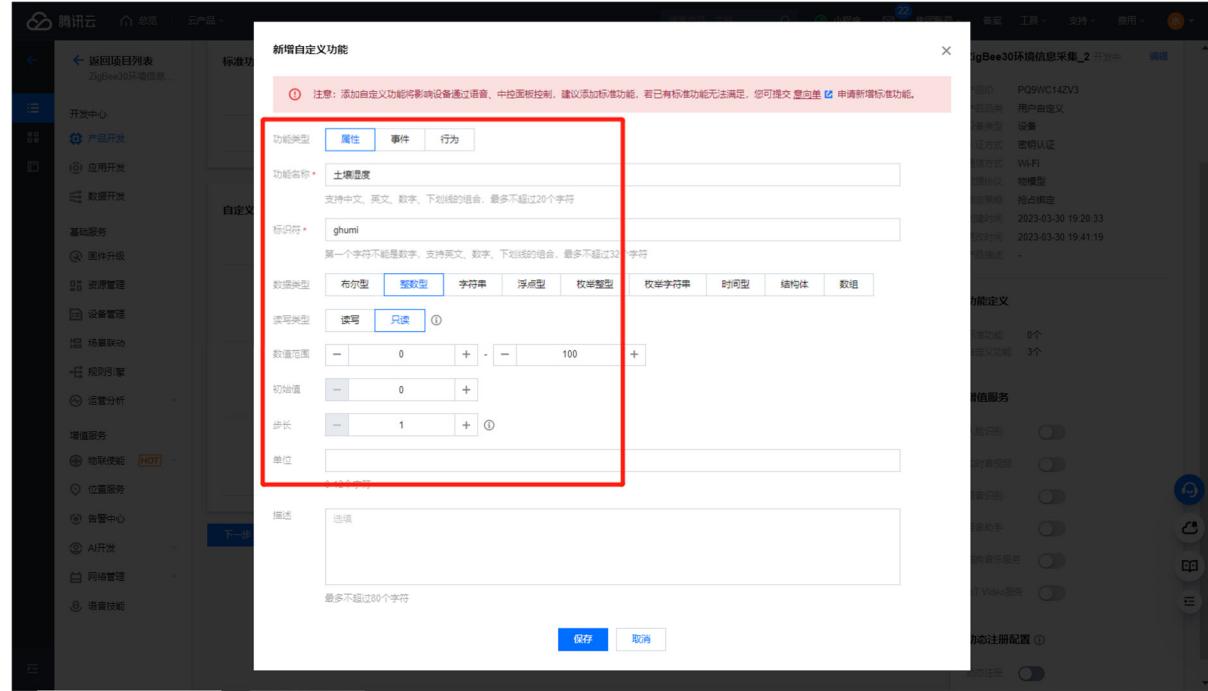
(5) Create a humidity collection function in the same way, with the function parameters shown in the figure. Note that the identifier must be set to humi.

This screenshot shows the continuation of the project setup. The '新增自定义功能' (Add Custom Function) dialog is open, and a new custom function named '环境湿度' (Environment Humidity) is being created. The configuration parameters are identical to the temperature function: type '属性' (Attribute), name 'humi', identifier 'humi', data type '整数型' (Integer), read/write type '只读' (Read-only), range '-100-100', initial value '0', step '1', and unit '%'. The right side of the screen shows the product's basic information, including ID 'PQ9WC14ZV3', name 'ZigBee30环境信息采集\_2', and creation time '2023-03-30 19:20:33'.

(6) Create a light acquisition function in the same way, with the function parameters as shown in the figure. Note that the identifier must be set to light.



(7) Create a soil moisture collection function in the same way, with the function parameters shown in the figure. Note that the identifier must be set to ghum1.



## 5. Equipment development and configuration

(1) After creation is complete, click "Next", as shown in the figure.

This screenshot shows the 'Product Creation' page in the Tencent Cloud IoT console. On the left sidebar, '物联使能' (IoT Enablement) is selected. In the main area, there's a table for '自定义功能' (Custom Functions) with three entries: '环境温度' (Environment Temperature), '环境湿度' (Environment Humidity), and '土壤湿度' (Soil Humidity). Each entry includes fields for '属性' (Attribute), '功能名称' (Function Name), '标识符' (Identifier), '数据类型' (Data Type), '读写类型' (Read/Write Type), '数据定义' (Data Definition), and '操作' (Operations). To the right of the table, detailed product information is displayed, including Product ID (PQWC14ZV3), Product Category (User-defined), Device Type (Device), Authentication Method (Certificate Authentication), Communication Method (Wi-Fi), Data Model (Object Model), Schedule Policy (Occupancy Policy), Creation Time (2023-03-30 19:20:33), Last Update (2023-03-30 19:42:25), and Product Description. A large red arrow points to the '下一步' (Next) button at the bottom left of the main content area.

(2) The system will enter the device development page, just click the "Next" link, as shown in the figure.

This screenshot shows the 'Device Development' configuration page for a '温湿度传感器 NB-IoT' (Temperature and Humidity Sensor NB-IoT) device. The left sidebar shows '设备开发' (Device Development) is selected. The main area has tabs for '设备开发' (Device Development) and 'Topic列表' (Topic List). Below these tabs, there are three sections: '请选择设备开发方式' (Select Device Development Method), '基于板载开发' (On-board Development), '基于SDK开发' (Based on SDK), and '基于OS开发' (Based on OS). Each section contains descriptive text and links to documentation. A large red arrow points to the '下一步' (Next) button at the bottom left of the main content area. On the right side, detailed device information is shown, including Product ID (UVGW0091Z), Product Category (智慧城市-环境监测-温湿度检测), Device Type (设备), Connection Method (Certificate Authentication), Communication Method (2G/3G/4G), Data Model (Object Model), Schedule Policy (2022-05-06 11:52:51), Last Update (2022-05-06 12:09:18), and Product Description (采集环境温湿度并通过NB-IoT将数据上传到云平台). A watermark '知乎 @Study 学长' is visible on the right.

## 6. Interactive Development Configuration

(1) On the interactive development page, open "Connect to Tencent Lianlian Official Mini Program" and "Use the Universal Version APP to Control Products", and then click the Product Display Configuration button, as shown in the figure.

The screenshot shows the Tencent Cloud Interactive Development Platform interface. On the left sidebar, under '产品开发' (Product Development), several options like '设备中心' (Device Center), '应用开发' (Application Development), and '数据开发' (Data Development) are listed. In the main content area, there's a section titled '接入腾讯连连官方小程序' (Connect to Tencent Lianlian Official Mini Program) with a toggle switch. Below it is another section titled '使用通用版APP控制产品' (Use Universal Version APP to Control Product) with another toggle switch. A large red arrow points from the bottom right towards the '配置' (Configure) button located next to the second switch. The overall background is white with blue and black text.

(2) Enter the product information as prompted and click Save, as shown in the figure.

The screenshot shows the Tencent Cloud Product Management Platform. On the left sidebar, under '产品开发' (Product Development), various sub-options like '应用开发' (Application Development) and '数据开发' (Data Development) are visible. The main area is titled '产品展示配置' (Product Display Configuration). It includes fields for '产品展示图标' (Product Display Icon), '产品展示名称' (Product Display Name) with a placeholder 'ZigBee30环境信息采集\_2', '厂家名称' (Manufacturer Name) with a placeholder 'Study J', '产品型号' (Product Model) with a placeholder 'sensor-v2', and a 'UI强制显示在线' (Force UI to show online) toggle switch. At the bottom left, a prominent blue '保存' (Save) button is highlighted with a red arrow. The overall interface is clean with a white background and blue accents for buttons and links.

(3) Click the Return button, as shown in the figure.

The screenshot shows the Tencent Cloud IoT product configuration interface. On the left, there's a sidebar with various development and management options. The main area is titled 'Product Display Configuration'. It includes fields for 'Product Display Icon' (with a preview window showing a device icon and weather information), 'Product Display Name' (set to 'ZigBee30环境信息采集\_2'), 'Manufacturer Name' (set to 'Study J'), 'Product Model' (set to 'sensor-v2'), and a 'UI Force Online' toggle switch. At the bottom right of the main area are several circular icons for navigation and settings. A large red arrow points to the back arrow icon located at the top left of the main content area.

(4) Click the Smart Linkage Configuration button, as shown in the figure.

The screenshot shows the Tencent Cloud IoT product configuration interface, specifically the 'Smart Linkage Configuration' section. The sidebar on the left remains the same. The main area has several configuration sections: 'Quick Access Configuration', 'Panel Configuration', 'Device Binding Guidance', 'Scan Product Introduction', and 'Smart Interaction Configuration'. Each section has a brief description and a blue 'Configuration' button with a help icon. A large red arrow points to the 'Configuration' button for the 'Smart Interaction Configuration' section. The text '知乎 @Study 学长' is overlaid on the bottom right of the page.

(5) Select the ambient temperature and humidity as the conditions for smart linkage, and then click the "Save" link, as shown in the figure.

The screenshot shows the 'Smart Linkage Configuration' page. On the left sidebar, 'Product Development' is selected. In the main area, there's a table with columns for 'Function' and 'As Condition'. Under 'Function', 'Environment Temperature' and 'Environment Humidity' are checked. Under 'As Condition', both have checkboxes checked. A red box highlights this section. At the bottom left is a blue 'Save' button, which has a red arrow pointing to it.

(6) Click the Next button, as shown in the figure.

The screenshot shows the 'Next Step' configuration page. It includes sections for '接入腾讯连连官方小程序' (Connect to Tencent LianLian Official Mini Program), '使用通用版APP控制产品' (Control products via General APP), and various configuration options like '产品展示配置' (Product Display Configuration), '快捷入口配置' (Quick Entry Configuration), etc. A large red arrow points to the '下一步' (Next Step) button at the bottom left.

## 7. Device debugging configuration

(1) Click the "New Device" button in the Device Debugging column, as shown in the figure.

The screenshot shows the Tencent Cloud IoT console interface. The top navigation bar includes '搜索产品、文档...' (Search products, documents...), '小程序' (Microservices), '集团账号' (Corporate account), '备案' (Filing), '工具' (Tools), '支持' (Support), and '费用' (Cost). The main title is '产品开发 / 温湿度传感器 NB-IoT'. The breadcrumb trail shows: 物模型 > 设备开发 > 交互开发 > 4. 设备调试. The right panel displays product details for '温湿度传感器 NB-IoT' (Product ID: UVGWDD0Y91Z, Product Type: 智能城市-环境监测-温湿度检测, Device Type: 设备, Authentication Method: 密钥认证, Communication Method: 2G/3G/4G, Data Protocol: 物模型, Creation Time: 2022-05-06 11:52:51, Last Update: 2022-05-06 12:09:18, Product Description: 采集环境温湿度并用通过NB-IoT将数据上报到云平台). The bottom right corner has a watermark: 知乎 @Study 学长.

(2) Enter the device name as prompted, and then click the "Save" button, as shown in the figure.

The screenshot shows the Tencent Cloud IoT console interface for a different project ('ZigBee30环境信息采集\_2'). The breadcrumb trail shows: 物模型 > 设备开发 > 交互开发 > 4. 设备调试 > 5. 批量投产. The right panel displays product details for 'ZigBee30环境信息采集\_2' (Product ID: PQ9WC14ZV3, Product Type: 用户自定义, Device Type: 设备, Authentication Method: 密钥认证, Communication Method: Wi-Fi, Data Protocol: 物模型, Location: 杭州地区, Creation Time: 2023-03-30 19:20:33, Last Update: 2023-03-30 20:26:59, Product Description: -). The bottom right corner has a watermark: 知乎 @Study 学长.

(3) In the device you just created, click the “QR Code” link, as shown in the figure.

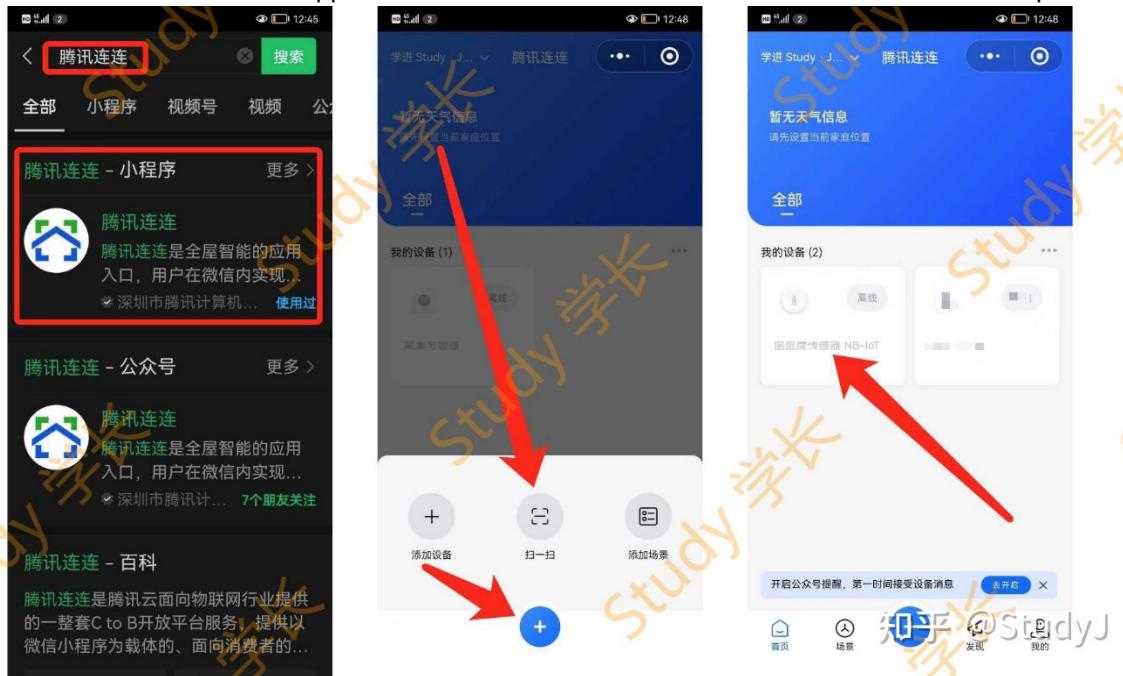
The screenshot shows the 'Device Commissioning' step of a product development process. The main interface displays a table of devices, with one entry for a 'Sensor' device. To the right of the device table, there is a detailed product configuration panel. At the bottom right of this panel, there is a 'Scan QR code' button, which is highlighted by a large red arrow. The configuration panel also contains sections for 'Product Definition' and 'Value-added Services'.

(4) A QR code will pop up on the web page, as shown in the figure.

The screenshot shows a QR code overlay on a web page. The text inside the QR code overlay reads: '消费者设备可使用腾讯连接扫描绑定加快调试效率' (Consumer devices can use Tencent Connection Scan to bind and accelerate commissioning efficiency). The background shows the same IoT console interface as the previous screenshot, with the 'Device Commissioning' step selected.

知乎 @StudyJ

(5) Use the "Tencent Lianlian" applet to scan the QR code and add the device. Follow the steps as shown in the figure.



## 8. Get the MQTT service address and port

(1) Click the device name you just created, as shown in the figure.

The screenshot shows the Tencent Cloud IoT console's device management interface. On the left, there's a sidebar with various service categories like Product Development, Data Development, and Rule Triggers. The main area shows a list of devices under a project named 'ZigBee30环境信息采集\_2'. One device, 'Sensor', is highlighted with a red arrow. To the right, a detailed view of the 'Sensor' device is shown, including its product ID (PQ9WC14ZV3), product category (User-defined), and connection method (MQTT). The right side also includes sections for Function Definition and Value-added Services.

(2) Copy the product ID as shown in the figure.

The screenshot shows the Tencent Cloud IoT console interface. On the left sidebar, under the '产品开发' section, there are various service icons like Application Development, Data Development, and Device Management. The main content area displays a device named 'TH\_Sensor\_Test'. Under the '设备信息' tab, the '设备名称' is 'TH\_Sensor\_Test', '所属产品' is '温湿度传感器 NB-IoT', and the '产品ID' is 'UVGW...'. The '产品ID' field is highlighted with a red box. Other tabs include Online Debugging, Cloud Diagnosis Log, Device Cloud Log, Device Local Log, and Extension Information. On the right side, there are edit and delete buttons, and a watermark for 'Study 学长' and '知乎 @Study 学长'.

(3) Add ".iotcloud.tencentdevices.com" after the product ID to get the MQTT server address. For example, the author's server address is:

UVGWDOY8XX.iotcloud.tencentdevices.com

(4) In addition, according to the official Tencent Cloud documentation, the default port for MQTT service key authentication is 1883. (Reference document: <https://cloud.tencent.com/document/product/634/32546#.E5.92.8C.E6.A0.87.E5.87.86-mqtt-.E5.8C.BA.E5.88.AB>)

(5) So far, the two parameters obtained in this step are as follows:

MQTT服务器地址 : UVGWDOY8XX.iotcloud.tencentdevices.com

秘钥认证端口号 : 1883

## 9. Generate MQTT link parameters

(1) Copy the product ID, device name, and device key respectively, as shown in the figure.

The screenshot shows the Tencent Cloud IoT console interface. On the left sidebar, under the '产品开发' section, there are various service icons like Application Development, Data Development, and Device Management. The main content area displays a device named 'Sensor'. Under the '设备信息' tab, the '设备名称' is 'Sensor' and the '产品ID' is '4ZV3...'. Both of these fields are highlighted with red boxes. Other tabs include Online Debugging, Cloud Diagnosis Log, Device Cloud Log, Device Local Log, and Extension Information. On the right side, there are edit and delete buttons, and a watermark for 'Study 学长' and '知乎 @Study 学长'.

(2) Find the "MQTT Connection Parameter Generation Tool" in the development toolkit that comes with this course, as shown in the figure.

返回上一级 | 全部文件 > ... > 课程配套学习资料 (... > ZigBee > 《ZigBee3.0开发指...

<input type="checkbox"/> 文件名	修改时间	大小
电路原理图	2023-03-02 22:27	-
MQTT连接参数生成工具.zip	2023-03-30 20:38	18KB
友善串口调试助手.zip	2022-06-29 22:13	20.3M
SmartRF04EB Driver.zip	2022-06-29 11:57	2M
TI Z-Stack 3.0.2 (CSDN视频课程专用).exe	2021-10-03 12:50	22.6M
PuTTY.zip	2021-09-05 21:55	2.4M
Ubiqua Protocol Analyzer.rar	2021-08-18 15:14	3.9M
CH34x USB转串口驱动.zip	2021-08-17 23:55	229KB
TI Z-Stack3.0.1.zip	2021-08-16 16:38	23.3M
IAR EW for 8051 10.10.1.zip	2021-08-16 16:36	327.8M
善学坊官方学习指南.html	2021-08-16 16:34	212B

(3) After decompressing the compressed package, use a browser to open the URL inside, as shown in the figure.

此电脑 > 文件 (H:) > 学习资料 > 《STM32+无线通信开发指南》配套软件工具 > MQTT连接参数生成工具

名称	修改日期	类型	大小
crypto-js.min.js	2020/7/13 17:53	JavaScript 文件	47 KB
善学坊MQTT连接参数生成工具.html	2022/4/6 14:33	Chromium HTM...	5 KB
善学坊官方学习指南.html	2022/4/8 21:10	Chromium HTM...	1 KB

(4) After opening, fill in the information copied previously according to the prompts, then select the signature algorithm and click Generate to generate UserName and Password, as shown in the figure.

请输入设备信息:

Study 学长

ProductID: UVGWDOY (产品ID)

DeviceName: TH...t (设备名称)

DeviceSecret: MuxH4msaUyQgw7dyF5 (设备秘钥)

Hmac签名算法: HMAC-SHA256 (Hmac签名算法)

结果:

UserName: UVGWDOY\_Sensor\_Test\_12345\_165242

Password: 3a1fe29981188fd63726250cd6bdbf9db7fbce41e1dd32b97184fe0\_Hmacsha256

知乎 @Study 学长

(5) So far, this step has obtained the following information:

ProductID (产品ID) : UVGWDOY8XX

UserName : UVGWDOY9IXXX\_Sensor\_Test;12010888;3YY2L;1652428888

Password : 3a1fe29981188fd63726250cd6bdbf9db7fbce41e1dd32b97184fe0

## 10. Get the theme

(1) Click Device Development and Topic List in turn, and copy the topics for attribute publishing and subscription, as shown in the figure.

The screenshot shows the Tencent Cloud IoT console interface. On the left sidebar, under '产品开发', there are several categories: '物模型', '设备开发' (selected), '交互开发', '设备调试', and '批量投产'. In the main content area, the path '产品开发 / ZigBee30环境信息采集\_2' is displayed. Below this, there are tabs for 'Topic列表' (selected), '物模型Topic', '系统级Topic', and '自定义Topic'. A table lists various topics with columns for 'Topic权限' (Topic Permission), '操作权限' (Operation Permission), and '备注' (Remarks). Two specific topics are highlighted with a red box: '\$thing/up/property/PQ9W \${deviceName}' and '\$thing/down/property/PQ9V \${deviceName}'. To the right, a detailed product information panel is visible for 'ZigBee30环境信息采集\_2', including fields like Product ID, Product Category, Device Type, Authentication Method, Communication Method, and Creation Time.

(2) Replace "\${deviceName}" with the device name obtained in the previous step. The replaced topic is as follows:

属性发布主题 : \$thing/up/property/PQ9WC1XXX/Sensor

属性订阅主题 : \$thing/down/property/PQ9WC1XXX/Senor

## 8.2.5. Download and install MQTT.fx

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### Introduction to MQTT.fx

MQTT.fx is a PC MQTT client that can connect to the device just created in Tencent Cloud.

### MQTT.fx Download

(1) Download address: See "Supporting Resource Download" in the catalog of this tutorial.

(2) After downloading the supporting resources, the location of the installation package is shown in the figure.

[返回上一级](#) | [全部文件](#) > ... > [课程配套学习资料](#) (... > [ZigBee](#) > [《ZigBee3.0开发指...](#)

<input type="checkbox"/>	文件名	修改时间	大小
	电路原理图	2023-03-02 22:27	-
	MQTT.fx v1.7.1 for Windows 64位系统.exe	2023-03-30 20:45	51M
	MQTT连接参数生成工具.zip	2023-03-30 20:38	18KB
	友善串口调试助手.zip	2022-06-29 22:13	20.3M
	SmartRF04EB Driver.zip	2022-06-29 11:57	2M
	TI Z-Stack 3.0.2 (CSDN视频课程专用).exe	2021-10-03 12:50	22.6M
	PuTTY.zip	2021-09-05 21:55	2.4M
	Ubiqua Protocol Analyzer.rar	2021-08-18 15:14	3.9M
	CH34x USB转串口驱动.zip	2021-08-17 23:55	229KB
	TI Z-Stack3.0.1.zip	2021-08-16 16:38	23.3M

(3) Simply install the software after downloading it.

## 8.2.6. MQTT.fx connects to Tencent Cloud

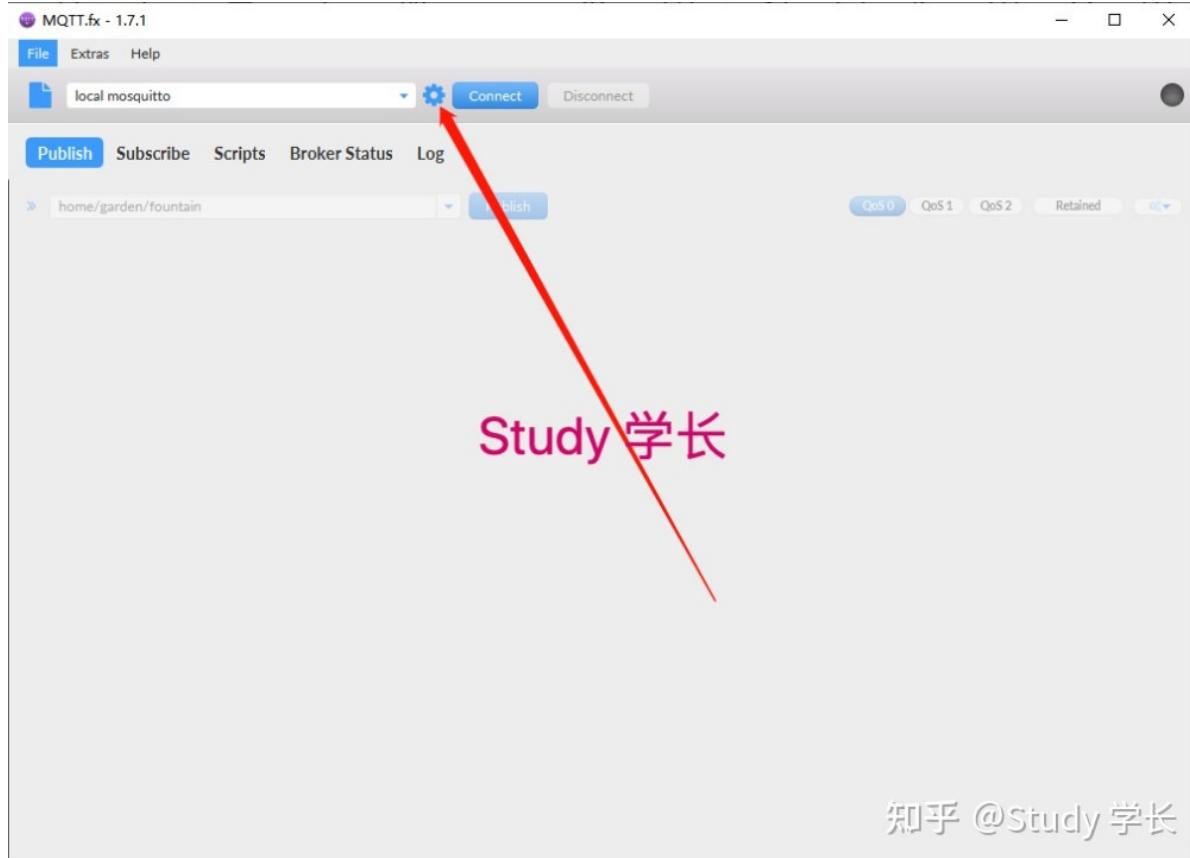
This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

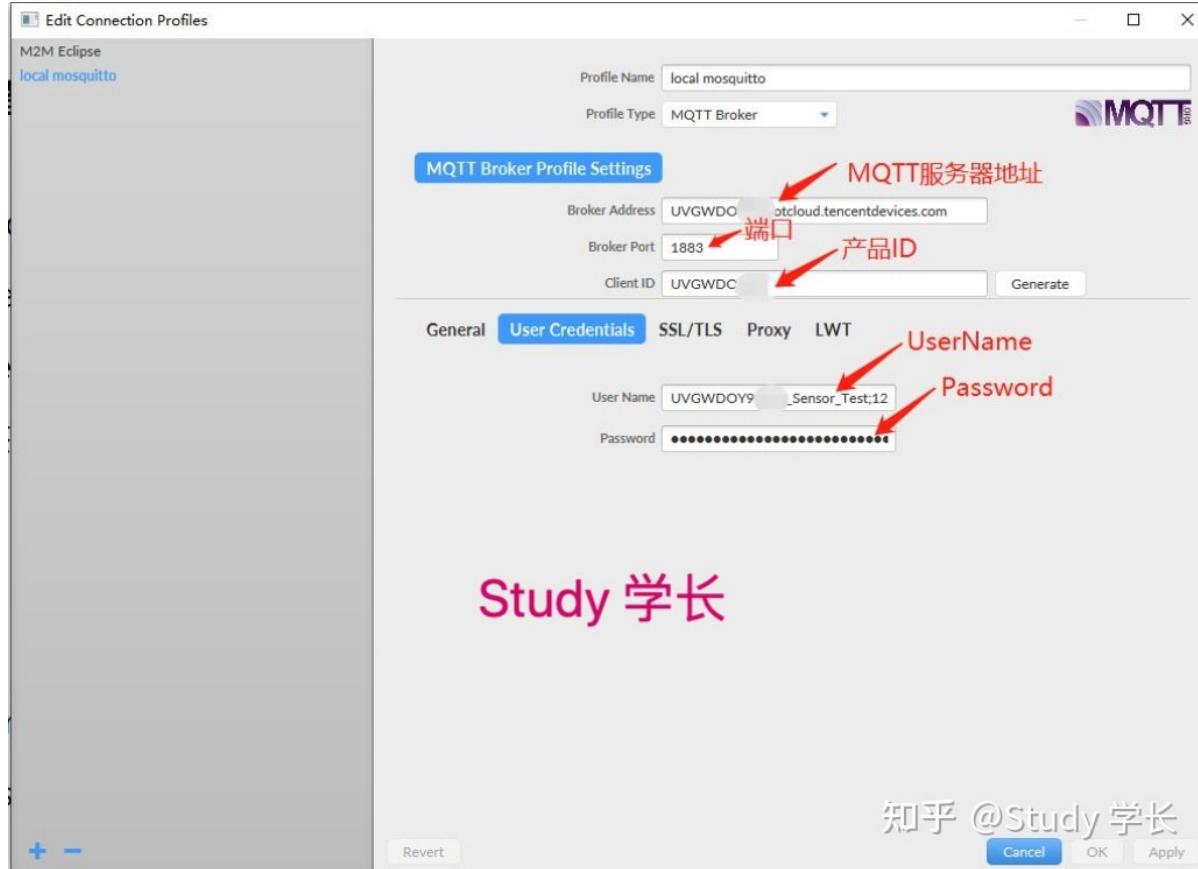
This step explains how to use MQTT.fx to connect to the device you just created.

### MQTT.fx connects to Tencent Cloud

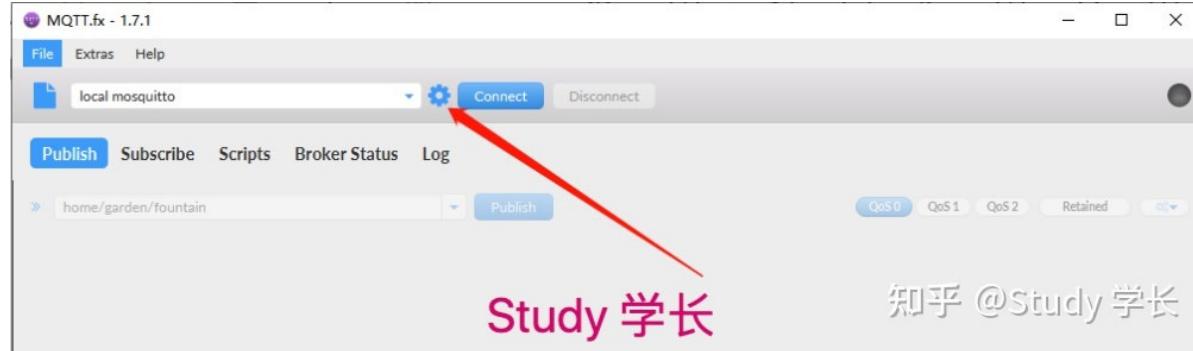
(1) After installation, open MQTT.fx and click the Settings button, as shown in the figure.



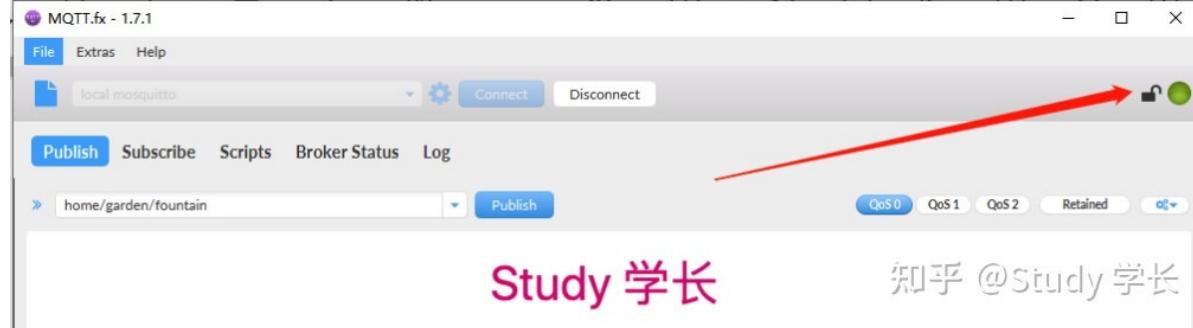
(2) Fill in the relevant information as shown in the figure, and click the Apply button in the lower right corner to save the settings, as shown in the figure.



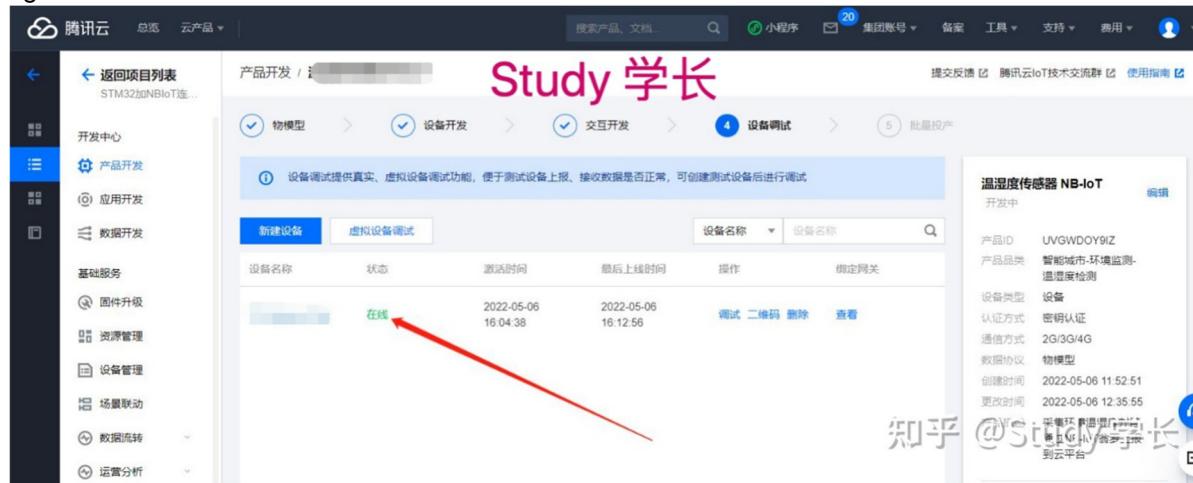
(3) Return to the software homepage and click the Connect button, as shown in the figure.



(4) The MQTT connection is successful, as shown in the figure.



(5) Return to the device debugging page, and you can see that the device status has switched to "Online", as shown in the figure.



(6) In the Tencent LianLian applet, you can also see that the device status has switched to "Online", as shown in the figure.



## 8.2.7. MQTT.fx publishes messages

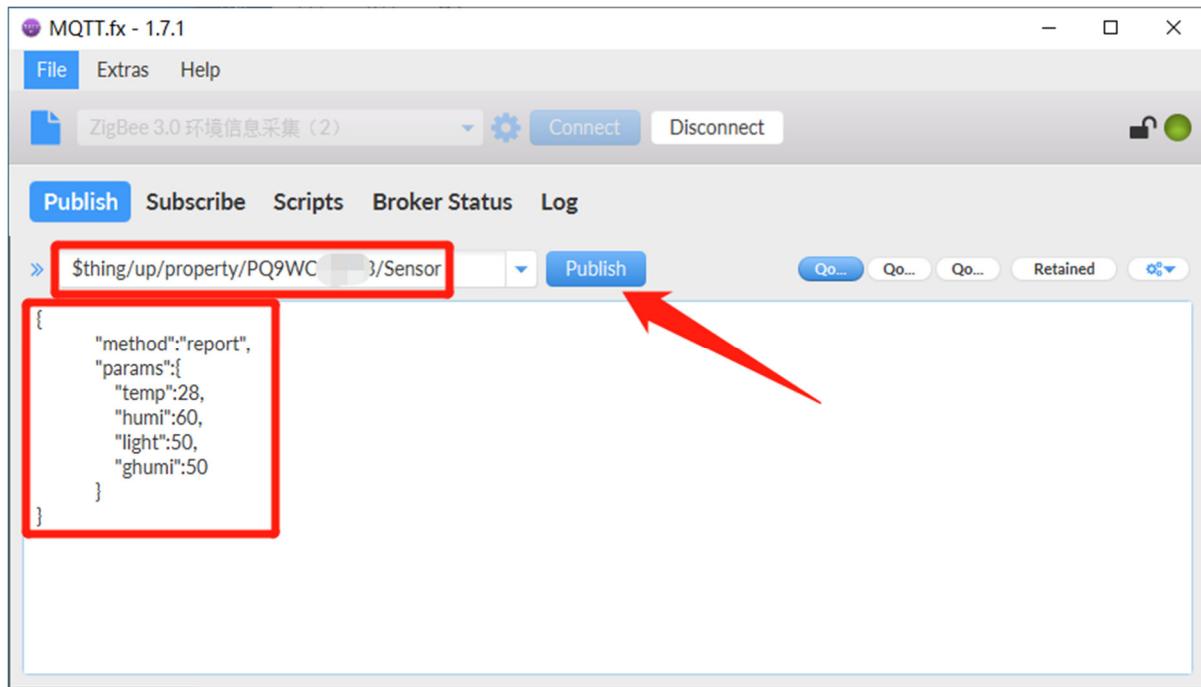
This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### MQTT.fx publishes messages

(1) After MQTT.fx successfully connects to Tencent Cloud, copy the attribute publishing topic obtained from Tencent Cloud into the Topic input box, copy the following code string into the Send box, and then click the Publish button, as shown in the figure.

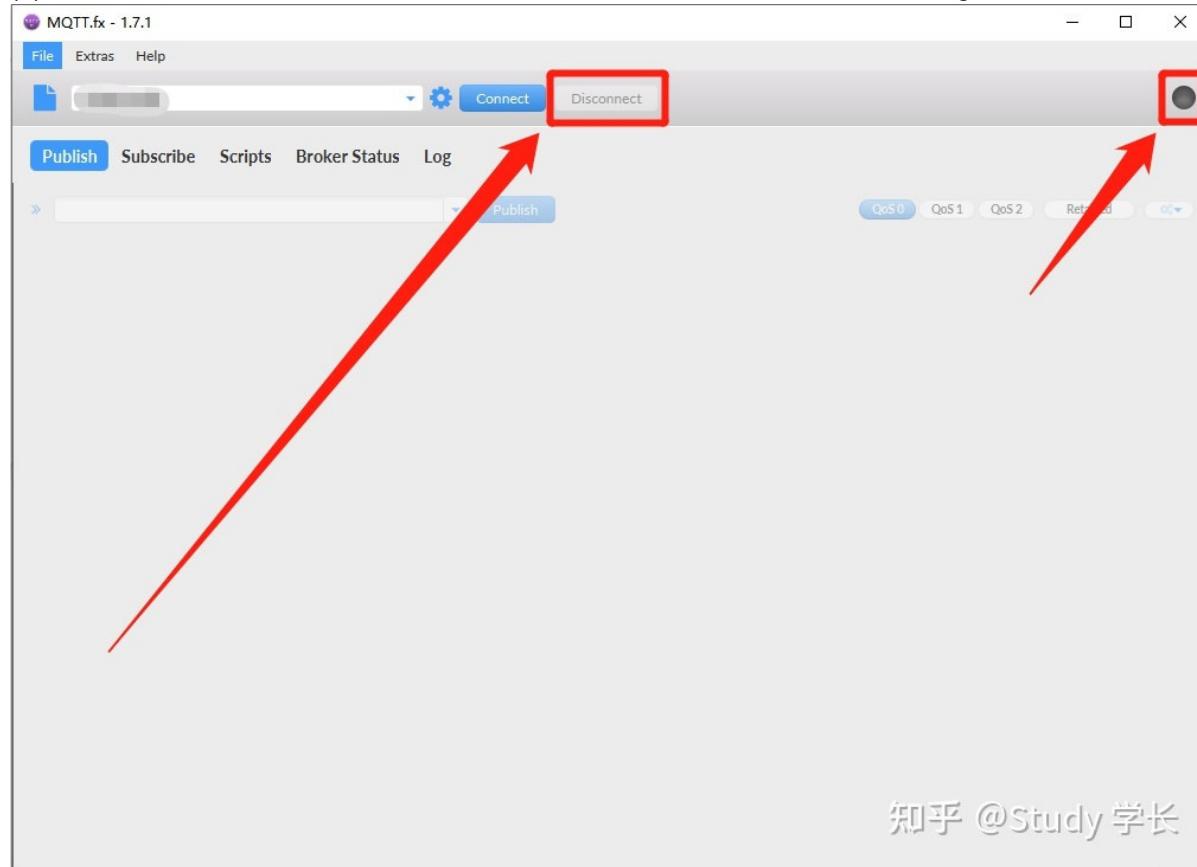
```
{
    "method": "report",
    "params": {
        "temp": 28,
        "humi": 60,
        "light": 50,
        "ghumi": 50
    }
}
```



(2) Then, you can see that the temperature and humidity have been updated in the Tencent LianLian mini program, as shown in the figure.



(3) Click the Disconnect button to close the MQTT connection, as shown in the figure.



知乎 @Study 学长

## 8.2.8. Configuring the Development Board

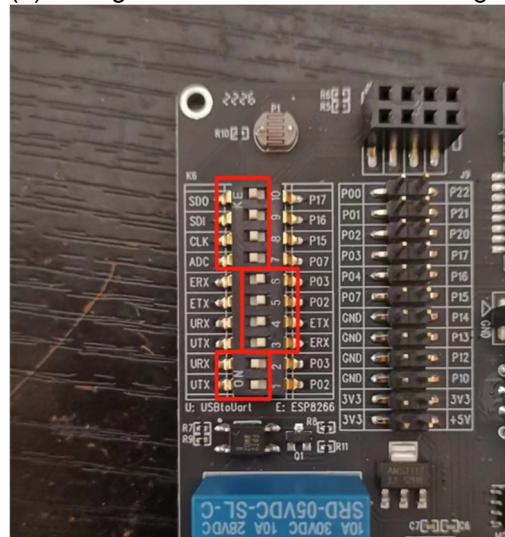
This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

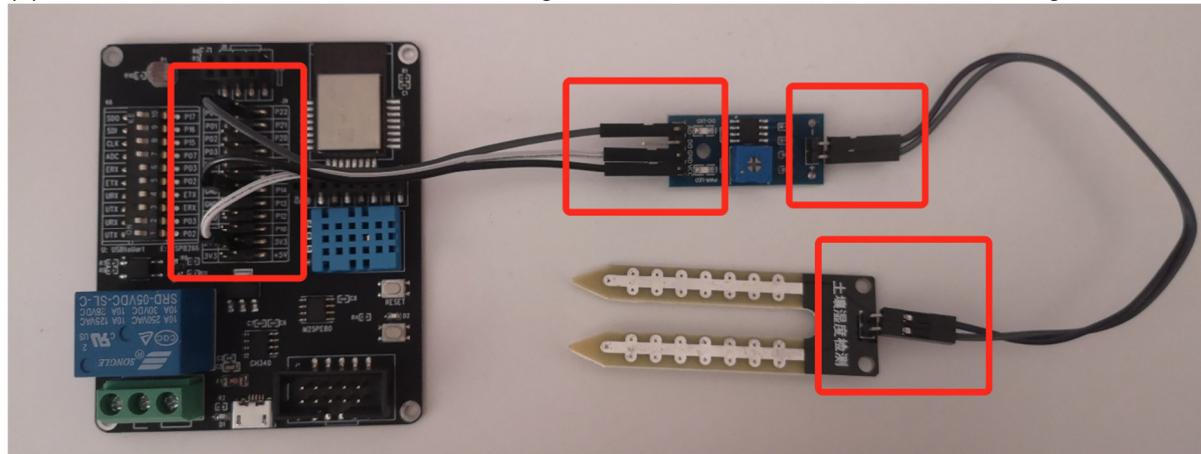
### Configuring the Development Board

(1) Install the necessary development software and driver software according to Part 1: Preparation.

(2) Configure the DIP switches in the ZigBee 3.0 standard board as shown in the figure.



(3) Connect the soil moisture sensor to the ZigBee 3.0 standard board as shown in the figure.



The pin connection instructions are as follows:

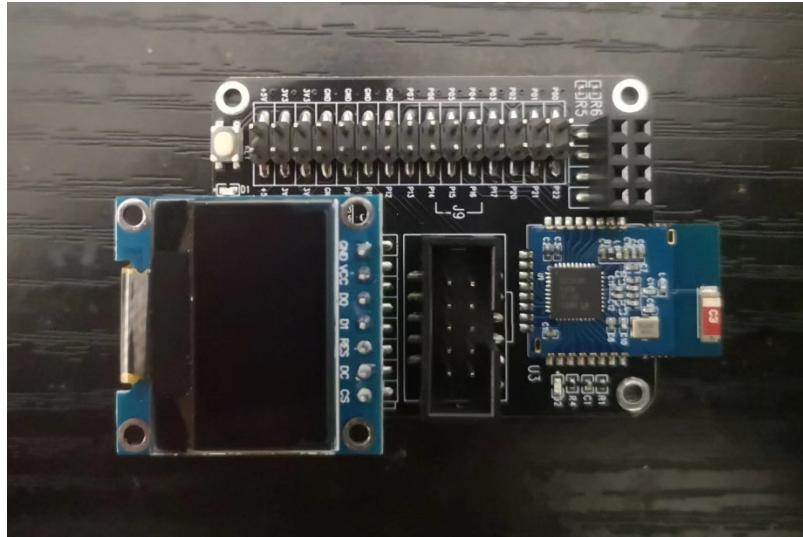
- (3A) The soil moisture probe and the sensor main board can be connected with two Dupont wires, and there is no requirement for pin docking;
- (3B) The VCC pin of the sensor main board is connected to any 3v3 pin of the ZigBee standard board;
- (3C) The GND pin of the sensor main board is connected to any GND pin of the ZigBee standard board;
- (3D) The AO pin of the sensor main board is connected to the P0\_0 pin of the ZigBee standard board;
- (3E) The DO pin of the sensor main board is left blank;

**Tip:** Why does the AO pin above need to be connected to the P0\_0 pin? This is determined by the code written. Readers can also modify the code to use other pins.

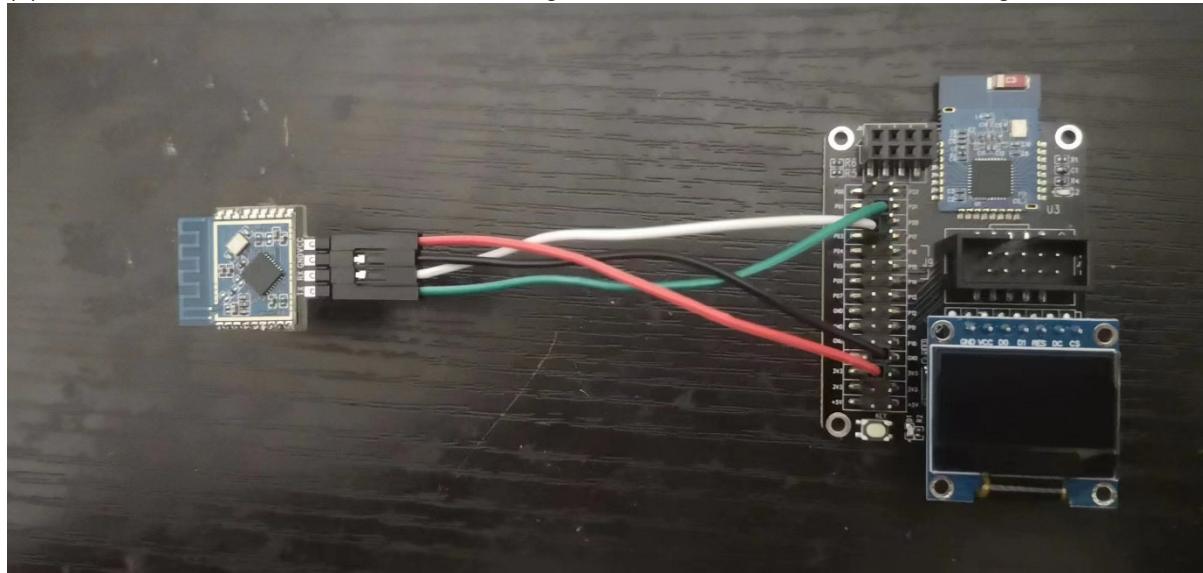
(4) Insert the display into the ZigBee 3.0 standard board, as shown in the figure.



(5) Insert the display into the ZigBee 3.0 Mini board as shown in the figure.



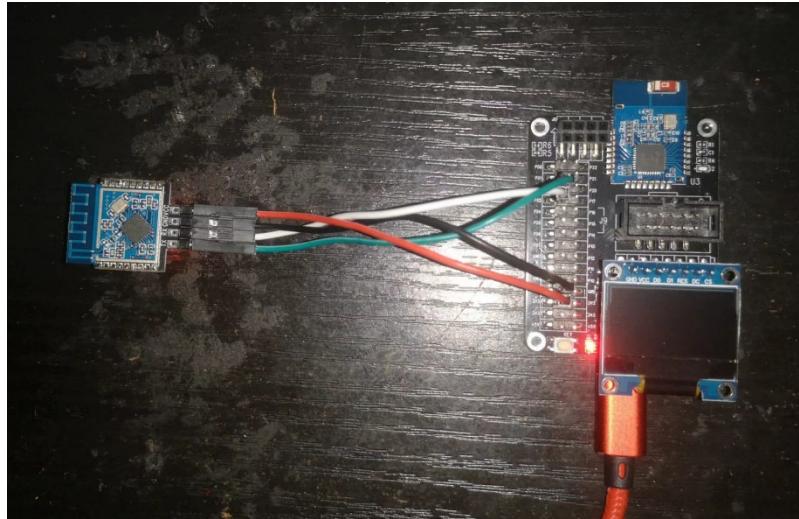
(6) Connect the 32C3-1 WiFi module to the ZigBee 3.0 Mini board as shown in the figure.



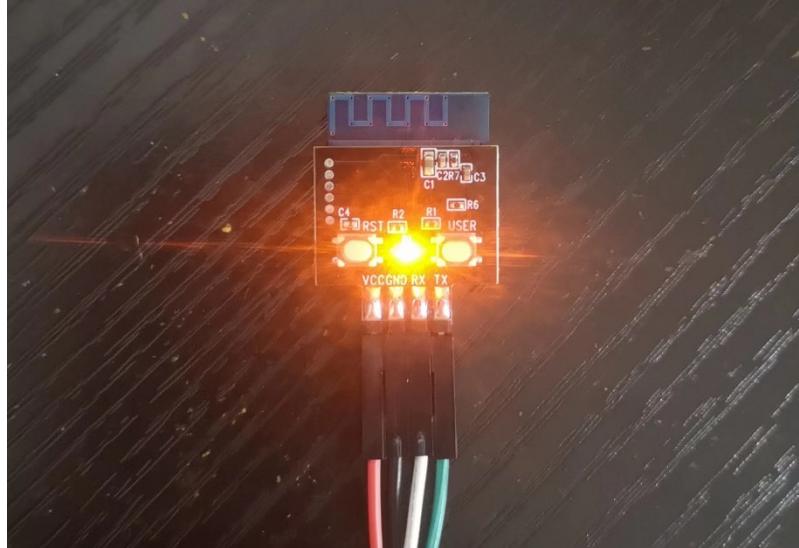
The pin connection instructions are as follows:

- Connect the VCC of 32C3-1 to the 3v3 of the Mini board;
- Connect the GND of 32C3-1 to the GND of the Mini board;
- Connect **TX** of 32C3-1 to **P02** of Mini board ;
- Connect **the RX** of 32C3-1 to **P03** of the Mini board.

(7) Use a Micro USB cable to power the ZigBee 3.0 Mini board, as shown in the figure.



(8) The indicator light on the back of the 32C3-1 module will flash rapidly, as shown in the figure.



(9) Configure the 32C3-1 module according to the specification.

- Specifications: <https://z7po9bxpe4.k.tophink.com/@wifi>
- For network configuration methods, refer to "Specifications" → "Module Access Instructions" → "Bluetooth Smart Network Configuration": <https://z7po9bxpe4.k.tophink.com/@wifi/3.1zhinenglanyapeiwang.html>

## 8.2.9. Burn the coordinator firmware

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.  
(2) If you have not learned parts 1 to 4, you must stop reading!  
(3) In principle, this part **does not have any technical support**. Please understand!

### 1. Build a development environment

Install the development software according to Part 1: Preparation of this tutorial.

### 2. Modify MQTT parameters

(1) After downloading the source code of this project, open the ZLinker.eww file, as shown in the figure.

名称	修改日期	类型	大小
Category	2023/3/18 22:37	文件夹	
Settings	2023/3/18 22:37	文件夹	
Tools	2023/3/18 22:37	文件夹	
BuildLog.log	2021/3/27 15:38	文本文档	1 KB
ZLinker.dep	2023/3/30 16:16	DEP 文件	375 KB
ZLinker.ewd	2021/3/29 0:47	EWD 文件	75 KB
ZLinker.ewp	2023/3/19 11:21	EWP 文件	160 KB
ZLinker.ewt	2023/3/18 11:29	EWT 文件	265 KB
ZLinker.eww	2021/3/27 14:39	IAR IDE Works...	1 KB

(2) Open the userconf.h file, as shown in the figure.

```
#ifndef __USERCONF_H__
#define __USERCONF_H__

/* 远程服务器信息配置 */
#define USERCONF_MQTT_URL      ("PQ9WC1[REDACTED].iotcloud.tencentdevices.com:1883")
#define USERCONF_MQTT_USERNAME   ("PQ9WC1[REDACTED]_Sensor;12010126;T2D0L;1970496[REDACTED]3")
#define USERCONF_MQTT_PASSWORD   ("764ea15b28c78bb7703022471a087c40152d738988d72e374bf1b985fc0[REDACTED]a;hmacsha256")
#define USERCONF_MQTT_RECV_TOPIC  ("$thing/down/property/PQ9WC1[REDACTED]3/Sensor")
#define USERCONF_MQTT_RECV_QOS    (1)
#define USERCONF_MQTT_SEND_TOPIC  ("$thing/up/property/PQ9WC1[REDACTED]3/Sensor")
#define USERCONF_MQTT_SEND_QOS    (1)
#define USER_UUID                 ("PQ9WC1[REDACTED]3")

#endif /* #ifndef __USERCONF_H__ */
```

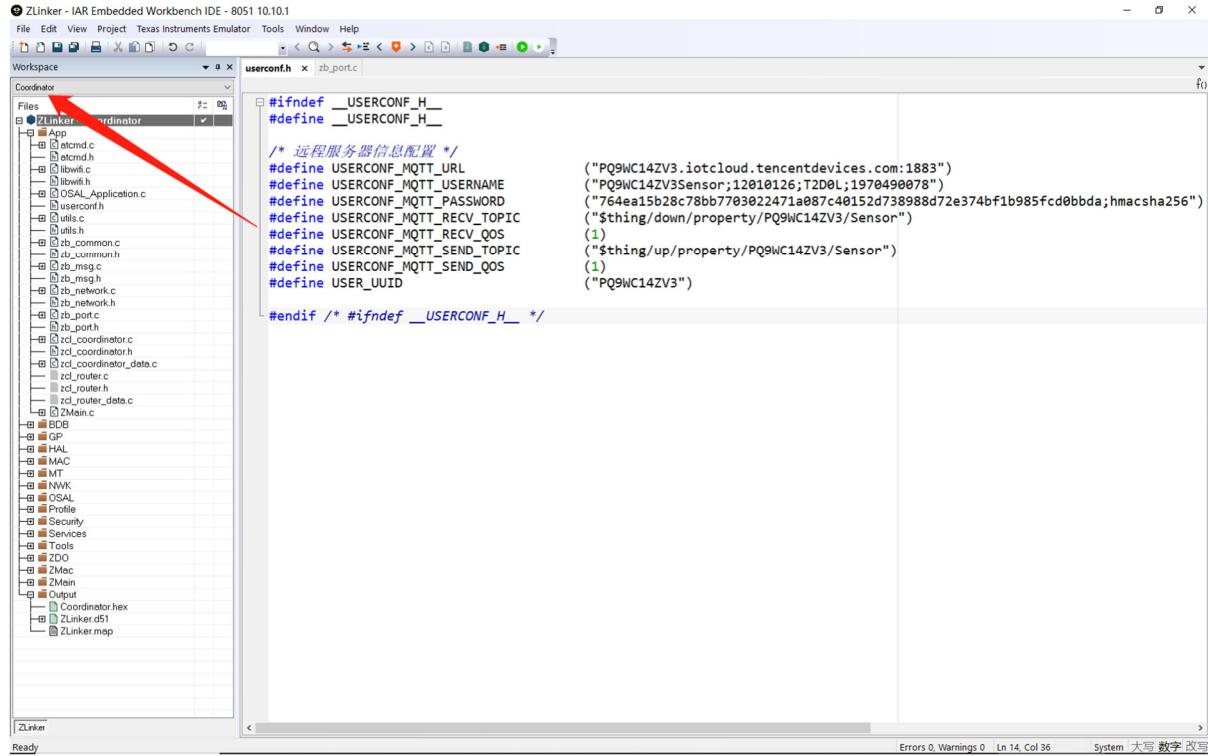
(3) Fill in the Tencent Cloud MQTT service information as follows:

- USERCONF\_MQTT\_URL: server address: **port number**
- USERCONF\_MQTT\_USERNAME: Username
- USERCONF\_MQTT\_PASSWORD : Password
- USERCONF\_MQTT\_RECV\_TOPIC: Property **subscription** topic
- USERCONF\_MQTT\_SEND\_TOPIC: property **publishing** topic
- USER\_UUID: Product ID

(4) Check again whether the information you have filled in is correct.

### 3. Burn the coordinator firmware

(1) Make sure Coordinator is selected, as shown in the figure.



The screenshot shows the IAR Embedded Workbench IDE interface. The workspace contains a project named "Coordinator". The project tree on the left shows various files under "Coordinator" and "ZLinker". The main editor window displays the content of "userconf.h". The code defines MQTT configuration parameters such as URL, username, password, topics, and QoS levels. The status bar at the bottom indicates "Errors 0, Warnings 0, Ln 14, Col 36".

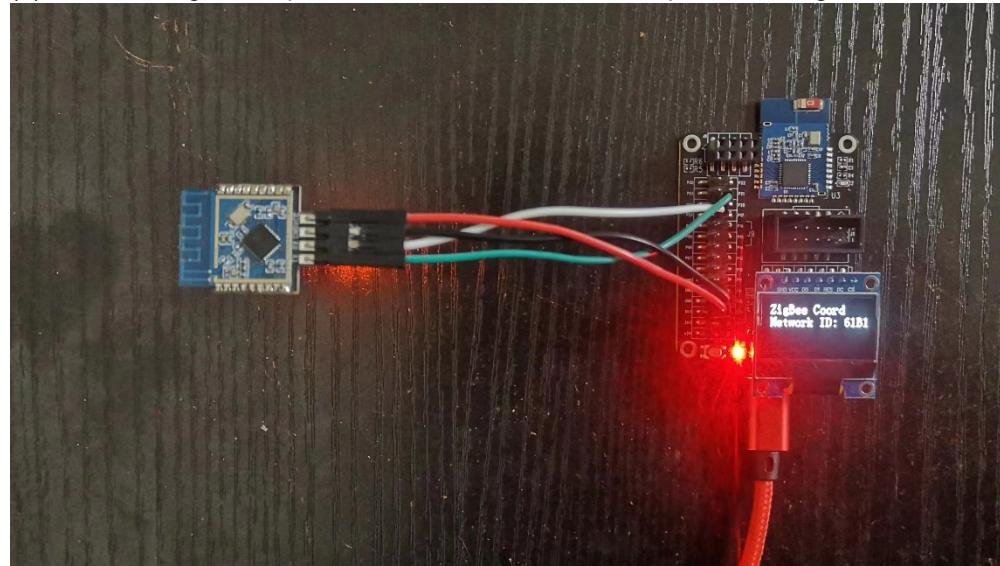
```
#ifndef __USERCONF_H__
#define __USERCONF_H__

/* 远程服务器信息配置 */
#define USERCONF_MQTT_URL ("PQ9WC14ZV3.iotcloud.tencentdevices.com:1883")
#define USERCONF_MQTT_USERNAME ("PQ9WC14ZV3Sensor;12010126;T2D01;1970490078")
#define USERCONF_MQTT_PASSWORD ("764ea15b28c78bb7703022471a087c40152d738988d72e374bf1b985fc0bbda;hmacsha256")
#define USERCONF_MQTT_RECV_TOPIC ("$thing/down/property/PQ9WC14ZV3/Sensor")
#define USERCONF_MQTT_SEND_TOPIC ("$thing/up/property/PQ9WC14ZV3/Sensor")
#define USERCONF_MQTT_SEND_QOS (1)
#define USERCONF_UUID ("PQ9WC14ZV3")

#endif /* #ifndef __USERCONF_H__ */
```

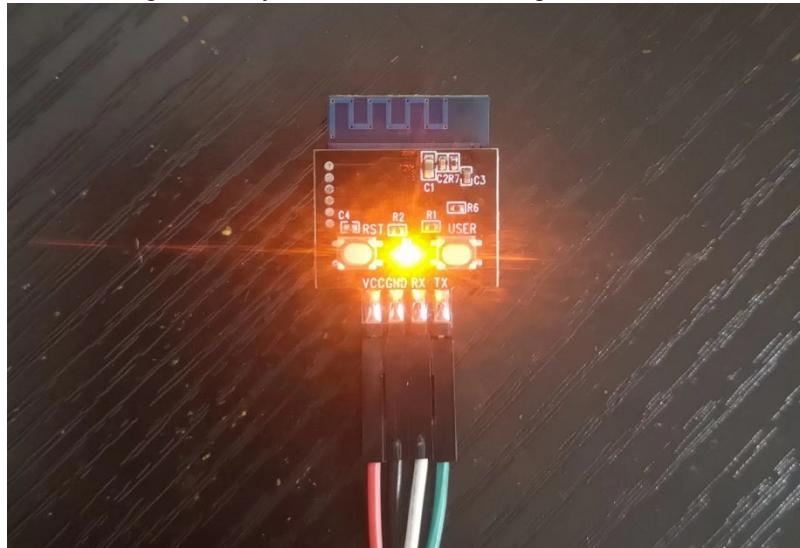
(2) Compile and link the source code, generate the firmware, and burn it into the ZigBee 3.0 Mini board. For detailed operation steps, please refer to "Source code writing and compilation" and "Program download and simulation" in "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: CC2530 Basic Development Experiment".

(3) After burning is complete, use a Micro USB cable to power the ZigBee 3.0 Mini board, as shown in the figure.



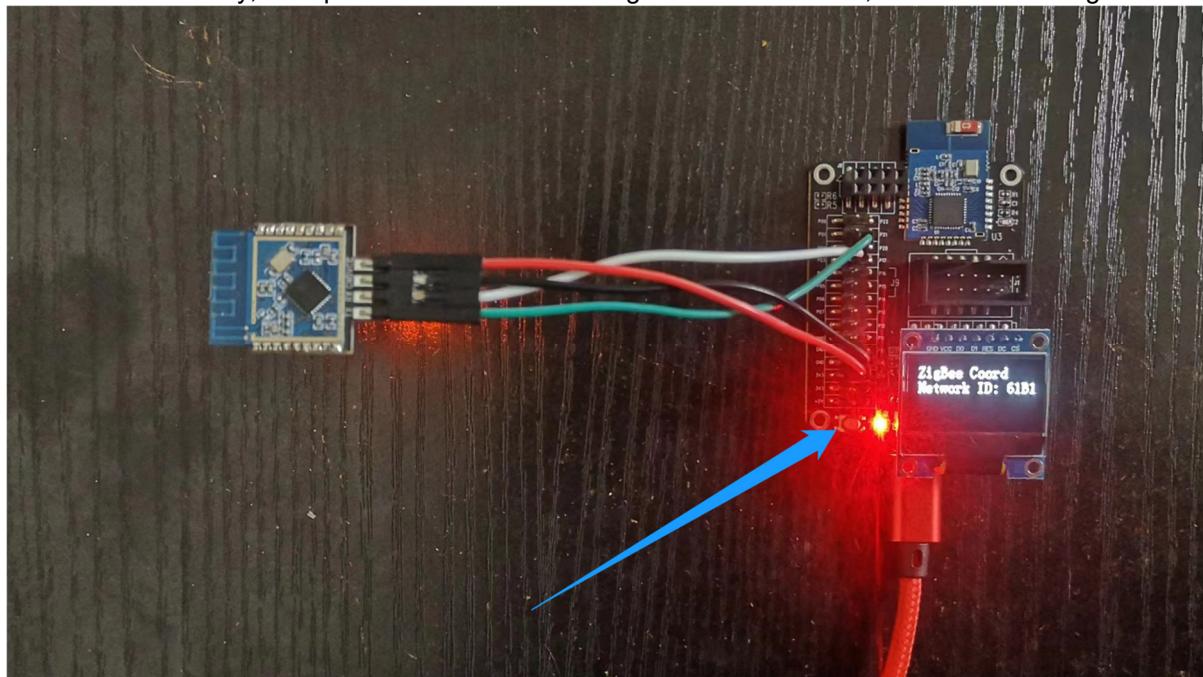
(4) After the ZigBee 3.0 Mini board is initialized, "ZigBee Coord Network ID: XXX X" will be displayed on the screen.

(5) After the 32C3-1 module is networked according to the above instructions, the indicator light on its back will change from slow flashing to steady on, as shown in the figure.

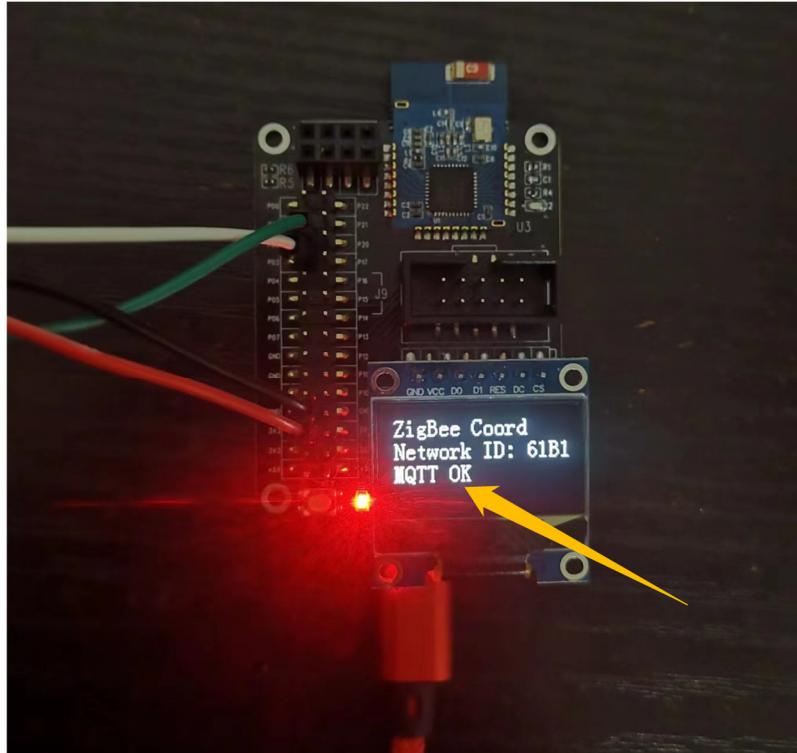


**Tip:** If the indicator light fails to change from slow flashing to steady on, turn off the power for 3 seconds and then turn it on again.

(6) Make sure that: A. the indicator light on the 32C3-1 module is always on; B. the connected WiFi hotspot or router can access the Internet normally, then press the button on the ZigBee 3.0 Mini board, as shown in the figure.



(8) The display will finally show "MQTT OK", as shown in the figure.



**Tip:** If the indicator light fails to display "MQTT OK", you can turn off the power for 3 seconds and then turn it on again to test again.

## 8.2.10. Burning Router Firmware

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### 1. Build a development environment

Install the development software according to Part 1: Preparation of this tutorial.

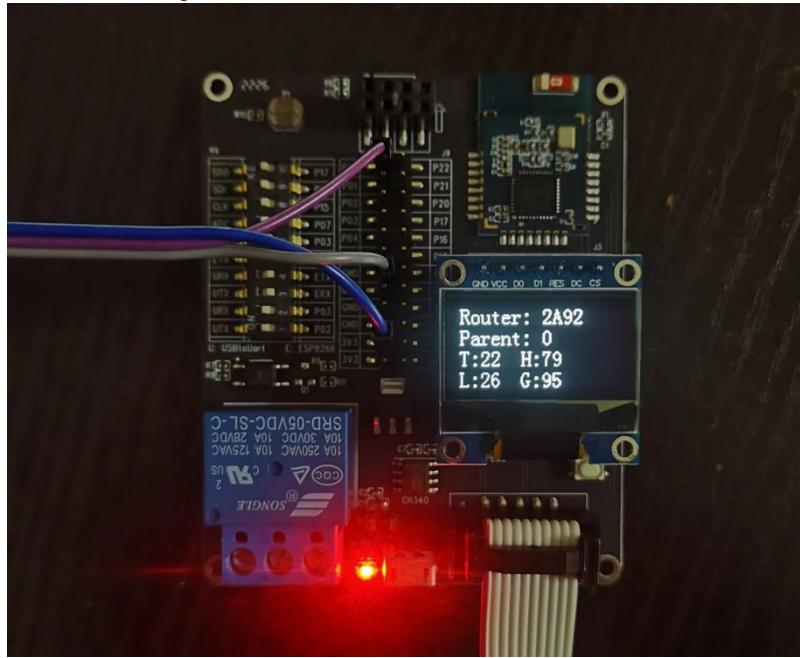
### 2. Burn the router firmware

- (1) After downloading the source code of this project, find the router firmware, as shown in the figure.

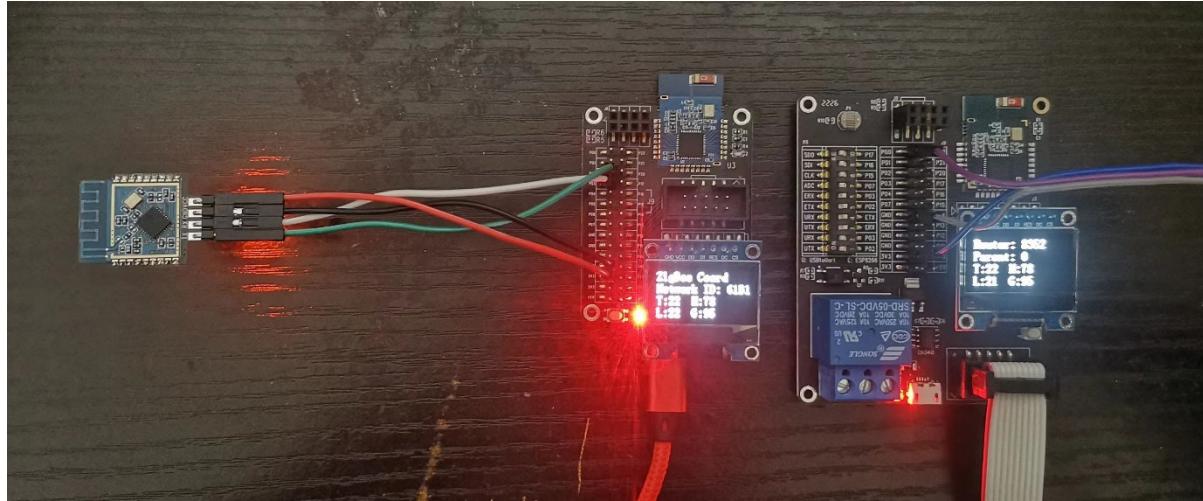
名称	修改日期	类型	大小
路由器固件 (烧到到标准板) .hex	2023/3/30 23:31	HEX文件	601 KB

- (2) Burn this firmware into the ZigBee 3.0 standard board. For detailed operation steps, please refer to "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: CC2530 Development Basic Experiment" → "Firmware Burning".

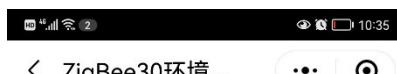
(3) The ZigBee 3.0 standard board will automatically collect ambient temperature, humidity, light intensity and soil moisture, as shown in the figure.



(4) The ZigBee 3.0 standard board will automatically join the ZigBee network created by the Mini board and send data to the Mini board, as shown in the figure.



(5) After receiving the data, ZigBee 3.0 Mini will automatically send it to Tencent Cloud. Users can see the data update in the Tencent LianLian applet.



环境温度

28°C

环境温度

28°C

环境湿度

60%

环境光照度

50

土壤湿度

50

## 8.2.11. Source code description

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

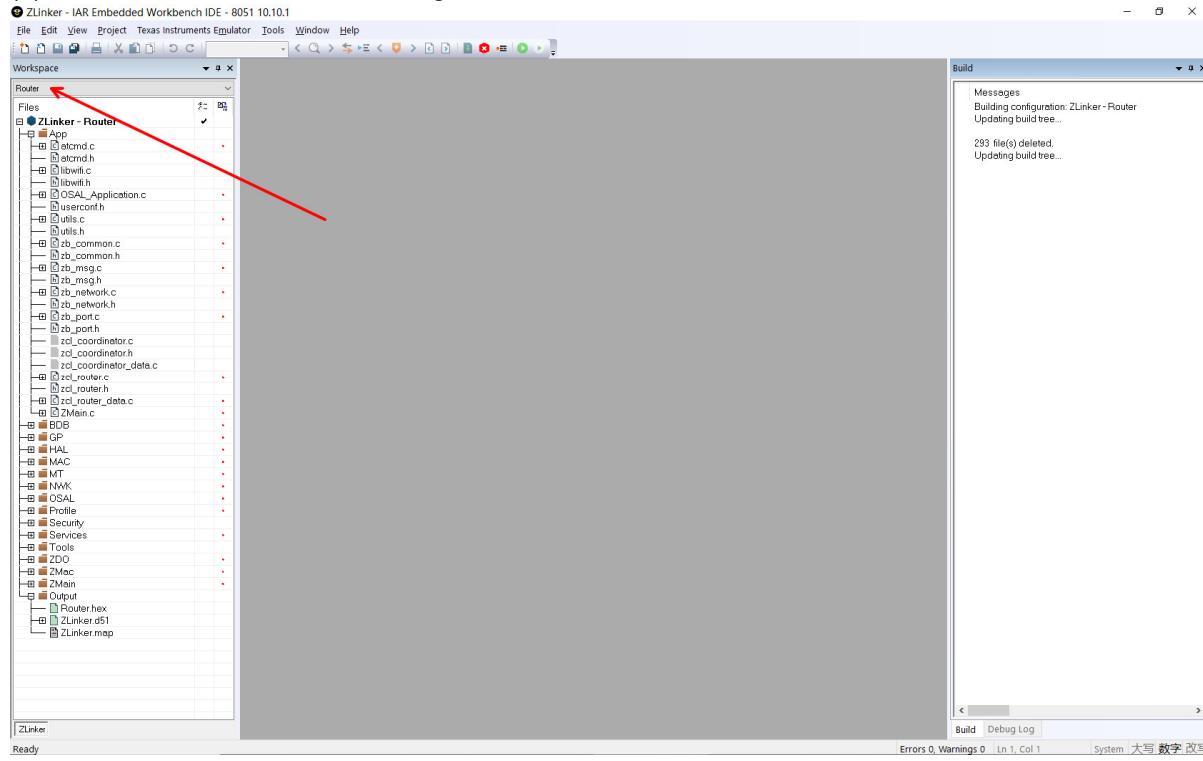
### 1. Compile the source code

(1) Install the development software according to "Part 1: Preparation" of this tutorial.

(2) Use IAR EW for 8051 to open the project file, as shown in the figure.

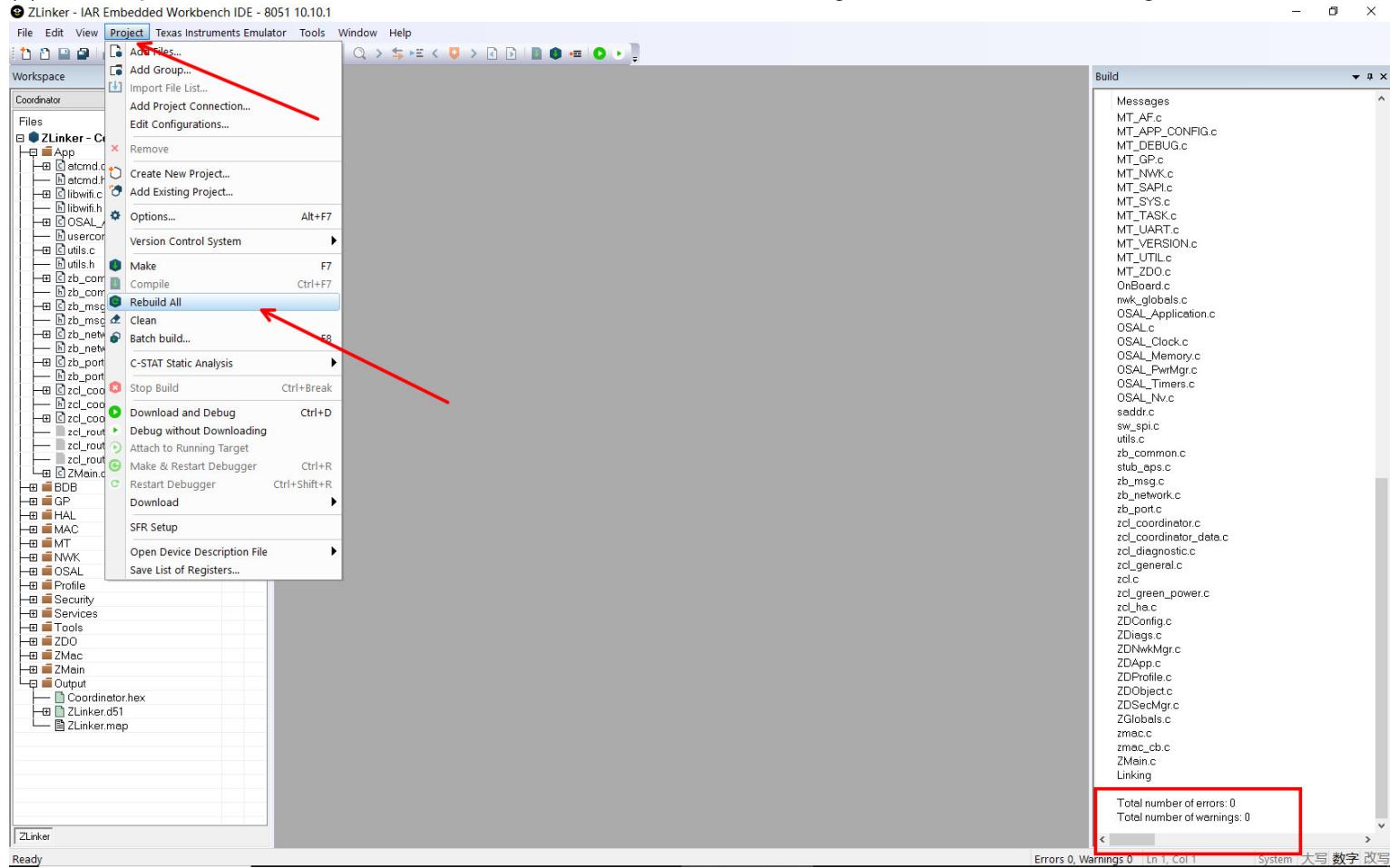
此电脑 > 文件 (D:) > 学习资料 > ZigBee > ZigBee 3.0 环境信息采集 (2) > Projects > CC2530			
名称	修改日期	类型	大小
Category	2023/3/18 22:37	文件夹	
Settings	2023/3/18 22:37	文件夹	
Tools	2023/3/18 22:37	文件夹	
BuildLog.log	2021/3/27 15:38	文本文档	1 KB
ZLinker.dep	2023/4/19 13:46	DEP 文件	376 KB
ZLinker.edw	2021/3/29 0:47	EWD 文件	75 KB
ZLinker.ewp	2023/3/19 11:21	EWP 文件	160 KB
ZLinker.ewt	2023/3/18 11:29	EWT 文件	265 KB
ZLinker.eww	2021/3/27 14:39	IAR IDE Works...	1 KB

### (3) Select Router, as shown in the figure.

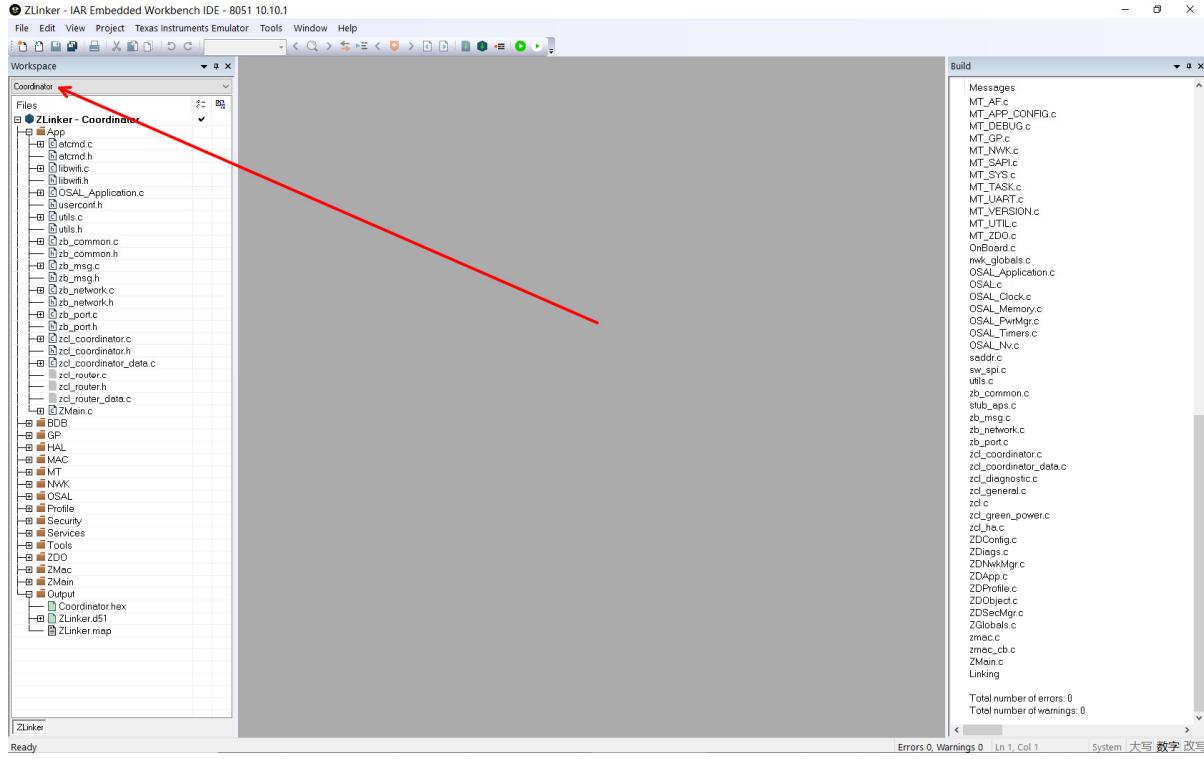


Note: You need to select the Router role, and you cannot select EndDevice for the time being.

### (4) The compilation and connection are successful, and the node firmware is generated, as shown in the figure.

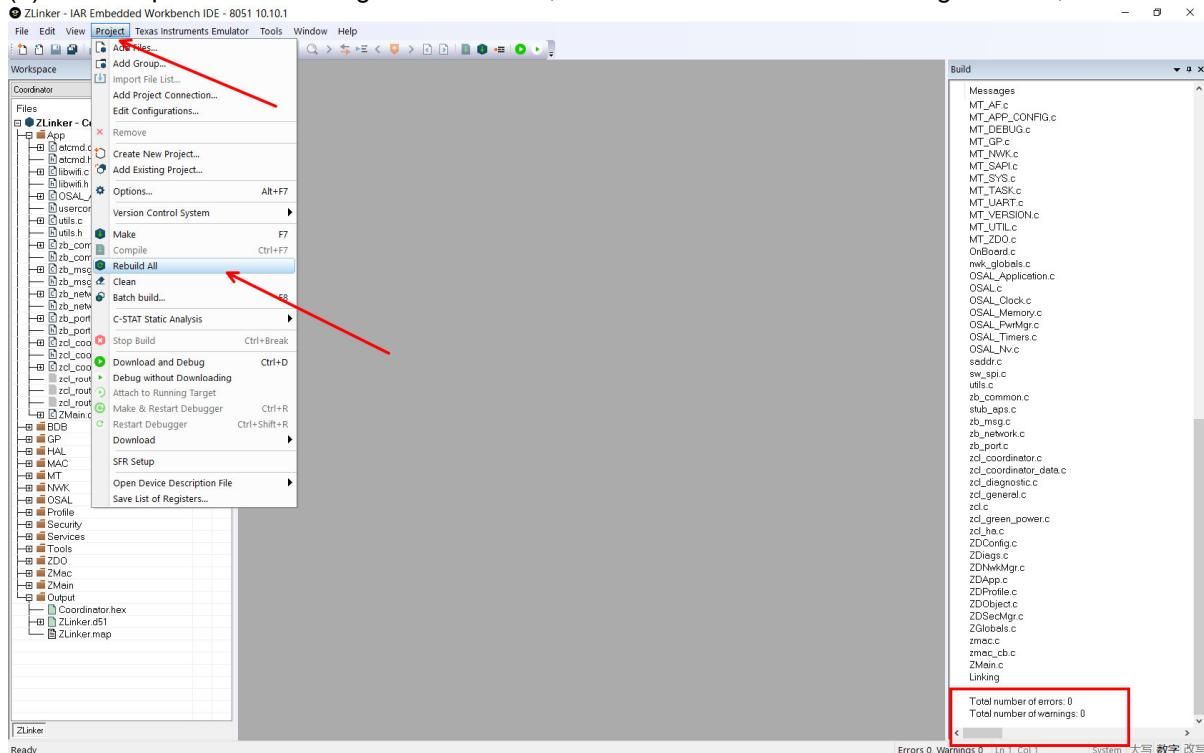


## (5) Select Coordinator, as shown in the figure.



**Note:** If you need to connect to the cloud platform, you need to modify the cloud platform parameters. For specific operations, please refer to "Burn the Coordinator Firmware" in this project tutorial.

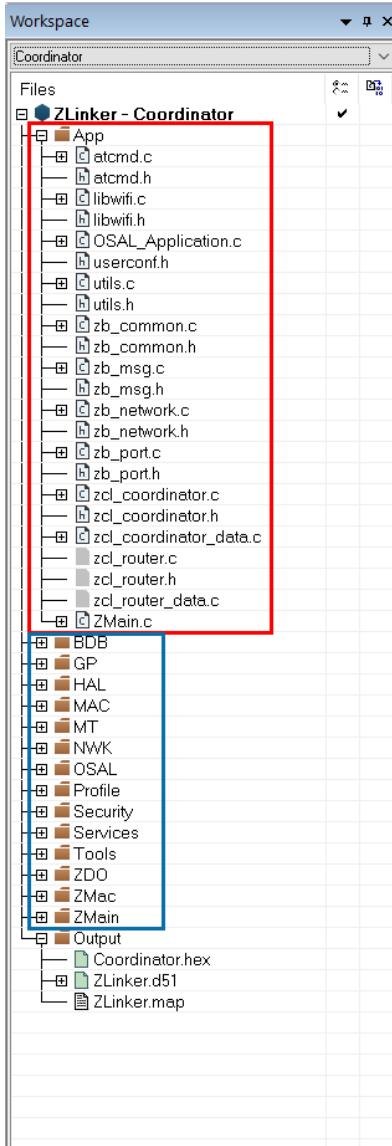
## (6) The compilation and linking are successful, and the coordinator firmware is generated, as shown in the figure.



## 2. Source code file structure

**Note:** Since the first to fourth parts have explained each part of the code in detail, they will not be explained in detail here again. Please be sure to fully understand the first to fourth parts before reading this project code.

(1) The source code of this project is shown in the figure.



(2) The code in the blue box is mainly the code of Z-Stack 3.0 itself, provided by Texas Instruments. For detailed explanation of Z-Stack 3.0, please refer to "Part 3: Z-Stack 3.0 Detailed Explanation" and "Part 4: ZigBee 3.0 Network Programming" of this tutorial.

(3) The code in the red box is mainly written by us for this project. The brief introduction of each file is as follows:

- (A) atcmd.h/c: WiFi module SDK file.
- (B) libwifi.h/c: WiFi module SDK file.
- (C) utils.h/c: WiFi module SDK file.
- (D) userconfig.h/c: WiFi module SDK file.
- (E) OSAL\_Application.c: OSAL application layer code. For detailed description, please refer to "Part 3: Z-Stack 3.0 Detailed Explanation" → "Chapter 3: OSAL Detailed Explanation"
- (F) zb\_common.h/c: stores common definition content.
- (G) zb\_msg.h/c: Files related to Uart functions.
- (H) zb\_network.h/c: Files related to ZigBee networking.
- (I) zb\_port.h/c: stores files related to general logic processing.
- (J) zcl\_coordinator.h/c: Files related to coordinator functions.
- (K) zcl\_coordinator\_data.c: File related to coordinator data.

- (L) zcl\_router.h/c: Files related to routers.
- (M) zcl\_router\_data.c: Files related to router data.
- (N) ZMain.c: The file where the main() function is located.

(4) **zb\_port.h/c** is the file most relevant to the function of this project, and if the function of this project needs to be modified, it is generally started from **zb\_port.h/c**. Therefore, the following article will focus on **zb\_port.h/c**.

### 3. Detailed explanation of zb\_port.h/c files

(1) A zb\_port component is defined in zb\_port.h, and the content is as follows:

```
#ifndef __ZB_PORT_H__
#define __ZB_PORT_H__

#include "zb_common.h"

#ifndef __cplusplus
extern "C"
{

#endif

/*
* zb_port 组件定义
*/
typedef struct {
    void (*init)(void);
    void (*on_click)(void);
    void (*on_priv_msg)(uint16_t nwkaddr, uint16_t cluster, const uint8_t *data, uint16_t len);
    void (*on_timer)(uint16_t ms);
} zb_port_t;

extern const zb_port_t *zbport_;


#endif /* __ZB_PORT_H__ */
```

(2) Open the zb\_port.c file. The zb\_port component initialization function zb\_port\_init() mainly performs some preparatory work. The code is as follows.

```
/*
* zb_port组件初始化
```

```
*/  
void zb_port_init()  
{  
    //printf("zb_port_init()\n");  
  
    /* 协调器 */  
  
    #if ZG_BUILD_COORDINATOR_TYPE  
        /* 所需的组件初始化 */  
  
        zbmmsg_->init();  
        atcmd_init();  
        libwifi_init();  
  
        /* 配置串口接收回调函数 */  
        zbmmsg_->set_receiver(libwifi_on_message);  
  
        /* 配置WiFi模块的发送器 */  
        atcmd_set_sender(zbmmsg_->send);  
  
        /* 配置WiFi模块的回调函数 */  
        libwifi_on_responded_callback(task_nwk_on_responded);  
        libwifi_on_ip_ready_callback(task_nwk_on_ip_ready);  
        libwifi_on_mqtt_connected_callback(task_nwk_on_mqtt_connected);  
        libwifi_on_mqtt_disconnected_callback(task_nwk_on_mqtt_disconnected);  
        libwifi_on_mqtt_message_callback(task_nwk_on_mqtt_message);  
  
        /* 协调器允许入网 */  
        zbnwk_->permit(255, 0);  
  
        /* 初始化WiFi网络状态标记 */  
        task_nwk_status_ = TASK_NWK_INIT;  
  
        /* 初始化WiFi网络繁忙标记 */  
        task_nwk_is_busy_ = 0;  
  
        /* 路由器 */  
  
    #elif ZG_BUILD_RTRONLY_TYPE  
        /* 初始化所用到的GPIO */  
        CC2530_IO_INPUT(0, 0, CC2530_INPUT_PULLUP);  
    #else  
  
    #endif  
}
```

(3) In the coordinator role, when the user presses the onboard User button, zb\_port\_on\_click() is automatically called. The code is as follows.

```
/*
 * 按钮事件处理函数
 */
void zb_port_on_click(void)
{
    //省略
}
```

(4) The zb\_port\_on\_timer() function will be automatically executed every 1 second. You can add code that needs to be executed repeatedly.

```
/*
 * 每隔1秒自动执行1次本函数
*/
void zb_port_on_timer(uint16_t ms)
{
    //省略
}
```

(5) When a ZigBee message is received, the zb\_port\_on\_priv\_msg() function will be called automatically. You can add message processing code in it.

```
/*
 * 接收到ZigBee消息时，自动调用此函数
 *
 * @param nwkaddr 发送者的网络地址
 * @param cluster Cluster
 * @param data 数据内容
 * @param len 数据长度
 */
void zb_port_on_priv_msg(uint16_t nwkaddr, uint16_t cluster, const uint8_t *data, uint16_t len)
{
    //省略
}
```

## 8.3. ZigBee Temperature and Humidity Monitoring & Sound and Light Alarm System (1)

### 1. Required Reading

This course is an extracurricular course, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise **please stop studying!**

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530

- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

## 2. Contents

- 8.3.1. System Introduction
- 8.3.2. Source code download
- 8.3.3. Hardware List
- 8.3.4. System Construction
- 8.3.5. Source code description

### 8.3.1. System Introduction

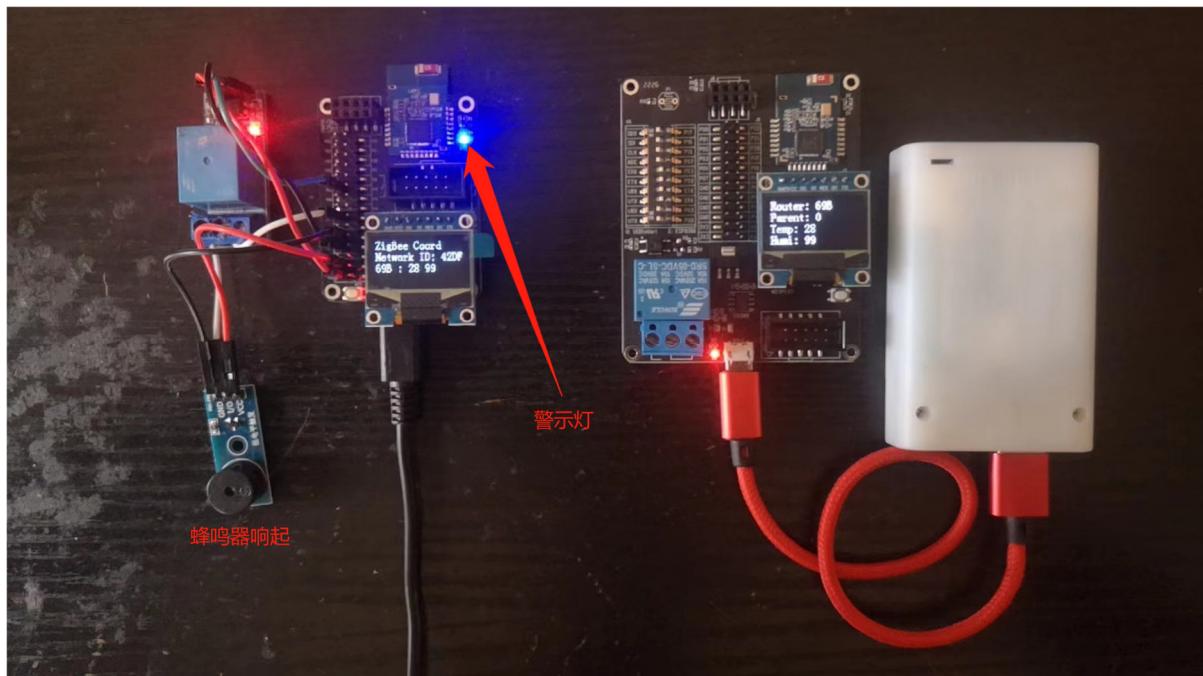
This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### System Introduction

- (1) This system can collect ambient temperature and humidity and report them to the ZigBee coordinator;
- (2) When abnormal temperature or humidity is detected, the warning light will automatically turn on and the buzzer will sound an alarm (controlled by a relay);
- (3) It supports the deployment of one coordinator and **multiple temperature and humidity collection nodes**.

### System Demonstration



### 8.3.2. Source code download

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

## Source code download

(1) Users who have purchased the product can download the additional materials from the "Supplementary Materials Download" section in the catalog.

(2) This item is located in the additional materials as shown in the figure.

返回上一级 | 全部文件 > ... > ZigBee > 《ZigBee3.0开发指... > 谈外篇：项目实战

<input type="checkbox"/>	文件名	修改时间	大小
	其他项目 1	2022-06-29 10:59	-
	其他项目 2	2022-06-29 10:59	-
	ZigBee 温湿度监测 & 声光报警系统 (1) .zip	2023-04-04 14:31	56.4M
	ZigBee 3.0 环境信息采集 (2) .zip	2023-03-31 12:53	56.1M
	ZigBee 3.0 环境信息采集 (1) (多节点版) .zip	2023-03-30 16:28	281.5M
	ZigBee 3.0 环境信息采集 (1) .zip	2023-03-30 16:28	64.6M
	基于Zigbee的温湿度 & 光照度采集系统.zip	2022-06-29 10:59	33.6M
	基于ZigBee的文件传送系统.zip	2022-06-29 10:59	27.1M
	基于ZigBee的光照自动开关窗帘.zip	2022-06-29 10:59	38.8M
	基于ZigBee的农业环境信息采集.zip	2022-06-29 10:59	69.8M

### 8.3.3. Hardware List

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

## Hardware List

(1) ZigBee 3.0 standard board × 1 ( or more ): <https://item.taobao.com/item.htm?id=683089996879>

(2) ZigBee 3.0 Mini board × 1: <https://item.taobao.com/item.htm?id=683089996879>

(3) 0.96 OLED12864 display × 2 (or more, the same number as the development board): <https://item.taobao.com/item.htm?id=683089996879>

(4) SmartRF04EB emulator × 1: <https://item.taobao.com/item.htm?id=683089996879>

(5) Micro USB data cable × 2 (or more, the same number as the development board): Free when purchasing the development board

(6) Dry cell battery box × 2 (or more): <https://item.taobao.com/item.htm?id=683115420758>

(7) Relay & Dupont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

(8) Buzzer & Dupont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

## Buying Guide

**Tip:** You can purchase all the hardware required for this project by purchasing as follows.

(1) ZigBee development kit version selection is shown in the figure.

The screenshot shows a product listing for a "ZigBee 3.0" development board. The main image displays the board with various components and a smartphone showing a temperature and humidity monitoring app. Key details from the page include:

- Price:** ¥ 238.00, **Sale Price:** ¥ 237.98
- Discount:** 淘金币可抵7.13元 (Tmall Gold Coin discount)
- Delivery:** 广东广州 至 广东广州番禺区, 快递 免运费 (Free shipping via express delivery from Guangzhou to Guangzhou Baiyun District)
- Product Type:** WiFi模块+USB串口 (WiFi module + USB serial port)
- LCD Screen:** 2个 (2 pieces)
- Recorder:** SmartRF04EB 麦录器 (Not applicable)
- Version:** ZigBee 标准板+Mini板 (ZigBee Standard Board + Mini Board)

(2) Select the battery box as shown in the figure.

The screenshot shows a product listing for a "干电池盒" (Dry Battery Box). The main image shows a white rectangular box with a USB port. Key details from the page include:

- Price:** ¥ 9.80
- Discount:** 店铺优惠券 (Shop coupon) 30元, 满1500元可用
- Delivery:** 广东广州 至 广东广州番禺区, 快递 免运费 (Free shipping via express delivery from Guangzhou to Guangzhou Baiyun District)
- Color:** 白色 (White)
- Quantity Selection:** A red arrow points to the quantity selector set to 2.
- Buttons:** 立即购买 (Buy Now) and 加入购物车 (Add to Cart)

(3) Relay & DuPont line selection is shown in the figure.

The screenshot shows a product listing for a "松乐 继电器" (Songle Relay). The main image shows a blue relay module with two blue terminal blocks. Key details from the page include:

- Price:** ¥ 10.00
- Discount:** 淘金币可抵0.35元 (Tmall Gold Coin discount)
- Delivery:** 广东广州 至 广东广州番禺区, 快递 免运费 (Free shipping via express delivery from Guangzhou to Guangzhou Baiyun District)
- Connector Type:** 杜邦线 (Duo邦线) (DuPont line)
- Color Selection:** 母对母 10cm长 (Male to Male 10cm long)
- Quantity Selection:** A red arrow points to the quantity selector set to 1.

(4) Buzzer & Dupont line selection as shown in the figure.

The screenshot shows a product listing for a "有源蜂鸣器" (Active Buzzer). The main image shows a blue buzzer module with a black speaker. Key details from the page include:

- Price:** ¥ 5.00-25.10
- Discount:** 淘金币最高可抵商品价 3% (Tmall Gold Coin discount up to 3%)
- Delivery:** 广东广州 至 广东广州番禺区, 快递 免运费 (Free shipping via express delivery from Guangzhou to Guangzhou Baiyun District)
- Connector Type:** 杜邦线 (Duo邦线) (DuPont line)
- Color Selection:** 母对母 10cm长 (Male to Male 10cm long)
- Quantity Selection:** A red arrow points to the quantity selector set to 1.

## 8.3.4. System Construction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

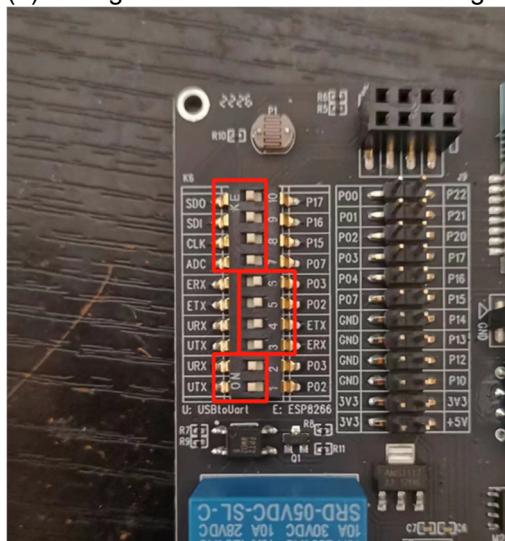
- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### 1. Build a development environment

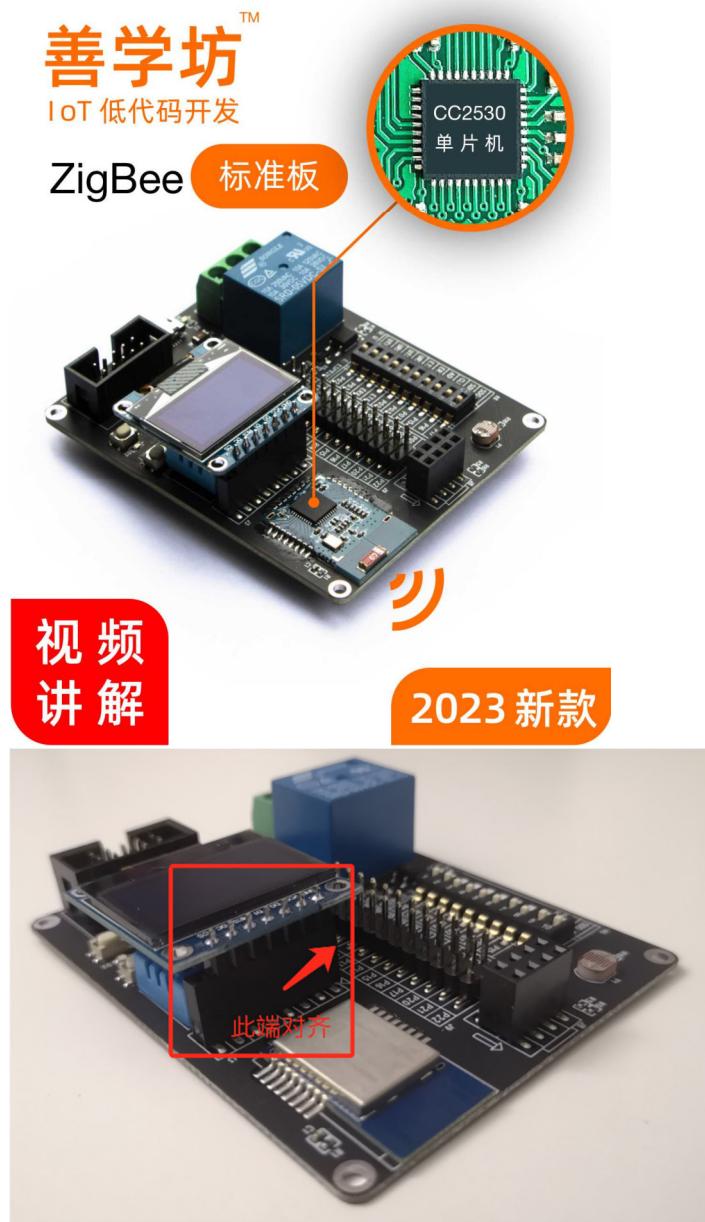
Install the development software according to Part 1: Preparation of this tutorial.

### 2. Configure ZigBee 3.0 standard board

- (1) Configure the DIP switches in the ZigBee 3.0 standard board as shown in the figure.



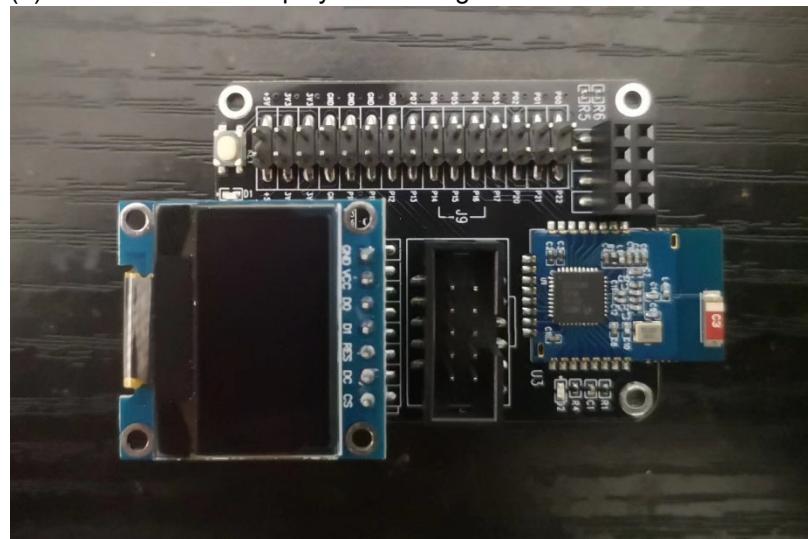
(2) Insert the OLED display into the ZigBee 3.0 standard board, as shown in the figure.



(3) If you need to deploy more acquisition nodes, you can configure more ZigBee 3.0 standard boards according to the above steps.

### 3. Configure ZigBee 3.0 Mini Board

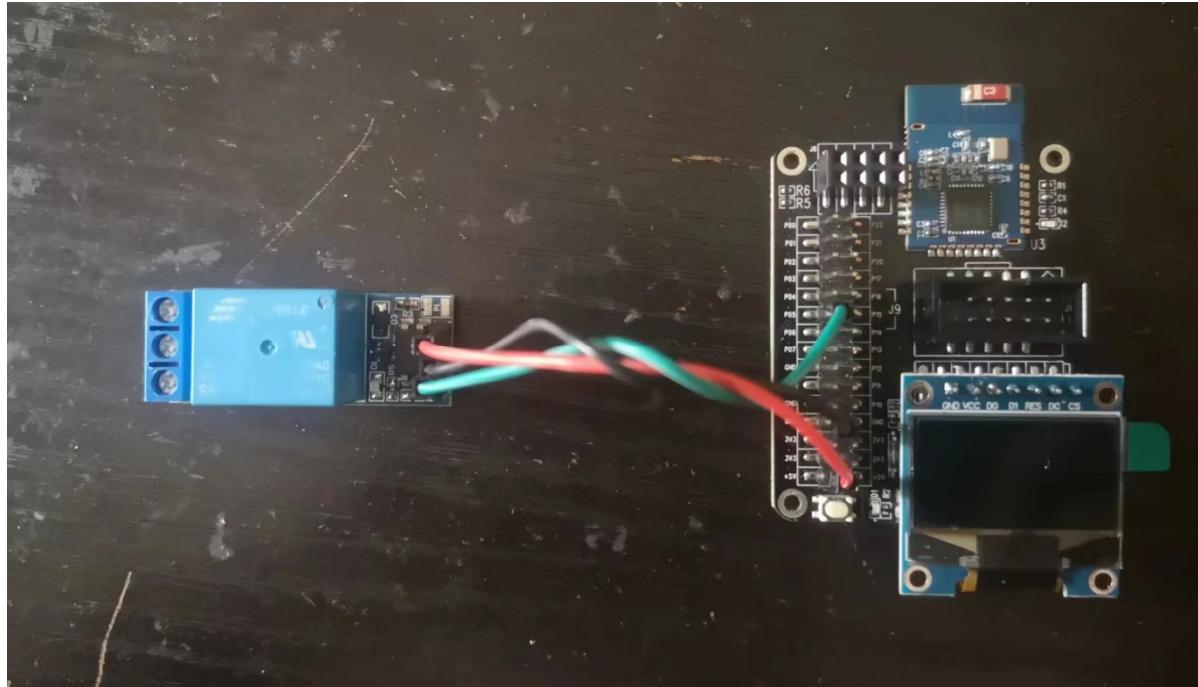
(1) Insert the OLED display into the ZigBee 3.0 Min board as shown in the figure.



(2) Use Dupont wire to connect the relay to the ZigBee 3.0 Mini board. The wiring instructions are as follows:

- (2A) Connect the VCC of the relay to **the 5v** of the development board.
- (2B) Connect the GND of the relay to the GND of the development board.
- **(2C) The control interface CH1 of the relay is connected to P05 of the development board.**

(3) After the wires are connected, it will be as shown in the figure.



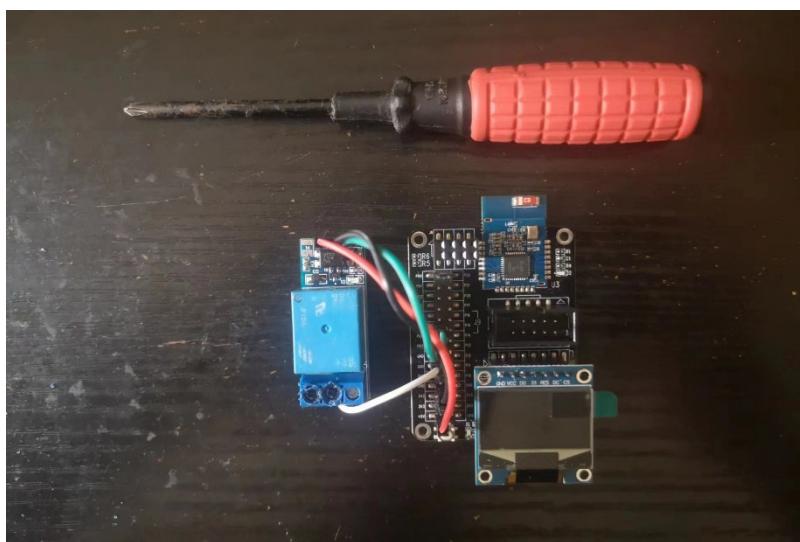
(4) Use Dupont wire to connect the buzzer to the ZigBee 3.0 Mini board and hang it on the relay. The wiring instructions are as follows:

- (4A) Split a DuPont wire into two halves and expose the copper wire as shown in the figure.

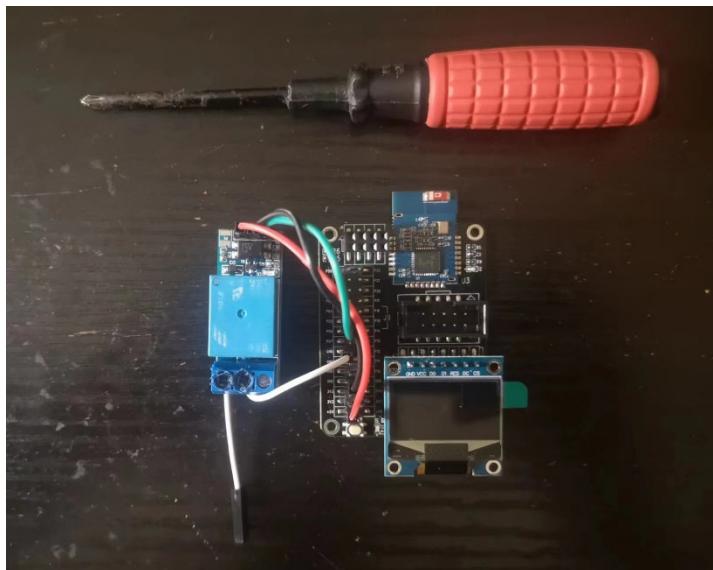


- (4B) Use a screwdriver to unscrew the port in the middle of the relay and connect a copper wire of the DuPont cable.

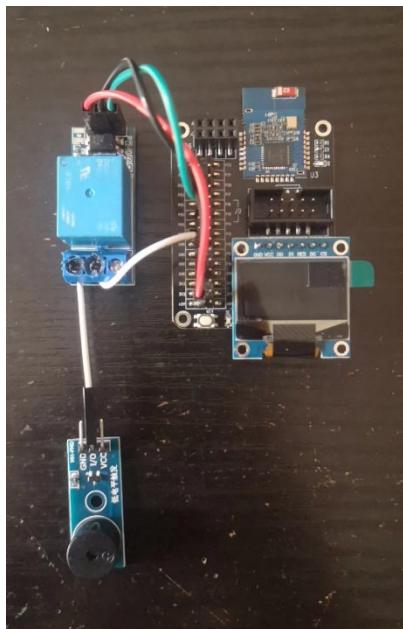
Then insert the other end of the DuPont cable into the GND interface of the development board, as shown in the figure.



- (4B) Connect another DuPont wire to the **normally open** port of the relay, as shown in the figure.

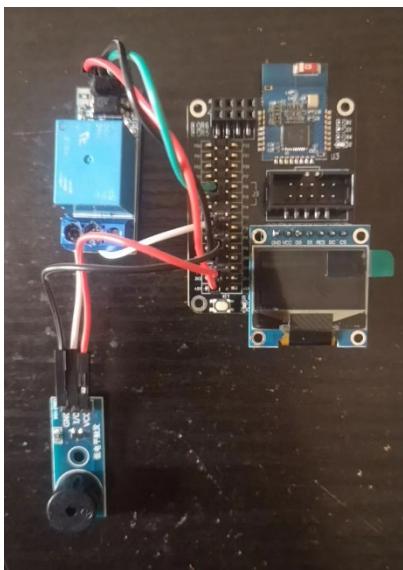


- (4C) Connect the white wire in the figure to the I/O interface of the buzzer, as shown in the figure.



- (4D) Use another Dupont line to connect the buzzer's VCC and GND to the 3v3 and GND interfaces of the development

board respectively, as shown in the figure.

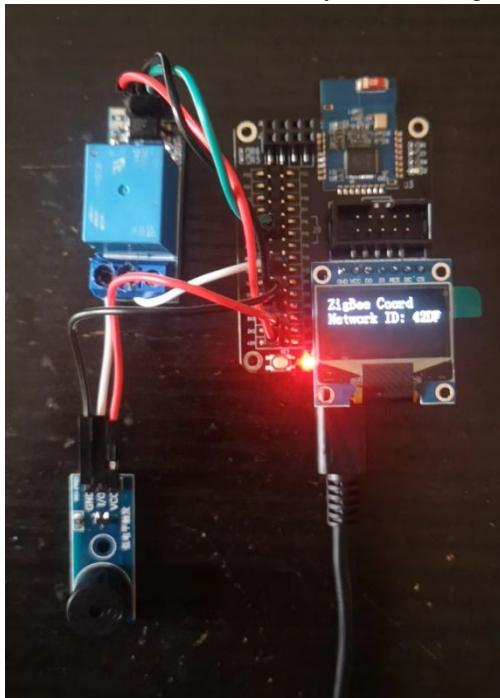


## 4. Burn the firmware

(1) After downloading the source code of this project, find the firmware, as shown in the figure.

文件 (D:) > 学习资料 > ZigBee > ZigBee 温湿度监测 & 声光报警系统 (1) > 固件			
名称	修改日期	类型	大小
节点 (烧录到ZB标准板).hex	2023/4/4 14:15	HEX 文件	600 KB
协调器 (烧录到ZB Mini板.hex)	2023/4/4 14:19	HEX 文件	610 KB

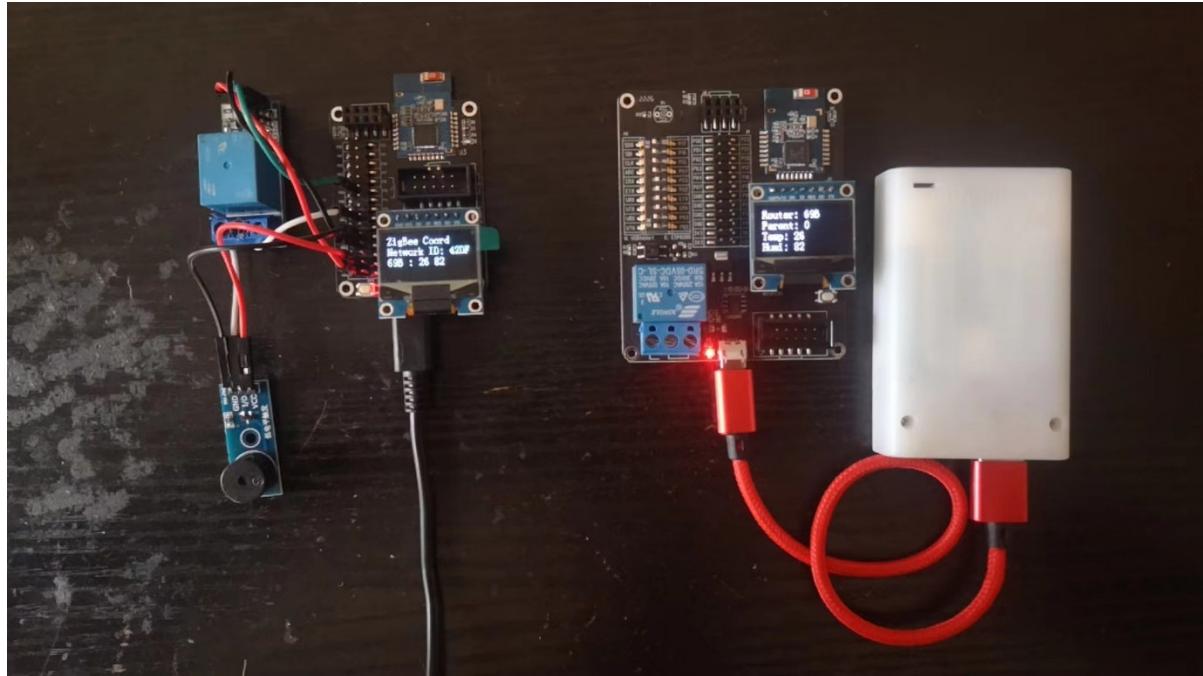
(2) Burn the coordinator firmware into the ZigBee 3.0 Mini board. After burning, **use Micro USB to power it**. The ZigBee 3.0 Mini board will automatically create a ZigBee network. After the creation is complete, the display will be as shown in the figure.



(3) Burn the node firmware into the ZigBee 3.0 standard board.

**Tip:** For detailed operation steps, please refer to "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: Basic Experiment of CC2530 Development" → "Firmware Burning".

(4) The ZigBee 3.0 standard board will automatically collect data and send it to the Mini board, as shown in the figure.

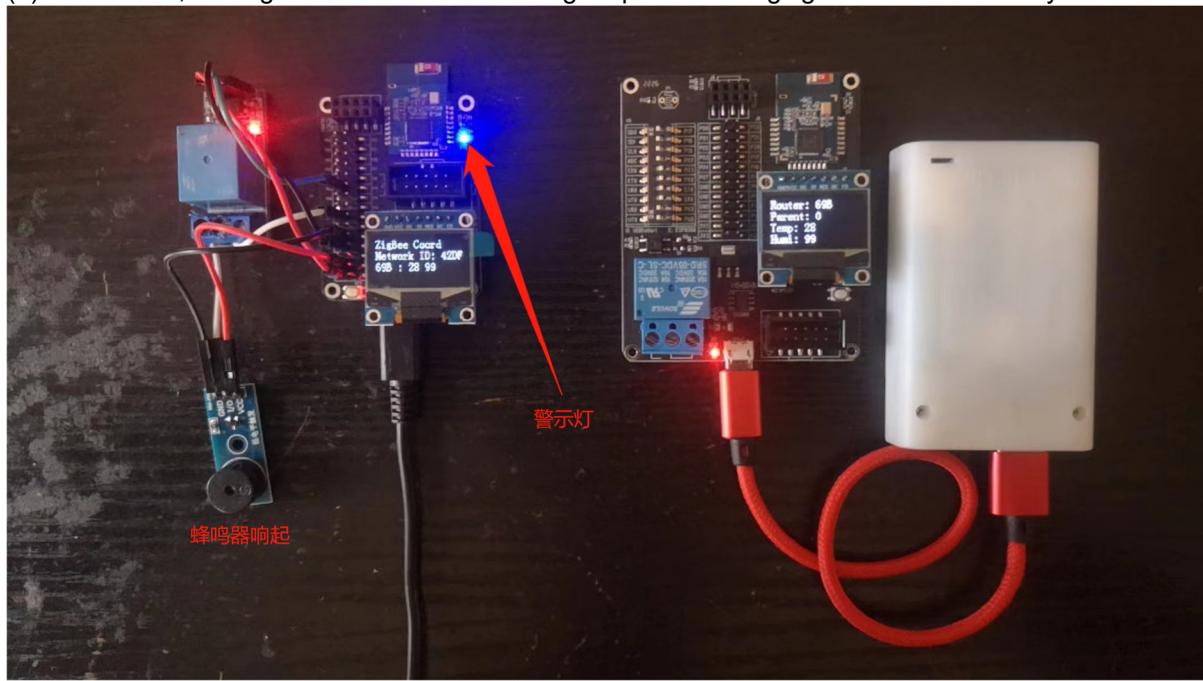


(5) If you need to deploy more acquisition nodes, you can use the same method to burn the firmware to more ZigBee 3.0 standard boards.

## 5. System Testing

(1) You can blow on the temperature and humidity sensor of one of the ZigBee 3.0 standard boards to make the ambient humidity value exceed 90%.

(2) At this time, the ZigBee 3.0 Mini board will light up the warning light and start the relay



(3) When all abnormal nodes are restored, the alarm status will be automatically released.

### 8.3.5. Source code description

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions!
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

# Source code description

The source code structure of this project is basically the same as the following two projects. Please refer to the source code descriptions:

- ZigBee 3.0 Environmental Information Collection (1)
- ZigBee 3.0 Environmental Information Collection (2)

## Note:

- (1) After learning parts 1 to 4, you will understand the code of this project.  
(2) If you have not learned parts 1 to 4, you must stop reading the source code!  
(3) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.

## 8.4. ZigBee automatic light on - based on light intensity & human body monitoring (1)

### 1. Required Reading

This course is an extracurricular course, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise please stop studying!

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530
- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

### 2. Contents

- 8.4.1. System Introduction
- 8.4.2. Source code download
- 8.4.3. Hardware List
- 8.4.4. System Construction
- 8.4.5. Source code description

### 8.4.1. System Introduction

This part is extracurricular content, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.  
(2) If you have not learned parts 1 to 4, you must stop reading!  
(3) In principle, this part does not have any technical support. Please understand!

### System Introduction

(1) This system can collect ambient light illumination (photoresistor ADC value), detect the presence of human body, and report to the ZigBee coordinator;

- (2) When the light illumination is low and there is human body, the light will be automatically turned on (controlled by relay);  
(3) It supports the deployment of one coordinator and **multiple information collection nodes**.

## 8.4.2. Source code download

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.  
(2) If you have not learned parts 1 to 4, you must stop reading!  
(3) In principle, this part does not have any technical support. Please understand!

### Source code download

(1) Users who have purchased the product can download the additional materials from the "Supplementary Materials Download" section in the catalog.

(2) This item is located in the additional materials as shown in the figure.

文件名	修改时间	大小
其他项目 1	2022-06-29 10:59	-
其他项目 2	2022-06-29 10:59	-
ZigBee 自动开灯——基于光亮度&人体监测 (1).zip	2023-05-06 21:38	56.4M
ZigBee 温湿度监测 & 声光报...		
ZigBee 3.0 环境信息采集 (2).zip	2023-03-31 12:53	56.1M
ZigBee 3.0 环境信息采集 (1) (多节点版).zip	2023-03-30 16:28	281.5M
ZigBee 3.0 环境信息采集 (1).zip	2023-03-30 16:28	64.6M
基于Zigbee的温湿度 & 光照度采集系统.zip	2022-06-29 10:59	33.6M
基于ZigBee的文件传送系统.zip	2022-06-29 10:59	27.1M
基于ZigBee的光照自动开关窗帘.zip	2022-06-29 10:59	38.8M
基于ZigBee的农业环境信息采集.zip	2022-06-29 10:59	69.8M

## 8.4.3. Hardware List

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.  
(2) If you have not learned parts 1 to 4, you must stop reading!  
(3) In principle, this part does not have any technical support. Please understand!

### Hardware List

(1) ZigBee 3.0 standard board × 1 ( or more ): <https://item.taobao.com/item.htm?id=683089996879>

(2) ZigBee 3.0 Mini board × 1: <https://item.taobao.com/item.htm?id=683089996879>

(3) 0.96 OLED12864 display × 2 (or more, the same number as the development board): <https://item.taobao.com/item.htm?id=683089996879>

(4) SmartRF04EB emulator × 1: <https://item.taobao.com/item.htm?id=683089996879>

(5) Micro USB data cable × 2 (or more, the same number as the development board): Free when purchasing the development board

(6) Dry cell battery box × 2 (or more, the same quantity as the standard version): <https://item.taobao.com/item.htm?id=683115420758>

(7) Relay & Dupont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

(8) Human presence sensor & DuPont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

(8) LED light & Dupont cable × 1: Can be omitted or purchased separately from Taobao.

## Buying Guide

**Tip:** You can purchase all the hardware required for this project by purchasing as follows.

(1) ZigBee development kit version selection is shown in the figure.



(2) Select the battery box as shown in the figure.



(3) Relay & DuPont line selection is shown in the figure.



(4) Human presence sensor & DuPont line selection are shown in the figure.



## 8.4.4. System Construction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

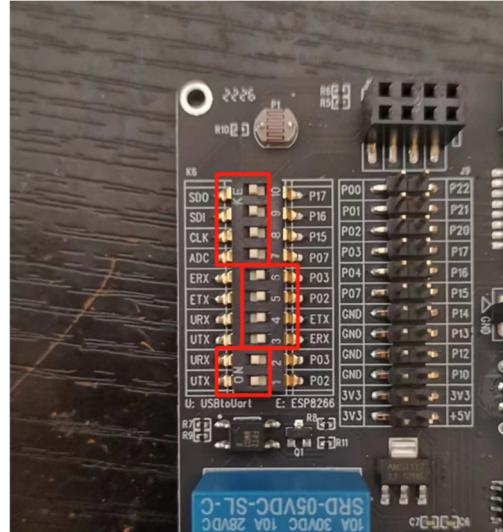
- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### 1. Build a development environment

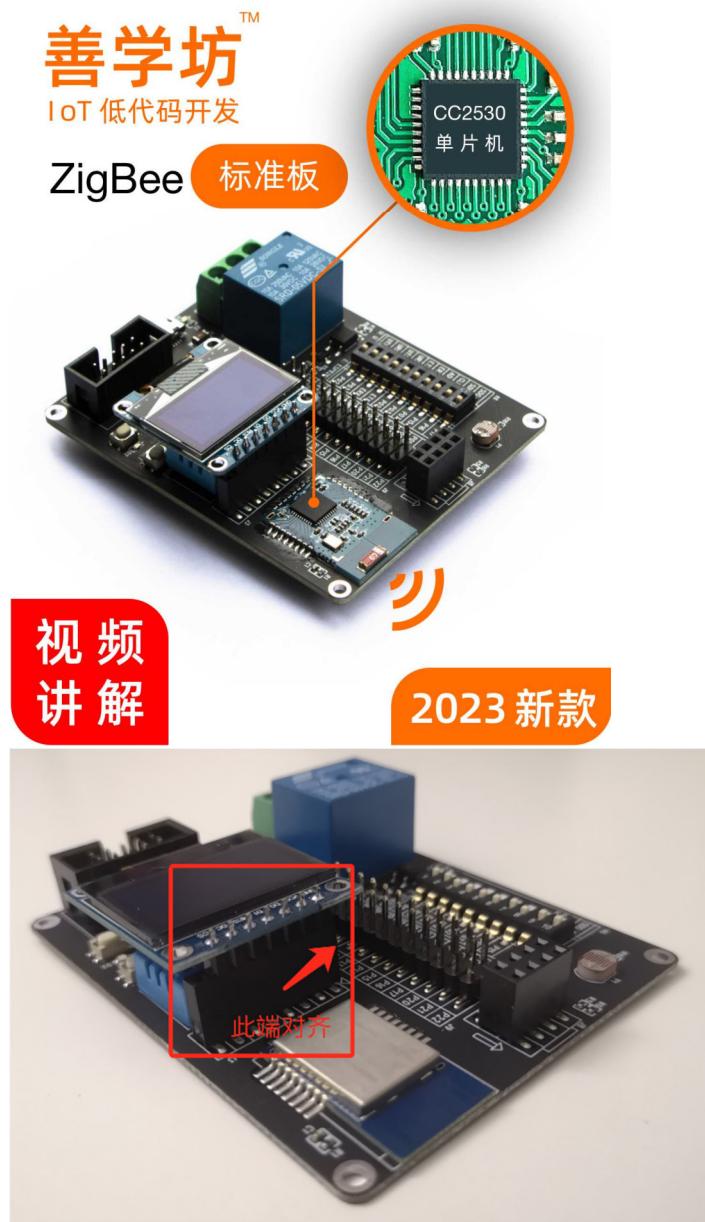
Install the development software according to Part 1: Preparation of this tutorial.

### 2. Configure ZigBee 3.0 standard board

- (1) Configure the DIP switches in the ZigBee 3.0 standard board as shown in the figure.



(2) Insert the OLED display into the ZigBee 3.0 standard board, as shown in the figure.

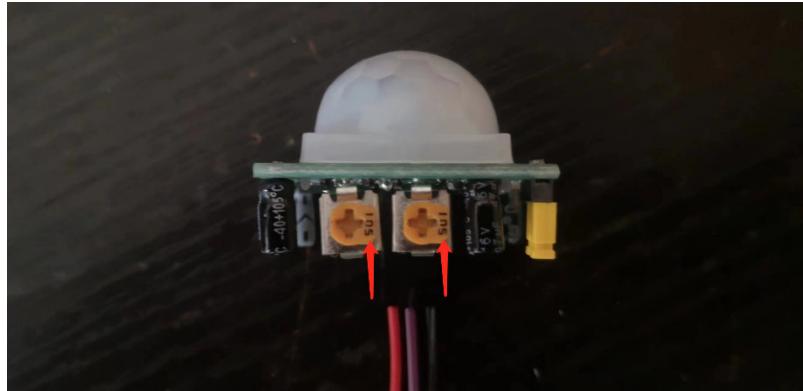


(3) Use DuPont wire to connect the human infrared sensor to the standard board. The wiring method is as follows:

- VCC is connected to the 3v3 interface of the development board;
- GND is connected to the GND interface of the development board;
- The output is connected to P00 of the development board;



(4) Turn the sensor knob to align to the right, as shown in the figure.



#### Tips:

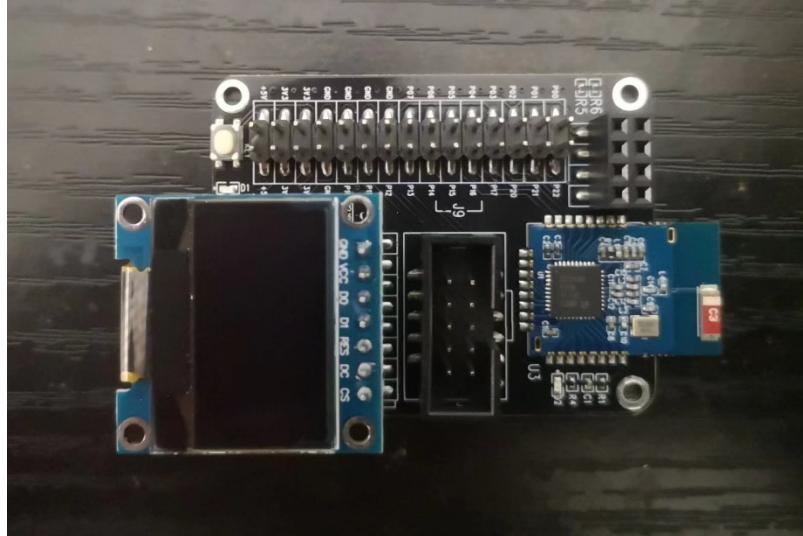
(1) Rotating to the above position is for testing purposes. Readers can rotate to other positions according to different requirements.

(2) For detailed functions of the knob, please refer to: <https://zhuanlan.zhihu.com/p/483565939>

(4) If you need to deploy more acquisition nodes, you can configure more ZigBee 3.0 standard boards according to the above steps.

## 3. Configure ZigBee 3.0 Mini Board

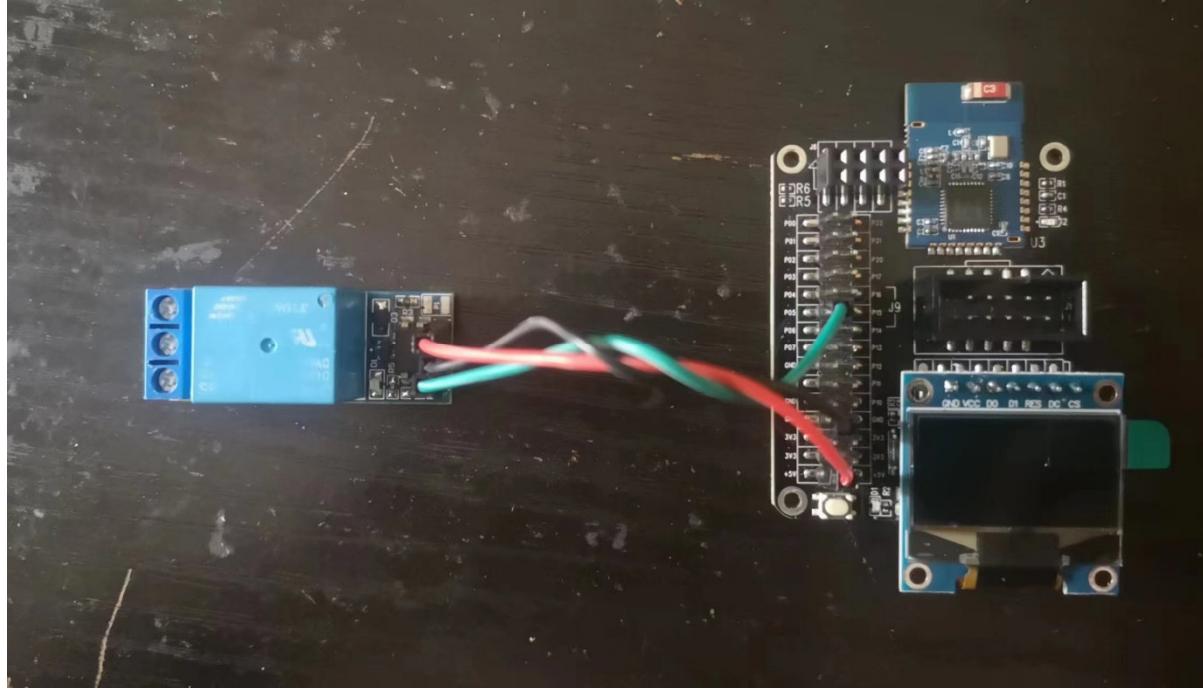
(1) Insert the OLED display into the ZigBee 3.0 Min board as shown in the figure.



(2) Use Dupont wire to connect the relay to the ZigBee 3.0 Mini board. The wiring instructions are as follows:

- (2A) Connect the VCC of the relay to **the 5v** of the development board.
- (2B) Connect the GND of the relay to the GND of the development board.
- **(2C) The control interface CH1 of the relay is connected to P05 of the development board.**

(3) After the wires are connected, it will be as shown in the figure.



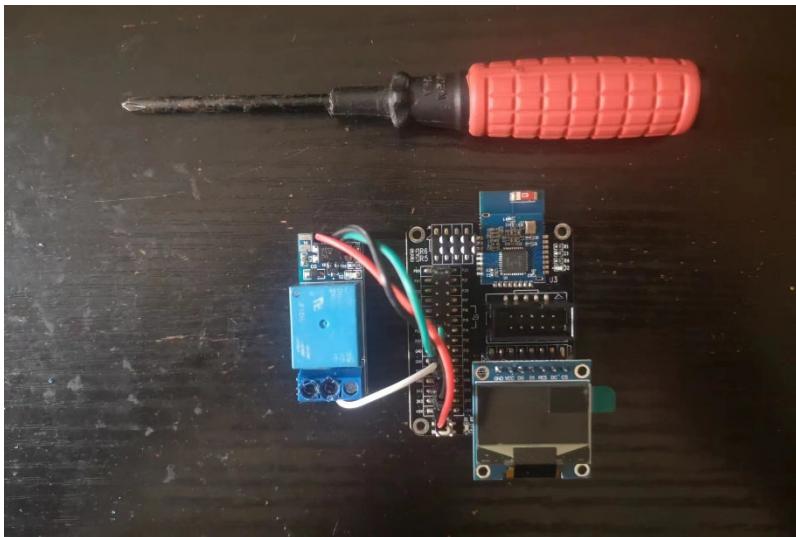
(4) Use Dupont wire to connect the buzzer to the ZigBee 3.0 Mini board and hang it on the relay. The wiring instructions are as follows:

- (4A) Split a DuPont wire into two halves and expose the copper wire as shown in the figure.

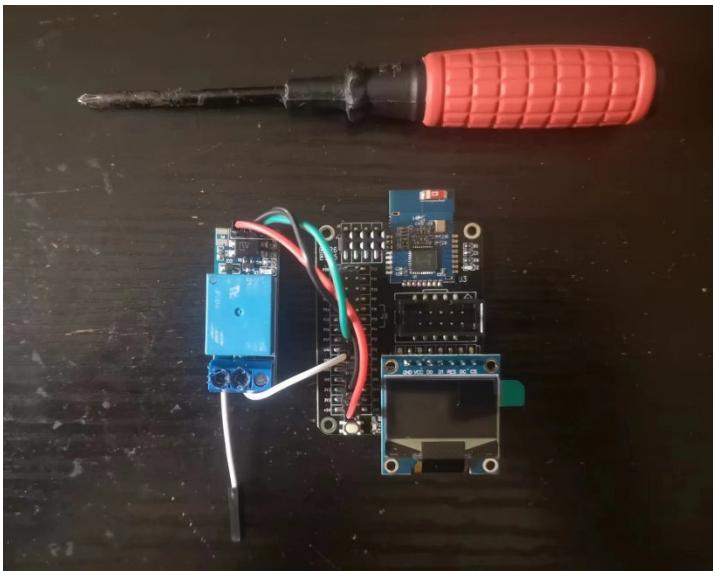


- (4B) Use a screwdriver to unscrew the port in the middle of the relay and connect a copper wire of the DuPont cable.

Then insert the other end of the DuPont cable into the GND interface of the development board, as shown in the figure.



- (4B) Connect another DuPont wire to the **normally open** port of the relay, as shown in the figure.



## 4. Burn the firmware

(1) After downloading the source code of this project, find the firmware, as shown in the figure.

此电脑 > 文件 (D:) > 学习资料 > ZigBee > ZigBee 自动开灯——基于光照度&人体监测 (1) > 固件			
名称	修改日期	类型	大小
节点 (烧录到ZB标准板).hex	2023/5/6 21:31	HEX文件	605 KB
协调器 (烧录到ZB迷你板).hex	2023/5/6 21:35	HEX文件	613 KB

(2) Burn the coordinator firmware into the ZigBee 3.0 Mini board. After burning, **use Micro USB to power it**. The ZigBee 3.0 Mini board will automatically create a ZigBee network. After the creation is complete, the display will be as shown in the figure.



(3) Burn the node firmware into the ZigBee 3.0 standard board.

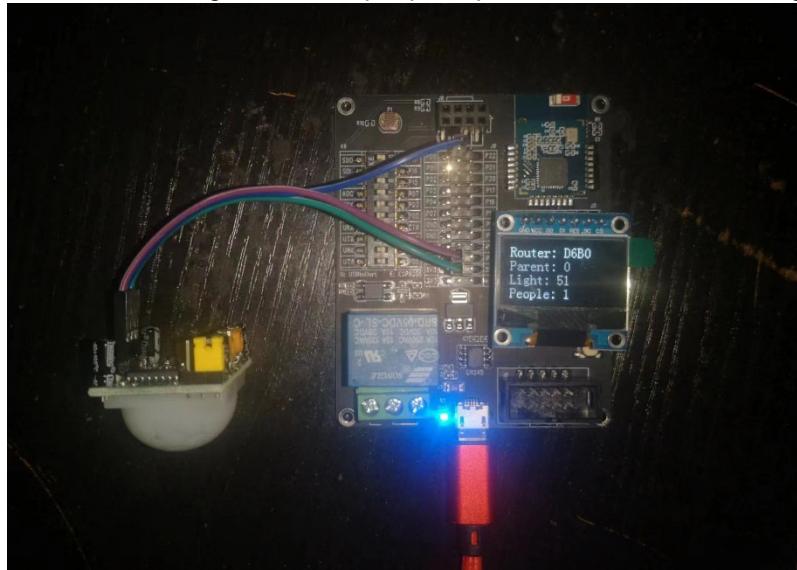
**Tip:** For detailed operation steps, please refer to "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: Basic Experiment of CC2530 Development" → "Firmware Burning".

(4) The ZigBee 3.0 standard board will automatically collect data and send it to the Mini board.

(5) If you need to deploy more acquisition nodes, you can use the same method to burn the firmware to more ZigBee 3.0 standard boards.

## 5. System Testing

(1) You can press the photoresistor of one of the ZigBee 3.0 standard boards and shake your hand next to the human infrared sensor to make light < 70 and people equal to 1, as shown in the figure.



**Note:** The sensitivity of the human body sensor can be adjusted. For adjustment methods, refer to the official website "Smart Hardware" → "Test Sensor" → "Human Infrared Sensor": <https://zhuanlan.zhihu.com/p/483565939?>

(2) At this point, the ZigBee 3.0 Mini board will light up and activate the relay.

(3) When all nodes are restored, the lights will automatically turn off and the relays will automatically turn off.

## 8.4.5. Source code description

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Source code description

The source code structure of this project is basically the same as the following two projects. Please refer to the source code descriptions:

- ZigBee 3.0 Environmental Information Collection (1)
- ZigBee 3.0 Environmental Information Collection (2)

**Note:**

- (1) After learning parts 1 to 4, you will understand the code of this project.
- (2) If you have not learned parts 1 to 4, you must stop reading the source code!
- (3) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.

## 8.5. File transmission system based on ZigBee

This project is an **extracurricular learning content**. Its content is **relatively complex** and **has a certain degree of difficulty**. Therefore:

- (1) **If you have not learned** the previous chapters, **please stop studying**.
- (2) **In principle**, provide technical support for this project. Please understand.

### 1. Field of focus

This solution is aimed at course design or function verification demo areas that have general requirements for communication stability, response speed and communication distance.

## 2. System Architecture Description and Function Demonstration

This system uses ZigBee to achieve the transmission of large files between two nodes.

- Architecture Description: <https://zhuanlan.zhihu.com/p/394985077>
- Functional demonstration: <https://www.bilibili.com/video/BV1Yb4y1y7g7/> (the development board in the video is the old version, and a new version has been released, which is compatible with the old version)



## 3. Download supporting resources

The supporting resources of this project are additional parts and are not yet publicly available on the Internet. They are only provided free of charge to friends who have purchased the development kit. Go to "Supporting Resource Download" in this course: <https://z7po9bxpe4.k.tophink.com/@zigbee-dev-guide/yuandaimaxiaozai.html>

- Resource screenshots



## 4. Hardware required for the system

- Shanxuefang ZigBee 3.0 Development Kit

## 产品清单



ZigBee 标准板



WiFi模块



USB线×2 (赠送)



ZigBee Mini板



仿真器

- 仿真线×1
- Mini5P线×1

## 善学坊

IoT 低代码开发

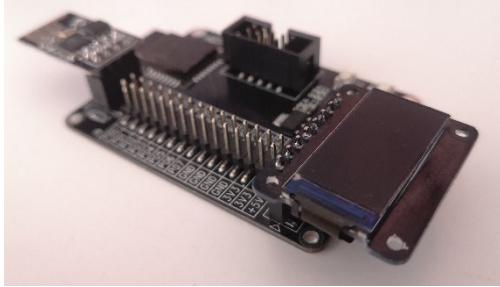
\*以上为推荐搭配清单，不代表实际清单

## 5. Build your own system

(1) Run the exe executable program on the PC.

(2) Connect the ZigBee device (a new version of the development board is used here to demonstrate the hardware connection)

- Insert the 0.96 TFT display into the ZigBee Mini board as shown in the picture



- If you use a standard board, insert the 0.96 TFT display in a similar way.

The 0.96 OLED display and TFT display that come with the new development board are no longer available. You need to switch the screen type first. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

(3) Find the corresponding source code or firmware in the supporting resources, compile and burn the corresponding firmware.

(4) Follow the instructions in the demonstration video to connect the two development boards to the computer.

(5) After ensuring that all steps are executed correctly, you can transfer files between two ZigBee nodes. For specific results, please refer to the "System Demonstration Effect" section in this article.

## 8.6. ZigBee-based automatic curtain opening and closing

This project is **an extracurricular learning content**, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this project **in principle**. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

## 1. Field of focus

This solution is aimed at **course design** or **function verification demo** areas that have general requirements for communication stability, response speed and communication distance.

## 2. System Architecture Description and Function Demonstration

This system can collect information such as ambient light intensity and automatically control the curtain switch

- Architecture Description: <https://zhuanlan.zhihu.com/p/407690248>
- Functional demonstration: <https://www.bilibili.com/video/BV1oq4y1d78s/>



## 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not yet publicly available on the Internet. They are only provided free of charge to friends who have purchased **the development kit**. Go to "Supporting Resource Download" in this course: <https://z7po9bxpe4.k.topthink.com/@zigbee-dev-guide/yuandaimaxiazai.html>

- Resource screenshots



## 4. Hardware required for the system

- Shaxuefang ZigBee 3.0 Development Kit



## 善学坊

IoT 低代码开发

\*以上为推荐搭配清单，不代表实际清单

- 2 buttons: ordinary buttons are enough, refer to the Taobao link: <https://item.taobao.com/item.htm?id=627735113474>



- Stepper motor: ordinary stepper motor is fine, refer to Taobao link: <https://m.tb.cn/h.fVCkegZ?sm=d24e14>
- Battery box (optional): <https://item.taobao.com/item.htm?id=662015967667>

## 5. ZigBee device link and firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

(1) Since this project requires the light sensor and P03 pin on the standard board, you need to turn the dip switches **to the ADC side and the P03 side respectively**.



(3) Connect the button to the standard board. Wiring method:

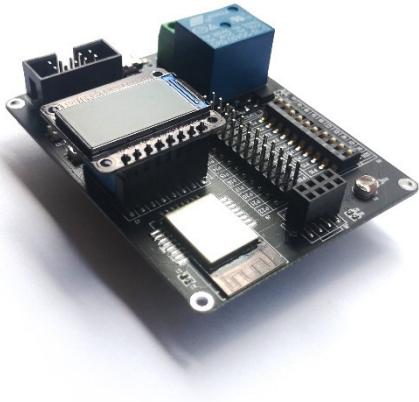
- (Black) Touch button SIG pin -> Standard board P03
- (Blue) Touch button SIG pin -> Standard board P04
- VCC pin -> Standard board 3v3
- GND pin -> Standard board GND

The OUT pin is the signal output pin.

(3) Connect the stepper motor to the ZigBee Mini board. Wiring method:

- Stepper motor IN1 pin -> MiNi board P13
- Stepper motor IN2 pin -> MiNi board P14
- Stepper motor IN3 pin -> MiNi board P15
- Stepper motor IN4 pin -> MiNi board P16
- Stepper Motor + Pin -> MiNi Board +5V
- Stepper Motor - Pin -> MiNi Board GND

(4) Insert the 0.96 TFT display into the ZigBee standard board, as shown in the figure



TFT screens are no longer available. You can switch to OLED screens. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

(5) Find the corresponding firmware in the supporting resources, and burn the gateway firmware to the Mini board and the terminal firmware to the standard board.

(7) After burning the firmware, power on each development board via the USB cable.

(8) After ensuring that all steps are performed correctly, you can use the light intensity to automatically or manually control the curtain opening and closing. For specific effects, please refer to the "System Demonstration Effect" section in this article.

## 6. Source code explanation

- B station video: <https://www.bilibili.com/video/BV1oq4y1d78s?p=2>

## 8.7. ZigBee-based temperature, humidity & light intensity acquisition system

This project is an **extracurricular learning content**. Its content is relatively complex and has a certain degree of difficulty.

Therefore:

(1) If you have not learned the previous chapters, please stop studying.

(2) In principle, provide technical support for this project. Please understand.

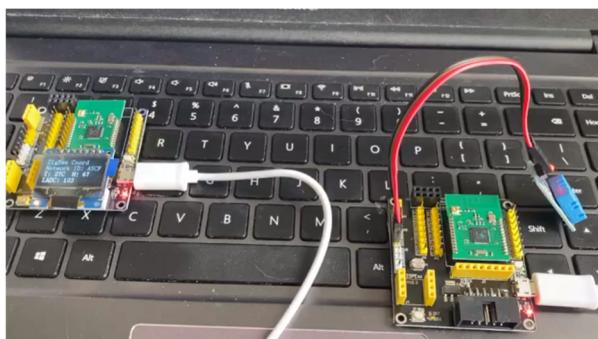
## 1. Field of focus

This solution is aimed at **course design** or **functional verification demo** fields that have general requirements for communication stability, response speed and communication distance.

## 2. System Architecture Description and Function Demonstration

The system can detect ambient temperature, humidity & light intensity, and support displaying data on the PC

- Functional demonstration: <https://www.bilibili.com/video/BV19U4y1A7va/> (the development board in the video is the old version, and a new version has been released, which is compatible with the old version)



## 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not yet publicly available on the Internet. They are only provided free of charge to friends who have purchased **the development kit**. Go to "Supporting Resource Download" in this course: <https://z7po9bxpe4.k.tophink.com/@zigbee-dev-guide/yuandaimaxiaza.html>

- Resource screenshots



## 4. Hardware required for the system

- ZigBee MiNi board × 1
- ZigBee standard board × 1
- 0.96 TFT display

TFT screens are no longer available. You can switch to OLED screens. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

## 5. Build your own system

## 1. PC program

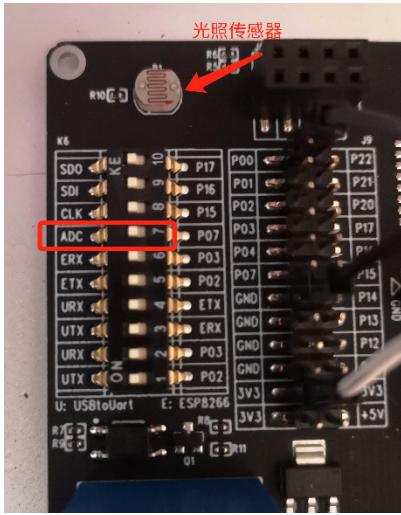
- Find and run the PC-side exe program in the supporting resources.

## 2. ZigBee device connection firmware burning

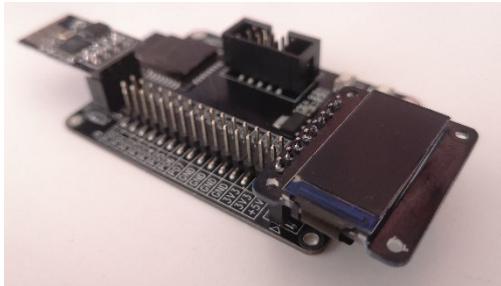
This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

(1) Connect the hardware according to the steps (the new version of the development board is used here to demonstrate the hardware connection)

- Since this project requires the light sensor on the standard board, the dip switch needs to be turned to the ADC side, as shown in the figure



- Insert the 0.96 TFT display into the ZigBee Mini board as shown in the picture



TFT screens are no longer available. You can switch to OLED screens. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

(3) Find the corresponding firmware in the supporting resources and burn the gateway firmware to the Mini board and the terminal firmware to the standard board.

(4) After burning the firmware, power on the development board and connect it.

(5) After ensuring that all steps are performed correctly, you can check the temperature, humidity, light intensity and other information.

## 6. Source code explanation

- B station video: <https://www.bilibili.com/video/BV19U4y1A7va?p=2>

# 8.8. Agricultural environmental information collection based on ZigBee

## 1. Required Reading

This course is an extracurricular course, and its content is relatively complex and has a certain degree of difficulty. Therefore: you need to complete the following courses first, otherwise **please stop studying!**

- Part 1: Preparation
- Part 2: Introduction to 51 MCU - Based on CC2530
- Part 3: Detailed Explanation of Z-Stack 3.0
- Part 4: Z-Stack 3.0 Network Programming

## 2. Contents

- 8.8.1. System Introduction
- 8.8.2. Source code download
- 8.8.3. Hardware List
- 8.8.4. Using a private cloud server
- 8.8.5. Configuring the Development Board
- 8.8.6. Networking and WiFi Connection
- 8.8.7. Using Android APP
- 8.8.8. Source code description

### 8.8.1. System Introduction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

#### System Introduction

- (1) This system can collect ambient temperature, humidity, light intensity and soil moisture, and upload them to a private cloud server and a self-developed Android APP.

(2) The system architecture is shown in the figure.



(3) System Demonstration

System function demonstration: <https://www.bilibili.com/video/BV1mf4y1A7wB>

**Tip:** The development board in the video is an old version. A new version has been released and is compatible with the old version.



## 8.8.2. Source code download

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

# Source code download

(1) Users who have purchased the product can download the additional materials from the "Supplementary Materials Download" section in the catalog.

(2) This item is located in the additional materials as shown in the figure.

全部文件 > ... > ZigBee > 《ZigBee3.0开发指南》配套资源（附加部分） > 考外篇：项目实战		
文件名	大小	操作
其他项目 1	-	
其他项目 2	-	
ZigBee 3.0 环境信息采集 (1) .zip	64.58 MB	
ZigBee 3.0 环境信息采集 (1) (多节点版) .zip	281.51 MB	
ZigBee 3.0 环境信息采集 (2) .zip	56.09 MB	
基于ZigBee的光照自动开关窗帘.zip	38.80 MB	
基于ZigBee的农业环境信息采集.zip	69.76 MB	
基于Zigbee的温湿度 & 光照度采集系统.zip	33.58 MB	
基于ZigBee的温湿度 & 信号强度探测系统.zip	33.51 MB	
基于ZigBee的文件传送系统.zip	27.13 MB	

## 8.8.3. Hardware List

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

## Hardware List

(1) ZigBee 3.0 standard board × 1: <https://item.taobao.com/item.htm?id=683089996879>

(2) ZigBee 3.0 Mini board × 1: <https://item.taobao.com/item.htm?id=683089996879>

(3) 0.96 OLED12864 display × 2

(4) SmartRF04EB emulator × 1: <https://item.taobao.com/item.htm?id=683089996879>

(5) Micro USB data cable × 2: Free when purchasing the development board

(6) Dry cell battery box × 2 (optional): <https://item.taobao.com/item.htm?id=683115420758>

(7) ESP8266 WiFi module × 1: <https://item.taobao.com/item.htm?id=683089996879>



善学坊™

(8) Soil moisture sensor & DuPont cable × 1: <https://item.taobao.com/item.htm?id=683757286045>

# Buying Guide

**Tip:** You can purchase all the hardware required for this project by purchasing as follows.

(1) ZigBee development kit version selection is shown in the figure.



**Note:** The above version already includes a WiFi module.

(2) Select the battery box as shown in the figure.



(3) Soil moisture sensor & Dupont line version selection is shown in the figure.

举报

土壤湿度传感器

TRSD-A1

价格 ¥9.80 100+  
月销量

优惠 淘金币可抵0.29元

店铺优惠券 30元店铺优惠券，满1500元可用 领取

店铺优惠券 20元店铺优惠券，满1000元可用 领取

广东广州至 全国 快递 免运费 现货，付款后24小时内发货

杜邦线 母对母 10cm长 无

颜色分类

数量 1 件

立即购买 加入购物车

收藏宝贝 (54人气)

## 8.8.4. Using a private cloud server

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### 1. Download and install Xshello 7

<https://zhuanlan.zhihu.com/p/486650064>

### 2. Log in to the cloud server

(1) Follow the XShell 7 tutorial instructions to log in to the server via SSH. The login information is as follows:

- IP address: **1.15.27.206**
- Port Number: 22
- Account: guest
- Password: 12345678

Note:

(1) If this account & password fails to log in, you can use the following information to log in:

- Username: **lucker**
- Password: **12345678**

- (2) If you still fail to log in, please report it to the community: <https://bbs.csdn.net/forums/zigbee>  
(3) Login failure may occur due to reasons such as private servers being subject to national security controls and some users not being able to operate in accordance with regulations.

(2) After successfully logging into the server, the following figure is shown:

```
ssh://guest:*****@1.15.119.13:22
要添加当前会话，点击左侧的箭头按钮。
1 TencentCloudGuest + [x]
Connecting to 1.15.119.13:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-118-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Tue Apr 20 18:01:36 CST 2021

 System load: 0.0          Processes:      90
 Usage of /: 12.9% of 49.15GB Users logged in:    0
 Memory usage: 27%          IP address for eth0: 172.17.0.2
 Swap usage:  0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.
   https://microk8s.io/high-availability

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue Apr 20 16:41:22 2021 from 120.235.227.52
guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$
```

Note: After logging into the server, it is prohibited to delete or modify any files, otherwise you will be responsible for the relevant security!

## 2. Enable cloud services

(1) Enter the cd /home/guest/workspace/iot\_master command to switch to the iot\_master directory, as shown in the figure.

```
guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$ guest@VM-0-2-ubuntu:~$ cd /home/guest/workspace/iot_master
```

(2) Enter the following command to start the MQTT service.

```
./3rdparty/mosquitto-1.6.12/mosquitto &
```

```
guest@VM-0-2-ubuntu:~/workspace/iot_master$ guest@VM-0-2-ubuntu:~/workspace/iot_master$ ./3rdparty/mosquitto-1.6.12/mosquitto &
ssh://guest@1.15.119.13:22
```

(3) Enter the following command to start the user service.

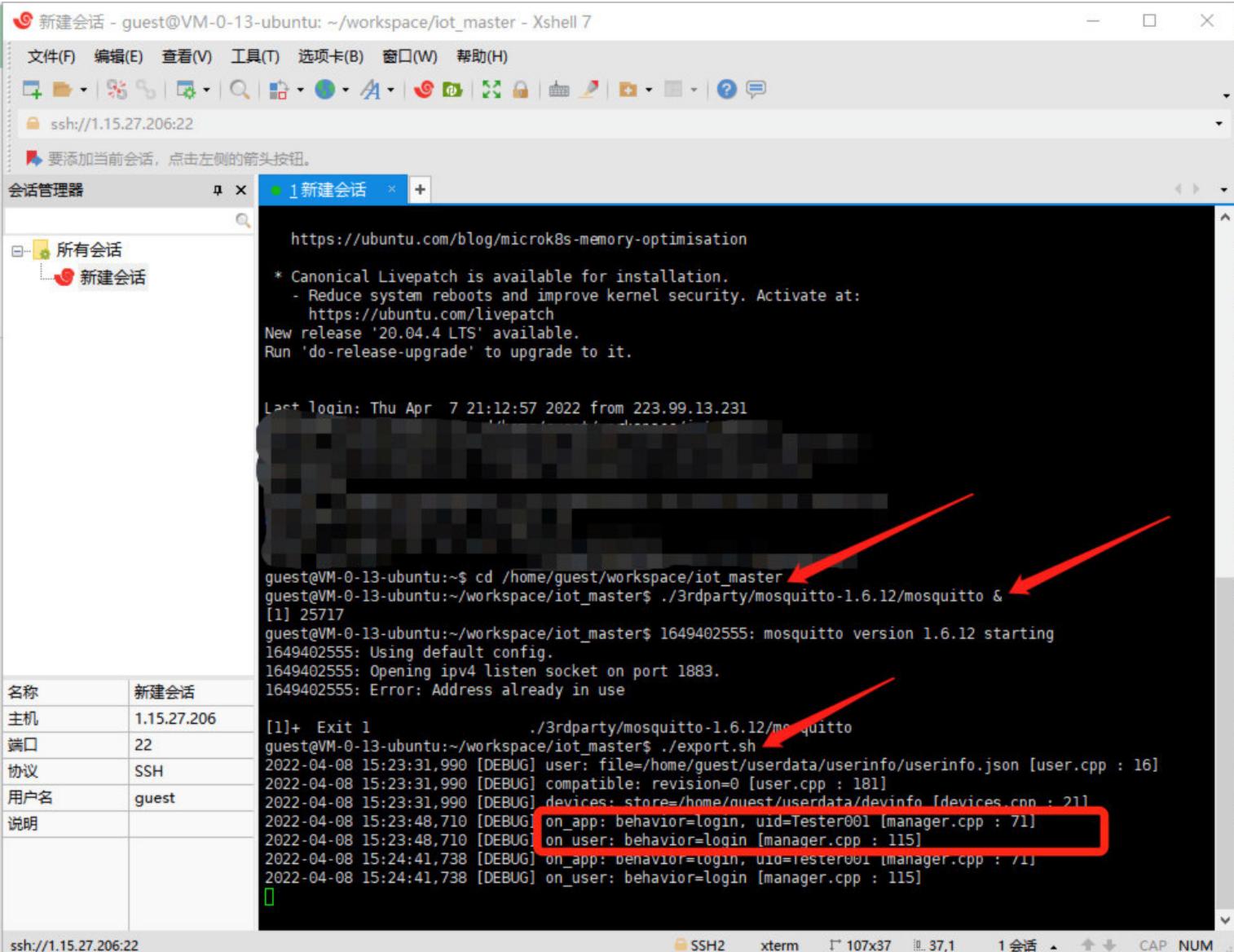
```
./export.sh
```

```
guest@VM-0-2-ubuntu:~/workspace/iot_master$ guest@VM-0-2-ubuntu:~/workspace/iot_master$ ./export.sh
```

ssh://guest@1.15.119.13:22

#### (4) Verify whether the service is successfully started.

In the subsequent steps, when the development board reports data or the APP user logs in, you can see the response log printed out. As shown in the red box in the figure, it is the log information of the successful login of the App user.



```
ssh://guest@1.15.27.206:22
* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch
New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr  7 21:12:57 2022 from 223.99.13.231
[1] 25717
guest@VM-0-13-ubuntu:~$ cd /home/guest/workspace/iot_master
guest@VM-0-13-ubuntu:~/workspace/iot_master$ ./3rdparty/mosquitto-1.6.12/mosquitto &
[1] 25717
guest@VM-0-13-ubuntu:~/workspace/iot_master$ 1649402555: mosquitto version 1.6.12 starting
1649402555: Using default config.
1649402555: Opening ipv4 listen socket on port 1883.
1649402555: Error: Address already in use
[1]+ Exit 1 ./3rdparty/mosquitto-1.6.12/mosquitto
guest@VM-0-13-ubuntu:~/workspace/iot_master$ ./export.sh
2022-04-08 15:23:31,990 [DEBUG] user: file=/home/guest/userdata/userinfo userinfo.json [userinfo.cpp : 16]
2022-04-08 15:23:31,990 [DEBUG] compatible: revision=0 [userinfo.cpp : 181]
2022-04-08 15:23:31,990 [DEBUG] devices: store=/home/guest/userdata/devinfo [devices.cpp : 21]
2022-04-08 15:23:48,710 [DEBUG] on_app: behavior=login, uid=Tester001 [manager.cpp : 71]
2022-04-08 15:23:48,710 [DEBUG] on_user: behavior=login [manager.cpp : 115]
2022-04-08 15:24:41,738 [DEBUG] on_app: behavior=login, uid=Tester001 [manager.cpp : 71]
2022-04-08 15:24:41,738 [DEBUG] on_user: behavior=login [manager.cpp : 115]
```

## 8.8.5. Configuring the Development Board

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

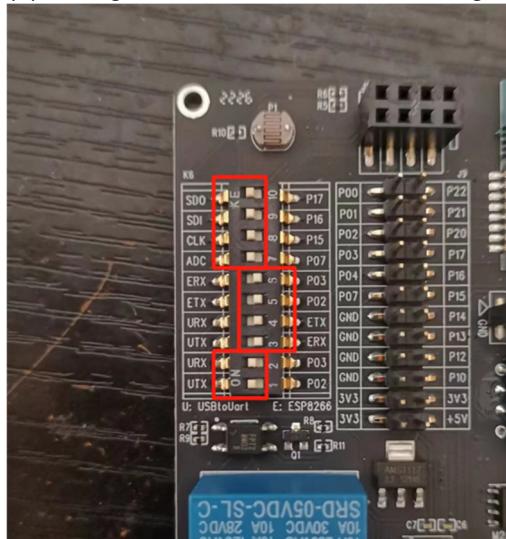
- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### 1. Build a development environment

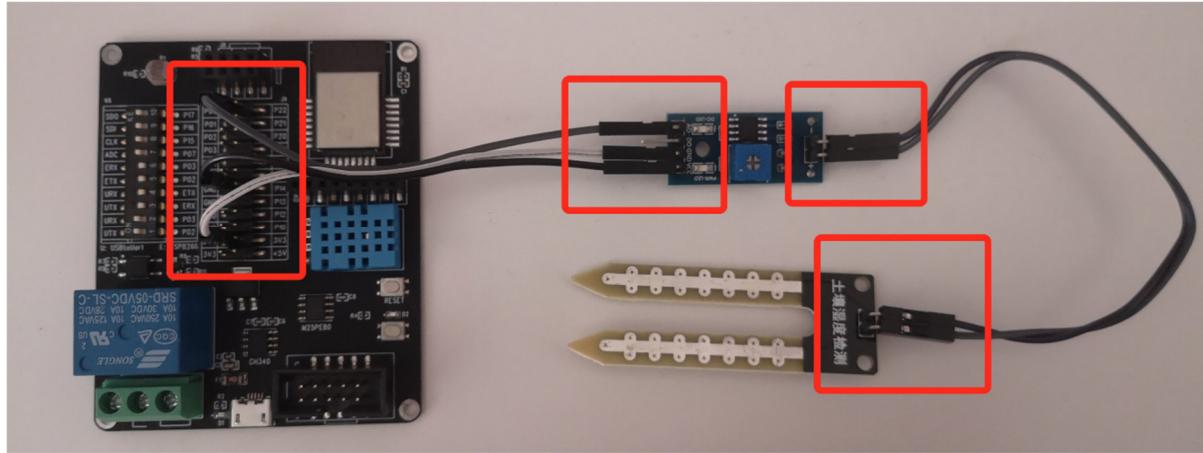
Install the development software according to Part 1: Preparation of this tutorial.

### 2. Configure the development board

(1) Configure the DIP switches in the ZigBee 3.0 standard board as shown in the figure.



(2) Connect the soil moisture sensor to the ZigBee 3.0 standard board as shown in the figure.



The pin connection instructions are as follows:

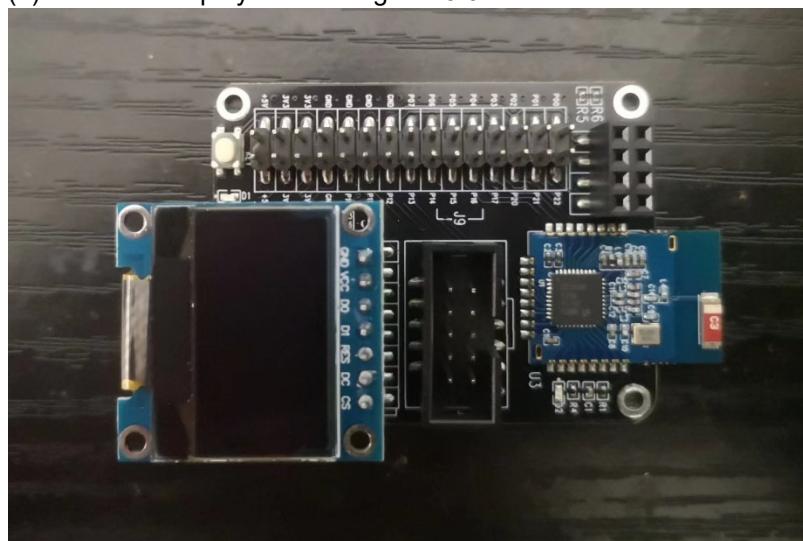
- (2A) The soil moisture probe and the sensor main board can be connected with two Dupont wires, and there is no requirement for pin docking;
- (2B) The VCC pin of the sensor main board is connected to any 3v3 pin of the ZigBee standard board;
- (2C) The GND pin of the sensor main board is connected to any GND pin of the ZigBee standard board;
- (2D) The AO pin of the sensor main board is connected to the P0\_0 pin of the ZigBee standard board;
- (2E) The DO pin of the sensor main board is left blank;

**Tip:** Why does the AO pin above need to be connected to the P0\_0 pin? This is determined by the code written. Readers can also modify the code to use other pins.

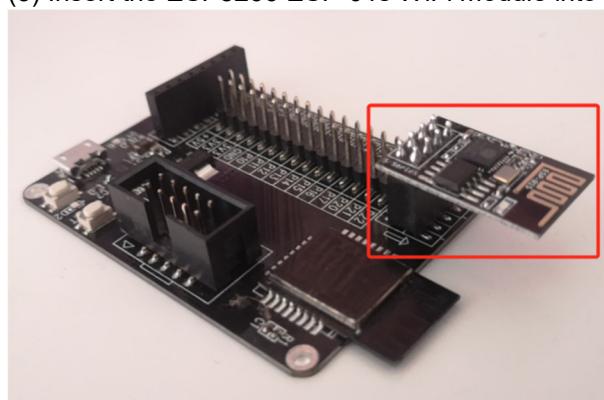
(3) Insert the display into the ZigBee 3.0 standard board, as shown in the figure.



(4) Insert the display into the ZigBee 3.0 Mini board as shown in the figure.



(5) Insert the ESP8266 ESP-01s WiFi module into the ZigBee Mini board as shown in the figure.



### 3. Burn the firmware

(1) After downloading the source code of this project, find the firmware, as shown in the figure.

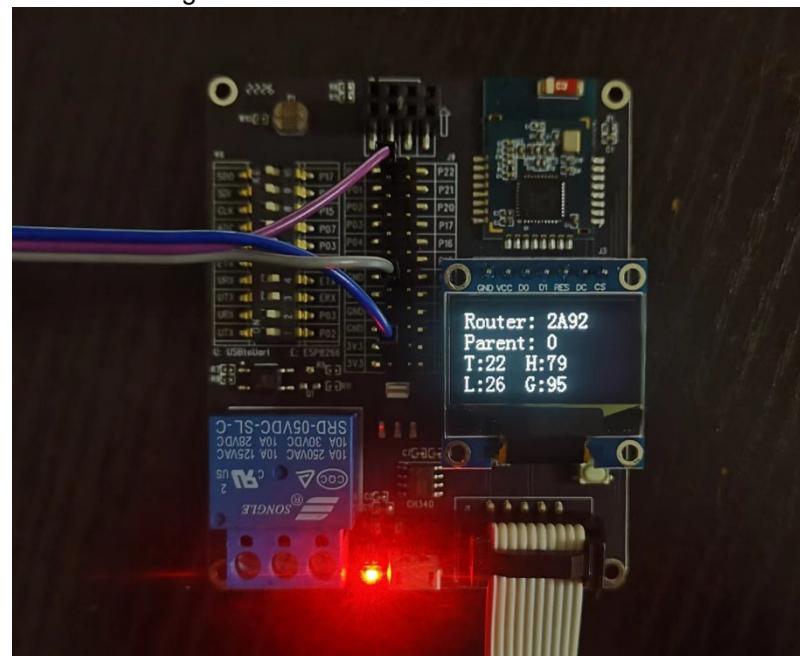


(2) Burn the Coordinator firmware into the ZigBee 3.0 Mini board. After burning, use Micro USB to power it.

(3) Burn the Router firmware into the ZigBee 3.0 standard board.

**Tip:** For detailed operation steps, please refer to "Part 2: Introduction to 51 MCU - Based on CC2530" → "Chapter 1: Basic Experiment of CC2530 Development" → "Firmware Burning".

(4) The ZigBee 3.0 standard board will automatically collect ambient temperature, humidity, light intensity and soil moisture, as shown in the figure.



## 8.8.6. Networking and WiFi Connection

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### 1. Create a WiFi hotspot

Use your phone to create a WiFi hotspot and set the name and password of the hotspot to:

- Name: auto
- Password: 12345678

### 2. Networking and WiFi Connection

(1) After burning the firmware, use the battery box to power the ZigBee 3.0 standard board.

(2) Use a Micro USB cable to connect the ZigBee 3.0 Mini board to the computer.

**Note:** Do not use the emulator to power the Mini board.

(3) The ZigBee 3.0 standard board will automatically join the ZigBee network created by the Mini board and send data to the Mini board.

(4) The ESP8266 WiFi module will automatically connect to the WiFi hotspot created earlier.

(5) The ZigBee 3.0 Mini board will report the received data to the cloud server, and the cloud server **will print the corresponding log information**.

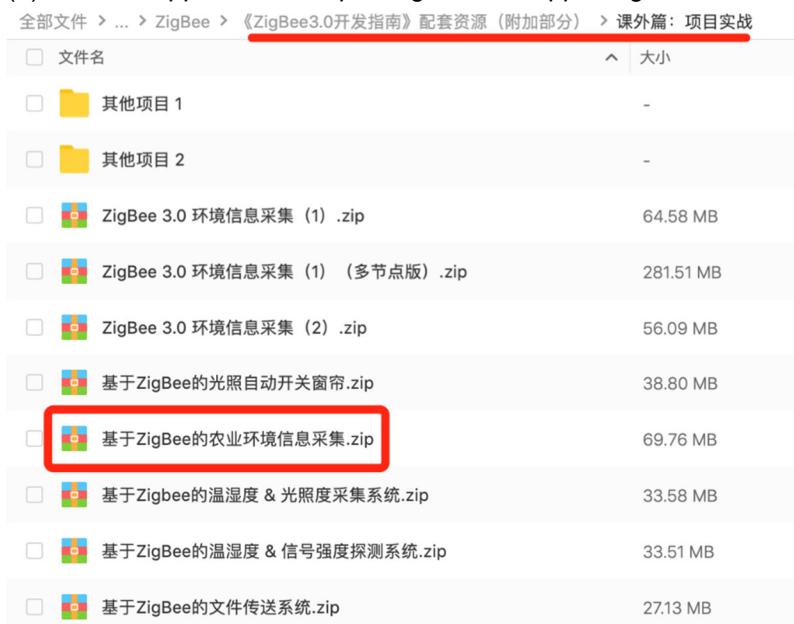
## 8.8.7. Using Android APP

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

### Using Android APP

(1) Find the App installation package in the supporting resources, as shown in the figure.



(2) After launching the App, log in to the App using this account and password:

- Account: Tester001
- Password: 12345678

(3) After ensuring that all steps are executed correctly, you can see information such as temperature and humidity, soil moisture, and light intensity in the App, as shown in the figure.



**Note:** This Android App is a test app and may have the following problems:

- (1) The interface is rough and the functions are simple;
- (2) Some new Android phones may **not be able to display the device list**.

If you need a more stable APP, we recommend:

- (1) Using the Tencent Cloud applet in "ZigBee 3.0 Environmental Information Collection (2)".
- (2) Developing your own APP.

## 8.8.8. Source code description

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Source code description

The source code structure of this project is basically the same as the following two projects. Please refer to the source code description:

- ZigBee 3.0 Environmental Information Collection (1)
- ZigBee 3.0 Environmental Information Collection (2)

#### Note:

- (1) After learning parts 1 to 4, you will understand the code of this project.
- (2) If you have not learned parts 1 to 4, you must stop reading the source code!
- (3) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.

## 8.9. Common problems in project construction

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady **does not have any technical knowledge** and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part **does not have any technical support**. Please understand!

### Node cannot join the coordinator?

When building a project, be sure to follow the following sequence:

- (1) Burn the coordinator firmware first and power it on;
- (2) Burn the router (node) firmware second and power it on;

- (3) The router (node) will automatically search for the coordinator and join it.
- (4) If the router (node) still does not automatically join, press the Reset button on the coordinator to restart the coordinator, and the router (node) will automatically join.

## How do routers (nodes) join a new coordinator?

A simple method is:

- (1) Re-burn the router (node) firmware;
- (2) After burning is complete, it will automatically search for the coordinator and join it.

# 9. Extracurricular: Advanced Electives

- 9.1. Description of Extracurricular - Advanced Elective
- 9.2. Chapter 1 - Serial Communication Protocol Design
- 9.3. Chapter 2 - Optimizing the Coordinator Project Structure
- 9.4. Chapter 3 - Coordinator host computer debugging
- 9.5. Chapter 4 - Dynamic Modification of Channels and PanId
- 9.6. Chapter 5 - Obtaining Network Short Address and MAC Address
- 9.7. Chapter 6 - Network Access Control and Whitelist
- 9.8. Chapter 7 - Coordinator Partition Storage Management
- 9.9. ZigBee 2 WiFi - Based on ESP8266
- 9.10. Capturing and analyzing ZigBee wireless messages
- 9.11. Connect to Xiaomi Aqara smart socket and temperature and humidity sensor
- 9.12. NV Application of Z-Stack
- 9.13. HAL-based external FLASH application
- 9.14. TFT display experiment (optional)
- 9.15. Lighting project source code analysis
- 9.16. TemperatureSensor project source code analysis
- 9.17. Other extracurricular projects
- 9.18. IAR EW for 8051 FAQ & Solutions

## 9.1. Description of "Extracurricular: Advanced Elective"

This project is **an extracurricular learning content**, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this part **in principle**. Please understand.

## Resource Downloads

- The supporting code for "Extracurricular: Advanced Electives" is additional information and will be automatically sent by the Taobao customer service robot after the module has been purchased and received.

- The information is as follows (continuously updated):

全部文件 > ... > ZigBee > 《ZigBee3.0开发指南》配套资源（附加部分）> 考外篇：进阶选修		^ 大小
<input type="checkbox"/>	文件名	
<input type="checkbox"/>	1. 串口通信协议设计.zip	423.00 B
<input type="checkbox"/>	2. 优化协调器工程结构.zip	7.68 MB
<input type="checkbox"/>	3. 协调器上位机调试.zip	9.56 MB
<input type="checkbox"/>	4. 信道及PanID的动态修改.zip	12.92 MB
<input type="checkbox"/>	5. ZigBee网络短地址及Mac地址的获取.zip	13.79 MB
<input type="checkbox"/>	6. ZigBee网络层入网控制及白名单.zip	13.64 MB
<input type="checkbox"/>	7. 协调器分区存储管理.zip	16.95 MB
<input type="checkbox"/>	Lighting工程源码分析.zip	28.68 MB
<input type="checkbox"/>	TemperatureSensor工程源码分析.zip	23.16 MB
<input type="checkbox"/>	Z-Stack的NV应用.zip	16.07 MB
<input type="checkbox"/>	ZigBee2Wifi_ESP8266.zip	27.21 MB
<input type="checkbox"/>	必读说明.txt	533.00 B
<input type="checkbox"/>	基于HAL的外部FLASH应用.zip	31.83 MB
<input type="checkbox"/>	接入小米智能插座和温湿度传感器.zip	43.74 MB

## Introduction to Extracurricular Courses: Advanced Electives

- Advanced content such as serial communication protocol design and optimization of coordinator engineering structure will be explained.
- Some of the study cases in "Extracurricular: Advanced Electives" will be written based on **ZLink 1.0.1**, which will help readers learn advanced knowledge and learn how to use ZLink at the same time.

## About ZLink



(2) Characteristics

- We have done a lot of packaging and secondary development on Z-Stack 3.0, providing multiple convenient APIs to greatly speed up the development process.
- Remove the cumbersome and redundant code in Z-Stack 3.0, making the protocol stack more streamlined and more efficient
- Maintained by senior Xiaomi engineers to ensure industrial-grade stability
- An excellent code architecture learning material

(3) Learn more: <https://gitee.com/study-j/zlink-for-learning>

## 9.2. Chapter 1: Serial Communication Protocol Design

### 9.2.1. Design basis

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

The protocol referred to here is the application layer protocol. When designing the application protocol, there are several basic points that need to be paid attention to: identifiability, compatibility, access control, traceability, and data integrity verification.

- First, it is **identifiable**. Generally, we use a frame header to indicate the starting position of the entire message. This frame header can be identified by a 32-bit (uint32\_t) value, such as 0xFE01A0BC. The big endian sequence is 0xFE, 0x01, 0xA0, 0xBC.

We usually call this value a magic number.

- Then there is **compatibility**. Generally, we use one byte to identify the version number of the message. The purpose of this version number is to ensure compatibility between the upper and lower sides when the protocol format changes in the future. We call this byte identifier: revision. The compatibility method in application logic can be as follows: the upper computer checks the revision of the lower computer and negotiates a version that can be parsed and processed by both sides for data processing.
- **Access control** can use 1 byte (`uint8_t`) or 2 bytes (`uint16_t`), which is called access control; the data organization format can use a one-hot code, that is, each bit represents an identifier, 1 byte (8 bits) has a maximum of 8 identifiers, and 2 bytes (16 bits) have a maximum of 16 identifiers. For example, `0x0100` represents a synchronization frame, `0x1000` means a response is required, then `0x0100 | 0x1000` indicates that this frame is a synchronization frame and a response is required. We can use access control to do the logic of packet transmission.
- **Traceability** generally means that the sent message and the response message must be able to match, so a sequence code is required in each message, called a sequence number; it can be 1 to 2 bytes, and each transmission is self-added. When the message needs to be responded to, the sequence code of the response message remains consistent. Next is the user data, which generally uses 2 bytes (up to 65535) to indicate the data length, followed by the user data, also called payload.

**Data integrity check** can be performed using CRC16 calculation, which requires 2 bytes. CRC16 calculation is performed from the packet header to the end of the payload to ensure data integrity (no tampering or bit errors).

## 9.2.2. Protocol Format

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Based on the above starting point, we can define the protocol as follows:

起始码	revision	访问控制	序列码	数据长度	用户数据	CRC16
4 bytes	1 byte	2 bytes	1 byte	2 bytes	—	2 bytes

Such a protocol can basically meet most application scenarios; access control can also be further refined into control segments and data segments, for example, the first byte represents control information and the second byte stores data.

At the same time, it is also necessary to explain the byte order, that is, the big endian or little endian issue. Here we recommend using the big endian order. The so-called big and small **endian order** is actually the order of the bytes. For example, the data `0x12AB` is 16 bits, 2 bytes. The big endian order is: `0x12 0xAB`, and the little endian order is the opposite: `0xAB 0x12`; just keep it consistent when the upper and lower computers communicate.

Another thing to note is that the data transmitted by serial communication is essentially **streaming data**, that is, the start and end need to be judged and processed by ourselves. This is also where the most problems occur. We often read serial data and process it. However, when transmitting large data blocks, we often find that there are errors or packet loss. This is actually caused by insufficient understanding of streaming data. Therefore, every time we read serial data, we need to pre-process the data to get a complete message before applying it. Finally, I would like to add that serial communication is full-duplex communication, which is a typical asynchronous data processing logic. This is very different from SPI. This is why it is strongly recommended to use serial communication instead of SPI between master and slave, because asynchronous processing can be done to improve overall efficiency. SPI transmission is fast, but it requires synchronous processing, which is very troublesome.

## 9.3. Chapter 2: Optimizing the Coordinator Project Structure

### 9.3.1. Project Structure

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

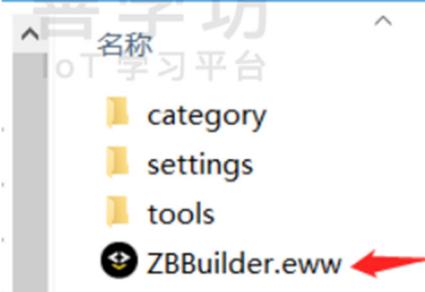
1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers **will** answer community questions as soon as possible, but they are front-line developers and [**cannot guarantee**] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

We put ZStack in the ZBProject directory, and the optimized project structure is as follows:

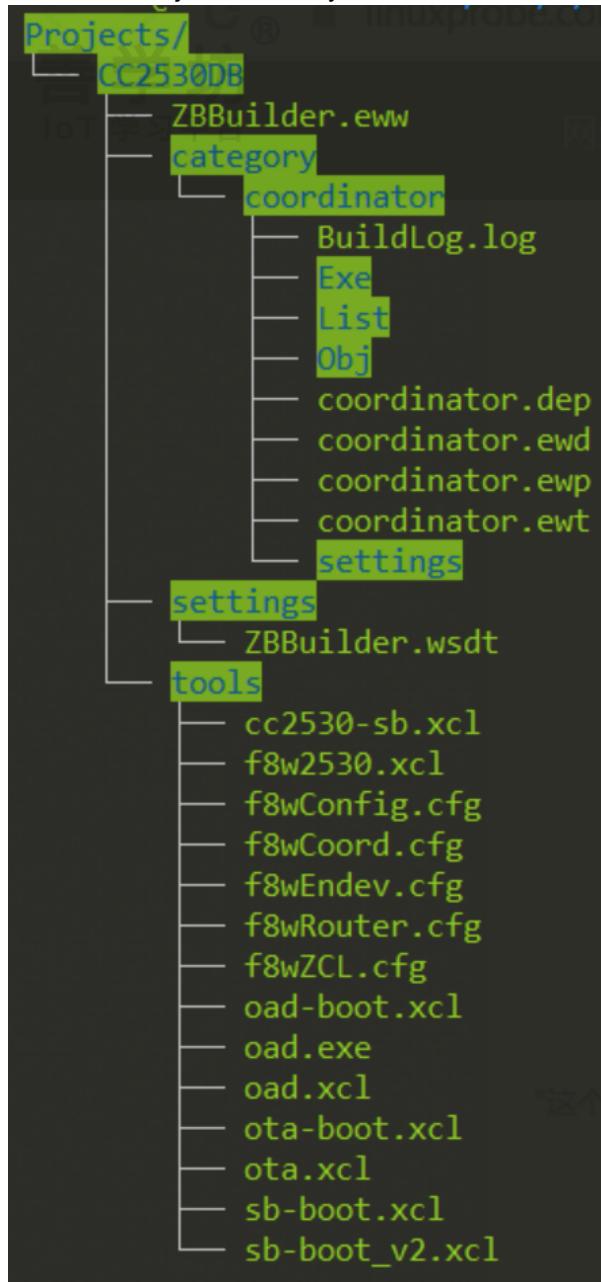
4. 深入篇\2. 优化协调器工程结构\ZBProject\Z-Stack 3.0.1		
名称		修改日
APPs		2020/
Components		2020/
Libraries		2020/
Projects		2020/

- **APPs**: Store our own application code.
- **Components**: Components of the solution provider.
- **Libraries**: library files of solution providers.
- **Projects**: Directory for storing project files.

We can open the project directly in Projects:

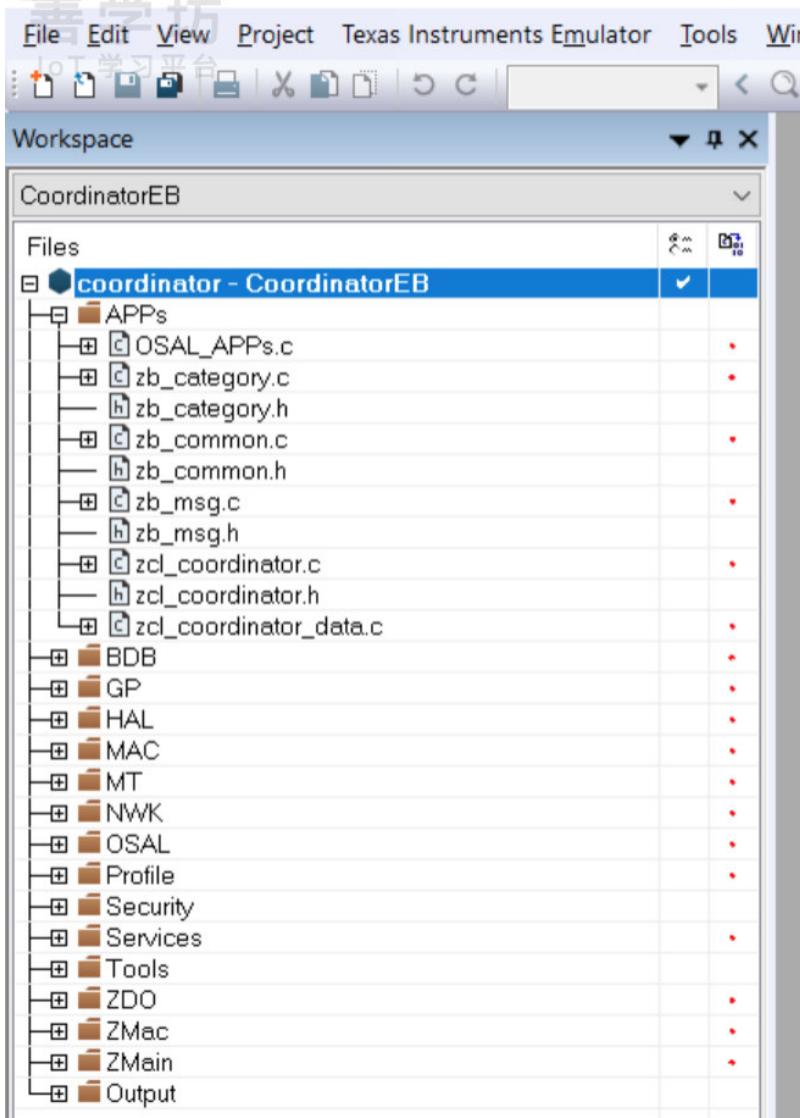


The entire Projects directory structure is as follows:



- **category:** Stores project files of different device types, such as coordinators, smart sockets, sensors, etc. The final output burning files are in the Exe directory.
  - **tools:** Stores engineering tool files, which will be discussed later and involve the configuration of some parameters.
  - **settings:** Stores project settings files.
- Other directories are mainly where project files are stored, which is not the focus.

After opening the project:



The source code corresponding to APPs can be found in the project directory APPs.

`zb_msg.h` and `zb_msg.c` complete the functions of serial port communication and protocol analysis.

`zb_category.h` and `zb_category.c` process the message application content of `zb_msg`, and are also the intermediate paths for the upper layer to access `zb_msg`, thus achieving decoupling.

`zcl_coordinator` is the coordinator application logic code, which is related to the ZigBee protocol.

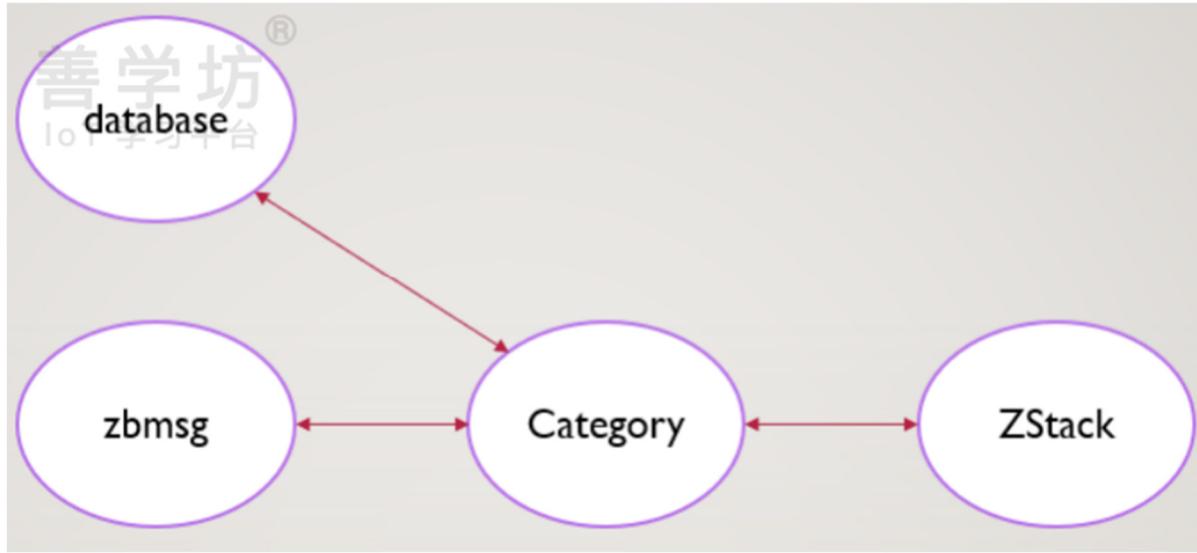
## 9.3.2. Detailed explanation of application framework

### 9.3.2.1. Framework Description

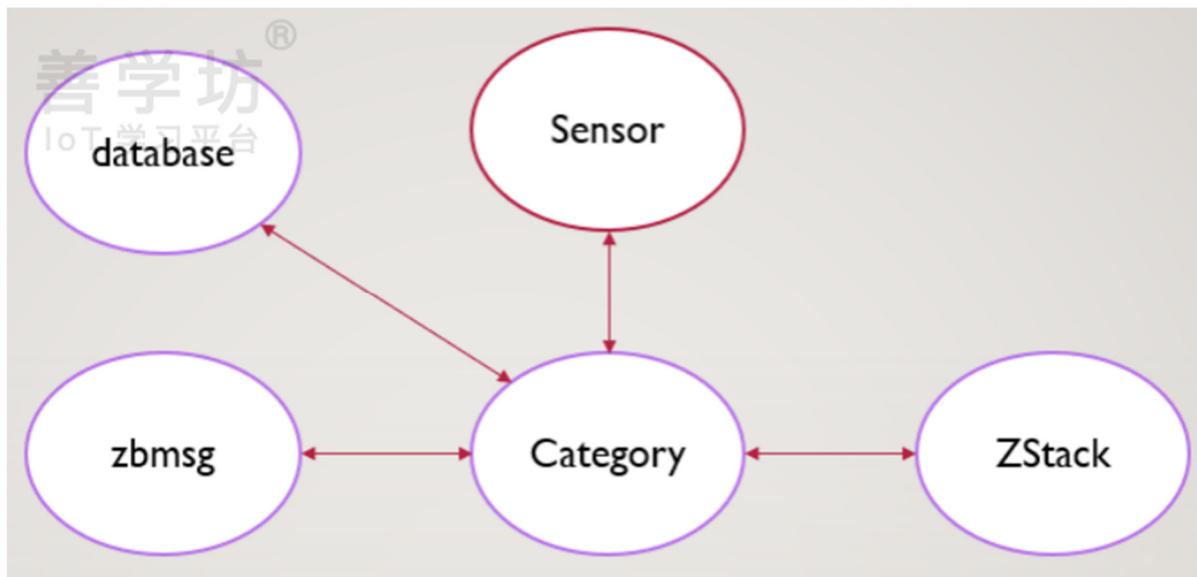
Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!



Currently, there is not much content in the application code. Basically, zmsgid completes the protocol processing of serial communication, and then sends the processed application data to the classifier Category for distribution and processing. The advantage of this architecture is that it decouples different functions very well. For example, if we want to add a new application logic of a temperature and humidity sensor in the future, we can schedule it in Category as follows:



For example, if we use external Flash as a database function, then access to the database must be uniformly scheduled by the category, and functions such as device management can also be abstracted.

### 9.3.2.2. zmsgid

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Due to the limitation of computing power and storage size, we cannot use C++ to write applications; therefore, we use C language to do object programming, starting with zb\_msg.h:

```

1 #ifndef __ZB_MSG_H__
2 #define __ZB_MSG_H__
3
4 #include "zb_common.h"
5
6 #ifdef __cplusplus
7 extern "C"
8 {
9 #endif
10
11 typedef struct {
12     void (*init)(void);
13     int (*send)(uint8_t *payload, uint8_t size);
14 } zb_msg_t;
15
16 extern const zb_msg_t *zbmsg_;
17
18 #ifdef __cplusplus
19 }
20#endif
21
22#endif /* __ZB_MSG_H__ */
23

```

The structure pointer variable `zbmsg_` is similar to the singleton mode in C++. We can access functions through the `zbmsg_` pointer: `init`, `send`; obviously, the function pointer variable is stored in the structure `zb_msg_t`, and then we assign it in the `zb_msg.c` source code:

```

const zb_msg_t *zbmsg_ = &(const zb_msg_t){
    .init = zb_msg_init,
    .send = zb_msg_send,
};

```

There is an important function in `zbmsg` as follows, `zb_msg_handler`:

```

void zb_msg_handler(uint8_t *buf, uint16_t len)
{
    /* Start code */
    if(len < 4 || !(buf[0] == 0xFE && buf[1] == 0x01 && buf[2] == 0xA0 && buf[3] == 0xBC)) return;

    /* CRC16 */
    uint16_t crc16 = ((uint16_t)buf[len-2] << 8) | (uint16_t)buf[len-1];

    /* Check CRC16 */
    if (crc16 != crc16_x25(buf, len - 2)) return;

    /* Revision */
    uint8_t revision = buf[4];

    /* Access control: control flag */
    uint8_t actrl = buf[5];

    if (actrl & ZB_MSG_ACTRL_SYNC) {
        if (revision > zb_msg_match_revision_) zb_msg_commit(NULL, 0, (uint16_t)ZB_MSG_ACTRL_SYNC << 8, 255);
        else zb_msg_match_revision_ = revision;

        return;
    }

    if (revision > zb_msg_match_revision_) { zb_msg_commit(NULL, 0, (uint16_t)ZB_MSG_ACTRL_REVISION_ERR << 8, 0xFF); return; }

    if (revision == 0x00) {
        /* Access control: data flag */
        uint8_t adata = buf[6]; (void)adata;

        /* Seq */
        uint8_t seq = buf[7];

        /* Payload length */
        uint16_t size = ((uint16_t)buf[8] << 8) | (uint16_t)buf[9];

        if (actrl & ZB_MSG_ACTRL_ACK_REQ) zb_msg_commit(NULL, 0, (uint16_t)ZB_MSG_ACTRL_ACK_RSP << 8, seq);

        zbcategory_->on_host_msg(&buf[10], size); ←
    }
}

```

This function mainly performs protocol analysis on the received serial port data, and finally extracts the application data and transfers it through zbcategory\_. Obviously, zbcategory\_ is also an object-oriented design idea. There is a group of member functions that can be used, such as the on\_host\_msg function called in the figure. The design of zb\_msg is very concise, and you can browse the code by yourself~

### 9.3.2.3. zbcategory

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support:

1. Customer service provides simple technical support, mainly self-study.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answers. Technical answers are hard, please understand!

```

10
11     typedef struct {
12         void (*init)(void);
13         void (*on_host_msg)(uint8_t *payload, uint16_t size);
14     } zb_category_t;
15
16     extern const zb_category_t *zbcategory_;
17

```

Obviously, we can see the same definition in zb\_category.h. The zb\_category\_t structure carries the member functions. Currently, zb\_category has no actual functions, which will be supplemented later~

## 9.4. Chapter 3: Coordinator host computer debugging

### 9.4.1. Host computer description

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

## Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

(1) In the previous two chapters, we designed a set of serial port protocols and a framework for the coordinator. To facilitate debugging, our team also designed a host computer for debugging the coordinator. The host computer can be used to send packaged serial port protocols to facilitate development and debugging. In this way, we can focus more work on the application function development of the coordinator.

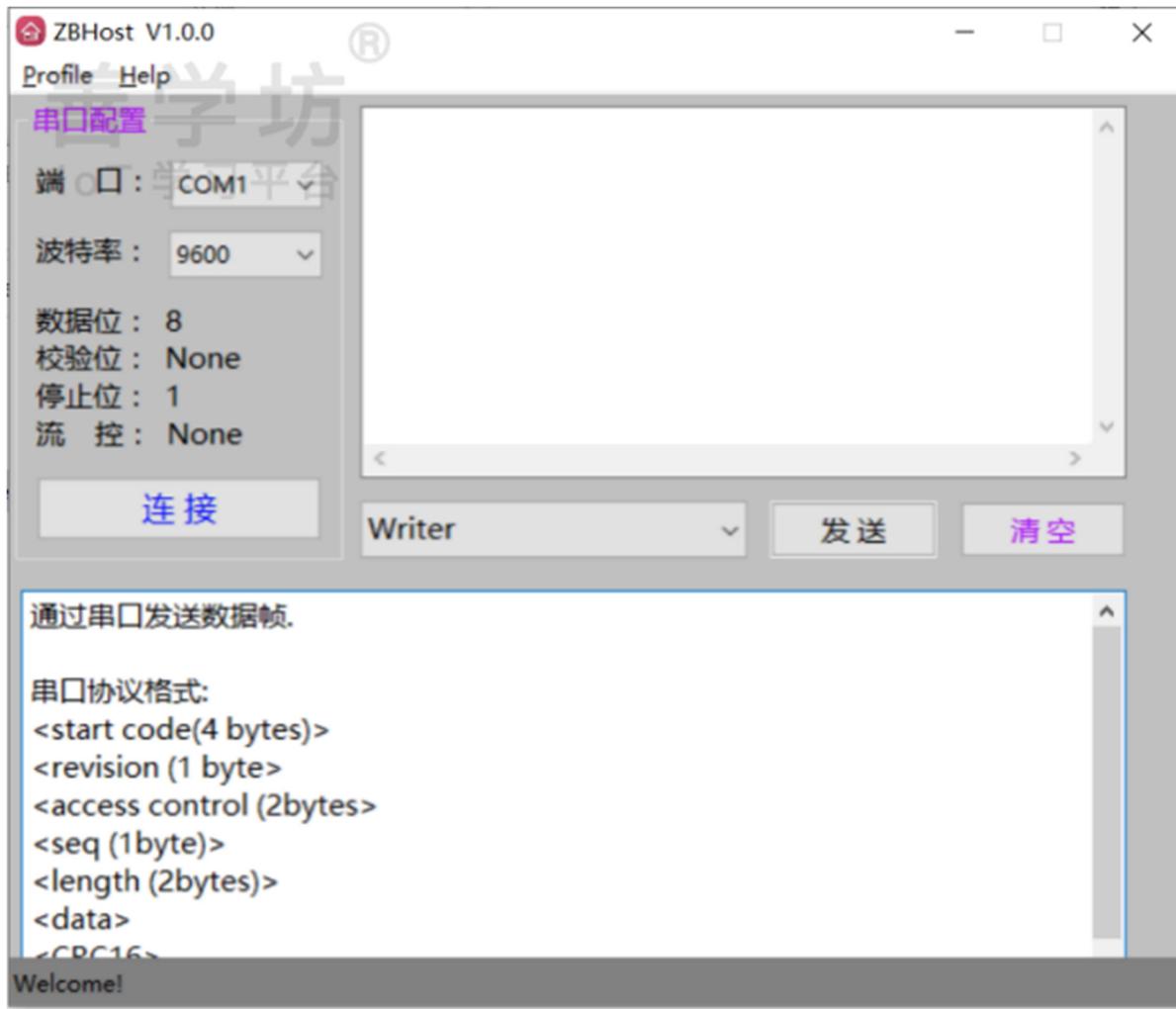
(2) The location of the host computer is shown in the figure.

The screenshot shows a file management interface with the following structure:

- Top navigation bar: 上级 (Up), 新建文件夹 (New Folder), 新建在线文档 (New Online Document), 离线下载 (Offline Download).
- Breadcrumbs: 返回上一级 (Return to Previous Level) > 全部文件 > ... > ZigBee > 《ZigBee3.0开发指...》 (Advanced Selection).
- File list table:

文件名	修改时间	大小
1.串口通信协议设计.zip	2023-07-13 15:05	423B
2.优化协调器工程建构.zip	2023-07-13 15:05	7.7M
<b>3.协调器上位机调试.zip</b>	<b>2023-07-13 15:05</b>	<b>9.6M</b>
4.信道及PanId的动态修改.zip	2023-07-13 15:05	12.9M
5.ZigBee网络短地址及Mac地址的获取.zip	2023-07-13 15:05	13.8M
<b>6.ZigBee网络层入网控制及白名单.zip</b>	<b>2023-07-13 15:05</b>	<b>16.9M</b>
7.协调器分区存储管理.zip	2023-07-13 15:05	533B
必读说明.txt	2023-07-13 15:05	31.8M
基于HAL的外部FLASH应用.zip	2023-07-13 15:05	43.7M
接入小米智能插座和温湿度传感器.zip	2023-07-13 15:05	28.7M
Lighting工程源码分析.zip	2023-07-13 15:05	23.2M
Z-Stack的NV应用.zip	2023-07-13 15:05	16.1M
ZigBee2Wifi_ESP8266.zip	2023-07-13 15:05	27.2M

(3) Run the software as shown in the figure.



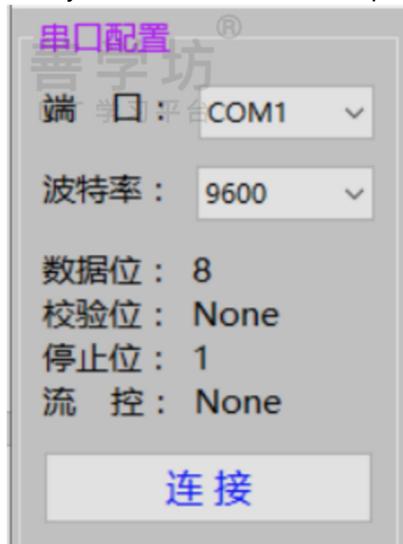
## 9.4.2. Debugging Instructions

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

First you need to set the serial port modification configuration, that is, which serial port and baud rate:



Then connect, the connection is successful as follows:

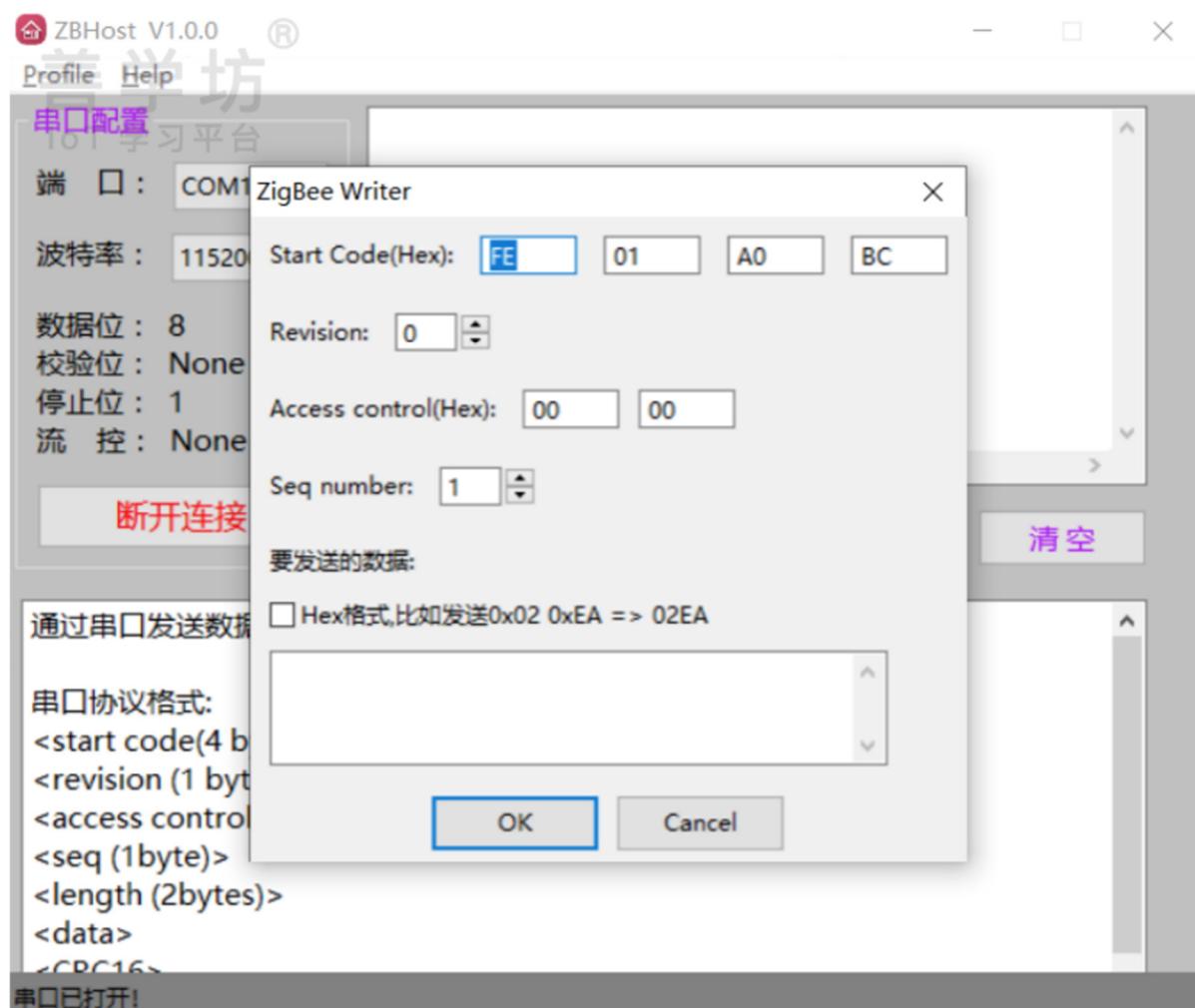


To disconnect, click: "Disconnect"

Then select the command to be executed, such as Writer, where Clear is used to clear the receiving buffer:



The Send button will pop up the corresponding dialog box:



We can enter the data we want to send in the input box. If it is a string, do not check the Hex option. After entering, click OK to send it according to the previously defined serial port protocol content. We can also receive the data in the coordinator.

At the same time, if the data from the coordinator is received, it will be displayed in the display box in hexadecimal format. We will practice it in the next chapter~

## 9.5. Chapter 4: Dynamic Modification of Channels and PanId

### 9.5.1. Serial Port Protocol

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

The serial port protocol mentioned in the current chapter refers to the protocol of the application layer. For example, in this chapter, we define the structure of the application layer protocol as follows:

Action	Data
0x00	PanId[15:8] PanId[7:0] Channel

The first byte of the application protocol is used to describe the action, called Action. For example, 0x00 indicates the function of modifying the network's Panid and channel. Different data is then carried depending on the Action. For example, this chapter requires the use of new panid and channel. The byte order is big endian, panid occupies 2 bytes, and channel occupies 1 byte.

### 9.5.2. Important interface description

#### 9.5.2.1. NIB

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

In ZStack, the network information of the device is stored in a database called NIB. The definition of this database can be found in nwk.h: nwkIB\_t

NIB has a global external variable declared in nwk.h: extern nwkIB\_t \_NIB; Therefore, we can indirectly access the contents of NIB. The contents of NIB will also be stored in NV, which is a Flash space that is saved when power is off, so that NIB can be loaded from NV to memory the next time power is turned on.

#### 9.5.2.2. NLME\_UpdateNV

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.

2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>  
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

As mentioned above, NIB will be stored in NV. We can store the information through the function NLME\_UpdateNV. Then the idea of modifying Panid and Channel is very clear. First modify the NIB in memory and then update it to NV. Restart and complete the data reload.

The definition in NIB is as follows:

```
264 // non-settable
265     uint16 nwkDevAddress;
266     byte nwkLogicalChannel;
267     uint16 nwkCoordAddress;
268     byte nwkCoordExtAddress[Z_EXTADDR_LEN];
269     uint16 nwkPanId;
270
```

These two places are where we need to modify!

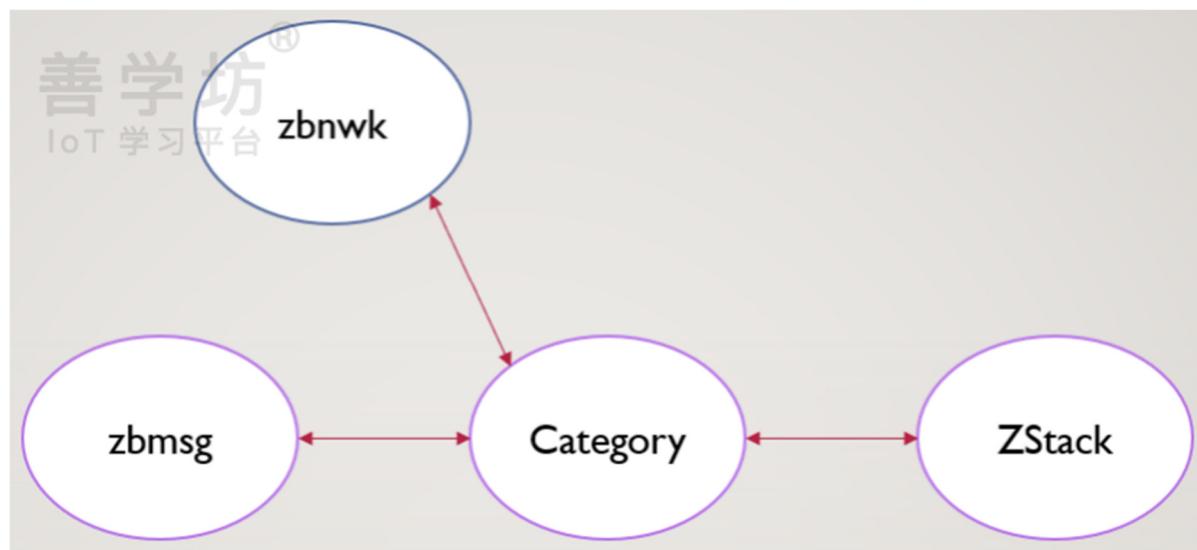
### 9.5.3. Architecture Adjustment

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

We call the network operation a service and create new files zb\_network.h/c. These two files are dedicated to managing network-related operations. In the coordinator architecture, they belong to the classifier's subordinate function module. The adjusted framework is as follows:



interface:

```
11 typedef struct {
12     void (*set_panid_channel) (uint16_t panid, uint8_t channel);
13 } zb_network_t;
14
15 extern const zb_network_t *zbnwk_;
```

## 9.5.4. Application

### 9.5.4.1. zbnwk interface implementation

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

In zb\_network.c, we implemented the interface for modifying panid and channel:

```
1 #include "zb_network.h"
2
3 #include "nwk.h"
4
5 static void zb_nwk_set_panid_channel(uint16_t panid, uint8_t channel);
6
7 const zb_network_t *zbnwk_ = &(const zb_network_t){
8     .set_panid_channel = zb_nwk_set_panid_channel,
9 };
10
11 void zb_nwk_set_panid_channel(uint16_t panid, uint8_t channel)
12 {
13     _NIB.nwkPanId = panid;
14     _NIB.nwkLogicalChannel = channel;
15
16     NLME_UpdateNV(NWK_NV_NIB_ENABLE);
17 }
18
```

### 9.5.4.2. Serial communication analysis

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

The application protocol is finally parsed in the classifier:

## zb\_category\_on\_host\_msg(uint8 \*, uint16)

```

1 #include "zb_category.h"
2
3 #include "zb_network.h"
4
5 #include "zb_msg.h"
6
7 static void zb_category_init(void);
8 static void zb_category_on_host_msg(uint8_t *payload, uint16_t size);
9
10 const zb_category_t *zbcategory_ = &(const zb_category_t){
11     .init = zb_category_init,
12     .on_host_msg = zb_category_on_host_msg,
13 };
14
15 void zb_category_init()
16 {
17     zbmmsg_->init();
18 }
19
20 void zb_category_on_host_msg(uint8_t *payload, uint16_t size)
21 {
22     uint8_t action = payload[0];
23
24     switch (action) {
25         case 0x00: zbnwk_->set_panid_channel((uint16_t)payload[1] << 8 | payload[2], payload[3]); break;
26     }
27 }
```

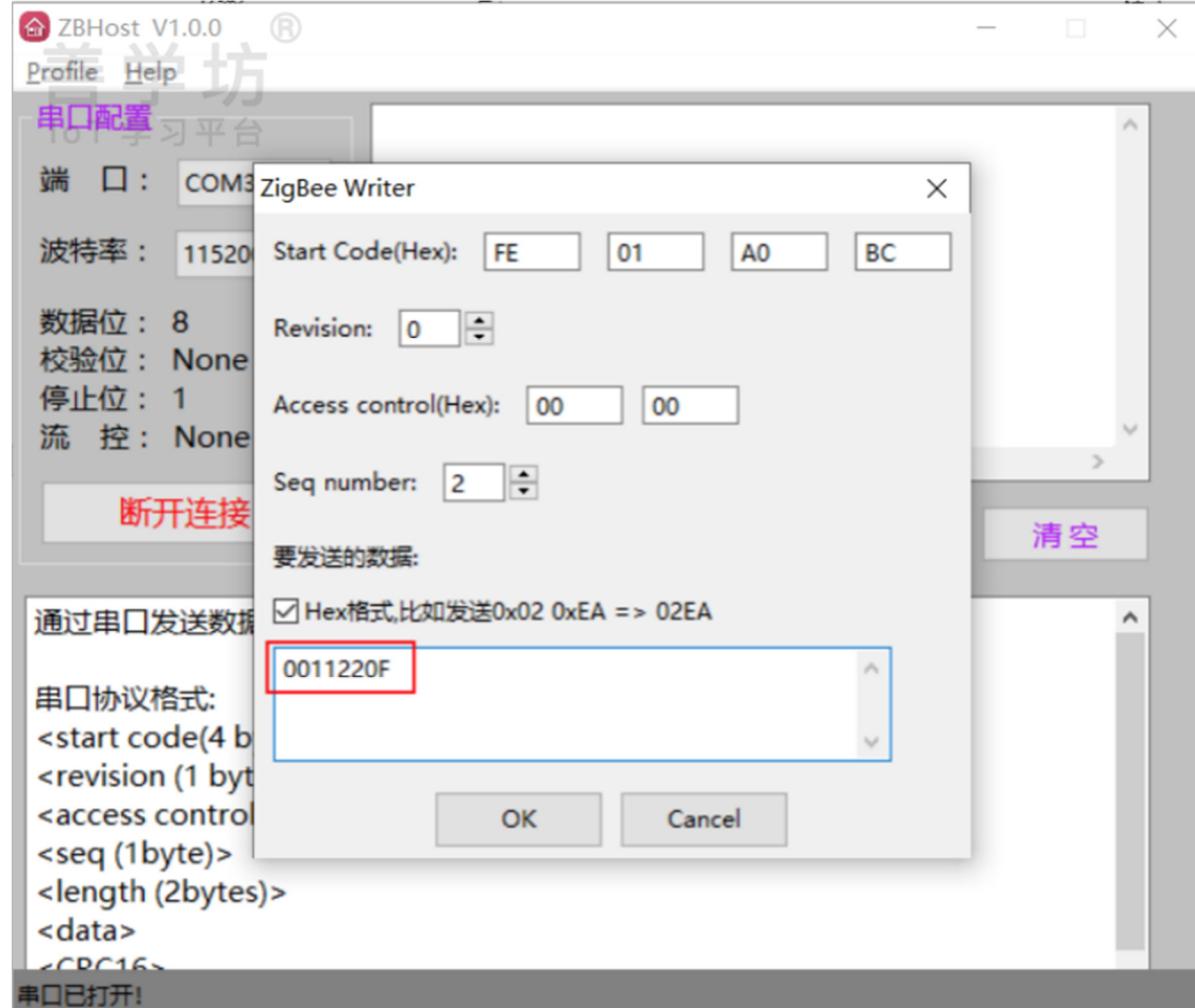
### 9.5.4.3. Burning and debugging

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

**Technical support instructions:**

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

After compiling the project, burn the program to the board, then connect the serial cable and open the debugging tool ZBHost:



The input meaning of the above picture is: 0x00, 0x11, 0x22, 0x0F

- 0x00 is the action to set panid and channel
- 0x11 is the high 8 bits of panid
- 0x22 is the lower 8 bits of panid
- 0x0F is channel, channel 15

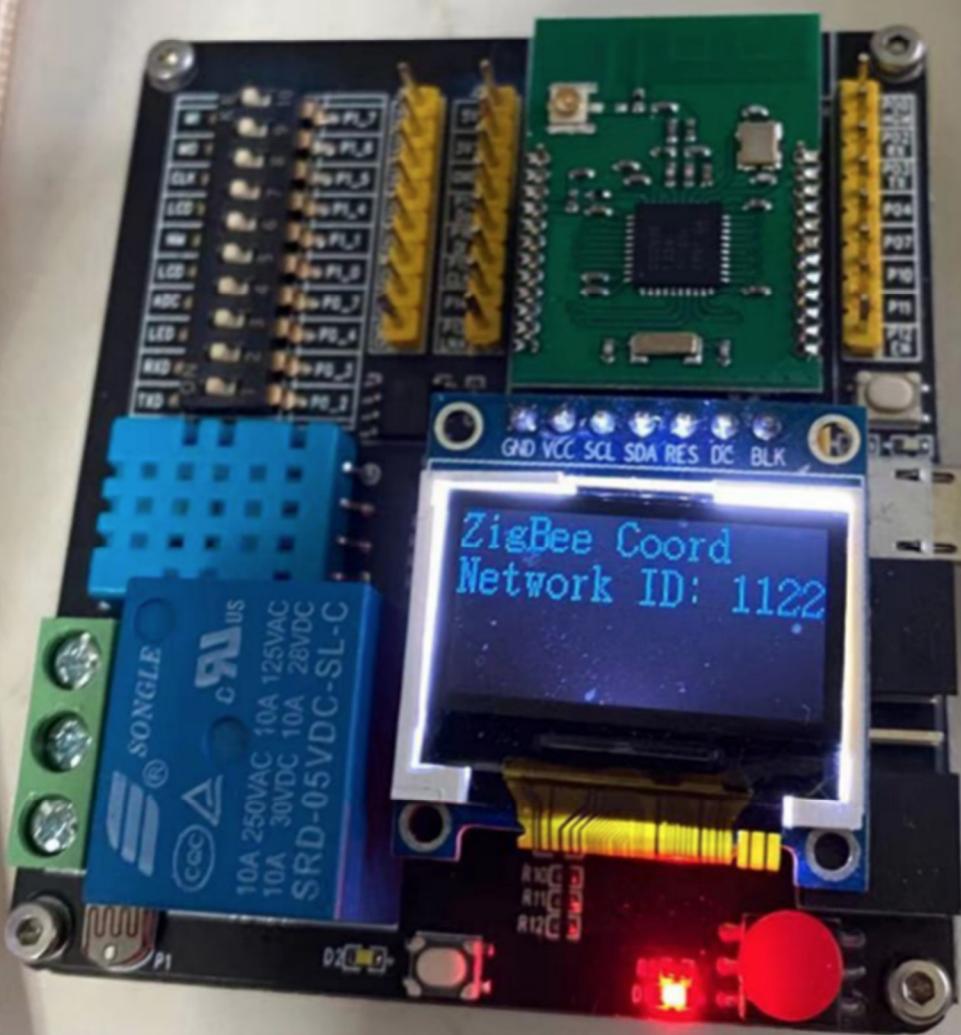
Click OK to send the message.

Note that the Hex format needs to be checked, and it is best not to modify other places because the protocols of the upper and lower computers need to be consistent.

The coordinator information after burning is as follows:



The updated coordinator information is as follows:



# 9.6. Chapter 5: Obtaining Network Short Address and MAC Address

## 9.6.1. Interface Description

### 9.6.1.1. Description

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

In ZigBee communication, the most common addresses are 16-bit network addresses and 64-bit Mac addresses. In this section, we explain how to obtain the Mac address through the network address, and how to obtain the network address through the Mac address.

- ZDP\_NwkAddrReq: Get network address
- ZDP\_IEEEAddrReq: Get Mac address

Note: These two functions do not return results directly, but send wireless messages to the sub-device and wait for the sub-device's data to return results, so they are asynchronous.

### 9.6.1.2. Calling Process

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

In the coordinator framework, we encapsulate these two interfaces in the network service:

```
--  
11  typedef struct {  
12      void (*set_panid_channel) (uint16_t panid, uint8_t channel);  
13  
14      void (*nwkaddr_req) (uint8 *IEEEAddress);  
15      void (*macaddr_req) (uint16_t nwkaddr);  
16  } zb_network_t;
```

Specific implementation:

```

26 void zb_nwk_nwkaddr_req(uint8_t *mac)
27 {
28     ZDP_NwkAddrReq(mac, ZDP_ADDR_REQTYPE_SINGLE, 0, 0);
29 }
30
31 void zb_nwk_macaddr_req(uint16_t nwkaddr)
32 {
33     ZDP_IEEEAddrReq(nwkaddr, ZDP_ADDR_REQTYPE_SINGLE, 0, 0);
34 }

```

Interface encapsulation is for unified management and convenient for calling other applications. After the above encapsulation is completed, we can process it in the classifier, so we define the application layer serial port protocol as follows:

Action	Data
0x01	mac[7], mac[6].....mac[1], mac[0]
0x02	nwkaddr[15:8], nwkaddr[7:0]

Processing logic in the classifier:

```

20 void zb_category_on_host_msg(uint8_t *payload, uint16_t size)
21 {
22     uint8_t action = payload[0];
23
24     switch (action) {
25         case 0x00: zbnwk_->set_panid_channel((uint16_t)payload[1] << 8 | payload[2], payload[3]); break;
26
27         case 0x01: zbnwk_->nwkaddr_req(&payload[1]); break;
28
29         case 0x02: zbnwk_->macaddr_req((uint16_t)payload[1] << 8 | payload[2]); break;
30     }
31 }

```

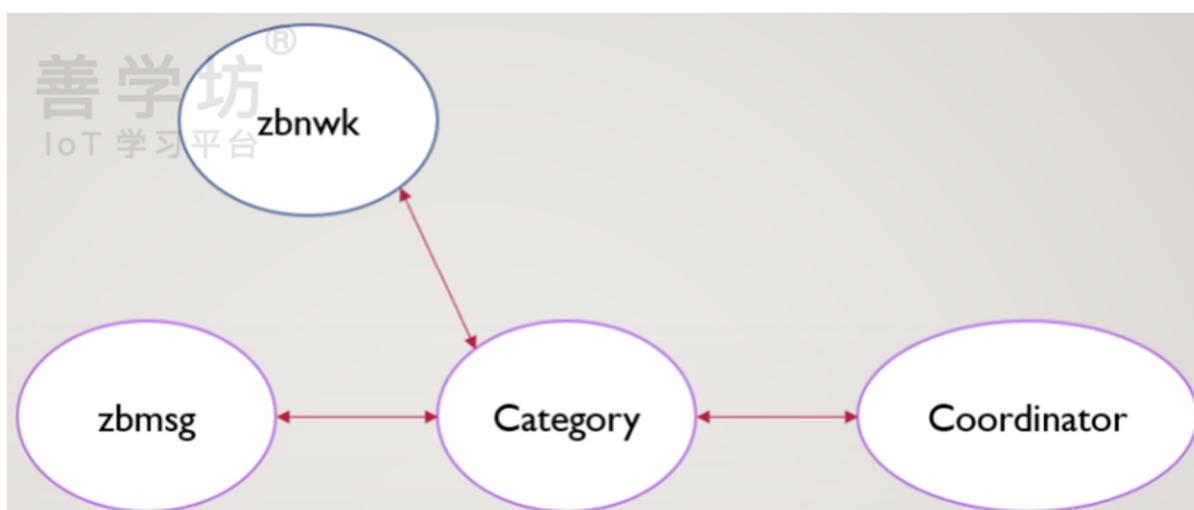
### 9.6.1.3. Asynchronous Data

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

- Generally, self-study is the main method.
- You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
- Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

After receiving the request message, the sub-device will package the application data to the coordinator. This data will eventually be received by ZDO, so we need to parse the input data of ZDO and review the application framework we designed:



Obviously, we need to parse the data in the coordinator and give it to the classifier, so we need to process the ZDO message in coordinator.c.

Register ZDO messages: NWK\_addr\_rsp, IEEE\_addr\_rsp

```
52     NULL                                // RSSI Location Response command
53 };
54
55 void zcl_coordinator_init(byte task_id)
56 {
57     zcl_coordinator_taskid = task_id;
58
59     // This app is part of the Home Automation Profile
60     bdb_RegisterSimpleDescriptor(&simpledesc_zha);
61     bdb_RegisterSimpleDescriptor(&simpledesc_zll);
62
63     // Register the ZCL General Cluster Library callback functions
64     zclGeneral_RegisterCmdCallbacks(ENDPOINT_ZHA, &zcl_coordinator_cmdcallbacks);
65     zclGeneral_RegisterCmdCallbacks(ENDPOINT_ZLL, &zcl_coordinator_cmdcallbacks);
66
67     // Register the application's attribute list
68     zcl_registerAttrList(ENDPOINT_ZHA, zcl_coordinator_attrs_len, zcl_coordinatorAttrs);
69     zcl_registerAttrList(ENDPOINT_ZLL, zcl_coordinator_attrs_len, zcl_coordinatorAttrs);
70
71     // Register the Application to receive the unprocessed Foundation command/response messages
72     zcl_registerForMsgExt(zcl_coordinator_taskid, ENDPOINT_ZHA);
73     zcl_registerForMsgExt(zcl_coordinator_taskid, ENDPOINT_ZLL);
74
75     ZDO_RegisterForZDOMsg(zcl_coordinator_taskid, NWK_addr_rsp);
76     ZDO_RegisterForZDOMsg(zcl_coordinator_taskid, IEEE_addr_rsp);
77
78     bdb_RegisterCommissioningStatusCB(zcl_coordinator_on_commissioning);
79
80     bdb_StartCommissioning(BDB_COMMISSIONING_MODE_NWK_FORMATION | BDB_COMMISSIONING_MODE_FINDING_BINDING);
81
82     zbcategory_->init();
83 }
```

Intercept ZDO messages:

```
85     uint16 zcl_coordinator_event_loop(uint8 task_id, uint16 events)
86 {
87     afIncomingMSGPacket_t *msg;
88
89     (void)task_id;
90
91     if (events & SYS_EVENT_MSG) {
92         while ((msg = (afIncomingMSGPacket_t *)osal_msg_receive(zcl_coordinator_taskid))) {
93             switch (msg->hdr.event) {
94                 case ZCL_INCOMING_MSG: zcl_coordinator_on_incoming_msgs((zclIncomingMsg_t *)msg); break;
95
96                 case ZDO_CB_MSG: zcl_coordinator_on_zdo_msg((zdoIncomingMsg_t *)msg); break;
97
98                 case ZDO_STATE_CHANGE: break;
99
100                default: break;
101            /*switch*/
102
103            osal_msg_deallocate((uint8 *)msg);
104        /* while */
105
106        return (events ^ SYS_EVENT_MSG);
107    }
108
109    return 0;
110 }
```

Note that this place will only be called if the event is registered.

We process it in this function, the content is as follows:

```

202 void zcl_coordinator_on_zdo_msg(zdoIncomingMsg_t *msg)
203 {
204     ZDO_NwkIEEEAddrResp_t *pAddrRsp = NULL;
205
206     switch(msg->clusterID) {
207         /* Network address responded. */
208         case NWK_addr_rsp:
209             if ((pAddrRsp = ZDO_ParseAddrRsp(msg)) != NULL && pAddrRsp->status == ZSuccess)
210                 zbcategory_->on_nwkaddr_rsp(pAddrRsp->extAddr, pAddrRsp->nwkAddr);
211             break;
212
213         /* MAC address responded. */
214         case IEEE_addr_rsp:
215             if ((pAddrRsp = ZDO_ParseAddrRsp(msg)) != NULL && pAddrRsp->status == ZSuccess)
216                 zbcategory_->on_macaddr_rsp(pAddrRsp->extAddr, pAddrRsp->nwkAddr);
217             break;
218     }
219
220     if (pAddrRsp != NULL) osal_mem_free(pAddrRsp);
221 }
222

```

After parsing is completed, the key information is directly transferred to the classifier for processing. The specific operations of the classifier are:

```

39 void zb_category_on_nwkaddr_rsp(uint8_t *mac, uint16_t nwkaddr)
40 {
41     uint8_t payload[] = {
42         0x00,
43         mac[7], mac[6], mac[5], mac[4], mac[3], mac[2], mac[1], mac[0],
44         nwkaddr >> 8, (uint8_t)nwkaddr
45     };
46
47     zbmsg_->send(payload, sizeof(payload));
48 }
49
50 void zb_category_on_macaddr_rsp(uint8_t *mac, uint16_t nwkaddr)
51 {
52     uint8_t payload[] = {
53         0x01,
54         mac[7], mac[6], mac[5], mac[4], mac[3], mac[2], mac[1], mac[0],
55         nwkaddr >> 8, (uint8_t)nwkaddr
56     };
57
58     zbmsg_->send(payload, sizeof(payload));
59 }
60

```

The content is very simple, it directly packages the application layer data and then sends it back to the host computer through zbmsg\_-.

Let's review again, zbmsg\_- is packaged into the serial port protocol data we defined. By default, we will not modify zbmsg\_. Sub-packet transmission and other access controls are required, so zbmsg is a function of the link layer.

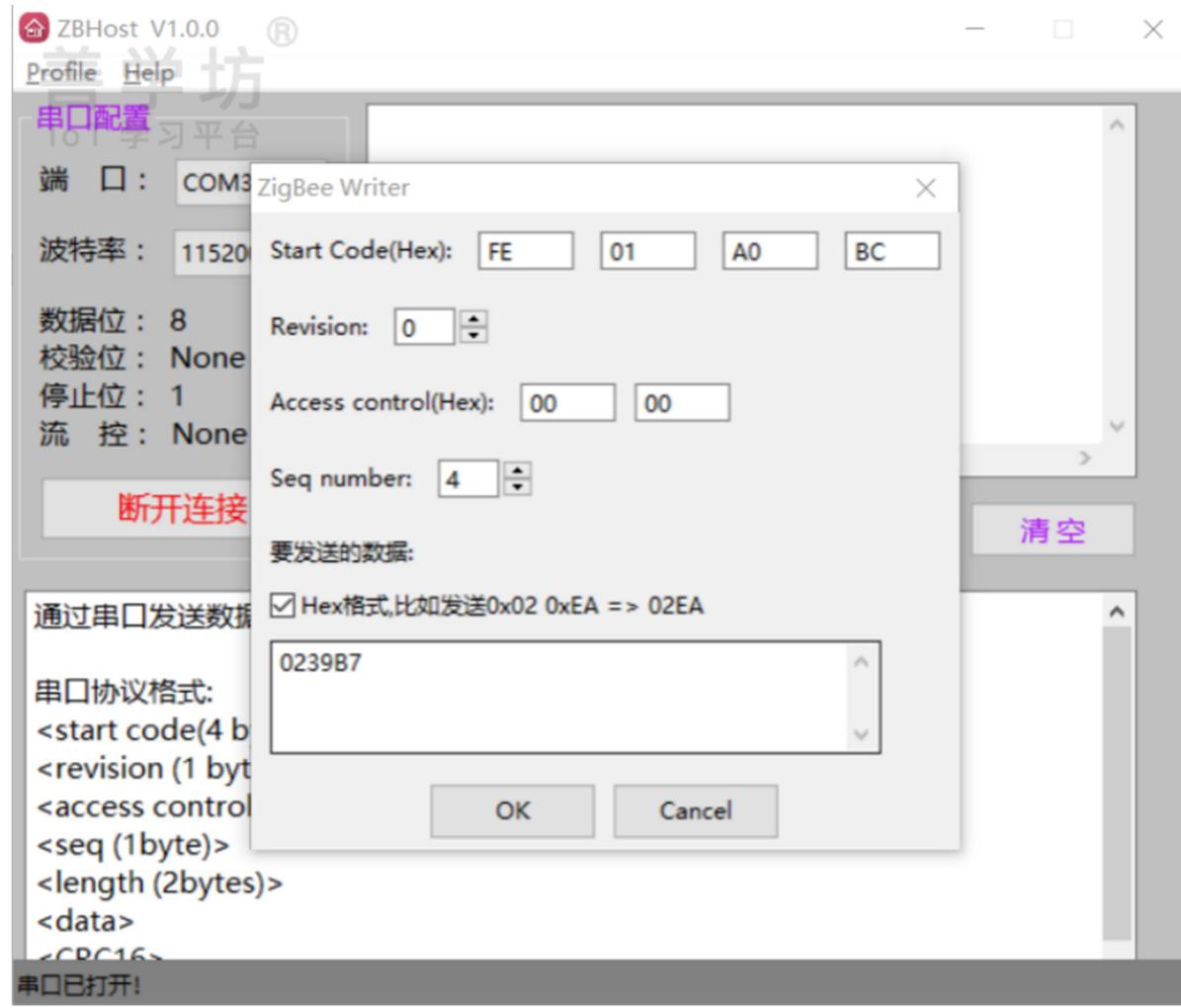
## 9.6.2. Debugging

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

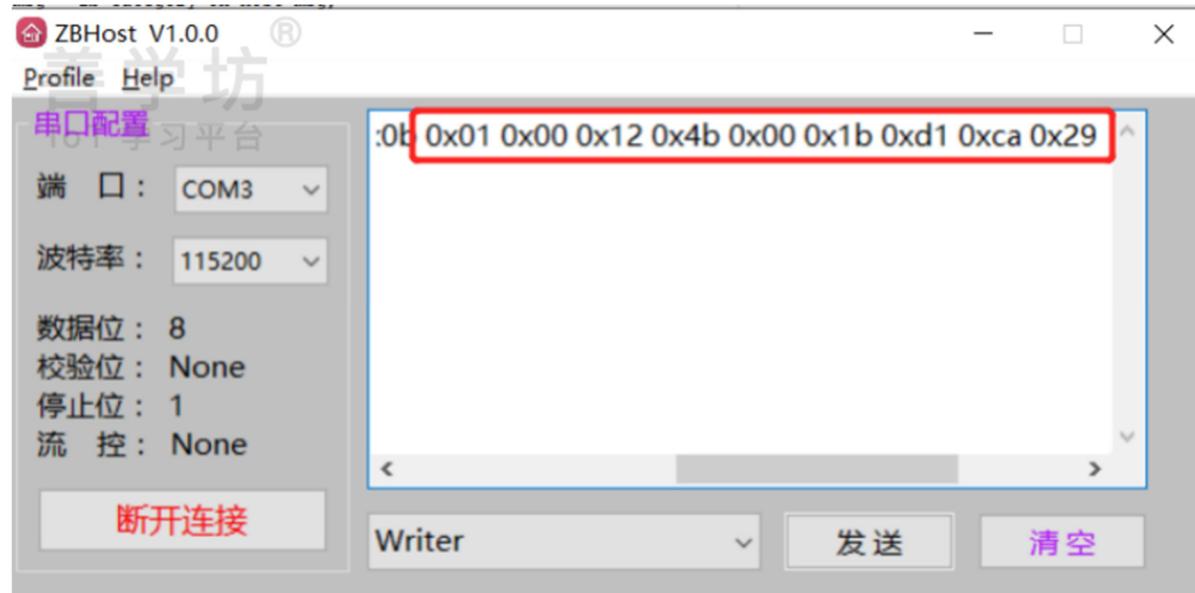
### Technical support instructions:

- Generally, self-study is the main method.
- You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
- Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

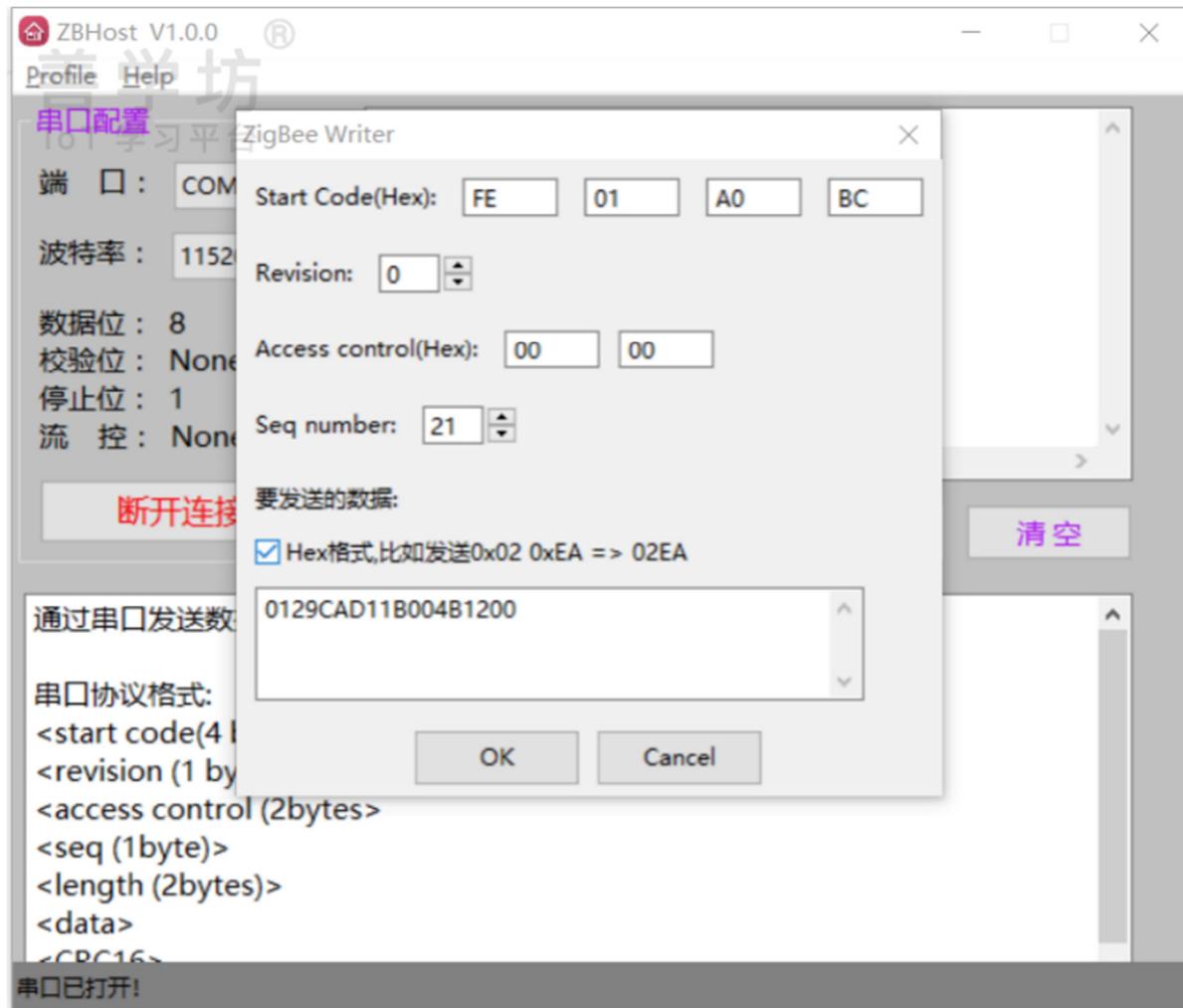
Get Mac address:



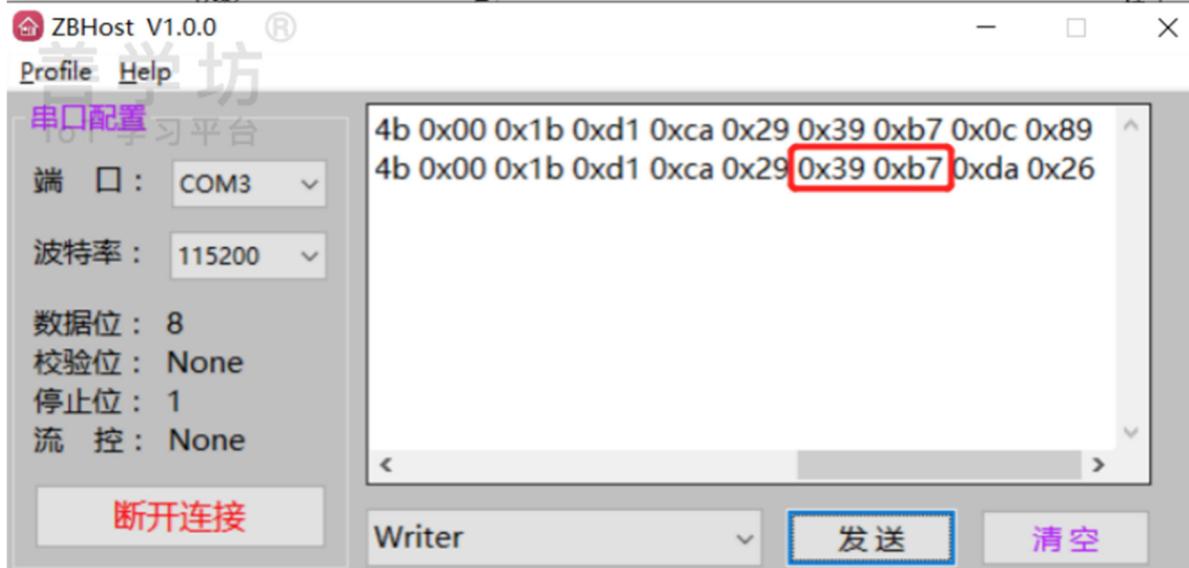
Successfully obtained MAC address returns:



Get the network address:



Successfully obtained the network address:



## 9.7. Chapter 6: Network Access Control and Whitelist

### 9.7.1. Basic content

#### 9.7.1.1. Network access control

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

Technical support instructions:

1. Generally, self-study is the main method.

2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>  
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

In the previous chapter, we added sub-devices to the coordinator for experimentation. The process of adding sub-devices can be called permitting network access. This process can be controlled. The API that controls whether the coordinator itself is allowed to join the network can be found in NLMEDE.h: **NLME\_PermitJoiningRequest**

The parameter of this function is a variable of type `uint8_t`. The value and meaning of this variable are as follows:

0: Disable permit join.

1 ~ 254: Allow to join for a period of 1 to 254 seconds.

255: Permit join is always allowed.

## 9.7.1.2. Whitelist

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.  
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>  
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

The concept of whitelist is mainly used to allow specific devices to join. If the coordinator detects that a device wants to join, it will first determine whether the device is in the whitelist. If not, it will not be allowed to join, which is similar to the function of a firewall. When a sub-device joins, it will pass the Join Indication primitive to the upper layer. When we receive the indication, we decide whether to allow the sub-device to join. This primitive is captured in ZDO, in the file ZDApp.c:

### ZDO\_JoinIndicationCB

```
3007
3008 ZStatus_t ZDO_JoinIndicationCB(uint16 ShortAddress, uint8 *ExtendedAddress,
3009                         uint8 CapabilityFlags, uint8 type)
3010
3011 (void)ExtendedAddress;
```

## 9.7.2. Function Encapsulation

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.  
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>  
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

We encapsulate the following interfaces in zbnwk:

```

11  typedef struct {
12      void (*set_panid_channel)(uint16_t panid, uint8_t channel);
13
14      void (*nwkaddr_req)(uint8_t *IEEEAddress);
15      void (*macaddr_req)(uint16_t nwkaddr);
16
17      void (*permit)(uint8_t timeout);
18      void (*allow_join)(uint8_t *IEEEAddress);
19      bool (*is_allow)(uint8_t *IEEEAddress);
20  } zb_network_t;
21
22  extern const zb_network_t *zbnwk_;
23

```

The contents of the package are as follows:

```

49  static void zb_nwk_permit(uint8_t timeout)
50  {
51      NLME_PermitJoiningRequest(timeout);
52  }
53
54  static void zb_nwk_allow_join(uint8_t *IEEEAddress)
55  {
56      memcpy(whitelist_item_, IEEEAddress, 8);
57  }
58
59  static bool zb_nwk_is_allow(uint8_t *IEEEAddress)
60  {
61 #ifdef WHITELIST_DISABLE
62     return true;
63 #endif
64
65     return memcmp(IEEEAddress, whitelist_item_, 8) == 0;
66 }

```

- zb\_nwk\_permit: Enable permit join
- zb\_nwk\_allow\_join: add whitelist
- zb\_nwk\_is\_allow: Check if it is in the whitelist

There is also a macro definition that can be enabled: WHITELIST\_DISABLE

If this macro definition is enabled, all devices are allowed to join. This is done through pre-compilation. Of course, we can also design it to be dynamically enabled/disabled.

Next is the definition of the serial port application protocol:

Action	Data
0x03	timeout
0x04	mac[7], mac[6].....mac[1], mac[0]

Processing in the classifier:

```

26 void zb_category_on_host_msg(uint8_t *payload, uint16_t size)
27 {
28     uint8_t action = payload[0];
29
30     switch (action) {
31     case 0x00: zbnwk_->set_panid_channel((uint16_t)payload[1] << 8 | payload[2], payload[3]); break;
32
33     case 0x01: zbnwk_->nwkaddr_req(&payload[1]); break;
34
35     case 0x02: zbnwk_->macaddr_req((uint16_t)payload[1] << 8 | payload[2]); break;
36
37     case 0x03: zbnwk_->permit(payload[1]); break;
38
39     case 0x04: zbnwk_->allow_join(&payload[1]); break;
40 }
41 }

```

Then make a judgment in ZDO\_JoinIndicationCB in ZDApp.c:

zb_category.c	zb_network.h	zb_network.c	zd_coordinator.c	zb_category.h	<b>ZDApp.c</b>	X
---------------	--------------	--------------	------------------	---------------	----------------	---

```

ZDO_JoinIndicationCB(uint16, uint8 *, uint8, uint8)
2991
2992 /**
2993  * @fn      ZDO_JoinIndicationCB
2994  *
2995  * @brief   This function allows the next higher layer of a
2996  *          coordinator to be notified of a remote join request.
2997  *
2998  * @param   ShortAddress - 16-bit address
2999  * @param   ExtendedAddress - IEEE (64-bit) address
3000  * @param   CapabilityFlags - Association Capability Flags
3001  * @param   type - of joining -
3002  *           NWK_ASSOC_JOIN
3003  *           NWK_ASSOC_REJOIN_UNSECURE
3004  *           NWK_ASSOC_REJOIN_SECURE
3005  *
3006  * @return  ZStatus_t
3007 */
3008 ZStatus_t ZDO_JoinIndicationCB(uint16 ShortAddress, uint8 *ExtendedAddress,
3009                                     uint8 CapabilityFlags, uint8 type)
3010 {
3011     (void)ExtendedAddress;
3012     //check if the device is leaving before responding to rejoin request
3013     if( osal_get_timeoutEx( ZDAppTaskID , ZDO_DEVICE_RESET) )
3014     {
3015         return ZFailure; // device leaving , hence do not allow rejoin
3016     }
3017
3018     /* Check whitelist! */
3019     if (type == NWK_ASSOC_JOIN && !zbnwk_->is_allow(ExtendedAddress)) return ZFailure;
3020 }

```

The meaning of this code is to judge whether to access the network and return failure when it is not in the whitelist.

## 9.7.3. Program Debugging

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support:

1. Customer service provides simple technical support, mainly self-study.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answers. Technical answers are hard, please understand!

After compiling, burn the program, and then we turn on the network access permission function:

## 串口配置

端 口 : COM3 ZigBee Writer

波特率 : 11520

数据位 : 8

校验位 : None

停止位 : 1

流 控 : None

断开连接

通过串口发送数据

串口协议格式:

&lt;start code(4 b)

&lt;revision (1 byt)

&lt;access control

&lt;seq (1byte)&gt;

&lt;length (2bytes)&gt;

&lt;data&gt;

&lt;CRC16&gt;

串口已打开!

Start Code(Hex): FE 01 A0 BC

Revision: 0

Access control(Hex): 00 00

Seq number: 19

要发送的数据:

 Hex格式,比如发送0x02 0xEA => 02EA

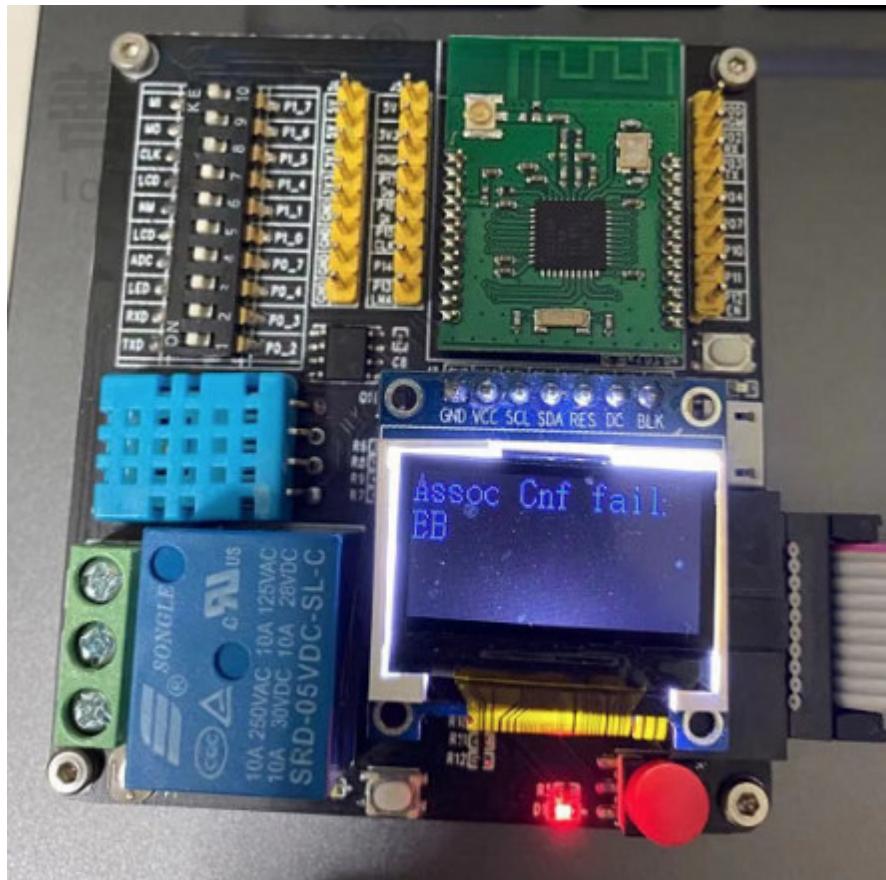
03F0

清空

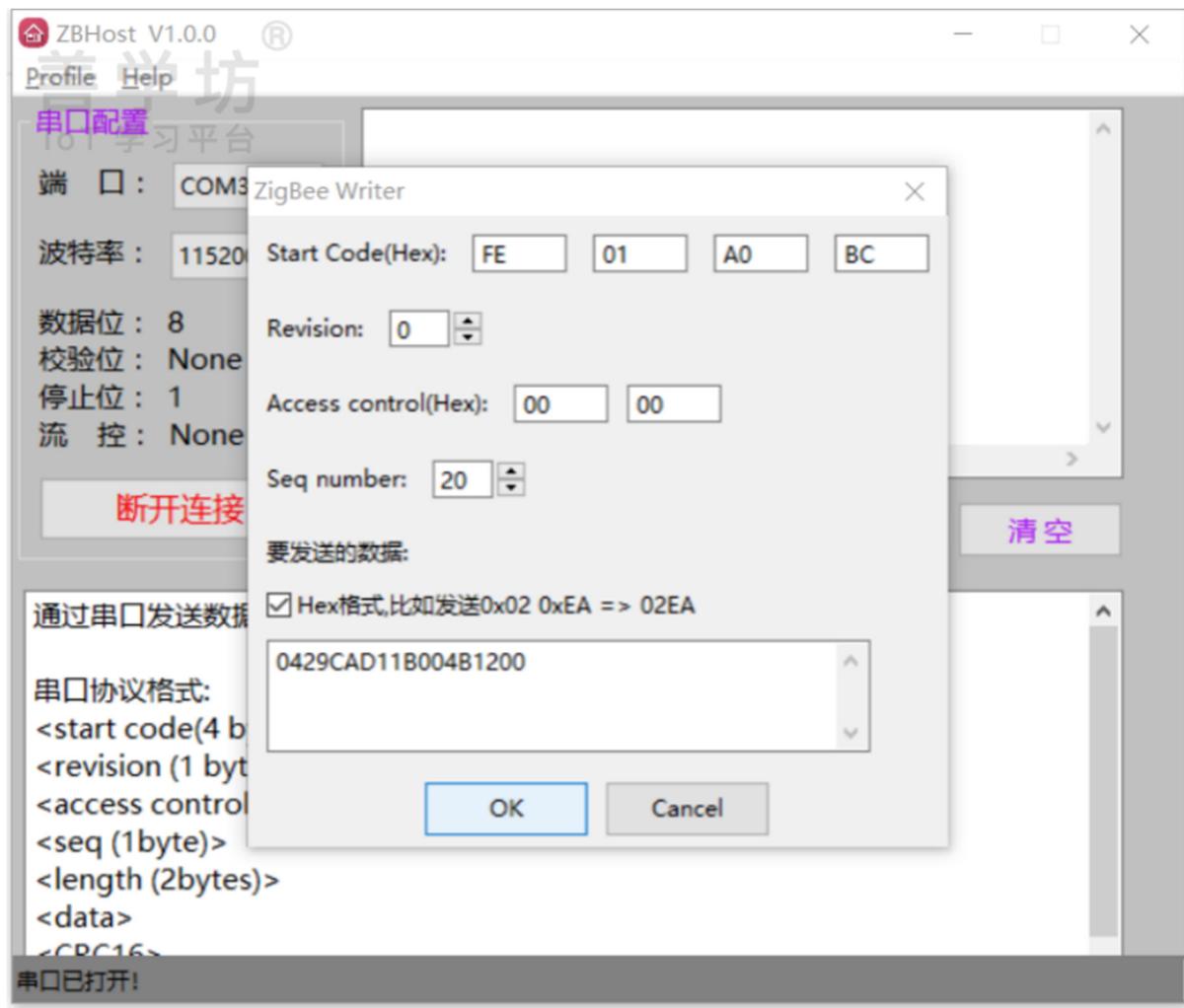
OK

Cancel

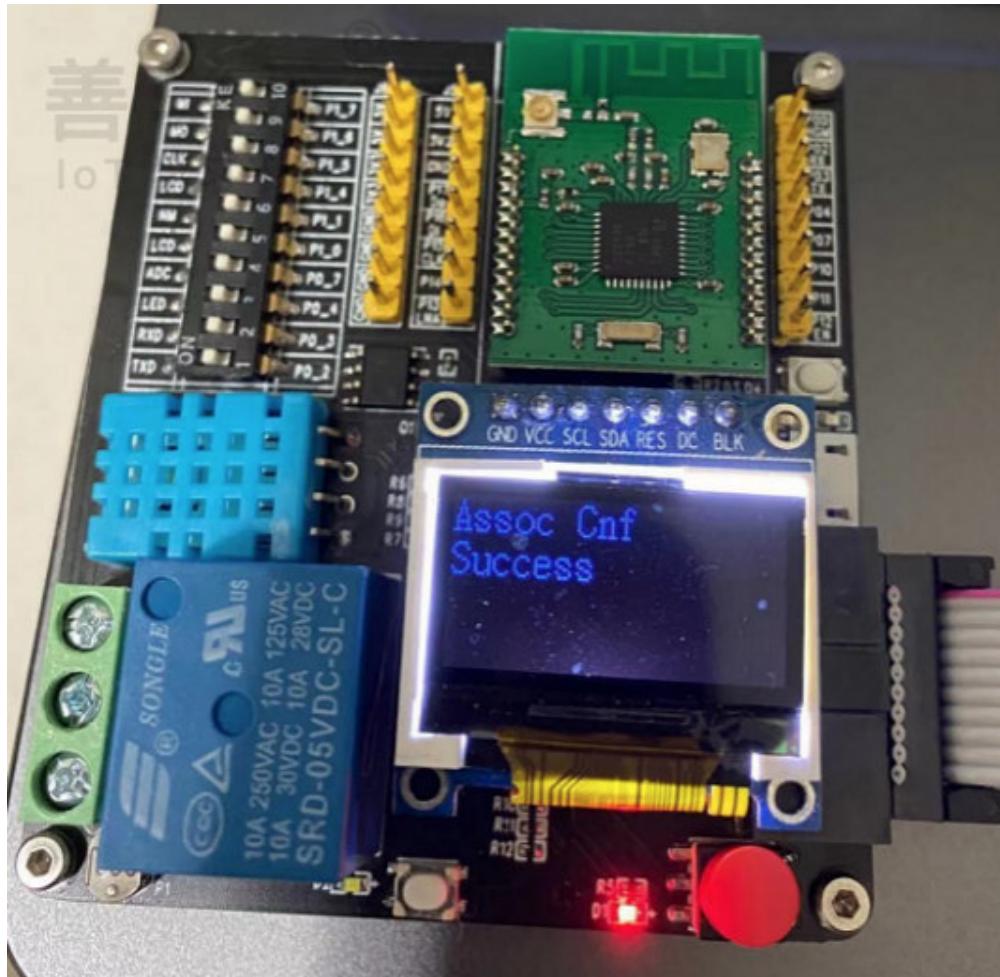
Since there is no whitelist, the child device fails to join:



Now add the whitelist:



Reset the sub-device and add it again:



Added successfully:



## 9.8. Chapter 7: Coordinator Partition Storage Management

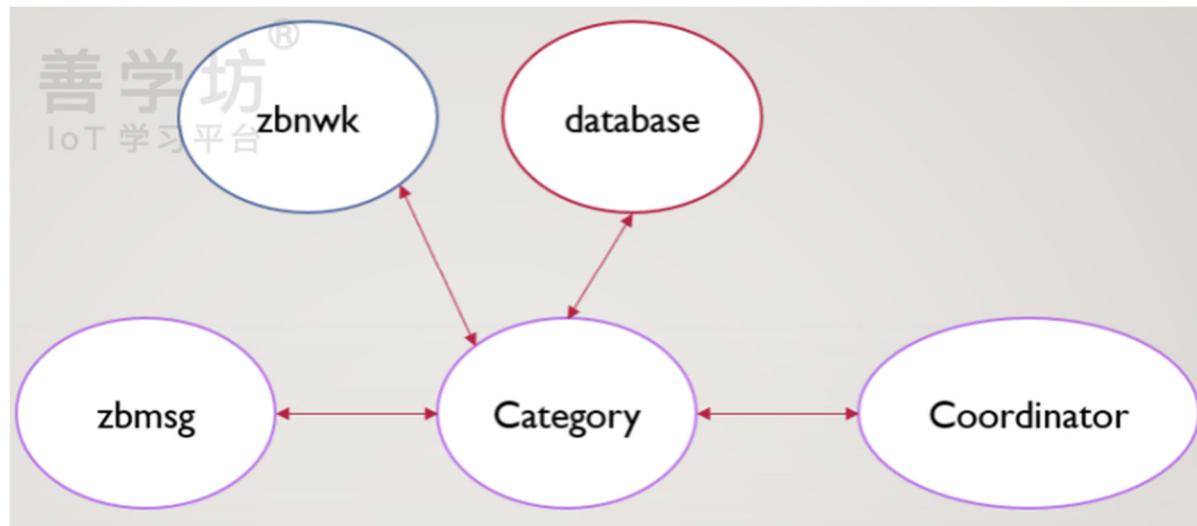
### 9.8.1. Software Framework

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Our board is equipped with an 8Mbit external Flash. We can perform partition management and divide a space for database storage. We can make adjustments on the software framework as follows:



Source code files: zb\_database.h and zb\_database.c

The interface is as follows:

```
11  typedef struct {
12      void (*init) (void);
13
14      bool (*write) (uint8_t *buf, uint8_t len);
15      bool (*read) (uint8_t *buf, uint8_t len);
16  } zb_database_t;
17
18  extern const zb_database_t *zbdb_;
```

- init: Initialize the hardware and database logic.
- write: write a piece of data
- read: read a piece of data

Since the database is mounted under the classifier, the initialization is also called by the classifier.

## 9.8.2. Application

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Definition of serial port application protocol:

Action	Data
0x05	length, data
0x06	length

- Action 0x05 means writing data
- Action 0x06 means reading data

Processing in the classifier:

```
34 void zb_category_on_host_msg(uint8_t *payload, uint16_t size)
35 {
36     uint8_t action = payload[0];
37
38     switch (action) {
39     case 0x00: zbnwk_->set_panid_channel((uint16_t)payload[1] << 8 | payload[2], payload[3]); break;
40
41     case 0x01: zbnwk_->nwkaddr_req(&payload[1]); break;
42
43     case 0x02: zbnwk_->macaddr_req((uint16_t)payload[1] << 8 | payload[2]); break;
44
45     case 0x03: zbnwk_->permit(payload[1]); break;
46
47     case 0x04: zbnwk_->allow_join(&payload[1]); break;
48
49     case 0x05: zbdb_->write(&payload[2], payload[1]); break;
50
51     case 0x06: zb_category_read_db(payload[1]); break;
52     }
53 }
54 }
```

zb\_category\_read\_db content:

```
77 void zb_category_read_db(uint8_t len)
78 {
79     uint8_t payload[255] = { 0x02 };
80
81     zbdb_->read(&payload[1], len);
82
83     zbmmsg_->send(payload, len + 1);
84 }
```

The logic of applying the above is relatively simple. After receiving the command, the data is given to zbdb\_ for processing. If it is a read, a response message is returned. Note that the response header is 0x02.

Note: Our external Flash uses SPI for communication, which occupies the SPI1 port of the main controller. Since the LCD display is also driven by SPI, we need to turn off the LCD function in this chapter.

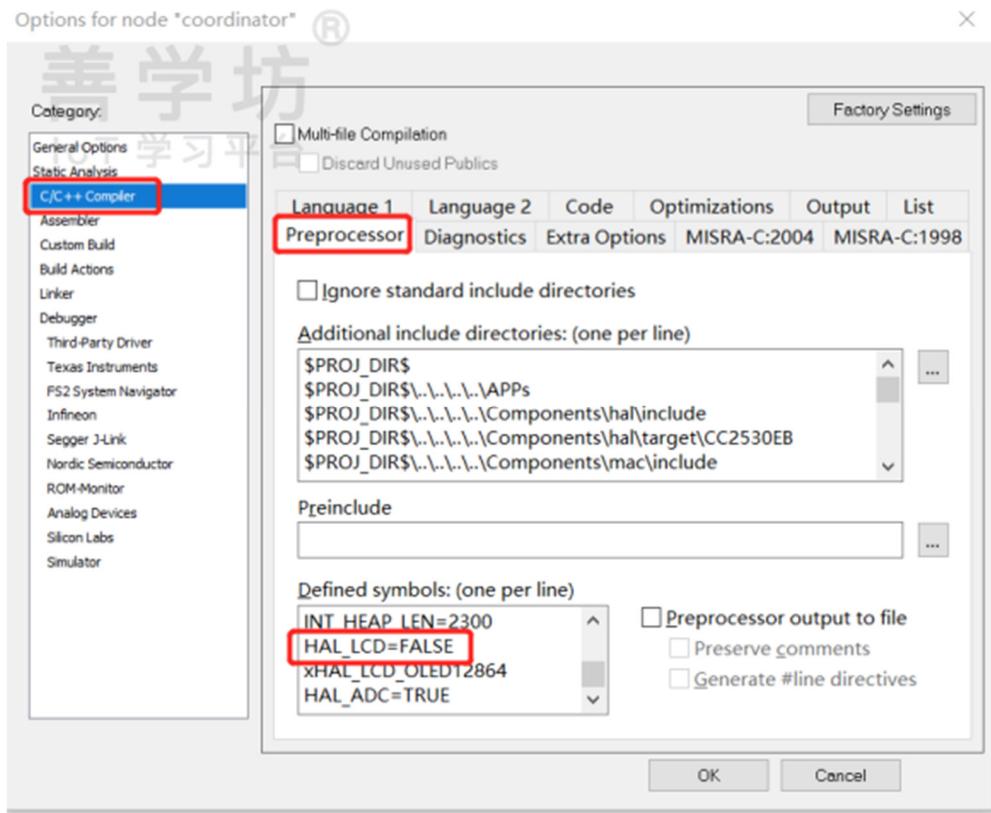
## 9.8.3. Debugging

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

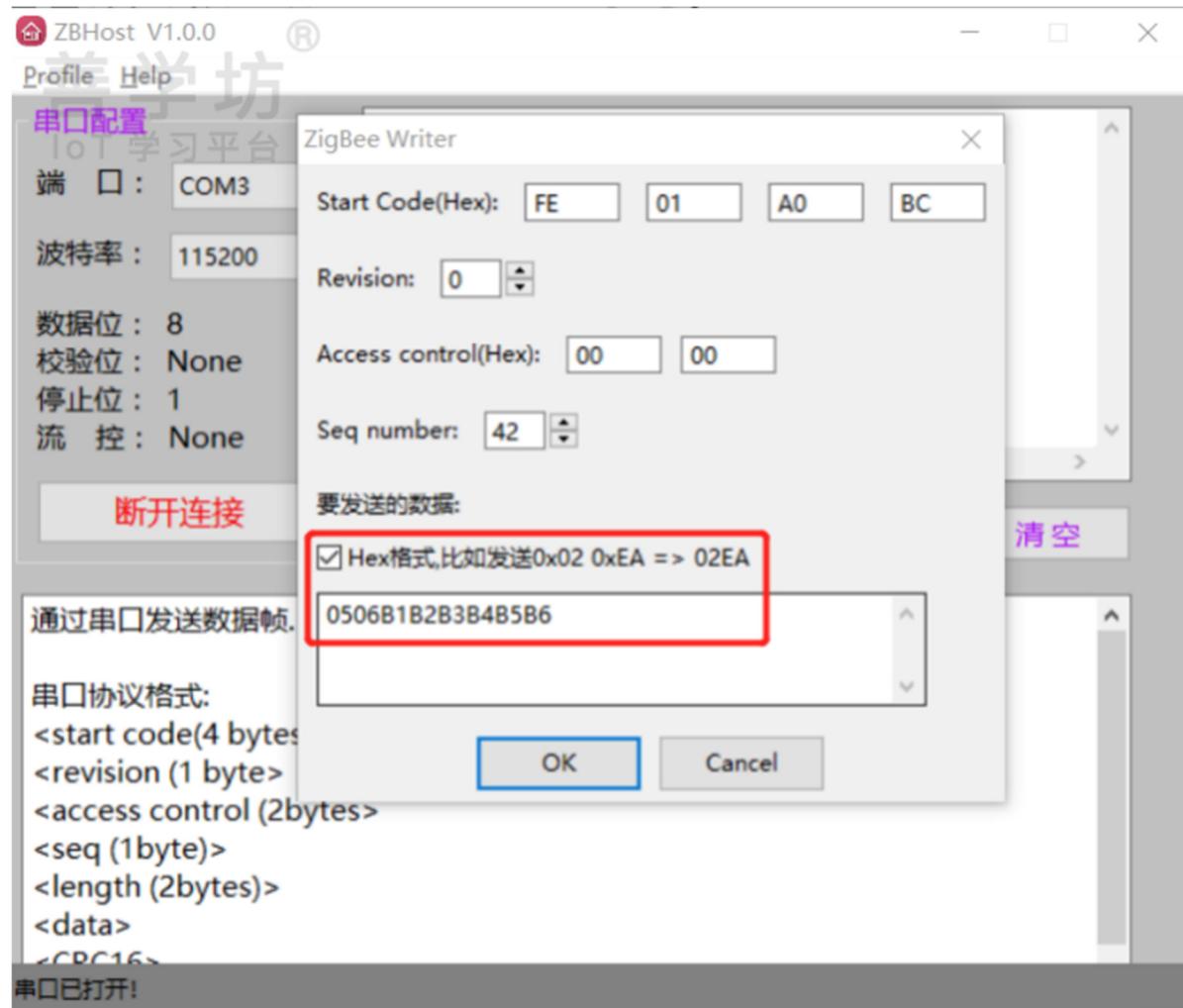
### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

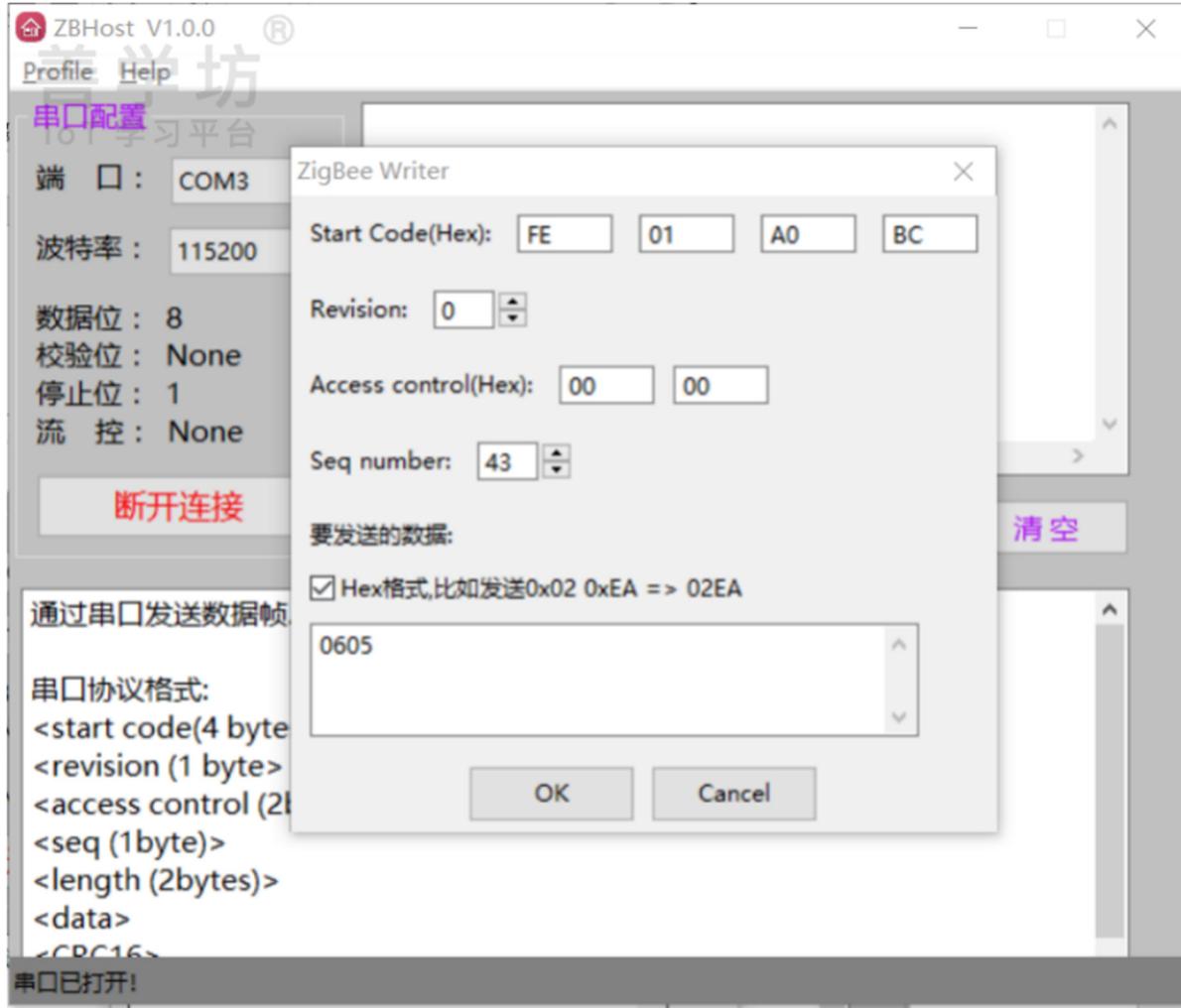
First, turn off the LCD function, then compile the entire project and burn it to the board:



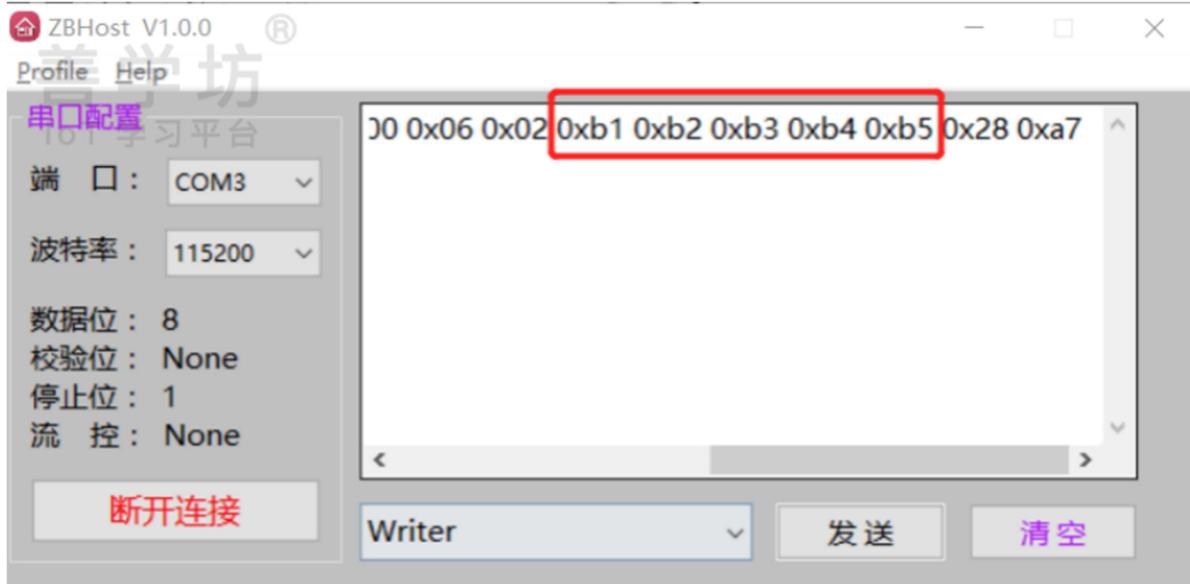
Then we write 6 bytes through the host computer:



Then read 5 more bytes to see if they are correct:



The returned content is correct!



## 9.9. ZigBee 2 WiFi - Based on ESP8266

### 9.9.1. Use cloud server

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>

3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

## Remote Server Description

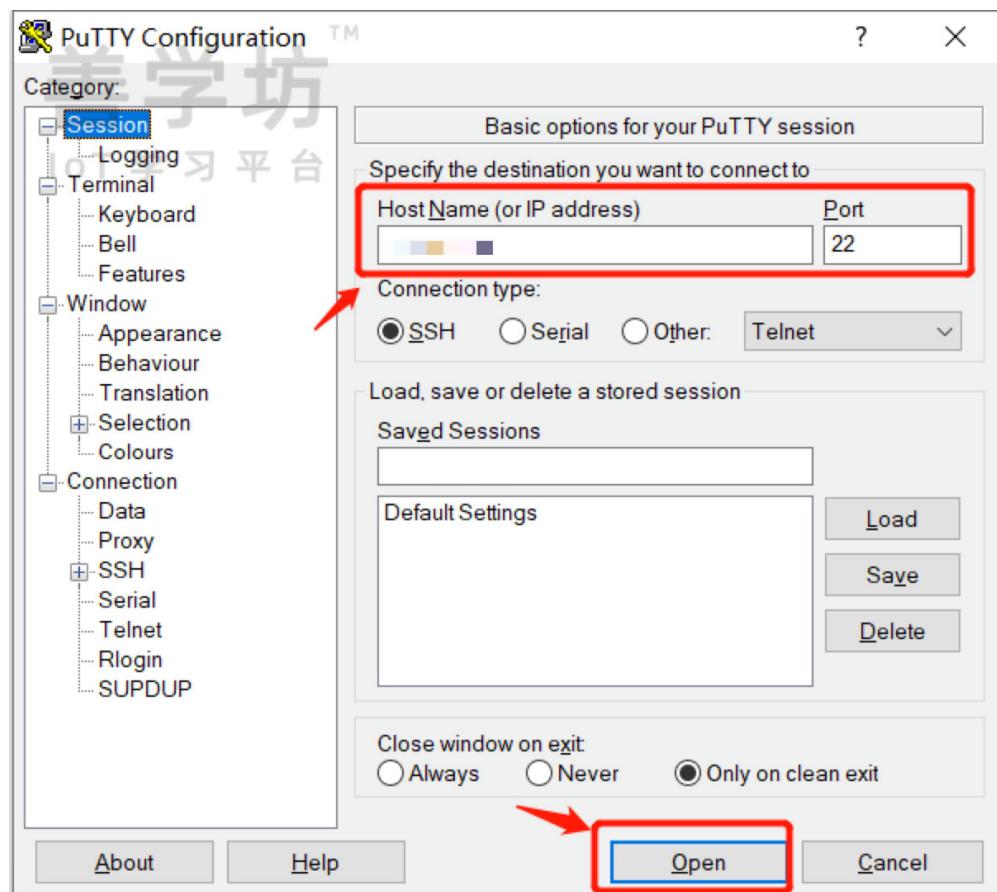
The author provides readers with a free remote service, which readers can log in and use through PuTTY. This greatly facilitates readers to develop and debug the communication between the development board and the server (mobile phone app).

### Login to remote server

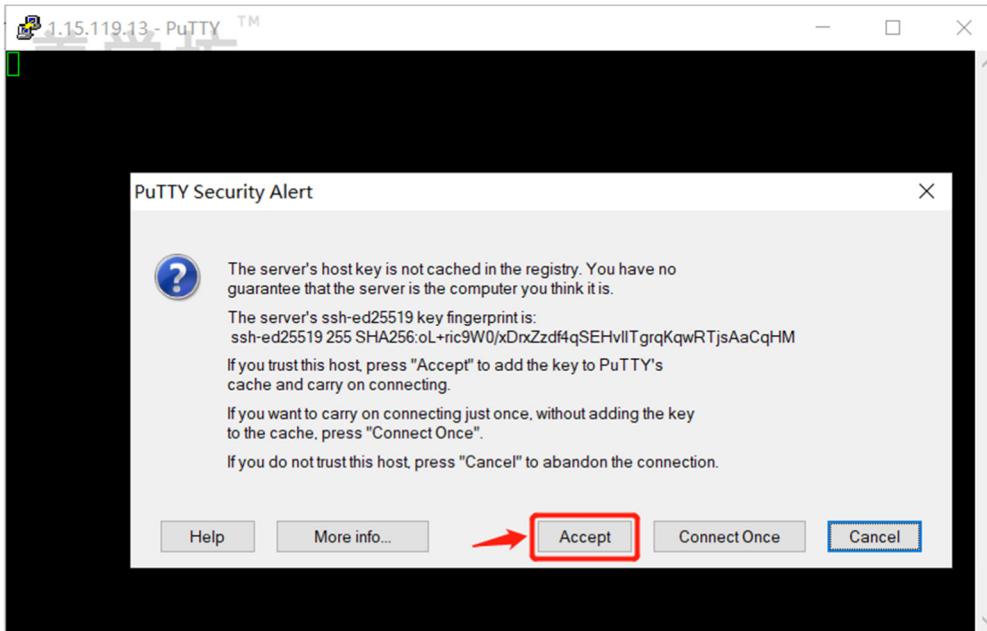
- Download and install PuTTY: <https://zhuanlan.zhihu.com/p/535091460>
- Run the installed PuTTY



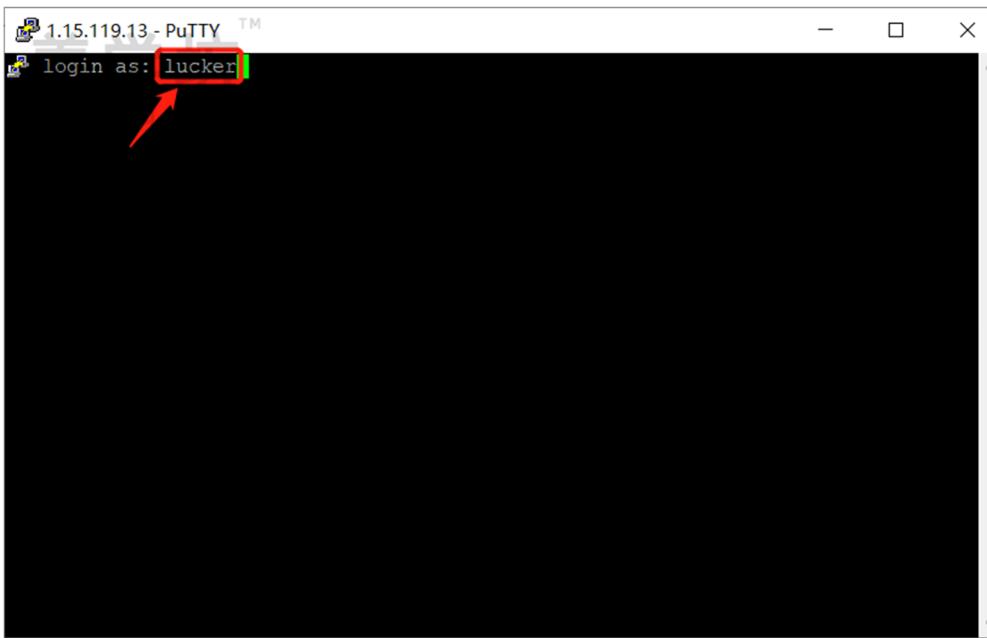
- Enter the remote server address in Host Name: **1.15.27.206**, enter 22 in Port, and then click the Open button, as shown in the figure.



- If the following dialog box pops up, be sure to click Accept



- Enter the username luckyr and press Enter, as shown in the figure.



- Enter the password (note that no characters are displayed when you enter the password): 12345678, then press Enter.



- Login to the remote server is successful, as shown in the figure.

```
1.15.119.13 - PuTTY TM
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Aug 22 00:10:37 2021 from 59.41.245.254
$
```

- Enter the ls command to view the programs (files) in the current directory, and enter the cat readme.txt command to view related instructions.

```
1.15.119.13 - PuTTY TM
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Aug 22 00:19:07 2021 from 59.41.245.254
$ ls
killall  mosquitto  readme.txt  tcpserver  udpserver
$ cat readme.txt
MQTT PORT : 1883
TCP PORT : 12300
UDP PORT : 12301
$
```

## Related programs and port descriptions

### Program Description:

- mosquitto: MQTT server program
- tcpserver: TCP server program
- udpserver: UDP server
- killall: Close all programs. This program can only be called if it is occupied by other programs.

### Port Description:

1883: MQTT native port

12300: TCP server port

12301: UDP server port

## 9.9.2. Source code description and testing

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

## Important configuration

### WiFi configuration

Open the project that comes with this lesson. The directory where the project comes with is: "ZigBee3.0 Development Guide" supporting resources (**additional part**)/Extracurricular: Advanced elective/ZigBee2Wifi\_ESP8266, then open the zb\_common.h file, which contains the WiFi name and password definitions, as shown in the figure.

```
#ifndef __ZB_COMMON_H__
#define __ZB_COMMON_H__

#ifndef __cplusplus
extern "C"
{
#endif

typedef unsigned char uint8_t;
typedef unsigned int uint16_t;

/* Device/User Info */
#define WIFI_NAME           "Wifi-2G4"
#define WIFI_PASSWD         "1234567890zxcv"

/* Endpoints */
#define ENDPOINT_ZHA        0x08
#define ENDPOINT_ZLL        0x0B
#define ENDPOINT_PRIVATE    120

/* Private clusters */
#define PRIV_CID_MESSAGE   0x0000
#define PRIV_CID_SETON     0x0001
#define PRIV_CID_SETOFF    0x0002

/* Events */
#define ZB_USER_EVENT       0x0001
#define ZB_USER_EVENT_PERIOD 1000

uint16_t crc16_x25(const uint8_t *buf, uint16_t buf_len);

uint8_t zb_generate_seqnum(void);

#endif /* __ZB_COMMON_H__ */
```

Readers need to change the WiFi name and password here to the WiFi name and password at home so that the development kit can access the WiFi router at home.

## Source code explanation

We use AT commands to drive ESP8266; the key program is in zb\_port.c; there is a timer in zb\_port.c:

```
void zb_port_on_timer(uint16_t ms)
{
#if ZG_BUILD_COORDINATOR_TYPE
    zb_port_on_coordinator_timer();
#elif ZG_BUILD_RTRONLY_TYPE
```

```
#else  
#endif  
}
```

This timer runs every 1 second. You can see that we have done something for the coordinator: calling `zb_port_on_coordinator_timer()`:

```
void zb_port_on_coordinator_timer(void)  
{  
    static uint8_t cnt = 0;  
    if (++cnt < 6) return;  
  
    cnt = 3;  
  
    static uint8_t connWifi = 0;  
    static uint8_t connTcp = 0;  
  
    char buf[50] = { 0 };  
  
    switch(espstep) {  
        /* AT test */  
        case 0:  
            zbmmsg_->send("AT\r\n", 4);  
            HallLcdWriteString("step-1", 3);  
            break;  
  
        /* Setting */  
        case 1:  
            zbmmsg_->send("AT+CWMODE_DEF=1\r\n", 17);  
            HallLcdWriteString("step-2", 3);  
            break;  
  
        /* Connect WiFi */  
        case 2:  
            if (connWifi == 0) {  
                sprintf(buf, "AT+CWJAP_DEF=\"%s\", \"%s\"\r\n", WIFI_NAME, WIFI_PASSWD);  
                connWifi = 1;  
                HallLcdWriteString("step-3", 3);  
                zbmmsg_->send((uint8_t*)buf, strlen(buf));  
            }  
            break;  
  
        /* Config Mode */  
        case 3:  
            HallLcdWriteString("step-4", 3);  
            zbmmsg_->send("AT+CIPMUX=0\r\n", 13);
```

```

break;

/* Connect to TCP server */

case 4:

#define TCP_SERVER "AT+CIPSTART=\"TCP\",\"1.15.27.206\",12300\r\n"

if (connTcp == 0) {

    connTcp = 1;

    HalLcdWriteString("step-5", 3);

    zbmmsg_->send((uint8_t *)TCP_SERVER, strlen(TCP_SERVER));

}

break;

/* Send message to TCP server */

case 5:

#define TCP_MESSAGE "Hello World!"

sprintf(buf, "AT+CIPSEND=%d\r\n", strlen(TCP_MESSAGE));

zbmmsg_->send((uint8_t *)buf, strlen(buf));

break;

case 6:

zbmmsg_->send((uint8_t *)TCP_MESSAGE, strlen(TCP_MESSAGE));

break;

/* Error */

case 7:

connTcp = 0;

espstep = 3;

break;

default: break;

}

}

```

Here we are actually driving ESP8266 to connect to WiFi. After the connection is successful, TCP will be opened to send data to the server. After we send AT commands to ESP8266 through the serial port, we need to wait for the module's response (serial port reception) to tell us whether the AT command is executed successfully:

```

void zb_port_on_coordinator_uart(const uint8_t *data, uint8_t len)

{

switch(espstep) {

case 0:

if (strstr((char *)data, "OK") != NULL) { espstep++; HalLcdWriteString("ok-1", 4); }

else if (strstr((char *)data, "WIFI CONNECTED") != NULL) { espstep = 3; HalLcdWriteString("ok-wifi", 4); }

break;

case 1:

if (strstr((char *)data, "OK") != NULL) {

```

```

espstep++;

HalLcdWriteString("ok-2", 4);

}

break;

case 2:

if (strstr((char *)data, "IP") != NULL || strstr((char *)data, "OK") != NULL) {

espstep++;

HalLcdWriteString("ok-3", 4);

}

break;

case 3:

if (strstr((char *)data, "OK") != NULL) {

espstep++;

HalLcdWriteString("ok-4", 4);

}

break;

case 4:

if (strstr((char *)data, "OK") != NULL) {

espstep++;

HalLcdWriteString("ok-5", 4);

}

break;

case 5:

if (strstr((char *)data, ">") != NULL) {

espstep++;

HalLcdWriteString("ok-6", 4);

}

break;

case 6:

if (strstr((char *)data, "CLOSED") != NULL) {

espstep++;

HalLcdWriteString("err-1", 4);

}

break;

default: break;
}
}

```

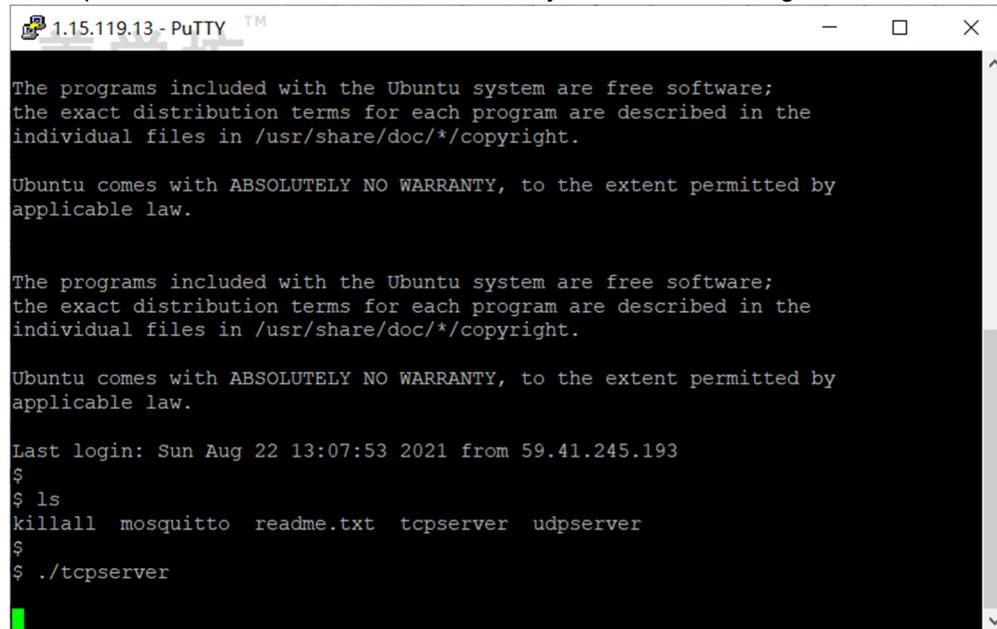
Obviously, when we send an AT command, if we receive a normal response from ESP8266, we will send the next AT command, thus realizing an asynchronous method.

# Program Testing

(1) Log in to the remote server through PuTTY and enter the following command to run the tcpserver service:

```
./tcpserver
```

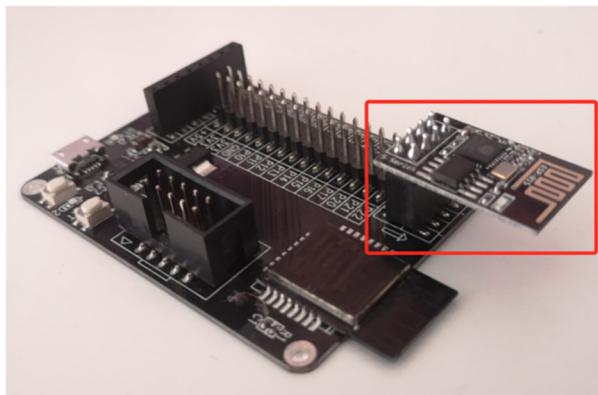
The tcpserver service was started successfully, as shown in the figure.



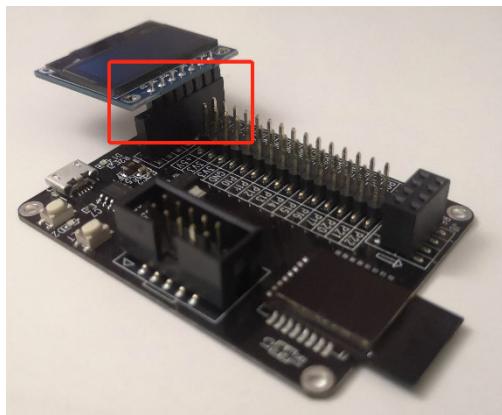
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
Last login: Sun Aug 22 13:07:53 2021 from 59.41.245.193  
\$  
\$ ls  
killall mosquitto readme.txt tcpserver udpserver  
\$  
\$ ./tcpserver

(3) If using a Mini board for testing, then:

- Insert the ESP8266 WiFi module into the ZigBee Mini board as shown in the picture.



- Insert the 0.96 OLED display into the ZigBee Mini board as shown in the picture.



(4) If you use a standard board for testing, after inserting the ESP8266 WiFi module and the 0.96 OLED display, you also need to configure the dip switches as shown in the figure. That is, the 1st, 2nd, 3rd, and 4th positions are set to the right, and the 5th,

6th, 8th, 9th, and 10th positions are set to the left. The 7th position does not need to be configured, as shown in the figure.

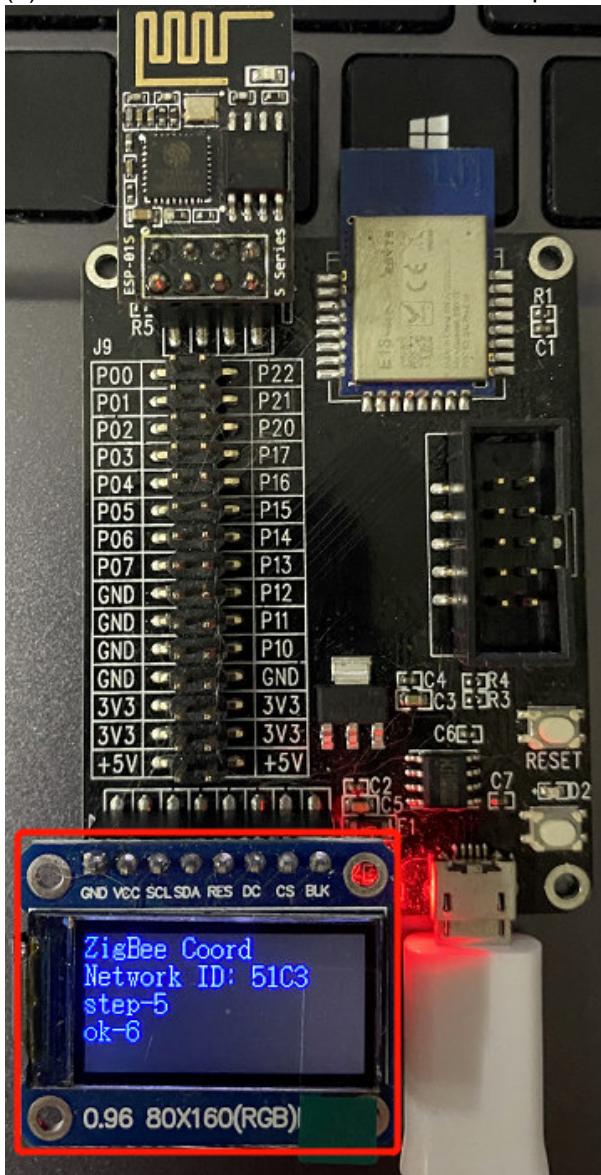


(5) Select the Coordinator role to compile the accompanying experimental code and download it to the development board.

Note:

- (1) When using the emulator to burn the program, you need to use a Micro USB cable to power the development board.
- (2) When using CC2530 to drive the WiFi module using AT commands, it is recommended to unplug the emulator to avoid causing unknown problems.

(6) You can see the AT command execution process on the display screen, as shown in the figure.



And the message is also received on the server, as shown in the figure.

The screenshot shows a PuTTY terminal window titled "1.15.119.13 - PuTTY". The terminal displays several lines of text, likely representing a ZigBee message exchange. The text includes copyright notices for the Ubuntu system, a "Last login" timestamp, directory listing commands, and a message indicating a connection and a "Hello World!" packet. The terminal has a dark background with white text and a standard window title bar.

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Aug 22 14:03:56 2021 from 59.41.245.193
$ 
$ ls
killall mosquitto readme.txt tcpserver udpserver
$ ./tcpserver

linker on_connected, fd=7
packet=Hello World!, len=12
```

## 9.10. Capturing and analyzing ZigBee wireless messages

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

### Capturing and analyzing ZigBee wireless messages

Zhihu tutorial: <https://zhuanlan.zhihu.com/p/489493842>

## 9.11. Connect to Xiaomi Aqara smart socket and temperature and humidity sensor

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz?p=23>

Found "Extracurricular Chapter: Connecting Xiaomi Smart Socket and Temperature and Humidity Sensor"

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

### Introduction

This lesson will explain how to connect Xiaomi's Aqara series smart sockets and temperature and humidity sensors to the ZigBee 3.0 development board. Readers can follow the examples to connect more ZigBee 3.0 smart devices, such as Tuya smart sensor devices, Philips ZigBee 3.0 bulbs, etc.

- Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz?p=23>
- Found "Extracurricular Chapter: Connecting Xiaomi Smart Socket and Temperature and Humidity Sensor"

# Hardware

- ZigBee Mini board or standard board
- Aqara smart socket: <https://aqara.com/cn/productDetail/d37>. Readers can also search for keywords such as "AQARA socket T1" and "Green Rice socket T1" in the Xianyu App.
- Aqara temperature and humidity sensor: <https://item.jd.com/5273037.html>

## 9.12. NV Application of Z-Stack

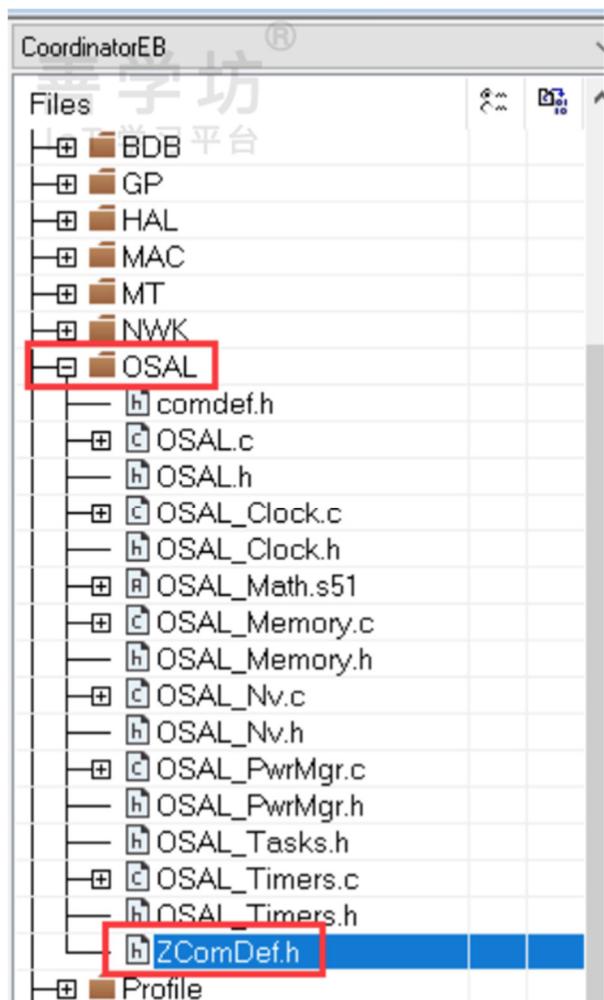
### 9.12.1. Introduction to NV

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

NV is non-volatile memory. Simply put, Z-Stack takes a piece of FLASH space inside the chip (such as CC2530) to store data. Part of this space is used by Z-Stack. The usage can be checked in the file ZComDef.h:



What we need to focus on is how much space ZStack reserves for users. The corresponding definition can also be found in the file ZComDef.h:

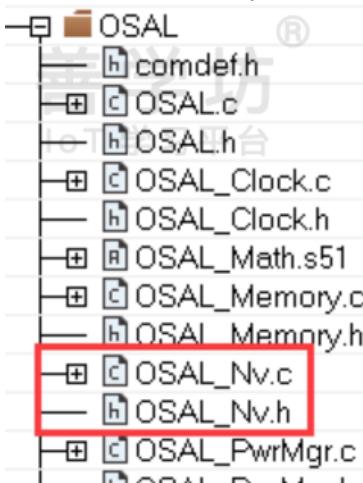
```
277 #define ZCD_NV_PROXY_TABLE_END 0x033F
278
279 // NV Items Reserved for applications (user applications)
280 // 0x0401 ?0xFFFF
281
282
```

That is, addresses from 0x0401 to 0xFFFF are reserved for users. If users need to use them, they can define them here. For example, if we want to use the NV space at address 0x0401 (each address can store one byte of data):

```
279 // NV Items Reserved for applications (user applications)
280 // 0x0401 ?0xFFFF
281 #define USER_NV_TEST 0x0401
```

## Operate NV files and API

There are several important APIs in OSAL\_Nv.h:



```
1./*
2. * Initialize an item in NV
3. */
4.extern uint8 osal_nv_item_init( uint16 id, uint16 len, void *buf );
5.
6./*
7. * Read an NV attribute
8. */
9.extern uint8 osal_nv_read(uint16 id,uint16 offset, uint16 len, void *buf);
10.*/
11. * Write an NV attribute
12. */
13.extern uint8 osal_nv_write(uint16 id,uint16 offset,uint16 len, void *buf);
```

The first API is used to initialize the NV at the specified address. For example, we want to initialize the space at address 0x0401:

```
1.osal_nv_item_init(0x0401, 1, NULL); // 参数len为1表示一个字节,
2.                                // 如果len不为1表示起始地
3.                                // 址后连续len个字节的空间
```

The second and third APIs are used to read/write the specified address NV, where the parameter id indicates the starting address, offset indicates how many bytes the starting address needs to be offset to as the address to be operated, and len indicates the number of bytes to be operated. For example, I want to write and read the address 0x0401 respectively:

```
1.uint8 writeVal;
2.uint8 readVal;
3.
4.*/
5.* Write, USER_NV_TEST -> 0x0401
6.*/
7.osal_nv_write( USER_NV_TEST, 0, 1, &writeVal ); // 写
8.
9.*/
10.* Read, USER_NV_TEST -> 0x0401
11.*/
12.osal_nv_read( USER_NV_TEST, 0, 1, &readVal); // 读
```

When using NV, you must initialize it before you can perform read/write operations. The space at the address specified by NV only needs to be initialized once. Calling the initialization function to the address later will not work. You will understand it later with examples!

## 9.12.2. NV reading and writing

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Let's design an experiment: define a variable. When a key is pressed, write the variable into NV and then read it out from NV. If the written and read values are the same, the operation is successful! Since the key function is used, we put the program in the key processing function:

```
1.static void zclSampleSw_HandleKeys( byte shift, byte keys )
2.{ 
3. UI_MainStateMachine(keys);
4.
5. if(keys & HAL_KEY_SW_6)
6. {
7.     uint8 readVal; // 存储从NV读取到的数据
8.     static uint8 writeVal = 0; // 要写入的变量
9.
10.    char readValStr[30];
11.    char writeValStr[30];
12.
13.    // init USER_NV_TEST
14.    osal_nv_item_init(USER_NV_TEST, 1, NULL); // 初始化只会生效一次
15.
```

```

16. // write

17. sprintf(writeValStr, "Write: %d", writeVal); // 写入数值字符串

18. osal_nv_write( USER_NV_TEST, 0, 1, &writeVal ); // 写NV

19.

20. writeVal++; // 每次写入后变量加1

21.

22. // read

23. osal_nv_read( USER_NV_TEST, 0, 1, &readVal ); // 读NV

24. sprintf(readValStr, "Read: %d", readVal); // 读取数值字符串

25.

26. // lcd show

27. HalLcdWriteString(writeValStr, HAL_LCD_LINE_1); // 显示写入值

28. HalLcdWriteString(readValStr, HAL_LCD_LINE_2); // 显示读取值

29.

30. /* 除了显示到显示屏上面之外，把数据也显示到串口助手上！ */

31. // uart show

32. HalUARTWrite( HAL_UART_PORT_0, (uint8 *)writeValStr, osal_strlen(writeValStr));

33. HalUARTWrite( HAL_UART_PORT_0, "\r\n", 2);

34. HalUARTWrite( HAL_UART_PORT_0, (uint8 *)readValStr, osal_strlen(readValStr));

35. HalUARTWrite( HAL_UART_PORT_0, "\r\n", 2);

36.

37. // led indication

38. HalLedSet( HAL_LED_1, HAL_LED_MODE_TOGGLE); // 指示灯

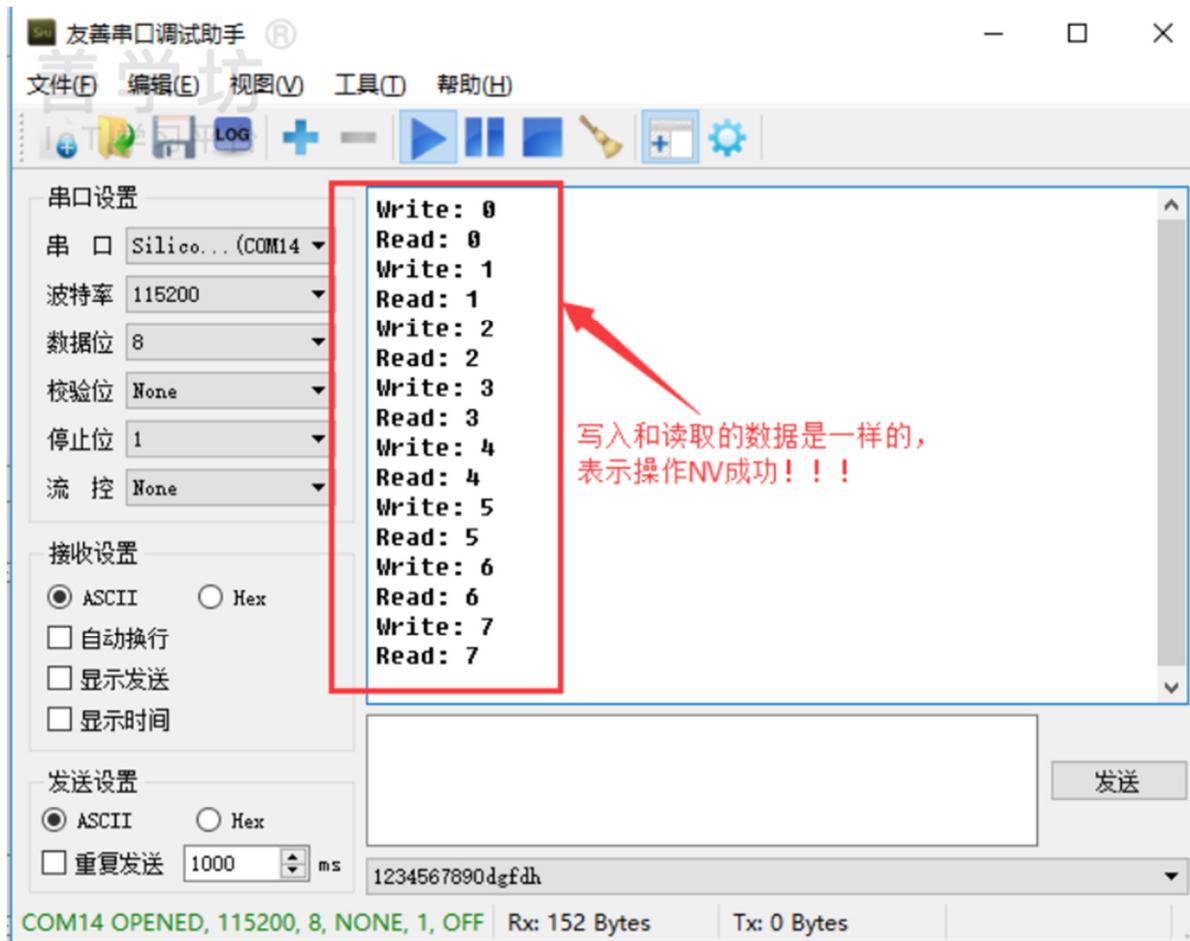
39. }

40.

```

## Simulation debugging

After compiling the project, burn the program to the development board, connect the development board and the computer via a USB cable, and open the serial port assistant. When we press the button, we can see the data written to NV and the data read back on the display and serial port:



## 9.13. HAL-based external FLASH application

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

### Learning objectives for this chapter

- Porting the M25PE80 driver to the protocol stack
- Mastering the use of M25PE80 driver

### Driver Preparation

We create a directory HW\_XNV in the HAL (Hardware Adaptation Layer) of the protocol stack to store the external FLASH driver:

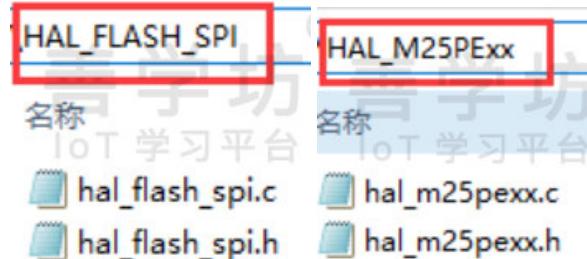
Z-Stack 3.0.1\Components\hal\target\CC2530EB

名称	修改
Common	外部FLASH驱动程序!
HW_LCD	201
HW_XNV	201
SPI	201
hal_uart_dma.c	201

Under the directory HW\_XNV:



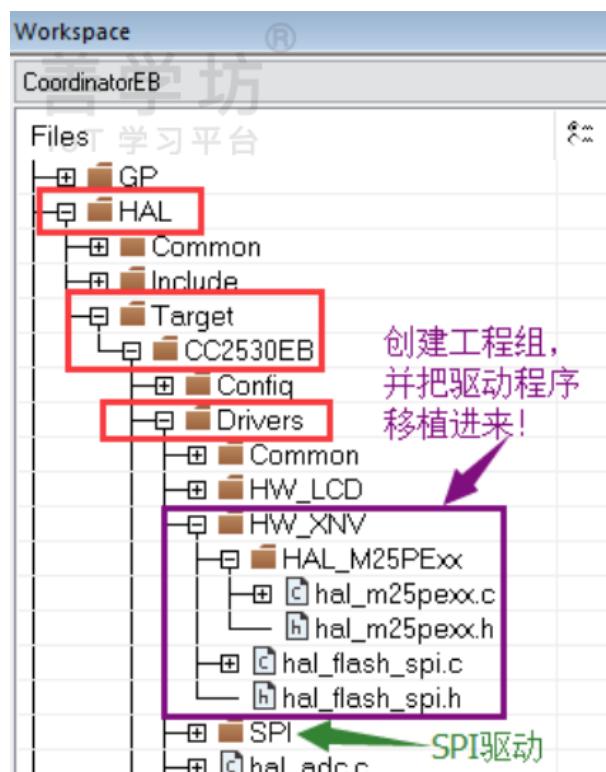
The driver is from the basics:



Among them, hal\_flash\_spi.h/c is used to adapt to the SPI bus, because our external FLASH uses the chip model M25PE80, and the driver bus is SPI; and the SPI driver is the same as the program used by the screen! hal\_m25pexx.h/c is the driver of the external FLASH.

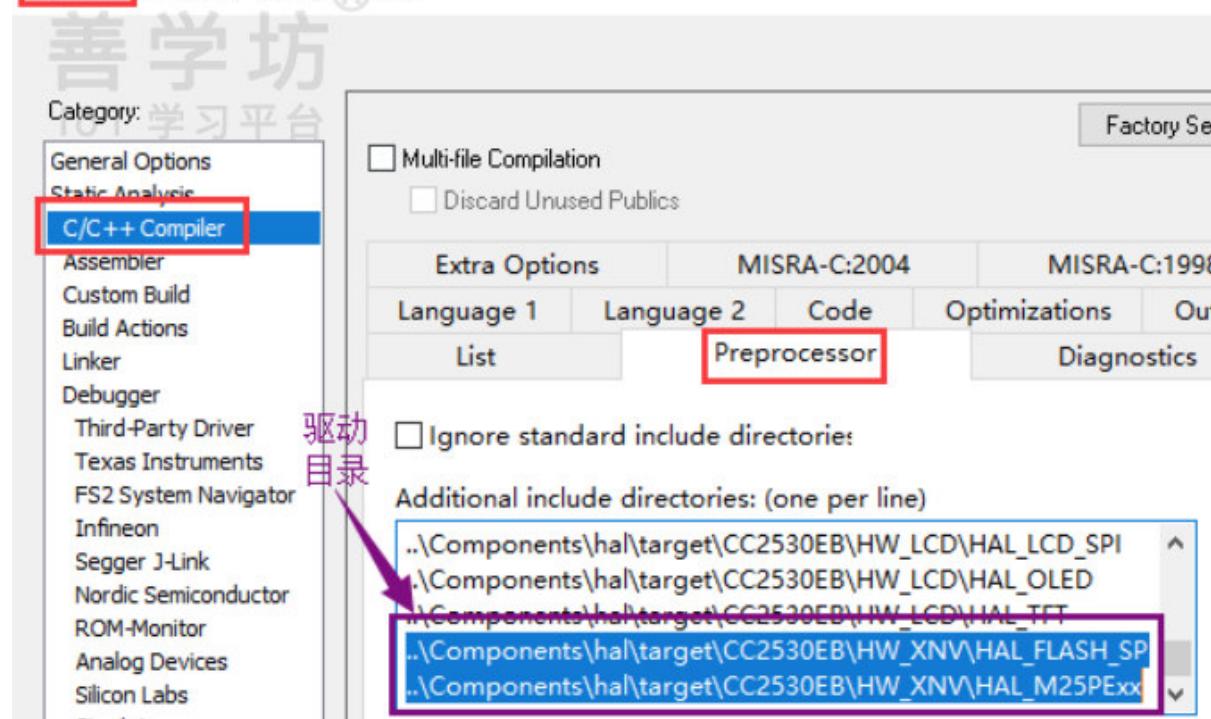
## Driver Porting

Create a corresponding project group in the project and add the driver:



Add the program directory to the project settings:

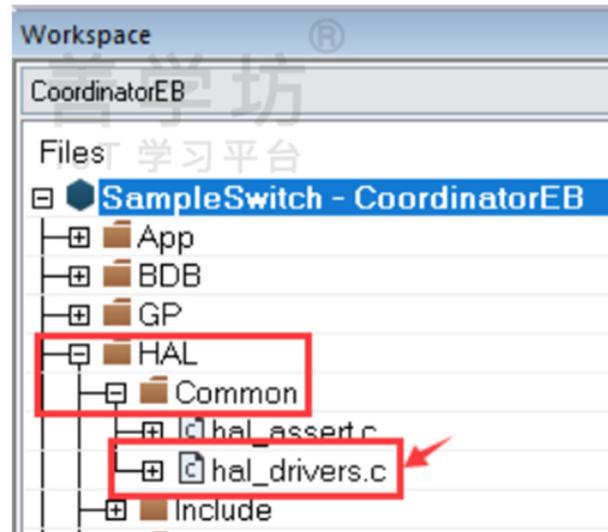
### Options for node "SampleSwitch"



Added content:

```
$PROJ_DIR$..\..\..\..\..\Components\hal\target\CC2530EB\HW_XNV\HAL_FLASH_SPI  
$PROJ_DIR$..\..\..\..\..\Components\hal\target\CC2530EB\HW_XNV\HAL_M25PExx
```

Initialize external FLASH in the file hal\_drivers.c:



Initialization (remember to include the hal\_m25pexx.h header file):

## HalDriverInit()

```

114 void HalDriverInit (void)
115 {
116     /* TIMER */
117     #if (defined HAL_TIMER) && (HAL_TIMER == TRUE)
118
119     /* ADC */
120     #if (defined HAL_ADC) && (HAL_ADC == TRUE)
121
122     /* DMA */
123     #if (defined HAL_DMA) && (HAL_DMA == TRUE)
124
125     /* AES */
126     #if (defined HAL_AES) && (HAL_AES == TRUE)
127
128     /* LCD */
129     #if (defined HAL_LCD) && (HAL_LCD == TRUE)
130
131     /* LED */
132     #if (defined HAL_LED) && (HAL_LED == TRUE)
133
134     /* UART */
135     #if (defined HAL_UART) && (HAL_UART == TRUE)
136
137     /* KEY */
138     #if (defined HAL_KEY) && (HAL_KEY == TRUE)
139
140     /* SPI */
141     #if (defined HAL_SPI) && (HAL_SPI == TRUE)
142
143     /* HID */
144     #if (defined HAL_HID) && (HAL_HID == TRUE)
145
146
147     halM25PExxInit(); ← 初始化外部FLASH
148 }

```

## Driver Application

In the application zcl\_samplesw.c, we set a periodic event, which will read and write the external FLASH. In the application layer event processing function zclSampleSw\_event\_loop, we add a piece of code:

zcl\_samplesw.c x hal\_lcd\_spi.h hal\_m25pexx.h hal\_drivers.c

文件: zcl\_samplesw.c  
函数: zclSampleSw\_event\_loop

```
405 // Test Event
406 if ( events & SAMPLEAPP_TEST_EVT )
407 {
408     uint8 readVal;
409     static uint8 writeVal = 0;
410     writeVal++;
411
412     // Write
413     halM25PExxWrite(1234, &writeVal, 1); 写和读的值显示到显示屏上!
414
415     // Read
416     halM25PExxRead(1234, &readVal, 1);
417
418     HalLcdWriteStringValue("Write:", writeVal, 10, 3);
419     HalLcdWriteStringValue("Read:", readVal, 10, 4);
420
421     osal_start_timerEx(zclSampleSw_TaskID,
422                         SAMPLEAPP_TEST_EVT,
423                         3000);
424
425
426     return ( events ^ SAMPLEAPP_TEST_EVT );
427 }
```

The event handling function does three things. First, it writes the value of a variable to the external FLASH address 1234; then it reads the data from this address to another variable; finally, it displays the written value and the read value on the display!

## Simulation debugging

After compiling the project, burn the program to the development board. You can see that the display shows the values of reading and writing external FLASH. The written and read values are the same, indicating that the external FLASH is working properly and there is no problem with the driver.

## 9.14. TFT display experiment (optional)

### Technical support instructions

1. Customer service provides simple technical support, which is generally based on self-study
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer, thank you for your understanding!

Before studying this section, check whether your screen is a TFT screen. If not, you can skip this section.

The number of pins on a 0.96-inch TFT screen is 8, and the pin names are GND, VCC, SCL, SDA, RES, DC, CS, and BLK. If this is consistent with yours, then you have a 0.96-inch TFT screen.

The previous chapter explained how to use OLED12864, but OLED12864 can only display a single color, not color. This lesson will explain that TFT is a type of LCD display that supports color display.

## Introduction to TFT Display

TFT-LCD (Thin Film Transistor-Liquid Crystal Display) is a thin film transistor liquid crystal display. Each pixel in the screen is equipped with a thin film transistor.

The color of this transistor can be configured using a 16-bit RGB value. The left 5 bits of this 16-bit RGB value represent Red, the middle 6 bits represent Green, and the right 5 bits represent Blue, as shown in the figure below.



The actual picture of the 0.96-inch TFT display in the development kit is shown in the figure below. Its resolution is  $64 \times 128$  pixels, that is, there are 64 pixels vertically and 128 pixels horizontally.



### The communication protocol

is the same as that of OLED12864. This TFT display also supports SPI protocol driver. If you need to know more about SPI protocol, please refer to the OLED12864 section.

### The font library and picture library

are similar to OLED12864. In order for the TFT screen to display various characters and pictures normally, the font library and picture library also need to be prepared.

## TFT display driver API

In order to drive the TFT display, the author designed a corresponding driver API for readers to learn and use.

Similar to OLED12864, the driver API of TFT display can be divided into two levels, namely **the upper TFT display API** and **the lower SPI driver API**. The lower SPI driver API is consistent with that of OLED12864. For more information, please refer to the OLED12864 section.

The driver API also requires the cooperation of font and image library files. These three are located as shown in the figure.



## TFT Display API

After configuring the display pins and font files, call the TFT display API to display the specified content on the screen. Open the hal\_tft.h file in the TFT folder and you can find the TFT display API definition code:

```
/**  
 * @fn    halTFTInit  
 *  
 * @brief  初始化TFT屏幕，使用TFT屏幕前必须先调用此函数  
 */
```

```
* @param screenColor - 屏幕颜色的RGB值，用于指定屏幕的默认颜色
*/
void halTFTInit(uint16 screenColor);

/***
* @fn    halTFTSetScreen
*
* @brief 设定整个屏幕的颜色，具有清除屏幕显示内容的作用
*
* @param screenColor - 屏幕颜色的RGB值
*/
void halTFTSetScreen(uint16 screenColor);

/***
* @fn    halTFTShowX16
*
* @brief 在屏幕上显示高度为16像素的字符串，支持的字符格式：1. 8x16 ASCII码；2. 16x16 汉字
*
*
* @param x - 指定在横向从左边数起第x个像素开始显示图像，参数值范围：0 ~ 127
* @param y - 指定在纵向从上边数起的第y个像素开始显示图像，参数值范围：0 ~ 64
* @param fontColor - 字体颜色的RGB值
* @param backgroundColor - 背景颜色的RGB值
* @param str - 待显示的字符串
*
* @warning 16x16汉字
的需要先在此文件中定义FONT_TABLE_CHINESE_16x16
*/
void halTFTShowX16(uint16 x, uint16 y, uint16 fontColor, uint16 backgroundColor, const uint8 *str);

/***
* @fn    halTFTShowPicture
*
* @brief Show Picture
```

```

*
* @param x - 指定在横向从左边数起第x个像素开始显示图像，参数值范围：0 ~ 127
* @param y - 指定在纵向从上边数起的第y个像素开始显示图像，参数值范围：0 ~ 64
* @param picWidth - 图片的宽度，参数值范围：1~128
* @param picHeight - 图片的高度，参数值范围：1~64
* @param pic - 待显示的图片
*/
void halTFTShowPicture(uint8 x, uint8 y, uint8 picWidth, uint8 picHeight, const uint8 *pic);

```

### halTFTShowX16 usage details

The resolution of the TFT display is 64x128 pixels, which can be understood as a two-dimensional table with **64 rows and 128 columns**. The halTFTShowX16 function supports the display of 8x16 standard ASCII characters and 16x16 Chinese characters, where 8x16 refers to a font that occupies 8 rows and 16 columns, and 16x16 refers to a font that occupies 16 rows and 16 columns.

### RGB value of color The

RGB value is mentioned in the above code. The author predefines some RGB values for reference. You can find the relevant code definition in the hal\_tft.h file:

```

#define HAL_TFT_PIXEL_RED    0xF800 //!< Red.
#define HAL_TFT_PIXEL_GREEN   0x07E0 //!< Green.
#define HAL_TFT_PIXEL_BLUE    0x001F //!< Blue.
#define HAL_TFT_PIXEL_BLACK   0x0000 //!< Black.
#define HAL_TFT_PIXEL_WHITE   0xFFFF //!< White.
#define HAL_TFT_PIXEL_YELLOW  0xFFE0 //!< Yellow.
#define HAL_TFT_PIXEL_GRAY    0xEF7D //!< Gray.

```

## Pin configuration of the display

Before calling the above display API, you need to configure the display pins first, that is, configure the display pins with the GPIO interface of CC2530. The TFT display contains the following 8 pins:

- GND
- VCC
- SCL
- SDA
- RES
- DC
- CS
- BLK

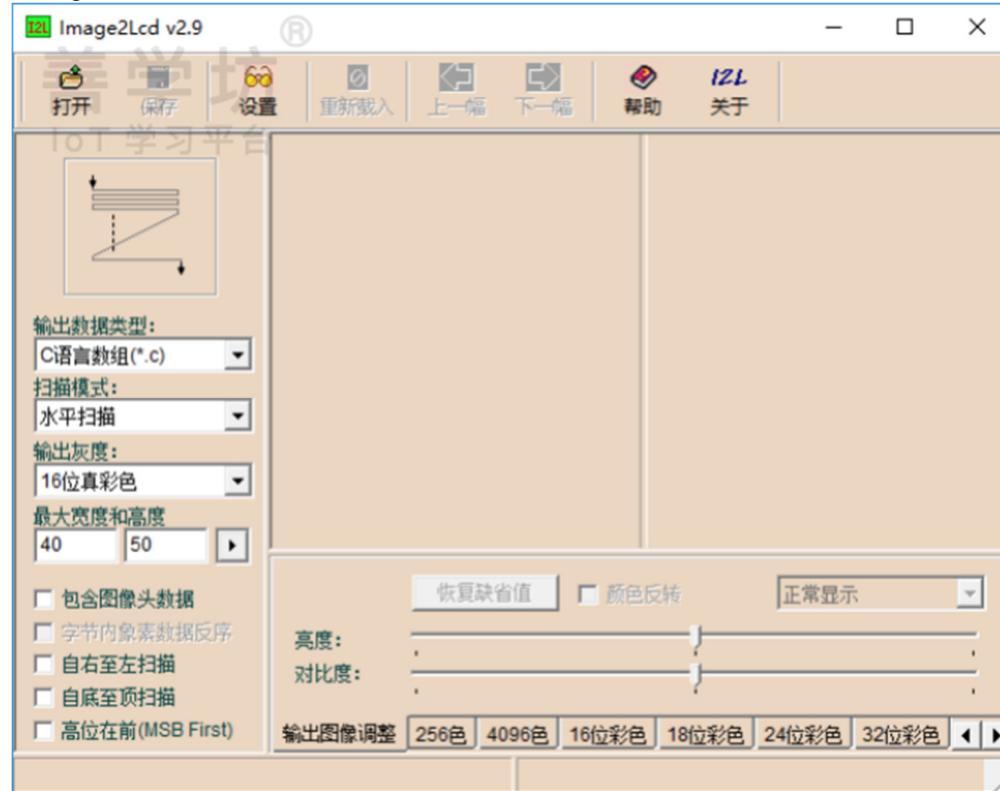
Although the pinout of the TFT display is slightly different from that of the OLED12864, the pinout configuration is the same. They both use the same configuration code, and you can refer to the pinout configuration instructions in the OLED for details.

## Configure fonts

The font library configuration method is almost the same as that used in OLED12864, with slightly different knowledge modulo parameters: **horizontal modulo** is used, and **byte reversal** is not performed.

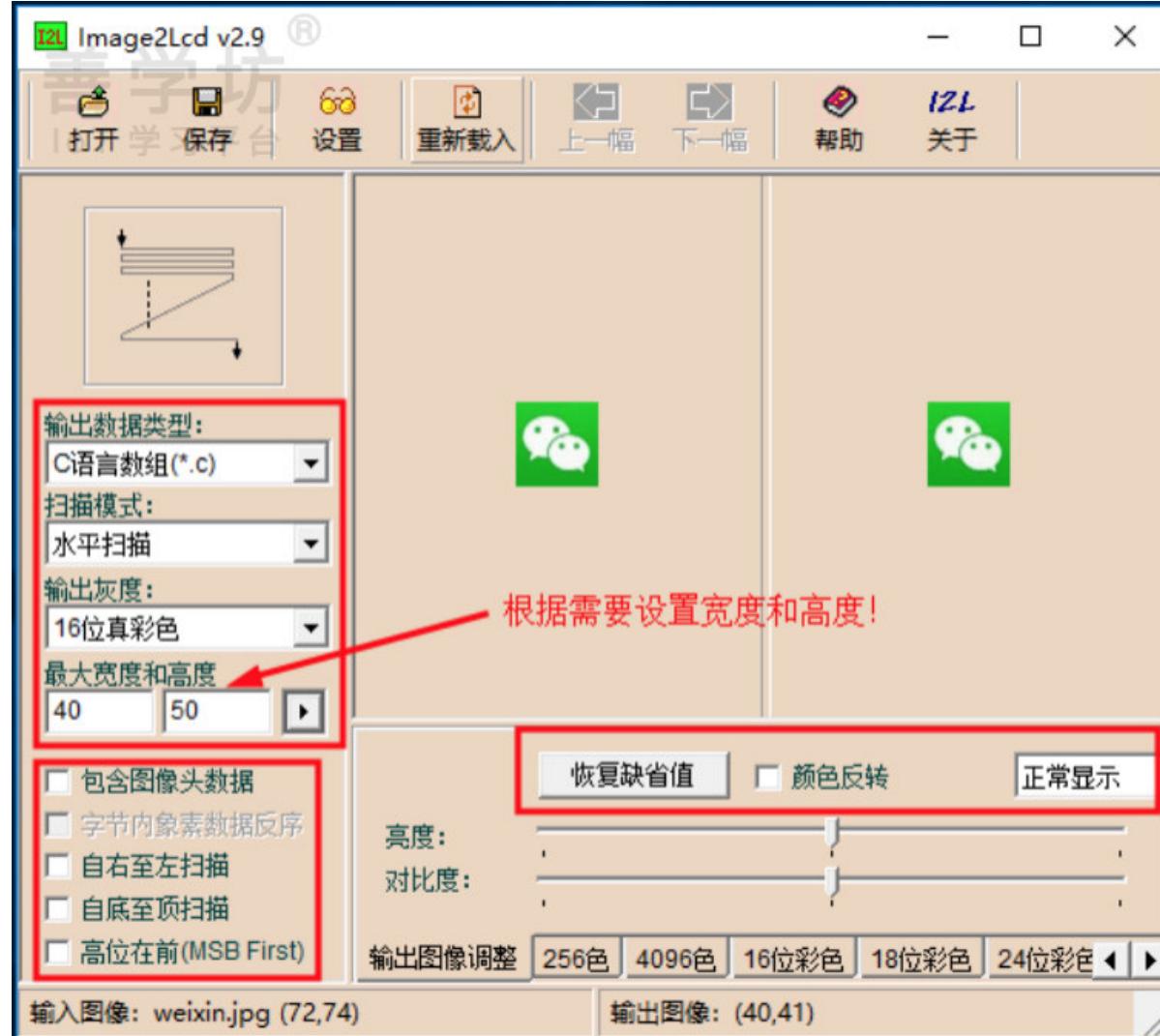
# Configure Gallery

Since the TFT display supports color display, its image acquisition method is quite different from that of OLED12864. You can find "Image Acquisition Software\Image2Lcd 2.9" in the software tools supporting this course and run the software as shown in the figure.



If you have not downloaded the software tools for this course, you can download them from the download page : [Download Page](#)

After importing the picture, set the modeling method according to the following parameters, and click Save to generate the corresponding gallery.



## Using TFT Display API

After configuring the display pins, font library and image library files, call the TFT display API to display the content on the screen. Open the supporting project, open main.c in the TFT12864 folder, and you can see the sample code for using the TFT display.

```
#include "hal_TFT.h"
#include "font_rgb_picture.h"
#include "hal_delay.h"

#define RED  HAL_TFT_PIXEL_RED
#define GREEN  HAL_TFT_PIXEL_GREEN
#define BLUE  HAL_TFT_PIXEL_BLUE
#define YELLOW  HAL_TFT_PIXEL_YELLOW
#define WHITE  HAL_TFT_PIXEL_WHITE
#define BLACK  HAL_TFT_PIXEL_BLACK

void main(void)
{
    setSystemClk32MHZ(); // 初始化系统时钟频率
```

```

halTFTInit(HAL_TFT_PIXEL_WHITE); // 初始化TFT屏幕，并指定默认颜色为白色

delayMs(SYCLK_32MHZ, 500); // 延后0.5ms在接着执行下面的代码

while(1)
{
    /* Test1 - 变换屏幕颜色 */

    halTFTSetScreen(RED);
    delayMs(SYCLK_32MHZ, 500);

    halTFTSetScreen(GREEN);
    delayMs(SYCLK_32MHZ, 500);

    halTFTSetScreen(BLUE);
    delayMs(SYCLK_32MHZ, 500);

    halTFTSetScreen(YELLOW);
    delayMs(SYCLK_32MHZ, 500);

    halTFTSetScreen(WHITE);
    delayMs(SYCLK_32MHZ, 1000);

    /* Test2 - 显示 8x16 ASCII字符 and 16x16 汉字 */

    halTFTShowX16(0, 0, BLACK, WHITE, "今天: 2018/1/31");
    halTFTShowX16(0, 16, RED, WHITE, "气温(T & H):");
    halTFTShowX16(0, 32, BLUE, WHITE, "Temperature: 25");
    halTFTShowX16(0, 48, BLUE, WHITE, "Humidity: 30%");

    /* 注意：对于汉字，必须先取字模后存放到汉字字库文件font_chinese_v_16x16.h中*/

    delayMs(SYCLK_32MHZ, 4000); // 延后4ms

    halTFTSetScreen(WHITE); // 设备屏幕颜色为白色

    /* Test3 - 在坐标 (40像素, 12像素) 处显示分辨率为40x40像素的图片 */

    halTFTShowPicture(40, 12, 40, 40, Picture_40x40_WeiXinIco);

    /* 注意：对于图片，系需要先取模后存放在字库文件font_v_picture.h中*/

    delayMs(SYCLK_32MHZ, 4000); // 延迟
}
}

```

## Debugging Simulation

Before learning this lesson, you need to master the basic program download and simulation operations. Please refer to: <https://z7po9bxpe4.k.topthink.com/@zigbee-dev-guide/TFTxianshiqishiyan%EF%BC%88xuanxiu%EF%BC%89.html#!>

Insert the screen into the development board as shown in the figure.



If you are using a ZigBee standard board, you need to set the dip switches as shown in the figure.



Open the project code for this course, compile and download it to the development board, and you can see the corresponding content displayed on the screen in a loop!

## 9.15. Lighting project source code analysis

### 9.15.1. ZHA Lighting Project

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

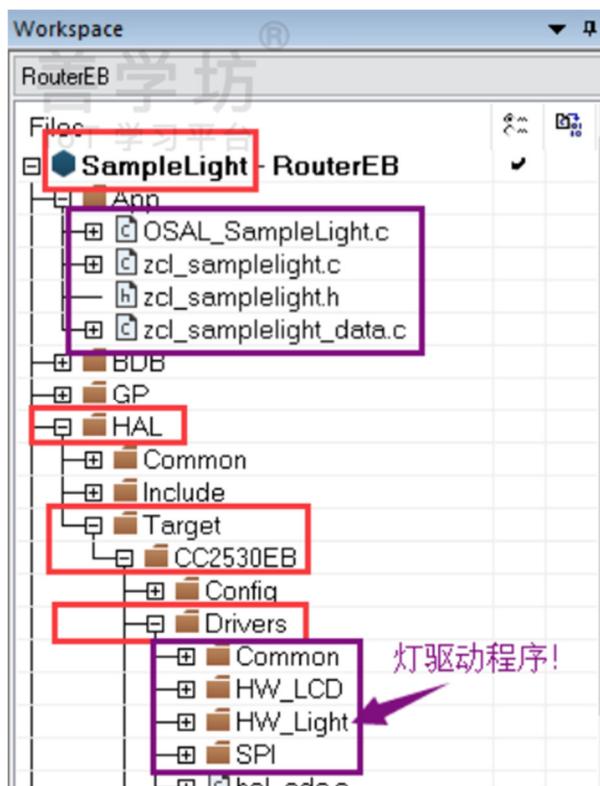
Our previous courses were all based on SampleSwitch, which is the application of smart sockets. The reason for using SampleSwitch is that smart sockets are indispensable for actual enterprises to develop ZigBee products. It is also relatively simple and easy for beginners to learn. In this class, we will switch to another device type: Lighting. The project file is in:

名称	修改
GenericApp	201
SampleDoorLock	201
SampleDoorLockController	201
SampleLight	201
SampleSwitch	201
SampleTemperatureSensor	201
SampleThermostat	201
Source	201

**ZHA-Lighting**

There are four common types of ZigBee Lighting: On/Off Lighting that can only perform switching actions, Dimmable Lighting that supports brightness adjustment, CCT Lighting that supports color temperature adjustment, and RGB Lighting that supports color adjustment! Different types of lights have different hardware requirements.

As with the previous project, we need to transplant the LCD driver, and then set the project configuration; at the same time, we remove the application layer UI function, and the final project is as follows:



Add the directory HW\_Light to provide the driver for the light. This chapter supports On/Off Lighting. If you need to support Dimmable Lighting, you can connect an LED to P1\_4. The IO port P0\_4 on our board is only for On/Off control, but the sample program comes with the control code for P1\_4, which can be directly supported!

## 9.15.2. ZHA Lighting source code analysis

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

Device Simple Descriptor:

```
777 SimpleDescriptionFormat_t zclSampleLight_SimpleDesc =  
778 {  
779     SAMPLELIGHT_ENDPOINT, // 应用端点  
780     ZCL_HA_PROFILE_ID, // Profile: ZHA  
781     #ifdef ZCL_LEVEL_CTRL  
782         ZCL_HA_DEVICEID_DIMMABLE_LIGHT,  
783     #else  
784         ZCL_HA_DEVICEID_ON_OFF_LIGHT,  
785     #endif  
786     SAMPLELIGHT_DEVICE_VERSION, // int App  
787     SAMPLELIGHT_FLAGS, // int App  
788     ZCLSAMPLELIGHT_MAX_INCLUSTERS, // byte App  
789     (cId_t *)zclSampleLight_InClusterList, // byte *pAppInClusterList  
790     0, // byte AppNumInClusters;  
791     NULL // byte *pAppInClusterList;  
792 }.
```

The engineering structure of ZHA Lighting is exactly the same as our previous smart socket project:



First, add a coordinator to establish a network and a device to join the network (a timed event will be started to try again if the joining fails):

```
zcl_samplelight.c * x  
zclSampleLight_Init(byte)  
336 #ifdef ZDO_COORDINATOR  
337     bdb_StartCommissioning( BDB_COMMISSIONING_MODE_NWK_FORMATION |  
338                             BDB_COMMISSIONING_MODE_FINDING_BINDING );  
339  
340     NLME_PermitJoiningRequest(255);  
341 #else  
342     bdb_StartCommissioning( BDB_COMMISSIONING_MODE_NWK_STEERING |  
343                             BDB_COMMISSIONING_MODE_FINDING_BINDING );  
344
```

## ZCL General command processing function

View the list of ZCL's "General" command processing functions:

```

233 /* **** */
234 * ZCL General Profile Callback table ZCL General命令处理函
235 */
236 static zclGeneral_AppCallbacks_t zclSampleLight_CmdCallbacks
237 {
238     zclSampleLight_BasicResetCB, // Basic Cluster Re
239     NULL, // Identify Trigger
240     zclSampleLight_OnOffCB, // On/Off cluster c
241     NULL, // On/Off cluster e
242     NULL, // On/Off cluster e
243     NULL, // On/Off cluster e
244 #ifdef ZCL_LEVEL_CTRL
245     zclSampleLight_LevelControlMoveToLevelCB, // Level Control
246     zclSampleLight_LevelControlMoveCB, // Level Control
247     zclSampleLight_LevelControlStepCB, // Level Control
248     zclSampleLight_LevelControlStopCB, // Level Control
249 #endif
250 #ifdef ZCL_GROUPS
251     NULL, // Group Response c

```

调节亮度!

The most important function is zclSampleLight\_LevelControlMoveToLevelCB, which is entered when we send a command to adjust the brightness of the light to the device through ZCL. The processing flow of this function is as follows:

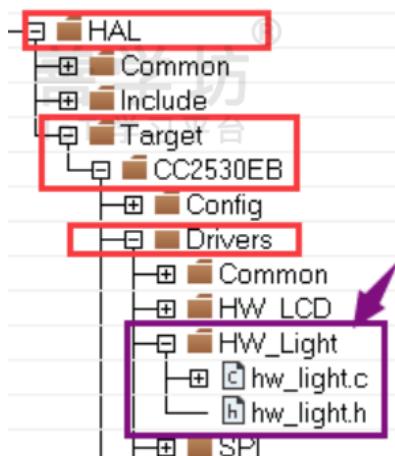
```

zclSampleLight_LevelControlMoveToLevelCB
--> Call function: zclSampleLight_MoveBasedOnTime
--> Start event: SAMPLELIGHT_LEVEL_CTRL_EVT
--> Call in event handler: zclSampleLight_AdjustLightLevel

```

That is to say, it will enter the function zclSampleLight\_AdjustLightLevel in the end. The calls and contents of the above functions are ultimately for adjusting the brightness of the light. You may ask, why are there so many steps to adjust the brightness? Isn't it enough to just call an API? This is because ZCL supports time-sharing adjustment of the brightness. For example, it takes 2 seconds to adjust the brightness from 100% to 1%. The advantage of this is that it improves the user experience and gives people a gradual warm effect. The time-sharing duration can be set, but the minimum unit is 100 milliseconds. For example, 2 seconds is 20 100 milliseconds.

The logic protocol stack for brightness adjustment has been prepared for us, but no driver for the lamp is provided, so we only need to write the driver for the lamp and then call it in the brightness adjustment function. The driver for the lamp (a program written by the author, not very rigorous, only for reference and not for commercial use):



The driver provides several APIs:

```

51 void hwLight_Init(uint8 type, uint8 gpio);
52
53 void hwLight_SetOn(uint8 gpio);
54 void hwLight_SetOff(uint8 gpio);
55 void hwLight_SetLevel(uint8 gpio, uint8 level);

```

API parameter description:

- type - provides two modes: On/Off and Dimmable, that is, the light can be configured as On/Off Lighting that only supports switching, or Dimmable Lighting that supports dimming
- gpio - Hardware connection of the light. We can connect an LED light to P1\_4 for dimming, or use the onboard P0\_4 LED for On/Off operation.

The function hwLight\_SetLevel is used to set the brightness of the LED. This function comes from the timer PWM output dimming chapter of the basics. The program is relatively simple and will not be expanded.

To use this driver you first need to initialize it:

File: hal\_drivers.c

--> Function: HalDriverInit

--> Last call:

```

hwLight_Init( HW_LIGHT_TYPE_ONOFF, HW_LIGHT_GPIO_P04 );
//hwLight_Init( HW_LIGHT_TYPE_DIMMABLE, HW_LIGHT_GPIO_P14 );

```

Then we just need to call hwLight\_SetLevel in the zclSampleLight\_AdjustLightLevel function mentioned above:

File: zcl\_samplelight.c

--> Function: zclSampleLight\_AdjustLightLevel

--> Last call (not called by the coordinator): hwLight\_SetLevel

If it is P0\_4, then we determine whether the brightness value is less than 254 / 2. If so, turn off the light, otherwise turn on the light. At this point, the logic of adjusting the brightness of the light is complete! !

### 9.15.3. Lighting brightness adjustment experiment

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

### The coordinator sends the ZCL brightness adjustment command

The API for adjusting brightness belongs to the ZCL General command and can be found in zcl\_general.h:

```
1832 #ifdef ZCL_LEVEL_CTRL  
1833 /*  
1834 * Call to send out a Level Control Move to Level Request  
1835 *      cmd - Move or Move with On/Off  
1836 *      level - what level to move to  
1837 *      transitionTime - how long to take to get to the level (ms)  
1838 */  
1839 extern ZStatus_t zclGeneral_SendLevelControlMoveToLevelRequest(
```

First, we start a periodic event in the application layer initialization function of the coordinator:

文件: zcl\_samplelight.c  
函数: zclSampleLight\_Init

```
zcl_samplelight.c x
zclSampleLight_Init(byte)
342
343     // level test event
344     osal_start_timerEx(zclSampleLight_TaskID,
345                         SAMPLEAPP_LEVEL_TEST_EVT,
346                         SAMPLEAPP_LEVEL_TEST_PERIOD);
```

The event is handled as follows:

```
zcl_samplelight.c x ®
zclSampleLight_event_loop(uint8, uint16)

414 #ifdef ZDO_COORDINATOR
415     // Level test
416     if ( events & SAMPLEAPP_LEVEL_TEST_EVT )
417     {
418         zclSampleLight_LevelTest();
419
420         osal_start_timerEx(zclSampleLight_TaskID,
421                             SAMPLEAPP_LEVEL_TEST_EVT,
422                             SAMPLEAPP_LEVEL_TEST_PERIOD);
423
424         return ( events ^ SAMPLEAPP_LEVEL_TEST_EVT );
425     }

```

That is, the custom function will be called in the end: `zclSampleLight_LevelTest`, function contents:

```

13.         COMMAND_LEVEL_MOVE_TO_LEVEL,
14.         level, // 亮度值
15.         10, // 10个100ms完成亮度变化
16.         TRUE,
17.         seqNum++ );
18.
19. // 显示亮度值
20. HalLcdWriteStringValue("Set Level:", level, 10, 4);
21.
22. // 调整亮度值
23. if(level < 50) level = 254;
24. else level -= 50;
25.}

```

This function creates an event and broadcasts the brightness value. If the terminal device joins the network, it will receive this broadcast packet and set the corresponding brightness! At the same time, the coordinator will display the brightness value on the display! You may have noticed that the brightness value is from 1 to 254. This is the API requirement. After receiving this data, the device needs to convert it to 1% to 100% for use!

## Simulation debugging

Compile the programs of the coordinator and router respectively, and then burn them into the two development boards respectively; after power-on, you can see the broadcasted brightness value displayed on the coordinator screen. Note: If a dimming light (P1\_4) is used, the router does not use the screen in the experiment of this chapter, because the GPIO of the light is multiplexed with the screen, which will cause interference; but the program runs by default with P0\_4 used as the On/Off light, so there is no impact.

## 9.16. TemperatureSensor project source code analysis

### 9.16.1. ZHA TemperatureSensor Project

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

#### Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers **will** answer community questions as soon as possible, but they are front-line developers and [**cannot guarantee**] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

We have previously discussed smart sockets (SampleSwitch) and lights (Lighting). In this section, we will discuss sensor-type devices: temperature sensors, which are used in:

Z-Stack 3.0.1\Projects\zstack\HomeAutomation

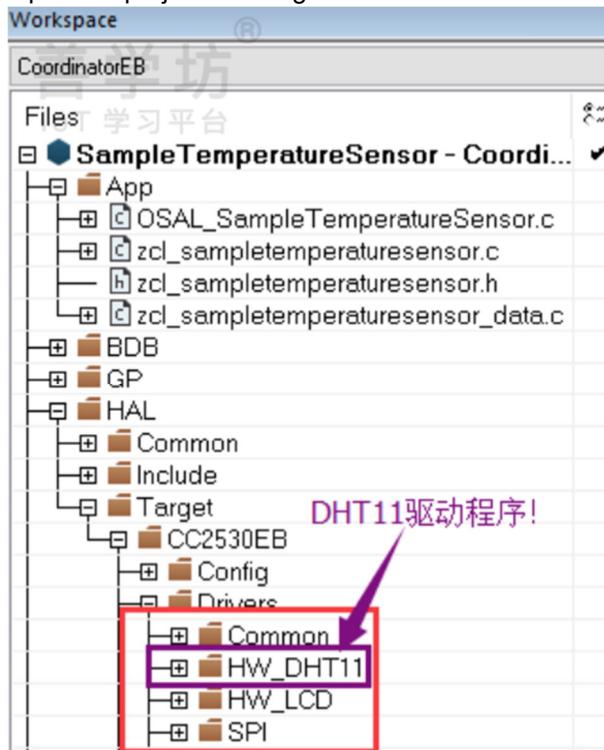
名称	修改
GenericApp	201
SampleDoorLock	201
SampleDoorLockController	201
SampleLight	201
SampleSwitch	201
SampleTemperatureSensor	201
SampleThermostat	201
Source	201

The temperature sensor we use is DHT11, so in addition to porting the LCD driver to the project, we also need to port the DHT11 driver from the basics:

Z-Stack 3.0.1\Components\hal\target\CC2530EB

名称	修改
Common	201
HW_DHT11	201
HW_LCD	201
HW_Light	201
SPI	201

Open the project and migrate the driver:



## 9.16.2. ZHA TemperatureSensor Source Code Analysis

Tutor video explanation: <https://www.bilibili.com/video/BV1k34y1D7Vz/>

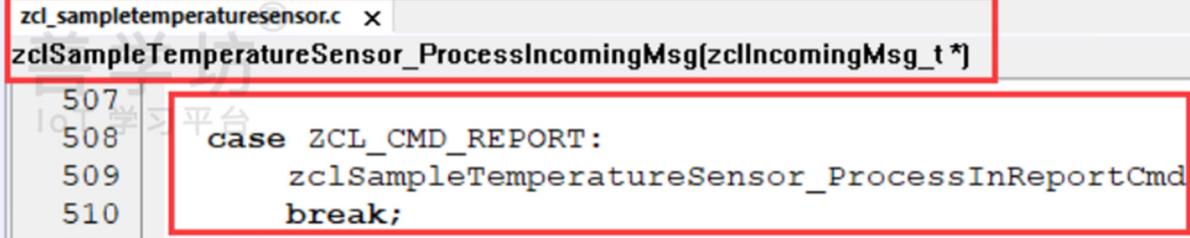
## Technical support instructions:

1. Generally, self-study is the main method.
2. You can ask questions in the official Q&A community: <https://bbs.csdn.net/forums/zigbee>
3. Engineers will answer community questions as soon as possible, but they are front-line developers and [cannot guarantee] the timeliness of the answer. It is hard work to answer them, thank you for your understanding!

The experiment in this chapter is very similar to the attribute reporting explained previously. The function we implement is: the device will start a periodic event to detect whether the temperature has changed. If it has changed, the data will be reported to the coordinator; when the coordinator receives the data, it will be displayed on the screen!

## Coordinator

Receive the "report" command and process it:



```
zcl_sampletemperaturesensor.c x
zclSampleTemperatureSensor_ProcessIncomingMsg(zclIncomingMsg_t *)
```

```
507
508     case ZCL_CMD_REPORT:
509         zclSampleTemperatureSensor_ProcessInReportCmd();
510         break;
```

Final calling function: zclSampleTemperatureSensor\_ProcessInReportCmd

```
1. static uint8 zclSampleTemperatureSensor_ProcessInReportCmd( zclIncomingMsg_t *pInMsg )
2. {
3.     zclReportCmd_t *reportCmd;
4.     uint8 i;
5.
6.     HalLcdWriteString("pro report", 3);
7.
8.     reportCmd = (zclReportCmd_t *)pInMsg->attrCmd;
9.     for ( i = 0; i < reportCmd->numAttr; i++ )
10.    {
11.        if( pInMsg->clusterId == ZCL_CLUSTER_ID_MS_TEMPERATURE_MEASUREMENT &&
12.            reportCmd->attrList[i].attrID == ATTRID_MS_TEMPERATURE_MEASURED_VALUE)
13.        {
14.            int16 temp = *((uint16 *)reportCmd->attrList[i].attrData);
15.            HalLcdWriteStringValue("Rx Temp:", temp, 10, 4);
16.        }
17.    }
18.
19.    return ( TRUE );
20.}
```

Cluster: ZCL\_CLUSTER\_ID\_MS\_TEMPERATURE\_MEASUREMENT,

Attribute: ATTRID\_MS\_TEMPERATURE\_MEASURED\_VALUE

Also remember to turn on macros in "option":

ZCL\_REPORT\_DESTINATION\_DEVICE

ZCL\_REPORT

## Equipment

Define periodic "report" events:

zcl\_sampletemperaturesensor.h

```
68 // Report Event
69 #define SAMPLETEMPERATURESENSOR_REPORT_EVT
70 #define SAMPLETEMPERATURESENSOR_REPORT_PERIOD
```

Start periodic reporting of events in the application layer initialization function:

zcl\_sampletemperaturesensor.c

zclSampleTemperatureSensor\_Init(byte)

文件: zcl\_sampletemperaturesensor.c

函数: zclSampleTemperatureSensor\_Init

```
264
265 // Report Event
266 osal_start_timerEx(zclSampleTemperatureSensor_TaskID,
267                      SAMPLETEMPERATURESENSOR_REPORT_EVT,
268                      SAMPLETEMPERATURESENSOR_REPORT_PERIOD);
```

Process the event in the application layer event handler:

zcl\_sampletemperaturesensor.c

zclSampleTemperatureSensor\_event\_loop(uint8, uint16)

```
335
336 // Report
337 if (events & SAMPLETEMPERATURESENSOR_REPORT_EVT)
338 {
339     zclSampleTemperatureSensor_GetAndReport();
340
341     osal_start_timerEx(zclSampleTemperatureSensor_TaskID,
342                         SAMPLETEMPERATURESENSOR_REPORT_EVT,
343                         SAMPLETEMPERATURESENSOR_REPORT_PERIOD);
344
345     return (events ^ SAMPLETEMPERATURESENSOR_REPORT_EVT);
346 }
```

Finally, the function zclSampleTemperatureSensor\_GetAndReport is called. The function gets the temperature value through the DHT11 driver, and then determines whether the temperature changes. If it changes, it reports the data to the coordinator!

```
1. static void zclSampleTemperatureSensor_GetAndReport(void)
2. {
3.     halDHT11Data_t dat = halDHT11GetData(); // 获取温度值
4.     if(dat.ok) // 检测是否获取成功
5.     {
6.         HalLcdWriteStringValue("Temp:", dat.temp, 10, 3);
7.
8.         // 如果温度值没有变化，直接忽略 (return)
9.         if(zclSampleTemperatureSensor_MeasuredValue == ((int16)dat.temp * 100))
10.            return;
11.         zclSampleTemperatureSensor_MeasuredValue = (int16)dat.temp * 100;
12.     }
13.     else
14.     {
15.         HalLcdWriteStringValue("Error Code:", dat.ok, 10, 3);
16.     }
}
```

```

17. }
18. if( zclSampleTemperatureSensor_MeasuredValue > zclSampleTemperatureSensor_MaxMeasuredValue )
19. {
20.     zclSampleTemperatureSensor_MeasuredValue = zclSampleTemperatureSensor_MaxMeasuredValue;
21. }
22. else if( zclSampleTemperatureSensor_MeasuredValue < zclSampleTemperatureSensor_MinMeasuredValue )
23. {
24.     zclSampleTemperatureSensor_MeasuredValue = zclSampleTemperatureSensor_MinMeasuredValue;
25. }
26.
27. // Report data // 上报温度值
28.zclSampleTemperatureSensor_ReportTemp(zclSampleTemperatureSensor_MeasuredValue);
29.

```

Finally, the data is reported through the function `zclSampleTemperatureSensor_ReportTemp`. The structure of this function is basically the same as the content of the attribute reporting explained earlier, so it will not be expanded here!

## Simulation debugging

Compile the programs of the coordinator and router respectively, and then burn them into the two development boards respectively; we can blow air on the DHT11 temperature and humidity sensor to change the temperature value, and we can see the data reported and displayed:

## 9.17. Other extracurricular projects

### 9.17.1. Temperature and harmful gas SMS alarm system based on ZigBee

This project is **an extracurricular learning content**, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this project **in principle**. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

The description of this project is relatively concise and is only suitable for readers with a certain foundation, **not for beginners**

## 1. Field of focus

This solution is aimed at **course design** or **function verification demo** areas that have general requirements for communication stability, response speed and communication distance.

## 2. System Architecture Description and Function Demonstration

The system can detect the ambient temperature and the presence of harmful gases, and can send the data to the computer through the serial port and display the data on the computer. When an abnormal situation occurs, the warning light will light up and a text message will be sent to the mobile phone.

- Functional demonstration: <https://www.bilibili.com/video/BV14341127aW/> (the development board in the video is the old version, and a new version has been released, which is compatible with the old version)



### 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not publicly available on the Internet. They are currently only provided free of charge to friends who have purchased **the development kit**. <https://z7po9bxpe4.k.tothink.com/@zigbee-dev-guide/jiyuZigBeedewenduheyoushuiqitiduanxinbaojingxitong.html#!>

- Resource screenshots



### 4. Hardware required for the system

- Shanyuefang ZigBee Standard Board
- Shanyuefang ZigBee Mini Board
- Send SMS module
- Hazardous gas sensor
- 0.96 TFT display

The 0.96 TFT display is no longer available. You can switch to a 0.96 OLED display. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

### 5. Build your own system

#### 1. PC program

- Find and run the PC-side exe program in the supporting resources

#### 2. ZigBee device firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

- Connect the hardware as shown in the demo video or by observing the source code.
- Find the corresponding firmware in the supporting resources and burn the corresponding firmware
- After burning the firmware, power on the development board and connect

After ensuring that all steps are executed correctly, the alarm function can be realized in the temperature, harmful gas and other information. For specific effects, please refer to the "System Demonstration Effect" section in this article.

## 9.17.2. Multi-sensor detection and lighting alarm system based on ZigBee

This project is an extracurricular learning content, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this project in principle. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

The description of this project is relatively concise and is only suitable for readers with a certain foundation, **not for beginners**

### 1. Field of focus

This solution is aimed at course design or function verification demo areas that have general requirements for communication stability, response speed and communication distance.

### 2. System Architecture Description and Function Demonstration

The system demonstrates the use of CC2530 terminals equipped with multiple sensors, and sends data to the coordinator via ZigBee. When the coordinator detects abnormal data, the alarm light turns on.

- System architecture:

## Multi-sensor detection and lighting alarm system based on ZigBee

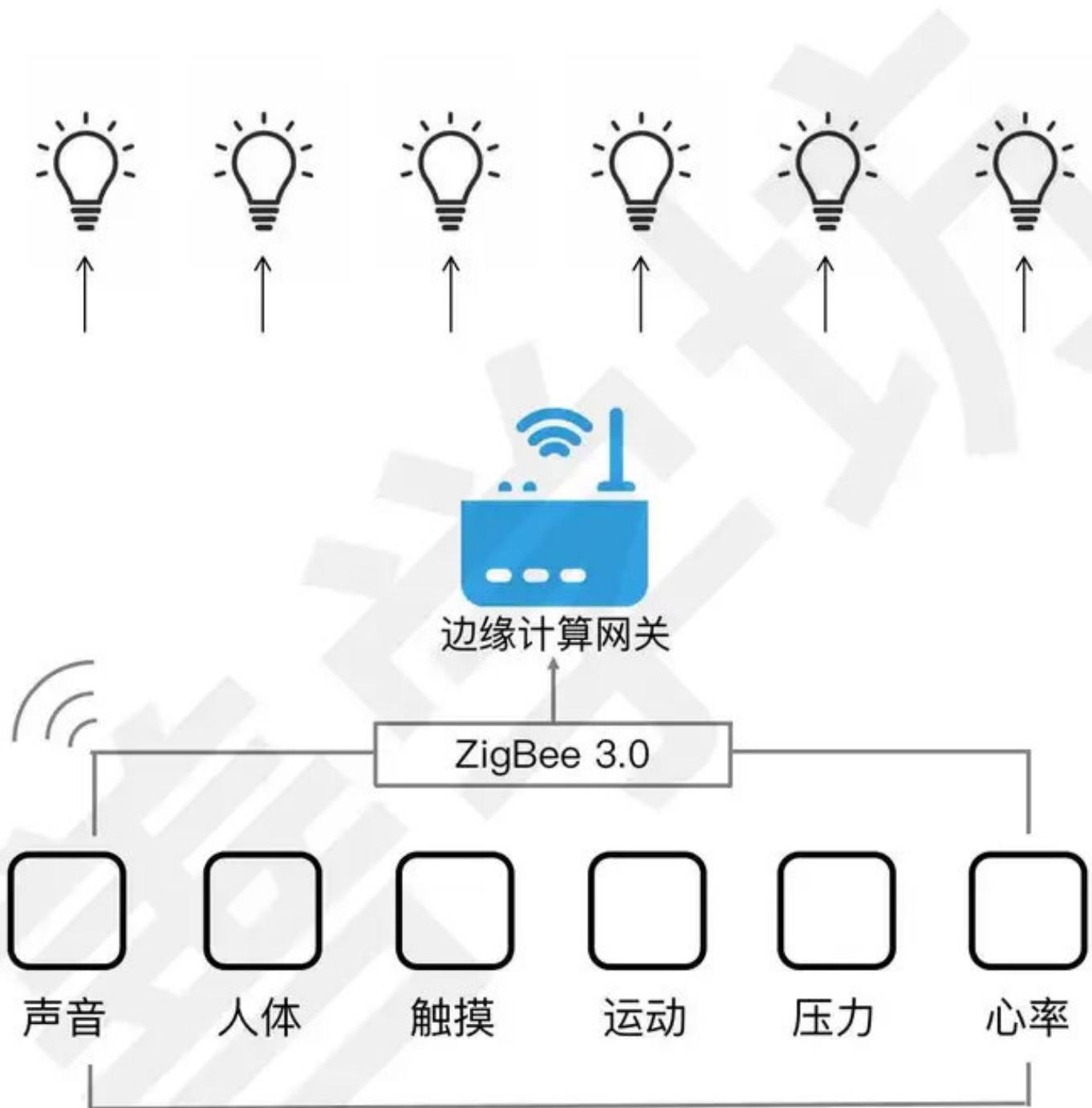
<https://zhuanlan.zhihu.com/p/394980004>

### 1. Field of focus

This solution is aimed at smart homes, industrial Internet of Things, smart cities, smart agriculture and other fields that have high requirements for communication stability, response speed and communication distance.

### 2. Brief description of the technical solution principle

The technical solution schematic is shown below:



#### Technical principle overview:

The human-computer interaction device is equipped with sound sensors, human infrared sensors, touch buttons, pressure sensors, heart rate sensors, and motion sensors. It can sense sound, human body, touch, pressure, heart rate, and motion, and complete the recognition process, and send the recognition results to the edge computing gateway through ZigBee wireless communication.

The ZigBee gateway is equipped with a ZigBee 3.0 module and multiple LED lights, which can receive the identification results sent by the human-computer interaction device and make corresponding responses based on the identification results. The response methods are.

1. If the sound sensor detects a loud sound in the environment, the light will turn on, and if there is no sound for 10 seconds, the light will turn off.
2. If the human body sensor detects that someone is approaching, the light will turn on, and if no one is approaching for 10 seconds, the light will turn off.
3. In the light-off state, if the touch button is pressed, the light will turn on, and if it is pressed again, the light will turn off.
4. In the light-off state, if the pressure sensor is pressed once, the light will turn on, and if it is pressed again, the light will turn off.
5. If the heart rate is detected to exceed the specified threshold, the light will turn on, and if it is lower than the specified threshold, the light will turn off.
6. If the motion sensor detects that the human-computer interaction device changes from a static or uniform motion state to a general motion state, the light will turn on, otherwise the light will turn off.

### **3. Main hardware components and related technical parameters of edge computing gateway**

#### **Main hardware components**

Zigbee wireless MCU: TI CC2530F256

USB to serial port chip : CH340

#### **Main technical parameters**

ZigBee wireless communication distance: The communication distance between adjacent nodes in open space is about 150 meters

ZigBee wireless communication rate: 240kb/s (theoretical value)

ZigBee electromagnetic wave frequency: 2.4GHz

ZigBee application protocol version: ZigBee 3.0

ZigBee core protocol version: ZigBee 2007 Pro

ZigBee wireless communication delay: <= 1 second

### **4. Main hardware components and related technical parameters of terminal equipment**

#### **Main hardware components**

Zigbee wireless MCU: TI CC2530F256

USB to serial port chip: CH340

Sound sensor: Shanxuefang SY01

Human infrared sensor: HC-SR501

Touch Button: TTP223 1-bit Touch Sensor Module

Pressure sensor: BE120-3AA foil resistor

Heart rate sensor: Shanxuefang XL01

Motion sensor : To be determined

#### **Main technical parameters**

ZigBee wireless communication distance: The communication distance between adjacent nodes in open space is about 150 meters

ZigBee wireless communication rate: 240kb/s (theoretical value)

ZigBee electromagnetic wave frequency: 2.4GHz

ZigBee application protocol version: ZigBee 3.0

ZigBee core protocol version: ZigBee 2007 Pro

ZigBee wireless communication delay: <= 1 second

Sound detection principle: vibration detection

Human body detection principle: Human body infrared detection, when a person enters its sensing range, it outputs a high level, and when a person leaves the sensing range, it automatically delays the high level and outputs a low level

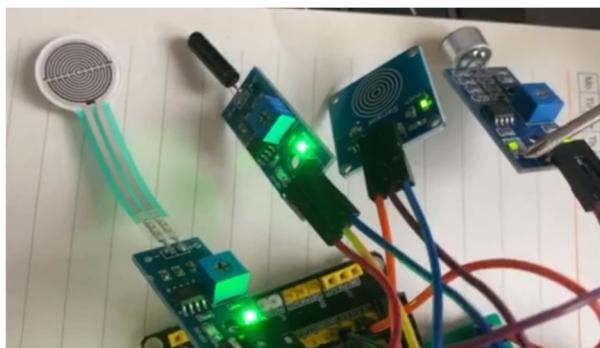
Touch detection principle: capacitive

Pressure detection principle: varistor

Heart rate detection principle: photoelectric reflection

Motion detection principle: To be determined

- Functional demonstration: <https://www.bilibili.com/video/BV1mf4y1w7WU/> (the development board in the video is the old version, and a new version has been released, which is compatible with the old version)



### **3. Download supporting resources**

The supporting resources of this project are additional parts and are not publicly available on the Internet. They are currently only provided free of charge to friends who have purchased the development kit. <https://z7po9bxpe4.k.tothink.com/@zigbee-dev-guide/jiyuZigBeeDevDuheyouhaiqitidianxinbaojingxitong.html#!>

- Resource screenshots



## 4. Hardware required for the system

- Shaxuefang ZigBee Standard Board
- Shaxuefang ZigBee Mini Board
- Sound Sensor
- Human infrared sensor
- Touch Sensor
- Motion Sensor
- Pressure Sensors
- Heart rate sensor
- 0.96 TFT display

The 0.96 TFT display is no longer available. You can switch to a 0.96 OLED display. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

## 5. Build your own system

### 1. PC program

- Find and run the PC-side exe program in the supporting resources

### 2. ZigBee device firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

- Connect the hardware as shown in the demo video or by observing the source code.
- Find the corresponding firmware in the supporting resources and burn the corresponding firmware
- After burning the firmware, power on the development board and connect

After ensuring that all steps are executed correctly, you can see the information of each sensor and realize the alarm function. For specific effects, please see the "System Demonstration Effect" section in this article.

### 9.17.3. Temperature, humidity, human infrared and sound and light alarm system based on ZigBee

This project is **an extracurricular learning content**, its content is **relatively complex** and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, **please stop studying**.

Shanxuefang does not provide technical support for this project **in principle**. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

The description of this project is relatively concise and is only suitable for readers with a certain foundation, **not for beginners**

## 1. Field of focus

**This solution is aimed at course design or function verification demo** areas that have general requirements for communication stability, response speed and communication distance.

## 2. System Architecture Description and Function Demonstration

The system demonstrates the use of CC2530 terminals equipped with a variety of sensors (the sensor list can be viewed in "Required Hardware for the System"), and sends data to the coordinator via ZigBee. The coordinator will sound a buzzer when it detects data anomalies.

- Functional demonstration: <https://www.bilibili.com/video/BV1bf4y1E7Zn/> (the development board in the video is the old version, and a new version has been released, which is compatible with the old version)



## 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not publicly available on the Internet. They are currently only provided free of charge to friends who have purchased **the development kit**. <https://z7po9bxpe4.k.tophink.com/@zigbee-dev-guide/jiyuZigBeedewenduhaiqitiduanxinbaojingxitong.html#!>

- Resource screenshots



## 4. Hardware required for the system

- Shaxuefang ZigBee Standard Board
- Shaxuefang ZigBee Mini Board
- Human infrared sensor
- Temperature and humidity sensor
- buzzer
- LED Light
- 0.96 TFT display

The 0.96 TFT display is no longer available. You can switch to a 0.96 OLED display. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

## 5. Build your own system

### 1. PC program

- Find and run the PC-side exe program in the supporting resources

### 2. ZigBee device firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

- Connect the hardware as shown in the demo video or by observing the source code.
- Find the corresponding firmware in the supporting resources and burn the corresponding firmware
- After burning the firmware, power on the development board and connect

After ensuring that all steps are executed correctly, you can see the information of each sensor and realize the alarm function. For specific effects, please see the "System Demonstration Effect" section in this article.

## 9.17.4. ZigBee 3.0 multi-node networking practice

This project is an extracurricular learning content, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this project **in principle**. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

The description of this project is relatively concise and is only suitable for readers with a certain foundation, **not for beginners**

## 1. Field of application

This solution is aimed at **course design** or **function verification demo** areas that have general requirements for communication stability, response speed and communication distance.

## 2. System architecture description and function demonstration

- Architecture and function description: <https://zhuanlan.zhihu.com/p/415439395>
- Functional demonstration: <https://www.bilibili.com/video/BV1UZ4y1Q7M8/>

## 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not publicly available on the Internet. They are currently only provided free of charge to friends who have purchased **the development kit**. <https://z7po9bxpe4.k.tophink.com/@zigbee-dev-guide/jiyuZigBeedewenduheyoushiquiduanxinbaojingxitong.html#!>

- Resource screenshots



## 4. Hardware required for the system

- ZigBee MiNi board × 1
- ZigBee standard board × 4

## 5. Build your own system

### 1. Install the serial port debugging assistant on the PC

according to "Part 1" of this tutorial.

### 2. ZigBee device firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide".

(1) Find the corresponding firmware in the supporting resources and burn the coordinator firmware to the Mini board and the terminal firmware to the standard board.

(2) After burning the firmware, power on each development board to realize multi-node networking and data communication functions. For architecture and function description, please refer to:

- Architecture and function description: <https://zhuanlan.zhihu.com/p/415439395>

## 6. System source code explanation

- Video explanation: <https://www.bilibili.com/video/BV1UZ4y1Q7M8?p=3>

The author of this system: Li Xiang

## 9.17.5. Temperature, humidity & signal strength detection system based on ZigBee

This project is **an extracurricular learning content**, its content is relatively complex and has a certain degree of difficulty, therefore:

If you have not learned the previous chapters, please stop studying.

Shanxuefang does not provide technical support for this project **in principle**. Please understand.

If you need assistance in building this system, you will need to pay additional technical support fees. Please contact the project manager at the end of this article for details.

### 1. Field of focus

This solution is aimed at **course design** or **function verification demo** areas that have general requirements for communication stability, response speed and communication distance.

### 2. System Architecture Description and Function Demonstration

The system can detect ambient temperature and humidity, as well as ZigBee wireless signal strength

- System architecture: <https://zhuanlan.zhihu.com/p/394984198>

### 3. Download supporting resources

The **supporting resources** of this project are **additional parts** and are not publicly available on the Internet. They are currently only provided free of charge to friends who have purchased **the development kit**. <https://z7po9bxpe4.k.tophink.com/@zigbee-dev-guide/jiyuZigBeedewenduheyoushidianxinxibaopingxitong.html#!>

- Resource screenshots



### 4. Hardware required for the system

- ZigBee MiNi board × 1
- ZigBee standard board × 2
- 0.96 TFT display

TFT screens are no longer available. You can switch to OLED screens. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

## 5. Build your own system

### 1. PC program

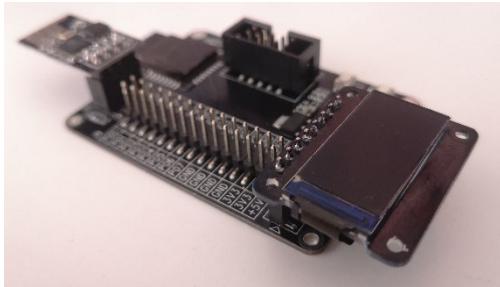
- Find and run the PC-side exe program in the supporting resources

### 2. ZigBee device firmware burning

This step requires some experience in ZigBee development. If you do not have it, please first study the "ZigBee 3.0 Development Guide"

1. Connect the hardware according to the steps (the new version of the development board is used here to demonstrate the hardware connection)

- Insert the 0.96 TFT display into the ZigBee Mini board as shown in the picture



The 0.96 TFT display is now out of stock. You can switch to an OLED screen. For more information, please refer to: <https://bbs.csdn.net/topics/603889850>

- The standard board has its own temperature and humidity sensor, so no additional connection is required.

2. Find the corresponding firmware in the supporting resources, burn the gateway firmware to the Mini board and the terminal firmware to the standard board

3. After burning the firmware, power on the development board and connect

After ensuring that all steps are performed correctly, you can check the temperature, humidity, and ZigBee wireless signal strength information.

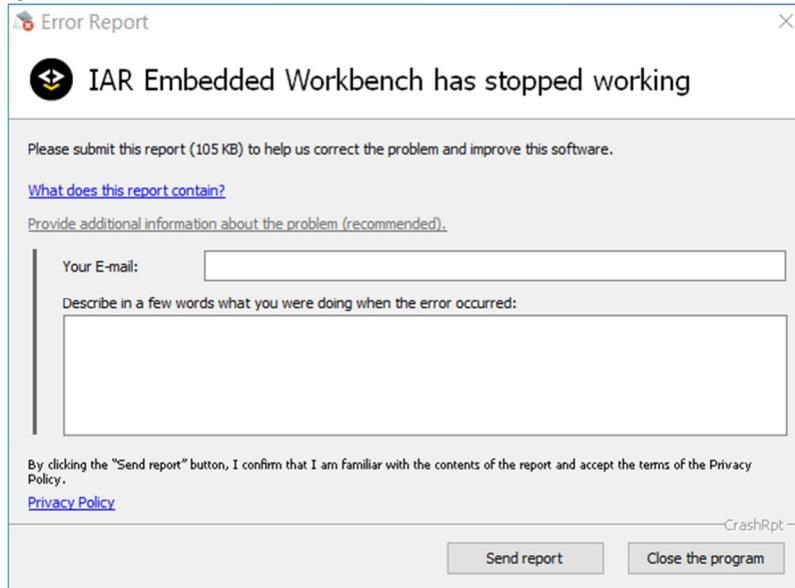
## 9.18. IAR EW for 8051 FAQ & Solutions

This part is **extracurricular content**, which is relatively complex and difficult. Therefore:

- (1) The Taobao customer service lady does not have any technical knowledge and will not answer any technical questions.
- (2) If you have not learned parts 1 to 4, you must stop reading!
- (3) In principle, this part does not have any technical support. Please understand!

## IAR EW has stopped working

## question:



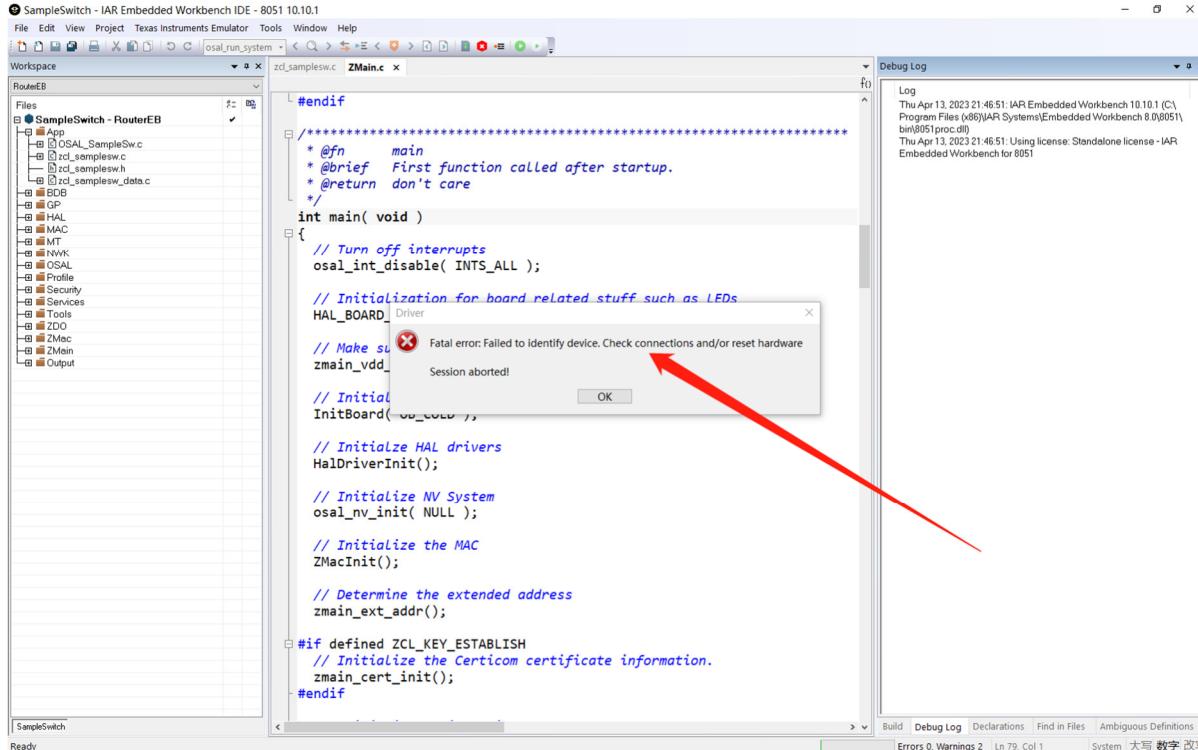
## Solution:

Click Close the program, then press the RST button in the emulator, and finally re-emulate or burn the program.

## Failed to identify device

### Problem:

Fatal error: Failed to identify device. Check connections and/or reset hardware.



## Solution:

Click OK, then press the RST button in the emulator, and finally re-emulate or burn the program.

## This device has been locked for Debugging...

## question:

The screenshot shows the IAR Embedded Workbench IDE interface. In the center, there is a code editor window displaying C code for a project named 'SampleSwitch - CoordinatorEB'. A modal dialog box titled 'Driver' is overlaid on the code editor. The dialog contains the following text:  
This device has been locked for debugging. To enable debugging and continue with download, the device need to be erased.  
To erase and continue press Ok, or Cancel to abort.

At the bottom right of the dialog are two buttons: '确定' (Confirm) and '取消' (Cancel). A red arrow points from the text above to the '确定' button.

## Solution:

Click OK, then press the RST button in the emulator, and finally re-emulate or burn the program.

## Stuck, no response

## question:

The screenshot shows the IAR Embedded Workbench IDE interface. In the center, there is a code editor window displaying C code for a project named 'SampleSwitch - RouterEB'. On the right side of the interface, there is a 'Build' window with the following content:  
Messages  
Building configuration: SampleSwitch - CoordinatorEB  
Updating build tree.  
Configuration is up-to-date.

## Solution:

Wait for the following prompt to appear, click to close the program and then reopen the project, as shown in the figure.

