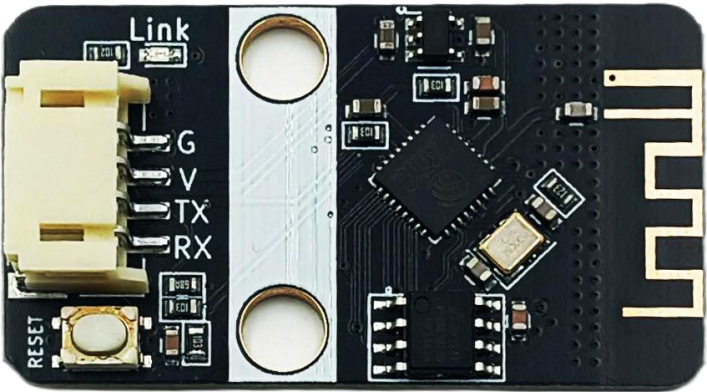
# MQTT物联网模块实验

## 模块介绍

 ESP8266-MQTT无线模块是emakefun公司基于乐鑫科技的wifi芯片ESP8266基础上重新研发的串口转wifi的物联网模块，该模块采用AT配置方式来支持wifi无线通信，AT指令全面兼容[乐鑫官方指令库（V3.0.0）](https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_cn.pdf)，在此基础上添加了MQTT指令，并且全部封装成scratch，mixly，Makecode图形化编程块支持arduino，micro:bit。让用户非常容易接收和发送物联网信息，远程物联网控制从未如此简单。详细资料请查看技术文档：<https://emakefun-docs.readthedocs.io/zh-cn/latest/sensors/smart_modules/esp8266_mqtt/>



**模块特点**

·内置低功率 32 位 CPU：可以兼作应用处理器

·内置协议：TCP/IP 协议栈

·加密类型：WPA WPA2/WPA2–PSK

·支持乐鑫官方AT标准指令集

·支持连接标准MQTT协议和TTL串口到无线的应用

**模块参数**

·工作电压：5V

·接口速率：9600 bps

·无线频率：2.4GHz

·接口类型：PH2.0-4Pin (G V TX TX)

·无线模式：IEEE802.11b/g/n

·SRAM：160KB

·外置Flash：4MB

·支持低功耗：<240mA

·模块尺寸：4 \* 2.1cm

·安装方式：M4螺钉螺母固定

## 实验目的

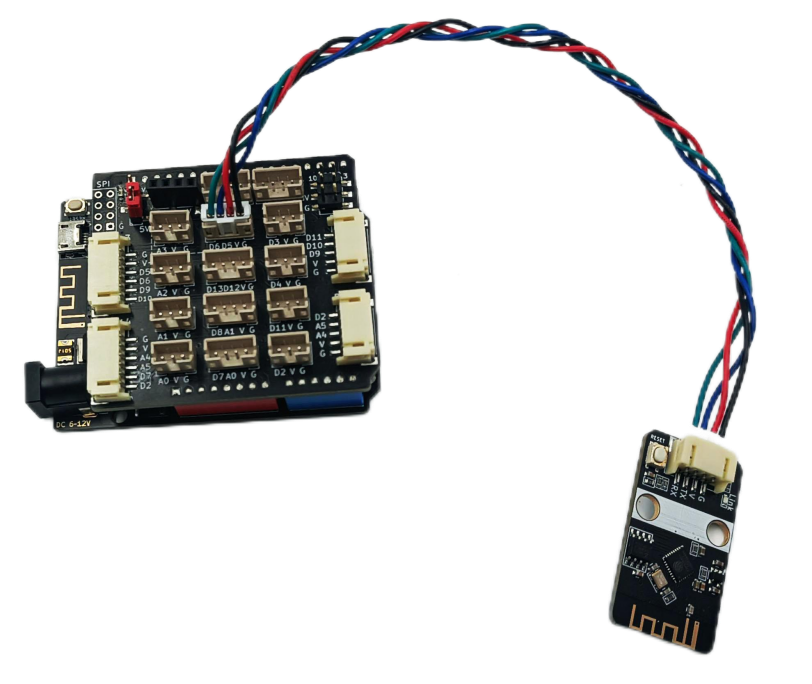
通过IC2扩展板控制和读取常见设备，了解IC2扩展板的工作原理。

## 组件清单

* PH2.0 Sensor 扩展板：1
* BLE-UNO主板：1
* MQTT物联网模块：1
* USB 数据线 ：1
* 若干连接线

## 实物接线图

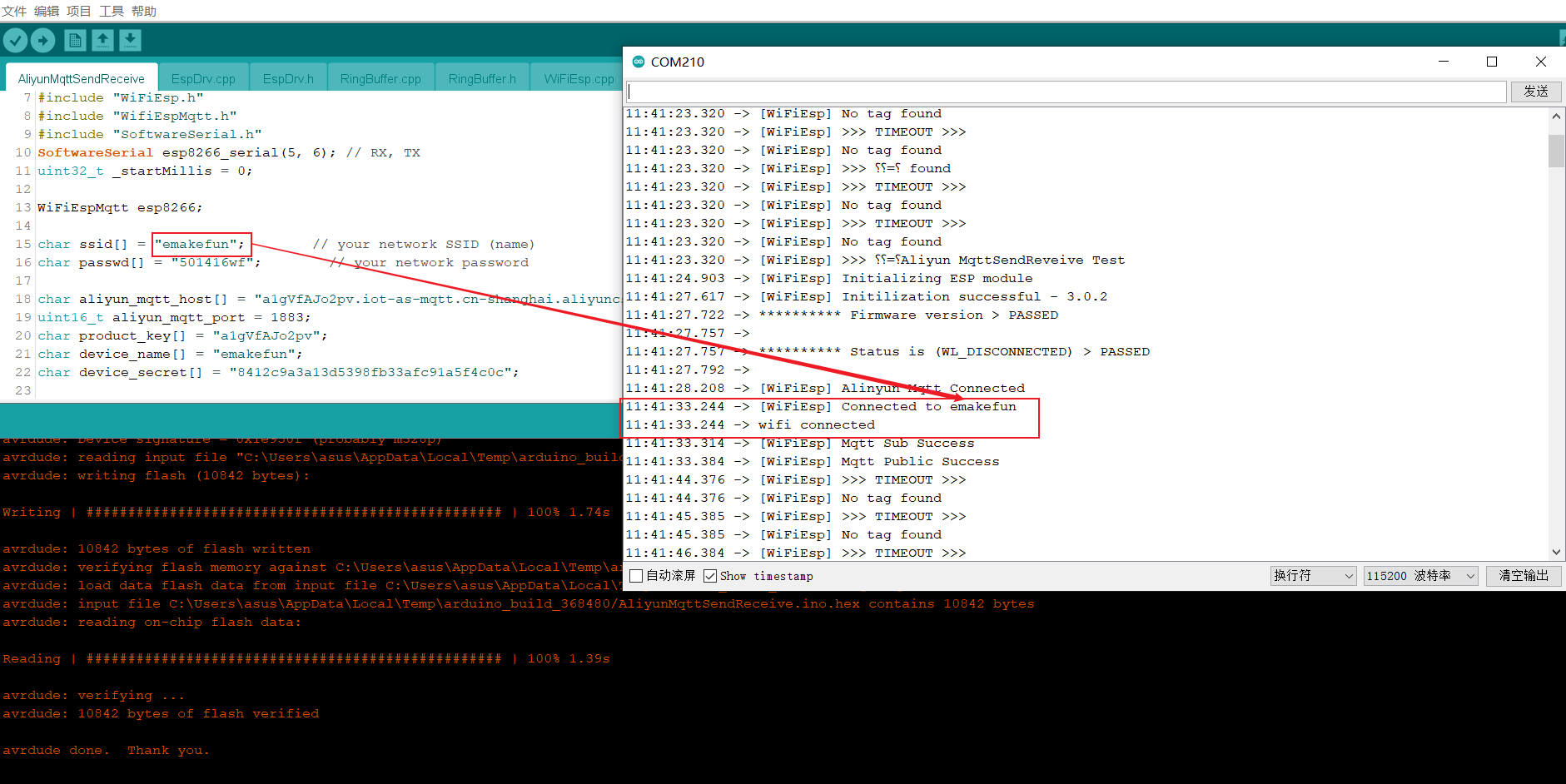
|  |  |  |
| --- | --- | --- |
| 引脚名称 | 描述 | Arduino uno |
| G | GND地线 |  |
| V | 5V电源引脚 |  |
| TX | 串口发送端 | D5 |
| RX | 串口接收端 | D6 |



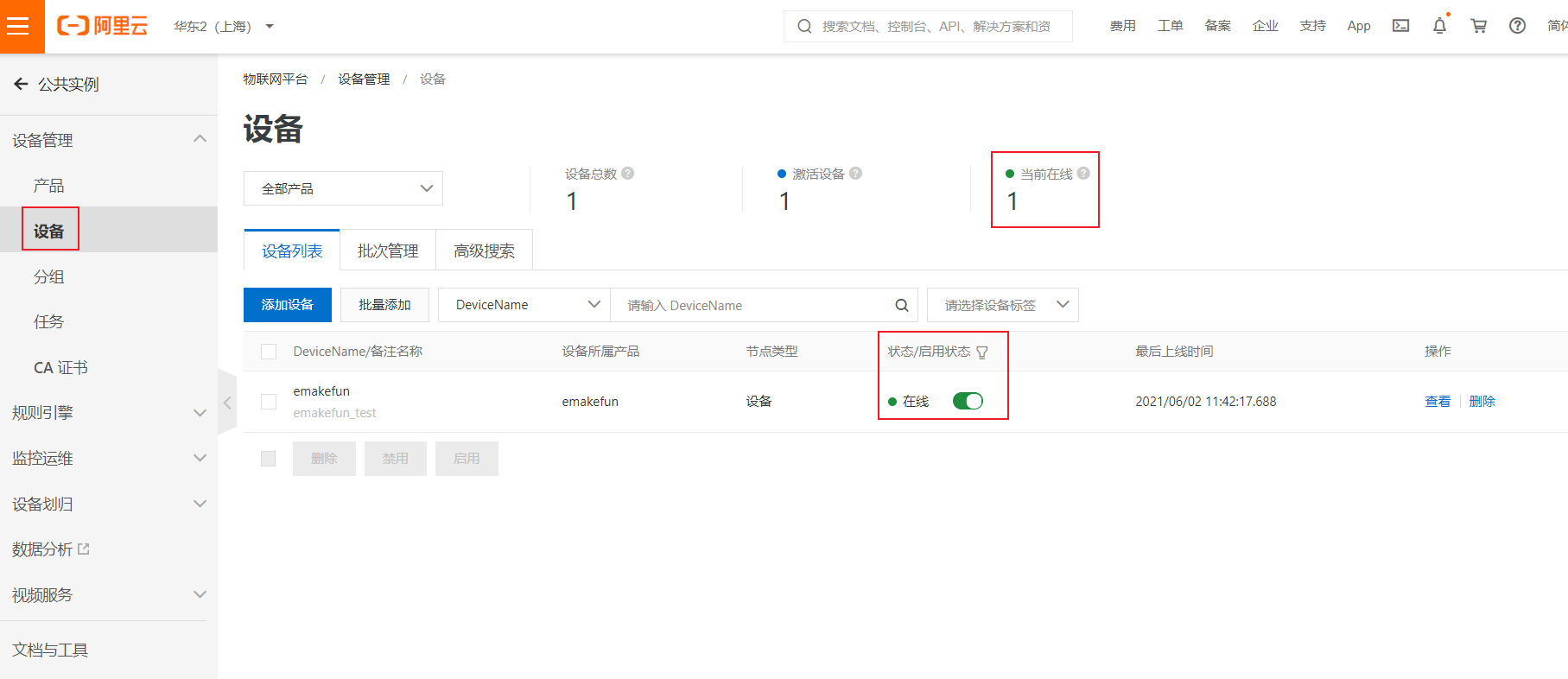
## 程序代码（示例程序）

|  |
| --- |
| /\*  WiFiEsp test: BasicTest  Performs basic connectivity test and checks.  \*/  #include "WiFiEsp.h"  #include "WifiEspMqtt.h"  #include "SoftwareSerial.h"  SoftwareSerial esp8266\_serial(5, 6); // RX, TX  uint32\_t \_startMillis = 0;  WiFiEspMqtt esp8266;  char ssid[] = "emakefun"; // your network SSID (name)  char passwd[] = "501416wf"; // your network password  char aliyun\_mqtt\_host[] = "a1gVfAJo2pv.iot-as-mqtt.cn-shanghai.aliyuncs.com"; // 阿里云物联网服务器host  uint16\_t aliyun\_mqtt\_port = 1883; // 阿里云物联网服务器端口  char product\_key[] = "a1gVfAJo2pv"; // 设备所属产品的ProductKey，即物联网平台为产品颁发的全局唯一标识符  char device\_name[] = "emakefun"; // 设备在产品内的唯一标识符。DeviceName与设备所属产品的ProductKey组合，作为设备标识，用来与物联网平台进行连接认证和通信。  char device\_secret[] = "8412c9a3a13d5398fb33afc91a5f4c0c"; // 物联网平台为设备颁发的设备密钥，用于认证加密。需与DeviceName成对使用。  void setup()  {  Serial.begin(115200);  esp8266\_serial.begin(9600);  Serial.println("Aliyun MqttSendReveive Test");  WiFi.init(&esp8266\_serial);  assertEquals("Firmware version", WiFi.firmwareVersion(), "3.0.2");  assertEquals("Status is (WL\_DISCONNECTED)", WiFi.status(), WL\_DISCONNECTED);  esp8266.mqtt\_connect\_aliyun(aliyun\_mqtt\_host, aliyun\_mqtt\_port, product\_key, device\_name, device\_secret, 0);  if (WiFi.begin(ssid, passwd) == WL\_CONNECTED)  {  Serial.println("wifi connected");  esp8266.mqtt\_sub("/a1gVfAJo2pv/emakefun/user/get", 0); // 订阅topic  }  esp8266.mqtt\_public("/a1gVfAJo2pv/emakefun/user/add", "on", 0); // 发布topic 数据为 "on"  delay(10000);  }  void loop()  {  if (esp8266.mqtt\_receive())  {  Serial.print("topic:");  Serial.println(esp8266.mqtt\_topic); // 打印订阅的topic  Serial.print("message:");  Serial.println(esp8266.mqtt\_message); // 打印订阅的topic的数据  }  }  void assertNotEquals(const char\* test, int actual, int expected)  {  if(actual!=expected)  pass(test);  else  fail(test, actual, expected);  }  void assertEquals(const char\* test, int actual, int expected)  {  if(actual==expected)  pass(test);  else  fail(test, actual, expected);  }  void assertEquals(const char\* test, char\* actual, char\* expected)  {  if(strcmp(actual, expected) == 0)  pass(test);  else  fail(test, actual, expected);  }  void pass(const char\* test)  {  Serial.print(F("\*\*\*\*\*\*\*\*\*\* "));  Serial.print(test);  Serial.println(" > PASSED");  Serial.println();  }  void fail(const char\* test, char\* actual, char\* expected)  {  Serial.print(F("\*\*\*\*\*\*\*\*\*\* "));  Serial.print(test);  Serial.print(" > FAILED");  Serial.print(" (actual=\"");  Serial.print(actual);  Serial.print("\", expected=\"");  Serial.print(expected);  Serial.println("\")");  Serial.println();  delay(10000);  }  void fail(const char\* test, int actual, int expected)  {  Serial.print(F("\*\*\*\*\*\*\*\*\*\* "));  Serial.print(test);  Serial.print(" > FAILED");  Serial.print(" (actual=");  Serial.print(actual);  Serial.print(", expected=");  Serial.print(expected);  Serial.println(")");  Serial.println();  delay(10000);  } |

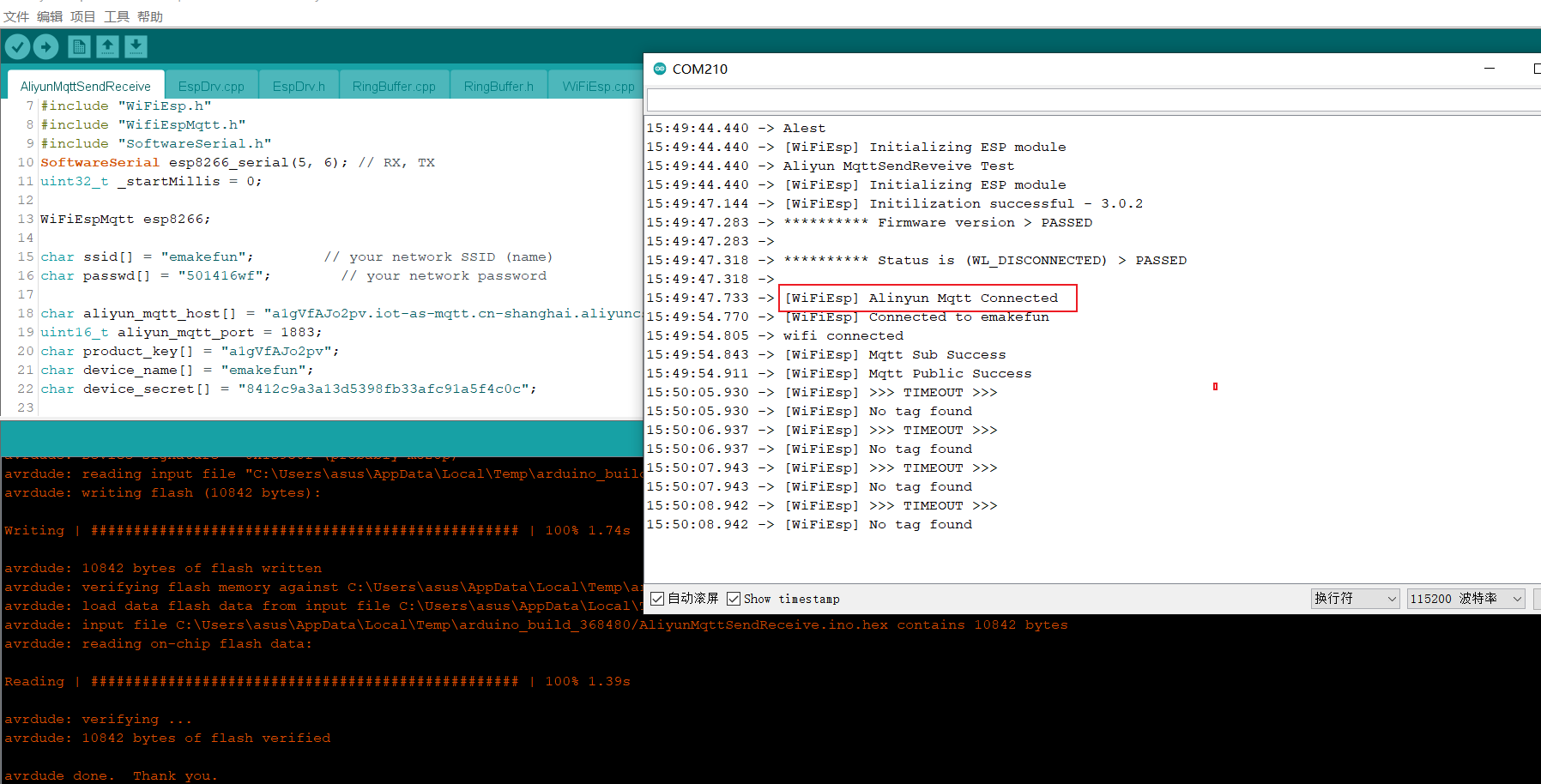
当连接上WIFI时，物联网模块的蓝灯会常亮，否则蓝灯会闪烁，同时串口监视器会显示WiFi connected,如下图所示。



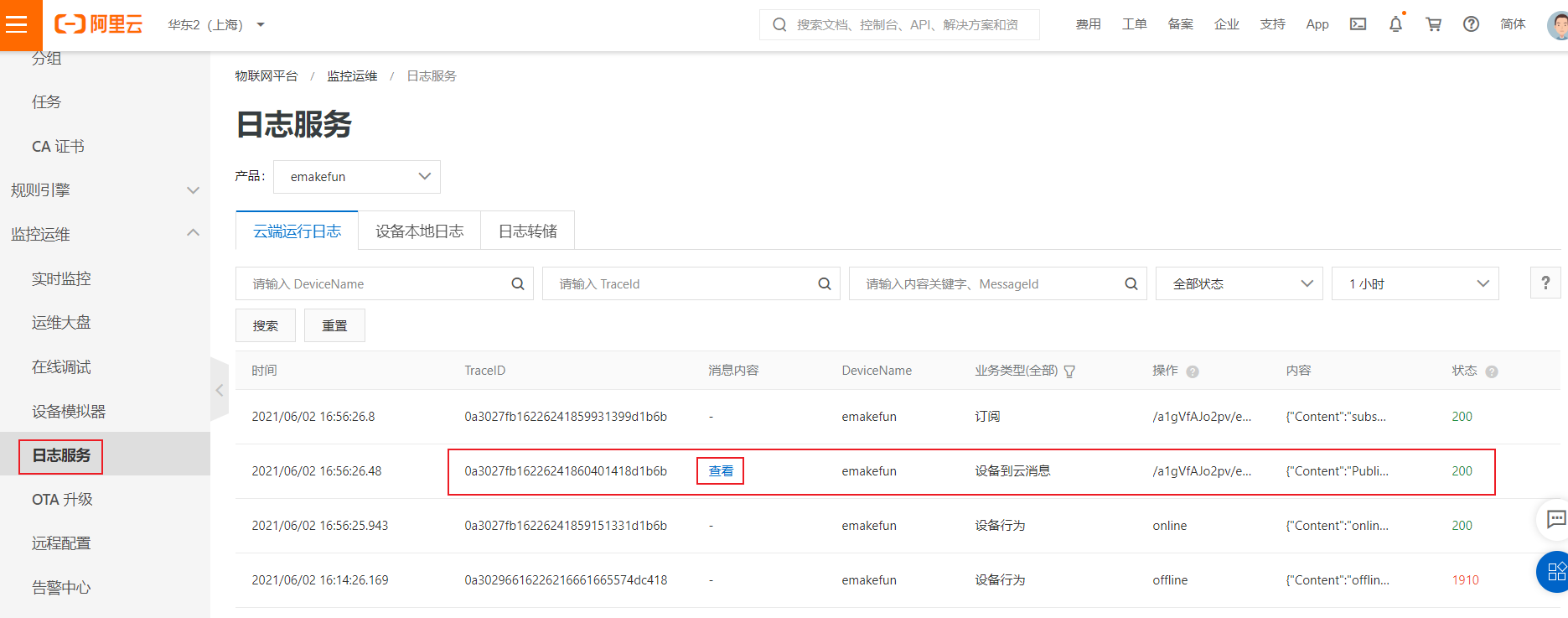
当连上阿里云服务器时，选择阿里云设备菜单时，会显示当前在线的设备数量，并且所连接的设备的状态为在线状态，如下图所示。

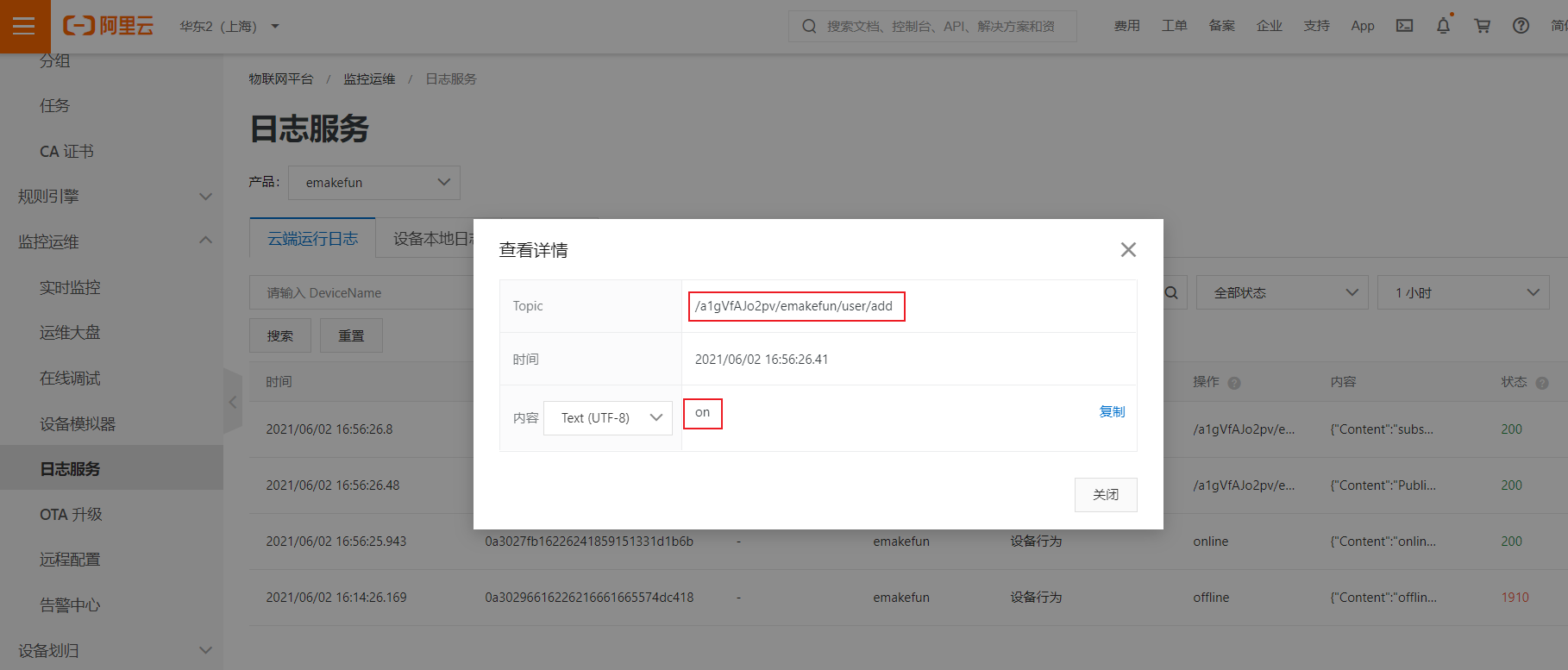


同时在串口监视器里面可以看到是否连接成功，如下图所示。

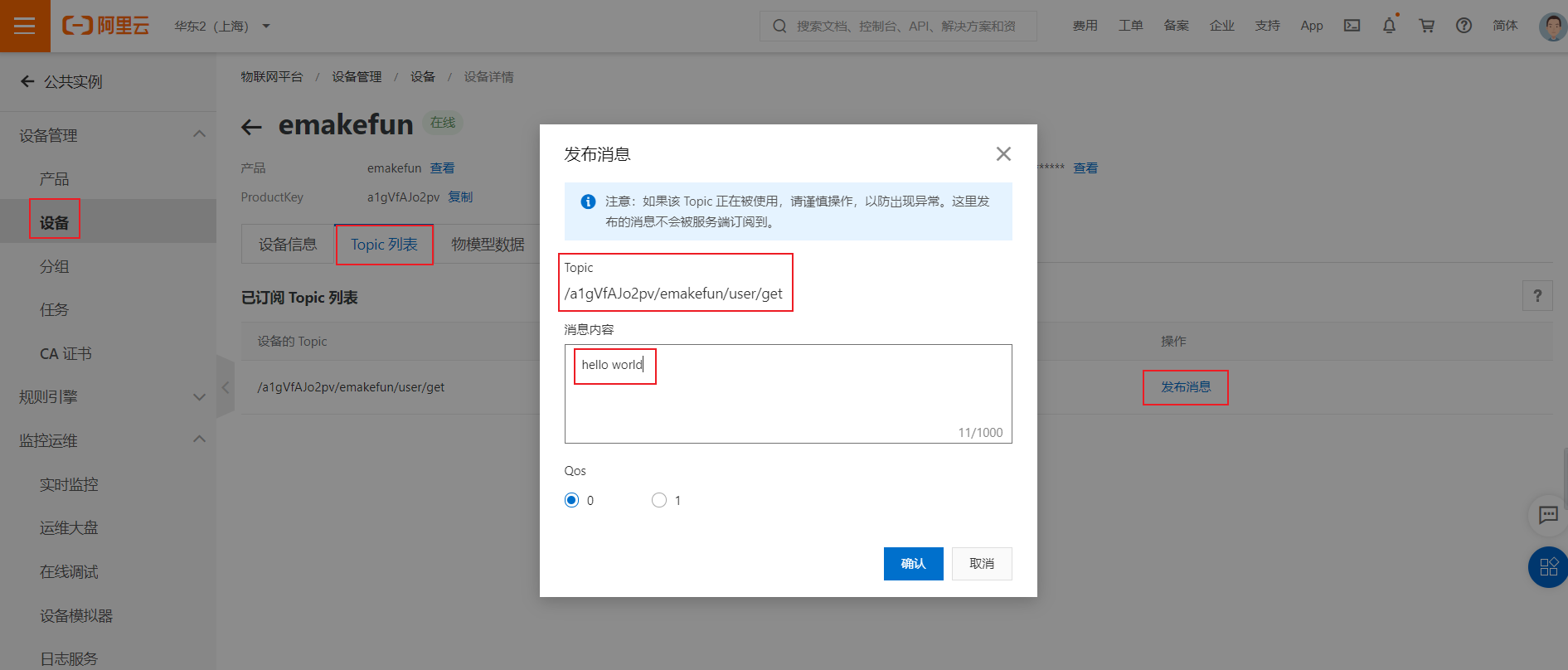


当前程序会在连接阿里云服务器成功之后，会向服务器发布和订阅相关的主题(topic)。 发布: 向云端发送数据。比如温度、湿度、气压值、停车位...... 可以在阿里云的监控运维->日志服务里面可以看到当前发布的记录，并且点击查看可以看到发送的数据。

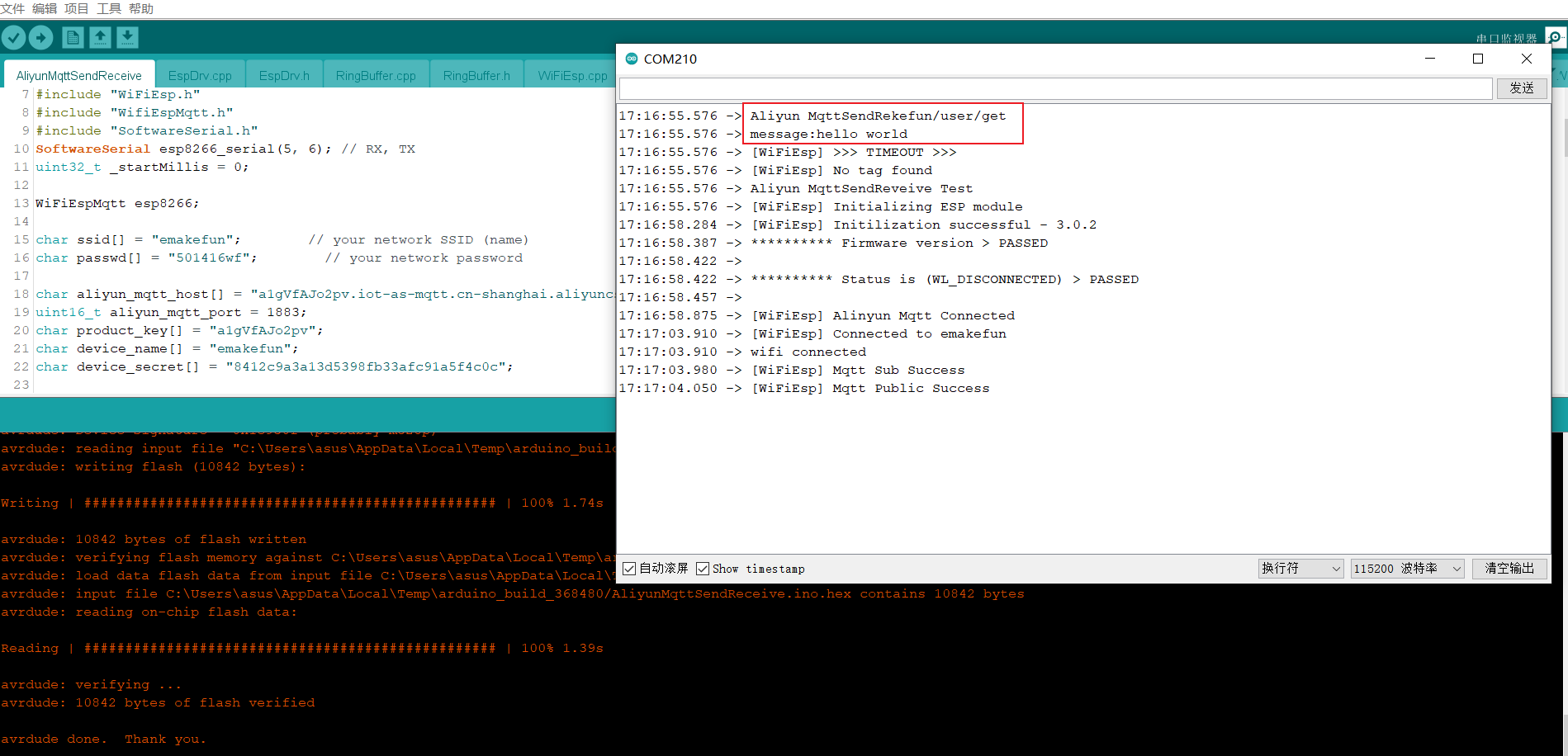




订阅: 获取云端的数据。比如天气预报..... 可在设备的topic列表里找到相应的topic，并且点击发布消息，发布想要发布的数据。



同时在串口监视器里面我们可以看到订阅的数据。



## MagicBlock图形化编程程序

MagicBlock编写MQTT物联网模块测试程序如下图所示：

**Mind+图形化编程**

Mind+编写MQTT物联网模块测试程序如下图所示：

## Mixly图形化编程

Mixly编写MQTT物联网模块测试程序如下图所示：