

BC7215AC

Arduino 空调遥控库

应用示例

目录

ESP8266:.....	3
ESP32:.....	4
Arduino NANO33 IoT:.....	4
串口监视器版.....	5
ESP32 LCD 版.....	9
ESP32 MQTT 版.....	11
1. 程序编译前准备.....	11
2. 网络控制空调.....	12
MQTT 客户端.....	13
空调状态上报.....	14
ESP32 Home Assistant 版.....	14
ESP8266 Home Assistant 版.....	15

BC7215AC 空调遥控库提供了 5 个应用的例子，多数例子都提供了英文和中文两个版本，分别为：

- ESP8266 串口监视器非阻塞版
- ESP32 串口监视器版
- Nano33 IoT 串口监视器版(英文)
- ESP32 LCD 版
- ESP32 MQTT 版
- ESP32 Home Assistant 版(英文)
- ESP8266 Home Assistant 版

串口监视器版为最简单的演示，仅需将任何 ESP8266 或者 ESP32 的 Arduino 开发板连接 BC7215A 的红外收发板，然后以 Arduino IDE 自带的串口监视器作为人机交互手段，即可实现控制空调的功能。

LCD 板和 MQTT 板要求使用 LilyGO TTGO T-Display 的 ESP32 开发板，该开发板自带一个 LCD 显示屏和 2 个实体按键，演示程序利用这些外部元件作为交互手段，从而可以不依赖电脑完全独立运行。MQTT 版在 LCD 版的基础上，增加了 MQTT 协议联网功能，用户可以通过公共 MQTT 代理，测试通过网络控制空调功能，同时 MQTT 版也保留 LCD 版的本机按键操作功能，可将操作后的空调状态上传至 MQTT 服务器。

例程中使用了 ESP8266 Node MCU 板和 ESP32 TTGO T-Display 板作为硬件，各个版本通用。硬件连接如下：

ESP8266:

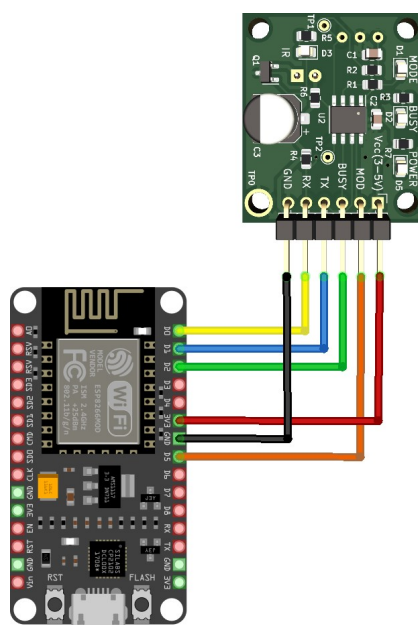
GPIO5 - BC7215A TX

GPIO16 - BC7215A RX

GPIO14 - BC7215A MOD

GPIO4 - BC7215A BUSY

3.3V - BC7215A VCC



ESP32:

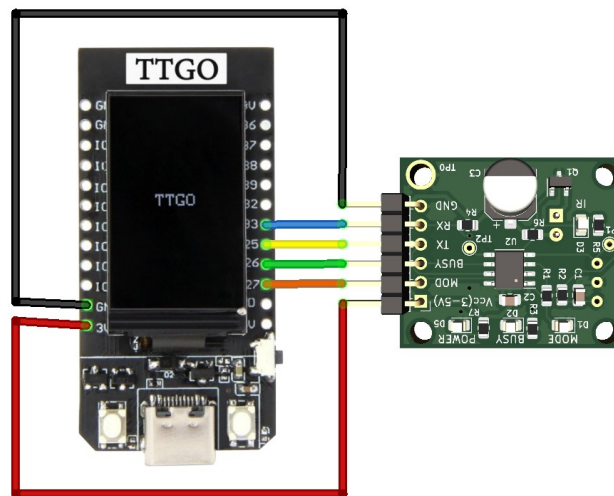
GPIO25 - BC7215A TX

GPIO33 - BC7215A RX

GPIO27 - BC7215A MOD

GPIO26 - BC7215A BUSY

3.3V - BC7215A VCC



fritzing

Arduino NANO33 IoT:

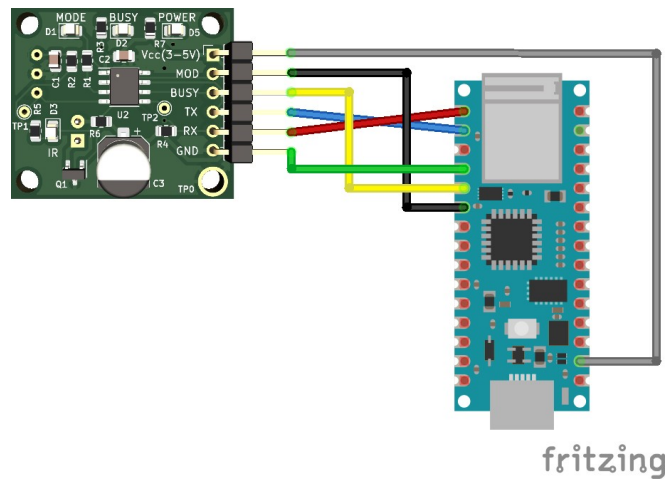
RX - BC7215A TX

TX - BC7215A RX

D3 - BC7215A MOD

D2 - BC7215A BUSY

3.3V - BC7215A VCC



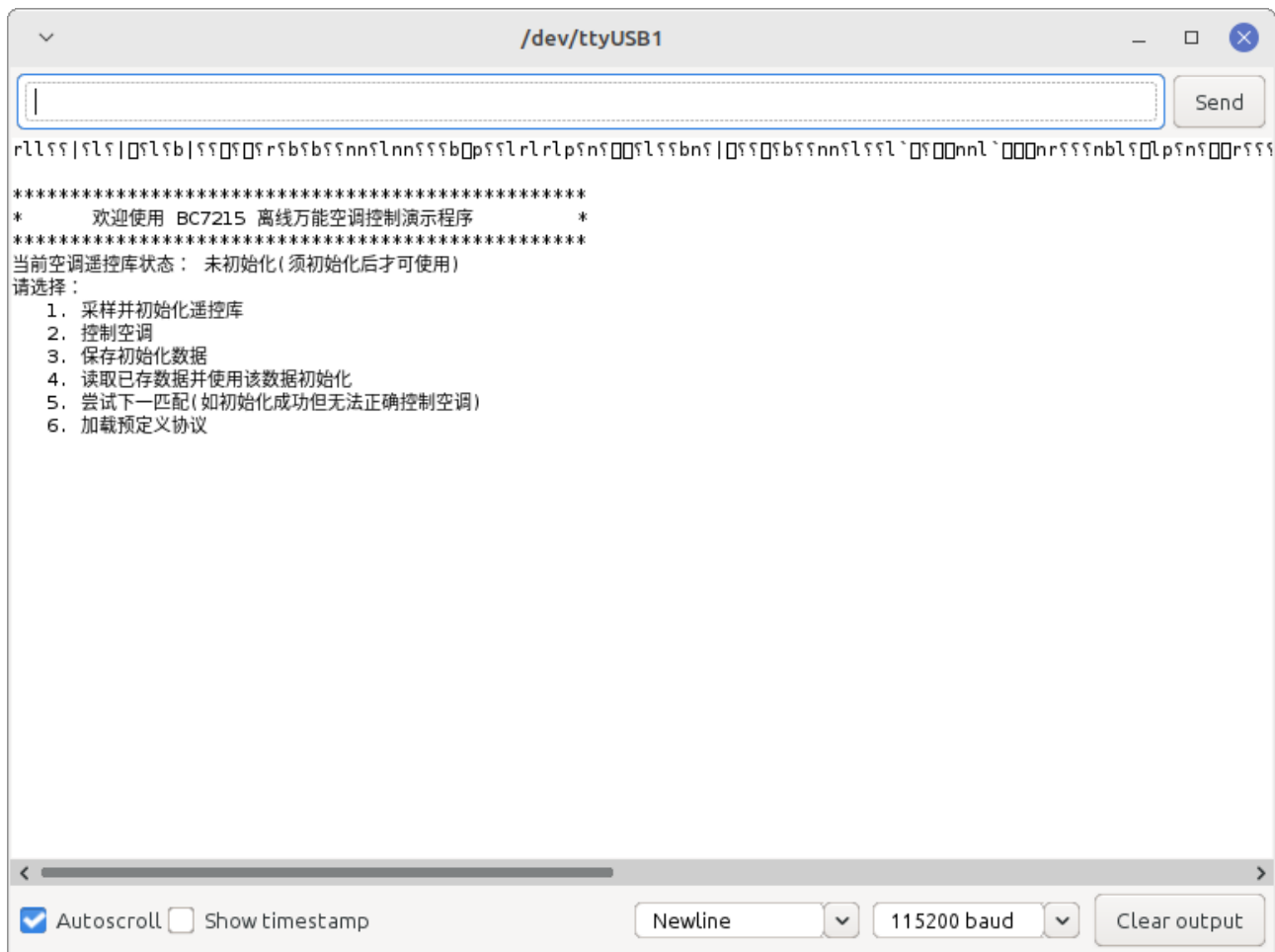
下面详细解说各个版本的使用：

串口监视器版

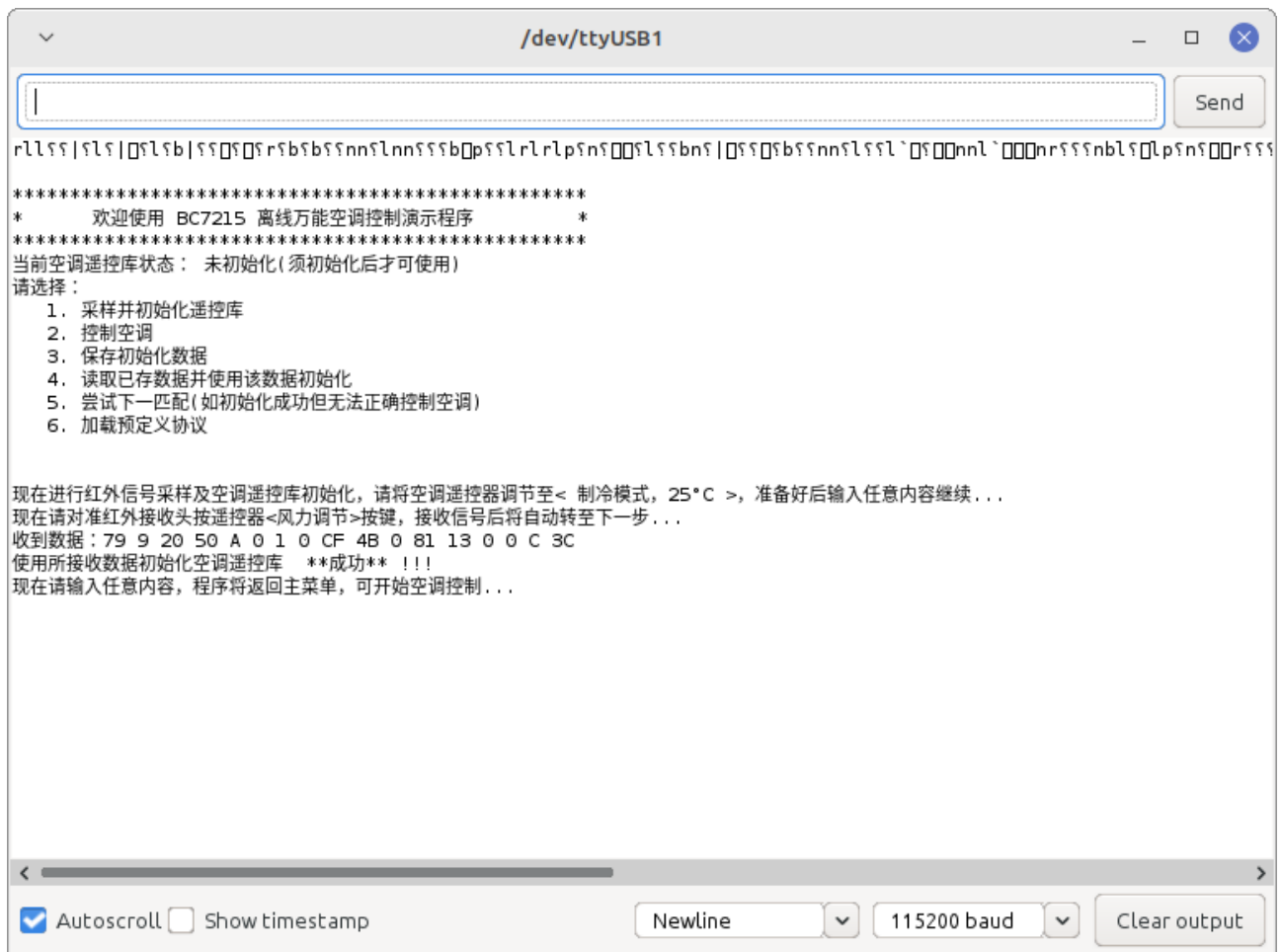
通过 Arduino IDE 的串口监视器使用，串口波特率为 115200. 提供了 3 种不同的程序：ESP8266 阻塞式设计，ESP8266 非阻塞式设计，以及 ESP32 版。阻塞式设计逻辑上最为简单，在完成一个操作之前，比如等待用户输入，程序会做无限循环查询，直到条件满足，适于用作了解驱动库的使用流程。非阻塞版使用状态机机制，程序可以“同时”完成多项任务，不会卡在某个环节。ESP32 版是 ESP8266 非阻塞版的 ESP32 移植。

不同版本的菜单界面相同。

程序上传到 Arduino 板后，即可在串口监视器内看到主菜单，如果没有显示，是因为程序上传后串口监视器的启动迟于程序的启动，此时直接输入回车或者重启 Arduino 板，即可看到主菜单显示



初次使用时，应进行被控空调的遥控器的采样，并用采样数据进行遥控库的初始化



初始化会分步进行，用户按照屏幕提示按步骤完成即可。通常将会初始化成功，如果多次尝试失败，且检查空调遥控器设置没有错误，则有可能所用的空调是极少数 BC7215A 无法直接解码的型号，这时可以逐个尝试使用“预定义协议”，测试是否可以控制。

```

/dev/ttyUSB1

*****
*      欢迎使用 BC7215 离线万能空调控制演示程序      *
*****
当前空调遥控库状态： ***已初始化***
请选择：
  1. 采样并初始化遥控库
  2. 控制空调
  3. 保存初始化数据
  4. 读取已存数据并使用该数据初始化
  5. 尝试下一匹配(如初始化成功但无法正确控制空调)
  6. 加载预定义协议

有少数协议不支持直接由BC7215A芯片解码，以预定义协议方式提供。
当直接采样不成功或初始化失败时，可尝试使用预定义数据控制空调。
协议名称中所列品牌仅供参考，如不能控制，请逐个尝试所有预定义协议
请选择：
  0. M96b (XIAOMI)
  1. M100b (SHINCO/SAMSUNG/ELECTROLUX)
  2. D131b (RSD/MCQUAY/TICA)
  3. T102b (WHIRLPOOL/BOSCH/AIRWELL)
  4. M128b (FUJITSU/McQUAY)

已选择第 1 种
64 0 2 5 2C 4 D5 56 1 6C 50 D0 12
35 13 9D 24 DD 13 7D 69 3D 9D 1D 2E DB F2 9B BD FE F 2 CE 5F 81 3A A3 22 74 0 75 5 73 0 26 13
初始化成功 !!! 输入任意内容继续

Autoscroll ☐ Show timestamp Newline 115200 baud Clear output
```

初始化成功后即可开始控制空调，控制空调有2级菜单，第一级选择控制类型，是温度等参数，还是开关机，第二级为参数输入，可以输入温度、工作模式、风力大小。



ESP32 LCD 版

LCD 版使用 TTGO T-Display 的 Arduino 板, 自带 135x240 的液晶屏, 驱动器为 ST7789, 例子使用了 Bodmer 的 TFT_eSPI 库, 可在 Arduino IDE 的库管理器中安装, 该库需要用户根据硬件修改一个 User_Setup.h 的文件, 该文件位于安装好的 TFT_eSPI 库的根目录, 适合本例程的设置文件在所安装 BC7215AC 库的 extras/config 目录中, 请将该目录中的 User_Setup.h 文件拷贝到 TFT_eSPI 库的根目录并替换同名文件。

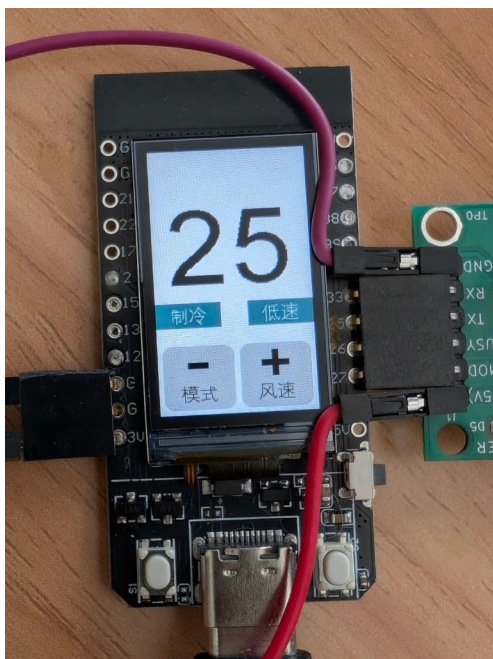
程序初次使用时, 程序运行后, 会进入菜单, 用板上左键 SEL 选择菜单项, 右键 OK 确认。



第一步请先进行配对，按照屏幕提示完成遥控器信号的采集。



如果初始化成功，程序就会进入空调控制页面：



控制页面的按键功能：

左键短按 - 温度减

右键短按 - 温度加

左键长按 - 切换模式

右键长按 - 切换风速

双键短按 - 进入开关机页面，左右键变为开机/关机按键，分别发送开关机指令

双键长按 - 进入菜单

当所做操作会改变空调状态时，程序会驱动 BC7215A 芯片发射相应红外信号，同时屏幕右上角会有红外发射的指示标识。

ESP32 MQTT 版

MQTT 版在 LCD 版的基础上增加了使用 MQTT 协议联网控制和空调状态上报的功能。MQTT 联网功能，使用的是 Nick O'Leary 的 PubSubClient 库，可从 Arduino IDE 的库管理器中安装。

MQTT 版的本机操作部分和 LCD 版完全相同，联网操作部分，请按照以下步骤进行：

1. 程序编译前准备

在程序编译之前，请先去掉源文件中以下几行的注释，并填入自己的内容，否则程序无法正常运行

```
// WiFi 和设备配置 - 请替换为您自己的值  
  
// #define MY_WIFI_SSID      "你的 WiFi 名称" // 替换为您的 WiFi 名称  
  
// #define MY_WIFI_PASSWORD "你的 WiFi 密码" // 替换为您的 WiFi 密码  
  
// #define MY_UUID          "你的 UUID"      // 使用 UUID 生成器创建唯一设备 ID
```

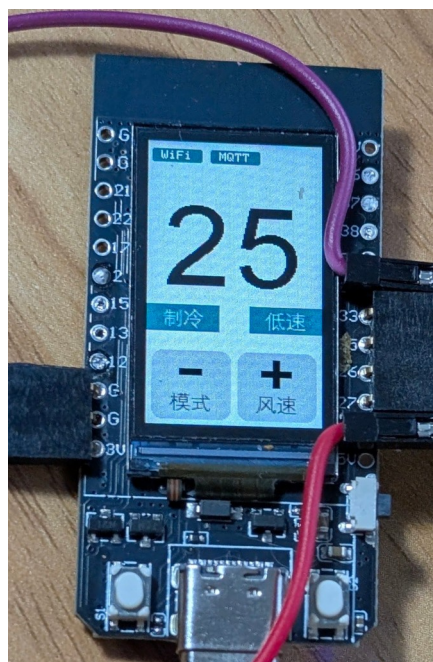
除了 WiFi 的名称和密码，请使用任何一种 UUID 生成器（网上有 UUID 生成器）生成你自己的 UUID，以确保唯一性，否则如果与他人使用相同的 UUID，当他人设备上线时，你的设备会被 MQTT 服务器踢下线。一个 UUID 看起来如下面的格式：b1225e25-81c8-43d7-8183-6f5793408242

程序的 MQTT 代理(服务器)，默认使用 broker.hivemq.com，也可替换为任何支持不加密 1883 端口访问且不需要帐号/密码的公共 MQTT 代理服务器，如中国境内的 broker.emqx.io。

本示例仅为演示使用，使用的是不加密连接，生产环境中建议使用加密连接，以提高安全性。

2. 网络控制空调

程序开始运行后，即开始尝试 WiFi 连接，WiFi 连上后，就会开始尝试 MQTT 服务器的连接，当 MQTT 连接成功后，会在屏幕顶部看两个 WiFi 和 MQTT 的标志，如下图（假设空调遥控库已经初始化完成）：



连接成功后，即可通过 MQTT 协议来控制空调。空调控制的话题(Topic)分别如下：

温度：BC7215A/ (UUID) /var/temp

模式：BC7215A/ (UUID) /var/mode

风力：BC7215A/ (UUID) /var/fan

电源：BC7215A/ (UUID) /var/power

其中 UUID 即为源程序中定义的 MY_UUID. 实际的话题最终为类似下面的样子：

BC7215A/b1225e25-81c8-43d7-8183-6f5793408242/var/temp

发布控制消息时，内容为 ASCII 形式的数字，如温度为字符"16"至"30"，模式为"0"至"4"

temp: 范围 16-30 的整数

mode:

- 0 - 自动
- 1 - 制冷
- 2 - 制热
- 3 - 除湿
- 4 - 送风

fan:

- 0 - 自动
- 1 - 低
- 2 - 中
- 3 - 高

power:

- 0 - 关
- 1 - 开

MQTT 客户端

一般公共免费 MQTT 服务器都会同时提供免费的客户端供用户使用，当然亦可以使用任何用户所习惯使用的 MQTT 客户端发布控制消息。

演示程序使用非加密连接，但发布控制消息的 MQTT 客户端，可以采用任何连接方式和连接端口，仅需连接同一个服务器且话题(topic)相同即可。

hivehq.com 的网页版 MQTT 客户端：<https://www.hivemq.com/demos/websocket-client/>

MQTT Websocket Client x +

www.hivemq.com/demos/websocket-client/

HIVEMQ Websockets Client Showcase

HIVEMQ CLOUD Need a fully managed MQTT broker? Get your own Cloud broker and connect up to 100 devices for free. [Get your free account](#)

Connection connected

Publish

Topic: 2-0f8c-4431-a49f-30a6d9cb3737/var/temp QoS: 0 Retain: ☐ [Publish](#)

Message: 24

Subscriptions

[Add New Topic Subscription](#)

Qos: 2 BC7215A/11741ad2... X

Qos: 2 BC7215A/11741ad2... X

Messages

2025-08-28 21:45:18	Topic: BC7215A/11741ad2-0f8c-4431...	Qos: 0	Temp=24, Mode= COOL , Fan= LOW
2025-08-28 21:44:16	Topic: BC7215A/11741ad2-0f8c-4431...	Qos: 0	Retained online

空调状态上报

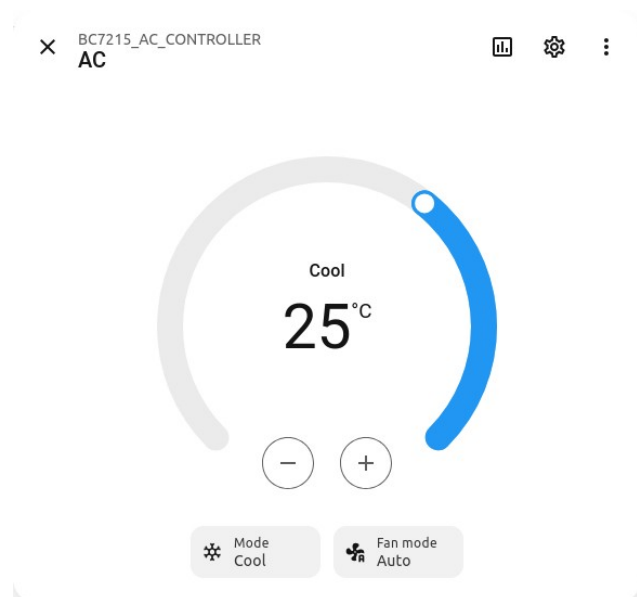
每次发射红外指令改变空调状态，包括通过本机上按键操作，都会同时将状态上报至 MQTT 服务器，如果客户端订阅了报告话题，就会在客户端看到更新后的空调状态。演示程序的报告话题是：

BC7215A/ (UUID) /var/report

ESP32 Home Assistant 版

在 ESP32 MQTT 版的基础上，将 MQTT 的参数针对 Home Assistant 的要求进行了修改，在使用 Home Assistant 且为其安装了 MQTT 插件的情况下，能够自动接入 MQTT 且被其发现而无需任何设置，能够以即插即用的方式并入 HA 的系统，并用 HA 控制空调。

当上电且空调配对成功后，Home Assistant 中，就会自动出现空调的控制器



程序中配置为使用本地局域网的 MQTT 服务器，因为在使用 Home Assistant 的情况下，最佳的选择就是使用本地的 MQTT 服务器，这样因为无安全方面的担忧可以免去帐号密码以及加密连接等一系列繁琐的设置。

Home Assistant 版保留了 ESP32 MQTT 版的所有操作，只是将 MQTT 服务器设置和 MQTT 主题等按照 Home Assistant 的需求进行了修改。

ESP8266 Home Assistant 版

在 ESP32 版的基础上，去掉了 LCD 显示和按键操作部分，仅使用板上的唯一按键作为重新配对的按钮，当长按 2s 后，程序会进入配对模式，此时用户直接对着红外接收头发射“25°C 制冷模式”的信号即可。因为去掉了显示，仅以板载的一个 LED 作为工作状态指示：

- LED 慢闪 —— 未连接 BC7215A
- LED 较快闪烁 —— 配对状态，等待接收红外信号
- LED 每 3s 短暂闪烁一下 —— 已配对成功，指令解析状态，等待接收红外信号

因为缺少了用户交互，因此省略了很多操作，如选择“选择温度制式”，“下一匹配”和“使用预定义数据”，但也足以覆盖绝大多数的情况了。同时，因为没有在本机操作控制空调的功能，配对成功后，会直接进入解析模式，如果用户用红外遥控器操作空调，最新空调状态会同步到 HA。