

**BC7215AC**

**Arduino**

**万能空调遥控库**

**V7.x**

# 目录

简介.....	3
空调遥控协议简介.....	3
BC7215AC Arduino 空调遥控库使用方法.....	4
第一步：与空调机配对.....	4
第二步：设置需要的工作状态，或解析红外信号.....	4
最佳实践.....	4
系统要求.....	4
空调遥控库详解.....	4
组成.....	4
硬件连接.....	5
公开属性.....	5
接口函数.....	5
1. 开始/停止采样函数.....	5
2. 采样状态查询函数.....	6
3. 初始化函数.....	6
4. 空调设置函数.....	6
温度 temp:.....	6
模式 mode:.....	6
风力 fan:.....	7
按键 key:.....	7
5. 空调开关函数.....	7
6. 红外指令解析函数.....	8
6. 寻找下一匹配函数.....	8
8. 预定义协议.....	8
获取预定义协议数量.....	8
获取预定义协议的名称.....	8
使用预定义协议初始化.....	9
9. 获取版本信息函数.....	9
10. 获取基础数据包及获取基础格式包.....	9
11. 设置温度制式.....	9
12. 查询当前温度制式.....	9
13. 查询 BC7215A 芯片状态.....	9
编程流程.....	10
问题提交.....	12

## 简介

BC7215AC Arduino 万能空调遥控库（以下简称空调遥控库），是设计专门用于 Arduino 系统的遥控码库，它基于 c 语言版 BC7215A 离线空调遥控码库，并根据 Arduino 系统的特点进行了封装。驱动库须配合 BC7215A 芯片使用，故本驱动库实际包含了 BC7215 芯片驱动库和万能空调遥控两个库，用户仅需进行 1 次安装。目前版本已经覆盖全部绝大多数空调品牌型号，且还在持续增加中，如果用户遇到尚不支持的机型，通过将反馈表格通过邮件提交后，可在一周内得到升级支持。

有别于其它基于样本数据库的码库，本空调码库使用使用基于编码规则的方式，设置时不是通过品牌型号来选择，而是通过解码红外信号自动识别其编码规则，实现简单的“**配对-控制**”的两步式使用，甚至不必知道空调的品牌型号。基于规则的万能遥控方案具有以下优点：

- 不依赖网络和数据库，完全离线，可运行于低配置单片机上。
- 未来型号天然支持。空调红外遥控器功能基本固定，新产品大概率会沿用现有的控制协议。
- 越少见品牌被支持概率越大，小品牌通常会选用大厂家的成熟 OEM 方案，而不是自行开发。

本空调遥控库无需联网除 BC7215A 芯片外不依赖任何外部资源，可运行于任何有足够程序容量的 Arduino 系统上。随库提供的例子包括在 ESP8266、ESP32 以及 NANO33 IoT 模块上的应用。

驱动库提供 4 个基本的空调控制功能：

- 控制温度
- 控制工作模式
- 控制风力大小
- 控制电源开关

同时，还有空调红外信号解析（解码）功能，能够从遥控器的红外信号中提取出温度、工作模式、风力以及电源状态 4 个参数。

## 空调遥控协议简介

空调的红外遥控，总体上分为两种，一种是固定码遥控，其特点是遥控器上每个按键发射的红外编码固定，判断特征是这类遥控器上没有液晶显示屏。这类遥控器和普通影音设备的红外遥控器本质一样，仅需简单复制其红外信号即可，使用 BC7215(A)的基本功能即可实现，无需专门的驱动库，用户可参考 BC7215 芯片 Arduino 驱动库中提供的"学习型遥控器"例子。

另外一种是最常见的，可变编码遥控，其特点是遥控信号的长度很长，每一帧的遥控信号中，都包含了空调控制的完整信息，如设定温度，工作模式，风力大小等，判断特征也很简单，即这类的空调遥控器上，都会带有一个液晶显示屏。这类的空调遥控器，同一个按键，每次按下时发出的红外编码根据当前的机器工作状态不同是不一样的，而且不同的空调厂家和型号的空调的编码方式，也是不一样的，产生这样的空调遥控器信号，就需要借助本空调遥控库来实现。

# BC7215AC Arduino 空调遥控库使用方法

**注意：**因空调遥控库依赖于BC7215A 芯片，建议阅读下面内容前，用户先对BC7215A 的工作原理及输入输出数据格式，以及BC7215 芯片驱动库的函数有所了解，可以利于对以下内容的理解。相关资料可以在本遥控库的文档目录中找到。

本驱动库的使用非常简单，仅需 2 个步骤，即可完成任意空调的控制。

## 第一步：与空调机配对

接收解码被控空调遥控器的特定信号（制冷模式，25°C/78°F），采样的数据用以供驱动库分析获知原空调遥控信号的格式，即完成与空调机的配对。

**注意：**

不要使用“万能遥控器”作为信号源，因为该类遥控器通常每次按键会连续发出多种不同协议的遥控信号，以达到广泛适用的效果，但这种多重信号会令配对失败。

## 第二步：设置需要的工作状态，或解析红外信号

配对以后，即可以控制空调的工作状态，或解析红外遥控信号。

## 最佳实践

初始化成功后，可以将配对数据保存起来，这样下次可以开机自动完成配对，方法请参阅例程。

## 系统要求

原理上可用于任何有足够程序和 RAM 空间的 Arduino 系统，示例程序使用了 ESP8266、ESP32 以及 NANO33 作为演示。

## 空调遥控库详解

### 组成

BC7215AC 空调遥控库由两个独立的驱动库组成，分为为 BC7215(A)芯片的驱动库，完成基本的对 BC7215(A)芯片的控制操作，BC7215(A)芯片驱动库提供了 4 个演示程序，分别是：

- 任意遥控器解码
- 学习型遥控器
- 2 路红外遥控开关
- 红外数据传输

用于空调控制时，须使用 BC7215A 芯片，不带 A 的 BC7215 芯片不适用于解码空调信号。

BC7215AC 空调遥控库专用于空调控制，提供了 8 个演示程序，分别为：

- ESP8266 演示非阻塞版(使用软件串口与 BC7215A 通讯，并利用 Arduino IDE 的串口监视器作为用户交互渠道，软件为非阻塞式设计)
- ESP32 演示串口监视器英文版(ESP8266 演示的 ESP32 移植)
- ESP32 演示串口监视器中文版(ESP8266 演示的 ESP32 移植)
- ESP32 演示 LCD 英文版(使用 LILYGO T-Display 的板载按键和 LCD 屏)
- ESP32 演示 LCD 中文版(使用 LILYGO T-Display 的板载按键和 LCD 屏)
- ESP32 演示 MQTT 英文版(在 LCD 版基础上增加了联网功能，可利用 MQTT 协议进行空调的联网控制，以及将空调状态进行网络上报)
- ESP32 演示 MQTT 中文版(在 LCD 版基础上增加了联网功能，可利用 MQTT 协议进行空调的联网控制，以及将空调状态进行网络上报)
- Nano33 IoT 英文版，使用串口监视器。
- ESP32 HA (Home Assistant)版 (在 ESP32 MQTT 版基础上修改为可自动接入 HA)
- ESP8266 HA 版 (无 LCD 显示的 HA 版)

## 硬件连接

供需要 4 个连接线：2 个用于 UART，2 根用于 GPIO。实现空调控制的核心器件 BC7215A 芯片，采用串口和 Arduino 控制连接，同时还需要 2 个辅助信号 MOD 和 BUSY，MOD 为 Arduino 的输出信号，控制红外信号是发射还是接收模式，BUSY 是 BC7215A 芯片的输出信号，用来控制串口的数据传送。

## 公开属性

在一个空调控制类 BC7215AC 的实例中，提供了若干公开可访问的属性，包括：

<code>bool</code>	<code>initOK;</code>	—— 布尔类型，表示是否已经与空调正确配对
<code>uint8_t</code>	<code>sampleCount;</code>	—— 8 位无符号整数，为采样所获得的信号的段数，最多 4 段
<code>uint8_t</code>	<code>sampleStatus[4];</code>	—— 8 为无符号整数数组，采样所得每段数据的 status
<code>bc7215DataMaxPkt_t</code>	<code>sampleData[4];</code>	—— 采样所得的各段数据
<code>bc7215FormatPkt_t</code>	<code>sampleFormat[4];</code>	—— 采样所得的各段数据的格式信息

## 接口函数

### 1. 开始/停止采样函数

```
startCapture();
stopCapture();
```

空调遥控库的初始化前，需要采集特定的空调遥控器信号，这两个函数用来控制 BC7215A 芯片进入和退出接收状态。

## 2. 采样状态查询函数

```
bool signalCaptured();
```

用来查询采样是否已经完成（接收到了完整空调遥控信号）。

此函数返回为 bool 值，采样完成时，返回 true，否则返回 false。

## 3. 初始化函数

```
bool init();
```

```
bool init(const bc7215DataMaxPkt_t& data, const bc7215FormatPkt_t& format);
```

该函数用于利用采样的数据初始化遥控库从而完成配对过程。该函数有 2 种形式，最简单的无任何参数的形式，也是最常用的方式，直接使用刚刚采样完存在内部缓冲区的数据完成初始化。第 2 种形式显式地输入遥控信号的原始数据和格式信息，通常用于读取事前保存的配对信息后初始化遥控库。

返回参数为一个 bool 值，为 true 时，表示初始化成功，false 时表示初始化失败。

初始化的输入采样数据，如果是摄氏温度的空调，必须是空调在“制冷模式，25°C”时的数据，而如果是华氏温度的空调，则必须是在“制冷模式，78°F”时的数据，如果输入的数据不是这个指定状态下的数据，将造成初始化失败。建议先将空调遥控器设置到制冷模式和指定温度，然后按“风力调节”按键采集信号。

## 4. 空调设置函数

```
const bc7215DataVarPkt_t* setTo(int temp, int mode = -1, int fan = -1, int key = 0);
```

初始化成功后，使用此函数设置空调的状态。

函数具有 4 个输入参数，前三个分别为温度、工作模式，以及风力，第 4 个参数为按键，部分空调的红外编码中，带有按键的信息，比如同样 25°C，从 26°C 按“温度-”按键和从 24°C 按“温度+”按键，虽然最终空调状态一样，但遥控器发射的编码是不同的。这种情况出现在个别品种的空调上，如果不确定是否需要，一般可设置为“保持不变”或者“温度+”。

除温度外，其他参数有默认值，即可以按顺序省略，如 setTo(26), setTo(26, 1), setTo(26, 1, 2) 这样的方式，分别代表设置到 26 度其他不变；设置到 26 度制冷模式，其它不变，设置到 26 度制冷模式风力中。

所有的输入参数，均为 **8 位有符号整数**，每个参数均有各自的取值范围，超出取值范围的输入值，均视为该项设置保持不变。

呼叫此函数后，将直接控制 BC7215A 芯片发送相应的红外信号，同时函数的返回值为指向所发送的红外数据的指针。

### **温度 temp:**

为摄氏温度模式时，温度 temp 范围为 16-30；设置为华氏模式时，温度 temp 范围为 60-88. 超出这个范围的参数值，视为现有设置“保持不变”

### **模式 mode:**

取值范围 0-4, 分别对应为：

0 - 自动(Auto)

1 - 制冷(Cool)

2 - 制热(Heat)

3 - 除湿(Dry)

4 - 通风(Fan)

超出这个范围的参数值，视为现有设置“保持不变”。

### **风力fan:**

取值范围 0-3, 分别对应为:

0 - 自动(Auto)

1 - 低(Low)

2 - 中(Med)

3 - 高(High)

超出这个范围的参数值，视为现有设置“保持不变”。

### **按键key:**

取值范围 0-3, 分别对应为:

0 - 温度+ 键

1 - 温度- 键

2 - 模式按键

3 - 风力按键

超出这个范围的参数值，视为现有设置“保持不变”。多数空调此参数无影响，但对某些空调这个参数则比较重要，比如当改变工作模式时，必须将此设为“2-模式按键”指令才会有效。

### **注意:**

1. **BC7215AC 空调遥控库**是基于编码规则的码库，仅负责产生复合编码规则的数据包，但不会检查其设置是否有效。在各种设置的组合中，有很多是实际空调无法达到的，各种机型会有各自的限制，比如，多数机型可能在“通风”模式下设置温度无效，有的机型可能最低温度只能设置到 18°C 等等，使用本码库时应该以所控制机型的实际限制为准，如果设置超出该机型的允许范围，空调机的反应将不可预知。

## **5. 空调开关函数**

```
const bc7215DataVarPkt_t* on(void);
```

```
const bc7215DataVarPkt_t* off(void);
```

这两个函数用来控制空调的开/关。呼叫后，将直接发射相应红外信号，同时，返回的指针指向实际所发射的数据。

多数空调无需专门的开机指令，在关机状态下，向空调发送设置状态的指令，空调就会开机，对于这样的机型，调用 `on()` 函数，将空调设置到最后一次调用 `setTo()` 函数时的状态，或者初始化采样时的状态。有个别型号开机需要专门的指令，这时，`on()` 函数呼叫后，仅意味着空调会开机，开机后的状态并不确定，需要再使用 `setTo()` 将空调设置为用户需要的状态。

## 6. 红外指令解析函数

```
bool parse(int temp, int mode, int fan, int power);
```

解析刚刚采样的红外信号，解析结果写入对应的参数。输出结果的数值范围和 `setTo()` 函数一致，超出范围的数值表示该项解析不成功。同时，函数的返回值，如果为 `false`，也表示解析不成功。

## 6. 寻找下一匹配函数

```
bool matchNext();
```

初始化时，码库会根据采集的数据，自动匹配空调所用的编码协议，不过，因为市面上空调品牌型号众多，会有不同型号的空调编码协议很接近，但又有细微差异的情况，会有初始化后，空调机的实际动作和设置不相符的情况，如设置风速为高，实际却为低风速等。如果发生这种情况，可以尝试调用此函数，查找库是否有其他和被控空调相符的协议。此函数无输入参数，返回也是一个 `bool` 值，意义和初始化函数中一致。

如果此函数返回为 `true`，则表示空调遥控库已经按照新协议初始化成功，用户可以尝试控制空调看是否能够正常操作，如果仍不正常，还可以继续再次调用，直至返回为 `false`，表示已经没有更多能够匹配的编码协议。

此函数仅可以在初始化正确后调用，调用后，如果想恢复使用初始化后的编码协议，必须重新调用初始化函数。

## 8. 预定义协议

有极少数的几种空调遥控协议，无法由 BC7215A 芯片直接解码，用户须获取红外波形数据后通过使用在线的转换工具完成转换。BC7215A 空调遥控库随库提供已经经过转换的预置数据，供用户直接调取。当发现 BC7215A 无法完成数据采集，或者采集到的信号无法正常初始化时，可以尝试使用预定义数据看是否可以工作。

有关预定义数据相关的函数主要有：

### 获取预定义协议数量

```
uint8_t cntPredef();
```

返回值为库中内置的预定义协议数据的数量。

### 获取预定义协议的名称

```
const char* getPredefName(uint8_t index);
```

输入参数为预定义协议的序号，而返回为一个字符串，为该预置协议的助记名称，帮助用户区分。



## 使用预定义协议初始化

```
bool initPredef(uint8_t index);
```

使用空调遥控库的内置协议数据初始化。输入参数为预定义协议的序号。

## 9. 获取版本信息函数

```
const char* getLibVer();
```

该函数无需输入参数，返回值为一个字符串，包含当前空调遥控库的版本信息。

## 10. 获取基础数据包及获取基础格式包

```
const bc7215DataVarPkt_t* getDataPkt();
```

```
const bc7215FormatPkt_t* getFormatPkt();
```

当初始化成功后，希望将初始化数据保存起来，以便下次开机可直接用于初始化时，可以通过这两个命令来获取当前所使用的初始化数据。

## 11. 设置温度制式

```
void setFahrenheit();
```

```
void setCelsius();
```

针对使用摄氏温度和使用华氏温度的空调，必须在进行配对之前设置空调库的温度制式，否则可能会造成配对失败以及温度控制混乱。

如果未进行设置，默认的工作制式为摄氏。

## 12. 查询当前温度制式

```
bool isCelsius();
```

查询当前的工作制式是否为摄氏模式，返回 true 表示当前为摄氏模式，否则为华氏模式。

## 13. 查询 BC7215A 芯片状态

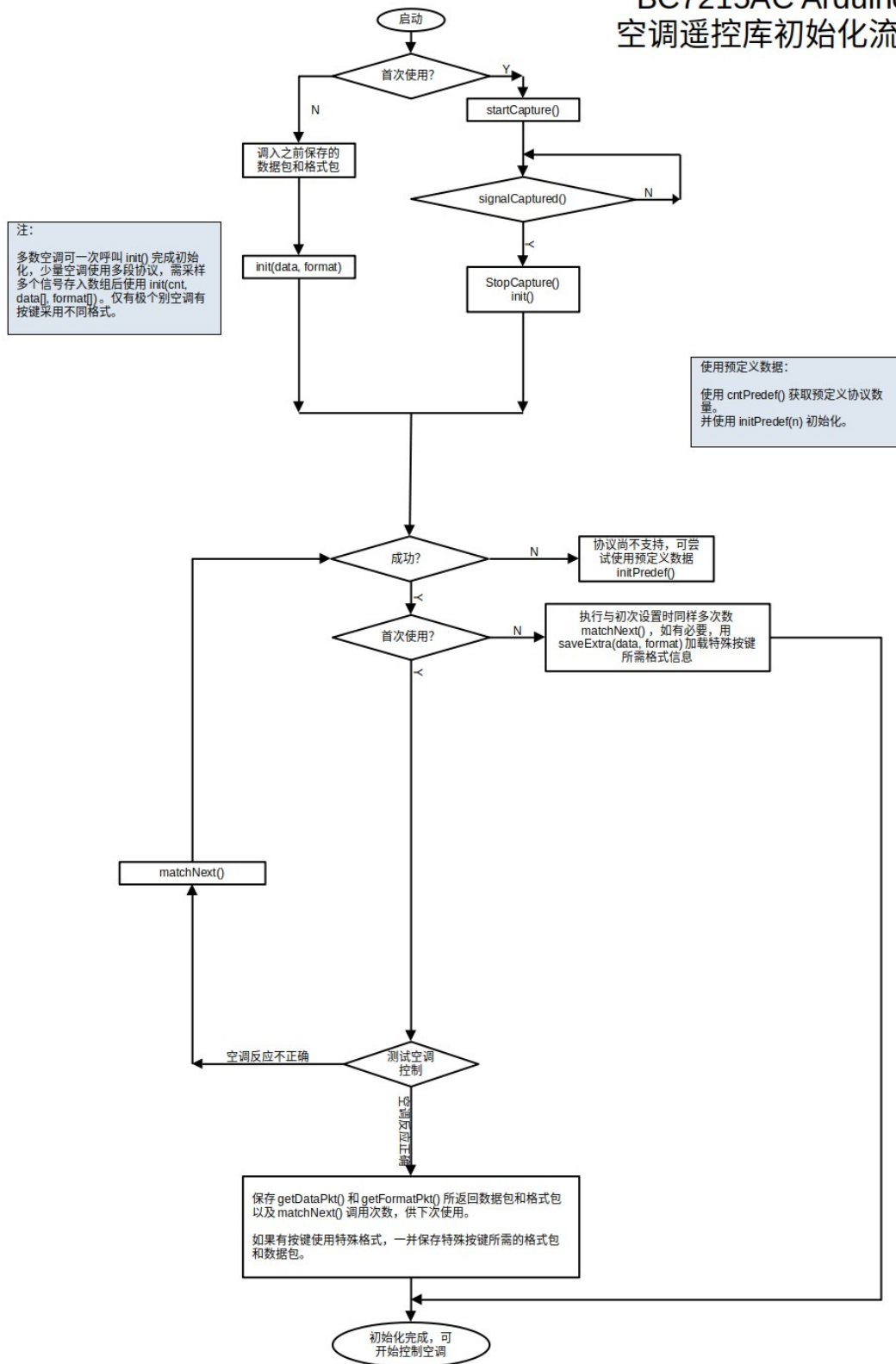
请注意这是个一个 BC7215 驱动库的函数，并非空调控制库函数。

```
bool isBusy();
```

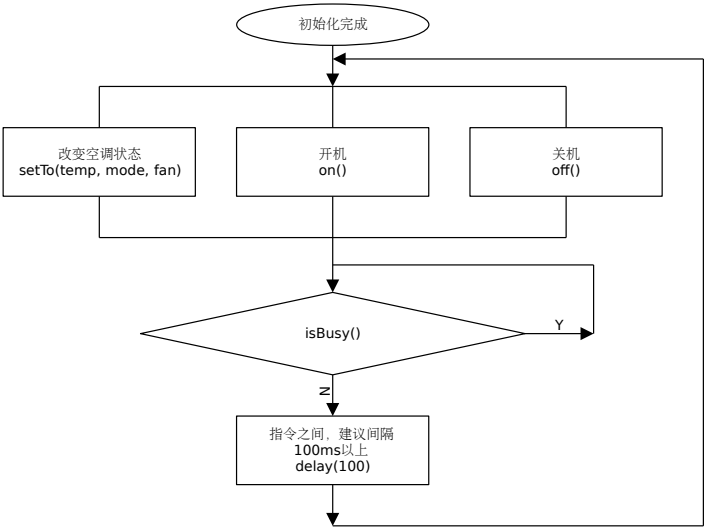
如果 BC7215A 芯片处于接收或者发送红外信号过程中，则返回为 true. 当发射红外信号时，红外信号发送的速度通常比较慢，setTo()和 on(), off()函数返回时，红外信号仍处在发射的过程中，如果需要发送下一个红外信号，必须等待当前的信号完全发送结束，且间隔一段时间(通常需>100ms)后才可以发送下一个信号，否则会造成接收端(空调机)不能正确接收和识别指令。

## 编程流程

### BC7215AC Arduino 空调遥控库初始化流程



# 空调基本控制流程



# 问题提交

如果您遇到了尚不支持的空调机型，或者发现有能正常配对，但控制空调机却有问题，欢迎您通过提交问题上报表格来反馈，问题确认后，通常一周内，您即可获得版本升级解决。问题上报表格文件为 AC\_Lib\_Report-空调码库问题上报.pdf, 在本文档同目录中可找到。

## BC7215A 离线空调码库问题报告单

### BC7215A Offline A/C Library Issue Report

如果您在使用 BC7215A 离线空调码库的过程中遇到了码库初始化失败或能初始化但控制空调失败的情况，请填写此表格并扫描右包的表格和遇到问题的空调遥控器照片，以及空调机照片，一并作为附件邮件发送附件到: ac-lib-issue@bitcode.com.cn，您将会收到确认邮件，问题经技术人员确认后，您将收到报告单号，以及问题解决后的解决方案。

If you experience library initialization failure or control failure (after successful initialization) when using the BC7215A offline AC control library, please complete this form. Email the completed form as an attachment to ac-lib-issue@bitcode.com.cn, including photos of the AC remote control and the AC unit. You will receive an initial confirmation email. After our technical team validates the issue, we will provide you with a ticket number and subsequent solution.

带\*项目为必填  
Fields marked with an asterisk (\*) are required.

1. 用户信息	称呼 Title		姓名 Name	
User Information	公司名称 Company name			
	*联系电话 Contact email			
2. 空调信息	*品牌 Brand			
A/C Information	型号 Model			
	*生产年份 Year of manufacture			
3. 使用信息	*使用码库版本 AC Lib Version			

Usage Information	*是否使用了官方演示板和演示软件测试 Tested with official demo board and demo software	<input type="radio"/> Yes	<input type="radio"/> No
4. 问题详情 Issue Details	*是否已尝试了所有的预定义数据 Have already tried all the pre-defined data	<input type="radio"/> Yes	<input type="radio"/> No
	*是否能正确解码红外信号 IR signal can be decoded	<input type="radio"/> Yes	<input type="radio"/> No 转至第 6 项 Go to section 6
	*是否能正确初始化 Initialization returns OK	<input type="radio"/> Yes	<input type="radio"/> No