BC7215A

离线

空调红外遥控码库

V5.x

- ✔ 免费使用
- ✔ 100%离线
- ✔ C 源码方式提供
- ✔ 体积小巧
- ✔ 系统要求低

目录

简介.....3
空调遥控协议简介.....3
BC7215A 空调码库使用方法.....4
第一步：采样原空调遥控器信号.....4
第二步：初始化码库.....4
第三步：设置需要的温度、模式及风力大小.....4
其它：电源开/关.....4
最佳实践.....4
系统要求.....4
码库详解.....5
组成.....5
数据类型.....5
接口函数.....6
1. 初始化函数.....6
2. 初始化函数 2（V4.0 版新增）.....6
3.空调设置函数.....7
温度 temp:.....7
模式 mode:.....7
风力 fan:.....7
按键 key:.....7
4. 空调开关函数.....8
5. 指令解析函数(V5.0 版新增，试验版).....9
6. 寻找下一匹配函数.....9
7. 预定义数据包.....10
获取预定义数据数量.....10
获取预定义数据的格式信息包.....10
获取预定义数据的原始数据包.....10
8. 获取版本信息函数.....10
9. 检查协议是否需要特殊格式(V3.0 版新增).....10
10. 保存特殊格式信息(V3.0 版新增).....11
11. 替换基础数据包(V3.0 版新增).....11
12. 获取基础数据包(V3.0 版新增).....11
13. 获取基础格式包(V3.0 版新增).....11

简介

BC7215A 离线空调红外遥控码库（以下简称 BC7215A 空调码库），是设计专门用于 BC7215A 芯片的空调红外遥控码库，配合 BC7215A 芯片使用，即可实现离线万能空调遥控的目的。目前 V4.0 已经覆盖全部绝大多数空调品牌型号，且还在快速迭代更新中。

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，从而实现不依赖数据库的万能空调控制。市场上空调的品牌繁多，累计不同品牌和型号，有数千种之多，但所使用的遥控协议则仅有百余种，不同品牌产品，甚至不同大品牌产品间使用相同红外遥控协议的情况非常普遍。因此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。

本空调遥控码库无需联网不依赖任何外部资源，以 C 源码的形式提供（经混淆处理）,对处理器速度要求低，可用于从 8 位单片机到 64 位 Linux 和 Windows 操作系统的各类环境。驱动库体积小巧，V4.0 版本在 ARM Cortex-M3 处理器上编译后体积为 40K 左右，所需 RAM 约 900 字节（具体因处理器而异）。

本空调遥控码库充分利用了 BC7215A 的万能解码功能，实现了基于空调遥控编码规则的通用驱动，而不是简单的波形数据库，这也是其能做到完全离线而体积小巧的原因。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

结合从 V3.0 版起增加的“基础数据替换”功能，在需要时，用户可以实现空调任意设置的控制，比如设置风向、设置经济模式等等。

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。

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，就需要借助码库来实现。

BC7215A 空调码库使用方法

本驱动库的使用非常简单，仅需 3 个步骤，即可完成任意空调的控制。

第一步：采样原空调遥控器信号

利用 BC7215A 的解码功能，接收解码被控空调遥控器的特定信号（制冷模式，25°C)，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。

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，信号采集时通常只会采集第一个红外信号，如果实际起效的遥控信号不是位于第一个，则 BC7215A 会采集到错误的信号。**

第二步：初始化码库

使用解码出的数据作为参数，调用本驱动库的初始化函数，使驱动库识别所控制空调的编码格式 。

第三步：设置需要的温度、模式及风力大小

使用驱动库的设置函数，即可设置为需要的参数，函数会返回一个新的数据包，将数据包通过 BC7215A 发射出去，即可控制空调。

其它：电源开/关

驱动库提供了空调开关函数，用以控制空调的开关机，详情请见后文。

最佳实践

实际应用中，可以将所采样数据保存起来，每次启动后直接用所保存数据完成初始化，这样用户不必每次执行采样操作，做到开机即用。

系统要求

本 BC7215A 空调码库以 C 源码方式提供，要求编译器 C 语言版本为 C99 或以上，理论上可用于任何可使用 C 语言的系统中，不过系统应有足够多的程序和 RAM 空间供本驱动库使用。

码库详解

组成

BC7215A 空调码库由两个文件组成,

`bc7215_ac_lib.h`

`bc7215_ac_lib.c`

其中 bc7215_ac_lib.c 为经混淆后的 C 语言源码。因为驱动库是必须和 BC7215A 共同使用的，会使用到 BC7215(A)驱动库中的数据结构，因此，系统中还应该包含

`bc7215_lib.h`

以及其配置文件

`bc7215_lib_config.h`

因为有的空调遥控码长度比较长，建议 BC7215 驱动库中 `BC7215_MAX_RX_DATA_SIZE` 值设置为 48 或以上。（数值越大，消耗 RAM 越多）

数据类型

从 V3.0 起，BC7215A 空调码库增加了一个专用数据格式

```
typedef struct {
    uint16_t    bitLen;      /**< 为和数据包格式兼容的占位符，其值永远为 0 */
    union {
        uint8_t data[1];     /**< 数据包的数据起始位置 */
        struct {
            const bc7215FormatPkt_t*    fmt;         /**< 格式包指针 */
            const bc7215DataVarPkt_t*   datPkt;      /**< 数据包指针 */
        } msg;               /**< 新的数据结构包含两个指针 */
    } body;                  /**< 新数据格式利用原数据包的数据部分存放指向格式和数据包的指针 */
} bc7215CombinedMsg_t;
```

该数据格式主要用于在码库和用户应用程序之间传递红外信号的格式信息，格式信息用于让 BC7215A 芯片产生和原红外信号一致的红外波形。码库本身并不需要也不处理格式信息，在 3.0 之前的版本中，格式信息完全由用户程序处理，V3.0 中增加了格式信息的管理功能，同时支持了较不常见的不同指令使用不同红外格式的空调协议，码库会针对需要特殊格式的指令给出相应的格式数据。

新的数据结构设计为与数据包 `bc7215DataVarPkt_t` 兼容，低版本库函数中传递数据包指针的地方，均可以用来传递新的数据结构，用户只要检查数据包的 bitLen，当发现 bitLen=0 时，（正常数据包 bitLen 不可能为 0）,即可知道实际传递的是 `combinedMsg_t` 结构，将指针进行强制类型转换，即可得到所传递的格式和数据包。

接口函数

用户将本库的文件加入 C 项目中，并在用户源文件中包含 BC7215A 空调码库的头文件后，即可通过调用接口函数实现空调控制。

1. 初始化函数

```
bool bc7215_ac_init(uint8_t status, const bc7215DataVarPkt_t* dataPktCool25C);
```

该函数用于初始化驱动库，输入参数有两个，一个是 BC7215A 采集被控空调遥控信号时，解码输出原始数据包中的状态字节，一个是指向同时输出的原始数据包的指针。

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。

初始化的输入原始数据包，必须是空调在“制冷模式，25°C”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。可以先将空调遥控器设置到制冷 25°C，然后按“风力调节”按键采集信号，以确保工作模式和温度设置保持不变。

V3.0 版起，第二个参数支持接受 `combinedMsg_t` 格式，但必须按 `const bc7215DataVarPkt_t*` 类型传入，驱动库内部会通过 bitLen 的数值来判断实际的类型，当 bitLen=0 时，即认为实际传入的是 `combinedMsg_t` 格式，会将其中的格式信息保存一个副本在驱动库内部，并可由 `bc7215_ac_get_base_fmt()` 函数获取。

2. 初始化函数 2（V4.0 版新增）

```
bool bc7215_ac_init2(uint8_t msgCnt, const bc7215CombinedMsg_t msgs[], uint8_t segGap);
```

此函数自 V4.0 版本起增加，用于处理红外信号被分为多段的遥控协议。BC7215A 芯片依据信号间的间隔来识别红外信号的结束，目前的设置为 36ms，绝大多数空调红外遥控信号由一段连续信号组成，但仍有少数协议会将一个控制指令分为 2 段甚至更多段发送，中间间隔超过 36ms。V4.0 版码库可自动识别这种情况，通过 `bc7215_ac_need_extra_sample()` 函数反馈给用户，提醒用户需采集多段信息以正确完成初始化。输入的数据仍须为空调在“制冷模式，25°C”时的数据。

返回参数为 bool 值，含义与 bc7215_ac_init()函数相同。

输入参数有 3 个，第一个 msgCnt, 为输入的红外信号的段数，取值范围为 1-4; 第二个参数为一个指向 bc7215CombinedMsg_t 类型数组的指针，数组用到的元素包含指向各个红外信号段的数据和格式信息的指针，比如 msgCnt 值为 2, 则数组前 2 个元素应该包含指向 2 段红外信号的数据和格式信息的指针，而数组中后面的元素则可以没有或为任意内容。第三个参数为红外信号段之间的间隔时间，单位为 ms, 不过 BC7215A 最大能产生的间隔时间为 72ms，超过这个值会被限定在 72ms. 如果输入 0, 间隔时间会采用最常见的 60ms。此间隔时间通常不是关键因素，在很大变化范围内，空调都可以正确接收。

因为函数输入参数中没有接收状态信息，因此用户在将所接收数据传入前，应检查所接收原始数据包的状态字，如果其 b6:REV 反相标志置 1, 则用户应将数据反相复原后再传递给此函数。bc7215_ac_init() 函数因为输入参数中已包含 status 字节，会由函数自己完成这部分检查。

如果输入数据段数为 1, 此函数的作用将等效于 bc7125_ac_init()函数。

3.空调设置函数

```
const bc7215DataVarPkt_t* bc7215_ac_set(int8_t temp, int8_t mode, int8_t fan, int8_t
key);
```

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℃，从 26℃ 按“温度-”按键和从 24℃ 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+”。

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。

温度 temp:

取值范围 0-14, 分别对应温度值 16℃ - 30℃，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”。

本函数的返回值，为一个指向 BC7215(A)数据包的指针，该数据包即为按照被控空调格式编码的包含用户所需设置的数据包，使用发射指令（F5 02）将其发给空调机，即可实现空调的控制。有关发射指令详情，请参照 BC7215(A)数据手册及 BC7215(A)驱动库手册。

如果发现返回结果所指向的数据包的 bitLen 为 0, 则说明输出的实际是一个 `combinedMsg_t*`指针，包含格式信息包的指针，说明这个指令（比如设置模式操作）所使用的信号格式与普通信号不同，需要使用专用的数据包。使用 combinedMsg_t 结构中所包含的格式和数据包指针获取相应的数据。

注意：

1. **BC7215A 空调码库是基于编码规则的码库，仅负责产生符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，有的机型可能没有“制热”模式等等，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。**
2. **数据长度超过 16 字节（BC7215(A)内部缓存大小）时，必须使用基于 BUSY 信号的流控制，否则会造成发出的红外信号出错，空调机无法识别。有关流控制，详情请参阅 BC7215(A)数据手册。**
3. **格式信息包包含了红外信号的调制格式，用户必须在 BC7215A 芯片进入发射模式后将其加载到芯片。加载后，只要 BC7215A 芯片不退出发射模式或断电，则格式信息会一直存在于芯片内，无需每次发射前再次加载。**

4. 空调开关函数

```
const bc7215DataVarPkt_t* bc7215_ac_on(void);

const bc7215DataVarPkt_t* bc7215_ac_off(void);
```

这两个函数用来产生用于控制空调开/关的指令。

通常情况下，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 bc7215_ac_on()函数，会返回 NULL。有个别型号可能开机需要专门的指令，这时，函数会返回一个有效的指针，指向开机指令的数据包。

关机指令会返回一个指向关机指令数据包的指针，发送该数据包，空调即可关机。

如果发现返回结果所指向的数据包的 bitLen 为 0, 则说明输出的实际是一个 `combinedMsg_t*`指针，包含格式信息包的指针，意味着这个指令（开机或关机）所使用的红外信号格式与普通信号不同，需要使用专用的数据包。使用 combinedMsg_t 结构中所包含的格式包指针获取该指令专用的格式信息，并在发射前加载到 BC7215A 芯片。需提醒注意的是，在恢复发送其它普通数据的时候，必须再将芯片内的格式信息恢复为普通格式。

5. 指令解析函数(V5.0 版新增，试验版)

```
bool bc7215_ac_parse(int8_t* temp, int8_t* mode, int8_t* fan, int8_t* power);
```

该函数可以用来从红外信号中解析出其中的温度、模式、风速、及电源状态参数。

函数的输入参数为 4 个指向 8 位有符号整数的指针，解析的结果将写入这些指针所指向的变量。输出的数值的含义，与空调设置函数 `bc7215_ac_set()` 函数中相同。输出的 power 值有 3 个有效值：0, 1, 2, 其中 0=off, 1=on, 2=toggle，toggle 值是指，有的遥控协议中，电源控制指令的作用是翻转电源状态，开关机的指令数据是一样的。除此之外的数值，表示该项无法解析。

函数的返回值为一个 bool 值，返回 true 表示正常解析，返回 false 表示解析失败，比如输入的数据格式错误。

只有在空调遥控库正确初始化后，才能使用此函数。调用此函数前，必须将被解析的数据使用 `bc7215_ac_replace_base()` 函数加载，用户必须保证加载的数据是和初始化时的数据是同一协议格式的数据，否则会解析失败。

此函数目前为试验版，具有以下局限：

- **目前，仅支持与基础数据包同格式的数据解析，尚不支持特殊格式的指令和通过 bc7215_ac_init2()函数初始化的多段数据格式。**
- **并非每个参数都确保解析结果与遥控器上一致，对于以下这些情况，解析结果可能不准确：**

温度	**对有些工作模式，比如“除湿”和“送风”，温度设置并无直接意义，此时厂家可能会在遥控器上给指令中温度项设置一个特定的温度，也可能会设置为一个普通温度范围以外的特殊值，取决于厂家，此时温度解析就会失败。**
模式	**除了“制冷”和“制热”外，其它模式可能并无专门的设置值与之对应，而仅是某特定的设置组合，比如，某厂家可能仅是将“温度 16 度”+“制冷”+“风力低”这个组合称为“除湿”模式，这时的解析结果，模式仍然会解析为“制冷”，会和遥控器上显示的不一致。**
风力	**各个厂商有可能会针对不同型号在标准的协议基础上做少量修改，最常见的是对预设的风力值的改动，有时会造成解析出的结果不对或无法解析。**
电源	**电源开关经常会采用特殊的指令格式或内容，有时会造成无法解析，或对明确控制电源开/关的指令仅能得到“翻转”而不能确定电源的开关状态。**

- **执行过 bc7215_ac_replace_base()后，会改变 BC7215A 空调码库的基础数据，之后的 bc7215_ac_set()的输出都会以最后一次加载的数据包为基础。**

6. 寻找下一匹配函数

```
bool bc7215_ac_find_next(void);
```

初始化时，码库会根据采集的数据，自动判断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，但又有所不同的情况，偶尔会有码库初始化后，空调机的实际动作和设置不相符的情况。如果发生这种情况，可以尝试调用此函数，查找码库中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，则表示码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。

7. 预定义数据包

有极少数的几种空调遥控协议，不能由 BC7215A 芯片直接解码，用户可以通过使用在线的转换工具完成转换。BC7215A 空调码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工作时，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：

获取预定义数据数量

```
uint8_t bc7215_ac_predefined_cnt(void);
```

返回值为库中内置的预定义数据的数量。

获取预定义数据的格式信息包

```
const bc7215FormatPkt_t* bc7215_ac_predefined_fmt(uint8_t index);
```

输入参数为预定义数据的序号，取值范围从 0 开始，其值必须小于上面 `bc7215_ac_predefined_cnt()` 函数的返回值。

获取预定义数据的原始数据包

```
const bc7215DataVarPkt_t* bc7215_ac_predefined_data(uint8_t index);
```

输入参数与上面函数一致，返回为指向数据包的指针，用户可以直接用返回的指针初始化 BC7215A 空调码库。

获取预定义数据的名称

```
const char* bc7215_ac_predefined_name(uint8_t index);
```

输入参数为预定义数据的序号，而返回为一个字符串，为该预置数据的助记名称，帮助用户区分。

8. 获取版本信息函数

const char* bc7215_ac_get_ver(void);

该函数无需输入参数，返回值为一个字符串，包含当前码库的版本信息。

9. 检查协议是否需要特殊格式(V3.0 版新增)

初始化成功后，可以调用

```
uint8_t bc7215_ac_need_extra_sample(void);
```

函数，该函数返回一个 8 位整数，指示当前初始化的空调协议是否有指令使用和普通指令不同的特殊格式。返回结果可能是 0, 1, 2, 3，4 中的一个值：

0. 无需特殊格式
1. 温度设置指令需特殊格式
2. 模式设置指令需特殊格式
3. 风力设置指令需特殊格式

4. 该协议使用多段红外信号，需用 bc7215_ac_init2()函数进行初始化

10. 保存特殊格式信息(V3.0 版新增)

如果上面函数得到非 0 的结果，则用户需要再对需要特殊格式的指令再多做一次采样，以便驱动库了解特殊格式的信息。函数输入和初始化函数类似，但只接收 combinedMsg_t 类型的指针，执行后，特殊格式的格式信息和数据包会保存一份样本在驱动库中。函数接口如下：

```
bool bc7215_ac_save_2nd_base(uint8_t status, const combinedMsg_t* message);
```

保存成功，则返回 true, 否则返回 false;

11. 替换基础数据包(V3.0 版新增)

用于替换产生空调数据包的基础数据。

```
bool bc7215_ac_replace_base(uint8_t status, const bc7215DataVarPkt_t* altDataPkt);
```

此函数的主要用途是用来**控制空调的任意功能**。BC7215 空调码库只提供了温度、模式、风力、电源四项基本控制功能，通常空调还会有很多其它功能，比如调节风向，或者节能模式等等，这些功能的控制，通常都是对应在空调指令中的某些数据位，只要使用带有所需控制信息的数据包替换掉初始化时所用的数据包，即可达到控制该功能的目的。

比如，初始化码库时，空调的状态是固定风向，如果我们此时希望将空调调整为自动摆风，则只需使用此函数，用自动摆风状态的数据包替换掉初始化时的基础数据包，就可达到开启摆风的目的，切换两种数据包为基础数据包，就实现了摆风的随意控制。

用来替换的数据包中温度、模式、风力等设置，可以为任意状态。

12. 获取基础数据包(V3.0 版新增)

```
const bc7215DataVarPkt_t* bc7215_ac_get_base_data(void);
```

调用后得到初始化时所使用的基础数据包。

13. 获取基础格式包(V3.0 版新增)

```
const bc7215FormatPkt_t* bc7215_ac_get_base_fmt(void);
```

如何初始化时使用的是包含格式包的格式，调用此函数可得到该初始化时所用的格式包，如果初始化时为使用格式信息，则返回结果为未定义。