

BC7215

红外数据传输及万能遥控编解码芯片

BC7215 是一款可支持几乎所有常见红外遥控格式的专用编/解码（收发）芯片，自动识别红外遥控信号的编码格式，可以支持 99.5% 以上的空调遥控器和 98% 以上的影音及其它设备红外遥控器。该芯片集红外遥控的解码（接收）和编码（发射）于一体，单片即可完成红外遥控的解码和编码发射功能。

可以收发任意长度的数据，可用于数据通讯。

该芯片红外遥控格式信息和被编码原始数据分开处理，解码时直接输出原始数据，而发射时根据用户输入进行实时编码，因此，该芯片与主机间的

数据通讯量很少，便于用于资源有限的系统。同时，因为可直接输入和输出原始数据，可将该芯片用于低速率的红外数据传输。

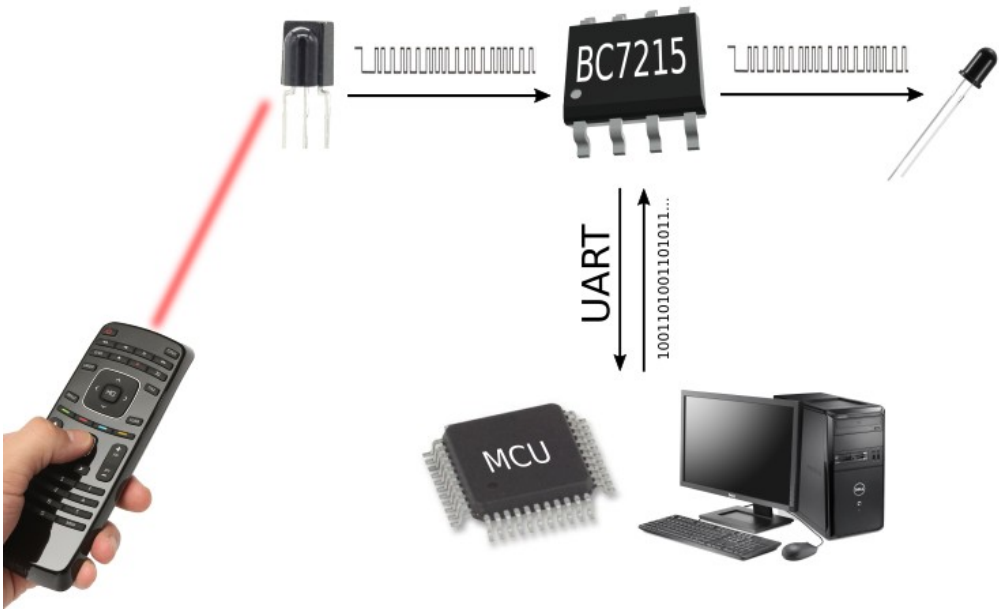
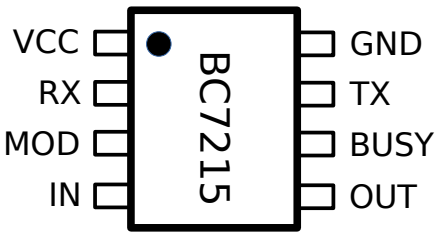
BC7215 的数据接口采用 UART 接口，可以方便地与各类单片机、嵌入式系统、安卓系统 Linux 及 Windows 电脑系统连接而无需底层驱动或 I/O 口级编程。

该芯片可广泛应用于物联网智能家居、万能学习型遥控器、通用红外控制器、数据采集、低速红外数据通讯等各类产品和领域。

特点

- 自动识别红外遥控信号格式
- 集接收解码和编码发送于一体
- 可用于双向低速数据通讯
- 将输出原始数据送回，即可复制遥控信号
- UART 接口
- 内置载波发生器，支持 37.5k 和 56k 载波
- 7uA@3.3V 停机模式

引脚图



目录

特点.....	1
引脚图.....	1
引脚说明.....	4
工作模式.....	4
典型电路.....	5
典型应用电路 1（PC）	5
典型应用电路 2（MCU）	5
UART 接口.....	6
字节填充编码.....	7
红外编码格式.....	7
红外接收模式.....	8
简单模式.....	8
原始数据包格式.....	8
状态特征字节.....	9
原始数据比特数.....	9
数据位对齐方式.....	9
PWM 调制信号.....	9
PPM 调制信号.....	10
复合模式.....	10
格式信息包格式.....	10
接收模式控制字.....	11
输出控制位 FMT.....	11
解码优先级控制位 INV.....	11
发送模式.....	12
格式信息下载指令 F6 01.....	12
红外发送指令 F5 02.....	12
复位指令 7A.....	13
停机指令 F7 00.....	14
非法数据及通讯错误.....	14
软件流程.....	15
UART 发送.....	15
有硬件流控制时.....	15
无硬件流控制时.....	15
UART 接收.....	15
红外发送模式时.....	15
红外接收模式时.....	16

主机数据接收处理流程.....	16
1. 主机无操作系统时.....	16
2. 主机有操作系统时 (含 Arduino).....	19
典型应用场景.....	20
用作红外数据传输.....	20
用作设备的遥控接收.....	21
接受任意遥控器的红外控制.....	21
复制红外遥控信号.....	21
万能学习型遥控器.....	21
红外控制中心.....	21
极限参数.....	22
电气特性.....	22
封装信息.....	22
标识说明.....	22
包装信息.....	22
封装尺寸.....	24

引脚说明

引脚号	引脚名称	功能说明
1	VCC	电源引脚，电压范围 2.7-5.5V
2	RX	UART 数据接收
3	MOD	工作模式选择，MOD 为高时，为接收解码模式，为低时，为编码发射模式
4	IN	红外信号输入引脚，一般接红外接收头的输出
5	OUT	红外驱动信号输出，低电平有效输，根据设置可叠加或不叠加载波。
6	BUSY	工作状态指示，红外解码过程中和发送数据缓冲区已满时，输出高电平
7	TX	UART 数据发送
8	GND	接地

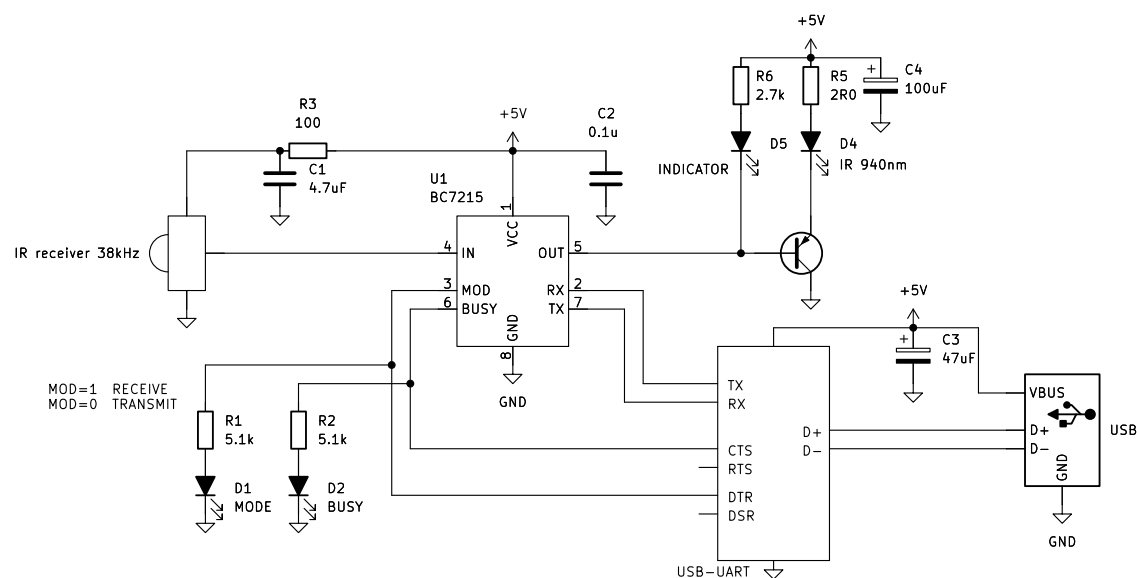
工作模式

BC7215 具有 2 个主要工作模式：接收（解码）模式和发送（编码）模式，通过 MOD 引脚的电平来控制。后文中，统一将两种工作模式简称为接收模式和发送模式。当 MOD 为高电平时，BC7215 工作于接收模式，MOD 为低电平时，工作于发送模式，在发射模式，用户可以通过发送指令让 BC7215 芯片进入停机模式。

用户可以在任何时候切换工作模式，不过请注意，如果在接收或者发送的中途切换工作模式，会造成当前的操作被终止。工作模式的切换可能会需要一定的时间，接收模式向发送模式转换最长可能需要多至 1ms，而发送模式向接收模式转换，最多可能需要 18ms，操作者需要注意。在转换未完成的时候向 BC7215 发送 UART 指令，会丢失数据或得到错误的执行结果，因此在 MOD 电平变化后的转换时间内，不应向 BC7215 发送任何数据。

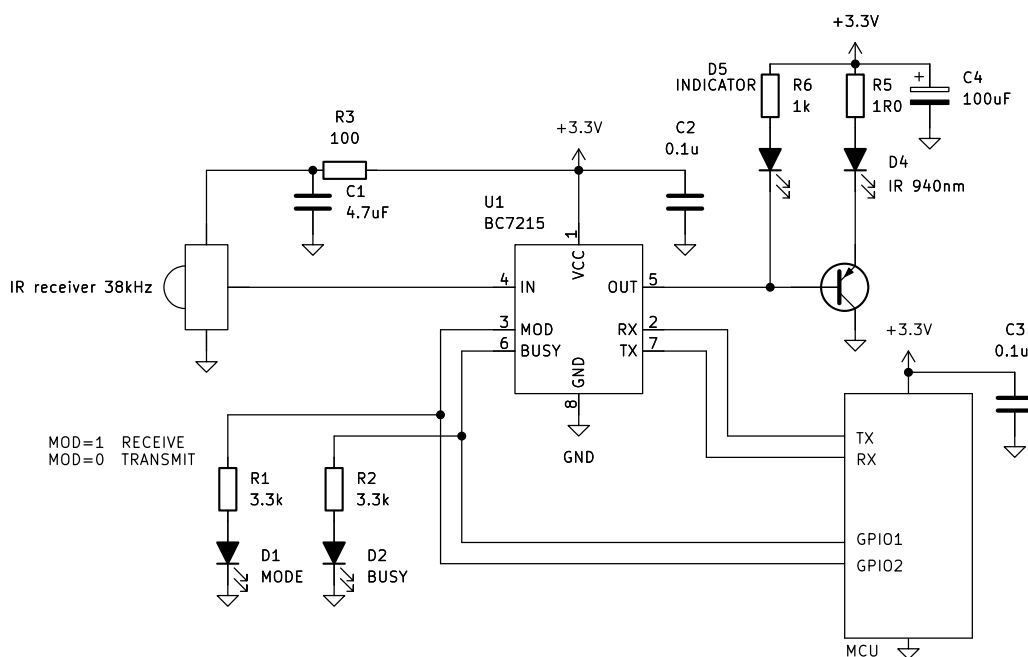
典型电路

典型应用电路 1（PC）



（与电脑连接时，请注意所使用的系统驱动应支持 USB-UART 芯片的硬件流控制，否则可能会造成 BC7215 UART 缓冲区溢出而引起长数据的红外数据发送失败）

典型应用电路 2（MCU）



红外发射管通常要求比较大的峰值电流，才能达到较远的传输距离，很多红外发射管可以承受高达 1A 的瞬间电流，发射时红外发射管工作电流（有载波情况下），可达 200mA 以上，因此在电源设计时需要给予考虑。

UART 接口

BC7215 的 UART 接口，采用固定的通讯参数不可改变，波特率为 19200, 8 个数据位，**2 个停止位**，无奇偶校验。用户在向 BC7215 发送每个字节前，必须检查 BUSY 引脚的状态，在 BUSY 为高电平时发送数据，将造成数据丢失。如果是使用电脑连接 BC7215，可以使用电脑串口的硬件流控制功能，将 BUSY 信号用作 CTS 信号，这样 BC7215 可自动控制数据的发送。

BC7215 在两个情况下会向外发送数据，一个是在接收模式下，接收到了有效的红外输入信号时，BC7215 每接收到 8 个数据位就将其从 UART 输出。**在解码过程中，从输出第一个字节起，BUSY 信号将置高，不允许主机向 BC7215 发送数据。**BUSY 信号将在接收解码结束发送完结束符后，恢复为低电平。BUSY 信号在接收模式时，可以作为主机除了串口数据外通过 I/O 口电平判断是否有红外解码数据的一个标志。

另一个 BC7215 会从 UART 输出数据的情况是在发送模式时，当一个发送指令中所有数据均发送完成时，BC7215 将从 UART 发送一个结束符 0x7A（单一字节）通知上位机；在接收到停机指令后，BC7215 也会返回一个结束符 0x7A 确认。

BC7215 的 UART 数据发送不受外部的控制，完全由芯片自身的工作状态决定。

BC7215 的 UART 接收部分，有一个 16 字节的接收缓存区。当芯片处理数据的速度低于 UART 接收的速度，（比如红外发射时，红外的编码数据发送码率通常会比较低，发送会需要较长的时间，如果待发送的数据帧数据比较长，就有可能出现 UART 缓冲区被写满的情况），这时 BUSY 信号会变为高电平，当随着数据从红外输出口发出，缓存区有空余的空间时，BUSY 会恢复成低电平。用户在发送每一个字节前，须检查 BUSY 的状态，如果为高，则应暂停发送，否则会造成数据丢失。

当在发送模式主机向 BC7215 发送数据或者指令时，应该连续发送完整的指令和数据，各数据字节之间的间隔不应超过 15ms，否则可能会被 BC7215 判断为通讯中断，造成指令失败。如果 BUSY 为高点平时，则不受此 15ms 限制，用户应等待 BUSY 恢复为低电平时再发送下一个字节。

字节填充编码

BC7215 所接收和发出的 UART 数据，均使用字节填充的编码方式进行处理。在 BC7215 的 UART 数据流中，字节 0x7A（ASCII 字符 ‘z’）为特殊字符，表示数据结尾或复位。在接收模式时，该字符仅出现在数据包的结尾处，用户可使用该字符为标志判断数据包的结尾；而在发送模式下 BC7215 在发送结束和进入停机模式前，会发出 0x7A 表示动作结束和确认，而主机向 BC7215 发送 0x7A，将令 BC7215 中止当前操作恢复等待指令状态。在 UART 传输的数据正文中，永远不会出现 0x7A 这个字符。

数据正文中，如果有 0x7A 这个值，必须对其进行替换。处理规则非常简单：如果原始数据为 0x7A，则将其替换为 0x7B 和 0xFA 两个字节；如果原始数据为 0x7B，则将其替换为 0x7B 和 0xFB 两个字节。

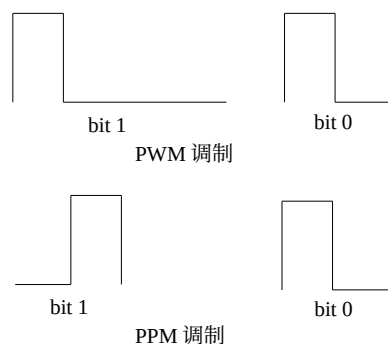
字节填充编码的还原，处理方法也非常简单：如果数据正文出现 0x7B，则将 0x7B 字节丢弃，并将下一个字节的最高位清零。

当仅需要复制遥控信号时，因为 BC7215 输出和输入给 BC7215 的数据都是经过字节填充编码的数据，因此用户可无需进行字节填充编码的还原，将 BC7215 的输出数据直接送回 BC7215 即可。当芯片用作数据传输或需对接收模式下 BC7215 输出的数据进行分析，则用户必须进行字节填充编码的还原，以得到原始数据，向 BC7215 发送的数据也必须实施字节填充编码。

字节填充编码处理可能增加数据的字节数，正文中每出现一次 0x7A，数据就增加一个字节。

红外编码格式

每一种遥控器都可能自定义的编码格式，但整体上遥控编码调制可以分为两类：脉冲宽度 PWM 调制和脉冲相位 PPM 调制(又称双相调制 BPM)。两种调制方式的应用最广泛的代表分别为 NEC 格式和 RC5 格式。



调制的类型决定一个关键因素，即数据发送的顺序。红外数据是串行数据，不同的遥控器格式规定了不同的数据排列顺序，是高位 **MSB** 先传送还是低位 **LSB** 先传送。**BC7215** 对所有采用脉冲宽度 **PWM** 编码的红外信号，均按照低位 **LSB** 在前进行解码，与该格式最常见的遥控芯片 **NEC** 编码所使用的方式相同。而对于 **PPM** 格式，则按照高位 **MSB** 在前进行解码，同该格式最常见的 **RC5** 编码遥控芯片。请参见后文《数据位对齐方式》部分。

不同的编码格式，可能规定不同的调制对应方式，如 **PWM** 模式下较长的脉冲代表 0 或者代表 1，**BC7215** 固定采用较短脉冲代表 0，较长脉冲代表 1 的方式（参见上图）。

很多遥控器的编码格式中，有一些字节或位具有特殊意义，比如有地址码，控制码等，空调遥控数据格式中，有些字节用来表示设定的温度等，**BC7215** 并不区分这些特殊位和字节，如果用户需要理解遥控数据的内在含义，需自行了解所使用的遥控器数据的格式。

BC7215 依靠时间的间隔来区分遥控信号，当 **IN** 引脚上电平保持不变超过 18ms 时，**BC7215** 即认为当前的红外遥控信号已经结束，再收到新的信号会重新开始新的解码过程。

红外接收模式

接收模式下，输出数据包有两种，一种是解码出的原始数据，一种是红外编码的格式信息。两种数据包均以结束符结尾，原始数据包的结束符为单字节 0x7A，格式信息包的结束符为两个连续的 0x7A 字节。芯片在接收模式时，**UART** 输出有两种模式：简单模式和复合模式。简单模式下只输出原始数据和少量额外信息，而复合模式则在输出原始数据包后，紧接着再输出格式信息包。

简单模式

芯片上电默认工作在简单模式，当 **BC7215** 接收到有效的红外信号，每接收到 8 个数据位，就会输出一个字节的原始数据，当红外信号结束，**BC7215** 会再输出 3 个字节的额外信息，形成一个完整的原始数据包，并在最后发送一个结束符 0x7A。

原始数据包格式

原始数据包的长度由所接收的数据决定，因此长度不确定，长度范围可能从 1-512 个字节。因为红外遥控编码格式的速率一般很低，通常每秒仅能传送几十个字节，当数据量大时，过长的传输时间会加

大受到外界干扰的可能性造成可靠性降低，因此通常见到的遥控信号的原始数据长度，一般在几个到几十个字节以内。原始数据包的结尾，BC7215 会紧接着发送一个结束符。原始数据包格式如下：

原始数据包				结束符
原始数据(1-512 字节)	状态特征字(1 字节)	数据比特数低字节(1 字节)	数据比特数高字节(1 字节)	0x7A(1 字节)

状态特征字节

除了结束符外的倒数第三个字节，为红外遥控信号的状态特征字。特征字包含了所接收到的红外编码信号的一些基本特征参数和有关解码结果的一些信息。用户需要关注的，为其最高 4 位：

b7: ERR	b6: REV	b5: TP ₁	b4: TP ₀	b3	b2	b1	b0
---------	---------	---------------------	---------------------	----	----	----	----

b7: ERR 错误标志，当该位为 1 时，表明所收到的信号并未能被正确解码。

b6: REV 反相标志，当该位为 1 时，表明原始数据部分的数据为反相表示，即 0 和 1 反相，输出 0xFF 实际代表原始数据的 0x00。

b5:b4 TP₁:TP₀ 编码类型。TP₁:TP₀ 为 11 时，表明红外信号为脉冲宽度 PWM 调制格式，数据按低位 LSB 在前解码；TP₁:TP₀ 为其它值时，表明信号为相位调制 PPM 方式，数据按高位 MSB 在前解码。

原始数据比特数

该部分占两个字节，为一个 16 位数值，按低字节在前方式排列，该值最大允许值为 0xFFF。请注意比特数并不一定是 8 的整数倍，很多遥控器的数据位数并非 8 的整数倍，比如 RC5 遥控器数据长度为 14 位。BC7215 要求遥控信号的原始数据最少为 8 比特。

数据位对齐方式

PWM 调制信号

如果是 PWM 调制的红外信号（特征字中 TP₁:TP₀ 为 11），则 BC7215 会认为数据为低位在前，即把先收到的比特放在输出字节的低位，比如 uPD6121 芯片输出的遥控信号（NEC 格式），其原始数据为：

C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C _{0'}	C _{1'}	C _{2'}	C _{3'}	C _{4'}	C _{5'}	C _{6'}	C _{7'}	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₀ [¯]	D ₁ [¯]	D ₂ [¯]	D ₃ [¯]	D ₄ [¯]	D ₅ [¯]	D ₆ [¯]	D ₇ [¯]
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

数据从 C₀ 开始发送，C₀-C₇ 为用户码低 8 位，C_{0'}-C_{7'} 为用户码高 8 位，D₀-D₇ 为键码，D₀[¯]-D₇[¯] 为键码的反码。BC7215 输出的原始数据包前 4 个字节将为：

第一个字节								第二个字节								第三个字节								第四个字节							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	C _{7'}	C _{6'}	C _{5'}	C _{4'}	C _{3'}	C _{2'}	C _{1'}	C _{0'}	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	D ₇ [¯]	D ₆ [¯]	D ₅ [¯]	D ₄ [¯]	D ₃ [¯]	D ₂ [¯]	D ₁ [¯]	D ₀ [¯]

总比特数为 32 个，因此最后 3 个字节将为 0x34 0x20 0x00。

如果原始数据的总数据位不是 8 的整数倍，则原始数据的最后一个字节将不是所有位都是有效位。有效位从最后一个字节的最低位用起，向高位排列。比如 TC9148 芯片输出的数据总共 12 位，

C1	C2	C3	H	S1	S2	K1	K2	K3	K4	K5	K6
----	----	----	---	----	----	----	----	----	----	----	----

TC9148 数据从 C1 开始发送，最后 BC7215 输出的原始数据包前 2 个字节为：

第一个字节								第二个字节							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
K2	K1	S2	S1	H	C3	C2	C1	-	-	-	-	K6	K5	K4	K3

总比特数为 12，因此原始数据包最后 3 个字节为 0x3D 0x0C 0x00。

注：'-' 表示无意义数据，可能为任意值，下同。

PPM 调制信号

如果是 PPM 调制信号（特征字中 TP₁:TP₀ 为 11 以外的任何值），则 BC7215 将按照高位在前的方式解码，即先收到的比特位将放在输出字节的高位，如果最后一个字节不足 8 位，有效位将在字节的最左侧。例如 SAA3010 芯片采用 RC5 格式，数据总共 14 位：

1	1	CTL	SYS4	SYS3	SYS2	SYS1	SYS0	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
---	---	-----	------	------	------	------	------	------	------	------	------	------	------

RC5 码固定由 2 个 '1' 开始，然后紧跟控制位(翻转位)、系统码和剩余数据位。BC7215 对其按照高位在前方式解码，因此最后的输出原始数据字节前 2 个字节为：

第一个字节								第二个字节							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
1	1	CTL	SYS4	SYS3	SYS2	SYS1	SYS0	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0	-	-

总比特数为 14，因此原始数据包最后 3 个字节将为 0x05 0x0E 0x00。

复合模式

复合模式即 BC7215 在输出完原始数据包后，紧跟着输出一个格式信息包，里面包含了当前所解码的红外信号的编码格式的全部信息。

格式信息包格式

BC7215 输出的格式信息包的长度是固定的 33 个字节，结束符为两个连续的 0x7A：

格式信息包		结束符
特征字 (1 字节)	编码格式信息 (32 字节)	0x7A 0x7A (2 字节)

特征字与原始数据包中的状态特征字基本相同，但去掉了高两位的解码状态信息。和编码格式信息一起，构成了还原红外遥控信号所需的全部信息。

格式信息在每次接收到新的遥控信号后，都会更新。只要接收到了 4 个以上红外脉冲，BC7215 就会开始对其进行新的解码操作，内存中现有的格式信息就会被覆盖。

因为物理上接收头增益、灵敏度、红外线强度、外界干扰等诸多可变因素，即便是同一只遥控器发出的相同信号，每次输入到 BC7215 芯片的信号都可能会有细微的差别，因此每次产生的格式信息，也

可能会有差别。因此**格式信息包不能作为判断两次遥控信号格式是否相同的依据**，它的作用在于还原所接收到的红外信号格式，并允许用户使用自定义的数据生成红外遥控信号。

需注意的是，BC7215 在输出格式信息包的过程中，BUSY 信号也会置为高电平，此时无法接收 UART 数据，不仅如此，为保护格式信息的完整性，此期间也不会接收红外信号。输出完整的格式信息包大约需要 18ms，如果所接收的红外信号为连续发送，且红外信号数据帧之间的间隔小于 BC7215 设定的红外信号间隔 18ms 加上发送格式信息包所需的 18ms 共 36ms，则会造成后面的红外数据帧前半部分信息丢失，不能正确解码，因此连续接收间隔较短的红外信号时，应该采用简单模式。

接收模式控制字

接收模式下，当 BUSY 不为高电平时，用户可以通过 UART 发送控制字来控制解码的工作模式。控制字只有一个字节，BC7215 接收到控制字后，并不给出任何反馈信号，新的模式控制字将覆盖原有设置，实际起作用的永远是最后一个收到的数据。

控制字的高 6 位为保留位，目前可为任意值，实际有效位为低 2 位：

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	INV	FMT

输出控制位 FMT

FMT 位控制是否工作于复合模式，当该位为 0 时（上电默认状态），芯片工作于简单模式，每次接收到遥控信号，只输出原始数据包；该位为 1 时，工作于复合模式，会在原始数据包后紧跟着输出格式信息包。

输出控制设置后会一直保留，直至收到新的控制字。

解码优先级控制位 INV

该位控制尝试解码方案的优先级。因为 BC7215 是一款通用型的遥控信号解码器，对输入的遥控信号并没有预期的格式，而是从接收到的信号的各种特征来判断其编码的格式，而 BC7215 又是即时将解码出的原始数据输出到 UART 口，因此就会存在一种情况，某两种遥控信号的特征比较相似，只有在接收到足够多的位数后才能正确区分其格式，但如果此时已经将之前按错误格式解码出的数据输出到 UART 口了，则已经无法再更正，此时原始数据包特征字中的“错误”ERR 指示位（bit7）会被置位。BC7215 在默认情况下，总是优先尝试用 PWM 的方式解码，（原始数据包中特征字节的 4,5 位 TP₀,TP₁ 均为 1），如果用户发现输出是 PWM 格式，而且有错误（状态特征字节的第 7 位 ERR 为 1），此时用户就可以尝试发送控制字将 INV 位置 1，这样将强制 BC7215 在下次接收到红外信号时优先使用 PPM 的方式来尝试解码收到的遥控信号，从而将原先无法正确解码的信号正确解码出来。

INV 控制位为单次有效，仅对设置后的下一次接收起作用，过后会自动恢复为 PWM 优先的解码方式。

发送模式

发送模式即红外发送模式，BC7215 按照格式信息所指明的红外编码格式，将从 UART 接收到的数据编码后再发送出去。发送的红外编码信号的格式，完全取决于格式信息，即格式信息包的数据，只需将完整的格式信息包重新载入 BC7215，就可按照相应的格式发送红外信号。如果要复制刚刚接收到的红外信号，可在接收到红外信号后立刻转为发送模式，因为格式信息已经存在于芯片内，可以略过下载格式信息的步骤，直接送入待发送的原始数据，即可按原格式发送红外信号。

红外发射模式时，发送的数据长度可以由用户随意指定，而无需与原红外信号一致，这个特性使得 BC7215 适于用作数据传输，因为可以根据需要收发任意长度的数据。

在发送模式时，BC7215 不接收红外信号，因此不用担心内存中的格式信息被收到的红外信号所覆盖。芯片中的格式信息会一直保留直至收到新的格式信息包或者切换到接收模式。

发送模式时，BC7215 可接受 4 种指令：格式信息下载指令、发送指令、复位指令以及停机指令。格式信息下载、发送指令和停机指令均为 16 位指令，由 2 个字节组成，而复位指令则为单个字符 0x7A。格式信息下载指令和发送指令的内容，必须经过字节填充编码，内容中不可出现字符 0x7A。如果在等待状态下 BC7215 收到除此 4 个指令之外的任何数据，都将被忽略。

格式信息下载指令 F6 01

顺序发送 0xF6 0x01, BC7215 就进入接收格式信息包状态，此后收到的 33 个字节，将被写入到芯片内部的格式信息内存。

格式信息下载指令		格式信息包	
0xF6(1 字节)	0x01(1 字节)	特征字(1 字节)	格式信息(32 字节)

此处下载的格式信息，即上面接收模式复合输出时输出的格式信息包（不包括结尾的结束符 0x7A）。特征字中，最高的两位是用户可更改的。在接收模式输出的格式信息包中，特征字最高的两位始终为 0，在发送模式下载的格式信息包中，此 2 位具有特殊控制作用。下载格式信息指令中的特征字：

b7: NOCA	b6: C56K	b5: TP ₁	b4: TP ₀	b3	b2	b1	b0
----------	----------	---------------------	---------------------	----	----	----	----

其中 b7 为 NOCA，当该位为 1 时，OUT 上输出的红外发射二极管驱动信号将不包括载波信号，仅输出高低电平变化，适合驱动用户自有载波电路或其他用途。

b6 为 C56K，该位为 1 时，输出的载波频率将由 37.5K 变为 56K。

红外发送指令 F5 02

等待状态下，顺序接收到 0xF5 0x02, BC7215 就进入红外发送状态。发送指令的格式如下：

红外发送指令		待发送数据长度		数据
0xF5 (1 字节)	0x02 (1 字节)	发送总比特数低字节 (1 字节)	发送总比特数高字节 (1 字节)	待发送原始数据

在收到待发送原始数据的第一个字节后，BC7215 会立刻开始从 OUT 口按芯片内部的格式信息输出红外信号。BC7215 内部具有一个 16 字节的 UART 缓冲区，因为红外调制的速率一般均较低仅约几十字节每秒，因此开始红外发送后，后面的数据会写入缓冲区等待发送。如果待发送的数据多于 16 个字节（128 比特），可能会造成缓冲区被写满，此时 BUSY 信号会变成高电平，因此用户发送每个字节前必须检查 BUSY 的状态，如果是电脑系统建议将 BUSY 用作硬件流控制信号，避免 UART 缓冲区溢出。

如果接收缓存已经读空，但仍期待更多的数据，则芯片会一直在当前状态等待数据，如果是红外发射的过程中，则红外发射也会被停在当前状态，此时载波输出可能处于关闭，也可能处于工作状态，取决于最后一个可发射数据位的状态，这种中断将造成输出的红外信号时序被打断，从而无法被正确接收。

在执行发送指令时，必须保证前面的发送总比特数与后面的原始数据长度正确对应，并保证所有的数据及时送入芯片。

格式信息包中，并不包括数据的长度信息，发送数据的长度完全取决于发送指令中的长度数据，因此可以做到使用任何一种编码格式，发送任意长度的数据。利用这个特点，用户可以使用任何一种 BC7215 能够正确解码的红外编码格式，发送任意长度的数据，从而用作数据通讯。当用作复制红外遥控信号时，请注意如果数据长度不同，接收设备可能会拒绝接收。

所有数据发送外完成后，BC7215 会输出一个结束符 0x7A。用户可通过这个结束符判断红外数据是否发送完成。BC7215 不会控制各个红外数据帧之间的时间间隔，如果两个发送指令连续发送给 BC7215，则两个红外数据帧会直接连续发出，中间不会有间隔。红外遥控信息一般依靠信号前后的时间间隔来区别数据的开始和结束，如果两个数据帧连续发送，这可能造成两个红外数据帧格式都被破坏，不能被接收方识别。**当需要连续发送多个数据帧时，必须由用户控制数据帧之间的间隔**，发送完第一个发送指令后，等待 BC7215 传回的结束符 0x7A，再等待所需的时间间隔后，发送下一个红外发送指令。建议的红外数据帧之间的时间间隔，为 40ms 以上，具体数值取决于数据格式的规定和接收端的要求。

复制遥控信号时，因为输出的数据已经经过字节填充编码，可直接将接收模式时的输出复制发送给 BC7215。但当发送用户自定义数据时，用户必须注意对数据进行字节填充编码，整个数据包中不可出现字符 0x7A。

红外发送指令可与格式信息下载指令无间隔连续发送，亦可分开发送。

复位指令 7A

发送模式下正常的数据流中，因为必须经字节填充编码处理，因此不可能出现字符 0x7A。在发送模式下，如果 BC7215 接收到了字符 0x7A，会强制芯片中止当前操作，清空 UART 缓冲区，复位工作状态。如果当前正处于红外发送指令执行过程中，BC7215 忽略当前缓冲区内尚未发送的数据，（但会发送完成当前的比特位后，才退出发送状态），因为 BC7215 最长的输出脉冲宽度为 18ms，因此建议发送 0x7A 复位指令至少 36ms 后，再向 BC7215 传送新的数据，以免数据被清除。

停机指令 F7 00

停机指令不带有数据。在发送模式等待指令状态，如果接收到 0xF7 0x00, BC7215 将进入停机模式。在停机模式下，芯片的一切功能都将停止，将不会再接收 UART 指令，只有断电复位或者通过改变 MOD 引脚电平，将芯片转为接收模式，才会令芯片重新进入工作状态。

接收到停机指令后，BC7215 会返回一个结束符 0x7A, 通知主机芯片已经进入停机模式。

非法数据及通讯错误

要使 BC7215 正确地发送红外信号，必须保证其接收到正确的数据，并保证不产生通讯错误。常见的通讯错误产生场景：

1. 在 BUSY 为高电平时发送数据。当内部缓冲区填满时，BUSY 会变为高电平，此时 BC7215 拒绝接收更多数据，此时发送的数据会丢失。因此用户在发送每个字节前，均应检查 BUSY 是否为高电平。

2. BUSY 为低电平时，数据字节间的间隔超过 15ms. 这种情况会被 BC7215 判断为通讯中断，从而产生内部复位。如果用户系统中有更高优先级的中断可能打断向 BC7215 发送指令的过程，并且占用时间超过 15ms，则需特别注意。

除了通讯错误，另一类错误是数据错误，格式信息下载和红外发送指令均带有数据部分，这两个指令的数据部分出错，均会造成输出的红外信号错误。

对于格式信息，必须严格复制 BC7215 输出的格式信息，无论长度还是内容，均必须保持一致。

对于红外发送指令，指令 F5 02 后的两个字节，是待发送的数据的比特长度，这个值必须和后面的数据对应。使用某种红外格式发送数据时，数据的长度可以由用户自由定义，但指令中标明的数据长度必须与实际后面跟随的数据相符。

- 如果标明的数据长度长于后面实际发送的字节数，比如，标明数据长度为 32 个比特，而实际仅发送了三个字节，则在第三个字节发送完后，会因为 BC7215 还在等待第四个字节而接收不到，超过 15ms 后因超时错误而退出红外发送。
- 如果标明的数据长度小于实际发送的字节数，比如，标明数据长度为 12 个比特，即 2 个字节，但实际后面跟随发送了 10 个字节，则多出的 8 个字节会被认为是本指令以外新的数据，BC7215 会试图从中寻找 F5 02, F6 01, 或 F7 00，如果有，则认为是新指令的开始，如果没有，则数据会被抛弃。

软件流程

UART 发送

有硬件流控制时

如果使用的是电脑等主机，因为其串口通常带有硬件流控制功能，UART 的发送部分无需做特别设计，设置使用 CTS/DSR 硬件流控制，将 BC7215 的 BUSY 输出作为 CTS 信号即可。用户只需将待发送数据写入发送缓冲区，其余可由硬件完成。同时，可使用 DTR 信号控制 MOD 电平，这样主机仅需纯粹的 UART 操作，即可 100% 完成对 BC7215 的操作。

无硬件流控制时

如果使用单片机类无硬件流控制的设备控制 BC7215，用户必须在发送每个字节前，检查 BUSY 引脚的状态，如果 BUSY 为高电平，必须等待其变为低电平后，才可以发送。MOD 的信号，也需要额外的 IO 口来控制。

发送的数据必须经过字节填充编码处理，即如果原始数据中存在 0x7A 或 0x7B 字节，必须替换为替代字节，具体规则如下：

0x7A：替换为 0x7B 0xFA 两个字节

0x7B：替换为 0x7B 0xFB 两个字节

需要注意如果要连续执行红外发射操作，两个操作之间需要人为加入至少 40ms 的时间间隔，否则接收端无论是 BC7215 或电视机等硬件设备，均可能无法正确接收。

UART 接收

BC7215 从 UART 输出的数据，分为接收模式和发送模式两种情况。

红外发送模式时

发送模式的 UART 输出很简单，仅在以下两个指令执行完成后，会输出单个字节 0x7A 表示指令执行结束：

- 红外发送指令 F5 02
- 停机指令 F7 00

用户可在指令发出后监视是否收到 0x7A 以判断指令是否已经执行完成，或者忽略这个信息。在连续执行红外发射指令时，用户可以通过这个信息判断上一条指令何时结束，或者直接等待 100ms 以上时间确保上一条发送指令执行完成及 40ms 的红外发送间隔。

红外接收模式时

BC7215 在接收模式时，存在两种情况：在简单模式时，会输出单独的原始数据包；在复合模式时，会连续输出原始数据包和格式信息包。

原始数据包的长度并不固定，由所接收到的信号决定，普通影音类设备的遥控器，数据量一般在 4 个字节以内，空调类遥控器，数据量通常在 32 个字节以内，如果用作数据传输，则理论上最长数据包可达 512 字节，但因为红外信号的发射速度，长的数据包将需要很长时间才能发送完成，增加出错机会，因此通常数据通讯建议使用不大于 16 个字节的数据包。

主机一侧接收 BC7215 发出的数据，因为 BC7215 会在接收到红外信号时立刻主动输出 UART 数据，且内部无缓存，因此主机一侧应该设置一个 UART 缓冲区暂存接收到的数据，且大小至少应该为原始数据包+格式数据包的长度。因为会有接收到超过设计长度的数据包的可能性，主机一侧必须还要考虑缓冲区溢出的情况。

主机数据接收处理流程

处理 UART 接收到的数据，根据主机一侧的不同情况，大致可分为两种类型：

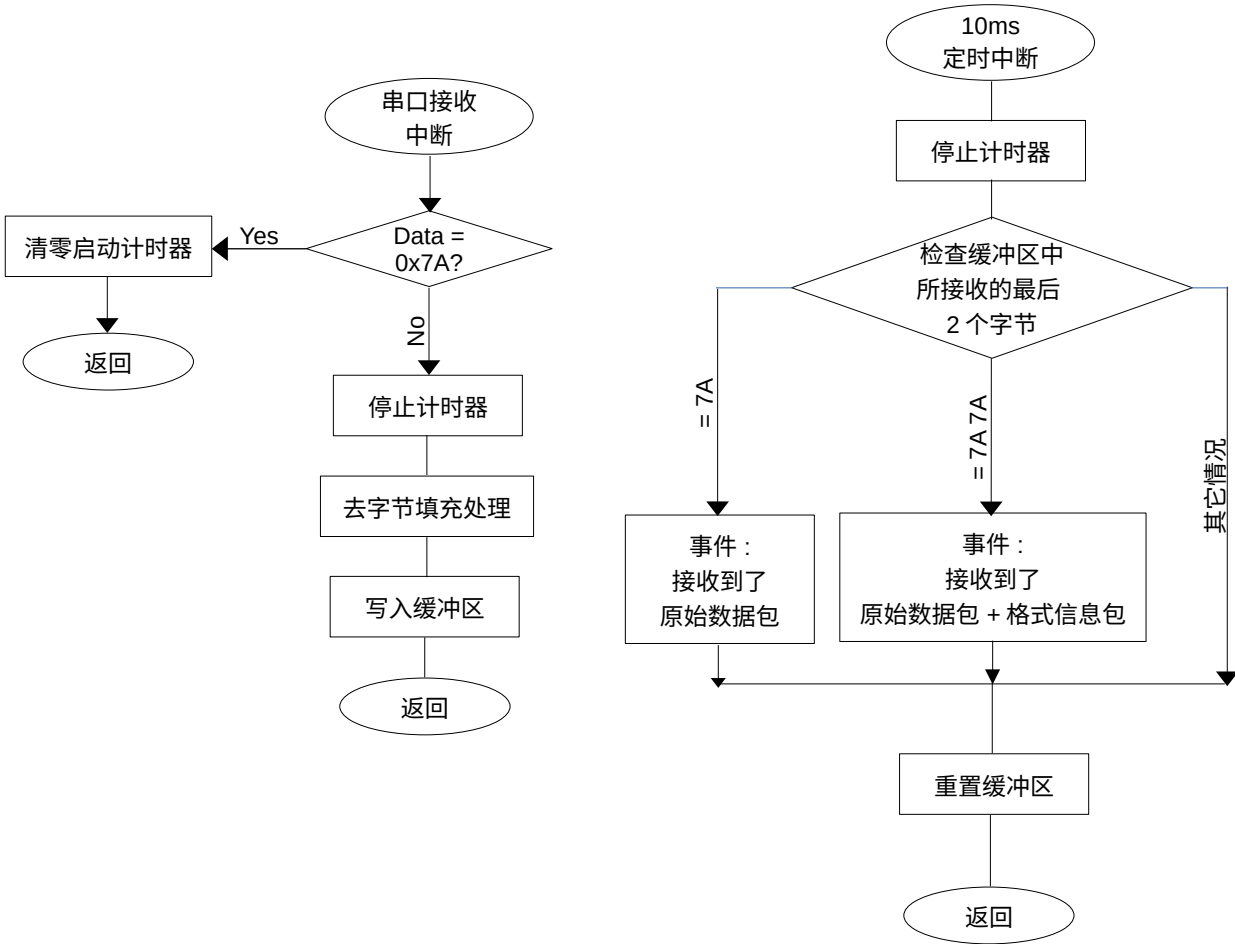
1. 主机无操作系统时

此处的无操作系统，系指用户需要自行完成串口的底层收发操作，用户软件可直接进行硬件 UART 口的读写，也可通过中断的方式获得串口数据接收的通知。

可以有两种方法，如果用户能够提供一个定时器，可以利用格式信息包和原始数据包是连续无间断输出的特点，通过判断 0x7A 字符后的空白时间比较容易的判断数据包是否接收完成；如果用户不能提供定时器，则处理稍复杂。二者均利用了 BC7215 的输出中的结束符 0x7A 作为判别标准。

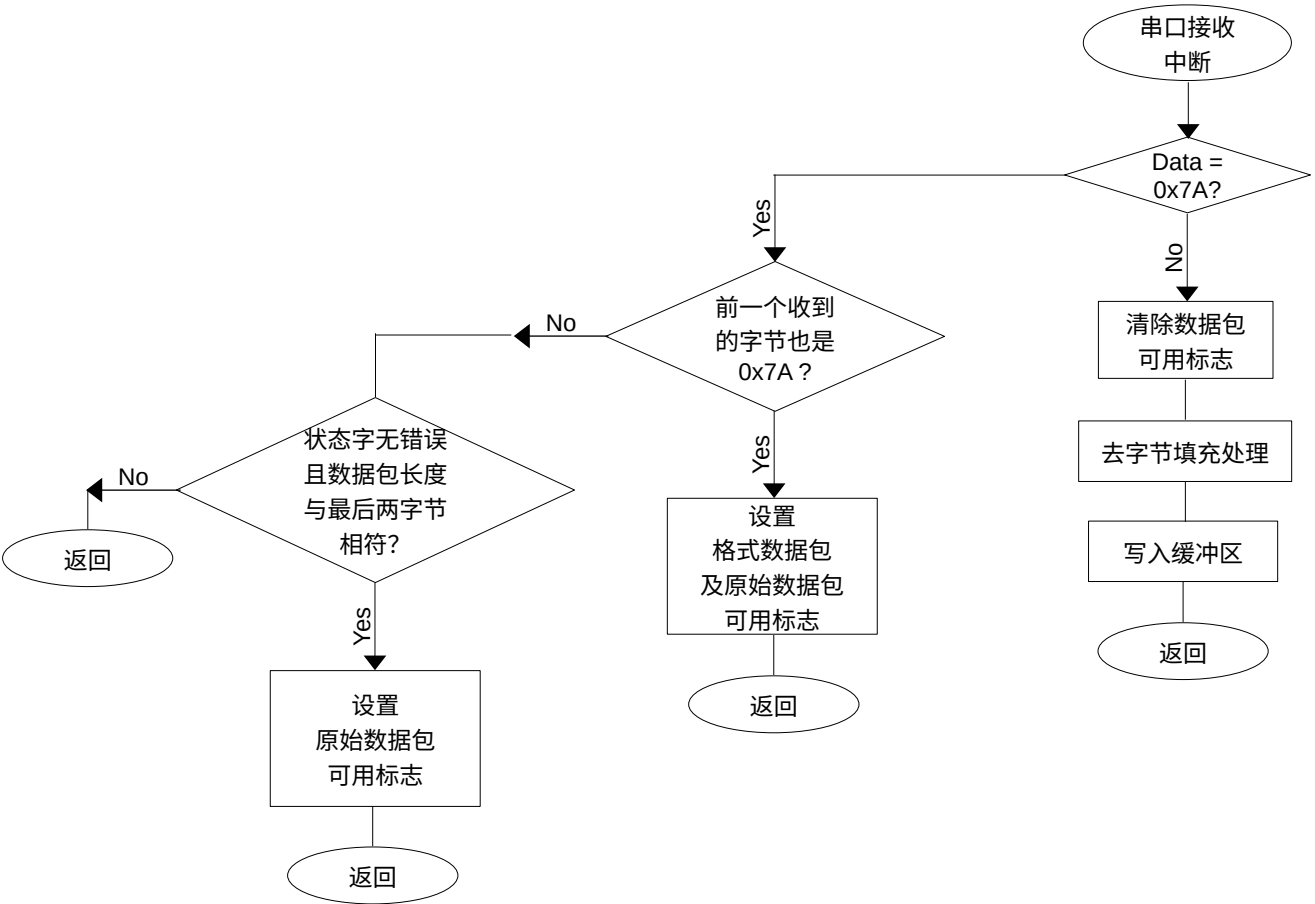
使用定时器的 UART 中断数据处理流程

处理原理：因为复合模式时，原始数据包和格式数据包会连续输出中间没有间隔，而 BC7215 每次解码输出之间，必有一至少约 20ms 的时间间隔(红外信号间隔+8 比特时间)，因此可利用 0x7A 字符后的空白时间长度，判断数据包输出是否结束。



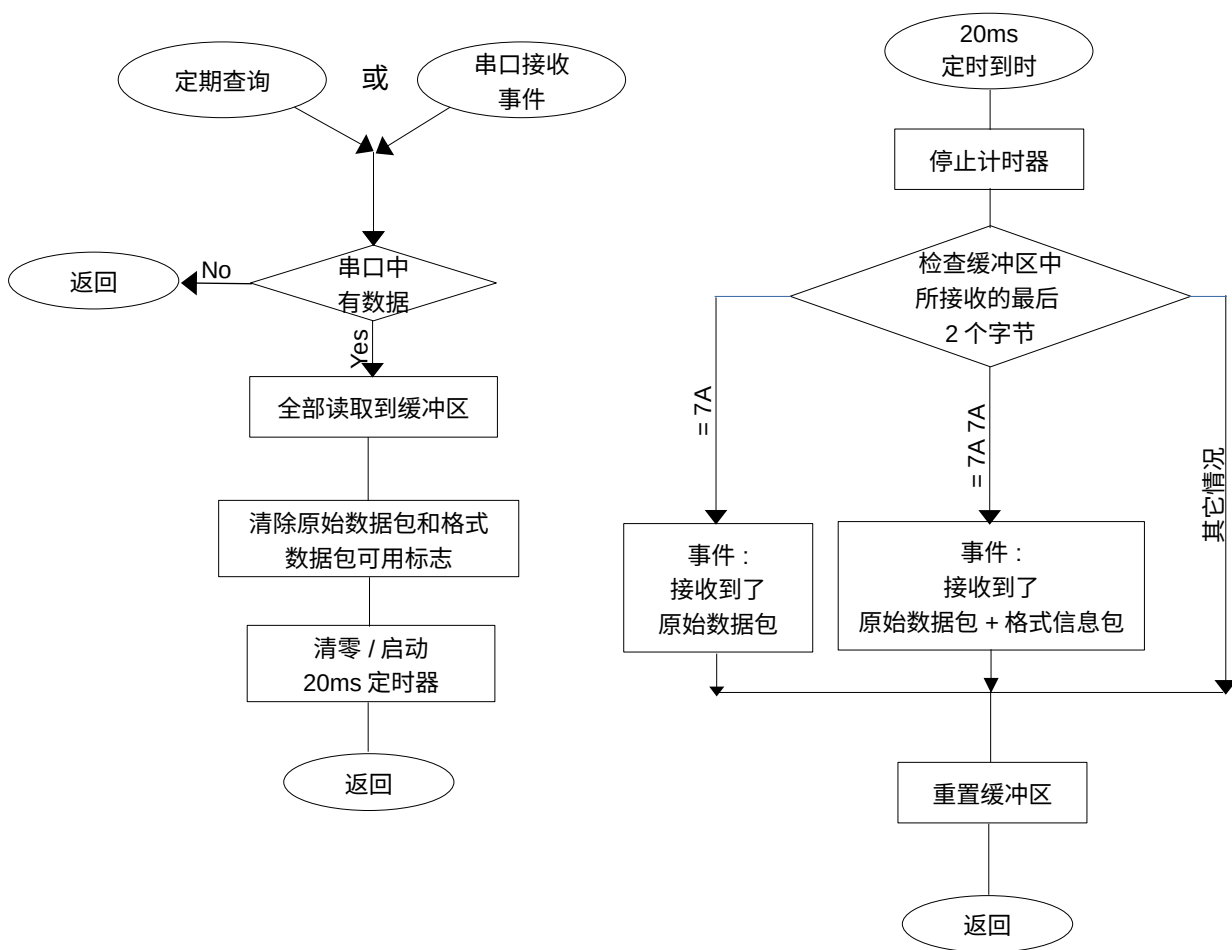
不使用定时器的 UART 中断数据处理流程

处理原理：使用 0x7A 作为数据包结束判断标志，但接收到格式信息包尾部的第一个 0x7A 时，必须通过一些对数据的检查来识别所接收到的是原始数据包的结束符还是格式信息包结束符的第一个字节。



2. 主机有操作系统时(含 Arduino)

此处所指有操作系统，是指系统已经完成了串口底层的操作，用户无需操作硬件，仅需对系统提供的串口的缓冲区进行操作。这种情况下，对用户来说，接收到的数据会在后台自动写入串口的缓冲区，用户仅需定期查询或等待系统提供的串口事件即可。有操作系统的通常会提供系统的定时器，因此最简便的方法是使用定时器检测串口数据的空闲时间，因为输出数据包时，数据字节间的间隔最大不会大于红外数据 8 个比特的时间，只要检测到 UART 接口大于 20ms 的空闲时间，就说明检测到了数据包的结尾。这里与无操作系统的定时器方法的不同在于，无操作系统的处理通过检测结束符 0x7A 来启动定时器，而这里每次探测到串口缓冲区中有新的数据后，均会清零重启定时器。



典型应用场景

用作红外数据传输

红外遥控的编码方式是经过时间验证的可靠的红外数据传输方式，虽然速率较低，但对于无实时性要求，不需传输大量数据的应用场景，比如传感器、遥控数据采集，远程控制等场景，可使用 BC7215，可以构成一个简单、可靠、低成本的解决方案。BC7215 单片即具有收、发功能，因此只需一个芯片，配合红外接收头和红外发射二极管，即可构成一个完整的具有收发功能的红外数据通讯电路。

BC7215 的格式信息包包含有关调制编码格式的全部信息，但不包含数据长度的信息，数据的长度完全由原始数据包中的长度数据决定，因此，**BC7215 可以用任何格式发送和接收任意长度的数据**。这个特点决定了 BC7215 可以方便地用于少量数据的双向数据通讯。不过选择红外编码格式时请注意，某些格式对数据内容有特殊的要求，比如 RC5 格式要求所传送的数据的起始 2 位必须为"1"，这就使其不适宜用作数据传输。通常建议使用特征字为 0x34 的红外格式作为数据传输。

用户可自行使用 BC7215 接收模式采集格式信息。下面这个格式信息包数据将按照 NEC 码的格式进行红外信号调制，载波频率 37.5kHz。标准的 NEC 码为 4 个字节，前两个字节为地址码，后两个字节为键码和键码的反码。但用作数据传输时，数据长度完全可由用户自由定义，而数据内容也完全由用户自行定义，用户只需在红外发射指令中填入需要的数据长度和原始数据即可。一个产生符合 NEC 码编码格式的格式信息包：

```
34 14 5D 0D 5D 14 3D 3D 1D 1C 9C 62 A0 29 B2 99 44 00 00 C2 36 9F
F7 FA B8 E2 9A A3 26 EA 90 87 30
```

请注意因为前文所述的原因，格式信息包并非唯一的，用户自行采集的 NEC 格式信息包可能不同于上面的例子。

因为 BC7215 并不能选择接收特定格式的信号，在用作红外通讯时，接收方有可能接收到并非来自发射方的数据。因此除了通过特征字和数据长度来判断外，用户最好在数据包中加入自己的如地址码等识别特征，并且，应对数据的完整性进行校验，比如使用校验和或者 CRC 校验等方式，以排除其它红外遥控信号的干扰。

另外如果一个数据包占用时间过长，也会使其更容易受到外界的干扰，建议用作数据传输时，一个数据包的长度不要超过 16 字节。如果有较多的数据需要传送，建议分拆成每个不多于 16 个字节的数据包。

请注意 BC7215 在接收模式时，每次收到红外信号，都会将内部的红外格式信息更新，如果通讯时原来处于接收状态的一方转为发送模式，最好重新下载格式信息，以确保始终以相同的数据格式发送。

用作数据通讯时，请注意所有 BC7215 输出的数据均为经过字节填充编码替换的数据，用户须进行还原处理，而输入 BC7215 的数据，也必须经过字节填充编码。

用作设备的遥控接收

因为 BC7215 可以自动识别和解码红外遥控信号，直接输出原始数据，因此可以将其用作用户设备的遥控接收器件。不过因为 BC7215 可以接收任何遥控器的信号，因此在做设备的遥控接收时，用户必须自行判断所接收的数据是否来自指定的遥控器。做到这一点首先用户需要对比输出数据的比特数和特征字，这两个数值相同，可大致认为接收到的遥控数据来自相同的遥控器型号（同样遥控器芯片），另外很多遥控器均有可由用户定制的用户码（地址码）等可供识别，如果用户码也一致，可判定信号来自指定的遥控器。如果需要对特定遥控器按键执行特定的动作，则需要比较全部的原始数据，只有原始数据完全相同，才可认定特定的遥控器按键。

这种应用方式需要用户对所使用的遥控器数据格式有所了解。

接受任意遥控器的红外控制

因为 BC7215 可以接收任何遥控器的信号，可以利用这个功能做成可以接受任何遥控器控制的设备，比如制作可以利用家中其他设备遥控器空闲按键控制的遥控电源开关等。用户程序只需预先记录某个遥控器某按键的原始数据包（原始数据、特征字及数据长度），再收到相同的数据，即可判定收到了这个遥控上特定按键的信号。

复制红外遥控信号

利用 BC7215 的红外发送功能，可以非常容易地复制所接收到的红外遥控信号。对刚刚接收到的红外遥控信号，仅需将 BC7215 及时转换为发送模式（接收到 4 个新红外脉冲之前），因为格式信息已经存在于芯片内，此时无需下载格式信息包，直接使用发送指令，送入解码出的原始数据，即可复制收到的遥控信号。

万能学习型遥控器

只需记录各种不同的遥控器的格式信息，记录各按键的原始数据，或者分析规律后主动生成待发送数据，即可成为一个万能的学习型遥控器，可以将家中各类不同设备集中用 BC7215 控制。

红外控制中心

在物联网应用中，BC7215 可以作为红外控制 HUB 的关键器件，可由远程服务器下载或本地出厂预存，抑或本地学习红外信号格式，完成 1 台设备控制区域内所有的红外遥控设备，或者形成一个红外集中控制网络，网络包含分布不同地点的红外发射节点，各节点从控制中心接收指令，完成对分布在不同地点的红外设备的网络集中控制。

极限参数

参数	符号	范围
储存温度	Ts	-55℃ ~ +125℃
工作温度	Ta	-20℃ ~ +85℃
电源电压	Vcc	-0.3V ~ 5.5V
任意脚对地电压	Vpg	-0.3V ~ Vcc+0.3

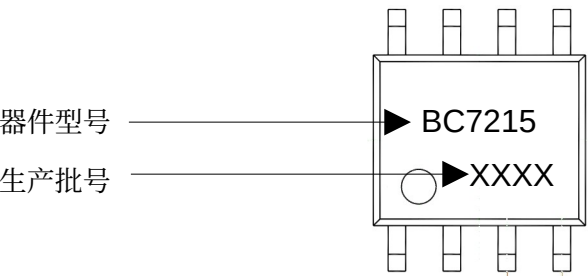
电气特性

注：Ta=25℃

参数	符号	最小值	典型值	最大值	单位	备注
电源电压	Vcc	2.7		5.5	V	
工作电流	Icc		1.7		mA	Vcc=3.3V
			2.5		mA	Vcc=5V
			7		uA	停机模式，Vcc=3.3V
输入低电平	VIL			0.3Vcc	V	
输入高电平	VIH	0.7Vcc			V	
输出引脚吸入电流	IOL			-20	mA	TX, BUSY 引脚
				-60	mA	OUT 引脚, Vcc=3.3V
				-90	mA	OUT 引脚, Vcc=5V
输出引脚输出电流	IOH			20	mA	TX, BUSY 引脚

封装信息

标识说明

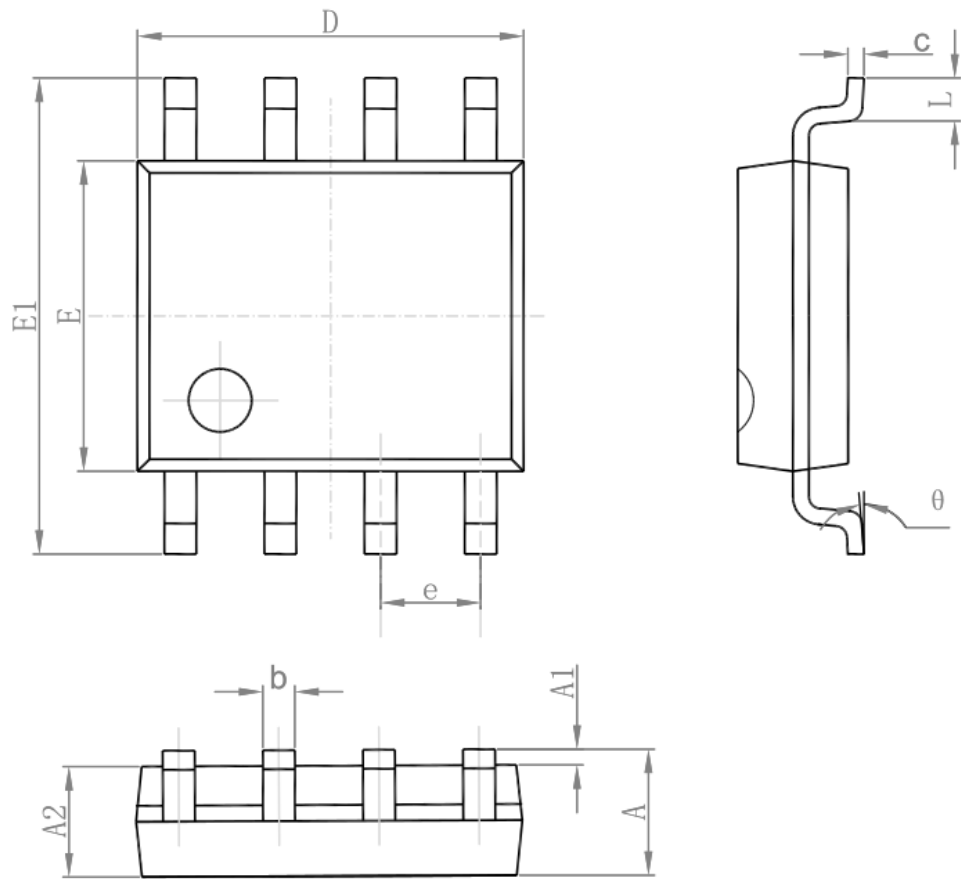


包装信息

订购型号	包装形式	每包装数量
------	------	-------

BC7215-T	管装	10000
BC7215-RS	编带	1800

封装尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°