**红外遥控数字描述方法**

**版本<V0.5>**

文件号：

拟 制：黄金柱

审 核：

批 准：

日 期：2012-3-28

创维数字技术股份有限公司

**文档配置说明**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文件名 | 红外遥控数字描述方法 | | | | | |
| 适 用 | 定义红外遥控码数字化描述方法。 | | | | | |
| 创 建 | 作 者 | 黄金柱 | 时间 | 2012-3-19 | 版本号 | V0.1 |
| 说 明：介绍了NEC、RC5、SONY红外编码方法，提出了利用时间和载波周期数描述红外遥控信号的方法。 | | | | | |
| 修 改 | 作 者 | 黄金柱 | 时间 | 2012-3-21 | 版本号 | V0.2 |
| 说 明：增加了RC6等6种新的红外编码介绍，根据赵总的意见介绍了码表法，并定义了帧格式。 | | | | | |
| 修 改 | 作 者 | 黄金柱 | 时间 | 2012-3-22 | 版本号 | V0.3 |
| 说 明：根据赵总的意见修改码表法为码元符号法，定义了码元符号帧、遥控数据帧以及控制指令帧的格式。并以NEC码为代表，给出了一个利用码元符号法编码红外数据的示例。 | | | | | |
| 修 改 | 作 者 | 黄金柱 | 时间 | 2012-3-24 | 版本号 | V0.4 |
| 说 明：增加了RC-5码的码元符号法编码红外数据的示例。  将1.2节中RC-5码载波频率由36K改为38K（经实测验证）。 | | | | | |
| 修 改 | 作 者 | 黄金柱 | 时间 | 2012-3-28 | 版本号 | V0.5 |
| 说 明： 将码元符号表信息头长度由5字节修正为3字节，控制指令信息头长度长度由5字节修正为2字节，皆减少了扩展数据长度。  修改了第4章中相应的实例。 | | | | | |

# 一、红外遥控协议种类

红外遥控协议种类很多，有NEC、Philip RC-5、Philip RC-6、SONY SIRC、ITT、SHARP、东芝、三菱、三洋（基本都是日本公司的协议）等等，每种协议介绍如下：

## 1.1 NEC协议

**NEC协议的代表发射芯片有：**

[uPD6121](E:\\project\\bluetooth2Ir\\文档\\datasheet\\NEC协议1_uPD6121.pdf)**，uPD6122，PT2222，SC6121，HS6222，HS6221。（部分芯片带有超链接，指向对应的芯片资料。）**

**NEC协议有如下特点：**

* 载波频率：38KHz
* 8位地址码，8位命令码
* 完整发射两次地址码和命令码（第二次发射的是第一次的反码），以提高可靠性。

**NEC协议的调制方式如下：**

NEC协议根据脉冲时间长短解码。每个脉冲为560us长的38KHz载波(约21个载波周期)。逻辑”1”脉冲时间为2.25ms，逻辑”0”脉冲时间为1.12ms。推荐的载波周期为1/4或者1/3。如图 1所示：



图 1 NEC协议中的0和1

**NEC协议的帧格式如下：**

协议规定低位首先发送，如图 2所示的情况，发送的地址码为”59”，命令码为”16”。每次发送的信息首先是用于调整红外接收器增益的9ms AGC(自动增益控制)高电平脉冲，接着是4.5ms的低电平，接下来便是地址码和命令码。地址码和命令码发送两次，第二次发送的是反码 (如：1111 0000的反码为0000 1111)，用于验证接收的信息的准确性。因为每位都发送一次它的反码，所以总体的发送时间是恒定的(即每次发送时，无论是1或0，发送的时间都是它及它反码发送时间总和)。



图 2 NEC脉冲信息链

## 1.2Philip RC-5协议

**RC-5协议的代表发射芯片有：**

[SAA3010](E:\\project\\bluetooth2Ir\\文档\\datasheet\\Philip RC5协议_SAA3010.pdf)，HS3010，SC3010。

**RC-5协议有如下特点：**

* 38KHz载波信号
* 5位地址码和6位命令吗
* 双相编码（亦称曼切斯特编码）

**RC-5的调制方式如下：**

该协议使用38KHz载波双相调制方式（所谓的曼切斯特编码），所有的位时间长度都是1.8ms，每位的一半时间是38KHz的载波，另一半时间是低电平。首先0.9ms时间为38KHz载波，接着0.9ms低电平组成的位表示逻辑“0”；反之表示逻辑“1”。为减少功耗，脉冲占空比推荐为1/3或则1/4。



图 3 RC-5协议中的0和1

**RC-5的帧格式如下：**

RC-5的脉冲信息链如图 4所示：



图 4 RC-5脉冲信息链

这里发送的是$35命令码和$05地址码。首先发送的两个脉冲是启动脉冲，都是逻辑“1”。请注意，位的前面一半时间消逝后，接收器才检测到发送信息的真正起始脉冲。扩展的RC5仅使用1位起始位，发送第6位（下标0） S2以组成7个完整的命令位。第3位是拓扑位，当遥控器按键松开到再次按下时，这位会反转（0→1, 1→0），通过这种方式，接收器可以辨别按键是否一直按下或者是重复按键。接下来是5位器件地址码，最高有效位首先发送。跟着是6位命令码，依然是最高有效位首先发送。一串信息由14位组成，总发送时间为25.2ms。有时总发送时间会短一些，因为第一位起始位S1依然空闲，或者是最后一位信息是逻辑0。

## 1.3 Philip RC-6协议

Philip RC-6协议是RC-5的扩展，它具有非常好的通用性。

**RC-6协议有如下特点：**

* 根据用途，RC-6有多种应用模式；
* 专用于Philip或其OEM设备；
* 根据操作模式，指令码（command code）长度可变；
* 双相（曼彻斯特）编码；
* 36K载波，占空比可从25%~50%（占空比少可以降低功耗）。
* 基本时间单元444us，也就是16个周期载波（1/36k \* 16 = 444µs）。

**RC-6****的调制方式如下：**

如图 5所示，RC-6的起始信号由2.666m（6T）载波和0.889ms（2T）的低电平组成。起始信号又可用来表示调节红外接收器的增益。

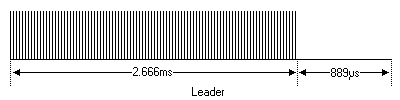


图 5 RC-6起始信号

如图 6所示，RC-6协议与RC-5协议相同，都是采用曼彻斯特码表示0和1。但是，RC-6的表示方法有两点与RC-5不同，第一点是RC-6中先无载波后有载波表示“0”，而RC-5中则是先有载波后无载波表示“0”；第二点是位时间长度不同，RC-5是1.8ms，RC-6则是888us。

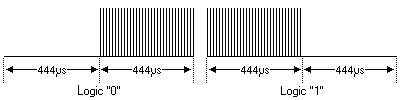


图 6 RC-6协议中的0和1

另外，RC-6还特有一个Trailer位，如图 7所示。它由0.889ms（2T）的载波和0.889ms（2T）的低电平组成。

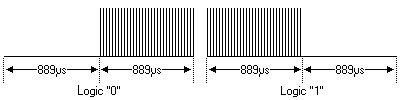


图 7 Trailer位

**RC-6的帧格式如下：**

这里只介绍RC-6的模式0，因为模式0最常用。另外，其它模式的资料也比较难找。

RC-6的帧格式如图 8所示：

RC-6 Mode 0

图 8 RC-6的帧格式

RC-6的帧包括4部分：

帧头：首先传送的是LS（起始信号中2.666ms的载波信号），用于使红外接收器调节增益。然后传输的是SB（起始信号中889us的低电平），用于校正红外接收器的时钟。mb2~mb0是选择模式，由于是模式0，故这三位都是0。帧头的最后是Trailer位，它用于指示按键是否连续接下。

控制码：此部分有8位（a0~a7），用于当作地址码，这就意味着在模式0中，最多可以控制256个不同的设备。

信息码：此部分有8位（c0~c7），用于当作命令字，这意味着每个设备最多可以接受256个不同的命令。

信号释放时间：用于指示哪些帧数据发送完成，时间为2.666ms（6T）。

## 1.4 SONY SIRC协议

**SONY SIRC有如下特点：**

* 40KHz载波；
* 脉冲宽度调制；
* 协议有3种版本，位长分别是12位、15位和20位；
* 每位时间长度为1.8ms（逻辑1）或1.2ms（逻辑0）。

**SONY SIRC的调制方式如下：**

SIRC协议使用脉冲宽度调制。以1.2ms的40K载波脉冲重现逻辑“1”，0.6ms重现逻辑“0”。脉冲间隔均为0.6ms。推荐的载波占空比是1/4或1/3。



图 9 SONY SIRC协议中的0和1

**SONY SIRC的帧格式如下：**

SONY SIRC协议的三种版本包括起始码、命令码、设备码和扩展码（只有20位版本有），如图 10所示，它们的的区别在于有不同长度的设备码以及是否有扩展码。

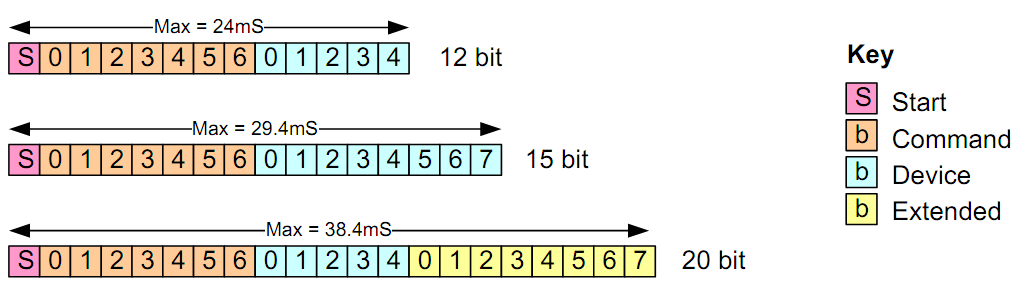


图 10 SONY SIRC的三种模式

图 11展示的是典型的SIRC协议脉冲链。该协议定义最低有效位首先发送。启动脉冲载波总是2.4ms，接着是标准的0.6ms低电平。与需要传送的信息相比，这个启动脉冲常用于接收端对接收信号的自动增益控制。紧接着发送的是7位命令码和5位器件地址码，图 11发送的是命令吗19和地址码1。



图 11 SONY SIRC脉冲信息链

## 1.5 ITT协议

**ITT协议的代表发射芯片有：**

**IRT1250、IRT1260、**[SAA1250](file:///E:\project\bluetooth2Ir\文档\datasheet\ITT协议_SAA1250.pdf)**、SAA1260。**

**ITT协议有如下特点：**

* **没有载波（是不是很有意思）；**
* **脉冲距离编码；**
* **每帧只有14个红外短脉冲（可想而知，很省电吧）；**
* **通信速度快，一个信号发送时间在1.7ms到2.7ms之间；**
* **4位地址码和6位命令码。**

**ITT协议的“0”“1”表示方式如下：**

如图 12所示，ITT协议中脉冲的长度是固定的10us。两脉冲间距为100us表示逻辑0，间距为200us表示逻辑1，间距为300us表示起始脉冲和结束脉冲。

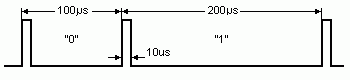


图 12 ITT协议中的0和1

**ITT协议的帧格式如下：**

ITT协议的帧格式如图 13所示：

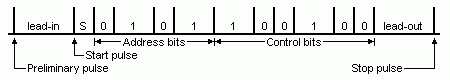


图 13 ITT协议帧格式

Preliminary（我们记作预备脉冲）脉冲信号被接收头用作设置内部放大器的增益参数。当开始脉冲（Start）发送后，紧接着的300us时长的起始条件脉冲（lead-in），被发送的第一位总是100us时长的逻辑 0，而开始位（Start）可以用于校正接收头的时间参数。开始位发送完毕便是信号的高 4 位有效地址位（Adress bits），接着是高6位有效位的命令位和结尾位（Control bits）。最后发送的是另一个300us时长用作结束条件的脉冲（lead-out）。

解码软件简单，很容易就能从接受的信号里检测出有效的信息。结束条件脉冲（lead-out）的时间间隔应该大于开始位（Start）间隔（100us）的 3 倍。每位脉冲时间不能超出逻辑 0 时间的 20%，或者是逻辑1 的40%。 接收部分在接收最后的脉冲信号 360us 后，应软件设置不再等待信号（即进入待机状态）。正如发送被中断或者没有信号发送时，就不需要继续等待了。

预备脉冲信号仅用于 AGC 目的，一些接收解码软件可能忽略这些脉冲信号，这时解码需要从开始脉冲（Start）算起。

## 1.6 SHARP协议

**SHARP协议的代表发射芯片有：**

[LR3715](file:///E:\project\bluetooth2Ir\文档\datasheet\SHARP协议_LR3715.pdf)**（正是由于这款芯片是SHARP出的，所以本协议暂命为SHARP协议。）**

**SHARP协议有如下特点：**

* **38K载波；**
* **5位系统码、5位数据码、4位扩展码；**
* **扩展码中末位表示数据码和扩展码是否反向。**
* 基本时间单元264us，也就是10个周期载波（1/36k \* 10=264µs）。

**SHARP协议的调制方式如下：**

图 14所示为载波中0和1的表述方式：

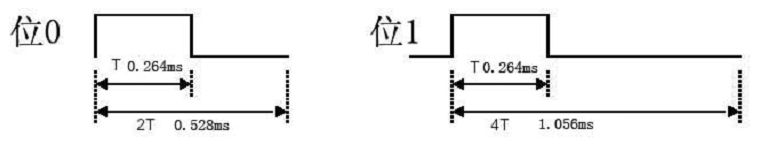


图 14 SHARP协议中的0和1

**SHARP协议的帧格式如下：**

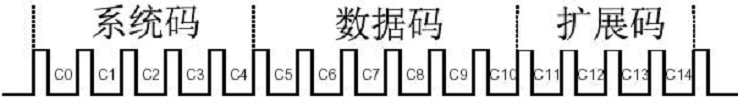
 

图 15 SHARP协议帧格式

## 1.7 东芝协议

**东芝协议的代表发射芯片有：**

**TC9012、**[TC9028](file:///E:\project\bluetooth2Ir\文档\datasheet\东芝协议1_TC9028.pdf)**、TC9243、SC9012、SC9028、SC9243、HS9012。**

**东芝协议有如下特点：**

* **38K载波；**
* **“0”、“1”表示方式与NEC协议相同。**

**东芝协议的调制方式如下：**

与NEC协议相同，在此不浪费篇幅。

**东芝协议的帧格式如下：**

东芝协议帧格式与NEC相似，如图 16所示：

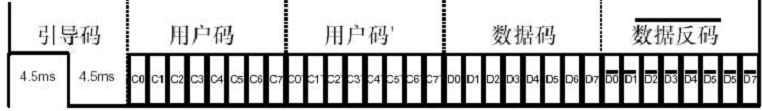


图 16 东芝协议帧格式

数据格式包括了引导码、用户码、数据码和数据反码，编码共占 32 位。数据反码是数据码反相后的编码，编码时可用于对数据的纠错。用户码是可以用二极管在遥控应用电路板上定义，这样可以把同一型号的芯片用在不同的设备中，也称系统码。

与NEC协议不同的地方在于引导码（起始码）时间不同。NEC协议的引导码为9ms载波4.5ms无载波；东莞协议的引导码为4.5 ms载波4.5ms无载波。

## 1.8 三菱协议

**三菱协议的代表发射芯片有：**

**M50560-001、M50560-003、M34280、**[PT2560](file:///E:\project\bluetooth2Ir\文档\datasheet\三菱协议_PT2560-002.pdf)**、SC50560、HS50560。**

**三菱协议的调制方式如下：**

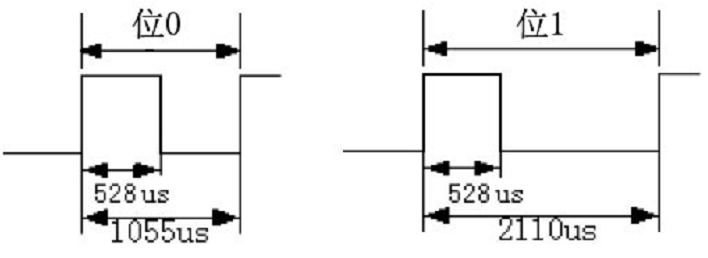


图 17 三菱协议中的0和1

**三菱协议的帧格式如下：**

数据格式为每一帧数据包括 8 位自定义码和 8 位数据码，共 16 位，自定义码和数据码后还有同步位。如图 18所示：

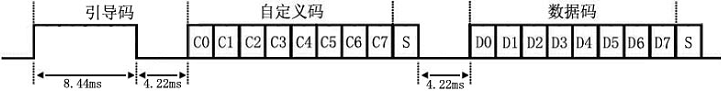


图 18 三菱协议帧格式

## 1.9 三洋协议

**协议的代表发射芯片有：**

[LC7461](file:///E:\project\bluetooth2Ir\文档\datasheet\三洋协议1_LC7461.pdf)**、LC7462。**

**与NEC协议相似，唯一不同之处在于NEC协议用户码为8位，而三洋协议为13位。**

# 二、方案选择

## 2.1 红外遥控协议特点

红外遥控协议大同小异，它们有如下共同的特点：

* 载波（载波频率有所不同，但都在38K左右）；
* 通过高低电平的时间或顺序表示0和1；
* 有起始信号、地址码、命令码。

## 2.2 数字化描述需求

项目需要以一种数字化的方式描述遥控码，编码需满足以下要求：

* 无需获知红外遥控码的含义，只需描述即可；
* 复现时计算量小，减小运算负荷，从而降低硬件要求，降低成本；
* 编码精简，冗余少，降低能耗。

不同协议的遥控码有三种特征：载波、“0”“1”描述方法、信号链。所以，描述红外遥控信号的数字编码需要准确地描述以上特征。

## 2.3载波产生方案选择

**对于载波，有两种产生方法：**

* 硬件发生：使用硬件振荡器产生载波信号，再利用与门将红外遥控信号调制到载波上。
* 软件发生：由MCU定时器产生载波，在单片机内部利用软件的方法将红外遥控信号调至到载波上，然后输出驱动红外发射管。

**由于不同的红外协议载波频率不同，如果用硬件产生载波，不方便。但用软件的方法，只需根据载波频率调整定时器初始时间即可，非常方便，而且节省了硬件成本。故载波由软件发生。**

## 2.4 数字描述方案选择

对于每一种特定的红外遥控协议，载波频率是固定的，“0”和“1”中高电平和低电平的时间也是固定的。由于无需分析遥控码的含义，只需描述，所以，只要描述好红外遥控序列中高电平和低电平的时间即可。

### 2.4.1 用时间描述

例如创维某型号机顶盒遥控器（NEC协议）中，“静音”的序列是80 BF 39 C6，二进制是：10000000 10111111 00111001 11000110

那么它可以描述为：

9000，4500，560，1690，560，560，……，560，1690，560，560

头 1 0 1 0

MCU只需根据数据包中的时间长度，设定时器的初值，定时器时间一到，就翻转输出引脚。

### 2.4.2用载波的周期数描述

以NEC-协议为例。NEC协议中逻辑1有560us高电平，1690us的低电平，逻辑0有560us高电平，560us的低电平。560us约相当于21个周期的载波，1690相当于64周期的载波。所以逻辑1可以用（21，64）来表示，逻辑0则可用（21，21）来表示。

静音可以表述为：

342、171、21，64，21，21，21，21……，21，64，21，64，21，21

头 1 0 0 1 1 0

**用时间和载波的周期数来表述并没有本质的区别，但是用载波周期数表述时软件编写方便，故使用载波周期数来表述。**

通过这种方法，就可以描述每种红外信号的编码。从实现的角度来看，还需要加入传输帧头。完整的传输帧格式如下：

**帧头(0x55)-载波频率-数据长度-扩展位若干-数据-校验（帧尾）**

### 2.4.3 利用码元符号的方法描述

仔细分析第1章所介绍的所有红外协议编码，发现每段红外遥控信号有两个典型特征：

* 是否有载波；
* 持续时间。

以NEC协议为例，仅有以下五种情况：

* 有载波，9ms；
* 无载波，4.5ms；
* 有载波，560us；
* 无载波，560us；
* 无载波，1690us。

对它们进行编码（例如用0001表示9ms有载波，0010表示4.5ms无载波等），就形成了码元符号表。在传输的时候，只需传输码元符号就可以了。

### 2.4.4 各方法对比

利用时间和载波周期数描述的方法是相似的，它们有如下优点：

* 编程方便，对MCU运算能力要求低；
* 可以较精确地描述红外遥控信号。

但是，这种方法的缺点也是明显的：

* 长度太长了，需要存储和发射的数据过多，一是浪费存储空间，二是能耗大。

利用码表的方法可以大幅度地通过编码减小数据量。而且，码表法一个码表示一个高电平（有载波）和低电平（无载波）组合，高低电平与代码在序列中所在的位置无关，故受到干扰时不会导致所有位都错位。

故使用码表法描述红外遥控信号。

# 三、码元符号法说明

在第2.4.3节中，说明了码元符号法的基本原理。本章描述码元符号法实现的具体细节。

## 3.1帧（包）的类型

在产品应用中，需要发送码元符号表、编码后的数据。有时还有一些控制任务，所以还要发送控制数据。所以，需要发送三种类型的帧（包），帧结构如图 19所示：



图 19 三种类型的帧结构

* 码元符号帧：用于传输码元符号。蓝牙每一次配对成功后、没找到数据帧中要求编号的码元符号表时传输。
* 遥控数据帧：用于传输码元符号表编码后的红外数据。每次有遥控任务时传输。
* 控制指令帧：用于手机对终端的控制，如开启、休眠、学习等。有控制任务时传输。

## 3.2 帧头格式

帧头用于描述本帧的长度类型等信息，帧头的长度是定长的；各信息头用于描述特定种类帧的属性，也是定长的；码元符号表、遥控数据、设置指令的长度则是不定长的。

**码元符号法中****帧头的定义如所示：**



图 20 码表法帧头的定义

帧头包含四部分，共4Byte。这三种类型的帧的帧头结构都一样，包含了帧长度和类型的信息。

## 3.3 信息头格式

数据帧的类型不同，其对应的信息头也不同。有三种信息头：码元符号信息头，数据信息头，控制指令信息头。需要注意的是，信息头就

**码元符号表信息头的定义如所示：**



图 21 码元符号表信息头定义

码元符号表信息头定长为5Byte。它包含了以下信息：

**nID：**此次传输的码元符号表的ID号。每个码元符号表都有一个ID号，红外编码数据通过ID号来找到合适的码元符号表解码。nID有8bit，所以可以容纳255个（虽然8bit最多表示256种，但0x00一般不用，故减去1种，为255种）码元符号表。

**fCarr：**载波频率。

**nCode：**码元符号表的长度。因为不同的红外遥控码所需的码元符号表的长度是不一样的，所以需要nCode标示其长度。

**tBasic：**基本的时间单元。码表中记录的时间以基本时间单元为单位。tBasic本身则以载波的周期数来描述。例如NEC码的基本时间单元是560us，那么tBasic则为560/26.3=21。当碰到没有载波的红外遥控信号时（**fCarr=0x00**），tBasic为真实时间，以10us为单位。

**遥控数据信息头****的定义如图 22所示：**



图 22 遥控数据信息头定义

**控制指令信息头的定义如所示：**



图 23 控制指令信息头

cLength用于指示控制指令的长度。其余1Byte保留用于扩展。

## 3.4 码元符号表格式

码元符号表中记录了红外遥控数据编码的方式。每个码元占用2Byte，码元数量视红外遥控码的需要变化。

**码元符号表中****单个码元符号的定义如****图 24所示：**



图 24 单个码元符号的定义

cID是在一个码元符号表中某码元符号的编号。遥控数据就是根据这个码获知其确切的物理含义的。

以NEC红外协议为例构建码元符号表。NEC红外协议有如下5种情况，它们对应的编码为：

* 有载波，9ms: 00010001 00010000 = 0x1110
* 无载波，4.5ms：00100000 00001000 = 0x2008
* 有载波，560us：00110001 00000001 = 0x3101
* 无载波，560us：01000000 00000001 = 0x4001
* 无载波，1690us：01010000 00000011 = 0x5003

## 3.5 遥控数据和控制指令格式

**遥控数据格式定义如所示：**



图 25 遥控数据格式

每个数据长度为4bit。这些数据其实就是码元符号表中的cID：码元符号序号。根据这些序号在码原符号表中查找即可分析出遥控数据的物理含义，进而恢复出**红外遥控波形**。

遥控数据的长度是可变的，因为每种红外遥控码的长度是不同的。若数据无法占满整个字节，补0（因为在码元符号表中，0不代表任何码字，可作为填充码使用）。

**控制指令格式暂不定义。（需协商完成）**

# 四、码元符号法描述举例

## 4.1NEC码描述

同样以创维某型号机顶盒遥控器（NEC协议）为例，“静音”的序列是80 BF 39 C6，二进制是：10000000 10111111 00111001 11000110。计算它的编码。

**码原符号信息帧计算：**

码元符号表信息头：

不妨将此码元符号表的编号nID设为0x01，即二进制00000001。

NEC协议的载波是38K，所以fCarr=0x38 = 00111000。

3.4节已介绍，NEC码需要5种码（判断需要有5种码不需要提前预知本遥控信号为NEC码）元符号描述， 每个码元符号长度为2Byte。故nCode = 0x0A = 00001010。

NEC码的基本时间单元可定义为560us（因为NEC码中的所有时间都是560us的整数倍）。因为tBasic以载波周期数描述，故：

所以码元符号表信息头为：0x01380A1500。

码元符号表3.4节已总结，是：0x1110 2008 3101 4001 5003。

码元符号帧帧头中，帧长度为4+5+10=19=0x13。帧类型为01.所以帧头为：0x47B8 1340。

**所以码原符号信息帧为：**

0x47B8 1340 0138 0A15 00 1110 2008 3101 4001 5003。

再加上校验数据即可。

**遥控数据帧计算：**

遥控数据信息头如下：

所用码元符号表编号为nID=0x01。

遥控数据“静音”的序列是0x80 BF 39 C6共32bit，每1bit需2个码元表示，每一码元4bit。再加上起始信号和结束信号需4个码元，故此序列需34Byte。故nLength=0x22。

10000000 10111111 00111001 11000110

所以遥控数据信息头为：0x012200。

利用3.4节总结的码元符号表，遥控数据为：

0x1235 3434 3434 3434 3435 3435 …… 3435 3534 34

其中最后的34是结束信号。

遥控数据帧帧头中，帧长度为4+3+34=41=0x29。帧类型为10。所以帧头为：0x47B82B80。

**所以遥控数据帧为：**

0x47B8 2980 0122 00 1235 3434 3434 3434 3435 3435 … 3435 3534 34

再加上校验数据即可。

## 4.2RC-5码描述

RC-5码与其它红外遥控编码有一个最大的区别：对于特定的遥控命令，其它红外遥控编码的物理波形是确定的，而RC-5则有两种情况。1.2节中图 4中可看到，RC-5码有一个T位，当遥控器按键松开后再按下时，这位才会翻转（0→1, 1→0），通过这种方式，接收器可以辨别按键是按着不放还是刚刚按下。当按着不放时，遥控器会每隔115ms发送一次图 4所示的遥控序列，在此过程中，T位不会变化。当按键弹起后再按下时，T位翻转。

RC-6码也有Trailer位，跟RC-5的T位功能一样。

为应付这个问题，有两种方案：

* 让手机端程序多做点事儿，由手机端程序判断某次发送时T位该是1还是0；
* 单位机端多做点事儿，手机每次会把这两种情况的编码都通过蓝牙发送过来，由单片机判断将哪组编码转换成红外发送出去。

若将T的判断放在单片机端，那么每次手机都需要将T=0和T=1的数据全部通过蓝牙传输到单片机，数据传输量增倍，不利于降低能耗。而且手机跑得快，资源多，多做点事儿也是应该的。故选择将T的判断放在手机端。

所以，在单片机端，只需将手机传来的编码忠实地翻译成红外信号即可。

**码原符号信息帧计算：**

码元符号表信息头：

不妨将此码元符号表的编号nID设为0x02，即二进制00000010。

RC-5协议的载波是38K，所以fCarr=0x38 = 00111000。

RC-5码的基本时间单元可定义为900us（因为RC-5码中的所有时间都是900us的整数倍）。因为tBasic以载波周期数描述，故：

计算的过程中虽然有近似，但红外遥控码对这种数量级的时间微调并不敏感，即使考虑积累误差。

RC-5码有需要4种码元符号，

* 有载波，900us： 00010001 00000001 = 0x1101
* 无载波，900us： 00100000 00000001 = 0x2001
* 有载波，1800us：00110001 00000010 = 0x3102
* 无载波，1800us：01000000 00000010 = 0x4002

每个码元符号长度为2Byte。故nCode = 0x08 = 00001000。

所以码元符号表信息头为：0x0238082200。

码元符号表是：0x1101 2001 3102 4002。

码元符号帧帧头中，帧长度为4+5+8=17=0x11。帧类型为01.所以帧头为：0x47B8 1140。

**所以码原符号信息帧为：**

0x47B8 1140 0238 0822 00 1101 2001 3102 4002。

再加上校验数据即可。

**遥控数据帧计算：**

遥控数据信息头如下：

所用码元符号表编号为nID=0x02。

中遥控数据利用码元符号表编码后的数据是：

0x2123 2121 4341 2123 4341。

此序列需10Byte。

所以遥控数据信息头为：0x020A00。

遥控数据帧帧头中，帧长度为4+3+10=17=0x11。帧类型为10。所以帧头为：0x47B81380。

**所以遥控数据帧为：**

0x47B8 1180 020A 00 2123 2121 4341 2123 4341。

再加上校验数据即可。