

红外线遥控系统原理及软件解码实例

简介： 红外线遥控是目前使用最广泛的一种通信和遥控手段。由于红外线遥控装置具有体积小、功耗低、功

关键字： 红外

红外线遥控是目前使用最广泛的一种通信和遥控手段。由于红外线遥控装置具有体积小、功耗低、功能强、成本低等特点，因而，继彩电、录像机之后，在录音机、音响设备、空调机以及玩具等其它小型电器装置上也纷纷采用红外线遥控。工业设备中，在高压、辐射、有毒气体、粉尘等环境下，采用红外线遥控不仅完全可靠而且能有效地隔离电气干扰。

1、 红外遥控系统

通用红外遥控系统由发射和接收两大部分组成。应用编解码专用集成电路芯片来进行控制操作，如图 1 所示。发射部分包括键盘矩阵、编码调制、LED 红外发送器；接收部分包括光、电转换放大器、解调、解码电路。

图 1 红外线遥控系统框图

2 、 遥控发射器及其编码

遥控发射器专用芯片很多，根据编码格式可以分成两大类，这里我们以运用比较广泛，解码比较容易的一类来加以说明，现以日本 NEC 的 uPD6121G 组成发射电路为例说明编码原理（一般家庭用的 DVD、VCD、音响都使用这种编码方式）。当发射器按键按下后，即有遥控码发出，所按的键不同遥控编码也不同。这种遥控码具有以下特征：

采用脉宽调制的串行码，以脉宽为 0.565ms、间隔 0.56ms、周期为 1.125 ms 的组合表示二进制的“0”；以脉宽为 0.565ms、间隔 1.685ms、周期为 2.25 ms 的组合表示二进制的“1”，其波形如图 2 所示。

图 2 遥控码的“0”和“1”（注：所有波形为接收端的与发射相反）

上述“0”和“1”组成的 32 位二进制码经 38kHz 的载频进行二次调制以提高发射效率，达到降低电源功耗的目的。然后再通过红外发射二极管产生红外线向空间发射，如图 3 所示。

图 3 遥控信号编码波形图

UPD6121G 产生的遥控编码是连续的 32 位二进制码组，其中前 16 位为用户识别码，能区别不同的电器设备，防止不同机种遥控码互相干扰。该芯片的用

户识别码固定为十六进制 01H；后 16 位为 8 位操作码（功能码）及其反码。U PD6121G 最多额 128 种不同组合的编码。

遥控器在按键按下后，周期性地发出同一种 32 位二进制码，周期约为 108 ms。一组码本身的持续时间随它包含的二进制“0”和“1”的个数不同而不同，大约在 45~63ms 之间，图 4 为发射波形图。

图 4 遥控连发信号波形

当一个键按下超过 36ms，振荡器使芯片激活，将发射一组 108ms 的编码脉冲，这 108ms 发射代码由一个引导码（9ms），一个结果码（4.5ms），低 8 位地址码（9ms~18ms），高 8 位地址码（9ms~18ms），8 位数据码（9ms~18ms）和这 8 位数据的反码（9ms~18ms）组成。如果键按下超过 108ms 仍未松开，接下来发射的代码（连发码）将仅由起始码（9ms）和结束码（2.25ms）组成。

图 5 引导码

图 6 连发码

3、遥控信号接收

接收电路可以使用一种集红外线接收和放大于一体的一体化红外线接收器，不需要任何外接元件，就能完成从红外线接收到输出与 TTL 电平信号兼容的所有工作，而体积和普通的塑封三极管大小一样，它适合于各种红外线遥控和红外线数据传输。

接收器对外只有 3 个引脚：Out、GND、Vcc 与单片机接口非常方便，如图 7 所示。

图 7

- ① 脉冲信号输出接，直接接单片机的 IO 口。
- ② GND 接系统的地线（0V）；
- ③ Vcc 接系统的电源正极（+5V）；

4 遥控信号的解码

下面是一个对 51 实验板配套的红外线遥控器的解码程序，它可以把红外遥控器每一个按键的键值读出来，并且通过实验板上 P1 口的 8 个 LED 显示出来，在解码成功的同时并且能发出“嘀嘀嘀”的提示音。

```
ORG 0000H
```

```
MAIN
```

```
    JNB  P2.2,IR    ;遥控扫描
```

```
    LJMP MAIN      ;在正常无遥控信号时，一体化红外接收头输出是高
```

电平，程序一直在循环。

```

;=====
; 解码程序
IR
;以下对遥控信号的 9000 微秒的初始低电平信号的识别，波形见图 5。
MOV R6,#10
IR_SB
ACALL DELAY882 ;调用 882 微秒延时子程序
JB P2.2,IR_ERROR ;延时 882 微秒后判断 P2.2 脚是否出现高电平如
果有就退出解码程序
DJNZ R6,IR_SB ;重复 10 次，目的是检测在 8820 微秒内如果出现
高电平就退出解码程序

;识别连发码，和跳过 4.5ma 的高电平。
JNB P2.2, $ ;等待高电平避开 9 毫秒低电平引导脉冲
ACALL DELAY2400
JNB P2.2,IR_Rp ;这里为低电平，认为是连发码信号，见图 6。
ACALL DELAY2400 ;延时 4.74 毫秒避开 4.5 毫秒的结果码

;以下 32 数据码的读取，0 和 1 的识别 请看图 2
MOV R1,#1AH ;设定 1AH 为起始 RAM 区
MOV R2,#4
IR_4BYTE
MOV R3,#8
IR_8BIT
JNB P2.2,$ ;等待地址码第一位的高电平信号
LCALL DELAY882 ;高电平开始后用 882 微秒的时间尺去判断信号
此时的高低电平状态
MOV C,P2.2 ;将 P2.2 引脚此时的电平状态 0 或 1 存入 C 中
JNC IR_8BIT_0 ;如果为 0 就跳转到 IR_8BIT_0
LCALL DELAY1000
IR_8BIT_0
MOV A,@R1 ;将 R1 中地址的给 A
RRC A ;将 C 中的值 0 或 1 移入 A 中的最低位
MOV @R1,A ;将 A 中的数暂时存放在 R1 中
DJNZ R3,IR_8BIT ;接收地址码的高 8 位
INC R1 ;对 R1 中的值加 1，换下一个 RAM
DJNZ R2,IR_4BYTE ;接收完 16 位地址码和 8 位数据码和 8 位数据，
;存放在 1AH1BH1CH1DH 的 RAM 中
;解码成功
JMP IR_GOTO

IR_Rp
;这里为重复码执行处
;按住遥控按键时，每过 108ms 就到这里来

```

```
JMP IR_GOTO
```

```
IR_ERROR
```

```
;错语退出
```

```
LJMP MAIN ;退出解码子程序
```

```
=====
```

```
;遥控执行部份
```

```
IR_GOTO
```

器 ;这里还要判断 1AH 和 1BH 两个系统码或用户码，用于识别不同的遥控器

```
;MOV A,1AH
```

```
;CJNE A,#xxH,IR_ERROR ;用户码 1 不对则退出
```

```
;MOV A,1BH
```

```
;CJNE A,#xxH,IR_ERROR ;用户码 2 不对则退出
```

```
;判断两个数据码是否相反
```

```
MOV A,1CH
```

```
CPL A
```

```
CJNE A,1DH,IR_ERROR ;两个数据码不相反则退出
```

```
;遥控执行部份
```

```
;MOV A,1DH ;判断对应按键
```

```
;CJNE A,#xxH,$+6
```

```
;LJMP -à 跳到对应按键执行处
```

```
;CJNE A,#xxH,$+6
```

```
;LJMP -à 跳到对应按键执行处
```

```
;
```

```
MOV P1,1DH ;将按键的键值通过 P1 口的 8 个 LED 显示出来!
```

```
CLR P2.3 ;蜂鸣器鸣响一滴滴滴一的声音，表示解码成功
```

```
LCALL DELAY2400
```

```
LCALL DELAY2400
```

```
LCALL DELAY2400
```

```
SETB P2.3 ;蜂鸣器停止
```

```
;清除遥控值使连接失效
```

```
MOV 1AH,#00H
```

```
MOV 1BH,#00H
```

```
MOV 1CH,#00H
```

```
MOV 1DH,#00H
```

```
LJMP MAIN
```

```
=====
```

