手把手教你学 PIC 单片机 C 语言教程 第 25 课

（红外线解码实验）

参考例程所在位置：HL-K18 配套光盘\ 12 IR LED

1、红外遥控编码

红外遥控编码常用的格式有两种：NEC 和 RC5，由于市场上大多数遥控器（包括我们配套的红外遥控器）都是 NEC 的，所以我们这里只分析 NEC：

NEC 格式的特征：

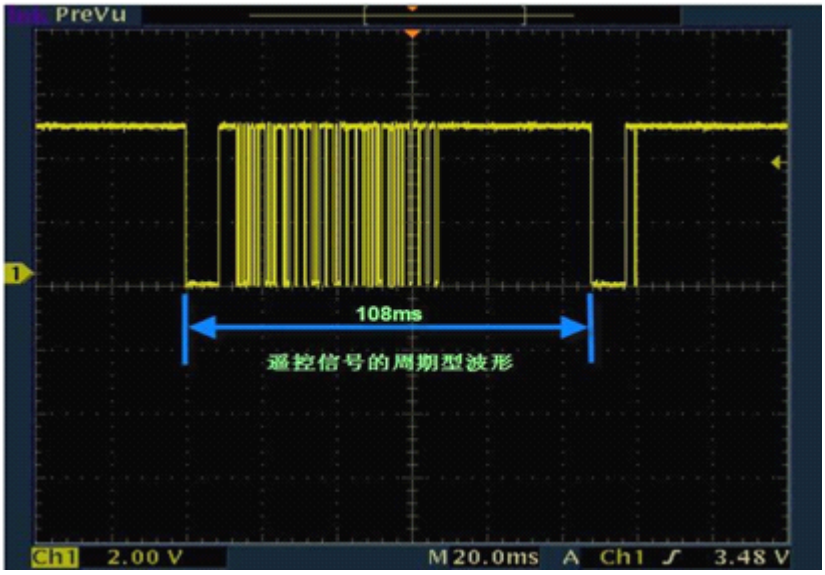
1：使用 38 kHz 载波频率

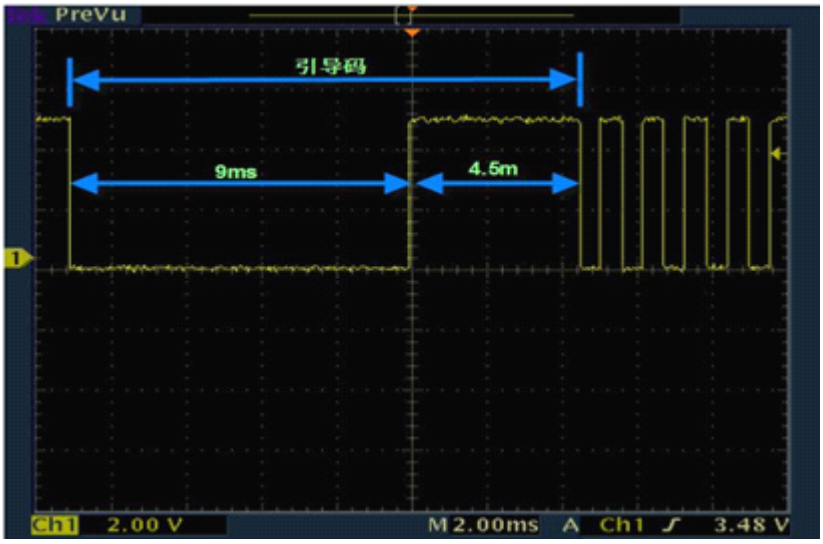
2：引导码间隔是 9 ms + 4.5 ms

3：使用 16 位客户代码

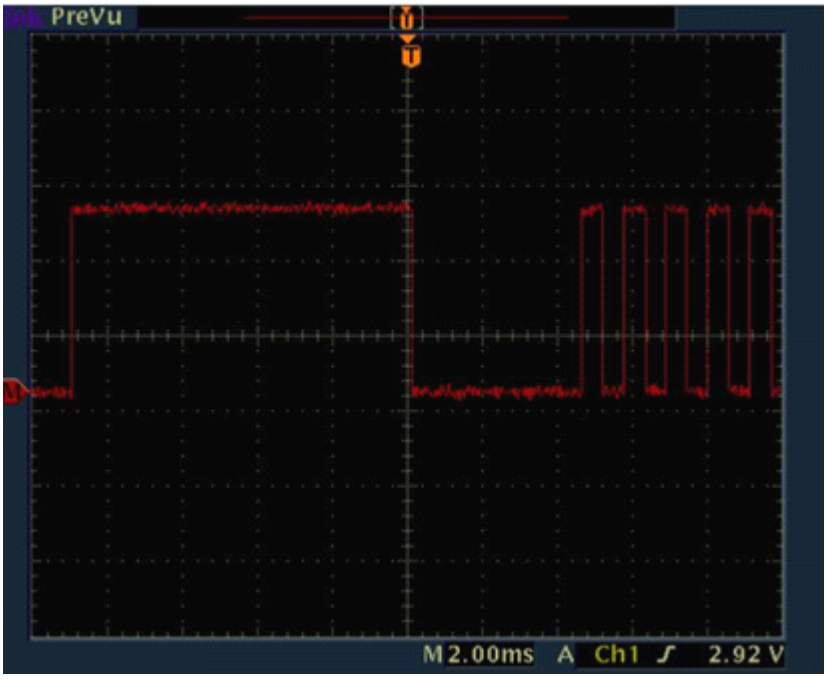
4：使用 8 位数据代码和 8 位取反的数据代码

下面的波形是从红外接收头上得到的波形：（调制信号转变成高低电平了）

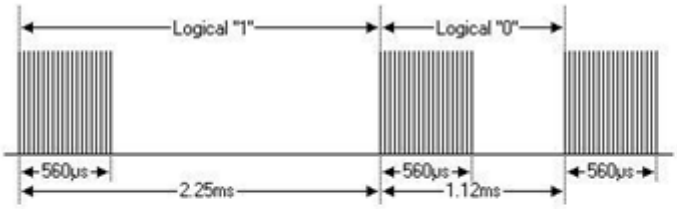


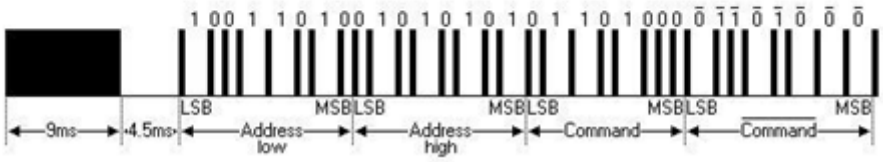


不过需要将波形反转一下才方便分析：

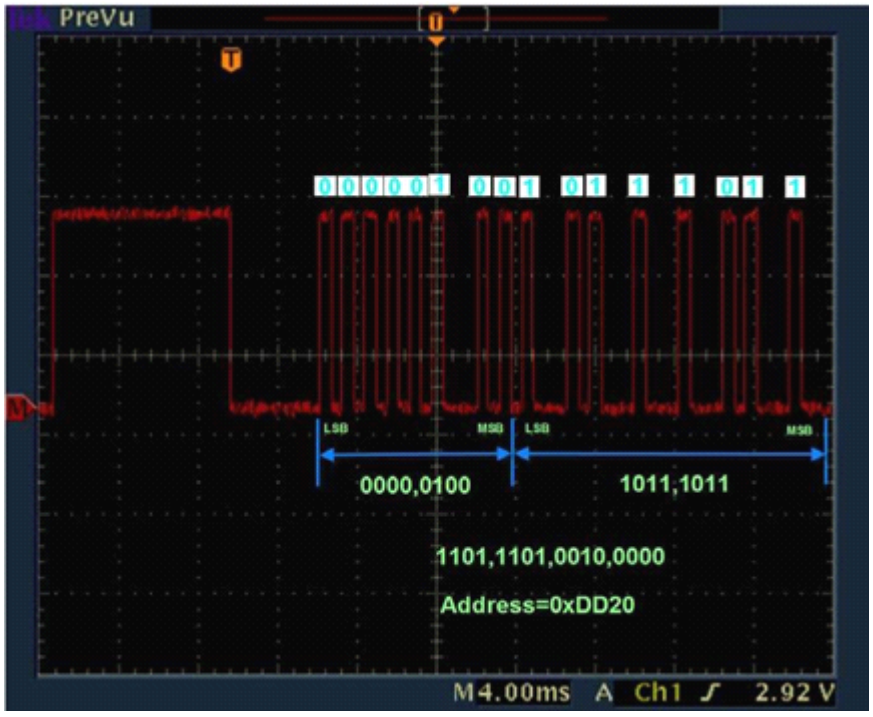


NEC 协议通过脉冲串之间的时间间隔来实现信号的调制（英文简写 PPM）。逻辑“0”是由 0.56ms 的 38KHZ 载波和 0.560ms 的无 载波间隔组成；逻辑“1”是由 0.56ms
的 38KHZ 载波和 1.68ms 的无载波间隔组成；结束位是 0.56ms 的 38K 载波。

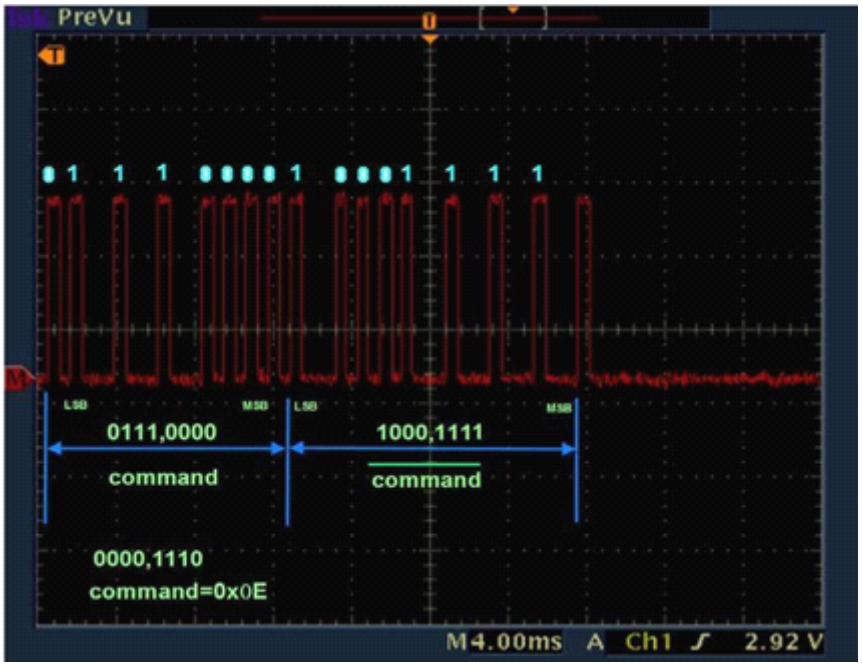




下面实例是某种遥控器的波形（注意：不是我们配套遥控器的波形，不同遥控器的主要区别是客户码不一样）：
遥控器的识别码是 Address=0xDD20;键值是 Command=0x0E；



注意波形先是发低位地址再发高位地址。所以 0000,0100,1011,1011 反转过来就是 1101,1101,0010,000 十六进制的 DD20；
键值波形如下：



也是要将 0111,0000 反转成 0000,1110 得到十六进制的 0E；另外注意 8 位的键值代码是取反后再发一次的，如图 0111,0000 取反后为 1000,1111。
最后一位是一个逻辑“1”。
2、配套红外遥控器参数
尺寸：8.5*4.0*0.5cm
键数：21 键

按键：PET 薄膜开关
电源：DC3.0V
电池：1*CR2025
发射 / 编码：NEC
各键客户码：统一为 BF 00
各键数据码如下：
第 1 行从左到右：00，01，02
第 2 行从左到右：04，05，06
第 3 行从左到右：08，09，0A
第 4 行从左到右：0C，0D，0E
第 5 行从左到右：10，11，12
第 6 行从左到右：14，15，16
第 7 行从左到右：18，19，1A
配套遥控器照片如下，使用前必须将下端的塑料片抽掉：

3、编程和实验结果
在数码管显示解码结果的红外遥控解码程序如下：
解码值在 IR_buff[2]中，当 IrOK=1 时解码有效。
用遥控器对准红外接收头，按下遥控器按键，在数码管后两位上就会显示对应按键的编码。
开发板设置：S2 的 4 位全部上拨，其余采用出厂默认设置。

实验程序

```
/*********************************************************************
*
 *
 *                              main.c
 *

*********************************************************************
 * 描      述：红外遥控解码程序(数码管显示)
 * 解码值在 IR_buff[2]中，当 IrOK=1 时解码有效。
 * 用遥控器对准红外接收头，按下遥控器按键，在数码管后两位上就会显示对应按键的编码
 * 开发板设置：S2 的 4 位全部上拨，其余采用出厂默认设置。
 * 开发平台：HL-K18 开发板+C18

*********************************************************************/

#include <p18cxxx.h>/*18F 系列单片机头文件*/
#include "k18.h"/*TOPPIC 开发板头文件*/
#include "delay.h"/*包含延时函数头文件*/

/*此处为晶振为 10 时的取值,如用其它频率的晶振时,要改变相应的取值。*/
```

```
#define Imax 18988
#define Imin 10850
#define Inum1 1967
#define Inum2 949
#define Inum3 4069

unsigned char IR_buff[4]={0x00,0x00,0x00,0x00};/*客户码低 8 位，客户码高 8 位，数据码，数据反码*/
unsigned char show[2]={0,0};
unsigned char flag_start;/*找到启动码标志*/
unsigned long m,Tc;
unsigned char IrOK;

/*0-F 共阴字形码表*/
const rom unsigned char sz[]={0x3f  , 0x06 , 0x5b , 0x4f , 0x66 ,
        0x6d ,0x7d , 0x07 , 0x7f  , 0x6f , 0x77 , 0x7c ,
        0x39 , 0x5e , 0x79 , 0x71 };

void display(void);

void PIC18F_High_isr(void);/*中断服务函数声明*/
void PIC18F_Low_isr(void);

#pragma code high_vector_section=0x8
/*高优先级中断响应时，会自动跳转到 0x8 处*/
/*利用预处理器指令#pragma code 来指定后面的程序在 ROM 中的起始地址为 0x08，*/
/*它是告诉连接器定位到特定的代码段,HIGH_INTERRUPT_VECTOR 是该特定代码段的段名*/
void high_vector (void)
{
    _asm goto PIC18F_High_isr _endasm/*通过一条跳转指令(汇编指令)，跳转到中断服务函数（中断服务程序）处*/
}

#pragma code low_vector_section=0x18
/*低优先级中断响应时，会自动跳转到 0x18 处*/
void low_vector (void)
{
    _asm goto PIC18F_Low_isr _endasm
}

#pragma code
/*这条语句不是多余的，它是告诉连接器回到默认的代码段，*/
```

```
/*如果不加的话，连接器就会傻傻地把后面的代码紧跟着上面的代码一直放下去。*/
/*而 18f4520.lkr 文件里定义了向量区地址最多到 0x29，所以如果没加此句通常会报错*/


/*---高优先级中断服务程序---*/
#pragma interrupt PIC18F_High_isr
/*利用预处理器指令#pragma interrupt 来声明后面的函数是低优先级中断服务函数（中断服务程序），*/
/*注意：关键字是 interrupt，和低优先级中断时不同*/
/*一旦指定后面的函数是低优先级中断服务程序，系统在进入该函数时，会自动保护现场，退出前自动恢复现场，*/
/*同时中断服务程序执行完毕后，会自动返回断点,*/
/*中断服务函数前必须加该语句*/
void PIC18F_High_isr (void)
{
    /*INT1 解码程序*/
    /*提取中断时间间隔时长。TMR0 读：先读低字节，后读高字节*/
    Tc=TMR0L;
    Tc=TMR0H*256+Tc;

    TMR0H=0;/*TMR0 置初值,先写高字节，后写低字节*/
    TMR0L=0;
    INTCON3bits.INT1IF=0;/*INT1 溢出标志清零*/

    /*寻找启始码*/
    if((Tc>Imin)&&(Tc<Imax))
    {
        m=0;
        flag_start=1;
        return;
    }

    /*找到启动码后，进入解码流程*/
    if(flag_start==1)
    {
        if(Tc>Inum1&&Tc<Inum3)
        {
            IR_buff[m/8]=IR_buff[m/8]>>1|0x80; m++;/*取码 1*/
        }
        if(Tc>Inum2&&Tc<Inum1)
        {
            IR_buff[m/8]=IR_buff[m/8]>>1; m++;/*取码 0*/
```

```
        }

        /*取码完成后判断码是否正确*/
        if(m==32)
        {
            m=0;
            flag_start=0;
            if(IR_buff[2]==~IR_buff[3])
            {
                IrOK=1;
            }
            else IrOK=0;
        }
    }
}

/*---低优先级中断服务程序---*/
#pragma interruptlow PIC18F_Low_isr
/*注意：关键字是 interruptlow，和高优先级中断时不同*/
void PIC18F_Low_isr (void)
{
}

void main(void)/*主程序*/
{
    RCONbits.IPEN=1; /*使能中断优先级*/
    k18_init();/*HL-K18 开发板初始化*/
    m=0;
    flag_start=0;

    T0CON=0b00000000;/*TMR0 设置：停止运行、16 位定时，预分频 1:2*/
    TMR0H=0;/*TMR0 置初值,先写高字节，后写低字节*/
    TMR0L=0;
    T0CONbits.TMR0ON=1;/*启动 TMR0*/

    INTCON2bits.INTEDG1=0;/*设定外部中断 1 触发边沿(下降沿)*/
    INTCON3bits.INT1IP=1;/*设置外部中断 1 为高优先级*/
    INTCON3bits.INT1IE=1;/*INT1 中断使能*/
    INTCONbits.GIE=1;/*全局中断允许*/

    while(1)
    {
        if(IrOK==1)
        {
```

```
            show[0]=IR_buff[2] & 0x0F;//取键码的低四位
            show[1]=IR_buff[2] >> 4; //取键码的高四位
            IrOK=0;
        }
        display();/*在数码管后两位上就会显示对应按键的编码*/
    }
}

/*在数码管后两位上更新数据码*/
void display(void)/*显示函数*/
{
    unsigned char wwm=0;
    COL4=0;/*取消千位的位选*/
    PORTD=sz[show[0]];/*送出个位的字形码*/
    COL1=1;/*个位的位选*/
    delayms(2);/*延时*/

    wwm=show[0];
    PORTD=sz[wwm];
    PORTD=sz[0];


    PORTD=sz[show[0]];/*送出个位的字形码*/
    COL1=1;/*个位的位选*/
    delayms(2);/*延时*/


    COL1=0;/*取消个位的位选*/
    PORTD=sz[show[1]];/*送出十位的字形码*/
    COL2=1;/*十位的位选*/
    delayms(2);/*延时*/

    COL2=0;/*取消十位的位选*/
    PORTD=0x00;/*送出百位的字形码*/
    COL3=1;/*百位的位选*/
    delayms(2);/*延时*/

    COL3=0;/*取消百位的位选*/
    PORTD=0x00;/*送出千位的字形码*/
    COL4=1;/*千位的位选*/
    delayms(2);/*延时*/
}
```

下面是有缘人看的，谢谢理解

善有善报，恶有恶报，不是不报，时候未到。
从古至今，阴司放过谁，大家得多行善。
行善积德，爱护动物，哪怕小蚂蚁也是生命。
可改变命运，可心想事成，有利保佑子孙后代更昌盛。
学习弟子规，教我们如何做人，看和谐拯救危机，教我们看宇宙。
看为什么不能吃它们，教我们慈悲心，看因果轮回纪录，教我们懂得因果报应。
切勿造恶，种瓜得瓜种豆得豆，一切都有过程，待成熟之时，福德或果报自来找你。

慧净
2008 年 8 月 8 日