

HT6221 发码的接收

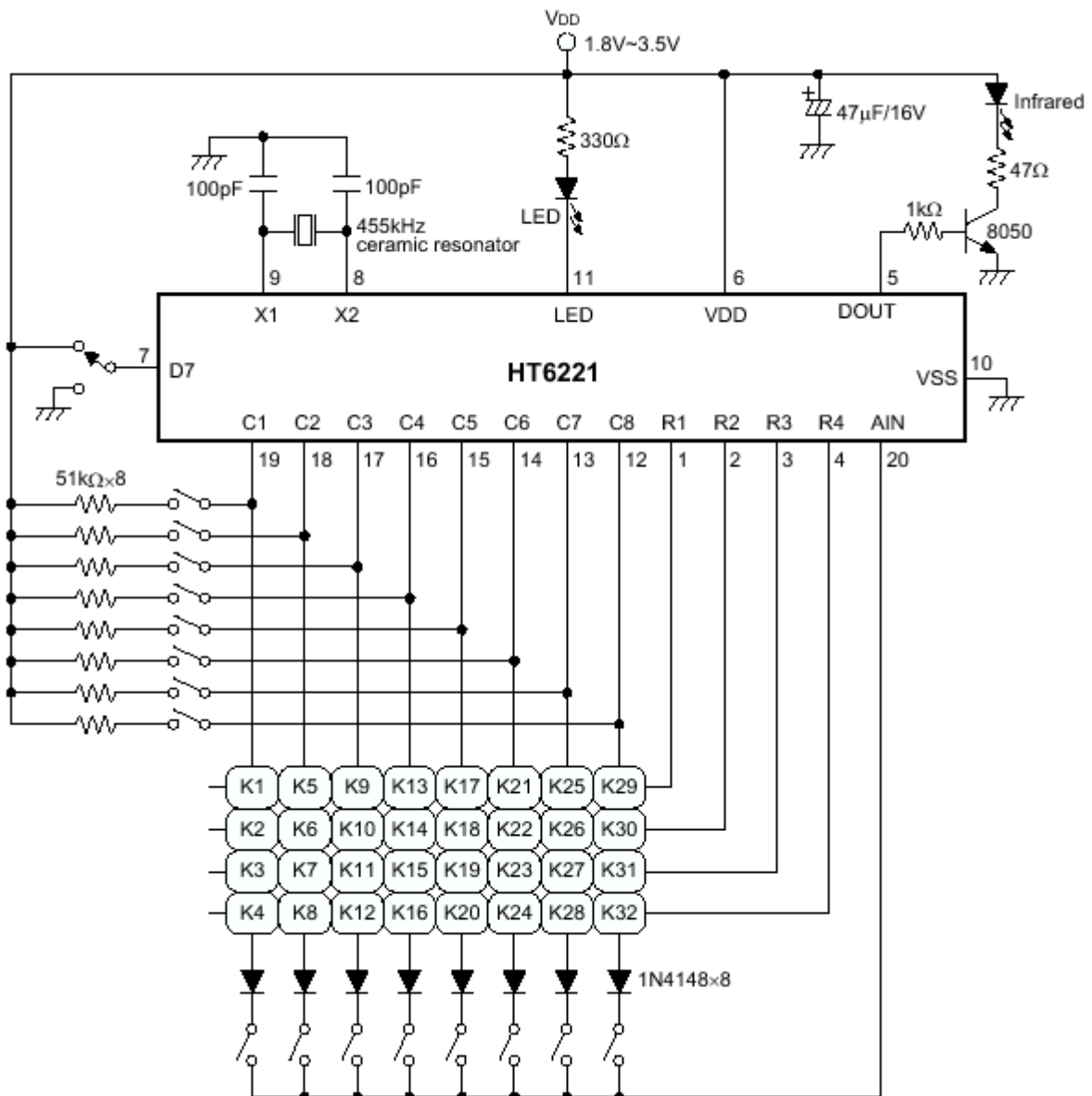
作者： 盛扬半导体（上海）有限公司软件部

时间： 2001/8/6

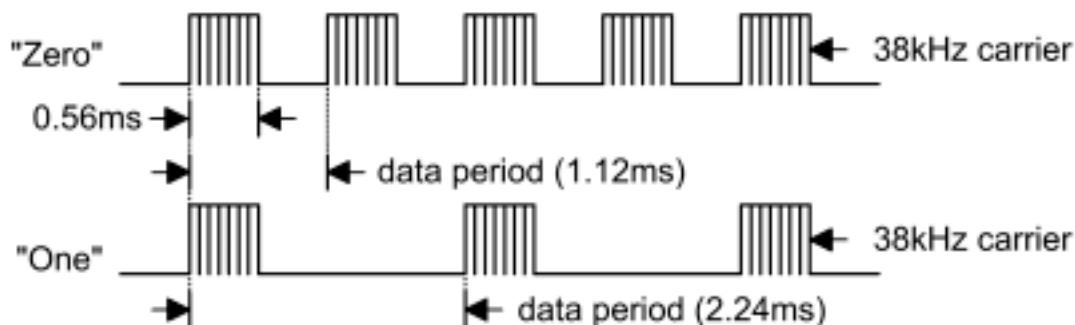
适用单片机： 48 系列单片机

简介：

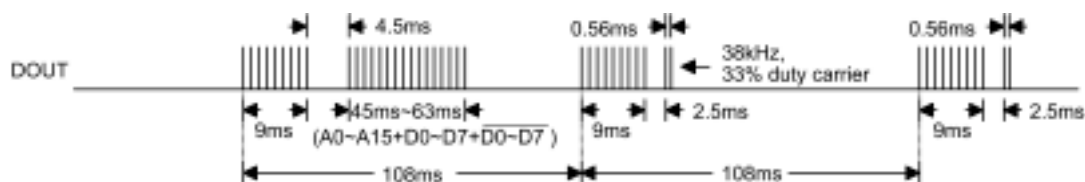
HT6221 的应用电路图如下：



HT6221 是 Holtek 公司生产的多功能编码芯片，采用 PPM（Pulse Position Modulation）进行编码，1.12ms 为 0，2.24ms 为 1，如下图：



每发送一个码，HT6221 会先送出一个 9ms 的头码和 4.5ms 的间隙，然后依次送出 16 位的地址码（18ms~36ms）、8 位数据码（9ms~18ms）和 8 位数据反码，如下图：



本文主要介绍用 HT48C30 来进行解码的程序。HT6221 通过红外发射管发出信号，红外接收管接到 HT48C30 的外部中断输入脚。

```

;File name:6221receiver.asm
;作者： 谢志刚
;说明： 程序开始时，先清除 RAM 区，然后打开主中断、外部中断及定时计数器中断。通
;过计算中断之间的时间间隔来解码，当接收到有效码后，会置标志位。在主程序中判
;断标志位是否置位来判断有效码的接收。当接受到有效码时，则跳到相应的程序（用
;户自定义）中去执行。掩膜选择系统时钟为 4000kHz。
#include ht48c30.inc
int_times      equ    [40h]      ;记录外部中断次数
tmr_times      equ    [41h]      ;记录定时/计数器中断次数
head_ok        equ    [42h]      ;9ms 头码是否接收到标志
code_data      equ    [43h]      ;信息码寄存器
lanth          equ    [45h]      ;记录单个码为 1 或 0
flag           equ    [46h]      ;是否接收到有效码标志
;*****
code   .section  'code'
      org     00h
      jmp     start
      org     04h
      jmp     do_int
      org     08h
      jmp     do_tmr

```

```
org    20h
jmp    do_key0
jmp    do_key1
jmp    do_key2
jmp    do_key3
jmp    do_key4
jmp    do_key5
jmp    do_key6
jmp    do_key7
jmp    do_key8
jmp    do_key9
jmp    do_keya
jmp    do_keyb
jmp    do_keyc
jmp    do_keyd
jmp    do_keye
jmp    do_keyf
jmp    do_key10
jmp    do_key11
jmp    do_key12
jmp    do_key13
jmp    do_key14
jmp    do_key15
jmp    do_key16
jmp    do_key17
jmp    do_key18
jmp    do_key19
jmp    do_key1a
jmp    do_key1b
jmp    do_key1c
jmp    do_key1d
jmp    do_key1e
jmp    do_key1f
org    40h
start:
    mov    a, 40h
    mov    mp0,a
ram_clr:                                ;清除内存
    clr    r0
    inc    mp0
    sdz    acc
    jmp    ram_clr
    clr    status
    set    intc.0                        ;开启主中断
```

```
    set    intc.1           ;开启外部中断
    clr    tmr              ;清除定时/计数器初值
    mov    a, 90h
    mov    tmrc,a          ;设定定时/计数器工作在内部计数模式
    set    intc.2          ;开启定时/计数器中断
sleep:
    sz     flag            ;判断是否接收到有效码
    jmp    get_key        ;如果接收到,取键值
    jmp    sleep          ;如果没有,等待
get_key:
    clr    flag            ;清除标志
    mov    a, 20h
    add    a, code_data
    mov    pcl,a          ;根据接收到的码来决定待处理的键
do_key0:
    jmp    $
do_key1:
    jmp    $
do_key2:
    jmp    $
do_key3:
    jmp    $
do_key4:
    jmp    $
do_key5:
    jmp    $
do_key6:
    jmp    $
do_key7:
    jmp    $
do_key8:
    jmp    $
do_key9:
    jmp    $
do_keya:
    jmp    $
do_keyb:
    jmp    $
do_keyc:
    jmp    $
do_keyd:
    jmp    $
do_keye:
    jmp    $
```

```
do_keyf:
    jmp    $
do_key10:
    jmp    $
do_key11:
    jmp    $
do_key12:
    jmp    $
do_key13:
    jmp    $
do_key14:
    jmp    $
do_key15:
    jmp    $
do_key16:
    jmp    $
do_key17:
    jmp    $
do_key18:
    jmp    $
do_key19:
    jmp    $
do_key1a:
    jmp    $
do_key1b:
    jmp    $
do_key1c:
    jmp    $
do_key1d:
    jmp    $
do_key1e:
    jmp    $
do_key1f:
    jmp    $
;*****
do_tmr:                                ;定时/计数器中断程序
    inc    tmr_times
    reti
;*****
do_int:
    clr    tmrc.4                       ;进入外部中断, 关闭定时/计数器中断
    clr    intc.2
    inc    int_times                     ;中断次数加1
judge_head:                             ;判断 9ms 的头码
```

```

sz    head_ok
jmp   decode
clr   c
mov   a, tmr_times           ;根据定时/计数器溢出中断次数判断码值
sub   a, 31h
snz   c
jmp   wrong_code           ;无效头码, 退出返回
clr   c
mov   a, 37h
sub   a, tmr_times
snz   c
jmp   wrong_code           ;无效头码, 退出返回
set   head_ok              ;有效头码, 置标志位
jmp   ok_next              ;重新开中断, 以处理下一次中断
decode:                     ;解码
call  one_or_zero          ;调用子程序来判断 0 或 1
clr   c
mov   a, int_times
sub   a, 12h                ;从第 17 位开始解码来取得 8 位数据码
sz    c
jmp   decode_ok
sz    lanth
jmp   $+2
jmp   wrong_code
jmp   ok_next
decode_ok:
mov   a, 19h
sub   a, int_times
sz    c
jmp   $+2
jmp   code_ok
jmp   get_code
sz    lanth
jmp   ok_next
jmp   wrong_code
get_code:                   ;通过循环移位取得完整的 8 位数据码
snz   lanth.0
jmp   get_one
clr   code_data.0
rr    code_data
jmp   ok_next
get_one:
set   code_data.0
rr    code_data

```

```
        jmp    ok_next
code_ok:                                ;取得有效码, 置标志位
        set    flag
        ret
wrong_code:                              ;取得无效码的处理
        clr    tmr_times
        clr    lanth
        clr    tmr
        clr    head_ok
        clr    int_times
        mov    a, 90h
        mov    tmrc,a
        set    intc.2
        reti
ok_next:                                  ;取得有效单个码值的处理
        clr    tmr_times
        clr    lanth
        clr    tmr
        mov    a, 90h
        mov    tmrc,a
        set    intc.2
        reti
one_or_zero  proc                        ;判断单个码值的子程序
        clr    lanth
        mov    a, tmr_times
        sub    a, 2h
        snz    c
        ret
        mov    a, 6h
        sub    a, tmr_times
        snz    c
        jmp    one
        set    lanth.0
        ret
one:
        mov    a, tmr_times
        sub    a, 7h
        snz    c
        ret
        mov    a, 0bh
        sub    a, tmr_times
        snz    c
        ret
        set    lanth.1
```

```
ret  
one_or_zero endp  
end
```

注：若不使用 HT6221 发码，也可使用 HT48CA0 编码发射 HT6221 码，可参考相关文档 HA0041s.doc（使用 HT48CA0 发射 HT6221 码）。