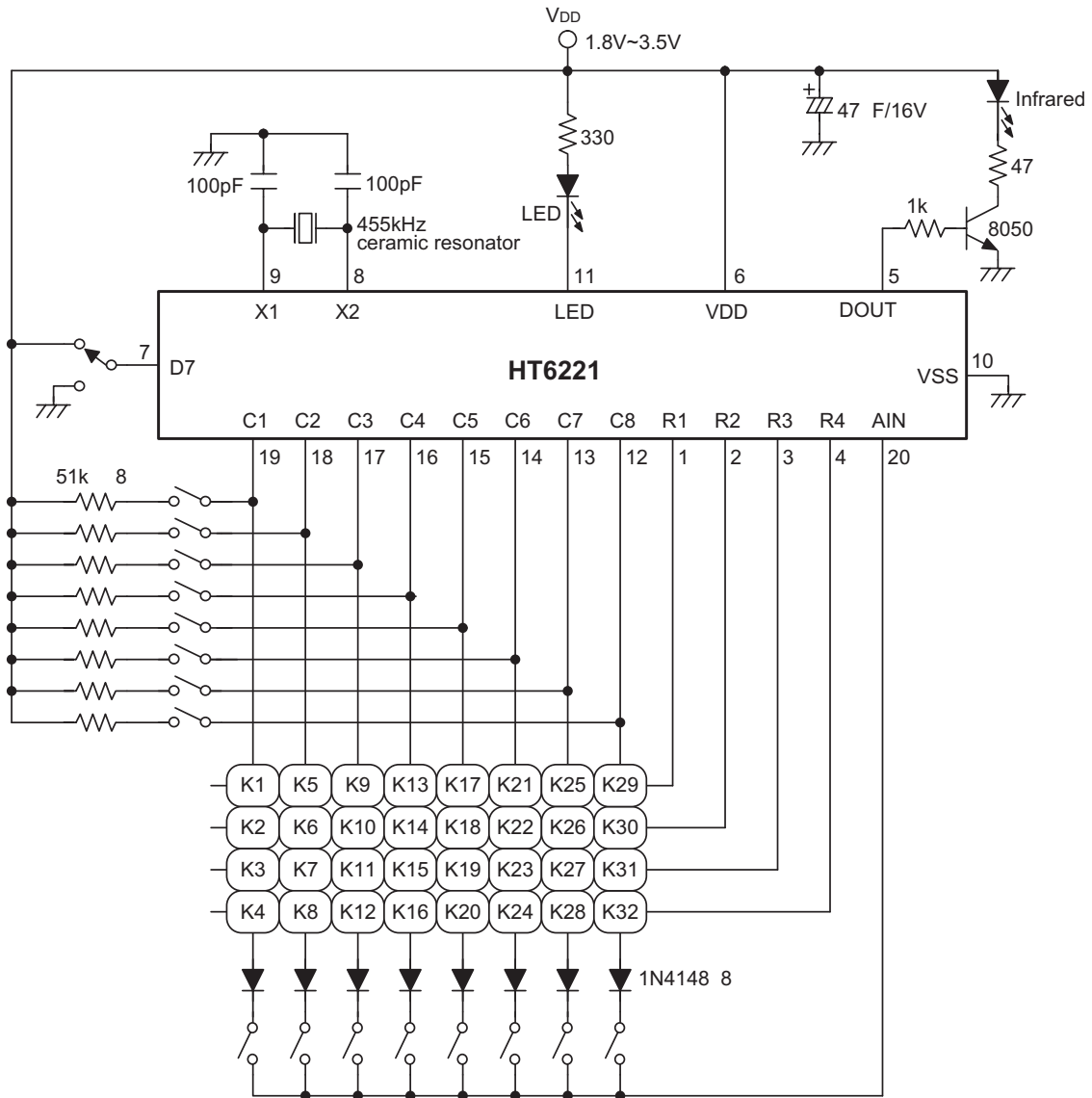


HT6221 发码的接收

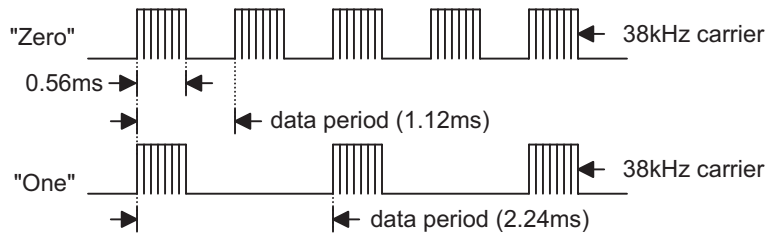
文件编码: HA0040s

简介:

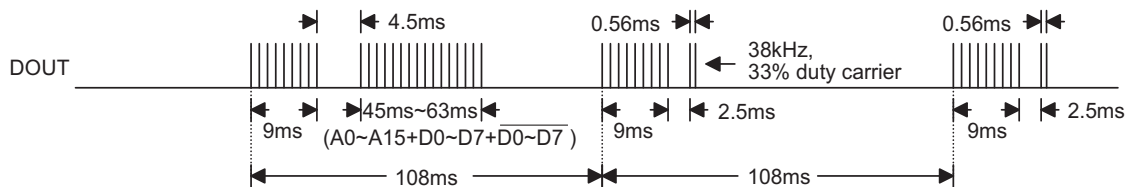
HT6221 的应用电路图如下:



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 HT48R30A-1 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 HT48R30A-1 的外部中断输入脚。

```
;File name:6221receiver.asm
;作者: KELVEN
;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。掩膜选择系统时钟为 4000kHz。
```

```
include ht48r30a-1.inc
; *****
; * Filename      : 6221RECEIVER.asm          *
; * Function      : DECODER HT6221           *
; * Microprocessor : HT48R30A-1              *
; * Crystal       : 4MHz                     *
; *****
data.section 'data'

card_no0      equ    [060h]      ;
card_no1      equ    [061h]
card_no2      equ    [062h]
card_no3      equ    [063h]      ;记录最终的数据

temp          equ    [07eh]
cint          db    ?           ;记录收数据 bit 个数的变量
```

```

count          db  ?
count_buf      db  ?
int_acc        db  ?          ;中断保护变量
right_tou      dbit
y_bit          dbit
;-----
code.sectionat 0000 'code'
    org 00h

    jmp     start

    org    04h          ;中断入口地址
    jmp    do_wai

    org    08h
    jmp    do_tmr      ;有 time 中断发生

    org    20h
start:
    set    pgc          ;中断口设为输入状态
    call   init
    mov    a, 81h       ;fsys/4 1M
    mov    tmrc, a
    mov    a, 56
    mov    tmr, a       ;200μs 中断一次
    mov    a, 6
    mov    intc, a
    set    tmrc.4
;-----
sleep:
    set    intc.0
    snz    y_bit        ;判断是否有外部中断发生?
    jmp    sleep        ;没有外部中断发生则继续等待
    call   decode_1     ;解码
    snz    right_tou    ;解码不正确, 返回重来
    jmp    start
    mov    a, cint
    sub    a, 32        ;准备收数据的 bit 个数
    snz    c            ;判断数据是否已收完
    jmp    sleep

    clr    tmrc         ;收码完毕
    clr    intc
    cpla   card_no2
    xor    a, card_no3  ;校验所收码是否正确

```

```

    snz     z
    jmp     error1
    jmp     start          ;正确
error1:
    jmp     start          ;错误
;-----
do_wai :
    clr     tmrc.4
    clr     intc.0
    mov     int_acc, a     ;入中断保护, push acc 值暂存
    mov     a, count
    mov     count_buf, a
    clr     count
    set     y_bit
over_int:
    mov     a, 56
    mov     tmr, a
    mov     a, int_acc     ;pop acc 值
    set     tmrc.4
    set     intc.0
    reti
;-----
do_tmr:
    inc     count
    reti
;~~~~~
decode_1 proc
    clr     intc.0
    clr     y_bit
    snz     right_tou
    jmp     judge_touma
    mov     a, count_buf
    sub     a, 4
    snz     c
    jmp     error          ;data 值小于 200μs*4=800μs 判断错误
    mov     a, count_buf
    sub     a, 13
    sz     c
    jmp     error          ;data 值大于 200μs*13=2.6ms 判断错误
    mov     a, count_buf   ;data 值大于 200μs*8=1.6ms
    sub     a, 8           ;c=1, data=1
                                ;data 值小于 200μs*8=1.6ms  c=0, data=0
    rrc     card_no3

```

```

    rrc    card_no2
    rrc    card_no1
    rrc    card_no0
    inc    cint           ;记录收到的 BIT 位数
    ret
error:
    clr    count
    clr    right_tou
    ret
judge_touma:
    clr    right_tou     ;头码判断 13.50ms
    mov    a, count_buf
    sub    a, 70         ;200*70=14ms
    sz     c
    ret     ;头码大于 14ms 判断错误
    mov    a, count_buf
    sub    a, 63         ;63*200=12.6ms
    snz    c
    ret     ;头码小于 12.6ms 判断错误
    set    right_tou
    ret
decode_1 endp
;~~~~~
;Function : init
;Purpose  : clear ram value
;Parameter:
;Return   :
;Modified : acc, status
;~~~~~
init proc
    clr    intc           ;清中断
    mov    a, 20h
    mov    mp0, a
    mov    a, 5fh
    mov    temp, a
ram_clr: ;清 RAM
    clr    r0
    inc    mp0
    sdz    temp
    jmp    ram_clr
    ret
init endp

```