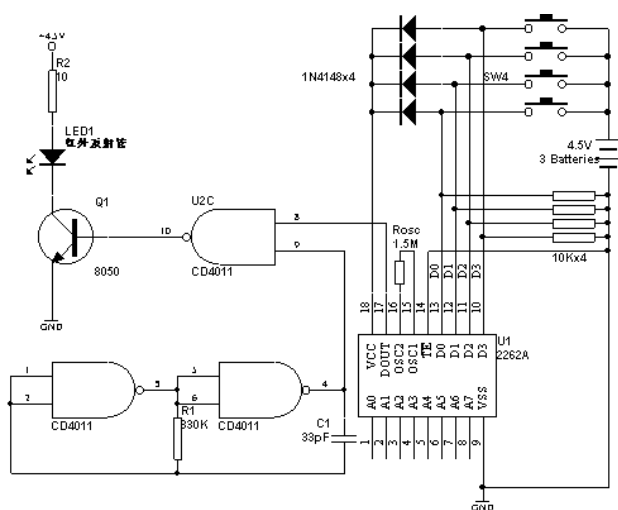


PT2262、PT2272 和 P87LPC764 单片机的接口设计

PT2262 和 PT2272 是一对专用的集成编解码器,当采用 4 位数据通信的时候可以有 6561 组码, A0--A7 为三态地址编码,即接正电源、接地和悬空。当采用 PT2262 和 P87LPC764 组成的遥控系统中可以把解码的部分都可以省略掉但相对的解码的速度会慢一点,如果用 PT2272 来解码,可以加快解码的速度减小需要耗费的单片机资源。下面以红外遥控为例,简要的说一下。

一、发射部分:

电路图如下:

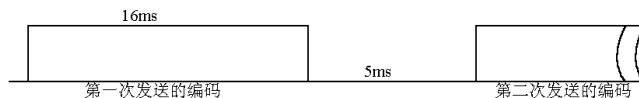


其中这里采用了 4 位数据码、8 位地址码的通信方式,地址编码端口为 A0~A7,图中 A0..A7 的编码为悬空,TE 为编码输出控制端,当接低电平时,PT2262 就通过第 17 脚不断地串行输出其编码串。对于每传送一组编码都自动连发四次,编码器用不同的占空比及组合表示不同的状态。

$$\text{振荡频率 } f \approx 1000 \times 16 / R_{osc} (\text{K}\Omega) \text{ KHz}$$

图中 Rosc 采用了 1.5M 的电阻,振荡频率约为 22KHz。

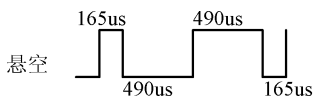
PT2262 的发送格式如下:



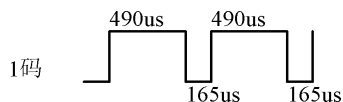
当按键有效时 PT2262 发送四次编码,每发送一次编码都有 5ms 宽度的低电平分开,而 16ms 的编码是有 A0~A11 十二个码组成, A0 在前, A11 在后。我们现在把 A8~A11 用来当作数据发送了。

PT2262 有三种编码:悬空、1 和 0。

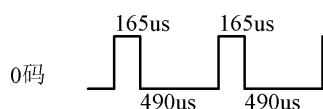
(1) 悬空时发送的码位如下:



(2) 接高电平时发送的码位如下:



(3) 接低电平时发送的码位如下:



二、用 P87LPC764 单片机来解码

接收到的信号为发送信号的反码，因为每检测到按键有效一次，PT2262 都发送四次编码，我们可以利用这一特点来解码。先检测有没有接收到信息，当有的时候我们就去掉第一次编码的信号，检测 5ms 的高电平来检测编码的开始信号，接收到 5ms 的高电平后，才开始解码。我们可以发现每个码的长度都是 1.3ms 左右，每一个编码都是由低电平开始，然后到高电平，又到低电平，又回到高电平，我们从第一个高电平的宽度可以把 1 码区分出来，剩下的悬空码和 0 码可以从第二个高电平的宽度区分出来。

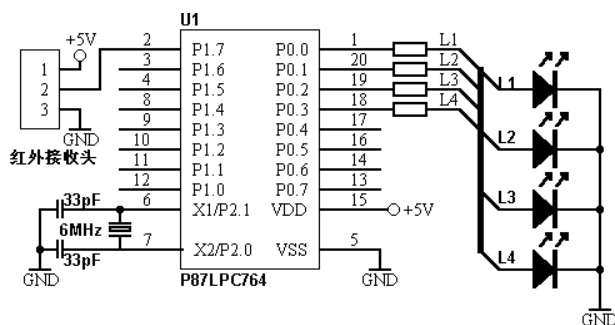
具体的单片机译码方法如下:

由高电平开始，检测到下降沿时就延时 300us，读取接收的状态记为 A0，然后再检测下一个下降沿，下降沿后又延时 300us，读取接收的状态记为 A1。这样就一个编码给译出来了。

A0, A1 和悬空、1 码、0 码的关系如下:

A0	A1	代码
0	0	1 码
0	1	错误
1	0	悬空
1	1	0 码

接收部分的电路图如下:



具体的程序如下:

```

REM EQU P1.7
P0M1 EQU 84H
P0M2 EQU 85H
P1M1 EQU 91H
P1M2 EQU 92H
P2M1 EQU 0A4H
P2M2 EQU 0A5H
    
```

;接收正确位

```

RECEIVE BIT 20H.0
    
```

;因为当按住 PT2262 的按键不放的时候, PT2262 会把编码不断的送出
;设置 ENABLE 位用来检测按键有没有放开过, 如果没有放开则不再响应

ENABLE BIT 20H.1

;30H, 32H:接收的 8 位地址编码

;31H, 33H:接收的 4 位数据编码

;36H, 37H:设定的 8 位地址密码

;其中以 30H.0 和 32.0 为例

;悬空: 30H.0=1, 32H.0=0

;1: 30H.0=0, 32H.0=0

;0: 30H.0=1, 32H.0=1

;38H:输出控制

ORG 0000H

LJMP BEGIN

ORG 0100H

BEGIN:

MOV SP, #60H

MOV P1M1, #0FFH ;设定 P1 口为斯密特输入状态

MOV P1M2, #00H

MOV P0M1, #00H ;设定 P0 口为上拉输出配置

MOV P0M2, #0FFH

MOV P0, #00H

MOV P1, #0FFH

;编码设定为:A0~A7 为悬空

MOV 36H, #0FFH

MOV 37H, #00H

LOOP:

LCALL REMOTE ;接收遥控码

LCALL WORK ;控制输出

JMP LOOP

REMOTE:

;检测 50ms 内有没有编码接收

MOV 40H, #50

REMOTEA:

MOV 41H, #250

REMOTEB:

JNB REM, REMOTE0 ;有码发送, 则跳转

DJNZ 41H, REMOTEB

DJNZ 40H, REMOTEA

CLR ENABLE

REMOTE_END:

RET

REMOTE0:

JB ENABLE, REMOTE_END ;按键没有放开则返回

;清除上次解码内容

;接收 12 位编码

MOV 34H, #12

;解码

;先找出接收码的开头, 即 5ms 左右的高电平

;设置高电平时间为 4~6ms

;检测和等待 4ms 的高电平

REMOTE1:

MOV 40H, #40

REMOTE2:

MOV 41H, #25

REMOTE3:

JNB REM, REMOTE1

DJNZ 41H, REMOTE3

DJNZ 40H, REMOTE2

;等待在 2ms 内接收到的低电平

MOV 40H, #20

REMOTE4:

MOV 41H, #25

REMOTE5:

;4ms 到 6ms 内接收到下降沿, 则跳去解码, 否则返回

JNB REM, REMOTE6

DJNZ 41H, REMOTE5

DJNZ 40H, REMOTE4

;超出 6ms, 接收错误, 返回

CLR ENABLE

JMP REMOTE_END

REMOTE6:

;等待 300us 后, 采集接收信号

MOV 40H, #150

DJNZ 40H, \$

;采集接收信号, 并记录

MOV C, REM

MOV A, 31H

RLC A

MOV 31H, A

MOV A, 30H

RLC A

MOV 30H, A

;等待第二个下降沿

JNB REM, \$

JB REM, \$
;等待 300us 后采集接收信号

MOV 40H, #150
DJNZ 40H, \$
MOV C, REM
MOV A, 33H
RLC A
MOV 33H, A
MOV A, 32H
RLC A
MOV 32H, A

;等待第二个码值的下降沿

JNB REM, \$
JB REM, \$
DJNZ 34H, REMOTE6

;把接收的编码左移 4 位, 将 8 位密码放在同一字节上

MOV 40H, #4

REMOTE7:

CLR C
MOV A, 31H
RLC A
MOV 31H, A
MOV A, 30H
RLC A
MOV 30H, A
CLR C
MOV A, 33H
RLC A
MOV 33H, A
MOV A, 32H
RLC A
MOV 32H, A
DJNZ 40H, REMOTE7

;把 4 位数据编码由高 4 位移到低 4 位上

MOV A, 31H
SWAP A
MOV 31H, A
MOV A, 33H
SWAP A
MOV 33H, A

;比较密码

MOV A, 30H
XRL A, 36H
JNZ REMOTE8

;密码不正确则跳转

```
MOV    A, 32H
XRL    A, 37H
JNZ    REMOTE8           ;密码不正确则跳转
;置接收正确位
```

```
SETB   RECEIVE
;置 ENABLE 用于检测按键有没有放开
```

```
SETB   ENABLE
RET
```

REMOTE8:

```
CLR    ENABLE
CLR    RECEIVE
RET
```

;把选中的 LED 取反

WORK:

```
JNB    RECEIVE, WORK_END
```

;每次按键都一次响应

```
CLR    RECEIVE
MOV    A, 33H
CPL    A
ANL    A, #0FH
MOV    33H, A
MOV    A, 38H
XRL    A, 33H
ANL    A, #0FH
MOV    38H, A
MOV    P0, 38H
NOP
```

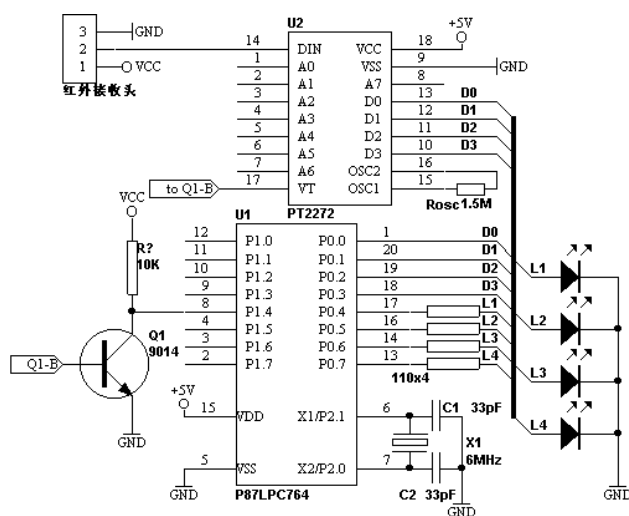
WORK_END:

```
RET
```

```
END
```

三、用 PT2272 接收解码和 P87LPC764 接口

电路图如下：



图中的接线方法当 PT2272 接收发射过来的信号时，VT 脚输出一个正脉冲，经过 9014 的取反后，向单片机 P87LPC764 产生一个中断请求，CPU 接到中断信号后读取 PT2272 芯片的内容，然后控制发光二极管。