

# 微机系统与接口实验-第二次实验报告

学号: 21210710

姓名: 宋斌

## 测试题目:

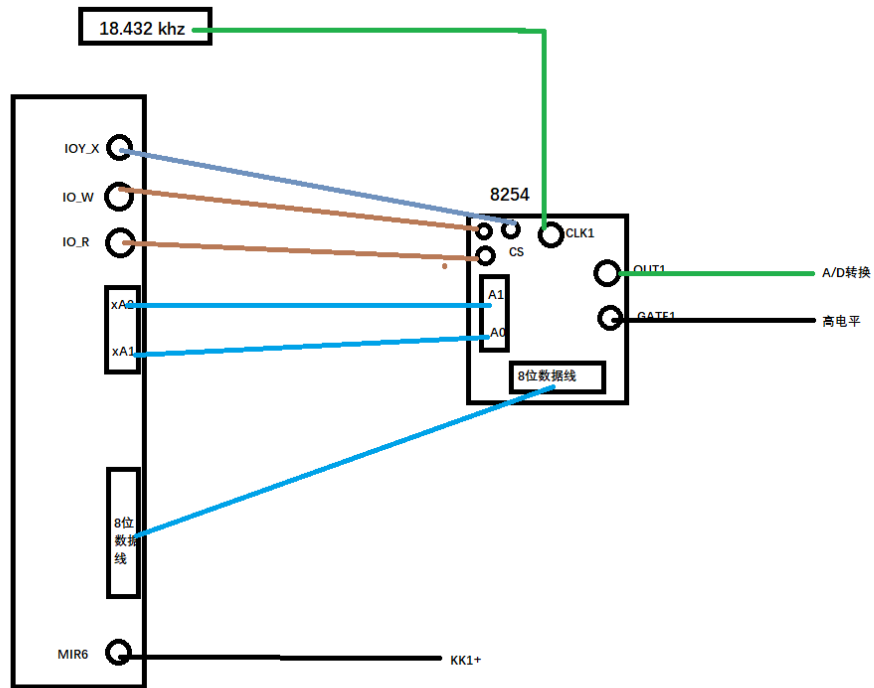
使用中断信号来控制输出的方波周期

## 实验原理:

1. 8254
2. 8259

## 连线:

(大致是这样的)



基本原理:

使用8254的工作方式3，当计数减为一半的时候输出低电平，从而输出方波。

使用中断控制方波的周期，只要修改计数值即可。

## 程序框图:

```
uint8_t old_status = 0xff;
uint8_t new_status = 0x00;

中断处理程序
{
    new_status = new_status + 1;
}

main()
{
    1. 安装中断处理程序
    2. 初始化8259
```

3. 初始化8254 ,计数值为0x4800 / 4

```
while(1)
{
    if(old_status == new_status)
        continue;

    old_status = new_status;

    switch(old_status)
    {
        case 0:
            //0.25的周期, 什么也不用做, 因为一开始就已经是这个周期了
            break;
        case 1: //周期0.5s
            改变8254 计数值为 0x2400
            break;
        case 2: //周期1s
            改变8254 计数值为 0x4800
            break;
        default : //周期2s
            改变8254 计数值为 0x9000
            break;
    }
}
```

或者直接在中断处理程序里面修改8254的计数值也可以

## 程序代码:

```
DATA    segment
        db 00h
        db 00h
DATA    ends

CODE    SEGMENT
ASSUME  CS:CODE,DS:DATA

;中断处理程序
irq6_handler:
    push ds
```

```

    push ax
    mov ax,DATA
    mov ds,ax

    ;update current n
    mov al,ds:[1]
    inc al
    mov ds:[1],al
    ;

    pop ax
    pop ds
    iret

install_interruption_handlers:
    push es

    xor ax,ax
    mov es,ax

    mov ax, OFFSET irq6_handler
    mov si, 0038H
    mov es:[si], ax

    mov ax, cs
    mov si, 003AH
    mov es:[si], ax

    pop es
    ret

init_8259:
    ;init 8259A
    cli

    mov al, 11H
    out 20H, al        ;ICW1

    mov al, 08H
    out 21H, al        ;ICW2

    mov al, 04H
    out 21H, al        ;ICW3

    mov al, 03H
    out 21H, al        ;ICW4

    mov al, 2FH        ;OCW1
    out 21H, al

```

```

        sti
        ret

init_8255:
        ;init 8255
        mov dx,686h
        mov al,90h
        out dx,al
        ret

main:
        ;install interrupt handler.
        ; IP : CS
        call install_interruption_handlers
        call init_8259
        call init_8255

        mov ax,DATA
        mov ds,ax
        mov byte ptr ds:[0],0FFh        ;last_n = 0FF
        mov byte ptr ds:[1],00h        ;current_n

        ;init 8254
        mov dx,0606h        ;counter 1
        mov al,076h
        out dx,al

        ;
loop_body:
        mov al,ds:[0]
        mov bl,ds:[1]

        cmp al,bl
        jz  do_nothing

        mov al,bl        ;update last n
        mov ds:[0],al

case_0:
        cmp al,00h
        jnz case_1
        mov dx,0602h
        mov al,00h
        out dx,al

        mov al,12h
        out dx,al
        jmp do_nothing

case_1:
        cmp al,01h

```

```

    jnz case_2

    mov dx,0602h
    mov al,00h
    out dx,al

    mov al,24h
    out dx,al
    jmp do_nothing

case_2:
    cmp al,02h
    jnz case_3
    mov dx,0602h
    mov al,00h
    out dx,al

    mov al,48h
    out dx,al
    jmp do_nothing

case_3:
    cmp al,03h
    jnz __default

    mov dx,0602h
    mov al,00h
    out dx,al

    mov al,90h
    out dx,al
    jmp do_nothing

__default:

do_nothing:
    jmp loop_body

ret

START:
    ;save registers
    push ax
    push cx
    push dx
    push bx
    push si
    push di
    pushf

```

```
    mov ax,DATA
    mov ds, ax

    call main

    popf
    pop di
    pop si
    pop bx
    pop dx
    pop cx
    pop ax

    mov ax,4c00h
    int 21h

CODE    ends

end    START
```