

工作日志

| | | | |
|--|-------------|----|------------|
| 标题 | 0909 工作日志 | | |
| 姓名 | 汪能志 | 学号 | U201713082 |
| 日期 | 2020年9月 9 日 | 天气 | 晴 |
| 日记撰写时间 | | | |
| 内容 | | | |
| <p>一、实验内容</p> <p>基于 PIC16 单片机的闪灯实验。通过调整延时子程序中的相关参数，可以调整 LED 的闪烁频率。</p> <p>二、实验基本思路</p> <p>对于单片机开发板来说，闪灯实验是最基本的程序。实现闪灯的基本思路如下：</p> <p>初始化引脚</p> <p>循环</p> <p> 反转引脚电平</p> <p> 延时</p> <p>循环结束</p> <p>后面将以此说明这三步的原理和实现方案。</p> <p>三、引脚初始化</p> <p>首先需要参考 I/O 引脚的电路图进行初始化操作。单片机的引脚电路图如下：</p> <p>在初始化时，需要对 ANSEL（模拟/数字选择），TRIS（数据方向选择）进行设置，将引脚设置为数字输出（ANSEL=0;TRIS=0）。</p> <pre> ; SETUP ; init ports, select port C, set as Digital Output ; select port C BANKSEL PORTC CLR PORTC </pre> | | | |

```

; select digital signal
BANKSEL    ANSEL
CLR        ANSEL
MOVLW      B'00000000'
MOVWF      ANSEL

```

```

; select output
BANKSEL    TRISC
MOVLW      B'00000000'
MOVWF      TRISC

```

```

; init output
MOVLW      B'00000000'
MOVWF      PORTC

```

四、主循环

通过对 PORTC 和 0xFF 进行异或运算来实现明暗切换，通过延时子程序来实现闪烁变化。由于工作寄存器在每次执行运算时都会被改写，因此在进行异或运算前，需要先将工作寄存器置为 0xFF。主循环需要使用 4 个指令周期。

```

; LOOP
; main loop
    LOOP
        ; W is used as XOR
        MOVLW      B'11111111'
        ; XOR port c & reg f
        XORWF      PORTC, 1
        ; delay
        CALL        DELAY
    GOTO    LOOP

```

五、延时

延时的基本思路是通过进行给定次数的循环占用一定的处理器时间，以此达到定时的目的。其进行的操作类似：

```
for(int i = N; i > 0; i--) {}
```

由于单片机中最多只能支持 8bit 的数据，为了达到更长时间的延时，可以通过嵌套循环的方式提高延迟时间。

由于 DEFCFSZ 指令在遇到 0 时会跳过下一条指令，因此是进行循环体中的主要指令。为了达到秒级的延时，需要三级嵌套循环：

```

DELAY
; Instruction Num = 3 * N0 * N1 * N0 + 4 * N1 * N0 + 4 * N0 + 5
MOVLW      0x01
MOVWF      N0
DELAY_LOOP_0

```

```
MOVLW    0x01
MOVWF    N1
DELAY_LOOP_1

    MOVLW    0x01
    MOVWF    N2
    DELAY_LOOP_2
        DECFSZ    N2
        GOTO     DELAY_LOOP_2

    DECFSZ    N1
    GOTO     DELAY_LOOP_1

    DECFSZ    N0
    GOTO     DELAY_LOOP_0

RETURN
```

通过查阅指令表和使用 IDE 软件调试功能中的跑表，可以得到这一段的 DELAY 子程序的指令周期数为： $3 * n0 * n1 * n0 + 4 * n1 * n0 + 4 * n0 + 5$

开发板的指令周期为 1us，因此可以计算 DELAY 子程序的延时。需要注意，主循环 LOOP 中也存在指令语句，需要占用时间，在计算时不可忽略。

六、实验中遇到的问题

实验中遇到的最大问题和工作寄存器相关。由于 PIC 单片机中，所有运算都有一字节数据来自工作寄存器，因此需要格外注意它的取值。最早在主循环中没有注意这一点，因此发现可以正常输出信号的引脚和设置的延时值相关。在通过 IDE 进行调试后，发现了问题的来源，并将其解决。