

工作日志

标题	0911 工作日志		
姓名	汪能志	学号	U201713082
日期	2020年9月 11 日	天气	晴
日记撰写时间			
内容	<p>一、调试故障的解决</p> <p>在昨天的调试过程中，选择在 bootloader 模式下进行调试时，会周期性触发程序运行时间过长的问题。同时还伴有内存重置，程序从头开始运行等问题。这和看门狗起效的现象相同，因此原因可能是在 bootloader 模式下调试是触发了看门狗。在 default 下调试时无此问题。</p> <p>由于进行定时器查询的程序占用了四个指令周期 (banksel, 查询, 跳转)，因此在调试时，定时器的输出周期表现为：T+1,T+1,T+1,T-3。这和跑表的结果一致。</p> <pre> Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499709 (499.709 ms) Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499713 (499.713 ms) Target halted. Stopwatch cycle count = 499709 (499.709 ms) </pre> <p>二、定时器查询闪灯的 c 语言重置</p> <p>在进行定时器中断实验之前，为了熟悉通过 C 语言对 PIC 单片机进行编程，我先用 C 语言重写了定时器查询闪灯的程序，并使用跑表进行了测试。</p> <pre> void setup(void) { /** * init PORTC */ PORTC = 0x00; ANSELC = 0x00; LATC = 0x00; TRISC = 0x00; /** * init TMRO */ TMR0H = 0xF3; TMR0L = 0x00; TMR0IF = 0x00; TOCON0 = 0xCF; TOCON1 = 0x47; </pre>		

```
}

void loop(void)
{

    if (PIR0bits.TMR0IF == 0)
    {
    }
    else
    {
        PIR0bits.TMR0IF = 0;
        PORTC = ~PORTC;
    }
}

void main(void)
{
    setup();
    while (1)
    {
        loop();
    }
    return;
}
```

```
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
```

三、其他问题

由于在实验中使用了定时器的预分频和后分频，因此调节时间的分辨率较差，不能做到恰好 500ms 的定时。这个问题可能的解决方法为：使用 16bit 定时器并且使用软件补偿误差值。在以后的实验中，可能会尝试解决这个问题。