

# 苏州大学电子信息学院

## 设计性实验报告

### 定时器实验

实验者姓名：龚烨

合作者姓名：

专业：通信工程

班级：2

学号：2128410206

指导老师：朱哲辰

实验日期：2023.5.8

# 目录

一 设计任务.....	*
二 方案选择与设计.....	*
三 软、硬件原理与实现.....	*
四 测试要求与设备.....	*
五 结果记录与讨论 .....	*
六 存在问题与改进对策.....	*
参考文献.....	*

## 一、设计任务

1. 在 PROTEUS 环境中,设计单片机外接 8 个 LED。调整晶振频率为 6 MHz。在 P3.4 处连接一个按钮开关,开关另一端接地。
2. 编写控制程序:利用定时器控制 LED 点亮。三种情况需使用不同的工作模式。
  - (1)使用定时器使得全部 LED 以 100 ms 的间隔点亮与熄灭。
  - (2)使用定时器使得全部 LED 以 10 ms 的间隔点亮与熄灭。
  - (3)使用定时器使得全部 LED 以 0.1 ms 的间隔点亮与熄灭。
3. 编写控制程序:利用计数器控制 LED 点亮。
  - (1)全部 LED 默认熄灭状态,按钮按下 5 次后全部 LED 改变点亮的状态。
  - (2)全部 LED 默认跑马灯模式,按钮按下 5 次后全部 LED 灯闪烁一次,随后恢复跑马灯的状态。

## 二、方案选择与设计

- 1.理论推导出应通过实验得出的结果。
- 2.利用 uVision Keil 编写代码,利用 Proteus8 搭建电路。
- 3.利用课上提供的部分代码,根据不同要求对源代码进行改进,利用定时器控制 LED 点亮。

- (1)使用定时器使得全部 LED 以 100 ms 的间隔点亮与熄灭。

考虑到实验要求用三种不同的定时器工作模式来实现,因此需要根据每个工作模式的最大计数值来确定工作模式的分配。由于工作模式 1 拥有最大的最大计数值,所以对 100ms 可以使用方式 1 来定时。

- (2)使用定时器使得全部 LED 以 10 ms 的间隔点亮与熄灭。

设计用方式 0 来定时 10ms 时间。

- (3)使用定时器使得全部 LED 以 0.1 ms 的间隔点亮与熄灭。

设计用方式 2 来定时 0.1ms 时间。

- 4.利用书本上的源代码和课上讲过的部分代码,根据不同要求对源代码进行改进,利用计数器控制 LED 点亮。

- (1)全部 LED 默认熄灭状态,按钮按下 5 次后全部 LED 改变点亮的状态。

考虑到实验要求用到计数器的计数功能,又在 P3.4 处连接了一个按钮作为外部输入信号,需要按下 5 次。则选用可自动装载计数初值且最大计数值为 256 的方式 2 作为计数器工作模式。按下 5 次后进入中断服务程序,控制全部 LED 灯的亮灭状态。

- (2)全部 LED 默认跑马灯模式,按钮按下 5 次后全部 LED 灯闪烁一次,随后恢复跑马灯的状态。

该要求同(1)考虑,同样利用方式 2 作为计数器的工作模式。仅需修改按下 5 次后进入的中断服务程序为全部 LED 灯的闪烁程序,随后退出至主函数继续跑马灯的流动,即可完成要求。

三、软、硬件原理与实现

硬件部分：

在 P3.4 引脚外接一个按钮，用来作为计数器 0 的外部信号输入源，每按下一次，计数值就会从计数初值往上增加 1，直到达到最大计数值，产生中断溢出标志，进入中断，执行中断服务程序。

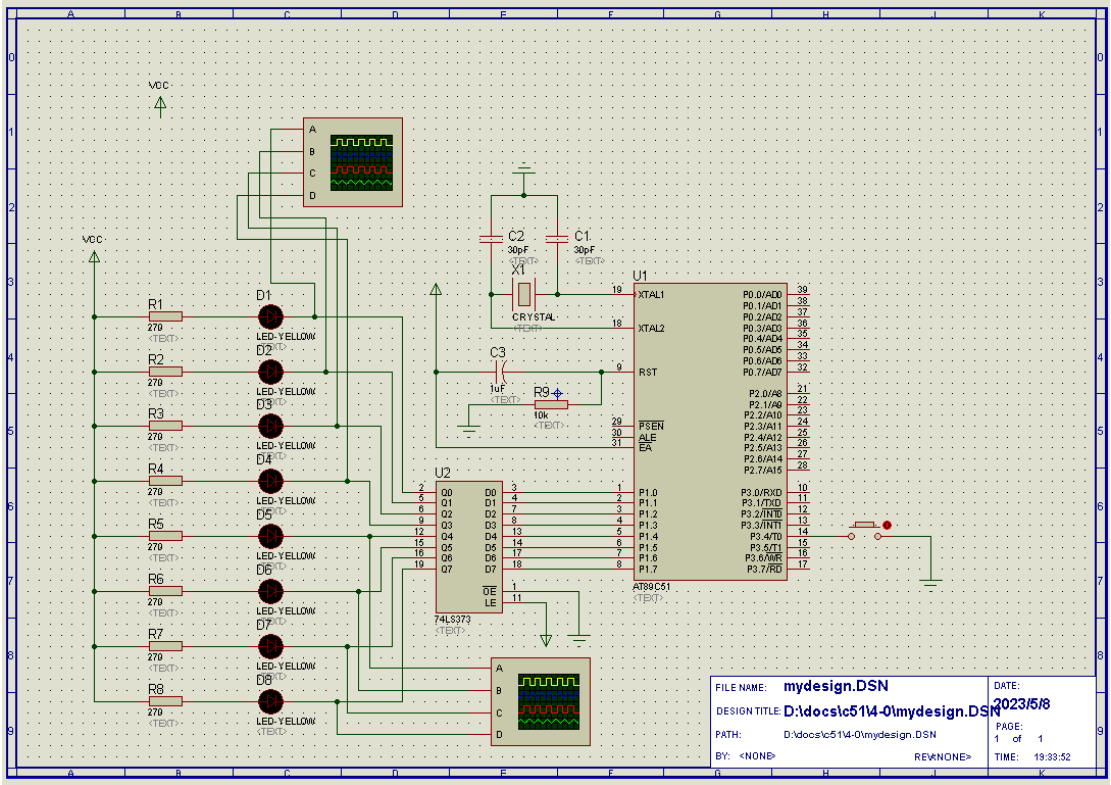


图 1 电路图

软件部分：

1. 跑马灯：对书中源代码进行分析，首先定义初始位置，再给累加器 A 赋初值 0FEH，由于所用灯为共阳极 LED 灯，因此 0FEH 代表仅有第一个灯亮。随后进入循环体内，将 A 中的值送入 P1 口输出，然后对 A 进行循环移位，使下一次亮灯位置左移一位，然后进入延时函数，保持灯的点亮状态，然后再次循环。

延时函数：DELAY 函数中，通过三层循环嵌套，对寄存器 R5、R6、R7 进行设置，让程序进行空指令的循环，从而达到延迟的目的。

定时器初始化及中断服务程序：定时器具有两种进入中断的方式（1）查询方式（2）中断方式。查询方式为查询中断溢出标志位是否置 1，若置 1 则进入中断服务程序。中断方式为在程序开头设置中断入口，待中断到来自动执行中断服务程序。在主函数中对中断的定时模式、工作方式、计数初值和控制寄存器进行设置。当定时模式为定时器时，在经过特定的时间后进入中断，而定时模式为计数器时，在接收到特定次数的外部信号输入后则会进入中断。在进入中断服务程序后需要保护现场，防止在中断过程中改变寄存器的值，造成程序错误。中

断服务程序运行完后在末尾加上中断返回指令，返回上一级程序，如主函数或低优先级的中断，继续程序的运行。

#### ①利用定时器控制 LED:

(1) 使用定时器使得全部 LED 以 100ms 的间隔点亮与熄灭。

TMOD 应设置定时器 0 为定时模式、方式 1，即为 01H。

对计数初值进行理论推导：因为方式 1 最大计数值为 65536，晶振 6MHz。

$$\text{初值 } X = 65536 - (t \times f_{osc}) / 12 = 65536 - (100 \times 6 \times 10^3) / 12 = 15536 = 3CB0H$$

所以将计数初值高位 TH0 设为 03CH，低位 TL0 设为 0B0H，即可达到 100ms 定时的结果。设置完工作模式、方式及初值后，将 TR0 置 1，启动定时器 0，开始全部 LED 灯以 100ms 为间隔的闪烁。

(2) 使用定时器使得全部 LED 以 10ms 的间隔点亮与熄灭。

TMOD 应设置定时器 0 为定时模式、方式 0，即为 00H。

对计数初值进行理论推导：因为方式 0 最大计数值为 8192，晶振 6MHz。

$$\text{初值 } X = 8192 - (t \times f_{osc}) / 12 = 8192 - (10 \times 6 \times 10^3) / 12 = 3192$$

在方式 0 中，TL0 只用低 5 位，高 3 位不用，因此数据的有效位必须装入 TL0 的低 5 位和 TH0 的 8 位中。3192 = 1100011 11000B，所以计数初值高位 TH0 设为 63H，低位 TL0 设为 18H，即可达到 10ms 定时的结果。设置完工作模式、方式及初值后，将 TR0 置 1，启动定时器 0，开始全部 LED 灯以 10ms 为间隔的闪烁

(3) 使用定时器使得全部 LED 以 0.1ms 的间隔点亮与熄灭。

TMOD 应设置定时器 0 为定时模式、方式 2，即为 02H。

对计数初值进行理论推导：因为方式 2 最大计数值为 256，晶振 6MHz。

初值。在方式 2 中，TL0 中为用于计算的计数初值，而 TH0 中则放置用于自动填装进 TL0 的计数初值。所以计数初值高位 TH0 与低位 TL0 都设为 CEH，即可达到 0.1ms 定时的结果。设置完工作模式、方式及初值后，将 TR0 置 1，启动定时器 0，开始全部 LED 灯以 0.1ms 为间隔的闪烁。

## 四、测试要求与设备

要求： 1.在 Keil  $\mu$  vision 中完成程序的编写编译；

2.在 Proteus 中完成整体电路的连接布线和仿真；

3.观察跑马灯的点亮时间和移动方式。

硬件设备：计算机一台

软件设备：windows 操作系统、Proteus 8 Professional、uVision 软件以及 51 系统开发环境。

## 五、结果记录与讨论

成功实现 LED 灯的四种点亮方式。

#### ①利用定时器控制 LED:

(1) 使用定时器使得全部 LED 以 100ms 的间隔点亮与熄灭。

根据代码运行实验，在每个 LED 灯右端连接示波器的一个通道，通过示波器对灯的点亮情况进行观察。100ms 为间隔亮灭可通过示波器观察得：

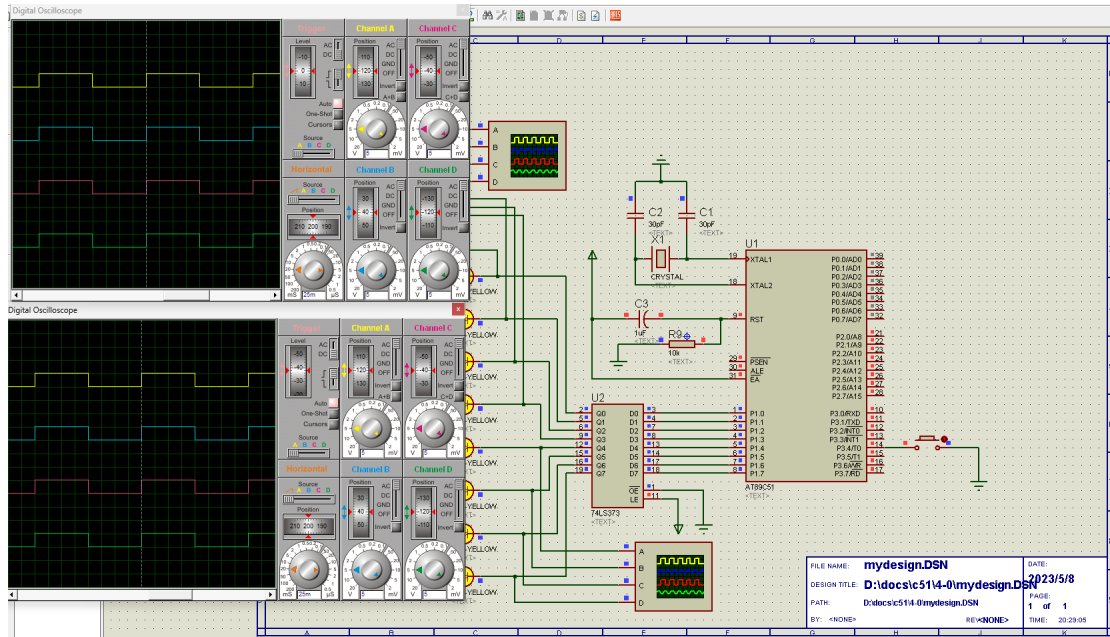


图 2 全部 LED 灯 100ms 亮灭情况

通过示波器可以在程序运行时全部 LED 灯引脚电平在高一低之间转换，亮或灭的时间为 100ms，达到了以 100ms 的间隔点亮与熄灭的要求，完成了实验目的。

代码如下：

```

ORG 0000H
LJMP MAIN
ORG 000BH
LJMP INT_T0
ORG 0100H
MAIN:
MOV SP, #30H
MOV A, #0FFH
MOV P1, A
ACALL INIT
SJMP $
INIT:
SETB EA
SETB ET0
MOV TMOD, #01
MOV TH0, #03CH
MOV TL0, #0B0H
SETB TR0
RET
INT_T0:
CPL A
MOV P1, A
MOV TH0, #3CH
MOV TL0, #0B0H
RETI
END

```

(2) 使用定时器使得全部 LED 以 10ms 的间隔点亮与熄灭。

根据代码运行实验，在每个 LED 灯右端连接示波器的一个通道，通过示波器对灯的点亮情况进行观察。则全部 LED 以 10ms 为间隔亮灭可通过示波器观察得：

图 5.2 全部 LED 灯 10ms 亮灭情况

通过示波器可以显示在程序运行时全部 LED 灯引脚电平在高一低之间转换，8 个 LED 灯同时亮灭。示波器的每一格代表了 5ms，在一次亮或灭的过程中，波形占据 2 格，则亮或灭的时间为 10ms，达到了以 10ms 的间隔点亮与熄灭的要求，完成了实验目的。

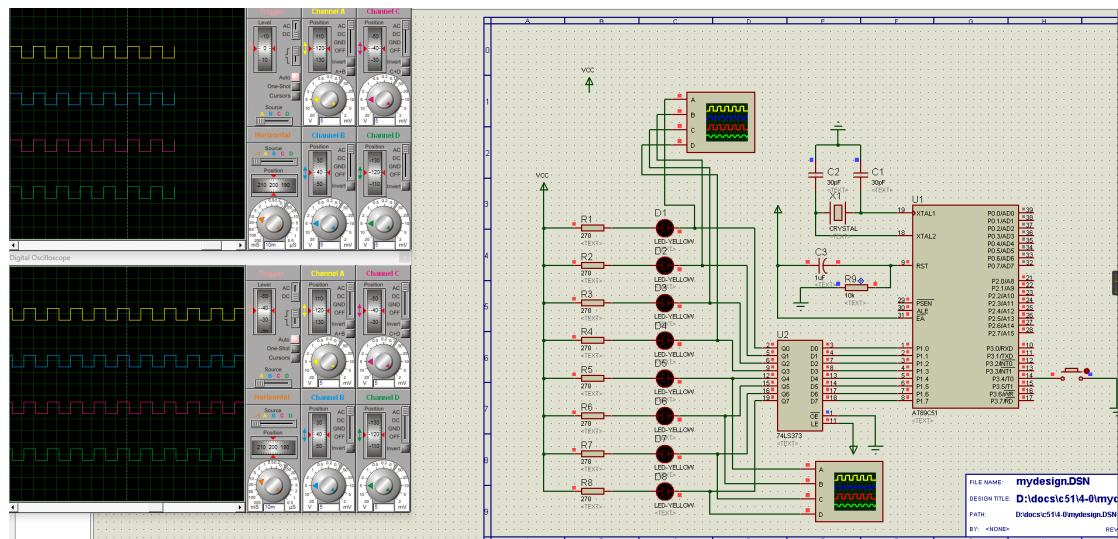


图 3 全部 LED 灯 10ms 亮灭情况

代码如下：

```
ORG 0000H
LJMP MAIN
ORG 0BH
LJMP INT_T0
ORG 0100H
MAIN:
MOV SP, #30H
MOV A, #0FFH
MOV P1, A
ACALL INIT
SJMP $
INIT:
SETB EA
SETB ET0

MOV TMOD, #00H
MOV TH0, #063H
MOV TL0, #018H
SETB TR0
RET
```

```

INT_T0:    CPL A
           MOV P1, A
           MOV TH0, #063H
           MOV TL0, #018H
           RETI
           END

```

(3) 使用定时器使得全部 LED 以 0.1ms 的间隔点亮与熄灭。

根据代码运行实验，在每个 LED 灯右端连接示波器的一个通道，通过示波器对灯的点亮情况进行观察。0.1ms 为间隔亮灭可通过示波器观察得：

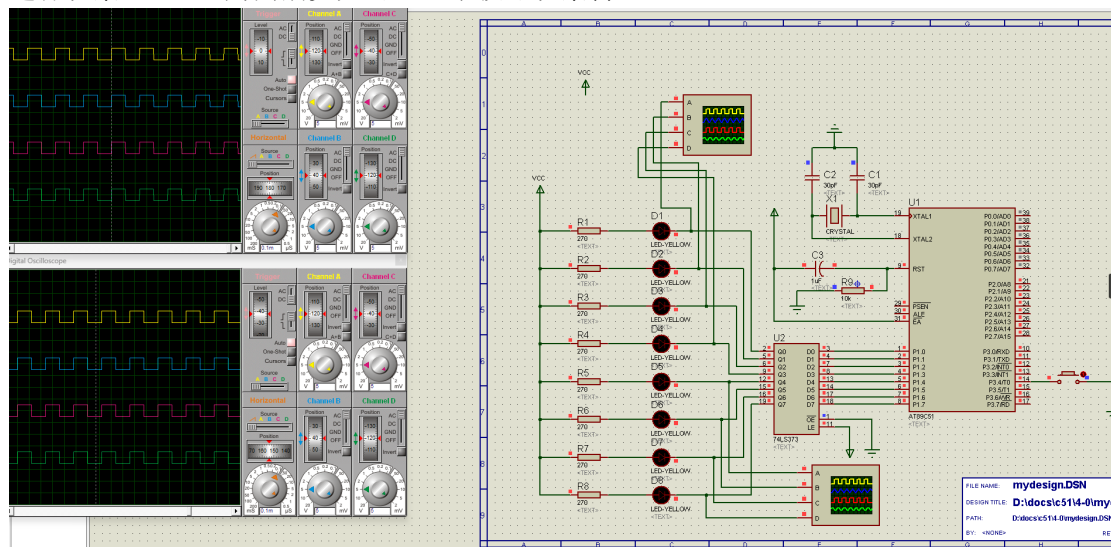


图 4 全部 LED 灯 0.1ms 亮灭情况

代码如下：

```

ORG 0000H
LJMP MAIN
ORG 0BH
LJMP INT_T0
ORG 0100H
MAIN:    MOV SP, #30H
         MOV A, #0FFH
         MOV P1, A
         ACALL INIT
         SJMP $
INIT:    SETB EA
         SETB ET0
         MOV TMOD, #02H
         MOV TH0, #0CEH
         MOV TL0, #0CEH
         SETB TR0
         RET
INT_T0:  CPL A

```



```

MOV P1, A
MOV TH0, #0CEH
MOV TL0, #0CEH
RETI
END

```

②利用计数器控制 LED:



图 5 计数器控制 LED 改变点亮的状态

在程序运行时，全部 LED 灯引脚电平在开始时都为高电平，在按钮按下第 5 次的瞬间，所有 LED 灯引脚都变为低电平，灯点亮。在等待程序稳定后，再次按下 5 次按钮，全部 LED 灯引脚电平都从低电平上拉至高电平，灯熄灭。达到了按钮按下 5 次后全部 LED 改变点亮状态的要求，完成了实验目的。

使用的代码如下：

```

ORG 0000H
LJMP MAIN
ORG 0BH
LJMP INT_T0
ORG 000BH
AJMP INT_T0
MAIN:
MOV SP, #40H
MOV TMOD, #06H
MOV TH0, #0FBH
MOV TL0, #0FBH
MOV A, #0FFH
MOV P1, A
SETB ET0
SETB EA

```

```

LOOP:      SETB TR0
           SJMP LOOP
ORG 0200H
INT_T0:    CPL A
           MOV P1, A
           RETI
           END

```

(2) 全部 LED 默认跑马灯模式，按钮按下 5 次后全部 LED 灯闪烁一次，随后恢复跑马灯灯的状态。

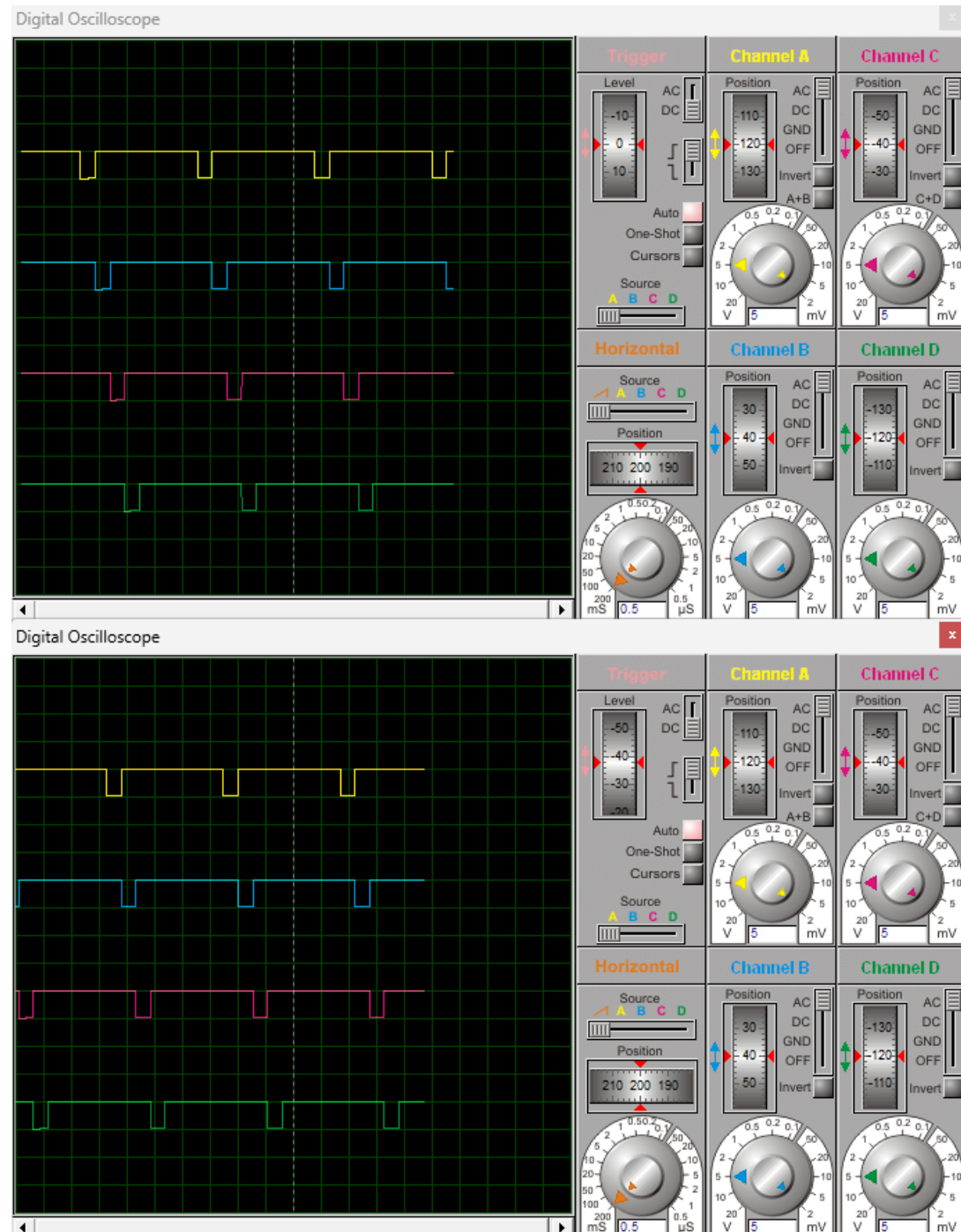


图 6 计数器控制 LED 跑马灯

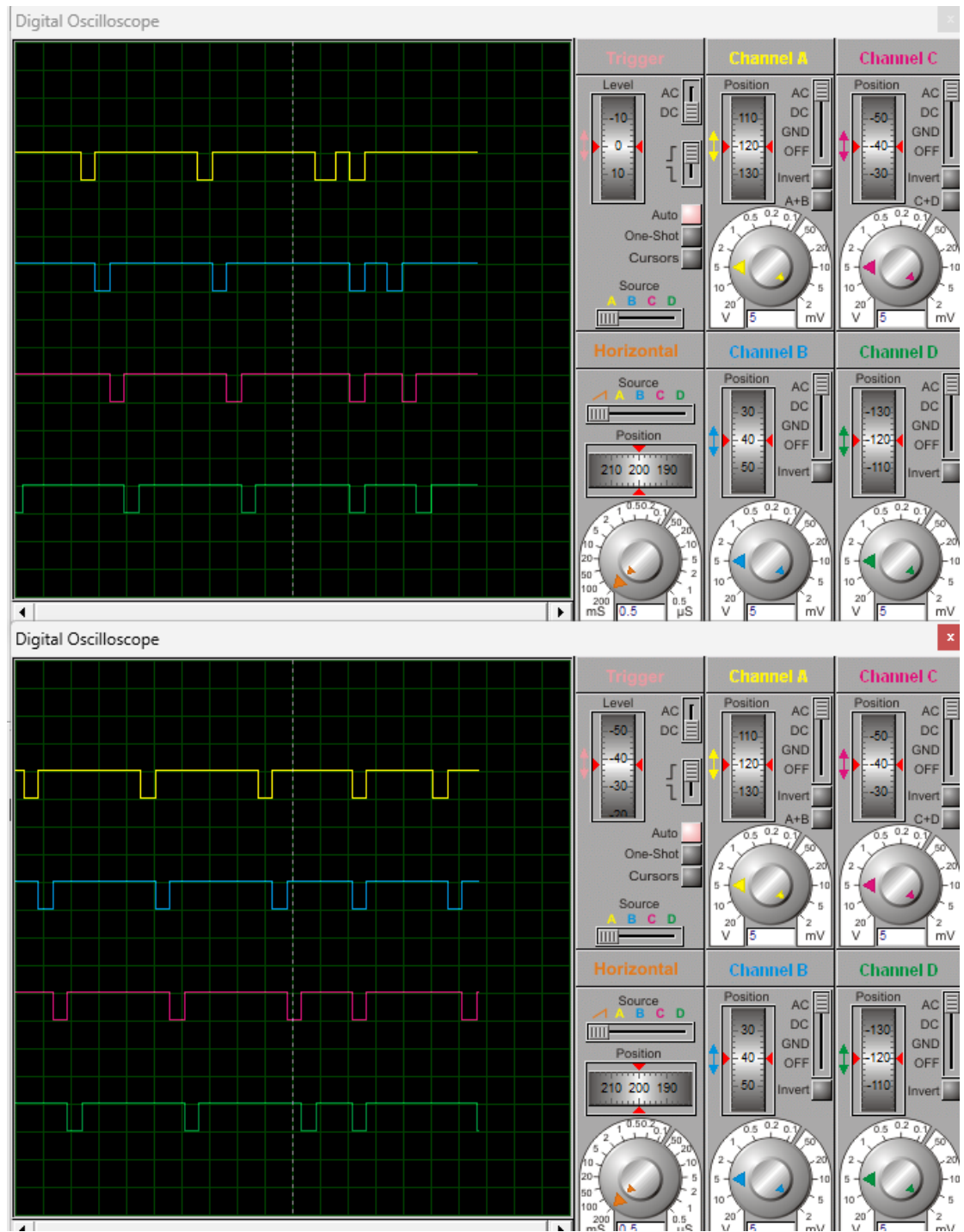


图 7 计数器控制 LED 闪烁一次

运行实验，在每个 LED 灯右端连接示波器的一个通道，在 P3.4 计数器外部信号输入引脚接入一按钮，通过示波器对灯的点亮情况进行观察，再手动按下 5 次按钮，对灯的点亮情况再次进行观察。全部 LED 的状态可通过示波器观察得到，达到了按钮按下 5 次后全部 LED 灯闪烁一次，随后恢复跑马灯的状态的要求，完成了实验目的。

代码如下所示：

```
ORG 0000H
AJMP MAIN
ORG 0BH
```

```

                                AJMP INT_T0
                                ORG 0100H
MAIN:                            MOV SP, #40H
                                MOV TMOD, #06H
                                MOV TH0, #0FBH
                                MOV TL0, #0FBH
                                MOV A, #0FEH
                                MOV P1, A
                                SETB ET0
                                SETB EA
                                SETB TR0
LOOP1:                          MOV P1, A
                                RL A
                                LCALL DELAY
                                AJMP LOOP1
                                ORG 0200H
INT_T0:                          PUSH ACC
                                PUSH PSW
                                PUSH 07H
                                PUSH 06H
                                MOV R5, #03H
                                MOV A, #00H
                                LCALL DELAY
LOOP2:                          CPL A
                                MOV P1, A
                                LCALL DELAY
                                DJNZ R5, LOOP2
                                POP 06H
                                POP 07H
                                POP PSW
                                POP ACC
                                RETI
DELAY:                          MOV R6, #00H
DELAY1:                          MOV R7, #00H
DELAY2:                          DJNZ R7, DELAY2
                                DJNZ R6, DELAY1
                                RET
                                END

```

## 五、结果记录与讨论

在选用电平触发方式开启外部中断时，若中断源开关一直导通，会使 LED 灯一直处于被点亮和熄灭状态之间来回切换，而最终 LED 会锁定在开关断开瞬间的状态，不论是点亮或熄灭。这是因为电平触发方式在外部中断源为低电平时始终有效，会不断发出中断请求，使得 LED 不断反转状态，直到引脚重新呈现高电平为止。而当最后断开开关时，LED 灯才会锁定在最后一个中断请求变换的状态下。在使用边沿触发方式时则不会出现这个问题，这是因为在开关接通时，该引脚只会从高电平跳变到低电平一次，如果开关不松开，该引脚始终处于低电平，不会产生新的下降沿，进而不会影响 LED 灯的状态。

在按下 SW1 之后迅速按下 SW2，LED 首先从跑马灯转换为奇偶间隔闪烁，闪烁 5 次后转换

为全体闪烁，全体闪烁 5 次后，回到跑马灯的主程序中，继续相继点亮。奇偶间隔闪烁和全体闪烁间隔均为 1s。

#### 参考文献

陈蕾，邓晶，仲兴荣.单片机原理与接口技术 [M]，机械工业出版社，2012

侯玉宝等 .基于 Proteus 的 51 系列单片机设计与仿真 [M]，电子工业出版社 2009.