**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。

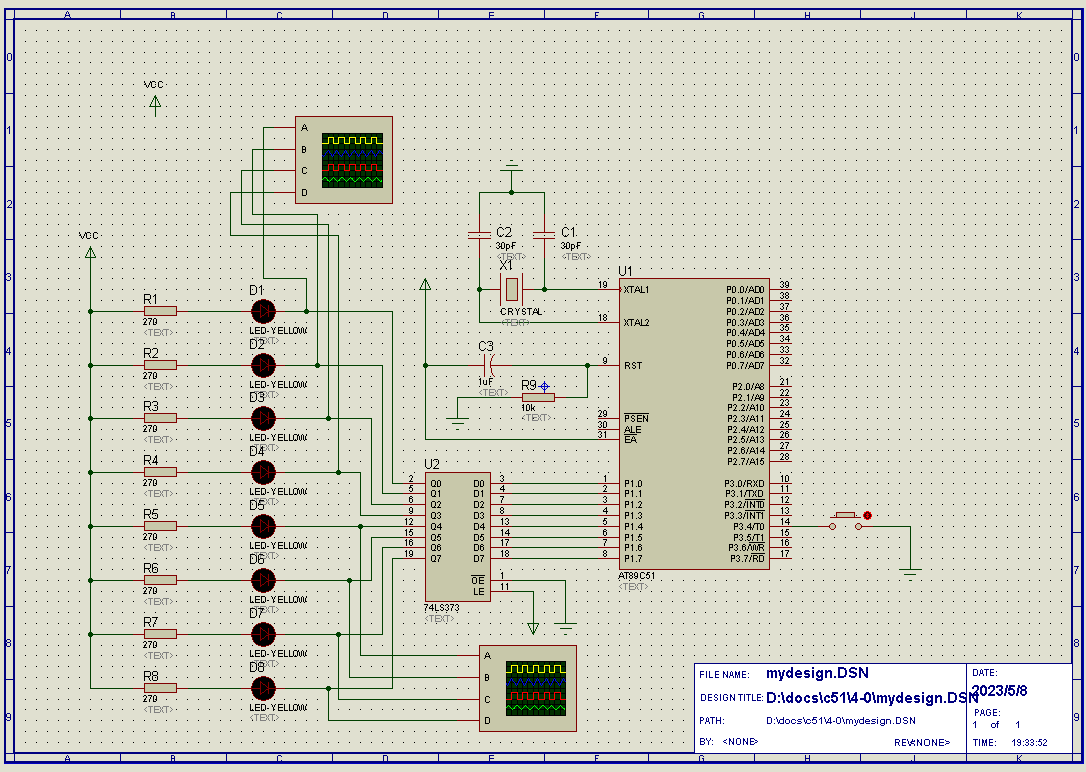


图1 电路图

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。

初值 

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1最大计数值为8192，晶振6MHz。

初值。

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 μvision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：

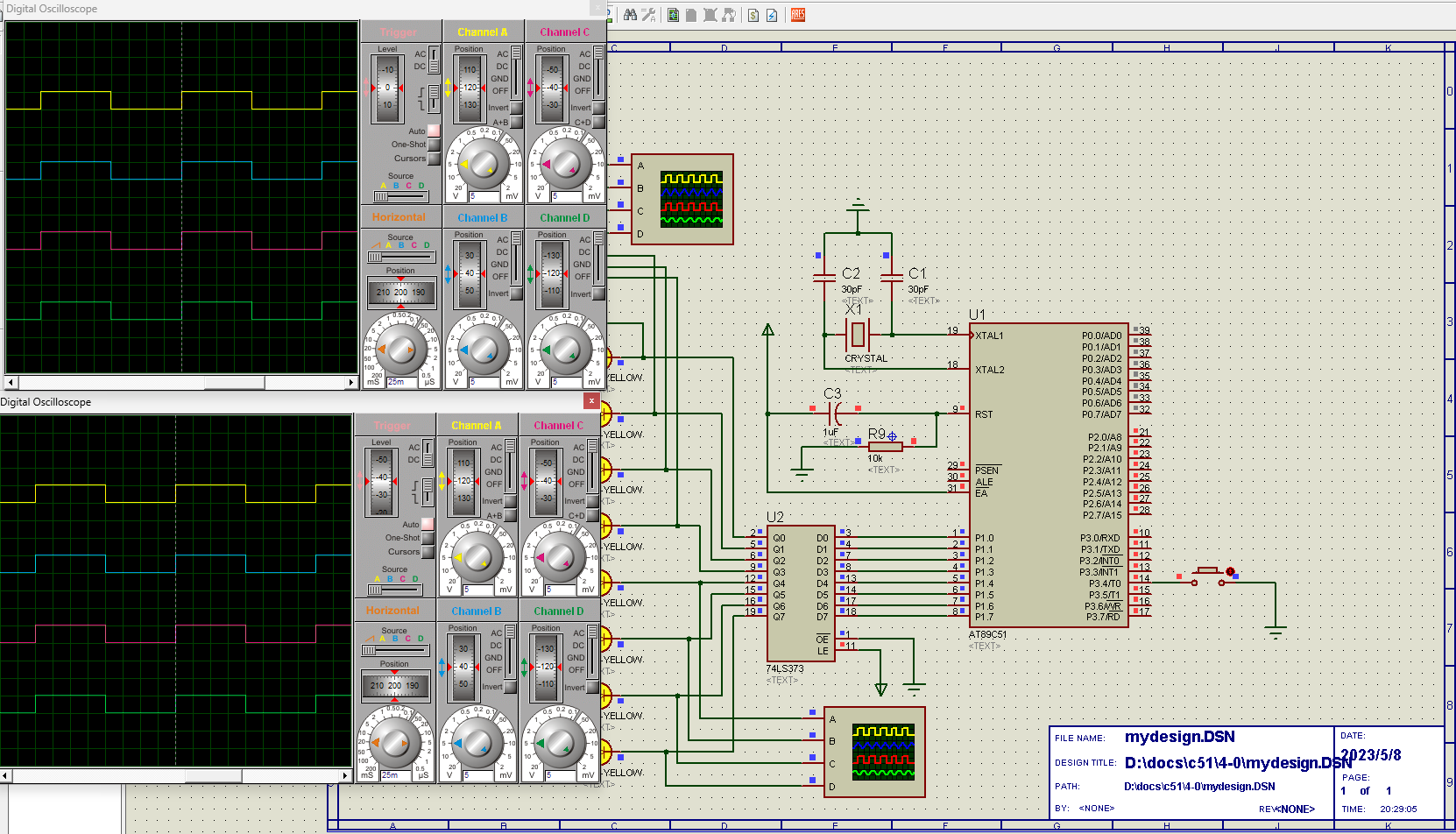


图2 全部LED灯100ms亮灭情况

通过示波器可以在程序运行时全部LED灯引脚电平在高—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—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。

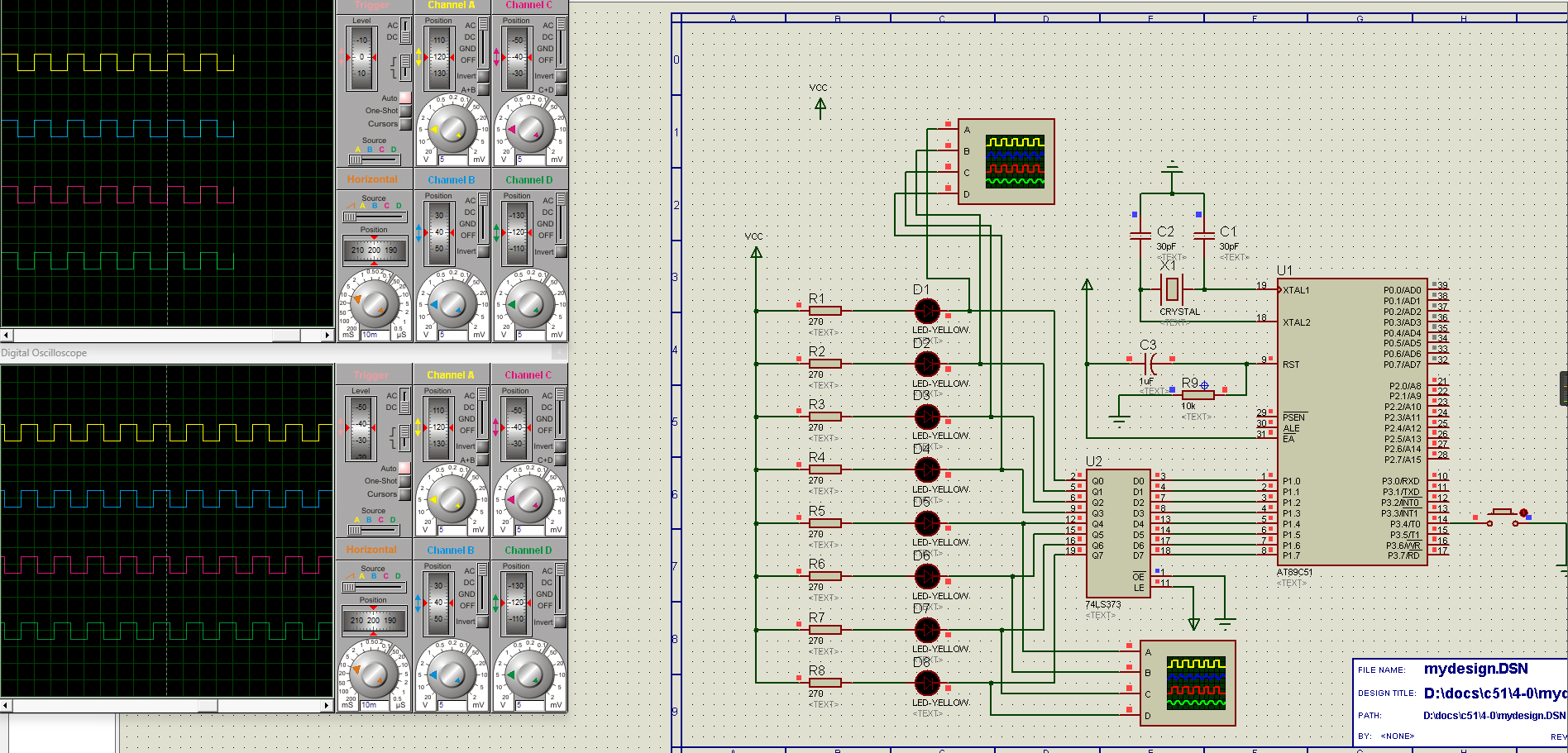


图3全部LED灯10ms亮灭情况

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：

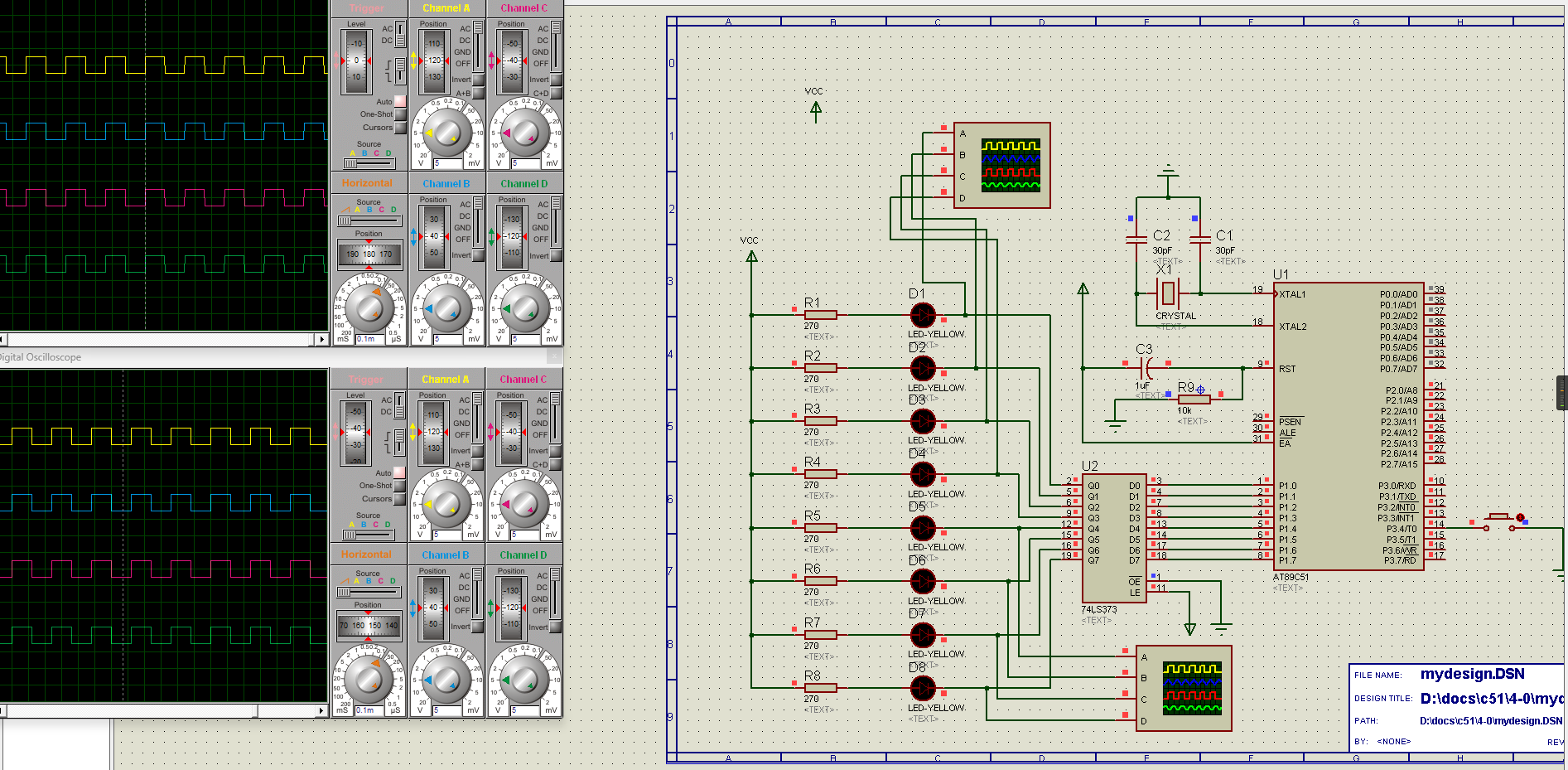


图4 全部LED灯0.1ms亮灭情况

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

            SETB TR0

LOOP:       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。

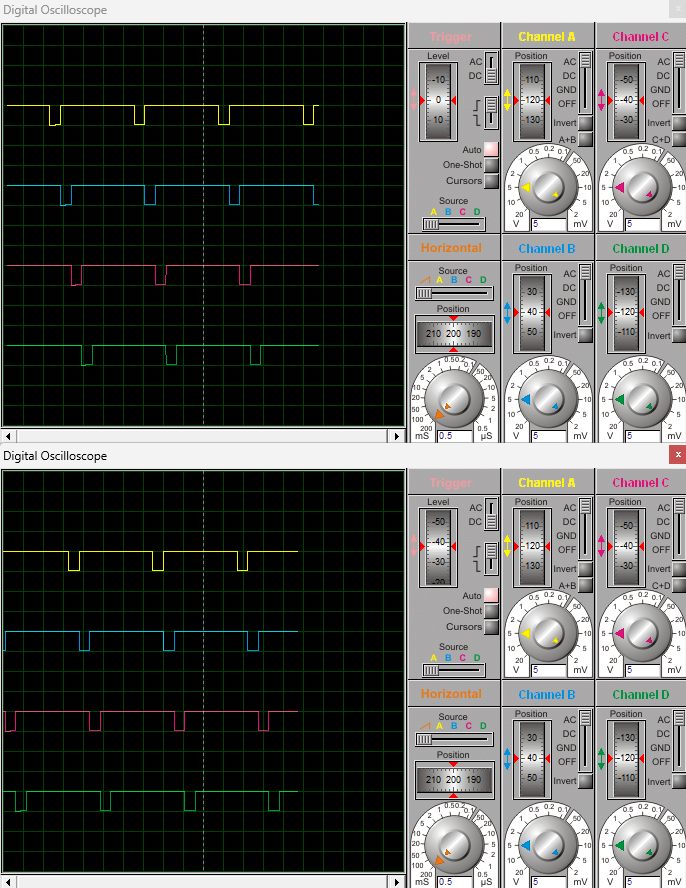


图6 计数器控制LED 跑马灯

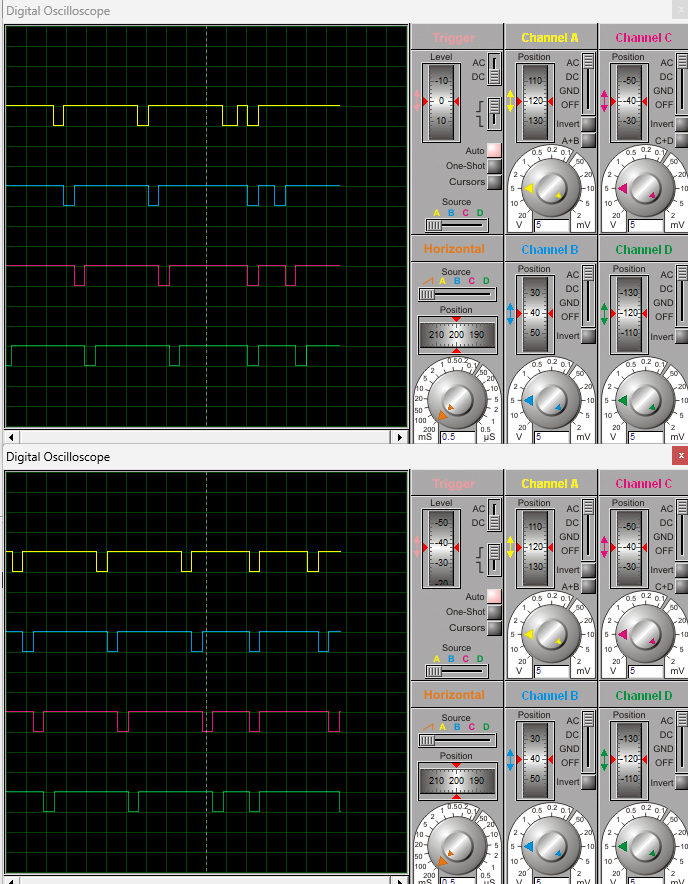


图7 计数器控制LED 闪烁一次

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.