# 定时/计数器实验

一、实验目的

掌握单片机定时/计数器的使用方法，包括初始化方法和中断服务程序的编写方法。

二、实验设备及器件

个人计算机1台，装载了Keil C51集成开发环境软件。DP-51PRO.NET单片机仿真器、编程器、实验仪三合一综合开发平台1台。

三、实验内容

（1） 8031内部定时计数器T0，按计数器模式和方式1工作，对P3.4（T0）引脚进行计数。将其数值按二进制数在P1口驱动LED灯上显示出来。同时在8位8段数码管上显示出来。

（2） 用CPU内部定时器中断方式计时，实现每1秒钟控制P1.0输出状态发生一次反转，P1.0接发光二极管LED8。

（3） 用P1口控制8个发光二极管LED1～LED8显示“跑马灯”效果，每按一次中断按钮，进入一种循环显示模式（共四种模式）。循环显示的延时要求用定时器实现。实验原理图见图3.6-1。

\*（4） 设计一个电子钟。设计一个秒表。

\*（5） 编写程序从P1.0口输出1kHz的方波，要求利用T0定时，工作在方式1；从P1.1口输出2kHz的方波，要求利用T1定时，工作在方式2。

原理图：



程序框图：

实验1.



实验2.



实验3.



实验4.



程序源码：

实验1.

;计数器模式下LED显示按键次数中断程序

;编写人：高洪伟

;编写日期2019-4-6

;修改日期2019-4-10

;30H为定时器按键计数存储单元

LATCH1 BIT P2.2

LATCH2 BIT P2.3

ORG 0000H

LJMP START

ORG 000BH

LJMP T\_0

START:

MOV TMOD, #05H ;T0中断初始化

MOV TCON, #11H

MOV IE, #82H

MOV TL0, #0FFH

MOV TH0, #0FFH ;按一次溢出一次

MOV DPTR, #TABLE

MOV SP, #60H ;分配堆栈

MOV 30H, #00H ;按键计数存储单元

CALL STORE ;先存储一次段码

SCAN:

MOV A, 30H

XRL A, #0FFH

MOV P1, A

MOV A, 22H

MOV P0, A

SETB LATCH1

CLR LATCH1

MOV P0, #11111110B

SETB LATCH2

CLR LATCH2

CALL DELAY

MOV A, 21H

MOV P0, A

SETB LATCH1

CLR LATCH1

MOV P0, #11111101B

SETB LATCH2

CLR LATCH2

CALL DELAY

MOV A, 20H

MOV P0, A

SETB LATCH1

CLR LATCH1

MOV P0, #11111011B

SETB LATCH2

CLR LATCH2

CALL DELAY

LJMP SCAN

DELAY: MOV R6, #4 ;扫描延时

D3: MOV R7, #248

DJNZ R7, $

DJNZ R6, D3

RET

T\_0:

CLR EA

PUSH PSW

PUSH Acc

SETB EA

MOV TL0, #0FFH

MOV TH0, #0FFH

INC 30H ;按键次数

CALL STORE ;转换为段码

CLR EA

POP Acc

POP PSW

SETB EA

RETI

;段码转换程序

STORE:

MOV A, 30H

MOV B, #100

DIV AB

MOVC A, @A+DPTR

MOV 22H, A

MOV A, B

MOV B, #10

DIV AB

MOVC A, @A+DPTR

MOV 21H, A

MOV A, B

MOVC A, @A+DPTR

MOV 20H,A

RET

TABLE:DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH ;共阴字码表

END

实验二：

;定时器中断反转P1.0程序

;编写人：高洪伟

;编写日期2019-4-6

;修改日期2019-4-10

;30H存放50ms次数，满20清零

LATCH1 BIT P2.2

LATCH2 BIT P2.3

ORG 0000H

LJMP START

ORG 000BH

LJMP T\_0

START:

MOV TMOD, #01H ;T0中断初始化

MOV TCON, #11H

MOV IE, #82H

MOV TL0, #0B0H ;定时50ms

MOV TH0, #3CH ;按一次溢出一次

MOV SP, #60H ;分配堆栈

MOV 30H, #00H ;50ms次数，满20清零

SCAN: JMP SCAN

T\_0:

CLR EA

PUSH PSW

PUSH Acc

SETB EA

MOV TL0, #0B0H

MOV TH0, #3CH

INC 30H

MOV A, 30H

CJNE A, #14H, EXIT

MOV 30H, #00H

CPL P1.0

EXIT:

CLR EA

POP Acc

POP PSW

SETB EA

RETI

END

实验三：

;定时器中断流水灯

;编写人：高洪伟

;编写日期2019-4-6

;修改日期2019-4-10

LATCH1 BIT P2.2

LATCH2 BIT P2.3

NEWFLAG BIT 7FH

ORG 0000H

LJMP START

ORG 0003H

LJMP INT\_0

ORG 000BH

LJMP T\_0

START: SETB EA

SETB EX0

SETB PX0

SETB IT0

MOV TMOD, #01H ;T0中断初始化

MOV TCON, #13H

MOV IE, #83H

MOV TL0, #0B0H ;定时50ms

MOV TH0, #3CH ;按一次溢出一次

MOV SP, #60H ;分配堆栈

MOV 30H, #00H ;50ms次数，满20清零

MOV 31H, #00H ;控制移位间隔时间

MOV R0, #0FEH

CLR NEWFLAG

HERE:

MOV P1, R0

DELAY:JNB NEWFLAG, DELAY

CLR NEWFLAG

MOV 31H, #00H

MOV A, R0

RL A

MOV R0, A

SJMP HERE

T\_0:

CLR EA

PUSH PSW

PUSH Acc

SETB EA

MOV TL0, #0B0H

MOV TH0, #3CH

MOV A, 30H

CJNE A, #01H, ONE

CJNE A, #02H, TWO

CJNE A, #03H, THREE

CJNE A, #04H, EXIT

MOV 30H, #00H

EXIT:

CLR EA

POP Acc

POP PSW

SETB EA

RETI

INT\_0:

CLR EA

PUSH PSW

PUSH Acc

SETB EA

INC 30H

CPL P3.7

CLR EA

POP Acc

POP PSW

SETB EA

RETI

ONE: SETB NEWFLAG

RET

TWO:INC 31H

MOV R1, 31H

CJNE R1, #05H, EXIT

SETB NEWFLAG

RET

THREE: INC 31H

MOV R1, 31H

CJNE R1, #0AH, EXIT

SETB NEWFLAG

RET

FOUR:INC 31H

MOV R1, 31H

CJNE R1, #0FH, EXIT

SETB NEWFLAG

RET

END

实验四.

;秒表程序

;编写人：高洪伟

;编写日期2019-4-6

;修改日期2019-4-9

;T0中断时，R1指向26H--29H，分别存储毫秒，秒，分，时

;R0指向1EH--25H，分别存储毫秒，秒，分，时的段码

;T1中断时R1指向25H--1EH，寄存器间接寻址输出到P0口

;R3存储段码值，T1中断时输出到P0口，然后左移

LATCH1 BIT P2.2

LATCH2 BIT P2.3

ORG 0000H

LJMP START ;主程序

ORG 0003H

LJMP INT\_0 ;按键中断

ORG 000BH

LJMP T0\_INT ;定时器0，10ms

ORG 001BH

LJMP T1\_INT ;定时器1，1ms

START:

MOV TCON, #51H

MOV IP, #02H

MOV IE, #8BH

MOV TMOD, #11H ;都工作在方式1

MOV TL0, #0F0H

MOV TH0, #0D8H ;10ms中断

MOV TL1, #18H

MOV TH1, #0FCH ;1ms中断

MOV DPTR, #TABLE

MOV SP, #60H

MOV 32H, #00H ;按键次数

MOV R1, #25H ;指针，访问BCD码

MOV R3, #11111110B;位码初值

HERE: SJMP HERE ;原地等待中断

T1\_INT:

MOV TL1, #18H

MOV TH1, #0FCH

MOV P0, @R1 ;显示程序，@R1为段码输出到P0

SETB LATCH1

CLR LATCH1

MOV P0, R3 ;输出位码

SETB LATCH2

CLR LATCH2

DEC R1

MOV A, R3

RL A

MOV R3, A

CJNE R1, #1DH,EXIT\_T1\_INT

MOV R1, #25H

EXIT\_T1\_INT:RETI

T0\_INT:

CLR EA

PUSH PSW

PUSH Acc

MOV 31H, R1 ;假装R1入栈

SETB EA

MOV R1, #26H ;指针，访问BCD码

MOV R0, #1EH ;指针，存段码

MOV TL0, #0F0H

MOV TH0, #0D8H

MOV A, 32H ;按键计数值，选功能

CJNE A, #01H ,EQU2 ;按键值为1，计数，并转为换段码

SETB TR0

CALL TIME

CALL STORE

SJMP EXIT\_T0\_INT

EQU2:CJNE A, #02H, EQU3 ;按键值为2，跳过计数过程，实现暂停

SJMP EXIT\_T0\_INT

EQU3:

MOV 26H, #00H ;按键值为3，清零

MOV 27H, #00H

MOV 28H, #00H

MOV 29H, #00H

CALL STORE

EXIT\_T0\_INT: CLR EA

MOV R1, 31H

POP Acc

POP PSW

SETB EA

RETI

;TIME为毫秒，秒，分，时

TIME:

INC 26H

MOV A, 26H

CJNE A, #64H, EXIT\_TIME

MOV 26H, #00H

INC 27H

MOV A, 27H

CJNE A, #3CH, EXIT\_TIME

MOV 27H, #00H

INC 28H

MOV A, 28H

CJNE A, #3CH, EXIT\_TIME

MOV 28H, #00H

INC 29H

MOV A, 29H

CJNE A, #18H, EXIT\_TIME

MOV 29H, #00H

EXIT\_TIME: RET

;STORE为毫秒，秒，分，时 的段码

STORE:

MOV A, @R1

MOV B, #10

DIV AB

MOVC A, @A+DPTR

INC R0

MOV @R0, A

MOV A,B

MOVC A, @A+DPTR

DEC R0

MOV @R0, A

INC R0

INC R0

INC R1

CJNE R1, #2AH, STORE

RET

INT\_0:

CLR EA

PUSH PSW

PUSH Acc

SETB EA

INC 32H ;按键中断计数

MOV A, 32H

CJNE A, #04H, NEXT

MOV 32H, #01H

NEXT:CLR EA

POP Acc

POP PSW

SETB EA

EXIT\_INT\_0: RETI

TABLE:DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH ;共阴字码表

END