# 《考试报告》

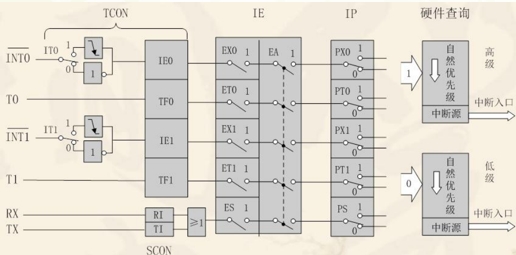
# 基本原理说明（附原理图及其他）

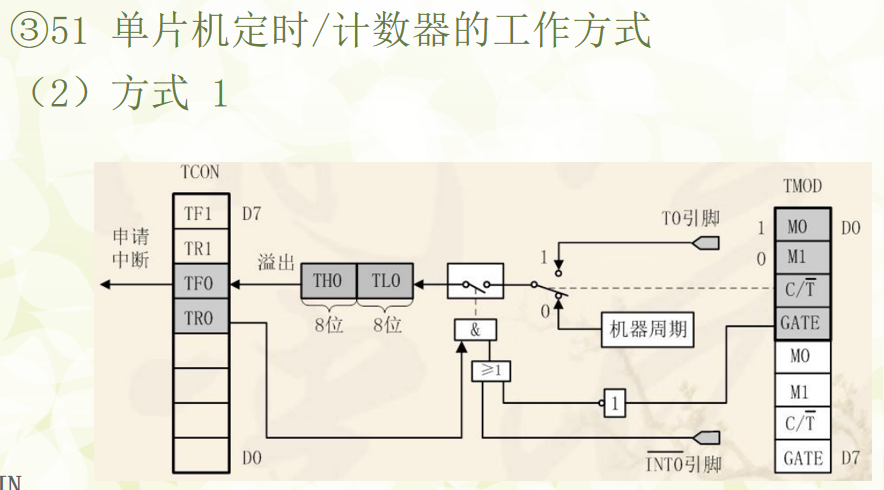
**实现目标 ： 使用定时器控制周期为 1s 的流水灯**

* 1. **定时器原理**



TMOD 的初始值设为 0x01, 使得 M1 = 0, M0 = 1，采用工作方式 1 ，16 位定时器





本次我们采用 T0 定时器，设定初值使其 1ms 中断一次。

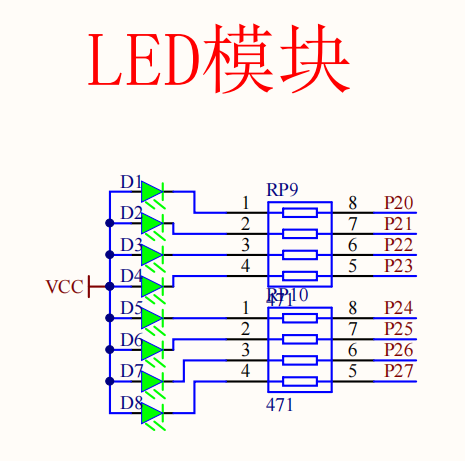
机器周期大约在 1ms, 初始值设为 65536 - 1000 = 64536 = 0xFC18 即可，即设定

TH0 = 0xFC, TL0 = 0x18。

然后开始定时器 0 的中断 ET0，总中断 EA 以及计时器的开关 TR0。

当接受中断次数 1000 次的时候，时间 1000ms = 1s, 此时我们依次点亮 LED 模块的对应 LED。假设当前点亮的编号是 ledNum, ledNum 从 0 ~ 7 循环，用 counter 表示总中断次数，counter 每次发生中断时加 1，加到 1000 时说明过了 1 s，此时将 counter 清零，点亮第 ledNum 个小灯。

* 1. **走马灯原理**



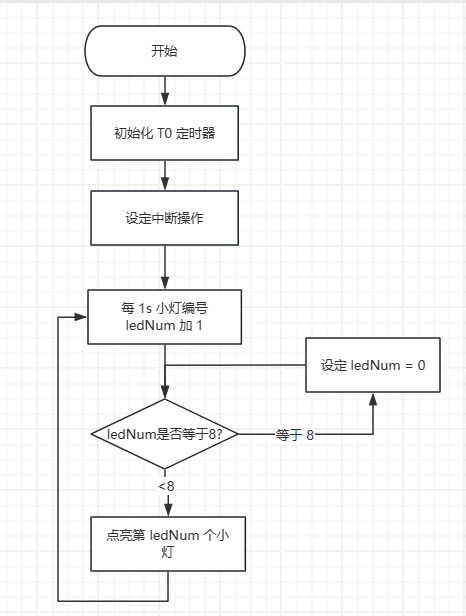
只需要实现一个点亮第 x 个灯的函数即可。

D1 到 D8 对应的引脚分别是 P20, P21, P22, P23, P24, P25, P26, P27, P28

对于 P2 ，我们将对应的位置为低电平 0 即可点亮对应小灯。

# 程序设计原理（附流程图）

**流程图设计如下 :**



我们最开始按照**基本原理说明部分**初始化配置定时器，设定中断操作，每 1s ledNum 加 1，然后点亮第 ledNum 个小灯，如果 ledNum 等于 8，将其改为 0。

如此一来就能循环点亮第 0 ~ 7 共 8 个小灯了。

# 程序代码及注释

## user 文件设计

### 3.1.1 main.c 文件

#include "public.h"

void main(){

T0\_1ms\_init(); // 初始化 T0 定时器

while(1){

}

}

### 3.1.2 public.c 文件

#include "public.h"

void delay\_10us(u16 ten\_us){ // 延时函数

while(ten\_us --);

}

### 3.1.3 public.h 头文件

#ifndef \_\_PUBLIC\_H\_\_

#define \_\_PUBLIC\_H\_\_

typedef unsigned char u8;

typedef unsigned int u16;

// 包含系统头文件

#include "reg52.h"

// 包含自己写的头文件, 注意要添加头文件的目录

#include "Eight\_Multiple\_Eight\_LED\_Matrix.h"

#include "timer.h"

// 函数声明

void delay\_10us(u16 ten\_us);

#endif

## LED 模块设计

### Eight\_Multiple\_Eight\_LED\_Matrix.h

#ifndef \_\_Eight\_Multiple\_Eight\_LED\_Matrix\_H\_\_

#define \_\_Eight\_Multiple\_Eight\_LED\_Matrix\_H\_\_

// 点亮 LED 模块第 x 个小灯

void light\_the\_x\_th\_led(u16 x);

// 流水灯

void waterfall\_light();

#endif

### Eight\_Multiple\_Eight\_LED\_Matrix.c

#include "public.h"

// 点亮单片机的第 x 个灯

void light\_the\_x\_th\_led(u16 x){

P2 = 0xff; // 全部熄灭

P2 &= ~(1 << x); // 将第 x 位置为 0

}

void waterfall\_light(){

u16 i, t;

for(i = 0; i < 8; i ++){

light\_the\_x\_th\_led(i);

delay\_10us(500 \* 100); // 延时 500 ms

}

}

## timer 定时器模块设计

### time.h

#include "public.h"

/\* 初始化 T0 计时器，使得 1ms 中断一次 \*/

void T0\_1ms\_init(){

/\*

设定T0定时器模式

M1 = 0, M0 = 1, 工作方式 1, 16 位定时器/计数器

\*/

TMOD = 0x01;

/\*

设定定时器初值, 使得 1ms 中断一次

两个 8 位计数器到 2^16 = 65536 会溢出

机器周期大约为 1us,

初始值设定为 64536 = 0xFC18 就能 1ms 中断一次(不精确)

\*/

TH0 = 0xFC;

TL0 = 0x18;

// 开启定时器0 中断

ET0 = 1;

// 开启总中断

EA = 1;

// 开启计时器开关 TR0

TR0 = 1;

}

void T0\_isr() interrupt 1{

// 中断发生总次数

static unsigned int counter = 0;

// 当前需要点亮小灯的编号

static unsigned int ledNum = 0;

counter ++;

if(counter == 1000){ // 1000 \* 1 = 1000 ms

counter = 0;

light\_the\_x\_th\_led(ledNum);

ledNum ++;

if(ledNum == 8) ledNum = 0;

}

// 保证 1ms 的计时器中断周期

TH0 = 0xFC;

TL0 = 0x18;

}

### time.c

#ifdef \_\_timer\_H\_\_

#define \_\_timer\_H\_\_

/\* 初始化 T0 计时器，使得 1ms 中断一次 \*/

void T0\_1ms\_init();

/\* 中断操作 \*/

void T0\_isr();

#endif