一、定时器的功能以及定时器的结构

定时器的功能

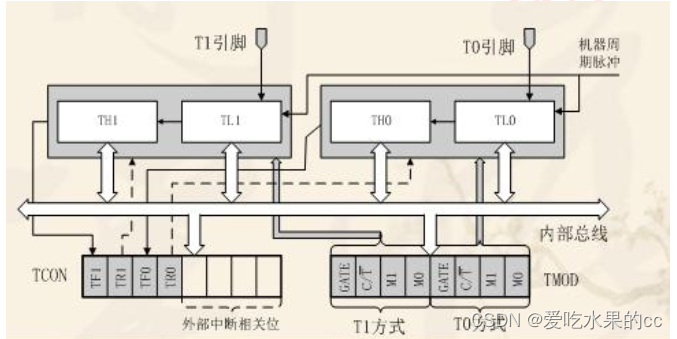
其实就是单片机的内部，通过系统时钟的每一个机器周期产生一个记数脉冲，即每一个机器周期计数器加一。

这里我的实验板的晶振是12MHZ,1MHZ信号每个脉冲的持续时间为1us,如果定时器T0对1MHZ的信号进行计数，从0~65536us,当达到最大的65536us的时候，定时器计数达到最大值，会溢出，于是产生中断信号，向中断系统申请中断，中断系统接受中断请求，执行中断子程序。

定时器的结构

定时器的结构如下图所示，主要包括

* **两个定时器/计数器**。T0和T1,每个定时器/计数器都是由两个8位的计数器所构成的16位计数器。
* **TCON 寄存器**。TCON为控制寄存器，用来控制两个定时器/计数器的启动和停止。
* **TMOD寄存器**。TMOD为工作方式控制寄存器，用来设置定时器/计数器的工作方式。



二、定时器的控制

工作模式寄存器TMOD

TMOD为工作方式控制寄存器，用来设置定时器/计数器的工作方式。如下图所示。

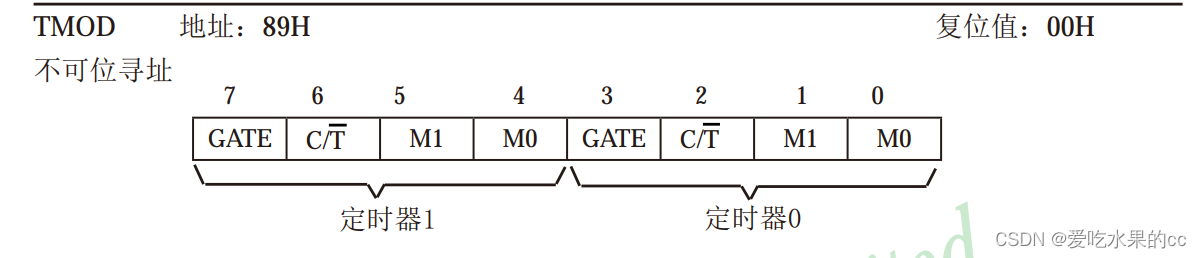
通过配置TMOD寄存器来对定时器T0和T1的工作模式进行控制。

注意这里TMOD的地址为89H，不可位寻址。

TMOD的高四位用于T1，低四位用于T0。

其中主要各位的功能：

* **C/-T**，定时器/计数器的选择控制位。置0，为定时器模式，置1，为计数器模式。
* **M1和M0**，模式选择控制位，通过对两位进行赋值，可以选择定时器的4种模式。00，模式0，13位计数器。01，模式1，16位计数器。10，模式2，自动重装8位计数器，11，模式3。
* **GATE**,置1后，就可由TR0或TR1单独控制定时器。



控制寄存器TCON

TCON寄存器，地址为88H,可以字节寻址，也可位寻址。寄存器各位如下图所示。

其中各位的功能：

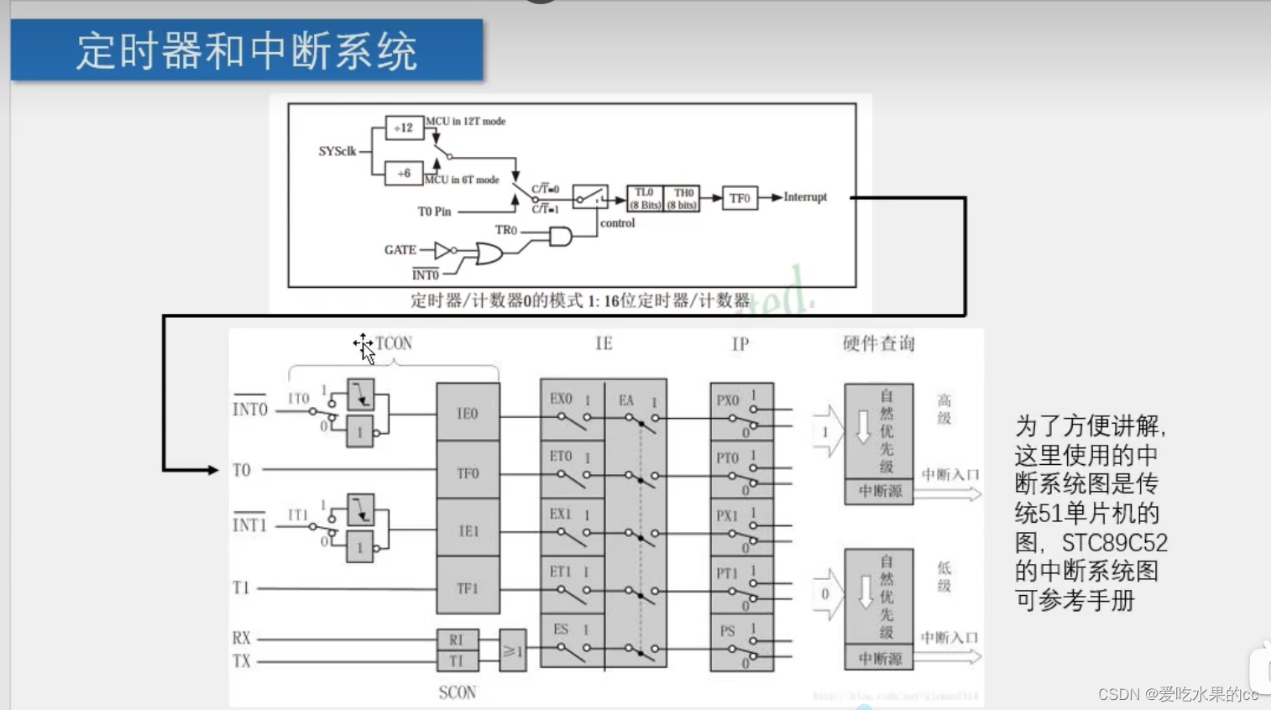
* **TF1（TF0）**。定时器T1（T0）溢出标志位。当T1（T0）溢出时，硬件自动使TF1（TF0）置1，并且向cpu申请中断。当Cpu响应中断，进入中断服务子程序后，TF1（TF0）由硬件自动清0，当然也可以用软件写代码清0。
* **TR0（TR1）**。定时器T1（T0）运行控制位，置1，定时器T1（T0）就开始运行，计数。
* **后面几个是外部中断控制位**。



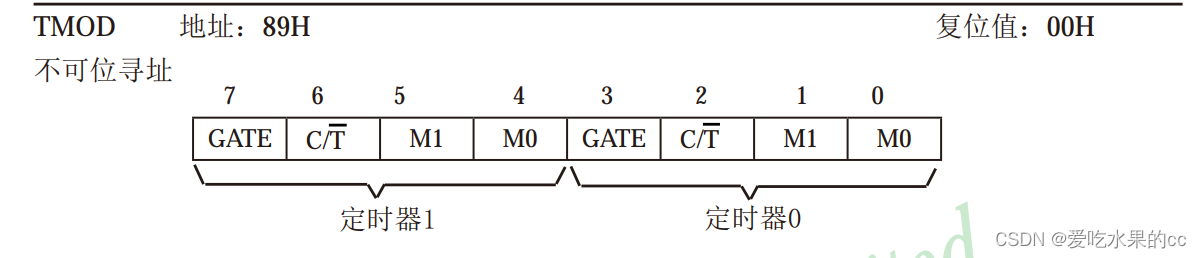
写代码来初始化定时器

定时器的配置主要是通过对两个寄存器TMOD和TCON进行配置，这里我通过配置定时器0，模式1引发中断，配置其他的定时器或者是不同的模式都是大同小异。看看模式1的结构。

**模式1的结构**



我们首先来配置寄存器TMOD，根据图来配置。



只需要配置定时器0，那么高四位就不管了，置0，而我们在控制定时器0的低四位中配置为0001。

GATE=0; //直接由TR0控制定时器0的开启

C/-T=0; //选择定时器模式

M1=0; //选择模式1

M2=1;

**继续配置寄存器TCON**。



只需要配置定时器0相关的部分就可以了，再一个，TCON寄存器是可位寻址的，所以只需要单独对其中的某一位进行置值就可以了。

所以：

TF0=0; //定时器0溢出控制标志，当计数到溢出65536us时，就会置1。

TR0=1; //定时器0启动，开启计时。

**配置中断**

当计数到溢出后，就会向cpu发出中断请求，申请中断，进入中断子程序。然后出来，TF0由1->0,然后循环循环。

所以：

ET0=1; //中断的配置

EA=1;

PT0=0;

三、定时器引发中断

**简易时钟**

**使用定时器，采用LCD1602，实现简易时钟，秒，分，时。**

下面是源代码：

主程序main.c

#include <REGX52.H>

#include "Delay.h"

#include "Timer0.h"

#include "LCD1602.h"

unsigned char sec=55,min=59,hour=23;

void main()

{

LCD\_Init(); //LCD初始化

LCD\_ShowString(1,1,"COLCK:");

Timer0Init(); //定时器0初始化

while(1)

{

LCD\_ShowNum(2,1,hour,2);

LCD\_ShowString(2,3,":");

LCD\_ShowNum(2,4,min,2);

LCD\_ShowString(2,6,":");

LCD\_ShowNum(2,7,sec,2);

}

}

void TimerRoutine() interrupt 1

{

static unsigned int T0Count;

//当触发中断后，每次中断结束后，初始值还是为64535 即1ms

TL0 = 0x66; //设置定时初值

TH0 = 0xFC; //设置定时初值

T0Count++;

if(T0Count>=1000) //一次是1ms，\*1000就是一秒

{

T0Count=0;

sec++;

if(sec>=60)

{

sec=0;

min++;

if(min>=60)

{

min=0;

hour++;

if(hour>=24)

{

hour=0;

sec=0;

min=0;

}

}

}

}

}

延时函数Delay.c

//延时

void Delay(unsigned char xms) //@11.0592MHz

{

unsigned char i, j;

while(xms--)

{

//\_nop\_();

i = 2;

j = 199;

do

{

while (--j);

} while (--i);

}

}

控制LCD162模块LCD1602.c

#include <REGX52.H>

//引脚配置：

sbit LCD\_RS=P2^6;

sbit LCD\_RW=P2^5;

sbit LCD\_EN=P2^7;

#define LCD\_DataPort P0

//函数定义：

/\*\*

\* @brief LCD1602延时函数，12MHz调用可延时1ms

\* @param 无

\* @retval 无

\*/

void LCD\_Delay()

{

unsigned char i, j;

i = 2;

j = 239;

do

{

while (--j);

} while (--i);

}

/\*\*

\* @brief LCD1602写命令

\* @param Command 要写入的命令

\* @retval 无

\*/

void LCD\_WriteCommand(unsigned char Command)

{

LCD\_RS=0;

LCD\_RW=0;

LCD\_DataPort=Command;

LCD\_EN=1;

LCD\_Delay();

LCD\_EN=0;

LCD\_Delay();

}

/\*\*

\* @brief LCD1602写数据

\* @param Data 要写入的数据

\* @retval 无

\*/

void LCD\_WriteData(unsigned char Data)

{

LCD\_RS=1;

LCD\_RW=0;

LCD\_DataPort=Data;

LCD\_EN=1;

LCD\_Delay();

LCD\_EN=0;

LCD\_Delay();

}

/\*\*

\* @brief LCD1602设置光标位置

\* @param Line 行位置，范围：1~2

\* @param Column 列位置，范围：1~16

\* @retval 无

\*/

void LCD\_SetCursor(unsigned char Line,unsigned char Column)

{

if(Line==1)

{

LCD\_WriteCommand(0x80|(Column-1));

}

else if(Line==2)

{

LCD\_WriteCommand(0x80|(Column-1+0x40));

}

}

/\*\*

\* @brief LCD1602初始化函数

\* @param 无

\* @retval 无

\*/

void LCD\_Init()

{

LCD\_WriteCommand(0x38);//八位数据接口，两行显示，5\*7点阵

LCD\_WriteCommand(0x0c);//显示开，光标关，闪烁关

LCD\_WriteCommand(0x06);//数据读写操作后，光标自动加一，画面不动

LCD\_WriteCommand(0x01);//光标复位，清屏

}

/\*\*

\* @brief 在LCD1602指定位置上显示一个字符

\* @param Line 行位置，范围：1~2

\* @param Column 列位置，范围：1~16

\* @param Char 要显示的字符

\* @retval 无

\*/

void LCD\_ShowChar(unsigned char Line,unsigned char Column,char Char)

{

LCD\_SetCursor(Line,Column);

LCD\_WriteData(Char);

}

/\*\*

\* @brief 在LCD1602指定位置开始显示所给字符串

\* @param Line 起始行位置，范围：1~2

\* @param Column 起始列位置，范围：1~16

\* @param String 要显示的字符串

\* @retval 无

\*/

void LCD\_ShowString(unsigned char Line,unsigned char Column,char \*String)

{

unsigned char i;

LCD\_SetCursor(Line,Column);

for(i=0;String[i]!='\0';i++)

{

LCD\_WriteData(String[i]);

}

}

/\*\*

\* @brief 返回值=X的Y次方

\*/

int LCD\_Pow(int X,int Y)

{

unsigned char i;

int Result=1;

for(i=0;i<Y;i++)

{

Result\*=X;

}

return Result;

}

/\*\*

\* @brief 在LCD1602指定位置开始显示所给数字

\* @param Line 起始行位置，范围：1~2

\* @param Column 起始列位置，范围：1~16

\* @param Number 要显示的数字，范围：0~65535

\* @param Length 要显示数字的长度，范围：1~5

\* @retval 无

\*/

void LCD\_ShowNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length)

{

unsigned char i;

LCD\_SetCursor(Line,Column);

for(i=Length;i>0;i--)

{

LCD\_WriteData(Number/LCD\_Pow(10,i-1)%10+'0');

}

}

/\*\*

\* @brief 在LCD1602指定位置开始以有符号十进制显示所给数字

\* @param Line 起始行位置，范围：1~2

\* @param Column 起始列位置，范围：1~16

\* @param Number 要显示的数字，范围：-32768~32767

\* @param Length 要显示数字的长度，范围：1~5

\* @retval 无

\*/

void LCD\_ShowSignedNum(unsigned char Line,unsigned char Column,int Number,unsigned char Length)

{

unsigned char i;

unsigned int Number1;

LCD\_SetCursor(Line,Column);

if(Number>=0)

{

LCD\_WriteData('+');

Number1=Number;

}

else

{

LCD\_WriteData('-');

Number1=-Number;

}

for(i=Length;i>0;i--)

{

LCD\_WriteData(Number1/LCD\_Pow(10,i-1)%10+'0');

}

}

/\*\*

\* @brief 在LCD1602指定位置开始以十六进制显示所给数字

\* @param Line 起始行位置，范围：1~2

\* @param Column 起始列位置，范围：1~16

\* @param Number 要显示的数字，范围：0~0xFFFF

\* @param Length 要显示数字的长度，范围：1~4

\* @retval 无

\*/

void LCD\_ShowHexNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length)

{

unsigned char i,SingleNumber;

LCD\_SetCursor(Line,Column);

for(i=Length;i>0;i--)

{

SingleNumber=Number/LCD\_Pow(16,i-1)%16;

if(SingleNumber<10)

{

LCD\_WriteData(SingleNumber+'0');

}

else

{

LCD\_WriteData(SingleNumber-10+'A');

}

}

}

/\*\*

\* @brief 在LCD1602指定位置开始以二进制显示所给数字

\* @param Line 起始行位置，范围：1~2

\* @param Column 起始列位置，范围：1~16

\* @param Number 要显示的数字，范围：0~1111 1111 1111 1111

\* @param Length 要显示数字的长度，范围：1~16

\* @retval 无

\*/

void LCD\_ShowBinNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length)

{

unsigned char i;

LCD\_SetCursor(Line,Column);

for(i=Length;i>0;i--)

{

LCD\_WriteData(Number/LCD\_Pow(2,i-1)%2+'0');

}

}

定时器0模块Timer0.c

主要是对定时器进行配置。

#include <REGX52.H>

/\*\*

\* @brief 定时器0初始化

\* @param

\* @retval

\*/

void Timer0Init()

{

TMOD&=0xF0; //高四位不变

TMOD|=0x01; //设置定时器模式1 以及设置为定时方式 0

TL0 = 0x66; //设置定时初值

TH0 = 0xFC; //设置定时初值

TF0=0; //定时器0溢出标志位

TR0=1; //定时器0运行控制位

TF0=1; //设置外部中断

ET0=1;

EA=1;

PT0=0;

}

四、实现效果

如下图。

