**Microcontroller Experiment**

Timer Interrupt Experiment

Name: Han Yichen 韩一尘

Student ID: 22722051

Date: 2023/11/1

1. **Experimental Principles**

We will use the timer interrupt of TIM3 to control the rollover of DS1, and use the rollover of DS0 in the main function to indicate that the program is running. The universal timer of STM32F1 is a 16-bit auto-loading counter (CNT) driven by a programmable prescaler (PSC). We will use a timer to generate an interrupt, and then flip the level on DS1 in the interrupt service function to indicate the timer interrupt.

1)TIM3 clock is enabled.

2) Initialize timer parameters, and set automatic reloading value, frequency division coefficient, counting mode, etc.

3) Set TIM3\_DIER to allow update interrupt.

4)TIM3 interrupt priority setting.

5) allow TIM3 to work, that is, enable TIM3.

6) Write the interrupt service function.

**2. Main program analysis**

#include "timer.h"

#include "led.h"

//通用定时器 3 中断初始化

//这里时钟选择为 APB1 的 2 倍，而 APB1 为 36M

//arr：自动重装值。

//psc：时钟预分频数

//这里使用的是定时器 3!

void TIM3\_Int\_Init(u16 arr,u16 psc)

{

TIM\_TimeBaseInitTypeDef TIM\_TimeBaseStructure;

NVIC\_InitTypeDef NVIC\_InitStructure;

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM3, ENABLE); //①时钟 TIM3 使能

//定时器 TIM3 初始化

TIM\_TimeBaseStructure.TIM\_Period = arr; //设置自动重装载寄存器周期的值

TIM\_TimeBaseStructure.TIM\_Prescaler =psc; //设置时钟频率除数的预分频值

TIM\_TimeBaseStructure.TIM\_ClockDivision = TIM\_CKD\_DIV1; //设置时钟分割

TIM\_TimeBaseStructure.TIM\_CounterMode = TIM\_CounterMode\_Up; //TIM 向上计数

TIM\_TimeBaseInit(TIM3, &TIM\_TimeBaseStructure); //②初始化 TIM3

TIM\_ITConfig(TIM3,TIM\_IT\_Update,ENABLE ); //③允许更新中断

//中断优先级 NVIC 设置

NVIC\_InitStructure.NVIC\_IRQChannel = TIM3\_IRQn; //TIM3 中断

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0; //先占优先级 0 级

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 3; //从优先级 3 级

NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //IRQ 通道被使能

NVIC\_Init(&NVIC\_InitStructure); //④初始化 NVIC 寄存器

TIM\_Cmd(TIM3, ENABLE); //⑤使能 TIM3

}

//定时器 3 中断服务程序⑥

void TIM3\_IRQHandler(void) //TIM3 中断

{

if (TIM\_GetITStatus(TIM3, TIM\_IT\_Update) != RESET) //检查 TIM3 更新中断发生与否

{

TIM\_ClearITPendingBit(TIM3, TIM\_IT\_Update ); //清除 TIM3 更新中断标志

LED1=!LED1;

}

}

We need to write our own time.c file, in which two parameters of the TIM3\_Int\_Init () function are used to set the overflow time of TIM3. TIM3' s clock is 72M, and then according to the values of arr and psc designed by us, the interrupt time can be calculated. The calculation formula is as follows:

Tout= ((arr+1)\*(psc+1))/Tclk； Where: TCL k: the input clock frequency of Tim 3 (in Mhz). Tout: Tim 3 overflow time (in us). Finally, we enter the following code in the main program:

int main(void)

{

delay\_init();

//延时函数初始化

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);

//设置 NVIC 中断分组 2

uart\_init(115200); //串口初始化波特率为 115200

LED\_Init();

//LED 端口初始化

TIM3\_Int\_Init(4999,7199); //10Khz 的计数频率，计数到 5000 为 500ms

while(1)

{

LED0=!LED0;

delay\_ms(200);

}

}

The code here is almost the same as before. After initializing TIM3, this code enters an infinite loop and waits for TIM3.

Overflow interrupt, when the value of TIM3\_CNT is equal to the value of TIM3\_ARR, the update interrupt of TIM3 will be generated. However,

After that, take the inverted LED1 and TIM3\_CNT in the interrupt and count from 0. According to the above formula, we can calculate the interrupt.

The overflow time is 500ms, that is, tout = ((4999+1) \* (7199+1))/72 = 500000us = 500ms.

**3.Experimental result :**

The LED light flashes automatically according to our expected frequency.

**4. Improvement and perfection**

For the problems encountered, the first experiment was unclear about the meanings of the two parameters controlling the flashing frequency of green light in the main function. By controlling the variable method, the values of the two parameters were changed several times and the experimental results were recorded. Finally, they were summarized as arr and PSC, and the relationship between them and interrupt output was roughly calculated. Finally, by changing various parameters, it is found that it meets the expectations.