

Lab3 TIM2 精准延时

一 实验环境

二 软件实现

一 实验环境

下载 STM32F10x_StdPeriph_Lib_V3.6.0

手动搭建 STM32 工程项目

WIN10 19045

Keil MDK v5.3.9.0.0

ARM Compiler v6.21

VSCode

二 软件实现

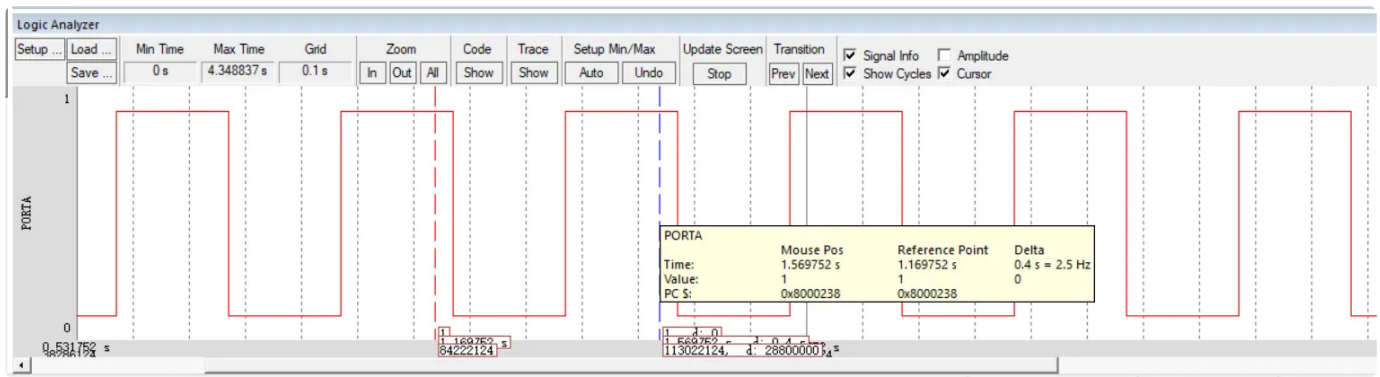
实验要求：用连接红色 LED 的 PA8 引脚的输出信号波形图描述软件仿真结果

复用 Lab2 中 LED 部分代码，而后定义一个新的函数 `TIM_Init`，为了保持代码简洁以及便于后期修改，定义了一些宏。

其中，`TIM_Init` 函数的描述及实现如下：

```
1 // 省略宏定义
2 void BASIC_TIM_Config(void)
3 {
4     TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
5
6     // 开启定时器时钟,即内部时钟CK_INT=72M
7     // 库函数中 APB1预分频的系数是2, 即 PCLK1=36M, 所以定时器时钟 TIMxCLK=36*2=72
8     M BASIC_TIM_APBxClock_FUN(BASIC_TIM_CLK, ENABLE);
9
10    // 自动重装载寄存器周的值(计数值)
11    // 累计TIM_Period个频率后产生一个更新或者中断
12    TIM_TimeBaseStructure.TIM_Period=1000;
13
14    // 定时器的定时时间等于计数器的中断周期乘以中断的次数
15    TIM_TimeBaseStructure.TIM_Prescaler= 84240; // 72*(117*10)
16
17
18    TIM_TimeBaseStructure.TIM_ClockDivision=TIM_CKD_DIV1;
19    TIM_TimeBaseStructure.TIM_CounterMode=TIM_CounterMode_Up;
20    TIM_TimeBaseStructure.TIM_RepetitionCounter=0;
21
22    // 初始化定时器
23    TIM_TimeBaseInit(BASIC_TIM, &TIM_TimeBaseStructure);
24
25    // 清除计数器中断标志位
26    TIM_ClearFlag(BASIC_TIM, TIM_FLAG_Update);
27
28    // 开启计数器中断
29    TIM_ITConfig(BASIC_TIM, TIM_IT_Update, ENABLE);
30
31    // 使能计数器
32    TIM_Cmd(BASIC_TIM, ENABLE);
33
34    // 暂时关闭定时器的时钟, 等待使用
35    BASIC_TIM_APBxClock_FUN(BASIC_TIM_CLK, DISABLE);
36 }
```

模拟结果如下:



存在的问题：

- 进行仿真时，个人不确定最终的仿真结果是否正确
- 反复阅读代码后确认代码无误
- 对定时器定时的公式未掌握

源代码：

```
1  /* Includes -----*/
2  #include "stm32f10x.h"
3  #include "stm32f10x_gpio.h"
4  #include "stm32f10x_rcc.h"
5  #include "stm32f10x_tim.h"
6  /* Macros -----*/
7  #define LED_PIN GPIO_Pin_8
8  #define LED_PORT GPIOA
9  #define LED_CLOCK RCC_APB2Periph_GPIOA
10
11 #define BASIC_TIM TIM2
12 #define BASIC_TIM_APBxClock_FUN RCC_APB1PeriphClockCmd
13 #define BASIC_TIM_CLK RCC_APB1Periph_TIM2
14
15 #define TIM_PERIOD 1000
16 /* LED0 初始化模板 (可复用) -----*/
17 void LED_Init(void)
18 {
19     // Define a structure of type GPIO_InitTypeDef.
20     GPIO_InitTypeDef GPIO_InitStructure;
21
22     // Enable the clock for GPIOA by opening the APB2 bus clock.
23     RCC_APB2PeriphClockCmd(LED_CLOCK, ENABLE);
24
25     // Configure Pin8 of GPIOB as push-pull output.
26     GPIO_InitStructure.GPIO_Pin = LED_PIN; // Specifies the GPIO pins
27     // to be configured.
28     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; // Push-pull output.
29     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz; // Specifies the speed
30     // for the selected pins.
31     GPIO_Init(LED_PORT, &GPIO_InitStructure);
32 }
33 void LED0_ON(void)
34 {
35     GPIO_ResetBits(LED_PORT, LED_PIN);
36 }
37 void LED0_OFF(void)
38 {
39     GPIO_SetBits(LED_PORT, LED_PIN); // GPIOx: A,B,C,D,E,F,G; GPIO_Pin: Specifies
40     // the port bit to be written.
```

```

39
40  /* 使用轮询 (Polling) 的 TIM2 精准控制 LED 的闪烁 (可复用) -----
    -----*/
41  /**
42   * @file main.c
43   * @brief Configure TIM2
44   * @return None
45   */
46  void BASIC_TIM_Config(void)
47  {
48      TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
49
50      // 开启定时器时钟,即内部时钟CK_INT=72M
51      // 库函数中 APB1预分频的系数是2, 即 PCLK1=36M, 所以定时器时钟 TIMxCLK=36*2=72
M
52      BASIC_TIM_APBxClock_FUN(BASIC_TIM_CLK, ENABLE);
53
54      // 自动重装载寄存器周的值(计数值)
55      // 累计TIM_Period个频率后产生一个更新或者中断
56      TIM_TimeBaseStructure.TIM_Period=1000;
57
58      // 定时器的定时时间等于计数器的中断周期乘以中断的次数
59      TIM_TimeBaseStructure.TIM_Prescaler= 84240; // 72*(117*10)
60
61
62      TIM_TimeBaseStructure.TIM_ClockDivision=TIM_CKD_DIV1;
63      TIM_TimeBaseStructure.TIM_CounterMode=TIM_CounterMode_Up;
64      TIM_TimeBaseStructure.TIM_RepetitionCounter=0;
65
66      // 初始化定时器
67      TIM_TimeBaseInit(BASIC_TIM, &TIM_TimeBaseStructure);
68
69      // 清除计数器中断标志位
70      TIM_ClearFlag(BASIC_TIM, TIM_FLAG_Update);
71
72      // 开启计数器中断
73      TIM_ITConfig(BASIC_TIM,TIM_IT_Update,ENABLE);
74
75      // 使能计数器
76      TIM_Cmd(BASIC_TIM, ENABLE);
77
78      // 暂时关闭定时器的时钟, 等待使用
79      BASIC_TIM_APBxClock_FUN(BASIC_TIM_CLK, DISABLE);
80  }
81
82
83  int main(void)

```

```
84  {
85      LED_Init();
86
87      while (1)
88      {
89          LED0_ON();
90          BASIC_TIM_Config();
91          LED0_OFF();
92          BASIC_TIM_Config();
93      }
94  }
```