

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

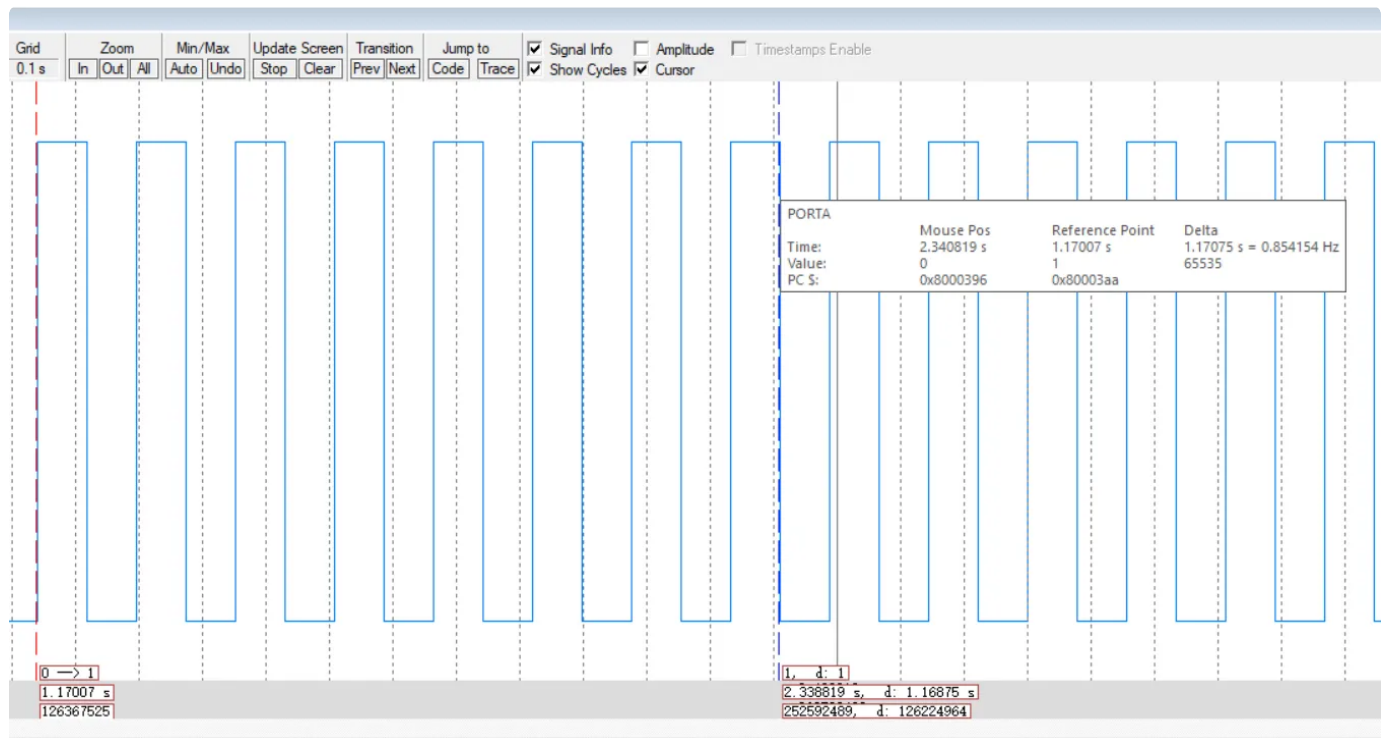
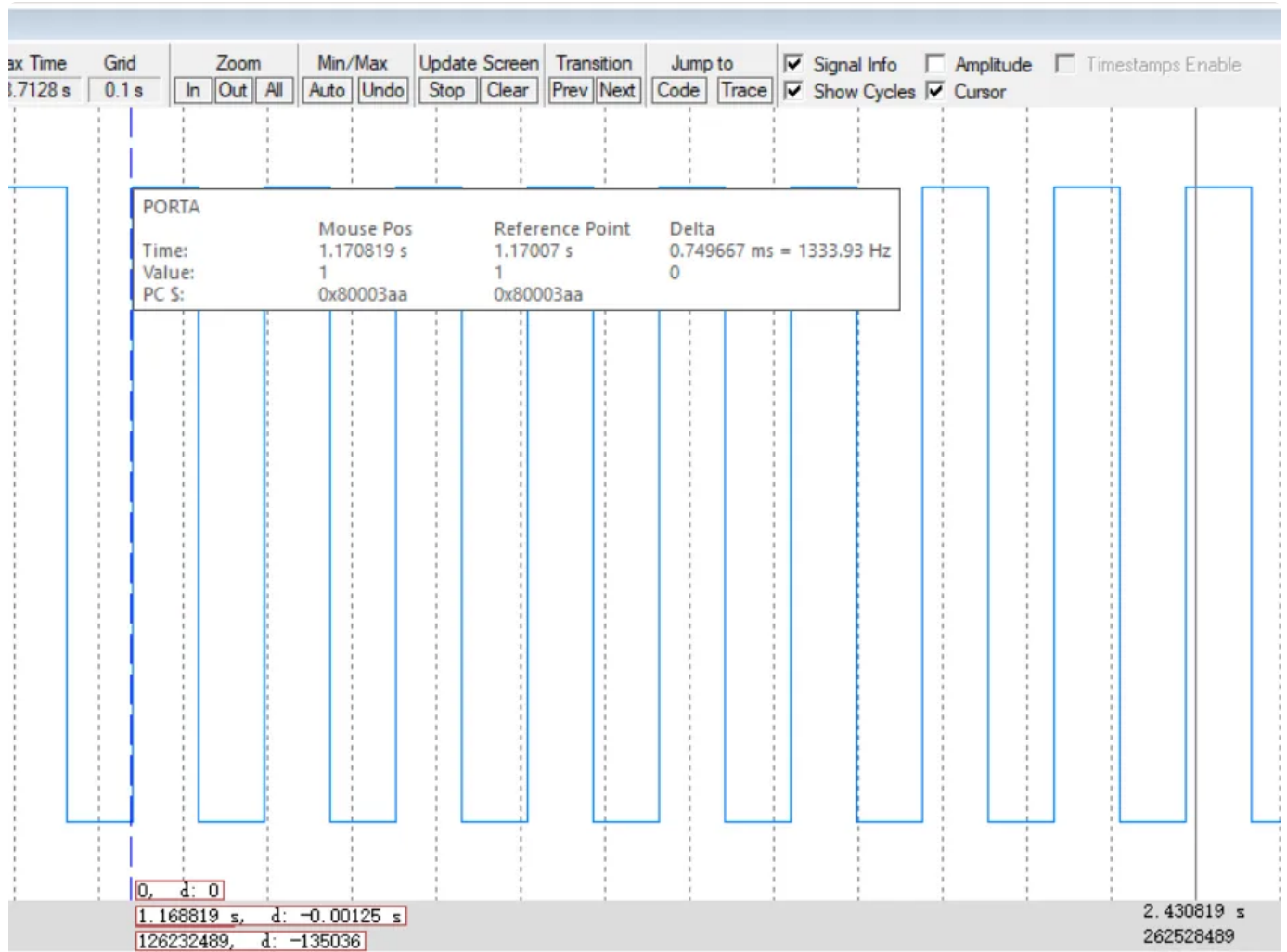
二 软件实现

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

实现：

为了保持代码简洁以及便于后期修改，定义了一些宏。

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
- Keil MDK 仿真器不会用

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 /* 定时器设置公式 -----
   -----
16 x=1ms, 即timer=1000  T*TCLK=x*(PSC+1)*(ARR+1)  PSC=72T-1
17 T=1.17s  ARR=1000-1  PSC=72*100*1.17-1
18 在这之中省略了一个x, 且其默认为1000ms, 为什么?
19 */
20
21 // 自定义延时(单位: s)
22 #define DELAY 1.17
23
24 #define TIM_ARR_VALUE 1000 - 1 // 设定自动重载寄存器的值
25 #define TIM_PSC_VALUE (72 * 100 * DELAY) - 1 // 设定预分频系数
26
27 /* LED0 初始化模板 (可复用) -----
   -----*/
28 void LED_Init(void)
29 {
30     // Define a structure of type GPIO_InitTypeDef.
31     GPIO_InitTypeDef GPIO_InitStructure;
32
33     // Enable the clock for GPIOA by opening the APB2 bus clock.
34     RCC_APB2PeriphClockCmd(LED_CLOCK, ENABLE);
35
36     // Configure Pin8 of GPIOB as push-pull output.
37     GPIO_InitStructure.GPIO_Pin = LED_PIN; // Specifies the GPIO pins
   to be configured.
38     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; // Push-pull output.
39     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz; // Specifies the speed
   for the selected pins.

```

```

40     GPIO_Init(LED_PORT, &GPIO_InitStructure);
41 }
42 void LED0_ON(void)
43 {
44     GPIO_ResetBits(LED_PORT, LED_PIN);
45 }
46 void LED0_OFF(void)
47 {
48     GPIO_SetBits(LED_PORT, LED_PIN); // GPIOx: A,B,C,D,E,F,G; GPIO_Pin: Specifies the port bit to be written.
49 }
50
51 /* 使用轮询 (Polling) 的 TIM2 精准控制 LED 的闪烁 (可复用) -----
-----*/
52 /**
53  * @file main.c
54  * @brief Configure TIM2
55  * @return None
56  */
57 void BASIC_TIM_Config(void)
58 {
59     TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
60
61     // 开启定时器时钟TCLK=72M,即内部时钟CK_INT=72M
62     // 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=72M
63     BASIC_TIM_APBxClock_FUN(BASIC_TIM_CLK, ENABLE);
64
65     // 自动重装载寄存器周期的值(计数值)
66     // 累计TIM_Period个频率后产生一个更新或者中断
67     TIM_TimeBaseStructure.TIM_Period = TIM_ARR_VALUE;
68
69     // 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
70     TIM_TimeBaseStructure.TIM_Prescaler = TIM_PSC_VALUE;
71
72     TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
73     TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
74     TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
75
76     // 初始化定时器
77     TIM_TimeBaseInit(BASIC_TIM, &TIM_TimeBaseStructure);
78
79     // 清除计数器中断标志位
80     TIM_ClearFlag(BASIC_TIM, TIM_FLAG_Update);
81
82     // 使能计数器
83     TIM_Cmd(BASIC_TIM, ENABLE);
84     // 等待计数器更新
85     while (TIM_GetFlagStatus(BASIC_TIM, TIM_FLAG_Update) == RESET);

```

```
86 }
87
88 int main(void)
89 {
90     LED_Init();
91
92     while (1)
93     {
94         LED0_ON();
95         BASIC_TIM_Config();
96         LED0_OFF();
97         BASIC_TIM_Config();
98     }
99 }
```

更新时间:

2024.05.22 03: 28