

Hi,

I set an interruption handler for a timer. Each time it calls the handler I change the state of a IO.

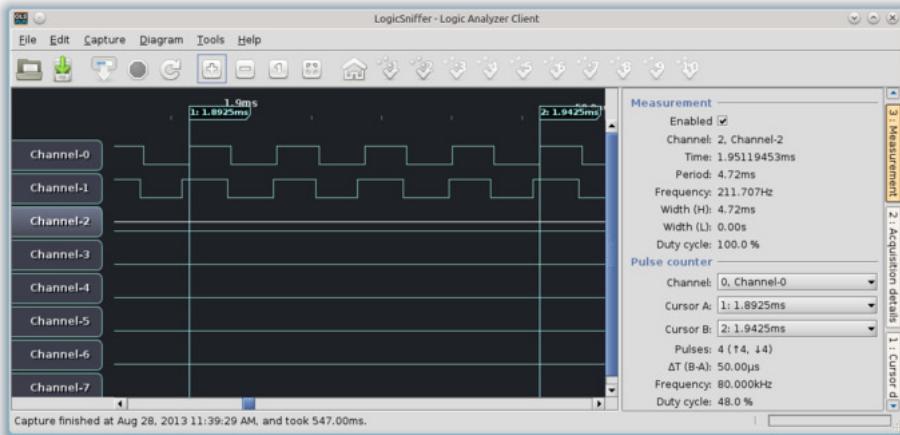
But I'm getting problems for frequency above 80kHz. If I set the period to:

```
TIM_TimeBaseStructure.TIM_Period = SystemCoreClock/160000 - 1;
```

I got an 80kHz in the channel output and in the IO that is changed by the interruption:

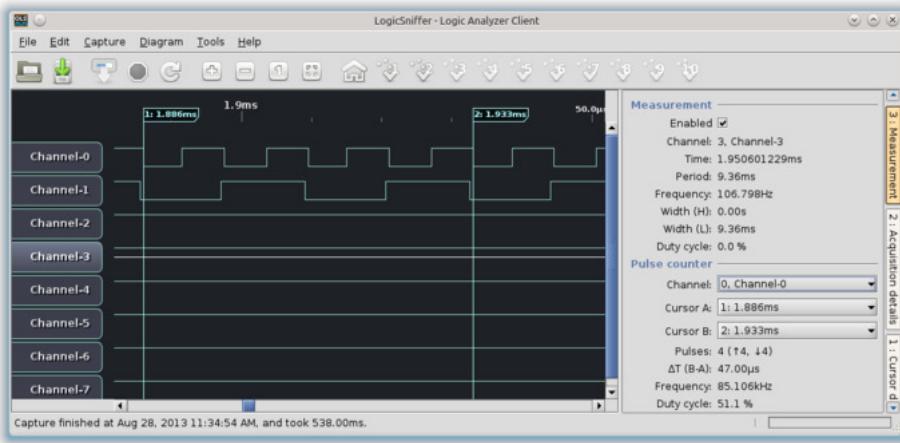
Channel-0: PA2 - Timer 2 Channel 2

Channel-1: PC9 - IO



But if I set the period to:

```
TIM_TimeBaseStructure.TIM_Period = SystemCoreClock/160000 - 1; It will create a 85kHz on OC2 but the IO drive by the interruption can't handle it. Am I
```



The code:

```
01. #include <stm32f10x.h>
02. #include <stm32f10x_rcc.h>
03. #include <stm32f10x_gpio.h>
04. #include <stm32f10x_tim.h>
05. #include <misc.h>
06.
07. #define STPDVR_TIMFREQ 100000
08.
09. static __IO uint8_t ledval;
10. static __IO uint8_t change;
11.
12. int main(void)
13. {
14.     ledval=0;
15.     change=0;
16.
17.     GPIO_InitTypeDef GPIO_InitStruct;
18.     TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
19.     TIM_OCInitTypeDef TIM_OCInitStructure;
20.
21.     //Configure Clocks
22.     RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
23.     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
24.     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);
25.
26.     //Configure GPIO PC9
27.     GPIO_StructInit(&GPIO_InitStruct);
28.     GPIO_InitStruct.GPIO_Pin = GPIO_Pin_9;
29.     GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
30.     GPIO_InitStruct.GPIO_Mode = GPIO_Mode_Out_PP;
31.     GPIO_Init(GPIOC, &GPIO_InitStruct);
32.
33.     //Configure GPIO PA1
34.     GPIO_InitStruct.GPIO_Pin = GPIO_Pin_1;
```

```

35.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
36.     GPIO_Init(GPIOA, &GPIO_InitStructure);
37.
38. //configure NVIC
39. NVIC_InitTypeDef NVIC_InitStructure;
40.
41. NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn;
42. NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
43. NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
44. NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
45. NVIC_Init(&NVIC_InitStructure);
46.
47. //configure timer
48. TIM_TimeBaseStructInit(&TIM_TimeBaseStructure);
49. TIM_TimeBaseStructure.TIM_Prescaler = 0;
50. TIM_TimeBaseStructure.TIM_Period = SystemCoreClock/170000 - 1;
51. TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
52. TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);
53.
54. //configure channel TIM2 C2
55. TIM_OCStructInit(&TIM_OCInitStructure);
56. TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Toggle;
57. TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
58. TIM_OC2Init(TIM2, &TIM_OCInitStructure);
59.
60. /* TIM2 TRGO selection */
61. TIM_ITConfig(TIM2, TIM_IT_Update, ENABLE);
62. TIM_Cmd(TIM2, ENABLE);
63.
64. while(1){
65.     if(change){
66.         GPIO_WriteBit(GPIOC, GPIO_Pin_9, (ledval)? Bit_SET : Bit_RESET);
67.         ledval = 1 - ledval;
68.         change=0;
69.     }
70.
71. };
72.
73. return 0;
74. }
75.
76. void TIM2_IRQHandler(void){
77.
78.     change=1;
79.
80.     TIM_ClearITPendingBit(TIM2,TIM_IT_Update);
81. }
82.
83.
84. #ifdef USE_FULL_ASSERT
85. void assert_failed(uint8_t* file, uint32_t line)
86. {
87.     while(1);
88. }
89. #endif

```

Thanks for your help :)