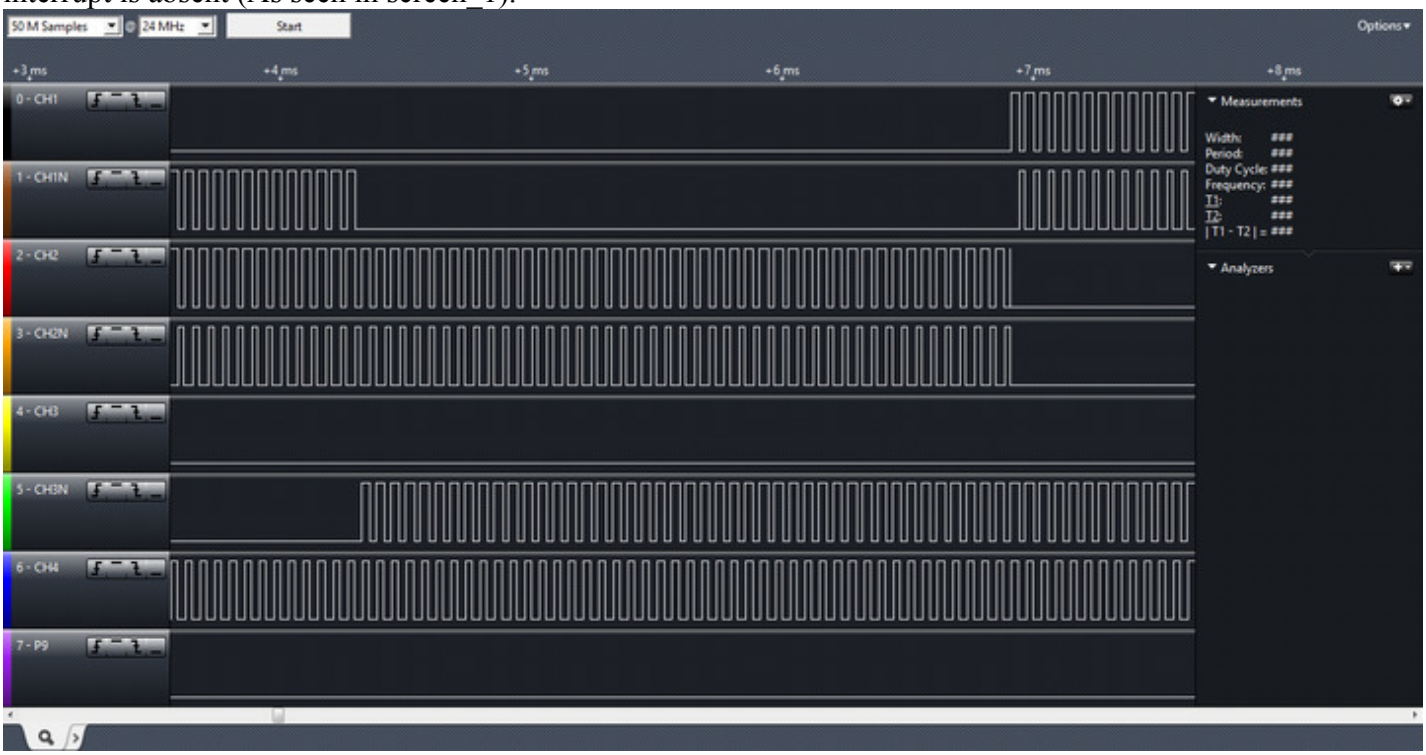
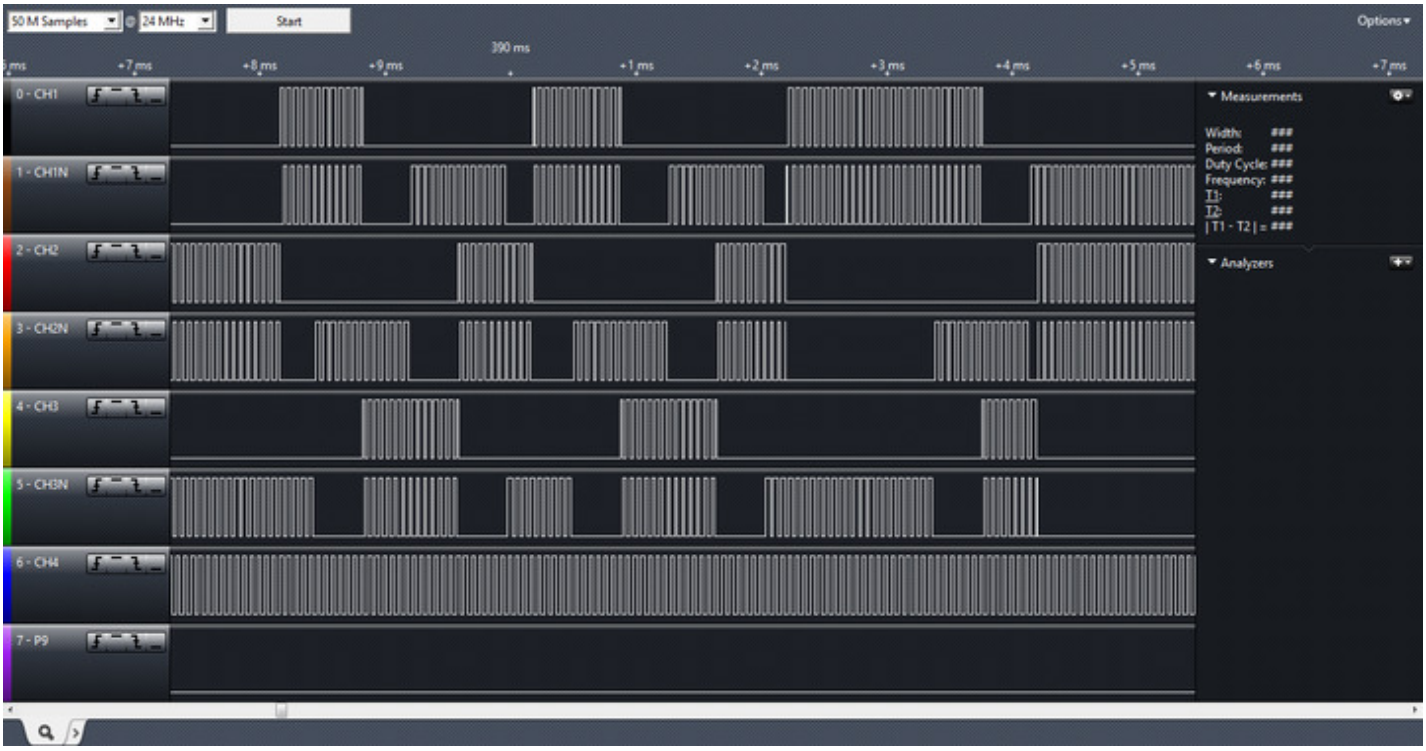
Hello,

I am experiencing unusual circumstance with stm32f103rb Timer1 configure as PWM on all 4 channel.CH1--CH3 complimentary output is enable with dead-time.I have a code that function well when Tim4 CH4 interrupt is absent (As seen in screen_1).



I plan to trigger ADC in Tim4 CH4 ISR. the code malfunction when I enable Tim4 CH4 interrupt (as seen in screen_2 below). The CH1..CH3 and it complimentary output trigger spuriously.it also stop Timer3 counter from incrementing.It took me time to figure it out.The bad news is my code can't do with it.I don't know how to overcome the problem.I will appreciate all help I can get.



TIMER1 CONFIG

==========

#include "stm32f10x.h"

```
#define PWM_FREQ              ((uint16_t)20000)   //PWM Frequency is 17KHz*/
#define DEADTIME_1US            ((uint16_t)0x001E)  //1us Dead Time Insertion



volatile uint16_t PWM_Period;
/*-------------------------------------------------------------------------
 This Function Configure Timer1 in PWM Independent Complementary Mode with
 PWM Frequency of 25KHz.
 *-------------------------------------------------------------------------*/
void PWM_Configuration(void)
 {

  PWM_Period = (SystemCoreClock/PWM_FREQ) - 1;//PWM Freq 20KHz
    TIM1->ARR = PWM_Period;
  TIM1->PSC = 0;

    //Channel4 configuration
    TIM1->CCMR2 |= TIM_CCMR2_OC4M_1 |          //PWM mode 1
                   TIM_CCMR2_OC4M_2;
    TIM1->CCER |= TIM_CCER_CC4E;              //Enable Channel4 Compare

//     TIM1->DIER = TIM_DIER_CC4IE;  // Program malfunction when uncommented

    TIM1->CCR4 = 1000;                    //Increment PWM DutyCycle

    TIM1->BDTR |= TIM_BDTR_MOE |              //Enable PWM Main Output
             TIM_BDTR_OSSR |             //Enable OSSR
             DEADTIME_1US;               //1us DeadTime b/w Complimentary Output

    TIM1->CR2 |= TIM_CR2_CCPC;                    //CCxE, CCxNE and OCxM bits are preloaded
  TIM1->CR1 |= TIM_CR1_CEN |              //Start Timer1
            TIM_CR1_ARPE;               //Automatic Reload Enable
 }

//Function change PWM channel depending on the Step
void Precommutate(void)
{

    if (Step == 1)  //Step 1
        {
        //Channel3 configuration
          TIM1->CCER &= ~TIM_CCER_CC3E;           //Disable Channel3 Compare */
          TIM1->CCER &= ~TIM_CCER_CC3NE;          //Disable Channel3 Complimentary Compare */

          //Channel1 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC1M_1 |       //PWM mode 1 */
                         TIM_CCMR1_OC1M_2;
          TIM1->CCER |= TIM_CCER_CC1E;            //Enable Channel1 Compare O/P */
          TIM1->CCER |= TIM_CCER_CC1NE;           //Disable Channel1 Complimentary Compare O/P */

          //Channel2 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC2M_1 |       //PWM mode 1 */
                         TIM_CCMR1_OC2M_2;
          TIM1->CCER &= ~TIM_CCER_CC2E;           //Disable Channel2 Compare */
          TIM1->CCER |= TIM_CCER_CC2NE;           //Enable Channel2 Complimentary Compare */
        }

   else if (Step == 2)  //Step 2
```

```
        {
        //Channel1 configuration
          TIM1->CCER &= ~TIM_CCER_CC1E;          //Disable Channel1 Compare */
          TIM1->CCER &= ~TIM_CCER_CC1NE;         //Disable Channel1 Complimentary Compare */

          //Channel3 configuration
          TIM1->CCMR2 |= TIM_CCMR2_OC3M_1 |      //PWM mode 1 */
                         TIM_CCMR2_OC3M_2;
          TIM1->CCER |= TIM_CCER_CC3E;           //Enable Channel3 Compare */
          TIM1->CCER |= TIM_CCER_CC3NE;          //Disable Channel3 Complimentary Compare */

          //Channel2 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC2M_1 |      //PWM mode 1 */
                         TIM_CCMR1_OC2M_2;
          TIM1->CCER &= ~TIM_CCER_CC2E;          //Disable Channel2 Compare */
          TIM1->CCER |= TIM_CCER_CC2NE;          //Enable Channel2 Complimentary Compare */
        }

    else if (Step == 3)   //Step 3
        {
        //Channel2 configuration
          TIM1->CCER &= ~TIM_CCER_CC2E;          //Disable Channel2 Compare */
          TIM1->CCER &= ~TIM_CCER_CC2NE;         //Disable Channel2 Complimentary Compare */

          //Channel1 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC1M_1 |      //PWM mode 1 */
                         TIM_CCMR1_OC1M_2;
          TIM1->CCER &= ~TIM_CCER_CC1E;          //Disable Channel1 Compare */
          TIM1->CCER |= TIM_CCER_CC1NE;          //Enable Channel1 Complimentary Compare */

          //Channel3 configuration
          TIM1->CCMR2 |= TIM_CCMR2_OC3M_1 |      //PWM mode 1 */
                         TIM_CCMR2_OC3M_2;
          TIM1->CCER |= TIM_CCER_CC3E;           //Enable Channel3 Compare */
          TIM1->CCER |= TIM_CCER_CC3NE;          //Disable Channel3 Complimentary Compare */
        }

    else if (Step == 4)   //Step 4
        {
        //Channel3 configuration
          TIM1->CCER &= ~TIM_CCER_CC3E;          //Disable Channel3 Compare */
          TIM1->CCER &= ~TIM_CCER_CC3NE;         //Disable Channel3 Complimentary Compare */

          //Channel1 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC1M_1 |      //PWM mode 1 */
                         TIM_CCMR1_OC1M_2;
          TIM1->CCER &= ~TIM_CCER_CC1E;          //Disable Channel1 Compare */
          TIM1->CCER |= TIM_CCER_CC1NE;          //Enable Channel1 Complimentary Compare */

          //Channel2 configuration
          TIM1->CCMR1 |= TIM_CCMR1_OC2M_1 |      //PWM mode 1 */
                         TIM_CCMR1_OC2M_2;
          TIM1->CCER |= TIM_CCER_CC2E;           //Enable Channel2 Compare */
          TIM1->CCER |= TIM_CCER_CC2NE;          //Disable Channel2 Complimentary Compare */
        }

    else if (Step == 5)   //Step 5
        {
        //Channel1 configuration
```

```
        TIM1->CCER &= ~TIM_CCER_CC1E;          //Disable Channel3 Compare */
        TIM1->CCER &= ~TIM_CCER_CC1NE;         //Disable Channel3 Complimentary Compare */

        //Channel3 configuration
        TIM1->CCMR2 |= TIM_CCMR2_OC3M_1 |      //PWM mode 1 */
                       TIM_CCMR2_OC3M_2;
        TIM1->CCER &= ~TIM_CCER_CC3E;          //Disable Channel3 Compare */
        TIM1->CCER |= TIM_CCER_CC3NE;          //Enable Channel3 Complimentary Compare */

        //Channel2 configuration
        TIM1->CCMR1 |= TIM_CCMR1_OC2M_1 |      //PWM mode 1 */
                       TIM_CCMR1_OC2M_2;
        TIM1->CCER |= TIM_CCER_CC2E;           //Enable Channel2 Compare */
        TIM1->CCER |= TIM_CCER_CC2NE;          //Disable Channel2 Complimentary Compare */
      }

 else  //Step 6
      {
      //Channel2 configuration
        TIM1->CCER &= ~TIM_CCER_CC2E;          /* Disable Channel3 Compare */
        TIM1->CCER &= ~TIM_CCER_CC2NE;         /* Disable Channel3 Complimentary Compare */

        //Channel1 configuration
        TIM1->CCMR1 |= TIM_CCMR1_OC1M_1 |      /* PWM mode 1 */
                       TIM_CCMR1_OC1M_2;
        TIM1->CCER |= TIM_CCER_CC1E;           /* Enable Channel1 Compare */
        TIM1->CCER |= TIM_CCER_CC1NE;          /* Disable Channel1 Complimentary Compare */

        //Channel3 configuration
        TIM1->CCMR2 |= TIM_CCMR2_OC3M_1 |      /* PWM mode 1 */
                       TIM_CCMR2_OC3M_2;
        TIM1->CCER &= ~TIM_CCER_CC3E;          /* Disable Channel3 Compare */
        TIM1->CCER |= TIM_CCER_CC3NE;          /* Enable Channel3 Complimentary Compare */
      }

 }
```

Thanks