omar_shishani94@live.com
Fri 31.08.2018, 10:34

Dear ST Customer Support,
Case #:00064565, Subject: Using Timer with STM32-MAT/TARGET

I have investigated further in the problem I'm having with Timer block with STM32-MAT/TARGET. By using the same model, I conclude the following:

In the main.c the 'Infinite loop' 'Real time from systickHandler' is running the function LED_Timer_step(); but this function in LED_Timer.c is empty. (See Figure 1)

```
/* Model step function */
void LED_Timer_step(void)
{
  /* S-Function Block: <Root>/STM32_Config */
}
```

Figure 1

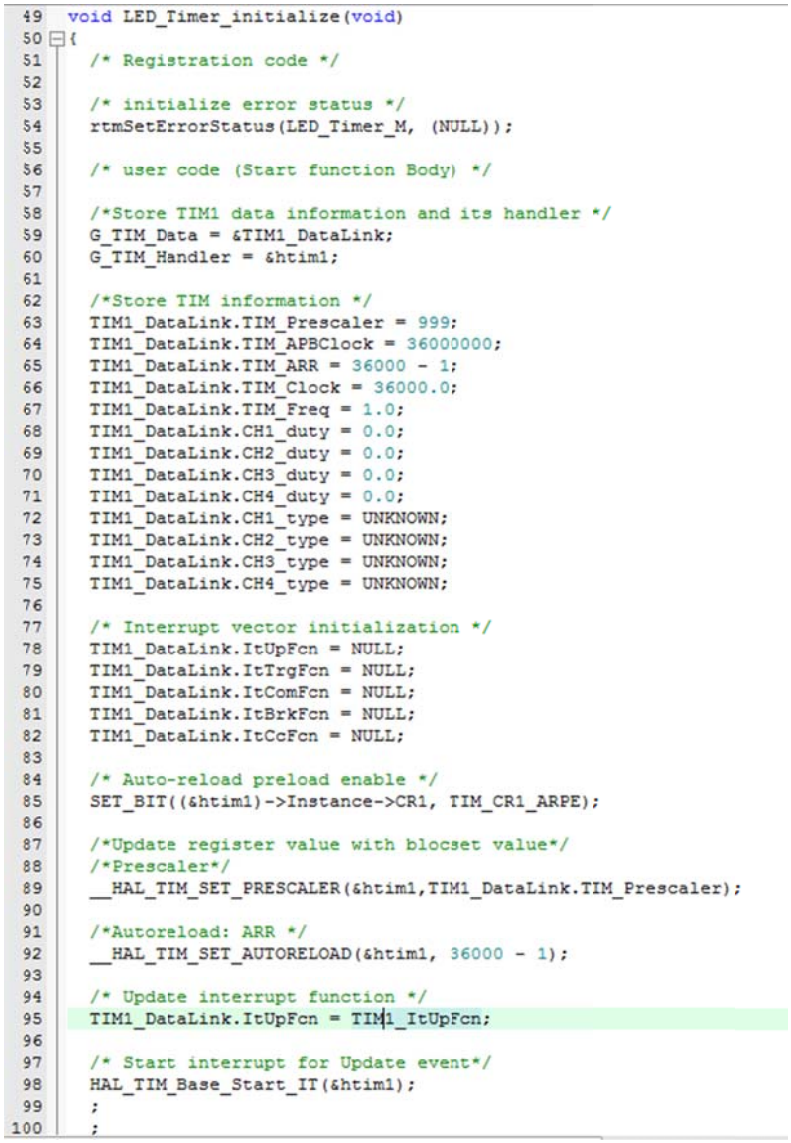
So the function LED_Timer_step(void) is doing nothing, and therefore the executed code in the 'Infinite loop' is doing nothing and the LED is not blinking.

The LED is just turning on because of the function LED_Timer_initialize(); which runs before the 'Infinite loop' . The initialization function is shown in Figure 3.

The main function is shown in figure 2

```
/* Model initialization call */
LED_Timer_initialize();

/* Infinite loop */
/* Real time from systickHandler */
while (1) {
  /*Process tasks every solver time*/
  if (remainAutoReloadTimerLoopVal_S == 0) {
    remainAutoReloadTimerLoopVal_S = autoReloadTimerLoopVal_S;

    /* Check base rate for overrun */
    if (OverrunFlags[0]) {
      rtmSetErrorStatus(LED_Timer_M, "Overrun");
    }

    OverrunFlags[0] = true;

    /* Step the model for base rate */
    LED_Timer_step();

    /* Get model outputs here */

    /* Indicate task for base rate complete */
    OverrunFlags[0] = false;
  }
}
```

Figure 2



```
49  void LED_Timer_initialize(void)
50  {
51    /* Registration code */
52
53    /* initialize error status */
54    rtmSetErrorStatus(LED_Timer_M, (NULL));
55
56    /* user code (Start function Body) */
57
58    /*Store TIM1 data information and its handler */
59    G_TIM_Data = &TIM1_DataLink;
60    G_TIM_Handler = &htim1;
61
62    /*Store TIM information */
63    TIM1_DataLink.TIM_Prescaler = 999;
64    TIM1_DataLink.TIM_APBClock = 36000000;
65    TIM1_DataLink.TIM_ARR = 36000 - 1;
66    TIM1_DataLink.TIM_Clock = 36000.0;
67    TIM1_DataLink.TIM_Freq = 1.0;
68    TIM1_DataLink.CH1_duty = 0.0;
69    TIM1_DataLink.CH2_duty = 0.0;
70    TIM1_DataLink.CH3_duty = 0.0;
71    TIM1_DataLink.CH4_duty = 0.0;
72    TIM1_DataLink.CH1_type = UNKNOWN;
73    TIM1_DataLink.CH2_type = UNKNOWN;
74    TIM1_DataLink.CH3_type = UNKNOWN;
75    TIM1_DataLink.CH4_type = UNKNOWN;
76
77    /* Interrupt vector initialization */
78    TIM1_DataLink.ItUpFcn = NULL;
79    TIM1_DataLink.ItTrgFcn = NULL;
80    TIM1_DataLink.ItComFcn = NULL;
81    TIM1_DataLink.ItBrkFcn = NULL;
82    TIM1_DataLink.ItCcFcn = NULL;
83
84    /* Auto-reload preload enable */
85    SET_BIT((&htim1)->Instance->CR1, TIM_CR1_ARPE);
86
87    /*Update register value with blocset value*/
88    /*Prescaler*/
89    __HAL_TIM_SET_PRESCALER(&htim1,TIM1_DataLink.TIM_Prescaler);
90
91    /*Autoreload: ARR */
92    __HAL_TIM_SET_AUTORELOAD(&htim1, 36000 - 1);
93
94    /* Update interrupt function */
95    TIM1_DataLink.ItUpFcn = TIM1_ItUpFcn;
96
97    /* Start interrupt for Update event*/
98    HAL_TIM_Base_Start_IT(&htim1);
99    ;
100   ;
```

Figure 3

The Matlab is not generating the code correctly, it should build a code for the function LED_Timer_step().

Another point; according to (STM32-MAT/TARGET Hands On Rev 2.2) three files are generated:

1) .mlproject: Generated from MATLAB®. Contains information about .c/.h files to add to project from MATLAB®.
2) .mxproject: Generated from STM32 CubeMX. Contains information about .c/.h files generated from STM32 CubeMX.
3) .script: Generated from MATLAB®. Contains STM32CubeMX command to generate project.

In my case .mlproject is replaced by .extSettings, see figure 4:

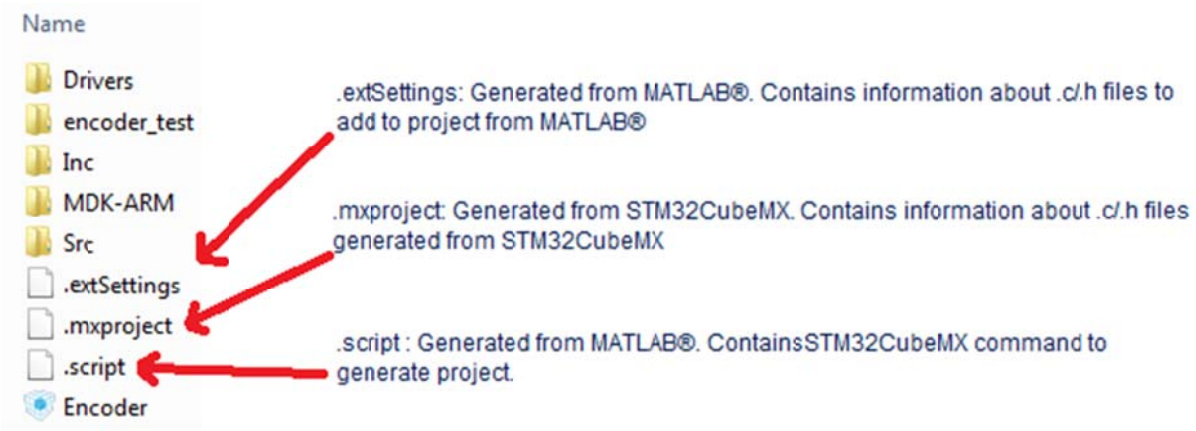


Figure 4

Does this have any effect?

P.S. While working on other blocks from Target Support Package-STM32 Adapter (except TIMERS) the simulation works with no errors.

Do you recommend using a certain version of MATLAB?
Why the code is not generating correctly?

Best Regards

Omar Alshishani