FAQ0121

Frequently Asked Questions

Overflow event in TMR encoder mode

Questions:

Notes on Overflow event in TMR encoder mode

Answer:

In encoder counter mode, when a counter counts back and forth between 0 and PR, no overflow event is generated under an underflow or overflow condition.

For example, set the TMRx_PR (period register) to 0xFFFF.

1. Underflow:

   No Overflow event is generated when the counter value TMRx_CVAL is 0 and the counter counts down to 0xFFFF.

   If counter value TMRx_CVAL is 1 and the counter counts down to 0xFFFF, an overflow event is triggered.

2. Overflow

   When the counter value TMRx_CVAL is 0xFFFF and the counter counts up to 0, no overflow event is generated. If the counter value is TMRx_CVAL=0xFFFE and the counter keeps counting up to 0, an overflow event is then triggered.

   In other words, when the counter value TMRx_CVAL counts back and forth between 0 and 0xFFFF, there will be no overflow event to occur if the counter does not count to 1 or 0xFFFE.

Solution 1:

Set the channel 3 and 4 of encoder TMR as output mode, and set TMRx_C3DT= TMRx_PR and TMRx_C4DT= 0, enable interrupts of channel 3 and 4.

```
/* tmre output configuration */
tmr_output_config_type tmr_oc_init_structure;
tmr_output_default_para_init(&tmr_oc_init_structure);
tmr_oc_init_structure.oc_mode = TMR_OUTPUT_CONTROL_SWITCH;
tmr_oc_init_structure.oc_output_state = FALSE;
tmr_oc_init_structure.occ_output_state = FALSE;
/* output compare toggle mode configuration: channel3*/
tmr_output_channel_config(TMR3, TMR_SELECT_CHANNEL_3, &tmr_oc_init_structure);
tmr_channel_value_set(TMR3, TMR_SELECT_CHANNEL_3, TMR3->pr);
/* output compare toggle mode configuration: channel4*/
tmr_output_channel_config(TMR3, TMR_SELECT_CHANNEL_4, &tmr_oc_init_structure);
tmr_channel_value_set(TMR3, TMR_SELECT_CHANNEL_4, 0x0);

/* nvic configuration */
nvic_priority_group_config(NVIC_PRIORITY_GROUP_4);
nvic_irq_enable(TMR3_GLOBAL_IRQn, 0, 0);
/* tmr it flag clear */
tmr_flag_clear(TMR3, TMR_C3_FLAG|TMR_C4_FLAG);
```

```
/* tmr it enable */
tmr_interrupt_enable(TMR3, TMR_C3_INT | TMR_C4_INT, TRUE);
/*
  If "channel 3 event & count down", it means "underflow event";
  If "channel 4 event & count up", it means "overflow event";
*/
void TMR3_GLOBAL_IRQHandler(void)
{
  if(tmr_flag_get(TMR3, TMR_C3_FLAG) != RESET)
  {
    tmr_flag_clear(TMR3, TMR_C3_FLAG);
    if((TMR3->ctrl1 & 0x10) == TMR_COUNT_DOWN)
    {
      /*printf("down\r\n");*/
    }
  }
 if(tmr_flag_get(TMR3, TMR_C4_FLAG) != RESET)
  {
    tmr_flag_clear(TMR3, TMR_C4_FLAG);
    if((TMR3->ctrl1 & 0x10) == TMR_COUNT_UP)
    {
      /*printf("up\r\n");*/
    }
  }
}
```

Limitation to this method: If the input signal frequency of encoder counter is so fast that it requires software to enter interrupt routine and handle them repeatedly, this may cause some interrupts unable to be handled in time. Therefore this method is applicable to the scenario where an encoder's external input frequency is not so fast.

Solution 2:

Use an enhanced timer (in this mode, TMRx_CVAL, TMRx_PR and TMRx_CxDT is extended from 16 bits to 32 bits) to expand the counting scope of positive and negative rotation of encoder. Setting TMRx_CVAL to TMRx_PR /2 would prevent timer from overflowing.

Taking TMR2 as an example, the following code is used to enable enhanced TMR.

```
/* enable tmr2 32bit function */
tmr_32_bit_function_enable(TMR2, TRUE);
```

Limitation to this method: the positive and negative of encoder counter can only be allowed to run within a certain range. Keep it rotate in a single direction would cause overflow event. Therefore this method is applicable to the scenario where the positive and negative rotation of encoder can be controlled to rotate within a certain range.

Type: MCU application

Applicable products: AT32F4xx series

Main function: TMR encoder

Other function: None

Document revision history

Date	Revision	Changes
2022.3.3	2.0.0	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license grant by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement of any patent, copyright or other intellectual property right.

Purchasers hereby agrees that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any air craft application; (C) any automotive application or environment; (D) any space application or environment, and/or (E) any weapon application. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and is solely responsible for meeting all legal and regulatory requirement in such use

Resale of ARTERY products with provisions different from the statements and/or technical features stated in this document shall immediately void any warranty grant by ARTERY for ARTERY products or services described herein and shall not create or expand in any manner whatsoever, any liability of ARTERY.