Outlines

This application note is a reference material for developing products using the interval timer function in the 32-bit timer event counter (T32A) function of M3H Group(2). This document helps the user check operation of the product and develop its program.

Target sample program: Timer_LED
# Table of Contents

Outlines ................................................................................................................................................. 1  
Table of Contents .................................................................................................................................. 2  
1. Preface .............................................................................................................................................. 3  
2. Reference Document ........................................................................................................................ 4  
3. Function to Use ................................................................................................................................ 4  
4. Target Device .................................................................................................................................... 4  
5. Operation confirmation condition ..................................................................................................... 5  
6. Evaluation Board Setting ................................................................................................................... 6  
7. Operation of Evaluation Board ........................................................................................................ 6  
8. Outline of Timer Event Counter (T32A) function ............................................................................... 7  
9. Sample Program ............................................................................................................................... 8  
  9.1. Initialization ................................................................................................................................... 8  
  9.2. Sample program main operation ..................................................................................................... 8  
  9.3. Output Example of Sample Program ............................................................................................... 9  
    9.3.1. Setting Example of Terminal Software ..................................................................................... 9  
  9.4. Timer setting change ....................................................................................................................... 10  
  9.5. Operating Flow of Sample Program ............................................................................................ 11  
10. Precaution ....................................................................................................................................... 17  
11. Revision History ............................................................................................................................. 17  
RESTRICTIONS ON PRODUCT USE ................................................................................................ 18
1. Preface

This sample program uses Timer A to control LEDs' lighting. Lighting and lights-off of an LED can be switched by Push switch.

Structure diagram of Sample program
2. Reference Document

- Datasheet
  TMPM3H group (2) datasheet Rev2.0 (Japanese edition)
- Reference manual
  32-bit timer/event counter (T32A-B) Rev2.1 (Japanese edition)
- Other reference document
  TMPM3H(2) Group Peripheral Driver User Manual (Doxygen)

3. Function to Use

<table>
<thead>
<tr>
<th>IP</th>
<th>Channel</th>
<th>Port</th>
<th>Function/Operation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit timer event counter</td>
<td>Timer A ch0</td>
<td>PV0 (Input Port) PV1</td>
<td>Interval timer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Input Port)</td>
<td></td>
</tr>
<tr>
<td>Input or output port</td>
<td></td>
<td>PK4 (Output Port) PK5</td>
<td>Input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Output Port) PK6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PK7 (Output Port)</td>
<td></td>
</tr>
<tr>
<td>Asynchronous serial communication circuit</td>
<td>ch0</td>
<td>PA1 (UT0TXDA) PA2</td>
<td>Asynchronous communication with PC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(UT0RXD)</td>
<td></td>
</tr>
</tbody>
</table>

4. Target Device

The target devices of application note are as follows.

<table>
<thead>
<tr>
<th>TMPM3HQFDFG</th>
<th>TMPM3HQFZFG</th>
<th>TMPM3HQFYFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMPM3HPFDFG</td>
<td>TMPM3HPFZFG</td>
<td>TMPM3HPFYFG</td>
</tr>
<tr>
<td>TMPM3HNFDFG</td>
<td>TMPM3HNFZFG</td>
<td>TMPM3HNFYFG</td>
</tr>
<tr>
<td>TMPM3HNFDDFG</td>
<td>TMPM3HNFZDFG</td>
<td>TMPM3HNFYDFG</td>
</tr>
<tr>
<td>TMPM3HMFDHG</td>
<td>TMPM3HMFDG</td>
<td>TMPM3HMFYFG</td>
</tr>
</tbody>
</table>

* This sample program operates on the evaluation board of TMPM3HQFDFG.
If other function than the TMPM3HQ one is checked, it is necessary that CMSIS Core related files (C startup file and I/O header file) should be changed properly.
The BSP related file is dedicated to the evaluation board (TMPM3HQ). If other function than the TMPM3HQ one is checked, the BSP related file should be changed properly.
5. Operation confirmation condition

Used microcontroller         TMPM3HQFDFG
Used board                 TMPM3HQFDFG Evaluation Board (Product of Sensyst)
Unified development environment         IAR Embedded Workbench for ARM 8.11.2.13606
Unified development environment         μVision MDK Version 5.24.2.0
Terminal software           Tera Term V4.96
Sample program            V1100

Evaluation board (TMPM3HQFDFG Evaluation Board) (Top view)

For purchasing the board, refer to the following homepage. (http://www.chip1stop.com/)
6. Evaluation Board Setting

The following pin connections should be done on the evaluation board.

<table>
<thead>
<tr>
<th>CN5</th>
<th>Board function</th>
<th>Through hole No.</th>
<th>Through hole No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED (D10)</td>
<td>33 : PORT_LED0</td>
<td>34 : PK4</td>
<td></td>
</tr>
<tr>
<td>LED (D9)</td>
<td>35 : PORT_LED1</td>
<td>36 : PK5</td>
<td></td>
</tr>
<tr>
<td>LED (D8)</td>
<td>37 : PORT_LED2</td>
<td>38 : PK6</td>
<td></td>
</tr>
<tr>
<td>LED (D7)</td>
<td>39 : PORT_LED3</td>
<td>40 : PK7</td>
<td></td>
</tr>
<tr>
<td>Push SW (S4)</td>
<td>49 : PORT_PSW0</td>
<td>50 : PV0</td>
<td></td>
</tr>
<tr>
<td>Push SW (S5)</td>
<td>51 : PORT_PSW1</td>
<td>52 : PV1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CN12</th>
<th>Board function</th>
<th>Through hole No.</th>
<th>Through hole No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB UART conversion</td>
<td>53 : PA1</td>
<td>54 : USB_TXD</td>
<td></td>
</tr>
<tr>
<td>USB UART conversion</td>
<td>55 : PA2</td>
<td>56 : USB_RXD</td>
<td></td>
</tr>
</tbody>
</table>

7. Operation of Evaluation Board

Push Switch can switch the lighting and the lights-off.

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Push SW (S4): PV0</td>
<td>LED blink stop</td>
</tr>
<tr>
<td>Push SW (S5): PV1</td>
<td>LED blink re-start</td>
</tr>
</tbody>
</table>
8. Outline of Timer Event Counter (T32A) function

T32A is composed of two 16-bit timers that can be used as Timer A and Timer B. Also, it can use a Timer C that is connected to Timer A and Timer B as a 32-bit timer. When using Timer C, Timer A and Timer B cannot be used.

The T32A has an interval timer, event counter, input capture, 2-phase counter input, PPG output, Synchronous Start, and Trigger start/stop functions.

The timer has the following functions:
- 16-bit timer: Timer A and Timer B
- 32-bit timer: Timer C
9. Sample Program

The sample program uses a timer to cycle on and off the LED every second.
The stop and re-start of the function are controlled by the Push Switch.

9.1. Initialization

The following initialization is done after power is supplied.
Initialize each clock setting and initialize port.
The main operation of the sample program is done after every initialization completes.

9.2. Sample program main operation

After the initialization operation, shift to the main function and perform the following initialization.
1: Initialization of BSP (Board Support Package)
2: Initialize the application
3: Initialization of driver
The timer settings in this sample program should be done.
The ch0 of Timer A is used.
UART ch0 setting and LED setting should be done.
The corresponding port of a used LED should be set, and the lighting time is set to the timer.
This sample program assigns PK4 to PK7 to OUTPUT for LED lighting.
The setting value in the timer for the lighting and lights-off times are predetermined as 1-ms, respectively,
which is defined as the following;
\[ p_{\text{timer}}->\text{init.interval} = 1000 \]

The lighting time has been set precisely as the followings;
Timer A generates 1-ms interval.
After 1000 times of the 1-ms interval are counted, the LED is lit.
In this manner, 1-second intervals for the lighting and the lights-off are generated, respectively.
The timer operation starts after every setting completes.
9.3. Output Example of Sample Program

After the sample program starts to operate, the LED lighting interval is displayed.

The displayed image has been specified by "printf". If "General Timer Period" is changed, the real setting time is different from the displayed data.

9.3.1. Setting Example of Terminal Software

The operation of the terminal software (Tera Term) has been checked with the following settings.
9.4. Timer setting change

The change of the timer setting time can be done as the followings.

Timer data change
Count-up count change

Timer data change
"static TXZ_Result driver_initialize(void)" in “main” function is used to set.

```
p_timer->init.interval = 1000
```

The above sets 1 ms.
If “1000 (1 μs*1000)” is changed, the blink time is modified.

Count-up count change
"static void timer_interval_handler(uint32_t id)" in “main” function is used to set.

```
if(count < 1000) { /* 1ms * 1000 = 1sec LED on */
  count++;
  on = 1;
} else if((count >= 1000) && (count < 2000)){ /* 1ms * 1000 = 1sec LED off */
  count++;
  on = 0;
```

If the count value above is changed, the blink time is modified.
9.5. Operating Flow of Sample Program

The operating flows of the sample program are shown in the following:

```
Main

<table>
<thead>
<tr>
<th>ref</th>
<th>BSP Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RAM Initialization</td>
</tr>
<tr>
<td></td>
<td>ref Driver Initialization</td>
</tr>
<tr>
<td></td>
<td>ref Application Initialization</td>
</tr>
<tr>
<td>loop [1, used SW count]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ref SW detection</td>
</tr>
</tbody>
</table>
```
Driver initialization

ref

T32A Driver initialization

T32A Driver initialization

BSP
(Application)

bsp_get_timer_ch(BSPTimer)

Channel number = bsp_get_timer_ch(-)

t32a_mode_init(Instance address)

result = t32a_mode_init(-):success

t32a_timer_init(Instance address)

result = t32a_timer_init(-):success

timer_initialize(Timer Instance address)

result = timer_initialize(-):success

Acquisition of Timer channel for 1-ms timer
Application initialization

- UART Application initialization
  - loop
    - [1, used LED count]
      - LED Application initialization
    - loop
      - [1, used SW count]
        - SW Application initialization

SW Application initialization

- SW Initial setting creation
  - sw_initialize(SW Instance address)

SW (Application)
LED Application initialization

LED (Application)

led_initialize (LED Instance address)

GPIO (Driver)

gpio_write_bit (GPIO Instance address, Group, Number, DATA, LED initial setting value)

led_initialize(-)

Initial setting creation
Timer interrupt

Timer interrupt

Procedure result:
Count less than 1000

Procedure result:
Count 1000 or more and less than 2000

Count reset

Timer irq_handler (Timer Instance address)

timer_interval_handler(id)

led_turn_on(Instant handle)

led_turn_off(Instant handle)

Procedure result:
Count 1000 or more and less than 2000

BSP (Application)

T32A (Application)

LED

led_turn_on(Instant handle)

led_turn_off(Instant handle)

Count reset

INTT32AxxA_IRQHandler

t32a_timer_IRQHandler()
Switch (SW) detection

**SW detection**

- `sw` (Application)
- `gpio` (Driver)

**sw_task**:

- **SW status check flag=ON**
- `gpio_read_bit` (Port instance address, group, num, DATA)

- **Port state changed**
- **Port status=Push**

- Assigned to Push SW
  - Processing = Stop / Start

- Registration handler for notification of SW status change
  - (SW instance address, Port status)

- Registration handler for notification of SW status change(-)

- sw_task(-)
10. Precaution

When using the sample program with CPU other than TMPM3HQ, please check operation sufficiently.

11. Revision History

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2018-05-18</td>
<td>-</td>
<td>First release</td>
</tr>
</tbody>
</table>
RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as "TOSHIBA".
Hardware, software and systems described in this document are collectively referred to as "Product".

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.

- PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE"). Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT. For details, please contact your TOSHIBA sales representative.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.