Freescale Semiconductor
Application Note

Document Number: AN4160
Rev. 0, 8/2010

MTIM Driver for the MC9S08GW64

by: Tanya Malik
Reference Design and Applications Group
India
IDC MSG NOIDA

Contents

1 Introduction

This document describes a driver for the Modulo Timer (MTIM) allowing customization of all the possible configurations for this peripheral.

The software architecture is designed to provide seamless migration between devices that have the same peripheral module.

In this application note, the driver interfaces are explained. Various applications for the MC9S08GW64 can make use of this driver. The following sections describe in detail and the steps for creating an application using the MTIM driver.

In the MC9S08GW64 there are two 8-bit modulo timers (MTIM1 and MTIM2) and one 16-bit modulo timer (MTIM3).

The bus clock to the MTIM1, MTIM2, and MTIM3 modules can be gated on and off using the MTIM1, MTIM2, and MTIM3 bits in the SCGC4 register.

2 Software Driver Description

The MTIM driver is provided as some C code files. You can add these files to your applications. With the integration of the MTIM driver, you can now call MTIM driver APIs to use the MTIM functionality in your application. There are four files associated with the MTIM driver. The following is a brief description:

mtim8_timer.h—It contains all the high level APIs declarations and the various macros to be used in the functions for the 8-bit modulo timer

mtim8_timer.c—It is the main file for the driver of the 8-bit modulo timer. It contains the various high level API definitions.

mtim16_timer.h—It contains all the high level APIs declarations and the various macros to be used in the functions for the 16-bit modulo timer.

mtim16_timer.c—It is the main file for the driver of the 16-bit modulo timer. It contains the various high level API definitions.

2.1 mtim8_timer.c

It contains the definition of various functions to configure and use the various features of the 8-bit MTIM.

2.1.1 MTIM1_Init

Description:

This function initializes MTIM1 by configuring the internal registers. The fixed frequency clock is selected for the MTIM1 and the clock is divided by 16. Thus, the MTIM1 is initialized to run at 1 Khz.

Prototype:

void MTIM1_Init(void *p1(unsigned char))

Input parameters:

- p1—The address of the callback function is passed in this argument. The callback function is the function called when the interrupt occurs. If the user does not want to use any callback function then "0" has to be passed in this argument.
- char— It is the argument passed in the callback function. It specifies which interrupt occurred, whether it is the MTIM1 interrupt, MTIM2 interrupt, or MTIM3 interrupt.

Output parameters:

None

Example:

```
void callback()
{
}
MTIM1_Init(&callback);
```

2.1.2 Enable_MTIM1

Description:

This function is used to enable the MTIM1 interrupt, and make the MTIM1 counter active.

Prototype:

void Enable_MTIM1(void)

Input parameters:

None

Output parameters:

None

Example:

Enable_MTIM1();

2.1.3 Disable_MTIM1

Description:
This function is used to disable the MTIM1 interrupt, and stop the MTIM1 counter.

Prototype:
void Disable_MTIM1(void)

Input parameters:
None

Output parameters:
None

Example:
Disable_MTIM1();

2.1.4 Set_TimeOut_MTIM1

Set_TimeOut_MTIM1
This function enables the MTIM1 counter and configures the MTIM1 to get the the required delay. The callback function is called after the delay given as input.

Prototype:
void Set_TimeOut_MTIM1(unsigned int num)

Input parameters:
num—Enter the milliseconds, after which the interrupt is required.

Output parameters:
None

Example:
Set_TimeOut_MTIM1(2000)
Provides a delay of two seconds.

2.1.5 Interrupt Subroutine for MTIM1

The interrupt subroutine is executed when the MTIM1 counter value matches the mode value provided by the user. If the user has passed an address to the callback function, then the code jumps to the callback function after clearing the interrupt. It then passes an argument to the callback function indicating that the MTIM1 interrupt has occurred and the user can take the required action.

Prototype:
void interrupt VectorNumber_Vmtim1 MTIM1_ISR()

NOTE

The functions are similar for MTIM2:

MTIM2_Init—Refer to MTIM1_Init
Enable_MTIM2—Refer to Enable_MTIM1
Disable_MTIM2—Refer to Disable_MTIM1
Set_TimeOut_MTIM2—Refer to Set_TimeOut_MTIM1
Interrupt subroutine for MTIM2—Refer to Interrupt Subroutine for MTIM1

2.2 mtim16_timer.c

This function contains the definition of various functions to configure and use the various features of the 16-bit MTIM.

2.2.1 MTIM3_Init

Description:

This function initializes MTIM3 by configuring the internal registers. The fixed frequency clock is selected for the MTIM3, and the clock is divided by 16. Thus, the MTIM3 is initialized to run at 1 Khz.

Prototype:

void MTIM3_Init(void *p1(unsigned char))

Input parameters:

- p1—The address of the callback function is passed in this argument. The callback function is the function called when the interrupt occurs. If the user does not want to use any callback function then "0" has to be passed in this argument.
- char—Is the argument passed in the callback function. It specifies which interrupt occurred, whether it is the MTIM1 interrupt, MTIM2 interrupt, or MTIM3 interrupt.

Output parameters:

None

Example:

```
void callback()
{
}
MTIM3_Init(&callback);
```

2.2.2 Enable_MTIM3

Description:

This function is used to configure the MTIM3 for the required delay, enables the MTIM3 interrupt, and makes the counter active.

Prototype:

void Enable_MTIM3(unsigned int num)

Input parameters:

num—Enter the delay in milliseconds, after which an MTIM3 interrupt is required.

Output parameters:

None

Example:

Enable_MTIM3(3000);

Enables the MTIM3 counter for a delay of 3 seconds.

2.2.3 Disable_MTIM3

Description:

This function is used to disable the MTIM3 interrupt and stop the MTIM3 counter.

Prototype:

void Disable_MTIM3(void)

Input parameters:
None

Output parameters:
None

Example:
Disable_MTIM3();

2.2.4 Interrupt Subroutine for MTIM3

The interrupt subroutine is executed when the MTIM3 counter value matches the mode value provided by the user. If the user has passed an address to the callback function, the code then jumps to the callback function after clearing the interrupt. It passes an argument to the callback function indicating that an MTIM3 interrupt has occurred and the user can take the required action.

Prototype:
void interrupt VectorNumber_Vmtim3 MTIM3_ISR()

2.2.5 Assumptions

The descriptions in this document assumes the person reading it has full knowledge of all the configuration registers of all the blocks in the MC9S08GW64, especially for the MTIM and Internal Clock Source (ICS) blocks.

2.2.6 Use Case

Assuming that the clock settings are done and the bus clock is running on 20 Mhz, include mtim8_timer.h and mtim16_timer.h in the main file.

Step 1—Define a callback function and initialize the MTIM1 with the required configuration.

```
Void mtim_callback()
{
}
MTIM1_Init(&mtim_callback);
```

Step 2—Configure the MTIM1 timer with the delay required
Set_TimeOut_MTIM1(2000);
Configures a delay of 2 seconds

The code jumps to the callback function after two seconds and the user can take the required action in the callback function.

2.2.7 Conclusion

This driver provides a software base for applications that need the MTIM implementation.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and sofware implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor prodcuts are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.