

# **Generating Operating System Tick Using RTI on a Hercules™ ARM™ Safety Microcontroller**

*Hari Udayakumar*

## **ABSTRACT**

The purpose of this application report is to provide help for setting up the RTI Module of Hercules Safety Microcontrollers. The Hercules family of microcontrollers from Texas Instruments is a family of 32-bit RISC microcontrollers with an advanced safety architecture and a rich peripheral set.

## **Contents**

1	Introduction .....	2
2	Operating System Tick Generation .....	3
Appendix A	Software Listing .....	9

## **List of Figures**

1	RTI Module Block Diagram .....	2
2	RTI Counter Block (x) Diagram .....	3
3	Operating System Tick Generation Block Diagram .....	3
4	RTI Clock Source Selection and Pre-Scaling .....	4
5	Operating System Tick Output Waveform .....	5
6	Software Flow Diagram .....	6
7	Output Waveform .....	7

Hercules is a trademark of Texas Instruments.  
ARM is a trademark of ARM Limited.  
All other trademarks are the property of their respective owners.

# 1 Introduction

## 1.1 Real Time Interrupt

The Real Time Interrupt Module (RTI) provides timer functionality for operating systems and for benchmarking code. The module incorporates multiple counters, which define the timebases needed for scheduling in the operating system.

This document needs to be referenced along with the Hercules Safety MCU (RM4x, TMS570LS31x/21x, TMS570LS20x/10x, TMS470M) RTI User's Guide found in the the appropriate Technical Reference Manual.

### Main Features

- Up to two independent counter blocks for generating different timebases. Each block consists of:
  - One 32-bit prescale counter
  - One 32-bit free running counter
- Up to two time stamp (capture) functions for system or peripheral interrupts, one for each counter block
- Free running counter 0 can be incremented by either the internal prescale counter or by an external event (for application synchronization to FlexRay network including clock supervision)
- Optional External Clock supervising circuit to switch to internal prescale counter 0, if external clock source fails to increment in a predefined window
- Four configurable compare registers for generating operating system ticks or DMA requests. Each event can be driven by either counter block 0 or counter block 1
- Automatic update of all compare registers on compare match to generate periodic interrupts
- Fast enabling/disabling of interrupts
- RTI clock input derived from any of the available clock sources, selectable in the System Module

## 1.2 RTI Module Block Diagram

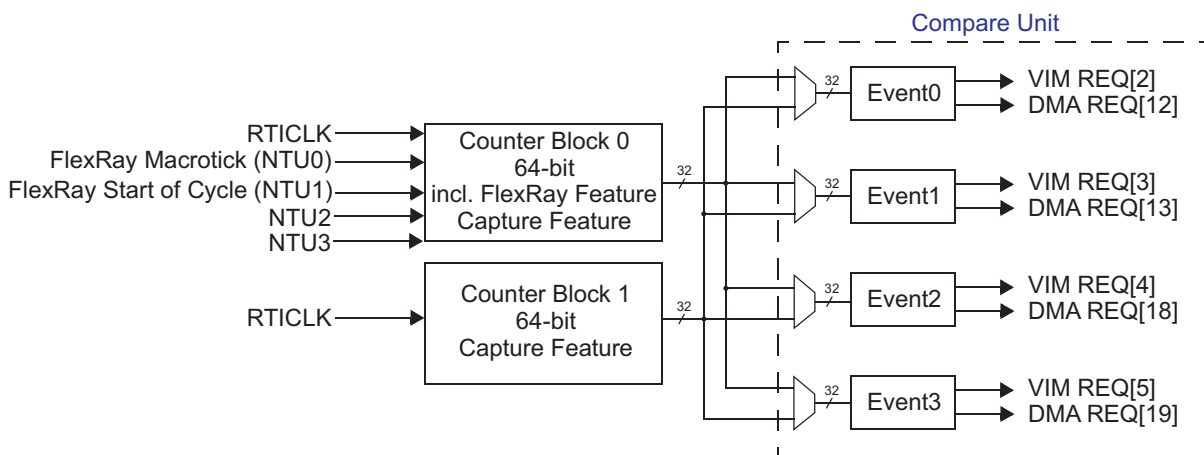


Figure 1. RTI Module Block Diagram

### 1.3 RTI Counter Block (x) Diagram

Figure 2 shows the simplified block diagram of the RTI Counter Block (x). The RTICLK is prescaled by 32 bit RTIUPCx Counter. The RTIUPCx Counter counts up until the compare value in the RTICPUCx register is reached. When the compare matches, RTIFRCx counter is incremented.

$$\begin{aligned} \text{If CPU}_x = 0: \quad FRC_x &= \frac{RTICLK}{2^{32}} \\ \text{If CPU}_x \neq 0: \quad FRC_x &= \frac{RTICLK}{CPU_x + 1} \end{aligned} \tag{1}$$

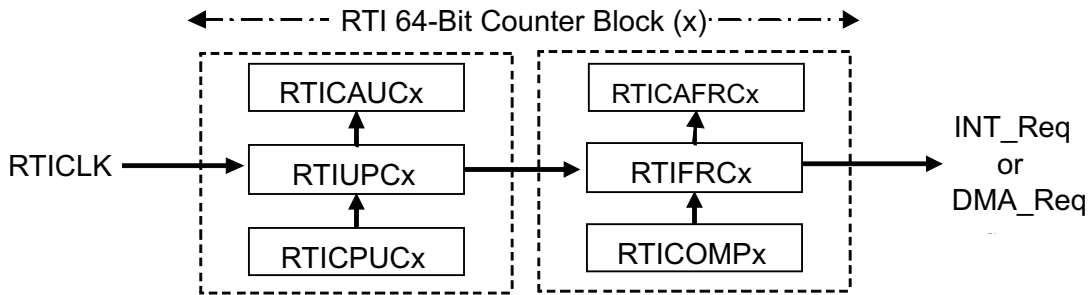


Figure 2. RTI Counter Block (x) Diagram

## 2 Operating System Tick Generation

This section describes how to use the RTI Module to generate Operating system tick interrupts with tunable period. Three compare interrupts are used to generate the required Operating system tick and the corresponding task is carried out. To visualize the task call, individual I/O pins are toggled in the example code. This application report shows how to configure three different interrupts to trigger tasks with different timebases.

### 2.1 Flow Diagram

Figure 3 shows how the RTI Module triggers tasks with different timebases using interrupts.

- Compare 0 Interrupt – Task1 (GIO- PortA [0] Pin Toggle) - 10mS.
- Compare 1 Interrupt – Task2 (GIO- PortA [1] Pin Toggle) - 5mS.
- Compare 2 Interrupt – Task3 (GIO- PortA [2] Pin Toggle) - 1mS.

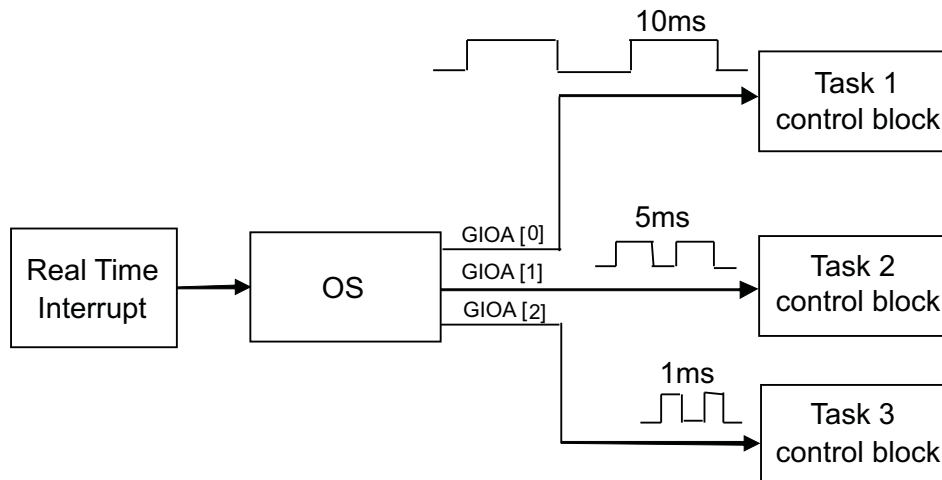
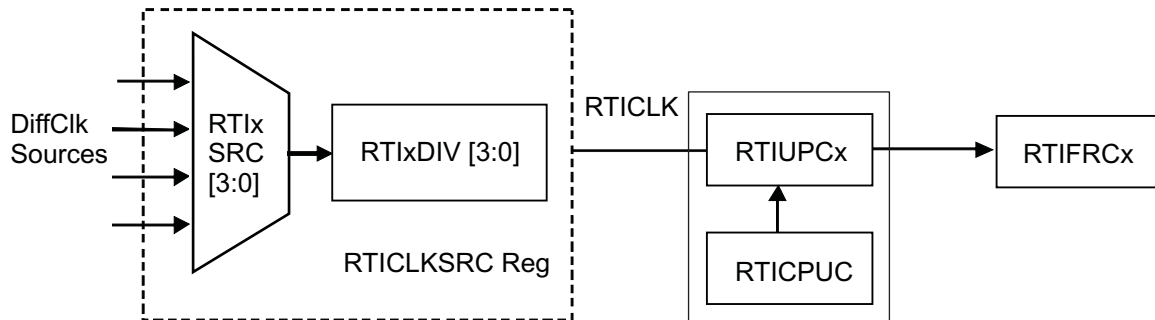


Figure 3. Operating System Tick Generation Block Diagram

## 2.2 Frequency and Tcount Calculation

The RTICK to the counter block can be programmed using the RTICKSRC register in the system module according to the application requirements. If RTICK source is anything other than VCLK, then it needs to be at least three times slower than the VCLK. This can be achieved by configuring the RTixDIV bits in RTICKSRC register. Figure 4 shows RTICK selection and pre-scaling using RTICKSRC register and RTICPUCx register.



**Figure 4. RTI Clock Source Selection and Pre-Scaling**

In the following example, the RTICK is pre-scaled by 2 using the RTICPUCx Register. For more information, see Section 1.3.

$$\begin{aligned}
 \bullet \text{ RTICK} &= \text{VCLK} = 8 \text{ MHz} \\
 \bullet \text{ FRC0CLK} &= \frac{\text{RTICK}}{\text{CPUC0} + 1} = \frac{8 \text{ MHz}}{1 + 1} = 4 \text{ MHz} \\
 \bullet \text{ Tcount} &= \frac{1}{\text{FRC0CLK}} = \frac{1}{4 \text{ MHz}} = 0.25 \mu\text{s}
 \end{aligned}
 \tag{2}$$

## 2.3 Compare Count Value Calculation

The Count values for the Compare register to generate Periodic Operating system tick is calculated with the knowledge of Tcount Period. The RTICK is pre-scaled by the UP Counter (x) Register and fed to the respective Free Running (x) counter.

- Task1 = T1period = 10ms
- Task2 = T2period = 5ms
- Task3 = T3period = 1ms

In the following example, interrupts (INT0, INT1, and INT2) are used for the generation for period system tick.

$$\begin{aligned}
 \bullet \text{ Compare 0 Count Value (Task1)} &= \frac{T1\text{period}}{T\text{count}} = \frac{10 \text{ ms}}{0.25 \mu\text{s}} = 40000 \\
 \bullet \text{ Compare 1 Count Value (Task2)} &= \frac{T2\text{period}}{T\text{count}} = \frac{5 \text{ ms}}{0.25 \mu\text{s}} = 20000 \\
 \bullet \text{ Compare 2 Count Value (Task3)} &= \frac{T3\text{period}}{T\text{count}} = \frac{1 \text{ ms}}{0.25 \mu\text{s}} = 4000
 \end{aligned}
 \tag{3}$$

## 2.4 Operating System Tick Output Waveform

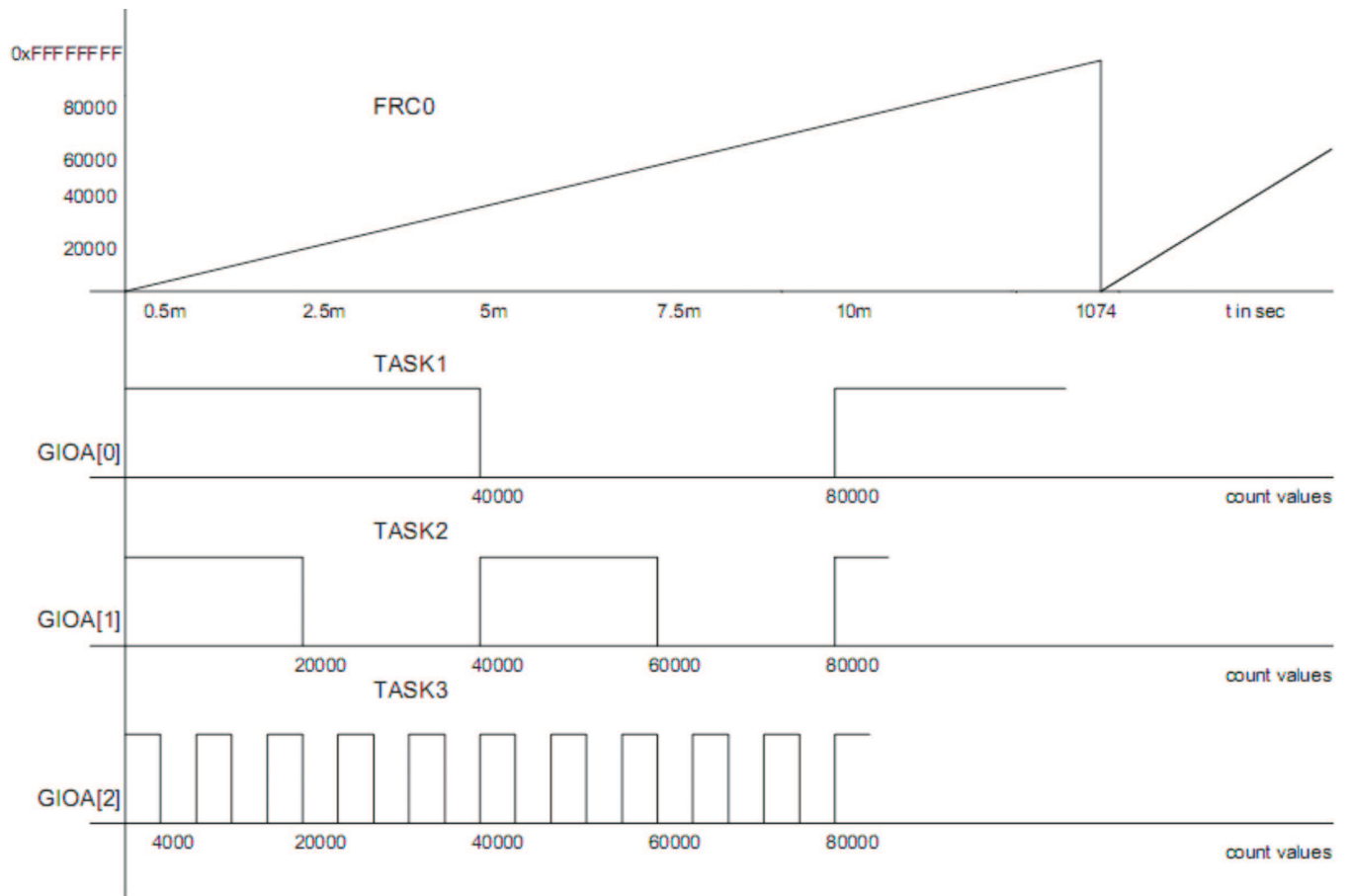
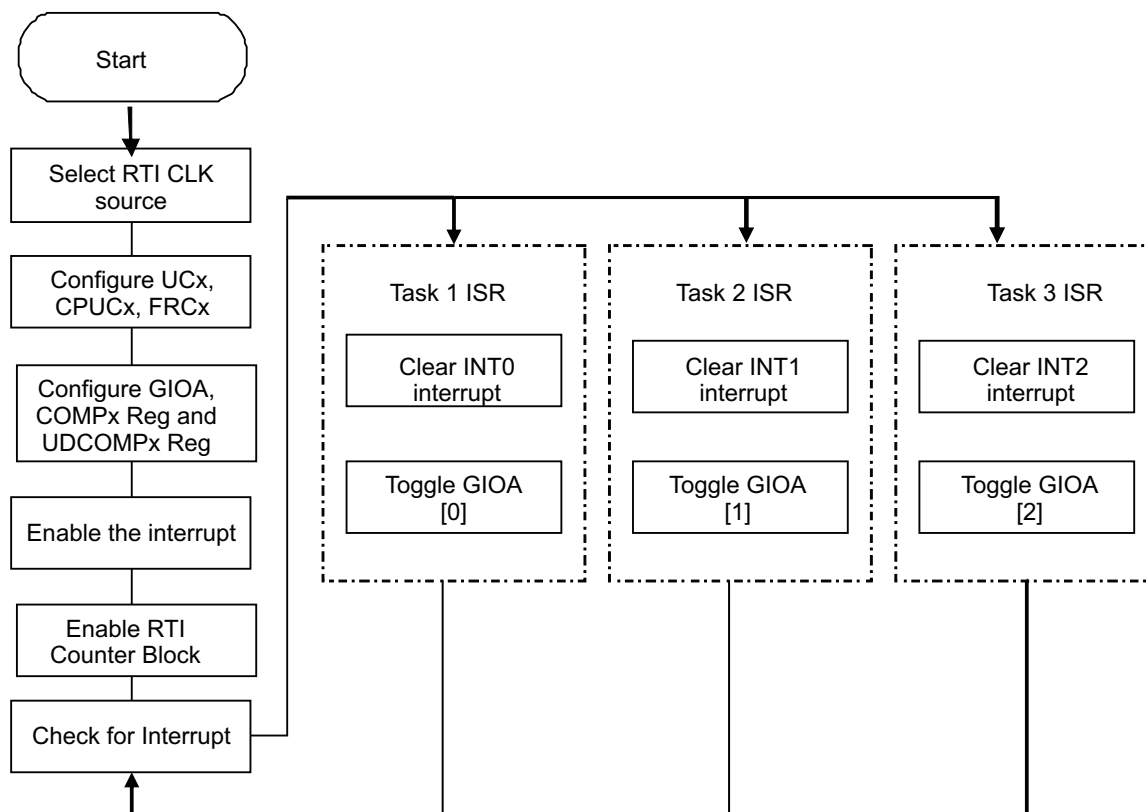


Figure 5. Operating System Tick Output Waveform

## 2.5 Configuration for Generating Operating System Tick

- Select VCLK using RTI clock source using RTICLKSRC register in system module.
- Select the following Clock Configurations.
  - $HCLK = OSCIN$
  - $VCLK = \frac{HCLK}{2}$
  - $RTICLK = VCLK$(4)
- Configure RTICOMP(x) register and the respective Update RTIUDCP(x) Register to initialize with the Compare (x) count Values.
- Configure the GIO Port A as output port.
- Configure RTI module Interrupt i.e. RTISETINT and Status Register i.e. RTIINTFLAG.
  - Pending interrupts are cleared.
  - Compare 0, Compare 1 and Compare 2 interrupt are enabled.
- Configure Compare 0 Interrupt ISR to do the following:
  - Comp0 interrupt flag is cleared by writing 1 to INTO bit in RTIINTFLAG register.
  - Toggle GIO Port A – Pin 0
- Configure Compare 1 Interrupt ISR to do the following:
  - Comp1 interrupt flag is cleared by writing 1 to INT1 bit in RTIINTFLAG register.
  - Toggle GIO Port A – Pin 1
- Configure Compare 2 Interrupt ISR to do the following:
  - Comp2 interrupt flag is cleared by writing 1 to INT2 bit in RTIINTFLAG register.
  - Toggle GIO Port A – Pin 2

## 2.6 Software Flow



**Figure 6. Software Flow Diagram**

### 2.7 Captured Output Waveform

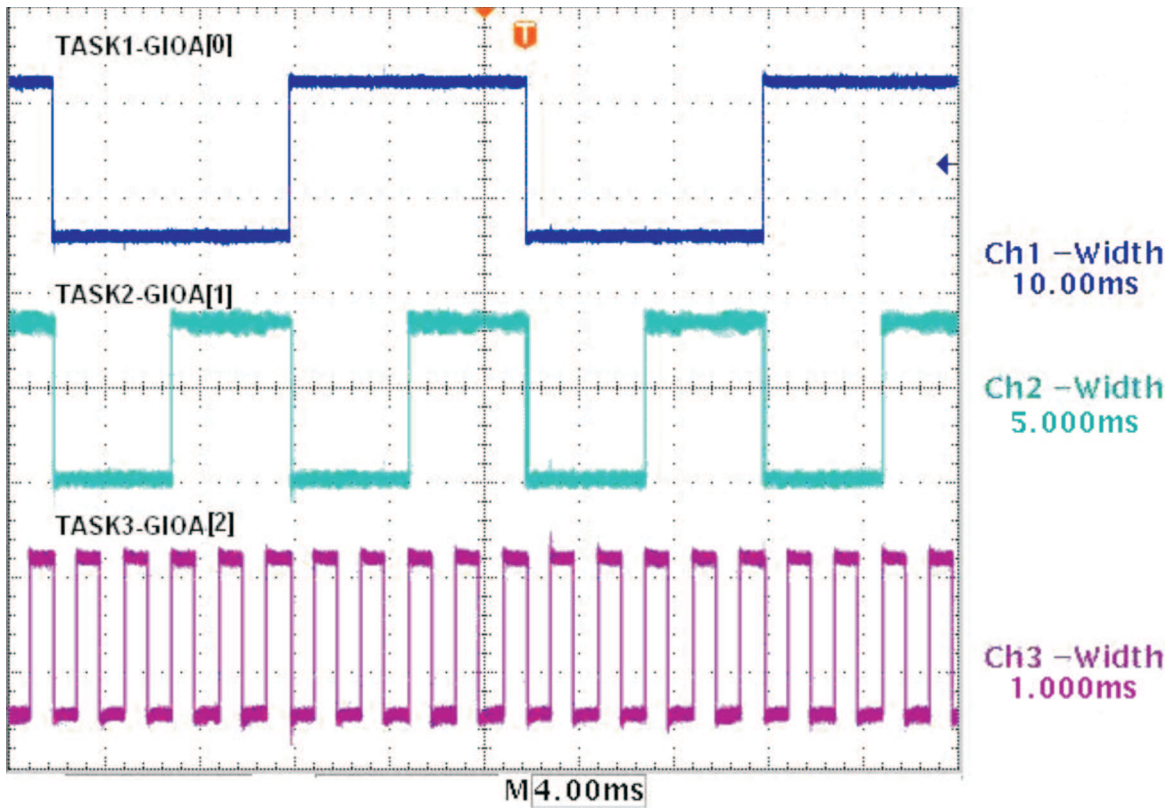


Figure 7. Output Waveform





## Appendix A Software Listing

```

/*****
***/
/* file:RTI1_OST.c */
/*
/* RTI generates Periodic operating system tick using compare interrupts */
/*
/* Purposes: Generate periodic operating system tick using compare interrupts */
/* Conditions: PowerOnReset must be applied prior to run test */
/*
/*****
***/
#include "rti.h"
#include "vim.h"
#include "pcr.h"
#include "system.h"
#include "swi_util.h"
#include "device.h"
#include "module.h"
#include "gio470.h"
SYSTEM_ST *SYS_Ptr = (SYSTEM_ST *) SYSTEM ;
PCR_ST *PCR_Ptr = (PCR_ST *) PCR;
VIM_ST *VIM_Ptr = (VIM_ST *) VIM;
VIM_RAM_ST *VIM_RAM_Ptr = (VIM_RAM_ST *) VIM_RAM;
RTI_ST *RTI_Ptr = (RTI_ST *) RTI;
GIO_ST *GIO_Ptr = (GIO_ST *) GIO1;
#define TASK1 40000
#define TASK2 20000
#define TASK3 4000
#define END_VALUE 5000
main()
{
/***** VIM
Initialization*****/
VIM_RAM_Init(); //VIM initialization
RTI_irq_int_enable(); //Enabling the RTI Interrupts in VIM
swi_enable_irq() ; //Int_irq_enable Enable only IRQ
/***** GIO PortA
Initialization*****/
GIO_Ptr->Gcr0_UN.Gcr0_ST.Reset_B1=1; //enabling GIO
GIO_Ptr->Port_ST[0].Dir_UL=0x07; //PortA.0,1,2 has output
GIO_Ptr->Port_ST[0].Dout_UL=0x00;
/*****RTI Clk Source
Initialization*****/
SYS_Ptr->RCLKSRC_UN.RCLKSRC_ST.RTI1SRC_B4 = 8; //clock source 8 i.e Vclk
SYS_Ptr->RCLKSRC_UN.RCLKSRC_ST.RTI1DIV_B2 = 0; //div the clk source 0 by 1
/*****RTI
Initialization*****/
RTI_Ptr->RTIGCTRL_UN.RTIGCTRL_UL= 0x00000000; //Disable RTIUC0 and RTIUC1
RTI_Ptr->RTIUC0_UL= 0x00000000; //Initialize up Counter
RTI_Ptr->RTIFRC0_UL= 0x00000000; //Initialize Free Running Counter
RTI_Ptr->RTICPUC0_UL= 0x00000001; //Set Compare Up Counter value to Prescale RTICLK.
/*****TASK
Initialization*****/
RTI_Ptr->RTICOMP0_UL = TASK1;
RTI_Ptr->RTIUDCP0_UL = TASK1;
RTI_Ptr->RTICOMP1_UL = TASK2;
RTI_Ptr->RTIUDCP1_UL = TASK2;
RTI_Ptr->RTICOMP2_UL = TASK3;
RTI_Ptr->RTIUDCP2_UL = TASK3;
RTI_Ptr->RTIINTFLAG_UN.RTIINTFLAG_UL = 0x0000000F; //Clearing the pending Interrupts Flags
RTI_Ptr->RTICOMPCTRL_UN.RTICOMPCTRL_UL = 0x00000000; //Initializing the Compare Counter

```

```

register
RTI_Ptr->RTISETINT_UN.RTISETINT_UL = 0x00000007;    //enabling RTI Interrupt in RTI
RTI_Ptr->RTIGCTRL_UN.RTIGCTRL_UL = 0x00000001;    //Enable the RTIUC0
/*****Infinite
loop*****/
while(1);
}
/*****ISR.c*****/
****/
#include "rti.h"
#include "vim.h"
#include "gio470.h"
#pragma INTERRUPT (Compare0_Handler, IRQ)
#pragma INTERRUPT (Compare1_Handler, IRQ)
#pragma INTERRUPT (Compare2_Handler, IRQ)
void Compare0_Handler(void);
void Compare1_Handler(void);
void Compare2_Handler(void);
void Compare0_Handler(void)
{
RTI_Ptr->RTIINTFLAG_UN.RTIINTFLAG_UL= 0x00000001;    //Clearing the Interrupt Flag 0
GIO_Ptr->Port_ST[0].Dout_UL ^= 0x01;                //Toggling the GIOA[0] Port
}
void Compare1_Handler(void)
{
RTI_Ptr->RTIINTFLAG_UN.RTIINTFLAG_UL= 0x00000002;    //Clearing the Interrupt Flag 1
GIO_Ptr->Port_ST[0].Dout_UL ^= 0x02;                //Toggling the GIOA[1] Port
}
void Compare2_Handler(void)
{
RTI_Ptr->RTIINTFLAG_UN.RTIINTFLAG_UL= 0x00000004;    //Clearing the Interrupt Flag 2
GIO_Ptr->Port_ST[0].Dout_UL ^= 0x04;                //Toggling the GIOA[2] Port
}
/*****END*****/
**/

```

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Transportation and Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2011, Texas Instruments Incorporated