Experiment 2
Hibernation and Wakeup on RTC Interrupt

Topics **Page**

Objective

The main objective of this experiment is to understand and configure the Hibernation module of the TM4C123GH6PM device to place the device in a low power state and then to wake up the device on RTC (Real-Time Clock) interrupt.

Introduction

The TM4C123GH6PM consists of a battery-backed hibernation module that provides logic to switch power off to the main processor and its peripherals while the processor is idle. In this experiment, the hibernation module of the TM4C123GH6PM processor is turned on by the software code. This is indicated by the green LED connected to a GPIO port (PF3). When the LED is ON, the device is in wake up mode, and when the LED is OFF the device is in hibernate mode and the Hibernation module is enabled. The device in hibernation can be woken up by two ways:

- An external signal, SW2 input to the GPIO wake up pin
- RTC wake up
- In this experiment, the RTC is used to wake up the processor after 5 seconds. The RTC is used in the RTC Match-Seconds mode. (Refer to TM4C123GH6PM Data Sheet for more details).

The functional block diagram as shown in **Figure 2-1** illustrates the working principle of the experiment.

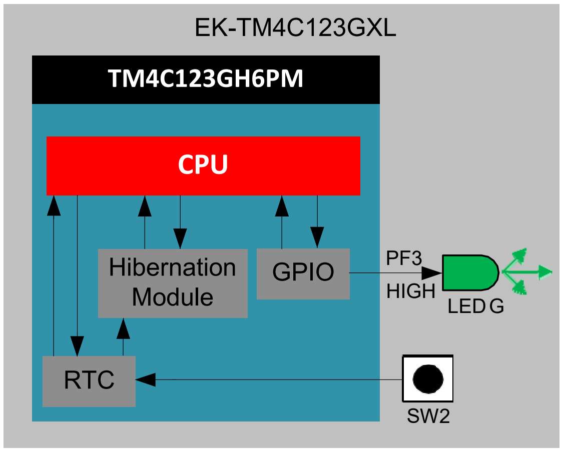


Figure 2-1 Functional Block Diagram

Hibernation Module

The Hibernation Module manages removal and restoration of power to the device and hence helps reduce system power consumption. When the processor and peripherals are idle, power can be completely cut off with only the Hibernation module powered up. Power can be restored based on an external signal or after a certain time using the built-in Real-Time Clock (RTC). Power for the Hibernation module can be independently supplied from an external battery or an auxiliary power supply.

Once in hibernation, the module signals an external voltage regulator to turn the power back on when an external pin (WAKE) is asserted or when the internal RTC reaches a certain value. The

Hibernation module can also detect when the battery voltage is low and optionally prevent hibernation or wake from hibernation when the battery voltage falls below a certain threshold.

The Hibernation module of the Tiva TM4C123GH6PM MCU has the following features:

- 32-bit real-time seconds counter (RTC) with 1/32,768 second resolution and a 15-bit sub-seconds counter
  - 32-bit RTC seconds match register and a 15-bit sub seconds match for timed wake-up and interrupt generation with 1/32,768 second resolution
  - RTC predivider trim to make fine adjustments to the clock rate
- Two mechanisms for power control
  - System power control using discrete external regulator
  - On-chip power control using internal switches under register control
- Dedicated pin to wake up the processor using an external signal
- RTC operational and hibernation memory valid as long as V_{DD} or V_{BAT} is valid
- Low-battery detection, signaling, and interrupt generation, with optional wake on low battery
- GPIO pin state can be retained during hibernation
- Clock source from a 32.768-kHz external crystal or oscillator
- Sixteen 32-bit words of battery-backed memory to save state during hibernation
- Programmable interrupts for:
  - RTC match
  - External wake
  - Low battery

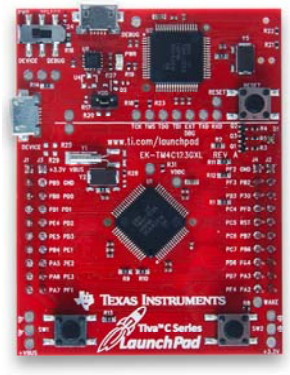
Component Requirements

Software Requirement

1. Keil
2. TivaWare_C_Series

Hardware Requirement

Table 2-1: Components Required for the Experiment

S.No	Components	Specification	Image
1.	Tiva LaunchPad	EK-TM4C123GXL LaunchPad	
2.	USB Cable		

Software

The software for the experiment is written in C and developed using the CCS Integrated Development Environment (IDE). Refer to "**Project Creation and Build**" in the Getting Started section of this manual for project build and debug using CCS. The software is programmed into the target device TM4C123GH6PM on the EK-TM4C123GXL using the USB interface.

Flowchart

The flowchart for the experiment is shown in **Figure 2-2**. The C program code enables and configures the system clock to 40MHz. The green LED on the EK-TM4C123GXL (connected to GPIO PF3) indicates if the device is in hibernate mode or wake up mode. For this purpose, the GPIO Port F of TM4C123GH6PM processor is enabled and pin 3 (PF3) configured as output. A HIGH is written on the GPIO pin to turn the green LED on.

The C code then enables the Hibernation module and configures the clock that feeds the module. It enables the GPIO pin state to be retained during hibernation and provides a 4-second delay for the user to observe the LED which is ON. The time interval for which the LED is ON can be programmed by specifying a the delay count value in the SysCtlDelay() function.

Calculation of delay

The number of counts required to get a time delay is given by

Number of Counts = Time delay required * System Clock Frequency

In hibernation mode, the processor uses the clock from the external crystal or oscillator. Hence in the program, the clock source used in hibernation mode is 16MHz. To get a required time delay of 4 second

Number of counts = $4 * 16 * 10^6 = 64 * 10^6$

The program configures the device to be woken up either by an input on the wake-up pin or by the RTC. It sets the RTC wake up parameters and turns the RTC on. The wake-up time is set to 5 seconds. The wake-up pin on the EK-TM4C123GXL is connected to SW2 on the board. When the switch SW2 is pressed and held, the device is woken up from the hibernation mode. Hence, the device can be woken up by either the switch SW2 or the RTC.

After setting the wake-up pin parameters, the device enters the hibernation mode and the green LED turns off. The Hibernation module is powered by a battery or an auxiliary power supply. If the supply to the Hibernation module is low or not present, the device may not enter the hibernate mode. Once in hibernate mode, the device can be woken up by pressing and holding the switch SW2 or by the RTC which is set for 5 seconds, whichever occurs earlier.

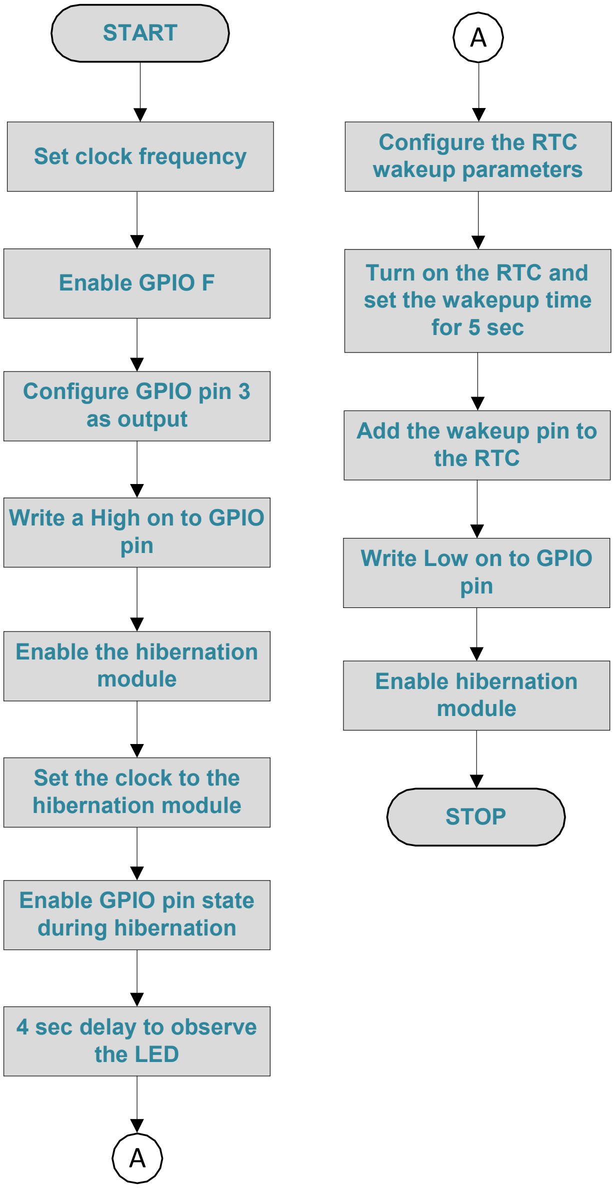

START
Set clock frequency
Enable GPIO F
Configure GPIO pin 3 as output
Write a High on to GPIO pin
Enable the hibernation module
Set the clock to the hibernation module
Enable GPIO pin state during hibernation
4 sec delay to observe the LED
A
A
Configure the RTC wakeup parameters
Turn on the RTC and set the wakepup time for 5 sec
Add the wakeup pin to the RTC
Write Low on to GPIO pin
Enable hibernation module
STOP

Figure 2-2 Flowchart for Hibernate Mode and Wake up using RTC

3.III.2 C Program Code for Hibernate Mode and Wake up using RTC

```
#include <stdint.h>
#include <stdbool.h>
#include "utils/ustdlib.h"
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/debug.h"
#include "driverlib/hibernate.h"
#include "driverlib/gpio.h"
int main(void)
{
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|
SYSCTL_OSC_MAIN);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|
GPIO_PIN_3);
GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x08);
SysCtlPeripheralEnable(SYSCTL_PERIPH_HIBERNATE);
HibernateEnableExpClk(SysCtlClockGet());
HibernateGPIORetentionEnable();
SysCtlDelay(64000000);
HibernateRTCSet(0);
HibernateRTCEnable();
HibernateRTCMatchSet(0,5);
HibernateWakeSet(HIBERNATE_WAKE_PIN | HIBERNATE_WAKE_RTC);
GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_3, 0x00);
HibernateRequest();
while(1)
{
}
}
```

Table 2-2: API Functions Used in the Application Program

API Function	Parameters	Description
HibernateEnableExpClk(uint32_t ui32HibClk)	• **Iui32HibClk** is the rate of the clock supplied to the Hibernation module.	This function enables the Hibernation module for operation.
HibernateGPIORetentionDisable(void)		This function disables the retention of the GPIO pin state during hibernation and allows the GPIO pins to be controlled by the system.
HibernateRTCSet (uint32_t ui32RTCValue)	• **ui32RTCValue i**s the new value for the RTC.	This function sets the value of the RTC
HibernateRTCEnable (void)		This function enables the RTC in the Hibernation module.
HibernateRTCMatchSet (uint32_t ui32Match, uint32_t ui32Value)	• **ui32Match** is the index of the match register • **ui32Value** is the value for the match register	This function sets a match register for the RTC.
HibernateWakeSet (uint32_t ui32WakeFlags)	• **ui32WakeFlags** specifies which conditions should be used for waking.	Configures the wake conditions for the Hibernation module.
HibernateRequest(void)		This function requests the Hibernation module to disable the external regulator, thus removing power from the processor and all peripherals.

Procedure

1. Connect the EK-TM4C123GXL to the PC using the USB cable supplied.

2. Build, program and debug the code to view the status of the green LED.

3. After 4 seconds, the green LED will switch off, indicating that the TM4C123GH6PM device has gone into hibernation.

4. Observe the status of the LED. After 5 seconds (RTC wake up time set in the code), the LED turns ON, indicating the RTC has woken the processor.

5. Also you can press and hold the SW2 button located at the lower right corner of the EK-TM4C123GXL to wake up the processor at any time.

6. On wake up the green LED will turn ON again.