Application Report
SPMA069 – January 2014

BQ32002 APIs for Tiva™

Jabir VS

ABSTRACT

This application report demonstrates the interfacing of the BQ32002 real-time clock with the Tiva microcontroller family. Basically, this document provides easy-to-use API calls for interacting with the BQ32002 real-time clock IC. An easy-to-use graphical user interface (GUI) software for the Windows® operating system is also provided for testing all the API calls using a PC.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/spma069.

Contents

List of Figures

1 Introduction

A software library is provided for interfacing the BQ32002 real-time clock IC with a 32-bit Tiva microcontroller. The library provides API calls for setting the values of various registers inside BQ32002. It also provides an API for reading all the time information from BQ32002. A GUI software running under Windows is also included to quickly evaluate these APIs without any efforts in the embedded source code development. The value of all the internal time keeping registers is displayed on the LCD screen of the TM4C123GH6PGE Evaluation Kit: http://www.ti.com/tool/ek-lm4f232. The entire software is tested using the TM4C123GH6PGE Evaluation Kit. Various APIs offered by the demo software are listed below:

- API for setting the Seconds register
- API for setting the Minute register
- API for setting the Hour register
- API for setting the Day of week register
- API for setting the Date register
- API for setting the Month register
- API for setting the Year register
- API for setting the all time keeping registers in single call

Tiva is a trademark of Texas Instruments.
Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries, or both.

- API for setting the IRQ at 1 Hz
- API for setting the IRQ at 512 Hz

2 System Block Diagram

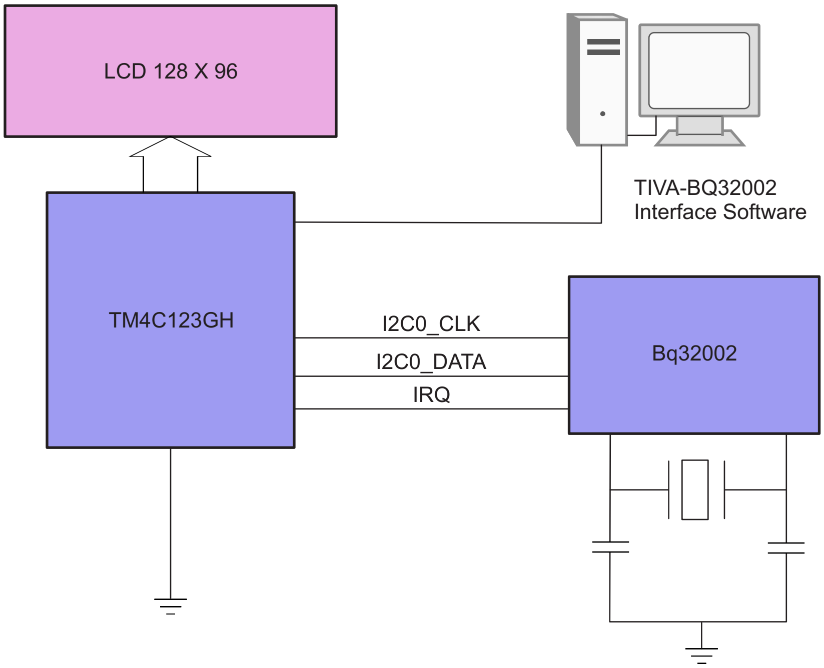


Figure 1. System Diagram

3 Theory of Operation

The real-time clock IC BQ32002 is interfaced with the Tiva TM4C123GH6PGE controller through inter-integrated circuit (I2C) interface. The I2C is configured to operate in fast mode of operation. The fast mode of operation gives a BQ32002 interface operating frequency of 400 Khz. The Tiva controller is connected to the PC using the universal asynchronous receiver/transmitter (UART) serial interface. The UART interface is configured to operate in 115200 baud rate with 8 data bits, one start bit, one stop bit, no parity and no flow control.

All the communication modules are operating on interrupt basis. Based on interrupts, all of the I2C transactions are working so that the CPU is not stalled for the transaction. Similarly, the UART data transaction is also interrupt based. Different levels of priority are given for various interrupts to ensure proper operation. For example, the I2C interrupt is given priority 4 and UART interrupt is given priority 7.

One additional interrupt used by the application is the general-purpose input/output (GPIO) interrupt. One GPIO(PORTB.BIT4) is configured to raise an interrupt whenever a falling edge is detected on that pin. The priority level for this interrupt is set as 0. In **Stellaris** controllers, the higher the priority number - the lower the priority of service.

On start up, the IRQ pin of BQ32002 is configured to go low on every one second; this pin is tied to the GPIO(PORTB.BIT0). This GPIO pin is configured to raise an interrupt whenever a falling edge is detected on the pin. Once the interrupt is triggered, the execution is taken to the interrupt handler for this GPIO. The function for reading the BQ32002 time keeping registers is initiated inside this interrupt handler and given as below:

```
BQ_Read_INT(timearr,0,7);
```

This function initiates the reading by sending the start condition and the BQ32002 address along with the R/W command bit on the I2C bus. Thereafter, all the additional transactions are initiated from the I2C interrupt handler for that particular API call.

This entire process will be repeated along with the change in the IRQ pin at each second.

Once the I2C operation is completed, all the time keeping register data from BQ32002 is loaded to an array named as timearr[ ]. After the above function call, the array timearr[ ] will have all the values from 7 internal registers of BQ32002, which are Seconds, Minutes, Hours, Day of week, Date, Month and Year registers. After retrieving the values from BQ32002, the application updates the values in to the display as shown in Figure 2 using the *displayfulldate()* function.

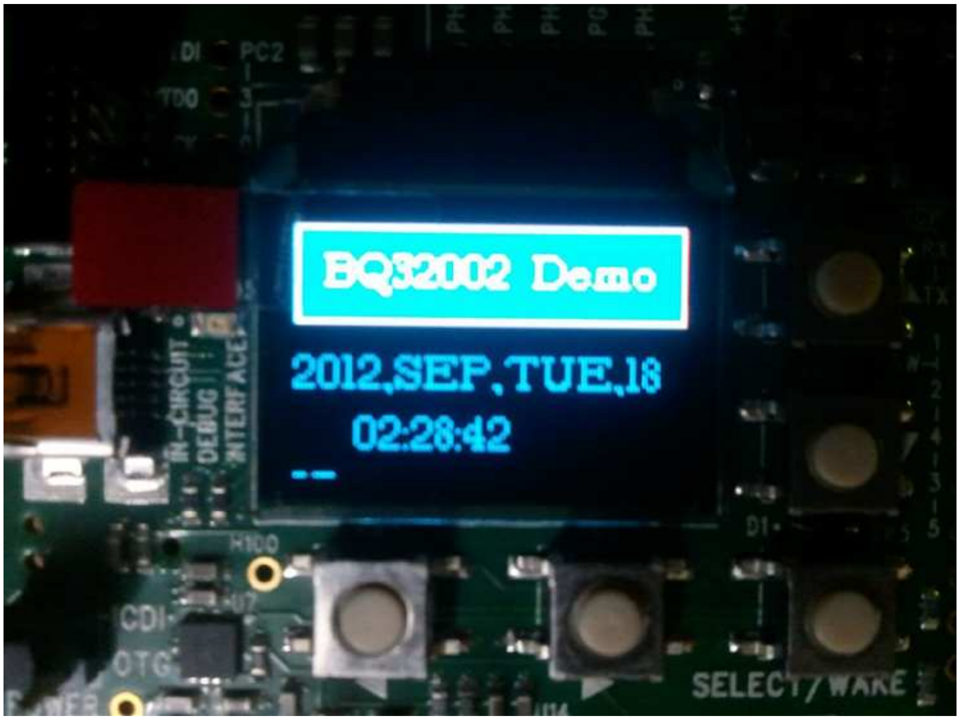


Figure 2. Time Display Format

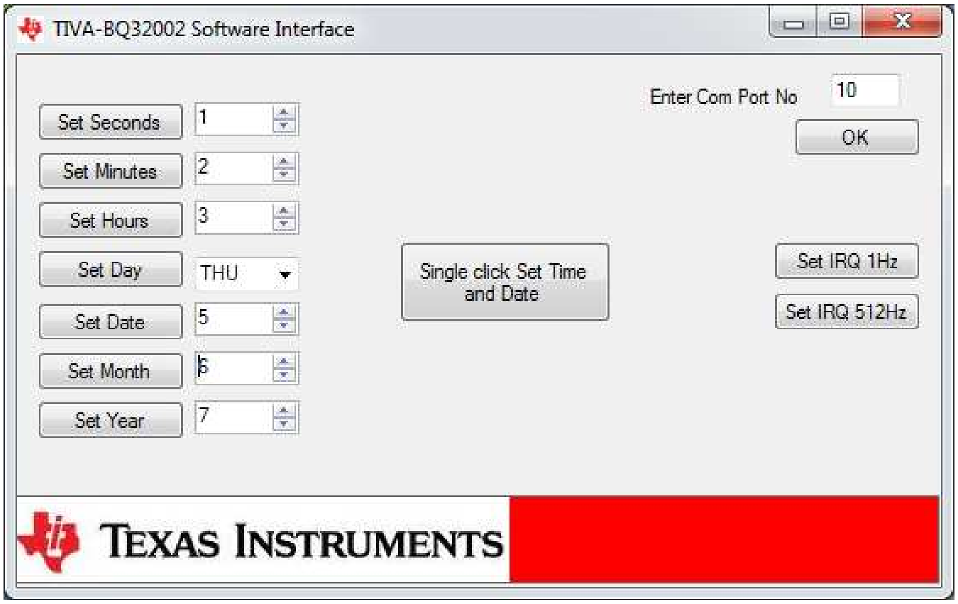
Table 1. Data Format of timearr[ ] After the Read Function Call

YEAR	MONTH	DATE	DAY	HOUR	MIN	SECOND
6	5	4	3	2	1	0

After all the initialization, the controller continuously checks the number of data bytes that arrive through the UART. This count is updated in the “ucount” variable for the use of application. Whenever the ucount variable is higher than five numbers, the controller starts checking the UART data stream for any pre-defined command from the computer for updating the BQ32002 registers. If any command is matched, then the corresponding API for setting the registers related to that command inside the BQ32002 is called.

4 Tiva BQ32002 PC Software

Figure 3 shows the screen shot of the Windows-based software for evaluating the functions for BQ32002 library.



TIVA-BQ32002 Software Interface

Set Seconds 1

Set Minutes 2

Set Hours 3

Set Day THU

Set Date 5

Set Month 6

Set Year 7

Enter Com Port No 10

Figure 3. Stellaris-BQ32002 PC Software

The Stellaris-BQ32002 interface software has a total of eleven functions available in the user interface. Features of the Stellaris-BQ32002 software interface are given below:

- Single click Set Time and Date – This is a push button switch through which you can directly set all the seven BQ32002 internal time keeping registers with the time of the personal computer on which the software is running.
- Set Seconds – This is an up and down menu type button that can be used for setting the value of seconds inside BQ32002 seconds register.
- Set Minutes – This is an up and down menu type button that can be used for setting the value of minutes inside BQ32002 minutes register.
- Set Hours – This is an up and down menu type button that can be used for setting the value of hours inside BQ32002 hours register.
- Set Day – This a pulldown menu type button that can be used for setting the day of the week inside BQ32002 day of week register.
- Set Date – This is an up and down menu type button that can be used for setting the value of the date inside BQ32002 date register.
- Set Month – This is an up and down menu type button that can be used for setting the value of the month inside BQ32002 month register.
- Set Year – This is an up and down menu type button that can be used for setting the value of the year inside BQ32002 year register.
- Set IRQ 1 Hz – This is a push button switch that sets the frequency of IRQ interrupt at 1 Hz.
- Set IRQ 512 Hz – This is a push button switch that sets the frequency of IRQ interrupt at 512 Hz.
- Set com port – The com port number through which the communication has to be set up is entered in this text box. After entering the port number, press the OK button.

5 APIs Available in the Demo Software

Below is the list of API calls available in the demo software.

5.1 API for Setting the Seconds

```
BQ_Write_INT(&data, seconds, 1);
```

"&data" is a pointer to the value to be written.

“Seconds” is a macro defined for the address of second register in BQ32002.

The third parameter should always be one for this function.

5.2 API for Setting the Minutes

```
BQ_Write_INT(&data, minute, 1);
```

"&data" is a pointer to the value to be written.

“minute” is a macro defined for the address of minute register in BQ32002.

The third parameter should always be one for this function.

5.3 API for Setting the Hours

```
BQ_Write_INT(&data, hour, 1);
```

"&data" is a pointer to the value to be written.

“hour” is a macro defined for the address of hour register in BQ32002.

The third parameter should always be one for this function.

5.4 API for Setting the Day

```
BQ_Write_INT(&data, day, 1);
```

"&data" is a pointer to the value to be written.

“day” is a macro defined for the address of day register in BQ32002.

The third parameter should always be one for this function.

5.5 API for Setting the Date

```
BQ_Write_INT(&data, date, 1);
```

"&data" is a pointer to the value to be written.

“date” is a macro defined for the address of date register in BQ32002.

The third parameter should always be one for this function.

5.6 API for Setting the Month

```
BQ_Write_INT(&data, month, 1);
```

"&data" is a pointer to the value to be written.

“month” is a macro defined for the address of month register in BQ32002.

The third parameter should always be one for this function.

5.7 API for Setting the Year

```
BQ_Write_INT(&data, year, 1);
```

"&data" is a pointer to the value to be written.

“year” is a macro defined for the address of year register in BQ32002.

The third parameter should always be one for this function.

5.8 API for Setting the IRQ at 1 Hz

```
setIRQ1();
```

No variable required

No macros required

5.9 API for Setting the IRQ at 512 Hz

```
setIRQ5();
```

No variable required

No macros required

5.10 API for Setting All the Time Keeping Registers in Single Call

```
void Settime(char tyear,char tmonth,char tdayofweek,char tdate,char thour, char tminute, char
tseconds);
```

Variables:

- tyear – Value for setting the year register of BQ32002 in decimals. Valid values are 0-99.
- tmonth – Value for setting the month register of BQ32002 in decimals. Valid values are 0-12.
- tmonth – Value for setting the month register of BQ32002 in decimals. Valid values are 0-12.
- tdate – Value for setting the date register of BQ32002 in decimals. Valid values are 0-31.
- thour – Value for setting the hour register of BQ32002 in decimals. Valid values are 0-59.
- tminute – Value for setting the minute register of BQ32002 in decimals. Valid values are 0-59.
- tseconds – Value for setting the seconds register of BQ32002 in decimals. Valid values are 0-59.

6 How to Use the APIs in an End Application

1. Include the header named BQ_I2C Functions.h in the main application file as shown below:

   ```
   #include "BQ_I2C_Functions.h"
   ```
2. Add the source file “BQ_I2C_Functions.c” in the project.
3. Replace the default interrupt handler for the I2C0 interrupts by writing the new I2C0 interrupt handler name in the startup_ccs.c file, as done in the demo project.
4. Declare the variable in the main application file, which is declared as external in the BQ_I2C_Functions.c file.

7 Summary

This application report helps to quickly add an external real-time clock IC into a Tiva controller-based system. Because the implementation is based on the interrupts and utilizes the internal I2C engines, it helps to integrate the code into an existing system without affecting the timing of existing application modules in the system. The provided software interface allows you to quickly test and evaluate the performance of the software and hardware without overheads.

8 References

- TM4C123GH6PGE Evaluation Kit http://www.ti.com/tool/ek-lm4f232
- TM4C123GH6PGE (ACTIVE) Tiva C Series Microcontroller: http://www.ti.com/product/tm4c123gh6pge
- BQ32000 (ACTIVE) Serial RTC with Battery Backup and Trickle Charge: http://www.ti.com/product/bq32000

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265