Application Brief

Add Housekeeping Functions to Your MSP430™ MCU: Real-Time Clock with Backup Memory

In a number of applications such as metering, building automation, and remote sensing, it is important to be able to keep track of the real time for monitoring or coordination purposes. Often additional information about system state also needs to be stored through a power loss to the main system. The MSP430FR2433 microcontroller (MCU) can be used as a low cost solution for this problem by making use of the internal real time clock (RTC) counter module and internal ferroelectric random access memory (FRAM) as backup memory. By using the UART, the host can set the initial time in POSIX format, and it can interrogate the MSP430™ MCU over the UART to provide the current time in the same format. Additionally, the host can write to and read from 16 bytes of backup memory, within the MSP430FR2433 device. This allows for data retrieval even after a total system power loss. To get started, download project files and a code example demonstrating this functionality

Note

This example can be used with any MSP430 LaunchPad™ Development Kit with the required MCU peripherals. For migrating pinouts and peripherals, see the device-specific data sheet.

Implementation

The RTC Counter module is used as an external RTC to provide the current real time in POSIX format. The host can set the initial time and then read the current time through a UART interface. The 16-MHz SMCLK is used as the clock source for the RTC Counter and UART modules. Alternatively, an external 32768-Hz crystal can be used as the clock source if the crystal selection resistors are changed on the MSP430FR2433 LaunchPad Development Kit and the __ENABLE_XT1__ symbol is added to the compiler settings in the CCS project. The RTC Counter module sourced from the 32768-Hz crystal is configured to generate an interrupt once per second. The interrupt service routine updates the timestamp value stored in FRAM. The host must set the initial timestamp the very first time after programming using the WRITE TIME command. From then on, the MSP430 device will keep updating the timestamp value. The current timestamp value will be retained through a reset or power loss.

Note

This example requires resistors R2/R3 to be populated and R4/R5 to be removed to use the LF XTAL for the MSP430FR2433 as shown in Figure 1. For the exact specifications to enable the LF XTAL, check your LP documentation.

The MSP430FR2433 LaunchPad Development Kit (Figure 1) was used for testing this solution, but it can be used on any MSP430 MCU with the proper peripheral support and code migration. The LaunchPad's eZFET back-channel UART interface can be connected to a PC terminal program or GUI Composer at 9600 baud for evaluating and further testing.

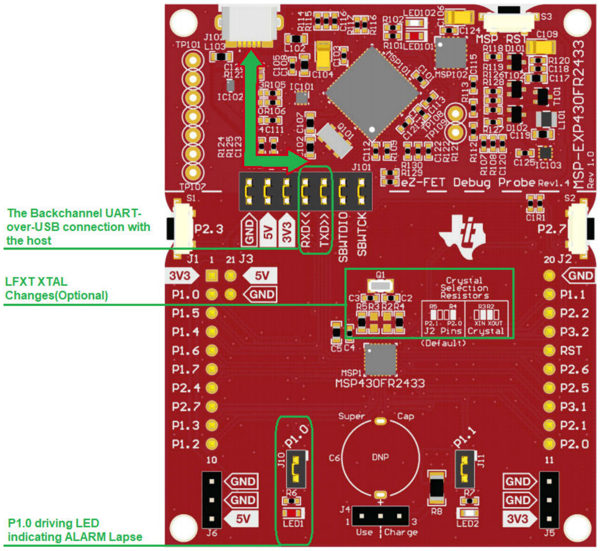


Figure 1. MSP-EXP430FR2433 LaunchPad Development Kit

The firmware implements the following UART communication protocol:

COMMAND	D0	D1	D2	D3

For the READ commands, the data bytes are requested as a response from the MSP430 MCU. The command byte for each response is included for identifying the data more easily. For the WRITE commands, the data bytes are sent to the MSP430 MCU to be processed. Both the timestamp and alarm data are sent LSB first. The timestamp data should be interpreted as D3D2D1D0h, and the lapse time data as D0h.

RTC_READ_TIME Command

00h	D0	D1	D2	D3

RTC_WRITE_TIME Command

01h	D0	D1	D2	D3

ALARM_WRITE_TIME Command

02h	D0

BACKUP_MEM_ADDRESS Command

03h	D0

BACKUP_MEM_READ_DATA Command

04h	D0

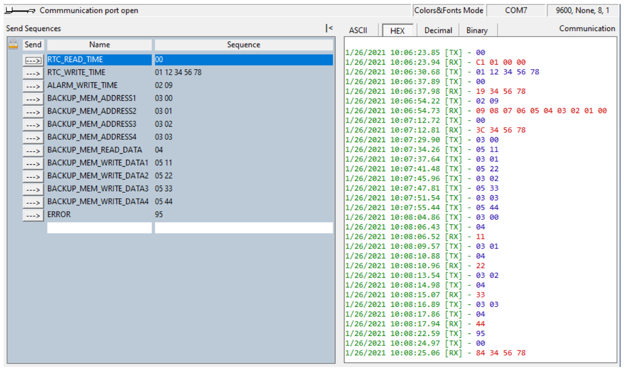
BACKUP_MEM_WRITE_DATA

05h	D0

Performance

In this solution both timestamp and the backup memory are stored in FRAM and are nonvolatile, so they will be retained even through a total power loss. By default, the FRAM is write-protected. The FRAM is only unprotected when a timestamp and must be stored, and then it is protected again after the RTC Counter module in the device updates the real time once per second. The host can request the current real time by sending the RTC_READ_TIME command to the MSP430 MCU. If the real time must be updated, the host can send the RTC_WRITE_TIME command along with the current real time. Also sending the ALARM_WRITE_TIME command to the MSP430 MCU sets the Alarm and starts the count down and glows the LED once the time reduced to 0. For storing the backup memory in FRAM, First set the address using BACKUP_MEM_ADDRESS command and the write the 8 bit data using the BACKUP_MEM_WRITE_DATA command and to read the data, first set the address using BACKUP_MEM_ADDRESS command and send BACKUP_MEM_READ_DATA command which will read the data from designated address.

For the standalone demo, a PC terminal program (for example, Docklight, Tera Term, and so forth) can be used as the host. An example demonstrating all of the commands and data including reading the initial RTC time, setting a new time and reading the time to confirm it was updated is shown in Figure 2. Observe that the RTC_WRITE_TIME command was sent to the MSP430 MCU at 10:06:30 and set to 78563412h. 7 second later at 10:06:30, the RTC_READ_TIME command was sent, and the reply shows the current timestamp value was incremented by 7 second to 78563419h. Notice that an unsupported command of 95h was sent but was correctly ignored by the MSP430 MCU at 10:08:22.

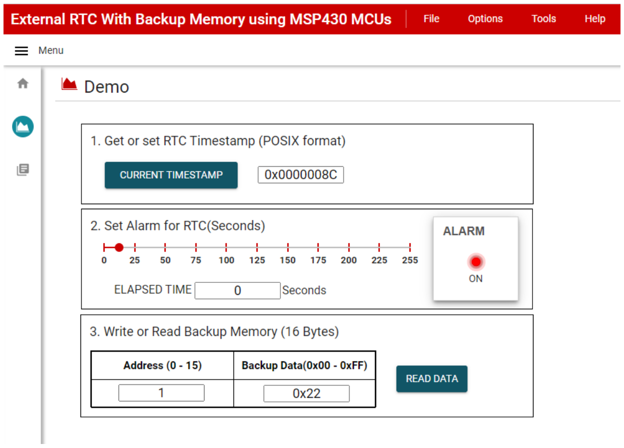


Send	Name	Sequence
--->	RTC_READ_TIME	00
--->	RTC_WRITE_TIME	01 12 34 56 78
--->	ALARM_WRITE_TIME	02 09
--->	BACKUP_MEM_ADDRESS1	03 00
--->	BACKUP_MEM_ADDRESS2	03 01
--->	BACKUP_MEM_ADDRESS3	03 02
--->	BACKUP_MEM_ADDRESS4	03 03
--->	BACKUP_MEM_READ_DATA	04
--->	BACKUP_MEM_WRITE_DATA1	05 11
--->	BACKUP_MEM_WRITE_DATA2	05 22
--->	BACKUP_MEM_WRITE_DATA3	05 33
--->	BACKUP_MEM_WRITE_DATA4	05 44
--->	ERROR	95

```
1/26/2021 10:06:23.85 [TX] - 00
1/26/2021 10:06:23.94 [RX] - C1 01 00 00
1/26/2021 10:06:30.68 [TX] - 01 12 34 56 78
1/26/2021 10:06:37.89 [TX] - 00
1/26/2021 10:06:37.98 [RX] - 19 34 56 78
1/26/2021 10:06:54.22 [TX] - 02 09
1/26/2021 10:06:54.73 [RX] - 09 08 07 06 05 04 03 02 01 00
1/26/2021 10:07:12.72 [TX] - 00
1/26/2021 10:07:12.81 [RX] - 3C 34 56 78
1/26/2021 10:07:29.90 [TX] - 03 00
1/26/2021 10:07:34.26 [TX] - 05 11
1/26/2021 10:07:37.64 [TX] - 03 01
1/26/2021 10:07:41.48 [TX] - 05 22
1/26/2021 10:07:45.96 [TX] - 03 02
1/26/2021 10:07:47.81 [TX] - 05 33
1/26/2021 10:07:51.54 [TX] - 03 03
1/26/2021 10:07:55.44 [TX] - 05 44
1/26/2021 10:08:04.86 [TX] - 03 00
1/26/2021 10:08:06.43 [TX] - 04
1/26/2021 10:08:06.52 [RX] - 11
1/26/2021 10:08:09.57 [TX] - 03 01
1/26/2021 10:08:10.88 [TX] - 04
1/26/2021 10:08:10.96 [RX] - 22
1/26/2021 10:08:13.54 [TX] - 03 02
1/26/2021 10:08:14.98 [TX] - 04
1/26/2021 10:08:15.07 [RX] - 33
1/26/2021 10:08:16.89 [TX] - 03 03
1/26/2021 10:08:17.86 [TX] - 04
1/26/2021 10:08:17.94 [RX] - 44
1/26/2021 10:08:22.59 [TX] - 95
1/26/2021 10:08:24.97 [TX] - 00
1/26/2021 10:08:25.06 [RX] - 84 34 56 78
```

Figure 2. Memory backup with a Timestamp Standalone Demo

For the GUI demo, the GUI can be used as the host as shown in Figure 3. The RTC can be read by clicking the CURRENT TIMESTAMP button in the first section. A new timestamp can be set by entering an 8-digit POSIX timestamp in the text box and pressing Enter. The Alarm can be set using the slider control (maximum of 255 seconds). Once the Alarm is set, ELAPSED TIME field shows the countdown time and once this value reduced to 0, the ALARM LED on the GUI turns ON also glows LED (P1.0) on The MSP430FR2433 LaunchPad Development Kit. Note that the GUI demo code can be used in standalone mode by deleting the __ENABLE_GUI__ symbol in the compiler settings in the CCS project, recompiling the code and programming the device.



1. Get or set RTC Timestamp (POSIX format)

CURRENT TIMESTAMP: 0x0000008C

2. Set Alarm for RTC(Seconds)

ELAPSED TIME 0 Seconds

ALARM: ON

3. Write or Read Backup Memory (16 Bytes)

Address (0 - 15)	Backup Data(0x00 - 0xFF)
1	0x22

Figure 3. Memory backup with a Timestamp GUI Demo

Limitations of this implementation include that no temperature compensation or calibration to adjust for small ppm errors of the 32768-Hz crystal is implemented. This means that the RTC could drift slightly over time, so it would be recommended for the host to periodically re-sync the RTC by sending the WRITE TIME command (for example, once per day).

To Get Started

1. Watch the training video “Real-Time Clock with Backup Memory using a Housekeeping MCU” to learn how to use the GUI to set a timestamp and then find out when a power failure occurred.
2. Order a MSP430FR2433 LaunchPad Development Kit to evaluate the Voltage Monitor with a Timestamp example GUI and code.
3. Download and test the Real-Time Clock with Backup Memory example GUI to easily configure and view the timestamps and thresholds.
4. Evaluate the Real-Time Clock with Backup Memory example code using the MSP430FR2433 LaunchPad Development Kit.

Table 8. Device Recommendations

Part Number	Key Features
MSP430FR2433	16KB FRAM, 4KB SRAM, 10-bit ADC, UART/SPI/I2C, Timer
MSP430FR2422	8KB FRAM, 2KB SRAM, 10-bit ADC, UART/SPI/I2C, Timer

Trademarks

MSP430™ and LaunchPad™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

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