Freescale MQX RTOS Example Guide

RTC example

This document explains the RTC example, what to expect from the example and a brief introduction to the API.

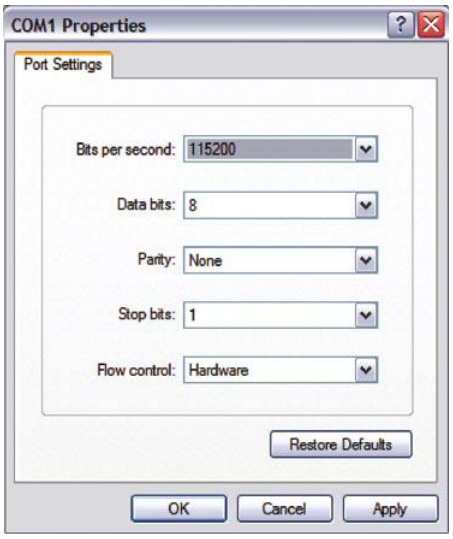
The example

The RTC example synchronizes the RTC module time with MQX RTOS system time, sets the minute stopwatch interrupt and handler, sets the second alarm interrupt and handler, shows the IRTC counter up demo, the IRTC standby RAM demo and the Tamper demo.

Running the example

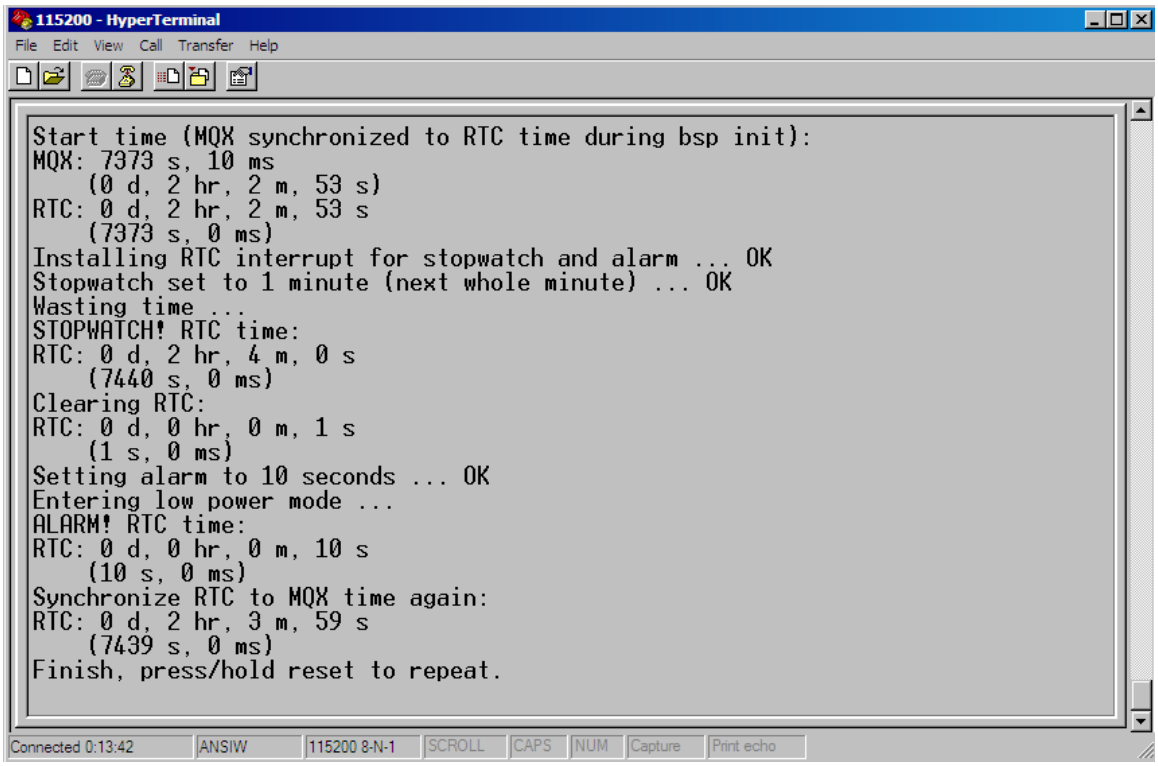
Run HyperTerminal on the PC (Start menu->Programs->Accessories->Communications).
Make a connection to the serial port which is connected to the board (usually will be COM1).

Set it with 115200 baud, no parity, 8 bits and click OK.



Run the code, the serial terminal will show current MQX RTOS system absolute time (Second/millisecond), RTC clock synchronize time with system time. Install RTC interrupt for minute stopwatch and second

alarm. Then the stopwatch interrupt demo sets the stopwatch value to 1 minute, with 1 minute went by, serial terminal will show “STOPWATCH”. Then clear RTC time to 0.
Following is second alarm interrupt demo, the demo sets 10 seconds alarm and the microcontroller enters into low power mode, after 10 seconds the serial terminal shows “ALARM”. The Microcontroller wakes up from low power mode and re-synchronizes time with MQX RTOS system time. Below are included the serial terminal snapshot that shows the RTC example running on a M52259EVB board:



Note: if using MCF51CN or MCF51EM, the microcontroller does not enter into low power mode; just prints “Wasting time ...” in serial terminal.

If the microcontroller has the IRTC module, the code will execute the counter up demo and standby RAM demo.

If microcontroller is MCF51EM, code will execute the tamper demo. Press tamper button or remove battery, the serial terminal will display the temper event and cause and time stamp.

Note: IRTC counter up and standby RAM demos run at M51EMDEMO or TWRMCF51CN boards; Tamper demo only runs at M51EMDEMO board.

Explaining the example

The application example creates only one main task. The flow of the task is described in next figure. With RTC difference, IRTC counter up and standby RAM demos run at M51EMDEMO or TWRMCF51CN boards; Tamper demo only runs at M51EMDEMO board. Flow chart is with related mark.

The main task starts with the RTC clock synchronize time with MQX system time. MQX system time format is absolute second/millisecond

time, RTC time format is day/minute/second time; then it calls following lines to do time format transform between RTC time with MQX time.

```
_rtc_time_from_mqx_time (&time_mqx, &time_rtc);
_rtc_time_to_mqx_time (&time_rtc, &time_mqx);
```

RTC demo is using two interrupt sources: minute stopwatch and second alarm. Install RTC interrupt and enable related interrupts of RTC. Minute stopwatch interrupt demo is using following line to set minute stopwatch interrupt to trigger 1 minute later:

```
_rtc_set_stopwatch (1);
```

Then with the while loop waits for 1 minute stopwatch interrupt to happen and in ISR routine sets the related state bit.

Between two interrupt demos, it calls RTC initialization function to clear RTC time to 0 and clear interrupt enable bit in RTC:

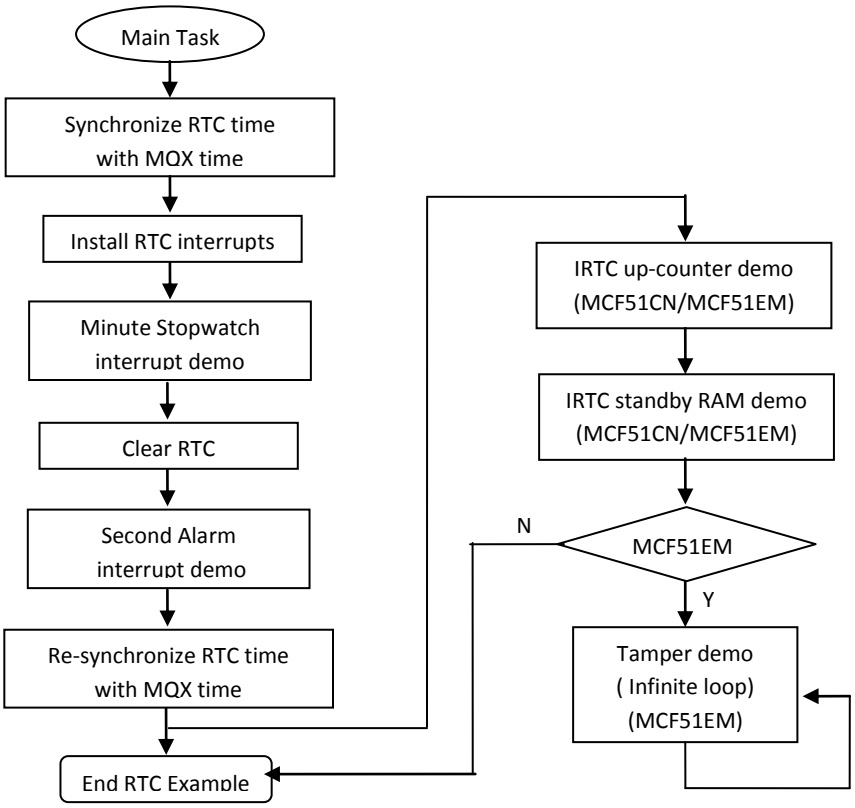
```
_rtc_init (RTC_INIT_FLAG_CLEAR | RTC_INIT_FLAG_ENABLE);
```

Second alarm interrupt demo sets 10 seconds interrupt time interval with following line:

```
time_rtc.seconds = 10;
_rtc_set_alarm (&time_rtc);
```

During second 10 seconds time interrupt, CPU enters into low power mode and RTC interrupt can wake up the CPU to reenter on run mode.

After the second interrupt happens, RTC time synchronizes to MQX system time again.



RTC interrupt service routine returns the interrupt status info:

```
state = _rtc_get_status ();
```

IRTC increments counter value by 12 and standby RAM demo writes data to standy-RAM and reads back to verify, which are available for IRTC module.

Tamper demo runs in infinite loop and displays the cause of tamper event and its timestamp, this demo runs at M51EMDEMO board.