

Freescall MQX Example Guide

hwtimer example

This document describes the hwtimer component example application. The example hwtimer handles two different hwtimers instances with the callback functionality. It shows how to work with the component and how to use API functions.

Running the example

Start a terminal application on your PC and set the serial connection for 115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.

Start hwtimer example on the target platform. For instructions how to do that in different IDEs and for different debuggers, see the MQX documentation (<MQX installation folder>/doc/tools).

After starting the application, you will see the printed message as the following.

```
-----START OF HWTIMER EXAMPLE-----
Initialization of hwtimer1   : OK
Try to set frequency 250 Hz to hwtimer1
Read frequency from hwtimer1 : 250 Hz
Read period from hwtimer1    : 4000 us
Read modulo from hwtimer1    : 192000
Register callback for hwtimer1

Initialization of hwtimer2   : OK
Try to set period 1000000 us to hwtimer2
Read frequency from hwtimer2 : 1 Hz
Read period from hwtimer2    : 1000000 us
Read modulo from hwtimer2    : 48000000
Register callback for hwtimer2
Start hwtimer2
hwtimer2_callback 1 tick : start hwtimer1
.....
.....
.....
hwtimer2_callback 2 ticks: block hwtimer1_callback

hwtimer2_callback 4 ticks: unblock hwtimer1_callback
.....
.....
.....
.....
.....
.....
hwtimer2_callback 6 ticks: stop hwtimer1
hwtimer2_callback 6 ticks: stop hwtimer2
hwtimer2_callback 6 ticks: hwtimer1 = 1249 ticks
Deinit hwtimer1
Deinit hwtimer2
-----END OF HWTIMER EXAMPLE-----
```

Explanation of the example

There is just one task (hwtimer_task) and callback function for each hwtimer (hwtimer1_callback, hwtimer2_callback).

hwtimer_task:

- Create job_done lw semaphore.
- Initialize hwtimer1.
- Set the frequency of 250Hz for hwtimer1.
- Register callback hwtimer1_callback() for hwtimer1.
- Initialize hwtimer2.
- Set the period of 1000000ms for hwtimer2.
- Register callback hwtimer2_callback() for hwtimer2.
- Start hwtimer2.
- Wait on job_done lw semaphore.
- Deinitialize hwtimer1 and hwtimer2.
- Destroy job_done lw semaphore.

hwtimer1_callback:

- Print "."

hwtimer2_callback:

- Store ticks of hwtimer2.
- After 1s (ticks == 1) start hwtimer1.
- After 1s (ticks == 2) block callback of hwtimer1.
- After 2s (ticks == 4) unblock callback of hwtimer1.
- After 2s (ticks == 6) stop both hwtimers and post job_done lw semaphore.

