# Freescale MQX RTOS Example Guide
# usermode

This document describes the usermode example which demonstrates use of memory protection, dynamic memory allocations and dynamic task creation in the user-mode tasks (Kinetis platforms only).

## Pre-requisities

This example requires the support for User Mode to be compiled in the MQX kernel. Edit the <mqx_installation>/config/<board>/user_config.h file and add the

```
#define MQX_ENABLE_USER_MODE 1
```

Then rebuild the MQX as described in the MQX Getting Started document. **Also make sure the board jumpers are set properly as described in this document.**

Note that this application uses a different linker command file than the other examples which do not make use of User mode. The linker file for User mode applications defines additional memory areas for different levels of protection as well as the area for the memory heap used by User mode tasks. The linker files suitable for the User mode examples are named with the *_usr* postfix.

Note that not all build tools and processor platforms are supported by the User-mode feature.

## More Reading

Read more information about the User vs. Privileged execution mode and about memory protection in the MQX RTOS User Guide. See also User mode API in the MQX API Reference Manual. The User mode functions all have the *_usr* prefix.

See also other User mode examples in the MQX distribution: access_usr, lwsem_usr and lwdemo_usr.

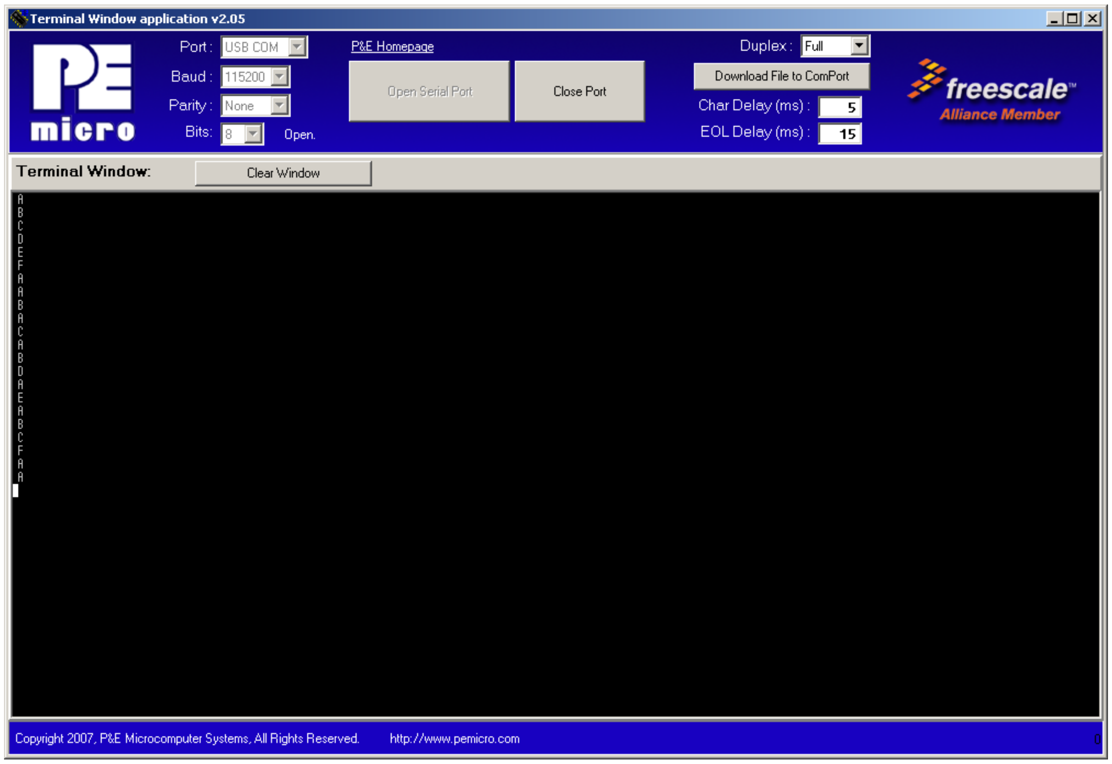
## The Example

The example demonstrates use of dynamic memory allocation and dynamic task creation from within a User mode tasks. Also it demonstrates lightweight message passing from multiple instances of User mode tasks to the Privilege task.

The example uses the following tasks as defined in the usermode.c file:

- **privilege_task**: The auto-start task which creates two lightweight memory pools, one set read-only and the other read-write for User mode tasks. The first is used for message queue (must be read-only for User mode tasks) and the other is used for dynamic allocation by User mode tasks.
- **usr_main_task**: User mode task created by the privilege_task. This task further creates USR_TASK_CNT number of other User tasks using the user mode API.
- **usr_task**: There are USR_TASK_CNT instances of this task running in User mode. Each task allocates the message object by using a User mode API and uses it to pass data to the privilege_task.

If everything runs correctly, the user tasks pass its identification letter up to the privilege task which prints that to the console. Each user task sleeps different number of ticks so the output will look “random” like A B C D E F A A B A...



Port : USB COM
Baud : 115200
Parity : None
Bits: 8
Open.
Duplex : Full
Char Delay (ms) : 5
EOL Delay (ms) : 15

```
A
B
C
D
E
F
A
A
B
A
C
A
B
D
A
E
A
B
C
F
A
A
```

Copyright 2007, P&E Microcomputer Systems, All Rights Reserved.   http://www.pemicro.com