Freescale MQX RTOS Example Guide

Benchmrk example

This document describes the benchmark codesize example application. The application is to analyze the memory ROM and RAM usage of MQX RTOS components. The result is generated in the form of an .html file which shows the memory space in bytes occupied by the specified components.

Running the example

The codesize example is designed to run for every tool chains including IAR, KDS, DS5, GCC, KEIL,… and platform family: Vybrid, Kinetis, Coldfire, PPC,… that supported Makefile. It is also autobuild Libraries (BSP, PSP) and application by one click. Following steps are required to generate a map file that contains the memory usage.

- Skip this step if you just report ROM size, goto <MQX_ROOTDIR>\config\frdmk22f120m\user_config.h add a line: #define MQX_MEM_MONITOR 1
- Modify the path to compilers: *<MQX_ROOTDIR>\mqx\examples\benchmrk\codesize\make\launchers\<board_name>.bat* Or *<MQX_ROOTDIR>\build\common\make\global.mak*
- Run the Bat file at: *<MQX_ROOTDIR>\mqx\examples\benchmrk\codesize\make\launchers\<board_name>.bat*

After compilation complete, the map files can be found at: <MQX_ROOTDIR>\mqx\examples\benchmrk\codesize\output\apps\<config_build>

Explanation of the example

The benchmark codesize example is not designed to generate any physical output on any target platform but rather it is to analyze the MQX RTOS components of how much memory (ROM/RAM) they occupy to assist user in design of application using MQX RTOS and the accompanied elements and to select appropriate platform with affordable RAM/ROM for desired applications.

There are three different configurations for MQX RTOS kernel. They are **maximum** configuration with all the optional and core components of MQX RTOS, **typical** configuration without some optional components like semaphores, events, logs, etc... and the **small** configuration with some core components of MQX RTOS. For detail of the components used please take a look at the **verif_enabled_config.h** of each configuration.

The .html report file lists the memory usage of components in a tabular form. Those components not available for specific MQX RTOS configuration and application build target is shown with "-" symbol.

Benchmrk Tool script

Freescale Semiconductor MQX(TM) CodeSize script (codesize.pl)

The script analyzes the generated xMAP file produced by CodeWarrior, IAR Embedded Workbench, DS5, KDS, Keil or GCC ARM build tools and creates codesize report in an HTML format.

The script PERL source code is located in the current folder, the script executable is located in codesize directory.

The BSP and PSP libraries as well as the application code needs to be built in a special way in different configurations and different include search-paths to the customized **verif_enabled_config.h** files. See the "make" folder for custom makefiles used for this purpose. Note that the makefiles depend on the master Makefiles of the MQX in the top-level /build/<*board_name*>/make and hello/build/<*board_name*>/ folder.

The reports created with this script for different build tools are available in the "report" folder. The MQX kernel configurations used for these results are available in the "config" directory.

```
===========================================================================

Supported tools:
    CodeWarrior for MCU/ColdFire 6.3 (use -c cwcf6)
    CodeWarrior for ColdFire 7.2 (use -c cwcf7)
    CodeWarrior for mobileGT 9.2 (use -c cwmpc9)
    CodeWarrior version 10 for ColdFire platforms (use -c cwcf10)
    CodeWarrior version 10 for Kinetis platforms  (use -c cwarm10)
    IAR Embedded Workbench for ARM v6.10 or later (use -c iararm6)
    ARM-MDK Keil uVision v4.23 or later (use -c uv4)
    ARM-DS5 Development Studion v5.11 or later (use -c ds5)
    GNU Tools for ARM Embedded Processors version 4.8 - or later (use -c
    gcc_arm)

Options:
    -M ... print detailed MAP file analysis report
    -t ... dump text output to console
    -W ... enable warning messages
    -o <FILE>  ... specify report output file name
    -n <NAME>  ... alternative MAP file name displayed in the report
    -c <FORMAT> .. MAP file format specifier (use before the MAP file
name)
       supported formats: cwcf7, cwmcu6, cwmpc9, cwcf10, cwarm10,
iararm6, uv4

Report Options:
    +PSP  or -PSP  .. force or suppress PSP details (default +)
    +BSP  or -BSP  .. force or suppress BSP details (default +)
    +MFS  or -MFS  .. force or suppress MFS File System details
(default -)
    +RTCS or -RTCS .. force or suppress RTCS TCP/IP stack details
(default -)
    +USBH or -USBH .. force or suppress USB Host stack details (default
-)
    +USBL or -USBL .. force or suppress USB Dev stack details  (default
-)
    +SUMM or -SUMM .. force or suppress summary report (default +)
```

```
Usage examples:
    codesize.exe -c cwcf7 <MAP_file>
    codesize.exe -M -c cwcf10 <MAP_file>
    codesize.exe +MFS -PSP -BSP -c cwcf10 <MAP_file>
    codesize.exe -c cwcf10 <MAP_file1> -c cwmpc9 <MAP_file2>
```