

# Freescal MQX RTOS 4.1.2 Vybrid Release Notes

## Contents

## 1 Introduction

This is the Release Notes for the Vybrid standalone package for Freescal MQX™ RTOS 4.1.2. Freescal SVF522REVB and TWR-VF65GS10 belong to the Vybrid series processor family of 32-bit dual-core microcontrollers. The software is built based on the MQX RTOS version 4.1.1. It includes the full set of RTOS services and a standard set of peripheral drivers. Prior installation of MQX RTOS 4.1.1 is not required to install this package. For more information, see *Freescal MQX™ RTOS Release Notes* and *Getting Started with Freescal MQX™ RTOS*.

1	Introduction.....	1
2	What Is New.....	2
3	Release Content.....	5
4	MQX RTOS Release Overview.....	6
5	Set up installation instructions and technical support.....	11
6	Known Issues.....	12

## 1.1 Development Tools Requirements

The boards included in Vybrid release are tested with these development tools:

- IAR Embedded Workbench for ARM® Version 7.30.1
  - See build projects in iar subdirectories
- GNU Tools for ARM Embedded Processors version 4.8-2014-q1
  - Makefiles build option: TOOL =gcc\_arm
- Makefile support (mingw32-make version 4.8.1)

- Library makefiles are located in <install\_dir>/build/make/<board>
- Application makefiles are located in <example\_dir>/make/<board>
- Arm DS-5 Development Studio Version 5.19.0
  - See build projects in ds5 subdirectories

## 1.2 System Requirements

System requirements are based on the requirements for the development tools. There are no special host system requirements for hosting the Freescale MQX RTOS distribution itself.

Minimum PC configuration:

- As required by Development and Build Tools

Recommended PC configuration:

- 2 GHz processor – 2 GB RAM - 2 GB free disk space

Software requirements:

- OS: Windows® 7 or later

## 1.3 Target Requirements

The Vybrid package was tested with the following hardware configurations:

- X-SVF522R-EVB Vybrid Evaluation Board Rev. B with the SVF522R3MK4 processor.
- TWR-VF65GS10 Development Kit Rev. H with the MVF61NS10MK50 processor.

## 2 What Is New

This section describes the major changes and new features implemented in this release.

- This Board Support Package was added:
  - BSP package for TWR-VF600 was replaced by new TWR-VF65GS10
  - Vybrid Auto EVB board was replaced by new X-SVF522R-EVB board
- MQX RTOS 4.1.2 for Vybrid enables development on Linux machines (tested on Ubuntu 13.10). MQX RTOS is distributed in two variants, classic Windows® installer and Linux distribution in the form of a zip package. The Linux package supports these features:
  - DS5
  - IAR
  - GCC make and GDB debugging
- MFS library was updated with this functionality:
  - Support for multiple concurrent readers/writers of the same file
  - Support of mutex in addition of lwsem - solution to priority inversion
  - Locking mechanism reviewed
  - Read only checks - improved efficiency
  - Better abstraction of FAT chain handling – better performance of random seek
  - Scalable sector cache layer – dedicated directory and FAT sector buffers were removed
  - True file append mode – seek to end of file atomically prior each write operation

- Legacy open mode "x" for creating temporary files was removed – the caller shall generate a unique file name according to the use case
- Speed and CodeSize optimizations
- New USB stack was added.
  - Composite device
  - PHDC class (Peripheral and Host)
- Exception support for cortex A kernel
- Video ADC + VIU example application – demo/vadc
- RTCS package was updated by this features:
  - HTTP Server new features:
    - Web-sockets
    - Chunk transfer encoding (variable length CGI...)
  - TCP and UDP code size optimization
  - IPv6 TFTP Server
  - IPv6 Telnet Server
  - Echo server and client
  - DHCPv6 client
  - BSD-like socket select()
  - Socket send flags

This release includes package for new X-SVF522R-EVB development platform.

- SVF522REVB\_A5
  - Core clock: 396 MHz (High Speed Run mode, default)
  - Bus clock: 132 MHz (High Speed Run mode, default)
  - BSP Timer: Global Timer
  - Default console: ttypc
  - Target for GDB: SVF522R3K1CMK4\_A5
- SVF522REVB\_M4
  - Core clock: 132 MHz (High Speed Run mode, default)
  - Bus clock: 66 MHz (High Speed Run mode, default)
  - BSP Timer: SysTick
  - Default console: ttypc
  - Target for GDB: SVF522R3K1CMK4\_M4

BSP library for TWR-VG65GS10 was updated to the board revision. Additionally, new header files are provided for all Vybrid supported boards.

- TWRVF65GS10\_A5
  - Core clock: 396 MHz (High Speed Run mode, default)
  - Bus clock: 132 MHz (High Speed Run mode, default)
  - BSP Timer: Global Timer
  - Default console: ttyb
  - Target for GDB: VF6xx\_A5
- TWRVF65GS10\_M4
  - Core clock: 132 MHz (High Speed Run mode, default)
  - Bus clock: 66 MHz (High Speed Run mode, default)
  - BSP Timer: SysTick
  - Default console: ttypc
  - Target for GDB: VF6xx\_M4

Resolved issues:

- MQX-4660 RTC time is not accurate. The issue was solved by proper initialization of RTC clock source.
- MQX-4593 ARP shell command does not work with ENET device number 1. Problem solved with added multi-interface support.
- MQX-4411 Wrong IOMUX setting for CAN pins was fixed.

## What Is New

- MQX-4204 Lwtimer callback wasn't call in case that task was blocked. Issue was fixed with setting default stdin, stdout and stderr for idle task in `_bsp_init()`. When printing from ISR is used, standard output is used from active task. Printing from ISR is now possible in case that interrupted task is idle task.
- MQX-4012 QuadSPI – Issue with dual/quad io mode, when set read clk  $\geq$  80MHz and write clk = 66MHz transmission failure fixed.
- MQX-4470 MFS incorrect update of directory record was fixed.
- MQX-4406 Change of bool type size and casting return types can cause stack corruption. IOCTL bool param handling in `_io_cdc_serial_ioctl()` corrected. IOCTL bool param handling in `NANDFLASH_IOCTL_WRITE_PROTECTED` ioctl + minor changes in esai example have been made.
- MQX-4151 PTC28 wrong definition in Vybrid lwgpio fixed by correct replacement of LWGPIO pin definitions.
- MQX-4146 MFS – File size is not updated in file handler after successful writing on MFS device. Problem solved by unconditionally update of `SIZE` in `MQX_FILE_PTR`.
- MQX-4114 Function `_task_get_parameter_for` does not check the invalid input parameter. The issue was solved by added error check into function `_task_get_parameter_for()`.
- MQX-3915 FlexCan shutdown does not work correctly. The issue was solved by adding destroy lwevent in shutdown process, some functions to uninstall interrupt and disabling interrupt in function `flexcan_shutdown()`.
- MQX-3883 UART – Miss ioctl function `IO_IOCTL_SERIAL_GET_CONFIG` solved by added implementation of missing functionality.
- MQX-3889 Invalid vector number passed to KLOG ISR functions in “dispatch\_gic.s”. Problem was solved by fix of the vector number passing to both `_klog_isr_start_internal` function and `_klog_isr_end_internal` function in A5 gic interrupt handler fixed.
- MQX-3887 I2C – Ioctl function still return OK when call it with invalid parameter. Solved by added `IO_ERROR_INVALID_IOCTL_CMD` return value in case of wrong IOCTL command into driver.
- MQX-3881 UART – Miss the operation with “RX\_NOISE\_ERRORS” in uart statistical information was fixed.
- MQX-3061 FLEXCAN – The file “kflexcan.c” has bug when it tries to write `CLKSRC` bit to 1 was resolved.
- MQX-1110 RTCS/HttpServer – Wrong access of data in session buffer while searching for SSI solved.
- MQX-4387 RTCS- `select()` does not return activity if `fd_set` has zero socket handles. Solved with first non-zero `nfds` sockets in each `fd_set` are examined for activity.
- MQX-4265 RTCS - function `getaddrinfo()` fails if address is specified solved.
- MQX-4461 RTCS - Httpsrv chatclient disconnect with transferring big amount of characters. Problem was resolved by ignoring data without ID instead of closing connection.
- MQX-4196 RTCS - C++ includes wrapper missing from MQX files. Resolved with `_cplusplus` condition to all RTCS header files.
- MQX-4019 TCP/IP – Unhandled Interrupt generated when using IPv6. The Issue was solved with added missed parameter-structure deletion from the parameter-structure list, when ping6 timeout occurs.
- MQX-3984 RTCS – Bug in sending TCP packets causes dropped connections through firewalls. Problem was solved with unconditional sending MSS and window size with SYN.
- MQX-3896 RTCS/Shell – Exec command doesn't work properly in TWR\_VF65GS10 in IAR. The issue was solved with remove bootloader (srec, bin, etc) support from RTCS.
- MQX-1353 RTCS/ICMP – ICMP redirect to invalid gateway is not rejected. The issue was solved with added additional check for routing address.
- MQX-1086 RTCS – The issue with TFTP client returning wrong message if file is not found was resolved.

This package contains applications demonstrating kernel, peripheral, Ethernet, USB functionality on the X-SVF522R-EVB and TWR-VF65GS10 development boards. The applications can be found at these locations:

- `<install_dir>/mqx/examples`: Standard set of examples for kernel features and basic peripheral drivers
- `<install_dir>/mfs/examples`: Example applications demonstrating the MFS file system features
- `<install_dir>/rtcs/examples`: Example applications demonstrating the Ethernet system features
- `<install_dir>/usb_v2/host(device)/example`: Examples demonstrating USB stack features and class drivers
- `<install_dir>/demo`: Various demo application showing more complex examples

### 3 Release Content

This release contains these directories:

**Table 1. Release Contents**

Deliverable	Location
Configuration Files and Mass-Build Projects	<install_dir>/config/...
Configuration and mass-build project for all supported boards	.../config/<board>
MQX PSP, BSP Source Code, and Examples	<install_dir>/mqx/...
MQX PSP source code for Kinetis/Vybrid ARM Cortex-M core	.../mqx/source/psp/cortex_m
MQX PSP source code for Kinetis/Vybrid ARM Cortex-A core	.../mqx/source/psp/cortex_a
MQX PSP source code for ColdFire	.../mqx/source/psp/coldfire
MQX PSP build projects	.../mqx/build/<compiler>/psp_<board>
MQX BSP source code	.../mqx/source/bsp/<board>
MQX BSP build projects	.../mqx/build/<compiler>/bsp_<board>
RTCS source code and examples	<install_dir>/rtcs/...
RTCS source code	.../rtcs/source
RTCS build projects	.../rtcs/build/<compiler>/rtcs_<board>
RTCS example applications	.../rtcs/examples
MFS source code and examples	<install_dir>/mfs/...
MFS source code	.../mfs/source
MFS build projects	.../mfs/build/<compiler>/mfs_<board>
MFS example applications	.../mfs/examples
USB Host driver source code and examples	<install_dir>/usb/host/...
USB Host source code and class drivers	.../usb_v2/host/source
HUB Class Driver	.../usb_v2/host/source/classes/hub
Audio Class Driver	.../usb_v2/host/source/classes/audio
Personal Healthcare Device Class (PHDC) Driver	.../usb_v2/host/source/classes/phdc
Human Interface Device (HID) Class Driver	.../usb_v2/host/source/classes/hid
Mass Storage (MSD) Class Driver	.../usb_v2/host/source/classes/msd
Printer Class Driver	.../usb_v2/host/source/classes/printer
CDC Class Driver	.../usb_v2/host/source/classes/cdc
USB Host build projects	.../usb_v2/host/build/<compiler>/usbh_<board>
USB Host example applications (HID, MSD, HUB)	.../usb_v2/host/examples
USB Device drivers source code and examples	<install_dir>/usb/device/...
USB Device source code	.../usb_v2/device/source
USB Device build projects	.../usb_v2/device/build/<compiler>/usbh_<board>
USB Device example applications (HID, MSD, CDC, PHDC)	.../usb_v2/device/examples
Shell Library Source Code	<install_dir>/shell/...

*Table continues on the next page...*

**Table 1. Release Contents (continued)**

Shell source code	.../shell/source
Shell build projects	.../shell/build/<compiler>/ shell_<board>
Build tools plug-ins	<CodeWarrior_dir>/...
FFS source code and examples	<install_dir>/ffs/...
FFS project	ffs/build/<compiler>/<board>
FFS source code	ffs/source
MFS on FFS example	ffs/examples/mfs_nandflash
IAR Task Aware Debugging plugin (TAD)	.../tools/iar_extensions/
PC Host tools	<install_dir>/tools
BSP cloning wizard	.../tools/BSPCloningWizard/ BSPCloningWizard.exe
TFS Make Utility	.../tools/mktfs.exe
Check for Latest Version tool	.../tools/webchk.exe
AWK interpreter (GNU General Public License)	.../tools/gawk.exe
SNMP code generation scripts	.../tools/snmp/*.awk
Timing HTML report tool (for mqx/examples/benchmrk/timing)	.../tools/timing.exe
Code size HTML report tool (for mqx/examples/benchmrk/codesize)	.../tools/codesize.exe
TAD string and configuration files	.../tools/tad
Demo Applications	<install_dir>/demo
Various demo applications demonstrating complex MQX functionalities.	.../demo/...
Documentation	<install_dir>/doc
User Guides and Reference Manuals for MQX RTOS, RTCS, MFS, I/O Drivers, USB etc.	.../doc

**NOTE**

<compiler> can be IAR, GCC and DS-5

## 4 MQX RTOS Release Overview

The Freescale MQX RTOS is intended for Vybrid and Kinetis Microcontrollers. The release consists of:

- MQX RTOS real time kernel and system components
- TCP/IP networking stack (RTCS)
- FAT file system (MFS)
- NAND flash file system (FFS)
- USB Host and Device stacks
- Platform and Board support packages
- I/O drivers

This table shows the availability of various components and I/O drivers on supported boards.

	MQX PSP+BSP Libraries	MFS Library (FAT file System)	RTCS Library (TCP/IP Stack)	Shell Library	USB_v2 Host Library	USB_v2 Device Library	Flash File System (NAND)	UART (polled and interrupt driven)	I2C (polled and interrupt driven)	SPI	LWGPIO	HW Timer (PIT, SysTick, GPT)	LWADC	Audio driver I2S or SAI	QuadSPI	FLASHX	NAND flash driver	ESDHC	SD card driver (SPI or SDHC based)	RTC, IRTC (Real Time Clock)	DCU	FlexCAN/msCAN	Ethernet driver
<b>Platform</b>																							
TWRVF65GS10_A5	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
TWRVF65GS10_M4	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SVF522REVB_A5	•	•	1)	•	•	•	2)	•	•	•	•	•	•	•	•	•	2)	•	•	•	•	•	•
SVF522REVB_M4	•	•	1)	•	•	•	2)	•	•	•	•	•	•	•	•	•	2)	•	•	•	•	•	•

**Figure 1. MQX RTOS Release Overview**

**NOTE**

Items in red represent new features in this release

1. Onchip Ethernet not available, RTCS can be used with PPP or custom ENET driver (e.g., Wi-Fi over SPI)
2. Driver is supported but no example exists for platform.

## 4.1 Example applications

The examples are written to demonstrate the most frequently used features of the Freescale MQX RTOS.

In addition to these demo applications, there are simpler example applications available in MQX RTOS, RTCS, MFS, and USB directories.

The tables summarize all demo and example applications provided in this release.

### MQX RTOS Example Applications

**Table 2.** `mqx/examples`

Name	Description
benchmrk	Contains benchmarks codes for timing and code size for different components.
bootloader_vybrid	Shows how to load application images from FAT system on SD card or flash QSPI boot loader and application images to QSPI Flash memory with SD boot loader.
bootloader_vybrid_qspixip	Shows how the QSPI boot loader boots from the QSPI flash memory and loads application images from FAT file system on SD card or the raw image data from the flash memory.
can/flexcan	Shows usage of FlexCAN API functions to transmit and receive CAN frames.
clkapi	Shows usage of the clktree management APIs.
cplus	Shows simple C++ application.

*Table continues on the next page...*

**Table 2.** `mqx/examples` (continued)

Name	Description
dcu4	Shows the use of the DCU driver functionality on multiple layers: display, event handling, timing update, and alpha blending.
demo	Shows MQX RTOS multitasking and inter-process communication using standard objects like semaphores, events, or messages. See <code>lwdemo</code> for the same example using the lightweight objects.
esai_asrc	Shows how to play back/record by using ESAI and how to enable ASRC for sample rate conversion while playing ESAI audio.
event	Simple demonstration of MQX RTOS events.
fbdev	Shows the use of the abstract FBdev driver functionality on the frame buffer configuration, multiple buffers flipping, and VSYNC waiting.
flashx	Demonstration of FlashX driver functionality.
ftm	Demonstrates usage of the FTM Quadrature Decoder Driver.
gpio	Shows usage of LWGPIO driver to control on-board LEDs and switches.
hello	A trivial Hello World application using a single task.
hello2	A trivial Hello World application spread across two tasks.
hwtimer	Shows usage of HW timer driver abstraction. Demonstrates how to initialize HW timer for various modules, set frequency, callback, start, and stop the timer.
i2c	Shows how to read/write data from/to external EEPROM. Additional HW setup is needed.
i2c_slave	Shows usage of the I2C driver in slave mode – emulates the external EEPROM behavior. The current version of the MQX RTOS I/O driver supports two I2C IP modules. The legacy module requires an additional configuration. To confirm which IP you are using, see the figure in the MQX RTOS Release Overview section.
i2s_demo	Demonstrates use of audio I2S driver. TWR-AUDIO card is needed to run this example.
io	Demonstrates use of an alternate UART port as a console output.
io_expander	Shows how to operate a pin on the I/O Expander chip.
ipc	UART-based inter-processor communication demonstration.
irda	Demonstrates use of IrDA driver for transmit and receive.
isr	Shows how to install an interrupt service routine and how to chain it with the previous handler.
klog	Shows kernel events being logged and later the log entries dumped on the console.
log	Shows the application-specific logging feature.
lowpower_vybrid	Shows Vybrid circling through different power modes (RUN / SLEEP / WAIT / STOP).
lwadc	Shows usage of the ADC driver, sampling analog values from the two ADC channels.
lwdemo	Same as the "demo" application, but implemented using lightweight components only.
lwevent	Simple demonstration of MQX RTOS lightweight events.
lwlog	Simple demonstration of MQX RTOS lightweight log feature.
lwmsgq	Simple demonstration of MQX RTOS lightweight inter-process messaging.
lwsem	Simple demonstration of MQX RTOS task synchronization using the lightweight semaphore object.
lwsem_usr	Simple demonstration of MQX RTOS user/privilege task synchronization using the lightweight semaphore object.
msg	Simple demonstration of MQX RTOS inter-process message passing.
multicore	Shows usage of the multicore communication components.
mutex	Simple demonstration of MQX RTOS task synchronization using the mutex object.

*Table continues on the next page...*



**Table 2.** `mqx/examples` (continued)

Name	Description
nandflash	A demonstration of the NAND Flash driver functionality.
nill	Even simpler than Hello World. A void application which may be used for copy/paste to start custom application.
qspi	Demonstrates basic operation of QuadSPI driver, interfacing to QSPI flash.
rs485	Shows how to use the RS485 over a serial driver.
rtc	Shows the Real Time Clock module API. Demonstrates how to synchronize RTC and MQX RTOS time and how to use RTC alarm interrupts.
sai_dma_demo	Shows the use of the DMA driven SAI driver. TWR-AUDIO-SGTL board is needed for this example.
sai_duplex_demo	Shows the duplex feature of the SAI driver. TWR-AUDIO-SGTL board is needed for this example.
sem	Simple demonstration of MQX RTOS task synchronization using the semaphore object.
spi	Shows how to read/write data from/to external SPI EEPROM. Additional HW setup is needed.
taskat	Shows how task can be created within statically allocated memory buffer (avoid heap allocation for task stack and context).
taskq	Shows custom task queue and how the queue can be suspended and resumed.
test	Shows the self-testing feature of each MQX RTOS component.
tfs	Shows the usage of ROM-based Trivial File System in an MQX RTOS application.
timer	Simple demonstration of MQX RTOS timer component.
watchdog	Simple demonstration of the MQX RTOS task timeout detection using the kernel (not to be confused with watchdog) component.

**RTCS Example Applications****Table 3.** `rtcs/examples/...`

Name	Description
eth_to_serial	Simple character passing between the UART console and the telnet session. Shows custom "lightweight" telnet.
httpsrv	Simple web server with CGI-like scripts and web pages stored in internal flash.
shell	Shell command line providing commands for network management.
snmp	SNMP protocol example providing microprocessor state information.

**FFS Example Applications****Table 4.** `ffs/examples/mfs_nandflash/...`

Name	Description
mfs_nandflash	Console shell-based example showing how to access an MFS filesystem mounted on the NAND flash memory.

**MFS Example Applications**
**Table 5.** `mfs/examples/...`

Name	Description
ramdisk	Shows use of MFS accessing the external RAM (or MRAM).
sdcard	Shows use of MFS accessing the SDHC or SPI-connect SD Card.

**USB Host Example Applications**
**Table 6.** `usb_v2/host/examples/...`

Name	Description
audio/microphone	Enables connecting a USB microphone and record the sound to SD Card (wav format!).
audio/speaker	Enables connecting a USB speaker and play the sound from SD Card (wav format).
cdc/cdc_serial	This example demonstrates the virtual serial port capability with abstract control model. Redirects the communication from CDC device, which is connected to the board, to the serial port.
hid/keyboard	This application echoes keys pressed on the USB keyboard onto the serial console.
hid/mouse	Displays USB mouse events on the serial console.
hid/keyboard+mouse	Keyboard and mouse demos combined in a single application.
msd/msd_cmd	Executes the standard "mass storage device" commands to the USB disk and shows the response on the serial console (see MFS examples for USB filesystem access).
msd/msd_fatfs	Console shell-based example showing how to access an MFS filesystem mounted on the u-disk memory.
phdc/11073Manager_Demo	This application demonstrates basic host personal healthcare class functionality.

**USB Device Example Applications**
**Table 7.** `usb_v2/device/examples/...`

Name	Description
audio/generator	Acts as a USB microphone, playing out a short audio loop.
audio/speaker	Receives audio stream data from the host (PC) and plays it out through the I2S driver.
cdc/virtual_com	Implements a virtual serial line loopback.cdc/virtual_nicImplements a virtual network interface cards.
cdc/virtual_nic	Implements a virtual network interface cards.
composite/hid_audio	Shows basic functionality of composite device using hid and audio classes.
composite/msd_cdc	Shows basic functionality of composite device using mass storage and CDC classes.
hid/mouse	Creates a virtual mouse which keeps moving in a square loop, 100 pixels in size.
hid/hid_keyboard	Creates a virtual keyboard which can scroll the screen up and down.
msd	Implements small storage device in internal RAM memory.
phdc/weighscale	Implements personal healthcare device.

## Demo Applications

**Table 8.** demo/...

Name	Description
vadc	This example demonstrates simple configuration of the AFE, VDEC and VIU3 modules with video streaming from Vin port to DCU4.
web_hvac_v2	HVAC demo with the HTTP server implementing the GUI. Ajax-based pages demonstrating the advanced use of the HTTP server with new usb_v2 stack.

## 5 Set up installation instructions and technical support

Run the Vybrid package installer and proceed according to instructions. This package can be used independently of the current MQX 4.1.1 RTOS installation folder.

For building procedures, see *Getting Started with Freescale MQX™ RTOS* (document MQXGSRTOS).

### 5.1 Default jumper settings

These are the default jumper settings for SVF522REVB Rev. A operation:

- J1, J2, J3, J8, J14, J15, J16, J17, J18, J19 inserted.
- J4 on position 2-3
- J7, J9, P21, P23, P24 jumpers are not inserted.
- RCON switches are in off position.
- For boot selection are important J14 and J15 jumpers
  - For boot from Fuses J14 and J15 have to be in position 1-2.
  - For boot from Serial Download J14 has to be in position 1-2 and J15 in position 2-3.
  - For boot from RCON Sw J14 has to be in position 2-3 and J15 in position 1-2.
    - For Quad-SPI Flash boot SW8, SW9, SW10, SW11 all in OFF position.
    - For SD card boot SW10, SW11 all in OFF position, SW8 – switches 6 and 7 in ON position, SW9 – switch 4 in ON position.
- For USB Device mode, use onboard P9 micro USB connector and the default jumper settings.
- For USB Host mode, use onboard P1 USB 2.0 connector and the default jumper settings.
- J-link debugger populates the ARM Standard JTAG (P15) or ARM Cortex® (P16) connector.

Important Jumper Settings TWR-VF65GS10 (Rev. H)

- Serial Port Settings
  - Default: J23 (1-3 & 2-4) and (7-9 & 8-10)
  - Vybrid SCI1 (ttyb:) to OpenSDA (K20)
  - Vybrid SCI2 (ttyc:) to Elev UART1 (TWR-SER)
  - Alternative 1: J23 (1-2 & 3-4) and (7-8 & 9-10)
  - Vybrid SCI1 (ttyb:) to Elev UART1 (TWR-SER)
  - Vybrid SCI2 (ttyc:) to OpenSDA (K20)
  - Alternative 2: J23 (2-4) and (8-10)
  - Vybrid SCI1 (ttyb:) to Elev UART0 (0Rs connected)
  - Vybrid SCI2 (ttyc:) to Elev UART1 (TWR-SER)
- USB0 (EHCI0) Module
  - Default BSP setting – use USB0 port `#define USBCFG_DEFAULT_DEVICE_CONTROLLER (&_bsp_usb_dev_ehci0_if)#define USBCFG_DEFAULT_HOST_CONTROLLER`

- (`&_bsp_usb_host_ehci0_if`) in the `<install_dir>/mqx/source/bsp/twrvf65gs10_a5/twrvf65gs10_a5.h`
  - USB0 connector setting (TWR-SER USB Mini-AB vs. on-board USB Micro-AB connector)
  - USB0 host mode
  - TWR-VF65GS10 board – J19 1-2
  - TWR-VF65GS10 board – J20 1-2
  - USB0 device mode
  - TWR-VF65GS10 board – J20 2-2
- USB1 (EHCI1) Module
  - To enable USB1 (EHCI1), change the BSP settings as follows and recompile the BSP. The USB1 (EHCI1) is connected to USB Standard-A. `#define USBCFG_DEFAULT_DEVICE_CONTROLLER`  
`(&_bsp_usb_dev_ehci1_if)#define USBCFG_DEFAULT_HOST_CONTROLLER`  
`(&_bsp_usb_host_ehci1_if)` in the `<install_dir>/mqx/source/bsp/twrvf65gs10_a5/twrvf65gs10_a5.h`
- USB1 host mode
  - TWR-VF65GS10 board – J19 1-2
  - TWR-VF65GS10 board – J21 1-2

## 5.2 Board-specific Build Targets

Internal Ram (Debug and Release), DDR ram (Debug and Release) - These targets enable building applications suitable for booting the system from the Internal Ram memory.

## 6 Known Issues

- TWR-VF65GS10
  - A freeze bit in the MCR register allows the PIT timers to stop when the device enters debug mode. When either core is stopped by a debugger, the PIT timer stops too. For the multicore debugging, this process may cause confusion.
  - CAN functionality requires the R10 and R11 to be connected.
  - Both ARM Cortex-A5 and ARM Cortex-M4 Int\_Ram target code space is limited and most of the RTCS-based applications are too large to fit into the memory. Using DS-5 with CMSIS-DAP to download more than 128 KB of code into SRAM memory is not working correctly. This issue is caused by a limitation of the CMSIS-DAP firmware. See the [Updated CMSIS-DAP application issue](#).
- SVF522REVB
  - Both ARM Cortex-A5 and ARM Cortex-M4 Int\_Ram target code space is limited and most of the RTCS-based applications are too large to fit into the memory.
  - Lowpower\_vybrid example M4 core doesn't stay in sleep/stop mode and wakes up immediately.
  - Before testing USB svf522revb\_m4 demos, make these changes in the `user_config.h` file, and rebuild all libs.

```
#define BSPCFG_ENABLE_ASRC 0
#define BSPCFG_ENABLE_ESAI 0
#define BSPCFG_ENABLE_ESAI_TX0 0
#define BSPCFG_ENABLE_ESAI_RX0 0
#define BSPCFG_ENABLE_IO_EXPANDER_MAX7310 0
```

- Because of the the conflict between the boot and the demos, the Vybrid M4 core can't be executed while debugging when the IP EHCI\_0 is used in the USB device mode. To resolve this, use the A5 bootloader while loading the Vybrid M4 demo.
- The FTM incremental encoder example doesn't count by one step after turning up/down. The `ftm_quaddec_isr` occurs a few times in one turn because of the incorrect hardware design.

- Vybrid RTC driver doesn't support counter rollover at default. An additional RTC rollover interrupt can be set to handle counter maximum value.
- SVF522REVB\_M4 and TWR\_VF65GS10\_M4
  - Known issue in experimental phase implementation of TLSF allocator. To use them with K70 and Vybrid platforms, modify line 673 in tlsf.c to:

```
const size_t pool_bytes = align_down(bytes - pool_overhead, alignment);
    and lines 1307-8 in tlsf_adaptation.c to:
        +      tlsf_add_pool(tlsf_pool_ptr, start + block_header_overhead,
(unsigned char *) CORTEX_MEMORY_BARRIER_ADDR -
        (unsigned char *) start - (PSP_MEMORY_ALIGNMENT + 1) -
block_header_overhead, (PSP_MEMORY_ALIGNMENT + 1));
        +      tlsf_add_pool(tlsf_pool_ptr, CORTEX_MEMORY_BARRIER_ADDR +
block_header_overhead, (unsigned char *) end -
        (unsigned char *) CORTEX_MEMORY_BARRIER_ADDR -
(PSP_MEMORY_ALIGNMENT + 1) - block_header_overhead, (PSP_MEMORY_ALIGNMENT + 1));
```

**How to Reach Us:****Home Page:**[freescale.com](http://freescale.com)**Web Support:**[freescale.com/support](http://freescale.com/support)

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [freescale.com/SalesTermsandConditions](http://freescale.com/SalesTermsandConditions).

Freescale, the Freescale logo, Kinetis, CodeWarrior, ColdFire, and Processor Expert are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Vybrid and Tower are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2014 Freescale Semiconductor, Inc.

Document Number MQXRN  
Revision 4.1.2, 11/2014

