

Getting Started with Freescale MQX™ RTOS for Kinetis SDK and Kinetis Design Studio IDE

1 Overview

This section describes the steps required to configure KDS to build, run, and debug MQX™ RTOS demo applications and necessary driver libraries provided in the KSDK framework. The Hello World demo application targeted for the TWR-K64F120M Tower System hardware platform is used as an example in this guide.

Contents

1	Overview	1
2	Installing KSDK Eclipse update	2
2.1	Installing MQX RTOS Task Aware Debugger for GDB plug-in (TAD)	3
3	Building MQX RTOS example project and libraries ..	5
3.1	Import example project and libraries to workspace	5
3.2	Build the libraries	7
3.3	Build demo application.....	9
4	Run the demo application.....	10
4.1	Run demo application	10
4.2	Using MQX RTOS Task Aware Debugger for GDB plug-in.....	13
5	Revision history	16

2 Installing KSDK Eclipse update

Before using any Eclipse-based IDE with KSDK, the KSDK Eclipse update must be applied. Without this update, Eclipse cannot generate KSDK-compatible projects. To install the update, follow these instructions:

1. Select “Help” then “Install New Software” in the toolbar menu.

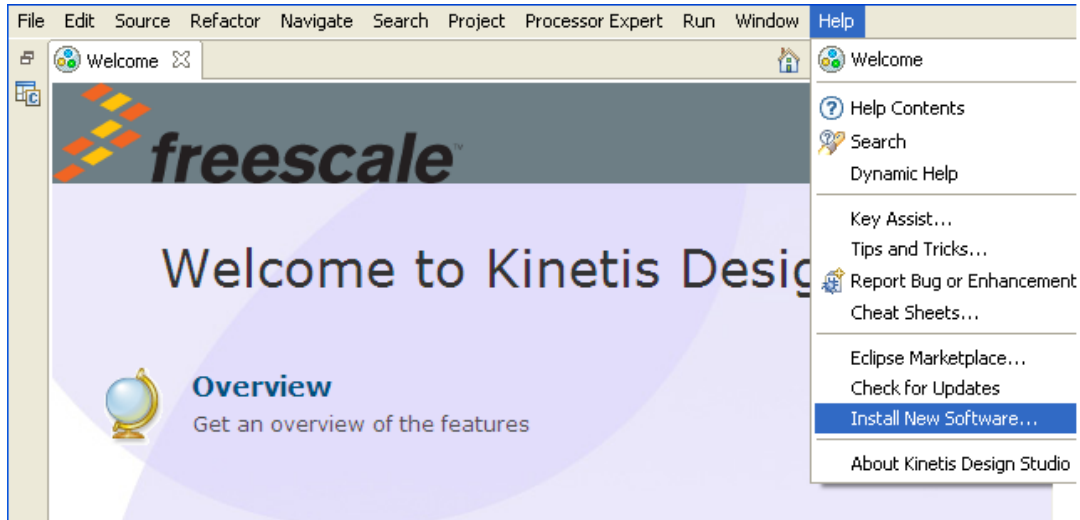


Figure 1 Install new software

2. In the Install dialog box, click the “Add” button in the upper right corner. The “Add Repository” pop up window will open. Click the “Archive” button to browse to this folder:
`<KSDK_install_dir>/tools/eclipse_update`
3. Select the KSDK_<version>_Eclipse_Update.zip file if it exists. If there is no archive file, simply cancel the process because no update needs to be installed. Refer to section 3.
4. Click “Open” then click the “OK” button. The KSDK update shows up in the list of the original install dialogs.
5. Check the box to the left of the KSDK Eclipse update and click the “Next” button in the lower right corner.

Follow the remaining instructions to finish the installation of the update.

6. After the update is applied, restart the KDS/Eclipse IDE for the changes to take effect.

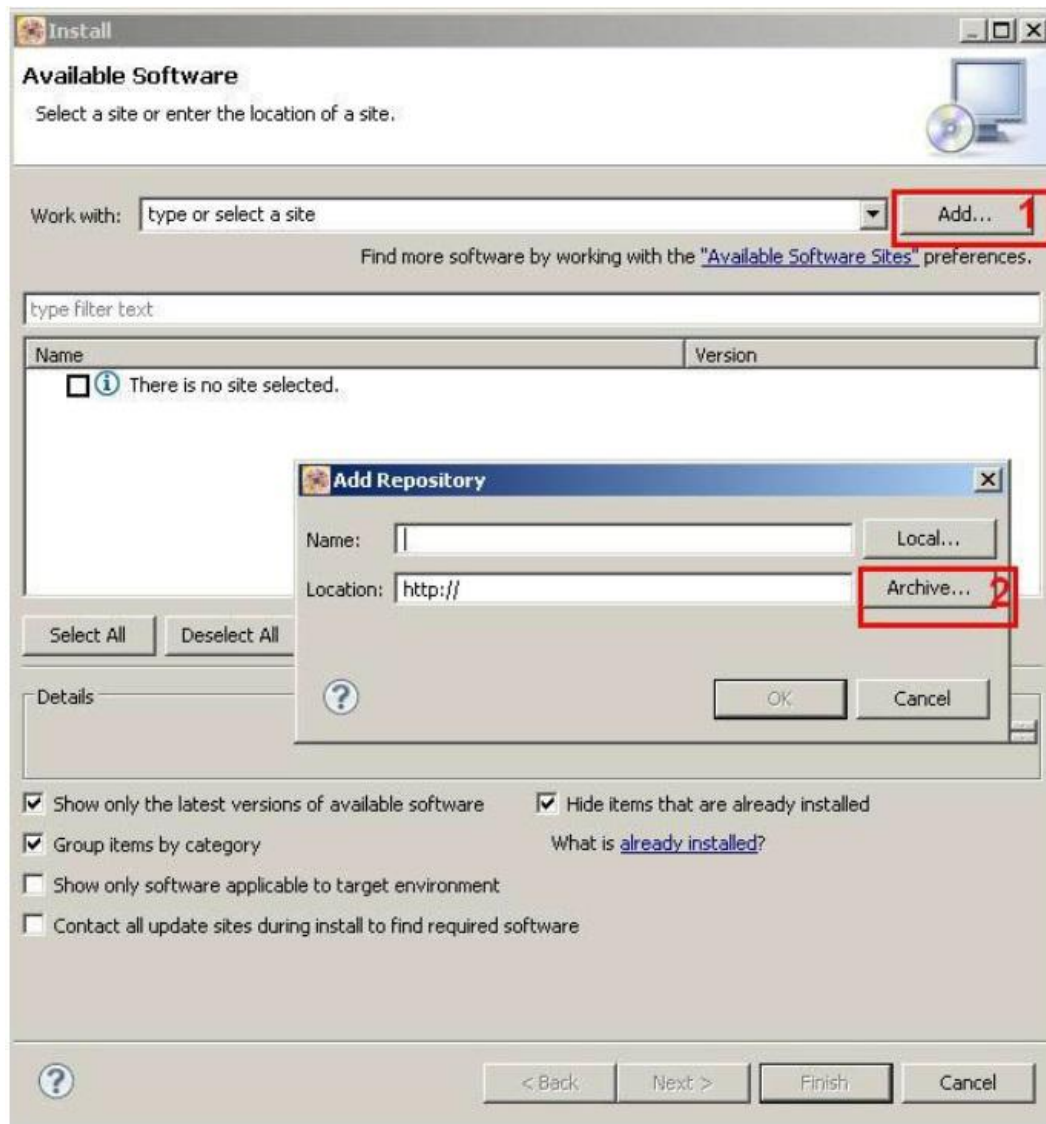


Figure 2: Browse the update file

2.1 Installing MQX RTOS Task Aware Debugger for GDB plug-in (TAD)

1. To install MQX RTOS Task Aware Debugger for GDB plug-in go again to Help, then Install New Software... in the toolbar menu.

2. In the Install dialog box, choose the Freescale Update Site from the Work with menu.

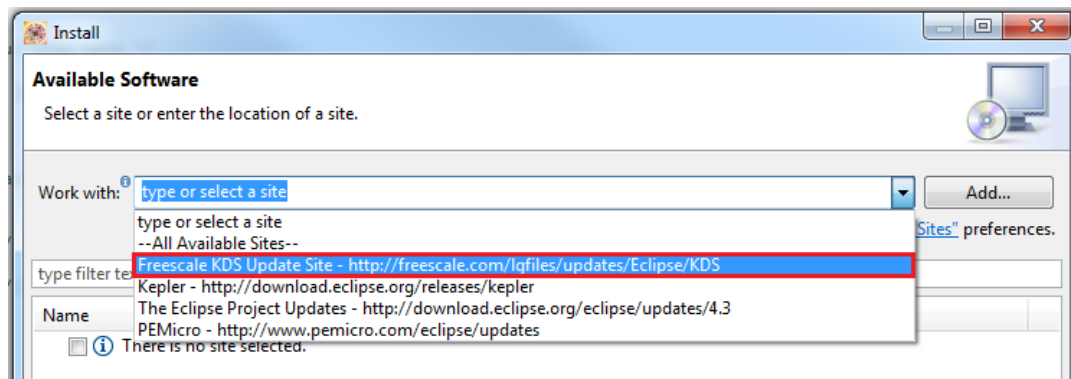


Figure 3: Freescale KDS Update Site

Now three categories from update site are displayed.

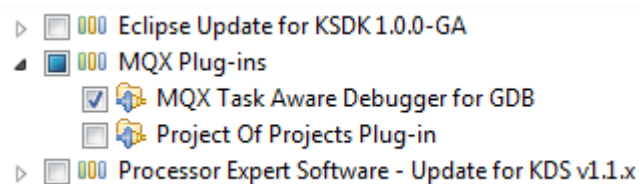


Figure 4: Categories on Update Site

3. Check the box next to MQX RTOS Task Aware Debugger for GDB in MQX RTOS Plug-ins category (Project of Project Plug-in is already a part of KDS 2.0.0 or higher).
4. Click the “Next” button and follow the remaining instructions to finish the installation TAD plug-in.
5. After the update is applied, restart the KDS/Eclipse IDE for the changes to take effect.

3 Building MQX RTOS example project and libraries

3.1 Import example project and libraries to workspace

Every demo application in MQX RTOS for KSDK has one workspace file that contains all the required library project files and the application project itself. Import the workspace file in this folder:

```
<install_dir>/rtos/mqx/mqx/examples/<demo_name>/build/kds/<demo_name>_<board_name>/<demo_name>_<board_name>.wsd
```

The KDS will import the application project file and all the necessary library project files.

1. In the Import dialog window, select the “Existing Projects Sets” file and click the “Next” button.



Figure 5: Import workspace file (feature of KDS 2.0 or higher)

2. Click the “Browse” button to load the workspace file in the specific location. Click the “Finish” button when complete. The MQX RTOS example application and associated libraries will be opened in your KDS workspace.

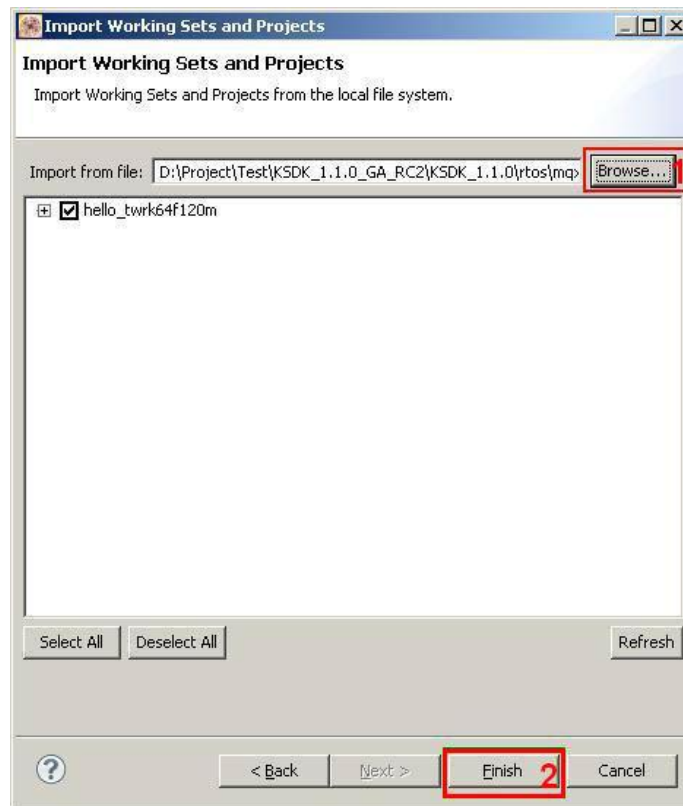


Figure 6: Import workspace file 2 (feature of KDS 2.0 or higher)

3.2 Build the libraries

Before building and debugging demo applications, all libraries must be built.

To build the library for a device, follow these instructions:

1. Select project files that you want to batch build by pressing Ctrl and left-clicking the project files. Right-click the selected projects and select “Build Configurations” then “Build Selected”.

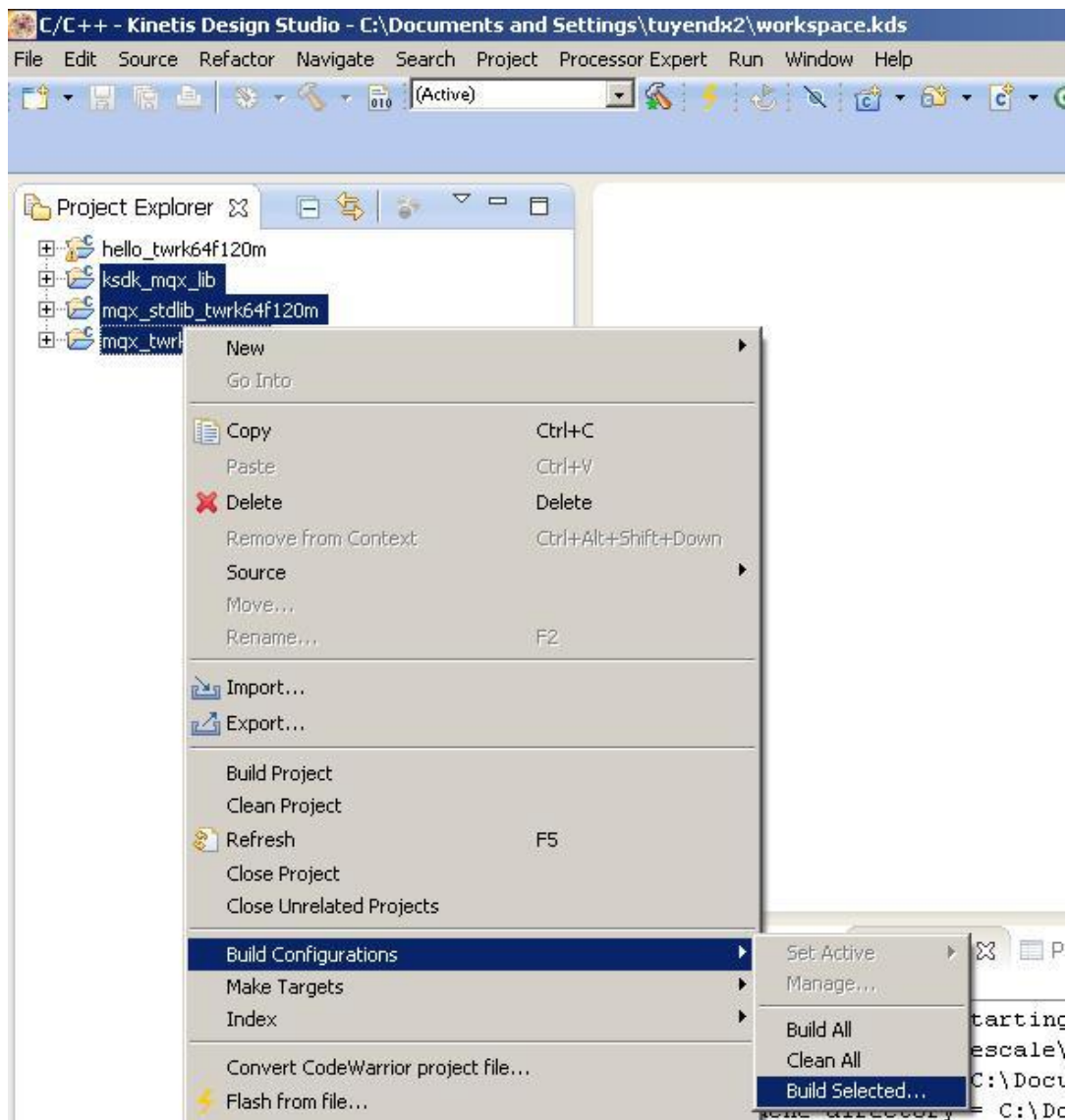


Figure 7: Select build target for the library

- When the Clean and Rebuild Configurations dialog window opens, check the projects you want to batch build. Click the “OK” button to start the batch build process.

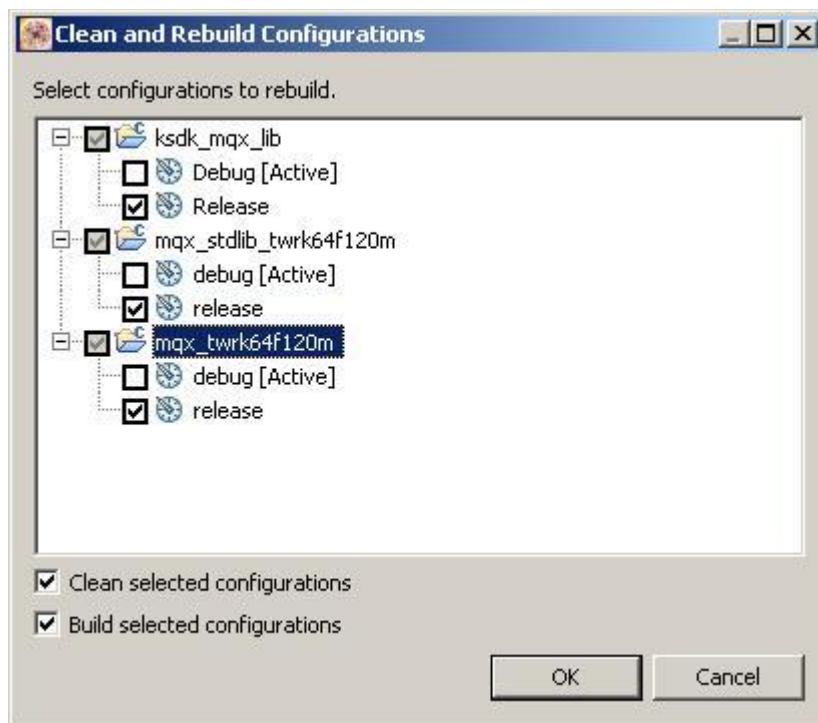


Figure 8: Select projects for batch build

The archive files are generated after a successful build of the library project files. They are located in the following folders for libraries ksdk_mqx_lib, mqx, and mqx_stdlib.

<install_dir>/lib/ksdk_mqx_lib/kds/<device_name>/<build_target>/<libksdk_platform_mqx>.a

<install_dir>/rtos/mqx/lib/<board_name>.kds/<build_target>/mqx/<lib_mqx>.a

<install_dir>/rtos/mqx/lib/<board_name>.kds/<build_target>/mqx_stdlib/<lib_mqx_stdlib>.a

3.3 Build demo application

Refer to Section 3.2 to select the build target and build project for the demo application. This figure is an example for the hello demo for TWR-K64F120M.



Figure 9: Build demo application

The “Console” tab view shows the build summary of the build for hello_world demo.

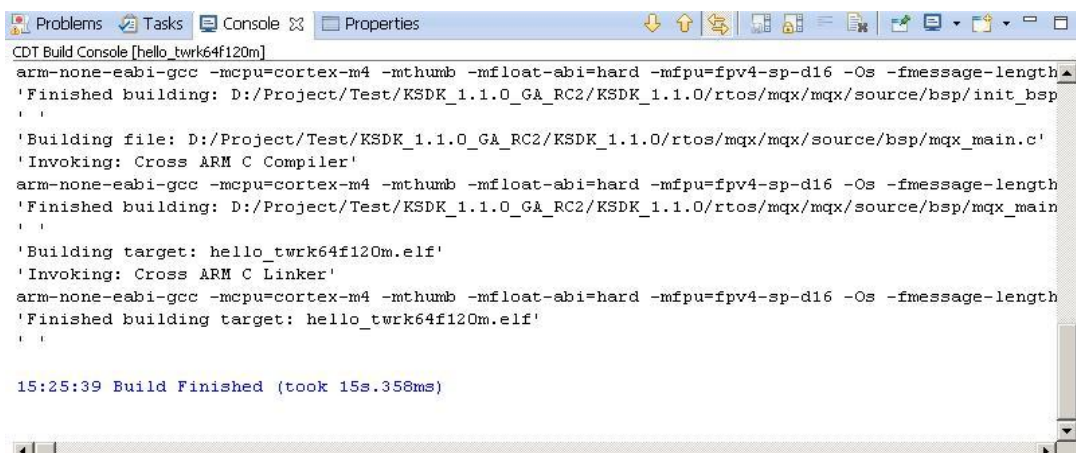


Figure 10: Build demo application output console

4 Run the demo application

4.1 Run demo application

This section describes steps to run a demo application using the J-Link GDB Server application.

To download and run the application, follow these steps:

1. Connect the development platform to your PC via USB cable between the OpenSDA USB connector and the PC USB connector.
2. Open the terminal application on the PC, such as PuTTY or TeraTerm, and connect to the OpenSDA serial port number. Configure the terminal with these settings:
 - a) 115200 baud rate
 - b) No parity
 - c) 8 data bits
 - d) 1 stop bit

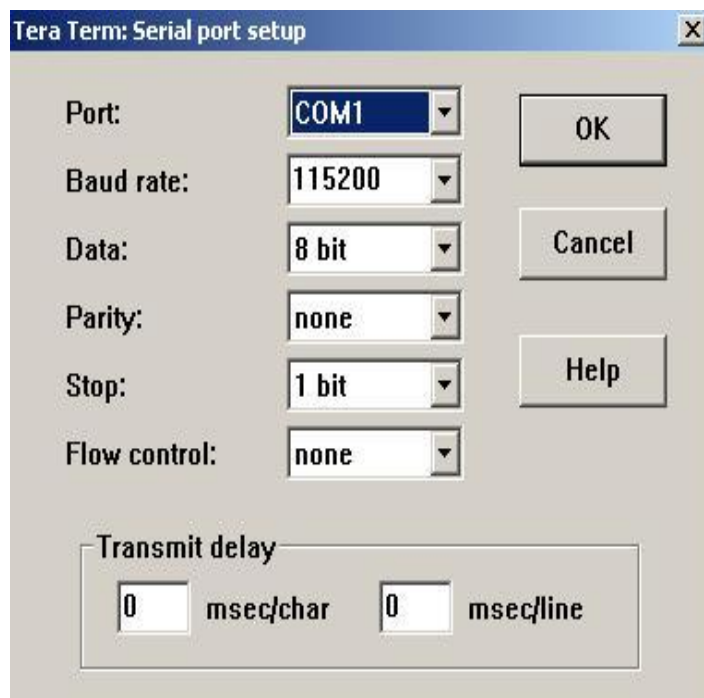


Figure 11: Terminal configuration

3. Left-click the application project, then click the bug icon to launch the debug configurations window.



Figure 12: Open debug configuration

4. In the Debug configuration window, select the appropriate debug option for your board. Most Freescale Freedom development boards support CMSIS-DAP debugger (available under GDB OpenOCD Debugging). Other provided options are the PE Micro and SEGGER J-Link, as shown in the figure. After selecting the Debugging option, click “Debug” button to start the debugging session.

The application image is loaded into the platform’s MCU, and the program runs into the main() function and stops.

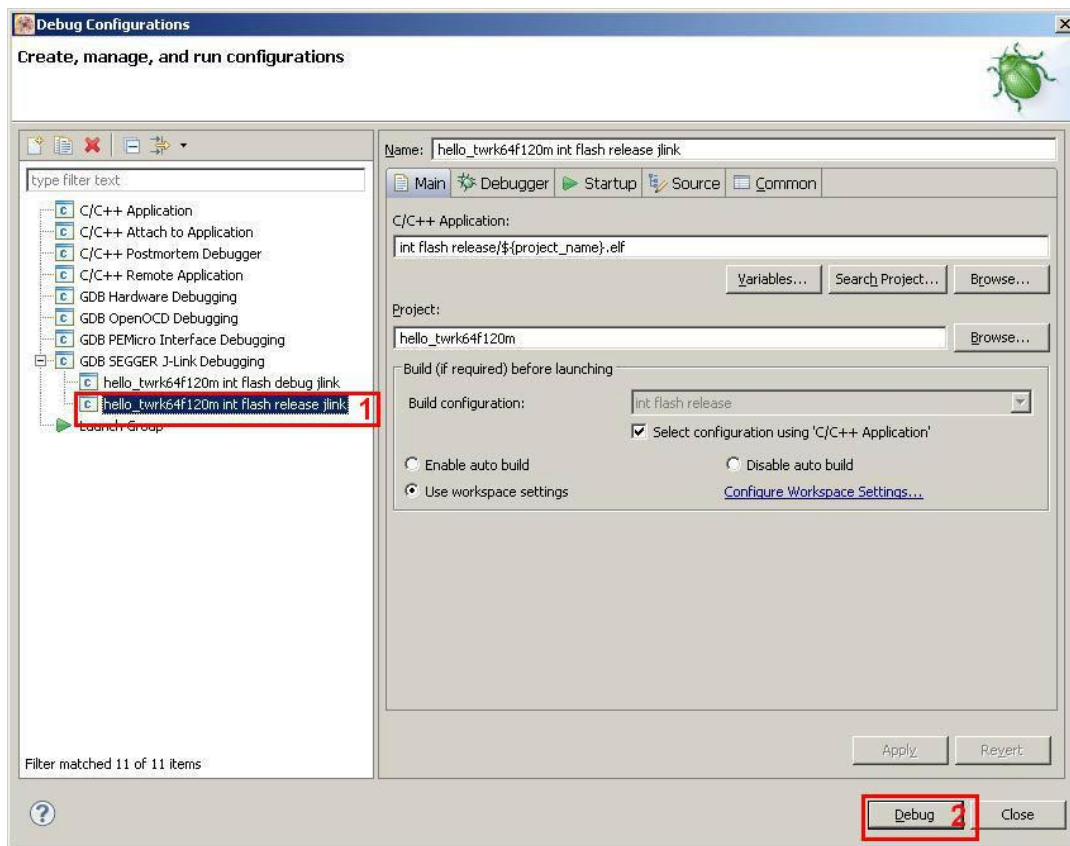


Figure 13: Run demo application

- Hit the “Resume” button (in red) to run the application.

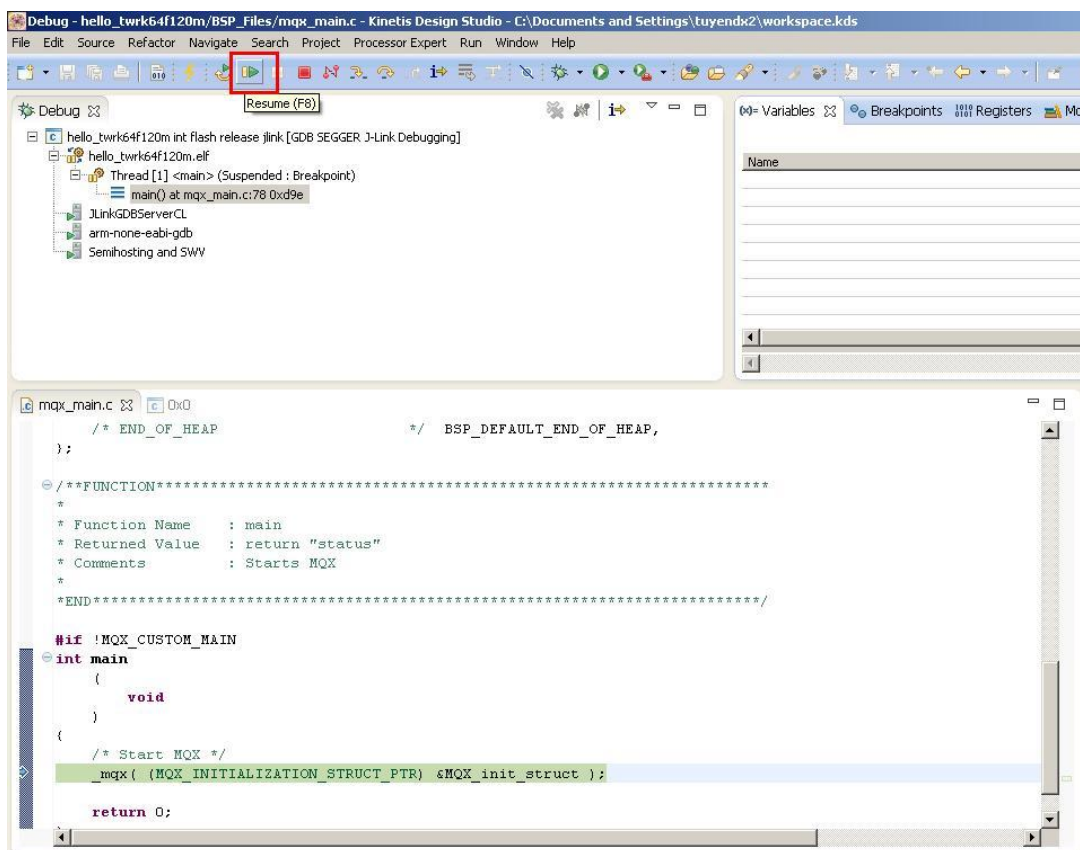


Figure 14: Run demo application – 2

The output of the hello_world demo application is shown on the terminal.

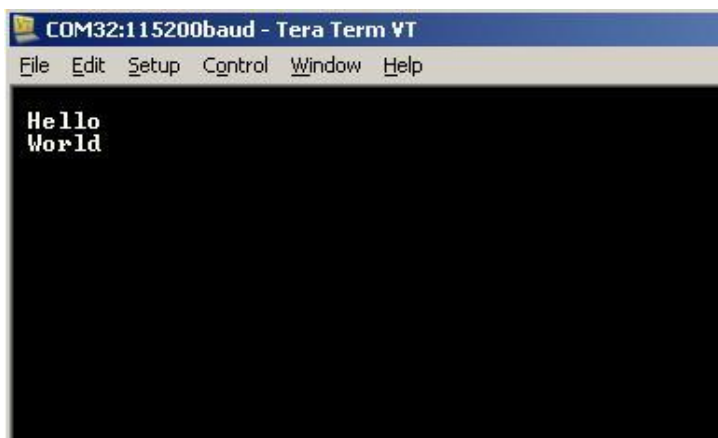


Figure 15: Output message of demo application

4.2 Using MQX RTOS Task Aware Debugger for GDB plug-in

MQX RTOS Task Aware Debugging plug-in (TAD) is an optional extension to a debugger tool which enables easy debugging of multi-tasking applications. It helps to visualize internal MQX RTOS data structures, task-specific information, I/O device drivers, and other MQX RTOS context data.

MQX RTOS TAD Screens are accessible from MQX RTOS menu which is only displayed during the debug session.

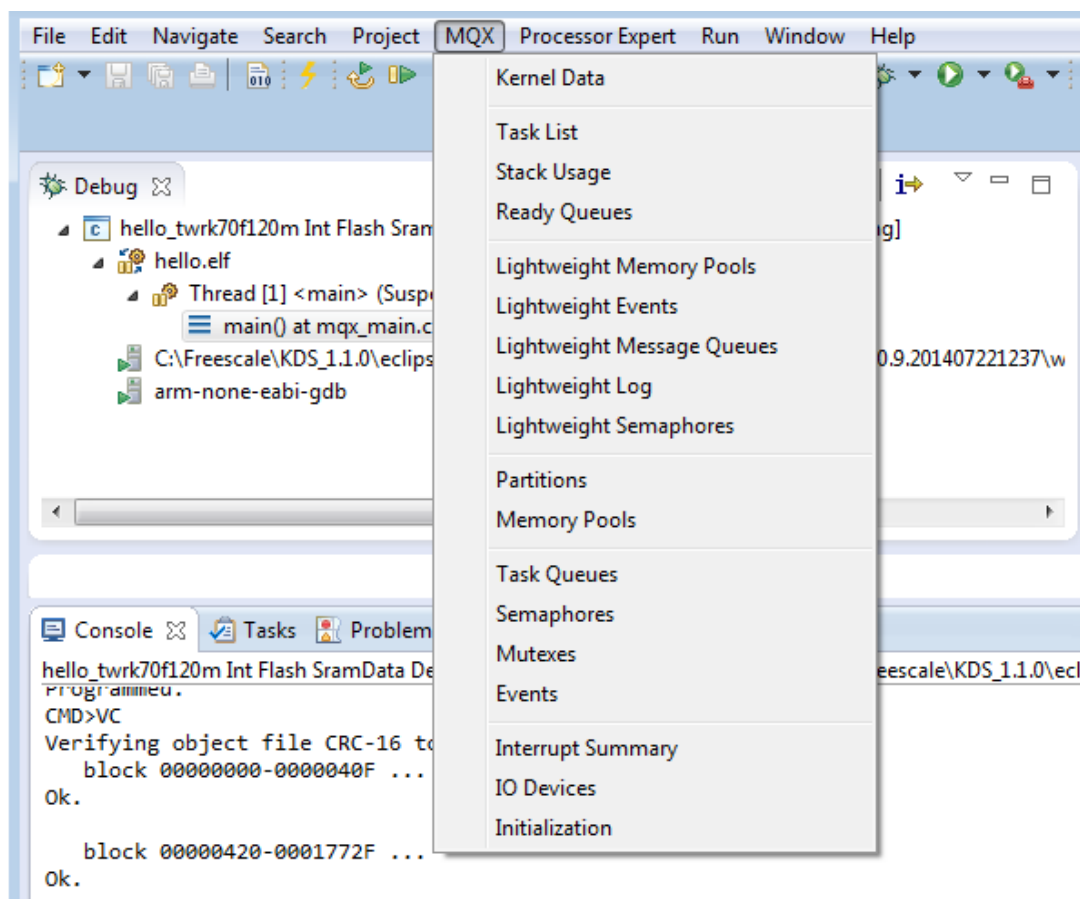


Figure 16: List of available TAD screens

Resume (press F8 or the “Run/Resume” button)  the debug session and then suspend  (of Run/Suspend) it again to initialize MQX RTOS structures needed by MQX RTOS TAD.

The most helpful and frequently used screens are shown in the images below:

- Task List – Overview about all tasks created in the MQX RTOS application

Task Name	Task ID	TD Address	Priority	State	Task Error C...
▸ _mqx_idle_task	0x10002	0x1fff1a80	10	Active	OK (0x0000)
▸ hello	0x10003	0x1fff27b0	8	Blocked	OK (0x0000)

Figure 17: MQX RTOS Task List screen

- Stack Usage – Displays information about interrupt and task stacks. Typically, a stack overflow is a root cause for vast majority of problems in MQX RTOS user applications.

Task Name	Stack Base	Stack Limit	Stack Used	% Used	Stack Overflow	Stack Usage Graph
▸ _mqx_idle_task	0x1fff1c40	0x1fff1b70	0x1fff1bc4	59.6%	No	
▸ hello	0x1fff2e50	0x1fff2880	0x1fff2b8c	47.6%	No	

Figure 18. MQX RTOS Stack Usage screen

- Memory Pools (or Lightweight Memory Pools) – Displays address, size, and type information about each memory block allocated in the selected memory pool by the MQX RTOS system or applications.

Start Address	End Address	Size (Hex)	Size (Dec)	Highest Used Address	Usage	Usage Graph
0x1fff09c0	0x2000ffc0	0x0001f630	128560 (125.5 kB)	0x2000000f	0x0000f64f	

Address	Size (Hex)	Size (Dec)	Owner	Type	Category	Remarks
0x1fff09e0	0x00000410	1040	System	Interrupt Stack	System/Interr...	
0x1fff0e10	0x000000e0	224	System	System Stack	System/Tasks	
0x1fff0f10	0x000000b0	176	System	Ready Qs	System/Tasks ...	
0x1fff0fe0	0x00000040	64	System	Interrupt Table	System/Interr...	
0x1fff1040	0x00000020	32	System	Interrupt Vec...	System/Interr...	
0x1fff1080	0x000000960	2400	Free			
0x1fff1a00	0x00000060	96	System			
0x1fff1a80	0x000000b0	176	0x10002	Task Descript...	System/Tasks	
0x1fff1b50	0x00000100	256	0x10002	Task Stack	System/Tasks	
0x1fff1c70	0x00000020	32	System	Interrupt Vec...	System/Interr...	

Figure 19: MQX RTOS Memory Pools screen

- Semaphores, Events, Mutexes (or Lightweight Semaphores, Lightweight Events) - Displays address and status of synchronization objects created by the MQX RTOS system or application. When a synchronization object is allocated either as a global or static variable in the system, or as an array element or as a structure member allocated as global or static variable, the TAD plug-in also displays the symbolic name of the object.

MQX Semaphores						
Semaphore	Owning #	Waiting #	Count	Destroy	Name	
0x1fff1190	0	2	0	No	sem.write	
0x1fff1200	0	0	2	No	sem.read	
0x1fff1270	0	0	0	No	sem.index	
Property		Value				
Semaphore ID		0x1fff1190				
Name		sem.write				
Valid?		Yes				
Destroy?		No				
Priority queuing?		No				
Priority inherit?		No				
Count		0				
Max count		Unknown				
Strict?		No				
Own/Wait:		Task:				
▸ Wait		write, ID:0x10001, TD:0x1fff12e0				
▸ Wait		write, ID:0x10004, TD:0x1fff13b0				

Figure 20: MQX RTOS Semaphores screen

5 Revision History

This table summarizes revisions to this document.

Table 1 Revision History		
Revision number	Date	Substantial changes
1	04/2015	Kinetis SDK 1.2.0 release
0	12/2014	Kinetis SDK 1.1.0 release

How to Reach Us:**Home Page:**freescale.com**Web Support:**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.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2015 Freescale Semiconductor, Inc.