Document Number: MQXKDSUG
Rev. 0, 08/2014

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

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

Document Number: MQXKSDKUG
Rev. 0, 08/2014

Table of Contents

1 Read Me First

This document describes how to use the Kinetis Design Studio (KDS) IDE for the MQX™ RTOS basic development tasks. See *Getting Started with Freescale MQX RTOS* (document MQXGSRTOS) and other user documentation included in the latest Freescale MQX RTOS installation for more information that is not specifically related to the KDS IDE tools.

Use the latest Freescale MQX RTOS available at freescale.com/mqx.

2 MQX Build – initial steps

The MQX RTOS release provides the KDS IDE native projects to more conveniently build MQX RTOS libraries and applications.

This chapter concentrates on KDS IDE-specific steps only. For details about the generic build process and compile time configuration, see Chapter 2 of the *Getting Started with Freescale MQX™ RTOS* (document MQXGSRTOS).

Install the MQX RTOS KDS IDE plug-ins using ***Help | Install New Software*** menu. Chose the ***Freescale Update Site*** from the ***Work with*** menu, and select the checkboxes next to **MQX Plug-ins** items.

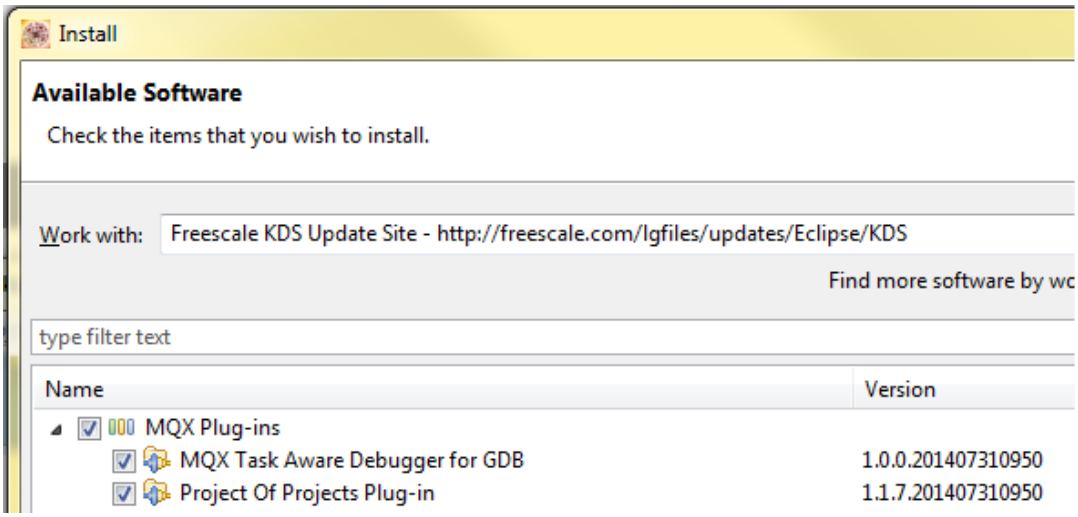


Figure 1 Install KDS IDE plug-ins

Install GDB server for Kinetis Devices application from PE Micro web site or alternative GDB server software. For a step-by-step guide on J-Link GDB server debugging see the `<mqx_install_dir>/doc/tools/gnu/MQX_GNU_Getting_Started.pdf` document.

3 Building MQX libraries

To build the MQX libraries, import the <mqx_install_dir>/build/<board>/kds/build_libs.wsd working set description file using ***File|Import|MQX|Import Working Sets*** menu. The MQX library projects will be imported to KDS IDE working space together with build configurations settings.

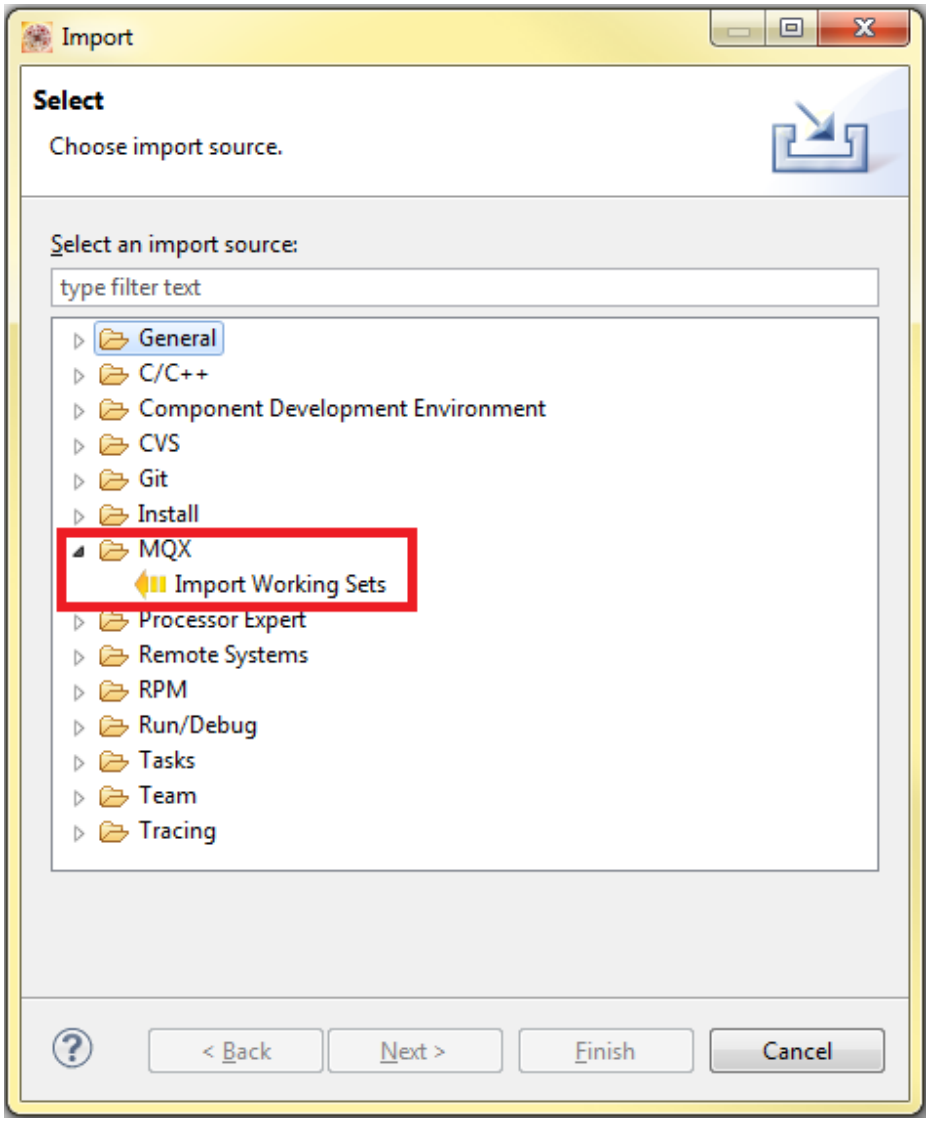


Figure 2 Import MQX library

- These projects will be imported to your workspace:

```
<mqx_install_dir>/mqx/build/kds/bsp_<board>/.project
<mqx_install_dir>/mqx/build/kds/psp_<board>/.project
<mqx_install_dir>/mfs/build/kds/mfs_<board>/.project
<mqx_install_dir>/rtcs/build/kds/rtcs_<board>/.project
<mqx_install_dir>/usb/host/build/kds/usbh_<board>/.project
<mqx_install_dir>/usb/device/build/kds/usbd_<board>/.project
<mqx_install_dir>/shell/build/kds/shell_<board>/.project
```

- Build all libraries using the KDS IDE ***Project/Build All*** menu.

4 Running and Debugging MQX applications

This description is provided for the Kinetis TWR-K64F120 BSP and Hello World example applications. The same procedure applies for all other BSPs and example applications distributed in the MQX RTOS release package.

4.1 Debugging MQX Hello World program

- Connect a USB cable to OpenSDA debug connector. Set the communication speed to 115200 in the terminal program.
- Select menu ***File/Import/General/Existing Projects into Workspace*** and import the Hello World example application.

<mqx_install_dir>/mqx/*examples/hello/build/kds/hello_twrk64f120/.project*

- Click on the compile button  to build the application ***Int. Flash Debug target***.
- Click the arrow next to the Debug button and select ***Debug Configurations***.

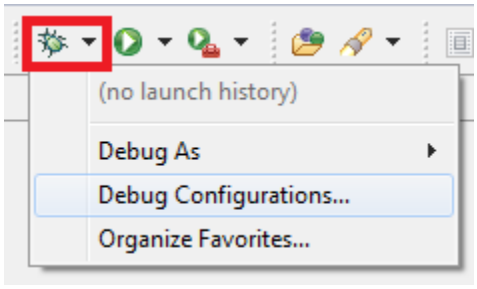


Figure 3 Debug configurations

- Select debugging using P&E Micro GDB Interface and click on the “Debug button”.

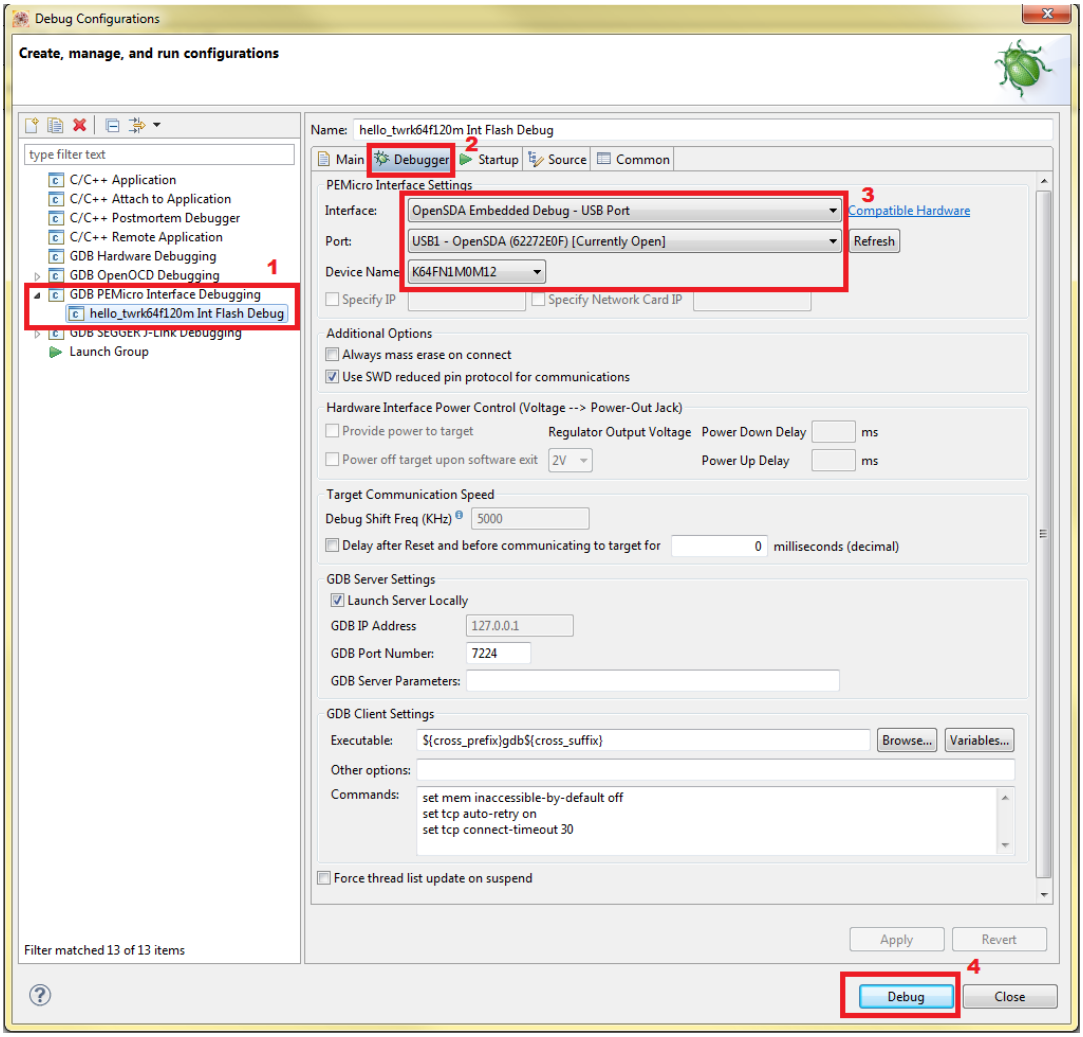


Figure 4 Debug button

Do not forget to specify Interface - OpenSDA and Device name before starting your debugger session.

The GDB Server application should flash the microcontroller (download the firmware to the target) with your application and the debugger should stop at the main function.

4.2 MQX Task Aware Debugging

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

4.2.1 Using MQX TAD Screens

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

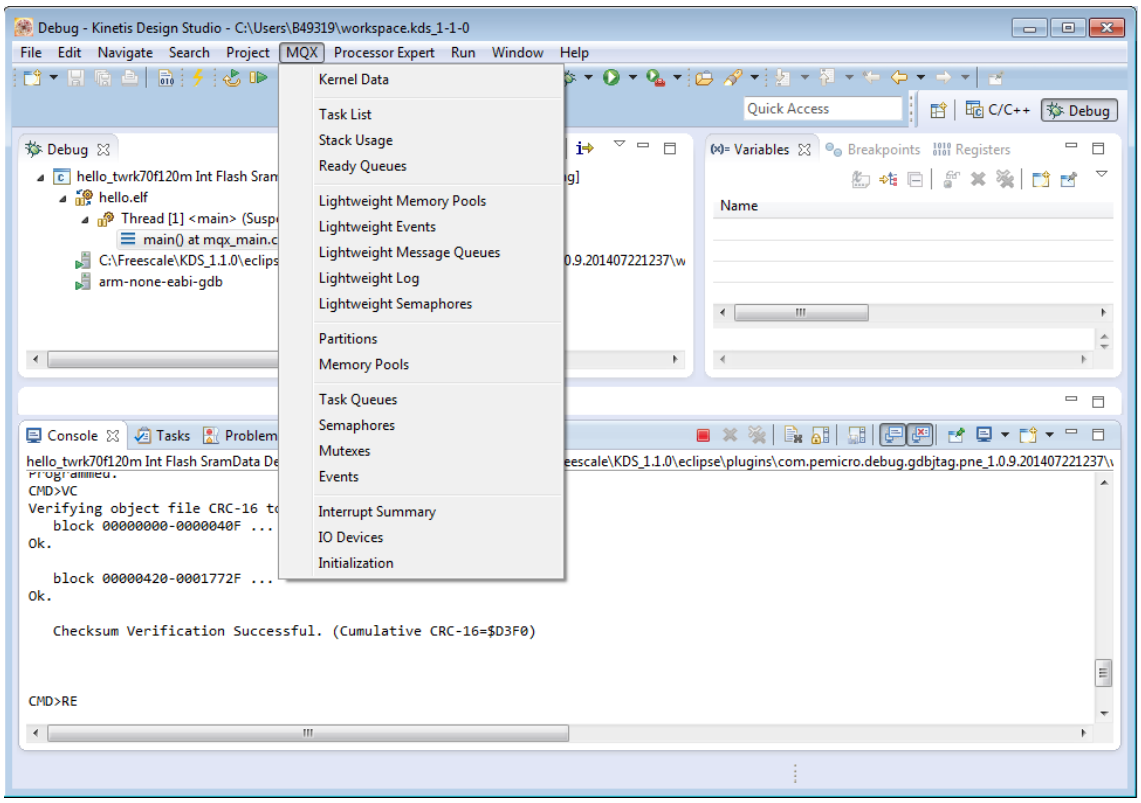
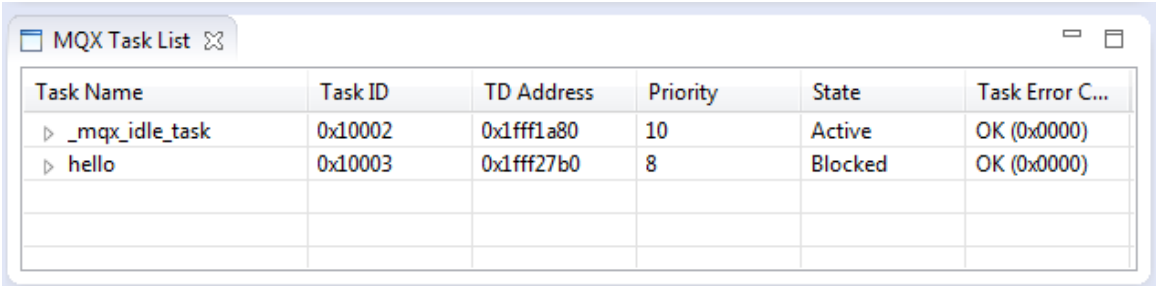


Figure 5 MQX RTOS menu

Resume (F8 or Run/Resume)  the debug session and then suspend  (Run/Suspend) it again to initialize MQX structures needed for the MQX TAD.

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

- Task List – overview about all tasks created in the MQX 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 6 Task list

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

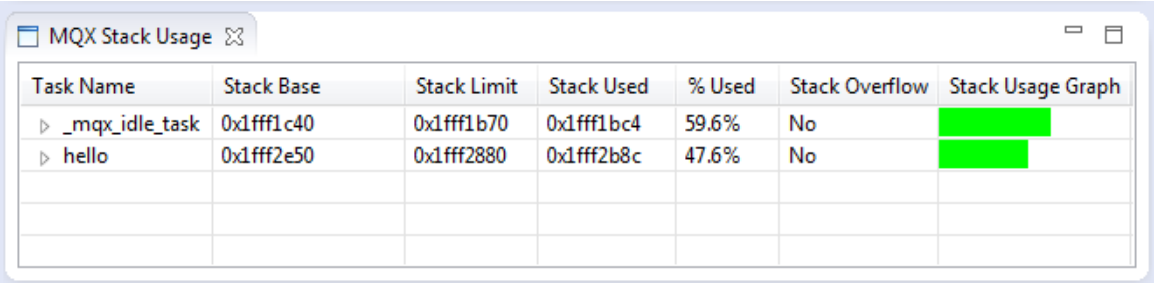


Figure 7 Stack usage

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

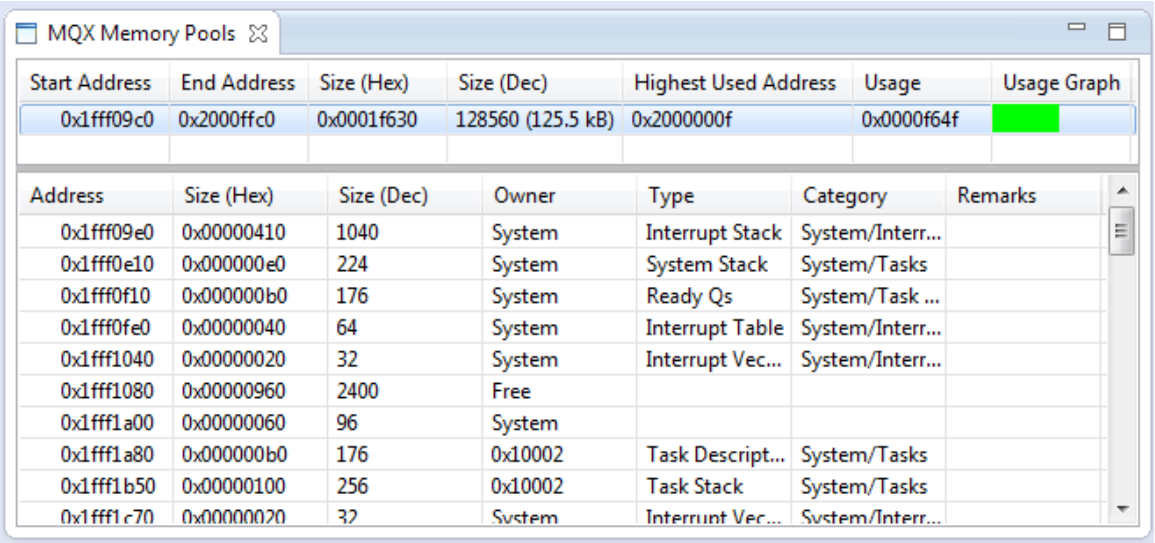


Figure 8 Memory pools

- Semaphores, Events, Mutexes (or Ligthweight Semaphores, Ligthweight Events) - displays address and status of synchronization objects created by the MQX 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.

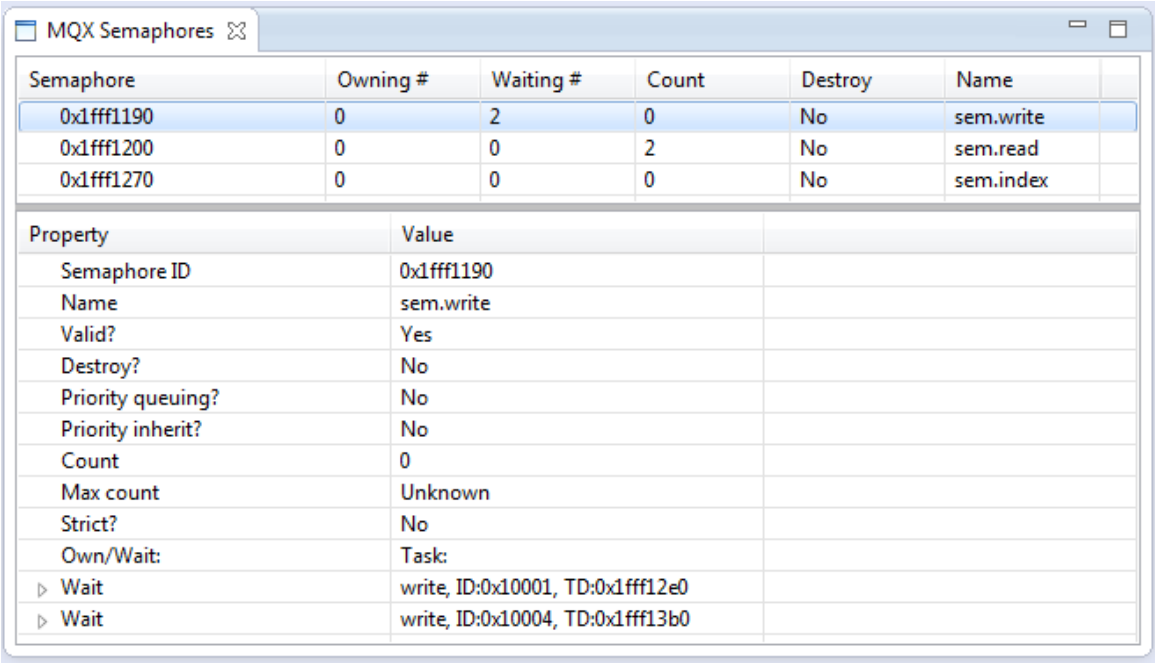


Figure 9 Semaphores

- I/O Devices – displays name, type and address of I/O Devices used by MQX application.

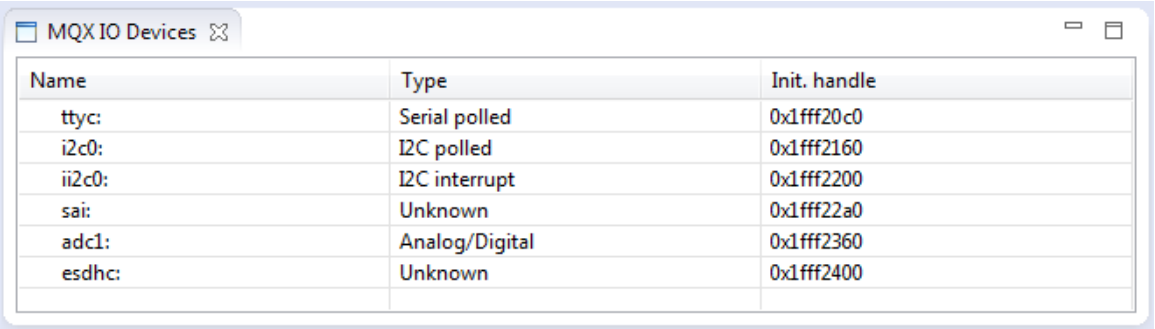


Figure 10 I/O devices