

## **Oracle® Java ME Embedded**

Getting Started Guide for the Reference Platform (Freescale  
FRDM-K64F)

Release 8.2 Developer Preview

**E55005-03**

September 2015

Beta Draft

Oracle Java ME Embedded Getting Started Guide for the Reference Platform (Freescale FRDM-K64F),  
Release 8.2 Developer Preview

E55005-03

Copyright © 2014, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Alla Redko

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

**This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.**

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	vii
Conventions .....	vii
<b>1 Installing Oracle Java ME Embedded Software</b>	
Required Software and Hardware Items .....	1-1
Updating the Firmware .....	1-2
Configuring Windows for Serial Communication with K64F Board .....	1-3
Configuring USBSerial Interface .....	1-4
Installing Oracle Java ME SDK 8.2 Platform .....	1-5
Installing Java ME Embedded 8.2 on Freescale FRDM-K64F Board .....	1-5
Installing Java ME Embedded 8.2 Software Using the Device Installation Wizard .....	1-5
Downloading and Installing Java ME Embedded 8.2 Manually .....	1-7
Setting Up a Network Configuration .....	1-8
Installing Certificates .....	1-8
<b>2 Managing Applications</b>	
Connecting to the K64F Board via a Serial Port .....	2-1
Adding the K64F Board to the Device Connection Manager .....	2-2
Running an Application with the Java ME SDK .....	2-4
Running an Application Using Command-Line Interface (CLI) .....	2-6
Setting Up the Date and Time .....	2-7
Dealing with Memory Constraints on the Freescale FRDM-K64F Board .....	2-8
<b>3 Troubleshooting</b>	
Connecting to the Freescale FRDM-K64F Board .....	3-1
Running Applications on the K64F-Board .....	3-1
<b>A Device I/O Preconfigured List</b>	
GPIO Pins .....	A-1
GPIO Ports .....	A-2
I2C .....	A-2

PWM .....	A-3
SPI .....	A-3
UART .....	A-4
DAC.....	A-4
Pulse Counter.....	A-4
ADC.....	A-5

## **B Configuring the Java Runtime Properties**

Modifying the jwc_properties.ini File .....	B-1
Using the CLI set-property Command .....	B-1

## List of Figures

1-1	Freescle FRDM-K64F Board .....	1-2
1-2	Windows Device Manager Displaying an mbed Serial Port .....	1-4
1-3	Windows Device Manager Displaying an Mbed Virtual Serial Port .....	1-5
1-4	Oracle Java ME SDK Device Manager Icon .....	1-6
1-5	Device Connections Manager .....	1-6
1-6	FRDM-K64F Board Is Registered.....	1-7
2-1	Serial Connection to the K64F Board .....	2-2
2-2	Adding IP Address of the K64F Board .....	2-3
2-3	Device Connections Manager with K64F Connected .....	2-3
2-4	Device Selector Window .....	2-4
2-5	Running Application .....	2-5
2-6	Output Console .....	2-5
2-7	Installing Application.....	2-6
2-8	Command-Line Interface.....	2-6

## List of Tables

3-1	Problems and Solutions: Connecting to the Freescale FRDM-K64F Board .....	3-1
3-2	Problems and Solutions: Running Applications on the Freescale FRDM-K64F Board ...	3-1

---

---

# Preface

This guide describes how to install Oracle Java ME Embedded software onto a Freescale FRDM-K64F device and how to install and run applications.

## Audience

This guide is for developers who want to run Oracle Java ME Embedded software on a Freescale FRDM-K64F device.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

- Oracle Java ME Embedded Release Notes for the Reference Platform (Freescale FRDM-K64F)

## Conventions

The following text conventions are used in this guide:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.





---

---

# Installing Oracle Java ME Embedded Software

This chapter describes how to prepare your Freescale FRDM-K64F board and install the Oracle Java ME Embedded 8.2 software on it.

This chapter contains the following sections:

- [Required Software and Hardware Items](#)
- [Updating the Firmware](#)
- [Configuring Windows for Serial Communication with K64F Board](#)
- [Configuring USBSerial Interface](#)
- [Installing Oracle Java ME SDK 8.2 Platform](#)
- [Installing Java ME Embedded 8.2 on Freescale FRDM-K64F Board](#)
- [Setting Up a Network Configuration](#)
- [Installing Certificates](#)

## Required Software and Hardware Items

The following hardware and software items are required for developing on the Freescale-K64F board:

- Freescale FRDM-K64F board
- Micro-USB power supply of 0.7 or greater amps, and 5 volts. Note that the power supply must have a *micro*-USB type B connector, not a regular USB or mini-USB connector.
- Micro-USB cable with a micro-USB type B connector (optional, used only for tooling over serial connection)
- Micro SD card of 4 GB or less formatted with FAT32

---

---

**Note:** You must use only SD cards. Work with SDHC cards is not guaranteed.

---

---

- Ethernet cable with an RJ-45 connection, as well as a connection to a network with a DHCP server (optional)
- Oracle Java ME Embedded 8.2 software distribution
- A terminal emulator program, such as PuTTY.

The executable file `putty.exe` is directly downloadable from

<http://www.putty.org/>

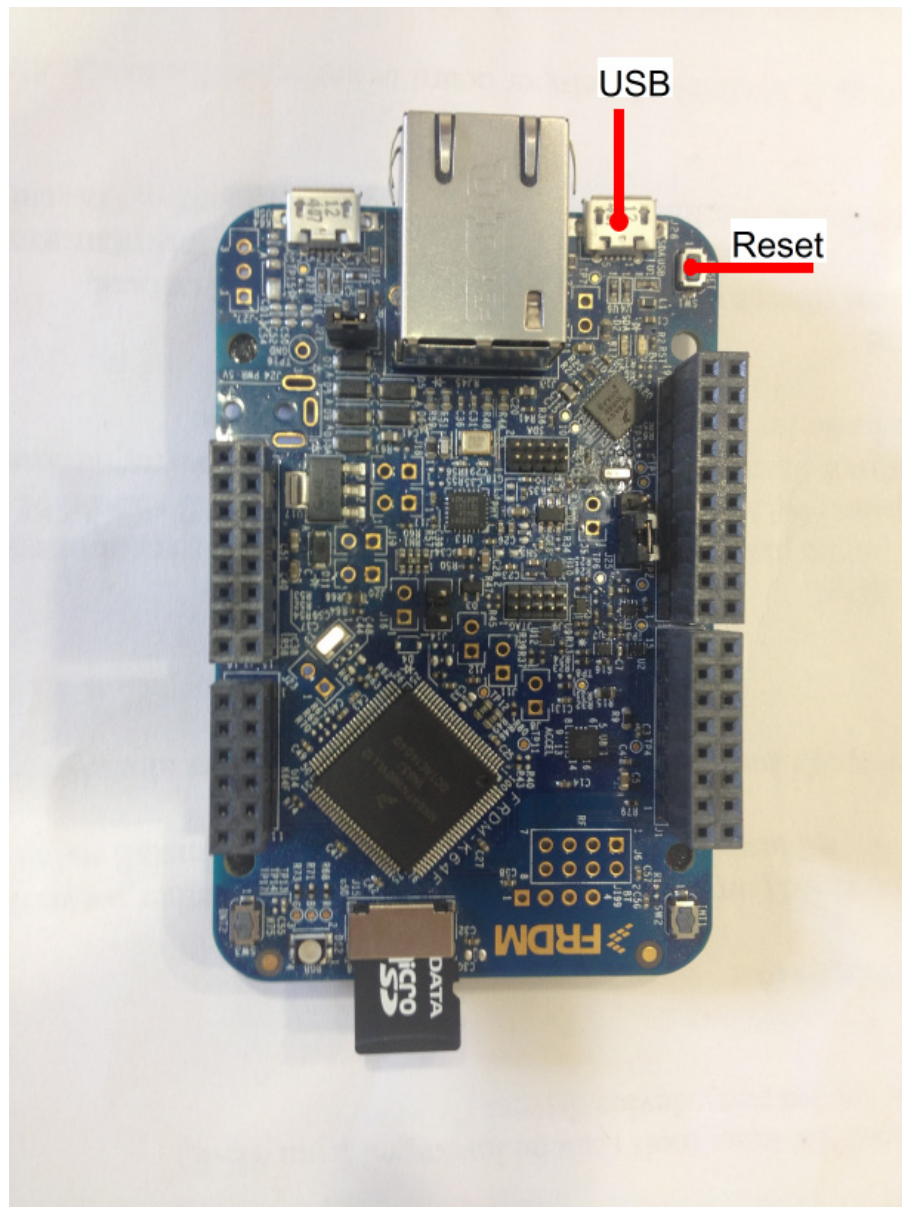
## Updating the Firmware

Ensure that you have the latest firmware on your K64F board.

1. Connect the K64F board to your host computer with a micro-USB cable using the slot marked USB as shown in [Figure 1-1](#). The flashing status light indicates the board has power.

In a few seconds your computer recognizes the mbed microcontroller and displays it as a standard USB drive.

**Figure 1-1** Freescale FRDM-K64F Board



2. Open the `details.txt` file on your MBED device in a text editor to learn the firmware version.
3. Open the `mbed.htm` file in a browser to log in at [mbed.org](http://mbed.org).  
If you do not have an mbed user account, click **Signup** and create the account. Otherwise, login in with your username and password.
4. Find the latest firmware version and instructions explaining how to upgrade the firmware at  
<http://mbed.org/handbook/Firmware-FRDM-K64F>

---

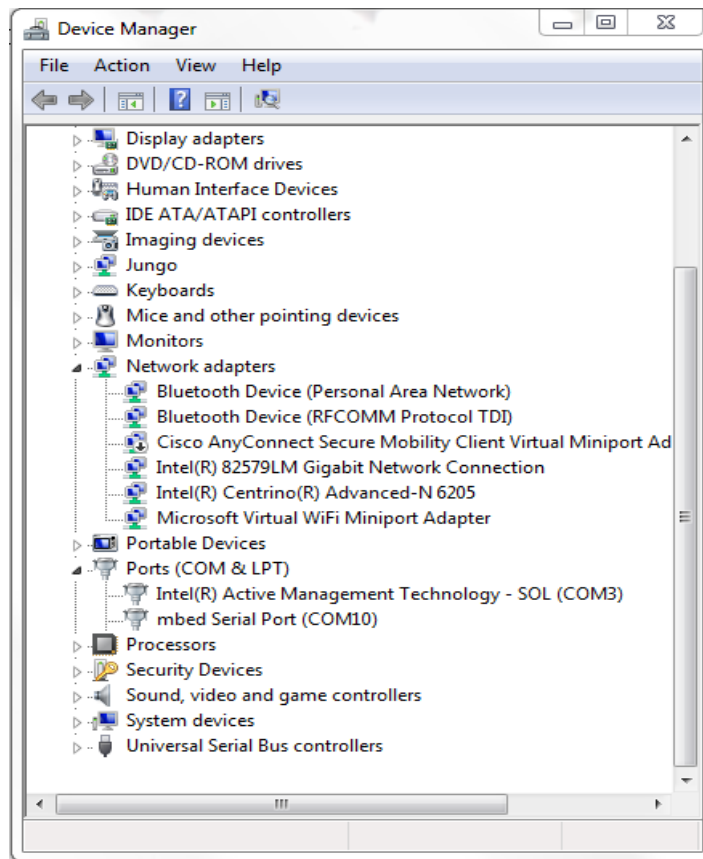
**Note:** Jumper J25 is responsible for the configuration of the Reset button. For a successful firmware update, it is important that jumper J25 connects SDA and SW1 pins. Turn over the board to ensure that the correct pins are connected.

---

## Configuring Windows for Serial Communication with K64F Board

To enable serial communication with the K64F board on Windows, perform the following steps:

1. Download and install the mbed Windows serial port driver from  
<http://mbed.org/handbook/Windows-serial-configuration>.
2. Open the Windows Device Manager by clicking **Start**, then **Control Panel** and **Device Manager** and expand the Ports (COM & LPT) line as shown in [Figure 1-2](#) to learn the identity of the serial port.

**Figure 1–2 Windows Device Manager Displaying an mbed Serial Port**

## Configuring USBSerial Interface

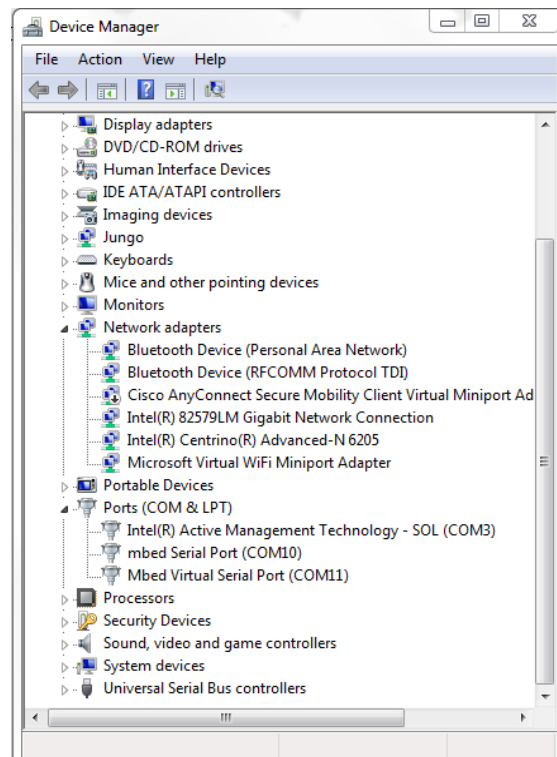
The USBSerial interface is used to emulate a serial port over USB. If you want to communicate to your K64F board using this serial port or install the Java ME Embedded software on your board using a feature of the Oracle Java ME SDK called Device Installation Wizard, you need to configure the USBSerial interface.

On Windows, perform the following steps.

1. Download an archive from <http://mbed.org/handbook/USBSerial> and extract it on your host computer.
2. Open the `serial.inf` file and replace the `[DeviceList]` and `[DeviceList.NTamd64]` sections with the following lines:
 

```
[DeviceList]
%DESCRIPTION%=DriverInstall, USB\VID_0472&PID_145A
[DeviceList.NTamd64]
%DESCRIPTION%=DriverInstall, USB\VID_0472&PID_145A
```
3. Connect the K64F board to your host computer with the second USB cable using the slot marked K64F USB.
4. Install the driver using the Windows Device Manager as described at <http://mbed.org/handbook/USBSerial>.

After you successfully installed the driver, the Windows Device Manager displays the Mbed Virtual Serial Port (COM ID) as shown in [Figure 1–3](#).

**Figure 1–3 Windows Device Manager Displaying an Mbed Virtual Serial Port**

## Installing Oracle Java ME SDK 8.2 Platform

To install Oracle Java ME SDK 8.2 platform, see *Oracle Java ME SDK Developer's Guide* for complete instructions.

## Installing Java ME Embedded 8.2 on Freescale FRDM-K64F Board

You have two options to install the Java ME Embedded software on the K64Fboard: using the Device Installation Wizard, a feature of the Oracle Java ME SDK, or manually downloading the binary file and installing it. These options are described in the following sections.

- [Installing Java ME Embedded 8.2 Software Using the Device Installation Wizard](#)
- [Downloading and Installing Java ME Embedded 8.2 Manually](#)

### Installing Java ME Embedded 8.2 Software Using the Device Installation Wizard

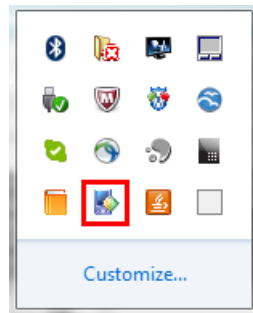
This section describes how to install the Java ME Embedded software using the Device Installation Wizard.

Ensure that an SD card is inserted into the K64F board, the board is connected to your host computer with two USB cables, and also connected to a network socket with an Ethernet cable.

To install Java ME Embedded software on your K64F board, perform the following steps.

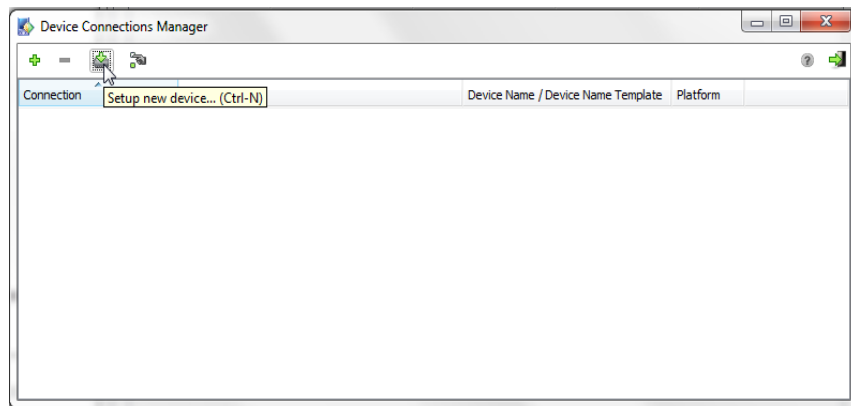
1. Right-click the Oracle Java ME SDK Device Manager icon on the notification area as shown in [Figure 1–4](#) and choose **Manage Device Connections**.

**Figure 1–4 Oracle Java ME SDK Device Manager Icon**



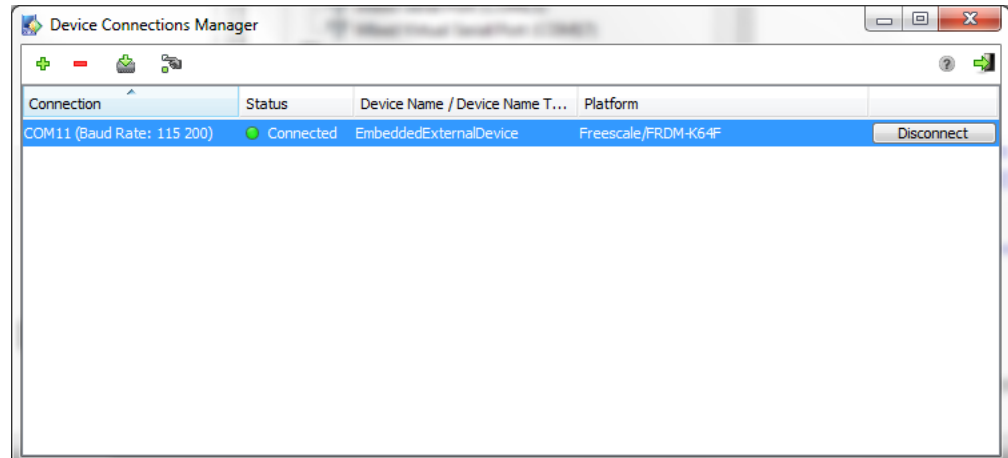
If there is no Device Manager icon, start the Java ME SDK Device Connections Manager located at `<SDK Installation Folder>/bin/device-manager.exe`

**Figure 1–5 Device Connections Manager**



2. Click **Setup new device** icon. The Flash Devices window opens up.
3. Select an mbed device and click **Flash**.
4. Wait while the Java ME Embedded runtime image is being installed or updated.
5. Press the Reset button on the board.

The FRDM-K64F board is now registered. It has the Connected status and is ready for work as shown in [Figure 1–6](#).

**Figure 1–6 FRDM-K64F Board Is Registered**

Note that the Java ME Embedded runtime image is written into the on-chip flash and does not appear on the SD card's file system.

There is no need to copy the `jwc_properties.ini` file onto the SD card unless you want to modify some of the default values. The version of the file with the default property values is already written to the Java ME Embedded runtime image.

## Downloading and Installing Java ME Embedded 8.2 Manually

If you want to install the Java ME Embedded software on your K64F board manually, perform the following steps:

1. Download the binary zip file called `oracle-jmee-8-2-dp2-fs-k64-mbed-bin.zip` from <http://www.oracle.com/technetwork/java/embedded/javame/embed-me/downloads/index.html>
2. Extract the contents of the zip in a directory on your computer.
3. Copy the `flash\JavaVM.bin` file from the extracted directory onto your MBED drive.

---

**Note:** There is no need to copy the `jwc_properties.ini` file on the SD card unless you want to modify some default property values such as network setup. The next three steps are optional.

---

4. Mount the SD card to your desktop computer and copy the modified `sd_card\java\jwc_properties.ini` file from the extracted directory to the `java` directory on the SD card.
5. Eject or unmount the SD card from the desktop computer.
6. After the green light on the board stops blinking, insert the SD card into the K64F board.
7. Press the Reset button.

## Setting Up a Network Configuration

By default the K64F board obtains an IP address from a DHCP server. However, you can setup using a static IP address by modifying the `jwc_properties.ini` file as follows:

```
ip_method=static
ip_addr=XXX.XXX.XXX.XXX
ip_netmask=XXX.XXX.XXX.XXX
ip_gateway=XXX.XXX.XXX.XXX
```

To switch to using the DHCP server, remove the `ip_method` parameter or set

```
ip_method=dhcp
```

## Installing Certificates

By default, FRDM-K64F does not contains certificates. To make SSL/HTTPS connections from the K64F board, import certificates using either of the following procedures.

- Create an empty `/java/_main.ks` file on the SD card and use the `mekeytool.exe` tool to import appropriate certificates into the K64F board.
- Create the `_main.ks` file by using the `MEKeyTool.jar` with `-version restricted` option and copy it in the `java` directory on the SD card.

For more information about using the `mekeytool.exe` tool, see *Manage Certificates (mekeytool)* in the *Oracle Java ME SDK Developer's Guide* at <https://docs.oracle.com/javame>.



---

---

## Managing Applications

This chapter contains information about using Java ME SDK 8.2 to install and run applications.

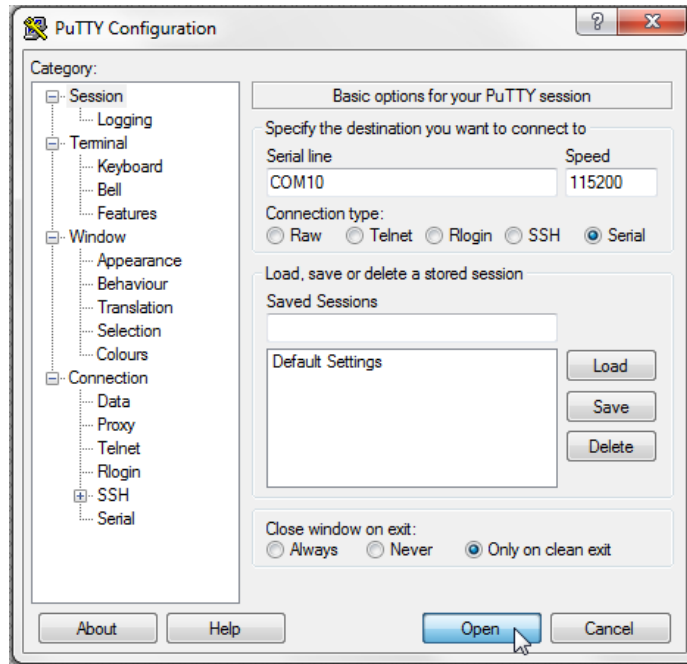
This chapter contains the following sections:

- [Connecting to the K64F Board via a Serial Port](#)
- [Adding the K64F Board to the Device Connection Manager](#)
- [Running an Application with the Java ME SDK](#)
- [Running an Application Using Command-Line Interface \(CLI\)](#)
- [Setting Up the Date and Time](#)
- [Dealing with Memory Constraints on the Freescale FRDM-K64F Board](#)

### Connecting to the K64F Board via a Serial Port

By connecting to the K64F board, you get access to the device log. To connect to the board via a serial port, perform the following steps.

1. Plug in a micro-USB type B connector to the K64F board and another end of the USB cable to your host computer.
2. Plug one end of a network cable into a switch, router, or network wall socket, and the other end to the K64F board.
3. Start a PuTTY executable file on your desktop computer.
4. Select **Serial** as a connection type.
5. Specify the COM port that corresponds to the K64F board, which is displayed in the Windows Device Manager as mbed SerialPort(COMxx), and the default baud rate 115200, as shown in [Figure 2-1](#).

**Figure 2–1 Serial Connection to the K64F Board**

6. Click **Terminal** category and select the **Implicit CR in every LF** option.

---

**Note:** It is recommended that you save the session and load it in future connections.

---

7. Click **Open**.
8. Press Reset button on the board.
9. After the connection is established, the PuTTY terminal windows displays the IP address of the K64F board and other logging messages.

## Adding the K64F Board to the Device Connection Manager

If you manually installed the Oracle Java ME Embedded software on the K64F board, then you need to add the board to the Device Connection Manager as described in this section.

Ensure that the K64F board is connected to your host computer and to a network socket.

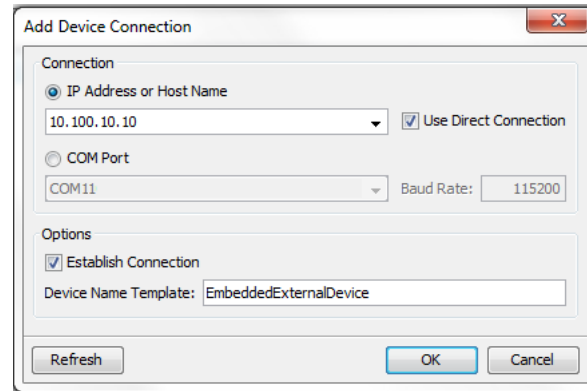
1. Right-click the Oracle Java ME SDK Device Manager icon on the notification area and choose **Manage Device Connections**.

If there is no Device Manager icon, start the Java ME SDK Device Connections Manager located at `<SDK Installation Folder>/bin/device-manager.exe`

2. Click **Add** and select the type of connection.
  - Select the **IP Address** or **Host Name** option if you want to establish a network connection to the board. Then type the IP address of the K64F board reported earlier in the PuTTY terminal window and click **OK** as shown in [Figure 2–2](#).

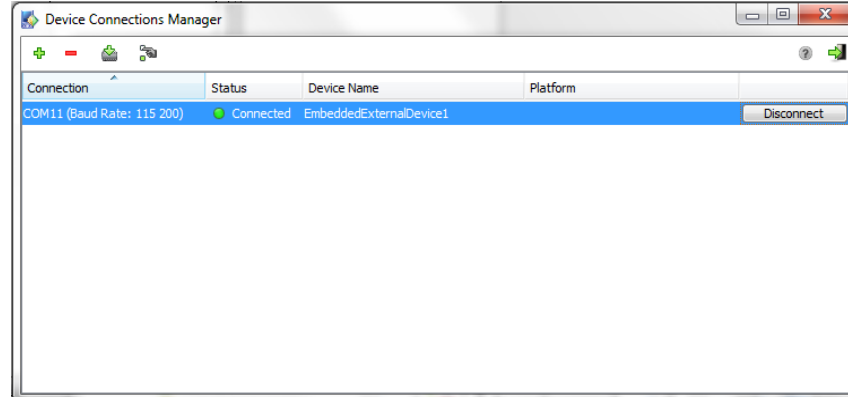
- Select the **COM Port** option, if you want to establish a serial connection. Then choose the COM port number in the drop-down list and click **OK**. The corresponding COM port is called “Mbed Virtual Serial Port” in a Windows Device Manager.

**Figure 2–2 Adding IP Address of the K64F Board**



3. After the K64F board is registered, the Device Connections Manager displays its IP address in the Connections column and its status is Connected as shown in [Figure 2–3](#).

**Figure 2–3 Device Connections Manager with K64F Connected**



Note that the host-side firewall, if not configured properly, can cause connection issues. In this case the Status column displays Connecting. Edit your firewall configuration settings to open TCP port 2201.

The registered device obtains a name of the form EmbeddedExternalDevice1. To learn the name of the device, right-click the Java ME Device Manager icon on the notification area and click Registered Devices.

If you want to change the type of connection to the K64F board, do the following.

1. In the Device Connections Manager window, remove the K64F board.
2. Right-click the Device Connections Manager icon on the notification area and choose **Exit**.

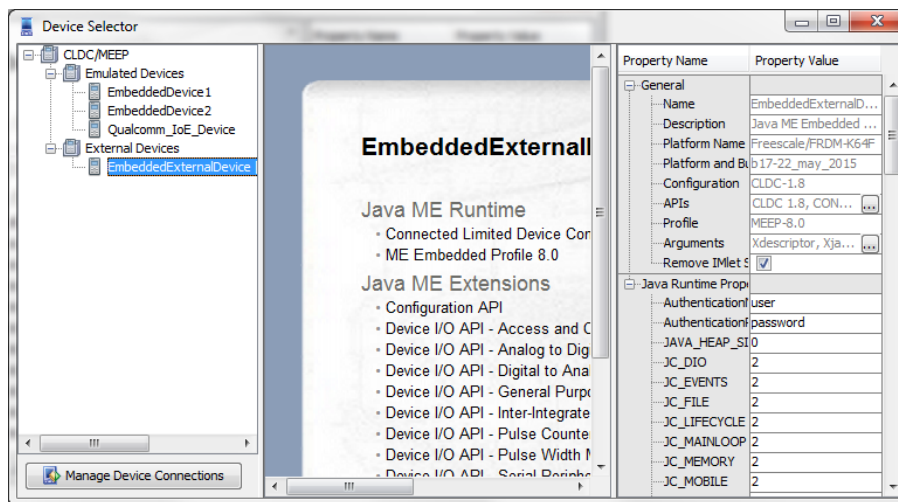
3. Start the Device Connection Manager again and add the K64F board using another type of connection.

## Running an Application with the Java ME SDK

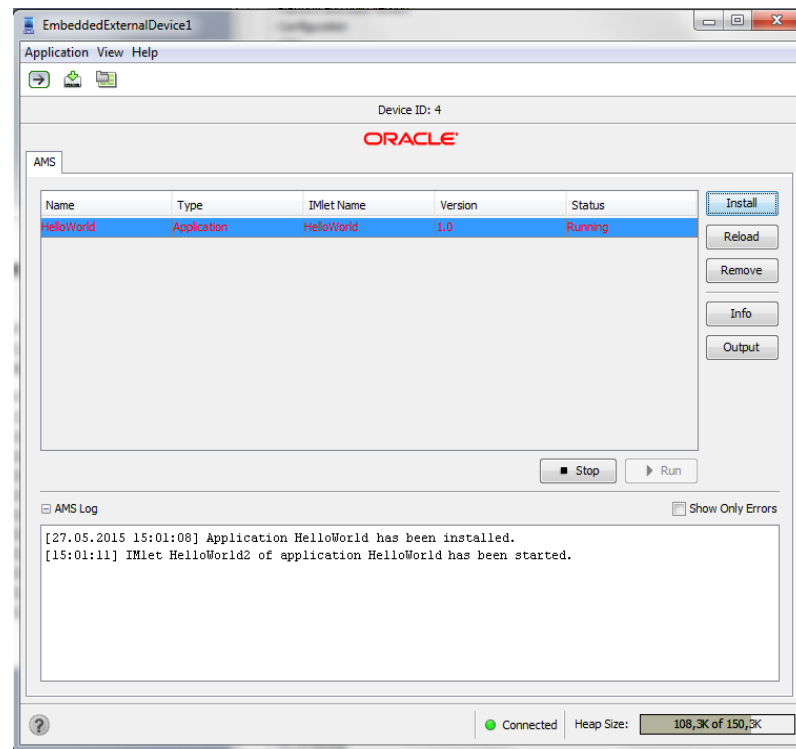
You can use the Java ME Platform SDK 8.2 to install and run applications. Note that you can run only one application at a time.

1. In the Windows Start menu, find the Java ME Platform SDK 8.2 and launch the Device Selector utility.
2. Because you have added the device to the Device Connections Manager, the Device Selector displays it among the available external devices. EmbeddedExternalDevice1 [Freescale/FRDM-K64F, <IP Address>] in [Figure 2-4](#) corresponds to the Freescale FRDM-K64F board.

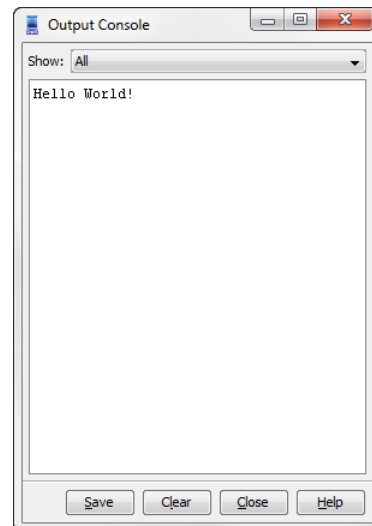
**Figure 2-4** Device Selector Window



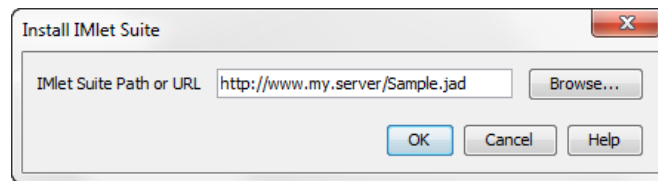
3. Right click on **EmbeddedExternalDevice1 [Freescale/FRDM-K64F]** and select **Run JAR or JAD** command.
4. Select the target .JAR or .JAD file and wait until the Device Selector installs the application.
5. If an IMlet is installed successfully, the Device Selector runs it automatically. [Figure 2-5](#) shows the HelloWorld application running. You can use **Stop** and **Run** buttons to manage the application.

**Figure 2–5** Running Application

6. Click **Output** to view the results of the IMlet execution.

**Figure 2–6** Output Console

At any time you can install more IMlets by clicking **Install** button or the corresponding icon. You can specify any location in the file system of the host computer or any URL. The latter option is shown in [Figure 2–7](#).

**Figure 2–7 Installing Application**

## Running an Application Using Command-Line Interface (CLI)

You can use the Command-Line Interface (CLI) to install, run, stop, and remove your applications. Before you proceed with procedures below, study the CLI introduction at *Oracle Java Micro Edition Software Development Kit Developer's Guide* (see at <http://docs.oracle.com/javame/>).

The following procedure explains how to install and run an IMlet by using the CLI.

1. Connect to the K64F board as described in [Connecting to the K64F Board via a Serial Port](#).
2. Change to the directory with the extracted binary ZIP on the host desktop and then change to the util directory.
3. Start the Developer Agent program by typing the following command:

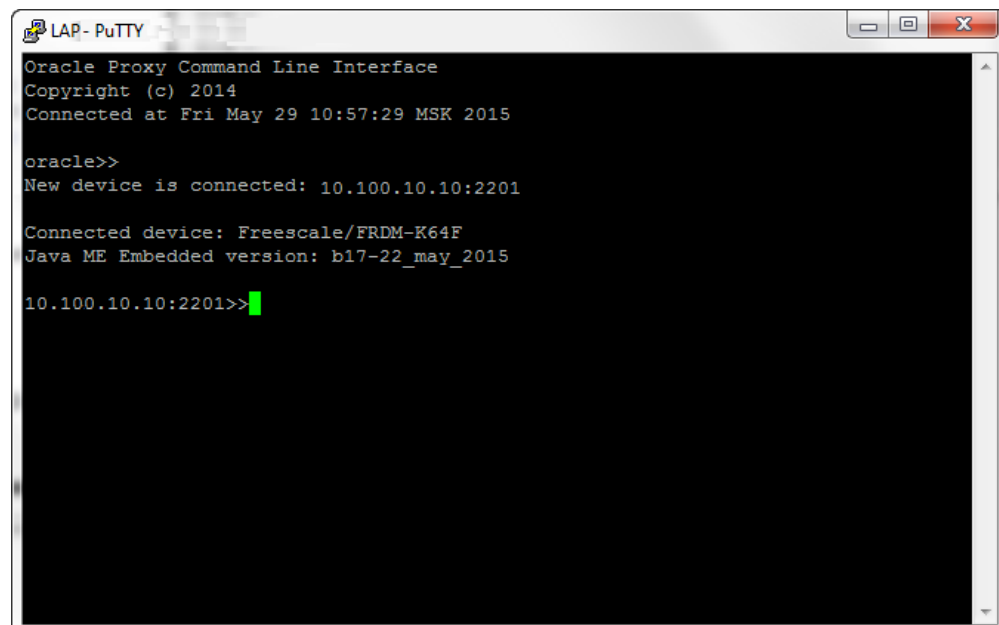
```
C:\mydir\util java -jar proxy.jar -socket <K64F IP Address>
```

You should see an output similar to the following:

```
Trying to open socket connection with device: <IP Address>:2201
```

```
Connected to the socket Socket[addr=/<IP address>, port 2201, localport=54784]
```

4. Connect to the CLI, as described in the *Java ME Embedded Developer's Guide*. After you successfully connected to the CLI, you should see the following output in the PuTTY terminal window.

**Figure 2–8 Command-Line Interface**

5. Type the following command:

```
ams-install file:///C:/<some directory>/HelloWorld.jad hostdownload
```

Note that *<some directory>* must be an actual path to your HelloWorld.jad file. You should see the following feedback as the application is installed:

```
<<ams-install,start install,file:///c:<some directory>/HelloWorld.jad
<<ams-install,install status: stage DOWNLOADING_DESCRIPTOR, 0%
<<ams-install,install status: stage VERIFYING, 100%
<<ams-install,install status: stage DONE, 100%
<<ams-install,OK,install success
```

You can also install an IMlet from a remote server by using the following command:

```
<<ams-install http://<some URL>/HelloWorld.jad
```

6. Type `ams-list` at the prompt to check if the HelloWorld IMlet is listed. In the following example, the suite id of HelloWorld is 0.

```
ams-list
<<ams-list,0.HelloWorld|Vendor,STOPPED
<<ams-list,OK,1 suites are installed
```

7. Start the HelloWorld application with the following command:

```
ams-run 0
<<ams-run,OK,started
```

8. Stop the HelloWorld application with the following command:

```
ams-stop 0
<<ams-stop,OK,stopped
```

See the Using the Command-Line Interface section of the *Java ME Embedded Developer's Guide* for more information about the available CLI commands.

## Setting Up the Date and Time

To run the signed IMlets on the K64F board, ensure that the system date and time are set on the board correctly.

The date and time are set by the `rtc.time` property in the `jwc_properties.ini` file. The default value is 2014:01:01:00:00:00.

To change the time, edit the `jwc_properties.ini` file on the SD card or use the `set-property` command as follows.

1. Connect to the CLI and type the following command:

```
set-property rtc.time YYYY:MM:DD:HH:MM:SS
save-properties
```

2. Press the Reset button to restart Java on the board.

Whenever you encounter that the time changed to its default value, follow the described procedure to restore the actual time.

## Dealing with Memory Constraints on the Freescale FRDM-K64F Board

The Freescale FRDM-K64F board is a memory constrained device which implies that developers should take care to reduce memory usage of applications. The K64F board has approximately 60 Kb RAM for applications.

To reduce the memory usage of application, apply the following guidelines.

- Turn off the debugging information during compilation
- Use obfuscation to remove unnecessary classes and reduce the size of symbol information
- Limit the number and size of application classes and objects
- Consider removing or reducing non-essential functionality in the application code



## Troubleshooting

This chapter contains a list of common problems that you may encounter while installing and running the Oracle Java ME embedded software on the Freescale FRDM-K64F board. This chapter provides information on the causes of these problems and possible solutions for them.

### Connecting to the Freescale FRDM-K64F Board

[Table 3–1](#) contains information about problems and solutions when connecting to the K64F board.

**Table 3–1 Problems and Solutions: Connecting to the Freescale FRDM-K64F Board**

Problem	Cause	Solution
When connecting to the board using the PuTTY terminal emulator program, the message "No disk, or could not put SD card into SPI idle state. Didn't get a response from the disk. Set 512-byte block timed out" is displayed.	The SD card is absent.	Insert the SD card and press the Reset button.
	The SD card is not formatted.	Format the SD card.
	The SD card is not compatible.	Replace the SD card.
	The platform failed to initialize the SD card.	Press the Reset button or use the CLI and perform the <code>shutdown -r</code> command.
The Device Connections Manager displays the status of the board as Connecting.	The host-side firewall might not be configured properly.	Edit the firewall configuration settings to free TCP port 2201.

### Running Applications on the K64F-Board

[Table 3–2](#) contains information about problems and solutions when running applications on the K64F board.

**Table 3–2 Problems and Solutions: Running Applications on the Freescale FRDM-K64F Board**

Problem	Cause	Solution
The application terminates due to <code>OutOfMemoryError</code> .	The application requires more memory than actually available.	Review <a href="#">Dealing with Memory Constraints on the Freescale FRDM-K64F Board</a> and follow the suggestions as applicable.
The application does not work because "the certificate is not yet valid."	The date and time setting on the K64F board is incorrect.	Review <a href="#">Setting Up the Date and Time</a> and edit the date and time setting as described.



---



---

## Device I/O Preconfigured List

This appendix describes the proper ID and names for the various peripheral ports for the Freescale FRDM-K64F board, which are accessible using the Device I/O APIs.

The tables use the following legend:

- **Device ID:** an integer identifier that can be used to open the device with the methods of the `DeviceManager` class.
- **Device Name:** the string name of a device that can be used to open it by name with the methods of the `DeviceManager` class.
- **Mapped:** all hardware-related information regarding a peripheral, such as physical location, mapping, or port. This information enables the user to determine the peripheral's location on a target board.
- **Configuration:** properties that are passed to the specific `DeviceConfig` constructor to open the peripheral by ID or name. The configuration can be used to open the peripheral using the `DeviceManager` with the appropriate configuration.

### GPIO Pins

The following GPIO pins are preconfigured.

Device ID	Device Name	Mapped	Configuration
1	LED1	LED1	<pre> controllerNumber = 1 pinNumber = 22 direction = GPIOPinConfig.DIR_ OUTPUT_ONLY mode = GPIOPinConfig.MODE_OUTPUT_ PUSH_PULL trigger = GPIOPinConfig.TRIGGER_NONE initValue = false </pre>
2	LED2	LED2	<pre> controllerNumber = 4 pinNumber = 26 direction = GPIOPinConfig.DIR_ OUTPUT_ONLY mode = GPIOPinConfig.MODE_OUTPUT_ PUSH_PULL trigger = GPIOPinConfig.TRIGGER_NONE initValue = false </pre>

Device ID	Device Name	Mapped	Configuration
3	LED3	LED3	<pre> controllerNumber = 1 pinNumber = 21 direction = GPIOPinConfig.DIR_OUTPUT_ONLY mode = GPIOPinConfig.MODE_OUTPUT_PUSH_PULL trigger = GPIOPinConfig.TRIGGER_NONE initValue = false </pre>
4	SW2	SW2	<pre> controllerNumber = 2 pinNumber = 6 direction = GPIOPinConfig.DIR_INPUT_ONLY mode = GPIOPinConfig.MODE_INPUT_PULL_DOWN trigger = GPIOPinConfig.TRIGGER_RISING_EDGE initValue = ignored </pre>
5	SW3	SW3	<pre> controllerNumber = 0 pinNumber = 4 direction = GPIOPinConfig.DIR_INPUT_ONLY mode = GPIOPinConfig.MODE_INPUT_PULL_UP trigger = GPIOPinConfig.TRIGGER_FALLING_EDGE initValue = ignored </pre>

## GPIO Ports

The following GPIO ports are preconfigured.

Device ID	Device Name	Mapped	Configuration
6	LEDS	LED1, LED2, LED3	<pre> direction = GPIOPinConfig.DIR_OUTPUT_ONLY initValue = 0 </pre> <p>For a particular pin, see GPIO pin details for IDs 1, 2, and 3.</p>

## I2C

The following I2C devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
200	FXOS8700CQ	FXOS8700CQ: Digital Sensor - 3D Accelerometer + 3D Magnetometer	controllerNumber = 0 address = 29 addressSize = UNASSIGNED clockFrequency = 100000

## PWM

The following PWM devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
600	PWM1.1	PTA0	controllerNumber = UNASSIGNED channelNumber = UNASSIGNED pulsePeriod = UNASSIGNED pulseAlignment = PWMChannelConfig.ALIGN_LEFT idleState = PWMChannelConfig.IDLE_STATE_LOW outputGpioPinConfig.controllerNumber = 0 outputGpioPinConfig.pinNumber = 0
601	PWM2.1	PTC4	controllerNumber = UNASSIGNED channelNumber = UNASSIGNED pulsePeriod = UNASSIGNED pulseAlignment = PWMChannelConfig.ALIGN_LEFT idleState = PWMChannelConfig.IDLE_STATE_LOW outputGpioPinConfig.controllerNumber = 2 outputGpioPinConfig.pinNumber = 4

## SPI

The following SPI devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
300	SPI0.0	SPI0	controllerNumber = 0 address = 0 wordLength = 8 clockFrequency = 1000000 clockMode = CS_ACTIVE_HIGH bitOrdering = Device.BIG_ENDIAN

The clockMode parameter defines a clock polarity and phase as follows:

- 0 = CPOL\_Low | CPHA\_1Edge;

- 1 = CPOL\_Low | CPHA\_2Edge;
- 2 = CPOL\_High | CPHA\_1Edge;
- 3 = CPOL\_High | CPHA\_2Edge

The `bitOrdering` parameter defines the bit ordering of the slave device. The possible values are 0 - `Device.LITTLE_ENDIAN`, 1 - `Device.BIG_ENDIAN`.

## UART

The following UART devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
100	UART3	UART3	<pre>controllerName = UART3 baudRate = 9600 dataBits = DATABITS_8 parity = PARITY_NONE stopBits = STOPBITS_1 flowcontrol = FLOWCONTROL_NONE</pre>

## DAC

The following Digital-to-Analog (DAC) devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
500	DAC0.0	DAC0_OUT	<pre>converterNumber = UNASSIGNED channelNumber = 0 resolution = UNASSIGNED samplingInterval = UNASSIGNED</pre>

## Pulse Counter

The following Pulse Counter devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
700	PULSE 0.0	PTA1	<pre>controllerNumber = UNASSIGNED type = TYPE_FALLING_EDGE_ONLY channelNumber = UNASSIGNED outputGpioPinConfig.controllerNumber = 0 outputGpioPinConfig.pinNumber = 1</pre>
701	PULSE1.0	PTA4	<pre>controllerNumber = UNASSIGNED channelNumber = UNASSIGNED type = TYPE_FALLING_EDGE_ONLY outputGpioPinConfig.controllerNumber = 0 outputGpioPinConfig.pinNumber = 4</pre>

## ADC

The following Analog-to-Digital (ADC) devices are preconfigured.

Device ID	Device Name	Mapped	Configuration
400	ADC0_PTB2	ADC0 Pin PTB2	channelNumber = 0 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED
401	ADC1_PTB3	ADC1 Pin PTB3	channelNumber = 1 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED
402	ADC2_PTB10	ADC2 Pin PTB10	channelNumber = 2 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED
403	ADC3_PTB11	ADC3 Pin PTB11	channelNumber = 3 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED
404	ADC4_PTC11	ADC4 Pin PTC11	channelNumber = 4 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED
405	ADC5_PTC10	ADC5 Pin PTC10	channelNumber = 5 controllerNumber = UNASSIGNED resolution = UNASSIGNED samplingInterval = UNASSIGNED samplingTime = UNASSIGNED





---

## Configuring the Java Runtime Properties

This appendix contains information about changing Java runtime properties.

There are several ways to change the value of a property that affects Java's configuration or behavior at runtime:

- [Modifying the `jwc\_properties.ini` File](#)
- [Using the CLI `set-property` Command](#)

### Modifying the `jwc_properties.ini` File

The `jwc_properties.ini` file contains all the properties that affect Java configuration and behavior at runtime. In order to edit this file, do the following:

1. Eject the SD card from the K64F board.
2. Mount the SD card to your desktop computer and open the `jwc_properties.ini` file.
3. Find the property that should be changed, modify its value, and save the file.
4. Eject or unmount the SD card from the desktop computer.
5. Insert the SD card into the K64F board and press the Reset button.

### Using the CLI `set-property` Command

To modify a property using the `set-property` command in the command-line interface (CLI), do the following.

1. Connect to the board using command-line interface (CLI).
2. Execute the `set-property <property_name> <desired_property_value>` command.
3. Execute the `save-properties` command and restart Java on the board or execute the `shutdown - r` command.

