MicroEJ Platform Reference Implementation

Developer's Guide

MurataType1LD 1.0.1

Reference:	TLT-0839-DGI-PlatformReferenceImplementation-MurataType1LD
Version:	1.0.1
Revision:	1.0.1

Confidentiality & Intellectual Property

All rights reserved. Information, technical data and tutorials contained in this document are confidential and proprietary under copyright Law of Industrial Smart Software Technology (IS2T S.A.) operating under the brand name MicroEJ®. Without written permission from IS2T S.A., *copying or sending parts of the document or the entire document by any means to third parties is not permitted*. Granted authorizations for using parts of the document or the entire document do not mean IS2T S.A. gives public full access rights.

The information contained herein is not warranted to be error-free. IS2T® and MicroEJ® and all relative logos are trademarks or registered trademarks of IS2T S.A. in France and other Countries.

Java™ is Sun Microsystems' trademark for a technology for developing application software and deploying it in cross-platform, networked environments. When it is used in this documentation without adding the ™ symbol, it includes implementations of the technology by companies other than Sun.

Java™,all Java-based marks and all related logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States and other Countries.

Other trademarks are proprietary of their authors.

Revision History		
Revision 1.0.1	October 26th 2018	
Fix issue on documentation screenshots.		
Revision 1.0.0	October 1th 2018	
First release. Integrate MEJ32 Core, Net, SSL, ECOM-WIFI and ECOM-NETWORK features.		

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Intended Audience

The intended audience for this document are developers who wish to develop their first MicroEJ plaform with MicroEJ SDK and deploy a MicroEJ standalone application onto. Notes:

- This document is for the Murata Murata Type1LD EVB board.
- This document is not a user guide for the C development environment used for the final application link. Please consult the supplier of the C development environment for more information.
- Please visit the website https://developer.microej.com for more information about Murata Type1LD EVB products (platforms, videos, examples, application notes, etc.).

1.2. Scope

This document describes, step by step, how to start your development with MicroEJ SDK

- Create a MicroEJ platform for Murata Type1LD EVB board.
- Run a MicroEJ standalone application on the MicroEJ simulator.
- Run a MicroEJ standalone application on the MicroEJ platform and deploy it on the Murata Type1LD EVB board.

1.3. Prerequisites

- PC with Windows 7 or later.
- The MicroEJ SDK environment must be installed.
- Murata Type1LD EVB board.
- A GNU-GCC-based C development environment. The examples are packaged ready to run using the MicroEJ SDK IDE (including CDT packaging), which this document assumes has been successfully installed. Please visit the website mentioned above to obtain a version of the MicroEJ SDK IDE. Note, however, that developers are free to use a different CDT packaging.

Chapter 2. Create and Use Your First MicroEJ Platform

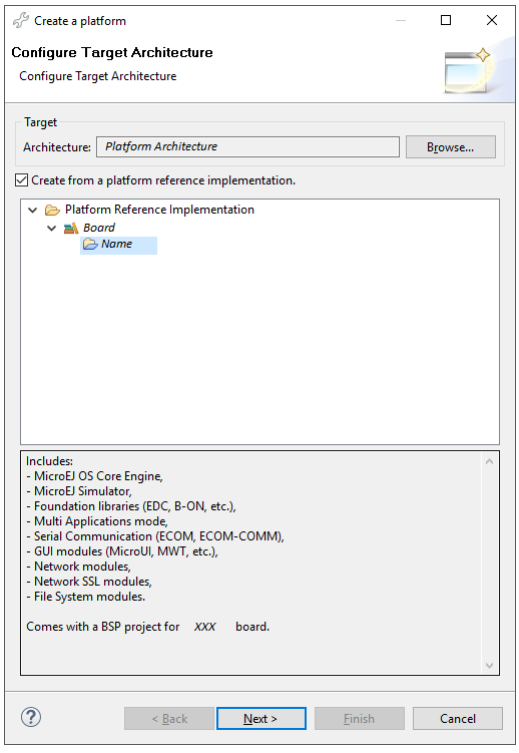
2.1. Create a MicroEJ Platform

The aim of this chapter is to create a MicroEJ platform from a MicroEJ architecture. The platform will then be used to run a MicroEJ standalone application in subsequent chapters.

Although it is possible to use MicroEJ SDK to create every aspect of a MicroEJ platform in accordance with specific requirements, in this chapter we will use a pre-packaged example of a MicroEJ platform that is already configured for the Murata Type1LD EVB.

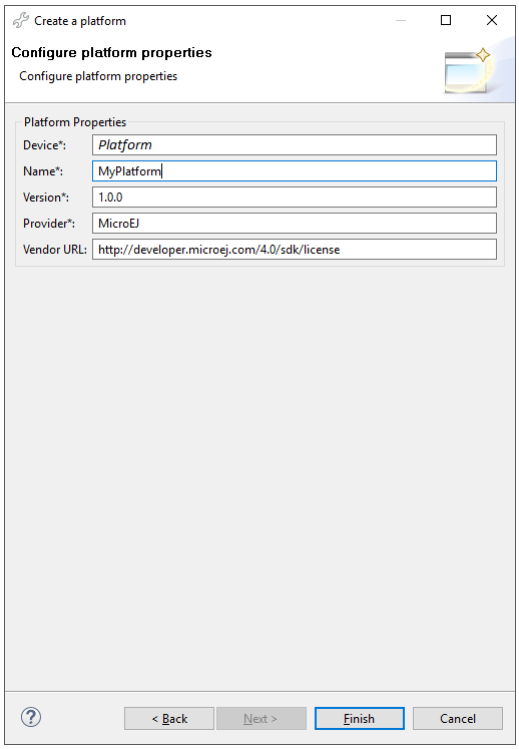
- Open MicroEJ SDK.
- Open the MicroEJ platform wizard: `File > New > MicroEJ Platform Project`.
- Select the MicroEJ architecture ARM Cortex-M3 GCC from the combo box. A MicroEJ Platform Reference Implementation is available:

Figure 2.1. MicroEJ Platform Reference Implementation Selection



- Select the MicroEJ platform SingleApp for the Murata Type1LD EVB from the combo box.
- Click on Next. Give a name which be used as prefix for all MicroEJ platform projects. For instance: `MyPlatform`.

Figure 2.2. New MicroEJ Platform Naming



- Click on `Finish`. The selected example is imported as several projects prefixed by the given name:

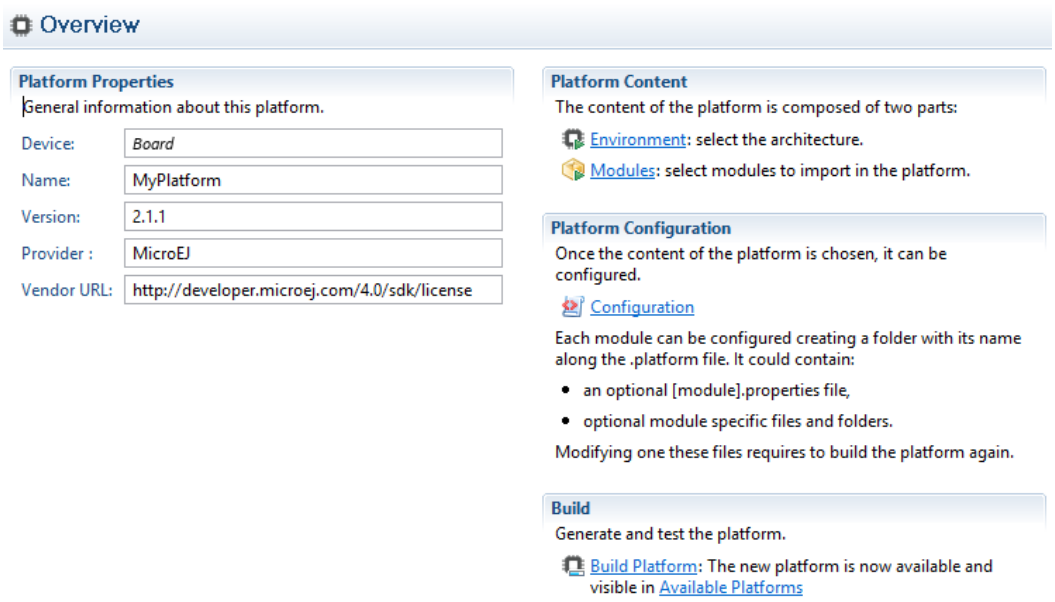
  - MurataType1LD-MyPlatform-CM3_GCC48-configuration: Contains the platform reference implementation configuration description. Some modules are described in a specific sub-folder / with some optional configuration files (`.properties` and / or `.xml`).

  - MurataType1LD-MyPlatform-CM3_GCC48-bsp: Contains a ready-to-use BSP software project for the Murata Type1LD EVB board, including a MicroEJ SDK project, an implementation of MicroEJ core engine (and extensions) port on FreeRTOS RTOS and the Murata Type1LD EVB board support package.

  The MicroEJ platform configuration file is automatically opened.

- From the MicroEJ platform configuration file, click on the link `Build Platform`

Figure 2.3. MicroEJ Platform Build



The build starts. This step may take several minutes. You can see the progress of the build steps in the MicroEJ console. Please wait for the final message:

```
BUILD SUCCESSFUL
```

At the end of the execution the MicroEJ platform is fully built for the Murata Type1LD EVB board and is ready to be linked into the MicroEJ SDK project. Its name is `MurataType1LD-MyPlatform-CM3_GCC48`.

The MicroEJ platform is now ready for use and available in the MicroEJ platforms list of your MicroEJ repository (`Windows > Preferences > MicroEJ > Platforms in workspace`).

2.2. Run an Example on the MicroEJ Simulator

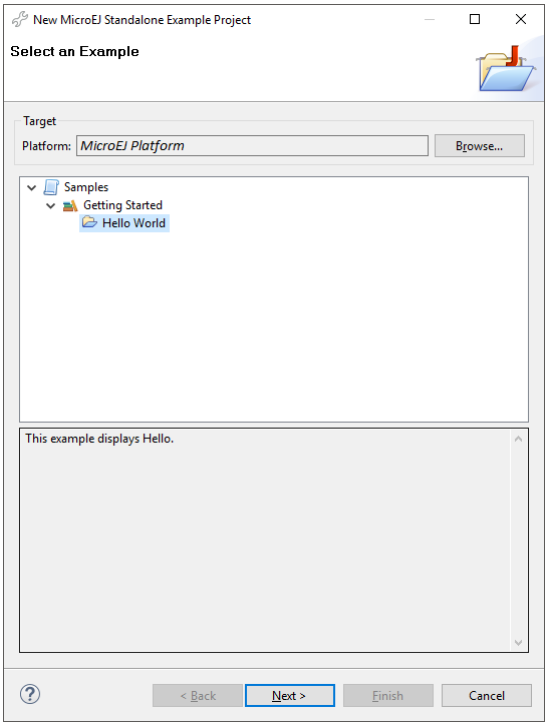
The aim of this chapter is to create a MicroEJ standalone application from a built-in example. Initially, this example will run on the MicroEJ simulator. Then, in the next section, this application will be compiled and deployed on the Murata Type1LD EVB board using the MicroEJ platform.

2.2.1. Create Example

- Open MicroEJ SDK.
- Open the `File > New > MicroEJ Standalone Example Project` menu.
- Select the MicroEJ platform MurataType1LD-MyPlatform-CM3_GCC48 from the combo box.

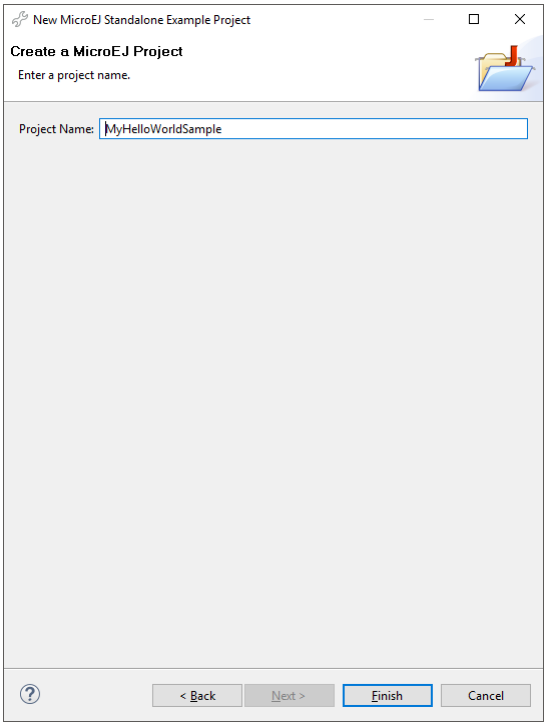
- Select the example `Samples > Getting Started > Hello World`.

Figure 2.4. MicroEJ Standalone Application Selection



- Click on Next. The next page suggests a name for the new project.

Figure 2.5. MicroEJ Standalone Application Naming

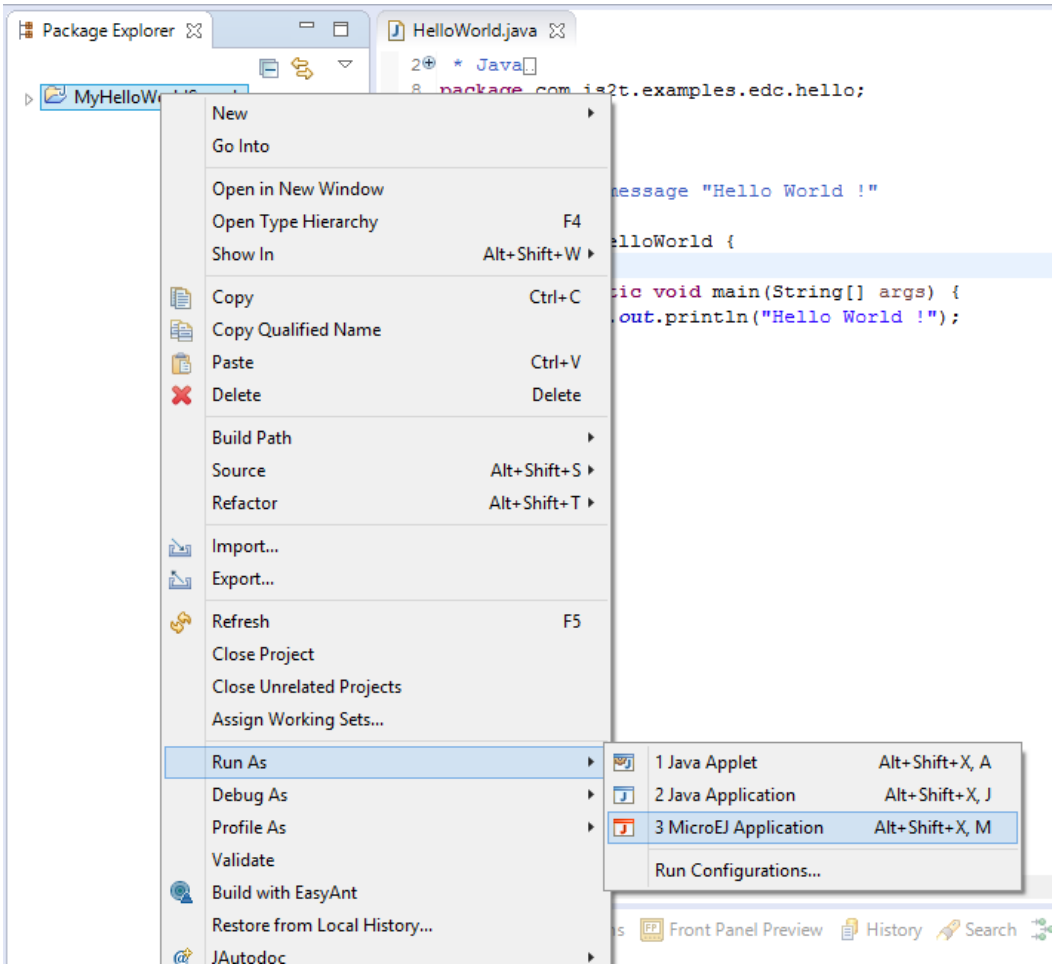


- Click on Finish. The selected example is imported into a project with the given name. The main class (the class which contains the `main()` method) is automatically opened.

2.2.2. Run Example

- Select the project in the Package Explorer tree
- Right-click on this project and select `Run As > MicroEJ Application`

Figure 2.6. MicroEJ Standalone Application Running



The application starts. It is executed on the MicroEJ simulator of the selected MicroEJ platform (MurataType1LD-MyPlatform-CM3_GCC48). The result of the test is printed in the console:

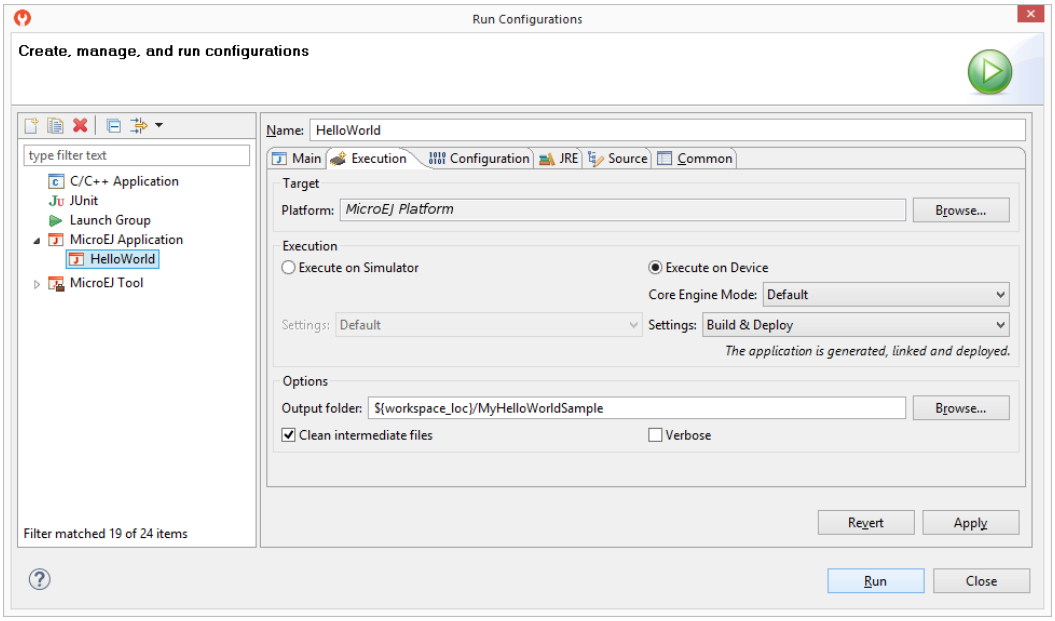
```
Hello World !
```

2.3. Run the Example on the Murata Type1LD EVB Board

2.3.1. Compile MicroEJ Standalone Application

- Open the run dialog (`Run > Run configurations...`).
- Select the MicroEJ Application launcher `HelloWorld`.
- Open `Execution` tab.
- Select `Execute on Device`.

Figure 2.7. Execution on Device



- Open `Configuration` tab and sub menu `Target > Deploy`. By default, an option is set to deploy the application library at a location known by the third-party IDE. If you want to deploy it elsewhere, unselect this option and enter your output path in the field below.
- Click Run: the application is compiled, and the compilation result (an ELF file) is copied into a well known location in the workspace. The MicroEJ SDK BSP project will search for it there when it performs the final link.

2.3.2. Link and Deploy MicroEJ Standalone Application

The aim of the final step is to:

- Compile the BSP project (such as drivers).
- Link the BSP and the others libraries (MicroEJ Core Engine, C stacks, MicroEJ standalone application etc.).
- Deploy a MicroEJ standalone application on the Murata Type1LD EVB board.

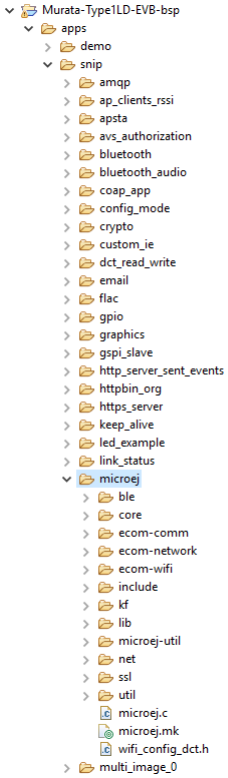
Note

This final step uses MicroEJ SDK.

The following steps are performed within MicroEJ.

- In MicroEJ SDK, expand the project MurataType1LD-MyPlatform-CM3_GCC48-bsp and the folder `apps/snip/microej/microej`. A Makefile (`microej.mk`) describes files to includes (C source files and header files) during the build part and build options.

  Figure 2.8. MicroEJ SDK Project Selection

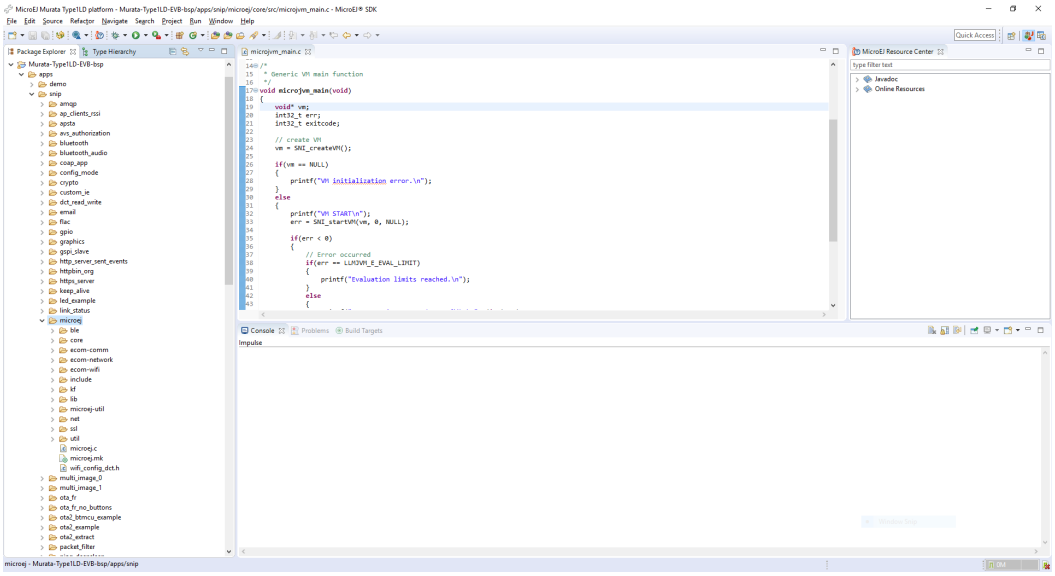
  Open the `Build Targets` view in your MicroEJ SDK (`Window > Show View > Other... > Make > Build Targets`). Then, click on the `snip.microej-MurataType1LD-FreeRTOS-LwIP-SDIO download download_apps run` make target to run the compilation and flash the board. You can now open a serial terminal link to the USB COM port of your board and see your application traces.

The following steps are performed within MicroEJ SDK.

Note

The target board can be flashed using OpenOCD.

- Figure 2.9. MicroEJ SDK IDE



Build the MicroEJ SDK project by clicking on the menu `Project > Build target`. The project is compiled and linked. See “Mandatory Connectors” to use the right connectors.

The application starts. The result of the execution is output on printf COM port. (See “Mandatory Connectors” to use the right connectors). Congratulations, you have deployed a MicroEJ standalone application on a MicroEJ platform.

Chapter 3. Specification

3.1. Overview

MicroEJ platform on Murata Type1LD EVB is based on board support package provided in the Cypress WICED SDK 6.2 and the associated Murata Type1LD package: (Cypress WICED SDK website [https://community.cypress.com/community/wiced-wifi]). It includes FreeRTOS, a TCP/IP network connection, some custom GPIOs. MicroEJ platform has been built MicroEJ SDK 4.1.5 IDE.

3.2. MicroEJ Platform Configuration

MicroEJ platform is based on MicroEJ architecture for ARM Cortex-M3.

Table 3.1. MCU Technical Specifications

MCU architecture	Cortex-M4 (STM32F412RGY6)
MCU Clock speed	96 MHz
Internal RAM	256 KB
External Flash	2 MB (QSPI)

MicroEJ platform uses several architecture extensions. The following table illustrates the MicroEJ architecture and extensions versions.

Table 3.2. MicroEJ Configuration

Name	Version
MicroEJ architecture	6.9.2
Network	7.0.0
Network Addons	1.1.2

3.3. Platform Output stream

MicroEJ platform uses USB connector as output print stream.

Implementation Note

The COM port is also used as the output stream for the *printf* calls.

The COM port uses the following parameters:

- Baudrate: 115200
- Data bits bits: 8
- Parity bits: None
- Stop bits: 1
- Flow control: None

3.4. RTOS Configuration

MicroEJ platform uses FreeRTOS 9.0.0. RTOS uses a heap to allocate all its objects: tasks stacks, task monitors, semaphores etc. The heap size is: 75 KB and is allocated in internal RAM. The following table illustrates the available tasks and their stack size.

Table 3.3. FreeRTOS Tasks

Task name	Size	Priority
RTOS idle task	350 B	0
RTOS timer	1 KB	9
MicroJvm	4 KB	8
SPI Slave	2 KB	9
Network Delegate	4 KB	9
System Monitor	512 B	9
Application Thread	6 KB	7
LwIP TCP/IP	4 KB	7

3.5. Memories

MicroEJ Plaform uses several internal and external memories. The following table illustrates the MCU and board memory layouts and sizes fixed by the MicroEJ platform.

Table 3.4. Internal RAM (256 KB)

Section Name	Size
MicroEJ standalone application stack blocks	512 * *n* bytes [a]
Pre-installed MicroEJ sandboxed application	*n* bytes [b]
MicroEJ platform internal heap	*n* bytes [c]
Any RW	*n* bytes [d]
MicroEJ standalone application heaps	1536 KB [e]
Multi applications working buffer	0 bytes
SSL buffers	Linked to C malloc heap

[a] *n* is the number of stack blocks defined in MicroEJ Application launcher options.

[b] *n* depends on the size defined in MicroEJ Application launcher options.

[c] *n* depends on memory configuration set in MicroEJ Application launcher options.

[d] *n* depends on MicroEJ application libraries used.

[e] Maximum size of the addition of MicroEJ heap size and MicroEJ immortal heap size. These sizes are defined in MicroEJ Application launcher options.

Table 3.5. Internal flash: ROM (1 MB)

Section Name	Size
Any RO	*n* bytes [a]

[a] *n* depends on MicroEJ application, MicroEJ libraries, Board support package, RTOS, drivers, etc.

Table 3.6. External flash: QSPI (2 MB)

Section Name	Size
MicroEJ standalone application resources	*n* bytes [a]

[a] *n* is the size of all MicroEJ standalone application resources.

3.6. Network

MicroEJ platform features a network interface with Wi-Fi as an underlying hardware media. A limited number of 8 sockets could be used for TCP connections, 1 for TCP listening (server) connections, 5 for incoming TCP (client) connections to a TCP server and 8 for UDP connections. A DHCP client could be activated to retrieve IP address. All DNS requests could be handled by a MicroEJ software implementation or a native one.

Implementation Note

MicroEJ platform uses LwIP v2.0.3 contained in the WICED SDK. This implementation uses the `newlib` dynamic memory allocator for all its memory allocation. The TCP MSS is 1460 bytes.

The network portage uses a WICED based implementation. A mechanism named network delegate, with a dedicated task, is used to request non-blocking operations and wait for completion or timeout.

The DHCP client is handled by LwIP and the DNS features use a MicroEJ native implementation based on the WICED SDK.

3.7. SSL

MicroEJ platform features a network secure interface. Available secured protocols are SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2. Supported keys and certificates formats are PKCS#5 and PKCS#12, PEM- or DER- encoded.

Implementation Note

MicroEJ platform uses WICED security APIs based on BESL library and mbedTLS v2.6.0.

3.8. Serial Communications

3.8.1. UART Connector

MicroEJ platform does not provide any serial connection.

Chapter 4. Board Configuration

Murata Type1LD EVB provides several connectors, each connector is used by the MicroEJ Core Engine itself or by a foundation library.

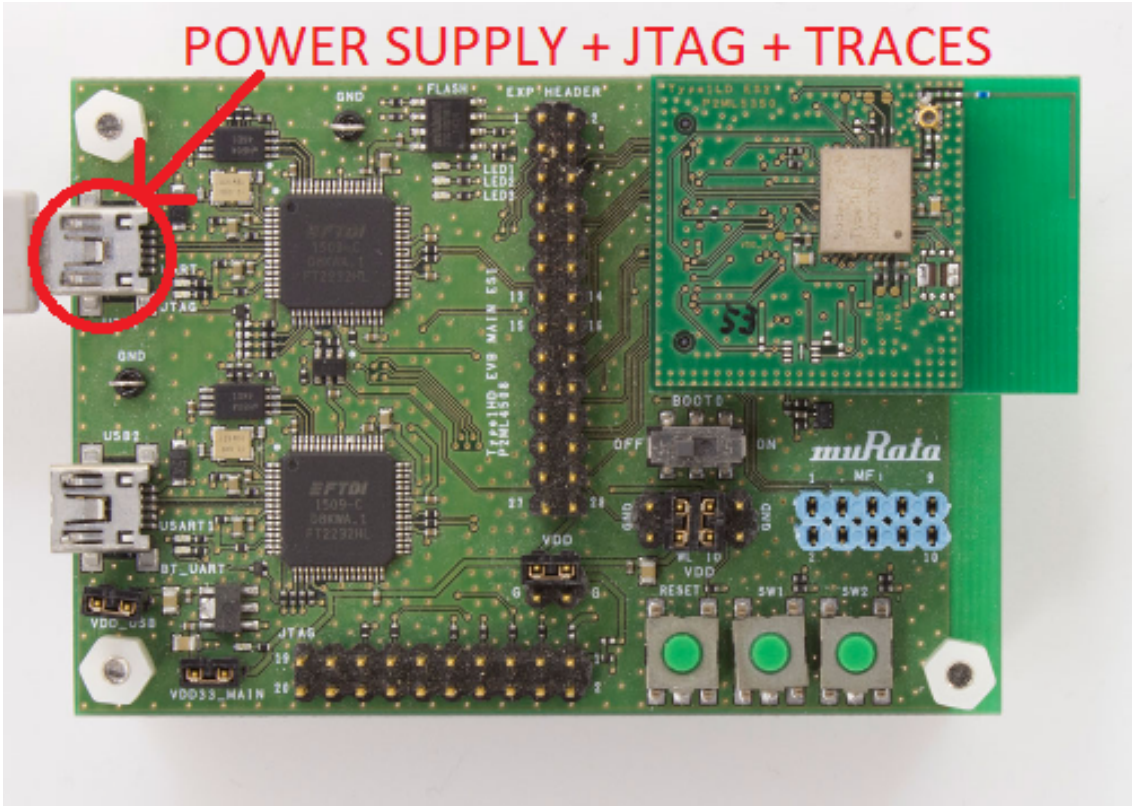
4.1. Mandatory Connectors

Murata Type1LD EVB provides a multi function USB port used as:

- Power supply connector
- Probe connector
- Virtual COM port

First of all, install the USB driver associated to your board. It's provided in the WICED SDK BSP (folder `[MurataType1LD_BSP_PATH]/tools/drivers/CYW9WCD1EVAL1/InstallDriver.exe`). Take a FTDI USB wire. Connect it to your PC and launch a the serial sniffer software of your choice.

Figure 4.1. Mandatory Connectors



Chapter 5. MicroEJ SDK Configuration

5.1. Install MicroEJ SDK

This section describes how to install a MicroEJ SDK development environment.

5.1.1. Download MicroEJ SDK

- Go to `https://developer.microej.com/getting-started-sdk.html`.
- Press the `Download MicroEJ SDK` button.
- Download the executable file (e.g. `MicroEJ-SDK-Installer-Win64-4.1.5.exe`).
- Run executable file and follow the installation steps. A new application named `MicroEJ SDK 4.1.5` shall have been installed.

Chapter 6. Changelog

6.1. Version 1.0.1

Bug fixes:

- Platform documentation.
- Board support package link step not called when `microejapp.o` file changed.

6.2. Version 1.0.0

Initial release of the platform.