MicroEJ Platform

Developer's Guide

MICROEJ®

S7G2DK 2.3.1

Reference:	TLT-802-DGI-Platform-S7G2DK
Version:	2.3.1
Revision:	2.3.1

Confidentiality & Intellectual Property

Revision History		
Revision 2.3.1	January 12th 2018	
Update of the changelog.		
Revision 2.3.0	December 22nd 2017	
Fixed the revisions history and updated the software versions.		
Revision 2.2.0	September 26th 2017	
Internal version		
Revision 1.0.1	December 30th 2016	
Early release for MicroEJ 4.1		
Revision 1.0	August 23th 2016	BJ
Initial version		

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 standalone application with MicroEJ SDK. Notes:

- This document is for the Renesas S7G2DK board.
- Please visit the website https://developer.microej.com for more information about S7G2DK products (platforms, videos, examples, application notes, etc.).

1.2. Scope

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

- Run a MicroEJ standalone application on the MicroEJ simulator.
- Run a MicroEJ standalone application on the MicroEJ platform and deploy it on the S7G2DK board.

1.3. Prerequisites

- PC with Windows 7 or later.
- The MicroEJ SDK environment must be installed.
- S7G2DK board.
- The Segger J-Link software.

Chapter 2. Develop and Run Your First MicroEJ Standalone Application

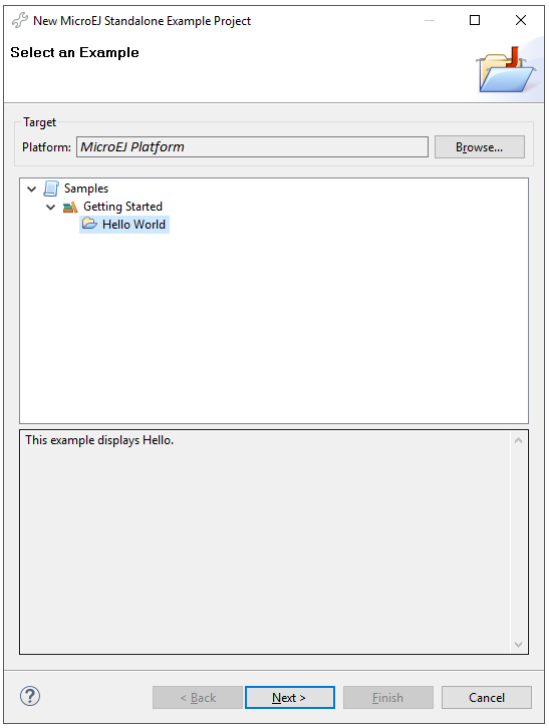
2.1. Run an Example on the MicroEJ Simulator

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 S7G2DK board using the MicroEJ platform.

2.1.1. Create Example

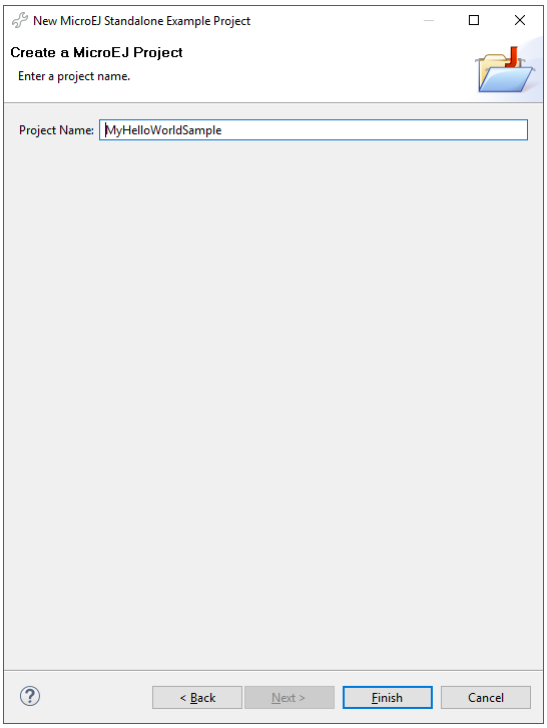
- Open MicroEJ SDK.
- Open the `File > New > MicroEJ Standalone Example Project` menu.
- Select the MicroEJ platform S7G2DK-LW9OX from the combo box.
- Select the example `Samples > Getting Started > Hello World`.

Figure 2.1. MicroEJ Standalone Application Selection



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

Figure 2.2. MicroEJ Standalone Application Naming

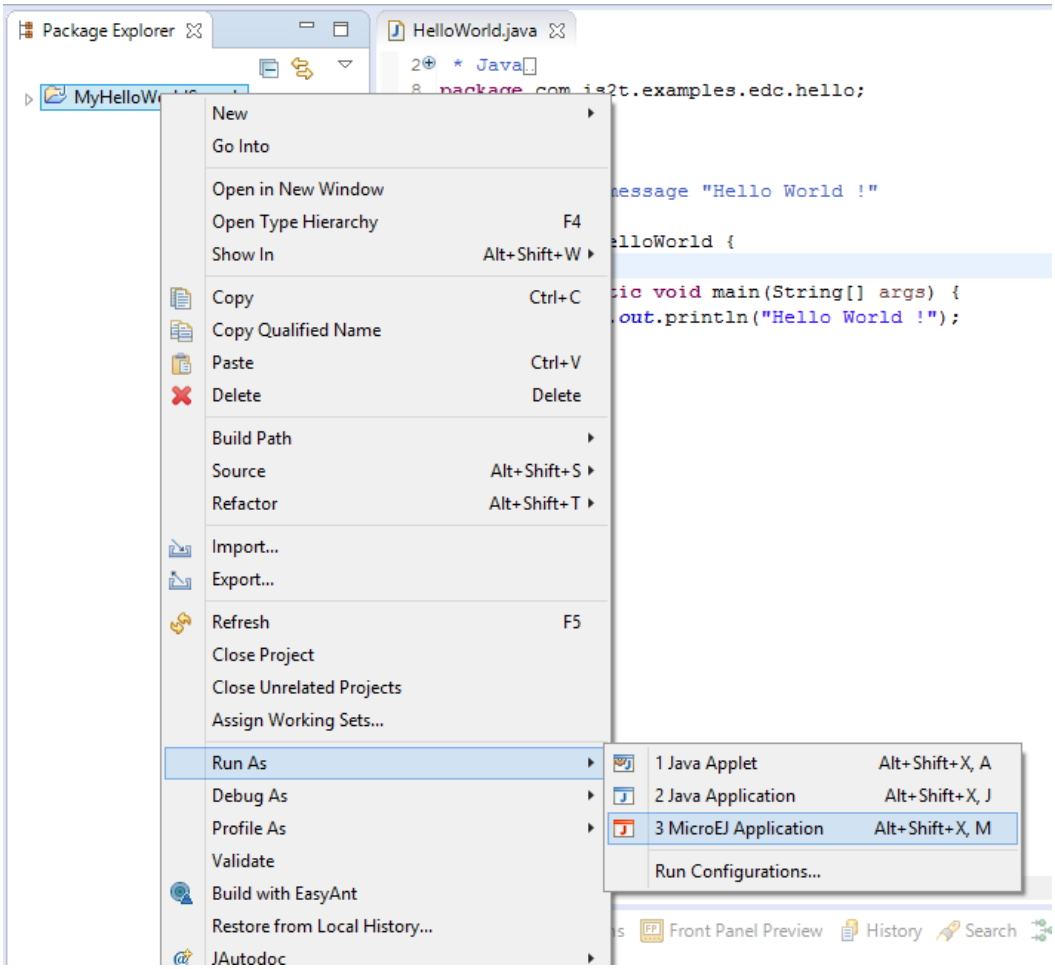


- Click on Finish. The selected example is imported into a project with the given name. The main class (the class which contains the `main()` method) is automatically opened.

2.1.2. Run Example

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

Figure 2.3. MicroEJ Standalone Application Running



The application starts. It is executed on the MicroEJ simulator of the selected MicroEJ platform (S7G2DK-LW9OX). The result of the test is printed in the console:

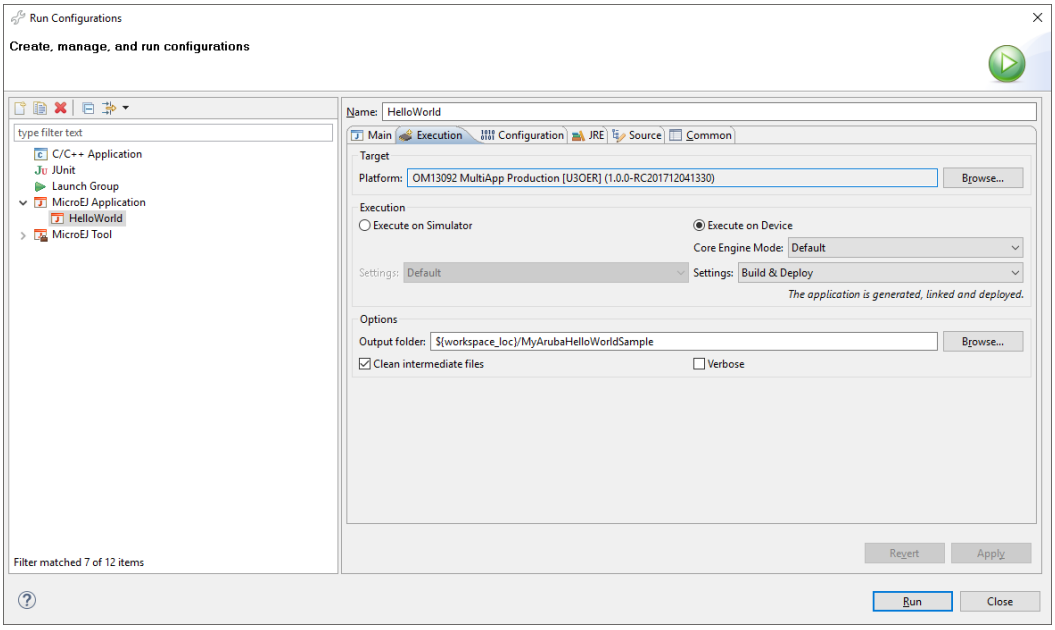
```
Hello World !
```

2.2. Run the Example on the S7G2DK Board

2.2.1. 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 2.4. Execution on Device



- In the JRE tab, pass the following VM argument: `-Dtoolchain.dir` and set its value to the path to the toolchain directory of IAR (typically `C:\Program Files (x86)\IAR Systems\Embedded Workbench 7.80\arm\bin`).

  Click Run: the application is compiled, and the compilation result (an ELF file) is copied into a well-known location in the example project. The SEGGER J-Link ARM tool has to be used to load the program on the board.

2.2.2. SEGGER J-Link Programming Tool

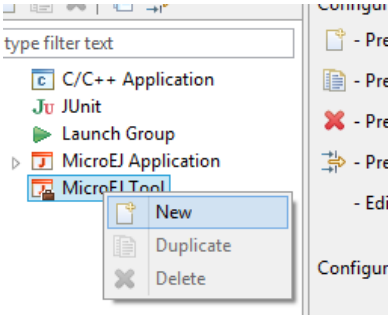
The aim of this section is to program a binary on the S7G2DK board.

Prerequisites

Download SEGGER J-Link ARM software and documentation pack from `https://www.segger.com/jlink-software.html` and install it on your machine.

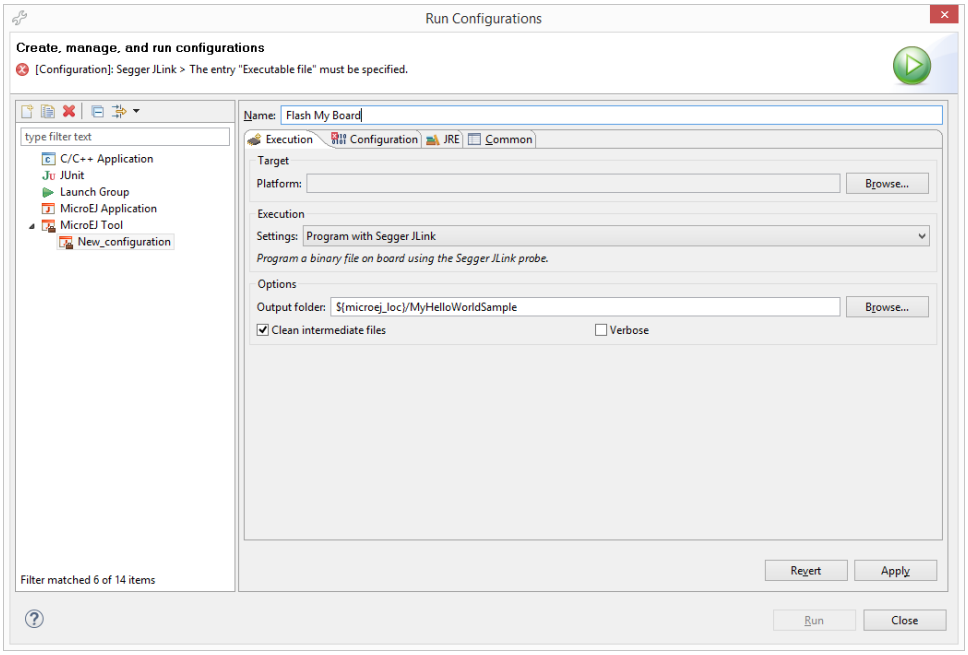
- Click on `Run > Run Configurations...`. Then right click on sub menu of `MicroEJ Tool` and select `New` to create a new MicroEJ Tool launcher:

Figure 2.5. MicroEJ Tool Launcher Creation



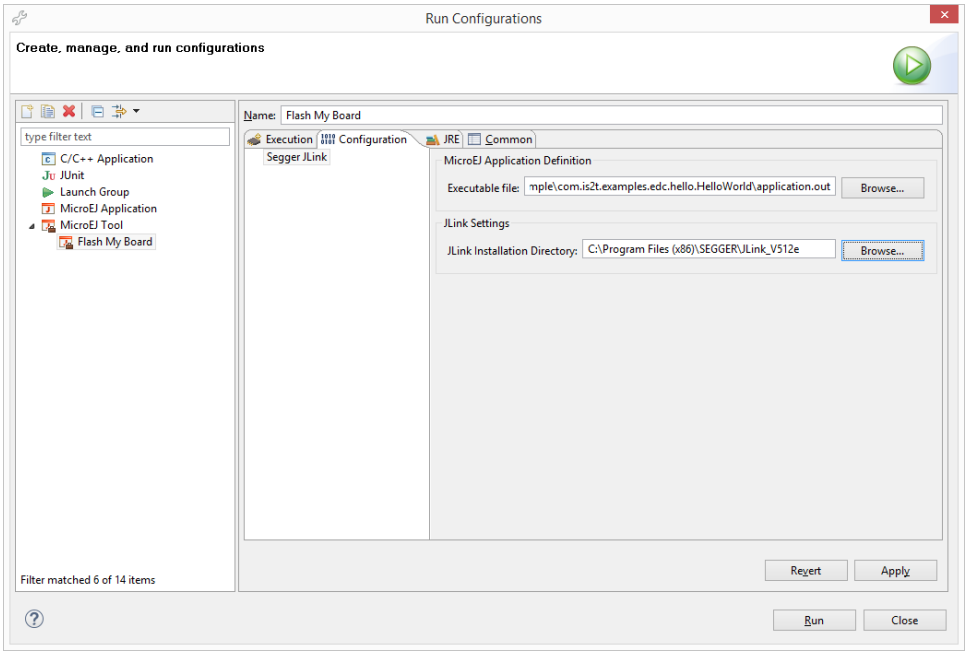
- A new window appears. Give a name to the launcher and set the `MicroEJ platform` field to `S7G2DK-LW90X` and the `Settings` field to `Program with Segger J-Link`

Figure 2.6. SEGGER J-Link MicroEJ SDK Tool Window



- Click on `Configuration` tab select the `application.out` file available in the MicroEJ project.

Figure 2.7. SEGGER J-Link MicroEJ SDK Tool Configuration Window



- Click on Run to program the binary.

  At the end of the execution the following message appears:

```
Flash programming complete successfully.
```

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 S7G2DK includes a graphical user interface, a TCP/IP network connection, a file system on SD card, COM connections (via UART) and some custom GPIOs.

3.2. MicroEJ Platform Configuration

MicroEJ platform is based on MicroEJ architecture for ARM Cortex-M4.

Table 3.1. MCU Technical Specifications

MCU architecture	Cortex-M4
MCU Clock speed	240 MHz
Internal RAM	640 KB
External RAM	32 MB
Internal Flash	4 MB
External Flash	32 MB

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	6.9.0
UI	9.0.2
Network	6.1.4
File System	3.0.0
HAL	1.0.4

3.3. Platform Output stream

MicroEJ platform uses a COM port as output print stream. The COM port is available on both connectors J112 and J7 and is connected to the MCU UART 1.

The COM port uses the following parameters:

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

3.4. 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 3.3. Internal RAM: SRAM (640 KB)

Section Name	Size
MicroEJ standalone application heaps	16384 bytes [a]
MicroEJ standalone application stack blocks	512 * *n* bytes [b]
MicroEJ platform internal heap	*n* bytes [c]
SSL buffers	128 KB

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

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

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

Table 3.4. External RAM: SDRAM (32 MB)

Section Name	Size
Display buffers	255 KB
MicroUI working buffer	4 MB
Multi applications working buffer	3 MB
Any RW	*n* bytes [a]

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

Table 3.5. Internal flash: Program Flash (4 KB)

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

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

3.5. Multi Applications

This MicroEJ platform includes the Multi applications mode. Multi applications mode allows to build a firmware that can manage MicroEJ sandboxed applications. Multi applications mode requires a specific memory area to load MicroEJ sandboxed applications. This memory area is located in external RAM and its default size is 3 MB.

3.6. Graphical User Interface

This MicroEJ plaform features a graphical user interface. It includes a display, a touch panel, three user buttons and a runtime PNG decoder.

3.6.1. Display

The display module drives a 480 x 272 TFT display. The pixel format is 16 bits-per-pixel: 5 bits for red color component, 6 bits for green color component and 5 bits for blue color component.

MicroUI requires a RAM buffer to store the dynamic images data. A dynamic image is an image decoded at runtime (PNG image) or an image created by the MicroEJ application using the `Image.create(width, height)` API. This buffer is located in SDRAM and the reserved size is 3 MB.

3.6.2. Inputs

Touch panel: All touch panel events are sent to the MicroEJ application using a `Pointer` event generator.

User buttons: The user buttons are reserved to the multi applications feature: they allow to force the kill of a sandboxed application.

3.7. Network

MicroEJ plaform features a network interface. Sockets are limited to 32. A DHCP client can be activated to retrieve an IP address.

3.8. File System

MicroEJ plaform features a file system interface. A SD card is used for the storage (previously formatted to a FAT32 file system). Up to 2 files can be opened simultaneously.

3.9. Serial Communications

3.9.1. UART Connectors

MicroEJ platform provides serial connections (ECOM COMM) on UART1, UART3, UART4 and UART8 ports.

UART1 pins are (RTS/CTS mode is not used):

- TX: P709; available on connectors J112 and J7
- RX: P708; available on connectors J112 and J7

UART3 pins are (RTS/CTS mode is not used):

- TX: P409; available on connector J10
- RX: P408; available on connector J10

UART4 pins are (RTS/CTS mode is not used):

- TX: P512; available on connector J10
- RX: P511; available on connector J10

UART8 pins are (RTS/CTS mode is not used):

- TX: PB04; available on connector J8
- RX: PB05; available on connector J8

3.10. HAL

MicroEJ platform provides several GPIOs programmable via the HAL foundation library. All GPIOs are available on external connectors (PMODA and PMODB). Digital pins are implemented by a GPIO access.

Analog input pins (ADC) are driven by ADC channels of ADC 1 and analog output pins (DAC) drive PWM channels via DAC A (channels 6 and 8)and DAC B (channel 8).

Each GPIO port / pin value is accessible using either:

- The global MCU declaration designation: all pins of all ports are grouped under only one virtual port (port 0) and have consecutive values: P000 has the ID 0, P001, the ID 1, P015 the ID 15, P100 the ID 16 and so on. For instance pin *P400* is accessible by `(0, 64)`. This designation is useful to target all MCU pins using only one virtual port.
- The standard MCU declaration designation: Port 0 has the ID 1, Port 1 the ID 2 etc. Each pin of each port is a value between 0 (Port*N*0) to 15 (Port*N*15). For instance pin *P400* is accessible by `(4, 0)`. This designation is useful to target a specific MCU pin.
- The physical board connectors designation. Board has 2 connectors: PMODA and PMODB, with respectively these IDs: 64, 65. For instance pin *P400* is accessible on connector PMODB, pin 4: `(65, 4)`. This designation is useful to target a physical connector pin without knowing which MCU pin it is.

The following table summarizes the exhaustive list of GPIOs ports accessible from HAL library, and the ranges of pins IDs:

Table 3.6. HAL GPIOs Ports and Pins

Port name	HAL port ID	Pins range
Global MCU virtual port	0	0 to 173
MCU port 0	1	0 to 15
MCU port 4	2	0 to 15
MCU port 9	4	0 to 15
MCU port B	6	0 to 15
Board physical port "PMODA"	64	1 to 12
Board physical port "PMODB"	65	1 to 12

The following table shows the exhaustive list of GPIOs connected to the HAL library, their IDs according the ports IDs and pins IDs (see before):

Table 3.7. HAL GPIOs Pins Designation Mapping

Port / Pin	MCU virtual port (1)	MCU port (2)	Board physical port (3)
P004	0, 4	1, 4	64, 7
P009	0, 9	1, 9	65, 1
P400	0, 64	5, 0	65, 0
P911	0, 155	10, 11	64, 8
P912	0, 156	10, 12	64, 9
P913	0, 157	10, 13	64, 10
PB02	0, 172	12, 2	64, 1
PB03	0, 173	12, 3	64, 4

The following table lists the hardware analog devices (ADC / DAC channels) used by HAL analog pins:

Table 3.8. HAL Analog IOs Pins Designation Mapping

Port / Pin	ADC id / channel	PWM timer / channel
P004	1 / 0	-
P400	-	A / 6
P911	-	B / 8
P912	-	A / 8

Chapter 4. Foundation Libraries

4.1. List

This table illustrates the available foundation libraries in the MicroEJ platform, and their versions.

Table 4.1. Foundation Libraries

Name	Version
EDC	1.2
BON	1.2
ECOM	1.1
ECOM-COMM	1.1
NLS	2.0
SNI	1.2
SP	2.0
KF	1.4
MicroUI	2.0
NET	1.1
SSL	2.0
FS	2.0

Chapter 5. Board Configuration

S7G2DK provides several connectors, each connector is used by the MicroEJ Core Engine itself or by a foundation library.

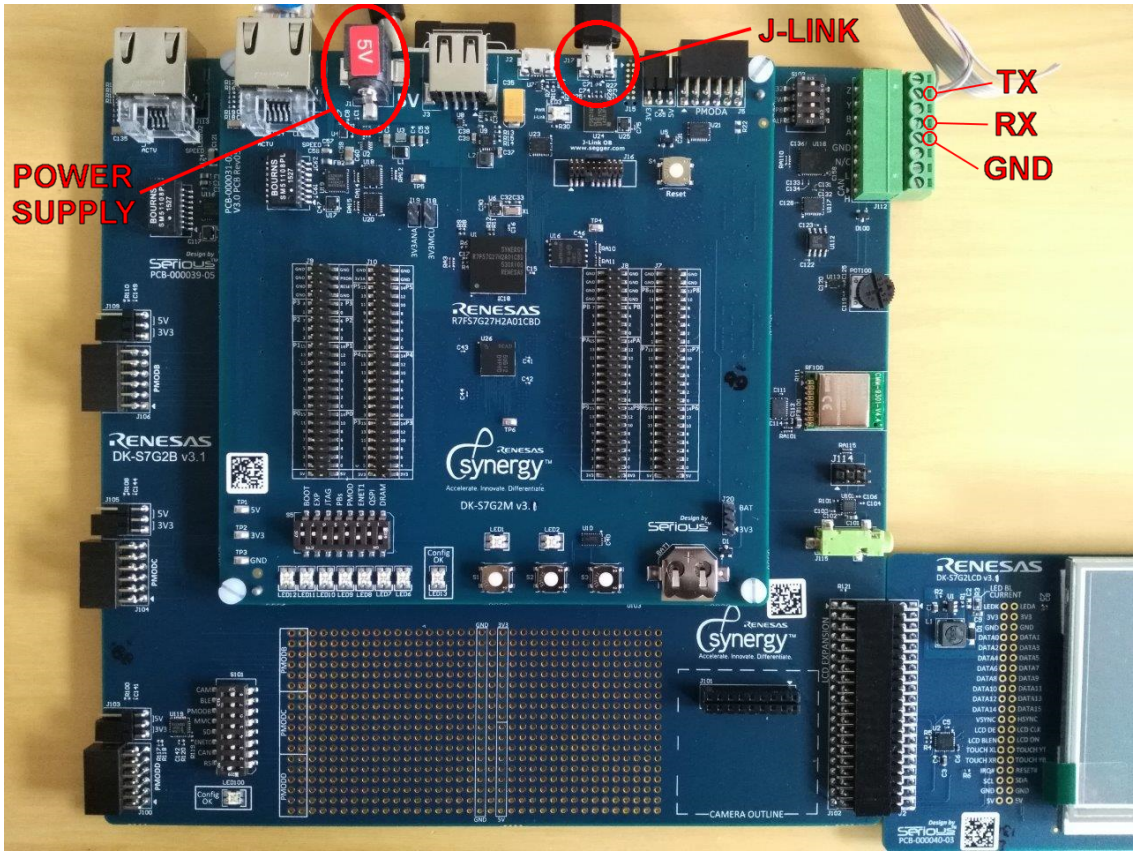
5.1. Mandatory Connectors

S7G2DK provides three connectors used as:

- Power supply connector
- Probe connector
- COM port

Plug a serial cable to a computer to see the MicroEJ standalone application `System.out.print` traces.

Figure 5.1. Mandatory Connectors

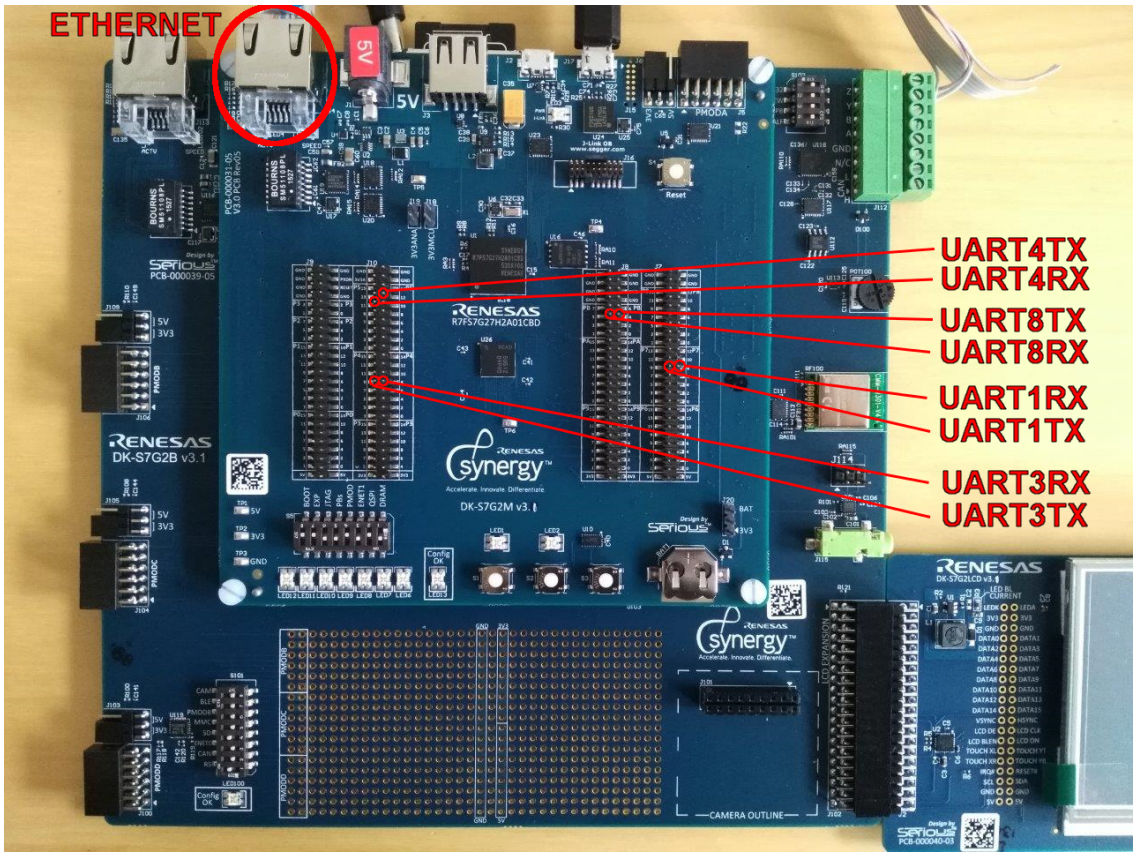


5.2. Communication Connectors

S7G2DK provides several communication ports:

- Ethernet
- Serial communication

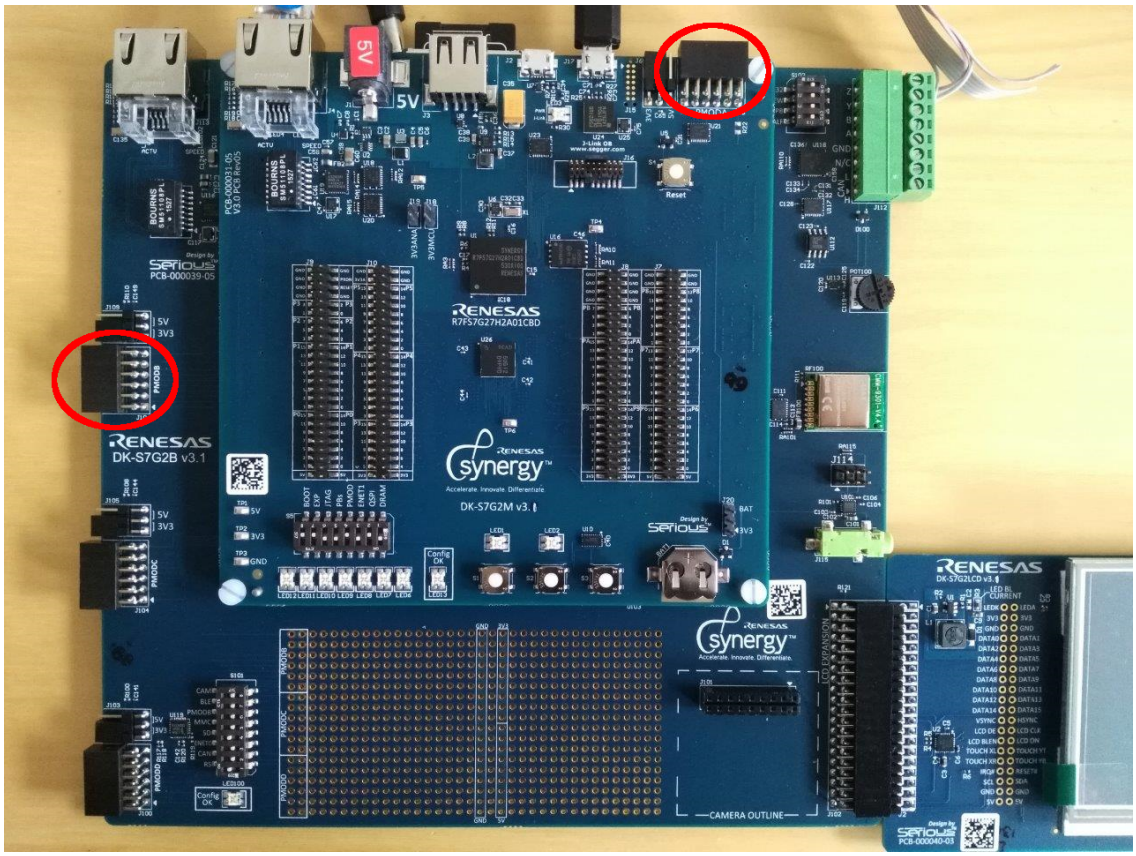
Figure 5.2. Communication Connectors



5.3. HAL Connectors

S7G2DK provides several HAL GPIOs on connectors PMODA and PMODB

Figure 5.3. HAL Connectors



Chapter 6. Changelog

6.1. Version 2.3.1

- Fixed license path in the build flow
- Documentation files update

6.2. Version 2.2.0

- Removed MWT from MicroUI
- Update of the wolfSSL license
- Update of the base architecture
- Update of the foundation libraries

6.3. Version 1.0.1

- WI 12372: invalid getcurrenttime returned by the LLMJVM_getcurrenttime
- WI 17713: improve the number of display stacks to build
- WI 17890: SDRAM or cache configuration problem
- WI 18363: NET 1.1 port for S7G2DK
- WI 18504: Missing implementation of LLLEDS_IMPL_getIntensity for S7G2DK

6.4. Version 1.0

Initial release of the platform.