MicroEJ Platform

Developer's Guide

NXP TWRK65F180M 1.2.0

Reference:	TLT-0792-DGI-Platform-NXP TWRK65F180M
Version:	1.2.0
Revision:	1.2.0

Revision History		
Revision 1.2.0	September 05th 2016	
Updated the simulator		
Revision 1.1.5	July 05th 2016	
Refactored the changelog section		
Revision 1.1.4	July 05th 2016	
Added changelog section		
Revision 1.1.3	June 30th 2016	
Fixed regression on display issue at reset		
Revision 1.1.2	June 28th 2016	
Fixed display issue at reset		
Revision 1.1.1	June 23th 2016	
Added a picture for the touchscreen strap instructions		
Revision 1.1.0	June 4th 2016	
First release		

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 NXP TWR-K65F180M board.

> **Note**
>
> In this document, all the references to the TWR-K65F180M board point to a full NXP Tower system made of the following:
>
> - 2 TWR-ELEV boards (primary and secondary)
> - 1 TWR-K65F180M board
> - 1 TWR-SER board
> - 1 TWR-LCD board connected to the primary TWR-ELEV board

- Please visit the website https://developer.microej.com for more information about TWR-K65F180M 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 TWR-K65F180M board.

1.3. Prerequisites

- PC with Windows 7 or later.
- The MicroEJ SDK environment must be installed.
- TWR-K65F180M board.
- The Segger J-Link software.

Chapter 2. Develop and Run Your First MicroEJ Standalone Application

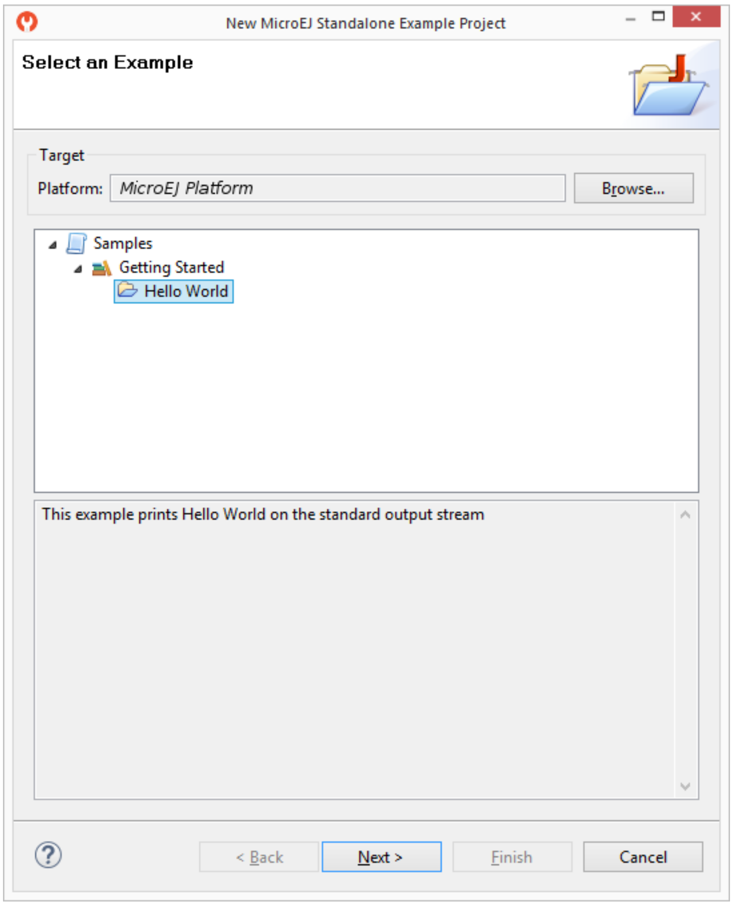
2.1. Run an Example on the MicroEJ Simulator

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 TWR-K65F180M board using the MicroEJ platform.

2.1.1. Create Example

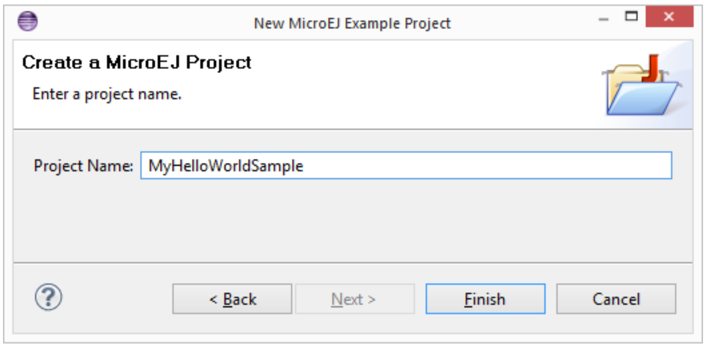
- Open MicroEJ SDK.
- Open the `File > New > MicroEJ Standalone Example Project` menu.
- Select the MicroEJ platform TWRK65F180M-LNTOC from the combo box.
- Select the example `Samples > Getting Started > Hello World`.

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

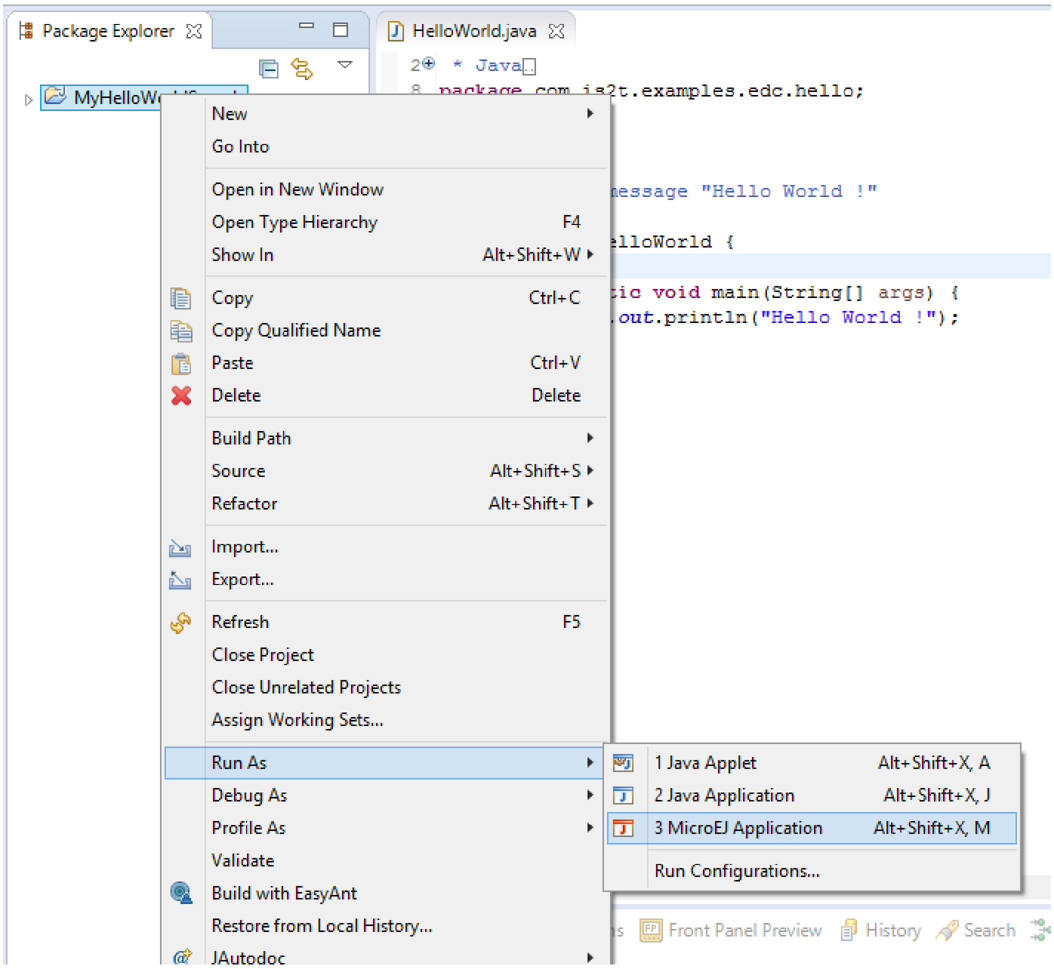


- Click on Finish. The selected example is imported into a project with the given name. The main class (the class which contains the `main()` method) is automatically opened.

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 (TWRK65F180M-LNTOC). The result of the test is printed in the console:

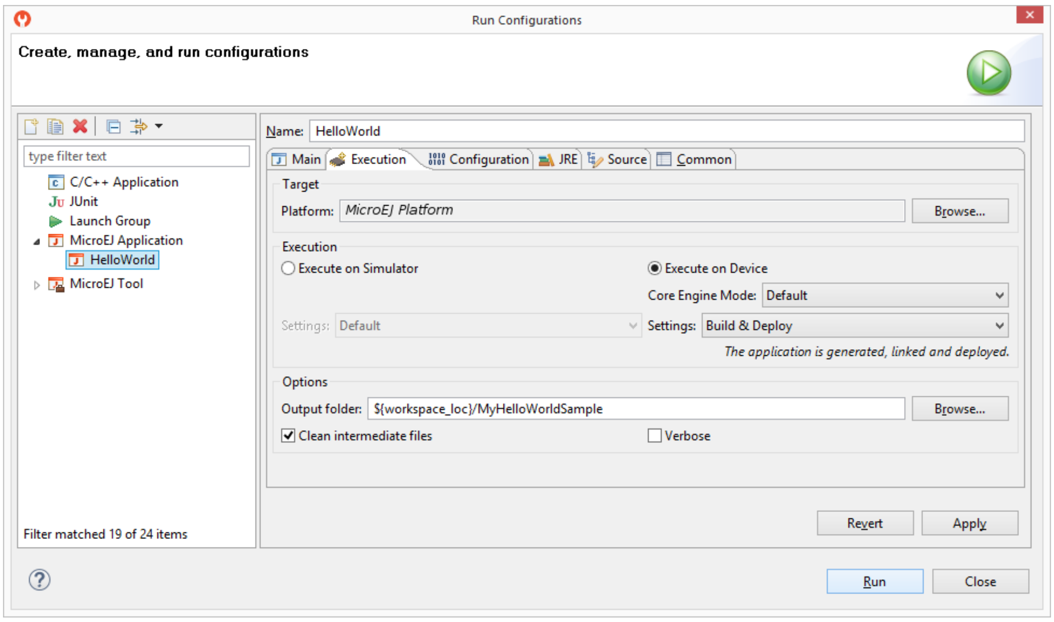
```
Hello World !
```

2.2. Run the Example on the TWR-K65F180M 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



- 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

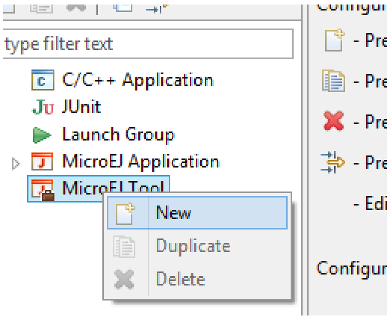
The aim of this section is to program a binary on the TWR-K65F180M board.

Prerequisites

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

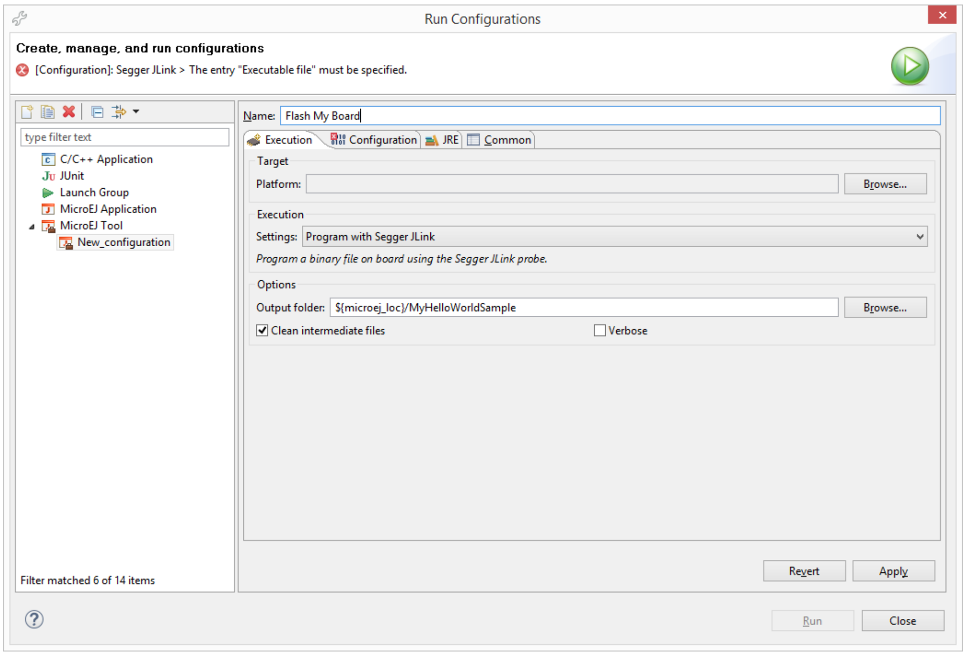
- Click on `Run > Run Configurations...`. Then right click on sub menu of `MicroEJ Tool` and select New to create a new MicroEJ Tool launcher:

Figure 2.5. MicroEJ Tool Launcher Creation



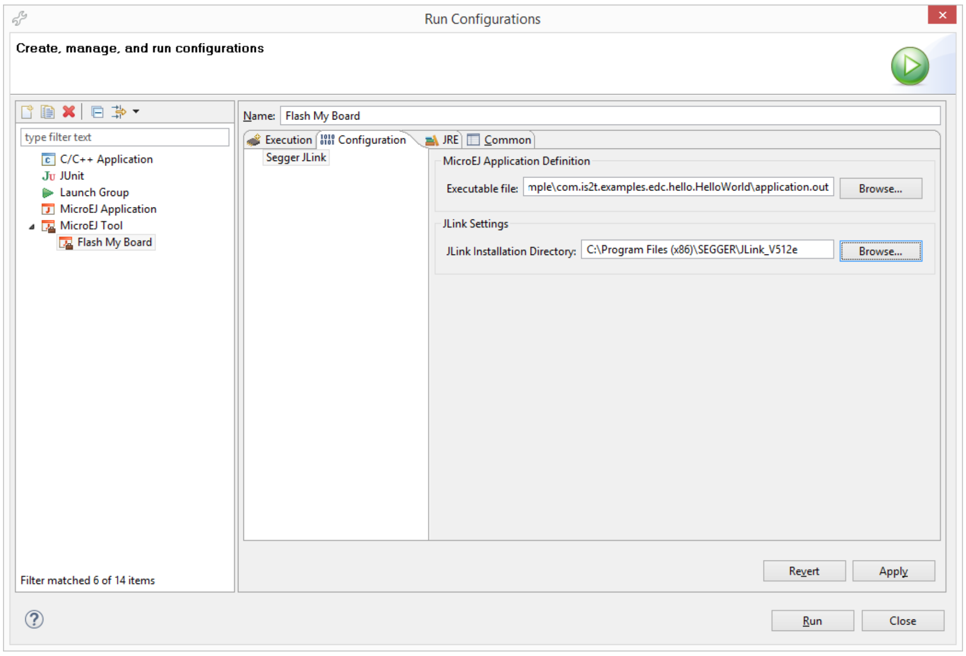
- A new window appears. Give a name to the launcher and set the `MicroEJ platform` field to `TWRK65F180M-LNTOC` and the `Settings` field to `Program with Segger J-Link`

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. Congratulations, you have deployed a MicroEJ standalone application on a MicroEJ platform.

Chapter 3. Specification

3.1. Overview

MicroEJ platform on TWR-K65F180M includes a graphical user interface, a TCP/IP network connection, a file system on SD-Card, a serial connection 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 (MK65FN2M0VMI18)
MCU Clock speed	120 MHz
Internal Flash	2 MB
Internal RAM	256 KB
External RAM	256 MB (SDRAM)

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

Table 3.2. MicroEJ Configuration

Name	Version
MicroEJ architecture	6.1.1
UI	7.2.1
Network	5.0.0
File System	2.1.1
HAL	1.0.2

3.3. Platform Output stream

MicroEJ platform uses a COM port as output print stream. The COM port is available via the OpenSDA connector and is connected to the MCU UART2.

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_L (64 KB)

Section Name	Size
MicroEJ standalone application stack blocks	512 * *n* bytes [a]
MicroEJ platform internal heap	*n* bytes [b]

[a] *n* is the number of stack blocks defined in MicroEJ Application launcher options.
[b] *n* depends on memory configuration set in MicroEJ Application launcher options.

Table 3.4. Internal RAM: SRAM_U (192 KB)

Section Name	Size
Ethernet buffers	15560 KB
Any RW	*n* bytes [a]

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

Table 3.5. External RAM: SDRAM (256 MB)

Section Name	Size
Display buffers	300 KB
MicroUI working buffer	3 MB
Multi applications working buffer	3 MB
SSL buffers	128 KB
MicroEJ standalone application heaps	2048 bytes [a]

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

Table 3.6. Internal flash: Program Flash (2 MB)

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

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

3.5. Graphical User Interface

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

3.5.1. Display

The display module drives a 320 x 240 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.

3.5.2. 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.

3.6. Network

MicroEJ plaform features a network interface. Socket limits are :

- 10 sockets for TCP client connections
- 5 sockets for TCP server connections
- 6 sockets for UDP connections

A DHCP client can be activated to retrieve an IP address.

3.7. File System

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

3.8. Serial Communications

3.8.1. UART Connector

MicroEJ platform provides one serial connection (ECOM COMM) on UART0 port. UART0 pins are (RTS/CTS mode is not used):

- RX: PTA1; available on connector J8, plot 41
- TX: PTA2; available on connector J8, plot 42

3.9. HAL

MicroEJ platform provides several GPIOs programmable via the HAL foundation library. All GPIOs are available either on the elevator connectors (J5 and J6 on the secondary elevator) or on the internal soldering plots (on the primary elevator). Digital pins are implemented by a GPIO access, analog input pin (ADC) is driven by ADC channel of ADC 1 and analog output pins (DAC) use DAC0 and DAC1.

Each GPIO port / pin value is accessible using either:

- The global MCU designation: all pins of all ports are grouped under only one virtual port (port 0) and have consecutive values: PTA0 has the ID 0, PTA1, the ID 1, PTA15 the ID 15, PTB0 the ID 32 and so on. For instance pin *PTD11* is accessible by `(0, 139)`. This designation is useful to target all MCU pins using only one virtual port.
- The standard MCU designation: PORTA has the ID 1, PORTB the ID 2 etc. Each pin of each port is a value between 0 (PORT*N*-0) to 31 (PORT*N*-31). For instance pin *PTD11* is accessible by `(4, 11)`. This designation is useful to target a specific MCU pin.
- The physical board connectors designation. The elevators have 4 connectors: J8, J9, J5 and J6, with respectively these IDs: 64, 65, 66 and 67. For instance pin *PTD11* is accessible on connector J5, pin46: `(65, 46)`. 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 GPIO ports accessible from HAL library, and the ranges of pin IDs:

Table 3.7. HAL GPIO Ports and Pins

Port name	HAL port ID	Pins range
Global MCU virtual port	0	0 to 159
MCU port A	1	0 to 31
MCU port B	2	0 to 31
MCU port D	4	0 to 31
MCU port E	5	0 to 31
Board physical port "J6"	67	1 to 80
Board physical port "J8"	64	1 to 80
Board physical port "J9"	65	1 to 80

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

Table 3.8. HAL GPIOs Pins Designation Mapping

Port / Pin	MCU virtual port (1)	MCU port (2)	Board physical port (3)
PTA0	0, 0	1, 0	-
PTA3	0, 3	1, 3	-
PTA9	0, 9	1, 9	64, 11
PTA11	0, 11	1, 11	-
PTA18	0, 18	1, 18	-

Port / Pin	MCU virtual port (1)	MCU port (2)	Board physical port (3)
PTA19	0, 19	1, 19	-
PTA28	0, 28	1, 28	65, 35
PTA29	0, 29	1, 29	64, 35
PTB19	0, 51	2, 19	65, 72
PTD0	0, 128	4, 0	65, 63
PTD10	0, 138	4, 10	65, 23
PTD11	0, 139	4, 11	65, 46
PTD12	0, 140	4, 12	65, 48
PTD13	0, 141	4, 13	65, 45
PTD14	0, 142	4, 14	65, 44
PTE6	0, 262	5, 6	64, 21
PTE7	0, 263	5, 7	64, 24
PTE8	0, 264	5, 8	64, 59 / 64, 60
PTE9	0, 265	5, 9	64, 58 / 64, 61
PTE10	0, 266	5, 10	64, 25
PTE11	0, 267	5, 11	64, 23
PTE12	0, 268	5, 12	64, 22
PTE18	0, 274	5, 18	64, 8
PTE19	0, 275	5, 19	64, 7
PTE27	0, 283	5, 27	65, 55

This table indicates the useful ADC / DAC channels for HAL analog pins:

Table 3.9. HAL Analog IOs Pins Designation Mapping

Port / Pin	Board physical port	ADC channel	DAC output
DAC0	64, 32	-	0
DAC1	65, 32	-	1
ADC1_SE16	67, 30	16	-

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
MicroUI	2.0
MWT	2.1
NET	1.0
FS	2.0

Chapter 5. Board Configuration

TWR-K65F180M provides several connectors, each connector is used by the MicroEJ Core Engine itself or by a foundation library.

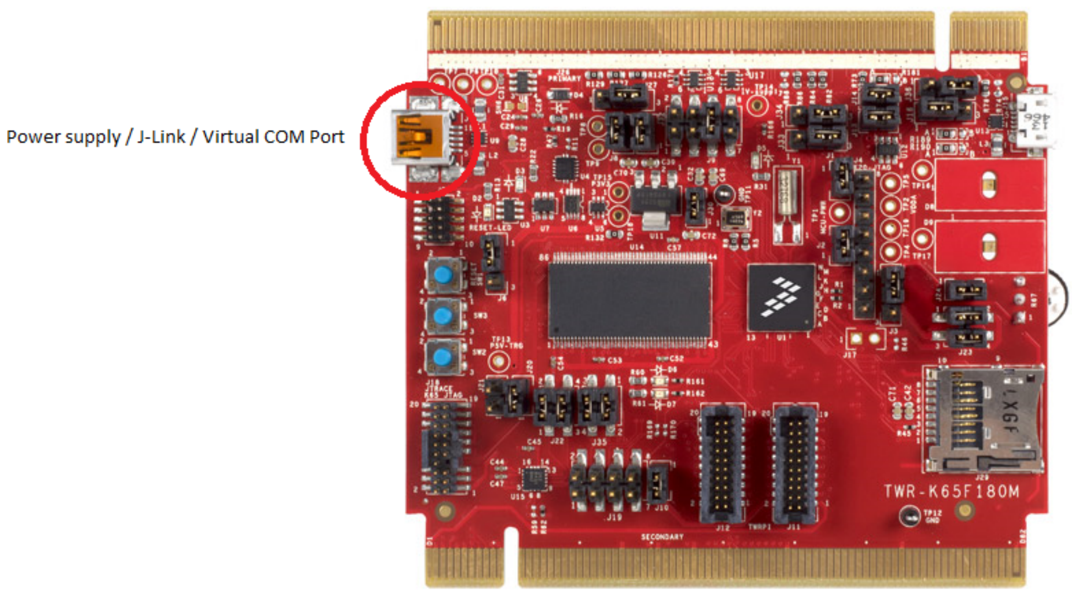
5.1. Mandatory Connectors

TWR-K65F180M provides a multi function USB port used as:

- Power supply connector
- Probe connector
- Virtual COM port

First of all, ensure the jumpers are set to their factory settings. Then just plug a mini-USB cable to a computer to power on the board, be able to program an application on it and to see the MicroEJ stand-alone application `System.out.print` traces.

Figure 5.1. Mandatory Connectors



5.2. Communication Connectors

TWR-K65F180M provides several communication ports:

- Ethernet
- Serial communication

To enable the Ethernet communications, make sure to have the following jumper settings on the TWR-SER board:

- *J2 (CLK_SEL)* to position 3-4
- *J3 (CLKIN_SEL)* to position 2-3
- *J12 (ETH_CONFIG)* to position 9-10

Figure 5.2. Communication Connectors - Ethernet

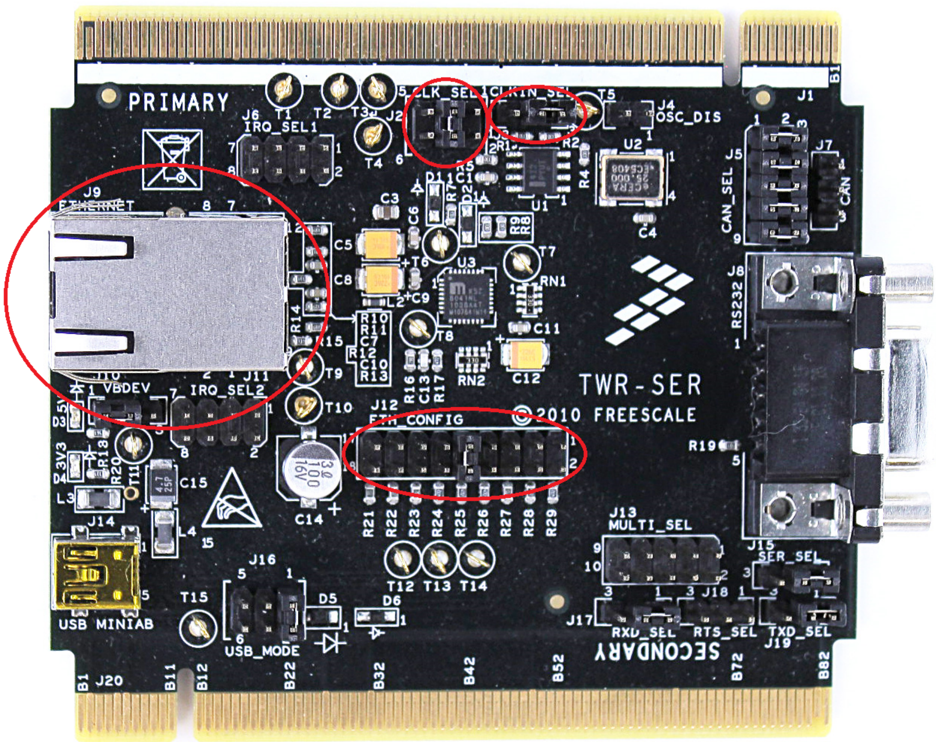
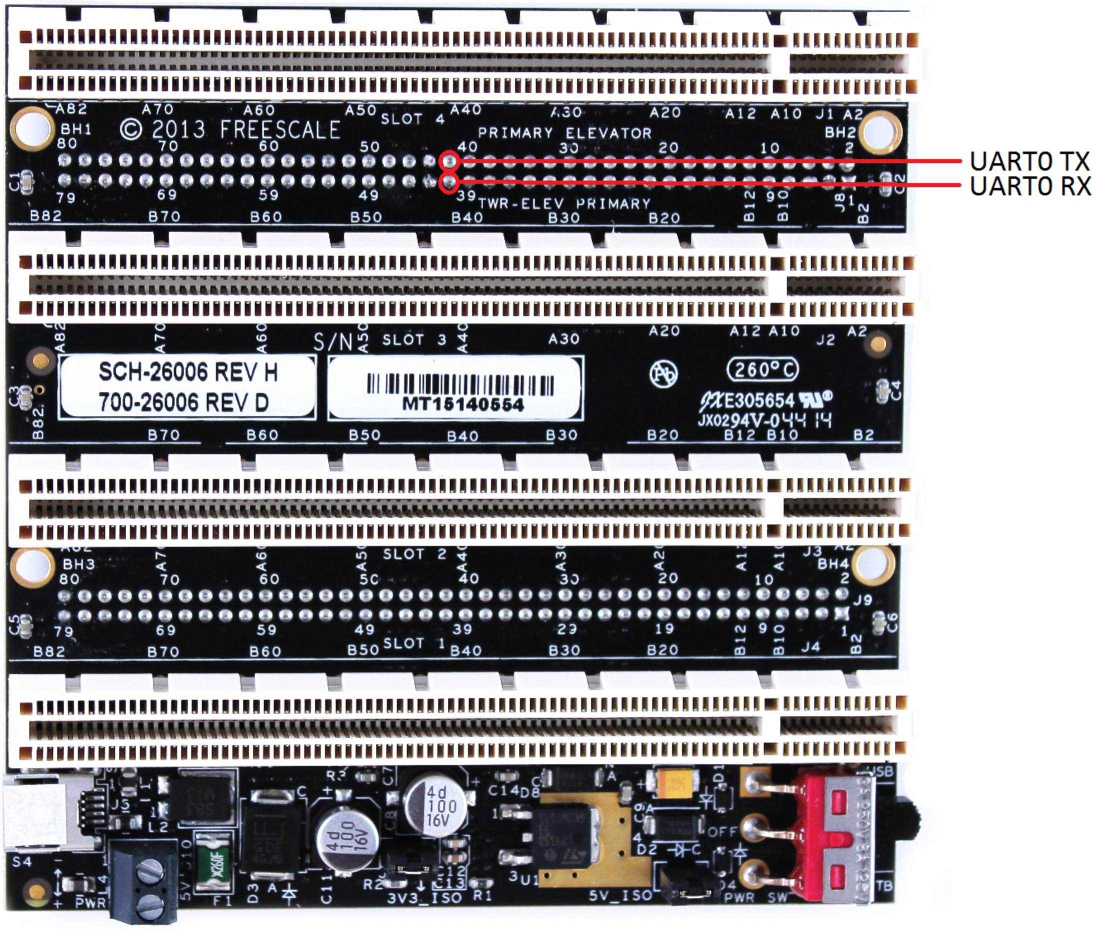


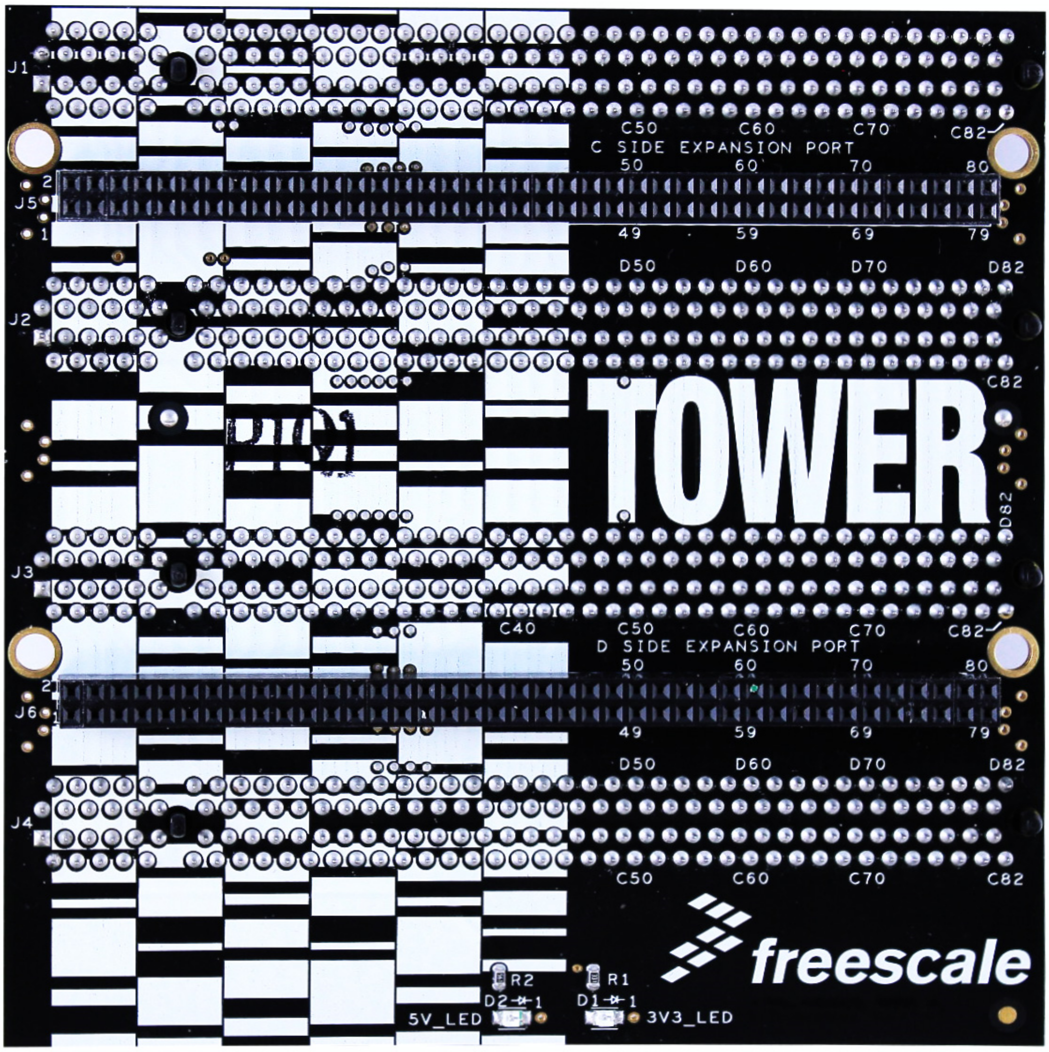
Figure 5.3. Communication Connectors - Serial



5.3. HAL Connectors

TWR-K65F180M provides several HAL GPIOs on connectors J5, J6, J8 and J9 of the TWR-ELEV boards. Since the TWR-LCD board is plugged in the connectors J8 and J9, all the GPIOs normally available through them should be accessed from the inner side of the TWR-ELEV board via the soldering points.

Figure 5.4. HAL Connectors



5.4. UI Setup

TWR-K65F180M provides a LCD display via its TWR-LCD board. This board must be set up to correctly display the MicroEJ standalone application and to relay the touch information.

First of all, the TWR-LCD board must be flashed with a neutral firmware. This board embeds a MCU to be able to independently drive the LCD screen and use the touch interface and we need to be sure that it won't interfere with the UI operations of MicroEJ. You can get this firmware from the DVD included in the TWR-LCD box:

- Insert the DVD in the disc drive of your computer.
- In a file browser, look for the `TWR_LCD_Demo_Projects.exe` binary in the `fscommand` folder and run it.
- Follow the installation steps and when you are prompted to browse the installed files, click the OK button.

The firmware to use is the file called *JM128_Bootloader.S19*.

To update the firmware of the TWR-LCD board, follow this steps:

- Connect the mini-USB connector to a computer to power the TWR-LCD board. It must be separated from the other components of the TWR-K65F180M. Leaving it attached to its TWR-ELEV board is OK but no other board must be attached to it. If so, make sure to use the USB connector of the TWR-LCD board, not the TWR-ELEV one.

- Hold the *SW4 (BTLD)* button (bottom-right) and press the *SW3 (JMRST)* button (bottom-left). The board should beep and be enumerated as a Mass Storage Device on the host computer. Release the *SW4 (BTLD)* button.

- Copy the firmware file (e.g. *JM128_Bootloader.S19*) to the BOOTLOADER drive. The board will load the .S19 file, beep, and restart. Once restarted, the board is identified and a file named *SUCCESS.TXT* should be present.

The next step is to setup the configuration switches. There are 2 of them: SW1 (bottom-left) and SW5 (top of the reverse side).

Table 5.1. SW1 Configuration Switches

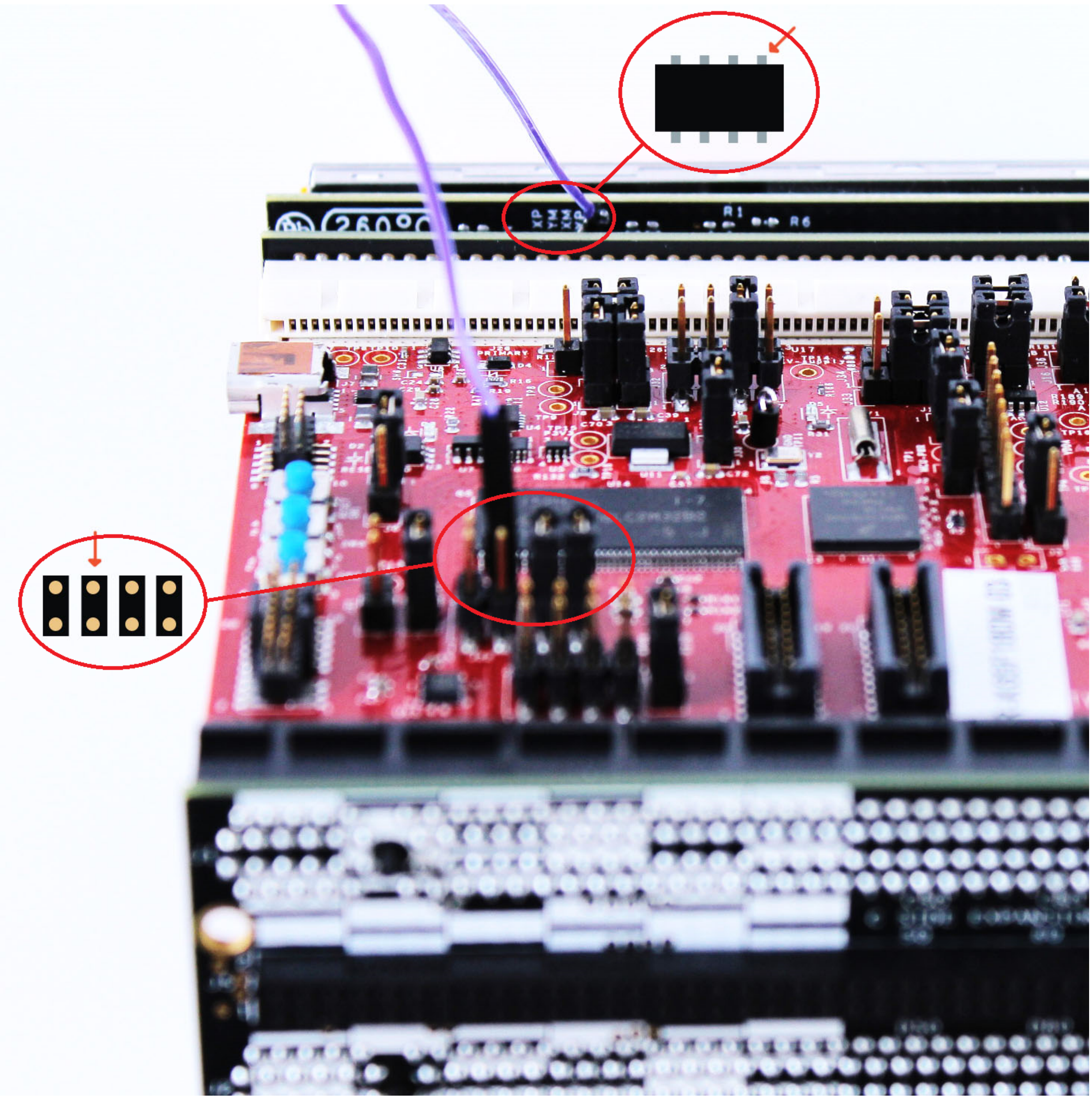
Switch #	Value
1	ON
2	OFF
3	ON
4	OFF
5	ON
6	ON
7	ON
8	OFF

Table 5.2. SW5 Configuration Switches

Switch #	Value
1	ON
2	ON
3	ON
4	OFF

Finally, the *YP* switch (number 4) of the *SW5* jumper needs to be redirected to the pin 4 of the *J22* jumper of the TWR-K65F180M board.

Figure 5.5. Touchscreen Strap



Chapter 6. Changelog

6.1. Version 1.2.0

Updated documentation. Updated the simulator frontpanel.

6.2. Version 1.1.5

Initial release of the platform.