

MicroEJ Platform

Developer's Guide



MICROEJ®

STM32F429IEVAL 2.2.1

Reference:	TLT-0816-DGI-Platform-STM32F429IEVAL
Version:	2.2.1
Revision:	2.2.1

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 2.2.1	January 17th 2017	
First release		
Revision 2.2.0	July 5th 2016	
First release		

Table of Contents

1. Introduction	1
1.1. Intended Audience	1
1.2. Scope	1
1.3. Prerequisites	1
2. Develop and Run Your First MicroEJ Standalone Application	2
2.1. Run an Example on the MicroEJ Simulator	2
2.1.1. Create Example	2
2.1.2. Run Example	3
2.2. Run the Example on the STM32F429I-EVAL Board	4
2.2.1. Compile MicroEJ Standalone Application	4
2.2.2. STM32 ST-LINK utility Programming Tool	4
3. Specification	7
3.1. Overview	7
3.2. MicroEJ Platform Configuration	7
3.3. Platform Output stream	7
3.4. Memories	8
3.5. Multi Applications	8
3.6. Graphical User Interface	9
3.6.1. Display	9
3.6.2. Inputs	9
3.7. Network	9
3.8. File System	9
3.9. Serial Communications	10
3.9.1. UART Connector	10
3.9.2. USB Connector	10
3.10. HAL	10
4. Foundation Libraries	13
4.1. List	13
5. Board Configuration	14
5.1. Mandatory Connectors	14
5.2. Communication Connectors	14
5.3. HAL Connectors	15
6. Changelog	17
6.1. Version 2.2.0	17

List of Figures

- 2.1. MicroEJ Standalone Application Selection 2
- 2.2. MicroEJ Standalone Application Naming 3
- 2.3. MicroEJ Standalone Application Running 3
- 2.4. Execution on Device 4
- 2.5. MicroEJ Tool Launcher Creation 5
- 2.6. STLink MicroEJ SDK Tool Window 5
- 2.7. STLink MicroEJ SDK Tool Configuration Window 6
- 5.1. Mandatory Connectors 14
- 5.2. Communication Connectors 15
- 5.3. HAL Connectors 16

List of Tables

3.1. MCU Technical Specifications	7
3.2. MicroEJ Configuration	7
3.3. Internal RAMs (192 + 64 KB)	8
3.4. External RAM: SDRAM (8 MB)	8
3.5. Internal flash (2 MB)	8
3.6. HAL GPIOs Ports and Pins	11
3.7. HAL GPIOs Declaration (port, pin)	11
3.8. HAL Analog GPIOs Devices	12
4.1. Foundation Libraries	13

Chapter 1. Introduction

1.1. Intended Audience

The intended audience for this document are developers who wish to develop their first MicroEJ stand-alone application with MicroEJ SDK. Notes:

- This document is for the STM32F429I-EVAL board.
- Please visit the website <https://developer.microej.com> for more information about STM32F429I-EVAL 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 STM32F429I-EVAL board.

1.3. Prerequisites

- PC with Windows 7 or later.
- The MicroEJ SDK environment must be installed.
- STM32F429I-EVAL board.
- The STM32 ST-LINK utility (minimal version 3.7.1).

Chapter 2. Develop and Run Your First MicroEJ Standalone Application

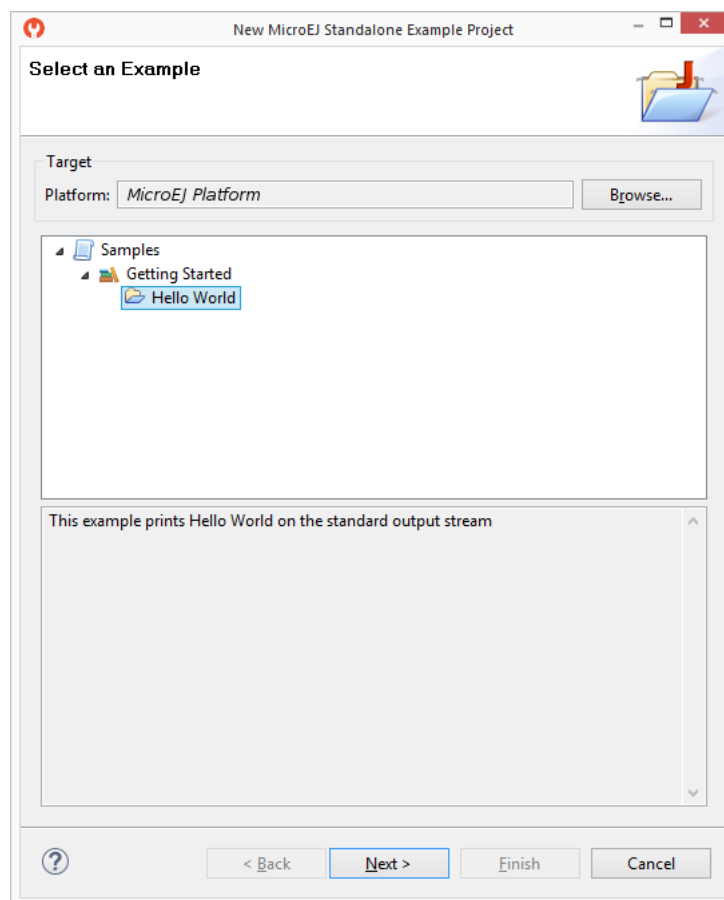
2.1. Run an Example on the MicroEJ Simulator

The aim of this chapter is to create a MicroEJ MicroEJ standalone application from a built-in example. This example will initially be run on the MicroEJ simulator. Then, in the next section, this application will be compiled and deployed on the STM32F429I-EVAL board using the MicroEJ platform.

2.1.1. Create Example

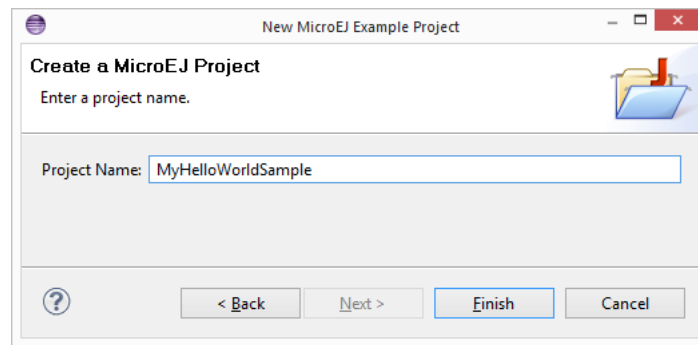
- Open MicroEJ SDK.
- Open the menu: `File > New > MicroEJ Standalone Example Project`.
- Select the MicroEJ platform `STM32F429I-EVAL-WithoutSSL-CM4hardfp_ARMCC5-FreeRTOS-eval` from the combo box.
- Select the example `Samples > Getting Started > Hello World`.

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

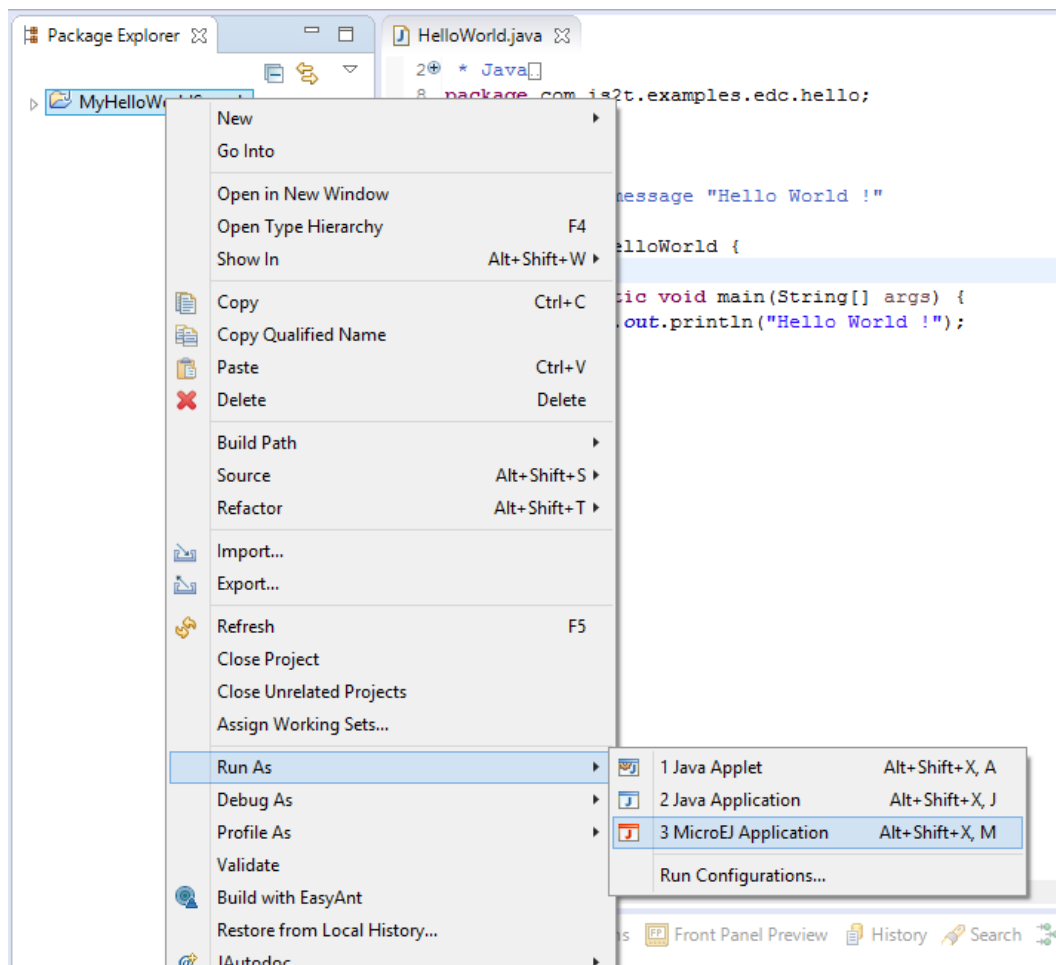


- Click on Finish. The selected example is imported into a project with the given name. The main class (the class which contains the `main()` method) is automatically opened.

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 (STM32F429IEVAL-WithoutSSL-CM4hardfp_ARMCC5-FreeRTOS-eval). The result of the test is printed in the console:

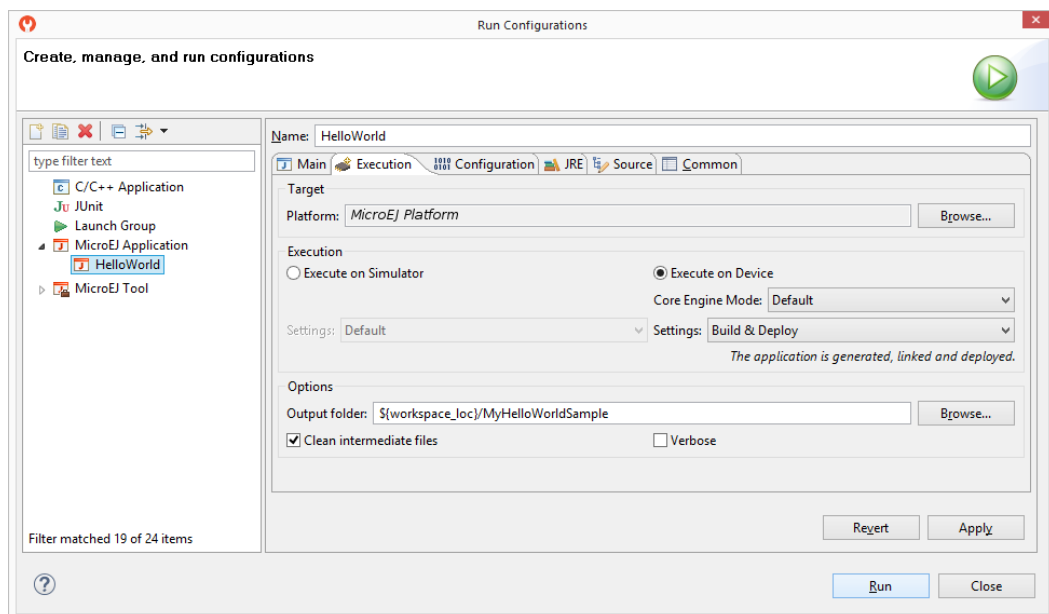
```
Hello World !
```

2.2. Run the Example on the STM32F429I-EVAL 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 Execution 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 project example. The STM32 ST-LINK utility tool has to be used to load the program on the board.

2.2.2. STM32 ST-LINK utility Programming Tool

The aim of this section is to program a binary on the STM32F429I-EVAL board.

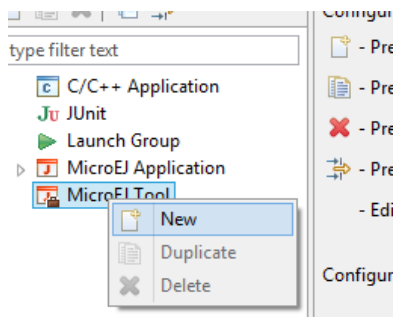


Prerequisites

Download STM32 ST-LINK utility software from http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link004.html and install it on your machine.

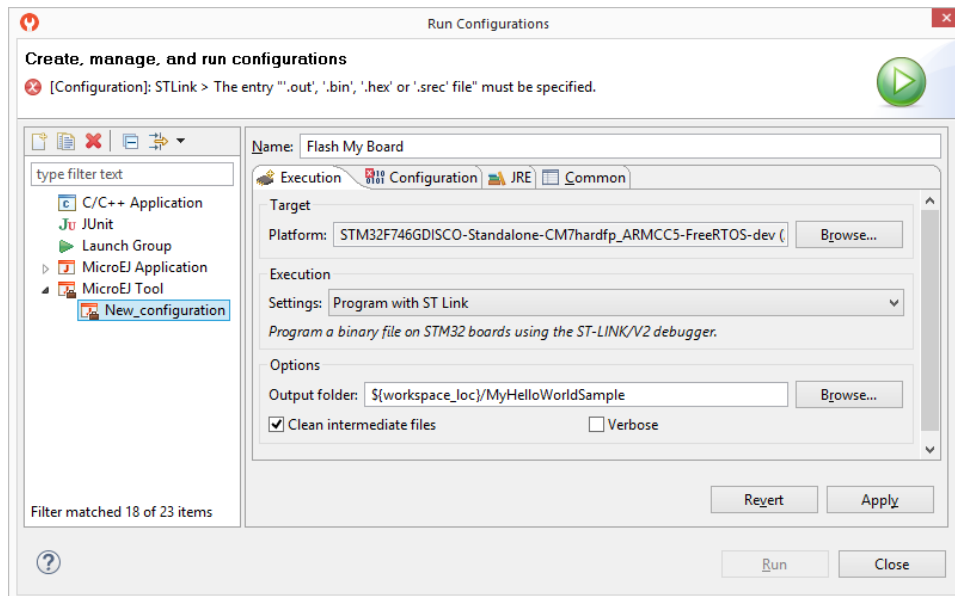
- Click on Run > Run Configurations... Then right click on sub menu of MicroEJ Tool and select New to create a new MicroEJ Tool launcher:

Figure 2.5. MicroEJ Tool Launcher Creation



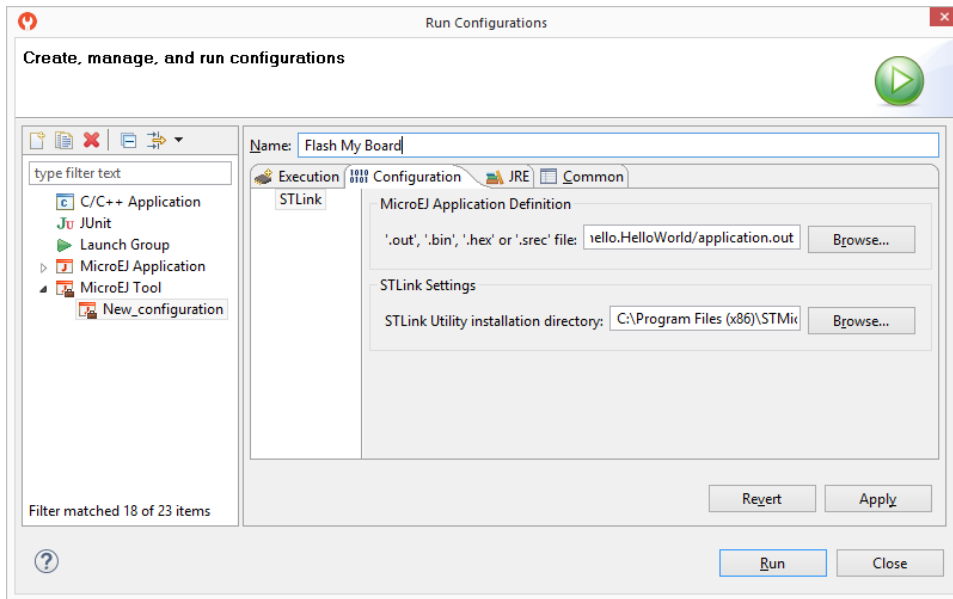
- A new window appears. Give a name to the launcher and select the right MicroEJ platform: STM32F429IEVAL-WithoutSSL-CM4hardfp_ARMCC5-FreeRTOS-eval and the right Settings: Program with ST link:

Figure 2.6. STLink MicroEJ SDK Tool Window



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

Figure 2.7. STLink MicroEJ SDK Tool Configuration Window



- Click on Run to program the binary.

At the end of the execution the following message appeared:

```
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 STM32F429I-EVAL includes a graphical user interface, a TCP/IP network connection, a file system on microSD card, COM connections (UART and USB-CDC) and some custom GPIOs.

3.2. MicroEJ Platform Configuration

MicroEJ platform is based on MicroEJ architecture for STMicroelectronics STM32F4J.

Table 3.1. MCU Technical Specifications

MCU architecture	Cortex-M4 (STM32F429NI)
MCU Clock speed	180 MHz
Internal Flash	2 MB
Internal RAM	256 KB
External RAM	8 MB (SDRAM)

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	6.6.0
UI	7.4.7
Network	5.0.0
File System	2.2.1
HAL	1.0.3

3.3. Platform Output stream

MicroEJ platform uses CN8 connector as output print stream; this COM port is connected to the MCU USART 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 Platform 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 RAMs (192 + 64 KB)

Section Name	Size
Ethernet buffers	15560 KB
MicroEJ standalone application stack blocks	512 * n bytes ^a
MicroEJ platform internal heap	n bytes ^b
Any RW	n bytes ^c

^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.

^c n depends on MicroEJ application libraries used.

Table 3.4. External RAM: SDRAM (8 MB)

Section Name	Size
Display buffers	510 KB
MicroUI working buffer	3 MB
Multi applications working buffer	3 MB
SSL buffers	65 KB
MicroEJ standalone application heaps	1408 KB ^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 3.5. Internal flash (2 MB)

Section Name	Size
Any RO	n bytes ^a
MicroEJ standalone application resources	n bytes ^b
Pre-installed MicroEJ sandboxed application	n bytes ^c

^a n depends on MicroEJ application, MicroEJ libraries, Board support package, RTOS, drivers, etc.

^b n is the size of all MicroEJ standalone application resources.

^c n depends on the size defined in MicroEJ Application launcher options.

3.5. Multi Applications

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

3.6. Graphical User Interface

MicroEJ platform features a graphical user interface. It includes a display, a touch panel, two user buttons, a joystick 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. The display device is clocked at 60Hz and the MicroEJ application drawings are synchronized on this display tick.

The display module uses the STM32F4 hardware acceleration to perform some drawings: the ChromArt (DMA2D). The DMA2D renders all *fill rectangles* (`GraphicsContext.fillRectangle()`) and performs the drawings of all images.

MicroUI requires a RAM buffer to store the dynamic images data. A dynamic is an image decoded at runtime (PNG image) or an image created by the MicroEJ application thanks the API `Image.create(width, height)`. 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 thanks a `Pointer` event generator.

User button: The button "WAKEUP" is reserved to the multi applications feature: it allows to force to kill a sandboxed application. The button "TAMPER" event is sent to the MicroEJ application thanks a `Commands` event generator. The command is `MENU`.

Joystick: All joystick events are sent to the MicroEJ application thanks a `Commands` event generator. The command are `UP`, `DOWN`, `RIGHT`, `LEFT`.

3.7. Network

MicroEJ platform features a network interface. A limited number of 10 sockets could be used for TCP connections, 5 for TCP listening (server) connections and 6 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.

3.8. File System

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

3.9. Serial Communications

3.9.1. UART Connector

MicroEJ platform provides three serial communications (ECOM COMM) on UART2, UART6 and UART7. UARTn pins are (RTS/CTS mode is not used):

- UART2 TX: PD5; available on connector CN12 26
- UART2 RX: PD6; available on connector CN12 6
- UART6 TX: PC6; available on connector CN7 58
- UART6 RX: PC7; available on connector CN7 56
- UART7 TX: PF7; available on connector CN6 8
- UART7 RX: PF6; available on connector CN6 7

3.9.2. USB Connector

MicroEJ platform features a serial communication (ECOM COMM) on USB port USBH: USB CDC.

3.10. HAL

MicroEJ platform uses provides several GPIOs to connect HAL foundation library. All GPIOs are available on ARDUINO connectors (CN6, CN7, CN11 and CN12). Digital pins are implemented by a GPIO access, analog input pins (ADC) are driven by ADC channels of ADC 0 and 3 and analog output pins (DAC) drive PWM channels of timers 1, 3, 4, 12 and 13.

Each GPIO port / pin value is accessible by several ways:

1. Using the global MCU declaration: all pins of all ports are grouped under only one virtual port (port 0) and have consecutive values: PA0 has the ID 0, PA1, the ID 1, PA15 the ID 15, PB0 the ID 16 and so on. For instance pin *PF8* is accessible by (0 , 88). This declaration is useful to target all MCU pins using only one virtual port.
2. Using the standard MCU declaration: PortA has the ID 1, PortB the ID 2 etc. Each pin of each port is a value between 0 (PortN-0) to 15 (PortN-15). For instance pin *PF8* is accessible by (6 , 8). This declaration is useful to target a specific MCU pin.
3. Using the physical board connectors. Board has 3 connectors: CN4, CN5 and CN7 (CN6 is not connected to the MCU), with respectively these IDs: 64, 65 and 67. For instance pin *PF8* is accessible on connector CN6, pin17: (64 , 17). This declaration is useful to target a physical connector pin without knowing which MCU pin it is.

The following table summaries 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 143
MCU port A	1	0 to 15
MCU port B	2	0 to 15
MCU port D	4	0 to 15
MCU port F	6	0 to 15
MCU port G	7	0 to 15
Board physical port "CN6"	64	1 to 60
Board physical port "CN7"	65	1 to 60
Board physical port "CN11"	66	1 to 40
Board physical port "CN12"	67	1 to 40

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

Table 3.7. HAL GPIOs Declaration (port, pin)

Port / Pin	MCU virtual port (1)	MCU port (2)	Board physical port (3)
PA4	0, 4	1, 4	64, 21
PA6	0, 6	1, 6	64, 20
PA8	0, 8	1, 8	65, 52
PA15	0, 15	1, 15	65, 32
PB4	0, 20	2, 4	65, 24
PB7	0, 23	2, 7	65, 21
PB14	0, 30	2, 14	64, 35
PB15	0, 31	2, 15	64, 38
PD3	0, 51	4, 3	65, 23
PD4	0, 52	4, 4	67, 28
PD7	0, 55	4, 7	65, 1
PD11	0, 59	4, 11	66, 21
PD12	0, 60	4, 12	66, 23
PD13	0, 61	4, 13	66, 31
PF8	0, 88	6, 8	64, 17
PF9	0, 89	6, 9	64, 12
PF10	0, 90	6, 10	64, 9
PG3	0, 99	7, 3	66, 34
PG9	0, 105	7, 9	67, 27

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

Table 3.8. HAL Analog GPIOs Devices

Port / Pin	ADC id / channel	PWM timer / channel
PA4	1 / 4	-
PA6	1 / 6	13 / 1
PA8	-	1 / 1
PB4	-	3 / 1
PB7	-	4 / 2
PB14	-	12 / 1
PB15	-	12 / 2
PD12	-	4 / 1
PF8	3 / 6	-
PF9	3 / 7	-
PF10	3 / 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
MWT	2.1
NET	1.0
FS	2.0

Chapter 5. Board Configuration

STM32F429I-EVAL provides several connectors, each connector is used by the MicroEJ Core Engine itself or by a foundation library.

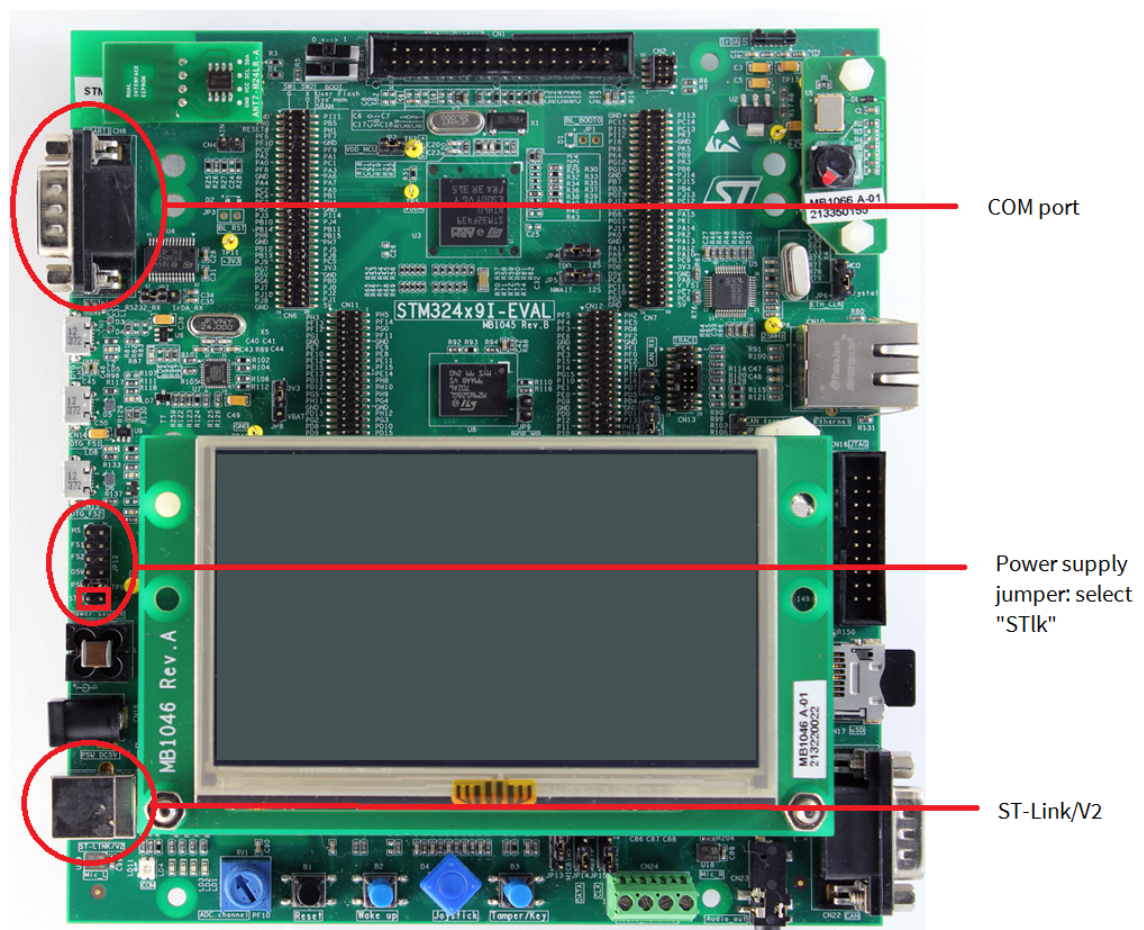
5.1. Mandatory Connectors

STM32F429I-EVAL provides a multi function USB port used as:

- Power supply connector
- Probe connector

First of all, ensure the *Power Supply jumper* is fit to the second option: *5V Link*. Then just plug an USB cable to a computer to power on the board and be able to program an application on it. Then plug a serial cable on CN8 connector (DB9 connector) to see the MicroEJ standalone application `System.out.print` traces.

Figure 5.1. Mandatory Connectors

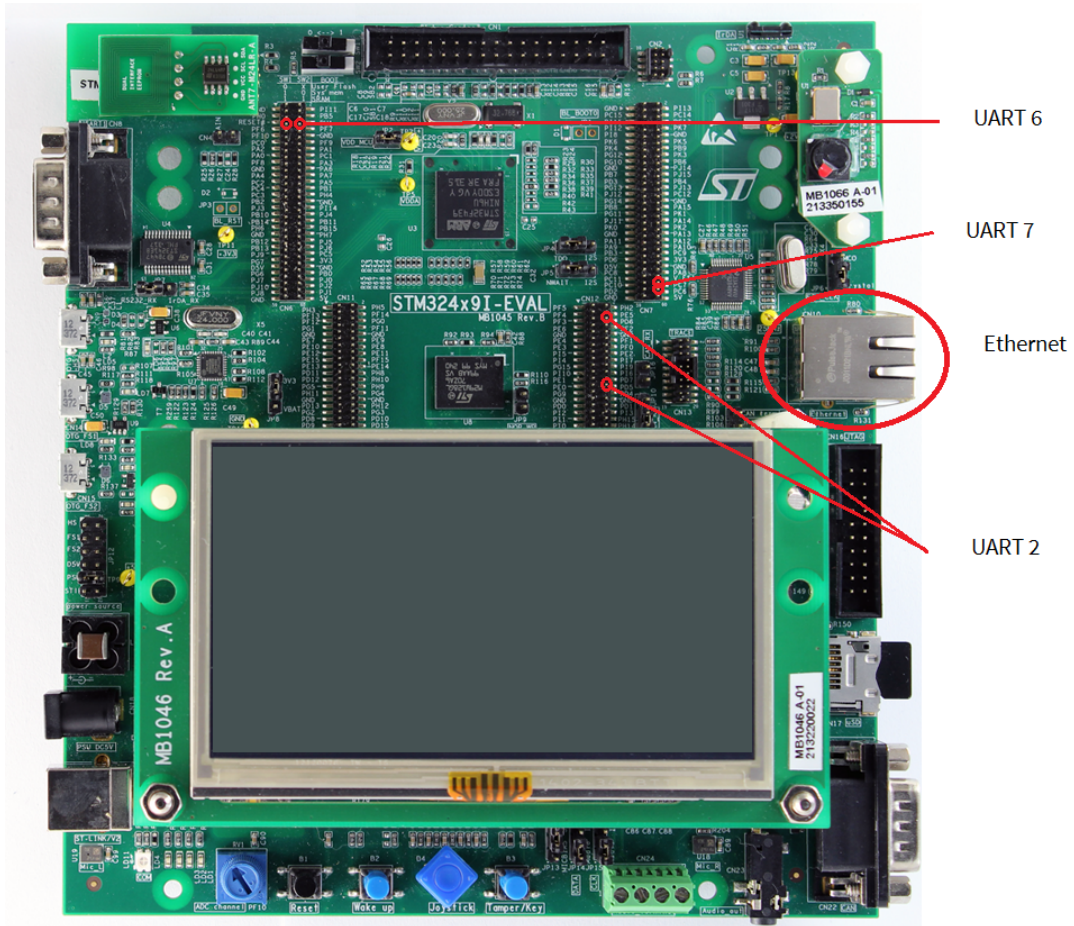


5.2. Communication Connectors

STM32F429I-EVAL provides several communication ports:

- Ethernet
- Serial communication
- CDC communication

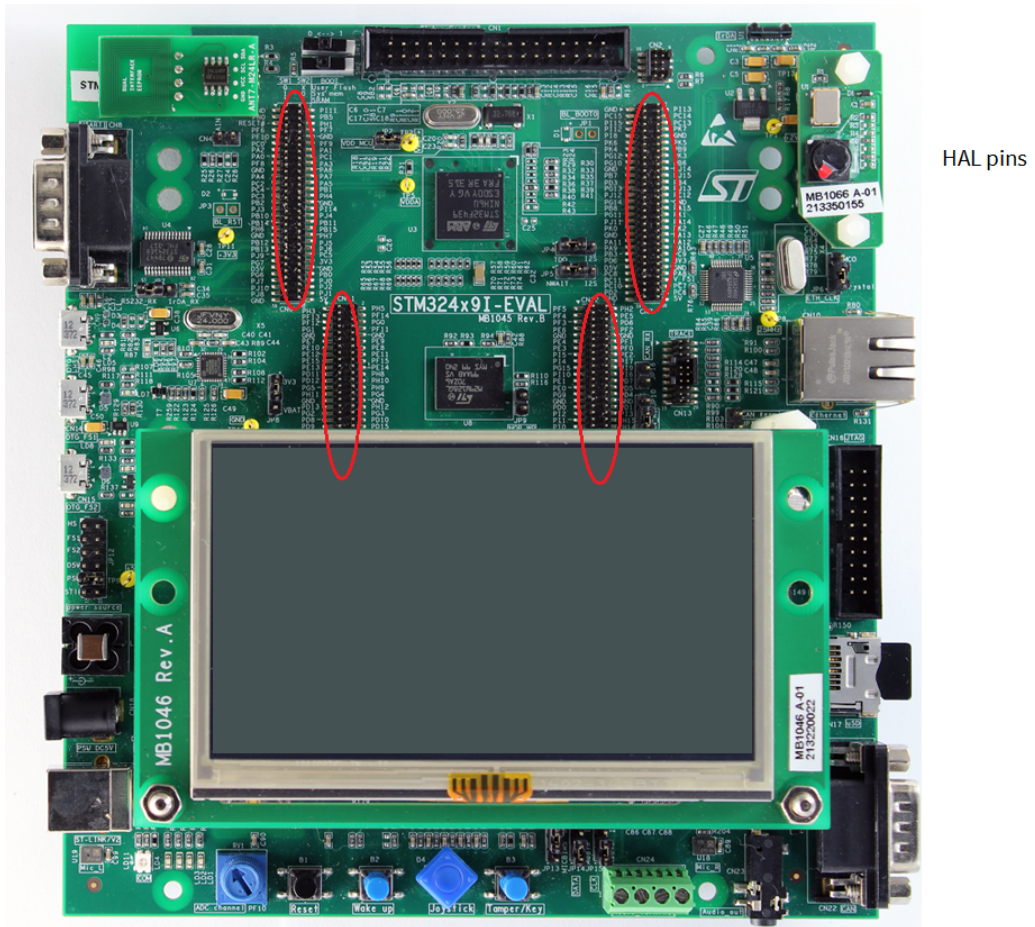
Figure 5.2. Communication Connectors



5.3. HAL Connectors

STM32F429I-EVAL provides several HAL GPIOs on connector CN6, CN7, CN11 and CN12.

Figure 5.3. HAL Connectors



Chapter 6. Changelog

6.1. Version 2.2.0

Initial release of the platform.