STM32Java

Build Your First Java Platform On STM32429I-EVAL

Getting Started



Reference:TLT-0691-MAN-PlatformGettingStarted-STM32429I-EVALRevision:BArchitecture:STM32JavaF4Compiler:Keil uVisionBoard:STM32429I-EVAL

Confidentiality & Intellectual Property

All right reserved. Information, technical data and tutorials contained in this document are confidential, secret and IS2T S.A. Proprietary under Copyright Law. Without any 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* including but not limited to electronic communication, photocopies, mechanical reproduction systems. Granted authorizations for using parts of the document or the entire document do not mean they give public full access rights.

IceTea®, IS2T®, MicroJvm®, MicroEJ®, S3[™], SNI[™], SOAR®, Drag Emb'Drop[™], IceOS®, Shielded Plug[™] and all associated logos are trademarks or registered trademarks of IS2T S.A. in France, Europe, United States or others Countries.

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

JavaTM, 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.

Table of Contents

1. Introduction	. 5
1.1. Intended Audience	5
1.2. Scope	5
1.3. Prerequisites	. 5
1.4. Terminology	. 5
2. Product Registration	. 7
3. Overview	. 8
3.1. Platforms	. 8
3.2. STM32JavaF4 ARMCCv4	. 8
4. Create and Use Your First Java Platform	. 9
5. Run an Example on the Simulated Java Platform (SimJPF)	11
5.1. Create Example	11
5.2. Run Example	12
6. Run the Example on the STM32429I-EVAL Board (EmbJPF)	13
6.1. Compile Java Application	13
6.2. Link and Deploy Java Application	13
7. Appendix	15
7.1. STM32429I-EVAL Technical Specifications	15
7.2. Board Setup	15
7.3. Keil µVision	16
7.4. Demos Installation	17
8. Document History	18

List of Figures

3.1. Available Platforms 8 4.1. Java Platform Configuration (page 1) 9 4.2. Java Platform Configuration (page 2) 10 4.3. Platform Configuration File 10 5.1. New STM32Java Example Project (page 1) 11 5.2. New STM32Java Example Project (page 2) 12 5.3. MVC Example on the Simulator 12 6.1. Keil µVision Project File 13 6.2. C IDE 14 7.1. Serial Cable 15 7.2. Board configuration 16 7.3. ULINK Probe Configuration 17	1.1. Platforms Flow2.1. STM32Java Welcome Page	6 7
4.1. Java Platform Configuration (page 1)94.2. Java Platform Configuration (page 2)104.3. Platform Configuration File105.1. New STM32Java Example Project (page 1)115.2. New STM32Java Example Project (page 2)125.3. MVC Example on the Simulator126.1. Keil µVision Project File136.2. C IDE147.1. Serial Cable157.2. Board configuration167.3. ULINK Probe Configuration17	3.1. Available Platforms	8
4.2. Java Platform Configuration (page 2)104.3. Platform Configuration File105.1. New STM32Java Example Project (page 1)115.2. New STM32Java Example Project (page 2)125.3. MVC Example on the Simulator126.1. Keil µVision Project File136.2. C IDE147.1. Serial Cable157.2. Board configuration167.3. ULINK Probe Configuration17	4.1. Java Platform Configuration (page 1)	9
4.3. Platform Configuration File105.1. New STM32Java Example Project (page 1)115.2. New STM32Java Example Project (page 2)125.3. MVC Example on the Simulator126.1. Keil µVision Project File136.2. C IDE147.1. Serial Cable157.2. Board configuration167.3. ULINK Probe Configuration17	4.2. Java Platform Configuration (page 2)	10
5.1. New STM32Java Example Project (page 1)115.2. New STM32Java Example Project (page 2)125.3. MVC Example on the Simulator126.1. Keil μVision Project File136.2. C IDE147.1. Serial Cable157.2. Board configuration167.3. ULINK Probe Configuration17	4.3. Platform Configuration File	10
5.2. New STM32Java Example Project (page 2) 12 5.3. MVC Example on the Simulator 12 6.1. Keil μVision Project File 12 6.2. C IDE 14 7.1. Serial Cable 15 7.2. Board configuration 16 7.3. ULINK Probe Configuration 17	5.1. New STM32Java Example Project (page 1)	11
5.3. MVC Example on the Simulator 12 6.1. Keil μVision Project File 13 6.2. C IDE 14 7.1. Serial Cable 15 7.2. Board configuration 16 7.3. ULINK Probe Configuration 17	5.2. New STM32Java Example Project (page 2)	12
6.1. Keil μVision Project File136.2. C IDE147.1. Serial Cable157.2. Board configuration167.3. ULINK Probe Configuration17	5.3. MVC Example on the Simulator	12
6.2. C IDE 14 7.1. Serial Cable 15 7.2. Board configuration 16 7.3. ULINK Probe Configuration 17	6.1. Keil µVision Project File	13
7.1. Serial Cable 15 7.2. Board configuration 16 7.3. ULINK Probe Configuration 17	6.2. C IDE	14
7.2. Board configuration	7.1. Serial Cable	15
7.3 ULINK Probe Configuration 17	7.2. Board configuration	16
is children in the comparation in the second s	7.3. ULINK Probe Configuration	17

List of Tables

7.1. STM32429I-EVAL Technical Specifications	1!	5
7.1. OTHOL 201 LITTL Technical opecifications	т,	

1 Introduction

1.1 Intended Audience

The intended audience for this document are developers who wish to develop their first Java application with STM32Java. Notes:

- This document is for STM32429I-EVAL 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 http://www.stm32java.com for more information about STM32Java products (platforms, videos, examples, application notes, etc.).

1.2 Scope

This document describes, step by step, how to start your development with STM32Java:

- Register your product.
- Understand the package.
- Create your own platform.
- Run a Java application on the simulation platform (SimJPF) and on the embedded platform (EmbJPF).

1.3 Prerequisites

- PC with Windows XP or later.
- The STM32Java environment must be installed.
- STM32429I-EVAL board.
- The ST-LINK utility (normally installed as part of the STM32Java installation).
- Keil μVision 4.60 or higher. The Keil μVision evaluation version is 32Kb code size limited. To get a Keil μVision evaluation license for STM32Java, please visit the website https://is2t.microej.com. Keil μVision 4.5x is sufficient except for deploying your application on the board using a ST-LINK/ V2. Refer to the appendix to use a ULINK probe instead of using the integrated ST-LINK/V2.

1.4 Terminology

A Java platform (JPF) is a library that, at a minimum, embeds a JVM (the MicroJvm® virtual machine from IS2T), some native libraries (also called firmware, or drivers, or BSP), and some Java libraries that characterize an application domain. Depending on these libraries, some specific tools are provided too.

Every embedded JPF (EmbJPF) has a sibling platform that simulates it on a PC, called the SimJPF. The SimJPF exactly mimics the semantics of the JPF. The SimJPF can be extended with Mocks to simulate the native (C, Ada, asm, ...) code of the JPF, such as drivers, or to connect hardware to the workstation.

Mocks can feed the SimJPF with real external stimuli, so that the Java application can behave as if it was running on the EmbJPF. SimJPF allows the design and functional testing of an embedded Java application in a development environment.



Figure 1.1. Platforms Flow

2 Product Registration

The development version of STM32Java uses hardware activation keys.

- Open STM32Java.
- Open welcome page: Help > Welcome.



Figure 2.1. STM32Java Welcome Page

- Insert your USB dongle.
- Press the Manage Licenses button. Hardware dongles are automatically detected when the STM32Java preferences main page is shown.
- Click on Refresh when a new hardware dongle is plugged in.



Note

The STM32Java preferences page is also accessible from the menu Window > Preferences > STM32Java.

3 Overview

3.1 Platforms

Several Java platforms (JPF) are pre-installed in STM32Java.

- Open STM32Java.
- Open the welcome page: Help > Welcome.
- Press the Manage Platforms button.

E Preferences					_ D X
type filter text		Available Platforms			⇔ ▼ ⇔ ▼ ▼
General Ant C/C++	^	Add or remove platforms. Platforms and Packs:			
Cache		Name	Version	License	Select All
Help Install/Update		Platform	x.x.x	~	Deselect All
Java	=				Import
MicroEJ Available Platforms				l	Uninstall
Naming Convention					Get UID
Plug-in Development Run/Debug					
Team Validation	-			Restore Defaults	Apply
0			[OK	Cancel

Figure 3.1. Available Platforms

This Available Platforms page shows all platforms installed in STM32Java. Each platform has its own characteristics and specific behavior.



Note

- The Available Platforms page is automatically populated by the work-in-progress JPFs.
- The Available Platforms page is also accessible from the menu Window > Preferences > STM32Java > Available Platforms.

3.2 STM32JavaF4 ARMCCv4

This Java platform architecture has been designed to be extended to run on any board with a COR-TEX-M4-based CPU. The Section 4, "Create and Use Your First Java Platform" uses this platform to run your first Java application on your first extended platform.

This platform requires the Keil μ Vision C compiler to create the final binary file for a Java application.

4 Create and Use Your First Java Platform

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

Although it is possible to use STM32Java to create every aspect of a platform in accordance with specific requirements, in this chapter we will use a pre-packaged example of a platform that is already configured for the STM32429I-EVAL.

- Open STM32Java.
- Open the Java Platform Example wizard: File > New > Java Platform.
- Select the Java platform architectue STM32JavaF4 ARMCCv4 from the combo box. A list of examples is available:

Target				
Extensible JPF:	Platform			
Create a platfo	orm from an exampl	e or a template.		
⊿ 🗁 Platforr	n			
a 📄 Boa	ard			
6	Basic			
E	Basic-SNI and SP			
	Full			
🔺 🛋 Ten	nplate			
i 🗁 -	Java Platform Templ	late		
		ine. CLDC/EDC. B-(ON, SNI, SP, NLS, E	COM-
Includes Micro	EJ Java Virtual Mach			
Includes Micro COMM and en	EJ Java Virtual Mach bedded UI extension	n (MicroUl, MWT).		
Includes Micro COMM and en Comes with a l	EJ Java Virtual Mach nbedded UI extension 3SP example for the l	n (MicroUl, MWT). board.		
Includes Micro COMM and en Comes with a I	EJ Java Virtual Mach nbedded UI extension SSP example for the l	n (MicroUl, MWT). board.		

Figure 4.1. Java Platform Configuration (page 1)

- Select the example configuration Full in menu STM32429I-EVAL.
- Click on Next. Give a name which be used as prefix for all platform projects. For instance: MyPlat-form.

reate a Java Pla	tform		
Create a new Java Plat	form.		
Project name: MyPl	attorm		
✔ Use <u>d</u> efault locat	on		
Location: C:\Users\			Browse
IPF Properties			
Device":			
Name*:	MyPlatform		
/ersion*:	1.0.0		
Provider*:	MyCompany		
/endor URL:			
MicroEJ Min Version:	1.7.0		

Figure 4.2. Java Platform Configuration (page 2)

- Click on Finish. The selected example is imported as several projects prefixed by the given name:
 - [example]-bsp contains a ready-to-use package for Keil µVision on STM32429I-EVAL board.
 - [example]-configuration contains some files to configure the platform automatically.
 - [example]-microuiGeneratorExtension contains some files to extend the MicroUI Image Generator tool.

A README file and a platform configuration file are automatically opened. The README file explains the aim of the projects and how to launch an example. We recommend you follow the steps described in this document, and refer to the README file for more detail and latest changes.

The platform needs to be built with several additional modules (Java and natives libraries, scripts etc.).

• From the platform configuration file, click on the link Build Platform

Java Platform Properties		Platform Content
General informa	tion about this Java platform.	The content of the platform is composed of two parts:
Device:	MyBoard	Environment: select the Java platform environment architecture and version.
Name:	MyPlatform	Modules: select modules to import in the Java platform.
Version:	1.0.0	
Provider :	MyCompany	Platform Configuration
/andor URL:		Once the content of the platform is chosen, it can be configured.
		2 Configuration
MicroEJ Min Ver	sion: 1.7.0	Each module can be configured creating a folder with its name along the .platform file. It could contain:
		 an optional [module].properties file,
		 optional module specific files and folders.
		Modifying one these files requires to build the platform again.
		Build
		Generate and test the Java platform.
		Build Platform: The new platform is now available and visible in Available Platforms

Figure 4.3. Platform Configuration File

The build starts. This step can take several minutes. You can see the progress of the build steps in the Eclipse console. Please wait for the final message BUILD SUCCESSFUL.

At the end of the execution the platform is fully built for the STM32429I-EVAL board and is ready to be linked into the Keil μ Vision project. The name of this platform is STM32429IEVAL-MyPlatform-CM4_ARMCC.

The platform is now ready for use and available in the Platforms list of your STM32Java repository (Windows > Preferences > STM32Java)

5 Run an Example on the Simulated Java Platform (SimJPF)

The aim of this chapter is to create a simple Java application from a built-in example. This example will initially be run on the simulator (SimJPF) of the platform created in the previous chapter. Then, in the next chapter, this application will be compiled and deployed on the STM32429I-EVAL board using the EmbJPF.

5.1 Create Example

- Open STM32Java.
- Open the welcome page: Help > Welcome.
- Press the Try out Java Examples button to open the New Java Example Project wizard.
- Select the Java platform STM32429IEVAL-MyPlatform-CM4_ARMCC from the combo box.
- Select the example Examples > MicroUI > MVC.

elect an Example	
Target	
JPF: Platform	¥
▲ ☐ Samples	^
▶ ➡ BON	
ECOM-COMM	
EDC	
Getting Started	
MWT	
MicroUI	
B Check Input Events	
🛱 Filters	
🛱 Fonts	
🗁 Hello World	
🗁 Images	
🔁 LEDs	
🔁 Layers	
B MVC	
	~
This example shows how to create and use a MVC design pattern	^
	v
	Const

Figure 5.1. New STM32Java Example Project (page 1)

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

	New MicroEJ Example Project	>
Create a MicroE	J Project	
Enter a project nam	ne.	
Project Name: M	MVCSample	

Figure 5.2. New STM32Java Example Project (page 2)

• Click on Finish. The selected example is imported into a project with the given name. The main class (the Java class which contains the main() method, here: MVCDemo.java) is automatically opened.



Note

Note: The New Java Example Project wizard is also accessible from the menu File > New > Java Example.

5.2 Run Example

- Open the run dialog (Run > Run configurations...) and select the STM32Java launcher MyMVCSample SimJPF.
- Click on Run: the application starts. It is executed on the simulator (SimJPF) of the selected JPF (STM32429IEVAL-MyPlatform-CM4_ARMCC). A picture of the board appears. The result of the execution is shown on the display of the board. The touchscreen can be used to adjust the division of the screen areas.



Figure 5.3. MVC Example on the Simulator

6 Run the Example on the STM32429I-EVAL Board (EmbJPF)

6.1 Compile Java Application

- Open the run dialog (Run > Run configurations...) and select the MicroEJ launcher MyMVCSample EmbJPF.
- Click Run: the application is compiled, and the compilation result (an ELF file) is copied into a wellknown location in the workspace. The Keil µVision BSP project will look for it there when it performs the final link.

6.2 Link and Deploy Java Application

The aim of the final step is to:

- Compile the BSP project (such as drivers).
- Link the BSP and the others libraries (MicroJvm Virtual Machine, native stacks, Java application).
- Deploy the full application on the STM32429I-EVAL board.



Note

This final step uses Keil µVision.

• In STM32Java, expand the project [example]-bsp and the folder Project/MicroEJ/MDK-ARM. A Keil µVision project file (Project.uvproj) is available.



Figure 6.1. Keil µVision Project File

• Double-click on this file to open Keil µVision.

The remaining steps are performed within Keil μ Vision.



Figure 6.2. C IDE

- Build the Keil $\mu Vision$ project by clicking on the menu Project > Build target. The project is compiled and linked.
- Deploy the link result on the STM32429I-EVAL board by clicking on the menu Flash > Download.

The application starts. The result of the execution is shown on the display of the board.

Congratulations, you have built the Java application on your own platform!

7 Appendix

7.1 STM32429I-EVAL Technical Specifications

The following table illustrates some board features used by the Java platform. It doesn't list all board features (such as all available memories on the board).

MCU architecture	Cortex-M4 (STM32F429NIH6)
MCU Clock speed	180MHz (225 DMIPS)
Internal Flash	2 MBytes
Internal RAM	128 KBytes
External RAM	8 MBytes (SDRAM)

Table 7.1. STM32429I-EVAL Technical Specifications

7.2 Board Setup

This section explains how to configure and connect the STM32429I-EVAL board .

7.2.1 Jumpers and Switches

Set all jumpers and switches to their default settings: please refer to the STM32429I-EVAL documentation (STMicroelectronics documentation: User manual STM32429I-EVAL evaluation board).

7.2.2 Power Supply

Select your power supply mode: please refer to the STM32429I-EVAL documentation (STMicroelectronics documentation: UM1668 User manual STM32429I-EVAL evaluation board).

7.2.3 PC-Board Connection

Plug a serial cable between the PC and the board USART 1 connector (CN8). Note the PC's COM port, it will be used later in the application launcher. This allows all traces to be received from the board and printed on the console.

Use a simple serial cable with only data lines (crossed) and signal ground. See the illustration below :



Figure 7.1. Serial Cable

7.2.4 Summary



Figure 7.2. Board configuration

7.3 Keil µVision

7.3.1 Project Compatibility

The μ Vision projects available as Java platform examples are compatible Keil μ Vision [4.60, 5.0[. Several options have to be updated manually to be compatible with Keil μ Vision 5.x (such as the choice of the microprocessor).

7.3.2 Probe Configuration

This chapter explains how to use a ULINK probe instead of using the ST-LINK/V2. This step is useful if you use Keil μ Vision 4.5.x, as older versions of Keil μ Vision do not manage the ST-LINK/V2 probe well.

- Open a Keil µVision project.
- Open the menu Project > Options for Target '(your target)'.
- Open the tab Utilities.
- Select the item ULINK2/ME Cortex Debugger in the combo box of the menu Use Target Driver for Flash Programming.

Configure Flash	Menu Command
 Use Targe 	Driver for Flash Programming
Init File:	ULINAZINE Cortex Debugger Setungs PC Optate Farger Delote Debugging
C	
O Use Extern	al Tool for Flash Programming
Command:	
Arguments:	
	🗖 Run Independent

Figure 7.3. ULINK Probe Configuration

• Click on OK.

7.4 Demos Installation

This chapter explains how to install extra demos and application notes. They have to be imported as a standard Eclipse project archive file:

- Go to the website http://www.stm32java.com/portal/?q=resources/demos.
- Download zip files containing one or more demos.
- Open STM32Java.
- Open the wizard Import: File > Import.
- Select the item General > Existing Projects into Workspace.
- Click on Next.
- Select the downloaded zip file.

8 Document History

Date	Revision	Description
November 20th 2013	А	First release
May 22th 2014	В	MicroEJ 3.0 compatibility