Selected Topics in Java

Java SE Embedded, Raspberry Pi

Martin Deinhofer, WS2018

Java SE Embedded, Recommended Development Setups

Development Setup – Only RPi

- **Program development and execution on RPi directly**

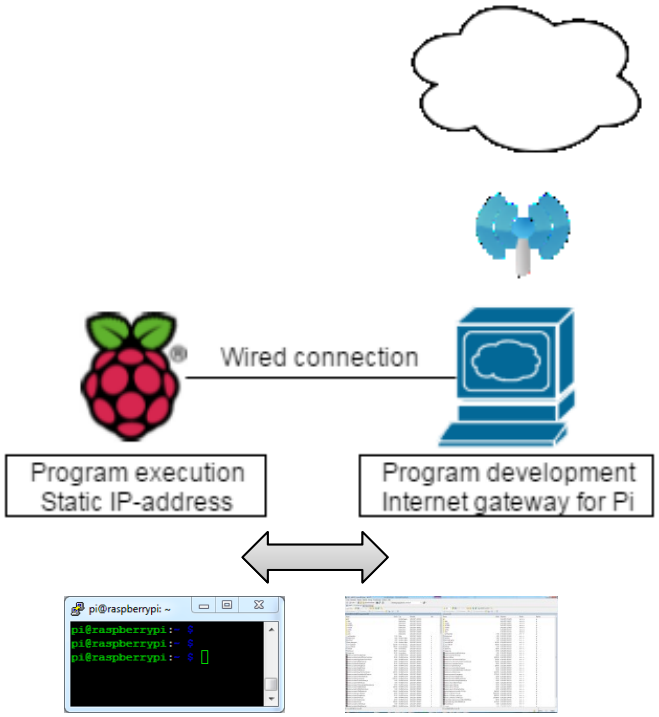
- Pros
  - Simple workflow (no file transfer,...)
  - Immediate testing
- Cons
  - Very slow (when using typical IDEs e.g. Eclipse)
  - → Use lightweight editors like geany and build scripts

Development Setup – RPi+Development PC

- **Program development on development PC**
- **Program (compilation) and execution on RPi**

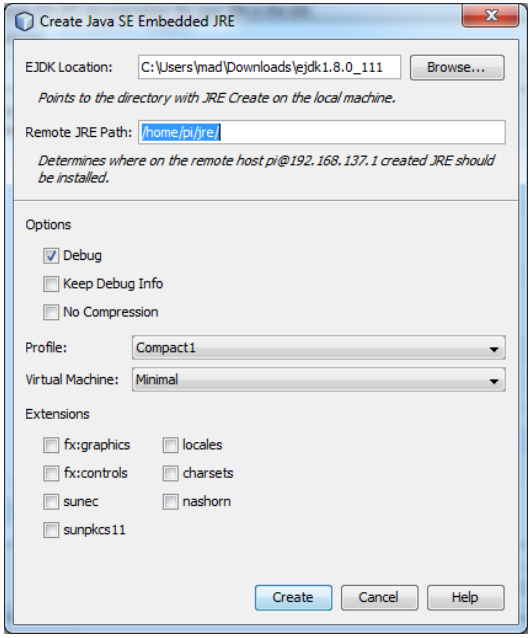
- Pros
  - Powerfeatures of IDE
  - fast
- Cons
  - Complex workflow (file transfer,...)
  - Several steps before testing

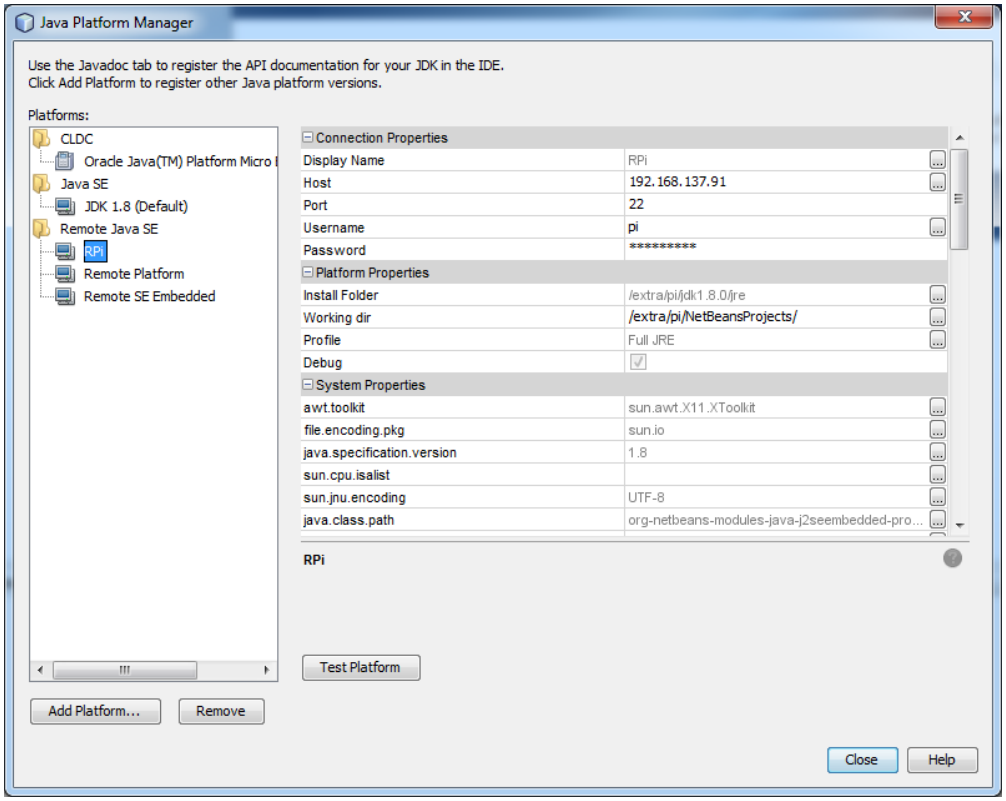
- Suggested Setup
  - Create project on Pi
  - Use remote file system share to edit files
  - Use remote shell to login to RPi
  - Use build script to compile and run program on RPi via remote shell



Workflow with NetBeans IDE

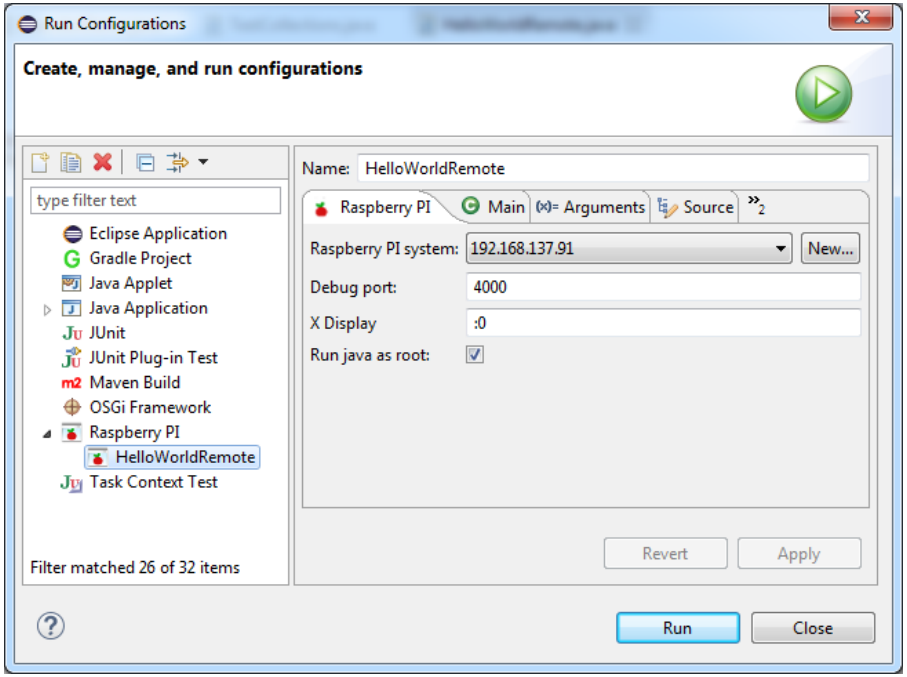
1. Create **Remote Java Runtime (SE or SE Embedded)**
2. Execute project on **Remote Java Runtime**





Workflow with Eclipse
(LaunchPi)

- Plugin LaunchPi: Uses scp, ssh to deploy and run program remotely



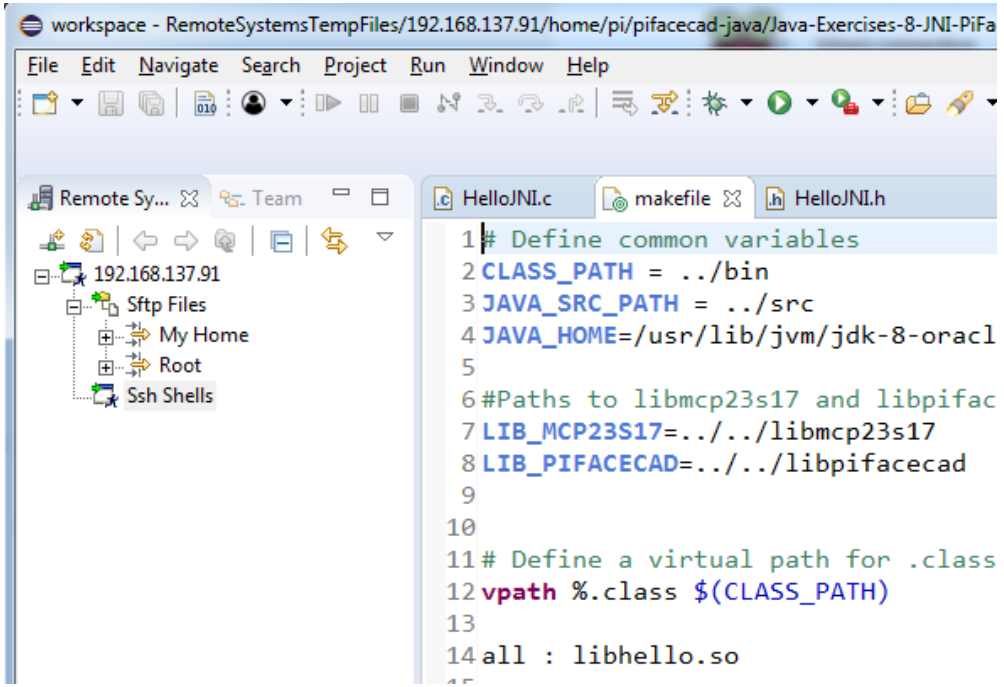
Workflow with Eclipse

(Remote System Explorer, SSH-Client)

- Edit files directly in built-in **Remote System Explorer** perspective
- Login to RPi with built-in SSH shell and execute build script.

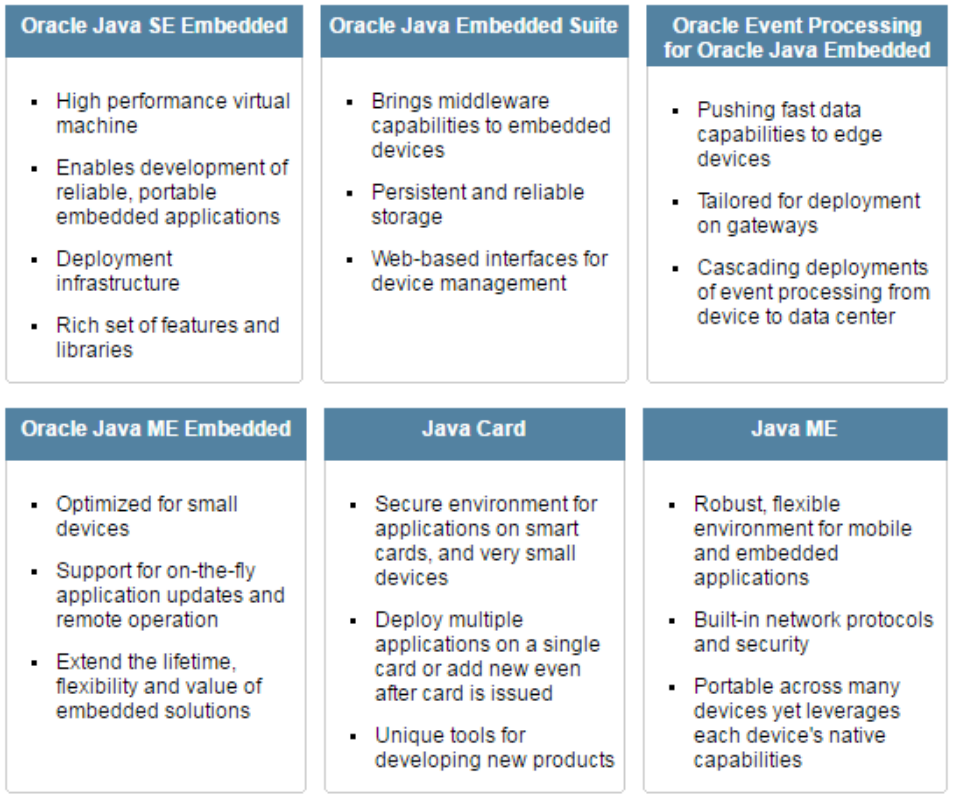
Alternatives

- SSH-Client: putty
- File transfer: WinSCP
- Remote file system: samba or nfs mounted as network share



Overview of Oracle Embedded technologies

- Oracle provides
  - Java SE and Java SE Embedded for ARM CPUs
  - **Java ME** for low foot print devices
  - **Java Card** for Smartcards



Java SE Embedded - Profiles

- Supports customization of Java VM
  - 3 profiles only containing a subset of Java API

Full SE API	Beans	Input Methods	IDL
	Preferences	Accessibility	Print Service
	RMI-IIOP	CORBA	Java 2D
	Sound	Swing	
	AWT	Drag and Drop	
	Image I/O	JAX-WS	
compact3	Security[1]	JMX	JNDI
	XML JAXP[2]	Management	Instrumentation
compact2	JDBC	RMI	XML JAXP
compact1	Core (`java.lang.*`)	Security	Serialization
	Networking	Ref Objects	Regular Expressions
	Date and Time	Input/Output	Collections
	Logging	Concurrency	Reflection
	JAR	ZIP	Versioning
	Internationalization	JNI	Override Mechanism
	Extension Mechanism	Scripting	

Java SE Embedded – Command line

You can create a custom Java VM with Netbeans or via the `jrecreate.sh` command line tool

1. Download and extract **Java SE Embedded** to your pi
2. Execute the tool `jrecreate.sh`

```
pi@raspberrypi:~/ejdk1.8.0_111/bin $ JAVA_HOME=/usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt/ ./jrecreate.sh -d my-jre-compact1 -p compact1
Building JRE using Options {
    ejdk-home: /home/pi/ejdk1.8.0_111
    dest: /home/pi/ejdk1.8.0_111/bin/my-jre-compact1
    target: linux_armv6_vfp_hflt
    vm: minimal
    runtime: compact1 profile
    debug: false
    keep-debug-info: false
    no-compression: false
    dry-run: false
    verbose: false
    extension: []
}


Target JRE Size is 10,813 KB (on disk usage may be greater).
Embedded JRE created successfully
pi@raspberrypi:~/ejdk1.8.0_111/bin $
```

Breakout – Suggested Exercises

- Setup Raspberry Pi with PiFace Control and Display (PiFaceCAD) add-on board, according to this tutorial.
- Use JNI to send the following two-line text to the add-on display from within Java:
  Java for Embedded rocks!!
- Use the C-libraries libmcp23s17 and libpifacecad to interface the PiFaceCAD.