STM32Java

Embedded Java Solutions for STM32

Agenda

Presentation

Speakers

STM32Java Overview

Embedded Java

Adding Java to your system
Focus on GUI Design

Q&A

Laurent Desseignes (ST)

Régis Latawiec (IS2T)

What is STM32Java ?

- STM32Java is a complete solution to develop and to deploy applications on STM32 microcontrollers using Java technologies

- The solution was developed by
  - STMicroelectronics - semiconductor company
  - IS2T - software editor (embedded Java solutions provider)

STM32Java Solution

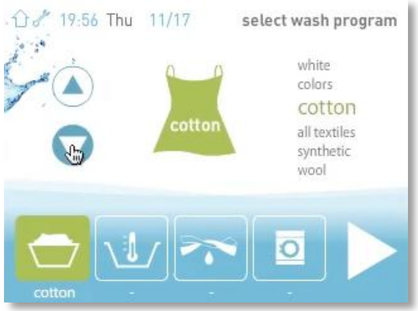
- Dedicated STM32 Java enabled Devices
  - « U » Part numbers for sampling (ex: STM32F407IGT6U)
  - « J » Part numbers for production (ex: STM32F407IGT6J)

- An embedded Java SDK
  - Based on MicroEJ® by IS2T
  - On PC simulator for easy prototyping
  - Java Virtual Machine with standard libraries & BSP
  - A set of libraries & tools for Embedded Graphics:
    - Widgets, Fonts, Images, …

Target Markets for Embedded Java

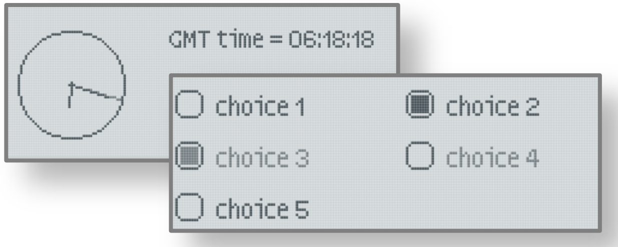
Consumer

- Kitchen equipment (Cookers, Ovens, Washing machines)



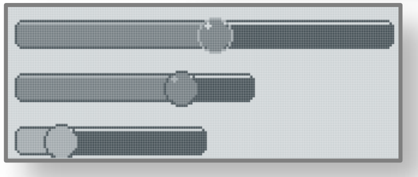
Home appliance

- Comfort (Heating)
- Security (Alarms, Access control)



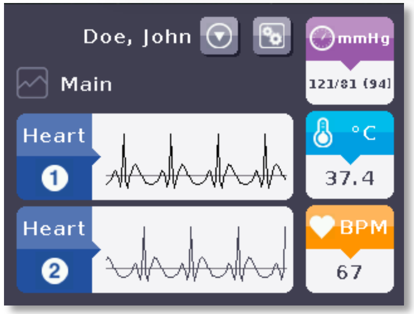
Industrial, Metering

- Automation, Smart-metering



Healthcare

- Body control equipment



Solved Business Issues

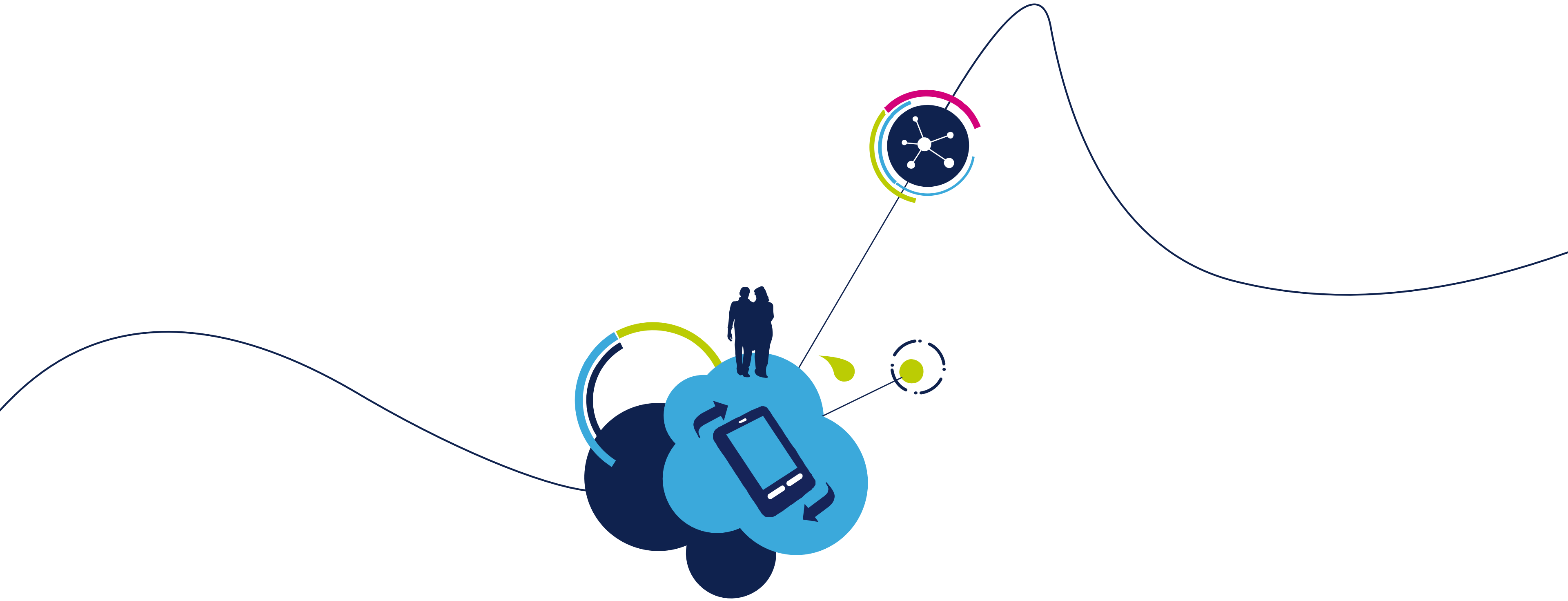
Marketing Directors

- Broaden product portfolio
  - Improved software scalability to derivate new products from older ones
- Shorten product definition phase
  - Use simulation capabilities to test prototypes, especially for GUI design
- Speed-up product first sales
  - Train sales force using simulators before products are ready

Solved Business Issues

R&D Directors

- Deliver on time
  - Increase software productivity
- Improve software engineering skills
  - Train your technical staff on Object Oriented Programming, access to “pure” software people.
- Maintain quality level
  - Improve testing capabilities



Embedded Java

Embedded Software Challenges

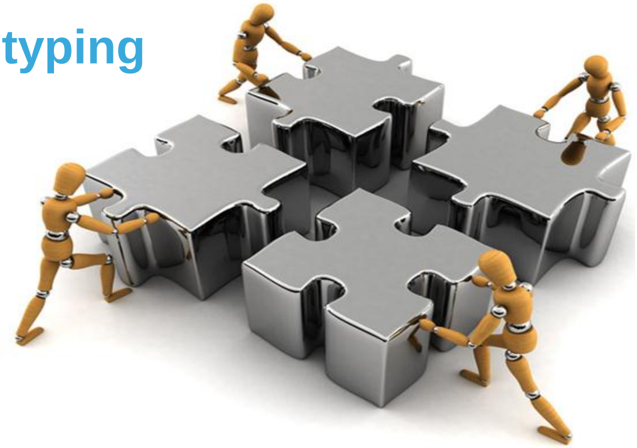
- Microcontrollers now offer more than 1M Bytes on-chip Flash
  - Embedded software complexity increases
  - Testing time increases
  - New topics complex to address such as Graphical User Interfaces

- Hardware & OS market fragmentation
  - How to deal with a different ranges of processors?
  - How to reuse existing embedded software, including RTOS?

Embedded Software Challenges

Need to industrialize embedded software production.

Object Oriented Programming & Virtual Platforms are the known best solutions since the 80’s.



Prototyping

Agility

Object Oriented Programming

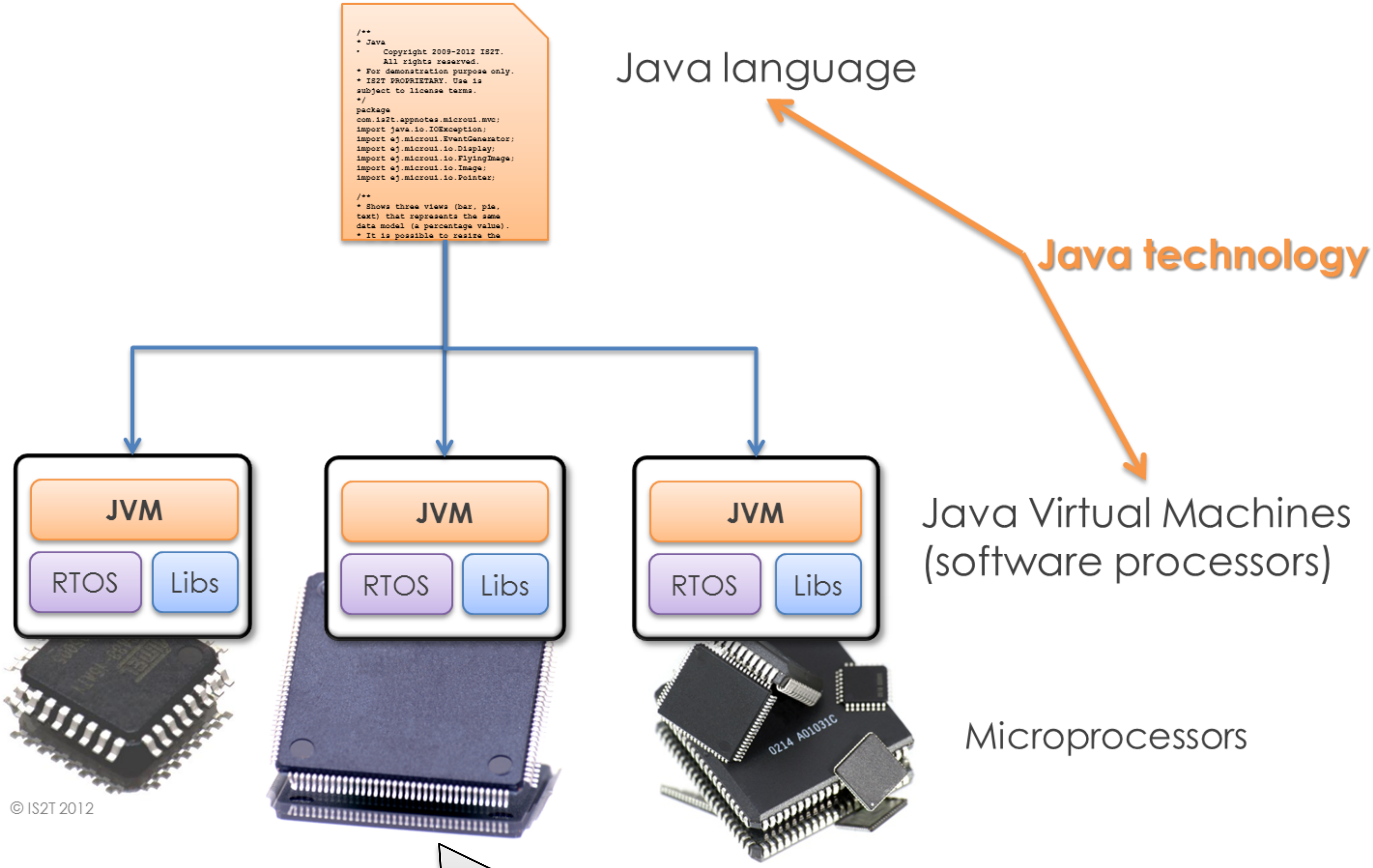
Java Virtual Platforms

Java Technology Key Benefits

- Use a modern programming technology
  - Object oriented to enable loose entity coupling and complex entity organizations (business code, widgets, servers)
  - Portable across a wide range of platforms including 3rd party HW like smart-phones
  - Automatic RAM management, ideal for event driven systems like GUIs

- Improve your software design process
  - Short learning curve
  - Better software productivity (x3 to x5 usually)
  - Large engineer community
  - Keep hardware & software design cycles independent

Embedded Java Platforms



```
/**
* Java
*     Copyright 2009-2012 IS2T.
      All rights reserved.
* For demonstration purpose only.
* IS2T PROPRIETARY. Use is
subject to license terms.
*/
package
com.is2t.appnotes.microui.mvc;
import java.io.IOException;
import ej.microui.EventGenerator;
import ej.microui.io.Display;
import ej.microui.io.FlyingImage;
import ej.microui.io.Image;
import ej.microui.io.Pointer;

/**
* Shows three views (bar, pie,
text) that represents the same
data model (a percentage value).
* It is possible to resize the
```

Java language

Java technology

Java Virtual Machines
(software processors)

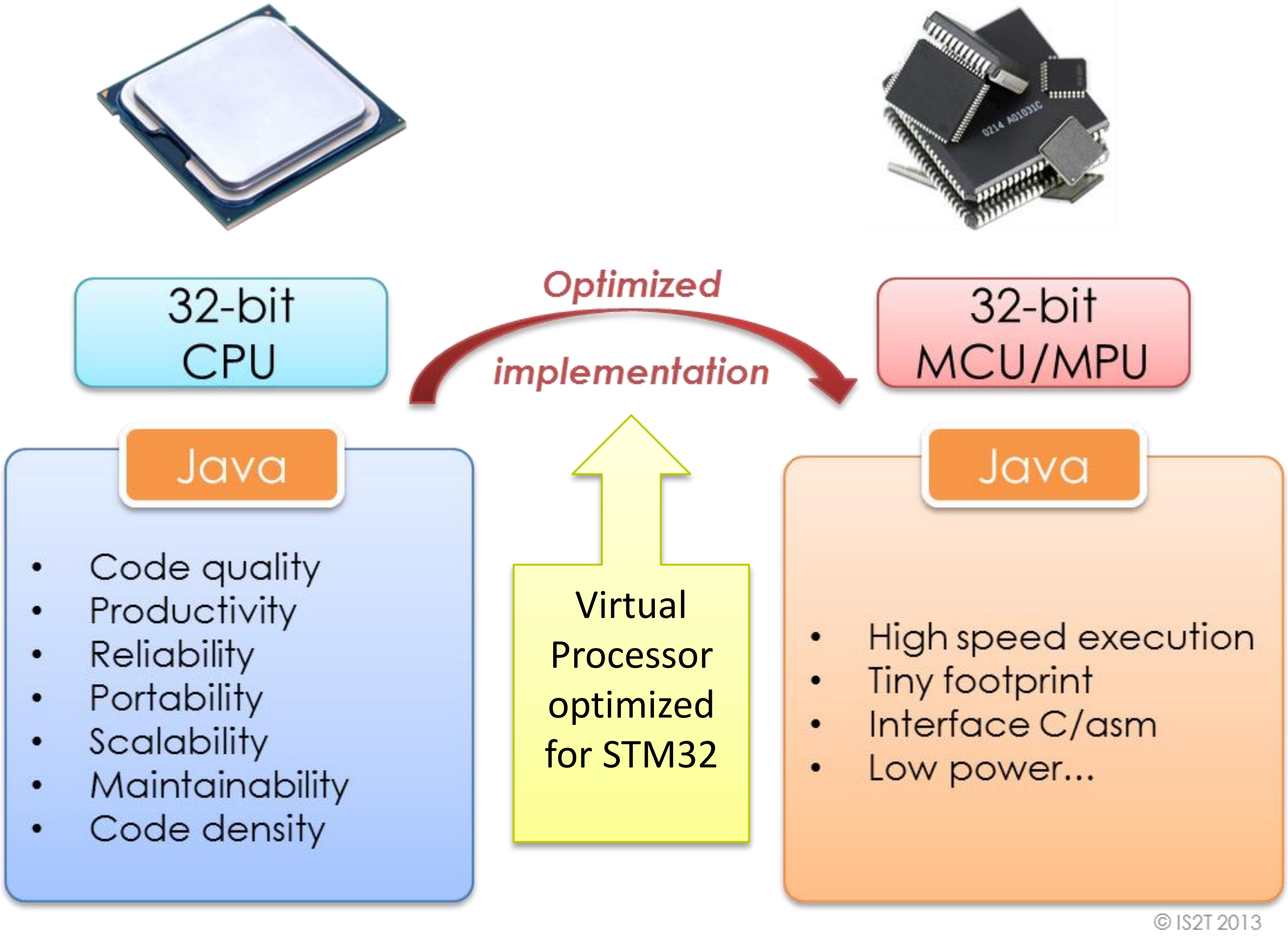
Microprocessors

Flexible JVM architectures to fit hardware constraints

A Virtual Machine: a Virtual Processor

- 32-bit instructions set (~200 instructions, high density)
- Multitasking operation
- Automatic memory optimizer
- Run-time error handling
- Interface to other languages like C

Optimized MicroJvm® VMs by IS2T



32-bit CPU

Optimized implementation

32-bit MCU/MPU

Java

- Code quality
- Productivity
- Reliability
- Portability
- Scalability
- Maintainability
- Code density

Virtual Processor optimized for STM32

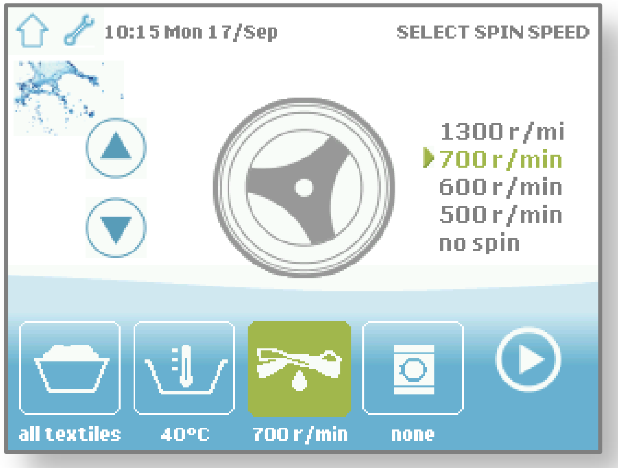
Java

- High speed execution
- Tiny footprint
- Interface C/asm
- Low power...

© IS2T 2013

Embedded Java Platform Example

- STM32F2x (Cortex-M3) – 120MHz
- 16-bit col. QVGA LCD, Touch
- APIs: CLDC, B-ON, MicroUI, MWT, SNI
- Boot time (reset to **main(String[] args)**): 2ms

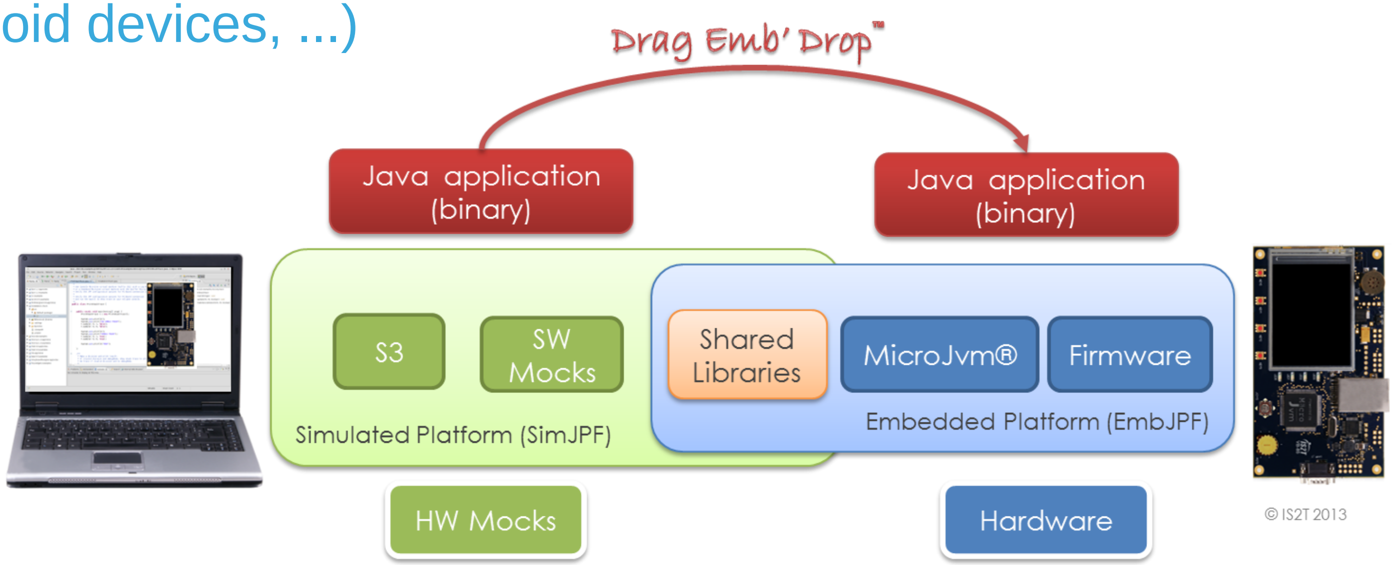


Application Memory Requirements

	Flash	422KB	RAM	42KB
Java needs	Virtual Machine (runtime & GC)	28KB	Virtual Machine	1KB
	Libraries (graphics, com, float…)	132KB	Native Stack	28KB
	Graphical resources (images)	228KB		
	Application	34KB	Application	13KB

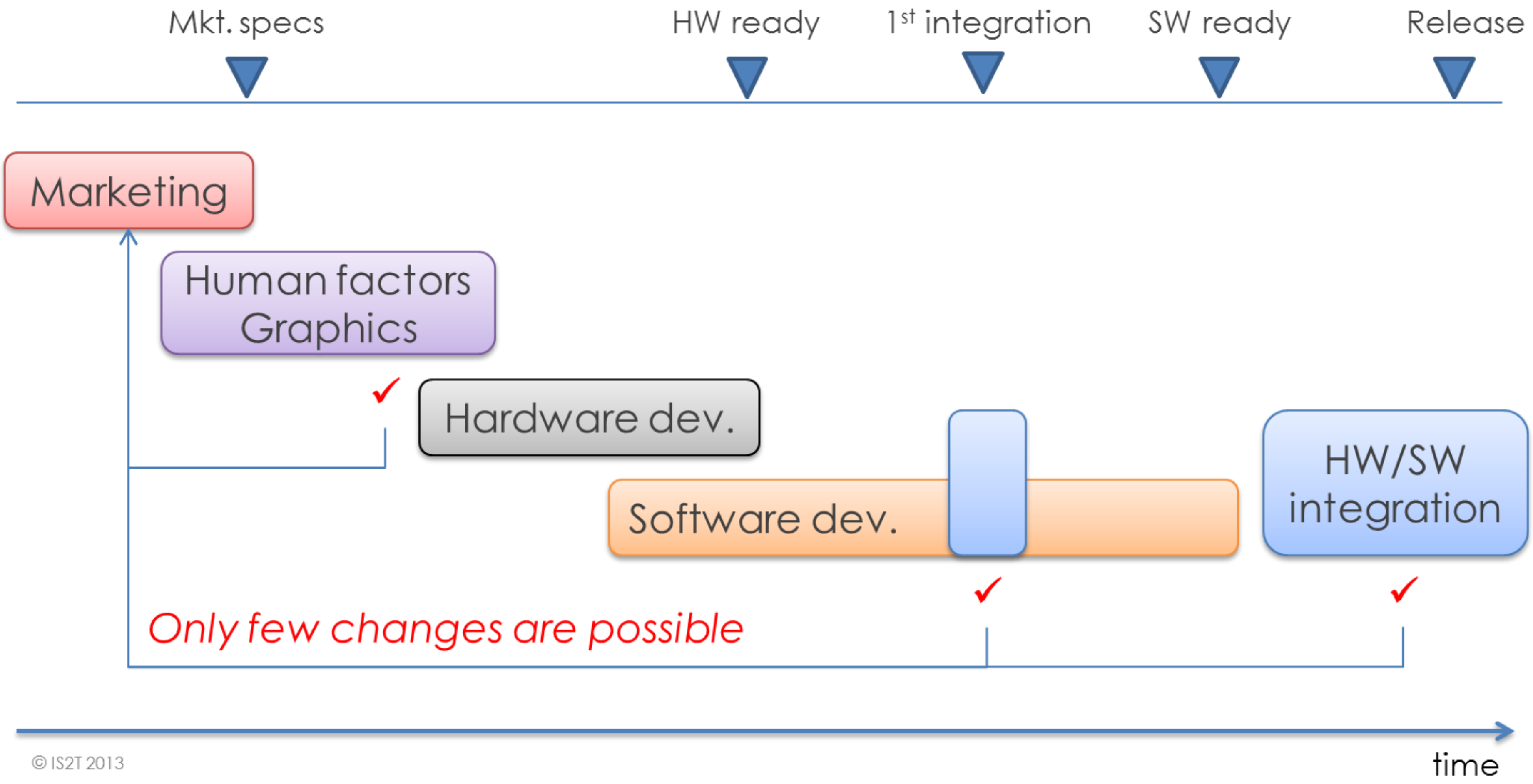
Prototyping Using Simulation

- Ubiquity
  - Same binary code can be executed by different Java platforms on various software and hardware platforms (PC, MCU/MPU, iOS & Android devices, ...)


Drag Emb' Drop™
Java application (binary)
Java application (binary)
S3
SW Mocks
Simulated Platform (SimJPF)
Shared Libraries
MicroJvm®
Firmware
Embedded Platform (EmbJPF)
HW Mocks
Hardware
© IS2T 2013

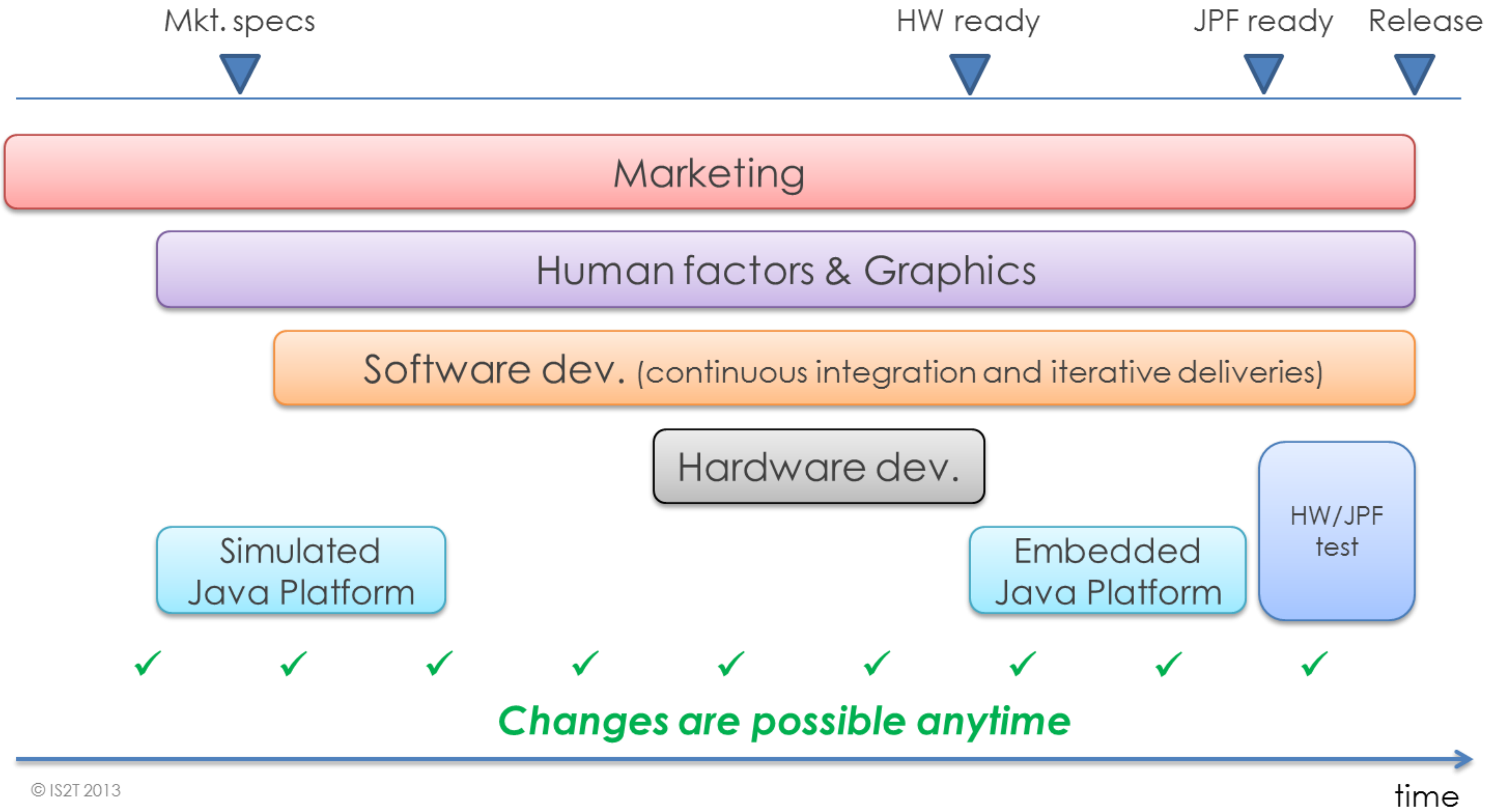
- Embedded / Simulated platforms
  - Java calls C/asm (firmware, drivers) on target
  - Native code simulated by mocks (SW/HW) on simulator

Replace Waterfall Design Process…



Only few changes are possible

Agility – Combine Simulation & OOP



Mkt. specs

HW ready

JPF ready

Release

Marketing

Human factors & Graphics

Software dev. (continuous integration and iterative deliveries)

Hardware dev.

Simulated Java Platform

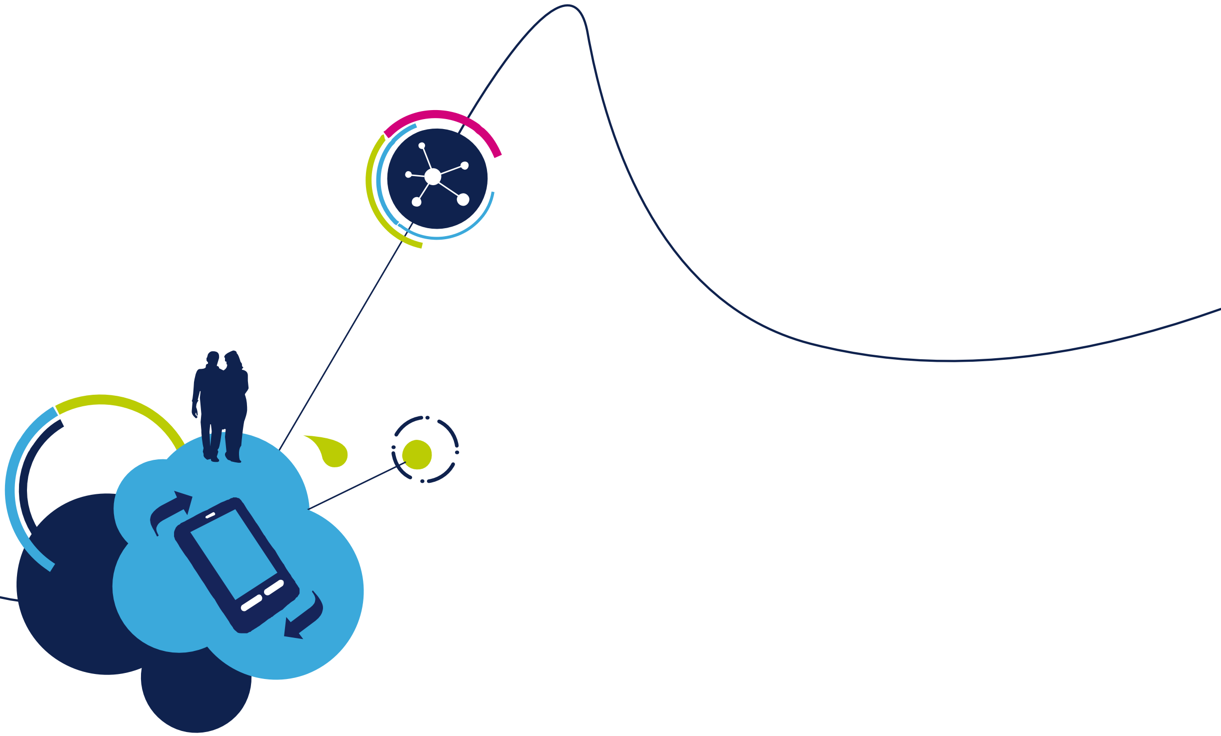
Embedded Java Platform

HW/JPF test

✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Changes are possible anytime

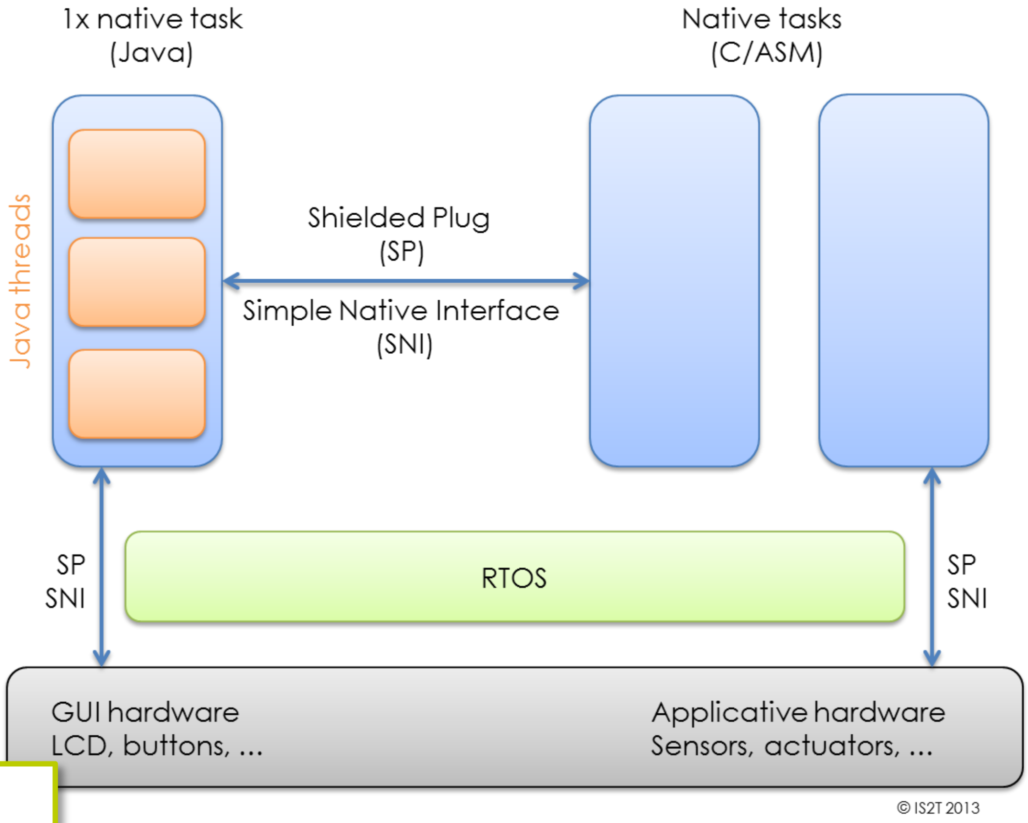
time



Adding Java to your system

Easy RTOS Integration (Green Thread)

- Multi-threaded Java environment within a single RTOS task
- Java thread scheduling policy independent of the RTOS
- Easy control of CPU resource usage for Java world
- CPU resource allocation irrespective of the number of threads



RTOS Examples

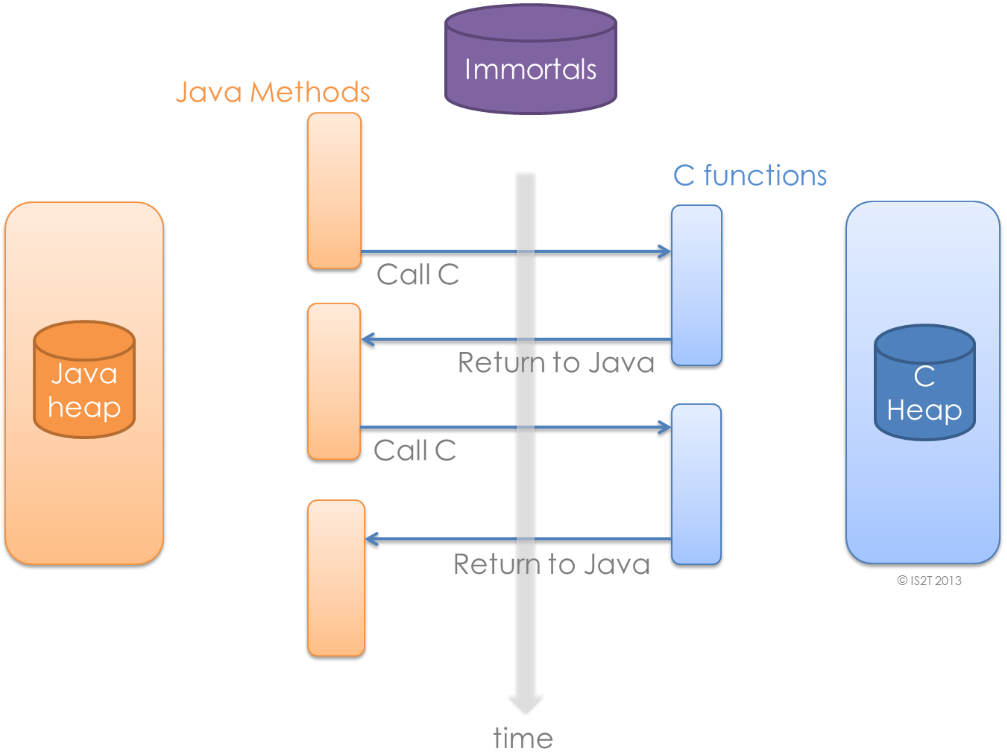
- FreeRTOS
- ThreadX, µC/OS, EmbOS
- Yours!

Easy Java ↔ C Interface (Calls 1/2)

- SNI (ESR 012) : Simple Native Interface
- Call Java world → C/asm
- Arguments: base types (int, float, double , char)

esr consortium

www.e-s-r.net



© IS2T 2013

life.augmented

Easy Java ↔ C Interface (Calls 2/2)

- Easy mapping using naming convention

```
package GPIO;

public class Main {
    public static native void toggle();

    public static void main(String[] a) throws InterruptedException
{
        while(true){
            toggle();
            Thread.sleep(10);
        }
    }
}
```

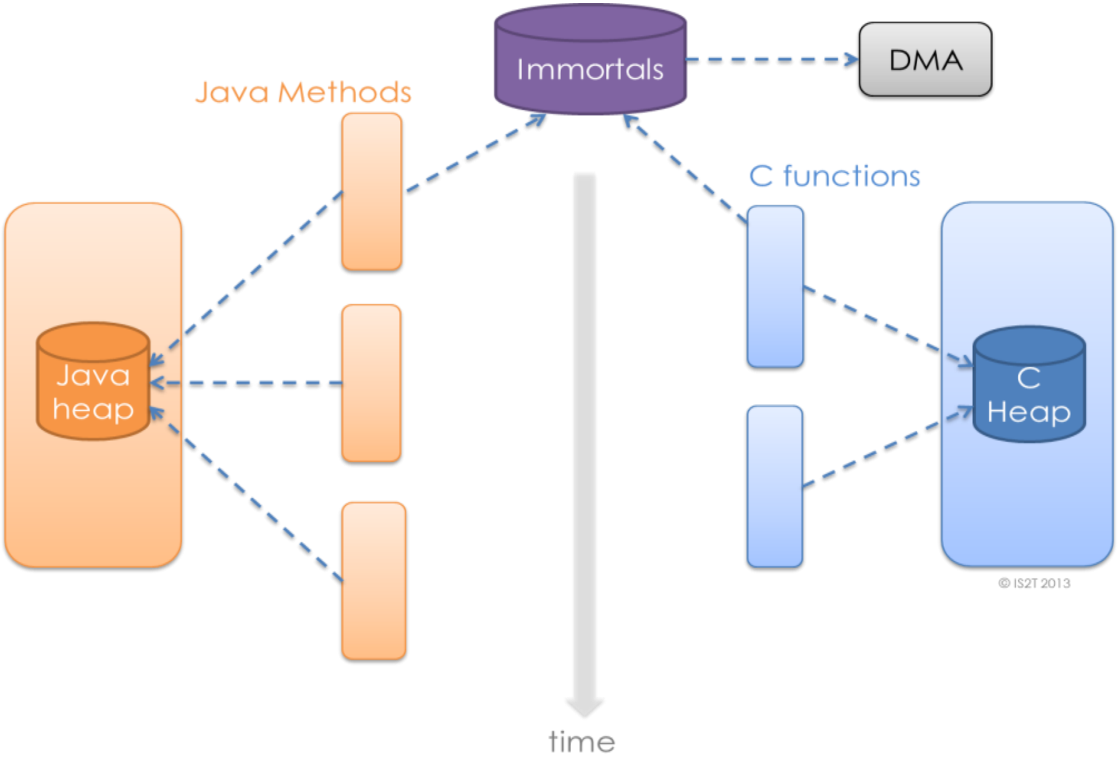
```
#include <sni.h>
#include “gpio.h”

void Java_GPIO_Main_toggle(){
    GPIOE->ODR ^= GPIO_Pin_2 ;
}
```

Easy Java ↔ C Interface (Data 1/2)

- SNI (ESR 012): Simple Native Interface
- Share arrays of base types
- Zero copy buffers and compatible with DMA systems

esr consortium

www.e-s-r.net



© IS2T 2013

life.augmented

Easy Java ↔ C Interface (Data 2/2)

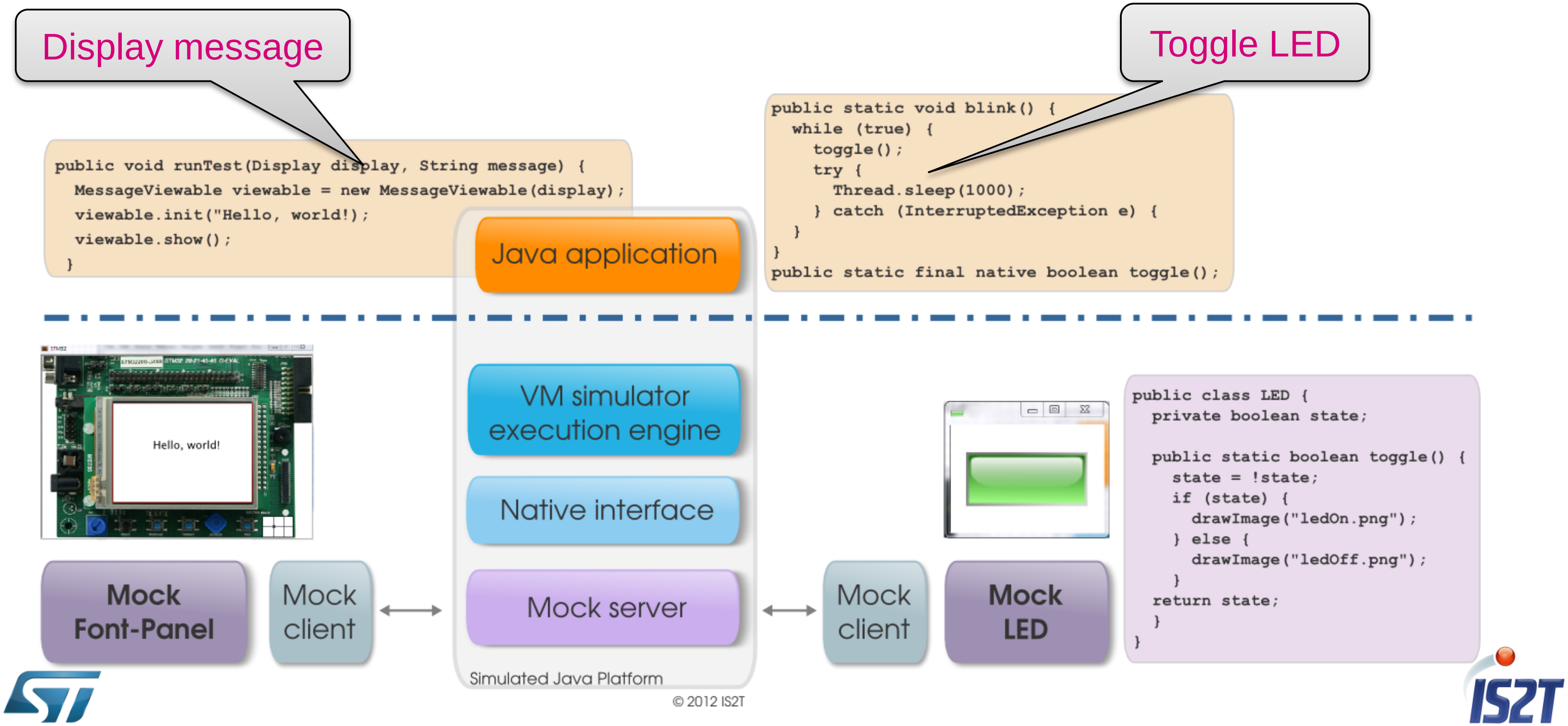
- Immortals are used to share data memory between Java and C

```
package com.corp.examples;
public class Hello {

   static int[] array = (int[])Immortals.setImmortal(new int[50]);
   public static native int getData(int[] array);

   public static void main(String[] args){
      int nb = getData(array);
   }
}
```

```
#include <sni.h>

jint Java_com_corp_examples_Hello_getData(jint* array){
    array[0] = 0xBEEF;
    return 1 ;
}
```

Simulation Platform Example

- Simulation is key to Design GUI
  - Prototype several GUI options
  - Anticipate human factors issues
  - Check graphical design against display characteristics

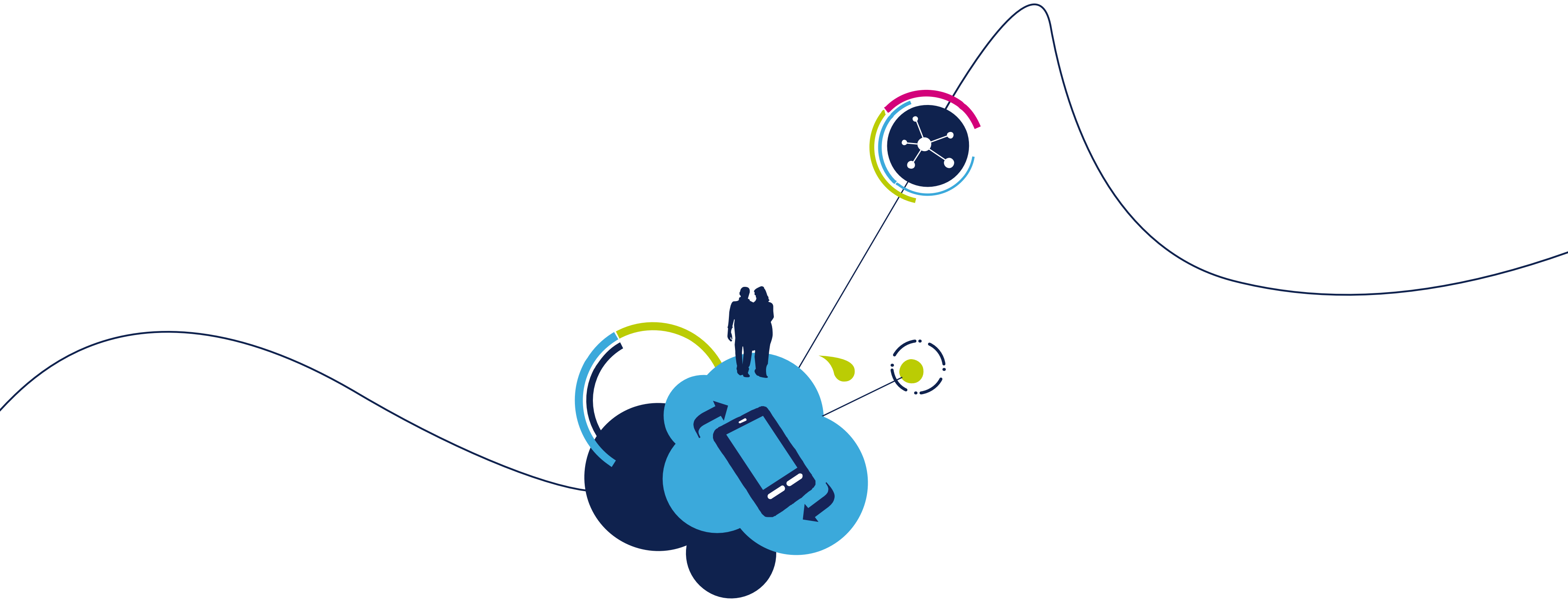


```
public void runTest(Display display, String message) {
  MessageViewable viewable = new MessageViewable(display);
  viewable.init("Hello, world!);
  viewable.show();
}
```

```
public static void blink() {
  while (true) {
    toggle();
    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
  }
}
public static final native boolean toggle();
```

```
public class LED {
  private boolean state;

  public static boolean toggle() {
    state = !state;
    if (state) {
      drawImage("ledOn.png");
    } else {
      drawImage("ledOff.png");
    }
  return state;
  }
}
```

© 2012 IS2T

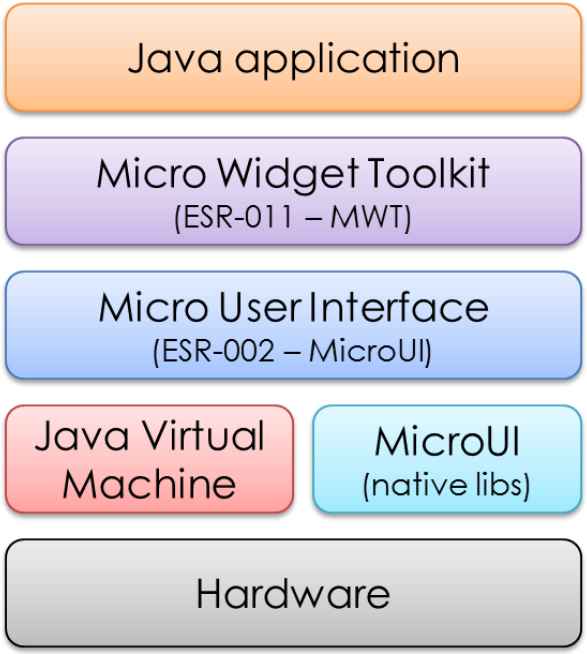


Focus on GUI Design

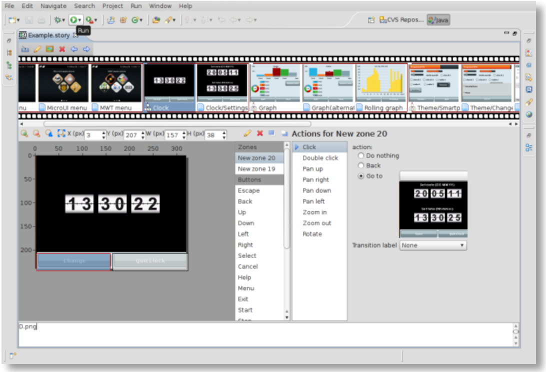
life.augmented

Design GUI with MicroUI® & MWT

- MicroUI (ESR 002)
  - 2D graphical APIs, character LCD, multi-display, fonts
  - Button, joystick, LED, sound, touch & multi-touch
  - Thread-Safe
  - Model-View-Controller based
- MWT (ESR 011)
  - Composite & Layout
  - Focus management
  - Look & Feel
  - Full Java implementation (high portability)



GUI Pack – Tools for GUI Design

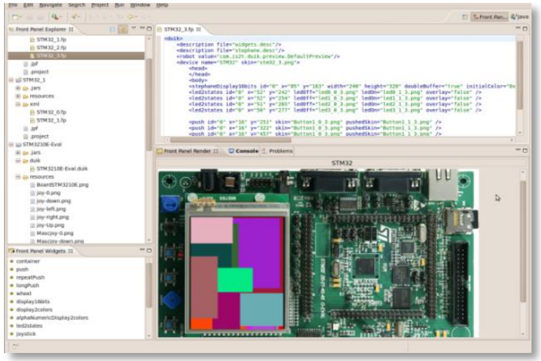


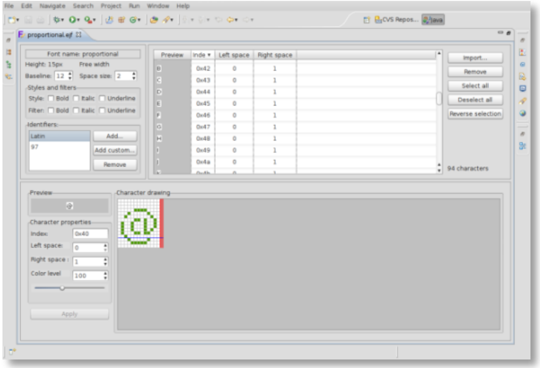
StoryBoard Designer

Prototype how users move around menus

FrontPanel Designer

Design mechanical front panel with displays, LEDs, buttons etc.





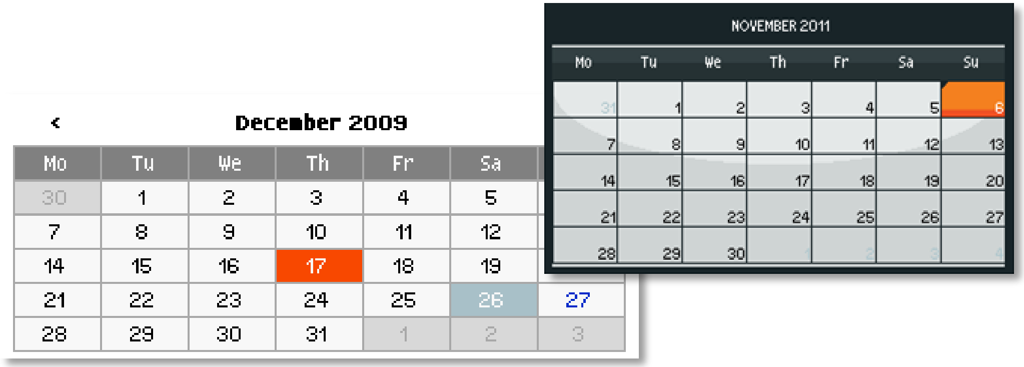
Font Designer

Import fonts or design fonts from scratch

GUI Pack – Widget Libraries

- Simple models
- Small memory footprint
- Look & Feel support
- More complex (composite)
- ….

Source given!

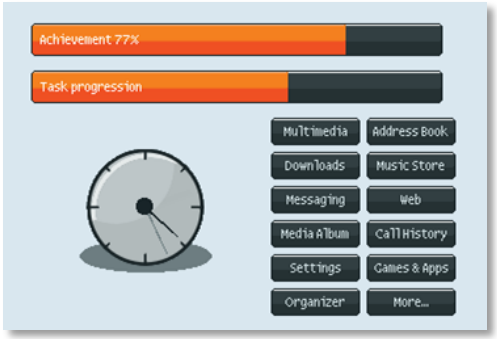


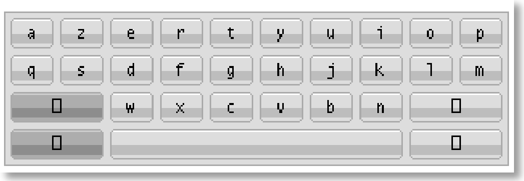
December 2009

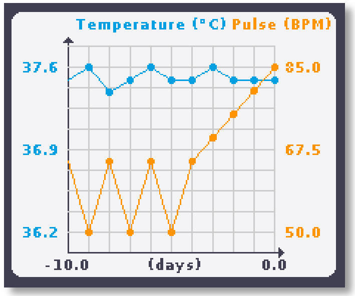
Mo	Tu	We	Th	Fr	Sa	
30	1	2	3	4	5	
7	8	9	10	11	12	
14	15	16	17	18	19	
21	22	23	24	25	26	27
28	29	30	31	1	2	3

NOVEMBER 2011

Mo	Tu	We	Th	Fr	Sa	Su
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4







Multi-lines

Textfield

GUI Portability – Java Platform Level

- Use flexible Java platforms based on the MicroJvm® VM
  - Available on a wide range of processor configurations
  - OS agnostic
  - Easy to interface to C/asm

- Typical usage
  - Microcontrollers (simple RTOS) ↔ Processors (Linux)
  - Android / iOS
- Other advantages
  - Application code is portable too! (not only GUI)
  - Green-thread guarantees same scheduling

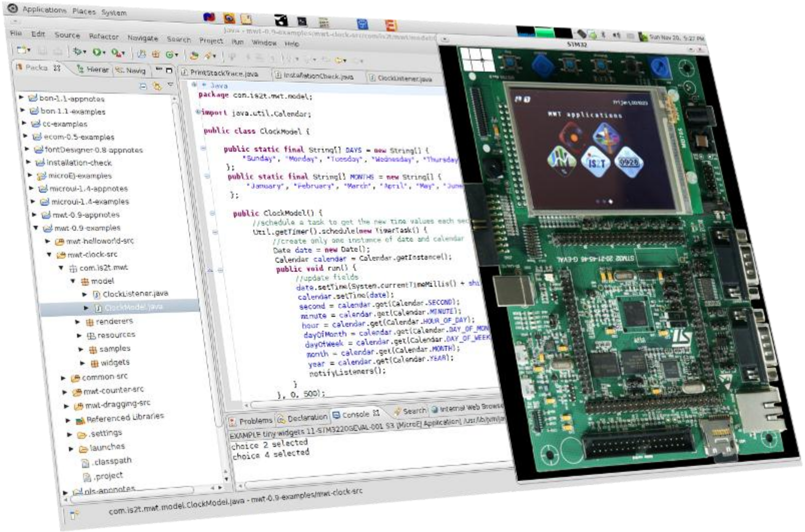
Want to Try STM32Java?

- Free download at www.stm32java.com

- Embedded Java Evaluation Kit
  - STM3220G-JAVA
  - STM3240G-JAVA

- Web resources
  - www.st.com/stm32-java
  - www.stm32java.com

- ST Support
  - support.stm32java@microej.com

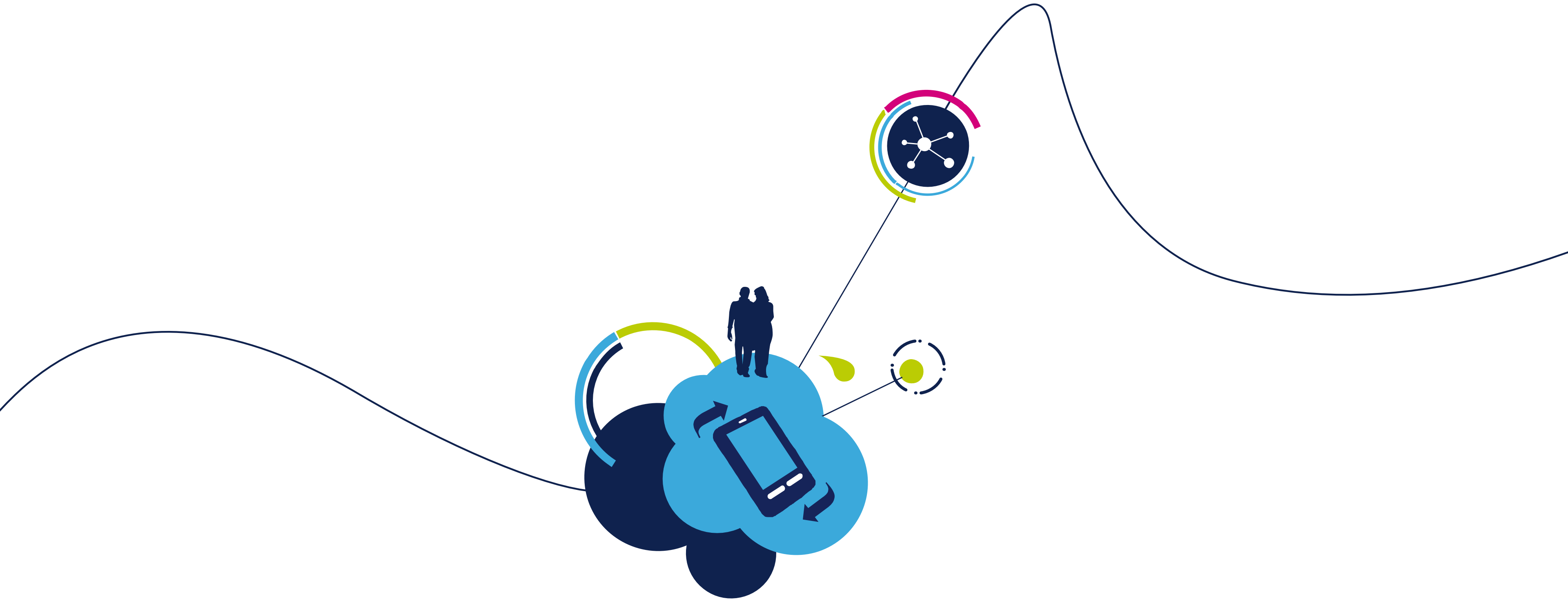


```
package com.is2t.mwt.model;

import java.util.Calendar;

public class ClockModel {

    public static final String[] DAYS = new String[] {
        "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday
    };
    public static final String[] MONTHS = new String[] {
        "January", "February", "March", "April", "May", "June
    };

    public ClockModel() {
        //schedule a task to get the new time values each sec
        Util.getTimer().schedule(new TimerTask() {
            //create only one instance of date and calendar
            Date date = new Date();
            Calendar calendar = Calendar.getInstance();
            public void run() {
                //update fields
                date.setTime(System.currentTimeMillis() + shi
                calendar.setTime(date);
                second = calendar.get(Calendar.SECOND);
                minute = calendar.get(Calendar.MINUTE);
                hour = calendar.get(Calendar.HOUR_OF_DAY);
                dayOfMonth = calendar.get(Calendar.DAY_OF_MON
                dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK
                month = calendar.get(Calendar.MONTH);
                year = calendar.get(Calendar.YEAR);
                notifyListeners();
            }
        }, 0, 500);
```

```
EXAMPLE tiny widgets 11-STM3220GEVAL-001 S3 [MicroEJ Application] /usr/lib/jvm/ja
choice 2 selected
choice 4 selected
```

Summary

- STM32Java solution brings:
  - The intrinsic strengths of Java technology to STM32:
    - OOP, Garbage Collector, Simulation, …
  - Footprints and performances that fit STM32 portfolio
  - Easy integration with existing C code
  - An easy way to build up GUIs
  - A complete answer with silicon, libraries and development environment



Q&A

life.augmented