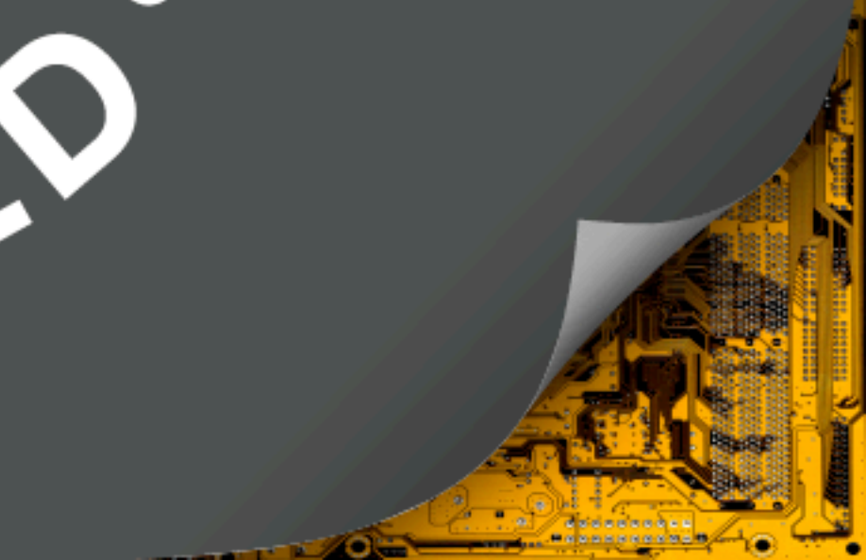
FUN WITH

JavaFx

EMBEDDED



Gerrit Grunwald

canoo Engineering AG

TWITTER: @hansolo_

WEB: harmonic-code.org

first...

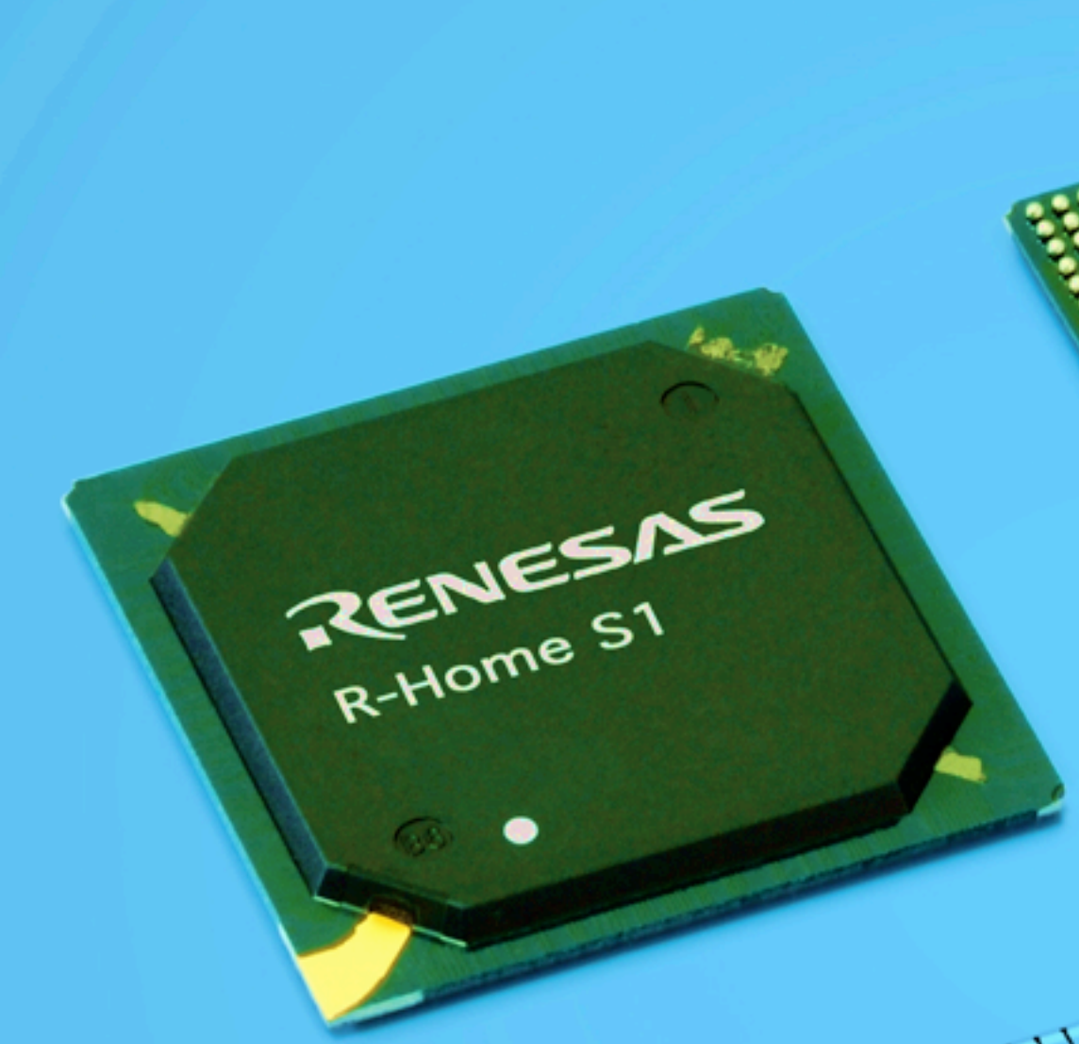
what embedded

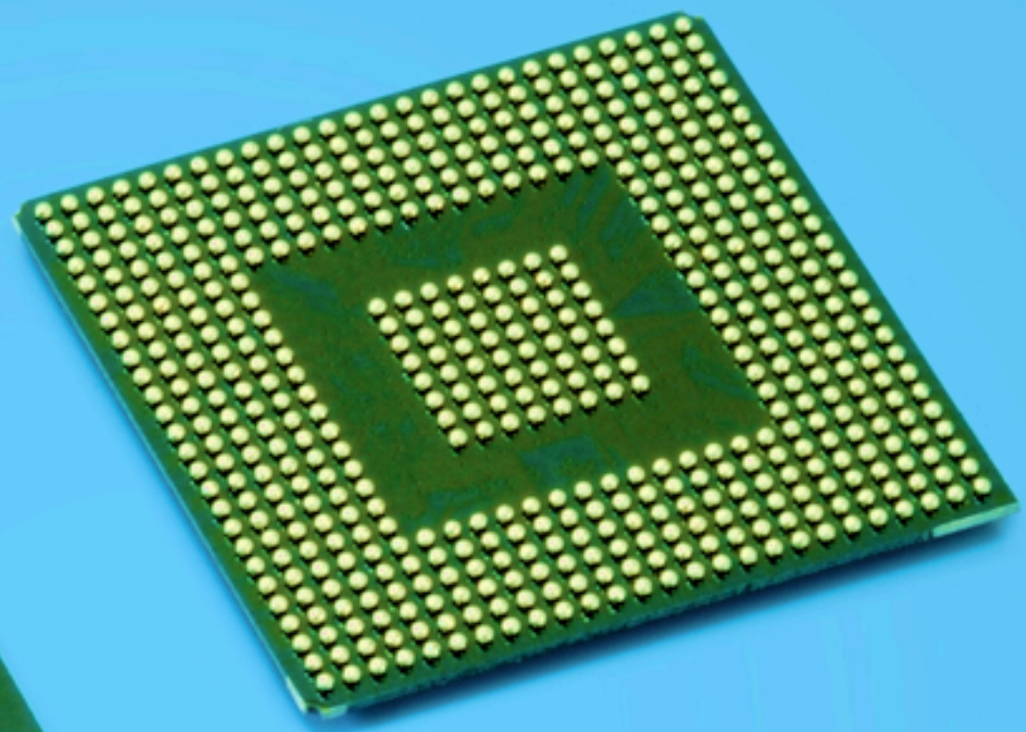
IS...

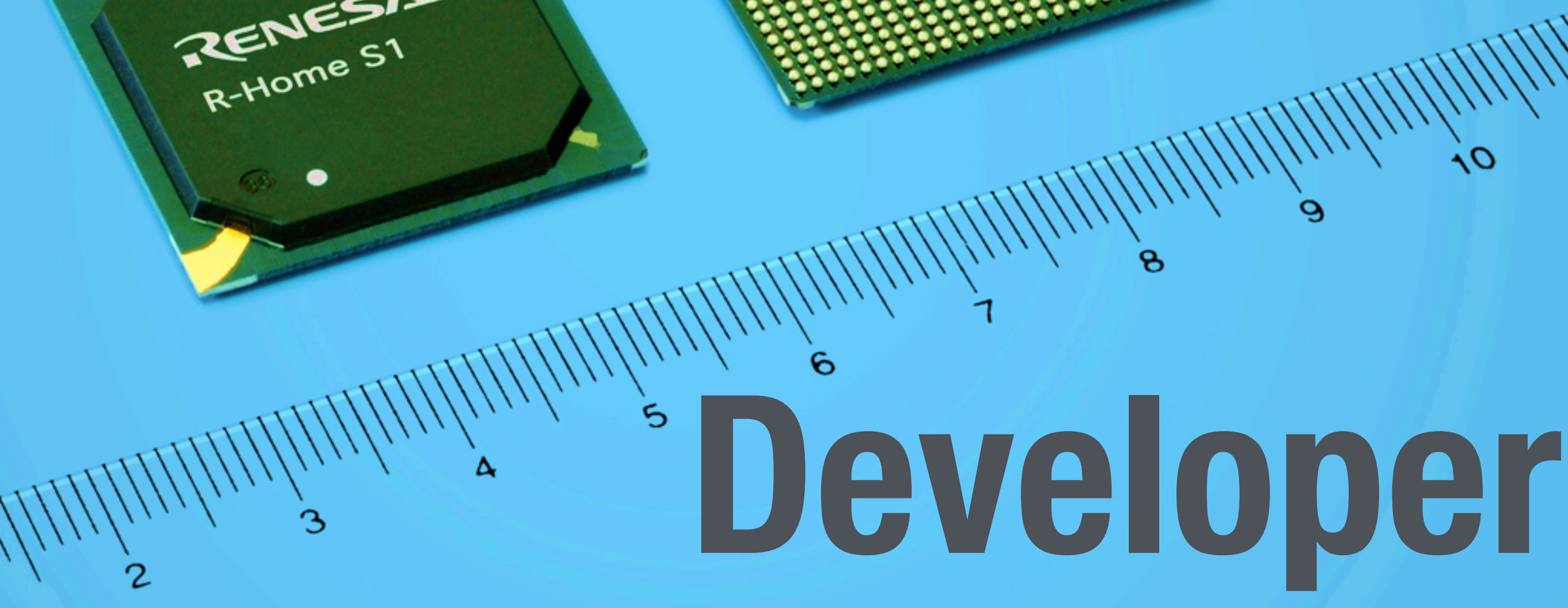
...depends on

the point of view

Hardware

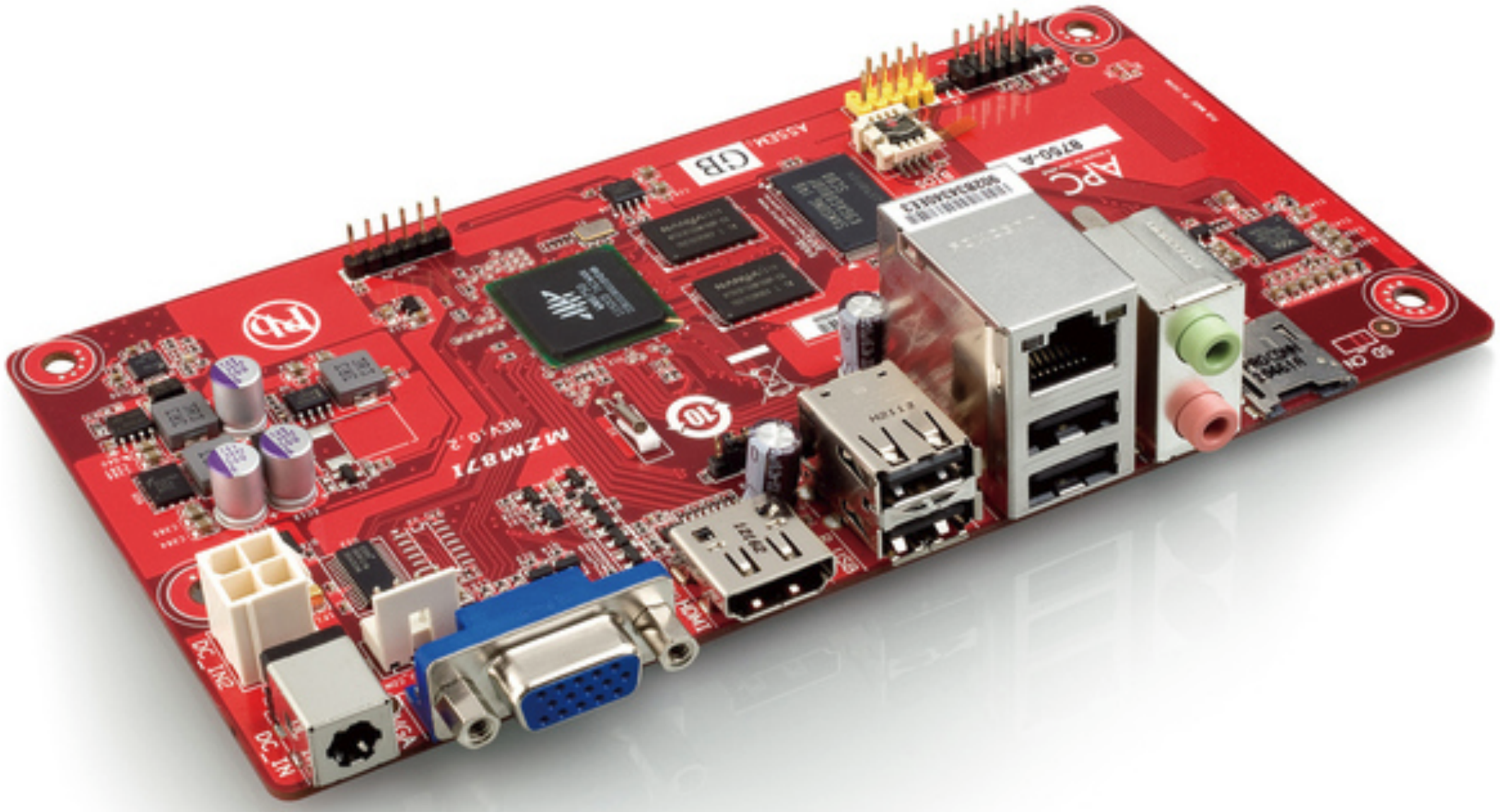






Developer

Software



Developer

JAVA ONE

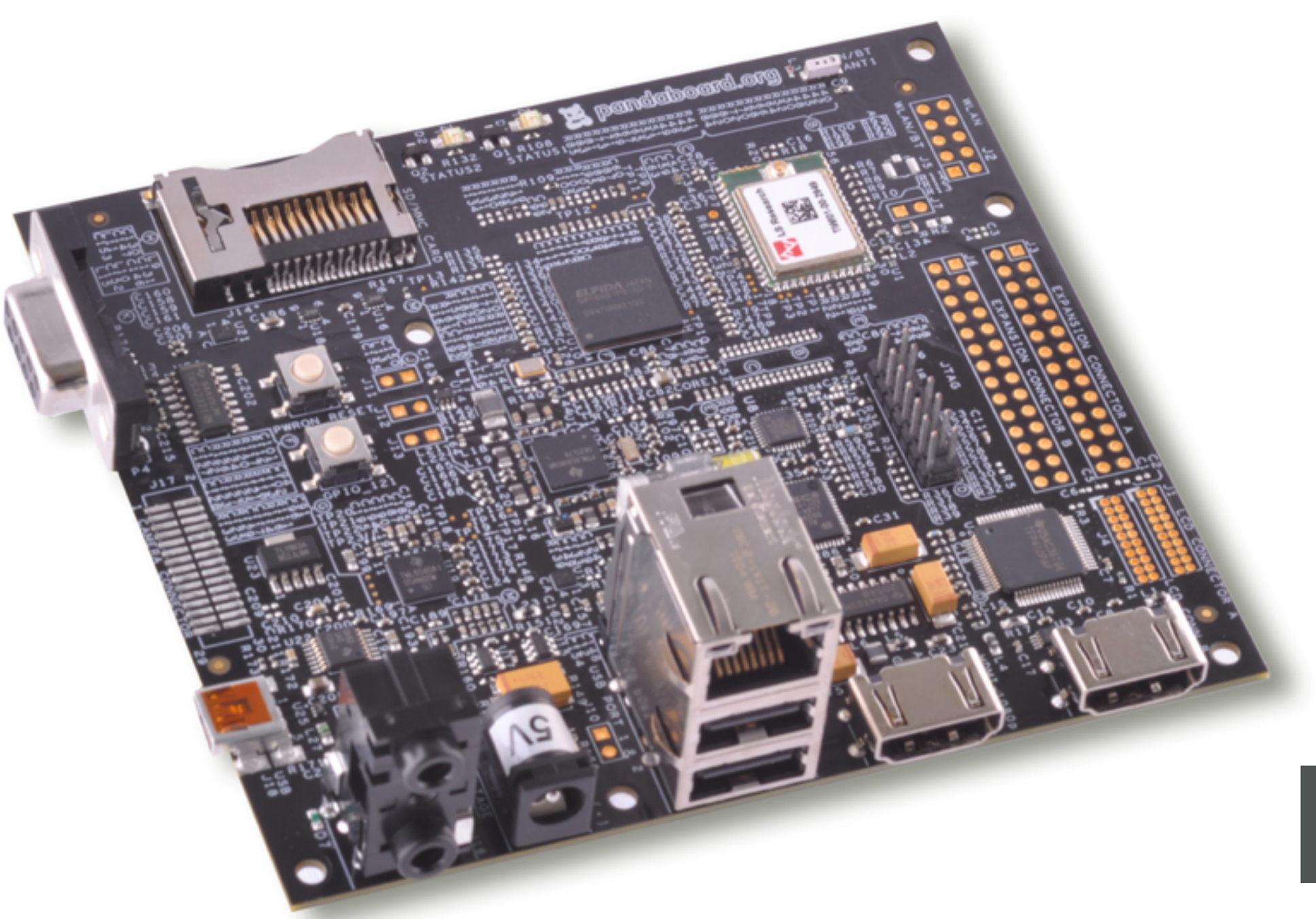
2012

JavaFX

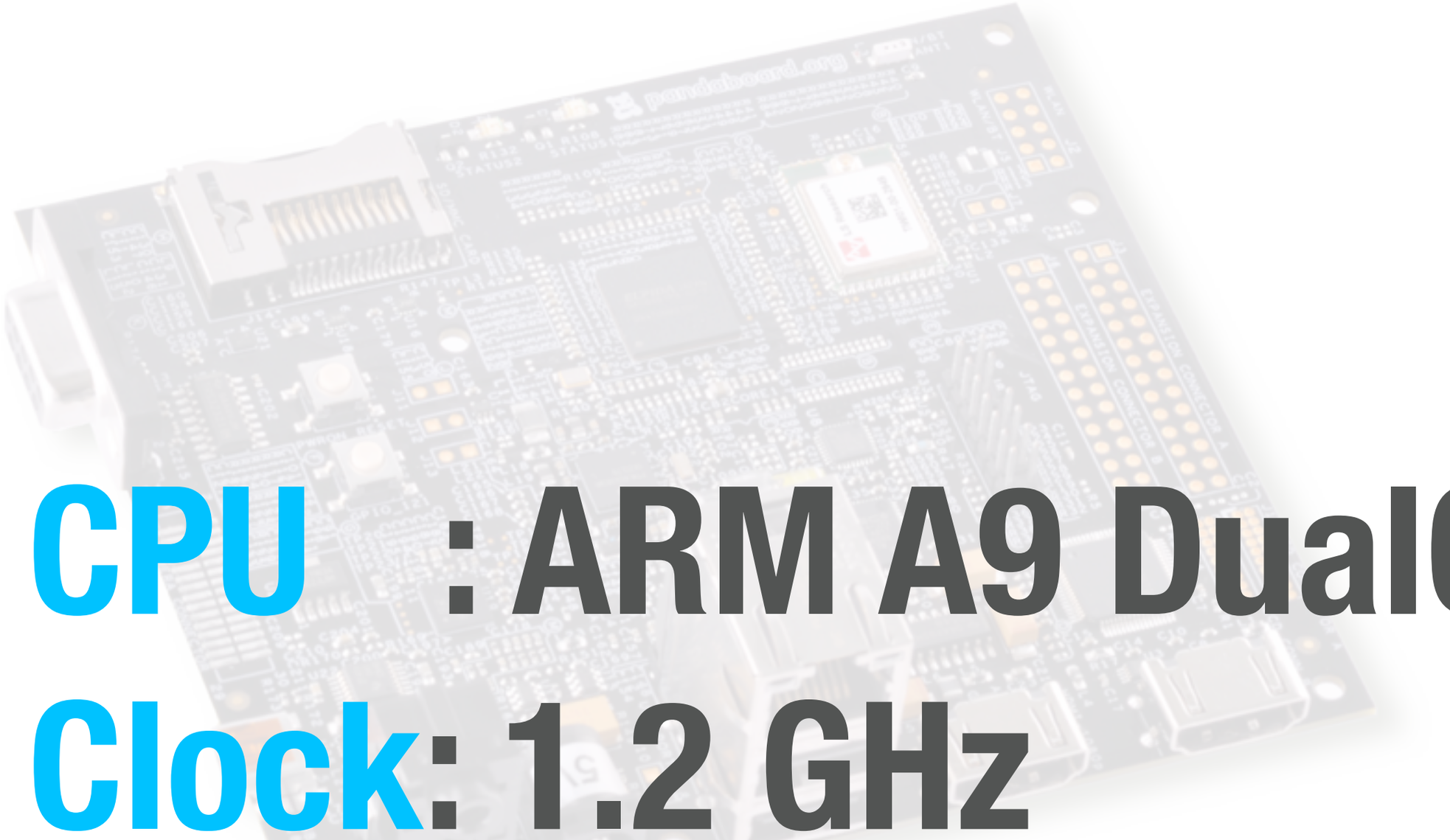
powered

Kiosk

'running on...


pandaboard.org
EXPANSION CONNECTOR A
EXPANSION CONNECTOR B
JTAG
PWRON RESET
5V

Panda Board



CPU : ARM A9 DualCore

Clock: 1.2 GHz

Ram : 1 GB

GPU : Power VR SGX540

Why Java?

Benefits of Java

- ★ **Superb developer toolchain**
- ★ **Mature, fast, widespread**
- ★ **Thousands of libraries**
- ★ **Huge community**
- ★ **No standard on embedded**

WTF ?

JavaFX

5

Possible

USE CASES

- *Home automation*
- *Home entertainment*
- *Medical devices*
- *Information Kiosks*
- *Education*

WHAT IS JAVAFX ON EMBEDDED ?

A Subset of **F**

JAVAx

Without support for

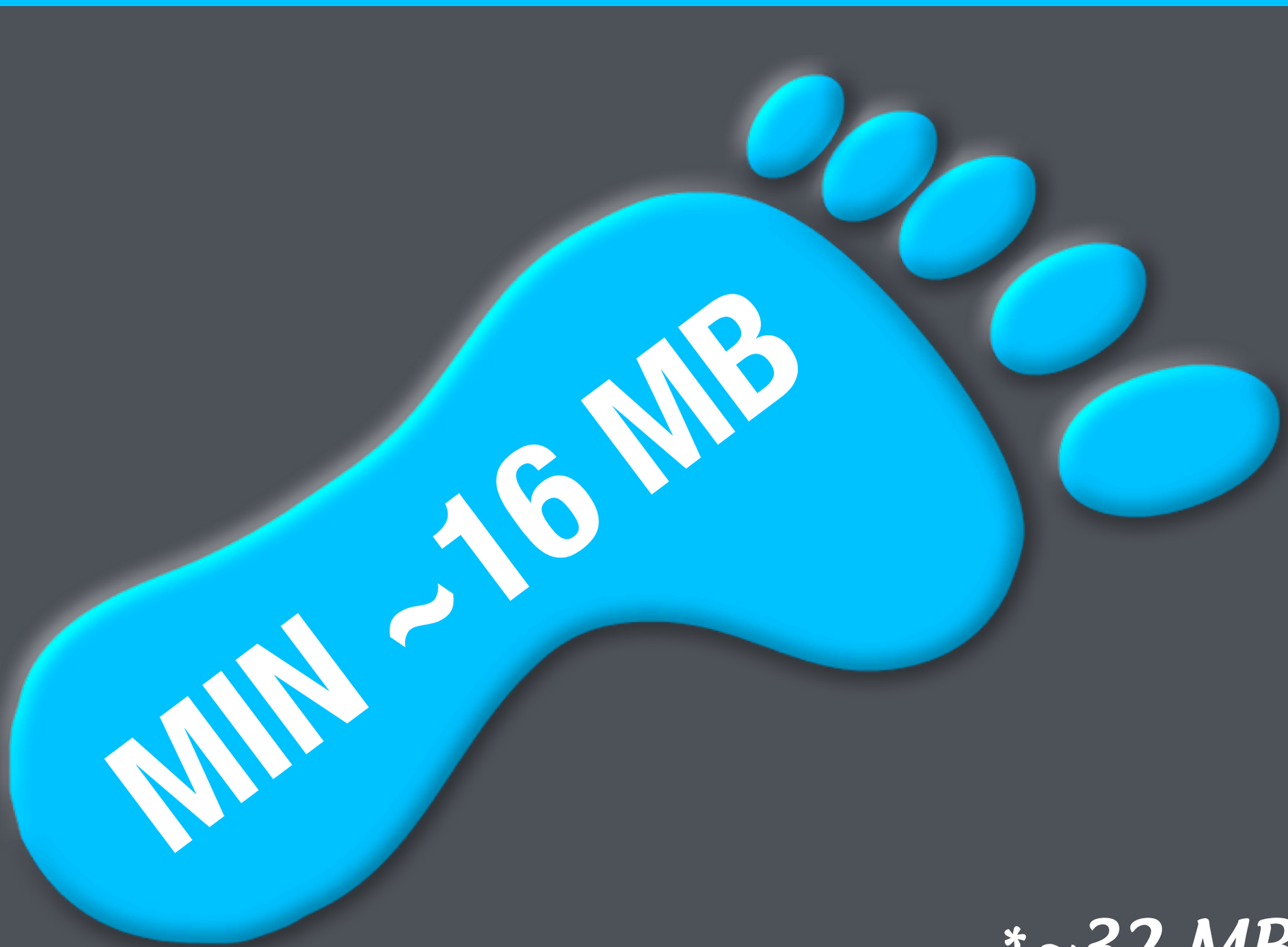
- ★ **Swing/SWT**
- ★ **System Menu**
- ★ **Drag'n Drop**
- ★ **WebView**
- ★ **Media (e.g. AudioClip)**

Available JDK's

04/2013

- JDK 7 (JFX 2)
- JDK 8 (JFX 8)

Target footprint*



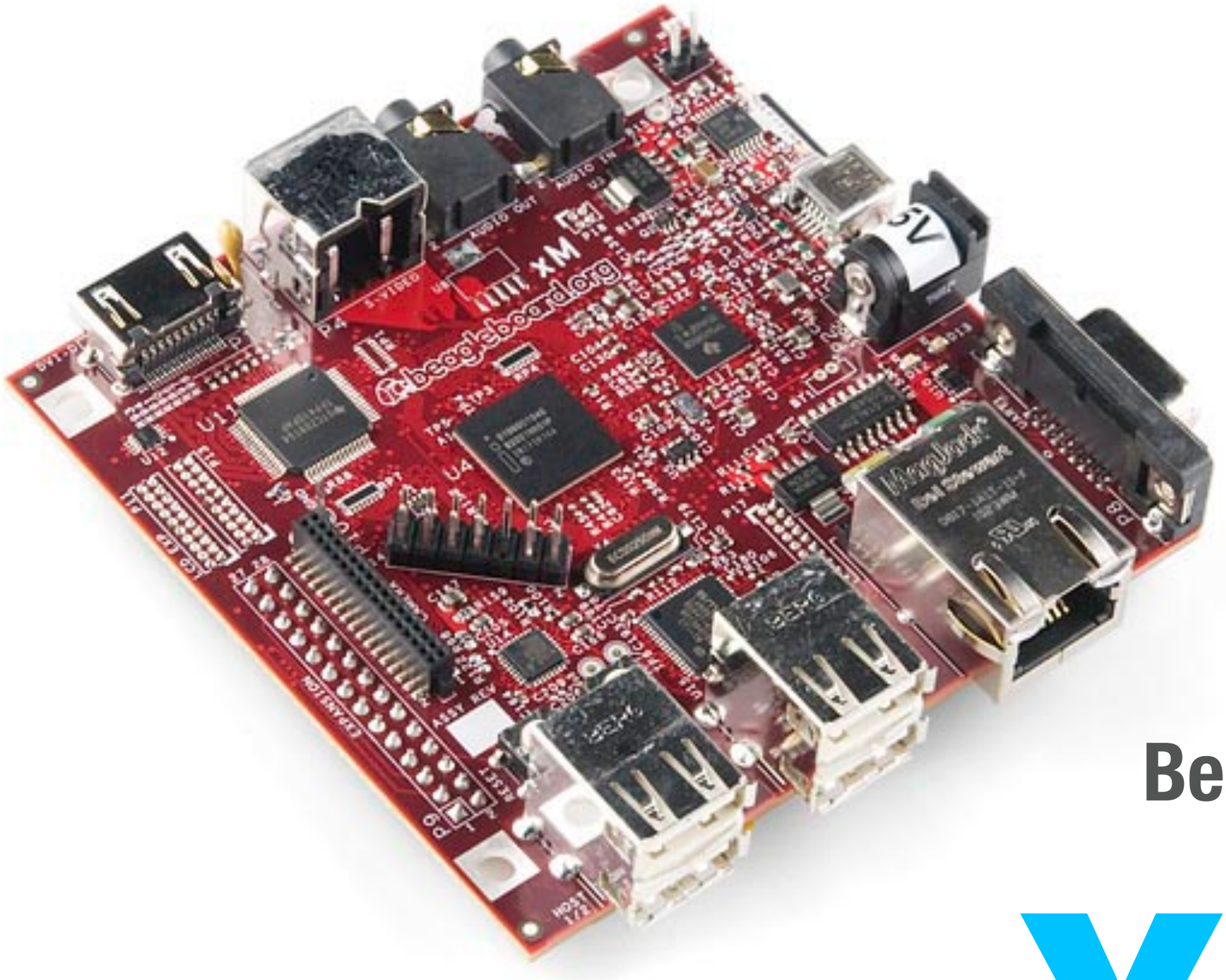
** ~32 MB today*

WHAT ARE THE SUPPORTED PLATFORMS ?

04/2013

BeagleBoard xM

Raspberry Pi



BeagleBoard

xM

- ***ARM A8, 1 GHz***
- ***512 MB RAM***
- ***4 x USB***
- ***ETHERNET RJ45***
- ***HDMI***
- ***I^2C, JTAG, SPI***

running on

Angström

based on Debian

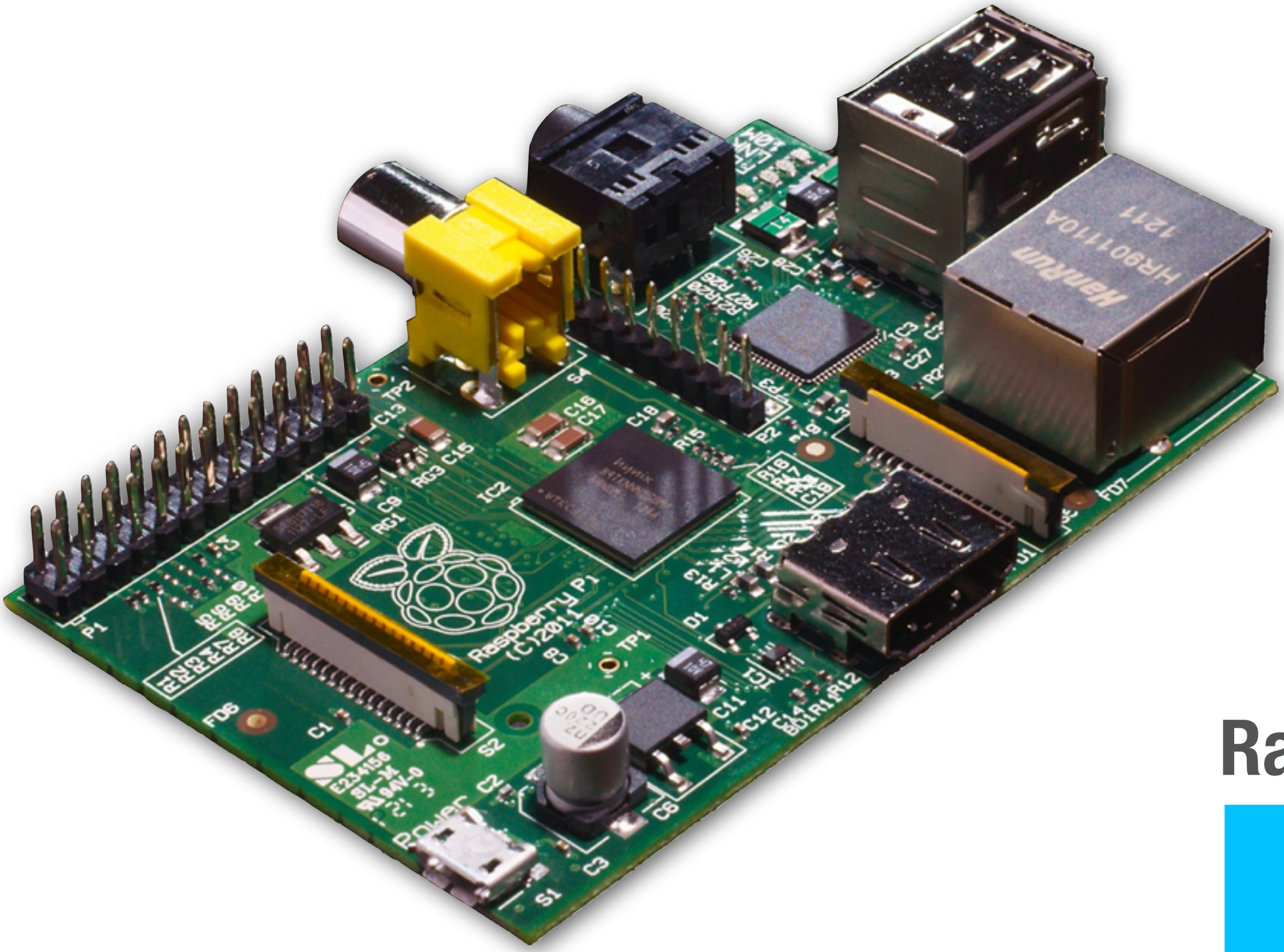
JDK 7

JavaFX 2

with SoftFloat support

BeagleBoard

xM



Raspberry Pi

- ARM v6, 700 MHz
- 512 MB RAM
- 2 x USB
- ETHERNET RJ45
- HDMI, COMPOSITE
- GPIO, I^2C, UART, SPI

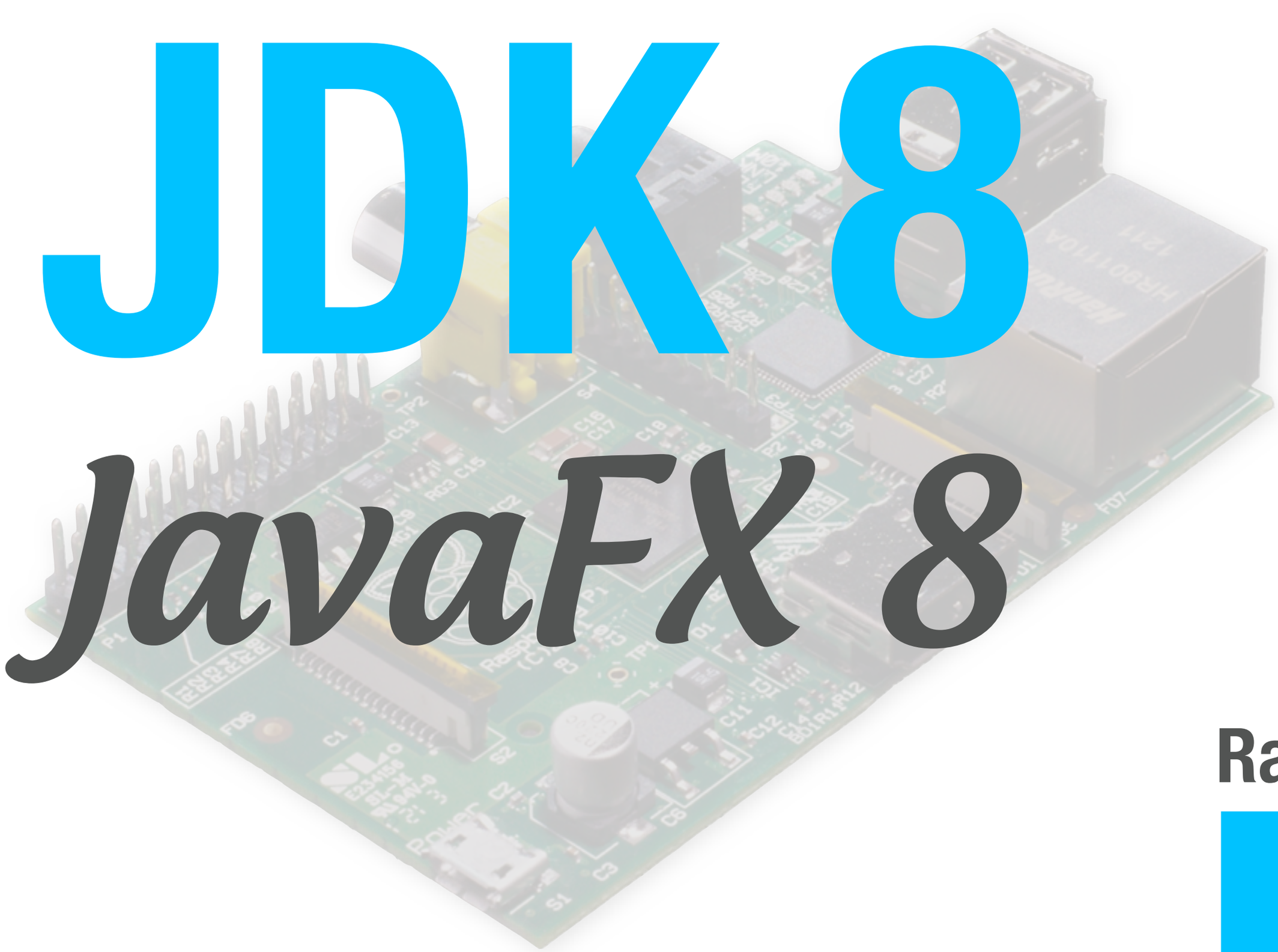
running on

Raspbian

based on Debian

JDK 8

JavaFX 8



with HardFloat support

Raspberry Pi

So....having

Java

means...

WRITE ONCE RUN ANYWHERE ?

FIRST...

SOME FACTS

Macbook Pro

Intel i7 Quadcore

2.3 GHz

16 GB ram

Nvidia GeForce

GT 650m

BeagleBoard xM

ARM A8

1 Ghz

512 MB ram

Power VR

SGX series 5

Macbook Pro

Nvidia GeForce GT 650m

- ★ 384 Cores
- ★ ~15 GPixel/s
- ★ ~600 GFlops

BeagleBoard xM

Power VR SGX series 5

- ★ 1 Core
- ★ ~500 MPixel/s
- ★ ~1.6 GFlops

Embedded

REQUIREMENTS

Requirements

- ★ **touchable user interface**
- ★ **reasonable controls**
- ★ **no mouse and keyboard**
- ★ **restriced screen estate**

Things you

BETTER AVOID

Things to avoid

- **huge amount of nodes**
- **intense use of animations**
- **intense use of effects**
- **many overlapping nodes**

CONCLUSION

NO

WRITE ONCE RUN

ANYWHERE !

But you can

RECYCLE A LOT

60 cm

34 cm

50 mm

90 mm

and learn

FROM MOBILE

EXAMPLE

Temperature Monitoring

Requirements

- **Measure the temperature**
- **Monitor on site**
- **Monitor on desktop**
- **Monitor on mobile**

Requirements

- **Feedback at the sensor**
- **Feedback on site**
- **Feedback on desktop**
- **No platform dependency**

and...

NO

SOLDER

Hardware

Hardware

- **Raspberry Pi for measuring the temperature**
- **BeagleBoard xM with lcd for on site monitoring**

A Raspberry Pi for

measuring ?

Isn't that

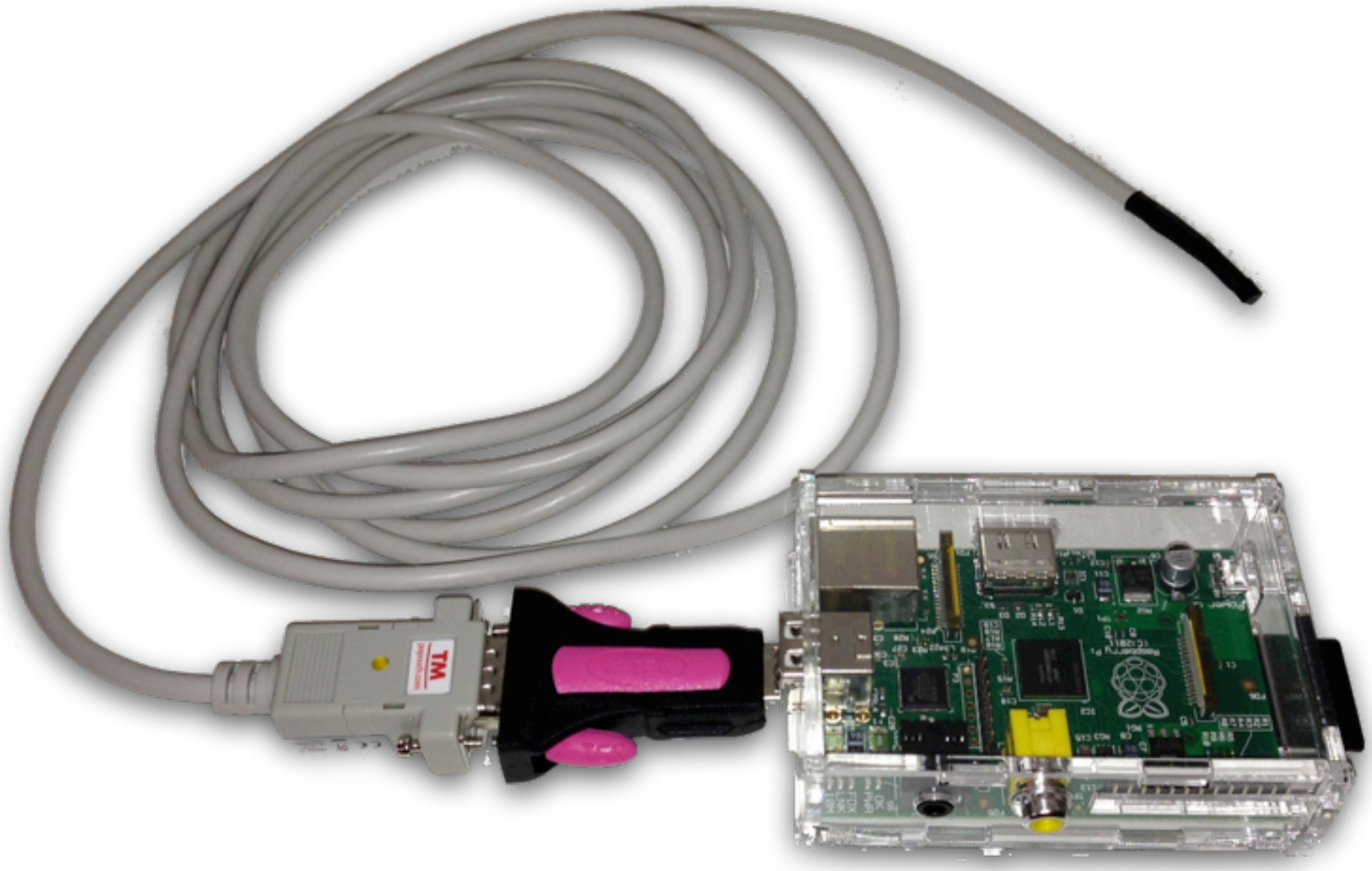
Overkill ?

OF COURSE

but...

it's

cheap



Raspberry

Pi

simple version

Raspberry Pi

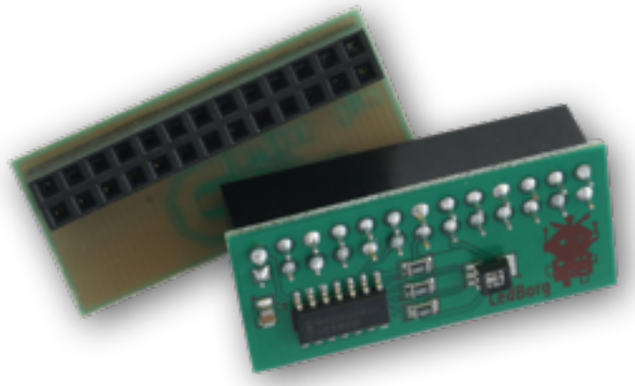
- ★ Raspberry Pi
- ★ Case for the Pi
- ★ Power Supply
- ★ Serial to USB Converter
- ★ Industrial serial temp sensor
- ★ Network connection

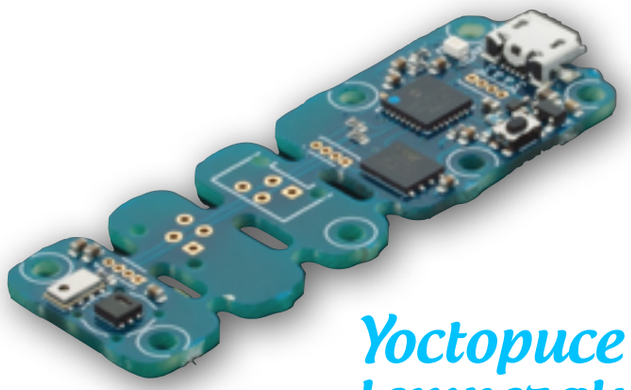
Estimated

100 $

Yoctopuce USB
OLED Display

PiBorgs
LedBorg RGB led





Yoctopuce USB
temperature, humidity
+ pressure sensor

Raspberry
Pi

more advanced version

Raspberry Pi

advanced

- ★ Raspberry Pi
- ★ Case for the Pi
- ★ Power Supply
- ★ Yoctopuce Meteo sensor
- ★ Yoctopuce MaxiDisplay
- ★ PiBorg LedBorg RGB led
- ★ Network connection

Estimated

240 $

BeagleBoard

xM

BeagleBoard

- ★ Beagleboard xM
- ★ 10" LCD touchscreen
- ★ Case
- ★ Power Supply
- ★ Yoctopuce MaxiRelay*
- ★ Some indicator (Werma deSIGN42)*

*optional

Estimated

350 $

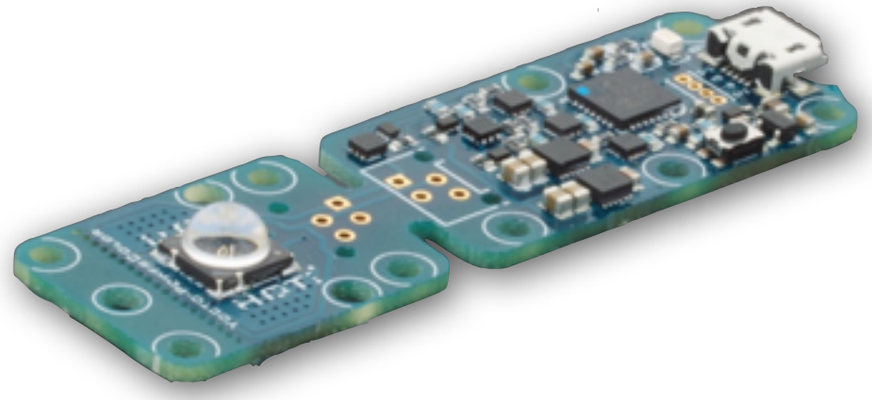
excl. the Signaltower

Indicators

Werma deSIGN 42

~ 390 $

or



Yoctopuce PowerColor

~ 85 $

Desktop

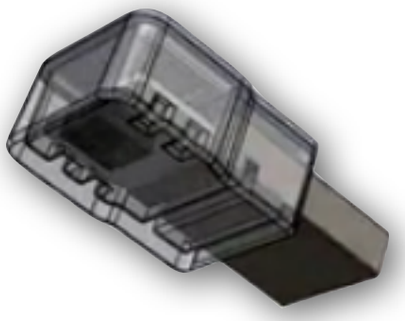
Desktop

- ★ Windows, OS X or Linux
- ★ Java Virtual Machine (> JDK 7u6)
- ★ Network connection
- ★ blink(1) Led indicator

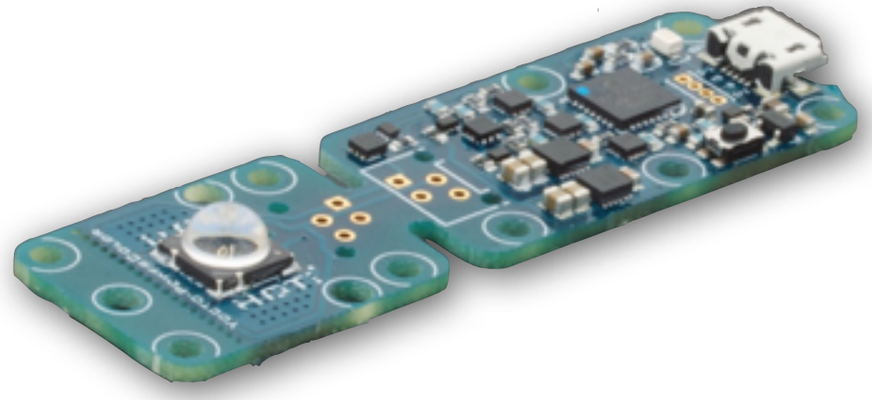
Indicators

ThingM blink(1)

~ 30 $



or



Yoctopuce PowerColor

~ 85 $

Mobile

Mobile

- **iOS, Android, ...**
- **HTML5 capable browser**
- **Network connection**

Software

Platforms

- **Raspberry Pi on Java 8**
- **BeagleBoard xM on Java 7**
- **Desktop Client on Java 7**
- **Mobile Client on HTML5**

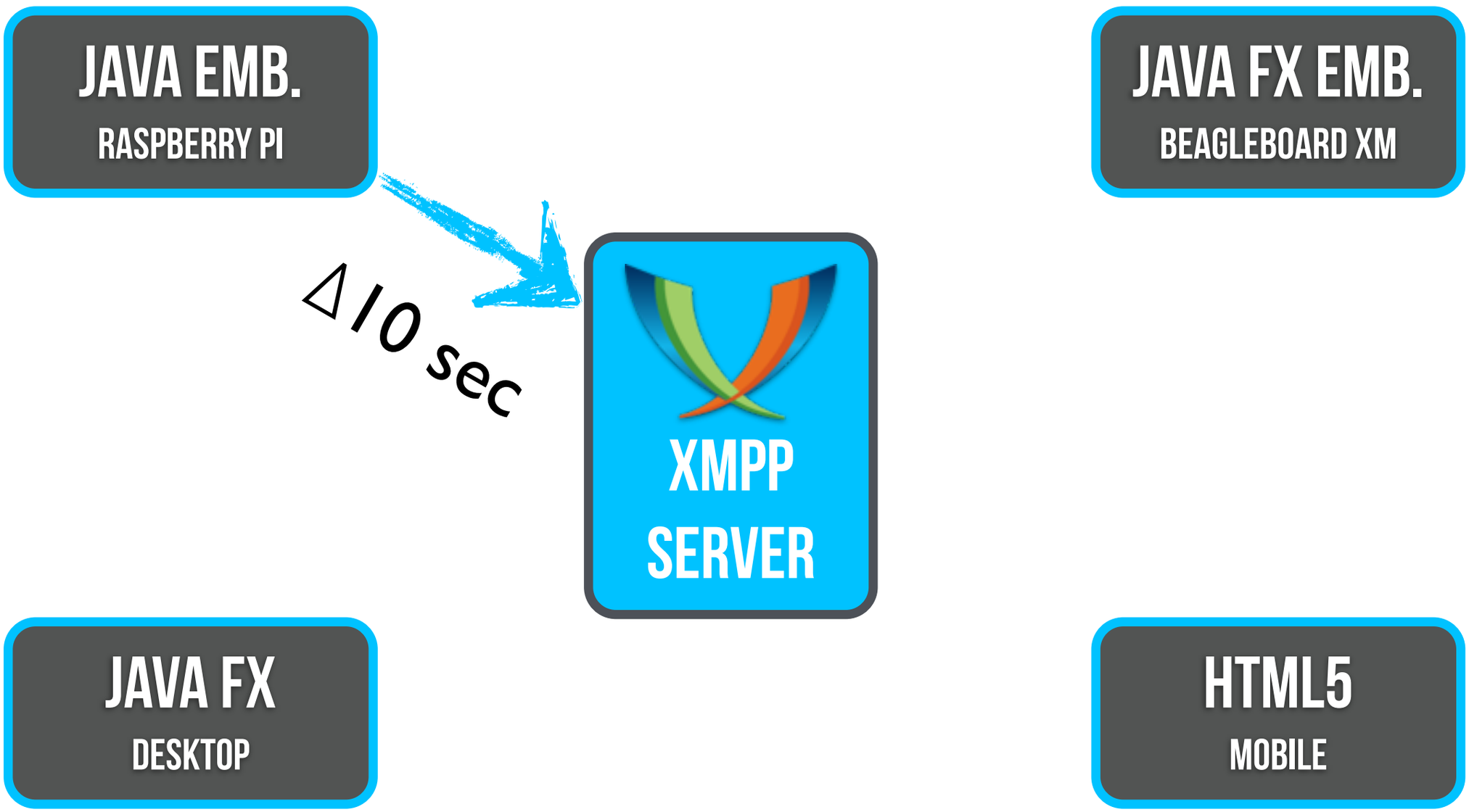
What about

COMMUNICATION

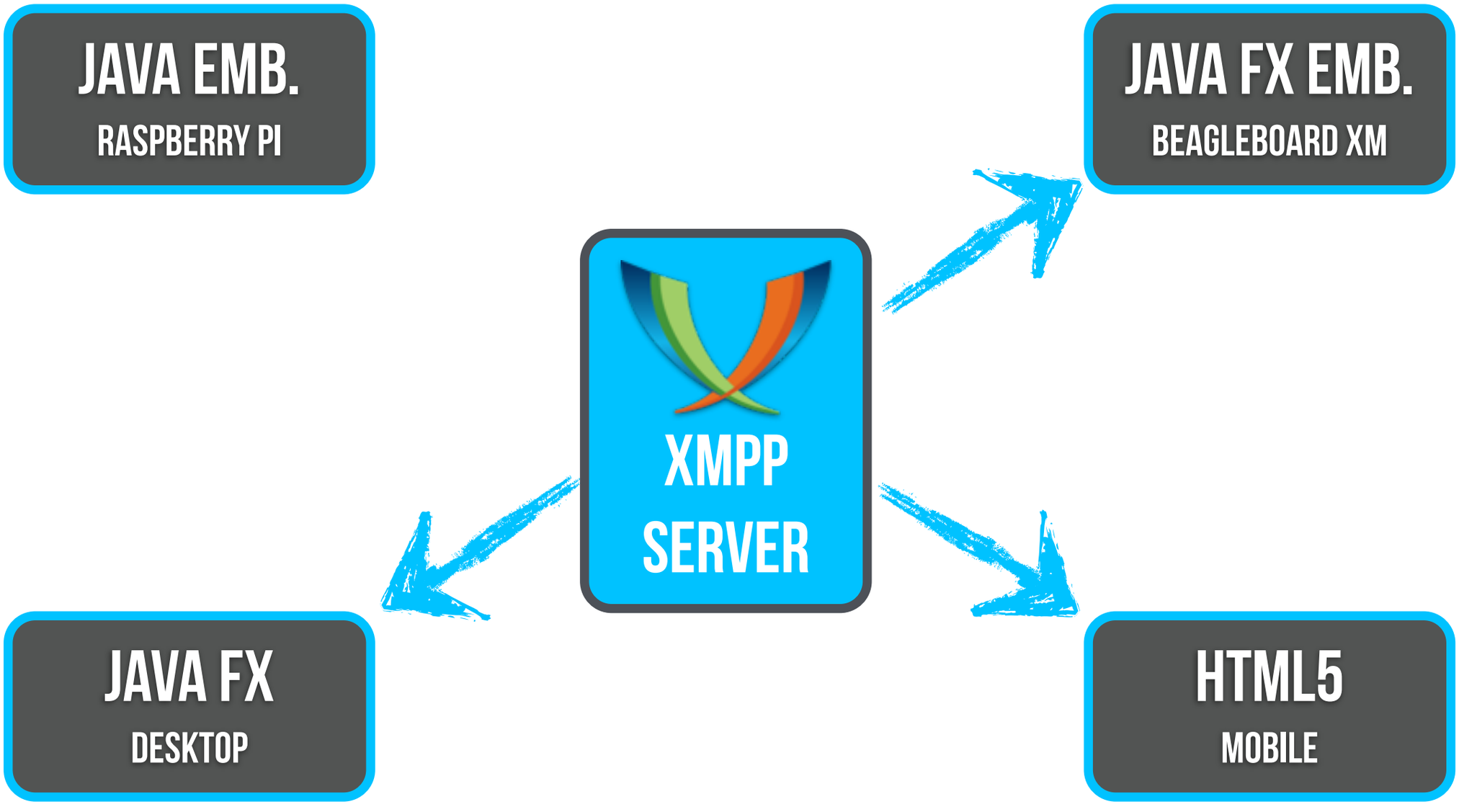
Communication

- **xmpp** (extensible messaging and presence protocol)
- **smack xmpp java library**
- **bosh** (bidirectional streams over synchronous http)

Overview


JAVA EMB.
RASPBERRY PI
JAVA FX EMB.
BEAGLEBOARD XM
Δ10 sec
XMPP
SERVER
JAVA FX
DESKTOP
HTML5
MOBILE

Overview


JAVA EMB.
RASPBERRY PI
JAVA FX EMB.
BEAGLEBOARD XM
XMPP
SERVER
JAVA FX
DESKTOP
HTML5
MOBILE

Advantage of xmpp

- **has free infrastructure**
- **is widely used**
- **is extensible**
- **supports presence**

Advantage of xmpp

- ★ supports resources
- ★ is fast
- ★ is mature

1

Raspberry Pi

APPLICATION

Requirements

- measure the temperature
- distribute the data via xmpp
- log the data*

*optional

running on JDK8

using

JavaFX 8

JavaFX

ON HEADLESS

Sensor Class

XMPP Class

Sensor Class

```
DoubleProperty celsius = new SimpleDoubleProperty();

public ReadOnlyDoubleProperty celsiusProperty() {
    return celsius;
}
```

XMPP Class

Sensor Class

```
DoubleProperty celsius = new SimpleDoubleProperty();

public ReadOnlyDoubleProperty celsiusProperty() {
    return celsius;
}


...
  BufferedReader br =
    new BufferedReader(new InputStreamReader(is));
  while (running) {
    try {
      while((br.ready() && (line = br.readline) != null)) {
        celsius.set(Double.parseDouble(line));
    } catch (Exception exception) {}
  }
...
```

XMPP Class

Sensor Class

```
DoubleProperty celsius = new SimpleDoubleProperty();

public ReadOnlyDoubleProperty celsiusProperty() {
    return celsius;
}
```

XMPP Class

```
sensorClass.celsiusProperty().addListener(
  new ChangeListener<Number>() {
    @Override public void changed(
      ObservableValue<? extends Number> ov,
      Number oldC, Number newC) {
        sendMessage(newC.doubleValue(), RECEIVER);
      }
  });
```

ChangeListener fires only when value changed !

xmpp is

extensible

xmpp is extensible

```
// Exchange data with others
public void sendData(String id, double celsius, double fahrenheit,
                     double humidity, double pressure, double latitude,
                     double longitude, String signalTowerColor,
                     String JID) throws XMPPException {

  Message message = new Message();
  message.setProperty("id", id);
  message.setProperty("celsius", celsius);
  message.setProperty("fahrenheit", fahrenheit);
  message.setProperty("humidity", humidity);
  message.setProperty("pressure", pressure);
  message.setProperty("latitude", latitude);
  message.setProperty("longitude", longitude);
  message.setProperty("signalTowerColor", signalTowerColor);

  Chat chat = chatManager.createChat(JID, messageListener);
  chat.sendMessage(message);
}
```

xmpp == jabber

why not chatting

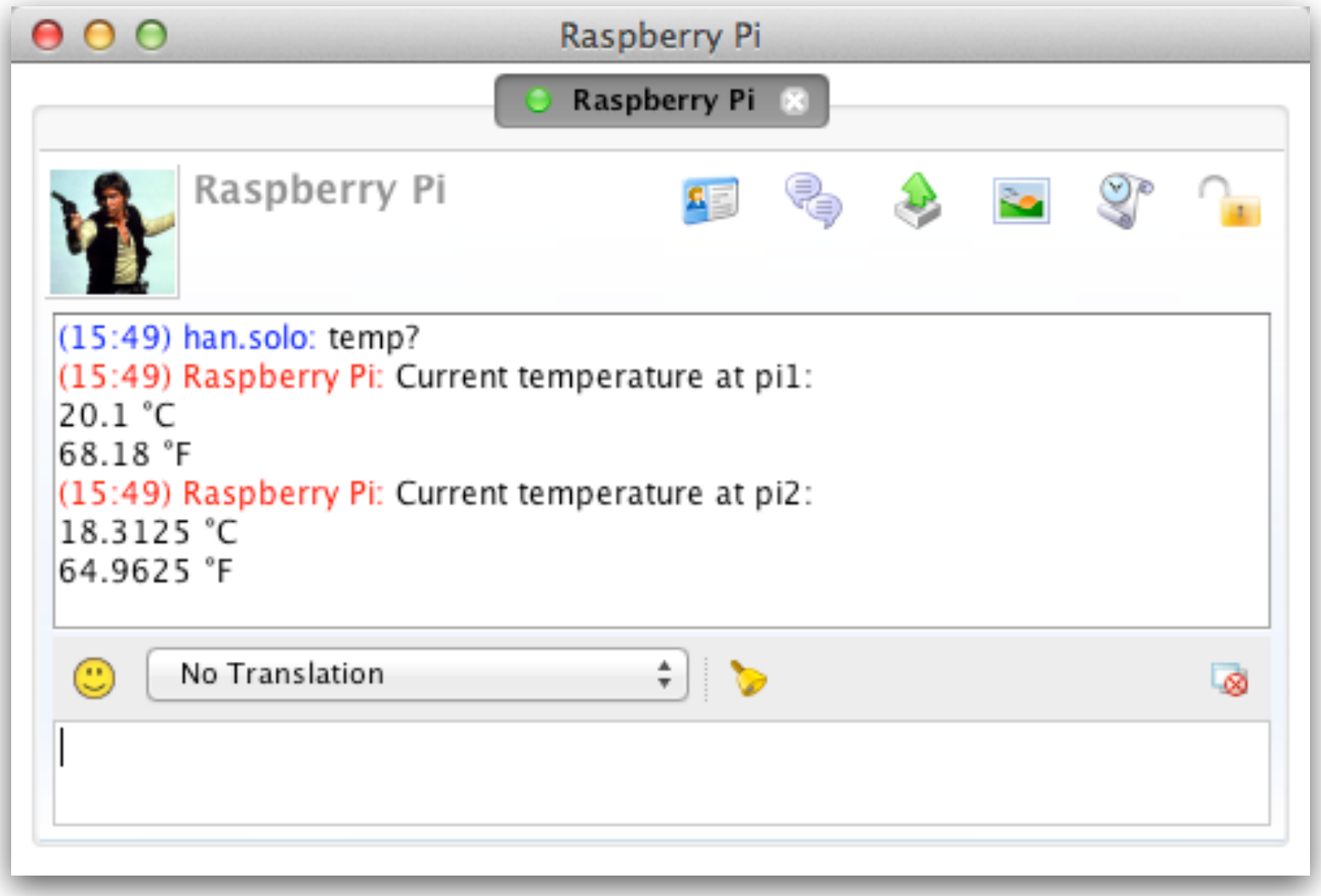
Chat with the Pi

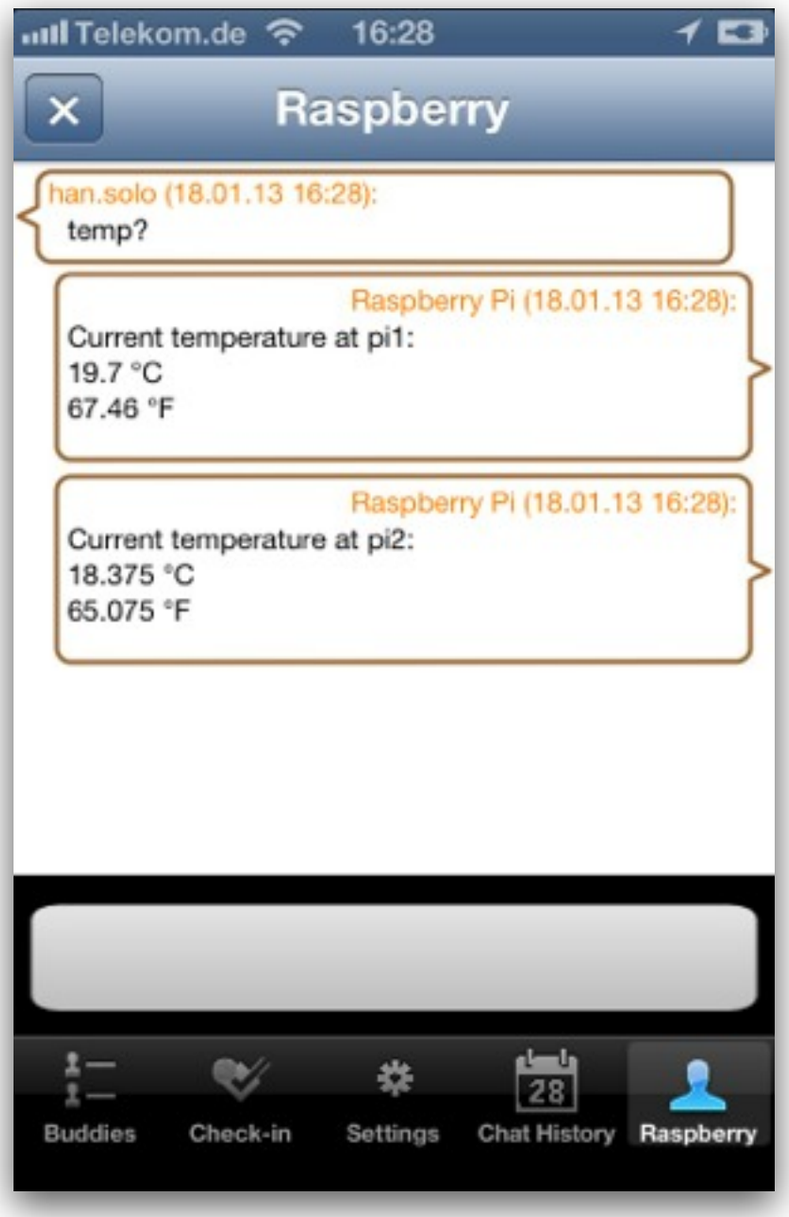
```
// The XMPP Packet listener running on the Raspberry Pi
private class XmppPacketListener implements PacketListener {
  @Override public void processPacket(Packet packet) {
    String from = ((Message) packet).getFrom();
    String body = ((Message) packet).getBody();

    if (body.toLowerCase().equals("temp")) {
      answerTempRequest(from);
    } else if (body.toLowerCase().equals("location")) {
      answerLocationRequest(from);
    } else if (body.toLowerCase().equals("history")) {
      answerHistoryRequest(from);
    } else if (body.toLowerCase().equals("humidity")) {
      answerHumidityRequest(from);
    } else if (body.toLowerCase().equals("pressure")) {
      answerPressureRequest(from);
    }
  }
}
```

Chat with the Pi

```
// Answering the temperature request
public void answerTempRequest(final String JID) {
  new Thread(new Runnable() {
    @Override public void run() {
      try {
        Message message = new Message();
        message.setBody("Current temperature at " + id + ":\n" +
                        celsius + " °C\n" +
                        fahrenheit + " °F");
        Chat chat = chatManager.createChat(JID, messageListener);
        chat.sendMessage(message);
      } catch (XMPPException exception) {...}
    }
  }).start();
}
```

Chat with the Pi





2

BeagleBoard

APPLICATION

running on

JDK7

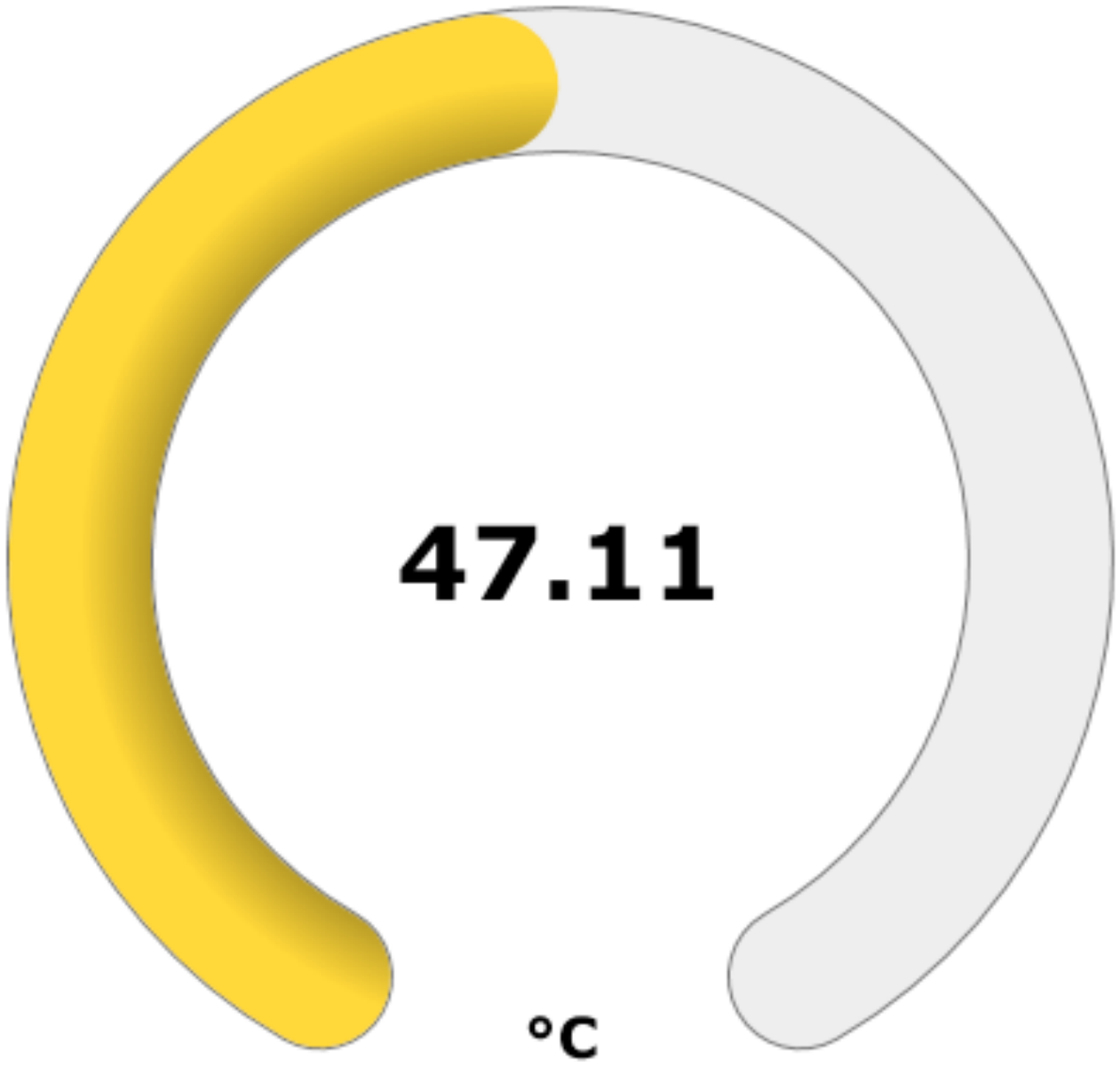
using JavaFX 2

Requirements

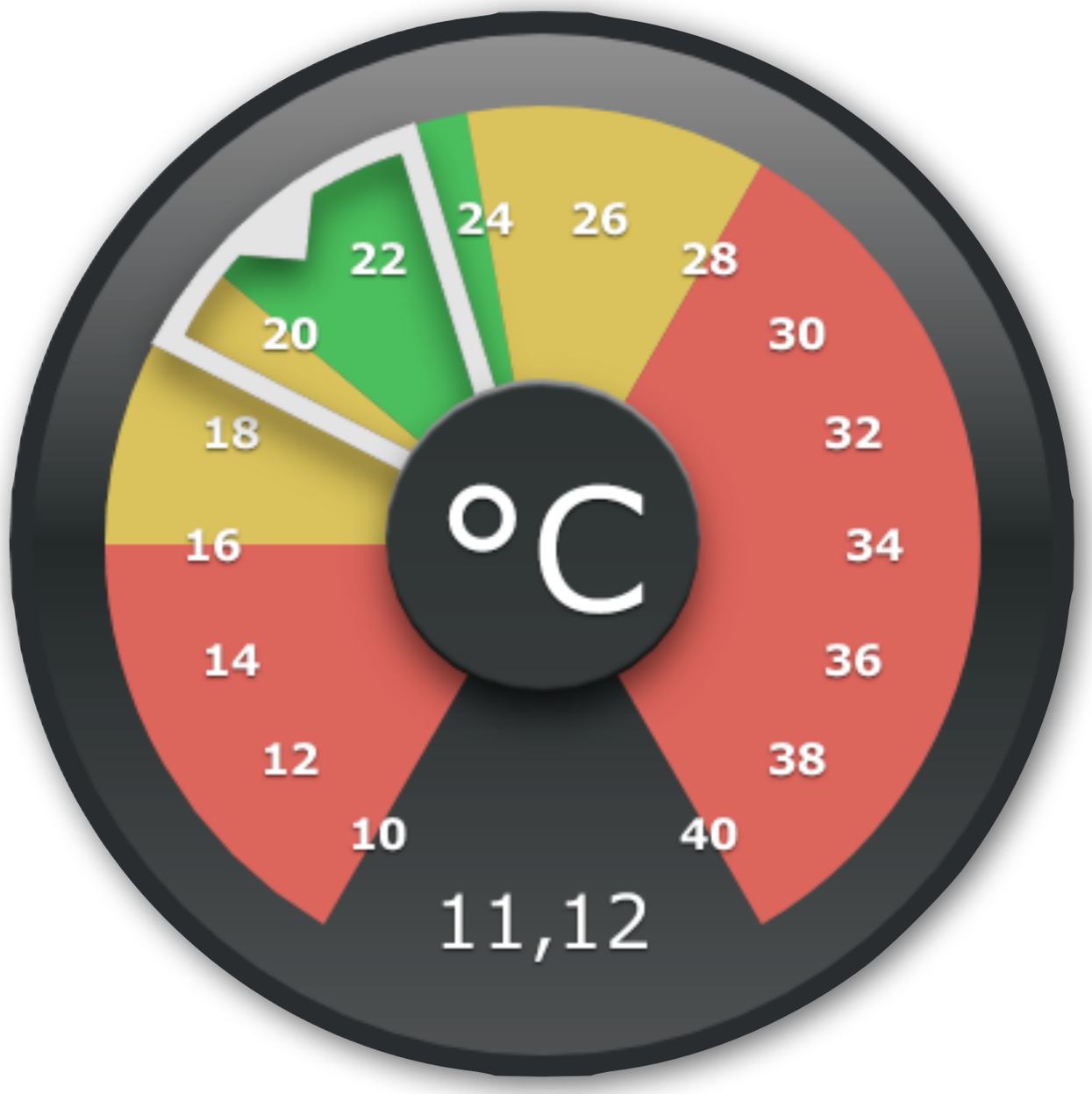
- **Visualize the temperature on connected lcd**
- **Indicate the status on the outside**

Do we need this ?

Isn't this enough ?



Ok, let's take this

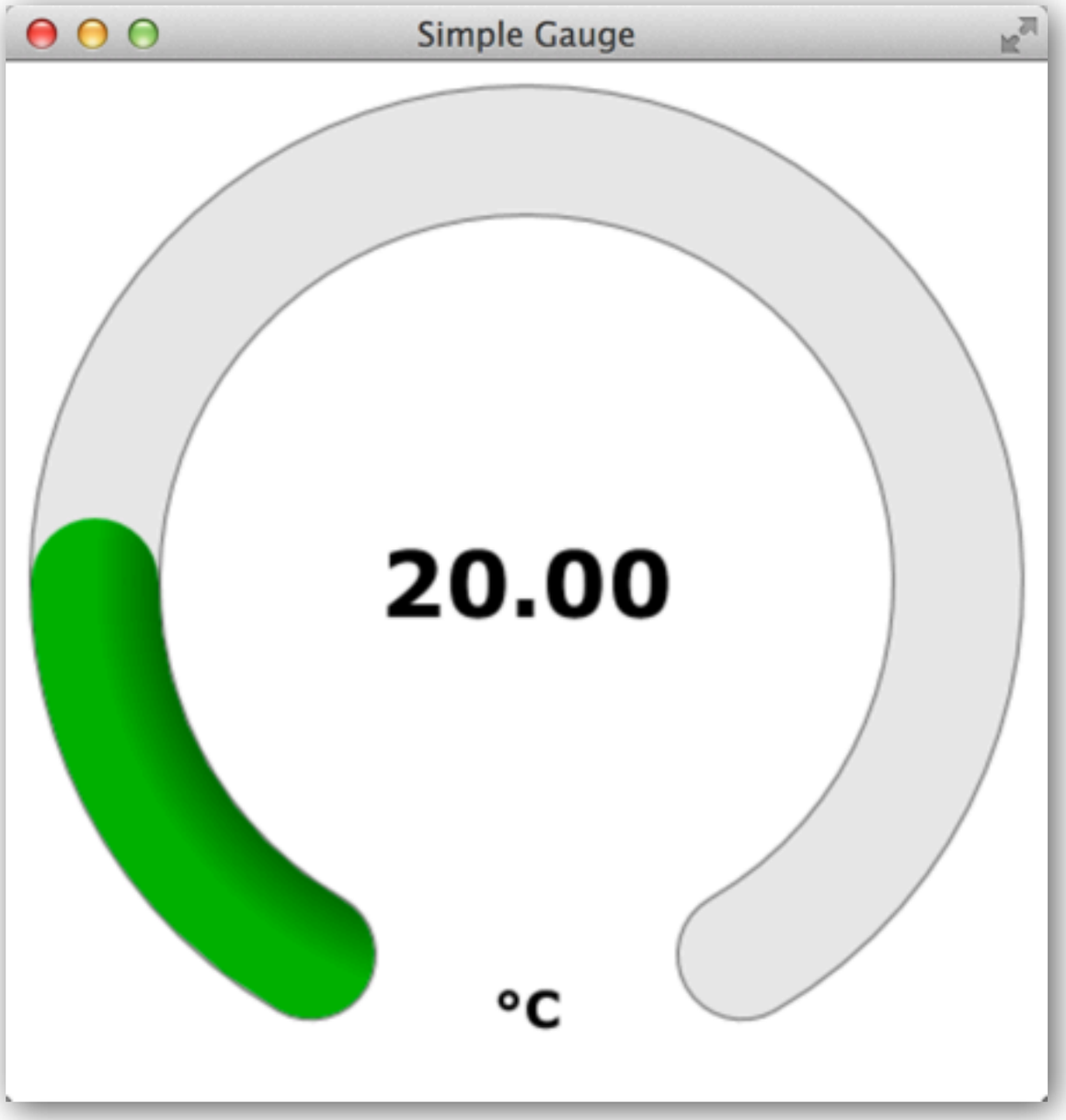


IT'S ABOUT

CONTENT OVER CHROME

CONTENT

CHROME

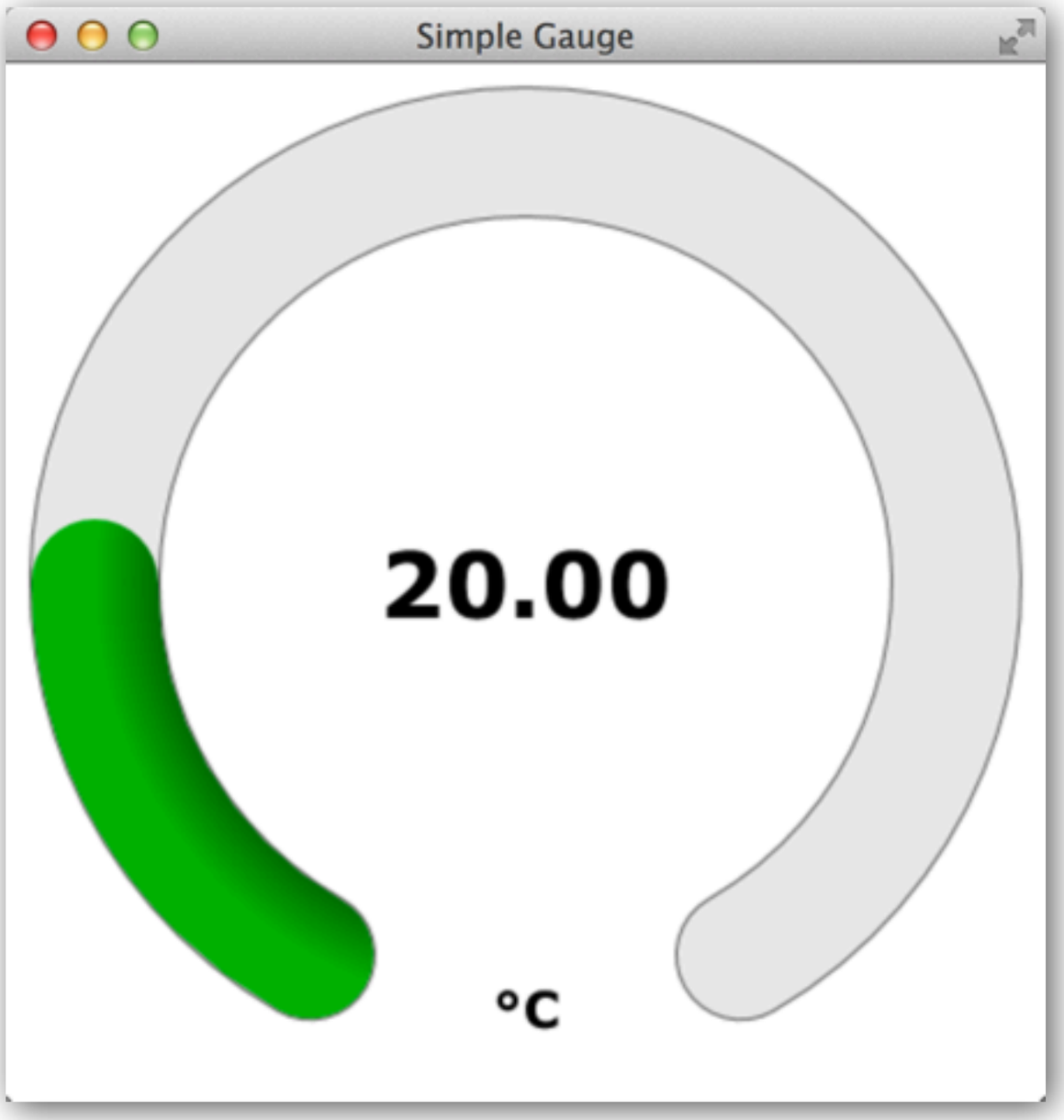


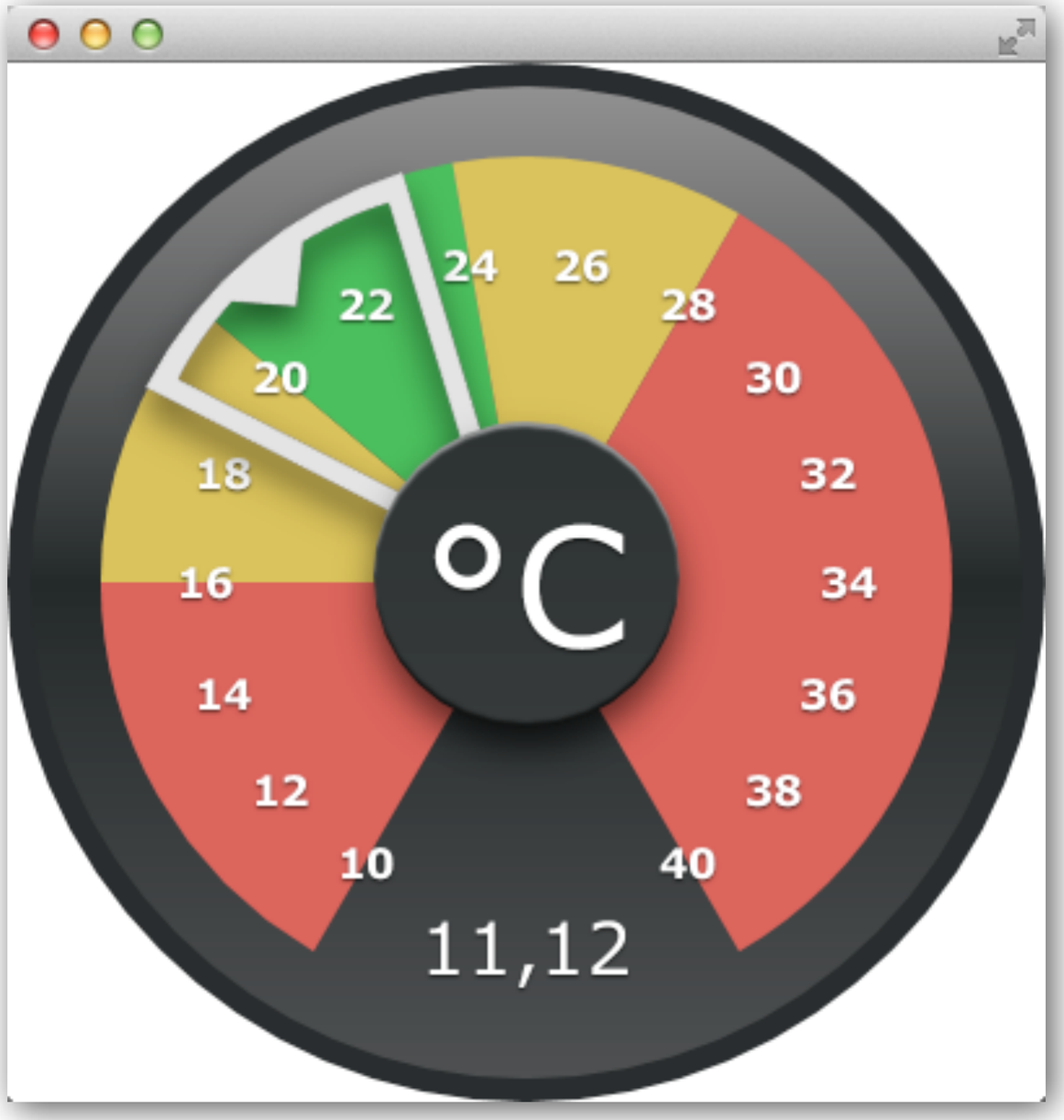
3 Nodes

245 Nodes

CONTENT

COMPROMISE

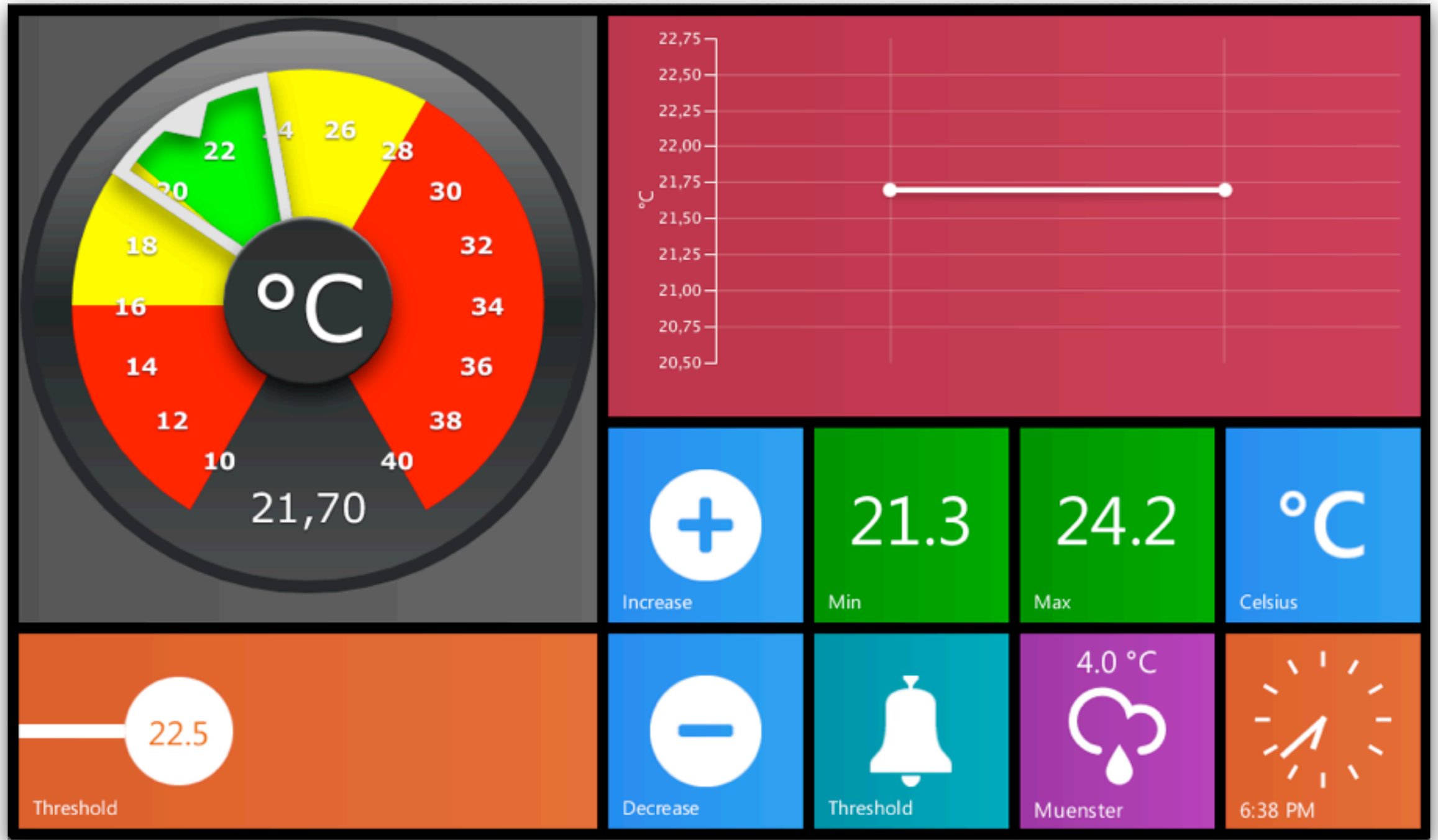




3 Nodes

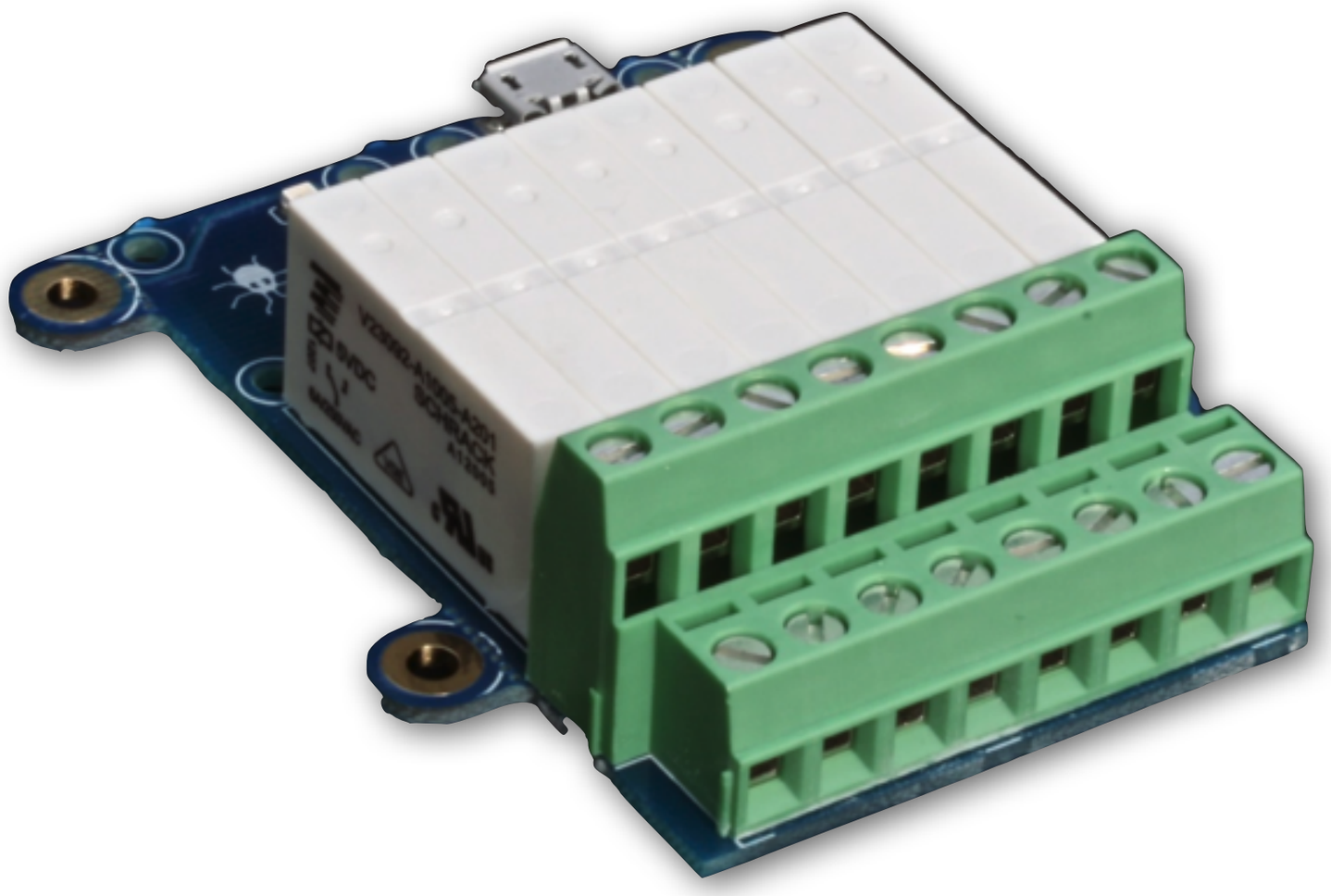
33 Nodes

BeagleBoard xM



USB MaxiRelay

★ Used to control SignalTower



3

Desktop

APPLICATION

running on

JDK7

using JavaFX 2

Requirements

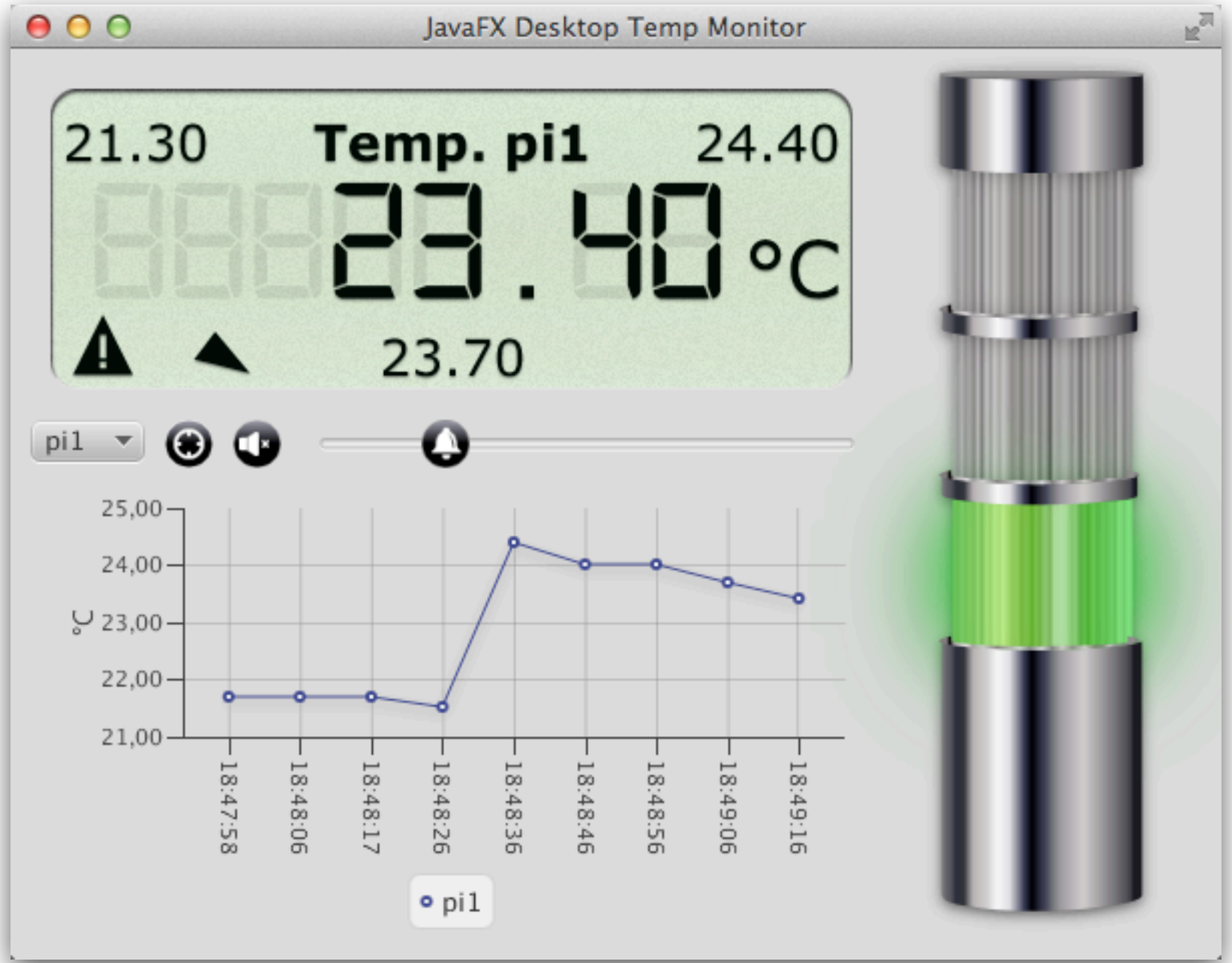
- ★ **Visualize the temperature**
- ★ **No platform dependency**
- ★ **Feedback on current value**
- ★ **Visualize logged data***

*optional

JavaFX Benefits

- Cross platform
- Good graphic support
- Good audio support
- Easy to implement

Desktop Client



ThingM blink(1)

- **Used to indicate the status**

HTML5

4

APPLICATION

running on HTML5

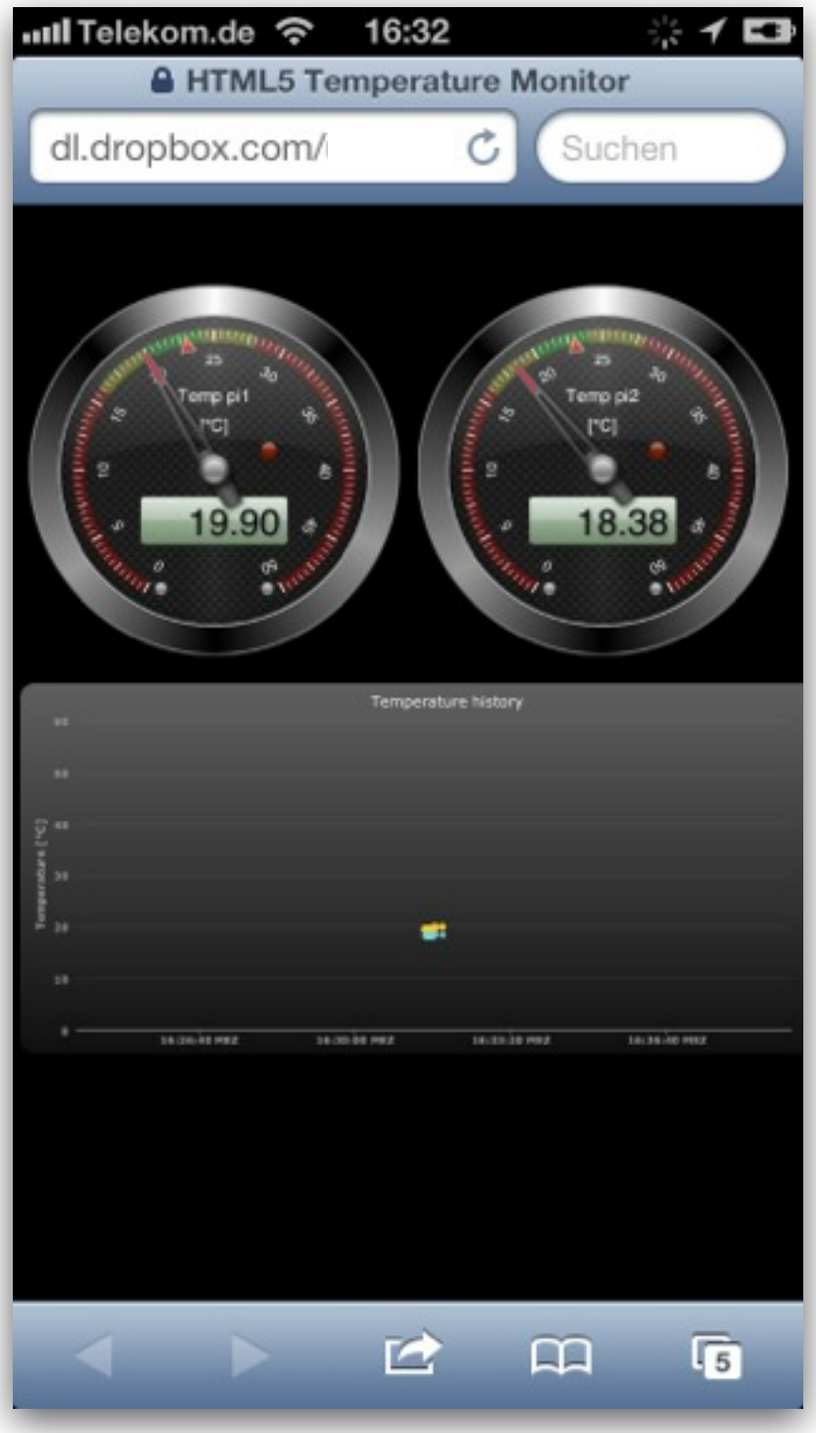
using CANVAS

Requirements

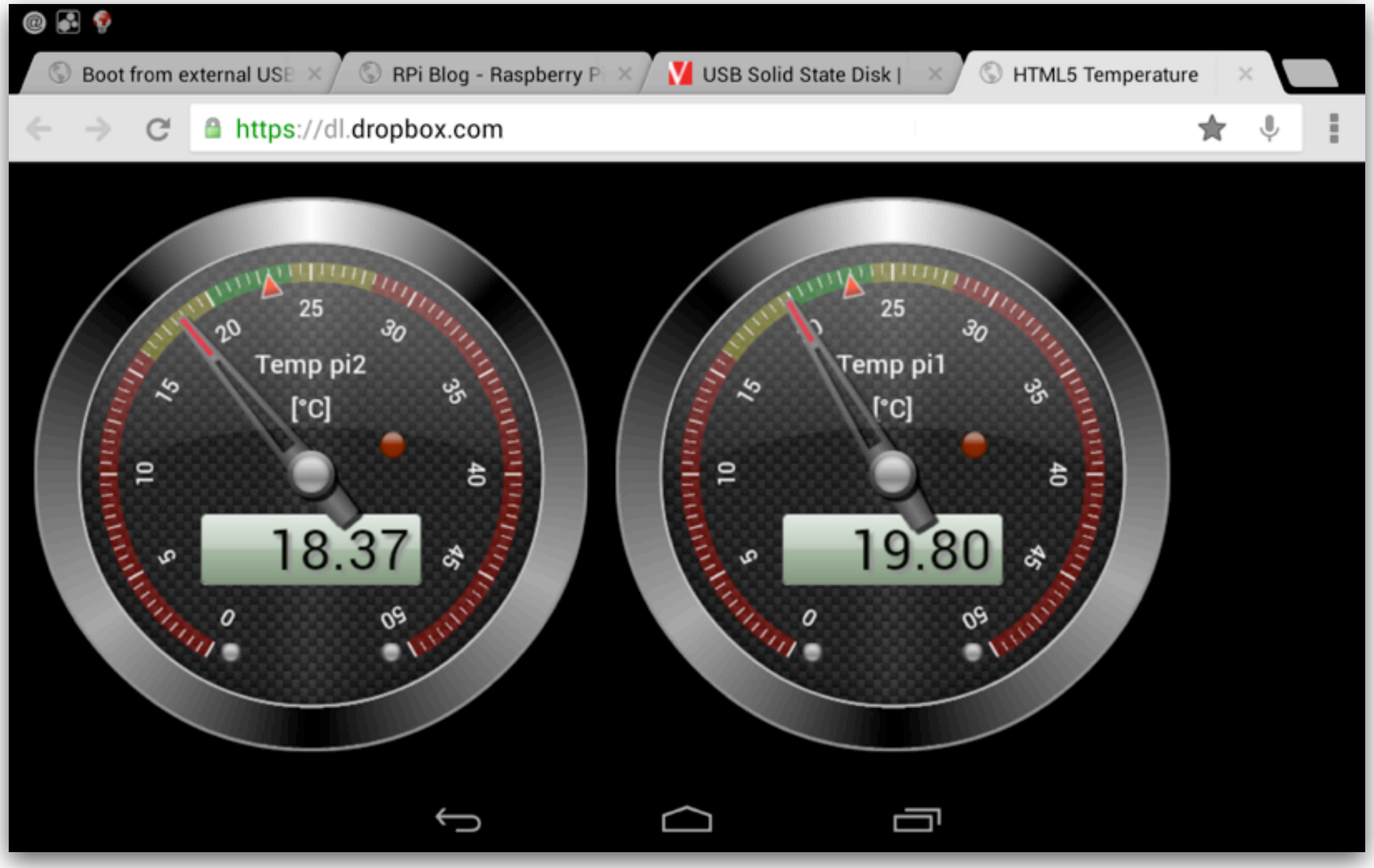
- **Monitor the temperature**

HTML5 on Phone

- Pure JavaScript
- Using SteelSeries
- Using Highcharts



HTML5 on Tablet



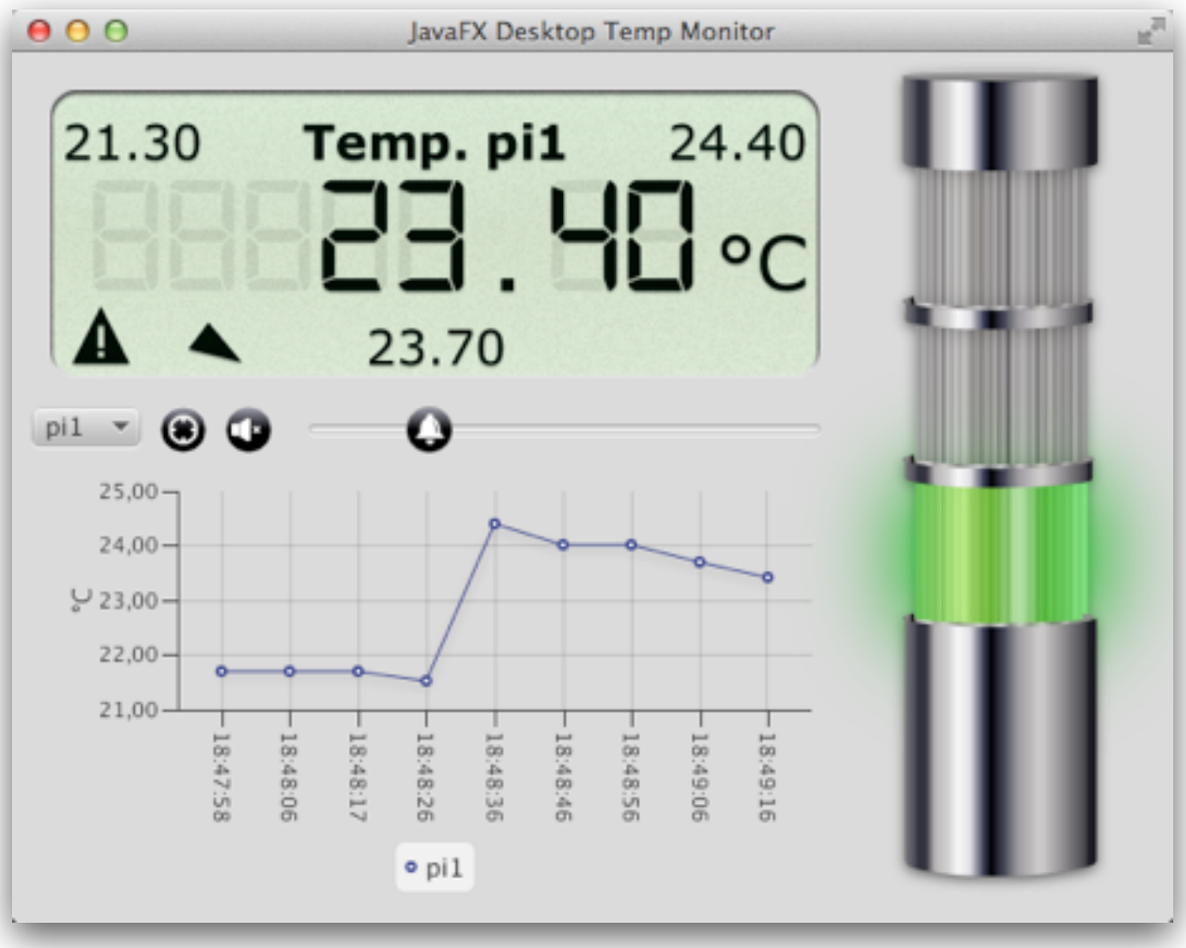
HTML5 on ???

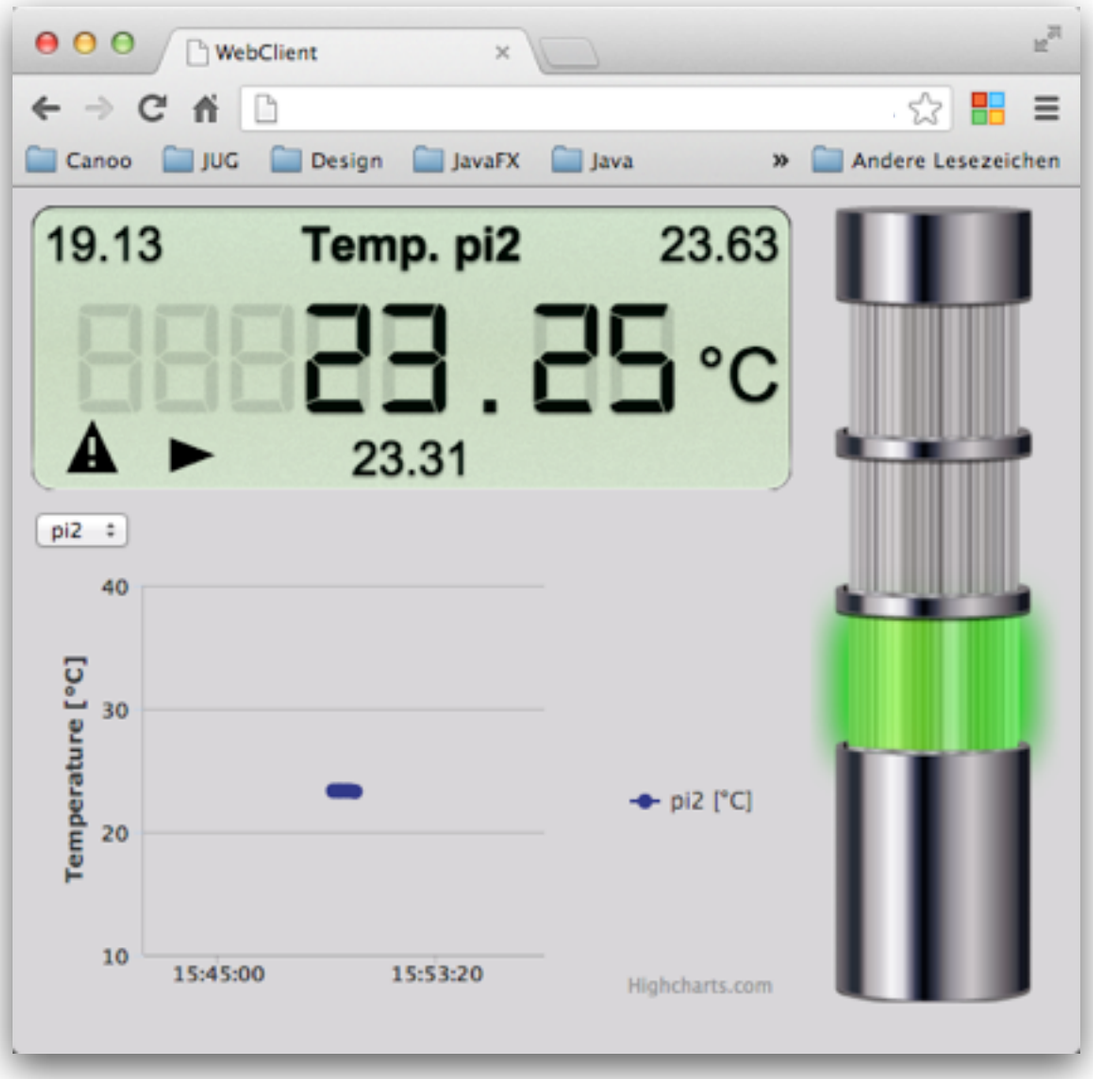

d.dropbox.com
Temp pi1
[°C]
16.30
Temp pi2
[°C]
16.38
START
SELECT
Wii U
MIC
HOME
TV
BATTERY
CONTROL
POWER

And because

we can...

HTML5 Variation





JavaFX

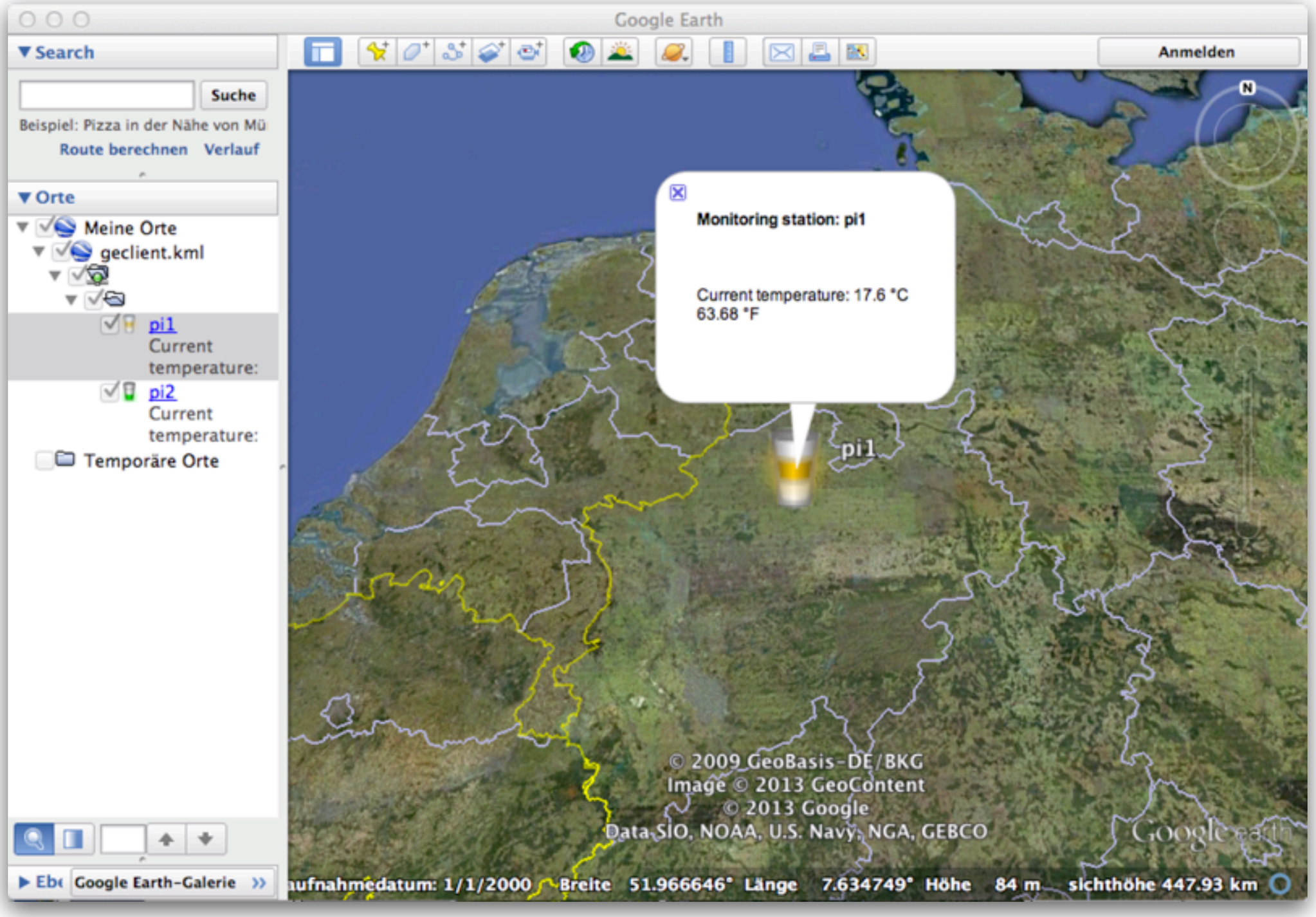
HTML5

5

Just for the

FUN OF IT...

Google Earth client



Conclusion

JAVA(FX) ON EMBEDDED

really rocks...

Demo


keep coding...