

2.– 5. September 2013
in Nürnberg



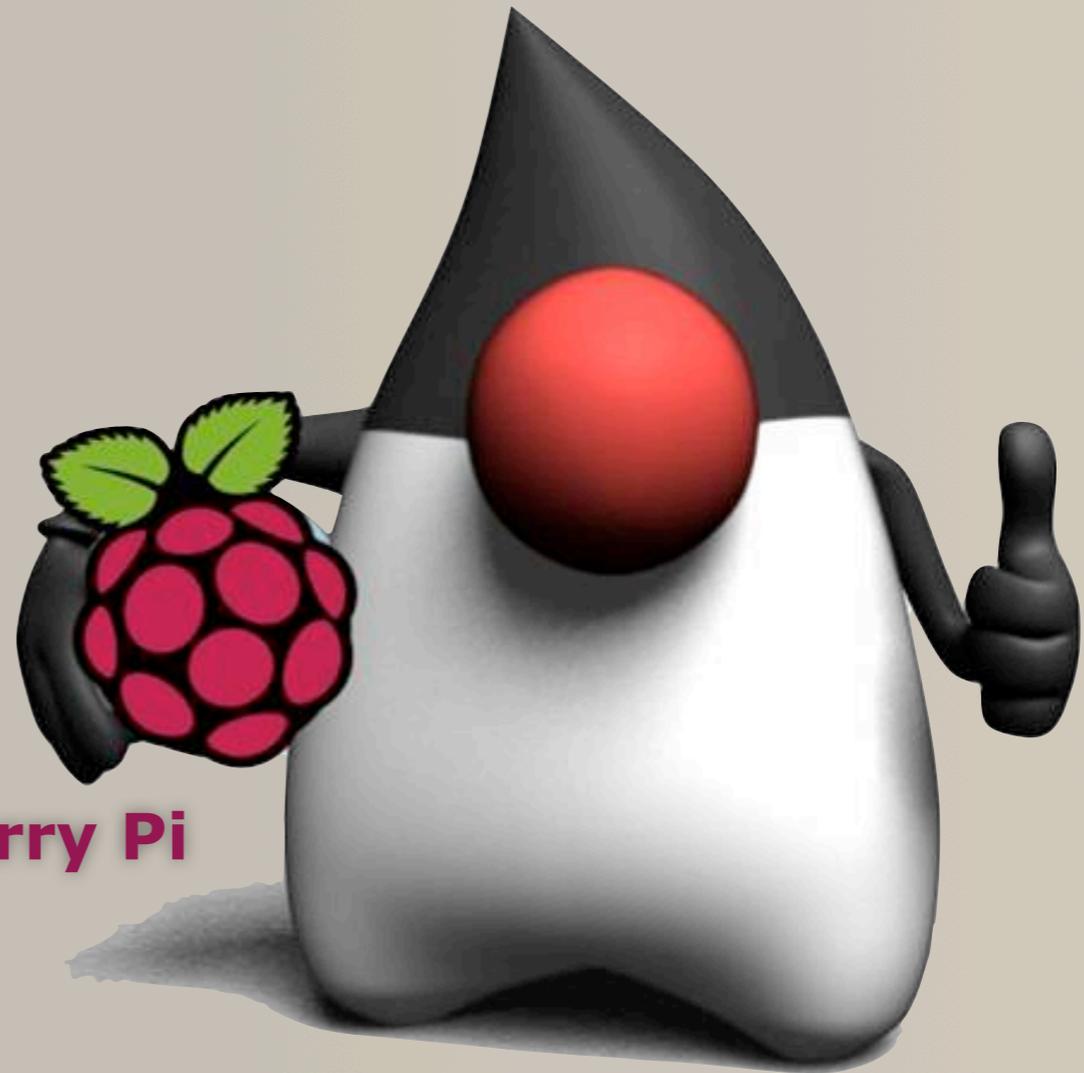
Herbstcampus

Wissenstransfer
par excellence

Himbeeren im Herbst

Embedded Java auf dem Raspberry Pi

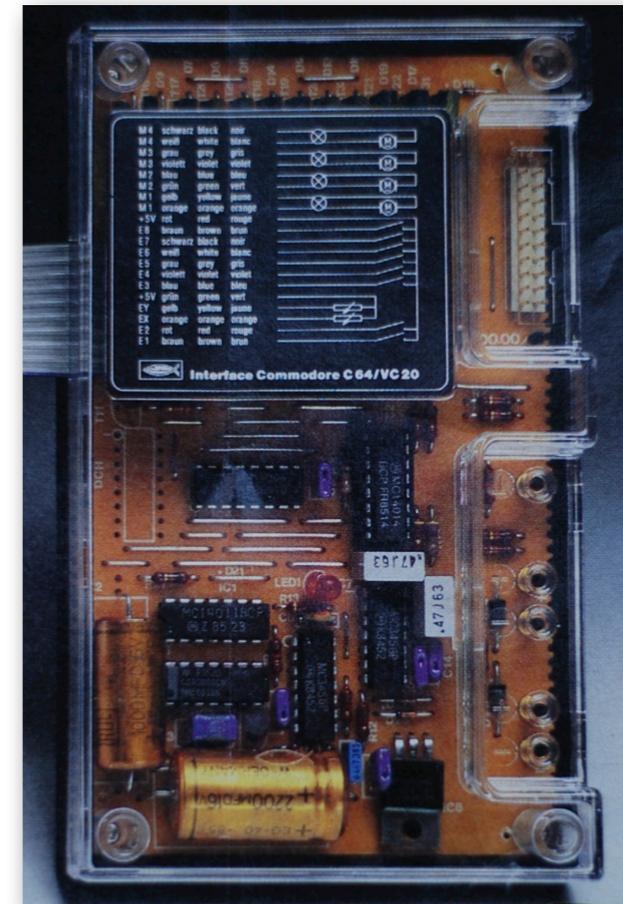
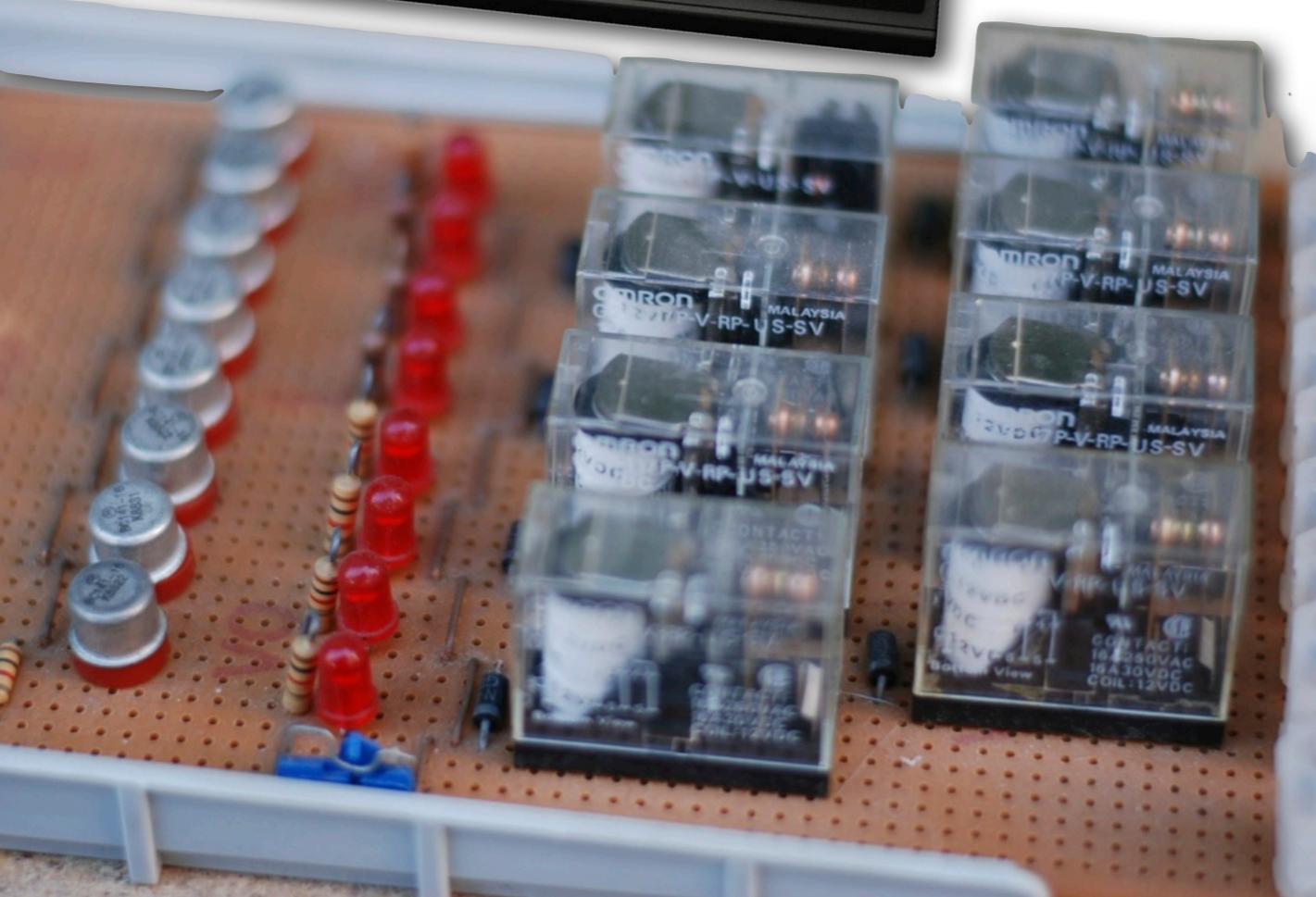
Jens Deters



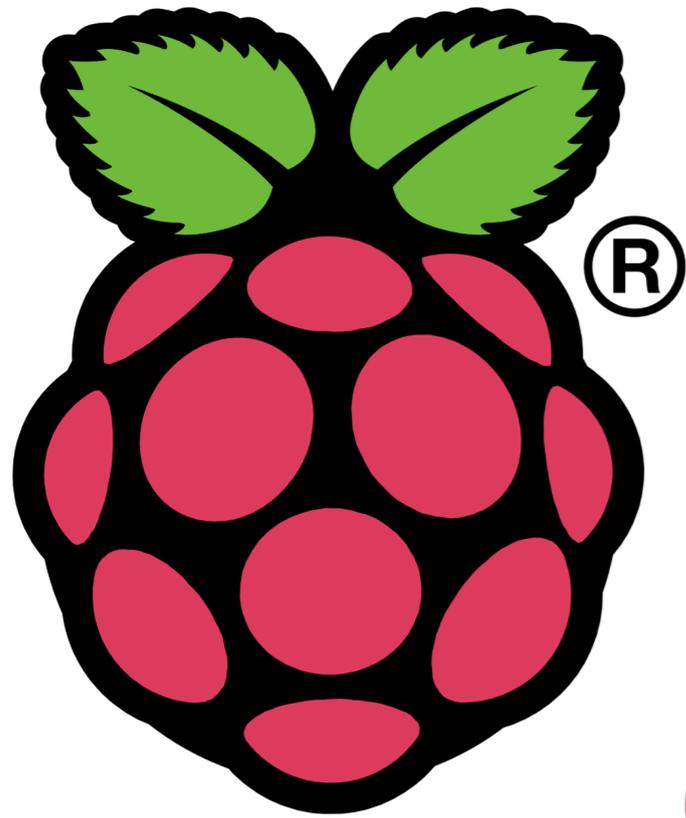
Über mich

- Ostern 1984: Commodore VC20
- 15 Jahre IT
- aktuell:
 - Senior Software Developer bei EMPIC
 - Zielgruppe: zivile und militärische Flugbehörden
 - Mein Schwerpunkt: Web-Client + Rich-Client

Vor 25 Jahren...



Heute: „CreditCard PCs“



pandaboard.org

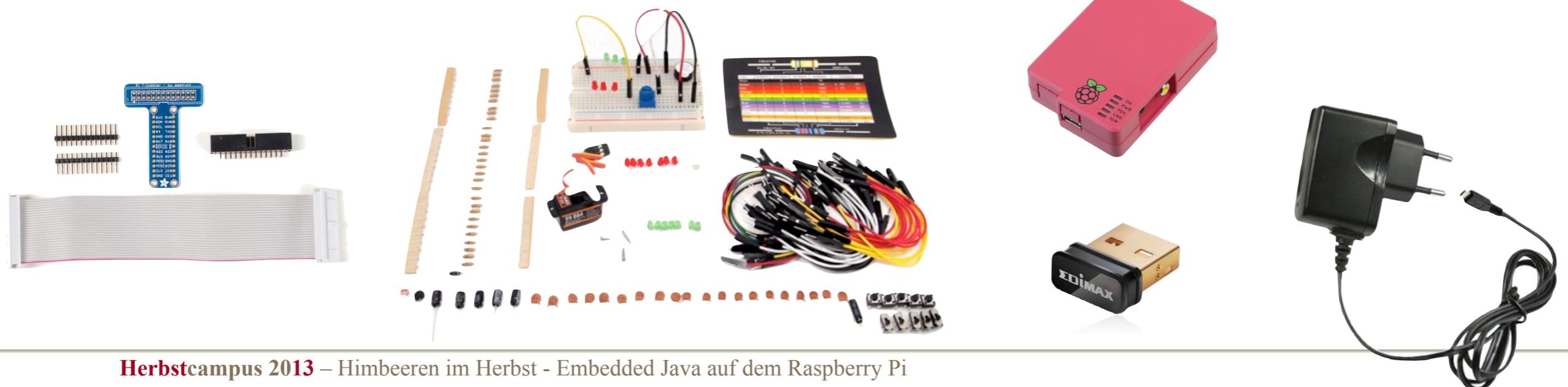
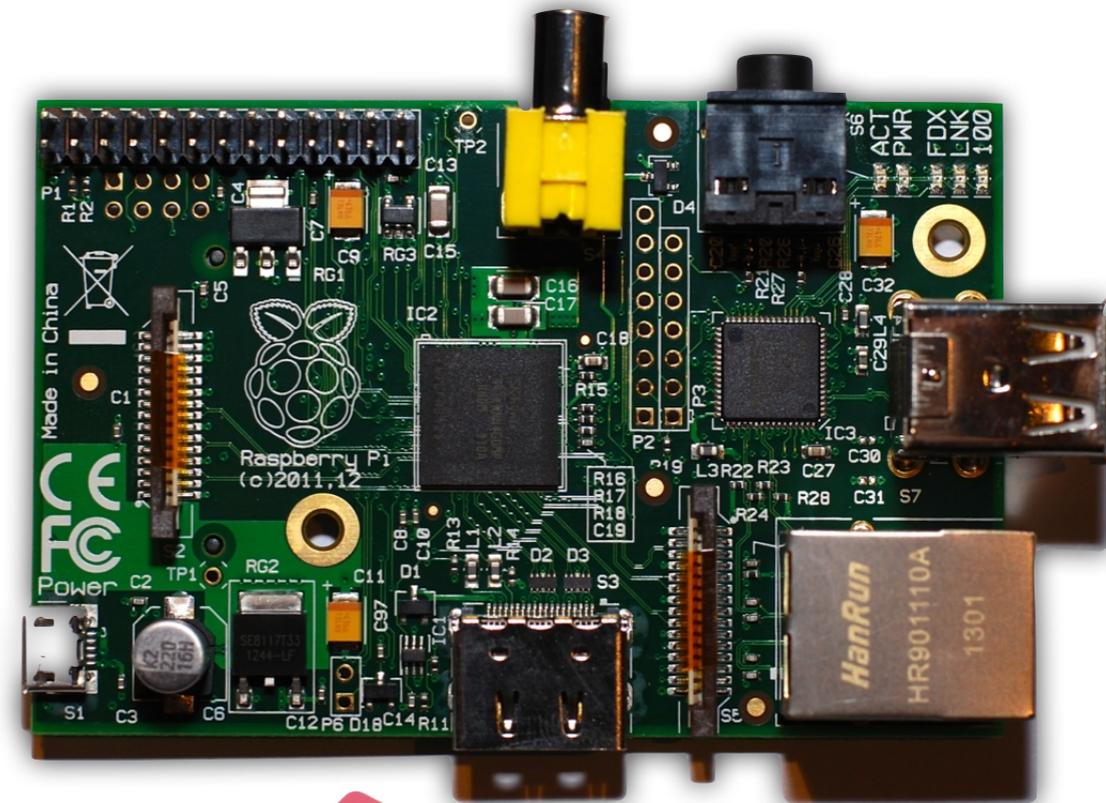


cubieboard

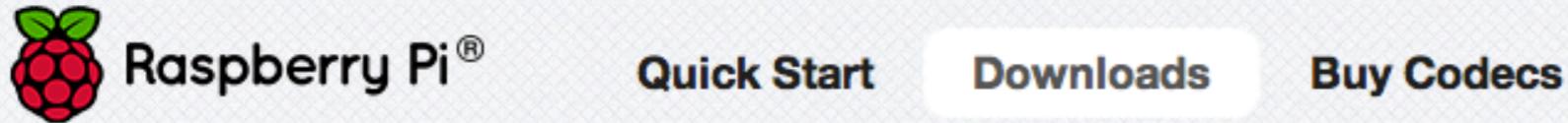
open arm box

Getting started...

RaspberryPi (Model B Revision 2.0)	40 €
Adafruit Pi Cobbler Breakout Kit	7 €
Arduino Sidekick Basic Kit	19 €
Gehäuse	10 €
USB Wireless Adapter	10 €
Netzteil	8 €



SOFTWARE



Raspbian "wheezy"

If you're just starting out, **this is the image we recommend you use**. It's a reference root filesystem from Alex and Dom, based on the [Raspbian](#) optimised version of Debian, and containing LXDE, Midori, development tools and example source code for multimedia functions.

Torrent [2013-07-26-wheezy-raspbian.zip.torrent](#)

JDK 8 Project

Building the next generation of the JDK platform

Download JDK 8

- [JDK 8 snapshot releases](#)
- [Source code](#) (instructions)
- [Early Access Build Test Results](#) (instructions)

JDK 8 Early Access Now Available!

Try it out today!

**Linux ARMv6/7 VFP, 32-bit
HardFP ABI**

Embedded JavaFX

- JDK7 (JavaFX 2.x)
- JDK8 (JavaFX 8)

- Subset von JavaFX
 - kein WebView
 - kein Media support
 - kein Drag & Drop

Raspberry Pi Mod. B Rev 2.0

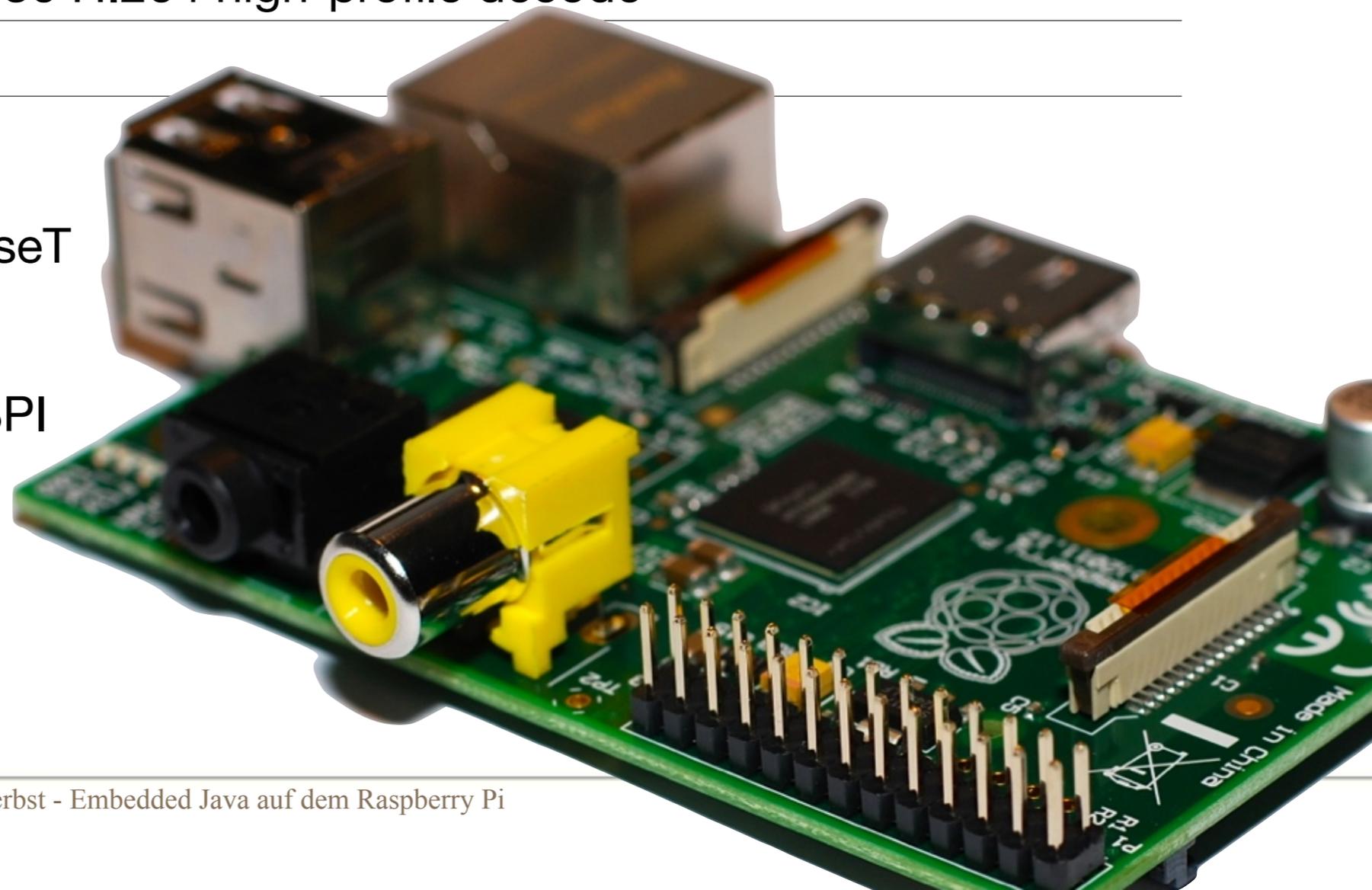
SoC Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port)

CPU ARM v6 mit 700MHz (mit FPU)

GPU OpenGL ES 2.0
Hardwarebeschleunigt:
OpenVG und 1080p30 H.264 high-profile decode

RAM 512 MB (SDRAM)

IO SD card socket
2 x USB 2.0
Ethernet 10/100 BaseT
HDMI, Composite
Audio Out
GPIO, I2O, UART, SPI

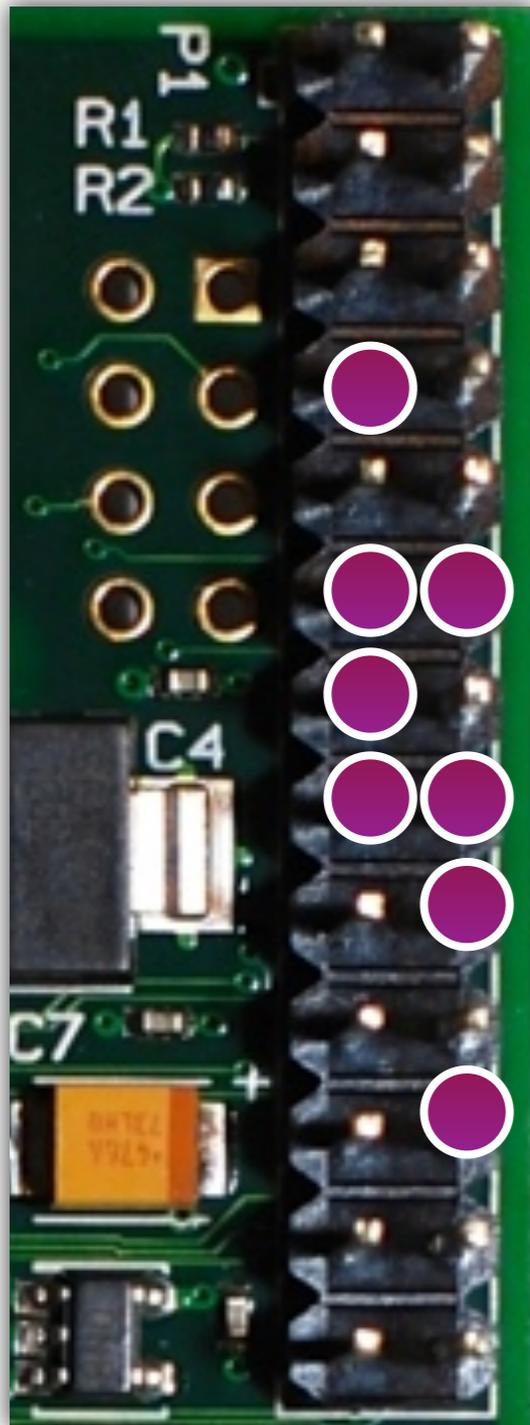


GPIO

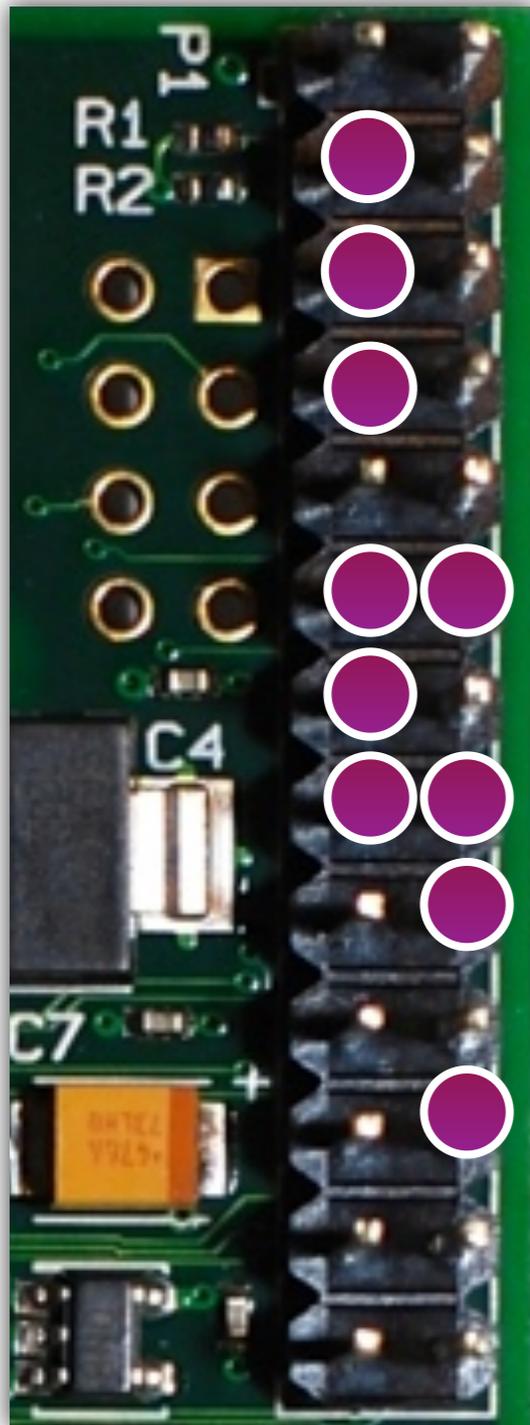
General-purpose input/output





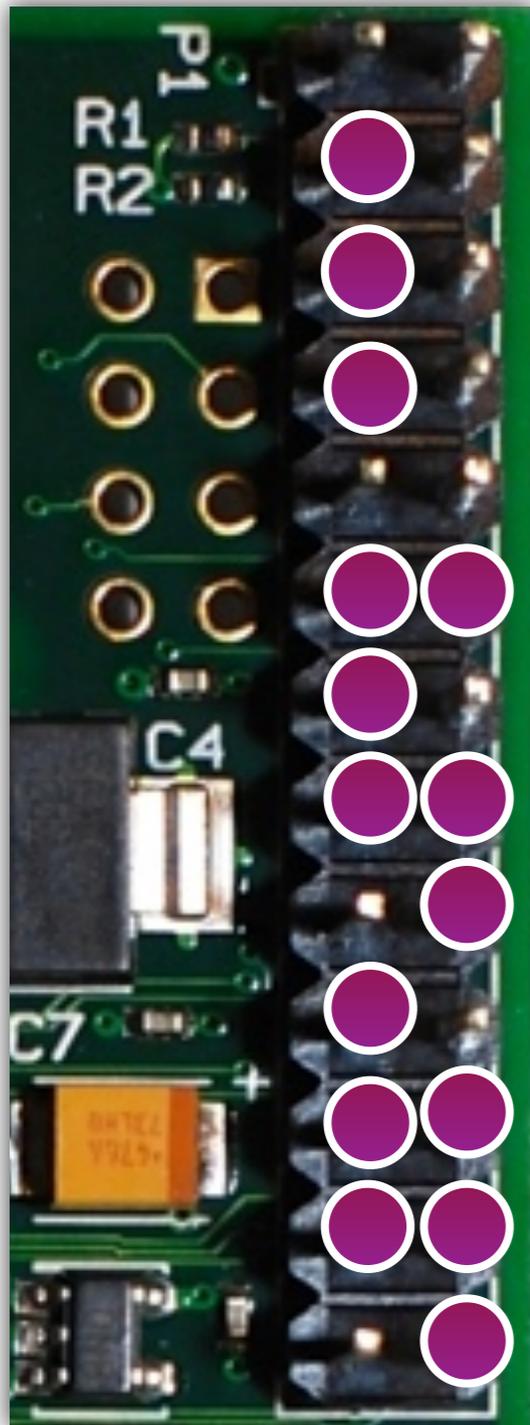


8 Pins GPIO pins



8 Pins GPIO pins

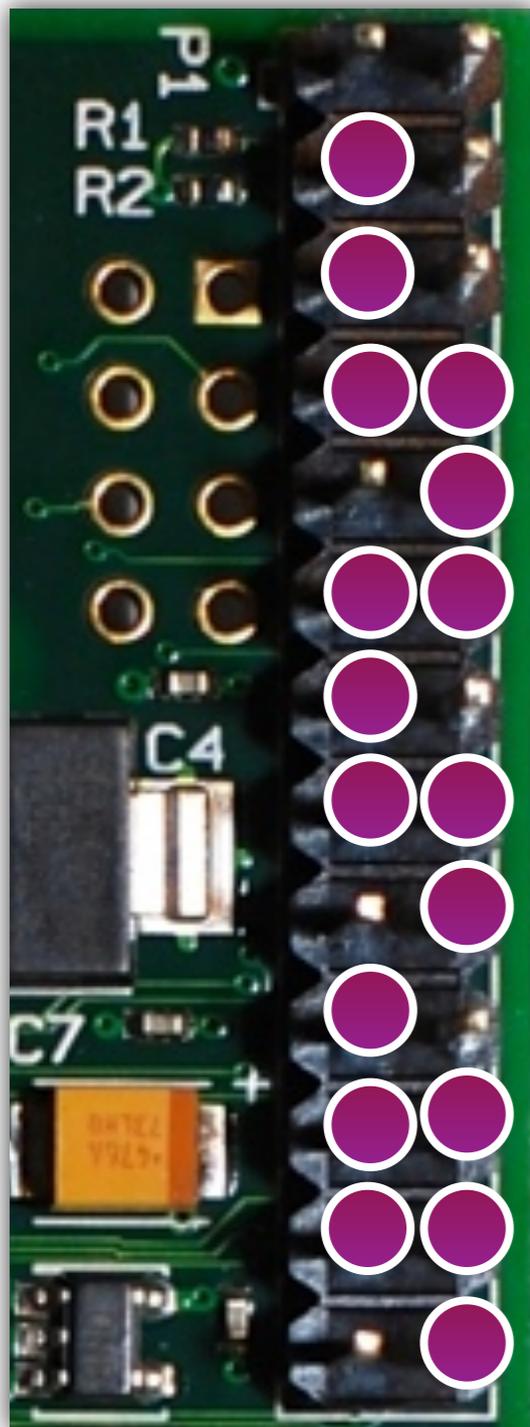
2 Pins I2C interface



8 Pins GPIO pins

2 Pins I2C interface

5 Pins Serial Peripheral Interface

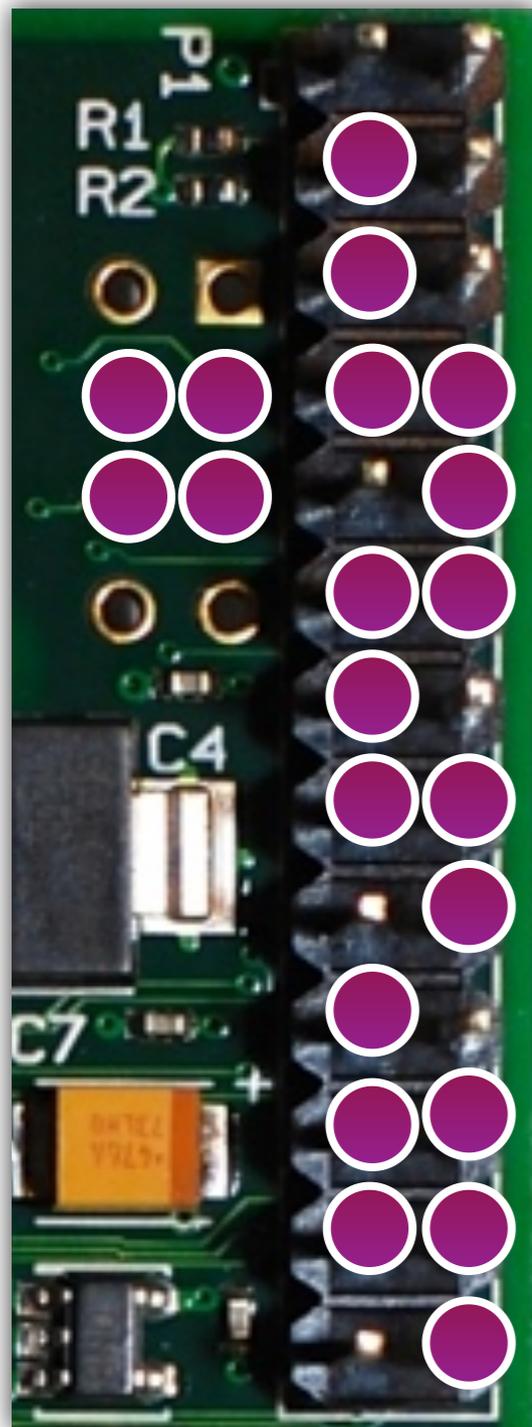


8 Pins GPIO pins

2 Pins I2C interface

5 Pins Serial Peripheral Interface

2 Pins Serial UART



8 Pins GPIO pins

2 Pins I2C interface

5 Pins Serial Peripheral Interface

2 Pins Serial UART

+ 4 weitere GPIOs via P5 Connector (Rev. 2.0)

21 GPIOs

		3.3V	1	2	5V	
I2C	SDA		3	4	5V	
	SCL		5	6	GND	
GPIO	GPIO7		7	8	TxD	UART
	GND		9	10	RxD	
GPIO	GPIO0		11	12	GPIO1	GPIO
	GPIO2		13	14	GND	
	GPIO3		15	16	GPIO4	
		3.3V	17	18	GPIO5	
SPI	MOSI		19	20	GND	GPIO
	MISO		21	22	GPIO6	
	SCLK		23	24	CE0	
		GND	25	26	CE1	

Gordon's Wiring Pi

GPIO Interface library written in C for the Raspberry Pi

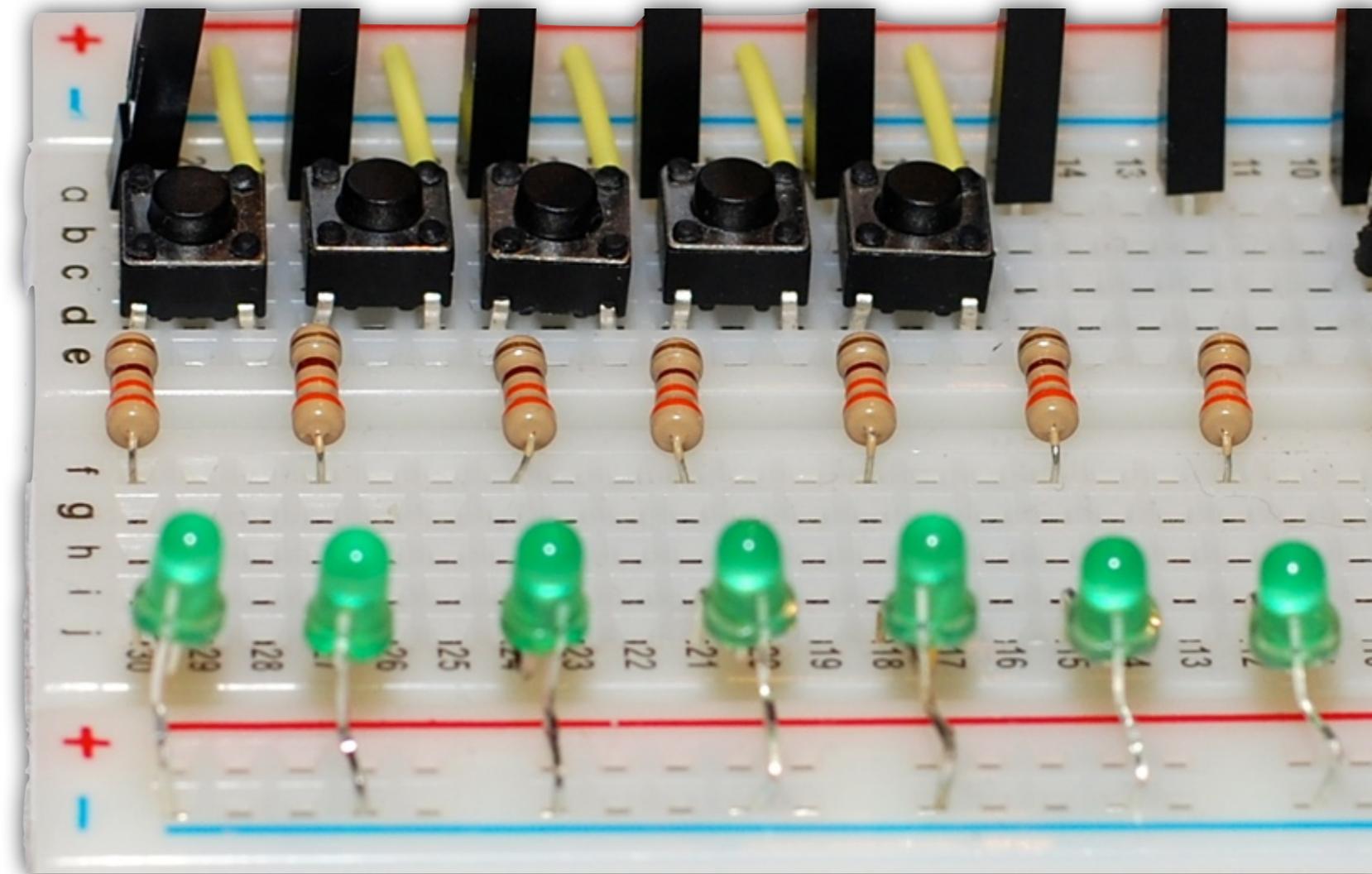


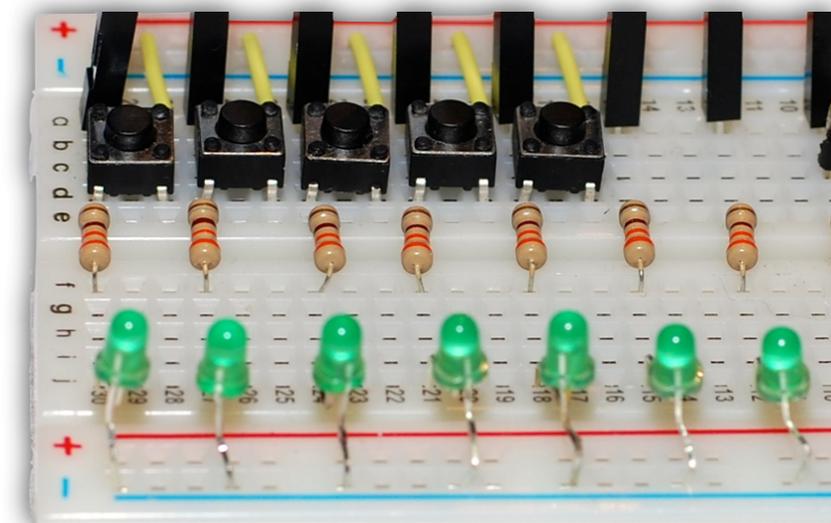
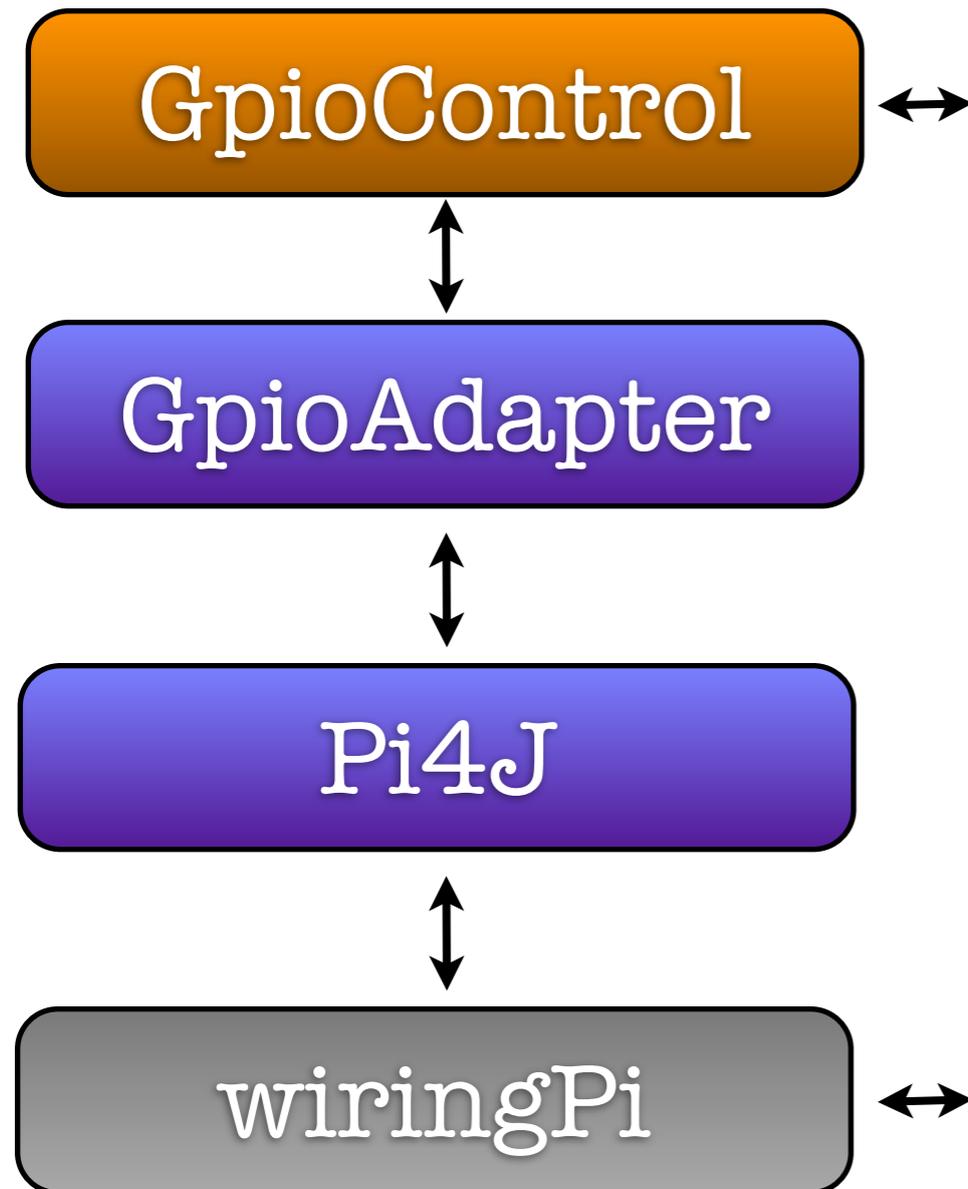
The Pi4J Project

Connecting Java to the Raspberry Pi

GPIO

Controller





GpioAdapter.java

```
private GpioController gpio;
```

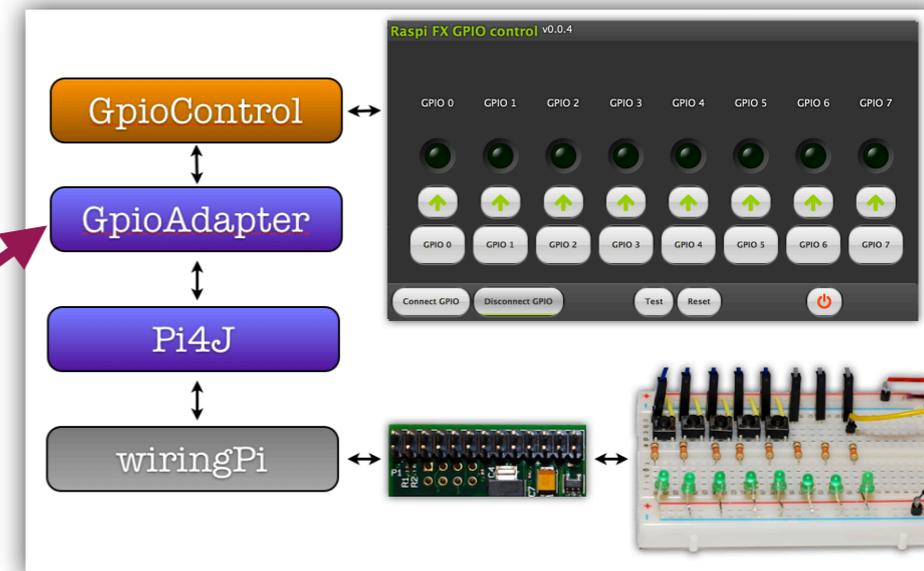
```
private GpioPinDigitalMultipurpose pin0;
```

```
gpio0StateProperty = new SimpleBooleanProperty(Boolean.FALSE);
gpio0ModeProperty = new SimpleObjectProperty<>(PinMode.DIGITAL_OUTPUT);
```

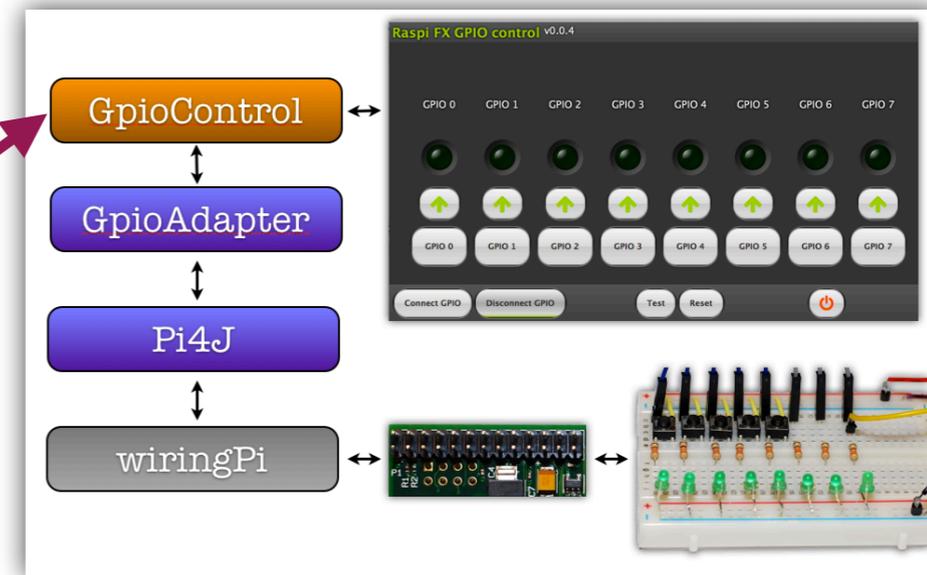
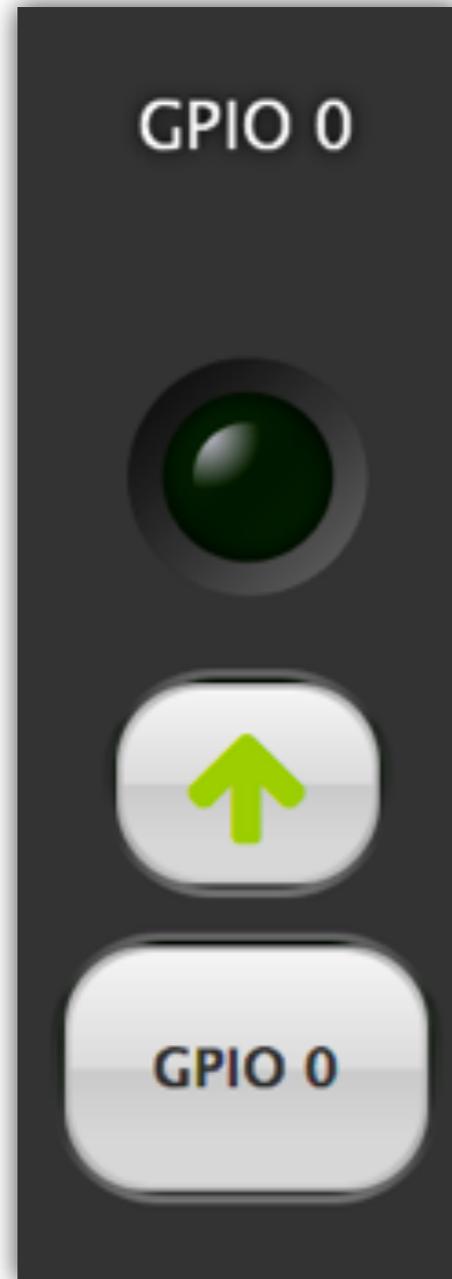
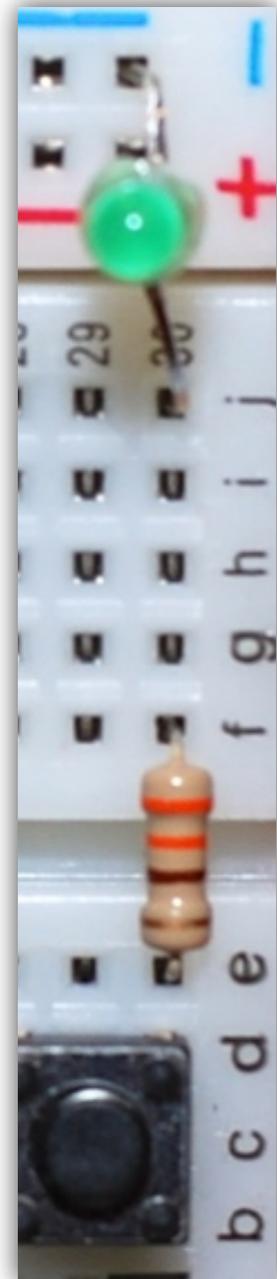
```
gpio = GpioFactory.getInstance();
```

```
pin0 = gpio.provisionDigitalMultipurposePin(RaspiPin.GPIO_00, gpio0ModeProperty.get(), PinPullResistance.PULL_DOWN);
```

```
gpio0StateProperty.addListener(createPinStatePropertyListener(pin0));
addGpioInputListener(pin0, gpio0StateProperty);
```



GpioControl.java

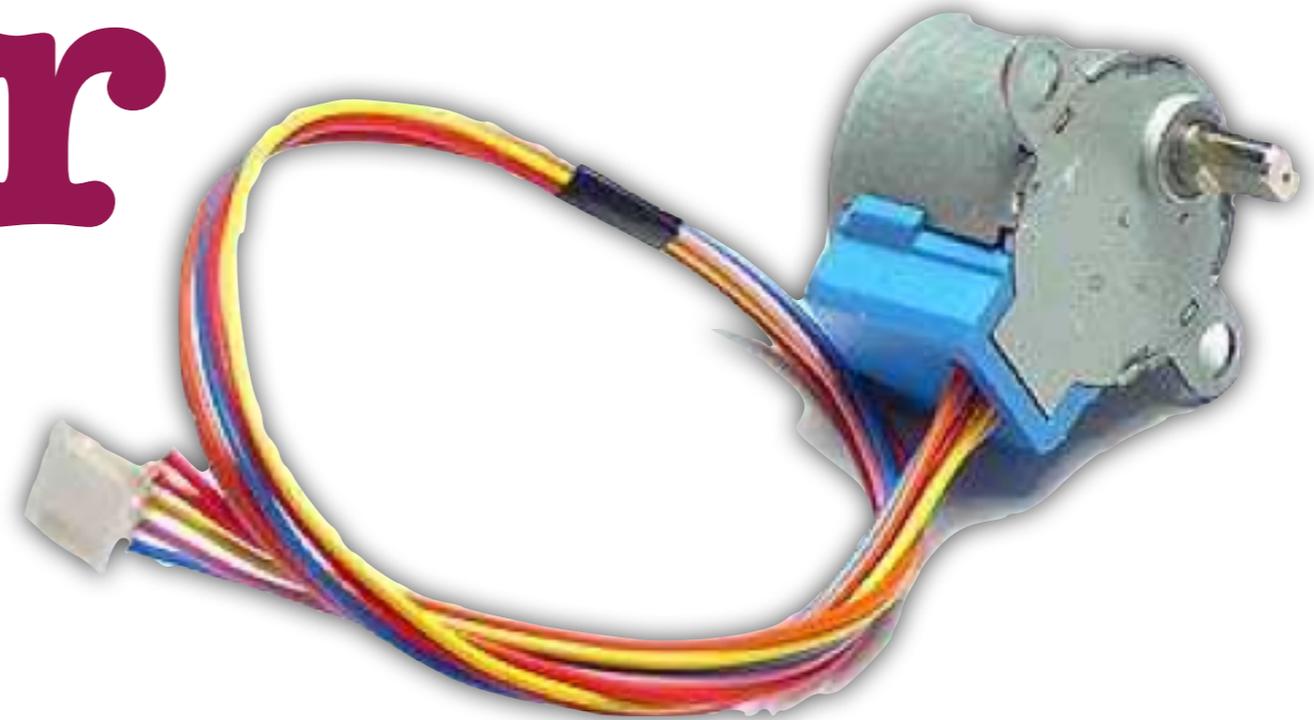


```
indicatorGPIO0.onProperty().
    bindBidirectional(gpioAdapter.gpio0StateProperty());
```

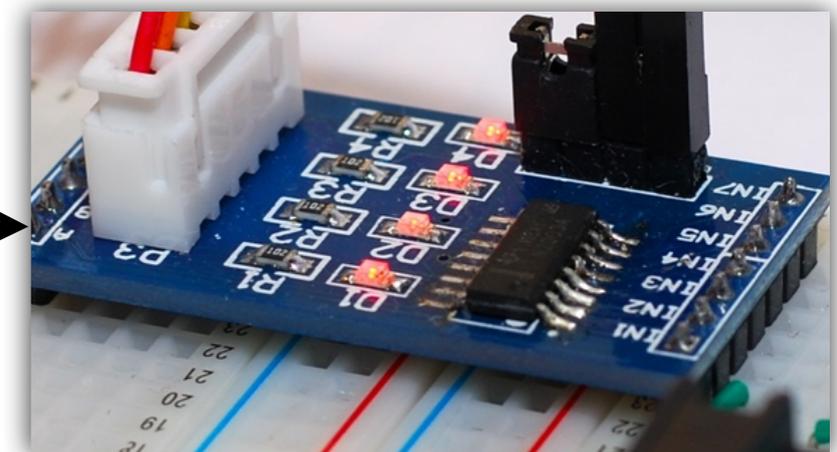
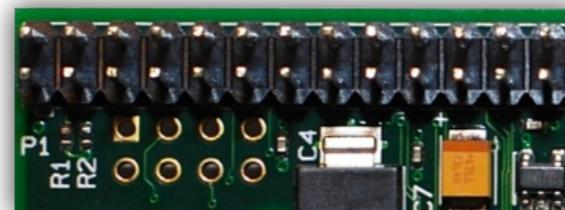
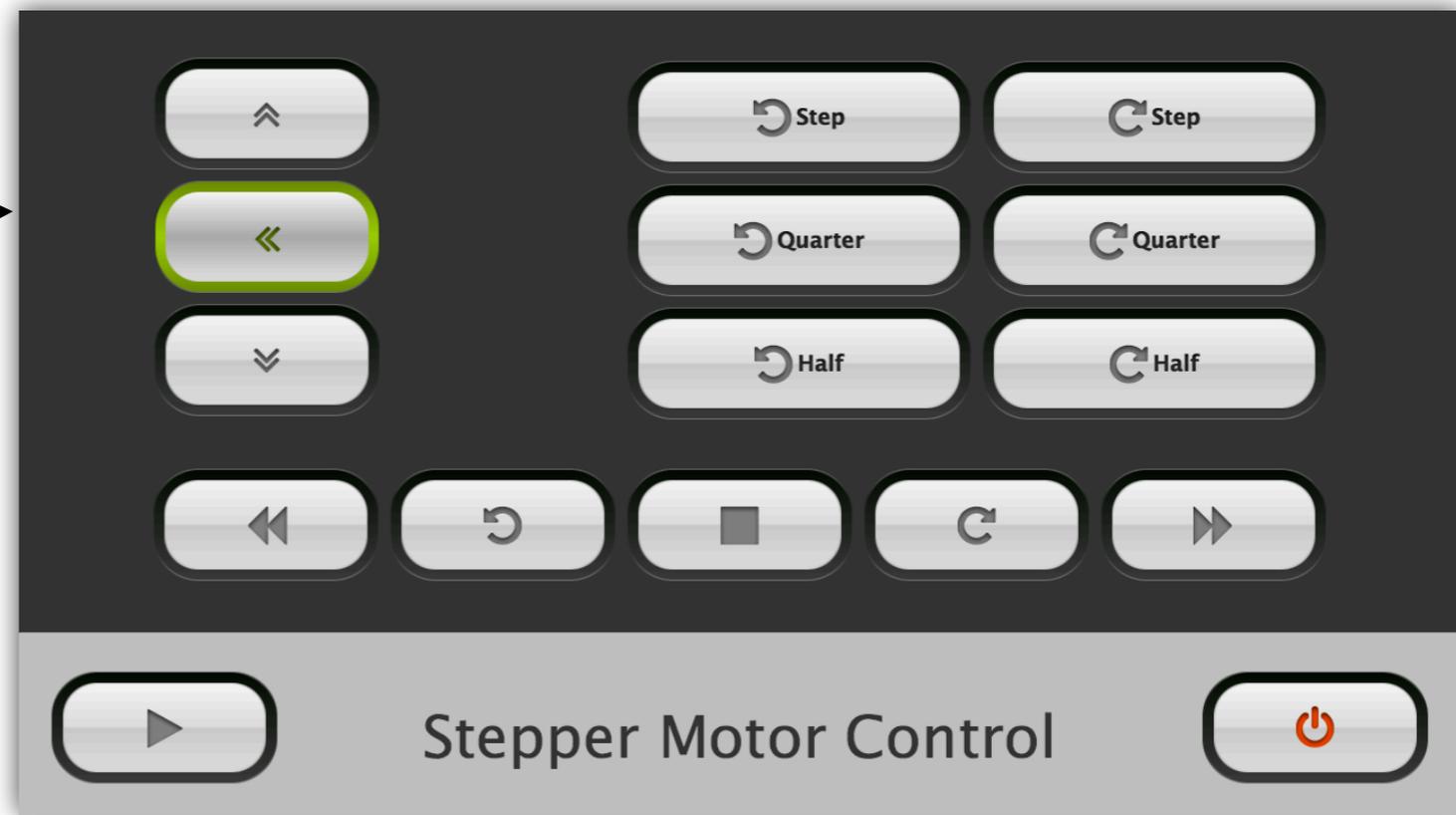
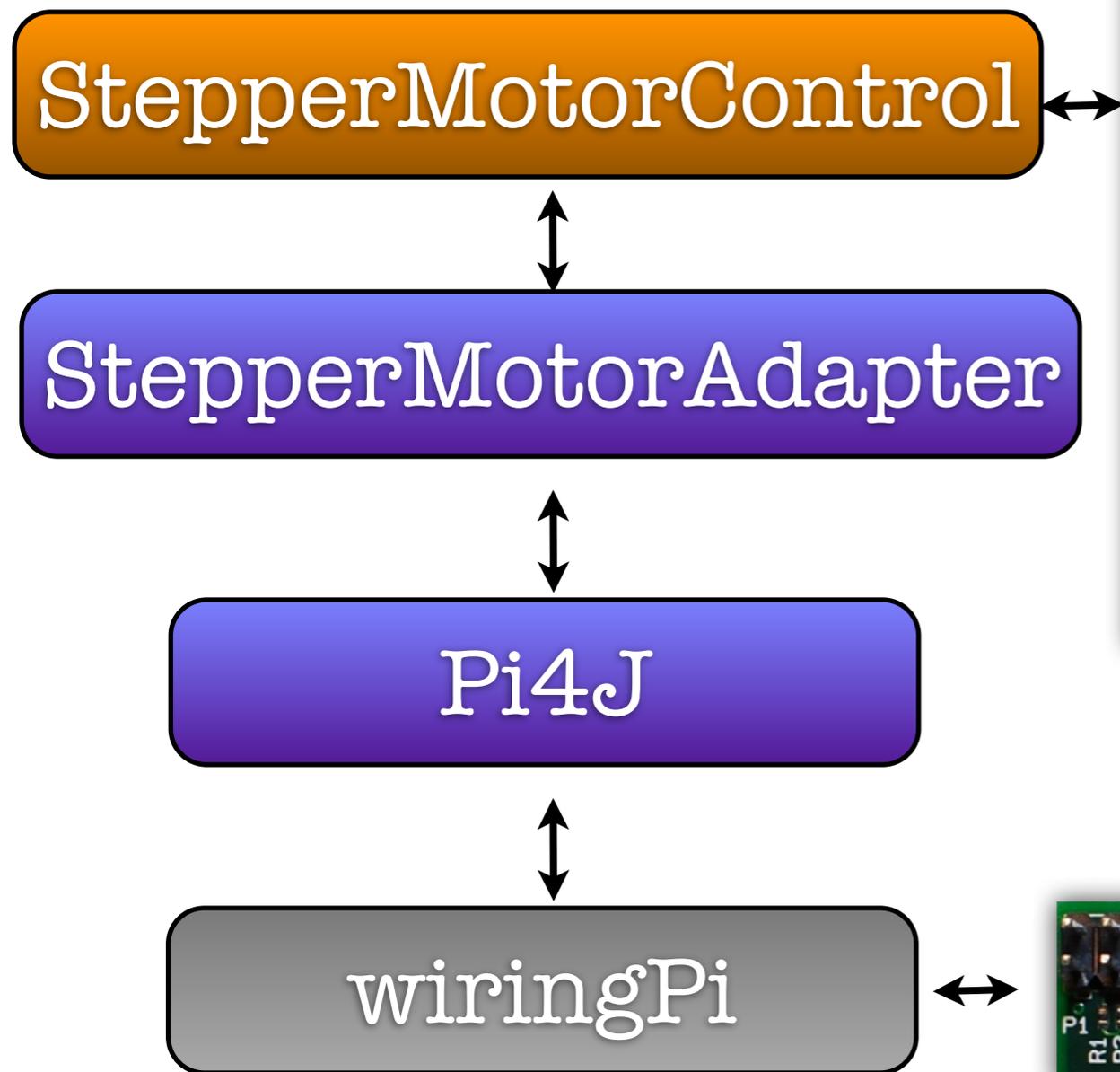
```
toggleModeGPIO0.setOnAction(new ToggleModeEventHandler(toggleModeGPIO0, 0));
gpioAdapter.gpio0ModeProperty()
    .addListener(new IOModeChangeEventHandler(toggleModeGPIO0));
```

```
toggleGPIO0.selectedProperty().
    bindBidirectional(gpioAdapter.gpio0StateProperty());
toggleGPIO0.visibleProperty().
    bind(toggleModeGPIO0.selectedProperty().
        not());
```

Stepper Motor



Controller



StepperMotorAdapter.java

```
private final int oneRevolution = 2038;
private final int quarterRevolution = oneRevolution / 4;
private final int halfRevolution = oneRevolution / 2;
private final int oneDegreeRevolution = oneRevolution / 360;
```

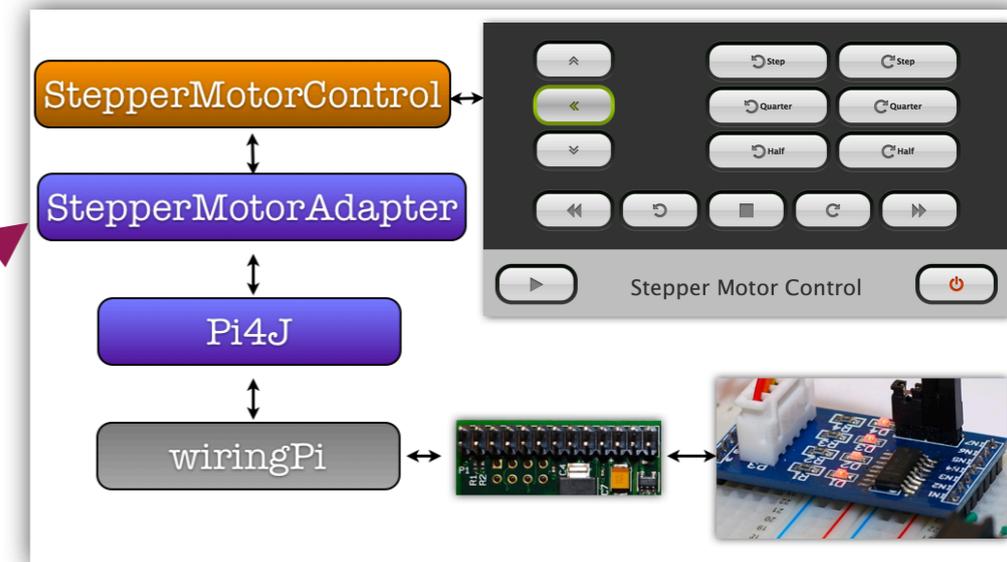
```
singleStepSequence = new byte[4];
singleStepSequence[0] = (byte) 0b0001;
singleStepSequence[1] = (byte) 0b0010;
singleStepSequence[2] = (byte) 0b0100;
singleStepSequence[3] = (byte) 0b1000;
```

```
public void connect() {
    LOGGER.info("connect...");
    gpio = GpioFactory.getInstance();
    final GpioPinDigitalOutput[] pins = {
        gpio.provisionDigitalOutputPin(RaspiPin.GPIO_00, PinState.LOW),
        gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, PinState.LOW),
        gpio.provisionDigitalOutputPin(RaspiPin.GPIO_02, PinState.LOW),
        gpio.provisionDigitalOutputPin(RaspiPin.GPIO_03, PinState.LOW)};

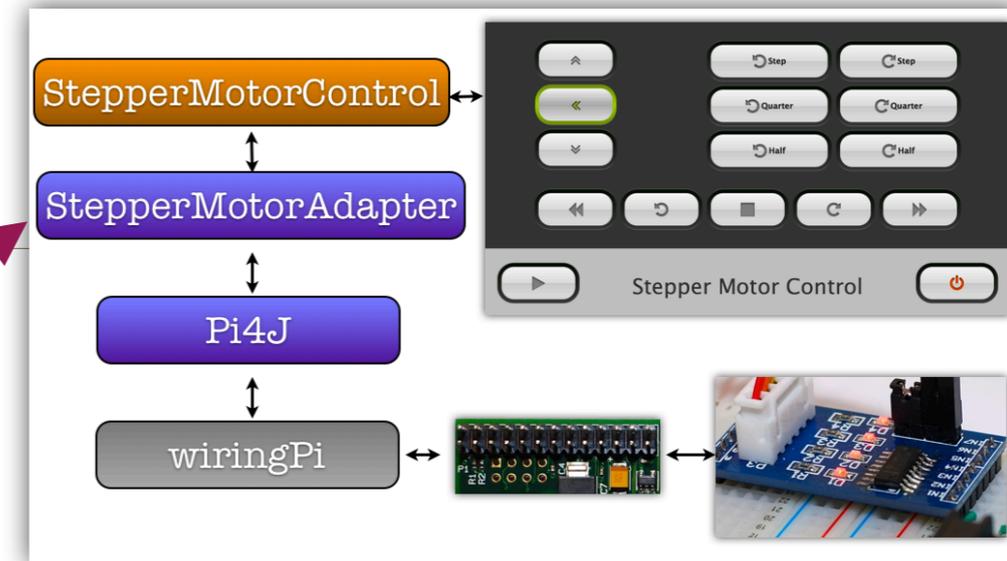
    gpio.setShutdownOptions(true, PinState.LOW, pins);

    motor = new GpioStepperMotorComponent(pins);
    motor.setStepInterval(2);
    motor.setStepSequence(singleStepSequence);
    motor.setStepsPerRevolution(oneRevolution);

    connectedProperty.setValue(Boolean.TRUE);
}
}
```



StepperMotorAdapter.java



```

public void forward() {
    motor.reverse();
}

public void backward() {
    motor.forward();
}

public void stop() {
    motor.stop();
}
  
```

```

public void halfRevolutionBackward() {
    motor.step(halfRevolution);
}

public void halfRevolutionForward() {
    motor.step(-halfRevolution);
}
  
```

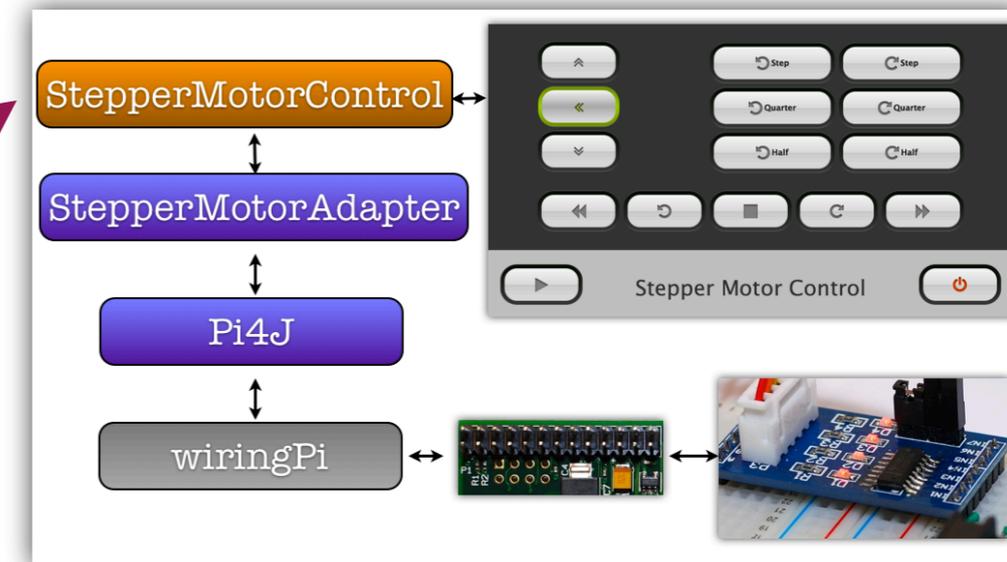
StepperMotorControl.java

Feinjustierung

```

adjustBackwardButton.setOnMousePressed(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent t) {
        stepperMotorAdapter.backward();
    }
});
adjustBackwardButton.setOnMouseReleased(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent t) {
        stepperMotorAdapter.stop();
    }
});
adjustForwardButton.setOnMousePressed(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent t) {
        stepperMotorAdapter.forward();
    }
});
adjustForwardButton.setOnMouseReleased(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent t) {
        stepperMotorAdapter.stop();
    }
});

```



DIE IMMO

Increase WAF...

Wife Acceptance Factor,

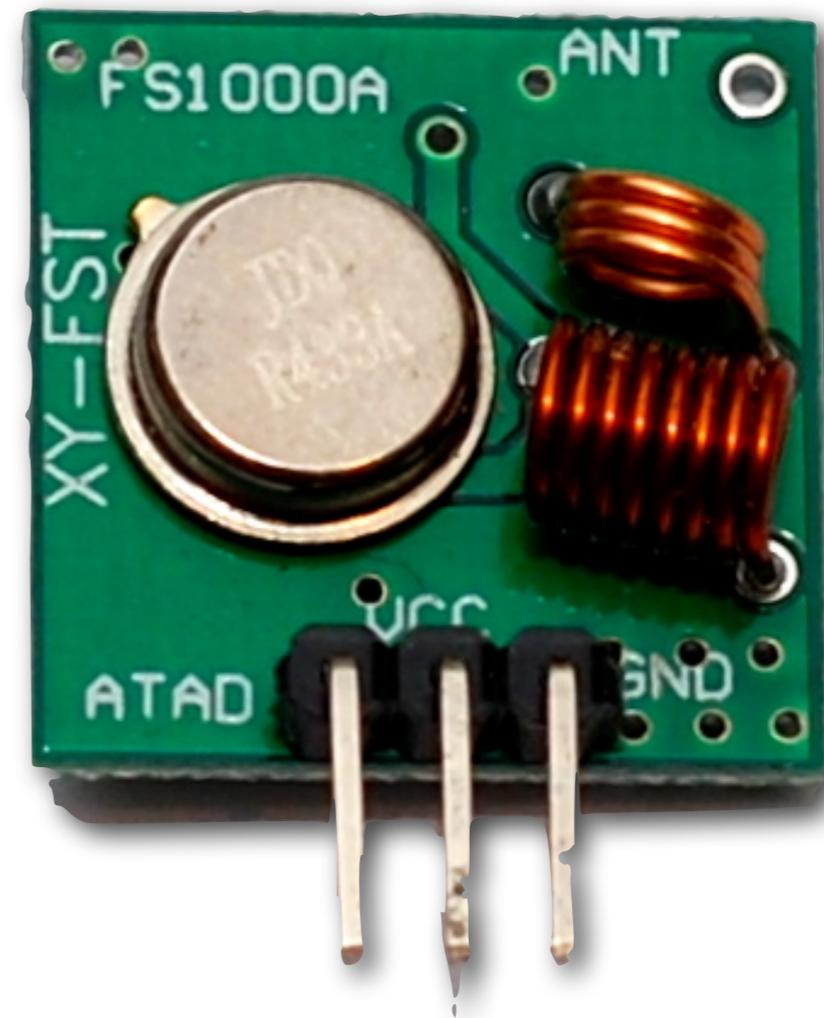
Wife Approval Factor, or

Wife Appeal Factor[1] (WAF),

are design elements that increase the likelihood a wife will approve the purchase of expensive

consumer electronics products such as high-fidelity loudspeakers, home theater systems and personal computers*.

Home! Sweet Home!





Licht Haustür



Licht Terrasse



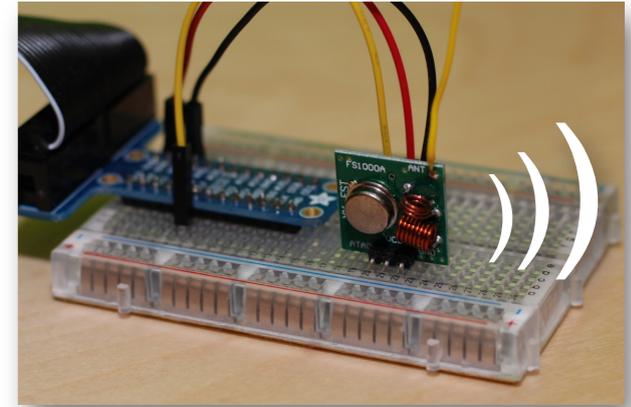
Springbrunnen



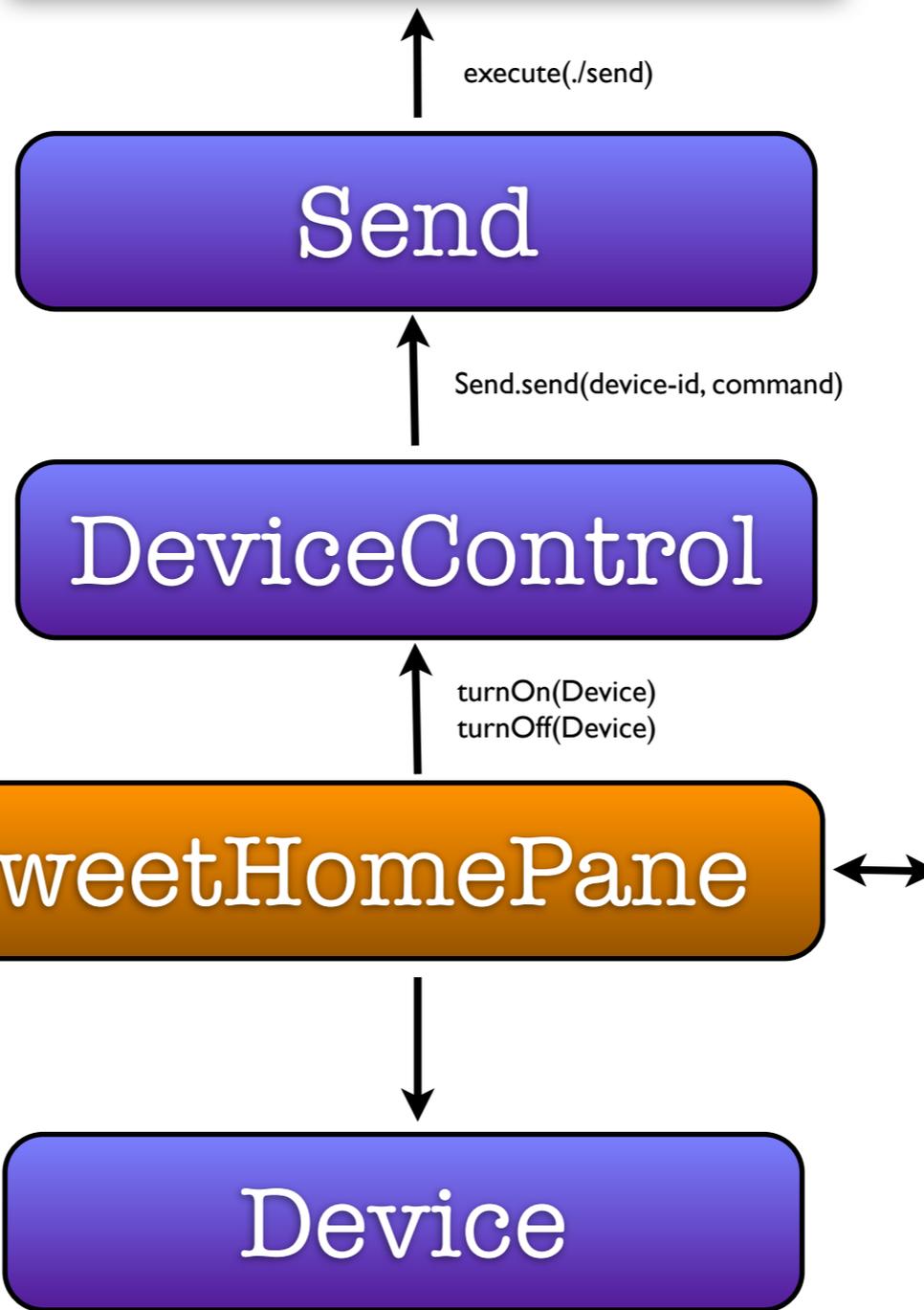
Talk to
192.168.0.61

SweetHomeFX
v1.0.1





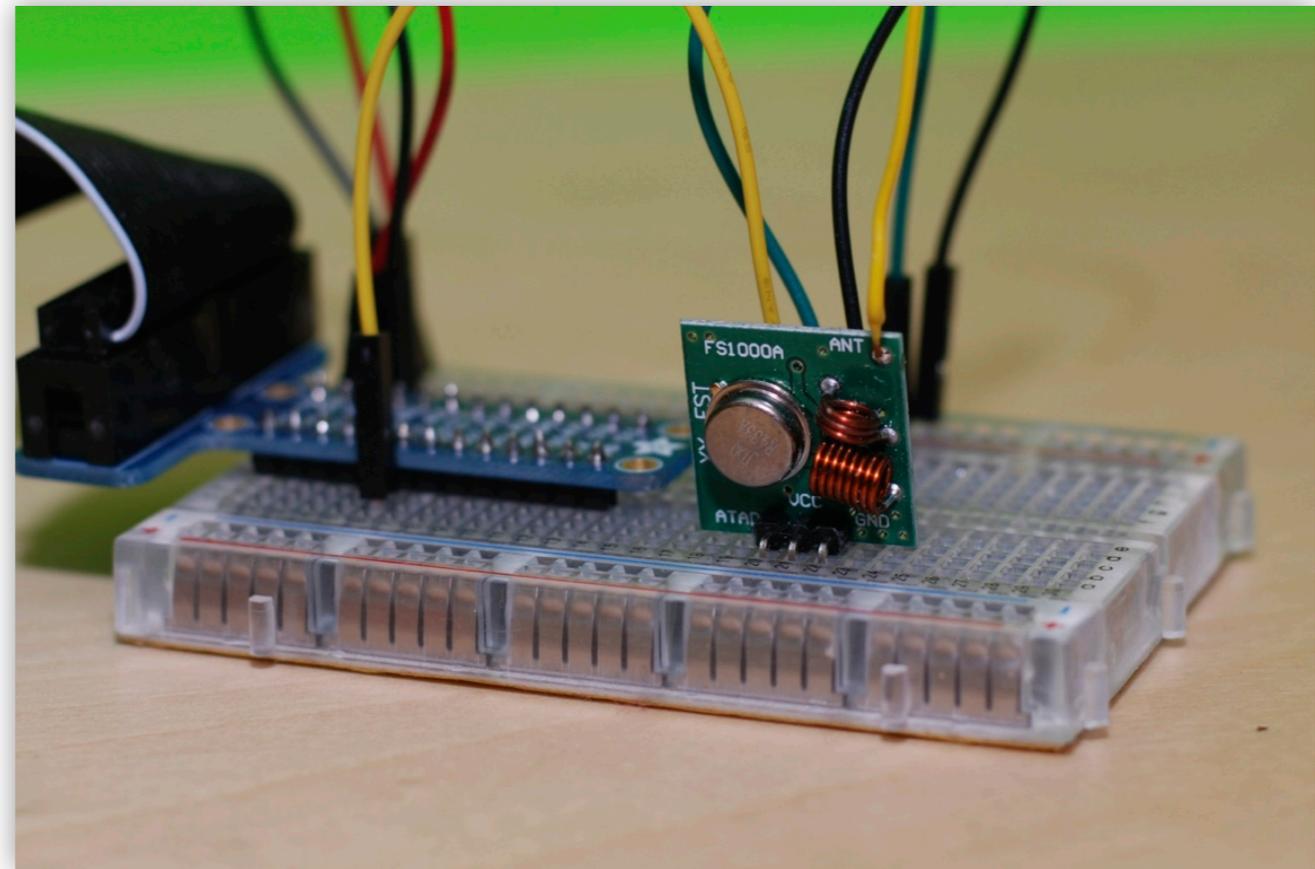
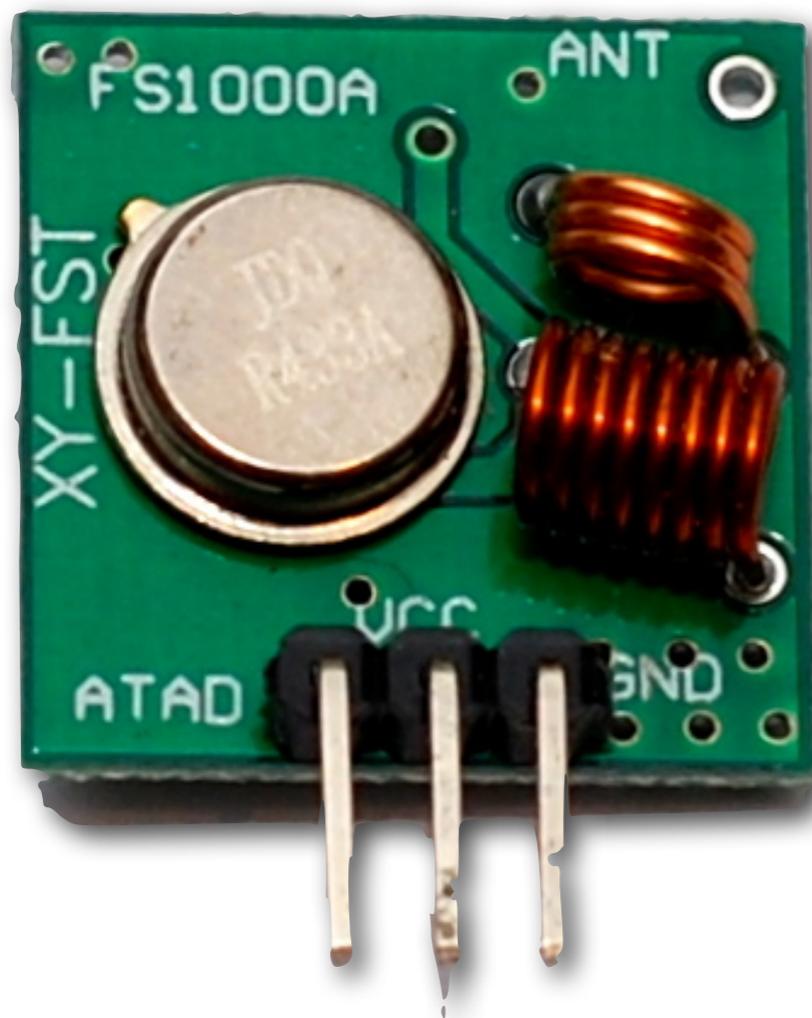
```
pi@raspberrypi ~/rcswitch-pi $  
pi@raspberrypi ~/rcswitch-pi $ sudo ./send a 1 1 0  
sending: houseCode:a group:1 device:1 command: 0  
pi@raspberrypi ~/rcswitch-pi $
```



Baumarkt Funkschalter



Funksteckdosen schalten



Kodierung senden

- Bit-Codierung wird via GPIO pin an den Transmitter gesendet (bit-banging)
- Problem mit Java:
 - bit-banging ist sehr zeitkritisch
 - dieses Timing kann mit Java nicht garantiert werden
 - also: C



=



rc-switch

Arduino library to operate low cost 315 MHz / 433 MHz remote control devices



r10r / rcswitch-pi

RCSwitch implementation for the Raspberry Pi

send.cpp

```
[...]  
int PIN = 0;  
  
if (wiringPiSetup() == -1) return 1;  
RCSwitch mySwitch = RCSwitch();  
mySwitch.enableTransmit(PIN);  
  
char* houseCode = argv[1];  
int group = atoi(argv[2]);  
int device = atoi(argv[3]);  
int command = atoi(argv[4]);  
  
[...]  
  
switch(command) {  
    case 1:  
        mySwitch.switchOn(houseCode[0], group, device);  
        break;  
    case 0:  
        mySwitch.switchOff(houseCode[0], group, device);  
        break;  
}  
  
[...]
```

./send [houseCode] [group] [device] [command]

Send.java

```
public enum Command {
    TURN_OFF, TURN_ON;
}
public static final String SEND_COMMAND = "/home/pi/rcswitch-pi/send";

public String send(String deviveCode, Command command) {
    StringBuilder builder = new StringBuilder();
    try {
        String commandLine = String.
            format("%s %s %s", SEND_COMMAND, deviveCode, command.ordinal());
        logger.log(Level.INFO, "send: {0}", commandLine);

        Runtime rt = Runtime.getRuntime();
        Process pr = rt.exec(commandLine);
        BufferedReader rd = new BufferedReader(new InputStreamReader(pr.
            getInputStream()));
        String line;
        while ((line = rd.readLine()) != null) {
            logger.log(Level.INFO, line);
            builder.append(line);
        }

    } catch (IOException ex) {
        Logger.getLogger(Send.class.getName()).
            log(Level.SEVERE, null, ex);
    } finally {
        return builder.toString();
    }
}
```

Device.java

```
public class Device {  
  
    private String name;  
    private String houseCode;  
    private String groupId;  
    private String deviceId;  
}
```

DeviceControl.java

```
public class DeviceControl {

    private Send send = new Send();
    private static final Logger logger = Logger.getLogger(Configuration.class.getName());

    public void turnOn(Device device) {
        doSwitch(device, Send.Command.TURN_ON);
    }

    public void turnOff(Device device) {
        doSwitch(device, Send.Command.TURN_OFF);
    }

    public void doSwitch(Device device, Send.Command command) {
        logger.log(Level.INFO, "About to {0} {1} ({2})", new Object[]{command, device.getName(), device.getId()});
        if (send.isSendCommandExecutable()) {
            send.send(device.getId(), command);
        } else {
            logger.log(Level.SEVERE, "{0} could not be executed!", command);
        }
    }
}
```

DevicePane

DevicePane.fxml

```
<fx:root type="javafx.scene.layout.AnchorPane" prefHeight="150.0" prefWidth="1220.0"
  <children>
    <Button id="buttonOn" fx:id="onButton" alignment="CENTER_LEFT" graphicTextGap="2
    <Button id="buttonOff" fx:id="offButton" maxHeight="-1.0" maxWidth="-1.0" minHei
  </children>
</fx:root>
```

DevicePane.java

```
public DevicePane(final Device device, final Configuration configuration) {
    this.configuration = configuration;
    this.device = device;
    init();
}

private void init() {
    FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("/fxml/DevicePane.fxml"));
    fxmlLoader.setRoot(this);
    fxmlLoader.setController(this);
    try {
        fxmlLoader.load();
    } catch (IOException ex) {
        Logger.getLogger(DevicePane.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```



Licht Haustür



SwitchService in DevicePane

```
turnOnService = new SwitchService(device, Send.Command.TURN_ON);  
turnOffService = new SwitchService(device, Send.Command.TURN_OFF);
```

```
offButton.setOnAction(new EventHandler<ActionEvent>() {  
    @Override  
    public void handle(ActionEvent t) {  
        turnOffService.restart();  
    }  
});  
  
onButton.setOnAction(new EventHandler<ActionEvent>() {  
    @Override  
    public void handle(ActionEvent t) {  
        turnOnService.restart();  
    }  
});
```

```
private class SwitchService extends Service<String> {  
  
    private Device device;  
    private Send.Command command;  
  
    public SwitchService(Device device, Send.Command command) {  
        this.device = device;  
        this.command = command;  
    }  
  
    @Override  
    protected Task<String> createTask() {  
        return new Task<String>() {  
            @Override  
            protected String call() {  
                deviceControl.doSwitch(device, command);  
                return httpDeviceControl.doSwitch(device, command);  
            }  
        };  
    }  
}
```

SweetHomePane

SweetHomePane.fxml

```
<fx:root type="javafx.scene.layout.AnchorPane" maxHeight="720.0" max  
  <children>  
    <Text layoutY="673.0" strokeWidth="0.0" styleClass="copyright" t  
    <Button fx:id="exitButton" mnemonicParsing="false" prefHeight="-  
    <VBox id="devicesPanel" fx:id="devicesPane" prefHeight="529.5" p  
    <ToggleButton fx:id="locationButton" layoutX="30.0" layoutY="602  
    <Label fx:id="locationLabel" layoutX="207.0" layoutY="646.0" sty  
    <Text layoutY="659.0" styleClass="title" text="SweetHomeFX" Anch  
    <Label layoutX="207.0" layoutY="621.0" styleClass="key-label" te  
    <Text layoutX="1003.0" layoutY="691.0" styleClass="version" text  
  </children>  
</fx:root>
```

SweetHomePane.java

```
List<Device> devices = configuration.getDevices();  
for (Device device : devices) {  
    addDevice(device);  
}
```

```
public void addDevice(Device device) {  
    DevicePane devicePane = new DevicePane(device, configuration);  
    devicesPane.getChildren().add(devicePane);  
}
```

Konfiguration (JAXB)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration>
  <atHomeValue>true</atHomeValue>
  <device>
    <deviceId>1</deviceId>
    <groupId>1</groupId>
    <houseCode>a</houseCode>
    <name>Licht Haustür</name>
  </device>
  <device>
    <deviceId>2</deviceId>
    <groupId>1</groupId>
    <houseCode>a</houseCode>
    <name>Licht Terrasse</name>
  </device>
  <device>
    <deviceId>3</deviceId>
    <groupId>1</groupId>
    <houseCode>a</houseCode>
    <name>Springbrunnen</name>
  </device>
  <httpContextValue>██████████</httpContextValue>
  <httpPortValue>██████</httpPortValue>
  <lanServerValue>192.168.0.61</lanServerValue>
  <serverNameValue>192.168.0.61</serverNameValue>
  <wanServerValue>████████████████████</wanServerValue>
</configuration>
```

Cut
the
rope



Cut the rope

Nicht schön:

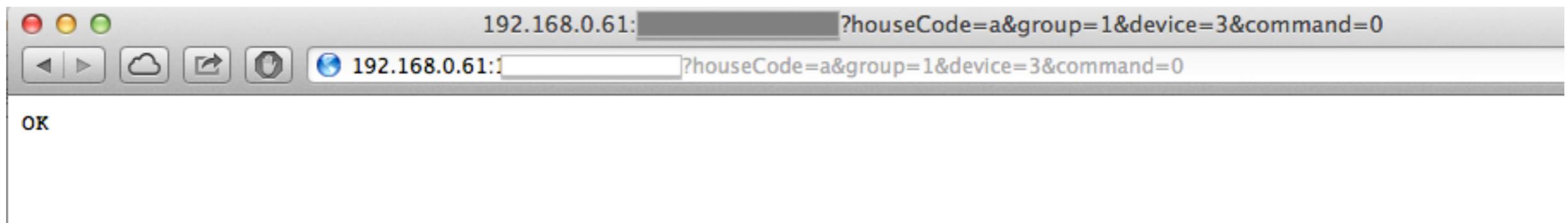
alle UI-Actions lokal

UI muss auf dem (gleichen) Raspberry Pi laufen

Cut the rope

Ziel: Schalten via URL-Aufruf:

http://192.168.0.61:<PORT>/<CONTEXT>?
houseCode=a&group=1&device=2&command=0



Cut the rope

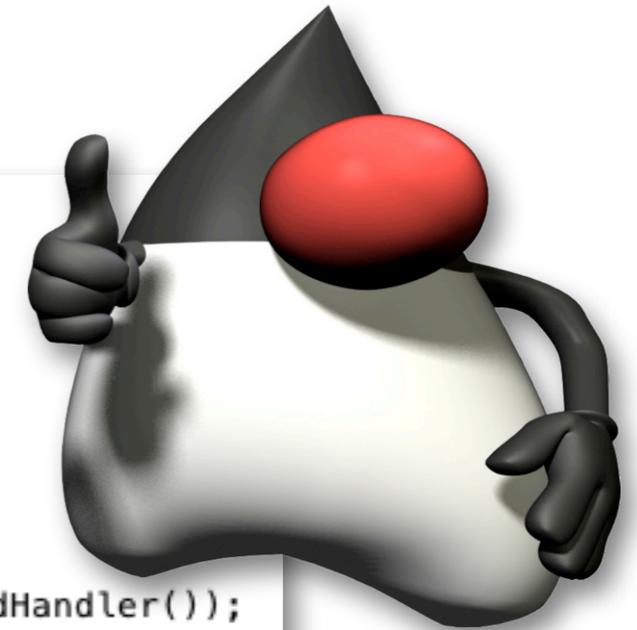
- Apache/PHP: zu wenig Java
- GlassFish/Java EE



SweetHomeServer.java

```
public SweetHomeServer(final int httpPort, final String httpContext) {
    this.httpPort = httpPort;
    this.httpContext = httpContext;
    init();
}

private void init() {
    try {
        httpServer = HttpServer.create(new InetSocketAddress(httpPort), 0);
        HttpContext context = httpServer.createContext("/") + httpContext, new HttpCommandHandler());
        context.getFilters().add(new GenericParameterFilter());
    } catch (Exception ex) {
        Logger.getLogger(SweetHomeServer.class.getName()).log(Level.SEVERE, ex.getMessage());
    }
}
```



```
Lummerland — pi@raspberrypi: ~ — ssh — 100x30
pi@raspberrypi ~ $ runfxapp de.jensd.sweethome.server.Start
-----
- S W E E T H O M E S E R V E R -
-----

Context: 
Port: 
Enter 'q' to shutdown server.
Waiting for actions...
Jul 21, 2013 2:16:57 PM de.jensd.sweethome.server.SendCommandHttpHandler handle
INFO: received params:{command=1, device=3, group=1, houseCode=a}
Jul 21, 2013 2:16:58 PM de.jensd.sweethome.server.SendCommandHttpHandler handle
INFO: from :/192.168.0.13:55810
Jul 21, 2013 2:16:58 PM de.jensd.sweethome.server.SendCommandHttpHandler handle
INFO: Send: a 1 3 TURN_ON
Jul 21, 2013 2:16:58 PM de.jensd.sweethome.core.Send send
INFO: send: /home/pi/rcswitch-pi/send a 1 3 1
Jul 21, 2013 2:16:58 PM de.jensd.sweethome.core.Send send
INFO: sending: houseCode:a group:1 device:3 command: 1
```

HttpDeviceControl.java

```
@Override
public String doSwitch(Device device, Send.Command command) {
    logger.log(Level.INFO, "doSwitch: about to {0} {1} ({2})", new Object[]{command, device.getName(), device.getId()});
    return send(device.getHouseCode(), device.getGroupId(), device.getDeviceId(), String.valueOf(command.ordinal()));
}

public String send(String houseCode, String group, String device, String command) {
    StringBuilder result = new StringBuilder();
    try {
        String requestParams = createRequestParams(houseCode, group, device, command);

        logger.log(Level.INFO, "{0}?{1}", new Object[]{url.toExternalForm(), requestParams});

        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("POST");
        connection.setDoInput(true);
        connection.setDoOutput(true);
        connection.setUseCaches(false);
        connection.setRequestProperty("Content-Type",
            "application/x-www-form-urlencoded");
        connection.setRequestProperty("Content-Length", String.valueOf(requestParams.length()));
    }
}
```



Jens Deters @Jerady

9h

Small contest: be the first to switch off the light of my terrace by click on this link: [\[REDACTED\]](#)

Expand



Jens Deters @Jerady

9h

OFF now.
Congrats: '84.125. [REDACTED] won ;-)

Expand



José Pereda @JPeredaDnr

8h

@Jerady That was me :D

Hide conversation

Reply

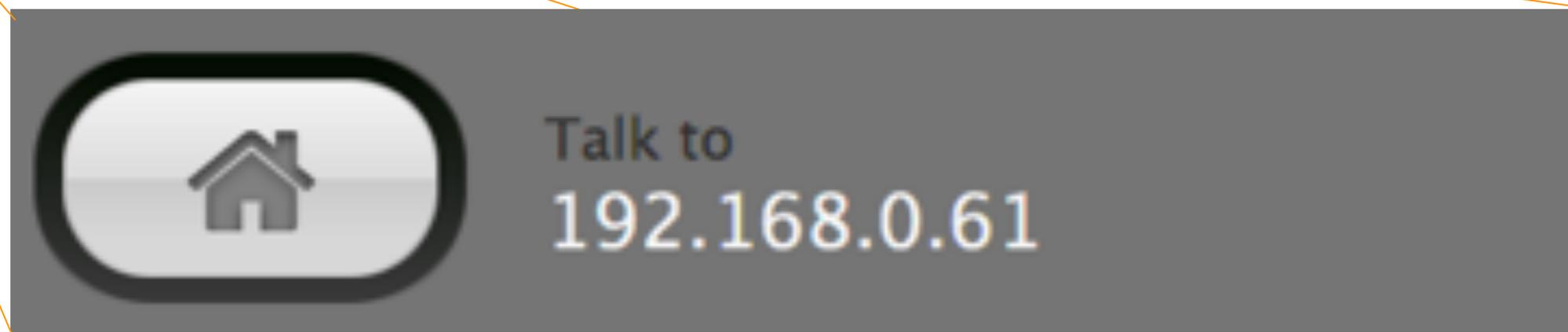
Retweet

Favorite

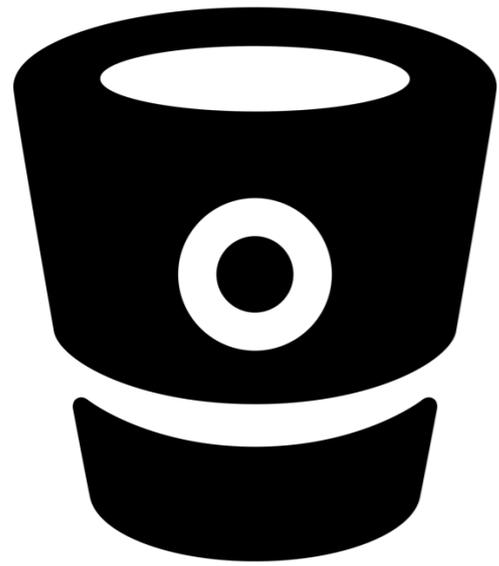
More

11:56 PM - 17 Jul 13 - Details

Standort umschalten



DIE IMMO



Get the Code

<https://bitbucket.org/Jerady>

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Jens Deters

@jerady

mail@jensd.de

www.jensd.de

LINKS

- <http://www.raspberrypi.org>
- <http://wiringpi.com>
- <http://pi4j.com>
- <https://code.google.com/p/rc-switch/>
- <https://github.com/r10r/rcswitch-pi>
- http://elinux.org/RPi_Low-level_peripherals
- <http://intertechno.at>