Himbeerkuchen zum Kaffee
Java für Raspberry Pi

Frank Pientka

Oft sind es die einfachen Dinge im Leben, die viel Freude machen. Wer sich selbst mal mit dem Internet-der-Dinge beschäftigen möchte, für den bietet der Ein-Platinen-Computer Raspberry Pi aufgrund seiner guten Erweiterungsmöglichkeit einen guten Einstieg. Nach einer Einführung in den Raspberry Pi wird gezeigt wie dessen Schnittstellen mit Java angesprochen werden können.

Einstieg

Der Raspberry Pi ist ein kreditkartengroßer Einplatinen-Computer, der von der Raspberry Pi Foundation entwickelt wird, um Schüler wieder mehr für Informatik zu begeistern. Die anfängliche Idee dazu hatte 2006 der Informatikdoktorand Eben Upton an der Cambridge University, da viele seiner Informatikstudenten nicht wussten, wie ein Computer funktioniert. Phonetisch lehnt sich der Name an das englische Wort für Himbeerkuchen an, wobei „Pi“ für „Python Interpreter“ steht, der ursprünglich mitgeliefert werden sollte.

Um das Ganze auf eine breitere Basis zu stellen, wurde 2009 die Raspberry Pi Foundation [RPiFound] gegründet, zu der auch der Elite-Spieleentwickler David Braben und Jack Lang, der den Heimcomputer BBC Micro mitentwickelte, gehören. Die Vorstellung der ersten Prototypen und die Erstproduktion stießen schon auf großes öffentliches Interesse. So war es nicht verwunderlich, dass es zu längeren Lieferzeiten beim Verkaufsstart 2012 kam.

Wenn man einen Raspberry Pi Model B erstanden hat, benötigt man noch ein wenig Zubehör. Neben einer Micro-USB-Stromversorgung braucht man eine mindestens 2 GB große SD-Karte. Noch besser ist es, einen DVI-HDMI-Adapter für den Monitor oder ein HDMI-Kabel für den Fernseher zu kaufen. Wer möchte, kann sich sogar passende Gehäuse bestellen, sodass man mit 60 Euro einen kleinen Server erhält. Wer hier wissen möchte, welche Hardware bereits mit dem Raspberry Pi überprüft wurde, findet unter [RPiVP] eine gute Übersicht. Über einen Audio-, zwei Videoausgänge, eine 10/100-MBit-Netzkarte und zwei USB-Anschlüssen können Sie noch weitere Geräte anschließen.

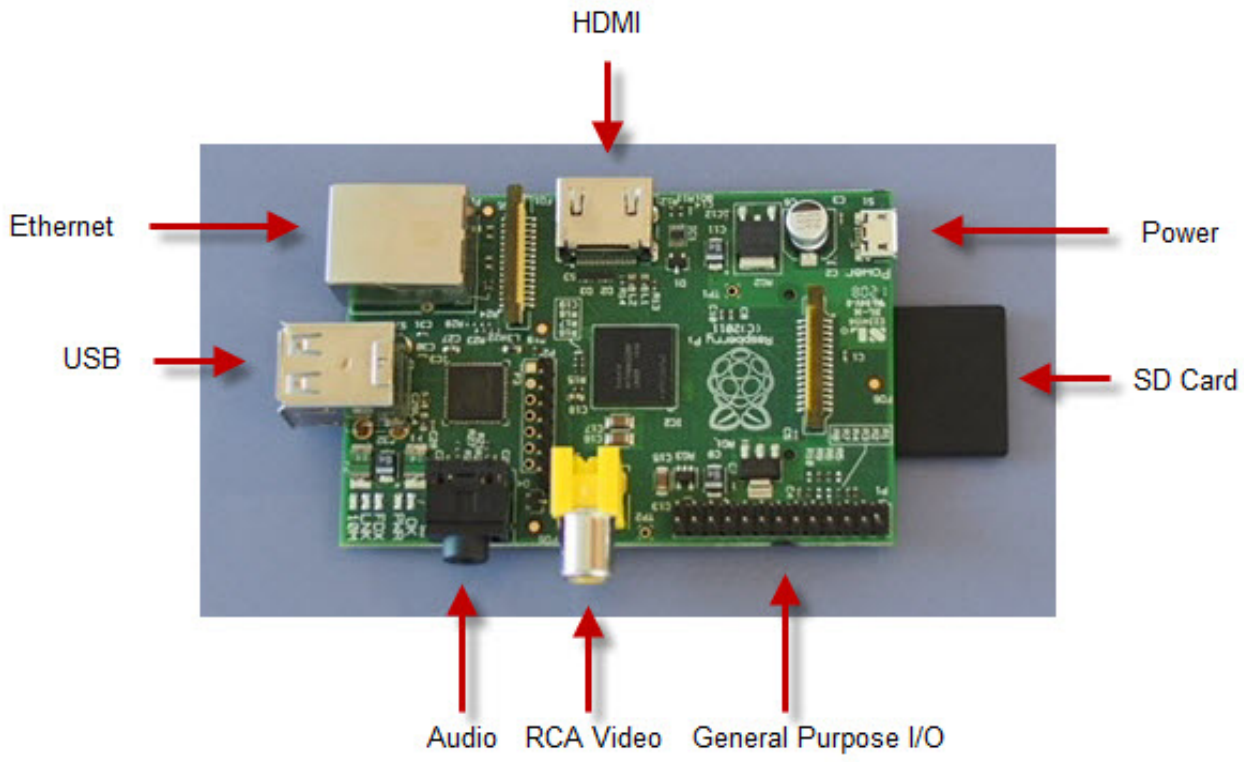

Abb. 1: Raspberry Pi Model B, Anschlüsse (raspberrypi.org)

Mit DD oder dem Win32DiskImager können Sie das von [RPiFound] heruntergeladene Debian/Raspbian-Image auf die SD-Karte schreiben. Nachdem Sie die Karte in den Pi gesteckt haben, können Sie davon booten, da der Pi

kein BIOS hat. Sie melden sich dann mit dem Benutzer pi und dem Passwort raspberry an. Dann können Sie weitere Software, wie zum Beispiel das JDK 8 für ARM-Prozessoren [JDK8ARM], installieren. Um die spätere Arbeit zu vereinfachen, sollten Sie die Partition vergrößern, um so den kompletten Platz Ihrer SD-Karte verfügbar zu machen und statt der dynamisch per DHCP vergebenen IP-Adresse eine feste IP-Adresse eingeben. Über sudo raspi-config können Sie die Konfiguration des Pi ändern, wie die Partitionserweiterung über den Menüpunkt expand_rootfs oder die Tastaturbelegung über configure_keyboard. Die Version Ihres Pi können Sie mit dem Befehl cat /proc/cpuinfo kontrollieren, damit sollte für Model B Revision 7 angezeigt werden. Ihre Betriebssystemversion können Sie mit uname –a anzeigen. Danach starten Sie die grafische X-Oberfläche mit startx.

Der ARM-Prozessor ist mit Broadcoms „VideoCore“-Grafikkoprozessor ausgestattet, um Filme in FullHD-Auflösung mit H.264-Codec zu decodieren. Wenn die Mediacenter-Distribution XBMC ("Xbox Media Center") 12.2 für den Raspberry Pi angepasst ist, steht sie als Image oder per Setup-Programm zur Initialisierung der SD-Karte als Raspbmc unter [Raspbmc] zur Verfügung. Selbst eine ERP-Software, Mercaware, ist inzwischen für Kleinfirmen für den Pi erhältlich.

Der Raspberry Pi bietet die frei-programmierbare Schnittstelle GPIO (General Purpose Input/Output) an. Darüber können die LEDs, Sensoren, Displays und andere Geräte, wie eine Kamera angesteuert werden. Um GPIO aus Java heraus anzusprechen, benötigt man dazu den Wrapper Pi4j [Pi4J].

Neben vielen Onlineinformationen [RPiComm] und einem eigenen kostenlosen Pi-Magazin [RPiWiki] empfiehlt sich das kostenlose PDF-Handbuch [RPiMan].

Wir überprüfen zunächst die vergebene IP-Adresse mit hostname –I, und in welchem Zustand die Netzwerkschnittstellen sind mit ifconfig -a. Danach passen Sie mit dem Editor vi die Datei /etc/network/interfaces an die von Ihnen gewünschte statische IP-Adresse an:

```
iface eth0 inet static /* feste IP Adresse
address <IP-Adresse> /* Gewünschte IP Adresse des RPI
netmask <Netz Maske> /* Im Heimnetzwerk meistens 255.255.255.0
gateway <IP-Adresse> /* IP Adresse des Routers
```

Die Änderungen werden erst nach einem Neustart der Netzwerk Services aktiv.

```
sudo /etc/init.d/networking restart
```

Sollten die Netzwerkschnittstellen nicht funktionieren, können Sie diese auch mit sudo ifdown eth0 deaktivieren und mit sudo ifup eth0 wieder aktivieren. Sogar WLAN wird über USB unterstützt. In einer X-Konsole können Sie Ihr erstes Spiel in cd python_games mit python flippy.py aufrufen.
Sie sollten JDK8 [JDK8ARM] mit der hardwarebasierten Gleitkommazahlenberechnungsunterstützung verwenden, was sich günstig auf die Performance auswirkt.

Nachdem Sie das JDK8 in das Verzeichnis /opt/jdk8 entpackt haben, müssen Sie nur noch die Umgebungsvariablen anpassen. Statt Java SE können Sie auch die Embedded oder die ME Embedded 3.3 for Raspberry Pi Model B (Early Access) verwenden.

Um Java zu verwenden, setzen Sie folgende Umgebungsvariablen in der Datei /etc/profile

```
JAVA_HOME=/home/pi/java/jdk1.8.0
PATH=$PATH:$JAVA_HOME/db/bin:$JAVA_HOME/bin
CLASSPATH=.:/home/pi/java/jdk1.8.0/lib:/opt/pi4j/lib:$JAVA_HOME/jre/lib
```

Mit java –version bzw. javac- version können Sie testen, ob Sie Java korrekt installiert haben. Danach können Sie über das Internet Ihr System mit sudo apt-get update aktualisieren lassen. Herunterfahren können Sie den Pi, mangels Ausschalter, nur durch „System herunterfahren“:

```
$ sudo shutdown -h now
```

Kaffeekochen

Um mit der GPIO zu arbeiten, gibt es mehrere Möglichkeiten. Wir zeigen zunächst, wie sich mit Unix-Befehlen die GPIO-Schnittstelle ansprechen lässt.

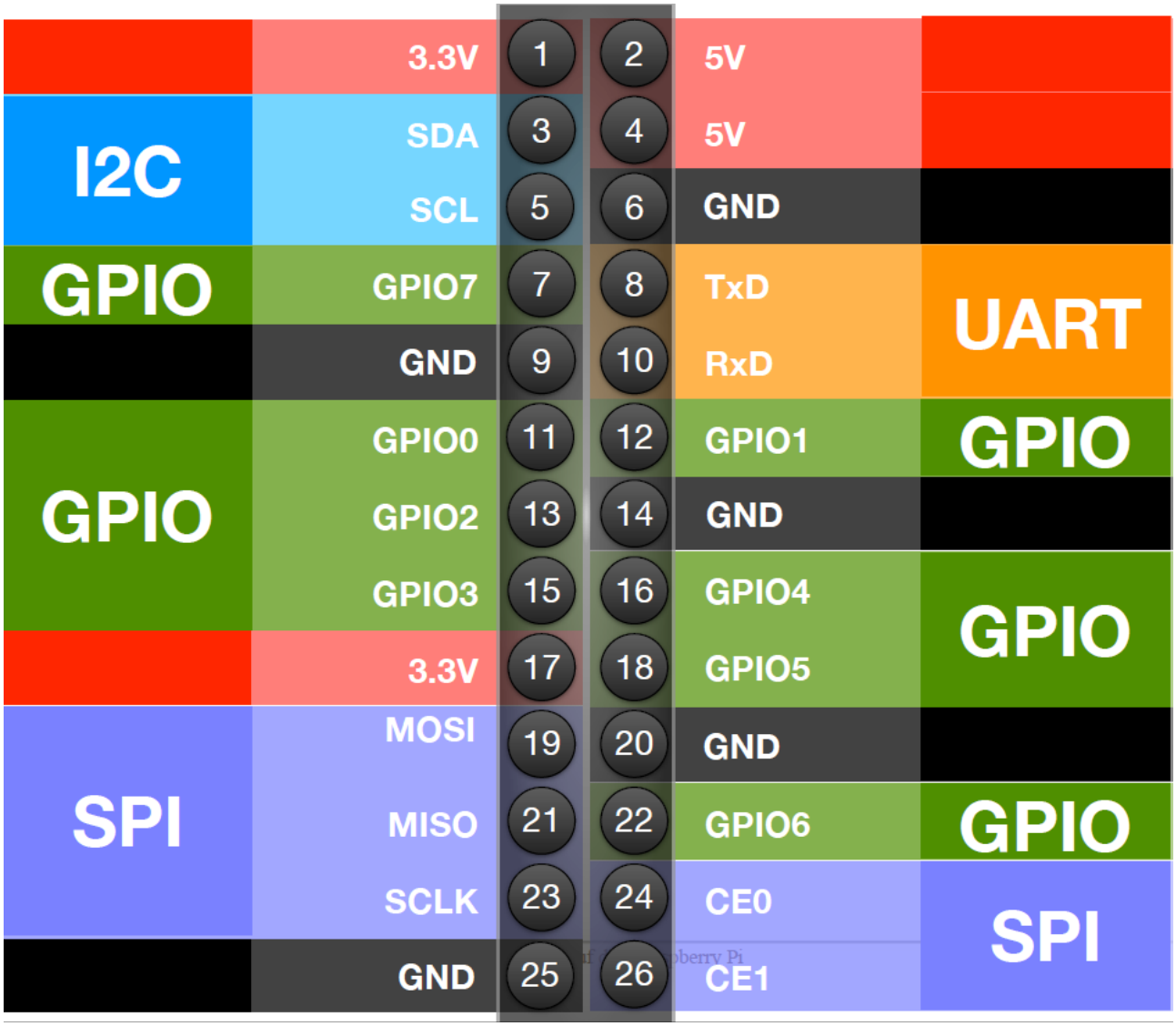

Abb. 2: GPIO-Belegung

```
sudo su
# set up GPIO 7 and set to input
echo "7" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio7/direction
# write output
echo "1" > /sys/class/gpio/gpio7/value
# read from input
cat /sys/class/gpio/gpio7/value
# clean up
echo "7" > /sys/class/gpio/unexport
```

Etwas einfacher als über Betriebssystembefehle geht es mit dem in C geschriebenen Kommandozeilen-Werkzeug wiringPi [wiringPi]. Nach dem Herunterladen müssen Sie dieses erst übersetzen

```
tar xfz wiringPi-02a3bd8.tar.gz
cd wiringPi-02a3bd8
./build
```

Danach können Sie das Werkzeug, wie folgt aufrufen

```
gpio -v
man gpio
gpio readall
gpio -g write 7 1
```

Wenn Sie GPIO aus Java heraus ansprechen wollen, so verwenden Sie dazu die Pi4J-Bibliothek. Entweder übersetzen Sie alle Beispiele unter *examples* zusammen oder jedes für sich.

Mit dem Beispiel BlinkGpioExample können Sie die GPIO-Dioden 1 und 3 zum Blinken bringen. Die Blinkfrequenz wird bei einem Knopfdruck auf die GPIO-Nummer 2 erhöht:

```
GpioController gpio = GpioFactory.getInstance(;
GpioPinDigitalOutput led1 =
  gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01;
GpioPinDigitalOutput led2 =
  gpio.provisionDigitalOutputPin(RaspiPin.GPIO_03)
GpioPinDigitalInput myButton =
  gpio.provisionDigitalInputPin(RaspiPin.GPIO_02,
    PinPullResistance.PULL_DOWN;
myButton.addListener(new GpioPinListenerDigital() {
  public void handleGpioPinDigitalStateChangeEvent
    (GpioPinDigitalStateChangeEvent event) {
      if(event.getState().isHigh())
        led2.blink(200);
      else
        led2.blink(1000);
}});
led1.blink(500, 15000);
led2.blink(1000);
```

Das mit Pi4J mitgelieferte Beispiel können Sie wie folgt übersetzen und aufrufen:

```
javac -classpath .:classes:/opt/pi4j/lib/* -d . BlinkGpioExample.java
sudo java -classpath .:classes:/opt/pi4j/lib/* BlinkGpioExample
WiringPiSoftPWMExample ShutdownGpioExample
```

Webanwendung

Eine andere Möglichkeit, die GPIO über eine Weboberfläche zu kontrollieren, bietet [Bub]. Zunächst installieren Sie den Java-Webserver Apache Tomcat

```
wget http://www.eu.apache.org/dist/tomcat/tomcat-7/v7.0.42/bin/apache-tomcat-7.0.42.tar.gz
tar xzf apache-tomcat-7.0.42.tar.gz
mv apache-tomcat-7.0.42 tomcat
```

Damit Sie die Verwaltungsoberfläche von Tomcat verwenden können, müssen Sie die Datei tomcat/conf/tomcat-users.xml anpassen um

```
  <role rolename="tomcat"/>
  <role rolename="manager-status"/>
  <role rolename="manager-gui"/>
<user username="tomcat" password="tomcat"
  roles="tomcat,manager-status"/>
```

Danach können Sie über tomcat/bin/startup.sh den Tomcat-Server starten und Sie können sich über http://localhost:8080/ anmelden. Dann kopieren Sie folgende Derby-Bibliotheken in folgende Verzeichnisse mit

```
cp ~/java/jdk1.8.0/db/lib/derby.war webapps/
cp ~/java/jdk1.8.0/db/lib/derby.jar lib/
cp ~/java/jdk1.8.0/db/lib/derbynet.jar lib/
cp ~/java/jdk1.8.0/db/lib/derbyclient.jar lib/
```

Nach einem Neustart des Tomcats können Sie die mit dem JDK mitgelieferte Derby-Datenbank administrieren über http://localhost:8080/derby/derbynet. Eine Datenbank können Sie über die Derby-Kommandozeile mit

```
ij> connect 'jdbc:derby://localhost:1527/toursdb;create=true';
```

anlegen. Um die GPIO Ihres Pi über HTTP anzusteuern, können Sie die Anwendung raspberry-pi-gpio-web-control-1.0-SNAPSHOT.war ebenso ins webapps-Verzeichnis kopieren. Mit folgenden Aufrufen können Sie dann die GPIO-Schnittstelle remote ansteuern

```
http://localhost:8080/handle?g0=1&g1=0
http://localhost:8080/handle?darknessSensor1in==1&lamp1out=1
http://localhost:8080/handle?otherOutputPort==0&lamp1out=1
http://localhost:8080/handle?analogSensor2ti=IN&analogSensor2ti>10000&lamp1out=1
```

Optimieren

Gerade bei einem so spartanischen Rechner wie dem Raspberry Pi sollten die Ressourcen optimal genutzt werden. Da die SD-Karte die langsamste Komponente ist, sollte nur in Notfällen mit Swap-Dateien gearbeitet werden. Besser ist es, eher sparsam mit dem Hauptspeicher umzugehen.

Am Einfachsten nimmt man Anpassungen am Pi mit sudo raspi-config vor. Die CPU-Frequenz lässt sich dort in 5 Stufen von 700 MHz bis 1 GHz übertakten. Ein guter Wert ist Medium mit 900 MHz. Dadurch, dass nur bei Bedarf die Taktfrequenz dynamisch angehoben und auch die Temperatur des Prozessors überwacht wird, hat das wenig Auswirkung auf die Lebensdauer, verbessert jedoch die Performance. Standardmäßig erhält der Grafikprozessor 64

Mbyte der 512 Mbyte Hauptspeicher. Über das memory_split-Menü können den Speicher für den Grafikprozessor auf 32 MByte reduzieren, so dass insgesamt 480 MByte für die Anwendungen zur Verfügung stehen.

Ausblick

Der Raspberry Pi knüpft an die Zeiten der Amigas, ZX 81 oder Commodore 64 an, mit Hilfe derer sich viele intensiv und kreativ mit Computern auseinander setzten. Er macht inzwischen nicht nur der ursprünglichen Zielgruppe der Schüler, sondern auch deren Vätern viel Freude.

Dadurch, dass der Pi auf Standardkomponenten und Open Source beruht, kann er beliebig erweitert werden. Durch die Verwendung von Linux und Java fühlt man sich hier schnell zu Hause. Gerade für den embedded Bereich, wie beispielsweise in der Heimautomation, Kiosksystemen oder als Multimediaserver [Raspbmc], bietet er gute und kostengünstige Einstiegsmöglichkeiten. Durch seine weite Verbreitung steht für den Raspberry Pi eine große Anzahl an Erweiterungen, Hilfen oder Informationen zur Verfügung. Mit ihm wird das Internet-der-Dinge und der Einstieg in die embedded Programmierung leicht, da auf bisheriges Wissen aufgebaut werden kann.

Links und Literatur

[Bub] S. Bub, Raspberry Pi GPIO Web Control Interface, https://bitbucket.org/sbub/raspberry-pi-gpio-web-control/

[JDK8ARM] JDK 8 for ARM, Early Access, https://jdk8.java.net/download.html
https://wiki.openjdk.java.net/display/OpenJFX/OpenJFX+on+the+Raspberry+Pi

[Pi4J] Java library for Raspberry Pi, http://pi4j.com/usage.html

[Raspbmc] Raspbmc XBMC für Raspberry Pi, http://www.raspbmc.com/download/

[RPiComm] Deutschsprachige Raspberry Pi Community, http://raspberrycenter.de/

[RPiFound] Raspberry Pi Foundation, http://www.raspberrypi.org

[RPiJavaME] Oracle Java ME Embedded 3.3 for Raspberry Pi Model B,
http://www.oracle.com/technetwork/java/embedded/downloads/javame/index.html

[RPiMan] Raspberry Pi, Handbuch, http://downloads.raspberrypi.org/Raspberry_Pi_Education_Manual.pdf

[RPiVP] Für Raspberry Pi überprüfte Hardware, http://elinux.org/RPi_VerifiedPeripherals

[RPiWiki] Zeitschrift MagPi und Wikis zu Raspberry Pi, http://www.themagpi.com/, http://elinux.org/Category:RaspberryPi, http://elinux.org/RPi_General_History

[wiringPi] https://git.drogon.net/?p=wiringPi und https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/

Getting Started with Java SE Embedded on the Raspberry Pi www.oracle.com/technetwork/articles/java/raspberrypi-1704896.html

Frank Pientka **ist Senior Architect bei der MATERNA GmbH in Dortmund. Er ist seit mehreren Jahren im Bereich Java EE tätig. Seine Schwerpunkte sind JavaEE-Anwendungen. Dazu hat er auch schon mehrere Fachartikel und ein Buch über Geronimo veröffentlicht. E-Mail: frank.pientka@gmx.de**