

Kapitel 11

Hörsaalübung: Embedded Webserver

Übersicht

Nachdem die vorhergehenden Kapitel viel Theorie über das Betriebssystem Linux und den Single-Board Computer Raspberry Pi enthalten haben, sollen jetzt die erworbenen Kenntnisse angewendet und gefestigt werden.

Lernziele

Nach Abschluss dieses Kapitels

- können Sie auf einem Embedded System, wie dem Raspberry Pi, eine Linux-Distribution einrichten und
- das System bedienen und konfigurieren.

11.1 Einleitung

Viele elektronische Geräte bieten zwischenzeitlich die Möglichkeit per Webinterface bedient zu werden. Zuhause können dies der Videorekorder, das DSL Modem, oder im gewerblichen Umfeld Digital Signage Systeme zur Präsentation von Videos auf Plätzen oder in Geschäften sein. Die Anforderungen an Embedded Hardware ist damit einerseits sehr vielfältig andererseits erfordert sie leistungsfähige und günstige Hardware sowie flexible aber anspruchslose Betriebssysteme. Auf Grundlage der Rapidprototyping Hardware wollen wir eine einfache Plattform für einen konkreten Anwendungsfall anpassen.

11.2 Aufgabenstellung

In dieser Hörsaalübung wollen wir eine Prototyping-Plattform für eine webbasierte Bedienoberfläche für ein Embedded System aufbauen. Dieses basiert hardwareseitig auf einem Raspberry Pi, der sowohl für die Konfiguration des Systems per SSH als auch später über die Weboberfläche über Netzwerkzugriff bedienbar sein soll. Ein Zugriff über serielle Schnittstelle ist nicht vorgesehen. Die Nutzung von Tastatur, Maus und Bildschirm ist nur bei der Initialisierung des Systems vorgesehen. Eine graphische Nutzeroberfläche ist nicht vorzusehen. Als Betriebssystem wählen wir das Debian Linux basierte Raspbian, eine für den Raspberry Pi optimierte Distribution. Als Webserver werden wir den Apache Webserver installieren und konfigurieren, der als Paket unter Raspbian verfügbar ist.

11.3 Anforderungen und Material

Für unser Projekt benötigen wir die folgende Ausstattung:

1. Linux basierte Entwicklungsumgebung (PC oder Notebook) mit SD-Kartenleser zur Vorbereitung der Speicherkarte mit dem verwendeten Betriebssystem
2. Administrator Zugriff auf der Entwicklungsumgebung
3. Ein Raspberry Pi als Hardware-Plattform inkl. Netzkabel, Stromanschluss, USB Tastatur und Monitor (Anschlusskabel z.B. HDMI-DVI erforderlich) für die Installation
4. Eine SD Speicherkarte mit mindestens 4 GB Speicherplatz
5. Anschlussmöglichkeit für Entwicklungsumgebung und Raspberry Pi an ein Netzwerk (Switch, Hub oder auch WLAN), das über einen DHCP verfügt und über das ein Zugriff aufs Internet möglich ist (Pakete nachinstallieren)

11.4 Installation auf SD

Unabhängig von der verwendeten Distribution kann in der Regel folgender Ablauf eingehalten werden:

1. Melden Sie sich an Ihrer Entwicklungsumgebung an. (Root-Rechte erforderlich!)
2. Download der Distribution z.B von <http://www.raspberrypi.org/downloads>.
Hier wird *Raspbian* verwendet.
3. Entpacken des Images mit `unzip <pfad>/<datei.zip>`

```
1 # unzip ./2014-09-09-wheezy-raspbian.zip
```
4. Mit `df -h` kann man die aktuell verfügbaren Speichergeräte prüfen.

```
1 # df -h
```

5. Schieben Sie nun die SD Karte in den Reader an ihrem Notebook ein.
6. Lassen Sie nun `df -h` wieder auf. Dort sollten sie nun das Device-File der SD-Karte sehen. In der Regel wird es sich um `/dev/mmcblk0p1` oder `/dev/sdd1` handeln. Es können übrigens mehrere Partitionen sein, die gemäß dem Muster `/dev/mmcblk0p*` oder `/dev/sdd*` aufgelistet werden. Im weiteren Verlauf wird nur das Haupt-Device ohne Nummer am Ende verwendet: `/dev/mmcblk0p` oder `/dev/sdd`

```
1 # df -h
```

7. Vorher müssen aber die Partitionen (alle bei mehreren Partionen) noch durch `umount` aus dem System entfernt werden. Erledigen Sie das mit dem Kommando `umount /dev/sdd1` bzw. nach dem Muster `/dev/sdd*`

```
1 # sudo umount /dev/sdd*
```

oder

```
1 # sudo umount /dev/mmcblk0p*
```

8. Schreiben Sie nun mit dem Programm `dd` die Daten des Images auf die SD Karte: `dd bs=4M if=<Pfad zum Imagefile> of=<Pfad zur SD-Karte>`.

```
1 # sudo dd if=2014-09-09-wheezy-raspbian.img of=/dev/sdd bs=1M
```

oder

```
1 # sudo dd if=2014-09-09-wheezy-raspbian.img of=/dev/mmcblk0 bs=1M
```

Achtung: Achten Sie darauf, dass Sie das richtige Device nutzen, da `dd` mit einem vollständigen Datenverlust auf dem Zieldevice verbunden ist, das mit `of` angesprochen wird!

9. Sollte sich der Schreibprozess nicht beenden (auf normalem PC zw. 10 und 15 Minuten) können Sie auf einer anderen Konsole als Root den Schreibprozess mit `kill -USR1 -n -x dd` beenden.
10. Entfernen Sie nach Abschluss die SD-Karte aus dem Kartenleser.

11.5 Initialisierung des Raspi – Erster Systemstart

Der Raspberry Pi ist bisher noch ohne Betriebssystem. Ein Start würde zu einer Fehlermeldung führen bzw. keine Bildschirmanzeige ergeben. Achten Sie nun darauf, dass der Raspi noch nicht mit dem Netzteil bzw. USB-Kabel für die Stromversorgung verbunden ist.

1. Legen Sie die fertige SD Karte in den SD Karten Slot des Raspi ein.
2. Schließen Sie Maus, Tastatur, Netzwerk und Bildschirm an.
3. Verbinden Sie nun das Gerät über den Mikro USB Port mit Netzstecker.

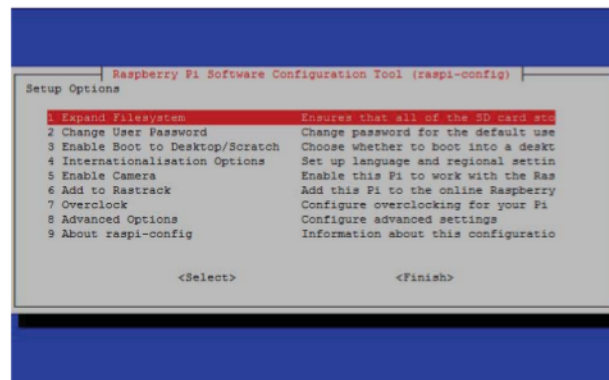


Abbildung 11.1: Raspberry Pi Software Configuration Tool

4. Wir sehen eine Reihe Statusmeldungen, die vom Systemstart und den ersten Initialisierung und Einrichtung des Standardsystems stammen.

Nach einigen Sekunden wechselt der Bildschirm und wir erhalten eine Konfigurationsmenu. Hier lohnt es sich gleich einige Einstellungen vorzunehmen, um das System optimal und einfach auf den Einsatz vorzubereiten.

5. Passen Sie unter Internationalisation Options die Einstellungen für Locale, Timezone und Keyboard Layout an.

Locale: de_DE.UTF-8 UTF-8

Timezone: Europe, Berlin

Keyboard: Generic 105-key (Intl) PC, Other, German, German

6. Ändern Sie das Kennwort für den User pi auf ein Kennwort Ihrer Wahl
7. Wechseln Sie in Advanced Options und aktivieren Sie die Einstellung für SSH.
8. Vergeben Sie einen Hostname in Advanced Options. Damit haben Sie neben der IP Adresse eine zweite Möglichkeit das System anzusprechen. Name kann hierbei Ihr Nachname sein (Vorlesungssystem) oder auch ein freier Name.
9. Erweitern Sie den Speicherplatz, so dass die gesamte SD Karte genutzt wird, indem sie Expand Filesystem auswählen.
10. Beenden Sie die Konfiguration und erlauben Sie einen Neustart des Systems, damit diese Einstellungen alle übernommen werden.
11. Nach dem Neustart können Sie sich mit ihrem neuen Passwort und dem Benutzernamen pi anmelden. Wenn Sie das Passwort nicht geändert haben lautet es raspberry .
12. Prüfen Sie die Netzwerkverbindung und stellen Sie die IP des Systems fest, da wir diese später für den Remotezugriff benötigen werden. Mit ifconfig bzw. ifconfig — less erhalten Sie eine Aufstellung aller Netzwerkgeräte und deren Konfiguration. Den für uns relevanten Wert finden Sie unter eth0 unter IP Adress.

```
1 # ifconfig
2 # ...
```

13. Notieren Sie sich die IP Adresse des Systems (z.B. 192.168.100.123).
 14. Wechseln Sie mit `sudo su` in den Superuser mode. Damit können Sie nun mit `init 0` das System herunterfahren
- ```
1 # sudo su
2 # init 0
```
15. Trennen Sie das System vom Strom.
  16. Jetzt können Sie Tastatur, Maus und Bildschirm vom System trennen.
  17. Wenn Sie nun das System wieder an den Strom anschließen, startet es autonom und ist über das Netzwerk erreichbar.

## 11.6 Zugriff mit SSH

Raspbian wird mit einem SSH-Server ausgeliefert, so dass ein Netzwerkzugriff zu Administrationszwecken möglich ist. Da man Embedded Systeme selten mit Monitor und Tastatur ausgestattet findet, ist man oft auf diesen Zugang angewiesen. Mit den Kenntnissen aus den vorhergehenden Kapiteln ist diese Art der Verwaltung eines Systems keine Herausforderung mehr.

Für einen Zugriff per SSH kann unter Windows das Programm PuTTY heruntergeladen und verwendet werden. Unter Linux verwenden wir das Programm `ssh <Nutzername> @ <IP-Adresse>`.

1. Öffnen Sie auf Ihrem Entwicklungssystem eine Terminal.
  2. Prüfen Sie die Verfügbarkeit des Raspberry Pi über das Netzwerk mit einem Ping auf die IP Adresse des Pi
- ```
1 # ping 192.168.100.123
```
3. Rufen Sie SSH auf und melden Sie sich als User `pi` auf dem Raspberry Pi remote an. Zum Beispiel:
- ```
1 # ssh pi@192.168.100.123
```
4. Verschaffen Sie sich einen Überblick über den Aufbau des Systems und prüfen Sie das Dateisystem und den Aufbau der Partitionen mit `mount`, `fdisk`, ...
  5. Rufen Sie das Programm `raspi-config` nochmals auf, und prüfen Sie, ob alle Einstellungen korrekt sind.

## 11.7 Administration und Paketverwaltung

Da unser Grundsystem jetzt funktionsfähig ist, wollen nun den Webserver und den FTP-Server einrichten. Dafür müssen wir einige Pakete (Apache Webserver, ProFTP) nachinstallieren, die bei einer Standardintegration noch nicht Systembestandteil sind.

1. Aktualisieren Sie die verwendeten Paketquellen mit dem Befehl `apt-get update`.

```
1 # sudo apt-get update
```

2. Führen Sie mit `apt-get upgrade` ein Systemupdate durch.

```
1 # sudo apt-get upgrade
```

3. Für die Installation und Betrieb brauchen wir eine eigen Gruppe. Legen Sie diese mit `groupadd www-data` und `usermod -a -G www-data www-data` an. (Abhängig von der verwendeten Linux Distribution existiert diese Gruppe bereits.)

```
1 # sudo groupadd www-data
```

```
2 # sudo usermod -a -G www-data www-data
```

4. Mit einem `apt-get install` können Sie das fehlende Paket `apache2` (Webserver) nachinstallieren.

```
1 # sudo apt-get install apache2
```

5. Testen Sie nun den Webserver, in dem Sie auf dem Entwicklersystem einen Browser öffnen und dort in der Adresszeile die IP des Raspberry Pi mit `http://` vorangestellt eingeben.

6. Öffnet dies eine Willkommensseite, dann ist der Apache Webserver bereit und kann verwendet werden.

7. Die Daten, die der Webserver ausliefert, finden Sie unter `/var/www`.

8. Für den FTP-Server installieren wir das fehlende Paket `proftpd` nach. Während der Installation wählen wir aus, dass wir den Server-Modus verwenden möchten.

```
1 # sudo apt-get install proftpd
```

9. Wir müssen nun noch den FTP Server konfigurieren, um Daten von einem beliebigen Rechner per FTP-Protokoll auf den Pi übertragen zu können. Öffnen Sie hierzu die `/etc/proftpd/proftpd.conf` mit dem Editor `nano`.

```
1 # sudo nano /etc/proftpd/proftpd.conf
```

10. Ergänzen Sie folgende Code am Ende der Datei:

```
1 DefaultRoot ~
```

```
2 AuthOrder mod_auth_file.c mod_auth_unix.c
```

```
3 AuthUserFile /etc/proftpd/ftpd.passwd
```

```
4 AuthPAM off
```

```
5 RequireValidShell off
```

11. Erstellen sie im `home`-Verzeichnis einen Ordner für den FTP-Server und vergeben Sie die erforderlichen Rechte.

```
1 # mkdir ftp-folder
```

```
2 # chmod g+s ftp-folder/
```

```
3 # chmod 775 ftp-folder/
```

```
4 # chown -R www-data:www-data ftp-folder/
```



12. Wechseln Sie in das Verzeichnis `/etc/proftpd`.

```
1 # cd /etc/proftpd
```

13. Geben Sie die UserID und GroupID des Benutzers `www-user` mit dem Befehl `id` aus.

```
1 # id www-user
```

14. Erstellen Sie nun einen neuen Benutzer für den FTP-Zugang mit dem Befehl `ftpasswd -passwd -name <Benutzername> -uid <UserID> -gid <GroupID> -home <VerzeichnisFuerFTP> -shell /bin/false`. Geben sie an den entsprechenden Stellen die im vorhergehenden Schritt ermittelten IDs ein.

```
1 # ftpasswd -passwd -name pi -uid 33 -gid 33
2 -home /home/pi/ftp-folder -shell /bin/false
```

15. Um die veränderte Konfiguration wirksam zu schalten müssen wir noch den FTP-Server neu starten.

```
1 # sudo /etc/init.d/proftpd restart
```

16. Per FTP können Sie nun eigene Dateien auf den Server laden (z.B. eine Textdatei). Kopieren Sie diese dazu in Ihr FTP-Verzeichnis und ändern Sie die Rechte so, dass others Leserechte für die Dateien haben. Per Browser sollten diese Dateien nach vorheriger Anmeldung verfügbar sein.

```
1 ftp://192.168.100.123
```

17. Legen Sie auf dem Entwicklungsrechner eine HTML Datei an mit dem Namen `index.html` und folgendem Inhalt an:

```
1 <html>
2 <body>
3 <H1>Meine erste Seite</H1>
4 <p>Das ist meine erste Website auf dem Pi</p>
5 </body>
6 </html>
```

18. Kopieren Sie nun per ftp die Datei vom Entwicklungsrechner auf den Pi. Unter Linux können Sie dafür das Programm `ftp` in der Konsole verwenden.

```
1 # ftp 192.168.100.123
2 # ...
3 ftp: send index.html
4 ftp: exit
```

19. Verschieben Sie die Datei in der SSH nun auf dem Pi in das Verzeichnis des Webservers und ändern Sie gegebenenfalls die Zugriffsrechte.

```
1 # sudo mv index.html /var/www/index.html
```

20. Wechseln Sie wieder per SSH auf den Pi und öffnen Sie dort nun mit dem nano die Datei und nehmen Sie Änderungen darin vor. Speichern Sie diese und lassen Sie sich die Änderungen im Browser anzeigen.
21. Verschieben Sie die Datei in der SSH nun auf dem Pi in das Verzeichnis des Webservers und ändern Sie gegebenenfalls die Zugriffsrechte.

```
1 # sudo nano /var/www/index.html
```

## 11.8 Erweiterte Funktionalität

Um den Webserver etwas auszubauen, wollen wir die Grundlage schaffen, um Datenbanken nutzen und auch dynamische Elemente nutzen zu können.

1. Wir installieren einige Software für den apache nach: libapache2-mod-php5 , php5 , php-pear , php5-xcacha. Diese erlaubt es uns Programme in php laufen zu lassen.

```
1 # sudo apt-get install libapache2-mod-php5
2 # sudo apt-get install php5
3 # sudo apt-get install php-pear
4 # sudo apt-get install php5-xcacha
```

2. Einen Datenbankserver richten wir ein mit php5-mysql , mysql-server und mysql-client ein.

```
1 # sudo apt-get install php5-mysql
2 # sudo apt-get install mysql-server
3 # sudo apt-get install mysql-client
```

3. Um den DB-Server nun auch bequem per Weboberfläche bedienen zu können, richten wir hier phpMyAdmin ein. Wir erlauben das Konfigurieren der Datenbank während der Installation.

```
1 # sudo apt-get install phpmyadmin
```

4. Um nun PHP zu testen, wechseln wir in /var/www und legen dort eine Datei phpinfo.php mit dem nano an.

```
1 # cd /var/www
2 # sudo nano phpinfo.php
```

5. Der Inhalt der Datei ist ein einfaches PHP Statement:

```
1 <?php
2 phpinfo ();
3 ?>
```

6. Rufen Sie nun diese Datei mit dem Browser auf Ihrem Entwicklungssystem auf. Dort sollen Sie nun eine Website mit Infos zum System sehen.

```
1 192.168.100.123/phpinfo.php
```



7. Um unsere Datenbank via Web bedienen zu können brauchen wir noch eine Änderung der Apache Konfiguration `/etc/apache2/apache2.conf`.

```
1 # sudo nano /etc/apache2/apache2.conf
```

8. Dort fügen Sie folgende Zeile am Ende der Datei ein:

```
1 Include /etc/phpmyadmin/apache.conf
```

9. Die Änderung der Konfiguration wird mit einem Neustart des Webservers oder des Systems wirksam.

```
1 # sudo /etc/init.d/apache2 restart
```

10. Mit Aufruf des Unterordners `/phpmyadmin` über den Browser sollte nun die Datenbankadministration zugänglich sein. Der Standardnutzer lautet `root` und das Passwort wurde bei der Installation vergeben und ist alternativ leer (Vorsicht: ein leeres Passwort für den Datenbankadministrator ist ein erhebliches Sicherheitsrisiko.)

```
1 192.168.100.123/phpmyadmin
```

11. Mit dieser Konfiguration lässt sich nun eine eigene Website (bisher nur als Intranet möglich) ein Blog, wie Wordpress, das persönliche Mediawiki oder ähnliche PHP und Datenbank basierte Anwendungen betreiben.