

Himbeere im Hintergrund

Raspi als Kiosksystem

Digitale Anzeigelösungen laufen oft auf klassischer PC-Hardware. Versteckt in Industrieschränken und geschützten Gehäusen verplempern völlig überdimensionierte Systeme ihr Potenzial mit der Darstellung eines simplen Dashboards. Das schaffen Sie mit einem Kiosk-Raspi besser und sparsamer.

Von Tomas Jakobs

Kiosksysteme begegnen einem im öffentlichen Raum meistens als Einzweck-Terminals, die entweder passiv Informationen darstellen (Digital Signage), beispielsweise als Anzeigetafeln am Flughafen, oder als Selbstbedienungsterminals simple Nutzereingaben verarbeiten, etwa um die Warteschlange im Bürgeramt zu organisieren. Ein Kiosksystem sollte leise, zuverlässig und möglichst unsichtbar seinen Dienst verrichten. Wir zeigen, wie ein Raspi mit einem auf Kiosk getrimmten Debian diese Herausforderung meistert. Für die Anleitung brauchen Sie einen Raspi 3 oder neuer. Linux-Kenntnisse sind hilfreich, aber keine Voraussetzung.

Sie bekommen ein wartungsarmes Minimalsystem, das „still“ bootet und einen Webbrowser im Kiosk-Modus, also im Vollbild und ohne unnötige Bedienelemente öffnet. Praktisch: Wenn Sie den Kiosk-Raspi erst einmal konfiguriert haben, können Sie von der SD-Karte ein Image anfertigen, das Sie stets wieder verwenden können, etwa falls Sie Ihre Flotte an Kiosk-Raspis erweitern wollen oder doch mal eine SD-Karte den Geist aufgibt.

Spar-Beere

Einplatinenrechner wie der Raspi mit seinem ARM-Prozessor punkten durch ihre Effizienz. Das fertig konfigurierte Kiosksystem schluckt nicht mehr als 5 Watt, das ergibt im Rund-um-die-Uhr-Betrieb im Jahr nur ungefähr 44 Kilowattstunden und schlägt bei einem Strompreis von 0,40 Euro/kWh mit Stromkosten von etwa 17,50 Euro zu Buche. Der Minicomputer im Scheckkartenformat lässt sich gut hinter Monitoren und Anzeigetafeln verstecken. Wo zuvor sperrige Gerätschaften

aufwendig transportiert und bevorratet wurden, lässt sich ein Kiosk-Raspi mit Stückkosten um die 60 Euro in einem Brief verschicken. Leider macht der Chipmangel auch nicht vor dem Raspi Halt. Je nach Shop müssen Sie aktuell mit langen Lieferzeiten oder aufgeblasenen Preisen rechnen.

Das ist allerdings immer noch günstiger, als einen ausgewachsenen PC anzuschaffen und zu betreiben. Die erzielbaren Kosteneinsparungen lassen Geschäftsführer, Buchhalter und Sparfüchse aufhorchen – in Zeiten hoher Strompreise und der Forderung nach einem möglichst nachhaltigen Umgang mit Ressourcen mehr denn je.

Grundinstallation von Debian

Für den Raspi gibt es vorgefertigte Images, die voll und ganz auf Digital Signage zugeschnitten sind. Eine solche Lösung ist „info-beamer“, die wir unter ct.de/y72d verlinkt haben. Diese Komplettlösungen sind jedoch je nach Funktionsumfang kostenpflichtig und hängen oft an der Cloud des Herstellers. Wenn der sein Projekt plötzlich einstampft, guckt man unter Umständen in die Röhre. Die Wahl des Betriebssystems fiel auf das freie Debian GNU/Linux, weil es besonders auf Langlebigkeit und Stabilität ausgerichtet ist.

Laden Sie zunächst das passende Image für Ihren Raspi aus dem Downloadbereich der Debian-Website, den wir unter ct.de/y72d verlinkt haben. Für den produktiven Einsatz eignen sich die Stable-Versionen von Debian 11 (Bullseye). Wer aktuellere Pakete benötigt und bereit ist, mit einem System zu arbeiten, das sich in

ct kompakt

- Kiosksysteme verrichten ihre Arbeit möglichst unsichtbar.
- Ein Raspi-Debian mit minimaler grafischer Oberfläche startet Firefox im Kiosk-Modus.
- Sie konfigurieren ein wartungsarmes System, das Sie leicht klonen können.

der Entwicklung befindet, kann sich prinzipiell auch an Debian 12 (Bookworm) wagen. Diese Version versieht das Debian-Projekt jedoch derzeit mit der Markierung „unstable“.

Bei Debian gibt es nach dem ersten Start keinen Standardbenutzer, kein sudo und auch keine Tools, die durch eine Installation führen. Als Debian-Nutzer müssen Sie selbst Hand anlegen. Der Lohn für diese Mühen ist ein solides und schlankes System, frei von nicht benötigten Komponenten.

Die Images mit einer durchschnittlichen Größe um die 400 MByte sind schnell geladen. Bringen Sie das Image im Anschluss mit dem Tool Ihrer Wahl auf die MicroSD-Karte. Das klappt beispielsweise mit dem Raspberry Pi Imager (erhältlich für Windows, Linux, macOS), wenn Sie die Option „Eigenes Image“ wählen. Legen Sie anschließend die MicroSD-Karte in den Raspi und verbinden Sie ihn mit Bildschirm, Ethernet, Tastatur und der USB-Stromversorgung.

Beim ersten Bootvorgang melden Sie sich auf der Kommandozeile als root mit einem leeren Passwort an. Die erste Maßnahme nach dem Login: Setzen Sie mit dem Befehl `passwd` ein Root-Kennwort. Beachten Sie, dass das System noch nicht lokalisiert ist und der Raspi folglich nur die englische Tastaturbelegung kennt.

Um die weitere Konfiguration bequem von einem anderen Rechner aus erledigen zu können, richten Sie zunächst den Zugang mittels SSH (Secure Shell) ein. Bearbeiten Sie dazu mit dem vorinstallierten Editor nano die SSH-Konfiguration:

```
nano /etc/ssh/sshd_config
```

Entfernen Sie die Auskommentierungen und ändern Sie die folgenden Zeilen, um den SSH-Zugang mittels Passwortauthentifizierung zu erlauben:

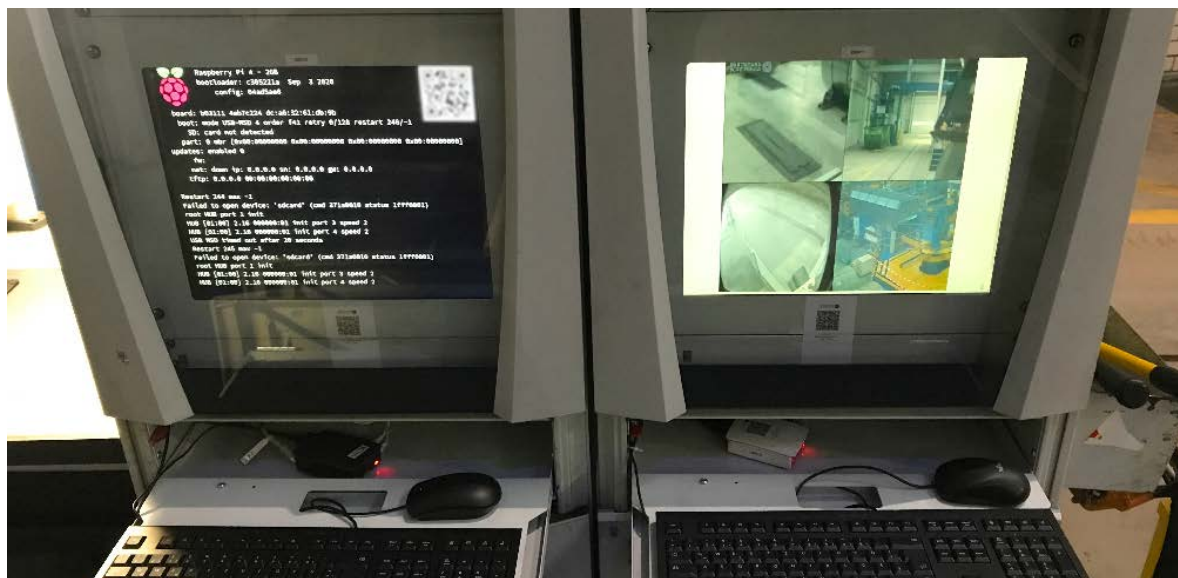
```
PasswordAuthentication yes  
PermitRootLogin yes
```

Mit Strg+O und Enter speichern Sie die Datei und verlassen den Editor mit Strg+X. Starten Sie jetzt den SSH-Dienst neu, um die Einstellungen zu übernehmen:

```
systemctl restart sshd
```

Ab jetzt können Sie sich von einem anderen Rechner im Netzwerk via SSH aufschalten – zum Beispiel von einem Windows-PC aus mit dem SSH-Client PuTTY – oder wahlweise weiter mit angeschlossener Tastatur fortfahren. Der Befehl `ip a show eth0` entlockt dem Raspi die IP-Adresse.

In der Industrie dienen Kiosksysteme beispielsweise zum Anzeigen von Video-Feeds.



Deutschkurs

Im nächsten Schritt hieven Sie das System mit apt auf den aktuellen Update-Stand:

```
apt update -y && apt upgrade -y
```

Mit den nachfolgenden Befehlen bringen Sie Ihrem Raspi Deutsch bei und richten die deutsche Tastaturbelegung ein:

```
apt install locales ↵  
↳keyboard-configuration ↵  
↳console-setup task-german -y
```

```
dpkg-reconfigure locales
```

```
dpkg-reconfigure ↵  
↳keyboard-configuration
```

Bei den Menü-Abfragen wählen Sie nacheinander „Deutsch“, „UTF8“ und „de_DE.UTF8“ aus. Die Frage nach der angeschlossenen Tastatur beantworten Sie mit „Generische 105-Tasten PC-Tastatur (Int.)“ und erneut mit „Deutsch“. Nach einem weiteren Neustart mit reboot steht die Lokalisierung.

Zweiten HDMI-Port freischalten

Raspis der vierten Generation haben zwei Micro-HDMI-Ports verbaut, von denen standardmäßig nur einer aktiv ist. Je nach Einsatzzweck bietet sich die Zuschaltung der zweiten HDMI-Schnittstelle an, beispielsweise wenn Sie mehrere Bildschirme ansteuern wollen und HDMI-Splitter oder ähnliches Zubehör fehlen. Beide HDMI-Anschlüsse parallel zu betreiben kommt mit einer Einschränkung: Eine Auflösung von 4k auf beiden Displays limitiert die Bildwiederholrate auf 30 Hertz, was aber für die

meisten Anwendungsfälle ausreichen dürfte, insbesondere im Kiosk-Betrieb.

Fügen Sie die folgenden Zeilen zur Datei /boot/firmware/config.txt hinzu, um den zweiten HDMI-Anschluss freizuschalten. Die Werte sollten für die meisten gängigen Displays funktionieren:

```
disable_splash=1  
hdmi_force_hotplug=1  
hdmi_drive=2  
hdmi_group=2  
hdmi_force_mode=1
```

Schweigsamer Start

Meistens will man auf die vorbeilaufenden Systemmeldungen beim Linux-Bootvorgang nicht verzichten, allein schon um eventuelle Probleme zu identifizieren. Wenn der Raspi jedoch als öffentlich einsehbares Kiosksystem betrieben wird, ist diese Art der Geschwätzigkeit unerwünscht. Mit etwas Feintuning bringen Sie den Kiosk-Raspi zu einem stillen Bootvorgang. Dazu tragen Sie mit nano die folgenden Bootparameter in die Datei /boot/firmware/cmdline.txt ein.

```
console=tty3 quiet loglevel=0 ↵  
↳logo.nologo ↵  
↳vt.global_cursor_default=0 ↵  
↳root=/dev/mmcblk1p2 rw ↵  
↳fsck.repair=yes net.ifnames=0 ↵  
↳rootwait
```

Die Ausgaben beim Systemstart werden nun auf ein nicht sichtbares Terminal umgeleitet. Der blinkende Cursor und das farbige Raspi-Logo sind deaktiviert. Ein Hinweis zu dem Parameter root=/dev/

mmcblk1p2: Üblicherweise ist das die Root-Partition bei einem Raspi-Debian, wenn Sie es auf eine MicroSD-Karte befördern. Wenn Ihre Root-Partition anders lautet, beispielsweise root=LABEL=RASPIROOT, dann lassen Sie den Wert einfach unangetastet. Damit Sie nach dem stillen Start wieder über einen blinkenden Cursor verfügen, ergänzen Sie die Zeile setterm -cursor on in der Datei /root/.bashrc.

Ihr Raspi bootet jetzt deutlich „leiser“. Fehlermeldungen oder neu entdeckte Gerätschaften führen mitunter trotzdem zu sichtbaren Meldungen. Passen Sie deswegen die Konfigurationsdateien für systemd und udev an, um deren Ausgaben zu unterdrücken beziehungsweise umzuleiten. Tragen Sie in /etc/sysctl.d/99-sysctl.conf beim Schlüssel kernel.printk = den Wert 3 3 3 3 ein. In /etc/udev/udev.conf ersetzen Sie udev_log=info durch udev_log=err. Zum Schluss tauschen Sie in /etc/systemd/logind.conf die Zeile NAutoVTs=6 gegen NAutoVTs=1. Entfernen Sie jeweils die vorangestellten Kommentarzeichen.

Auto-Kiosk

Im nächsten Schritt richten Sie einen Standardbenutzer ein, der automatisch ohne eine Abfrage eines Passworts angemeldet wird. Das Konto darf und soll keine Privilegien erhalten, sondern erfüllt einzig und allein die Aufgabe, einen Webbrowser zu öffnen. Erstellen Sie dazu einen neuen Benutzer mit dem Namen kiosk und vergeben ein Passwort:

```
useradd -m kiosk  
passwd kiosk
```

Die automatische Anmeldung des Benutzers kiosk teilen Sie dem System mit agetty mit. Legen Sie dazu zunächst mit dem Befehl mkdir -p /etc/systemd/system/getty@tty1.service.d das nötige Verzeichnis an und erstellen Sie dann in dem Ordner eine Datei namens override.conf mit folgendem Inhalt:

```
[Service]  
ExecStart=  
ExecStart=/sbin/agetty ↵  
↳--skip-login --login-options ↵  
↳-f kiosk" %I 38400 linux
```

Um die Willkommensnachricht („Message-of-the-Day“) zu unterdrücken, legen Sie eine leere Datei mit Namen .hushlogin im Home-Verzeichnis des Benutzers kiosk an:

```
<?xml version="1.0" encoding="UTF-8"?>  
<openbox_menu xmlns="http://openbox.org/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://openbox.org/ file:///usr/share/openbox/menu.xsd">  
  <menu id="root-menu" label="Openbox">  
    <item label="Web-Browser">  
      <action name="Execute">  
        <execute>x-www-browser</execute>  
      </action>  
    </item>  
  </menu>  
</openbox_menu>
```

Das Openbox-Menü wird mit der Datei /etc/xdg/openbox/menu.xml konfiguriert. Für das Kiosksystem beschränkt es sich auf das absolut Notwendige, um einen Webbrowser zu starten.

```
touch /home/kiosk/.hushlogin
```

Kiosk-Puzzle

Die Grundinstallation ist abgeschlossen. Jetzt gilt es, einen Firefox im grafischen Modus zu starten und eine vorgegebene Website im Kiosk-Modus ohne Menüleisten anzuzeigen. Dafür benötigen Sie einen X-Server und einen Window-Manager, aber ohne den Ballast eines ausgewachsenen Desktops. Openbox bietet eine solche minimale Umgebung und kommt ohne Auswahl- oder Einstellmöglichkeiten in der Bedienoberfläche aus. Praktisch: Die Konfiguration besteht lediglich aus zwei XML-Dateien, was eine reproduzierbare Umgebung für weitere Geräte oder Anwendungen schafft. Installieren Sie zunächst die notwendigen Pakete:

```
apt install --no-install-recommends ↵
xserver-xorg x11-xserver-utils ↵
xinit firefox-esr-110n-de openbox -y
```

Legen Sie dann ein Backup der ursprünglichen Openbox-Konfigurationsdatei an:

```
mv /etc/xdg/openbox/menu.xml ↵
cp /etc/xdg/openbox/menu.xml.backup
```

Erstellen Sie jetzt die Blaupause für das Openbox-Menü, indem Sie mit einem Texteditor die im Kasten auf Seite 163 gezeigten Bausteine in die Datei `/etc/xdg/openbox/menu.xml` eintragen. Durch Verzicht auf weitere Tools, beispielsweise eine grafische Terminal-Emulation, bleibt das Gesamtsystem schlank und die Openbox-XML-Datei für das Menü übersichtlich.

Openbox konfrontiert den Anwender nur mit wenigen Menüs und Texten. Trotzdem sollten die Ausgaben in deutscher Lokalisierung erfolgen. Dazu ändern Sie in der Environment-Datei `/etc/xdg/openbox/environment` die Zeile `#LANG=en_CA.UTF8` zu `LANG=de_DE.UTF8`.

Um Buttons und virtuelle Desktops vor dem Standardbenutzer zu verbergen, legen Sie eine weitere Openbox-XML-Datei als Kopie in dessen Home-Verzeichnis:

```
mkdir -p /home/kiosk/.config/openbox
cp /etc/xdg/openbox/rc.xml ↵
cp /home/kiosk/.config/openbox/rc.xml
```

Im Anschluss schreiben Sie in der Datei `rc.xml` die Zeile `<titleLayout>NLIMC</titleLayout>` zu `<titleLayout>L</titleLayout>` um.



Im mageren Openbox-Menü kann man sich nicht verirren. Es dient nur dazu, einen Browser zu öffnen.

`titleLayout>` und `<number>4</number>` zu `<number>1</number>` um.

Starten Sie jetzt den Raspi neu, um die bisherige Konfiguration zu testen. Es begrüßt Sie eine leere Shell, in der Sie automatisch als Benutzer `kiosk` angemeldet sind. Starten Sie die minimale grafische Umgebung:

```
startx
```

X-Server und Openbox zeigen einen leeren, schwarzen Desktop. Mit einem Rechtsklick öffnen Sie das zuvor angelegte Openbox-Menü mit dem einzigen Eintrag „Web-Browser“. Öffnen Sie damit Firefox, klicken Sie alle Willkommensnachrichten weg und richten Sie den Browser so ein, wie Sie ihn haben möchten. Empfehlenswert ist beispielsweise, den Session-Restore nach einem Crash zu deaktivieren.

Start ohne Stützräder

Funktioniert alles wie gewünscht, setzen Sie abschließend die Bausteine zu einem Ganzen zusammen. Der Kiosk-Modus soll künftig nicht mehr auf Ihr Zutun angewiesen sein, sondern vollautomatisch anspringen. Tragen Sie dazu folgende Zeile in `/home/kiosk/.profile` ein:

```
exec startx -- -keeptty -nocursor ↵
cp ~/.xorg.log 2>&1
```

Wenn Ihr Kiosk-Raspi nicht nur passiv etwas darstellen, sondern Eingaben mit der Maus entgegennehmen soll oder Sie Ihren Kiosk-Raspi im Verbund mit einem Touchscreen betreiben, dann verzichten Sie auf den Parameter `-nocursor`.

Nun soll der Kiosk-Firefox noch automatisch starten und eine vorgegebene URL aufrufen. Legen Sie dazu zunächst wieder eine Sicherheitskopie an:

```
mv /etc/xdg/openbox/autostart ↵
cp /etc/xdg/openbox/autostart.backup
```

Dann tragen Sie mit `nano` folgende Zeilen in `/etc/xdg/openbox/autostart` ein:

```
xset -dpms
xset s off
xset s noblank
firefox https://heise.de/ct --kiosk &
```

Ersetzen Sie `https://heise.de/ct` durch die URL, die auf die Inhalte verweist, die Sie darstellen wollen. Das kann auch eine Webanwendung, ein Dashboard oder ein Ticket-System sein. Die anderen Parameter verhindern, dass der Raspi ungewollt in den Energiesparmodus wechselt. Ihr Raspi-Kiosk ist einsatzbereit: Ab sofort öffnet der Raspi nach jedem Neustart die Website ohne störende Systemmeldungen, Menüleisten oder Benutzerinteraktionen.

Fazit

Kiosksysteme sollen in der Regel unbeaufsichtigt laufen. Je nach Anwendungsfall und Anwendungsort sollten Sie in Erwägung ziehen, das System weiter abzusichern, beispielsweise indem Sie die SSH-Authentifikation mit Passwort deaktivieren und nur mit Zertifikat erlauben. Mit dem Programm `usbguard` verhindern Sie, dass unbekannte USB-Geräte Schaden anrichten, aber besser Sie schließen den Raspi weg, damit Unbefugte keinen physischen Zugriff erlangen können.

Mit Cronjobs gönnen Sie Ihrem Kiosk-Raspi gelegentliche Verschnaufpausen, beispielsweise in Form eines nächtlichen Reboot, oder prüfen, ob der Firefox nicht abgestürzt ist oder geschlossen wurde. Wenn Sie die SD-Karte schonen wollen, schalten Sie das Dateisystem in den Read-Only-Modus [1]. Der verhindert auch, dass das Dateisystem durch einen plötzlichen Stromausfall Schaden nimmt. Vom automatischen Abgleich der Systemzeit über das Monitoring bis hin zum eigenen Branding bleibt viel Raum für weitere Anpassungen.

(ndi@ct.de) **ct**

Literatur

[1] Mirko Dölle, Vollschatz, Raspberry Pi mit schreibgeschütztem Linux, ct 16/2021, S. 168

Debian-Images für den Raspberry Pi, „info-beamer“-Digital-Signage: ct.de/y72d