

Initial installation of Raspberry Pi 4B

=====

1. Prepare SD card

- format with the SD formatter: <https://www.sdcard.org/downloads/formatter/>
- Download 'NOOBS 3.2.1.zip' as a zip archive:
<https://www.raspberrypi.org/downloads/noobs/>
- unzip and copy to the SD card

2. Insert SD card into the Raspberry and boot

- Language/Character set/Keyboard: DE/UTF-8/DE
- Settings:
 - * Change default user (pi/raspberrypi): pi/zwu8pm
 - * Enter computer name: Triton
 - * Enter Wi-Fi SSID and Wi-Fi password
- Install updates
- Reboot

2a. Reset an already installed system:

Press the SHIFT key during startup: --> Configuration menu

3. Check or change settings in the Sppedport W724V:

Wi-Fi connection: 192.168.2.103, Name: Triton-WLAN

LAN connection: 192.168.2.105, Name: Triton-LAN

2. Establish an SSH connection:

- on the Raspberry: create a file 'SSH' in the root directory
<https://www.elektronik-kompndium.de/sites/raspberry-pi/1906281.htm>
- under Windows:
In the DOS box: SSH p@Triton-WLAN , then enter username and password
or: Putty: Enter connection data and user name 'pi'

3. Establish RDP connection:

- Install RDP server on the Raspberry Pi
<https://www.elektronik-kompndium.de/sites/raspberry-pi/2109031.htm>
 - * sudo apt-get update
 - * sudo apt-get install xrdp
 - * sudo systemctl status xrdp
 - * sudo shutdown 0
- Start RDP under Windows: %windir%\system32\mstsc.exe
Connection data and user name and enter
- delete old connection entries under:
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default

4. Java and JavaFX

=====

4.1 Determine installed version:

```
$ java -version
openjdk version "11.0.5" 2019-10-15
OpenJDK Runtime Environment (build 11.0.5+10-post-Raspbian-1deb10u1)
OpenJDK Server VM (build 11.0.5+10-post-Raspbian-1deb10u1, mixed mode)
```

4.2 Manual installation of other Java versions:

Source: <http://www.gridtec.at/java-8-auf-dem-raspberry-pi-installieren-und-testen/>

- Use Java from Oracle "Linux ARM 32 Hard Float ABI", downloadable from here:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

- Copy 'jdk-8u231-linux-arm32-vfp-hflt.tar.gz' to the Raspberry, e.g. to /home/pi

- unpack to /opt:

```
$ sudo tar xvfz jdk-8-linux-arm-vfp-hflt.gz -C /opt
```

- activate:

```
sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_33/bin/javac 1
```

```
sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_33/bin/java 1
```

- if you have multiple Java versions, select the newly installed version as the default version with:

```
$ sudo update-alternatives --config java
```

```
$ sudo update-alternatives --config javac
```

- display active version with:

```
$ java -version
```

- delete downloaded archive:

```
$ rm jdk-8-linux-arm-vfp-hflt.tar.gz
```

- Compile and run 'HelloWorld':

```
HelloWorld.java:
```

```
-----
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello Gridtec!");
    }
}
-----
```

```
$ javac HelloWorld.java
```

```
$ java HelloWorld
```

4.3 Manual installation of JavaFX in Oracle Java (from JDK 8)

Source: <https://stackoverflow.com/questions/40481455/running-javafx-gui-on-the-raspberry-pi/40483500#40483500>

- Download 'JavaFX Embedded SDK' from here:
<https://gluonhq.com/products/mobile/javafxports/get/>

- Copy 'jdk-8u231-linux-arm32-vfp-hflt.tar.gz' to the Raspberry, e.g. to /home/pi
Copy armv6hf-sdk-8.60.12.zip to the Raspberry, e.g. to /home/pi

- unpack to /tmp/armv6hf-sdk: provides the following directory structure:

```
armv6hf-sdk
\lib
javafx-mx.rar
\rt
\arm
.....
\ext
jfxrt.jar
jfx.platform.properties
javafx.properties
jfxswt.jar
```

- Transfer files to the Oracle Java version:

```
$ cd tmp
$ sudo chown -R root:root armv6hf-sdk
$ cd armv6hf-sdk
$ sudo mv lib/javafx-mx.jar /opt/jdk1.8.0_111/lib/
$ cd rt/lib/
$ sudo mv j* /opt/jdk1.8.0_111/jre/lib/
$ sudo mv arm/* /opt/jdk1.8.0_111/jre/lib/arm/
$ sudo mv ext/* /opt/jdk1.8.0_111/jre/lib/ext/
```

4.4 Selecting a Java version from several installed versions

```
-----
sudo update-alternatives --config java
```

4.5 Starting JavaFX programs:

- Copy the following command (adapted) into a file, e.g. 'start.sh'

```
-----
/usr/lib/jvm/java-1.11.0-openjdk-armhf/bin/java --module-path /usr/share/openjfx/lib --
add-modules=javafx.base,javafx.controls,javafx.fxml -jar RasPiTrigger.jar
-----
```

- Adjust permissions: \$chmod +x start.sh

- start: ./start

4.6 Compiling and running Java programs:

- Example program 'Hello.java':

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello Gridtec!");  
    }  
}
```

- \$ javac Hello.java
\$ java Hello

5. Compiling C programs

gcc -o 01_dule_colorled 01_dule_color_led.c -lwiringPI

Erst-Installation Raspberry Pi 4B

=====

1. SD-Karte vorbereiten

- formatieren mit dem SD-Formater: <https://www.sdcard.org/downloads/formatter/>
- 'NOOBS 3.2.1.zip' downloaden als zip-Archiv:

<https://www.raspberrypi.org/downloads/noobs/>

- entpacken und auf die SD-Karte kopieren

2. SD-Karte in den Raspberry stecken und booten

- Sprache/Zeichensatz/Tastatur: DE/ UTF-8/DE
- Einstellungen:
 - * Default-Benutzer (pi/raspberrypi) umstellen: pi/zwu8pm
 - * Rechner-Name eingeben: Triton
 - * WLAN-SSID und WLAN-Passwort eingeben
- Updates nachinstallieren
- Reboot

2a. Bereits installiertes System zurücksetzen:

Beim Startup die SHIFT-Taste drücken: --> Konfigurations-Menü

3. Einstellungen im Sppedport W724V überprüfen bzw. ändern:

WLAN-Verbindung: 192.168.2.103, Name: Triton-WLAN

LAN-Verbindung: 192.168.2.105, Name: Triton-LAN

2. SSH-Verbindung herstellen:

- auf den Raspberry: eine Datei 'SSH' im Root-Verzeichnis anlegen
<https://www.elektronik-kompendium.de/sites/raspberry-pi/1906281.htm>
- unter Wiondows:
In der DOS-Box: SSH p@Triton-WLAN , danach Benutzername und Passwort eingeben
oder: Putty: Verbindungsdaten und Benutzername 'pi' eingeben

3. RDP-Verbindung herstellen:

- RDP-Server auf dem Raspberry Pi installieren
<https://www.elektronik-kompendium.de/sites/raspberry-pi/2109031.htm>
 - * sudo apt-get update
 - * sudo apt-get install xrdp
 - * sudo systemctl status xrdp
 - * sudo shutdown 0
- RDP unter Windows starten: %windir%\system32\mstsc.exe
Verbindungsdaten und Benutzername und eingeben
- alte Verbindungs-Einträge löschen unter:
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default

4. Java und JavaFX

=====

4.1 Installierte Version feststellen:

```
$ java -version
openjdk version "11.0.5" 2019-10-15
OpenJDK Runtime Environment (build 11.0.5+10-post-Raspbian-1deb10u1)
OpenJDK Server VM (build 11.0.5+10-post-Raspbian-1deb10u1, mixed mode)
```

4.2 Manuelle Installation anderer Java-Versionen:

Quelle: <http://www.gridtec.at/java-8-auf-dem-raspberry-pi-installieren-und-testen/>

- Java von Oracle "Linux ARM 32 Hard Float ABI" verwenden, downloadbar von hier:
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- 'jdk-8u231-linux-arm32-vfp-hflt.tar.gz' auf den Raspberry kopieren, z.B. nach /home/pi

- entpacken nach /opt:
\$ sudo tar xvfz jdk-8-linux-arm-vfp-hflt.gz -C /opt

- aktivieren:
sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_33/bin/javac 1
sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_33/bin/java 1

- bei mehreren Java-Versionen die neu installierte Version als Default-Version auswählen mit:

```
$ sudo update-alternatives --config java
$ sudo update-alternatives --config javac
```

- aktive Version anzeigen lassen mit:
\$ java -version

- heruntergeladenes Archiv löschen:
\$ rm jdk-8-linux-arm-vfp-hflt.tar.gz

- 'HelloWorld' compilieren und ausführen:
HelloWorld.java:

```
-----
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello Gridtec!");
    }
}
-----
```

```
$ javac HelloWorld.java
$ java HelloWorld
```

4.3 Manuelle Installation von JavaFX in Oracle Java (ab JDK 8)

Quelle: <https://stackoverflow.com/questions/40481455/running-javafx-gui-on-the-raspberry-pi/40483500#40483500>

- 'JavaFX Embedded SDK' downloaden von hier:
<https://gluonhq.com/products/mobile/javafxports/get/>
- 'jdk-8u231-linux-arm32-vfp-hflt.tar.gz' auf den Raspberry kopieren, z.B. nach /home/pi
armv6hf-sdk-8.60.12.zip auf den Raspberry kopieren, z.B. nach /home/pi
- entpacken nach /tmp/armv6hf-sdk: liefert die folgende Verzeichnis-Struktur:


```

armv6hf-sdk
├── lib
│   └── javafx-mx.rar
├── rt
│   └── arm
│       └── .....
└── ext
    ├── jfxrt.jar
    ├── jfx.platform.properties
    ├── javafx.properties
    └── jfxswt.jar
      
```

- Dateien in die Oracle-Java-Version übertragen:


```

$ cd tmp
$ sudo chown -R root:root armv6hf-sdk
$ cd armv6hf-sdk
$ sudo mv lib/javafx-mx.jar /opt/jdk1.8.0_111/lib/
$ cd rt/lib/
$ sudo mv j* /opt/jdk1.8.0_111/jre/lib/
$ sudo mv arm/* /opt/jdk1.8.0_111/jre/lib/arm/
$ sudo mv ext/* /opt/jdk1.8.0_111/jre/lib/ext/
      
```

4.4 Java-Version unter mehreren installierten auswählen

```

-----
sudo update-alternatives --config java

```

4.5 starten von JavaFX-Programmen:

- Folgenden Befehl (angepasst) in eine Datei, z.B. 'start.sh' übertragen


```

-----
/usr/lib/jvm/java-1.11.0-openjdk-armhf/bin/java --module-path /usr/share/openjfx/lib --
add-modules=javafx.base,javafx.controls,javafx.fxml -jar RasPiTrigger.jar
-----

```
- Rechte anpassen: `$ chmod +x start.sh`
- starten: `./start`

4.6 Java-Programme übersetzen und ausführen:

- Beispielprogramm 'Hello.java':


```

-----

```

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello Gridtec!");  
    }  
}
```

```
- $ javac Hello.java  
$ java Hello
```

5. C-Programme compilieren

```
gcc -o 01_dule_colorled 01_dule_color_led.c -lwiringPI
```