Versuch:

Server-installation

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Studiengänge:

- Informationssystemtechnik/
  Multimediatechnik/
  Elektrotechnik
  Media and Acoustical Engineering

Ausbildungsziel:

- Kennen lernen von allgemeinen administrativen Aufgaben zur Verwaltung von Webserver, FTP-Server, VNC-Server
- Realisierung einfacher Hardwareansteuerung (433MHz-Sender, Temperatur-Sensor) über eine HTTPS-Schnittstelle

Ausbildungsinhalte:

- Installation von WWW-Server, FTP-Server, VNC-Server
- Installation von PHP als Basis zur serverseitigen Scriptverarbeitung
- Realisierung einer HTTPS-Verbindung durch Erzeugen eines privaten SSL-Zertifikates
- Erzeugen eines einfachen PHP-Scriptes zum Ein- und Ausschalten von 433MHz-Funksteckdosen
- Erzeugen eines einfachen PHP-Scriptes zum Einlesen einer Raumtemperatur und dessen Bereitstellung über HTTPS

Gerätetechnik:

- 1 Raspberry Pi
- 1 433MHz-Sender
- 1 PC für die Administration des Raspberry Pi per SSH und VNC

Vorkenntnisse:

- Kenntnisse im Bereich Grundlagen der KT

Versuchsumfeld

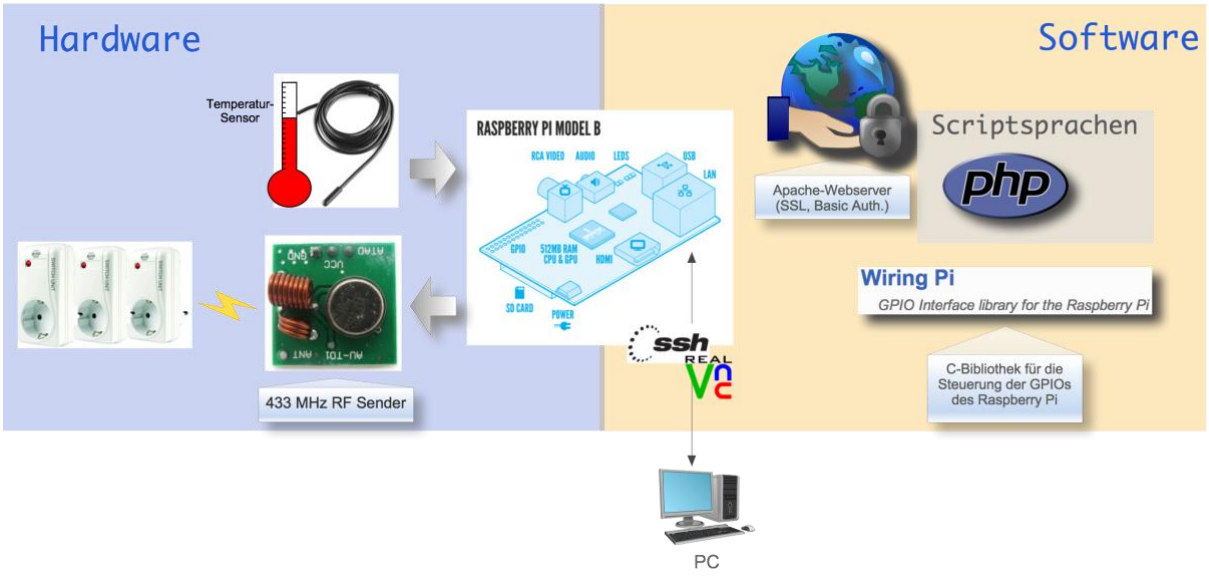


Abbildung 1: Versuchsumfeld

Einführung

Viele Geräte beinhalten zunehmend einen Webserver, welcher aufgrund des standardisierten Anwendungsprotokolls HTTP eine problemlose Administration mittels PC, Tablet oder auch Smartphone ermöglicht. Durch das weltweite routen von HTTP-Paketen ist darüberhinaus ein Zugriff auf die Geräte auch von außerhalb des eigenen Netzwerkes möglich. Desweiteren bieten standardisierte Lösungen zur Datenverschlüsselung (z.B. HTTPS) und Authentifizierung (z.B. Basic Authentication) die Möglichkeit, den Zugriff auf die Daten zu beschränken.

Ziel dieses Versuches soll es sein, auf Basis eines Raspberry Pi eine einfache Haussteuerung per Webinterface zu realisieren. Der Raspberry Pi ist ein von der gleichnamigen Raspberry Pi Foundation entwickelter Mini-Computer in der Größe einer Kreditkarte. Aufgrund des geringen Stromverbrauchs und der überschaubaren Anschaffungskosten von ca. 40,- € eignet er sich gut zum Einsatz als Steuereinheit für Hausautomationsaufgaben. Wie in Abbildung 1: Versuchsumfeld ersichtlich, verfügt der Mini-Computer über Audio-, USB-, LAN- und HDMI-Schnittstellen. Als Betriebssystem kommt eine ARM-optimierte Linux-Version (Debian) zum Einsatz, was die problemlose Installation von Standardkomponenten, wie z.B. Webserver, Datenbankserver, FTP-Server oder VNC-Server ermöglicht. Darüber hinaus können Applikationen mittels verschiedenster Programmiersprachen umgesetzt werden, wie z.B. C++, Java, Python oder PHP.

Aufbauend auf diesen Versuch, soll in einem zweiten Teil eine Smartphone-App zum Ein- und Ausschalten von Haushaltsgeräten entworfen und programmiert werden.

Versuchsdurchführung

Docker-Container erzeugen

1. Öffnen Sie den URL https://www.staff.hs-mittweida.de/~rthomane/intranet/praktikumkt/raspberry/getDockerContainer/
2. Melden Sie sich mir Ihrem Hochschul-Nutzername und Hochschul-Passwort an!
3. Anschließend wird automatisch für Sie ein Docker-Container mit einer Debian-Distribution erzeugt.

Diese Debian-Distribution verhält sich nahezu identisch zum Raspberry Pi-Betriebssystem. Alle Arbeitsschritte können somit auch auf einem Raspberry Pi durchgeführt werden. Kleine Abweichungen bei der Adminstration wurden in dieser Anleitung gekennzeichnet.

Verbindung mit Docker-Container herstellen

Zur Administration des Docker-Containers soll SSH eingesetzt werden. Secure Shell (SSH) steht für ein Protokoll, mit dem entsprechende Programme (Clients) den Zugriff und die Ausführung von Befehlen oder Aktionen auf einem entfernten Computer ermöglichen. Das Protokoll SSH gehört auf PCs und Servern mit Linux oder einem anderen Unix-ähnlichen Betriebssystem zu den fest installierten Standardwerkzeugen und wird von vielen Administratoren bevorzugt eingesetzt, um einen Computer per Remote-Zugriff zu administrieren. SSH besitzt keine grafische Benutzeroberfläche (GUI), weshalb es sehr effizient arbeitet und nur wenige Ressourcen belegt.

Nach erfolgreicher Erstellung des Docker-Containers werden Ihnen auf der gerade geöffneten Webseite die für Ihren Docker-Container erzeugten Zugangsdaten angezeigt. Führen Sie nun folgenden Schritte durch, um eine Verbindung zum Linux-Betriebssystem (Debian) des Docker-Containers herzustellen.

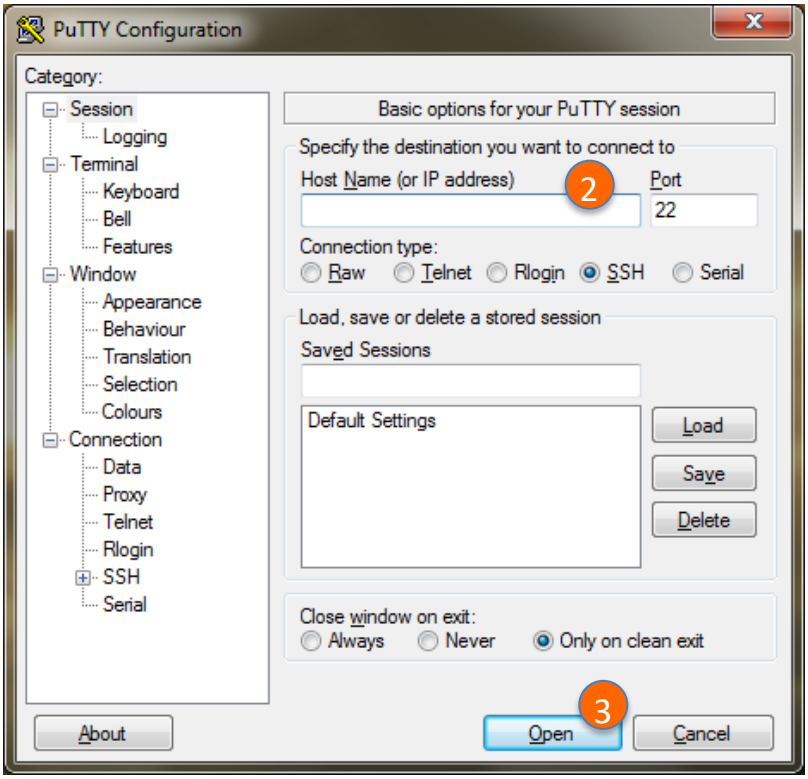
1. Installieren Sie (falls benötigt) die für Ihr Betriebssystem notwendige SSH-Client-Anwendung!
2. Starten Sie die Anwendung und tragen Sie die benötigten Zugangsdaten (Hostname, Port, Nutzername und Passwort) in die zugehörigen Eingabefelder ein!
3. Starten Sie anschließend die Verbindung!

Sollten Sie ein Mac OS oder Linux als Betriebssystem einsetzen, benötigen Sie keine zusätzlichen SSH-Client-Anwendungen. Wie auf der Webseite beschrieben, können Sie für die SSH-Verbindung direkt die Terminal-Applikation verwenden.

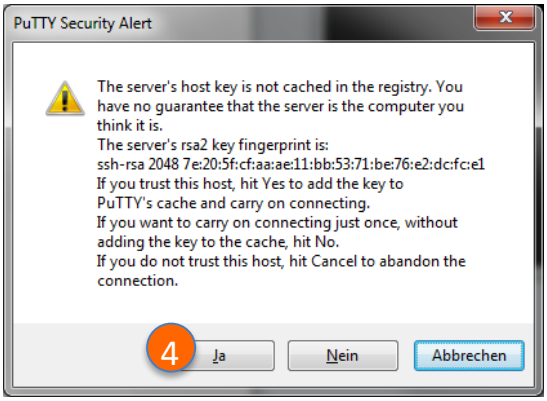
Die folgenden Schritte sind beispielhaft für die Putty-Anwendung unter Windows durchzuführen:

1. Starten Sie auf Ihren PC das Programm „PuTTy“!
2. Geben Sie im Feld „Host Name (or IP address)“ den aufgedruckten Hostname Ihres Raspberry Pi’s an, z.B. tc-raspberry-01.eit.hs-mittweida.de!

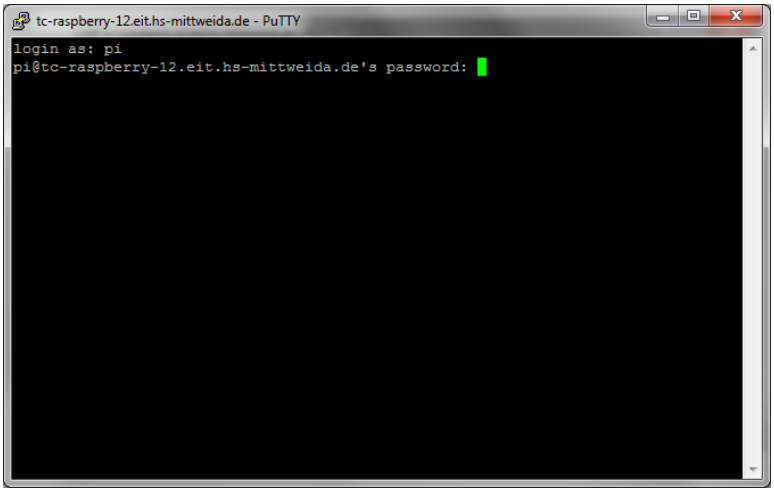
3. Klicken Sie anschließend auf „Open“!



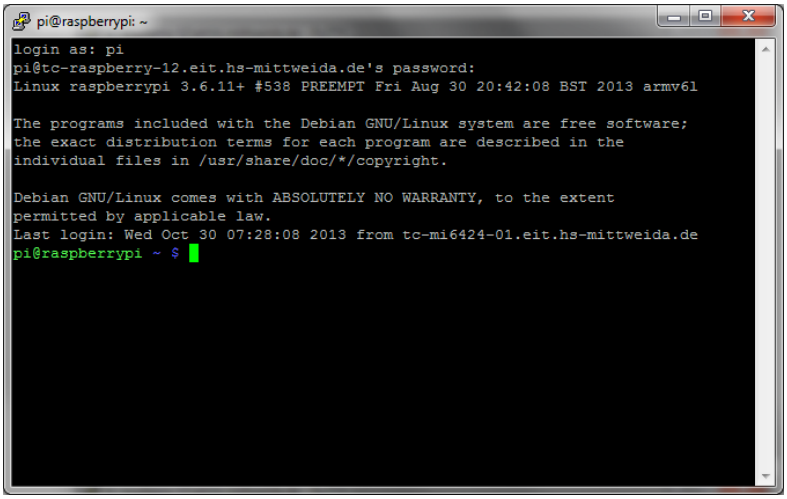
4. Die anschließende Warnung können Sie mit „Ja“ bestätigen.



5. Geben Sie bitte den Nutzername „pi“ ein und drücken Sie Enter!
6. Geben Sie bitte das Passwort „praktikumkt“ ein und drücken Sie Enter!



7. Sie sind nun mit der Kommandozeile des Raspberry Pi per SSH verbunden.

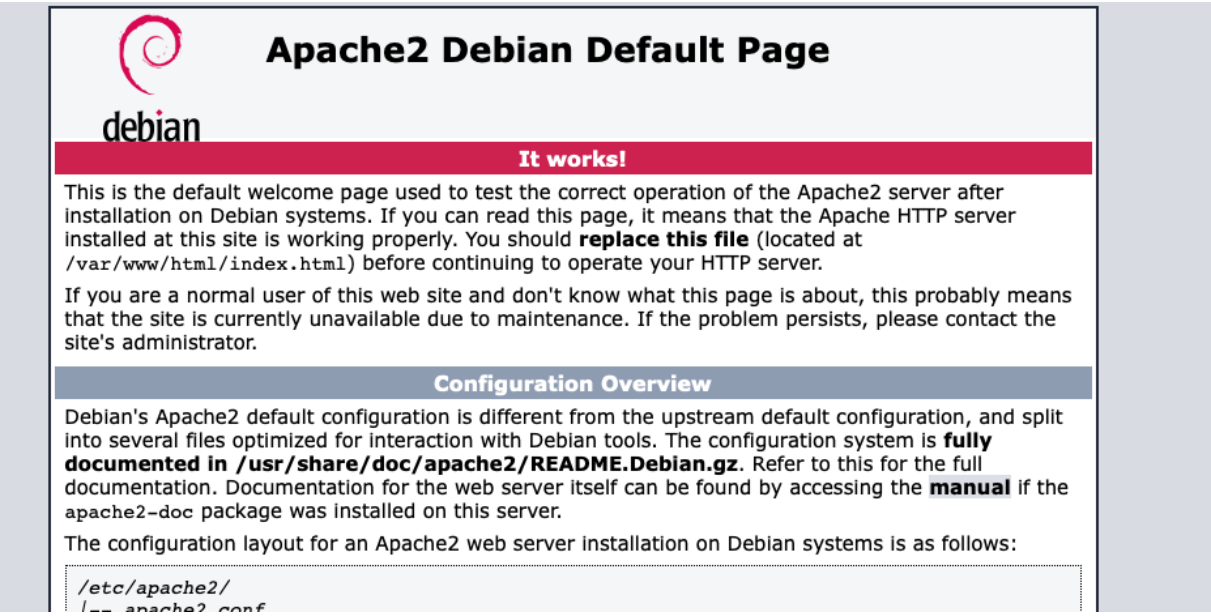


Einrichten des Webservers

Als Webserver wird Apache2 eingesetzt, welcher ein freier Open-Source Webserver ist. Apache ist ein freier Open-Source-Webserver. Der Apache-Webserver ist der beliebteste Webserver, der derzeit von etwa der Hälfte aller Webpräsenzen weltweit genutzt wird. Die erste Version wurde im Jahr 1995 veröffentlicht. Der Webserver wird von der Apache Software Foundation (ASF) entwickelt und verwaltet. Die vollständige Bezeichnung lautet Apache HTTP Server. Apache unterstützt dabei alle relevanten Funktionen, wie z.B. Auslieferung von Ressourcen per HTTP und HTTPS. Auch die Erweiterung um serverseitige Scriptsprachen wie z.B. PHP oder Python wird durch Apache unterstützt. Das Standartdateiverzeichnis für Apache unter Debian ist */var/www*, die Konfgurationsdatei befindet sich unter */etc/apache2/apache2.conf*. Zusätzliche Konfigurationen findet man unter */etc/apache2*.

1. Installieren Sie Apache2 mittels des Befehls:
   Achtung: Das Symbol „↵“ kennzeichnet nur einen Zeilenumbruch in dieser Anleitung. Es handelt sich demzufolge um einen Befehl, der in einer Zeile, ohne das Symbol „↵“, anzugeben ist!
   ```
   $ sudo apt-get install apache2 apache2-doc apache2-utils ↵
   apache2-mpm-prefork
   ```
2. Installieren Sie weiterhin die PHP-Erweiterung mittels der Befehle:
   ```
   $ sudo apt-get install php5 libapache2-mod-php5
   $ sudo /etc/init.d/apache2 restart
   ```
3. Das Wurzelverzeichnis des Webservers besitzt nach der Installation nur Schreibrechte für den root-Nutzer. Damit auch der Webserver selbst, welcher mit dem Gruppen- und Nutzerbezeichner „www-data“ auftritt, Änderungsrechte besitzt, sind folgende Befehle auszuführen:
   ```
   $ sudo chown www-data:www-data /var/www/html
   $ sudo chmod 775 /var/www/html
   $ sudo usermod –a –G www-data www-data
   $ sudo usermod -G www-data -a pi
   ```
4. Starten Sie nun den Webserver neu, um die Änderungen zu übernehmen!
   ```
   $ sudo /etc/init.d/apache2 restart
   ```

5. Starten Sie nun auf Ihrem PC einen Webbrowser und verwenden Sie die URL, die für Ihren Docker-Container angelegt wurde! Daraufhin müsste eine einfache Webseite erscheinen.



Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
```

6. Erzeugen Sie im Verzeichnis `/var/www/html` eine PHP-Datei mit der Bezeichnung „phpinfo.php“!

```
$ sudo nano /var/www/html/phpinfo.php
```

7. Fügen Sie in diese Datei folgenden Inhalt ein:

```
<?php
  phpinfo();
?>
```

8. Speichern Sie die Datei mittels „Strg+x“ und bestätigen Sie die Abfrage mit „J“!
9. Geben Sie in Ihrem Webbrowser als URL die Adresse Ihres Docker-Containers ein und fügen Sie anschließend den Bezeichner „/phpinfo.php“ hinzu:
   z.B.: http://mtm-kt01.hs-mittweida.de:50212/phpinfo.php
   Als Ergebnis sehen Sie eine Übersicht der Installierten PHP-Version. Würde PHP auf dem Webserver nicht richtig funktionieren, würde an dieser Stelle nur der Quellcode der Datei „phpinfo.php“ angezeigt werden.

PHP Version 5.6.40-0+deb8u12

System	Linux 8e78ea98e539 5.8.0-49-generic #55-Ubuntu SMP Wed Mar 24 14:45:45 UTC 2021 x86_64
Build Date	Jun 28 2020 09:17:16
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled

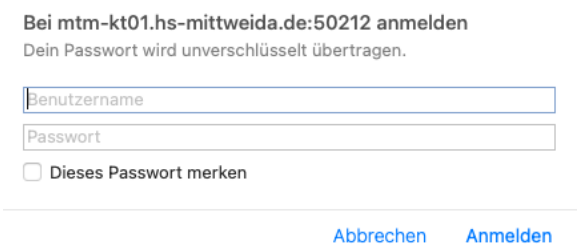
Passwort geschützter Zugang für Webserver einrichten

Um einen unberechtigten Zugang auf den Webserver zu unterbinden, soll das Wurzelverzeichnis des Webservers (`/var/www/html`) mit einer Basic-Authentikation gesichert werden.
Hierfür ist die Konfiguration des Apache-Webservers notwendig, was über die Konfigurationsdatei `/etc/apache2/sites-enabled/000-default.conf` erfolgt. Zuvor muss noch eine Passwort-Datei mittels des Befehls `htpasswd` angelegt werden, die üblicherweise als versteckte Datei im Verzeichnis /`etc/apache2/` als `.htpasswd`-Datei angelegt wird.

1. Erzeugen Sie eine `.htpasswd`–Datei im Verzeichnis `/etc/apache2`! Der Nutzername soll hierbei „**praktikum**“ lauten! Nach Ausführung des Befehls werden Sie aufgefordert das Passwort 2-mal einzugeben. Verwenden Sie hierfür das Passwort „**praktikumkt**“! Wenn die Passwort-Datei erfolgreich erzeugt wurde, wird angezeigt: „Adding password for user praktikum“.
   ```
   $ sudo htpasswd -c /etc/apache2/.htpasswd praktikum
   ```
2. Den Inhalt der .htpasswd-Datei können Sie sich mittels des folgenden Befehls anzeigen lassen:
   ```
   $ cat /etc/apache2/.htpasswd
   ```
3. Öffnen Sie die Apache-Konfigurationsdatei im Editor
   ```
   $ sudo nano /etc/apache2/sites-enabled/000-default.conf
   ```
4. Fügen Sie folgenden fett dargestellten Inhalt in die Zeile vor `</VirualHost>` ein!
   ```
   <VirtualHost *:80> …
   <Directory "/var/www/html">
      AuthType Basic
      AuthName "Restricted Content"
      AuthUserFile /etc/apache2/.htpasswd
      Require valid-user
   </Directory>
   </VirtualHost>
   ```

5. Starten Sie den Apache-Service neu!
   ```
   sudo /etc/init.d/apache2 restart
   ```

   Rufen Sie, zum Test, in Ihrem Webserver noch einmal die Ressource „phpinfo.php“ auf! Als Ergebnis müssten Sie nun eine Eingabeaufforderung für Nutzername und Passwort erhalten. Geben Sie als Nutzername „praktikum“ und als Passwort „praktikumkt“ ein!



SSL-Verschlüsselung für die HTTP-Verbindung

Im vorherigen Abschnitt haben Sie den Zugang auf den Webserver mit einem Passwort gesichert. Die mit HTTP übertragenen Daten werden allerdings immer noch unverschlüsselt transportiert. Dies gilt auch für die Anmeldesequenz, wodurch Nutzername und Kennwort ebenfalls im Klartext übertragen werden. Um das Mitlesen der gesendeten Daten zu unterbinden ist es sinnvoll, auf eine HTTPS-Verbindung umzustellen. Normalerweise sollten die für HTTPS notwendigen Zertifikate durch eine registrierte Zertifizierungsinstanz bestätigt werden. Hierfür sind allerdings u.a. Angaben zum Domain-Bezeichner notwendig. Darüber hinaus benötigt der Host einen Fully Qualified Domain Name (FQDN), wie z.B. www.homecontrol-raspberry.de. Dies würde demzufolge die Beantragung eines Domainbezeichners erfordern, welcher mit Kosten verbunden wäre.

Für den privaten Einsatz von HTTPS reicht es allerdings aus, ein selbst-signiertes Zertifikat zu verwenden, was in den folgenden Schritten durchgeführt werden soll. Dies führt allerdings zu einer Warnung im Webbrowser, dass die Identität des Webservers nicht bestätigt ist. Dennoch wird aber die gesamte Kommunikation erfolgreich verschlüsselt.

1. Aktivieren Sie das Apache SSL Modul:
   ```
   $ sudo a2enmod ssl
   ```
2. Die Standard-Apache-Website wird mit einer nützlichen Vorlage für die Aktivierung von SSL geliefert, weswegen jetzt die Standard-Website aktiviert wird!
   ```
   $ sudo a2ensite default-ssl
   ```
3. Starten Sie den Apache-Webserver neu!
   ```
   $ sudo /etc/init.d/apache2 restart
   ```
4. Erzeugen Sie ein Verzeichnis „`ssl`“ unter `/etc/apache2`!
   ```
   $ sudo mkdir /etc/apache2/ssl
   ```
5. Erstellen Sie ein Zertifikat mittels folgenden Befehls und folgen Sie den Eingabeaufforderungen!
   - Das Flag **days** gibt an, wie lange das Zertifikat gültig bleiben soll. In diesem Beispiel wird das Zertifikat ein Jahr lang gültig sein
   - Das **keyout**-Flag gibt den Pfad zu unserem generierten Schlüssel an
   - Das **out**-Flag gibt den Pfad zu unserem generierten Zertifikat an

   Achtung: Das Symbol „↲“ kennzeichnet nur einen Zeilenumbruch in dieser Anleitung. Es handelt sich demzufolge um einen Befehl, der in einer Zeile, ohne das Symbol „↲“, anzugeben ist!

   ```
   $ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 ↲
   -keyout /etc/apache2/ssl/apache.key -out ↲
   /etc/apache2/ssl/apache.crt
   ```

```
pi@8e78ea98e539:/etc/apache2$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a 2048 bit RSA private key
.....................................................+++
.....................+++
writing new private key to '/etc/apache2/ssl/apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Sachsen
Locality Name (eg, city) []:Mittweida
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Hochschule Mittweida
Organizational Unit Name (eg, section) []:Medien
Common Name (e.g. server FQDN or YOUR name) []:mtm-kt01.hs-mittweida.de
Email Address []:praktikum@hs-mittweida.de
```

6. Legen Sie die Dateiberechtigungen fest, um Ihren privaten Schlüssel und Ihr Zertifikat zu schützen!
   ```
   $ sudo chmod 6000 /etc/apache2/ssl/*
   ```

7. Als nächstes konfigurieren wir den Apache für die Verwendung des SSL-Schlüssels und -Zertifikats. Nach dieser Änderung beginnt unser Server, HTTPS- anstelle von HTTP-Anfragen für die Standardsite zu bedienen.
   Öffnen Sie hierfür die Datei default-ssl.conf!
   ```
   $ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
   ```

8. Suchen Sie den Abschnitt, der mit <VirtualHost _default_:443> beginnt, und nehmen Sie die folgenden Änderungen vor. Fügen Sie eine Zeile mit Ihrem Servernamen direkt unter der **ServerAdmin**-E-Mail-Zeile ein. Dies kann Ihr Domain-Name oder Ihre IP-Adresse sein:
   ```
   ServerAdmin webmaster@localhost
   ServerName mtm-kt01.hs-mittweida.de:443
   ```

9. Suchen Sie die folgenden beiden Zeilen und aktualisieren Sie die Pfade so, dass sie mit den Speicherorten des Zertifikats und des Schlüssels übereinstimmen, die wir zuvor erzeugt haben.
   ```
   SSLCertificateFile      /etc/apache2/ssl/apache.crt
   SSLCertificateKeyFile   /etc/apache2/ssl/apache.key
   ```

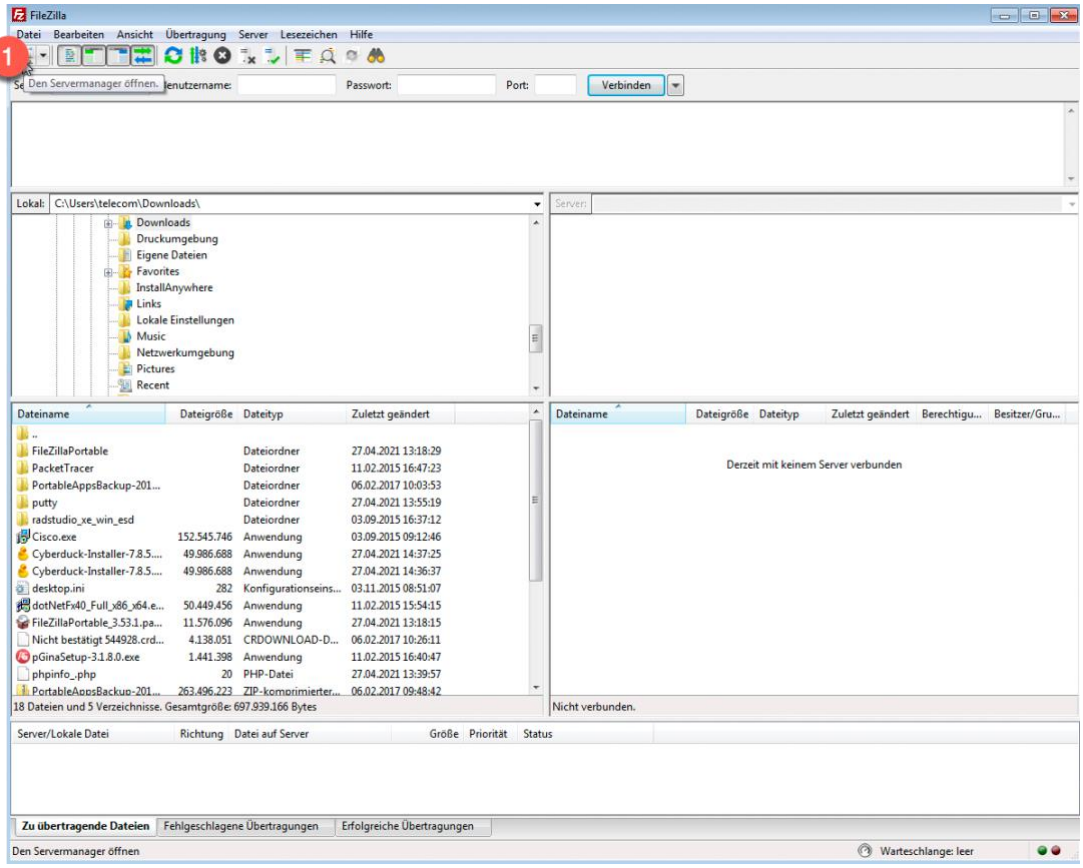
10. Speichern Sie das Konfigurationsfile mittels Strg+x und Starten Sie den Webserver neu!
    ```
    $ sudo /etc/init.d/apache2 restart
    ```

11. Testen Sie die Funktionalität der HTTPS-Verbindung durch Aufruf der URL https://<Docker-Container-FQDN:PORT>/phpinfo.php! Ersetzen Sie den Bezeichner <Docker-Container-FQDN:PÜORT> durch die FQDN und den HTTPS-Port Ihres Docker-Containers.
    Der Webserver zeigt Ihnen daraufhin die bereits besagte Warnung an. Bestätigen Sie daher die Warnung!
    z.B.

Filetransfer mittels SFTP

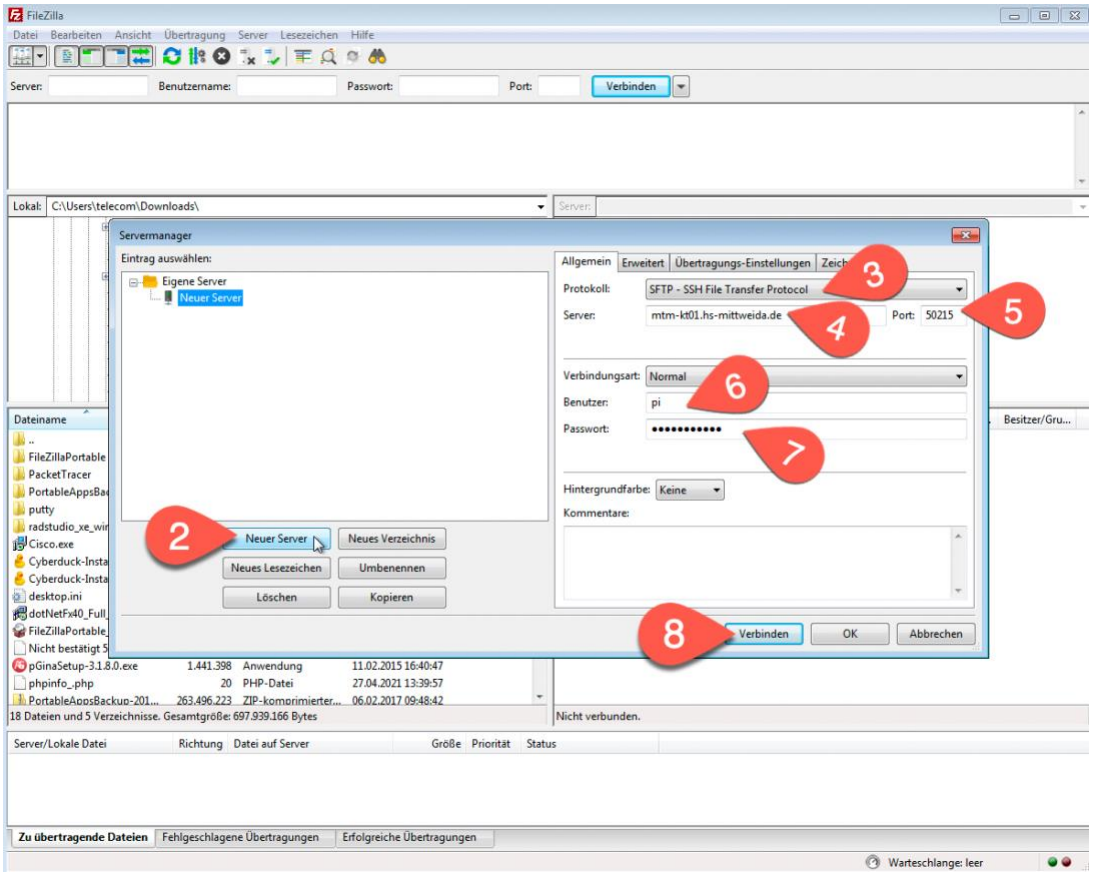
SFTP oder Secure File Transfer Protocol ist ein Protokoll, das auch als SSH File Transfer Protocol bezeichnet wird. Es ist ein Netzwerkprotokoll, um Dateien auf entfernte Systeme übertragen zu können. SFTP verwendet SSH (Secure Shell) zur Datenübertragung. Dabei muss sich der Client am Server authentifizieren. Kommandos und Daten werden verschlüsselt. Dadurch werden Passwörter und andere sensible Informationen nicht im Klartext über das Netzwerk gesendet und können nicht gelesen oder abgehört werden. SFTP wurde von der IETF (Internet Engineering Task Force) entwickelt, um die sichere Übertragung und Verwaltung von Dateien über TCP/IP-Netzwerke zu ermöglichen. SFTP verwendet die gleichen Befehle wie das herkömmliche File Transfer Protocol, das auch als FTP bekannt ist.

1. Laden Sie sich einen SFTP-Client herunter (z.B. FileZilla, CyberDuck) und installieren Sie die Software.
2. Starten Sie den SFTP-Client und richten Sie eine neue Verbindung ein (hier beispielhaft für FileZilla)

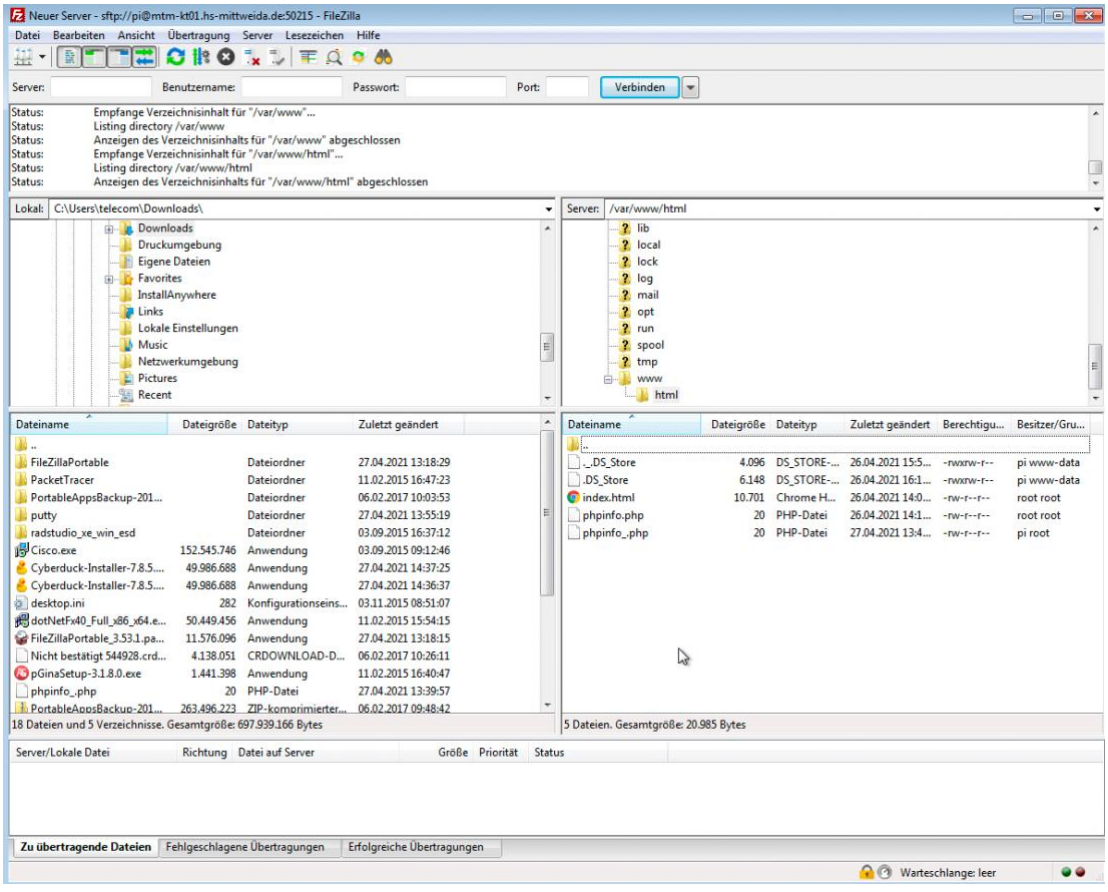


Verwenden Sie für das Server- und Port-Feld die Ihnen zugewiesenen Daten für Ihren Docker-

Container.



3. Nach erfolgreicher Verbindung können Sie Dateien zwischen Ihrem lokalen Host und dem Remote-Host austauschen.



Einrichten eines Samba-Servers

Mit CIFS (Common Internet File System) wird ein Netzwerkprotokoll für praktisch alle Betriebssysteme bezeichnet. Es wurde ursprünglich von Microsoft und IBM unter dem Namen SMB (Server Message Block) entwickelt. Darauf aufbauend erhielt das Open-Source-Projekt unter Linux den Namen Samba . SMB/CIFS ermöglicht den Dateitransfer zwischen Windows- und Unix-basierten Systemen. Unter Windows ist CIFS/SMB das Standardprotokoll für Netzwerkfreigaben, und Mac-OS X beherrscht das Protokoll ebenfalls.

1. Installieren Sie den Samba-Server!
   ```
   $ sudo apt-get install libcups2 samba samba-common cups
   ```
2. Archivieren Sie die aktuelle smb.conf-Konfigurationsdatei und erzeugen Sie eine Neue!
   ```
   $ sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
   $ sudo nano /etc/samba/smb.conf
   ```
3. Fügen Sie in smb.conf folgenden Inhalt ein!

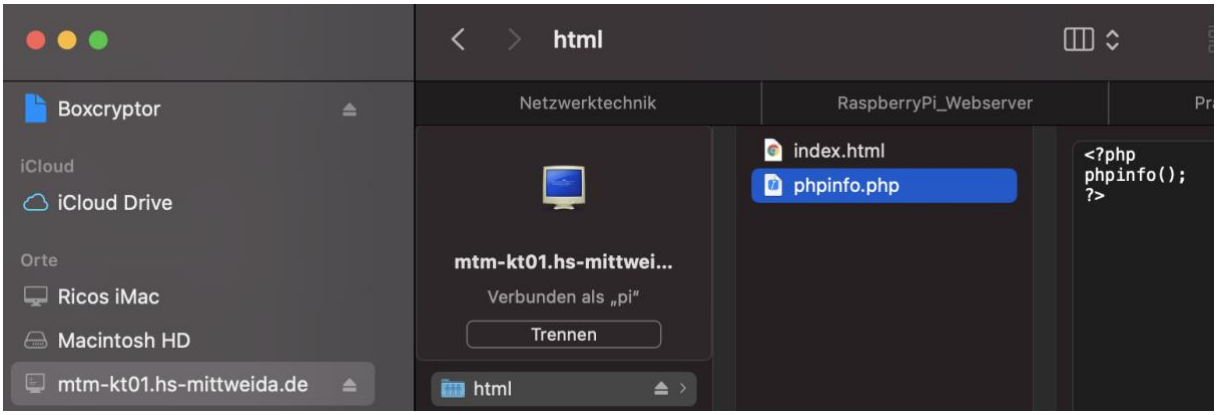
   ```
   [global]
   workgroup = WORKGROUP
   server string = Samba Server %v
   netbios name = Debian
   security = user
   map to guest = bad user
   dns proxy = no
   smb ports = 21

   [html]
   path = /var/www/html
   available = yes
   valid users = @www-data
   read only = no
   browsable = yes
   public = no
   writable = yes
   create mask = 0775
   force user = pi
   force group = www-data
   ```

4. Erzeugen Sie ein Samba-Passwort für den Nutzer pi! Verwenden Sie als Passwort wieder „praktikumkt“!
   ```
   $ sudo smbpasswd -a pi
   ```

5. Starten Sie den Samba-Dienst neu!
   ```
   $ sudo /etc/init.d/smbd restart
   ```

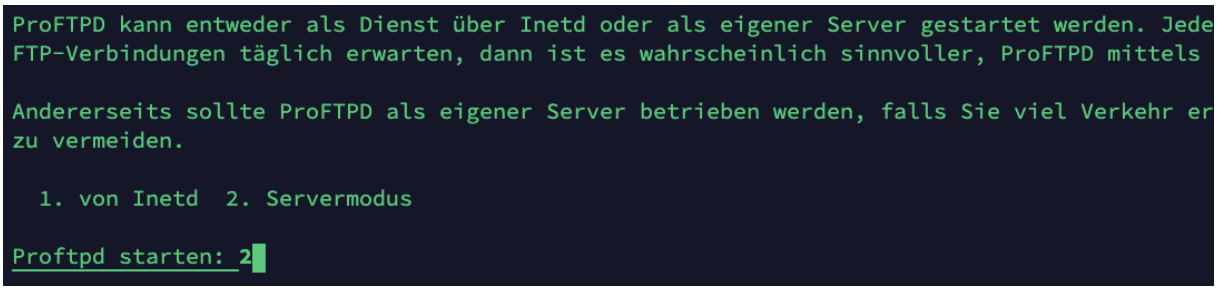
6. Verbinden Sie sich nun mit dem Samba-Server!
   Unter Windows können Sie das im Datei-Explorer mittels \\mtm-kt01.hs-

mittweida.de:<PORT> realisieren. Nutzen Sie als Nutzername „pi“ und als Passwort „praktikumkt“!
Unter Mac OS können Sie im Finder->Gehe Zu->Mit Server verbinden... aufrufen. Nutzen Sie hier smb://mtm-kt01.hs-mittweida.de:<PORT>!



Einrichten eines FTP-Servers

Um einen komfortablen Dateitransfer zwischen Webserver und PC zur realisieren, soll der FTP-Server „Proftpd“ installiert werden. Hierüber können dann beispielsweise PHP-Dateien am PC programmiert und anschließend auf den Raspberry Pi kopiert werden.

1. Installieren Sie Proftpd mittels folgenden Befehl!
   ```
   $ sudo apt-get install proftpd
   ```
2. Wählen Sie im erscheinenden Dialog „Servermodus“ und bestätigen Sie mit Enter!
   

3. Die Konfiguration des FTP-Servers erfolgt über die Datei `/etc/proftpd/proftpd.conf`. Hier müssen die freizugebenden Verzeichnisse und die Nutzerverwaltung definiert werden.
   Öffnen Sie daher die Datei `/etc/proftpd/proftpd.conf` mittels:
   ```
   $ sudo nano /etc/proftpd/proftpd.conf
   ```
4. In der geöffneten Datei muss folgender Inhalt vor der Zeile „# RequireValidShell off“ eingefügt werden:
   ```
   DefaultRoot ~
   AuthOrder       mod_auth_file.c mod_auth_unix.c
   AuthUserFile    /etc/proftpd/ftpd.passwd
   AuthPAM off
   RequireValidShell off
   ```
5. Speichern Sie die Datei mittels „Strg+x“ und bestätigen Sie die Abfrage mit „J“!
6. Legen Sie nun einen virtuellen Nutzer an, welcher für die Anmeldung am FTP-Server genutzt werden soll!
   Wechseln Sie hierfür in das Proftpd-Verzeichnis:
   ```
   $ cd /etc/proftpd
   ```
7. Damit die durch den virtuellen Nutzer erzeugten Dateien auch durch den Webserver aufgerufen werden können, soll dieser als „www-data“-Nutzer auftreten. Hierfür muss vorerst die uid des „www-data“-Nutzers ermittelt werden.
   ```
   $ id www-data
   ```
   →
   ```
   pi@raspberrypi /etc/proftpd $ id www-data
   uid=33(www-data) gid=33(www-data) Gruppen=33(www-data)
   ```
8. Jetzt soll der virtuelle Nutzer „praktikum“ mittels folgenden Befehls angelegt werden. Achten Sie dabei darauf, dass Sie den im vorherigen Befehl ausgegebenen uid- und gid-Wert nutzen. Im hier vorliegenden Beispiel entspricht dies dem Wert „33“. Geben Sie bei der Aufforderung zur Passworteingabe das Kennwort „praktikumkt“ an!
   Achtung: Das Symbol „ ↵“ kennzeichnet nur einen Zeilenumbruch in dieser Anleitung. Es

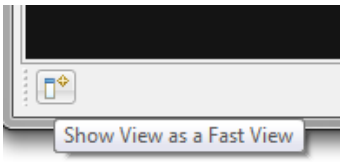
handelt sich demzufolge um einen Befehl, der in einer Zeile, ohne das Symbol „↵“, anzugeben ist!

```
$ sudo ftpasswd --passwd --name praktikum --uid 33 --gid 33↵
 --home /var/www/html  --shell /bin/false
```

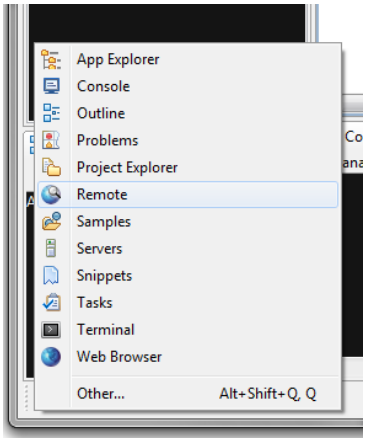
9. Starten Sie nun den FTP-Server neu!

```
$ sudo /etc/init.d/proftpd restart
```

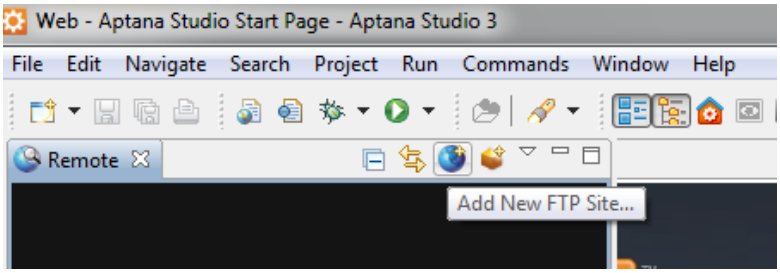
10. Öffnen Sie die Software „Aptana“ auf Ihrem Rechner!
    Drücken Sie den links unten befindlichen Button.



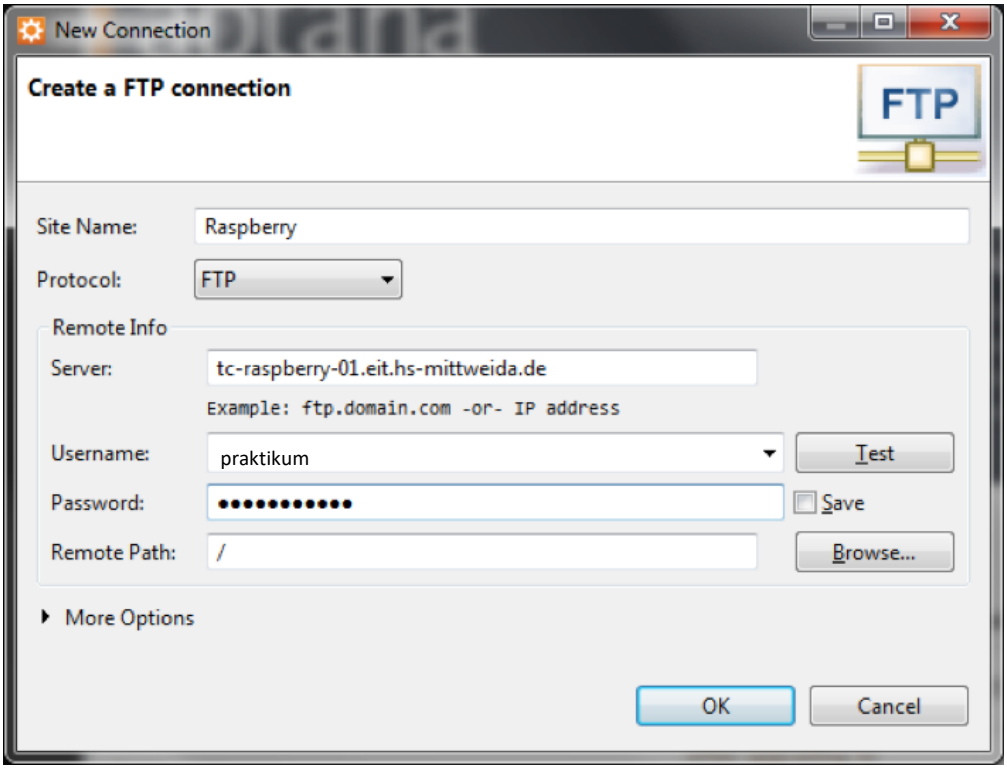
Wählen Sie „Remote“ aus!



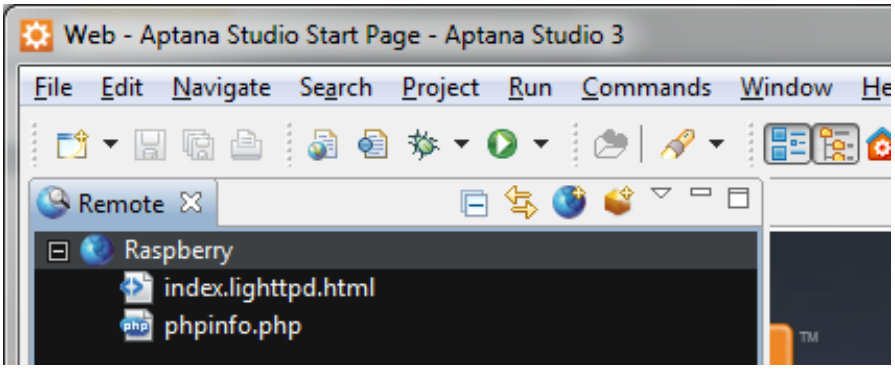
Wählen Sie „Add New FTP Site...“!



Tragen Sie im folgenden Dialog Ihre Verbindungsdaten ein:
Site Name: *Raspberry*
Server: *„aufgedruckte Adresse Ihres Raspberry Pi's“*
Username: *praktikum*
Password: *praktikumkt*
Wählen Sie anschließend „OK“



Sie haben nun Zugriff auf das Wurzelverzeichnis des Raspberry-Webservers.



Schalten einer Funksteckdose mittels Raspberry-Pi

Über die GPIO-Ausgänge des Raspberry Pi besteht die Möglichkeit verschiedenste Geräte anzuschalten. Für diesen Versuch wurde am Raspberry Pi ein 433MHz-Sender angeschlossen. Über diesen Sender besteht die Möglichkeit handelsübliche Funktsteckdosen ein- bzw. auszuschalten. Um dies zu realisieren, ist eine Software notwendig, welche wie folgt zu installieren ist.

1. Führen Sie eine Aktualisierung des Betriebssystems durch! Rufen Sie hierfür folgende Befehle auf (kann ca. 10-15min dauern):
   ```
   $ sudo apt-get update && sudo apt-get upgrade -y
   ```

2. Die Software zum Schalten von 433MHz-Steckdosen benötigt die wiringPi-Bibliotheken. Diese sind im Docker-Container bereits installiert. Bei der Einrichtung der Software auf einem

realen Raspberry Pi müssen diese eventuell zuvor noch installiert werden

3. Die Software zum Schalten der 433MHz-Steckdosen lautet 433Utils. Für das Praktikum wird nur das Programm zum Senden von Befehlen benötigt. Diese Software ist auf dem realen Raspberry Pi installiert, welchen Sie im Zoom-Meeting sehen können. Für dessen Ansteuerung können Sie das send-Programm wie folgt in Ihrem Docker-Container installieren:

```
$ cd ~
$ wget https://www.staff.hs-↲
mittweida.de/~rthomane/praktikumkt/raspberry/remoteDocker/↲
DebianJessie/send
$ chmod +x send
$ sudo cp send /usr/bin
$ rm /home/pi/send
```

4. Versuchen Sie nun eine Funktsteckdose mit dem Rasberry Pi ein- bzw. auszuschalten! Hierfür wird das Programm „send“ verwendet, welches auf dem Remote-Raspberry im Verzeichnis „/home/pi/wiringPi/433Utils/RPi_utils/“ zu finden ist. Das Programm „send“ erfordert hierbei folgende Parameter:
   - die GPIO (**G**eneral **P**urpose **I**nput/**O**utput) Nummer, wo der 433MHz-Sender angeschlossen ist:
   → *Nutzen Sie hierfür den Wert **15***
   - der Systemcode, der in der Steckdose eingestellt ist:
   → *Nutzen Sie hierfür die Nummer Ihres Raspberry Pi in binärer Darstellung (5 Bit), z.B.*
   → *z.B. tc-raspberry-**07**.eit.hs-mittweida → **7**d = **00111***
   - der Unitcode, der die Funktsteckdose spezifiziert:
   → *Nutzen Sie hierfür den Wert **1***

   Für den Raspberry Pi mit der Nummer 07 sieht der Programmaufruf dann beispielsweise wie folgt aus:
   Einschalten:

```
$ send -p 15 -s 00111 -u 1 -c 1
```

   Auschalten:

```
$ send -p 15 -s 00111 -u 1 -c 0
```

Funksteckdose mittels HTTP ein- bzw. ausschalten

Das Ein- bzw. Ausschalten der Funktsteckdose funktioniert bis jetzt nur direkt vom Raspberry Pi aus. Wünschenswert wäre allerdings auch eine Ansteuerung von jedem , am Netzwerk angeschalteten, Gerät. Um dies zu ermöglichen, muss der Raspberry Pi Steuerbefehle über das Netzwerk empfangen und daraufhin die Funksteckdose schalten. Um möglichst viele verschiedene Geräte zu unterstützen, sollte man zur Kommunikation zwischen Raspberry Pi und Steuergerät ein standardisiertes Protokoll verwenden, welches auf einer Server – Client – Architektur basiert. Ein weltweit standardisiertes Protokoll stellt dabei das HTTP (Hypertext Transfer Protocol) dar. Der Raspberry Pi müsste dabei die Rolle des Servers übernehmen, was durch den bereits installierten Webserver gegeben ist. Der auf einem entfernten Rechner installierte Webbrowser würde dann die Rolle des Clients übernehmen. Das schalten der Funktsteckdose kann dann ganz bequem über den Aufruf einer URL erfolgen. Um den Webserver zum Ausführen des „send"-Programms zu bewegen, kann dieser Befehl in einer PHP-Datei untergebracht werden.

1. Öffnen Sie einen Text-Editor!
2. Schreiben Sie folgendes in den Text-Editor:

3. Öffnen Sie die vorhin eingerichtete SFTP-Verbindung!

```
<?php
$gpio=$_GET['gpio'];
$systemcode = $_GET['systemcode'];
$unitcode = $_GET['unitcode'];
$onoff = $_GET['onoff'];
echo @shell_exec("/home/pi/dist/send -p $gpio -s $systemcode -u
$unitcode -c $onoff");
?>
```

4. Speichern Sie die Datei als „switchOnOff.php" lokal auf Ihrem Rechner!
5. Öffnen Sie einen Webbrowser auf Ihren PC und geben Sie die URL zur switchOnOff.php-Datei an! Achten Sie darauf, den für Ihre Steckdose eingestellten Systemcode und Unitcode zu verwenden.
   z.B.
   https://<DockerContainerFQDN>:<PORT>/switchOnOff.php?gpio=15&systemcode=00111&unitcode=1&onoff=0

Sie sollten nun in der Lage sein, die Funksteckdose mittels des Webbrosers ein- bzw. auszuschalten. Hierfür müssen Sie für den Parameter „onoff" den Wert „0" oder „1" angeben!

Das Ergebnis sollten Sie über Zoom sehen können.

Temperatur auslesen

Am Ihrem Raspberry Pi ist weiterhin ein Temperatur-Sensor angeschlossen. Die Sensoren werden im Betriebssystem des Raspberry Pi wie eine Datei behandelt. Der Temperatur-Sensor ist hierbei im Verzeichnis „/sys/bus/w1/devices/“ zu finden. Da an dem Docker-Container natürlich kein Temperatur-Sensor angeschlossen ist, können Sie über das Verzeichnis „/rsys/bus/w1/devices“ remote auf einen realen Temperatur-Sensor zugreifen. Für jeden am Raspberry Pi angeschlossenen 1-wire-Sensor, existiert unter „/sys/bus/w1/devices/“ ein eigenes Verzeichnis. Dieses Verzeichnis entspricht der digitalen Kennung des Sensors, z.B. 28-000004f19477. Innerhalb des Sensor-Verzeichnisses existiert dann eine Datei „w1_slave“, welche die aktuelle Temperatur beinhaltet.

1. Finden Sie die digitale Kennung Ihres Remote-Temperatur-Sensors heraus! Rufen Sie hierfür folgenden Befehl auf!

   ```
   $ ls -l /rsys/bus/w1/devices/
   ```

   Als Ergebnis erscheint:

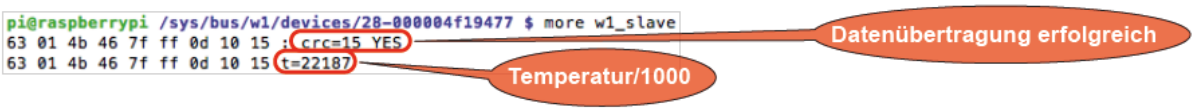
   ```
   insgesamt 0
   lrwxrwxrwx 1 root root 0 Nov 13 14:03 28-000004f19477 -> ../../../devices/w1_bus_master1/28-000004f19477
   lrwxrwxrwx 1 root root 0 Nov 13 13:48 w1_bus_master1 -> ../../../devices/w1_bus_master1
   ```

   Sie sehen als Ergebnis eine Auflistung aller Verzeichnisse innerhalb von „/rsys/bus/w1/devices“. Der Verzeichnisname des Temperatur-Sensors beginnt dabei immer mit einer Ziffer. In vorliegenden Beispiel ist der angeschlossene Temperatur-Sensor demzufolge das Verzeichnis 28-000004f19477.

2. Zeigen Sie sich die aktuelle Temperatur an! Rufen Sie hierfür die Datei „w1_slave“ im Verzeichnis des Remote-Temperatur-Sensors auf! Achten Sie darauf, den Verzeichnisname Ihres Temperatur-Sensors zu verwenden!

   ```
   $ /rsys/bus/w1/devices/28-000004f19477/w1_slave
   ```

   Als Ergebnis erscheint:

   Wie in der Grafik ersichtlich, wird die Temperatur mit t=22187 angezeigt. Dieser Wert muss noch durch 1000 geteilt werden. Daher würde sich im vorliegenden Beispiel die Temperatur 22.187°C ergeben.

3. Schreiben Sie sich den Verzeichnisname Ihres Temperatur-Sensors auf!

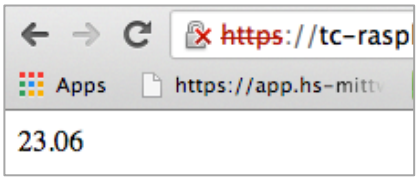
Temperatur per HTTPS zur Verfügung stellen

Ziel soll es sein, die Raumtemperatur auch über den Webserver anzubieten.

1. Erzeugen Sie daher in Aptana eine neue PHP-Datei „getTemperature.php“!
2. Fügen Sie folgenden Inhalt in die Datei „getTemperature.php“ ein! Achten Sie darauf den Verzeichnisname Ihres Temperatur-Sensors zu nutzen!

```
<?php
   //Sensor-File öffnen
    $tempText = @shell_exec("/rsys/bus/w1/devices/28-
000004c01606/w1_slave");

   //File auslesen

   //Text anhand des Textes "t=" in ein Array wandeln
   $tempArray = split("t=", $tempText);
   //Temperatur-Wert durch 1000 teilen und runden
   $temperatur = round(floatval($tempArray[1])/1000,2);
   //Temperatur ausgeben
   echo $temperatur;
?>
```

3. Speichern Sie die Datei!
4. Öffnen Sie einen Webbrowser und rufen Sie die Datei „getTemperature.php“ mittels HTTPS-Request auf!



Als Ergebnis wird die aktuelle Temperatur angezeigt.