Tutorial-Skript: Einrichtung der Umgebung auf einem Raspberry Pi

Von Manuel Yates

Guten Tag,
Ich werde ihnen heute zeigen, wie wir unseren Raspberry Pi für unsere Wetterumgebung vorbereiten. In diesem Tutorial werden wir folgende Stationen durchlaufen.

1. Vorbereitung
2. Installation des Betriebssystems
3. Installation der benötigten Software
4. Anpassung von Systemdateien

Als Vorwort sind hier noch einige Dinge zu beachten:
In diesem Tutorial werden wir folgende Dienste installieren und konfigurieren. Hierbei werden wir die Sicherheitsthematik bewusst begrenzen, da diese den Rahmen dieses Tutorials sprengen würde. Wir werden auch bestimmte Befehle nur im Groben erklären, Links zu ausführlicheren Erklärungen finden Sie allerdings in der Videobeschreibung oder in den Repository-Informationen auf Github.

Schritt 1. Vorbereitung

Um die Wetterstation für unsere Zwecke einzurichten, verwenden wir:

Hardwareseitig:

- Einen Raspberry Pi
- Eine MicroSD-Karte mit min. 8GB Speicher
- Einen Computer mit einem der gängigen Betriebssysteme
- Ein Octoboard
- Einen Adapter zum Anschließen der MicroSD Karte

Softwareseitig:

- Raspberry Pi Imager
- PuTTY
- MQTT.fx

In diesem Tutorial verwenden wir einen Raspberry Pi 4 mit 2GB Arbeitsspeicher. Dieser reicht für die unsere Basis-Einrichtung vollkommen aus und kann bei Bedarf problemlos durch einen leistungsstärkeren Pi ersetzt werden. Bei der MicroSD-Karte sollte bei dem Aufbau eines realen „Produktionssystem“ auf eine ausreichende Klassifizierung und Speichergröße geachtet werden. Grundsätzlich sollte aber für einen Testaufbau jede normale Karte ausreichen. In diesem Tutorial werden wir das Betriebssystem Windows 10 verwenden. Die Arbeitsschritte können allerdings mit ein wenig Recherche auch problemlos auf anderen Betriebssystemen vollzogen werden. Rechenleistung spielt hierbei übrigens keinerlei Rolle, da, bis auf die Installation des Betriebssystems auf der SD Karte, alle Arbeitsschritte auf dem Raspberry Pi stattfinden.

Schritt 2. Installation des Betriebssystems

Um das Betriebssystem auf die SD-Karte zu flashen, verwenden wir den von Raspberry angebotenen Imager. Dieser ist unter www.raspberrypi.com/software/ zu finden. Nach der Installation des Imagers kann an der Oberfläche des Programmes das Betriebssystem ausgewählt werden. In diesem Tutorial

verwenden wir „Raspberry Pi OS Lite 64 Bit“. Der Installationsprozess kann einige Zeit in Anspruch nehmen.

Nachdem die Installation und Verifizierung abgeschlossen ist, muss die MicroSD Karte einmal entfernt und wieder angeschlossen werden. Nun zeigt Windows ein Datenträger mit der Bezeichnung „boot“ im Explorer an. Um im weiteren Verlauf über eine SSH Verbindung uns auf den PI zu verbinden, müssen wir hierfür eine Datei mit der Bezeichnung „ssh“ anlegen. Hierbei ist zu beachten, dass es sich um eine reine Datei handelt, somit muss die Datei ohne Dateiendung angelegt werden.

Nun ist noch wichtig, dass wir unserem Raspberry Pi in unserem Router eine statische IP Adresse hinterlegen. Dies ist je nach Router und Netzwerkinfrastruktur unterschiedlich aber im allgemeinen gelten hier folgende Arbeitsschritte:

1. Den Raspberry Pi mit Strom versorgen und mit dem Router verbinden
2. Kontrolloberfläche des Routers aufrufen (z.B. 192.168.0.1, fritz.box)
3. In den Einstellungen des DHCP Dienstes die MacAdresse mit einer statischen IP Adresse hinterlegen.
4. Den Pi nun einmal vom Strom trennen und nach kurzer Zeit wieder mit Strom versorgen.

Nun sollte in der Geräteübersicht in der Kontrolloberfläche des Routers der Raspberry Pi mit der korrekten statischen IP Adresse angezeigt werden.

Damit ist die Installation und Vorkonfiguration abgeschlossen und wir können mit Installation und Einrichtung unserer Umgebung loslegen.

Schritt 3: Installation der benötigten Software.

Nachdem der Pi hochgefahren ist (Ein generelles Zeichen ist das Blinken der Netzwerk-Dose) können wir nun über das Tool Putty uns mit dem Pi verbinden. Hierzu geben wir einfach in der Oberfläche die IP-Adresse unseres Pi’s ein und drücken „Connect“. Nun folgt ein Warnhinweis, welcher abfragt ob wir dem Gerät vertrauen möchten. Dies bestätigen wir mit „Always“.

Nun sehen wir ein Konsolenfenster in welchem wir nach unseren Anmeldedaten gefragt werden. Standardmäßig wird hier der Username „pi“ mit dem Passwort „raspberry“ verwendet.

Als erstes sollte bei jeder neuen Installation des Betriebsystems folgende Befehle ausgeführt werden:

- ***sudo apt update***
- ***sudo apt upgrade***
- ***sudo raspi-config***

Mit Letztem rufen wir das Konfigurationsmenü unseres Raspberry Pi’s auf. Hier können sowohl die Anmeldedaten als auch Lokalisationseinstellungen verändert werden.

Als kurzes Zwischenthema bietet sich hier nun an über einige grundlegende Befehle zu sprechen:

- ***sudo systemctl status*** *Programmname*
  Mit diesem Befehl können Wir feststellen, ob das Programm läuft.
- ***sudo systemctl start*** *Programmname*
  Mit diesem Befehl können Wir ein Programm starten.
- ***sudo systemctl stop*** *Programmname*
- Mit diesem Befehl können Wir ein Programm stoppen.
- ***sudo systemctl restart*** *Programmname*
- ***sudo systemctl enable*** *Programmname*

Mit diesem Befehl können Wir das automatische Starten des Programmes beim Hochfahren einschalten.

- ***sudo systemctl disable*** *Programmname*
  Mit diesem Befehl können Wir das automatische Starten des Programmes beim Hochfahren ausschalten.
- ***sudo*** *reboot*
  Mit diesem Befehl können Wir den gesamten Raspberry Pi neu starten.

Nun beginnen wir mit der Installation unseres MQTT Brokers. Hierfür verwenden wir die Software Mosquitto von Eclipse. Der Installationsbefehl lautet:

- sudo apt install mosquitto

Mögliche auftretende Abfragen bestätigen wir mit einem „Y“.

Nun verwenden wir, den bereits im Betriebssystem installierten Editor Nano, um eine Konfigurationsdatei von Mosquitto zu verändern. Der Befehl hierfür lautet:

- sudo nano /etc/mosquitto/mosquitto.conf

In dieser Datei fügen wir nun zwei Befehle hinzu:

- listener 1883
- allow_anonymous true

Ersterer besagt lediglich, dass Mosquitto auf dem Port 1883 „hören“ soll, also über diesen Port Daten empfangen kann. Der Port 1883 ist der standartmäßige MQTT Port.

Der zweite Befehl besagt hier lediglich, dass bei der Datenübertragung keine Sicherheitsabfrage gemacht wird, sondern alle Clients über diesen Port Daten senden können. Hier sei erneut darauf hingewiesen, dass in diesem Tutorial kein großer Fokus auf die Sicherheitsaspekte gelegt wird. Dies können wir bei Interesse aber noch nachreichen.

Nun setzen wir mit der Installation von NodeRed fort. NodeRed ist ein umfangreiches Low-Code-Tool mit welchen Programmabfolgen über eine einfache Benutzeroberfläche erstellt werden können. In unserer Konfiguration stellt dies eine Schnittstelle zwischen MQTT und unserer MySQL Datenbank dar. Installiert wird dies über den Befehl:

- bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

Nachdem die Installation vollzogen ist können wir direkt mit der Installation von MariaDB und MySql weiter machen. Die Installation erfolgt über den Befehl:

- sudo apt install mariadb-server

Nach der Installation können wir über den Befehl

- sudo mysql -u root -p

auf die Datenbank-Konsole zugreifen. Wichtig ist hierbei, dass die Passwortabfrage einfach mit einem Enter übersprungen wird, da der User Root nach der Installation noch über kein Passwort verfügt. Nun können wir über die SQL Konsole bereits einige Dinge konfigurieren.

Als Erstes erstellen wir eine Datenbank. In unserem Fall trägt diese den Namen „BBS_Weatherstation“.

- CREATE DATABASE BBS_Weatherstation;

Anschließend legen wir drei Benutzerkonten für diese Datenbank an:

- Der User „Admin“ wird von Uns als Administrator verwendet um die Datenbank später verwalten zu können.
  CREATE USER 'Admin'@'%' IDENTIFIED BY 'root';
- Der User „NodeRed“ ist der dedizierte Account unseres NodeRed Dienstes, welcher uns die MQTT-Daten in die Datenbank schreibt.
  CREATE USER 'NodeRed'@'%' IDENTIFIED BY 'nodered';
- Der User „Webserver“ wird von unserem Blazor Webserver verwendet um auf die Datenbank zuzugreifen.
  CREATE USER 'WebServer'@'%' IDENTIFIED BY 'Wittlich';

Mit dieser Methode können im Nachhinein auch klare Restriktionen der einzelnen Benutzer festgelegt werden. Um die verwendeten Befehle kurz zu erläutern. Der Befehl CREATE USER sollte wohl selbsterklärend sein. Folgend benennen wir den Benutzer, darauf folgend ein „@“ Zeichen und in der darauffolgenden Stelle, welche hier mit einem % gefüllt wird, können wir genau definieren von welchem Netzwerkclient der Zugriff stattfinden darf. Das % ist hier eine Wildcard für jeden Netzwerkclient, hier könnte aber auch wahlweise „localhost“ oder „192.168.0.123“ stehen. So könnten z.B.: auch mehrere Raspberry Pi’s miteinander verbunden werden um jedem Pi eine bestimmte Aufgabe zuzuweisen. Zuletzt folgt noch ein IDENTIFIED BY und im darauffolgenden Platz wird nun das Passwort des jeweiligen Accounts definiert.

Nun müssen diesen Accounts noch Berechtigungen freigegeben werden und da wir uns nicht mit dem Sicherheitsaspekt auseinandersetzen, werden wir diesen Benutzern sämtliche Rechte auf unserer Datenbank einräumen. Dies erfolgt über:

- GRANT ALL PRIVILEGES ON BBS_Weatherstation.* to 'Benutzername'@'%' IDENTIFIED BY 'Passwort';

Um diese Berechtigungen nun direkt zu aktivieren, nutzen wir den Befehl

- FLUSH PRIVILEGES;

Zuletzt passen wir noch eine Konfigurationsdatei der MariaDB an.

- sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

In dieser kommentieren wir das Feld „bind-adress“ ein und weisen diesem statt „127.0.0.1“ die IP Adresse „0.0.0.0“ zu. So können wir gleich auch von außerhalb, also von einem anderen Netzwerkgerät, auf die Datenbank zugreifen.