

Handbuch zur Inbetriebnahme des Backends und des Arduinos

Arduino

Der Arduino verwendet noch eine veraltete Softwareversion und sollte daher vor der ersten Inbetriebnahme aktualisiert werden.

Zum Upload eines neuen Programms auf den Arduino ist die Entwicklungsumgebung [Visual Studio Code](#) mit der Erweiterung [PlatformIO](#) notwendig.

Anschließend kann der PlatformIO-Projektordner `01_Arduino_MQTT/Arduino` geöffnet werden. Alle benötigten Bibliotheken sind in der Datei `platformio.ini` konfiguriert und werden daher automatisch installiert.

In der unteren linken Ecke des Visual Studio Code-Fensters sollten jetzt mehrere [PlatformIO-Funktionen](#) angezeigt werden. Hier muss die Upload-Funktion (Pfeil nach rechts) gewählt werden. Dadurch wird die Arduino-Software kompiliert und auf den Arduino geladen.

Die Bestellung läuft automatisch ab, nachdem ein Glas auf die Startposition gestellt wurde. Nur bei einer Fehlermeldung muss die Bestellung mit dem "Fortsetzen"-Taster (siehe Schaltplan) fortgesetzt werden.

Python-Bridge

Für die Python-Bridge steht ein Docker-Container zur Verfügung, d. h. es müssen weder Python noch benötigte Bibliotheken installiert werden. Einzige Voraussetzung ist die Installation von [Docker](#).

Da die Python-Bridge über die serielle Schnittstelle mit dem Arduino kommuniziert, muss der Rechner, auf dem die Python-Bridge ausgeführt wird, per USB mit dem Arduino verbunden sein.

Damit die Bridge auf die serielle Schnittstelle zugreifen kann, sind je nach Betriebssystem noch einige Einstellungen nötig:

- Linux: In der Datei `04_Docker\Python_Bridge\docker-compose.yml` wird in dem Abschnitt `devices` der serielle Port für den Container freigegeben. Je nach Rechner kann es allerdings sein, dass der Port einen anderen Namen hat. Ist der Arduino beispielsweise unter dem Port `/dev/ttyUSB1` erreichbar, muss die Zeile `/dev/ttyUSB0:/dev/ttyUSB0` zu `/dev/ttyUSB1:/dev/ttyUSB0` geändert werden.
- Windows: Hier ist die Einrichtung komplizierter, da der Container in einer virtuellen Maschine ausgeführt wird und nur über einen Umweg auf den seriellen Port zugreifen kann. Zuerst muss das Tool `usbipd` anhand der [verlinkten Anleitung](#) installiert werden. Anschließend muss man unter Windows ein Administrator-Terminal öffnen. Mit `usbipd wsl list` kann man sich eine Liste der verfügbaren USB-Geräte anzeigen lassen. Hier muss man die Bus-ID der Zeile bestimmen, die zum Arduino gehört. Mit `usbipd wsl attach --busid <Bus-ID>` kann man dann den seriellen Port für den Container freigegeben.

Gestartet werden kann der Docker-Container mit dem Befehl `docker compose -f "04_Docker\Python_Bridge\docker-compose.yml" up --build`. Hat man die Docker-Erweiterung für

Visual Studio Code installiert, kann der Container auch nach einem Rechtsklick auf die Docker Compose-Datei über die Option "Compose Up" gestartet werden.

Backend

Das Backend, die dazugehörigene Datenbank und das Tool "phpMyAdmin" werden ebenfalls innerhalb von Docker-Containern ausgeführt, daher muss auch hier nur Docker installiert sein. Außerdem muss das Backend nicht zwingend auf dem Rechner laufen, auf dem die Python-Bridge ausgeführt wird.

Mit dem Befehl `docker compose -f "04_Docker\docker-compose.yml" up --build` werden alle drei Container gestartet. Anschließend ist das Backend einsatzbereit.

Falls notwendig, kann man jetzt unter der Adresse <http://localhost:8080/> auf die phpMyAdmin-Oberfläche zugreifen.