

Steuerung Füllanlage 200 Bar Druckluft des Sporttauchverein Hückelhoven e.V.

Tom Schönijahn

Juli 2023

1 Einführung

Das vorliegende Projekt wurde im Rahmen des Kurses Mikrocontrollertechnik im Sommersemester 2023 des Studiengangs Angewandte Mathematik und Informatik der FH Aachen erarbeitet. Die Aufgabenstellung stammte dabei aus dem Tauchverein Hückelhoven e.V. nach Neubeschaffung einer Füllanlage für Tauchflaschen.

Aus Gründen der Betriebssicherheit darf die Füllanlage nur durch nachweislich unterwiesene Personen betrieben werden. Zur Einhaltung dieser Vorschrift wurde entschieden das jedes unterwiesene Vereinsmitglied eine Möglichkeit erhalten solle die Zuleitung der Anlage einschalten zu können, dabei solle es jedoch möglich sein die Nutzung der Anlage zu überwachen und den Zugang für einzelne Personen nachträglich auch sperren zu können. Aufgrund der baulichen Begebenheiten ist ein Zugriffsschutz auf die Anlage nicht möglich. In der konkreten Planung wurde sich für den Einbau eines NC-Magnetventils mit 24V Eingang in die Versorgungsleitung entschieden, welches dann über die neu entwickelte Steuerung aktiviert wird. Die Authentifizierung der Benutzer solle möglichst einfach erfolgen, sodass sich für die Verwendung von RFID-Chips entschieden wurde. Ein weiterer Vorteil dieser ist, dass kein direkter Kontakt notwendig ist, sodass die gesamte Anlage staubdicht installiert werden kann. Zur Verwaltung der Zugriffe soll eine übersichtliche Website verwendet werden in der auch Auswertungen erzeugt werden können. Aufgrund der vorhanden WLAN-Abdeckung soll der Controller direkt mit einer Online-Datenquelle kommunizieren können. Aus Gründen des Komforts für den Benutzer ist ebenfalls eine Textanzeige erwünscht, um nicht auf farbliche Statuscodes angewiesen zu sein.

2 Steuerung

Die Entwicklung der Steuerung gliederte sich in zwei wesentliche Aspekte, zum einen die Auswahl und Installation der Hardware inklusive benötigter Stromversorgung und zum anderen die Entwicklung der Software für den Mikrocontroller. Aufgrund der möglichen Besonderheiten der Hardware konnte mit der Softwareentwicklung erst nach Aufbau der Komponenten begonnen werden.

2.1 Hardware

Die Elektronik ist auf zwei Gehäuse in Form von Abzweigdosen mit Kabeldurchführungen aufgeteilt. Das erste Gehäuse zur Spannungsversorgung enthält die Spannungswandler für die Umwandlung der $230VAC$ auf $24VDC$ und $5VDC$ sowie das Relais, zur Schaltung der Ausgangsleitung. Somit befindet sich dort der gesamte Niederspannungsanteil des Aufbaus. Benötigt der

Ausgang keine 24V Schaltspannung kann das erste Netzteil durch eins für die entsprechende Spannung ersetzt oder entfernt werden, da die gesamte Elektronik ausschließlich eine 5V Versorgung benötigt.

In dem zweiten Gehäuse zur Steuerung befinden sich jene Komponenten, die zur Interaktion und Authentifizierung notwendig sind. In diesem ist nur Kleinspannung mit höchstens 5V vorhanden. Zur Verbindung untereinander wird ein dreiadriges Kabel verwendet. Der Anschluss der Spannungsversorgungsbox wird durch ein zweiadriges Kabel sichergestellt und auf der Ausgangsseite befindet sich dasselbe Kabel. Bei Verwendung einer externen Spannungsversorgung für den anzusteuernenden Aktor muss das Kabel auf der Ausgangsseite entsprechend angepasst werden.

2.1.1 Mikrocontroller

Die Elektronik verwendet einen Raspberry Pi Pico W der ersten Generation als Mikrocontroller. Das WLAN-Modul wird zur Herstellung einer Netzwerk-Verbindung für die Kommunikation mit einer API verwendet. An das Board sind mithilfe eines Steckbrett die weiteren Komponenten direkt angeschlossen und die Spannungsversorgung wird über die 5V Verbindung aus der Versorgungsbox geleistet.

2.1.2 RFID

Authentifizierungen an der Füllanlage werden mittels eines RFID-Tags auf 13,56 MHz durchgeführt. Ausgelesen werden können diese mit dem integrierten RFID-RC522. Die Kommunikation zwischen RFID-Reader und Mikrocontroller geschieht über eine SPI-Verbindung mit 3,3V. Des Weiteren ist eine Verbindung zu der 3,3V und *GND* Versorgung vorhanden. Als RFID-Tag können alle verfügbaren Tags mit derselben Frequenz verwendet werden, es wird die Verwendung von Karten oder Schlüsselanhängern empfohlen.

2.1.3 Visualisierung

Zur Visualisierung der Status und der Kommunikation mit dem Benutzer befinden sich eine RGB-LED und ein LCD mit je 20 Zeichen in 4 Zeilen von außen sichtbar in dem Gehäuse. Die RGB-LED ist mit 3 verschiedenen PWM-Pins des Controllers verbunden und hat zusätzlich eine *GND* Verbindung. Das Display wurde um eine HD44780 kompatible I2C-Schnittstelle erweitert und ist somit mit der Spannungsversorgung 3,3V, *GND* und den Datenleitungen *SDA* und *SCL* für die Kommunikation verbunden. Mithilfe des Potentiometers auf dem I2C-Adapter kann der Kontrast des Displays verändert werden. Ebenfalls ist eine Deaktivierung der LCD-Beleuchtung mittels des Jumpers oder über die I2C-Schnittstelle möglich.

2.1.4 Ventilsteuerung

Die Ansteuerung des Magnetventils oder möglicherweise auch anderer Aktoren geschieht über zwei einpolige Relais, welche zu einem zweipoligen Relais zusammengeschaltet wurden. Die Relais können bei einem Strom von bis zu 10A eine Wechselspannung von bis zu 250V oder eine Gleichspannung von bis zu 30V schalten. Bei Aktoren die eine 24V Gleichspannung mit bis zu 1A benötigen, kann die Spannungsversorgung direkt von dem integrierten 24V Wandler abgegriffen und auf die Eingangsklemmen der beiden Relais geklemmt werden. Bei Nutzung einer externen Spannungsversorgung entfällt diese Verbindung und das Relais wird in Reihe geschaltet. Im Allgemeinen ist die Verwendung von NO-Ausgängen zu bevorzugen, um bei einem Spannungsverlust oder Systemausfall in demselben Grundzustand zu sein. Dies führt zur Sicherheit, dass die Anlage nicht unerwartet aktiv sein kann.

2.2 Software

Der Raspberry Pi Pico W ist mit der offiziell veröffentlichten Micropython Firmware geflasht. Somit ist auch der gesamte Code in Python mit den entsprechenden Micropython Bibliotheken geschrieben. Es wurden keine Bibliotheken zusätzlich installiert, jedoch zum Teil in den eigenen Codebestand hinzugefügt, um den Upload zu vereinfachen. Die Quelle zu den Projekten ist in der jeweiligen Datei angegeben. Die neuste getestete Micropython-Version ist im Code-Repository des Controllers angegeben. Alle Konfigurationsmöglichkeiten befinden sich in der *config.py*. Bei dieser muss es sich um eine valide Python-Datei handeln, da ihre Klassen und Variablen direkt im Code eingebunden sind.

2.2.1 WiFi & API

Die Verbindung mit dem konfigurierten WLAN wird über das network Paket der Micropython-Version hergestellt. Es ist eine Kommunikation mit unverschlüsselten sowie WPA2-PSK Verschlüsselten Netzwerken möglich. Ebenfalls sind Ansätze für WPA2-Enterprise basierte Authentifizierungen möglich. Anschließend sind API-Anfragen über die angepasste urequests Bibliothek möglich. Sie beinhaltet alle grundlegenden Implementationen für HTTP und HTTPS Verbindungen. Konkret wurde die WiFi Verbindung sowie die API-Verbindung in eine eigene Klasse gekapselt, um mithilfe entsprechender Methoden die Funktionen wiederverwenden zu können.

2.2.2 RFID-Reader

Die Kommunikation mit dem RFID-Reader basiert auf einer SPI-Kommunikation, wessen Spezifika in einer importierten Klasse definiert sind. Diese Funktionalitäten wurden jedoch erneut in eine eigene Klasse implementiert, sodass eine wartende Abfrage mit Rückgabe des Tokens einfach verwendbar ist. Dazu fragt der Controller ständig den Reader-Chip an, bis ein Lesevorgang erfolgreich war. In diesem Fall wird der Token des RFID-Tag zurückgegeben. Die Anschlusspins werden beim Initialisieren der Klasse standardmäßig gesetzt.

2.2.3 Input & Output

Das Relais wird ebenfalls über eine eigene Klasse gesteuert, dabei wird ausschließlich die grundlegende GPIO-Funktionalität verwendet. Die RGB-LED verwendet in ihrer Klasse ebenfalls ausschließlich die standardmäßige PWM-Implementation. Für das LCD mit dem I2C-Adapter wurde, wie bereits bei dem RFID-Reader, auf eine vorhandene Bibliothek zurückgegriffen und in einer eigenen Klasse implementiert. Beide Zugriffsklassen werden über eine weitere Klasse zusammengefasst, diese erlaubt eine zusammenpassende Anpassung der Anzeige entsprechend der auf einem nachgebauten Enum basierenden Zustand mit anpassbarer Displaytexte.

2.2.4 Programmablauf

Sobald die Steuereinheit mit Strom versorgt wird, beginnt der Raspberry Pi mit der Verbindung mit dem konfigurierten WLAN (*config.Network*) und zeigt anschließend die per DHCP erhaltene IP-Adresse an. Als nächstes erfolgt ein Verbindungsversuch mit der REST-API und Anzeige des Ergebnisses. Sollte einer der beiden Schritte fehlschlagen wechselt die Anzeige in einen Fehlerzustand und die Einheit startet nach 10 Minuten neu. War der Startprozess erfolgreich zeigt das Display eine Anleitung für die Verwendung der Anlage an, welche jedoch auch zusätzlich neben dem Gehäuse mit Bildern erklärt ist. Anschließend wartet die Steuerung auf den Scan eines RFID-Tags. Sobald ein Chip gescannt wurde, wird per HTTP-Request die

Berechtigung des Tokens geprüft. Sollte diese Anfrage fehlschlagen zeigt das Display die Fehlermeldung der API an, ansonsten das Ergebnis der Anfrage. Wird der Key verweigert wechselt das System wieder in den wartenden Zustand. Bei einer genehmigten Nutzung erhält die Steuerung die Information über die benutzerdefinierte Schaltzeit für das Relais und somit den Aktor. Während der Betätigung des Aktors wird die verbleibende Zeit auf dem Display angezeigt. Nach Ablauf der Zeit wechselt der Zustand zurück auf den initialen wartenden Zustand. Im Falle einer unerwarteten Exception an wird das System neugestartet.

3 REST-API

Serverseitig befindet sich eine moderne REST-API als Schnittstelle zur Datenbank und Sicherstellung einer Authentifizierung vor dem Zugriff auf die Daten. Alle Antworten werden in Form von JSON-Objekten gesendet. Der erste Verwendungszweck für die API ist die Authentifizierung der RFID-Tokens. Dazu wird für jeden Token geprüft, ob dieser aktuell einem User zugeordnet ist und sowohl der Key als auch der User einen Zugriff auf die Anlage besitzen. Im Falle eines genehmigten Zugriffs erhält die Steuerung die Öffnungszeit für das Relais, andernfalls eine entsprechende Information. Zusätzlich werden durch die API alle Anfragen für die Verwendung eines RFID-Tags protokolliert. Die andere Verwendung für die API ist durch die Verwaltungssoftware, sodass dort die Benutzer und Keys angelegt und entsprechend zugewiesen werden. Ebenfalls ist anschließend eine Erzeugung und Abfrage von Historien möglich.

3.1 Datenspeicherung

Alle Daten inklusive der erzeugten Log-Einträge werden in einer MySQL8-Datenbank gespeichert, welche exklusiv über die REST-API verwaltet wird. Dabei sind alle Daten mit Ablaufdaten möglich, um eine Löschung von Einträgen zu vermeiden.

3.2 Technologie

In der ersten Implementierung wurde aus Gründen der Einfachheit eine aus Python3 basierende REST-API mit *Flask* als Webserver und dem *mysql.connector* Paket für die Datenbankkommunikation entwickelt. In der weiteren Entwicklung soll dafür jedoch eine Node.JS Applikation erstellt werden um gleichzeitig als Webserver für die Benutzeroberfläche dienen zu können.

4 Verwaltungssoftware

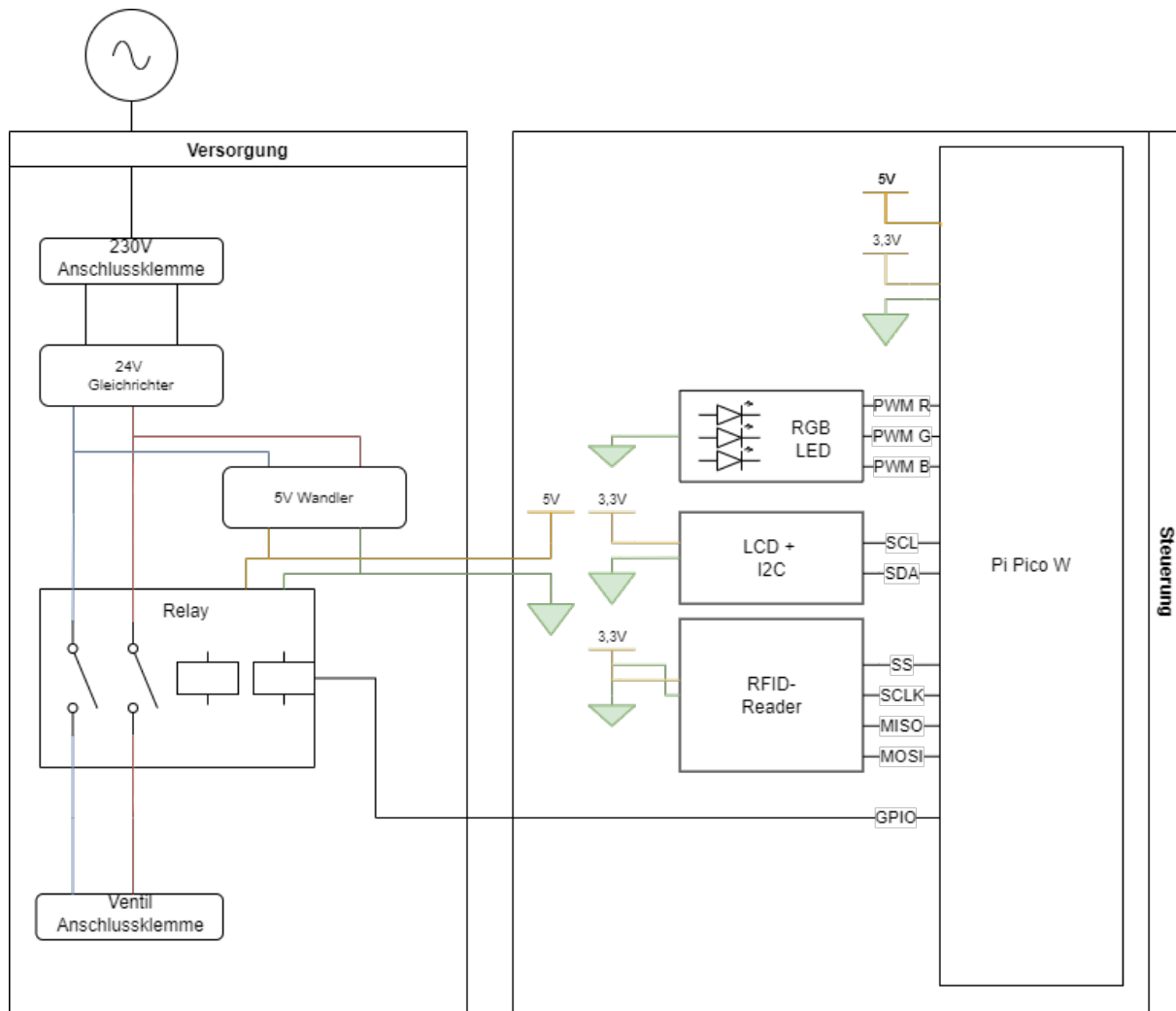
Für die Benutzerverwaltung und RFID-Zuordnung ist die Erstellung einer Node.js basierten Web-Applikation geplant. Diese soll dann direkt mit der ebenfalls enthaltenen REST-API kommunizieren. Im ersten Probelauf wird die Konfiguration jedoch direkt innerhalb der Datenbank der REST-API erledigt.

5 weiterführende Dokumente

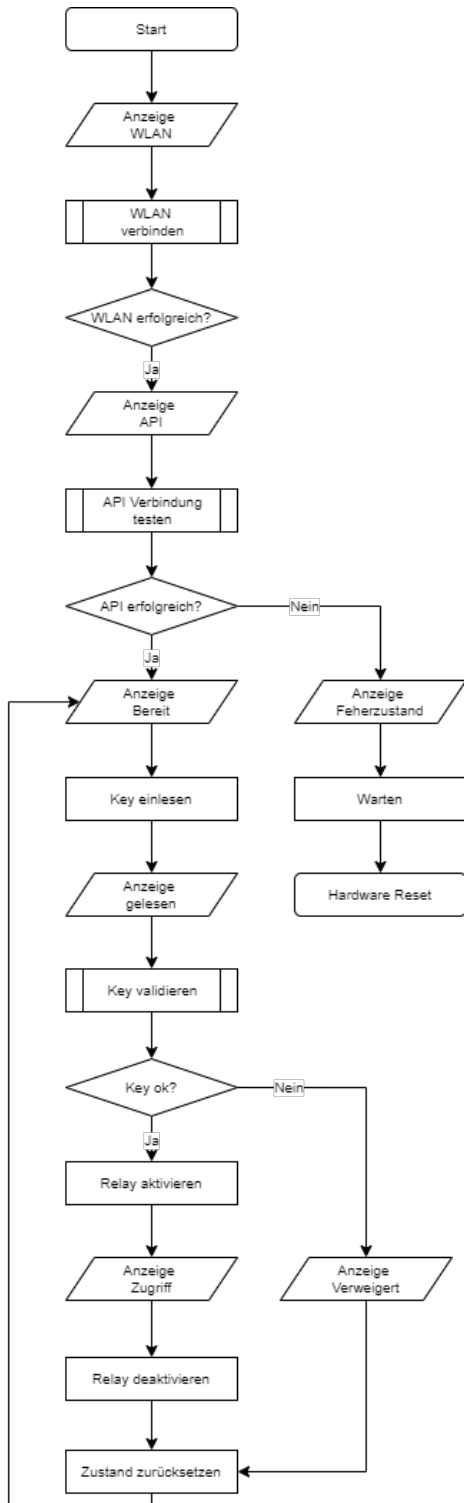
- Dokumentation inklusive Datenblätter:
<https://github.com/t-schoenijahn/sptvh-fuellanlage>
- Repository Controller Code:
<https://github.com/t-schoenijahn/sptvh-fuellanlage-controller>
- Repository Web Code:
<https://github.com/t-schoenijahn/sptvh-fuellanlage-web>
- Videovorstellung Projekt:
Noch ausstehend

6 Anhang

6.1 Schalt- und Anschlussplan



6.2 Programmablaufplan Controller



6.3 Workflow Nutzerverwaltung

1. Unterweisung des Mitglieds
2. Hinzufügen des Nutzers zur Verwaltungssoftware
3. Zuweisung eines (persönlichen) Token an den Benutzer
4. Möglichkeit die Füllanlage zu Benutzen
5. jährliche Auffrischung der Unterweisung
6. Erweiterung des Nutzungszeitraums in der Verwaltungssoftware
7. Automatisches Ende der Füllberechtigung bei ausbleiben der Unterweisung

6.4 Benutzeranleitung (Kurzfassung)

1. Stromverbindung herstellen, wenn nicht vorhanden
2. Startprozess der Steuereinheit abwarten
3. Flasche entsprechend der Anweisungen vorbereiten und mit dem Füllanschluss verbinden
4. RFID-Token an markierter Fläche der Steuerung scannen
5. Freischaltung Ventil abwarten
6. Flasche und Füllventil öffnen
7. Füllvorgang ggf. durch erneutes Scannen des Tokens wiederholen
8. Flaschenventil und anschließend Füllventil schließen
9. Flasche entfernen