



Aufgabenbeschreibung labortechnische Espressomaschine

Verlagerung der Regelkreise aus der MATLAB®-GUI auf die Mikrocontroller der Steuerungselektronik

Task description of laboratory espresso machine

Relocation of the control loops from the MATLAB® GUI to the microcontrollers of the control electronics

11. April 2024

Armin Rohnen
Hochschule München - Munich University of Applied Sciences
Fakultät 03 - Maschinenbau, Fahrzeugtechnik, Flugzeugtechnik
Dachauer Straße 98b, 80335 München
eMail: armin.rohnen@hm.edu

1 Zusammenfassung / Abstract

Dieses Dokument analysiert die Aufgabenstellung, welche sich aus der Zielsetzung der Verlagerung der Regelkreise für die labortechnische Espressomaschine ergibt. Diese sind aktuell in einer MATLAB[®]-GUI realisiert. Da die Bedienung der Espressomaschine im finalen Bauzustand nicht über einen externen PC erfolgen soll, müssen die Regelkreise auf die Mikrocontroller der Steuerung übertragen werden. Die aktuelle MATLAB[®]-GUI soll dabei in eine Wartungs-APP überführt werden.

This document analyses the task resulting from the objective of relocating the control loops for the laboratory espresso machine. These are currently realised in a MATLAB[®] GUI. As the espresso machine is not to be operated via an external PC in its final state, control loops must be transferred to the microcontroller of the control system. The current MATLAB[®] GUI is to be converted into a maintenance APP.

Inhaltsverzeichnis

1	Zusammenfassung / Abstract	2
2	Verzeichnis der Formelzeichen	4
3	Verzeichnis der Abkürzungen	4
4	Aktuelle Realisierung der Regelungstechnik	5
4.1	Steuerungselektronik	5
4.2	Regelungstechnik	7
4.2.1	Regelungstechnik - Boilerfüllstand	8
4.2.2	Regelungstechnik - Boilerdruck mit Entschichtung	8
4.2.3	Regelungstechnik - Mischwassertemperatur	8
4.2.4	Regelungstechnik - Durchflussregler	9
5	Verlagerung der Regelkreise auf die MCUs	10
5.1	Kommunikation per I2C	10
5.2	Kommunikation per UART (Konzept)	10
5.3	Mehrkernnutzung und/oder Multitasking	12
5.4	Verlagerung des Füllstandsreglers auf das Basisboard	12
5.5	Verlagerung der Boilerdruckregelung auf die Messplatine	14
5.6	Verlagerung des Mischtemperaturreglers auf das Basisboard	14
5.7	Verlagerung des Durchflussreglers auf das Basisboard	14
5.8	Erstellung der Funktionssteuerung - Preinfusion auf dem Basisboard	15
5.9	Erstellung der Funktionssteuerung - Kaffeebezug auf dem Basisboard	15
5.10	Erstellung der Funktionssteuerung - Wasserbezug auf dem Basisboard	15
5.11	Erstellung der Funktionssteuerung - Dampfbezug auf dem Basisboard	16
5.12	Erstellung der Funktionssteuerung - Spülen auf dem Basisboard	16
5.13	Erstellung der Funktionssteuerung - Grundreinigung auf dem Basisboard	17
5.14	Integration eines Displays mit Touchfunktion	17
5.15	Integration eines Vertikalhebels	18
5.16	Visualisierung und Interaktion über die MATLAB [®] -GUI	18
6	Literatur und Quellen	19

2 Verzeichnis der Formelzeichen

$p_{B,soll}$	[mbar]	Boilerdrucksollwert
$\vartheta_{M,soll}$	[°C]	Mischwassersolltemperatur
$\vartheta_{M,soll}(t)$	[°C]	Mischwassersolltemperaturverlauf
$\vartheta_{T,Boiler}$	[°C]	Boilertemperatur
$\dot{V}_{K,soll}$	[ml/s]	Solldurchflussrate (Flowrate) für den Kaffeebezug
$\dot{V}_{K,soll}(t)$	[ml/s]	Solldurchflussratenverlauf (Flowrateprofil) für den Kaffeebezug
$\dot{V}_{T,soll}$	[ml/s]	Solldurchflussrate (Flowrate) für den Teewasserbezug
U_P	[mV]	Pumpensteuerspannung
$U_{VH}(t)$	[mV]	Zeitverlauf Spannungswert des Vertikalhebels
$U_{VH,oben}$	[mV]	Spannungswert Vertikalhebel oberer Anschlag
$U_{VH,unten}$	[mV]	Spannungswert Vertikalhebel unterer Anschlag
$U_{VH,ue,oben}$	[mV]	Spannungswert Vertikalhebel oberer Anschlag überdrückt
$U_{VH,ue,unten}$	[mV]	Spannungswert Vertikalhebel unterer Anschlag überdrückt

3 Verzeichnis der Abkürzungen

ADC	Analog Digital Converter
I2C	Inter-integrated circuit, 2-Draht System zur Datenübertragung zwischen Peripheriegeräten
ISR	Interrupt Service Routine
MCU	Mikrocontroller Unit
PIN	einzelner, unterschiedlich nutzbarer Anschluss am Mikrocontroller
RxD	Leitung für eingehende Daten
SPS	Abtastrate Samples per Second
TxD	Leitung für ausgehende Daten
UART	Universal Asynchronous Receiver / Transmitter, Regelsatz für eine 2-Draht Datenübertragung zwischen Peripheriegeräten

Abbildungsverzeichnis

1	Madita vom Stein, Prozessschaubild der MATLAB®-GUI [103], Stand April-2024	5
2	Armin Rohnen, Konzept der Systemsteuerung mit drei Platinen, Stand April-2024	6
3	Armin Rohnen, Konzept der Systemsteuerung mit drei Platinen und Token-Ring-Kommunikation, Stand April-2024	11

Tabellenverzeichnis

1	Definition der Token	13
---	--------------------------------	----

4 Aktuelle Realisierung der Regelungstechnik

Für die Betriebssoftware der labortechnischen espressomaschine wurde eine (unvollständige) FMEA [79] durchgeführt. Trotz der Unvollständigkeit der FMEA ist dieses Dokument eine wichtige Hilfe in der Umsetzung.

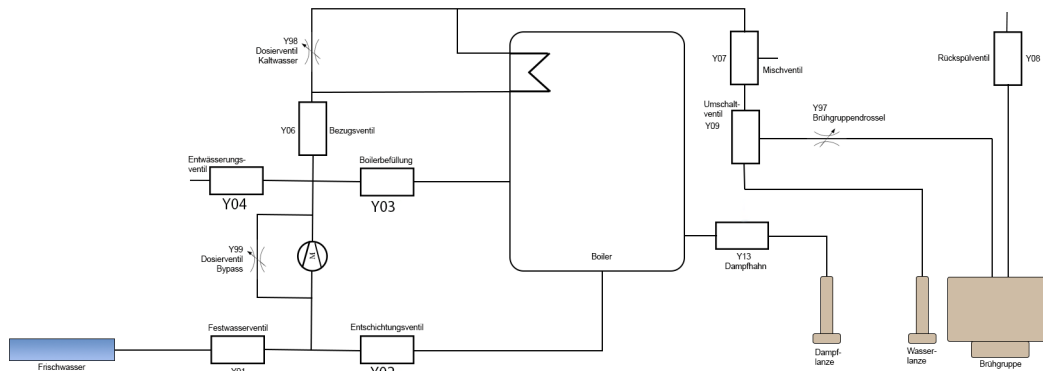


Abbildung 1: Madita vom Stein, Prozessschaubild der MATLAB®-GUI [103], Stand April-2024

Mit der Projektarbeit [103], wurde eine funktionsfähige MATLAB®-GUI mit einem Prozessschaubild (Abbildung 1) erstellt. Diese Projektarbeit beinhaltet den Kommunikationsaufbau zwischen den drei Mikrocontroller der Systemsteuerung und der MATLAB®-GUI, sowie den erforderlichen Datenaustausch zwischen den einzelnen Platinen der Steuerung und der MATLAB®-GUI. Eine Kommunikation zwischen den Platinen ist im Platinenlayout vorgesehen, findet jedoch nicht statt. Diese Kommunikation wird zu diesem Projektstand nicht benötigt.

4.1 Steuerungselektronik

Der Systemaufbau der Maschinensteuerung (Abbildung 2) sieht drei Platinen mit jeweils einem eigenen Raspberry Pi Pico Mikrocontroller vor. Feste Vorgabe im Projekt ist die Verwendung von MicroPython als Laufzeitumgebung auf den MCUs. Für die Kommunikation untereinander ist jeweils die erste UART-Schnittstelle der jeweiligen MCU mit den PINs TxD und RxD auf die Klemmenleisten herausgeführt. Der Anschluss eines Displays ist via UART von einer der drei Platinen vorgesehen, die dazu über eine weitere herausgeführte TxD und RxD PINs verfügt. Die Verdrahtung der Kommunikation ist aktuell nicht durchgeführt und die konkrete Form der Kommunikation ist undefiniert. Die Kommunikation der MCUs erfolgt über USB (virtueller serielle Schnittstelle) zur MATLAB®-GUI und zurück.

Die Basisplatine sendet in dem fest eingestellten Takt von 8 je Sekunde die Durchflussrate, Anzahl der erfassten Flowmeter-Impulse und den erkannten Boilerfüllstand an die MATLAB®-GUI. Je nach Bedarf durch die Regelungs- und Steuerungsberechnung in der GUI werden der Stellwert der Pumpensteuerspannung, und die Verstellschrittzahlen für die Brühgruppendrossel und dem Pumpenbypass von der GUI an die Basisplatine übermittelt. Der Pumpenbypass wurde stillgelegt und die Brühgruppendrossel wird nur während der Initialisierung verstellt. Aktuell ist noch ungeklärt ob Zukünftig eine Brühgruppendrossel benötigt wird und ob diese über einen Schrittmotor angesteuert wird.

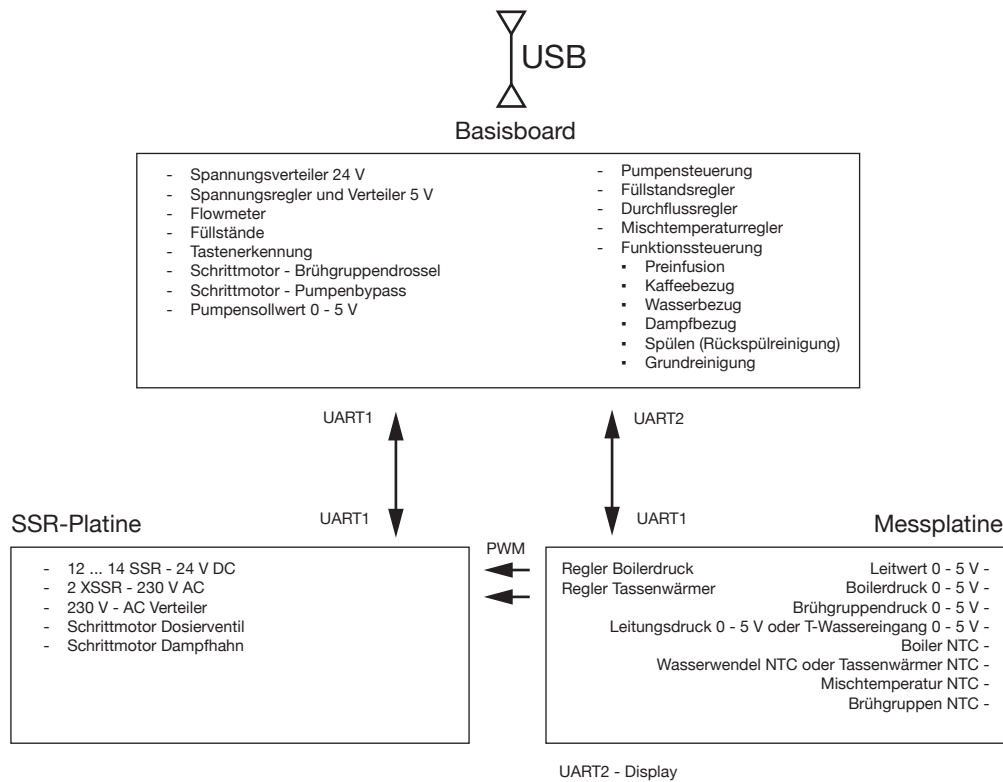


Abbildung 2: Armin Rohnen, Konzept der Systemsteuerung mit drei Platinen, Stand April-2024

Die Tastenerkennung ist derzeit inaktiv. Die Regelkreise Füllstandsregler, Durchflussregler und Mischtemperaturregler werden in der MATLAB®-GUI abgearbeitet. Gleiches gilt für die Funktionssteuerungen, wovon allerdings lediglich der einfache manuelle Kaffeebezug funktional ist. Vorbereitet wurde durch [103] die Funktionalität Flush/Rückspülreinigung.

Die Messwertplatine sendet, wenn die acht Messwerte erfasst sind, diese an die MATLAB®-GUI. Das ergibt einen Übermittlungstakt, der in Abhängigkeit der Abtastrate des AD-Wandlers steht und könnte theoretisch bis zu 53 Messdatensätze je Sekunde groß werden. Die Messwerte auf der Messwertplatine werden durch einen ADS1115 16-Bit-ADC erfasst. Dieser wird im kontinuierlichen Mode betrieben und meldet über einen Interrupt den Abschluss der Wandlung an die MCU. Über einen von der MCU gesteuerten 8-Kanal-Multiplexer erfolgt die Durchschaltung der 8 Analogeingänge der Platine auf den einen Analogeingang des ADCs. Da die erste AD-Wandlung während des Umschaltens am Multiplexer erfolgt, muss prozessbedingt jeder Messkanal zweimal erfasst werden. Das erste Wandlungsergebnis wird verworfen und es halbiert sich daher die maximalmögliche Abtastrate.

Die beiden Regelkreise auf der Messplatine sind aktuell inaktiv. Diese Funktionalität übernimmt die MATLAB®-GUI und die erforderlichen PWM-Signale werden auf der SSR-Platine direkt erzeugt. Auch die Verdrahtung der PWM-Leitungen von der Messplatine zur SSR-Platine existiert derzeit nicht.

Die Messwertreihenfolge der Datenübermittlung zur GUI ist derzeit mit

1. Boilertemperatur

2. BoilertemperaturMischwassertemperatur
3. nicht belegt, also 0 Wert
4. Brühgruppentemperatur
5. Brühgruppendruck
6. Boilerdruck
7. Leitungsdruck
8. Wassertemperatur vor der Pumpe

definiert und muss auch weiterhin so an die GUI gesendet werden.

Die SSR-Platine schaltet die Magnetventile und über ein XSSR erfolgt die Leistungssteuerung des Boilerheizelements. Ein Tassenwärmer ist aktuell nicht angeschlossen. In der aktuell verbauten Version sind die Magnetventile mit 230 V AC Schaltung. Von den Schrittmotorsteuerungen ist aktuell das Dosierventil aktiv. Um ein Magnetventil zu schalten ist lediglich die Zustandsänderung an einem PIN erforderlich. Für die Schrittmotorverstellung müssen Schrittsequenzen durchfahren werden. Diese blockieren die weitere Ausführung von Programmen und ggf. auch die Kommunikation auf den MCUs.

Die Anweisungen für die Steuerungen erfolgen aus der MATLAB[®]-GUI.

Unter [119] befindet sich eine Sammlung von Verlinkungen zu Libraries, Frameworks, Programmen und Ressourcen für MicroPython. Allerdings sind nicht alle Tools auf der Raspberry Pi Pico MCU (RP2040) lauffähig.

Der MicroPython-Code nutzt aktuell je einen Prozessorkern der MCUs. Die volle Nutzung der Dual-Core-MCUs ist derzeit in MicroPython nicht realisiert. In der Linksammlung [119] befinden sich einige Beispiele zum Scheduling und zu Frameworks mit denen diese Problematik u. u. behoben werden könnte. Die Programmierung mit uasyncio wäre eine weitere Möglichkeit diese Problematik zu behandeln.

4.2 Regelungstechnik

Im aktuellen Stand sind

- Boilerfüllstand, was keine Regelung im regelungstechnischen Sinn ist
- Boilerdruck mit Entschichtung
- Mischwassertemperatur
- Durchflussrate

als Regelkreise und ein rudimentärer manueller Kaffeebezug mit Vorgabe der Mischwassertemperatur sowie der Durchflussrate realisiert.

Die Berechnung der Regelkreise erfolgt in der MATLAB[®]-GUI. Das erforderliche Hochfahren der Espresso-Maschine, in der Abfolge von Boilerbefüllung und Boileraufheizen, erfolgt in manueller Betätigung. Ursächlicher Auftrag ist, diese Regelkreise auf die MCUs zu verlagern und das Hochfahren der Espresso-Maschine zu automatisieren.

4.2.1 Regelungstechnik - Boilerfüllstand

Der Boilerfüllstand wird über Kurzschlussdetektion erkannt. Es stehen 4 Kurzschlussdetektoren zur Verfügung, welche im Zusammenhang mit der Glasboilermaschine auch alle benötigt werden. In der labortechnischen Espresso-Maschine wird einer davon für den Boilerfüllstand benötigt.

Wird kein Kurzschluss detektiert, wird dies als unzureichender und undefinierter Füllstand interpretiert. Dieser Zustand schaltet den Boilerdruckregler ab, da ggf. ein nicht gekühltes Heizelement vorliegen könnte. Es werden die erforderlichen Magnetventile für die Boilerbefüllung geschaltet und die Pumpe mit Steuerungsspannung 4000 mV auf eine definierte Pumpendrehzahl eingestellt. Ist der Sollfüllstand erreicht wird dieser um 1,5 Sekunden mit Pumpensteuerungsspannung 1000 mV überfüllt, um ein flattern der Befüllsteuerung zu unterbinden. Anschließend wird in der Reihenfolge Magnetventilschaltungen und Pumpenabschaltung der Vorgang beendet.

Die Befüllung des Boilers führt zur Erhöhung des Boilerdrucks. Das wird beim kalten Boiler durch Öffnung des Dampfahns zu verhindern. Es wird auch nur dann problematisch, wenn große Wassermengen nachgefüllt werden müssen. Was lediglich im kalten Zustand vorkommt.

Diese Steuerung erfolgt in der MATLAB[®]-GUI und für die Verriegelung der Boilerdruckregelung wird ein Steuerungsflag verwendet.

4.2.2 Regelungstechnik - Boilerdruck mit Entschichtung

Bei kurzschlussdetektiertem Füllstand des Boilers darf die Boilerdruckregelung aktiv sein. Solange die Boiler-temperatur kleiner 95 °C ist, erfolgt zusätzlich die Boilerentschichtung. Zur Aktivierung der Boilerentschichtung wird die Pumpensteuerungsspannung auf 2500 mV eingestellt und die Magnetventile für die Entschichtung geschaltet. Bei Erreichen der Boilertemperatur von 95 °C erfolgt die Abschaltung in umgekehrter Reihenfolge.

Ab $\vartheta_{T,Boiler} = 105$ °C kann der Kaffee- bzw. Teewasserbezug freigegeben werden.

Die Boilerdruckregelung erfolgt für einen Sollwert aus der MATLAB[®]-GUI. Dieser ist im Nennwert $p_{B,soll} = 1350$ mbar und ist im Bereich von $p_{B,soll} = 1000$ mbar bis $p_{B,soll} = 1500$ mbar veränderlich. Nach [114] wird für die Boilerdruckregelung ein PID-Regler verwendet. Stellgröße ist ein PWM-Signal zwischen 0 und 100 % generiert. Beim geöffneten Dampfahn wird der Sollwert $p_{B,soll}$ um 200 mbar nach unten korrigiert. Damit wird ein unnötiges Aufheizen des Boilers unterbunden. Der Stellwert des PWM-Signals wird an die SSR-Platine übermittelt.

4.2.3 Regelungstechnik - Mischwassertemperatur

Kaffee- und Teewasserbezug erfolgen mit geregelter Mischwassertemperatur $\vartheta_{M,soll}$. Die Regelung erfolgt für einen Sollwert aus der MATLAB[®]-GUI. Es soll zukünftig möglich sein auch einen Sollwertverlauf $\vartheta_{M,soll}(t)$ zu regeln. Der Wertebereich dazu liegt zwischen $\vartheta_{M,soll} = 80$ °C und $\vartheta_{M,soll} = 96$ °C. Die Regelung erfolgt nach [114] über einen PID-Regler. Die Stellgröße ist die Verstellung des Mischventils in Form von 1/2-Schritten mit Richtungsvorzeichen und wird von der MATLAB[®]-GUI an die SSR-Platine übermittelt.

Das Einschwingen der Mischwassertemperatur erfolgt mit einem Vorsteuerwert für die Pumpensteuerung und der letzte Stellwert des Mischventils bleibt nach dem Kaffeebezug für den nächsten Kaffeebezug erhalten.

Wenn die Mischwassertemperatur erreicht ist und der Mischwasserregler sich im eingeschwungenen Zustand befindet wird über Schaltung der Magnetventile von Wasserüberlauf auf Kaffee- oder Teewasserbezug umgeschaltet.

4.2.4 Regelungstechnik - Durchflussregler

Der Durchflussregler stellt sicher dass bei einem Espresso Bezug $\dot{V}_{K,soll} = 1 \text{ ml/s}$ für den Eintassenbezug, ein $\dot{V}_{K,soll} = 2 \text{ ml/s}$ für den Zweitassenbezug bzw. ein beliebiges $\dot{V}_{K,soll}(t)$ geregelt wird. Für den Teewasserbezug sind aktuell $\dot{V}_{T,soll} = 10 \text{ ml/s}$ definiert.

Der Durchflussregler wird aktiv, wenn der Mischwasserregler eingeschwungen ist und seinen Sollwert erreicht hat. Der Durchflussregler ist nach [114] als PID-Regler mit Anti-Windup realisiert. Die Pumpensteuerungsspannung U_P wird aktuell auf den Spannungsbereich $U_P = 0 \text{ mV}$ bis $U_P = 2100 \text{ mV}$ begrenzt. Der obere Wert stellt aktuell eine Absicherung dar und gilt über Versuchsreihen genauer zu definieren.

Der Spannungssollwert wird von der MATLAB[®]-GUI an die Basis-Platine übermittelt. Durch $U_P = 0 \text{ mV}$ wird der Kaffee- bzw. Teewasserbezug beendet. Dies erfolgt aufgrund Interaktion mit der MATLAB[®]-GUI und soll in einer späteren Version aufgrund der erfassten Füllmenge erfolgen.

5 Verlagerung der Regelkreise auf die MCUs

Für die Verlagerung der Regelkreise von der MATLAB®-GUI auf die MCUs ist zwingend die Herstellung einer Kommunikation zwischen den MCUs und eine Änderung in der Kommunikation zur MATLAB®-GUI erforderlich. Bei der Kommunikation der MCUs untereinander ist die Einbindung eines Displays und eine zukünftige WLAN-Verbindung mit zu betrachten. Letzteres führt zur Überlegung dass eine weitere MCU für die Funktionalität WLAN und Display hinzugefügt wird.

Wenn die Kommunikation der MCUs untereinander an einem Prototypen funktional erprobt ist, kann die eigentliche Verlagerung der Regelkreise und die damit verbundene Abänderung der MATLAB®-GUI durchgeführt werden.

Für die Kommunikation der MCUs untereinander kommen zwei verschiedenen Kommunikationsschnittstellen in Betracht.

5.1 Kommunikation per I2C

Ein Datenaustausch über I2C, bei dem eine MCU als Slave konfiguriert ist, ist unter der Linksammlung auf [119] beschrieben. Allerdings handelt es sich bei diesem Beispiel um eine Anwendung für den Pyboard I2C-slave mode. Die Raspberry Pi Pico Hardware (RP2040 MCU) Hardware unterstützt keinen I2C-Slave-Mode. Obendrein würden mehr als die zwei herausgeführten PINs benötigt werden.

5.2 Kommunikation per UART (Konzept)

Seitens des Platinenlayouts für die Steuerungselektronik wurden die jeweiligen PINs der ersten UART-Schnittstelle für die spätere Inter-Kommunikation der Steuerung herausgelegt. Die Raspberry Pi Pico MCU verfügt über einen jeweils 32 Byte großen Eingangs- bzw. Ausgangspuffer für die UART-Schnittstelle. Dieser soll genutzt werden um die Kommunikation zu ermöglichen. Die jeweilige Kommunikation selbst wird dadurch auf diese 32 Byte begrenzt. Hardwareseitig ist es denkbar, dass je MCU zwei weitere PINs für die Kommunikation verwendet werden. Lediglich auf der SSR-Platine ist dazu ein Eingriff ins Platinenlayout erforderlich. Dieser kann bei einem Prototypen handwerklich durchgeführt werden. Die zwei zusätzlichen PINs für die Kommunikation (GP04, GP05 auf der Basisplatine, GP03, GP04 auf der SSR-Platine, GP06, GP07 auf der Messplatine) fungieren als eingehender und ausgehender Interrupt, so dass die Kommunikation als ISR programmiert werden kann.

Angelehnt an das 1984 von IBM eingeführte Token Ring Network soll eine „im Kreis laufende“ Kommunikation aufgebaut werden. Ausgehend von der Messplatine wird ein Token über die einzelnen Platinen durchgereicht (Abbildung 3). Unter Berücksichtigung der Kommunikationsregeln können so eine beliebige Anzahl an Steuerungsplatinen in den Kommunikationsring eingebunden werden. Allerdings sollte dabei die Reihenfolge beachtet werden.

Die Messplatine sendet über UART am Ende jedes gültigen Messwertsatzes einen oder mehrere Token an die Basisplatine. In den Token befinden sich die aktuellen Messwerte und alle anderen „im Kreis laufende“ Daten. Beim ersten Senden werden hier u.u. 0-Werte verwendet, bis dass der erste Token wieder bei der Messplatine angekommen ist. Ggf. unterbricht die Messplatine auch das Senden des Token bzw. der ersten Token-Serie, bis zum ersten Eintreffen eines Tokens. Die Messplatine ist damit der Taktgeber der Kommunikation. Konkret erfolgt dies in Abhängigkeit der Abtastrate des AD-Wandlers. Wenn ein Token versendet

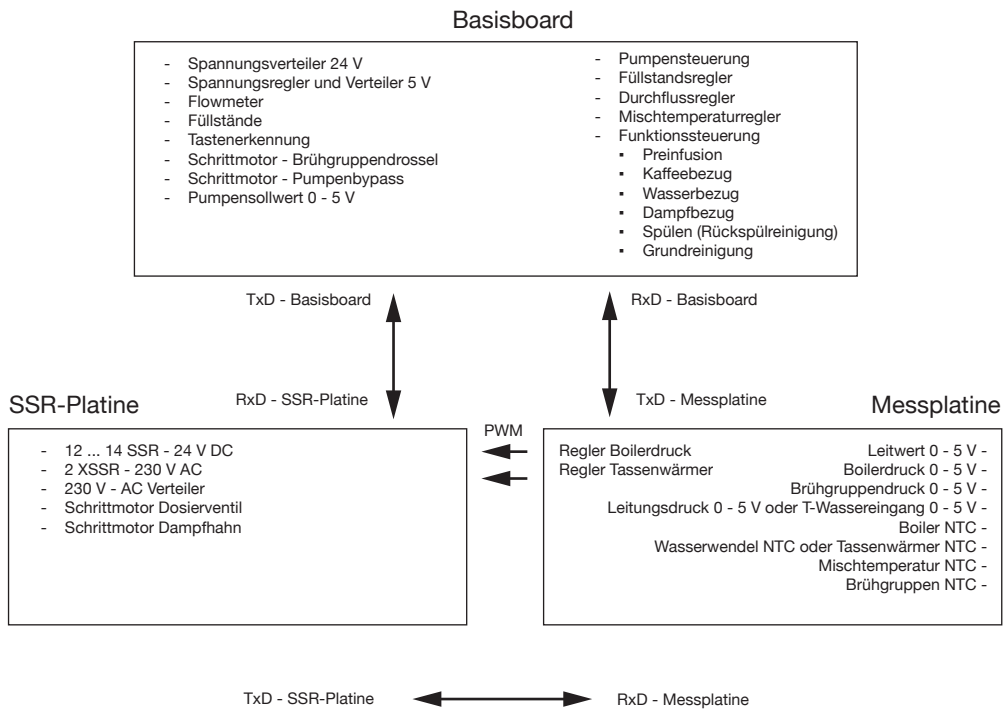


Abbildung 3: Armin Rohnen, Konzept der Systemsteuerung mit drei Platinen und Token-Ring-Kommunikation, Stand April-2024

ist, wechselt die Messplatine den Logikzustand des ausgehenden Interrupt-PINs. Bevor der nächste Token versendet wird, wird wieder der Ausgangslogikzustand des Interrupt-PINs hergestellt.

Sobald die Messplatine auch einen Token empfangen hat wird auch der PWM-Stellwert für das Boilerheizelement (aus Regler Boilerdruck) und für den Tassenwärmer im Token übermittelt.

Von der Messplatine erhält die Basisplatte den Token. Die UART-Schnittstelle wird ausgelesen, wenn am eingehenden Interrupt-PIN der Logikzustand wechselt. Die Basisplatte tauscht im Token die Daten für die eigenen Messwerte, die Schaltzustände der Tasten, die Schaltzustände der Magnetventile usw. aus und sendet den Token an die SSR-Platine weiter. Ist der Token versendet, wechselt die Basisplatte den Logikzustand des ausgehenden Interrupt-PINs und arbeitet in gleicher Logik weiter wie die Messplatine.

Die SSR-Platine bekommt von der Basisplatte den Token durchgereicht. Die SSR-Platine wertet den Token lediglich aus, nimmt keine Veränderungen darin vor und reicht diesen dann weiter.

Nach der SSR-Platine kann eine Display-Platine nachgeschaltet werden. Diese wertet die Informationen im Token aus und visualisiert diese bzw. stellt die Informationen über WLAN zur Verfügung. Im Falle eines Touch-Displays werden die Bytes für die Schaltzustände der Tasten ausgetauscht.

Abschließend erhält die Messplatine wieder den Token und übernimmt die relevanten Informationen daraus in den neu zu sendenden Token.

Anstelle des eingehenden und ausgehenden Interrupt-PINs kann u.u. auch die MicroPython-Funktionalität `uart.irq` verwendet werden. Diese ruft eine ISR auf, sobald Daten im Eingangspuffer der UART-Schnittstelle eingegangen sind.

Das Konzept ist auszugestalten und der bzw. die Token sind exakt zu definieren. Dabei ist auch zu definieren (und zu erproben) wie die dann übergeordnete MATLAB[®]-GUI die Daten für die Visualisierung erhält.

Der einzelne Token darf aufgrund der Limitierung durch den UART-Puffer des Raspberry Pi Picos 32 Byte nicht überschreiten. Als maximale Datenrate sind 115200 Baud möglich, was unter Berücksichtigung von START / STOP-Bits und Paritätsbit eine maximale Kommunikationsrate von 11520 Byte je Sekunde ergibt. Damit wären 360 jeweils 32 Byte lange Token in der Kommunikation möglich.

Nach aktuellem Stand der Recherchen überschreitet die Datenmenge für die Kommunikation die 32 Byte Grenze, so dass mit mehreren Token gearbeitet werden muss. Jeder Token weist im ersten Byte eine Token-Nr auf, so dass der Token zweifelsfrei identifiziert werden kann. Alle Token müssen die gleiche Länge in Bytes aufweisen. Für zukünftige Erweiterungen sind Leer-Bytes in den Token vorhanden. Jeder der 4 Token muss die gleiche Länge in Bytes aufweisen.

Getaktet wird die Kommunikation durch die Datenerfassung der Messplatine. Aufgrund der jeweils ungültigen ersten Wandlung eines Messkanals ist eine maximale Abtastrate von 430 Messwerten je Sekunde aufgeteilt auf acht Messkanäle, was ca. 53 Datensätze je Sekunde ergibt. Bei einer Abtastrate von 475 SPS ergeben sich 30 gültige Messdatensätze je Sekunde. Wird nach jedem zweiten gültigen Messkanal ein 25 Byte langer Token von der Messplatine gesendet, dann ergibt dies eine Inter-Kommunikation zwischen den MCUs von 120 Token bzw. 3000 Byte je Sekunde, was unter der als Maximum angesehenen Datenrate liegt. Selbst bei einer Verlängerung der Token auf 32 Byte würde lediglich 1/3 des theoretisch möglichen Datentransfers durchgeführt.

5.3 Mehrkernnutzung und/oder Multitasking

Grundsätzlich verfügt der ARM-Prozessor der Raspberry Pi Pico MCU RP2040 über zwei Kerne, die im Gegensatz zu anderen MCUs gleichwertig sein sollen. Die Quellenlage hierzu lässt jedoch eine eindeutige Beurteilung dieser Annahme nicht zu. Eine ggf. verbessernde Quellenlage könnte hierzu neue Ansätze liefern. Für die Verlagerung der Regelkreise auf die MCUs ist zuvor eine Festlegung zu treffen in welcher Form eine Mehrkernnutzung und/oder ein Multitasking implementiert wird.

Damit ist ein Framework für die Systemsteuerung zu definieren. Hierzu sind in der Linksammlung [119] unter Scheduling, Threading und Frameworks einige Libraries aufgelistet.

MicroPython stellt hierzu die Module `uasyncio` und `_thread` zur Verfügung. Letzteres verwendet für die Programmausführung den zweiten Prozessorkern. `_thread` wird allerdings als sehr experimentell bezeichnet. Allerdings lassen sich keine Negativinformationen recherchieren.

5.4 Verlagerung des Füllstandsreglers auf das Basisboard

Der Füllstandsregler soll wie unter 4.2.1 beschrieben aus der MATLAB[®]-GUI auf die Basisplatine verlagert werden. Negativer Füllstand verriegelt den Boilerdruckregler auf der Messplatine. Der Füllstandsregler wird

Tabelle 1: Definition der Token

Byte	Token 1	Token 2	Token 3	Token 4
0	Token Nr - 1	Token Nr - 2	Token Nr - 3	Token Nr - 4
1 (Highbyte)	Leitwert	BG-Drossel-1	BG-Drossel-2	BG-Drossel-3
2 (Lowbyte)	Leitwert	BG-Drossel-1	BG-Drossel-2	BG-Drossel-3
1 (Highbyte)	Brühgruppendruck-1	Boilertemperatur-1	Mischtemperatur-1	T-Brühgruppe-1
2 (Lowbyte)	Brühgruppendruck-1	Boilertemperatur -1	Mischtemperatur-1	T-Brühgruppe-1
3 (Highbyte)	Brühgruppendruck-2	Boilertemperatur-2	Mischtemperatur-2	T-Brühgruppe-2
4 (Lowbyte)	Brühgruppendruck-2	Boilertemperatur -2	Mischtemperatur-2	T-Brühgruppe-2
5 (Highbyte)	Brühgruppendruck-3	Boilertemperatur-3	Mischtemperatur-3	T-Brühgruppe-3
6 (Lowbyte)	Brühgruppendruck-3	Boilertemperatur -3	Mischtemperatur-3	T-Brühgruppe-3
7 (Highbyte)	Boilerdruck	Leitungsdruck (später T-Wasser)	T-Tassenwärmer (aktuell frei)	Dampfhahn
8 (Lowbyte)	Boilerdruck	Leitungsdruck (später T-Wasser)	T-Tassenwärmer (aktuell frei)	Dampfhahn
9	Tasten 1 bis 8	Magnetventil 1 - 8	Magnetventil 25 - 32	Füllstand 1 - 8
10	Tasten 9 bis 16	Magnetventil 9 - 16	Magnetventil 33 - 40	0 - frei
11	Tasten 17 bis 24	Magnetventil 17 - 24	Magnetventil 41 - 48	0 - frei
12 (Highbyte)	Inc-Flowmeter-1	Inc-Flowmeter-2	Inc-Flowmeter-3	PWM-Tassenwärmer
13 (Lowbyte)	Inc-Flowmeter-1	Inc-Flowmeter-2	Inc-Flowmeter-3	PWM-Tassenwärmer
14 (Highbyte)	Dosierventil-1	Dosierventil-2	Dosierventil-3	TSoll-Mischer-1
15 (Lowbyte)	Dosierventil-1	Dosierventil-2	Dosierventil-3	TSoll-Mischer-1
16 (Highbyte)	PWM-Heizelement-1	PWM-Heizelement-2	PWM-Heizelement-3	TSoll-Mischer-2
17 (Lowbyte)	PWM-Heizelement-1	PWM-Heizelement-2	PWM-Heizelement-3	TSoll-Mischer-2
18 (Highbyte)	U-Pumpe-1	U-Pumpe-2	U-Pumpe-3	TSoll-Mischer-3
19 (Lowbyte)	U-Pumpe-1	U-Pumpe-2	U-Pumpe-3	TSoll-Mischer-3
20 (Highbyte)	Flow-Soll-1	Flow-Soll-2	Flow-Soll-3	0 - frei
21 (Lowbyte)	Flow-Soll-1	Flow-Soll-2	Flow-Soll-3	0 - frei
22 (Highbyte)	0 - frei	0 - frei	0 - frei	0 - frei
23 (Lowbyte)	0 - frei	0 - frei	0 - frei	0 - frei
24 (Lowbyte)	0 - frei	0 - frei	0 - frei	0 - frei

durch den Maschinenstart aktiviert. Dieser wiederum wird über die MATLAB[®]-GUI durch Start der Messwerterfassung angestoßen. Im finalen Stand soll der Hauptschalter der Maschine die Messwerterfassung mit einer Zeitverzögerung starten.

5.5 Verlagerung der Boilerdruckregelung auf die Messplatine

Die Boilerdruckregelung soll wie unter 4.2.2 beschrieben auf die Messplatine verlagert werden. Das PWM-Signal für die Steuerung des Heizelements wird dabei auf der Messplatine erzeugt und über die Verdrahtung an die SSR-Platine weiter geleitet. Dabei muss das System zwischen Aufheizen und Normal-Regelung unterscheiden. In der Aufheizphase $\vartheta_{T,Boiler} < 105\text{ °C}$ müssen die Funktionssteuerungen deaktiviert bleiben.

Auf dem Basisboard muss mit der Entschichtung durch Umpumpen eine weitere Funktionalität eingeführt werden. Die Entschichtung ist lediglich während der Aufheizphase aktiviert, solange $\vartheta_{T,Boiler} < 95\text{ °C}$ ist. Die aktive Entschichtung oder $\vartheta_{T,Boiler} < 95\text{ °C}$ verriegelt die Funktionssteuerung.

Die Boilerdruckregelung ist dauerhaft aktiv und wird lediglich durch die negative Kurzschkussdetektion des Boilerfüllstands zeitweilig deaktiviert.

5.6 Verlagerung des Mischtemperaturreglers auf das Basisboard

Die Mischwassertemperaturregelung soll wie unter 4.2.3 beschrieben auf das Basisboard verlagert werden. Die Auslösung für den Kaffee- bzw. Teewasserbezug erfolgt in der finalen Version über die Tasten. Bis zur Fertigstellung der finalen Version ist der Auslöser die MATLAB[®]-GUI. Die erforderlichen Stellvorgänge werden entweder auf der Platine selbst umgesetzt oder über den Token an die SSR-Platine übermittelt.

Der Sollwert für die Mischwasserregelung wird über den Tastendruck oder die Auslösung durch die MATLAB[®]-GUI übermittelt. Im Zweifelsfall ist dies der Erstwert im Temperaturverlaufsprofil.

Wenn die Mischwassertemperatur erreicht ist und der Mischwasserregler sich im eingeschwungenen Zustand befindet, wird über Schaltung der Magnetventile auf Kaffee- oder Teewasserbezug umgeschaltet und der Durchflussregler gestartet.

Der Verstellwert der Stellgröße, die Anzahl der zu verstellenden 1/2-Schritte nebst Richtungsvorzeichen, werden über den Token an die SSR-Platine übermittelt. Ebenso werden die zu stellenden Magnetventile behandelt.

5.7 Verlagerung des Durchflussreglers auf das Basisboard

Der Durchflussregler Mischwassertemperaturregelung soll in der Funktionalität wie unter 4.2.3 beschrieben auf das Basisboard als Kaskadenregler zum Mischtemperaturregler verlagert werden. Ist der Durchflussregler aktiv ergibt sich u.u. für den Mischtemperaturregler ein Temperaturprofil. Auch für die einzuregelnde Durchflussrate sind Verlaufsprofile möglich.

Die Sollwerte werden über den Tastendruck oder die Auslösung durch die MATLAB[®]-GUI übermittelt.

Der Kaffee- bzw. Teewasserbezug wird durch Tastendruck aktuell durch die MATLAB[®]-GUI und zukünftig auch bei Erreichen einer Bezugsmenge beendet.

Die Pumpenleistung wird durch Stellen der Pumpensteuerung direkt auf dem Basisboard eingestellt. Alle anderen Stellgrößen werden über den Token an die SSR-Platine übermittelt.

5.8 Erstellung der Funktionssteuerung - Preinfusion auf dem Basisboard

Für die Realisierung einer Preinfusionssteuerung liegen derzeit keine ausreichenden Informationen vor. Nach Abschluss der Messreihen vorhandener Espressomaschinen wird über ein Konzept zur Realisierung einer Preinfusion entschieden werden. Bis dahin bleibt diese Funktionssteuerung on hold.

Unter Preinfusion wird der Zeitraum verstanden, der zwischen einschalten der Pumpe und dem ersten, aus dem Siebträger heraustropfenden, sichtbaren Kaffeetropfen verstreicht. Diese Zeitspanne wird durch eingestellte bzw. geregelte Pumpenleistung unterschiedlich gestaltet. Sie bewirkt, dass Geschmackstoffe aus dem Kaffeepuck herausgelöst werden, sich der Kaffeepuck mit Wasser füllt und aufquillt, ohne dass Kaffee gefördert wird. Eine gewisse (Grund)Preinfusion findet immer statt, da immer eine Menge Wasser vom Kaffeepuck aufgenommen wird, bevor der Kaffee aus dem Siebträger austritt. Bei der Preinfusions-Steuerung (Regelung) handelt es sich um eine bewusste Verlängerung dieser bereits vorhandenen Phase.

5.9 Erstellung der Funktionssteuerung - Kaffeebezug auf dem Basisboard

Für einen Kaffeebezug in der Siebträger-Espressomaschine ist die Abfolge Mischwasserregelung - Durchflussregelung - Fördermenge erforderlich.

Durch Tastendruck bzw. durch Auslösung auf der MATLAB[®]-GUI für der Kaffeebezug gestartet. Es stehen 4 Tasten für den Kaffeebezug zur Verfügung. Diese können für 4 unterschiedliche Kaffeerezepte (Presets) verwendet werden. Ein Tastenpreset (Kaffeerezept) ist beschrieben durch die Kaffeebezugstemperatur, was die Mischwassersolltemperatur $\vartheta_{M,soll}(t)$ ergibt, die gewünschte Flowrate $\dot{V}_{K,soll}$ und die Kaffeebezugsmenge. Mischwassersolltemperatur und Flowrate können auch Werteverläufe über die Zeit (Profile) sein.

Der Kaffeebezug startet mit einem Vorsteuerwert für die Pumpenleistung und beginnt mit dem Einschwingen des Mischtemperaturreglers. Während der Einschwingphase des Mischtemperaturreglers wird das Bezugswasser in den Maschinenüberlauf geleitet. Ist die Mischwassertemperatur erreicht und der Mischwasserregler befindet sich im eingeschwungenen Zustand wird durch Schalten der zugehörigen Magnetventile auf Kaffeebezug umgeschaltet und es wird der Durchflussregler gestartet. Wenn die gewünschte Kaffeemenge erreicht ist wird der Kaffeebezug durch abschalten der Pumpe und dem Ruhezustand der Magnetventile beendet. Die beteiligten Regelkreise werden abgeschaltet, die letzten vorhandenen Stellgrößen werden als Vorsteuerwerte für den nächsten Kaffeebezug gespeichert. Die Abschaltung erfolgt entweder durch zweiten Tastendruck oder durch Füllmengenberechnung.

5.10 Erstellung der Funktionssteuerung - Wasserbezug auf dem Basisboard

Für einen Teewasserbezug in der Siebträger-Espressomaschine ist ebenfalls die Abfolge Mischwasserregelung - Durchflussregelung - Fördermenge erforderlich.

Durch Knopfdruck bzw. durch Auslösung auf der MATLAB[®]-GUI für der Teewasserbezug. Es steht ein Druck-Dreh-Knopf für den Teewasserbezug zur Verfügung. Über die Drehung des Knopfes wird die Solltemperatur des Teewasserbezugs eingestellt was die Mischwassersolltemperatur $\vartheta_{M,soll}(t)$ ergibt. Die gewünschte Flowrate $\dot{V}_{K,soll}$ ist im System voreingestellt.

Der Teewasserbezug startet mit einem Vorsteuerwert für die Pumpenleistung und beginnt mit dem Einschwingen des Mischtemperaturreglers. Während der Einschwingphase des Mischtemperaturreglers wird das Bezugswasser in den Maschinenüberlauf geleitet. Ist die Mischwassertemperatur erreicht und der Mischwasserregler befindet sich im eingeschwungenen Zustand wird durch Schalten der zugehörigen Magnetventile auf Teewasserbezug umgeschaltet und es wird der Durchflussregler gestartet. Wenn die gewünschte Teewassermenge erreicht ist wird der Teewasserbezug durch abschalten der Pumpe und dem Ruhezustand der Magnetventile beendet. Die beteiligten Regelkreise werden abgeschaltet, die letzten vorhandenen Stellgrößen werden als Vorsteuerwerte für den nächsten Teewasserbezug gespeichert. Die Abschaltung erfolgt entweder durch zweiten Tastendruck oder durch Füllmengenberechnung.

5.11 Erstellung der Funktionssteuerung - Dampfbezug auf dem Basisboard

Der Dampfbezug wird durch Druck am Dampf-Druck-Drehknopf ausgelöst. Aktuell wird das entsprechende 2/3-Wege-Magnetventil auf Dampfbezug umgeschaltet. Geplant ist für den Dampfbezug eine Soft-Open-Funktionalität in Form eines schrittmotorbetätigten Kugelhahns. Der Kugelhan wird dem Auslösen auf den vorherigen Öffnungswert geöffnet. Über die Drehinkremente des Druck-Drehknopfes wird der Dampfahn mehr oder weniger geöffnet.

Ein Zweiter Druck beendet dem Dampfbezug.

5.12 Erstellung der Funktionssteuerung - Spülen auf dem Basisboard

Die Spülvorgänge sind angelegt, jedoch nur rudimentär getestet. Fehler im Programcode sind derzeit nicht auszuschließen und die Funktionalität ist noch nicht erprobt. Insbesondere die Wirksamkeit der Schaltung des Rückspülventils am Ende des Pumpvorganges ist derzeit noch nicht nachgewiesen.

In der MATLAB®-GUI ist die Funktionssteuerung „Flush / Reinigung“ eingeführt. Drücken der Taste „Flush / Reinigung“ wird die Funktionssteuerung gestartet. Die aktuelle Realisierung ist im Wiki unter Systemsoftware - ToDo-Liste MATLAB®-Funktionen und GUI im Arbeitspaket „Spülvorgänge Multi-MCU“ beschrieben.

Die Funktionssteuerung Spülen kennt 4 Funktionszustände. Jeder Wert größer 0 hat eine Bedeutung für die Reinigung der Maschine.

Drücken der Taste „Flush / Reinigung“ löst den Start der Maschinenreinigung aus. Dies kann ein Flush (kurzes Durchspülen der Brühgruppe) oder eine Rückspülreinigung sein. Für den Start der Reinigung ist dies jedoch unerheblich. Das Dosierventil wird auf 100 gesetzt, alternativ kann hier eine Solltemperatur gesetzt und der Mischregler aufgerufen werden. Es werden die erforderlichen Magnetventile geschaltet und das Rückspülventil wird in den Ruhezustand versetzt. Damit wird das Wasser bei eingeschalteter Pumpe mit $U_P = 2500 \text{ mV}$ in die Brühgruppe geleitet. Dieser Maschinenzustand wird 15 Sekunden beibehalten. Erst danach erfolgt weitere Funktionalität.

Ist nach 15 Sekunden Wasserpumpen der Brühgruppendruck größer 0,75 bar, dann ist aller Wahrscheinlichkeit nach ein Blindsieb eingesetzt und es soll eine Rückspülreinigung vorgenommen werden. Dazu wird $U_P = 0 \text{ mV}$ gesetzt und das Frischwasserventil geschlossen. Für den Druckabbau wird das Rückspülventil geöffnet. Es wird ein Statustext ausgegeben gesetzt.

Ist nach 15 Sekunden Wasserpumpen der Brühgruppendruck kleiner 0,75 bar, dann sollte aller Wahrscheinlichkeit nach eine Durchspülung der Brühgruppe (Flush) erfolgen. Das ist damit dann auch erfolgt und der Reinigungsvorgang wird beendet. Die vorher geschalteten Magnetventile werden wieder in den Ruhezustand und die Pumpenleistung auf 0 gesetzt.

Der Rückspülvorgang kennt zwei Zustände, pmpen und einwirken. Es wird jeweils 5 Sekunden gegen das Blindsieb gepumpt. Danach wird die Pumpe abgeschaltet und das Entschichtungsventil wieder geöffnet. Anschließend wird das Reinigungsmittel einwirken gelassen. Bis zum Ende des aus pumpen und einwirken bestehenden einzelnen Rückspülzyklus wird gewartet. Danach erfolgt wieder erneutes Anpumpen gegen das Blindsieb. Damit beginnt der Rückspülzyklus von neuem. Aktuell wird mit 10 Rückspülzyklen gearbeitet. Dieser Wert muss noch durch Erprobung bestätigt werden. Ist das Ende der Rückspülung erreicht. Es wird eine Statusmeldung ausgegeben. Die Maschine befindet sich jetzt in der Durchspülung. Sind noch nicht alle Rückspülzyklen durchgeführt, wird das Rückspülventil geschlossen und im Status eine Meldung ausgegeben wie viele Rückspülungen noch durchgeführt werden.

Zum Abschluss der Rückspülreinigung erfolgt eine Durchspülung zur Beseitigung der Reinigungsmittelreste. Nach aktuell 90 Sekunden wird die Durchspülung beendet. Es wird die Pumpensteuerspannung auf 0 und die Magnetventile in den Ruhezustand geschaltet. Mit der Ausgabe eines Statustextes wird die Reinigung beendet.

5.13 Erstellung der Funktionssteuerung - Grundreinigung auf dem Basisboard

Als Service-Funktion soll eine Grundreinigung der Maschine durchgeführt werden. Die Reinigung soll weitestgehend automatisiert erfolgen. Die Funktionalität ist unter [79, Abschnitt 7.16] beschrieben. Im Wesentlichen handelt es sich dabei um Entkalkungen des Boilers bzw. des Gesamtsystems.

Für eine Entkalkung muss zunächst der Boiler vollständig entleert werden. Sobald der Durchflusssensor keine Impulse zurück gibt ist der Boiler vollständig entleert (was bisher noch nicht nachgewiesen ist). Es wird danach ein Reinigungsmittel eingebracht und der Boiler wird bis ca. 95% seines Gesamtvolumens befüllt, was eine Überfüllung von 136458 Impulsen des Flowmeters entspricht. Der Boiler wird durch die Heizwendel auf eine noch zu definierende Temperatur mit Entschichtung erhitzt.

Für die Entkalkung von Brühgruppe und Teelanze wird die bereits erwärmte und für die Boilerentkalkung verwendete Reinigungsflüssigkeit durch die Wasserwendel und das Dosierventil weiter in die Brühgruppe gepumpt. Die Brühgruppe sollte mit einem Blindsieb verschlossen sein, so dass das Wasser über die Rückspüleleitung in den Überlauf weiter geleitet wird. Durch Umschaltung von Kaffeebezug auf Teewasserbezug wird auch die Teewasserlanze gereinigt.

Am Ende wird das komplette System mit Frischwasser in gleicher Weise wie bei der Reinigung durchgespült.

5.14 Integration eines Displays mit Touchfunktion

Ein Display mit Touchfunktion wird zur Visualisierung und als Tastenersatz angestrebt. Das Display enthält entweder selbst eine MCU oder ist an einer weiteren MCU angeschlossen. Bei einer weiteren MCU bestünde die Möglichkeit eine WLAN-Funktionalität einzuführen. Ein lokaler Webserver auf der Maschine führt zu verbesserter Konnektivität sowie zur Erleichterung von Änderungen der Tastenbelegungen und der Profilverwaltung.

Die Display-MCU wertet den Token aus und verändert diesen im Falle von Tastenaktivität. Ggf. würde die Display-MCU den Spannungswert des Vertikalhebels erfassen.

5.15 Integration eines Vertikalhebels

Es ist die Integration eines Vertikalhebels in Vorbereitung. Dieser wird über ein Drehpotentiometer ein Spannungssignal proportional zu der Hebelstellung generieren. Es sind 4 Lagen und ein variabler Bereich definiert. In Ruhestellung befindet sich der Vertikalhebel im oberen Anschlag und das Signal hat den Spannungswert von $U_{VH,oben}$. Zwischen dem oberen Anschlag und dem unteren Anschlag kann der Spannungswert sich aufgrund der Hebelbewegung variabel verändern ($U_{VH}(t)$). Das Signal wird für die Simulation der Hebelbewegung einer Handhebel-Espressomaschine verwendet. Beide Endlagen des Vertikalhebels können überdrückt werden, was die Spannungswerte $U_{VH,ue,oben}$ und $U_{VH,ue,unten}$ ergibt. Die Überdrückungsstellungen werden als zwei weitere Tasten interpretiert und als Tastenschaltung in die Token aufgenommen.

5.16 Visualisierung und Interaktion über die MATLAB®-GUI

In der Übergangsphase soll der Betriebsstart der Espressomaschine von der MATLAB®-GUI ausgelöst werden.

Die aktuelle MATLAB®-GUI soll in eine komfortable Bedienoberfläche für Wartungs- und Einstellarbeiten überführt werden. Alle Tastenfunktionalitäten und die Vertikalhebel-funktionalität sollen über die Oberfläche durchgeführt werden können. Wie bisher soll jeder Aktor (Magnetventile, Pumpenleistung und Schrittmotorverstellungen) über die GUI ausgelöst werden können. Ebenso soll die bisherige Visualisierung der Messwerte und Schaltzustände erhalten bleiben. Dies allerdings mit einer geringeren Aktualisierungsrate. Vier Aktualisierungen je Sekunde werden als ausreichend angesehen.

6 Literatur und Quellen

Bei den Quellenreferenzierungen in diesem Dokument handelt es sich um Referenzierungen des allgemeinen Literaturverzeichnisses, welches unter <http://www.ifk.bayern> öffentlich zugänglich ist.