Hochschule München University of Applied Sciences

Fakultät für Elektrotechnik und Informationstechnik

Bachelorstudiengang Elektrotechnik- Elektromobilität (EMB)

Bachelorarbeit von Noureddine Ait Ouhamou

Umbau der Systemelektronik der Kaffeemaschine und Anpassung der Regelung

Reconstruction of the system electronics of the coffee machine and adjustment of the control

Betreuerin/Betreuer an der Hochschule: Dr. Klemens Graf

Betreuerin/Betreuer in der Firma: Armin Rohnen, Fakultät 03

Bearbeitungsbeginn: 24. April 2023

Abgabetermin: 24. Oktober 2023

lfd.Nr.: 2363

Hochschule München
Fakultät für Elektrotechnik und Informationstechnik

Hochschule München University of Applied Sciences

Erklärung der Bearbeiterin bzw. Erklärung des Bearbeiters:

Ait Ouhamou	Noureddine
Name	Vorname

Ich erkläre hiermit, dass ich die vorliegende Bachelorarbeit selbständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe.

Sämtliche benutzte Quellen und Hilfsmittel sind angegeben. Wörtliche und sinngemäße Zitate sind als solche gekennzeichnet.

München, 10.10.2023	
Ort, Datum	Unterschrift

Kurzfassung/Abstract

In diesem Bericht werden im Rahmen einer Bachelorarbeit die Fortschritte und Ergebnisse an der labortechnischen Espressomaschine, die im Semester SoSe2023 erreicht werden, beschrieben. Damit wird gewährleistet, dass nachfolgenden Projektgruppen bzw. Studenten die Einarbeitung erleichtert wird. Der Schwerpunkt dieser Arbeit liegt an der Hardware und deren Programmierung sowie die Weiterentwicklung der Regler. Dazu wird der Aufbau der Systemelektronik genauer erläutert, um das Vorgehen in der Programmierung besser erklären zu können. Die Durchfluss-, Mischtemperatur-, Füllstands- und Heizregler sind schon von einem vorherigen Bacheloranten implementiert worden. Diese gilt es zu verbessern, da die Regler zu langsam reagieren. Das Ziel der Bachelorarbeit ist, dass an der Kaffeemaschine mit Hilfe der neuen Systemelektronik und den überarbeiteten Reglern Laborversuche besser durchgeführt werden können. Im Vordergrund dieses Berichts steht die Erarbeitung und das Vorgehen zu diesen Zielen, die in der Bearbeitungszeit erreicht werden.

This report describes the progress and results achieved on the laboratory espresso machine in the semester SoSe2023 as part of a Bachelor's thesis. This ensures that subsequent project groups or students will find it easier to familiarise themselves with the machine. The focus of this work is on the hardware and its programming as well as the further development of the controllers. For this purpose, the structure of the system electronics is explained in more detail in order to better explain the programming procedure. The flow, mixing temperature, level and heating controllers have already been implemented by a previous Bachelorant. These need to be improved because the controllers react too slowly. The aim of the bachelor's thesis is to make it easier to carry out laboratory tests on the coffee machine with the help of the new system electronics and the revised controllers. The focus of this report is on the development and procedure for these goals, which will be achieved in the processing time.

Abkürzungsverzeichnis

Abkürzung	Beschreibung
MATLAB-GUI	MATLAB Graphic User Interface
STM32	STM32F411RE Microcontroller
SSR	Solid State Relais (Schalter)
IC	Integrated Circuit
NTC	Negative Temperature Coefficient Thermistor
LED	Light-emitting Diode
ADC	Analog Digital Converter
DAC	Digital Analog Converter
PWM	Pulsweitenmodulation
I2C	Inter-Integrated Curcuit

Formelverzeichnis

Formelzeichen	Einheit	Beschreibung
R	Ohm	Widerstand
U	V	Spannung
I	A	Strom
R31	Ohm	Widerstand 31
C19	uF	Kondensator 19
switch	1	Schalter eines Magnetventils
chan_list	1	Liste, in der die Messwerte abgespeichert werden
ch	1	Index
K_p	1	Verstärkungsfaktor P-Teil eines PID-Reglers
K_i	1	Verstärkungsfaktor I-Teil eines PID-Reglers
K_d	1	Verstärkungsfaktor D-Teil eines PID-Reglers
K_{pkrit}	1	Kritischer Verstärkungsfaktor P-Teil eines PID-Reglers

Inhaltsverzeichnis

1. Einleitung

Mit Hilfe von Projekten und Kooperationen werden derzeit im Labor der Hochschule München unter Leitung von Herrn Rohnen verschiedene Kaffeemaschinen entwickelt. Die labortechnische Espressomaschine, an der die Arbeiten für die Bachelorarbeit durchgeführt werden, ist eine von ihnen. Das Ziel der Laborarbeiten ist, dass die unterschiedlichen Maschinen, die es zurzeit auf dem Markt erhältlich sind, zu reproduzieren, um Verbesserungen zu erarbeiten.

Die Labormaschine befindet sich zum Anfang dieses Projekts in einem einsatzfähigen Zustand. Mit der STM32 (eine MCU) werden die Aktoren betrieben und die Sensoren ausgelesen. Die graphische Oberfläche, um die Maschine zu bedienen, wurde in MATLAB® APPDESIGNER realisiert. Durch das Programm wird ermöglicht die Magnetventile, die Pumpe und die Heizung im Boiler zu steuern. Dies wird ermöglicht durch verschiedene Bedienelemente in der App (MATLAB®-GUI).

Da sich die STM32F411RE jeweils nach 15 Minuten in den Ruhemodus versetzt hat, ist die Entscheidung getroffen worden, die Systemelektronik neu aufzusetzen. Ein Ziel dieser Arbeit ist, ein Umbau der Kaffeemaschine an der elektrischen Hardwarekomponente zu vollziehen und die Funktionen der STM32 an die neue Multi-MCU bestehend aus drei einzelnen Platinen mit Hilfe der Programmiersprache MicroPython anzupassen. Dazu sind die jetzigen implementierten Durchfluss-, Mischtemperatur-, Füllstands- und Heizregler in der MATLAB®-GUI sehr träge und sollen verbessert werden. Dabei müssen die Parameter der Regler angepasst oder eventuell ein anderer Ansatz gewählt werden. Dabei wird in dieser Arbeit versucht, einen Überblick zum Vorgehen und zu den Lösungen zu schaffen, damit ein nachfolgender Student oder eine nachfolgende Projektgruppe mit Hilfe dieses Berichts die zu erledigenden Aufgaben bewältigen kann. Dennoch wird so gut wie möglich ins Detail gegangen, um mögliche Komplikationen zu vermeiden. Zudem werden Lösungsideen zu Problemen, die nicht in dieser Bachelorarbeit erledigt werden, erklärt und präsentiert. Die erarbeiteten Ergebnisse sind im Nachhinein auf die anderen Espressomaschinen anwendbar, die mit der gleichen Systemlogik betrieben werden.

2. Stand der Entwicklung

Am Anfang der Bachelorarbeit befindet sich die labortechnische Espressomaschine bereits in einem fortgeschrittenen Zustand. Die nötigen Ventile, die Pumpe und die Sensoren sind jeweils schon verbaut. Die Maschine wird zu Beginn mit einer Hauptplatine (STM32F411RE) gesteuert, die mit einer Erweiterungsplatine verbunden ist, da die STM32 nicht genug Anschlüsse bereitstellt, um die Espressomaschine zu bedienen. Dazu sind auf der Erweiterungsplatine Messkarten aufgesteckt, um die Sensoren auszuwerten [85]. An der STM32 gibt es ein ungelöstes Problem: Der Microcontroller legt sich nach 15 Minuten schlafen und reagiert nicht mehr. Um dies zu umgehen, muss die Stromzufuhr komplett unterbrochen werden, damit beim Anschalten die STM32 wieder ansprechbar ist. Daher werden die Aufgaben der STM32 auf drei Platinen aufgeteilt.

Die Software ist in einem funktionierenden Zustand. Die Grundfunktionen für die STM32 wurden mit Hilfe der zusammengefassten Aktoren- und Sensorenansteuerung [41] realisiert. Diese Funktionen werden für die neue Systemelektronik entwickelt und angepasst, da sich die Aktorik und Sensorik der Maschine nicht ändert und beide mit der Programmiersprache MicroPython realisiert werden. Die labortechnische Espressomaschine wird durch die MATLAB®-GUI bedient. Das Interface wurde im MATLAB® APPDESIGNER entwickelt. Es liegt bereits eine Bachelorarbeit vor, in der die Schnittstelle zwischen der STM32 und der MATLAB®-GUI beschrieben und erarbeitet wurde [85]. Momentan werden die Ventile durch die graphische Oberfläche geöffnet und geschlossen, die Werte der Sensorik ausgelesen und separat in einer Datei gespeichert. Die Anpassung der MATLAB®-GUI an den neuen Hydraulikplan und die Kommunikation zu den Platinen wird durch eine Projektgruppe durchgeführt und von Herrn Armin Rohnen finalisiert.

Das Grundkonzept der Regelung für die Espressomaschine ist ebenfalls in der MATLAB®-GUI bereits implementiert, aber noch nicht ausreihend getestet worden [85]. Diese sind im Moment noch ziemlich träge. Deshalb werden die Regler verbessert. Dabei werden die passenden Verstärkungsfaktoren herausgearbeitet oder möglicherweise ein anderer Regelansatz verwendet. Mit Hilfe der neuen Regler wird ein automatisierter Kaffeebezug und Teebezug an der labortechnischen Espressomaschine möglich sein.

3. Aufbau der Systemelektronik

Die neue Systemelektronik besteht aus drei Platinen: SSR-Platine, Messplatine und Basisplatine. Die Aufgaben der STM32 werden demnach aufgeteilt. Dadurch arbeiten die Platinen unabhängig voneinander und werden von der MATLAB®-GUI dementsprechend vernetzt. Es werden auf den Platinen nicht alle Anschlüsse verwendet, da die labortechnische Espressomaschine nicht über mehr Komponenten verfügt. Die Platinen sind für die andere Kaffeemaschinen, die im Labor entwickelt werden, anwendbar.

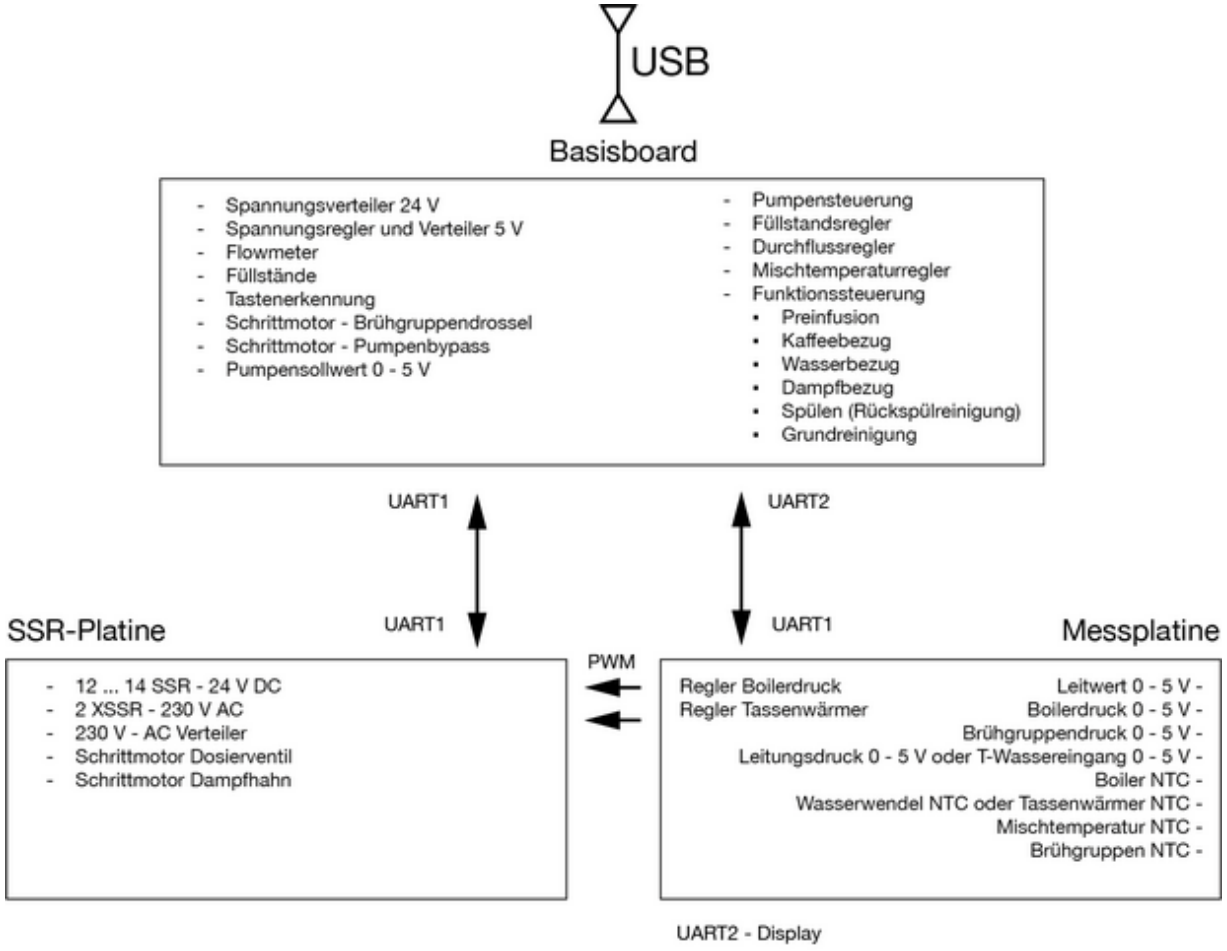


Abbildung 1: Aufbau der Systemelektronik [x]

In Abbildung 1 sieht man zusammengefasst welche Aufgaben jede Platine zu erfüllen hat. Dabei ist zu erkennen, dass die Platinen bzw. die Microcontroller in Zukunft untereinander kommunizieren und miteinander verbunden werden. Der Aufbau und die Funktion der einzelnen Komponenten wird in diesem Kapitel beschrieben.

3.1 SSR-Platine

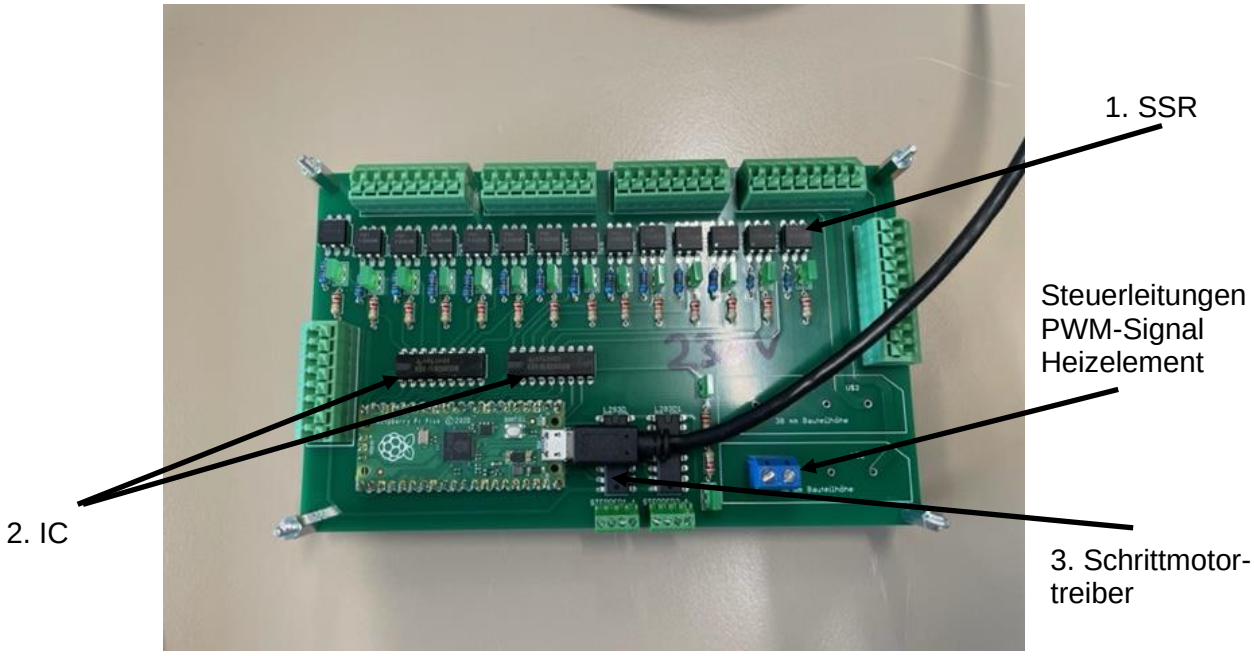


Abbildung 2: SSR-Platine

In der Abbildung 2 wird die SSR-Platine gezeigt:

1. SSR (VOR1121A6)
2. IC (ULN2803A)
3. Schrittmotortreiber (L293D)

Der Name der Platine kommt von den Solid State Relais (SSR). Diese Komponenten werden für das Schalten der Magnetventile verwendet. Dabei sind die Schalter normally open. Das heißt, dass die SSRs im unbestromten Zustand geöffnet sind und somit den Magnetventilen keine Spannung zur Verfügung steht [i], damit sie sich je nach Magnetventilart öffnen oder verstellen lassen. Parallel dazu sind LEDs verbaut, um eine visuelle Möglichkeit zu haben, die geschlossen Schalter zu sehen. Die Vorwiderstände werden mit dieser Rechnung dimensioniert.

$$R = (U_{Quelle} - U_{Komponente}) / I_{Komponente}$$

Dabei hat sich für den Vorwiderstand des Relais (SSR-Relais) ein Wert von 2,2 kOhm bei einer Spannung von 1,5 V und einem Strom von 10 mA am Relais ergeben. Für die LEDs errechnet sich ein Vorwiderstand von 1,075 kOhm bei einer Spannung von 2,5 V und einem Strom von 20 mA. Da dieser Widerstand der LEDs als Bauteil nicht verfügbar ist, wird ein Widerstand von 1,2 kOhm verbaut. Es ist besser einen etwas höheren Widerstand zu wählen, wenn der ausgerechnete nicht exakt zur

Verfügung steht, weil die Komponente meistens trotzdem funktioniert. Mit einem etwas kleineren Widerstand kann die Spannung oder der Strom zu hoch sein und das Bauteil beschädigen. Die Schaltung der Relais wird mit einer Steuerspannung von 24 V betrieben. Die benötigte Spannung der Komponenten und der Strom wird aus den Datenblättern entnommen.
Des Weiteren sind zwei ULN2803A ICs zum Schalten der Ventile und zum Heizen notwendig, um den Stromkreis der Relais zu schließen. In diesen ICs sind achtmal zwei Transistoren miteinander verschaltet. Dadurch werden große Lastströme mit Hilfe kleinen Strömen, die zum Beispiel von Microcontrollern erzeugt werden, geschalten. Damit ist es möglich die Schalter der Magnetventile zu verwenden, da diese unbelastet sein sollen. Somit treten keine unvorhergesehenen Schaltereignisse auf [99]. Auf diese Art und Weise werden die Ventile und das Heizelement gesteuert.

Die letzten wichtigen Komponenten auf dieser Platine sind die Schrittmotortreiber. Da der Schrittmotor für den Dampfhahn bisher nicht fertig verbaut ist, wird nur der Schrittmotor für das Dosierventil zum Mischen des Heiß- und Kaltwassers gesteuert. Wie die Treiber und die Motoren angeschlossen werden müssen, entnimmt man aus Anhang 1. Um den Motor zu steuern, werden die Spulen in einer bestimmten Sequenz bestromt. Dadurch bildet sich ein magnetischer Fluss aus und der Rotor dreht sich immer zum größtmöglichen Fluss. Damit legt der Motor jedes Mal den gleichen Winkel zurück [100]. Somit ist die Temperatur an der Brühgruppe gut einstellbar.

3.2 Messplatine

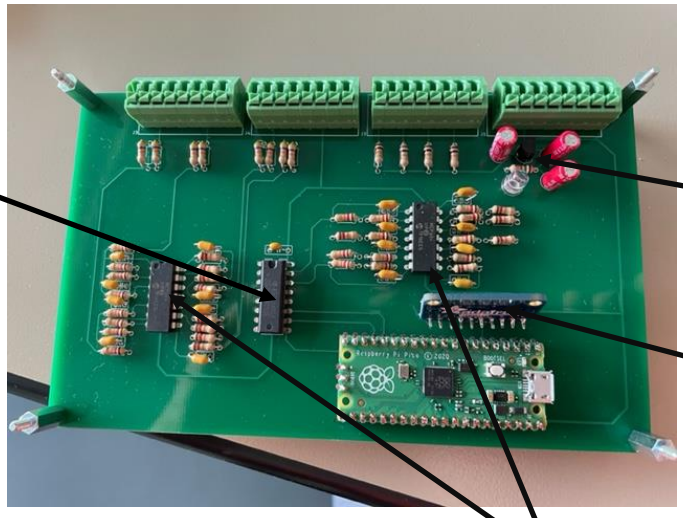


Abbildung 3: Messplatine

In Abbildung 3 sieht man den Aufbau der Messplatine:

1. ADC (ADS1115)
2. Operationsverstärker (MCP604P)
3. 8 zu 1 Multiplexer (4051N)
4. Präzisionsspannungsregler (MC1541TO)

Hierbei werden 8 analoge Eingänge mit Hilfe eines ADCs ausgewertet und an die MATLAB®-GUI übertragen, um mit diesen Werten die Regelung an der labortechnischen Espressomaschine durchzuführen. Angeschlossen werden drei NTCs, drei Drucksensoren und ein Temperatursensor. Die Messkanäle haben einen 15 Hz Tiefpassfilter, mit dem das Messrauschen minimiert wird. Die NTC-Eingänge sind mit einem 4096 mV Spannungsregler verbunden. Aus Anhang 2 ist zu erkennen, dass die analogen Messeingänge durch Spannungsteiler halbiert und mit Hilfe der Operationsverstärker mit Tiefpassfilter wieder mit dem Verstärkungsfaktor 2 verstärkt werden, um die Spannung in einen Bereich von 0 bis 5 V zu bringen. Der Multiplexer schaltet die acht analogen Eingänge durch und somit werden die Messwerte nacheinander einzeln am ADC eingelesen. Der Analog-Digital-Converter wird im kontinuierlichen Modus betrieben. Deshalb schickt der Interrupt Pin vom ADC ein Signal an den Microcontroller, mit dem die Auswertung des Messsignals beginnt [ii]. Das Ziel dabei ist, dass mindestens zehn Messwerte pro Kanal die Sekunde an die MATLAB®-GUI gesendet werden.

3.3 Basisplatine

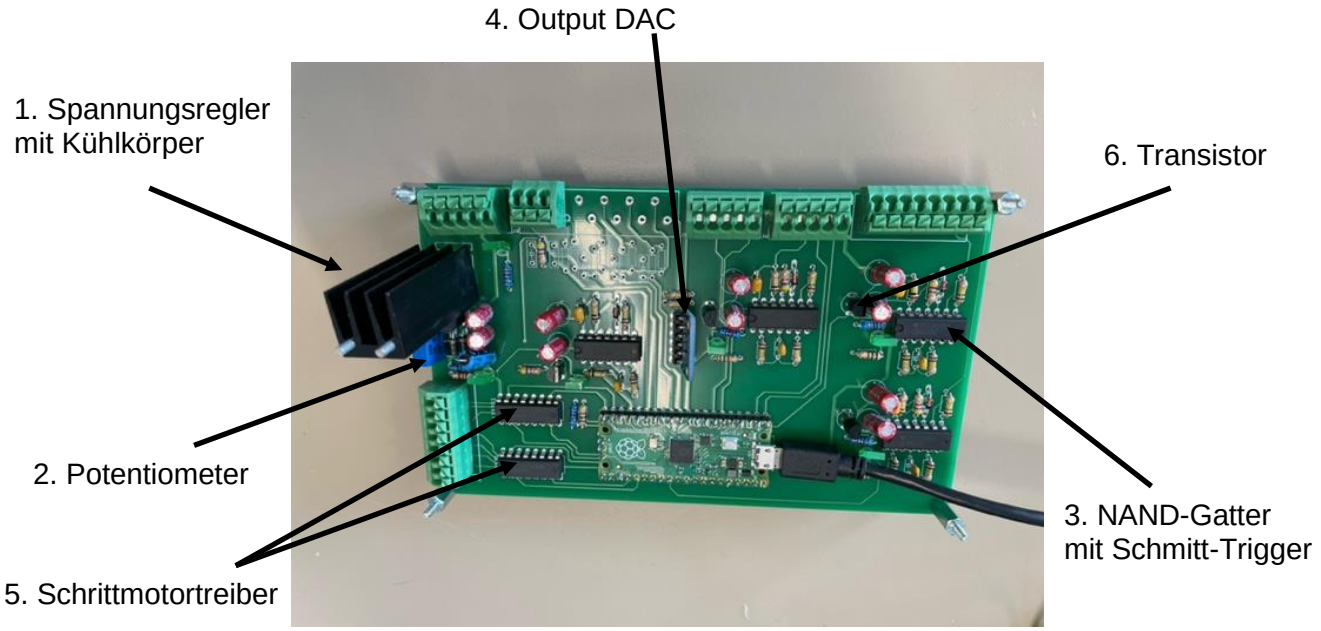


Abbildung 4: Basisplatine

Dies ist der Aufbau der letzten Platine der neuen Systemelektronik:

1. Spannungsregler (LM338)
2. Potentiometer
3. NAND-Gatter mit Schmitt-Trigger (4093N)
4. Output DAC (MCP4725)
5. Schrittmotortreiber (L293D)
6. Transistor (BC547)

Mit dieser Platine werden mehrere Funktionen zugleich abgedeckt. Die Getriebepumpe wird betrieben, der Füllstand wird überwacht, Schrittmotoren werden gesteuert, der Durchfluss wird gemessen und die 5 V Spannung für die Basisplatine und die anderen Platinen wird bereitgestellt.

Durch den Spannungsregler und den Potentiometer wird die 24 V Quellenspannung auf 5 V herunter gesetzt, indem der Potentiometer mit dem Widerstand R31 in der Schaltung (siehe Anhang 3) die Spannung am Ausgang vom LM338 festlegt. Mit den Dioden werden mögliche Spannungsspitzen, die potentiell am Ausgang des Spannungsreglers entstehen können, abgefangen und der Stromkreis wird geschützt. Die Spannung kann entweder am Kondensator C19 oder an der Versorgungspannung eines ICs zu Ground gemessen werden [101].

Die Pumpe braucht die 24V Quellenspannung zur Versorgung. Die Steuerung wird mit dem DAC (Digital Analog Converter) getätigt. Prinzipiell ist diese Komponente die Umkehrung eines ADC. Dabei wird vom Microcontroller eine 12-Bit Nachricht über den verbundenen I2C-Bus (eine Kommunikationsverbindung zwischen Microcontroller und DAC) gesendet. Dadurch sendet der MCP4725 den richtigen Spannungswert über den V_{out}-Pin bzw. Pumpensollwert-Pin (siehe Anhang 3) an die Getriebepumpe. Die Steuerspannung für den Betrieb darf zwischen 0 V und 5 V betragen. Dazu entspricht 1 mV vom DAC 1 Umdrehung/min. Zu beachten ist, dass die Pumpe erst ab einem Wert von 300mV anfängt sich zu drehen. Das heißt, die minimale Drehzahl beträgt 300 U/min [65, 85].

Die Überwachung des Füllstandes wird mit Hilfe der NAND-Gatter durchgeführt. Dabei wird vom Füllstandssensor ein Kurzschluss erzeugt, wenn der Boiler ausreichend mit Wasser befüllt worden ist. Die Funktion der Gatter sind negierte bzw. invertierte AND-Gatter:

Signal 1	Signal 2	Ausgangssignal
0	0	1
1	0	1
0	1	1
1	1	0

Tabelle 1: Signalverhalten NAND-Gatter [xv]

In Anhang 4 sieht man den Schaltplan zur Erfassung der Füllstandsdetektion. Sobald ein Kurzschluss durch den Füllstandssensor entsteht, gibt das NAND-Gatter durch das Signal vom Füllstand (Signal 2 gleich 1) und dem Signal einer vorher angelegten Spannung (Signal 1 gleich 1), keine Spannung an den Transistor weiter. Da dieser Transistor keinen Basisstrom mehr gespeist bekommt, wird der Strom, der durch die LED fließen würde, geblockt und die LED erlischt. Stattdessen fließt der Strom zum vorhergesehenen Pin am Microcontroller. Um zu verhindern, dass die NAND-Gatter bei jeder kleinen Spannungsanhebung am Eingang ihr Ausgangssignal ändern, haben diese Schmitt-Trigger verbaut. Somit ändern sich die Signale erst, wenn eine bestimmte Schwellenspannung unter oder überschritten wird [105].

Für die Messung des Durchflusses, wird ein Flowmeter verwendet. Der zeitliche Abstand wird von Impulsen am Pin des Raspberry Picos gemessen. Dabei werden pro ml Durchfluss 39,9 Impulse erzeugt [69]. Die Inbetriebnahme ist in [85] beschrieben.
Auf dieser Platine sind dieselben Schrittmotortreiber verbaut wie auf der SSR-Platine und funktionieren auf dieselbe Art und Weise.

3.4 Sensorik

Für den Leitungsdruck und den Boilerdruck sind neue Drucksensoren in die labortechnische Espressomaschine verbaut worden. Der Leitungsdrucksensor ermittelt einen Druck von 0 bis 16 bar und der Boilerdrucksensor von 0 bis 4 bar. Da die Umrechnungskurven im Datenblatt keine logischen Werte erbracht hatten, wurden die Sensoren selbst nachgemessen. Dabei wird Druck mit Hilfe einer Luftpumpe in einem geschlossenen System erzeugt und die Spannung des Sensors und eine Referenzspannung eines Referenzdrucksensors gemessen. Die zwei zu messenden Sensoren haben einen Offset von 0,5 V bei 0 bar. Dieser Offset ist von den gemessenen Spannungen abgezogen

worden. Der Referenzdrucksensor misst zwischen 0 und 40 bar in einem Spannungsbereich von 0 bis 10 V. Dazu ergeben sich diese Werte:

Leitungsdruck in V	Referenzdruck in V	Referenzdruck in bar
0,1	0	0
0,75	0,53	2,12
0,96	0,72	2,88
1,2	0,89	3,56
1,9	1,52	6,08

Tabelle 2: Leitungsdruckkalibrierung

Boilerdruck in V	Referenzdruck in V	Referenzdruck in bar
0,1	0	0
2,6	0,5	2
3,4	0,68	2,72

Tabelle 3: Boilerdruckkalibrierung

Mit diesen Werten wird eine Gerade extrapoliert, um die Geradengleichung für die Sensoren aufzustellen und für die Umrechnung in MATLAB® zu übernehmen. (siehe Anhang 5 und 6).

3.5 Troubleshooting

Beim Aufbau der Basisplatine ist die erste Platine wegen eines defekten Potentiometers durchgebrannt. In Abbildung 5 sieht man einen prinzipiellen Aufbau.

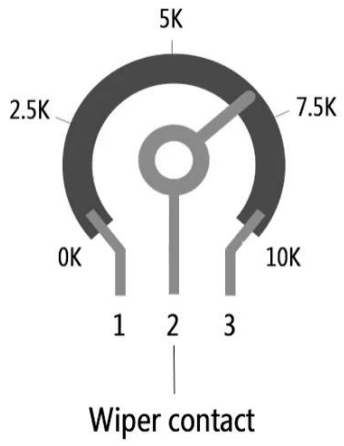


Abbildung 5: Aufbau Potentiometer [xi]

Die Quellenspannung von 24 V ist an den Kontakten 1 und 2 angeschlossen. Durch das Drehen der Schraube sollte sich die Kontaktstelle bzw. der Schleifpunkt, der mit dem Widerstandsmaterial verbunden ist, verschieben. Diese Stelle hat sich an der defekten Komponente nicht geändert, da die eingestellte Spannung von 5,1 V gleich geblieben ist. Nachdem weiter an der Stellschraube gedreht wurde, sind die LEDs erloschen und die angelegte Quellenspannung von 24 V ist auf 2,3 V eingebrochen. Im Anschluss wurde untersucht, wodurch die Spannung einbricht. Bei der Widerstandsmessung ist festgestellt worden, dass der Potentiometer einen Widerstand von 0 Ohm am Eingang 1 und Ausgang 2 hat. Es wird vermutet, dass beim Weiterdrehen der Stellschraube diese überdreht wurde und die Kontaktstelle zu 0 Ohm sich verbunden hat.

Daher wird vor dem Einbau erst das Potentiometer nachgemessen, ob sich der Widerstand beim Verstellen ändert, um in Zukunft dieses Ereignis zu verhindern.

Des Weiteren ist beim Entwerfen der Messplatine vergessen worden, eine Verbindung des ALERT/RDY Pin vom ADC zum Microcontroller einzubauen. Dieses Problem wird durch das Löten einer Verbindung mit einem Draht behoben.

Somit sind alle Platinen betriebsbereit und funktionsfähig. Ein Anschlussplan für die Platinen ist erstellt worden. Im Anschluss werden die Microcontroller der Platinen programmiert, damit die Sensoren ausgelesen werden und die Aktoren gestellt werden können, um die labortechnische Espressomaschine in Betrieb zunehmen.

4. Hardwarenahe Programmierung der Systemelektronik

Um die Platinen mit den Microcontrollern zu steuern, wird der benötigte Programmcode in ihrer Funktionalität und ihrer Benutzung erklärt. Dabei muss beachtet werden, dass die MATLAB®-GUI mit dem enthaltenen Programm agieren kann. Prinzipiell kommuniziert MicroPython mit MATLAB®, indem Zeichenketten versendet werden. Genauer ist die Kommunikation in [40, 41, 85] beschrieben. Bei einer Neuentwicklung der Systemelektronik, verändern sich möglicherweise die Pinbelegungen.

4.1 Magnetventilsteuerung

Zuerst werden die Pins, die einen Schalter bedienen, initialisiert. Dabei wird der Pin auf diese Weise festgelegt:
switch1 = Pin(Pinnummer, Pin.OUT) und switch1.value(Wert)

Das heißt, der Pin mit der Pinnummer wird als Ausgang am Microcontroller festgelegt. Somit kann der Pin ein elektrische Signal an die Darlingtonschaltung schicken, die den Schalter aktiviert, oder inaktiv bleibt. Mit dem Zuweisen der Werte 0 und 1 erfolgt dies. Bei der Initialisierung sind die Magnetventile unbestromt. Somit wird jedem switch den Wert 0 zugewiesen.
Um im Betrieb die Ventile einstellen zu können sind zwei Funktionen implementiert worden, mit denen die Magnetventile über den .value(Wert) Befehl über 0 und 1 entsprechend gestellt werden können. Auf diese Weise wurden die Schaltung für die Ventile getestet.

Für die Kommunikation der MATLAB®-GUI werden die Schalter zum Ventile schalten bzw. die Pins in einen Vektor gespeichert und initialisiert. Durch das Zugreifen von den Index des Vektors durch "ventile[index].value(1)" wird der richtige Schalter bestromt und das Ventil ändert die Stellung [iii].

Die Ventilbelegung lautet wie folgt:
switch1 = Y01, switch2 = Y02, switch4 = Y04, switch6 = Y06, switch7 = Y07, switch8 = Y08, switch9 = Y09, switch11 = Y03, switch13 = Y13.

4.2 Schrittmotorsteuerung

Im Moment werden drei Schrittmotoren angesteuert: Das Dosierventil, die Brühgruppendrossel und der Pumpenbypass. Die Programmierung der Schrittmotoren sind in [40] und in [iv] ausführlich beschrieben. Für die Initialisierung der Schrittmotoren sind folgende Schritte zu beachten. Als Erstes werden die Pins in einer Variable hinterlegt. Anschließend werden die Pins des gewählten Schrittmotors initialisiert und die Sequenzposition -1 gewählt, da damit gewährleistet wird, dass der Schrittmotor mit der ersten Sequenz beginnt. Zuletzt wird die Schrittmotorsequenz aufgerufen. Da nach

der letzten Anwendung die Position des Schrittmotors nicht erkennbar ist, wird in der Initialisierung das Ventil komplett geschlossen und auf die gewünschte Position gefahren. Mit den Funktionen forwardStep() und backwardStep() verschließen oder öffnen sich die Schrittmotoren um die eingegebenen Halbschritte und ein Endtext wird übergeben, damit bei der Kommunikation mit MATLAB® eine Kennung übergeben wird [v].

4.3 Heizelementsteuerung

Das Heizelement wird mit einem PWM-Signal gesteuert. Ein PWM-Signal ist eine periodische Abfolge von An- und Aussignalen zwischen 0 und der Maximalspannung. Dies ermöglicht dem Heizelement jede mögliche effektive Spannung einzustellen. Je länger in einer Periode die maximale Spannung anlegt ist, desto höher ist die daraus resultierende Spannung. Dieses Verhältnis zwischen Einschaltzeit und Periodendauer ist der Tastgrad. Mit Hilfe des Tastgrads wird die effektive angelegte Spannung errechnet. [106].

Bei der Initialisierung des Signals wird der Pin 3 als Ausgang definiert und als PWM mit Variablenname = PWM(Pin(3, Pin.OUT) deklariert. Zum Einstellen der Parameter wird für die Frequenz mit Variablenname.freq(Wert) angelegt und für den Tastgrad mit Variablenname.duty_u16(Wert) angelegt. Die Frequenz muss mindestens 8 Hz betragen und der Tastgrad hat eine Auflösung von 16-Bit. Das bedeutet, dass für einen Tastgrad von 100% ein Wert von 65535 eingeben werden muss. Initialisiert wird mit einem Tastgrad von 0, damit das Heizelement nicht anfängt zu heizen. Die Steuerung erfolgt über MATLAB®, in dem die erwünschte Leistung des PWM-Signals mit dem gleichen Befehl gesendet wird.

4.4 Messdatenerfassung Messplatine

Messdaten werden mit Hilfe des ADCs und dem Multiplexer erfasst. Um den ADC zu bedienen, wird eine Klassenbibliothek von Robert Hammelrath [102] und die I2C aus der machine-Bibliothek des Raspberry Picos verwendet. Dabei wird zuerst an den Pins 4 und 5 eine I2C-Bus Schnittstelle mit einer Frequenz von 400 kHz angelegt. Pin 4 ist für die Daten verantwortlich und Pin 5 gibt den Takt weiter. Sobald ein Messwert vom ADC erstellt wurde, schickt der ALERT-Pin des ADCs ein Signal an den Microcontroller.

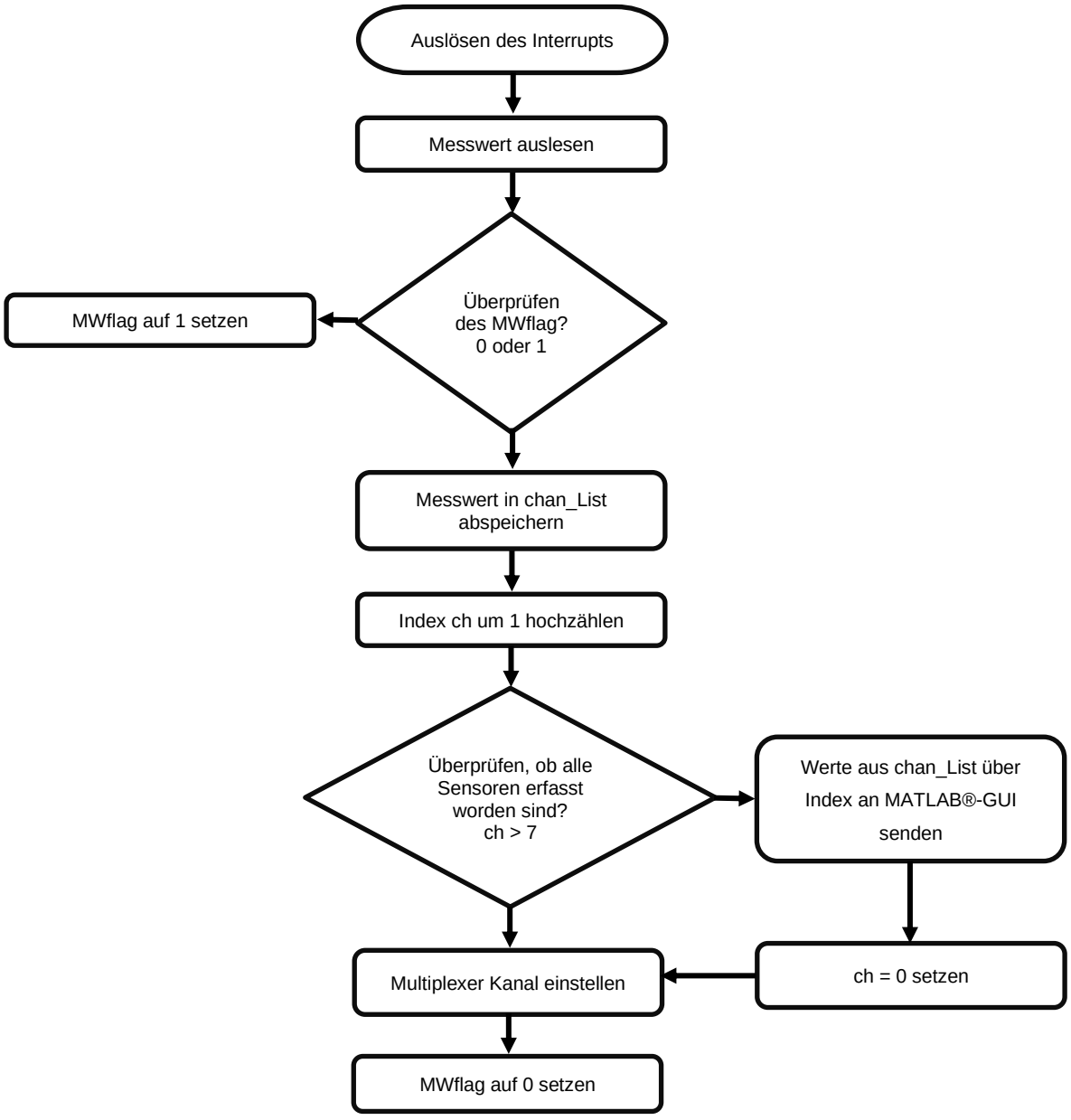
```
irq_pin = Pin(7, Pin.IN, Pin.PULL_UP)
irq_pin.irq(trigger = Pin.IRQ_FALLING, handler = sample)
```

In der ersten Codezeile wird der Pin 7 am Raspberry Pico als Eingangspin und als Pull-Up-Widerstand initialisiert. Durch diese Einstellung wird der Pin 7 beim erhalten des Signals auf ein HIGH-Potential gezogen. [107]. Anschließend wird definiert, wie der Interrupt ausgelöst wird und welche Funktion dazu

abgearbeitet wird. In diesem Fall löst der Interrupt bei einer fallenden Flanke aus. Also registriert der Microcontroller den Spannungsabfall vom geschickten Impuls des ADCs. Dadurch fängt der Pico an, den handler zu bearbeiten bzw. die zugewiesene Funktion auszuführen.

Die Pins zum Schalten des Multiplexers werden wie in der Ventilsteuerung initialisiert. Für die richtigen Schalterstellungen wird ein Vektor angelegt, in dem die richtigen Werte zum Ansteuern der Pins hinterlegt sind. Diese werden an die Pins adresse0, adresse1 und adresse2 übergeben. Dadurch wird mit den richtigen Indices der entsprechende Kanal eingestellt.

Der Ablauf der Messdatenerfassung wird an Hand des Flussdiagramms erklärt.



Zuerst wird der Interrupt ausgelöst. Somit beginnt die Auswertung an einem der verbauten Messsensors. Dieses Signal wird aus einem analogen Spannungswert zu einer 16-Bit digitalen Zahl umgewandelt. Um die Zahl zu verstehen, wird diese umgerechnet in einen Spannungswert von 0 bis 5000 mV. Danach wird überprüft, ob das Messwertflag (MWflag) schon gesetzt ist. Wenn der MWflag gleich 0 ist, wird der Messwert verworfen und das Flag wird auf 1 gesetzt. Somit wird sichergestellt, dass ein sinnvoller Messwert ausgewertet wird. Wenn der MWflag auf 1 gesetzt ist, wird Wert für die MATLAB®-GUI angepasst und in die vorangelegte chan_List eingespeichert. Dazu wird die Laufvariable ch um 1 erhöht, damit der nächste Kanal eingestellt werden kann und der kommende Messwert in die chan_List auf den nächsten Index eingetragen wird. Solange nicht alle acht Kanäle durchlaufen sind, wird jeder 2. Messwert erfasst und in die Liste gespeichert. Nachdem alle Sensoren erfasst wurden, werden die Werte mit Hilfe einer Printanweisung und eines Kennbuchstaben an die MATLAB®-GUI gesendet.

- chan_list[0] = Boilertemperatur
- chan_list[1] = Mischwassertemperatur
- chan_list[2] = nicht belegt
- chan_list[3] = Brühgruppentemperatur
- chan_list[4] = Brühgruppendruck
- chan_list[5] = Boilerdruck
- chan_list[6] = Leitungsdruck
- chan_list[7] = Wassertemperatur vor der Pumpe

Im Anschluss wird die Laufvariable auf 0 gesetzt und der erste Kanal am Multiplexer wird wieder eingestellt. Damit weiterhin der erste Messwert verworfen wird, ist das MWflag wieder auf 0 gesetzt.

4.5 Messdatenerfassung Basisplatine

Bei der Messdatenerfassung der Basisplatine wird der Durchfluss und der Füllstand an die MATLAB®-GUI übersendet.

Für den Füllstand ist der Pin 12, der als Eingang mit Pull-Up Widerstand konfiguriert ist, als Anschluss initialisiert worden. Da beim Füllstand überprüft wird, ob der Sensor kurzschließt, wird vom Pin der Wert abgefragt. Dieser Wert kann entweder 0 oder 1 betragen. Dabei steht 0 für „der Boiler ist noch nicht ausreichend gefüllt“ und 1 für den umgekehrten Fall. Der Wert wird abgefragt über Pinname.value(). Mit diesem Befehl wird der Füllstand mit dem Durchfluss über eine Printanweisung an die MATLAB®-GUI versendet.

Der Durchfluss wird mit zwei verschiedenen Funktionen bestimmt. Ursprünglich wird eine Funktion

über ein Interrupt vom Flowmeter ausgelöst und die andere mit Hilfe eines Timers periodisch mit einer Frequenz von 8 Hz wiederholt. Allerdings ist festgestellt worden, dass nach dem Starten des Timers der Microcontroller keine weiteren Befehle mehr entgegen nimmt und dadurch beispielsweise die Pumpe nicht mehr angesteuert werden kann. Somit wird versucht den Timer mit Hilfe eines PWM-Signals zu simulieren und die zweite Funktion ebenfalls mit einem Interrupt auslösen zu lassen. Beim Testen an einem anderen Raspberry Pico wurden zwei verschiedene Interrupts mit jeweils einem anderen PWM-Signal ausgelöst. Beide Funktionen wurden nacheinander abwechselnd ausgelöst. Um die Grenzen des Vorhabens zu kennen, wurde ein PWM bei 8 Hz gelassen und das andere Signal mit einer Frequenz zwischen 1 kHz und 7 kHz betrieben. Dabei stellte sich heraus, dass ungefähr ab 5,5 kHz der schnelle Interrupt vom langsameren verschluckt wurde. Das Flowmeter versendet pro ml 39,9 Impulse [65]. Das bedeutet, bei 100 ml/s sind es 3990 Impulse die Sekunde. Beim Kaffeebezug wird die Espressomaschine in der Regel mit einem Durchfluss von 1ml/s betrieben. Somit ist die Häufigkeit der Impulse niedriger als die herausgefundene Grenze. Diese Lösung hat die Situation ebenfalls nicht verbessert, bis die Initialisierung des Timers und des Interrupts über MATLAB® erfolgt ist und eine leere While-Schleife, die das Programm aufgehalten hat, entfernt wurde. Vermutet wird, dass die While-Schleife den Microcontroller daran hindert neue Befehle von der MATLAB®-GUI zu verarbeiten. Denn der Prozessor kann nur eine Aufgabe zur selben Zeit abarbeiten.

Die Lösung des Problems ist, dass die Initialisierung des Timers und des Interrupts von der MATLAB®-GUI durchgeführt wird. Somit läuft auf dem Raspberry Pico kein Programm mehr, sondern der Timer löst periodisch eine Funktion aus und wird durch den Interrupt immer wieder kurz unterbrochen. Dadurch wird die Abarbeitung der Funktionen auf der Basisplatine gewährleistet.

Die erste Funktion, die bereits vorliegt, misst die Zeit zwischen den gesendeten Impulsen und errechnet daraus den momentanen Durchfluss. Danach wird der Mittelwert auf zwei Nachkommastellen genau gebildet. Als nächstes wird abgefragt, ob schon 20 Werte vom Flowmeter in die Liste eingetragen wurden. Ist dies der Fall wird der Listenindexzähler flow1_List_Counter auf 0 gesetzt. Wenn der Mittelwert des Durchflusses einen Wert von 0,1 nicht übersteigt, wird der momentane errechnete Durchfluss direkt übernommen und durch vier geteilt, da sich bei geringem Durchfluss die Liste langsam füllt. Anschließend werden große Messspitzen herausgefiltert, die zu sehr vom Mittelwert abweichen. Zuletzt wird der neue Startzeitpunkt zur Durchflussmessung festgelegt und der Interruptzähler wird um eins erhöht.

Die zweite angepasste Funktion, die mit einem Timer getriggert wird, gibt die Messwerte der Platine an die MATLAB®-GUI weiter und setzt bei keinem Auslösen der Flowmeterfunktion an der entsprechende Stelle in der Liste den Wert auf 0. Zuerst wird abgefragt, ob der Interruptzähler der ersten Funktion auf 0 ist. Tritt dieser Fall auf, wird der Mittelwert des Durchflusses ebenfalls auf 0 gesetzt. Danach wird überprüft, ob der flow1_List_Counter größer als 19 ist. Sobald diese Bedingung erfüllt ist, wird der Zähler zurückgesetzt. Anschließend wird in der Liste an der richtigen Stelle eine 0 gespeichert und der Listenzähler wird um eins erhöht. Zuletzt wird mit einer Printanweisung in dem das

Kürzel „B!“ enthalten ist, die Messwerte an die MATLAB®-GUI gesendet und der Interruptzähler wird auf 0 gesetzt.

4.6 Pumpensteuerung

Die Pumpe wird über einen I2C-Bus mit dem DAC gesteuert. Dabei gibt der Pin 3 den Takt mit einer Frequenz von 400 kHz wieder und der Pin 2 ist für die Übertragung des digitalen Signals an den DAC verantwortlich. Anschließend wird die Adresse des Digital-Analog-Converters mit dem i2c.scan() [41] ermittelt. Betrieben wird die Pumpe, in dem mit dem Befehl ‚dac.value(Wert)‘ ein entsprechendes Spannungssignal an das DAC geschickt wird. Dabei entspricht ein Wert von 0 gleich 0 V und ein digitaler Wert von 4095 gleich der maximalen Steuerspannung von 5 V.
Beim Starten der Platine hat der DAC einen vorgelegten Initialwert, wodurch der Motor der Pumpe anfängt sich zu drehen. Um dies zu verhindern, wird ein Bootprogramm für die Basisplatine installiert. Sobald der Pico eine Spannung bekommt und sich einschaltet, wird durch das Bootprogramm die I2C-Leitung initialisiert. Zuletzt wird der Wert im DAC auf 0 gesetzt.

4.7 Troubleshooting

Die Systemelektronik hat im ersten Aufbau, Umstieg von der STM32 zu den Raspberry Picos, im Gegensatz zum jetzigen Unterschiede gehabt. Die SSR-Platine soll in Zukunft keine 230 V Magnetventile mehr schalten sondern 24 V Magnetventile. Zudem sind diese 24 V Ventile aus Keramik, wodurch beim Durchlaufen des warmen Wassers weniger Wärmeverluste entstehen. Deswegen ist die Verbindung zu den 230 V Magnetventilen mit der 24 V SSR-Platine über eine Zwischenlösung realisiert worden. Die zuvor eingebaute SSR-Insel hat die 230 V Ventile mit Darlington-Transistoren und Relais geschalten. Um diese zu Steuern wurde eine Zwischenplatine gebaut, mit der die Magnetventile gesteuert werden können (siehe Anhang 7). Die 24 V SSR-Platine hat mit dem Schalten der SSR-Schalter den Strom bzw. die Spannung auf Masse gezogen oder die SSR-Insel hat mit der Spannung das Ventil geöffnet. Durch dieses Vorgehen sind die Ventile bei unbestromten Zustand der Schalter an der 24 V SSR-Platine immer alle geöffnet. Somit läuft das Wasser beim Öffnen der Frischwasserleitung durch das ganze System, weil in der Hausleitung ein gewisser Druck zum Durchströmen vorliegt. Das hat dazu geführt, dass die Abtropfschale überflutet und das Labor nass wird. Also entwickelt sich die Idee, wie bei der Pumpensteuerung beim Einschalten der SSR-Platine alle Relais zu aktivieren und somit alle Magnetventile zu schließen. Anschließend wird getestet, ob die Ventile sich schalten lassen. Beim Ändern des Zustands wusste der Microcontroller nicht, ob der Schalter an der 24 V SSR-Platine geöffnet oder geschlossen sein soll. Dadurch schaltet das Relais ganz schnell hin und her, wodurch sich das Magnetventil ebenfalls mit geschlossen und geöffnet hat. Somit entsteht bei einer stromdurchflossenen Spule im Magnetventil eine Spannung durch Induktion.

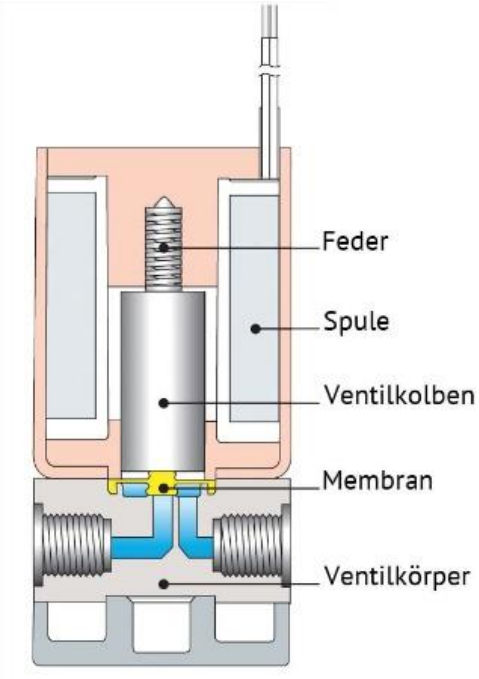


Abbildung 6: Aufbau Magnetventil [xii]

Hierbei entsteht eine Überspannung, die der 230 V Spannungsregler auch herunterregelt. Deswegen erhält der 24 V Spannungsregler mehr Spannung als er wandeln kann. Die Komponenten sind für bestimmte Ströme und Spannungen ausgelegt. Die drei Platinen sind durch den Programmierfehler alle durchgebrannt. Ein besserer Lösungsansatz um das Problem zu lösen ohne die Systemelektronik zu überlasten, ist die Initialisierung zur Schließung der Magnetventile über MATLAB®-GUI zu implementieren. Damit wird ein Pin am Raspberry Pico nicht mit zwei Zuständen gleichzeitig belegt, sondern wird einmal geschlossen und das System wird nicht mit Wasser durchströmt. Die Wasserleitung wird erst nach der Einstellung händisch geöffnet. Deswegen ist die 230 V SSR-Platine eingebaut worden und die Überbrückung mit Hilfe der Zwischenplatine entfallen.

5. Überblick MATLAB® APPDESIGNER

Die Bedienung der labortechnischen Espressomaschine erfolgt über das Programm MATLAB® Appdesigner. Mit diesem Programm ist es möglich eine graphische Oberfläche zu erstellen und dabei die Funktionen von MATLAB® selbst zu nutzen. Es wird ein Überblick verschafft, wie die Aktoren über die MATLAB®-GUI gesteuert und Sensoren ausgelesen werden.

Die Aktoren werden in MATLAB® über Callbacks gesteuert. Callbacks sind dabei Funktionen, die durch eine Aktion auf der Bedienoberfläche eine andere Funktion auslösen [111]. Soll das Ventil zum Öffnen des Frischwasser seinen Zustand ändern, so wird beim Betätigen des Buttons eine Callbackfunktion ausgelöst. Dabei wechselt das Bild des Ventils auf dem Button zwischen Durchfluss blockiert oder geöffnet. Zudem wird an die SSR-Platine der richtige Pin angesprochen, welcher den Schalter des Ventils steuert und das Ventil sich öffnet oder schließt [vi]. Auf diese Weise werden die Ereignisse, die durch die manuelle Bedienung auftreten, für die jeweiligen Aktoren implementiert. Für die automatisierten Vorgänge, wie die Regler, wird in den Messdatenverarbeitungen der Messplatine und der Basisplatine überprüft, ob die jeweilige Bedingung erfüllt ist. Zum Beispiel wird überwacht, ob gerade ein Kaffeebezug vollzogen wird.

Das Auslesen der Sensoren wird zeilenweise durchgeführt. Dabei schickt die jeweilige Platine ihre Messdaten mit einem Kürzel. Somit weis MATLAB®, welche der zwei Messdatenroutinen ausgelöst wird. Danach werden die Messdaten aufgespalten und in die jeweiligen vorgesehenen Felder auf der Benutzeroberfläche angezeigt. Zuvor werden diese aus Spannungswerten in physikalische Messwerte umgerechnet. Des Weiteren speichert MATLAB® die Messwerte in einem Puffer, welcher in einer separaten Datei abgespeichert wird. Anschließend ist es möglich, die Messwerte in MATLAB® anzuschauen [vii]. Die Regler beziehen sich auf die Werte in den Messwertfeldern für den Vergleichswert, um die Abweichung zu berechnen. MATLAB® bietet ebenfalls die Möglichkeit die Messwerte direkt in einem Graphen darzustellen. Dies ist in der jetzigen MATLAB®-GUI nicht vorgesehen.

Auf diese Weise wird die labortechnische Espressomaschine bedient. In Zukunft wird die MATLAB®-GUI als Instanz arbeiten, um die Kommunikation zwischen den Platinen aufrechtzuerhalten. Dabei werden die meisten Funktionen und die Regler auf den Platinen ausgelagert.

6. Anpassung und Programmierung der Regelung

Die bisherigen Regler sind bis auf den Füllstandsregler träge. Die Regelkreise sollen jeweils mit Hilfe eines PID-Reglers realisiert werden. Ein PID-Regler besteht aus drei Komponenten: dem Proportionalregler, dem Integralregler und den Differentialregler. Der P-Regler reagiert auf die aktuelle Abweichung (Sollwert – Messwert) und verstärkt diese mit dem Faktor K_p. Der I-Regler reagiert, wenn der P-Regler es nicht schafft den Sollwert zu erreichen. Dabei addiert dieser jede Abweichung in Abhängigkeit einer Regelfrequenz (Abweichung/Regelfrequenz). Diese Summer wird mit K_i verstärkt. Der D-Regler steuert gegen zukünftige Änderungen. Das heißt, ist die neue Abweichung viel größer als die letzte Abweichung, reagiert der D-Regler stärker und umgekehrt. Überschwinger werden dadurch gedämpft. In der Rechnung wird die Differenz der neuen Abweichung und der gemerkten Abweichung mit der Regelfrequenz multipliziert ((Abweichung neu – Abweichung alt) * Regelfrequenz) und mit dem Verstärkungsfaktor K_d angepasst [108]. Diese Faktoren lassen sich auf verschiedene Weise ermitteln. Meistens wird in der Praxis das empirische Verfahren gewählt. Das bedeutet, dass die Werte experimentell ermittelt werden, sei es direkt an der Espressomaschine oder über eine Simulation [76]. Der Füllstandsregler ist eine Ausnahme, da beim Erreichen des Füllstands die Befüllung aufhört. Dennoch wird auf die Funktionalität eingegangen.

6.1 Füllstandsregler

Wie eben erwähnt, ist der Füllstandsregler kein klassischer Regler. Im Prinzip überprüft das Programm über die Kurzschlussdetektion des Füllstands, ob der Boiler befühlt werden muss. Der Füllstandsregler wird über die Messdatenverarbeitung der Basisplatine in der MATLAB®-GUI aufgerufen. Dabei wird mit Hilfe eines Flags, welches die Aktivität des Füllstandsreglers anzeigt, verhindert, dass andere Programmabläufe von der Boilerbefüllung nicht unterbrochen werden. Zuerst wird abgefragt, ob der Boiler befüllt ist und der Füllstandsregler noch nicht eingeschalten ist. Sind die Bedingungen erfüllt, wird das Magnetventil für das Frischwasser und das Ventil zum Boiler geöffnet. Mit einer Pumpenspannung von 4000 mV wird der Boiler befüllt. Sobald die Kurzschlussdetektion eine 1 liefert und der Füllstands aktiv ist, ist die Abschaltbedingung erfüllt. Dann werden die Magnetventile geschlossen und die Pumpe ausgeschaltet [viii].

Am Füllstandsregler sind einige Probleme noch nicht behoben worden. Zum einen entsteht bei Befüllen des Boilers ein Druck. Dieser muss wieder abgelassen werden. Deswegen wird der Dampfhahn bei der Befüllung ebenfalls geöffnet, wodurch kein Druck aufgebaut wird. Des Weiteren kommt es sporadisch vor, dass beim Aufheizen durch die Bewegungen des Wassers beim Entschichten der Füllstand zwischen 1 und 0 hin und her springt. Beim Heizen setzt sich das kalte Wasser im unteren Teil des Boilers ab, somit muss diese Schicht gebrochen werden [ix]. Um einen undefinierten Zustand

des Füllstands zu verhindern, wird nach dem Erreichen des Füllstands die Pumpe nicht direkt abgeschaltet. Für 1,5 s wird der Boiler mit einer Pumpenspannung von 1000 mV leicht überfüllt. Somit ist für die Entschichtung des Wasser genug Flüssigkeit im Tank. Alternativ muss beim Entschichten das Aufheizen und das Umpumpen beendet werden. Anschließend darf bis zum Erreichen des Füllstands nachgefüllt werden. Beim Entschichten über den Dampfhahn ist dieses Phänomen unproblematisch, da kein Magnetventil im Wasserkreislauf geöffnet ist. Zudem ist es nicht möglich beim Umschalten des Schalters von an zu aus, den Füllstandsregler abrupt zu stoppen. Die Funktion wird durchgeführt bis der Füllstand erreichet ist. Erst wenn der Füllstand erreicht wurde, ist der Regler deaktivierbar.

6.2 Boilerdruckregler

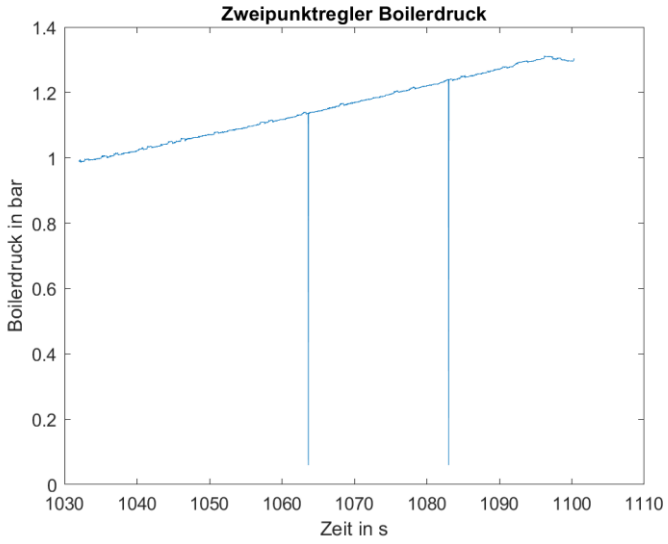


Abbildung 7: Verlauf 2-Punkt-Regler

Der Boilerdruckregler war zuvor ein Zweipunktregler. Das heißt, dass der Boiler mit einer festgelegten Leistung aufgeheizt wird. Sobald der Solldruck erreicht wird, schaltet sich das PWM-Signal aus und die Heizspirale wird ausgeschaltet. Ein Vorteil für diesen Regler ist, dass dieser sehr robust ist und einfach einzustellen [85]. Dennoch wird sich für ein PID-Regler entschieden, denn beim Zweipunktregler sieht man an Hand des Graphen, dass es selbst bei nur 75% der Heizleistung zu Überschwinger kommt. Somit wird das PWM-Signal immer wieder kurzzeitig aktiv und deaktiviert. Zudem ist eine Aufheizung mit 100% nicht möglich, da die Überschwinger noch größer werden.

Beim jetzigen Boilerdruckregler wird zuerst überprüft, ob Boilerfüllstand erreicht ist [ix]. Somit ist sichergestellt, dass die Heizwendel mit Wasser im Boiler umhüllt ist. Danach wird die Abweichung vom Solldruck berechnet. Da die Druckwerte in mbar angegeben sind, werden diese in bar umgerechnet. Anschließend wird das Stellsignal für die Heizung berechnet. Für die Berechnung des Integrals wird

das Rechteckverfahren angewendet (Ergebnis-Intergral = Ergebnis-Integral + (Abweichung / Regelfrequenz)). Es wird auf den letzten Wert des Integrals die anliegende Abweichung, die durch die Regelfrequenz dividiert bzw. mit der Abtastrate multipliziert wird, addiert. Um einen Windup des I-Anteils zu verhindern, wird die Aufsummierung durch eine Abfrage eingefroren (Anti-Windup). Das „Windup“ Phänomen entsteht durch Integrationsfehler des I-Anteils. Dieser steigt weiter an, obwohl das Stellglied sein Maximum bzw. Minimum erreicht hat. Der Integrator steigt so lange an bis es zu einem Vorzeichenwechsel vom Regelfehler kommt. Somit zieht sich der I-Anteil auf und destabilisiert das System [109]. Das bedeutet, dass dieser sich nicht ändert solange das Stellsignal sich nicht innerhalb der Grenzen befindet. In diesem Fall beträgt die Heizleistung zwischen 0% und 100%. Zuletzt wird für die nächste Berechnung die alte Abweichung gespeichert. Nach der Berechnung des Stellsignals wird überprüft, ob dieser sich innerhalb der Grenzen befindet. Ist dies der Fall, wird die berechnete Heizleistung übernommen. Anderenfalls wird bei einer Heizleistung kleiner 0% auf 0 und bei einem Stellsignal größer 100% auf 100 gesetzt. Zuletzt wird das Ergebnis in ein PWM-Signal umgewandelt und an die SSR-Platine weitergeleitet.

Zum Einstellen des Reglers wurde mit Hilfe des vorherigen Zweipunkt-Reglers die Strecke aufgenommen. Dabei stellt man wie in Abbildung 7 fest, dass es sich um eine I-Strecke bzw. Integrale Strecke handelt. Dies ist daran zu erkennen, dass der Graph stetig wie eine Gerade steigt. Dabei ergibt sich eine Steigung von 0,005. Mit dem MATLAB® Programm Simulink ist es möglich Regelkreise zu simulieren. Dadurch hat sich nach einigen Simulationen ergeben, dass der Regler mit einem K_p von 1000, einem K_i von 0 und K_d von 10 am besten funktioniert. Beim Ermitteln der Proportionalverstärkung wird mit der Überlegung begonnen, bei welcher Verstärkung die Heizleistung maximal wird, wenn die Abweichung so groß wie möglich ist. Bei der Boilerregelung beträgt der Sollwert 1300 mbar bzw. 1,3 bar. Daraus resultiert eine Mindestverstärkung von 77, wenn die maximale Abweichung von 1,3 bar durch die 100 % Heizleistung dividiert wird. Danach ist festgelegt worden, ab welcher Sollwertabweichung das Herunterregeln des PWM-Signals beginnt. Somit ergibt sich für K_p ein Wert von 800, weil das PWM-Signal ab einer Abweichung von 125 mbar anfangen soll kleiner zu werden. Dies wurde dann auch an der Maschine getestet und ein Endwert von ungefähr 1290 mbar hat sich eingependelt. Dies ist an der folgenden Abbildung zu erkennen.

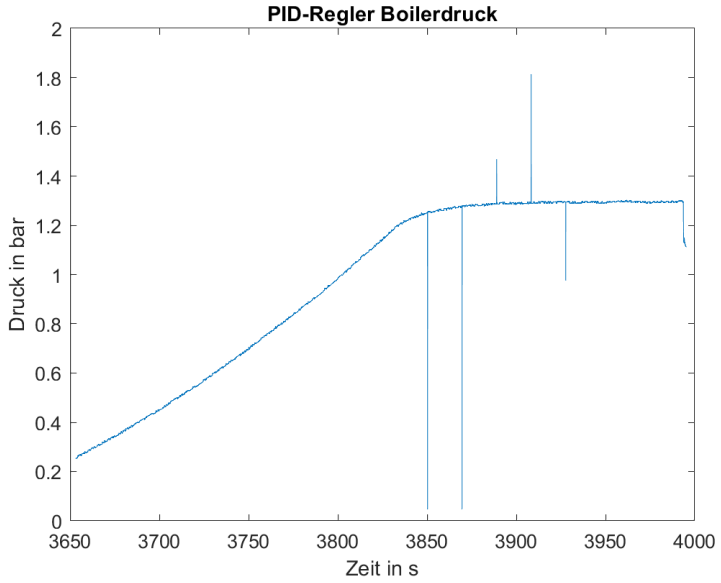


Abbildung 8: Endgültiger Boilerregler Verlauf

Der Integralfaktor des PID-Reglers muss bei der Boilerdruckregelung auf 0 gesetzt sein. Bei einem Versuch mit $K_i = 5$ wurde durch den I-Anteil im Regler der Regelkreis instabil. Somit wird der Solldruck von 1300 mbar erreicht, aber das Überschwingverhalten verstärkt sich. Die Entscheidung fällt trotz des kleinen Nachteils, dass die stationäre Genauigkeit erst nach sehr langer Zeit erreicht werden kann, auf den Einsatz eines PD-Reglers. Der Differentialfaktor wird aus der Simulation entnommen, da bei zu großem K_d das System und das Stellglied zu sehr schwingt.

6.3 Mischregler

Der Mischregler sorgt dafür, dass an der Brühgruppe, an der der Kaffee mit heißem Wasser durchflossen wird, die richtige Wassertemperatur herrscht. Zuerst wird unterschieden, ob ein Kaffeebezug oder ein Teebezug angefordert wird. Dabei wird überprüft, ob das Ventil zum Umschalten zwischen Brühgruppe und Teelanze bestromt ist. Ist dies nicht der Fall, handelt es sich um einen Kaffeebezug. Die prinzipielle Funktion des Reglers zwischen beiden Fällen ist die gleiche, aber für Versuchszwecke soll dafür eine Unterscheidung getroffen werden. Anschließend wird der PID-Regler erst reagieren, wenn die Isttemperatur am Mischerwassersensor größer als 80 °C beträgt. Dadurch werden Überschwinger reduziert. Die ideale Brühtemperatur beträgt für einen Kaffee 90 +/- 2 °C [74]. Danach wird wie beim Mischregler die Abweichung berechnet. Bei dieser Abweichung wird der Messwert vom Sollwert abgezogen, da bei einer zu hohen Temperatur das Dosierventil aufgedreht wird, um mehr kaltes Wasser zum heißen Wasser zu mischen. Mit Hilfe der richtigen Verstärkungsfaktoren berechnet der Regler durch die Temperaturdifferenz die gerundete Schrittanzahl, um die der Schrittmotor verstellt werden muss. Danach wird die Schrittzahl mit der jetzigen Schrittmotorposition verrechnet. Zuletzt merkt sich der Regler die alte Abweichung und berechnet den neuen Integralwert. Die Berechnung des Integralwerts erfolgt wie beim Boilerdruckregler

über das Rechteckverfahren. Ebenso wird durch Anti-Windup verhindert, dass der Integralwert unkontrolliert zu groß bzw. zu klein wird. Der I-Anteil wird eingefroren, wenn die Mischerpostion kleiner als 50 und größer als 200 beträgt. Beim Einstellen des Reglers ist die Mischerpostion des Dosierventils meistens zwischen 40 und 180 variiert. Nach dem die richtige Position bestimmt wurde, wird auf die Stellgrenzen überprüft, ob die neue Dosierventilposition eingestellt werden darf. Dabei wird bei einer Position kleiner als 0 das Ventil komplett geschlossen, da dies die mechanische Grenze ist. Durch die vorherigen Versuche ist die obere Grenze bei 250 Schritten festgelegt worden. Falls die Temperatur doch zu schnell steigt und deswegen das Dosierventil zu weit öffnet, fällt die Mischtemperatur unter 80 °C, da der Regler der Temperatur entgegen wirkt und somit die untere Grenze erreicht. Dadurch arbeitet durch die Abfrage „Mischtemperatur > 80°C“ der Regler nicht mehr. Der Grund für dieses Phänomen besteht darin, dass das System eine kurze Zeit braucht auf die Temperaturänderungen zu reagieren.

Damit der PID-Regler auch wie gewünscht funktioniert, werden die Verstärkungsfaktoren der jeweiligen Anteile ermittelt. Für K_p wird der Schwingversuch von Ziegler Nicholson durchgeführt. Dabei wird der Proportionalfaktor solange erhöht, bis das System anfängt zu schwingen [110]. Somit ergibt sich ein K_{pkrit} von 0,5. Mit der Tabelle aus [110] wird K_p = 0,3 errechnet. Nach weiteren Tests hat sich ergeben, dass für K_p = 0,4 der Regler am besten arbeitet. Die Periodendauer ist bei dem Versuch nicht ablesbar gewesen, wodurch für K_i und K_d der experimentelle Ansatz gewählt wurde. Dadurch ergeben sich die Werte K_i = 0.005 und K_d = 0.01. In der Abbildung 9 wird die geregelte Strecke unter dem Einfluss des Durchflusses gezeigt.

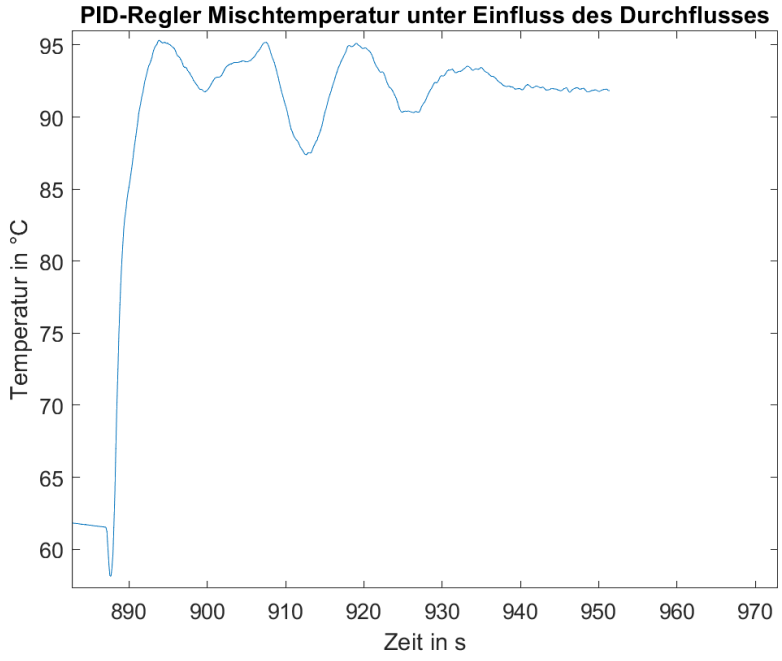


Abbildung 9: Finaler Stand Mischregler

Die gewünschte Solltemperatur wird meistens nicht richtig erreicht. Die Abweichung im eingeregelten

Zustand beträgt zwischen 0,5 °C bis 1 °C. Diese Problematik entsteht durch das Runden der Schrittzahl zum Einstellen der Temperatur. Im Moment wird der errechnete Wert auf die nächst kleinere ganze Zahl mit dem Rundungsbefehl floor() in MATLAB® abgerundet. Mit der Methode ceil() wird der Wert auf die nächst höhere ganze Zahl aufgerundet. Dadurch entsteht der gegenteilige Effekt, dass die Temperatur um die gleiche Differenz zu klein ist. Da von der Mischposition zur Brühgruppe ein Temperaturverlust von 5-8 K vorliegt, ist eine etwas höhere Temperatur nicht problematisch. Des Weiteren ist die Wasserwendel, in der sich das Wasser aufheizt, bevor es sich mit dem kalten Wasser mischt, teilweise undicht. Dadurch gelangt kaltes Wasser in den Boiler. Dabei sinkt die Temperatur im Boiler und der Wärmeaustausch unterstützt dies. Für den nächsten Durchgang wird jedes Mal der Dampfhahn kurz geöffnet, damit Druck aus dem Boiler entweicht. Der Boilerregler erhöht das Stellsignal um seinen Sollwert zu erreichen, wodurch die Kesseltemperatur von 120 °C wieder erreicht wird. Zuletzt ist die gemessene Temperatur ebenfalls abhängig vom Durchfluss. Ist dieser geringer, wird das Dosierventil weiter geöffnet um die richtige Temperatur einzustellen und umgekehrt. Deswegen wird der Durchflussregler mit einer Verzögerung eingesetzt, damit keine noch größeren Änderungen auftreten, wodurch der Mischregler zu lange brauchen würde, diese Temperaturschwankungen auszugleichen. Außerdem muss die Datenrate, von 30 Messwerten pro Sekunde auf 8 Messwerte pro Sekunde, der Messplatine gesenkt werden. Sobald der Mischregler eingesetzt hat, ist die MATLAB®-GUI eingefroren. Dies hat den Nachteil, dass die Regler weniger Messwerte bekommen. Dennoch wird die Solltemperatur mit der Toleranz von 0,5 °C bis 1 °C erreicht.

6.4 Durchflussregler

Durch den Durchflussregler wird sichergestellt, dass für einen einfachen Espresso eine Durchflussrate von 1 ml/s und für einen doppelten eine Durchflussrate von 2 ml/s eingehalten werden [74]. Zuerst wird gewartet, bis der Mischregler im Vorlauf eingeschwungen ist. Danach wird das Ventil, welches das Wasser in dieser Zeit in den Überlauf leitet, auf die Brühgruppe umgeschaltet. Anschließend muss das warme Wasser den Kaffeepuck in der Brühgruppe befeuchten und aufquellen lassen. In der Zwischenzeit wird ein Druck in der Brühgruppe aufgebaut. Dabei hat sich herausgestellt, dass eine Verzögerung 6s eine gute Wartezeit ist, bis der Durchflussregler anfängt zu reagieren. Sobald diese Schritte abgeschlossen sind, fließt der Kaffee in die Tasse und der Durchflussregler soll währenddessen den Solldurchfluss einhalten.

Für den Durchfluss wird ebenfalls ein PID-Regler verwendet. Wie bei den vorherigen Reglern wird ebenfalls eine Abweichung bestimmt. Anschließend wird mit den richtigen Verstärkungsfaktoren die Pumpenspannungsdifferenz berechnet. Danach ändert sich die Pumpenspannung um den errechneten Wert. Bevor die neue Pumpenspannung von MATLAB® an die Basisplatine übergeben wird, wird die alte Durchflussabweichung gespeichert und der Integralwert berechnet. Dieser ermittelt sich ebenfalls mit dem Rechteckverfahren, in dem der zuvor berechnete Integralwert zu der Division

von der Abweichung durch die Regelfrequenz addiert wird. Wegen dem enthaltenen I-Anteil muss ein Anti-Windup in die Regelung eingebaut werden. Sobald die Pumpenspannung unter 1000 mV fällt oder 2100 mV überschreitet, wird der letzte Wert des I-Anteils behalten, um ein Übersteuern und Überschwinger zu verhindern. Bei den Versuchen einen Solldurchfluss von 2 ml/s einzuhalten, hält sich die Leistung in dem erwähnten Bereich. Zuletzt wird überprüft, ob die berechnete Pumpenspannung die minimale Spannung von 0 unterschreitet oder die maximale von 5000 mV überschreitet. Ist dies nicht der Fall wird der digitale Spannungswert für den DAC errechnet. Dazu gibt es eine Überprüfung, ob der neue berechnete Spannungswert dem alten entspricht. Somit sendet MATLAB® nur eine neue Pumpenspannung an die Basisplatine, wenn sich die Spannung ändert, um die Datenmenge im Netzwerk zu reduzieren. Falls eine andere Spannung ermittelt wurde, wird der neue Stellwert an die Platine übergeben und für die Abfrage in MATLAB® abgespeichert.

Damit der PID-Regler das Stellsignal richtig berechnet, werden die Verstärkungsfaktoren für den Regler ermittelt. Zuerst wird wie beim Mischregler der Schwingversuch von Ziegler Nicholson durchgeführt. Dabei ergibt sich ein K_{pkrit} von 16. Daraus errechnet sich ein K_p von 9,6. Nach dem K_i und K_d herausgefunden wurden, ist K_p = 10 am besten gewesen. Die Periodendauer der Schwingung ist schwer ablesbar, wodurch K_i und K_d experimentell ermittelt werden. Für K_i ergibt sich der Wert 0,05. Wird dieser kleiner gewählt, ist der Einfluss des I-Anteils zu gering. Der Verstärkungsfaktor K_d hat den Wert 1. Wird dieser zu groß eingestellt, fängt das System an zu schwingen. So werden die Überschwinger abgedämpft ohne das System träge werden zu lassen. Diese Vorfaktoren wurden mit einem leeren Siebträger ermittelt und mit einem gefüllten ausgetestet. Die Ergebnisse decken sich. Die geregelte Strecke ist in Abbildung 10 zu sehen.

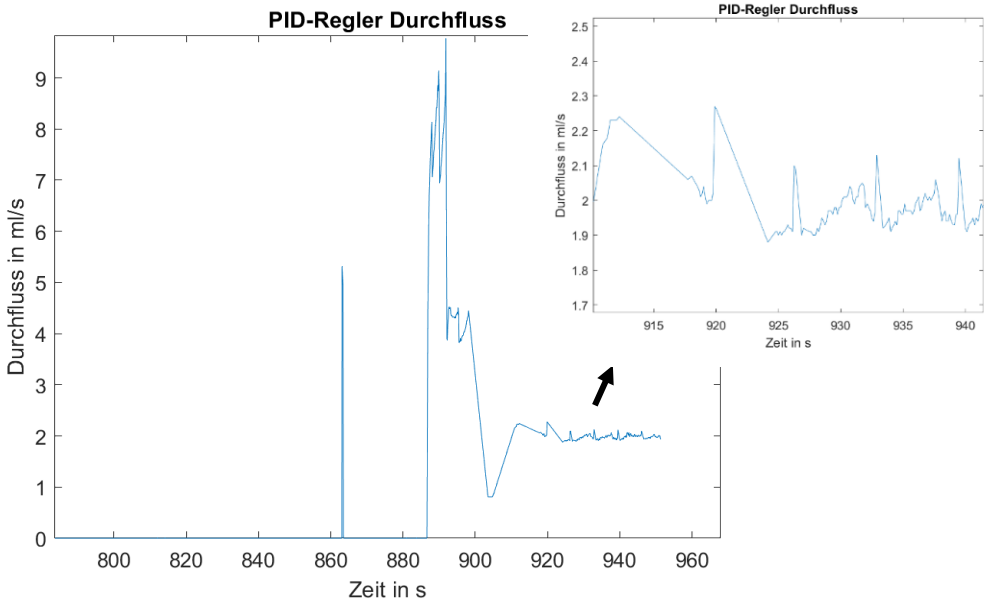


Abbildung 10: Geregelter Durchfluss

Dabei ist zu sehen, dass der Durchfluss von 2 ml/s nicht exakt gehalten wird. Trotzdem schafft der

Regler den Durchfluss mit einer Toleranz von +/- 0,1 ml/s zu halten. Das Problem ist, dass MATLAB® den erhaltenen Durchflusswert von der Basisplatine ab und zu nicht schnell genug verarbeitet. Somit ist der Istwert, der in einem Textfeld gespeichert wird, bei manchen Durchläufen des Reglers nicht korrekt, wodurch die Pumpenspannung zu sehr steigt oder abfällt. Dadurch kommt das System aus seinem eingeschwungenen Zustand heraus und muss sich neu einregeln. Deswegen entstehen die Messspitzen außerhalb der Toleranz nach dem Einschwingen. Wird der Wert direkt von der Basisplatine in eine Variable gespeichert und im Regler in das richtige Datenformat gewandelt, hängt sich MATLAB®, solange die Funktion nicht wieder beendet wird, auf. Hingegen wird die Umwandlung des Formats in der Datenverarbeitung der Basisplatine in MATLAB® durchgeführt und in eine separate Variable gespeichert. Dasselbe Problem tritt auf, wenn der Durchflussregler den Istwert aus dem Textfeld nimmt. Dieses Problem ist zu lösen, in dem der Computer MATLAB® mehr Rechenleistung abgibt. Dies ist durch das Freigeben eines weiteren Rechenkerns zu ermöglichen. Denn MATLAB® arbeitet nur mit einem Kern. Dadurch ergibt sich die Möglichkeit, Prozesse parallel ablaufen zu lassen. Dies wird in der Zukunft getestet.

7. Zusammenfassung und Ausblick

Ein automatischer Kaffebezug ist nach dem Umbau der Systemelektronik und der Anpassung der Regler möglich. Zudem werden die Sollparameter vom Nutzer festgelegt und die Regelparameter sowie der letzte Wert des Stellglieds gespeichert. Des Weiteren ist es möglich, weiterhin einen Kaffeebezug manuell durchzuführen.

Die Systemelektronik ist komplett überarbeitet worden. Dabei teilen sich die Aufgaben von zuvor einer Platine auf drei Platinen auf. Die SSR-Platine schaltet die Magnetventile, steuert über ein PWM-Signal die Heizwendel und stellt den Schrittmotor für den Mischer ein. Die Messplatine übernimmt die Messdatenverarbeitung der Sensoren. Dabei ist zu beachten, dass der ADC so eingestellt wurde, dass dieser nur Spannungen von 4096 mV aufnimmt, da die Sensoren in der Regel keine Spannungen über 4096 mV an den ADC abgeben. Falls für andere Versuche die komplette Bandbreite der Sensoren gebraucht wird, stellt man dies im MicroPython-Programm um. Zuletzt steuert die Basisplatine die Brühgruppendrossel und den Bypass ein. Außerdem wird an dieser Platine der Durchfluss gemessen und die Pumpe angesteuert. Beim Einstellen des Durchflussreglers ist das Bypassventil an der Getriebepumpe komplett geschlossen gewesen. Da dieses Ventil nicht gebraucht wurde, könnte es in Zukunft entfernt werden.

Die vier Regler sind alle funktionsfähig und angepasst bzw. neu implementiert worden. Dabei wird das Problem mit dem Messwert des Durchflusses nicht behoben. Die Regler werden in nächster Zeit auf die Raspberry Picos verlegt. Somit ist die Datenmenge für die Kommunikation mit MATLAB® geringer, wodurch die Messplatine wieder mehr Messdaten an die Regler weiterleiten wird. Zudem bekommt der Durchflussregler direkt den richtigen Durchflusswert, ohne dass dieser verarbeitet wird. Des Weiteren ist der Mischregler bei einem Boilerdruck von 1000 mbar und 1500 mbar getestet worden. Dabei stellte sich eine Temperatur von 114 °C im Kessel bei 1000 mbar ein. Somit ergibt sich eine Mischtemperatur von 93 °C. Bei einem Boilerdruck von 1500 mbar und einer Kesseltemperatur von 123 °C fängt das System des Mischers an zu schwingen. Deswegen müssen noch für die anderen Boilerdrücke die richtigen Vorfaktoren des PID-Reglers ermittelt werden.
Da der Mischregler abhängig vom eingestellten Durchfluss ist, würde sich eine Kaskadenregelung anbieten. Das bedeutet, dass das Ergebnis des Durchflussreglers im Mischregler berücksichtigt wird. Der Regelung des Durchflusses stellt sich schneller ein als der Mischregler, wodurch der Durchflussregler sich als innerer Regelkreis anbietet. Dabei muss nach der Verlegung der Regler auf die Picos die Kommunikation zwischen Basisplatine und SSR-Platine berücksichtigt werden.

Um in Zukunft Fehlfunktionen eines Potentiometers zu vermeiden, werden an den äußeren Anschlüssen jeweils ein Widerstand verbaut. Sollte bei der Bedienung des Potentiometers ein

Kurzschluss verursacht werden, lassen die Widerstände den Strom nicht schlagartig unendlich groß werden. Dadurch werden die Komponenten auf der Platine nicht beschädigt.

Magnetventile haben die Eigenschaft, dass die Ventile in Flussrichtung komplett abdichten und in die andere Richtung ab einem bestimmten Druck undicht werden. Beim Betrieb der Espressomaschine ist dies hin und wieder beim Ablassventil des Boilers vorgekommen. Als Gegenmaßnahme werden an den Magnetventilen, an denen die Problematik auftreten kann, Rückschlagventile eingebaut. Diese helfen den Leitungsstrang in beiden Richtungen abzudichten. Bei manchen Versuchen ist es vorgekommen, dass sich der Boiler über das Ablassventil gefüllt hat.

Abbildungsverzeichnis

Anhangsverzeichnis

Literaturverzeichnis

1. Literaturverzeichnis des Institut für Kaffeetechnologie

http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Technische_Beeinflussbarkeit_der_Geschmacksache_Kaffee:Literatur

2. Externes Literaturverzeichnis

[i] Rohnen, Armin: Multi-MCU-Elektronik:SSR-Platine (23.02.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Multi-MCU-Elektronik:SSR-Platine (Stand: 09.07.2023)

[ii] Rohnen, Armin: Multi-MCU-Elektronik:SSR-Platine (23.02.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Multi-MCU-Elektronik:Messplatine (Stand: 13.07.2023)

[iii] Rohnen, Armin: Schalten Magnetventile SSR-Platine Multi-MCU (30.07.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Schalten_Magnetventile_SSR-Platine_Multi-MCU (Stand: 16.08.2023)

[iv] Rohnen, Armin: Schrittmotorensteuerung Mischer (15.08.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Schrittmotorensteuerung_Mischer (Stand:16.08.2023)

[v] Rohnen, Armin: Schrittmotorsteuerungen SSR-Platine Multi-MCU (30.07.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Schrittmotorsteuerungen_SSR-Platine_Multi-MCU (Stand: 16.08.2023)

[vi] Rohnen, Armin: MATLAB® -GUI Schalten Magnetventile (30.07.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=MATLAB%C2%AE-GUI_Schalten_Magnetventile (Stand: 15.09.2023)

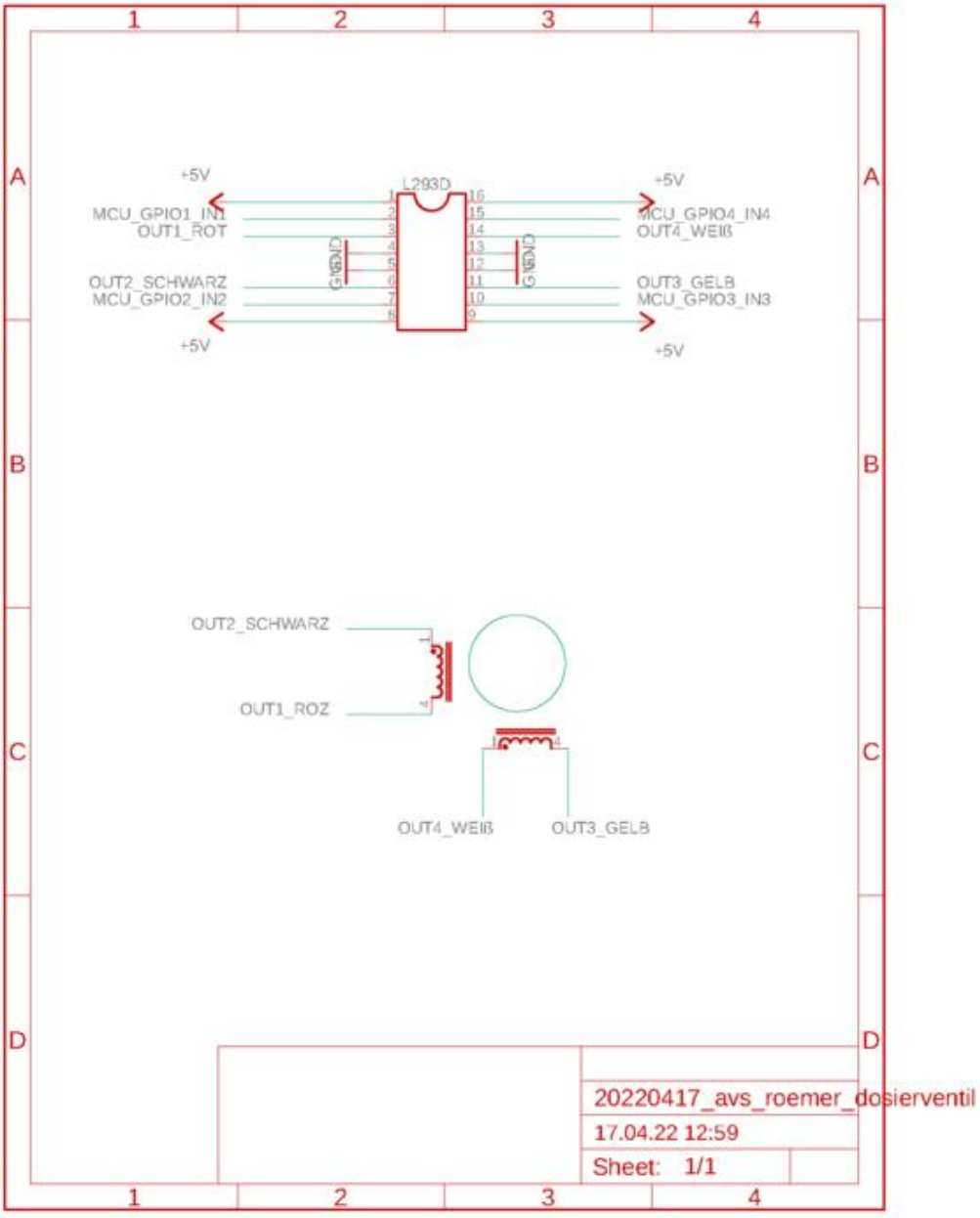
[vii] Rohnen, Armin: Messwerte erfassen Multi-MCU (13.08.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Messwerte_erfassen_Multi-MCU (Stand: 15.09.2023)

[viii] Rohnen, Armin: Füllstandsregler Multi-MCU (13.08.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=F%C3%BCllstandsregler_Multi-MCU (Stand: 08.09.2023)

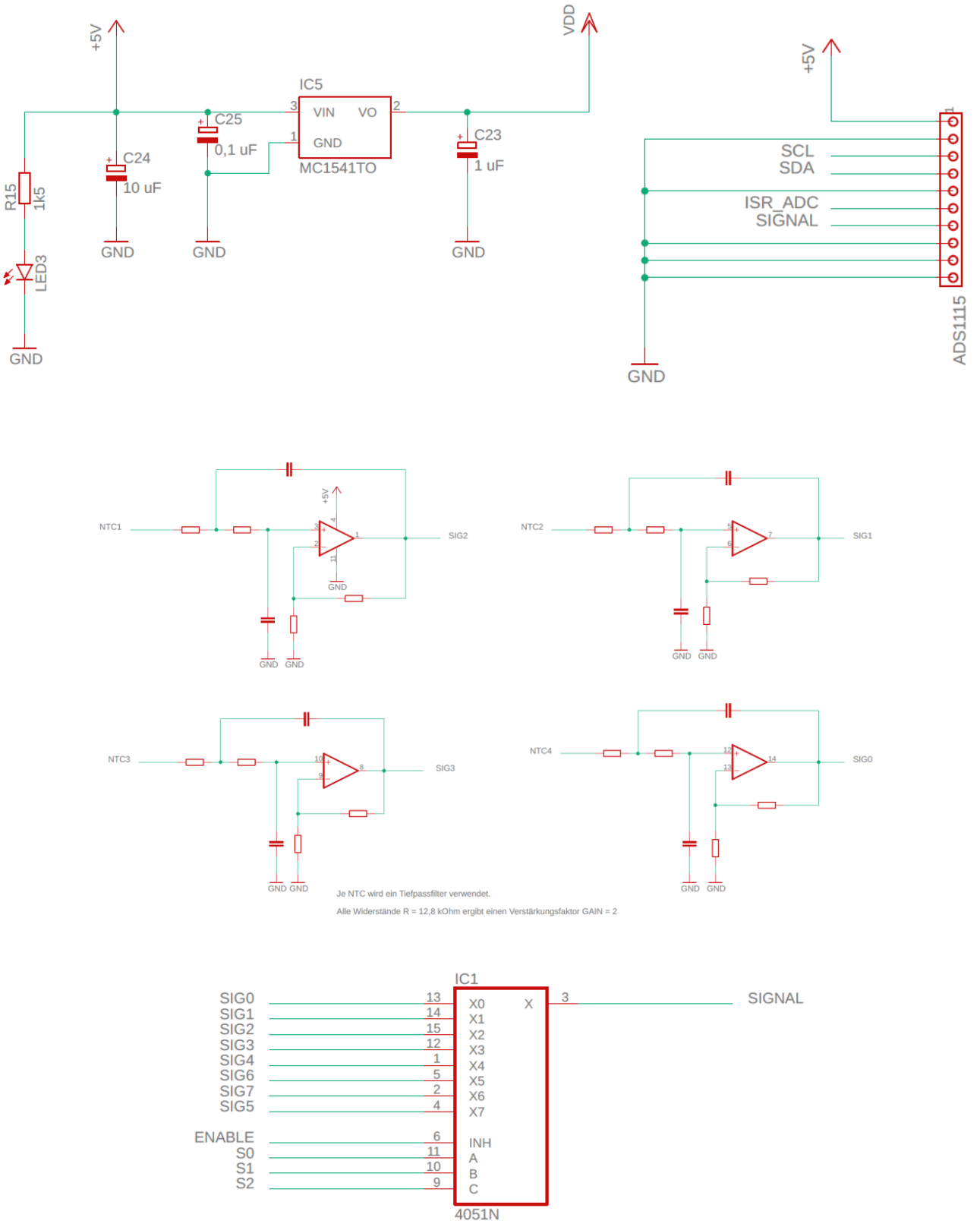
[ix] Rohnen, Armin: Regler Boilerdruck Multi-MCU (13.08.2023) http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Regler_Boilerdruck_Multi-MCU (Stand: 12.09.2023)

[x] Rohnen, Armin: Systemelektronik (07.02.2023)
http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=Systemelektronik
(Stand: 25.09.2023)
[xi] Makeabilitylab: Lesson 4: Potentiometers (o.J)
https://makeabilitylab.github.io/physcomp/arduino/potentiometers.html (Stand: 25.09.2023)
[xii] BMT Fluid Control Solution: Technische Informationen zu mediengetrennten Magnetventilen (o.J)
https://www.pumpen-ventile.de/technische-informationen/mediengetrennte-magnetventile/ (Stand: 25.09.2023)
[xiii] Rohnen, Armin: AVS Römer Elektronisches Dosierventil (17.04.2022)
http://www.institut-fuer-kaffeetechnologie.de/Wiki/index.php?title=AVS_R%C3%B6mer_Elektronisches_Dosierventil
(Stand: 26.09.2023)
[xiv] Rohnen, Armin: Dateien Systemelektronik (o.J)
http://www.institut-fuer-kaffeetechnologie.de/Intern/index.php?title=Dateien_Systemelektronik (Stand: 26.09.2023)
[xv] Electronic Tutorials: NAND-Gatter Tutorial (o.J)
https://www.electronics-tutorials.ws/de/logische/nand-gatter.html (Stand: 09.10.2023)

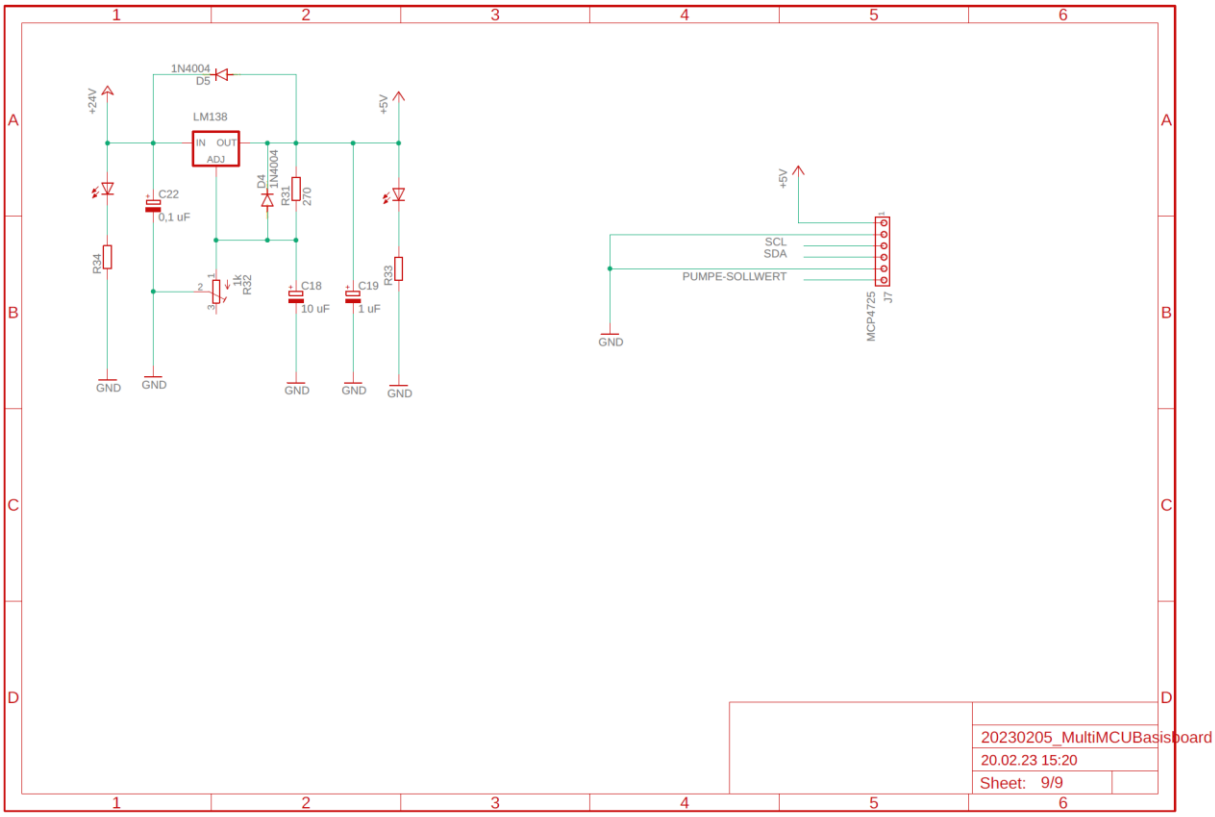
Anhang



Anhang 1: Schaltplan Schrittmotortreiber mit Schrittmotor [xiii]

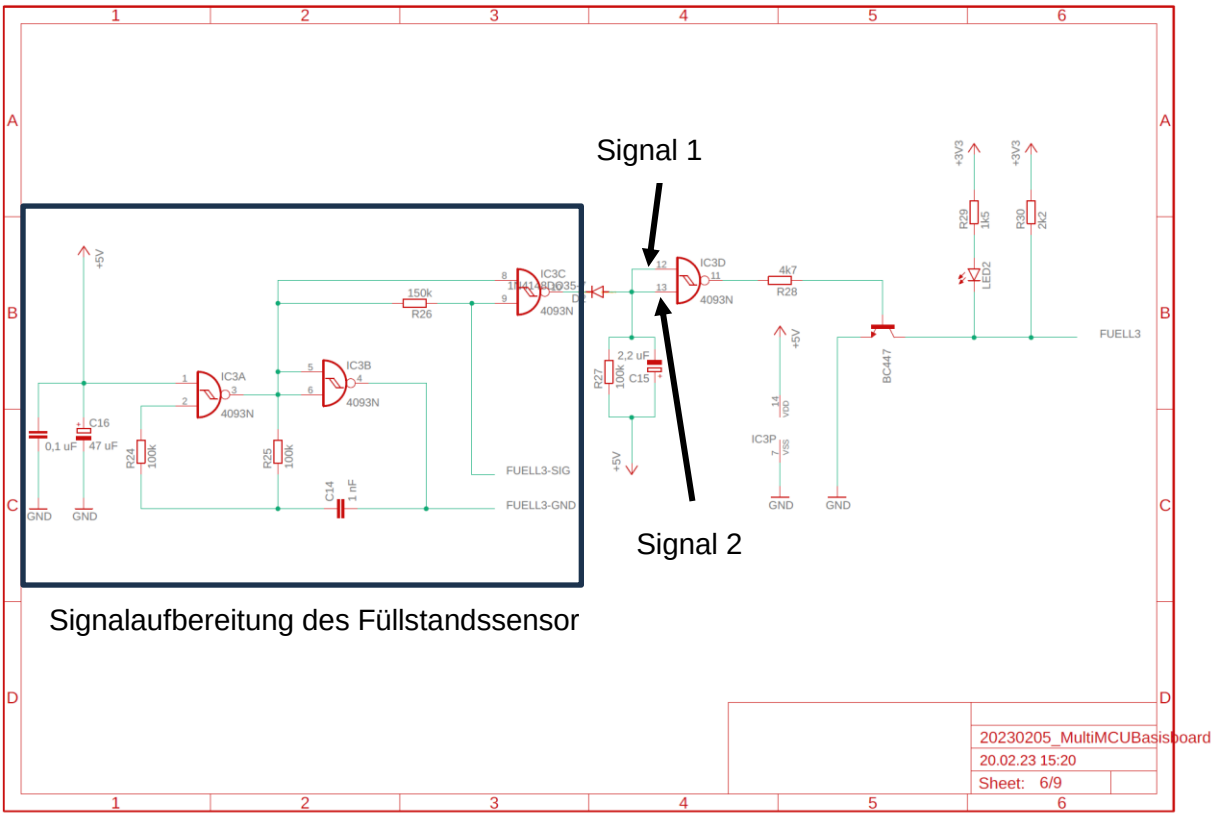


Anhang 2: Schaltplanauschnitt Messplatine (Spannungsregler, ADC, Operationsverstärker, Multiplexer) [xiv]



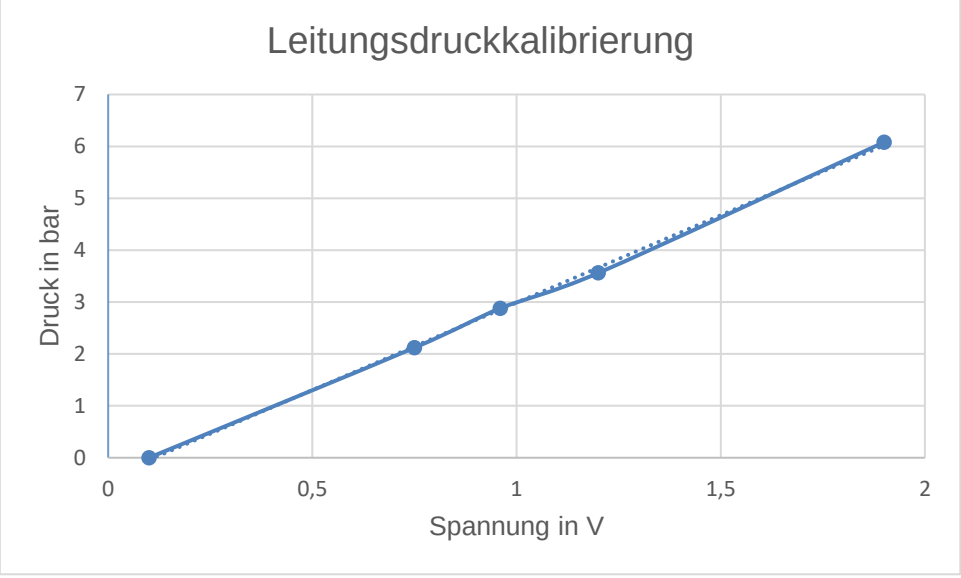
23.02.23 13:02 /Volumes/HOME_ARMIN/20_Auftraege_Projekte/04_Espressomaschine/07_Elektronik_u_Signalaufbereitung/11_20220514_Multi_MCU_Elektronik/01_Basisboard/2023

Anhang 3: Schaltplan Spannungsregler und Digital-Analog-Converter [xiv]

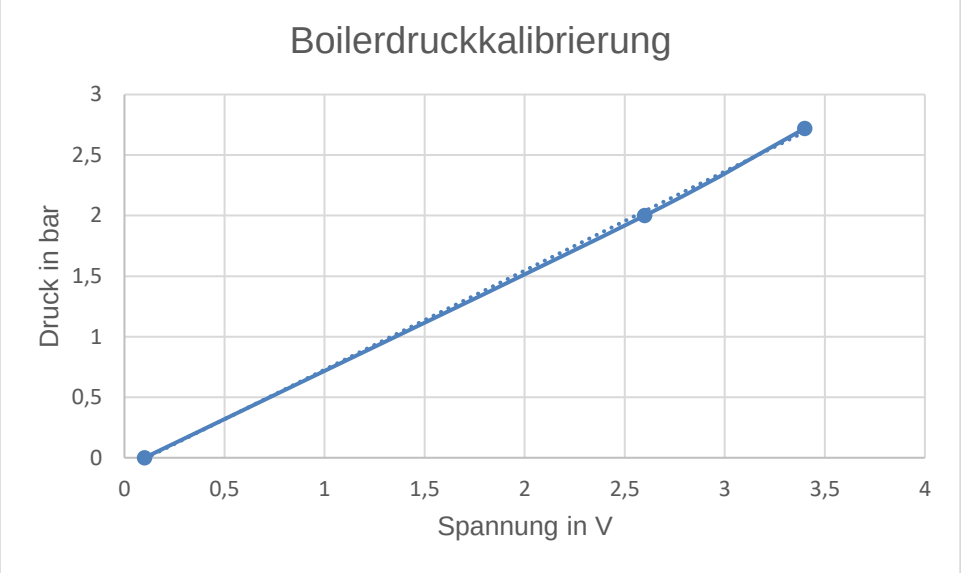


23.02.23 13:02 /Volumes/HOME_ARMIN/20_Auftraege_Projekte/04_Espressomaschine/07_Elektronik_u_Signalaufbereitung/11_20220514_Multi_MCU_Elektronik/01_Basisboard/2023

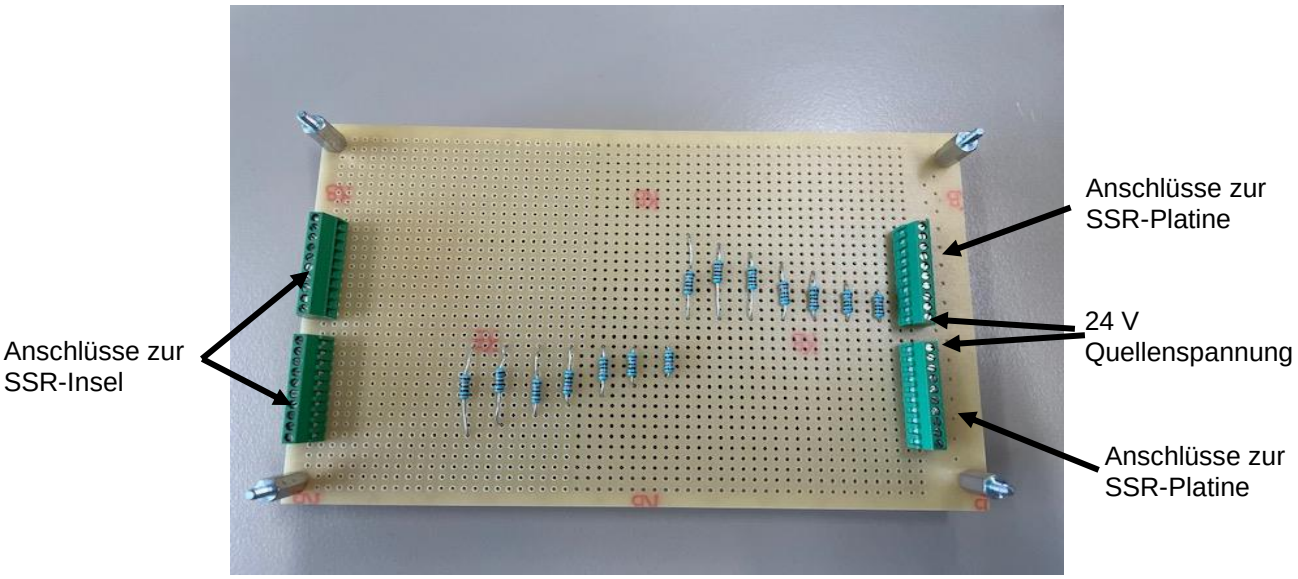
Anhang 4: Schaltplan Füllstand [xiv]



Anhang 5: Leitungsdruckkalibrierungskurve



Anhang 6: Boilerdruckkalibrierungskurve



Anhang 7: Zwischenplatine