

Mikrocontroller Programmierung in MicroPython

04.03.2026 Dipl.-Ing. Armin Rohnen LfbA



Agenda

- Top 1) Protokollführer
- Top 2) Vorstellungsrunde
 - Motivation zum Thema?
 - Lernziele Projektmodul?
- Top 3) Projekt / Projektteam
- Top 4) Projektstand
- Top 5) Regeltermin
- Top 6) Laboreinweisung (Optional)

Projekt - Lernziele

- Sich im Team organisieren
- Projekt im Team selbständig planen und realisieren
- Aufgabenstellungen interpretieren
- Daraus Entwicklungsthemen formulieren
- Komplexe Aufgabenstellung selbständig lösen

Wie wird ein Projekt erfolgreich?

- Klare Definition - wissen was man will / soll
- Fachkompetenz
- Zuverlässigkeit und Verbindlichkeit
- Vorausschauen
- Ehrlichkeit
- Respekt
- Team vor Einzelperson

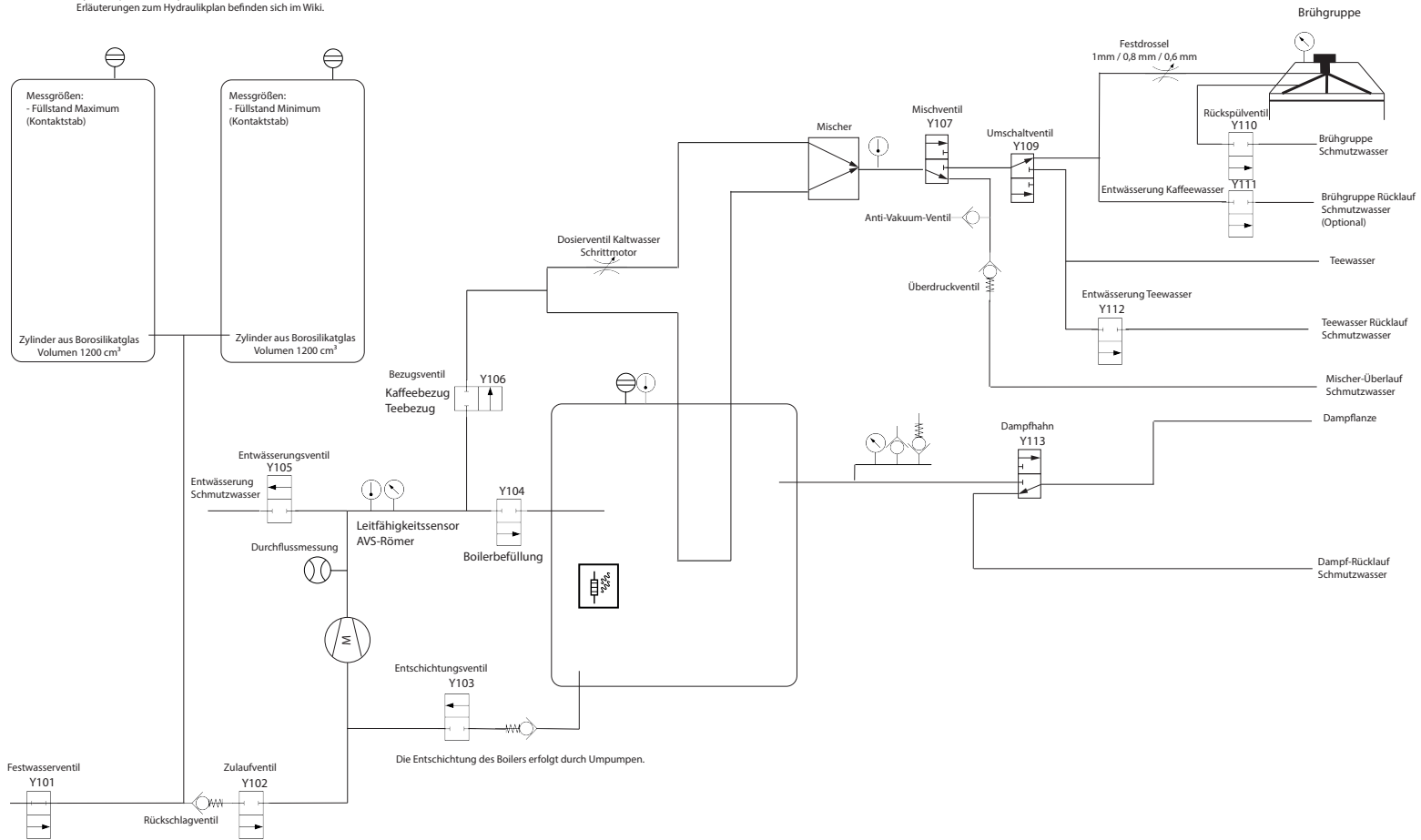
Wie wird ein Projekt erfolgreich? - Vorausschauen

Name:

Projektplanung KW (diese Woche)													
Samstag	h	Sonntag	h	Montag	h	Dienstag	h	Mittwoch	h	Donnerstag	h	Freitag	h
Projektplanung KW (nächste Woche)													
diese Woche			nächste Woche			diesen Monat			nächsten Monat				
Sollstunden	10			10									
Ist Plaung													
Ist Durchführung													
Differenz													

Projektstand

Hydraulikplan MMM Style und Labor - Stand: 24. November 2023
 Erläuterungen zum Hydraulikplan befinden sich im Wiki.



Projektstand - Maschinensimulator

- Hardware rudimentär aber funktional vorhanden
- Python-Code vorhanden
- Temperatursimulation Werte unstimmig
- Füllstandssimulation (Kurzschluss) funktional
- Flowmetersimulation fehlt
- Weitere Funktionalität erforderlich
 - Hardwareoptimierung
 - Aufheizsimulator
 - Einbindung in Kommunikationsring
 - MATLAB®-App für Bedienung

Projektstand - Füllstandsregler

- Ort: Basisplatine
- Boilerfüllstand und Tankfüllstand

- Wenn Tank-Minimum erreicht, dann bis Maximum befüllen
- Während Tankbefüllung keine anderen Aktivitäten mit Wasserbedarf

- Wenn Boilerfüllstand unterschritten, dann bis Füllstandssignal befüllen (etwas überfüllen)
- Wenn Boilerfüllstand unterschritten, dann keine Heizleistung

- Aufgabenanalyse vorhanden
- Code-Entwurf vorhanden

Projektstand - Boilerdruckregelung

- Ort: Messplatine
- PD-Regler-Code vorhanden in MATLAB® [114]
- Aufgabenanalyse vorhanden
- Aufheizphase zum Systemstart
- Entschichtung bis zu einstellbarer Grenztemperatur
- Stellgröße: Heizleistung über PWM-Sollwert an SSR-Platine
- Boilerfüllstand unterbindet Boilerdruckregler

Projektstand - Mischtemperaturregler

- Ort: Basisplatine (hinterfragen)
- Aufgabenanalyse vorhanden

- Regler-Code vorhanden in MATLAB® [114]
- Festwert und Temperaturverlauf ermöglichen
- Ankopplung an Durchflussregler (Kaskadenregelung)
- Stellgröße: Verstellung in Halbschritten an SSR-Platine

Projektstand - Durchflussregler

- Ort: Basisplatine
- Aufgabenanalyse vorhanden

- Regler-Code vorhanden in MATLAB® [114]
- Festwert und Durchflussverlauf ermöglichen
- Optional Preinfusion
- Ankopplung an Mischtemperaturregler (Kaskadenregelung)
- Stellgröße: Pumpensollwert 0 bis 5 V

Projektstand - Startprozedur

- Platinenstart: Messwertplatine
 - Initialisierung Kommunikation
 - Initialisierung Platine
 - Sendet 0-Token
 - Empfang Token
 - Startet Messwerterfassung
 - Sendet kontinuierlich Token
- Platinenstart: SSR-Platine
 - Initialisierung Kommunikation
 - Empfang Token
 - Senden Token
 - Keine Verarbeitung Token
- Platinenstart: Basisplatine, Display und Maschinensimulator offen

Projektstand - Startprozedur

- Funktionaler Maschinenstart
 - Tankbefüllung
 - Boilerbefüllung
 - Boileraufheizen
 - Bereitschaft Kaffee-/Teewasserbezug
 - Dampfbereitschaft
 - ...
- Keine Definition der Startprozedur vorhanden
- Teamaufgabe

Projektstand - UART-Kommunikation

- Kommunikation über UART
 - Messwertplatine (Taktgeber) -> Basisplatine -> SSR-Platine -> Display -> Messwertplatine
 - Sendeleitung (TX) der einen Platine an Empfangsleitung (RX) der nächsten Platine
 - UART-Library für alle Platinen gleich
- Token 74 Byte definiert (ggf. hinterfragen)
- Prüfsumme nicht umgesetzt (verlängert den Token)
- Paniktoken nicht umgesetzt
- Keine Fehlererkennung

Projektstand - Mehrkernnutzung

- Problem: Schrittmotorsteuerung blockiert Programmablauf
- MCUs RP2040 und RP2350 echte Dualcore ARM
- MicroPython uasyncio - nutzt sleep für Codeausführung
- MicroPython `_thread` nutzt zweiten Kern, aber Experimentell
- Erste erfolgreiche Versuche

Projektstand - Display

- 3 Displays zur Auswahl
 - Waveshare RP2350-Touch-LCD-2.1
 - Waveshare RP2350-Touch-LCD-2
 - Waveshare RP2350-Touch-LCD-1.69
- Jeweils mit Display-MCU-Platine und weiteren Sensoren
- Weitere MCU-PINs verfügbar
 - UART möglich
 - Dig in / Dig out möglich
 - ADC (12Bit) möglich
- Jeweils Demo-Code vorhanden
- Waveshare RP2350-Touch-LCD-2.1 allerdings in C

Projektstand - Display

- Vorhandene Aufgabenanalyse hinfällig
- Verknüpfung mit Vertikalhebel und zwei Tasten erforderlich
 - Vertikalhebel: Sollwerte, Handhebelsimulation, Spannungswert
 - Taste: Start Kaffeebezug / Teewasserbezug
 - Taste: Flush / Rückspülreinigung
- Touch mit x Flächen
 - 4 Kaffeeprogramme
 - Teewasser
 - Dampf
 - ... ?

Projektstand - Display

- Messwertanzeigen
 - Boilerdruck und Boilertemperatur
 - Kaffeebezugsdruck
 - Durchflussrate
 - Verlaufsdiagramme
- Statusmeldungen
- Log während Startprozedur
- Schrittweise Funktionsfreigabe
 - Teewasserbezug ab 100 °C Boilertemperatur
 - Kaffeebezug ab 110 °C Boilertemperatur
 - Dampfbezug ab Endwert Boilerdruck

Projektstand - STM32

- Zusammenführung Basisplatine, Messwertplatine und SSR-Platine
- Reduktion der Kommunikation über UART
- Revision Token
- Ggf. eigenständige Schrittmotorplatine erforderlich

- STM32H743ZI2 und STM32H753ZI
 - Unterschied: Kryptounterstützung (wird nicht benötigt)
 - Gleiches MicroPython-Derivat könnte funktionieren
 - 3 Stck. STM32H743ZI2 vorhanden (kurzfristig ausreichend)
 - 5 Stck. STM32H753ZI vorhanden, weitere möglich

Projektstand - STM32

- Stand 23.03.2026 MicroPython für STM32H753ZI verfügbar
 - Derivat mit ADC 16 Bit erstellen (Doku dazu vorhanden)
- Lauffähigkeit 16Bit-ADC MicroPython-Derivat des STM32H743ZI2 auf dem STM32H753ZI testen
- Nutzbare ADC-PINs
- Nutzbare UART PINs
- Nutzbare PWM-Pins bzw. MicroPython-Algorithmus für 8 Hz PWM-Signal mit 0 bis 100% einstellbaren Tastverhältnis
- Nutzbare Dig-Out-PINs
- Nutzbare Dig-In-PINs

- Kontaktleisten CN7, CN8, CN9 und CN10 betrachten
- Teststeuerung herstellen (Rohnen)
- PIN-Belegung für Zweizylinder Glasboiler definieren

Projektstand - Wartung APP

- MATLAB®-Wartungs APP
- Orientierung an vorhandener MATLAB®-APP zur Maschinensteuerung
- Gateway zwischen Steuerung und MATLAB®-APP undefiniert
- Konzept undefiniert
- Funktionsumfang undefiniert
- Bis funktionsfähiger Prototyp erforderlich