

Lab@Home – Eine OpenSource-Experimentierplattform

Prof.-Dr.-Ing Klaus M. Liebler¹

Zusammenfassung

Lab@Home ist eine Open-Source-Experimentierplattform für Laborpraktika in der Regelungstechnik, der Automationstechnik und der Informatik. Sie besteht aus einer elektronisch-mechanischen Baugruppe, einer webbasierten Experimentiersoftware sowie der Dokumentation verschiedener differenzierbarer Lernmodule. Die Bauteile für eine Einheit kosten unter 30€ und können innerhalb von 30min zusammengebaut werden. Für die Durchführung der Experimente stehen eine webbasierte Benutzerschnittstelle (WebApp) und eine Schnittstelle zu WinFACT Boris zur Verfügung. Lab@Home wird seit mehreren Semestern in verschiedenen Lehrveranstaltungen erfolgreich eingesetzt.

Stichwörter

Laborpraktikum, Experimentierplattform, Regelungstechnik, Gebäudeautomation, Informatik, IoT

1 Motivation und Ziel

Während der Corona-Pandemie waren Laborpraktika in ingenieurwissenschaftlichen Studiengängen teilweise nicht möglich. Gerade in Hochschulen der angewandten Wissenschaften (HAW) spielt dieser fachpraktische Teil der Lehrveranstaltungen eine entscheidende Rolle [1]. Sie stellen die im Studium erlernten Konzepte und Methoden für einen begrenzten Zeitraum in den Kontext der späteren beruflichen Realität. Für eine gute Aufnahme von Zusammenhängen ist ein möglichst viele Sinne ansprechendes Lernsetting erforderlich [2], was durch rein eine visuelle Darstellung von Experimenten (Simulation, Video) nicht gewährleistet ist. Das Projekt Lab@home entstand nun mit dem Ziel, für ingenieurwissenschaftliche Studierende der HAW eine Experimentierplattform zu schaffen. Versuche in der Regelungstechnik, der Automationstechnik und der Informatik sollten damit orts- und zeitunabhängig und ohne weitere Laborinfrastruktur (insbesondere: von zu Hause aus) durchgeführt werden können. Die Kosten und der Fertigungsaufwand müssen dabei so ausgelegt sein, dass jedem Studierenden einer Kohorte aus den üblichen Laborbudgets ein eigener Aufbau zur Verfügung gestellt werden kann.

2 Stand der Technik

Das Experiment gilt als ein guter Weg, um Studierenden die Verarbeitung und Verfestigung von Fakten- und Zusammenhangswissen zu erlauben und die Ausbildung ganzheitlicher Kompetenz zu ermöglichen [3]. Auch sog. Soft-Skills, wie Kreativität, Teamarbeit und die Fähigkeit zur textuellen und grafischen Dokumentation von Ergebnissen werden geschult.

¹ Prof. Dr.-Ing. Klaus Liebler, Fakultät Ingenieurwissenschaften und Informatik, Hochschule Osnabrück, Albrechtstraße 30, 49076 Osnabrück, k.liebler@hs-osnabrueck.de

Es existiert eine Vielzahl an Methoden, um Studierende in Laborpraktika sehr praxisorientiert anzuleiten. Im Folgenden sollen typische Aufbauten für Experimente im Bereich der Regelungstechnik vorgestellt, abgegrenzt und bewertet werden.

2.1 Experimentierstände von Laborausstattungen

Eine Vielzahl an Unternehmen bietet fertig entwickelte und didaktisch aufbereitete Experimentieraufbauten für den Einsatz in der Hochschullehre an. Als exemplarische Vertreter können hier genannt werden

- Das Übungssystem „Füllstandsregelung RT010“ der G.U.N.T GmbH [4]
- Die Anlage „Vorratsbehältersteuerung LC2010“ der hps SystemTechnik GmbH [5]
- Das Trainingssystem „IAC12 Füllstandsregelung“ der Lucas-Nülle GmbH [6]
- Der „Kompakttrainer Prozessregelung“ der LD Didactic GmbH [7]

Die Hersteller liefern in der Regel aufbereitete Lehr- und Lernmaterialien mit. Die Verwendung ist sicher und sorgt für zuverlässige Erfolgserlebnisse bei den Studierenden.

Die Bedienung der Aufbauten erfolgt über lokale Bedieneinrichtungen oder über Windows-Software. Es existieren teilweise Schnittstellen zu einschlägiger Drittsoftware, wie beispielsweise MATLAB/Simulink oder WinFACT BORIS. Bei Kosten im vier- oder fünfstelligen Bereich pro Einheit können im Regelfall nur wenige Exemplare beschafft werden.

2.2 Reine Simulationsexperimente

Simulationswerkzeuge, wie MATLAB/Simulink oder WinFACT BORIS ermöglichen es, Simulationen regelungs- oder steuerungstechnischer Realitäten mit grafischen Darstellungen zu hinterlegen. Die Simulationen werden häufig mit vereinfachten „perfekten“ Strecken umgesetzt [8, S. 13]. Herausforderungen aus der Praxis, wie beispielsweise Nichtlinearitäten, Messrauschen, Begrenzungen der Stellgrößen, bleiben unberücksichtigt. Zudem werden schlicht weniger Sinne beim Experimentieren angesprochen und die Intensität des Lehrerlebnisses nimmt ab [9].

2.3 Hochschulprojekte

Eine Vielzahl von Experimentieraufbauten für die Regelungstechnik wurde in den letzten Jahren an Hochschulen entwickelt. Oft war es das Ziel, im Kontext der neuen und neu entdeckten Lehrformate während der COVID19-Pandemie auch das Experimentieren von zu Hause aus zu ermöglichen [10].

Wang et al. [11] beschreiben den Aufbau der Experimentierplattform „Motor Control“ für die Regelungstechnik. Die Regelstrecke ist ein DC-Motor, der mit einem DC-Generator mechanisch verbunden ist. Beides wird über einen STM32F103-Mikrocontroller an eine PC-Software angekoppelt, die als Bedien- und Auswertungsoberfläche dient. Die Kosten pro Aufbau belaufen sich auf etwa 90US\$ pro Einheit.

Lab-Tec@Home [10] nutzt im Wesentlichen einen Arduino® UNO, eine LED-Matrix und einen lichtabhängigen Widerstand, um daraus eine Helligkeitsregelstrecke bzw. einen Helligkeitsregelkreis aufzubauen. Zur Streckenidentifikation und zum Test der Reglerauslegung kommt das Simulink® Support Package for Arduino® zum Einsatz. Eine Einheit kostet etwa 20US\$. Die Nutzung des Experimentieraufbaus und damit auch die Einführung eines komplett veränderten Lehrformates wurde einer Evaluation unterzogen, um die Effektivität des Lernens zu prüfen. Gemäß den dort erhobenen Kennzahlen wirkte sich der Einsatz des zu Hause nutzbaren Experimentiersystems signifikant positiv auf die Qualität der Lehrveranstaltung aus.

Rossiter et al. [12] stellen drei „Take Home“-Laborexperimente vor, die die Autoren des Papiers in ihren jeweiligen Lehrbereichen verwenden. Dabei ist das „Temperature Control Lab“ am ehesten vergleichbar mit dem hier beschriebenen Projekt. Es wurde für den Arduino® Uno ein einfaches Shield entwickelt, das aus zwei gesteuerten Leistungstransistoren auf Kühlkörpern mit jeweils einem Temperatursensor besteht. Es können damit Temperaturregelstrecken mit Wärmetransport auf Basis direkter Wärmeleitung, Konvektion und Strahlung untersucht werden. Zur Ansteuerung wird wahlweise MATLAB/Simulink mit dem Arduino Design Package, eine Python-Installation oder Java benötigt. Bemerkenswert ist, dass im Rahmen des Projektes wirklich sehr umfassende Lehr- und Lernmaterialien zur Verfügung gestellt werden.

2.4 Zusammenfassende Bewertung und Ableitung von Anforderungen

Zunächst ist aufgrund der Vielzahl entsprechender Veröffentlichungen – auch schon vor COVID19 – zu konstatieren, dass „at Home“-Lehr- und Lernumgebungen an Popularität gewinnen. Nach Meinung des Autors werden auch in postpandemischen Zeiten hybride Formen des Unterrichts nachgefragt werden. Präsenzformate werden bleiben und sind von ungemeiner Wichtigkeit für ein vertrautes Miteinander. Als sinnvolles Komplement dazu werden sich auch orts- und/oder zeitunabhängige Formate etablieren. Diese bieten häufig einen höheren wahrgenommenen Lernkomfort und lassen sich besser mit der Lebenswirklichkeit vieler Studierender vereinen.

Ziel des im Folgenden beschriebenen Konzeptes ist es, die Vorzüge der zuvor genannten Experimentierplattformen zu verbinden und gleichzeitig Nachteile zu vermeiden. Anforderungen sind:

1. Die Hardware soll kostengünstig auch in kleinen Stückzahlen produzierbar sein, damit für jeden Studierenden eine Einheit des Experimentiersystems angeschafft werden kann.
2. Die Experimentierplattform soll reale physische Experimente ermöglichen („es muss heiß werden, plätschern, rauchen und blinken“), um viele Sinne der Studierenden anzusprechen und einen echten Spieltrieb zu wecken.
3. Das System soll sich aus einem Webbrowser heraus bedienen lassen, um einerseits zusätzliche Lizenzkosten zu sparen und andererseits um auch Studierenden ohne klassischem Windows PC einen betriebssystemunabhängigen Zugang zu ermöglichen.
4. Ein fehlerträchtiges Zusammenstecken von Sensoren, Leistungstreibern und Aktoren mit einer Hauptplatine ist zu vermeiden, um Fehlfunktionen oder Zerstörungen von Systemkomponenten zu vermeiden.
5. Das Experimentiersystem soll verschiedene Versuche aus verschiedenen ingenieurwissenschaftlichen Domänen ermöglichen, um den Studierenden über einen längeren Zeitraum zu binden.
6. Auch wenn die Hardware des Arduino®-Projektes wegen Punkt (4) nicht unmittelbar zum Einsatz kommen soll, so soll zumindest eine Programmierung des Systems auf Basis der populären Arduino-Softwarebibliotheken mit der Arduino®-IDE oder vergleichbaren Integrierten Entwicklungsumgebungen möglich sein.
7. Auch wenn es wegen Punkt (3) nicht zwingend erforderlich sein darf, das System mit Drittsoftware zu betreiben, so soll dies bei Bedarf aber möglich sein.
8. Neben der Hard- und Software sind Lehr- und Lernmaterialien bereit zu stellen, damit das Projekt von Lehrenden und Lernenden einfach genutzt und in Betrieb genommen werden kann.
9. Alle Projektressourcen sollen unter einer permissiblen Lizenz bereit gestellt werden, damit eine Verbreitung und Weiterentwicklung möglich ist.

3 Umsetzung

Lab@Home soll nun all diesen Anforderungen gerecht werden und besteht aus drei eng miteinander verzahnten Bausteinen:

- Einer mechatronischen Baugruppe
- Einer Software mit webbasierter Benutzerschnittstelle
- Einem Lehrkonzept.

Alle Bausteine wurden unter einer permissiblen Lizenz veröffentlicht. Die Designdokumente stehen frei auf der Github-Seite des Projektes [13] zu Einsicht, Verwendung und Verbesserung zu Verfügung. Im Folgenden sollen ausgewählte Besonderheiten der Umsetzung dargestellt werden.

3.1 Mechatronische Architektur

Elektronischer und gleichzeitig mechanischer Kern von Lab@Home ist eine quadratische zweilagige Leiterplatte mit 10cm Kantenlänge und 1,6mm Dicke. Diese lässt sich bei üblichen PCBA-Anbietern günstig fertigen und weist eine hohe Stabilität auf. Über Abstandhalter, Montagewinkel und Klebeflächen werden Heizwiderstand, Lüfter und Lausprecher montiert. Auch die Frontplatte ist letztlich eine Leiterplatte, weil diese zu geringen Kosten (<1€ pro Stück) gefräst, gebohrt und bedruckt werden kann. Die Frontplatte wird über Distanzbolzen im Abstand von 6mm über der Hauptplatine befestigt (siehe Abbildung 1).

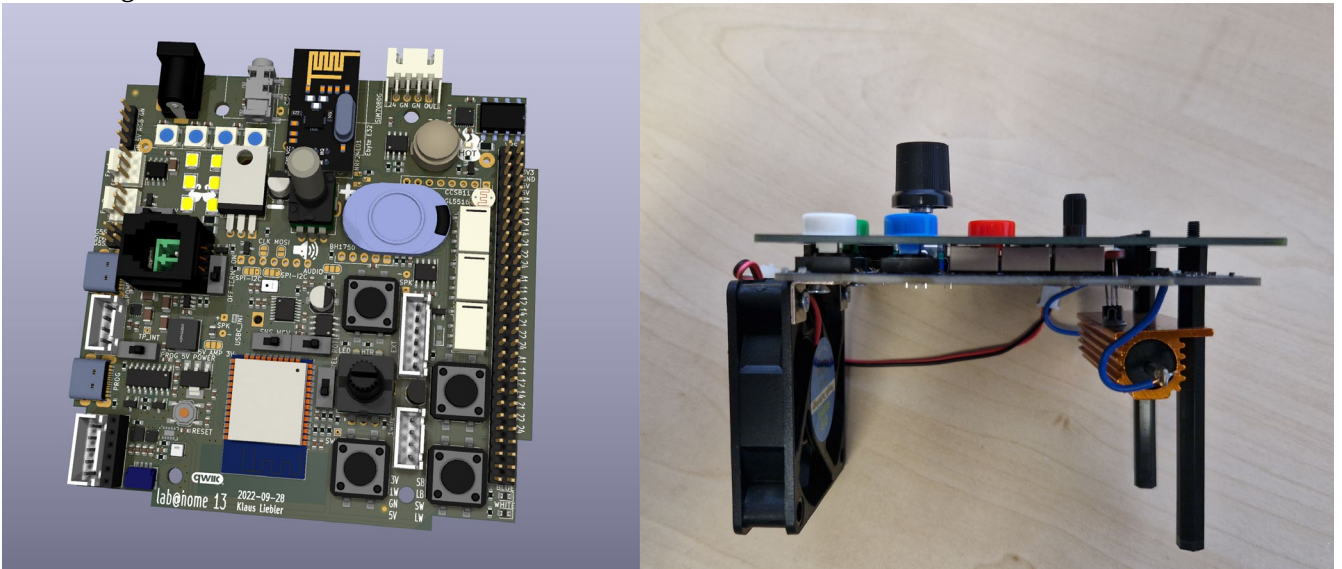


Abbildung 1: Links: Rendering der Hauptplatine; Rechts: Foto der realen Baugruppe (Seitenansicht, Ventilator, Heizwiderstand und Frontplatte montiert)

Die elektronische Schaltung gruppiert sich um einen Mikrocontroller des Typs Espressif ESP32. Dieser zeichnet sich durch eine vergleichsweise hohe Leistung, flexible und vielfältige Schnittstellen und einen günstigen Preis aus. Hervorzuheben ist außerdem, dass der ESP32 eine WLAN-Schnittstelle integriert, die sowohl im Accesspoint- als im Station-Modus betrieben werden kann. Abbildung 2 zeigt das Schaltungs-Blockdiagramm von Lab@Home. Bei der danach folgenden Beschreibung der Schaltung werden die zur Umsetzung verwendeten integrierten Schaltkreise in Klammern angegeben. Wichtige Kriterien bei der grundsätzlichen Bauteileauswahl waren eine gute Verfügbarkeit, die leichte Austauschbarkeit mit alternativen Bauteilen, ein hoher Verbreitungsgrad in der (Arduino-) Community und nicht zuletzt ein günstiger Preis.

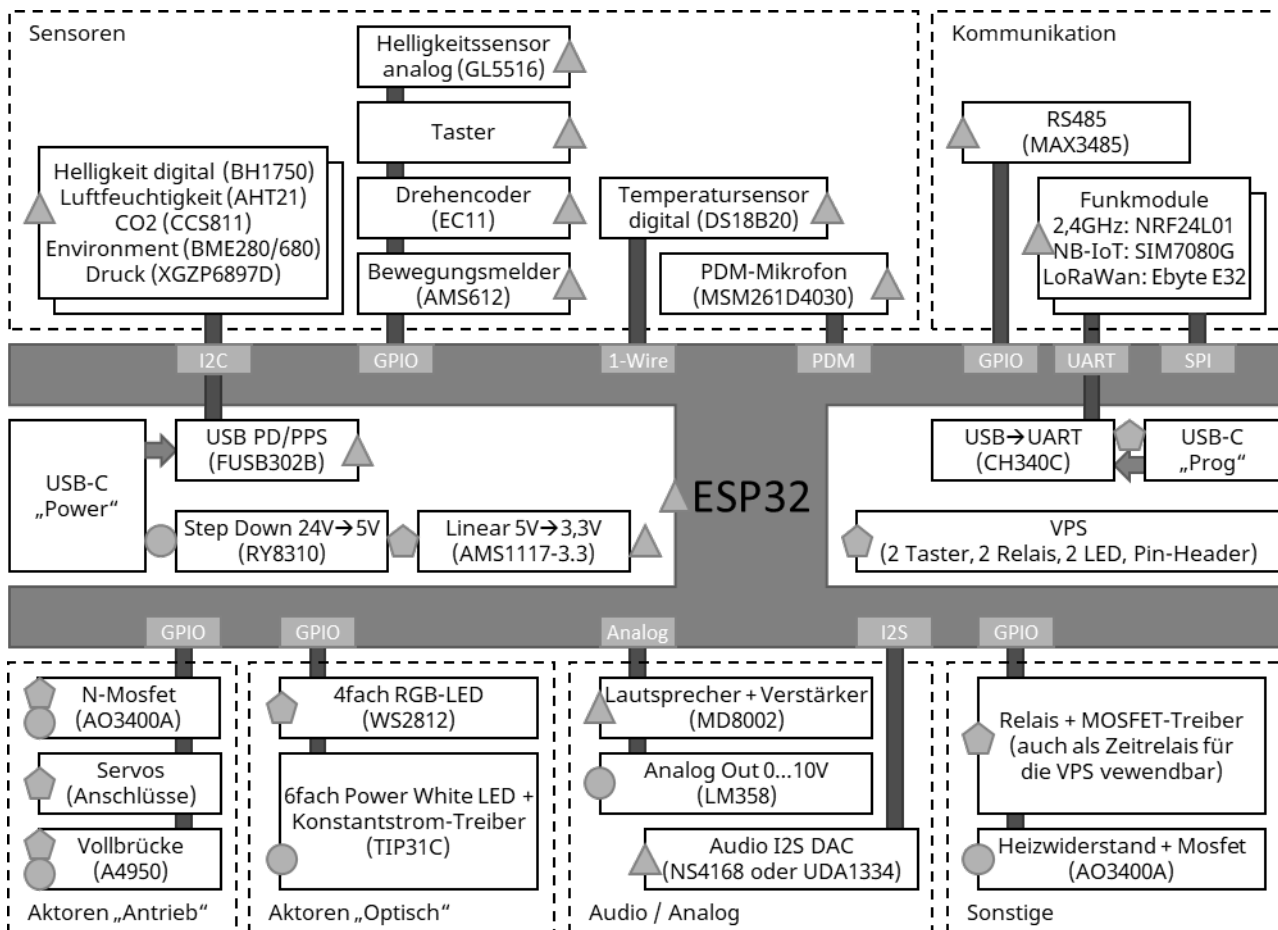


Abbildung 2: Blockdiagramm Hardware (Dreieck = Spannungsschiene 3,3V, Fünfeck=5V, Kreis=20...24V)

Die gesamte Schaltung kann von einem USB PD Netzteil (20V, min. 2A) versorgt werden. Diese werden heute üblicherweise für leistungsfähige Tablets oder Notebooks eingesetzt. Viele potenzielle Nutzer von Lab@Home verfügen also bereits über die passende Stromversorgung. Sofern das Netzteil den PPS-Modus beherrscht, kann Lab@Home auch diesen nutzen und eine in weiten Bereichen regelbare Spannungsquelle / Stromquelle realisieren. Die Umsetzung des USB-PD- bzw. USB-PPS-Protokolls wurde hauptsächlich in Software realisiert, hardwareseitig ist nur ein einfacher PHY (FUSB302b) erforderlich. Alternativ ist die Versorgung mit einer 18-30V-Gleichspannung über eine Klinkenbuchse möglich. Ein synchroner getakteter Abwärtswandler (RY8310) erzeugt daraus 5.0V. Ein Linearregler (AMS1117-3.3) reduziert dies weiter auf 3,3V.

Sofern weder der Heizwiderstand noch die Hochleistungs-LED benötigt werden und auch kein externer 24V-Verbraucher angeschlossen ist, genügt die 5V-Versorgung über die USB-C-Programmierschnittstelle, in der ein USB-Seriell-Wandler (CH340) zum Einsatz kommt.

Für die Audio-Eingabe steht ein MEMS-Mikrofon mit PDM-Schnittstelle (MSM261D4030H1CPM) zur Verfügung. Im Bereich der Audio-Ausgabe finden sich einige Bestückungsalternativen. Der Standardweg beginnt integrierten Digital-Analog-Converter (DAC) des ESP32 und führt über einen Verstärker-IC (MD8002) auf einen Lautsprecher. Die Audio-Qualität ist hierbei gering, weil der DAC eine geringe Auflösung von 8bit hat. Optional ist deshalb auch die Bestückung von Verstärker-ICs mit I2S-Eingang und einem integrierten hochwertigen DAC möglich, die entweder einen Lautsprecher (via NS4168) oder einen Line-Out-Klinkenanschluss (via UDA1334) ansteuern.

Der DAC-Ausgang führt parallel an einen Operationsverstärker (LM358, Verstärkung 3,2), um einen 0-10V-Ausgang zu realisieren. Die Konstantstromquelle für die Hochleistungs-LED wurde diskret mit Bipolar-Transistoren aufgebaut (2N2222+TIP31C). Daneben finden sich vier einzeln ansteuerbare RGB-LED (WS2812). Zur Ansteuerung des Heizwiderstandes und des Ventilators kommen MOSFETs

(AO3400A) zum Einsatz. Ein Vollbrückentreiber (A4950) ermöglicht die Kontrolle eines einfachen DC-Motors. Im Bereich der Kommunikation existiert neben den im ESP32 integrierten Drahtlosschnittstellen eine RS-485-Schnittstelle (SP3485EN). Optional können Module für 2,4GHz-Kommunikation (NRF24L01, im Rendering oben dargestellt), NB-IoT (SIM7080G) oder LoRaWan (EByte E32) bestückt werden.

Lab@Home kann seine Umgebung mit einer Vielzahl von Sensoren wahrnehmen. Zur Erkennung von Bewegung ist ein PIR-Sensor (BM612) vorgesehen. Auch die aktuelle Umgebungstemperatur (DS18B20) und die Luftfeuchtigkeit (AHT21) können erfasst werden. Ein analoger Helligkeitssensor (GL5516) ist ebenfalls stets verbaut. Bestückungsoptionen existieren für einen CO₂-Sensor (CCS811, sowohl als Chip als auch als steckbares Modul), einen kompletten sog. „Environment“-Sensor (BME280/BME680), einen digitalen Helligkeitssensor (BH1750), einen Differenzdrucksensor (XGZP6897D) und einen 6DOF-Sensor (MPU6050, Gyroskop & Beschleunigung).

Direkt auf der Platine befinden sich zwei Regelstrecken

- Temperaturregelstrecke (näherungsweise P-Tn-Verhalten), bestehend aus dem steuerbaren Heizwiderstand, einem daran montierten Temperatursensor (DS18B20) und dem Ventilator zur Erzeugung einer Störung
- Helligkeitsregelstrecke (näherungsweise P-Verhalten), bestehen aus den steuerbaren Hochleistungs-LED und dem Helligkeitssensor.

Schließlich befinden sich zur Umsetzung einfacher sog. verbindungsprogrammierbarer Steuerungen (VPS) zwei Relais, zwei Taster und zwei LED auf der Platine. Ein drittes Relais wird unmittelbar vom Microcontroller gesteuert und kann via Software ein Zeitrelais simulieren. Alle Anschlüsse sind auch auf eine zweireihige Pfostenleiste geführt. Die konkrete Verschaltung lässt sich dadurch einfach mit sog. Jumper-Kabeln definieren.

Beim Schaltungsdesign war es besonders herausfordernd, mit den wenigen Pins des ESP32 viele Funktionen zu realisieren. Die folgenden Techniken wurden angewandt:

- Alle Sensoren verwenden den I2C-Bus.
- Alle Taster sind über ein Widerstandsnetzwerk an einen analogen Eingang angeschlossen.
- Ein Impulseingang des Drehencoders und der analoge Helligkeitssensor teilen sich einen Pin. Wenn dieser Pin binär abgefragt wird, detektiert er nur die Drehimpulse des Encoders, bei analoger Abfrage kann die Helligkeit gemessen werden.
- Einige Signale, deren gemeinsame Nutzung unwahrscheinlich ist, können mechanisch umgeschaltet werden.

Beim Design erwies es sich als vorteilhaft, dass der ESP32 eine äußerst flexible GPIO-Matrix hat. Bei der konkreten Zuordnung von internen Signalen zu physischen Pins konnte so oft auch die Platzierung der Komponenten auf der Platine berücksichtigt werden, um die Voraussetzungen für einen nur zweilagigen und damit günstigen Platinenaufbau zu schaffen. Für den Schaltungsentwurf und das Platinenlayout kam die Software KiCad in der Version 6 zum Einsatz. Hierfür stehen auch Projektdaten zur Verfügung. Die Auslegung der mittels PWM ansteuerbaren Konstantstromquelle für die Hochleistungs-LED wurde vorab per Handrechnung ermittelt und simulativ mit LTSpice abgesichert.

Die Kosten setzen sich aus den Herstellungs- und Lieferkosten der SMD-bestückten Platine einerseits und den Material- und Lieferkosten für die nachträglich zu bestückenden Bauteile zusammen. Beim Referenzanbieter JLCPCB und einer Bestellmenge von 25 Platinen kostet eine SMD-bestückte Platine rechnerisch knapp 20€ DDP². Alle weiteren Kosten (DDP) können der folgenden Tabelle entnommen werden. Ohne optionale Module liegen die reinen Materialkosten also bei unter 30€ pro Einheit.

² DDP ist eine übliche Ankunfts-klausel bei Handelsbeziehungen und steht für „geliefert sowie Zoll und Steuer bezahlt“ (engl: Delivered Duty Paid)

Typ	Kosten (DDP)
Hauptplatine bestückt	19,31 €
Lüfter 6015	1,47 €
Widerstand 100W, 22R	1,42 €
PIR-Sensor BM612	0,98 €
Frontplatte gefräst gedruckt	0,92 €
Winkel 10mm	0,60 €
Abstandshalter 50mm	0,29 €
Lautsprecher 2415	0,28 €
Drehencoder EC11	0,27 €
Temperatursensor DS18B20	0,27 €
RJ11 Buchse	0,24 €
Weitere Bauteile in Summe	1,31 €
Summe	27,54 €

3.2 Softwarearchitektur

Die Software gliedert sich in die Firmware und die Webapplikation. Die Firmware wurde in C++ 17 entwickelt, baut auf dem ESP-IDF 4.4.2 und dem Echtzeit-Betriebssystem FreeRTOS [14] auf und folgt klassischen Gestaltungsprinzipien (siehe Abbildung 3). Eine Hardware-Abstraktionsebene (HAL) macht unterschiedliche Versionen und Ausbaustufen der Elektronik untereinander kompatibel und schafft eine normalisierte Umgebung für den sog. DeviceManager. Dieser fungiert als zentrale Management-Instanz des Systems und führt analog einer SPS einen immerwährenden Zyklus aus. In dieser Hauptschleife findet im Wesentlichen die Unterscheidung des grundsätzlichen Betriebsmodus statt. Über diesen Betriebsmodus wird im Wesentlichen festgelegt, welche Systemkomponente schreibenden Zugriff auf die Aktorik von Lab@Home hat. Unterschieden wird zwischen:

- Abarbeitung des Funktionsblock-Diagramms
- Experiment „Temperaturregelkreis“, gesteuert von der Webapp über den Webserver
- Experiment „Helligkeitsregelkreis“, gesteuert von der Webapp über den Webserver
- Steuerung über den UDP Connector

Unmittelbar nach dem Einschalten befindet sich das System im Modus „Abarbeitung des Funktionsblock-Diagramms“. Sofern ein Anwender die Webapplikation öffnet und dort auf eine der Experimentierseiten wechselt oder alternativ über ein UDP-Protokoll auf das System zugreift, findet ein Wechsel in den entsprechen Modus statt. Sowohl das UDP-Protokoll als auch die Experimentierseiten führen im Regelbetrieb einen ständigen Datenaustausch mit Lab@Home durch. Wenn nun für 3 Sekunden keine Kommunikation stattfindet, wechselt das System wieder in die Abarbeitung des Funktionsblock-Diagramms. Für den Heizwiderstand gibt es in allen Modi eine Sicherheitsabschaltung bei 85°C.

In aller Kürze soll noch auf einige weitere Module eingegangen werden. Gemäß den Ausführungen in [15, 265ff] wurde ein zeitdiskreter PID-T1-Regler implementiert, der sowohl bei den Experimenten als auch in einem PID-Funktionsblock verwendet wird. Die Signale, die vom integrierten Mikrofon aufgenommen werden, werden ständig einer FFT-Analyse zugeführt und das Spektrum einer Webseite zur Verfügung gestellt. Ein MP3-Player auf Basis des minimp3-Projektes arbeitet sehr ressourcenschonend und übernimmt effizient die Codierung von MP3-Dateien.

Lab@Home stellt seine primäre Bedienoberfläche – eine Single Page Webapp – über einen Webserver zur Verfügung, die betriebssystemunabhängig und ohne weitere Softwareinstallation im Browser genutzt werden kann. Die Webapp wurde vollständig in HTML, SVG, TypeScript und SASS entwickelt. Auf jegliche Toolkits, Frameworks oder Hilfsbibliotheken (jquery, Vue, Angular etc.) wurde verzichtet. Stattdessen werden aktuelle Varianten des HTML5- und des DOM-Standards vorausgesetzt und

verwendet. Ein vollständig automatisierter Build-Prozess führt zunächst eine Translation von TypeScript zu JavaScript und von SASS zu CSS durch. Im Anschluss werden alle Ressourcen (JavaScript, CSS, SVG, PNG) in ein zentrales HTML-Dokument „inline“ eingefügt. Die gesamte Webapp besteht dann tatsächlich aus einer einzigen HTML-Datei, die nach GZIP-Komprimierung nur noch knapp 100kB groß ist. Im Rahmen des Build-Prozesses der Firmware wird sie direkt binär in das Firmware-Image eingebunden.

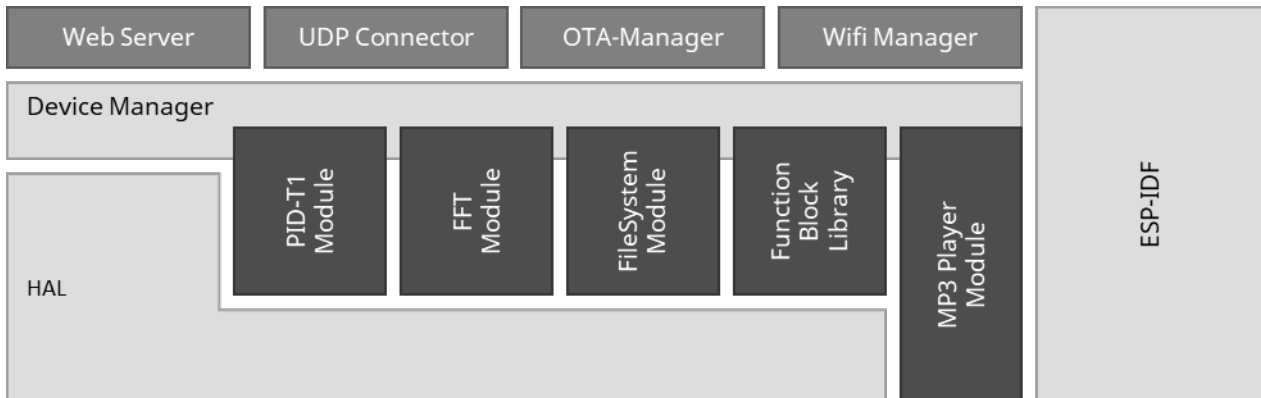


Abbildung 3: Softwarearchitektur

Funktionell hervorzuheben ist dabei sicherlich die Integrierte Entwicklungsumgebung für die Funktionsbausteinsprache in Anlehnung an die DIN EN 61131 ([16]) (siehe Abbildung 4 oben). Aus einer erweiterbaren Bausteinbibliothek zieht der Nutzer Funktionsbausteine auf die Arbeitsfläche und verbindet diese zu einem vollständigen Automationsprogramm. Neben klassischen Logikgattern stehen Schaltwerke und viele Bausteine der VDI3814 ([17]) zur Verfügung. Ein- und Ausgabe-Blöcke implementieren den Zugriff auf Sensoren, Aktoren und ausgewählte Kommunikationsprotokolle. Die Entwicklungsumgebung unterstützt interaktives Debugging sowie einen Simulationsmodus. Der Bytecode-Compiler zur Erstellung eines lauffähigen Programms wurde in TypeScript implementiert, läuft vollständig im Browser und sendet sein Kompilat online an die dazu passende Laufzeitumgebung in der Firmware. Das Laden und die Initialisierung einer neuen Programmversion erfolgt asynchron, während die vorhergehende Version noch aktiv arbeitet. Wenn das neue Programm dann bereit ist, geschieht die Umschaltung durch schnelle Zeigerumschaltung innerhalb eines einzigen Abarbeitungszyklus der Laufzeitumgebung. Das Prinzip gleicht dem eines Doppelkupplungsgetriebes, bei dem kein Leistungsabfall mit dem Umschalten einhergeht.

Sowohl für die Temperaturregelstrecke als auch für die Helligkeitsregelstrecke existieren Experimentierseiten, die sowohl eine Streckenidentifikation als auch eine experimentelle Evaluation von Reglerparametern (ClosedLoop-Modus, Abbildung 4 Mitte) ermöglichen. Sowohl die Grafik oben als auch die Tabelle unten werden im Sekundenrhythmus live aktualisiert. Ein einfacher Datenexport nach Excel oder MATLAB zu Auswertungszwecken ist möglich.

Schließlich bietet die Webapp noch einen Bereich für die Darstellung der FFT-Auswertung des Signals am Mikrofon (siehe Abbildung 4 unten). Abgetastet wird mit einer Frequenz von 11025Hz und es wird eine 2048-Punkt-FFT durchgeführt. Die 1024 Spektrallinien werden zu 64 Frequenzbändern zusammengefasst. Je höher die Frequenz, desto breiter das Band. Die Amplitude eines Bandes ergibt sich aus dem Maximum der in ihm enthaltenen Spektrallinien. Das Spektrum wird in der Webapp sekundlich aktualisiert. Eine Komfortfunktion ermöglicht es, das gegenwärtige Spektrum „einzufrieren und in den Hintergrund zu legen“. So kann es mit späteren Spektren einfach grafisch verglichen werden.

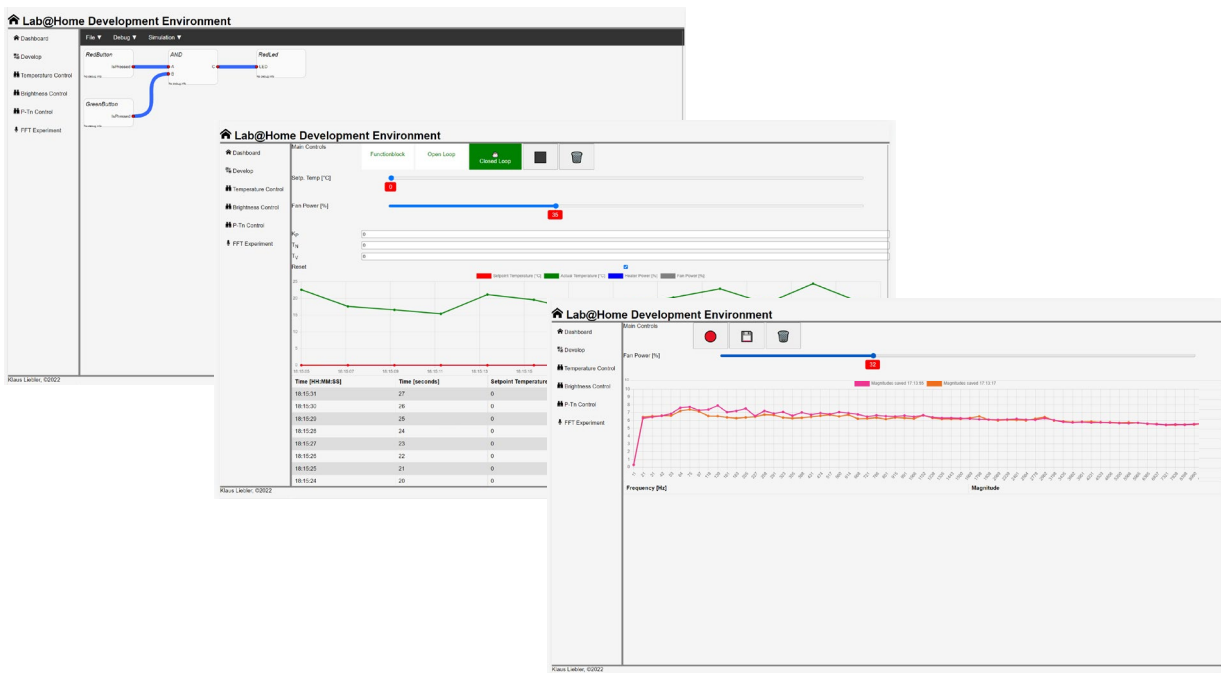


Abbildung 4: Die webbasierte Bedienoberfläche ("Webapp") von Lab@Home

3.3 Aufbau und Inbetriebnahme

Selbst aufgebauten Produkten wird häufig eine hohe Wertschätzung zu teil und die damit erzielbaren Arbeitsergebnisse sind besser [18]. Diese Eigenschaft soll auch in Verbindung mit Lab@Home genutzt werden. Als Grundlage des Aufbaus dient eine Platine, auf der alle oberflächenmontierten Bauteile bereits bestückt und verlötet sind. Aufgabe der Studierenden ist es nun, eher größere Bauteile zur Durchsteckmontage zu bestücken und zu verlöten. Außerdem sind Lüfter, Heizwiderstand, Frontplatte und Bedienknöpfe zu montieren. In einem etwa 40minütigen Video werden alle Arbeiten in Echtzeit erläutert und durchgeführt. Mit Ausnahme des PIR-Sensors und eines Leistungstransistors handelt es sich um unempfindliche mechanische Bauteile. In der Praxis hat sich gezeigt, dass der Aufbau für die Studierenden kein Hindernis darstellt. Die initiale Basis-Firmware kann über die USB-Schnittstelle webbasiert mit Hilfe der ESP-Web-Tools (<https://esphome.github.io/esp-web-tools/>) installiert werden. Weitere Updates kommen dann über einen „Over-the-Air“-Mechanismus. Ein sog. Wifi-Manager ermöglicht es, Zugangsdaten für ein lokales WLAN-Netzwerk einzugeben, mit dem sich Lab@Home dann verbindet.

4 Anwendung

Im Folgenden sollen aus allen oben genannten Fächern typische Praktikumsaufgaben mit Lab@Home skizziert werden.

4.1 Regelungstechnik – Analyse einer Regelstrecke und Parametrierung eines PID-Reglers

In der Regelungstechnik ist Lab@Home in mehreren Praktika im Einsatz und deckt den inhaltlichen Spannungsbogen einer Bachelor-Veranstaltung ab. Die Temperaturregelstrecke kann einer experimentellen statischen (\rightarrow Kennlinien) und dynamischen (\rightarrow Sprungantwort) Analyse unterzogen

werden. Das thermische Verhalten kann auch (näherungsweise) mit Differenzialgleichungen beschrieben werden. Es schließen sich Verfahren zur qualitativen und quantitativen (→Wendetangentenverfahren, [15, S. 141]) Identifikation an. Schließlich können die Studierenden Regler mit den üblichen „Faustformelverfahren“ ([19], [20]) entwerfen und die Regelgüte anhand der bekannten Kennzahlen [21, S. 94] bestimmen.

Als Differenzierung für besonders interessierte Studierende wird auf die Effekte des integrierten Anti-Windup und des tatsächlich implementieren digitalen PID-T1 eingegangen.

4.2 Steuerungstechnik – MaxComfortMinEnergy

Die Studierenden entwickeln eine Licht- und Lüfter-Steuerung für einen WC-Bereich mit Tageslichteinfall. Dabei soll der Bewegungsmelder, der Helligkeitssensor, die LEDs und der Ventilator verwendet werden. Beim Betreten soll der Lüfter mit Nachlauf und die Beleuchtung in angemessener Helligkeit aktiviert werden. Als Differenzierung für interessierte Studierende bietet es sich an, die abgestrahlte Lichtfarbe der Tageszeit anzupassen („Human Centric Lighting“, [22]).

4.3 Mathematik, Operations Research – Condition Monitoring und Predictive Maintenance einer Lüftungsanlage

Ein kleiner Streifen Gewebeklebeband auf einem Lüfterflügel versetzt den Ventilator von Lab@Home in Unwucht und simuliert damit einen Lagerschaden. Die Veränderung im Geräusch kann bei genauem Hinhören wahrgenommen werden. In der Spektralanalyse des vom Mikrofon aufgenommenen Audiosignals macht sich die Unwucht deutlich bemerkbar. Die Studierenden werden so mit einer sehr praktischen Anwendung der schnellen Fouriertransformation vertraut gemacht, nämlich des automatisierten sensorbasierten Condition Monitorings mit dem Ziel der vorbeugenden Instandhaltung von Anlagen.

4.4 Informatik – Die barrierefreie IoT-Behaglichkeitsampel

Lab@Home wird ohne die Standard-Firmware genutzt. Aufgabe der Studierenden ist es, ein C++-Codebeispiel zu komplettieren, das Lab@Home zum IoT-Sensor mit integrierter barrierefreier Behaglichkeitsampel macht. Das Programm liest zyklisch die Messwerte verschiedener Sensoren ein und sendet diese an einen MQTT-Broker. Die Messwerte werden dann auf einem NodeRed-basierten Dashboard dargestellt. Lokal zeigen die RGB-LEDs die CO₂-Konzentration an. Das Muster ist auch für Personen mit Rot-Grün-Schwäche erkennbar. Zusätzlich warnen Alarmtöne und eine synthetische Stimme auch stark sehingeschränkte Personen. Als Differenzierung für besonders interessierte Studierende können beim Überschreiten von Grenzwerten von NodeRed Nachrichten versendet werden. Außerdem bietet es sich an, für die Datenauswertung leistungsfähige Tools und Datenbanken, beispielsweise Grafana und InfluxDB zu verwenden.

4.5 Projektarbeiten

Lab@Home bildet bereits zum dritten Mal die Basis von projektbasierten Arbeiten im Master-Studiengang „Gebäudeautomation“ der Hochschulen Biberach und Münster. Die Studierenden entwickeln dort in kleinen Teams innovative IoT-Lösungen. Eine Gruppe realisierte beispielsweise eine Bewässerungssteuerung für Zimmerpflanzen. Einzelne Blumentöpfe – jeweils mit Lab@Home ausgestattet – konnten von zentralen Reservoirs für Wasser und Düngemittel einen Vordruck auf den Versorgungsleitungen anfordern und dann den lokalen Wasser- und Düngerzufluss über Magnetventile freigeben. Es kamen dabei u.a. NPK-Sensoren mit Modbus-RTU-Schnittstelle zum Einsatz.

5 Fazit und Ausblick

Lab@Home stellt eine kostengünstige Experimentierplattform für verschiedene Disziplinen innerhalb der Ingenieurwissenschaften dar. Ein Teilesatz lässt sich für unter 30€ zusammenstellen und innerhalb von 45min zu einer vollfunktionsfähigen Lösung aufbauen. Verschiedenste Experimente aus der Regelungstechnik und der Steuerungstechnik lassen sich sehr real und anschaulich gestalten. Studierende können die Experimente wahlweise über einen Webbrowser, MATLAB/Simulink oder WinFACT BORIS betreiben. Nach dem einmaligen Zusammenbau und der Inbetriebnahme ist die Baugruppe ohne weitere Vorrüstungen einsetzbar. Neben der Nutzung über eine Oberfläche ist eine freie Programmierung in C++ mit Arduino®-Bibliotheken möglich. Der im Projekt verwendete Mikrocontroller wird mittlerweile auch vom OpenPLC-Projekt [23][24] unterstützt. Alle Projektressourcen sind unter der MIT-Lizenz veröffentlicht.

6 Literatur

- [1] A. E. Tekkaya, U. Wilkesmann, C. Terkowsky, C. Pleul, M. Radtke und F. Maevus, Hg. *Das Labor in der ingenieurwissenschaftlichen Ausbildung: Zukunftsorientierte Ansätze aus dem Projekt IngLab : acatech Studie* (Acatech Studie). München: Herbert Utz Verlag GmbH, 2016.
- [2] M. S. Gleßner, S. Knutzen und P. Salden, *Die Spannung steigern : Laborpraktika didaktisch gestalten*. TUHH Universitätsbibliothek, 2015. [Online]. Verfügbar unter: <https://tore.tuhh.de/handle/11420/1280>
- [3] L. D. Feisel und A. J. Rosa, “The Role of the Laboratory in Undergraduate Engineering Education,” *Journal of Engineering Education*, Jg. 94, Nr. 1, S. 121–130, 2005, doi: 10.1002/j.2168-9830.2005.tb00833.x.
- [4] G.U.N.T. Gerätebau GmbH, Hg., “RT010 Übungssystem Füllstandsregelung,” 2021. Zugriff am: 4. Dezember 2022. [Online]. Verfügbar unter: https://www.gunt.de/images/datasheet/1042/RT-010-UEbungssystem-Fuellstandsregelung-HSI-gunt-1042-pdf_1_de-DE.pdf
- [5] hps SystemTechnik Lehr- + Lernmittel GmbH, Hg., “Praktikumsanlage LC2010,” Zugriff am: 4. Dezember 2022. [Online]. Verfügbar unter: <https://hps-systemtechnik.com/wp-content/uploads/content/deutsch/LC2010.pdf>
- [6] Lucas-Nülle GmbH, Hg., “Mess- und Regelungstechnik: Trainingssysteme für die Ausbildung von Facharbeitern und Ingenieuren,” 2020. Zugriff am: 4. Dezember 2022. [Online]. Verfügbar unter: <https://print.lucas-nuelle.de/ln/download/4143>
- [7] LD DIDACTIC GmbH. “Kompakttrainer: Prozessregelung.” <https://www.leybold-shop.de/ve6-3-3-1.html> (Zugriff am: 11. Dez. 2022).
- [8] Ingenieurbüro Dr.Kahlert, Hg., “Produktübersicht 2022,” Zugriff am: 5. Dezember 2022. [Online]. Verfügbar unter: <https://www.kahlert.com/web/download/produktuebersicht.pdf>
- [9] W. Schroeder, “Lust statt Frust– Lehren und Lernen mit allen Sinnen,” Institut für berufliche Bildung und Weiterbildung e. V., Göttingen, 2005.
- [10] D. Sotelo *et al.*, “Lab-Tec@Home: A Cost-Effective Kit for Online Control Engineering Education,” *Electronics*, Jg. 11, Nr. 6, S. 907, 2022, doi: 10.3390/electronics11060907.
- [11] S. Wang, F. Zhang, Q. Tang, X. Zhang und R. Zhao, “A Take-Home Motor Control Teaching Experiment Platform for Control Engineering-Related Courses,” *IEEE Transactions on Education*, Jg. 65, Nr. 2, S. 115–123, 2022, doi: 10.1109/TE.2021.3094981.
- [12] J. A. Rossiter, S. A. Pope, B. Jones und J. D. Hedengren, “Evaluation and demonstration of take home laboratory kit,” *IFAC-PapersOnLine*, Jg. 52, Nr. 9, S. 56–61, 2019, doi: 10.1016/j.ifacol.2019.08.124.
- [13] K. Liebler. “GitHub-Repository des Lab@Home-Projektes.” <https://github.com/klaus-liebler/labathome> (Zugriff am: 7. Dez. 2022).

- [14] D. Déharbe, S. Galvão und A. M. Moreira, “Formalizing FreeRTOS: First Steps,” in *Formal Methods: Foundations and Applications* (Lecture Notes in Computer Science), D. Hutchison et al., Hg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 101–117.
- [15] H. Mann, H. Schiffelgen, R. Froriep und K. Webers, *Einführung in die Regelungstechnik: Analoge und digitale Regelung, Fuzzy-Regler, Regler-Realisierung, Software*, 12. Aufl. München: Hanser, 2019. [Online]. Verfügbar unter: <https://www.hanser-elibrary.com/doi/book/10.3139/9783446456945>
- [16] *Speicherprogrammierbare Steuerungen: Teil 3: Programmiersprachen*, DIN EN 61131-3, DIN, Berlin, Jun. 2014.
- [17] *Gebäudeautomation (GA): GA-Funktionen - Automationsfunktionen*, VDI 3814 Blatt 3.1, VDI, Berlin, Jan. 2019.
- [18] S. Köcher und K. Wilcox, ““I Made It Work”: How Using a Self-Assembled Product Increases Task Performance,” *J Consum Psychol*, Jg. 32, Nr. 3, S. 492–499, 2022, doi: 10.1002/jcpy.1262.
- [19] K. L. Chien, J. A. Hrones und J. B. Reswick, “On the Automatic Control of Generalized Passive Systems,” *J. Fluids Eng*, Jg. 74, Nr. 2, S. 175–183, 1952, doi: 10.1115/1.4015724.
- [20] ZIEGLER J. G., “Optimum Settings for Automatic Controllers,” *Trans. ASME*, Jg. 64, S. 759–768, 1942.
- [21] *Internationales Elektrotechnisches Wörterbuch: Teil 351: Leittechnik*, DIN IEC 60050-351, DIN, Berlin, Sep. 2014.
- [22] K. W. Houser und T. Esposito, “Human-Centric Lighting: Foundational Considerations and a Five-Step Design Process,” *Frontiers in Neurology*, Early Access. doi: 10.3389/fneur.2021.630553.
- [23] T. R. Alves, M. Buratto, F. M. de Souza und T. V. Rodrigues, “OpenPLC: An open source alternative to automation,” in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, 2014, S. 585–589, doi: 10.1109/GHTC.2014.6970342.
- [24] T. Alves und T. Morris, “OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research,” *Computers & Security*, Jg. 78, S. 364–379, 2018. doi: 10.1016/j.cose.2018.07.007. [Online]. Verfügbar unter: <https://www.sciencedirect.com/science/article/pii/S0167404818305388>