

## Vom Remote zum immersiven VR-Lab am Beispiel der industriellen Steuerungstechnik

Matthias Haack <sup>1</sup>, David M. Jozefiak <sup>2</sup>, Johannes Paehr <sup>3</sup>, Thomas N. Jambor <sup>4</sup>

**Abstract:** In einer Lehrveranstaltung aus dem Bereich der industriellen Steuerungstechnik haben die Autoren ein Remote Lab zur Aktivierung von Studierenden eingesetzt und evaluiert. Ausgehend von den Nachteilen eines Remote Lab wird in diesem Beitrag die Möglichkeit untersucht, Steuerungsprogramme für Mikrocontroller und Speicherprogrammierbare Steuerungen mittels eines Virtual Reality Lab (VR-Lab) zu kontrollieren. Das VR-Lab besteht einerseits aus einem selbstentwickelten Simulator für Mikrocontroller und dem existierenden Simulator PLCWinNT für Speicherprogrammierbare Steuerungen. Andererseits umfasst das VR-Lab eine Simulationsumgebung, bestehend aus einem 3D-Modell einer Produktionsanlage sowie Funktionalitäten zur Kontrolle der Steuerungsprogramme. Das VR-Lab ist in einer Laborstudie erprobt und anhand von semistrukturierten Leitfadeninterviews evaluiert worden. Aufgrund der geringen Vorerfahrung mit VR-Geräten waren die Studierenden neugierig und motiviert, die Technologie auszuprobieren. Sie nutzten das VR-Lab zur Kontrolle ihrer Programme sowie zur Fehleranalyse und lobten die intuitiven Interaktionsmöglichkeiten, variablen Perspektiven und zeitliche Verfügbarkeit. Die Studierenden empfanden die reale Produktionsanlage im Remote Lab dennoch motivierender als das virtuelle Modell.

**Keywords:** Virtual Lab, Remote Lab, Virtual Reality, Simulation, Arduino, Speicherprogrammierbare Steuerung, Industrielle Steuerungstechnik, Unity

### 1 Motivation

#### 1.1 Problemstellung

Eine Herausforderung für Lehrpersonen an Universitäten ist die Aktivierung und Motivierung großer, inhomogener Kohorten in einem klassischen Hörsaal. Die Notwendigkeit der Aktivierung großer Kohorten wird an mehreren beobachtbaren Phänomenen deutlich, wenn Studierende eine passive, rezipierende Rolle einnehmen. Hierzu zählen sinkende Aufmerksamkeit und Motivation, der Lehrperson kognitiv zu folgen, selbst wenn die Studierenden eine hohe intrinsische Motivation aufweisen [Sm01]

---

<sup>1</sup> Leibniz Universität Hannover, Fachgruppe für Didaktik der Elektrotechnik und Informatik, Appelstraße 9A, Hannover, 30167, haack@dei.uni-hannover.de, <https://orcid.org/0000-0002-3046-1460>

<sup>2</sup> Leibniz Universität Hannover, Fachgruppe für Didaktik der Elektrotechnik und Informatik, Appelstraße 9A, Hannover, 30167, jozefiak@zdt.uni-hannover.de, <https://orcid.org/0000-0001-6637-2518>

<sup>3</sup> Leibniz Universität Hannover, Fachgruppe für Didaktik der Elektrotechnik und Informatik, Appelstraße 9A, Hannover, 30167, paehr@dei.uni-hannover.de, <https://orcid.org/0000-0002-6801-1764>

<sup>4</sup> Leibniz Universität Hannover, Fachgruppe für Didaktik der Elektrotechnik und Informatik, Appelstraße 9A, Hannover, 30167, jambor@dei.uni-hannover.de, <https://orcid.org/0000-0002-4360-5339>

[Fr08] [KSE05]. Überdies werden die individuellen Vorkenntnisse und Interessen der Studierenden nicht berücksichtigt, sodass aus konstruktivistischer Sicht im ungünstigsten Fall Reize als nicht viabel angesehen werden. Andernfalls werden neue Informationen ggf. als träges Wissen assimiliert bzw. akkommodiert [Am04]. Aufgrund fehlender Interaktion mit anderen Studierenden bleiben Fehlvorstellungen unentdeckt, sodass schlechtere Lernleistungen die Folge sind [Si05] [Fr08].

## 1.2 Vorausgegangene Forschungsarbeit

Ausgehend von der Herausforderung, eine große, inhomogene Kohorte in universitären Lernsituationen zu aktivieren, ist in der Fachgruppe für Didaktik der Elektrotechnik und Informatik der Leibniz Universität Hannover ein fachdidaktisches Konzept, speziell für Lehrveranstaltungen der (technischen) Informatik und Elektrotechnik, entwickelt [HJ18] und erprobt [HJ20a] [HJ20b] [HJ20c] worden. Ausgangspunkt einer Lernsituation ist eine realitätsnahe Problemstellung, welche nach dem Flipped Classroom [LPT00] strukturiert ist. Die Studierenden informieren sich über die Problemstellung sowie die relevanten Inhalte, planen mögliche Lösungen und entscheiden sich zu Hause für eine Option. In der Universität werden zunächst offene Fragen geklärt und anschließend wird in kollaborativer Form eine Problemlösung von den Studierenden realisiert. Die Kontrolle der Lösung findet exemplarisch am Problemprodukt statt. Unter einem Problemprodukt wird ein (immaterieller) motivierender, viabler Lerngegenstand verstanden, welcher aus dem späteren Tätigkeitsfeld von Studierenden stammt und an dem sich mögliche Problemlösungen kontrollieren lassen. Aufgrund der großen Kohorte (über 100 Studierende) können nicht alle Lösungen vor Ort kontrolliert werden. Daher ist das Problemprodukt Bestandteil eines Remote Lab, sodass alle Studierenden die Möglichkeit haben, ihre Lösungen von zu Hause aus zu überprüfen.

Das Konzept ist in den Jahren 2019 und 2020 in der Lehrveranstaltung „Industrielle Steuerungstechnik und Echtzeitsysteme“ erprobt worden. Als Problemprodukt sind Modelle von industriellen Produktionsanlagen eingesetzt worden, welche mit Mikrocontrollern ( $\mu\text{C}$ ) und Speicherprogrammierbaren Steuerungen (SPS) von den Studierenden gesteuert wurden. Die Evaluationsergebnisse zeigen auf der einen Seite, dass anwendungsbezogene Aufgabenstellungen sowie die Kontrolle der eigenen Lösungen an realer Hardware (dem Problemprodukt) eine motivierende Wirkung auf die Studierenden hat und zur Verbesserung der Prüfungsergebnisse beiträgt. Auf der anderen Seite bemängeln die Studierenden, dass es beim zeitgleichen Zugriff mehrerer Studierenden auf das Remote Lab oder bei technischen Defekten an der realen Hardware zu längeren Wartezeiten kommt. Darüber hinaus bieten Remote Labs nur aus festen Kamerapositionen eine zweidimensionale Ansicht auf das Problemprodukt, sodass aufgrund der Komplexität der Anlage nicht alle Prozesse für Lernende vollständig zugänglich sind. Zuletzt sei der Nachteil erwähnt, dass ein manueller Eingriff in den Remote Versuch durch den Lernenden wie in einem Präsenzlabor nicht möglich ist, sodass ein automatisierter Initialisierungsprozess den Versuch zurücksetzen muss.

### 1.3 Zielsetzung dieses Beitrags

Aufgrund der dargestellten Nachteile eines Remote Lab haben die Autoren entschieden, dieses durch eine virtuelle Simulationsumgebung zu ersetzen. Lernende können Simulationsumgebungen nicht nur zweidimensional auf einem Bildschirm<sup>5</sup> betrachten, sondern diese mit Virtual Reality Brillen (VR-Brillen) immersiv sowie dreidimensional erleben und mit dem virtuellen Problemprodukt intuitiv interagieren. In diesem Beitrag werden folgende Forschungsfragen evaluiert:

1. Welche Systemarchitektur (Struktur von Hard- und Software) eignet sich für die Kontrolle von Steuerungsprogrammen für  $\mu$ C und SPS durch Studierende in VR-Anwendungen mittels VR-Brillen?
2. Welche Vor- und Nachteile bieten VR- gegenüber Remote Labs speziell im Bereich der industriellen Steuerungstechnik für Studierende?

## 2 Stand der Forschung

Trentsios et al. [TWF20] erläutern in ihrem Beitrag die Implementierung von immersiven Visualisierungsmethoden ausgehend von einem Remote Lab, welches aus physikalischen Sensoren und Aktoren besteht, die mit einer SPS verbunden sind. Ein Server bietet den Lernenden Zugriff über das Internet und stellt bestimmte vordefinierte Zeitslots zur Verfügung, um ein Experiment auf dem Remote Lab auszuführen. Aufgrund der eingeschränkten Immersion in einem Remote Lab verwenden Trentsios et al. [TWF20] verschiedene Visualisierungsmethoden, um die Sinneseindrücke von Lernenden aus einem Präsenzlabor im Remote Lab zu imitieren. Folgende zwei Methoden existieren und sind mit der Unity-Entwicklungsplattform umgesetzt:

- die Konstruktion einer 360°-Ansicht des Remote Lab durch die Aufnahme mehrerer Fotos aus verschiedenen Winkeln sowie
- die Konstruktion einer virtuellen Kopie des Remote Lab als 3D-Modell.

Beim ersten Ansatz sind die Bilder mit der Unity-Entwicklungsplattform zu einer 360°-Ansicht kombiniert worden. Dies ist vergleichbar mit der Ansicht, welche von „Google Street View“ beim Betrachten einer Straße geboten wird<sup>6</sup>. Für die Umsetzung des zweiten Ansatzes wurde ein virtuelles 3D-Modell anhand des bestehenden realen Labors konstruiert. Dadurch entsteht, im Gegensatz zu den zuvor beschriebenen Bildaufnahmen, ein leicht unterschiedliches Erscheinungsbild des experimentellen Aufbaus. Studierende können sich schließlich in einem virtuellen Laborraum mit dem darin enthaltenen 3D-Modell der Anlage frei bewegen.

---

<sup>5</sup> Im Folgenden Bildschirmsimulation genannt.

<sup>6</sup> Die daraus resultierende Szene kann schlussendlich auf einem Computermonitor angesehen oder mit einem VR-Gerät betrachtet werden.

Da beide Ansätze, welche von Trentsios et al. [TWF20] erläutert werden, eine Visualisierungsmethode für das bestehende Remote Lab darstellen, besteht weiterhin der Nachteil, dass immer nur ein Studierender gleichzeitig mit diesem interagieren kann. Als möglicher Lösungsansatz für diese Einschränkung wurde vorgeschlagen, dass Lernende das Remote Lab wie bisher steuern und zudem beliebig viele Studierende die Ausführung beobachten, aber nicht damit interagieren können. Ferner ist eine Konvertierung des Remote Lab in ein Virtual Lab denkbar. Beide Ansätze werden von Trentsios et al. [TWF20] nicht mit Studierenden evaluiert.

Überdies untersuchen Alvarez et al. [ADM19] den Einsatz eines Virtual Lab zur Kontrolle von SPS-Programmen. Studierende haben innerhalb der Universität die Möglichkeit, ihre erstellten Steuerungsprogramme zunächst mithilfe eines bildschirmbasierten Virtual Lab zu kontrollieren. Nachdem die Korrektheit der Lösung durch die Simulation verifiziert wurde, können sie diese auf einem Remote Lab ausführen. Bei dieser Implementierung existiert, im Vergleich zu den beiden Ansätzen von Trentsios et al. [TWF20], neben dem Remote Lab ein separates, unabhängiges Virtual Lab und somit nicht nur eine reine Visualisierung des bereits bestehenden Remote Lab. Der Einsatz einer Simulation wird von Alvarez et al. [ADM19] mit der Eliminierung der Wartezeiten und Zugangsbeschränkung zum Remote Lab begründet. Außerdem können Kosten, welche bei der Installation und Instandhaltung des Remote Lab entstehen, reduziert werden. Während der Nutzung der Simulation besteht keine Gefahr einer Beschädigung der physikalischen Hardware des Remote Lab durch fehlende Kompetenzen der Studierenden und daraus resultierende Programmierfehler. Im Rahmen einer Nutzerstudie haben Alvarez et al. [ADM19] ihre Umsetzung mit einer Gruppe von 40 Studierenden erprobt. Eine Umfrage unter den Probanden hat unter anderem ergeben, dass die Innovation der eingesetzten Technologie geschätzt wurde. Weiterhin wurde das Virtual Lab in persönlichen Interviews als motivierend beschrieben und das Interesse für die darauffolgende Nutzung des Remote Lab geweckt. Insgesamt wurde auch die Entwicklung von Fähigkeiten zum zukünftigen Arbeiten mit ferngesteuerten Prozessen gelobt.

Der Vergleich des Einsatzes von einem Präsenzlabor, einem Remote Lab oder einem VR-Lab wurde von Hernández-De-Menéndez et al. [HVM19] in einer Vergleichsstudie untersucht. Durch den Vergleich unterschiedlicher Labore an verschiedenen universitären Einrichtungen sind die Autoren zu dem Entschluss gekommen, dass sowohl ein Remote Lab als auch ein VR-Lab eine angemessene Option ist, um Präsenzlabor zu ersetzen. Im Falle eines VR-Lab wird dies mit den geringen Kosten und der einfachen Reproduzierbarkeit der Konfiguration sowie der Vielfältigkeit der VR-Technologie begründet. Unter Studierenden von technischen Studiengängen wurden auch eine große Akzeptanz sowie Lernerfolge bei der Nutzung eines VR-Lab berichtet.

Im Gegensatz zu den vorgestellten Ansätzen wird in diesem Beitrag die Implementierung und Evaluation eines vollständigen immersiven VR-Lab vorgestellt. Dieses bietet nicht nur eine Visualisierung, sondern auch die Möglichkeit zur eigenständigen, vom Remote Lab unabhängigen Kontrolle von Steuerungsprogrammen.

### 3 Implementierung

Nachfolgend wird die Architektur und Implementierung der Simulationsumgebung beschrieben. Dabei wird auf die Auswahl der verschiedenen Komponenten eingegangen und der Zusammenhang sowie die Kommunikation zwischen diesen erläutert.

Die Systemarchitektur des VR-Lab (vgl. Abb. 1) besteht einerseits aus einem Rechner (üblicher Desktop-PC oder Laptop), auf welchem die Entwicklungsumgebung sowie ein Simulator für  $\mu\text{C}$  (vgl. Abschnitt 3.1) und SPS (vgl. Abschnitt 3.2) ausgeführt werden. Andererseits enthält das VR-Lab ein Standalone-VR-Gerät, bestehend aus einem Headset und zwei Controllern, auf welchem die Simulationsumgebung (vgl. Abschnitt 3.3) installiert ist.

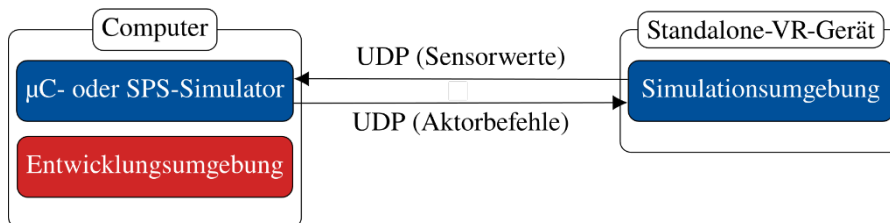


Abb. 1: Systemarchitektur des VR-Lab

Der  $\mu\text{C}$ - bzw. SPS-Simulator ersetzt das reale Steuergerät des Remote Lab, auf welchem die kompilierten Steuerungsprogramme (entwickelt mit der Arduino-IDE bzw. CODESYS) ausgeführt werden. Das Problemprodukt des Remote Lab wird durch ein reaktives System innerhalb der Simulationsumgebung in Form einer Unity-Anwendung substituiert. Die Kommunikation zwischen dem  $\mu\text{C}$ - bzw. SPS-Simulator und der Simulationsumgebung erfolgt über das Netzwerkprotokoll UDP<sup>7</sup> innerhalb eines gemeinsamen Netzwerks (vgl. Abschnitt 3.4).

Das VR-Lab der Autoren ist mit einer Oculus Quest als VR-Gerät realisiert, da dieses ein Standalone-Gerät darstellt und somit keine kabelgebundene Verbindung zwischen dem Rechner und dem VR-Gerät bestehen muss. Für die Anwendung in der Lehre (z. B. in einem Hörsaal oder Labor) erscheint dies aus Sicht der Autoren dieses Beitrags zweckmäßiger.

#### 3.1 Simulation eines Mikrocontrollers

Da die Programmierung eines Arduino in der Programmiersprache C/C++ erfolgt, besteht die Möglichkeit, diesen Programmcode auch für die Ausführung auf einem Computer zu kompilieren. Hierfür wird eine eigene Implementierung der Arduino-Bibliothek benötigt. Arduino-spezifische Funktionsaufrufe (wie beispielsweise `digitalRead()` und

<sup>7</sup> Die Kommunikation mittels Netzwerkvariablen basiert auf dem Netzwerkprotokoll UDP.

`digitalWrite()`) können dabei so implementiert werden, dass die Werte über das Netzwerk an die Simulationsumgebung gesendet oder von dieser empfangen werden. Dieser Ansatz erfordert jedoch einen erhöhten Programmieraufwand, da die Implementierung des Verhaltens von Interrupts nicht trivial ist, und stellt ein Beispiel für Unterschiede zwischen Programmen für Mikrocontroller und Computeranwendung dar.

Die Autoren dieses Beitrags haben sich aufgrund der o. g. Nachteile gegen den zuvor beschriebenen Ansatz entschieden. Stattdessen wird ein  $\mu$ C-Simulator, basierend auf der Bibliothek SimAVR<sup>8</sup>, einschließlich einer grafischen Benutzeroberfläche und Funktionen zur Netzwerkkommunikation mit der Simulationsumgebung entwickelt, welcher den Maschinencode in Software interpretiert und ausführt. Eine Alternative stellt der auf JavaScript basierte Simulator AVR8js<sup>9</sup> dar, welcher jedoch aufgrund der mangelhaften Performance ungeeignet ist.

Bei SimAVR handelt es sich um eine Bibliothek zur Interpretation und Ausführung von Maschinencode, welche die Möglichkeit zur Simulation eines AVR-Mikrocontroller bereitstellt. Die Bibliothek erlaubt die Verwendung zahlreicher Komponenten wie EEPROM, IO-Ports und Timern sowie die Verwendung von Interrupts. Die Bibliothek ist in der Programmiersprache C entwickelt und ermöglicht die Ausführung von Programmen, welche mit `avr-gcc` kompiliert sind. Dabei handelt es sich um den Compiler, welcher auch Teil der Arduino-IDE ist (vgl. Abb. 2, links). Dieser erzeugt aus dem Quellcode in der Programmiersprache C/C++ Dateien im ELF-Format (Dateiformat für ausführbare Dateien und kompilierte Bibliotheken), welche mit dem Programm `avr-objcopy` in ein Binärabbild konvertiert werden, das ausschließlich den Programmcode enthält. Die daraus resultierenden Daten werden mit SimAVR (vgl. Abb. 2, rechts) direkt in den Programmspeicher des (virtuellen) Mikrocontrollers geladen.

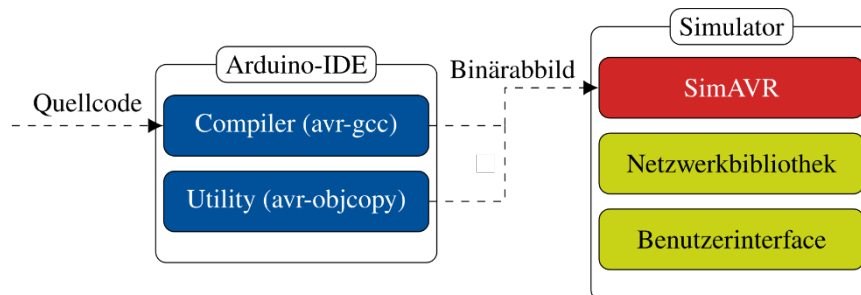


Abb. 2: Struktur des Simulators für Arduino-Programme

<sup>8</sup> <https://github.com/buserror/simavr>

<sup>9</sup> <https://github.com/wokwi/avr8js>

### 3.2 Simulation einer SPS mit CODESYS

Die Software PLCWinNT gehört mit weiteren Anwendungen zur standardmäßigen Installation einer CODESYS-Umgebung und erlaubt es, eine SPS zu simulieren. In CODESYS können verschiedene physikalische Zielsysteme verwendet werden. Für die Verwendung mit PLCWinNT muss das entsprechende gleichnamige Zielsystem ausgewählt werden. Die Kommunikation der Simulationsumgebung mit dem Simulator PLCWinNT erfolgt, ähnlich zur realen Kommunikation, zwischen zwei physikalischen Steuerungen über Netzwerkvariablen. Die Verbindung zwischen der Entwicklungsumgebung und dem Simulator zur Programmierung und Überwachung erfolgt dagegen über TCP.

### 3.3 Simulationsumgebung

Die Simulationsumgebung ersetzt das Problemprodukt des Remote Lab. Die einzelnen Teilsysteme sind als 3D-Modelle nachgebildet und in einer Unity-Anwendung zu einem reaktiven System zusammengesetzt. Die resultierende Simulationsumgebung stellt einen virtuellen Laborraum dar, in dem sich die Studierenden frei bewegen können, die Problemprodukte auf einem Tisch sehen und mit ihnen, wie in der realen Welt, interagieren können.

Wenn Studierende das VR-Gerät verwenden und die Anwendung gestartet ist, muss die Verbindung zwischen dem  $\mu$ C- oder SPS-Simulator und der Simulationsumgebung eingerichtet werden (vgl. Abb. 3). Neben dieser Funktionalität haben Studierende die Möglichkeit, über ein Menü einerseits das Problemprodukt vollständig in die Ausgangslage zurückzusetzen und andererseits das Werkstück auf die Startposition zu setzen.

Sowohl im laufenden als auch im ruhenden Zustand des Steuerungsprogramms können Studierende Werkstücke greifen bzw. ablegen (vgl. Abb. 3), Taster betätigen oder Sensoren (z. B. Lichtschranken) auslösen.

Zur Optimierung der Vorstellung vom Aufbau des Problemproduktes sowie zur Fehleranalyse sind die einzelnen Sensoren und Aktoren mit ihren entsprechenden Bezeichnungen (beispielsweise Endlagentaster E4) beschriftet. Die Farbe kennzeichnet den aktuellen Zustand „true“ (orange/blau) oder „false“ (grau). Darüber hinaus existiert zusätzlich eine Zustandstabelle auf dem Benutzerinterface.

Sollte das Steuerungsprogramm des Studierenden zu einem Fehler führen (z. B. das Fahren eines Motors über einen Endlagentaster), wird an der entsprechenden Position des Aktors ein rotes Ausrufezeichen dargestellt (vgl. Abb. 3). Mit Berühren des Ausrufezeichens mit der VR-Hand sehen Studierende eine entsprechende Fehlerbeschreibung.

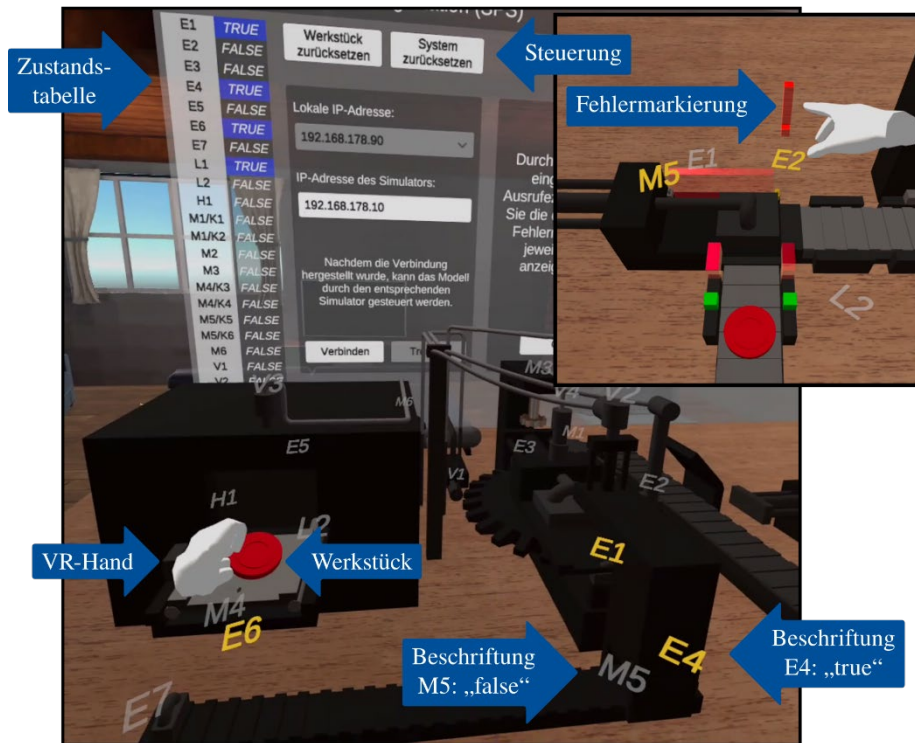


Abb. 3: Simulationsumgebung

### 3.4 Schnittstelle zwischen Simulator und Simulationsumgebung

Eine Möglichkeit zur Kommunikation zwischen zwei physikalischen Steuerungen ist der Austausch von Netzwerkvariablen<sup>10</sup>. Die Steuergeräte sind Teil eines lokalen Netzwerks und tauschen Variablen durch das Senden von UDP-Paketen bei Änderung eines Wertes bzw. zyklisch aus. Diese Art der Kommunikation kann auch bei virtuellen Steuerungen verwendet werden.

Aus diesem Grund haben sich die Autoren dieses Beitrags entschieden, die Kommunikation zwischen dem SPS-Simulator und der Simulationsumgebung im VR-Lab über Netzwerkvariablen zu realisieren. Für die Kommunikation mit dem  $\mu$ C-Simulator kann diese Methode nicht verwendet werden. Im Gegensatz zum SPS-Simulator kann hierbei nicht auf die Variablen des Steuerungsprogramms zugegriffen werden, da das Programm als Maschinencode vorliegt. Stattdessen werden die Sensoren und Aktoren der Simulationsumgebung mit den virtuellen Pins des simulierten  $\mu$ C verbunden. Hierfür

<sup>10</sup> <https://de.codesys.com/produkte/codesys-communication/proprietare-kommunikationsprotokolle.html>



werden die Zustände ebenfalls mittels UDP-Paketen an den Simulator oder die Simulationsumgebung gesendet.

## 4 Evaluation

### 4.1 Forschungsdesign

Das VR-Lab ist in einer Laborstudie im Frühjahr 2021 mit zehn Studierenden (davon neun männlich) erprobt worden. Sieben Studierende befinden sich im Bachelor- sowie drei im Masterstudium. Drei gaben an, dass Deutsch nicht ihre Muttersprache ist. Alle Studierende haben Vorkenntnisse in mindestens einer Programmiersprache (i. d. R. C++ oder Java). Drei Studierende haben die Lehrveranstaltung, in welcher das Remote Lab eingesetzt worden ist (vgl. Abschnitt 1.2), bereits belegt.

In der Studie erhielten die Studierenden zunächst eine Einführung in die Programmierung einer SPS mit der Entwicklungsumgebung CODESYS. Anschließend haben sich die Studierenden gemäß dem fachdidaktischen Konzept (vgl. Abschnitt 1.2) zu einer Problemstellung informiert, Lösungsoptionen geplant und sich anschließend für eine entschieden. Das Implementieren und Kontrollieren fand, im Gegensatz zum fachdidaktischen Konzept, aufgrund der Corona-Pandemie in Einzelarbeit statt. Zum Kontrollieren stand den Studierenden das in Kapitel 3 beschriebene VR-Lab und als Alternative der CODESYS-Simulator, welcher auf der manuellen Eingabe von Parametern basiert, zur Verfügung. Anschließend haben die Studierenden eine Einführung in die Arduino-spezifische  $\mu$ C-Programmierung erhalten und eine weitere Problemstellung gelöst. Am Ende der Studie haben die Studierenden Fragebögen zur Lernsituation und Mediennutzung sowie Fragen in einem semistrukturierten Leitfadenterview zum VR-Lab beantwortet. Im Interview sind Fragen zu den vier Bereichen *allgemeine Vorerfahrungen mit Simulationen und VR*, *Einsatzzweck* und *Bewertung dieses VR-Lab* sowie *Präferenzen zwischen verschiedenen Simulationsmöglichkeiten (VR, Remote, (nicht) grafische Bildschirmsimulation)* gestellt worden. Die Interviews sind in einer Inhaltsanalyse nach Mayring ausgewertet worden.

### 4.2 Ergebnisse

Aus den Antworten der Studierenden zu den Interviewfragen aus dem Bereich *allgemeiner Vorerfahrungen mit Simulationen und VR* geht hervor, dass alle Teilnehmenden bis auf einen im Studium mit Simulationssoftware (z. B. MATLAB, Simulink, LTspice, SolidWorks, CODESYS, Factory I/O) gearbeitet haben. Alle Studierenden waren unerfahren mit VR-Geräten in Lehrveranstaltungen. Vier Probanden haben allerdings im

privaten Bereich VR einmalig ausprobiert und zwei Studierende besitzen ein Google Cardboard<sup>11</sup>, welches sie jedoch nicht zu Lernzwecken einsetzen.

Nachdem die Vorerfahrungen der Studierenden dargelegt worden sind, wird im Folgenden der *Einsatzzweck des VR-Lab* erläutert. Aufgrund der Tatsache, dass das VR-Lab zur Kontrolle der Steuerungsprogramme von der Lehrperson vorgestellt wurde und die VR-Geräte für die meisten Studierenden neu waren, verwundert es nicht, dass alle Studierenden dieses zur Kontrolle einsetzten. Auf Nachfragen gaben acht von zehn Probanden an, nicht nur die Korrektheit des Steuerungsprogramms durch Beobachtung des Produktionsablaufs überprüft zu haben, sondern auch eventuelle Fehler möglichst präzise eingegrenzt zu haben. Zu diesem Zweck haben sie den Versuch ggf. wiederholt. Nur ein Student gab an, dass er das VR-Lab benutzt hat, um eine bessere Vorstellung vom Aufbau und der Funktionsweise zu erhalten, da dies für ihn aus den Abbildungen und textuellen Beschreibungen im Informationsmaterial nicht ersichtlich war. Dies könnte darauf zurückzuführen sein, dass Deutsch nicht seine Muttersprache ist.

Im Hinblick auf die *Bewertung des VR-Lab* äußern sich die Studierenden grundsätzlich positiv. Besonders lobend wurde das real wirkende Problemprodukt, die Immersion sowie die intuitive Bedienung und Interaktionsmöglichkeiten mit den Controllern erwähnt. Darüber hinaus geht aus den Interviews hervor, dass weiterführende Kontrollmöglichkeiten im VR-Lab, wie z. B. ein Zustandsgraph und eine Stoppuhr für zeitabhängige Vorgänge, zur Verbesserung der Fehleranalyse durch Studierende beitragen können. Zusätzlich lassen sich mit einer Aufzeichnung des Ablaufs Wiederholungen von Versuchen vermeiden. Ferner beschrieben zwei Probanden, dass sie die Befürchtung hatten, mit Gegenständen der realen Umgebung zu kollidieren, während sie die VR-Brille getragen hatten. Ein weiterer Proband hatte währenddessen Kopfschmerzen bekommen.

Im letzten Teil des Interviews wurden die Probanden zu ihren *Präferenzen zu den verschiedenen Simulationsmöglichkeiten (VR, Remote, (nicht) grafische Bildschirmsimulation)* befragt. Fast alle Studierenden bevorzugten das VR-Lab gegenüber dem Remote Lab aufgrund der Verfügbarkeit, den intuitiven Interaktionsmöglichkeiten und variablen Perspektiven. Zwei Drittel empfanden die reale Hardware als motivierender. Alle Studierenden bevorzugten eine grafische Bildschirmsimulation gegenüber einer nicht grafischen, da sie den Zeitaufwand als minimaler ein- und die geringe Abstraktion wertschätzten. Für das VR-Lab spricht aus Sicht der Studierenden zudem die motivierende Lernumgebung. Der Zeitaufwand wird bei einer grafischen Bildschirmsimulation dagegen von drei Probanden als geringer beurteilt.

---

<sup>11</sup> Google Cardboard besteht aus einer Halterung aus Karton und ermöglicht es, zusammen mit der gleichnamigen mobilen Applikation, einfache VR-Anwendungen mit einem Smartphone zu verwenden.

## 5 Diskussion, Fazit und Ausblick

Ausgehend von der ersten Forschungsfrage nach einer geeigneten Systemarchitektur für ein VR-Lab zur Kontrolle von Steuerungsprogrammen für  $\mu\text{C}$  und SPS (vgl. Abschnitt 1.3), ist in Kapitel 3 die Struktur eines solchen VR-Labs erläutert worden. Über die Arduino-IDE bzw. CODESYS implementieren Studierende Programme für  $\mu\text{C}$  bzw. SPS und überprüfen diese durch das Kompilieren auf syntaktische Fehler. Der von den Autoren implementierte Simulator für  $\mu\text{C}$  bzw. der vorhandene Simulator PLCWinNT für SPS ermöglichen zusammen mit der selbstentwickelten Simulationsumgebung eine Kontrolle an einem virtuellen Problemprodukt.

In einer Laborstudie ist das VR-Lab von zehn Studierenden erprobt und bewertet worden. Die Ergebnisse zeigen, dass die neue Technologie Neugier bei den Studierenden aufgrund der geringen Vorerfahrung weckt sowie eine motivierende Wirkung hat. Diese Erkenntnis deckt sich mit denen von Alvarez et al. [ADM19]. Ferner bietet das VR-Lab gegenüber dem Remote Lab grundsätzlich eine bessere Möglichkeit für Studierende, Steuerungsprogramme aufgrund der zeitlichen Verfügbarkeit, variablen Perspektiven und intuitiven Interaktion mit dem Problemprodukt zu kontrollieren. Jedoch zeigt sich auch, dass ein virtuelles Problemprodukt im VR-Lab eine geringere Motivation auf die Studierenden hat und somit der Zielsetzung des fachdidaktischen Konzeptes entgegensteht (vgl. Abschnitt 1.2). Der Einsatz des VR-Lab in der Lehrveranstaltung ist aufgrund der noch nicht vorhandenen VR-Geräte bei den Studierenden zu Hause nicht möglich, andernfalls wäre dies jedoch vorstellbar. Im Hörsaal gestaltet sich der Einsatz des VR-Lab aufgrund der eingeschränkten Bewegungsfreiheit in den Sitzreihen schwierig. Sowohl im Hörsaal als auch zu Hause kommen daher zunächst nur Bildschirmsimulationen in Betracht. Insbesondere haben so alle Studierenden im Hörsaal die Möglichkeit, eine Kontrolle mittels grafischer Simulation durchzuführen. Die Kontrolle von ein bis zwei Lösungen im Plenum im VR-Lab ist denkbar. Da der Hörsaal jedoch der einzige Ort ist, an dem die Studierenden das Problemprodukt real erleben können, ist es aus Sicht der Autoren motivatorisch nicht sinnvoll, an dieser Stelle das Remote Lab durch ein VR-Lab zu ersetzen. Zu einem ähnlichen Ergebnis kommen auch Hernández-De-Menéndez et al. [HVM19] in ihrer Vergleichsstudie (vgl. Kapitel 2).

Im Hinblick auf die zweite Forschungsfrage (vgl. Abschnitt 1.3) ist in Zukunft, aufgrund der genannten Vor- und Nachteile des VR-Lab sowie des Einsatzes im Rahmen des fachdidaktischen Konzeptes, eine parallele Bereitstellung beider Labortypen (Remote und VR bzw. Bildschirmsimulation) für die selbstständige Kontrolle zu Hause geplant. In einem Feldversuch werden die Autoren beide Labortypen unter fachdidaktischen Aspekten quantitativ genauer evaluieren. Ferner ist eine Erweiterung der bildschirmbasierten Simulationsumgebung auf Basis der Evaluationsergebnisse zur Verbesserung der Fehleranalyse (vgl. Abschnitt 4.2) vorgesehen. Überdies soll für alternative Einsatzorte des VR-Lab (z. B. in einem Präsenzlabor) ein Livestream aus der Perspektive des Lernenden implementiert werden, um Gruppenarbeit sowie Unterstützung durch die Lehrperson zu ermöglichen.

## Literaturverzeichnis

- [ADM19] Alvarez, J.; Díaz, G.; Macías, M.: Programming logical controllers using remote labs and virtual reality. In: 2019 IEEE International Conference on Engineering Veracruz (ICEV), Bd. 1, S. 1–4, 2019.
- [AM04] Ameln, F.: Konstruktivismus. Francke, A., 2004
- [Fr08] Froberg, D.: Mobile Learning. Dissertation. Wirtschaftswissenschaftliche Fakultät der Universität Zürich, Fulda, 2008.
- [HJ18] Haack, M.; Jambor, T.: Lernendenzentrierte aktivierende Lehre in einem Hörsaal mit großen Kohorten In: Diversität und kulturelle Vielfalt, Differenzieren, Individualisieren - oder integrieren? Referate der 13. Ingenieurpädagogischen Regionaltagung, S. 261-266, 2018
- [HJ20a] Haack, M.; Jambor, T.: Investigation of Variables Related to University Drop-out in Problem-Based Learning. In: IEEE Frontiers in Education Conference, Uppsala, 2020
- [HJ20b] Haack, M.; Jambor, T.: Influence of Problem-Based Learning on Student Performance. In A. Cardoso; G. R. Alves; T. Restivo (Hrsg.): Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON), S. 295-299, Porto, 2020
- [HJ20c] Haack, M.; Jambor, T.: Seamless Learning im problembasierten Flipped Classroom mit einem Remote Lab. In J. Erlemann; C. M. Werder (Hrsg.): Seamless Learning – lebenslanges, durchgängiges Lernen ermöglichen, Waxmann, S. 101-109, 2020
- [HVM19] Hernández-de-Menéndez, M.; Vallejo Guevara, A.; Morales-Menendez, R.: Virtual reality laboratories: a review of experiences. International Journal on Interactive Design and Manufacturing (IJIDeM) 13/3, S. 947–966, 2019, ISSN: 1955-2505.
- [KSE05] Kopf, S.; Scheele, N.; Effelsberg, W.: The Interactive Lecture: Teaching and Learning Technologies for Large Classrooms. Institut für Informatik. Forschungsbericht, 2005.
- [LPT00] Lage, M. J.; Platt, G. J. & Treglia, M.: Inverting the Classroom: A Gateway to Creating an Inclusive Learning Environment. In: The Journal of Economic Education, S. 30-43, 2000
- [Si05] Siebert, H.: Pädagogischer Konstruktivismus: lernzentrierte Pädagogik in Schule und Erwachsenenbildung. Beltz, Weinheim/Basel, 2005, ISBN: 978-3-40725-399-6.
- [Sm01] Smith, B.: Just give us the right answer. In: Lecturing Case studies, experience and practice, o. O., 2001.
- [TWF20] Trentsios, P.; Wolf, M.; Frerich, S.: Remote Lab meets Virtual Reality - Enabling immersive access to high tech laboratories from afar. Procedia Manufacturing 43/, Sustainable Manufacturing - Hand in Hand to Sustainability on Globe: Proceedings of the 17th Global Conference on Sustainable Manufacturing, S. 25–31, 2020