

Juni 2025

Praktikum zur Lehrveranstaltung Software Engineering Vertiefung Versuch: ART1 Türsteuerung

Organisation des Praktikumsversuchs: M. Herhold

Inhaltsverzeichnis

1	Versuchsziel	1
2	Einführung	1
3	Beschreibung des vorgegebenen Versuchsaufbaus	2
4	Aufgabenstellung	3
5	Beschreibung Türverhalten	4
6	Vorschläge zu Design und Entwicklung der Steuerung (optional)	6
7	Hinweise zum Praktikumstermin und dessen Durchführung	7
8	Downloads	8
9	Arbeits- und Brandschutzhinweise	8

1 Versuchsziel

Festigung des Wissens der Vorlesung Software Engineering Vertiefung. Der Fokus liegt auf der Beherrschung der Grundlagen der imperativen Programmiersprache C und dem Umgang mit den Eigenheiten der Sprache. Das Praktikum umfasst den Entwurf und die Implementierung einer Steuerung einer automatischen Tür (Steuerungssoftware).

Ein weiteres Ziel dieses Versuchs ist, durch Übung, Erfahrungen zu sammeln, wie ein gut wartbares Steuerungsprogramm entworfen und implementiert werden kann. Das hohe Ideal für diese Aufgabe wäre ein Programm, welches nur wenige oder keine Änderungen benötigt um eine Tür mit etwas abweichender Hardware-Konfiguration zu betreiben.

2 Einführung

Automatisch bewegte Türen können z.B. bei Fahrzeugen, Garagen, Geschäften, Schutz- und Sicherheitsräumen eingesetzt werden. Automatisierungsziele können u.a. sein, die Passagezeiten zu verringern, den Sicherheitsstandard und einen Komfort nach dem Stand der Technik zu erfüllen. Teilaufgaben für die Automatisierung einer Türsteuerung sind u.a.:

- automatisches Türöffnen bei Nutzeranwesenheit, Befehlseingaben oder im Störfall
- automatisches Türschließen nach Nutzerpassage, Befehlseingabe oder Zeitereignissen
- Verriegelung und Freigabe der Türnutzung
- automatisches Unterbrechen der Türbewegung bei Störungen
- ermöglichen von Reparatur- und Wartungsarbeiten

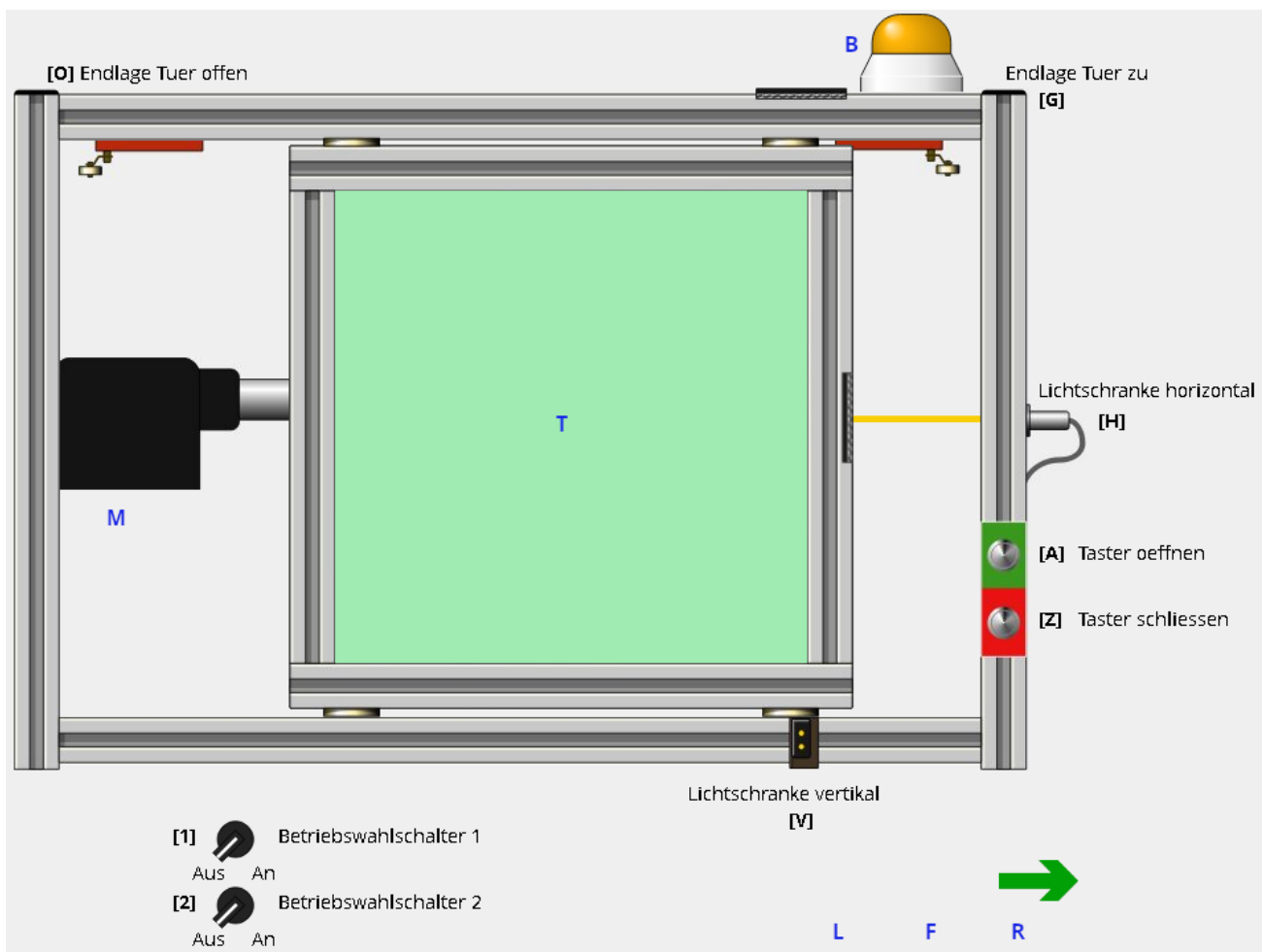
Im Versuch soll die Steuerungssoftware einer automatischen Tür für verschiedene Einsatzzwecke realisiert werden. Die Steuerung hat die Aufgabe, die Tür bei bestimmten Ereignissen zu öffnen oder zu schließen und soll unterschiedliche in Hardware realisierte Türen steuern. Auch geänderte Sicherheitsvorschriften müssen in der Steuerungssoftware einfach umsetzbar bleiben. Aus diesen Gründen wird auf Wart- und Erweiterbarkeit der Software Wert gelegt.

3 Beschreibung des vorgegebenen Versuchsaufbaus

Zum Praktikumstermin steht ein reales Türmodell zur Verfügung. Dieses Türmodell besitzt eine Reihe von Sensoren und Aktoren, welche über eine angeschlossene I/O-Karte ausgelesen bzw. angesteuert werden können. Die I/O-Karte ist auf den Praktikums-PCs betriebsbereit eingebunden. Als Programmier- und Laufzeitumgebung wird die, aus dem Praktikumsversuch 3 – Labyrinth des Wintersemesters bekannte, Virtuelle Maschine¹ verwendet.

Der vorgegebene Programmrahmen implementiert eine Abstraktionsschicht (*door-interface*) um die reale Tür bzw. den Softwaresimulator anzusteuern.

Das Verhalten der Hardware wird vollständig von **Ihrer** Steuerungssoftware bestimmt.



Legende Türmodell:

[1]	Betriebswahlschalter 1	BW1	[O]	Endlagenschalter Tür vollständig geöffnet	ELO
[2]	Betriebswahlschalter 2	BW2	[G]	Endlagenschalter Tür vollständig geschlossen	ELG
[H]	Lichtschranke horizontal	LSH	[A]	Nutzertaster zum Öffnen der Tür	NTA
[V]	Lichtschranke vertikal	LSV	[Z]	Nutzertaster zum Schließen der Tür	NTZ
B	Warnleuchte	AWL	L	Anzeige Antriebseinheit Linksbewegung	AML
T	bewegliches Türblatt		R	Anzeige Antriebseinheit Rechtsbewegung	AMR
M	Antriebseinheit		F	Anzeige für problematisches Antriebssignal	

Während der Simulation können die Signale, der mit eckigen Klammern markierten Sensoren, per Tastatur umgeschaltet werden. Die Taste **Q** beendet die Simulation.

Das 3-stellige Kürzel hinter der Beschreibung (zum Beispiel: BW1) bezeichnet den, im weiteren Dokument verwendeten, Namen des Sensors bzw. Aktors, welchen Ihre Software für Steuerungszwecke nutzen kann.

¹im nachfolgenden Text als VM bezeichnet

Sensoren (Steuerungssoftware-Input) und Aktoren (Steuerungssoftware-Output) sind digital und haben einen 0- oder 1-Pegel.

Der Zugriff auf die Hardware/Simulator erfolgt über die Funktionen, welche in der Datei `door_interface.h` deklariert sind.

Sensoren können mit `dio_read()` ausgelesen und Aktoren mit `dio_write()` beeinflusst werden.

Der Zugriff auf die Hardware ist über Ports mit jeweils 8 Pins organisiert. Pin 0 ist das niederwertigste Bit.

Die im Simulator dargestellte Hardware realisiert folgende Zuordnungen:

Name	Sensor	Aktor	Port	Pin
BW1	X		0	0
BW2	X		0	1
NTA	X		0	2
NTZ	X		0	3
ELO	X		0	4
reserviert	X		0	5
ELG	X		0	6
LSH	X		0	7
LSV	X		1	0
reserviert	X		1	1
reserviert	X		1	2
reserviert	X		1	3
reserviert	X		1	4
reserviert	X		1	5
reserviert	X		1	6
reserviert	X		1	7

Name	Sensor	Aktor	Port	Pin
AML		X	2	0
AMR		X	2	1
AWL		X	2	2
reserviert		X	2	3
reserviert		X	2	4
reserviert		X	2	5
reserviert		X	2	6
reserviert		X	2	7

Die Funktion `dio_read(unsigned int port, unsigned int *pins)` erwartet 2 Parameter:

1. Die Portnummer (`port`), welcher ausgelesen werden soll.
2. Eine Speicheradresse (`*pins`), zu welcher der Zustand der 8 zugeordneten Sensoren geschrieben werden soll.

Die Funktion `dio_write(unsigned int port, unsigned int pins)` erwartet ebenfalls 2 Parameter:

1. Die Portnummer (`port`), an welche geschrieben werden soll.
2. Die 8 zugeordneten Aktorsignale (`pins`), welche auf der Hardware gesetzt werden sollen.

Sensoren können **High-** oder **Low-aktiv** sein (siehe [Wikipedia: Logic level, 2-level logic, Active state](#)).

In der Simulation sind die Sensoren BW1, BW2, NTA, NTZ, ELO, ELG und LSV High-aktiv. Der Sensor LSV ist Low-aktiv. Die realen Türmodelle im Praktikumsversuch haben jeweils eine eigene einzigartige High-/Low-aktiv-Konfiguration und besitzen mehr Sensoren als die Simulation.

Der Motor der Tür hat eine unveränderliche Geschwindigkeit und wird über 2 Signale gesteuert:

1. AML : öffne die Tür
2. AMR : schließe die Tür

Wenn keines der beiden Signale anliegt, stoppt der Motor. Wenn beide Signale gleichzeitig anliegen, gilt dies als Fehler.

Der Aktor der Warnleuchte (AWL) lässt die Leuchte im Aktiven-Zustand selbstständig blinken. Ein an- und abschalten des Signals um ein Blinken zu erzeugen ist nicht notwendig.

4 Aufgabenstellung

Entwerfen, entwickeln und dokumentieren Sie im Team eine Steuerungssoftware für eine automatische Tür. Nutzen Sie für die Implementierung die Programmiersprache C² und den für den Versuch vorgegebenen Quellcoderrahmen und Simulator!

Das gewünschte Verhalten einer beispielhaften Tür wird im Abschnitt 5 beschrieben.

Verfolgen Sie einen strukturierten Ansatz³.

Fassen Sie die verschiedenen Eigenschaften (unter anderem: Name, Pin, High/Low-aktiv, in-Benutzung) eines Sensors/Aktors in einer C-Struktur zusammen!

²Eine Implementierung in der Programmiersprache C++ ist für diesen Versuch ausdrücklich nicht erwünscht!

³https://de.wikipedia.org/wiki/Strukturiertes_Design

Implementieren Sie einen einfachen Zustandsautomaten, welcher die Steuerung der Tür übernimmt, in einer Datei `steuerung.c`. Implementieren Sie einen Mechanismus zur Protokollierung der Zustandsänderungen des Zustandsautomaten. Die Kernkomponenten des Protokollierungsmechanismus sollen in einer Datei `protokollierung.c` zu finden sein.

Fügen Sie alle zusätzlichen Quellcode- und Header-Dateien dem Projektrahmen hinzu.

Achten Sie auf korrekte Fehlerbehandlung beim Aufrufen von (Bibliotheks-) Funktionen.

Verfolgen Sie das Ziel ein gut wartbares Steuerungsprogramm zu entwerfen und zu implementieren, welches mit nur wenigen Änderungen eine Tür mit anderer Pin/Port- und High/Low-aktive-Konfiguration steuern kann.

4.1 Entwicklung und Inbetriebnahme

Stellen Sie Ihre Steuerungssoftware vor dem Praktikumstermin mit Hilfe der Simulation fertig!

Während des Praktikumstermins wird Ihre Steuerung am echten Türmodell in Betrieb genommen. Zusätzliche am Modell zu findende Sensoren und unterschiedliche High/Low-aktive-Konfigurationen sollen von Ihnen während des Termins Ihrer Steuerungssoftware hinzugefügt werden.

Die Steuerungssoftware muss in der VM kompilieren und lauffähig sein.

4.2 Abschließende Dokumentation ihrer Arbeit (Inhalt)

- Einleitung & Ausblick
- Erläuterung der Arbeitsweise ihrer Steuerung, inklusive Ablaufdiagramm/Strukturdiagramm.
- Geben Sie die Booleschen Funktionen, die Sie zur Steuerung der Tür ausgearbeitet haben in geeigneter Weise an (zB. tabellarisch oder KV-Diagramm oder als Formel).
- Analysieren und dokumentieren Sie Probleme welche während der Entwicklung auftraten. (Verbesserungsvorschläge zur Praktikumsaufgabe/Simulator gehören nicht in die Dokumentation, werden aber gern – als E-Mail an den Betreuer – gesehen.)
- Diskutieren Sie Vor- und Nachteile Ihrer Lösung in Hinsicht auf zukünftige Verwendung und Erweiterbarkeit, anhand Ihrer Erfahrungen im Praktikumstermin.
- Nur wichtige Quellcode-Listings und nur als kurze Abschnitte mit zusätzlicher Erklärung.
- Umfang: 6 bis 9 Seiten, mit zusätzlichem herunterladbaren Standard-Deckblatt (siehe [Downloads](#)).

5 Beschreibung Türverhalten

Hinweis: bei folgender Beschreibung der Tür, handelt es sich um die Aussagen eines gedachten Auftraggebers. Fehlende, unklare oder widersprüchliche Aussagen sollen durch sinnvolles Verhalten der Software ausgeglichen werden. Für unsinnige Sensorwerte soll selbständig ein geeignetes Türverhalten entworfen und dokumentiert werden. Im Falle einer notwendigen Rücksprache mit dem gedachten zukünftigen Betreiber, nehmen Sie bitte Kontakt mit dem Praktikumsbetreuer auf.

5.1 Grundsätzliches Verhalten der Steuerungssoftware

- Sobald die Tür eine Endlage (komplett offen oder geschlossen) erreicht hat, soll der Motor abgeschaltet werden und die Signalleuchte soll aufhören zu blinken.
- Die Ausgänge AML und AMR gleichzeitig aktiv zu schalten, ist ein ungültiger Zustand und soll vermieden werden.
- Ein Wechsel der Betriebsart ist jederzeit möglich.

5.1.1 Logging / Protokolldatei⁴

- Um die korrekte Arbeitsweise der Türsteuerung verifizieren zu können, soll die Steuerungssoftware während Ihres Betriebs automatisch eine Protokolldatei erstellen.
- Die Protokolldatei, soll den sekundengenauen Startzeitpunkt des Programms auf der ersten Zeile vermerken.
- In den nächsten Zeilen folgt eine Liste aller Sensoren mit Namen, High/Low-aktiv-Konfiguration, Pin und Port.
 - Dafür soll eine Textzeile pro Sensor verwendet werden.
- Es folgt eine Trennzeile um den Protokollkopf vom Zustandsprotokoll abzutrennen.
- Danach soll jede Zustandsänderung der Türsteuerung im folgenden Format protokolliert werden:
 - Für jeden Zustandswechsel soll exakt eine Textzeile verwendet werden.
 - Zu Beginn der Textzeile steht eine sekundengenaue Zeitangabe.
 - Anschließend soll das auslösende Ereignis aufgelistet werden.
 - Als letztes folgen der aktuelle Zustand und der Folgezustand der Tür.

5.2 Betriebsarten:

Es gibt 4 mögliche Betriebsarten welche durch BW1 und BW2 geschaltet werden:

- Automatikbetrieb, Handbetrieb, Reparaturbetrieb, Ausgeschaltet

5.2.1 Automatikbetrieb

Im Automatikbetrieb befinden sich beide Betriebswahlschalter BW1 und BW2 im Aktiven-Zustand.

Aus Sicherheitsgründen soll die Signalleuchte 5 Sekunden nach dem Aktivieren der Tür leuchten um anzuzeigen, dass die Automatik aktiv ist. Während dieser Zeit, soll keine Aktion seitens der Tür erfolgen, alle Eingaben und Sensorwerte werden verworfen. Anschließend arbeitet die Automatiksteuerung nach folgendem Schema:

1. Sobald zu einem beliebigen Zeitpunkt an einem oder mehreren Sensoren NTA, LSH oder LSV ein aktiver Zustand detektiert wurde, öffnet sich die Tür bis diese den vollständig geöffneten Zustand erreicht hat.
2. Nachdem die Tür vollständig geöffnet wurde, verharrt diese für 3 Sekunden in dieser Position. Nach Ablauf dieser Zeit schließt sich die Tür. Aktive Zustände an den Sensoren NTA, LSH oder LSV starten diesen 3 Sekunden Timer sofort von vorn.
3. Ein Betätigen des Tasters NTZ, zu einem beliebigem Zeitpunkt, soll zum sofortigen Schließen der Tür führen. Die Tür soll sich unabhängig davon, ob NTZ gedrückt bleibt oder nicht, vollständig schließen. Automatikbetrieb-Regel 1 hat Vorrang und ist zu beachten.
4. Denkbar wäre ein zusätzlicher, noch nicht vorhandener, Sicherheitssensor, welcher die Tür beim Schließen sofort stoppt.

5.2.2 Manueller Betrieb:

Dieser Betriebsmodus ist dadurch definiert, dass BW1 aktiv und BW2 inaktiv ist.

1. Falls die Tür sich öffnet läuft dieser Vorgang~ immer vollständig bis zur Endlage.
2. Die Sensoren LSH, LSV und NTZ werden ignoriert, solange die Tür sich öffnet.
3. Falls die Tür still steht, öffnet sie sich, sobald NTA betätigt wurde.
4. Die Tür schließt sich vollständig bis zur Endlage, sobald NTZ betätigt wurde.
5. Einklemmschutz: mindestens ein aktiver Sensor LSH, LSV oder NTA, stoppt die Tür sofort, falls diese sich gerade schließt.
6. Die Sensoren LSH und LSV werden ignoriert, falls die Tür sich nicht bewegt.

⁴Hinweis: dies ist keine Praktikumsdokumentation wie in Abschnitt 4.2 beschrieben.

5.2.3 Reparaturbetrieb

In diesem Betriebsmodus ist BW1 inaktiv und BW2 aktiv.

1. Die Tür bewegt sich nur, solange exakt ein Sensor (NTA, NTZ, LSH oder LSV) aktiv ist.
2. NTZ und LSV schließen die Tür.
3. NTA und LSH öffnen die Tür.
4. Sind keine oder mehr als einer der oben genannten Sensoren aktiv, bleibt die Tür sofort stehen.
5. Die Signalleuchte soll aktiviert sein, solange die Tür sich bewegt.

5.2.4 Ausgeschaltet

Wenn BW1 und BW2 beide inaktiv sind soll die Anlage den Zustand abgeschaltet haben.

1. Die Tür reagiert auf keine Sensoren, außer auf BW1 und BW2.

6 Vorschläge zu Design und Entwicklung der Steuerung (optional)

- Implementieren Sie eine Funktion, welche anhand des Pin-Wertes und der Information über die High-/Low-aktiv-Konfiguration entscheidet, ob ein Sensor aktiv oder inaktiv ist.
- Überlegen Sie gemeinsam im Team, welche Zustände die Tür im Zustandsautomat haben kann.
- Schreiben Sie ein eigenständiges C-Modul für das Logging, welches sich selbstständig um die Ausgabe in eine Protokolldatei kümmert.
- Möglichkeit um einen Sekundengenauen Zeitstempel zu erzeugen

```
#include <stdio.h>
#include <time.h>
//[...]
time_t zeitstempel = time(NULL);
printf("%s", ctime(&zeitstempel));
```

- Behandeln Sie alle Fehler, welche beim Aufrufen von (Bibliotheks-) Funktionen auftreten können. Unbehandelte Fehler (z.B.: bei `fopen`) können zum Programmabsturz führen.
- Lesen Sie die Sensor-Konfiguration aus einer Textdatei um flexibel unterschiedliche Türkonfigurationen zu beherrschen.
 - Vorteilhaft ist ein menschenlesbares Format für die Konfigurationsdatei um Änderungen und Prüfungen an der Konfiguration mit einem Texteditor vornehmen zu können.
- Testen Sie Ideen zur Implementierung in einem separaten minimalen Programmquellcode und erforschen Sie dessen Verhalten, bevor Sie die Ideen in den Quellcode Ihrer Steuerungssoftware einbauen.
- Konzentrieren Sie sich auf die korrekte Funktionsweise Ihres Quellcodes und vermeiden Sie Speicher- und Geschwindigkeitsoptimierungen solange, bis Sie die komplette Funktionalität implementiert haben.
- Halten Sie einzelne Funktionen kurz und übersichtlich. Teilen Sie lange komplizierte Funktionen besser auf mehrere Funktionen auf. Funktionen deren Quellcode nicht ohne zu scrollen auf dem Bildschirm angezeigt werden können, sind zu lang und sollten aufgeteilt werden.
- Es ist ihnen freigestellt, welche Entwicklungswerkzeuge Sie verwenden. Wichtig ist, dass Ihre Steuerungssoftware in der VM kompiliert und läuft und Sie sich mit Ihrem Quellcode auskennen.

6.1 Hinweise zur Inbetriebnahme des vorbereiteten Programmrahmens

Bitte benutzen Sie die angebotene virtuelle Maschine (VM) (siehe [Downloads](#)). Diese VM ist bereits vorbereitet, alle benötigten Softwarekomponenten (Bibliotheken, Compiler, IDE) sind darin installiert und wurden auf Funktion mit der Aufgabenstellung getestet.

Bei dem vorbereiteten Programmrahmen handelt es sich um ein komplettes Quellcode-Projekt. Sie können die Dateien mit jedem beliebigen Texteditor öffnen und bearbeiten. Kompilieren Sie das Projekt anschließend mit dem Befehl `make` über die Kommandozeile im Projektverzeichnis.

Nach erfolgreichem Kompilieren per `make`, können Sie das Programm starten. geben Sie dazu `make run` oder `./_tuersteuerung` auf der Kommandozeile ein.

Alternativ ist das Projekt auch für die IDE `Code::Blocks` eingerichtet. Um diese zu nutzen, klicken Sie, in der VM im Projektverzeichnis, doppelt auf die Datei `Türsteuerung.cbp`. Das `Code::Blocks`-Projekt ist so konfiguriert, dass es das Makefile benutzt. Anleitungen zur Benutzung der IDE finden Sie im Internet.

Um weitere eigene Quellcodedateien ins Projekt aufzunehmen, editieren Sie bitte das `Makefile`. Entsprechende Kommentare zur Hilfe finden Sie ebenfalls im `Makefile`.

6.1.1 FAQ:

- *Wie bekomme ich Dateien in die VM?:* Sie haben mehrere Möglichkeiten Dateien mit der VM auszutauschen. Über die Einstellungen der VM, können Sie vor dem Start der VM einen `Gemeinsamen Ordner` festlegen, welcher dann auf Ihrem PC und in der VM sichtbar ist. Sie können auch einen physischen USB-Massenspeicher über das `Geräte`-Menü in die VM einbinden. Sie können mit `git` arbeiten und ein selbst angelegtes Repository klonen. Sie können die Daten mittels `wget` von einem Web-Server herunterladen.

7 Hinweise zum Praktikumstermin und dessen Durchführung

7.1 Vorbereitung

- Laden sie den vorbereiteten Programmrahmen herunter. (siehe Abschnitt [Downloads](#))
- Stellen Sie Ihre Steuerungssoftware als Gruppenarbeit zu Hause, mit Hilfe der VM (siehe [Downloads](#)), fertig.
- Bringen Sie den von Ihnen so vorbereiteten Quellcode zur Versuchsdurchführung mit. (*Hinweis:* Archivierungsdateiformat nutzen, z.B.: `zip` oder `tar.gz` um die Datenintegrität zu gewährleisten)
- Fragen zur Versuchsaufgabe und Programmierung oder bei Problemen bezüglich der Lösung der Aufgabe können Sie per E-Mail an M.Herhold⁵ (Lehrstuhl für Automatisierungstechnik) stellen. Klar formulierte Fragen führen nicht zu Punktabzug bei der Leistungsbewertung. Die Beantwortung erfolgt während der regulären Büroarbeitszeiten.

7.2 Aufbau und Nutzung des Versuchsplatzes

Der Versuchsplatz ist ein PC mit Linux und vorhandener Internetanbindung. Auf dem System ist die VM installiert und betriebsbereit. Das Türmodell ist angeschlossen und betriebsbereit. Ihr Quellcode muss zum Bestehen des Versuchs in der VM kompilieren und das Programm in der VM laufen.

Die Übertragung Ihres Quellcodes ist per USB-Stick oder Download aus dem Internet möglich.

⁵mario.herhold@tt.tu-dresden.de

7.3 Aufgaben während der praktischen Versuchsdurchführung zum Praktikumstermin

Absolvierung eines 20 minütigen schriftlichen Eingangstests, individuelle Bewertung.

Passen Sie Ihre vorbereitete, fertige und in der Simulation erprobte Steuerungssoftware an die reale Hardware und deren abweichende Sensor-Konfiguration an. Erproben Sie systematisch die Funktion ihrer Steuerungssoftware am Türmodell.

Präsentieren Sie Ihr Anwendungsprogramm dem anwesenden Betreuer und beantworten Sie Fragen zu Ihrem Quellcode. Aufgrund der KI-Problematik muss jedes Gruppenmitglied zum Bestehen selbständig signifikante Änderungen an Ihrer Steuerungssoftware vornehmen können. Nach erfolgreicher Präsentation Ihrer Anwendung, kann der Versuchstermin vorzeitig beendet werden. Die Präsentation geht in die Bewertung des Praktikumsversuchs ein.

Der so entstandene Quellcode verbleibt in Kopie, für Archivierungs-, Plagiatprüfungs- und Bewertungszwecke, beim Praktikumsbetreuer.

7.4 Dokumentation

Zum Bestehen des Praktikums ist von jeder Praktikumsgruppe eine gemeinsame Dokumentation, innerhalb von 2 Wochen nach dem Versuchstermin, abzugeben.

Inhalt der Dokumentation, siehe Aufgabenstellung im Abschnitt 4.2.

Die Dokumentation ist zusammen mit dem zu verwendenden Deckblatt (siehe [Downloads](#)), im OPAL-Kurs SwEngV zum Versuch ART1, als [PDF](#), hochzuladen.

Die Dokumentation fließt in die Bewertung des Praktikumsversuchs ein.

8 Downloads

Virtuelle Maschine, Programmrahmen und Dokumentationsdeckblatt siehe [Webseite des Lehrstuhls für Automatisierungstechnik zum Praktikumsversuch Software Engineering Vertiefung / Mikrorechentchnik 2, ART-1 und ART-3](#)

9 Arbeits- und Brandschutzhinweise

9.1 Vorbeugende Maßnahmen:

- Die Praktikumssteilnehmer haben sich so zu verhalten, dass Gefahrensituationen und Unfälle vermieden werden.
- Die Befugnis zum Bedienen und Nutzen von Geräten ist auf den zugewiesenen Praktikumsplatz beschränkt.
- Eingriffe in die zum Praktikumsaufbau gehörenden Geräte sind nicht erlaubt.
- Der Anschluss und der Betrieb privater Geräte in den Praktikumsräumen ist verboten.
- Defekte an Geräten oder Gebäudeeinrichtungen sind unverzüglich dem Betreuer mitzuteilen. Betroffene Geräte sind außer Betrieb zu nehmen. Andere Personen sind vor Gefahren zu warnen.
- Den Anweisungen der Praktikumsbetreuer bzw. anderer aufsichtsführender Personen ist unbedingt Folge zu leisten.
- Betriebsfremde dürfen sich nur mit Erlaubnis des Praktikumsbetreuers in den Praktikumsräumen aufhalten.
- Rauchen und Umgang mit offenem Feuer ist nicht gestattet.
- Nach Ende des Praktikums ist der Arbeitsplatz in sauberem und aufgeräumtem Zustand zu hinterlassen.
- Außergewöhnliche Ereignisse bzw. besondere Vorkommnisse sind umgehend dem Betreuer oder dem diensthabenden Assistenten zu melden.

9.2 Verhalten im Falle eines Brandes:

- Beachten der richtigen Reihenfolge: **MELDEN - RETTEN - LÖSCHEN**

9.2.1 Feuer melden:

- Telefonische Brandmeldung:
 - Notruf 112 der Feuerwehr (von jedem Telefon aus möglich)
 - Notruf HA 34515 der Technischen Leitzentrale der TUD
- Deutliche, genaue und vollständige Angaben:
- Wo brennt es?
- Was brennt?
- Angaben zu verletzten oder gefährdeten Personen
- Wer meldet?

9.2.2 Personen retten:

- Erste Hilfe leisten
- Weitere Hilfe organisieren, medizinische Hilfe anfordern
- Gefahrenbereich räumen; Fluchtwege benutzen, keine Aufzüge
- Andere Personen warnen, Sammelplatz (Platz vor Turmeingang zum Barkhausenbau) aufsuchen
- Behinderten und älteren Personen helfen

9.2.3 Löschversuch unternehmen

- Feuerlöscher verwenden (Standorte: Gänge des Barkhausenbaues), dabei sich nicht selbst gefährden
- Fenster und Türen schließen, aber nicht abschließen
- Möglichst elektrische Verbraucher abschalten

9.3 Rufnummern für Notfälle:

Helper	Telefonnummer
Rettungsdienst	112
Polizei	110
TUD-Notruf	34515
Betriebsärztlicher Dienst	36199
Klinikum Friedrichstadt Notaufnahme	0351-480 1938