

Praktikum zur Lehrveranstaltung Mikrorechentechnik II Versuch: ART1 Türsteuerung

Organisation des Praktikumsversuchs: M. Herhold

Inhaltsverzeichnis

1	Versuchsziel	1
2	Einführung	1
3	Beschreibung des gegebenen 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	7
9	Programmierschnittstelle der DoorInterface-Klasse	8
10	Arbeits- und Brandschutzhinweise	8

1 Versuchsziel

Versuchsziel ist die Festigung der Vorlesungskennnisse der objektorientierten Programmierung mittels der Programmiersprache C++. Das Praktikum umfasst den Entwurf und die Implementierung einer Steuerung einer automatischen Tür.

2 Einführung

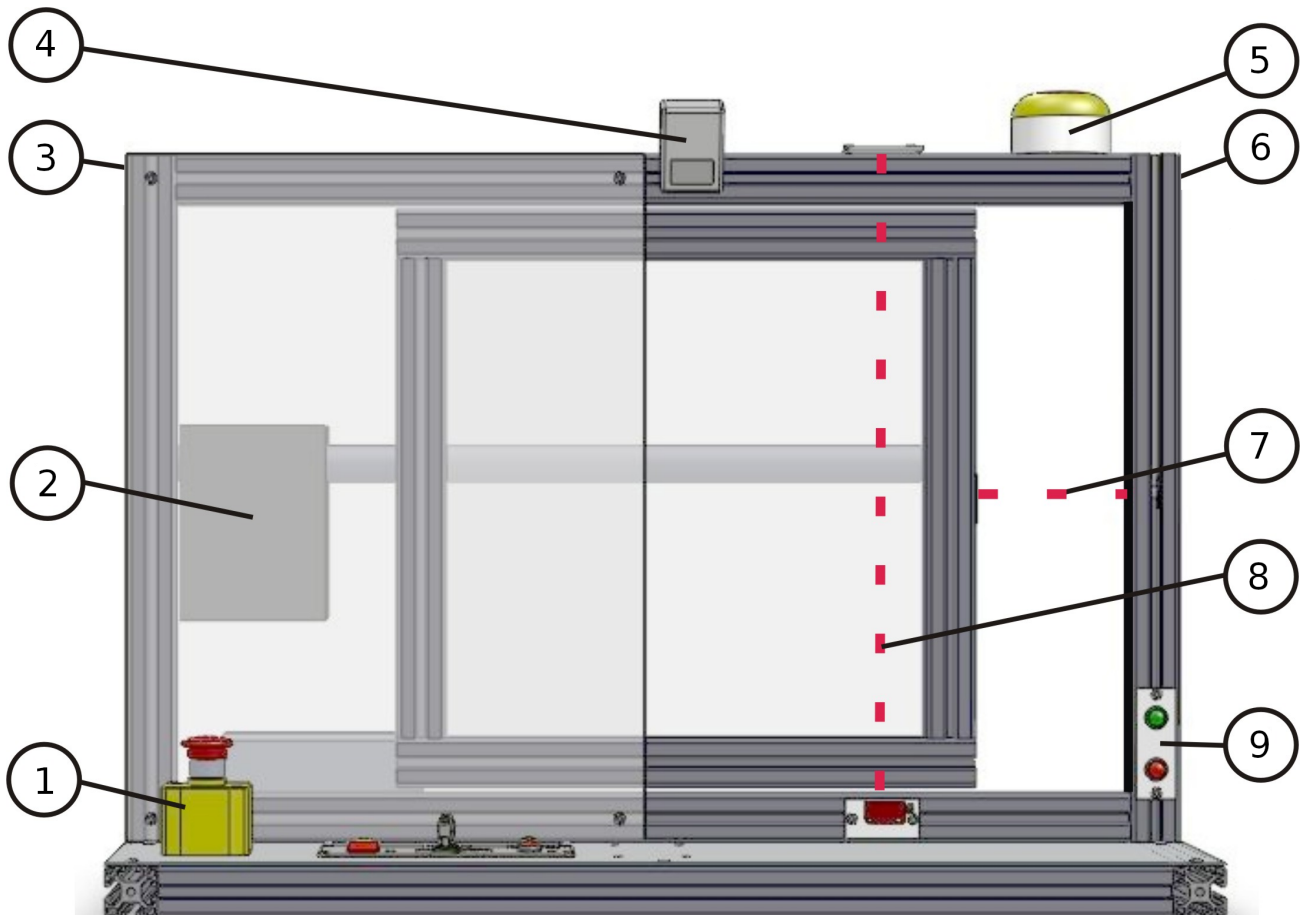
Automatisch bewegte Türen können z.B. bei Fahrzeugen, Garagen, Geschäften, Schutz- und Sicherheitsräumen eingesetzt werden. Automatisierungsziele können u.a. sein, die Passagezeiten zu verringern, den Sicherheitsstandard und einen Komfort nach dem Stand der Technik zu erfüllen. Teilaufgaben für die Automatisierung 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 die Wart- und Erweiterbarkeit der Software besonders Wert gelegt.

3 Beschreibung des gegebenen 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 in die, aus der Vorlesung und dem Wintersemester bekannten, MRT-Appliance¹ eingebunden. Die technischen Details zur Ansteuerung der Hardware sind im vorgegebenen Programmrahmen bereits fertig implementiert.



Legende Türmodell:

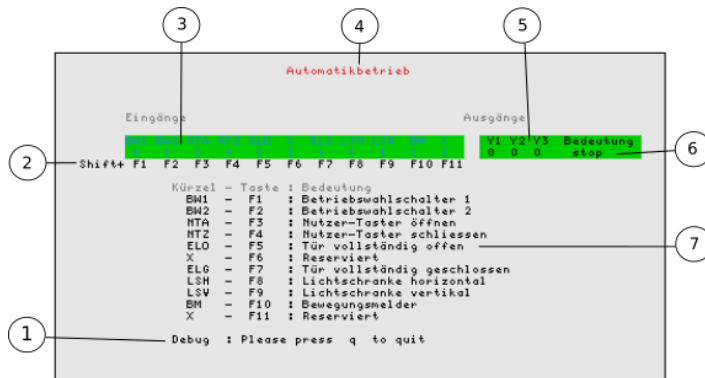
1. Betriebswahlschalter BW1 und BW2 (und Notaus)
2. Antriebseinheit Y1 und Y2
3. Tür-Endlagesensor Offen ELO
4. Bewegungsmelder BM
5. Signal-Blinkleuchte Y3
6. Tür-Endlagesensor Geschlossen ELG
7. Lichtschanke horizontal LSH
8. Lichtschanke vertikal LSV
9. Nutzer-Taster zum Öffnen NTA und Schließen NTZ

Die Signalleuchte blinkt selbstständig, sobald Y3 aktiviert ist. Der Notausschalter ist nicht per Software auslesbar.

Das Verhalten der Hardware wird vollständig von der Steuersoftware bestimmt. Somit ist bei fehlerhafter Steuersoftware ein Fehlverhalten der Hardware möglich. Das Türmodell ist so konstruiert, dass die Hardware keinen Schaden nimmt.

Der vorgegebene Programmrahmen simuliert ein technisch eingeschränktes Gerätedisplay.

¹im nachfolgenden Text nur noch als VM bezeichnet



Legende Gerätedisplay:

1. einzeilige Ausgabemöglichkeit für die Steuersoftware, nutzbar mit `DoorInterface::DebugString()`
2. Tastenkürzel zum Ändern der Sensorzustände im Simulationsmodus
3. Auflistung der Eingänge und deren aktueller Zustand, Kennung und Pegel
4. Anzeige des aktuellen Betriebszustands, abhängig von `BW1` und `BW2`
5. Auflistung der Ausgänge und deren aktueller Zustand
6. im Simulationsmodus Anzeige der Motorbewegung und Signalleuchte
7. Kurzbeschreibung der Eingänge

Der Zustand der Sensor-Eingänge und Aktor-Ausgänge wird durch einen 0- oder 1-Pegel auf dem entsprechenden Port am entsprechenden Pin markiert. Pin 0 ist nach dem Aufruf von `DoorInterface::DIO_Read()` auf dem niederwertigsten Bit zu finden.

Port	Pin	Name	Sensor	Aktor
0	0	BW1	X	
0	1	BW2	X	
0	2	NTA	X	
0	3	NTZ	X	
0	4	ELO	X	
0	5	reserviert	X	
0	6	ELG	X	
0	7	LSH	X	
1	0	LSV	X	
1	1	BM	X	
1	2	reserviert	X	
1	3	reserviert	X	
1	4	reserviert	X	
1	5	reserviert	X	
1	6	reserviert	X	
1	7	reserviert	X	
2	0	Y1		X
2	1	Y2		X
2	2	Y3		X
2	3	reserviert		X
2	4	reserviert		X
2	5	reserviert		X
2	6	reserviert		X
2	7	reserviert		X

4 Aufgabenstellung

Entwerfen, entwickeln und dokumentieren sie im Team eine Steuerungssoftware für eine automatische Tür. Das gewünschte Verhalten der Tür wird im Abschnitt 5 beschrieben.

4.1 Entwurf und Design

Verfolgen sie einen **objektorientierten Ansatz** und realisieren sie die einzelnen **Sensoren & Aktoren als Instanzen** von Klassen (mit einer sinnvollen Vererbungshierarchie)!

Erweitern sie die Klasse `DoorControl`. Diese Klasse stellt, durch Ableitung von der Klasse `DoorInterface`, ihre Schnittstelle zur Hardware und Simulationsumgebung dar.

Entwerfen sie eine **zusätzliche Klasse**, welche Details **zur Sensorkonfiguration** aus einer Datei einliest und die Eigenschaft (High- oder Low-aktiv) der einzelnen Sensoren entsprechend verändert.

4.2 Entwicklung und Inbetriebnahme

Stellen sie die Steuerung 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 sollen ihrer Software hinzugefügt werden.
Die Steuerungssoftware muss in der VM² kompilieren und lauffähig sein.

4.3 Abschließende Dokumentation ihrer Arbeit

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).
Dokumentieren sie ihre Arbeit in Form von UML2 Zustands- und Klassendiagrammen.
Analysieren und dokumentieren sie Vor- und Nachteile ihrer Lösung in Hinsicht auf zukünftige Verwendung und Erweiterbarkeit.
Diese Dokumentation ist einzureichen, per E-Mail an den Praktikumsbetreuer, innerhalb von 1 Woche nach dem Praktikumstermin, in Form eines PDF-Dokuments.

5 Beschreibung Türverhalten

Hinweis: bei folgender Beschreibung der Tür, handelt es sich um die Aussagen des gedachten zukünftigen Betreiber der Tür. 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 Anzahl der Sensoren und Aktoren

Es ist damit zu rechnen, dass zusätzliche noch unbekannte Sensoren oder Aktoren der Hardware (während des Praktikumstermins) hinzugefügt werden, welche Einfluss auf das Verhalten der Tür nehmen sollen. Diesem Umstand ist beim Entwurf der Steuerungssoftware Rechnung zu tragen. Die Software soll mit nur einem geringen Maß an Programmierung an die neuen Sensoren/Aktoren angepasst werden können.

5.2 Interpretation der Sensorzustandswerte

Normalerweise sind die Sensoren und Aktoren High-aktiv (1-Pegel bedeutet aktiv), es gibt aber auch Hardwaremodelle der Tür bei denen einzelne oder alle Sensoren Low-Aktiv (0-Pegel bedeutet aktiv) sind. Die Steuerungssoftware soll diesem Umstand Rechnung tragen, indem sie sich dieses Detail für jeden Sensor per Konfigurationsdatei konfigurieren lässt. Die Konfigurationsdatei soll beim Programmstart einmalig eingelesen und beachtet werden. Die konkrete Strukturierung und Inhalt der Konfigurationsdatei ist dem Entwicklerteam überlassen.

5.3 Aktorzustandswerte

sind bei jedem Modell High-aktiv.

5.4 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 Y1 und Y2 gleichzeitig aktiv zu schalten, ist ein ungültiger Zustand und soll unbedingt vermieden werden.

²MRT-Appliance

5.5 Betriebsarten:

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

- Automatikbetrieb, Handbetrieb, Reparaturbetrieb, Ausgeschaltet

5.5.1 Automatikbetrieb

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

1. 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:
2. Sobald zu einem beliebigen Zeitpunkt an einem oder mehreren Sensoren NTA, LSH, LSV oder BM ein aktiver Zustand detektiert wurde, öffnet sich die Tür bis diese den vollständig geöffneten Zustand erreicht hat.
3. Nachdem die Tür vollständig geöffnet wurde, verharrt diese für 3 Sekunden in dieser Position, bevor sich die Tür selbständig schließt. Aktive Zustände an den Sensoren NTA, LSH, LSV oder BM starten diesen 3 Sekunden Timer sofort von vorn.
4. Ein Betätigen des Tasters NTZ soll zum augenblicklichen Schließen der Tür führen. Die Tür soll sich unabhängig davon, ob NTZ gedrückt bleibt oder nicht, vollständig schließen. Automatikbetrieb-Regel 2 ist zu beachten.

5.5.2 Handbetrieb:

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

1. Die Sensoren LSH, LSV und BM werden ignoriert.
2. Die Tür öffnet sich nur, solange NTA betätigt ist.
3. Die Tür schließt sich nur, solange NTZ betätigt ist.
4. Wenn keiner der Taster betätigt wird, bleibt die Tür stehen.
5. Der Motor ist abzuschalten, sobald eine Endlage (Tür vollständig offen oder vollständig geschlossen) erreicht ist.

5.5.3 Reparaturbetrieb

In diesem Betriebsmodus ist BW1 inaktiv und BW2 aktiv.

1. Generelles Verhalten wie im Automatikbetrieb. Die Tür bewegt sich allerdings nur solange, wie ein Sensor (NTA, NTZ, LSH, LSV, BW) aktiv ist.
2. Sind keine der oben genannten Sensoren aktiv, bleibt die Tür sofort stehen.
3. Auf das 5 sekundige Aktivieren der Signalleuchte beim Start des Betriebsmodus wird verzichtet (Automatikbetrieb-Regel 1).
4. Die Signalleuchte soll bei jeder Türbewegung aktiviert werden.

5.5.4 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)

- Überlegen sie sich gemeinsam im Team, welche Klassen und welche Hierarchie sinnvoll ist. Verteilen sie die so entstandenen Klassen auf Teammitglieder und spielen sie in einer Art Rollenspiel das Zusammenwirken der Klassen während des Programmablaufs durch.
- Es ist sinnvoll auf die in der Vorlesung vorgestellten Lösungen (Automat, Singleton-Design-Pattern) zurückzugreifen um die geforderte Flexibilität der Software zu erreichen.
- Die Instanz eines Sensors kann intelligent mit Hilfe von `get()`-/`set()`-Methoden entscheiden, ob der Sensor aktiv ist. Durch diese Herangehensweise ist es einfach auf die unterschiedliche Interpretation der Sensorwerte im Entwurf und der Realisierung zu reagieren. (Stichwort Datenkapselung)
- Die Klasse, welche die Konfigurationsdatei verarbeitet, könnte die Instanzen der Sensoren mit der entsprechenden Konfiguration erzeugen.
- Vorteilhaft ist ein menschenlesbares Format für die Konfigurationsdatei um Änderungen und Prüfungen an der Konfiguration mit einem Texteditor vornehmen zu können.
- Arbeiten sie mit den Containern welche C++ (zB. `std::list`) anbietet um eine variierende Anzahl aktiver Sensoren zu beherrschen.
- Aufgrund der eingeschränkten Fähigkeiten des (simulierten) Displays könnte es angebracht sein, Zustände und Debugausgaben zur Fehleranalyse in einer Datei zu speichern. Ausgaben mittels `printf()` oder `std::cout` « stören das simulierte Display.
- Halten sie einzelne Funktionen/Methoden kurz und übersichtlich. Teilen sie lange komplizierte Funktionen/Methoden besser auf mehrere Funktionen/Methoden auf.
- Es ist ihnen freigestellt welche Entwicklungswerkzeuge sie verwenden. Wichtig ist, dass die Steuersoftware in der VM kompiliert und läuft.

6.1 Hinweise zur Inbetriebnahme des vorbereiteten Programmrahmens

Bei dem vorbereiteten Programmrahmen handelt es sich um einen kompletten Eclipse-Workspace, welcher, nach dem Entpacken, über folgenden Menüpunkt in Eclipse geladen werden kann:

"File" → "Switch Workspace" → "Other" ... (dann Auswahl des Workspace-Verzeichnisses und Bestätigen mit "OK")

Nach erfolgreichem Kompilieren in Eclipse oder per `Makefile`, können sie das Programm starten. Das fertig kompilierte Programm muss in einem laufenden Terminal-Programm (zB. `xterm`) gestartet werden. Der Menüpunkt External Tools → run in `xterm` hilft ihnen dabei.

Nach dem Start des Programms sehen sie das simulierte Gerätedisplay.

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.
- *Warum kann ich mein Programm in Eclipse in der VM nicht debuggen?:* Installieren sie den GNU-Debugger indem sie das Paket `gdb` in der VM installieren.
Shell Kommando: `sudo apt-get update; sudo apt-get install gdb`

7 Hinweise zum Praktikumstermin und dessen Durchführung

7.1 Vorbereitung

- Arbeiten sie die Aufgabenstellung und die Arbeitssicherheitshinweise in diesem Dokument durch.
- Laden sie den vorbereiteten Programmrahmen (Eclipse Workspace) herunter. (siehe Abschnitt [Downloads](#))
- Stellen sie ihre Steuerungssoftware zu Hause, mit Hilfe der VM (siehe [Downloads](#)), soweit wie möglich fertig.
- Bringen sie den von Ihnen so vorbereiteten Eclipse Workspace 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.Herhold (Lehrstuhl für Automatisierungstechnik) stellen, die Beantwortung erfolgt während der regulären Büroarbeitszeiten.

7.2 Aufbau und Nutzung des Versuchsplatzes

Der Versuchsplatz ist ein PC mit Linux und vorhandener (leicht eingeschränkter) Internetanbindung. Im System ist die VM installiert und betriebsbereit. Das Türmodell ist angeschlossen und betriebsbereit.

Die Übertragung ihres Quellcodes ist per USB-Stick oder Download aus dem Internet möglich. Versehen sie das Verzeichnis mit ihrer Gruppennummer zB. `workspace_GrNr-xx`.

7.3 Ablauf der praktischen Versuchsdurchführung zum Praktikumstermin

Passen sie ihre Steuerungssoftware an die reale Hardware und deren abweichendes Sensorverhalten an. Erproben sie systematisch die Funktion ihrer Steuerungssoftware. Führen sie einen Nachweis zur korrekten Arbeitsweise der einzelnen Komponenten aus!

Präsentieren sie ihr Anwendungsprogramm dem anwesenden Betreuer und beantworten sie Fragen zu ihrer Lösung. 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- und Bewertungszwecke, beim Praktikumsbetreuer.

7.3.1 Kontrollfragen zur Präsentation

- Begründen sie den Entwurf ihrer programmiertechnischen Lösung!
- Begründen und erklären sie Implementierungsdetails!
- Erklären sie ihre Vorgehensweise zum Test der korrekten Implementierung!

7.4 Dokumentation

Zum Bestehen des Praktikums ist von jeder Praktikumsgruppe eine gemeinsame Dokumentation, innerhalb von 1 Woche nach dem Versuchstermin, abzugeben.

Inhalt der Dokumentation, siehe Aufgabenstellung.

Die Dokumentation ist zusammen mit dem zu verwendenden Deckblatt, per E-Mail an M.Herhold, im PDF-Format einzureichen.

Die Dokumentation fließt in die Bewertung des Praktikumsversuchs ein.

8 Downloads

VirtualBox Appliance (VM) siehe Opal-Ressourcen zur Vorlesung Mikrorechentchnik 1 & 2

Programmrahmen (Eclipse Workspace) und Dokumentationsdeckblatt siehe [Webseite des Lehrstuhls für Automatisierungstechnik zum Praktikum Mikrorechentchnik 2, ART-1 und ART-3](#)

9 Programmierschnittstelle der DoorInterface-Klasse

```
/*
 * Boolesche Variable zum Detektieren des gewünschten Programmabbruchs
 * 'true' bedeutet der Programmabbruch ist gewünscht
 * den Variablenwert regelmäßig prüfen! */
static std::atomic<bool> quit_doorcontrol_flag;

/*
 * Konstruktor des Interface zur Hardware / Simulation
 * real_door: legt fest ob auf die Hardware zugegriffen werden soll
 * show_ui: legt fest, ob die Ein-/Ausgabe ueber ncurses genutzt wird
 * wirft Exception 'std::runtime_error' im Fehlerfall */
DoorInterface(bool real_door=false, bool show_ui=true);

/*
 * Liest alle Pins eines Eingabe-Ports.
 * port: Port-Nummer
 * bits: ausgelesenen Kanäle des Ports (Rueckgabewert) */
void DIO_Read(const unsigned port, unsigned char *pins);

/*
 * Aendert alle Pins eines Ausgabe-Ports.
 * port: Ausgabe-Port-Nummer
 * pins: die auszugebenden Kanäle des Ports */
void DIO_Write(const unsigned port, const unsigned char pins);

/*
 * Methode zum Anzeigen eines Strings in der Nachrichtenzeile des Displays
 * s: auszugebender String (maximale Laenge 50 alphanumerische Zeichen) */
void DebugString(const std::string s);
```

10 Arbeits- und Brandschutzhinweise

10.1 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.
- Die [aktuell gültigen Verhaltensregeln in Bezug auf die COVID 19 Erkrankung](#) sind einzuhalten.

10.2 Verhalten im Falle eines Brandes:

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

10.2.1 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?

10.2.2 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

10.2.3 Löschversuch unternehmen

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

10.3 Rufnummern für Notfälle:

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