

Echtzeitbetriebssysteme Übung/Praktikum (Sommersemester 2016)

Oliver Jack
Ernst-Abbe-Hochschule Jena
Fachbereich Elektrotechnik und Informationstechnik
April 2016

Inhaltsverzeichnis

1 Organisatorisches	2
2 Bewertung	2
3 Hinweise zur Bearbeitung	3
4 Hochregallager-Steuerung	4
4.1 Das Hochregallager-Modell	4
4.1.1 Sensoren	4
4.1.2 Aktoren	5
4.2 Das Aktor-Sensor-Interface (ASI) EN 50295 und IEC 62026-2	6
4.3 Die Steuerung	7
4.4 Zusatz: Automatisierung	7

1 Organisatorisches

- Bilden Sie Arbeitsgruppen gemäß Seteinteilung.
- Lösen Sie die in Abschnitt 4 beschriebene Aufgabe unter Verwendung von *VxWorks* und der zugehörigen Entwicklungsumgebung *Wind River Workbench* in Gruppenarbeit bis zum vorletzten Praktikumstermin gemäß Praktikumsgruppenplan.
- Präsentieren Sie Ihre Lösung. Jedes Arbeitsgruppenmitglied präsentiert einen Teil der Lösung. Für die Präsentation stehen pro Gruppe 15 Minuten zur Verfügung. Anschließend stehen fünf Minuten für Fragen aus dem Auditorium, einschließlich Dozent, zur Verfügung.
- Die Präsentationen finden am letzten Praktikumstermin statt. Die Termineinteilung für die Projektgruppen wird rechtzeitig bekanntgegeben.
- Geben Sie pro Arbeitsgruppe bis zum Präsentationstermin
 1. eine schriftliche Ausarbeitung (Folien) Ihrer Präsentation,
 2. den Quell-Code und
 3. die Entwicklungsdokumentation ab,per E-Mail an Oliver.Jack@eah-jena.de mit dem Betreff „APL Echtzeitbetriebssysteme SS 2016“. Geben Sie die Zuordnung der Arbeitspakete zu den einzelnen Arbeitsgruppenmitgliedern an.

2 Bewertung

- Darstellung des Entwurfs (20 %)
- Funktionsfähigkeit der Lösung (20 %)
- Realisierung mit Echtzeitbetriebssystem-Konzepten (20 %)
- Verständlichkeit der Implementierung (20 %)
- Entwicklungsdokumentation (20 %)

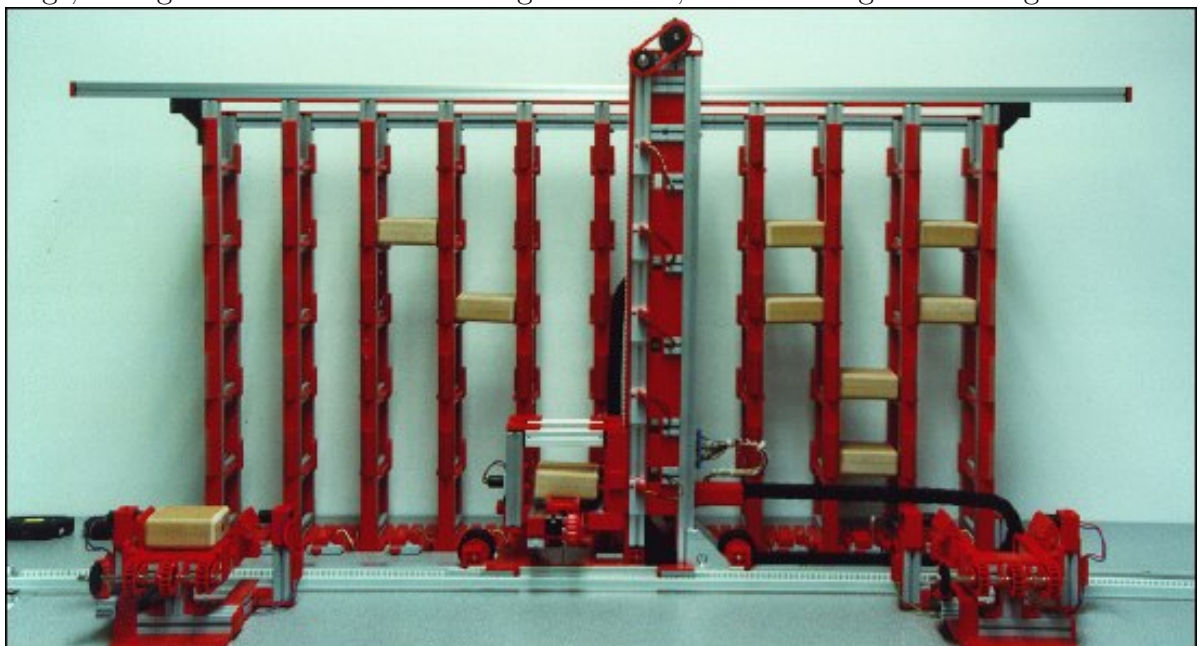
3 Hinweise zur Bearbeitung

- Beginnen Sie bei der Bearbeitung der Aufgabe mit der Modellierung der des Systems. Empfehlenswert für die Modellierung ist die Strukturierte Analyse und die Real-time Analyse.
- Modellieren Sie gegebenenfalls relevante Prozesse (Tasks, Threads) mit Zustandsautomaten.
- Erstellen Sie ein Gantt-Diagramm für den Prozessablauf.
- Die anzusteuern Hardware ist nicht vorhanden. Simulieren Sie soweit sinnvoll, das heißt bis zu einem vertretbaren Detailaufwand, die anzusteuern Hardware. Realisieren Sie die Simulation durch entsprechende Prozesse, die nebenläufig zu Ihrem Steuerungssystem.
- Teilen Sie die Bearbeitung der Aufgabe auf die einzelnen Gruppenmitglieder auf, nachdem Sie die Systemarchitektur und die Schnittstellen der Systemkomponenten festgelegt haben, so dass die Implementierung der einzelnen Komponenten arbeitsteilig erfolgen kann.
- Machen Sie sich vor der Implementierung Gedanken, mit welchen Testfällen Sie Ihr Programm sowohl während der Entwicklung als auch zum Abnahmetest prüfen und dokumentieren Sie diese Testfälle zum Beispiel in Form einer Checkliste. Dokumentieren Sie ebenfalls die Gründe, warum Sie diese Testfälle gewählt haben und welchen Aussagewert sie für die Qualität Ihres Programms haben. Dokumentieren Sie Ihre Testergebnisse.

4 Hochregallager-Steuerung

4.1 Das Hochregallager-Modell

Das Modell besteht aus einem Bereich mit Ablagefächern ($5 * 10 = 50$), einem Schlitten um Klötzchen zu transportieren und zwei Ein-/Ausgabeslots (der Eingabeslot ist links). Ein Turm, der den Schlitten in Y-Richtung (hoch/runter) bewegt, kann auf einer Schiene entlang des Regals in X-Richtung bewegt werden. Im Schlitten befindet sich ein Ausleger, der in Z-Richtung bewegt werden kann. Die für diese Operationen benötigten Aktoren/Sensoren sind über ein ASI-Bus mit dem Microcontroller verbunden. Es ist darauf zu achten, nur gültige Operationen durchzuführen. Wird der Turm zum Beispiel bewegt, solange der Schlitten sich im Regal befindet, kann das Regal beschädigt werden.



4.1.1 Sensoren

Die Position des Schlitten wird über verschiedene Taster bestimmt, die beim „Überqueren“ ausgelöst werden. Ein Taster ist gedrückt, wenn das entsprechende Bit auf 0 gesetzt ist (negative Logik). Für jede Raumachse existieren eine bestimmte Anzahl von Schaltern:

- X-Achse: 10 Schalter mit den Positionen 1-10: Ein Schalter ist gedrückt, wenn der Schlitten die optimale Position in einer Regalspalte erreicht hat. Position 1 befindet sich „links“ bzw Position 10 befindet sich „rechts“.
- Y-Achse: Jeweils 2 Schalter für ein Ablagefach. Einer oberhalb, einer unterhalb des Ablagefachs, um ein Ablegen beziehungsweise Aufnehmen der Klötzchen zu ermöglichen. Position 1 der Ablagefächer befindet sich „oben“, Position 5 befindet sich „unten“.
- Z-Achse: 3 Schalter für die Positionen im Regal, in der Mitte (also im Turm) und im Ein-/Ausgabeslot.

Zusätzlich gibt es noch jeweils eine Lichtschranke für die Ein-/Ausgabeslots und einen Lichttaster im Schlitten, um deren Belegungszustände zu erfassen.

Die Zuordnung der Adressen zu den Sensoren ist in Tabelle 1 dargestellt.

ASI Adresse	Stecker	Slavebit 3	Slavebit 2	Slavebit 1	Slavebit 0
Slave 1	X2-12	Y über Pos. 5	Y unter Pos. 5	Y über Pos. 4	Y unter Pos. 4
Slave 2	X2-13	Y über Pos. 3	Y unter Pos. 3	Y über Pos. 2	Y unter Pos. 2
Slave 3	X2-14	Z Regal	Z Mitte	Y über Pos. 1	Y unter Pos. 1
Slave 4	X2-15	-	-	Lichttaster aus	Z E/A-Station
Slave 5	X2-31	X Pos. 8	X Pos. 7	X Pos. 6	X Pos. 5
Slave 6	X2-32	X Pos. 4	X Pos. 3	X Pos. 2	X Pos. 1
Slave 7	X2-33	X Pos. 10	Lichtschranke R	Lichtschranke L	X Pos. 9

Tabelle 1: Zuordnung der ASI-Slaveadressen zu den Sensoren

4.1.2 Aktoren

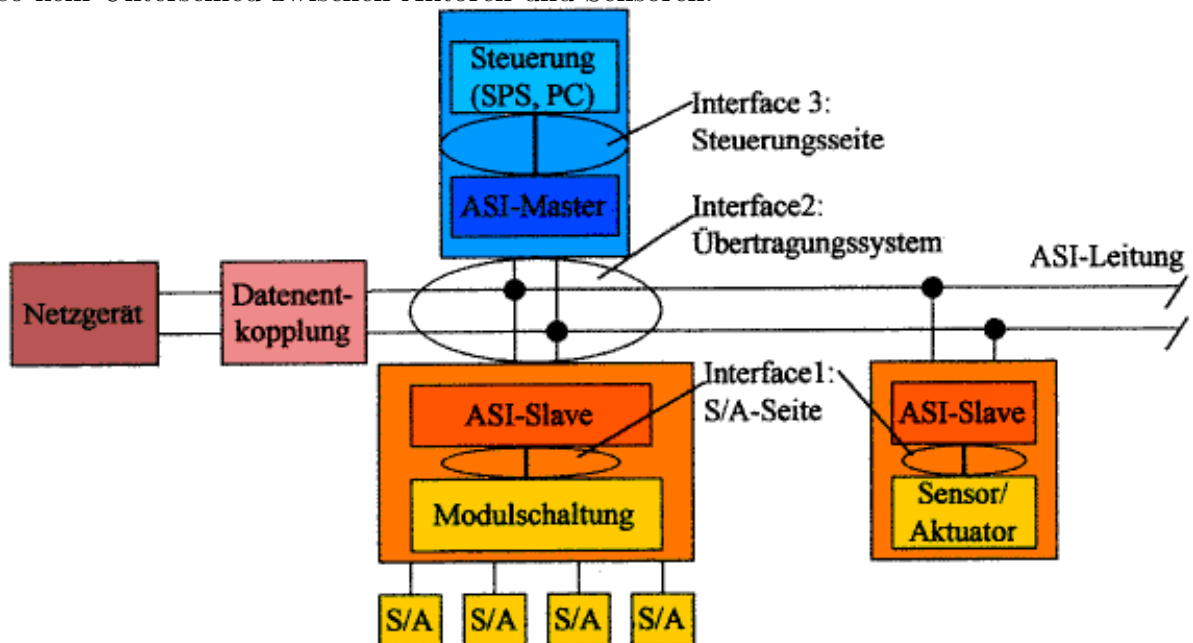
Die Motoren zum Bewegen in den drei Raumrichtungen werden mit jeweils 2 Bit angesteuert. Im Gegensatz zu den Sensoren werden die Aktoren logisch positiv angesteuert, d. h. ein Motor wird angeschaltet, wenn dessen entsprechendes Bit auf 1 gesetzt ist. Für die X-Richtung existiert zusätzlich noch ein weiteres Bit, um den Turm schneller zu bewegen, da in X-Richtung besonders genau positioniert werden muss. Um ein Klötzchen vom Eingabeslot aufzunehmen oder im Ausgabeslot abzulegen, können beide Bänder separat in Gang gesetzt werden. Die Zuordnung der Adressen zu den Aktoren ist in Tabelle 2 dargestellt.

ASI Adresse	Slavebit 3	Slavebit 2	Slavebit 1	Slavebit 0
Slave 1	-	X schnell	X rechts	X links
Slave 2	Y oben	Y unten	Z vorne	Z hinten
Slave 3	E/A rechts raus	E/A rechts rein	E/A links raus	E/A links rein

Tabelle 2: Zuordnung der ASI-Slaveadressen zu den Aktoren

4.2 Das Aktor-Sensor-Interface (ASI) EN 50295 und IEC 62026-2

Seitdem in der Mess- und Automatisierungstechnik Microcontroller im weiter vermehrt auftreten ist man auf der Suche nach möglichst einfachen (preislich und technologisch) Möglichkeiten um viele Sensoren (für die Messwerterfassung) und Aktoren (zu Ausführen von Aktionen) an einen Microcontroller oder auch einen PC anzuschließen. Dies führte auf die Entwicklung des ASI. Hier können ab Version 2.11 an einen Bus unter fest definierten Adressen bis zu 62 Aktoren/Sensoren angesprochen werden. Der Bus selbst besteht aus einem zweiadrigen Kabel über das sowohl die gesamte Kommunikation als auch die Stromversorgung erfolgt. Gesteuert wird das System durch den ASI Master. Dieser spricht zyklisch jeden am Bus angeschlossenen Slave an, sendet ein Kommando mit einer definierten Einheitslänge und erwartet von jedem Slave eine Antwort in einer ebenfalls definierten Standardlänge. Bustechnisch gesehen besteht für den ASI Master also kein Unterschied zwischen Aktoren und Sensoren.



4.3 Die Steuerung

Es soll eine kleine Regalverwaltung realisiert werden, über die dann ein Modell eines Hochregallagers gesteuert werden soll. Die Aufgabe der Regalverwaltung soll darin bestehen, neue Klötzchen einzulagern und bereits Eingelagerte wieder auszugeben. Entwickeln Sie ein Commandline-Interface mit folgenden Befehlen:

<code>getspace</code>	Zeigt die aktuelle Belegung des Hochregallagers an.
<code>vsetspace [posx] [posy]</code>	Definiert einen Platz als schon belegt.
<code>clearspace [posx] [posy]</code>	Definiert einen Platz als frei.
<code>insert [posx] [posy]</code>	Holt ein Klötzchen vom Eingabe-Slot und legt es an der Position (posx, posy) im Hochregallager ab.
<code>remove [posx] [posy]</code>	Holt ein Klötzchen von der Position (posx, posy) und legt es im Ausgabe-Slot ab.

Das Programm soll dabei zuerst prüfen, ob der Befehl überhaupt ausführbar ist. Beim Einlagern eines Klötzchens muss zum Beispiel überprüft werden, ob die gewünschte Lagerposition noch frei ist. Falls ein Befehl nicht ausgeführt werden kann, soll eine Warnung ausgegeben werden. Implementieren Sie geeignete Unterfunktionen, die

- den Schlitten bewegen
- überprüfen, an welcher Position sich der Schlitten befindet und
- testen, ob die Lichtschranken an den Ein-/Ausgabe Slots unterbrochen sind.

Entwickeln Sie eine geeignete Konsolen-Darstellung der Belegung des Hochregallagers.

Entwickeln Sie geeignete Synchronisationsmittel für die Bewegungen im Hochregallager und die Ausgabe über die Konsole.

4.4 Zusatz: Automatisierung

Diese Teilaufgabe ist optional.

Fügen Sie eine Timer-Routine hinzu, die den Eingabe-Slot des Hochregallagers alle zwei Sekunden überprüft und gegebenenfalls ein neues Klötzchen automatisch an eine freie Stelle im Hochregallager platziert. Die ausgewählte Position soll ausgegeben werden.