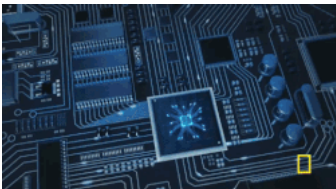


Einführung

course on LiaScript

| Parameter | Kursinformationen |
|------------------|---|
| Veranstaltung: | Digitale Systeme / Eingebettete Systeme |
| Semester | Wintersemester 2022/23 |
| Hochschule: | Technische Universität Freiberg |
| Inhalte: | Motivation der Vorlesung "Eingebettete Systeme" und Beschreibung der Organisation der Veranstaltung |
| Link auf GitHub: | https://github.com/TUBAF-lfi-LiaScript/VL_Softwareentwicklung/blob/master/00_Einfuehrung.md |
| Autoren | Sebastian Zug, André Dietrich & fjangfaragesh, FnHm, gjaeger, ShyFlyGuy, Lalelele |



Zielstellung aus dem Modulhandbuch

Qualifikationsziele /Kompetenzen:

Mit der erfolgreichen Teilnahme an der Veranstaltung sollen die Studierenden in der Lage sein:

- die Teilkomponenten eines Rechners ausgehend von der Schaltnetzen/-werken zu beschreiben und ausschnittshafte Teilelemente selbstständig entwerfen zu können,
- die Integration der Elemente und die Abläufe bei der Programmabarbeitung in verschiedenen Modellrechnern zu beherrschen und die Vor- und Nachteile verschiedener Konfigurationen bewerten zu können,
- die konkrete Realisierung von eingebetteten Systemen in entsprechenden Anwendungen aus den Schaltplänen zu erfassen und die softwareseitige Umsetzungen daraus abzuleiten
- einfache, eingebettete Systeme zu entwerfen und zu realisieren

Inhalte

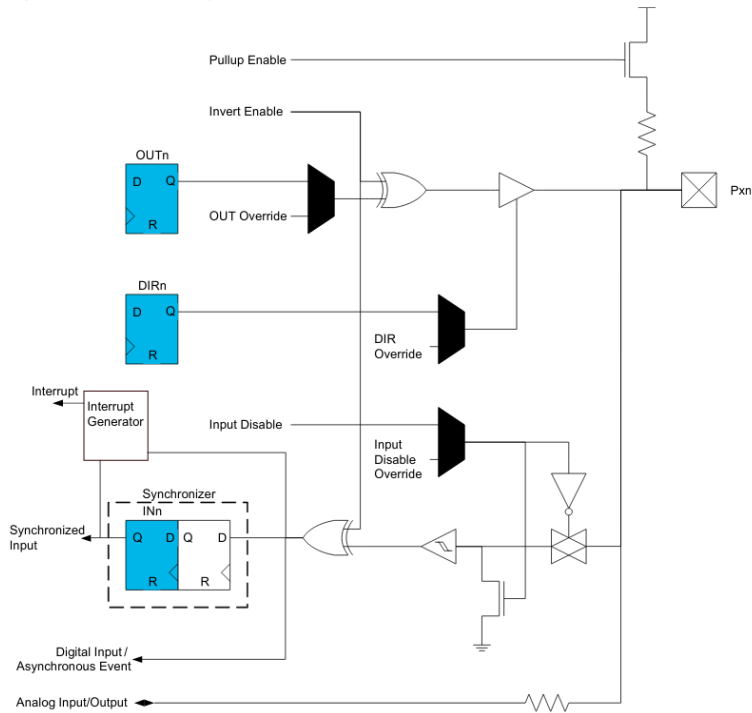
Grundlegende Prinzipien der Modellierung digitaler Systeme: Boolesche Algebren und Funktionen, kombinatorische und sequentielle Schaltungen, Herleitung eines Modellrechners und Abbildung von dessen Funktionsweise, Einführung in die Entwicklung eingebetteter Systeme(Sensoren, Aktoren, elektrische Peripherie, Programmierkonzepte), Anwendungsfelder

Etwas weniger abstrakt bitte!

Hardware

Was sind die Bausteine des Rechners

16.2.1 Block Diagram
Figure 16-1. PORT Block Diagram

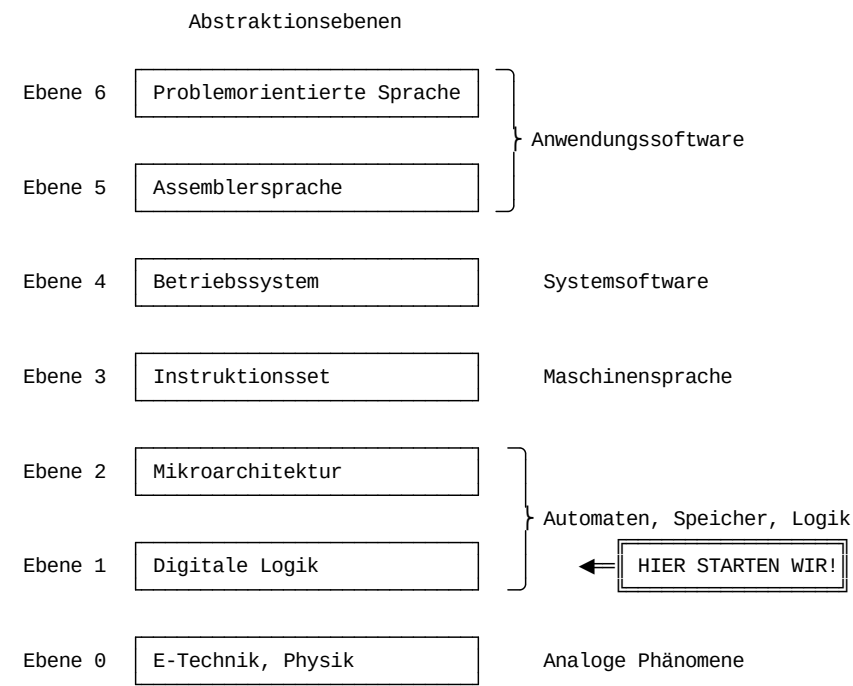


Darstellung der Input/Output Beschaltung eines Microcontrollers

Auf welche konkreten Systeme schauen wir dabei insbesondere?

- ATmega4808/4809
- ATmega32

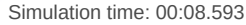
Wie kann ich die Perspektiven systematisieren



Software

Und wie wirkt sich das auf die Software aus?

Nehmen wir an, Sie realisieren ein Arduino Beispielprogramm wie dieses:

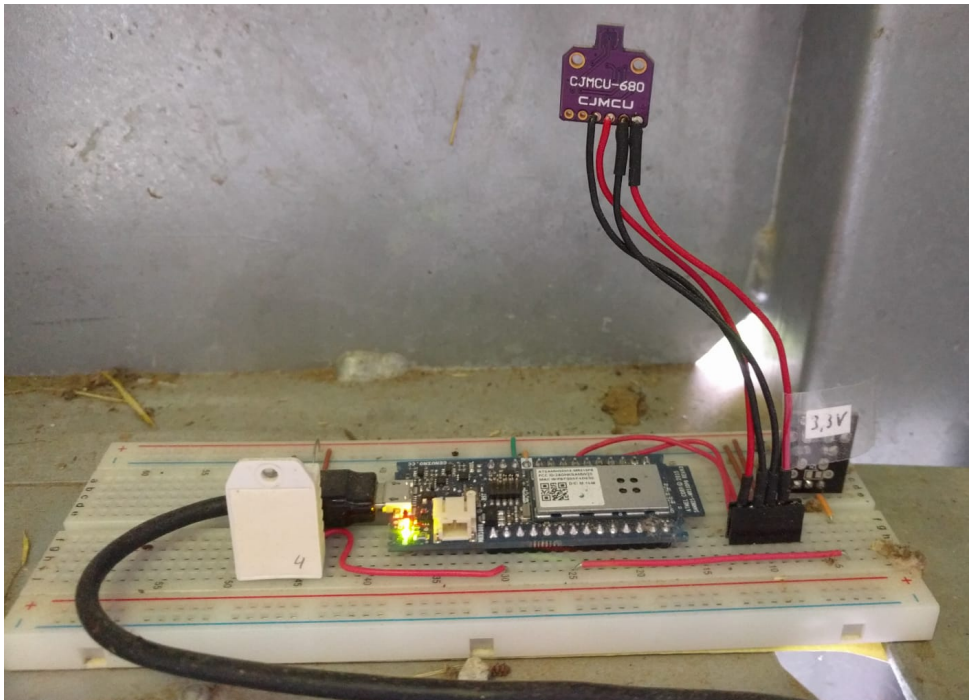


```
1000000000C9472000C947E000C947E000C947E0084
100010000C947E000C947E000C947E000C947E0068
100020000C947E000C947E000C947E000C947E0058
.
1000A0000C947E000C947E000C947E000C947E00D8
1000B0000C947E000C947E000C947E000C947E00C8
1000C0000C947E000C947E000C947E000C947E00B8
1000D0000C947E000C947E000C947E000C947E00A8
1000E0000C947E0011241FBCEFFED1E2DEBFCDBF46
1000F00000E00CBF0E9480000C9483000C94000070
0A010000279A2F98FECFF894FFEC 4
```

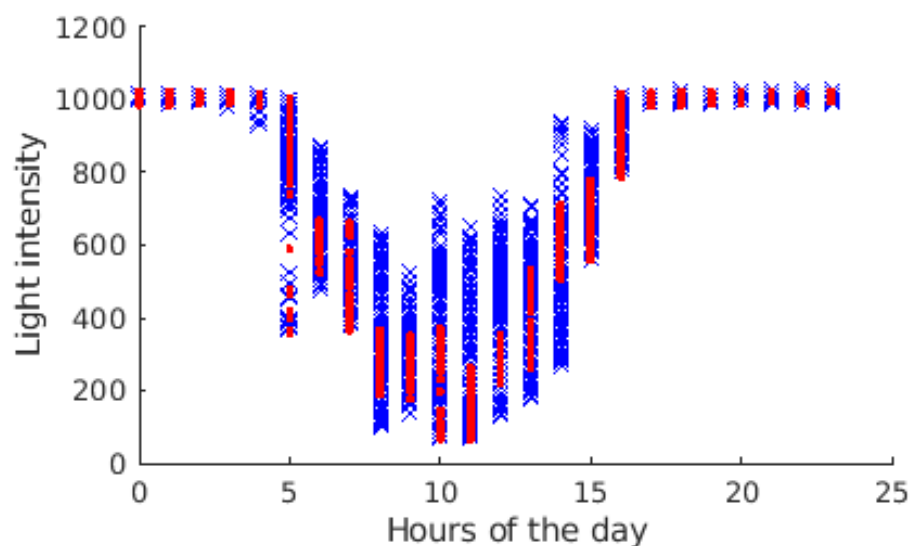
```
:100000000C9472000C947E000C947E000C947E0084
:100010000C947E000C947E000C947E000C947E0068
:100020000C947E000C947E000C947E000C947E0058
...
:1000A0000C947E000C947E000C947E000C947E00D8
:1000B0000C947E000C947E000C947E000C947E00C8
:1000C0000C947E000C947E000C947E000C947E00B8
:1000D0000C947E000C947E000C947E000C947E00A8
:1000E0000C947E0011241FBECFEFD1E2DEBFCDBF46
:1000F0000E00CBF0E9480000C9483000C94000070
:0A010000279A2F98FFCF894FFCF 4
```

Anwendungen

Und wie nützt mir das?



Wenn wir noch einen Schritt weitergehen, können wir die Daten auch an einen Server übergeben. Dieser übernimmt die Aufbereitung und Visualisierung. Hier lassen sich dann die eigentlichen "Untersuchungen" realisieren und zum Beispiel die Frage beantworten, ob die Sonne am Wochenende häufiger scheint.



Die roten Punkte stellen die Verteilung der Wochenendmessungen der vergangenen Woche dar, während die blauen Kreuze die Wochentage illustrieren. Dunkelheit wird durch einen Wert nahe 1023 ausgedrückt, während helle Messsituationen durch kleine Werte dargestellt werden.

Aber ich will Webentwickler werden ...

Antwort A:

Problem with the value at json.ok: [[{"row": 0, "column": 1, "text": " stray '@' in program", "type": "error"}, {"row": 0, "column": 7, "text": " expected constructor, destructor, or type conversion before '(' token", "type": "error"}]] Expecting a BOOL

Das Studium vermittelt ein breitgefächertes Weltbild und keine eng zugeschnittene Sicht.

Antwort B: Die Fähigkeit in Algorithmen zu denken ist eine Grundlage wissenschaftlichen Arbeitens.

Antwort C: Am Ende steht Ihnen das Rüstzeug zur Verfügung kleine eingebettete C-Projekte selbst anzugehen.

Organisation

| Name | Email |
|-------------------------|---|
| Prof. Dr. Sebastian Zug | sebastian.zug@informatik.tu-freiberg.de |
| Bastian Zötzl | Bastian-Gabriel.Zoetzl@student.tu-freiberg.de |

Bitte melden Sie sich im OPAL unter [Digitale Systeme](#) für die Veranstaltung an. Dies ist im Kontext der Pandemiesituation Teil des Hygienekonzepts der Hochschule.

Zeitplan

Die Veranstaltung wird sowohl für die Vorlesung als auch die Übung in Präsenz durchgeführt.

| Veranstaltungen | Tag | Zeitslot | Ort | Bemerkung |
|-----------------|------------|---------------|----------|---------------|
| Vorlesung I | Mittwoch | 16.00 - 17.30 | SPQ-1301 | wöchentlich |
| Vorlesung II | Donnerstag | 16.00 - 17.30 | SPQ-1301 | gerade Wochen |

Die zugehörigen Übungen starten im Dezember und werden dann wöchentlich durchgeführt.

- Übung 1 (ROB), Mittwochs, 14.00 - 15.30 Uhr, KKB-2097
- Übung 2 (BM, Mm, BAI), Freitags, 14.00 - 15.30 Uhr, KKB-2097

Wir gehen gegenwärtig noch davon aus, dass die Übungen auch in Präsenz stattfinden. Dort haben Sie dann insbesondere ab Januar Gelegenheit anhand spezifischer Mikrocontrollerschaltungen Ihre Fähigkeiten zu vertiefen.

Prüfungsmodalitäten

Credit-Points: 6

Arbeitsaufwand: Der Zeitaufwand beträgt 180h und setzt sich zusammen aus **60h Präsenzzeit** und **120h Selbststudium**. Letzteres umfasst die Vor- und Nachbereitung der Lehrveranstaltung, die eigenständige Lösung von Übungsaufgaben sowie die Prüfungsvorbereitung.

Prüfungsform: Die Veranstaltung wird mit einer schriftlichen Prüfung abgeschlossen. Diese wird als Open Book Klausur entworfen.

Literaturempfehlungen

1. Umfassende Lehrbücher
 - David A. Patterson, John L. Hennessy: Computer Organization & Design
 - B. Becker, R. Drechsler, P. Molitor: Technische Informatik - Eine Einführung, Pearson Studium, 2005
 - Hoffmann, D. W.: Grundlagen der technischen Informatik, Hanser Verlag
2. Videos
 - Youtube – „How a CPU Works“ [Link](#)

Bei den jeweiligen Vorlesungsinhalten werden zusätzliche Materialien angegeben.

Schreiben Sie an den Materialien mit!

Die Lehrmaterialien finden Sie unter GitHub, einer Webseite für das Versionsmanagement und die Projektverwaltung.

https://github.com/TUBAF-lfl-LiaScript/VL_EingebetteteSysteme/blob/dev/01_HistorischerUeberblick.md

Die Unterlagen selbst sind in der Auszeichnungssprache LiaScript verfasst und öffentlich verfügbar.

Markdown ist eine Auszeichnungssprache für die Gliederung und Formatierung von Texten und anderen Daten. Analog zu HTML oder LaTeX werden die Eigenschaften und Organisation von Textelementen (Zeichen, Wörtern, Absätzen) beschrieben. Dazu werden entsprechende "Schlüsselworte", die sogenannten Tags, verwendet.

Markdown wurde von John Gruber und Aaron Swartz mit dem Ziel entworfen, die Komplexität der Darstellung so weit zu reduzieren, dass schon der Code sehr einfach lesbar ist. Als Auszeichnungselemente werden entsprechend möglichst kompakte Darstellungen genutzt.

| HelloWorld.md | |
|---|--|
| <pre># Überschrift __eine__ Betonung __in kursiver Umgebung__ + Punkt 1 + Punkt 2 Und noch eine Zeile mit einer mathematischen Notation \$a=\cos(b)\$!</pre> | |

Überschrift

eine Betonung in kursiver Umgebung

- Punkt 1
- Punkt 2

Und noch eine Zeile mit einer mathematischen Notation $a = \cos(b)$!

Eine gute Einführung zu Markdown finden Sie zum Beispiel unter:

- [MarkdownGuide](#)
- [GitHubMarkdownIntro](#)

Mit einem entsprechenden Editor und einigen Paketen macht das Ganze dann auch Spaß.

- Wichtigstes Element ist ein Previewer, der es Ihnen erlaubt "online" die Korrektheit der Eingaben zu prüfen
- Tools zur Unterstützung komplexerer Eingaben wie zum Beispiel der Tabellen (zum Beispiel für Atom mit [markdown-table-editor](#))
- Visualisierungsmethoden, die schon bei der Eingabe unterstützen
- Rechtschreibprüfung (!)

Allerdings vermisst Markdown trotz seiner Einfachheit einige Features, die für die Anwendbarkeit in der (Informatik-)Lehre sprechen:

- Ausführbarer Code
- Möglichkeiten zur Visualisierung
- Quizze, Tests und Aufgaben
- Spezifische Tools für die Modellierung, Simulationen etc.

ArduinoSimulator.ino

```
1 void setup() {  
2   Serial.println("Hello stuff.");  
3 }  
4  
5 void thing(char i) {  
6   switch(i) {  
7     case 0: Serial.println("a pear"); break;  
8     case 1: Serial.println("an apple"); break;  
9     case 2: Serial.println("an elephant"); break;  
10    case 3: Serial.println("an arduino"); break;  
11  }  
12 }  
13  
14 void loop() {  
15   Serial.print("here's ");  
16   thing(random(4));  
17 }
```

Sketch uses 2062 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 252 bytes (12%) of dynamic memory, leaving 1796 bytes for local variables. Maximum is 2048 bytes.

Eine Reihe von Einführungsvideos findet sich unter [Youtube](#). Die Dokumentation von LiaScript ist hier [verlinkt](#)

Trotz Simulation und Virtuellem ...

... braucht es aber auch immer etwas zum anfassen.

Blick hinter eine Arduino-Anwendung

Wie können Sie zum Gelingen der Veranstaltung beitragen?

- Stellen Sie Fragen, seien Sie kommunikativ!

Hinweis auf das OPAL Forum!

- Organisieren Sie sich in Arbeitsgruppen!

Hinweis auf Repl.it

Hinweis auf ThinkerCAD

- Bringen Sie sich mit Implementierungen in die Veranstaltung ein.

Und wenn Sie dann immer noch programmieren wollen ...

Dann wartet das **racetech** Team auf Sie ... autonomes Fahren im Formular Student Kontext.

Hausaufgabe

- Legen Sie sich einen GitHub Account an ... und seien Sie der Erste, der einen Typo in den Unterlagen findet und diesen als Contributor korrigiert 😊