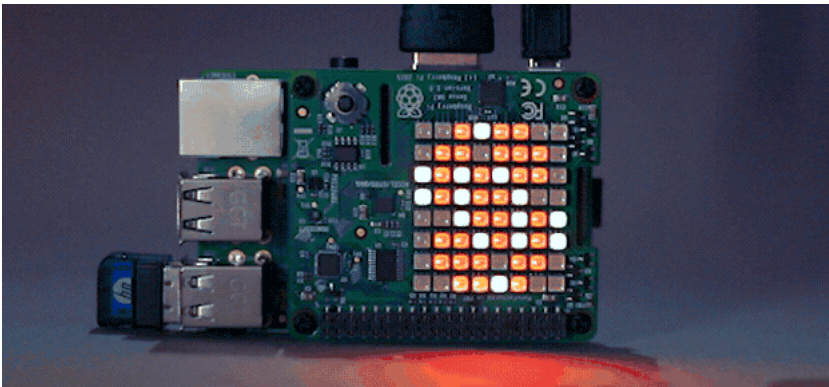


# Einführung und Motivation

Parameter	Kursinformationen
Veranstaltung:	Vorlesung Digitale Systeme
Semester	Sommersemester 2022
Hochschule:	Technische Universität Freiberg
Inhalte:	Motiviation und Organisation der Veranstaltung
Link auf den GitHub:	<a href="https://github.com/TUBAF-lfi-LiaScript/VL_DigitaleSysteme/blob/main/lectures/00_Einfuehrung.md">https://github.com/TUBAF-lfi-LiaScript/VL_DigitaleSysteme/blob/main/lectures/00_Einfuehrung.md</a>
Autoren	Sebastian Zug, Karl Fessel & André Dietrich



avrlibc.cpp

```
// preprocessor definition
#define F_CPU 16000000UL
#define ANSWER_TO_LIFE 42

#include <avr/io.h>
#include <util/delay.h>

int main (void) {
    Serial.begin(9600);

    volatile byte a;

    asm ("ldi %0, %1\n\t"
        : "=r" (a)
        : "M" (ANSWER_TO_LIFE));

    Serial.print("Antwort auf die Frage, warum ich an dieser Vorlesung teilnehme: ")
    );
    Serial.println(a);

    while(1) {
        _delay_ms(1000);
    }
    return 0;
}
```

## Zielstellung

Was steht im Modulhandbuch über diesen Kurs?

Qualifikationsziele /Kompetenzen:


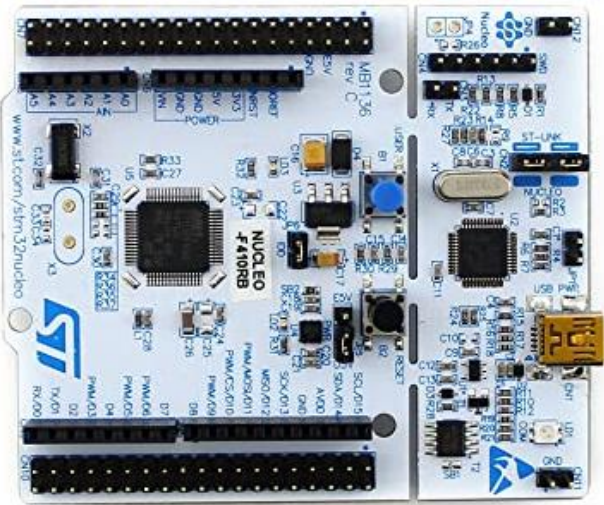
- die Komponenten realer eingebetteter Controller-Architekturen(8Bit -32Bit) zu beschreiben und analysieren zu können
- Controller im Hinblick auf bestimmte Anforderungsprofile zu beurteilen
- Elemente eingebetteter Anwendungen (insbesondere Sensoren)in ihrer Funktion und Eignung auszuwählen und in Software und Hardware in eine Anwendung zu integrieren Methoden des Softwareentwurfes und verschiedenen Tool-Chains für die Implementierung eingebetteter Systeme anwenden zu können
- Codefragmente im Hinblick auf die Qualität und mögliche Fehler zu analysieren

#### Inhalte

- Gegenüberstellung verschieden Architekturen etablierter Controller
- Integration von Controllern in eingebetteten Anwendungen
- Erweiterung als Sensor-Aktor-Systeme,
- Parameter von Sensorssystemen,
- Betriebssystemkonzepte für eingebettete Controller

Und was heißt das nun konkret? Worum geht es?

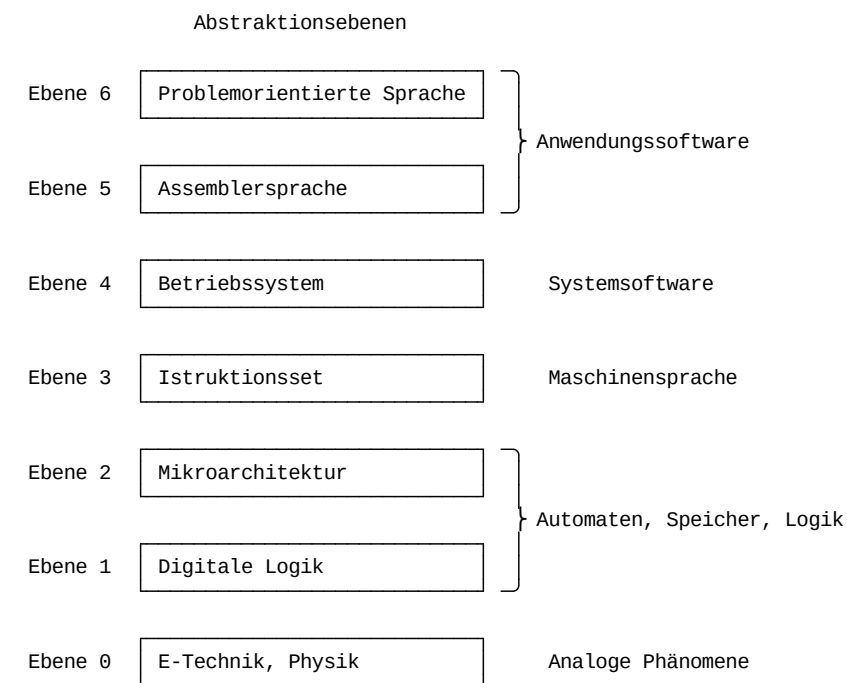
#### Hardware-Plattformen

Arduino Uno Board	Nucleo 64
	
Microchip ATmega 328p	STM32F401
8-Bit AVR Familie	32-Bit Cortex M4 Prozessor
C, Assembler	C, C++
avrlibc, FreeRTOS	CMSIS, mbedOS, FreeRTOS
10 Bit Analog-Digital-Wandler, 16 Bit Timer,	10 timers, 16- and 32-bit (84 MHz), 12-bit ADC

#### Inhalte der Vorlesung

VL	Tag	Inhalt der Vorlesung
0	06.04.	Einführung und Motivation
1	13.04.	ATmega Architektur
2	20.04.	ATmega Komponenten
3	27.04.	Performancebeschränkungen der Architektur
4	04.04.	Analog Digitalwandler
5	12.05.	<i>Ausfall wegen Konferenzreise</i>
6	18.05.	Kommunikationsprotokolle
7	25.05.	Echtzeitanwendungen
8	01.06.	Scheduling Algorithmen
9	08.06.	FreeRTOS
10	15.06.	megaAVR 0 Series - xMega Controller
11	22.06.	megaAVR 0 Series - xMega Controller Abgrenzung
12	29.06.	ARM Controllern Architektur
13	06.07.	Komponenten des STM32F4
14	13.07.	mbedOS

Dabei werden unterschiedliche Ebenen der Abstraktion der Abläufe im Rechner adressiert.



## Organisation

Name	Email
Prof. Dr. Sebastian Zug	sebastian.zug@informatik.tu-freiberg.de
Georg Jäger	georg.jaeger@informatik.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 Universität.

## Zeitplan

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

Veranstaltungen	Tag	Zeitslot	Bemerkung
Vorlesung	Mittwoch	9:15-10:45	
Praktikum	Donnerstag	7:30-9:00	ungerade Woche
Übungen	Donnerstag	7:30-9:00	gerade Woche

## 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 mündlichen Prüfung abgeschlossen. Dabei starten wir zunächst mit einem von Ihnen gewählten Codebeispiel und gehen dann auf die Inhalte der Vorlesungen und Übungen ein.

## Schreiben Sie an den Materialien mit!

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

[https://github.com/TUBAF-lfi-LiaScript/VL\\_DigitaleSysteme/tree/main/lectures](https://github.com/TUBAF-lfi-LiaScript/VL_DigitaleSysteme/tree/main/lectures)

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

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 Simulation etc.



13 Simulation time: 00:24.281

## avrlibc.cpp

```
1  #ifndef F_CPU
2  #define F_CPU 16000000UL // 16 MHz clock speed
3  #endif
4
5  //16.000.000 Hz / 1024 = 15.625
6
7  int main(void)
8  {
9      Serial.begin(9600);
10     DDRB |= (1 << PB5); // Ausgabe LED festlegen
11     // Timer 1 Konfiguration
12     //      keine Pins verbunden
13     TCCR1A = 0;
14     TCCR1B = 0;
15     // Timerwert
16     TCNT1 = 0;
17     TCCR1B |= (1 << CS12) | (1 << CS10); // 1024 als Prescale-Wert
18
19     while (1) //infinite loop
20     {
21         if (TCNT1 > 15625) {
22             TCNT1 = 0;
23             PINB = (1 << PB5); // LED ein und aus
24         }
25     }
26 }
```

Sketch uses 1150 bytes (3%) of program storage space. Maximum is 32256 bytes.  
Global variables use 175 bytes (8%) of dynamic memory, leaving 1873 bytes for local variables. Maximum is 2048 bytes.

**Frage:** Welche Komponente des Controllers wird hier genutzt? An welcher Stelle hätten Sie alternative Ideen?

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.

Wir stellen Ihnen die entsprechenden Controller beider Plattformen zur Verfügung.

## Wie können Sie zum Gelingen der Veranstaltung beitragen?

- Stellen Sie Fragen, seien Sie kommunikativ!

Hinweis auf OPAL Forum!

- Organisieren Sie sich in Arbeitsgruppen!

Bearbeiten Sie Aufgaben gemeinsam online.

- Bringen Sie sich mit Implementierungen in die Veranstaltung ein.

## Und jetzt sind Sie an der Reihe

Mit welchen Systemen haben Sie bereits gearbeitet und welche Projekte haben Sie damit verwirklicht?

## Hausaufgabe

- Legen Sie sich, sofern das noch nicht geschehen ist, einen GitHub Account an ... und seien Sie der erste, der einen Fehler, eine unglückliche Formulierung usw. in den Unterlagen findet und diesen als *Contributer* korrigiert 😊
- Überlegen Sie sich, zu welches Thema Sie im Praktikum vorstellen wollen:
  - DSP Implementierung im STM32F4
  - STM32 Cube.AI
  - mbedOS vs FreeRTOS
  - LCD Programmierung mit dem STM
  - CAN Bus - Theorie und Anwendung
  - Micropython im Kontext von Mikrocontrollern

Die Präsentationen sollten 30 Minuten einnehmen und mit einer praktischen Demonstration kombiniert werden.

- Setzen Sie sich mit [platformio.org](https://platformio.org) auseinander und testen Sie die Features zum Debugging.