

Liebe Studentinnen und Studenten,

Corona lässt grüßen. Es sind Maßnahmen getroffen worden, die von uns viel Flexibilität und Improvisationstalent verlangen. Wir beginnen mit der Vorlesung pünktlich zum 18.03.2021 1. bis 3. Stunde. Da der Präsenz Unterricht abgesagt wurde, wird die Vorlesung incl. Praktikum mittels ZOOM-Videokonferenz und Moodle abgehalten. Die Termine sind, wie im offiziellen Stundenplan bereits festgelegt, fix.

Der **Moodlekurs** heißt: Rechnerarchitektur AI2
Der Zugangscode lautet: RA_AI2_S_2021

Der Link zum **ZOOM-Meeting „Rechnerarchitektur mit Praktikum“** lautet:
<https://hs-offenburg.zoom.us/j/332134384?pwd=dG1LL0k2WWE1cVZiUzlwNEFCdVFQQT09>
Meeting-ID: 332 134 384
Passwort: 392361

Achten Sie unbedingt darauf, dass Sie sich mit ihrer Hochschuladresse anmelden. Sonst sind sie nur Gast, mit eingeschränkten Möglichkeiten.

Was kommt auf mich zu?

Für dieses Praktikum hat sich gezeigt, dass ein starkes Engagement seitens der Studenten notwendig ist um das Ziel überhaupt zu erreichen. Außerdem liegt der Abgabetermin weit vor den Klausurwochen damit hier kein Problem im Zeitmanagement jedes Einzelnen entsteht. Die Erfahrung zeigt: Wer nicht von Anfang an hochkonzentriert an dem Projekt arbeitet, hat kaum eine Chance das Ziel zu erreichen.

Die Vorgehensweise wird im Prinzip wie in einer "normalen" Vorlesung sein. In der Zeit von 8.00 Uhr bis 9.30 Uhr sind Inhalte allgemeiner Art auf dem Plan. Hierzu zählen die theoretischen Grundlagen der Rechnerarchitektur. Anfangs ist aber der Schwerpunkt auf den PIC-Microcontroller zu richten. Dazu werden die Dinge angesprochen, die Sie unbedingt Wissen müssen, um den Simulator (Inhalt des Praktikums) schreiben zu können. Deshalb sind bei den ersten beiden Terminen auch die 2. und 3. Stunde für **ALLE** Pflichttermine.

Der übliche Terminplan kann in Moodle eingesehen werden.

Für die Vorlesung per ZOOM sind natürlich bestimmte Abläufe vorgegeben.

- Ich werde nicht alles, was ich in der Vorlesung erzähle, auf das White Board niederschreiben.
- Sie arbeiten hauptsächlich mit den Unterlagen die ich ihnen zur Verfügung stelle.

- Wir arbeiten dann diese Unterlagen Absatz für Absatz oder Seite für Seite durch.
- Nachdem Sie den jeweiligen Absatz durchgearbeitet haben, können sie im Fragen stellen. Da ich nicht sehe, wann und wie Sie fragend die Stirn runzeln, müssen sie aktiv diese Fragen auch stellen. (Das ist einer der großen Unterschiede zwischen ZOOM und Anwesenheit!)
- Es kann sein, dass ich, ähnlich wie im Unterricht, die eine oder andere Frage nicht sofort beantworte, sondern u.U. eine zur Antwort führende Gegenfrage stelle. Auch ein Hinweis, dass die Frage in naher Zukunft beantwortet wird, werde ich mir ab und zu erlauben.
- Es macht Sinn, wie im normalen Vorlesungsablauf, einige Vorarbeiten zu machen. Dazu arbeiten Sie am Besten das nächste Kapitel in den Unterlagen durch. Damit entsteht weniger Leerlauf und Langeweile.

Wie Online-Vorlesungen ablaufen, haben wir in den letzten zwei Semester kennen gelernt. Es wird nicht alles optimal laufen, wenn alle am selben Strang ziehen, wird es aber dennoch zum Erfolg führen.

Je größer die Anstrengung - desto größer das Erfolgserlebnis (= Lernerfolg)

Was ist zu tun? Wie gehen wir vor?

Laden Sie sich folgende Unterlagen auf den Rechner. Drucken Sie davon das Wichtigste auf Papier aus.

Ich beginne mit der Vorstellung des PIC 16F84. Das ist der Mikrocontroller, dessen Funktionalität im Praktikum nachgebildet werden muss. Diese Nachbildung besteht nicht auf der Hardwareebene sondern eine Stufe höher. Die einzelnen Assemblerbefehle sollen in ihrer Funktion per Java beschrieben werden. Somit muss für den Additionsbefehl ADDLW nicht ein 8-bittiger Volladdierer auf Binärebene realisiert werden, sondern es kommt der +-Operator von Java zum Einsatz.

Ich stelle ihnen die Testdatei Nr. 1 vor. Sie zeigt ein sogenanntes LST-File, das alle Informationen, die ein Simulator braucht, beinhaltet. Diese LST-Datei muss Zeile für Zeile eingelesen und in die entsprechenden Bestandteile aufgeteilt werden. Der extrahierte Binärcode muss im Programmspeicher abgelegt werden. (=> Kapitel 3).

Im *Themenblatt_Simulator* ist beschrieben, wie der PIC-Mikrocontroller intern aufgebaut ist (Kapitel 4ff)

Ein wichtiger Bestandteil eines Mikrocontrollers ist der Befehlsdekoder und diesen gilt es zu implementieren (Kapitel 13).

Was tun, wenn man noch nie etwas mit Assembler zu tun hatte?

Für alle diejenigen, für die die obige Aussage zutrifft, habe ich das *Themenblatt_PIC_programmieren* geschrieben.

Vorneweg!!! Assemblerprogrammierung ist nicht schwerer als eine Hochsprache. In meinen Augen sogar etwas einfacher, da der Befehlssatz, speziell bei der PIC-Familie mit seinen 32 Befehlen, sehr überschaubar ist. Im Gegensatz zu einer Hochsprache muss man allerdings die Hardware relativ gut kennen, weil es keine Betriebssystemaufrufe gibt, die einem diese Arbeit abnehmen. Wenn ich z.B. einen Ausgang von Low auf High setzen will, muss ich in der Hochsprache i.d.R. nur einen API-Aufruf mit den entsprechenden Übergabeparameter zu machen. Bei Assemblerprogrammen muss ich dagegen die Hardwareadresse des sogenannten I/O-Registers kennen um den Ausgang auf High zu setzen.

Ein weiterer Nachteil von Assembler ist, dass man sich um alles, aber auch wirklich alles, selber kümmern muss. Das fängt mit den Variablen an. Es gibt kein Integer- oder Float, sondern nur ein Register mit 0 und 1 darin. Das es ein Integer oder Float ist, das bestimmt die Hochsprache, die wiederum entsprechende Softwareroutinen besitzt um diese Null und Einsen in den Registern entsprechend zu interpretieren. Wer C kennt, weiß dass ich auf eine Floatvariable auch mit einem Stringbefehl zugreifen kann. Und genauso ist das in Assembler möglich. Jede Freiheit, aber auch volle Verantwortung.

Was tun, wenn meine Programmierkenntnisse nicht ausreichend sind?

Dafür ist dieses Projekt da, um Sie in Java besser zu machen. Aber das ist dann vor allem von Ihnen selbst abhängig, ob es ein Erfolg wird oder nicht. Die langjährige Erfahrung zeigt, dass selbst jene von ihnen, die im letzten Semester zum ersten Mal Java näher in Kontakt gekommen sind auch in der Lage sind dieses Projekt erfolgreich zu stemmen. Die Hauptschwierigkeit liegt in 90 % der Fälle bei der GUI-Programmierung. Damit hier keine unüberwindlichen Probleme auftauchen oder auch

um die ersten Schritte schneller und einfacher zu machen, habe ich eine, von ihrem zu erstellenden Programm, unabhängige GUI entwickelt. Sie haben so die Möglichkeit, von vornherein auf eine eigene GUI zu verzichten. Sobald Sie sich mit der Vorgehensweise vertraut gemacht haben, können Sie noch immer ihre eigene GUI realisieren. Dass dies Sinn macht, ist aus dem Bewertungsschema ersichtlich. Die eigene GUI bringt natürlich auch Punkte ein. Will man ohne eigene GUI auskommen, muss man mehr der PIC-Funktionen im Simulator realisieren um auf die notwendige Punktzahl zu kommen.

Wie diese "fremde" GUI funktioniert und angesteuert wird, können Sie im Dokument *GUI_PicSim.pdf* nachlesen. Diese befindet sich zusammen mit dem eigentlichen Programm *GUI_PicSim.exe* und einen "einfachen" Backendprogramm (*Backend.exe*) in der ZIP-Datei *GUI_PicSim.zip*.