

Name	Matrikel	Anmerkungen
Datum	Raster (z.B. Mo-2x)	Testat/Datum

Legende: V: Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Praktikum 1

Lernziele: GNU-C-Compiler, C-Programmierung, einfache Ein- und Ausgaben. Simulation eines Mikrocontroller-Programms auf PC (pandemiebedingt). Formatierte Ausgabe von Integer in dezimal, binär und hexadezimal. Eingabe von Terminal und Verarbeitung im Programm, ASCII. Einfache Menusteuerung. Integerformate, 8-, 16-, 32-bit, signed und unsigned. Macros in C.

Für die Bearbeitung der Aufgaben ist ein Termin angesetzt.

1. Installieren Sie einen GNU-C-Compiler auf Ihrem PC, beispielsweise *CodeBlocks* [1][2].
 - a. Testen Sie die Entwicklungsumgebung durch Erstellen eines Konsolenprojekts (*Console application*).
 - b. Achtung: bei *CodeBlocks* muss die Konsole explizit von Ihnen geschlossen werden, bevor neu kompiliert und ausgeführt werden kann!!
 - c. Nehmen Sie einige einfache Änderungen und Erweiterungen an Ihrem Programm vor und testen Sie.
2. Betrachten Sie das beigelegte Projekt und erweitern Sie
 - a. Was machen die Precompiler-Statements `__DATE__`, `__TIME__`, `__LINE__`?
 - b. Geben Sie eine 8bit-Integer Zahl (`uint8_t`) als Dezimal- und Hexadezimal-Wert aus.
 - c. Wie kann der Wert binär ausgegeben werden? (eigener Code erforderlich)
 - d. Ist Ihre Binärdarstellung little- oder big-endian?
3. Characters lesen vom Terminal, d.h. auf dem PC von der Tastatur
 - a. Um das Verhalten der Funktion `getc` auf einem ARM-Prozessor zu simulieren, ist die Funktion `getch_nb()` (*nicht blockierend*) vorgeschlagen. (Ohne Eingabe wird der Wert '0' zurückgegeben oder bei Tastendruck der ASCII-Wert)
 - b. Testen Sie das Verhalten der Funktion.
 - c. Führen Sie bestimmte Aktionen aus bei einer Reihe von Eingaben, z.B. 0, 1, 2, 3.
4. Setzen und Löschen einzelner Bits.
 - a. Wie können mit C-Code einzelnen Bits gesetzt werden?
 - b. Wie können mit C-Code einzelnen Bits gelöscht werden?
 - c. Schreiben Sie zwei wiederverwendbare Funktionen `uint8_t sbit(bytevalue, pos)` und `uint8_t cbit(bytevalue, pos)`.
 - d. Testen Sie mit 2.c.
 - e. *Was ist ein Macro?*
Freiwillig: Schreiben Sie für beide Funktionen ein C-Macro `SBIT` und `CBIT`.
 - f. Freiwillig: Schreiben Sie ein Macro `MULT(a, b)` welches *a* und *b* multipliziert.
Testen Sie mit a=3, b=4 und a=2+1, b=4 !

5. Kommentieren Sie – gegebenenfalls nach dem Praktikum zu Hause – Ihren Code. Archivieren Sie Ihr Projekt zu Ihrem späteren Gebrauch.
6. Schreiben Sie ein kurzes Protokoll und fassen Sie Ihre Erkenntnisse zusammen und fügen Sie die jeweiligen Codeabschnitte hinzu. Laden Sie Ihren Code⁺⁾ als *.zip und Ihr Protokoll als *.pdf in Moodle hoch bis **maximal** 1 Woche nach dem Termin.

Bereiten Sie sich auf den Praktikumstermin 1 so vor, dass die Zeit zur Durchführung während des Termins sicher ausreicht. *(Lesen Sie bitte die Aufgabenstellung und Begleitmaterial vor dem Praktikumstermin.)*

Die Themen und Erkenntnisse aus diesem Praktikum werden im Lauf des Semesters weiter benötigt! Arbeiten und dokumentieren Sie Ihre Ergebnisse sorgfältig!

Die Teilaufgaben sind schriftlich zu dokumentieren. Laden Sie Ihr Protokoll wie in 6. beschrieben zu Termins 1 hoch.

Viel Spaß und Erfolg

*+) im *.zip bitte **nur** den Ordner mit *.c, *.h und gegebenenfalls Projektdatei.*

[1] <https://www.codeblocks.org/downloads/binaries/>

[2] unter Windows `codeblocks-20.03mingw-setup.exe`, ggf. als Administrator