Christian-Albrechts-Universität zu Kiel
Institut für Informatik

Lehrstuhl für Technische Informatik
Prof. Dr. Manfred Schimmler
M.Sc. Jan Christian Kässens

Rechnergestützter Entwurf digitaler Systeme

Sommersemester 2016

Serie 3

Ausgabe: 16. Mai 2016
Abgabe: 29. Mai 2016 via OLAT

Aufgabe 1 (60 Punkte)

Schreiben Sie für den Mikrocontroller auf dem Breadboard ein Programm in der Programmiersprache C. Richten Sie sich dazu eine entsprechende Entwicklungsumgebung ein. Unter Linux benötigen Sie dazu die folgende Software:

- AVR cross-compiler `gcc-avr`
- AVR C-Bibliothek `avr-libc`
- ISP-Programmiersoftware `avrdude`

Falls nicht standardmäßig in Ihrer Distribution enthalten, benötigen Sie außerdem noch die Softwarepakete `make` und `perl`.

Implementieren Sie nun das in der Übung vorgeführte Programm:

- Zu Beginn des Programms sind alle LEDs ausgeschaltet
- Bei einem Tastendruck wird eine LED-Farbe eingeschaltet, und zwar so lange wie die Taste gedrückt gehalten wird:
  - S201 und S204 schalten *rot*
  - S202 und S205 schalten *grün*
  - S203 und S206 schalten *blau*

Nutzen Sie dazu das Rahmenprojekt und das Makefile aus dem Materialordner im OLAT. Das Makefile unterstützt die folgenden Targets:

- `compile`, kompiliert die C-Datei, die in dem Makro `SOURCE` angegeben ist und erzeugt den ausführbaren Maschinencode
- `install`, schreibt die Fuse-Bits und das Programm in den Speicher des Mikrocontrollers

Führen Sie also nach dem Erstellen des Programms `make compile` aus und anschließend `make install`, um das Programm auf dem Mikrocontroller zu überspielen.

Hinweis: Ziehen Sie die Batterie von der Schaltung ab, bevor Sie den Programmieradapter anschließen und schließen Sie sie anschließend wieder an. Sie können den Programmieradapter auch während des normalen Betriebs angeschlossen lassen.

Machen Sie sich nun mit dem Programmieren des Mikrocontrollers vertraut und studieren Sie die Ausgaben des `make compile`-Befehls. Beobachten Sie die Größe Ihres Programms genau.

Sollte das Überspielen nicht funktionieren, versuchen Sie, den `make install`-Schritt als root auszuführen. Als dauerhafte Lösung können Sie dann in `/etc/udev/rules.d/` eine Datei `99-avr.rules` erstellen, die den Zugriff auf den Programmieradapter regelt:

```
SUBSYSTEM=="usb",ATTRS{idVendor}=="03eb",ATTRS{idProduct}=="2104",MODE="0666"
```

Wenn Sie nun den Adapter entfernen und wieder anschließen, sollten die Berechtigungen den letzten Schritt erlauben.

Aufgabe 2 (40 Punkte)

In dieser Aufgabe geht es darum, das Verständnis und die Anwendung der Peripherie zu vertiefen. Nutzen Sie ebenfalls den Programmrahmen und das Makefile aus dem OLAT und erstellen Sie ein Programm, welches mindestens zwei der folgenden Funktionen implementiert:

- Toggle-Buttons: LEDs können per Tastendruck ein- und ausgeschaltet werden. *Hinweis:* Interrupts können Ihren Code vereinfachen
- Timer: lassen Sie LEDs blinken
- PWM: machen Sie die LEDs per Tastendruck dimmbar
- EEPROM: speichern Sie den Zustand der LEDs im EEPROM, sodass dieser einen Stromausfall überdauert und wiederhergestellt wird.

Seien Sie kreativ! Denken Sie sich weitere Funktionen aus. Als Grenzen sind Ihnen lediglich die Größen von Flash und SRAM gesetzt.