

# Praktikum 6./7. Dezember 2010

Jörn Loviscach

Versionsstand: 6. Dezember 2010, 18:45



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

In diesem Praktikum soll ein Taschenrechner mit „innovativer“ Ein-Knopf-Bedienung programmiert werden.

Aufgabe zu Hause als Vorbereitung: Wie müsste der Code für die zweite Aufgabe aussehen? Schreiben Sie den auf oder programmieren Sie den ohne die Hardware!

Das C-Projekt basiert auf der kleinen Funktionsbibliothek, die aus `display02.c` und `display02.h` besteht. Die Funktionsnamen und -parameter sollten weitgehend selbsterklärend sein. Von besonderem Interesse sind:

```
void initialize(void);
void writeString(char s[]);
void writeNumber(unsigned int a);
int getIncrement(void);
unsigned int getMillisecondsSinceLastChange(void);
void waitMilliseconds(unsigned int duration);
```

Die Funktion `getIncrement` gibt dabei zurück, um wieviele Schritte der Encoder seit dem letztem Aufruf dieser Funktion im Uhrzeigersinn (Rückgabewert positiv) oder gegen den Uhrzeigersinn (Rückgabewert negativ) gedreht worden ist.

Schließen Sie das Display (Link zur Anschlussbelegung) so an das LaunchPad: jeweils Versorgungsspannung und Masse verbinden, Pin 1.0 an RS (Register Select, hier ein Signal zur Unterscheidung zwischen Befehlen und Daten), Pin 1.1 an E (Enable, das Signal, eine Datenübertragung durchzuführen), Pins 1.4 bis 1.7 an die Datenleitungen D4 bis D7. Verbinden Sie den „Common“-Pin des Dreh-Encoders (Link zur Anschlussbelegung) mit Masse, seinen Anschluss A mit Pin 1.2 und B mit Pin 1.3.

Lassen Sie den Aufbau sicherheitshalber kontrollieren, bevor Sie ihn mit dem PC verbinden.

Im Prinzip kann das Display *zwei* Zeilen (mal acht Zeichen) darstellen; dazu müsste man aber eine negative Spannung an seinen Pin 3 anlegen ( $V_0$ ). Der Einfachheit halber betreiben wir nur die obere Zeile des Displays.

## Programmieraufgaben:

1. Prüfen Sie den Aufbau mit einer `main`-Funktion, die einige Zeichenketten und Zahlen auf das Display schreibt und danach jeweils einige Zeit wartet.
2. Schreiben Sie die Funktion `main` so, dass eine Variable zu Beginn auf den Wert 1000 gesetzt wird und man dann in einer Endlosschleife den Encoder benutzen kann, um diese Variable einzustellen. Es soll dabei immer auch der aktuelle Wert angezeigt werden.
3. Ergänzen Sie die vorige Funktion `main` so, dass die Variable auf  $50^{c1}$  gesetzt wird, wenn man den Encoder eine Sekunde lang nicht gedreht hat. Außerdem soll man nie eine größere Zahl als 100 oder eine kleinere Zahl als 0 wählen können.

$c1$ : 1000

4. Schreiben Sie die Funktion `main` so (Link zum Demo-Video):
  - die Zeichenkette `Zahl 1:` ausgeben
  - warten, bis gedreht wird
  - eine Zahl zwischen 0 und 100 per Dreh-Encoder einstellen, bis mehr als eine Sekunde lang keine Eingabe mehr erfolgt
  - die Zeichenkette `Rechne:` ausgeben
  - warten, bis gedreht wird
  - Eine Rechenoperation aus `plus`, `minus`, `mal`, `geteilt` per Dreh-Encoder wählen, bis mehr als eine Sekunde lang keine Eingabe mehr erfolgt
  - die Zeichenkette `Zahl 2:` ausgeben
  - warten, bis gedreht wird
  - eine Zahl zwischen 0 und 100 per Dreh-Encoder einstellen, bis mehr als eine Sekunde lang keine Eingabe mehr erfolgt
  - die Zeichenkette `Ergibt:` ausgeben
  - eine halbe Sekunde warten
  - zwei Sekunden lang das Ergebnis der gewählten Rechenaufgabe ausgeben, falls es nicht negativ ist, sonst eine Fehlermeldung
  - von vorne beginnen
5. Je nach verbleibender Zeit erweitern Sie das Programm zum Beispiel so:
  - Erlauben Sie auch negative Eingaben und Ergebnisse.
  - Arbeiten Sie mit Festkommazahlen.
  - Erweitern Sie den Rechner um eine Potenzfunktion.
  - Geben Sie bei Überlauf Fehlermeldungen aus.