

Inhaltsverzeichnis

- Tipps fürs Programmieren** 1
 - Software System Design** 1
 - Vorgaben für Bewertung** 1
 - Häufige Fehler und Debugging** 1
 - Allgemeines** 2
 - serielle Schnittstellen** 2
 - Programmierung des ST7565 im Display ERC 128 64 - 1** 2
 - Verwenden von Ports** 3
- Simulide** 3
 - Schrittmotortreiber** 3

Tipps fürs Programmieren

- Sofern es sich um eine Aufgabe mit Hardware-Software-Co-Design handelt, kann mit der Erstellung der Software bzw. der Softwaresystem-Entwicklung bereits parallel zur Schaltplan-Erstellung bzw. direkt danach begonnen werden.
- Machen Sie sich dazu zunächst Gedanken,
 - **was** die Software tun muss (übergeordnete Tätigkeiten),
 - und in **welcher Abfolge** sie das tun soll
- Dann können Sie sich überlegen,
 - wie diese einzelnen Tätigkeiten **C-Funktionen** zugeordnet werden können,
 - wie die C-Funktionen von einander abhängen,
 - welche Schnittstellen die C-Funktionen untereinander benötigen, (Datentyp, Wertebereich, Name)
- **Erst dann sollten Sie überlegen wie der Code aussieht.** Hier hilft ein Blick in die Datenblätter und Application Notes der µController und Chips. Dort sind nicht selten Algorithmen oder Codeschnipsel vorgegeben.

Software System Design

Falls Sie noch nicht wissen, wie die Software oder die zu verwendende Hardware genau genutzt wird, sollten folgende Tipps helfen:

- Suchen Sie die Komponente nicht (nur) auf deutsch. Die Ergebnisse vermehren sich um ein Vielfaches, wenn in Englisch gesucht wird. Zur Übersetzung empfiehlt sich [Linguee](#).
- Nutzen Sie auch die Bildersuche, wenn Sie den genauen Begriff nicht kennen.
- Bei unklaren Begriffen, bieten sich auch folgende Suchworte an: *Arduino* + *<englische Übersetzung von dem „Ding“ was gesucht wird>* + ggf. *Project*. Alternativ bietet sich statt Arduino auch AVR oder Atmel an. Für C-Code kann ein „c code“ zusätzlich gesucht werden. Code liegt häufig auf Github, entsprechend kann sich in Google ein Hinzufügen von `site:github.com` empfehlen.

Vorgaben für Bewertung

Die [Vorgaben für die Programmierung](#) enthalten Hinweise, wie der abgegebene Code aussehen soll.

Häufige Fehler und Debugging

- Tipps zum Debugging und zu häufigen Fehlern finden sich bei den [Tipps für die Fehlersuche \(Elektronik-Labor\)](#).
- Versuchen Sie möglichst nach jeder kleinen Änderung Ihr Programm zu testen. Wenn Sie drei Punkte ändern und dann erst testen, dann wissen Sie nicht, an welcher Änderung es liegt!
 - Die 25 häufigsten Fehler beim Programmieren

[oder als](#)

Paper

[und weitere](#)

41 häufige Fehler

Verwenden von Ports

- Folgendes ist zu beachten, falls Sie JTAG-Ports - z.B. PF4..7 bei ATMEGA16U4 - anderweitig verwenden wollen/müssen (JTAG Ports = Ports an denen die ProgrammierHW angeschlossen wird): Die JTAG-Ports können nicht ohne weiteres direkt genutzt werden. Die JTAG Schnittstelle muss zunächst über folgenden Code deaktiviert werden.

```
MCUCR |= (1<<JTD) ;  
MCUCR |= (1<<JTD) ;
```

Wichtig: Es das Control Register muss zweimal geschrieben werden.

- Für eine zeitkritische Ausgabe von aufeinanderfolgenden Bits (z.B. für die Ansteuerung von [intelligenten LEDs](#)) müssen unbedingt Interrupts genutzt werden. Es lohnt sich zusätzlich auf [USART](#) zurückzugreifen. Bei USART werden die zu versendenden Daten zunächst ins UDRn Register gegeben und dann in das Shiftregister übertragen.
- Falls Sie einen externen Oszillator oder Quarz benutzen, werden zwei Ports dafür verwendet (Ports XTAL = „Crystal“). Wenn Sie diese Ports per DDR versehentlich zu einem Ausgang definieren, hat der Chip keinen Takt mehr. Das heißt diese Portzuweisung ist das letzte was der Chip macht... Es ist danach nur noch per Debuggerschnittstelle möglich diesen wieder zu beleben.

Simulide

Schrittmotortreiber

Verschiedene Schrittmotortreiber bzw. Steppermotor Driver (wie z.B. [DRV8825](#)) ermöglichen durch die Angabe der Richtung und ein Takten des STEP Pins das Antreiben eines Schrittmotors. Solche Treiber sind in Simulide nicht verfügbar. Folgende Simulation ermöglicht aber zumindest die Nachbildung [stepper.rar](#)

From:

<https://wiki.mexle.org/> - **MEXLE-Wiki**

Permanent link:

https://wiki.mexle.org/microcontrollertechnik/tipps_fuers_programmieren

Last update: **2022/04/05 01:35**

