

Allgemeines

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

- Systemdesign 2
 - Allgemeines* 2
 - Motoren* 3
 - Motorelektronik* 3
- Auswahl des Prozessors 4
 - für die Veranstaltungen EST, Elektronik und ES an der HS Heilbronn* 4
 - für weitere Projekte* 4

Systemdesign

Allgemeines

- **Schaltung modularisieren** : Bei Schaltungen und Software bietet es sich an, stark zu modularisieren. Ziel ist dann unter anderem die einzelnen Module separat entwickeln und testen zu können. Dazu sollte gewährleistet werden, dass die Module auch tatsächlich einzeln testbar sind. Hierfür können niederohmige Widerstände ("0 Ohm") oder Lötjumper genutzt werden. Mit diesen lassen sich Teile der Schaltung bei Bedarf separieren oder verbinden.
- **Komponenten suchen**: Falls Sie noch nicht wissen, wie Ihre gesuchte Komponente heißt, sollten Ihnen folgende Tipps helfen:
 - Suchen Sie die Komponente nicht (nur) auf deutsch. Die Ergebnisse vermehren sich um ein Vielfaches, wenn in Englisch gesucht wird.
 - Nutzen Sie 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 Häufig hilft auch die Suche nach "IC" + *<..>* für integrierte Schaltkreise.
- **Eignung der gefundenen Komponenten überprüfen**: Mouser und TME bieten eine gute Detailsuche für bekannte Komponenten. Diese lässt eine Eingrenzung der Komponenten durch die Auswahl von Eigenschaften zu. Nutzen sie diese, wenn Sie auf der Suche nach neuen Komponenten sind.
 - Wählen Sie bei der Suche auch "auf Lager" aus, damit nur die vorhandenen Komponenten angezeigt werden.
 - Wichtig ist bei einem geeigneten Bauteil immer der Blick ins Datenblatt. Die Suchmaschinen verwenden gelegentlich veraltete und falsche Angaben.
- **Blick in Datenblätter** : Prüfen Sie
 - den elektrischen Arbeitsbereich. Die "**Absolute maximum ratings**" sind absolute Grenzen! D.h. in der Regel keine Spannungen, Ströme und Temperaturen für den Normalgebrauch. Um die Komponente für den Normalgebrauch auszulegen, ist die Betrachtung der "**Electric Characteristics**" besser geeignet. Bitte den Fokus auf typical und nicht min/max Ratings legen.
 - ob die Komponente für Ihre **Versorgungsspannung** geeignet ist
 - ob das **Packaging** für Ihre Platine geeignet ist (THT oder SMD?),
 - welche **weiteren Komponenten** (Widerstände, Induktivitäten, Kapazitäten, Dioden etc) benötigt werden,
 - bei **Strommess-Widerständen** (auch Shunt, Rsense) nicht nur die Formel im Datenblatt des Treibers zu berücksichtigen, sondern auch die maximale Leistung des Widerstands. Wenn z.B. 0.2 Ω Widerstände nur bis 0.2 Watt Maximalleistung verfügbar sind, so wäre der Maximalstrom: $I = \sqrt{P \over R} = 1 \text{ A}$
 - verschiedene Größen bei **Induktivitäten**: Achten Sie darauf, dass Induktivitäten **ab 10..100 μH** nicht mehr günstig in einem kleinen Formfaktor (z.B. 0603 oder 0402) erhältlich sind. Diese benötigen somit eine **größere Fläche** auf dem Board. Weiterhin haben Induktivitäten häufig ein Strom-begrenzenden Widerstand, u.a. um eine Sättigung zu vermeiden. Beachten Sie die begrenzte Auswahl der Distributoren!
 - verschiedene Größen bei **Kapazitäten**: Auch Kapazitäten **ab 10 μF** sind nicht mehr günstig in einem kleinen Formfaktor (z.B. 0603 oder 0402) erhältlich. Gelegentlich werden für Anwendungen oder Bauteile bestimmte, teurere **Dielektrika** (z.B. NP0) benötigt. Hier ist das Datenblatt / die Anwendung auf Hinweise zu überprüfen (Frequenzbereich, Relevanz von Alterung und Temperatur-/Spannungsabhängigkeiten, ..).

- die **Schnittstellen**: Nicht nur der Footprint einer Komponente ist wichtig, sondern auch die Belegung.
- Wenn Sie die Komponenten gesucht haben, sollten Sie die Kosten und Verfügbarkeit prüfen (z.B. sind nur noch 10 beim Distributor verfügbar?). Es sind Komponenten aus dem Elektroniklager (siehe [Part-DB](#)) zu bevorzugen. Sind Ihre gewünschten Komponenten nicht vorhanden, suchen Sie diese bei [unseren Lieferanten](#) und geben Sie dem Betreuer bescheid.

Motoren

Soll ein Motor verwendet werden, so muss eine sinnvolle Auswahl des Motors getroffen werden. Dabei ist folgendes zu beachten:

	Schrittmotor	Gleichstrommotor	Servoantrieb	Synchronmotor	Asynchronmotor
Alternative Bezeichnungen	Steppermotor	DC-Motor	Servo	PSM, ESM	ASM
Positioniergenauigkeit (open loop)	hoch, wenn ohne Überlast betrieben (Teilschritte möglich)	niedrig	mittel-hoch (abhängig vom internen Regler)	niedrig	niedrig
Drehzahlbereich	niedrig (bis ca. 20 Hz)	hoch (bis 1000 Hz)	mittel	hoch	hoch
Drehmoment	hoch	hoch	niedrig	hoch	hoch
Wirkungsgrad	niedrig	mittel	mittel	hoch	hoch

Gelegentlich werden bei Schrittmotoren 2 weitere Spulen mit höherem Widerstand verbaut (z.B. 2x 1 Ohm und 2x 10 Ohm). Es werden nur 2 Spulen mit gleichem Widerstand zum Betrieb benötigt.

Weiterhin gibt es mechanische Möglichkeiten die o.g. Eigenschaften zu ändern (z.B. DC-Motor mit 1:1000 Getriebe). Dabei ist zu beachten, dass diese i.d.R. in Akustik und im Wirkungsgrad negative Einflüsse zeigen. Zusätzlich muss speziell bei der Verwendung eines Schneckengetriebe beachtet werden, dass eine Selbsthemmung auftritt - Die Achse lässt sich nicht freidrehen.

Die Ansteuerung von Schrittmotor, Gleichstrommotor, Synchronmotor und Asynchronmotor geschieht über ein Leistungsendstufen. Details zu den Konzepten der Halb- und Vollbrücken finden Sie z.B. auf mikrocontroller.net. Heutzutage bietet es sich aber an die Leistungselektronik nicht mehr diskret aufzubauen, sondern integrierte Komponenten zu verwenden (siehe [motoren_und_treiber](#)).

Motorelektronik

1. Außer bei Servomotoren und einigen BLDC-Motoren, ist das Ansteuern des Motors nicht direkt über die Pins des Mikrocontrollers möglich. Bei Schrittmotoren und DC-Motoren muss dazu hinreichend Strom zur Verfügung gestellt werden. Dies geschieht über folgende notwendige Komponenten:
 1. Leistungsschalter
 1. schaltet den benötigten Strom
 2. meist MOSFETs
 3. in Struktur einer Halb-, Voll-, B6-Brücke
 2. Gatetreiber
 1. stellt die richtige Spannung zum Schalten der Leistungsschalter bereit
 2. verhindert Kurzschlüsse
2. Einige ICs wie DRV88xx haben bereits beide Komponenten in einem Chip vereint.

3. Bei einem diskreten Aufbau der Leistungsschalter sollten Leistungsdioden vorgesehen werden
4. Es kann sich anbieten bei Leistungselektronik-Komponenten - auch beim DRV88xx - eine Möglichkeit des Austauschs vorzusehen, z.B. über einen Sockel. Dazu müssen die Komponenten als [Through Hole Device](#) ausgelegt werden.

Auswahl des Prozessors

für die Veranstaltungen EST, Elektronik und ES an der HS Heilbronn

- Als Microcontroller wird ein 8bit AVR Microchip Chip (ehemals ATMEL AVR) empfohlen. Auf diesem Chipsatz wurde in Microcontrollertechnik aufgebaut. Im Elektronik-Lager stehen einige ATmega32x8 zur Verfügung.
- In den grundlegenden 1×1 MexlePlatinen ([MmC1x1 328_1.x](#), [MmC1x1 328_2.x](#), [MmC1x1 32U4](#)) sind ATmega32xx verbaut, welche prinzipiell mit 5 V oder 3,3 V betrieben werden können. Die gesamte Schaltung sollte aber auf 3,3 V ausgelegt werden. Damit ist die Verwendung aktueller Sensoren möglich.
- Für die Auswahl eines Chips für Zusatzplatinen ist auf der Herstellerseiten von [AVR](#) eine übersichtliche Tabellen zu finden. Für ATtiny104, ATmega88, ATmega32xx ist Prototypen-Equipment vorhanden.
 - Die Chip-Varianten mit "U" haben dabei direkt eine Möglichkeit einen USB-Anschluss direkt zu nutzen

für weitere Projekte

Um den geeigneten Mikrocontroller für ein eigenes Projekt herauszusuchen sollten folgende Fragen geklärt werden:

- Wie viel **Speicher** wird benötigt?
 - Wie groß wird das Programm? Werden größere Librarys eingebunden?
 - Wie groß muss der Flash-Speicher sein (in diesem wird der Code gespeichert)
 - Wie viele Variablen und Konstanten beinhaltet das Programm und wie groß sind diese (Arrays? int8, int32, float?)
 - Wie groß muss das RAM sein? (Konstanten können alternativ auch im Flash liegen)
 - Sollen Werte permanent gespeichert werden? Wie viele?
 - Wie groß muss das EEPROM sein?
- Welcher **Prozessortakt** wird benötigt?
 - Wie schnell muss der Mikrocontroller mit wie vielen Daten reagieren? Muss z.B. für eine Regelungsschleife Daten schnell aufgenommen, verarbeitet oder ausgegeben werden?
 - Können ggf. Funktionalitäten einem Timer/Counter übergeben werden?
 - Soll der Takt intern erzeugt werden, oder soll ein externer Quarz verwendet werden?
- Welche **Interfaces** werden benötigt?
 - Welche digitalen Schnittstellen (I2C, SPI, USB, USART, PWM) werden benötigt?
 - Wie schnell und genau müssen wieviele Analogwerte aufgenommen werden?
 - Wie schnell und genau müssen wieviele Analogwerte ausgegeben werden?
 - Welche und wie viele der genannten Schnittstellen werden gleichzeitig benötigt? Pin Count
- Welche **Infrastruktur im Controller** wird benötigt?
 - Soll auf ein oder mehrere Ereignisse (Timer, Benutzereingabe etc.) asynchron reagiert

- werden? Interrupts
- Wird viel mit Gleitkommawerten gerechnet? Controller mit FPU (z.B. PIC)
- Welche **Infrastruktur um den Controller** ist zu erwarten?
 - Temperaturbereich
 - Leistungsaufnahme
 - Spannungsniveau
- Welche **nicht-elektronischen Randbedingungen** sind zu beachten?
 - Kosten und Liefersituation
 - Wissensbasis und Verbreitung
 - Handling bei der Produktion (Chipgehäuse, Verpackung, ..)
 - Pre-programmed lieferbar?

From:

<https://mexle.te.hs-heilbronn.de/> - **MEXLE Wiki**

Permanent link:

https://mexle.te.hs-heilbronn.de/elektronik_labor/tipps_fuer_komplexere_aufbauten?rev=1585091830

Last update: **2021/05/09 10:07**

