

# 3 Entwickeln des Schaltplans

## Student Group

First Name	Surname	Matrikel Nr.

## Table of Contents

- 3 Entwickeln des Schaltplans** ..... 2
  - Prozess 3: Schaltplan erstellen* ..... 2
- 101 - Schaltplan zeichnen** ..... 2
  - Anlegen der Dateien* ..... 2
  - Beschriftung und erste Struktur* ..... 2
  - Verbindungen und Potentiale* ..... 3
  - Komponenten allgemein* ..... 4
  - passive Elemente* ..... 5
  - Mikrocontroller* ..... 5
  - Analoge Eingangswerte (ADC)* ..... 6
  - Digitale Schnittstellen* ..... 6
- Zum Abschluss** ..... 6
- Generelles** ..... 7

# 3 Entwickeln des Schaltplans

## Prozess 3: Schaltplan erstellen

Input Products	Work	Output Products
IP1: Projekidee IP2: Projekt-Definition	W1. Durchsicht der benötigten Datenblätter W2. Erstellen der Schaltung (*.sch) in eagle (Ändern eines vorandenen Boards) W3. Regelmäßige Durchsprache mit dem Betreuer W4. Ablegen der genutzten Datenblätter W5. Informieren des Betreuers	OP1: Schematic-files OP2: Datenblätter
Input Condition		Output Condition
IC1: Projektidee vom Betreuer abgenommen		OC1: Files abgelegt
<a href="#">checkliste_fuer_die_schaltungserstellung</a> Anleitung für die <a href="#">weitergabe_der_bom</a> eine kurze Liste von <a href="#">Eagle Tipps</a>		

## 101 - Schaltplan zeichnen

### Anlegen der Dateien

Eine neue Platine des Mexle-Systems sollte auf Basis einer vorhandenen Module erstellt werden, z.B.

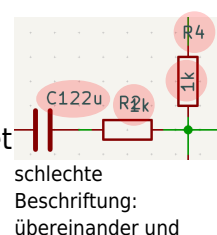
1. Einzoll-Module auf Grundlage der [MMC 1x1 328PB](#) Platine
2. Hookup-Module auf Grundlage der [MCB 1x1 Basis Hookup](#) Platine
3. Mehrzollmodule auf Grundlage der [MCB 1x1 Basis Platine](#)

Zum Anlegen einer neuen Datei ist auf die Templates ([hier](#) oder im ILIAS Kurs). Nachdem die \*.zip Datei heruntergeladen wurde, sollte wie folgt vorgegangen werden:

1. Die zip Datei muss im Ordner \Dokumente\KiCAD\7.0\template geschoben und entpackt werden.
2. Anpassen der Schematic:
  1. Die Vorlage kann in KiCAD durch Datei » Neues Projekt aus einer Vorlage ... geöffnet werden
  2. Die nicht notwendigen Komponenten können durch Markieren (z.B. Ziehen einer Auswahl mit der Maus) und Löschen (z.B. drücken der <Entf>-Taste) entfernt werden

### Beschriftung und erste Struktur

- Füllen Sie den Kommentarbereich in der Zeichnung aus und halten Sie ihn aktuell (Autor, Schaltungszeichnung, Version, Datum).
- Legen Sie zunächst das grobe Layout auf Basis der Funktion fest. Eingänge sollten links, Ausgänge rechts zu positioniert sein. Bei kleineren Boards bietet sich die Struktur an, wie sie zukünftig auf der Platine wiederzufinden sein wird.
- Der Name (unter Eigenschaften) von Bauteilen sollte möglichst kurz sein. Die



Bezeichnung von Anschlüsse (z.B. Motor, RXD, ADC) und Bauteile (z.B. ATmega88) sollte sinnvoll und aussagekräftig sein. Hinweise für die Benennung der Komponenten finden Sie in [Wikipedia](#).

über Komponente

- Name und Bezeichnung sollten möglichst nahe am Bauteil positioniert sein. Vermeiden Sie Überlagerungen von Text bzw von Text und Komponenten. Falls Sie eine doppelte Beschriftung haben, entfernen Sie eine davon.
- Falls Sie bei den Bezeichnern eine negierte Benennung benötigen (Z.B. Out Nicht Reset), so erhalten Sie einen Überstrich mittels Ausrufezeichen:  
$$\text{OUT}_{\sim\{\text{Reset}\}} = \text{OUT}_{\overline{\{\text{Reset}\}}}$$
- Nutzen Sie aussagekräftige Darstellungen für Bauteile, also z.B. Transistoren statt Blackboxen.
- Falls Sie eine Komponente neu anlegen: Der Ursprung ("Origin") soll mittig positioniert werden; die neue Komponente sollte möglichst kompakt gestaltet sein.
- Zeichnen Sie nur Bauteile ein, welche auch auf der Platine zu finden sein werden. Also keine externen Sensoren, Motoren etc. die keinen Platz auf der Platinen benötigen.

## Verbindungen und Potentiale

- Geben Sie den Verbindungen kurze und aussagekräftige Namen, bevorzugt eine Funktionsbezeichnung ("MISO"). Alternativ kann auch der Pin-Namen vom Mikrocontroller ("PB4") oder eine Kombination ("PB4\_MISO") verwendet werden.
- Nicht benötigte Verbindungen sollten entfernt werden.
- Versuchen Sie Verbindungen möglichst kurz und kreuzungsfrei zu zeichnen.
- Gegen zuviel Spaghetti auf der Schaltung, hilft die Verwendung von:
  - **Linien-Stummel**: an den beiden Komponenten wird jeweils nur eine kurze, gerade Verbindung gesetzt. Durch die "Netzbezeichner"-Funktion (Shortcut L und Klick auf die Verbindung) können beide kurze Enden gleich benamt werden.
  - **vorgegebene Potentialbauteile** wie GND, -5V oder +5V häufiger nutzen (Shortcut P für power connector)
- Als Bezeichnungen für die verschiedenen Versorgungsspannungen von IC werden in der Industrie i.d.R. entweder VDD5, VDD33, VDD18 oder VDD5V, VDD3V3, VDD1V8 geschrieben. Bei älteren Schaltungen wurde VCC (Common Collector Voltage) statt VDD verwendet.

Symbol	Anwendung	Empfehlung
	<p><b>GNDA:</b> Analog Ground: ungestörte Masse für Analogsignale, sollte für alle analoge Signale genutzt werden</p> <p><b>GND:</b> Ground: allgemeines Masse-Symbol</p> <p><b>GNDD:</b> Digital Ground: Masse für Bauteil-interne Logiken, hat meist hochfrequente Spannungswechsel</p> <p><b>GNDPWR:</b> Power Ground: Masse für leistungsführende Potentiale, z.B. Motorspannung, hat meist hochfrequente Spannungswechsel und kann durch den geführten Strom einen Spannungsabfall beinhalten</p> <p><b>GNDS:</b> (hochspannungs)abgeschirmte Datenübertragung, Bezug für IO-Pins</p> <p><b>GNDS</b> Signal Ground: eingangsseitige Masse (Alternative für GNDA falls belegt)</p> <p><b>GNDREF</b> Reference Ground: Bezugspotential, z.B. über einen Spannungsteiler erzeugt. Kann u.a. bei der Mittenspannung für unipolar versorgte Operationsverstärker genutzt werden</p>	<p>Verwendung von GND und AGND</p>
	<p><b>+3.3V:</b> positive Spannungsversorgung (nach Rücksprache auch als 5.0V o.ä.)</p> <p><b>+3V3:</b> diese Bezeichnung der positiven Spannungsversorgung bitte nicht nutzen</p> <p><b>-3V3:</b> diese Bezeichnung der negativen Spannungsversorgung bitte nicht direkt nutzen. Benennen Sie diesen um zu -3.3V</p> <p><b>VSW</b> Spannung für die Simulation der Schaltung, bitte nicht nutzen</p> <p><b>VDC:</b> positive Gleichspannung, kann ggf. genutzt werden</p> <p><b>+3.3VADC:</b> positive Spannungsversorgung eines ADC, kann ggf. genutzt werden</p> <p><b>+3.3VDAC:</b> positive Spannungsversorgung eines Digital-Analog-Wandlers, kann ggf. genutzt werden</p> <p><b>+3.3VDA:</b> positive Spannungsversorgung von Analogkomponenten, kann ggf. genutzt werden</p> <p><b>+3.3VDP:</b> positive Spannungsversorgung mit Leistung - z.B. für Motoren, kann ggf. genutzt werden</p>	<p>Verwendung von VCC-Symbol mit Spannungsbeschriftung bei unipolar gespeisten Schaltungen. Verwendung von positiver und negativer Spannungsversorgung bei bipolar gespeisten Schaltungen</p>

## Komponenten allgemein

- Teilweise haben Komponenten, welche über Seiten von [Distributoren](#) (z.B. Mouser) stammen, unglückliche Formen und Pin-Sortierungen. Z.B. Transistoren oder LEDs, die als viereckige (Black)box erscheinen oder eine Pinsortierung, welche zwar die des Chips entspricht, aber nicht eine sinnvollen Reihenfolge (z.B. wie die Schaltungsskizze im Datenblatt oder der numerischen Sortierung).

## passive Elemente

- Verwenden Sie als Wert für Widerstände, Kondensatoren und Induktivitäten für die Größenbeschreibung folgende Symbole: f, p, n, u, m, k, M, G, T (kein  $\mu$ , MEGA, o.ä.). Die physikalische Einheit wie  $\Omega$ ,  $F$ ,  $H$  ist wegzulassen (im Gegensatz zu Rechnungen).

## Mikrocontroller



Die folgenden Tipps beziehen sich allgemein auf Schaltungen mit Mikrocontroller.

- Resetleitung
  - Der Resetpin sollte über einen Pull-up/-down Widerstand auf der nicht resetierenden Spannung gehalten werden (bei ATmega32U4 per Pull-up auf High). Weiterhin ist ein Mikrotaster von Vorteil, über welchen der Pin auf die resetierende Spannung gezogen werden kann (=“Reset-Knopf”, bei ATmega32U4 per Pulldown auf Low).
- Taktgeber
  - Überlegen Sie sich, ob Ihre Anwendung eine genaue Taktfrequenz oder kompakten/kostengünstigen Aufbau benötigt

Eigenschaften	interner RC-Oszillator	Quarz	(Quarz-)Oszillator
<b>Genauigkeit</b>	$\pm 1\%$ <sup>1)</sup>	$<0,01\%$ <sup>2)</sup>	$<0,01\%$ <sup>3)</sup>
<b>Auswahl mögl. Frequenzen</b>	eine feste (z.B. Atmega32U4 8 MHz), die per Prescaler geteilt werden kann (CLKPR)	große Auswahl (alle Baudraten)	kleine(re) Auswahl
<b>notwendige externe Bauteile</b>	keine	2 weitere Kondensatoren und Quarz	nur Oszillator
<b>Geeignet für</b>	kostenkritische Anwendungen	zeitkritische Anwendungen: UART, USART, genaue Uhr	

Details sind im Data Sheet unter Kapitel “Clock Sources” (bei ATmega32U4 Kapitel 6.2) zu finden. Für energiesparende Anwendungen sollte der Takt reduziert werden (siehe Tabelle).

Frequenz	Anwendung
32,768 kHz	für energiesparende Schaltungen (Stimmgabelquarz)
18,432 MHz	<a href="#">Baudratenquarz</a> , notwendig um viele Schnittstellentakte zu generieren. Maximalwert für die meisten AVR-Chips

- Weitere passive Komponenten
  - Überprüfen Sie das Data Sheet auf weitere notwendige passive Elemente, z.B. für interne Regler o.ä.  
Diese sind unter anderem über Suchwörter wie “Design Guidelines” zu finden.
- USB Anschluss / Anschluss an externe Elektronik

- Bei digitalen Anschlüssen an externe Elektronik bietet es sich an ESD Schutzdioden zu verwenden, um die Komponenten zu schützen.

## Analoge Eingangswerte (ADC)

- Nutzen Sie für alle analoge Komponenten (Operationsverstärker, MOSFETs, ...) die Signale "Analog Vcc" (AVCC) und "Analog Ground" (AGND). Damit reduzieren sich Störungen auf der analogen Seite.
- Analog Ground und Digital Ground sollten nahe am Prozessor nur an einer Stelle verbunden werden. Dazu ist am einfachsten ein kurzgeschlossener Lötjumper zwischen AGND und GND zu verwenden.
- Sinnvoll beim Einlesen von Analogwerten ist es weiterhin ein Tiefpass vorzuschalten, um Spannungsspitzen abzdämpfen. Z.B. (Grenzfrequenz beachten!)
  - 1. Ordnung: Widerstand mit 1 kOhm und 100 nC gegen Masse, oder
  - 2. Ordnung: Induktivität mit 10 µH und 100 nC gegen Masse.

## Digitale Schnittstellen

- Schalter als Input werden sinnvollerweise bei Schaltungen mit ATmega-Prozessoren gegen GND geschaltet. Dann können die internen Pull-up Widerstände genutzt werden.
- Falls Sie einen USB-Anschluss robust auslegen wollen, so sollten Sie sich an die üblichen Design Guides richten (z.B. [Design Guide von DataKey.com](#) )
- **Wir wollen einen RasPi über I2C anschließen ODER Wir wollen 5V und 3,3V I2C an einem Bus haben.** Es ist zu beachten, dass der I2C des RasPi für High Pegel 3,3V liefert. Die meisten älteren Schaltungen benötigen 5V. Als Lösung können Sie den Levelshifter in folgende Application Note betrachten: [an10441.pdf](#)
- **Wie kann ich einen zweiten SPI an meinen ATmega anschließen?** Verwenden Sie dazu die USART Schnittstelle als zweite SPI. Details dazu finden Sie im Datenblatt des Prozessors unter USART. zweite Alternative: nutzen Sie einen geeigneten AVR-Chip (z.B. ATmega328PB)
- **Ich will viele Schalter einlesen. Ich will einen möglichst energiesparenden Aufbau.** dazu bietet sich ein Blick in folgende Dateien [PIC-Tipps](#). Einige der Tipps sind inzwischen überholt, können aber genutzt werden.
- Falls Sie einen SMD Mikrocontroller verwenden, sollten Sie nicht vergessen einen Programmierport herauszuführen - am besten ISP über [debugWire](#) (6-polige Buchse). Ansonsten wird das Programm nie den Weg auf den Chip finden.

## Zum Abschluss

- **Prüfen, ob alle Teile in der Schaltung im Haupttraster liegen:** in der Kommandozeile "run snap-on-grid-sch.ulp" eingeben
- **Electrical-Rule-Check durchführen:** in der Kommandozeile "ERC" eingeben.
- Spannungsniveaus beschreiben: Schreiben Sie, für welche Spannung(en) die Schaltung ausgelegt ist. z.B. VCC: \$3.3 ... 5.0 ~\rm V\$, V\_power: \$20..36 ~\rm V\$, Vin: \$1.8 ~\rm V\$

# Generelles

- Alle ICs mit Kondensatoren an den Stromversorgungspins versehen!!! Diese müssen nahe am IC liegen, um gepulste Leistungsabfrage zu kompensieren (siehe auch [Simulation von möglichen Störungen](#)). Die Kondensatoren kompensieren Störungen auf der Versorgungsleitung. Sinnvoll sind hier Keramikkondensatoren mit 100 nF, da diese einen geringen [äquivalenten Serienwiderstand](#) haben. Auch bei Operationsverstärker, ADC-ICs, ... sind diese vorzusehen.
- Theoretisch können Sie alle Ports zu Ein/Ausgänge umkonfigurieren. In der Praxis sind die verwendeten Ports **weise** zu wählen:
  - Es ist umständlich, aus einem mit JTAG belegten Pin einen Ausgabe-Pin zu machen.
  - Es ist manchmal besser einen Analoginput zur Eingabe von mehreren Binärwerten zu nutzen (z.B. über [R-2R-Netzwerk](#), [Falstad](#)).
- Berücksichtigen Sie bereits bei Der Schaltungserstellung das Layout. Es ist einfacher von einer vorhandenen Schematic **und Board** auszugehen und dort die nicht benötigten Komponenten zu löschen.
- Falls Sie Eagle per Tastatur bedienen wollen, können Sie auf den Seiten des [MIT](#) die Editor-Commands finden

1)

Datenblatt ATmega32U4

2)

<https://www.mouser.de/search/refine.aspx?N=11749875>

3)

[https://www.mouser.de/Passive-Components/Frequency-Control-Timing-Devices/Oscillators/\\_/N-7jdf/](https://www.mouser.de/Passive-Components/Frequency-Control-Timing-Devices/Oscillators/_/N-7jdf/)

From:

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

Permanent link:

[https://wiki.mexle.org/elektronik\\_labor/3\\_entwickeln\\_des\\_schaltplans?rev=1714641116](https://wiki.mexle.org/elektronik_labor/3_entwickeln_des_schaltplans?rev=1714641116)

Last update: **2024/05/02 11:11**

