

# Fachbericht

PROJEKT 6 COCKTAILMASCHINE - TEAM SCHENK & AEBI  
2. April 2020

**Betreuender Dozent:**

Prof. Dr. Schleuniger, Pascal

**Team:**

Schenk, Kim  
Aebi, Robin

**Studiengang:**

Elektro- und Informationstechnik

**Semester:**

Frühlingssemester 2020

## **Abstract**

In diesem Projekt wurde ein Konzept erstellt, um eine Cocktailmaschine zu bauen. Dies reicht von der Analyse, was für Cocktailmaschinen es bereits gibt, über die Erstellung eines Grob- und eines Detailkonzeptes bis hin zur Evaluation der Komponenten. Der Aufbau wurde so gewählt, dass ein Glas mittels eines Linearantriebes auf einem Schlitten hin- und her gefahren wird und unter dem gewünschten Flüssigkeitsauslass stehen bleibt, wo es dann befüllt wird. Die Bedienung soll über ein Touch-Display geschehen. Die Verarbeitung der Daten wird ein Mikrocontroller übernehmen. Als mechanische Komponente wird pro Zutat eine Pumpe und ein Durchflusssensor verwendet sowie ein einzelner Motor, welcher den Linearantrieb mit dem Schlitten betreibt. Als Motor wurde ein bürstenloser Gleichstrommotor verwendet, da dieser ein sehr gutes Leistungs-/Gewicht-Verhältnis aufweist und in seiner Ansteuerung sehr interessant ist. Ziel des Projekt 5 war es, anhand des Konzeptes die einzelnen Teilsysteme aufzubauen und deren Funktion zu verifizieren und zu dokumentieren. Softwaremässig wurde die Basis für den Mikrocontroller geschrieben. Dies bedeutet, dass die Teilsysteme kontrollierbar sind und im Projekt 6 ausgebaut und zusammengeführt verwendet werden können. Die Software wurde komplett in C geschrieben und ausgiebig dokumentiert. Das Resultat zeigt, dass die Komponenten zusammenpassen und der Cocktailmaschine im Projekt 6 nichts im Weg steht.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Ausgangslage</b>	<b>2</b>
2.1	Blockschaltbild . . . . .	2
2.2	Komponentenauswahl . . . . .	2
<b>3</b>	<b>Neue Hardware</b>	<b>3</b>
3.1	USB-B . . . . .	3
3.2	Wirelessmodul . . . . .	4
3.3	RFID . . . . .	5
3.4	Beleuchtung . . . . .	7
<b>4</b>	<b>Printaufbau</b>	<b>8</b>
<b>5</b>	<b>Teilsysteme</b>	<b>8</b>
5.1	Speisungen . . . . .	8
5.1.1	48V Speisung . . . . .	8
5.1.2	12V Speisung . . . . .	9
5.1.3	5V Speisung . . . . .	10
5.1.4	3.3V Speisung . . . . .	11
5.2	Motor . . . . .	12
5.2.1	BLDC und H-Brücke . . . . .	12
5.2.2	ABN-Encoder . . . . .	12
5.2.3	Treiber . . . . .	12
5.3	Flüssigkeitsbeförderung . . . . .	12
5.3.1	Pumpen . . . . .	12
5.3.2	Durchflussmessgeräte . . . . .	12
5.4	Benutzerschnittstellen . . . . .	13
5.4.1	Display . . . . .	13
5.4.2	ESP . . . . .	13
5.4.3	USB-C . . . . .	13
5.4.4	RFID . . . . .	13
5.5	Beleuchtung . . . . .	14
5.6	Mikrocontroller . . . . .	16

<b>6</b>	<b>Inbetriebnahme</b>	<b>16</b>
6.1	Speisungen . . . . .	16
6.1.1	12V Speisung . . . . .	16
6.1.2	5V Speisung . . . . .	16
6.1.3	3.3V Speisung . . . . .	16
6.2	Motor . . . . .	16
6.2.1	BLDC und H-Brücke . . . . .	16
6.2.2	ABN-Encoder . . . . .	16
6.2.3	Treiber . . . . .	16
6.3	Flüssigkeitsbeförderung . . . . .	16
6.3.1	Pumpen . . . . .	16
6.3.2	Durchflussmessgeräte . . . . .	16
6.4	Benutzerschnittstellen . . . . .	16
6.4.1	Touch-Display . . . . .	16
6.4.2	ESP . . . . .	16
6.4.3	USB-C . . . . .	16
6.4.4	RFID . . . . .	16
6.5	Beleuchtung . . . . .	16
<b>7</b>	<b>Software</b>	<b>16</b>
7.1	Strukturplan . . . . .	17
7.2	Programmflussdiagramm . . . . .	17
<b>8</b>	<b>Evaluation</b>	<b>17</b>
<b>9</b>	<b>Fazit</b>	<b>17</b>
9.1	Zielerreichung . . . . .	17
9.2	Kosten . . . . .	17
<b>10</b>	<b>Schlusswort</b>	<b>17</b>
<b>11</b>	<b>Ehrlichkeitserklärung</b>	<b>17</b>

# 1 Einleitung

Eine gelungene Party auf die Beine zu stellen verlangt einem einiges ab. Vor allem kostet es eine Menge Aufwand und Zeit. Dies gilt besonders, wenn es darum geht mit vielen Freunden zusammen zu feiern. Neben der gelungenen Musikauswahl und den Snacks darf eines auf gar keinen Fall fehlen, die Getränke. Um diese sicherzustellen, gibt es mehrere Möglichkeiten. Einerseits könnte jeder seine eigenen Getränke mitbringen, was jedoch bedeutet, dass es unter Umständen eine riesige Sauerei gibt oder viele Flaschen in der Gegend rumstehen. Andererseits könnte man als Gastgeber selber anbieten Cocktails zu mixen und so den Getränkenachschub zu gewährleisten. Da gibt es jedoch ein grosses Problem. Denn wären wir die Gastgeber, so würden wir nicht den ganzen Abend hinter der Bar stehen wollen, sondern lieber bedenkenlos mitfeiern. Damit genau dies möglich ist haben wir uns in diesem und dem nächsten Projekt (5&6) dazu entschieden eine automatisierte Cocktailmaschine zu entwerfen. Diese soll vollkommen autonom arbeiten und sollte problemlos von jeder beliebigen Person und in fast jedem Zustand bedient werden können.

In den folgenden Kapiteln ist dokumentiert, wie die Cocktailmaschine aussehen soll und aus welchen Teilsystemen diese bestehen wird. Ausserdem werden die einzelnen Teilsysteme genauer unter die Lupe genommen und in einem systemspezifischen Testverfahren evaluiert. Dieses Projekt bietet demnach die Basis des Projekt 6 und soll dieses so gut wie möglich vorbereiten.

## **2 Ausgangslage**

### **2.1 Blockschaltbild**

### **2.2 Komponentenauswahl**

## 3 Neue Hardware

Im folgenden Kapitel werden die Hardware-Teile beschrieben, welche im Projekt 5 noch nicht erarbeitet wurden. Zuerst werden grundlegende Anforderungen und Inhalte beschrieben, welche zur Auswahl der Bauteile geführt hat.

### 3.1 USB-B

Auf der Leiterplatte des PartyMixer's gibt es zwei Komponenten, welche programmiert werden müssen. Der Mikrocontroller und das WiFi-Modul. Beide werden über die UART-Schnittstelle 0 programmiert. Um diese zu programmieren braucht es eine entsprechende Schnittstelle, welche mit einer USB-B-Schnittstelle realisiert wird.

Die USB-B-Schnittstelle benötigt jedoch nur zwei Kommunikationsleitungen (D+ und D-). Deswegen benötigt es einen USB-UART-Converter. Hier wurde auf das Flash-Interface von Arduino zurückgegriffen `arduino_cc_arduino_2017`, um die Schaltung für den Mikrocontroller zu planen und auf das Flash-Interface des ESP32 `espressif_systems_esp32_2016`, um die Schaltung für den ESP32 zu planen. Darin ist ersichtlich, dass für den Mikrocontroller und ESP folgende Leitungen nötig sind:

- Mikrocontroller
  - RX ==> TX
  - TX ==> RX
  - Reset ==> über Kondensator an DTR
- ESP32
  - RX ==> TX
  - TX ==> RX
  - EN ==> über Schaltung an DTR
  - IO\_0 ==> über Schaltung an RTS
  - IO\_13 ==> über Widerstand an RTS
  - IO\_15 ==> über Widerstand an CTS

Auch hier wurde vom Arduino Uno-Board abgeklupft und der selbe Chip verwendet, um die USB-Signale in UART-Signale zu konvertieren. Das vorkommende Bauteil ist der CP2102N. Da dieser Chip ein TQFP-28-Gehäuse hat, könnte es auch Schwierigkeiten geben beim Löten. Deswegen wird auch ein Breakout-Board (BOB) mitgeplant, damit es eine Ausweichmöglichkeit gibt falls die Bauteile zu klein sind. Dabei ist zu achten, dass das BOB genügend Ausgangspins hat, speziell beim ESP32.

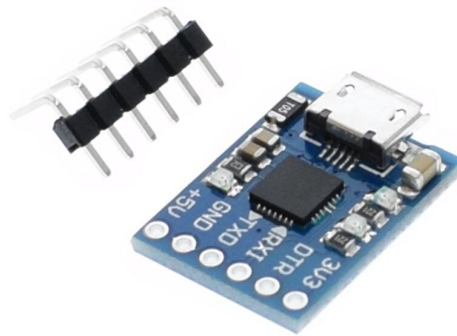


Abbildung 3.1: CP2102N-BOB für uC.

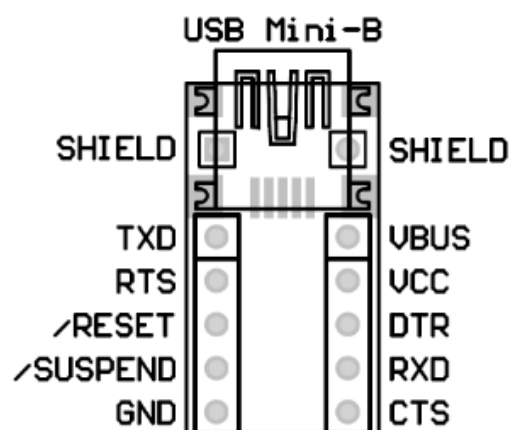


Abbildung 3.2: CP2102N-BOB für ESP.

## 3.2 Wirelessmodul

Über einen Web-Host soll der User die Möglichkeit haben, Getränke auszuwählen und seinem RFID-Chip zuzuordnen, sowie diverse kleinere Einstellungen an der Maschine vorzunehmen. Dazu wird ein WiFi-Modul benötigt.

Aufgrund schon bestehender Erfahrungen wurde ein Espressif ESP-Modul ausgewählt. Grundsätzlich standen zwei Modelle zur Auswahl. Das ESP8266 und das ESP32. Für die Cocktailmaschine wurde das ESP32 ausgewählt, da dies einfach Leistungsstärker ist. Die genauen Datenvergleiche sind in Tabelle 3.1 ersichtlich.

Wichtig ist, dass ein ESP ausgewählt wird, welches einen Anschluss für eine abgesetzte Antenne hat, da die Leiterplatte im Gehäuse verbaut wird. Dazu eignet sich der Espressif ESP32-32U. Dieser ist in Abbildung ?? als Wroom dargestellt und in Abbildung 3.4 als Development Kit.



MCU	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
802.11 b/g/n Wi-Fi	HT20	HT40
Bluetooth	No	Bluetooth 4.2 and BLE
Typical Frequency	80 MHz	160 MHz
SRAM	No	Yes
Flash	No	Yes
GPIO	17	36
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
Ethernet MAC Interface	No	Yes
Touch Sensor	No	Yes
Temperature Sensor	No	Yes
Hall effect sensor	No	Yes
Working Temperature	-40°C to 125°C	-40°C to 125°C
Price	\$ (3\$ - \$6)	\$\$ (\$6 - \$12)

Tabelle 3.1: Vergleich ESP8266 zu ESP32.

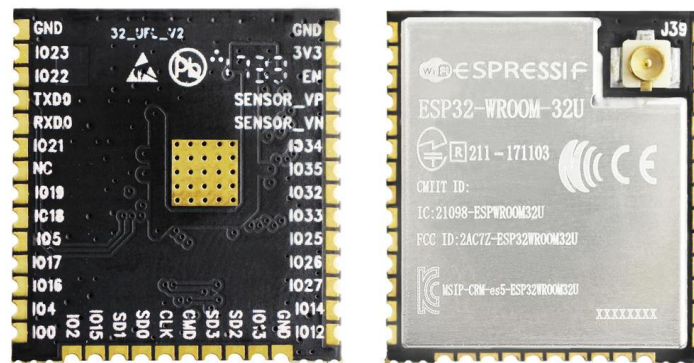


Abbildung 3.3: ESP32-32U Wroom.

### 3.3 RFID

Eine vorgesehene Funktion der Maschine ist, dass der User ohne Suchen sein Lieblingsgetränk zubereiten lassen kann. Dafür wird auf ein System zurückgegriffen, was öfters für Zutrittskontrollen oder Ähnliches verwendet wird. Nämlich RFID <sup>1</sup>.

Um schnell ein funktionierendes Teilsystem testen zu können, soll ein Breakout-Board verwendet werden. Eine Recherche hat ergeben, dass es nicht allzu viele Produkte gibt. Ein Auswahlkriterium war, dass es es für den Ausgewählten IC eine vorgeschriebene Library gibt, welche sich mit Arduino bewährt hat. Der Chip, welcher diese Anforderungen erfüllt, ist der **Mifare MFRC522**. Er ist in Abbildung 3.5 abgebildet. Dieses Brakeout-Board kann ausserdem an der Verschalung der Maschine angeschraubt werden. Somit kann er sich an einer Stelle befinden, welche weiter weg von der Hauptplatine ist.

<sup>1</sup>Radio Frequency Identification

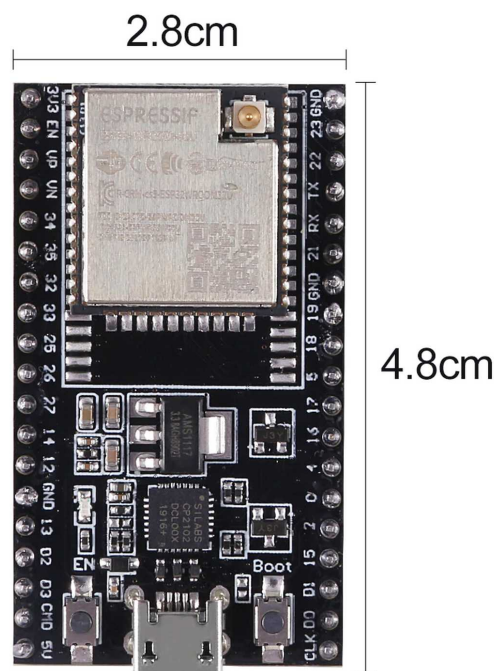


Abbildung 3.4: ESP32-32U DevKit.



Abbildung 3.5: MFRC522.

### 3.4 Beleuchtung

Damit die Maschine optisch etwas hergibt, wurde entschieden die ganze Maschinerie mit LED-Streifen zu verzieren. Dazu ist einerseits das LED-Band von nöten und die geeignete Ansterung dazu. Für die Cocktailmaschine wurde festgelegt, dass der LED-Streifen die benötigten Widerstände für die LED's schon aufgeklebt hat und die Ansteuerung folglich direkt über FET's geschehen kann. Der Aufbau ähnelt dann dem in Abbildung 3.6 gezeigten Schaltung. Der Streifen wäre Rot eingrahmt, die LED's und Widerstände befinden sich auf dem Band und die zu sehenden Fähnchen für R, G, B und W führen zum Mikrocontroller.

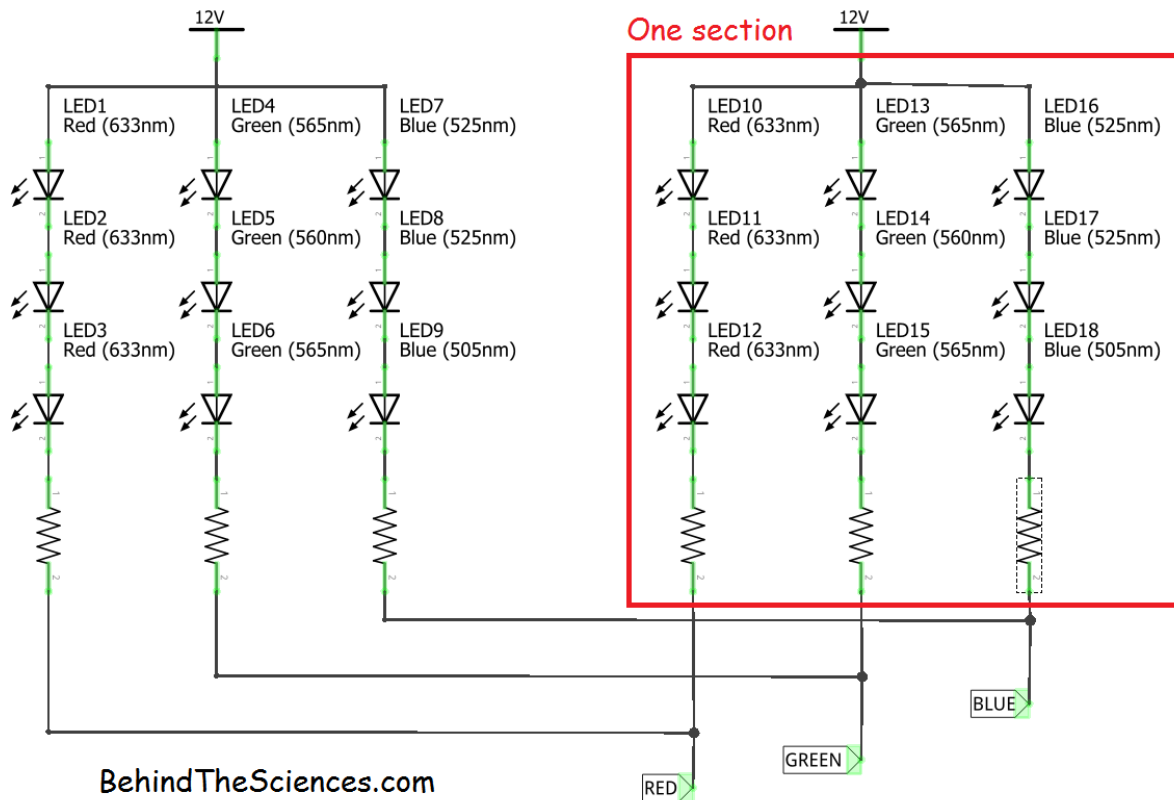


Abbildung 3.6: Hallo

## 4 Printaufbau

## 5 Teilsysteme

Da der Partymixer aus vielen kleineren und grösseren Teilsystemen besteht, werden diese in diesem Kapitel einzeln aufgelistet und im Detail angeschaut. Es wird dabei bei jedem Teilsystem auf drei Punkte eingegangen, die Problemstellung (Was ist der Zweck der Teilschaltung und weshalb wird sie benötigt?), das Schema und der Funktionsbeschrieb der Schaltung.

### 5.1 Speisungen

Ohne Speisung funktioniert keine elektronische Schaltung. Sie bildet daher einen essentiellen Bestandteil des Partymixer's. In dem System befinden sich vier verschiedene Speisungen. Die Ausgangsspannung für alle anderen Speisespannungen bildet dabei ein 48V Netzteil. Aus dieser Spannung wird mittels Step-Down Reglern eine 12V und eine 5V Speisung erzeugt. Bei der vierten Spannung handelt es sich um einen einfachen Linearregler, welcher aus den 5V eine 3.3V Speisung realisiert.

#### 5.1.1 48V Speisung

Der Motor wird mit einer Spannung von 48V betrieben. Dies ist zugleich auch die höchste verwendete Speisespannung. Um diese Speisung gewährleisten zu können, wird ein fertiges Netzteil gemäss **Fachbericht 5** eingesetzt.

Es wurde im Projekt 5 entschieden, dass die 48V Speisung extern als fertiges Netzteil eingekauft wird. Somit entfällt das Schema für diesen Speisungsteil. Ein Anschauungsbild des eingesetzten Netzteils kann in Abbildung 5.1 begutachtet werden.



Abbildung 5.1: Anschauungsbild des 48V Netzteils

Es musste jedoch unbedingt eine Leistungsabschätzung gemacht werden. Auch diese wurde im Projekt 5 durchgeführt. Unter Berücksichtigung der Schaltungsteile welche noch im Projekt 6 ergänzt werden, wurde dieses dann ausgewählt und eingekauft. Die Leistungsabschätzung kann im **Fachbericht 5** eingesehen werden.

### 5.1.2 12V Speisung

Die Pumpen werden mit 12V betrieben, was zur Folge hat, dass eine 12V Speisung implementiert werden musste. Dazu wird ein Schaltspannungsregler verwendet. Dieser wandelt mittels Step-Down Prinzip die 48V des Netzteils in eine Konstantspannungsquelle von 12V. Es handelt sich hierbei um einen Regler von Monolithic Power Systems. Genauer gesagt um den MP24943DN-LF. Die Auswahl ist auf dieses Bauteil gefallen, da mit 48V eine relativ hohe Eingangsspannung verarbeitet werden muss. Der MP24943DN-LF kann am Eingang mit Spannungen von 4.5-55V arbeiten und dabei eine Ausgangsspannung von 0.8-45V erzeugen. Dies bei einem maximalen Strom von bis zu 3A. Die Realisierung der 12V Speisung kann in Abbildung 5.2 betrachtet werden.

#### Schema

Das Schema in Abbildung 5.2 kann in fünf Teile unterteilt werden. Da wäre zuerst der Eingangsfilter, welcher mit C32, C34 & C36 realisiert ist. Dieser Eingangsfilter wird gefolgt von einem Spannungsteiler, welcher den Enable auf aktiv setzt. Der eigentliche Regler wird mittels des IC7, D6 & L3 realisiert. Mittels zweier Spannungsteiler, wird die gewünschte Ausgangsspannung, sowie die Övertoltage-Protection eingestellt. Vor dem Ausgang der Schaltung ist dann erneut eine Filterstufe implementiert, welche das Ausgangssignal glättet.

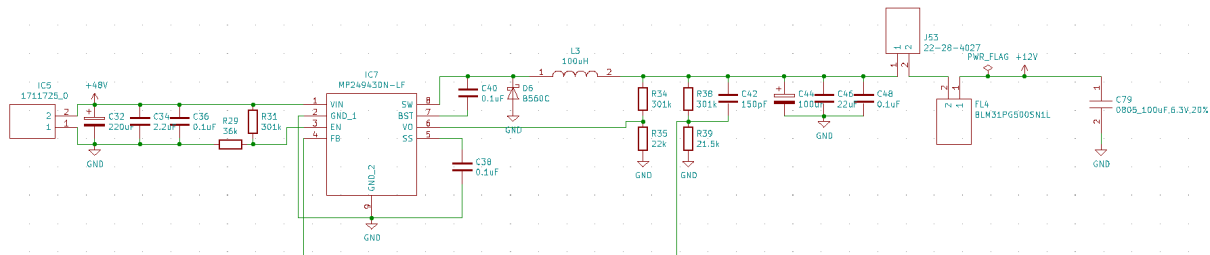


Abbildung 5.2: Schema der 12V Speisung

#### Funktionsbeschreibung der Schaltung

Um den MP24943DN-LF auf aktiv zu setzen, wird eine minimale Spannung von 1.8V vorausgesetzt. Fällt diese unter 0.4V, so wird dieser auf inaktiv gesetzt. Damit der Spannungsregler immer eingeschaltet ist, wird mittels zweier Widerstände R29 & R31 ein Spannungsteiler realisiert, welcher den Enable (EN) Pin auf 5V und somit auf aktiv setzt. Dieser Spannungsteiler musste implementiert werden, da alle Eingangspins ausser dem  $V_{in}$  einen maximale Eingangspegel von 6.5V verkraften können.

Die gewünschte Ausgangsspannung wird mittels Spannungsteiler R38 & R39 eingestellt, welche auf den Feedback Eingang (FB) rückgekoppelt werden. Diese berechnet sich laut Datenblatt gemäss Formel 5.1.

$$R39 = \frac{R38}{\frac{V_{out}}{0.8} - 1} \quad (5.1)$$

Bei einem Widerstandsverhältnis von  $R38=301k\Omega$  &  $R39=21.5k\Omega$  entspricht dies einer Ausgangsspannung von 12V.

Um einer Überspannung vorbeugen zu können, wird am Eingang Voltage-Overshoot (VO) ein Spannungsteiler implementiert. Diese wird am VO-Eingang mit einer Referenzspannung von 0.9V verglichen. Übersteigt die Spannung an VO die Referenzspannung von 0.9V, so wird der Regler ausgeschaltet, bis die Spannung wieder unter 0.9V fällt. Als maximale Ausgangsspannung wurde hierbei eine Spannung von 13V gewählt. Diese Wahl wurde getroffen, da die 12V ausschliesslich für die Ansteuerung der Pumpen verwendet wird und diese eine Spannung von 13V verkraften können ohne Schaden zu nehmen. Der Spannungsteiler wird gemäss Datenblatt mit der Formel 5.2 berechnet.

$$R35 = \frac{R34}{\frac{V_{ovp}}{V_{ovref}} - 1} \quad (5.2)$$

Bei einem Widerstandsverhältnis von  $R34=301k\Omega$  &  $R35=22k\Omega$  entspricht dies einer Überspannungsschutzschwelle von 13.21V.

Der Rippel des Spulenstroms lässt sich gemäss Formel 5.3 berechnen. Dieser sollte gemäss Datenblatt ca. 30% des maximalen Ausgangsstroms von 3A betragen.

$$L3 = \frac{V_{out} * (V_{in} - V_{out})}{V_{in} * \Delta I_L * f_{osc}} \quad (5.3)$$

Der interne Oszillator läuft dabei bei einer Frequenz von 100kHz. Bei der ausgewählten Spule von  $100\mu H$  erhalten wir ein  $\Delta I_L$  von 0.9A. Ausserdem wird im Datenblatt darauf hingewiesen, dass die gewählte Spule auf mindestens 125% des maximalen Ausgangsstroms von 3A ausgelegt werden soll. Auch der Gleichstromwiderstand der Spule sollte  $\leq 200m\Omega$  sein.

Mit den Kondensatoren C44, C46 & C48 wird die Ausgangsspannung zum Abschluss noch geglättet. Bei den Eingangskondensatoren, sowie den Ausgangskondensatoren sollte es sich um low ESR Typen handeln. Um die restlichen hochfrequenten Störungen herauszufiltern, ist zum Abschluss ein Ferrit implementiert worden.

### 5.1.3 5V Speisung

Der Mikrocontroller, sowie die Durchflussmessgeräte und das Display werden mit 5V betrieben. Aus diesem Grund wurde eine 5V Speisung implementiert. Dazu wird der selbe Schaltspannungsregler wie bei der 12V Speisung in Kapitel 5.1.2 verwendet. Die Realisierung der 5V Speisung kann in Abbildung 5.3 betrachtet werden.

## Schema

Das Schema in Abbildung 5.3 kann wie bei der 12V Speisung gemäss Kapitel 5.1.2 in fünf Teile unterteilt werden. Da wäre zuerst der Eingangsfilter, welcher mit C31, C33 & C35 realisiert ist. Dieser Eingangsfilter wird wiederum gefolgt von einem Spannungsteiler, welcher den Enable auf aktiv setzt. Der eigentliche Regler wird auch hier mittels des IC6, D5 & L4 realisiert. Mittels zweier Spannungsteiler, wird die gewünschte Ausgangsspannung, sowie die Überspannungseingestellt. Vor dem Ausgang der Schaltung ist dann erneut eine Filterstufe implementiert, welche das Ausgangssignal glättet.

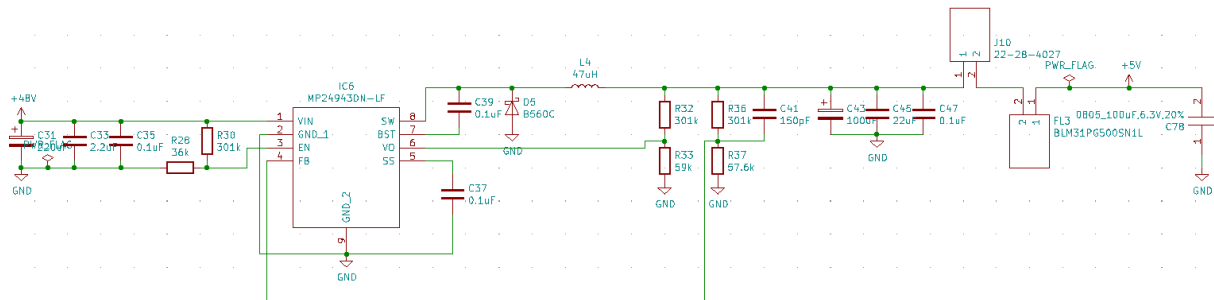


Abbildung 5.3: Schema der 5V Speisung

## Funktionsbeschreibung der Schaltung

Auch bei der 5V Speisung wurde mittels R29 & R31 ein Spannungsteiler realisiert, welcher das IC gemäss Kapitel 5.1.2 auf aktiv setzt.

Das Widerstandsverhältnis von R36 & R37, welches die Ausgangsspannung definiert, wurde gemäss Formel 5.1 berechnet. Somit ergeben sich für R36=301k $\Omega$  und für R37=57.6k $\Omega$ , Was einer Ausgangsspannung von 4.98V entspricht.

Beim Überspannungsschutz musste darauf geachtet werden, dass der Mikrokontroller AtMega2560-16AU nur in einem Spannungsbereich von 4.5V-5.5V betrieben werden darf. Die maximal verträgliche Eingangsspannung liegt laut Datenblatt bei 6V. Somit muss der Überspannungsschutz so gestaltet werden, dass die Schwelle von 6V nicht überschritten werden kann. Um dies erreichen zu können, wurde für R32=301k $\Omega$  und R33=53k $\Omega$  gewählt. Gemäss Formel 5.2 erhält man so eine Überspannungsschutzschwelle von 6V.

Der interne Oszillator läuft wiederum bei einer Frequenz von 100kHz. Bei der ausgewählten Spule von 47 $\mu$ H erhält man mittels Formel 5.3 ein  $\Delta I_L$  von 0.953A. Auch hier gilt gemäss Datenblatt, dass die gewählte Spule auf mindestens 125% des maximalen Ausgangsstroms von 3A ausgelegt werden soll. Auch der Gleichstromwiderstand der Spule sollte  $\leq 200$ m $\Omega$  sein.

Mit den Kondensatoren C43, C45 & C47 wird die Ausgangsspannung zum Abschluss auch noch geglättet. Bei den Eingangskondensatoren, sowie den Ausgangskondensatoren sollte es sich um low ESR Typen handeln. Auch hier wurde noch zum Abschluss ein Ferrit implementiert, welcher allfällige hochfrequente Störungen herausfiltern soll.

### 5.1.4 3.3V Speisung

Um die Treiber der Motorenansteuerung, das Wirelessmodul und die RFID-Schaltung betreiben zu können, wird zusätzlich eine 3,3V Speisung verbaut. Da es sich dabei nicht um enorm Leis-



tungstreibende Elemente handelt, wurde entschieden einen einfachen Linearregler einzusetzen, welcher von der 5V Speisung aus betrieben wird.

## Schema

Bei dem Linearregler handelt es sich konkret um den LF33CDT-TRY von STMicroelectronics. Dieser hat eine fixe Ausgangsspannung von 3.3V, bei einem maximalen Strom von 1A. Das dazugehörige Schema kann in Abbildung 5.4 begutachtet werden.

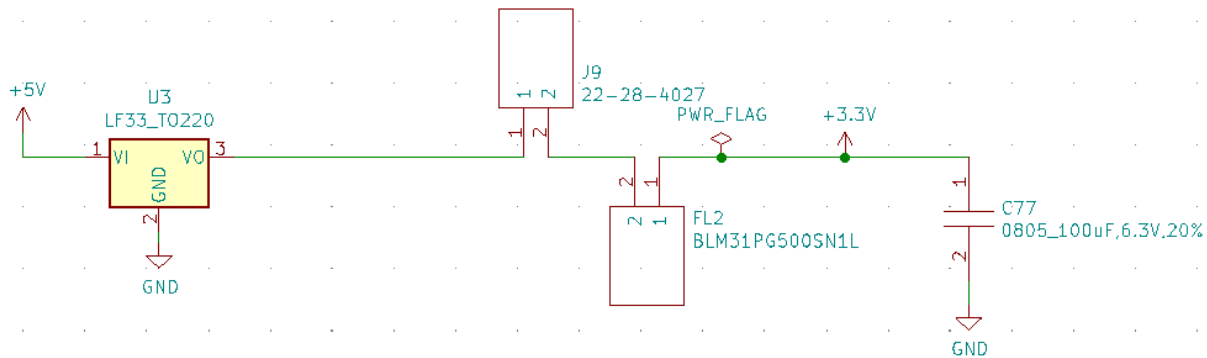


Abbildung 5.4: Schema der 3.3V Speisung

## Funktionsbeschreibung der Schaltung

Der Linearregler benötigt keine Spezielle Beschaltung. Er wird lediglich an die 5V Speisung angeschlossen. Am Eingang und am Ausgang ist ein Filterkondensator implementiert.

## 5.2 Motor

### 5.2.1 BLDC und H-Brücke

### 5.2.2 ABN-Encoder

### 5.2.3 Treiber

## 5.3 Flüssigkeitsbeförderung

### 5.3.1 Pumpen

### 5.3.2 Durchflussmessgeräte



## 5.4 Benutzerschnittstellen

### 5.4.1 Display

### 5.4.2 ESP

### 5.4.3 USB-C

### 5.4.4 RFID

Ein Bestandteil der Cocktailmaschine ist, mittels einem persönlichen kontaktlosen Chip oder Badge das Lieblingsgetränk auswählen zu können. Dazu wird auf die RFID<sup>2</sup>-Technologie gesetzt. Sie ermöglicht es mittels RFID-Tags oder auch mit NFC<sup>3</sup> des Handys Daten zwischen Tag und Maschine auszutauschen. So kann der User identifiziert werden.

Damit die Datenübertragung stattfinden kann, wird an der Antenne ein Wechselfeld angelegt, welches im Folgenden Energiesignal genannt wird. Dieses induziert beim Empfänger eine Spannung, welche als Energieversorgung dient. Der Sender moduliert ein Datensignal über das Energiesignal. Die Informationen werden im Tag demoduliert und verwertet. Bei den Befehlen für den Tag geht es hauptsächlich um Lese- und Schreibmethoden. Bei einer Lesemethode des RFID wird ein Teil des Speichers abgefragt, bei Schreibmethoden wird der Speicher beschrieben.

## Schaltungsaufbau

Bild 5.5 zeigt den Schaltungsaufbau des RFID-Moduls. Darin ersichtlich sind folgende Teilbereiche:

- MFRC522 RFID IC
- Antenne
- Anpassnetzwerk für Antenne
- Kommunikaitonsschnittstellen

## Funktionsbeschreibung der Schaltung

Der IC steuert den Ablauf, damit die Datenübertragung stattfinden kann. Er verbindet die Antenne mit dem System und wandelt die Informationen brauchbar von einer Seite auf die andere. Das Anpassnetzwerk sollte die gleiche Impedanz haben wie die Antenne. Wie die Bauteile dimensioniert werden, kann in einer Anleitung nachgesehen werden. Sie dazu Anleitung: `nxp_bv_2010_antenna_2010`.

Die Anschlüsse sind so gestaltet, dass falls eine einseitig gelayoutete Antenne verwendet wird, die Kommunikation zwischen UART und SPI gewählt werden kann. UART lässt sich einfacher in die Software einbinden, so die bisherige Erfahrung.

Für unser System wurde ein Breakout-Board verwendet. In erster Linie damit keine Zeit gebraucht wird, sich mit einem Antennendesign auseinandersetzen zu müssen. Aber auch, weil der IC in ein TQFP28-Gehäuse gepackt ist, und Schwierigkeiten bereiten könnte beim Löten.

---

<sup>2</sup>Radio-Frequency Identification

<sup>3</sup>Near-Field-Communication

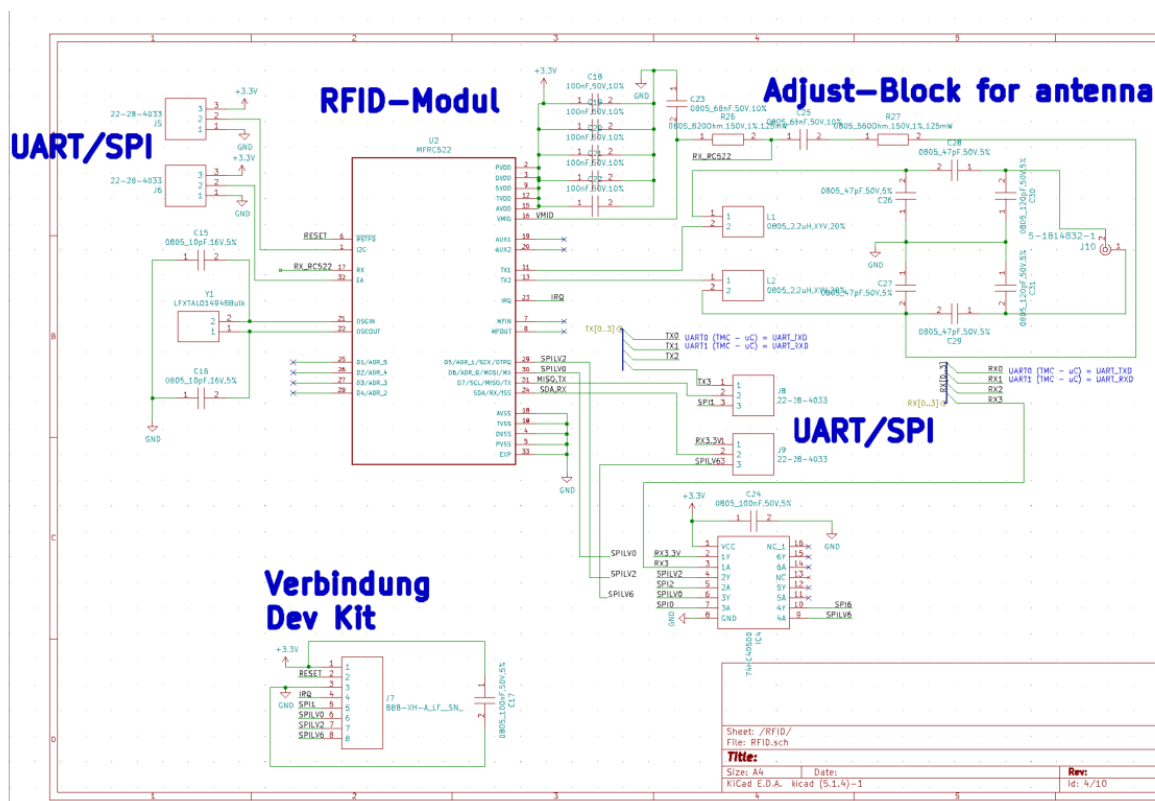


Abbildung 5.5: Schema des RFID Sender/Empfänger

## 5.5 Beleuchtung

Damit die Maschine ein richtiger Hingucker wird, müssen die Blicke auf sie gezogen werden. Dazu eignet sich ein LED-Band hervorragend, am besten wenn es noch verschiedenfarbig ist. Dazu wird eine geeignete Ansteuerung benötigt. Es wird davon ausgegangen, dass das LED-Band die benötigten Widerstände schon eingebaut hat und die Anschlüsse somit direkt an die FET's gehen können.

### Schaltungsangbau

Abbildung 5.6 zeigt den Schaltungsaufbau der LED-Steuerung. Damit die LED's angesteuert werden können, braucht es ein Bauteil, welches mit einer 5V-Ansteuerung 12V schalten können. Dazu wird ein MOSFET verwendet. Über die Widerstände an den Gates wird der Strom zum Schutz des Gates begrenzt. Die Leitungen führen direkt auf den Klemmblock für die LED-Streifen. Parallel dazu wurden noch für jede Lichtfarbe ein Kontroll-LED installiert, welche es ermöglicht auch ohne LED-Band etwas zu programmieren.

### Funktionsbeschreibung der Schaltung

Licht sind bekanntlich elektromagnetische Wellen im sichtbaren Wellenlängen-Bereich. Dabei gibt es drei oder vier Hauptfarben, Rot, Grün, Blau und je nach dem Weiss. Zwar kann Weiss auch aus einer Kombination aller drei Farben erstellt werden, kommt aber besser mit einer

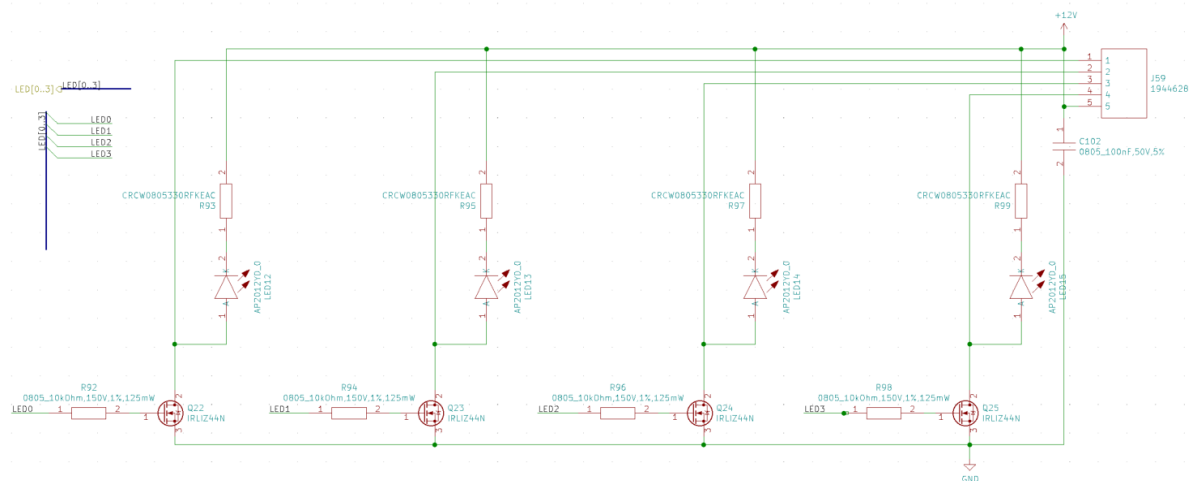


Abbildung 5.6: Schema der LED-Ansteuerung

separaten Diode. Das resultierende Licht des Bandes ist eine Überlagerung der Wellenlängen. Diese Überlagerung kann vom Mikrocontroller über den Duty-Cycle eines PWM-Signals gesteuert werden. So lassen sich mit dem Licht praktisch aus den Grundfarben praktisch alle Farben mischen.

## 5.6 Mikrocontroller

# 6 Inbetriebnahme

## 6.1 Speisungen

### 6.1.1 12V Speisung

### 6.1.2 5V Speisung

### 6.1.3 3.3V Speisung

## 6.2 Motor

### 6.2.1 BLDC und H-Brücke

### 6.2.2 ABN-Encoder

### 6.2.3 Treiber

## 6.3 Flüssigkeitsbeförderung

### 6.3.1 Pumpen

### 6.3.2 Durchflussmessgeräte

## 6.4 Benutzerschnittstellen

### 6.4.1 Touch-Display

### 6.4.2 ESP

## Problem

## Schaltungsaufbau

## Funktionsbeschreibung der Schaltung

### 6.4.3 USB-C

### 6.4.4 RFID

## 6.5 Beleuchtung

# 7 Software

**7.1 Strukturplan****7.2 Programmflussdiagramm****8 Evaluation****9 Fazit****9.1 Zielerreichung****9.2 Kosten****10 Schlusswort****11 Ehrlichkeitserklärung**

Mit der Unterschrift bestätigt der Unterzeichnende Projektleiter, dass die vorliegende Projektdokumentation selbstständig im Team und ohne Verwendung anderer, als der angegebenen Hilfsmittel verfasst wurde, sämtliche verwendeten Quellen erwähnt und die gängigen Zitierregeln eingehalten wurden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Unterschrift:

---

Ort, Datum:

---