

Automatisierung essentieller Bestandteile

Zentrale Steuereinheit



### Inhaltsverzeichnis

1.	Projekthalt und Projektziele .....	3
1.1.	Grundgedanke des Projekts .....	3
1.2.	Zielsetzung .....	3
2.	Konzept und Durchführung .....	4
2.1.	Konzept.....	4
	Folgende Bauteile werden benötigt: .....	4
	Die Komponenten wurde folgendermaßen angeschlossen: .....	5
	Ablauf/Ausführung des Systems: .....	5
2.2.	Vorgehen .....	6
2.3.	Durchführung .....	6
3.	Statusbericht .....	7
4.	Reflexion und Ausblick .....	7
	Ausblick .....	7

### 1. Projektinhalt und Projektziele

#### 1.1. Grundgedanke des Projekts

In diesem Projekt soll eine Mikrobrauanlage vollkommen automatisiert werden. Dabei sollen im Verlauf von vier Semester mit Sensoren, Pumpen und elektrischen Schaltungen die Prozessschritte kontrolliert und über eine grafische Oberfläche angesteuert werden. Während der Durchführung des Projektes sollen Optimierungsmöglichkeiten, bezüglich Kosten und Energieoptimierung, entwickelt werden.

#### 1.2. Zielsetzung

Der Aufbau der Mikrobrauanlage (siehe Abbildung 1: Schematischer Aufbau) soll aus einer Maischepfanne mit Rührwerk und einem elektrischen Kochfeld bestehen. Im ersten Semester des Projekts soll der Brauvorgang bis zum Abläutern automatisiert werden.

Als zentrale Steuereinheit wird ein Raspberry PI 3 (RPI) zum Einsatz kommen. Die Temperatur der Maische wird durch einen Sensor überwacht, dessen Wert der RPI regelmäßig ausliest. Per Servomotor wird die Leistung der Heizplatte hoch oder runter reguliert, so dass ein manueller Eingriff während des Einmischens nicht mehr nötig ist.

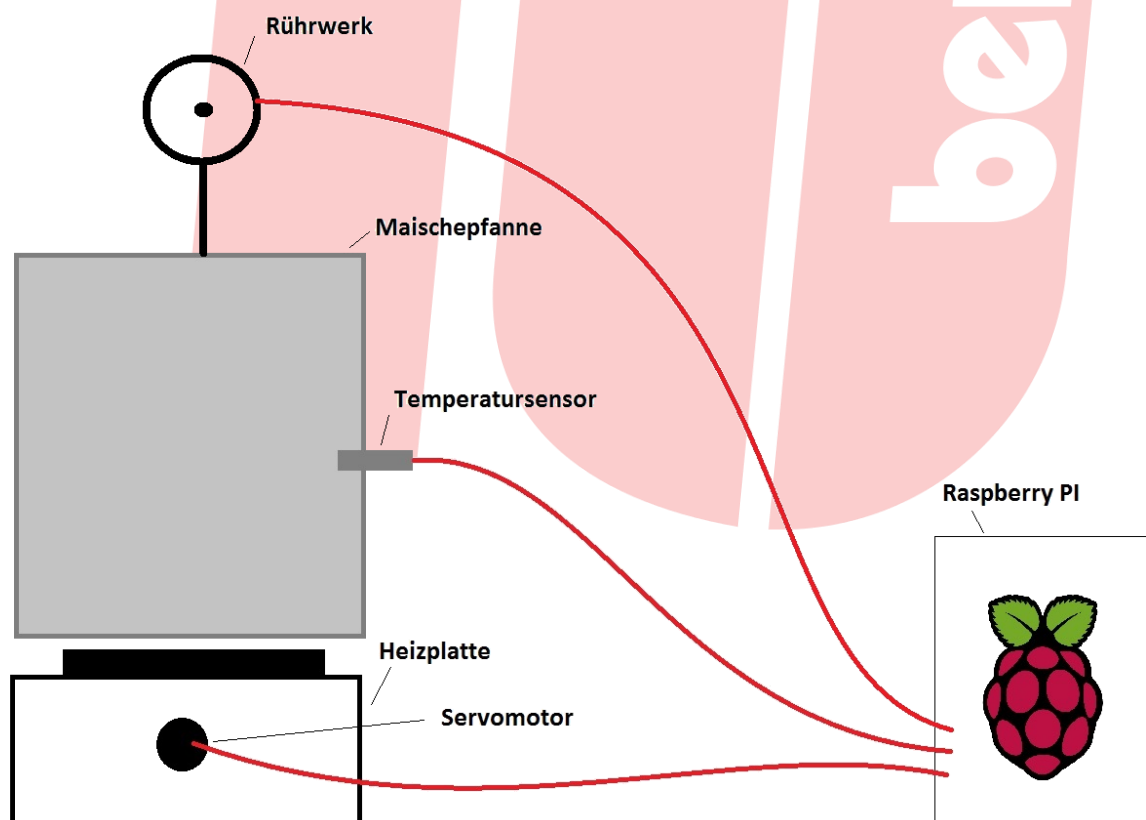


Abbildung 1: Schematischer Aufbau

## 2. Konzept und Durchführung

### 2.1. Konzept

Der RPI soll die Steuerung des Einmischens übernehmen. Die Software zur Steuerung wird aus zwei Teilen bestehen - einem Frontend und einem Backend. Das Backend wird in Python geschrieben und kontrolliert die angeschlossene Peripherie.

Folgende Bauteile werden benötigt:

#### 1. Temperatursensor DS18B20



Abbildung 2: Sensor DS18B20 <sup>1</sup>

Der Sensor wird in die Maischepfanne eingebaut, sodass die aktuelle Temperatur ausgelesen werden kann. Eine detaillierte Beschreibung ist dem Datenblatt zu entnehmen

#### 2. Servomotor SG90 9g



Abbildung 3: Motor SG90 9g <sup>2</sup>

Der Servomotor wird an die Regulierung der Heizplatte montiert und steuert so deren Leistung. Eine detaillierte Beschreibung ist dem Datenblatt zu entnehmen

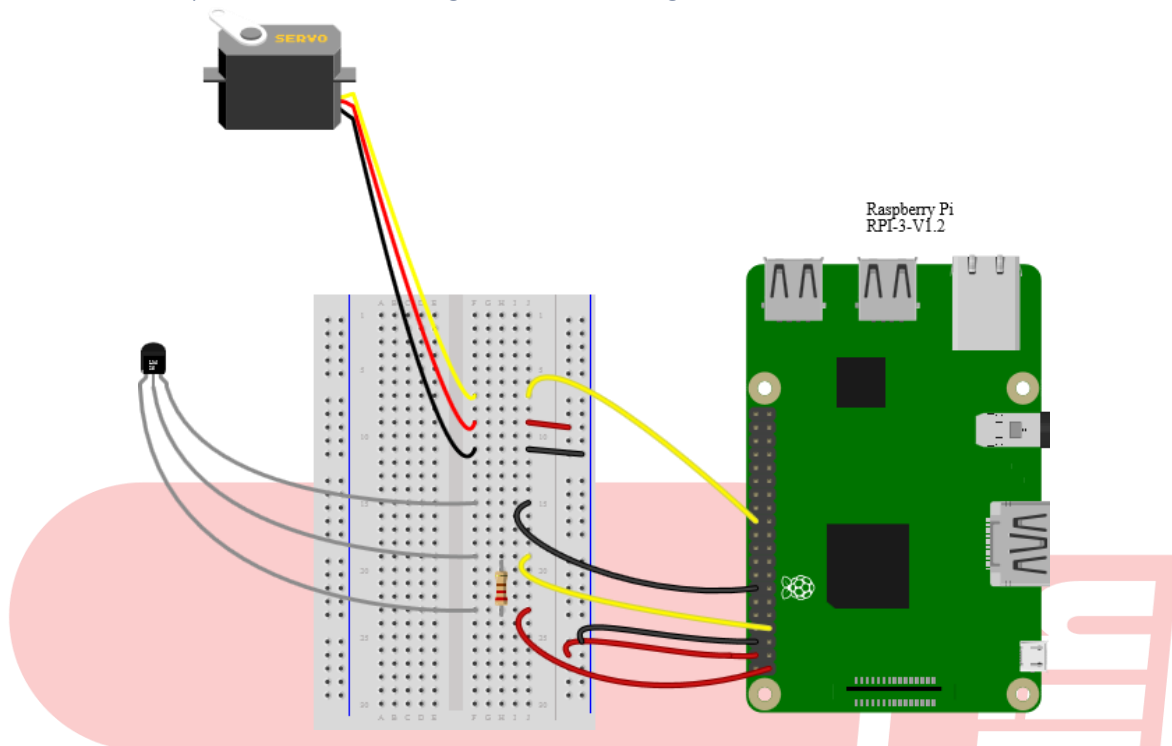
<sup>1</sup> <https://www.roboter-bausatz.de/246/sg90-9g-micro-servomotor>

<sup>2</sup> [https://www.makerlab-electronics.com/my\\_uploads/2016/08/waterproof-temperature-sensor-ds18b20-1.jpg](https://www.makerlab-electronics.com/my_uploads/2016/08/waterproof-temperature-sensor-ds18b20-1.jpg)

# Anlagen- und Prozessautomatisierung einer Mikrobrauanlage

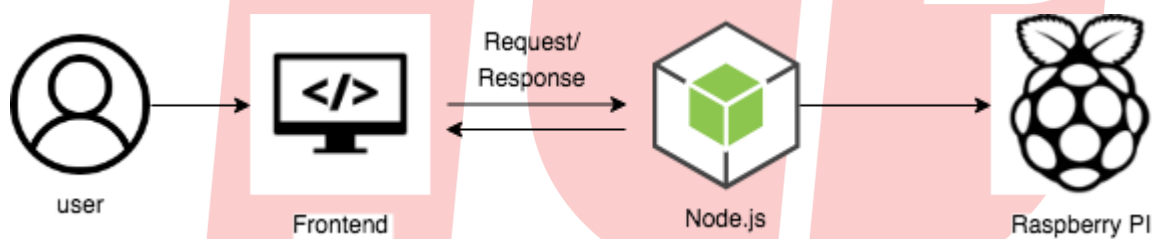
## Automatisierung essentieller Bestandteile

Die Komponenten wurde folgendermaßen angeschlossen:



Hierbei ist zu beachten, dass zwischen die Datenleitung und die Spannungsversorgung des Temperatursensors noch ein 4.7 kΩ Widerstand geschaltet werden muss.

Ablauf/Ausführung des Systems:



Der Benutzer kann über das Frontend Rezepte erstellen. Diese werden lokal gespeichert und können über das Frontend wieder abgerufen werden. Es ermöglicht dem Benutzer außerdem das starten des Systems und somit die zuvor vorgestellten Komponenten. Dafür wird dem Backend ein zuvor erstelltes Rezept übergeben, wo zum Beispiel die Zeiten und Temperaturen für das Brauen bestimmt wurden.

Nach dem Start eines Brauvorgangs, soll es möglich sein den Aktuellen Status zu sehen. Das heißt, in welchem Schritt befindet sich der Vorgang zurzeit, welche Temperatur hat der Kessel im Moment und wie lange braucht er noch für diesen Schritt. Dies ermöglicht dem Benutzer eine bessere Kontrolle über den Vorgang, um gegebenenfalls eingreifen zu können.

### 2.2. Vorgehen

Für die Versionierung wird ein Github Projekt benutzt. Die Kommunikation läuft hauptsächlich über private Chats da wir nur eine 2-Mann-Gruppe sind. Das Konzept wurde am Anfang des Semesters ausgearbeitet und im Laufe des Projekts angepasst.

### 2.3. Durchführung

Die Vorarbeiten um den RPI als Steuereinheit nutzen zu können, werden hier nicht näher im Detail betrachtet. Im wesentlichen Bestehen diese aus der Installation und Konfiguration des Betriebssystems (hier: Raspbian).

Die einzelnen Komponenten (Temperatursensor, Servomotor und Rührwerk) werden durch ein in Python geschriebenes Programm gesteuert. Der Programmstart erfolgt über das Web-Frontend. Hierbei werden auch die Eingabeparameter übergeben.

Der Programmstart nach dem Auswählen eines Rezepts und dem Drücken eines Buttons im Frontend über Node.js mit dieser Zeile Code ausgeführt:

```
exec('python ../brew.py ' + <Name des Rezepts>);
```

Dadurch wird das Programm brew.py gestartet und der Name des ausgewählten Rezepts übergeben.

Die Eingabeparameter werden in einem JSON Format übergeben. Dies ist der Aufbau der JSON-Datei:

```
{
  "recipe": "",
  "Step": [
    {
      "id": "",
      "name": "",
      "target_temp": "",
      "duration": ""
    }
  ]
}
```

Im Frontend können flexibel viele Schritte dem Rezept hinzugefügt werden, diese werden dynamisch in dem Array „Step“ gespeichert.

### 3. Statusbericht

Im Moment ist es möglich über das Frontend ein Rezept zu erstellen, welches lokal auf dem Raspberry Pi gespeichert wird und welches für den Brauvorgang ausgewählt werden kann. Außerdem kann der Brauvorgang aus dem Frontend gestartet werden und somit wird das ausgewählte Rezept an das Raspberry Pi übergeben. Nach dem Starten des Programms werden mit den Variablen des Rezepts (Zeit, Temperatur) der Brauvorgang durchgeführt.

### 4. Reflexion und Ausblick

#### Ausblick:

Durch die begrenzte Zeit konnten wir leider nicht alles umsetzen, im folgenden Abschnitt werden diese offenen Punkte kurz beschrieben.

Im Frontend sind noch viele Verbesserungen möglich. Das Design haben wir nicht weiterbearbeitet und es hat sicher Optimierungspotential.

Rezepte können im Moment noch nicht bearbeitet werden, dadurch müsste der Benutzer für eine kleine Anpassungen, kein neues Rezept erstellen.

Die Komponenten einzeln über das Frontend anzusprechen würde es dem Benutzer ermöglichen sofort in dem Brauvorgang Justierungen vorzunehmen.

Die im Konzept schon beschriebene Statusanzeige wurde von uns noch nicht fertig implementiert. Der Status wird aufgeteilt in den aktuellen Status und den generellen Status.

Der generelle Status soll in einem Graphen angezeigt werden (siehe Abbildung 4). Die Gradzahl auf der Y-Achse und die Zeit auf der X-Achse. Im Moment benutzen wir im Frontend noch mock-Daten, die Statusdaten sollten jedoch vom Raspberry Pi in eine Log-Datei gespeichert werden und über Node.js ausgelesen und im Graphen angezeigt werden.

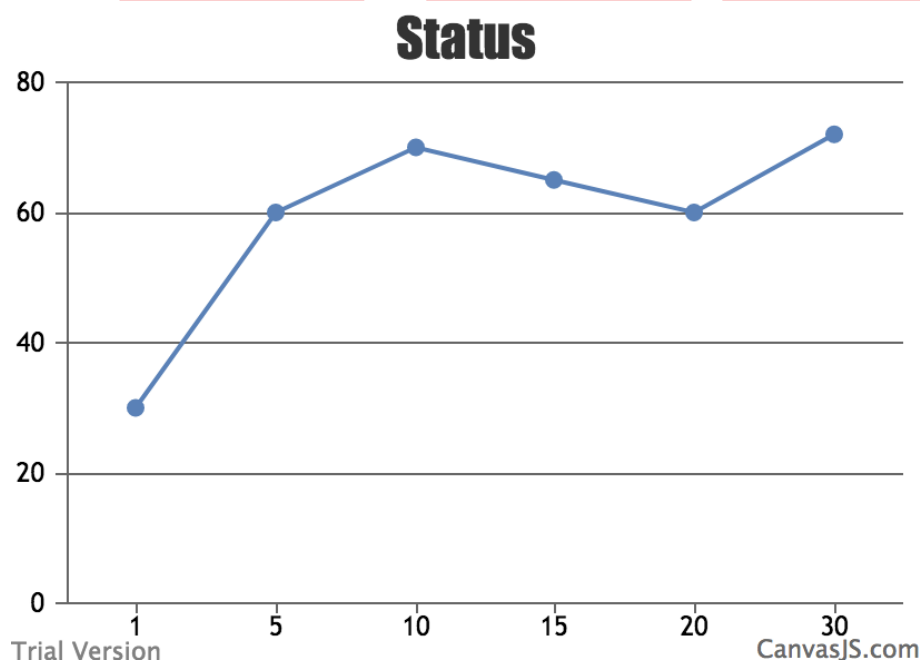


Abbildung 4 Status