

Audit 2

# Entwicklung eines Systems zur Überwachung und Automatisierung von Aquaponikanlagen

---

Anne Germund und Verena Heissbach





## Feedback aus Audit 1

---

- Wird es eine DIY-Variante geben?
- Was gibt es für Alternativen zum Raspberry Pi?
- Wie können wir die Überwachung überwachen? Wie können wir feststellen, ob die Werte korrekt sind?
- Welche Sicherheitsvorkehrungen werden getroffen?
- Wie und wo werden die Daten gespeichert?
- Wird eine API bereitgestellt, über welche die Messdaten abgerufen werden können?

## Risiken und abgeleitete Proof-of-Concepts





# Identifizierte Risiken

---

## Technische Risiken

- Schwierigkeiten bei der Installation des Raspberry Pi
- Die Spezifikation (z.B. Messauflösung) der verwendeten Hardware passt nicht zueinander
- Überflutungsgefahr bei fehlerhaften Messwerten oder Fehlinterpretation
- Fehlerhafte Datenspeicherung
- Feststellung eines kritischen Wasserpegels kompliziert, da der Wasserstand aufgrund des Ebbe-Flut-Beckens oszilliert. Den Hinweis auf einen fehlerhaften Kreislauf liefert also nicht nur der aktuelle Wasserstand, sondern auch die fehlende periodische Veränderung des Wasserpegels

## Risiken in der Kommunikation/Interaktion der Anwendungsobjekte

- Der Raspberry Pi empfängt keine Daten vom Temperatursensor PT100
- Der Raspberry Pi empfängt keine Daten vom Wasserstandssensor
- Die Ansteuerung der Relais funktioniert nicht
- Keine Benachrichtigung bei kritischen Messwerten oder Aktivitäten des Systems

## Kompetenzbezogene Risiken

- Fehlendes Basiswissen in den Bereichen Elektrik/Elektronik, Datenaufnahme, -analyse, -aufbereitung, Messlogik
- Kurzschluss und resultierende Schäden an Hardware
- Geplante Realisation in vorgegebener Zeit ggf. schwierig → rechtzeitig einschreiten, falls Erfüllung des Zeitplans kritisch erscheinen sollte



# Proof-of-Concept #1

---

## Erste Implementierungen und Installationen am Raspberry Pi

### Beschreibung

Um den Raspberry Pi für das Projekt nutzen zu können, muss dieser in mehreren Schritten für die Arbeit vorbereitet werden. Die Schritte orientieren sich an folgendem Tutorial: <https://projects.raspberrypi.org/de-DE/projects/raspberry-pi-getting-started/0>

### Schritte

- Betriebssystem auf SD-Karte installieren
- Raspberry Pi mit Strom versorgen
- Netzwerkkonfiguration und SSH-Verbindung einrichten
- Updates installieren
- benötigte Bibliotheken für den Betrieb der Sensoren installieren (SpiDev, Adafruits)

### Exit Kriterien

Die beschriebenen Schritte konnten erfolgreich ausgeführt werden und der Raspberry Pi ist für den weiteren Projektverlauf vorbereitet.

### Fail Kriterien

- Probleme bei der Stromversorgung
- Installationen nicht erfolgreich
- Keine Netzwerkverbindung möglich

### Fallbacks

Zurücksetzen des Raspberry Pi, dann eine erneute Durchführung der Schritte, ggf. andere Tutorials, Betriebssysteme und Bibliotheken nutzen.



## Proof-of-Concept #2

---

### Messdaten des Temperatursensors PT 100 am Raspberry Pi empfangen

#### Beschreibung

Der Temperatursensor PT 100 sowie der AD-Wandler sind an den Raspberry Pi angeschlossen und die Messdatenübertragung funktioniert.

#### Schritte

- Installation des AD-Wandlers auf dem Breadboard
- Verbindung der Komponenten über Jumper Kabel
- Aktivierung des SPI-Bus am Raspberry Pi
- Auslesen der Daten über Python-Skript → Kalibrierung notwendig, da digitale Daten ohne Umrechnung nicht sinnvoll verarbeitet werden können

#### Exit Kriterien

Es können Messdaten vom Temperatursensor PT 100 ausgelesen werden.

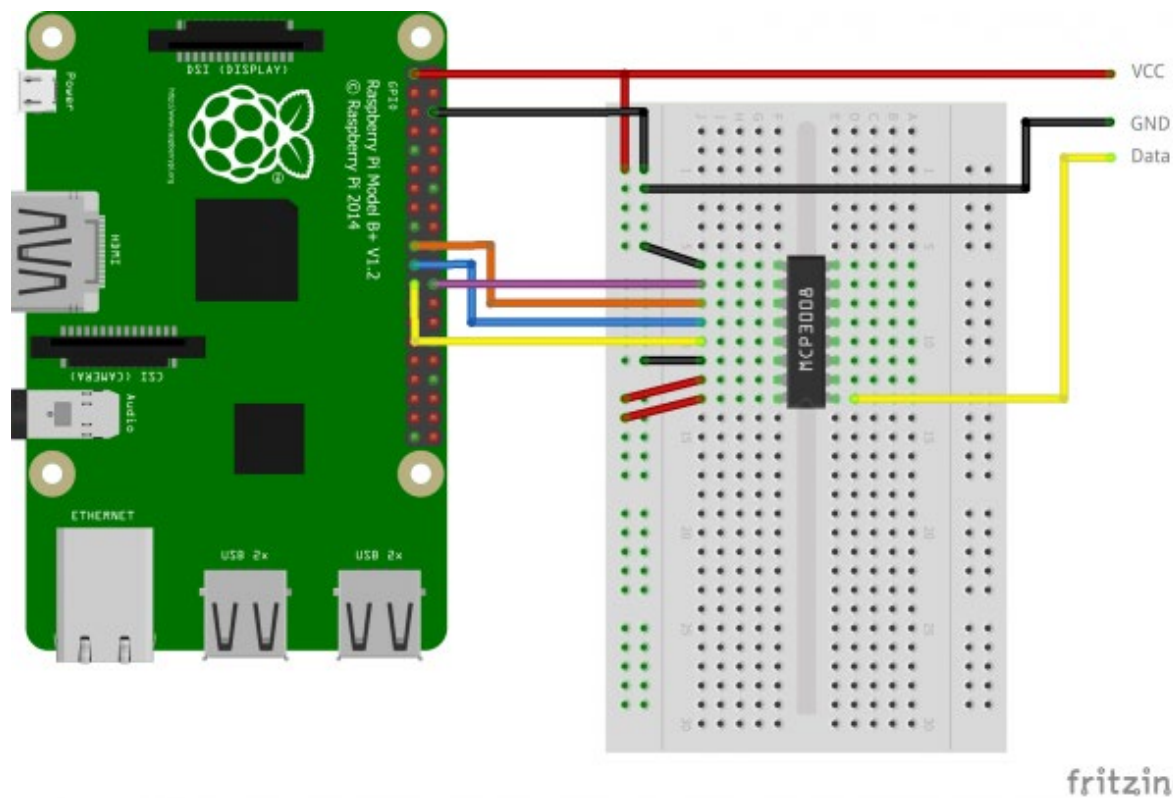
#### Fail Kriterien

- Der Sensor misst keine Veränderung von Werten
- Der Raspberry Pi empfängt keine Daten
- Die Umwandlung der analogen Daten ist fehlerhaft

#### Fallbacks

Sollten die Sensoren keine Werte messen können, sollten sie erneut kalibriert und evtl. neu installiert werden. Ist es nicht möglich eine Verknüpfung zwischen den Sensoren und der Schnittstelle herzustellen, soll mit Dummy Daten gearbeitet werden. Ggf. hilft eine neue Installation und Konfigurierung des Raspberry Pi.

## Proof-of-Concept #2 | Durchführung



Schematische Darstellung der Verbindung zwischen Raspberry Pi und AD-Wandler MCP3008.  
Quelle: <https://tutorials-raspberrypi.de/raspberry-pi-mcp3008-analoge-signale-auslesen/>

### Messdaten des Temperatursensors PT 100 am Raspberry Pi empfangen

Für die Installation des AD-Wandlers wurde dieses Tutorial befolgt: <https://tutorials-raspberrypi.de/raspberry-pi-mcp3008-analoge-signale-auslesen/>

Die Abbildung zeigt den Aufbau der Verbindung zwischen Raspberry Pi und dem AD-Wandler MCP3008. Nicht dargestellt ist der Temperatursensor, der an das hier gelb gefärbte, rechte Kabel angeschlossen ist.

Zunächst wurde zur Durchführung der SPI-Bus aktiviert und anschließend die SpiDev-Bibliothek installiert. Mithilfe einer eigenen Python-Klasse für den MCP3008 konnten die analogen Kanäle erfolgreich ausgelesen werden. Hierbei konnte festgestellt werden, dass der AD-Wandler digitale Werte von 0 bis 4095 ausgibt, was sich in der Formel zum Auslesen der Werte niederschlägt:

$$(\text{Digitaler Wert} / 4095) * 3,3 \text{ V}$$





## Proof-of-Concept #3

---

### Messdaten des Wasserstandssensors am Raspberry Pi empfangen

#### Beschreibung

Über den Raspberry Pi können die Wasserstandsdaten des zugehörigen Sensors ausgelesen werden.

#### Schritte

- Verbindung der Komponenten über Jumper Kabel
- Auslesen der Daten über Python-Skript

#### Exit Kriterien

Es können Messdaten vom Wasserstandssensor ausgelesen werden.

#### Fail Kriterien

- Der Sensor misst keine Veränderung des Wasserstands
- Der Raspberry Pi empfängt keine Daten

#### Fallbacks

Sollte der Wasserstandssensor keine Werte messen können, sollte die Verbindung der Jumper-Kabel überprüft werden. Ist es nicht möglich eine Verknüpfung zwischen den Sensoren und dem Raspberry Pi herzustellen, muss mit Dummy Daten gearbeitet werden. Ggf. hilft eine neue Installation und Konfigurierung des Raspberry Pi.





## Proof-of-Concept #4

---

### Kompatibilität der Hardwarekomponenten

#### Beschreibung

Die Spezifikation der gewählten Hardware sollte zueinander passen, um eine korrekte Datenübertragung bzw. Interaktion der Architekturkomponenten zu ermöglichen.

#### Schritte

Die gewählte Hardware liefert für den Raspberry Pi auswertbare Daten.

#### Exit Kriterien

Die Schnittstelle kann erfolgreich mit der gegebenen Hardware kommunizieren und die gelieferten Daten sinnvoll verarbeiten.

#### Fail Kriterien

Die Spannungsunterschiede der Sensoren sind für die Hardware nicht messbar.

#### Fallbacks

Einsatz von alternativer Sensorik, z.B. rein digitale Sensoren.



## Proof-of-Concept #5

---

### Ansprechen der Relais (Aktor)

#### Beschreibung

Über die GPIO-Pins des Raspberry Pi ist eine direkte Ansteuerung der Relais möglich. Die Relais dienen zum Schalten der größeren Verbraucher (z.B. Magnetventil).

#### Schritte

- Der Raspberry Pi empfängt Wasserstandsmesswerte, die die untere Grenze unterschreiten
- Der Raspberry Pi schaltet das Relais, welches das Magnetventil öffnet
- Bei Erreichen des gewünschten Pegels wird das Magnetventil geschlossen

#### Exit Kriterien

Der Wasserstand konnte erfolgreich angepasst werden und befindet sich wieder im gesetzten Bereich.

#### Fail Kriterien

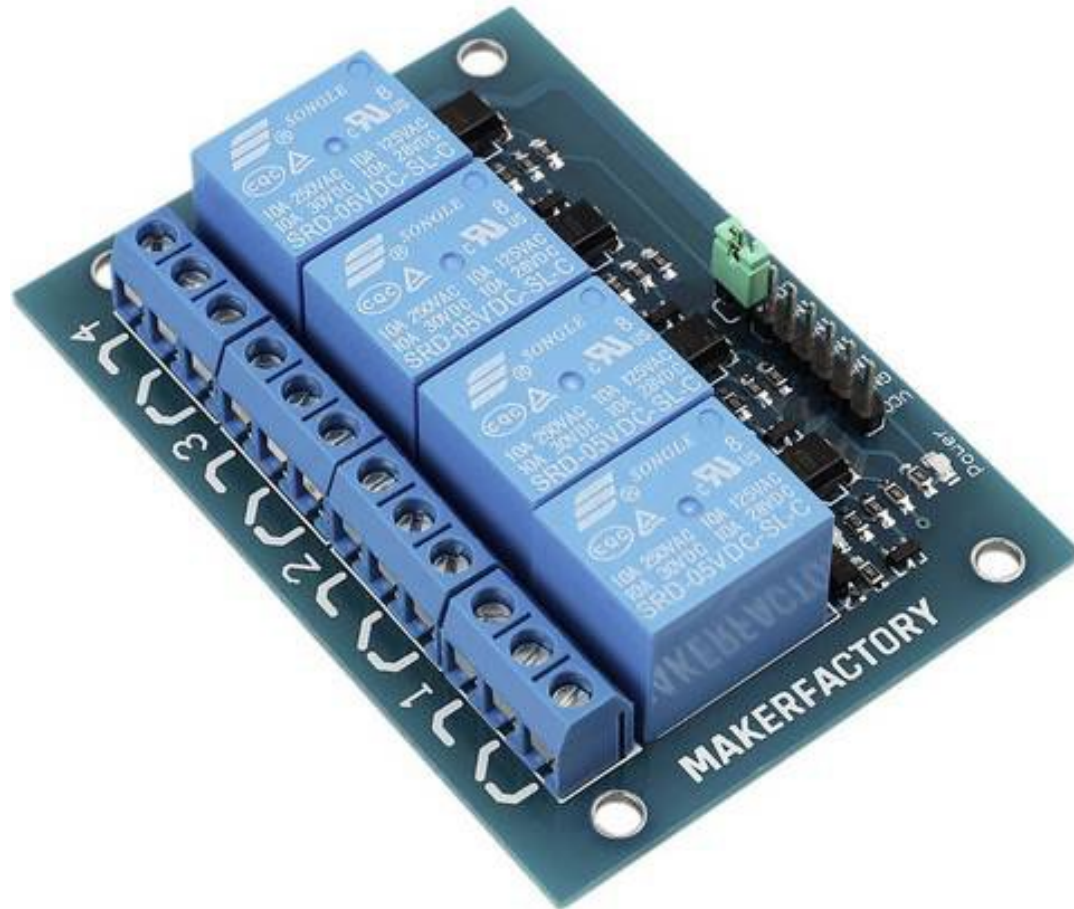
- Falsche Messwerte, was zu einer falschen Aktivierung führen könnte
- Relais lassen sich nicht aktivieren bzw. deaktivieren → Überflutungsgefahr oder Trockenlauf
- Magnetventil öffnet sich nicht oder schließt nicht mehr

#### Fallbacks

Ein mechanisches Absperrventil kann im Falle eines Ausfalls vor Überschwemmung schützen. Verbindungen zwischen Raspberry Pi, Relais und Ventil neu installieren. Ggf. muss ein neues Ventil angeschafft werden.

## Proof-of-Concept #5 | Durchführung

---



Verwendetes 4-Kanal Relais-Modul.

Quelle: <https://www.conrad.de/de/p/makerfactory-mf-6402393-relais-modul-1-st-2134131.html>

### Ansprechen der Relais (Aktor)

Um die Verbindung zwischen Raspberry Pi und Relais zu testen, wurde dieses Tutorial befolgt: <https://tutorials-raspberrypi.de/raspberry-pi-relais-schalter-steuern/>

Mithilfe von zwei Jumper Kabeln wurde die Relais-Platine zunächst mit einer Spannung von 3,3 Volt versorgt. Anschließend wurde das Relais 1 mit einem GPIO-Pin des Raspberry Pi verbunden und über ein Python-Skript aktiviert und deaktiviert.

Das Magnetventil wurde bisher noch nicht getestet und soll in einem nächsten Schritt angeschlossen werden.



## Proof-of-Concept #6

---

### Senden von Push-Benachrichtigungen über Pushbullet

#### Beschreibung

Das System soll den Nutzer bei kritischen oder fehlerhaften Werten sowie bei Auffüllen des Wasserstands via Pushbullet benachrichtigen.

#### Schritte

- Es wird ein Wert ermittelt, welcher sich außerhalb des zugelassenen Wertebereichs befindet
- Der Nutzer wird via Pushbullet über dieses Ereignis benachrichtigt
- In Sonderfällen (z.B. zu niedriger Wasserstand) werden die Relais geschaltet, welche die Wasserzufuhr steuern  
→ auch hierzu wird dem Nutzer eine Push Benachrichtigung gesendet

#### Exit Kriterien

Die Benachrichtigung erreicht den Nutzer korrekt und unmittelbar.

#### Fail Kriterien

- Der Raspberry Pi kann keine Nachricht senden
- Die Nachricht kommt nicht beim Empfänger an
- Fehlerhafte Benachrichtigung, falscher Inhalt
- Der Push Dienst funktioniert nicht
- Fehlerhafte Einbindung des Push Dienstes

#### Fallbacks

Erneute Installation von Pushbullet oder einen anderen Push Anbieter verwenden. Falls auch hier Schwierigkeiten auftreten, sollte auf andere Benachrichtigungsmechanismen zurückgegriffen werden (z.B. Telegram oder E-Mail).



## Proof-of-Concept #7

---

### Datenspeicherung in lokaler Datenbank

#### Beschreibung

Die aufgezeichneten Messdaten werden in einer lokalen MySQL-Datenbank abgelegt, damit diese z.B. auf einer Website ausgegeben werden können.

#### Schritte

- MySQL auf dem Raspberry Pi installieren
- Erstellen der Datenbank und der Tabellen für jeden Sensor
- Auslesen der Sensoren und entsprechender Eintrag in der Datenbank

#### Exit Kriterien

Die Daten wurden in die Datenbank geschrieben und können per SQL-Query ausgelesen werden.

#### Fail Kriterien

- Installation der MySQL-Datenbank nicht erfolgreich
- Eintrag in Datenbank nicht möglich

#### Fallbacks

Alternative Formate zur Datenspeicherung betrachten (z.B. InfluxDB). Ggf. muss auf externe Anbieter zur Online-Datenspeicherung zurückgegriffen werden (z.B. ThingSpeak).

# Proof-of-Concept #8

---

## Visualisierung der Messdaten auf einer eigenen Website

### Beschreibung

Die aufgezeichneten Messdaten aller Sensoren werden als Graphen auf einer Website im lokalen Netz zugänglich gemacht. Das Hosting übernimmt ein Webserver, der auf dem Raspberry Pi installiert ist.

### Schritte

- Webserver auf Raspberry Pi einrichten
- Website erstellen (PHP, CSS, JS)
- Einbindung der Daten aus der MySQL-Datenbank

### Exit Kriterien

Über den Aufruf der lokalen IP-Adresse oder dem Namen des Raspberry Pi kann im lokalen Netz die Messdaten-Website angezeigt werden.

### Fail Kriterien

- Webserver-Installation nicht erfolgreich
- Schwierigkeiten bei der Erstellung der Website
- Daten werden nicht korrekt angezeigt

### Fallbacks

Ausweichen auf externe Anbieter zur Datenvisualisierung (z.B. ThingSpeak, Grafana). Alternativ: Ausgabe der Messdaten nur auf dem Display des Raspberry Pi.

## Projektplan & Deliverables für Audit 3





# Gantt-Diagramm zur Projektplanung

## Entwicklung eines Systems zur Überwachung von Aquaponikanlagen

