

Use Case #1

use case Grafana-Daten per Fernzugriff einsehen

actors Aquaponikbetreiber

trigger Der Aquaponikbetreiber möchte die Daten auf dem Grafana-Dashboard per Fernzugriff einsehen.

precondition Verbindung zu lokalem Netz besteht UND Überwachung ist aktiv

main flow

1. Der Betreiber gibt in der Browsersuchleiste die IP-Adresse des Raspberry Pi sowie den zu Grafana gehörigen Port ein.
2. Der Betreiber loggt sich mit seinen Zugangsdaten bei Grafana ein.
3. Der Betreiber wählt das anzuzeigende Dashboard aus.

postcondition Der Betreiber kann die aufgezeichneten Sensordaten einsehen.

exceptional flow

- 2.a Der Betreiber kennt seine Login-Daten nicht.
- 3.a Es werden keine aktuellen Daten angezeigt.

postcondition Kein Einsehen der Daten möglich

end Grafana-Daten per Fernzugriff einsehen

Use Case #2

use Case Daten lokal über Display einsehen

actors Aquaponikbetreiber

trigger Der Aquaponikbetreiber möchte die vom System gemessenen Daten, lokal über ein Display einsehen.

precondition Es sind bereits Messungen vorhanden (Überwachung ist aktiv), das Display ist am Raspberry Pi angeschlossen und mit Grafana verknüpft. Zusätzlicher Grafana User ist vorhanden und eingeloggt, Dashboards wurden erstellt.

main flow

1. Der Aquaponikbetreiber geht zum Display
2. Er kann nun die gemessenen Daten über das Display sehen

alternative flow

- 2.a Der Betreiber sieht die Daten Remote ein (siehe Use Case Grafana-Daten per Fernzugriff einsehen).
- 2.b Messdaten über die Kommandozeile des Raspberry Pi anzeigen lassen.

postcondition Der Aquaponikbetreiber konnte die gewünschten Daten einsehen.

exceptional flow

- 2.b Das Display zeigt keine Daten bzw. die falschen Daten an.

postcondition Es können keine Daten eingesehen werden

end Daten lokal über Display einsehen

Use Case #3

use case Daten vom Wasserstandssensor abrufen

actors Raspberry Pi

trigger Das Skript zur Wasserstandskontrolle wird ausgeführt.

precondition Stromversorgung besteht UND Wasserstandssensor ist angeschlossen

main flow

1. Der Raspberry Pi liest über den Kanal des AD-Wandlers, an welchen der Sensor angeschlossen ist, zehnmal die digitalen Wasserstandsdaten aus und speichert sie in einer Variable.
2. Der Raspberry Pi berechnet den durchschnittlichen Wasserstandswert, indem der Variablenwert durch die Anzahl der Messungen geteilt wird.
3. Der Raspberry Pi legt den ermittelten Wert in der Influx-Datenbank ab (include „Daten in InfluxDB ablegen“).
4. Der Raspberry Pi prüft den ermittelten Wert auf die festgelegten Wasserstandsgrenzen.

postcondition Der aktuelle Wasserstand wurde in der Datenbank abgelegt.

exceptional flow Kein Datenabruf

- 1.a Der Raspberry Pi kann keine Daten auslesen.

exceptional flow Wasserstand zu niedrig

- 4.a Sollte der Wasserstand zu niedrig sein, startet der Raspberry Pi die Befüllung (extension point: Wasserstand zu niedrig).
- 4.b Der Raspberry Pi sendet eine Benachrichtigung über den zu niedrigen Wasserstand sowie die Befüllung an den Betreiber (extension point: Benachrichtigung).

postcondition Der Wasserstand wurde korrigiert und eine Benachrichtigung an den Betreiber der Anlage gesendet.

end Daten vom Wasserstandssensor abrufen

Use Case #4

use case Daten vom Temperatursensor abrufen

actors Raspberry Pi

trigger Das Skript zur Temperaturkontrolle wird ausgeführt.

precondition Stromversorgung besteht UND Temperatursensor ist angeschlossen

main flow

1. Der Temperatursensor führt je 10 Messungen aus (diese 10 Messungen ergeben dann einen Temperaturwert). Diese werden an den Raspberry Pi gesendet.
2. Der Raspberry Pi berechnet die durchschnittliche Temperatur, indem der Variablenwert durch die Anzahl der Messungen geteilt wird.
3. Der Raspberry Pi legt den ermittelten Wert in der Influx-Datenbank ab (include „Daten in InfluxDB ablegen“).
4. Der Raspberry prüft den Messwert auf die definierten Grenzen.

postcondition Die aktuelle Temperatur wurde in der Datenbank abgelegt.

exceptional flow

- 1.a Der Temperatursensor konnte keine Messungen durchführen.
- 1.b Es konnten keine Messwerte an den Raspberry Pi nicht gesendet werden.
- 3.a Die Daten konnten nicht in der Datenbank gespeichert werden.
- 4.a Sollte der Messwert außerhalb der definierten Grenzen liegen wird der Nutzer über Pushbullet darüber benachrichtigt (extension point: Benachrichtigung)

postcondition Es können keine Messungen abgerufen werden

end Daten vom Temperatursensor abrufen

Use Case #5

use case Daten in InfluxDB ablegen

actors Raspberry Pi

trigger Es wurden aktuelle Sensordaten abgerufen.

precondition InfluxDB ist installiert UND es existiert eine Datenbank für die Aquaponik-Sensordaten

main flow

1. Der Raspberry Pi erstellt den Datenpunkt mit den zugehörigen Messwerten und Sensortyp im JSON-Format.
2. Der Raspberry Pi übergibt die Sensordaten im JSON-Format an den InfluxDB-Client.

postcondition Der Datenpunkt konnte erfolgreich in die Influx-Datenbank abgelegt werden.

exceptional flow Kein Zugang zur Datenbank möglich

- 2.a Im Skript sind die falschen Zugangsdaten zur Datenbank hinterlegt.

postcondition Der Datenpunkt konnte nicht in der Influx-Datenbank abgelegt werden.

end Daten in influxDB ablegen

Use Case #6

use case Visualisierung der Messdaten mittels Grafana

actors Raspberry Pi

trigger Grafana kann auf die gespeicherten Messdaten in der Datenbank zugreifen.

precondition Grafana ist installiert und mit der Datenbank verknüpft. Es befinden sich Messdaten in der Datenbank. Der Betreiber ist als User bei Grafana angemeldet und eingeloggt.

main flow

1. Grafana greift auf die Datenbank der Aquaponiknalage zu.
2. Der Betreiber wählt das anzuzeigende Dashboard aus.
3. Die Messdaten werden im gewählten Dashboard dargestellt.

postcondition Der Betreiber kann die aufgezeichneten Sensordaten einsehen.

alternative flow

- 3.a Messdaten über die Kommandozeile des Raspberry Pi anzeigen lassen.

postcondition Die Daten konnten dargestellt werden.

exceptional flow

- 1.a Grafana kann nicht auf die Messwerte der Datenbank zugreifen.
- 2.a Es wurde kein Dashboard zur Visualisierung ausgewählt.
- 3.b Es werden keine aktuellen Daten angezeigt.

postcondition Keine Darstellung der Daten mittels Grafana möglich.

end Visualisierung der Messdaten mittels Grafana

Use Case #7

use case Benachrichtigung über Pushbullet senden

actors Raspberry Pi

trigger Der Raspberry Pi hat einen kritischen Sensorwert festgestellt und hat ggf. das Magnetventil geöffnet.

precondition Pushbullet ist installiert UND der API-Key ist hinterlegt UND das Gerät des Betreibers ist Pushbullet registriert

main flow

1. Der Raspberry Pi sendet über die Pushbullet-API eine Benachrichtigung über den kritischen Temperaturwert an das Gerät des Betreibers.

alternative flow

- 1.a Der Raspberry Pi sendet über die Pushbullet-API eine Benachrichtigung über den kritischen Wasserstand und das Öffnen des Magnetventils.
- 1.b Sobald der Wasserstand wieder im Normalbereich ist, sendet der Raspberry Pi eine Entwarnung an das Gerät des Betreibers.

postcondition Der Betreiber erhält eine Benachrichtigung über die kritischen Messwerte sowie mögliche Aktivitäten des Systems.

exceptional flow

- 1.a Das Gerät kann nicht gefunden werden.

postcondition Es konnte keine Benachrichtigung gesendet werden.

end Benachrichtigung über Pushbullet senden

Use Case #8

use case Magnetventil öffnen/schließen

actors Raspberry Pi

trigger Es wurde ein zu niedriger Wasserstand festgestellt und der Raspberry Pi leitet die Befüllung ein.

precondition Relais sind angeschlossen

main flow

1. Der Raspberry Pi versorgt den GPIO-Pin, über welchen das Relais-Modul angeschlossen ist, mit Spannung, um das Ventil zu öffnen.

alternative flow

- 1a. Der Raspberry Pi entfernt die Spannung vom GPIO-Pin, über welchen das Relais-Modul angeschlossen ist, um das Ventil zu schließen.

postcondition Das Magnetventil wurde erfolgreich geöffnet bzw. geschlossen.

exceptional flow

- 1.a Das Relais schaltet nicht.

postcondition Das Magnetventil kann nicht geöffnet bzw. geschlossen werden.

end Magnetventil öffnen/schließen