IoT Projekt „Kikeriki“

Paul Kramer

BZTG-Oldenburg

1 Inhaltsverzeichnis

2 Projektbeschreibung

Das Projekt Kikeriki ist eine automatisch gesteuerte Hühnerklappe, die abhängig von der gemessenen Helligkeit oder der aktuellen Uhrzeit, geöffnet und geschlossen wird. Es wird zusätzlich mithilfe von zwei RFID-Lesern gezählt, wie viele Hühner sich im Stall bzw. außerhalb des Stalles befinden. Die Werte des Hühnerzählers werden sowohl über eine Webanwendung visualisiert als auch auf einem Display angezeigt.

Die Webanwendung läuft über Node-Red und soll die Möglichkeit geben, die Klappe auch über einen Handbetrieb zu steuern. Außerdem können die Uhrzeiten, die Helligkeitsgrenzen, der Fahrbetrieb (nach Lux oder Uhrzeit), das Warten auf alle Hühner vor dem Schließen der Klappe und die Anzahl der Hühner für die Automatiksteuerung manuell angepasst werden. Die vom Microcontroller ausgewerteten Daten, wie die aktuelle Helligkeit in Lux, die sich im und außerhalb des Stalls befindenden Hühner, der Klappenzustand, sowie der aktuelle Verbindungsstatus der Anlage, werden ebenfalls auf der Weboberfläche angezeigt.

Die vom Microcontroller ausgewerteten Daten werden zusätzlich auf einer Datenbank langfristig gespeichert.

3 Ziele

Die Anlage soll im täglichen Betrieb weiter getestet und nach Kundenwunsch angepasst werden. Außerdem wird an einer selbst anpassbaren Offline-Version gearbeitet.

Die Anlage soll nicht mehr mit einer Batterie betrieben, sondern mit einem Stecker am Netz angeschlossen werden.

Es wird noch nach einer besseren Möglichkeit gesucht, die RFID-Chips auszulesen, da dies momentan nur eingeschränkt möglich ist (siehe Fazit). Es ist auch ein kompletter Ersatz für das RFID-System denkbar.

4 Hardwarekomponenten

Komponente	Modell/Typ	Funktion
Mikrocontroller	ESP32-S3-Devkitc-1	Verarbeiten des Steuerprogrammes
Sensor 1	BH1750	Messen der Helligkeit in Lux
Sensor 2 und 3	2x RDM6300	RFID-Leser
Aktor 1	28BYJ-48 Schrittmotor	Auf- und Zufahren der Klappe
Aktor 1	ULN2003 Stepper Motor Driver Board Module	Ansteuern des Schrittmotors
Aktor 2	ST7789	Anzeige der Hühneranzahl im Stall
Stromversorgung	9V Batterieblock	Stellt die Stromversorgung von 9V
Stromversorgung	Anschlusskabel für einen 9V Batterieblock	Ermöglicht einen sicheren Anschluss des 9V Batterieblocks
Netzteil	Abwärtswandler-Modul 9V DC zu 5V DC	Wandelt die 9V Eingangsspannung zu den benötigten 5V Ausgangsspannung um
Winde mit Seil	Winde mit Seil	Bewegt die Klappe
Verdrahtungszeug	Kabel	Verbindet die einzelnen Hardwarekomponenten

5 Softwarekomponenten

Komponente	Technologie	Funktion
Microcontroller-Code	Micro Python	Umsetzung des Steuerprogramms
Webinterface	Node-Red	User-Interface für einen Überblick der Hühner und der Steuerung der Klappe
Broker	MQTT	Ermöglicht das Publishen und Subscriben von Daten
Datenbank	MariaDB	Speicherung der Anlagendaten

6 Quellcode

Der gesamte Quellcode ist unter folgendem GitHub-Link zu finden:
https://github.com/xQund/Projekt-Kikeriki/tree/main
Folgende Dateien müssen vorhanden sein:

- bh1750.py
- bh1750_helligkeitssensor.py
- boot.py
- main.py
- projekt_kikeriki.py
- rdm6300_rfid.py
- st7789.py
- st7789_display.py
- uln2003_stepper.py
- vga1_16x32.py
- vga1_8x8.py
- wlan.py
- Node-Red flows.json

7 Technische Besonderheiten

Die Anlage kann auch im Offlinebetrieb arbeiten, hierbei befindet sie sich dauerhaft im Automatikbetrieb und fährt nach der Helligkeitsgrenze von 200 Lux. Die voreingestellte Anzahl an Hühnern ist 7 und es wird auf die Hühner vor dem Schließen gewartet.

Wenn ein Huhn im Bereich um die Klappe erkannt wird, durch die RFID-Leser, wird das Schließen der Klappe für den Zeitpunkt pausiert.

Die Klappe kann sowohl im Handbetrieb als auch im Automatikbetrieb betrieben werden.

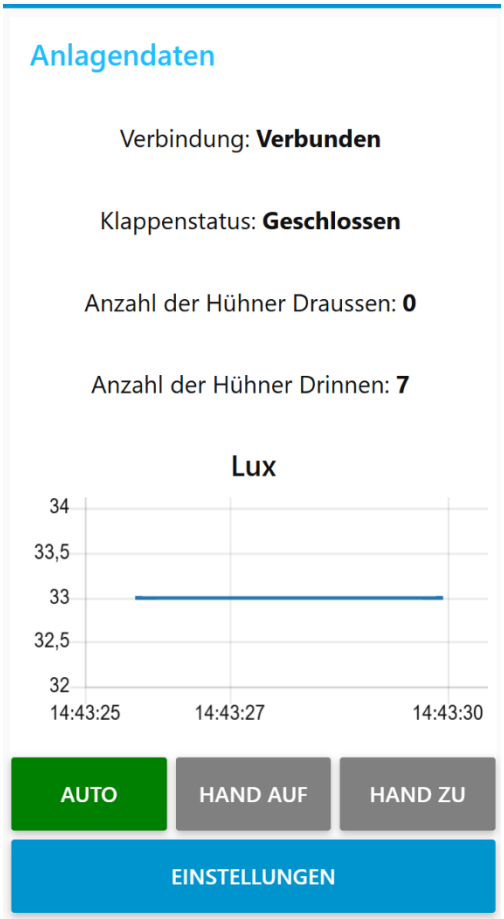
Die Einstellungen für den Automatikbetrieb können über die Weboberfläche angepasst werden.

Die vom Microcontroller aufbereiteten Daten, werden im Onlinebetrieb auf der Weboberfläche in Echtzeit angezeigt und in einer Datenbank gespeichert.

Es werden im Betrieb 1,9Wh, bzw. 380mAh, verbraucht, ein durchschnittlicher 9V Batterieblock kann den Betrieb für maximal 3 Stunden aufrechterhalten (für bessere Lösungen siehe Fazit).

8 Node-Red

8.1 User-Interface

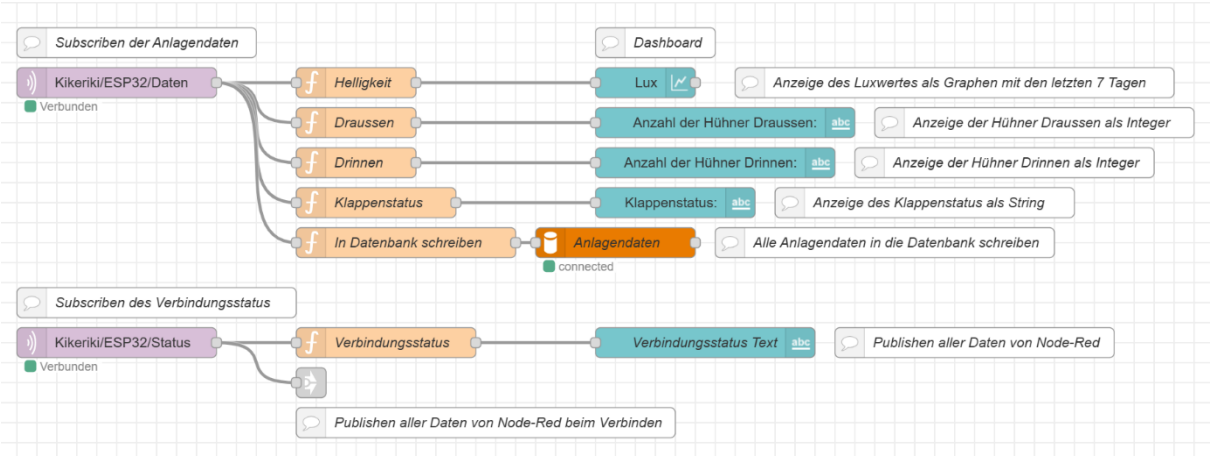


Das User-Interface unterteilt sich in zwei Reiter, einmal den Reiter „Anlagendaten“ in dem die von der Anlage kommenden Daten visualisiert werden und die Automatik-/Handsteuerung umgeschaltet werden kann. Um zu dem zweiten Reiter „Einstellungen“ zu gelangen, ist ein Knopfdruck auf „Einstellungen“ erforderlich. Hier können die Bedingungen für den Automatikbetrieb angepasst werden.

Für eine genauere Bedienbeschreibung siehe bitte Kapitel „12.1 Weboberfläche Node-Red“.

8.2 Flows

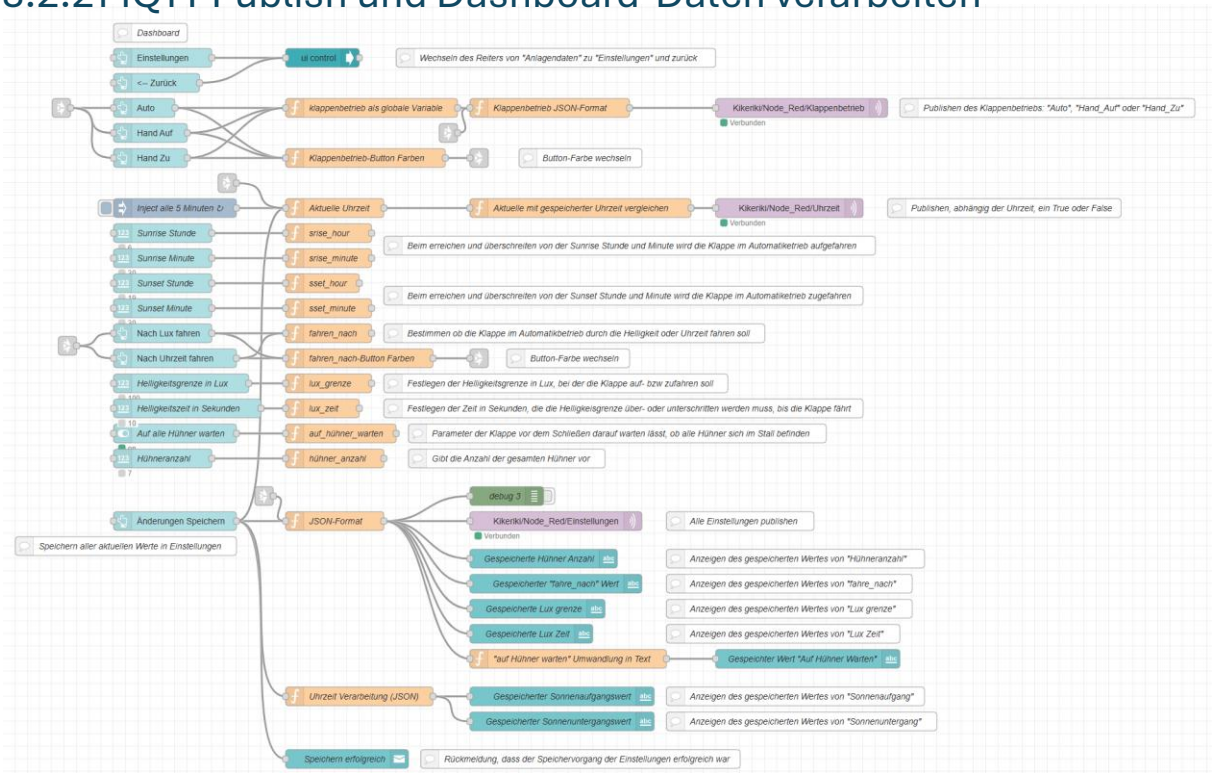
8.2.1 MQTT Subscriben und Anlagendaten weiterverarbeiten



In dem ersten Flow wird das Subscriben der vom Microcontroller kommenden Daten und die Visualisierung dieser Daten realisiert. Außerdem werden diese in der Datenbank gespeichert.

Die von der Anlage kommenden Daten teilen sich in zwei Topics auf, das erste Topic ist „Kikeriki/ESP32/Daten“ in dem alle vom Microcontroller erfassten Daten gepublisht werden und in dem Topic „Kikeriki/ESP32/Status“ wird der Verbindungsstatus des Microcontrollers, mit Hilfe eines Last Will Befehls, überwacht.

8.2.2 MQTT Publish und Dashboard-Daten verarbeiten



In dem zweiten Flow werden die von dem User über das User-Interface eingestellten Daten in dem Reiter „Einstellungen“, mit dem Betätigen des Knopfes „Änderungen Speichern“, über das Topic „Kikeriki/Node_Red/Einstellungen“ gepublisht.

Außerdem wird hier der gesamte Reiter „Einstellungen“ visualisiert, sowie die Automatik-/Handsteuerung und der Knopf „Einstellungen“ unter dem Reiter „Anlagendaten“.

Die Meldung in welcher Betriebsart die Anlage fahren soll, wird über das Topic „Kikeriki/Node_Red/Klappenbetrieb“ gepublisht, sobald einer der drei Knöpfe „AUTO“, „HAND AUF“ oder „HAND ZU“ betätigt wird.

Zuletzt werden die von dem User eingestellten Uhrzeiten für den Sonnenaufgang, sowie Sonnenuntergang mit der aktuellen Uhrzeit verglichen. Dabei wird geschaut, ob die aktuelle Uhrzeit zwischen den beiden eingestellten Uhrzeiten liegt oder außerhalb. Sollte sie dazwischen liegen, wird die Klappe aufgefahren, und außerhalb zugefahren. Dieser Datenwert wird alle 5 Minuten oder beim Betätigen des Knopfes „Änderungen Speichern“, über das Topic „Kikeriki/Node_Red/Uhrzeit“ gepublisht.

Alle Topics werden außerdem mit ihren dazugehörigen Daten gepublisht, wenn der Microcontroller eine Statusmeldung publisht.

9 MQTT

Als Broker wird MQTT verwendet und alle Daten werden im JSON-Format gepublisht.

9.1 Topic „Kikeriki/ESP32/Daten“

Publisher: Microcontroller
Daten:

"**Helligkeit**" = Helligkeitswert in Lux
"**Klappenstatus**" = Aktueller Öffnungsstatus der Klappe
"**Draussen**" = Anzahl der Hühner außerhalb des Stalls
"**Drinnen**" = Anzahl der Hühner im Stall

9.2 Topic „Kikeriki/ESP32/Status“

Publisher: Microcontroller

Daten:

" **Status**" = Verbindungsstatus des Microcontrollers

9.3 Topic „Kikeriki/ Node_Red/Einstellungen“

Publisher: Node-Red

Daten:

"**lux_grenze**" = Vom User eingestellte Helligkeitsgrenze in Lux

"**lux_zeit**" = Vom User eingestellte Helligkeitszeit in Sekunden

"**auf_huehner_warten**" = Warten, bis alle Hühner im Stall sind?

"**fahre_nach**" = Vom User eingestellte fahr Bedingung

"**huehner_anzahl**" = Vom User eingestellte Hühneranzahl

9.4 Topic „Kikeriki/ Node_Red/Klappenbetrieb“

Publisher: Node-Red

Daten:

"**klappenbetrieb**" = Betriebsart Hand oder Automatik

9.5 Topic „Kikeriki/ Node_Red/Uhrzeit“

Publisher: Node-Red

Daten:

"**uhrzeit**" = Ist die jetzige Uhrzeit zwischen den eingestellten Uhrzeiten

10 Datenbank

Als Datenbank wird MariaDB verwendet, in der die Anlagendaten über Node-Red gespeichert werden.

ID	Zeit	Helligkeit	Anzahl_Draussen	Anzahl_Drinnen	Klappenstatus
643	2025-05-04 15:06:52	54	0	7	Geschlossen
644	2025-05-04 15:06:52	53	0	7	Geschlossen
645	2025-05-04 15:06:52	53	0	7	Geschlossen
646	2025-05-04 15:08:22	53	0	7	Geschlossen
647	2025-05-04 15:08:23	53	0	7	Geschlossen
648	2025-05-04 15:08:23	53	0	7	Geschlossen
649	2025-05-04 15:08:24	53	0	7	Geschlossen
650	2025-05-04 15:08:24	53	0	7	Geschlossen
651	2025-05-04 15:08:25	53	0	7	Geschlossen

ID: Fortlaufender Identifikationsschlüssel

Zeit: Datum und Uhrzeit des Publishens

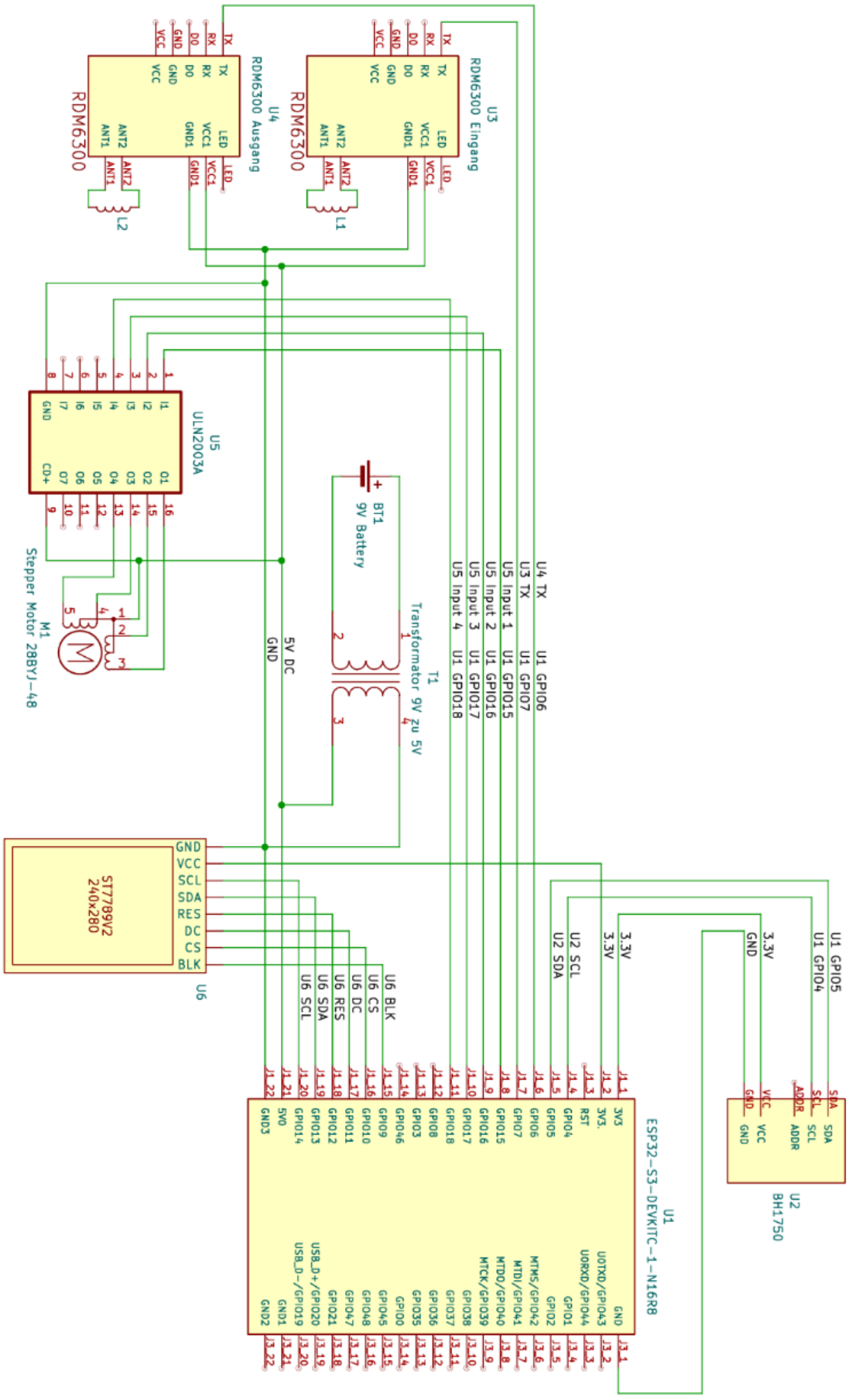
Helligkeit: Gemessene Helligkeit in Lux

Anzahl_Draussen: Anzahl der Hühner außerhalb des Stalls

Anzahl_Drinnen: Anzahl der sich im Stall befindenden Hühner

Klappenstatus: Öffnungsstatus der Klappe

11 Schaltplan



12 Pin-Belegungsplan

Microcontroller (ESP32-S3-Wroom-1-N16R8) [U1]:

GPIO 4 = [U2] SCL
GPIO 5 = [U2] SDA
GPIO 6 = [U4] RX
GPIO 7 = [U3] RX
GPIO 8 = [U6] MISO (Nicht Angeklemmt)
GPIO 9 = [U6] BLK
GPIO 10 = [U6] CS
GPIO 11 = [U6] DC
GPIO 12 = [U6] RES
GPIO 13 = [U6] MOSI (Als SDA bezeichnet)
GPIO 14 = [U6] SCK (Als SCL bezeichnet)
GPIO 15 = [U5] Input 1
GPIO 16 = [U5] Input 2
GPIO 17 = [U5] Input 3
GPIO 18 = [U5] Input 4
GPIO 19 = [U3] TX (Nicht Angeklemmt)
GPIO 20 = [U4] TX (Nicht Angeklemmt)

Helligkeitssensor (BH1750) [U2]:

SCL = [U1] GPIO 4
SDA = [U1] GPIO 5

RFID-Leser Eingang (RDM6300) [U3]:

RX = [U1] GPIO 7
TX = [U1] GPIO 19 (Nicht Angeklemmt)

RFID-Leser Ausgang (RDM6300) [U4]:

RX = [U1] GPIO 6
TX = [U1] GPIO 20 (Nicht Angeklemmt)

Stepper-Control-Board (ULN2003) [U5]:

Input 1 = [U1] GPIO 15
Input 2 = [U1] GPIO 16
Input 3 = [U1] GPIO 17
Input 4 = [U1] GPIO 18

Display (ST7789V2) [U6]:

SCK = [U1] GPIO 14 (Als SCL bezeichnet)
MOSI = [U1] GPIO 13 (Als SDA bezeichnet)
MISO = [U1] GPIO 8 (Nicht Angeklemmt)
CS = [U1] GPIO 10
RES = [U1] GPIO 12
DC = [U1] GPIO 11
BLK = [U1] GPIO 9

13 Installationsanleitung

ACHTUNG: Schließen Sie die Batterie als Allerletztes an!

Die Anlage darf nur mit geschlossener Klappe gestartet werden!

1. Schließen Sie das "Anschlusskabel für einen 9V Batterieblock" an die Eingangsseite des "Abwärtswandler-Moduls", achten Sie auf die richtige Polung.

2. Die Ausgangsseite des "Abwärtswandler-Moduls" wird an den "ESP32-S3-Devkitc-1" angeschlossen, dabei wird der Pluspol an den 5V-Pin und der Minuspol an einen der GND-Pins angeschlossen.

3. Der Anschluss der Sensoren und Aktoren geschieht nach dem Pin-Belegungsplan.

4. Alle Sensoren und Aktoren haben einen GND-Pin, dieser muss an einen GND-Pin des "ESP32-S3-Devkitc-1" angeschlossen werden.

5. Der VCC-Pin vom "BH1750“ und "ST7789V2" müssen an einen 3,3V-Pin vom "ESP32-S3-Devkitc-1" angeschlossen werden.

6. Der VCC-Pin von den "RDM6300" und "ULN2003 Stepper Motor Driver Board Module" muss an der 5V Ausgangsspannung des "Abwärtswandler-Moduls" angeschlossen werden.

7. Der Stepper Motor wird mit der Winde über der Klappe angebracht und das Seil wird sowohl an der Winde als auch an der Klappe befestigt. Der Stepper Motor wird anschließend, mit der vorinstallierten Steckvorrichtung an das "ULN2003 Stepper Motor Driver Board Module" eingesteckt.

8. Die beiden Antennen der "RDM6300" werden mit der vorinstallierten Steckvorrichtung an die "RDM6300" eingesteckt.

9. Die Antenne des „RDM6300“, welche im Pin-Belegungsplan als "Ausgang" beschriftet ist, wird außerhalb des Stalles vor der Klappe angebracht und das mit der Beschriftung "Eingang", innerhalb des Stalles vor der Klappe.

10. Ändern Sie im Quellcode die Variablenwerte der Variablen „ssid“ und „passwort“ zu ihrer WiFi SSID und Passwort um.

11. Die Broker Adresse muss gegebenenfalls sowohl im Python-Quellcode als auch in Node-Red angepasst werden.

12. Nach der Installation der Hardware, wird die Software auf den "ESP32-S3-Devkitc-1" aufgespielt.

13. Zum Abschluss muss nur noch der 9V Batterieblock angeklemmt werden, an das dafür vorgesehene Anschlusskabel.

14 Bedienungsanleitung

14.1 Weboberfläche Node-Red

14.1.1 Reiter "Anlagendaten"

Verbindung:

"Verbunden" = Die Anlage ist in Betrieb und mit dem WLAN verbunden.

"Verbindung getrennt" = Die Anlage ist nicht in Betrieb oder mit dem WLAN verbunden.

Klappenstatus:

"Offen" = Klappe ist geöffnet

"Oeffnet" = Klappe ist am Öffnen

"Schliesst" = Klappe ist am Schließen

"Geschlossen" = Klappe ist geschlossen

Anzahl der Hühner Draussen:

Gibt die aktuelle Anzahl der sich im Stall befindenden Hühner als Ganzzahl an.

Anzahl der Hühner Drinnen:

Gibt die aktuelle Anzahl der sich außerhalb des Stalls befindenden Hühner als Ganzzahl an.

Lux:

Zeigt einen Graphen der Helligkeitswerte in Lux der letzten 7 Tage an.

Knöpfe "AUTO", "HAND AUF", „HAND ZU":

"AUTO" = Anlage befindet sich im Automatikbetrieb.

"HAND AUF" = Anlage befindet sich im Handbetrieb, die Klappe wird aufgefahren und verharrt in dem Zustand.

"HAND ZU" = Anlage befindet sich im Handbetrieb, die Klappe wird zugefahren und verharrt in dem Zustand.

Knopf "Einstellungen":

Wechselt zu dem Reiter Einstellungen, in dem die Einstellungen für den Automatikbetrieb bestimmt werden können.

14.1.2 Reiter "Einstellungen"

Knopf "<-- ZURÜCK":

Wechselt zu dem Reiter "Daten".

Schalter "Auf alle Hühner vor dem Schließen warten":

Aktiviert = Vor dem Schließen der Klappe im Automatikbetrieb, wird gewartet bis sich alle Hühner wieder im Stall befinden.

Deaktiviert = Die Klappe schließt im Automatikbetrieb, auch wenn sich nicht alle Hühner im Stall befinden.

Knöpfe "NACH LUX FAHREN", "NACH UHRZEIT FAHREN":

"NACH LUX FAHREN" = Die Klappe schließt und öffnet sich im Automatikbetrieb nach der eingestellten Helligkeitsgrenze.

"NACH UHRZEIT FAHREN" = Die Klappe schließt und öffnet sich im Automatikbetrieb nach den eingestellten Uhrzeiten.

Helligkeitsgrenze:

Die Helligkeitsgrenze wird in Lux eingestellt. Sobald die eingestellte Grenze für eine eingestellte Zeit über- oder unterschritten wird, fährt die Klappe im Automatikbetrieb. Der Wert kann von 0-1000 Lux in 10er Schritten eingestellt werden.

Helligkeitszeit in Sekunden:

Die Zeit wird in Sekunden eingestellt. Sobald die Helligkeit die eingestellte Helligkeitsgrenze für die Dauer dieser Zeit über- oder unterschreitet, fährt die Klappe im Automatikbetrieb. Der Wert kann von 0-10000 Lux in 1er Schritten eingestellt werden.

Sonnenaufgang:

Die Uhrzeit für das Auffahren der Klappe wird hier in Stunden und Minuten eingestellt. Die Stunden können von 0-11 in 1er Schritten und die Minuten von 0-55 in 5er Schritten eingestellt werden. Nach überschreiten der eingestellten Uhrzeit wird die Klappe im Automatikbetrieb aufgefahren.

Sonnenuntergang:

Die Uhrzeit für das Zufahren der Klappe wird hier in Stunden und Minuten eingestellt. Die Stunden können von 12-23 in 1er Schritten und die Minuten von 0-55 in 5er Schritten eingestellt werden. Nach überschreiten der eingestellten Uhrzeit wird die Klappe im Automatikbetrieb zugefahren.

Hühneranzahl:

Gibt die Gesamtanzahl der Hühner an.

Knopf "ÄNDERUNG SPEICHERN":

Speichert alle im Reiter "EINSTELLUNGEN" bearbeiteten Werte. Die aktuell eingestellten Werte werden auf der rechten Seite festgehalten.

14.2 Display

Das Display zeigt die Anzahl der sich im und außerhalb des Stalles befindenden Hühner an.

14.2.1 Fehlermeldung: "Keine WLAN Verbindung!"

Der Microcontroller konnte sich nicht mit dem WLAN verbinden. Die Anlage sollte mit dem Abklemmen der Batterie neu gestartet werden.

14.3 Allgemeine Infos

Die Anlage kann auch im Offlinebetrieb arbeiten, hierbei befindet sie sich dauerhaft im Automatikbetrieb und fährt nach der Helligkeitsgrenze von 200 Lux. Die voreingestellte Anzahl an Hühnern ist 7 und es wird auf die Hühner vor dem Schließen gewartet.

Wenn ein Huhn im Bereich um die Klappe erkannt wird, durch die RFID-Leser, wird das Schließen der Klappe für den Zeitpunkt pausiert.

Wenn ein Helligkeitswert von -1Lux gemessen wird, ist die Messung gescheitert. Bei anstehenden gescheiterten Messungen, bitte den Anschluss des Helligkeitssensors überprüfen und gegebenenfalls den Sensor wechseln.

Neue RFID-Chips können am RFID-Leser „Eingang“ registriert werden, dies ist nicht unbedingt erforderlich, da die RFID-Chips auch registriert werden, sobald die Hühner den Stall verlassen.

Alle registrierten RFID-Chips werden durch einen Neustart gelöscht.

15 Fazit

Das Projekt ist nicht ganz wie erhofft zu Ende gegangen, da die RFID-Leser Probleme damit haben, bewegende RFID-Chips zu lesen. Außerdem ist die Reichweite der RFID-Leser zu gering für einen realen Einsatz. Dieses Problem könnte zwar mit hochwertigeren Antennen und RFID-Chips behoben werden, dies müsste vorher allerdings ausreichend getestet werden. Eine bessere Lösung wäre der Einsatz von Lichtschranken, auch wenn diese ihre eigenen Probleme haben.

Ein weiteres Problem stellt die Stromversorgung über einen 9V Batterieblock dar. Die Anlage kann mit solch einer Batterie gerade einmal 3 Stunden im Betrieb gehen, da die Anlage 1,9 Wh verbraucht und ein 9V Batterieblock im Durchschnitt 5-6 Wh bietet. Auch der Einsatz von der Funktion „time.sleep“ konnte den Wert nicht nennenswert herunterbringen. Hier wäre die bessere Lösung vermutlich der Anschluss an das 230V Netz mit einem Schuko-Stecker.

Ein behobenes Problem war ein instabiler Verbindungsaufbau zwischen dem Microcontroller und dem WLAN-Netzwerk. Dies konnte nach einigen Stunden Internetrecherche mit der Python-Funktion „network.config(txpower=8.5)“ behoben werden, in der die Sendeleistung auf 8,5 dB gesetzt wird.

Der Rest des Projektes ist erfolgreich gewesen und es bestehen keine weiteren Problematiken. Die Anlage kann auch ohne gut funktionierenden Hühnerzähler ohne Probleme laufen und es ist nur eine Funktion (die Hühnerklappe fährt nur, wenn sich alle Hühner im Stall befinden) eingeschränkt. Da diese Funktion allerdings über die Weboberfläche deaktiviert werden kann, ist die Anlage dennoch einsatzbereit.

16 Quellen

16.1 Benutzte Bibliotheken

uln2003_stepper: larsks
https://github.com/larsks/micropython-stepper-motor/blob/master/motor.py

bh1750: flrrth
https://github.com/flrrth/pico-bh1750

st7789: russhughes
https://github.com/russhughes/st7789_mpy

vga1_16x32: russhughes
https://github.com/russhughes/st7789_mpy/blob/master/fonts/bitmap/vga1_16x32.py

vga1_8x8: russhughes
https://github.com/russhughes/st7789_mpy/blob/master/fonts/bitmap/vga1_8x8.py

rdm6300_rfid: membermatters
https://github.com/membermatters/urdm6300/blob/main/urdm6300/__init__.py

16.2 Hilfsmittel

KI-Chatbot: ChatGPT

Node-Red Videos: https://www.youtube.com/@Opto22

JavaScript Video: All JavaScript Syntax in 53 Minutes – Tutorial

Hilfreiches Forum bei WLAN-Verbindungsproblemen mit dem ESP32:
ESP32-S3 Wifi sometimes working sometimes not / esp32-S3-DevkitC wifi problem [Solved] - ESPHome - Home Assistant Community