Update 15.11.2023

Intelligente Vase, 1985124

## Hardware Einkauf:

* Lichtsensor
* Arduino Uno mit Wifi
* Wasserpumpe mit Erdfeuchtemesser
* Temperatur und Feuchtigkeitsmesser

*Ich bin dabei ausführlich zu schreiben, wieso ich genau diese Teile gekauft habe*

## Erste Schritte mit dem Arduino

*Ich habe einen ersten Testcode, wie ich mir die Verbindung mit dem Arduino vorstelle. Dieser soll über den Sensor Daten erhalten, diese über MQTT weiterschicken (später drahtlos, deswegen die WiFi Variante) und Infos über MQTT erhalten. Im Beispiel um den Widerstand des Lichtsensors einzustellen, da nur Sonnenlicht gemessen werden soll.*

//SetUp M5 Lightsensor, Wifi-Connection and MQTT-Connection

 void setup() {

  Serial.behin(9600);

  M5.begin();

  pinMode(inputPin, INPUT);

  connectWifi();

  client.setServer(mqttServer, mqttPort);

  client.setCallBack(callback);

  client.subscribe(mqttTopic);

}

// Gets the value from the Input and compares it to the one before. If it is different, the value is sent

void loop() {

  if (!client.connected()) {

    reconnect();

  }

  client.loop();

  int previousLightValue = lightValue;

  int sensorReading = digitalRead(inputPin);

  if (sensorReading != previousLightValue) {

    lightValue = sensorReading;

    char message[50];

    sprintf(message, "%d", lightValue);

    client.publish("sensor/light", message);

  }

  delay(1000);

}

void connectWiFi() {

  while (WiFi.status() != WL\_CONNECTED) {

    Serial.print("Attempting to connect to SSID: ");

    Serial.println(ssid);

    if (WiFi.begin(ssid, password) != WL\_CONNECTED) {

      Serial.println("Connection Failed! Rebooting...");

      delay(5000);

      ESP.restart();

    }

  }

  Serial.println("Connected to WiFi");

}

// Logic to implement a base resistance. The light should only measure sunlight, meaning the resistance should be on the level of the base-brightness of the room

void callback(char\* topic, byte\* payload, unsigned int length) {

  String receivedPayload;

  for (int i = 0; i < length; i++) {

    receivedPayload += (char)payload[i];

  }

  // Have to wait until we know how the sensor works

  int newResistance = receivedPayload.toInt();

}

void reconnect() {

  while (!client.connected()) {

    M5.Lcd.print("Connecting to MQTT...");

    if (client.connect("ArduinoClient", mqttUser, mqttPassword)) {

    } else {

      delay(5000);

    }

  }

}

## Erste Schritte des Controllers zwischen Datenbank und Arduino

*Hier werden die Daten über MQTT empfangen und an die Datenbank weitergeben (momentan eine NoSQL-Datenbank, MongoDB).*

def on\_connect(client, userdata, flags, rc):

    print("Connected with result code "+str(rc))

    client.subscribe(topic)

def on\_message(client, userdata, msg):

    received\_number = int(msg.payload.decode())

    send\_light\_to\_mongodb(received\_number)

def send\_light\_to\_mongodb(value):

    try:

        client = pymongo.MongoClient(URI, ssl\_ca\_certs=CERTIFICATE\_PATH)

        db = client[DATABASE\_NAME]

        collection = db[COLLECTION\_NAME]

        data = {"LightInput": value}

        collection.insert\_one(data)

    except Exception as e:

        print("Error:", e)

## Einrichten von MongoDB und Git

Das Projekt wird auf GitHub verwaltet unter:

<https://github.com/tyjga29/IntelligentVase_SA>

## Nächste Schritte

* Reales Arbeiten mit dem Arduino, Einrichten von Sensoren
* Einrichten der Datenbank für verschiedene Pflanzen.
* Recherche der Kamera zur Aufzeichnung von Pflanzenwachstum