1

LUND UNIVERSITY

Technical Requirements

Alan Al-Barazanji, Anton Johannesson, Dlovan Naser, Samuel Schindler Budil

1. Introductions

1.1. Purpose

The purpose of this project is to oversee the optimization of growing environment of fungi, where the project and it's resources can be utilized for other purposes with small changes a~~nd optimiz~~ation.

1.2. Target Audience and Application

The target audience is farms, private persons and businesses.

1.3. Definitions

IDE	Integrated Development Environment
VSCode	Visual Studio Code
VOC	Volatile Organic Compounds
eCO2	equivalent CO2

2. Overall Description

2.1 System Overview

Key components:

- Adafruit Feather RP2040
- Adafruit SGP30 Air Quality Sensor Breakout - VOC and eCO2
- Adafruit AHT20 - Temperature & Humidity Sensor Breakout Board
- Piezo Buzzer
- Protoboard
- Buttons Hur många ?
- Potentiometer

o Skärmen

2.2 System Limitations

- Limited processing and Memory on the Adafruit Feather RP2040
- Limitations such as sensor range and calibration accuracy

Formatering

3. Functional Requirements

3.1. Data Collection

FR1. The system shall collect data on VOC and eCO2 using the Adafruit SGP30 sensor

FR2. The system shall measure temperature and humidity using the Adafruit AHT20 sensor. Mer detalj, hur ofta?

3.2. Data Processing and Analysis

FR4. The system shall compare the sensor readings against predefined threshold values. Vilken?, varför

3.3. User Interaction and Interface

FR5. The system shall display current sensor readings on a Quad alphanumeric display. Vilket format?

kan slås ihop

3.4. Alarms and Notifications

FR6. The alarm should sound at a user specified level using the potentiometer and a user interface. Hur ser detta UI ut?

FR7. The piezo buzzer should be used when initiating the alarm.

4. Non Functional Requirements

4.1. Performance

NFR1. The system shall update and display values of the measurements every x ? seconds.

NFR2. The system shall have easy code that users can easily interact with and understand for further development Hur skulle ni mäta detta?

4.2. Usability

NFR3. The usability of the system shall be simple and easy for users to understand.

NFR4. The systems hardware such as LED screen, shall be easy to understand and analyze for the user

4.3. Reliability

NFR5. The system shall have enough reliability that users should still be able to use the system when incidents occur. What "incidents"?

4.4. Maintainability

NFR6. The code and hardware shall be made so it's easy to maintain, but also developed so users can easily handle the hardware and code

Säger samma sak 2 ggr och upprepar NFR2

5. System Interfaces

5.1 Hardware Interfaces

5.1.1 I2C Communication

- SGP30 (CO2/eVOC)
- AHT20 (Temperature/Humidity)

5.1.2 PWM

- Piezo Buzzer

5.1.3 GPIO

- Button(s) Hur många?
- Potentiometer

5.2 Software Interfaces

5.2.1 Libraries

- “adafruit_gp30” for CO2/eVOC readings
- “adafruit_aht20” for temperature/humidity data
- “pwmio” for PWM control
- “board” for I2C control

6. Design Constraints and Assumptions

6.1. Technical Constraints

Hardware Limitations: The Adafruit Feather RP2040 is the core microcontroller, which has limited processing power and memory. Sensors such as the Adafruit SGP30 (VOC & eCO2) and AHT20 (temperature & humidity) are constrained by range and calibration accuracy. These limitations impose a cap on the complexity of the codebase, sensor sampling rate, and processing logic.

Component Integration: The system must integrate at least two different sensors (already fulfilled by SGP30 and AHT20). It must include interactive elements like buttons (already listed) or potentiometers/rotary encoders. Must provide feedback/output using displays, LEDs, or a buzzer—the project includes a Quad alphanumeric display and a Piezo buzzer.

State Machine Design: The system logic must be structured as a state machine.to manage different operational modes (e.g., idle, measuring, alerting, error handling). This adds a constraint on software architecture.

6.2. Environmental Usage Assumptions

Use in controlled environments (e.g., farms, greenhouses). Tolerance to typical temperature and humidity variations. Stable power supply or battery operation constraints.

7. Testing Requirements and Verifications

7.1 Functional Testing

Sensor Validation: Ensure the VOC, eCO2, temperature, and humidity readings are collected and processed accurately. Validate thresholds and correct triggering of alarms or feedback mechanisms (buzzer/display). Huh?

State Transitions: Test all transitions between states in the state machine: From idle to active, active to alert, alert to reset, etc. Simulate scenarios like sensor faults or environmental anomalies to confirm appropriate responses.

User Interaction: Validate correct input recognition from buttons or any other interactive components. Ensure the display shows accurate and updated data.

7.2. Non Functional Testing

Performance: Ensure that sensor values are updated and displayed every X ? seconds. Check for any latency or lag in data collection or interface feedback.

Usability: Test the interface for ease of use—ensure users can intuitively read outputs and understand system states. Verify that display messages and buzzer alerts are clear and meaningful.

Reliability: Simulate errors (e.g., sensor disconnection, power loss) and verify that the system remains usable or fails gracefully. Vilka, vad är "fails gracefully"?

Maintainability: Check that the hardware setup and codebase are structured and documented for easy maintenance and extensibility.