Your-Communal-Drink-Dispenser

DTU

Gruppe 7

Ditte Marie Næss Lindgreen	S223971
Mads August Claussen	S223952
Valdemar Grabowski Secher	S223973
Tobias Hjelholt Svendsen	S215600
Konrad Sommerfeld Gran	S223943
Lasse Rindal Blücher Jensen	S205367
Mads Møller Bjørnskov	S223940
Andreas Rattenborg Holm-Hansen	S215000

Can Dispensing Machine System Overview

Your-communal-drink-dispenser (YCDD) is designed to dispense cans when a user interacts with a user interface separate from the dispenser mechanism. The design is modular at its core and the architecture is developed to be able to deploy multiple devices and modules.

The dispenser is housed inside the fridge, ensuring that the cans remain chilled until dispensed. Each dispenser box can hold up to 17 cans, which are guided through the box via rails. At the bottom of these rails, two parallel stepper motors are installed to control the wheel that dispenses the cans.

When an order is sent from the user interface, the user will first pull out a tray located at the bottom of the dispenser box. Once the tray is pulled out, the stepper motors receive a signal to start dispensing the cans. The dispenser box is equipped with a counting mechanism that tracks the number of cans dispensed, ensuring accurate inventory management. The tray can hold a maximum of three cans per order per machine. After the cans are removed, the user pushes the tray back into place. When the machine's inventory is running low and it is restocked, the system resets its inventory count to maintain accuracy.

The user interface used to operate the dispensing machine is not physically wired to the dispenser box; instead, it communicates with MQTT. This interface is mounted on the outside of the fridge door. To use the interface, the user must first scan their RFID chip on a chip reader. Following this, a rotary encoder is used to navigate and select options displayed on an LCD screen. Once the order is confirmed, it is sent with MQTT to Node-RED which logs the order, and sends a new MQTT message to the dispensing mechanism inside the fridge. The user then opens the fridge and pulls out the tray containing the ordered cans.

In addition to the physical user interface, the system includes an app for administrative purposes. This app serves as a tool for the machine administrator to manage users, oversee payments, and monitor the dispensing machine's content for restocking. The admin can add or remove users and modify the content of the dispensing boxes as needed. The app provides a real-time count of the cans available in each machine, ensuring efficient inventory management. When payments are due, the admin can send invoices to users via email, detailing the payment amount and instructions.

When the dispenser box is about to run out of cans, the admin receives a notice via the app. The admin must then manually refill the box. To do this, the admin can open the front of the dispensing box, providing access for refilling the cans. This front panel also allows the admin to access the electronics, making it easier to perform maintenance or adjustments as needed. This ensures that the machine remains in good working order and can continue to dispense cans efficiently.

Evaluation of the prototype

We achieved a functional prototype that will do as described above. But via testing and prototyping we learned a lot about our design and how we could improve it.

First of all we experienced problems with the quality of the components causing problems along the way when prototyping. The motor selection will be crucial in the future and the possibility of implementing mechanical gearing for smoother operations could be an option. Testing the rails was fruitful as we experienced a lot of issues with the feeding of cans. The first iterations proved “too good” as the cans accelerated down the rails and smashed into the motor and dispensing mechanism. The rails needed to be more refined not to accelerate the cans. Further development is needed regarding the rails. We also learned that it was necessary to implement stopping mechanisms to secure that if cans are stuck the dispensing stops. We also need to deploy another sensor to secure the beers are dispensed.

All of the above results in a fairly unreliable prototype and points in the directions we can work to make the product better and make the product more redundant. But after all that is why we make prototypes and we are content with our beta prototype and its functionality.

Key aspects from the process documentation

The development of this product employed parallel and iterative prototyping methodologies. We established five parallel tracks, each dedicated to a subsystem of the dispenser unit, as depicted in the Protomap.

During Sprint 1, we assigned each subsystem to different group members. Based on test results, we selected concepts for the UI, cabinet/guiding, tray, and refill/security subsystems.

In Sprint 2, we did further selection on the dispensing and security subsystems while achieving further refinement of the other subsystems. By the end of Sprint 2, we assembled a Beta-1.0 prototype incorporating all subsystems, allowing us to test their interactions.

Sprint 3 was primarily dedicated to implementing minor improvements and assembling the Beta-2.0 prototype. By the end, both the Beta-1.0 and Beta-2.0 prototypes were operational and integrating seamlessly with the database and website. Protomap in appendix

Reflections for future improvements

As we move forward with our project, we have a clear plan for both short-term and long-term developments. Our focus is on improving functionality, reliability, and user experience.

In the short term, we are concentrating on several important tasks. We will research an option to develop mechanical gearing to optimize operation and select the best motors to enhance performance and durability. We are also extending the rails to provide a better feeding flow of the cans. Further development of a “kill switch” and/or other sensors to stop the feeding mechanism should problems arise. To improve reliability, we are introducing mechanisms to verify that a beverage has been successfully dispensed.

On the software side, we are making key short-term improvements. We are developing robust error handling for mechanical inputs to effectively manage potential issues. We are setting up a reliable hosting service for seamless connectivity and operations. Additionally, we are creating an administrative database to efficiently manage multiple fridges and customer data, allowing for future expansion.

Looking ahead to the long term, we have several goals. We will conduct user testing to gather feedback and refine the product. We plan to expand our product line to include a family of products for different needs and environments. We are exploring the integration of biosensors to enhance functionality and potentially offer personalized user experiences. We are committed to improving the reliability of the system to ensure consistent performance. Optimizing the system topology will help improve efficiency and reduce operational costs. We are also working on better cooling mechanisms to maintain optimal temperatures and extend the shelf life of stored items. Finally, we will implement comprehensive electronic testing in different environments to ensure all components meet high standards of quality and performance.

Our approach allows us to address immediate needs while planning for future improvements. By focusing on both mechanical and software enhancements, we aim to deliver a reliable and user-friendly solution. We will continue to provide updates as we make progress.

BOM

Control unit

Part Name	Quantity	Mass per unit (g)	Material	Manufacturing Process	Part cost per unit (DKK)
ESP32S-HiLetgo	1	10	PCB, electronic components	SMT	50
KY-040 Rotary encoder	1	15	PCB, electronic components	SMT	15
LCD screen	1	50	LCD, plastic	Injection Molding, SMT	100
LCM1602 IIC	1	30	LCD, plastic	Injection Molding, SMT	75
RFID RC522	1	12	PCB, electronic components	SMT	40
YwRobot Breadboard Power Supply	1	25	Plastic, electronic components	Injection Molding, SMT	20
Male to Male cables	X	1 (each)	Copper, PVC	Extrusion	0,5
M3 bolt	4	1	steel	Extrusion	0,5
3D printed electronics case	1	156	PLA	FDM	15

Beta dispenser

Part Name	Quantity	Mass per unit (g)	Material	Manufacturing Process	Part cost per unit (DKK)
28BYJ-48 Stepper Motor	2	50	Electronic components	SMT	20
ESP32S-HiLetgo	1	10	PCB, electronic components	SMT	50
HC-SR04 Ultrasonic Distance Sensor	1	30	Electronic components	SMT	100
KY-006 Passive Buzzer Module	1	5	Electronic components	SMT	10
KY-040 Rotary encoder	1	15	PCB, electronic components	SMT	15
Pushbutton	2	5	Electronic components	SMT	5
Green LED	2	5	Electronic components	SMT	10
Red LED	2	5	Electronic components	SMT	10
X113647 Stepper Driver Board	2	20	Electronic components	SMT	40
YwRobot Breadboard Power Supply	1	25	lastic, electronic component	Injection Molding, SMT	20
830 Tie Points BreadBoard	1	79	ABS Plastic	Injection Molding	69
Male to Male cables	x	2	Copper, PVC	Extrusion	0,5
USB A to mini USB	1	20	Plastic, Copper	Assembly	10-30
The Hook rotor	2	10	Metal, Plastic	3D Printing	5
3D printed knobs	1	3	PLA	D printing, Carving, Sandin	1
3D printed stepper brackets	2	12	PLA	3D Printing	5
Shell/Box	1	500	Acrylic	CNC	300
POM Rails	2	200	Acrylic	CNC	100
Lock and key	1	50	Metal, Plastic	Mix	85
The Hook rotor	2	10	Metal, Plastic	3D Printing	5
Al welding rods Ø3.2mm,128 mm	30	5	metal	Exstrusion	1
Spacers (steel tube)	30	3	Steel	Exstrusion	1

Pre-Beta dispenser

Part Name	Quantity	Mass per unit (g)	Material	Manufacturing Process	Part cost per unit (DKK)
28BYJ-48 Stepper Motor	2	50	Electronic components	SMT	20
ESP32S-HiLetgo	1	10	PCB, electronic components	SMT	50
HC-SR04 Ultrasonic Distance Sensor	1	30	Electronic components	SMT	100
KY-006 Passive Buzzer Module	1	5	Electronic components	SMT	10
KY-040 Rotary encoder	1	15	PCB, electronic components	SMT	15
Green LED	2	5	Electronic components	SMT	10
Pushbutton	2	5	Electronic components	SMT	5
Red (633nm) LED	2	5	Electronic components	SMT	10
X113647 Stepper Driver Board	2	20	Electronic components	SMT	40
YwRobot Breadboard Power Supply	1	25	lastic, electronic component	Injection Molding, SMT	20
830 Tie Points BreadBoard	1	79	ABS Plastic	Injection Molding	69
Male to Male cables	x	2	Copper, PVC	Extrusion	0,5
USB A to mini USB	1	20	Plastic, Copper	Assembly	10-30
The Hook rotor	2	10	Metal, Plastic	3D Printing	5
3D printed numbs	1	3	PLA	D printing, Carving, Sandin	1
3D printed stepper brackets	2	12	PLA	3D Printing	5
Shell/Box	1	500	Wood	Lasercut	100
Wood Rails	2	200	Wood	Lasercut	100
Lock and key	1	50	Metal, Plastic	Mix	85
The Hook rotor	2	10	Metal, Plastic	3D Printing	5
Al welding rods Ø3.2mm,128 mm	30	5	metal	Exstrusion	1

PRD

OVERVIEW

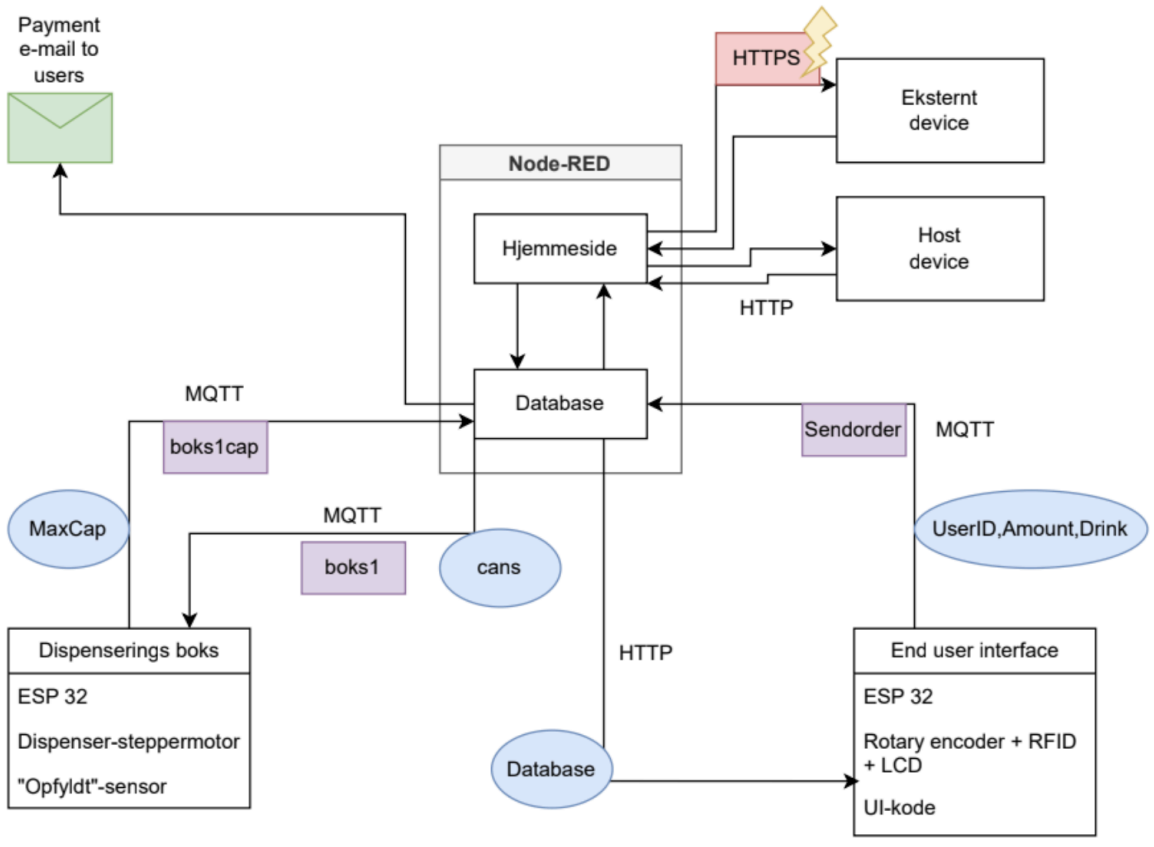
Product Description: "Your communal drink dispenser" is a can-sharing mechatronic system designed to manage beer sharing in communal fridges. It addresses the issues of theft, forgetfulness in registering beers, and inconsistent payment intervals.	
Market Need	The product solves the problem of trust breakdown in communal beer-sharing systems, ensuring accurate tracking and fair payment for shared beverages.
Key Features/ Functionality	● Automated beer dispensing upon user registration ● Monthly billing for consumed beverages ● Easy integration into existing fridges due the dimensions fit a standard fridge ● Registration required each time three cans are taken ● Cold storage for beers ● Robust design
Other Product Compatibility, Ecosystem, etc.	● Compatible with standard fridge sizes ● Uses ESP32, Stepper motor, and IoT integration via Node-RED ,MQTT, HTTP and app support
Stakeholders	
Target User	● Residents in dorms ● Employees in workplaces with communal fridges
Other Stakeholders	● Admins responsible for restocking and maintaining the system

DESIGN

Brand	The product should communicate reliability, convenience, and fairness in communal sharing. It should reflect the values of trust and technological innovation.
Connectors (Power, USB, Lighting, Audio)	● Power connector for the system ● USB-C for programming and updates
Visual Interface (Screen size and type, LEDs)	● 16x2 LCD screen for user interaction and status updates on a separate unit outside the fridge ● Red and Green LEDs to indicate operation status and errors
Touch Interface (Mechanical actuators/switches, touch sensitivity, haptics)	● Keypad for user registration on a separate laptop. ● User input recorded via rotary encoder on physical UI.
Audio Interface (Microphones, speakers)	● Passive buzzer for auditory feedback on user actions

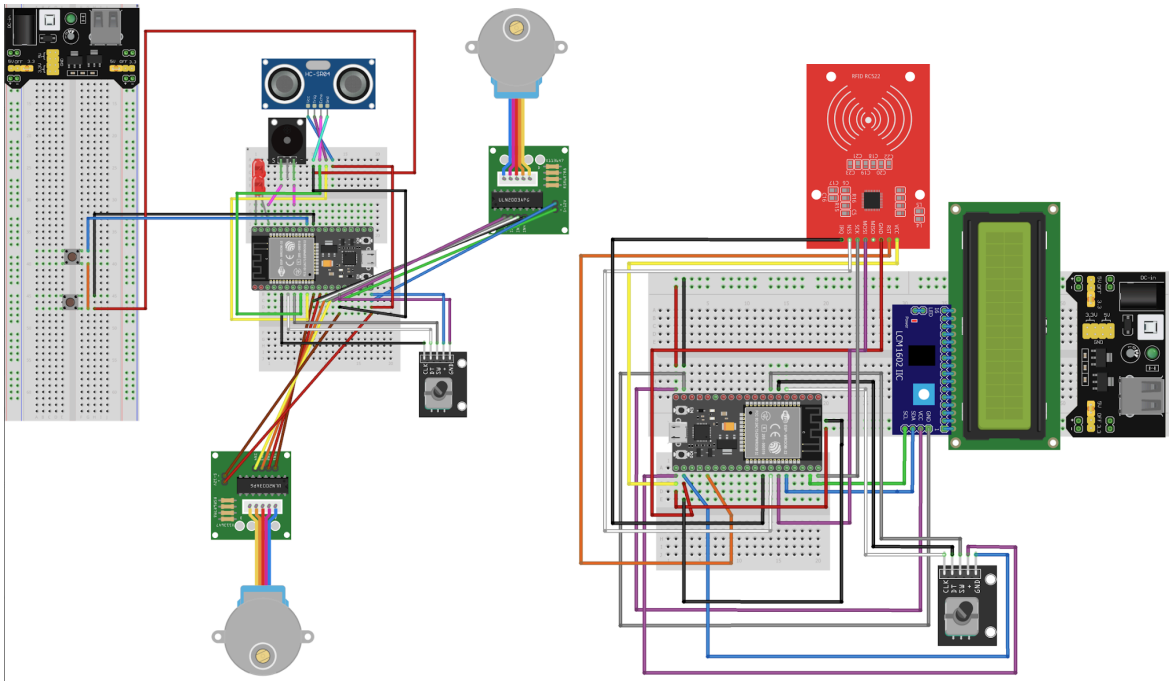
SOFTWARE ARCHITECTURE AND DATA PROCESSING

Block Diagram of Data Flow



ELECTRICAL, HARDWARE AND SENSORS

Block Diagram of Electrical Hardware



Left is the main system and right is the RFID system/ operation box

Input/Sensor Requirements	• Rotary encoder input for user end ordering sequence. • Sensors to detect beer can presence and movement
Output/Actuator Requirements	• Stepper motor to dispense beer cans • LEDs for status indication • Buzzer for auditory feedback
Critical BOM Components	• Stepper motors • Breadboards small and large • Cables • Red LED • Green LED • ESP32 • Potentiometer • Case for drink dispenser unit • Case for End-User-Interface
Communication Requirements	• WiFi for IoT connectivity and data transfer • MQTT for messaging and notifications
Power Requirements	• The system should be powered via an outlet

	• Optionally, rechargeable batteries can be considered for backup power, with a target operational time of several days between charges

REFERENCES

Database development:

https://chatgpt.com/share/cebc4e06-9598-4eb9-90b7-fc8de7a6a790

https://noderedguide.com/tutorial-sqlite-and-node-red/

(dispenser unit, MQTT communication)
https://chatgpt.com/share/70d04354-8ce2-4edc-954a-9aff5ea8c3b0

Development af database, hjemmeside og arduino kode (Mads August):

RFID og Encoder Integration
https://chatgpt.com/share/0985c826-c8bd-4fe0-8f65-3c0dea9bcb28
Connect RFID RC522 with ESP32
https://chatgpt.com/share/a4ef0673-1a18-463d-a936-da9d9491e6d1
Flatten JSON efficiently
https://chatgpt.com/share/5fd2710f-7df1-45c2-b238-ef7ee816d0a6
ESP32compatibility update
https://chatgpt.com/share/369c880f-6f41-4adf-885f-46abf7b075d7
ESP32 better for MQTT
https://chatgpt.com/share/2aff84c1-efaf-439f-b0c7-e6c057a1f19d
Arduino wifi http get
https://chatgpt.com/share/4dc3987e-96be-4b54-b781-69d9bbf4970a
Html form integer values
https://chatgpt.com/share/ea6ea0ba-7e42-4333-b1ae-398b5338d3d3
Arduino Program issues
https://chatgpt.com/share/2b112835-987f-4d7a-b99e-b67ec63a0e91
Convert int to string

https://chatgpt.com/share/4093ef7b-802a-4d9f-b140-1462bb06c3ea
Fix code global variables
https://chatgpt.com/share/7a7d7dde-028e-4831-9856-ab32f1400ed7
Convert JSON to array
https://chatgpt.com/share/e4b24074-61ef-4d5e-8b05-d7128eb99287
MQTT JSON send/receive
https://chatgpt.com/share/a9bd2349-0863-4f65-8fe6-8bcd237aa8b2
MQTT function integration
https://chatgpt.com/share/076111f1-eb01-4767-8a07-dc937b2bdbc4
Get Drinks JSON response
https://chatgpt.com/share/a2d3fdb7-5402-451e-a621-e7160c5dc1fa
Node-RED flow for MQTT/Sqlite
https://chatgpt.com/share/7c85f591-6aa8-421c-ab86-ebb6214da135
Troubleshoot CSS margin
https://chatgpt.com/share/90d11725-9559-4431-ac90-944dad236d6f
Toggle burger menu functionality
https://chatgpt.com/share/bb656a2b-8414-4aa1-91c5-b6c72d65fee1
Encourage right swipe on website
https://chatgpt.com/share/2056ab6c-5c04-40de-8ad4-da0da43233c7
Beer Dispenser speed control
https://chatgpt.com/share/45069c6d-5671-4006-bd3a-09461a50eb9a
Website creation with Node-RED
https://chatgpt.com/share/30707781-8c82-41ba-a888-0600c2766a00
OLED vs. LCD Display.
https://chatgpt.com/share/227f7a8f-c53e-4c5e-a4ab-c1ba6322f974