The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

~~EIE4430~~ / EIE4433 Honours Project

1. Student Name: _____Sam Pak Sam TAM____________ (Student No.: 19016111D)

2. Programme Code: 42470-SY / ~~42477~~

3. Project Title: Water Drinking and Pill Taking Reminder

4. Supervisor Name: Dr Lawrence Chi-Chung CHEUNG

5. Project summary (State clearly the project objectives and results achieved by yourself.)
   [Please list in point form where appropriate]

This project aims to develop a system that helps the staff in hospital wards and nursing homes to monitor the water and pill consumption of the people receiving care in a convenient and systematic way and reminds the staff if any action is required.

The system is called Water Drinking and Pill Taking Reminder. It is an IoT system that includes a server and IoT devices. They communicate wirelessly through LoRa. The battery-powered IoT device measures the water and pill consumption of the person receiving care. The server analyses the data and displays them through a web interface designed for both computers and smartphones.

This report illustrates the hardware and software development and the performance of the system and discusses the decision made and the difficulties and solutions during the development process.

DECLARATION OF ORIGINALITY

Except where reference is made in the text of this report, I declare that this report contains no material published elsewhere or extracted in whole or in part from any works or assignments presented by me or any other parties for another subject. In addition, it has not been submitted for the award of any other degree or diploma in any other tertiary institution.

No other person's work has been used without due acknowledgement in the main text of the Report.

I fully understand that any discrepancy from the above statements will constitute a case of plagiarism and be subject to the associated.

Signature

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Department of Electronic and Information Engineering

Final Year Project Final Report

(2022/23)

Water Drinking and Pill Taking Reminder

Student Name:	Sam Pak Sam TAM
Student ID:	19016111D
Programme Code:	42470-SY
Supervisor:	Dr Lawrence Chi-Chung CHEUNG
Submission Date:	9th April 2023

Abstract

This project aims to develop a system that helps the staff in hospital wards and nursing homes to monitor the water and pill consumption of the people receiving care in a convenient and systematic way and reminds the staff if any action is required.

The system is called Water Drinking and Pill Taking Reminder. It is an IoT system that includes a server and IoT devices. They communicate wirelessly through LoRa. The battery-powered IoT device measures the water and pill consumption of the person receiving care. The server analyses the data and displays them through a web interface designed for both computers and smartphones.

This report illustrates the hardware and software development and the performance of the system and discusses the decision made and the difficulties and solutions during the development process.

Table of Contents

Abstract....................2
Table of Contents....................3
Acknowledgements....................5
Introduction....................6
Background....................7
Overall Design....................11
Implementation....................15
Hardware Development....................15
The electronic circuits of the IoT device....................15
The electronic circuit of the server....................16
The case of an IoT device....................17
The case of the server....................21
Software Development....................24
The software development for an IoT device....................25
Server’s software....................27
LoRa module’s settings....................38
Performance Investigation....................39
IoT device’s power consumption....................39
IoT device’s battery life....................40
IoT device’s response rate....................40
Water scale accuracy....................40
Pill scale accuracy....................41
LoRa’s response time....................42
Water reminder accuracy....................42
Pill reminder accuracy....................43
Discussion....................44
Decision justification....................44

Choosing LoRa as the wireless communication technology 44

Choosing LoRa module A39C-T400A22D1a 44

Choosing Arduino as the MCU of the IoT device 44

Choosing an 18650 battery as the power source of the IoT device 45

Choosing Raspberry Pi 4B as the server 45

Choosing Linux over Windows as the server operating system 46

Choosing 3D printing as the case's fabrication technique 46

Choosing ABS as the 3D-printed case material 46

Difficulties and solutions 47

A39C-T400A22D1a (LoRa module) setup 47

Battery life 47

Reducing the Arduino running voltage to 3.3V 47

Load cell readings fluctuation and drift 47

Purchasing Raspberry Pi 4B 48

Accessing a GPIO pin with multiple processes 48

Wrapping and bed adhesion problem in 3D printing ABS 48

Conclusion 49

References 50

Acknowledgements

I would like to express my deepest appreciation to the Hong Kong Polytechnic University, the Department of Electronic and Information Engineering, and my project supervisor, Dr Lawrence Chi-Chung Cheung, for their invaluable support and guidance throughout my university life and my final year project.

First and foremost, I would like to express my sincere gratitude and appreciation to the Hong Kong Polytechnic University for providing me with a valuable opportunity to study at one of the best universities in the world to develop my hobbies and my future carrier. I am grateful for the resources and support provided by the university to help me acquire my knowledge.

I would also like to extend my heartfelt thanks to the Department of Electronic and Information Engineering (EIE) for providing some fantastic courses and some helpful staff. With their support, I had an exceptional learning experience in my university life. Thanks, EIE, for equipping me with the knowledge and skills and providing an irreplaceable opportunity for me to demonstrate my accomplishment in the undergraduate course through this project.

Furthermore, this journey would not have been possible without my project supervisor, Dr Lawrence Chi-Chung Cheung. I am deeply indebted to his passionate guidance, extensive support and insightful feedback throughout my college life and my final year project. His expertise, passion and patience have been encouraging me and guiding me toward success. His quality lessons laid a great foundation for my programming skills.

Last but not least, I would like to extend my special thanks to my parents and my friends for their encouragement and mental support. Specifically, thank you to my parents for the financial support throughout my college life. Without them, I would not have had a chance to study at a university.

Once again, thank you to the Hong Kong Polytechnic University, the Department of Electronic and Information Engineering and Dr Lawrence Cheung for their invaluable support and guidance throughout my project.

Introduction

It is important to help elderly people and patients to drink water and take pills regularly. Traditionally, these tasks are processed by caregivers. They walk around and check if everything is fine. However, it is repetitive. Moreover, the workload may be very high if the number of people being served is large.

Edkins G. [1] shows that humans make 3-6 errors per hour regardless of the task being conducted. Passing the above task to a machine can improve the service quality and reduce the number of errors caused by humans. Additionally, the workload of the caregivers can be reduced so that they can focus on other tasks to improve service quality.

Currently, there are some commercial products for water reminders, water checking and logging, or pill reminders. The existing commercial products are designed for personal use and for either water or pills, but not both. Thus, they have not been able to satisfy the needs of hospital wards and nursing homes as their staff cannot monitor those data easily. Moreover, they have a limited coverage area as they typically use Wi-Fi or Bluetooth for wireless communication. For example, the MedMinder Medication Dispenser [2] aims to help the staffs dispense pills but not assists the staffs to check if the pill is taken and log the data. HidrateSpark's water bottles [3] provide water reminders, checking and logging. However, the bottles connect to smartphones or tablets through Bluetooth, and the connection distance is short. Also, the application can only connect to one water bottle at the same time. Tinylogics' smart pillbox: Memo Box [4] reminds the user to take pills and logs the data. However, it connects to smartphones or tablets through Bluetooth, and the connection distance is short.

Based on the above limitations, we develop a system called Water Drinking and Pill Taking Reminder. Water Drinking and Pill Taking Reminder aims to help staff in nursing homes and hospital wards monitor the water and pill consumption of the elderlies and the patients. It has four main features:

- Monitor and log the water consumption of the user.
- Check if the user has taken the pill and log the time of taking the pill.
- Show the above data in a web interface.
- Notify the staff if any action is needed.

Background

In the project, we use many devices and software in the design and implementation. In this section, we give some brief descriptions about different components:

1. IoT

Water Drinking and Pill Taking Reminder (WDPTR) is an Internet of Things (IoT) system. IoT describes the physical objects embedded with sensors and software connected to the network [5].

2. MCU

MCU is the short form for microcontroller unit. The core elements of an MCU are the processor, memory and input/output peripherals [6].

3. Arduino and Arduino board

Arduino is an open-source electronics platform [7]. It mainly includes Arduino board (hardware) and Arduino IDE (software) [8].

An Arduino board is a physical circuit board with a programmable MCU, a USB port and different input/output pins. The developer can programme the MCU through a USB cable instead of a separate piece of hardware [8].

The input/output pins on the Arduino board include GPIO pins and ADC pins. General-Purpose Input-Output (GPIO) sends or receives electrical signal [9], which is either high or low. Analogue to Digital Converter (ADC) translates the analogue electrical signal into a digital output code which represents the analogue signal [10].

The Arduino board uses different communication protocols, including UART and I^2C, to communicate with other MCUs. Both Universal Asynchronous Receiver-Transmitter (UART) and Inter-Integrated Circuit (I^2C) are protocols for serial data transfer [11] [12].

The Arduino board uses an LDO regulator to regulate the board voltage. A linear and low-dropout (LDO) regulator is a voltage regulator that provides a regulated output voltage powered from a higher input voltage [13].

4. Arduino IDE

An integrated development environment (IDE) is a software application that facilitates programme code development [14]. Arduino IDE provides a platform for editing and compiling programme codes, uploading codes to the Arduino board, and a debugger for debugging [15].

5. Arduino language

The Arduino language is implemented in C/C++ language [16]. It is a programming language for programming Arduino boards.

6. LoRa

LoRa, which stands for Long Range, is a patented wireless communication technology that combines ultra-low power consumption with an effective long-range [17]. LoRa typically transmits data over more than 15km in rural areas and up to 5km in urban environments [18]. It is possible to transmit data over 832km using 25mW [19].

7. Load cell

A load cell is a transducer that converts force into a measurable electrical output [20], such as voltage. The typical sensitivity values are 1 to 3 mV/V [21], which means the output voltage is 1 to 3 thousandths of the input voltage.

8. Raspberry Pi 4B

Raspberry Pi 4B is a small single-board computer produced by Raspberry Pi Foundation [22]. It provides input and output connections, including Wi-Fi, USB, HDMI, and GPIO pins. Raspberry Pi 4B runs on a variety of operating systems, including Raspberry Pi OS.

9. Raspberry Pi OS

Raspberry Pi OS is a free operating system based on Debian [23], a particular distribution of the Linux operating system [24]. The Raspberry Pi Foundation builds the Raspberry Pi OS specifically for the Raspberry Pi computers to have the necessary drivers pre-installed.

10. LAMP server

A server is a software and hardware device that responds to requests sent by other devices on the network [25]. A LAMP server is a server that runs on the Linux operating system and provides an Apache Web server, MariaDB (or MySQL) database management system (DBMS), and PHP and Python scripting language [26].

- Linux

Linux is an open-source operating system that provides flexibility to run any programme for any purpose [27].

- Apache

Apache is a free open-source Web server software that allows users to host a Web server on an operating system [28].

- MariaDB and MySQL

MariaDB and MySQL are both DBMSs. MariaDB is a drop-in replacement for MySQL [29] as it is faster and completely open-source [30].

- PHP and Python

Hypertext Pre-processor (PHP) is a scripting language. It suits for web development and can be embedded into HTML [31]. Python is a high-level object-oriented programming language [32]. LAMP servers can run Python scripts as services.

- SQL

Structured query language (SQL) is a programming language for accessing a database [33].

- Service

Service in Linux is a script, a programme or an application that the user does not need to aware of runs in the background [34].

- IPC

Inter-process communication (IPC) is a mechanism that allows processes to communicate with each other [35].

11. Library

A programming library is a collection of pre-written programme code that programmers can use to optimise tasks [36].

12. Master-slave mode

Master-slave mode is an asymmetric communication model that allows one device (the "master") to communicate with all devices in the same network (the "slaves") while the slaves can only communicate with the master.

13. 18650 Li-ion battery

18650 lithium-ion battery is a rechargeable cylindrical battery that has no memory effect and has a theoretical cycle life of 1000. It has a diameter of 18mm and a length of 65mm. It usually has an operating voltage of 3.7V and is fully charged at 4.2V [37].

14. FDM 3D printing

3D printing is a manufacturing process that constructs three-dimensional objects from digital files. Fused deposition modelling (FDM) is one of the 3D printing technologies. It prints parts layer by layer by extruding molten thermoplastic polymers, such as ABS or PLA, into pre-determined locations to form 3D objects [38].

15. Heat set inserts

A heat set insert is a small metal tube threaded on the inside and knurled on the outside [39]. We usually install them into thermal plastic parts for fastening.

Figure 1 - Heat set inserts.
Source: Adapted from [59]

Overall Design

The Water Drinking and Pill Taking Reminder (WDPTR) consists of a server and IoT devices. The block diagram below shows the structure of the system.

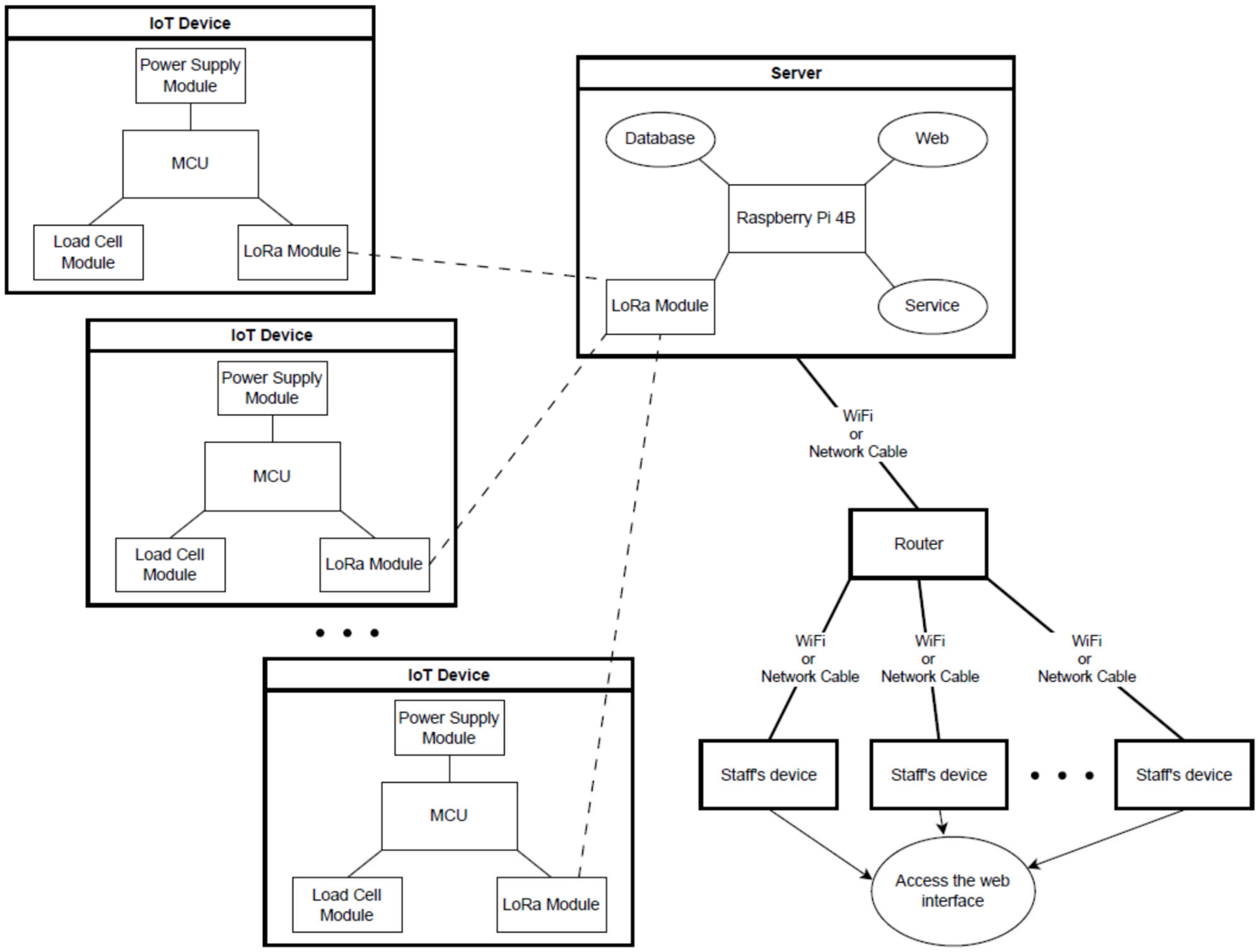

Figure 2 - WDPTR system structure block diagram

The system provides four functions:

1. *The system monitors and logs the water consumption of different users*: The server periodically broadcasts a request to all IoT devices (one IoT device for one user) through a LoRa network. When the LoRa module of an IoT device receives a request, it wakes the MCU up because the IoT device is in low-power mode to save power when it is idle. The MCU initialises its load cell and uses it to measure the weight of its water container with water inside. Then it sends the data to the server and goes back to low-power mode. The server receives the data, decrypts it and logs it into different databases respectively.
2. *The system checks if the user has taken pills and logs the time of taking the pills*: The IoT device is in low-power mode to save power when it is idle. The server

periodically broadcasts a request to all IoT devices through a LoRa network. When the LoRa module of an IoT device receives a request, it wakes the MCU up. The MCU initialises the load cell and uses it to measure the weight of its pill container with pills. Then it sends the data to the server and goes back to low-power mode. The server receives the data, decrypts it and logs it into the database. The server also checks if the pills are taken periodically. If the pills are taken, the server saves the time of taking the pills into the database.

3. *The system shows the current status of the water and pill containers through a web interface*: The system uses the database system to store the status of the water and pill containers and hosts the Web server to provide a platform for a web interface. The web server uses PHP files to implement the web interface. The PHP files get the status of the water and pill container and use different HTML codes to present the status through the web interface.
4. *The system notifies the administrator(s) if any action is required to be taken:* The server uses PHP codes with SQL statements to read the data from the databases and uses PHP codes to process the data. If any action is required to be taken, the server uses PHP files with HTML codes to show notifications on the homepage of the web interface. The following situations require the system to notify the administrator(s):
   - An IoT device is out of battery.
   - The pills are not taken within a specified time period.
   - The water container should be refilled.

The following photos show the outlook of the server, IoT device and web interface.

Figure 3 - Photo of WDPTR server

Figure 4 - Photo of WDPTR IoT device

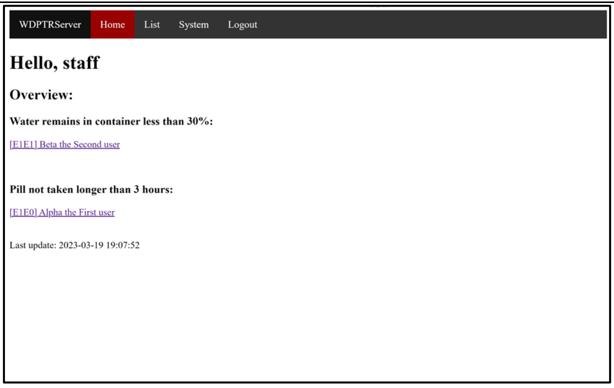

Figure 5 - Screen capture of the home page of the web interface on a PC (Dimension: 1017px x 632px)

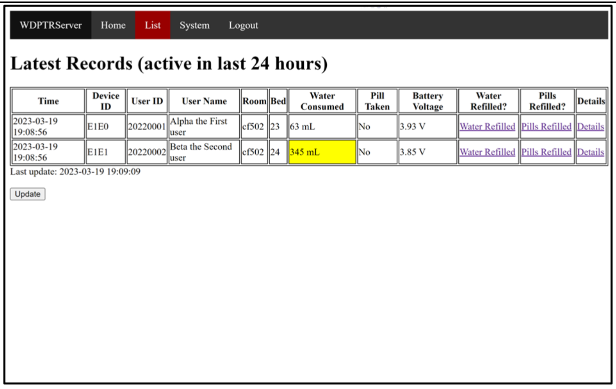

Figure 6 - Screen capture of the list page of the web interface on a PC (Dimension: 1017px x 632px)

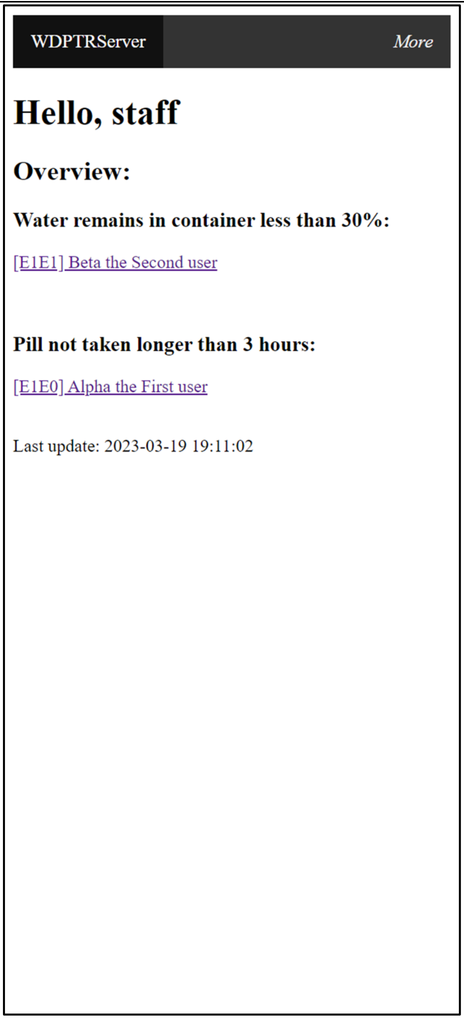

Figure 7 - Screen capture of the home page of the web interface on a mobile phone (Dimension: 412px x 915px)

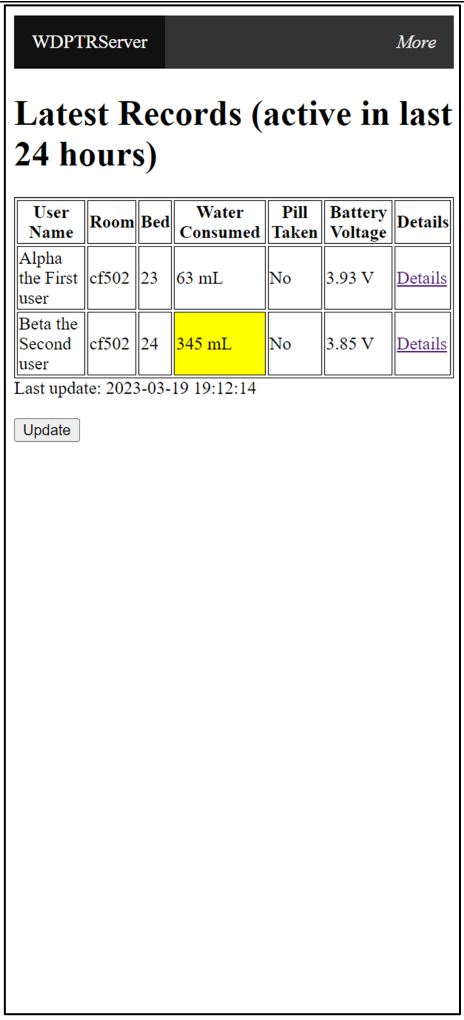

Figure 8 - Screen capture of the list page of the web interface on a mobile phone (Dimension: 412px x 915px)

Implementation

This section shows the implementation of the Water Drinking and Pill Taking Reminder (WDPTR). It covers the hardware and the software development for both the server and the IoT device.

Hardware Development

The electronic circuits of the IoT device

The figure below shows the schematic of the electronic circuit for the development of the IoT devices.

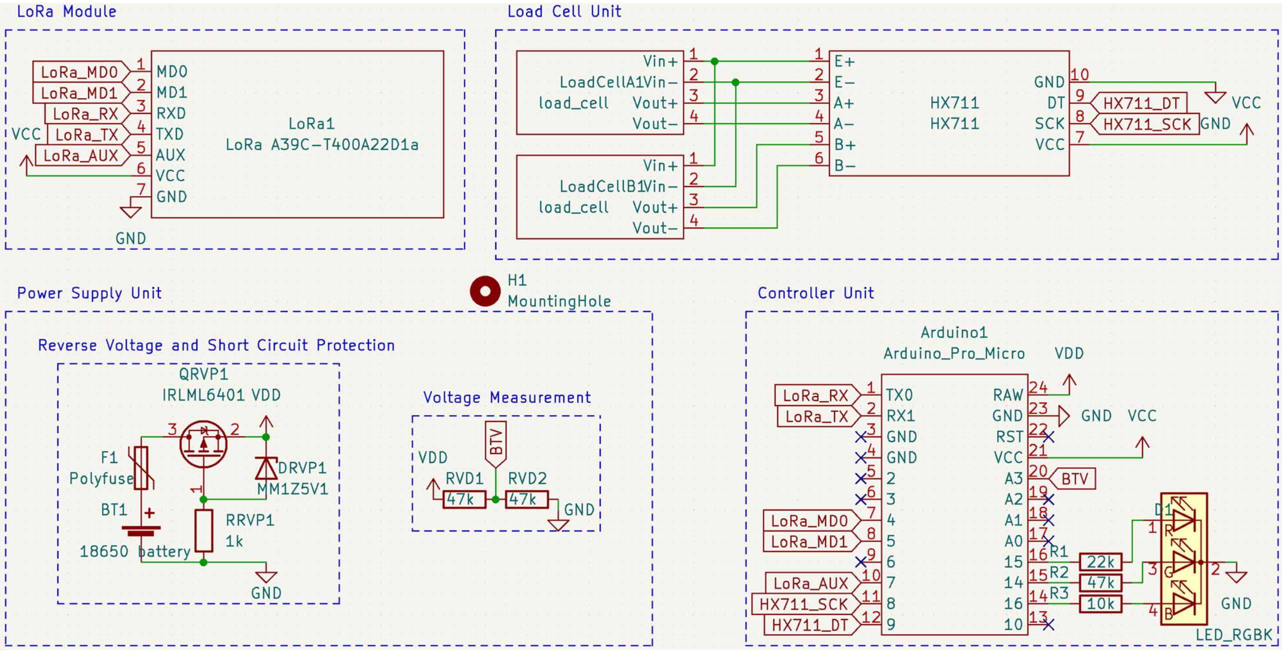

Figure 9 – Schematic of the electronic circuit for the development of the IoT device

There are four parts in the electronic circuit. They are the controller unit, power supply unit, load cell unit, and LoRa module.

The WDPTR IoT device uses an Arduino Pro Micro [40] with an ATmega32U4 [41] MCU. It runs on 5V/16MHz. We replace the LDO regulator on the Arduino Pro Micro (i.e., MIC5219-5.0YM5-TR) by MIC5219-3.3YM5-TR [42] to run the Arduino Pro Micro at 3.3V and save power [43]. The Arduino Pro Micro connects to a common cathode RGB LED to indicate the status of the MCU.

The WDPTR IoT device has a reverse voltage and a short circuit protection circuit. It uses a P-channel MOSFET, a Zener diode and a resistor to implement the reverse voltage protection circuit. It uses a 350mA resettable fuse to protect the electronic components from a short circuit. The output of the power supply unit feeds to the RAW input of the controller unit, and the battery voltage feeds to an ADC pin of the controller through a voltage divider.

The WDPTR IoT device uses an HX711 (load cell amplifier) and a precision 24-bit ADC [44] to read the voltage of a load cell. The HX711 sends the load cell data to the controller through a data pin and a clock pin.

The WDPTR IoT device uses A39C-T400A22D1a (LoRa module), which is developed by Gisemi Microelectronics [45], to communicate with the server. The LoRa module connects two data pins, two setting pins and a status pin to the controller unit.

The electronic circuit of the server

The figure below shows the schematic of the electronic circuit for the development of the server.

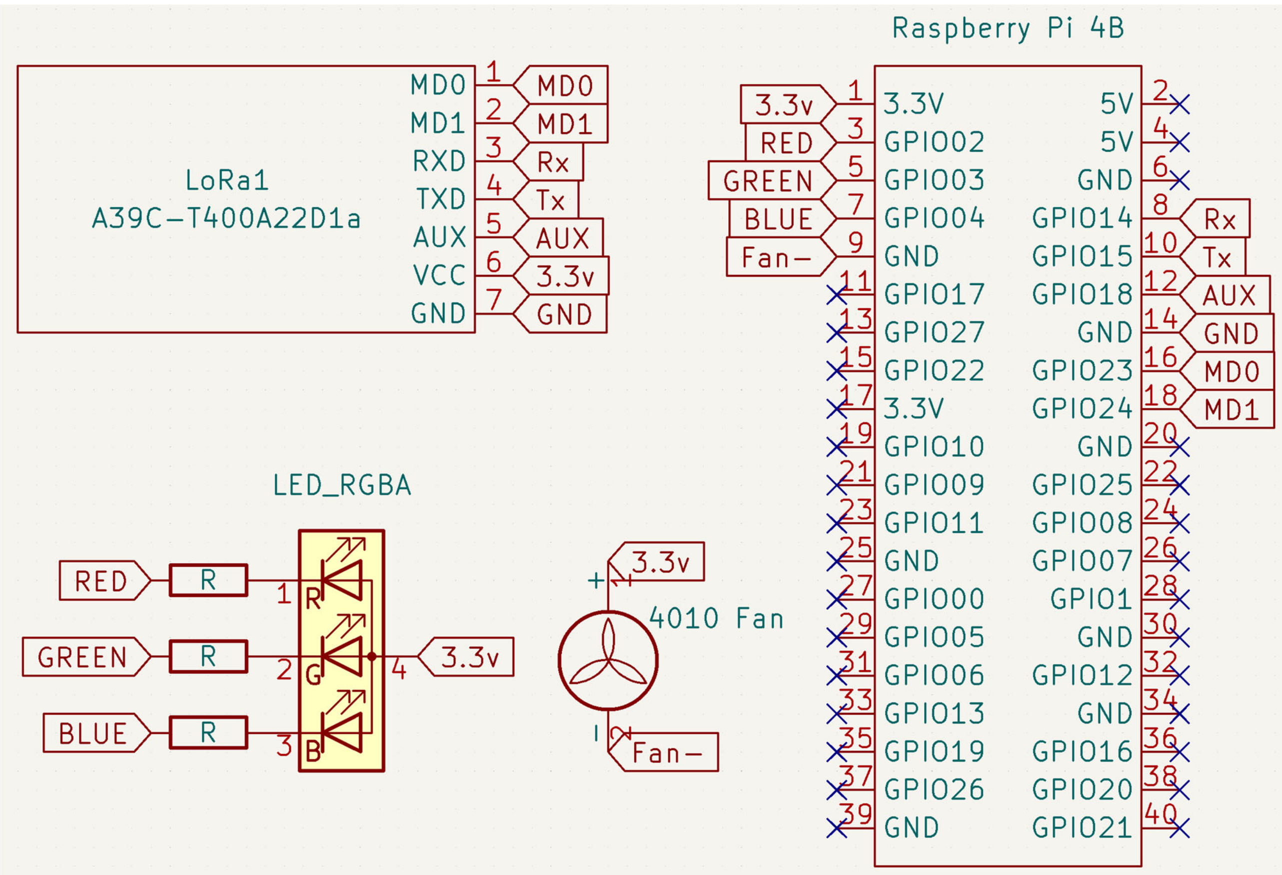

Figure 10 - Schematic of the electronic circuit for the development of the server

The WDPTR server consists of four components:

1. *A Raspberry Pi 4B* is the main computing unit of the WDPTR server. It is powered by a 5V/3A USB-C power supply. It uses a 128GB microSD card as the internal storage for the operating system and the data storage.
2. *A LoRa module A39C-T400A22D1a* is a LoRa module developed by Gisemi Microelectronics [45]. The LoRa module connects two data pins, two setting pins and a status pin to the Raspberry Pi 4B.
3. *An RGB LED* connects to three GPIO pins through resistors, and its common terminal connects to a 3.3v power source. It indicates the server's status.
4. *A 4010 fan* is a 40mm × 40mm × 10mm fan. It connects to the 3.3V pin and the ground pin of the Raspberry Pi 4B. It cools down the Raspberry Pi 4B because the Raspberry Pi 4B sometimes may overheat during operations.

The case of an IoT device

The following figure shows the 3D CAD of the WDPTR IoT device case.

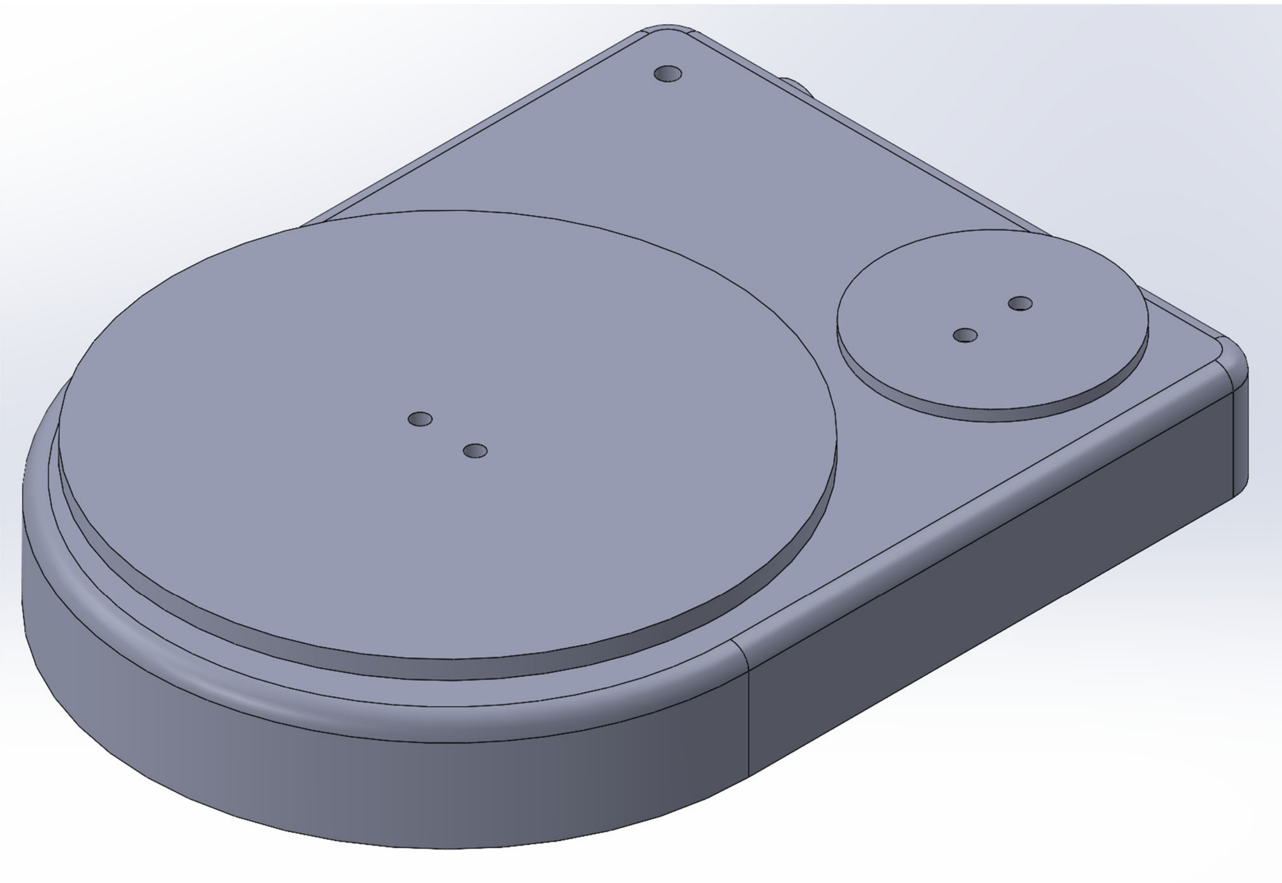

Figure 11 - 3D CAD of the IoT device case

We use FDM 3D printing to produce the case of the IoT device in acrylonitrile butadiene styrene (ABS). The plates on top are acrylic boards.

The figure below shows the front view of the IoT device after removing the plates.

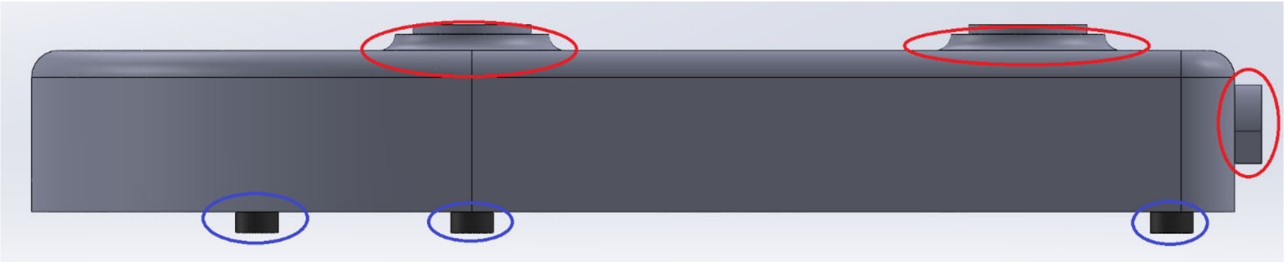

Figure 12 - 3D CAD front view of the IoT device case after removing the plates

We add some features (indicated by red and blue circles in the above diagram) in the case to prevent the water in the water container from damaging the electronic components inside the case. Note that the features circled in blue are rubber feet.

The figure below shows the bottom view of the IoT device.

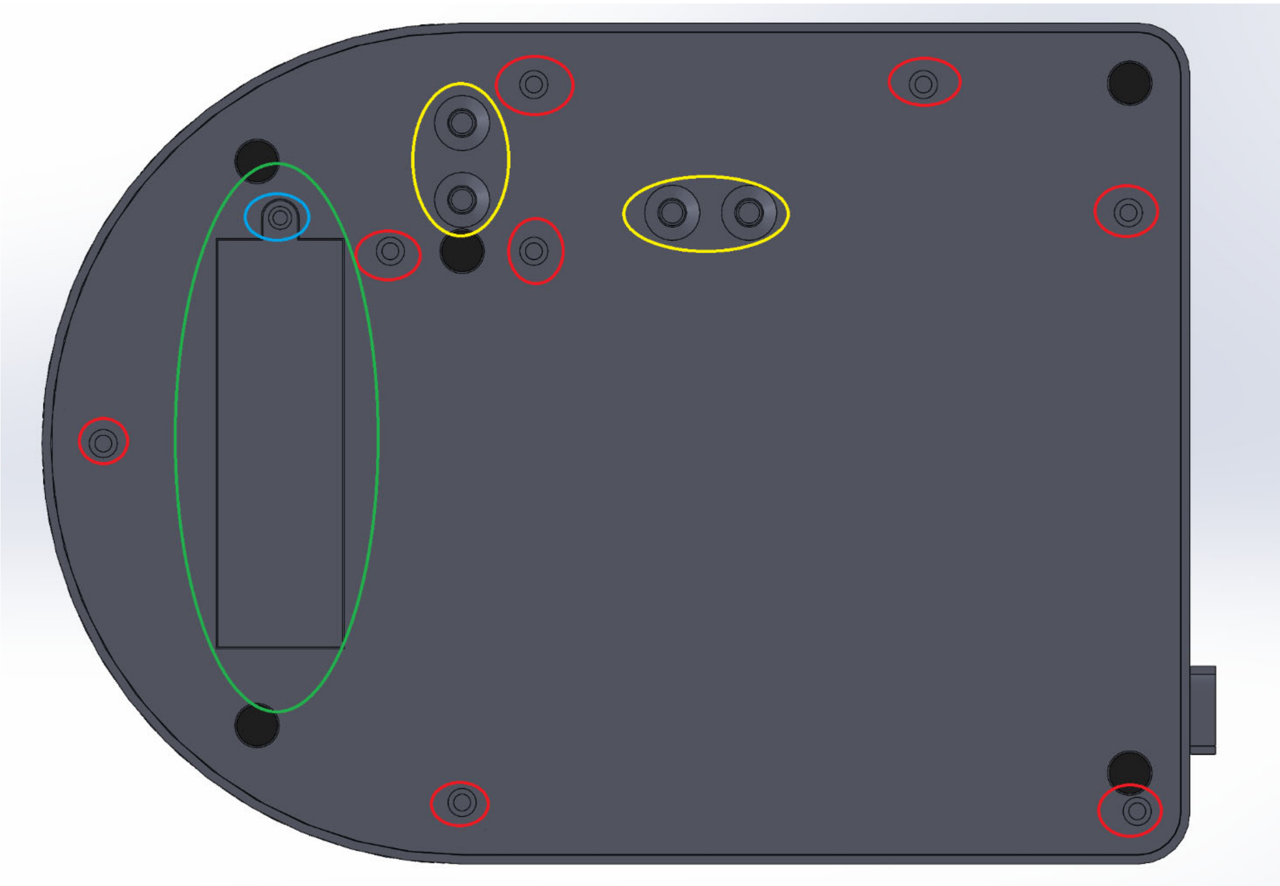

Figure 13 - 3D CAD bottom view of the IoT device

We mount the load cells to the bottom of the case through the holes circled in yellow. We mount the IoT device to the case through the holes circled in red. The cover circled in green is the battery cover. There is a battery underneath the battery cover. We use a screw to mount the battery cover to the bottom case through the hole circled in blue.

The figure below shows the top case of the IoT device, the printed circuit board assembly (PCBA) and the 18650 battery box.

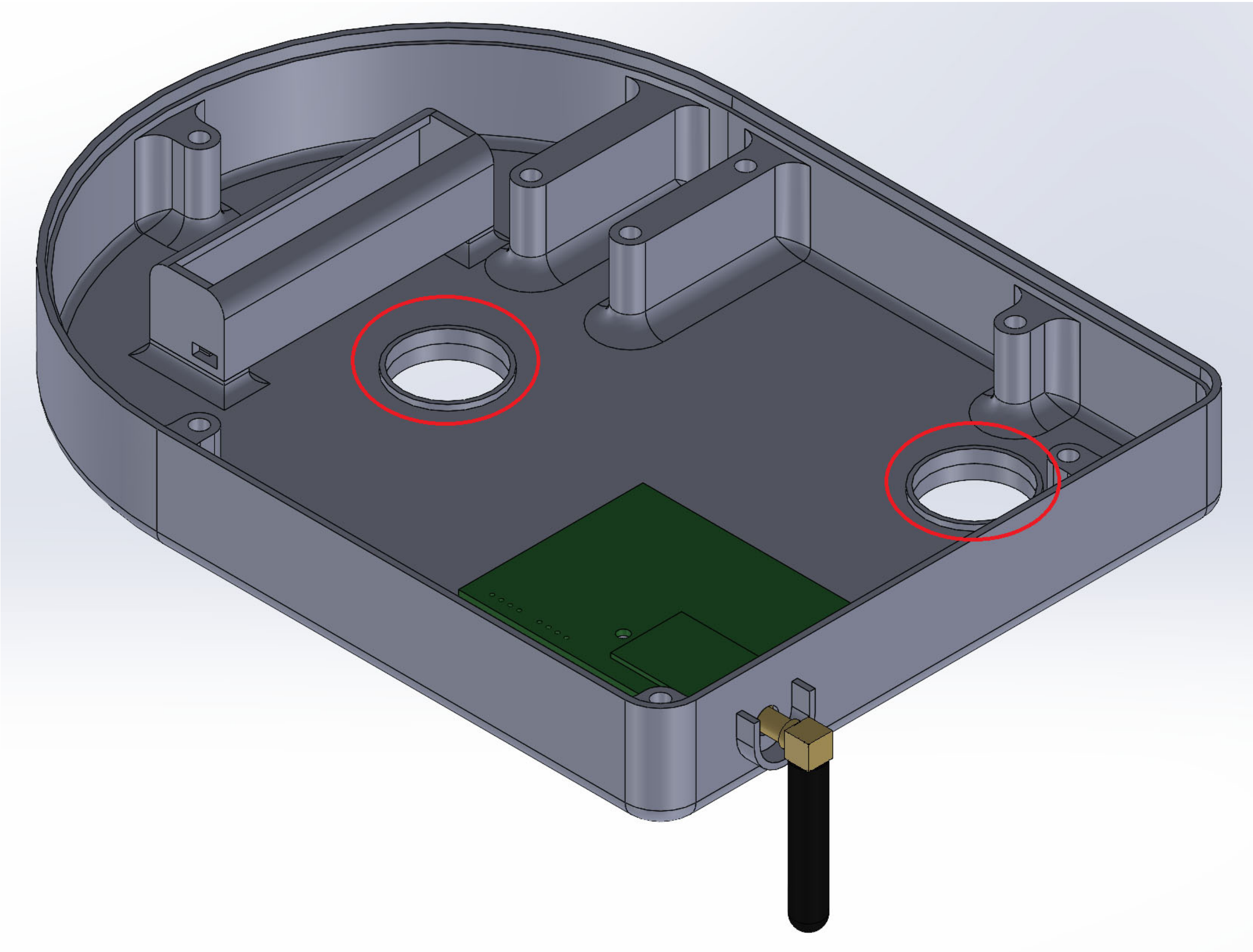

Figure 14 - 3D CAD of the top case of the IoT device,
the PCBA and the 18650 battery box

We mount the PCBA and the battery box on the top of the case to prevent the water in the water container from damaging the electronic components inside the case. The flanges circled in red allow the water to go inside the case downwards instead of hanging on the ceiling of the top of the case.

The figure below shows the bottom of the case and the location of the load cells.

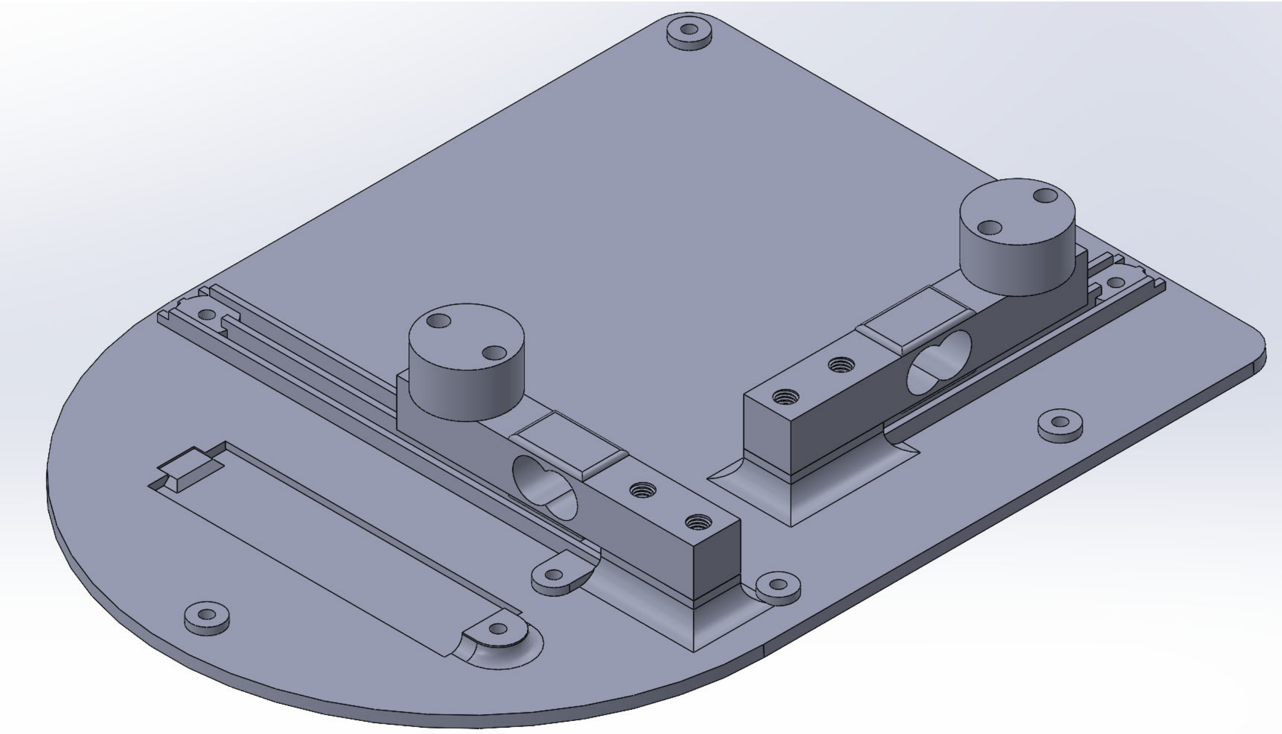

Figure 15 - 3D CAD of the bottom case
and the location of the load cells

We mount the load cells on the bottom of the case. There are reinforcement structures on the bottom to increase the strength of the case.

The case of the server

The following figure shows the 3D CAD of the WDPTR server case.

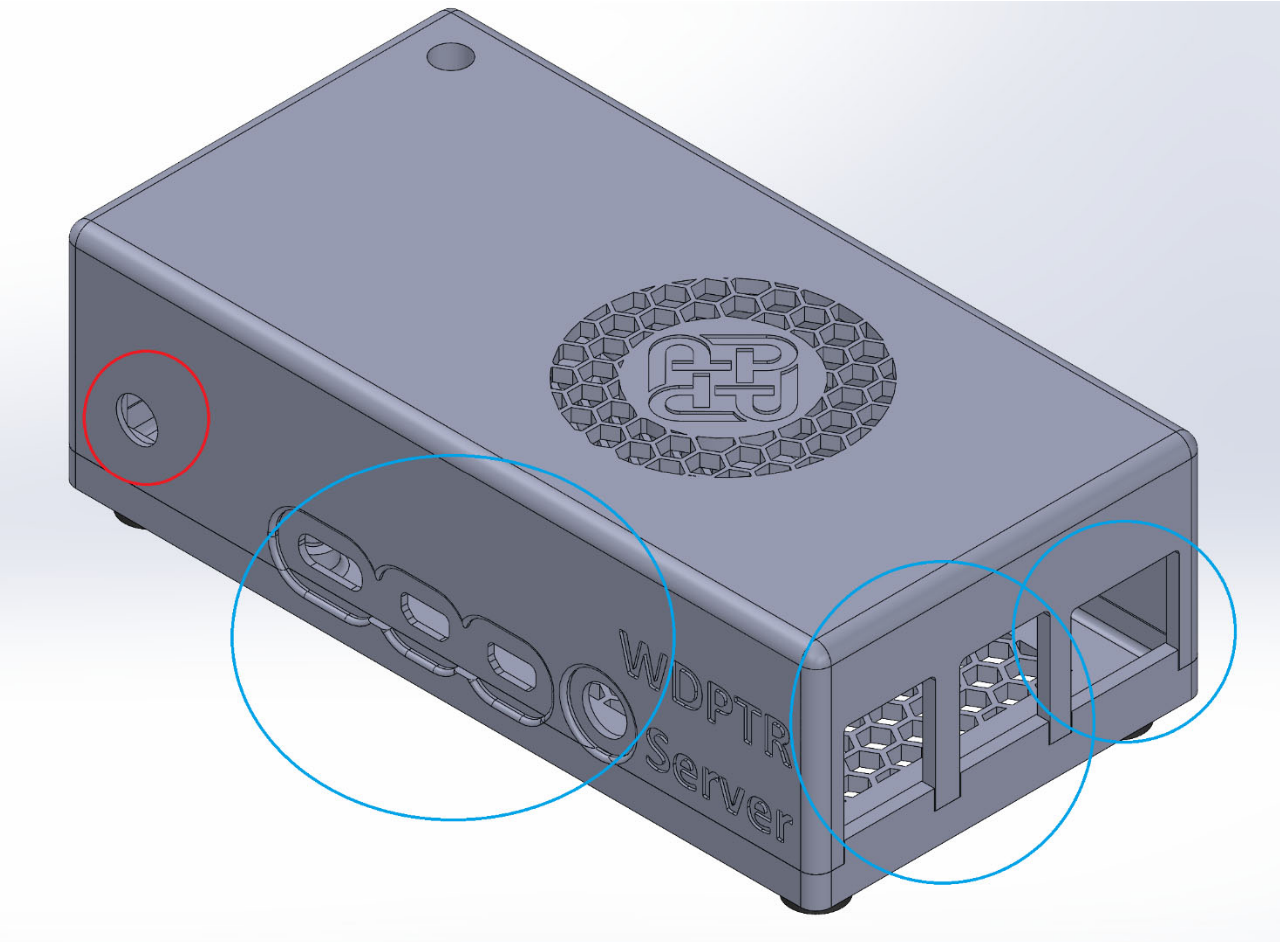

Figure 16 - 3D CAD of the WDPTR server case

We use FDM 3D printing to produce the case of the IoT device in acrylonitrile butadiene styrene (ABS).

The holes on the sides circled in blue are for input/output ports, including USB, USB-C, RJ-45, micro-HDMI and a 3.5mm jack. We mount the LoRa module through the hole circled in red.

The figure below shows the top of the case of the WDPTR server.

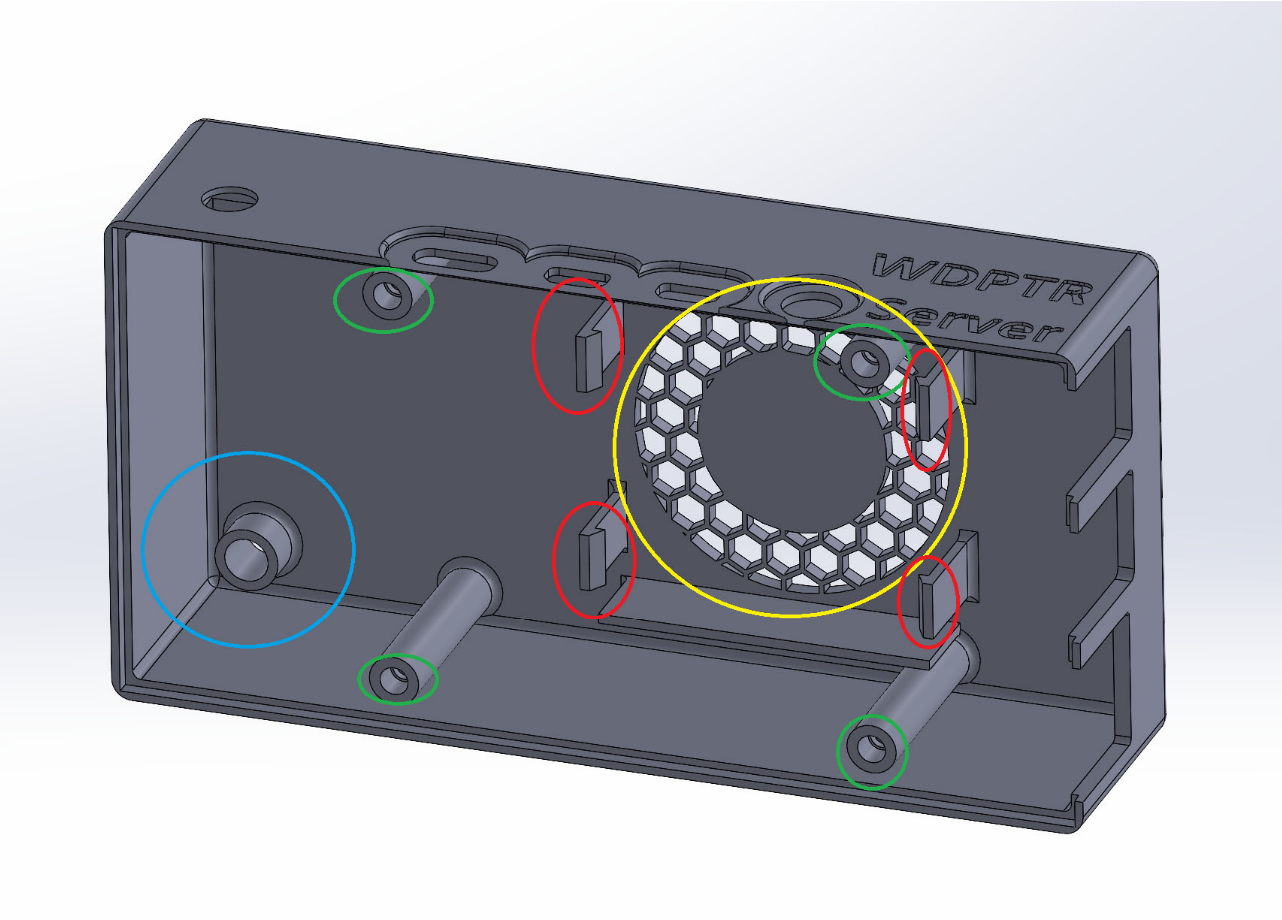

Figure 17 - 3D CAD of the top of the case of the WDPTR server

The hole circled in blue is to show the RGB LED indicator. We snap-fit the 4010 fan to the hooks circled in red. The hexagon holes circled in yellow allow the fan to inhale fresh air for cooling. We install heat set inserts into the holes circled in green to mount the bottom of the case with screws.

The figure below shows the bottom of the case of the WDPTR server.

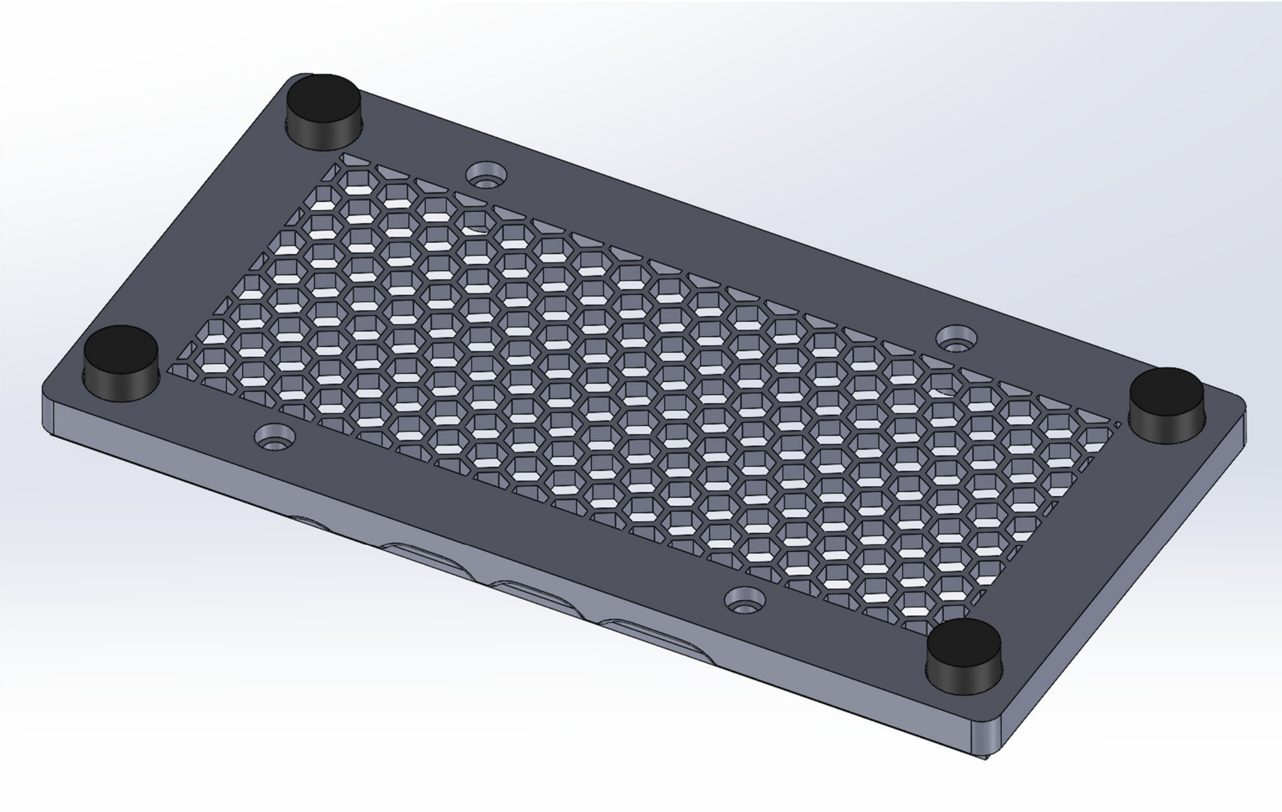

Figure 18 - 3D CAD of the bottom of the case of the WDPTR server

We install rubber feet to the bottom. It prevents the server from slipping. The hexagon holes allow hot air to be exhausted.

Software Development

The following block diagram briefly shows how the Water Drinking and Pill Taking Reminder (WDPTR) system works.

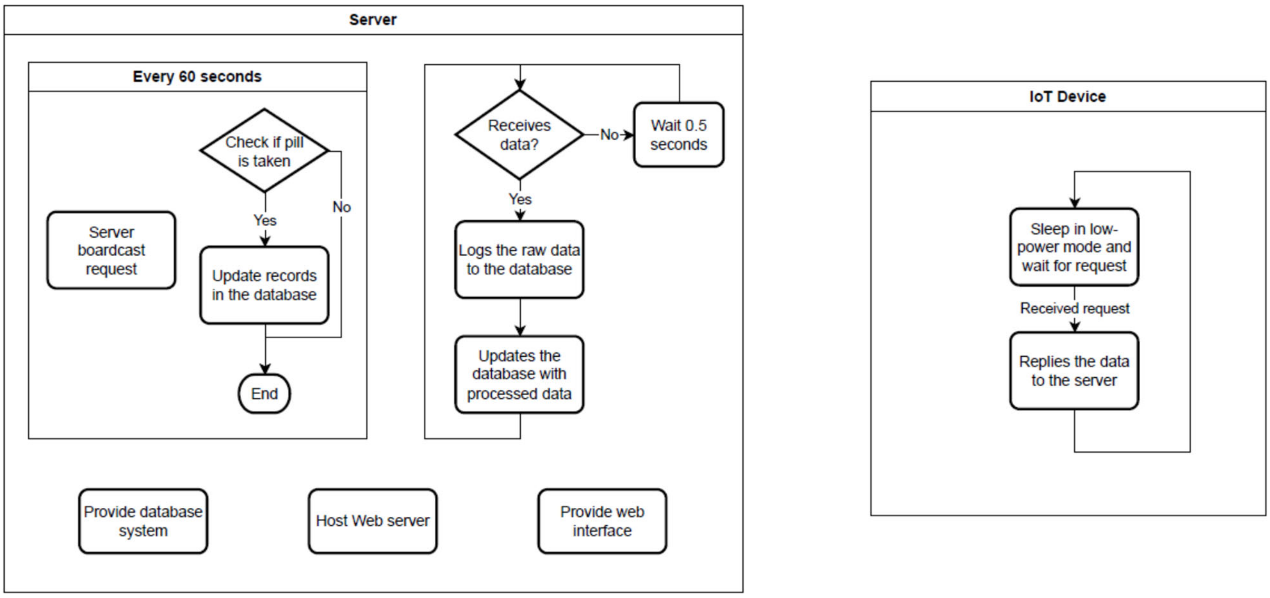

Figure 19 – A block diagram explaining how the WDPTR system works

In the sections below, we explain how the WDPTR system's software works.

The software development for an IoT device

The flowchart below shows the programme flow of the IoT device.

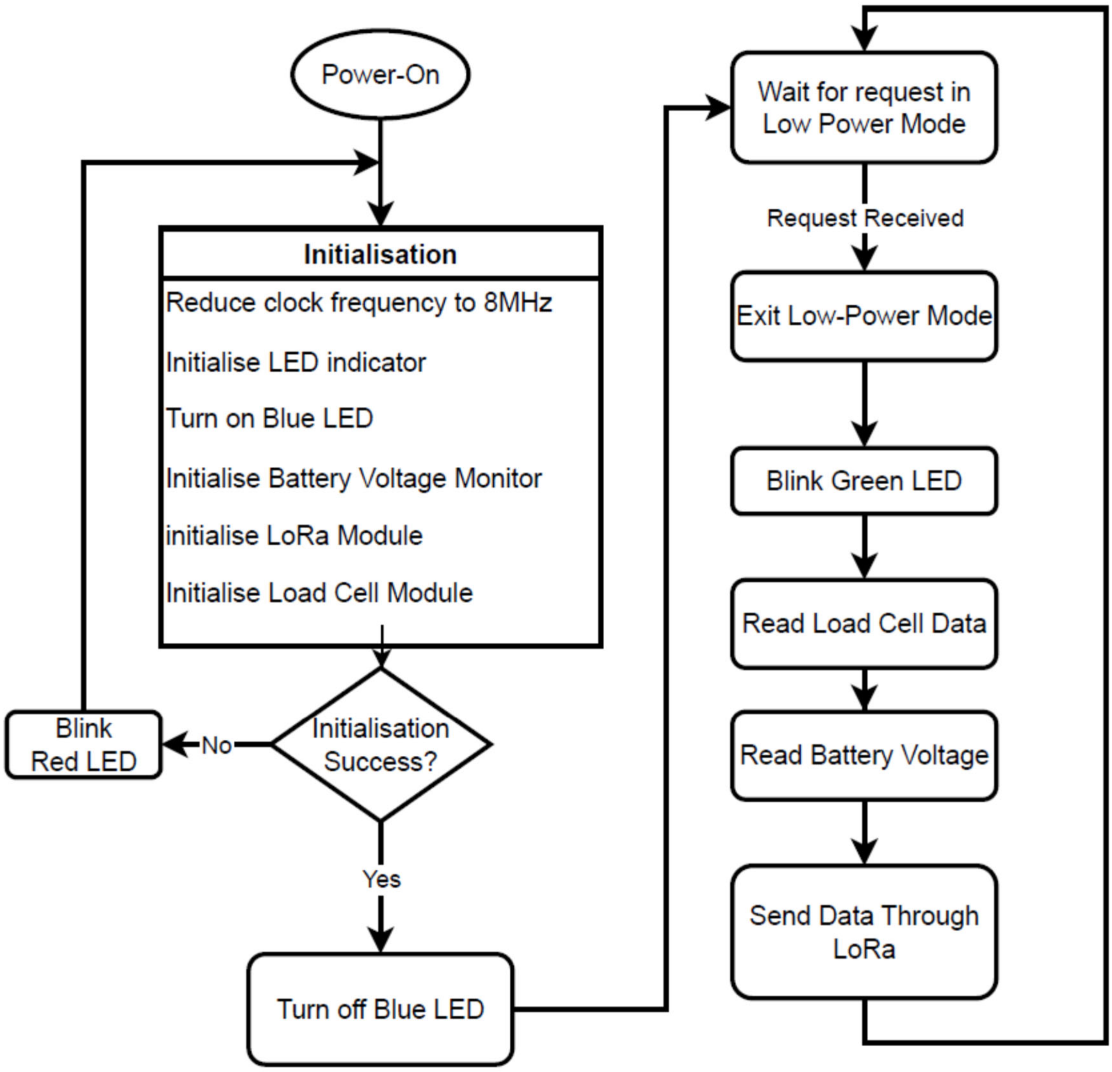

Figure 20 - Flowchart of the IoT device

The IoT device first initialises, and then provides the service.

Initialisation

There are eight steps to initialise the IoT device.

1. *Reduce the clock frequency to 8MHz*: We set the prescaler to two to reduce the clock frequency from 16MHz to 8MHz. Note that it also reduces the power consumption of the IoT device.
2. *Initialise the LED indicator*: We set the pin mode of the corresponding pins to OUTPUT.
3. *Set the LED to blue*: We set the LED to blue to indicate that the IoT device is initialising.

4. *Initialise the battery voltage monitor*: We set the battery voltage pin for further usage.
5. *Initialise the LoRa module*: We set the pin modes of the LoRa pins to the corresponding pin mode and the operating mode to the "Exit Setup Mode".
6. *Initialise the load cell module*: We assign the HX711's pins and set up the HX711 using the HX711 library. Then we set the scales of the load cells to the pre-calibrated values.
7. *Indicate whether initialisation is successful*: If the IoT device initialises successfully, turn off the LED; otherwise, set the LED to red, blink it and initialise the setup again.
8. *Switch to low-power mode*: We set the MCU, the LoRa module and the load cell module into low-power mode by calling the corresponding functions.

Providing the service

1. *Stay in the low-power mode and wait for incoming requests:* The IoT device stays in low-power mode. When it receives a request from the server, it triggers an interrupt and saves the request.
2. *Exit the low-power mode:* The MCU, the LoRa module and the load cell module exit the low-power mode.
3. *Blink green LED:* The MCU sets the LED to green and blinks it to indicate that it receives a request and responds to the request.
4. *Read the load cell data:* The MCU reads the load cell data through the HX711 using I^2C communication.
5. *Read the battery voltage:* The MCU calculates the battery voltage using the reading from the ADC.
6. *Send the data through LoRa:* The MCU rephrases the data and sends the data with the format shown below:
   "ID:{device_id}A:{load_cell_A_data}B:{load_cell_B_data}V:{battery_voltage}"
7. *Back to the low-power mode:* The MCU, the LoRa module and the load cell module are back to the low-power mode.
8. *Go back to Step 1.*

Server's software

The Raspberry Pi 4B runs a LAMP server.

- Linux: Raspberry Pi OS, Debian GNU/Linux 11 (bullseye)
- Apache: Apache/2.4.54 (Debian)
- MariaDB: Ver 15.1 Distrib 10.5.18-MariaDB
- PHP: PHP 7.4.30 (cli)
- Python: Python 3.9.2

Database

There are nine tables in the database. The entity relationship diagram (ERD) of the database below shows the structure of the database.

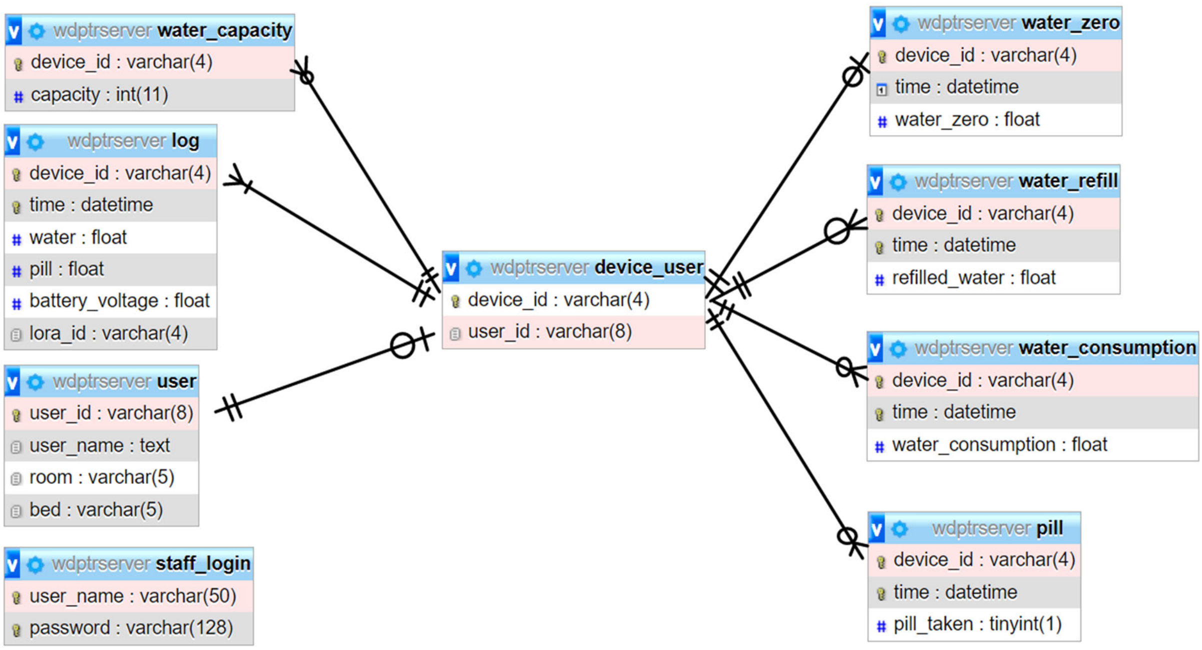

Figure 21 - ER diagram of the database

- staff_login: This table saves the login information of the web interface. The password column saves the MD5 hash of the account password.
- user: This table saves the people receiving care's information, such as their ID, their name, and their beds' location.
- device_user: This table saves the information of which person receiving care uses which IoT device.
- log: This table saves the decrypted raw data received from the IoT devices.

- water_capacity: This table saves the capacity of the water container of each IoT device.
- water_zero: This table saves the load cell reading when the water container is full. This data provides a reference for measuring the amount of water consumption.
- water_refill: This table saves the time and the amount of water being refilled into the water container.
- water_consumption: This table saves the time and the amount of the water consumption since the last refill.
- pill: This table saves the time and whether the pills are taken.

Service

The service uses inter-process communication (IPC) to communicate with other processes via the message passing method. The service follows the programme flows below to set up and check the IPC to receive the request from other processes.

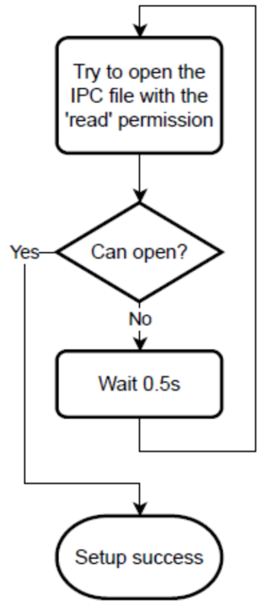

Figure 22 - Flowchart of IPC setup

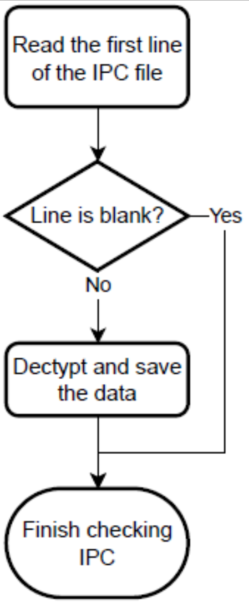

Figure 23 - Flowchart of IPC checking

The process follows the programme flow below to send a request through IPC.

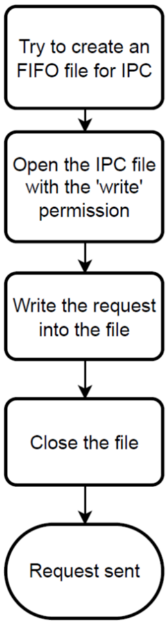

Figure 24 - Flowchart of sending request through IPC

When the service accesses the database, it follows the programme flow below.

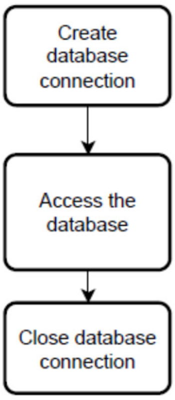

Figure 25 - Flowchart of accessing the database

There are three services running on the server using systemctl [46].

- wdptrserver_service:
  This service in charge of everything about LoRa. It sends requests and receives data through LoRa. The diagram below shows the structure of this service.

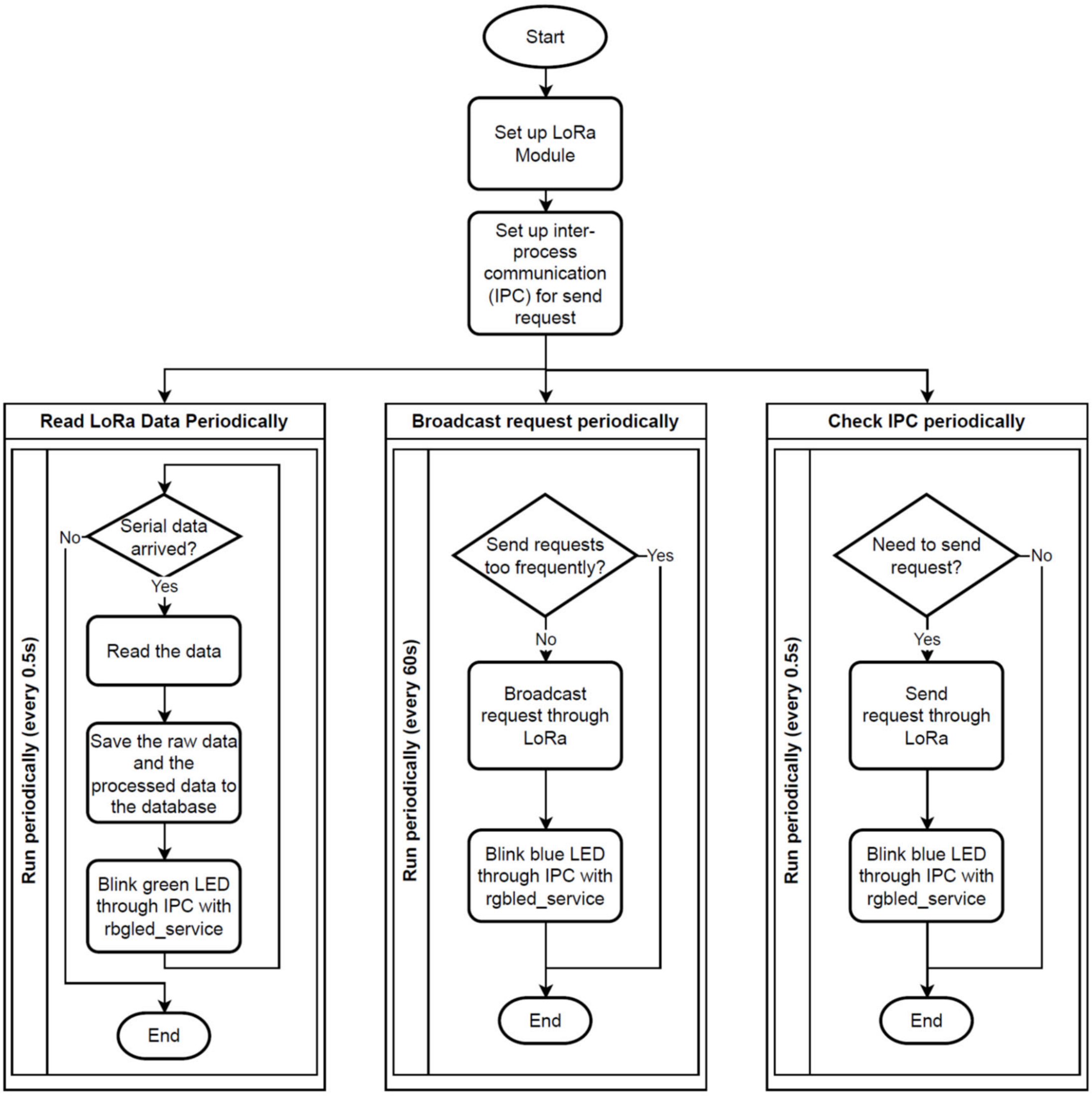

Figure 26 - Flowchart of wdptrserver_service

- rgbled_service

  This service provides a platform for other processes to blink the LED using IPC. The diagram below shows the programme flow of this service.

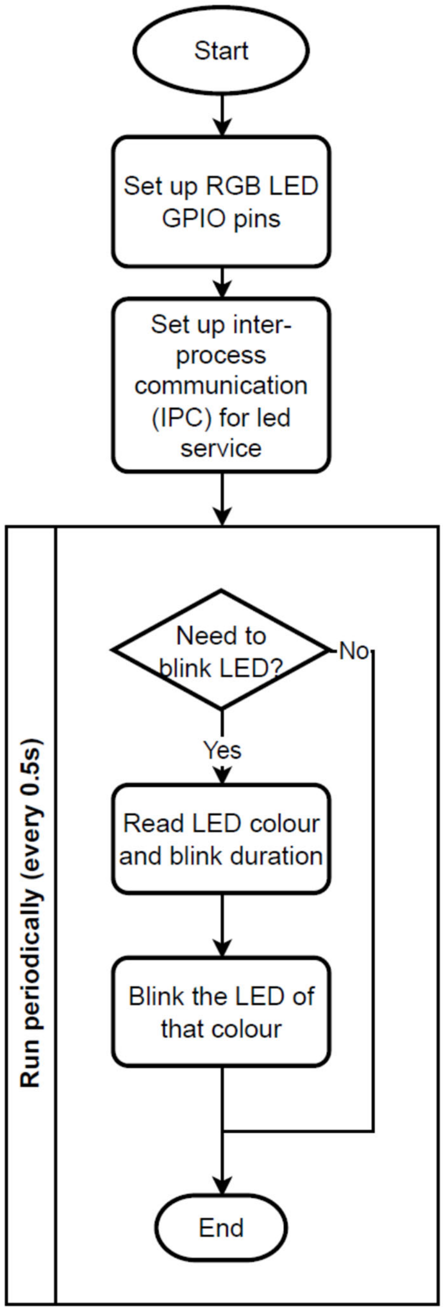

Figure 27 - Flowchart of rgbled_service

- check_pill_service

  This service checks whether the pills are taken and logs the data to the database if necessary. The diagram below shows the programme flow of this service.

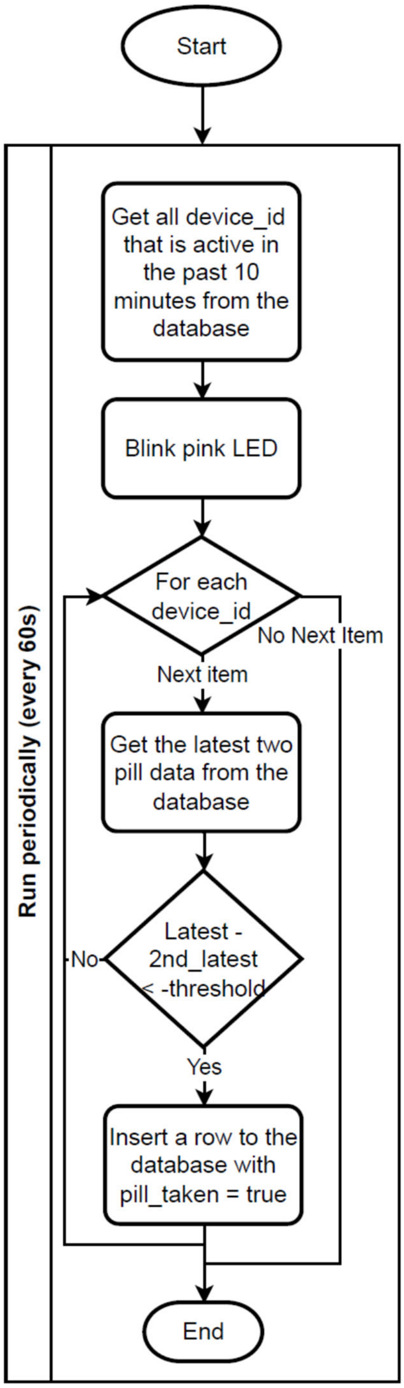

Figure 28 - Flowchart of check_pill_service

Web Interface

The diagram below shows the structure of the web interface.

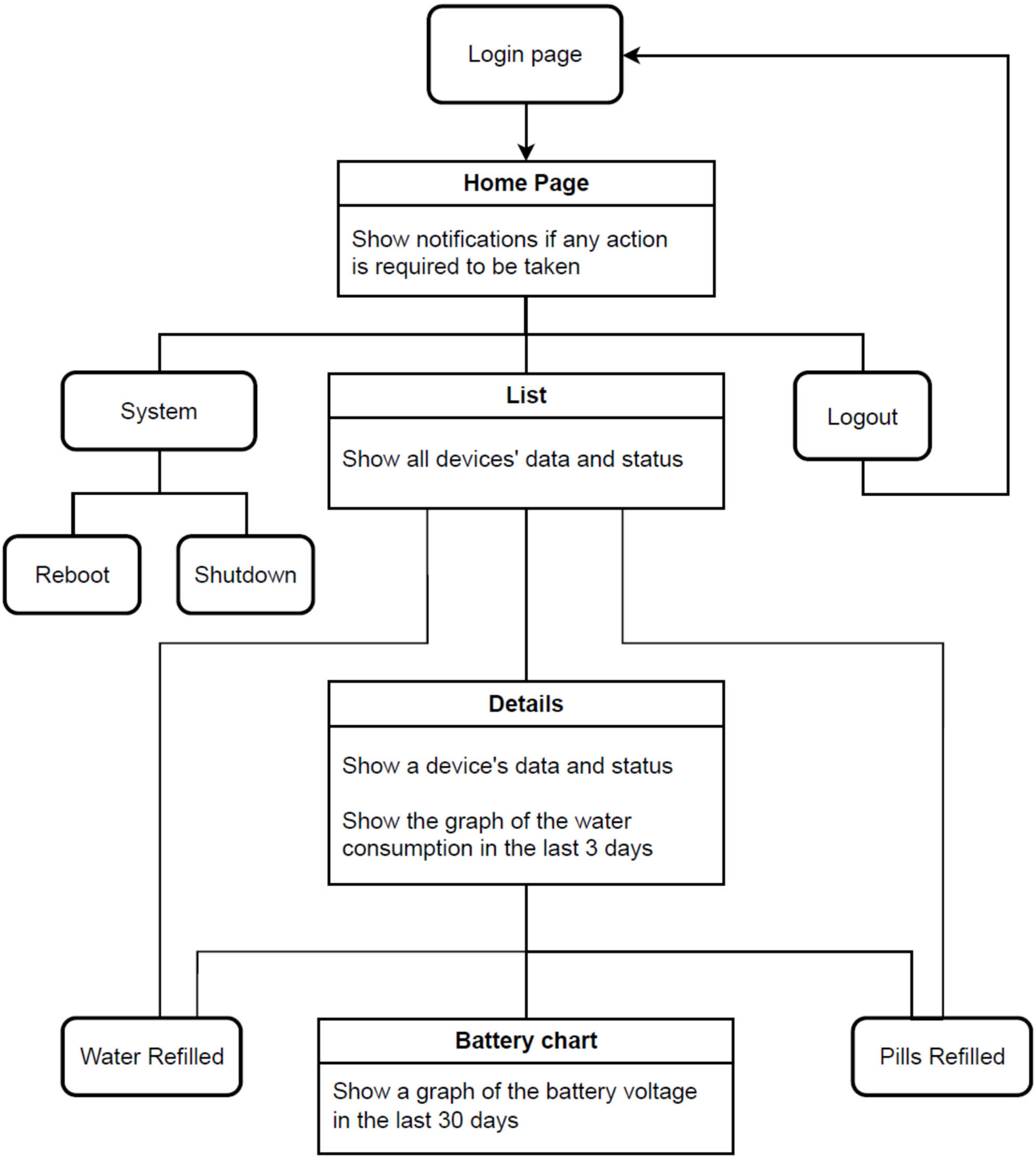

Figure 29 - A diagram showing the structure of the web interface

The web interface is implemented by the following PHP files.

- *index.php:* This page first checks if the web browser logs in. If yes, it directs to the home page. If not, it prompts text fields for a username and password. When the user clicks the login button, it pushes the username and password to login.php.

Figure 30 - Screen capture of the login page on a PC

- *login.php:* This file checks if the username and password match the records in the database. If yes, it considers the browser as logged in by setting the $_SESSION['user_name'] = the user's username and directs to the home page. If not, it tells the user the username and password are wrong.

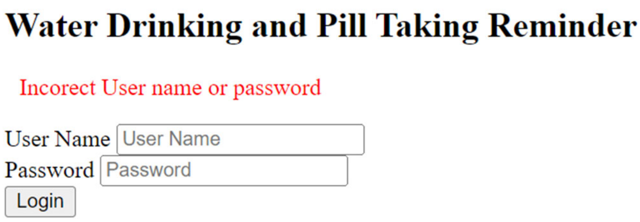

Figure 31 - Screen capture of the login page on a PC after wrong username or password is submitted

- *navbar.php:* It contains the navigation bar of the web interface. Every page of the web interface except the index includes this PHP file at the beginning.

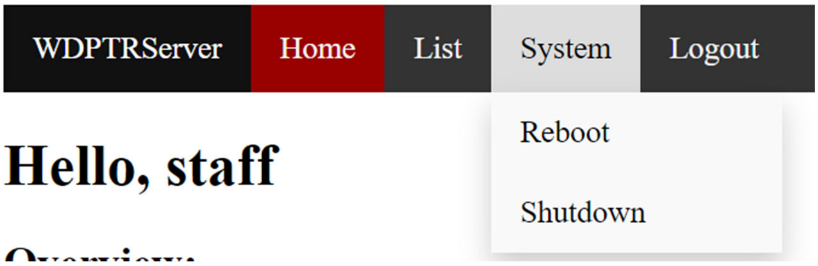

Figure 32 - Screen capture of the navigation bar on a PC

- *homepage.php:* It uses SQL to get the data from the database and uses PHP to check if any action is required to be taken, such as:
    - staffs need to refill water for certain people receiving care
    - staffs need to remind certain people receiving care to take the pill
    - staffs need to change the battery for specific IoT devices.

  If any action is required, it shows the notification on the home page. If not, it shows everything is good.

  This page refreshes automatically every 10 seconds.

Figure 33 - Screen capture of the home page on a PC when actions are required

Figure 34 - Screen capture of the home page on a PC when no actions are required

- *reboot.php:* It prompts a prompt box to ask for the user's confirmation. If the user confirms, it sends a command to reboot the system.

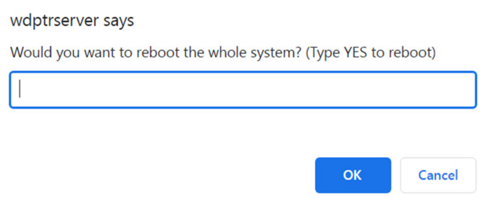

Figure 35 - Screen capture of the reboot prompt box on Chrome, Windows version

- *shutdown.php:* It prompts a prompt box to ask for the user's confirmation. If the user confirms, it sends a command to shut down the system.

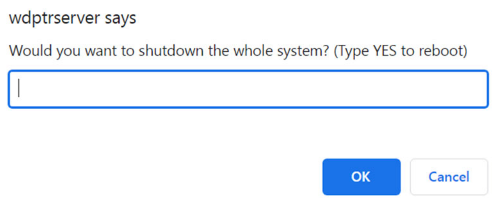

Figure 36 - Screen capture of the shutdown prompt box on Chrome, Windows version

- *logout.php:* It logs out of the browser by un-setting and destroying the session.
- *list.php:* It uses SQL to read the most recent data from the database and uses HTML code to display the data in a table. It provides the staff buttons to set "water refilled" and "pill refilled" for each person receiving care and to check the details page of a device by directing to the details.php. It also provides a button for refreshing all data.
  This page refreshes automatically every 10 seconds.

Latest Records (active in last 24 hours)

Time	Device ID	User ID	User Name	Room	Bed	Water Consumed	Pill Taken	Battery Voltage	Water Refilled?	Pills Refilled?	Details
2023-03-27 22:15:55	E1E0	20220001	Alpha the First user	cf502	23	5 mL	No	3.82 V	Water Refilled	Pills Refilled	Details
2023-03-27 22:15:54	E1E1	20220002	Beta the Second user	cf502	24	488 mL	No	3.94 V	Water Refilled	Pills Refilled	Details

Last update: 2023-03-27 22:16:25

Update

Figure 37 - Screen capture of the list page on a PC

- *details.php:* It uses SQL to read the data of a device from the database and uses HTML code to display the data in a table. It also shows the "water refilled" button and the "pills refilled" button and shows a chart of the water consumption over the past three days using CanvasJS. It allows the staff to change the water container capacity in this page by prompting a prompt box and updating the database using SQL with the inputted data in the prompt box. This page can direct to the battery_chart.php.

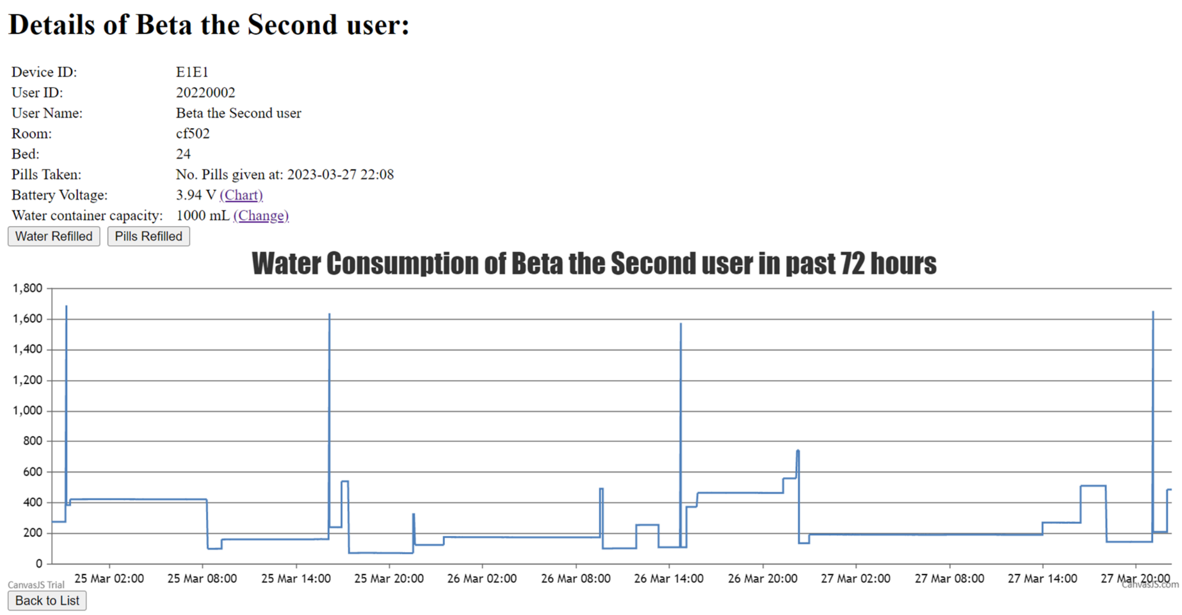

Figure 38 - Screen capture of the details page on a PC

- *batter_chart.php:* It uses SQL to read the battery voltage data of a device from the database and uses CanvasJS to show a chart of the battery voltage of a device in the past 30 days.

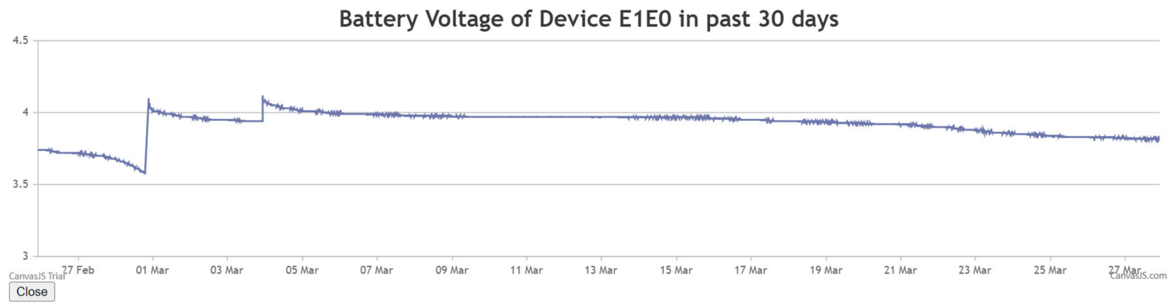

Figure 39 - Screen capture of the battery chart on a PC

- *refill_water.php:* This file uses SQL to update the water information in the database.
- *refill_pill.php:* This file uses SQL to update the pill information in the database.

We use Viewport [47] to design an interactive web interface that suits for both PC and smartphones.

LoRa module's settings

The figure below shows the settings of the LoRa module. The upper one is the server's setting. The lower one is the IoT device with the device_id = "E1E0" 's settings.

Figure 40 - Server's LoRa module's settings

Figure 41 - IoT device's LoRa module's settings with ID = "E1E0"

We set the LoRa modules to be 9600 baud, with a 4800kbps transmission rate. They transmit data with eight data bits, one stop bit and no parity check. They are in Master-Slave mode. The server is the master, and the IoT devices are the slaves. The master can communicate with all slaves, while the slaves can only communicate with the master. We set the sleeping period to four seconds to save power while keeping a reasonable response time.

Performance Investigation

We produce two prototype IoT devices and one server. Device A uses the old version case and device B uses the latest version case. Both IoT devices use same electronic components, electronic circuit and algorithm.

In this section, we illustrate the performance data of the Water Drinking and Pill Taking Reminder (WDPTR).

IoT device's power consumption

We use different approaches to reduce the power consumption of the IoT device. The average power consumption reduces by 98.5% from $3.7V \times 104.51mA \approx 386.69mW$ to $3.7V \times 1.55mA \approx 5.74mW$.

We use a multi-metre (HIOKI 3256-50) to measure the average current drawn from a 3.7V power source over a 15-minute period while the IoT device is providing service. We calculate the average power consumption using Ohm's Law: $Power = Voltage \times Current$.

The table below shows the power reduction of different methods:

Approach	**Power reduced (in mW) (average of 3)**
Arduino board on-board LED removal (@16MHz)	$3.7V \times 0.27mA \approx 1mW$
Using common cathode LED instead of common anode LED (@16MHz)	$3.7V \times 0.04mA \approx 0.15mW$
Utilise LoRa's power-saving mode (@16MHz)	$3.7V \times 24.6mA \approx 91.02mW$
Utilise HX711's power-saving mode (@16MHz)	$3.7V \times 16.6mA \approx 62.53mW$
Implement Arduino LowPower.powerDown (@16MHz)	$3.7V \times 59.8mA \approx 221.3mW$
Replace on-board voltage regulator with a 3.3V regulator and run on 8MHz, eliminating the need for a power converter to step up the 3.7V 18650 battery to 5V	$3.7V \times 1.7mA \approx 6.3mW$

Table 1 - Power reduction of different approaches

IoT device's battery life

Assuming the 18650 battery has 1500mAh, which is equivalent to $3.7V \times 1500mAh = 5550mWh$. Therefore, the IoT device has a theoretical battery life of: $5550mWh \div 5.74\ mW \div 24hr \approx 40\ days$

In an experiment, the voltage of a brand new, fully charged 1500mAh 18650 battery that is installed and powering the IoT device drops from 4.1V to 3.65V in 34.4 days.

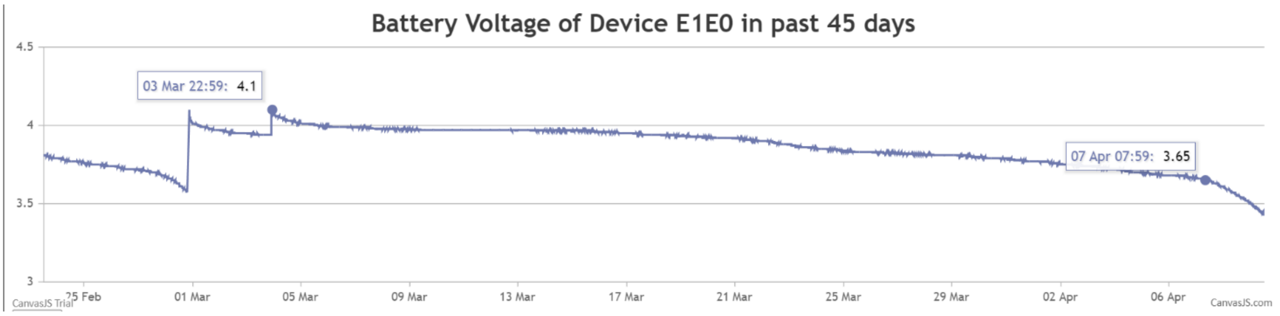

Figure 42 - Line graph of battery voltage of device E1E0 in the past 45 days

IoT device's response rate

The IoT devices have a response rate between 96.5% to 99.6%.

Total number of requests sent by the server: 43994		
Device A	Number of replies: 42434	96.4541%
Device B	Number of replies: 43617	99.5704%

Table 2 - Response rate of two IoT devices

Water scale accuracy

The water scale has an accuracy of 0.1 grams.

The scale drift is less than 0.2 grams at most of the time.

Device A's water scale data shows that 94.08% of readings exhibit a drift of ±0.2 grams or less per minute. Device B's water scale data shows that 99.38% of readings exhibit a drift of ±0.2 grams or less per minute.

Device A's water scale accuracy is significantly lower since it uses the older version case. The old version case deforms due to the insufficient structural strength. The deformation affects the reading and compromises the overall accuracy of the water scale.

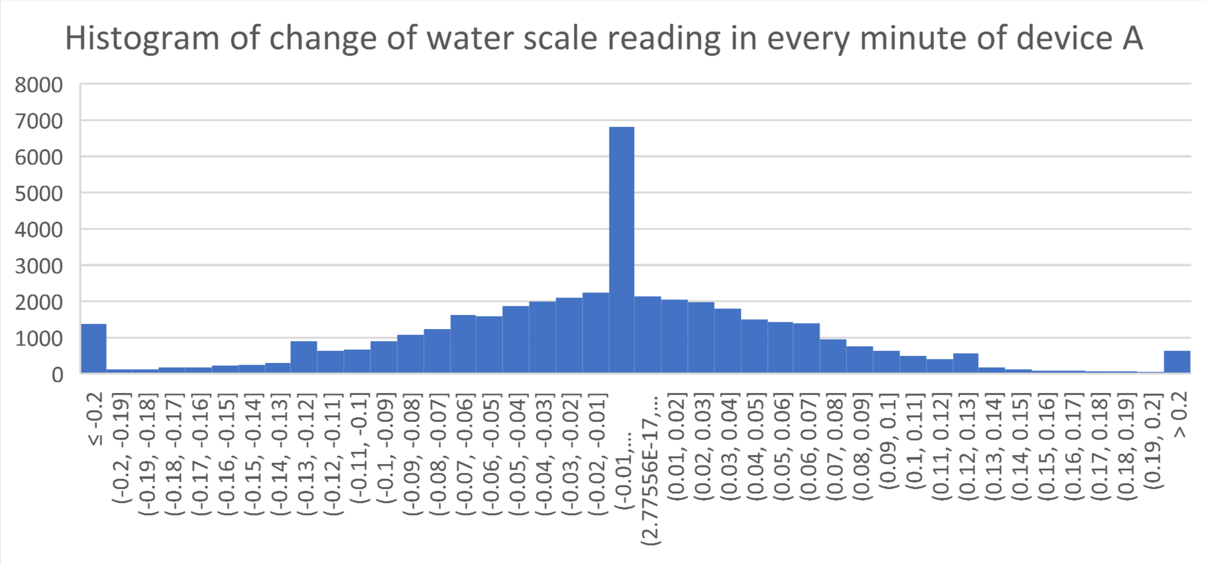

Figure 43 - Histogram of change of water scale reading in every minute of device A

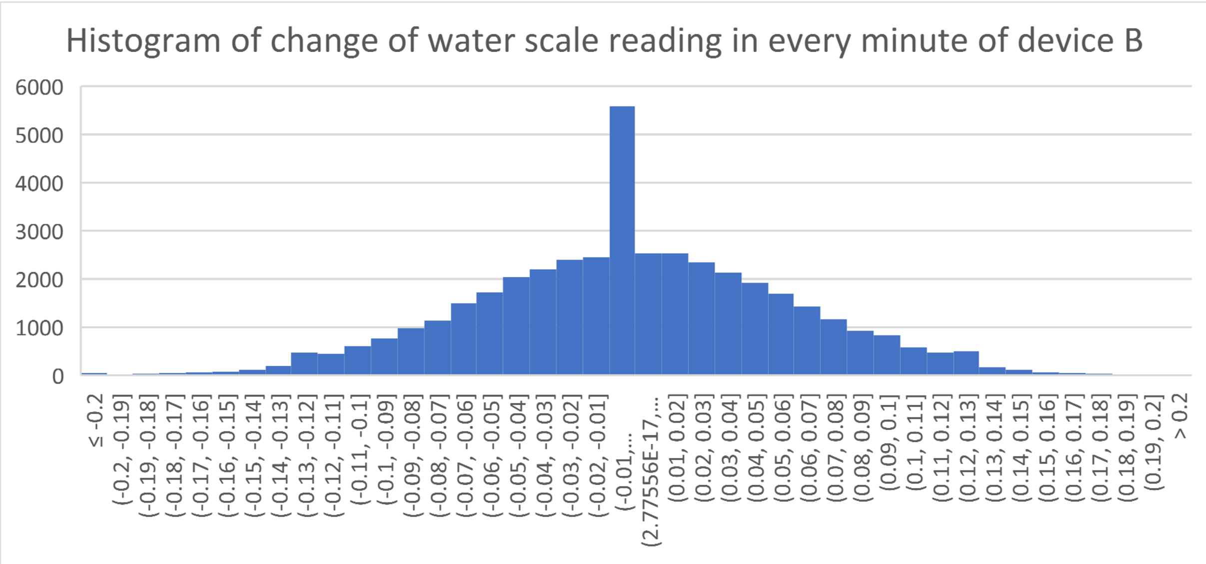

Figure 44 - Histogram of change of water scale reading in every minute of device B

Pill scale accuracy

The pill scale has an accuracy of 0.05 grams.

Device A's pill scale data shows that 99.90% of readings exhibit a drift of ±0.15 grams or less per minute, and 99.991% of readings exhibit a drift of ±0.3 grams or less per minute. Device B's pill scale data shows that 99.85% of readings exhibit a drift of ±0.15 grams or less per minute, and 99.962% of readings exhibit a drift of ±0.3 grams or less per minute.

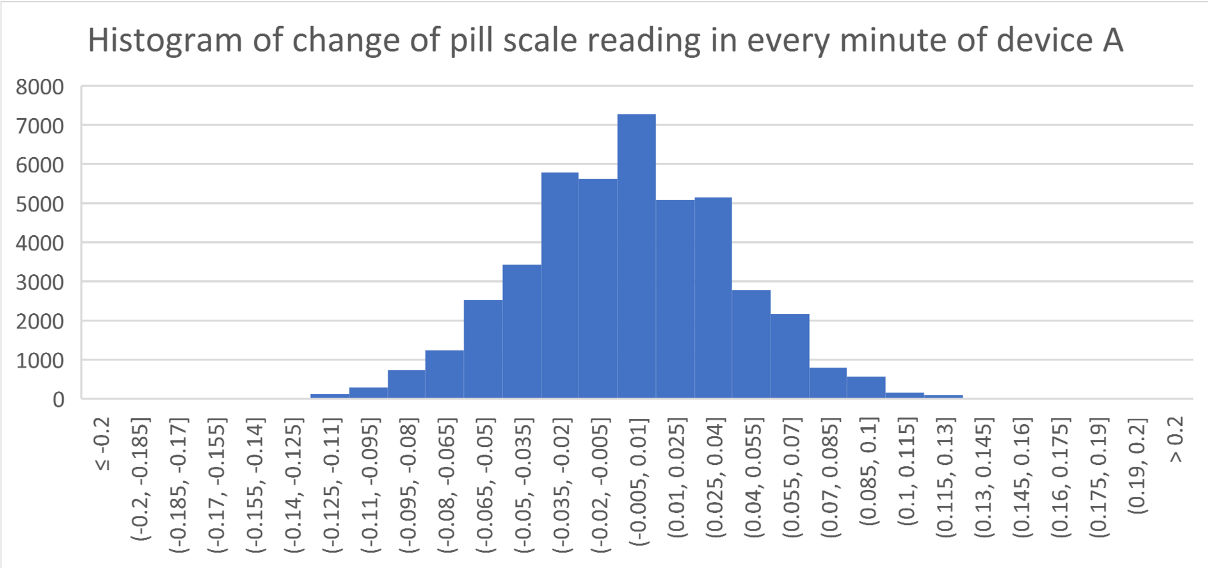

Figure 45 - Histogram of change of pill scale reading in every minute of device A

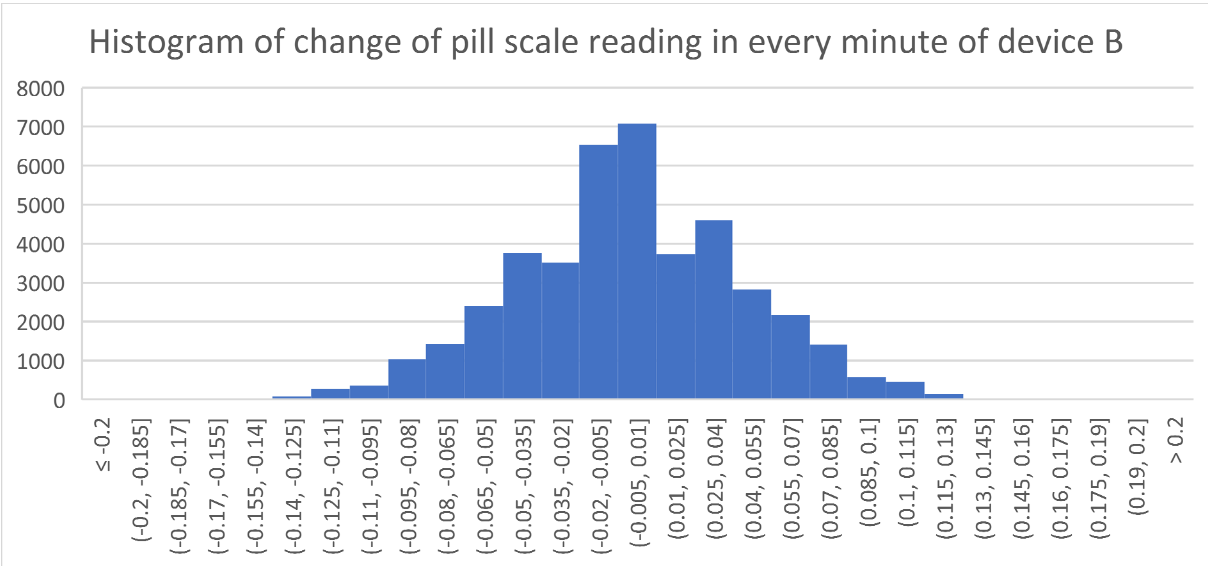

Figure 46 - Histogram of change of pill scale reading in every minute of device B

LoRa's response time

We set the LoRa module's sleep-time settings to $(28)_{16} \times 100ms = 4\ seconds$. It takes 4 × 2 = 8 seconds theoretically for the server to receive the reply from the IoT device after sending a request. In the experiment, it takes around 10.25 ± 0.75 seconds to receive a reply.

Water reminder accuracy

The water reminder has 100% accuracy since it compares the latest water reading to the water container capacity.

Pill reminder accuracy

The following table shows the definition of true/false positive/negative.

	The pill is taken	The pill is not taken
The pill reminder reports the pill is taken	True-positive	False-positive
The pill reminder reports the pill is not taken	False-negative	True-negative

Table 3 - Definition of true/false positive/negative

Pill weight: 0.6 grams, pill consumption threshold: 0.3 grams

In the experiment, the pill reminder has a 100% true positive accuracy based on 30 tests. The pill reminder reports two false-positives in 7 days.

Pill weight: 0.5 grams, pill consumption threshold: 0.25 grams

In the experiment, the pill reminder has a 100% true positive accuracy based on 30 tests. The pill reminder reports three false-positives in 7 days.

Pill weight: 0.4 grams, pill consumption threshold: 0.25 grams

In the experiment, the pill reminder has a 93.3% true positive accuracy based on 30 tests. The pill reminder reports three false-positives in 7 days.

Pill weight: 0.4 grams, pill consumption threshold: 0.2 grams

In the experiment, the pill reminder has a 96.6% true positive accuracy based on 30 tests. The pill reminder reports six false-positives in 7 days.

Discussion

In this section, we justify the decisions that we have made and describe the difficulties and solutions to the problem that we have encountered in the development process of the Water Drinking and Pill Taking Reminder (WDPTR).

Decision justification

Choosing LoRa as the wireless communication technology

There are some alternatives for wireless communication technology between the server and IoT devices. It includes Wi-Fi, Bluetooth, NB-IoT and LoRa [48].

The transmission range of NB-IoT and LoRa is much larger than Wi-Fi and Bluetooth. NB-IoT and LoRa urban transmission ranges are typically around or more than 1km [18] [49], while Wi-Fi and Bluetooth are usually less than 50m [50] [51]. As the WDPTR aims to provide services for hospital wards and elderly homes, which usually have a large coverage area, we do not choose Wi-Fi or Bluetooth.

LoRa consumes less power than NB-IoT. According to the information provided by Ubidots [52], the power consumption of LoRa is 60% less than NB-IoT. As the WDPTR IoT device is battery-powered, we choose LoRa to reduce the power consumption.

Choosing LoRa module A39C-T400A22D1a

There are some alternatives for choosing a LoRa module. We consider A39C-T400A22D1a because:

- An A39C-T400A22D1a has a simple interface. It uses UART to communicate with the connected device. It is easy to implement compared to I^2C and SPI.
- An A39C-T400A22D1a is an affordable option. It costs RMB 20 [53], which is around 40% cheaper than other LoRa modules [54].
- The transmission range of an A39C-T400A22D1a is up to 5km [45]. It is good enough to support WDPTR.

Choosing Arduino as the MCU of the IoT device

We choose Arduino because:

- Arduino is simple for the design and implementation due to its user-friendly IDE and Arduino language. It saves development time and effort.
- Arduino has a larger community. There are plenty of libraries and resources to support its use.

- The libraries and the Arduino language help reduce the code complexity.

Although STM32 is usually more powerful than Arduino, the processing power of Arduino is adequate for an WDPTR IoT device. Therefore, we choose Arduino but not STM32.

Choosing an 18650 battery as the power source of the IoT device

An IoT device is placed beside the person receiving care. We choose to use a battery as the power source since it has a great flexibility and mobility compared to a wired power source.

18650 battery is rechargeable and removable. The rechargeability reduces the cost of purchasing disposable batteries and reduces the environmental impact of the WDPTR system. The removability allows the staff to replace the used battery with a charged battery instead of removing and recharging the whole IoT device.

18650 battery is easy to purchase from a variety of sources. The administrators can buy the battery from different sources.

Choosing Raspberry Pi 4B as the server

There are some alternatives for the server, such as PCs, laptop computers and other single-board computers.

We choose a Raspberry Pi 4B as the server because:

- A Raspberry Pi 4B is more affordable than a PC and laptop computer. A Raspberry Pi 4B MSRP costs USD$35 [55].
- A Raspberry Pi 4B has a built-in Wi-Fi and some GPIO pins. Some PCs has a built-in Wi-Fi on the motherboard. However, almost no PC has built-in GPIO pins for communicating with the LoRa module.
- A Raspberry Pi 4B has a smaller size than PC and laptop computer. It does not occupy as much space as a PC or laptop computer.
- A PC or laptop computer does not support Linux but a Raspberry Pi 4B has an official image for the Linux OS.

We choose Raspberry Pi 4B over other single-board computers since it is a well-known brand in terms of single-board computers, which makes it a reliable and trusted choice. It has a large community and online forums. We can get resources, support and troubleshooting tips from them.

Choosing Linux over Windows as the server operating system

There are two main reasons for choosing Linus over Windows as the server operating system:

- *Cost*: Linux is an open-source operating system. We can use it for free. However, Windows requires a license and has associated costs. We use Linux over Windows to make the project more cost-effective.
- *Stability*: Linux is more stable and reliable than Windows since Linux has a Unix-based design. Moreover, Windows may require more frequent updates and maintenance.

Choosing 3D printing as the case's fabrication technique

There are three main reasons for choosing 3D printing as the case's fabrication technique:

- *Customisability*: 3D printing can produce plastic parts in almost any shape. This means we can design the case to fit in any specific dimensions and components of the WDPTR project.
- *Cost-effectiveness*: 3D printing is cost-effective for the small-scale production of customised parts. This means we can produce the WDPTR case at a lower cost compared to other fabrication techniques, such as CNC machining or injection moulding.
- *Short production period*: 3D printing allows us to fabricate the case fast. For example, it takes less than seven hours to print the whole IoT device case. This is useful for prototyping and developing a product by reducing the time cost.

Choosing ABS as the 3D-printed case material

Acrylonitrile butadiene styrene (ABS), polylactic acid (PLA) and polyethylene terephthalate glycol (PETG) are some common 3D printing materials. We choose ABS over the other two materials since:

- ABS deforms the least over time compared to PLA and PETG [56].
- ABS is the most heat resistant compared to PLA and PETG. ABS has the highest glass transition temperature among them [57].

Difficulties and solutions

A39C-T400A22D1a (LoRa module) setup

An A39C-T400A22D1a requires setup through UART Tx and Rx pins. We tried Arduino Uno and Arduino Pro Micro to be the computer's serial interface. However, the A39C-T400A22D1a did not respond to the setup request sent from the A39C-T400A22D1a's software. We used Serial Port Monitor [58] to ensure that we sent a setup request to the A39C-T400A22D1a, but there is still no response.

Solution: We start the serial monitor in Arduino IDE and then close it. Then, we can use A39C-T400A22D1a's software to set up the A39C-T400A22D1a.

Battery life

The IoT device consumed 387mW at the early development stage. That means the battery life of the IoT device is less than one day. The WDPTR system does not reduce the staff's workload effectively if the staff needs to change the battery every day.

Solution: We use the methods listed in Table 1 to reduce power consumption.

Reducing the Arduino running voltage to 3.3V

In the early development stage, we used an external voltage regulation circuit to regulate the 18650 battery voltage to 3.3V and feed the voltage to the Arduino. However, the IoT device's LoRa module did not provide reliable data transmission.

Solution: We use the voltage regulator on the Arduino Pro Micro and everything works.

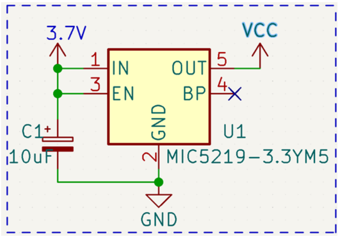

Figure 47 – Schematic of the external voltage regulation circuit in the early development stage

Load cell readings fluctuation and drift

The load cell readings were not stable in the first prototype. Sometimes, they fluctuated by more than two grams every minute.

Solution: We took two actions that improve the accuracy of the readings of the first prototype's load cell:

- The wire connections in the first prototype were loose. We re-soldered the wires carefully and reduced the fluctuation.
- The bottom of the IoT device case did not have enough structural strength and deformed easily. We re-designed the bottom of the case and improved the structural strength by adding reinforcing ribs.

Purchasing Raspberry Pi 4B

Due to the global chip shortage, it was impossible to purchase a Raspberry Pi 4B from the official dealer in Hong Kong in July 2022. The official dealer replied there would be restocking in January 2023, but they were not sure. It is too late to get the Raspberry Pi 4B for the development of the server.

Solution: We bought the Raspberry Pi 4B from Taobao, an online shopping website from mainland China, with the price almost doubled the MSRP.

Accessing a GPIO pin with multiple processes

We needed to access a GPIO pin from multiple processes simultaneously. When one of the processes terminates, all other processes cannot access the GPIO pin.

Solution: We create a service to access the GPIO pin. Other processes use IPC to access the GPIO pin through the service.

Wrapping and bed adhesion problem in 3D printing ABS

The large 3D-printed ABS case did not stick well to the heated bed surface and was tented to wrap.

Solution: We used a PEI sheet as the print bed, cleaned the PEI sheet with soap and steel wool, and increased the bed temperature and the enclosure temperature to 110°C and 50°C respectively.

Conclusion

This project develops the Water Drinking and Pill Taking Reminder (WDPTR) system to address the limitations of the existing commercial products to assist the staff in nursing homes and hospital wards in monitoring the water and pill consumption of the people receiving care. The WDPTR has four main features:

- Monitor and log the water consumption of the user.
- Check if the user has taken the pill and log the time of taking the pill.
- Show the above data in a web interface.
- Notify the staff if any action is needed.

We utilise LoRa technology for the IoT device’s wireless communication and load cell for measuring water and pill consumption. We also developed a web interface that allows staff to easily access and monitor the water and pill consumption of the user and alert the staff if any action is required to be taken.

We carry out some experiments to demonstrate the feasibility and effectiveness of the WDPTR with data analysis. The result shows that the WDPTR IoT device has a long battery life. It also shows that the WDPTR has a high accuracy for both water and pill consumption.

However, there are some rooms of improvement that need further investigation. The pill consumption accuracy is not high enough and may results a false-positive report every two days. Due to the limited time and budget, we are not able to test if some more expensive load cells can achieve better accuracy.

In summary, the Water Drinking and Pill Taking Reminder provides a system for hospital wards and nursing homes to monitor the water and pill consumption of the people receiving care in a convenient and systematic way and notify the staff if any action is required to be taken.

References

[1] G. Edkins, “Human Factors, Human Error and the role of bad luck in Incident Investigations,” 23 May 2016. [Online]. Available: https://www.linkedin.com/pulse/human-factors-error-role-bad-luck-incident-graham-edkins/. [Accessed 9 October 2022].

[2] “Top-Rated Pill Dispenser for the Medication Adherence MedMinder,” MedMinder, [Online]. Available: https://www.medminder.com/. [Accessed 17 March 2023].

[3] “Smart Water Bottle - HidrateSpark Bluetooth Water Bottle and Tracker App,” HidrateSpark, [Online]. Available: https://hidratespark.com/.

[4] “Memo Box Smart Pillbox,” https://pillbox.tinylogics.com/, [Online]. Available: https://pillbox.tinylogics.com/. [Accessed 17 March 2023].

[5] “What is IoT?,” Oracle Hong Kong SAR, PRC., [Online]. Available: https://www.oracle.com/hk/internet-of-things/what-is-iot/. [Accessed 5 October 2022].

[6] B. Lutkevich, “What is a Microcontroller and How Does it Work?,” techtarget, November 2019. [Online]. Available: https://www.techtarget.com/iotagenda/definition/microcontroller. [Accessed 17 March 2023].

[7] “What is Arduino? | Arduino,” Arduino, [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed 17 March 2023].

[8] b_e_n, “What is an Arduino?,” sparkfun, [Online]. Available: https://learn.sparkfun.com/tutorials/what-is-an-arduino/all. [Accessed 17 March 2023].

[9] S. Butler, “What Is GPIO, and What Can You Use If For?,” How-To Geek, 11 April 2022. [Online]. Available: https://www.howtogeek.com/787928/what-is-gpio/. [Accessed 20 March 2023].

[10] “A/D Converters (ADC),” ANALOG DEVICES, [Online]. Available: https://www.analog.com/en/product-category/analog-to-digital-converters.html. [Accessed 20 March 2023].

[11] E. Peňa and M. G. Legaspi, “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,” Analog Dialogue, December 2020. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html. [Accessed 20 March 2023].

[12] “I2C Communication Protocol,” GeeksforGeeks, 16 Feburary 2021. [Online]. Available: https://www.geeksforgeeks.org/i2c-communication-protocol/. [Accessed 20 March 2023].

[13] “Linear & low-dropout (LDO) regulators,” Texas Instruments, [Online]. Available: https://www.ti.com/power-management/linear-regulators-ldo/overview.html. [Accessed 24 March 2023].

[14] “Integrated Development Environment,” techopedia, 11 January 2017. [Online]. Available: https://www.techopedia.com/definition/26860/integrated-development-environment-ide. [Accessed 17 March 2023].

[15] “Getting Started with Arduino IDE 2.0,” Arduino, [Online]. Available: https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2. [Accessed 17 March 2023].

[16] “Arduino/Processing/Python Language Comparison,” Arduino, 9 March 2023. [Online]. Available: https://docs.arduino.cc/tutorials/generic/language-comparison. [Accessed 17 March 2023].

[17] “What Is LoRa®?,” SEMTECH, [Online]. Available: https://www.semtech.com/lora/what-is-lora. [Accessed 8 October 2022].

[18] “LoRa and LoRaWAN,” Semtech, [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/. [Accessed 15 March 2023].

[19] "LoRaWAN World Record Broken Twice in Single Experiment," The Things Network, [Online]. Available: https://www.thethingsnetwork.org/article/lorawan-world-record-broken-twice-in-single-experiment-1. [Accessed 15 March 2023].

[20] "What is a Load Cell & How Does it Work?," Omega Engineering, [Online]. Available: https://www.omega.com/en-us/resources/load-cells. [Accessed 15 March 2023].

[21] "What is a load cell?," Straightpoint, [Online]. Available: https://www.straightpoint.com/what-is-a-load-cell.html. [Accessed 17 March 2023].

[22] "Raspberry Pi Documentation," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/documentation/. [Accessed 20 March 2023].

[23] "Raspberry Pi OS," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/documentation/computers/os.html. [Accessed 20 March 2023].

[24] "1.2. What is Debian GNU/Linux?," Debian, [Online]. Available: https://www.debian.org/doc/manuals/debian-faq/basic-defs.en.html#whatisdebian. [Accessed 20 March 2023].

[25] C. Hope, "What is a Server?," Computer Hope, 31 December 2022. [Online]. Available: https://www.computerhope.com/jargon/s/server.htm. [Accessed 20 March 2023].

[26] "What is LAMP," java point, [Online]. Available: https://www.javatpoint.com/what-is-lamp. [Accessed 20 March 2023].

[27] "What is Linux?," Linux, [Online]. Available: https://www.linux.com/what-is-linux/. [Accessed 20 March 2023].

[28] "What is Apache?," Apache, [Online]. Available: https://cwiki.apache.org/confluence/display/httpd/FAQ#FAQ-WhatisApache?. [Accessed 20 March 2023].

[29] "About MariaDB Server," MariaDB, [Online]. Available: https://mariadb.org/about/. [Accessed 20 March 2023].

[30] J. A., “MariaDB vs MySQL – Key Differences, Pros and Cons, and More,” Hostinger Tutorials, 21 December 2022. [Online]. Available: https://www.hostinger.com/tutorials/mariadb-vs-mysql. [Accessed 20 March 2023].

[31] “What is PHP?,” php, [Online]. Available: https://www.php.net/manual/en/intro-whatis.php. [Accessed 20 March 2023].

[32] “What is Python? Executive Summary,” Python, [Online]. Available: https://www.python.org/doc/essays/blurb/. [Accessed 20 March 2023].

[33] “What is SQL?,” Amazon, [Online]. Available: https://aws.amazon.com/what-is/sql/. [Accessed 23 March 2023].

[34] Sandi, “What Is Service In Linux,” ImagineLinux, [Online]. Available: https://www.imaginelinux.com/service-in-linux/. [Accessed 20 March 2023].

[35] “Inter Process Communication (IPC),” GeeksForGeeks, 27 March 2023. [Online]. Available: https://www.geeksforgeeks.org/inter-process-communication-ipc/. [Accessed 27 March 2023].

[36] R. Meltzer, “What is a Programming Library? A Beginner’s Guide,” CareerFoundry, 3 January 2023. [Online]. Available: https://careerfoundry.com/en/blog/web-development/programming-library-guide/. [Accessed 20 March 2023].

[37] “What is a 18650 Lithium Battery?,” Utmel, 15 December 2021. [Online]. Available: https://www.utmel.com/blog/categories/battery/what-is-a-18650-lithium-battery. [Accessed 20 March 2023].

[38] “What is FDM (fused deposition modeling) 3D printing?,” HUBS, [Online]. Available: https://www.hubs.com/knowledge-base/what-is-fdm-3d-printing/. [Accessed 20 March 2023].

[39] “How to use heat set inserts to securely fasten 3D printed parts,” UltiMaker, 29 March 2022. [Online]. Available: https://ultimaker.com/learn/heat-set-inserts. [Accessed 26 March 2023].

[40] “Pro Micro - 5V/16MHz,” sparkfun, [Online]. Available: https://www.sparkfun.com/products/12640. [Accessed 24 March 2023].

[41] "ATmega16U4/32U4 Datasheet," Atmel, [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf. [Accessed 24 March 2023].

[42] "500 mA Peak Output LDO Regulator," Microchip, [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/MIC5219-500mA-Peak-Output-LDO-Regulator-DS20006021A.pdf. [Accessed 24 March 2023].

[43] "Reducing Arduino Power Consumption," Sparkfun, [Online]. Available: https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption/all. [Accessed 24 March 2023].

[44] "24-Bit Analog-to-Digital Converter (ADC) for Weight Scales," AVIA Semiconductor, [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. [Accessed 24 March 2023].

[45] "A39C-T400A22D1a," Gisemi Microelectronics, [Online]. Available: http://download.gisemi.com/datasheets/a39ct400a22d1acn.pdf. [Accessed 24 March 2023].

[46] "systemctl," freedesktop.org, [Online]. Available: https://www.freedesktop.org/software/systemd/man/systemctl.html. [Accessed 26 March 2023].

[47] R. W. D. -. T. Viewport, "W3Schools," [Online]. Available: https://www.w3schools.com/css/css_rwd_viewport.asp. [Accessed 2 April 2023].

[48] C. R. A. Inc, "3 Common Wireless Communication Technologies in the IoT," Linkedin, 12 November 2021. [Online]. Available: https://www.linkedin.com/pulse/3-common-wireless-communication-technologies-iot-ct-rf-antennas-inc. [Accessed 1 April 2023].

[49] "NB-IoT," Amazon aws, [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/nb-iot.html. [Accessed 1 April 2023].

[50] E. Bradstock, “WiFi Range: What is the typical distance of a WiFi signal?,” Canstar Blue Pty Ltd, 9 July 2021. [Online]. Available: https://www.canstarblue.com.au/internet/wifi-range-how-far/. [Accessed 1 April 2023].

[51] “What is the maximum communication range of the Bluetooth connection?,” Sony, 28 January 2023. [Online]. Available: https://www.sony.com/electronics/support/articles/00014085. [Accessed 1 April 2023].

[52] A. Pelaez, “LoRaWAN vs NB-IoT: A Comparison Between IoT Trend-Setters,” Ubidots, 4 February 2020. [Online]. Available: https://ubidots.com/blog/lorawan-vs-nb-iot/. [Accessed 1 April 2023].

[53] “LoRa 模塊通信 LLCC68 替 SX1278/62 芯片 433 無線串口模塊遠距離開發板,” Taobao, [Online]. Available: https://detail.tmall.com/item.htm?abbucket=9&id=667904570045&ns=1&spm=a230r.1.14.16.1255526aMkCgP8&skuId=4984447986213. [Accessed 1 April 2023].

[54] “SX1262/SX1268 串口收發 433/900M 自動組網低功耗射頻 lora 無線模塊,” Taobao, [Online]. Available: https://detail.tmall.com/item.htm?abbucket=9&id=597597634063&ns=1&spm=a230r.1.14.40.1255526aMkCgP8&skuId=4985242118578. [Accessed 1 April 2023].

[55] “Buy a Raspberry Pi 4 Model B,” Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. [Accessed 1 April 2023].

[56] “Creep test with 3D printed plastic materials: PLA, PETG, ASA, Nylon,” My Tech Fun, 6 October 2021. [Online]. Available: https://www.youtube.com/watch?v=88pk2cNOeGE&ab_channel=MyTechFun. [Accessed 2 April 2023].

[57] M. Mayer, “Is PLA heat resistant? ABS, ASA, PETG and more!,” 3D Solved, [Online]. Available: https://3dsolved.com/is-pla-heat-resistant-abs-asa-petg-and-more/. [Accessed 2 March 2023].

[58] “Serial Port Monitor - COM port sniffer & analyzer,” Electornic Team, Inc., [Online]. Available: https://www.eltima.com/products/serial-port-monitor/. [Accessed 2 April 2023].

[59] “M2.5 x 5mm Brass Heat Set Threaded Round Insert Nut,” Flyrobo. (n.d.), [Online]. Available: https://www.flyrobo.in/image/cache/wp/gj/m2.5-x-5mm-brass-heat-set-threaded-round-insert-nut/m2.5-x-5mm-brass-heat-set-threaded-round-insert-nut1-1000x1000.webp. [Accessed 26 March 2023].