Autonomous Home Brewing System

Brew Buddy Team: Sam Ward (BCIS & BEng), Mike Emmerson (BEng), Jacob Boon (BEng)
Project Supervisors: Minh Nguyen (BCIS) & Jeff Kilby (BEng)

BACKGROUND

The discovery and development of brewing techniques date back to about 12000BC and represent one of the most important technological achievements of humankind (Cabras & Higgins, 2016). In more recent times the revolution of small batch breweries and craft beer has taken hold around the globe, making its way into the everyday household with the increase in popularity of hobbyist home brewing.

The Brew Buddy team consists of three home brewing enthusiasts with 25 years of electronics engineering experience between them, that through their own experience have discovered a niche gap in the home brewing industry for retrofittable brewing automation software.

Brew Buddy is an autonomous control system that allows home brewing enthusiasts to add automation to their setup utilising their existing equipment in a semi-modular fashion; and is designed to be adaptable and scalable from small to medium sized home breweries. It provides users with automation functions that are not currently available on the home brewing market, releasing them from the tedious process of manual brewing with a simpler implementation and without having to invest in an all-in-one solution.

Fig. 1. Demonstration video

PROJECT CONCEPT

Market Research/Survey

To help us determine our system requirements and target areas we engaged with the Whenuapai Brew Club to give us some market feedback in the form of a survey. This deemed to be very helpful and highlighted some areas of improvement we had not considered.

Using this feedback along with our own experience we identified the following focus points of the project Brew Buddy: Safety by eliminating the need to move vessels containing hot liquids; time saving by being more hands off; and a further level of automation than currently available through temperature control, sparge automation, and cleaning automation.

Fig. 2. Common Frustrations among Home Brewers

Project Objectives

Our project goal was to deliver a viable market product over the course of two Semesters. This would be achieved through the following eye-level objectives identified during our planning phase:

- Simplify the brewing process and require the least amount of user interaction as possible.
- Interface with the users WIFI network and be controlled either through a mobile application or web app
- Continuously update with live sensor information and progress tracking
- Allow the user to input their desired recipe information into the mobile application or web app which defines the timings, temperature, flow rates and specific gravity that the system will operate within.
- To be semi-modular and to have the ability to interface with existing home brew setups.

App Prototype

An interactive prototype was first developed to aid with app development. This helped to fast-track design decisions among the team, allowing us to easily modify features and layout before moving on to building the app itself.

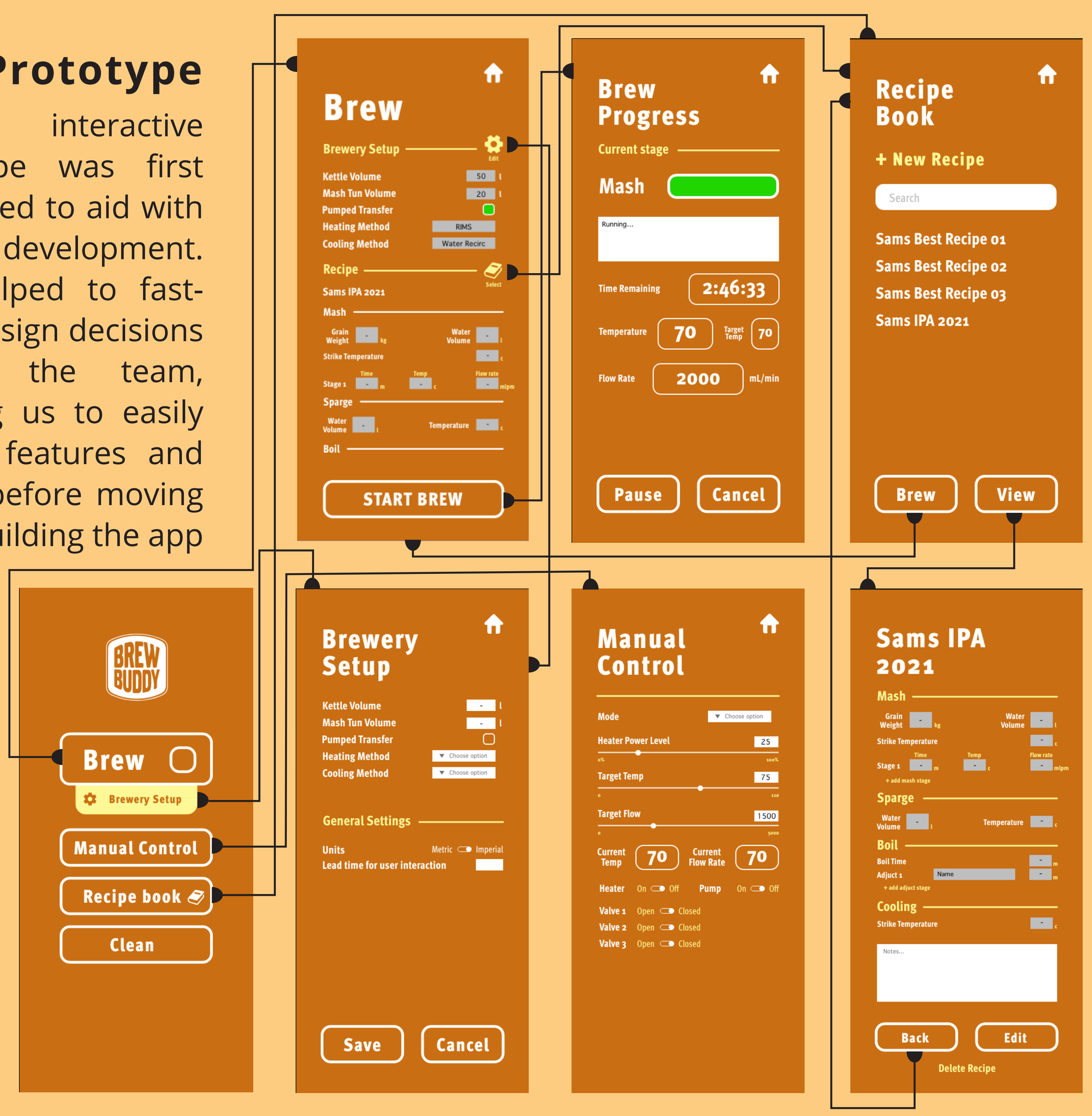

Fig. 3. Brew Buddy app prototype

DEVELOPMENT PROCESS

Hardware

ESP32 Microcontroller

The ESP32 microcontroller was selected for its powerful Xtensa® dual core processor and onboard Bluetooth/Wi-Fi (ESP32 Series Datasheet, 2021), suiting it to the heavy requirements of running both the control algorithms and web server. We decided to go with a dev board version with onboard programmer and 3.3v regulator for ease of integration.

Electrical Components

The Brew Buddy system is powered by a standard 230V AC outlet and consists of a high-power side and a 5V low power side, with an AC-DC converter and filtering circuit providing the 5V. The high-power side consists of relays for heater and pump control, and a proprietary power supply utilising PWM to provide variable power to the heating element.

Various sensors are employed to provide feedback to the microcontroller such as flow meters, temperature probes, and float switches. This feedback is interpreted by Brew Buddy's control algorithms to control pumps, heater elements, and servo valves. The servo valves are an in-house custom designed and printed three-way servo-controlled ball valve that allows for changing of flow direction and proportional flow control in either direction.

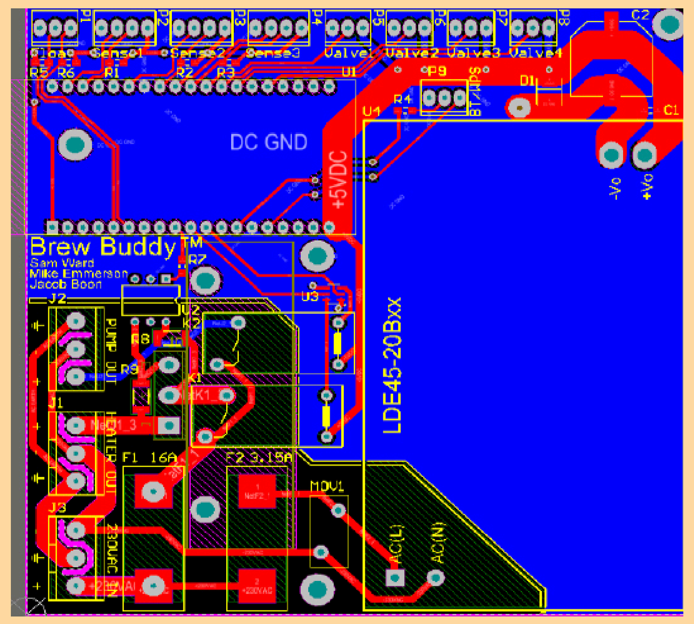

Fig. 4. PCB component layout

Printed Circuit Board (PCB)

A PCB was designed for housing the microcontroller and electronic circuits, and for connecting the peripherals. The PCB essentially consists of two electrically separate sides, the high-power(230VAC) side in the bottom left and the low-power(5VDC) side in the blue section with ESP32 breakout.

When designing the PCB, the following factors were considered: PCB must fit inside a small enclosure, tracks must support 230VAC and 15A, convenient placement of components requiring easy access, clear separation of AC and DC sides, and minimised noise interference to the ESP32 antenna.

Software

Control Algorithms

The ESP32 was coded using the ESP-IDF extension for Visual Studio Code. ESP-IDF is Espressif's official IoT Development Framework for the ESP32, providing a self-sufficient SDK for any generic application development using programming languages such as C and C++ (Espressif Systems, 2021). Functions were developed to control each of the individual components and the automation algorithms to tie them together.

Backend

The web server is hosted on the ESP32 and utilises the ESP-IDF REST server. It contains the various URI handlers which then call the appropriate control algorithms, using cJSON to pass data between the front and back end.

Frontend

The Brew Buddy web-app was developed in Vue.js using the Vuetify framework.

Vue was chosen as it is more lightweight than React, and allows for easy development of cross platform native apps by porting the same component model through weex.

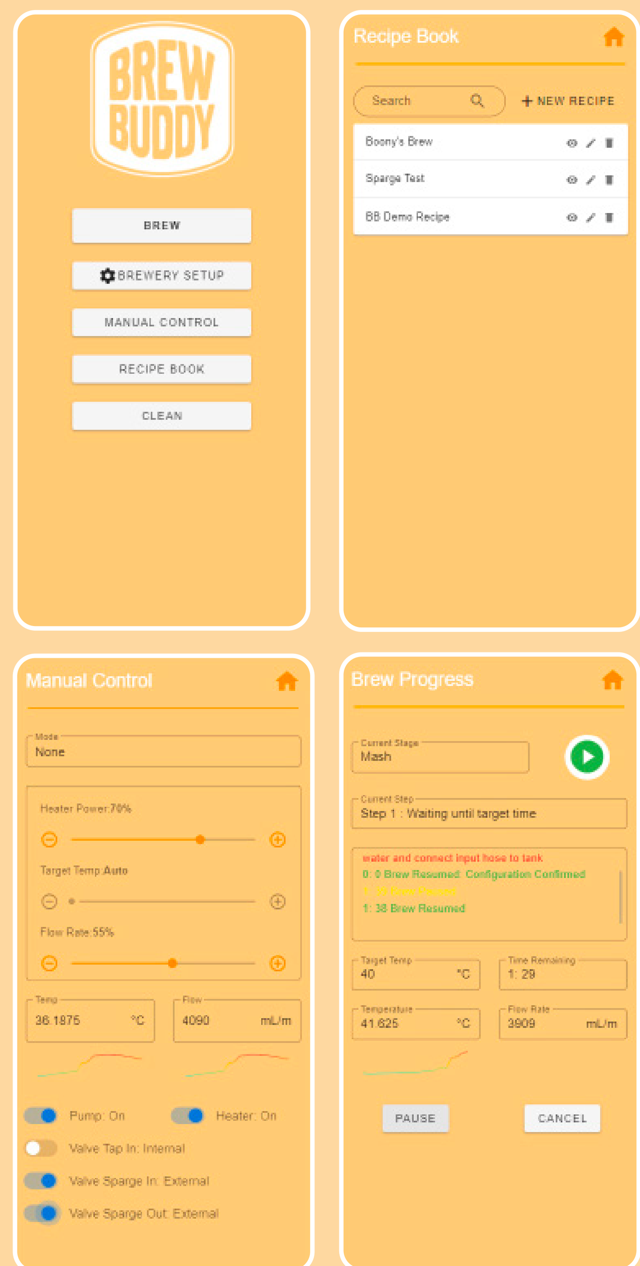

Fig. 5. Brew Buddy app

FUTURE DEVELOPMENT

Distillation

Originally part of the initial project proposal it was discarded due to low amount of interest from the community. This would be a relatively simple addition to Brew Buddy as it utilises the same control systems, just requires another algorithm to be developed.

Mobile App Development

Developing a native mobile app that the user can download from the app store rather than using the web browser would be a nice feature for further development. This coupled with the addition of push notifications would give the Brew Buddy user a true mobile experience.

CHALLENGES

Technical

- COVID-19 hindered our progress toward the end of the Semester by not allowing us to meet up or use the AUT laboratory, as such we were unable to get Brew Buddy electrically certified for the 230V.
- Had an issue with the ESP32 running out of flash memory. To remedy this I moved storage of the front-end files to an external web server acting as a content delivery network (CDN), unfortunately resulting in the app no longer being offline as originally proposed.

Non-technical

- Being the sole front and back-end developer presented its challenges. It would have been helpful to have another software developer with experience in the team to bounce ideas off, get software specific feedback, and to help with problems.
- Balancing the combined electronics and software project proved to be harder than expected. I found myself at times focusing too heavily on one aspect when another part of the project needed attention.

Lessons Learnt

- We didn't establish thorough coding standards at the beginning of the project, this led to a bit of confusion when helping or reviewing other's code due to different coding styles among the team.
- Steep learning curve for both ESP-IDF and Vue.js. More research into these at the project initiations stage would have been beneficial for establishing upskilling requirements.

References:

Cabras, I., & Higgins, D. M. (2016). Beer, brewing, and business history. *Business History*, 58(5), 609–624. https://doi.org/10.1080/00076791.2015.1122713

ESP32 Series Datasheet. (2021). https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

IoT Development Framework I Espressif Systems. (2021). Espressif.com. https://www.espressif.com/en/products/sdks/esp-idf