# Automatic Pet Feeder

CPSC-6820 Deliverable #2—System Overview Document

Logan Swoyer Sriram Vishnubhotla

February 2025

## 1 Executive Summary

It feels terrible to realize we did not feed our pets—whether it is because we were running late and were in a hurry or had too much on our mind and forgot about it. Times like these are when the Automatic Pet Feeder comes in handy. It comes with a nifty set of features to take care of your pet feeding needs including—the ability to set multiple food dispense times a day, on-device dispense schedule settings, internet connectivity to pair your mobile devices and publish logs, a mobile app to receive notifications and view logs as well as to set the dispense schedule and the ability to set dispense quantities per each dispense time. Internet connectivity is completely optional and is helpful to check logs and remotely edit the dispense schedule.

## 2 System Description

### 2.1 Software

The device's main loop can be logically divided into three phases: reading from and writing to I/O—including rendering on-device display based on button inputs and activating the buzzer, pushing device state to the web server and updating the local state from the web server's state, setting the timers based on the schedules and dispensing food when the timers are triggered. The following sections lay out more details of these phases.

#### 2.1.1 Device I/O

For the device to be self-sufficient without internet connectivity, the on-device display needs to act as a comprehensive user interface. ¡figure¿ shows the planned frame layout. The frame consists of a number input at the center to take schedule times and portion sizes as inputs, icons on the left—including internet connectivity indicator, storage level indicator and schedule identifier. The schedule identifier can be toggled to view or edit multiple schedules, each of which is identified by the identifier. The display would be controlled using

a joystick–select button combo input and a buzzer would be fired to alert the pet/user when dispensing food.

### 2.1.2 Web Interoperation

The device would be equipped with a Wi-Fi module that sets up a private wireless network to pair with a mobile device, which needs a client app installed. The user's home network credentials will then be exchanged with the device for it to establish a connection with the network. Once this step succeeds, the device would be ready to go online and synchronize device state with data on the server. This includes the device clock and the dispense schedule and portion sizes. From this point, the device would push activity data to the server—which would be used by the mobile app, for instance, to notify the user when food is dispensed or if the storage level is low.

### 2.1.3 Scheduling and Dispensing

Schedules are set using the microcontroller's online timers and TI's RTC library. The timer interrupts trigger the device to serve food in the container.

### 2.1.4 UI

The device's user interface would consist of a power on–off switch, a joystick, a buzzer and an LCD display. The joy stick will be used by the user to control selection on the LCD display. The LCD display will visualize setting the device clock, the dispense schedule and portion sizes and alert the user of warnings. The buzzer is used in order to alert the user or pet when food is being dispensed.

Besides this, a companion mobile app will be developed to achieve the same functionality remotely, while accessing device logs.

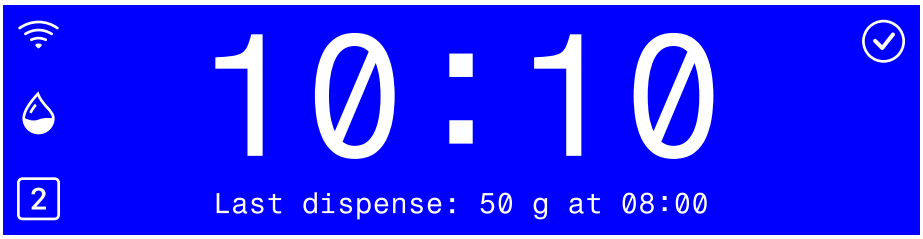


Figure 1: Device display prototype.

## 2.2 State Machine

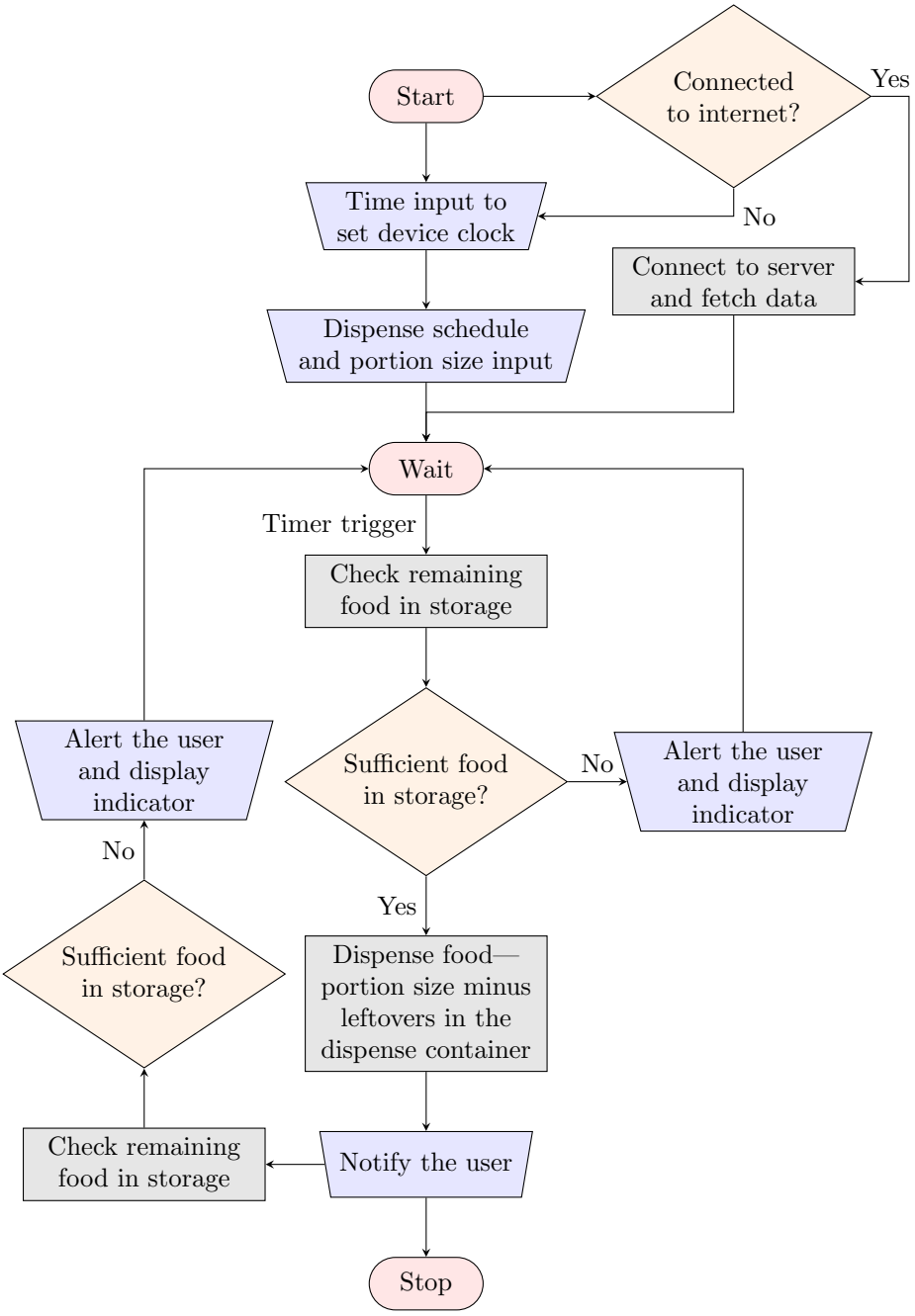


Figure 2: Automatic Pet Feeder's state machine.

## 2.3 Hardware

### 2.3.1 Sensors

The device will have two sensors, a force sensor and an optical sensor. The force sensor is used to measure the amount of food in the dispense container before and during dispensing. This will allow the device to dispense the proper amount of food set by the user. The optical sensor is used in order to measure the amount of food left in the storage. This allows the user to be alerted when food is running low so they can refill it.

### 2.3.2 Microcontroller

We plan to use MSP430FR5994 in order to control this device. Using this microcontroller eases the workload since we can reuse the knowledge of the MCU we learned in class.

### 2.3.3 Actuator

The device will be equipped with a linear actuator that will be in charge of opening and closing the dispenser. We made sure to choose one that can handle enough force to reliably hold the weight of the food.

### 2.3.4 Wireless communication protocol

For our wireless component we decided to use a Wi-Fi module. This module will be used to connect to the user's home network and communicate with the web server.

### 2.4 Parts

| Title | Part # | Price | Supplier |
|---|---|---|---|
| Rocker Switch | COM-11138 | 0.55 | Sparkfun |
| Joystick with Select Button | 512 | 5.95 | Adafruit |
| Buzzer | PS1240P02BT | 0.55 | TDK Corporation |
| Graphic LCD Display | NHD-C12832A1Z-NSW-BBW-3V3 | 12.82 | Newhaven Display Intl |
| WiFi Module | ESP32-S2FH4 | 1.48 | Espressif Systems |
| Linear actuator | FIT0803 | 24.00 | DFRobot |
| MSP430FR5994 Microcontroller | MSP-EXP430FR5994 | 20.39 | Texas Instruments |
| Force sensor | GHF-10 | 5.86 | UNEO Inc. |
| Optical sensor | VL6180V1NR/1 | 3.37 | STMicroelectronics |
| | | 74.97 | |

Table 1: Device part list and details.

## 3 Demo

The demo would begin with us showing the audience the design and features of the device. We would then fill the storage with food, set the dispense container up to collect the food and power the device on. Once done, we would demonstrate the setup phase—which will involve pairing the device with user's mobile and connecting it to a wireless network, followed by setting the feeding schedule and portion sizes. For demonstration purposes, we would set short time intervals to show the functionality. Finally, once everything is set up, we show the device alerting and dispensing the food on time, as well as pushing notifications to the user's mobile device.

## 4 Resources, References and Appendix

- TI—Using the Real-Time Clock Library
- TI—Timers
- TI—MSP430 workshop
- TI forums—Setting up alarms using RTC library