TeaBOT Order Packaging Machine Design

Qiao Ling Zhang¹, Ting Ai²

¹,²Armored Forces Engineering institute of Army, ChangChun, China
zhangqiaoling10@163.com; 65963974@qq.com

Abstracts: A Circuit design for meeting the rules of Tea Dispensing Machine, To achieve create data objects are what actually push the data objects to the database. Weight Sensor Load Cell and Motor Controller for The Raspberry Pi communicates both with the local hardware and with the database in order to dispense and complete orders as they come in.

1. INTRODUCTION

1.1. The circuit functions
The requirements set out to the user by TeaBOT cover both the web app and the dispensing machine. For the web app, TeaBOT provided to the user with three requirements: to be able to select up to three ingredients from a list, to be able to set custom amounts of each ingredient (totally 60g), and to be able to design a custom label to be placed on the tea bag. For the dispensing machine, TeaBOT provided to the user with four requirements: to accurately weigh 60g with an upper margin of error of 2g, to fill tea bag within 60 seconds of receiving the order, to be able to fill multiple bags simultaneously, and to print the custom label and place it on the bag.

Once the order was placed, the physical machine was able to find all incomplete orders in the database and process them one at a time until there were no more new orders. It was able to turn on various motors to dispense tea and could accurately weigh the dispensed ingredients up to a total weight of 85g, with a margin of error of 0.1g. It was unable to verify the total dispensing time, as the tea dispensing mechanisms sent to the user were indeterminacy. But it can say with relative assurance that TeaBOT has met the 60 second total dispensing time, as it can currently process an order in less than 20 seconds without worrying about waiting for the tea to dispense. TeaBOT has promised the user the dispensing mechanism will take about 30 seconds to dispense 60g of tea. Although the physical machine could not test the final system to verify, it can say with some degree of certainty that the physical machine have met the 60 second time requirement, with our system coming in at an estimated 50 seconds.

1.2. The circuit design
The prototype consists of a physical machine frame to dispense and weigh the ingredients, as well as a web interface for customers to place orders and design custom labels. The physical machine weighs 10lbs and measures 1’ X 1’ X 2.5’ (W:L:H). The machine communicates over WiFi, or an Ethernet cable in order to pull new orders from the database. The entire system is powered by a standard wall outlet.

The web application uses a Spring Boot framework and stores all data in a MongoDB database. There are TeaOrder and Label objects stores in the database, each stored in its own collection.
Libraries used in the application include JQuery and Fabric.js. There are three web pages: the tea creation page, the label creation page, and the order confirmation page. After the order being placed, it goes to tea dispensing process. Tea bag need to be detected loaded on weight scale before starting dispensing. The dispensing mechanism is set to dispense tea one by one to make sure each type of tea is served at desired quantity. The speed of dispensing slows down as the weight approaching 60g.

2. SOFTWARE (UI SPECIFICATIONS)

2.1. End user constraints
1) The user must select between one and three ingredients from the predefined list of ingredients
   2) The user must select the ingredient amounts, or else the ingredient amounts will be evenly distributed
   3) The user must create their own label, or else the label will remain blank except for listing the creator, blend name, and blend ingredients
   4) The user must use the provided clip art or upload their own when adding images to a label
   5) The user must provide shipping information to confirm their order

2.2. Create data objects
The HTML inputs are bounded to the data objects using Thyme leaf so that when the user submits their tea order, the TeaOrder object contains the user input and is passed to the TeaOrder Controller. The TeaOrder Controller contains the GET and POST requests for the TeaOrder page. In the POST command, the controller passes the TeaOrder object to the next controller, the Label Controller. This is done in case a user does not complete their order, then the TeaOrder object will not be pushed to the database.

2.3. Saving states
The user has finished building their blend and moves on to the next page, the inputted information (tea types, tea amounts, creator name, and blend name) is saved to the browser’s session storage to repopulate these fields if the user decides to edit the order at a later stage in the process. This information is also used to repopulate the user’s input if they decide to return to the page to edit their order. A similar method is utilized for the label creation page, saving both the label canvas and the data URL of the PNG version of the canvas.

2.4. Pushing to the database
Upon completing the label, the Label and TeaOrder objects are passed to the final page, the confirmation page. When a user confirms their order, these objects are passed to the Confirmation Service, and then passed to the TeaOrder Repository and Label Repository. These repository objects are what actually push the data objects to the database.

3. CIRCUIT PART DESIGN

3.1. Sensor acquisition signal

Fig1. Strain gauge weight sensor
As shown in Figure 1, a high accuracy and excellent stability aluminium alloy weighing sensor and a full-bridge strain gauge load cell are used to build the weight sensor. By adding force on the top edge, the strain gauge resistance changes, causing the voltage in between to change (as shown in Figure 2 and Figure 3). It converts the load to electrical signals. The output is voltage difference. An H-Bridge is a circuit that can drive a current in either polarity and be controlled by Pulse Width Modulation (PWM). Pin layout and connections are shown as below, a voltage divider must be used as shown Figure 4.

The range sensor detection is activated by sending a pulse signal to TRIP (gpio-out); it will automatically detect when there is a reflected signal. When the reflected signal returns, ECHO (gpio-in) outputs at a high level. The measured distance = (T(time to high level output)*340m/s)/2. Temperature is also a variable relative to speed of sound. as shown in the Figure5.
3.2. Motor Controller
The Raspberry Pi communicates both with the local hardware and with the database in order to dispense and complete orders as they come in. It also has a monitor connected to its HDMI port and uses this to display the status and state the machine is currently in. It also has an emergency stop button in the GUI, which stops everything and quits the program. Eventually, there will also be additional buttons for the user to take control of certain features like pausing an order as it is being dispensed. General circuit Figure 6.

![L298N Motor Driver](image)

Figure 6. L298N Motor Driver

3.3. Software details
The Raspberry Pi communicates with the MongoDB database by sending JSON queries over the internet VIA a USB WiFi dongle. It sends simple queries to find orders that have not been set to complete, retrieves and displays all the relevant information about the order being processed and begins the dispensing process. Once the Pi detects that the correct amount of tea has been dispensed, it then sets the order to ‘Complete’ and pushes this information back to the database. The WiFi dongle is to eventually be replaced with a permanent ethernet connection, once the Pi is in a more permanent home.

3.4. Features Tested
Tea Dispensing Motor: The tea dispensing motor includes two dual H bridge motor drivers and three tea dispensing motors. We tested the tea dispensing motor using the Raspberry Pi by running the three motors simultaneously and individually in different speed. All of our motors have ran smoothly in our trials.

Tea Dispensing Mechanism: The 3D printed dispensing mechanisms sent by TeaBOT are tested to be faulty. When loaded with tea, the mechanism would jam, causing the motor to turn off. Going forward, TeaBOT will replace the current mechanism with a functioning one.

Bag Detector: The ultrasonic range finder is used for tea bag detection, we tested the range finder with the Raspberry Pi and set the detecting range within 15cm. The range finder is tested by placing and removing an object in front of the sensor and checking if the Raspberry Pi receives the signal outputted by the range finder.
Weight Scale: The weight scale feature is including the strain gauge load cell and the weight scale circuit (the amplifier, the DC/DC converter, and the analog to digital converter). The load cell sensors were tested with a set of weights. The weight scale was found to have an accuracy of ±0.1g. In addition, the weight scale was tested with the Raspberry Pi to ensure the Pi gets the accurate input from the load cell sensor.

4. FUNCTION OF MACHINE REALIZATION
The following state machine demonstrates the logic used by the Rasperry Pi to process and complete new orders Figure 7

![State Machine Diagram]

Fig. 7 The logic used by the Raspberry Pi

Function of circuit realization delivers the following information to the TeaBOT staff. Status: The current state of the machine (Idle, Polling for orders, Load bad, Dispensing, Remove Bag, Order Complete, Idle), Number of Orders: The number of new orders waiting to be completed. Status of motors: The current status of the three dispensing motors and which ingredients they are currently dispensing. Console: A generic message area that is used to display the orders as they are processed, as well as a times tamp for each order and error codes that may arise during operation. Label: Here we display the label for the current order that is being dispensed to facilitate matching the printed label with the bag being filled.
5. CONCLUSION

In summary, Circuit Design for meeting the specifications document of Tea dispensing machine put to use a high accuracy and excellent stability aluminium alloy weighing sensor and a full-bridge strain gauge load, Realized TeaBOT can meet the demands of users: to accurately to fill tea bag seconds of receiving the order, to be able to fill multiple bags simultaneously, and to print the custom label and place it on the bag.

This design has the characteristics of convenience, quickness and individuation.

REFERENCES

[1] N. Okabe, Materials Science and Engineering, A143, 11-19(1991).
[2] Matsui, M., Masuda, M., and Yamada, N. J. of Ceramics, Vol.25, No.2, 1990, pp.138-141.
[3] weight sensor specification http://www.phidgets.com/products.php?product_id=3139
[4] Ultrasonic range finder specification http://www.alibaba.com/product-detail/Ultrasonic-Module-HY-SRF05-Distance-Sensor_1898343708.html
[5] L298N chip internal circuit http://www.bristolwatch.com/L298N/L298N_arduino.htm
[6] Lingyun, Wang, Fusheng Pan Metal matrix composite solid Research progress of preparation technology of liquid reaction [J], Chongqing University Journal, 2004, 24(11), 151-156 (in Chinese)