Boston University
Electrical & Computer Engineering
EC463 Capstone Senior Design Project

Second Prototype Testing Report

Blitz

Team 7
Blitz

Cooper Salmon cosalmon@bu.edu
John Gillig jgillig@bu.edu
Kyle Martin ksmartin@bu.edu
Lindsey Volk volkl@bu.edu
Mariano Pache mnpache@bu.edu

Testing Date: 02/27/2020

1.0 Required Materials

1.1 Hardware

1. 3 x ESP32 Modules
2. 3 x HX711 AD Modules
3. 12 x 50kg Half Bridge Load Cells
4. 12 x 3D Printed Load Cell Mounts
5. 3 x Breadboards
6. Jumper Wires
7. 6 x 3’ x 1.5’ Plywood Boards (Shelves)
8. Objects of known weight (apple juice and 2 x peanut butter jars)
9. USB to Micro USB (for power and data transfer)
10. Linksys N300 Router
11. Ethernet cable
12. Raspberry Pi 3 Model B
13. 2.4 Amp Micro USB charger (to power Raspberry Pi)
14. HDMI Cable
15. USB Mouse
16. USB Keyboard

1.2 Software

1. Arduino IDE Weight Testing Sketch
2. Node.js Server (server.js)
    a. This server receives data from the ESP32s through a UDP socket and updates the mySQL database

2.0 Setup

2.1 Overview

Hardware setup consists of the physical positioning of load cells on their 3D printed mounts and the placement of the composite shelf on the array of four load cells, in addition to the interconnection of the load cells and the HX711 and the interconnection of the HX711 and the ESP. The ESP and mySQL database are connected via UDP following the initialization of the Node server. This should be repeated for all shelves.

2.2 Hardware Procedure

1. Assemble the shelves
2. Upload the First_Prototype_Testing_Sketch.ino to the ESP32

2.3 Software Procedure

1. Open a terminal and start the node server on the Raspberry Pi (node ./20-07-Storemap/CPU/node.js)
2. Open another terminal and log into the mySQL interface using “mysql -u root -p EC464Team7”
3. Run “use store” to access the database
4. Observe current weight on shelves using “select * from shelves”

3.0 Testing

3.1 Overview

Initial system testing primarily focuses on the validation of basic hardware and software functionality, as well as each system's ability to interface with the other. Hardware functionality will be tested through the addition and removal of grocery store items. Two unique sequences of item addition and removal will be followed to test the system's adaptability. Following the addition or removal of any item, the shelves' recorded inventories in the mySQL database will be compared with the actual inventory on the shelves. This test's measurable criterion for success is that the shelves' inventory represented in the mySQL database matches the actual inventory on the shelves 90% of the time.

3.2 Procedure

1. Complete hardware and software set up procedures according to sections 2.2 and 2.3
2. Run "select * from products" in the mySQL terminal to ensure that the stored weights of each item to be placed on the shelves are correct
3. Remove all items from the shelves before running the testing sequences
4. Run Sequence 1
    a. After each item is placed on or removed from the shelves, run "select * from products" to view inventory updates.
    b. Mark "Yes" only if the expected inventory matches the inventory in the database
5. Run Sequence 2 in the same manner as Sequence 1
    a. After all items have been placed on the shelves, run "select * from products" to see if the inventory is accurate
    b. Remove all items from the shelves, run "select * from products", and note the accuracy

3.3 Measurable Criterion

Table 1: Item Name and Weight

Item Name	Weight
Apple Juice	4.15 lbs
Peanut Butter	1 lbs
Goldfish	0.40 lbs
Oatmeal	1 lbs
Cheerios	0.70 lbs

Table 2: Sequence 1

Expected Shelf 1 Inventory	Expected Shelf 2 Inventory	Expected Shelf 3 Inventory	Accurate Inventory In Database (Yes/No)
Peanut Butter: 4 Apple Juice: 2	Goldfish: 2 Oatmeal: 2	Cheerios: 2	Yes
Peanut Butter: 3 Apple Juice: 1	Goldfish: 1 Oatmeal: 1	Cheerios: 1	Yes
Peanut Butter: 0 Apple Juice: 0	Goldfish: 0 Oatmeal: 0	Cheerios: 0	Yes

Table 3: Sequence 2: All On, All Off

Expected Shelf 1 Inventory	Expected Shelf 2 Inventory	Expected Shelf 3 Inventory	Accurate Inventory In Database (Yes/No)
Peanut Butter: 4 Apple Juice: 2	Goldfish: 2 Oatmeal: 2	Cheerios: 2	Yes
Peanut Butter: 0 Apple Juice: 0	Goldfish: 0 Oatmeal: 0	Cheerios: 0	Yes

3.4 Schematic Diagrams

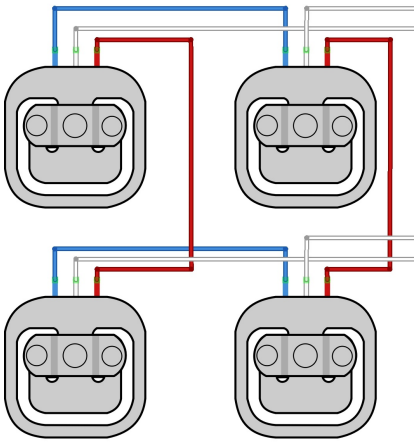

Figure 1: Load Cell Wiring Schematic

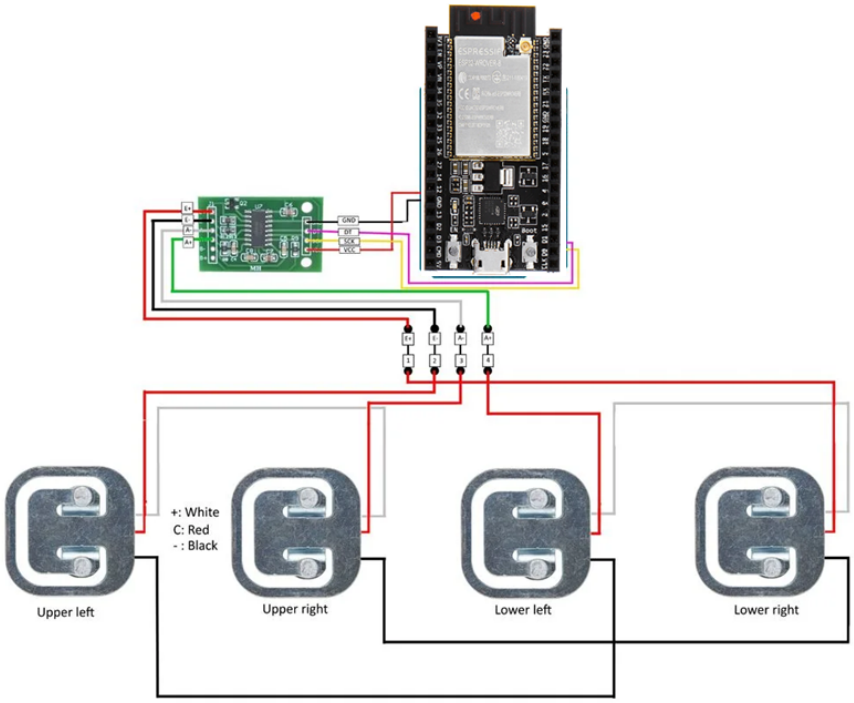

Figure 2: Load Cell, HX711, and ESP32 Wiring Schematic

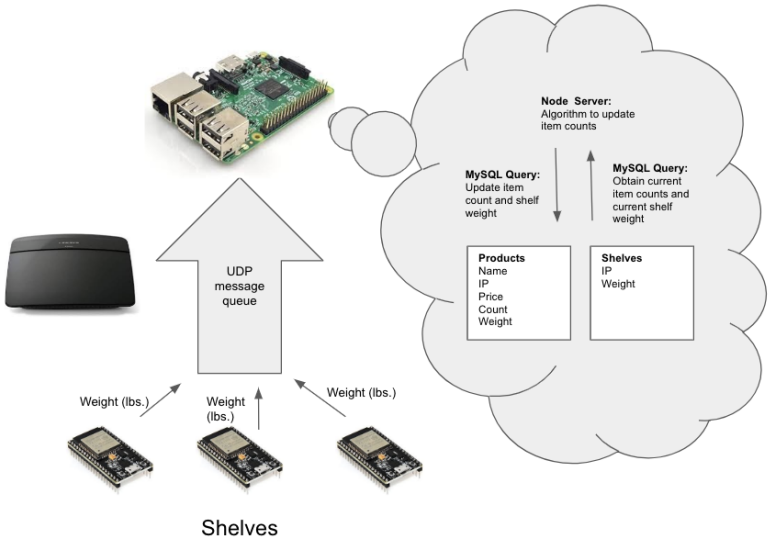

Figure 3: System Overview

3.5 Pinout

Table 1: ESP32 Pinout

Signal	Pin
VCC (from HX711)	3V
DT (from HX711)	Digital Pin 6
SCK (from HX711)	Digital Pin 5
GND (from HX711)	GND

Table 2: HX711 Pinout

Signal	Pin
Lower Right Load Cell Data	E+
Upper Left Load Cell Data	E-
Upper Right Load Cell Data	A-
Lower Left Load Cell Data	A+

4.0 Conclusion

As described in section 3.1, our second prototype testing consisted of two sequences of item placement and removal. We utilized three smart shelves and twelve grocery store items (four jars of peanut butter, two gallons of apple juice, two bags of goldfish, two containers of oatmeal, and two boxes of cheerios). One smart shelf tracked the inventory of peanut butter and apple juice, one smart shelf tracked the inventory of goldfish and oatmeal, and one smart shelf tracked the inventory of cheerios. In the database, we assigned the items and their respective weights to the smart shelves. We did so based on the IP address of each shelf, which can be identified by the ESP controlling the shelf. Based on these assignments and the current weight of the shelves, the database was able to identify the inventory of the shelves, and this database was checked after the addition or removal of any item during the testing. The accuracy of this database was our measurable criterion for the testing.

Our testing was a success, as the database accurately represented the inventory of the shelves 90% of the time. However, during testing, we identified some small communication errors between the shelves and the database. Due to these communication errors, the database would sometimes misrepresent the inventory of the shelves. However, we have identified the source of these errors as a lag in messages being sent from the shelves to the database via WIFI. Due to this lag, the database responds to old messages that are being sent from the shelves by adding or removing items from the database when items are not actually being added or removed from the shelves. To solve this error, we are going to investigate why there is a lag in messages. We suspect that this lag is due to an overload of messages, as this lag was not happening when we were only using one shelf. Therefore, we might have to send messages slower or through a different protocol. However, we are happy with our testing results, as we suspect that this lag will be an easy fix.

In the coming weeks, we plan to focus our energy on fixing the aforementioned communication lag and tidying up the shelving units so that they are more professional looking. We also plan to finish the application that will allow store managers and store customers to interact with our product.