

COE306

Embedded Systems Smart Retail Platform

by

Talal Khaled Al-Harbi

202036240-306-51

for

Dr. Hazem Selmi

Intro. To Embedded Systems

Descriptive Abstract

This report describes the development of a Smart Load Monitoring System using an ESP32 microcontroller, HX711 ADC, and load sensors, to be used in retail stores for the purpose of automating purchasing experience. The project aims to accurately measure weights and count items placed on a platform by leveraging the HX711 module and the Wi-Fi connectivity features of the ESP32. The report covers the hardware and software architecture, component design, system integration, and testing, along with a discussion on the results and challenges faced during the project.

18 May 2024

INTRODUCTION

The problem addressed by this project is the need for a precise and connected load monitoring system capable of measuring weights and counting items. The system should be able to communicate data wirelessly, allowing for remote monitoring and control. The requirements include accurate weight measurement, wireless connectivity, and a user-friendly interface for the user of this platform. These requirements are translated into specifications such as the use of an ESP32 microcontroller for its WiFi capabilities, HX711 ADC for precise weight measurement, and Arduino Cloud for data management and remote session access.

ARCHITECTURE

Hardware Architecture

- **ESP32 Microcontroller:** Chosen for its built-in WiFi module, the ESP32 serves as the central unit of the system, handling data acquisition, processing, and communication.
- **HX711 ADC:** A precision 24-bit ADC used to convert the analog signals from the load sensors to digital data. The HX711 operates over the I2C protocol.
- **Load Sensors:** Four degree load sensors mounted on a wooden platform via a 3D printed module. These sensors provide the raw weight data to the HX711.

Software Architecture

- **HX711_ADC Library:** Utilized to interface with the HX711 module, allowing for easy data reading and calibration.
- **Arduino Cloud:** Used for remote cloud variables, visualization, and session control.
- **Custom Code:** Developed for data processing, including weight measurement, item counting, balance calculation, and improper purchasing/loading of items when a session has not started.

DESIGN AND DESIGN ALTERNATIVES

1. **Initial Design:** Large and thin wooden platform. Failed due to inaccurate readings and inaccurate force distribution that affected the output data from the HX711 module.
2. **Final Design:** Smaller and thicker wooden platform. Provided more accurate readings with reduced error.

The final design was selected based on improved accuracy and stability in readings.

WHAT I LEARNED

- **Load Sensor Output:** The output voltage of the load sensors is between 0 to 50 millivolts, that is why we needed the HX711 amplifier module with selectable gain levels of 32, 64, and 128.
- **Item Counting:** The round function was found to be the best method for calculating the number of items due to the high error potential in sensor readings.

PROBLEMS FACED

- **Platform Instability:** The initial thin platform led to inaccurate readings.
- **Sensor Calibration:** Ensuring accurate calibration of the load sensors was challenging, which takes about 5-7 minutes to almost stabilize readings.
- **Wireless Connectivity:** Connecting to the KFUPM Wi-Fi and personal cellular network failed to connect the ESP32 to the cloud. However using a home Wi-Fi did the trick.

RESULT AND DISCUSSION

The final design provided accurate count, and not weight, measurements with minimal error, meeting the project goals. The use of a smaller, thicker platform significantly improved the system's performance. The integration with Arduino Cloud allowed for effective remote monitoring, making the system user-friendly and efficient.

IMPORTANT FUNCTIONS DESCRIPTION

`ArduinoCloud.update()`: Used in every loop to get updates from the cloud, otherwise we won't be able to receive changes on our cloud reading variables.

`LoadCell.update()`: Detects when the HX711 module sends new data serially.

`round(weight / ITEM_WEIGHT)`: Used the round function to calculate the item number

SAMPLE RUNS

```
12 float previousSensorValue; // Store previous sensor value
13
14 //HX711-----
15 const int HX711_dout = 16; //mcu > HX711 dout pin
16 const int HX711_sck = 4; //mcu > HX711 sck pin
17 float initialWeight;
18 float weight;
19 int initialItemNum;
20 int itemNum;
21 int balanceSubtractFactor;
22 #define ITEM_WEIGHT 1497 // Average weight of sold item in the platform
23 #define ITEM_PRICE 5 // The price for one item
24 //HX711 constructor:
25 HX711_ADC LoadCell(HX711_dout, HX711_sck);
26
27 const int calVal_eeepromAddress = 0;
28 unsigned long t = 0;
29 // -----
30
31 void setup() {
32   Serial.begin(9600);
33   delay(1500);
34   initProperties();
35   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
36   setDebugMessageLevel(2);
37   ArduinoCloud.printDebugInfo();
38 }
39
40 void loop() {
41   // -----
42   // Calibration
43   if (t == 0) {
44     Start calibration:
45     Place the load cell on a level stable surface.
46     Remove any load applied to the load cell.
47     Send 't' from serial monitor to set the tare offset.
48     Tare complete
49     Now, place your known mass on the loadcell.
50     Then send the weight of this mass (i.e. 100.0) from serial monitor.
51     // -----
52     // Calibration
53     if (t == 0) {
54       Start calibration:
55       Place the load cell on a level stable surface.
56       Remove any load applied to the load cell.
57       Send 't' from serial monitor to set the tare offset.
58       Tare complete
59       Now, place your known mass on the loadcell.
60       Then send the weight of this mass (i.e. 100.0) from serial monitor.
61     }
62     // -----
63     // Calibration
64     if (t == 0) {
65       Start calibration:
66       Place the load cell on a level stable surface.
67       Remove any load applied to the load cell.
68       Send 't' from serial monitor to set the tare offset.
69       Tare complete
70       Now, place your known mass on the loadcell.
71       Then send the weight of this mass (i.e. 100.0) from serial monitor.
72     }
73     // -----
74     // Calibration
75     if (t == 0) {
76       Start calibration:
77       Place the load cell on a level stable surface.
78       Remove any load applied to the load cell.
79       Send 't' from serial monitor to set the tare offset.
80       Tare complete
81       Now, place your known mass on the loadcell.
82       Then send the weight of this mass (i.e. 100.0) from serial monitor.
83     }
84     // -----
85     // Calibration
86     if (t == 0) {
87       Start calibration:
88       Place the load cell on a level stable surface.
89       Remove any load applied to the load cell.
90       Send 't' from serial monitor to set the tare offset.
91       Tare complete
92       Now, place your known mass on the loadcell.
93       Then send the weight of this mass (i.e. 100.0) from serial monitor.
94     }
95     // -----
96     // Calibration
97     if (t == 0) {
98       Start calibration:
99       Place the load cell on a level stable surface.
100      Remove any load applied to the load cell.
101      Send 't' from serial monitor to set the tare offset.
102      Tare complete
103      Now, place your known mass on the loadcell.
104      Then send the weight of this mass (i.e. 100.0) from serial monitor.
105    }
106    // -----
107    // Calibration
108    if (t == 0) {
109      Start calibration:
110      Place the load cell on a level stable surface.
111      Remove any load applied to the load cell.
112      Send 't' from serial monitor to set the tare offset.
113      Tare complete
114      Now, place your known mass on the loadcell.
115      Then send the weight of this mass (i.e. 100.0) from serial monitor.
116    }
117    // -----
118    // Calibration
119    if (t == 0) {
120      Start calibration:
121      Place the load cell on a level stable surface.
122      Remove any load applied to the load cell.
123      Send 't' from serial monitor to set the tare offset.
124      Tare complete
125      Now, place your known mass on the loadcell.
126      Then send the weight of this mass (i.e. 100.0) from serial monitor.
127    }
128    // -----
129    // Calibration
130    if (t == 0) {
131      Start calibration:
132      Place the load cell on a level stable surface.
133      Remove any load applied to the load cell.
134      Send 't' from serial monitor to set the tare offset.
135      Tare complete
136      Now, place your known mass on the loadcell.
137      Then send the weight of this mass (i.e. 100.0) from serial monitor.
138    }
139    // -----
140    // Calibration
141    if (t == 0) {
142      Start calibration:
143      Place the load cell on a level stable surface.
144      Remove any load applied to the load cell.
145      Send 't' from serial monitor to set the tare offset.
146      Tare complete
147      Now, place your known mass on the loadcell.
148      Then send the weight of this mass (i.e. 100.0) from serial monitor.
149    }
150    // -----
151    // Calibration
152    if (t == 0) {
153      Start calibration:
154      Place the load cell on a level stable surface.
155      Remove any load applied to the load cell.
156      Send 't' from serial monitor to set the tare offset.
157      Tare complete
158      Now, place your known mass on the loadcell.
159      Then send the weight of this mass (i.e. 100.0) from serial monitor.
160    }
161    // -----
162    // Calibration
163    if (t == 0) {
164      Start calibration:
165      Place the load cell on a level stable surface.
166      Remove any load applied to the load cell.
167      Send 't' from serial monitor to set the tare offset.
168      Tare complete
169      Now, place your known mass on the loadcell.
170      Then send the weight of this mass (i.e. 100.0) from serial monitor.
171    }
172    // -----
173    // Calibration
174    if (t == 0) {
175      Start calibration:
176      Place the load cell on a level stable surface.
177      Remove any load applied to the load cell.
178      Send 't' from serial monitor to set the tare offset.
179      Tare complete
180      Now, place your known mass on the loadcell.
181      Then send the weight of this mass (i.e. 100.0) from serial monitor.
182    }
183    // -----
184    // Calibration
185    if (t == 0) {
186      Start calibration:
187      Place the load cell on a level stable surface.
188      Remove any load applied to the load cell.
189      Send 't' from serial monitor to set the tare offset.
190      Tare complete
191      Now, place your known mass on the loadcell.
192      Then send the weight of this mass (i.e. 100.0) from serial monitor.
193    }
194    // -----
195    // Calibration
196    if (t == 0) {
197      Start calibration:
198      Place the load cell on a level stable surface.
199      Remove any load applied to the load cell.
200      Send 't' from serial monitor to set the tare offset.
201      Tare complete
202      Now, place your known mass on the loadcell.
203      Then send the weight of this mass (i.e. 100.0) from serial monitor.
204    }
205    // -----
206    // Calibration
207    if (t == 0) {
208      Start calibration:
209      Place the load cell on a level stable surface.
210      Remove any load applied to the load cell.
211      Send 't' from serial monitor to set the tare offset.
212      Tare complete
213      Now, place your known mass on the loadcell.
214      Then send the weight of this mass (i.e. 100.0) from serial monitor.
215    }
216    // -----
217    // Calibration
218    if (t == 0) {
219      Start calibration:
220      Place the load cell on a level stable surface.
221      Remove any load applied to the load cell.
222      Send 't' from serial monitor to set the tare offset.
223      Tare complete
224      Now, place your known mass on the loadcell.
225      Then send the weight of this mass (i.e. 100.0) from serial monitor.
226    }
227    // -----
228    // Calibration
229    if (t == 0) {
230      Start calibration:
231      Place the load cell on a level stable surface.
232      Remove any load applied to the load cell.
233      Send 't' from serial monitor to set the tare offset.
234      Tare complete
235      Now, place your known mass on the loadcell.
236      Then send the weight of this mass (i.e. 100.0) from serial monitor.
237    }
238    // -----
239    // Calibration
240    if (t == 0) {
241      Start calibration:
242      Place the load cell on a level stable surface.
243      Remove any load applied to the load cell.
244      Send 't' from serial monitor to set the tare offset.
245      Tare complete
246      Now, place your known mass on the loadcell.
247      Then send the weight of this mass (i.e. 100.0) from serial monitor.
248    }
249    // -----
250    // Calibration
251    if (t == 0) {
252      Start calibration:
253      Place the load cell on a level stable surface.
254      Remove any load applied to the load cell.
255      Send 't' from serial monitor to set the tare offset.
256      Tare complete
257      Now, place your known mass on the loadcell.
258      Then send the weight of this mass (i.e. 100.0) from serial monitor.
259    }
260    // -----
261    // Calibration
262    if (t == 0) {
263      Start calibration:
264      Place the load cell on a level stable surface.
265      Remove any load applied to the load cell.
266      Send 't' from serial monitor to set the tare offset.
267      Tare complete
268      Now, place your known mass on the loadcell.
269      Then send the weight of this mass (i.e. 100.0) from serial monitor.
270    }
271    // -----
272    // Calibration
273    if (t == 0) {
274      Start calibration:
275      Place the load cell on a level stable surface.
276      Remove any load applied to the load cell.
277      Send 't' from serial monitor to set the tare offset.
278      Tare complete
279      Now, place your known mass on the loadcell.
280      Then send the weight of this mass (i.e. 100.0) from serial monitor.
281    }
282    // -----
283    // Calibration
284    if (t == 0) {
285      Start calibration:
286      Place the load cell on a level stable surface.
287      Remove any load applied to the load cell.
288      Send 't' from serial monitor to set the tare offset.
289      Tare complete
290      Now, place your known mass on the loadcell.
291      Then send the weight of this mass (i.e. 100.0) from serial monitor.
292    }
293    // -----
294    // Calibration
295    if (t == 0) {
296      Start calibration:
297      Place the load cell on a level stable surface.
298      Remove any load applied to the load cell.
299      Send 't' from serial monitor to set the tare offset.
300      Tare complete
301      Now, place your known mass on the loadcell.
302      Then send the weight of this mass (i.e. 100.0) from serial monitor.
303    }
304    // -----
305    // Calibration
306    if (t == 0) {
307      Start calibration:
308      Place the load cell on a level stable surface.
309      Remove any load applied to the load cell.
310      Send 't' from serial monitor to set the tare offset.
311      Tare complete
312      Now, place your known mass on the loadcell.
313      Then send the weight of this mass (i.e. 100.0) from serial monitor.
314    }
315    // -----
316    // Calibration
317    if (t == 0) {
318      Start calibration:
319      Place the load cell on a level stable surface.
320      Remove any load applied to the load cell.
321      Send 't' from serial monitor to set the tare offset.
322      Tare complete
323      Now, place your known mass on the loadcell.
324      Then send the weight of this mass (i.e. 100.0) from serial monitor.
325    }
326    // -----
327    // Calibration
328    if (t == 0) {
329      Start calibration:
330      Place the load cell on a level stable surface.
331      Remove any load applied to the load cell.
332      Send 't' from serial monitor to set the tare offset.
333      Tare complete
334      Now, place your known mass on the loadcell.
335      Then send the weight of this mass (i.e. 100.0) from serial monitor.
336    }
337    // -----
338    // Calibration
339    if (t == 0) {
340      Start calibration:
341      Place the load cell on a level stable surface.
342      Remove any load applied to
```

CONCLUSION

The Smart Load Monitoring System successfully achieved its goals of accurate weight measurement and item counting with wireless connectivity. The project highlighted the importance of sensor calibration and platform stability. Future improvements could include exploring different materials for the platform and enhancing the user interface for easier calibration and monitoring.