SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Arduino IOT Based Inventory Management System Using Load Cell and NodeMCU

Muhammad Naufal Mansor[1,*], Noor Anida Abu Talib[1], Syahrul Affandi Saidi[1], Wan Azani Mustafa[1], Nurul Fakhirah Zamri[1]

[1] Faculty of Electrical Engineering Technology, Universiti Malaysia Perlis, UniCITI Alam Campus, Sungai Chuchuh, 02100 Padang Besar, Perlis, Malaysia

ARTICLE INFO

Article history:
Received 15 May 2023
Received in revised form 6 September 2023
Accepted 13 September 2023
Available online 1 October 2023

Keywords:
Air-cushion vehicle; hovercraft; insulation foam; bag skirt

ABSTRACT

Nowadays, everything is made simpler with information and communication technological advancements. It is preferable to track and monitor using devices rather than perform it manually. This resulted in the rapid growth of Internet of Things (IoT) technology and relevant markets. Low cost IoT products has made access to IoT much easier and desirable. These low cost IoT devices and related technologies are widely used in areas such as educational, transportation, tracking, inventory management and many more. The use of Arduino and RFID in the inventory management system lacks in some areas including hardware limitations. In conjunction to the limitation of using an Arduino and RFID technology, this project aims to develop an IoT based inventory management system that incorporates the uses of a NodeMCU and a load cell. In comparison of the NodeMCU to an Arduino, the NodeMCU stands out with the built in Wi-Fi module with much higher processor and additional properties of it being much smaller. While the use of a load cell is much more convenient as to suit all kinds of inventory management needs compared to the use of RFID that suits better to larger scale businesses with larger inventory and massive stocks. Towards the end, this project is expected to ease inventory management by the implementation of IoT with IoT Based Inventory Management System using Load Cell and NodeMCU. The project will generate the inventory count and automatically stores data in the cloud platform. These data can be accessed with internet connection. The project also alerts users when the inventory is low or high in balance. The output of the project is that the project's working prototype was successfully developed. Overall, the project is a success as all the objectives of the project was successfully achieved.

1. Introduction

There are numerous complications that can feasibly turn up in inventory management systems. Several regular drawbacks include unreliable tracking, imprecise data, restricted visibility, overstocking or understocking, documentation fault, and deficient interaction due to manual inventory management approach. To counteract the conundrum involving the problems concerning

* *Corresponding author.*
E-mail address: naufal@unimap.edu.my

https://doi.org/10.37934/araset.32.3.1225

the above manual inventory management method, several resolutions should be taken in action. It incorporates automated tracking, modernizing platforms utilized for inventory management system to a cloud-based platform. An inventory management system incorporated with IoT will be developed to solve this problem. The IoT Based Inventory Management System using Load Cell and NodeMCU will ease inventory management systems as it automatically generates stock availability and is accessible online through an IoT based cloud platform ThingSpeak. Data history and stock status will also be generated, and email notification will be sent whenever a certain threshold have been reached by the system.

Most large-scale businesses have already implemented IoT to ease their inventory management system. However, some small businesses still strive managing their inventory manually and nowadays it has become one of the major problems to maintain, record and keep track of common daily inventory. An Arduino microcontroller was used in the inventory management system developed for a photovoltaic panel powered freezer in the aim of promoting energy efficiency and responsible use of food [1-8]. The use of an Arduino based inventory management system lacks a built in Wi-Fi module and a much lower processor compared to a NodeMCU. A smart solution for an inventory management system incorporating a Radio Frequency Identification (RFID) system was developed [9,10]. However, the use of an RFID is quite costly and mostly used for large scale businesses with larger inventory and massive stocks. This is because it is quite impossible to tag most small items and not to mention uncountable items. Thus, the use of a load cell is the more convenient.

A study was conducted on developing an automatic inventory management system using Raspberry Pi and Arduino [11-14]. In the article, color sensor was utilized for product categorization while an ultrasound and an infrared sensor to prompt product availability in the automatic inventory management system. A database was built using MySQL for data storage. The article states that the interconnection between a web server and an application (i.e., Bluetooth and Wi-Fi) is employed by wireless communication devices as in the study, the authors found prediction of 6.1 billion smart phone users worldwide by 2020 according to recent statistics of Ericson. Findings from the study concludes that the system can recognize the color sensor correctly and the Color Judge Algorithm is properly performed. Likewise, a paper on the establishment of an inventory management system for photovoltaic-powered freezer points that their goal is to elevate the energy efficiency and responsibility towards food [16-18]. Stated in their research, nearly 1.3 billion tons of food across the world is wasted annually which is equivalent to one third of produced human food. The developed system is constituted of a GSM module attached to an Arduino microcontroller. It uses a digital temperature sensor DS18B20 to detect the food temperature and load cells with maximum capacity of 20Kg and 200Kg to determine the weight of the food. HX711 load cell amplifier is used to amplify the output and acrylic glass was chosen as a platform for the load cells. The study come up with a high percentage of success rate and showed a negligible amount of percentage error along with the system being able to function perfectly.

Correspondingly, previous researchers present an inventory management system for varying types of stocks be it in solid or liquid state [18-21]. The project development implements an ultrasonic transducer HC-SR04 and Raspberry Pi Model B. The inventory will be computed by the system and a mail will be sent for restocking. The current stocks availability will also appear on a web page hosted by their system. The authors wrote that one of the aspects that sets them apart from others is the functional operation of the ultrasound transducer in inventory evaluation process. The exact same transducer can handle both solid and liquid type stocks. It is also able to operate on battery power which gives extra bonus points to industries that do not depend solely on electric power. In the end of the study, the system proves that it is cost efficient. The use of an ultrasonic sensor simplifies stock management system for varying types of products. The system is also found to be self-sustained as it can avoid inadequate supply.

2. System Overview

The system block diagram shows the basic design structure of the IoT Based Inventory Management System using Load Cell and NodeMCU. This design structure is diverged into three main order which are the input, process, and the output of the project. The input of the project as shown in the system block diagram in Figure 1 below is the load cell. Loads will be placed on top of a platform used by the load cell for weight detection. The output signal produced by the load cell will be amplified by the HX711 load cell amplifier module. The microcontroller then automatically generates the weight of items and the total number of pieces in the inventory. The data will then be displayed on an OLED display and on ThingSpeak cloud platform. IFTTT is integrated with ThingSpeak for email alerts when the system passed the threshold limit that has been set.

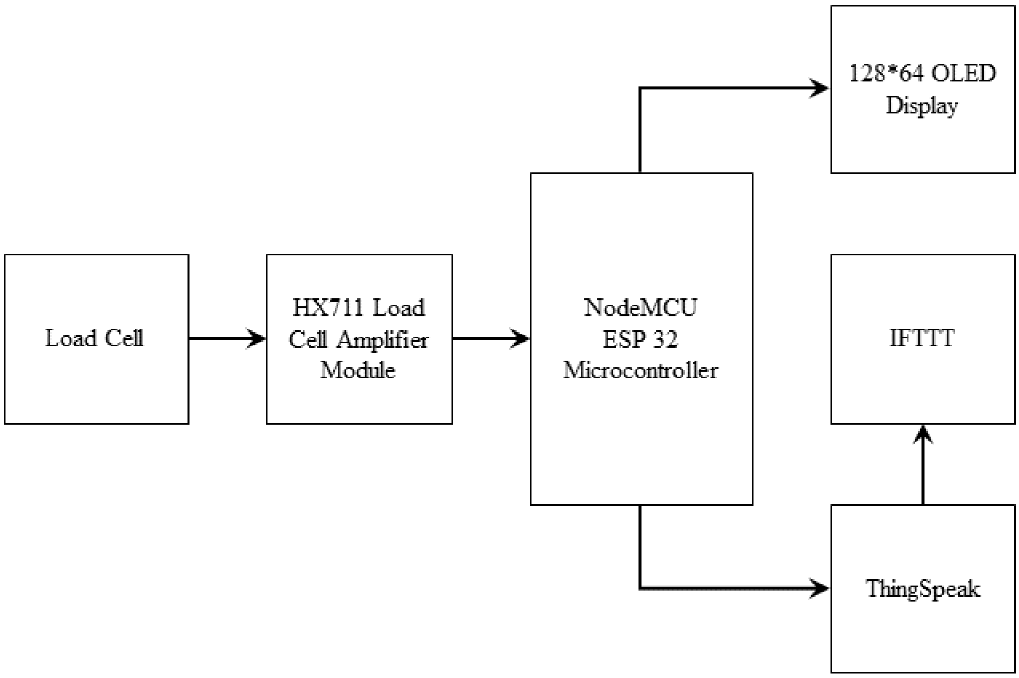


Fig. 1. The Block Diagram of the Project

2.1 System Flow Diagram

The system flow diagram illustrates the operating procedure of the IoT Based Inventory Management System using Load Cell and NodeMCU. Based on the system flow diagram shown in Figure 2 below, the process starts when the load cell detects the weights of items and produce an output signal that will then be amplified by a HX711 load cell amplifier module. NodeMCU ESP32 microcontroller will then generate the item weight and the total number of pieces that is then displayed on a 128*64 OLED display. The data will also be displayed and can be monitored online through ThingSpeak platform. IFTTT is integrated with ThingSpeak for email alerts when certain threshold is reached by the system. The threshold limits set are when the total number of pieces in the system is less than 3 pieces and when the total weight of items exceeds 900g. When the total number of pieces is less than 3 pieces, IFTTT is triggered and will send ‘minimum’ email alert for users to load more items. When the total weight of items exceeds 900g, IFTTT will send a ‘maximum’ email alert for users to stop loading items or remove some items. The email notification cycle will loop until there are actions taken by user.

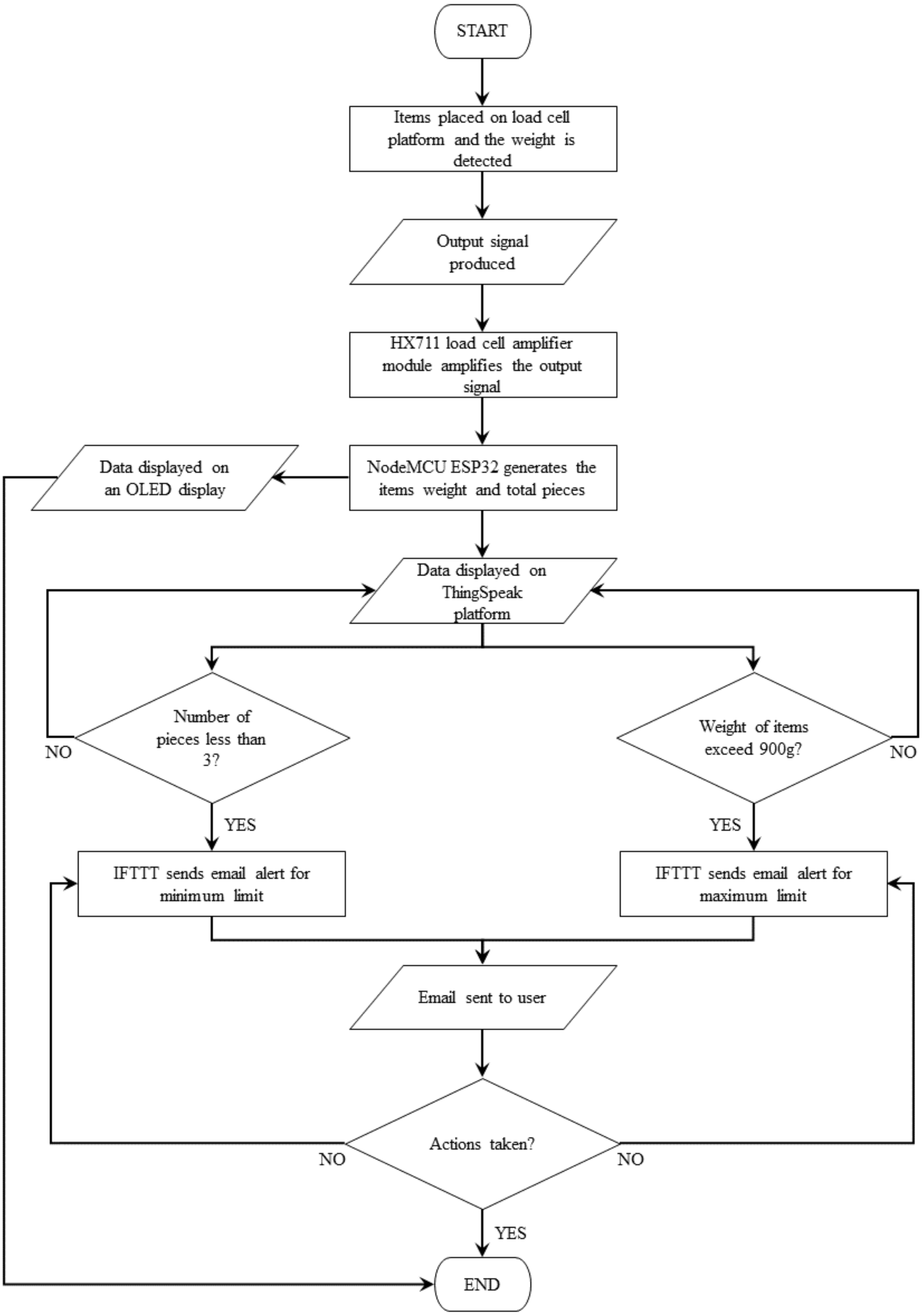


Fig. 2. System flow diagram of the project

2.2 Microcontroller Program

Arduino IDE software version 1.8.15 was used for the microcontroller program. The program code written was saved in INO file (.ino). The script for the project includes the requisite libraries for the program to be able to work with certain components in the project. The libraries include Wire.h, Adafruit_GFX.h, Adafruit_SSD1306.h, WiFi.h, ThingSpeak.h, HX711_ADC.h, and EEPROM.h. These libraries need to be initialized as they each have their own designated functions as demonstrated in Table 1 below such as to establish Wi-Fi connections and for data storage in the Electrically Erasable Programmable Read Only Memory (EEPROM).

Table 1
Programming library functions

Programming library functions Libraries	Functions
<Wire.h>	To operate with I2C based OLED module
<Adafruit_GFX.h>	To operate with OLED module
<Adafruit_SSD1306.h>	To operate with OLED module
<WiFi.h>	To establish Wi-Fi connection to the NodeMCU ESP32 board
"ThingSpeak.h"	To enable the ThingSpeak function for the system
<HX711_ADC.h>	To enable the HX711 function for the system
<EEPROM.h>	To store data in ESP32 EEPROM

2.3 Circuitry Design

The Figure 3 below shows the schematic diagram constructed using the Fritzing software. The connections for the load cell to the HX711 load cell amplifier module is as follows. Red to E+ pin, white to A+ pin, black to E-, and finally green to A-. The Dout and the CLK pin of the HX711 module is connected to the NodeMCU ESP 32 GPIO23 and GPIO 19 pins respectively. The OLED Display's SDA and SCK pins are also connected to the ESP 32 at the GPIO 21 and GPIO 22 respectively. The input voltage source for the system is 5V.

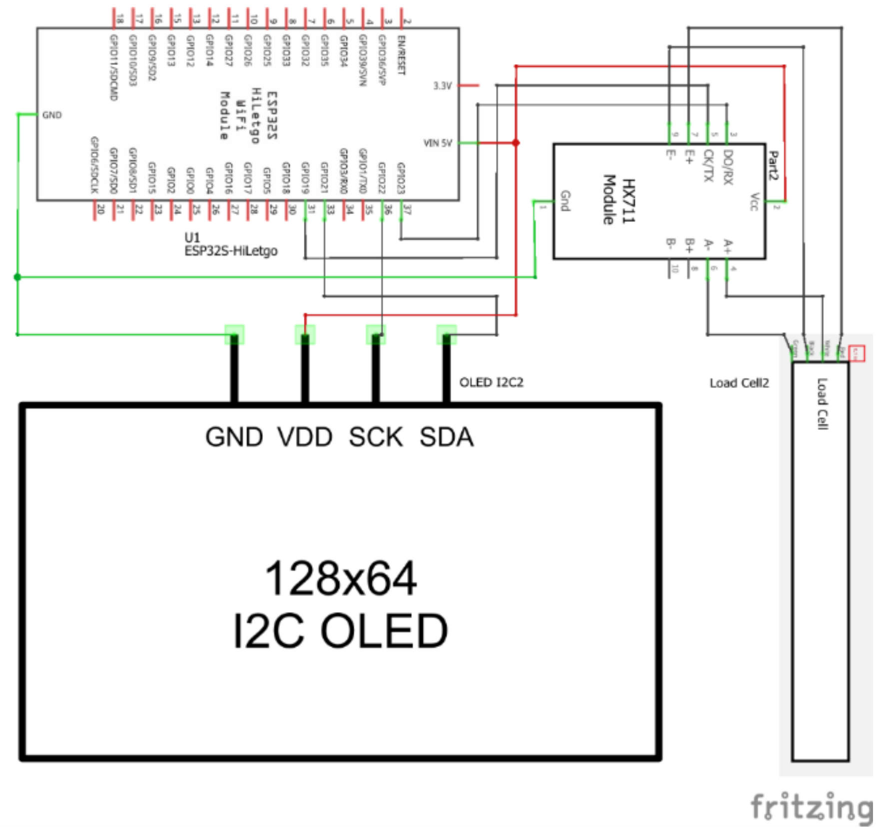


Fig. 3. Schematic diagram of the project

2.4 Data Sample

The sample used for the IoT Based Inventory Management System using Load Cell and NodeMCU is a bolt. The total number of samples used was 22 samples as shown in the Figure 4 below. The samples weight was collected by using a precision weight scale. The precision weight scale used has a 3Kg weight limit with 0.1g tolerance. The bolts weighted around 43.8g to 44.1g. The samples have the lowest weight difference of only 0.3g. The weight of each of the samples was collected and tabulated in the Table 2 below. The minimum weight of the sample is 43.8g. Therefore, the most suitable weight to input in the IoT based Inventory Management System using Load Cell and NodeMCU is between 43.5g – 43.7g. This is due to the tolerance and sensitivity of the load cell. The platform stability is also considered as it affects the accuracy of the load cell.

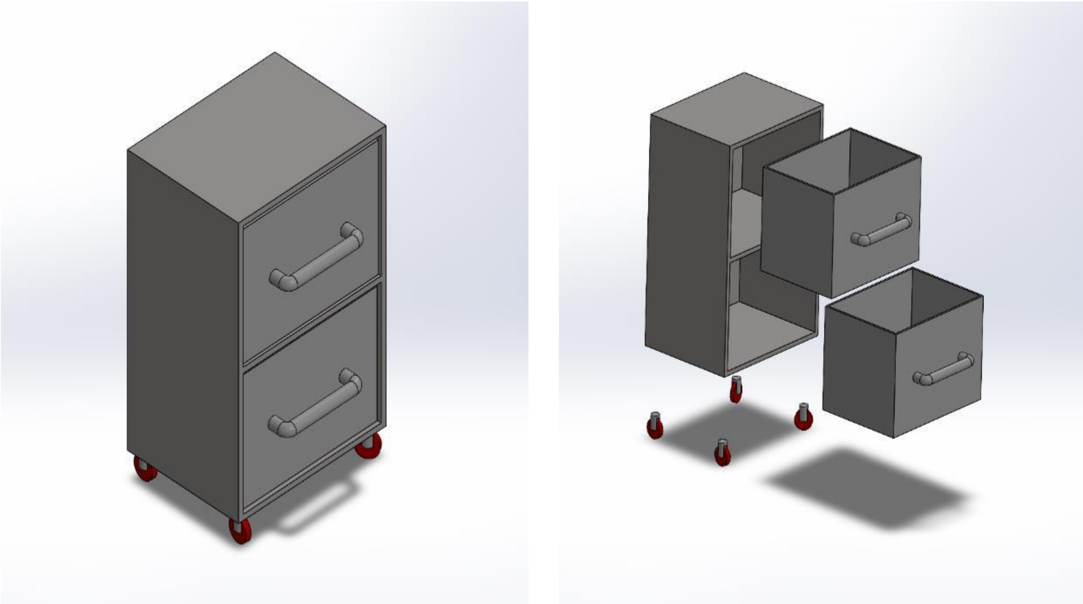
Fig. 4. Samples used for the project

Table 2
Sampling data table Sample

No.	Sample Weight (g)
1	43.8
2	43.9
3	43.8
4	44.1
5	43.8
6	43.9
7	44.0
8	43.9
9	44.0
10	43.8
11	43.8
12	43.8
13	43.8
14	44.0
15	43.9
16	44.1
17	43.9
18	43.9
19	43.9
20	43.9
21	43.8
22	44.1

2.5 Mechanical Design

The software mechanical design for the IoT Based Inventory Management System using Load Cell and NodeMCU was done by using the SolidWorks Computer Aided Design (CAD) software. The design of the prototype consists of the main body of the drawer, the compartment for the drawer, the wheel brackets and the wheels. The body of the prototype is of 312mm, 235mm, and 480mm in length, width, and height respectively. The thickness of the body frame is 10mm on all sides. The compartment of the prototype is of 291mm in length, 210mm in width, and 215mm in height. The thickness of the compartment is 5mm on all sides. Based on the mechanical part designs constructed in the SolidWorks, the complete assembly of the prototype was done. As shown in Figure 5(a) below, the complete assembly of the prototype includes a body of the prototype, two compartments, and four wheels. From the complete assembly of the prototype, the exploded view of the prototype was constructed as shown in Figure 5(b) below.



(a) (b)

Fig. 5. Prototype complete assembly and exploded view; (a) Prototype complete assembly, (b) Prototype exploded view

The hardware mechanical design of the IoT Based Inventory Management System using Load Cell and NodeMCU was done based on the constructed designs on SolidWorks. The material used for the body of the prototype is a fiberboard with a white outer layer finishing. The boards were aligned and screwed together according to the designs. The boards used for the compartments are a combination of a medium density fiberboard (MDF) and a polyvinyl chloride (PVC) foam board. The Figure 6 below shows the complete assembly of the prototype. The inside of the compartment is where the load cell platform is placed.

Fig. 6. Complete assembly of the prototype

For the load cell platform, holes were drilled onto the PVC foam board to hold the load cell. The load cell was then mounted on the platform. The Figure 7 below shows the load cell platform position inside the compartment. A border was placed on top of the platform to avoid the items falling off or moving around too much. The border will hold the items in place. Figure 7(a) below shows the top view of the boarder for the load cell platform.

(a) (b) (c)

Fig. 7. Load cell platform view; (a) front view, (b) top view, (c) side view

3. Results and Discussion

The result obtained from the ThingSpeak are the total number of item pieces and the total weight of the items. Figure 8 below shows the ThingSpeak result for one sample. The numeric display widget was chosen to display the total weight and total pieces of item in the system. Below the widgets are the field charts based on the data displayed in the widget. Figure 9 below shows the result for fifteen samples. The red dot on the lines is the time each data was sent to ThingSpeak.

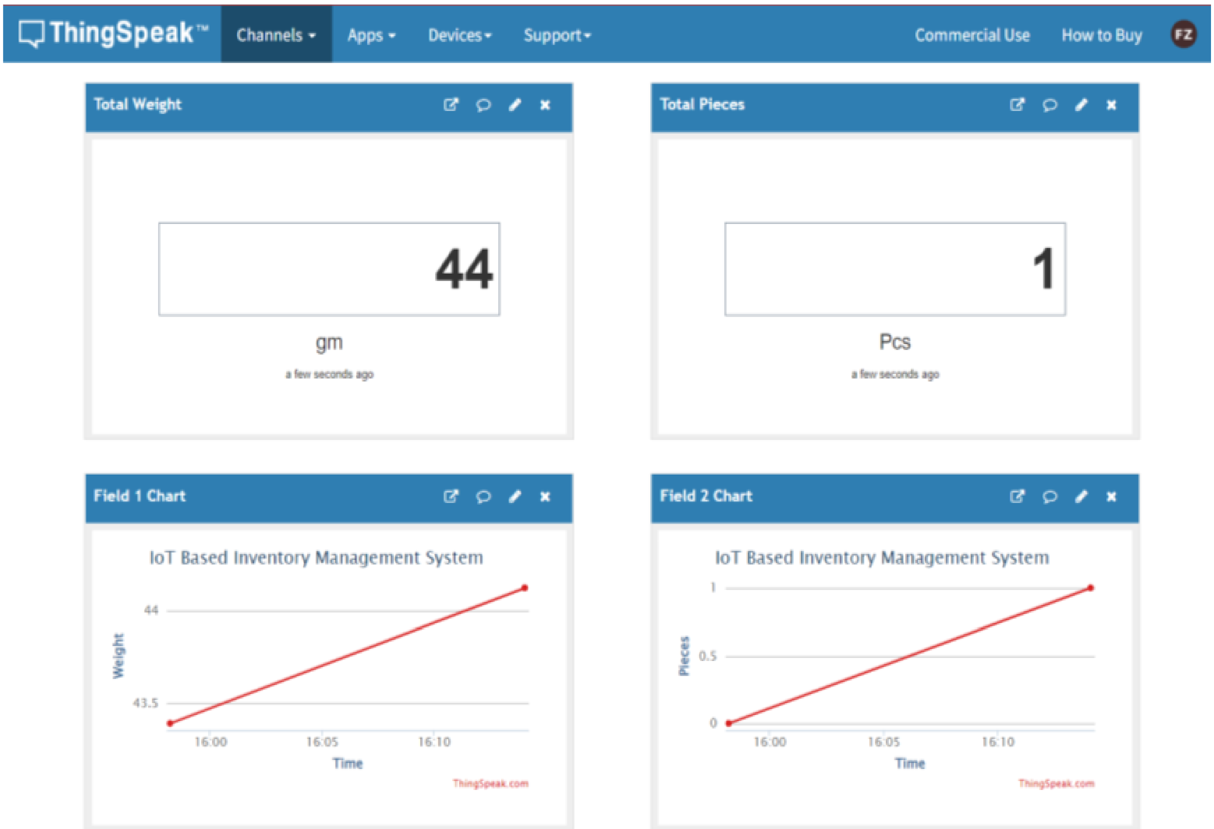


Fig. 8. ThingSpeak display for one sample

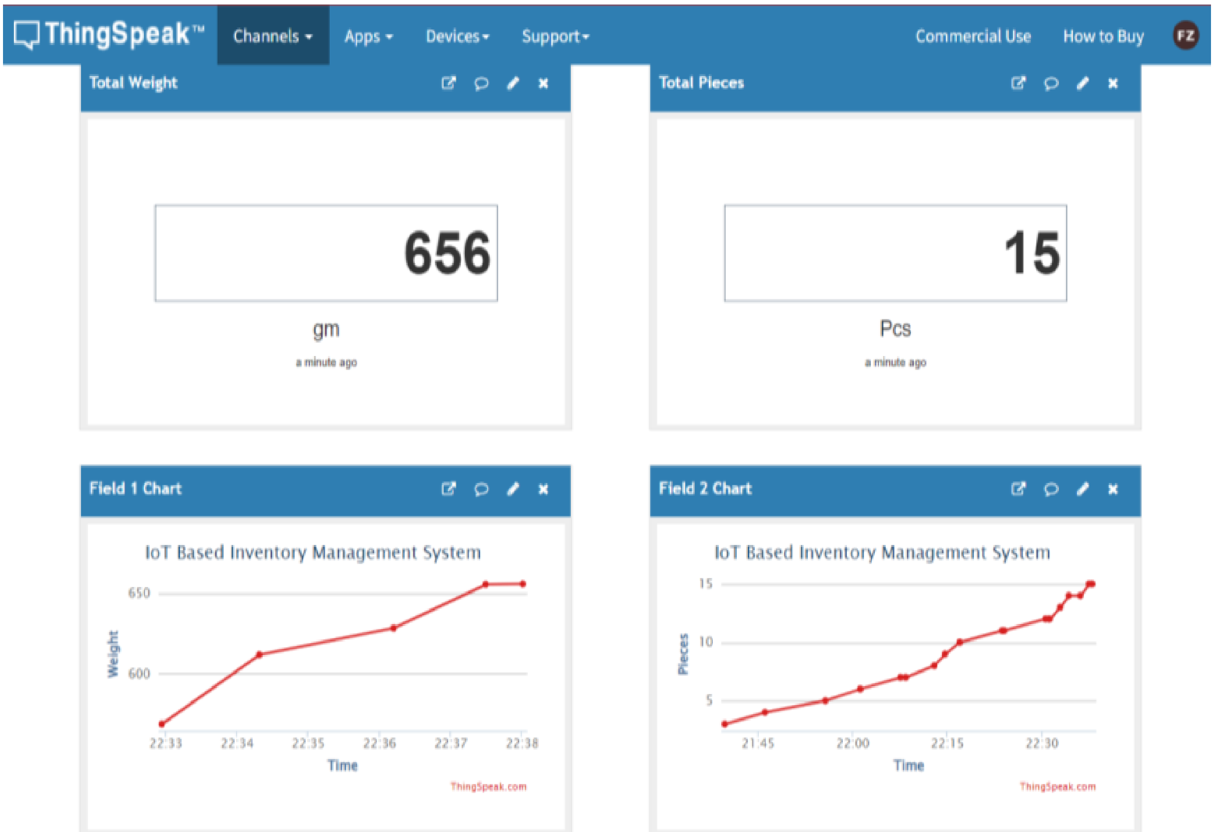
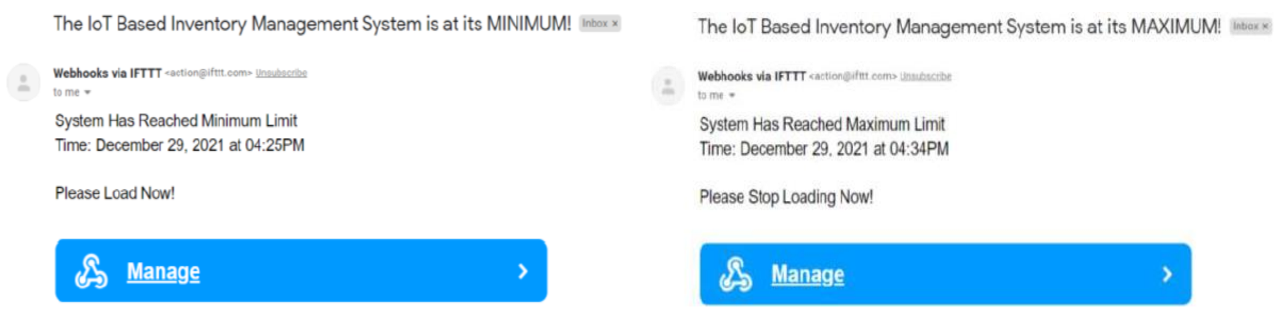


Fig. 9. ThingSpeak display for fifteen samples

For the IoT Based Inventory Management System using Load Cell and NodeMCU, the IFTTT is integrated with the ThingSpeak data to send email alerts. The result for the IFTTT comprises of the email sent when the threshold was reached by the system. The Figure 10(a) below shows the email alert for when the system is in minimum condition. This is when the total number of pieces displayed on ThingSpeak is less than 3 pieces. The maximum email alert as shown in Figure 10(b) below is for when the system is in maximum condition where the total weight of items displayed on ThingSpeak exceeds 900g.



(a) (b)

Fig. 10. IFTTT email alerts; (a) IFTTT minimum email alert, (b) IFTTT maximum email alert

3.1 Output Data

The IoT Based Inventory Management system using Load Cell and NodeMCU prototype produced several output data that was collected and tabulated. The first output data collected was the total cumulative weight of the samples by using a precision weight scale. The second data collected was the total weight and total number of pieces displayed on the OLED display of the prototype. The values measured by using a weight scale and the values displayed on the OLED display does not vary a lot. The total weight and the total number of pieces data displayed in ThingSpeak platform are also tabulated as well. The values displayed on ThingSpeak platform is rounded off thus there are some differences between the values of the ones measured by using a weighing scale. The IFTTT action data was then collected. These data were accumulated from the email alerts sent by IFTTT when the threshold limit was reached by the system in ThingSpeak. As seen in the Table 3 below, when the total number of samples in ThingSpak is less than three (3), the IFTT is triggered and sends a minimum email alert. When the total weight of the samples in ThingSpeak exceeds 900g, the IFTTT is triggered and sends a maximum email alert.

Table 3
Output data table for Weight scale, OLED Display, ThingSpeak and IFTTT trigger

No of sample (Pcs)	Total wight (g) (weight scale)	Total weight (g) (OLED Display)	Total Pieces (Pcs) (OLED Display)	Total Weight (g) (ThingSpeak)	Total Pieces (Pcs) (ThingSpeak)	IFTTT Trigger
1	43.8	43.6	1	44	1	Minimum Alert
2	87.6	84.6	2	88	2	Minimum Alert
3	131.4	132.3	3	132	3	No Trigger
4	175.4	175.5	4	176	4	No Trigger
5	219.9	219.8	5	219	5	No Trigger
6	263.0	263.0	6	263	6	No Trigger
7	307.0	306.8	7	307	7	No Trigger
8	350.1	350.1	8	350	8	No Trigger
9	394.6	394.5	9	395	9	No Trigger
10	438.6	438.4	10	438	10	No Trigger
11	482.4	481.7	11	481	11	No Trigger
12	526.3	525.1	12	525	12	No Trigger
13	570.0	568.7	13	569	13	No Trigger
14	614.0	612.8	14	612	14	No Trigger
15	658.0	656.1	15	656	15	No Trigger
16	702.1	701.9	16	699	16	No Trigger
17	746.2	745.8	17	746	17	No Trigger
18	790.0	790.5	18	790	18	No Trigger
19	834.1	833.9	19	834	19	No Trigger
20	877.9	878.1	20	878	20	No Trigger
21	921.8	921.9	21	922	21	Maximum Alert
22	965.8	965.9	22	965	22	Maximum Alert

From the data Table 3 above, a graphical representation of the data was constructed to better analyze the performance and observing the difference between both display output. The graph in Figure 11 below illustrates the output display data between OLED display and ThingSpeak and the values between both display output is quite similar.

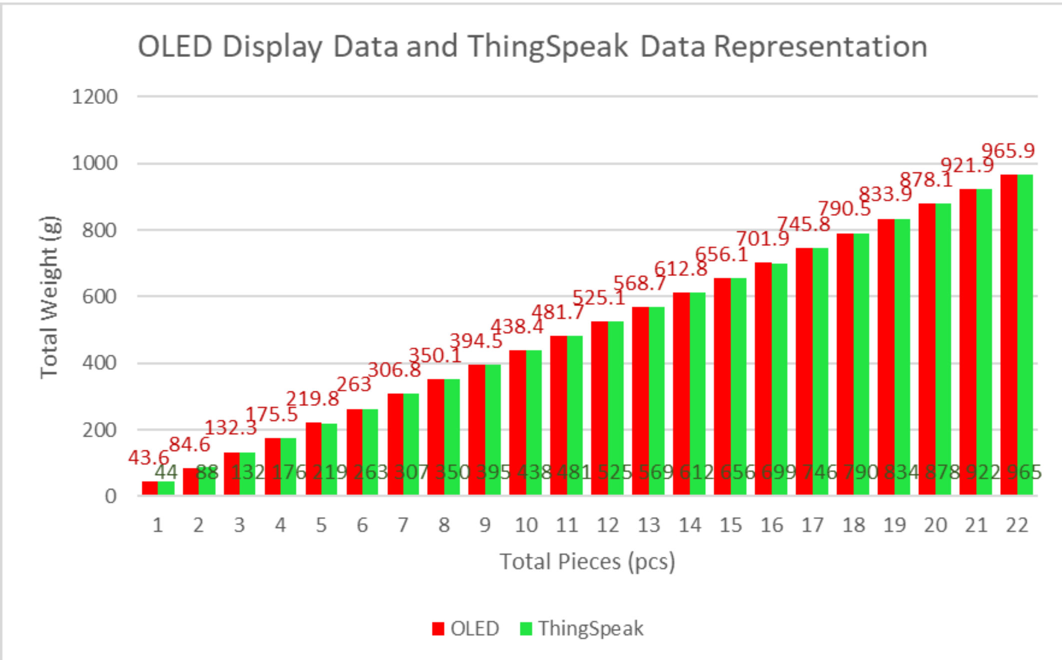


Fig. 11. OLED Display and ThingSpeak data representation graph.

For the IFTTT data analysis, the minimum alert trigger occurs in the beginning and the maximum alert trigger occurs at the end. From the tabulated data, a graphical representation was constructed to better understand the trigger pattern of the IFTTT based on the ThingSpeak output display data. The graph in the Figure 12 below illustrates the IFTTT trigger depending on the ThingSpeak data. As shown, when the total number of pieces displayed in ThingSpeak is less than 3 pieces, the IFTTT was triggered and sends a minimum email alert. Afterwards, when the total number of pieces exceeds 3 pieces, the IFTTT was not triggered. Then, when the total weight of the system exceeds 900g, the IFTTT was triggered and sends a maximum email alert.

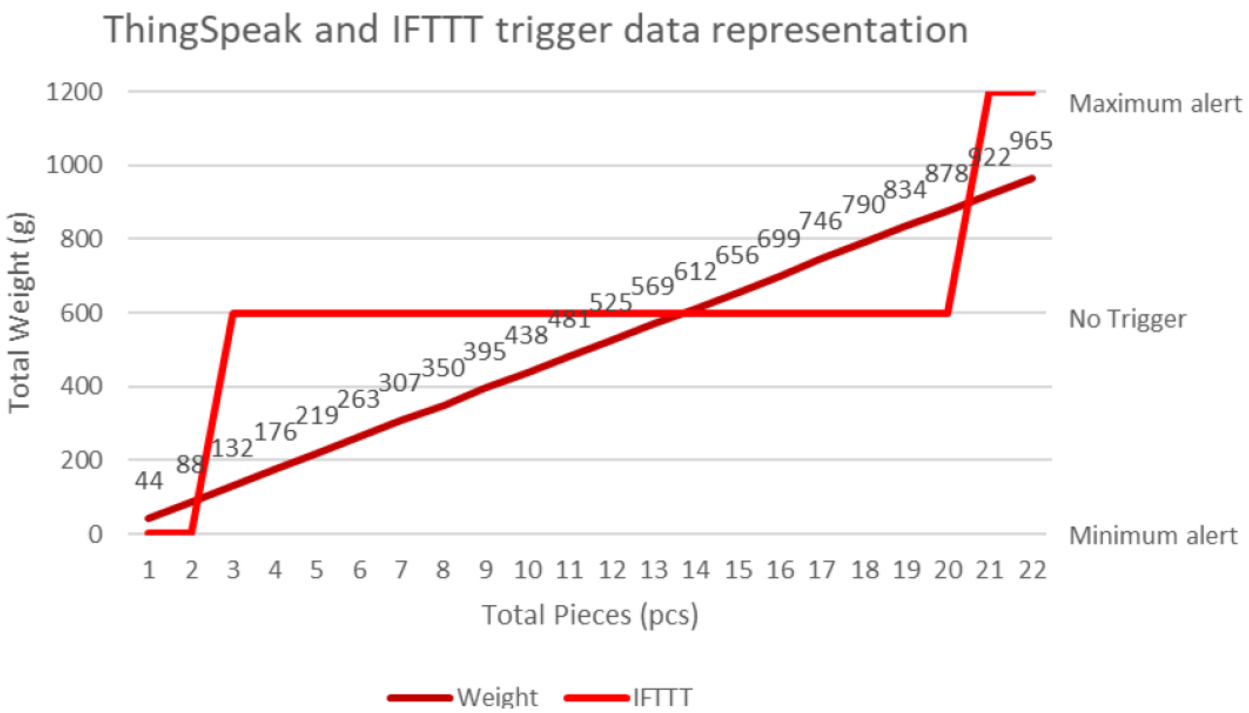


Fig. 12. ThingSpeak and IFTTT trigger data representation graph

4. Conclusions

The conclusions that can be made based on the development of the project is that the design process of the IoT Based Inventory Management System using Load Cell and NodeMCU was completed successfully. The designs done can be divided into two categories which are the software and the hardware designs. The system was monitored by using ThingSpeak cloud platform. The data generated from the prototype was sent to the ThingSpeak. From the ThingSpeak platform, the IFTTT triggers were set, and email alerts was retrieved upon reaching the threshold limits set. The performance of the IoT Based Inventory Management System using Load Cell and NodeMCU prototype was then analyzed. Data was collected from the experimental setup done using the bolts as samples.

To summarize all that has been indicated above, all the objectives of the IoT Based Inventory Management System using Load Cell and NodeMCU which is to design an IoT Based Inventory Management System using Load Cell and NodeMCU, to monitor the inventory management system with ThingSpeak cloud platform, and to analyze the performance of the IoT Based Inventory Management System using Load Cell and NodeMCU prototype was successfully achieved.

Acknowledgement
The authors would like to thank the Universiti Malaysia Perlis (UNIMAP) for giving the opportunity to conduct this research.

References
[1] Basa, Janus Jade A., Patrick Lourenz G. Cu, Nathaniel N. Malabag, Luigi Angelo V. Naag, Dan Frederico P. Abacco, Mar Jun M. Siquihod, Gilfred Allen Madrigal, and Lean Karlo S. Tolentino. "Smart Inventory Management System for Photovoltaic-Powered Freezer Using Wireless Sensor Network." *International Journal of Emerging Trends in Engineering Research* 7, no. 10 (2019): 393-397. https://doi.org/10.30534/ijeter/2019/057102019
[2] Alwadi, Ali, Amjad Gawanmeh, Sazia Parvin, and Jamal N. Al-Karaki. "Smart solutions for RFID based inventory management systems: A survey." *Scalable Computing: Practice and Experience* 18, no. 4 (2017): 347-360. https://doi.org/10.12694/scpe.v18i4.1333
[3] Choi, Eun-Soo, Min-Soo Kang, Yong Gyu Jung, and Jean Kyung Paik. "Implementation of IoT-based Automatic Inventory Management System." *International Journal of Advanced Culture Technology* 5, no. 1 (2017): 70-75. https://doi.org/10.17703/IJACT.2017.5.1.70
[4] Jayanth, S., M. B. Poorvi, and M. P. Sunil. "Inventory management system using IOT." In *Proceedings of the First International Conference on Computational Intelligence and Informatics: ICCII 2016*, pp. 201-210. Springer Singapore, 2017. https://doi.org/10.1007/978-981-10-2471-9_20
[5] Rezwan, Sifat, Wasit Ahmed, Mahrin Alam Mahia, and Mohammad Rezaul Islam. "IoT based smart inventory management system for kitchen using weight sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi module with website and app." In *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, pp. 1-6. IEEE, 2018. https://doi.org/10.1109/ICACCAF.2018.8776761
[6] Gillis, Alexander S. "What is the internet of things (IoT)?." *TechTarget*, 2022.
[7] Omega. "What is a Load Cell Amplifier?." *Omega*.
[8] UNLEASHED. " What is an inventory management system?." *Unleashed Software*. Accessed July 6, 2021.
[9] Fritzing. "Project view." *Fritzing*. Accessed July 7, 2021.
[10] Components101. "NodeMCU ESP8266." *Pinterest*.
[11] Trent, Dara. " Load Cell Specifications & Datasheets." *Load Cell Central*.
[12] Addicore. "OLED Display - 128x64 1.3in Monochrome." *ADDICORE*.
[13] Arduino. "Arduino Environment." *Arduino*. Accessed July 9, 2021.
[14] Buntak, Krešimir, Matija Kovačić, and Maja Mutavdžija. "Internet of things and smart warehouses as the future of logistics." *Tehnički Glasnik* 13, no. 3 (2019): 248-253. https://doi.org/10.31803/tg-20190215200430
[15] Dennert, Kate, Logan Friedrich, and Rajeev Kumar. "Creating an Affordable, User-Friendly Electronic Inventory System for Lab Samples." *SLAS TECHNOLOGY: Translating Life Sciences Innovation* 26, no. 3 (2021): 300-310. https://doi.org/10.1177/2472630320973594
[16] Doss, Robin, Rolando Trujillo-Rasua, and Selwyn Piramuthu. "Secure attribute-based search in RFID-based inventory control systems." *Decision Support Systems* 132 (2020): 113270. https://doi.org/10.1016/j.dss.2020.113270
[17] Lee, Carman KM, Yaqiong Lv, K. K. H. Ng, William Ho, and King Lun Choy. "Design and application of Internet of things-based warehouse management system for smart logistics." *International Journal of Production Research* 56, no. 8 (2018): 2753-2768. https://doi.org/10.1080/00207543.2017.1394592
[18] Panigrahi, Rashmi Ranjan, Duryodhan Jena, and Arpita Jena. "Deployment of RFID technology in steel manufacturing industry-An inventory management prospective." In *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, pp. 705-719. Singapore: Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-5243-4_67
[19] Patnaik, Srikanta, Xin-She Yang, and Ishwar K. Sethi, eds. *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*. Springer Nature, 2020. https://doi.org/10.1007/978-981-15-5243-4
[20] Shabandeh, M. "Apparel Market Worldwide." *Statista*.
[21] Van Geest, Maarten, Bedir Tekinerdogan, and Cagatay Catal. "Design of a reference architecture for developing smart warehouses in industry 4.0." *Computers in Industry* 124 (2021): 103343. https://doi.org/10.1016/j.compind.2020.103343