Chapter

Design and Implementation Smart Petrol Station Using Internet of Things

Zahra'a M. Baqir and Hassan J. Motlak

Abstract

The problem that still exists nowadays with the petrol station is the method of operation because the petrol station is currently operated manually. This chapter adopted an automatic fuel filling algorithm based on a Radio Frequency Identification (RFID) card that carries information about the user regarding the type of user, the available balance and the permitted amount of fuel. This data is sent via the Wi-Fi module (ESP-32S) to the IoT platform (the panel), where it is encrypted with the hashing function and the Advanced Encryption Standard (AES) algorithm before being displayed by authorized access to the platform dashboard, which is It is also encrypted with an ID and a secret number.

Keywords: Internet of Things (IoT), RFID, ESP32 microcontroller, sensors, ThingsBoard platform, petrol station

1. Introduction

Regarding to this research work, the research work will be divided into three categories, In first part, petrol station hardware which consist of an NFC RFID reader for cashless payment, sensors are going to be used in measurement of the petrol transferred to the customer's tank, levels and the temperature of fuel in storage tank and sensors for fire protection and environmental monitoring. Calculations and programming are to be made through at Arduino mega 2560 and ESP-32S microcontrollers. For the second part, the ESP-32S module is used to enable wireless communication interface between the petrol station hardware system and the third part, which is the petrol station monitoring system, which is represented by the IoT ThingsBoard server that is downloaded to Raspberry Pi as a local server for data collection, processing, visualization, and system management. Arduino programming language is used in order to create a systematic software as the main control system for this petrol station system.

2. Literature survey

2.1 RFID-based smart petrol station

Implementation of a smart petrol station system in several countries can be read in many literatures all these projects used RFID tag as a fee card.

In 2016, Punit Gupta, et al. [1] described the design and implementation of a smart fuel station that will calculate the amount of fuel in the petrol station and send the information to a main server. They apply ultrasonic sensor to measure the level of petrol.

In 2016, Gowri Shankar, et al. [2], the proposed Biotelemetry system is used in this research to confirm the singular user with their Petrol tag. Biotelemetry is being employed as a thumbprint sensor in this case.

In 2016, Wavekar Asrar, et al. [3], had the project's major goal is to create a system that uses RFID technology to automatically deduct the amount of gasoline dispensed from a user's card. The use of RFID technology to control fuel dispensing in Indian cities is proposed in this study.

In 2017, Kumaresan and Babu Sundaresan [4] presented paper, it allocates with automation of a petrol station selling outlet; this system will provide the transactions and supply report to the holder for each hour. In this thesis, the user can access the present status of the petrol station as well as the stock maintenance through the net request.

In 2018, Chandana et al. [5], had the goal of this project is to build a security mechanism for filling gasoline at petrol stations without involving humans. The RFID smart card eliminates the risk of carrying cash at all times and also has a prepaid recharging option.

In 2018, Dongarsane et al. [6], had the secure and atomized delivery of gasoline is making possible by the simple and proper use of microcontroller and GSM technologies. It comes with an easy-to-use mobile phone system and a colorful user interface (GUI).

In 2019, Rashmitha [7] suggested a project where the customer has a smart card in this project. The card is nothing unless it has a magnetic member inserted in it. When a consumer presents this tag to the reader, the reader reads the unique number and sends the sign to the microcontroller.

In 2019, Nitha Velayudhan et al. [8], presented study proposes an automated petrol station management system that addresses the shortcomings of the current system. They used GSM and AT Mega 328 to develop an automatic gasoline filling system. This solution has the potential to make the fueling process more convenient, reliable, and secure.

Compared with the projects presented in the above sources, it is clear that our designed project has combined a number of mechanisms used in the construction of a smart fuel filling station. In our project, the researchers used the Arav De card to deal with secure electronic payment. The petroleum card has also been distinguished with a unique number that distinguishes between the type of users and thus controlling the distribution of fuel in Iraqi cities.

2.2 Networked petrol station systems for the IoT

In 2017, Sahana Rao et al. [9], used RFID and GSM technology to automate the filling of fuel at a petrol station. The transactions are designed to be user-friendly, which means that they are available at the touch of a button on the customer's smartphone.

In 2017, Carlo Makdisie and Badia Haidar [10], produced the stated system by assistance of present programs (Labview, Protues, PIC C Compiler) to reach a high precise control of the needed parameters, furthermore to using SCADA for managing and controlling all the system to escape any sudden faults; similar fire disaster and building the system work with extreme precision.

In 2018, Deepa et al. [11], used GSM and RFID, this paper created an automated fuel pump. All users in this system have an RFID card, which can be charged via a few different points. The fuel station is prepared with a smart card reader that detects the amount in the card as well as all security information and displays it on the LCD.

In 2019, Pranto and Rahman [12] presented paper indicates a design in which a system will be capable to trace a vehicle's petrol refill amount, cost & time then notifies the vehicle's owner about the refill as well. The system is divided into two parts: A mobile application and a petrol station.

In 2021, Sembodo and Atmajaya [13] presented monitoring tool uses the Arduino ESP8266 which is combined with Internet of Things (IoT) Technology which is displayed on the LCD and the Internet Web. Customers can purchase the patrol at gas stations independently, by connecting the smartphone application with the gas station wireless and purchase and make payments through the application.

In our designed project, three filling stations were connected, each one controlled by the Arduino Miga and ESP32S, and connected to them by sensors to represent the lower layer (sensing layer) of the IoT infrastructure. Data is transferred to the network layer using the Message Queuing Telemetry Transport)MQTT(protocol to represent the middle layer. The data from this layer is sent to the local server (Raspberry Pi) or a cloud server. The data is displayed in a web application from the IoT platform, ThingsBoard.

3. The proposed system

In this chapter, we proposed a design system of three smart petrol stations that are monitored and managed remotely from a single interface on IoT platform. a smart petrol station system that performs each of the following tasks: Detecting the fuel level inside the storage tank, measuring the fuel temperature, an automatic fuel dispenser, electronic payment, monitoring the environment through the temperature and

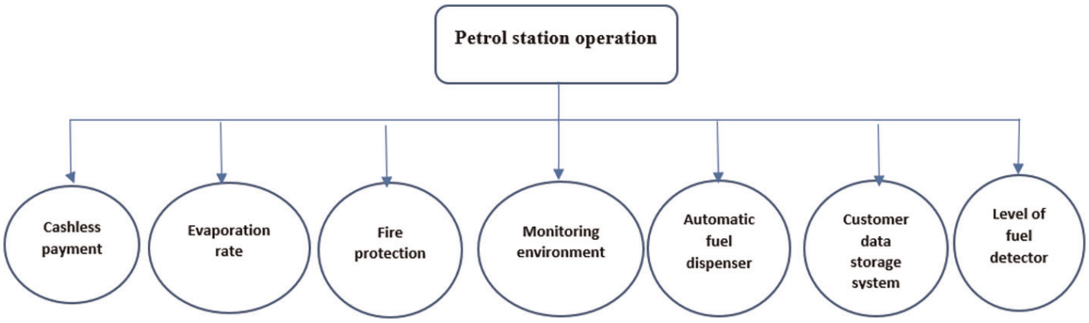


Figure 1.
Functions of the proposed smart petrol station as IoT application.

humidity sensor, detecting fires and extinguishing them automatically and storing all related data of customers and station with a database stored on a local server or applied to a cloud server. This data can be accessed, viewed and processed through the ThongsBoard web application, which can be accessed from any web browser on smart devices or desktop. **Figure 1**, shows the operations of the proposed system.

3.1 Architecture of the proposed system

Figure 2, shows the block diagram of designed system of three smart petrol stations based on IoT. The petrol station system connects to local server or cloud server over the internet. Online data can be visualized by a remote company office on ThingsBoard web user interface through chrome or Firefox web browser from desktop or smart phone or other devices. **Figure 3**, shows the flow chart of this system. The scheme offered consists of hardware part and software part.

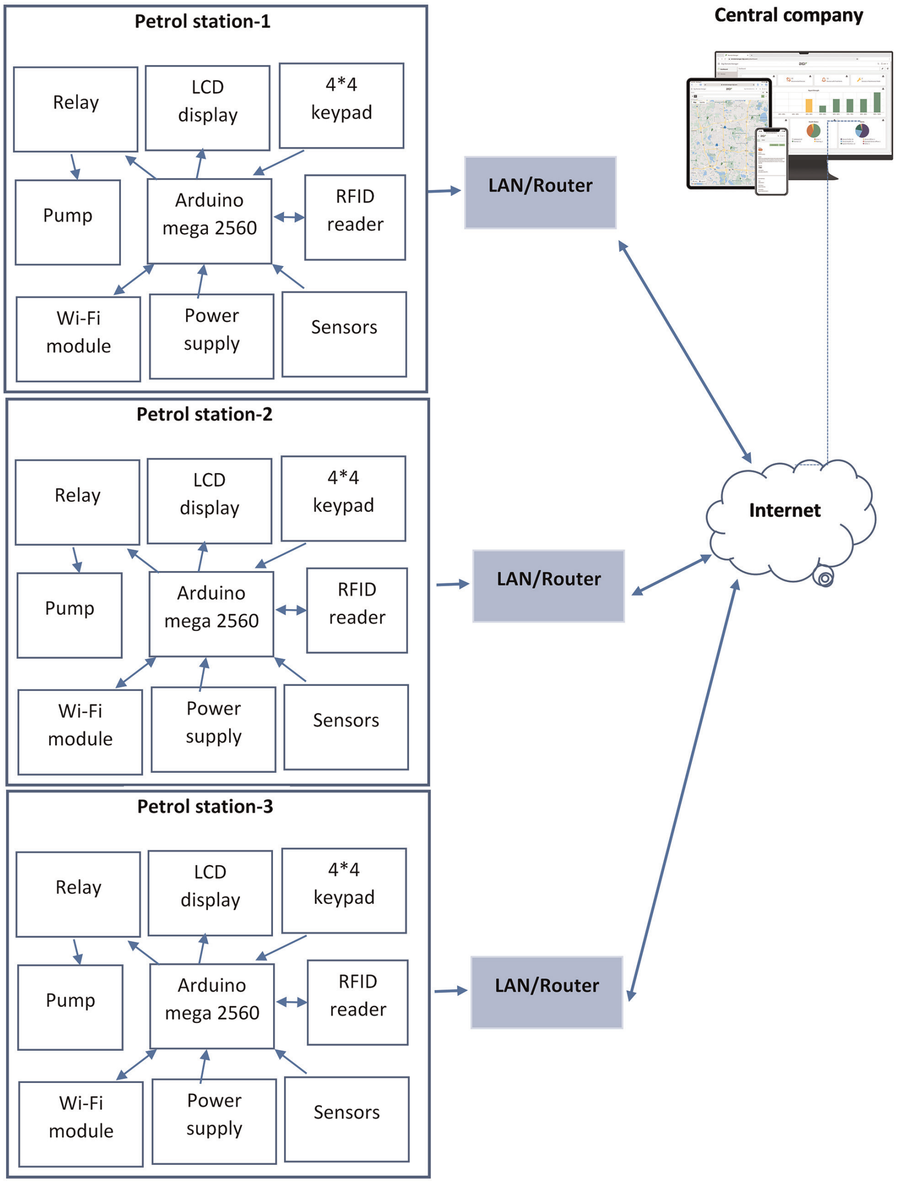


Figure 2.
Block diagram of the proposed system.

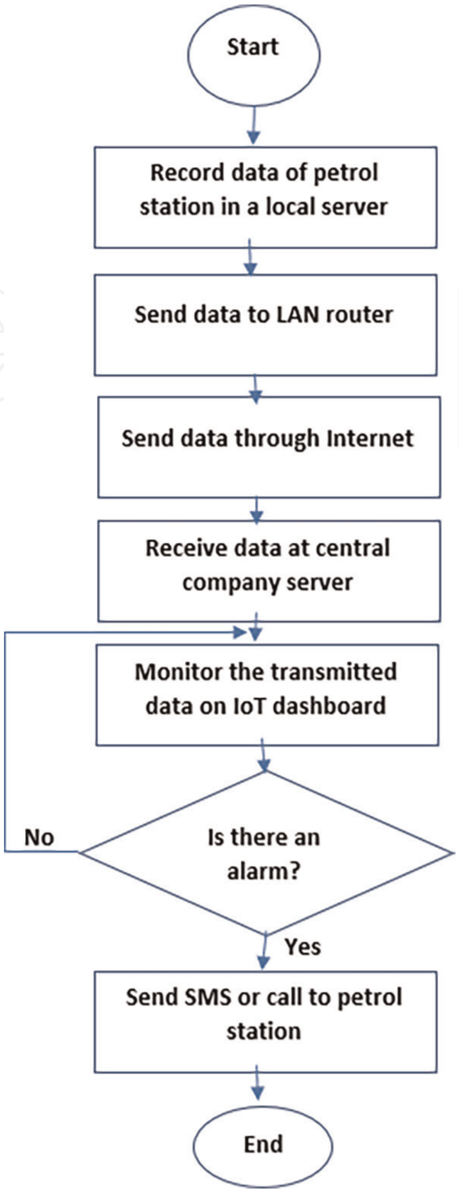

Start
Record data of petrol station in a local server
Send data to LAN router
Send data through Internet
Receive data at central company server
Monitor the transmitted data on IoT dashboard
Is there an alarm?
No
Yes
Send SMS or call to petrol station
End

Figure 3.
Flow chart of the proposed system.

3.1.1 Hardware part

It consists of four layers as IoT architecture. These layers are:

3.1.1.1 Sensing layer

This layer is represented by the control system that controls the fuel dispenser and storage tank.

3.1.1.2 Gateway layer

This layer is represented by the Wi-Fi router, which receives the sensor data from the station and sent by ESP-32S via the MQTT protocol to be directed to the server at the service layer.

3.1.1.3 Service layer

The service layer of the system designed from the server consists of a Raspberry Pi 4 model B minicomputer with a 1.5 GHz quad-core 64-bit processor that provides about 3 times better performance compared to the previous version, more ports and faster Internet connection, where the certified Raspbian operating system is downloaded [14]. On the Linux system, and downloading the ThingsBoard platform on this system for the purpose of reviewing all the variables and operations that the customer performs, and they occur at the packing station immediately after logging into the server by entering a fixed IP address in any web browser such as Chrome, Firefox or others. Or dealing with cloud ThingsBoard Server for the purpose of accessing station information from anywhere in the world via the Internet. For more security this server can be accessed using a username and password to view data in the application layer.

3.1.1.4 Application layer

It uses a user interface where the central company managing the three stations can view data from any station using any web browser from a desktop or smart phone.

3.1.2 Software Part

The petrol station is controlled by a system to enable that it operates efficiently. As this petrol station is expected to be left standalone and managed wirelessly, a system to control and monitor it from a distance is necessary. Software that was used in this application are:

3.1.2.1 Arduino Software (IDE)

The Arduino Integrated Development Environment (IDE) is a free and open-source cross-platform application (for Windows, macOS, and Linux) developed in C and C++ functions. The Arduino IDE makes it simple to develop code and upload it to the device [15]. This software can be used with any Arduino board. Arduino programming is used to control the petrol station system. The Arduino Mega is communicated with the ultrasonic sensor for the fuel leveling and with other sensors in the project for environmental monitoring and controlling. Aside from that, Arduino programming controls the 4 x 4 keypad number, RFID system, and LCD display that serves as the customer's interface. Flow meter and relay are responsible for determining the functionality of the pump, needed to be programmed by Arduino. Finally, Arduino IDE uses the ESP-32S module in programming to establish a connection with the internet, in order to provide these functionalities;

- Determines sales and store a database for the petrol station on local server.

- Monitors the temperature and humidity of weather.

- Provides information about the petrol level (stock) to the server.

- Provides information to the server about evaporation and density of fuel in the storage tank (stock).

3.1.2.2 Proteus program for simulation

"Virtual System Modelling" (VSM) by Proteus combines SPICE simulation in mixed modes with the industry's fastest microcontroller simulation. It allows for quick hardware and firmware prototyping in software [16]. Proteus VSM's most significant aspect is its ability to simulate the interface among software functioning on a microcontroller and any digital or analog devices linked to it. The microcontroller simulation is put on the schematic among the other parts of design of your product. It mimics object code execution in the same way as an actual chip does. Logic levels in the circuit will change whenever your program code writes to a port, and if the circuit varies the situation of the pins of processor, your program code will notice, just like in physical life. The VSM CPU models accurately emulate each supported processor's input/output ports, timers, interrupts, USARTs, and other components. Because the interface of each of these peripherals with the outside circuit is thoroughly described to the level of waveforms, the whole system is being tested, it is far more than a simple software simulator. With over 750 microprocessor variations supported, a lot of fixed SPICE models, as well as one of the major libraries of entrenched simulation peripherals in the world, for an embedded model, Proteus VSM is still the best option.

3.1.2.3 LabVIEW program for simulation

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbenches (where a national instruments platform and development environment for a visual programming language. The goal of this type of programming is to automate the use of laboratory decision-making and measurement equipment. It's a proprietary software package that makes it easier to design different types of data management, control, and measurement systems and is therefore ideal for GUI design.

3.2 Implementation of the proposed system

The designed system is divided into two main parts: the first part relates to the hardware on the board (control system) and the second part relates to the ThingsBoard web server. In **Figure 4**, the arrows pointing toward the Arduino mega 2560 shows the input, while the arrows pointing from Arduino shows the output of the processor. For a single petrol station two parts are implemented as shown in **Figure 4**.

First: the ultrasonic sensor HC—SR04 connected at the top of a fuel tank to calculate the fuel level by sending ultrasound waves. The distance is equal to the speed multiplied by the time, the speed of sound is known at a temperature of 20 degrees Celsius equivalent to 343 meters/second. The time that the sensor processor calculated from the start time of the pulse to the echo return time is divided by 2. By knowing the time and speed, it is possible to calculate the distance, determine the height level, send the data to the microcontroller (ATmega2560) connected to the ESP32 Wi-Fi module and send the data later to the server. A fluid temperature sensor (DS18B20) has been added to the tank, and it can be added to the pump hose to measure the temperature of fuel and the consequent result of calculating is the evaporation rate and density which dependent on temperature to determine types of mixtures. DS18B20 connects via 1-WIRE protocol requiring one data line to connect to the Arduino Mega Board.

Second: the fuel dispenser system is a box where the electronic parts appearing on it, which represented by a 4 * 4 keyboard, have been linked. The letters on the panel

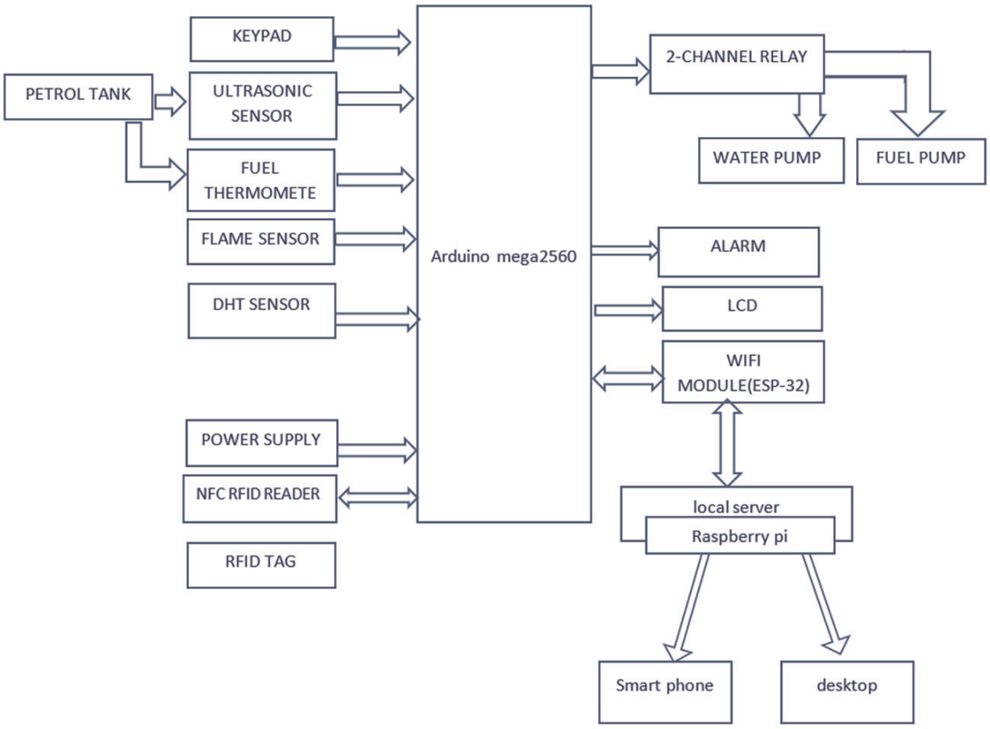


Figure 4.
The proposed smart petrol station.

perform a number of functions. Also, a 20 * 4 LCD screen was attached to the box to show the customer all the functions when pressing the keypad. NFC RFID to scan the RFID tag by establish a wireless connection with each other by touching or bringing them close at close range, usually not more than a few centimeters. NFC RFID and LCD communicate with each other and send data between them via the (I2C) protocol, which is a serial data communication protocol, which operate among two devices or a group of devices to transfer information among them. The protocol works in a "half-duplex" manner, one of the devices will be sending whereas the another will be at the receiving state and they cannot transmit at the same time. The DHT11 temperature and humidity sensor has been added to the contents box to monitor the temperature and humidity to be an example of monitoring the environment and showing the results directly on the server from the second petrol station. The fuel and water pumps are connected on the sides of the box; the fuel pump is connected to the fuel tank on the one hand and linked to a flow meter that calculates the number of liters required; the flow meter is connected to a hose on top of which hose there is a button that can be pressed to fill the required amount of fuel; and if the required quantity is filled, the pump will revert to its original position. As for the water pump, it has been linked to a flame sensor that sounds an alarm when a flame is detected, as it sends its data to the microcontroller that instructs the relay switch associated with the water pump to operate it. This pump can also be controlled manually via a push button and the pump is started. **Figure 5**, shows the connection map establishes by fritzing software for the hardware parts on the board. **Figure 6**, shows a section of the practical hardware of the designed and implemented system, which is installed outside the control system box.

Inside the control system box, there is an Arduino Mega 2560, a microcontroller board based on the ATmega2560 processor. The board contains everything needed to support the microcontroller; It can be connected to a computer with a USB cable or powered by an AC-to-DC adapter or battery to start. The ATmega2560 contains 256 KB

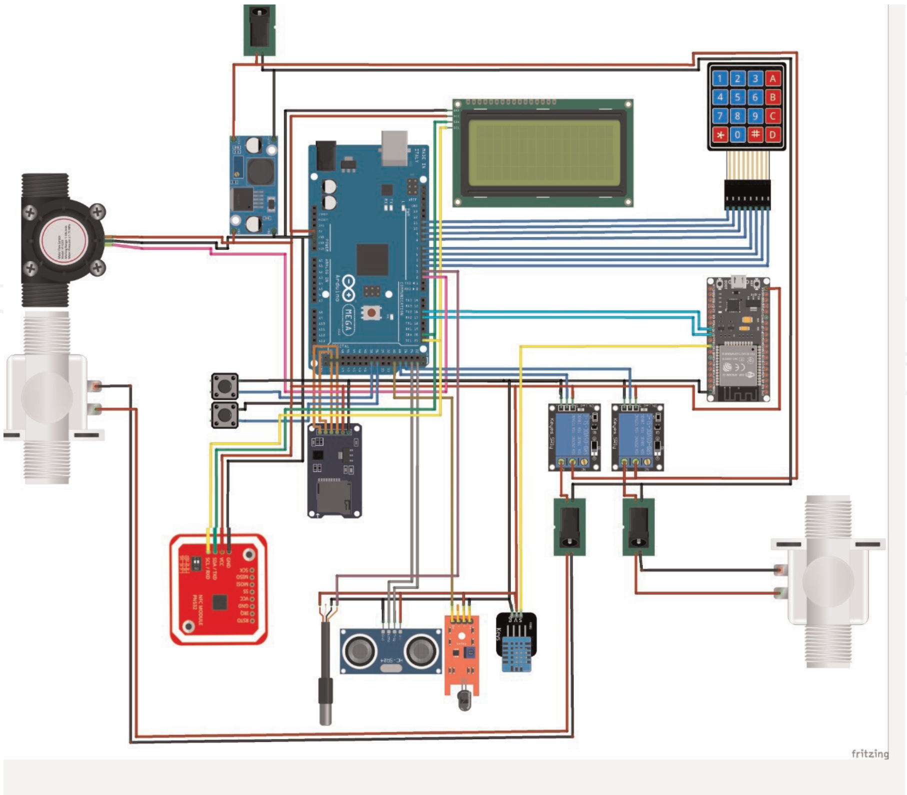


Figure 5.
Connection map on the board.

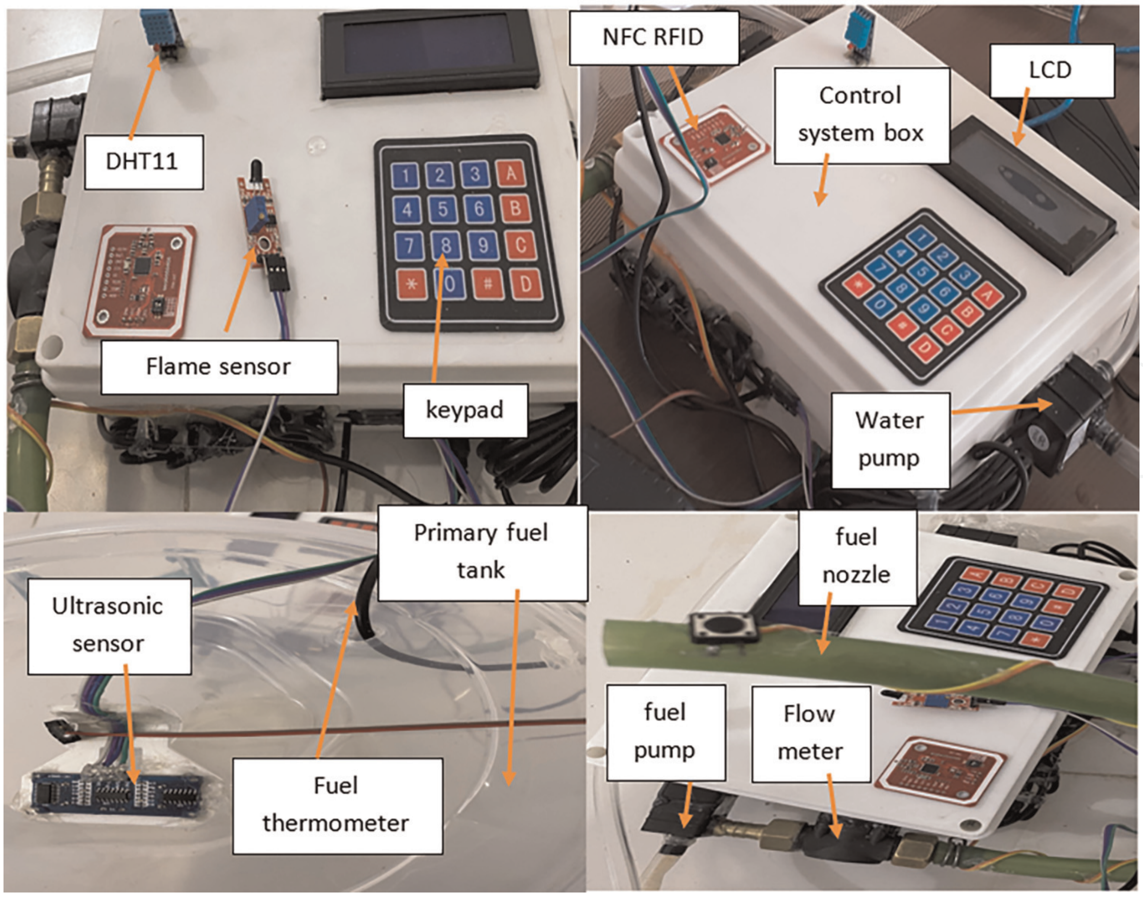


Figure 6.
Hardware section outside the box of the control system.

of flash memory to store the code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM. And because of this memory is little, the MicroSD card adapter module, which is a Micro SD card reader for reading and writing through the file system, and the SPI driver, has been linked, so this module connects with the microcontroller through the SPI serial communication protocol. This protocol works through three lines "MISO, MOSI & SCK". At any given time, only two devices can talk to each other. What distinguishes this protocol is that it is (Full-duplex) that is, the two devices can transmit and receive at the same time. Also, from the electronics that were provided for this system inside the box, the voltage regulator is fixed, as it has a voltage of 12 volts to feed the pumps and a voltage of 5 V comes out of it to feed the rest of the system elements. The two channel relay module MR009–004.1 is fixed inside the box, where is a module designed with two relays that provide 12 volt power to allow control of the fuel pump and the water pump. WiFi-BT-BLE MCU module the core of this module is an ESP32 chip that supports Secure Over-The-Air (OTA) upgrade. This module inside the box is responsible for providing a direct connection to the server via a WiFi router and it acts as a second microcontroller represent as second petrol station. **Figure 7**, shows the core components of the control system box.

3.2.1 Methodology of the proposed system in proteus simulator

In Proteus software, the methodology of the system designed for a single filling station has been simulated locally to simulate the following functions:

3.2.1.1 Fuel Filling

The NFC RFID detects the ID number of RFID card and sends it to the Arduino Mega 2560 microprocessor when the RFID card or tag comes within range of the RFID reader. The ID codes of the RFID cards are already recorded in the microcontroller. The microcontroller checks whether or not this RFID card is valid. If the RFID card is

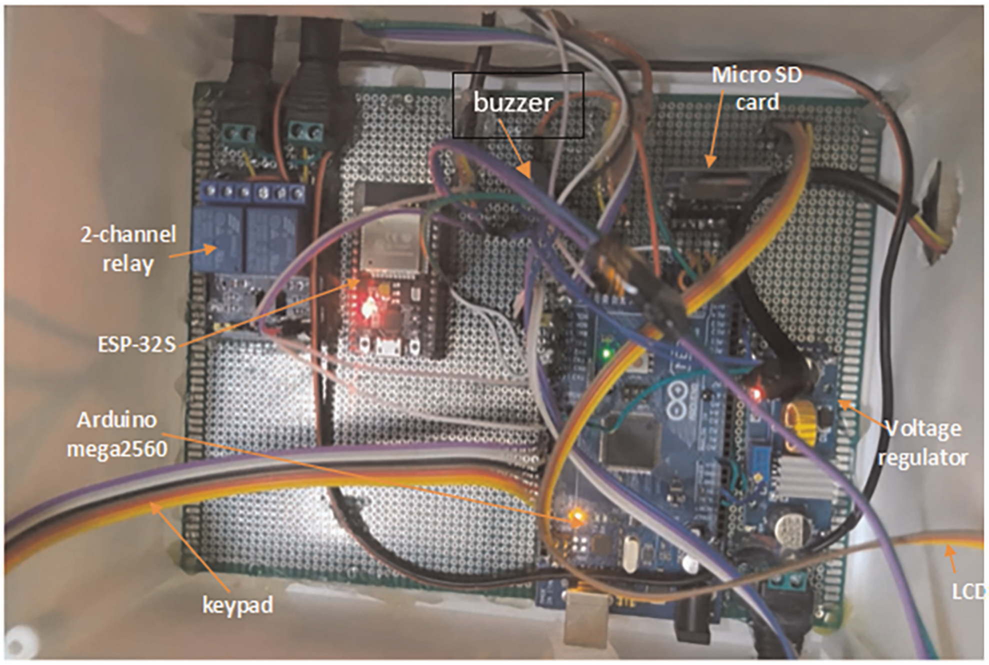


Figure 7.
Inside the control system box.

authorized, the microcontroller displays the RFID card user on the LCD screen after a check; otherwise, nothing appears, indicating that the RFID card is unauthorized. If the RFID card is authorized, the microcontroller allows the consumer to enter the amount of gasoline via the keypad, which is then displayed on the screen. The microcontroller then checks to see if the RFID card's available balance exceeds the amount entered. If there are more than that, the microcontroller activates the fuel pump motor through a relay switch for a set amount of time before turning it off. If the user wants to examine the quantity of petrol dispensed and the remaining balance of the RFID card after the procedure is completed, the information on the amount of petrol dispensed and the remaining balance of the RFID card is displayed on the LCD screen. Algorithm of automatic fuel filling pump shown in **Figure 8**. All this information about users is visualized to the IoT ThingsBoard server which installed on raspberry pi as local server, using Wi-Fi module ESP-32S. On a local server, we keep track of the amount of gasoline distributed by users, which is protected by a password that is only known by the fuel company. If the customer's available balance is less than the amount entered, the RFID card can be recharged. In this case, the RFID card functions as a prepaid card that may be recharged at the fuel station.

3.2.2 Methodology of the proposed system in LabVIEW

In LabVIEW software, the methodology of the system designed for a three-filling station has been simulated locally to simulate the following functions: level of fuel in

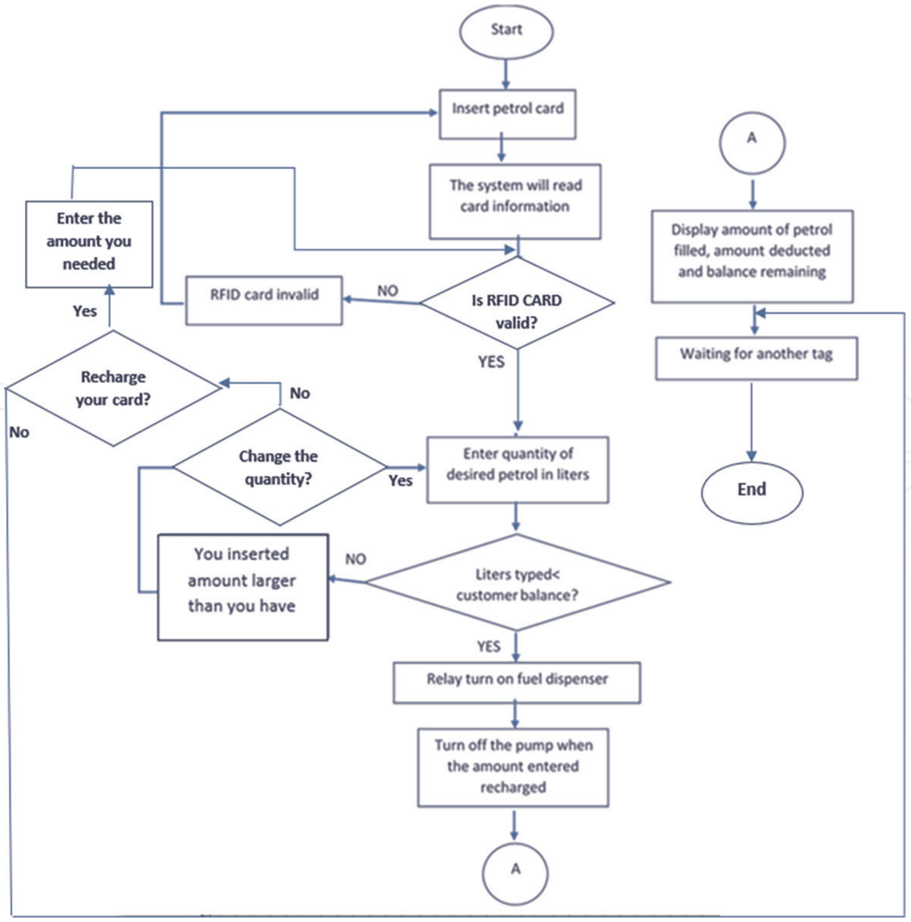

Start
Insert petrol card
The system will read card information
Is RFID CARD valid?
NO
RFID card invalid
YES
Enter the amount you needed
Yes
Recharge your card?
No
No
Change the quantity?
Yes
Enter quantity of desired petrol in liters
Liters typed< customer balance?
NO
You inserted amount larger than you have
YES
Relay turn on fuel dispenser
Turn off the pump when the amount entered recharged
A
A
Display amount of petrol filled, amount deducted and balance remaining
Waiting for another tag
End

Figure 8.
Algorithm of the automatic fuel filling pump.

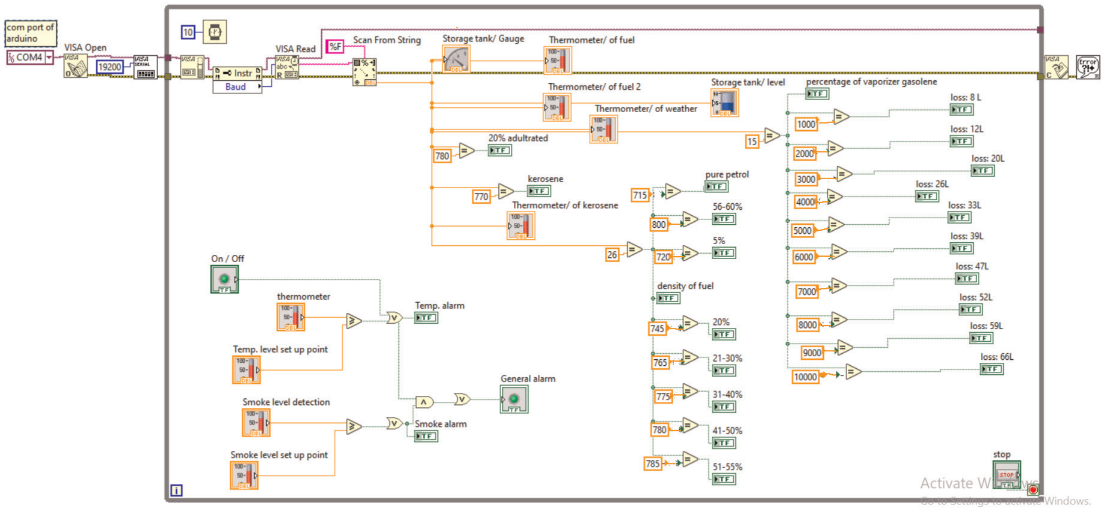


Figure 9.
Smart petrol station via LabVIEW.

the storage tank, the temperature of the fuel temperature and humidity of weather and fire alarm. As shown in **Figure 9**.

4. Simulation results

Simulation results of the system designed and implemented by Proteus and LabVIEW programs were achieved.

4.1 The results in proteus

The design suite Proteus 8.9 is an exclusive software implement suite that was principally used in this project for electronic project automation. In Proteus, a hex file to the microcontroller portion on the diagram was used to operate the Arduino mega2560 simulation. It is next co-simulated along through whichever digital and analog electronics linked to it. In **Figure 10**, we build a prototype by selecting a few blocks and using the Simulink block toolbar to join them together. Results were achieved on how to perform each of the following:

4.1.1 Monitoring and controlling storage tank

The spread of USTs represents an actual risk that requires difficult and costly methods for its management. But by adopting the method used in the simulation, it is possible to control and monitor the level of petrol available in the storage tanks in an easy and inexpensive way. **Figure 11**, shows the design of the power supply is simple and cheap, when it is turn ON. First, Arduino mega2560 checks the petrol level by ultrasonic sensor at the top of the tank. if a petrol level is low than threshold value, a buzzer is initiated and displayed low petrol level on the LCD and Arduino mega operates relay which connected to run DC motor of a fuel supply pump to supply fuel to the tank. In **Figure 12**, shows when a fuel level rises above the

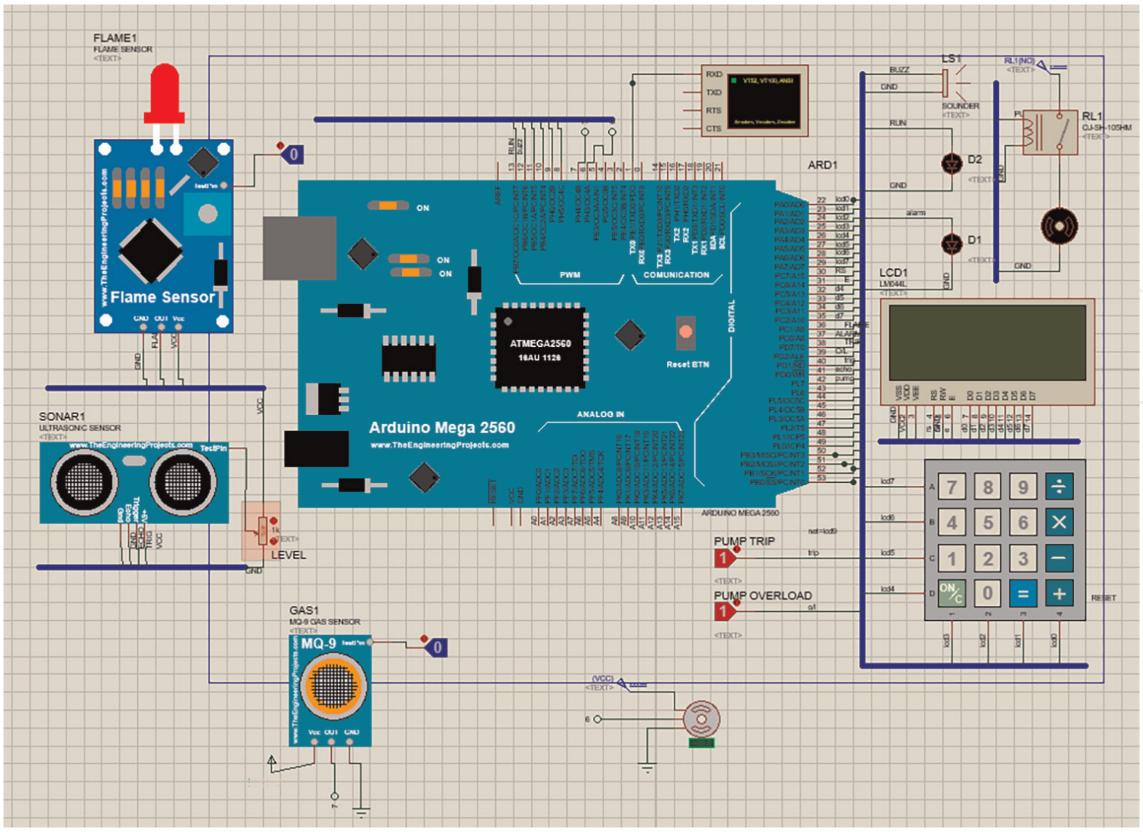
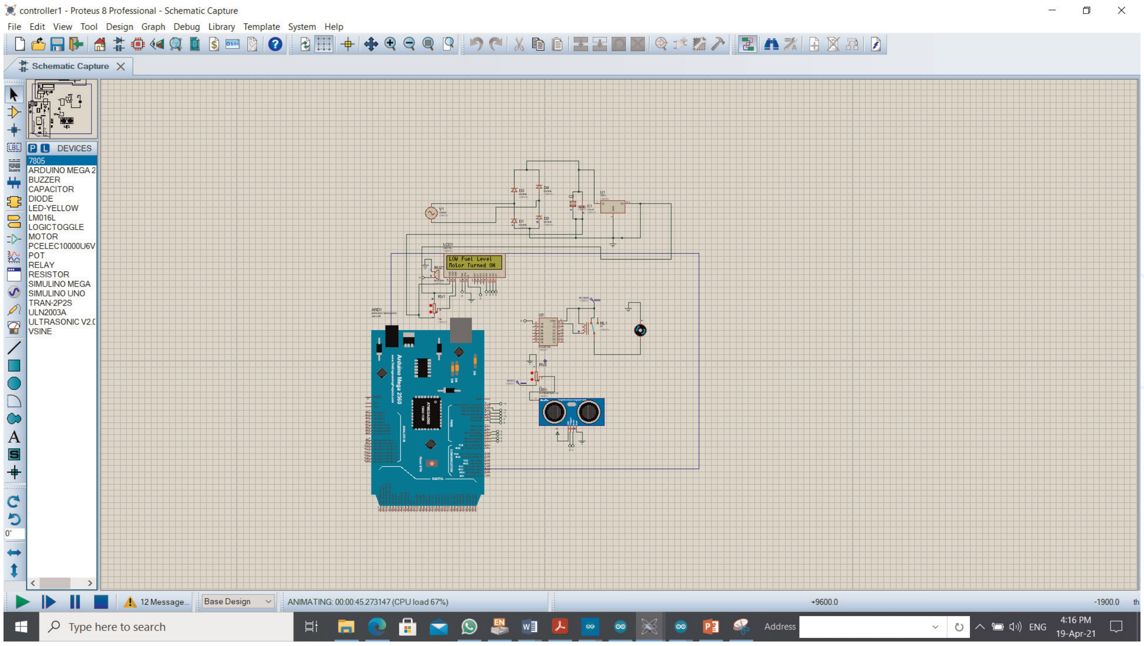


Figure 10.
Initialized subsystem before launched to IoT server in Proteus.



LOW Fuel Level
Motor Turned ON

ANIMATING: 00:00:45.273147 (CPU load 67%)

+9600.0

-1900.0

Figure 11.
Simulation of the management storage tank (low fuel level).

threshold value, the buzzer is turned off and levels of fuel displayed on LCD screen. When the fuel level reaches the full level as set in the programming, Arduino mega turns off the relay which connected to stop the dc motor of fuel supply pump running,

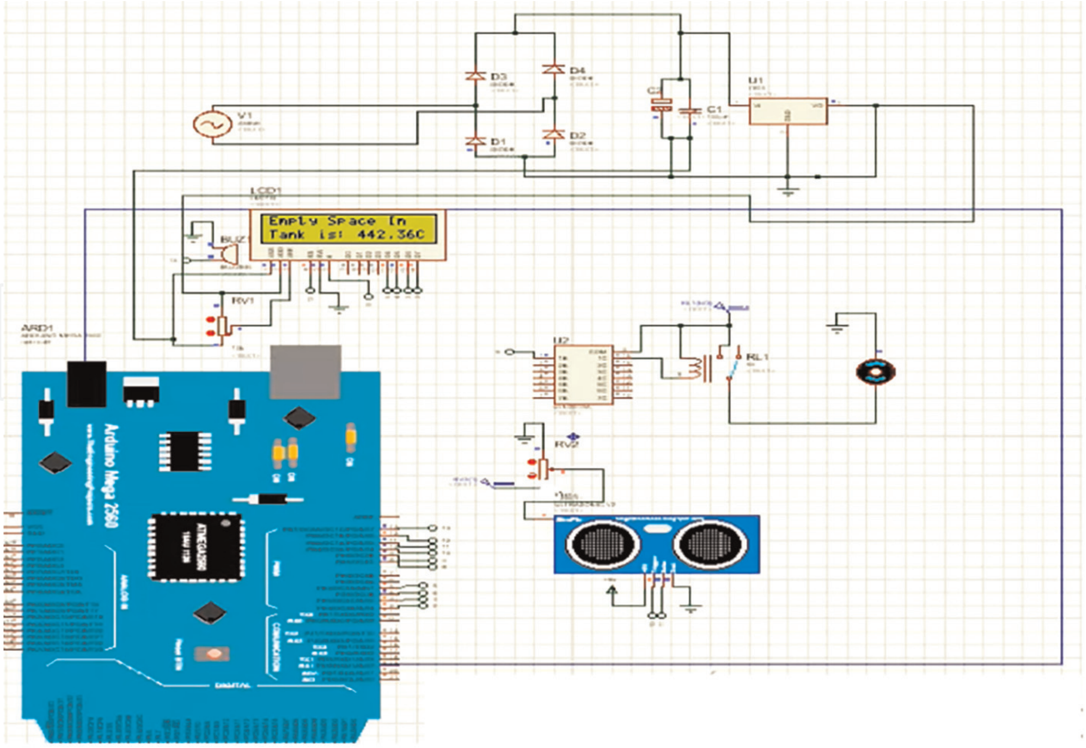


Figure 12.
Simulation of the management storage tank for the proposed system (display fuel level).

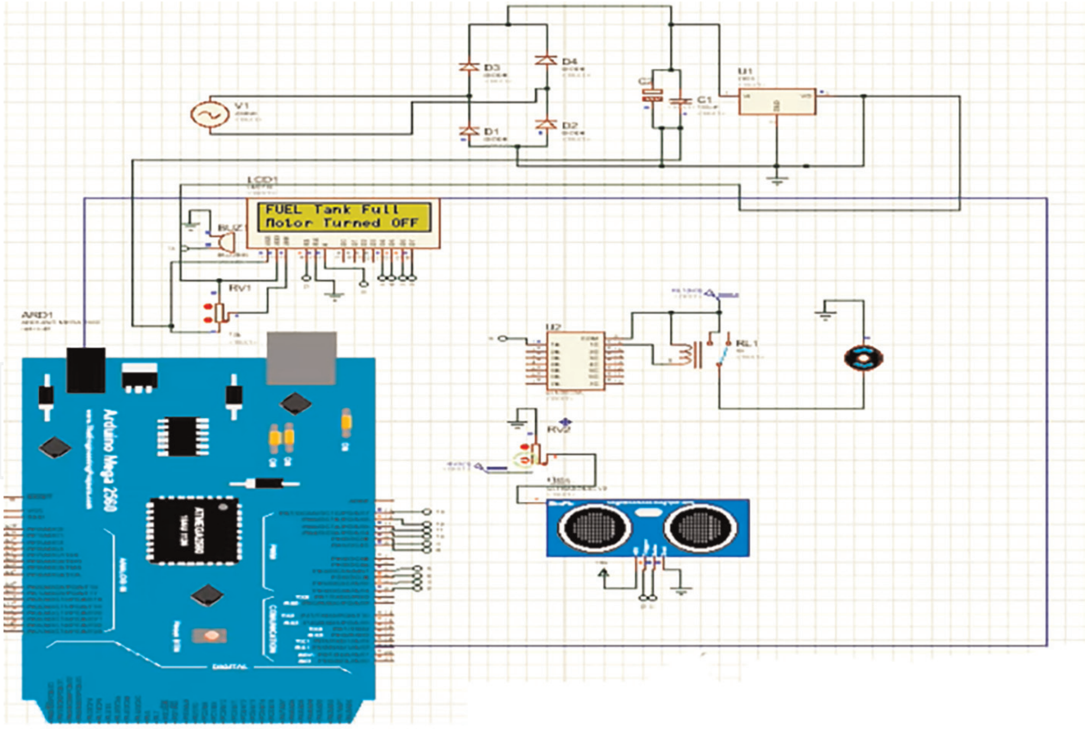
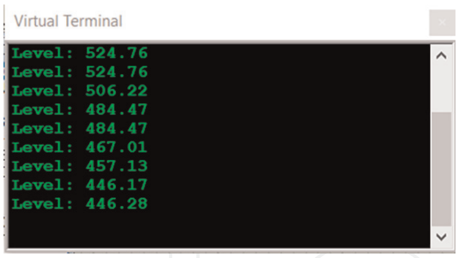


Figure 13.
Simulation of the management storage tank for the proposed system (full fuel tank).

the LCD screen displays that the tank is completely full as shown in **Figure 13**. **Figure 14**, shows how the fuel levels in the storage tanks can be displayed on the serial monitor.



```
Level: 524.76
Level: 524.76
Level: 506.22
Level: 484.47
Level: 484.47
Level: 467.01
Level: 457.13
Level: 446.17
Level: 446.28
```

Figure 14.
Fuel level of the storage tank on a serial monitor.

4.1.2 A fuel filling operation

Sub-system simulation of the proposed system in the Proteus program, to reveal how the system performs in the fuel filling process. The result of the subsystem, along with a step-by-step approach, is listed below:

Step 1: Swipe the RFID card and type in the ID number. Where it appears from the **Figure 15**, that Arduino mega2560 connected to the other parts of the subsystem, the green LED is activated to indicate that the petrol station is operating normally, which are of interest to us in this step, the connection between the microcontroller and Ultrasonic sensor installed at the top of the fuel tank, which measures the height of the fuel and send its data to Arduino mega to display the fuel level on the LCD screen, as for the benefit of connecting the microcontroller to the keypad, it appears by entering the ID number of the card swiped on the virtual card reader as shown in **Figure 16**.

Step 2: If RFID card is authorized, then the controller fetches the user available amount data from server and display on LCD, then microcontroller ask the user to enter the amount of petrol as shown in **Figure 17**.

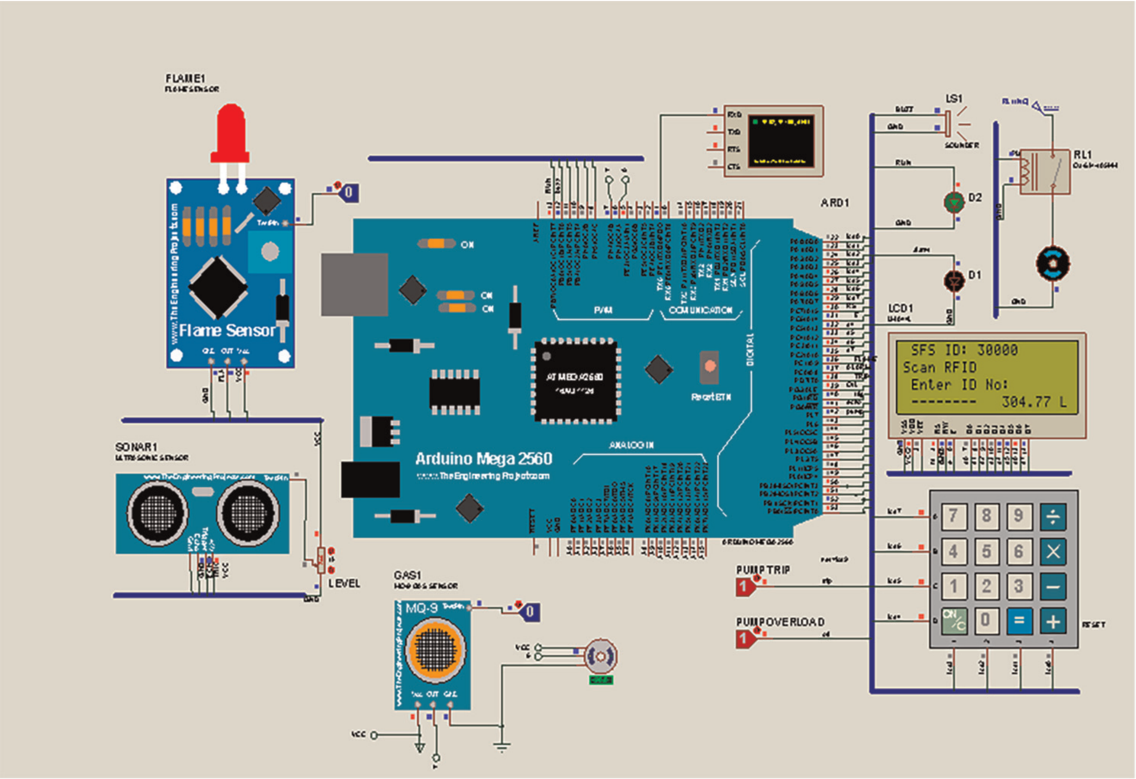
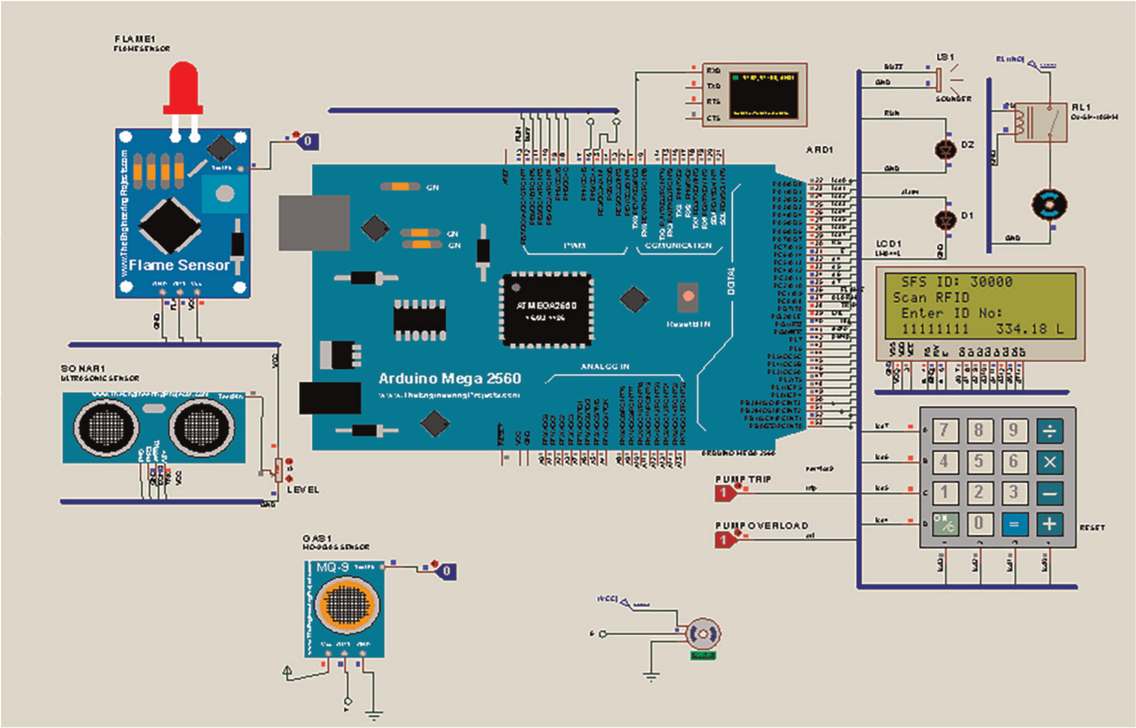
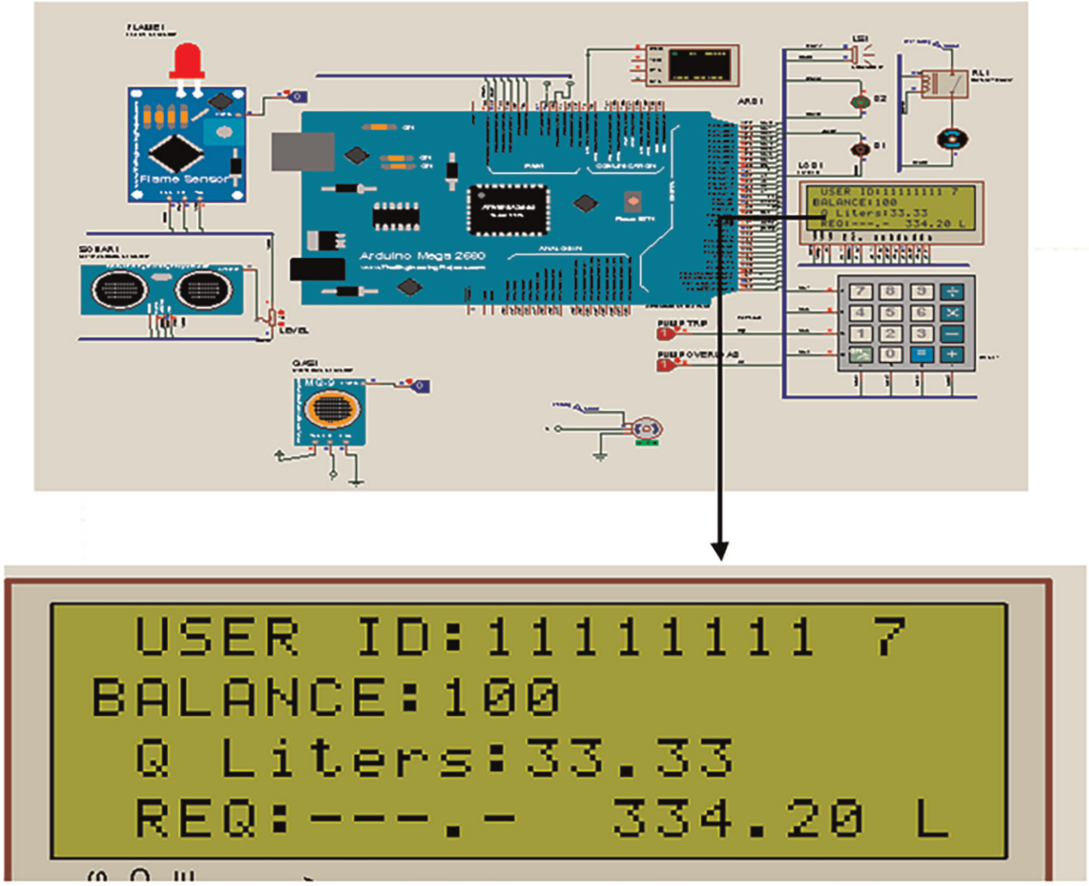


Figure 15.
LCD shows level of fuel tank.



```
SFS ID: 30000
Scan RFID
Enter ID No:
11111111    334.18 L
```

Figure 16.
Enter ID No by keypad.



```
 USER ID:11111111 7
BALANCE:100
 Q Liters:33.33
 REQ:---.-   334.20 L
```

Figure 17.
User ID and balance.

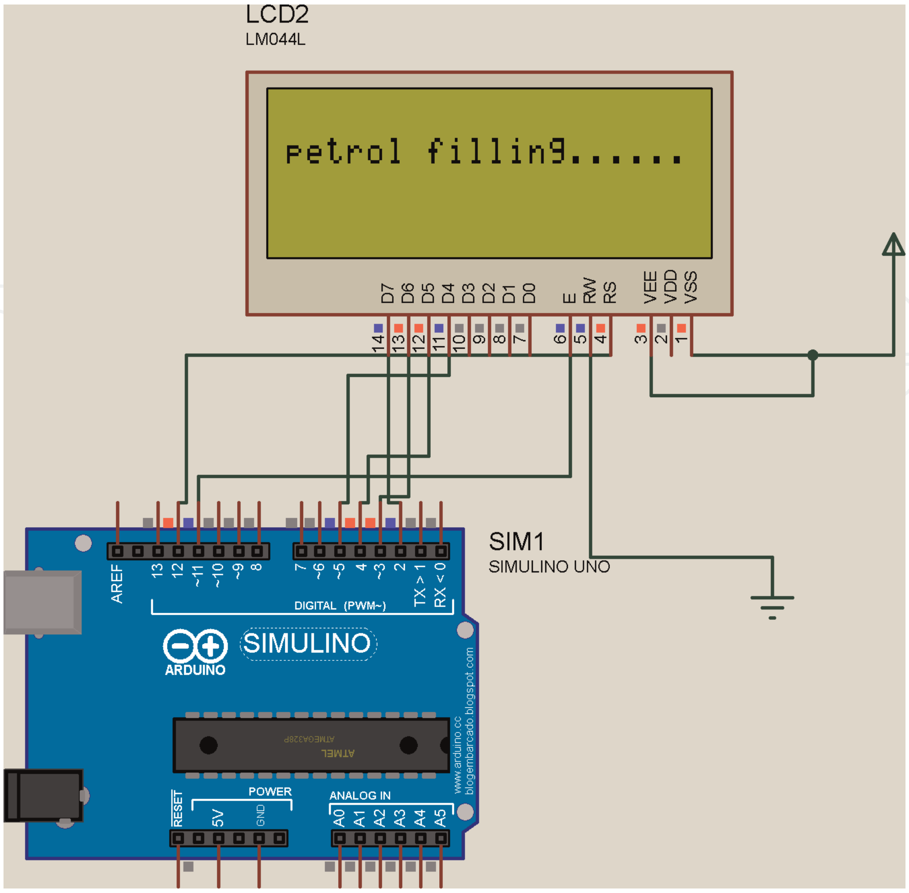


Figure 18.
The petrol filling process has started.

Step 3: If an entered amount is less than the available amount, then petrol filling process starts (i.e., Motor turns ON) as shown in **Figure 18**. Otherwise microcontroller say to recharge a suitable amount as shown in **Figure 19**.

Step 4: After filling the petrol, the petrol filled amount and available balance information is display on LCD screen as shown in **Figure 20**. And available balance is updated in the server.

4.1.3 Monitor fire protection

The microcontroller in this system is connected to a flame sensor and servo motor as a fire extinguisher (that discharges a jet of water, foam, gas, or other material to extinguish a fire). When there is a fire, the flame sensors detect it, then the warning system - indicated by the buzzer and red LED - is activated, and the servo motor is activated to convert its angle of 180 degrees. When the alarm state ends, the servo motor returns to its original position, and the LED and buzzer turn off. As seen in **Figure 21**, a fire alert shows on the LCD screen.

4.1.4 Monitor security of petrol station

Safety of fuel pump also tested in the occurrence of a buildup in the load, to indicate a warning situation on the LCD screen, the buzzer and the red LED were

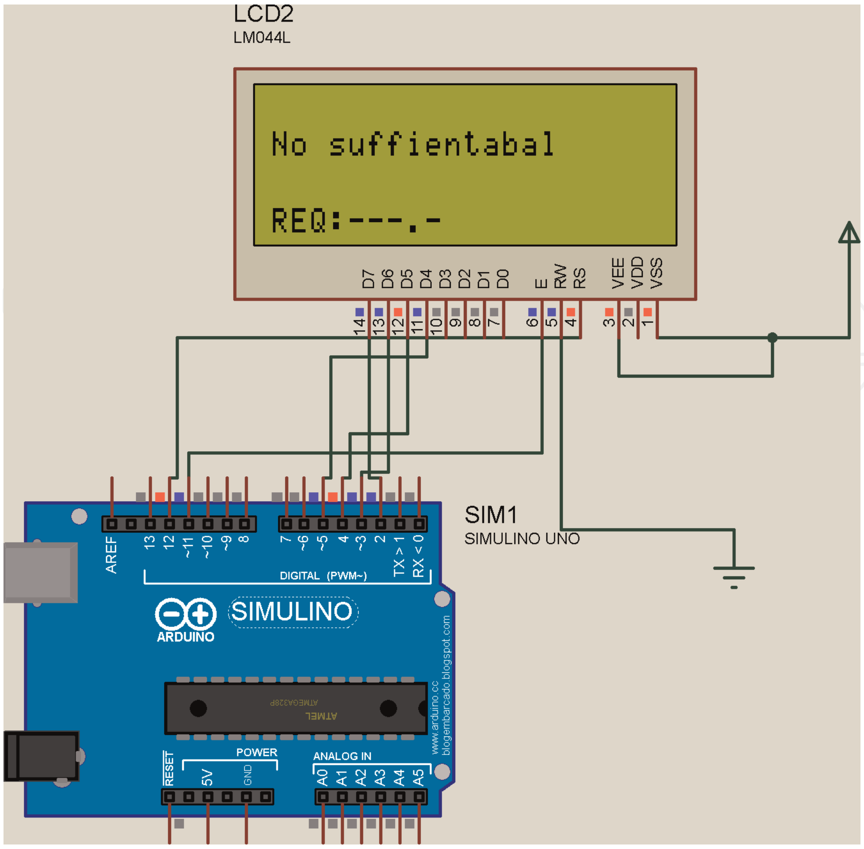


Figure 19.
If the entered amount is more than the available amount.

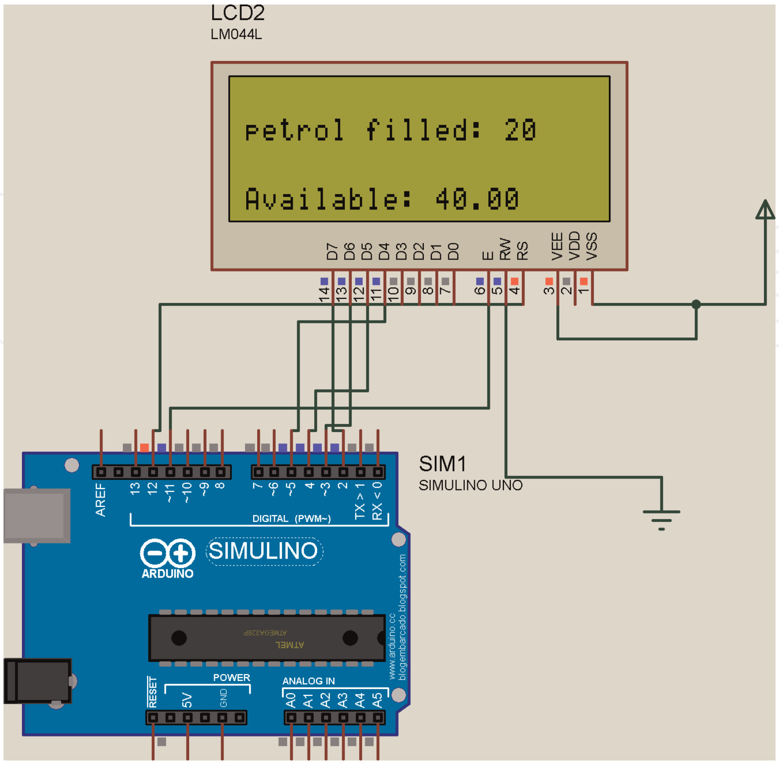


Figure 20.
The amount of petrol loaded and the available balance.

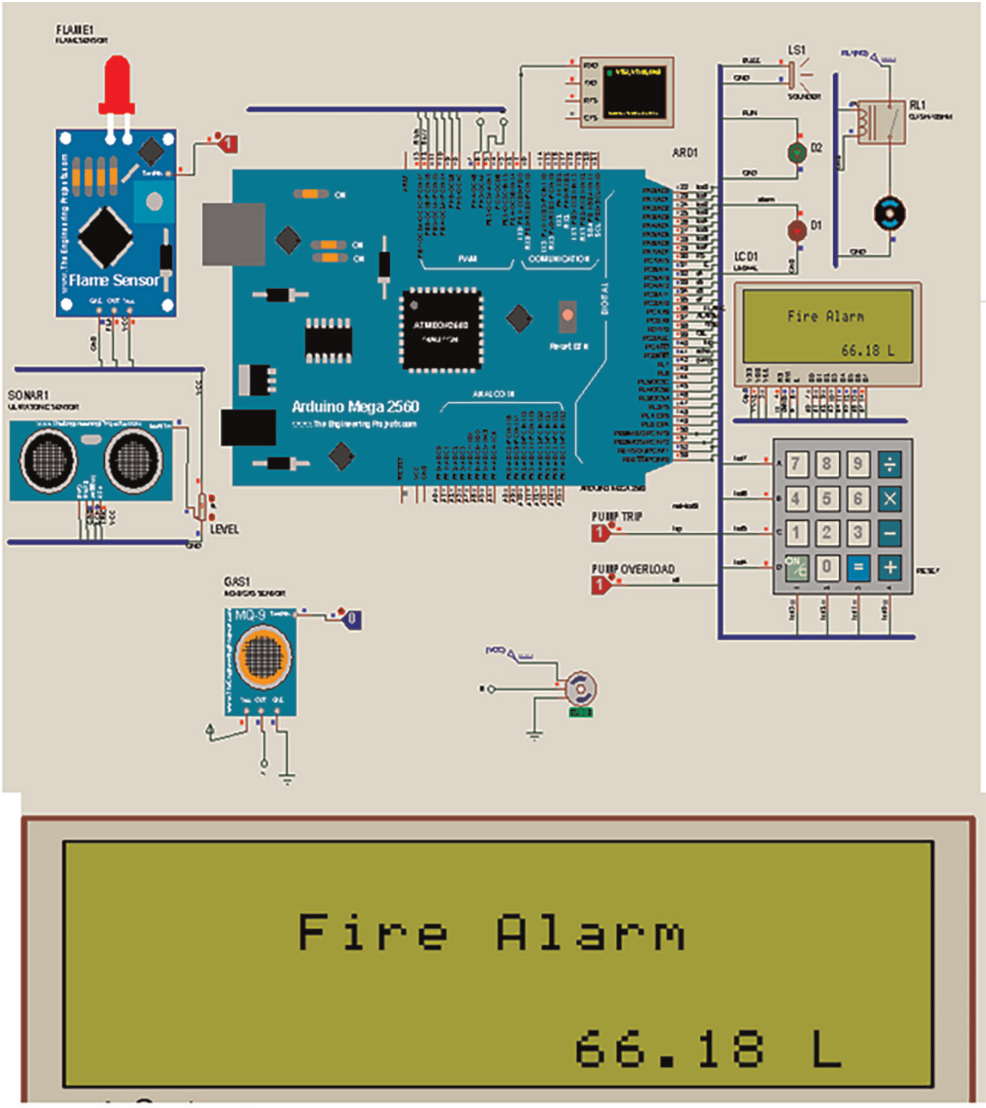


Figure 21.
Simulation of management fire protection of proposed system.

triggered to tell the viewer of the status of the load besides work to switch off the relay that runs the pump motor, as displayed in **Figure 22**.

4.2 The results in LabVIEW

The designed strategy is implemented in the LabVIEW VI environment, and simulation results are presented to validate the proposed controller in **Figure 23**. Where it appears from Figure monitoring the central station of three stations, the design assumes the following in each station:

1. Monitoring the fuel level inside the storage tanks for three products. Where a visual alert appears represented by a green LED indicating an acceptable level of fuel rise inside the storage tank, while if the fuel level drops below 20%, a visual alert represented by a red LED appears indicating the necessity of refilling the fuel to the indicated tank.

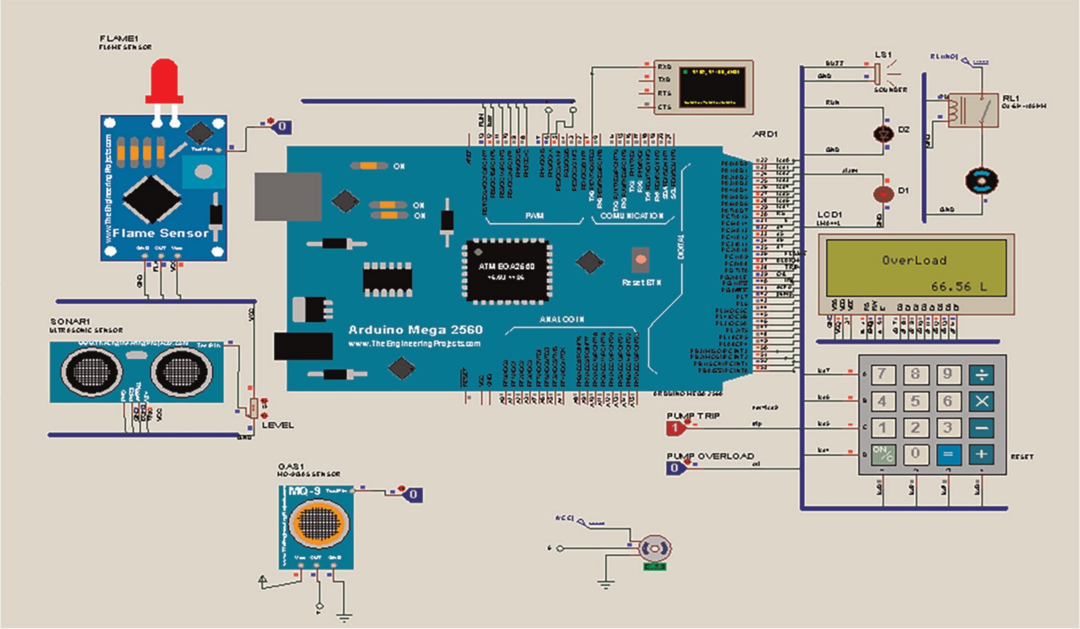
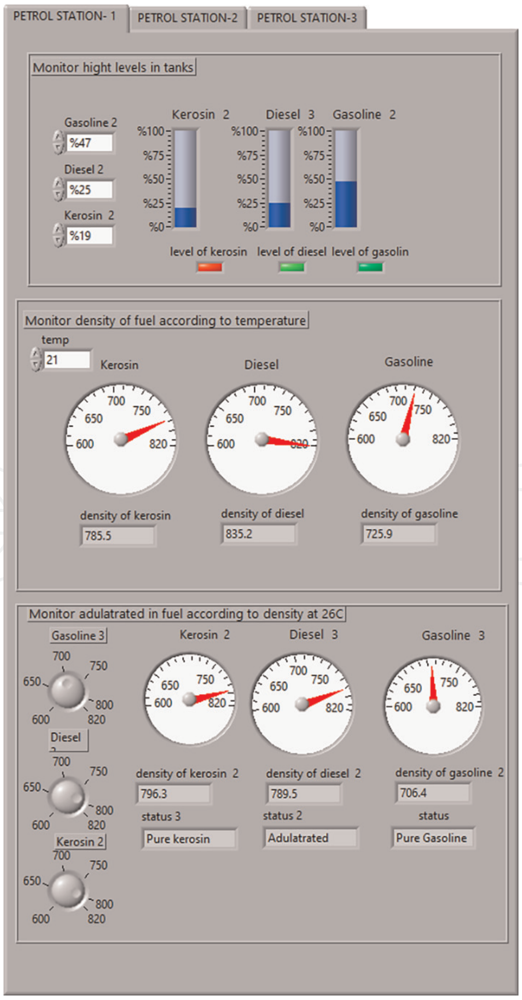


Figure 22.
Overload status.



Monitor hight levels in tanks

Gasoline 2: %47

Diesel 2: %25

Kerosin 2: %19

level of kerosin 🟥

level of diesel 🟩

level of gasolin 🟩

Monitor density of fuel according to temperature

temp: 21

density of kerosin: 785.5

density of diesel: 835.2

density of gasoline: 725.9

Monitor adulatrated in fuel according to density at 26C

density of kerosin 2: 796.3

density of diesel 2: 789.5

density of gasoline 2: 706.4

status 3: Pure kerosin

status 2: Adulatrated

status: Pure Gasoline

Figure 23.
The output parameters from LabVIEW.

2. Monitoring the density of the fuel according to the temperature. With a temperature of 21 Co, the density of gasoline was recorded at 725.9, while the density of diesel was 835.2, and the density of kerosene was recorded at 785.5

3. Quality control of the product according to the density: with a temperature of 26 Co, the density of gasoline was 706.9, and at this density the textual reference to the purity of the petroleum is made. At a density of 785.5 for diesel, the presence of impurities mixed with diesel was indicated, while kerosene at a density of 796.3 indicated the purity of kerosene.

In **Figure 24**, The fire alarm system consists of a thermometer for smoke and a thermometer for heat. The logic gate AND was used to verify the two conditions of high temperature and smoke to a level higher than the threshold limit. A fire alarm is triggered, but if one of the two conditions is met, the alarm is triggered by a red LED.

It is possible to place more than one temperature sensor in different locations of the same station and to show the data represented by a waveform. When an on signal arrives, the alert is done by text and using a red LED. But when there is no fire, there is no signal and the text is no fire. It is possible to show these functions in LabVIEW in **Figure 25**.

4.3 Practical implementation results

In the practical implementation of the project, two main outcomes can be observed. The first output is the hardware of an automated fuel pump station, which interacts with the second outcome, the petrol station monitor system, which is

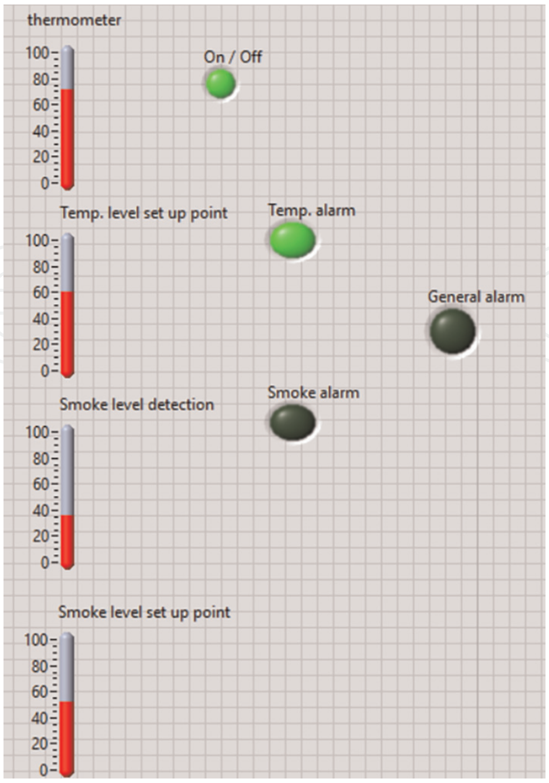


Figure 24.
Alarm fire protection.

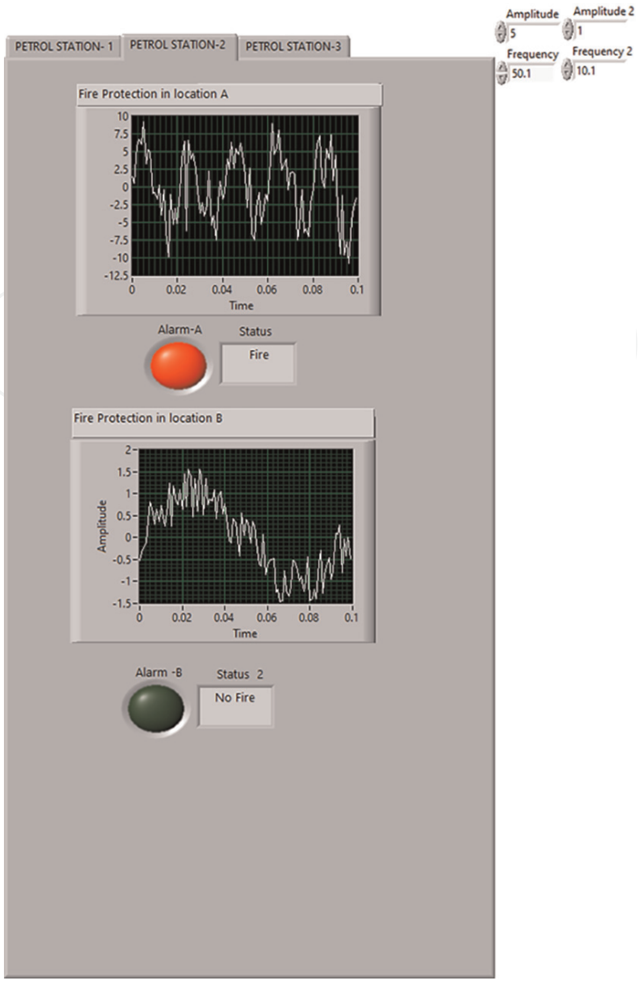


Figure 25.
Alarm fire protection from different locations.

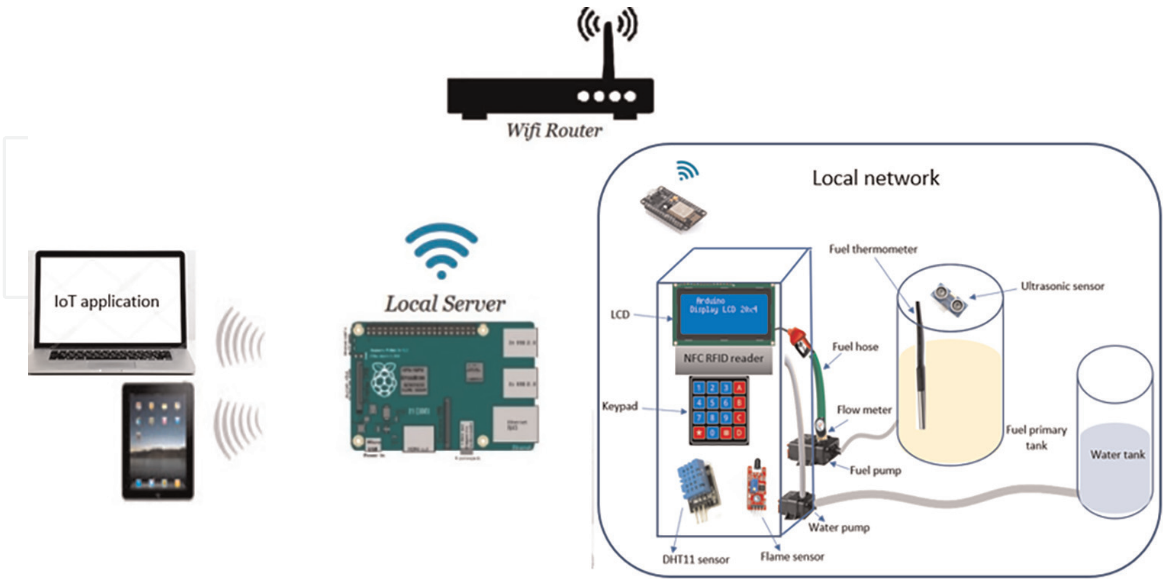


Figure 26.
The proposed model of the hardware system.

represented by the Internet of Things platform (ThingsBoard) installed on a Raspberry Pi as a local server. In this section, the functioning and interface of both are elaborated. **Figure 26** shows the implemented model of the hardware system.

4.3.1 Hardware petrol station system

From the **Figure 27**, it can be seen that the petrol station hardware system is equipped with the customer interface such as LCD screen, NFC RFID reader, temperature and humidity sensor and keypad membrane. The user(manager) is able to interact easily providing this software is user-friendly, and besides that, the hardware is robust, all in one. The nozzles of fuel pump are directly connected.

4.3.1.1 Petrol station hardware—user interface

Initially, when the power board is supplied, the LCD screen shows the functions that the board performs when pressing the letter buttons on the keypad as shown in **Figure 28**.

- When we want to check the balance in RFID tag, we press A letter in keypad as shown in **Figure 29**. After pressing A letter, the LCD screen will display a message telling the customer to insert his card for the purpose of checking as shown in **Figure 30**. After checking the card, a message will appear on the screen showing the user's balance in Iraqi dinar as shown in **Figure 31**.

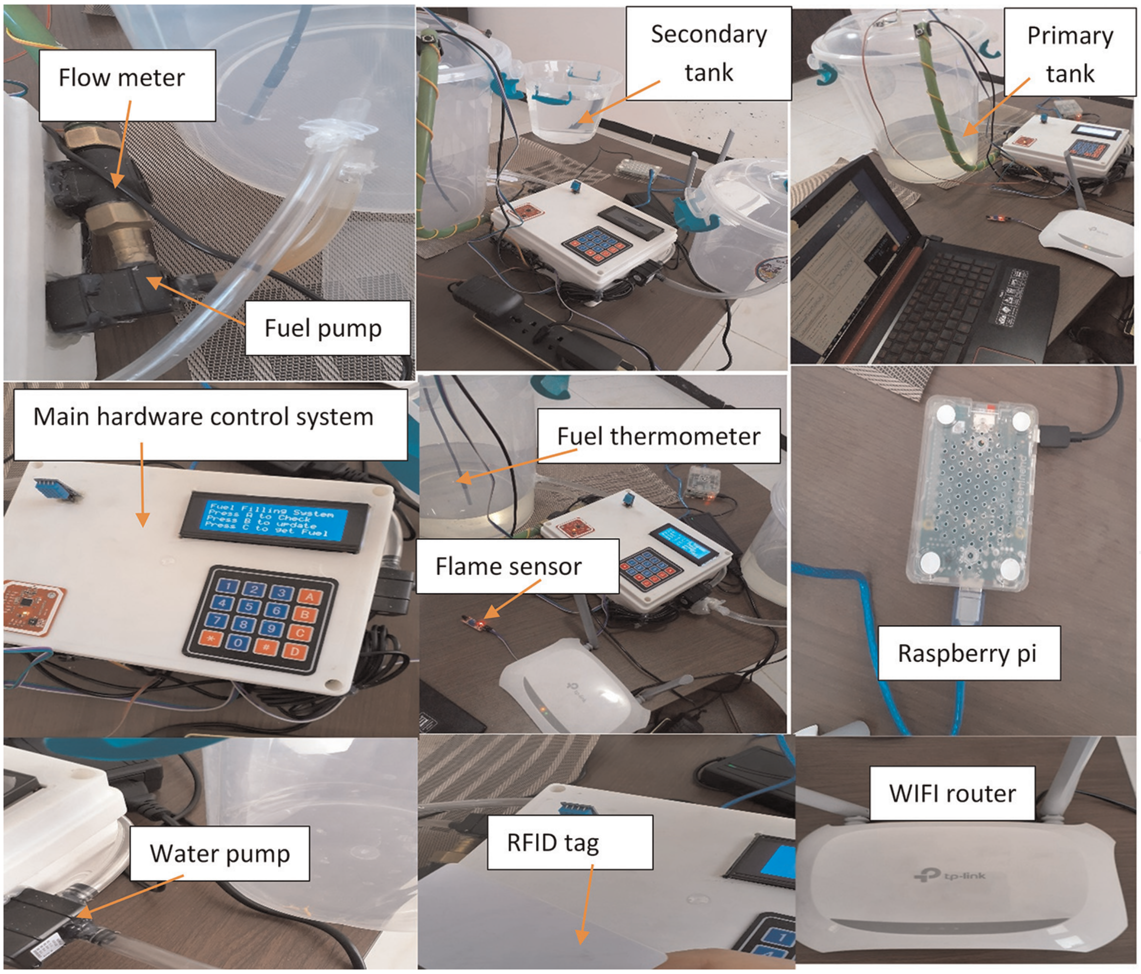


Figure 27.
The overall hardware system.

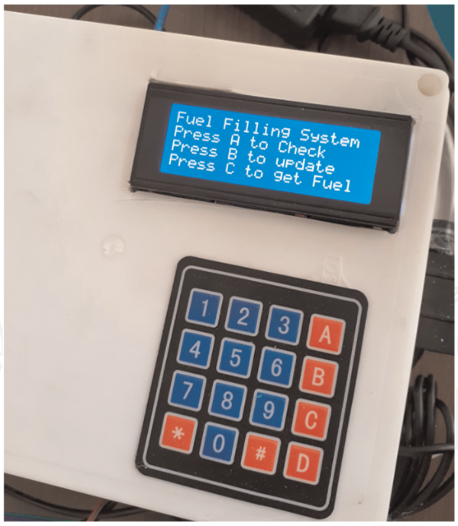


Figure 28.
Initial display on LCD.

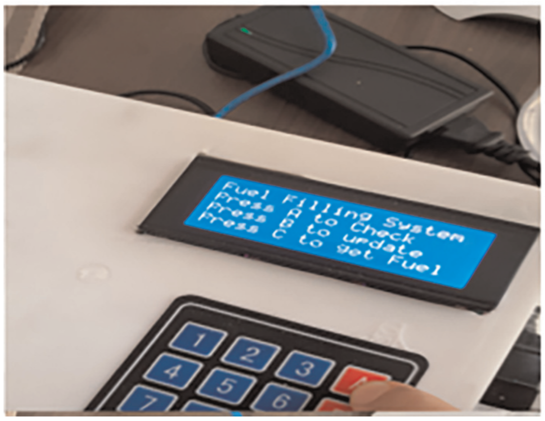


Figure 29.
Press A to check.

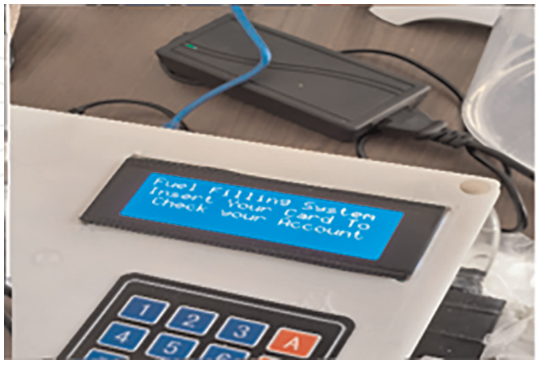
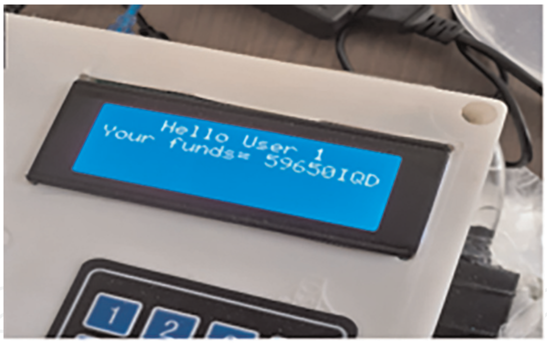


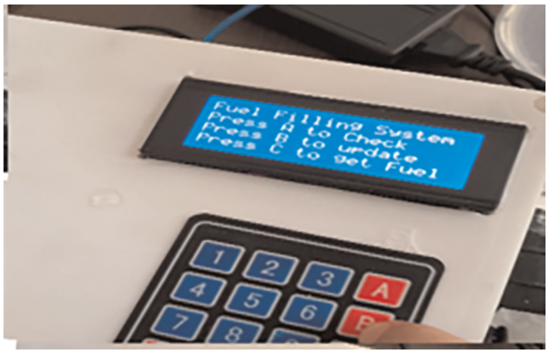
Figure 30.
Insert card to check.

- For the purpose of updating or recharging the balance in the card, we press the letter B as shown in **Figure 32**. After pressing B letter, the LCD screen will display a message telling the customer to insert his card for the purpose of updating as shown in **Figure 33**. After identifying the ID of the card, a message appears on the screen asking the user, the amount to be recharged in the card as shown in



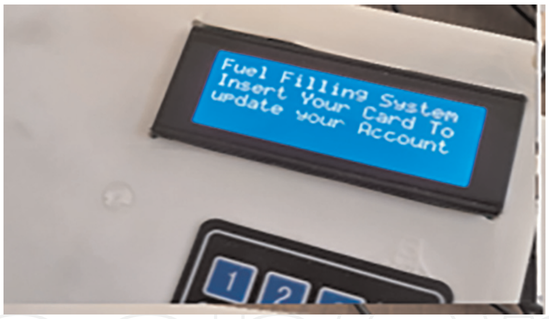
Hello User 1
Your funds= 59650IQD

Figure 31.
User ID and your balance in card.



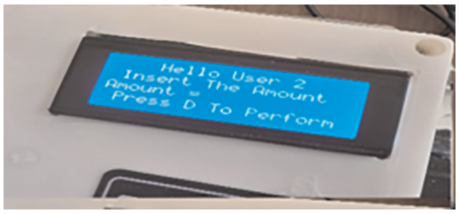
Fuel Filling System
Press A to Check
Press B to update
Press C to get Fuel

Figure 32.
Press B to update.



Fuel Filling System
Insert Your Card To
update your Account

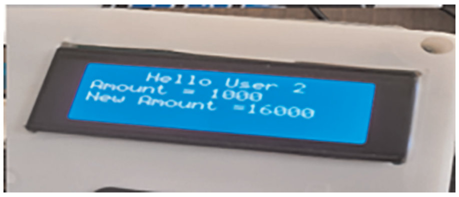
Figure 33.
Insert card to update.



Hello User 2
Insert The Amount
Amount =
Press D To Perform

Figure 34.
Insert the amount of balance.

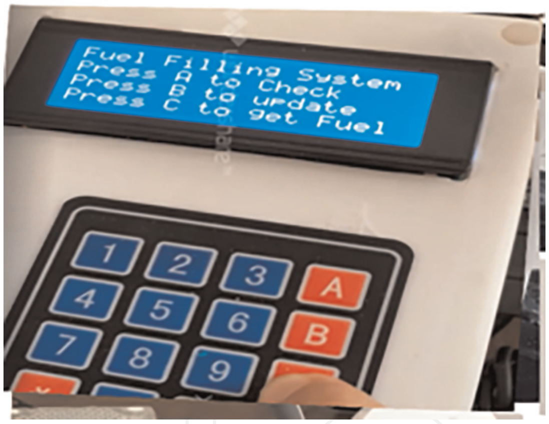
Figure 34. After entering the required amount and pressing the letter D, the updated quantity will appear on the screen in addition to the total amount after adding the new amount as shown in **Figure 35**.



Hello User 2
Amount = 1000
New Amount =16000

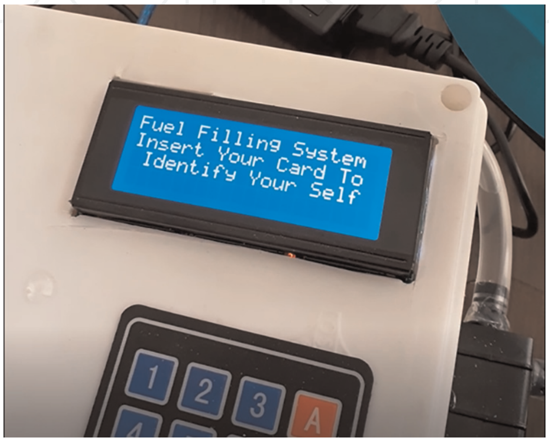
Figure 35.
New amount after update.

- In order to obtain fuel, we press the letter C as shown in **Figure 36**. This function is common to all types of regular users (who has a balance and is deducted from his card after filling the fuel) and government users (who does not have a balance on the card, but whose debts are recorded on the government's account). After press C, the LCD screen will display a message telling the customer to insert his card to identify yourself as shown in **Figure 37**. After identifying the ID of the card and thus determining the type of customer, a message appears on the screen telling the customer to enter the amount of fuel to be filled as shown in **Figure 38**. The filling process begins after entering an amount of fuel less than the available balance as shown in **Figure 39**, and the amount deducted from the card and the amount of fuel filled appear on the LCD screen as shown in **Figure 40**. Because the fuel meter has a small error rate, we enter the quantity minus the error percentage.



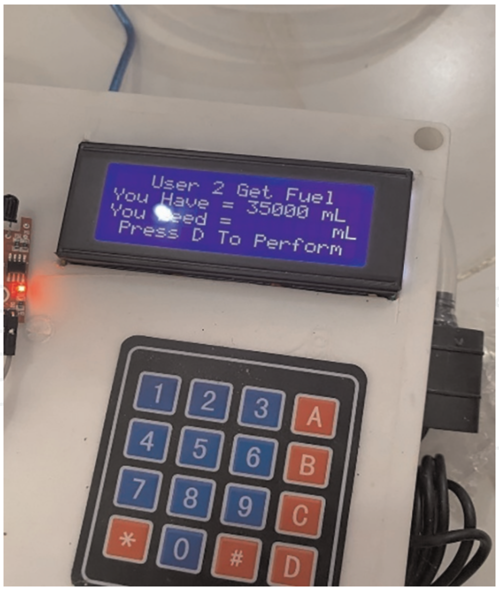
Fuel Filling System
Press A to Check
Press B to update
Press C to get Fuel

Figure 36.
Press C to get fuel.



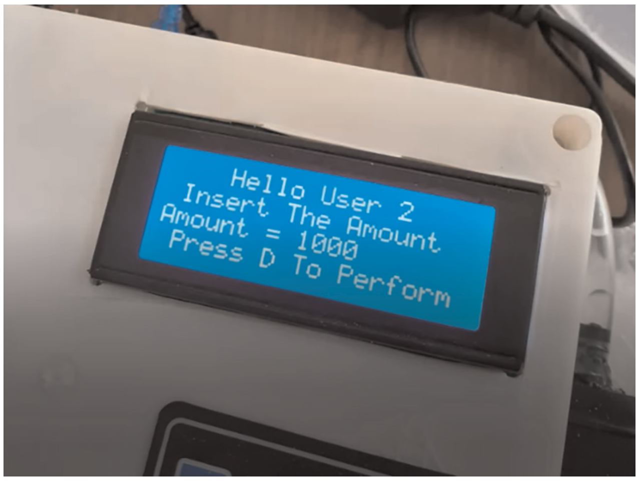
Fuel Filling System
Insert Your Card To
Identify Your Self

Figure 37.
Insert card to identify.



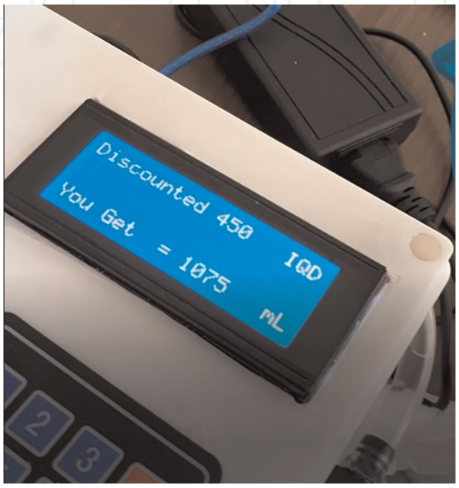
```
User 2 Get Fuel
You Have = 35000 mL
You Need =       mL
Press D To Perform
```

Figure 38.
Insert quantity of fuel we want to get.



```
Hello User 2
Insert The Amount
Amount = 1000
Press D To Perform
```

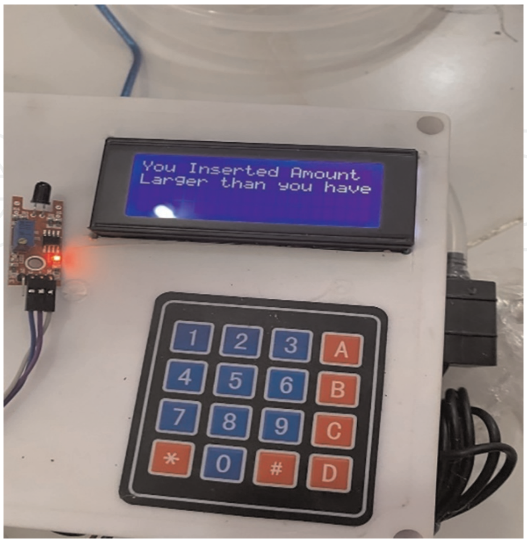
Figure 39.
Insert quantity of fuel we want to get.



```
Discounted 450    IQD
You Get   = 1075   mL
```

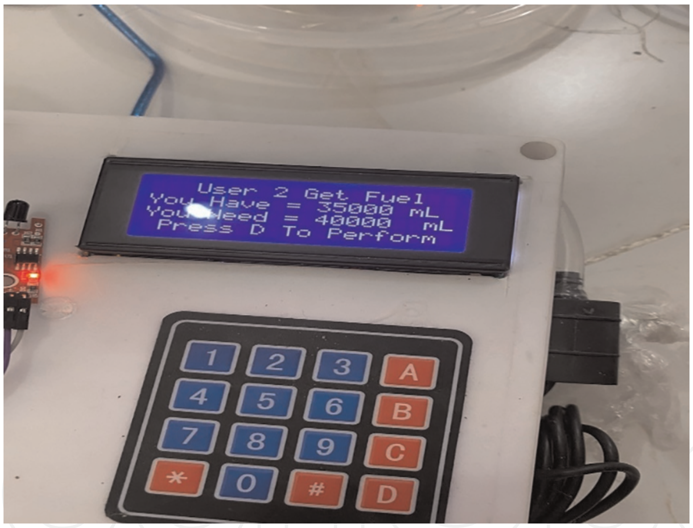
Figure 40.
Price and quantity of filled fuel.

- When entering a quantity of fuel greater than the available balance in the card as shown in **Figure 41**, a message will appear telling the customer that the quantity entered is greater than the available balance as in the **Figure 42**.



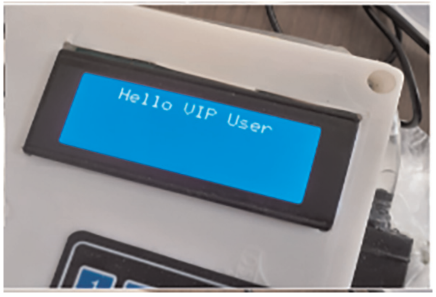
You Inserted Amount
Larger than you have

Figure 41.
The quantity entered is greater than the available balance.



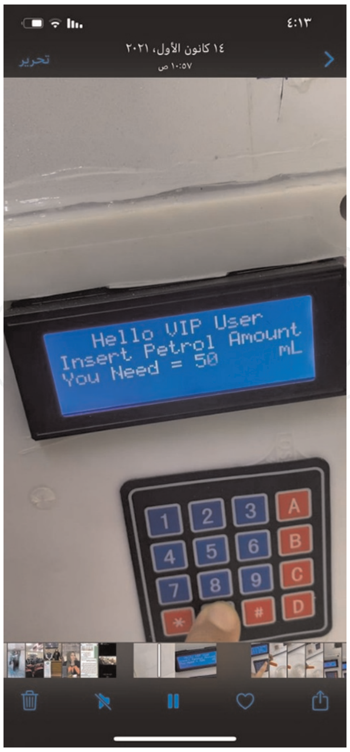
User 2 Get Fuel
You Have = 35000 mL
You Need = 40000 mL
Press D To Perform

Figure 42.
Entering more fuel than the available balance.



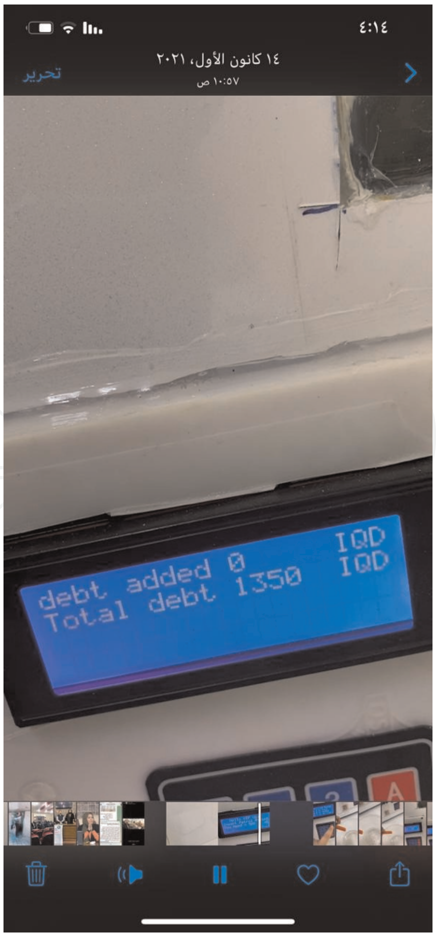
Hello VIP User

Figure 43.
Check VIP user.



```
Hello VIP User
Insert Petrol Amount
You Need = 50      mL
```

Figure 44.
VIP user identify to get fuel.



```
debt added 0      IQD
Total debt 1350   IQD
```

Figure 45.
Debts added and total debts.

- When VIP government customer (want to get fuel, initially he swipes his card on the NFC RFID reader to identify as shown in **Figure 43**. Then enter a quantity of fuel VIP wants to get as shown in **Figure 44**. After fuel filling message display on LCD screen telling VIP user about quantity filled and depts as shown in **Figure 45**.

4.3.1.2 Hardware for petrol stations—primary fuel tank

- Level of the primary fuel tank

  By adding Ultrasonic sensor installed at the top of the primary fuel tank for measuring fuel level as in the **Figure 46**. Ultrasonic sensor is installed on the Arduino Mega microcontroller.

- Fuel temperature

  By adding DS18B20 digital thermometer to measure a fuel temperature which enters inside the primary fuel tank as in the **Figure 47**, where by knowing the fuel temperature it is possible to predict the fuel evaporation rate and its density. Fuel thermometer is installed on the ESPS-32 microcontroller.

4.3.1.3 Petrol station hardware- temperature and humidity weather

By adding DHT11 sensor on the petrol station hardware box and connecting it to the ESP-32S microcontroller to represent a second fuel station, that sends temperature and humidity to ThingsBoard server.

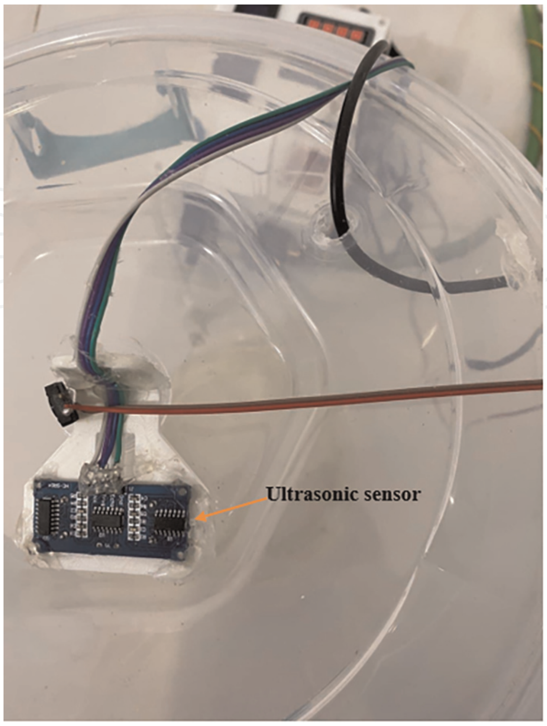


Figure 46.
Ultrasonic sensor at the top of the primary fuel tank.

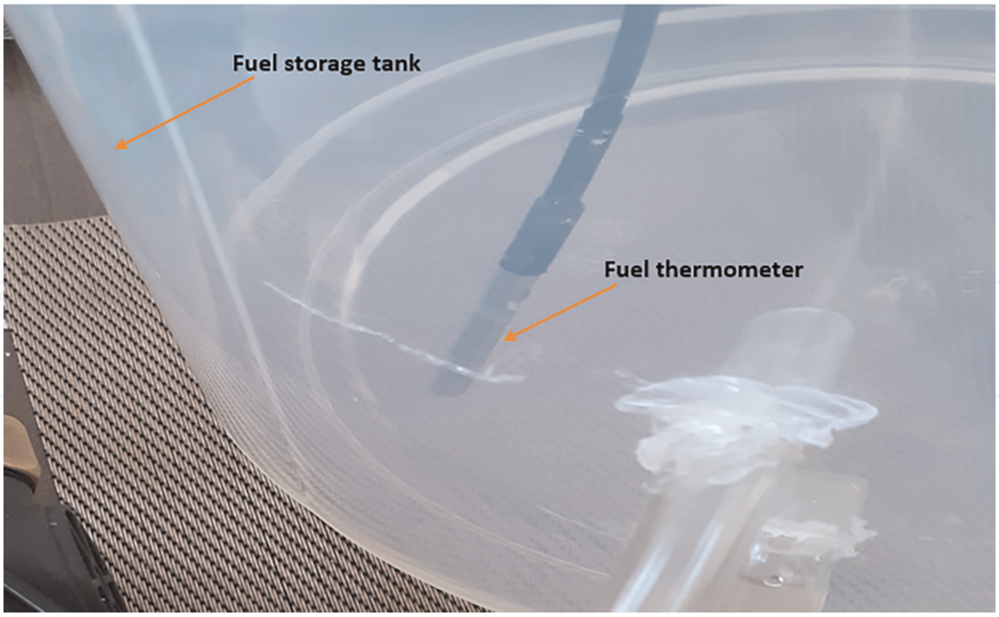


Figure 47.
Fuel thermometer inside the primary fuel tank.

4.3.1.4 Hardware for petrol stations with a fire-protection system

For the purpose of fire protection, a flame sensor was included in this system to sense fires and extinguish them automatically or manually. The water pump was connected to the sides of the main hardware box, which was connected to the flame sensor, which triggers an alarm sound when a flame is detected and sends its data to the microcontroller, which instructs the relay switch associated with the water pump to operate it. This pump can also be controlled manually via a push button and is done Turn on the pump. **Figure 48**, shows the water pump installed on the box and the flame sensor.

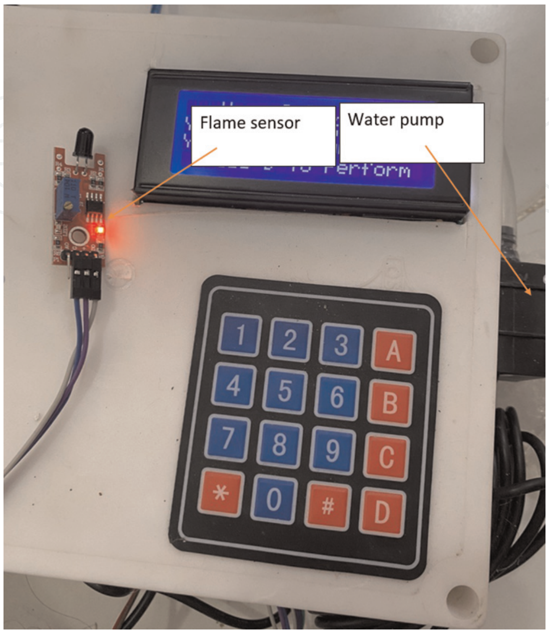


Figure 48.
Fire-protection system.

4.3.2 Petrol station monitor system

The monitor system to share data, manage and control is expected to provide the fuel stations manager with information about a petrol station in real time remotely shown in **Figure 49**. The system will show sales and the volume of petrol bought on a specific date, and will provide manager with data as follow:

1. Users interface

2. The primary tank of a fuel

3. Weather conditions (temperature and humidity)

4.3.2.1 User interface for a petrol station monitoring system

All of the customer's actions, such as filling the tank, paying the bill, total sales, the number of cars, debts, and so on, are sent to the server and monitored by the company that follows up on three fuel stations, where each station subscribes to the device's identity and the monitoring company's access token. **Figure 50**, shows the operations performed by the customers as they appear on the dashboard.

4.3.2.2 Petrol station monitor system-primary fuel tank

By monitoring the main fuel tank, it is possible to verify the following:

4.3.2.2.1 Level of fuel

Ultrasonic sensor is installed on the Arduino Mega microcontroller.
The sensor sends fuel level information to microcontroller considering each

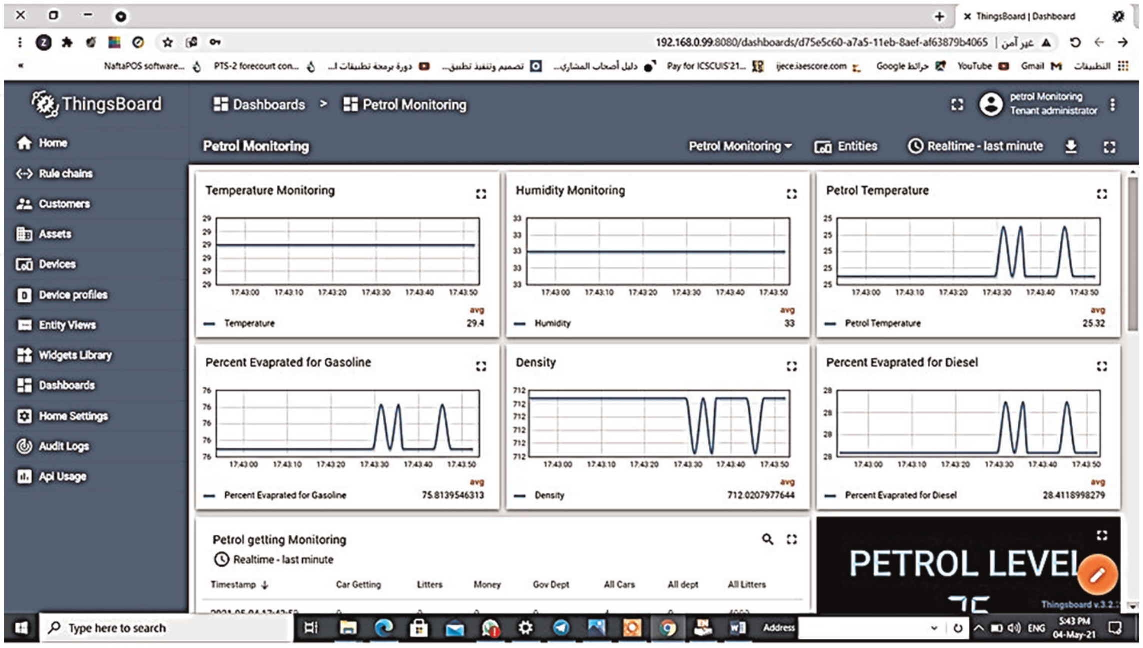
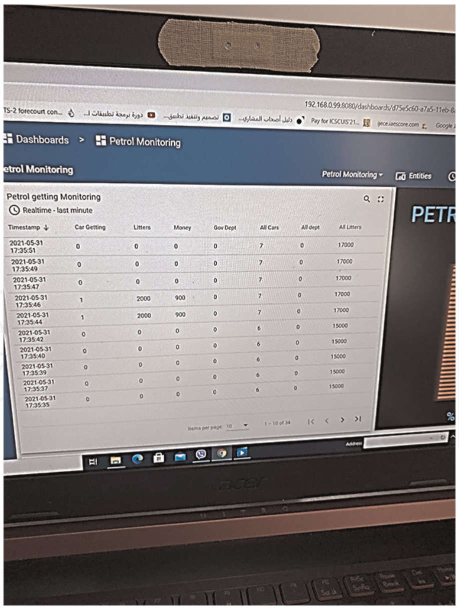


Figure 49.
Petrol station monitoring system Interface.



192.168.0.99:8080/dashboards/d75e5c60-a7a5-11eb-
Dashboards > Petrol Monitoring
Petrol Monitoring
Petrol Monitoring
Entities
PETR

Petrol getting Monitoring

Realtime - last minute

Timestamp ↓	Car Getting	Litters	Money	Gov Dept	All Cars	All dept	All Litters
2021-05-31 17:35:51	0	0	0	0	7	0	17000
2021-05-31 17:35:49	0	0	0	0	7	0	17000
2021-05-31 17:35:47	0	0	0	0	7	0	17000
2021-05-31 17:35:46	1	2000	900	0	7	0	17000
2021-05-31 17:35:44	1	2000	900	0	7	0	17000
2021-05-31 17:35:42	0	0	0	0	6	0	15000
2021-05-31 17:35:40	0	0	0	0	6	0	15000
2021-05-31 17:35:39	0	0	0	0	6	0	15000
2021-05-31 17:35:37	0	0	0	0	6	0	15000
2021-05-31 17:35:35	0	0	0	0			

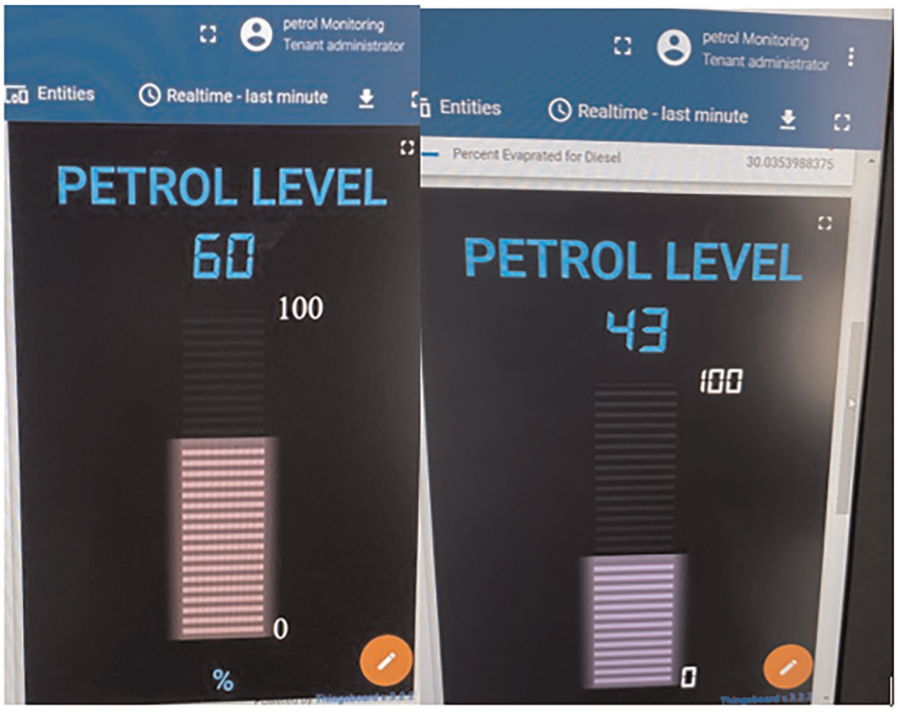
Items per page: 10 1 – 10 of 34

Figure 50.
Petrol monitoring on ThingsBoard dashboard.

microcontroller is a fuel station (MQTT client) and sends its information to the ThingsBoard server as shown in **Figure 51**.

4.3.2.2.2 A fuel temperature

In this thesis project, we use fuel temperature sensor to send their information from microcontroller (as MQTT Clint) to calculate percentage evaporation of fuel



Percent Evaprated for Diesel 30.0353988375

Figure 51.
Monitoring fuel level for two states on the dashboard.

gasoline and diesel and sends its information to the ThingsBoard server (as MQTT broker) installed on Raspberry Pi as a local server or ThingsBoard installed on a cloud server to be shown on the dashboard. **Figure 52a** and **Figure 52b** shows petrol temperature average 25.32 Co and 43.46 Co respectively.

4.3.2.2.2.1 Evaporation rate

Figure 53a and **Figure 53b** shows percentage evaporation of gasoline according to temperature change at average 25.32Co and 34.46 Co respectively, where the effect of temperature appears, with 25.32Co the percentage of evaporation of gasoline becomes 75.8 and then this percentage rises to 83 at 34.46 C°. **Figure 54a** and **Figure 54b** shows percentage evaporation of diesel fuel according to temperature change at average 25.32C° and 34.46 C° respectively, where the effect of temperature appears with 25.32Co the percentage of evaporation of diesel becomes 28.4 and then this percentage rises to 30 at 34.46 Co.

4.3.2.2.2.2 Density

Figure 55a and **Figure 55b** show the variation density of gasoline according to temperature change at average 25.32C° and 34.46 C° respectively, where the effect of

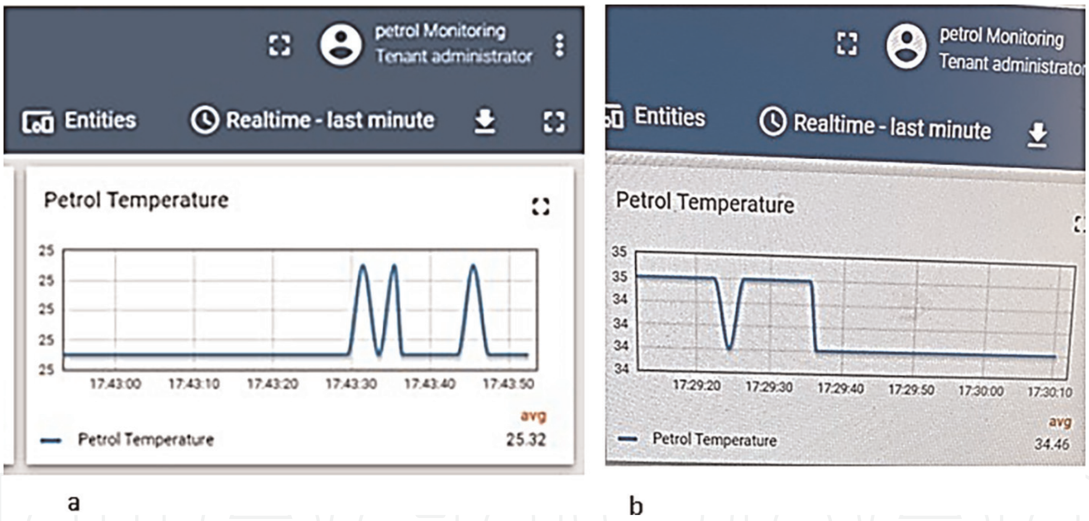


Figure 52.
Gasoline temperature as shown on dashboard.

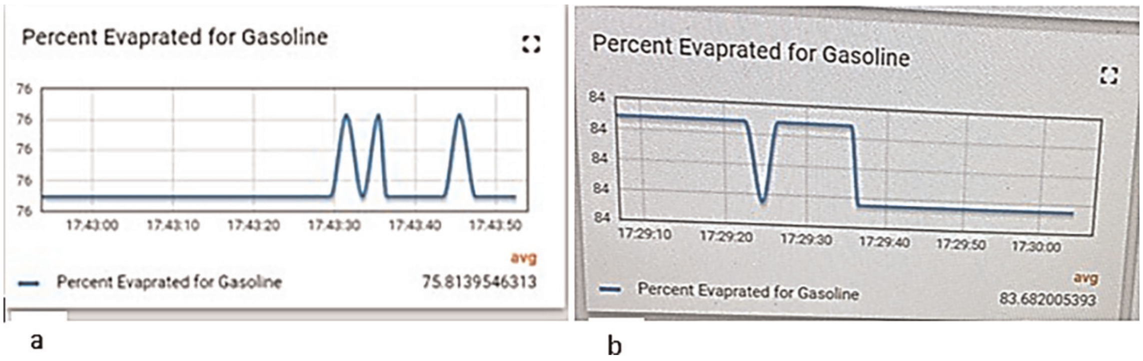


Figure 53.
Percent evaporated for gasoline on dashboard.

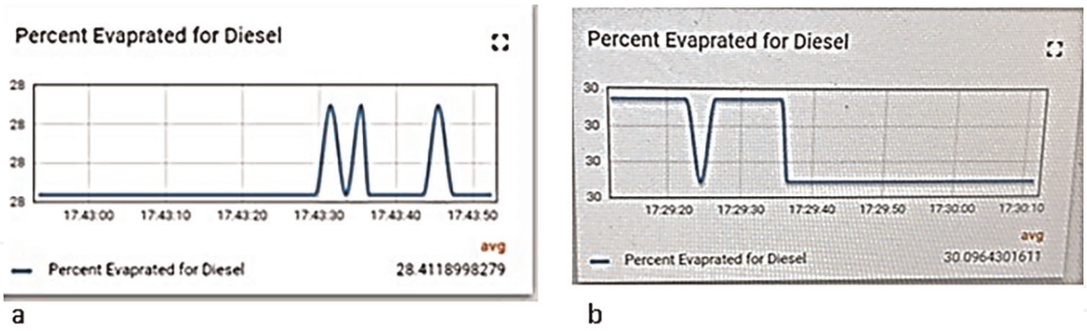


Figure 54.
Percent evaporated for diesel on dashboard.

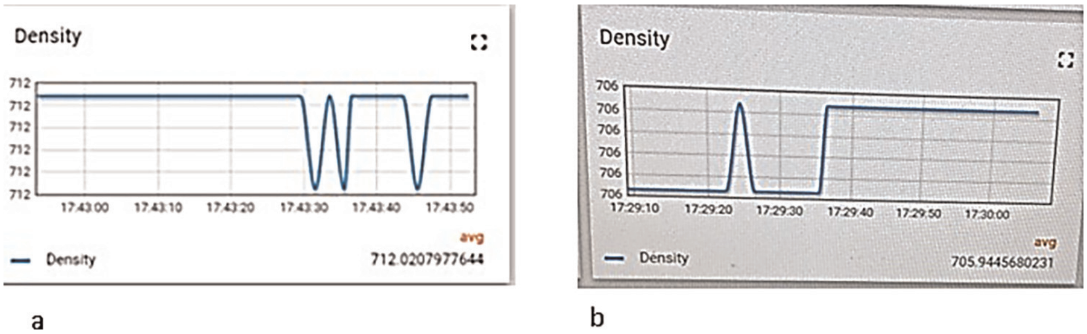


Figure 55.
The density of gasoline vs. temperature on the dashboard.

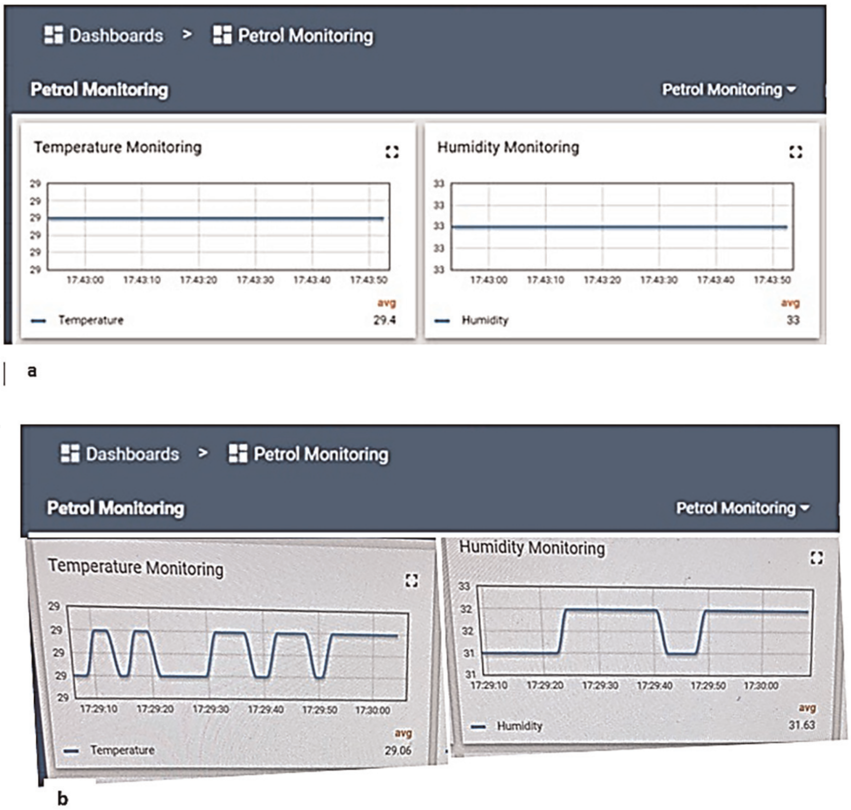
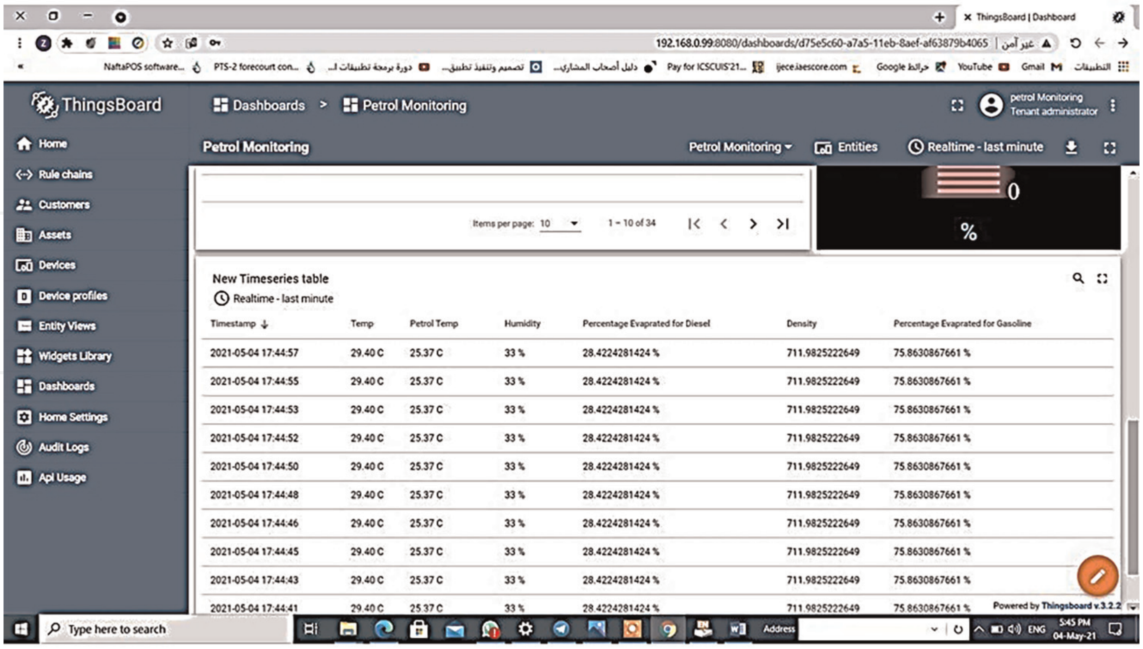


Figure 56.
Temperature and humidity are being monitored on the dashboard for two states from the second fuel station (a and b).



New Timeseries table

Realtime - last minute

Timestamp ↓	Temp	Petrol Temp	Humidity	Percentage Evaprated for Diesel	Density	Percentage Evaprated for Gasoline
2021-05-04 17:44:57	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:55	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:53	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:52	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:50	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:48	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:46	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:45	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:43	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %
2021-05-04 17:44:41	29.40 C	25.37 C	33 %	28.4224281424 %	711.9825222649	75.8630867661 %

Figure 57.
Timeseries table on the dashboard.

temperature appears with 25.32C° the density becomes 712 kg/m3 and then this density decreases to 7.6 at 34.46 C°.

4.3.2.2.3 Petrol station monitor system-temperature and humidity weather

DHT11 sends temperature and humidity information to a ThingsBoard server installed on a Raspberry and displays this information on the ThingsBoard dashboard, so that the fuel station is monitored and any station that sends its information on the access token and the identity of the device that is connected to the display dashboard. **Figure 56a** and **Figure 56b** shows the temperature and humidity of the second fuel station for two different states as they appear on the dashboard monitoring the three gas stations. Monitoring the temperature of the fuel and its consequences in determining the evaporation rates of fuel (gasoline and diesel fuel), as well as determining the density and the consequent determination of whether the fuel is tainted with invalid mixtures. Monitoring these coordinates from a fuel station whose client MQTT is microcontroller ESP-32S. All of these variables are monitored on the dashboard of the ThingsBoard server as shown in **Figure 57** in timeseries table of the dashboard.

5. Conclusions

The smart petrol station system is expected to solve various problems as stated in the first chapter. This project has added a new technology in which a wireless communication can be established between petrol station stations to monitor and control a system which may be located hundreds of kilometers away. Thus, the monitoring process of the petrol station stations can be simplified by this technology. The implementation and test the automation system which build based on IoT leads to the conclusions below.

1. Connected three petrol station systems to Internet represent the basic structure to IoT of this application in order to share several types of information related to users, environment and database of systems.

2. Practical results show that smart petrol station system-based on IoT scheme is effective and more manageable to resolve the all problems of traditional filling stations.

3. Using the method of distinguishing between government card customers and regular card customers is very useful for reducing corruption.

4. The establishment of this project in retail outlets can reduce the period utilization which associated with the depth of petrol in tanks which was ensured manually with the assistance of big gauging scale, testing the heat and density of petrol, stock totaling, etc.

5. Setting an alarm system in the event of a trip or an overload on the pump, thus extending the life of the fuel pump. Also, Maintenance of this system is very easy and low-cost.

6. Measuring temperature of weather is very useful where there are volume losses in hot weather and volume gains in cold weather of petrol due to thermal expansion.

Author details

Zahra'a M. Baqir[1]* and Hassan J. Motlak[2]

1 Department of Electrical and Electronic Engineering, Thi-Qar University, Nasiriyah, Iraq

2 Department of Electrical and Electronic Engineering, Babylon University, Babylon, Iraq

*Address all correspondence to: zmbhmaly@gmail.com

References

[1] Gupta P, Patodiya S, Singh D, Chhabra J, Shukla A. IoT based smart petrol pump. In: 2016 Fourth International Conference on Parallel Distributed and Grid Computing, Waknaghat, India. Vol. 2016. IEEE; 2016. pp. 28-32

[2] Gowri Shankar EMR, Baskaran D. Automation of petrol bunk using biotelemetry system and petro card. International Journal of Engineering Research & Technology. 2016;**4**:1-4

[3] Wavekar Asrar A, Patel Tosif N, Pathan Saddam I, Pawar H. RFID based automated petrol pump. International Journal for Scientific Research & Development. 2016;**4**:1333-1334

[4] Kumaresan P, Babu Sundaresan Y. IoT based retail automation of fuel station and alert system. In: Materials Science and Engineering Conference Series. IOP Conference Series; 2017. 042072

[5] Chandana K, Chirag M, Burugupalli PKK, Akarsh J. Raspberry pi based RFID smart card refuelling system. In: Perspectives in Communication, Embedded-Systems and Signal-Processing-PiCES Journal, Bangalore, India. Vol. 2. 2018. pp. 149-152

[6] Dongarsane C, Dalavi P, Golandag S, Powar S. Self-operated petrol pump. International Journal of Advance Research, Ideas and Innovations in Technology. 2017;**3**:231-234

[7] BK R. IOT based petrol bunk management for self-operation using RFID and raspberry Pi. International Journal of Recent Technology and Engineering (IJRTE). 2019;**8**

[8] Nitha C, Velayudhan RKR, Rashida MH, Risvana MP, Sreemol CV. Automatic fuel filling system. International Journal of Advanced Research in Computer and Communication Engineering. 2019;**8**: 128-133

[9] Rao SS, Prasad VS. Centralized automation of petrol bunk management and safety using RFID and GSM technology. In: 2017 International Conference on Intelligent Computing and Control (I2C2). Coimbatore, India: IEEE; 2017. pp. 1-5

[10] Makdisie C, Haidar B. Using SCADA & microcontroller for monitoring and automation of petroleum station tanks. International Journal of Engineering Research and General Science. 2017

[11] Deepa D, Deepashree B, Deepu N, Divya R. Multipurpose self fuel dispensing automated framework utilizing RFID prepaid cards. In: 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India. IEEE; 2018. pp. 69-74

[12] Pranto MB, Rahman MM. Vehicle fuel monitoring and management using RFID authentication and telematics notification: An automated petrol refueling & notifying system. In: 2019 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Bali, Indonesia. IEEE; 2019. pp. 477-482

[13] Sembodo BP, Atmajaya H. Design of smart fuel station and management system based on Arduino and internet of things. BEST: Journal of Applied Electrical, Science, & Technology. 2021; **3**:5-10

[14] Gupta N, Brandt SR, Wagle B, Wu N, Kheirkhahan A, Diehl P, et al. Deploying a task-based runtime system on raspberry pi clusters. In: 2020 IEEE/ACM 5th International Workshop on Extreme Scale Programming Models and Middleware (ESPM2). GA, USA: IEEE; 2020. pp. 11-20

[15] Arduino. Arduino Software. (n.d.). Available from: https://www.arduino.cc/en/software

[16] Labcenter Electronics. Proteus Software. (n.d.). Available from: https://www.labcenter.com/