2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan

Fuel Management System

Areeg Abubakr Ibrahim Ahmed, Siddig Ali Elamin Mohammed, Mohamed Almudather Mahmoud Hassan Satte,
Department of Electrical and Electronics Engineering
University of Khartoum
Khartoum, Sudan
areeg.abubakr.ib@hotmail.com,siddigelamin@hotmail.com,msatti17@hotmail.com

***Abstract*—Monitoring Systems are necessary to track and understand the changes that take place in environments. Remote monitoring and data collection systems are useful and effective tools to collect information from fuel storage tanks. The fuel management system is a monitoring device built on the Raspberry-Pi computer, it takes information about tank's fuel level in real time through its sensor and live streaming of the site, then uploads it directly to the internet, where it can be read anytime and anywhere through web application. This paper presents the implementation of monitoring system based on internet of things technology to protect the tower sites from theft and provide security to remote locations.**

Keywords—IoT; Raspberry-Pi;ultrasonic sensor;etape;xively.

I. INTRODUCTION

As fuel prices climb, some issues and challenges are facing telecom industry regarding the power supply of towers. "In some African countries up to 80% of the tower sites have no access to the power grid, those sites are usually powered by diesel or gasoline generators". "Between 20% and 35% of the fuel intended for powering the tower site in Africa is stolen", which results in financial losses to telecom companies. To avoid this, we are implementing such a system, we have used Internet of things which has become a basic and necessary technology for monitoring of remote location via web or android application. The installation of around 75,000 new towers around the world every year that are off-grid, shows clearly the rising importance of fuel management system.

Internet of Things (IoT) is a concept and a paradigm that considers pervasive presence in the environment of a variety of things that through wireless and wired connections and unique addressing schemes are able to interact with each other and cooperate with other things to create new applications/services and reach common goal. A world where the real, digital and the virtual are converging to create smart environments that make energy, transport, cities and many other areas more intelligent. The goal of the Internet of Things is to enable things to be connected anytime, anyplace, with anything and anyone ideally using any path/network and any service [1].

II. IOT ARCHITECTURE

IoT architecture consists of different suite of technologies supporting it. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IoT deployments in different scenarios. The functionality of each layer is described below:

- Sensor Layer

The lowest layer is made up of smart objects integrated with sensors. The main function of this layer is to obtain the various types of static / dynamic information of the real world through various types of sensors and to share with Internet access.

- Gateways and Networks

Large volume of data will be produced by these sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium. The network helps to distinguish each object that is interconnected in the physical world. Current networks, often tied with very different protocols, have been used to support machine-to-machine (M2M) networks and their applications.

- Management Service Layer

The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices. One of the important features of the management service layer is the business and process rule engines. Data management is the ability to manage data information flow. With data management in the management service layer, information can be accessed, integrated and controlled [2].

- Application Layer

This layer at the top of the stack is responsible for delivery of various applications to different users in IoT. It consists of protocols that focus on process-to-process communication across an IP network and provides a firm communication interface and end-user services [3].

III. COMPONENTS AND TECHNOLOGIES

A. Ultrasonic sensor

Ultrasonic sensors work by transmitting a pulse of sound, this pulse travels away from the range finder in a conical shape at the speed of sound (340 m/s). The sound reflects off an object and back to the range finder. The sensor interprets this as an echo and calculates the time interval between sending the signal and receiving the echo [4].

978-1-5090-1809-3/17/$31.00 ©2017 IEEE

B. *"Etape" Fluid Level Sensor*

The etape sensor's envelope is compressed by hydrostatic pressure of the fluid in which it is immersed resulting in a change in resistance which corresponds to the distance from the top of the sensor to the fluid surface. The etape sensor provides a resistive output that is inversely proportional to the level of the liquid: the lower the liquid level, the higher the output resistance; the higher the liquid level, the lower the output resistance.

C. *The Raspberry Pi Single Board Computer overview*

The Raspberry Pi is a low cost, credit-card sized single-board computer that is capable of doing everything a desktop computer would do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games [5].

D. *Stand-alone power supply*

A stand-alone power system (SAPS or SPS), also known as remote area power supply (RAPS), is an off-the-grid electricity system for locations that are not fitted with an electricity distribution system. Typical SAPS include one or more methods of electricity generation, energy storage, and regulation. Electricity is typically generated by one or more of the following methods: Photovoltaic system using solar panels, Wind turbine, geothermal source Micro combined heat and power Micro Hydro Diesel or bio-fuel generator Storage is typically implemented as a battery bank

E. *Xively*

Xively (formerly known as Cosm and Pachube) offers an Internet of Things (IoT) platform as a service, business services, and partners that enable businesses to quickly connect products and operations to the Internet.

IV. METHODOLOGY

A. *Hardware design*

The hardware design of the system includes designing the hardware units and the interface between those units:

1. Data acquisition
   - Sensing
   - Camera surveillance system
2. Data processing
3. Power unit

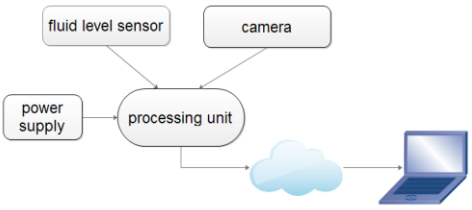


Fig. 1. Schematic diagram of the system

1) *Data acquisition*

a) *Sensing*

The purpose of this unit is to detect (sense) the fluid level using ultrasonic sensor or "etape" that were chosen carefully to achieve the best performance. The detected data should then be converted into digital formats corresponding to each of the measured values. This conversion depends on the type of sensor used. Digital sensors will give a digital output suitable for the R-Pi's (digital only) input, while analog sensors will need analog to digital conversion.

b) *Camera surveillance system*

The Raspberry Pi camera module transfers data through an extremely fast Camera Serial Interface (CSI-2) bus directly to the Broadcom BCM2835 system-on-chip (SoC) processor. It does this through a 15-pin ribbon cable, also known as a flex cable, and connects to the surface mount ZIF 15 socket. The two data lanes on the CSI-2 bus provide a theoretical 2 Gbps bandwidth, which approximates to around 5 MP resolution. Therefore, this is what is expected the new cameras to have. It is very likely to have a maximum video recording resolution of 1920 pixels × 1080 pixels at around 30 frames per second. This is reasonable for the current technology that is around. Fig. 2 shows flow chart shows this approach.

Fig. 2. Flow chart of the operation of Raspberry Pi camera

2) *Data processing*

The processing unit consists of two parts ADC and R-Pi.

a) *ADC*

The process of analog to digital conversion usually involves comparing two analogue signals: an input signal and some reference signal. The comparison is carried out with a circuit called a comparator. A comparator circuit is essentially a high-gain differential amplifier. The function of this unit is to convert analog output of sensor such as etape into digital because R-Pi doesn't handle analog output.

b) *Raspberry-Pi*

This is the most important unit and the core of the system. It handles all the processing and controlling needed for system

to function. It receives the sensing information, processes it, returns the corresponding values, and generates the necessary controls to guide the data to the desired destination. Fig. 3 shows the components of this unit.

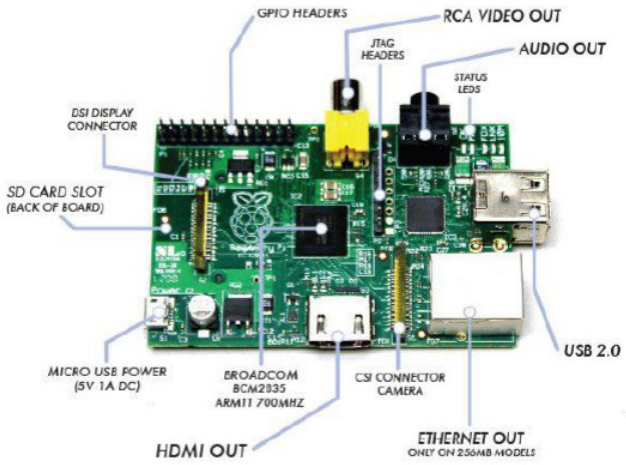


Fig. 3. The Raspberry Pi board components

The sensor will be connected to the R-Pi's GPIO pins in configuration that will be shown in the implementation part. The main concept of the wiring is that digital sensors are connected directly to the raspberry pi's GPIO, while analog sensors like etape will be connected to an analog to digital converter, which will be directly connected to the raspberry pi. The raspberry pi will handle the processing of data and upload it into cloud as shown in the below diagrams:

Fig. 4. Interface of R-Pi and ultrasonic

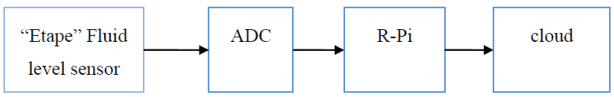


Fig. 5. Interface of R-Pi and etape

3) Power unit

Lithium Ion Battery Charger for Solar-Powered Systems charging from a solar power supply, a wall adapter or a USB port can also be used to charge Li-Ion/Li-polymer batteries. Fig. 6 shows an example of how to combine 2 power inputs. A P-channel MOSFET, M1, is used to prevent back conducting into the 2nd power supply when the 1st power supply is present and Schottky diode, D1, is used to prevent 2nd power supply loss through the 1kΩ pull-down resistor.

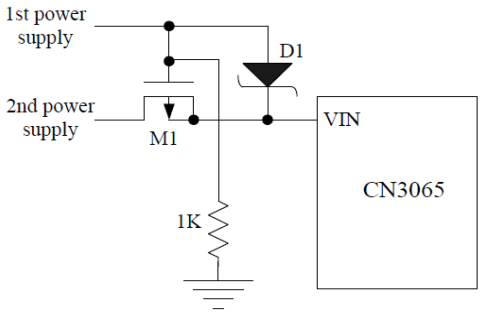


Fig. 6. Combining 2 Inputs Power Supply

B. *software design*

Before writing the code for the system, several software dependencies must be installed. These dependencies add more functionality to the use of Python on the R-Pi and make the software design process easier. For example, some dependencies let Python uses the Pi's interfaces or interface with its GPIO pins. The installation of dependencies requires internet connection. In the execution part, the software life cycle management phases were employed for the web application as well and those phases will be explained in details.

1) Web application

The sensor reads the data of the fuel level, transfers the value to processing unit (Raspberry-pi). The processing unit sends the value through point to point protocol to the internet (cloud). The website module retrieves the data from the internet and displays it to the users, using database for authentication.

This website is responsible of user operations, and includes:

- Add new user.
- See the current fuel status.
- See the fill date.
- See the fuel consumption rate.
- Watch the live streaming.

V. IMPLEMENTATION

The research experiment was set up using ultrasonic and etape in data acquisition part to find the effective sensor. The following experimental steps were taken

- The code for running the ultrasonic with the raspberry pi and sending data to xively cloud was written and the ultrasonic sensor was connected to Raspberry Pi.
- The code for running the etape with the raspberry pi and sending data to xively cloud was written and the etape sensor was connected to Raspberry Pi through ADC.
- The code was written for running the camera module with the raspberry pi and live streaming.
- The code was written to retrieve the data from the cloud and display it through the web application.

A. Hardware implementation

1) Connecting The Raspberry Pi To Ultrasonic

Connecting the VCC,TRIG,ECHO,GND of the ultrasonic to the GPIO 5v[pin 2],GPIO 23[pin 16],GPIO 24[pin 18],GPIO GND[pin 6] of the raspberry pi Respectively[6].

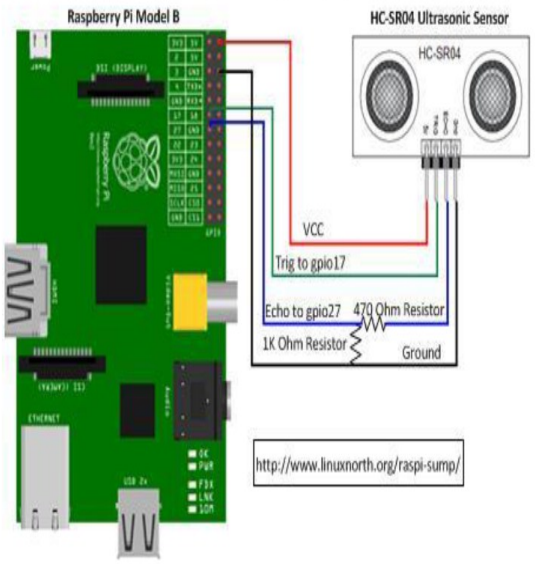


Fig. 7. R-Pi and ultrasonic (HC-SR04) connection schematic

2) Connecting The Raspberry Pi To etape

The etape has been connected to the Raspberry Pi through ADC

TABLE I. CONNECTING THE RASPBERRY PI ETAPE TO ADC (MCP3008)

MCP 3008 Pin	Pi GPIO Pin
16 (VDD)	1 (3.3 V)
15 (VREF)	1 (3.3 V)
14 (AGND)	6 (GND)
13 (CLK)	23 (GPIO11 SPI0_SCLK)
12 (DOUT)	21 (GPIO09 SPI0_MISO)
11 (DIN)	19 (GPIO10 SPI0_MOSI)
10 (CS)	24 (GPIO08 CE0)
9 (DGND)	6 (GND)

TABLE II. CONNECTING THE CHEMICAL ETAPE TO ADC

Etape pin	MCP 3008 Pin
Vs	16 (VDD)
Vout	1 (CH0)
GND	9 (DGND)

3) Connecting the camera module to the raspberry pi

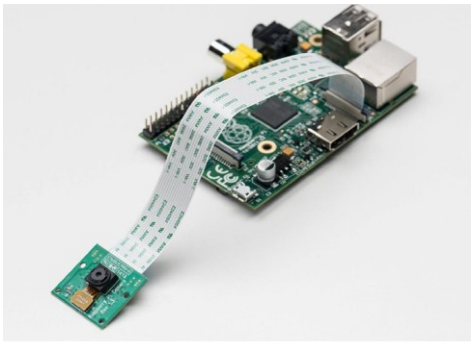


Fig. 8. R-Pi and camera module connection schematic

Fig. 9. Main hardware of fuel management system

B. Software Implementation

The software implementation process is divided into two parts: the software related to the hardware peripherals and software related to the web application.

1) Installing raspberry pi and sensors

The software (PuTTy) was used to communicate the raspberry pi without the need for hardware peripherals like (Screen, Keyboard and mouse). It was done by connecting the raspberry pi to the internet using the Wifi-Dongle and then providing PuTTy with the R-Pi's IP address. The reason behind running the raspberry pi headless is the power that is consumed by the raspberry pi peripherals. By using this method, the raspberry pi's power limitations were reduced in order to have a more efficient and reliable system.

Several dependencies were needed in order to have a fully-functional code. Those dependencies were installed prior to

code writing, because some of these libraries are essential and considered as parts of the written code. The installation required internet connection, and was done using the RPi's LXTerminal which used Linux commands.

- Git, python-dev, pip
  sudo apt-get update sudo apt-get install git-core python-dev python-pipRPi.GPIO
  sudo pip install rpi.gpio
- Python-eeml
  sudo apt-get install libxml2-dev libxslt1-dev python-lxmlgit clone https://github.com/petervizi/python-eeml.git cd python-eeml sudo python setup.py install
- python-picamera
  $ sudo apt-get install python-picamera
- Flask
  $ pip install flask

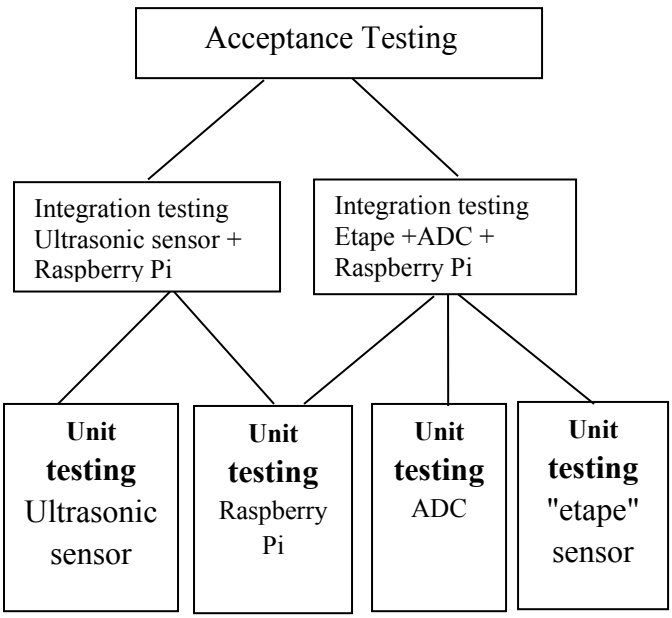
After that code writing started by coding for the main units of the system using Python programming language. Many modifications and enhancements have been made to the codes and the circuit. After a long tuning process, the final codes and circuit for the system were ready.

2) Web application implementation

Several steps have been performed to construct the web application; the work mainly consisted of using ready-made libraries, modifying existing open source projects and writing codes from scratch. Development platform that was used is notepad++. First screen allows signing up to the users. After entering valid password, the page of the information appears, each button in this page corresponds to specific data such as table of data, consumption rate, fill date and many other information.

VI. TESTING

Many types of testing were performed on the system during the design process, a unit testing is performed for each component to ensure that each component works very well separately, this type of testing involved testing for each sensor separately. Then an Integration Testing has been done to ensure that the components will work well when they are integrated together to form the system.



A. Unit testing:

1) Ultrasonic and etape Sensor

For testing the Sensors, testing involved two cases:

a) Functional Testing: Test the analog and digital output of sensor with respect to VCC and the distance.

b) Environmental Testing: Test against environment specifications.

B. Integration Test

The distance from sensor to surface has been changed and the voltages at output pins of the Raspberry Pi have been measured.

C. Acceptance Test

Acceptance Test needs to verify all requirements. There are several test cases for acceptance Test: Functionality and Environmental testing. The fuel level system detected the depth of a fuel in the container and the level of fuel has been displayed through the web application.

VII. RESULTS

The implementations were done in a number of steps in order to reduce the system complexity and to simplify the system integration process. Three results were taken during and after the implementation process of the system. Results were taken during the implementation for the unit testing purpose and after the implementation to ensure that the system is working fine (integration testing). Figures below show the typical setup and results for testing.

A. Hardware results

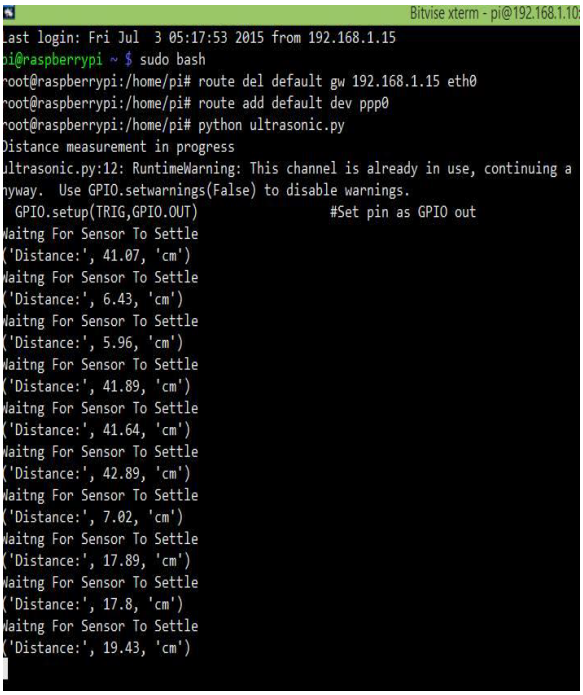


Fig. 10. ultrasonic/etape results on the linux terminal

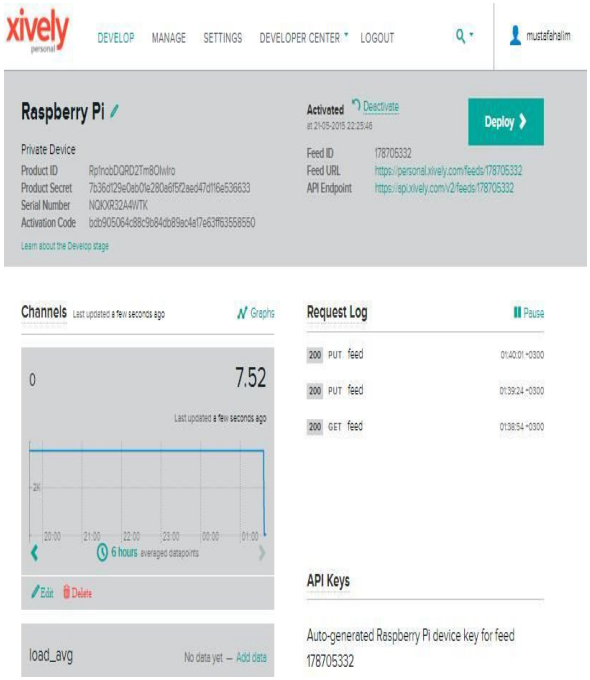


Fig. 11. Xively interface with the raspberry pi

B. *Web application pages*

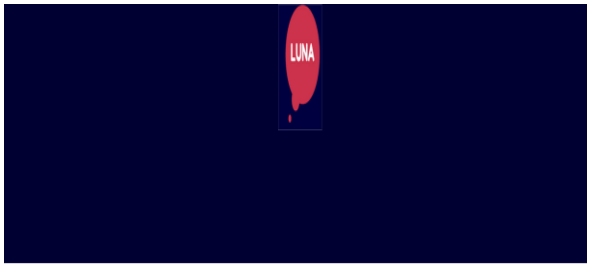


Fig. 12. first page

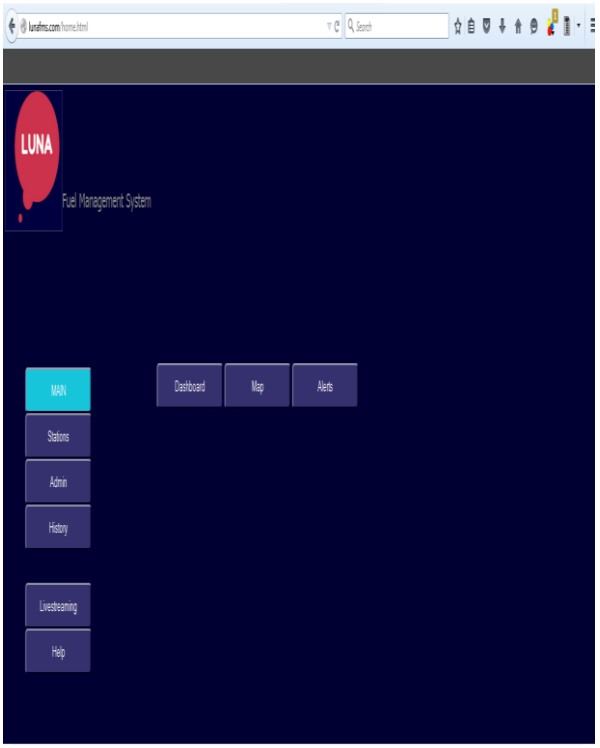


Fig. 13. Main screen

Fig. 14. Website dashboard

VIII. DISCUSSION

When the fuel in the tank was 40 cm the result on the linux terminal was 41.2 using ultrasonic sensor and this was due to gasoline thick vapors issue. Whereas, on the other hand, using etape the 40 cm appears as 40.07 on the linux terminal. The error using ultrasonic equal 3% while using etape equal approximately 0.2%. The result was appearing on the web application with delay equal to arround 1 second and this was due to our internet speed.

IX. CONCLUSION AND FUTURE WORK

This paper developed a fuel management system that measures tank's fuel level to be displayed through web based application and design of camera surveillance system for station. At the same time this management system can store the transaction information in the database that can generate daily, weekly, monthly and yearly business report. This system is more efficient, reliable and cheap compare to existing system. In addition, the system has been successfully designed to operate independently of the power grid, by utilizing solar panels. These panels additionally charge the system's batteries so that it remains powered at night or during cloudy days. The system employs two different sensors: The ultrasonic sensor, and the chemical Etape. Although cheaper than the latter, the results produced by the ultrasonic sensor suffer from inaccuracies caused by the gasoline thick vapors. The chemical Etape, being more expensive, resolves this issue and possesses a higher resolution.

One of the system's limitation is the response delay present in the web application, due to the limited internet speed. This might lead to the loss of real-time data, and thus decreasing the efficiency of the system.

Future work will comprise further enhancing by using more efficient sensor and adding more sensors to measure other environment parameters in the site. Constructing a weather box or (Stevenson Screen) to protect the instruments of the system from weather variations is one of the future plans. Enriching the fuel management system with secure web application to prevent the system from unauthorized access is also open issues to work on.

REFERENCES

[1] P. F. Ovidiu Vermesan, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. 2013.

[2] M. Abu-Elkheir, M. Hayajneh, and N. A. Ali, "Data management for the Internet of Things: Design primitives and solution," Sensors (Switzerland), vol. 13, no. 11, pp. 15582–15612, 2013.

[3] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," Wirel. Pers. Commun., vol. 58, no. 1, pp. 49–69, 2011.

[4] Dr. A.D.Shaligram, _Sensor &Transducer_ C.T.C,135.

[5] Raspberry Pi Foundation. [Online]. http://www.raspberrypi.org/help/what-is-araspberry-pi/

[6] "Raspberry Pi GPIO Connector". [Online]. Elinux.org

[7] J. Fraden, Handbook of Modern Sensors: Physics, Designs, and Applications, 2nd ed. 2010.

[8] O. Bello and S. Zeadally, "Intelligent Device-to-Device Communication in the Internet of Things," Ieeexplore.Ieee.Org, pp. 1–11, 2015.

[9] D. Rountree and I. Castrillo, "The Basics of Cloud Computing," Basics Cloud Comput., pp. 123–149, 2014.

[10] Carretero, J. & García, J. D. The Internet of Things: connecting the world. Personal Ubiquitous Computing (2013).

[11] SS Aher, RD Kokate, Fuel monitoring and vehicle tracking. Int. J. Eng. Innovat. Technol. 1(3), 166–169 (2012).

[12] GM Hemnandan, G Gajanan, R Anil, Remote monitoring of fuel level for diesel generator set, in National Conference on Electronic Technologies (Ponda-Goa, India, 2011), pp. 1–3