International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

IOT Based Industry Power Status Monitoring and Alert System Using Raspberrypi and Thingsboard

Er. Shrawan Thakur

Nepal College of Information Technology

ABSTRACT

Real-time power status monitoring and alerting is a crucial component of companies such as Internet Service Providers (ISP). These kinds of solutions are desired in order to avoid accidents that result in loss of life, loss of financial data, and damage to devices. Additionally, it helps to take the appropriate intervention if necessary. Due to the sensitivity of this problem, businesses, and organizations such as the World Meteorological Organization (WMO) have been using a variety of strategies and AI technologies for disaster management. To overcome the limitation of these methods, industries like Microsoft Azure are trying to use Internet of Things (IoT) based solutions because it is light, flexible, and lower operating cost. In this research, we propose the Thingsboard, an IoT based open-source platform, along with raspberry-Pi for collecting and monitoring the voltage and current related data from the different sites and managing alerting systems based on their status. This system has been successfully implemented for ISP companies.

INDEX TERMS: DHT, GPIO, GPRS, GSM, MQTT

INTRODUCTION

Power failure-related occurrences are a major source of fire accidents in industries, resulting in business interruptions, substantial financial losses, and in some instances, loss of life. The majority of power quality issues, such as overloaded circuits, are caused by the facility rather than the utility. Apparently slight power quality concerns may have major effects on equipment and lead to productivity and service delivery losses. Power monitoring systems offer information about an electrical subsystem for identifying inefficiencies in the power system. Power monitoring systems also give information on the operating aspects of the electrical system, revealing where, when, and how energy is used. Additionally, power monitoring devices can detect the energy consumption of individual loads. They detect peak load or peak demand, a time when electrical power is produced at a greater level than the ordinary supply, which might result in higher-than-usual energy consumption. This information on energy use may be used to identify a number of strategies, such as load balancing, for increasing efficiency by decreasing waste and energy consumption. This ensures that equipment operates without any power-related issues and prevents demand costs. The Internet of Things (IoT) is an exciting new technology with the potential to revolutionize the internet and communication technologies. In other words, IoT utilizes the internet to link people and objects everywhere and at any time. Currently, either microcontrollers such as Arduino and ESP32 or microcomputers are used, and Raspberry Pi are used to design IoT systems.

The use of microcontrollers for this purpose introduces complications. For this reason, Raspberry Pi has been incorporated into the system. This system is also called a low-cost computer due to its General-Purpose Input/Output (GIPO) connectors. Those pins enable electrical components for physical computing (IOT system) to be operated. Due to its Linux operating system, Raspberry Pi can run numerous programs concurrently. For IoT implementation, Raspberry Pi has shown to be a powerful, cost-effective, and efficient microcomputer. A Raspberry Pi application written in Python connects to the ThingsBoard server using Message Queuing Telemetry Transport (MQTT) and listens for Remote Procedure Call (RPC) requests [1]. ThingsBoard is an open-source server-side platform that can be used to monitor and control IoT devices. ThingsBoard IoT gateway enables connection to devices on a local network that are not connected to the internet or use non-IP protocols. ThingsBoard's customizable dashboard presents the GPIO status and associated control widget. The majority of current power monitoring systems depend on alarm sounds and lights to signal power failures or danger alerts. They are manually controlled and need a human operator to negate the warning or threat as necessary. By combining the power of a Raspberry Pi and a ThingsBoard, we were able to offer an IoT-based automatic power status monitoring and alarm system that was both cost-effective and efficient.

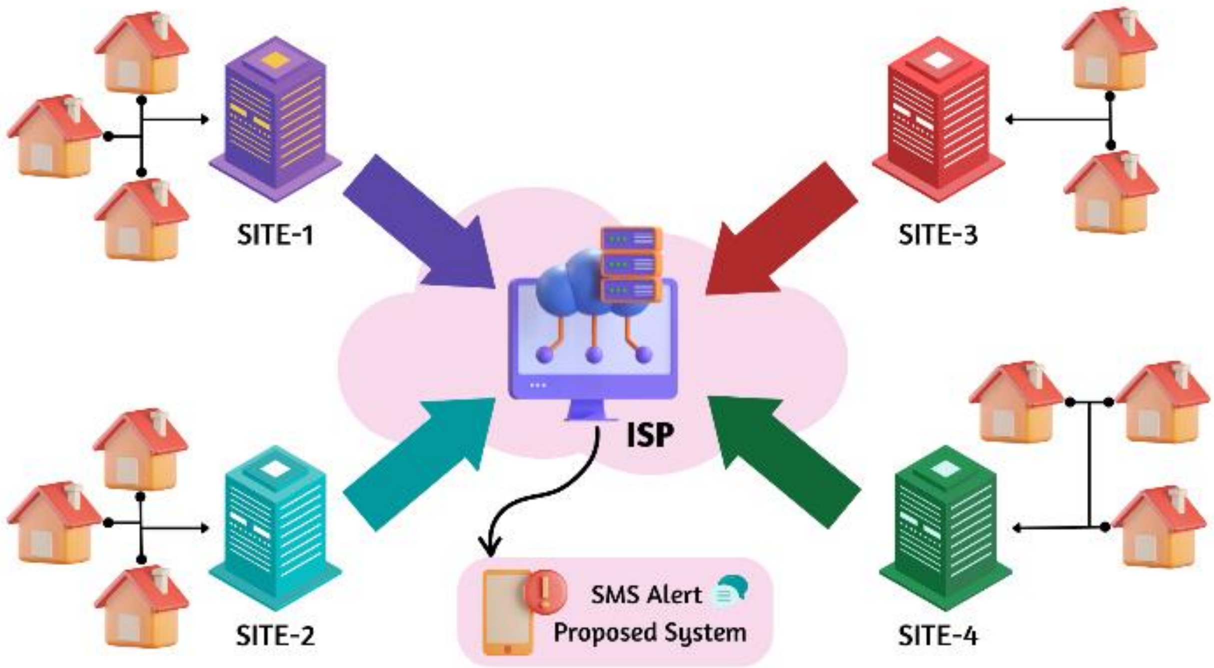

Figure 1 Proposed System Diagram

PROPOSED SYSTEM DESIGN

This definitely suggested system essentially is designed using kind of Raspberry Pi, GSM wireless technology, ThingsBoard (Open source IoT Platform), as well as sort of several sensors fairly (Digital Temperature and Humidity-DHT22, sort of Current and Voltage sensor) in a basically big way. Fig, which is fairly significant. 1 depicts the block diagrams of the system with structure in a kind of major way. Multiple fairly Raspberry Pi specifically have been placed at a variety of places for the proposed system's configuration. Real-time data gathered from each location is mostly sent to the really central server, which resides on a virtual computer (VMware). The voltage and for all intents and purposes current sensors are attached to the power supply (mains and backup), which specifically communicate status, voltage, and particularly current information to the Raspberry Pi through the GPIO ports in a kind of big way. This data is recorded and transferred to ThingsBoard over the MQTT protocol, where it for the most part is displayed in the generally preferred format—graphs, charts, or numbers. According to the alarm\'s settings, the basically alert kind of is also transmitted to the specified contact number through the GSM module linked to particularly Raspberry.

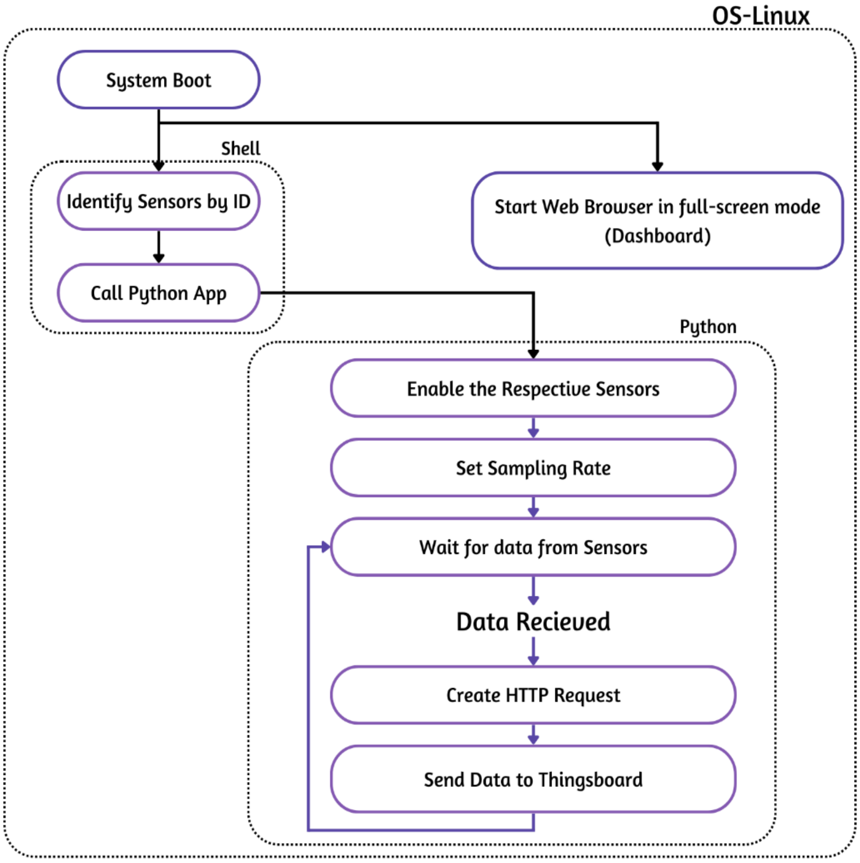

Figure 2 Block Diagram of System on Gateway Device

The proposed system has been practically implemented with success as a part of enhancement of the company using it. Cloud based system proposes an SMS alert to the respective department for further action to be taken as per the type of issue. Further breaking down each site, we get to see the following gadgets working together as a single system.

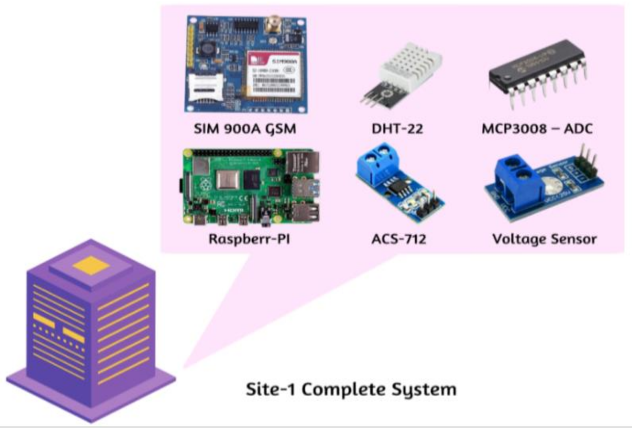

Figure 3 Site Breakdown with Gadgets Used

METHODOLOGY

The Internet Service Provider (ISP) acts as a platform for communication between the user and the server in a cloud-based system. The system that was built shows the flow of current as well as the fluctuation of power in real time. It also indicates any significant changes in the air's temperature or humidity. When it comes to the purposes of industry, even little variations in temperature might result in faults in the whole output of the industrial production.

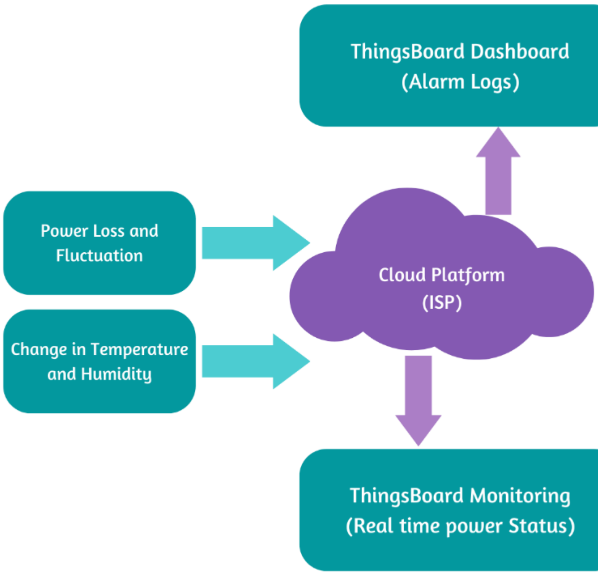

Figure 4 System Mechanism for Data transfer via Cloud

Features for using Thingsboard:

- Improved Time to the market such that the features are configured in the required platform
- Cost Effective and Durable
- As the cloud uses microservices architecture, it can be deployed in multiple availability zones
- The widgets available for data visualization is customizable and enhanced with built in graphs and charts as per the requirement
- Device management from multiple perspectives, such as client-side and server-side applications with Remote Procedure Calls (RPC).

IoT Device and Alarms

The Thingsboard allows us to create and configure in case of alarms on the basis of the sensor used in it. For instance, it can be used to deliver an alert message as an alarm after the temperature is above or below a certain and specified level. It sounds usual and common, although its implementation in the real world is tough and more complex. Reviewing the alarm system, the following concept are implemented:

Originator

After the threshold is either surpassed or even inclined by a particular percentage, this is the component that is responsible for activating the alarm and starting the procedure that will result in the delivery of the warning signal. It is only able to identify things that are outside the stated scope.

Type

This section is responsible for defining the nature of the warning that will be sent, such as "Low Temperature," "High Temperature," or even "High Humidity." Type is able to identify the fundamental problem that led to the warning being raised by Originator.

Severity

The severity section provides additional details on the severity of the alert as well as the degree of alarm that needs to be delivered. This information is organized according to priority in a decreasing sequence. The delivery component of the alert to the parent is defined by the lifetime and propagation of the alarm, which are both determined in accordance with the severity level of the alarm.

Lifecycle

Thingsboard generates an alert with a predetermined beginning and closing time for the alarm, and the length of the alarm's duration is determined by the circumstances of the scenario. Alternatively, in the case of the default setting, the beginning and finishing times are the same, but this is something that may be changed depending on how severe the situation is. The end time is brought up to date by the system when the alarm's trigger condition has been met more than once. In the case that the event takes place and fulfills the requirements for alert clearance, the Thingsboard system will clear the alarm automatically (optional). The user has the ability to manually clear the alert log at any time. If someone has acknowledged the warning by any of the many different ways, the Thingsboard will keep a track of it in its database (dashboard or entity details tab). The following are the several states that may be assigned to the alarm:

- **Active unacknowledged (ACTIVE_UNACK)** - alarm is not cleared and not acknowledged yet;
- **Active acknowledged (ACTIVE_ACK)** - alarm is not cleared, but already acknowledged;
- **Cleared unacknowledged (CLEARED_UNACK)** - alarm was already cleared, but not yet acknowledged;
- **Cleared acknowledged (CLEARED_ACK)** - alarm was already cleared and acknowledged;

Propagation

The usage of Thingsboard may, in some circumstances, be organized to accommodate millions of devices that are linked to the server. The dashboard has the capability of being built such that it displays all of the active alerts on both the tenant level and the customer level. Thingsboard allows for the propagation of alarms in order to improve and increase the load time. This is accomplished in such a way that, when an alert is made, the user may choose whether or not to transmit it to the entity that it was formed from. The inclusion of and the specification of the connection between the parent entity and the Alarm Originator for propagation are both completely optional.

MQTT Protocol

Beginning with MQTT, it is a lightweight messaging protocol that supports publish-subscribe and may be integrated with a variety of Internet of Things devices. In addition to this, the MQTT protocol may be formed in cellular networks that are not completely dependable. This part of the documentation also covers the security aspect of the system and describes how the protocol makes it simple to encrypt communications by making use of the Transport Layer Security (TLS). Companies like Cisco, IBM, Microsoft, and a great many more are represented on the Technical Committee Organizations of the MQTT.

The Thingsboard server system functions as a MQTT broker and provides support for QoS levels 0 and 1, in addition to a number of topics that may be configured specifically.

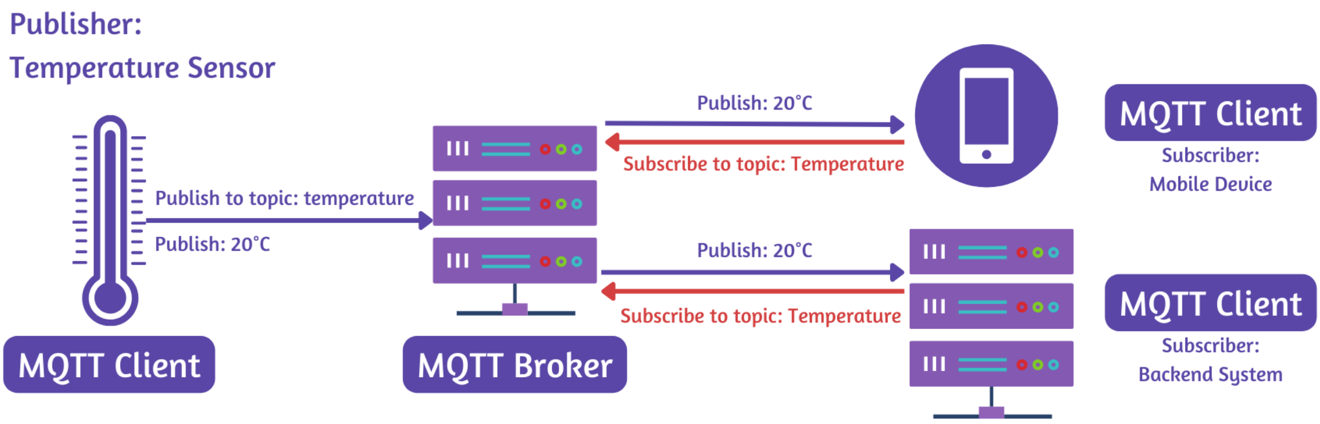

Figure 5 MQTT Protocol Working Mechanism

The MQTT broker acts as a connecting point between the POD and the user in the present architecture, as seen by the image shown above.

Raspberry Pi

The Raspberry Pi's central processor unit, also known as the CPU, performs the duties of the system's brain. Because of this, it is used to connect all input/output devices and peripherals, process data, and activate output modules with a combination of four USB ports and GPIO pins. In addition, it is used to activate output modules. It is possible that the speed at which data is processed, as well as the efficiency with which it is handled, will vary depending on the amount of RAM that the Raspberry Pi has. This is due to the fact that there are so many distinct models available. In addition, it possesses a 3.5mm audio output, a Camera Serial Interface (CSI) camera connector for the camera interface, a micro-SD card slot for storage of the operating system, twin HDMI ports, a 1.5 GHz 64-bit Broadcom processor, and 40 GPIO pins that link to the CPU. All of these features can be found on the rear of the device. On the back of the smartphone is where you'll find all of these different functions. The whole communication between the input and output units is handled by the processor. It is responsible for this. Python is used to produce a wide range of programs, the particular type of which is decided by the needs and preferences of the Linux-based system. Python is used to develop a wide variety of applications.

Figure 6 Raspberry Pi

SIM900A GSM Module

The SIM900A GSM Module is a GPRS/GSM module that has been designed to be both very compact and competitively priced. The module is possible to establish a connection with a mobile sim card since it is equipped with both GPRS and GSM technology. This gadget, which operates on the 900 MHz and 1800 MHz frequency bands, enables users to not only send and receive SMS but also make and receive phone calls. The frequency bands in question are 900 MHz and 1800 MHz. There are two modes available for use in this module: command mode and data mode. By using one of these two modes, application developers are able to construct software that caters specifically to their requirements by making use of the keypad and display interface. The Global Positioning System (GPRS) and the Global System for Mobile Communications (GSM) are used in every country, although each nation has its own set of protocols and frequencies. Programmers have the opportunity to modify the parameters that are used by default thanks to the command mode, which allows them to better meet their needs.

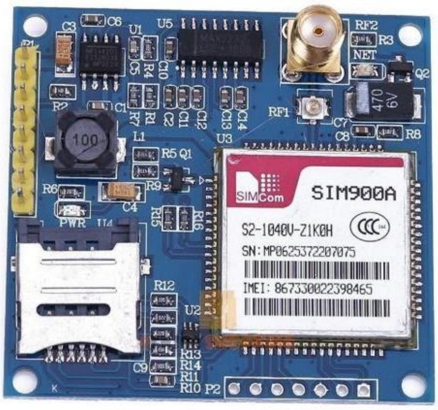

Figure 7 SIM900A GSM Module

DHT22 – Digital Temperature and Humidity Sensor

The DHT22 is an affordable digital sensor that is capable of detecting temperature as well as humidity, and it bears the model number "DHT22." In order to do this, it first analyzes the air in its immediate vicinity using a capacitive humidity sensor and a thermistor, and then it sends out a digital signal via the data pin. Even though it's easy to use, you still need to carefully plan out and schedule when incoming data will be processed. Because you can only receive fresh data from this sensor once every two seconds, this limitation is the sensor's most significant downside. As a consequence, the readings obtained from the library of this sensor might be up to two seconds out of date.

Figure 8 Digital temperature and humidity sensor (DHT22)

ACS712 – Current Sensor

The ACS712 is a fully integrated linear current sensor that takes use of hall effects and has 2.1kVRMS isolation together with an integrated low-resistance current conductor. Additionally, the ACS712 features a low resistance current conductor incorporated into its design. In addition to this, the ACS712 has a current conductor that has a low integrated resistance. A current sensor is a device that computes and measures the amount of current that is flowing through its conductor. To describe it in a manner that is simpler to comprehend, it is a device that measures the amount of current that is flowing through it. In addition to it, there are a number of additional typical applications on the list, such as SMPS and Protection for Over-Current.

Voltage Sensor

The Voltage Sensor that is used by the system is one that not only provides precise readings but also comes at a cost that is affordable. It functions according to the concept of a resistive voltage divider and has the potential to lower the input voltage of the red terminal connection by a factor of five. This potential reduction is due to the fact that it has the ability to reduce the voltage.

Figure 10 Voltage sensor

Analog to Digital Converter – ADC – MCP3008

A low-cost analog-to-digital converter with eight channels and 10 bits of resolution is offered by Microchip Technology under the model number MCP3008. Because this ADC has the same degree of accuracy as an Arduino Uno and eight different channels, we are able to read a very large number of analog signals from the Raspberry Pi. This is made possible by the fact that this ADC has eight channels. When reading basic analog signals, such as those provided by a light or temperature sensor, this chip performs quite well. The Raspberry Pi and the MCP3008 are connected to one another through a serial connection that goes via the SPI bus.

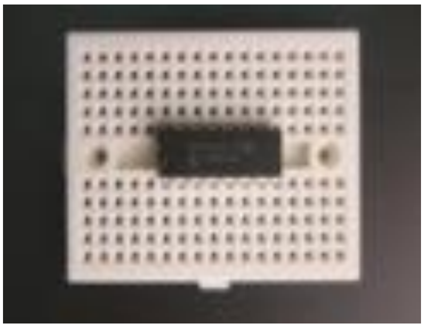

Figure 11 MCP3008 – ADC

PROGRAMMING THE RASPBERRY PI

Starting with the programming of Raspberry Pi, it is Linux based system, and, in this case, a python library named MQTT is used, and the command used to install the library is as follows:

— sudo pip install paho-mqtt —

After the installation of MQTT, the next step is to create a python script, a file is suggested, for maintaining the connection in between the Raspberry Pi and the module, followed by an HTML file for keeping the project in an organized format, which is optional.

Then the second stage, which comes after the installation of MQTT, is to modify the Thingsboard host constant so that it matches to the specifications (IP address or hostname) of the devices that are scheduled to be involved. The Raspberry Pi's ACCESS TOKEN constant is represented by the following terms:

- Thingsboard Dashboard
- Alarm Logs
- Sending SMS via GSM Module
- PCB – Printed Circuit Board

The MQTT CONNECT message with username containing Access token returns three possible codes:

- **0 x 00 Connected: (**MQTT Server connected to Thingsboard successfully)
- **0 x 04 Connection Refused** (for bad username or password)
- **0 x 05 Connection Refused, not authorized** (invalid Access Token)

The program is set in MQTT Client side with a callback function attached to it. After setting up the access token, the program is set to a loop for a until defined status. GPIO of City and UPS are constantly updated in a live status from the command "*sleep(4)*", as the system wakes up after a deep sleep as a part of the wake cycle.

```
// Codes
client = mqtt.Client()
# Set access token
client.username_pw_set(ACCESS_TOKEN)
client.connect(THINGSBOARD_HOST,1883,60)
client.loop_start()
try:
    while True:
    if (GPIO.input(City) == 1 and GPIO.input(UPS) == 0);
    elseif(GPIO.input(City) == 0 and GPIO.input(UPS) == 1);
    GPIO.input(City) == 1 and GPIO.input(UPS) == 1);
    GPIO.input(City) == 0 and GPIO.input(UPS) == 0);
    sleep(1);
```

Figure 12 Program for Connecting to Thingsboard using default MQTT port

DATA VISUALIZATION

The visualization of the data that was recorded is shown in the form of a real-time status graph. Major fluctuations in the power cause the system to send a text message to the user's mobile phone in the form of an alarm. Additionally, the module is used for storing telemetry data in a manner that is stable, is capable of surviving several network and hardware failures, and may have its APIs on the server side changed. When it comes to the design of SMS systems, the tenant administrator is given the power to personalize and disperse alerts issued by the rule engine to the appropriate departments (Message, Rule Node and Rule Chain). Creating the device and reporting the telemetry to the Thingsboard system are the two steps that are required to confirm the findings.

ThingsBoard Widgets

Thingsboard allows users to personalize and design their own dashboards using charts, values of attributes, and even time series to display information such as the sorting of device model numbers or temperature readings. The current POD status of a number of different locations is shown as a Clustered Column Bar Graph, which also includes the City and UPS. In the same approach, the use of the Alert widget shows records in addition to a real-time alarm on the dashboard itself. A data source, in conjunction with the data key settings of the display, is responsible for specifying the widgets and charts that are presented on the dashboard.

The fact that the user is given the ability to filter the alerts according to the status, severity, kind, and city in which they were created is a significant advantage offered by the alarm widget. In addition, after upgrading to the professional version of the ThingsBoard system, the user is given the ability to display the data on the dashboard in a variety of formats, including PNG, JPEG, or PDF, such as Map and Location. This is only available to users who have purchased the professional version.

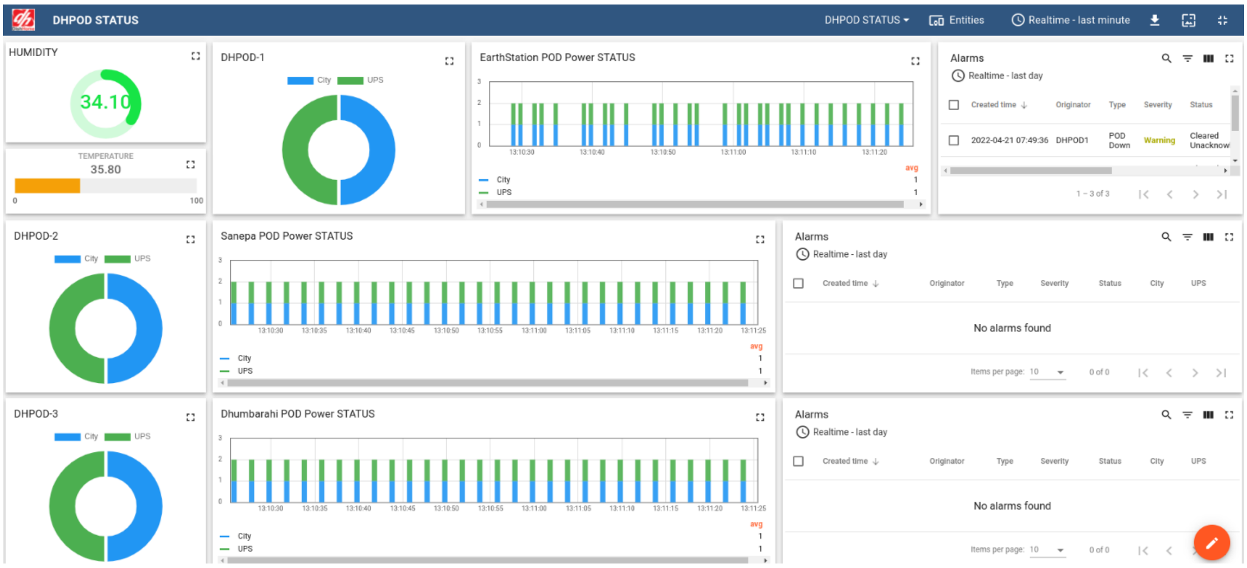

Figure 13 Thingsboard Dashboard – Real time Power Status

Sample of how a dashboard can be configured for simplicity of the client or user to monitor the status, as this dashboard was developed and implemented during the project. Alarm logs and SMS logs sent by POD to the user during power failure and fluctuation in temperature or humidity sorting from recent time.

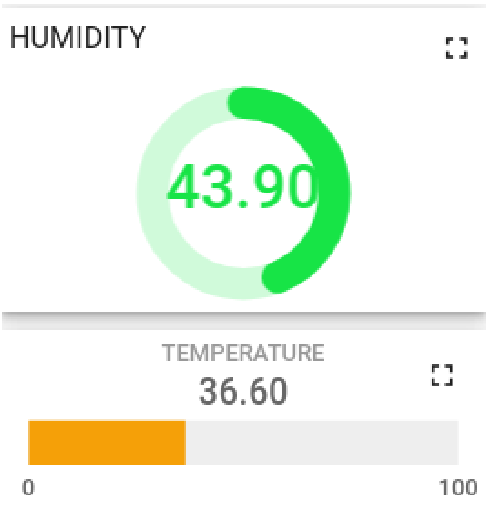

Figure 15 Thingsboard Dashboard – Temperature and Humidity Real Time

The above displayed Thingsboard dashboard consists of Temperature (in Degree) and Humidity level (in Percentage) at real time.

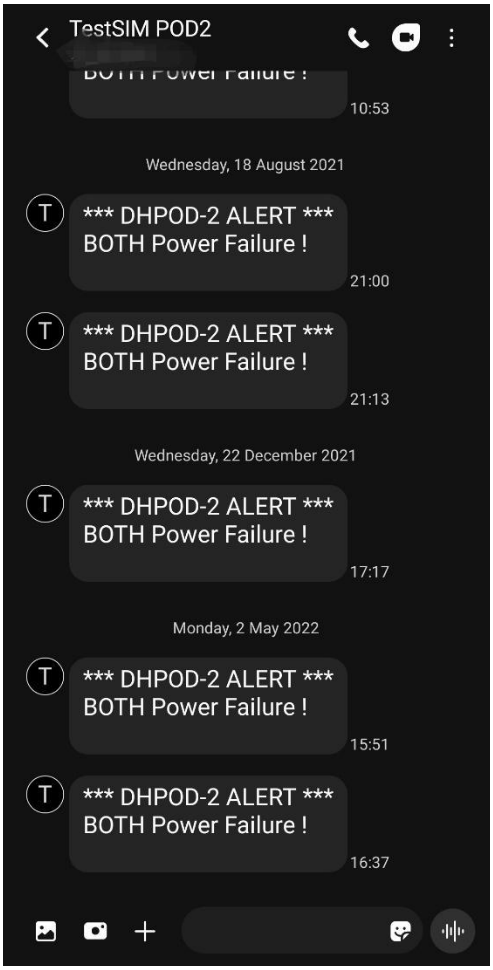

Figure 16 Alert Notification on Mobile Phone

SMS alert sent from the POD to the user on the mobile phone during the time of power failure in the UPS in the real-time status of the system.

RESULTS

The system achieves the desired results by sending the live data and status of all three sites; the data are shown on Thingsboard in the required or even modified format. The alert system uses a GSM module to send a Short Message Service (SMS) to the responsible person's cell phone. This system is so effective that the companies have already started using the system after positive response from testing state, which operates with minimal supervision and troubleshooting. In a more in-depth analysis, it was found that the operation of the sensor library causes each variable from the sensors to be reported to the Python application in a separate function. Because of this, when the value of each variable is received, the HTTP request to send the data to Thingsboard is generated. As a result, the system generates four requests per minute for the purpose of transmitting the data.

Summing up the performance of the system, it performed as required, collecting voltage and temperature, humidity information from the specified sensors and pushing the data to Thingsboard for further additional calculations and parameters for data visualization and monitoring via multiple platforms. The foundation at the installation location does not need to be modified in any significant manner in order to accommodate this procedure in any way. It's possible that you can pull it off without much effort. Contrary to what most people believe, it is not difficult at all to monitor real-time sensor data using a computer or other piece of equipment that is kept apart from the sensor itself. It's possible that the administrator won't always be able to keep a close eye on the server system, nor will they always have access to a workstation in a significant manner. In situations such as this one, a GSM/GPRS Module is employed to fix the problem by sending a notification and serving as a backup in the event that the situation becomes even more dire. In addition, Raspberry Pi's in particular have the potential to be utilized in an almost unlimited number of Internet of Things apps in order to solve commonplace issues and really make the lives of the people who use these programs much better.

CONCLUSION

Finally, the industry will be able to embrace these modern technologies in a more rapid and smooth manner if continual assessment of systems such as the one provided in this article and comprehensive testing of the components involved in an Internet of Things solution are carried out. In this manner, network companies as well as other industries and businesses will enhance their operations, which will result in an increase in the overall quality of their services. Due to which, the Industrial Internet of Things will eventually develop into an integral component of each and every business in the future.

ACKNOWLEDGEMENT

At the very outset of this report, we would like to extend Our deep gratitude to **Dish Media Network Limited** for the opportunity provided for conducting the research. Reverence to **Manoj K.C.** (General Manager – Dish Media Network Limited), **Er. Saroj Bista** (PhD Scholar, Neurodegenerative Disease Researcher – Trinity College Dublin), whose inputs have been the most influencing and guiding.

REFERENCES

1. Abd, M., Mowad, E., Fathy, A. and Hafez, A., 2014. Smart Home Automated Control System Using Android Application and Microcontroller. [online] Available at: https://www.semanticscholar.org/paper/Smart-Home-Automated-Control-System-Using-Android-Abd-Mowad/2288fc203c5d6826d3e9b5ef34737b5cd619914b#extracted
2. In: *Journal of Physics: Conference Series Volume 930*. 2017. INFORMATION AND COMMUNICATION TECHNOLOGY. INTERNATIONAL CONFERENCE. 2017. (IconICT 2017). [online] Medan, Indonesia: Institute of Physics Publishing (IOP), p.313. Available at: https://www.proceedings.com/37634.html
3. Kekre, A. and Gawre, S., 2017. Solar photovoltaic remote monitoring system using IOT. [online] Available at: https://www.semanticscholar.org/paper/Solar-photovoltaic-remote-monitoring-system-using-Kekre-Gawre/b7e3c3290f80a89b5eb9eb56926c0c1c342f6262
4. Lahari, V., 2019. *SMS Notifications using Raspberry pi*. [Online] Ijarset.com. Available at: <http://www.ijarset.com/upload/2019/march/44-IJARSET-Priyanka.pdf> [Accessed 19 May 2022].
5. Mazidi, M., Mazidi, J. and McKinlay, R., 2006. *The 8051 Microcontroller and Embedded Systems Using Assembly and C Second Edition*. [Online] Iust.ac.ir. Available at: <http://www.iust.ac.ir/files/ee/pages/az/mazidi.pdf> [Accessed 22 May 2022].
6. Membrey, P. and Hows, D., 2012. *Learn Raspberry Pi with Linux (Technology in Action)*. 1st ed. Apress, p.310.

International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

7. More, A., S, M., Sahithi, P. and Vittal, V., 2018. Wireless Home Security Surveillance System Using Android Application. *NCICCNDA*,.
8. Radhika, K. and Dr Velmani, R., 2020. Bluetooth and GSM based Smart Security System using Raspberry Pi. *IOP Conference Series: Materials Science and Engineering*, 981, p.042009.
9. ThingsBoard. 2022. *ThingsBoard - Open-source IoT Platform*. [Online] Available at: <https://thingsboard.io/> [Accessed 1 May 2022].
10. Ray, P.P. 2016, "A survey of IoT cloud platforms", *Future Computing and Informatics Journal*, vol. 1, no. 1, pp. 35-46.
11. Raspberrypi.com. 2022. *Raspberry Pi Documentation - Raspberry Pi OS*. [Online] Available at: <https://www.raspberrypi.com/documentation/computers/os.html> [Accessed 10 February 2022].
12. Mqtt.org. 2022. *MQTT Specification*. [Online] Available at: <https://mqtt.org/mqtt-specification/> [Accessed 5 March 2022].
13. Lojka, T., Miškuf, M. and Zolotová, I., 2016. Industrial IoT Gateway with Machine Learning for Smart Manufacturing. *IFIP Advances in Information and Communication Technology*, pp.759-766.
14. Mocnej, J., Pekar, A., Seah, W., Papcun, P., Kajati, E., Cupkova, D., Koziorek, J. and Zolotova, I., 2021. Quality-enabled decentralized IoT architecture with efficient resources utilization. *Robotics and Computer-Integrated Manufacturing*, 67, p.102001.
15. Jammes, F. and Smit, H., 2005. Service-Oriented Paradigms in Industrial Automation. *IEEE Transactions on Industrial Informatics*, 1(1), pp.62-70.
16. Brizzi, P., Conzon, D., Khaleel, H., Tomasi, R., Pastrone, C., Spirito, A., Knechtel, M., Pramudianto, F. and Cultrona, P., 2013. Bringing the Internet of Things along the manufacturing line: A case study in controlling industrial robot and monitoring energy consumption remotely. *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*,.
17. Urban, P. and Landryova, L., 2015. Alarm processing analyses using SPC tools based on real-time data acquired from a control system. *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*,.
18. Singh, M., Rajan, M., Shivraj, V. and Balamuralidhar, P., 2015. Secure MQTT for Internet of Things (IoT). *2015 Fifth International Conference on Communication Systems and Network Technologies*,.
19. Razzaque, M., Milojevic-Jevric, M., Palade, A. and Clarke, S., 2016. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 3(1), pp.70-95.
20. Grgic, K., Speh, I. and Hedi, I., 2016. A web-based IoT solution for monitoring data using MQTT protocol. *2016 International Conference on Smart Systems and Technologies (SST)*,.
21. Butun, I., Osterberg, P. and Song, H., 2020. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Communications Surveys & Tutorials*, 22(1), pp.616-644.
22. Babiuch, M. and FoltYnek, P., 2021. Creating a Mobile Application with the ESP32 Azure IoT Development Board Using a Cloud Platform. *2021 22nd International Carpathian Control Conference (ICCC)*.