IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CLOUD BASED SERVER ROOM MONITORING WITH ARDUINO AND MICROSOFT AZURE

[1]Nitin Kawde, [2]Mustafa Saifee, [3]Kalyani Pawar, 4Mangesh Panjabi, 5Bhushan Musale
[1]Asst. Professor, [2]Student, [3]Student, [4]Student, [5]Student

[1]Depertment of Information Technology,
[1]Shri Vile Parle Kelavani Mandal's Institute of Technology, Dhule, Maharashtra, India

Abstract: The project "Cloud based Server Room monitoring with Arduino and Microsoft Azure" is aimed at reducing the cost of the setup of the existing system. Reduce capital expenditure by leveraging Microsoft Azure. Increase portability by using the AZ3166 development kit. The components involved in this project are of very low power consumption; hence the Development Kit would require no separate specialized power source to run this. Also, the board can have multiple sensors in multiple rooms and monitor them on a single dashboard. This project's scope is within a single server room or a server rack, depending on the requirement.

I. INTRODUCTION

The server is the most critical component in a company's efficient working. Many physical properties require to be maintained if you wish to have a smooth functioning system in the organization. The three most critical properties that can affect the server physically are Temperature, Humidity, Pressure. The first two are meant for the machines, while the third one is for the operators and technicians. The hardware required to monitor this stat is costly and not feasible for start-ups and new businesses.

An innovative hardware solution is provided in the development kit. It is compatible with Arduino with peripherals and sensors. AZ3166 is used for the development of IoT. The device connects to mobile and Azure safely and fast. The period of research is reduced significantly for the customers. AZ3166 is a developer community supported with software and hardware, including a demo for quick connection to the cloud service development kit. You can control and operate by mobile, phone, or tablet. The main control unit AZ3166 is EMW3166 is low power consumption.

II. LITERATURE SURVEY

Title: Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT) [1]
Author: Moechammad Alvan Prastoyo Utomo, Abdul Aziz, Winarno, Bambang Harjito
Year of Publication: August 2019
Technology Used: Using raspberry PI and arduino assisted with a DHT22 sensor
Limitations: It uses telegram for receiving the stats
Title: Temperature and Humidity Monitoring System in Server Room Using Raspberry Pi [2]
Author: Rico Wijaya, Vincent Christian, Yustika Shofiani Yuwananda, Ivan Alexander
Year of Publication: October 2019
Technology Used: Use of a database server to record the data and sensors to record the stats using DHT11 sensor
Limitations: A user interface needs to be hosted and designed on a different server. Limited to just temperature and humidity
Title: Monitoring of Temperature in Retail Refrigerated Cabinets Applying IoT Over Open-Source Hardware and Software [3]
Author: José Ramírez-Faz, Luis Manuel Fernández-Ahumada , Elvira Fernández-Ahumada and Rafael López-Luque
Year of Publication: February 2020
Technology Used: Customized circuit board. Zigbee for connectivity. Cellular. Bluetooth and Wi-Fi modules
Limitations: A lot of customization comes at a hefty cost. Regulated power supply required.

III. METHODOLOGY

3.1 Setting up the cloud platform(Microsoft Azure)

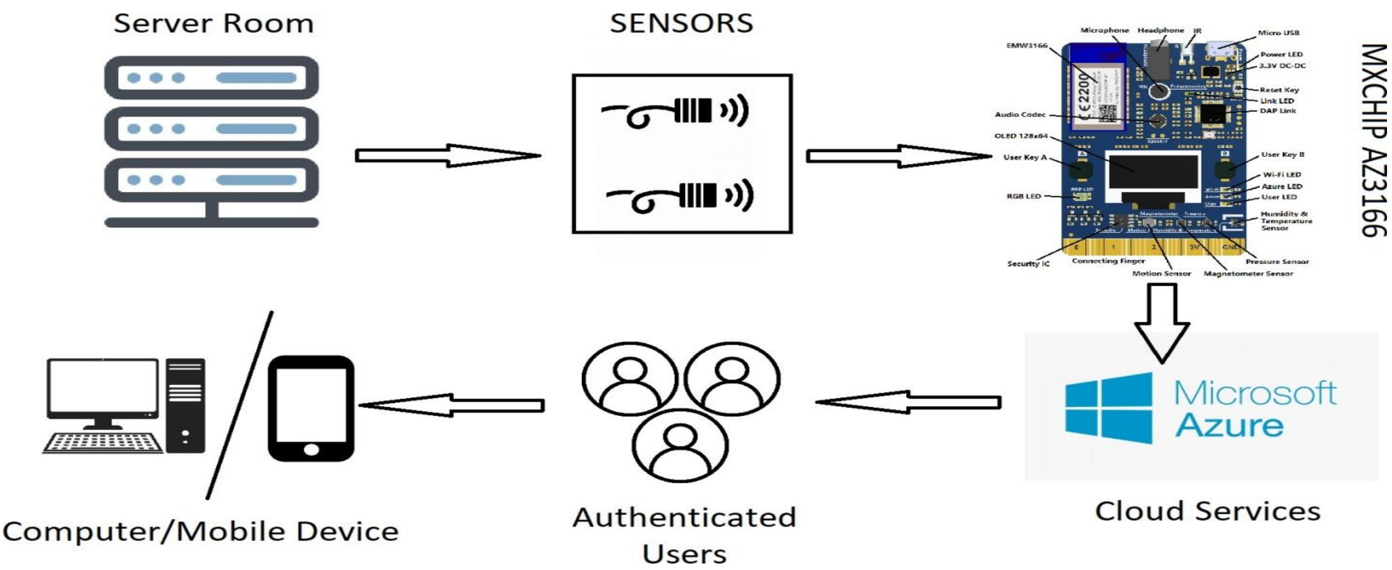
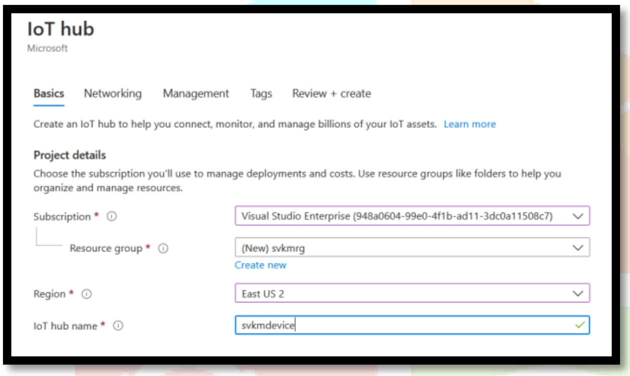
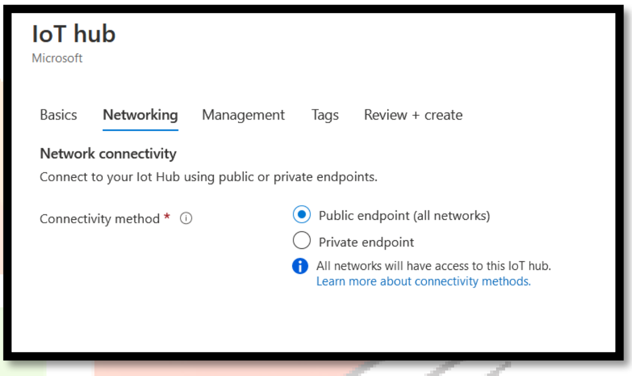
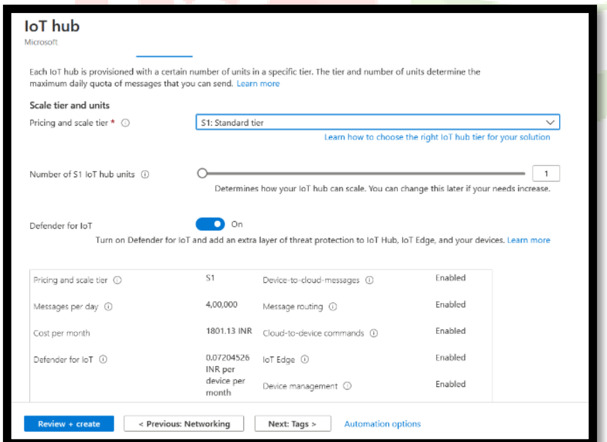


Fig 3.1

Sign in to the Azure portal(portal.azure.com). From the homepage, select the Create a resource button, enter IoT Hub in the search. You have to choose IoT Hub from the results and then select Create. Fill the respective fields as follows:







Select **Review + create** so that you can review your choices. Select **Create** to make your newly created hub. Forming the hub can take a few minutes.

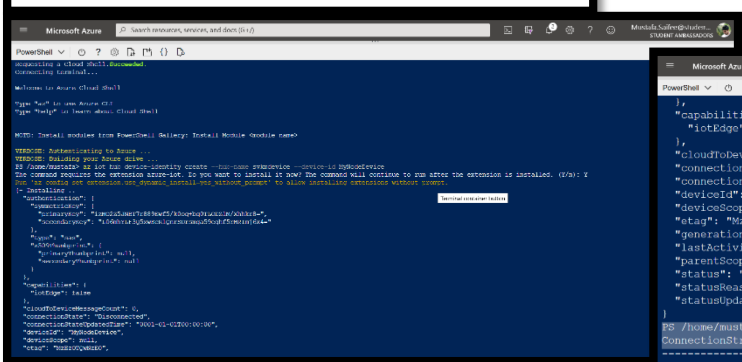
3.2 Register the device

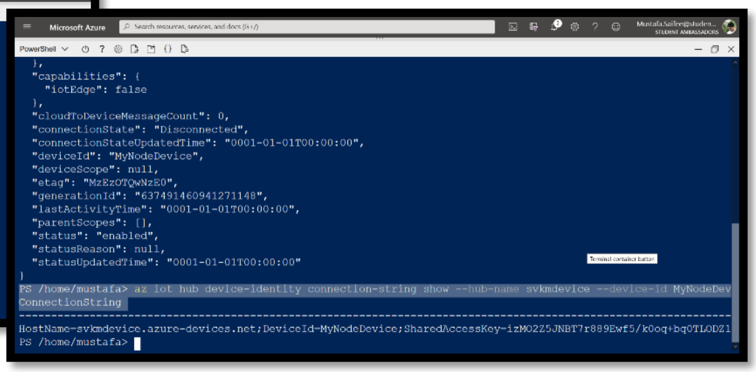
You have to register a device with your Azure IoT hub before it can connect. For creating the device identity, run the following command in Azure Cloud Shell.

az iot hub device-identity create --hub-name NameYourIoTHubName --device-id NameMyNodeDevice

NameYourIoTHubName: You have to replace this variable with the IoT hub name you chose.

NameMyNodeDevice: Give a sample name for the device here





You have to get the connection string of the device you registered, so you have to run the following command:

az iot hub device-identity connection-string show --hub-name NameYourIoTHubName --device-id NameMyNodeDevice --output table

You have to note the device connection string, which looks like:
HostName={NameYourIoTHubName}.azure-devices.net;DeviceId=NameMyNodeDevice;
SharedAccessKey={YourSharedAccessKey}

IV. RESULTS

4.1 Send DevKit telemetry

The DevKit connects to the endpoint on your Azure IoT hub and sends temperature and humidity telemetry data.

4.1.1 Download this file: *https://drive.google.com/file/d/1gGGb37plEgCFIBbkcfMVuKW Pmmq-zPiX/view?usp=drivesdk*

4.1.2 You have to connect the AZ IoT DevKit to your computer via USB. Open File Explorer, where you will find a USB mass storage device called AZ3166.

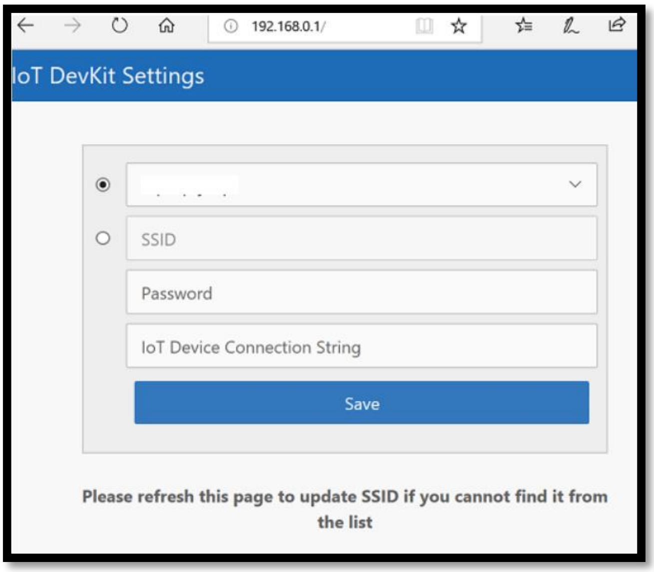
4.1.3 Open Windows Explorer

4.1.4 Drag and drop the firmware just downloaded into the mass storage device, and it will flash automatically.

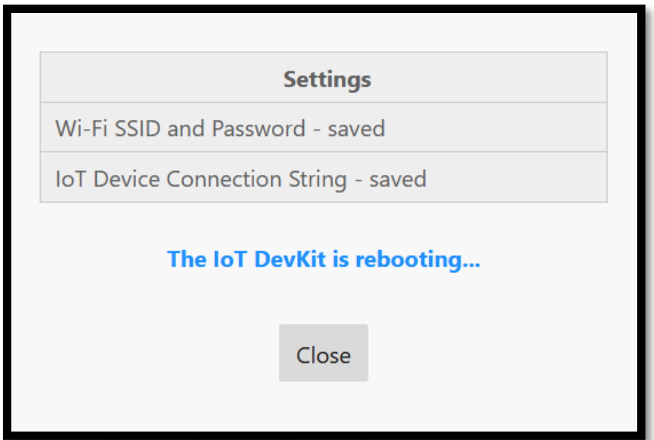
4.1.5 There is a button B on the DevKit; hold down that button; meanwhile, push and release the Reset button. After that, you can release button B. Doing this will make your Kit enter AP mode. The screen will display the service set identifier(SSID) of the Kit with the IP Address of the configuration portal.

4.1.6 On a different Wi-Fi-enabled device (computer or mobile phone), use a web browser to connect to the IoT DevKit.

4.1.7 Enter the IP 192.168.0.1 in the web browser. Find the Wi-Fi and select the one you want the Kit to connect to; provide the password and the device connection string that we had recorded previously, followed by saving it.

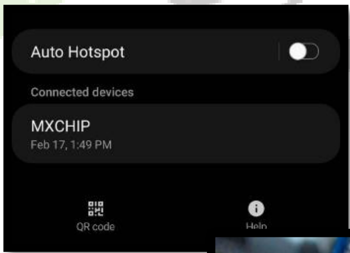


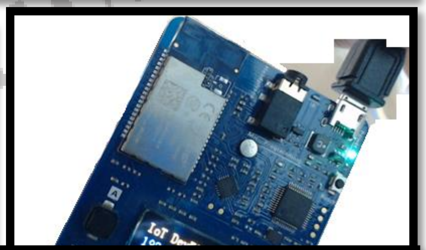
The Development Kit will save the Wi-Fi information and the device connection string when you see the result page.



Now you can see in your connected devices that the development kit is connected to the internet.

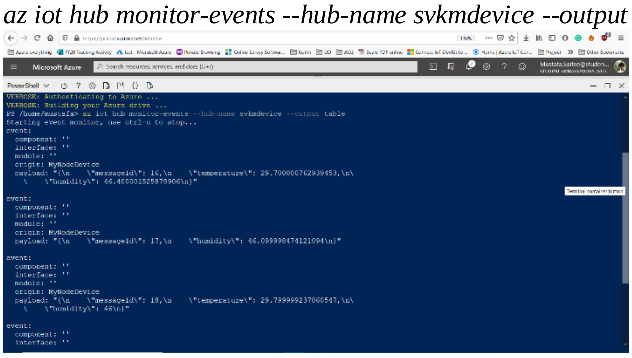
After Wi-Fi is configured, even if the device is unplugged, the device will retain your credentials.

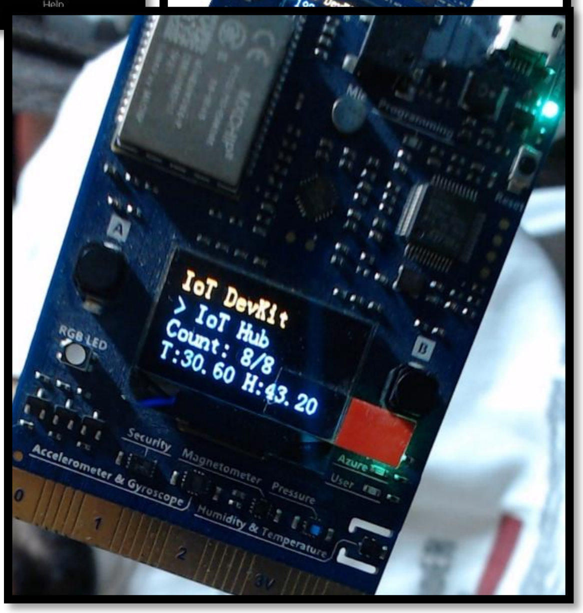




The Development Kit will reboot in a few seconds. On the Development Kit screen, you will find the IP address for the Development Kit, which is followed by the temperature and Humidity telemetry data value with a count of messages sent to the Azure Hub.

You have to run the following command in the Cloud Shell to verify the data sent to Azure:
az iot hub monitor-events --hub-name svkmdevice --output table





V. Conclusion

This project is not limited for Server Room monitoring but has endless possibilities and it can be used for the following purposes as well:

Cold Storages, Storage Trucks, Transport Vehicles, Locomotives, Boiler Rooms, Atomic and Nuclear Centers, Drone Technology, Speech Recognition Applications, Home Automation, Speech Translator

The scope of this project is to make the monitoring facility cost efficient, secure, independent, easily accessible and compact.

VI. References

[1] Moechammad Alvan Prastoyo Utomo, Abdul Aziz, Winarno, Bambang Harjito, "Server Room Temperature & Humidity Monitoring Based on Internet of Thing (IoT)", IOP Conf. Series: Journal of Physics: Conf. Series 1306 (2019) 012030, doi:10.1088/1742-6596/1306/1/012030, ISSN 1742-6596

[2] Rico Wijaya, Vincent Christian, Yustika Shofiani Yuwananda, Ivan Alexander, "Temperature and Humidity Monitoring System in Server Room Using Raspberry Pi", INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 10, OCTOBER 2019, ISSN 2277-8616

[3] José Ramírez-Faz, Luis Manuel Fernández-Ahumada , Elvira Fernández-Ahumada and Rafael López-Luque, "Monitoring of Temperature in Retail Refrigerated Cabinets Applying IoT Over Open-Source Hardware and Software", Licensee MDPI, Basel, Sensors 2020, 20, 846; doi:10.3390, 5 February 2020, ISSN 2003-0846