Global Scientific JOURNALS

GSJ: Volume 10, Issue 8, August 2022, Online: ISSN 2320-9186

www.globalscientificjournal.com

Design and Implementation of an IoT-Based Solar-Powered Inverter Control System.

A. I. Edeoghon, J. Chibuzor Okocha, J. S. Ebu, R. E. Ohiocheoya, O. V. Chukwunweike, G. Ebikade, M. Efemena, E. O. Akinlade, Ekwunife Onyeka, Odey Nicholas, Enofe. P

Emails: okochachibu242@gmail.com tochinicholas@gmail.com, ekwunifechris7@gmail.com ebujakes@gmail.com

Department of Computer Engineering, Faculty of Engineering, University of Benin, 1154, P.M.B, Ugbowo Lagos Road, Benin City, Nigeria.

Keywords: Internet of things, Mobile app, Energy Consumption, Greenhouse, NodeMcu, Solar power

Abstract

In this project, an intelligent IoT-based solar inverter was designed and implemented using the Node microcontroller unit (NodeMcu). The NodeMcu (Node Microcontroller Unit) is an open-source software and hardware development environment built around a low-cost System-on-a-Chip (SOC) called the ESP8266. The Internet of things (IoT) describes exactly the network of physical objects - "things" that are embedded with sensor components, software, and other technologies to connect and exchange data with other devices and systems over the Internet. In addition, these physical objects are provided with unique identifiers (UIDs) and the ability to transfer data gathered over a network without requiring human-to-human or human-to-computer interaction. [1]

The materials and methodology employed in this project include a solar-powered inverter unit. This unit consists of chain connections. First, a solar panel is connected to a charge controller. The primary function of the charge controller is to protect the battery bank from overcharging. This is done by monitoring the battery bank- when the bank is fully charged, the charge controller sends energy from the filled battery bank to a (diversion) load. Next, the battery stores up charges and is connected to the inverter. The function of the inverter is to translate the Direct Current (DC) power supply from the battery to an Alternating Current (AC) power supply- the usable form for home or office appliances. Next, the inverter is connected to the energy meter. The energy meter measures the load of the inverter. The energy meter used is the PZEM-004T V3 to measure the voltage, current, power, energy, frequency Power factor (frequency and PF is extra added in the new version) using a microcontroller unit. The Node microcontroller unit (NodeMcu) consists of a 32-bit controller and an ESP8266 Wi-Fi module, enabling IoT incorporation by hooking the system to the Internet over a Wi-Fi connection. The data, which includes the value of current, load on each outlet, and the battery level of the inverter system, can

be accessed via a designed mobile application interface. It can then be monitored and controlled remotely. [2]

1. Introduction

The Internet of Things (IoT) is not a new concept. IoT refers to a network of physical objects capable of gathering and sharing electronic information. It includes a wide variety of "smart" devices, from industrial machines that transmit data about the production process to sensors that track information about the human body. These sensors can use various types of local area connections such as Radio-Frequency Identification (RFID), Near-Field Communication (NFC), Wi-Fi, Bluetooth, and ZigBee [3]. Sensors can also have wide area connectivity such as Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), 4G, and Long-Term Evolution (LTE). Potential Internet of Things application areas include, Smart Cities (and regions), Smart Car and mobility, Smart Home and assisted living, Smart Industries, Public safety, Energy and environmental protection, Agriculture and Tourism as part of a future IoT Ecosystem have acquired high attention. Similarly, in recent times, there has been a continuous increase in energy consumption which has also led to the increase of renewable energy production. There are different sources of energy currently in use but unfortunately, most of the energy sources come from conventional fossil fuels. The environmental impacts of fossil fuels such as oil, coal and gas are very enormous and hazardous. Apart from contaminating the air, polluting and harming the environment, it has been identified as the main culprit in increasing greenhouse gases (GHGs) in the atmosphere, which is causing global climate change. One of the most immense methods of generating electricity without emissions or noise is through PV solar electricity by converting abundant sunlight to electrical energy. Solar photovoltaic (PV) systems are used for several applications since the maintenance required is low and no pollution discharged. However, the intermittency and variability nature of renewable energy sources may result in power system instability if intelligent interface is not provided. A power electronics inverter system should have a digital design, robust software facilities and a two-way communications ability in order to make the system intelligent. [4] The system typically comprises a reliable, robust and proficient silicon-based hardware, which can be controlled by an adaptable software environment by integrating a control structure capable of advanced performance monitoring. The intelligent or smart interface can be developed by employing Power electronics technology. Therefore, in order to effectively achieve reliability with the power output, a power electronics interface system such as a smart inverter system is required. So, with the help of IoT, the inverter can be monitored and controlled with the help of a mobile application. The remote controlling of the solar inverter helps to prevent overloading thereby increasing the life expectancy of the solar inverter. This also helps to prevent overheating of the capacitors because capacitors are extremely temperature sensitive. And also using inverters beyond their operating limit, either by choice or due to oversight or lack of knowledge, can contribute to inverter bridge failure. Using any component at a rating higher than its operating limit will decrease its lifespan and lead to failure, so avoiding this issue simply comes down to checking that all inverters are being run correctly. Over-current and Over-voltage can cause inverter failure. If either current or voltage increases to a level that the inverter is not rated for, it can cause damage to components in the device, most frequently the inverter bridge. Often this damage will be caused by the excess heat generated by the spike in voltage or current. So, this can be avoided by remote-controlling the inverter when the inverter has reached its maximum load.

2. Methodology

In this chapter, the solution and approach as earlier proposed is presented. This includes the flow chart which represents the stages of the solution and how the interfacing with each component for the project was realized. Also, the outcome of interfacing of components to realize a functional hardware system. To achieve these sets, the below schematic block diagram was developed

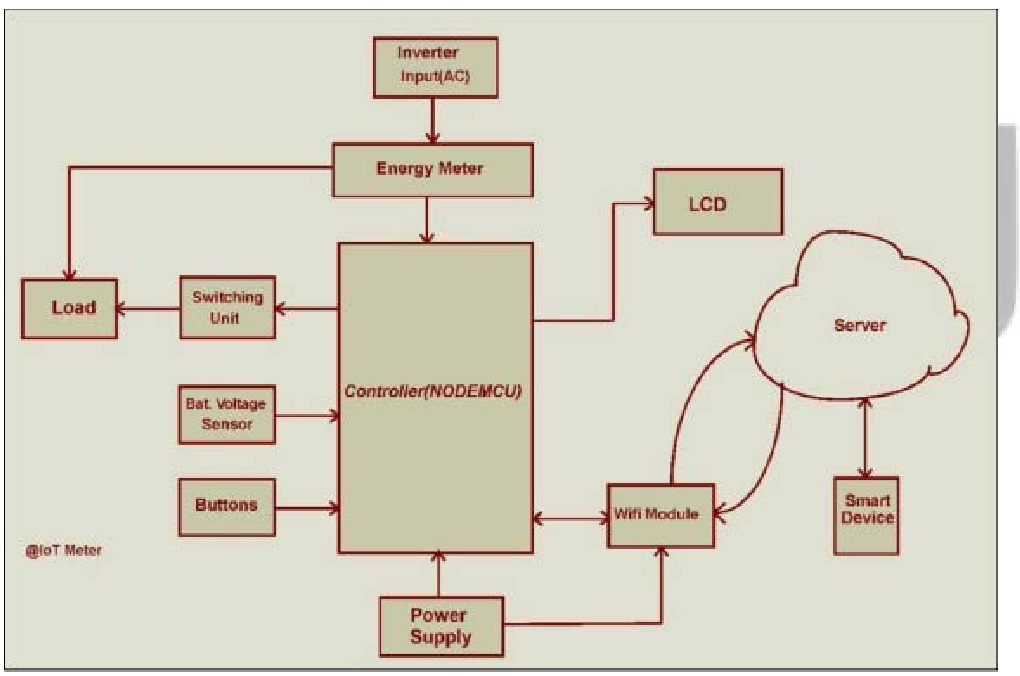

Figure 1: Block Diagram of the System

2.1 Overview of Overall System Implementation.

The solar-powered inverter system is a system that comprises an inverter, battery, solar panel, and a charge controller. The metering included in this system is to address the area of monitoring for the billing system that will be realized. The solar panel works to convert solar energy coming from the sun into electrical energy which is needed to charge the battery. Power generation from Solar Photovoltaic plants is variable due to changes in solar irradiance, temperature, and other factors. Thus, remote monitoring is essential. The Charge controller is connected between the solar panel and the battery to regulate the power coming from a solar panel with the right voltage and current before going into the battery. It regulates the fluctuating output of the solar panel at any point be it at the sun's high intensity or low. To ensure that batteries do not overcharge during the day and that power does not run back to the solar panel overnight and drain the batteries. The DC output from the battery will be sent into the inverter which does the conversion from DC to AC and to supply to the AC loads. The output from the inverter gets fed into the energy metering system (energy meter PZEM-004T V3) and the NodeMcu board (containing the Wi-Fi module ESP8266), which serves as the control unit and finally to the various loads connected to the system. The Wi-Fi module present in the NodeMcu hooks the system to the internet over a Wi-Fi connection. The data which includes the value of current, load on each outlet, the battery level of the inverter can be accessed via the mobile application interface. It can then be monitored and controlled remotely. [5]

2.2 Steps in Implementing IoT-Based Smart Controlled Inverter.

Step 1: PV panel converts the Green Solar Energy into Electrical Energy.

Step 2: Received Energy will be stored in the battery through the Charge Controller in the inverter.

Step 3: The inverter will convert the DC to AC and supply to the different Load through the 4- channel relay circuit.

Step 4: The energy meter calculates the energy, power, current, and voltage passing through it.

Step 5: If the load current level goes above the threshold level the user can disconnect and control the unwanted loads using the Android app via Wi-Fi communication.

Step 6: When the load current level goes below the threshold value, the entire load / required loads will be connected using Android app / mobile URL site ON-OFF control via Node MCU Wi-Fi communication. [6]

The Flowchart below explains the principle of operation of the proposed IoT-based Smart Controlled Inverter.

GSJ: Volume 10, Issue 8, August 2022
ISSN 2320-9186

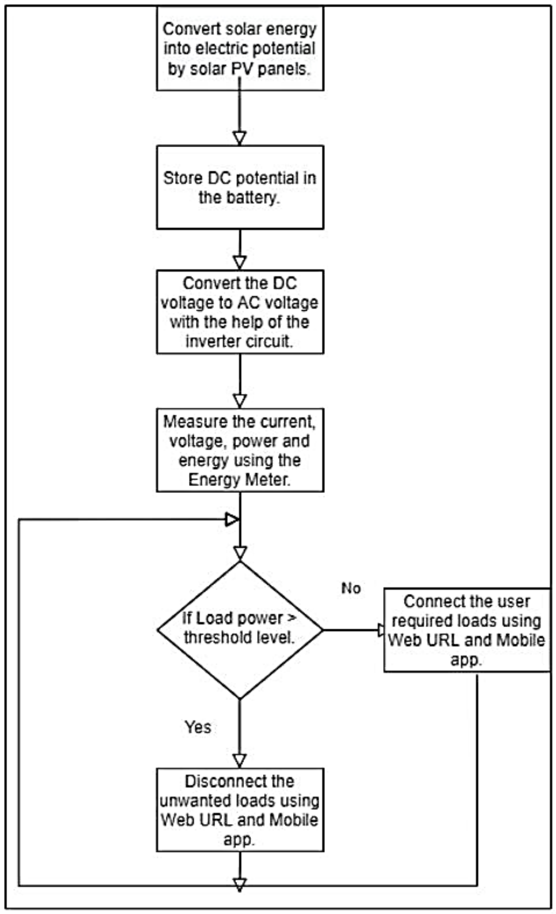

Figure 2 : System Flow Diagram

2.3 Project Compartmentalization.

The functional design of the IoT-based Smart Controlled Inverter is divided into sections:

1. Power Section.

2. NodeMcu and I²C and LCD Interface Section.

3. NodeMcu and input buttons Section.

4. Smart Energy meter (PZEM-004T module) and NodeMcu Interface Section.

5. The Relay Section.

6. The Cloud Section (back-end section).

7. The User Interface Section.

2.3.1 Power Section.

The proposed IoT-based smart controlled inverter is implemented by interconnecting the Solar PV panel, charge controller, inverter, battery, Wi-Fi Module, and current sensor with different types of loads through a 4-channel relay unit. In which the PV panel acts as the source of voltage which is stored in the battery through the inverter. The Voltage of the battery, after being reduced using a voltage divider circuit, is given to one of the ADC pins of the microcontroller. This digital value of the source voltage is used to perform necessary calculations to display the time for which the loads can run. ESP8266 is connected to the microcontroller to transmit and receive messages using the Transmit and Receive pins of the controller. When a user sends a message to the Wi-Fi module ESP8266, it sends the same to the controller which is programmed to accept the message and compare it with a predefined string. The diagram below illustrates the functional architecture of the Proposed IoT-based smart controlled Inverter. [7]

2.3.2 NodeMcu and I²C and LCD Interface Section.

The major components to be used in this section include;

• NodeMcu ESP8266

• 16x2 LCD using an I²C adapter for connection to the NodeMcu

• Energy Meter containing the Voltage sensing circuit and the Current sensing circuit

• Relay

• Connecting wires

• Bulb and lamp holder for visual inspection

• Power supply unit

As intended, the system should be able to read the previous values stored in EEPROM and restore them to the variables when powered up, they will check the available balance with the predefined value and take actions according to design. [8]

2.3.3 NodeMcu and input buttons Section

We used a button or switch as an input on a microcontroller (MCU), a "pullup resistor" was used so that the input is seen as a logical high when the (normally open) switch or button is closed. It was connected to Vcc, or logical high for input, and therefore "pulls up" the value on the pin to high. The corresponding program or software will need to match the state of a normally open (N.O.) switch as high when the switch is closed. "Normally open" is just that; its natural state is open until you close the switch. If you have no pullup resistor, your input will still read a state, but whether it will read high or low is unknown, since nothing is physically assigned to it. The exception is if your MCU has an internal pull-up resistor. When nothing is tied to a pin, it's commonly referred to as a "floating pin," because it has no assigned state of high/low, and noise can randomly influence the voltage level on the pin if it's left floating. If you press the button, the pin will be connected to the ground, and the only reason it doesn't short ground to the 5v supply is because of the pullup resistor. When the button is open in its normal state, the input seen is near the 5v (there is a negligible drop across the pullup resistor.) The value of resistance used is dependent on the NodeMcu ESP8266 which has an operating voltage of 3.3V. In general, you would use a pullup resistor that is an order of magnitude lower than the input impedance that's declared in the MCU's datasheet (in the case of the NodeMcu board, that would be the ESP8266 datasheet). Pulldown resistors are the same impedance value, but connected just the opposite as a pullup; the resistor is connected to the ground and the input is pulled low. The corresponding software would need to reflect the difference and the operation of the switch as normally open or normally closed would also factor into the logic of the whole setup. [9]

2.3.4. Smart Energy meter (PZEM-004T module) and NodeMcu Interface Section.

The term "smart" surfaces when trying to describe the operation of Energy Meters as such that is refined by measuring the quick estimations of voltage and current continuously to get the power consumption kilowatt-hours (KWh). The system will be designed with, AC-DC adapter with voltage divider, the NodeMcu ESP8266 board, and the 16x2 LCD. [10]

2.3.5 The Relay Section.

A relay is an electrically operated switch and like any other switch, it can be turned on or off, letting the current go through or not. It can be controlled with low voltages, like the 3.3V provided by the ESP8266 GPIOs, and allows us to control high voltages like 12V, 24V, or mains voltage. [11]

2.3.6. The Cloud Section (back-end section): Design of Database Model

Data in MongoDB has flexible schema documents in the same collection. They do not need to have the same set of fields or structures. Common fields in a collection's documents may hold different types of data. MongoDB provides two types of data models: — Embedded data model and Normalized data model. Based on the requirement, either of the models can be used. [12]

2.3.7. The User Interface Section (Mobile App).

The Mobile app was built with Blynk's mobile app development platform. Blynk mobile app helps us to build user interfaces rapidly, without the hassle of writing unnecessary code. Blynk is an "Internet of Things" (IoT) platform that allows you to build apps to control certain devices over the internet. Blynk is a platform with IOS and Android applications to control Arduino, Raspberry Pi and similar implementations over the internet. It's a digital dashboard where it can build a graphic interface for the user by dragging and dropping Control elements and widgets. Furthermore, because it creates code that can be shared across platforms, it makes it simple to develop for both Android and iOS at the same time.

Similar to React Native for the Web, React Native applications are created using a mixture of JavaScript and XML-Esque syntax, known as JSX. Then, beneath the hood, the React Native “bridge” accesses the native rendering APIs in Objective-C (for iOS) or Java (for Android). As a result, apps appear and feel like any other mobile app, as it will be rendered using native mobile UI components rather than web-views. Blynk exposes JavaScript interfaces for platform APIs, allowing apps to make use of platform capabilities such as the phone camera or the user's location. [13] [14][15]

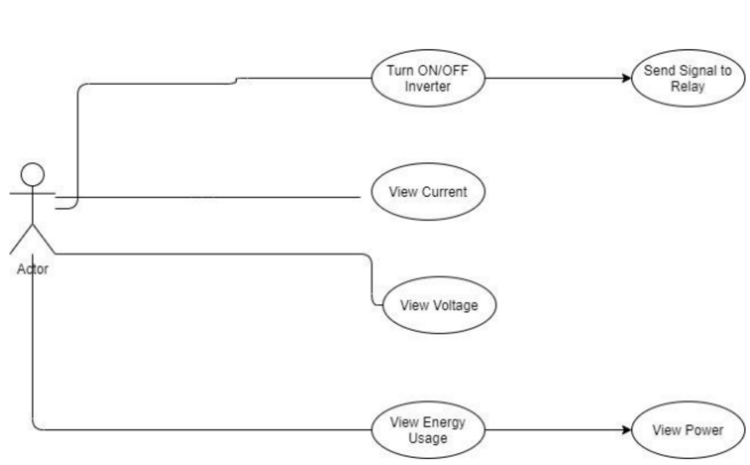

Figure 3: Use Case Diagram for Mobile Application

3. Results and Discussion

Using the specified components as described in the previous chapters and methodology as illustrated in the block diagram and the circuit diagram obtained from simulation of the project using Proteus computer-aided design software, a functional system of an IoT-based solar inverter using NodeMcu was implemented successfully. There was a seamless synchronization between the power section consisting of the battery and the solar-powered inverter, the microcontroller section, the cloud communication, the software and user interface section. Results obtained showed the test for different loads and the system operating at no-load (idle). Snippets from the graphical user interface of the application at different load options are shown below:

(a) At No-Load (idle) (b) System on Power

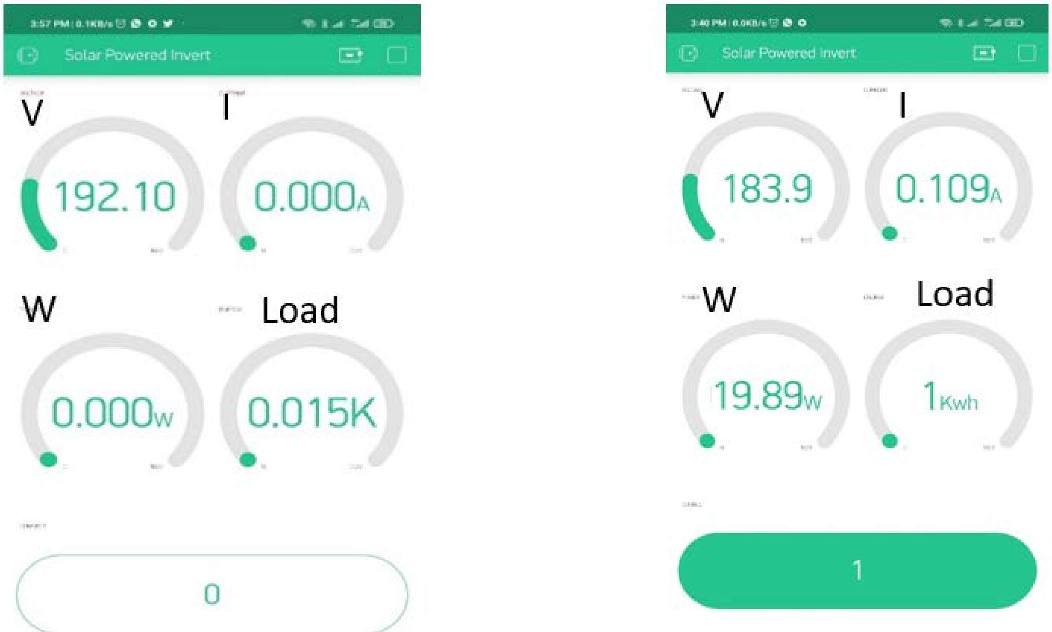

Figure 4: Screenshot of the homepage at no load and on load

3.1 Load Results

3.1.1 Load Under Test: 16w, 30w, 60w, 100w Load

Table 4.1 Result for Solar Inverter System

V_{AC}	V_{DC}	Load AC In Watts	Time In Minutes
220v	14.15v	Idle (0watts)	Start
220v	14.14v	16watts	After 2 minutes
220v	13.42v	30watts	After 2 minutes
220v	12.76v	60watts	After 2 minutes
220v	12.44v	100watts	After 2 minutes

Each load is removed from the outlets after two minutes and the system is tested with another load. On the software interface, the system is refreshed every 5 seconds and power consumption against time is updated on the analysis graph. This is shown below:

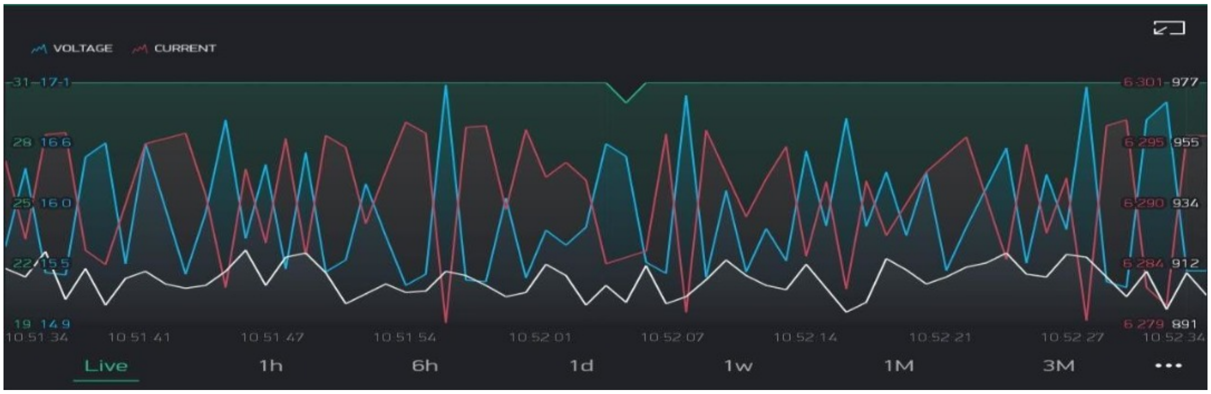

Figure 5: User Interface showing Graph of Voltage and Current against Time

The graph above represents the graph of voltage against time and current against time. The analysis graph records real time data, as well as hourly update.

4. Conclusion.

The aim of this project was to remotely monitor and control a solar powered based inverter using the internet of things, in the most efficient way. This was achieved at the end of this project. A user making use of his smartphone can remotely monitor the system and can cut off or shut off the system if load exceeds the desired threshold. This project will be of great use to individuals who own solar powered inverters and would want to be able to control and monitor their inverters regardless of their current location, it would also encourage users of alternative energy to purchase and exercise ownership of these systems which provides a greener source of energy and is safer for the environment, as this technique is developed further, it will go further than the scopes indicated and provide a more realistic attempt to fix solar inverter system inefficiency while also providing more helpful information regarding energy advancement.[15]

Nomenclature.

RFID	Radio Frequency Identification
IoT	Internet of Things
LTE	Long Term Evolution
PV	Photovoltaic
MCU	Micro Controller Unit
LCD	Liquid Crystal Display
AWS	Amazon Web Service
RES	Renewable Energy Sources
PEl	Power Electronic Interfaces
EMS	Energy Management Systems

BESS	Battery Energy Storage Systems
GTI	Grid Tied Inverters
PCC	Point of Common Coupling
NFC	Near Field Communication
GSM	Global System for Mobile Communication
GPRS	General Packet Radio Service

References

[1] Emodi, Boo, N., & Kyung-Jin. (2015, 6 16). Sustainable energy development in Nigeria: Current status and policy options. Renewable and Sustainable Energy Reviews.

[2] Gopal, G., Jidhun, O. D., Sai, K., & Siny, P. (n.d.). IoT Enabled Solar Smart Inverter. International Research Journal of Engineering and Technology (IRJET), 7(4). 6192-6196

[3] Kavitha, V., & Malathi, V. (2019). A Smart Solar PV Monitoring System Using IoT.

[4] Kurundkar, S., Kamthe, S., More, S., Marathe, P., & Gadade, Y. (2017). Remote Monitoring of Solar Inverter (An application of IoT). American Journal of Engineering Research (AJER).

[5] Qinghai, O., YanZhen, XiangzhenLi, YiyingZhang, & Lingkang, Z. (2021). Application of Internet of Things in Smart Grid Power Transmission. in proc. of Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing.

[6] Rathy, A. G., Gunasekaran, K., Perumal, K., & Sivasankar, P. (n.d.). International Journal of Computer Trends and Technology (IJCTT), 68(4). 2231-2803

[7] Rathy, G., Gunasekaran, K., Karthikeyan, P., & Sivasankar, P. (2020). IoT Based Smart Controlled Inverter. International Journal of Computer Trends and Technology (IJCTT), 68(4). 151-158

[8] Sathish, R. K., & Pooja, M. (n.d.). Single Phase Solar Inverter with Smart Charge Control. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 4(6) (992-996).

[9] Sheeja, N., Janvhi, J., & Muskaan, K. F. (n.d.). Smart Solar Power Inverter. Technium. 2(6). 64-73

[10] Shekhawat, R. S., Rao, R. S., KajalKumari, & Jumar, V. (n.d.). IoT Based Smart Controlled Inverter.

[11] Weiser, M. (1991). The Computer for the 21st Century. Scientific American, 265, 94-104. http://dx.doi.org/10.1038/scientificamerican0991-94

[12] What is AWS (Amazon web services) and how does it work. (n.d.). Techtarget. Retrieved January 30, 2021, from https://searchaws.techtarget.com/definition/Amazon-Web-Services

[13] React Native. (n.d.). Wikipedia. Retrieved January 30, 2021, from https://en.wikipedia.org/wiki/React_Native

[14] NodeJs. (n.d.). Wikipedia. Retrieved 1 30, 2021, from https://en.wikipedia.org/wiki/Node.js

[15] Nabilah binti Mazalan1 1Jabatan Kejuruteraan Elektrik, Politeknik Merlimau (2019) Application of Wireless Internet in Networking using NodeMCU and Blynk App