ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Development of IoT Solution for Solar Based Charge Controller

Tarun B, Dr.Venugopal N

M.Tech Student, Department of Electrical Engineering, PES Institute of Technology, Bangalore, India

Professor, Department of Electrical Engineering, PES Institute of Technology, Bangalore, India

ABSTRACT: Due to future industry requirements it is necessary to connect various hardware devices to the cloud for monitoring and diagnosing their behaviour over a period of time. The market leaders in cloud computing are Amazon web services and Microsoft Azure. Each of these clouds have their unique infrastructure in case of data storage, security and pricing. The objective of this paper is to send telemetry data from the solar based charge controller to Microsoft Azure IoT Hub using bare metal STM32 microcontroller and ESP8266 Wi-Fi module. An attempt has been made to reduce the overall memory consumption while connecting to the cloud.

KEYWORDS: IoT; FreeRTOS; Lwip; Azure.

I. INTRODUCTION

Internet of things is a future technology in the industry products. It mainly deals with cloud computation, Machine learning, Business Analytics .IoT can be dealt both from embedded point of view as well as application perspective. IoT from the application point of view deals with deploying Customized Web application on the cloud for Data logging and diagnoistics.From the embedded point of view a microcontroller is the heart of the project, From the Microcontroller the data can be sent through wired mechanism or wireless technique. In this project the Wi-Fi module is connected to the Micro controller through UART. Any electrical device can send some key parameters to the Microcontroller which processes it and sends the telemetry messages to the Wi-Fi module which in turn sends the same to the cloud. The main advantage of this is a performance of the device can be monitored at any place using the parameters which we sent to the cloud.

II. LITERATURE SURVEY

Internet of things is one of the fast emerging future technologies. In the future an estimated figure of 50 billion devices are expected to be connected to the Internet [1]. In order to meet the demand various protocols are developed in this area. The popular protocols used in this field are REST and MQTT (Message queuing telemetry transmit). When compared to REST, MQTT is a light weighted protocol that limits its header size to 2bytes and works well on memory limited platforms [2]. The Cloud architecture found in the market can be classified as public clouds and private clouds. The main difference is public clouds (Cloud MQTT, Eclipse paho) give a common URL to the users. In case of Private clouds (Amazon web service, Microsoft Azure) a unique URL is provided for each account created .These type of clouds expect a high level security from the embedded devices that connect to them.

III. FREE RTOS

An operating system is a computer program which takes care of the basic functions and provides various service to the application. The general purpose operating system run on Computers. But for embedded systems which has very less memory Real time operating systems are useful.STM32 used in this project is a bare metal system without any in built operating system unlike Raspberry Pi. FreeRTOS is an open source Real time operating system used in the project. It is light weight and was developed keeping the space crunched embedded systems in mind. It provides facility

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

for Task handling and memory management. It Implements Queues, Semaphores and Mutex. It is written in C and can be compiled using various C compilers (GCC). It doesn't restrict the number of Tasks being run as long there is sufficient memory in the hardware.

IV. TCP/IP STACK

TCP/IP stack is a set of communication protocols for interacting via the internet. It provides an end to end communication regarding how the raw data is to be picketed and transmitted through the internet The TCP/IP stack for Normal applications is not the same for IoT application. Light weight IP (LwIP) is an open source TCP/IP stack, is currently configured in the project.

The procedure for Porting TCP/IP stack to an NON-OS environment is as follows

- Develop a file to adapt Light weight IP to the compiler (GCC) and machine architecture.
- Create a Config file for the Light weight IP named as lwipopts.h. In this choose the various protocols required .Ex(Address Resolution Protocol,IPv4,IPv6, Transmission control protocol, UDP, Internet control message protocol, IGMP, HTTP, MQTT)
- Develop a network interface Driver:
- Create a Software Timer: In order to operate the Light weight IP in Non OS Mode certain functions need to be called at regular interval of time. Also the Presence of the hardware connection need to be verified by pooling the network interface every 250ms.

V. TRANSPORT LAYER SECURITY

Transport layer security pro-vide a secure communication between a client and a web server. Previously it was known as secured socket layer (SSL). Why Transport layer security is required?

- Privacy of data over Internet
- To avoid corruption of data Used by the server to check the integrity of the data from the client side so that it doesnt harm the server systems.

Each clouds have their own security standards. Amazon web service allows data from the client which are encrypted by TLS1.2 .It expects both server side as well as client side authentication. On the other hand Microsoft Azure expects only Client side authentication. In this project ARM mbed TLS has been used for Security purpose.

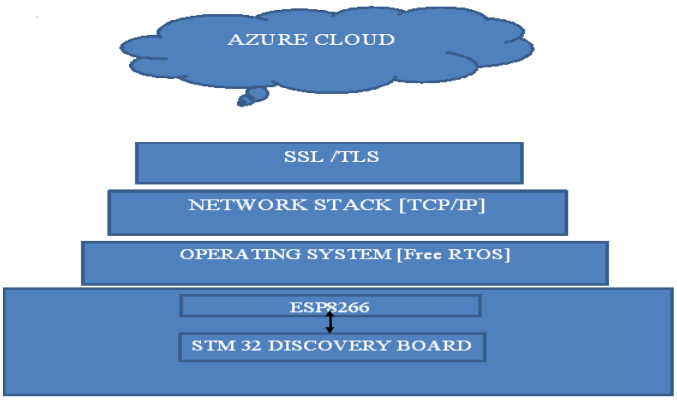

Figure 1: Overall data flow diagram for Azure connectivity

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

VI. SECURITY REQUIREMENTS FOR CONNECTING TO AZURE

The Microsoft Azure IoT hub provides a primary connection string for the end point devices. The Shared access token can be generated using this string. The generated token will be the PASSWORD in the MQTT connection packet.The Azure account will verify if the SaS token in the Connect packet matches with the primary connection string. Shared access signature token method of connection is suitable for devices with tightly constrained memory. The security check is straight forward. This method does not expect the embedded devices to provide the digital certificates. The Second options for bare metal devices to connect to Azure are generating a digital certificate. Azure itself does not provide any certificate for each end point device similar to Amazon web service. In this project Open SSL is used for generating the x.509 certificate. The x.509 provides an identity to the device. The various other information shared in the device are Certificate version , Time to Live (Shelf life of the certificate beyond which the certificate is invalid Region), /Country/Province ,Details of the organization and Encryption algorithm (SHA -1) Three certificates are usually generated for one device Id: Root CA, Client cert and Client Key.

VII. SOLAR CHARGE CONTROLLER

Charging of a battery requires regulated dc voltage. The main components in the solar battery charger are standard solar panels, a rechargeable battery, converter and a controller. The voltage supplied by a solar panel can change significantly depending upon the weather condition and irradiation from the sun. In order to charge the battery with a regulated voltage, a dc-dc converter is connected between the solar panel and the battery. In this circuit the input solar voltage is captured and sent to the STM32 board through Uart. It is possible to store the solar voltage over a period of time in the cloud. Pulse Width Modulation (PWM) is the most effective means to achieve constant voltage battery charging by switching the power devices. The heart of the charge controller is Arduino Uno board. The Arduino MCU senses the solar panel and battery voltages. In the Fig.1 the charge controller circuit is show interfaces to a Wi-Fi module through the STM32 microcontroller.

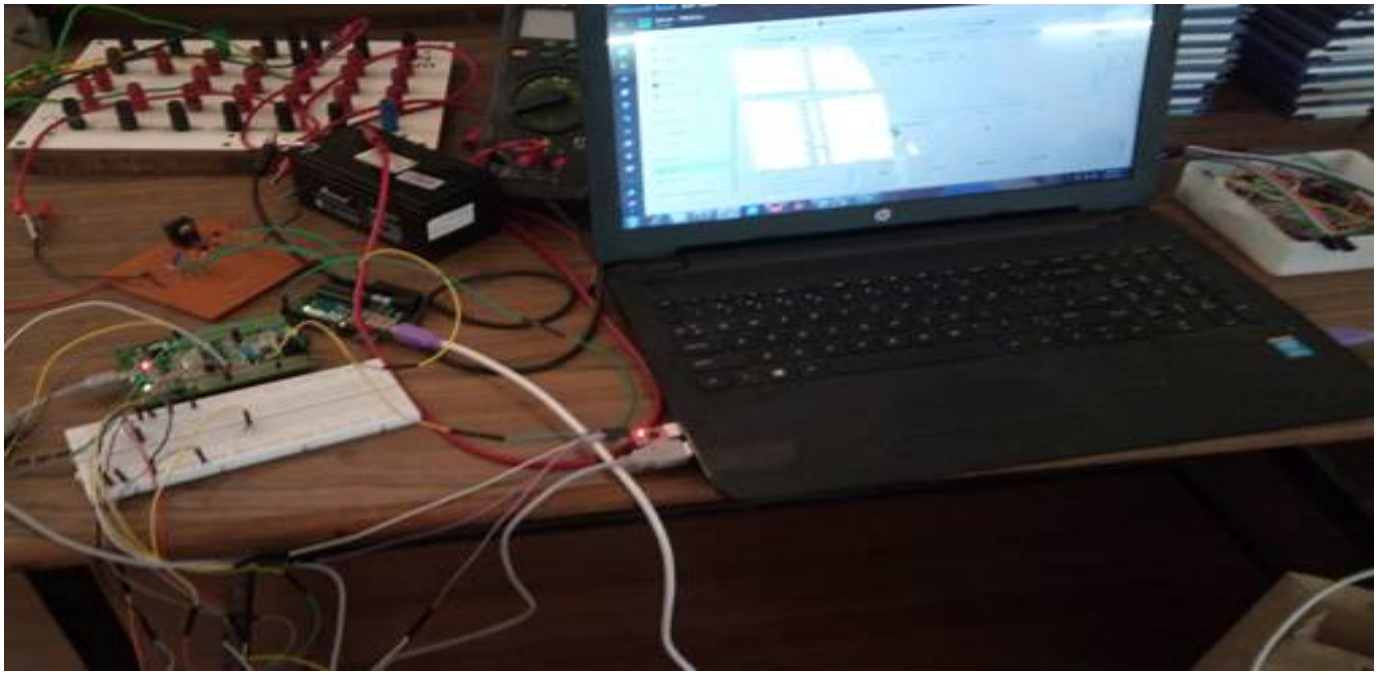

Figure 1: Picture of the solar input voltage data sent to the cloud using ESP8266.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

VIII. ADVANTAGES

- The setup is mainly designed for Memory constrained devices with RAM area less than 256Kb.
  FreeRTOS consumes around 50Kb (Depends on the number of Tasks Created)
  TCP/IP Stack consumes around 64Kb
  TLS/SSL consumes around 32Kb
  MQTT Library consumes around 20Kb.
- Compared to Linux based board using bare metal Micro controller boards is cost effective.
- In most of the IoT boards the Wi-Fi module is integrated to the Microcontroller. Any damage to the Wi-Fi module then the whole board has to be replaced. But in this setup the Wi-Fi module is externally connected can only this can be replaced if any problem arises

IX. RESULTS

In this Section the data that is logged in the Azure IoT Console is shown .In the IoT Hub the telemetry data sent can be seen over a range of time. The past sent telemetry messages can also be retrieved.

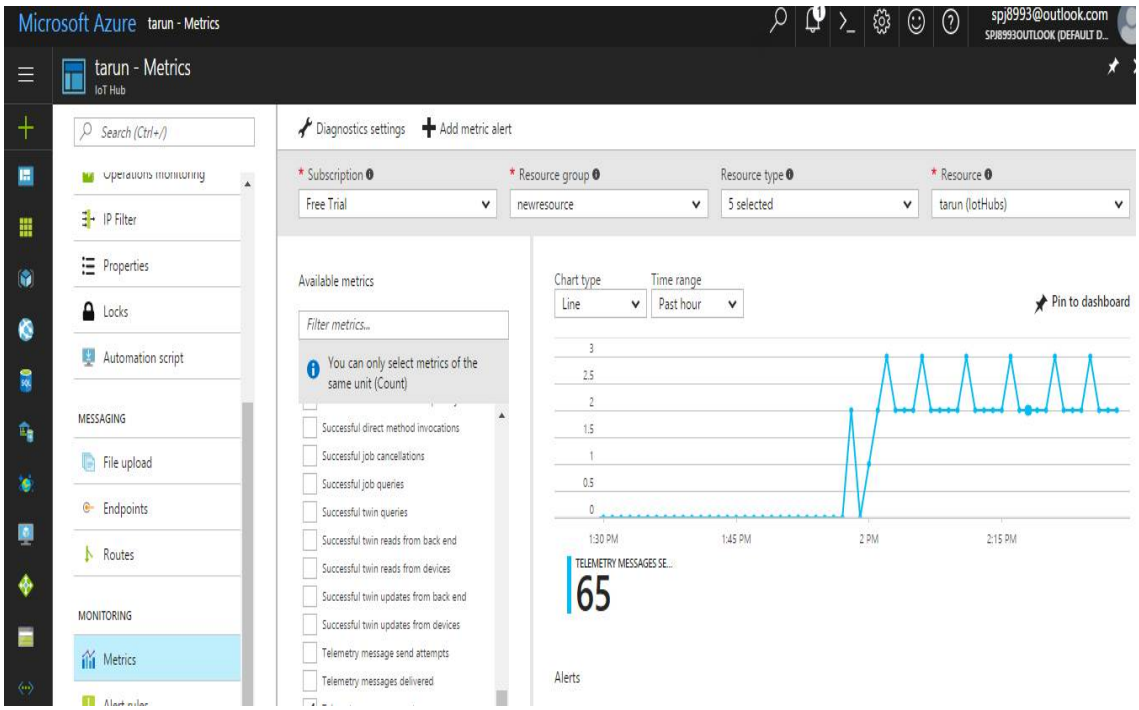

Figure 3: Data logging as visualized through Azure cloud Console

X. CONCLUSION AND FUTURE WORK

A Connection is established between the STM32 Board and Microsoft Azure. The connection was achieved by utilizing less than 200 Kb of RAM area. Any electrical device that can connect and send its key parameters to the STM32 Board and will be able to send the data to the Cloud. Using the data the performance of the device can be monitored. A device working in one part of the world can be diagnosed or monitored by a supervisor sitting in the other part of the world. In the future several devices can communicate with each other with the same technology.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

REFERENCES

[1] MQTT Website , http://mqtt.org
[2] Eclipse Paho Web page, eclipse.org/paho
[3] I. Stojmenovic, Machine-to-Machine Communications with In-Network Data Aggregation, Processing, and Actuation for Large-Scale CyberPhysical Systems, IEEE Internet of Things Journal, vol. 1, no. 2, pp. 122128, 2014.
[4] J. Sachs, N. Beijar, P. Elmdahl, J. Melen, F. Militano, and P. Salmela, Capillary Networks A Smart Way to Get Things Connected, Ericsson Review, vol. 8, pp. 18, 2014.
[5] D. Astely, E. Dahlman, G. Fodor, S. Parkvall, and J. Sachs, LTE Release 12 and Beyond, IEEE Commun. Mag., vol. 51, no. 7, pp. 154160, 2013.
[6] I. Vilajosana, J. Llosa, B. Martinez, M. Domingo-Prieto, A. Angles, and X. Vilajosana, Bootstrapping Smart Cities through a Self-Sustainable Model Based on Big Data Flows, IEEE Commun. Mag., vol. 51, no. 6, pp. 128134, 2013.
[7] A.Al-Fuqaha, Mohsen Guizani,MehdiMohammadi, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications- IEEE Communication Surveys and Tutorials, Vol. 17, No 4, 2015.
[8] Chiara Buratti, Andrea Stajkic, G.Gardasevic, S. Milardo, M. D. Abrignani, Testing Protocols for the Internet of Things on the EuWIn Platform IEEE IoT Journal, Vol.3, No.1, 2016.
[9] A. Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications," IEEE Commun. Surveys Tut., vol. 17, no. 4, Nov. 2015, pp. 2347–76.
[10]Azure_Developer_Guide_eBook, Microsoft ,2016