International Journal of Scientific Research in Mechanical and Materials Engineering
© 2019 IJSRMME | Volume 3 | Issue 4 | ISSN : 2457-0435

Implementation of IoT eith ESP8266 Part II – Home Automation

K. Priyanka Srinivasulu Reddy
Research and Development Engineer, Videocon Industries Ltd., Aurangabad, Maharashtra, India

ABSTRACT

This paper is the continuation and conclusion of my first paper 'IMPLEMENTATION OF IOT WITH ESP8266 PART I – CREATING A PROTOTYPE'. In the first part, a detailed understanding of IoT and Home Automation was given. Additionally, a small prototype was created for getting a feel of Home Automation through IoT. In this paper, a full-fledged home automation model has been created. Here we will sense various environmental parameters like temperature, light and smoke using sensors. The real-time plotting of these parameters will be done on MQTT cloud server. Our actuators – light, fan and buzzer will continuously access the status of these environmental parameters from our MQTT cloud server. Our actuators will be turned on and off accordingly. Additionally, we will create a mobile application for watching the real-time status of our actuators. Also, we will be able to turn our actuators on and off from our mobile application. The heart and soul of our entire project is a little hero 'ESP8266', which is a WiFi driven IoT chipset. This paper is a good overview to present what is a Smart Home and which are the necessary hardware and software components to make a Smart Home. [1,2,4,7]

Keywords : Internet of things (IOT), smart TV, smart devices – hand held devices, wireless sensor network, ESP8266, home automation, light sensor, temperature sensor, smoke detection, light bulb, fan, buzzer, mobile application.

I. INTRODUCTION

The building blocks of our IoT Home Automation project are – Hardware components, Programming software, Mobile Application development and MQTT Cloud server integration. [10, 11]

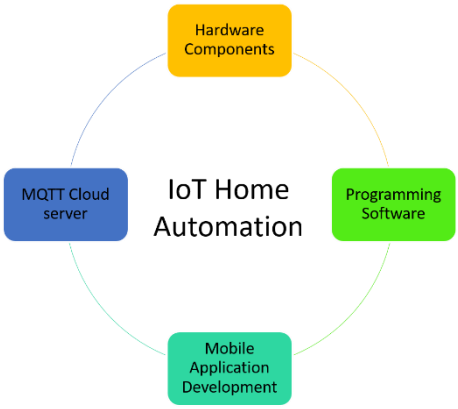


Fig. 1.1 : IoT Home Automation building blocks

Let us have a brief introduction about each building block, below.

Hardware Components:

a. IC ESP8266 - Espressif developed Wi-Fi enabled microchip ESP8266 proved to be a boon in the IoT field. This caused the arrival of various other ESPs and Open Source Development boards thereby allowing even a novice to make Wi-Fi featuring applications. [13,14]

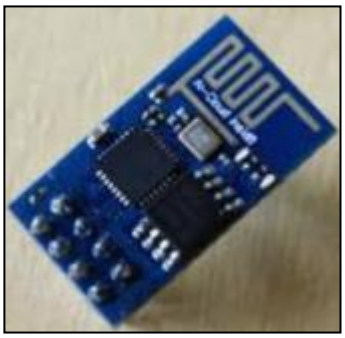


Fig. 1.2 : ESP8266 Chipset

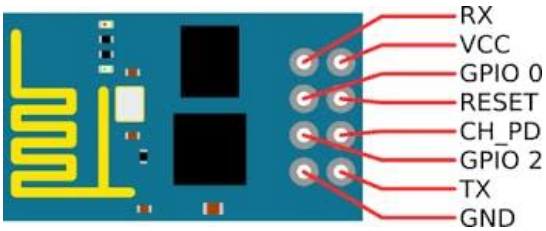


Fig. 1.3: ESP8266 Pinout diagram

ESP8266 has 8 pins, namely: RX, VCC, GPIO 0, RESET, CH_PD, GPIO 2, TX and GND.

VCC and GND are powering pins. RX and TX are used to communicate. You should also look at ESP8266 datasheet given in link below:

https://cdn.sparkfun.com/datasheets/Wireless/WiFi/ESP8266ModuleV1.pdf

b. LDR - Light dependent resistors, LDRs or photo resistors are often used in circuits where it is necessary to detect the presence or the level of light. They can be described by a variety of names from light dependent resistor, LDR, photo resistor, or even photo cell, photocell or photoconductor. Although other devices such as photodiodes or photo-transistor can also be used, LDRs or photo resistors are a particularly convenient electronics component to use. They provide large change in resistance for changes in light level. [13, 14]

Fig. 1.4 : LDR – Light dependent resistor

In view of their low cost, ease of manufacture, and ease of use LDRs have been used in a variety of different applications. At one time LDRs were used in photographic light meters, and even now they are still used in a variety of applications where it is necessary to detect light levels.

c. LM35 - The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full –55°C to 150°C temperature range. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air. [13, 14]

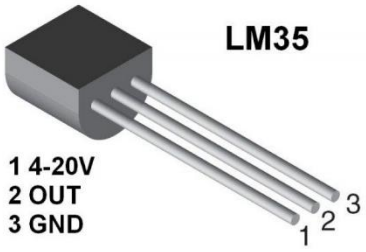


Fig. 1.5 : LM35 – Temperature sensor

d. LPG & Smoke sensor: The MQ-2 is a flammable gas and smoke sensor detects the concentrations of combustible gas in the air and outputs its reading as an analogue voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The MQ-2 gas sensor is sensitive to LPG, i-butane, propane, methane, alcohol, Hydrogen and smoke. They are used in gas leakage detecting equipment in family and industry and in portable gas detector. [13, 14]

- Supply Voltage:5V
- Sensitive to H2, LPG, CH4, CO, Alcohol, Smoke or Propane

- Analog and Digital Output
- Digital Out is high or Low based on an adjustable pre-set threshold.

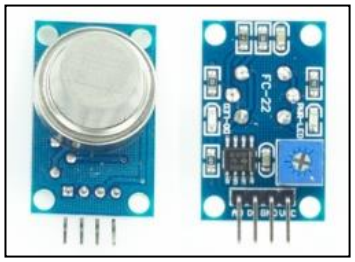


Fig. 1.6 : LPG & Smoke sensor

e. LM7805 IC: Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink. [13, 14]

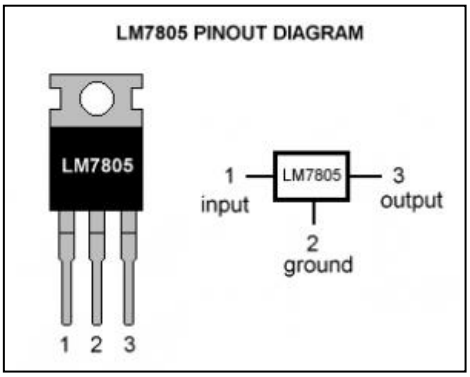


Fig. 1.7: LM7805 IC

LM7805 Rating:

- Input voltage range 7V- 35V
- Current rating Ic = 1A
- Output voltage range V (Max) = 5.2V, V (Min) = 4.8V

f. 5V, 2-channel relay: The Arduino Relay module allows a wide range of microcontroller such as Arduino, AVR, PIC, and ARM with digital outputs to control larger loads and devices like AC or DC Motors, electromagnets, solenoids, and incandescent light bulbs. This module is designed to be integrated with 2 relays that it is capable of control 2 relays. The relay shield use one QIANJI JQC-3F high-quality relay with rated load 7A/ 240V AC, 10A/ 125V AC, 10A/ 28V DC. The relay output state is individually indicated by a light-emitting diode. [3, 5]

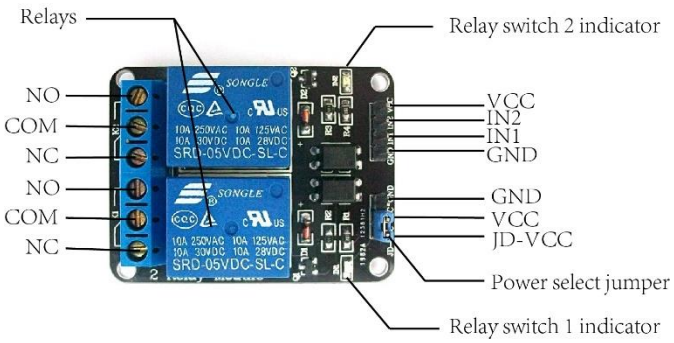


Fig. 1.8 : 5V, 2-channel relay module

g. Buzzer: In our project, we will use 5V DC buzzer. A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications. This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. [9, 12]

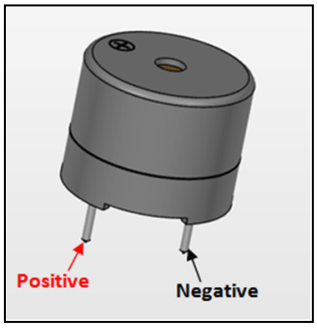


Fig. 1.8 : 5V, 2-channel relay module

h. 12V DC Fan and 230V AC bulb: We will use a generic 12V DC fan and 230 V AC bulb in our project.

Fig. 1.9: 12V DC Fan

Fig. 1.10: 230V AC Bulb

Programming Software:

For our project, we will be using Arduino IDE as programming platform. So what is Arduino IDE?

Fig. 1.11: Arduino IDE

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages. The IDE can be downloaded and installed from official Arduino website. [13, 14]

Mobile Application: For our project, we have created a smart home app using 'MIT App Inventor'.

MIT App Inventor is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to move from technology consumption to technology creation.

Fig. 1.12: MIT App Inventor

Blocks-based coding programs inspire intellectual and creative empowerment. MIT App Inventor goes beyond this to provide real empowerment for kids to make a difference -- a way to achieve social impact of immeasurable value to their communities. Many tutorial videos of creating 'Home Automation' app are available on YouTube. We can create and customize apps according to our requirements by following these tutorials.

MQTT Cloud Server: For our project, we have used 'ThingSpeak' cloud server for IoT communication. ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics. The ThingSpeak IoT platform enables clients to update and receive updates from channel feeds via the ThingSpeak MQTT broker. MQTT is a publish/subscribe communication protocol that uses TCP/IP sockets or WebSockets. MQTT over WebSockets can be secured with SSL. A client device connects to the MQTT broker and can publish to a channel or subscribe to updates from that channel. [6, 8, 13, 14]

Fig. 1.13 : ThingSpeak MQTT Cloud Server

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

- Easily configure devices to send data to ThingSpeak using popular IoT protocols.
- Visualize your sensor data in real-time.
- Aggregate data on-demand from third-party sources.
- Use the power of MATLAB to make sense of your IoT data.
- Run your IoT analytics automatically based on schedules or events.
- Prototype and build IoT systems without setting up servers or developing web software.
- Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

More information about ThingSpeak and how to use it can be found on its official website and this link: https://in.mathworks.com/help/thingspeak/mqtt-api.html

II. DESIGN AND DESCRIPTION OF PROTOTYPE

Hardware configuration:

The hardware configuration is as shown in the circuits below:

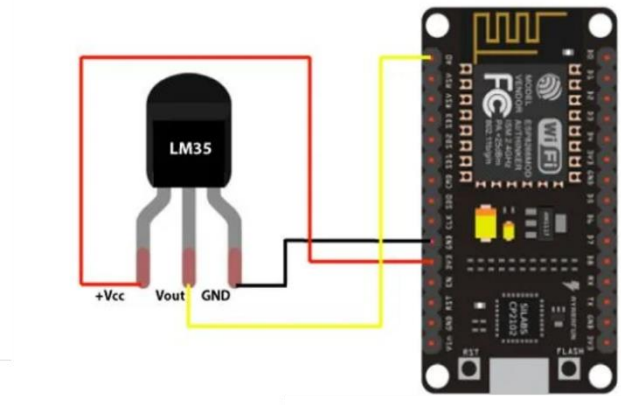


Fig. 2.1: Connecting LM35 to ESP8266

The circuit connections are made as follows:

Pin 1 of the LM35 goes into +3v of the NodeMCU.

Pin 2 of the LM35 goes into Analog Pin A0 of the NodeMCU.

Pin 3 of the LM35 goes into Ground Pin (GND) of the NodeMCU.

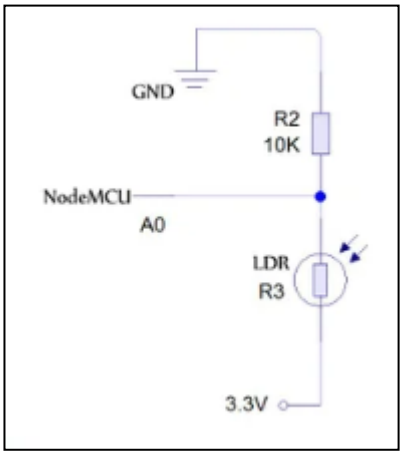


Fig. 2.2: Connecting LDR to ESP8266

The LDR output is actually analog in nature, so it gets connected to the A0 pin of the NodeMCU. Really quite simple right, just wire your prototype up like the schematic.

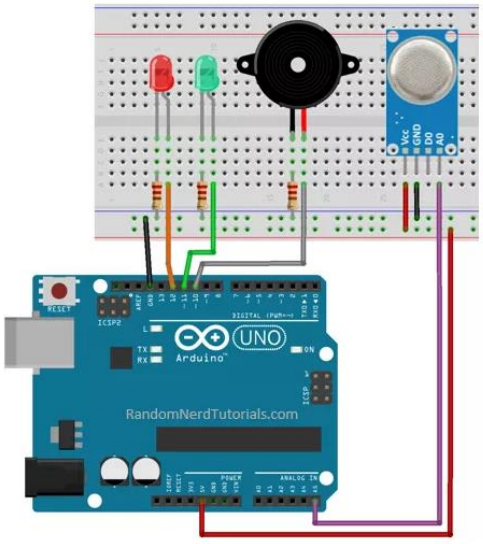


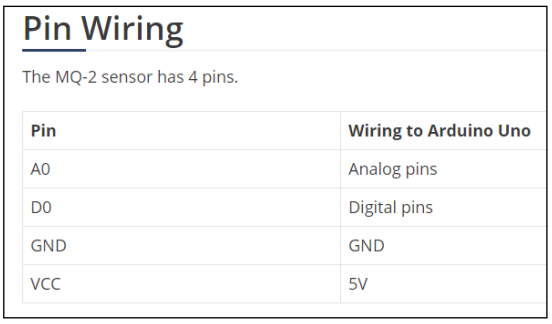
Fig. 2.3: Connecting smoke sensor MQ2 to Arduino

The voltage that the sensor outputs changes accordingly to the smoke/gas level that exists in the atmosphere. The sensor outputs a voltage that is proportional to the concentration of smoke/gas.

In other words, the relationship between voltage and gas concentration is the following:

- The greater the gas concentration, the greater the output voltage
- The lower the gas concentration, the lower the output voltage

The output can be an analog signal (A0) that can be read with an analog input of the Arduino or a digital output (D0) that can be read with a digital input of the Arduino.



Pin Wiring

The MQ-2 sensor has 4 pins.

Pin	Wiring to Arduino Uno
A0	Analog pins
D0	Digital pins
GND	GND
VCC	5V

Fig. 2.4 : Pin wiring for MQ-2 sensor

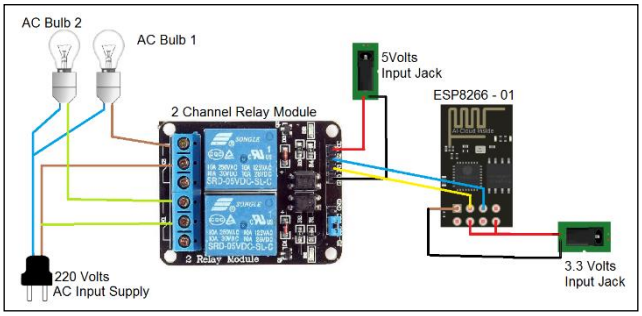


Fig. 2.5: Connecting bulbs to ESP8266 via relay

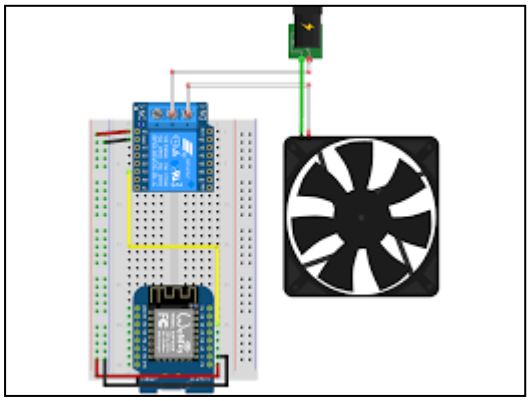


Fig. 2.6 : Connecting fan to ESP8266 via relay

Flow the circuit diagram for making the home automation system.

GPIO 0 -> Relay IN 1

GPIO 2 -> Relay IN 2

We will use 5 volts input supply for driving the Relay Module and 3.3 Volts Input for the ESP8266.

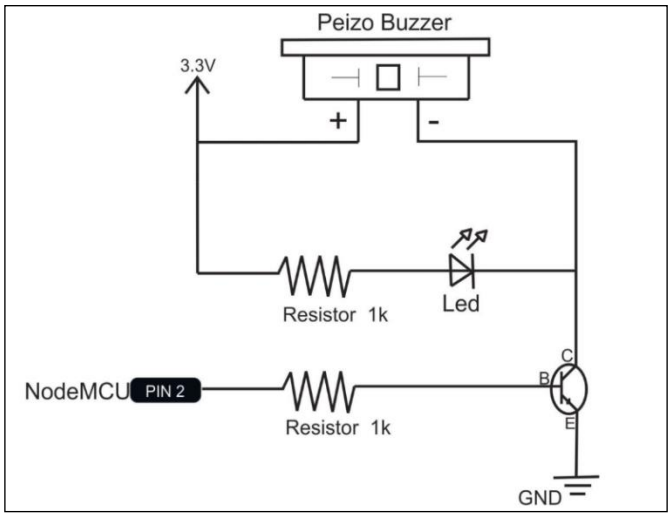


Fig. 2.7 : Connecting buzzer to ESP8266

Arduino programming:

Reading temperature –

Before getting the Celsius reading of the temperature, the analog output voltage from LM35 must first be read from the Vout pin of LM35. This will be the raw value divided by 1024 times 3300. It is divided by 1024 because a span of 1024 occupies 3.3v. Here we get the ratio of the raw value to the full span of 1024 and then multiply it by 3300 to get the millivolt value. Since the output pin can give out a maximum of 3.3 volts (1024), 1024 represents the possible range it can give out. Download the "LM35_NodeMCU.ino" file and open it up in the Arduino IDE. Then Create a new sketch and paste the code below in the Arduino IDE and hit Upload.

You can tinker with it if you like based on the application, or just use it as it is. [13, 14]

Mobile Application: We have made a simple mobile application using MIT App Inventor. In this app, we can control light, fan and buzzer. The app works like this – The status of these devices is read from the ThingSpeak.com cloud in real-time and fed into the app. The live status of these devices can be seen on the app. And when we press the button to turn a device on or off, this command is sent to ThingSpeak.com cloud. The cloud then communicates with our ESP8266 chipset controlling these devices and the devices are turned on and off accordingly.

Fig. 2.8 : Home Automation app

MQTT Cloud platform: The output on our MQTT cloud 'ThingSpeak.com' is as shown below –

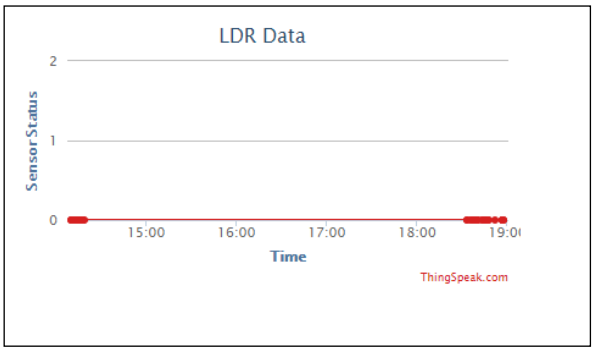


Fig. 2.9: LDR readings graph

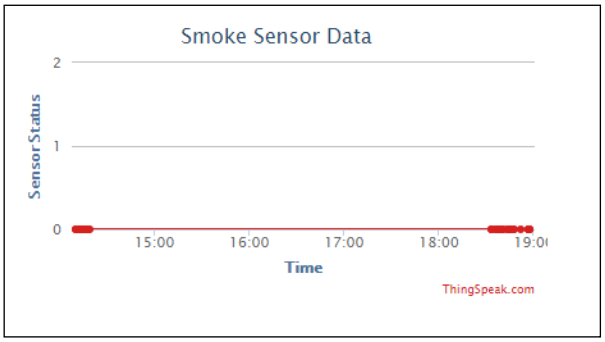


Fig. 2.10: Smoke sensor readings graph

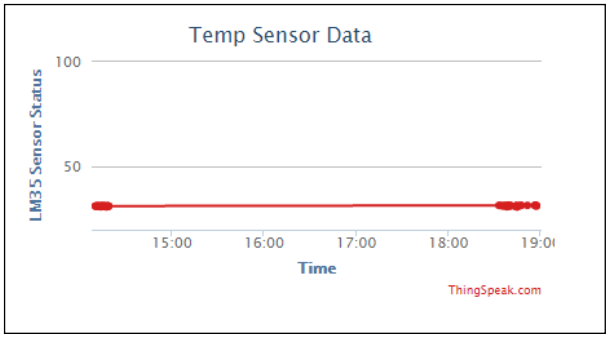


Fig. 2.11: Temperature readings graph

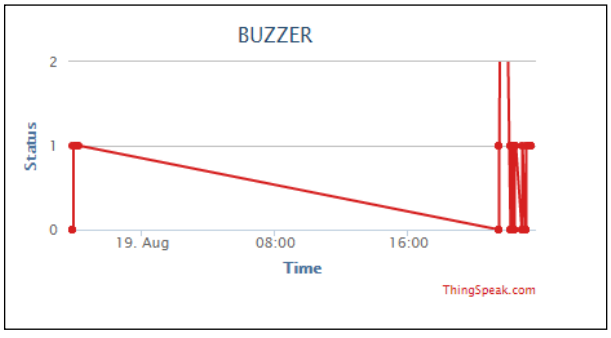


Fig. 2.12 : Buzzer status graph

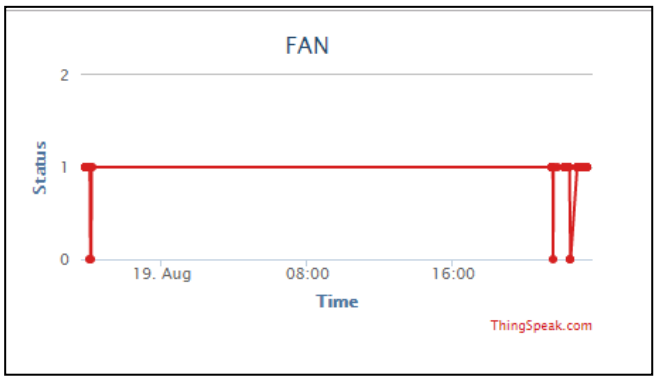


Fig. 2.13: Fan status graph

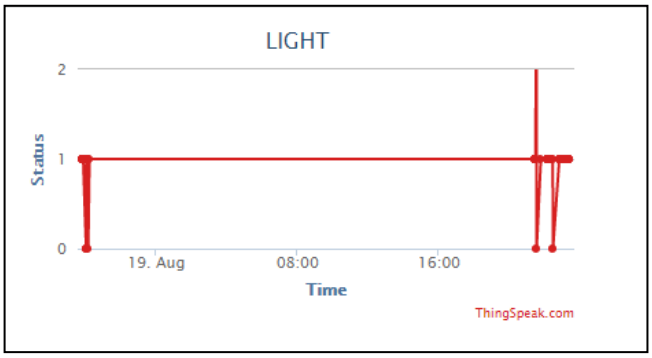


Fig. 2.14: Light status graph

Controlling devices:

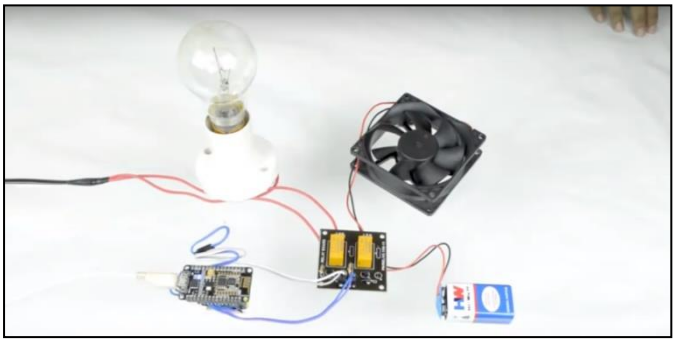


Fig. 2.15: Devices connection

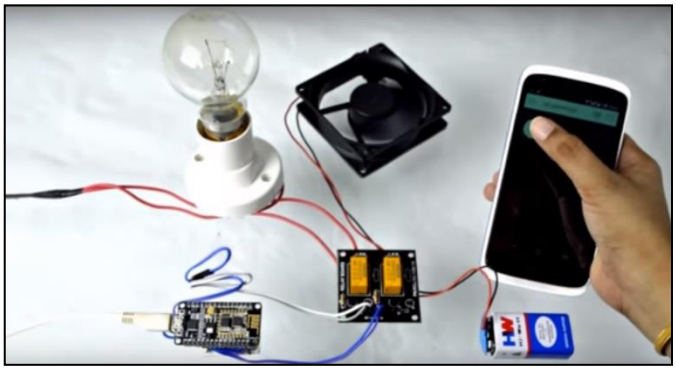


Fig. 2.16: Turning fan on/off

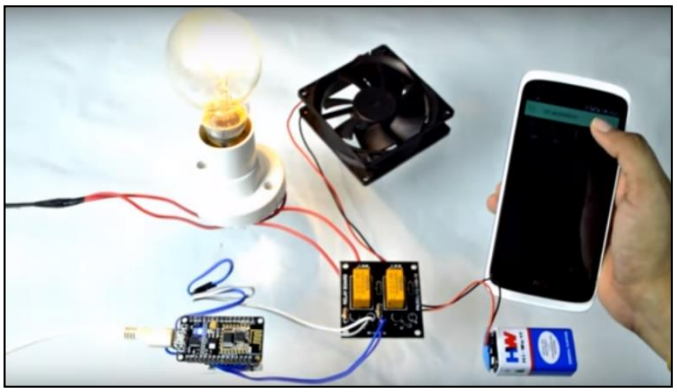


Fig. 2.17 : Turning light on/off

III. CONCLUSION

This paper, presents what a Smart Home is, which components are necessary to make a Smart Home. Firstly, a concept regarding indoor environment and smart home automation using Smart TV as the central monitoring and man to in presented machine interface unit. Second, the design and description of a prototype for controlling smart and dumb home appliances through IOT by the means of a Wifi driven chipset solution – ESP8266 is described in detail. In this, the hand held smart devices such as smart phones, tablets, smart wearables etc. act as the central monitoring interface for monitoring and control of smart home appliances. Lastly, the working and results of the project are given.

This Application notes detailed how to install software for developing IoT applications and testing them on home appliances, using the ESP8266-01. It then showed how to create Android program to control home appliances using this ESP module. The results with images were thereby depicted. The links provided in the rest of this document can be used to learn about IoT.

IV. REFERENCES

[1]. The Smart Home Concept : Our Immediate Future

[2]. https://www.researchgate.net/publication/224696459_The_Smart_Home_Concept_our_immediate_future

[3]. The Internet of Things and The Web of Things by Emmanuel Baccelli, Dave Raggett

[4]. Towards the Implementation of IoT for Environmental Condition Monitoring in Homes by Sean Dieter Tebje Kelly, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay, Fellow, IEEE

[5]. D. Surie, O. Laguionie, and T. Pederson, "Wireless sensor networking of everyday objects in a smart home environment," in Proc. Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process., 2008, pp. 189–194.

[6]. H. Sundmaeker, P. Guillemin, P. Friess, S. Woelffle, Vision and Challenges for Realizing the Internet of Things. Luxembourg, Germany: European Union, 2010, ISBN 9789279150883.

[7]. Internet 3.0: The Internet of Things, Analysys Mason, Singapore, 2010.

[8]. K. Ashton, That ―Internet of Things‖ Thing, RFiD Journal. (2009).

[9]. H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, Vision and challenges for realising the Internet of Things, Cluster of European Research Projects on the Internet of Things - CERP IoT, 2010.

[10]. J. Buckley, ed., The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems, Auerbach Publications, New York, 2006.

[11]. M. Weiser, R. Gold, The origins of ubiquitous computing research at PARC in the late 1980s, IBM Systems Journal. (1999).

[12]. Y. Rogers, Moving on from weiser's vision of calm computing: Engaging ubicomp

experiences, UbiComp 2006: Ubiquitous Computing. (2006).
[13]. https://randomnerdtutorials.com/guide-for-mq-2-gas-smoke-sensor-with-arduino/
[14]. https://electronicsforu.com/electronics-projects/smallest-iot-home-automation-esp8266-01

Cite this article as :

K. Priyanka Srinivasulu Reddy, "Implementation of IoT eith ESP8266 Part II - Home Automation ",International Journal of Scientific Research in Mechanical and Materials Engineering (IJSRMME), ISSN : 2457-0435, Volume 3 Issue 4, pp. 10-19, July-August 2019.
URL : http://ijsrmme.com/IJSRMME19346