

# Survey on NodeMCU and Raspberry pi: IoT

Anuradha Patel<sup>1</sup>, Dr. P. Devaki<sup>2</sup>

<sup>1</sup>Dept of IS&E, The National Institute of Engineering, Mysuru, Karnataka, India

<sup>2</sup>Asso. Professor, Dept of IS&E, The National Institute of Engineering, Mysuru, Karnataka, India

\*\*\*

**Abstract:** IoT (internet of things) is a big platform, where everyday devices are transformed in to informative automated system. In this paper, survey is done for the IoT, nodeMCU ( microcontroller unit) and raspberry pi. Because it's not compulsory that u can made the thing or solve the problem in one way, there are many another and different way to solve any problem or make any project.

So, we can make a project using nodeMCU or raspberry pi, it depend on us which device we want to use.

**Keywords:** IOT, Raspberry pi3B, NodeMCU, esp8266

## 1. INTRODUCTION

IoT is a big platform and became one of the most powerful communication paradigms of the 21th century. In a simple word, handling many things from one place without going physically overthere is IoT.

In [1], shows the basic of IoT, and compare between raspberry pi, arduino and esp8266.

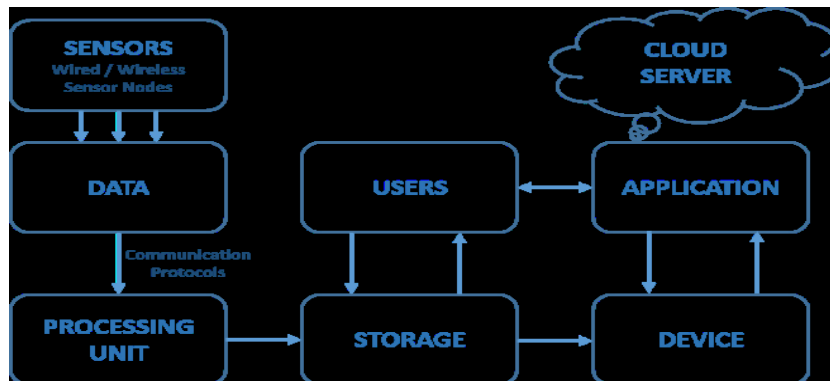


Fig 1.1 basic IoT functions block

Now a days, everything is going to be smart like, smart phone, smart car, smart city, etc.

Why are these all things called as a smart?



Fig 1.2 IoT application

Today we can do everything from our smart phone only, there is no need to go bank for transaction, no need to go cyber for the online booking, no need to go to hospital for the regular checkup because small-small checkup we can do at our home also, like blood pressure check-up, sugar level, temperature (for the fever). Now the world is going to be advanced, digitalized; so we can do our health-check up at home also.

Smart cities, including climate change, economic restructuring, the move to online retail and entertainment, ageing populations, urban population growth and pressures on public finances. As per [2], smart cities formed and it is shown in fig 1.2.



Fig 1.3 Bus network was implemented on the basis of smart city data analytics

## 2. COMPONENT:

- 1) **Raspberry pi:** [3] The Raspberry Pi Zero, a model first introduced in 2015. Several generations of Raspberry Pis have been released. The first generation (Raspberry Pi 1 Model B) was released in February 2012. It was followed by a simpler and inexpensive model, Model A. In 2014, the foundation released a board with an improved design in Raspberry Pi 1 Model B+. These boards are approximately credit-card sized and represent the standard mainline form-factor. Improved A+ and B+ models were released a year later. A "compute module" was released in April 2014 for embedded applications, and a Raspberry Pi Zero with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015. The Raspberry Pi 2 which added more RAM was released in February 2015. Raspberry Pi 3 Model B as shown in figure 2.1, released in February 2016 is bundled with on-board WiFi, Bluetooth and USB Boot capabilities. As of January 2017, Raspberry Pi 3 Model B is the newest mainline Raspberry Pi.

All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a Video Core IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either the SDHC or Micro SDHC sizes. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phone jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I<sup>2</sup>C. The B-models have an 8P8CEthernet port and the Pi 3 has on board Wi-Fi 802.11n and Bluetooth.



Figure 2.1.1:Raspberry Pi 3



**General purpose input-output (GPIO) connector:**

The GPIO pin diagram is as shown in the below figure 2.2.

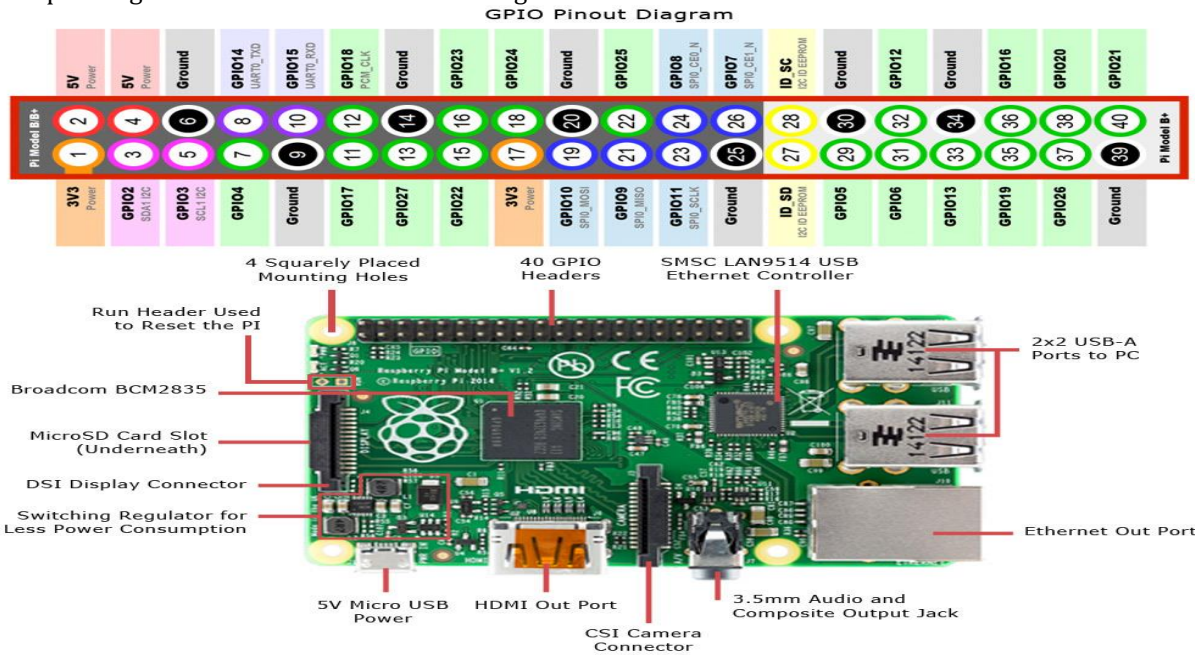


Figure 2.1.2: GPIO pin diagram

2) **NodeMCU:** "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. [4] NodeMCU is based on the Espressif ESP8266-12E WiFi System-On-Chip, loaded with an open-source, Lua-based firmware. It's perfect for IoT applications, and other situations where wireless connectivity is required.

There have been just two types of NodeMCU boards, versions 0.9 as shown in fig 2.2.1 and 1.0 as shown in fig 2.2.2.

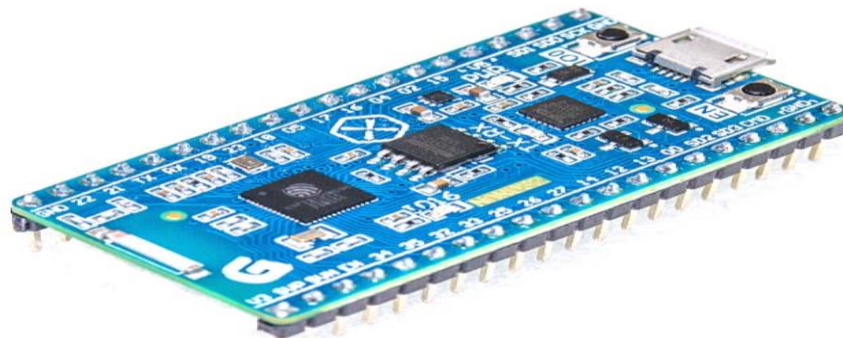


Fig 2.2.1: version 0.9

The 0.9 version is blue and comes loaded with the ESP-12 chip.

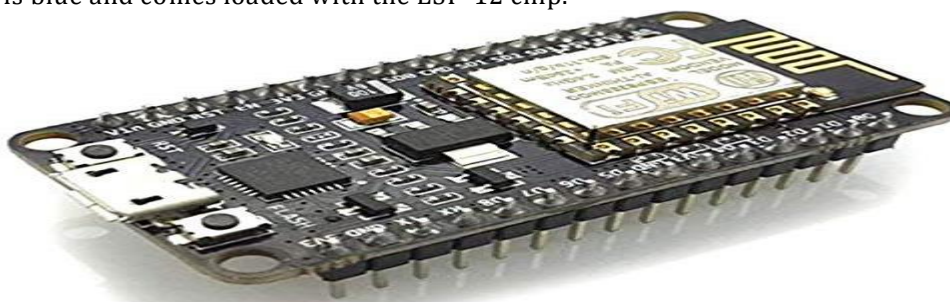


Fig 2.2.2: version 1.0

The 1.0 version is black and comes with the ESP-12E, it is updated version and the second version is 1.0.

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability. ESP-01 module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands.

**General purpose input-output (GPIO) connector:**

It can be either input pin or output pin, whose behavior can be controlled at the run time.

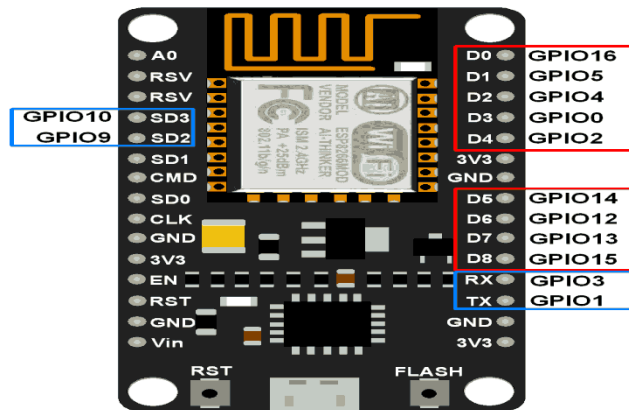


Fig 2.2.3 Pin configuration

The GPIO's shown in blue box (1, 3, 9, 10) are mostly not used for GPIO purpose on Dev Kit.

ESP8266 is a system on a chip (SoC) design with components like the processor chip. The processor has around 16 GPIO lines, some of which are used internally to interface with other components of the SoC, like flash memory.

Since several lines are used internally within the ESP8266 SoC, we have about 11 GPIO pins remaining for GPIO purpose.

Now again 2 pins out of 11 are generally reserved for RX and TX in order to communicate with a host PC from which compiled object code is downloaded.

Hence finally, this leaves just 9 general purpose I/O pins i.e. D0 to D8.

As shown in above figure of NodeMCU Dev Kit. We can see RX, TX, SD2, SD3 pins are not mostly used as GPIOs since they are used for other internal process. But we can try with SD3 (D12) pin which mostly like to respond for GPIO/PWM/interrupt like functions.

Note that D0/GPIO16 pin can be only used as GPIO read/write, no special functions are supported on it.

**Table 1 Comparison of Raspberry Pi-3b and ESP 8266:**

Si.No.	Parameters	Raspberry Pi B+	ESP-8266
1.	Processor	Quad-core ARM Cortex A53	-
2.	Operating System	Raspbian	-
3.	GPU	Broadcom VideoCore IV with 400 MHz	-
4.	Operating voltage	5V	3.3V
5.	Clock speed	1.2GHz	26 MHz – 52 MHz
6.	System memory	1 GB	<45kB
7.	Flash memory	-	up to 128MB
8.	EEPROM	-	-
9.	Communication supported	IEEE 802.11 b/g/n IEEE 802.15.4 433RF BLE 4.0 Ethernet Serial	IEEE 802.11 b/g/n
10.	Development environments	Any linux compatible IDE	Arduino IDE, Lua

			Loader
11.	Programming language	Python C C++ Java Scratch Ruby	Wiring, C, C++
12.	I/O Connectivity	SPI DSI UART SDIOCSI GPIO	UART, GPIO
13.	Cost	expensive	Low cost
14.	Used for	Complex project	Simple project
15.	GPIO Pin	26	17*

\*The ESP8266 has 17 GPIO pins (0-16), where only usable pins are 11 of them, because 6 pins (GPIO 6 - 11) are used to connect the flash memory chip. This is the small 8-legged chip right next to the ESP8266. If we try to use one of these pins, might crash our program.

### 3. CONCLUSION:

As, we can see that nodeMCU is cheaper than raspberry pi, so if we want to go for the simple and repetitive project then can go with nodeMCU like home automation, traffic light, or many repetitive project.

Whie if we think to make some complex and non repetitive project then we should go with raspberry pi, as we know that in nodeMCU we use C,C#, C++, so if we burnt that coding into in nodeMCU then updation can't possible while in raspberry we use python so, updation is possible and GPIO pins are also more.

So, the overall conclusion is that, if we are preparing something related to home automation then we can go through the npdeMCU because that the project will be cheap and a middle class family can also afford, and power saviour and the efficiency of the microcontroller will be good compare to raspberry.

If in future, updation can be possible or the project complex then we should go with Raspberry.

### REFERENCES

- [1] Dinkar R Patnaik Patnaikuni, "A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board", International Journal of Advanced Research in Computer Science(2017)
- [2] [https://en.wikipedia.org/wiki/Smart\\_city](https://en.wikipedia.org/wiki/Smart_city)
- [3] [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
- [4] <https://www.circuito.io/blog/nodemcu-esp8266/>
- [5] Vinay sagar K N, Kusuma S M "Home Automation Using Internet of Things", IRJET (2015)
- [6] Das, S.R., Chita, S., Peterson, N., Shirazi, B.A., Bhadkamkar, M, "Home automation and security for mobile devices,"IEEE PERCOMWorkshops, pp.141- 146, 2011.
- [7] Maksimović, Mirjana, et al. "Raspberry Pi as Internet of things hardware: performances and constraints." design issues 3 (2014)
- [8] John A. Stankovic," Research Directions for the Internet of Things", IEEE Internet Of Things Journal, Vol. 1, No. 1, pp 3 – 9,February 2014
- [9] <https://www.electronicwings.com/nodemcu/nodemcu-gpio-with-arduino-ide>