Node-Red based Industrial Automation Solution

Author
MOHAMMAD AHMED MEMON

Supervisor
Dr Bilal M. Khan

Thesis submitted in partial fulfillment to the requirement for the award of the degree of Master of Science in Electrical Engineering with Specialization in Communication

In

Department of Electronics and Power Engineering

Pakistan Navy Engineering College, Karachi

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, (NUST), ISLAMABAD

IN AUGUST 2019

Declaration

I certify that this research work titled "*Node-Red based Industrial Automation Solution"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

MOHAMMAD AHMED MEMON

NUST REGN # 00000118579

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr Bilal M Khan for his help throughout my thesis and also for Wireless Sensors and Networks course which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to Dr Sajjad Haider Zaidi for his tremendous support and cooperation. Each time I got stuck in something, he came up with the solution. Without his help I wouldn’t have been able to complete my thesis. I appreciate his patience and guidance throughout the whole thesis.

I would also like to thank Dr Lubna Moin for being on my thesis guidance and evaluation committee and express my special thanks for her help. I am also thankful to PNEC-NUST HQ for their support and cooperation.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

In the era of rapid smart industrialization, the Automation hierarchy of industries is smoothly transiting to more efficient, global, and easy to access and connected web of smart devices and cloud enterprises. With Industry 4.0 benchmarks and targets, the world is moving to not just towards smart industry but horizontal and vertical integration and scalability of automation layers. Use of Artificial Intelligence, Data analytics and standardization of Machine to Machine communication protocols, is where automation will converge

Industrial Internet of Things (IIOT) envisions and plug-and-play architecture for manufacturing sector, with not juts remote monitoring but also remote access and remote control of devices. The idea is to utilize robust IoT tools like Node-red, one of the powerful JavaScript based visual programming tool that takes advantages of open API's and integration with various affordable development boards ,cloud enterprises & light weight M2M communication wireless protocols like MQTT , and develop a prototype for Industry 4.0 as per its aims.

Comprehensible, open architectures, development platforms like Raspberry Pi with customizable architecture, and powerful processors with variety of interfaces is already creating a storm in the field of embedded systems.

Keywords: Node-Red, Industry 4.O, IIOT, MQTT, Raspberry Pi

Table of Contents

List of Figures

Chapter 1

1 Introduction

1.1 Motivation

In current dynamically changing industrial control automation scenario, IIOT (Industrial Internet of Things) and integration of cyber-physical systems (CPS) are getting a lot of attention from researchers all over. According to many future forecasts Internet of Things will continue to grow exponentially with estimated total of around $1.4 billion to be spent in 2021-(International Data Corporation).With IOT providing a test-bed for smart manufacturing and intelligent control which would enable to not just control the industrial floor from anywhere in the world but also centralize the overall control hierarchy and reduce the man-power thus paving way for industry 4.0.

Keeping in view the trends I have also proposed a Node-Red based industrial control prototype so as to conceptualize how real-time IIOT systems can be designed and what could be possible applications.

1.2 Problem Statement

In emerging world of new solutions where IOT application are increasingly developed and employed in every sector almost, a well-developed Industrial IoT will derive industry 4.0 and lead the race of manufacturing sector. Most of the IOT applications developed over the time for industrial automation are mostly based on remote access to data and remote monitoring of plant/manufacturing floor. As for Industrial sector there lacks a comprehensive solution that could enable user to remotely also control the manufacturing line/industrial floor or any actuators from anywhere in the world, unlike home automation where IOT provides many smart solution in just on application.

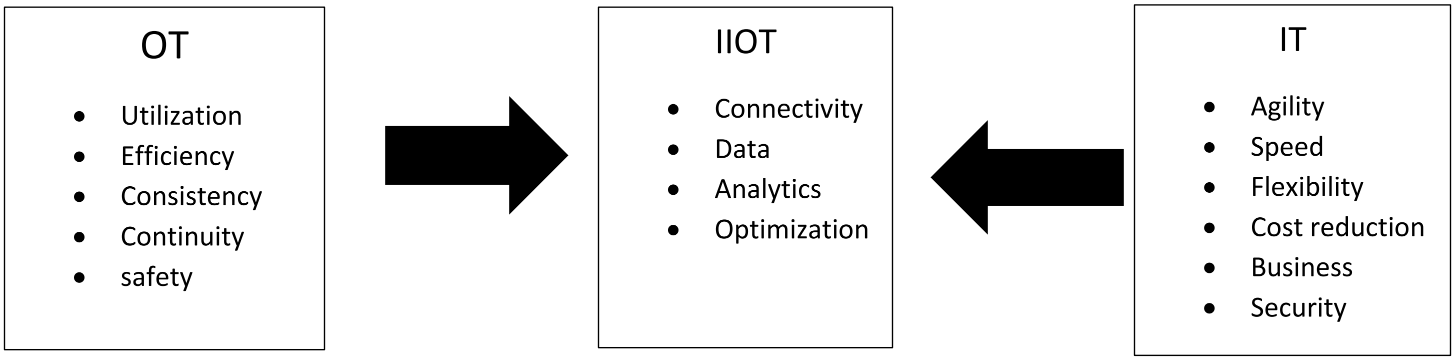


Figure 1-1 Genesis of Industrial Internet of Things

Therefore in my thesis I have provided a solution of IIOT that would cater the above identified issues, thus providing a solution for Industry 4.0 automation requirements.

1.3 Scope of Thesis

The research topic of my thesis is not just limited to a prototype but also will identify that how it suits Industry 4.0 application, how can one integrate power of a strong mini embedded system Raspberry Pi and a powerful java based flow-based programming tool Node-red along with A.I to further enhance the ease, productivity, cost effectiveness, preventive maintenance and shutdown failure predictions and control various manufacturing units globally from a just a centralized server.

1.4 Organization of Thesis

Hierarchy of my thesis will be like Chapter 2 will cover literature review and case studies. Chapter 3 will cover Industry 4.0, whereas chapter 4 will contain discussion about Node-Red as an important IOT tool and MQTT as pub/sub model communication for IOT applications. Chapter 5 will include Raspberry Pi and its benefits plus suitability for IIoT applications. In last Chapter 6 will contain our proposed solution

Chapter 2

2 Literature Review

2.1 Introduction

Before conceiving the idea for this research topic, I went through various papers related to Industrial Internet of Things (IIOT), its implementation, advantages and bottlenecks in the way forward. After cramming through many papers and analyzing the outdated industrial technology being used in our country, I realized the need for Industry 4.0 and its practical solution that can be made available to digitalize some aspects of industrial processes.

2.2 IBM Cloud Solutions for Home Automation

This papers suggests uses of IBM IOT platform, IBM Blue Mix for gathering data of various sensors and device on cloud. Author aims to utilize the services provided by IBM Blue mix like Speech to Text conversion, Visual Recognition and etc. using Node-red as a development tool and Raspberry Pi acting as a central controller. Home automation systems aims to bring ease to user to control and monitor various appliances sitting anywhere using internet. Devices can be connected to Raspberry Pi, acting as home gateway, using 6loWPAN technology. Whereas system proposes use of Amazon AWS EC2 platform for cloud server.
This hybrid home automation system uses services of IBM blue Mix , of which Audio-to-Text and Text-to-audio service did not provide accurate result, the paper also focusses on just home automation and therefore lacks any practical application for Industrial Automation. Paper shows the use of sensors that may not be compatible with Industrial

communication protocol. In addition to that the paper also lacks how secure system wireless network is or can be, but do provide an understanding of a how an industrial wireless network could be visualized by analyzing a home automation system.[1]

2.3 Environmental monitoring using Sense HAT based on IOT platform

The paper suggests a design for monitoring of environment using Raspberry Pi and an add-on module Sense Hat which contains various sensors like Humidity, temperature and etc. The idea is to connect sense HAT with raspberry pi module via 40 GPIO Pins. Various sensors data can be received my Raspberry Pi which is processed through Node-red and passed though IBM cloud services, from where it is stored in Cloudant NoSQL DB for monitoring and data analysis.
The design and idea is again more of remote monitoring and access, with little bit of design modification and more robust cloud data tool, but lacks any framework for industrial internet of things. Addition of Sense HAT shows how various sensors can be integrated with a standalone micro-processor unit and how this could be manipulated for some machine control.[2]

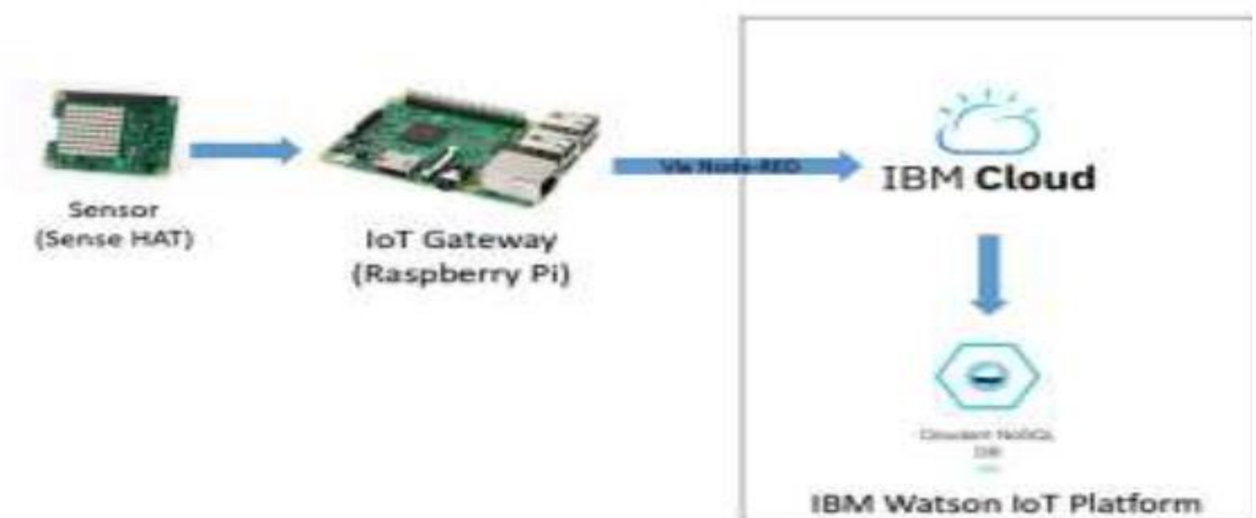


Figure 2-1 System Design[2]

2.4 Industrial communication based on Modbus and Node-Red

This paper identifies how Industrial Wireless Sensors Network can revolutionize the intelligent factories and how can it be deployed. In addition to this the paper also points out to the fact that how wired communication has limited the industrial ecosystem.[3]

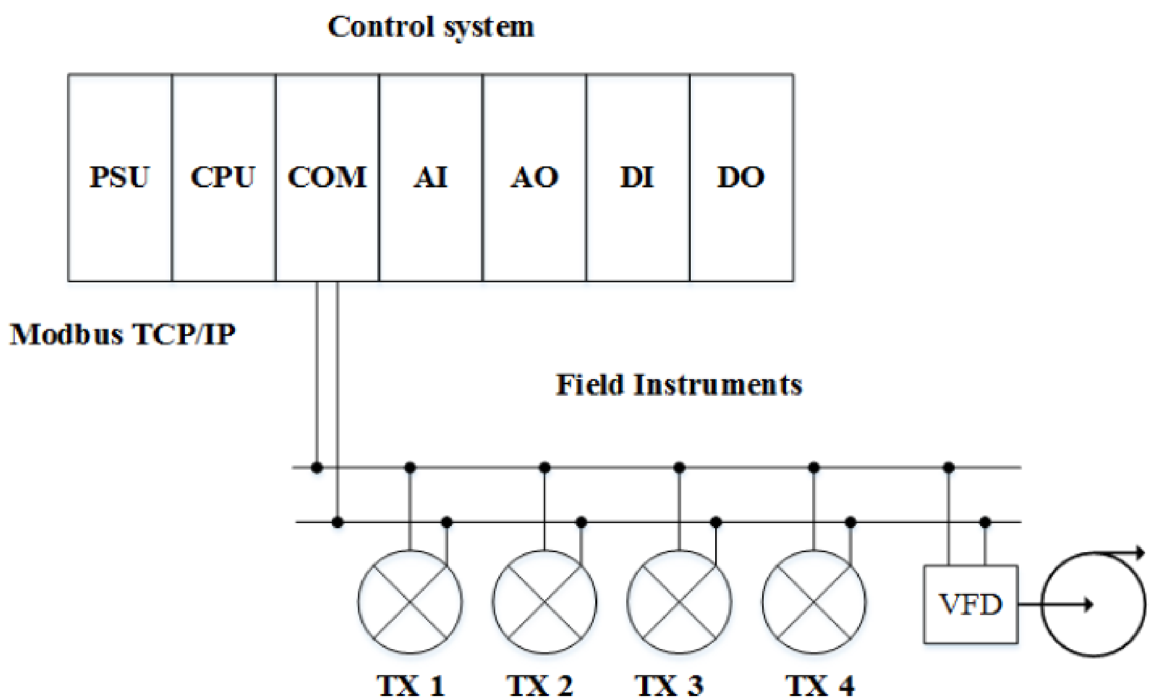
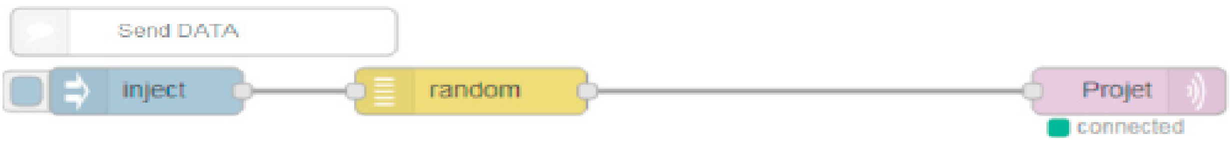


Figure 2-2 Field instruments and control system communication[3]

Paper presents simulation of industrial communication based on Node-RED framework using Modbus protocol and MQTT protocol. A layer of virtual Nodes (simulated MQTT clients), Node Red and Modbus protocol based industrial wireless communication architecture is created. Using mqtt-client libraries, virtual nodes were simulated to show master-slave communication flow in the system. A single on-board computer provided connectivity via MODBUS frame exchange.[3]



[3]
Figure 2-3 sending data

Data exchange can be seen in the figure below. The gateway setup to read/write Modbus registers and receive/transmit data via MQTT broker, accessed MODBUS register to read, formulate and send data to external message broker via MQTT managed by AMPQ cloud.

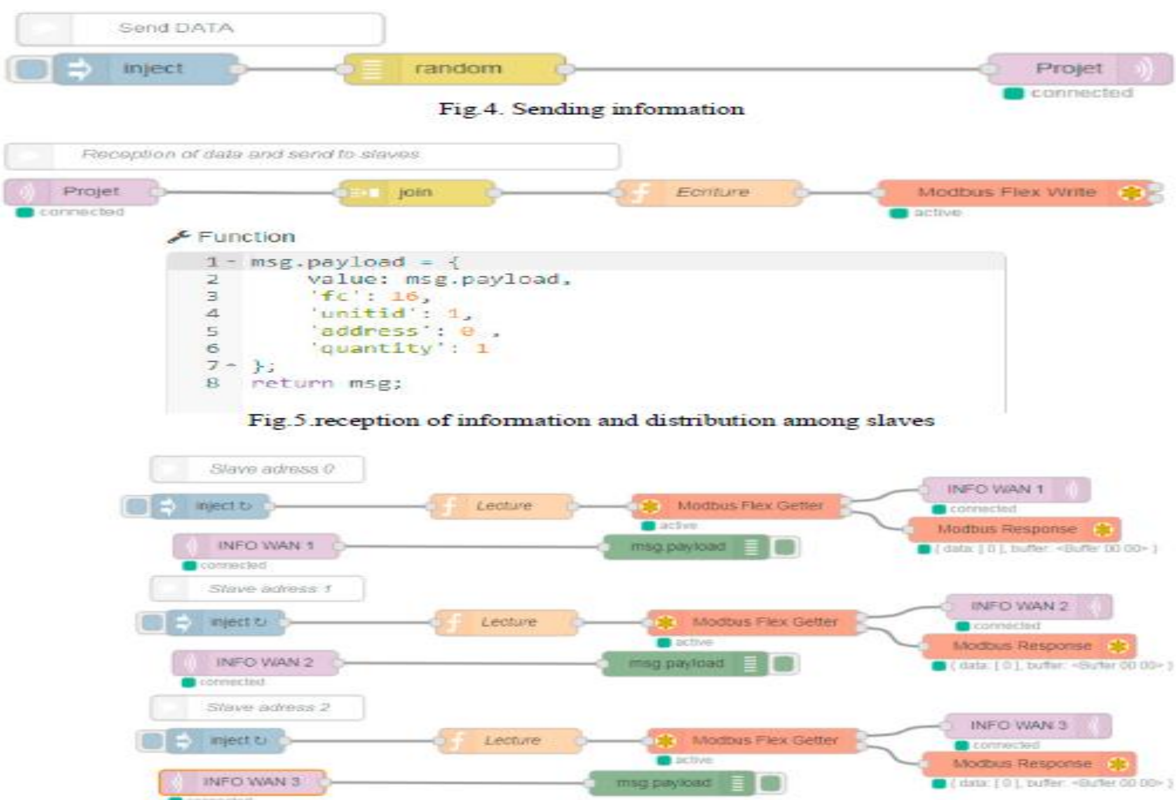


Figure 2-4Master/Slave communication via mqtt protocol

The test result show exchange of information between master and all slaves via MQTT and Modbus protocol and thereby envisions how M2M industrial communication can be made less reliable upon binding cables. The paper sets very important footnote on how IIOT communication can be updated with the help IWSNs and pub/sub messaging light weight protocols like MQTT, but at the same time presents an isolated environment simulation with virtual nodes not real-time industrial communication between equipment and does not provide results based on complex master/slaves combination of a real time industry.

Chapter 3

3 Industry 4.0

Industry 4.0 is about seamless connectivity and machine to machine communication. It envision a network of wireless sensors with agreed communication protocol , open architecture, interoperability and use of other digital technologies to further manipulate data and thus perform the required tasks efficiently. Although there are many hindrances in the path but a lot of frameworks are already set and worked upon. Merger of distributed physical, virtual and real world systems plus digital data form Cyber Physical Systems (CPS) which are networked to form ‘smart objects’.[4]

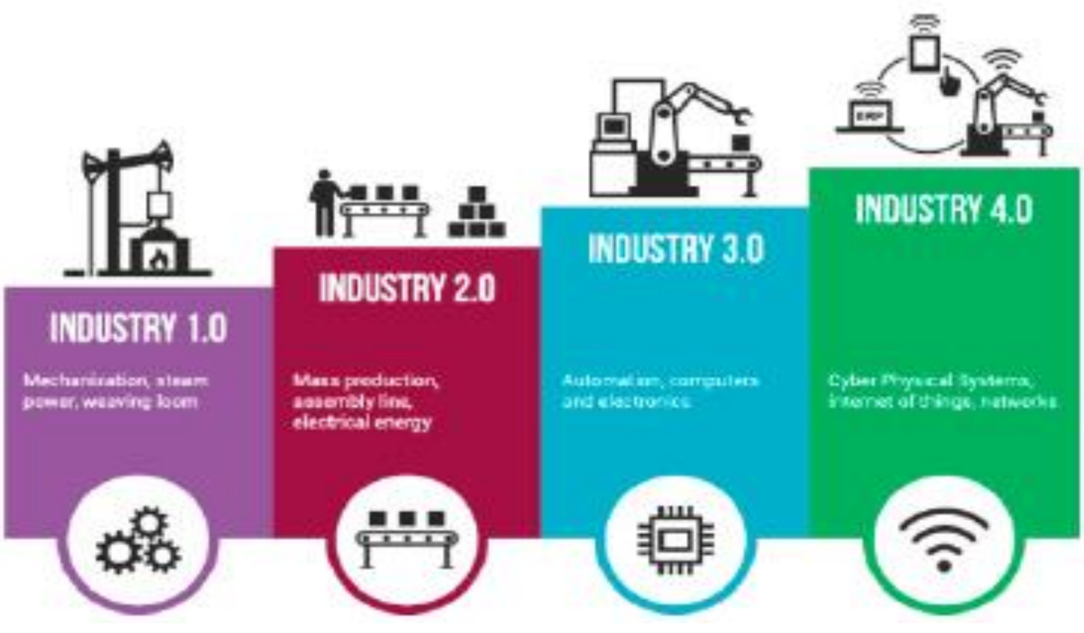


Figure 3-1 Industry up gradation cycle

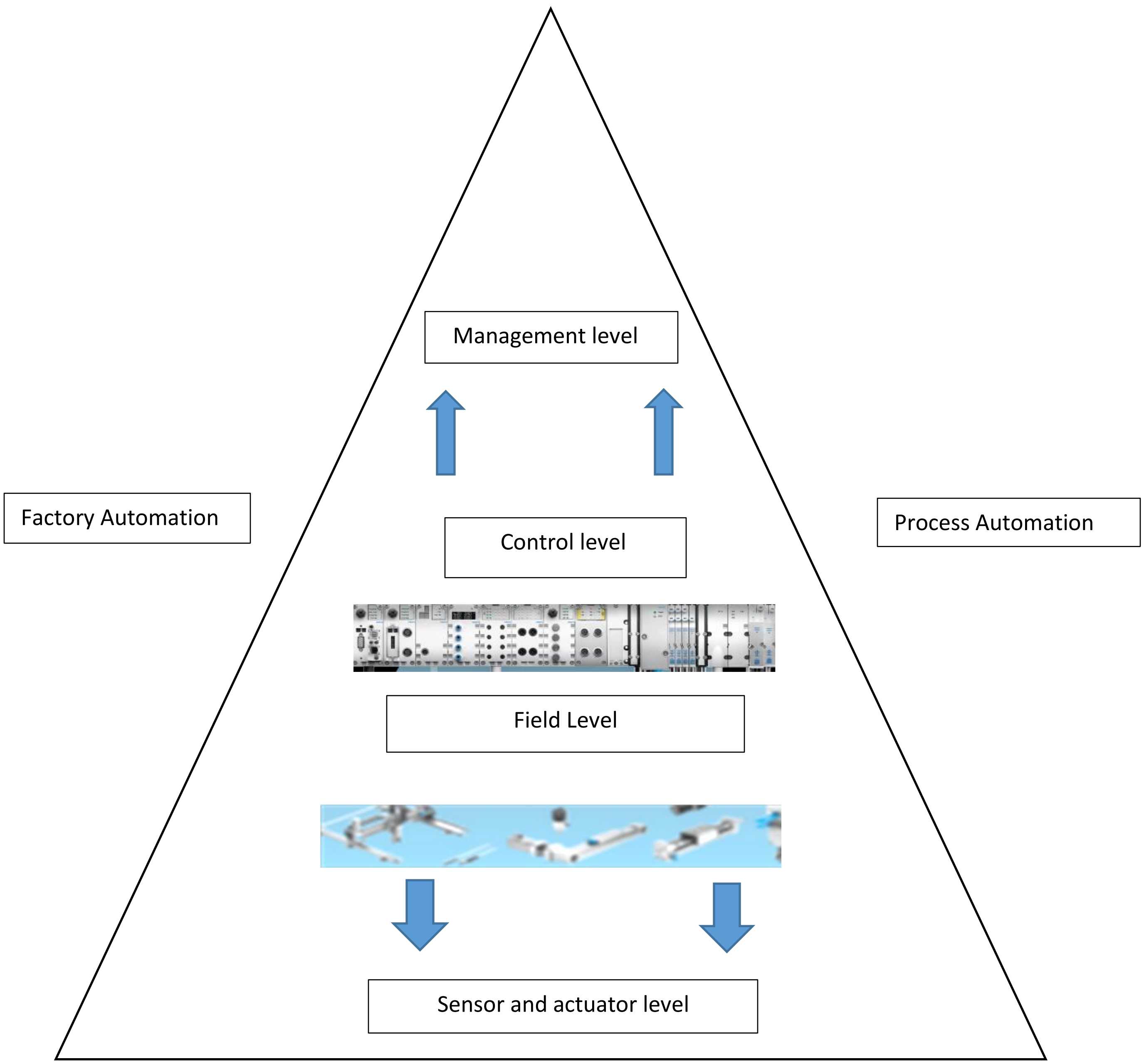
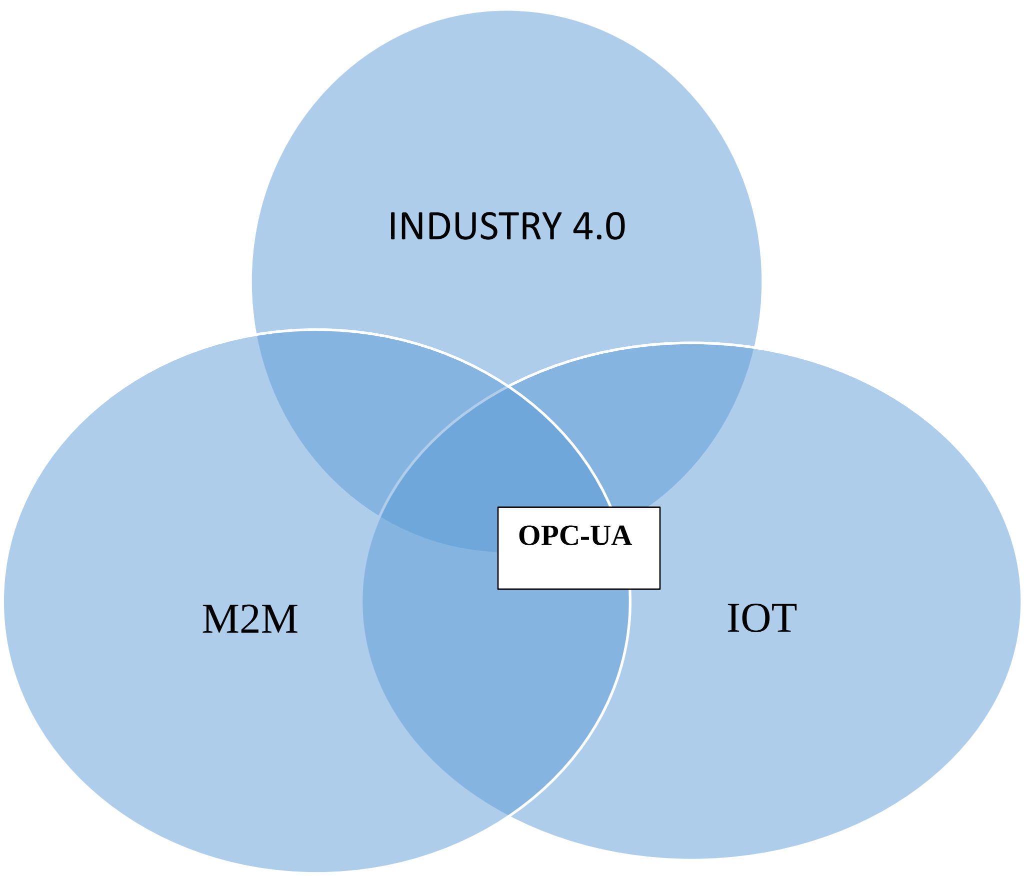


Figure 3-2 Automation Levels

3.1 OPC –Unified Architecture: Industrial Interoperability for IOT

For realization of IIOT it's important that scalability across all layers plus possibility of integration as well as vendor and platform independence is required. OPC-UA provides global standard of communication for all central devices and remote sensors. It operates as service-oriented architecture (SOA) and communication is independent of programming language in which the respective software was programmed and also independent of operating system on which the application runs. With it the service provider receives requests, processes them and sends them back with the results, contrasting with the classic web services that describe their services over WSDL, OPC-UA has pre-defined generic services.[5]



[5]

Figure 3-3 convergence of technologies

The information architecture with integrated address space is defined by OPC-UA in which process data, historical data and alarms can be represented together with function calls. The redundancy functions offered are integer able in both the client and server applications, thus ensuring implementation of high availability systems with maximum reliability. The security concepts that it employs' are proven ones that involve user and application authentication, signing of messages and encryption of transmitted data itself.

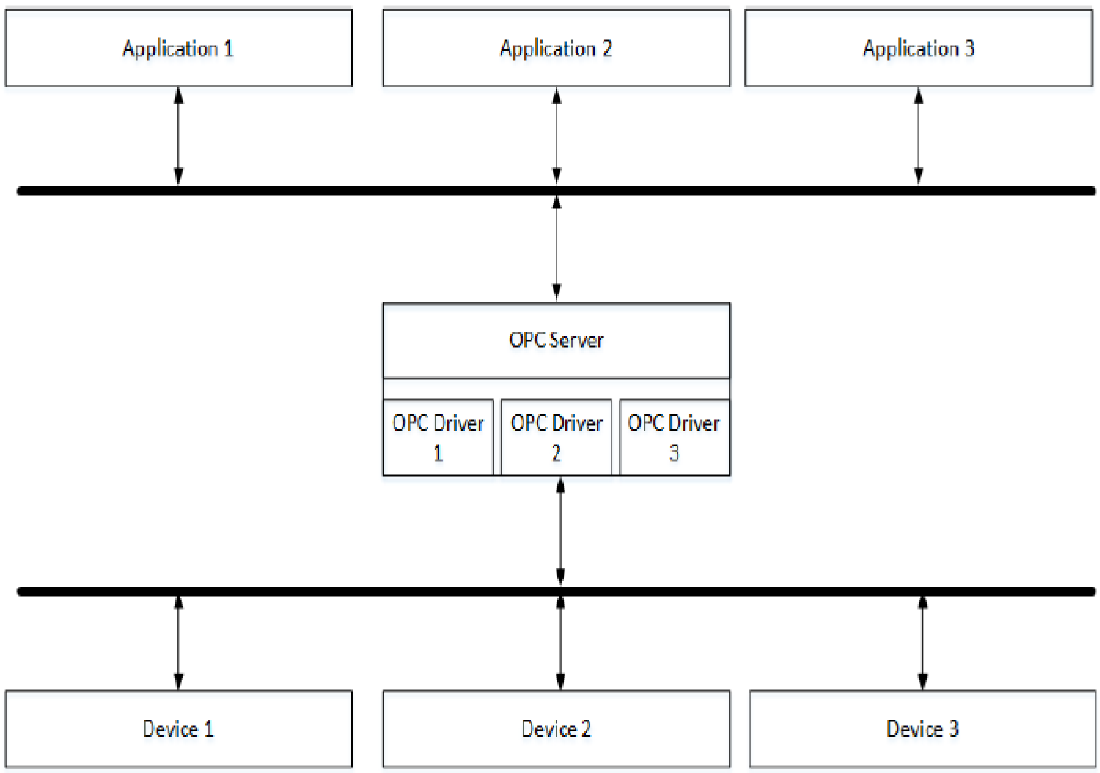


Figure 3-4 communication model based on OPC[4]

A lot of organizations are working for the development of more robust system, to summarize the requirements of Industry 4.0 and how OPC UA can fulfill them below is a table which will help one to comprehend the requirements.

Industry 4.0 requirements	OPC-UA Solutions
Independence from communication technology, manufacturer, sector, OS, programming language.	OPC foundation is non-profit, vendor-independent organization striving to develop neutral technological sector.OPC can run on all operating systems and there are even chip layer implementations without OS.It is compatible with all programming languages –currently stacks in Ansi C/C++, .Net and Java are available
Scalability for integrated networking ranging from for smallest sensors, to embedded systems, controllers and cloud applications etc. ensuring horizontal and vertical communication across layers.	OPC-UA is used in various embedded field devices and scales from 15kB through to single and multi-core variety of CPU architectures. There are many examples of successful projects in many cloud application like Amazon and Azure cloud
Secure authentication and transfer at both, user & application level	It supports signed and encrypted transfer with audit functionality available, whereas it utilizes X.500 certificates and user/password for application authentication.
Service Oriented Architecture, standard protocols like TCP/IP for transfer of live data, events and commands.	It is independent of transport method, but stacks ensure consistent transport of all data. Majorly available protocol binding are TCP-based binary protocol and HTTP/HTTPS web service with XML or binary coded messages. In addition to that Pub/Sub model can also be used.
Any complex information could be mapped for modelling of virtual data so as to represent real products and their production steps.	OPC-UA envisions plug-and-produce functionality without prior device configuration. It not only supports concept of fully networked for object oriented address space (for both hierarchical and meshed) including object description and metadata.
Incorporation into semantic and engineering extension	In order to promote optimizing interoperability between various engineering tools, the foundation has already made successful collaboration with many organizations like PLCopen, BACnet, FDI etc , and exploring new ventures also.
Conformity verification with defined standard	The foundation adheres to the various conformity standards and validations, plus to improve quality and compatibility there are additional test events (e.g. Plug fest). Foundation is already an IEC standard.(IEC 62541)

Figure 3-5 Industrie 4.0 needs and OPC-UA solution[5]

3.2 Architecture Alignment and Interoperability

So as to cater Industry4.0 needs and standardize it, two independent organizations, namely Industrial Internet Consortium (IIC) and Plattform Industrie 4.0, have independently developed reference architectures**:** **Industrial Internet Reference Architecture** (**IIRA**) and **Reference Architecture model for Industrie 4.0** (**RAMI 4.0**).Since Industrie4.0 is about making things *smartly,* whereas Industrial Internet is more about making things *work smartly,* therefore both the architectures are aimed at addressing different concerns of IIOT with different approaches that cover all layers and domains of industry 4.0 and also complement each other, thus opening room for interoperability.[4]

3.2.1 Industrial Internet Reference Architecture (IIRA)

To enhance the development of Industrial Internet Service and create a wide consensus for higher interoperability at industry level is what IIRA is pinning for. Orientation of IIRA is business-value driven and concern-resolution based IIoT reference architecture. Different architectural concerns identified by it are classified as view-points, namely *business, usage, functional and implementation,* which along with different business are core to IIRA model.[4, 6].

3.2.2 Reference Architecture Model for Industrie 4.0 (RAMI 4.0)

In order to facilitate Industry 4.0, RAMI 4.0 provides a 3 dimensional architectural model with service oriented architecture framework (SOA) which combines both services and data. It develops a common understanding and perspective, shared by all participants. The architecture of RAMI 4.0 can be described by six layer: *business,* *functional, information, communication, integration and asset.* RAMI 4.0 not only takes into consideration the factory machinery and operators, but also outcomes of manufacturing processes and external collaborators.

It introduces a notion of ‘administration shell’ by defining characteristics of a group of thing , so that it could be regarded as a component of I4.0[4, 6]

3.3 Case Study: Smart Factory

Realization of smart factory is where all research leads to. Smart factory envisions a network of distributed machines, flexible systems and functions where across all hierarchy levels participants can interact and network could cross company boundaries in which products are also a part.[7]

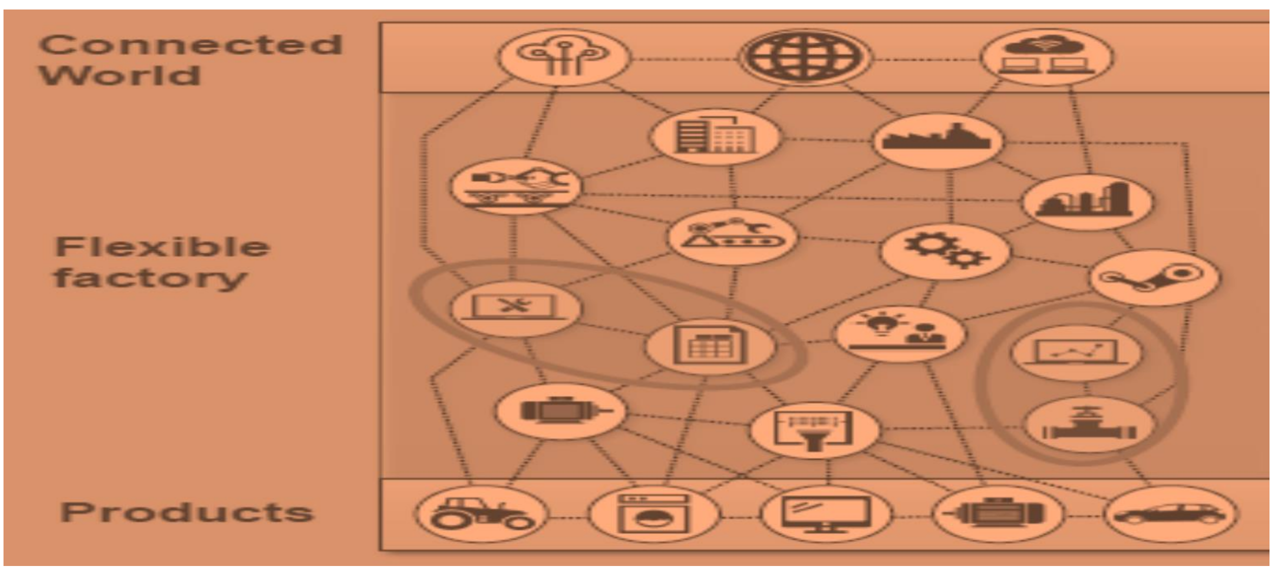


Figure 3-6 SMART FACTORY[7]

In order to visualize the infrastructure of a smart factory one must understand three important layers i-e Physical resource layer, Network layer and data application layer, and their hierarchy.

Although a lot of different researchers have proposed different models for each layer which are out of the scope of my thesis therefore I will discuss paradigms of *smart factory* in a precise manner.

I. Physical Resource layer

The physical equipment used must be intelligent in real time data acquisition and it must be ensured that there is standard communication protocol and transmission of heterogeneous data is at high speed.
The production line must be reconfigurable as well as adaptable to changes. Reconfigurable manufacturing units are deployed which can have modular units and reconfigurable controllers. Adaptive robots or mechanical arm which can be configured with adaptive algorithms or cognitive robotic island vertically integrated in industry which could perceive information uncertainty , work independently and adjust to scheduling changes of factory.
Configurability of Control system refers to ability to extend, integrated, re-use hardware and to work in multi-purpose scenario. It pins on open architecture with use of various automation tools and languages like (XML and Ptolemy II) and Field programmable gated array (FPGA) or advance distributed intelligent CNC control.
In addition to that a reconfigurable production line to produce variety of products is also mandatory for a smart factory. Although there are lot of bottlenecks to it but various researchers have proposed methods for it also that emphasize on minimization of reconstruction complexity and maximization of workbench recyclability. Workflow plan must be organized in such a way so as to assign products to dedicated

reconfigurable production lines and optimize system configuration and production plan.

For data acquisition wireless sensors networks are deployed which must also be intelligent .Commonly used WSN's like Zigbees, RFID and Bluetooth must be integrated to ensure heterogeneous data acquisition from various sensors. It is this data that is further processed and helps in decision process since they are the eyes of a control system thus RFID based customized data acquisition system or low powered blue tooth and zig bee based systems can be incorporated which must be compatible with industrial communication protocols like OPC, Open data base connectivity (ODBC), RS232 and Dynamic Data Exchange (DDE), plus these devices interface should be easily setup, flexible and scalable.[8]

II. Network layer

In Network layer, new technologies are supported be Industrial Internet of Things (IIOT) with scalability and high flexibility of new protocols and data format .Technologies like OLE for OPC-UA, device to device communication (D2D) and Software defined Radio Networks (SDN) to ensure QoS (Quality of Service), reliable communication and equipment cooperation Low latency, high reliability and high synchronous accuracy must be part of Industrial Wireless Sensors Networks when dealing with control service. Software defined IIOT architecture can also be used for network resource allocation and with customizable networking protocol information exchange acceleration can be done.[8]

III. Data Application Layer

In this layer Ontology based model is employed where cloud platform should analyze various data semantics which provides smart factory self-learning and self-adaption and basics of decision making. Further design optimization can be ensured by data mining

This model supports event driven interoperability in systems as well as provide novel technical assistance for construction of smart factory.[8]

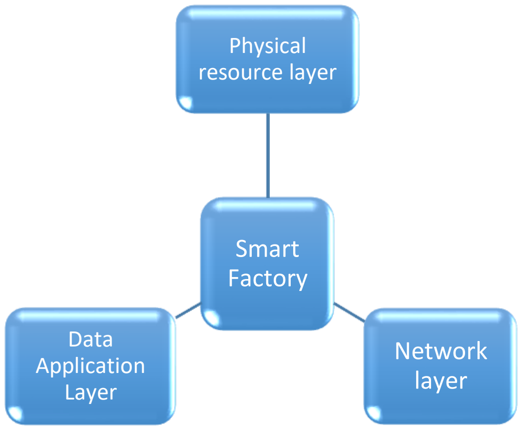


Figure 3-7 Smart Factory

Chapter 4

4 Node-Red

One of the most preferred tool for developing IoT applications is Node-Red, which is visual wiring tool for various hardware, API's and online cloud based services. Flow is the principle way to organize node, and is represented be a tab within editor workspace.[9] It provides a browser based flow-editor and provides various nodes that can be wired together to create instances normally found at following port *http//: localhost:1880*. This event driven non-blocking light-weight runtime is built on Node.Js and its rich text-editor allows creation of JavaScript functions. It is compatible with almost all embedded systems available in market like *Raspberry Pi*, *Beagle bone*, *Arduino* and etc. With repository of over 225,000 modules, it allows us to extend the palette of nodes to new extent.[2, 10]

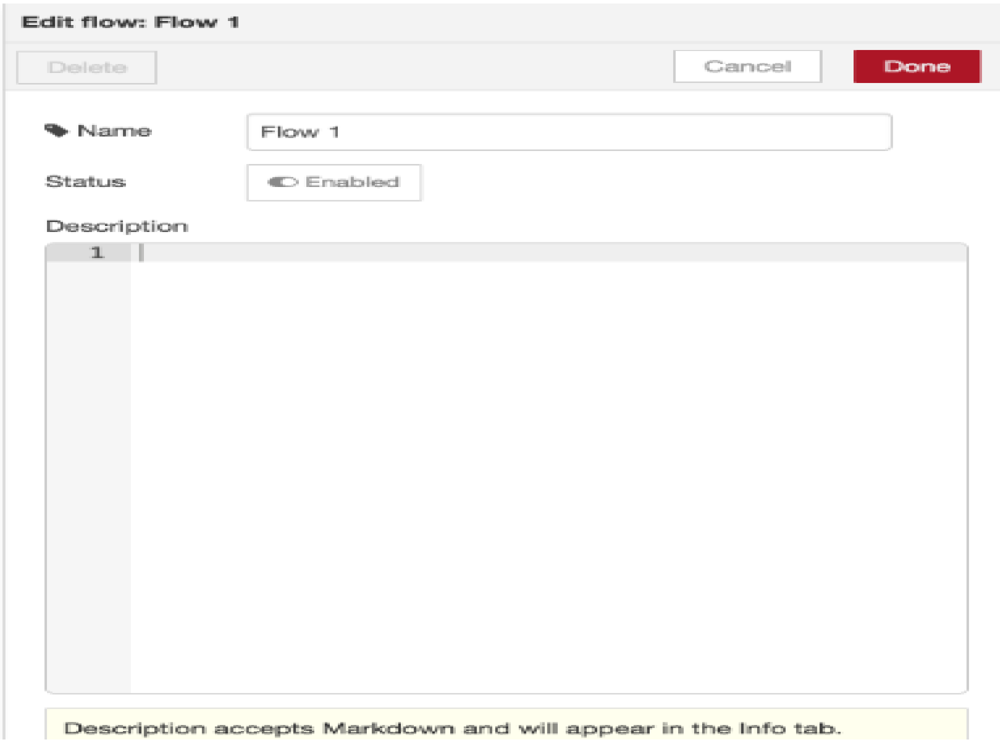


Figure 4-1 Flow editor

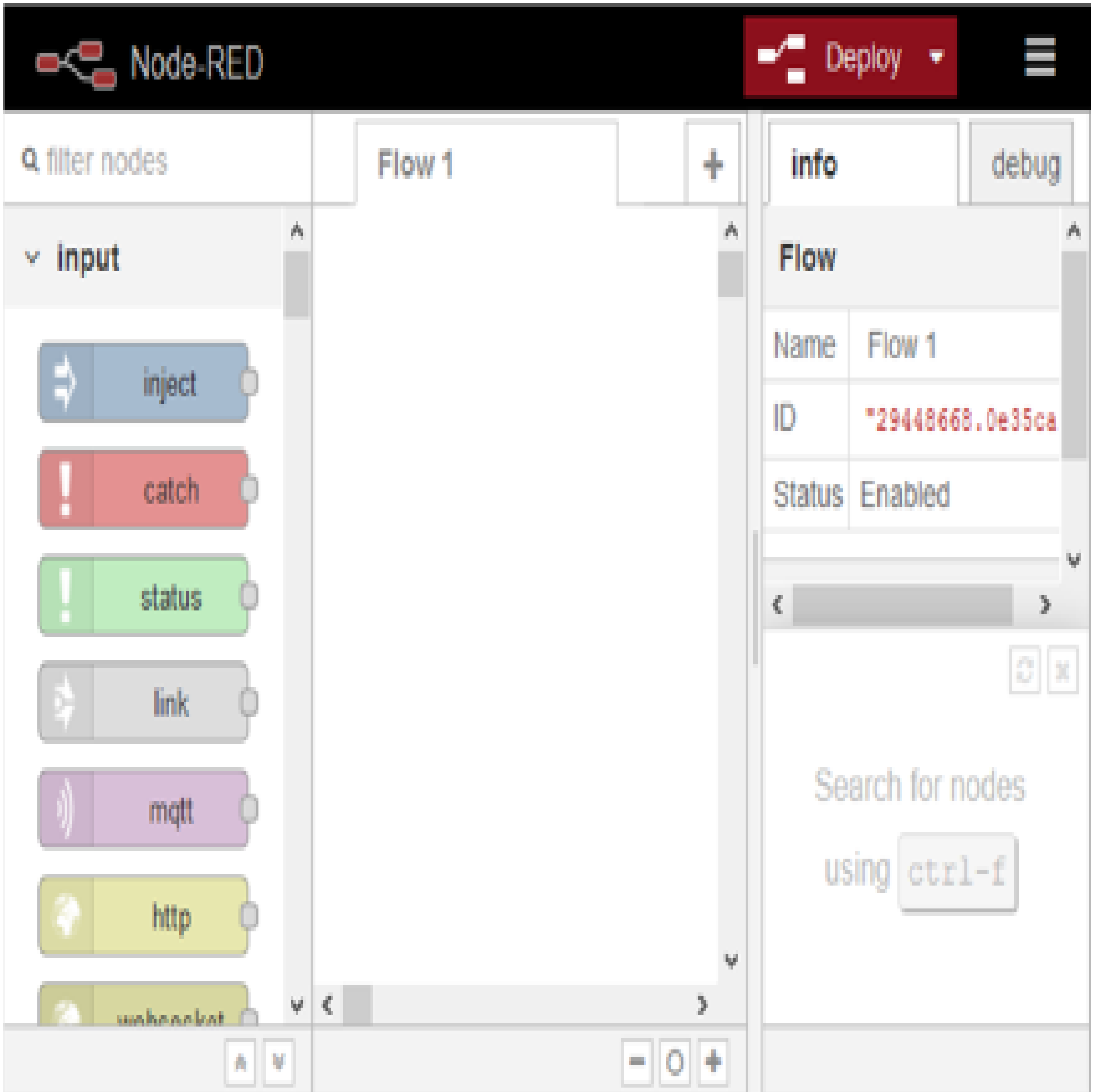


Figure 4-2 Node-red editor

4.1 Node-Red as an IoT tool

Why Node-red is a preferred tool, the table below summarizes the comparison of different open-source visual Programming Languages.

Type of VPL's	Name of VPL's	Programming environment	License	Repository	Platforms supported
Open-source	Node-Red	Web	Open-source Apache 2.0	Github	Raspberry Pi, Beagle-bone black , Arduino, Docker, IBM services, Amazon web services, Android
	NET lab toolkit	Web	GPL	Self	Raspberry Pi, Intel Galileo, Arduino, Azure web services
	Ardublock	Web	GPL	Self	Arduino
	Noodle	Web	NEUL	Self	Arduino, Android

Figure 4-3 open source VPL'S survey

With various sensors libraries available, flows that enable devices to send data, process commands and store data on some database server , can be built.[11]

In Node-Red all users manage a single flow, trying to access it, which may be shown on multiple pages. The user places the flow, once created or after a change, both by saving it to the server and restarting its execution on Node-Red server.[12]

A node, which is a flow's core building block can be activated by incoming message by a previous node, timer or GPIO hardware change, or by waiting for some external event. A special type of such node which carries a reusable configuration that can be shared by all regular nodes, is called Configuration (*config*) node.[9]

Few of other core nodes are listed below:

- Inject
  This node can manually or automatically trigger a flow/flows at prescribed intervals. It carries message that can a *payload* and *topic* properties set

- Debug
  Debug node is used to display message in Debug sidebar within the editor. With output that can be enabled and disabled this node can be configured to send messages to runtime log or to send status text.

- Function
  This node allows, against the messages passed through it, JavaScript code to run.

- Change
  This node enables user to change set contexts and message's properties without utilizing function node.

- Switch
  This node allows routing of message to different branches of flow after assessing set of rules against each message

- Template
  This node is used to create a text using message's properties in order to fill out a template

Palette

A palette contains all combinations of nodes that are available, in an organized way, hence enabling user to filter nodes according to requirement.
A palette manager is used to install new nodes into palette.[9]

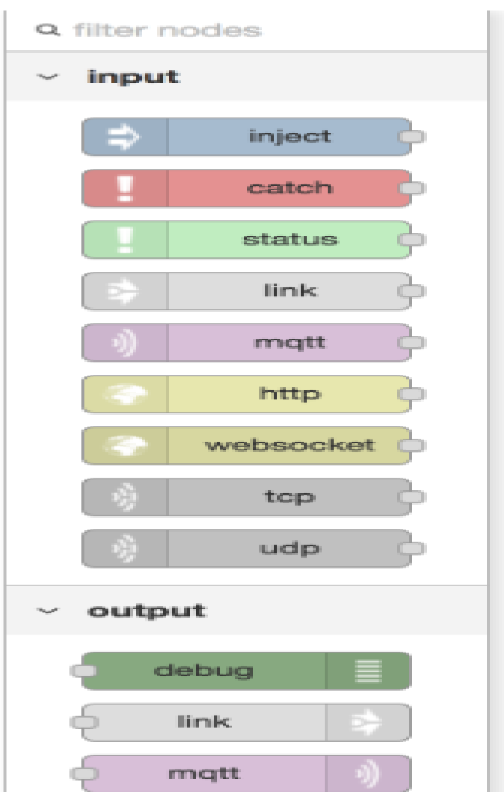


Figure 4-4 Palette

4.2 Applications and examples

Below is an example scenario of bulb being controlled by light intensity with the help of a sensor, and how MQTT subscribes to topic from sensor and publishes it to light on/off topic.[13]

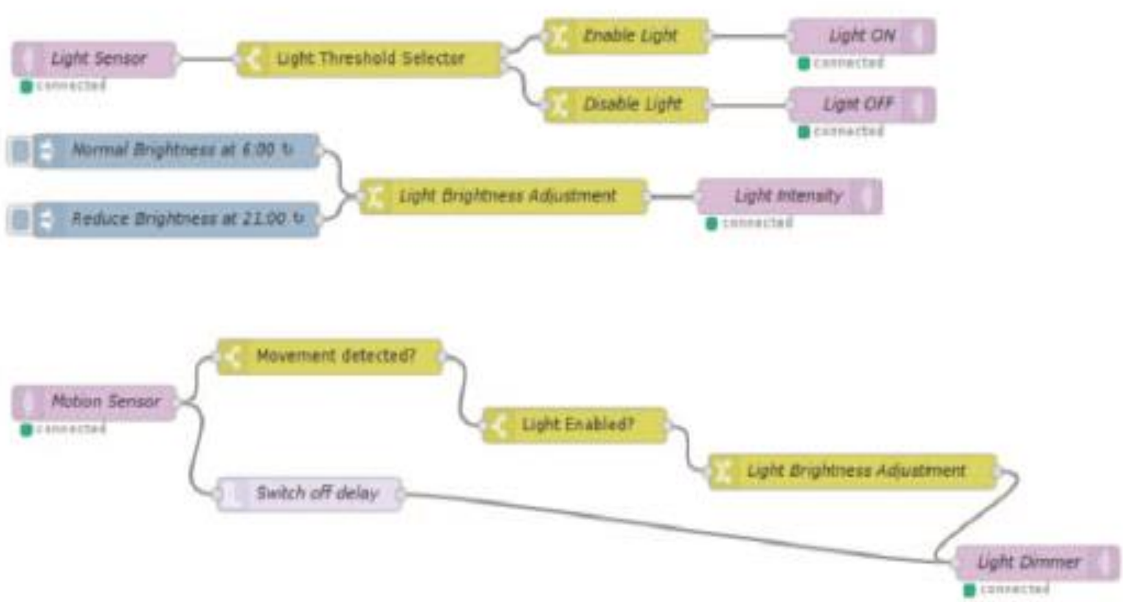


Figure 4-5Node-red: MQTT and hardware communication

Another example is of intrusion check case. Scenario below how ultrasonic sensor is connected to GPIO pin of Raspberry Pi.The pin level is checked on prescribed intervals and thus presence is informed to user through email.[13]

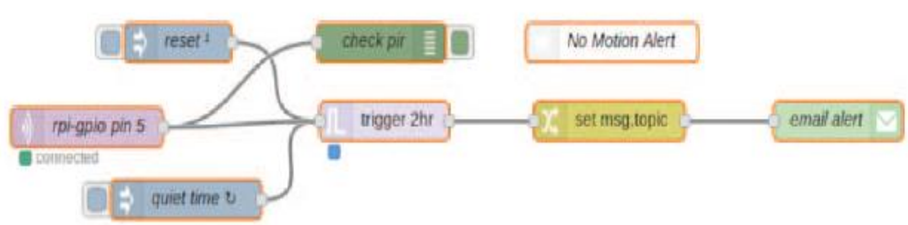


Figure 4-6 Intrusion Check

4.3 MQTT

Since IoT is mostly about wirelessly connecting smart devices, objects and people, therefore it is ideal that there should be a light-weight machine-to-machine (M2M) communication protocol, like MQTT (message queue telemetry transport).
It is open standard protocol that works on publish/subscribe model with minimum possible bandwidth requirement and uses TCP for transport.[14]

Since it is a many-to-many communication protocol its uses a central broker to transfer messages between devices. The pub/sub model that MQTT employs works on the principle model that certain devices or components (*subscribers)* want to *subscribe* to certain information register. Components that want to produce any data, do so by *publishing it* and therefore are called *publishers.[*15*]*, and this overall process of coordination between both parties is handled by a *broker*.

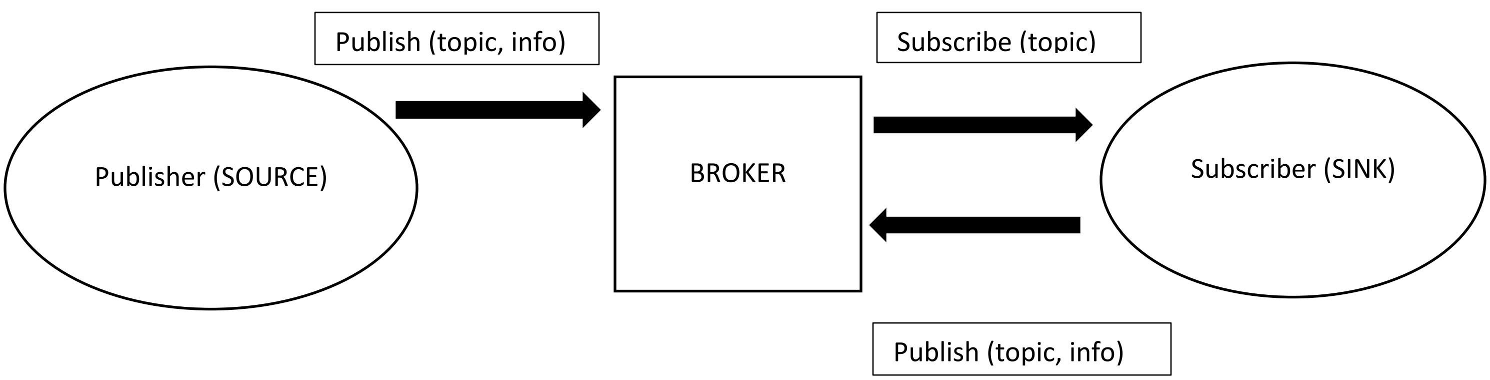


Figure 4-7 pub/sub model of MQTT

4.3.1 Quality of Service (QoS) levels of MQTT and Message format

MQTT offers three levels of Quality of Service (level 0, level 1, and level 2) in order to maintain authenticity & reliability of data. The levels of quality of service are summarized below in table.

QoS level	Meaning
level 0	message is delivered at most once, with no receive acknowledgment
level 1	One delivery at least with a confirmation of receiving
Level 2	Highest level of QoS , where delivery is ensured by four way handshake mechanism

Figure 4-8 QoS LEVELS [15]

Message format consists of fixed header of two bytes, where first one contains message type and flag fields whereas second byte contains variable header and payload.

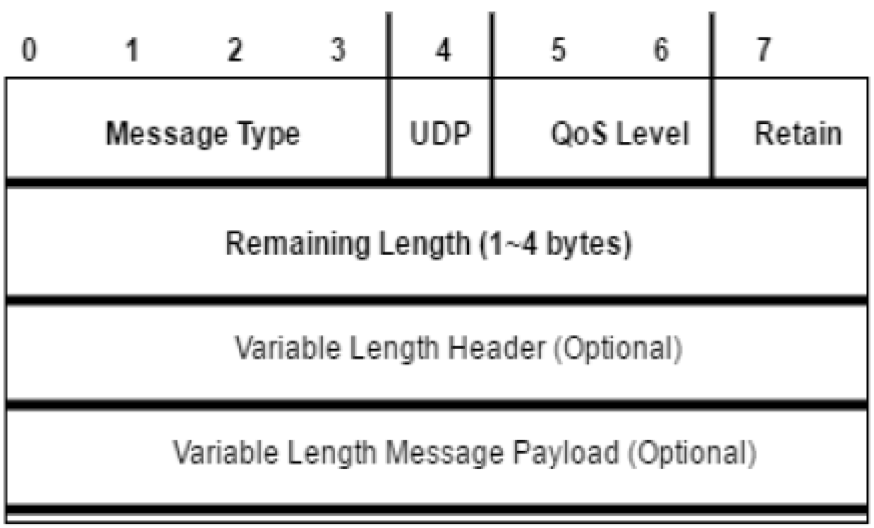


Figure 4-9MQTT message format[14]

4.3.2 MQTT and Internet of Things applications

Although various protocols are available for IoT applications like CoAP, HTTP and AMPQ one must be wondering why MQTT is most preferred protocol for IoT applications. Various researchers have conducted tests under controlled environment via common middleware have drawn convincing results.[14, 16, 17]
Mqtt is considered efficient in terms of latency rate and smaller overhead as compared to HTTP that has an additional data overhead. In poor quality and unstable environments it is best suited because of lighter-weight service and appropriation for constrained edge nodes.[14, 16]
Since it is TCP based it has higher delivery percentages under varying packet loss constraints and under high latency environments, it has minimalist packet loss as compared to other protocols.[14, 16, 17].
However the suitability of MQTT is still dependent upon type of environment and application for which it is being used.

Below schematic picture is an example of MQTT being employed for an IOT application case.

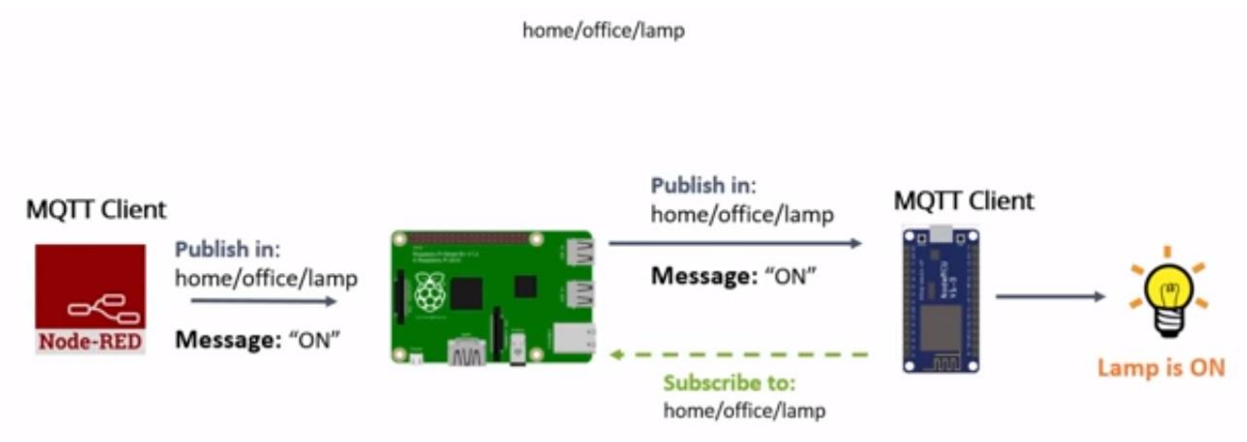


Figure 4-10 LAMP ON/OFF case using Node-red, MQTT and Raspberry Pi.

Chapter 5

5 Raspberry Pi

When all other things are decided as per the design of application, one must choose the suitable development platform for the IoT application, and one of them is *Raspberry pi B+* .It is a complete processor module with 64bit quad core processor, ease of cost and many other features, that will be discussed how they are suitable for IoT. The credit-card sized minicomputer runs a Linux based open source operation system *Raspbian,* with some non-Linux based alternatives also available. Below is the specification summary of Raspberry Pi 3 B+.

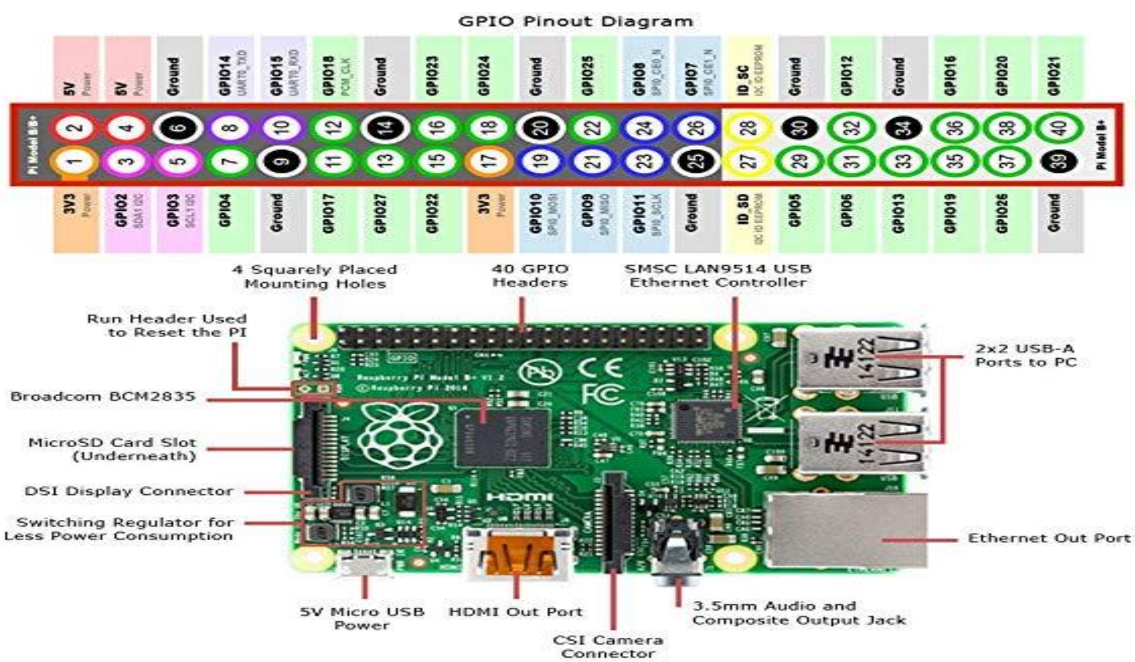


Figure 5-1 Raspberry Pi 3 B+[18]

5.1 Specifications:

Processor: Broadcom BCM2837B0, Cortex –A53 64bit SoC[19] @ 1.4 GHz

Memory: 1GB LPDDR2 SDRAM

Access: 40-pin GPIO header

Connectivity: 2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac wireless

SD Support: Micro SD format for loading OS and data storage

Input power: 5V/2.5A DC via micro USB connector

Multimedia: H.264, MPEG-4 decode; H.264 encode
[18]

5.2 Why Raspberry Pi?

When prototyping an IoT application, one must consider 3 parameters while selecting the suitable development board, 1) Specifications 2) Open hardware 3) Open API.
Specification play an important role like processor/microcontroller, connectivity, clock speed, DAC/ADC, GPIO pins and size plus price.
Open API is preferred to so that creativity and ideas don't get stuck or locked-up, open standards, libraries and community support improve efficiency of one's work.
Whereas open hardware helps in customizing the product according to market needs and help in easy deployment of solution[20]
Computational ability and operating power, USB ports to connect peripherals, OS support and high definition multimedia support , all a

play vital role while comparing the Raspberry Pi with other available board in market.[19]
To give a broad view in terms of comparison, let us compare Raspberry Pi with other boards available in market.

Name	Processor	Power	Lan (Mbit)	Ram	Wi-Fi
Raspberry Pi 3	BCM 2837	5V usb	Up to 300Mbps	1gb	IEEE 802.11 b/g/n
Arduino	AtmegaA8,Atmega168 Atmega1280	7-12V usb	10/100	16-32kb	IEEE 802.11 b/g/n
Beagle Bone black	AM335x 1Ghz ARM Cortex A8	5V	10/100	512	IEEE 802.11 b/g/n
Udoo(quad)	Freescale iMX6Quad, 4x ARM Cortex-A9 core	6-15V	10/100/1000	1Gb	IEEE 802.11 b/g/n

Figure 5-2 Comparison of development board in terms of 5 specifications

Although there are many other requirements for which other features of these boards are also vital and needed to be thoroughly considered but for the sake of simplicity I have compared only these specs.

5.3 Operating system and programming language

Irrespective of application's hierarchy, smart devices need to be programmed either using an Operating System or some higher level of abstraction.
Preferred operating systems for Raspberry Pi are based on Linux (Raspbian, Debian, Arch Linux, and Fedora Remix which are free and designed to run on Raspberry Pi's ARM processor.
The primary reason for using Raspbian is that it comes pre-installed, optimized for on-chip floating point calculations ,widespread community support and is easy to use since it's environment is similar to that of Windows and Mac.[18, 19]

Chapter 6

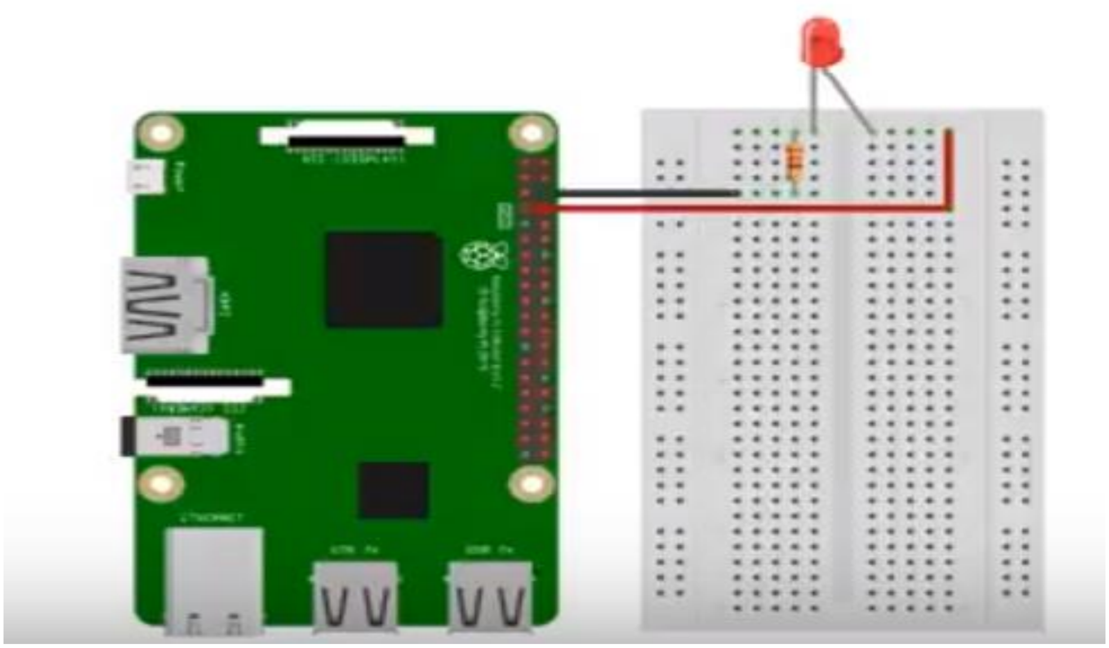
6 Proposed Solution

With above given detailed information about embedded systems, Node-Red and MQTT one can easily comprehend that this thesis is heading towards the convergence of all three.

6.1 Hardware design

Connection our device/component with Raspberry Pi 3 B+ with the help of GPIO pins of the controller. The connections are shown with the help of simulator snapshots.

GND and GP17 pins of the Pi are used.



Then once the Node-Red has been installed in raspberry Pi we will access our Board

6.2 Installation of Node-Red in Raspberry Pi

Since node red acts as our server in our design we will have to install node-red on Raspberry Pi 3 B+. Since we are using *Raspbian Jessie* OS, Node-red is included in it.So first we will login with the help of *putty* using our login, password and ip of Raspberry Pi and then start *vncserver*.

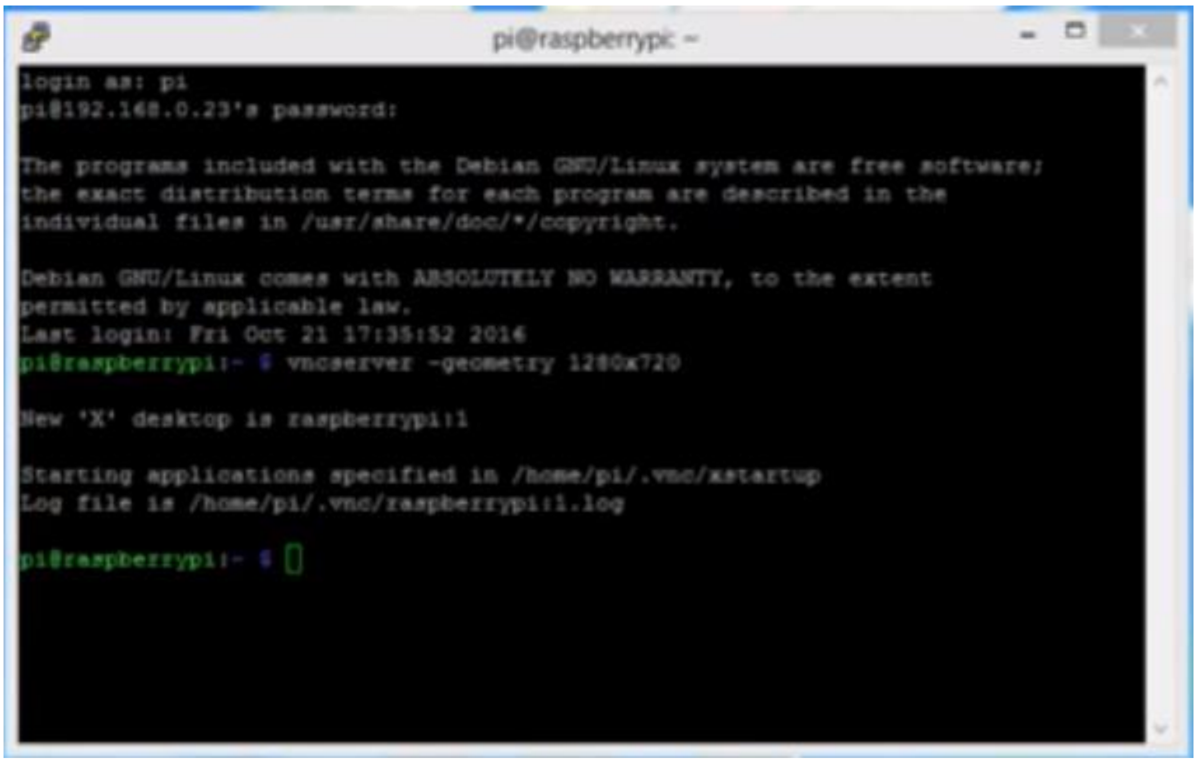


Figure 6-1 Accessing Raspberry Pi

In order to access node-red from external browser one must enter http://"ip-number":1880

Since there are already too many nodes available in palette, but we need ubidots nodes also so in order to install ubidots package we will go to 'Manage Palette' section and search for '*ubidots-nodered*' in search bar and install them as shown in figure [6-3].

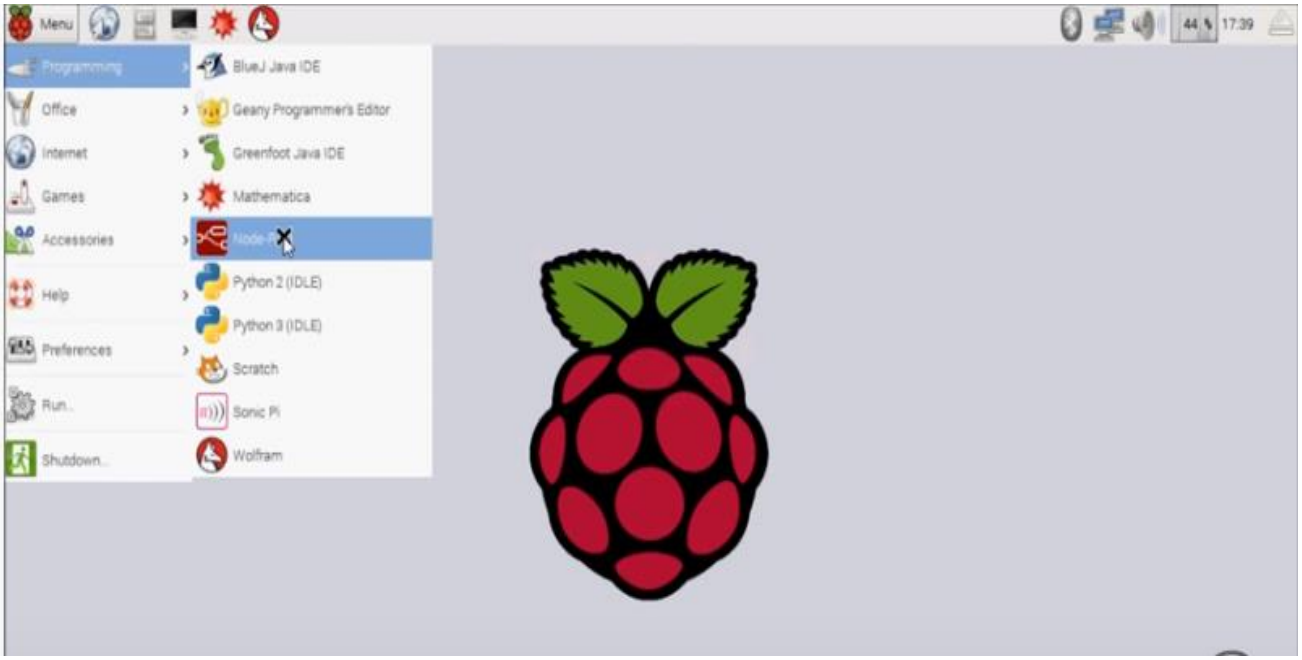


Figure 6-2 Selecting Node-Red from Raspbian Jesse

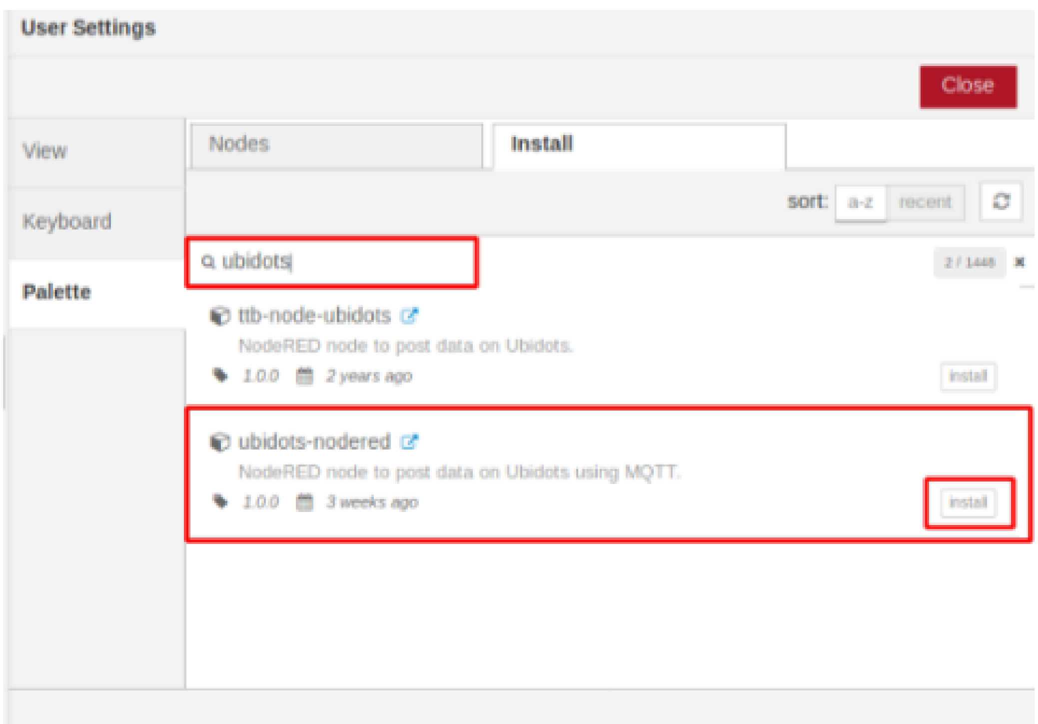


Figure 6-3 Installing Ubidots nodes

Now with Ubidots nodes already in palette we will install node "node-red-contrib-sensors-" which responsible for connected sensor's readings.

For the sake of this project I have signed up for Ubidots educational purpose account. (*app.ubidots.com)*

6.3 Setting up the flow and Nodes

Now since the idea is to control things from anywhere over the internet so we have controlled an LED.

Since we have already installed Ubidots nodes, we will be using Ubidots built-in MQTT broker to communicate with our MQTT clients (In our case it is ***RaspberryPiUbidots***). Design flow can be seen in the figure below

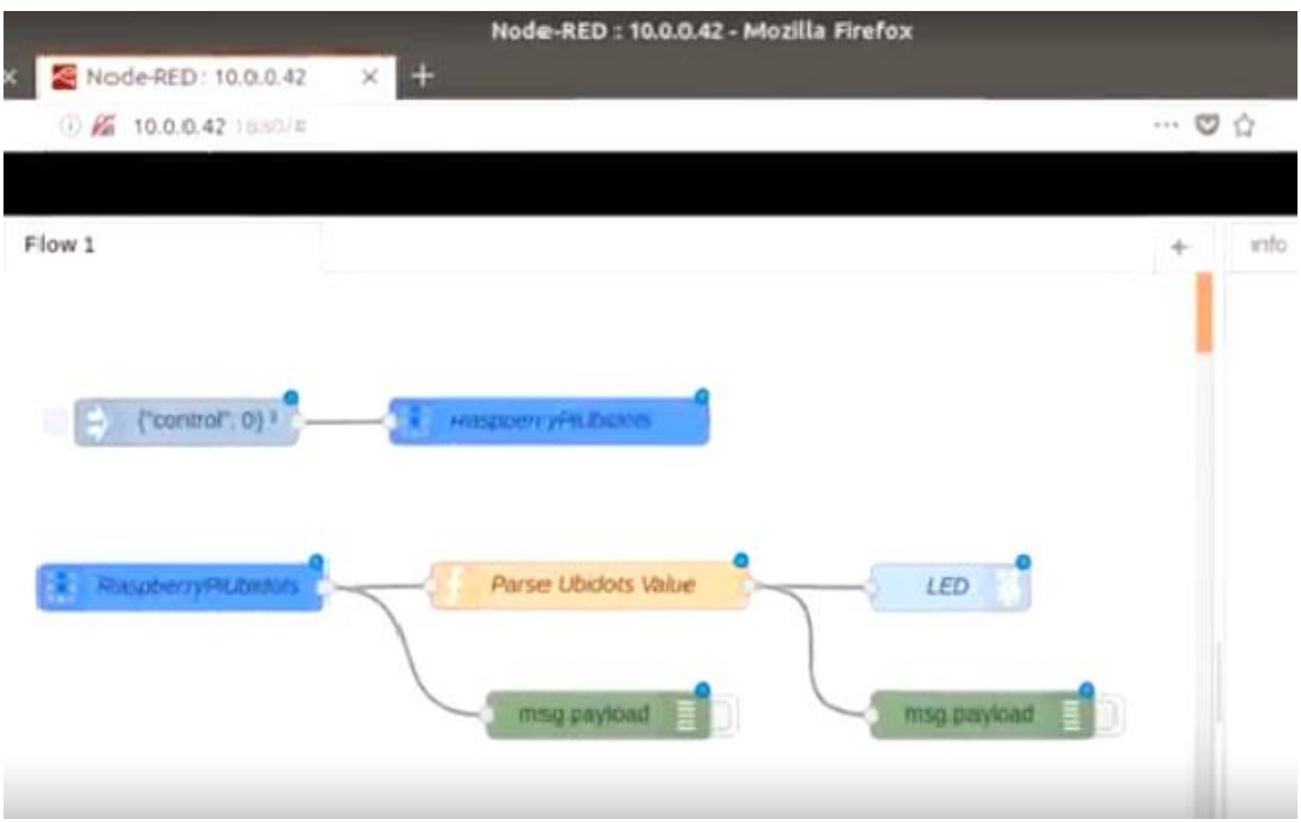


Figure 6-4 Node-Red flow

Then we will copy API credentials from Ubidots account(as shown in fig 6-5) and paste them into our clients (RaspberryPiUbidots) In and Out nodes *Token* space as shown in (Fig 6-6.)

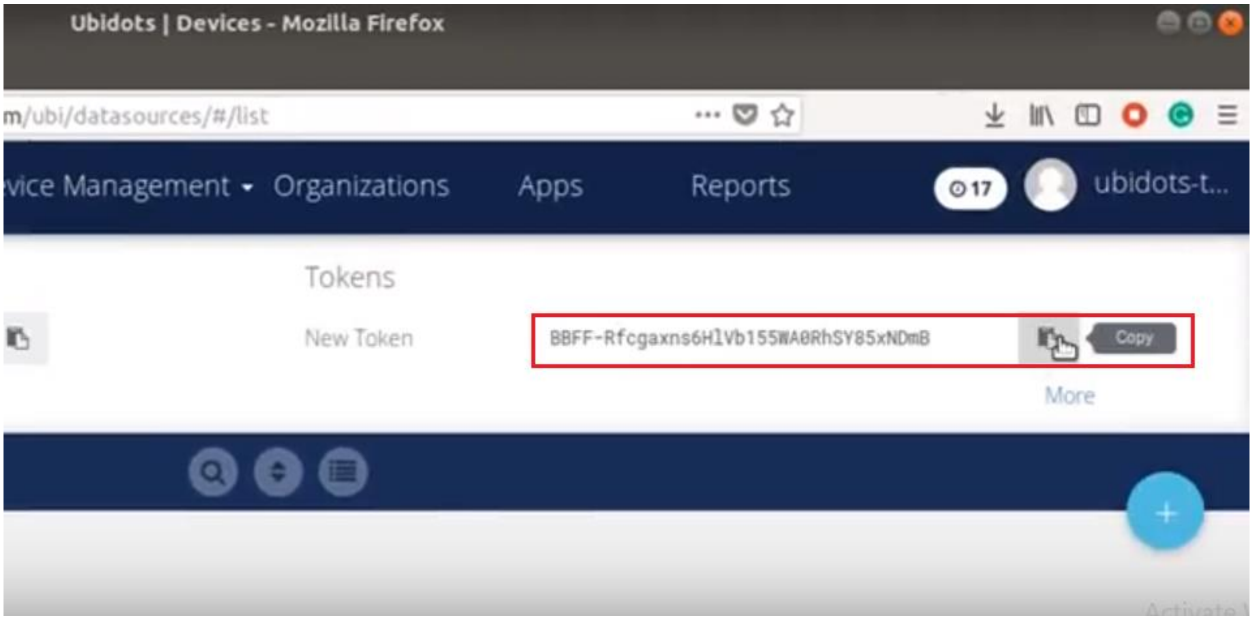


Figure 6-5 API credentials from Ubidots

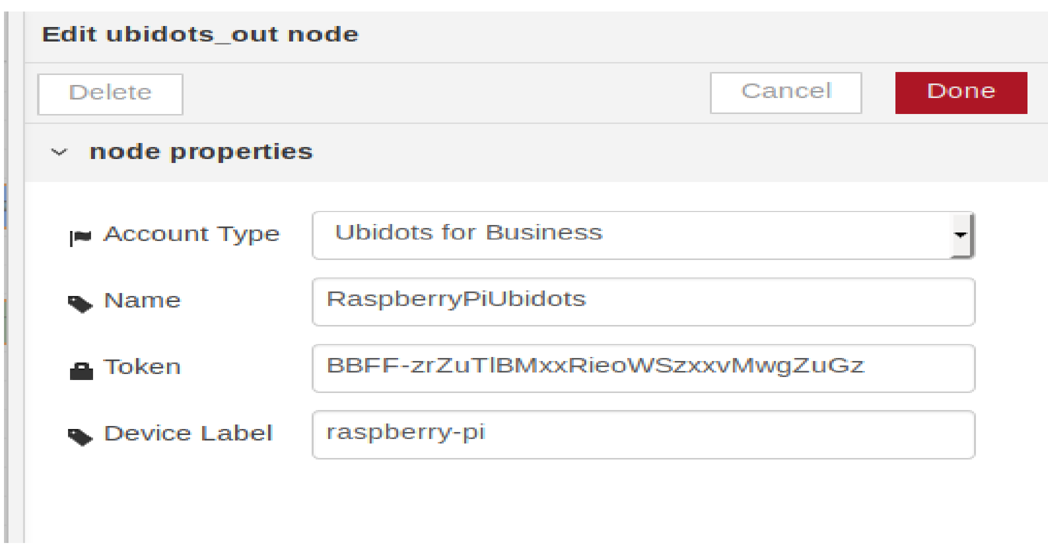
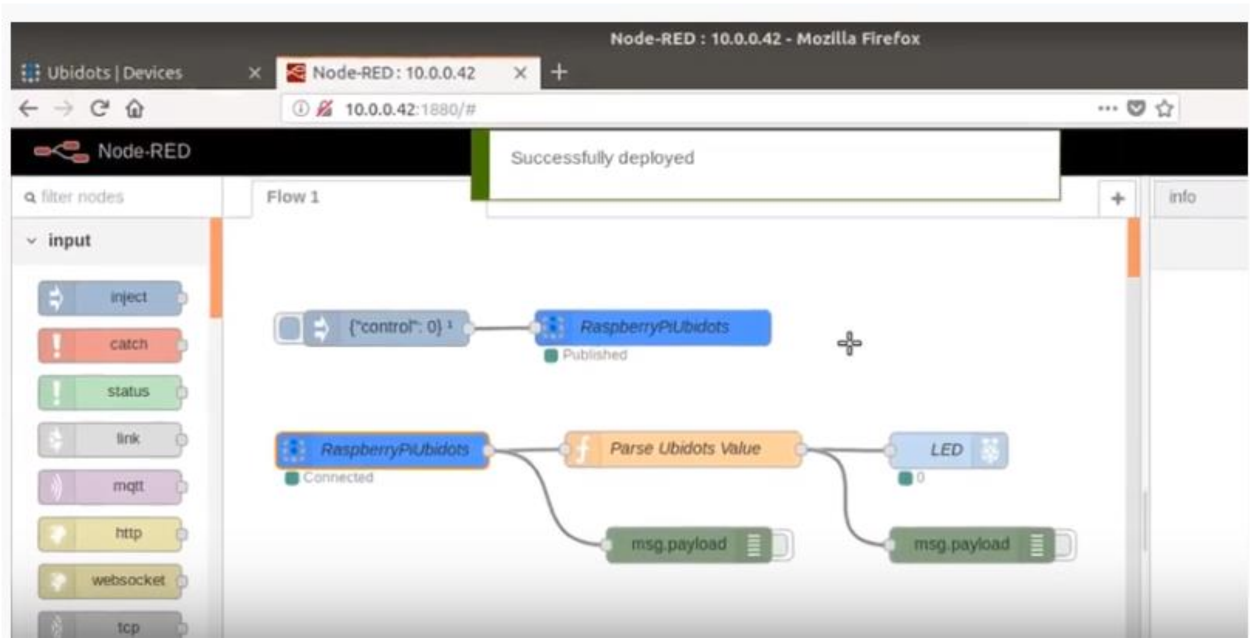


Figure 6-6 configuring Ubidots Token

6.4 Deploying the flow

After this has been done its time to deploy the flows by simply pressing ***DEPLOY*** button



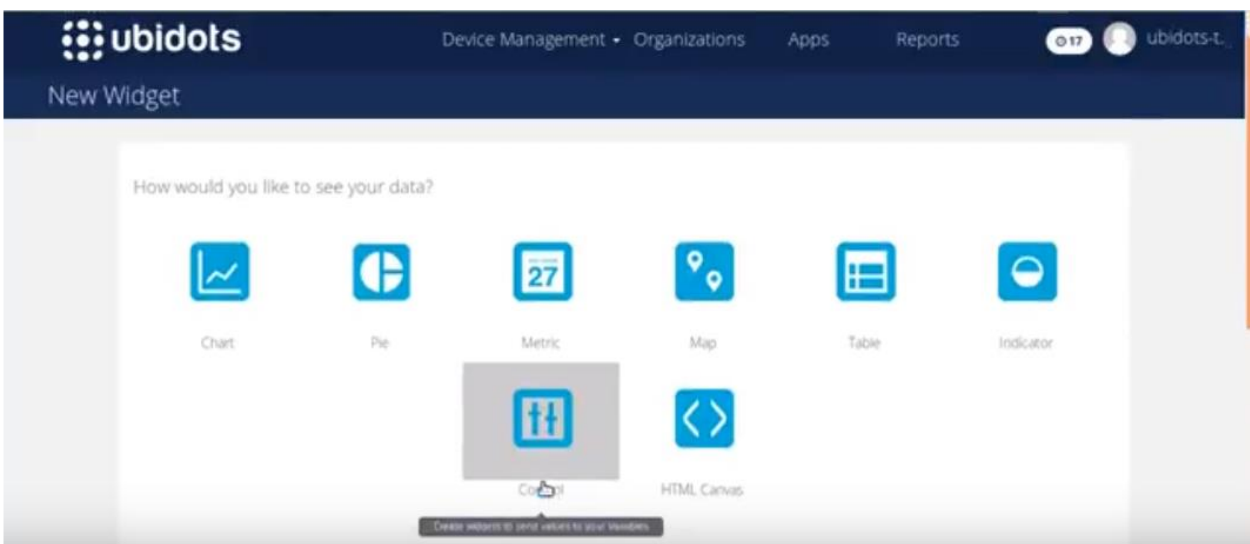


Figure 6-7 Selecting our device from 'Control'

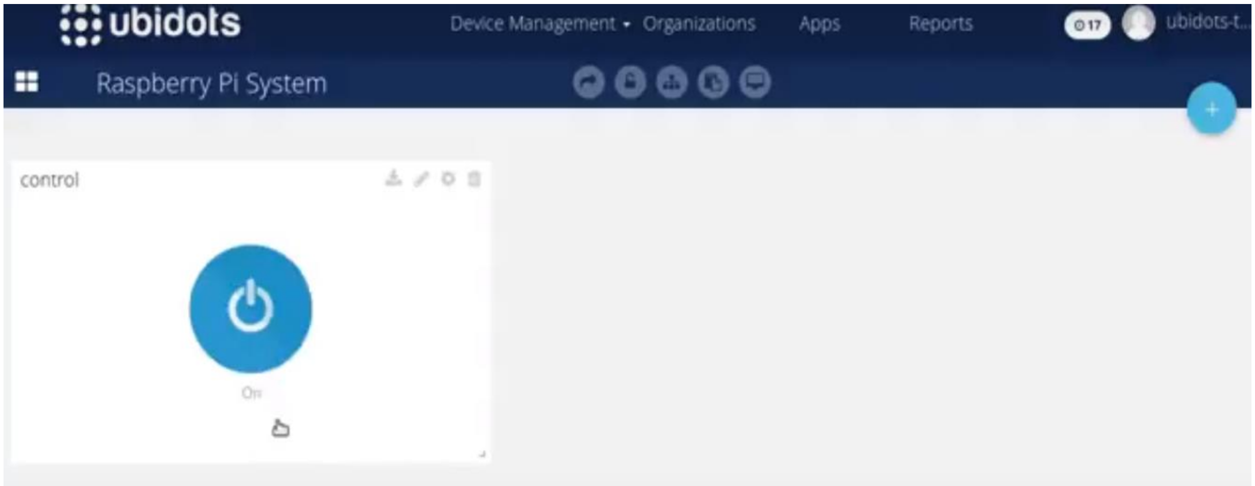


Figure 6-8 Switch widget successfully created

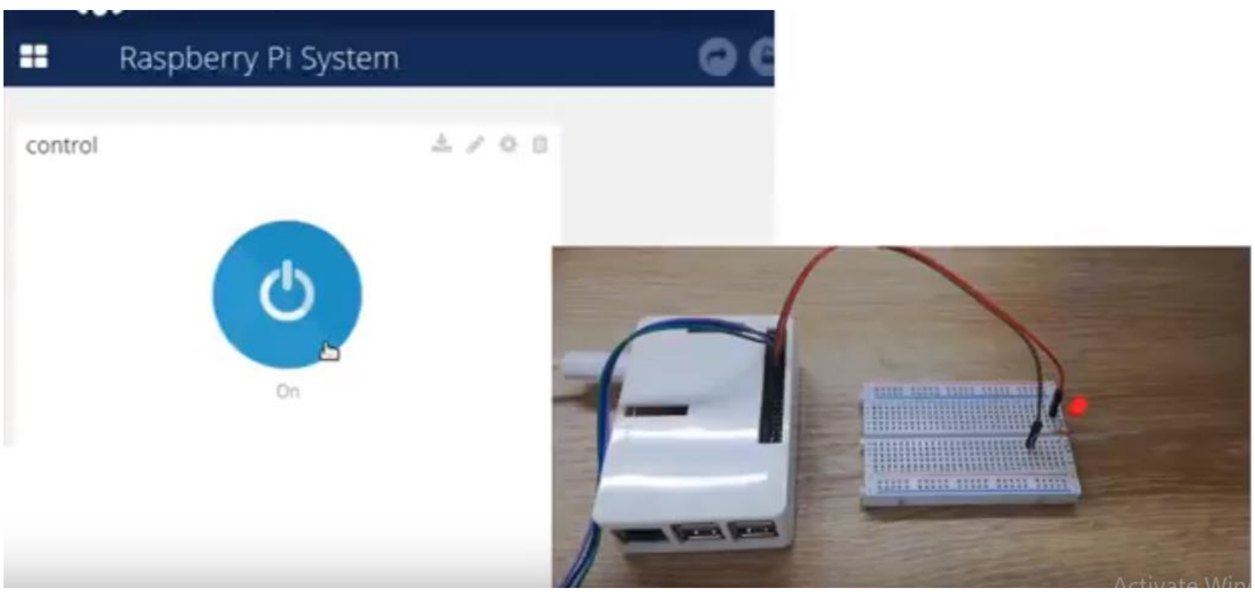


Figure 6-9 Controlling device

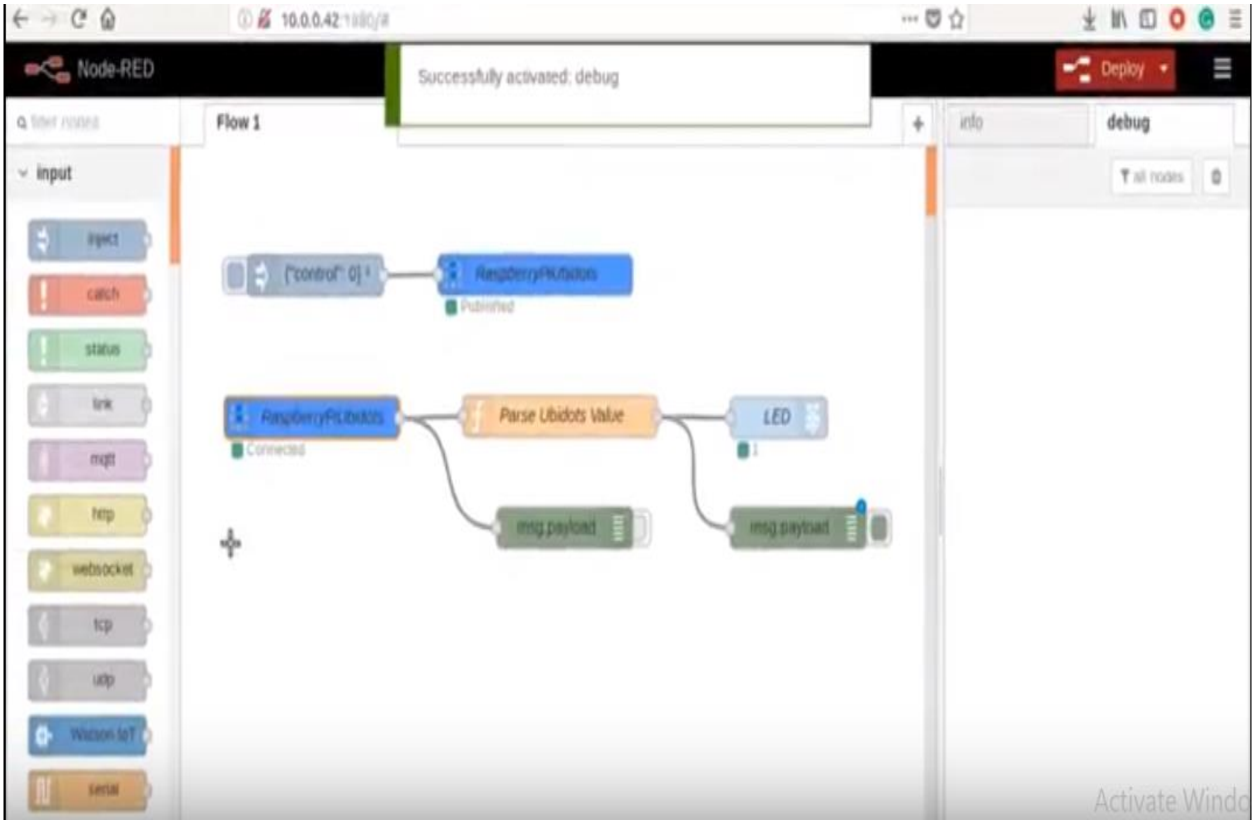


Figure 6-10 Debugged

References

1. Attitalla, S., V. Chocksi, and M. Potdar, *IBM Cloud Solutions for Home Automation.* International Journal of Computer Applications, 2017. **164**(4).
2. Chatterjee, J. and T. Debnath, *Environmental Monitoring Using Sense HAT based on IBM Watson IoT Platform.* 2018.
3. Tabaa, M., et al., *Industrial Communication based on Modbus and Node-RED.* Procedia computer science, 2018. **130**: p. 583-588.
4. Lin, S., et al., *Architecture alignment and interoperability.* An industrial internet consortium and plattform industrie 4.0 joint whitepaper, 2017: p. 1-17.
5. BURKE, T.J., *OPC-UA: Interoperability for Indusrie 4.0 and Internet of Things.* OPC-FOUNDATION.
6. Moghaddam, M., et al. *Next-generation enterprise architectures: Common vernacular and evolution towards service-orientation*. in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. 2017. IEEE.
7. 4.O, P.I., *RAMI 4.0 - A Reference framework for digitalisation*, P.I. 4.O, Editor.: Berlin, Germany.
8. Chen, B., et al., *Smart factory of industry 4.0: Key technologies, application case, and challenges.* IEEE Access, 2017. **6**: p. 6505-6519.
9. *https://nodered.org*. Available from: https://nodered.org.
10. Ray, P.P., *A survey of Visual Programming languages in Internet Of Things.* Hindawi, 2017. **2017**(Scientific Programming Issue 2017): p. 6.
11. Lekić, M. and G. Gardašević. *IoT sensor integration to Node-RED platform*. in *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*. 2018. IEEE.
12. Blackstock, M. and R. Lea. *Toward a distributed data flow platform for the web of things (distributed node-red)*. in *Proceedings of the 5th International Workshop on Web of Things*. 2014. ACM.
13. Rajalakshmi, A. and H. Shahnasser. *Internet of Things using Node-Red and alexa*. in *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*. 2017. IEEE.
14. Yassein, M.B., et al. *Internet of Things: Survey and open issues of MQTT protocol*. in *2017 International Conference on Engineering & MIS (ICEMIS)*. 2017. IEEE.
15. Hunkeler, U., H.L. Truong, and A. Stanford-Clark. *MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks*. in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. 2008. IEEE.
16. Shaout, A. and B. Crispin, *Using the MQTT protocol in real time for synchronizing IoT device state.* Int. Arab J. Inf. Technol., 2018. **15**(3A): p. 515-521.
17. Thangavel, D., et al. *Performance evaluation of MQTT and CoAP via a common middleware*. in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2014. IEEE.
18. *https://www.raspberrypi.org/products*. Available from: https://www.raspberrypi.org.
19. Maksimović, M., et al., *Raspberry Pi as Internet of things hardware: performances and constraints.* design issues, 2014. **3**(8).
20. Singh, K.J. and D.S. Kapoor, *Create Your Own Internet of Things: A survey of IoT platforms.* IEEE Consumer Electronics Magazine, 2017. **6**(2): p. 57-68.