



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electronique
Electronique des systèmes embarqués

Réf. : Entrez la référence du document

Présenté et soutenu par :
DJEMA Noureddine

Le : **dimanche 24 juin 2018**

Security and home automation based on Raspberry Pi, Arduino Uno and Node MCU

Jury :

M.	HEZABRA Adel	MAA	Université Mohamed Khider	Président
M.	BENELMIR Okba	MCB	Université Mohamed Khider	Rapporteur
M.	RAHMANI Nacereddine	MAA	Université Mohamed Khider	Examineur

Dedication

I dedicate this project to the soul of my parents, God bless them and to my wife for here sacrifice

And she was standing by me, giving me the courage to go forward for three long years of education. To reach where I am now Thanks a lot darling, and to my both kids Zakaria, Mohcen and my little girl Maram.

I don't want forget all my friends and all my family brothers, sisters and their sons, daughters

Acknowledgement

First, I would like to thank the Almighty God for granting me life and good health as well as supreme protection and guidance throughout my studies.

I am thankful to my supervisor, BENELMIR Okba for his informative and useful guidance and suggestions throughout the journey for project design.

I am very grateful to the Department of Electrical and Information Engineering and my lecturers for instilling in me the knowledge that has brought me this far.

I am thankful to my friends and classmates who have contributed to the success of my studies through their wise counsel and support.

Needless to mention, this project would not have been complete without reference to and inspiration from the work of others whose details are included in the reference section of this document.

TABLES LIST

Tab.II.1: Arduino Products	26
Tab.II.2: Arduino Instructions & Command	33
Tab.II.3: ESP8266 Parameters	37
Tab. II.4: ESP-12E Pin Description	38
Tab.II.5: ESP-12E Interface	41

FIGURES LIST

Fig.I.1: Automation System Using (IoT)	5
Fig.I.2: Smart Home & Automation	8
Fig.1.3: Screen of Home Automation on smart phone	10
Fig.II.1: Raspberry Pi Components	16
Fig.II.2: Raspberry Pi 3 Model B+	17
Fig.II.3: GPIO Pins Description of Raspberry Pi	20
Fig.II.4: PuTTY	22
Fig.II.5: Snapshot of Configuration Screen for Raspberry Pi	23
Fig.II.6: Desktop for Raspberry Pi	24
Fig.II.7: Arduino Boards	25
Fig.II.8: Arduino Uno Board	27
Fig. II.9: Arduino Uno Board Pin Description	28
Fig. II.10: Arduino IDE	30
Fig.II.11: IDE Toolbar Description	31
Fig.II.12: IDE Window Status	31
Fig.II.13: IDE Serial Monitor Window	32
Fig.II.14: ESP8266 Boards	33
Fig.II.15: ESP8266EX Block Diagram	35
Fig.II.16: ESP-12E Description	38
Fig.II.17: ESP-12E Packaging and Dimension	39
Fig. II.18: Node MCU V1.0 (LoLin)	42
Fig. II.19: Node MCU V1.0 Pin Description	43
Fig.II.20: URLs field in the Arduino V1.6+ preferences	44
Fig.II.21: Board Manager from Arduino IDE	45
Fig.II.22: Selection of Generic ESP8266 Module	45
Fig.II.23: Selection of baud rate 115200	46

Fig.III.1: Project Electric Schematic	48
Fig.III.2: MQ2 Smoke Sensor	49
Fig.III.3: MQ9 Carbon Monoxide Sensor (CO)	49
Fig.III.4: Flame Sensor	50
Fig.III.5: Light Dependent Resistor (LDR)	50
Fig.III.6: Four Channel Relayboard	50
Fig.III.7: USB Camera	52
Fig.III.8: Buzzer	52
Fig.III.9: Fan	53
Fig.III.10: Led	53
Fig.III.11: Breadboard	54
Fig.III.12: Ethernet Network	54
Fig.III.13: Mosquitto Broker	55
Fig.III.14: Testing the Mosquitto	56
Fig.III.15: Node-RED	57
Fig.III.16: Node-RED Description	58
Fig.III.17: Node-RED Dashboard	59
Fig.III.18: Dashboard Nodes	60
Fig.III.19: snapshot of Security & Home Automation	62
Fig.III.20: snapshot of Home Appliances from Node-RED	63
Fig.III.21: snapshot of Home Appliances from the browser	64
Fig.III.22: snapshot of Home Appliances from my laptop	65
Fig.III.23: Chart Flow for gas leak & smoke	66
Fig.III.24: Chart Flow for Detecting a Flame	67
Fig.III.25: Chart Flow for Detecting Carbon Monoxide CO	68
Fig.III.26: Chart Flow for Detecting Light	69

ABBREVIATION LIST

IoT	Internet of Things
GPIO	General Purpose Input Output
MQTT	Message Queuing Telemetry Transport
IP	Internet Protocol
HAS	Home Automation System
DHCP	Dynamic Host Connection Protocol
M2M	Machine to Machine
WHAS	Wireless Home Automation System
HAS	Home Automation System
AP	Access Point
ETH	Ethernet
GUI	Graphic User Interface
HDMI	High-Definition Multimedia Interface
ISP	Internet Service Provider
LED	Led Emiting Diode
OS	Operating System
ROS	Raspbian Operating System
SoC	System on Chip
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
USB	Universal Serial Bus
Wi-Fi	Wireless Internet Free Internet
WLAN	Wireless Local Area Network

ABSTRACT

With advancement of Automation technology, life is getting simpler and easier in all aspects. In today's world Automatic systems are being preferred over manual system. With the rapid increase in the number of users of internet over the past decade has made Internet a part and parcel of life, and IoT is the latest and emerging internet technology.

Internet of things is a growing network of everyday object-from industrial machine to consumer goods that can share information and complete tasks while you are busy with other activities. Wireless Home Automation system(WHAS) using IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through internet from anywhere around the world, an automated home is sometimes called a smart home. It is meant to save the electric power and human energy. The home automation system differs from other system by allowing the user to operate the system from anywhere around the world through internet connection.

In this project we present a Home Automation system (HAS) using Raspberry Pi 3, Arduino Uno and Node MCU wireless communication, to provide the user with remote control of various lights, fans, and appliances within their home and Security application. The system will automatically change on the basis of sensors'. This system is designed to be low cost and expandable allowing a variety of devices to be controlled. Key Words: Home automation System (HAS), Internet of Things (IoT), Cloud networking, Wi-Fi network. There's a four application of security gas leak and smoke using MQ2, detecting fire using flame sensor, detecting carbon monoxide using MQ9, detecting light using LDR and surveillance camera using webcam with Raspberry Pi 3 live streaming on the cloud. I faced to many problems, I bought too many components, I would like to use LCD and to let the webcam sends emails when detect motion, but not enough time left.

في أواخر القرن العشرين، تقدمت الإلكترونيك تقدماً باهراً فأعطت طفرة تقدمية لكل مجالات الحياة البشرية ومن بينها تكنولوجيا إنترنت الأشياء و بهذا التطور أصبحت الحياة أكثر بساطة وسهولة في جميع الجوانب. في العالم اليوم، يتم تفضيل الأنظمة التلقائية على النظام اليدوي. مع الزيادة السريعة في عدد مستخدمي الإنترنت على مدى العقد الماضي جعلت الإنترنت جزءاً لا يتجزأ من الحياة اليومية للإنسان وكانت إنترنت الأشياء هي أحدث تقنيات الإنترنت الناشئة.

إن إنترنت الأشياء عبارة عن شبكة متنامية من الأشياء المستخدمة من طرف الإنسان يومياً من بينها الآلات الصناعية أو مواد استهلاكية التي يمكنها مشاركة المعلومات وإكمال المهام بينما عندما يكون الإنسان مشغولاً بأنشطة أخرى. نظام رقمه المنزل اللاسلكي (WHAS) باستخدام IoT هو نظام يستخدم أجهزة الكمبيوتر أو الأجهزة المحمولة للتحكم في الوظائف والميزات المنزلية الأساسية تلقائياً من خلال الإنترنت من أي مكان حول العالم ويطلق على المنزل ذو التحكم التلقائي أحياناً اسم المنزل الذكي. الغرض منه هو توفير الطاقة الكهربائية والحياة الميسورة للإنسان. يختلف نظام التشغيل الآلي للمنزل عن أي نظام آخر لأنه يسمح للمستخدم بتشغيل النظام من أي مكان حول العالم من خلال الاتصال بالإنترنت.

في هذا المشروع ، نقدم نظام رقمه المنزل (HAS) باستخدام بطاقة Raspberry Pi 3 ، و Arduino Uno ، و Node MCU للاتصالات اللاسلكية ، لتزويد المستخدم بجهاز التحكم عن بعد مي مختلف الأجهزة و الآلات المنزلية الكهربائية مثل الأضواء ، المراوح ، والأجهزة المنزلية بصفة آمنة. ويتغير النظام تلقائياً على أساس أجهزة الاستشعار. تم تصميم هذا النظام ليكون منخفض التكلفة وقابل للتوسيع مما يسمح بالتحكم في مجموعة متنوعة من الأجهزة. الكلمات المفتاح نظام التشغيل الآلي للمنزل (HAS) ، إنترنت الأشياء (IoT) ، الشبكات العنكبوتية ، شبكة Wi-Fi. هناك أربعة تطبيقات التطبيق الأول هو لتسرب الغاز والدخان باستخدام MQ2 ، واكتشاف الحريق باستخدام جهاز استشعار اللهب ، وكشف أول أكسيد الكربون باستخدام MQ9 ، والكشف عن الضوء باستخدام LDR وكاميرا المراقبة باستخدام كاميرا الويب مع بث مباشر لـ Raspberry Pi 3 على السحابة. لقد واجهت العديد من المشاكل ، واشترت الكثير من المكونات ، وكنت أود أن استخدم شاشات الكريستال السائل وكذلك جعل الكاميرا ويب ترسل رسائل البريد الإلكتروني للمستخدم عند الكشف عن أي حركة مشتبها بها ، ولكن واجهتني العديد من الصعوبات و أمني أن يكمل الطلبة الجدد ماكنت بدأت به .

Table of Contents

GENERAL INTRODUCTION	1
<u>CHAPTER I: INTERNET OF THINGS (IoT)</u>	
I.1 Introduction	4
I.2 History	4
I.3 Applications of IoT	4
I.3.1 Smart Home (Home Automation)	5
I.3.2 Industrial Automation	5
I.3.3 Security System	5
I.3.4 Agricultural Automation	6
I.3.5 Benefits and Risks	6
I.3.5.1 Benefit	6
I.3.5.2 Risk	7
I.4 Home Automation	7
I.4.1 History	8
I.4.2 Applications and Technologies	9
I.4.3 Implementations	9
I.5 Home Security	9
I.5.1 Definition	10
I.5.2 Smart Home Fire Protection	10
I.5.3 Access Control	11
I.5.4 Artificial Intelligence Programs Protect the Homeowner	11
I.5.5 Protecting Confidential Information	11
I.6 Conclusion	12
<u>CHAPTER II: PROGRAMMABLE DEVICES</u>	
II.1 Introduction	15

II.2 Raspberry Pi - Single Board Computer	15
II.2.1 History	15
II.2.2 Definition of Raspberry Pi	15
II.2.3 Raspberry Pi Components	15
II.2.4 Different Versions of the Raspberry Pi	17
II.2.5 Technical specifications of Raspberry Pi 3 Model B+	17
II.2.6 Raspberry Pi 3 Model B Features	18
II.2.7 General Purpose Input Output on the Raspberry PI (GPIO)	19
II.2.8 Choosing the Operating System (OS)	21
II.2.9 Characteristics of Raspbian Operating System (RoS)	21
II.2.10 Installing Raspbian Operating System (ROS)	21
II.2.11 Definition of PuTTY	21
II.2.12 PuTTY Features	22
II.2.13 Remotely Control Raspberry PI 3 Using VNC Protocol	22
II.3 Arduino Uno	24
II.3.1 History	24
II.3.2 Definition of Arduino	24
II.3.3 Usage of Arduino Boards	25
II.3.4 Family of Arduino Boards	25
II.3.5 Timeline of Arduino Products	26
II.3.6 Arduino Uno Board	26
II.3.7 Technical Specifications	27
II.3.8 Software Requirements	30
II.3.8.1 Components of Arduino IDE	30
II.3.8.2 Arduino Programming Language	32
II.4 ESP8266	33
II.4.1 History	33
II.4.2 Definition	33

II.4.3 Technical Specifications	34
II.4.4 Preambles	34
II.4.5 ESP8266 Features	36
II.4.6 Parameters	37
II.4.7 Pin Descriptions	38
II.4.8 Packaging and Dimension	39
II.4.9 Functional Description	39
II.5 Node MCU V1.0 (ESP8266)	42
II.5.1 Definition of Node MCU V1.0	42
II.5.2 Specifications	42
II.5.3 Pin Definition	43
II.5.4 Using Arduino IDE	43
II.5.5 Setup ESP8266 Support	45
II.6 Conclusion	46
<u>CHAPTER III: HARD AND SOFTWARE MATERIALS</u>	
III.1 Introduction	48
III.2 Electric Schematic	48
III.3 Hardware Materials	49
III.3.1 MQ2 Smoke Sensor	49
III.3.2 MQ9 Carbon Monoxide Sensor (CO)	49
III.3.3 Flame Sensor	50
III.3.4 Light Sensor (LDR)	50
III.3.5 Relayboard	51
III.3.6 USB Camera	51
III.3.7 Buzzer	52
III.3.8 Fan	52
III.3.9 Led	53
III.3.10 Breadboard	53

III.3.11 Access Point	54
III.4 Software Materials	54
III.4.1 MQTT Protocol	54
III.4.1.1 Installing Mosquitto Broker	55
III.4.1.2 Testing the Mosquitto Broker	56
III.4.2 Node-RED	56
III.4.2.1 Installing Node-RED	57
III.4.2.2 Testing the Node-RED	58
III.4.2.3 Node-RED Dashboard	59
III.4.3 Motion	60
III.4.4 Arduino Uno Code	62
III.4.5 Node MCU Code	62
III.5 Results and Discussions	62
III.5.1 Security applications	62
III.6 Conclusion	
GENERAL CONCLUSION	65
Bibliography	67
Appendix	
Appendix A: different versions of raspberry Pi	68
Appendix B: Installing_Rspbian	70
Appendix C: MQ2	72
Appendix D: MQ9	75
Appendix E: Flame Sensor	79
Appendix F: Light dependent resistors	81
Appendix G: Arduino Uno Code	88
Appendix H : Node MCU V1.0 Code	95
Appendix I : How to use several softwares	104

General Introduction

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities. Internet of Things it's huge and vast field occupied of education, industrial, and medical, agricultural. We find IoT anywhere and everywhere because this last decade is all about internet.

The project we dealing with is security and home automation. In Security we deal with sensors (MQ2, MQ9, Flame sensor, LDR), And surveillance using webcam with Raspberry Pi 3 and software called motion need to be installed starting with what am looking for in this project, what am going to release what I need like hardware materials and software materials, and documentation to get the right view of the project using others experience and try to improve in the right way, combining the research and new opining. I need to search a lot about software's like MQTT is light weight protocol M2M, Node-RED is graphical interface reduces programming and easy to manipulate to create appliances like in this project. That was very useful documentation that took me too long to understand and well many things. The Thesis is on Home Automation and Security with aim of monitoring, logging and controlling home appliances. The primary focus is to make safe and secure the home we live in. The same can be done when the user himself enters the room and by virtue of the system he can make arrangements from his doorstep such that as soon as he enters his house he can make himself at full comfort without manually having to switch on the electrical appliances or his favourite T.V. channel for an example. Thus using the same set of sensors the dual problems of home security and home automation can be solved on a complementary basis.

I split the project on three chapters, the first one definition about Internet of Things, Home automation and Home security they blong to each other, or combined together to give us finale product in way of making our life easy and secure. The second chapter about the programmable devices which is Raspberry Pi3, Arduino Uno and Node MCU all are the master pieces of this project, The third chapter is the practice was the very hard part, because I had to find many sensors, other ways I had to buy them on site, when I bought all the materials needed for this project. I started experience with these sensors for a while to get my hand over it, and about softwares I did many researches, documentations, and books, to handle these softwares. By the end that's very good experience in releasing a project about IoT in field of Home Automation.

CHAPTER I

I.1 Introduction

The Internet of Things has made it easier than ever to set up a home in which you can remotely control your door locks, lights, thermostats, vacuums, lawnmowers, and even pet feeders, using your smart phone and an app. It's also made it simple (and relatively affordable) to monitor your home from pretty much anywhere. Smart security systems are highly customizable and available as do-it-yourself kits or as full-blown setups that include professional installation and monitoring. Security system there are plenty of individual devices available that let you monitor your home from anywhere using your phone or tablet, including indoor and outdoor security cameras, video doorbells, and smart locks.

I.2 History

The concept of Internet of Things became popular in 1999, through Auto-ID Centre at MIT and related market-analysis publications. Radio-frequency identification (RFID) was seen by Kevin Ashton (one of the founders for the original Auto-ID center) as a prerequisite for the Internet of things at that point. Ashton prefers the phrase "Internet for things." If all objects and people in daily life were equipped with identifiers, computers could manage and store them. Besides using RFID, the tagging of things may be achieved through such technologies as near field communication , barcodes, QR codes and digital watermarking.[1][2]

I.3 Applications of IoT

The applications for Internet connected devices are extensive and multiple, ranging from household appliances to heavy industrial machinery.

The ability to network embedded devices with limited CPU, memory and power resources means that IoT finds applications in nearly every field. Such systems could be in charge of collecting information. There is no end to the usage of IoT in technology, business and personal lives.[3]

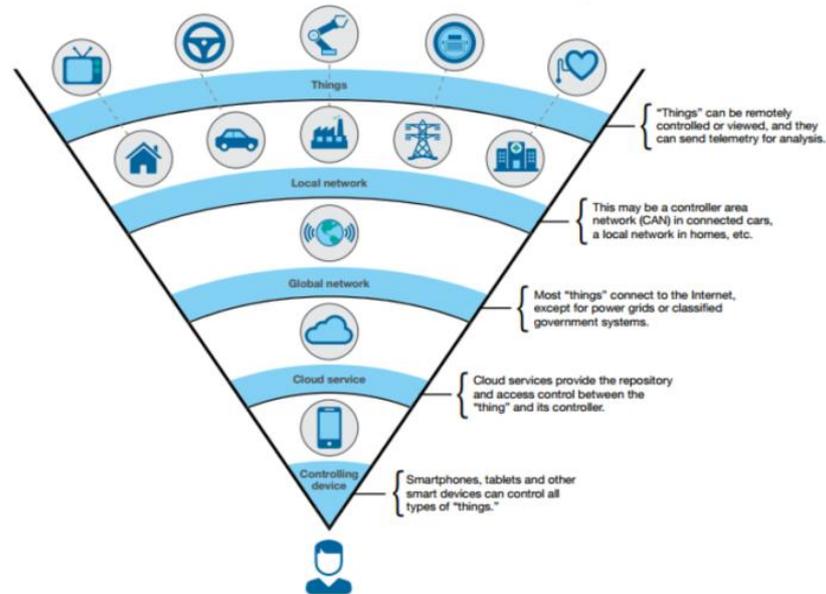


Fig.I.1 Automation System Using (IoT)

I.3.1 Smart Home (Home Automation)

Smart home technology applied can provide users with more freedom and a higher quality of life. Household devices connected to Internet can be monitored and controlled at Home or even from any part of the world.

A home automation system will control lighting, climate, entertainment systems, and appliances. It may also include home security such as access control and alarm systems. When connected with the Internet, home devices are an important constituent of the Internet of Things.

I.3.2 Industrial Automation

Automation in industry is very critical, as there are harsh and unpleasant conditions for human beings. Instead of following manual processes, when taking a step further with automated control – by connecting the systems and controlling over the Internet, it gives great control for the employer/manager and monitoring since you can monitor not only from the company’s premise but from anywhere. This facilitates productivity and ensures tasks/people can be monitored simultaneously.[3]

I.3.3 Security System

Burglar alarm, break detectors fitted in house or shops can be connected to Internet. This will give a high priority alert to owners and nearby police stations.

I.3.4 Agricultural Automation

Agricultural fields can be fitted with various sensors like soil moisture, humidity temperature and even water pumps can be connected to Internet. All this can be controlled and monitored over Internet.

I.3.5 Benefits and Risks

As any new technology, the IoT presents huge benefits and risks. Developing the policy approaches to

This industry, we must understand both. Here is a summary of the benefits and risks of IoT.

I.3.5.1 Benefits

The benefits of IOT are immense. In this practical world, you can integrate IOT into anything according to your creativity and need. Be it business conglomerates or an individual small home, the amount of benefit you can derive from IOT is enormous. IOT is the hot topic in the industry as well as house-holds because of its huge technical advancements and assistance in making our lives seamless and smarter.[2]

- **Real time monitoring**

The devices that are given Internet access will be updating their status or condition periodically or on user request. This gives users the exact information of the condition of that device at that instant which enables you to take necessary actions.

- **Real time data logging**

In previous systems data used to be logged on to system either by taking manual readings or through other wired protocol. But this limitation can be overcome by giving the device to update its data over Internet.

- **Data analytics**

The upcoming cloud technologies help to organize and present user data in very descriptive and graphical Ways like bar graphs, graphs, pie charts etc.

- **Remote monitoring**

Any place that has Internet facility can be monitored from all over the world through GPRS or Ethernet or wired connection.

- **Control based on actual situation's data**

Controls can be setup automatically on actual situation. Users can also give commands on basis of data.

- **Process optimization**

As the availability of data is time independent and location independent, work can be optimized accordingly.

- **Simplicity to access**

Controls are available through mobile devices. Even apps can be developed to give users ease of access.

I.3.5.2 Risks

Although IoT offers us an impressive set of benefits, it also presents a significant set of challenges. Here is a list of some of its main problems.

- **Security**

The Internet of Things creates an ecosystem of permanently connected devices that communicate over networks. The system offers less control despite all the security measures. This leaves users exposed to different types of attackers ex Hackers.

- **Complexity**

Some find IoT systems complicated in terms of design, deployment and maintenance, given the use of multiple technologies and a large number of new Technologies.

- **Confidentiality**

The sophistication of IoT provides substantial personal data down to the smallest detail without the active participation of the user.

I.4 Home Automation

Is building automation for a home, called a **smart home** or **smart house**. A home automation system will control lighting, climate, entertainment systems, and appliances. It may also include home security such as access control and alarm systems. When connected with the Internet, home devices are an important constituent of the Internet of Things.

A home automation system typically connects controlled devices to a central hub or "gateway". The user interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface, that may also be accessible off-site through the Internet.

While there are many competing vendors, there are very few worldwide accepted industry standards and the smart home space is heavily fragmented. Manufacturers often prevent independent implementations by withholding documentation and by litigation.[2]

- Occupancy-aware control system: it is possible to sense the occupancy of the home using smart meters and environmental sensors like CO sensors, which can be integrated into the building automation system to trigger automatic responses for energy efficiency and building comfort applications.
- Appliance control and integration with the smart grid and a smart meter, taking advantage, for instance, of high solar panel output in the middle of the day to run washing machines.
- Security: a household security system integrated with a home automation system can provide additional services such as remote surveillance of security cameras over the Internet, or central locking of all perimeter doors and windows.
- Leak detection, smoke and CO detectors
- Indoor positioning systems
- Home automation for elderly and disabled
- Pet Care, for example tracking the pets movements and controlling access rights[1]

I.4.3 Implementations

Home automation suffers from platform; platform fragmentation and lack of technical standards a situation where the variety of home automation devices, in terms of both hardware variations and differences in the software running on them, makes the task of developing applications that work consistently between different inconsistent technology ecosystems hard. People may be hesitant to bet their IoT future on proprietary software or hardware devices that use proprietary protocols that may fade or become difficult to customize and interconnect

I.5 Home Security

A smart home security system connects to your home Wi-Fi network so you can monitor and control your security devices using your smartphone and an app. Entry-level systems typically include a couple of door and window sensors, a motion detector, and a hub that communicates with these devices using one or more wireless protocols such as Wi-Fi, Z-Wave, Zigbee, or a proprietary mesh network. You can add extra door, motion, and window sensors to provide coverage for your entire house and build a comprehensive system that includes door locks, garage door openers, indoor and outdoor surveillance cameras, lights, sirens, smoke/CO detectors, water sensors, and more.[1]



Fig.1.3 Screen of Home Automation on smart phone

I.5.1 Definition

Wireless Home security and Home automation are the dual aspects of this project. The currently built prototype of the system sends alerts to the owner over voice calls using the Internet if any sort of human movement is sensed near the entrance of his house and raises an alarm optionally upon the user's discretion. The provision for sending alert messages to concerned security personnel in case of critical situation is also built into the system. On the other hand if the owner identifies that the person entering his house is not an intruder but an unexpected guest of his then instead of triggering the security alarm, the user/owner can make arrangements such as opening the door, switching on various appliances inside the house, which are also connected and controlled by the micro-controller in the system to welcome his guest.

Conventional security systems keep homeowners, and their property, safe from intruders. A smart home security system, however, offers many more benefits. Home automation technology notifies homeowners of any problems, so that they can investigate. Artificial intelligence programs keep track of the homeowner's habits, and other important information, and notify emergency personnel when necessary.[1]

I.5.2 Smart Home Fire Protection

Smart home security system offers much more protection than the typical fire alarm. This type of system checks carbon monoxide levels as well as watching for signs of fire and monitors all areas of the home. In the event of a fire, the smart home security system can alert the homeowner and notify emergency services. Artificial intelligence programs are even able to pinpoint the location of the fire, and provide that information to fire department personnel as they respond.

1.5.3 Access Control

Security codes, motion detectors, and cameras provide information to a smart home security system, allowing it to determine whether an individual is a resident, a cleared visitor, or an intruder. Motion detectors trigger an alert, letting the artificial intelligence program know that there is someone or something to be evaluated. Facial recognition software and security codes allow the security system to allow residents into the home, while based on pre-programmed information restrict access to other individuals.

1.5.4 Artificial Intelligence Programs Protect the Homeowner

Intruders and fires are not the only dangers in a home. A smart home security system also protects residents from unanticipated health problems. Using the same cameras and motion detectors that protect the outside of a home, smart houses can learn about the habits and normal movements of the residents. When the resident does something unexpected, and does not resume normal activities, the smart home can alert family members or emergency services. This aspect of a smart home is particularly helpful for the elderly, or those in fragile health.

Security cameras come in a wide variety of form factors so there is a camera for every security need. While building a surveillance system you can choose which type best meets your needs, so you can get the video quality and safety you want. Sturdy outdoor and vandal-proof cameras are designed to withstand extreme temperatures, snow and rain and even attempts to tamper or damage the camera. Night vision and thermal cameras make it possible to record video at night when sunlight or overhead lights aren't available, while megapixel cameras give you large detailed videos. Closed-loop security systems are systems that you must monitor yourself. These usually include motion detectors, silent alarms, door detectors and security cameras that can send text messages or push notifications to your mobile phone or even call you when an alarm is triggered. . Plus, most of them connect to mobile apps that allow you to remotely monitor [3]

1.5.5 Protecting Confidential Information

Confidential information can reside in two states on a network. It can reside on physical storage media such as a hard drive or memory, or it can reside in transit across the physical network wire in the form of packets. These states present multiple opportunities for attacks from users on your internal network, as well as those users on the Internet. We are primarily concerned with the second state, which involves network

security issues. The following are five common methods of attack that present opportunities to compromise the information on your network:

- Network packet sniffers
- Password attacks
- IP spoofing
- Man-in-the-middle attacks
- Distribution of sensitive internal information to external sources

When protecting your information from these attacks, your concern is to prevent the theft, destruction, corruption, and introduction of information that can cause irreparable damage to sensitive and confidential data. This section describes these common methods of attack and provides examples of how your information can be compromised.

I.6 Conclusion

While the concept of combining computers, sensors, and networks to monitor and control devices has been around for decades, the recent confluence of key technologies and market trends is ushering in a new reality for the “Internet of Things”. IoT promises to usher in a revolutionary, fully interconnected “smart” world, with relationships between objects and their environment and objects and people becoming more tightly intertwined. The prospect of the Internet of Things as a ubiquitous array of devices bound to the Internet might fundamentally change how people think about what it means to be “online”.

While the potential ramifications are significant, a number of potential challenges may stand in the way of this vision – particularly in the areas of security; privacy; interoperability and standards; legal, regulatory, and rights issues; and the inclusion of emerging economies. The Internet of Things involves a complex and evolving set of technological, social, and policy considerations across a diverse set of stakeholders. The Internet of Things is happening now, and there is a need to address its challenges and maximize its benefits while reducing its risks.

The Internet Society cares about IoT because it represents a growing aspect of how people and institutions are likely to interact with and incorporate the Internet and network

connectivity into their personal, social, and economic lives. Solutions to maximizing the benefits of IoT while minimizing the risks will not be found by engaging in a polarized debate that pits the promises of IoT against its possible perils. Rather, it will take informed engagement, dialogue, and collaboration across a range of stakeholders to plot the most effective ways forward.

CHAPTER II

II.1 Introduction

Home Automation is sought to solve these problems and provide user with the instant monitoring and controlling of home appliances. There are several devices for such monitoring and controlling. In this Master Thesis, we are focused on monitoring and controlling the home appliances with Raspberry Pi in extension with Arduino Uno and WiFi chip Node MCU V1.0 (ESP8266). As these devices are cheaper and simpler to use and handle, they are used in co-operation to widen the work and service functionality and comparatively.

II.2 Raspberry Pi - Single Board Computer

II.2.1 History

A Raspberry Pi is a credit-card sized computer originally designed for education, inspired by the 1981 BBC Micro. Creator Eben Upton's goal was to create a low-cost device that would improve programming skills and hardware understanding at the pre-university level. But thanks to its small size and accessible price, it was quickly adopted by tinkerers, makers, and electronics enthusiasts for projects that require more than a basic microcontroller (such as Arduino devices).

The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level.

II.2.2 Definition of Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards, mice and cases). However, some accessories have been included in several official and unofficial.

II.2.3 The Raspberry Pi Components

The Raspberry Pi device looks like a motherboard, with the mounted chips and ports exposed (something you'd expect to see only if you opened up your computer and looked at its internal boards), but it has all the components you need to connect input, output, and storage devices and start computing.

You'll encounter two models of the device: **Model A** and **Model B**. The only real differences are the addition of Ethernet and an extra USB port on the more expensive Model B.

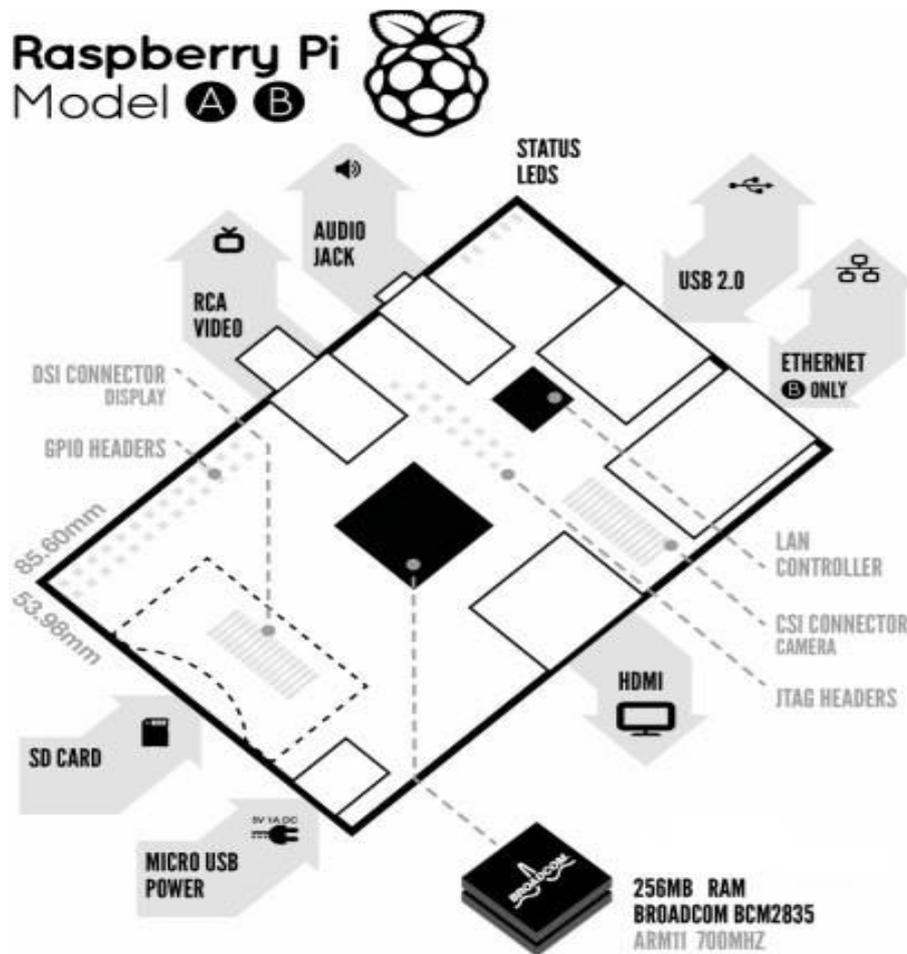


Fig.II.1 Raspberry pi components

Here are the various components on the Raspberry Pi board

- **ARM CPU/GPU** -- This is a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM central processing unit (CPU) and a Video core 4 graphics processing unit (GPU). The CPU handles all the computations that make a computer work (taking input, doing calculations and producing output), and the GPU handles graphics output.
- **GPIO** -- These are exposed general-purpose input/output connection points that will allow the real hardware hobbyists the opportunity to tinker.
- **RCA** -- An RCA jack allows connection of analog TVs and other similar output devices.
- **Audio out** -- This is a standard 3.55-millimeter jack for connection of audio output devices such as headphones or speakers. There is no audio in.
- **LEDs** -- Light-emitting diodes, for all of your indicator light needs.
- **USB** -- This is a common connection port for peripheral devices of all types (including your mouse and keyboard). Model A has one, and Model B has two. You can use a USB hub to expand the number of ports or plug your mouse into your keyboard if it has its own USB port.

- **HDMI** -- This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable.
- **Power** -- This is a 5v Micro USB power connector into which you can plug your compatible power supply.
- **SD card slot** -- This is a full-sized SD card slot. An SD card with an operating system (OS) installed is required for booting the device. They are available for purchase from the manufacturers, but you can also download an OS and save it to the card yourself if you have a Linux machine and the wherewithal.
- **Ethernet** -- This connector allows for wired network access and is only available on the Model B.

Many of the features that are missing, such as WiFi and audio in, can be added using the USB port(s) or a USB hub as needed. Next: More details on the device itself and its compatible operating systems. [5]

II.2.4 Different Versions of the Raspberry Pi

1. RaspberryPi model A
2. RaspberryPi model B
3. Raspberry Pi model B+
4. Raspberry Pi model B revision 2

All different versions of Raspberry Pi in [Appendix A].

II.2.5 Technical Specification of Raspberry Pi 3 Model B+

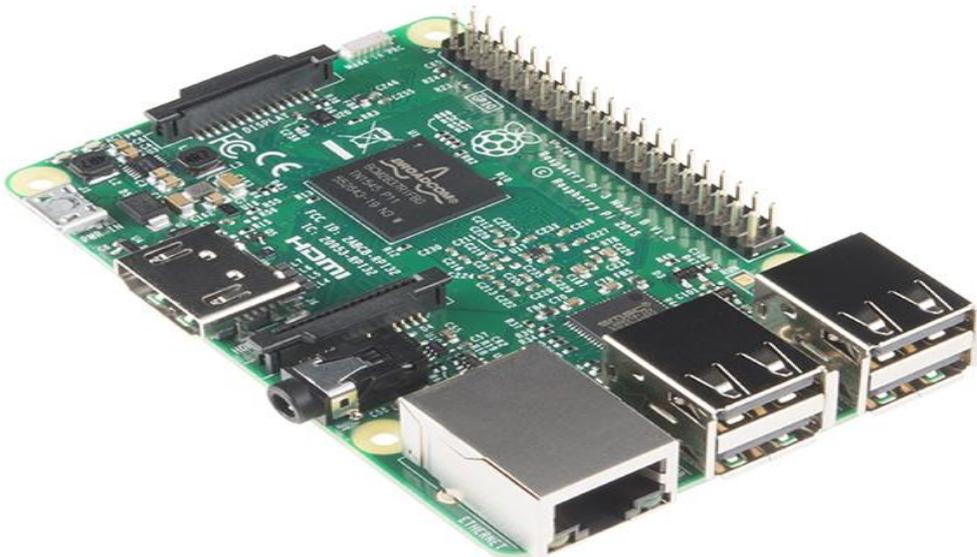


Fig .II.2 Raspberry Pi3 Model B+

- Broadcom BCM2387 chipset.

- 1.2GHz Quad-Core ARM Cortex-A53 (64Bit) 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
- IEEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100 meters.
- IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard (AES) with 128-bit key, the maximum range of 50 meters. GPU
- Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode.
- Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure Memory
- 1GB LPDDR2 Operating System
- Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT Dimensions • 85 x 56 x 17mm Power
- Micro USB socket 5V1, 2.5A Connectors: Ethernet
- 10/100 BaseT Ethernet socket Video Output • HDMI (rev 1.3 & 1.4) • Composite RCA (PAL and NTSC) Audio Output
- Audio Output 3.5mm jack
- HDMI
- USB 4 x USB 2.0 Connector GPIO Connector

II.2.6 Raspberry Pi 3 Model B Features

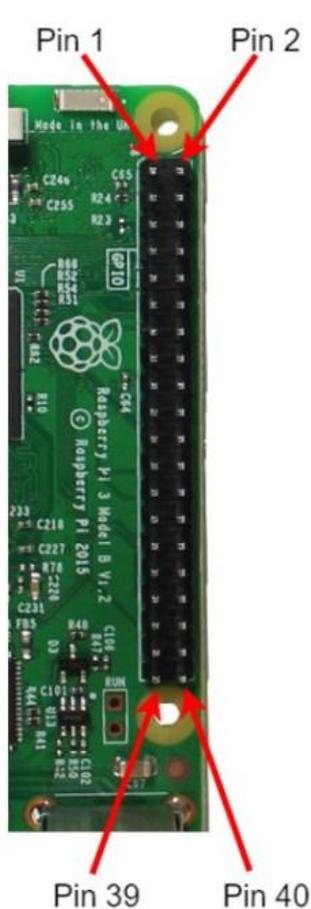
- Now **10x Faster** - Broadcom BCM2387 ARM Cortex-A53 Quad Core Processor powered Single Board Computer running at 1.2GHz!
- 1GB RAM so you can now run bigger and more powerful applications
- Fully HAT compatible
- 40 pins extended GPIO to enhance your “real world” projects.
- Connect a Raspberry Pi camera and touch screen display (each sold separately)

- Stream and watch Hi-definition video output at 1080
- Micro SD slot for storing information and loading your operating systems.
- 10/100 BaseT Ethernet socket to quickly connect the Raspberry Pi to the Internet

II.2.7 General Purpose Input Output on the Raspberry Pi (GPIO)

These pins are a physical interface between the Raspberry Pi and the outside world. You can program the Raspberry Pi to switch devices on and off (output), or receive data from sensors and switches (input). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins).

The GPIO pins are one way in which the Raspberry Pi can control and monitor the outside world by being connected to electronic circuits. Most models of the Raspberry Pi have 40 pins that look like this:



	Pin no.				
DC Power	3.3V	1	2	5V	DC Power
SDA1, I ² C	GPIO 2	3	4	5V	DC Power
SCL1, I ² C	GPIO 3	5	6	GND	
GPIO_GCLK	GPIO 4	7	8	GPIO 14	TXD0
	GND	9	10	GPIO 15	RXD0
GPIO_GEN0	GPIO 17	11	12	GPIO 18	GPIO_GEN1
GPIO_GEN2	GPIO 27	13	14	GND	
GPIO_GEN3	GPIO 22	15	16	GPIO 23	GPIO_GEN4
DC Power	3.3V	17	18	GPIO 24	GPIO_GEN5
SPI_MOSI	GPIO 10	19	20	GND	
SPI_MISO	GPIO 9	21	22	GPIO 25	GPIO_GEN6
SPI_CLK	GPIO 11	23	24	GPIO 8	SPI_CE0_N
	GND	25	26	GPIO 7	SPI_CE1_N
I ² C ID EEPROM	DNC	27	28	DNC	I ² C ID EEPROM
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	
	GND	39	40	GPIO 21	

Fig.II.3 GPIO Pins Descriptions of Raspberry Pi

The GPIO connector has a number of different types of connection:

The GPIO connector has a number of different types of connection:

- True GPIO (General Purpose Input Output) pins that you can use to turn LEDs on and off etc.
- I2C interface pins that allow you to connect hardware modules with just two control pins.
- SPI interfaces with SPI devices, a similar concept to I2C but uses a different standard
- Serial Rx and Tx pins for communication with serial peripherals.
- GPIO pins can be used as both digital outputs and digital inputs.
- Output: turn a particular pin HIGH or LOW.
- Setting it HIGH sets it to 3.3V; setting it LOW sets it to 0V.
- Input: detect the pin being at HIGH or LOW.
- We can connect switches and simple sensors to a pin and check whether it is open or closed (that is, activated or not).

II.2.8 Choosing the Operating System OS

The Raspberry Pi was designed for Linux operating system, and many Linux distributions now have a version optimized for the Raspberry Pi. Two of most popular option is Raspbian, which is based on the Debian operating system and Pidora, which is based on the Fedora operation system. Raspbian operating system is the best operating system for Raspberry Pi3.

II.2.9 Characteristics of Raspbian Operating System (ROS)

- The Raspbian is a lightweight OS which is based on Debian distribution
- It is specifically designed to work for Raspberry Pi devices
- It comes with pre-installed software like LibreOffice, email client, etc.
- It also includes a Minecraft special version to help you make something
- You will easily find the guidelines and troubleshooting for Raspbian
- There are dozens of videos available on YouTube also for the understanding and using Raspbian OS

II.2.10 Installing Raspbian Operating System (ROS)

Installing Raspbian on the Raspberry Pi is pretty straightforward. We'll be downloading Raspbian and writing the disc image to a microSD card, then booting the Raspberry Pi to that microSD card. For this project, you'll need a microSD card (go with at least 8 GB), a computer with a slot for it, and, of course, a Raspberry Pi and basic peripherals (a mouse, keyboard, screen, and power source). This isn't

the only method for installing Raspbian (more on that in a moment), but it's a useful technique to learn because it can also be used to install so many other operating systems on the Raspberry Pi. Once you know how to write a disc image to a microSD card, you open up a lot of options for fun Raspberry Pi projects. All steps explained in [appendix B] [4].

To Establish wireless communication with Raspberry Pi 3, we use PuTTY.

II.2.11 Definition of PuTTY

Is an open source application that supports multiple network protocols, such as - SSH, Telnet, SCP, rlogin, serial port and raw socket connection. PuTTY communication is established over a TCP/IP socket, such as PuTTY, but it uses the secure socket with public key encryption. A typed command is sent and received a text response.

Although the word "PuTTY" has no definitive meaning, 'TTY' is the name for a terminal in the Unix tradition. PuTTY which is still beta software was originally developed by Simon Tatham for Microsoft Windows, but over the years it has been ported to several other OS as well. Windows users may download the free terminal program PuTTY (<https://www.mines.edu/ccit/software-title/putty/>)

SSH, or Secure Shell, is a "terminal" program used to encrypt online communications from end to end, preventing unauthorized access to a data stream. Previous terminal programs, like Telnet, would send username and password credentials over the network in plain text, where they could be intercepted and used by anyone who happened to be watching the server traffic. For better security, CCIT no longer allows Telnet traffic to most CCIT-managed servers. Users of Linux or Mac OS X enjoy native SSH support in their terminal programs. [8] [Appendix I]

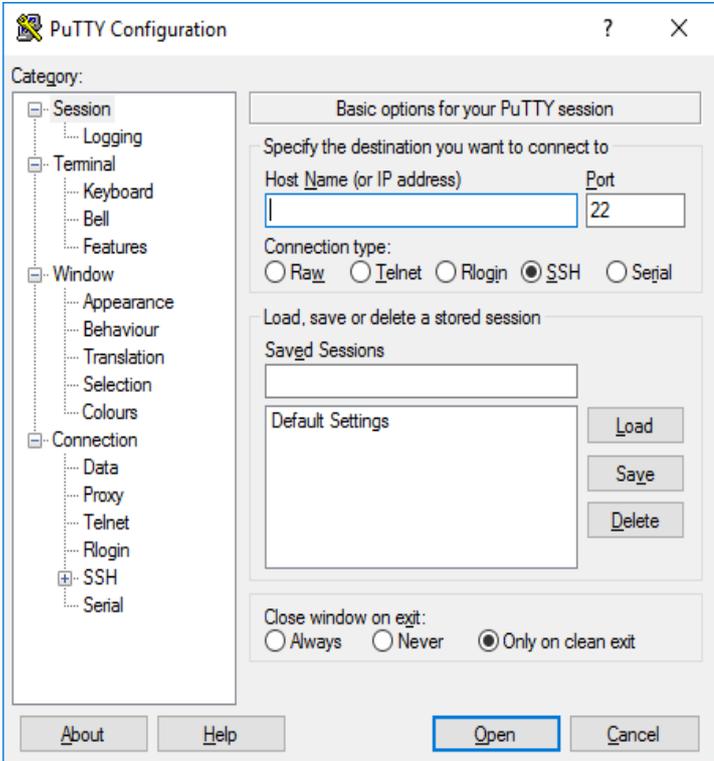


Fig.II.4 PuTTY

II.2.12 PuTTY Features

PuTTY supports multiple variations on the secure remote terminal.

- It provides control over the SSH encryption key and protocol version.
- It stores hosts and preferences for later usage.
- Command line SCP and SFTP clients.
- Emulate control sequences from
- It allows remote, local or dynamic port forwarding with SSH.
- Used with local serial port connections.

II.2.13 Remotely Control Raspberry Pi3 Using VNC Protocol

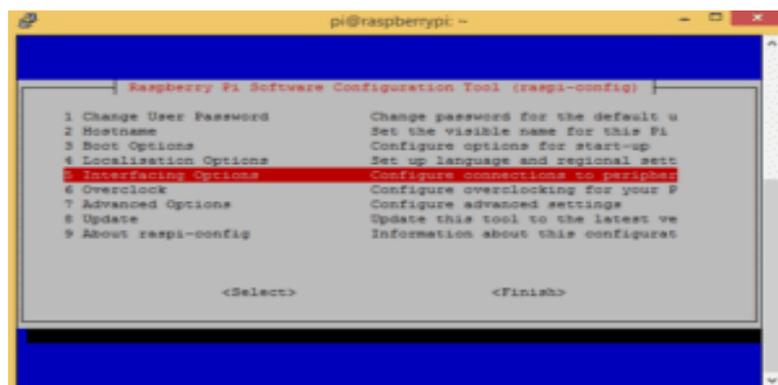
VNC Stands for Virtual Network Computing; it mainly allows viewing the Raspberry Pi desktop on any other device with a VNC Client/Viewer, regardless of the type of the operating system.

Sometimes it is not convenient to work directly on the raspberry Pi. Maybe you would like to work on it from another device by remote control.

VNC is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer (running VNC Server) from another computer or mobile device (running VNC Viewer) VNC Viewer transmits the keyboard and either mouse or touch events to VNC Server, and receives updates to the screen in return.

You will see the desktop of the Raspberry Pi inside a window on your computer or mobile device. You'll be able to control it as though you were working on the Raspberry Pi itself.

Real VNC Server is included in Raspbian Jessie releases after 9-23-16. All we need to do on the Pi is enabling VNC in the `sudo raspi-config` at the command prompt to access it. Then select 'Interfacing Options' from the menu [9] [Appendix I]



Then select 'VNC' to enable VNC

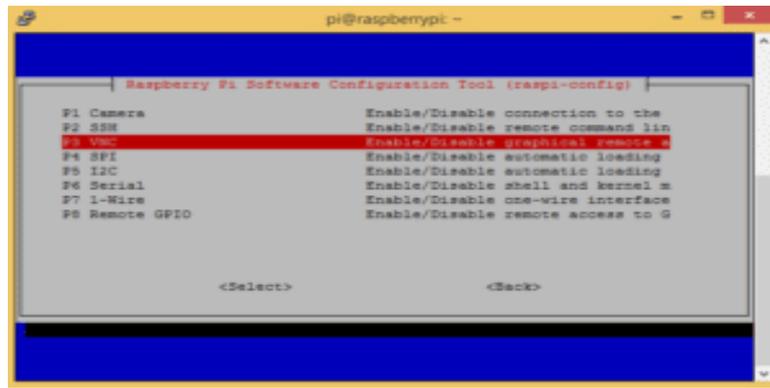


Fig.II.5 Snapshot of configuration screen for Raspberry Pi

Now reboot the Pi with `sudo reboot` command prompt. We can see our Raspberry Pi3 desktop from our computer.

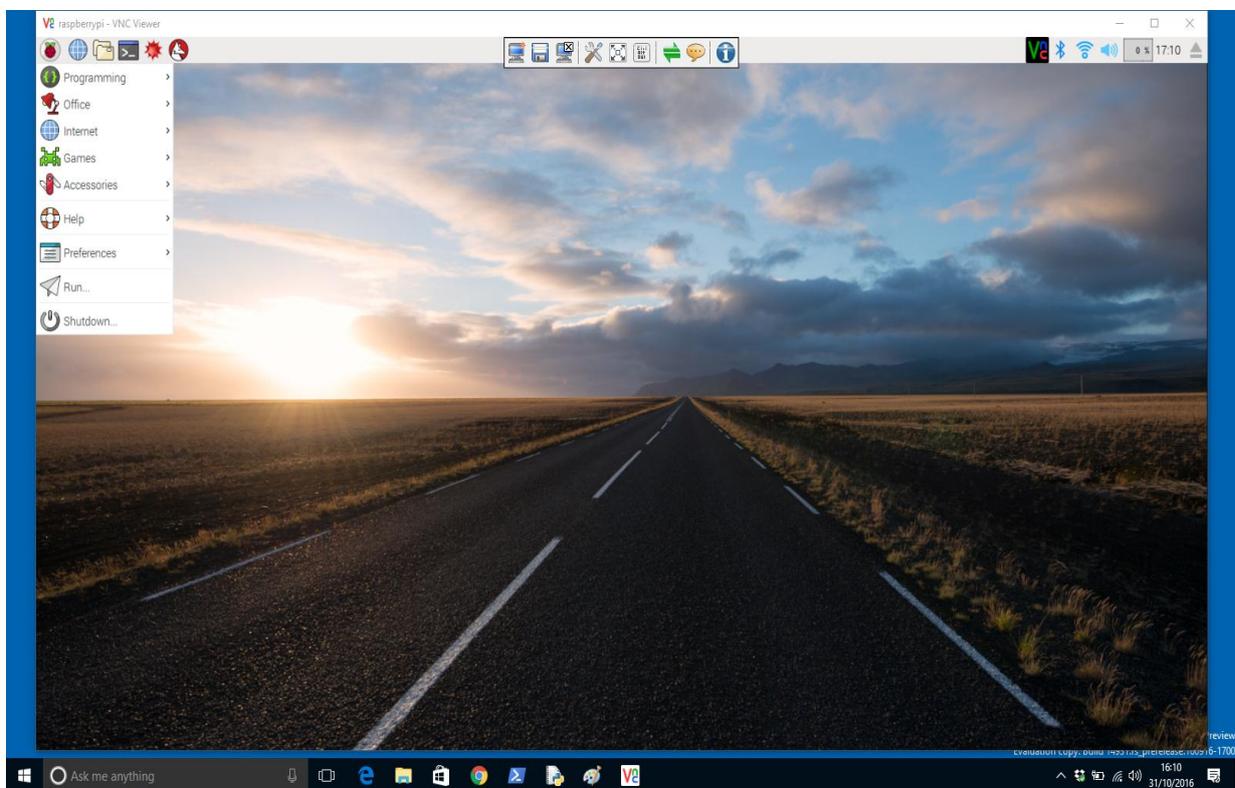


Fig.II.6 Desktop for Raspberry Pi

II.3 ARDUINO UNO

II.3.1 HISTORY

In 2005, in Ivrea, a project was initiated to make a device for controlling student-built interactive design projects that was less expensive than other prototyping system available at the time. One of the cofounders, Massimo Banzi, named this piece of hardware Arduino in

honor of Bar di Re Arduino (In 1002, KingArduin became the ruler of Italy. Today, the Bar di Re Arduino, a pub on cobblestoned street in town, honors his memory), and began producing boards in a small factory located in the same region as the computer company Olivetti.

The Arduino project is a fork of the open source Wiring platform and is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and notifications, and a Processing-based integrated development environment (IDE). [7]

II.3.2 Definition of Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

II.3.3 The Usage of The Arduino Board

This board gives us a huge and wide a range of projects in different fields like, Home Automation, controlling sensors..., etc

- Home Automation
- Robotics
- Controlling Sensors
-

II.3.4 The Family of Arduino Boards

Over the years the designers at Arduino.cc have developed a number of board designs. The first widely distributed Arduino board, the Diecimila, was released in 2007, and since its initial release the Arduino family has evolved to take advantage of the various types of Atmel AVR MCU devices. The Due, released in 2012, is the first Arduino to utilize a 32-bit ARM Cortex-M3 processor, and it breaks from the rest of the family in terms of both processing power and board pin out configuration. Other boards, like the LilyPad and the Nano, also do not have the same pin out as the other members of the family, and are intended for a different range of applications—wearable in the case of the LilyPad; handheld devices for the Esplora; and compact size in the case of the Mini, Micro, and Nano.[4]

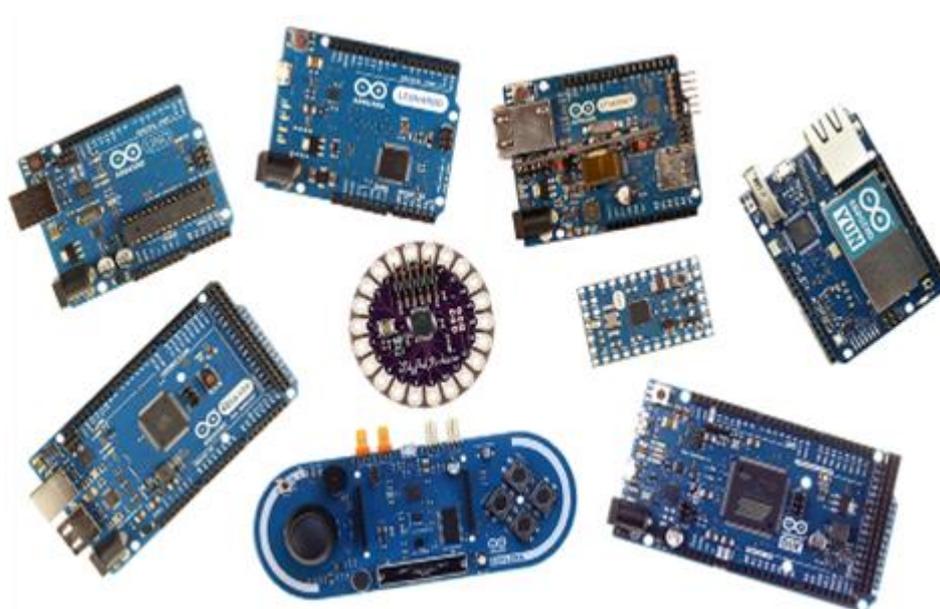


Fig.II.7 Arduino Boards

II.3.5 Timeline of Arduino products

Board name	Year	Microcontroller	Board name	Year	Microcontroller
Diecimila	2007	ATmega168V	Mega 2560	2010	ATmega2560
LilyPad	2007	ATmega168V/ATmega328V	Uno	2010	ATmega328P
Nano	2008	ATmega328/ATmega168	Ethernet	2011	ATmega328
Mini	2008	ATmega168	Mega ADK	2011	ATmega2560
Mini Pro	2008	ATmega328	Leonardo	2012	ATmega32U4
Duemilanove	2008	ATmega168/ATmega328	Esplora	2012	ATmega32U4
Mega	2009	ATmega1280	Micro	2012	ATmega32U4
Fio	2010	ATmega328P	Yún	2013	ATmega32U4 + Linino

Tab.II.1 Arduino Products

The latest additions to the Arduino family, the Leonardo, Esplora, Micro, and Yún, all use the ATmega32U4. While this part is similar to an ATmega328 it also incorporates an integrated USB-to-serial interface component, which eliminates one of the integrated circuit (IC) parts found on boards like the Uno and Duemilanove. The most known Board of the Arduino family is Arduino Uno.

II.3.6 Arduino Uno Board

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.[5]

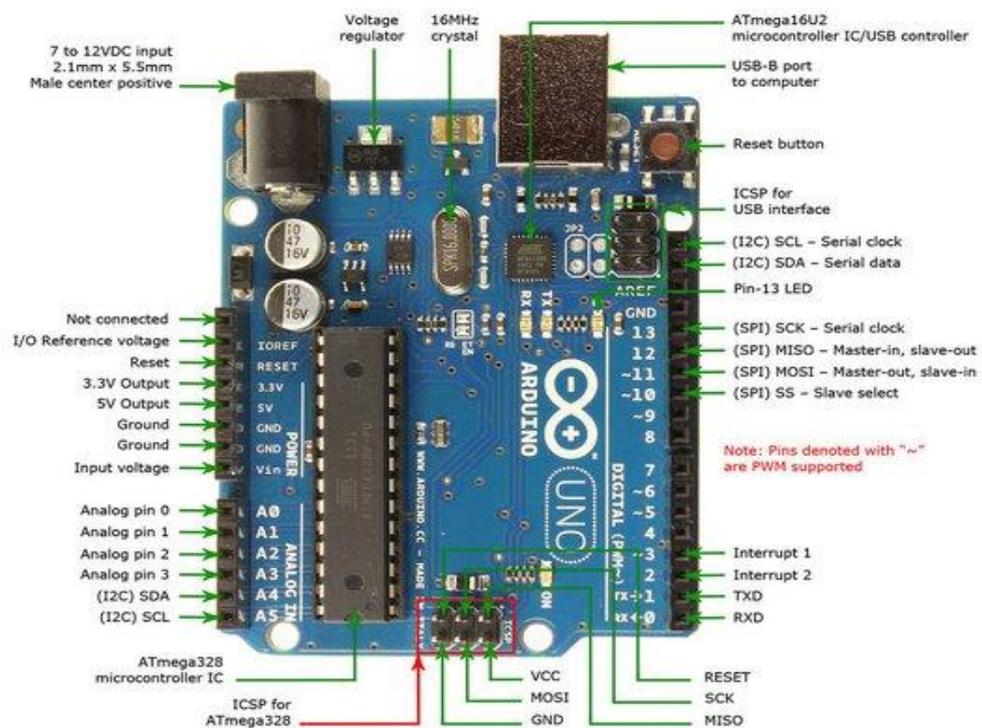


Fig.II.8 Arduino Uno Board

II.3.7 Technical Specifications

Microcontroller	ATmega328
Operating Voltage	5V

Input Voltage (recommended)	7- 12V
Input Voltage (limits)	6- 20V
Digital I/O Pins	14(of which 6 provide PWM Output)
Analog Input Pins	6
DC current per I/O Pin	40 mA
DC current for 3.3V Pin	50 mA
Flash Memory loader	32 KB of which 0.5 KB used by Boot loader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz



B

Fig.II.9 Arduino Uno Board Pin Description

- Analog Reference pin
- Digital Ground

- Digital Pins 2-13
- Digital Pins 0-1/Serial In/Out - TX/RX - *These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin).*
- Reset Button - S1
- In-circuit Serial Programmer
- Analog In Pins 0-5
- Power and Ground Pins
- External Power Supply In (9-12VDC) - X1
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board)

- **Digital Pins**

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the `pinMode ()`, `digitalRead ()`, and `digitalWrite ()` commands. Each pin has an internal pull-up resistor which can be turned on and off using `digitalWrite ()` (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- BT Reset: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

- **Analog Pins**

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the `analogRead()` function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

- I²C: 4 (SDA) and 5 (SCL). Support I²C (TWI) communication using the `Wire` library (documentation on the Wiring website).

- **Power Pins**

- VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltage ranges, please see the [documentation for your board](#). Also note that the LilyPad has no VIN pin and accepts only a regulated input.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- GND. Ground pins.

- **Other Pins**

- AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.[7]

II.3.8 Software Requirements

Arduino provides like a C language for programming Arduino boards. You will be using the Arduino IDE for writing code and uploading it to an Arduino board. You can install the latest version of Arduino IDE from <https://www.arduino.cc/en/Main/Software>. Once Arduino IDE has been installed on your machine, open it and, as shown below and it will load default code.



Fig.II.10 Arduino IDE

II.3.8.1 Components of Arduino IDE

- Toolbar
- Status window
- Serial Monitor window
- **Toolbar**

The toolbar on top of the IDE, as shown below, provides easy access to frequently used options.



Fig.II.11 IDE Toolbar Description

- **Verify/Compile:** This is the first button from the left (the tick mark). Click this button to verify and compile your code for correctness. You can view the results in the Status window at the bottom.
- **Upload:** This is the second button from left (right-pointing arrow). If your Arduino board is connected to your machine that is running the Arduino IDE, this will upload the code on the Arduino board. You can view the deployment results in the Status window at the bottom.
- **New/Open/Save:** The next three buttons, as their names suggest, let you open a new code window, open an existing code file, or save the currently open code. Arduino code files have an *.ino extension.
- **Serial/Monitor:** The last button on the right lets you open the Serial Monitor window.

- **Status Window**

When you verify the code or upload it to a board, the Status window shown in Figure 1-5 lists all the results. Any errors that occur during code verification or uploading will be shown in the Status window.

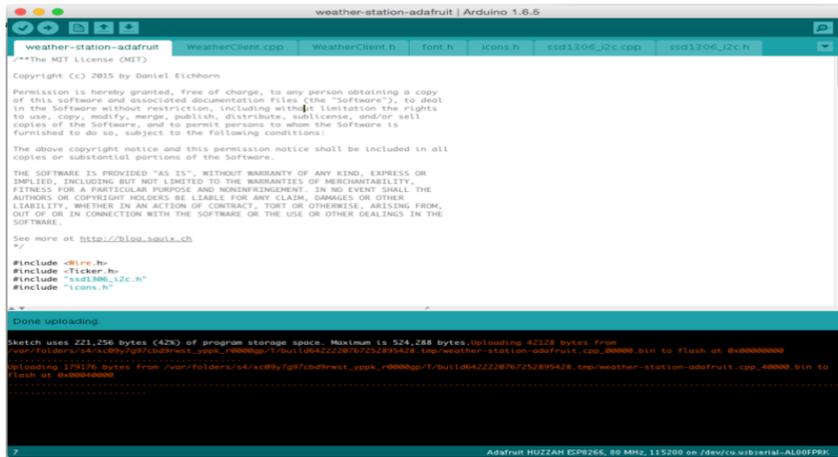


Fig.II.12 IDE Window status

- **Serial Monitor Window**

The Serial Monitor window shown below prints all log messages generated by the Serial.print() and Serial.println() functions in the code. In order to print any messages on the Serial Monitor window, you first need to initialize the message.

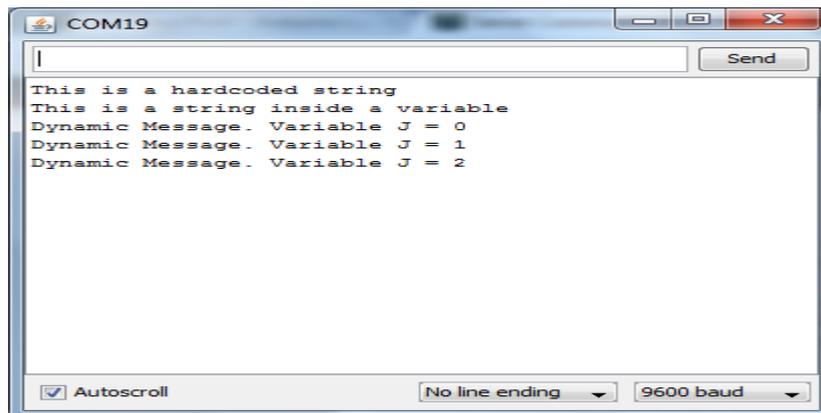


Fig.II.13 IDE Serial Monitor Window

II.3.8.2 Arduino programming language

Has quite a few constructs. However, provides the basics that have been used throughout the projects as below in the table

Code Construct	Description
int	Integer values, such as 123
float	Decimal values, such as 1.15
char	String values, such as “Arduino”
HIGH	Digital pin with current
LOW	Digital pin with no current
INPUT	Pin can only be read
OUTPUT	Pin can only be set
A0-A7	Constants for analog pins, varies by board

0 - 13	Value for digital pins, varies by board
analogRead()	Returns analog pin value(0 - 1023)
analogWrite()	Sets analog pin value
digitalRead()	Returns digital pin value(HIGH or LOW)
digitalWrite()	Sets digital pin value(HIGH or LOW)
Serial.begin()	Initializes serial monitor
Serial.print()	Logs message on serial monitor
Serial.println()	Logs message on serial monitor with new line
Delay(ms)	Adds a wait in processing
Setup()	Standard Arduino function called once
if	Checks for a true/false condition
If ... else	Checks for true/false condition,if false goes to else”
//	Single-line comment
/* */	Multiline comment
#define	Defines a constant
#include	Includes an external library

Tab. II.2 Arduino Instructions & command

- **External libraries:** Includes all required libraries. A library is a fully developed and tested piece of code that you can include and use in your code. For instance, if you wanted to communicate over the Internet using an Ethernet connection, instead of writing all of that code from scratch, you could simply import and include the Ethernet library using `#include <Ethernet.h>`.
- **Constants and variables:** Defines all constants and variables that will be used to read and manipulate data. Constants do not change, so you can, for instance, use them for port numbers on the board. Variables can change, so they can be used for reading sensor data.
- **Functions:** Provides implementation of all custom and standard functions. A function encapsulates a specific functionality. It is recommended to put your code in functions, especially when you are looking to reuse that piece of code. Functions help avoid code duplication.[7]

II.4 ESP8266 WIFI MODULE

II.4.1 History

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability produced by Shanghai-based Chinese manufacturer, Espressif Systems.

The chip first came to the attention of western makers in August 2014 with the **ESP-01** module, made by a third-party manufacturer, Ai-Thinker. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections

using Hayes-style commands. However, at the time there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very few external components on the module which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation.

II.4.2 Definition of ESP8266

The ESP8266 series, or family, of Wi-Fi chips is produced by Espressif Systems, a fabless semiconductor company operating out of Shanghai, China. The ESP8266 series presently includes the ESP8266EX and ESP8285 chips.[8]



Fig .II.14 ESP8266 Boards

II.4.3 Technical Specification

ESP8266EX (simply referred to as ESP8266) is a system-on-chip (SoC) which integrates a 32-bit Tensilica microcontroller, standard digital peripheral interfaces, antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules into a small package. It provides capabilities for 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2), general-purpose input/output (16 GPIO), Inter-Integrated Circuit (I²C), analog-to-digital conversion (10-bit ADC), Serial Peripheral Interface (SPI), I²S interfaces with DMA (sharing pins with GPIO), UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and pulse-width modulation (PWM). The processor core, called "L106" by Espressif, is based on Tensilica's Diamond Standard 106Micro 32-bit processor controller core and runs at 80 MHz (or overclocked to 160 MHz). It has a 64 KiB boot ROM, 32 KiB instruction RAM, and 80 KiB user data RAM. (Also, 32 KiB instruction cache RAM and 16 KiB ETS system data RAM.) External flash memory can be accessed through SPI. The silicon chip itself is housed within a 5 mm × 5 mm Quad Flat No-Leads package with 33 connection pads — 8 pads along each side and one large thermal/ground pad in the center.

II.4.4 Preambles

ESP-12E WiFi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra low

power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller. ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.[9]

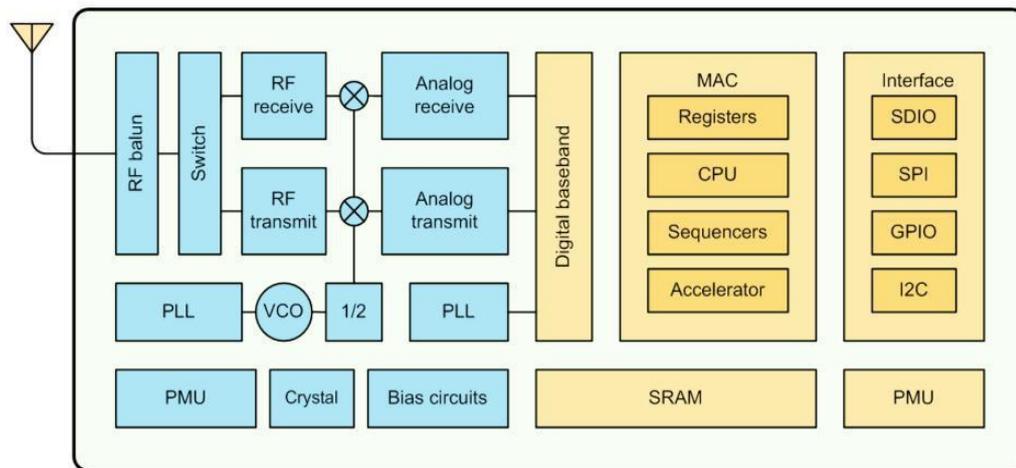


Fig.II.15 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Ten silica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP,

adaptive radio biasing. for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

II.4.5 ESP8266 Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and IOS devices
- Support Smart Link Function for both Android and IOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation and 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms

- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

II.4.6 Parameters

Categories	Items	Values
WiFi Paramters	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Paramaters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Contorl
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	16mm*24mm*3mm
	External Interface	N/A
Software Paramaters	Wi-Fi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via host
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Tab.II.3 ESP8266 Parameters

II.4.7 Pin Descriptions

There are altogether 22 pin counts

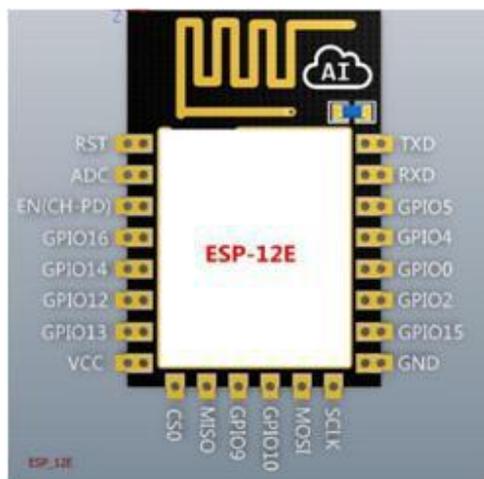


Fig.II.16 ESP-12E Pin Description

NO	Pin Name	Function
1	RST	Reset the module
2	ADC	A/D Conversion
3	EN	Chip enable pin Active high
4	IO16	Wake up from deep sleep
5	IO14	HSPI_CLK
6	IO12	HSPI_MISO
7	IO13	GPIO13,HSPI_MOSI;UART0_CTS
8	VCC	3.3V power supply(VDD)
9	CS0	Chip selection
10	MISO	Slave output Main input
11	IO9	GPIO9
12	IO10	GPIO10
13	MOSI	Main output slave input
14	SCLK	Clock
15	GND	GND
16	IO15	GPIO15;MTDO;HSPICS;UART0_RTS
17	IO2	GPIO2;UART1_TXD
18	IO0	GPIO0
19	IO4	GPIO4
20	IO5	GPIO5
21	RXD	UART0_RXD;GPIO3
22	TXD	UART0_TXD;GPIO1

Tab.II.4 ESP-12E Pin Description

II.4.8 Packaging and Dimension

The external size of the module is 16mm*24mm*3mm, as is illustrated in Figure 3 below. The type of flash integrated in this module is an SPI flash, the capacity of which is 4 MB,

and the package size of which is SOP-210mil. The antenna applied on this module is a 3DBi PCB-on-board antenna.

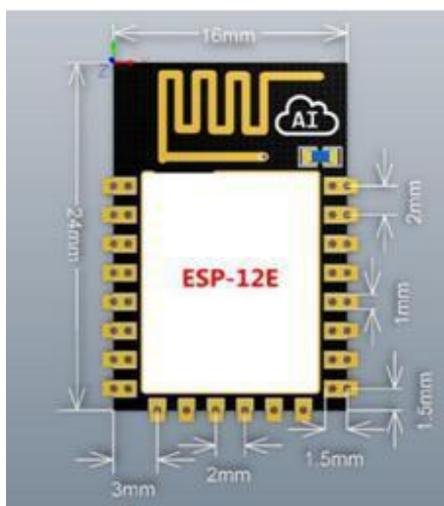


Fig.II.17 ESP-12E Packaging and Dimension

II.4.9 Functional Description

- **MCU**

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. ESP8266EX is often integrated with external sensors and other specific devices through its GPIOs; codes for such applications are provided in examples in the SDK.

- **Memory Organization**

- **Internal SRAM and ROM**

ESP8266EX WiFi SoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

RAM size < 36kB, that is to say, when ESP8266EX is working under the station mode and is connected to the

Router, programmable space accessible to user in heap and data section is around 36kB.)

- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

- **External SPI Flash**

This module is mounted with a 4 MB external SPI flash to store user programs. If larger definable storage space is required, a SPI flash with larger memory size is preferred. Theoretically speaking, up to 16 MB memory capacity can be supported.

- **Suggested SPI Flash memory capacity:**

OTA is disabled: the minimum flash memory that can be supported is 512 kB;

OTA is enabled: the minimum flash memory that can be supported is 1 MB.

Several SPI modes can be supported, including Standard SPI, Dual SPI, and Quad SPI.

- **Crystal**

Currently, the frequency of crystal oscillators supported includes 40MHz, 26MHz and 24MHz. The accuracy of crystal oscillators applied should be $\pm 10\text{PPM}$, and the operating temperature range should be between -20°C and 85°C .

When using the downloading tools, please remember to select the right crystal oscillator type. In circuit design, capacitors C1 and C2, which are connected to the earth, are added to the input and output terminals of the crystal oscillator respectively. The values of the two capacitors can be flexible, ranging from 6pF to 22pF, however, the specific capacitive values of C1 and C2 depend on further testing and adjustment on the overall performance of the whole circuit. Normally, the capacitive values of C1 and C2 are within 10pF if the crystal oscillator frequency is 26MHz, while the values of C1 and C2 are $10\text{pF} < \text{C1}, \text{C2} < 22\text{pF}$ if the crystal oscillator frequency is 40MHz.

- **Interfaces**

Interfaces	Pin Name	Description
HSPI	IO12(MISO) IO13(MOSI) IO14(CLK) IO15(CS)	SPI Flash 2,display screen,and MCU can be connected using HSPI interfaces
PWM	IO12(R) IO15(G) IO13(B)	Currently the PWM interface has four channels, but users can extend the channels according to their own needs. PWM interface can be used to control LED lights, buzzers,relays,and so on.
IR Remote	IO14(IR_T)	The functionality of infrared remote control interface can be implemented via software programming .NECcoding

Control	IO5(IR_R)	,modulation ,and demodulation are used by this interface.The frequency of modulated carrier signal is 38KHz.
ADC	TOUT	ESP8266EX integrate a 10 bit analog ADC, it can be used to test power supply voltage of VDD3P3(Pin3 and Pin4) and the input power voltage of tout(Pin6)However, these two functions cannot be used simultaneously. This interface is typically used in sensor product.
I2C	IO14(SCL) IO2(SDA)	I2C interface can be used to connect external sensor products and display screens,etc.
UART	UART0: TXD(U0TXD) RXD(U0RXD) IO15(RTS) IO13(CTS) UART1: IO2(TXD)	Devices with UART interfaces can be connected with the module Downloading.U0TXD U0RXD orGPIO2 U0RXD Communicating. UART0: U0TXD, U0RXD,M0RXD(U0RTS), MTCK(U0CTS). Debugging UART1_TXD(GPIO2) can be used to print debugging information.
I2S	I2S Input IO12 (I2S_DATA) IO13 (I2S_BCK) IO14(I2S_WS) I2S Output IO15 (I2SO_BCK) IO3 (I2SO_DATA) IO2(I2SO_WS)	I2S interface is mainly used for collecting, processing and transmission of audio data

Tab.II.4 ESP-12E Interfaces

There are a variety of board styles available. The one that I am going to focus on have been given the name **Node MCU V1.0** . It is important to note that there is only one **ESP8266** processor and it is this processor that is found on ALL breakout boards. What distinguishes one board from another is the number of **GPIO** pins exposed, the amount of flash memory provided, the style of connector pins and various other considerations related to construction. From a programming perspective, they are all the same.

II.5 Node MCU V1.0 (ESP8266)

II.5.1 Definition of Node MCU V1.0

The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained WiFi networking solution offering as a bridge from existing micro controller to WiFi and is als This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly. capable of running self-contained applications.

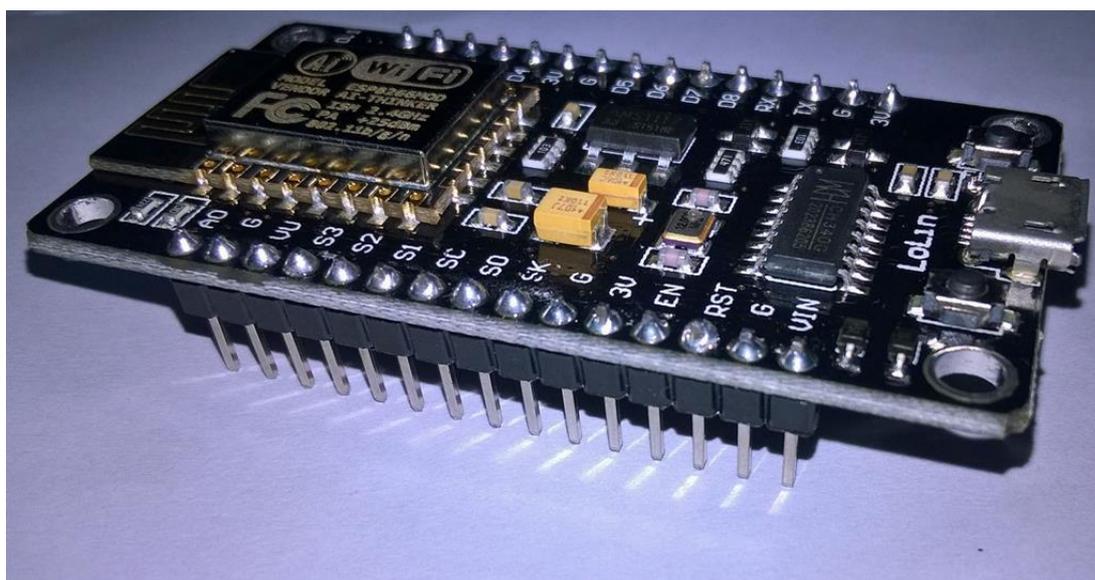


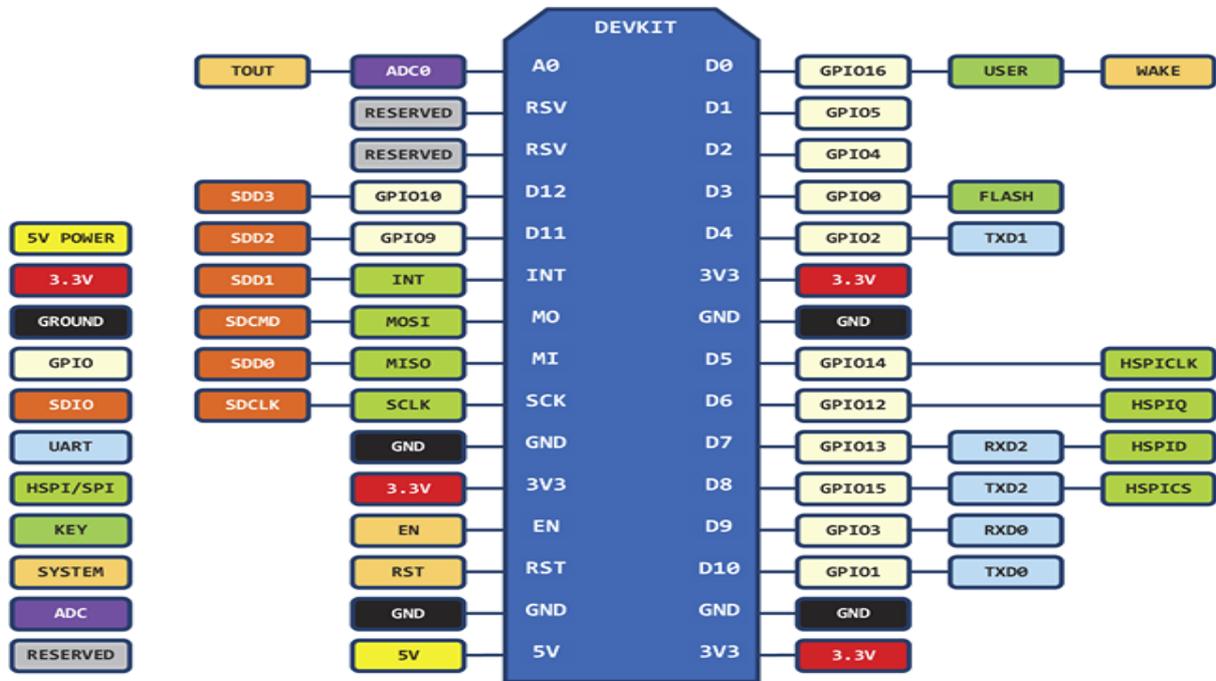
Fig.II.18 Node MCU V 1.0 (LoLin)

II.5.2 Specifications

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.

- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n. Maximum concurrent TCP connections: 5.

II.5.3 Pin Definition



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

Fig.II.19 Node MCU V 1.0 Pin Description

II.5.4 Using Arduino IDE

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266 project, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

- **Install the Arduino IDE 1.6.4 or greater**

Download Arduino IDE from [Arduino.cc](http://arduino.cc)

- **Install the ESP8266 Board Package**

Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into *Additional Board Manager URLs* field in the Arduino v1.6.4+ preferences.

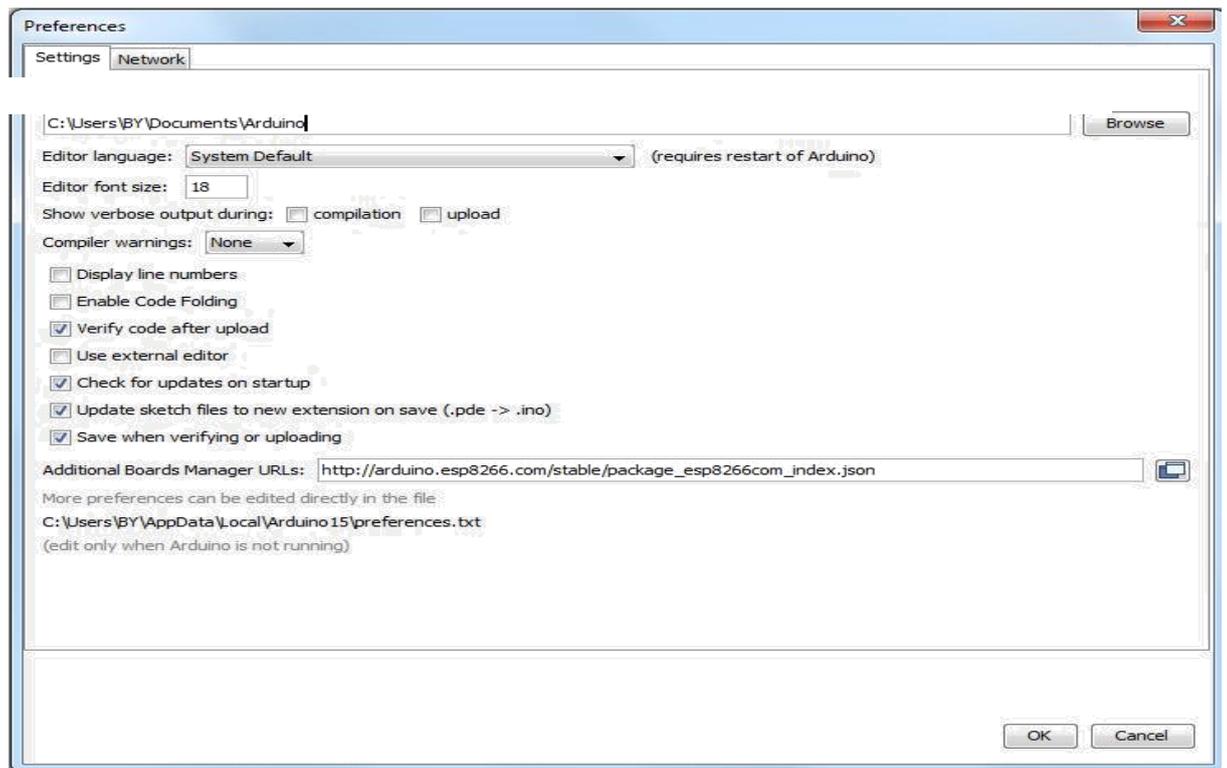


Fig.II.20 *URLs* field in the Arduino v1.6.4+ preferences

Next, use the Board manager to install the ESP8266 package.

Click 'Tools' -> 'Board:' -> 'Board Manager...'

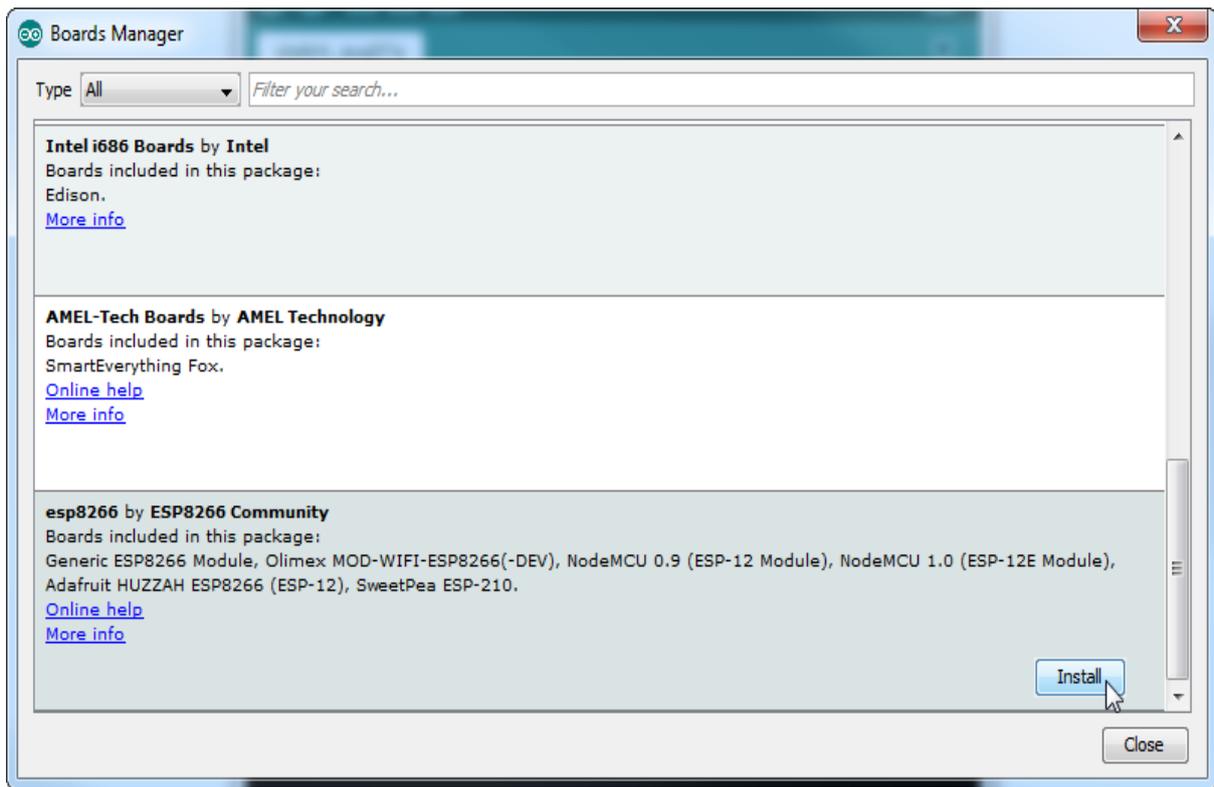


Fig.II.21 Board Manager from Arduino IDE

Scroll down to ‘ ESP8266 by ESP8266 Community ’ and click “Install” button to install the ESP8266 library package. Once installation completed, close and re-open Arduino IDE for ESP8266 library to take effect.

II.5.5 Setup ESP8266 Support

When you've restarted Arduino IDE, select ‘Generic ESP8266 Module’ from the ‘Tools’ -> ‘Board:’ dropdown menu.

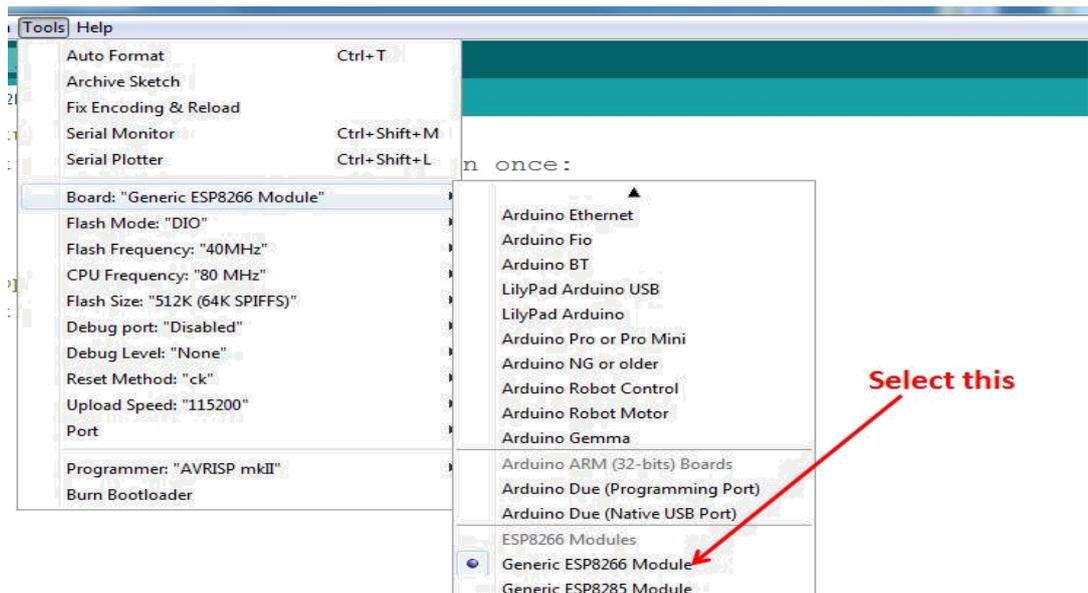


Fig.II.22 Selection of Generic ESP8266 Module

Select '115200' baud upload speed is a good place to start - later on you can try higher speeds but 115200 is a good safe place to start.

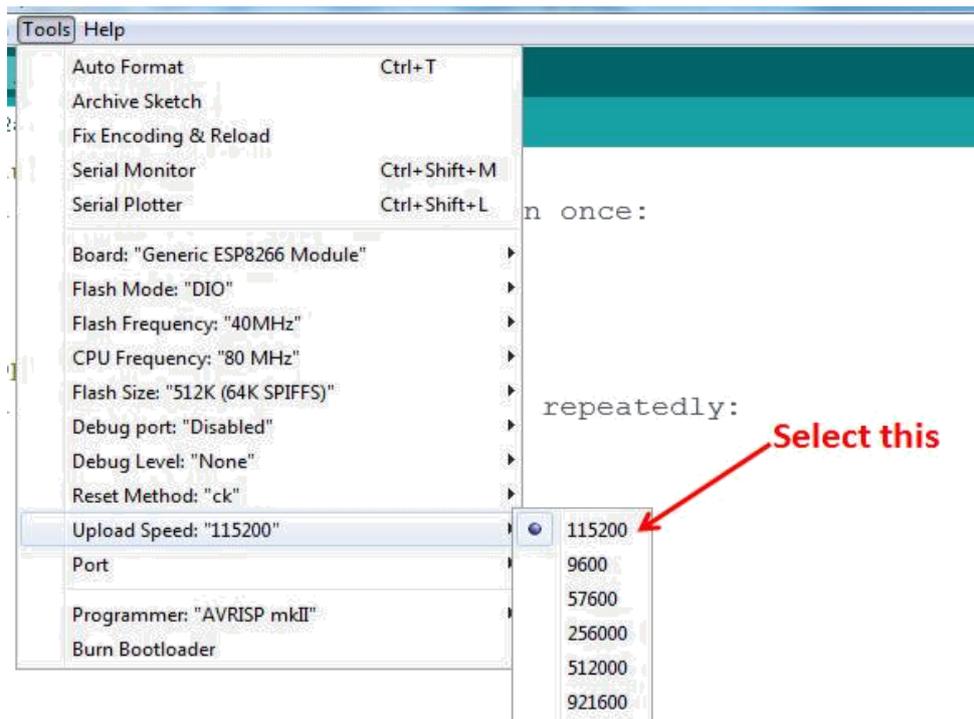


Fig.II.23 Selection of baud rate 115200

Now we are ready to start building, compiling and running sketches, With Node MCU V1.0.

II.6 Conclusion

In this chapter we get know all about the programmable devices of this project as Raspberry Pi 3, Arduino Uno and Node MCU V1.0 WiFi shield with all the installing needed. Like the programmable language for each of the devices and their IDE , because they are the master of this project.

CHAPTER III

III.1 Introduction

This project based on Raspberry Pi3 Board , Arduino Uno and Node MCU V1.0 allows to control Home Security and Automation . In this chapter we going to focus on two parts one it's the Hardware, the second it's the Software. The Hardware is all various components and elements that have been used in this project, The Software part is all programs of this project.

III.2 Electric schematic

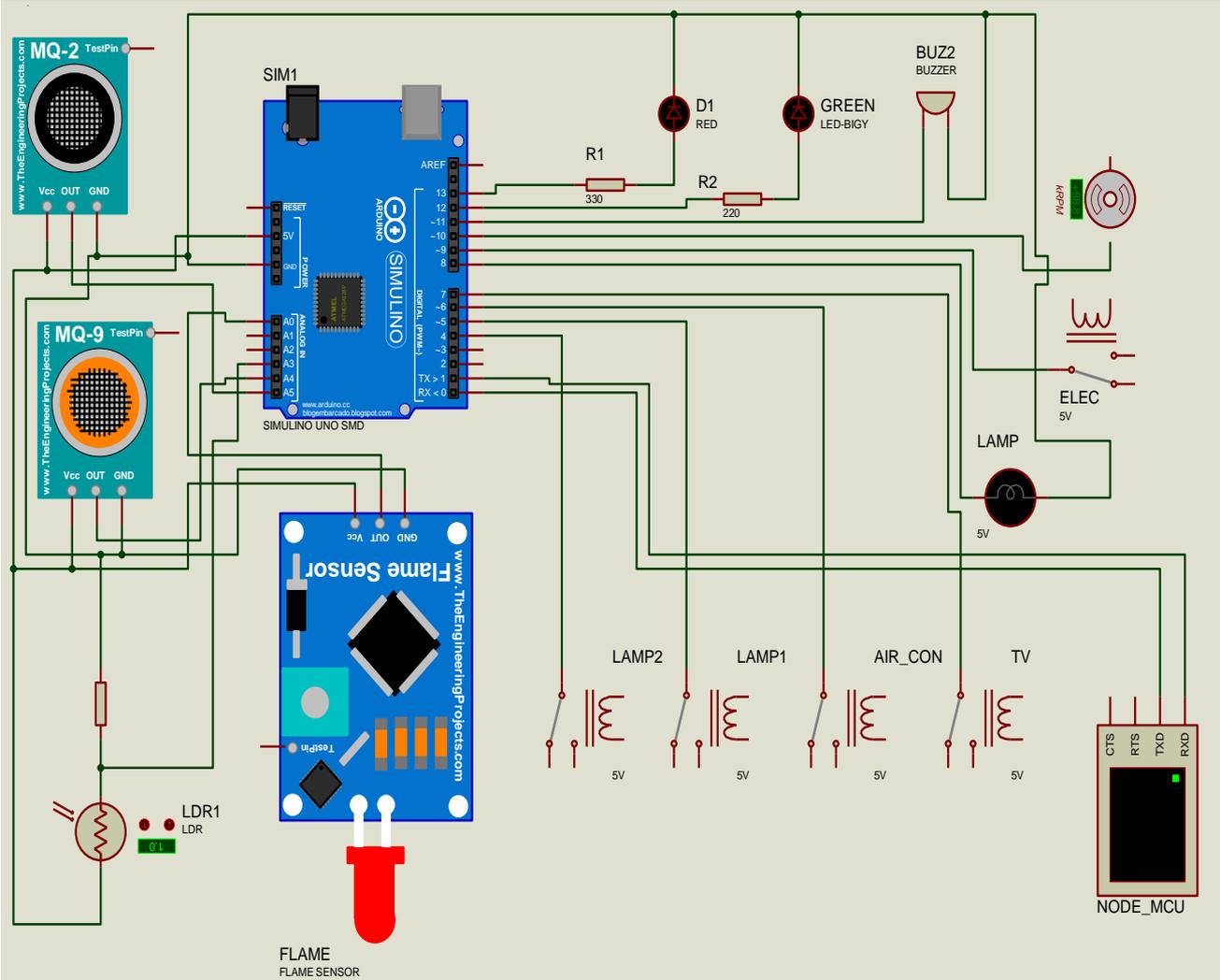


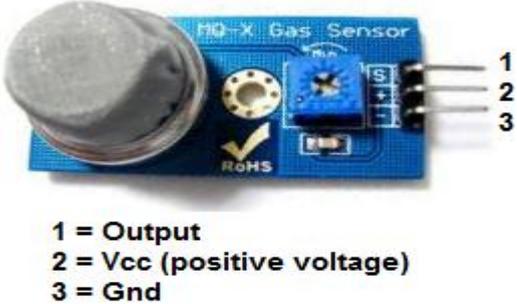
Fig.III.1 The project electric schematic

III.3 Hardware Materials

This section explains the main components used in this project.

III.3.1 MQ2 Smoke Sensor

MQ2 is a flammable gas and smoke sensor which detects the concentrations of combustible gas in the air and outputs reading as an analog voltage. It is sensitive to a wide range of gases and is used at room temperature. Some modules have a built-in variable resistor to adjust the sensitivity of the sensor. It falls under the category of electromechanical gas detectors which work by allowing gases to diffuse through a porous membrane to an electrode where it is either chemically oxidized or reduced. The amount of current produced is determined by how much of the gas is oxidized at the electrode, indicating concentration of the gas. For MQ2, the sensitive material used is SnO₂, whose conductivity is lower in clean air. Its conductivity increases as the concentration of combustible gases increases. [Appendix C]



1 = Output
2 = Vcc (positive voltage)
3 = Gnd

Fig.III.2 MQ2 Smoke Sensor

III.3.2 MQ9 Carbon Monoxide Sensor (CO)

MQ9 is This Carbon Monoxide (CO) and flammable gas sensor detects the concentrations of CO and combustible gas in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of CO from 10 to 1,000 ppm and flammable gas from 100 to 10,000 ppm. The sensor can operate at temperatures from -10 to 50°C and consumes less than 150 mA at 5 V. [Appendix D]



Fig.III.3 MQ9 Carbon Monoxide Sensor (CO)

III.3.3 Flame Sensor

The Flame Sensor is sensitive to the flame and radiation. It also can detect ordinary light source in the range of a wavelength 760nm-1100 nm. It can be used as a flame alarm or in fire fighting robots. [Appendix E]



Fig.III.4 Flame Sensor

III.3.4 Light Dependent Resistor (LDR)

LDR (Light Dependent Resistor) also known as a LDR, photo resistor, photoconductor or photocell, is a resistor whose resistance increases or decreases depending on the amount of light intensity. LDRs are usually made of many semi-conductive materials with high resistance. The reason they have a high resistance is that there are very few electrons that are free and able to move because they are held in crystal lattice and are unable to move. When light falls on the semi conductive material, it absorbs the light photons and the energy is transferred to the electrons, which allow them to break free from the crystal lattice and conduct electricity and lower the resistance of the LDR. [Appendix F]



Fig.III.5 Light Dependent Resistor LDR

III.3.5 Four Channel Relayboard

This is Four Channel relay board is a device which is generally used to control high voltages using very low voltage as an Input. This consists of a coil wrapped around a pole and a two small metal flaps (nodes) that are used to close the circuit. One of the nodes is fixed and other is movable. When ever electricity is passed through the coil, it creates a magnetic field and attracts the moving node towards the static node and the circuit gets completed. So, just by applying small voltage to power up the coil we can actually complete the circuit for the high voltage to travel. Also, as the static node is not physically connected to the coil there is very less chance that the Microcontroller powering the coil gets damaged if something goes wrong.

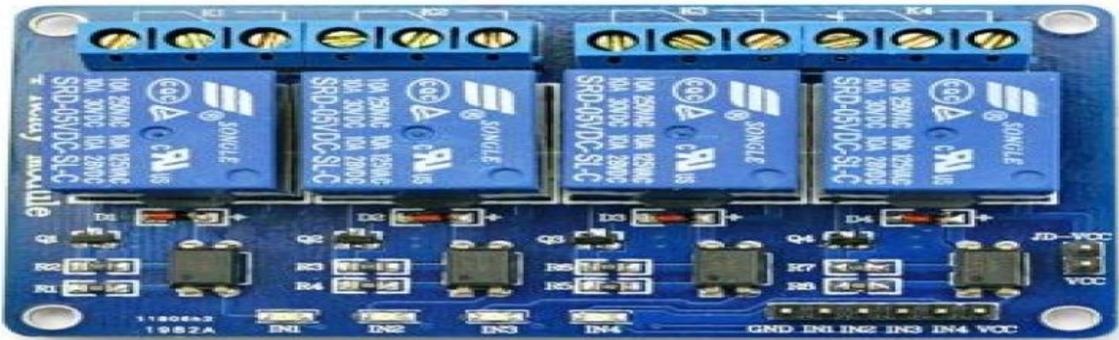


Fig.III.6 Four Channel Relayboard

III.3.6 USB Camera

USB Cameras is a camera that feeds or streams its image in real time to or through a computer to a computer network. It's imaging cameras that use USB technology (USB 2.0 or 3.0) to transfer image data. It designed to easily interface with computer systems by using the same USB technology that is found on most computers. The figure below shows USB camera that low manufacturing cost and their high flexibility, making them the lowest-cost. Used a standard USB webcam to take pictures and video on the raspberry pi connected by a USB cable, or similar cable, when "captured" by the raspberry pi, the pictures and video are saved or sent to other networks via systems such as the internet, and emailed as an attachment.



Fig.III.7 USB Camera

III.3.7 Buzzer

It has two wires. Red and black. Polarity matters: black=ground Apply an oscillating voltage to make a noise. Oscillating voltage alternately squeezes and releases the piezo element. Must apply fluctuating voltage, a steady HIGH or LOW won't work. The buzzer case supports the piezo element and has resonant cavity for sound and will be used as alarm.



Fig.III.8 Buzzer

III.3.8 Fan

The Fan is not for cooling the Home, but will be used for clearing away the air from the smoke, gas leak and Carbon Monoxide (CO) of the House to avoid accident and lost of human lives.



Fig.III.9 Fan

III.3.9 Led

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode, which emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined the energy band gap of the semiconductor.



Fig.III.10 Led

III.3.10 Breadboard

A thin plastic board used to hold electronic components (transistors, resistors, chips, etc.) that are wired together. Used to develop prototypes of electronic circuits, the boards can be reused for future jobs. Breadboards can also be used to create one-of-a-kind systems, although commercial products placed on printed circuit boards are typically much more robust and can handle greater frequencies.

The breadboard contains spring clip contacts typically arranged in matrices with certain blocks of clips already wired together. The components and jump wires (assorted wire lengths with pins at both ends) are plugged into the clips to create the circuit patterns. The boards also typically include metal strips along the side that are used for common power rails and signal buses.

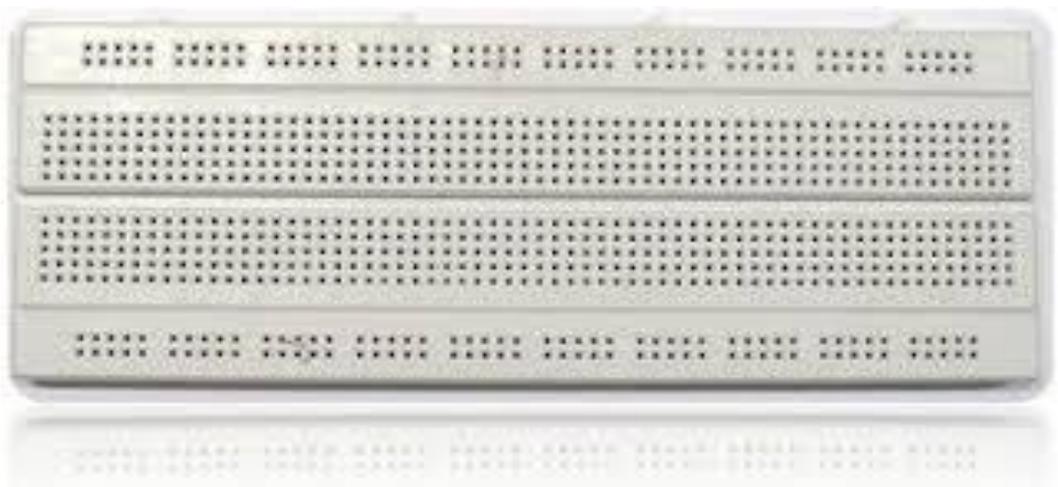


Fig.III.11 Breadboard

III.3.11 Access point

To connect to the home automation system LAN via the ENC28J60 Ethernet module, a gateway is required for the user to access the command application. Am sharing my mobile ethernet, using 4G Mobilis Sim Card, to connect to the hall system.

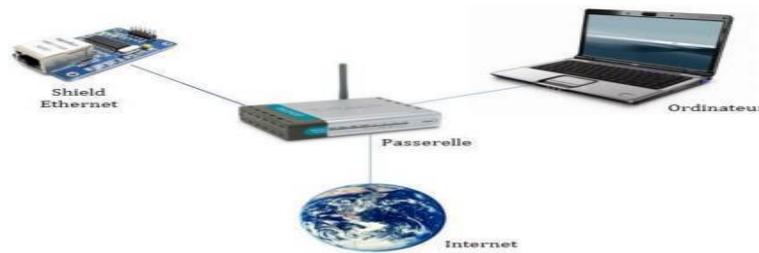


Fig.III.12 Ethernet Network

III.4 Software Materials

This part is dedicated to the all softwares must be installed on the Raspberry Pi 3 and the Codes to be loaded to the Arduino Uno and Node MCU of this application about Home security and automation system.

III.4.1 MQTT Protocol

MQTT stands for **MQ** Telemetry Transport and it is a nice lightweight publish and subscribe system where you can publish and receive messages as a client. It is a simple messaging protocol, designed for constrained devices and with low-bandwidth. So, it's the perfect solution for Internet of Things applications.

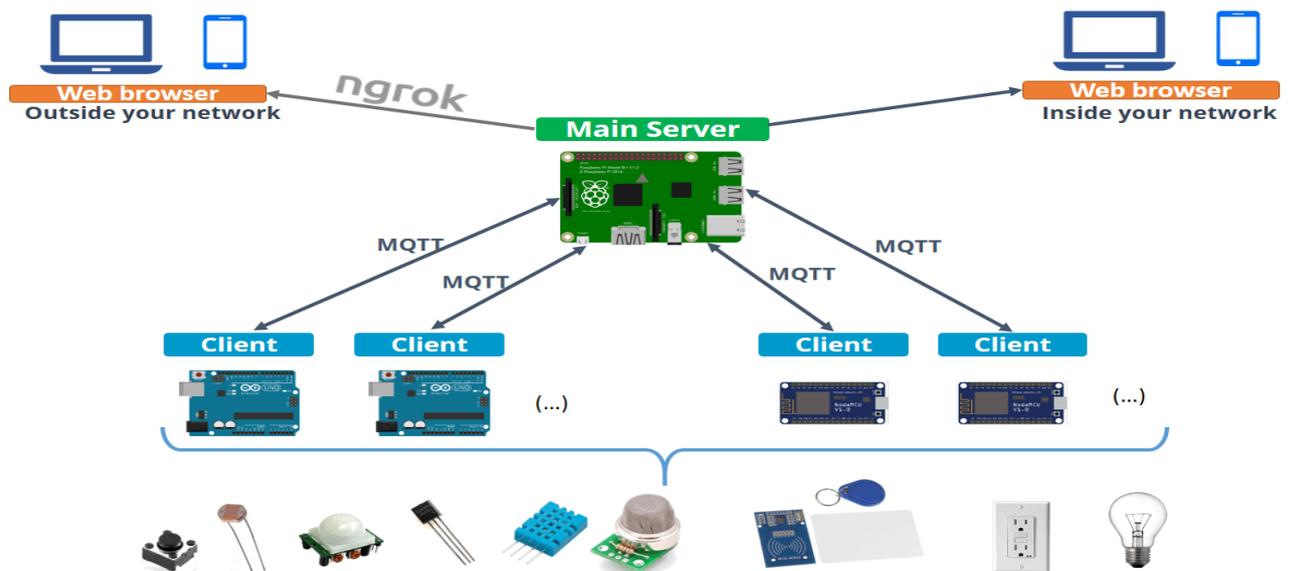


Fig.III.13 Mosquitto Broker

III.4.1.1 Installing Mosquitto Broker

In MQTT, the broker is primarily responsible for **receiving** all messages, **filtering** the messages, **decide** who is interested in it and then **publishing** the message to all subscribed clients.

There are several brokers you can use. In this tutorial we're going to use the **Mosquitto Broker** which needs to be installed on Raspberry Pi.

- **Updating the repository**

To update the repository you need to import the repository package signing key:
pi@raspberrypi:~ \$ **wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key**

```
pi@raspberrypi:~ $ sudo apt-key add mosquitto-repo.gpg.key
```

Make the repository available to be installed with apt:

```
pi@raspberrypi:~ $ cd /etc/apt/sources.list.d/
```

```
pi@raspberrypi:/etc/apt/sources.list.d $
```

Then, run the following command:

```
pi@raspberrypi:/etc/apt/sources.list.d $ sudo wget
```

```
http://repo.mosquitto.org/debian/mosquitto-jessie.list
```

Go back to the root directory:

```
pi@raspberrypi:/etc/apt/sources.list.d $ cd
```

```
pi@raspberrypi:~ $
```

Finally, update apt information:

```
pi@raspberrypi:~ $ sudo apt update
```

- **Installing**

To install the Mosquitto broker and make it auto start enter the following command:

```
pi@raspberrypi:~ $ sudo apt install mosquitto
```

```
pi@raspberrypi:~ $ sudo systemctl enable mosquitto.service
```

You need to type **Y** and press **Enter** to confirm the installation.

III.4.1.2 Testing the Mosquitto Broker

To see if Mosquitto broker was successfully installed, send the following command:

```
pi@raspberrypi:~ $ mosquitto -v
```

This returns the Mosquitto version that is currently running in your Raspberry Pi. It should be 1.4 or above

```
pi@raspberrypi:~ $ mosquitto -v
1498820350: mosquitto version 1.4.12 (build date Thu,
) starting
1498820350: Using default config.
1498820350: Opening ipv4 listen socket on port 1883.
1498820350: Error: Address already in use
```

Fig.III.14 Testing the mosquito

III.4.2 Node-RED

Node-RED is a powerful open source tool for building Internet of Things (IoT) applications with the goal of simplifying the programming component. It uses a visual programming that allows you to connect code blocks, known as **nodes**, together to perform a task. The nodes when wired together are called **flows**. Node-RED is open source and developed by IBM. The Raspberry Pi runs Node-RED perfectly. With Node-RED you can spend more time making cool stuff, rather than spending countless hours writing code.

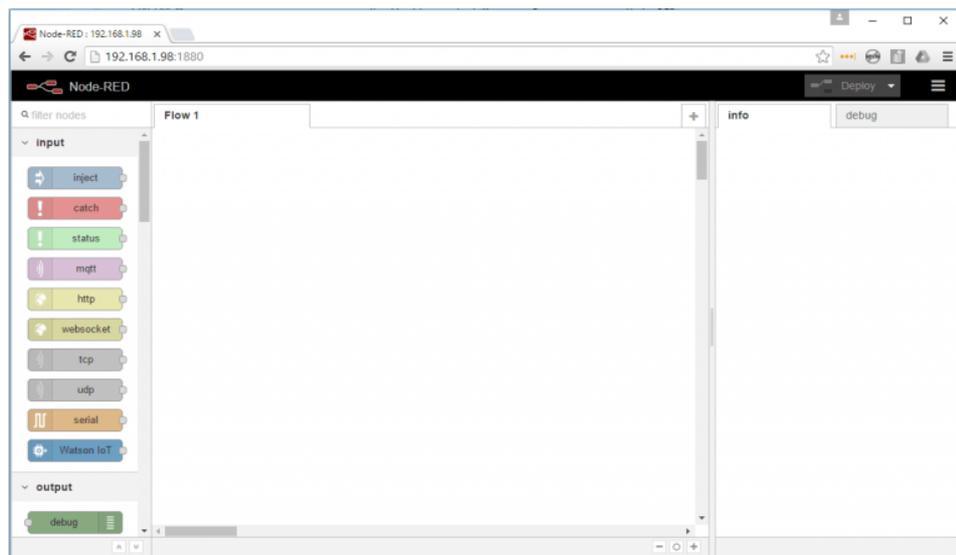


Fig.III.15 Node-RED

Node-RED makes it easy to:

- Access your RPi GPIOs
- Establish an MQTT connection with other boards (Arduino, ESP8266, etc)
- Create a responsive graphical user interface for your projects
- Communicate with third-party services (IFTTT.com, Adafruit.io, Thing Speak, etc)
- Retrieve data from the web (weather forecast, stock prices, emails. etc)
- Create time triggered events
- Store and retrieve data from a database

III.4.2.1 Installing Node-RED

Getting Node-RED installed in your Raspberry Pi is quick and easy. It just takes a few commands.

Having an SSH connection established with your Raspberry Pi, enter the following commands to install Node-RED:

```
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
```

The installation should be completed after a couple of minutes.

Autostart Node-RED on boot

To automatically run Node-RED when the Pi boots up, you need to enter the following command:

```
pi@raspberrypi:~ $ sudo systemctl enable nodered.service
```

Now, restart your Pi so the autostart takes effect:

```
pi@raspberrypi:~ $ sudo reboot
```

III.4.2.2 Testing the Node-RED

When your Pi is back on, you can test the installation by entering the IP address of your Pi in a web browser followed by the **1880** port number:

http://YOUR_RPi_IP_ADDRESS:1880

In my case is:

<http://192.168.43.233:1880>

A page like these loads:

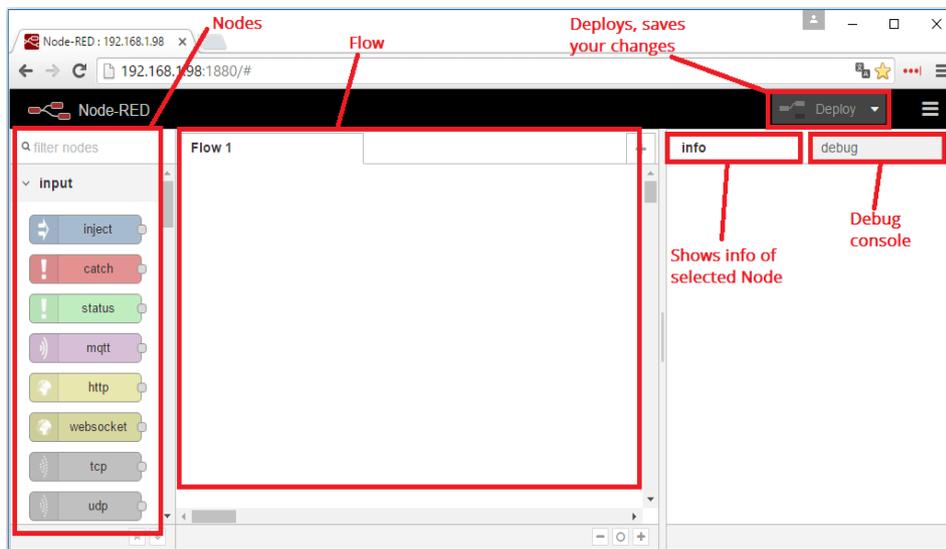


Fig.III.16 Node-RED Description

III.4.2.3 Node-RED Dashboard

To build a graphical user interface.

- **Installing Node-RED Dashboard**

To install the Node-RED Dashboard run the following commands:

```
pi@raspberrypi:~/node-red $ node-red-stop
```

```
pi@raspberrypi:~/node-red $ cd ~/node-red
```

```
pi@raspberrypi:~/node-red $ npm install node-red-dashboard
```

Then, reboot your Pi to ensure that all changes take effect on Node-RED software:

```
pi@raspberrypi:~/node-red $ sudo reboot
```

To open the Node-RED UI, type your [Raspberry Pi IP address](#) in a web browser followed by **:1880/ui** as shown below:

http://Your_RPi_IP_address:1880/ui

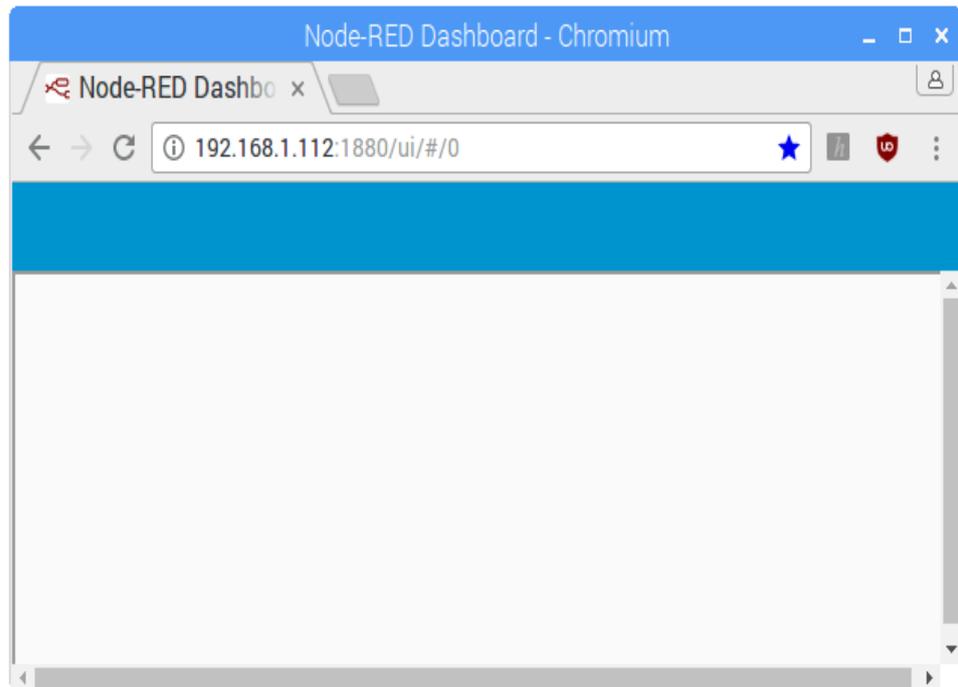


Fig.III.17 Node-RED Dashboard

In this section we're going to create your UI (User Interface) in Node-RED.

- **The Dashboard Layout**

Open another tab in your browser to access Node-RED with:

http://Your_RPi_IP_address:1880

Scroll down on the nodes section. You'll see you have a set of nodes called **dashboard** as shown in the following figure:

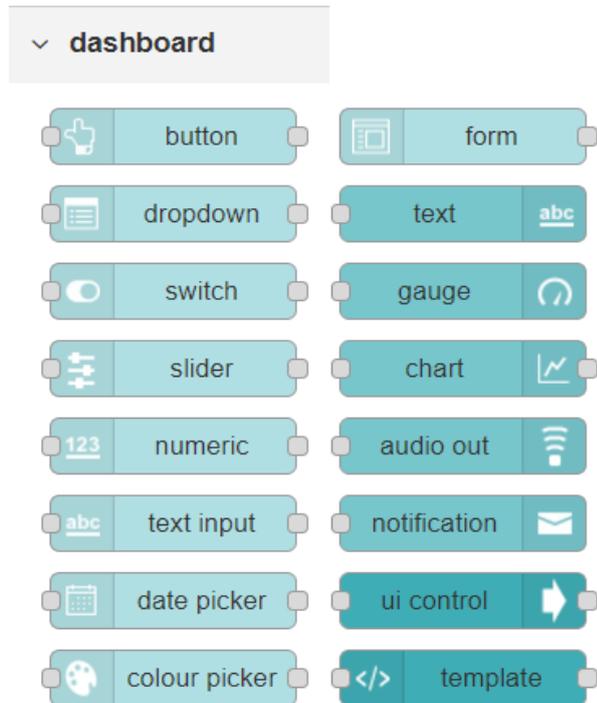


Fig.III.18 Dashboard Nodes

Nodes from the dashboard section provide widgets that show up in your application user interface (UI).

The user interface is organized in **tabs** and **groups**. Tabs are different pages on your user interface, like several tabs in a browser. Inside each tab you have groups that divide the tabs in different sections so that you can organize your widgets. Every widget should have an associated group that determines where the widget should appear on the user interface.

III.4.3 Motion

III.4.3.1 Install the Software Motion on the Raspberry Pi 3

type in the command '**sudo apt-get update**' and '**sudo apt-get upgrade**' one at a time.

Type in the command '**sudo apt-get install motion** ' to start the installation.

Now to make sure that the camera is correctly detected.

Type in the command '**lsusb**' and enter. You should see the name of your camera. If it is NOT there, then there is **some problem** in your camera or the camera is **not supported in 'motion'**.

After the installation is complete, type in the command '**sudo nano /etc/motion/motion.conf**' and press enter.

Then you have to change some settings in the .conf file. It might be difficult sometimes to find the settings but use '**ctrl + w**' to find it. So follow the steps

- Make sure '**daemon**' is **ON**.
- Set '**framerate**' anywhere in between **1000 to 1500**.
- Keep '**Stream_port**' to **8081**.
- '**Stream_quality**' should be **100**.
- Change '**Stream_localhost**' to **OFF**.
- Change '**webcontrol_localhost**' to **OFF**.
- Set '**quality**' to **100**.
- Set '**width**' & '**height**' to **640 & 480**.
- Set '**post_capture**' to **5**.
- Press **ctrl + x** to exit. Type **y** to save and **enter** to conform.

Again type in the command '**sudo nano /etc/default/motion** ' and press **enter**.

Set '**start_motion_daemon** ' to **yes**. Save and exit. [Appendix I]

III.4.3.2 Testing Motion

First of all you have to **restart** the motion software. To do it type in the command '**sudo service motion restart**' and press **enter**.

Again type in the command '**sudo motion**' and press **enter**. Now your server is ready.

Now open up your browser. Type in the IP address of your raspberry Pi and the port number in this way:

192.168.43.233:8081 (First there is the IP address, then a ':', then the port number). Press **Enter**.

Now **you can see the Live feed coming from your webcam** directly on your laptop or mobile or both at the same time.

III.4.4 Arduino Uno Code

The code will be loaded to the Arduino Uno. [Appendix G].

III.4.5 Node MCU V1.0 Code

The code will be loaded to the Node MCU. [Appendix H].

III.5 Results and Discussions

The project on two parts, Security applications and Smart Home appliances.

III.5.1 Security applications

Using sensors reading to control several applications for home security. Like a gas leak , Smoke, Fire, Detecting Carbon Monoxide CO is real killer. To prevent all these accident using a Home Security application

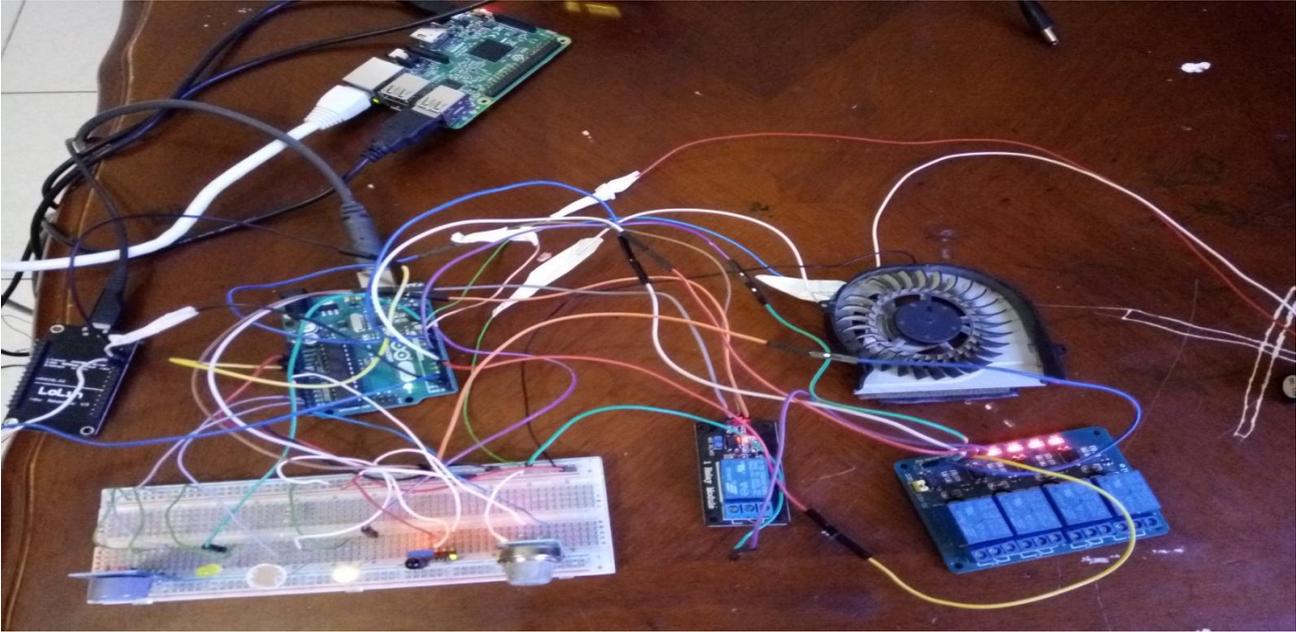


Fig.III.19 snapshot of Security & Home Automation

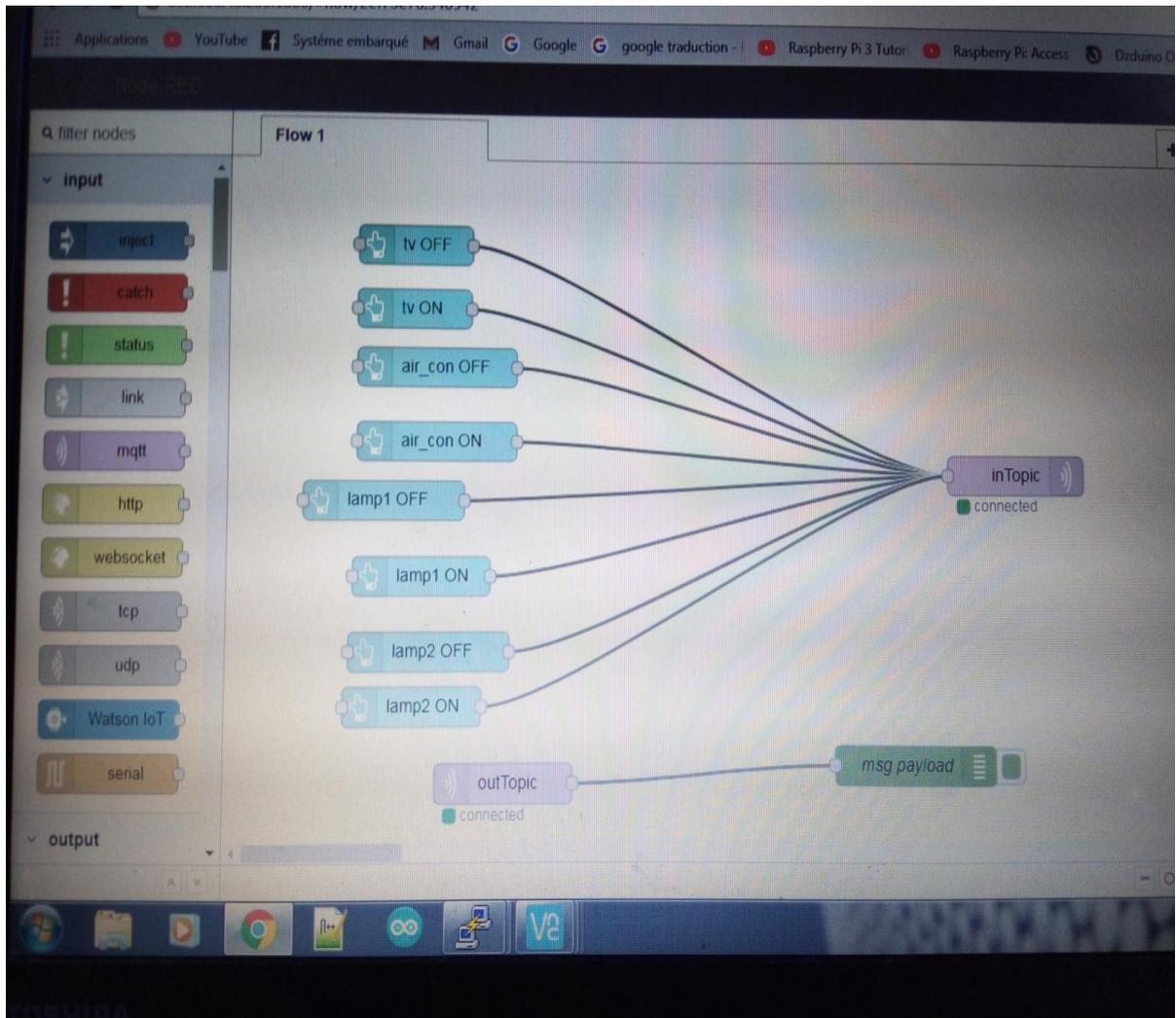


Fig.III.20 snapshot of Home Appliances from Node-RED

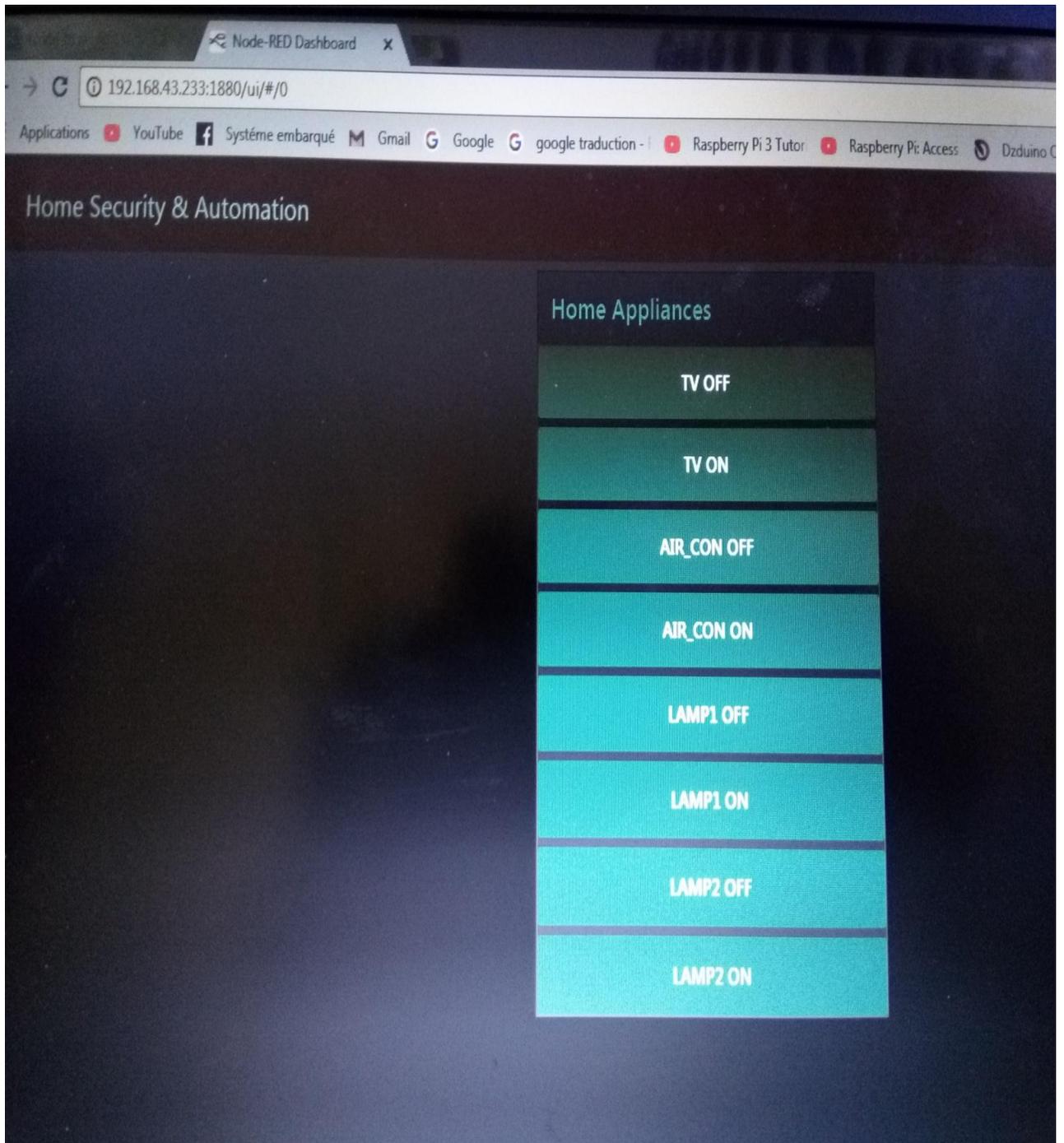


Fig.III.21 snapshot of Home Appliances from the browser

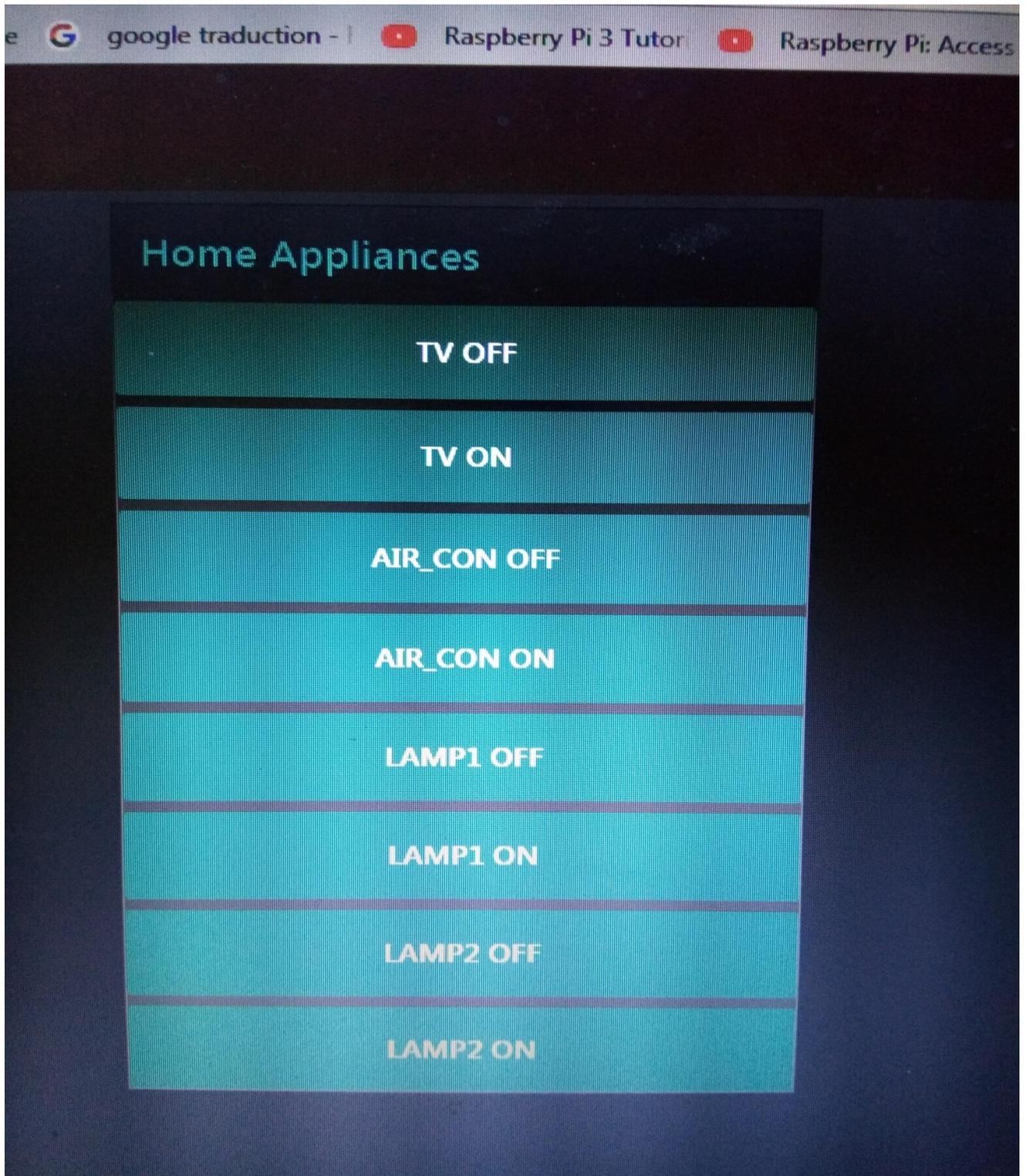


Fig.III.22 snapshot of Home Appliances from my laptop

- The chart Flow of these applications

- Detecting smoke and gas leak using MQ2 sensor

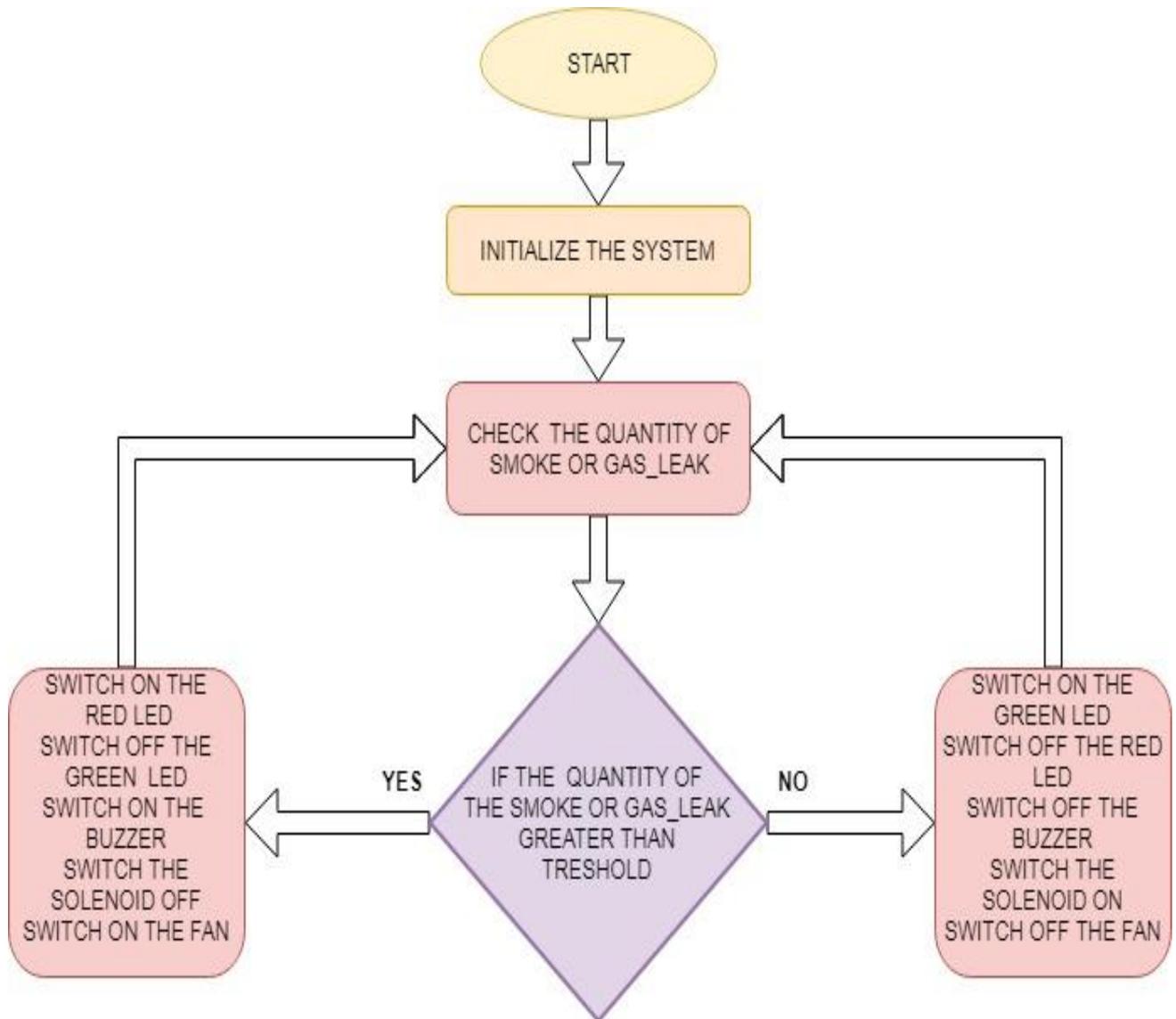


Fig.III.23 Chart Flow for gas leak & smoke

- Detecting Fire using Flame Sensor

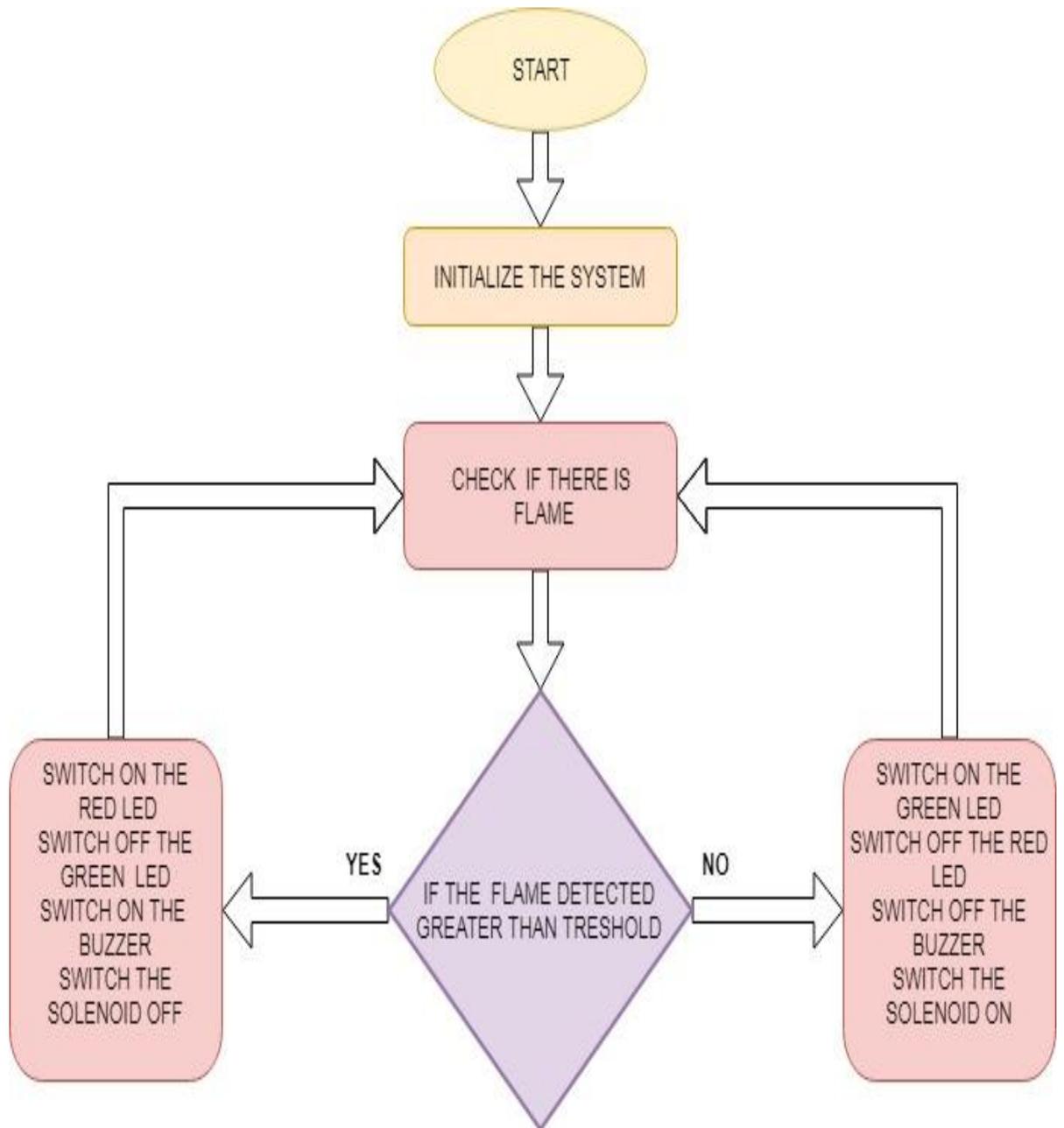


Fig.III.24 Chart Flow for Detecting a Flame

- Detecting Carbon monoxide using MQ9

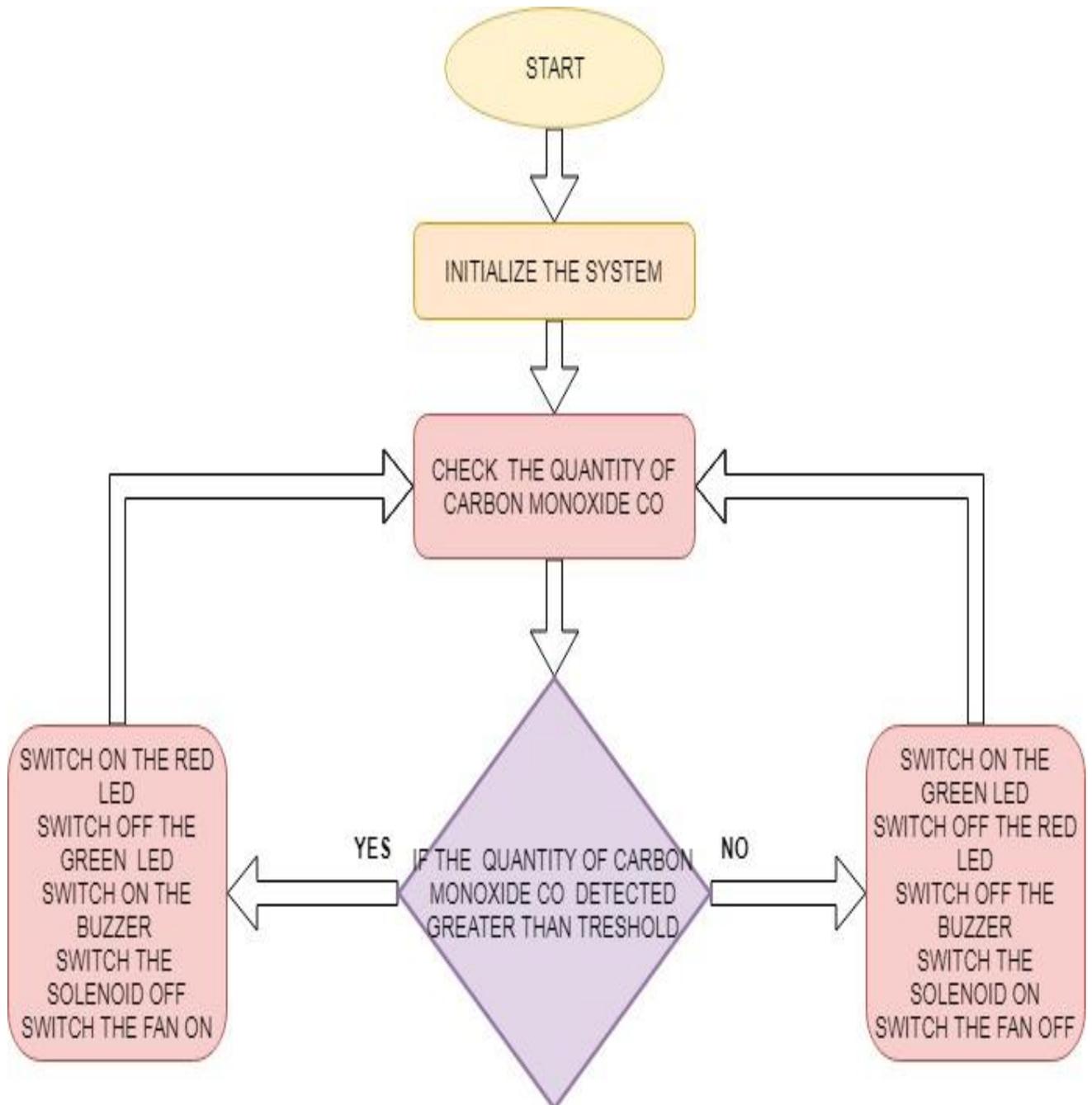


Fig.III.25 Chart Flow for Detecting Carbon Monoxide CO

- Detecting Light using LDR

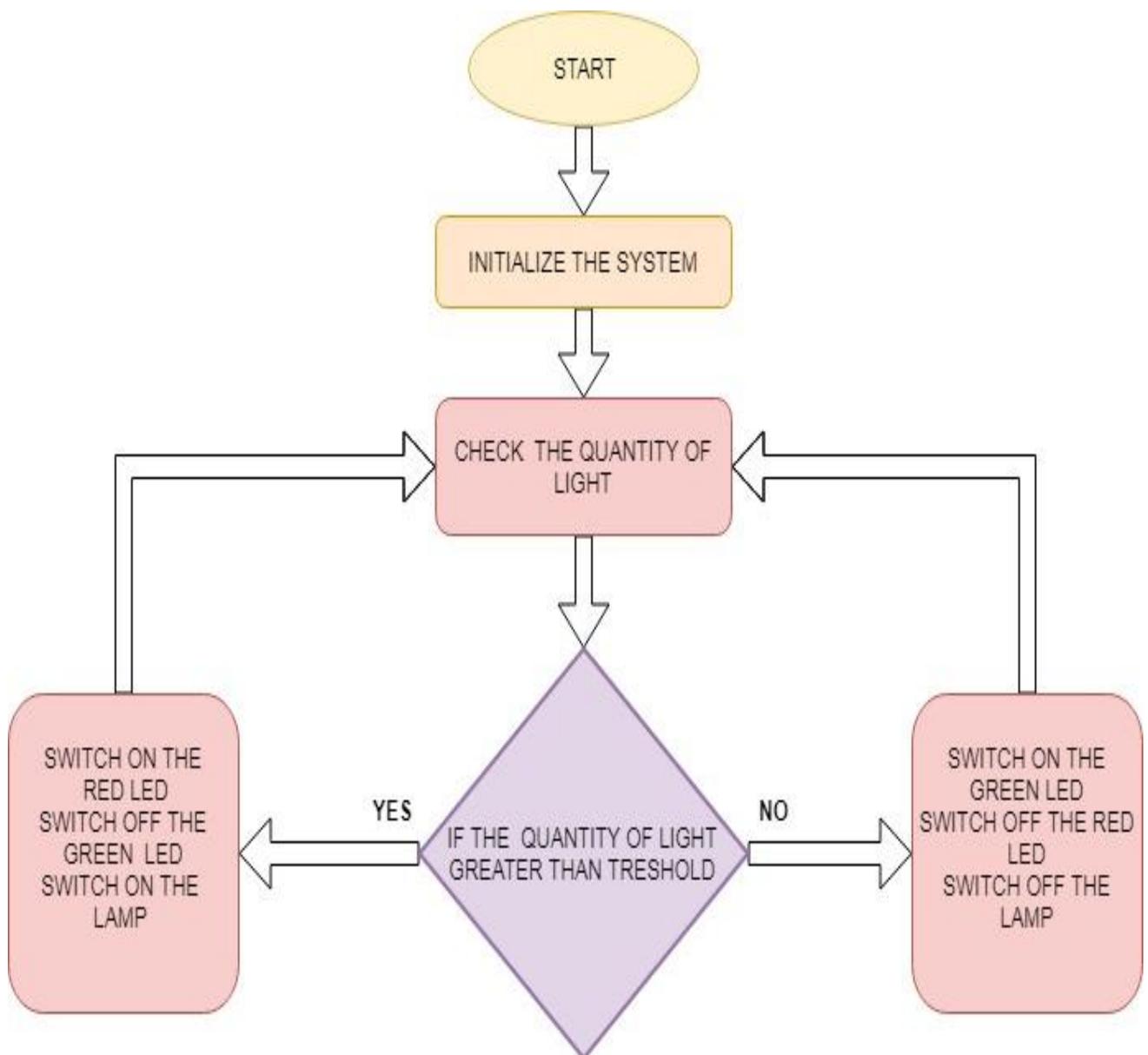


Fig.III.26 Chart Flow for Detecting Light

III.6 Conclusion

A significant transformation is to extend "things" from the data generated from devices to objects in the physical space. The thought-model for future interconnection environment . The model includes the notion of the ternary universe consists of the physical world, virtual world and mental world and a multi-level reference architecture with the nature and devices at the bottom level followed by the level of the Internet, sensor network, and mobile network, and intelligent human-machine communities at the top level, which supports geographically dispersed users to cooperatively accomplish tasks and solve problems by using the network to actively promote the flow of material, energy, techniques, information, knowledge, and services in this environment. This thought model envisioned the development trend of the Internet of things.

GENERAL CONCLUSION

We living in the world were technology its reaching high speed communication between human to machine and machine to machine(M2M) , even sharing information as well, all interconnected over public or private Internet protocol (IP). These interconnected Objects have data regularly collected, analyzed and used to initiate action, providing a supreme of intelligence for planning, management and decision making. IoT much more than things connected to the Internet. IoT is about making dumb things smarter by giving them the ability to sense, communicate, and respond.

A significant transformation is to extend "things" from the data generated from devices to objects in the physical space. The thought-model for future interconnection environment . The model includes the notion of the ternary universe consists of the physical world, virtual world and mental world and a multi-level reference architecture with the nature and devices at the bottom level followed by the level of the Internet, sensor network, and mobile network, and intelligent human-machine communities at the top level, which supports geographically dispersed users to cooperatively accomplish tasks and solve problems by using the network to actively promote the flow of material, energy, techniques, information, knowledge, and services in this environment. This thought model envisioned the development trend of the Internet of things.

Whenever smart home security system detects someone who is unknown, it can provide video of the visitor to the homeowner. Visitors that are welcome can be given clearance and allowed in the house remotely. Unwelcome visitors can be ignored, and individuals attempting to break in will trigger a email or message to the home owner. The target of this project is to achieve is making life easy and safe. The use of protocol like MQTT and graphical interface like Node-RED. I wish that the students coming after us can gain time and experience in field like security and Home Automation. In mater like the use of many types of sensors, Raspberry Pi3, Node MCU and Arduino Uno. The use of live streaming with web cam and software called Motion. I faced to many problems, I bought too many components, I would like to use LCD and to let the webcam sends emails when detect motion, Even to let your home talk to you in way you can monitor all electrical devices.

Bibliography

- [1] Peter Waher, Learning Internet of Things: Packt PUBLISHING

- [2] Méziane Boudellal, Habitat connecté, installations domotiques et multimédia : DUNOD

- [3] Marco Schwarz, Internet of Things with Arduino: Packt PUBLISHING

- [4] Cédric Locqueneux, LE GUIDE DE LA MAISON ET DES OBJETS CONNECTES : EYROLLES 2015 FRANCE

- [5] Andrew K. Dennis, Raspberry Pi Home Automation with Arduino: Packt PUBLISHING: BIRMINGHAM 2013

- [6] Andrew K. Dennis, Raspberry Pi Home Automation with Arduino Second Edition: Packt PUBLISHING: BIRMINGHAM 2015

- [7] Rick Golden, Raspberry Pi Networking Second Edition: Packt PUBLISHING: BIRMINGHAM 2015

- [8] Adeel Javed, Building Arduino Projects for the Internet of Things: Apress, illinot USA 2016

- [9] Tanay Pant, Building a Virtual Assistant for Raspberry Pi: Apress, Uttar Pradesh India 2016

- [10] Jorge R. Castro, Building a Home Security System with Arduino: Packt PUBLISHING, BIRMINGHAM 2015

- [11] Simo Landrault, Hippolyte Weissliger, Premiers pas en informatique embarquée : le blog d'Eskimon, 2014

- [12] Satya Sankar Sahoo, Getting Started with MQTT a practical Guide

Appendix

Appendix A

However, most of the boards out there are model B+ or 2, so you are more likely to end up with them. B+ is the most common and updated version of model B, offering four USB ports and other features. And the Raspberry pi 2 is the latest version that offers 1GB RAM and a processing power of a quad-core 900MHz processor. The pi 2 is about six times faster than the pi and even supports windows 10.

The basic specifications of the Raspberry Pi A, A+, B, B+ and pi 2 are

	Model A	Model A+	Model B	Model B+	Generation 2 Model B	Compute Module Note: all interfaces are via 200-pin DDR2 SO-DIMM connector.
Target price:	US\$25		US\$20	US\$35		US\$30 (in batches of 100)
SoC:	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port) ¹				Broadcom BCM2836	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, one USB port) ¹
CPU:	700 MHz single-core ARM1176JZF-S ¹				900 MHz quad-core ARM Cortex-A7	700 MHz single-core ARM1176JZF-S
GPU:	Broadcom VideoCore IV @ 250 MHz; OpenGL ES 2.0 (24 GFLOPS) MPEG-2 and VC-1 (with license), ¹ 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder ¹¹					
Memory (SDRAM):	256 MB (shared with GPU)		512 MB (shared with GPU) as of 15 October 2012		1 GB (shared with GPU)	512 MB (shared with GPU)
USB 2.0 ports: ¹	1 (direct from BCM2835 chip)		2 (via the on-board 3-port USB hub) ¹	4 (via the on-board 5-port USB hub) ¹		1 (direct from BCM2835 chip)
Video input:	15-pin MIPI camera interface (CSI) connector, used with the Raspberry Pi camera or Raspberry Pi NoIR camera ¹					2x MIPI camera interface (CSI) ¹
Video outputs:	HDMI (rev 1.3 & 1.4), 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards, composite video (PAL and NTSC)		HDMI (rev 1.3 & 1.4), 14 HDMI resolutions from	HDMI (rev 1.3 & 1.4), 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards,	HDMI (rev 1.3 & 1.4), 14 HDMI resolutions from 640x350 to 1920x1200 plus various PAL and NTSC standards, composite video (PAL and NTSC) via 3.5 mm TRRS jack shared	
			plus various PAL and NTSC standards,			HDMI, 2x MIPI display interface (DSI), ¹ MIPI display interface (DSI) for raw

Appendix A

	via RCA jack	640x350 to	composite video (PAL and NTSC) via RCA jack	with audio out	LCD panels, composite video	
Audio inputs:	As of revision 2 boards, I ² S					
Audio outputs:	Analog via 3.5 mm phone jack; digital via HDMI and, as of revision 2 boards, I ² S				Analog, HDMI, I ² S	
On-board storage:	SD / MMC / SDIO card slot (3.3 V with card power only)	Mi-	SD / MMC / SDIO card slot	MicroSD slot	4 GB eMMC flash memory chip ^[34] may or may not support external SD cards with configuration changes	
On-board network:	None		10/100 Mbit/s Ethernet (8P8C) USB adapter on the third/fifth port of the USB hub (SMSC lan9514-jzx)		None	
Low-level peripherals:	8x GPIO ^[48] plus the following, which can also be used as			17x GPIO plus the same specific functions, and HAT ID bus	46x GPIO, some of which can be used for specific functions includ-	
	300 mA (2.5 W) ⁱ	200 mA (1 W) ^[53]	700 mA (3.5 W)	600 mA (3.0 W) ⁱ	900 mA (4.5 to 5.5 W)	similar to Model A+
Power source:	5 V via MicroUSB or GPIO header				5 V	
Size:	85.60 mm × 56.5 mm (3.370 in × 2.224 in) – not including protruding connectors	65 mm	85.60 mm × 56.5 mm (3.370 in × 2.224 in) – not including protruding connectors		67.6 mm × 30 mm (2.66 in × 1.18 in)	
Weight:	45 g (1.6 oz)	23 g (0.81 oz)	45 g (1.6 oz)		7 g (0.25 oz) ⁱ	
	Model A	Model A+	Model B	Model B+	Generation 2 Model B	Compute Module

Appendix B

Step 1: Download Raspbian



The screenshot shows two download options for Raspbian Jessie. On the left is 'RASPBIAN JESSIE WITH PIXEL', described as an image with the PIXEL desktop based on Debian Jessie. It has a version of April 2017, a release date of 2017-04-10, and kernel version 4.4. On the right is 'RASPBIAN JESSIE LITE', described as a minimal image based on Debian Jessie, with the same version, release date, and kernel version. Both options include 'Download Torrent' and 'Download ZIP' buttons. Below each option is a SHA-1 hash: 6d7b11bb3d64524203edf6c80c499456fb5fef53 for the PIXEL version and c24a4c7dd1a5957f303193fee712d0d2c0c6372d for the LITE version.

Raspbian Jessie Lite File Image

2: Unzip the file

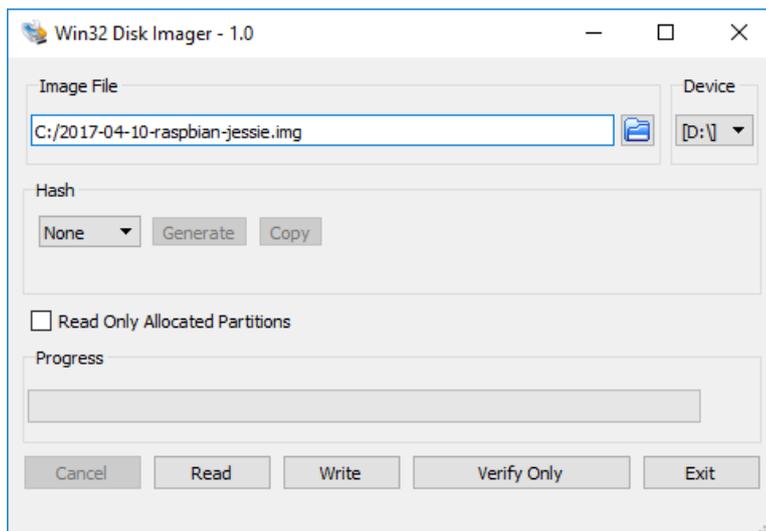
Step The Raspbian disc image is compressed, so you'll need to unzip it. The file uses the ZIP64 format, so depending on how current your built-in utilities are, you need to use certain programs to unzip it. If you have any trouble, try these programs recommended by the Raspberry Pi Foundation:

- Windows users, you'll want **7-Zip**

Step 3: Write the disc image to your micro SD card

Next, pop your micro SD card into your computer and write the disc image to it. You'll need a specific program to do this:

Windows users, your answer is **Win32 Disk Imager**



Unzip Raspbian Jessie with win32 Disk Imager

Appendix B

The process of actually writing the image will be slightly different across these programs, but it's pretty self-explanatory no matter what you're using. Each of these programs will have you select the destination (make sure you've picked your microSD card!) and the disc image (the unzipped Raspbian file). Choose, double-check, and then hit the button to write.

Step 4: Put the micro SD card in your Pi and boot up

Once the disc image has been written to the micro SD card, you're ready to go! Put that sucker into your Raspberry Pi, plug in the peripherals and power source, and enjoy. The current edition to Raspbian will boot directly to the desktop. Your default credentials are username **pi** and password **raspberry**.

MQ-2 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is SnO₂, which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

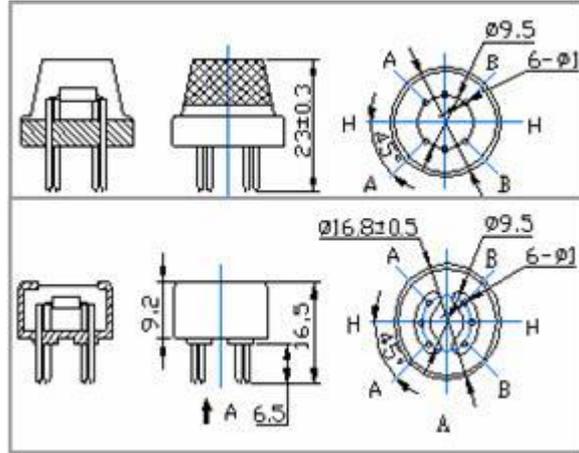
Character

- * Good sensitivity to Combustible gas in wide range
- * High sensitivity to LPG, Propane and Hydrogen
- * Long life and low cost
- * Simple drive circuit

Application

- * Domestic gas leakage detector
- * Industrial Combustible gas detector
- * Portable gas detector

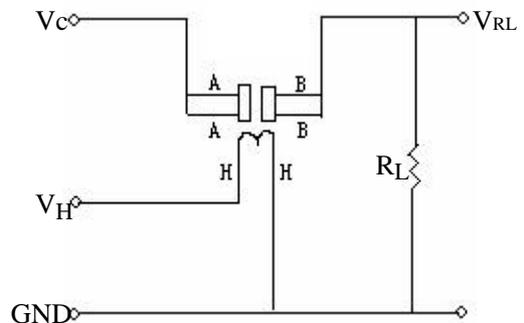
Configuration



Technical Data

Model No.		MQ-2	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Combustible gas and smoke	
Concentration		300-10000ppm (Combustible gas)	
Circuit	Loop Voltage	V _c	≤24V DC
	Heater Voltage	V _H	5.0V±0.2V AC or DC
	Load Resistance	R _L	Adjustable
Character	Heater Resistance	R _H	31Ω±3Ω (Room Tem.)
	Heater consumption	P _H	≤900mW
	Sensing Resistance	R _s	2KΩ-20KΩ (in 2000ppm C ₃ H ₈)
	Sensitivity	S	R _s (in air)/R _s (1000ppm isobutane) ≥ 5
	Slope	α	≤ 0.6 (R _{5000ppm} /R _{3000ppm} CH ₄)
Condition	Tem. Humidity	20°C±2°C; 65%±5%RH	
	Standard test circuit	V _c : 5.0V±0.1V; V _H : 5.0V±0.1V	
	Preheat time	Over 48 hours	

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage (V_H) and test voltage (V_C). V_H used to supply certified working temperature to the sensor, while V_C used to detect voltage (V_{RL}) on load resistance (R_L) whom is in series with sensor. The sensor has light polarity, V_c need DC power. V_c and V_H could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable R_L value is needed: Power of Sensitivity body(P_s):

$$P_s = V_c^2 \times R_s / (R_s + R_L)^2$$

Resistance of sensor(R_s): $R_s=(V_c/V_{RL}-1)\times R_L$

Sensitivity Characteristics

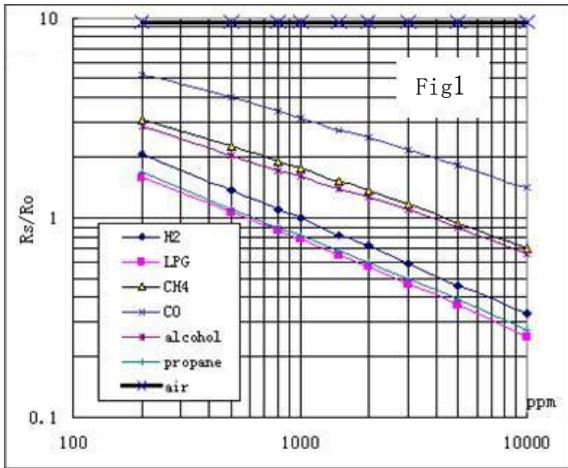


Fig.1 shows the typical sensitivity characteristics of the MQ-2, ordinate means resistance ratio of the sensor (R_s/R_o), abscissa is concentration of gases. R_s means resistance in different gases, R_o means resistance of sensor in 1000ppm Hydrogen. All test are under standard test conditions.

Influence of Temperature/Humidity

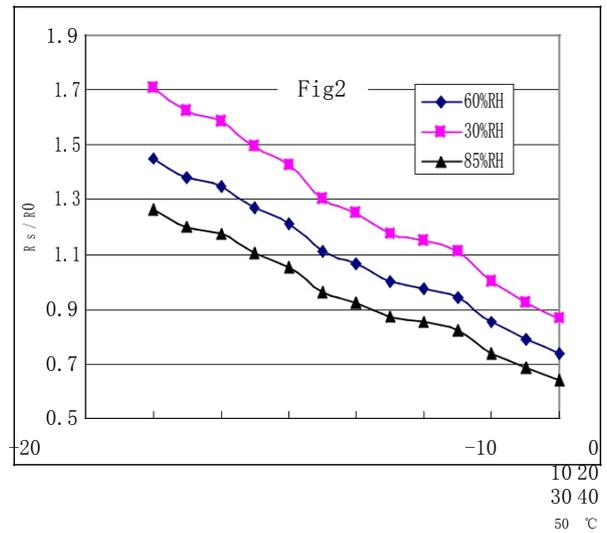
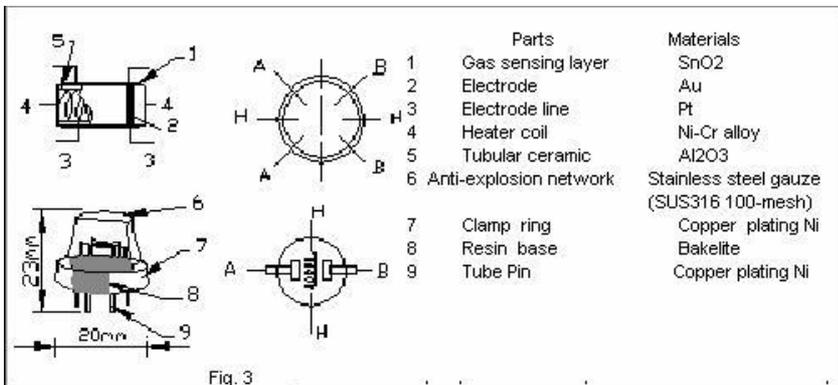


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (R_s/R_o), R_s means resistance of sensor in 1000ppm Butane under different tem. and humidity. R_o means resistance of the sensor in environment of 1000ppm Methane, 20°C/65%RH

Structure and configuration



Structure and configuration of MQ-2 gas sensor is shown as Fig. 3, sensor composed by micro AL2O3 ceramic tube, Tin Dioxide (SnO2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2S , SO_x , Cl_2 , HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

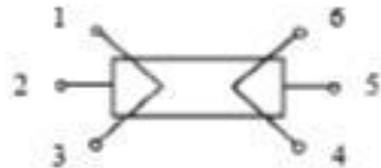
Do avoid icing on sensor's surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, it will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stbility before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature: $100 \pm 20^\circ C$

2.7.4 Welding temperature: $250 \pm 10^\circ C$

2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.

MQ-9 Semiconductor Sensor for CO/Combustible Gas

Sensitive material of MQ-9 gas sensor is SnO₂, which with lower conductivity in clean air. It make detection by method of cycle high and low temperature, and detect CO when low temperature (heated by 1.5V). The sensor's conductivity is more higher along with the gas concentration rising. When high temperature (heated by 5.0V), it detects Methane, Propane etc combustible gas and cleans the other gases adsorbed under low temperature. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-9 gas sensor has high sensitivity to Carbon Monoxide, Methane and LPG. The sensor could be used to detect different gases contains CO and combustible gases, it is with low cost and suitable for different application.

Character

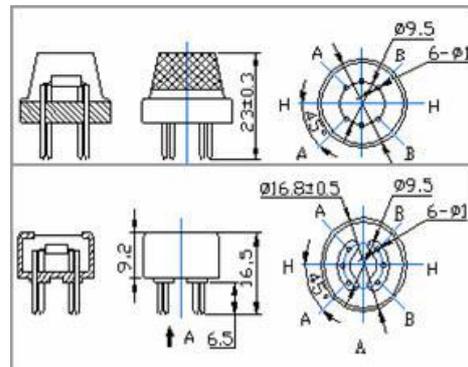
- * Good sensitivity to CO/Combustible gas
- * High sensitivity to Methane, Propane and CO
- * Long life and low cost
- * Simple drive circuit

Application

- * Domestic gas leakage detector
- * Industrial gas detector
- * Portable gas detector

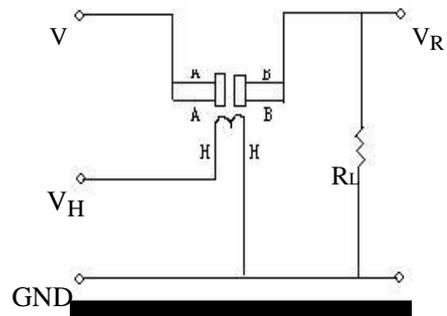
Technical Data

Configuration



Model No.		MQ-9	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite	
Detection Gas		CO and combustible gas	
Concentration		10-1000ppm CO 100-10000ppm combustible gas	
Circuit	Loop Voltage	V _c	≤10V DC
	Heater Voltage	V _H	5.0V±0.2V AC or DC (High) 1.5V±0.1V AC or DC (Low)
	Heater Time	T _L	60±1S (High) 90±1S (Low)
	Load Resistance	R _L	Adjustable
Character	Heater Resistance	R _H	31Ω±3Ω (Room Tem.)
	Heater consumption	P _H	≤350mW
	Sensing Resistance	R _s	2KΩ-20KΩ(in 100ppm CO)
	Sensitivity	S	R _s (in air)/R _s (100ppm CO) ≥5
	Slope	α	≤0.6(R _{300ppm} /R _{100ppm} CO)
Condition	Tem. Humidity	20°C±2°C; 65%±5%RH	
	Standard test circuit	V _c :5.0V±0.1V; V _H (High) : 5.0V±0.1V; V _H (Low) : 1.5V±0.1V	
	Preheat time	Over 48 hours	

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage (V_H) and test voltage (V_C). V_H used to supply certified working temperature to the sensor, while V_C used to detect voltage (V_R) on load resistance (R_L) whom is in series with sensor. The sensor has light polarity, V_c need DC power. V_C and V_H could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better

performance, suitable R_L value is needed:

$$\text{Power of Sensitivity body (Ps): } P_s = V_c^2 \times R_s / (R_s + R_L)^2$$

Resistance of sensor(R_s): $R_s=(V_c/V_{RL}-1)\times R_L$

Sensitivity Characteristics

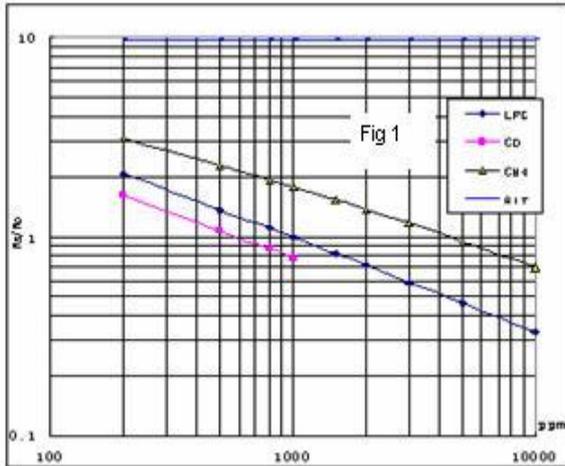


Fig.1 shows the typical sensitivity characteristics of the MQ-9, ordinate means resistance ratio of the sensor (R_s/R_o), abscissa is concentration of gases. R_s means resistance in different gases, R_o means resistance of sensor in 1000ppm LPG. All test are under standard test conditions.

Influence of Temperature/Humidity

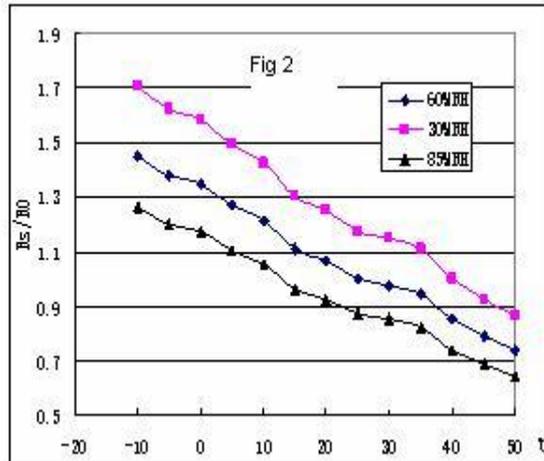
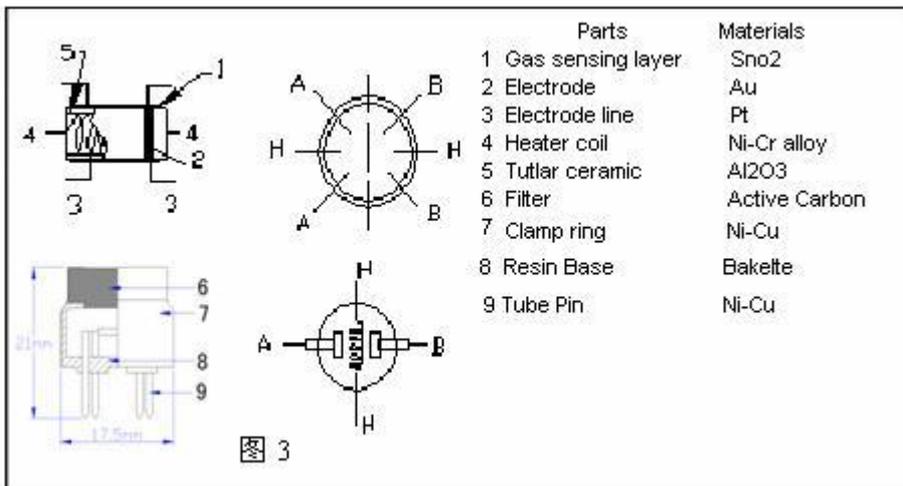


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (R_s/R_o), R_s means resistance of sensor in 1000ppm Propane under different tem. and humidity. R_o means resistance of the sensor in environment of 1000ppm Propane, 20°C/65%RH

Structure and configuration



Structure and configuration of MQ-9 gas sensor is shown as Fig. 3, sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-7 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2S , SO_x , Cl_2 , HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

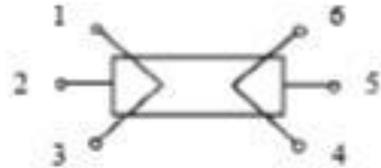
Do avoid icing on sensor's surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, if will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stbility before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature: $100\pm 20^{\circ}C$

2.7.4 Welding temperature: $250\pm 10^{\circ}C$

2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.

Flame Sensor User Manual

1. Features

Voltage comparator chip	LM393 (wide voltage range)
Detection wavelength	760nm-1100nm
Operating voltage	3.3V-5.3V
Detection angle	0 degree-60 degree
Operating temp.	-25°C-85°C
Dimensions	29.2mm*11.2mm
Fixing hole size	2.0mm

2. Applications

This module can be applied to fire detection system, fire-fighting robot, fire alarm system, etc.

3. Interfaces

Pin No.	Symbol	Descriptions
1	DOUT	Digital output
2	AOUT	Analog output
3	GND	Power ground
4	VCC	Positive power supply (3.3V-5.3V)

4. How to use

We will illustrate the usage of the module with an example of fire detection by connecting a development board.

- ① Download the relative codes to the development board.
- ② Connect the development board to a PC via a serial wire and the module to the development board. Then, power up the development board and start the serial debugging software.

Here is the configuration of the connection between the module and the development board.

Port	STM32 MUC pin
DOUT	GPIOA.4
AOUT	GPIOA.6
GND	GND
VCC	3.3V

Port	Arduino pin
DOUT	D2
AOUT	A0
GND	GND
VCC	5V

Here is the configuration of the serial port.

Baud rate	115200
Data bits	8
Stop bit	1
Parity bit	None

- ③ The detected result can be checked by a signal indicator on the module. The signal indicator will turn on, when the sensor is close to a fire. And it will turn off, when the sensor is away from the fire.
- ④ Start the serial debugging software, and you can find the serial output changes along with the distance from the sensor to the fire.

Notice: The flame sensor is designed to detect fire only, but it is not fireproof itself. When using it, please keep a safe distance from the fire to avoid burning out.



Data Sheet

Light dependent resistors

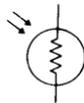
NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol



Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (RS stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

T_A = 25°C. 2854°K tungsten light source

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux	-	400	-	Ω
	10 lux	-	9	-	kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux	-	2.8	-	ms
	10 lux	-	18	-	ms
Fall time 2	1000 lux	-	48	-	ms
	10 lux	-	120	-	ms

1. Dark to 110% R_L

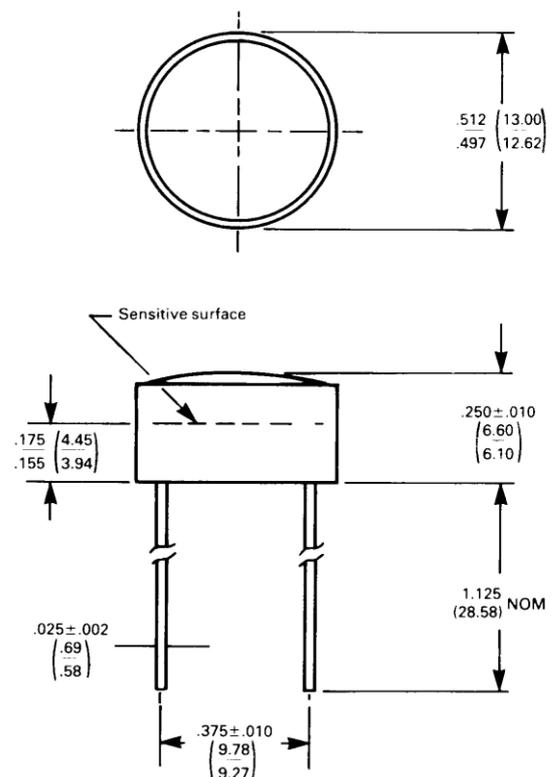
2. To 10/3 R_L

R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions



Units in inches (millimetres)

Figure 1 Power dissipation derating

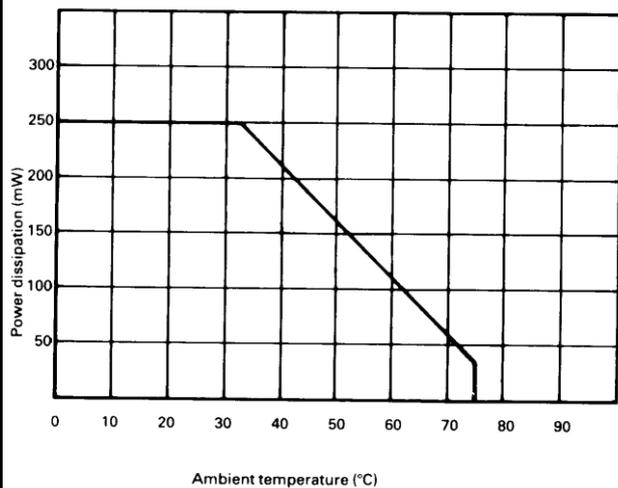
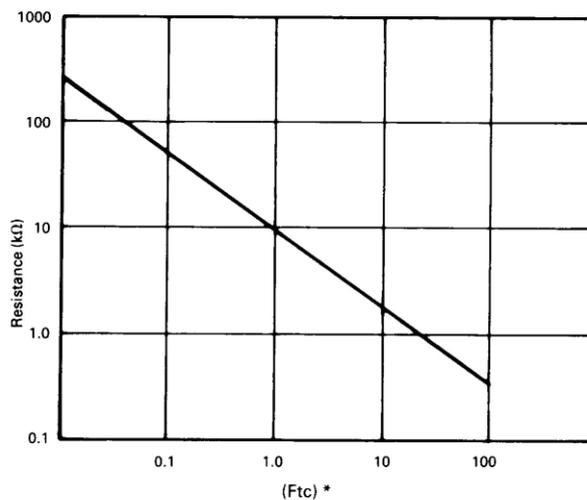
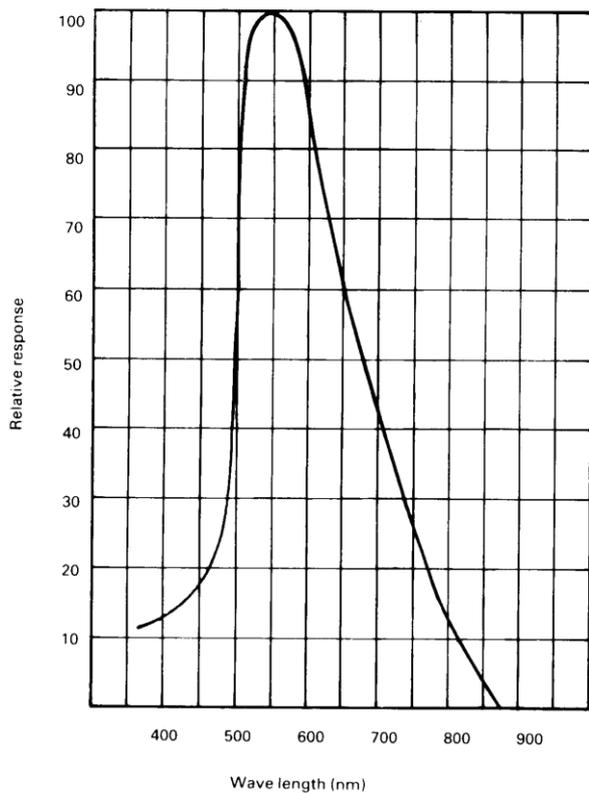


Figure 3 Resistance as a function of illumination



*1Ftc=10.764 lumens

Figure 2 Spectral response



Absolute maximum ratings

Voltage, ac or dc peak _____ 100V
 Current _____ 5mA
 Power dissipation at 25°C _____ 50mW*
 Operating temperature range _____ -25°C +75°C

*Derate linearly from 50mW at 25°C to 0W at 75°C.

Electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	10 lux	20	-	100	kΩ
	100 lux	-	5	-	kΩ
Dark resistance	10 lux after 10 sec	20	-	-	MΩ
Spectral response	-	-	550	-	nm
Rise time	10ftc	-	45	-	ms
Fall time	10ftc	-	55	-	ms

Figure 4 Resistance as a function illumination

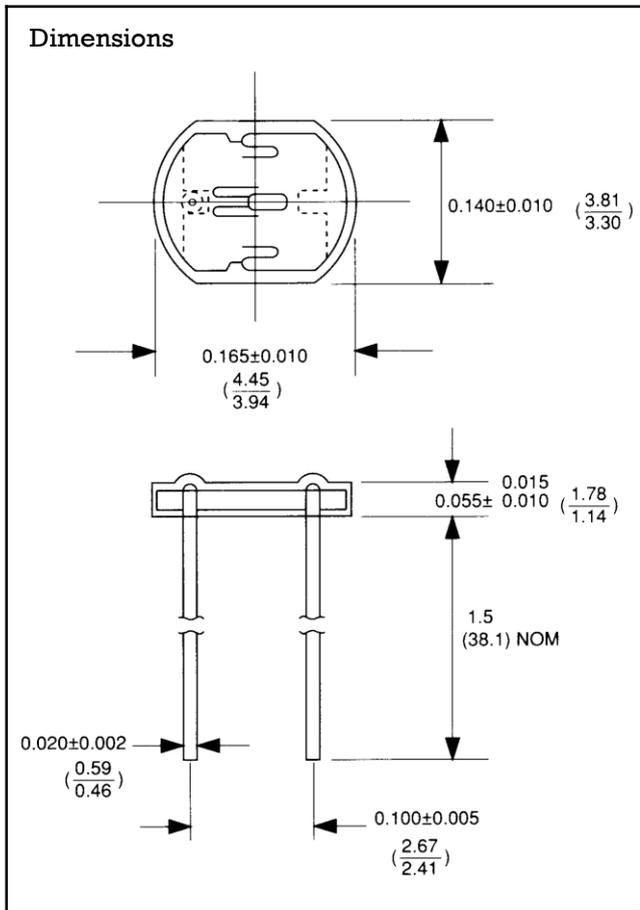
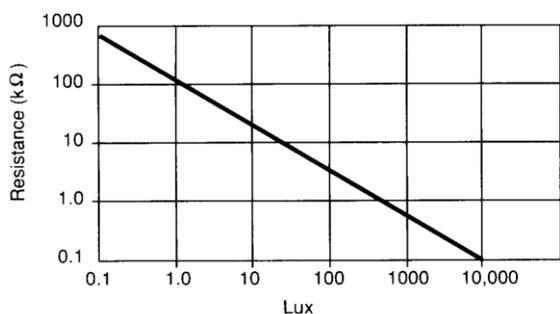
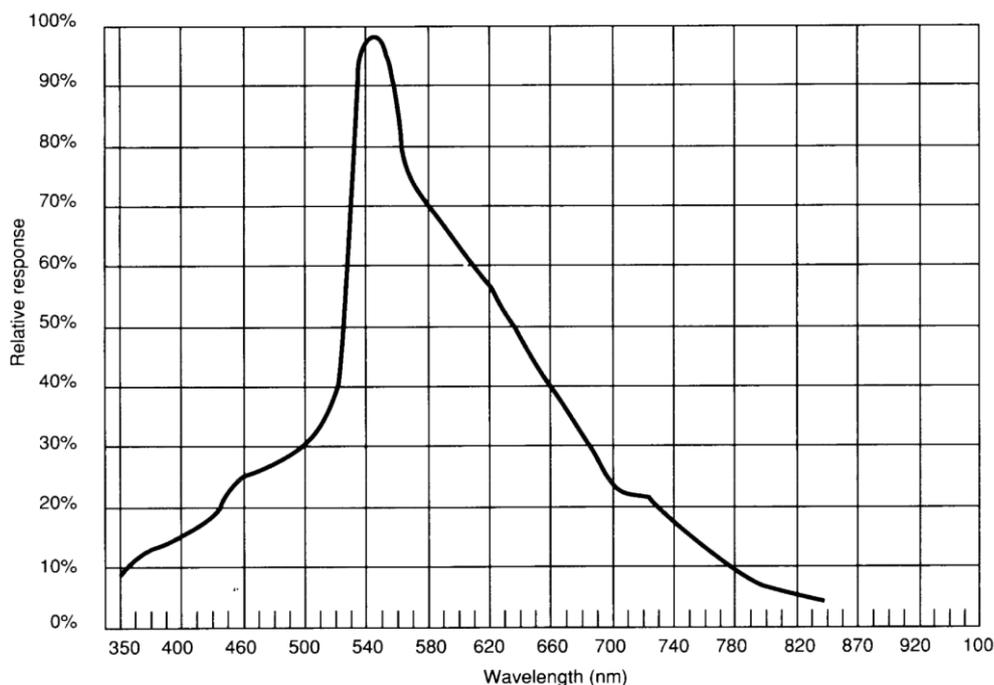
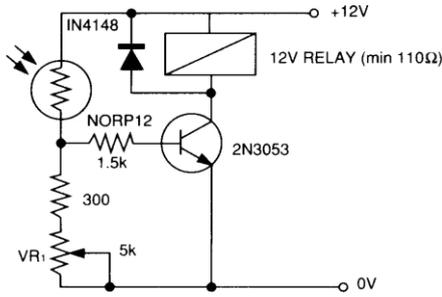


Figure 5 Spectral response



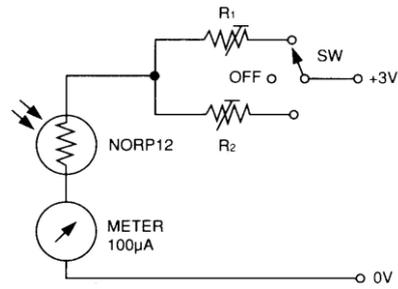
Typical application circuits

Figure 6 Sensitive light operated relay



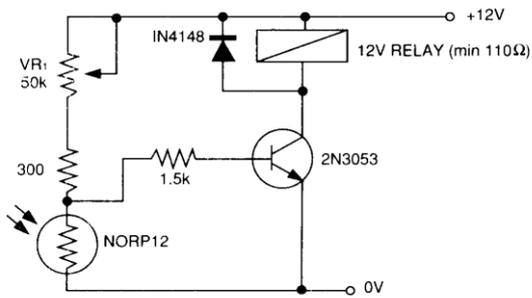
Relay energised when light level increases above the level set by VR₁

Figure 9 Logarithmic law photographic light meter



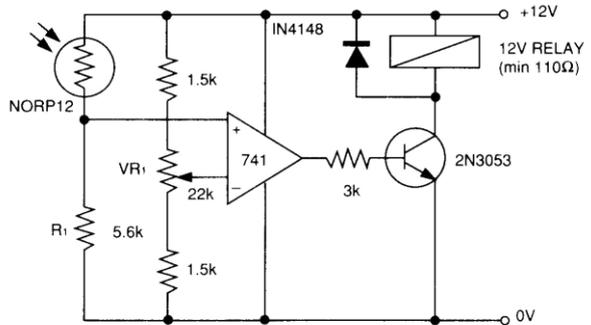
Typical value $R^1 = 100k\Omega$
 $R^2 = 200k\Omega$ preset to give two overlapping ranges.
 (Calibration should be made against an accurate meter.)

Figure 7 Light interruption detector



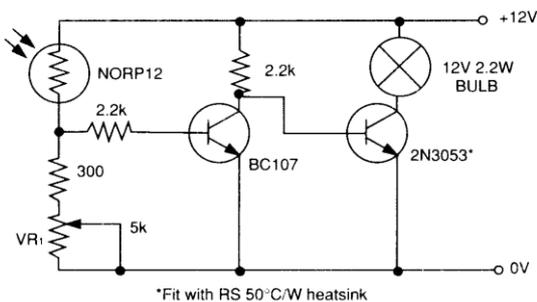
As Figure 6 relay energised when light level drops below the level set by VR₁

Figure 10 Extremely sensitive light operated relay



(Relay energised when light exceeds preset level.)
 Incorporates a balancing bridge and op-amp. R₁ and NORP12 may be interchanged for the reverse function.

Figure 8 Automatic light circuit



*Fit with RS 50°C/W heatsink

Adjust turn-on point with VR₁

The information provided in RS technical literature is believed to be accurate and reliable; however, RS Components assumes no responsibility for inaccuracies or omissions, or for the use of this information, and all use of such information shall be entirely at the user's own risk. No responsibility is assumed by RS Components for any infringements of patents or other rights of third parties which may result from its use. Specifications shown in RS Components technical literature are subject to change without notice.

Appendix G

Arduino Uno Code

```
const int redLed = 13;

const int greenLed = 12;

const int buzzer = 11;

const int fan = 10;

const int elctro_vanne = 9;

const int lamp = 8;

const int tv = 7;

const int air_con = 6;

const int lamp1 = 5;

const int lamp2 = 4;

void setup() {

  pinMode(redLed, OUTPUT);

  pinMode(greenLed,OUTPUT);

  pinMode(buzzer, OUTPUT);

  pinMode(fan, OUTPUT);

  pinMode(elctro_vanne, OUTPUT);

  pinMode(lamp, OUTPUT);

  pinMode(tv, OUTPUT);

  pinMode(air_con, OUTPUT);

  pinMode(lamp1, OUTPUT);

  pinMode(lamp2, OUTPUT);

  Serial.begin(9600);
```

Appendix G

```
}  
  
void smoke_gasleak(){  
  
    int smokeA0 = A5;  
  
    int sensorThres = 400;  
  
    digitalWrite(elctro_vanne,HIGH);  
  
    int analogSensor = analogRead(A5);  
  
    Serial.print("Pin A0: ");  
  
    Serial.println(analogSensor);  
  
    if (analogSensor >= sensorThres)  
    {  
        digitalWrite(redLed, HIGH);  
        digitalWrite(greenLed, LOW);  
        digitalWrite(fan,HIGH);  
        digitalWrite(elctro_vanne,LOW);  
        tone(buzzer , 200);  
    }  
    else  
    {  
        digitalWrite(redLed, LOW);  
        digitalWrite(greenLed, HIGH);  
        digitalWrite(fan,LOW);  
        digitalWrite(elctro_vanne,LOW);  
        noTone(buzzer);  
    }  
}
```

Appendix G

```
}  
  
delay(100);  
  
}  
  
void flame(){  
  
const int threshold=200;  
  
int flamepin= A0;  
  
digitalWrite(elctro_vanne,HIGH);  
  
int flamesensvalue=0;  
  
flamesensvalue=analogRead(A0);  
  
if (flamesensvalue<=threshold) {  
  
digitalWrite(greenLed,LOW);  
  
digitalWrite(redLed,HIGH);  
  
digitalWrite(fan,HIGH);  
  
digitalWrite(elctro_vanne,LOW);  
  
tone(buzzer,200);  
  
delay(1000);  
  
}  
  
else{  
  
digitalWrite(greenLed,HIGH);  
  
digitalWrite(redLed,LOW);  
  
digitalWrite(fan,LOW);  
  
digitalWrite(elctro_vanne,LOW);
```

Appendix G

```
noTone(buzzer);
}

}

void light_ldr(){

const int ldrPin = A3;

int ldrStatus = analogRead(A3);

if (ldrStatus <=300) {

digitalWrite(lamp, HIGH);

Serial.println("LDR is DARK, lamp is ON");

}

else {

digitalWrite(lamp, LOW);

Serial.println(" lamp is off");

}

}

void co_tex() {

const int threshold=300;

int co_pin= A4;

digitalWrite(elctro_vanne,HIGH);
```

Appendix G

```
int co_sensvalue=0;

co_sensvalue=analogRead(A4);

if (co_sensvalue<=threshold) {

digitalWrite(greenLed,LOW);

digitalWrite(redLed,HIGH);

digitalWrite(fan,HIGH);

digitalWrite(elctro_vanne,LOW);

tone(buzzer,200);

delay(1000);

}

else{

digitalWrite(greenLed,HIGH);

digitalWrite(redLed,LOW);

digitalWrite(fan,LOW);

digitalWrite(elctro_vanne,LOW);

noTone(buzzer);

}

}

void loop() {

smoke_gasleak();

flame();

light_ldr();
```

Appendix G

```
co_tex();

if(Serial.available()){

char val= Serial.read();

switch(val){

case'0':

digitalWrite(7,LOW);

Serial.print("a");

break;

case'1':

digitalWrite(7,HIGH);

Serial.print("b");

break;

case'2':

digitalWrite(6,LOW);

Serial.print("c");

break;

case'3':

digitalWrite(6,HIGH);

Serial.print("d");

break;

case'4':

digitalWrite(5,LOW);

Serial.print("e");
```

Appendix G

```
break;

case'5':

digitalWrite(5,HIGH);

Serial.print("f");

break;

case'6':

digitalWrite(4,LOW);

Serial.print("g");

break;

case'7':

DigitalWrite (4, HIGH);

Serial.print("h");

break;

}

}

}
```

Appendix H

The code will be loaded to the Node MCU

```
#include<ESP8266WiFi.h>
#include<PubSubClient.h>
const char *ssid="Allure A8y";
const char *password="d6b4e65c1767";
const char*mqtt_server="192.168.43.233";
char* Topic = "inTopic";
WiFiClient ESPClient;

void setup() {

  Serial.begin(9600);
  delay (100);
  WiFi.begin(ssid,password);
  reconnect();
  delay(2000);

}

PubSubClient client (mqtt_server,1883,callback,ESPClient);

void loop() {
  if (!client.connected()&&WiFi.status()==3){
    reconnect();
  }
}
```

Appendix H

```
client.loop();  
delay(20);  
}  
void callback(char*Topic,byte*payload,unsigned int lenght){  
  
String TopicStr = Topic;  
if (payload [0]== ('0')){  
    Serial.print("0");  
    respond();  
  
}  
if (payload [0]== ('1')){  
    Serial.print("1");  
    respond();  
}  
if (payload [0]== ('2')){  
    Serial.print("2");  
    respond();  
}  
if (payload [0]== ('3')){  
    Serial.print("3");  
    respond();  
}  
if (payload [0]== ('4')){  
    Serial.print("4");  
    respond();  
}
```

Appendix H

```
if (payload [0]==( '5')){
  Serial.print("5");
  respond();
}
if (payload [0]==( '6')){
  Serial.print("6");
  respond();
}
if (payload [0]==( '7')){
  Serial.print("7");
  respond();
}
}
void respond(){
  delay(5);
  if(Serial.available()){
    char val= Serial.read();
    switch(val){
      case'a':
        client.publish("outTopic","tv is OFF");
        break;
      case'b':
        client.publish("outTopic","tv is ON");
        break;
      case'c':
        client.publish("outTopic","air_con is OFF");
        break;
    }
  }
}
```

Appendix H

```
case'd':
client.publish("outTopic","air_con is ON");
break;
case'e':
client.publish("outTopic","lamp1 is OFF");
break;
case'f':
client.publish("outTopic","lamp1 is ON");
break;
case'g':
client.publish("outTopic","lamp2 is OFF");
break;
case'h':
client.publish("outTopic","lamp2 is ON");
break;

}
}
}
void reconnect() {

if (WiFi.status() != WL_CONNECTED){
while(WiFi.status() != WL_CONNECTED){
delay(500);
}
}

if (WiFi.status() == WL_CONNECTED ){
```

Appendix H

```
while(!client.connected() ){
    String clientName ;
    clientName += "ESPClient";
    uint8_t mac[6];
    WiFi.macAddress(mac);
    clientName += macToStr(mac);
    if (client.connect((char*) clientName.c_str())){
        Serial.print("t mqtt connected");
        client.subscribe(Topic);

    }
    else{Serial.println("tFailed.");
        abort();}
    }
}
```

```
String macToStr(const uint8_t* mac){
```

```
    String result;
    for (int i=0; i<6; ++i){
        result += String (mac[i], 16);
        if (i<5){

            result += ':'  ;

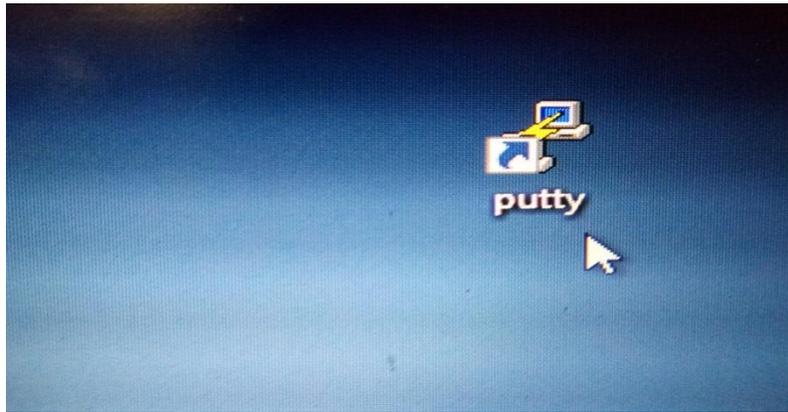
        }
    }
```

Appendix H

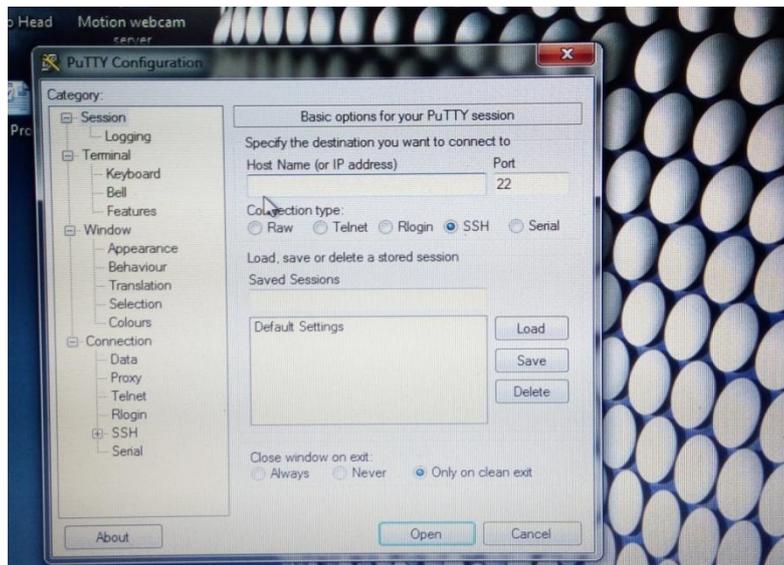
```
    }  
    return result;  
}
```

APPENDIX I

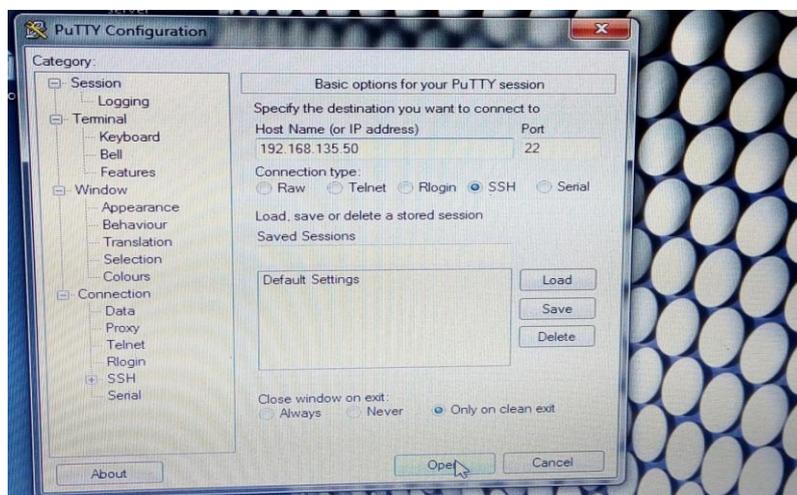
- How to use putty
 - ✓ Select the icon putty on your laptop



- ✓ This window show up

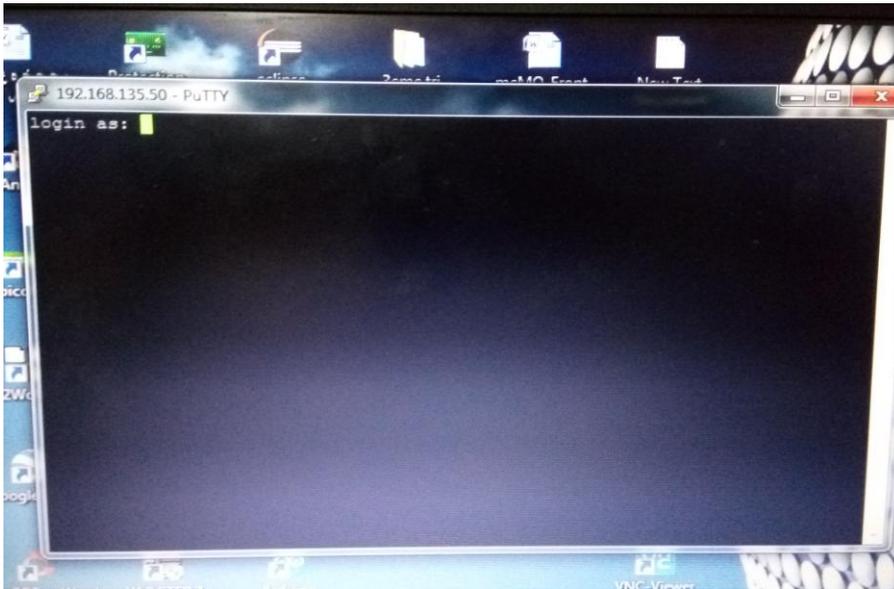


Write down the IP address of your Raspberry Pi then select SSH and click Open

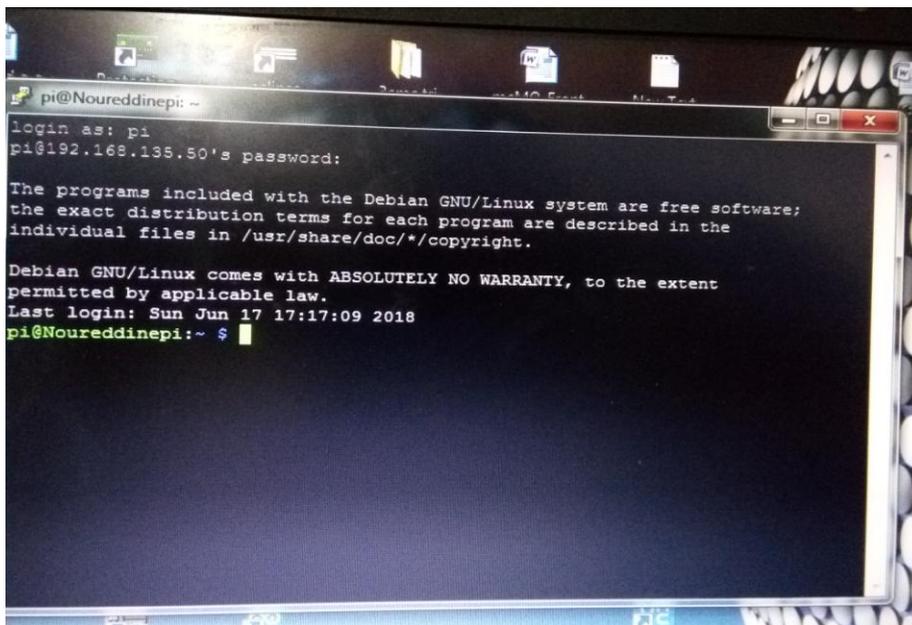


APPENDIX I

This window will appear just write down user name and password of your Raspberry Pi



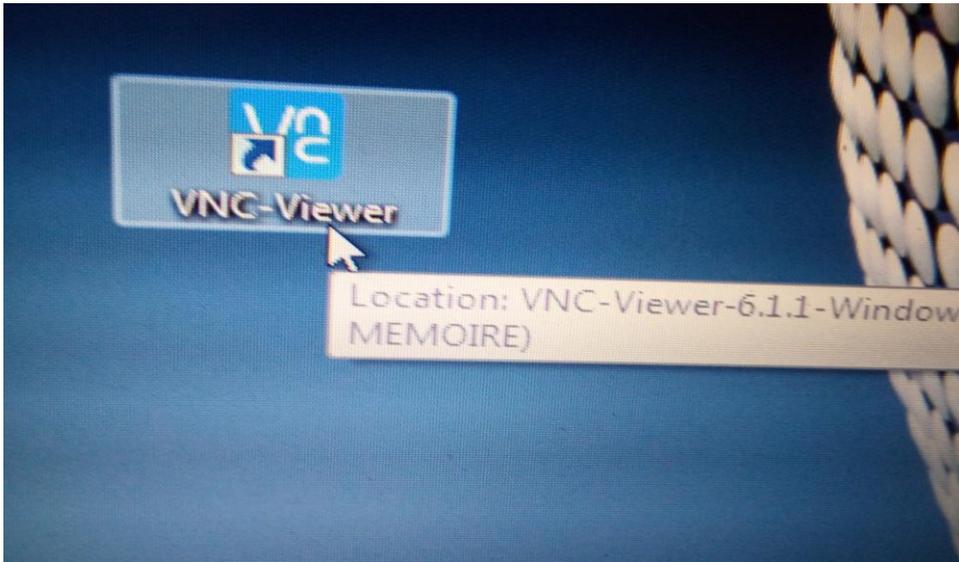
After you wrote your user name and password your Pi gives you the access



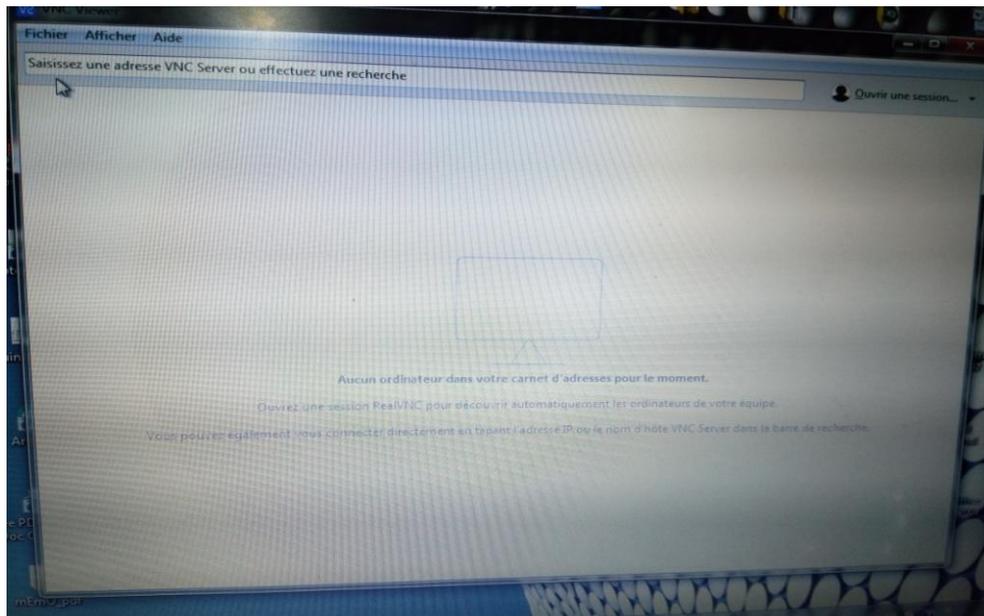
Now you can command your Raspberry Pi as you wish

APPENDIX I

- How to use VNC viewer
 - ✓ Select the icon VNC viewer on your laptop

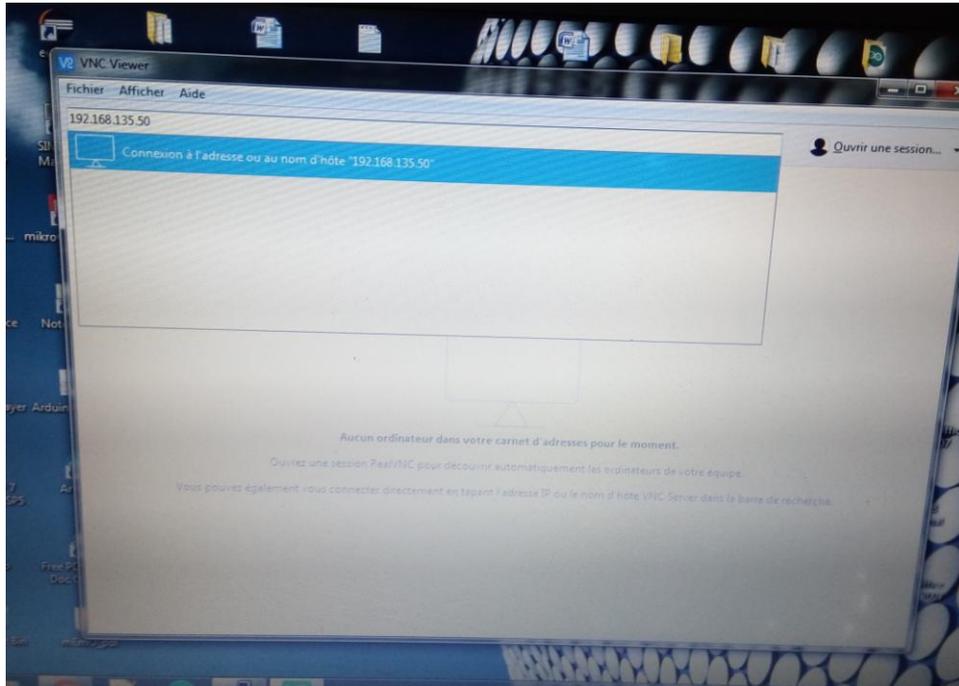


- ✓ This window show up

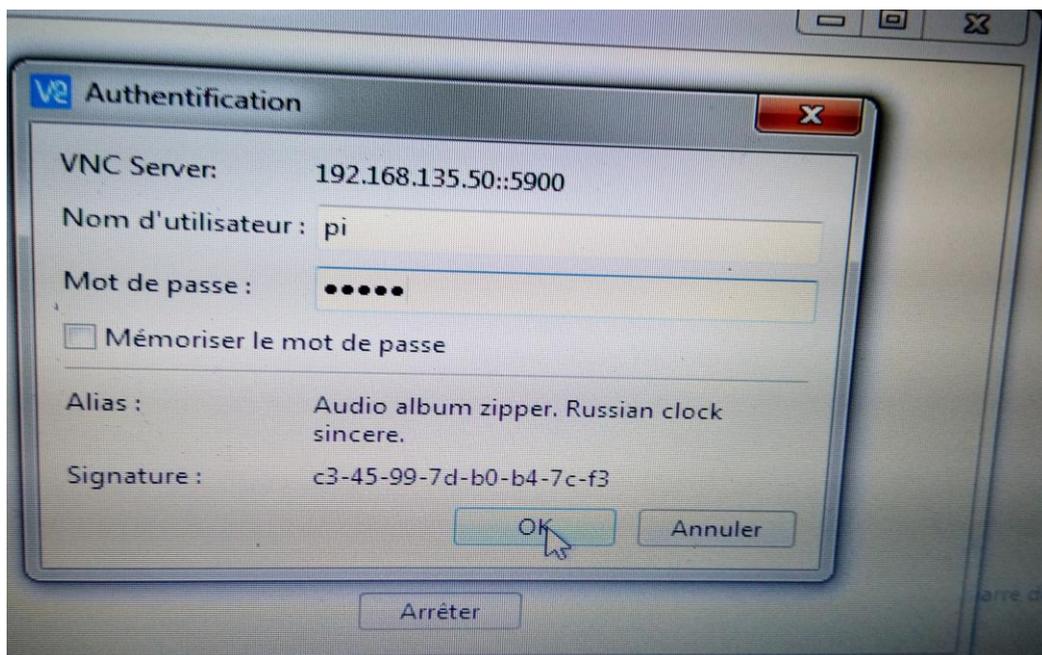


Write down the IP address of your Raspberry Pi then press enter

APPENDIX I



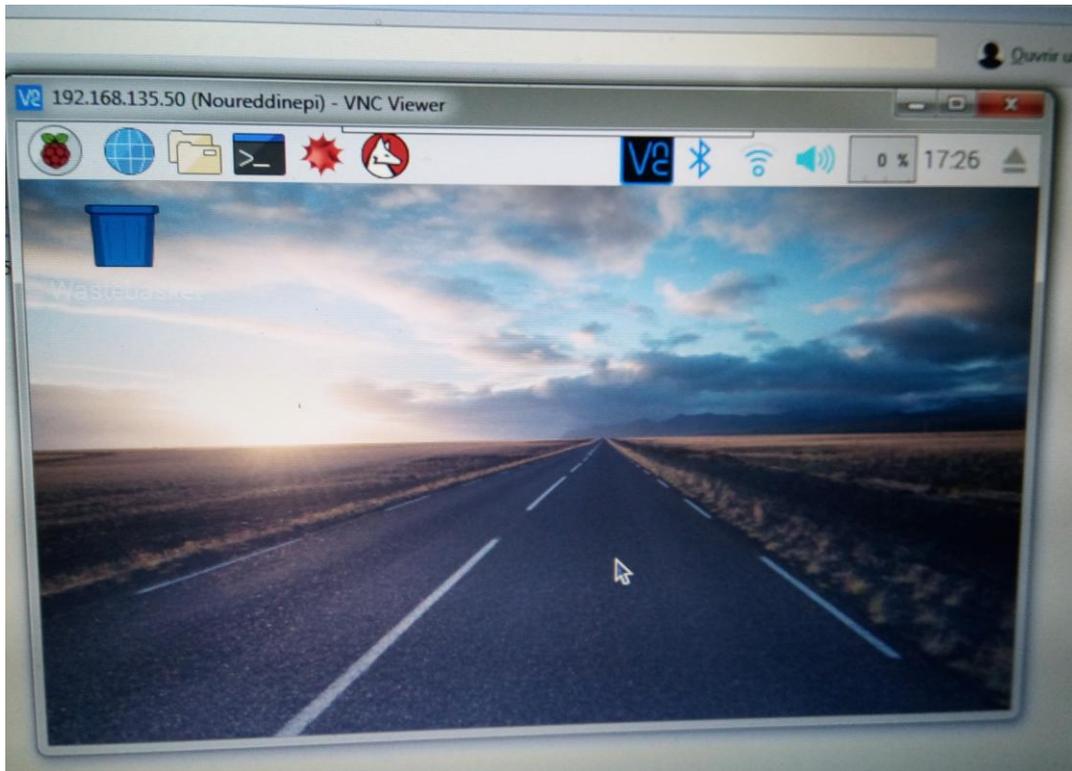
Other window will appear



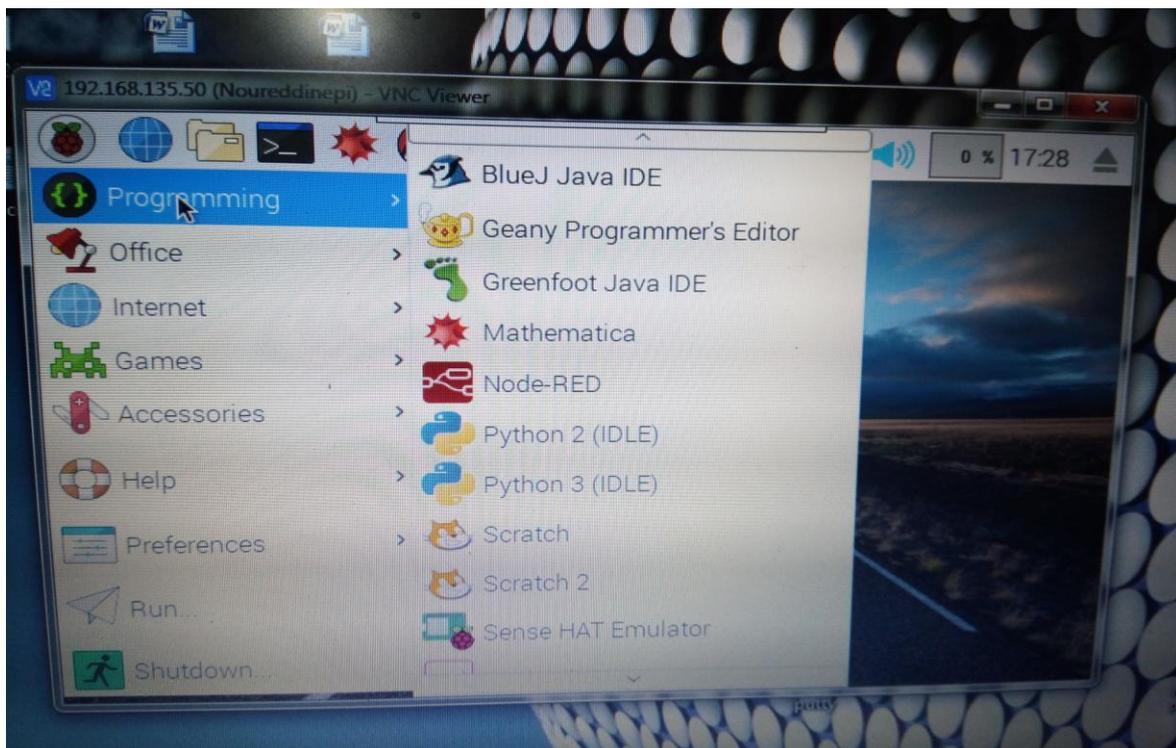
Write down the user name and password of your Raspberry PI and press OK

The Desktop of your raspberry PI will appear

APPENDIX I

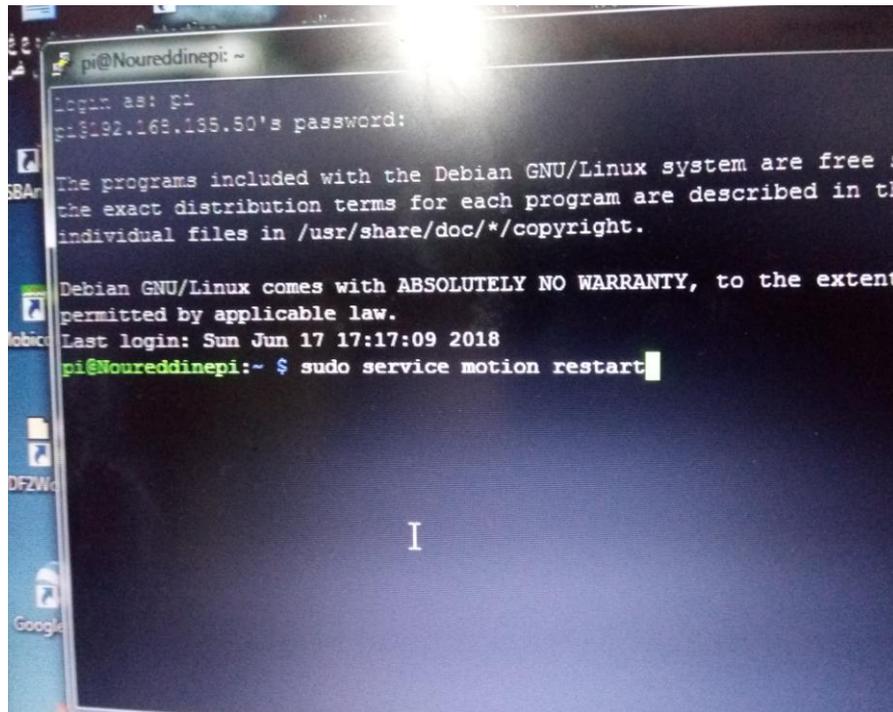


That's the menu of the Raspberry Pi



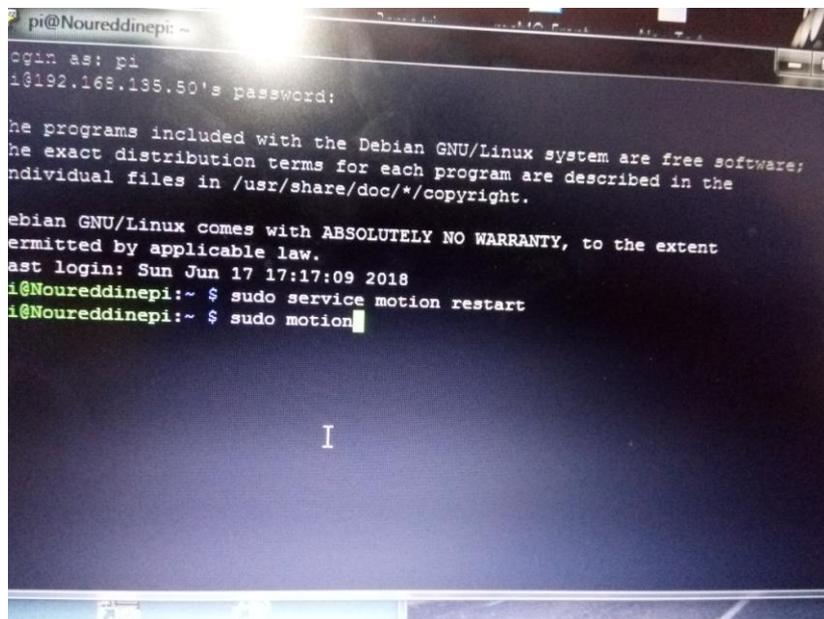
APPENDIX I

- How to use Motion
 - ✓ On the prompt command window write down the command
Sudo service motion restart



```
pi@Noureddinepi: ~  
login as: pi  
pi@192.168.135.50's password:  
The programs included with the Debian GNU/Linux system are free s  
the exact distribution terms for each program are described in th  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Jun 17 17:17:09 2018  
pi@Noureddinepi:~ $ sudo service motion restart
```

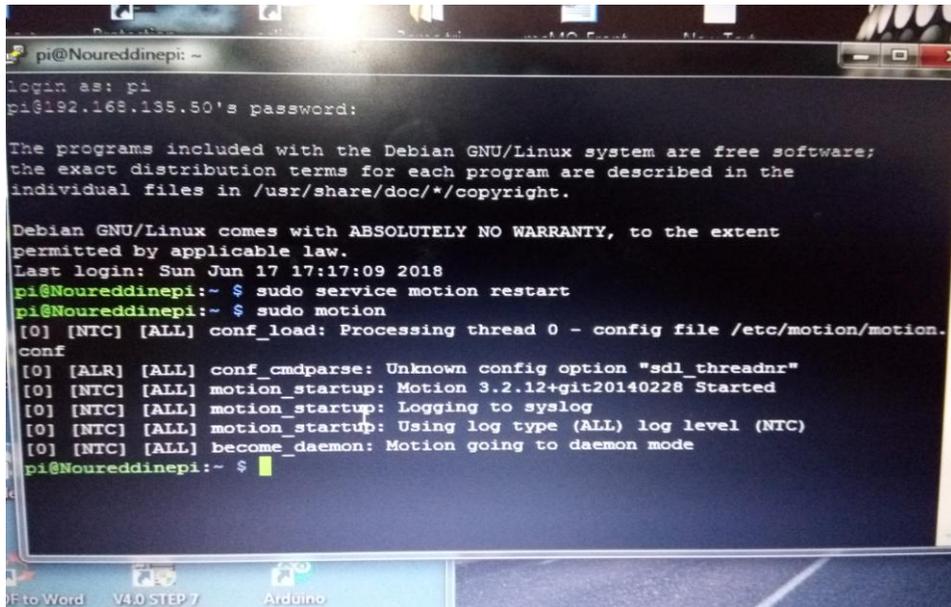
then
Sudo motion



```
pi@Noureddinepi: ~  
login as: pi  
pi@192.168.135.50's password:  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Jun 17 17:17:09 2018  
pi@Noureddinepi:~ $ sudo service motion restart  
pi@Noureddinepi:~ $ sudo motion
```

Confirmation that is motion on

APPENDIX I

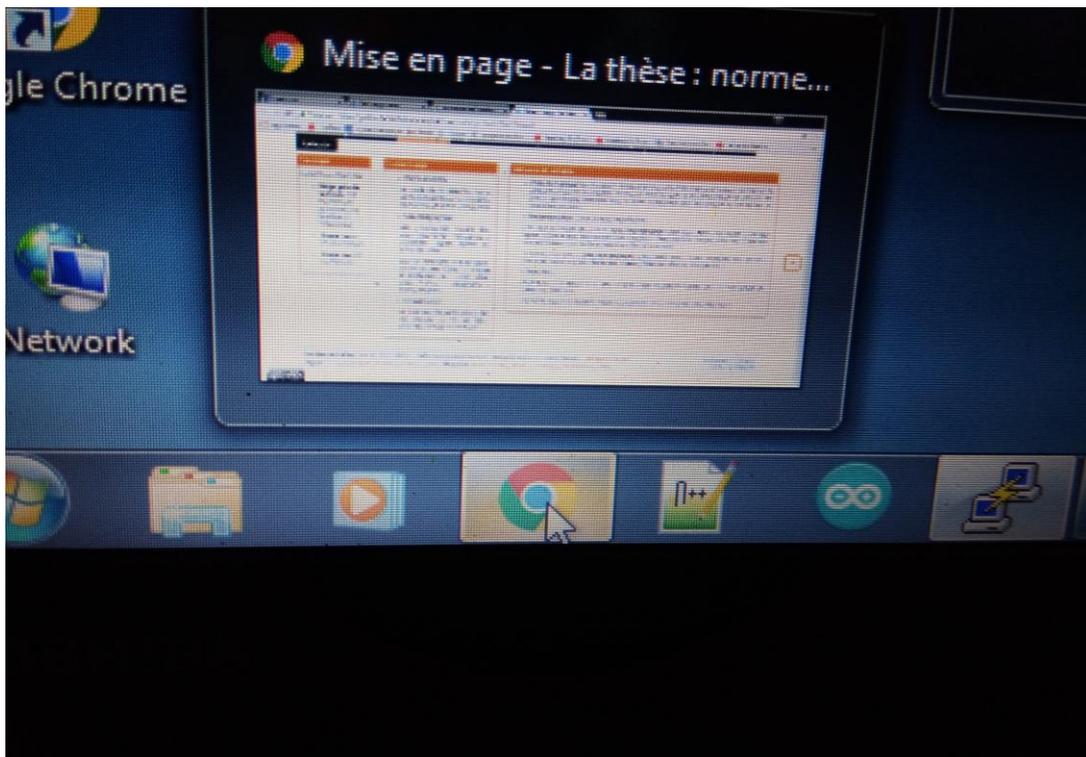


```
pi@Noureddinepi: ~
Login as: pi
pi@192.168.135.50's password:

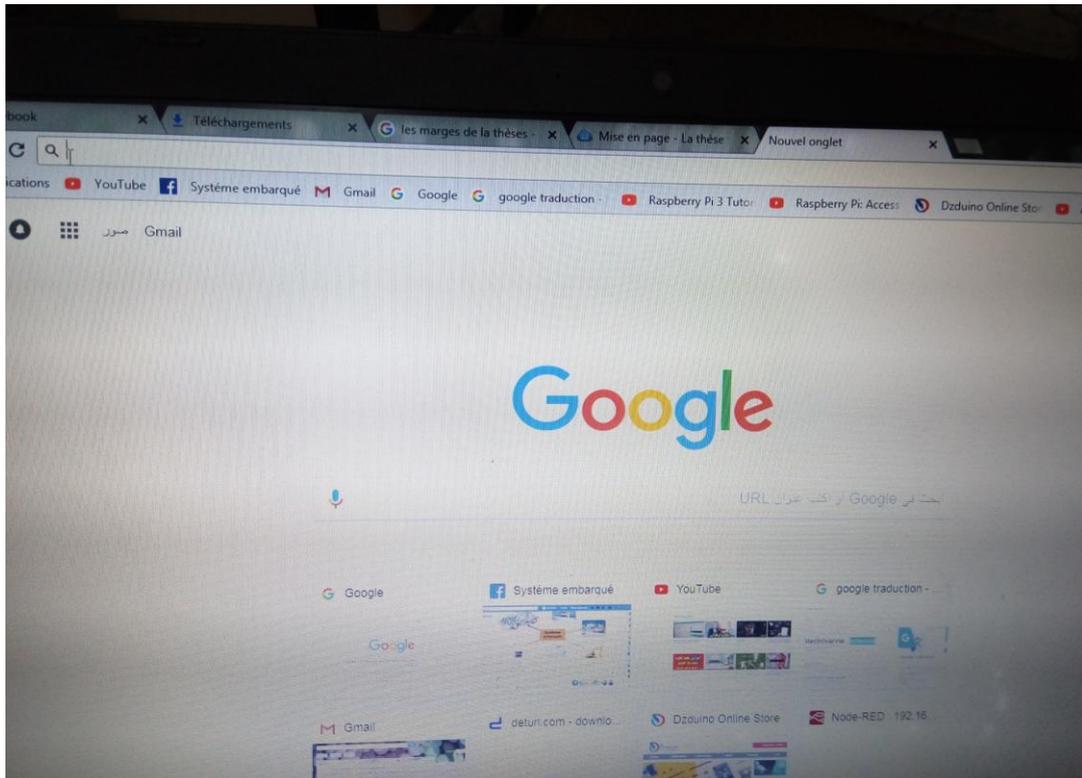
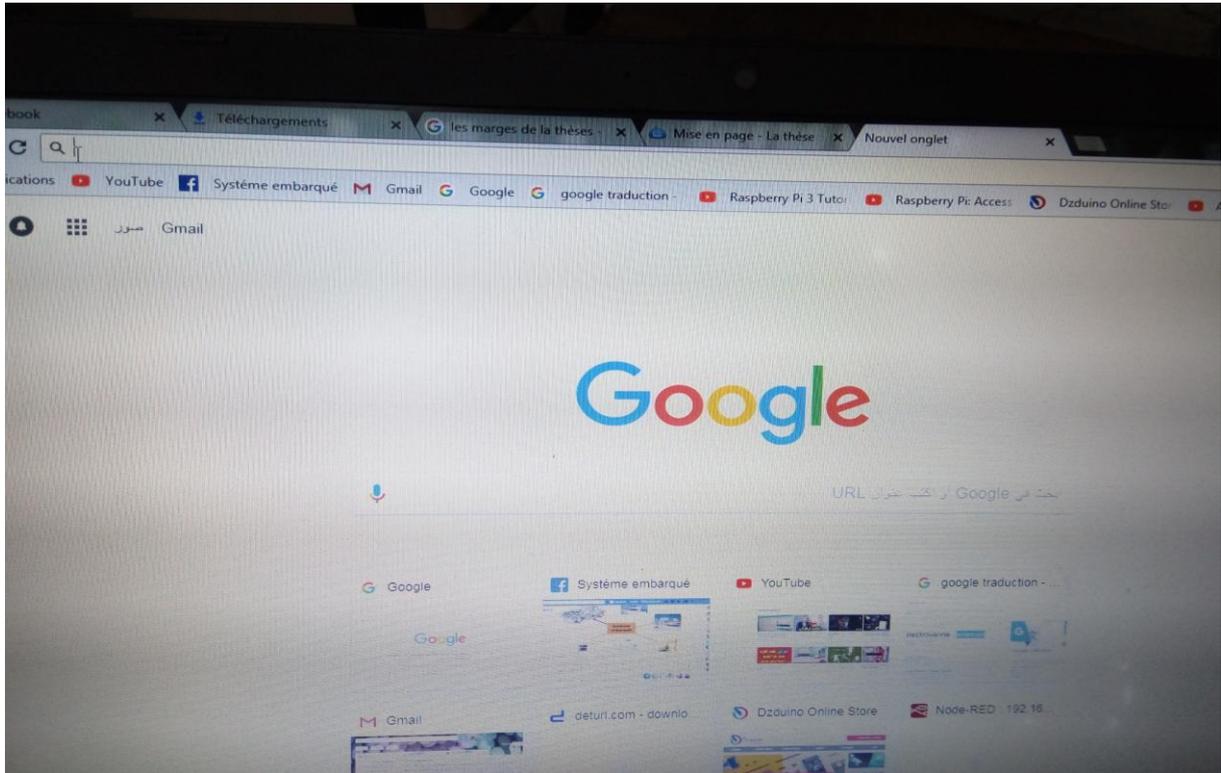
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun 17 17:17:09 2018
pi@Noureddinepi:~$ sudo service motion restart
pi@Noureddinepi:~$ sudo motion
[0] [NTC] [ALL] conf_load: Processing thread 0 - config file /etc/motion/motion.
conf
[0] [ALR] [ALL] conf_cmdparse: Unknown config option "sdl_threadnr"
[0] [NTC] [ALL] motion_startup: Motion 3.2.12+git20140228 Started
[0] [NTC] [ALL] motion_startup: Logging to syslog
[0] [NTC] [ALL] motion_startup: Using log type (ALL) log level (NTC)
[0] [NTC] [ALL] become_daemon: Motion going to daemon mode
pi@Noureddinepi:~$
```

Go to web browser and select

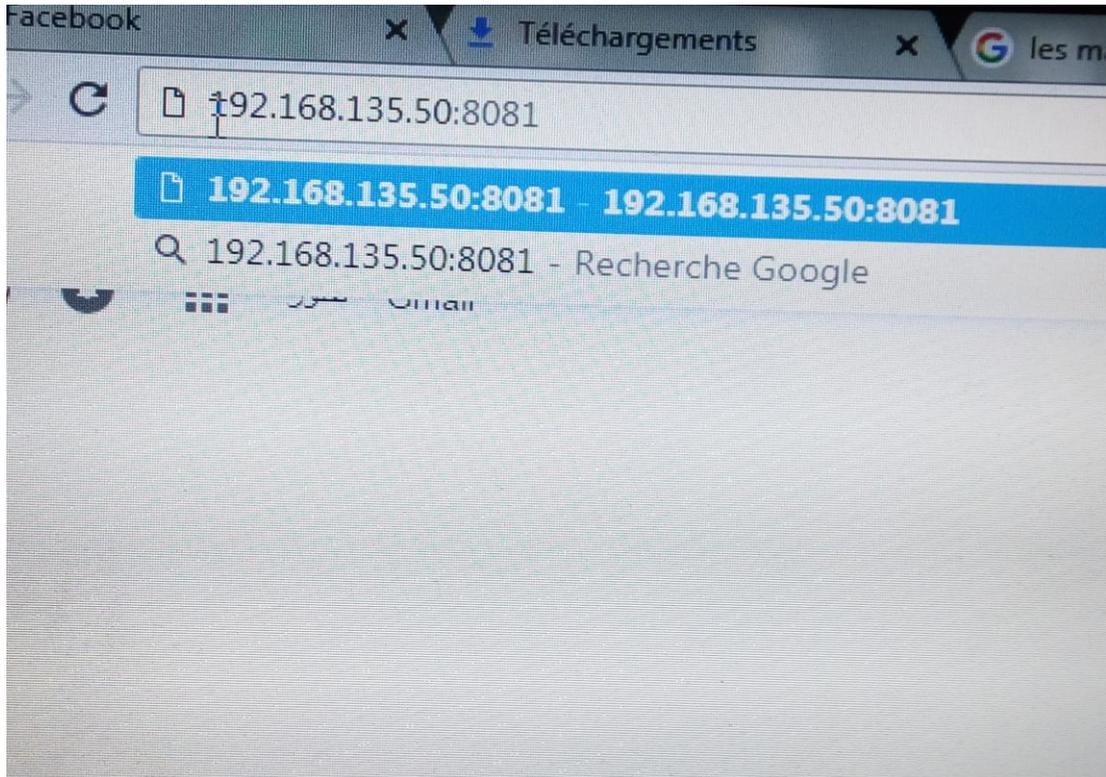


APPENDIX I



Write down the IP address of your Raspberry Pi:8081 press enter

APPENDIX I



The web cam start streaming

