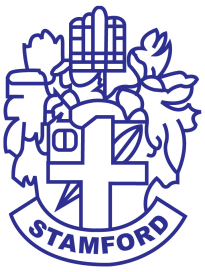
Applications of Internet of Things (IoT) Towards Smart Home Automation

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor in Computer Science & Engineering

by

Kamrul Hasan
CSE 04806354
&
Tanvir Ahmed
CSE 04806364

Supervised by: Dr. Kamruddin Nur
Associate Professor



Department of Computer Science and Engineering
STAMFORD UNIVERSITY BANGLADESH

March 2017

Abstract

There are endless opportunities of smart home IoT devices. With the emergence of Smart phones and Internet connectivity everywhere, smart homes with IoT devices can be controlled or operated from anywhere, anytime. In this project we present, how electrical appliances and IoT devices can be operated from a Smartphone with just a click. There are two major modules in this project, these are Micro-controller and Mobile Device. The main idea of this system is to automate certain electrical devices of a home and to provide home security facility, using micro-controller modules and controlling it by mobile devices. The current system is designed to control fan, light, TV, AC, RGB lamps, Live streaming of a home which are connected to micro-controller modules, and we control the devices from our phone using a user-friendly mobile application. This system can be accessed over the Internet using local Wi-Fi or/and mobile Internet. This project also features which micro-controller modules we used, which programming language we used to encode our micro-controllers and how our app works with the micro-controllers to produce the designated outputs that is required for the automation. This project has been successfully completed and the system is successfully developed.

Approval

The Project Report "Applications of Internet of Things (IoT) Towards Smart Home Automation" submitted by Kamrul Haasan ID: CSE 04806354, Tanvir Ahmed ID: CSE 04806364, to the Department of Computer Science & Engineering, Stamford University Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (Hons) in Computer Science & Engineering and approved as to its style and contents.

Board of Examiner's Name, Signature, and Date:

.....	.....	.....
(Board Member 1)	**(Board Member 2)**	**(Board Member 3)**
Date:	Date:	Date:

Supervisor's Signature and Date:

.....
Dr. Kamruddin Nur

Date:

Declaration

We, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of Dr. Kamruddin Nur, Associate Professor & Chairman, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signature and Date:

.....
Kamrul Hasan
Date:

.....
Tanvir Ahmed
Date:

Acknowledgments

First of all, we would like to thank the Almighty Allah (swt.) for all the blessings and guidance he bestowed upon us throughout this whole project and we hope this guidance will be with us all the way in our life.

We also wish to express our sense of gratitude to our supervisor Dr. Kamruddin Nur, Chairman of the Department of Computer Science & Engineering, Stamford University Bangladesh, for his unconditional guidance, insights, immense patience, unlimited encouragement and inspirations throughout this whole project. It was his pep talks and much more that kept us going and thriving for this success. It was a great honor for us working under him with this project. We acknowledge his contributions to enhance our knowledge on the subject.

We specially owe thanks to all the teachers of the faculty of Computer Science & Engineering, Stamford University Bangladesh for their help, valuable suggestions and discussions. Moreover, we would like to thank all our friends who have supported and helped us throughout this project.

Last of all we would also like to thank our parents who have walked along with us in every step of our life and always encouraged and inspired us to do greater things in our life.

Table of Contents

List of Figures

List of Tables

1 Introduction

Today, Internet of Things (IoT) are embracing almost every aspect of our lives. Now devices are connected to each other making a network completing our day to day tasks. With the cheaper availability of smart IoT devices, the applications of IoT is simply expanding its applications from home to business automation. In this chapter, we investigate the fundamental concepts, history, and progress of IoT and continue towards a solid background study.

1.1 Objective

The purpose of our project is to build an automation system for homes. This system will be brained by the ESP8266 12E wi-fi micro-controller which will also store the necessary coding so it can work simultaneously with the BLYNK mobile app and send out commands throughout the modules using the help of a relay board. BLYNK app is the UI and command center for this module. All the devices that will be connected in this system can be accessed and controlled through the BLYNK app. The platform also had to be modular and flexible enough to support different sensors and actuators that could be incorporated and relocated very easily. Though we first used Arduino Uno, we found the ESP8266 12E or NodeMCU to be more efficient and work-friendly. Besides the NodeMCU we also used many other devices and sensors for this project to make it as user-friendly as possible.

1.2 Background Study

Home automation technology and Smart home appeared very much in science fiction of the 1920s. But no one knows the exact date of the invention of home automation. Based on smart technology improving process, the home automation system does not come by immediate invention. It comes step by step with only insignificant improvement [1]. The previous step is almost same with the next step. The first-time people noticed the high

technology in dwelling, they did some connection with home automation, it was 1960s. It was called “wired homes” at that time. It was built by some hobbyist. After that, the first official name of home automation appeared in 1984 by the American Association of House Builders. This development is the key to the modern smart homes [2]. People at that time understood that a smart home is not owing to how well it is built, not how effectively it uses space, not due to how it is environmentally friendly. It is only because of how interactive technologies that it contains. Those are still useful rules for home automation technology today [3].

1.2.1 Present Situation of IoT

Nowadays home automation is a significant symbol of the human society civilization. The first automation facility which entered into peoples home was the automatic washing machine, automation air-condition and so on. When people think about home automation, most of them may imagine living in a smart home [4]: One remote controller for every household appliance, cooking the rise automatically, starting air conditioner automatically, heating water for bath automatically and shading the window automatically when night coming. To some extent home automation equals to smart home. They both bring out smart living condition and make our life more convenient and fast. Home automation satisfies the residents needs and desires by adjustable light, temperature, ambient music, automatic shading, safety & security, even arrangement of wire. Home automation technologies are the latest fascination with housing mechanism. However, with the appearance of new electronic technologies and their combination with older, traditional building technologies, the smart home is at last becoming a reality [5]. The basic idea of home automation is to monitor a dwelling place by using sensors and control systems. Through adjustable various mechanisms, user can enjoy customized heat, ventilation, lighting, and other servers in living condition. The more closely adjust the entire living mechanical system and loop control system, the intelligent home can provide a safer, more comfortable, and more energy economical living condition. For example, a home automation unit can notice that when the dweller is sleeping, the light is switched down or off, and the temperature control unit is switched down [1]. According to the clock that dweller made, the home automation control unit can lift the curtain automatically and play morning music. It can also monitor the security and fire alarm. In some countries, there are no district heating, home automation control unit can preheat water for bathing or washing. Nowadays people are very familiar with Smart Phone, which gives user intelligent phone service. Smart home is similar like smart phone. It gives dweller smart living service [4]. If you havent experienced a smart home, youll never know how amazing feeling dwelling

ever could be. Even though you do not have this kind of experience, you do not need to be regretful. With pictures, this thesis will lead you to a smart home world.

Figure 1.1: Normal private smart home design

The following picture, Figure 1.1, be said illustrates private smart home design and its location. However, it has to be said its a very normal home automation. In some developed countries, people have this kind of level home automation, they even didnt notice about home automation. However, for people living in developing countrys view point, wireless dimmer The following picture, Figure 1.2, illustrates the modern smart home in structural level. In this structure, PC and smart cell phone are Internet remote controls. They go through the Internet to dwellers intra-net and the control device such as visitor panel, electric lock, security module, monitoring. In the home, dweller can use remote controller or digital smart device like pad on the wall to control smart windows, electric curtain, lighting, air-condition, and background music.

1.2.2 Future home

In the future, home automation will be as popular as mobile phone. Some specialist predicts that to happen the 15 years. Every family will use smart home technology such as the central control system of household appliances, Wi-Fi controlled light and remote monitoring. Some optimistic expert even predicts less than 15 years smart home will be as popular as smart phone. There will be a system that can control all the household appliances as shown in figure 1.3. Now, Microsoft Company is testing a system that can control all the household appliances. The idea of this operating system is that dweller can control all the household appliances through by PC or smart phone.

With the improvement of high technology of smart home, in the future, screen refrigerator will let you live more conveniently. Now, Samsung, LG Company and so on are

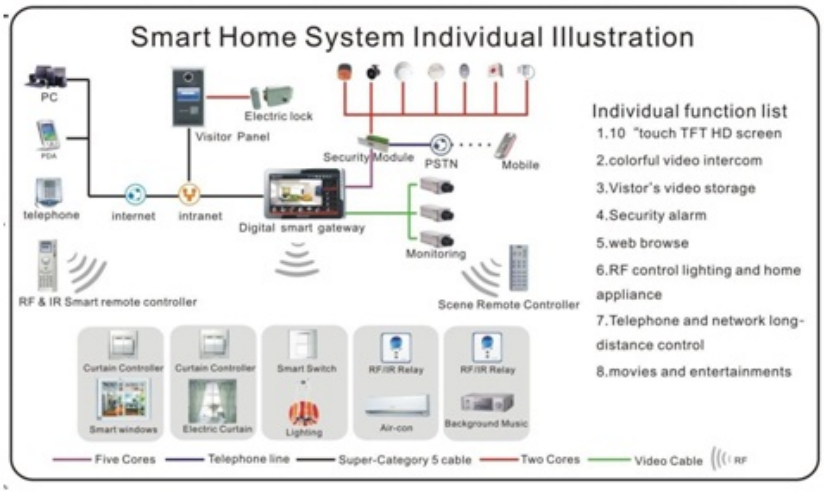


Figure 1.2: Smart home system individual illustration

going to develop screen refrigerator. In the previous Consumer Electronics Show (CES) Samsung Company demonstrated a smart screen refrigerator which uses android system [6]. People can use this refrigerator to manage food, and buy food online. Depending on the food in the refrigerator, it can also provide a recipe for the user.Touch screen will be everywhere. In the future, table cabinet, mirror, refrigerator, oven will also have a touch screen. They all will have a smart system, where people can download applications for their household appliances.

1.2.3 Entire elements of Home automation system

Future is always full of surprises. However, we must face the reality. Home automation nowadays is not as cheap as every home we own. Some component like Wi-Fi dimmer is cheap, but some component like house central control system is expensive. The following figure 1.4 is the host machine of the central control system. In the Chinese market, it costs around 5000 Euro

There is no doubt that the host machine of central control system is the most important element of Home Automation. After this, I will introduce other elements in the following list. Before you transform your house into home automation, you must take these things into consideration. Lets list few common things that we quite often use at our home or offices.

Figure 1.3: Central control system operated on the smart phone

Figure 1.4: Host machine of central control system

Cable and structured wiring products:

- Connectors
- Router and modem
- Batteries
- Rack mount
- Structured panels

- Wire and cable management tools

Cameras and surveillance:

- Cameras
- Surveillance screens
- Video monitoring kits
- Video recorders

Dimmers, lighting and appliance control:

- Dimmers, wall switches, keypads and touch-screens
- Hand-held and table top remotes
- Light bulbs, light fixtures
- Plug-in / screw-in modules or wire-in / outlet modules

Door locks, access control and Security:

- Door and gate access
- Garage door control
- Motorized gate & door control
- Vaults & mailboxes
- Intercoms
- Remote monitoring & control

Solar and energy management

- Automatic lighting

- Power monitoring and appliance efficiency
- Remote-control windows and coverings
- Solar panels
- Water and sprinkler management
- Heat pump

Speakers, a/v and home theater

- Home theater accessories
- Display screens or projector
- Speakers
- Volume controls and speaker switching
- Multi background music

Thermostats and HVAC controls

- Air filters
- Humidity sensing and control
- Temperature sensors and controllers
- Ventilation
- Weather stations and sensors

1.3 Background on Security System

This section introduces security aspects and some protocol solutions. General security concepts are briefly presented in the following Subsections. Arnaud Germis [7] stated following parts describe various security protocols acting at different network layers.

1.3.1 Security Concepts

Information security regroups many different concepts. There exist many different models to describe security requirements. The most usual requirements to ensure a transactions security (known as the CIA triad) are:

- Confidentiality: the transaction content is only disclosed to authorized parties.
- Integrity: the transaction is only emitted and modified by authorized parties.
- Availability: the system is available at all times.

To ensure the two first requirements different techniques were developed. The availability requirement mandates more abstract constraints such as the data and service redundancy Cryptography In cryptography, the encryption process transforms the information into seemingly meaningless data by using an algorithm and a key. The encrypted data can only be decrypted with the key or by the extremely long process of testing all the possible keys. There are two kinds of encryption algorithms: symmetric-key encryption and public-key encryption. Symmetric-Key Encryption In the case of symmetric-key encryption, the same key is used to encrypt and decrypt data. It is possible to use this method to ensure that the data are from a party knowing the secret key. Additionally, the identity of a party can be assessed by parties sharing a common secret key. example: In computer systems, a specific example is when a user identifies it-self (authenticate) with one of the many challenge/response protocols. The client sends an authentication request to the server. The server replies with a challenge to the client. The challenge is made in such a way that, only a client knowing the shared-key can respond. A secret password is a specific kind of key shared between the user that knows it and the system that stores it. Public-Key Encryption In the case of public-key encryption, a pair of keys is used. One key is public and is used to encrypt data. The other key is private and is used to decrypt the public-key encrypted data. This method can also be used to sign data. Signing data ensure to anybody having the public key that the data are from the party possessing the private key. Additionally, the identity of a party can be assessed by proving that it can decrypt with its private key data encrypted with its public key

1.3.2 Authentication

There are different ways to authenticate a party. A common secret key can be exchanged, as in the case of password or pass phrase. Nowadays, it is the most common way to

authenticate a user. The main drawback is that the key has to have a high entropy and being long enough. Otherwise, the key can be potentially guessed by trying all the possible keys. The password authentication is supported by HTTP, but the sent password is not encrypted. Therefore, this system provides very little security. Another way is to use public-key signatures. A model of trust has to be established with this technique. The two most common are the web of trust and the certificate authority models. In the certificate authority model, the identity of the parties is assured by a trusted third party that signs the public keys. Therefore, all parties can trust each other because the public keys, that they use to sign transactions, are them-self signed by a common trusted third party. In such a system, trusted third-party public keys are stored in the device. One advantage is that devices do not need prior knowledge of the other devices to authenticate them. Another advantage is that the trust into a party can be revoked by the trusted third party.

IPSEC

IPSEC is a protocol (on top of IP) to ensure the confidentiality and integrity of IP packets IPv4 has an optional support of it. In contrast, IPv6 requires the possibility to use IPsec. The packet content is encrypted and verified by using a symmetric-key encryption algorithm. To the exception of the content being encrypted, an IPsec packet is equivalent to regular IP packet.

1.3.3 WS-Security

WS-Security is a protocol to ensure the confidentiality and integrity of WS transactions. It has the advantage to ensure end-to-end security between the client and the WS application. Intermediaries have no access to the exchanged data and caching is still possible at a lower level. Additionally, there exists standards to sign and encrypt XML documents.

1.3.4 Firewall

A firewall is an intermediate network device that filters devices communication for security reasons. Commonly, it is used to separate the home network from the Internet. The transaction filtering is based on rules that define authorized communications. Often, the firewall acts at the IP level. Rules defines (source and destination) IP addresses and ports that are authorized. It can also, work at higher levels by analyzing exchanged transactions.

1.4 Overview

The Internet of Things (IoT) refers to the ever-growing network of physical objects that feature an IP address for Internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems. The Internet of Things extends Internet connectivity beyond traditional devices like desktop and laptop computers, smart phones and tablets to a diverse range of devices and everyday things that utilize embedded technology to communicate and interact with the external environment, all via the Internet. Examples of objects that can fall into the scope of Internet of Things include connected security systems, thermostats, cars, electronic appliances, lights in household and commercial environments, alarm clocks, speaker systems, vending machines and more. 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, smart homes, intelligent transportation and smart cities. Home automation or smart home (also known as domotics or domotica) is the residential extension of building or house automation and involves the control and automation of lighting, heating, ventilation, air conditioning (HVAC), and security, as well as home appliances such as washer/dryers, ovens or refrigerators/freezers that use Wi-Fi for remote monitoring. Modern systems generally consist of switches and sensors connected to a central hub from which the system is controlled with a user interface that is interacted either with a wall-mounted terminal, mobile phone software, tablet computer or a web interface, often but not always via Internet cloud services.So now we will build our home automation using ESP826612E wi-fi micro-controller connected to a Relay board and controlling the module with an app from any mobile device which runs on iOS or Android.So, our module basically consists of a micro-controller, relay board and some sensors. A module is a set of standardized parts or independent units that can be used to construct a more complex structure. Our system is optimized in cost and efficiency. It takes only $40 or 3500 BDT to build the module, Excluding the charge of a mobile device and the charges for using Internet. Costing varies depending on the number of devices and/or how powerful the system needs to be built for range. Our encoded micro-controller basically is a cloud server for receiving commands from mobile devices and then passing it to our Relay board which controls all the electrical devices.This micro-controller can be accessed over the Internet and/or using local wi-fi by an app from a mobile device.

1.5 Features

Our automation system feature things like turning on/off electrical appliances like fan, light, AC, TV etc. remotely, and because our system can be accessed over the Internet so one can check the status of the electrical device/devices through the mobile app and can manipulate it from anywhere he/she wishes. we also included a home security feature in our automation system which is unique in a sense of the true purpose of our module was to only automate the electrical devices of a home. In later chapters, we will describe how we managed to configure a simple security system out of it.In a nut-shell, we utilized our system in every possible aspect to the maximum level.The outcome of the project is a combination of embedded system and Programming.

1.6 Project Requirements

So, for this project we need roughly 7 components for fulfilling the project requirements and to build a complete automation system:

1. ESP8266 12E wi-fi/Node MCU
2. 4/8/16 channel Relay Board
3. USB TTL Serial Adapter
4. PIR Motion sensors
5. IR Sensors
6. Piezo Buzzer
7. Mobile or Tablet device

1.7 Advantages of the Project

With our system implemented on a home we take a step forward in future. As automation is one of the most required and used day-to-day basis modern technology for us, an automated home is a futuristic home to say the least. Our project aims for a home that is Safe, Green and interconnected at the same time. Our project aims for portability as if

you are carrying your home with you all the time. Using the remote controlling feature of our project one can easily find out if they turned off their lights, fans, AC, TV etc. this not only saves electricity for a better future also it is easy on those utility bills., Other risky electrical appliances like electrical stove or a coffee maker or an electrical kettle is dangerous when kept turned on for a long period of time as these can risk fire hazards, with our system implanted you can check the status of all these machines in your most used daily gadgets like a mobile or a tablet which you carry them around with you all the time. These are only some of the advantages of an interconnected system of electrical devices, there are a lot more. With our Home security feature, you can make your home even more secured. The cost of a Home security or to setup a burglar and fire alarm is paramount, but we have figured a way to make all these available to you 24/7 and its also cheap. So, a safe, green, interconnected home is the main advantages of our project. and yes, as there is no hesitation going on in your mind about these appliances on/off status a little peace of mind is also provided along with our system.

1.8 Chapter Summary

In this chapter, we explore the concepts of Internet of Things (IoT) and its prospects in building a home automation and security system. Here, we also try to define the project requirements and its possible implementation solutions.

2 Literature Review

In this chapter, we opt to review already existing related systems. These include Bluetooth, Zig-Bee, and wireless communication-based home automation systems etc.

2.1 Bluetooth Based Home Automation

Palaniappan et. al. [8] presented a Bluetooth based home automation system. In their research, they have presented (in Figure 2.1) the implementation of a home automation system using a cell phone and Bluetooth technology. Bluetooth technology is secured and low cost. It makes use of an Arduino Bluetooth board. An interactive python program is used in the cell phone to provide the user interface. The I/O ports of the Bluetooth board and relays are used for interfacing with the devices which are to be controlled. The Bluetooth is password protected to ensure that the system is secure and not misused by any intruders. The Bluetooth has a range of 10 to 100 meters, 2.4 GHz bandwidth and 3Mbps speed. The python app on the phone is portable. It is also a fast and cost effective system. The main drawback with respect to Bluetooth is that it takes a long time to discover and access devices in its vicinity. It does not provide energy conservation tips. Real time access cannot be achieved. Anywhere access to the devices cannot

be achieved. Access is limited to within the Bluetooth range. Bluetooth technology is used to control home appliances(as above figure 2.1showing). The client is a PC that is connected via USB to the Bluetooth module, sensor circuit and a pulse width modulation circuit. Sensors and actuators are used to control the circuit. The Bluetooth module that is connected to it will allow it to receive various commands via Bluetooth. Bluetooth devices can scan and detect other devices easily. It might also be possible to check whether devices are working properly or not. The system also has an illumination sensor that can turnon lights when external light is dull and a temperature sensor.

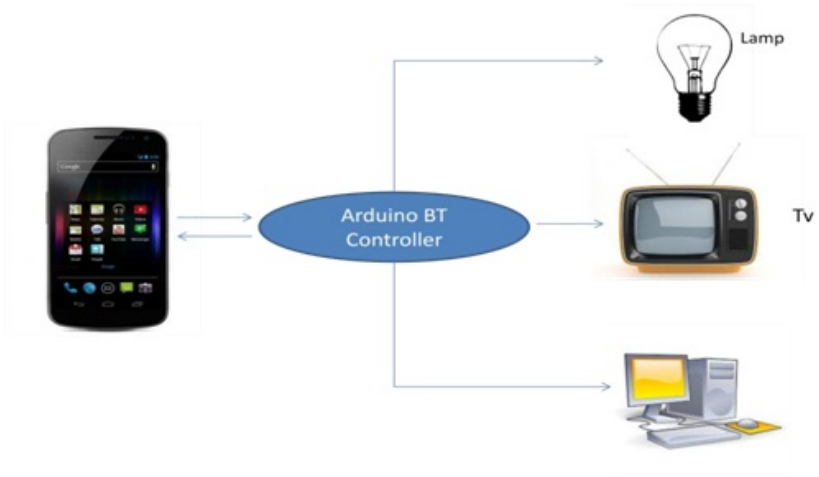


Figure 2.1: Bluetooth based home automation

2.2 Zig-Bee Based Home Automation

Palaniappan et. al. [8] also presented a Zig-Bee [9] based wireless communication technology can be applied for home automation system diagram as presented in Figure 2.2. The system uses PIC micro-controller and voice recognition for this purpose. The voice commands are taken from a Mic. They are compared with a voice store and processed. The PIC micro-controller then transmits the commands through Zig-bee to the receiver. The receiver unit has another PIC micro-controller that can process the command. It uses relays to control the respective appliances.This system has the drawback that Zig-bee is a low range communication medium. So, remote access is hindered from faraway locations. Also, the voice recognition module could become unwieldy. This system has the added feature of integrating a smoke detector to the system. When smoke is sensed, it sends a message to the users built-in mobile number.

unit has another PIC micro-controller that can process the command. It uses relays to control the respective appliances. This system has the drawback that Zig Bee is a low range communication medium. So, remote access is hindered from faraway locations. Also, the voice recognition module could become unwieldy. This system has the added feature of integrating a smoke detector to the system. When smoke is sensed, it sends a message to the users built-in mobile number.

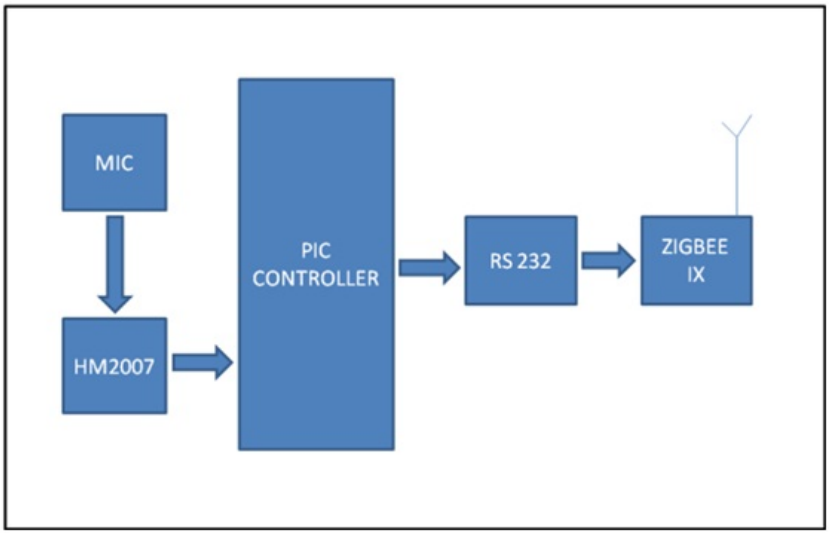


Figure 2.2: Zig-Bee based home automation

2.3 Wireless Control Systems

Palaniappan et. al. [8] presents a systems using wireless communication can be made by linking up standalone appliances that are present at home or in office and integrating to form a cooperating network. A combination of various technologies like Wi-Fi and Bluetooth are used to integrate the system. Such a system is laid out as illustrated in figure 3 & 4.The universal Plug and play capability is used to provide a transparent network of devices to the user. The system makes use of the Open Service Gateway Interface (OSGi). The appliances are connected via different networking technologies. The user application layer makes use of web browsers, pocket PC application and a central console. Speech based commands can also be used for controlling the appliances. Advanced features are provided such as device discovery and device connection. The entire system is implemented in a Linux platform. The system also has the ability to add intelligent control modules.In figure 2.3 and figure 2.4 we see block diagram of the proposed system.

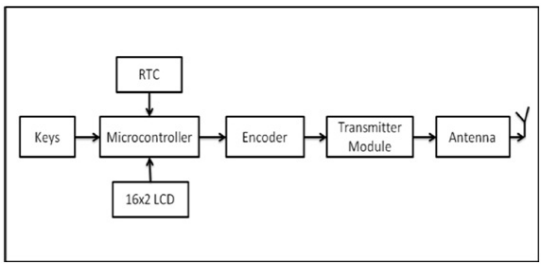


Figure 2.3: Block Diagram of Transmitter Section

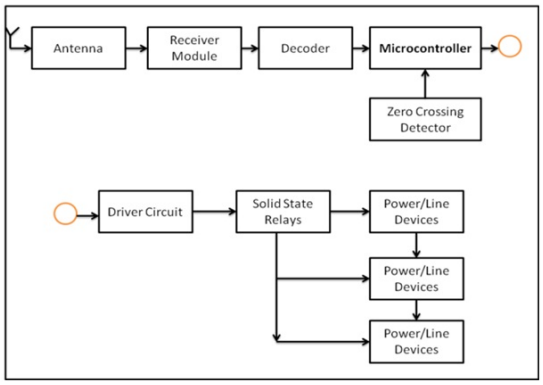


Figure 2.4: Block Diagram of a receiver Section

The system proposed uses an Arduino board along with Wi-Fi communication technology. It also has the option of another automated mode where there are temperature and light sensors which can be set to control the appliances depending on the conditions. The system uses an Android smart phone to communicate with a home PC those servers as a server. The PC is connected to and controls the Arduino board. The system proposed uses an Arduino board along with Wi-Fi communication technology. It also has the option of another automated mode where there are temperature and light sensors which can be set to control the appliances depending on the conditions. The system uses an Android smart phone to communicate with a home PC those servers as a server. The PC is connected to and controls the Arduino board.

2.4 Design and Implementation of a WiFi Based Home Automation System.

This paper presents a design and prototype implementation of new home automation system that uses WiFi technology as a network infrastructure connecting its parts. The

proposed system is a distributed home automation system, consists of server, hardware interface modules. Server controls hardware one interface module, and can be easily configured to handle more hardware interface module. The hardware interface module in turn controls its alarms and actuators. Server is a normal PC, with built in WiFi card, acts as web server. The web server software is developed using asp.net technology. System can be accessed from the web browser of any local PC in the same LAN using server IP, or remotely from any PC or mobile hand held device connected to the Internet.WiFi technology is selected to be the network infrastructure that connects server and hardware interface modules. WiFi is chosen to improve system security (by using secure WiFi connection), and to increase system mobility and scalability. The main functions of the server are to manage, control, and monitor distrusted system components, that enables hardware interface modules to execute their assigned tasks (through actuators), and to report server with triggered events (from sensors) Hardware interface modules are directly connected to sensors and actuator through direct wires connections. Hardware interface modules has the capabilities to control energy management systems like lighting, thermostats and HVAC (heating, ventilation, and cooling) systems, and security systems (door locks, cameras, motion detectors, fire alarm). The proposed home automation system has the capabilities to control the following components in users home and monitor the following alarms:

- Temperature and humidity
- Motion detection
- Fire and smoke detection
- Door status
- Light level
- Video monitoring

The proposed home automation system can control the following appliance:

- Serine
- Lights on/off/dim
- HVAC on/off

- Door lock
- Window shutdown
- on/off different appliance

Figure 2.5: The proposed home automation system layout

The Proposed home automation system(See figure 2.5) is implemented using ASP, HTML and CSS. The server application is implemented in ASP.Net, and the embedded hardware interface application shall be implemented using C Processing Language.

Assumptions and Dependencies:

- the component of the system will always be connected
- Each User must have a User ID and password
- There is only one Administrator.
- Server must always run under windows system
- There should be Internet connection available.
- Proper browsers should be installed
- Proper Hardware Components are available

- User is capable of using a computer

Software of the proposed home automation system is divided to server application software, and Micro-controller(Arduino) firmware. The server application software package for the proposed home automation system, is a web based application built using asp.net, Microsoft Visual Studio 2010. Server application software runs on windows OS, requires IIS web server, and .Net version 4.0 being installed. The server application software can be accessed from internal network or from Internet if the server has real IP on the Internet using any Internet navigator supports asp.net technology. Server application software is responsible of setup, configuration, maintain the whole home automation system. Server use database to keep log of home automation system components, we choose to use XML files to save system log. The Arduino software, built using C language, using IDE comes with the micro-controller itself. Arduino software is responsible for collecting events from connected sensors, then apply action to actuators and pre-programed in the server. Another job is to report the and record the history in the server DB.

2.5 NodeMCU/ESP8266 12E Wi-fi

NodeMCU (figure 2.6) is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi Soc from Espressif Systems, and hardware which is based on the ESP-12 module. The term ”NodeMCU” by default refers to the firmware rather than the Dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It is used in many open source projects, such as Lua-cjson, and spiffs.

2.6 Why NodeMCU/ESP8266 12E?

NodeMCU is the brain and operational head of our module. Though there are other renowned micro-controllers like Raspberry Pi, Arduino, Intel we found NodeMCU more compatible for our project. Here are some of the key features why we used NodeMCU in the first place.

Open source: NodeMCU is based on ESP-12 Module. The plans for the module is published under a common license so experienced and inexperienced designer can extend its capability by utilizing it to the way designers seems fit. And because its software is open source experts can upload their codes to help others. Anybody can download those codes and easily program it to make their own modules.

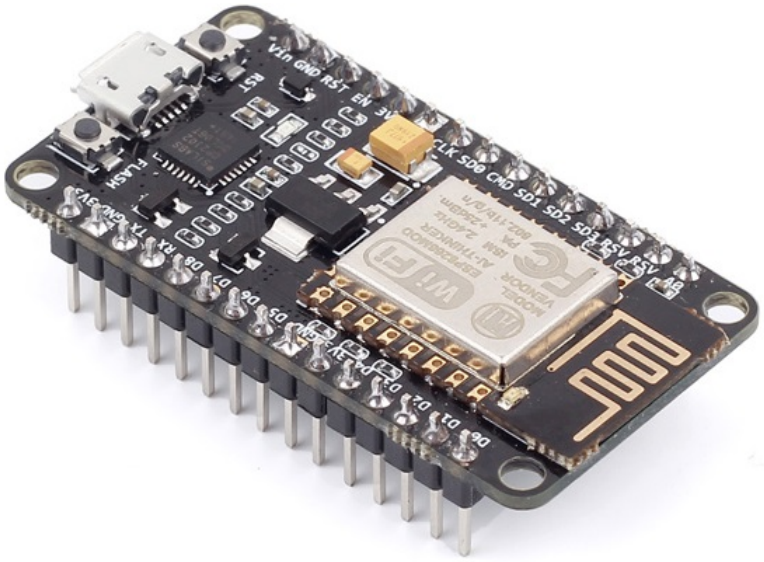


Figure 2.6: NodeMCU Micro-controller

Inexpensive: The NodeMCU board is relatively cheaper compared to other micro-controllers platforms. Including all the features it comes with, it is cheap to build a module using it. In fact, with the least amount of experience a simple module can be built by hand for under 30 dollars.

Built in Wi-Fi solution: ESP8266 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. In has integrated cache to improve the performance of the system in such applications.

Cross-platform: The NodeMCU drivers and software runs on Mac, Windows, and Linux operating systems and they are completely open source. Most micro-controller systems are limited to Windows for their IDE support.

Simple Board: Unlike some boards which are vastly complex to figure out and awfully difficult to program, the NodeMCU is easy to configure and encode program onto it. Extra feature of USB capability, ideal for Plug-n-Play solutions, allowing easy prototyping for developers.

Nodejs Style Network API: Event-driven API for network applications, which facilitates developers writing code running on a 5mm x 5mm sized MCU in Nodejs style. Greatly speed up your IOT application developing process.

Simple Programming Environment: To program the NodeMCU board you dont need a different or unknown IDE, it can be programmed easily by using the Arduino IDE. Necessary libraries, test codes, port connections and all other option one needs to program it can be easily found on Arduino IDE. Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. Code like Arduino, but interactively in Lua script.

2.7 Chapter Summary

Throughout this chapter, we saw multiple techniques proposed in different journals about some of the famous IoT applications. Also, a brief introduction to our Micro-controller and its benefits are also discussed.

3 System Design

In this chapter, we describe all the components of our proposed home automation and security system using Internet of Things (IoT) devices. Here we present details of IoT devices that we intend to use in order to implement the system. This includes low level details such PIN, Port, Power Consumption details that we will use in the next implementation chapter. Here we also describe the frameworks that these IoT devices support.

3.1 NodeMCU/ESP8266-12E

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi Soc from Espressif Systems, and hardware which is based on the ESP-12 module. The term ”NodeMCU” by default refers to the firmware rather than the Dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. NodeMCU (Figure 3.1) 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.

3.1.1 Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack

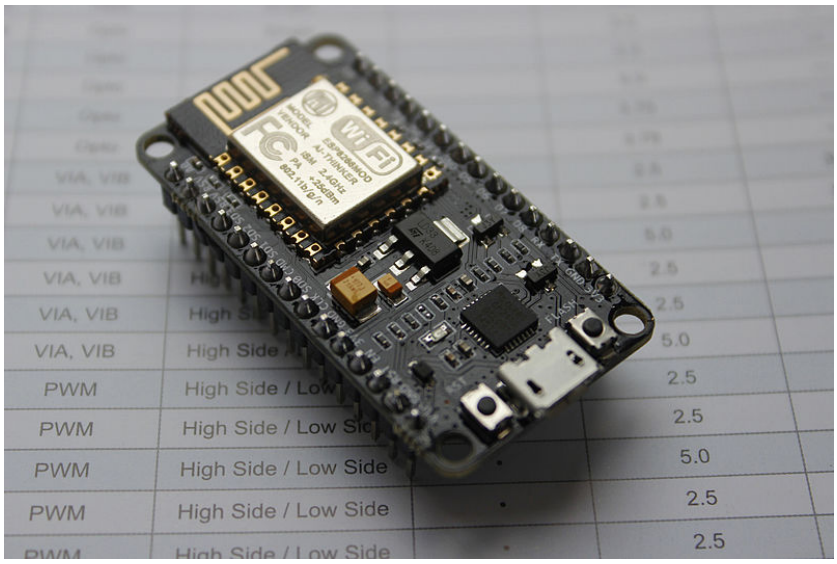


Figure 3.1: NodeMCU/ESP8266-12E

- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Wake up and transmit packets in ¡ 2ms
- Standby power consumption of ¡ 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode

- Operating temperature range -40C 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

3.1.2 Parameter

Here is a detailed Table 3.1 for the parameters of NodeMCU, which is based on the Expressif systems.

3.1.3 Pin Definition

In the Figure 3.2, the pin definition of NodeMCU is shown. And in the Figure 3.3, the pin out of NodeMCU is shown.Also in Table 3.2 a detailed pin description is given.

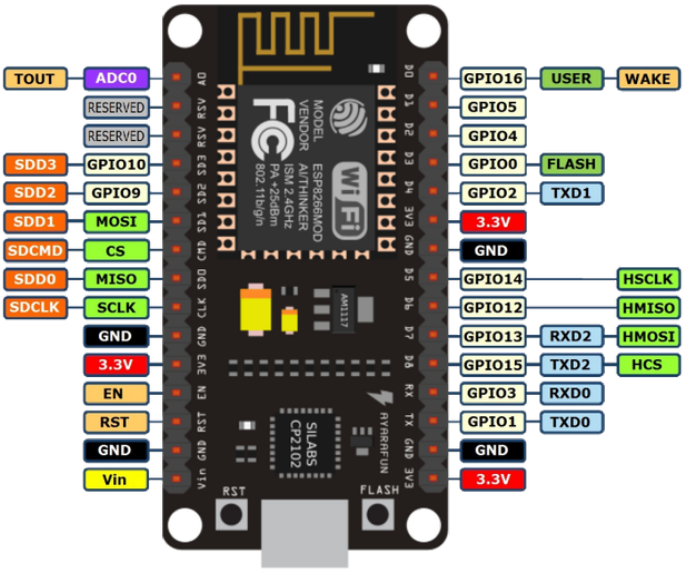


Figure 3.2: Pin Definition of NodeMCU

Table 3.1: Parameters of NodeMCU

Categories	Items	Values
Wi-Fi Parameters	certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	RX Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware Parameters	TX Power	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		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	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP temperature
	User Configuration	AT Instruction Set, Cloud Server, Android/ iOs App

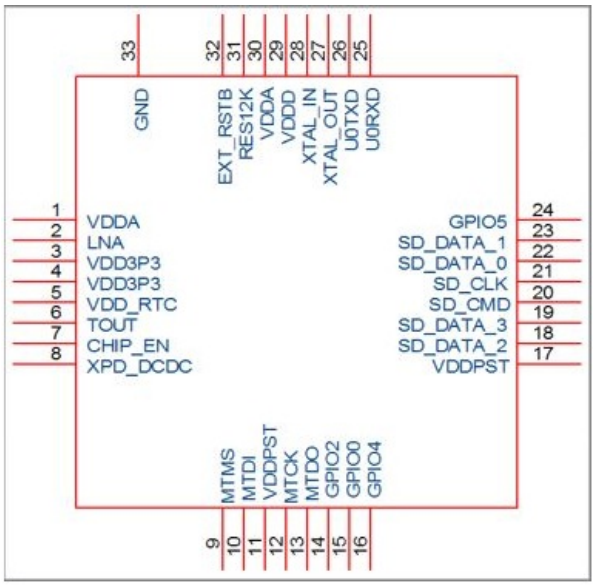


Figure 3.3: Pin out of NodeMCU

3.1.4 Power Consumption

The following current consumption is based on 3.3V supply, and 25C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitters measurements are based on 90% duty cycle, continuous transmit mode. See table 3.3 for detailed references, 1: Modem-Sleep requires the CPU to be working, as in PWM or I2S applications. According to 802.11 standards (like U-APSD), it saves power to shut down the WiFi Modem circuit while maintaining a WiFi connection with no data transmission. E.g. in DTIM3, to maintain a sleep 300mswake 3ms cycle to receive APs Beacon packages, the current is about 15mA 2: During Light-Sleep, the CPU may be suspended in applications like WiFi switch. Without data transmission, the WiFi Modem circuit can be turned off and CPU suspended to save power according to the 802.11 standard (U-APSD). E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive APs Beacon packages, the current is about 0.9mA.3: Deep-Sleep does not require WiFi connection to be maintained. For application with long time lags between data transmission, e.g. a temperature sensor that checks the temperature every 100s, sleep 300s and waking up to connect to the AP (taking about 0.3$\tilde{1}$s), the overall average current is less than 1mA

.

Table 3.2: Pin Description for NodeMCU

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.02 $\tilde{3}$.6 V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the π-type matching network is retained.
3	VDD3P3	P	Analog Power 3.02 $\tilde{3}$.6 V
4	VDD3P3	P	Analog Power 3.02 $\tilde{3}$.6 V
5	VDD3P3	P	Analog Power 3.02 $\tilde{3}$.6 V
6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin 4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep WakeupGPIO16
9	MTMS	I/O	GPIO14; HSPI_CLK
10	MTDI	I/O	GPIO12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V$\tilde{3}$.3V)
12	MTCK	I/O	GPIO13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	I/O UART TX during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	I/O GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V$\tilde{3}$.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V$\tilde{3}$.6V
30	VDDA	P	Analog Power 3.0V$\tilde{3}$.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)

Table 3.3: Power Consumption of NodeMCU

Parameter	Typical	unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm	170	mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm	140	mA
Tx 802.11n, MCS7, P OUT =+13dBm	120	mA
Rx 802.11b, 1024 bytes packet length, -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Modem-Sleep(1)	15	mA
Light-Sleep(2)	0.9	mA
Deep-Sleep(3)	10	uA
Power Off	0.5	uA

3.1.5 MCU

ESP8266EX is embedded with Ten silica 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. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the WiFi stack, the rest can all be used for user application programming and development. The following interfaces can be used to connect to the MCU embedded in ESP8266EX:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit external flash;
- Data RAM interface (dBus), which can connect with memory controller;
- AHB interface, can be used to visit the register.

3.1.6 Memory Organization

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 ¡ 36 KB, 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 36 KB.)

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

3.1.7 IR Remote Control

The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are used by this interface. The frequency of modulated carrier signal is 38 KHz, while the duty ratio of the square wave is 1/3. The length of data transmission, which is around 1m, is determined by two factors: one is the maximum value of rated current, the other is internal current-limiting resistance value in the infrared receiver. The larger the resistance value, the lower the current, so is the power, and vice versa. The transmission angle is between 15 and 30, and is mainly determined by the radiation direction of the infrared receiver.

3.1.8 Major applications

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics
- WiFi Location-aware Devices

- Security ID Tags
- WiFi Position System Beacons

3.2 Relay Board

A relay is a kind of switch which is controlled by an electric current. A relay makes it possible for a low voltage low current circuit to switch a high voltage and/or high current device safely for example, a small battery-powered light detector circuit can be used to control large mains powered security lights via a relay. The simplest relay board has just one relay, one LED used to indicate the status of the switch, one suppression diode to prevent back EMF from damaging connected electronics, and screw-in terminals for the voltage inputs and for the NC, NO, and COM connections on the relay. The relay board we are using (See figure 3.4) for our project is a 4-channel relay board which by the number of the channel explains has 4 relays

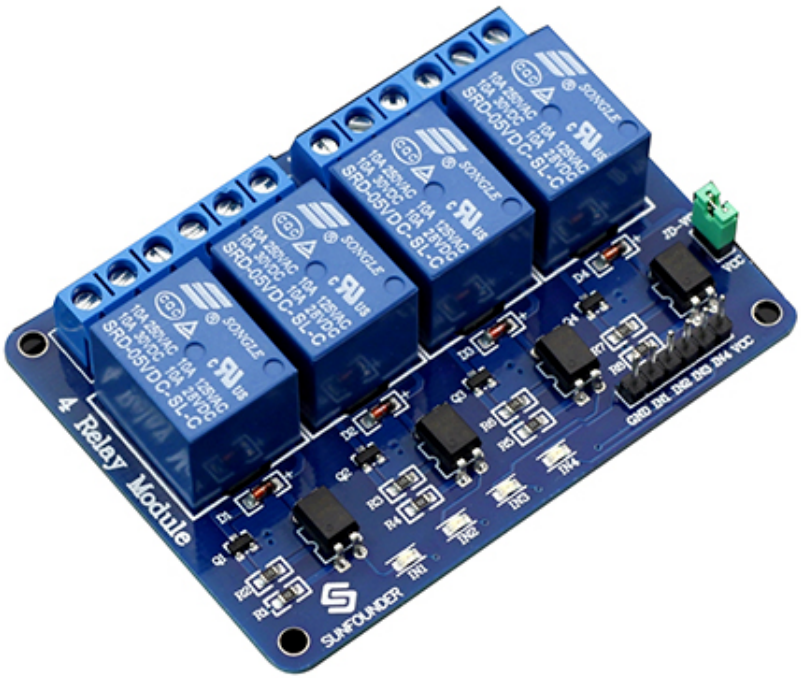


Figure 3.4: Relay Board

3.2.1 Pin Description

Pin descriptions of the Relay Board (see figure 3.5)is give below.
Input: Vcc: Positive supply voltage
GND: Ground
IN1–IN4: Relay control port
Output: Connect a load, DC 30V/10AAC 250V/10A

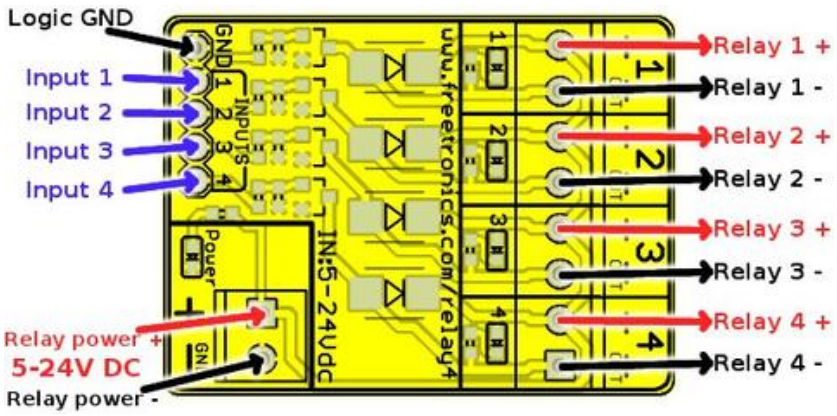


Figure 3.5: 4 channel relay board pin-outs.

- Logic GND: Connect to GND on your micro controller.
- Input 1: Connect to a digital output from your micro controller, or leave unconnected if channel not used.
- Input 2: Connect to a digital output from your micro controller, or leave unconnected if channel not used.
- Input 3: Connect to a digital output from your micro controller, or leave unconnected if channel not used.
- Input 4: Connect to a digital output from your micro controller, or leave unconnected if channel not used.
- Relay power +: Connect to the positive (+) lead of the power source for your relays. Can be 5 to 24V DC.

- Relay power -: Connect to the negative (-) lead of the power source for your relays.
- Relay 1 +: Connect to the + side of the coil of your first relay.
- Relay 1 -: Connect to the - side of the coil of your first relay. Relay 2/3/4 +: As per Relay 1 +
- Relay 2/3/4 -: As per Relay 1 -

3.2.2 Features

- Size: 75mm (Length) * 55mm (Width) * 19.3mm (Height)
- Weight: 61g
- PCB Color: Blue
- There are four fixed screw holes at each corner of the board, easy for install and fix. The diameter of the hole is 3.1mm
- High quality Single relay is used with single pole double throw, a common terminal, a normally open terminal, and a normally closed terminal
- Optical coupling isolation, good anti-interference.
- Closed at low level with indicator on, released at high level with indicator off
- VCC is system power source, and JD-VCC is relay power source. Ship 5V relay by default. Plug jumper cap to use
- The maximum output of the relay: DC 30V/10A, AC 250V/10A

3.2.3 Application

- Supports all MCU control.
- The industrial field.
- PLC control.
- Smart home control.

3.3 USB to TTL Adapter

A USB/TTL converter(See figure 3.6) based on CH340 high quality, high performance USB-to-Serial Bridge Controller that supports 3.3V and 5V TTL and CMOS levels using jumper selection. This is a low-cost USB to TTL module based on CH340 chip. CH340 is low cost alternative to FTDI USB serial Chips. The Module supports both 3.3V and 5V levels by a jumper selection. The module supports 50bps to 2Mbps baud rates

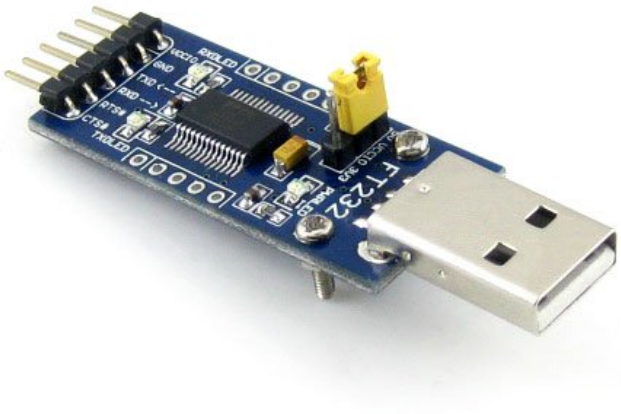


Figure 3.6: USB-TTL

3.3.1 Operating Instruction

To support 3.3V LVCMOS or LVTTL levels insert Jumper between VCC and 3.3V to support 5V CMOS or TTL levels insert Jumper between VCC and 5V.

- Supporting OS:
- Windows 10, 8, 7
- Macintosh OSX (v3.1)

- Linux 3.x.x,
- Linux 2.6.x
- Android

3.3.2 Features

- Implements full v2.0 USB protocol
- On board status indicator lights for RX and TX and Power
- Supported Baud rates: 50,75,100,110,134.5,150,300,600,900,1200,1800,2400,3600,4800,9600,14400,19

3.3.3 Applications

- Programming of Arduino Pro Mini with Automatic RESET through DTR pin
- Programming of Micro controllers that can be programmed using serial boot loaders at 3.3V or 5.0V logic levels
- Data exchange between micro controller and Desktop or Laptop Computers
- Connection with GPS Devices
- Connection with various external Phones, or modems, GSM Modules
- Simple UART communication,commonly used UART debugging tools in supper terminal
- 3.3V and 5V power supply interface easy for the DDWRT of different voltage system that need power.

3.4 PIR Motion Sensor

PIR sensors (see figure 3.7) allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and dont wear out. For that reason, they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, Passive Infrared, Pyroelectric, or IR motion sensors.

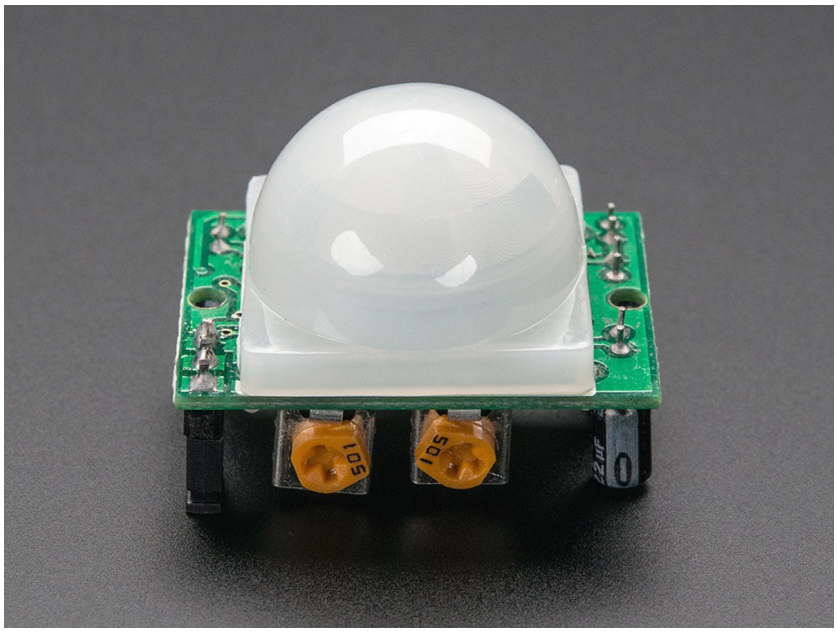


Figure 3.7: PIR motion sensor

3.4.1 Features

- Delay time adjustment Potentiometer
- Sensitivity adjustment Potentiometer
- Repeat Trigger Selection Jumper
- Complete with PIR, Motion Detection.
- Dual Element Sensor with Low Noise and High Sensitivity.
- Supply Voltage 5V.
- Delay Time Adjustable.
- Standard TTL Output.

3.4.2 Technical Specifications

- Dimension: 3.2cm x 2.4cm x 1.8cm (approx.)
- Infrared sensor with control circuit board
- The sensitivity and holding time can be adjusted
- Working Voltage Range: DC 4.5V- 20V
- Current drain:¡60uA
- Voltage Output: High/Low level signal:3.3V TTL output
- Detection distance: 37M (can be adjusted)
- Detection range: ¡140
- Delay time: 5-200S(can be adjusted, default 5s +-3
- Blockade time: 2.5 S (default)
- Trigger: L: Non-repeatable trigger H: Repeat Trigger (default)
- Work temperature: -20-+80C
- Trigger Method: L unrepeatable trigger / H repeatable trigger

3.4.3 Applications

- All outdoor Lights.
- Lift Lobby.
- Multi Apartment Complexes.
- Common staircases.
- For Basement or Covered Parking Area.

- Shopping Malls.
- For garden lights.
- Object detection.

3.5 IR receiver sensor

An infrared receiver, or IR receiver (See figure 3.8), is hardware that sends information from an infrared remote control to another device by receiving and decoding signals. In general, the receiver outputs a code to uniquely identify the infrared signal that it receives. This code is then used in order to convert signals from the remote control into a format that can be understood by the other device. It is the part of a device that receives infrared commands from a remote control. Because infrared is light, it requires line-of-sight visibility for the best possible operation, but can however still be reflected by items such as glass and walls.

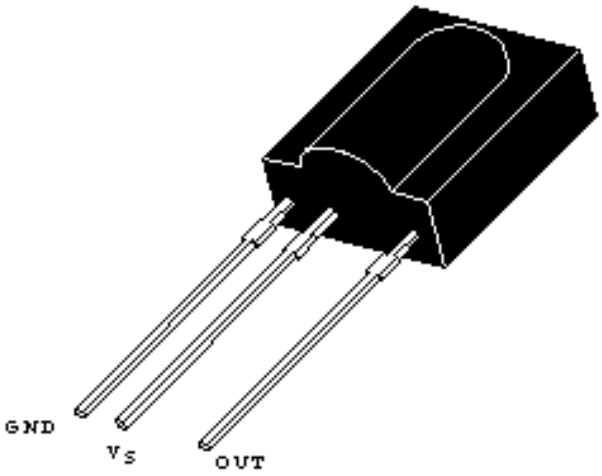


Figure 3.8: IR receiver

3.5.1 Features

- Very low supply current
- Photo detector and preamplifier in one package

- Internal filter for PCM frequency
- Improved shielding against EMI
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against ambient light
- Insensitive to supply voltage ripple and noise
- Component in accordance to RoHS 2002/95/EC and WEEE 2002/96/EC

3.5.2 Applications

- television remote controls.
- infrared ports such as PDAs, laptops, and computers.
- home theatres.
- cable or satellite receivers.
- VCRs, DVD and Blu-Ray players.
- audio amplifiers.
- industrial, military, aerospace and photography markets.

3.5.3 IR Transmitter

The IR (infrared) transmitter (see figure 3.9) sends a short pulse of modulated infrared light. You can activate it with a trigger from an input module (like a button). The IR transmitter sends an IR radiation (in the infrared wavelength region), which is reflected of a surface and falls upon a receiver due to the falling of light on the receiver a potential difference is created across the ends. This is recognized by a micro-controller as HIGH or LOW.

Figure 3.9: IR Transmitter

3.5.4 Features

- High reliability
- High radiant intensity
- Peak wavelength p=940nm
- 2.54mm Lead spacing
- Low forward voltage
- Pb free
- The product itself will remain within RoHS compliant version

3.5.5 Applications

- Free air transmission system.
- Infrared remote control units with high power requirement.
- Smoke detector.

- Infrared applied system.

3.6 Piezo Buzzer

Piezo buzzers (see figure 3.10) are used for making beeps, tones and alerts. This one is petite but loud! Drive it with 3-30V peak-to-peak square wave. To use, connect one pin to ground (either one) and the other pin to a square wave out from a timer or microcontroller. For the loudest tones, stay around 4 KHz, but works quite well from 2KHz to 10KHz. For extra loudness, you can connect both pins to a microcontroller and swap which pin is high or low ('differential drive') for double the volume.

Figure 3.10: Piezo Buzzer

3.6.1 Applications

- Judging panels
- Educational purposes

- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances
- Sporting events such as basketball games
- Electrical alarms
- Klaxon

3.7 BLYNK

Blynk [10] Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you on-line and ready for the Internet of Your Things. Blynk can be downloaded both from Apple Appstore and Google Play store.

3.7.1 Features

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using:
  - Ethernet
  - Wi-Fi
  - Bluetooth and BLE
  - USB (Serial)
- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing

- Easy to integrate and add new functionality using virtual pins
- History data monitoring via History Graph widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.

3.7.2 How does BLYNK work

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things. There are three major components in the platform:

- Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.
- Blynk Server - responsible for all the communications between the smart phone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. Its open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and out-coming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to space the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a Blynk of an eye. If you look at figure 3.11 the work flow of Blynk becomes a lot less complicated.

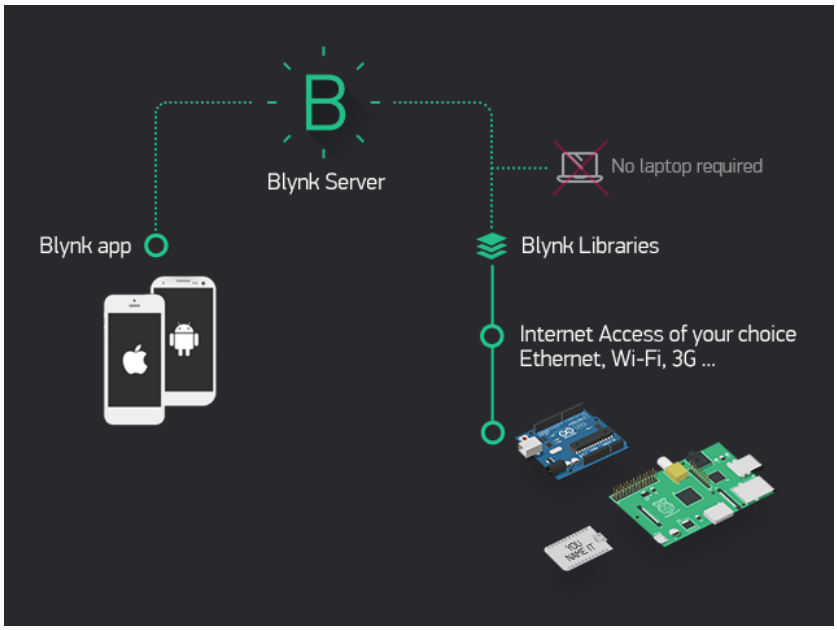


Figure 3.11: Blynk Work flow

3.7.3 Supported Platforms

Arduino

- Arduino Uno, Duemilanove
- Arduino Nano, Mini, Pro Mini, Pro Micro, Due, Mega
- Arduino 101 (Intel Curie, with BLE)
- Arduino MKR1000
- Arduino Zero
- Arduino Yn (onboard WiFi and Ethernet, via Bridge)

Arduino-like

- Blynk Board

- ESP8266 (Generic, NodeMCU, Witty Cloud, Huzzah, WeMos D1, Seeed Wio Link, etc.)
- ESP32 Dev Board
- Intel Edison
- Intel Galileo
- Teensy 3.2/3.1
- Blue Pill (STM32F103C)

Lua

- NodeMCU

Python (MicroPython)

- WiPy

3.7.4 Supported Connection Types

USB (Serial) connected to your laptop or desktop

Ethernet:

- Arduino Ethernet Shield (W5100)
- Arduino Ethernet Shield 2 (W5500)
- SeeedStudio Ethernet Shield V2.0 (W5200)
- ENC28J60-based modules

WiFi:

- ESP8266 as WiFi modem (running original firmware)

- Arduino WiFi 101 Shield
- Arduino WiFi Shield
- Adafruit CC3000 WiFi Breakout / Shield
- RN-XV WiFly

Bluetooth Smart (BLE 4.0):

- HM-10, HC-08
- DFRobot BLE-Link module
- Microduino/mCookie BLE
- RedBearLab BLE Mini
- nRF8001-based boards (Adafruit Bluefruit LE, etc.)

GSM/3G:

- SIMCom SIM800 series (SIM800A, SIM800C, SIM800L, SIM800H, SIM808, SIM868)
- SIMCom SIM900 series (SIM900A, SIM900D, SIM908, SIM968)
- GPRSbee
- Microduino GSM
- Adafruit FONA (Mini Cellular GSM Breakout)
- Adafruit FONA 800/808 Shield

3.8 Chapter Summary

So far in this chapter we see detailed information and specification of the components used in this project. the main aim of this chapter was to introduce all the components and their working architectures.

4 Implementation

In this chapter, we describe the implementation details of the project. Here we present the NodeMCU configuration with Relay, RGB-LED, IR-Receiver & IR-Sender, PIR sensor, and Piezo Buzzer for our proposed home automation and security system. Here we have written our program with Arduino IDE [11] using various standard libraries such as ESP8266, IR Transmit-Receive, Simple timer etc.

4.1 Configuring NodeMCU

Hardware Required:

1. NodeMCU/ESP8266 12E
2. USB cable
3. PC/Laptop

we have to be sure of the NodeMCU software serial number. Usual software serials are 4800, 9600, 115200 etc. Here we have used 115200 as our software serial.

4.1.1 Implementation

Listing 4.1: NodeMCU Configuration Code

```
1 #define BLYNK_PRINT Serial
2 #include <ESP8266WiFi.h>
3 #include <BlynkSimpleEsp8266.h>
4
5 char auth[] = "YourAuthToken";
```

```
6
7  char ssid[] = "YourNetworkName";
8  char pass[] = "YourPassword";
9  void setup()
10 {
11 Serial.begin(115200);
12 Blynk.begin(auth, ssid, pass);
13 }
14 void loop()
15 { Blynk.run(); }
```

4.2 Interfacing the NodeMCU with Relay board and RGB LED

Hardware required:

- NodeMCU/ESP 8266 12E
- Relay Board
- RGB LED
- Jumper wire

Figure 4.1 shows the connection schematics between NodeMCU with Relay and RGB.

4.2.1 Circuit

- NodeMCU Vcc and GND pin to Relay board Vcc and GND pin respectively.
- NodeMCUs GPIO pin D1, D2, D3, D4 to Relay Boards pin IN1, IN2, IN3, IN4 in orderly fashion.
- NodeMCU GPIO D7 pin to RGB LEDs R, G, B pin.
- NodeMCU Vcc pin to RGB LEDs anode pin.

we have to select NodeMCU 1.0 (ESP-12E module) from the board manager of the IDE.

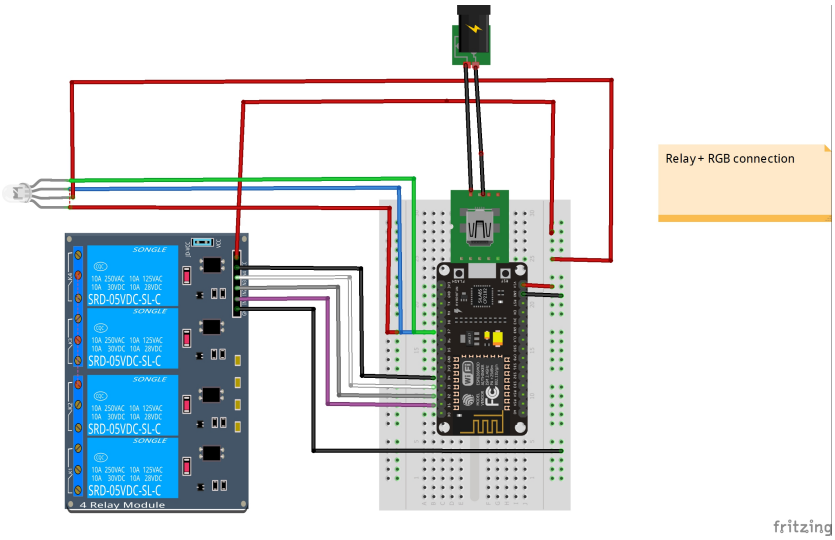


Figure 4.1: Relay + RGB connection schematics

4.2.2 Implementation

Listing 4.2: Code for interfacing NodeMCU and Relay board+RGB LED

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "FastLED.h"

#define LED_PIN     7
#define NUM_LEDS    1
#define BRIGHTNESS  64
#define LED_TYPE    WS2812
#define COLOR_ORDER GRB
CRGB leds[NUM_LEDS];

int r = 0;
int g = 0;
int b = 0;

char auth[] = "YourAuthToken";

```

```
19 char ssid[] = "YourNetworkName";
20 char pass[] = "YourPassword";
21
22 void setup()
23 {
24     delay( 3000 );
25 FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds,
        NUM_LEDS).setCorrection( TypicalSMD5050 );
26 FastLED.setBrightness( BRIGHTNESS );
27
28 Serial.begin(9600);
29 Blynk.begin(auth, ssid, pass);
30 }
31
32 void loop()
33 {
34 Blynk.run();
35 FastLED.clear();
36 leds[0] = CRGB(r, g, b);
37 FastLED.show();
38 }
39 BLYNK_WRITE(V15)
40 {
41    r = param[0].asInt();
42    g = param[1].asInt();
43    b = param[2].asInt();
44 }
```

4.3 Interfacing Relay Board to electrical appliances

Hardware required:

- Relay board
- Bulb
- Fan

- 220V power supply
- 9V battery
- Jumper wire

Figure 4.2 shows the connection schematics between Relay and Electrical Appliances.

4.3.1 Circuit

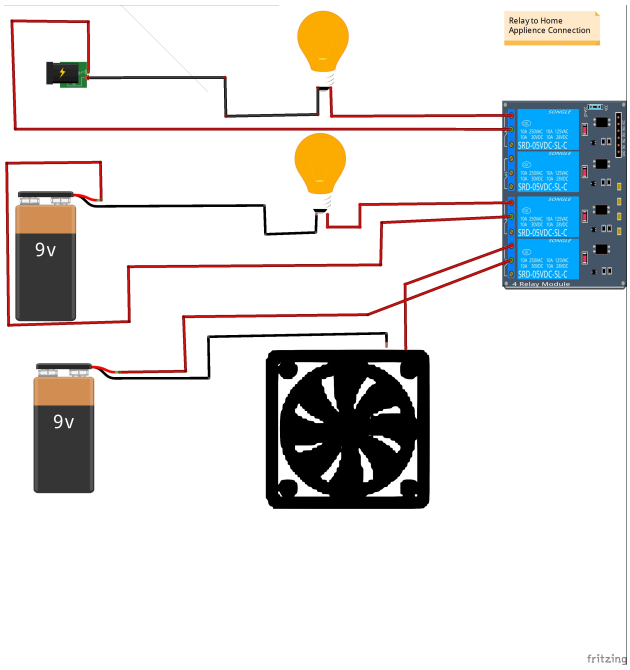


Figure 4.2: Relay to electrical appliances connection schematics

- Relay board NO (normally open) pin to Bulb and fans positive pin.
- Relay board COM (common) pin to Battery and power supplys positive pin.
- Batter and power supplys negative pin to fan and bulbs negative pin.

4.4 Interfacing the NodeMCU with IR receiver/ sender

Hardware required:

- NodeMCU
- IR LED
- 2N3904 Transistor
- 330 Ohm resistor
- IR receiver (TL1838) - Needed for reading codes

Figure 4.3 shows the connection schematics between NodeMCU with IR receiver/ sender.

4.4.1 Circuit

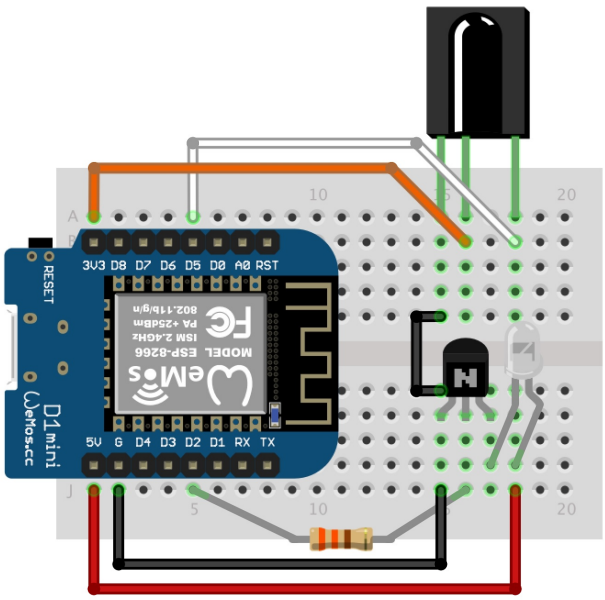


Figure 4.3: IR receiver/sender connection with NodeMCU

- NodeMCU 3.3V pin to IR receivers Vs pin.

- NodeMCU GPIO D5 pin to IR receivers OUT pin
- IR receivers GND pin to transistors Emitter pin.
- NodeMCU Vcc pin to IR LEDs Anode pin.
- NodeMCU GND pin to Transistors Emitter pin.
- NodeMCU GPIO D6 pin through a resistor to transistors Base pin.
- Transistors collector pin to IR LEDs cathode pin.

4.4.2 Implementation

Listing 4.3: Code for IR receiver

```
#include <IRremoteESP8266.h>

int RECV_PIN = 5;

IRrecvirrecv(RECV_PIN);

void setup ()
{
Serial.begin(9600);
irrecv.enableIRIn();
}

void  ircode (decode_results *results)
{

if (results->decode_type == PANASONIC) {
Serial.print(results->panasonicAddress, HEX);
Serial.print(":");
}

Serial.print(results->value, HEX);
}
```

```
23
24 void encoding (decode_results *results)
25 {
26 switch (results->decode_type) {
27 default:
28 case UNKNOWN:       Serial.print("UNKNOWN");    break ;
29 case NEC:           Serial.print("NEC");
      break ;
30 case SONY:          Serial.print("SONY");
      break ;
31 case RC5:           Serial.print("RC5");
      break ;
32     case RC6:            Serial.print("RC6");
         break ;
33     case DISH:           Serial.print("DISH");
         break ;
34     case SHARP:          Serial.print("SHARP");
         break ;
35     case JVC:            Serial.print("JVC");
         break ;
36     case SANYO:          Serial.print("SANYO");
         break ;
37     case MITSUBISHI:     Serial.print("MITSUBISHI");
         break ;
38     case SAMSUNG:        Serial.print("SAMSUNG");
         break ;
39     case LG:             Serial.print("LG");
         break ;
40     case WHYNTER:        Serial.print("WHYNTER");
         break ;
41     case AIWA_RC_T501:   Serial.print("AIWA_RC_T501");
         break ;
42     case PANASONIC:      Serial.print("PANASONIC");
         break ;
43   }
44 }
45 void  dumpInfo (decode_results *results)
```

```
46 {
47
48 Serial.print("Encoding  : ");
49   encoding(results);
50 Serial.println("");
51
52 Serial.print("Code      : ");
53 ircode(results);
54 Serial.print(" (");
55 Serial.print(results->bits, DEC);
56 Serial.println(" bits)");
57 }
58
59
60 void  dumpRaw (decode_results *results)
61 {
62
63 Serial.print("Timing[");
64 Serial.print(results->rawlen-1, DEC);
65 Serial.println("]: ");
66
67   for (inti = 1;  i< results->rawlen;  i++) {
68     unsigned long  x = results->rawbuf[i] * USECPERTICK
         ;
69     if (!(i& 1)) {  // even
70 Serial.print("-");
71       if (x < 1000)  Serial.print(" ") ;
72       if (x < 100)   Serial.print(" ") ;
73 Serial.print(x, DEC);
74     } else {
75 Serial.print("     ");
76 Serial.print("+");
77       if (x < 1000)  Serial.print(" ") ;
78       if (x < 100)   Serial.print(" ") ;
79 Serial.print(x, DEC);
80       if (i< results->rawlen-1) Serial.print(", ");
81     }
```

```
82      if (!(i % 8))  Serial.println("");
83    }
84  Serial.println("");
85  }
86  void  dumpCode (decode_results *results)
87  {
88
89  Serial.print("unsigned int  ");
90  Serial.print("rawData[");
91  Serial.print(results->rawlen - 1, DEC);
92  Serial.print("] = {");
93
94    for (inti = 1;  i< results->rawlen;  i++) {
95  Serial.print(results->rawbuf[i] * USECPERTICK, DEC);
96      if ( i< results->rawlen-1 ) Serial.print(",");
97      if (!(i& 1))  Serial.print(" ");
98    }
99
100 Serial.print("};");
101
102 Serial.print("  // ");
103   encoding(results);
104 Serial.print(" ");
105 ircode(results);
106
107 Serial.println("");
108
109   if (results->decode_type != UNKNOWN) {
110
111     if (results->decode_type == PANASONIC) {
112 Serial.print("unsigned intaddr = 0x");
113 Serial.print(results->panasonicAddress, HEX);
114 Serial.println(";");
115     }
116 Serial.print("unsigned int  data = 0x");
117 Serial.print(results->value, HEX);
118 Serial.println(";");
```

```
119     }
120  }
121
122  void  loop ( )
123  {
124  decode_results  results;
125
126     if (irrecv.decode(&results)) {
127  dumpInfo(&results);
128  dumpRaw(&results);
129  dumpCode(&results);
130  Serial.println("");
131  irrecv.resume();
132     }
133  }
```

Listing 4.4: Code for IR Sender

```
 1  #define BLYNK_PRINT Serial
 2  #include <ESP8266WiFi.h>
 3  #include <BlynkSimpleEsp8266.h>
 4  #include <IRremoteESP8266.h>
 5
 6  IRsendirsend(6);
 7  char auth[] = "Your Auth";
 8
 9
10  char ssid[] = "Your SSID";
11  char pass[] = "Your Password";
12
13  void setup()
14  {
15  Serial.begin(115200);
16  Blynk.begin(auth, ssid, pass);
17  }
18
19  void loop()
```

```
20 {
21 Blynk.run();
22 }
23
24 BLYNK_WRITE(10)
25 {
26   if (param.asInt()){
27 irsend.sendNEC(0xCF20D, 32);
28     delay(40);
29   }else {
30 irsend.sendNEC(0xC728D, 32);
31     delay(100);
32 irsend.sendNEC(0xC728D, 32);
33     delay(100);
34   }
35 }
36
37 BLYNK_WRITE(0)
38 {
39   if (param.asInt()){
40 irsend.sendNEC(0xC728D, 32);
41     delay(40);
42 irsend.sendNEC(0xC728D, 32);
43     delay(40);
44 irsend.sendNEC(0xC728D, 32);
45   }
46 }
47
48 BLYNK_WRITE(2)
49 {
50   if (param.asInt()){
51 irsend.sendNEC(0xC20DF, 32);
52     delay(40);
53   }
54 }
55
56 BLYNK_WRITE(3)
```

```
57 {
58   if (param.asInt()){
59 irsend.sendNEC(0xCD02F, 32);
60     delay(40);
61   }
62 }
63
64 BLYNK_WRITE(4)
65 {
66   if (param.asInt()){
67 irsend.sendNEC(0xC30CF, 32);
68     delay(40);
69   }
70 }
71
72 BLYNK_WRITE(5)
73 {
74   if (param.asInt()){
75 irsend.sendNEC(0xCA857, 32);
76     delay(40);
77   }
78 }
79
80 BLYNK_WRITE(6)
81 {
82   if (param.asInt()){
83 irsend.sendNEC(0xCA15E, 32);
84     delay(40);
85   }
86 }
87
88 BLYNK_WRITE(7)
89 {
90   if (param.asInt()){
91 irsend.sendNEC(0xC41BE, 32);
92     delay(40);
93   }
```

```
94  }
95
96
97  BLYNK_WRITE(8)
98  {
99     if (param.asInt()){
100 irsend.sendNEC(0xCC13E, 32);
101       delay(40);
102    }
103 }
```

4.5 Interfacing the NodeMCU with PIR sensor and Piezo Buzzer

Required Hardware:

- NodeMCU/ESP8266 12E
- PIR sensor
- Piezo Buzzer
- USB TTL adapter
- Jumper wire

Figure 4.4 Shows the connection schematics between NodeMCU with PIR sensor and Piezo Buzzer.

4.5.1 Circuit

- NodeMCU GPIO pin D0 to PIR sensors OUT pin
- USB TTL adapter 5V pin to PIR sensors Vcc pin
- USB TTL adapter GND pin to PIR sensors GND pin
- NodeMCU GPIO pin D8 to Piezo Buzzers Vcc pin.
- NodeMCU GND pin to Piezo Buzzer GND pin

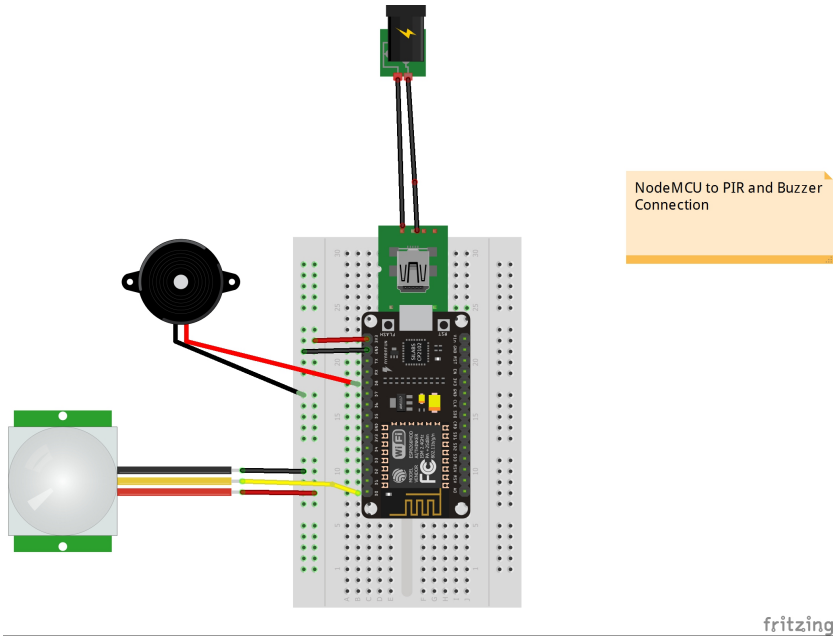


Figure 4.4: NodeMCU to PIR sensor + piezo buzzer connection schematics

4.5.2 Implementation

Listing 4.5: Code for NodeMCU with PIR sensor and Piezo Buzzer

```
1
2  #define BLYNK_PRINT Serial
3  #include <ESP8266WiFi.h>
4
5  #include <BlynkSimpleEsp8266.h>
6  #include <SimpleTimer.h>
7  char auth[] = "Your Auth";
8
9  char ssid[] = "Your SSID";
10 char pass[] = "Your Password";
11
12 #define SWITCH          V1
13 #define ledPin          D12
14 #define inputPin        D0
15 #define pinSpeaker      D8
16 #define STATUS_DELAY    2000L
```

```
17
18 intpirState = LOW;
19 intval = 0;
20 SimpleTimer timer;
21
22 bool enabled = false;
23 bool sendUpdate = false;
24
25 BLYNK_WRITE (SWITCH) {
26     enabled = param.asInt();
27 }
28
29 void sendNotify ( void ) {
30     if ( sendUpdate ) {
31 Blynk.notify("Motion detected");
32 Serial.println("Motion detected");
33 sendUpdate = false;
34     }
35 }
36
37 void setup() {
38
39 Serial.begin(115200);
40   delay(10);
41 Blynk.begin(auth, ssid, pass);
42
43 pinMode(ledPin, OUTPUT);
44 pinMode(inputPin, INPUT);
45 pinMode(pinSpeaker, OUTPUT);
46 timer.setInterval(STATUS_DELAY, sendNotify);
47 }
48
49 void loop(){
50 getPirValue();
51 Blynk.run();
52 timer.run();
53 }
```

```
54
55 void getPirValue(void){
56
57   if ( enabled ){
58 val = digitalRead(inputPin);
59   if (val == HIGH) {
60 digitalWrite(ledPin, HIGH);
61 playTone(300, 160);
62     delay(150);
63
64
65     if (pirState == LOW) {
66
67 sendUpdate = true;
68
69 pirState = HIGH;
70     }
71   } else {
72 digitalWrite(ledPin, LOW);
73 playTone(0, 0);
74       delay(300);
75       if (pirState == HIGH){
76 Serial.println("Motion ended!");
77 pirState = LOW;
78     }
79   }
80 }
81
82   }
83
84 void playTone(long duration, intfreq) {
85     duration *= 1000;
86 int period = (1.0 / freq) * 1000000;
87     long elapsed_time = 0;
88     while (elapsed_time< duration) {
89 digitalWrite(pinSpeaker,HIGH);
90 delayMicroseconds(period / 2);
```

```
91  digitalWrite(pinSpeaker, LOW);
92  delayMicroseconds(period / 2);
93  elapsed_time += (period);
94      }
95  }
```

4.6 Setting up the Live Stream with Blynk

4.2.2 Setting up the Live Stream with Blynk Here is the process of setting up a low cost and simple security system we designed along with our home automation system.

Required hardware:

- IP camera/mobile camera

Required Software:

- CameraFiLive (mobile app)
- youtube-dl(Command-line program to download videos from YouTube.com and other video sites)

4.6.1 Steps

1. Install CameraFi Live, configure it.
2. Install youtube-dl.
3. start live broadcasting at youtube.
4. Get youtube url.
5. Extract real url stream from youtube meta data.
6. Update blynk video widget.

4.7 Chapter Summary

In this chapter, System implementation of the project was shown throughly.Necessary Steps, Hardwares, codes and schematics were shown. Procedure for connecting all the components together was also briefly described.

5 Conclusion

In this very last chapter of the book we conclude all of our introduction, review, design and implementations. However, for this last chapter we would like to discuss about those problems we face during the development process of the project and also like to talk about our future goals concerning this very project.

5.1 Limitations

Parts availability

- It was hard finding a shop for purchasing good parts.
- Latest parts were not available.
- there were a not a lot of options to choose from.

Parts Costing

- Original parts costing was really high
- No warranty/guarantee was provided
- We saw lot of difference pricing from the original price of the parts manufacturer.

Risk assessment

- Fragile

- Working with 220V current is a risk itself.
- Burning the micro controller or any other parts was a risk which came down hard on costing.
- Electric shocks can be found regularly throughout the body if not careful.

Besides these main factors, we had to work with various voltages or power consumptions of the parts. which was difficult and tiresome. we lost parts in our experiments as we were novices in this matter to say the least.

5.2 Future Works

- We would like to showcase our project commercially.
- We would like to join different EXPO and display our project, and share our ideas for creating an automated system for a modern and advanced future approach.
- We would like to learn first-hand from the community itself and how we can bring more improvement in our project.
- We would like to Integrate our system with Voice Controlled Commands.
- Working on local server
- We would like to work on bigger scales.
- integrate more electrical appliances.
- We would like to explore further steps in Surveillance, Alarm systems and Emergency responses.

With this project completed a sense of achievement is felt by us. we stepped out of the way and thought of doing something that can be used in the real world and is a technology that is future proof. We declare the project complete as we achieved all the goals we targeted from the beginning.

References

[1] S. Tale, A. Kakad, N. Bhawarkar, D. Patil, and P. Nimat, "Control system for home automation based on iot," in *Proceedings of the National Conference on Innovative Trends in Science and Engineering*, ser. NC-ITSE'16, vol. 4, no. 7, 2016, pp. 324–326.

[2] R. Harper, *Inside the smart home*. Springer Science & Business Media, 2006.

[3] R. L. Smith, *Smart house: The coming revolution in housing*. Gp Courseware, 1988.

[4] Z. Zheng *et al.*, "Home automation: Smart home technology and template house design," 2013.

[5] M. Chauhan, N. Gulati, and A. Kaur, "Verbalised home automation," *International Journal of Engineering Research and Management Technology*, vol. 3, no. 2, 2016.

[6] Phys.org, "The latest at gadgets: Samsung fridge comes with screen (update 10)," https://phys.org/news/2016-01-latest-gadgets-armour-unveils.html, Accessed on: 2017-01-01.

[7] A. Germis, "An open home automation system."

[8] S. Palaniappan, N. Hariharan, N. T. Kesh, S. Vidhyalakshimi *et al.*, "Home automation systems-a study," *International Journal of Computer Applications*, vol. 116, no. 11, 2015.

[9] ZigBee Alliance, "What is ZigBee," http://www.zigbee.org/what-is-zigbee/, Accessed on: 2017-01-01.

[10] Pavel Baiborodin, “The most popular mobile app for the IOT,” http://www.blynk.cc/, Accessed on: 2017-01-01.

[11] Arduino, “Download the Arduino IDE,” https://www.arduino.cc/en/main/software , Accessed on: 2017-01-01.