# ​​​​​1.1 INTRODUCTION

A load controlled by computer systems has many advantages compared with manual controlled loads. Nowadays there are many programs and applications help to control things better using codes or python algorithms in artificial intelligence projects. In order to save energy and make loads monitored easily, this research suggests smart home project based on IoT technology. This smart home is an Internet of Things (IoT) project that controls loads with internet connection via Wireless Fidelity WIFI connection. A smart phone connected to internet with Blynk application as a control panel, and NodeMCU microcontroller kit in other side as a controller that receives control commands via WIFI signal. NodeMCU kit is built with ESP8266 WIFI receiver that able to process and analyze WIFI signal to input the microcontroller. The WIFI receiver and microcontroller are built in one kit to be used as IoT project. It’s called NodeMCU.

To connect the system to the Internet, needs a WiFi receiver. In my case I used ESP8266 that is connected as built-in in the NodeMCU board that contains a firmware runs with the ESP8266. The firmware is a low-level control computer software.

The NodeMCU is coded via Arduino Integrated Development Environment (IDE) with the Universal Serial Bus port (USB) to tell the NodeMCU what to do, I want to make the NodeMCU controls four-channel relay kit by Blynk hand phone application and shows the temperature that measured by LM35 sensor.

Parts used to create the project:

1. NodeMCU board. Open source internet of things platform.
2. AC-DC step down converter. Switch mode power supply to provide the project with power. This project needs 5 volts.
3. DC-DC step down converter as a regulator to convert the 12 V output of the power supply into regulated 5 V.
4. Four-channel relay kit. To drive loads from digital NodeMCU output pins.
5. LM35 temperature sensor. To measure room temperature.
6. Computer with Arduino (IDE) program installed to code the NodeMCU once.
7. Android smart phone with Blynk application installed to be used as control panel.

# METHOD

This research is conducted based on the important steps that are done by orienting on the success indicators in connecting the NodeMCU ESP8266 module and other devices so that it can be used to solve multi-objective problems. To achieve these indicators, the stages of this research are as follows:

1. Analysis of the problem. Analyze the problems to be studied regarding smart home.
2. Analysis of needs. In this case all needs in researching both from journals, literature books, tools, and materials.
3. System design. Designing tools to be built using the NodeMCU ESP8266 module, and the sensors used.
4. System programming. Make a program using the Arduino IDE and the Blynk android application.
5. Testing tools. Testing tools with program codes created and internet connections.
6. Making reports and summarizing the results of the experiment. See system responsiveness to commands given to smart home.

**Motivation**

It should be noted whenever customers want to buy any product through online from any e-commerce website he/she does visit the many e-commerce websites for getting the desire product. Like this customer surf lots of time in visiting of e-commerce websites for getting the desire product. The customer not only surfs lots of time in visiting of e-commerce sites, and quality of products, but also he/she suffers from limited option to choose the product. The proposed architecture, by making use of location-based service, offers a solution to those problems. That is, for customers there is no need to visits many e-commerce websites for buying the desire the product. A user can buy desired product without visiting many e-commerce website as well as without wasting of time in visiting of e-commerce website. Location-Based Smart Shopping using Android provides a stage to the customer where the user can get information of a particular product available in stores nearby. Moreover, a user is also provided navigation facility which will direct him to the shop he wants to buy the product from

**Advantages Over Current System**

In various areas there is a need of constant surveillance. The current surveillance system includes monitoring by using CCTV cameras and other monitoring system. Mostly these systems are stationary and they can cover a limited area. These systems are mostly control manually or through a computer. They cannot be used to cover a larger area as well as they cannot be controlled using any mobile device. In short we can say that these systems are dynamic enough much which gives the need for the development of a surveillance system which is more dynamic and can be controlled remotely. This project is aimed at developing a surveillance system which can be controlled remotely by using an Android App. It includes a robot with a Wireless Camera attach to it. This robot captures the high resolution video feed and transmits it to the connected Android device which is used to control the robot.

The robot in the project is made to move in all the four directions using the Android phone. The circuit is built around ATMEGA 328 controller board, Blue-tooth module HC-05, motor driver L293D (IC1), DC motors M1 and M2, and a few common components. The circuit uses two 9V batteries. First battery is used to power the ATMEGA 328 controller board and the other is used to power the motors. The regulated 5V supply for the rest of the circuit is provided by the ATMEGA 328 controller board itself. LED on the board indicates presence of power supply. Motor Driver H-Bridge is used to drive two motors which work on 9v DC batteries. DCmotors are interfaced to the Micro controller. The data received by the Blue-tooth module from Android smart phone is fed as input to the controller.

**Formulation of Problem With using Technology**

Why Arduino?

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia’s BX-24, Phidgets, MIT’s Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems. Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. Simple, clear programming environment - The Arduino programming environment is easy-touse for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it’s conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

Open source and extensible software- The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it’s based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to. Open source and extensible hardware - The Arduino is based on Atmel’s ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

**Review Of Literature**

What is Surveillance robot ? Surveillance robot is the robot used for the surveillance purpose. The remote areas are watched using the surveillance robots.

**Wireless Controlled Surveillance Robot**

**Description**

A mobile robot is a machine that is basically place or mounted on a movable platform and can be with the help of certain instructions. In todayâTMs world a lot of fields use mobile robots. Many of the complex robots that we now see have originated from the simpler mobile robots. This technology has increased many new applications in the industry.[4] The combination of mobile devices and robots are leading to new ideas in lots of fields. The mobile devices are now being used in many of the industrial applications this is mainly because of the reason that they are portable and have a longer battery life as compared to a laptop. Also they have a data plan through a cell phone carrier which is convenient as we can interact with the mobile robot once the connection is established. Mobile Robots: The mobile robots can be classified into different types. The track robot is the robot that uses tracks to move around. However such robots are costly to build.[10] Also they are not as flexible as the wheeled robots. The wheeled robots are the robots which use wheels for moving. Such robots can move only on smooth flat surfaces. The third type is the legged robots which are based on human form. They have legs which helps them to move around. These robots are very difficult to design.

Proposed System The new age of technology such as Android, GSM has redefined communication. Most people nowadays have access to mobile phones and thus the world indeed has become a global village. At any given moment, any particular individual can be contacted with the mobile phone. New innovations and ideas can be generated from it that can further enhance its capabilities. Technologies such as Infra-red, Bluetooth, Wi-Fi which has developed in recent years goes to show the very fact that improvements are in fact possible and these improvements have eased our life and the way we live. Remote management of several home and office appliances is a subject of growing interest and in recent years we have seen many systems providing such controls. Mobile robots are robots which have the ability to move around and interact with their environment and not just hinged to a particular place. There are many labs and research groups from various universities and industries which are completely dedicated on researching mobile robots, because of their immense potential and varied application in industry, military, security, and entertainment. The robot is specially designed for surveillance purpose. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through high speed image transmission. Initially, the robot will be equipped with an Android smartphone which will capture the scenario in front of it will transfer the images to the server on which the user will be controlling and watching the live feed.

Pros

1. Infrared LED: 8pcs infrared LED, automatic operate in dark environment Resolution: VGA (640x480)/ QVGA(320x240)/ QQVGA(160x120)

2. Motion detection to trigger alarm

Cons

1. Limited Frequency Range: The frequency range used for typical RF communication is near about 3KHz-3GHz. The use of channel separator increases the reliability but decreases the actual usable working frequency range.

2. Limited Functions: The limited number of channels causes less number of combinations and thus there are less numbers of available functions.

3. Limited Working Range:

The working range of RF circuits with transmitters and receiver is very small. It starts from a few meters to a few kilometres. The working varies from circuits to circuits, but mainly depends on the values of physical components used in the circuit. Mainly Wi-Fi and Wi-Max wireless services are used in RF transmitter and receiver circuits. The following table shows the actual working range of different wireless standards that can be used in wireless communication.

4. Reliability of Operation:

The RF circuits are very prone to errors due to external conditions such as EMI (Electro-Magnetic Interference), medium saturation, absorption due to repetitive reflections from surface. Hence the output recovered is not always what is expected. This might be a serious problem when working with scientific experimental components. 5. Security reasons: This is the main disadvantage of using a RF circuit and the main reason why RF circuits are not preferred today. The RF frequency band is available for almost all the users for data communication. So there might be a scenario where more than one user is trying to accommodate channel for its own communication. In such case the frequency band may get interference from another user. Or worst case would be, some user intentionally trying to jam our communication network. The RF jammer circuits are very easy to design; hence the question of security arises when RF circuit is used in the circuit. This security loop hole can be very dangerous when the robot is being used for very confidential purposes. In areas of military these security threats can produce disastrous outcomes.

How we overcome Those problem in Project

1. Wifi connection is used for the operation of robot.

2. This gives the high security.

3. It also provide much more reliability of operation working range it also includes wifi.

4. N type wifi is used for higher security purpose as well as for better range.

**Surveillance Security Robot With Automatic Patrolling Vehicle**

**Description**

The word surveillance may be applied to observation from a distance by means of electronic equipment (such as CCTV cameras), or interception of electronically transmitted information (such as Internet traffic or phone calls). It may also refer to simple, relatively no- or lowtechnology methods such as human intelligence agents and postal interception. Surveillance is very useful to governments and law enforcement to maintain social control, recognize and monitor threats, and prevent/investigate criminal activity. However, many civil rights and privacy groups, such as the Electronic Frontier Foundation and American Civil Liberties Union, have expressed concern that by allowing continual increases in government surveillance of citizens we will end up in a mass surveillance society, with extremely limited, or non-existent political and/or personal freedoms. An automatic patrolling vehicle acts as a security patroller in the security system, which can monitor those dead zones of the traditional fixed surveillance system. The remote monitoring capabilities can also be enhanced by using the wireless network. And the face detection system is adapted to record and analyze the invaders. System Architecture The proposed self-propelled monitoring and surveillance Vehicle can be divided into the following parts: Wireless IPCAM video capture system, face detection system, remote monitor and alarm transmitter system, RFID position detection systems, and cell phone monitoring and control system The diagram of system architecture The self-propelled vehicle uses RFID technology to control the moving direction. RFID tag is installed in the right hand side of the self-propelled vehicle. When the self-propelled vehicle moves to a predefined routing path installed with RFID reader, the RFID reader would detect the RFID tag and send the signals back to the server to show the detected position on the map to indicate the status of the automatic vehicle.

**Technological Review**

Android (operating system)

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics. As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. A developer survey conducted in April May 2013 found that 71 % of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013. As of 2015, Android has the largest installed base of all general purpose operating systems. Android’s source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Arduino Board

What is Arduino?

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It’s an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Why Arduino?

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia’s BX-24, Phidgets, MIT’s Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems: • Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50 • Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

• Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it’s conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino

• Open source and extensible software- The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it’s based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

• Open source and extensible hardware - The Arduino is based on Atmel’s ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

**Bluetooth**

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz[4]) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization. Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 25,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics.The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks.A manufacturer must make a device meet Bluetooth SIG standards to market it as a Bluetooth device. A network of patents apply to the technology, which are licensed to individual qualifying devices.

**Wi-Fi**

Wi-Fi (or WiFi) is a local area wireless technology that allows an electronic device to participate in computer networking using 2.4 GHz UHF and 5 GHz SHF ISM radio bands. The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network" (WLAN) product based on the Institute of Electrical and Electronics Engineers’ (IEEE) 802.11 standards". However, the term "Wi-Fi" is used in general English as a synonym for "WLAN" since most modern WLANs are based on these standards. "Wi-Fi" is a trademark of the Wi-Fi Alliance. The "Wi-Fi CERTIFIED" trademark can only be used by Wi-Fi products that successfully complete Wi-Fi Alliance interoperability certification testing. Many devices can use Wi-Fi, e.g. personal computers, video-game consoles, smart phones, digital cameras, tablet computers and digital audio players. These can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points. Depiction of a device sending information wirelessly to another device, both connected to the local network, in order to print a document. Wi-Fi can be less secure than wired connections, such as Ethernet, because an intruder does not need a physical connection. Web pages that use TLS are secure, but unencrypted internet access can easily be detected by intruders. Because of this, Wi-Fi has adopted various encryption technologies. The early encryption WEP proved easy to break. Higher quality protocols (WPA, WPA2) were added later. An optional feature added in 2007, called Wi-Fi Protected Setup (WPS), had a serious flaw that allowed an attacker to recover the router’s password. The Wi-Fi Alliance has since updated its test plan and certification program to ensure all newly certified devices resist attacks.

**Requirement Analysis**

**Platform Requirement :**

**Supportive Operating Systems for Server :**

1. Ubuntu 12.4 and above The supported Operating Systems For server include Linux. Linux is used as server operating system.

2. Window 7 and above Microsoft Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft. Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs).

3. Android OS Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

Supportive Operating Systems for Client:

1. Android OS Android is open source operating system based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation

**Software Requirement : The Software Requirements in this project include:**

**Eclipse :**

Eclipse Luna (4.4.1) Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment.

**JDK :**

Java Platform (JDK) 8u25 The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application.

**JVM :**

JVM Version 8 A Java virtual machine (JVM) is an abstract computing machine. There are three notions of the JVM: specification, implementation, and instance. The specification is a book that formally describes what is required of a JVM implementation. Having a single specification ensures all implementations are interoperable. A JVM implementation is a computer program that meets the requirements of the JVM specification in a compliant and preferably performant manner. An instance of the JVM is a process that executes a computer program compiled into Java bytecode.

**Hardware Requirement :**

The Hardware components required for our project are Min 1 GB of RAM,10 GB HDD, Dual core processor for the machine on which development will be done,Robot kit, nodemcu Module etc for developing the robot.

Hardware Required For Project Development:

1. Arduino robot environment

2. Rotor motor

3. Batteries

4. node mcu

5. Wifi module

6. Chassis

**Implementation Details**

Assumptions And Dependencies

The new age of technology such as Android, GSM has redefined communication. Most people nowadays have access to mobile phones and thus the world indeed has become a global village. At any given moment, any particular individual can be contacted with the mobile phone. New innovations and ideas can be generated from it that can further enhance its capabilities. Technologies such as Infra-red, Bluetooth, Wi-Fi which has developed in recent years goes to show the very fact that improvements are in fact possible and these improvements have eased our life and the way we live. Remote management of several home and office appliances is a subject of growing interest and in recent years we have seen many systems providing such controls. Mobile robots are robots which have the ability to move around and interact with their environment and not just hinged to a particular place. There are many labs and research groups from various universities and industries which are completely dedicated on researching mobile robots, because of their immense potential and varied application in industry, military, security, and entertainment. The robot is specially designed for surveillance purpose. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through high speed image transmission. Initially, the robot will be equipped with an Android Smartphone which will capture the scenario in front of it will transfer the images to the server on which the user will be controlling and watching the live feed.

Implementation Methodologies

The project is designed to control a robotic vehicle using an android application. Bluetooth device is interfaced to the control unit on the robot for sensing the signals transmitted by the android application. This data is conveyed to the control unit which moves the robot as desired. An Atmel 89C51 microcontroller is used in this project as control device. Remote operation is achieved by any smart-phone/Tablet etc., with Android OS, upon a GUI (Graphical User Interface) based touch screen operation. Transmitting end uses an android application device remote through which commands are transmitted. At the receiver end, these commands are used for controlling the robot in all directions such as forward, backward and left or right and captures the video and transmits to TV through RF signal At the receiving end the movement is achieved by two motors that are interfaced to the microcontroller. Serial communication data sent from the android application is received by a Bluetooth receiver interfaced to the microcontroller. The program on the microcontroller refers to the serial data to generate respective output based on the input data to operate the motors through a motor driver IC. The motors are interfaced to the control unit through motor driver IC.

**Modular Description of Project**

Bluetooth module

The module provides a method to connect wirelessly with a PC or Bluetooth phone to transmit/receive embedded data such as GPS data, ADC voltage reading and other parameters. Bluetooth module JY MCU BT used in the project can be connected to any device, via built in UART interface to communicate with other Bluetooth -enabled devices such as mobile phones, handheld computers and laptops. The module runs on a 3.6V to 6V supply.

Arduino

Arduino micro-controller is intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments.

L293d

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal to run four solenoids, two DC motors or one bi-polar or unipolar stepper with up to 600 mA per channel using the L293D. These are known as the drivers in the blynkt Motor shield.

The L293 and L293D are quadruple high-current half-H drivers. In L293 is designed to provide bidirectional drive currents up to 1 A at voltage range from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltage range from 4.5 V to 36 V. Both the devices are designed to drive inductive loads such as relays, dc and bipolar stepping motor as well as other high-current/high-voltage loads in positive-supply applications.

The Android application is basically divided into two modules i.e.

1. Video Streaming Module

2. Robot control module

A network camera or an Android mobile phone is mounted on the robot, which our application fetches the live video streaming display it. This video is achieved using WIFI technology. The second module is the control module. Our application provides a GUI to control the robot wirelessly. This control is achieved using Bluetooth technology. Button are used to control the robot in forward, backward, left or right direction.

Detailed Analysis and Description of Project The android application will be used to search for products and view stores at which those products are available. The android application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user. The GPS will provide the mobile application with locations of both the user and the stores and the distance between them, but it will also provide maps and the functionality to display the application data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner. Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The android application will communicate with the database. The user will use the application to get data from the database while the vendor will also add and modify data, which will be monitored by the admin. All of the database communication will go over the Internet. The mobile application has some restrictions about the resource allocation.

## Smartphone Applications

Over the past few years, the market of touch screen mobile devices has experienced an enormous growth to the extent that, according to the last surveys , around half of the **US** mobile consumers own smart phones. The Indianmobile market as measured by active subscribers of the top 50 networks is 860 million and the rate of smart phone adoption is accelerating, and is soon expected to reach a third of the sales.

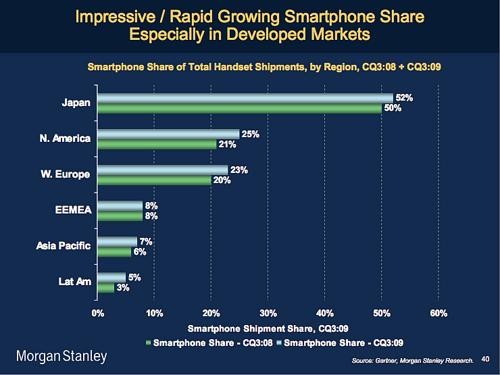
*A smartphone is a mobile phone built on an operating system, with more advanced computing capability than a feature phone (media players, camera, GPS, internet browsing, etc.).*

Although the term was coined years before, the real push that opened this market was the original iPhone by Apple Inc. in 2007, one of the first mobile phones to use a multi-touch interface. After, in July 2008, Apple announced its second generation phone with **3G** support. By then, the App Store reached over 1000 million downloads in the first year having started with only 500. Two more versions of the iPhone have been released so far, being Apple the leading company in all aspects from design to functionality .

Following the success of the Apple’s App Store other smartphone manufacturers soon launched their own software application stores, such as Google’s Android Market or Blackberry’s App World between others.

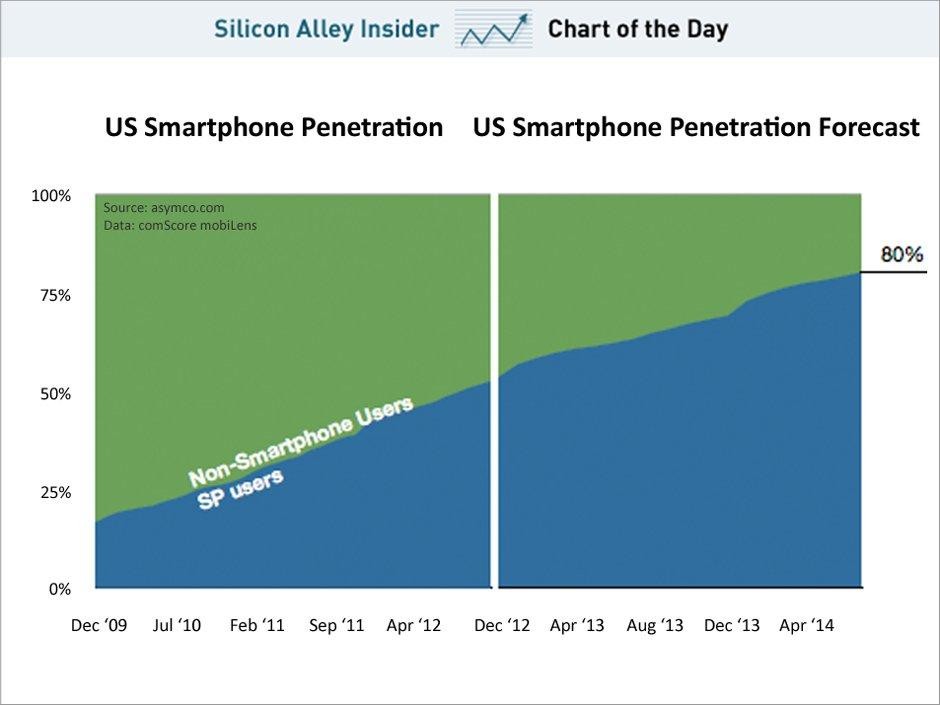
The Applications market is highly attractive for small companies and third-parties. In 2012, the Apple’s Store recorded $5782 million of revenues, relatively high compared to other competitor’s stores. This could be attributed to having the largest number of applications or apps available as well as the highest download volume in 2010. Also, only 28% of the apps in the Apple Store were free compared to the 57% in the Android Market .

In the next image we can see how deep the smart phone share is in each market:



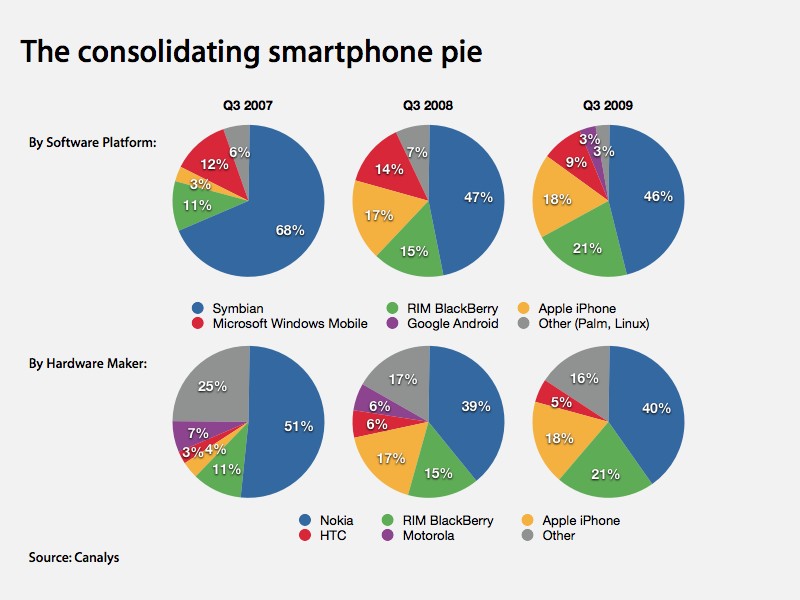
**Smartphone share per region**

As we can see, the leading market for smart phone sales is the Japanese market, followed by the Indian market. The special case of Japan can be explained by their early and massive use of mobile devices, with more rotation than in other markets and always 3 or 4 years ahead. In Japan, almost all technological advances are in more widespread than in other countries and the smart phone market has never been an exception for this. Forecasts are optimistic about a complete penetration of smartphones for all mobile device users as we can see next (by the end of 2014 an 80% of population is supposed to be carrying a smartphone):



**smartphone penetration**

In the rising years of smartphone technology the software and hardware market had the next distribution:

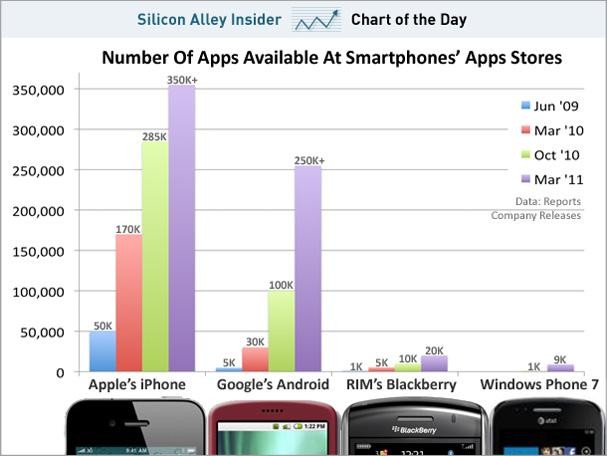


**Software and hardware platform pie**

We can see that both in software and hardware markets, Apple has experienced a notable growth while Nokia has decreased its share accordingly. It is important to remember that before the smartphone era, Nokia was the undisputable leader and Apple was not even a player in this market.

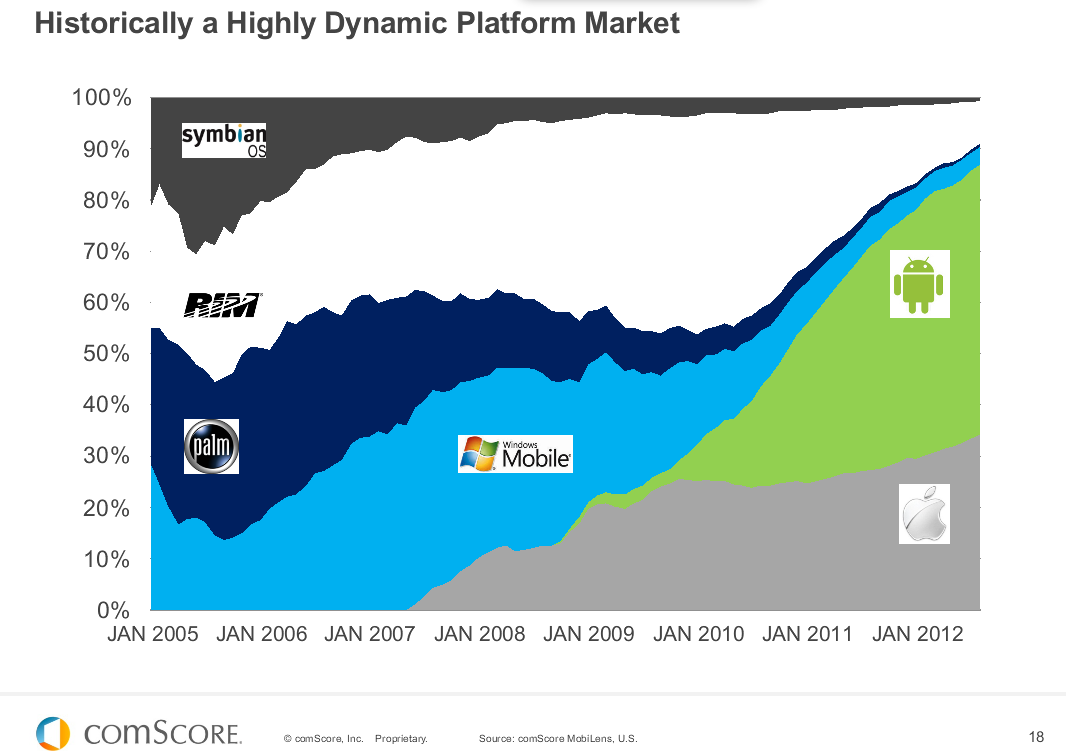
Nowadays, however, Apple’s iOS and Google’s Android are the two biggest competitors in both number of applications and market share, which translates into smartphone sales as users perceive the availability and utility of the applications software as a key factor for a mobile device election. Also, in the last 2 years tablets have joined this hardware market competing in the same application stores. Tablets are intended for different reasons (with less portability, bigger screens, less connectivity) but share almost all apps with smartphones.

The next chart shows the number of applications in each store for every of the last years.



**Apps available per platform**

This actually does not translate into sales (hardware and software) in the same relation as shown for number of applications offered per store. In the next chart we can see how Apple’s App Store is losing ground against Android, which is its major competitor right now. Before smartphones were ruling the technological panorama the most installed systems were from Nokia (Symbian), Microsoft, Palm and Blackberry (Rim).



**Platform market development**

As mentioned before, Apple’s iPhone meant a breaking point in both the perception users have from a mobile OS and the performance and usability that companies give to their systems (fig 31.). New mobile devices allow for user-friendly applications in all aspects possible, from internet browsing, web services, camera applications, etc. These requirements were only met first by Apple and also now by Android.

In conclusion, with this information we can foresee an attractive future for investors and developers in the applications software environment. Smartphones are becoming more than just a communicating tool to become all-in-one devices with which we can control and monitor any other electronic devices subject to wireless communications.

At this point, when developing applications most companies choose building multi-platform applications for both Android and iPhone. Both operating systems are widely used and cannot be set aside. However there are big differences in both worlds while developing and programming that should be noticed.

On one hand, **Android** is completely open source, has a large community of developers and is a light-weight operating system. It is completely free to start developing for it, and has a vast database of resources available on the internet, but applications should be runnable by many different devices, with different hardware configurations and computing power.

On the other hand, the **iOS** for iPhone is only intended to work in iPhones thus making it easier for developers to develop focusing less on hardware requirements and more on code. It is a proprietary system available only for Apple’s devices and it costs $99 to become an iOS developer and purchase the complete developing toolset (version for developers outside Apple).

# METHOD

This research is conducted based on the important steps that are done by orienting on the success indicators in connecting the NodeMCU ESP8266 module and other devices so that it can be used to solve multi-objective problems. To achieve these indicators, the stages of this research are as follows:

1. Analysis of the problem. Analyze the problems to be studied regarding smart home.
2. Analysis of needs. In this case all needs in researching both from journals, literature books, tools, and materials.
3. System design. Designing tools to be built using the NodeMCU ESP8266 module, and the sensors used.
4. System programming. Make a program using the Arduino IDE and the Blynk android application.
5. Testing tools. Testing tools with program codes created and internet connections.
6. Making reports and summarizing the results of the experiment. See system responsiveness to commands given to smart home.

# The Flow of The System

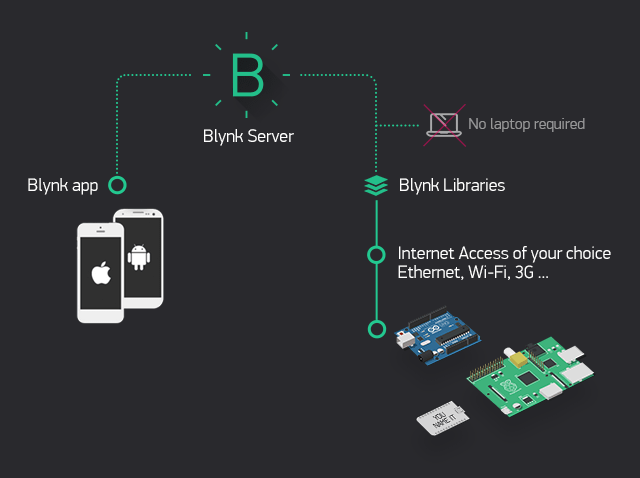
****

Figure 1. Blynk System Principle

The system is based on NodeMCU board as an internet of things system. The NodeMCU is connected to the internet from the hotspot of the smart phone via WIFI connection as the NodeMCU has ESP8266 circuit to connect with the internet.

NodeMCU to be connected to the hotspot of the smart phone, needs to be identified to the name of hotspot, the password and token code letting the server of Blynk connects them together. You may need the computer once to transfer code from Arduino IDE to the NodeMCU kit to prepare the software part of the project. Figure 1 shows that the server of Blynk application will process the smartphone-NodeMCU connection. Blynk libraries are ZIP files can be downloaded from Github website to be imported to the Arduino IDE library.

NodeMCU ESP8266 initialized

Blynk server will check for internet connection, NodeMCU with android hotspot, the NodeMCU code includes the token code, the name of hotspot and it’s password. The information included to the code must be match with the hotspot information to allow ESP8266 connect with the WIFI to be as a channel to exchange commands between smart phone and NodeMCU. Remaining processes are just commands sent from Blynk application to NodeMCU to control loads those are connected to the relay kit as shown in Figure 2. And sensor output value is sent reverse to the Blynk application from NodeMCU kit.



\

Figure 2. Flowchart of Load ON/OFF

To show the temperature value in Celsius degrees on the android display, NodeMCU will send sensor output value in voltage to the Blynk application back. Like the ON/OFF process last flowchart, Blynk server will check for internet connection and hotspot name and password, the sensor output value to show the temperature correctly. The temperature is showed by gauge tool in the Blynk application after setting the input pin and temperature scale as shown in Figure 3**.**



Temperature sensor





Figure 3. Flowchart of Temperature Sense

# The Block Diagram of the System

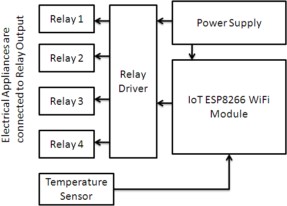
****

Figure 4. System Block Diagram

Figure 4 shows the system block diagram. The Power Supply will provide energy to the system through the relay and NodeMCU ESP8266 modules, so that all equipment can work and function properly. NodeMCU ESP8266 microcontroller will read the temperature by the Temperature sensor LM35, and then send the data to the Blynk server in TCP / IP format for display on the smart phone. NodeMCU ESP8266 microcontroller will also read commands that have been sent by the Blynk Server in TCP / IP format which will then be changed by giving the logic "HIGH" or "LOW" on certain pins by relay to regulate the on / off of the home lights. Cloud (internet) by utilizing Wi-Fi becomes the central connection between Blynk application and NodeMCU project.

# Blynk application and Arduino IDE Preparation and Running

This project is running by Blynk application. Down load the application to a smart phone from Google play store and then create a project on it with four switches and one gauge to be as a temperature scale. Set buttons to be switches on D1, D2, D3 and D4. Then set gauge on A0 because the sensor output is on A0 in NodeMCU board. Figure 5 shows screenshots from Blynk application

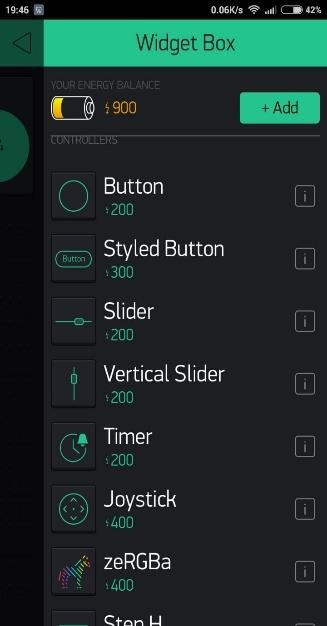
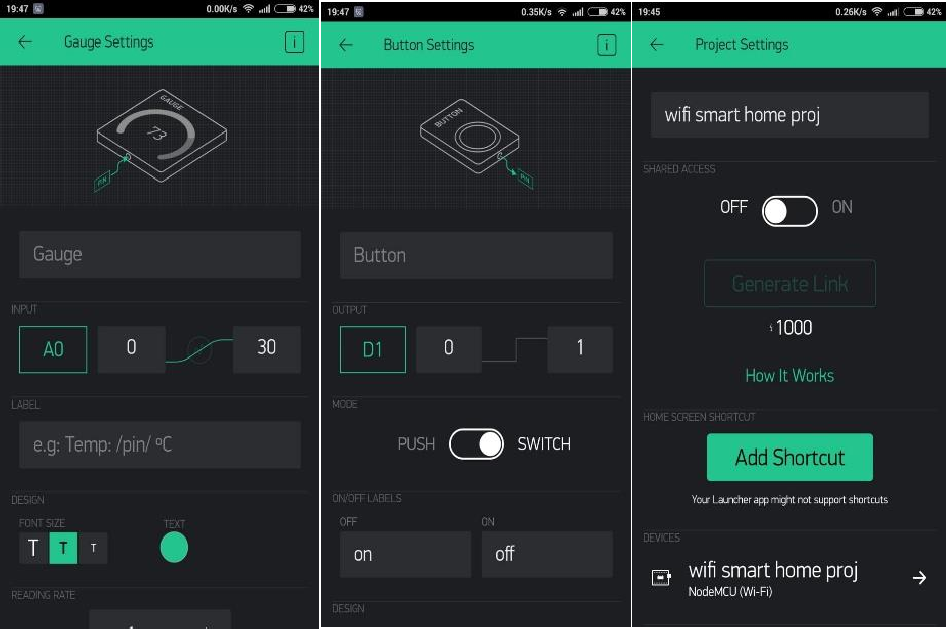


Figure 5. Screenshots from Blynk Application

# NodeMCU Code via Arduino IDE

To code NodeMCU via Arduino IDE, the NodeMCU needs to be added to Arduino IDE library first by adding this address to Arduino IDE preferences. After this reference is added to Arduino IDE, download nodeMCU to boards manager and then select NodeMCU 1.0 (ESP- 12E Module). After nodeMCU is added to Arduino IDE library, upload this code with changing hotspot name and password also token code. Shown in figure 6.

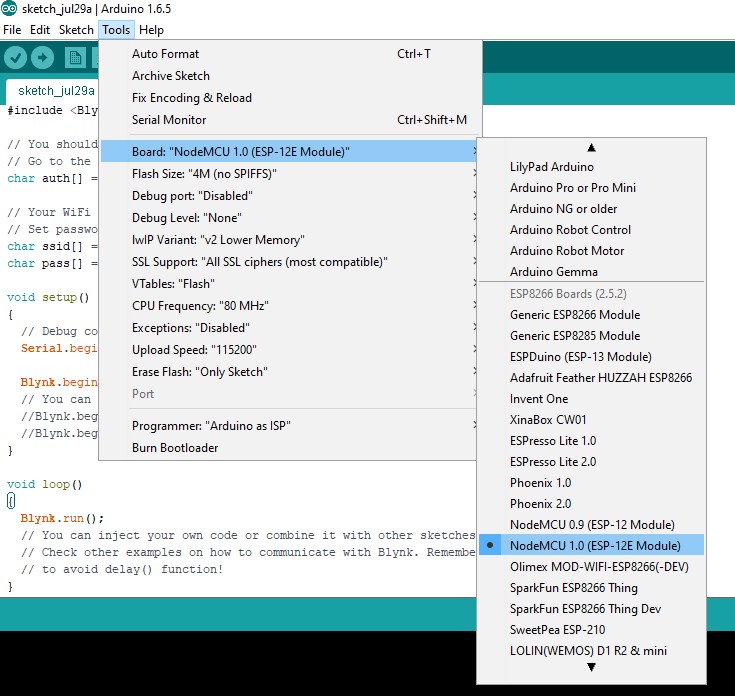


Figure 6. Setting up NodeMCU in Arduino IDE

Figure 7 shows the NodeMCU code. The code includes the hotspot name and password match with the android. The code does not need to identify the relay input, as it is included in

[Blynk.run();]. When auth (autho token) is given by Blynk application sent as email and SSID is the name of smart phone hotspot.

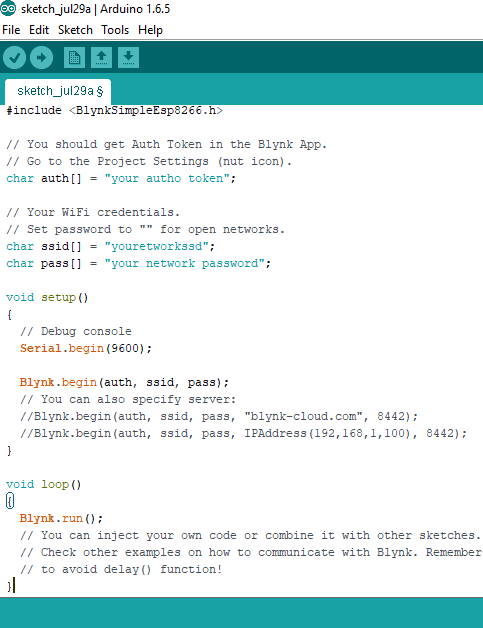


Figure 7. NodeMCU Code

# The Hardware of the System

As mentioned above, components used to build the circuit, NodeMCU needs 5VDC as a supply voltage Vin pin, AC-DC step down converter 12V and DC-DC step down converter 5V, in case using AC-DC step down converter 5V, no need to use DC-DC converter. Output voltage of the power supply is connected to Vin NodeMCU, Vcc of relay kit and VCC of LM35 temperature sensor. When the ground is common. D1,D2,D3 and D4 are outputs and A0 is an analog signal input is connected to the temperature sensor as shown in Figure 8. Using Fritzing software to draw and simulate the circuit as shown in Figure 9.

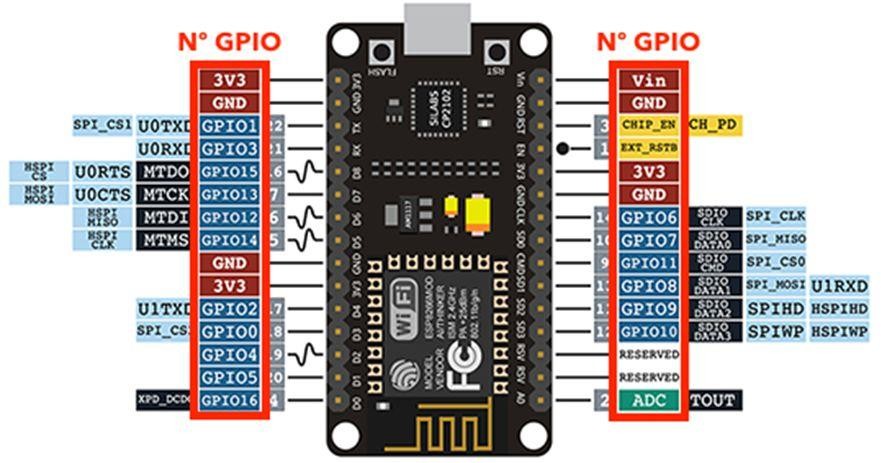


Figure 8. NodeMCU Pinout

Pins used:

* + 1. Vin is connected to power supply output 5VDC.
    2. GND is ground.
    3. D1,D2,D3 and D4 are used as digital outputs.
    4. A0 is used as analog signal input to input sensor signal.

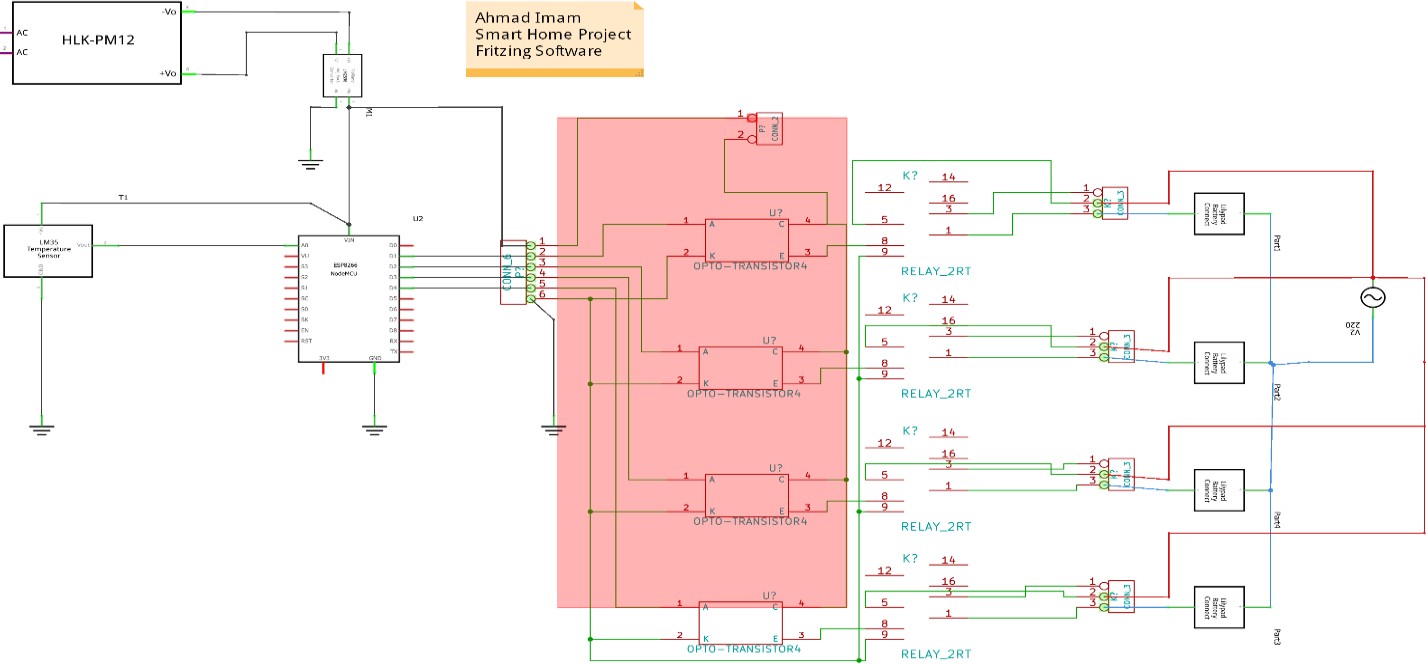


Figure 9. The Circuit Diagram

* + 1. Coding

#include <ESP8266WiFi.h>  
#include <ESPAsyncTCP.h>  
#include <ESPAsyncWebServer.h>

long duration;  
int distance;

const int trigP = D4;    
const int echoP = D8;    
const char\* ssid = "\*\*\*\*\*\*";  
const char\* password = "\*\*\*\*\*\*";  
const int trigP = D4;    
const int echoP = D8;    
const int MA1 = D6;       
const int MA2 = D7;     
const int MB1 = D1;      
const int MB2 = D3;

// Create AsyncWebServer object on port 80  
AsyncWebServer server(80);

String getDistance() {  
 digitalWrite(trigP, LOW);   // Makes trigPin low  
delayMicroseconds(2);       // 2 micro second delay

digitalWrite(trigP, HIGH);  // tigPin high  
delayMicroseconds(10);      // trigPin high for 10 micro seconds  
digitalWrite(trigP, LOW);   // trigPin low

duration = pulseIn(echoP, HIGH);   //Read echo pin, time in microseconds  
distance= duration\*0.034/2;        //Calculating actual/real distance

Serial.print("Distance = ");        Serial.println(distance);  
  return String(distance);  
}  
    
const char index\_html[] PROGMEM = R"rawliteral(  
<!DOCTYPE HTML><html>  
<head>  
  <title>Robo control Web Server</title>  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <style>  
  html {  
    font-family: Arial;  
    display: inline-block;  
    margin: 0px auto;  
    text-align: center;  
  }  
  h1 {  
    color: #0F3376;  
    padding: 2vh;  
  }  
  p {  
    font-size: 1.5rem;  
  }  
  .button {  
    display: inline-block;  
    background-color: #008CBA;  
    border: none;  
    border-radius: 4px;  
    color: white;  
    padding: 16px 40px;  
    text-decoration: none;  
    font-size: 30px;  
    margin: 2px;  
    cursor: pointer;  
  }  
  .button3 {  
    background-color: #f44336;  
  }

   </style>  
</head>  
<body>  
  <h1>Web Controlled Robo Car</h1>  
  <p>  
    <a href="/forward"><button class="button">FORWARD</button></a>  
    <a href="/backward"><button class="button button2">BACKWARD</button></a></p>  
  <p>  <a href="/stop"><button class="button button3">STOP</button></a></p>  
  <p>  <a href="/right"><button class="button button4">RIGHT</button></a>  
      <a href="/left"><button class="button button5">LEFT</button></a>  
  </p>  
  <p>  
    <span class="sensor-labels">Distance</span>  
    <span id="distance">%Distance% cm</span>  
  </p>  
</body>  
<script>  
  setInterval(function ( ) {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
      if (this.readyState == 4 && this.status == 200) {  
        document.getElementById("distance").innerHTML = this.responseText;  
      }  
    };  
    xhttp.open("GET", "/distance", true);  
    xhttp.send();  
  }, 5000 ) ;  
</script></html>)rawliteral";

void setup(){  
  Serial.begin(115200);

pinMode(trigP, OUTPUT);    
pinMode(echoP, INPUT);     
 pinMode(MA1, OUTPUT);  
  pinMode(MA2, OUTPUT);  
   pinMode(MB1, OUTPUT);  
    pinMode(MB2, OUTPUT);  
     pinMode(ENA, OUTPUT);  
      pinMode(ENB, OUTPUT);  
  // Connect to Wi-Fi  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL\_CONNECTED) {  
    delay(1000);  
    Serial.println("..");  
  }  
  Serial.println(WiFi.localIP());

   server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){  
    request->send\_P(200, "text/html", index\_html);  
  });  
   
  server.on("/forward", HTTP\_GET, [](AsyncWebServerRequest \*request){  
  digitalWrite(MA1,HIGH);  
  digitalWrite(MA2,LOW);  
  digitalWrite(MB1,HIGH);  
  digitalWrite(MB2,LOW);     
    request->send(200, "text/html", index\_html);  
  });  
  server.on("/backward", HTTP\_GET, [](AsyncWebServerRequest \*request){  
  digitalWrite(MA1,LOW);  
  digitalWrite(MA2,HIGH);  
  digitalWrite(MB1,LOW);  
  digitalWrite(MB2,HIGH);     
    request->send(200, "text/html", index\_html);  
  });  
   server.on("/stop", HTTP\_GET, [](AsyncWebServerRequest \*request){  
  digitalWrite(MA1,LOW);  
  digitalWrite(MA2,LOW);  
  digitalWrite(MB1,LOW);  
  digitalWrite(MB2,LOW);     
    request->send(200, "text/html", index\_html);  
  });  
   server.on("/right", HTTP\_GET, [](AsyncWebServerRequest \*request){  
  digitalWrite(MA1,HIGH);  
  digitalWrite(MA2,LOW);  
  digitalWrite(MB1,LOW);  
  digitalWrite(MB2,HIGH);     
    request->send(200, "text/html", index\_html);  
  });  
 server.on("/left", HTTP\_GET, [](AsyncWebServerRequest \*request){  
  digitalWrite(MA1,LOW);  
  digitalWrite(MA2,HIGH);  
  digitalWrite(MB1,HIGH);  
  digitalWrite(MB2,LOW);     
    request->send(200, "text/html", index\_html);  
  });  
    
  server.on("/distance", HTTP\_GET, [](AsyncWebServerRequest \*request){  
    request->send\_P(200, "text/plain", getDistance().c\_str());  
  });  
    
  // Start server  
  server.begin();  
}

* + 1. Relay Module

As shown in figure 10, relay module is being connected directly to digital circuits including microcontroller kits easily to control big loads by a microcontroller. The inputs IN1, IN2, IN3 and IN4 operate four relays with voltage between 3-5 volts DC. Input and output circuits are separated by Optocouplers to protect digital circuits in case connection mistakes happened or short circuits.

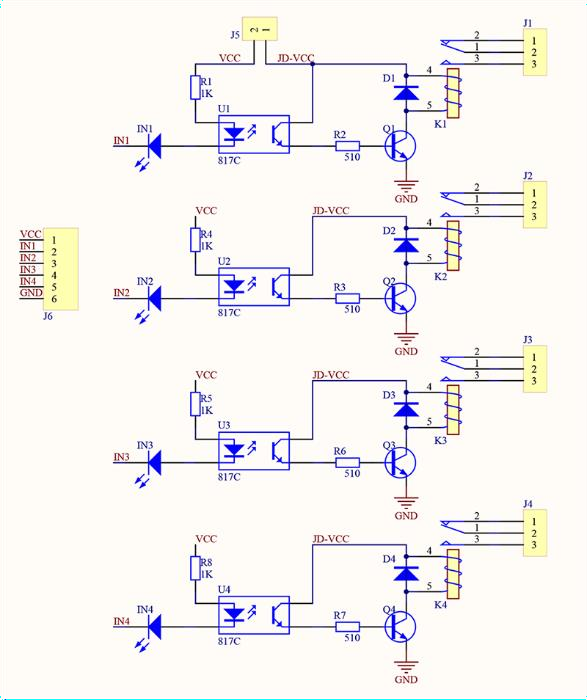


Figure 10. Relay Module Circuit Diagram

* + 1. LM35 Temperature Sensor

LM35 is an integrated circuit that works in range 4-20V DC as a temperature sensor with a precision at 10m V for one Celsius degree and low output impedance about 0.1 ohm at 1m A. LM35 is built in TO-92 package with three pins

1. Vcc
2. output
3. ground

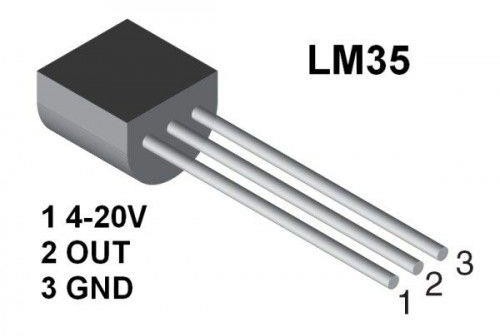


Figure 11. LM35 Pinout

# RESULTS AND DISCUSSION

* 1. **Light Control Test Results**

The Light Control Test is done by pressing the ON / OFF button widget on the Blynk application on the respective Android smart phone for lights and fans. This is done after the system is turned on and connected to a Wi-Fi internet connection. If at any time the internet connection is lost or bad signal, then it also affects system performance. Table 1 shows switches test results.

Table 1. Light Control Test

|  |  |  |
| --- | --- | --- |
| Switch status | On | Off |
|  | ---- | Relay 1  Relay 2  Relay 3  Relay 4 |
|  | Relay 1 | Relay 2  Relay 3  Relay 4 |
|  | Relay 2 | Relay 1  Relay 3  Relay 4 |
|  | Relay 3 | Relay 1  Relay 2 |
|  |  | Relay 4 |
|  | Relay 4 | Relay 1  Relay 2  Relay 3 |
|  | Relay 1  Relay 2  Relay 3  Relay 4 | ---- |

# LM35 Sensor Test

LM35 Sensor Testing is done by recording the temperature changes that occur every minute. This is done after the system is turned on and connected to a Wi-Fi internet connection. If at any time the internet connection is disconnected or bad signal, then it also affects system performance.

Table 2. Temperature Test

|  |  |
| --- | --- |
| Minute | Temperature |
| 1 | 32 |
| 2 | 33 |
| 3 | 33 |
| 4 | 34 |
| 5 | 33 |
| 6 | 32 |
| 7 | 33 |
| 8 | 34 |
| 9 | 33 |
| 10 | 33 |

# System Analysis

From testing the entire system above, the smart home works according to what is the purpose of this research. Comparison of this research with previous studies, namely this study uses temperature sensor and control buttons, thus increasing the diversity of the smart home system itself. Also, used a microcontroller that is different from previous studies that is the NodeMCU ESP8266 module which has advantages compared to other microcontrollers. The smart home has been successfully built with hardware arranged in such a way that it can achieve results that are as expected. In this case the hardware that plays a very important role as the main device is the NodeMCU ESP8266 module. The advantages of using the NodeMCU ESP8266 are more practical than buying various components and then assembling them by yourself.

# The Final Hardware Circuit Connection

Using components and materials mentioned above. Figure shows the project that’s used as an (IoT) system controlled by Blynk application is running. Loads used in this project are bulbs, they can be changed with other devices by changing bulbs with AC plugs to connect home-use devices or equipment.

**MICROCONTROLLER UNIT**

# MICROCONTROLLER AT89C51/89s52

# Features

• Compatible with MCS-51™ Products

• 8K Bytes of In-System Re programmable Flash Memory

• Endurance: 1,000 Write/Erase Cycles

• Fully Static Operation: 0 Hz to 24 MHz

• Three-level Program Memory Lock

• 256 x 8-bit Internal RAM

• 32 Programmable I/O Lines

•Three 16-bit Timer/Counters

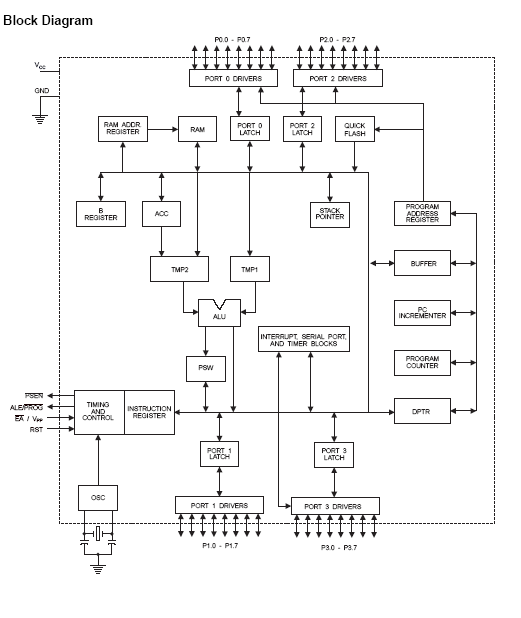
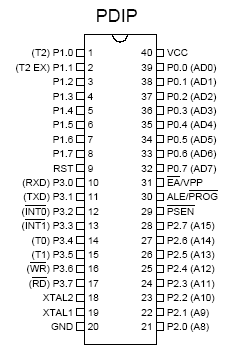
• Eight Interrupt Sources

• Programmable Serial Channel

• Low-power Idle and Power-down Modes

### DESCRIPTION

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer 8Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel ’s high-density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pin out.

The on-chip Flash allows the program memory to be reprogrammed in-system or by a Conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer that provides a highly flexible and cost-effective solution to many embedded control application.The Power-down mode saves the RAM contents but Freezes the oscillator, disabling all other chip functions until the next hardware reset

* 1. **Conclusion**

Based on the results of analysis of all data obtained by testing the smart home with the Internet of Things based NodeMCU ESP6288 module, the following conclusions can be drawn:

1. Smart Home with Internet of Things (IoT) based NodeMCU ESP8266 Module can be designed with various components hardware and software support so that it can be arranged into a smart home system that is controlled with the Blynk android application according to what is intended.
2. The Smart Home with this Internet of Things (IoT) based NodeMCU ESP8266 Module can be implemented to control some of the home electronics performance including lighting controls, fan control, temperature monitoring, early warning systems and etc.

# REFERENCES

1. Arduino Temperature Sensor Using LM35. Groups of Electronics Hobbyist, Roboticist. We Developed Electronics Project Tutorials Make Open for Everyone.
2. BOHORA, Bharat; MAHARJAN, Sunil; SHRESTHA, Bibek Raj. IoT Based Smart Home Using Blynk Framework. Zerone Scholar, [S.l.], v. 1, n. 1, p. 26-30, dec. 2016. ISSN 2542- 2774. google scholar.
3. DC-DC Step Down Converter Power Supply Provides Regulated 5VDC Output with Range Input of 10-32VDC, Model GTD21088L-1505-T2.
4. Home Automation Using Internet of Thing 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) Published: 2016. Google Scholar.
5. Internet of Things in Home Automation and Energy Efficient Smart Home Technologies Simon G. M. Koo Department of Computer Engineering, Santa Clara University, CA 95053, USA
6. Low Cost Implementation of Smart Home Automation Ravi Kishore Kodali Department of Electronics and Communication Engineering National Institute of Technology, Warangal , 506004 India
7. [Mobile based home automation using Internet of Things (IoT)](https://ieeexplore.ieee.org/document/7475301/) 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) Published: 2015
8. NodeMCU Features and Pinout. A Brief Tutorial on the Introduction to NodeMCU V3.
9. Yoyosteven in Circuits Microcontrollers. NODEMCU 1.0 (ESP8266) CONTROLLED RELAY USING BLYNK (OVER THE WEB).
10. 5V 4-Channel Relay Interface Board, Standard Interface that can be Controlled Directly by Microcontroller.
11. 15-17 March 2018 U. Venkanna IoT Based Smart Home Automation System Using Sensor Node. Google Scholar.