

# **HOME AUTOMATION USING ARDUINO WIFI MODULE ESP8266**

**A PROJECT REPORT**

*Submitted by*

**ILYAS BAIG  
CHIKTAY MUZAMIL  
SALAHUDDIN DALVI**

*in partial fulfillment for the award of the degree of*

**B.E**

**IN**

**ELECTRONICS & TELECOMMUNICATION**

**At**



**ANJUMAN-I-ISLAM'S  
KALSEKAR TECHNICAL CAMPUS  
PANVEL  
2015-2016**

## Project Report Approval for B.E

This project report entitled **HOME AUTOMATI ON USI NG ARDUI NO W FI MODULE ESP8266** by **I l yas Bai g Chi kt ay Muzami l Sal ahuddi n Dal vi** is approved for the degree of **Bachelor in Engineering**.

### Examiners:

1. \_\_\_\_\_.

2. \_\_\_\_\_.

Supervisor(s):

\_\_\_\_\_  
**Asst. Prof. BANDANAWAZ M. KOTIYAL**

H.O.D (EXTC):

\_\_\_\_\_  
**Asst. Prof. MUJIB A. TAMBOLI**

Date:

Place

---

## DECLARATION

We hereby declare that the project entitled "**HOME AUTOMATION USING ARDUINO WIFI MODULE ESP8266**" submitted for the **B. E.** Degree is Our original work and the project has not formed the basis for the award of any degree, association, fellowship or any other similar titles.

Signature of the Student

Ilyas Baig

Chikhtay Muzamil

Sal ahuddi n Dal vi

Pl ace: New Panvel

### **ACKNOWLEDGEMENT**

Before we get into thick of things I would like to add few heartfelt words for the people who are part of our team as they have been unending contribution right from the start of construction of the report.

Apart from the team I am indebted to the numbers of persons who have provided helpful and constructive guidance in the draft of material.

I acknowledge with deep sense of gratitude towards the encouragement In the form of substantial assistance provided each and every member of my team

I would like to extend my sincere thanks to our guide

**Asst . Prof . Banda Nawaz**

For providing us the required technical guidance in the process of preparing this report.

## **ABSTRACT**

This project 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 consists of two main components; the first part is the server (web server), which presents system core that manages, controls, and monitors users' home.

Users and system administrator can locally (LAN) or remotely (internet) manage and control system code. Second part is hardware interface module, which provides appropriate interface to sensors and actuator of home automation system

Unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface modules

as long as it exists on WiFi network coverage. System supports a wide range of home automation devices like power management components, and security components.

The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

## TABLE OF CONTENTS

### Cont ent s

APPROVAL.....	1
DECLARATI ON.....	2
ACKNOWLEDGEMENT .....	3
ABSTRACT.....	4
CHAPTER NO. 01.....	
I NTRODUCTI ON.....	6
CHAPTER NO. 02.....	
OBJECTI VE OF THE PROJECT.....	8
CHAPTER NO. 03.....	
LI TERATURE SURVEY.....	10
CHAPTER NO. 04.....	
SCOPE OF THE PROJECT.....	13
CHAPTER NO. 05.....	
METHODOLOGY .....	15
CHAPTER NO. 06.....	
HARDWARE.....	17
CHAPTER NO. 07.....	
SOFTWARE.....	34
CHAPTER NO. 08.....	
REFERENCES & BI BLI OGRAPHY.....	46

## **CHAPTER NO. 01**

### **I NTRODUCTI ON**



## INTRODUCTION

The project aims at designing an advanced home automation system using normal web server and W-Fi technology. The devices can be switched ON/OFF and sensors can be read using a Personal Computer (PC) through W-Fi.

Automation is the most frequently spelled term in the field of electronics. The hunger for automation brought many revolutions in the existing technologies. These had greater importance than any other technologies due to its user-friendly nature. These can be used as a replacement of the existing switches in home which produces sparks and also results in fire accidents in few situations. Considering the advantages of W-Fi an advanced automation system was developed to control the appliances in the house.

W-Fi (Short for Wireless Fidelity) is a wireless technology that uses radio frequency to transmit data through the air. W-Fi has initial speeds of 1mbps to 2mbps. W-Fi transmits data in the frequency band of 2.4 GHz. It implements the concept of frequency division multiplexing technology. Range of W-Fi technology is 40-300 feet.

The controlling device for the automation in the project is a Arduino UNO. The data sent from PC over W-Fi will be received by W-Fi module connected to

Ardui no UNQ. Ardui no UNO reads the data and decides the switching action of electrical devices connected to it through Relays.

## **CHAPTER NO. 02**

### **OBJECTIVE OF PROJECT**

- The goal of this project is to develop a home automation system that gives the user complete control over all remotely controllable aspects of his or her home.

- The automation system will have the ability to be controlled from a central host PC, the Internet, and also remotely accessed via a Pocket PC with a Windows Mobile based application.
- The System will also sense the Accidental Gas Leakage , water level and will notify the user by SMS.

## **CHAPTER NO. 03**

### **LITERATURE SURVEY**

#### **Literature survey:**

#### **Review of Related Literature:**

When people think about home automation, most of them may imagine living in a smart home: One remote controller for every household appliance, cooking

the rice 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.

### **Review of Foreign Studies:**

In their paper, Tan, Lee and Soh (2002) proposed the development of an Internet-based system to allow monitoring of important process variables from a distributed control system (DCS). This paper proposes hardware and software design considerations which enable the user to access the process variables on the DCS, remotely and effectively

Potamitis, Georgila, Fakotakis, and Kokkinakis, G (2003) suggested the use of speech to interact remotely with the home appliances to perform a particular action on behalf of the user. The approach is inclined for people with disability to perform real-life operations at home by directing appliances through speech. Voice separation strategy is selected to take appropriate decision by speech recognition

In the year 2006, S. M. Anamul Haque, S. M. Kamruzzaman and Md. Ashraf ul Islam proposed a system entitled "A System for Smart-Home Control of Appliances Based on Time and Speech Interaction" that controls the home appliances using the personal computer. This system is developed by using the Visual Basic 6.0 as programming language and Microsoft voice engine tools for speech recognition purpose. Appliances can be either controlled by timer or by voice command.

Giubotaru-Petrescu, Chiciudean, Gheorga, and Stănescu (2006) present a design and implementation of SMS based control for monitoring systems. The paper has three modules involving sensing unit for monitoring the complex applications. A processing unit, that is microcontroller and a communication module that uses GPRS modem or cell phone via serial port RS-232. The SMS is used for status reporting such as power failure.

Jawarkar, Ahmed, Ladhake, and Thakare (2008) propose remote monitoring through mobile phone involving the use of spoken commands. The spoken commands are generated and sent in the form of text SMS to the control system and then the microcontroller on the basis of SMS takes a decision of a particular task.

Prof. Era Johri Dept. Of Information And Technology K.J.Somaiya College Of Engineering VI DYAVI HAR, MUMBAI “ Remote Controlled Home Automation Using Android Application via WiFi Connectivity” .

**CHAPTER NO. 04**  
**SCOPE OF PROJECT**



Day by day, the field of automation is blooming and these systems are having great impact on human beings. The project which is to be implemented is a home automation using Easy IOT Webserver and WIFI and has very good future development.

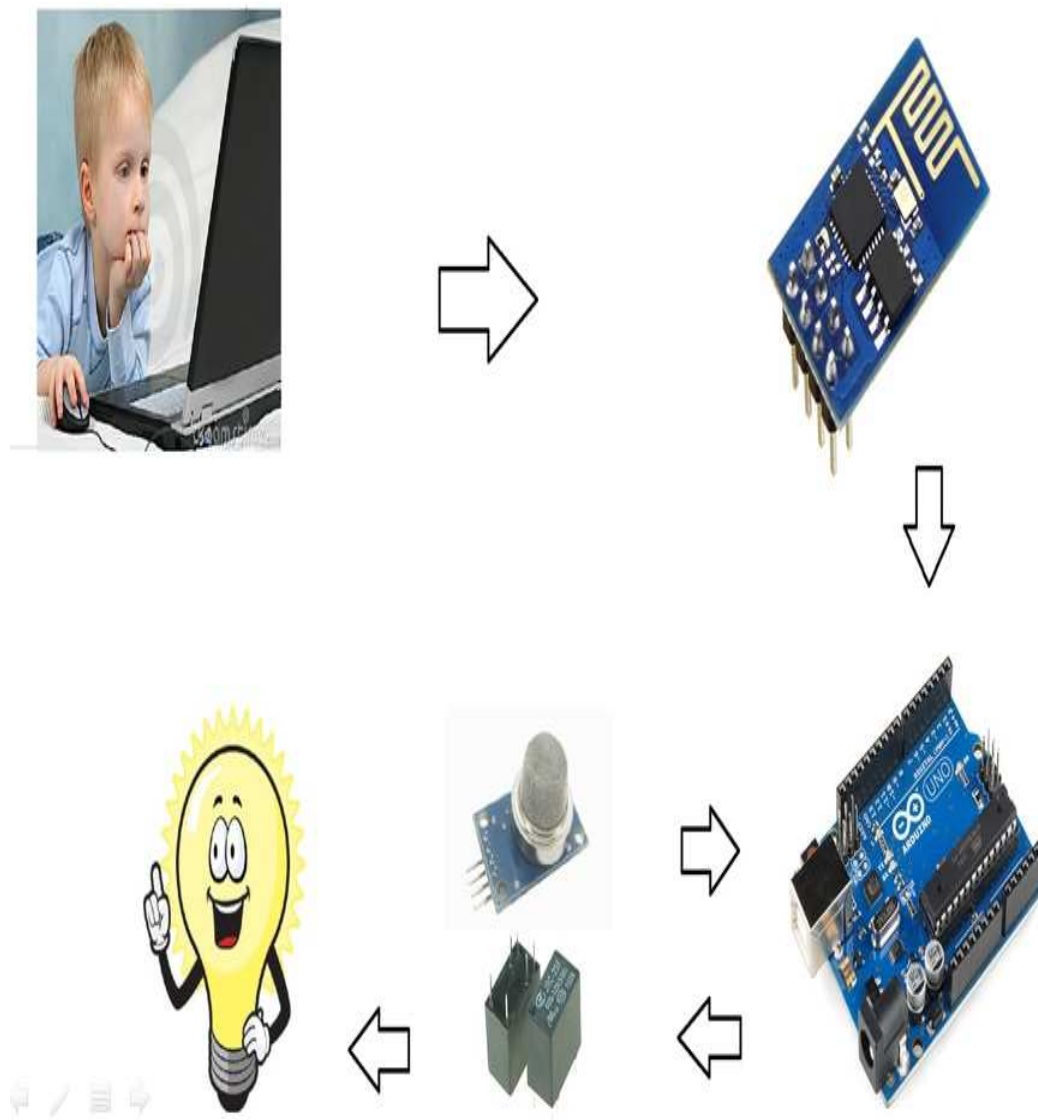
In the current system webserver is installed on a windows PC so the home appliances can be controlled using only by using the device on which webserver is installed. This can be further developed installing webserver on cloud.

Advantage of installing webserver on the cloud is that home can be controlled by using any device which has WIFI 802.11 and a web browser. By visiting the IP address of the cloud the control actions can be taken.



## CHAPTER NO. 05

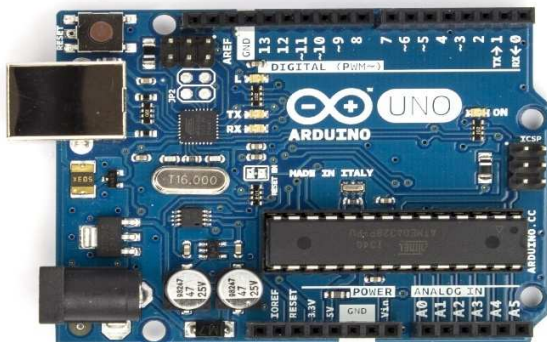
### METHODOLOGY



## **CHAPTER NO. 06**

### **HARDWARE**

**Ar dui no: -**



The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

The Uno is a microcontroller board based on the [ATmega328P](#). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started..

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards.

### **Technical specifications:-**

Microcontroller    ATmega328P

Operating Voltage    5V

Input Voltage (recommended)    7-12V

Input Voltage (limit)    6-20V

Digital I/O Pins    14 (of which 6 provide PWM output)

PWM Digital I/O Pins    6

Analog Input Pins    6

DC Current per I/O Pin    20 mA

DC Current for 3.3V Pin    50 mA



```

// and the RX line from the esp to the Arduino's pin
3

void setup()

{

  Serial.begin(9600);

  esp8266.begin(9600); // your esp's baud rate might be different


  pinMode(4, OUTPUT);

  digitalWrite(4, LOW);


  pinMode(5, OUTPUT);

  digitalWrite(5, LOW);


  pinMode(6, OUTPUT);

  digitalWrite(6, LOW);


  pinMode(7, OUTPUT);

  digitalWrite(7, LOW);

```



```

pinMode( 13, OUTPUT);

digitalWrite( 13, LOW);

sendData(" AT+RST\r\n", 2000, DEBUG); // reset module

sendData(" AT+CWMODE=2\r\n", 1000, DEBUG); // configure as access point

sendData(" AT+CIFSR\r\n", 1000, DEBUG); // get ip address

sendData(" AT+CWMUX=1\r\n", 1000, DEBUG); // configure for multiple
connections

sendData(" AT+CIPSERVER=1, 80\r\n", 1000, DEBUG); // turn on server on port 80
}

void loop()

{

    if(esp8266.available()) // check if the esp is sending a message

    {

        if(esp8266.find("+IPD,"))

        {

            delay(1000); // wait for the serial buffer to fill up (read all the
serial data)

            // get the connection id so that we can then disconnect

```

```
int connectid = esp8266.read()-48; // subtract 48 because the read()
function returns
```

```
                // the ASCII decimal value and 0
(the first decimal number) starts at 48
```

```
esp8266.find("pin="); // advance cursor to "pin="
```

```
int pinNumber = (esp8266.read()-48)*10; // get first number i.e. if the
pin is 13 then the 1st number is 1, then multiply to get 10
```

```
pinNumber += (esp8266.read()-48); // get second number, i.e. if the pin
number is 13 then the 2nd number is 3, then add to the first number
```

```
switch (pinNumber)
```

```
{
```

```
case 1://switch 1 on
```

```
    digitalWrite(4, HIGH);
```

```
    break;
```

```
case 2://switch 2 on
```

```
    digitalWrite(5, HIGH);
```

```
        break;

    case 3:// switch 3 on

        digitalWrite(6, HIGH);

        break;

    case 4:// switch 4 on

        digitalWrite(7, HIGH);

        break;

    case 5:// led on

        digitalWrite(13, HIGH);

        break;

    case 6:// switch 1 off

        digitalWrite(4, LOW);

        break;

    case 7:// switch 2 off
```

```

        di gi t al W i t e( 5, LOW;

        br eak;

    case 8:// swi t ch 3 of f

        di gi t al W i t e( 6, LOW;

        br eak;

    case 9:// swi t ch 4 of f

        di gi t al W i t e( 7, LOW;

        br eak;

    case 10:// l ed of f

        di gi t al W i t e( 13, LOW;

        br eak;

    def aul t :

        br eak;

}

// di gi t al W i t e( pi nNumber , ! di gi t al R ead( pi nNumber ) ) ; // t oggl e pi n

```

```

        // make close command

        String closeCommand = "AT+CLOSE=";

        closeCommand+=connectionId; // append connection id

        closeCommand+="\r\n";

        sendData(closeCommand, 1000, DEBUG); // close connection
    }

}

}

/*
 * Name: sendData
 * Description: Function used to send data to ESP8266.
 * Params: command - the data/command to send; timeout - the time to wait for a
response; debug - print to Serial window?(true = yes, false = no)
 * Returns: The response from the esp8266 (if there is a response)
 */

String sendData(String command, const int timeout, boolean debug)
{

```

```

String response = "";

esp8266.print(command); // send the read character to the esp8266

long int time = millis();

while( (time+timeout) > millis())

{

    while(esp8266.available())

    {

        if(debug)

        {

            Serial.print(response);

            return response;

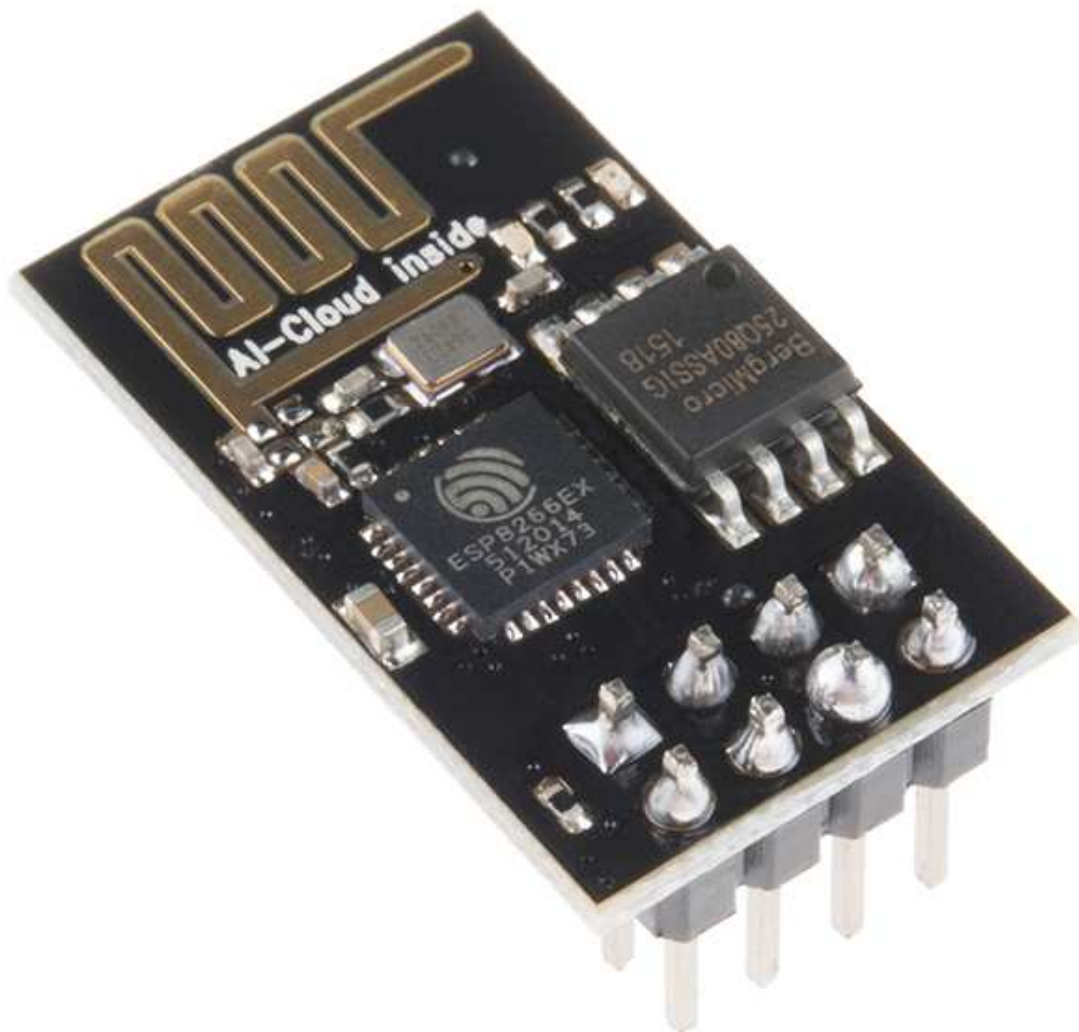
        }

    }

}

```

**Esp 8266: -**



**Description:** The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all W-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.



This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the *Documents* section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

### **Features:**

- 802.11 b/g/n
- W-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory

- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM8)

#### **Specification of ESP 8266:**

- W-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- 19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor

- Standby power consumption of  $< 1.0\text{mW}$  (DTI MB)

### Relay board: -



A relay is an electrical device which is generally used to control high voltages using very low voltage as an

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

This is Four Channel relay board controlled by computer USB port. The usb relay board is with 4 SPDT relays rated up to 10A each. You may control devices 220V / 120V (up to 4) directly with one such relay unit. It is fully powered by the computer USB port. Suitable for home automation applications, hobby projects, industrial automation. The free software allows to control relays manually, create timers (weekly and calendar) and multivibrators, use date and time for alarms or control from command line. We provide software examples in Labview, .NET, Java, Borland C++, Python

#### **Features: -**

- Datasheet - [here](#)
- Power led: Yes
- Relay leds: Yes High quality
- 4 SPDT Relay channels - selectable by user:

- JQC-3FC/ T73 DC5V (7A / 250VAC, 10A / 125VAC, 12A / 120VAC, 10A / 28VDC)
- [RAS-05-15](#) (10A / 250VAC, 15A / 120VAC, 15A / 24VDC)
- PCB parameters: FR4 / 1.5mm / two layers / metalized holes / HAL / white stamp / solder mask / extra PCB openings for better voltage isolation / doubled high voltage tracks
- Power supply: from USB port
- Current consumption: 400 mA
- Chip: FT245RL
- Size: 77mm x 56mm x 17mm
- Supported by [DFM software \(Windows and Linux\)](#): Yes
- Supported by [Denkovi Command line tool](#) (Windows, Linux): Yes
- Android software available (low cost but very useful): Yes - **New**
- Software examples - [here](#)
- Documentation: [here](#)

#### **Advantages: -**

- High quality
- Low cost
- No extra power supply
- Software with many functions
- Control electrical devices according weekday/date/time
- Create timers or pulses with our software

### **Appl i cat i ons: -**

- Home automati on
- Robot i cs
- Al ar ms
- Ti mer s
- Open doors and wi ndows vi a PC
- Aquari ums appl i cat i ons

### **Addi ti onal i nf or mat i on: -**

This is relay board with 4 SPDT Relays controlled from USB port of your computer. The main purpose of this USB relay module is to help you building projects regarding robotics and home automation (domotic). You may control differenet electrical devices like home lights, DC motors, pneumatic cylinders, lasers and so on. Each such board requires one USB port. The more USB ports you have the more such relay units you may connect and control. . The relay module outputs are controlled by FT245RL. It has 8 bit data output register (this device use only 4 of them). The usb relay card can not be controlled directly via COM port - you need to download our DRM Software to control the device. The usb relay unit can not work without PC. Only one such device can be supplied from single USB port. If you want to supply many such devices you need USB HUB with extra power supply.

### **USB TO TTL CONVERTER: -**



This USB to TTL converter combine the USB-232-1 (USB to Single RS232 Adapter) and TTL-232-1 (Port-powered RS232/TTL converter) allows you to convert USB to TTL/CMOS compatible levels and vice versa.

It can be used to set up [APC220 Radio Data Module \(SKU: TEL0005\)](#) wireless module.

It can be used as STC microcontroller program downloader.

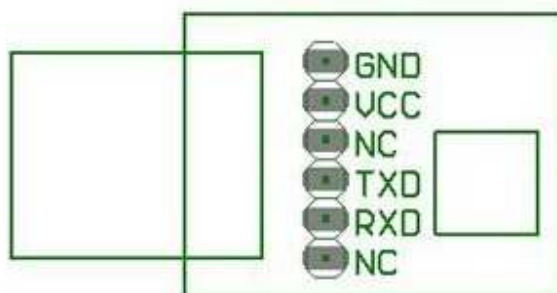
### Specification

---

- Voltage: 5V
- Chip: Silicon Laboratories CP210x chip

### Pin Definition:

---

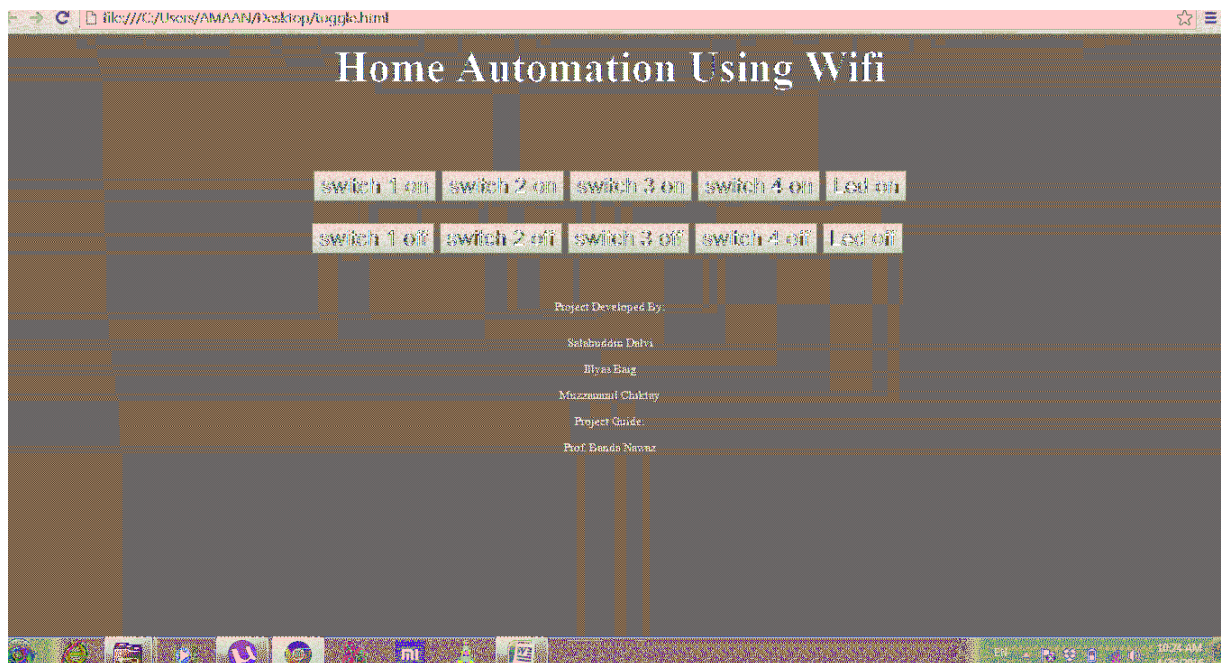


**CHAPTER NO. 07**

**SOFTWARE**



**Webserver :-**



HTML CODE:

```
<html>
```

```
  <head>
```

```
    <title>Home automation Using Wifi</title>
```

```
  </head>
```

```
  <body align="center" bgcolor="grey">
```

<p><b><font size=15>Home Automation Using Wfi </font></b></p>

<p> </p>

<br><br><br><br>

<!-- in the <button> tags below the ID attribute is the value sent to the arduino -->

<button id="01" class="switch"><font size=5>switch 1 on</button> <!-- button for pin 9 -->

<button id="02" class="switch"><font size=5>switch 2 on</button> <!-- button for pin 9 -->

<button id="03" class="switch"><font size=5>switch 3 on</button> <!-- button for pin 10 -->

<button id="04" class="switch"><font size=5>switch 4 on</button> <!-- button for pin 9 -->

<button id="05" class="switch"><font size=5>Led on</button> <!-- button for pin 11 -->

<p>

<button id="06" class="switch"><font size=5>switch 1 off</button> <!-- button for pin 9 -->

<button id="07" class="switch"><font size=5>switch 2 off</button> <!-- button for pin 12 -->

```
<button id="08" class="switch"><font size=5>switch 3 off</button> <!-- button  
for pin 9 -->
```

```
<button id="09" class="switch"><font size=5>switch 4 off</button> <!--  
button for pin 13 -->
```

```
<button id="10" class="switch"><font size=5>Led off</button> <!-- button for  
pin 9 -->
```

```
<br><br>
```

```
<font style="italic" face="calabri" color="white" size=2>
```

```
<p> Project Developed By: </p>
```

```
<p> Sal ahuddi n Dal vi </p>
```

```
<p> Ill yas Bai g</p>
```

```
<p> Muzzammi l Chi kt ay </p>
```

```
<p>Project Gui de: </p>
```

```
<p>Prof . Banda Nawaz</p>
```

```
</font>
```

```
<script src="jquery.min.js"></script>
```

```
<script type="text/javascript">
```

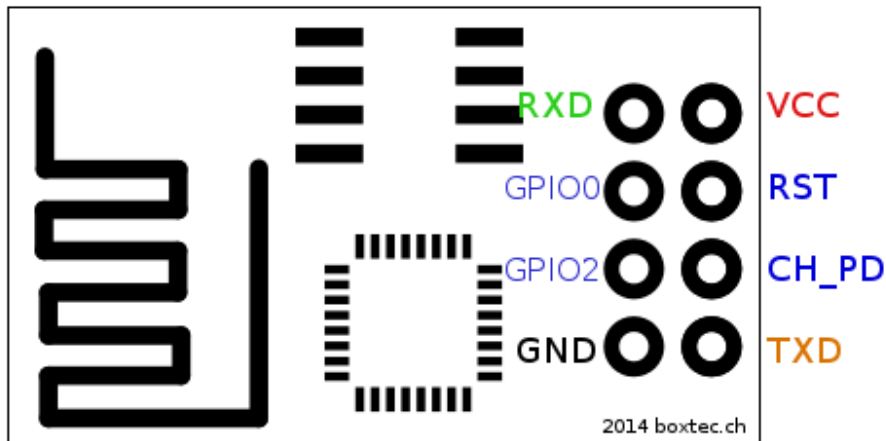
```
$(document).ready(function(){
```

```
$(".switch").click(function(){
```

## Esp 8266 firmware:-

### Required hardware and software:

- You will need a Windows PC for this update
- You will need some form of USB to Serial converter that allows operation at 3.3V. I used a [Foca](#) board. It allows easy plugging into a breadboard, which then allows me to hookup the pins of the ESP8266 module via jumper wires to the corresponding pins on the USB<->Serial board.
- The firmware updating software only works on COM ports 1-6. If your USB<->Serial device enumerates to a higher port number than that, you will have to change it via *Device Manager* in Windows



Above is a pin out diagram for the ESP8266 Module

You need to hookup these pins from the ESP8266 to your USB<->Serial board:

- VCC to 3.3V
- GND to ground
- CH\_PD to 3.3V
- TXD to RX, RXD to TX (*this may depend on the USB<->Serial board you are using. If it doesn't work, try swapping them around*)
- GPIO0 to ground (*for the duration of firmware upgrading. After all the upgrades have been loaded, it needs to be disconnected*)

*You will need to unplug and re-plug the USB cable 4 times during the process, so make sure you can reach it easily*

### Flashing the firmware

Inside the `esp_flasher.zip` file, you will find a couple of `.bin` files, an executable named `XTCOM_UTIL.exe` and also a `readme.txt` file containing the addresses to flash each of the `.bin` files to, which I will duplicate here for reference

```
boot_v1.1.bin----->0x00000
```

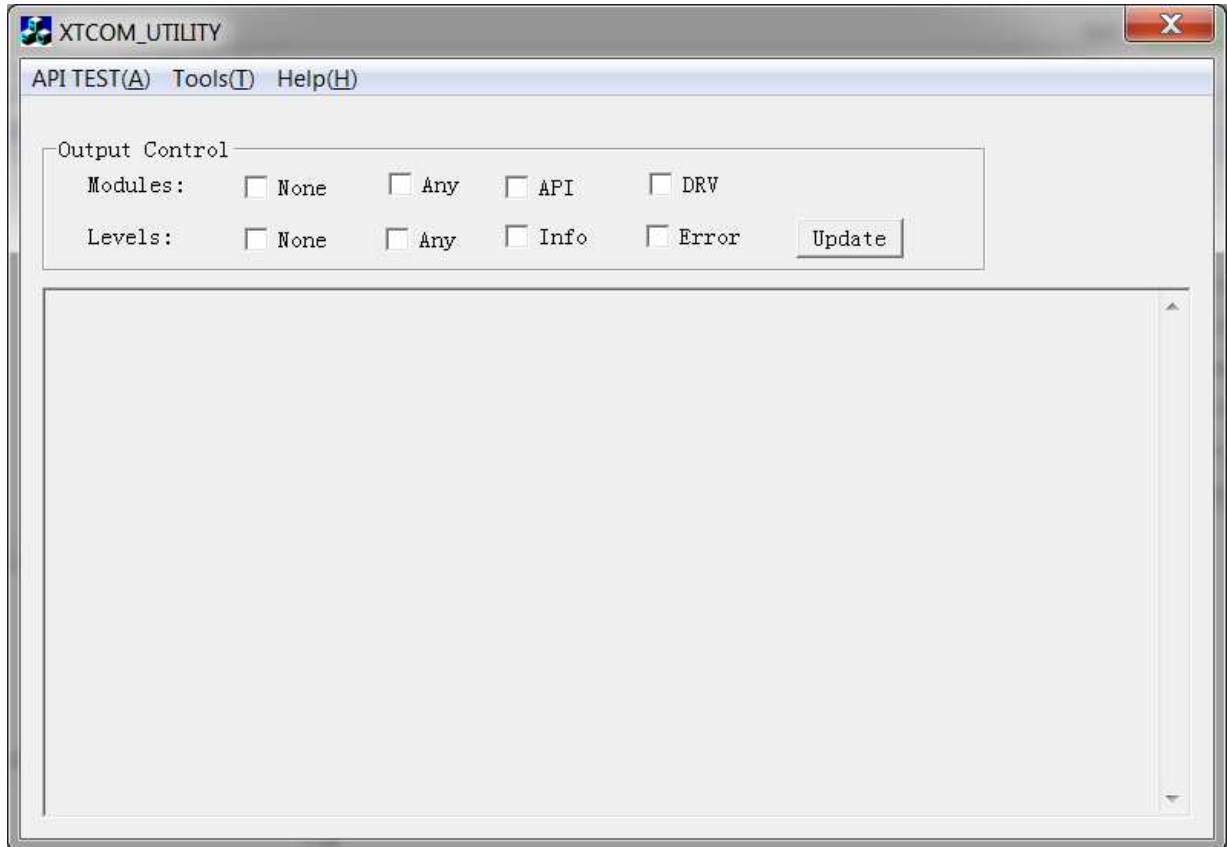
```
user1.bin----->0x01000
```

```
esp_init_data_default.bin---->0x70000
```

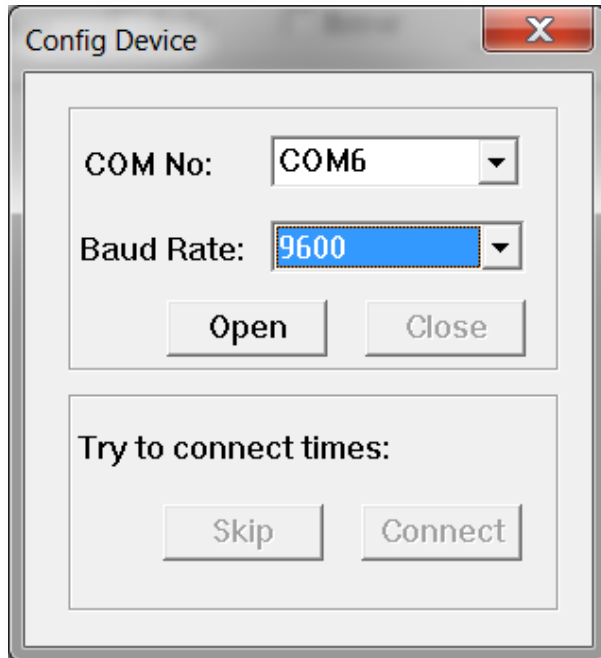
```
blank.bin----->0x7E000
```

Steps to flash each .bin file

1. Power everything by plugging into USB port . 2. Start XTCOM\_UTIL.exe

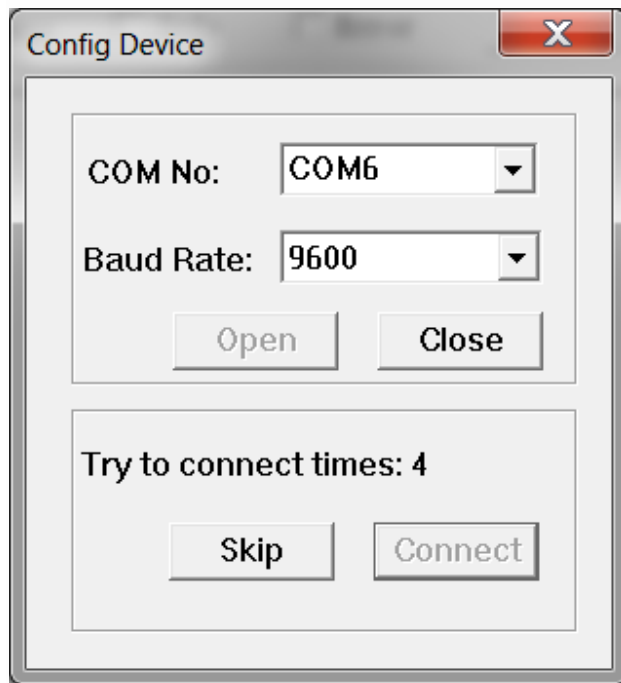


3. Hit *Tools* -> *Config Device* in the menu



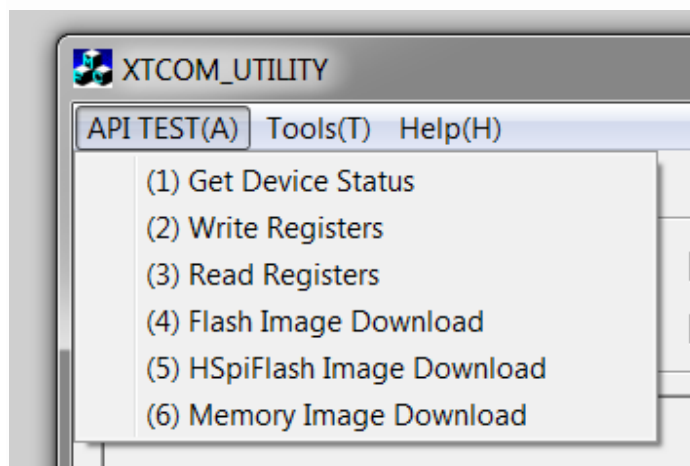
2. Configure the settings to whatever COM port you are using, and 9600 baud rate
3. Click on "Open". You should receive a notification saying "Operation Succeeded!"
4. Click on "Connect". The "Try to connect times: " message should increment a few times as the utility tries to contact the module.



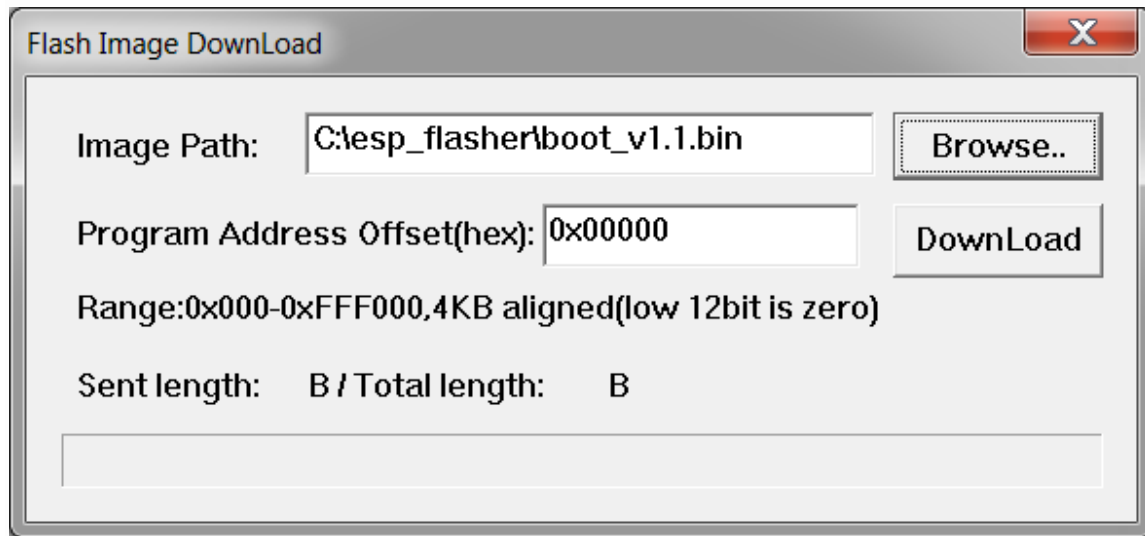


You should receive a notification about success before hitting 20 times. **If you don't, then make sure your connections are correct, and perhaps try swapping the RX and TX pins around.**

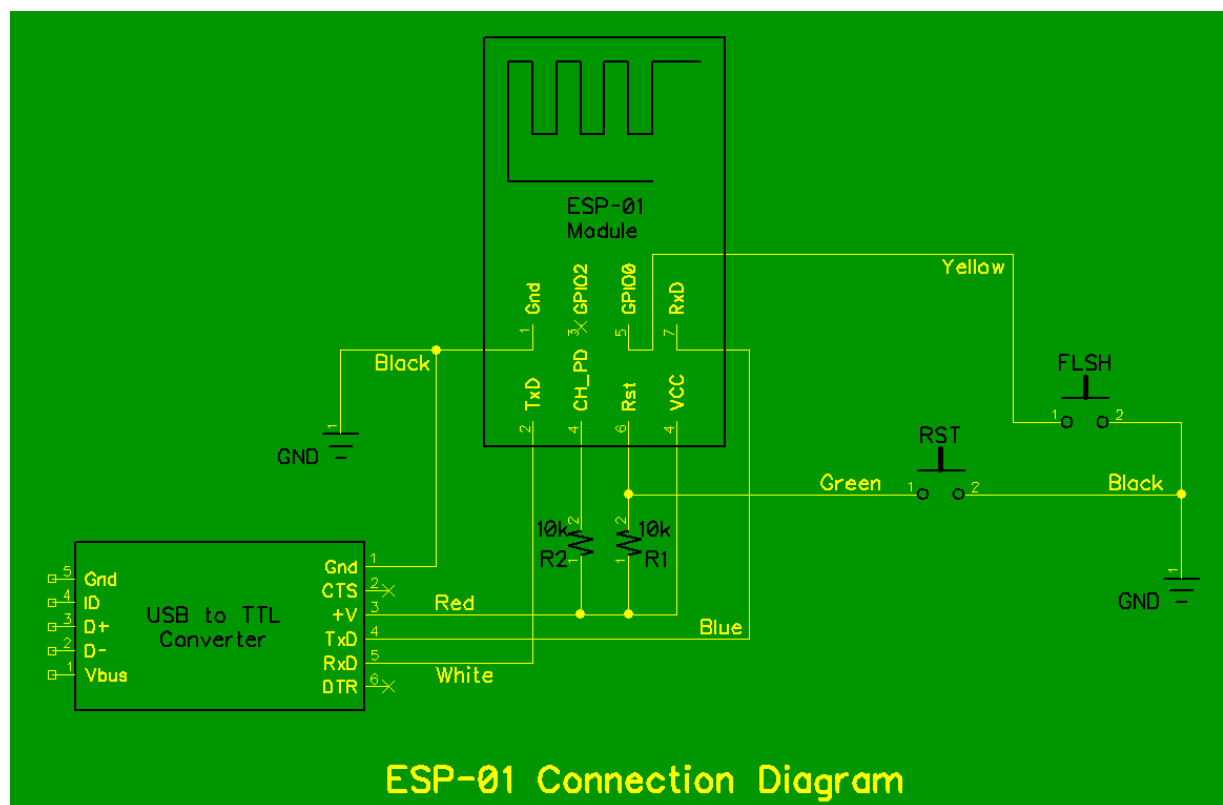
5. Close the "Config Device" window.
6. Hit *API TEST -> (4) Flash Image Download* in the menu



7. Select the .bin file to flash, and configure the correct destination address as in the readme.txt file (For Example: boot\_v1.1.bin @0x00000)



8. Click on "DownLoad" and wait for the flashing to complete.
9. Close down the entire XTCOM\_UTIL program and unplug the USB. \* VERY IMPORTANT This must be done between each .bin file
10. Repeat each of the above steps for each of the .bin files, making sure to leave the GPIO pin grounded during the entire process.



**CHAPTER NO. 08**

**REFERENCE AND BI BLI OGRAPHY**

## **References & Bibliography: -**

### **Websites:**

- <http://www.iot-playground.com>
- <http://www.instructables.com>
- <http://en.wikipedia.org>
- <http://www.journals.ei.sevier.com/easyiot>

### **Journals & other books:**

- 1 .Kusuma S M Assistant Professor, Department of telecommunication, MSRI T, Bangalore, India. “ Home Automation Using Internet of Things.”
- 2.Nharika Shrotriya, Anjali Kul Karni, Priti Gadhave, International Journal of Science, Engineering and Technology Research (IJSETR), “ SMART HOME USING W - FI ”
- Anushri Aware, Sonali Vaidya, Priyanka Ashture, Varsha Gaiwal PES' s Modern College of Engineering, Pune- 04, International Journal of Engineering Research and General Science Volume 3, “ Home Automation using Cloud Network”



