**ARDUINO HOME AUTOMATION**

**(WIFI CONTROLLED SWITCHING OF AC MAINS APPLIANCES)**

A REPORT

Submitted to M S Ramaiah Institute of Technology, Bangalore

inpartial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING inELECTRICAL and ELECTRONICS ENGINEERING.

FAISAL SIDDIQUI (1MS13EE019)

under the guidance of

**Smt Archana Diwakar & Sri. Ramakrishna Murthy**

MSRIT



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

M.S.RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

May 2016

M.S. RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE-54

(Autonomous Institute, Affiliated to VTU)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING



DECLARATION

We hereby declare that the entire work embodied in this report has been carried out by us at M.S. Ramaiah Institute of Technology under thesupervision of Dr. PremilaManohar, Head of the Department, EEE, MSRIT.This report has not been submitted in part or full, for the award of any diplomaor degree of this or any other University.

Place: MSRIT,BANGALORE FAISAL SIDDIQUI(1MS13EE019)

Date: 22ND APRIL, 2016

**ABSTRACT**

Home automation becomes important, because it gives the user the comfort and control of all the home devices. The implementation and design of wireless home automation control used two methods, WLAN technology and RF remote control handheld to control of the selective home devices with integral security and protected system. The devices that have been distributed throughout the residence have their own boards; these boards are connected to the wireless local area network through w5100 wiznet wifi shields available for arduino. The software consist of Java script as well as arduino programming languages for programming onboard arduino uno and developing interfacing/communication systems that are used to communicate between electronic devices (phones ,tablets etc) and onboard device boards, also Java script is used to design Graphical User Interface (GUI) which involves all the control instructions of appliances available for display in browser webpages . The system is low cost and flexible with the increasing variety of devices to be controlled and control schemes to be implemented.

**INTRODUCTION**

The home automation is control of home devices from a central control point, automation is today’s fact, where more things are being completed every day automatically, usually the basic tasks of turning on or off certain devices and beyond, either remotely or in close proximity . The popularity of wireless networks in home has increased in recent years, and the advanced computer technology has made the personal digital device to commonly have the capability to communicate through the wireless network. One of the possible applications are wireless networks for home automation. Imagine a private home equipped with motion, light, temperature and other sensors and actuators for opening the door, dimming the light, controlling the heating and so on . It can be as basic as dimming lights with a remote control or as complex as setting up a network of items in your home (such as a thermostat, security system, lighting and appliances) that can be programmed using a main controller [3]. The basic idea of home automation is to employ sensors and control systems to monitor a dwelling, and accordingly adjust the various mechanisms that provide heat, ventilation, lighting, and other services. The automated “intelligent” home can provide a safer, more comfortable, and more economical dwelling. In an intelligent home automation system there are many possible solutions for how and from where to control the automation system and single devices; a user interface can be a computer-based system, a mechanical switch, a single light, a loudspeaker with a microphone or a some kind of personal remote controller for all the home appliances, the home appliances can be controlled using normal PC, laptop or table PC by standalone software or **web-based user interface**. In the near future all electronic appliances in a home will be networked: PCs, telephones, stereos, refrigerators and even washing machines, heating and air conditioning, previously controlled by a single, fixed, manual thermostat, can now be managed by an intelligent controller with remote-access capabilities.

In this report, a home automation system has been designed by choosing a prototype appliance such as a 40watt incandescent bulb. Switching of this appliance as well as controlling the amount of power supplied to it (Intensity of light emitted by the bulb or brightness) will be controlled by the user through a remote webpage accessible from any smart device such as phones, personal computers, tablets using phase control or phase cutting of the power supplied to the appliance from 220v,50hz ac mains grid .

**SURVEY OF LITERATURE**

Switching an AC load with an Arduino uno and remotely controlled webpage is rather simple( either a mechanical relay or a solid state relay with an optically isolated Triac). It becomes a bit more tricky if one wants to control the power through a mains AC lamp with an arduino and webpage for user access/interface.

By just limiting the current through e.g. a transistor is not really possible due to the large power the transistor then will need to dissipate, resulting in much heat and it is also not efficient from an energy use point of view.

After successfully completing the survey of literature resources available, I have found that there are several ways of controlling voltage and in turn current through an ac appliance. Some of the methods are discussed below:

**1.Phase cutting**  
One way of doing this, is through phase control with a Triac: the Triac then is fully opened, but only during a part of the sinus AC wave. This is called leading edge cutting.  
One could let an Arduino just open the Triac for a number of microseconds, but that has the problem that it is unpredictable during what part of the sinus wave the triac opens and therefore the dimming level is unpredictable. One needs a reference point in the sinus wave.  
For that a zero crossing detector is necessary. This is a circuit that tells the Arduino (or another micro controller) when the sine wave goes through zero and therefore gives a defined point on that sinus wave.  
Opening the Triac after a number of microseconds delay starting from the zero crossing therefore gives a predictable level of dimming.

**2. Pulse Skip Modulation**  
Another way of doing this is by Pulse Skip Modulation. With PSM, one or more full cycles (sinuswaves) are transferred to the load and then one or more cycles are not. Though effective, it is not a good way to dim a light/bulb as there is chance for flickering. Though it might be tempting, in PSM one should always allow a full sinuswave to be passed to the load, not a half sinus as in that case the load will be fed factually from DC which is not a good thing for most AC loads. The difference between leading edge cutting and PSM is mainly in the software: in both cases one will need a circuit that detects the zero crossing and that can control a triac.  
  
A circuit that can do this is easy to build: The zero crossing is directly derived from the rectified mains AC lines – via an optocoupler of course- and gives a signal every time the wave goes through zero. Because the sine wave first goes through double phased rectification, the zero-crossing signal is given regardless whether the sinus wave goes up through zero or down through zero. This signal then can be used to trigger an interrupt in the Arduino.

**3. PWM dimming**  
It goes without saying that there needs to be a galvanic separation between the Arduino side of things and anything connected to the mains.   
The mains 220Volt voltage is led through two 30k resistors to a bridge rectifier that gives a double phased rectified signal to a 4N25 opto-coupler. The LED in this opto-coupler thus goes low with a frequency of 100Hz and the signal on the collector is going high with a frequency of 100Hz, in line with the sinusoid wave on the mains net. The signal of the 4N25 is fed to an interrupt pin in the Arduino (or other microprocessor). The interrupt routine feeds a signal of a specific length to one of the I/O pins. The I/O pin signal goes back to our circuit and opens the LED and a MOC3021, that triggers the Opto-Thyristor briefly. The LED in series with the MOC3021 indicates if there is any current going through the MOC3021.

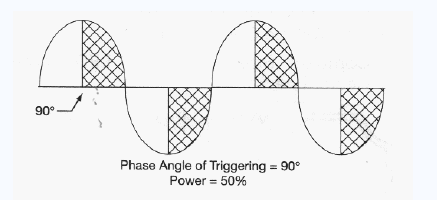
**Theory and Calculations in reference to this project**

Phase control of AC waves:

Phase-fired control is often used to control the amount of voltage, current or power that a power supply feeds to its load. It does this in much the same way that a pulse-width modulated (PWM) supply would pulse on and off to create an average value at its output. If the supply has a DC output, its time base is of no importance in deciding when to pulse the supply on or off, as the value that will be pulsed on and off is continuous.

PFC differs from PWM in that it addresses supplies that output a modulated waveform, such as the sinusoidal [AC](https://en.wikipedia.org/wiki/Alternating_current) waveform that the national grid outputs. Here, it becomes important for the supply to pulse on and off at the correct position in the modulation cycle for a known value to be achieved; for example, the controller could turn on at the peak of a waveform or at its base if the cycle's time base were not taken into consideration.

Phase-fired controllers take their name from that fact that they trigger a pulse of output at a certain phase of the input's modulation cycle. In essence, a PFC is a PWM controller that can synchronise itself with the modulation present at the input.



Most phase-fired controllers use thyristors or other solid state switching devices as their control elements. Thyristor-based controllers may utilise TRIACs(bt136 series) allowing the controller to decide when to switch the output on, but these triacs donot have the control to turn off using the gate signal. Moreover, an IC or system is required to instruct these triacs when to turn on such as a zero crossing detector IC.

**LINE COMMUTATION:**

Generally, if we consider AC supply, the current will flow through the zero crossing line while going from positive peak to negative peak. Thus, a reverse voltage will appear across the device simultaneously, which will turn off the thyristor immediately. This process is called as natural commutation as thyristor is turned off naturally without using any external components or circuit or supply for commutation purpose.

**Calculations:**

Calculations to find the time that the arduino board has to wait after a zero crossing is detected and if user has instructed, to turn on the triac in order to control brightness or just perform switching operation.

Frequency of input signal=50hz

This means that 50 cycles are completed in 1 second, hence time required for a single wave =(1/50)= 20ms. Or time taken by a single half wave=10ms.

Hence after a zero crossing is detected, we get about 10 ms to control the output power transferred to the load. If the triac is fired at 0ms is fired after detection of zero crossing, full power is transferred is to load, whereas if the triac is triggered is 5ms, then half power is transferred to load and so on.

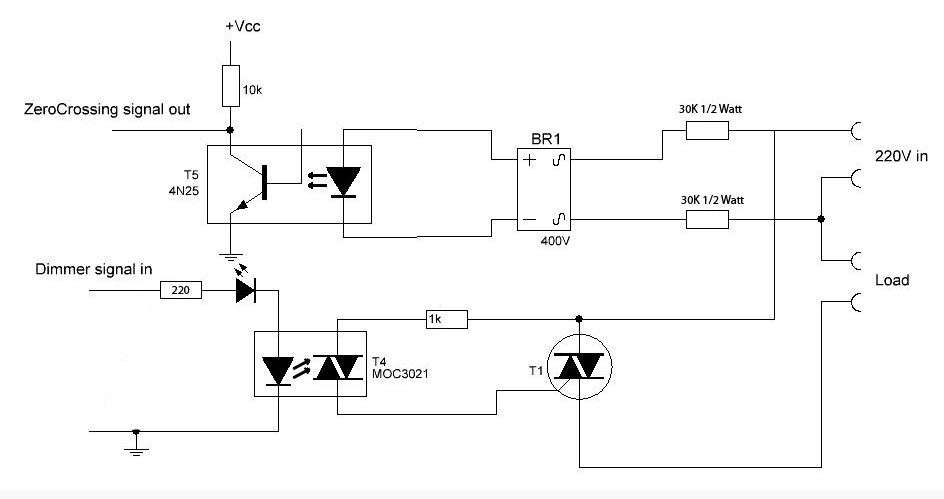
10ms=10,000us, let us divide this time interval into 128 steps for control purposes.

Hence 10,000/128=78.125, step length approximately=75.

Hence , if triac is fired after (75\*(128/2))microseconds, half power is transferred.

Or if triac is fired after (75\*128),no power is transferred to load.

**CIRCUIT DIAGRAM**



Zero crossing signal out is pin 2 of arduino(interrupt 0)

Dimmer signal in in pin 3 of arduino.

Vcc is +5 or +9v dc supply.

Components used:

* Zero crossing detector 4n25IC
* Resistor 10kohm(1)
* Bridge rectifier 400v(1)
* Resistor 1kohm(1)
* 30kohm resistor ½ watt (2)
* MOC3021IC
* Triac bt136
* Arduino Uno
* Wiznet w5100 ethernet shield for arduino uno.
* 40watt , 230v incandescent bulb.
* Wifi enabled modem
* Workstation in Arduino IDE installed.
* 230v, 50hz supply
* Ethernet cables
* Usb cables
* Connectors
* PCB  
  Approximate cost of project = Rs.3000

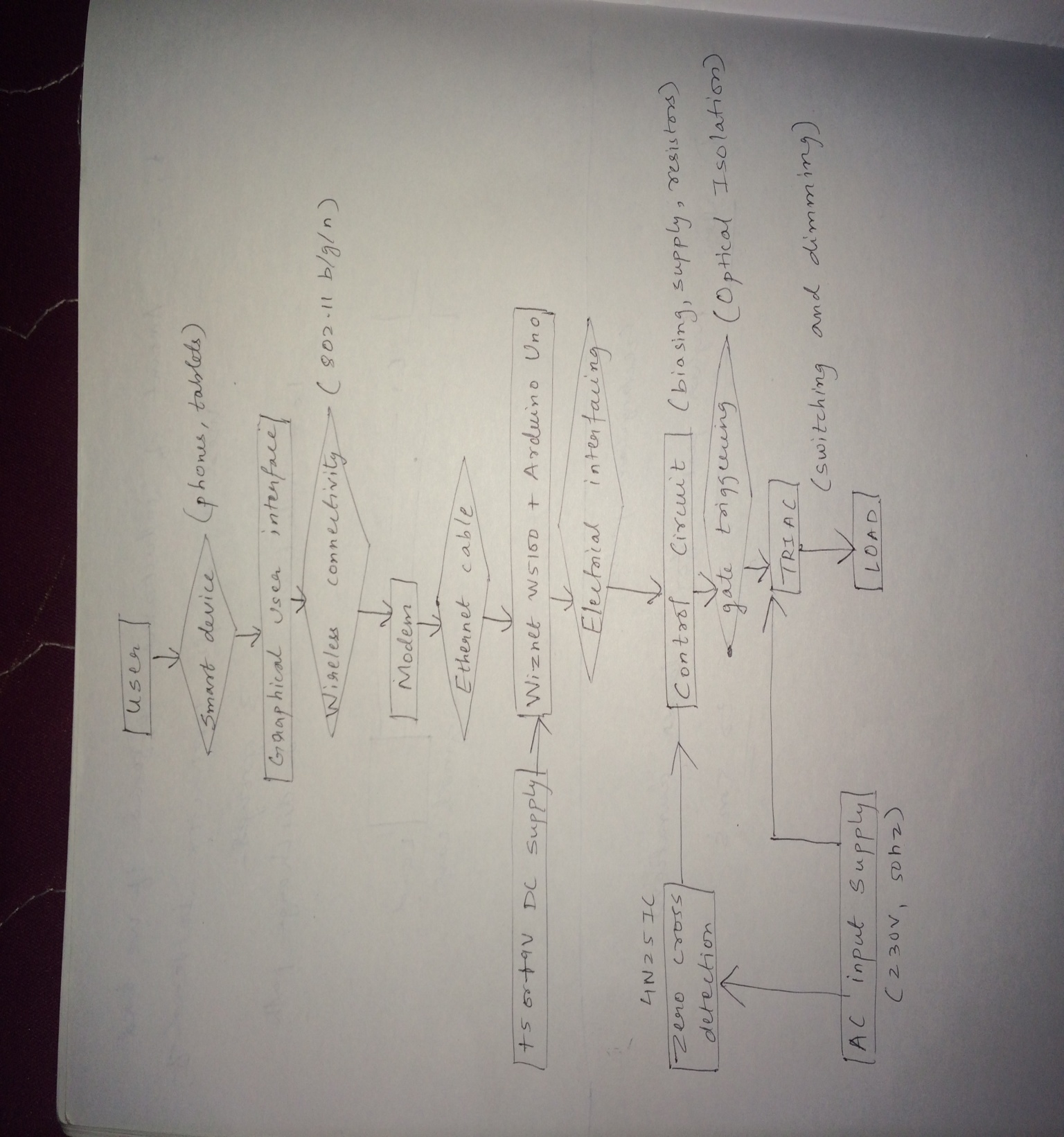
For information on components used, refer **APPENDIX SECTION.**

DESIGN

Electrical aspects of the design are adopted from the data sheets of the respected ICs provided in the Appendix section of this report, which contains all the biasing resistor selection procedures and gate interfacing circuit of triac to arduino.

Electrical aspects of the design are not discussed here since, there is no mathematical procedure done to select these values of components , rather they are adopted from empirical values which are component selective and can be readily obtained from the datasheets present in the Appendix section.

**Flow of signal**

****

**Working**

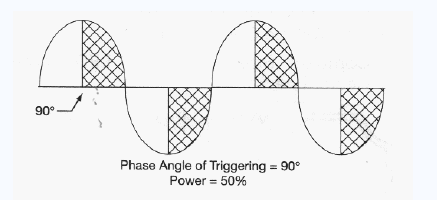
Refer the flow of signal as per the flowchart on the previous page.

Rectangular boxes represent Entities and diamond patterns indicate the relationship between entities.

As soon as the user inputs the signal through a GUI hosted on the local server designed by code on the wifi shield attached to arduino, The signal reaches the arduino board for processing through a series of channels including 802.11 b/g/n wifi and hard wired Ethernet cables attached from the modem to the arduino system.

Note that only the part of flowchart extending from user’s smart device to arduino board is wireless, everything except that connection is electrical.

The arduino then receives the user’s signal to control the brightness and checks when the next zero crossing on the sine wave is going to occur though an interrupt function. Once the zero crossing of the input wave occurs, the control is transferred to that interrupt function which calculates the amount of time to wait until firing the triac to achieve required chopping of wave to obtain the required brightness or voltage level. Once this time delay is calculated , the arduino waits for this time denoted by firing angle ‘alpha’ and then gives a high pulse to the gate of the triac. Hence, in this way a controlled output voltage is achieved across the load.



An example above shows, how the bulb(load) can be controlled through this scheme to achieve half brightness.

Code used:

Please consult the APPENDIX section on Arduino IDE and java script for more reference to the following pages

The code implemented can be broken down in to parts as:

Design of webpage;

Establishment of connection with user and arduino board;

Receiving input signals;

Processing input signals;

Control of electrical circuit through input signals.

//for use with IDE 1.0

//open serial monitor to see what the arduino receives

//use the \ slash to escape the " in the html

//for use with W5100 based ethernet shields

#include <SPI.h>

#include <Ethernet.h>

int AC\_LOAD = 3;

int dimming;

byte mac[] = { 0x8B, 0x13, 0xB8, 0xCF, 0x21, 0xB6 }; // <­­­­­­­ PUT YOUR MAC

byte ip[] = { 192, 168, 1, 7 }; // <­­­­­­­ PUT YOUR IP

byte gateway[] = { 192, 168, 1, 1 }; // <­­­­­­­ PUT YOUR ROUTERS

byte subnet[] = { 255, 255, 255, 0 }; //

String readString;

void setup(){

pinMode(AC\_LOAD, OUTPUT);

/start Ethernet

server.begin();

//enable serial data print

Serial.begin(9600);

Serial.println("server LED test 1.0"); // so that we can know what is getting

}

void zero\_crosss\_int() //function to be fired at the zero crossing to dim thlight

{

// Firing angle calculation : 1 full 50Hz wave =1/50=20ms

// Every zerocrossing thus: (50Hz)­> 10ms (1/2 Cycle)

// For 60Hz => 8.33ms (10.000/120)

// 10ms=10000us

// (10000us ­ 10us) / 128 = 75 (Approx) For 60Hz =>65

int dimtime = (75\*dimming); // For 60Hz =>65

delayMicroseconds(dimtime); // Wait till firing the TRIAC

digitalWrite(AC\_LOAD, HIGH); // Fire the TRIAC

delayMicroseconds(10); // triac On propogation delay

digitalWrite(AC\_LOAD, LOW); // No longer trigger the TRIAC (the next zero)

}

void loop()

{

// (for 60Hz use 8.33) Some Triacs need a longer period

// Create a client connection

EthernetClient client = server.available();

if (client) {

while (client.connected()) {

if (client.available()) {

char c = client.read();

//read char by char HTTP request

if (readString.length() < 100) {

//store characters to string

readString += c;

//Serial.print(c);

}

if (c == '\n') {

Serial.println(readString); //print to serial monitor for debuging

/\* Start OF HTML Section. Here Keep everything as it is unless you

understands its working \*/

client.println("HTTP/1.1 200 OK"); //send new page

client.println("Content­Type: text/html");

client.println();

//client.println("<meta http­equiv=\"refresh\" content=\"5\">");

client.println("<HTML>");

client.println("<HEAD>");

client.println("<meta name='apple­mobile­web­app­capable'

content='yes' />");

client.println("<meta name='apple­mobile­web­app­status­bar­style'

content='black­translucent' />");

client.println("<link rel=\"stylesheet\" type=\"text/css\"

href=\"http://arduino­autohome.googlecode.com/svn/trunk/autohome.css\" />");

client.println("</HEAD>");

client.println("<body bgcolor=\"#D0D0D0\">");

client.println("<hr/>");

client.println("<hr/>");

client.println("<h4><center><img border=\"2\"

src=\"https://lh3.googleusercontent.com/­

C6BoJrRUFko/UEUFeCwkvdI/AAAAAAAAAOc/E7gcYvPV6r4/s960/Logo.jpg\" /></center>

</h4>");

client.println("<hr/>");

client.println("<hr/>");

client.println("<br />");client.println("<br />");client.println("<br />");

client.println("<br />");

client.println("<br />");

// Relay Control Code

client.println("<a href=\"/?relay1on\"\">Light at 100 %</a>")

client.println("<a href=\"/?relay1off\"\">Light at 20 %</a><br />");client.println("<br />");

client.println("<br />");

client.println("<br />";

client.println("<br />")

client.println("<a href=\"/?relay2on\"\">Turn On Light 2</a>")

client.println("<a href=\"/?relay2off\"\">Turn Off Light 2</a><br

client.println("<br />);

client.println("<br />";

client.println("<br />")

client.println("<br />");

client.println("<a href=\"/?relay3on\"\">Turn On Light 3</a>");

client.println("<a href=\"/?relay3off\"\">Turn Off Light 3</a><br

/>");

client.println("<br />");

client.println("<br />");

client.println("<br />");

client.println("<br />");

client.println("<a href=\"/?relay4on\"\">Turn On Light 4</a>");

client.println("<a href=\"/?relay4off\"\">Turn Off Light 4</a><br

/>");

client.println("<br />");

client.println("<br />");

// control arduino pin via ethernet Start //

label1: { if(readString.indexOf("?relay1on") >0)//checks for on

{

while(1)

{

dimming = 20;

attachInterrupt(0, zero\_crosss\_int, RISING);

if(readString.indexOf("?relay1off") >0)

{

goto label2;

}

Serial.println("Led On");

// client.println("<link rel='apple­touch­ico'

href='http://chriscosma.co.cc/on.png' />");

//client.println("Light 1 Is On");

client.println("<br />");

}

}

}

label2: { if(readString.indexOf("?relay1off") >0)//checks for off

{

while(1)

{

dimming = 80;

attachInterrupt(0, zero\_crosss\_int, RISING);

if(readString.indexOf("?relay1on") >0)

{

goto label1;

}

}

Serial.println("Led Off");

//client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/off.png' />");

//client.println("Light 1 Is Off");

client.println("<br />");

}

}

if(readString.indexOf("?relay2on") >0)//checks for on

{

digitalWrite(7, HIGH); // set pin 4 high

Serial.println("Led On");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/on.png' />");

//client.println("Light 2 Is On");

client.println("<br />");

}

else{

if(readString.indexOf("?relay2off") >0)//checks for off

{

digitalWrite(7, LOW); // set pin 4 low

Serial.println("Led Off");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/off.png' />");

//client.println("Light 2 Is Off");

client.println("<br />");

}

}

if(readString.indexOf("?relay3on") >0)//checks for on

{

digitalWrite(8, HIGH); // set pin 4 high

Serial.println("Led On");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/on.png' />");

// client.println("Light 3 Is On");

client.println("<br />");

}

else{

if(readString.indexOf("?relay3off") >0)//checks for off

{

digitalWrite(8, LOW); // set pin 4 low

Serial.println("Led Off");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/off.png' />");

//client.println("Light 3 Is Off");

client.println("<br />");

}

}

if(readString.indexOf("?relay4on") >0)//checks for on

{

digitalWrite(9, HIGH); // set pin 4 high

Serial.println("Led On");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/on.png' />");

//client.println("Light 4 Is On");

client.println("<br />");

}

else{

if(readString.indexOf("?relay4off") >0)//checks for off

{

digitalWrite(9, LOW); // set pin 4 low

Serial.println("Led Off");

client.println("<link rel='apple­touch­icon'

href='http://chriscosma.co.cc/off.png' />");

//client.println("Light 4 Is Off");

client.println("<br />");

}

}

// control arduino pin via ethernet End //

// Relay Status Display

client.println("<center>");

client.println("<table border=\"5\">");

client.println("<tr>");

if (digitalRead(3))

{

client.print("<td>Light 1 is ON</td>");

}

else

{

client.print("<td>Light 1 is OFF</td>");

}

client.print("<td>Light 2 is ON</td>");

}

else

{

client.print("<td>Light 2 is OFF</td>");

}

if (digitalRead(8))

{

client.print("<td>Light 3 is ON</td>");

}

else

{

client.print("<td>Light 3 is OFF</td>");

}

if (digitalRead(9))

{

client.print("<td>Light 4 is ON</td>");

}

else

{

client.print("<td>Light 4 is OFF</td>");

}

client.println("</tr>");

client.println("</center>");

//clearing string for next read

readString="";

client.println("</body>");

client.println("</HTML>");

delay(1);

//stopping client

client.stop();

}

}

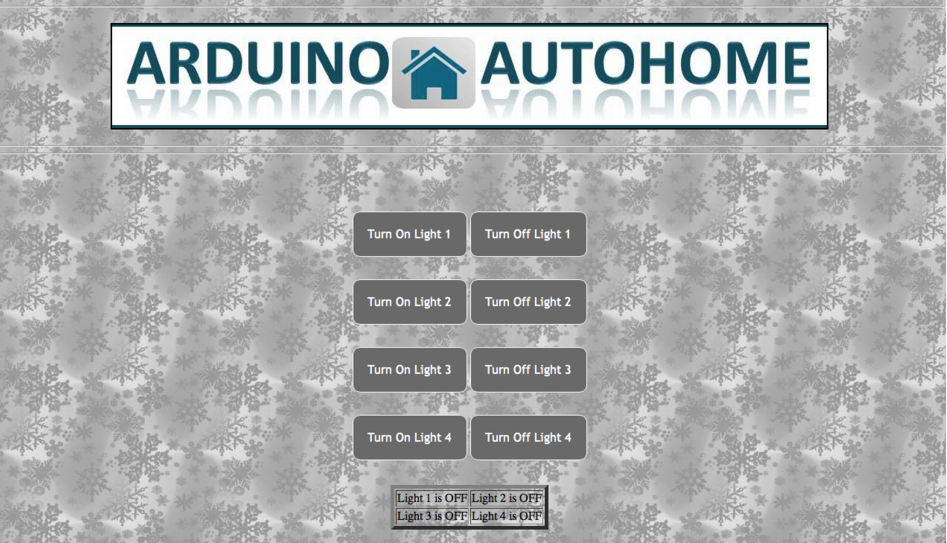
}

}

}

Graphical User Interface Developed:

Note: this is not the actual user interface developed, just an example.



Note: The actual user interface includes provisions to dim the bulbs which is not seen above, but will be produced at the time of experiment exhibition.

**Result:**

As per the successful completion of the project, the student has achieved the following:

* Development of prototype Home automation system to switch and dim an AC mains powered 40watt, 230v bulb, wirelessly from any smart device through internet.
* Successful design and implementation of a graphical user interface which is friendly to user and accessible through a WIFI connection on any browser based smart device.
* Integrating a Wiznet w5100 ethernet shield with an Arduino uno board to establish wireless connection with devices as well as controlling the connected electrical circuit to arduino.
* Imagination of concept, calculation of design variables, correct selection of hardware and software components utilised in this project.
* Safely and Appropriately controlling Ac mains voltage 230v, 50hz across the load with phase cutting or phase control technique utilising a bt136 triac and several other components.
* Student has combined skills and knowledge obtained from courses Power electronics and Linear integrated circuits

**Conclusion:**

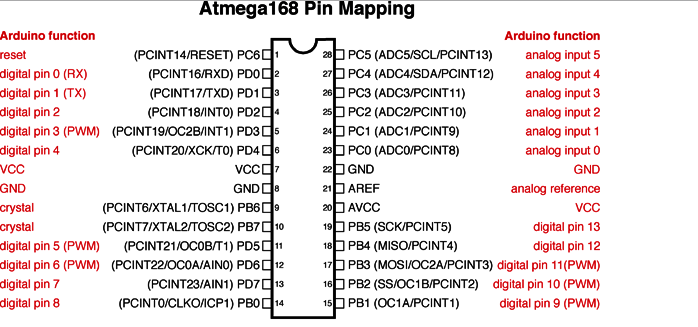
* Phase angle control or phase cutting can be used to successfully control the output voltage or power across an AC mains(230v,50hz) load.
* A zero crossing detector IC such as 4N25 plays a very important role in phase angle control of loads as it provides the programmer and arduino board a reference about when the AC mains sine wave voltage becomes zero in magnitude. This enables us to easily calculate firing angle using the 4N25 output signal as reference on the time axis.
* Wiznet w5100 integrated with an Arduino Uno board has sufficient computing and networking memory and power to host a user interface over a local wifi connection and simultaneously control the output power to load through phase cutting.
* BT136 series TRIAC has proved to be very effective in controlling 230v as well as firing at accurate timings by logic level signals which are supplied by Arduino output pins.
* Galvanic separation through optical isolation using MOC3021 series IC prevents any damage to and protects the source circuit to the gate terminal of TRIAC which is the arduino integrated system in this case, in case any fault occurs on the part of circuit carrying AC mains voltage.

**SCOPE FOR FUTURE WORK:**

* To implement complete automation in the system, user input can be totally eliminated or deprioritized by prioritizing natural environmental conditions as primary inputs. For example, the primary condition selecting the brightness of output bulb can be radiation intensity in the nearby environment. So, during the day, the bulb will be at lowest intensity and as night comes, the brightness automatically increases.
* The project developer can install sensors at different locations and continuously monitor the sensor states by programs stored in the Arduino Uno board and control necessary devices such as washing machines, garage doors ,speed of fan, fire alarm system etc with the help of advanced programs and fast speed communication channels.
* The project described by this report suffers through a limitation that the arduino board and user’s smart device should be on the same WIFI channel or network. This limitation can be eliminated by hosting the GUI webpage on the WORLD WIDE WEB by paying some amount to any prominent internet service provider. In this way, user can control his output devices connected to the arduino board from anywhere in the world.
* Instead of using a browser based webpage as a graphical user interface for the users, the project developer can develop an IOS , Andriod or windows operating system ‘APP’, to provide a more user friendly, swift and convenient method to control devices.

**Appendix**

1. Arduino uno : pin mapping



The complete arduino user manual is impractical to be attached here, for references or doubts, please consider <https://www.arduino.cc/>

1. JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Note: Java script has been extensively used in the code for this project to make the guided user interface, for details about functions and parameters used, consult : <http://www.tutorialspoint.com/javascript/>

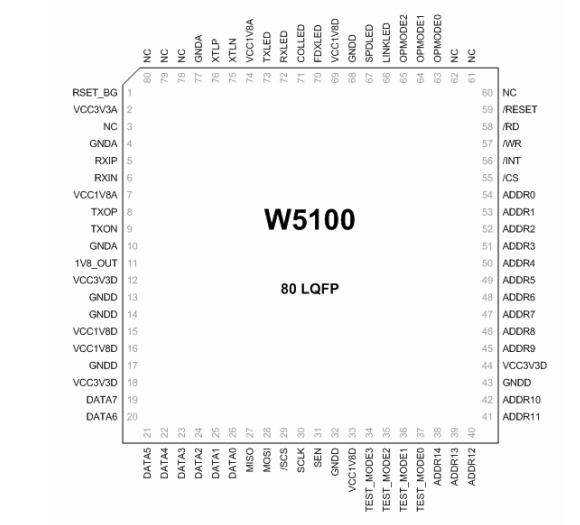
1. Arduino IDE

ARDUINO 1.6.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board.  
visit <https://www.arduino.cc/> for more details.

1. Wiznet w5100 ethernet shield for Arduino Uno

Target Applications The W5100 is well suited for many embedded applications, including: - Home Network Devices: Set-Top Boxes, PVRs, Digital Media Adapters - Serial-to-Ethernet: Access Controls, LED displays, Wireless AP relays, etc. - Parallel-to-Ethernet: POS / Mini Printers, Copiers - USB-to-Ethernet: Storage Devices, Network Printers - GPIO-to-Ethernet: Home Network Sensors - Security Systems: DVRs, Network Cameras, Kiosks - Factory and Building Automations - Medical Monitoring Equipments - Embedded Servers

Pin assignment : 

Features - Support Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet - 10BaseT/100BaseTX Ethernet PHY embedded - Support Auto Negotiation (Full-duplex and half duplex) - Support Auto MDI/MDIX - Support ADSL connection (with support PPPoE Protocol with PAP/CHAP Authentication mode) - Supports 4 independent sockets simultaneously - Not support IP Fragmentation - Internal 16Kbytes Memory for Tx/Rx Buffers - 0.18 µm CMOS technology - 3.3V operation with 5V I/O signal tolerance - Small 80 Pin LQFP Package .

**INDEX**

PAGE NUMBER TOPIC

3 ABSTRACT

4 INTRODUCTION

6 SURVEY OF LITERATURE

8 THEORY AND CALCULATIONS

10 CIRCUIT DIAGRAM

11 COMPONENTS USED AND DESIGN

13 WORKING OF SYSTEM

14 BEGINNING OF CODE

25 GRAPHICAL USER INTERFACE

26 RESULT

27 CONCLUSION

28 SCOPE FOR FUTURE WORK

29 APPENDIX

--- REFERENCES

**REFERENCES:**

# 1 Power Electronics: Converters, Applications, and Design, 3rd Edition

Ned Mohan, Tore M Underland, William P, Robbins

September 2002, ©2003

# 2 Building Internet of Things With the Arduino: Volume 1

Charalampos Doukas, 2012

# Foundations of Modern Networking: SDN, NFV, QoE, IoT, and CloudPaperback

by [William Stallings](http://www.amazon.in/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=William+Stallings&search-alias=stripbooks) , 2015