IP NETWORK BASED ENERGY MANAGEMENT AND CONTROL SYSTEM

MCS

By

NC Waleed Ahmed

NC Munazza Farooq

NC Jawad Tariq

NC Shameen Arif

Project Supervisor: Lt. Col. Dr. Adnan Ahmed Khan

Submitted to the Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Rawalpindi. In partial fulfillment for the requirements of a B.E Degree in Telecom Engineering June 2014.

ABSTRACT

Companies producing consumer products have begun working on ways to control their products over the IP (internet protocol). Multinational electronics firms are marking a fair share of their lines using technologies like the RF ID, a front end to the IP networks. The main feature that is achieved by the already-existing home automation systems is that, every piece of equipment integrated is to be controlled by a central unit. The proposed solutions based on this idea are relatively expensive and difficult to configure per each system environment. IP based systems are supposed to be simpler to be accessed and easier to manipulate, with the only necessity being, close proximity to a direct internet connection. With this in mind, we designed a low cost hardware/software framework based energy management and control system for home/industrial automation, which is to be accessed and controlled through a direct internet connection even in dynamic IP configurations. We have developed the software and hardware infrastructure necessary for this system.

There are many commercial home automation systems available. However, these are often proprietary and/or designed for limited use. This project seeks to pave the way for IP-enabling home appliances, making such devices part of the Internet. Therefore, these devices can individually be controlled both from within the home and remotely. Internet enabling each of these devices eliminates the need for special Internet connected control units, simplifying home automation.

DECLARATION

We declare that this thesis entitled "IP NETWORK BASED ENERGY MANAGEMENT AND CONTROL SYSTEM" is the result of our own work except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

DEDICATION

Dedicated to all our family members, friends and mentors for their utmost guidance and support.

ACKNOWLEDGEMENTS

We express total submission and eternal gratitude to ALMIGHTY ALLAH, The Most Gracious, The Most Merciful, with whose mercy and grace, this endeavor could materialize in its present form and in the stipulated time. We deem it an honor and privilege to pay thanks and gratitude to our project supervisor Lt. Col. Adan Ahmed Khan, for his guidance, continuous encouragement and inspiration, especially during hard and desperate moments of the research work. He helped us at every step in completing this tedious and gigantic task.

We thank Sir Mughees Sarwar Awan for his passion and spirit that guided us through all the difficulties in the most desperate of times, he was a great and dedicated mentor and this project would not have been possible without him.

No acknowledgement would ever adequately express our obligation to our parents who always wish to see us flying higher in the sky with success. Without their prayers, sacrifices and encouragements, the present study would have been merely a dream.

Table of Contents

Table of Figures

List of Abbreviations

IP	Internet Protocol
Wi-Fi	Wireless Fidelity
GUI	Graphical User Interface
AVR	Alf (Egil Bogen) and Vegard (Wollan)'s RISC processor
PCB	Printed Circuit Board
MCU	Micro Controller Unit
PC	Personal Computer
VB	Visual Basic
UDP	User Datagram Protocol
LAN	Local Area Network
SPI	Serial Peripheral Interface
MOSI	Master Out Slave In
MISO	Master In Slave Out
SCK	Serial Clock
SS’	Slave Select
PWM	Pulse Width Modulation

CHAPTER NO.1

1. Introduction:

Technological advancements have forced the world to shift everything on the IP network, such as, IPTVs and VoIP. People nowadays prefer to use Skype, Viber and Whatsapp instead of using traditional means. This revolution in the technology provided us with the motivation to design a low cost hardware/software framework, which, in the future, will allow the preexisting home/industrial automation systems to be accessed over an IP network.

Our final product shall consist of an android device, which, when connected to the network via Wi-Fi, will be capable of controlling states of light bulbs, fans and Air conditioners. All of these appliances will be assigned unique IP addresses and the user will be able to switch them On/Off through a control packet sent over the IP network. An android application will be developed in order to provide the user an easy platform, where, by simply touching the icon, the user will be able to switch the appliances in accordance with the user's requirements. At the other end, controller hardware will be interfaced with the appliances, which, after receiving a control packet, will perform the desired operations. This will allow the user to roam freely and access the devices from anywhere in the house as long as the user is connected to the network.

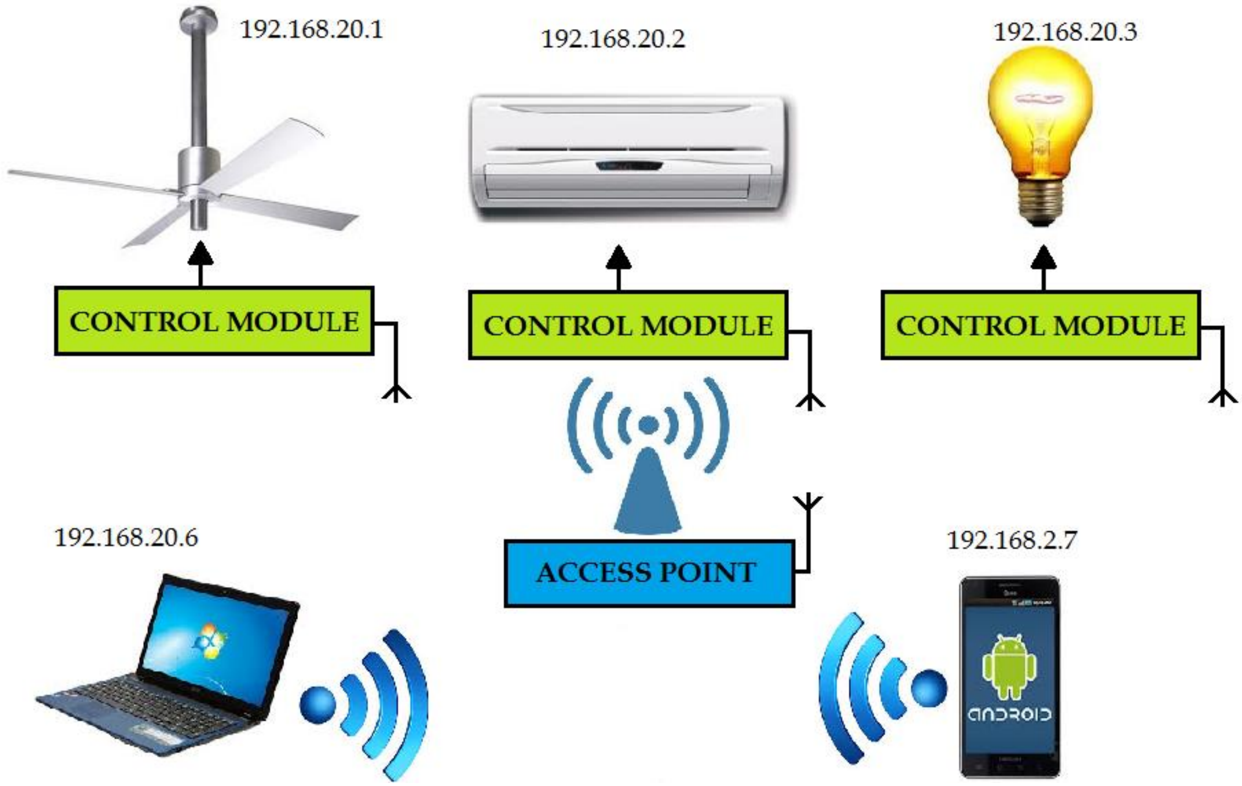


Figure 1. Block diagram of the Project

The method is split into 4 phases:

1st phase: Development of Visual Basic application on PC using Visual Studio software and establishing serial communication. Hardware control panel development by first programming microcontroller and designing the control module for switching on/off relays which is operated by PC application serially.

2nd phase: Developing application on VB which sends data on Ethernet and programming microcontroller to receive the packet and perform switching of devices as defined by information on the IP packet. Each device will have its own control module and its own IP address.

3rd phase: Development of Android app using Eclipse software and virtually testing it using Android SDK, along with the development of Client-server network.

4th phase: Hardware control panel development in which microcontroller interfaced with Wi-Fi module so that the control module is capable of communicating with android device. Command sent by user on android device and action performed by the control module.

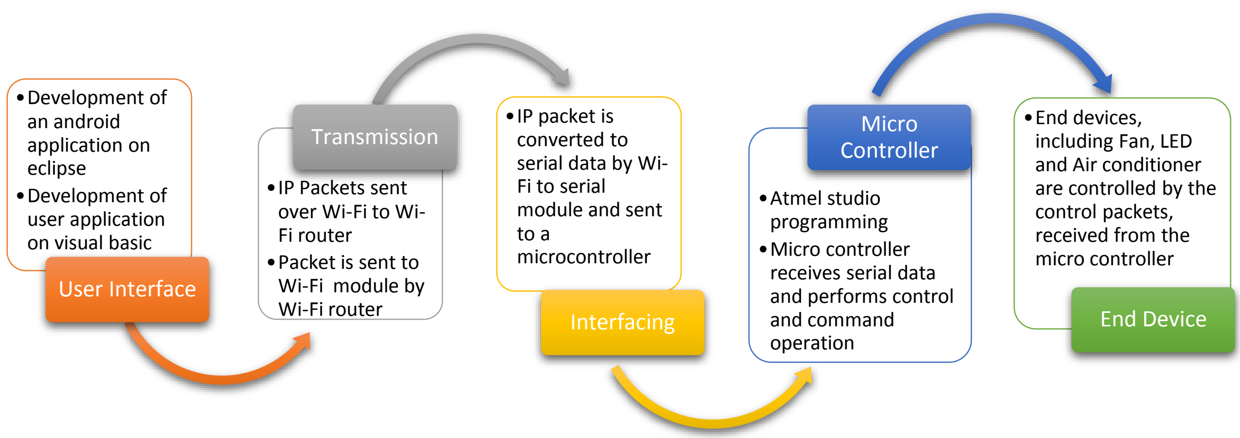


Figure 2. Flowchart of the Project

1.1.Background study:

Being a feature of science fiction for so many years, home automation opened up the flood gates since 20th century after the widespread introduction of electricity in houses with the ever increasing advancement of IT. Earlier, the remote control devices emerged in the late 18th century. Mankind has been trying to achieve comfort in combination with simplicity since the very beginning. This goal, despite being so bright, still stumbles with lots of obstacles. The literal definition of home is a place where people relax, spend quality time with their loved ones and even work. Building a home went through a lot of phases, from a cave in the Stone Age to the apartments or skyscrapers in 21st century. The concept of automation of houses, although considered a luxury today, actually paved way for the smart homes or the smart devices.

A lot of problems that popped up gradually, as the people continued to live in their homes, needed to be resolved. Some of the major problems that the people face in their ordinary houses is wastage of energy, security and time saving. The devices that are turned smart, in turn providing a collective concept of smart home, solve these problems quite efficiently. Such houses are supplied with the surveillance cameras that monitor the house 24/7 from every angle and the footage is displayed on the screens within the house. Also, warning alerts are used and the messages are sent out to the occupants of the house. The lighting system is connected to the sensors that automatically turn the lights on or off as per the user's wishes and in turn, improve the power consumption. These smart devices can be controlled from any portable smartphone, tablet or a PC be a special application. [1]

1.2. Objective:

The goal of our project is to build a system consisting of an android device, which, when connected to the network via Wi-Fi, will be capable of controlling states of light bulbs, fans and Air conditioners. All of these appliances will be assigned unique IP addresses and the user will be able to switch them On/Off through a control packet sent over the IP network. An android application will be developed in order to provide the user an easy platform, where, by simply touching the icon, the user will be able to switch the appliances in accordance with the user's requirements. At the other end, controller hardware will be interfaced with the appliances, which, after receiving a control packet, will perform the desired operations. This will allow the user to roam freely and access the devices from anywhere in the house as long as the user is connected to the network.

1.3. Motivation:

Technological advancements have forced the world to shift everything on the IP network, such as, IPTVs and VoIP. People nowadays prefer to use Skype, Viber and Whatsapp instead of using traditional means. This revolution in the technology provided us with the motivation to design a low cost hardware/software framework, which, in the future, will allow the preexisting home/industrial automation systems to be accessed over an IP network.

With Internet Protocol as the backbone of a wide network, everything that requires a standalone network today simply becomes encompassed under an IP master. This is a very high level interoperability. Controllers are designed for the IP network in the building management sector of this new environment. More importantly, they are actually designed to thrive over the IP network.

IP networks are the most common element linking intelligent devices today, and they will become more prevalent within buildings of all types. IP enabled devices can take advantage of the intelligence embedded in other IP enabled devices, thereby adding value to all of them. An investment in IP based control should be rewarded with leading-edge features and capabilities, including advanced control opportunities, Superior failure mode ability, Cost optimization, Scheduling, User Interface, System Security, Monitoring and Control, Transaction Recording, Alarm and Event Processing historical and Totaling Data. IP based control is the future of modern control and automation solutions.

1.4. Scope, specifications and deliverables of the project

Our final product will be an android tablet which is connected to the network via Wi-Fi and is capable of controlling different states of light bulbs, fans and Air conditioners. All of these three appliances will have their unique IP addresses and will be turned on/off through a control packet sent over the IP network. The android device will have a GUI (Graphical User Interface) which would provide the user an easy platform to just touch the icon for this app for controlling the devices as per will. On the other end, the control hardware will be interfaced with appliances, which will perform the desired operations after receiving a control packet over the Ethernet cable. The user will be able to roam freely and control the devices even from outside the room. Our project incorporates the design of control hardware capable of switching the electrical devices, network and system programming for establishing link between the device and system and also for setting up of IP network. Microcontroller programming will be required to program the controller to enable it to translate the control packet information into IP addresses and control commands. Android device will be programmed using the android application development tools and the application will be developed to communicate with the network via Wi-Fi. Power Electronics will be employed for the switching of appliances and controlling their states. Moreover, designing the burner of the controller ourselves enables us to utilize the concepts of the subject of electronic circuits and PCB designing.

CHAPTER NO.2

2. Literature review

2.1. Review of related projects in MCS

After explaining in detail what and how we propose to achieve with this project, let's take a look at the previous projects related to this concept in MCS (NUST):

a) Automated home system using BAS with GSM support:

The aim of this project was to control, manage and monitor all the electrical devices either digital or analog remotely by using the GSM modem and through PC using serial port. GSM mechanism of communication between the microcontroller and the client via Short Message Service (SMS). User will be able to control many electrical gadgets by sending SMS (standard format) to the control system based on microcontroller and validated. If the received SMS commands are valid then requisite action will take place to control the devices. The system constantly monitors the status of all the devices connected. If the user wants to know the status of any device or all the devices, he may send a status message and the system based on microcontroller automatically sends back the SMS based alerts to the client indicating the status of all the devices. [2]

b) Android based home automation system:

The aim of this project was to develop a home automation system, which is mainly integration of hardware devices control module and Android application. The hardware control part was done using microcontrollers, where the front end application was based on Android. The Android based application was used for performing several applications from mobile set. The Android application included image/video acquisition from camera, capable to ON and OFF certain switches, control fan speeds, brightness of bulbs and even manual control over all appliances too.

The Product was developed with a user-friendly GUI enabling user to control the home appliances within Wi-Fi/Internet zone. In addition to the above mentioned factors, a user also receives live video streaming on android device from security camera, door access through android device. It enables the user to monitor when anyone arrives at the door and open the door by just the specific button from his smart phone using the application. Reception of alerts on android device via SMS in case of emergency situations like if a burglar or fire breaks out in the residence boundaries. And a password scheme for client-server connection for security. [3]

2.2. Review of related projects industrially

Now, coming over to the projects being carried out at the commercial level in the world related to the technology that we propose to apply, following projects are amongst the most highlighted:

a) NXP's GreenChip smart lighting solution:

Using a 2.4-GHz IEEE 802.15.4 standard-compatible wireless microcontroller with a Tx/Rx current below 17 mA. Low-power, IP-based wireless connectivity enabled by JenNet-IP network layer software.

GreenChip-enabled light bulbs will be able to operate on the same wireless sensor networks consumers may be using at home for energy metering, smart appliances and security systems. NXP’s JenNet-IP network layer software provides the ultra-low-power wireless connectivity in the GreenChip smart lighting solution. JenNet-IP is a 6LoWPAN mesh-under tree network with low memory footprint, specifically targeting low-power IEEE 802.15.4-based networking for residential and industrial applications. [4]

b) AT&T Introduces Digital Life: IP-Based Home Automation And Security System With 24/7 Monitoring Centers:

AT&T has just announced a new security and home-automation system called Digital Life, which will be an IP-based platform that allows users to monitor and detect activity throughout their house remotely, and “take action” (as AT&T put it) on devices like PCs, tablets, and smartphones. Functions include access to automation, energy and water controls, and security systems. AT&T will thus be introducing a new branch called the Digital Life group, which will work in “AT&T owned-and-operated 24/7 security monitoring centers.” [5]

Taking a look at what our own country, Pakistan, has in store regarding this technology, we find one very promising product launch “Future Energy”. It is a complete Home Automation System in Pakistan. LightwaveRF is a radio system (like Bluetooth) that lets you operate light switches and power sockets with your phone where ever you are. The LightwaveRF Connect series is the beginning of a new revolution in home automation technology. It is breakthrough in design that, for the first time, has allowed a range of sophisticated and stylishly crafted devices to be assembled that are affordable and accessible to everyone.

LighwaveRF Lighting with built-in wireless communications protocol allows switches and dimmers to be installed into standard 25/35mm back boxes, in the same way as you wire any standard switch. All LightwaveRF devices can be operated by any of the range of remote controllers, the LightwaveRF Master wall switch or even a smartphone, ipad or ipod touch, from anywhere in the world. [6]

2.3. Microcontroller review

A microcontroller (sometimes abbreviated µC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications. [7]

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (single-digit milli-watts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nano-watts, making many of them well suited for long lasting battery applications.

2.3.1. History of microcontroller

The first single-chip microprocessor was the 4-bit Intel 4004 released in 1971, with the Intel 8008 and other more capable microprocessors becoming available over the next several years. However, both processors required external chips to implement a working system, raising total system cost, and making it impossible to economically computerize appliances.

The very first computer system on a chip has both RAM and ROM on the same chip. Two variants of the microcontrollers were present at that time. One had an erasable EPROM program memory, with a transparent quartz window in the lid of the package to allow it to be erased by exposure to ultraviolet light. The other was a PROM variant which was only programmable once.

By nineties, when EPROM was introduced it allowed microcontrollers to electrically erase their memory quickly, also being cost effective, it allowed rapid prototyping in the system programming. At the same time, Atmel introduced the first microcontroller using Flash memory, a special type of EEPROM. Other companies rapidly followed suit, with both memory types.

Prices have decreased over time; presently microcontrollers cost less and are easily available. In the future, MRAM has the potential to be utilized in the microcontrollers because it has the infinite staying power.

2.3.2. Programming

Typically microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assemblers are used to convert high-level language and assembler language codes into a compact machine code for storage in the microcontroller's memory. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory, or program memory that may be field-alterable flash or erasable read-only memory.

2.3.3. Microcontroller features

Microcontrollers usually contain from several to dozens of general purpose input/output pins (GPIO). GPIO pins are software configurable to either an input or an output state. When GPIO pins are configured to an input state, they are often used to read sensors or external signals. Configured to the output state, GPIO pins can drive external devices such as LEDs or motors.

Many embedded systems need to read sensors that produce analog signals. This is the purpose of the analog-to-digital converter (ADC). Since processors are built to interpret and process digital data, i.e. 1s and 0s, they are not able to do anything with the analog signals that may be sent to it by a device. So the analog to digital converter is used to convert the incoming data into a form that the processor can recognize. A less common feature on some microcontrollers is a digital-to-analog converter (DAC) that allows the processor to output analog signals or voltage levels.

In addition to the converters, many embedded microprocessors include a variety of timers as well. One of the most common types of timers is the Programmable Interval Timer (PIT). A PIT may either count down from some value to zero, or up to the capacity of the count-register, overflowing to zero. Once it reaches zero, it sends an interrupt to the processor indicating that it has finished counting.

2.3.4. Programming environment

Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target microcontrollers. These languages are either designed especially for the purpose, or versions of general purpose languages such as the C programming language. Compilers for general purpose languages will typically have some restrictions as well as enhancements to better support the unique characteristics of microcontrollers. Some microcontrollers have environments to aid developing certain types of applications. Microcontroller vendors often make tools freely available to make it easier to adopt their hardware.

Many microcontrollers are so quirky that they effectively require their own non-standard dialects of C, such as SDCC for the 8051, which prevent using standard tools (such as code libraries or static analysis tools) even for code unrelated to hardware features. Interpreters are often used to hide such low level quirks.

Simulators are available for some microcontrollers. These allow a developer to analyze what the behavior of the microcontroller and their program should be if they were using the actual part. A simulator will show the internal processor state and also that of the outputs, as well as allowing input signals to be generated. While on the one hand most simulators will be limited from being unable to simulate much other hardware in a system, they can exercise conditions that may otherwise be hard to reproduce at will in the physical implementation, and can be the quickest way to debug and analyze problems.

Recent microcontrollers are often integrated with on-chip debug circuitry that when accessed by an in-circuit emulator via JTAG, allow debugging of the firmware with a debugger.

2.3.5. Microcontroller ATMEGA88

The microcontroller incorporated by us in our project is ATMEGA88. It is an electronic integrated circuit microcontroller produced by the Atmel Corporation. It has the basic Atmel AVR instruction set. One of the packaging configurations is the dual in-line package (DIP). It has 23 I/O pins, and operates at up to 20 MHz for clock speed. It has an 8-bit core.

2.3.5.1.Microcontroller features

Following is the list of features specific to this ATMEGA88: [8]

- ✓ High performance, low power Atmel, AVR, 8-bit microcontroller
- ✓ Advanced RISC architecture

– 131 powerful instructions – most single clock cycle execution

– 32 × 8 general purpose working registers

– Fully static operation

– Up to 20 MIPS throughput at 20MHz

– On-chip 2-cycle multiplier

- ✓ High endurance non-volatile memory segments

– 4/8/16 Kbytes of in-system self-programmable flash program memory

– 256/512/512 bytes EEPROM

– 512/1K/1Kbytes internal SRAM

– Write/erase cycles: 10,000 flash/100,000 EEPROM

– Data retention: 20 years at 85°C/100 years at 25°C

– Optional boot code section with independent lock bits

- ✓ In-system programming by on-chip boot program
- ✓ True read-while-write operation

– Programming lock for software security

Q Touch library support

– Capacitive touch buttons, sliders and wheels

– Q Touch and Q Matrix acquisition

– Up to 64 sense channels

✓ Peripheral features

– Two 8-bit timer/counters with separate pre-scale and compare mode

– One 16-bit timer/counter with separate pre-scale, compare mode, and capture mode

– Real time counter with separate oscillator

– Six PWM channels

– 8-channel 10-bit ADC in TQFP and QFN/MLF package

– 6-channel 10-bit ADC in PDIP Package

– Programmable serial USART

– Master/slave SPI serial interface

– Byte-oriented 2-wire serial interface (Philips I2

C compatible)

– Programmable watchdog timer with separate on-chip oscillator

– On-chip analog comparator

– Interrupt and wake-up on pin change

✓ Special microcontroller features

– Debug WIRE on-chip debug system

– Power-on reset and programmable brown-out detection

– Internal calibrated oscillator

– External and internal interrupt sources

– Five sleep modes: Idle, ADC noise reduction, power-save, power-down, and standby

I/O and packages

– 23 programmable I/O lines

– 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

- ✓ Operating voltage:

– 1.8V - 5.5V for Atmel ATmega48V/88V/168V

– 2.7V - 5.5V for Atmel ATmega48/88/168

- ✓ Temperature range:

– -40°C to 85°C

- ✓ Speed grade:

– ATmega48V/88V/168V: 0 - 4MHz @ 1.8V - 5.5V, 0 - 10MHz @ 2.7V - 5.5V

– ATmega48/88/168: 0 - 10MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V

- ✓ Low power consumption

– Active mode:

250µA at 1MHz, 1.8V

15µA at 32 kHz, 1.8V (including oscillator)

– Power-down mode:

0.1µA at 1.8V

2.3.5.2.Pin configuration

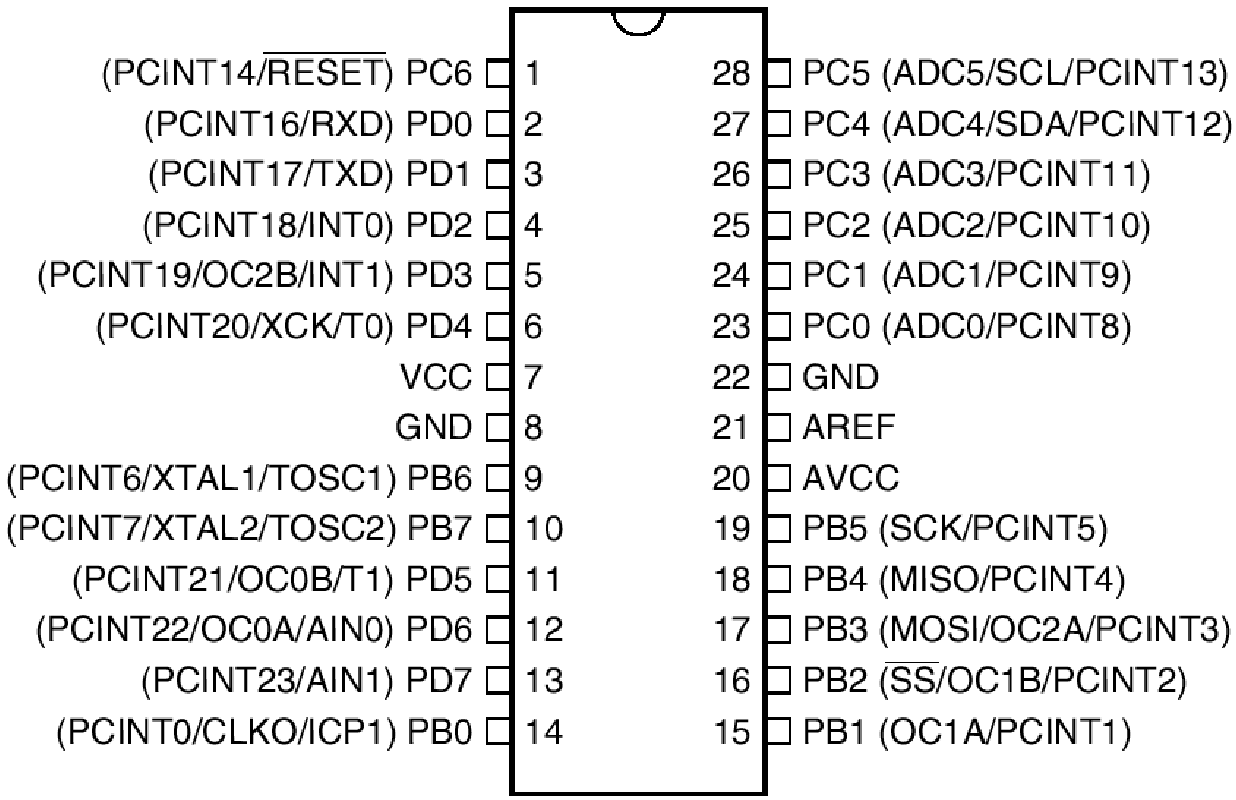


Figure 3. Microcontroller Pin Configuration

2.4.Power line communication

Power-line communication (PLC) carries data on a conductor that is also used simultaneously for AC electric power transmission or electric power distribution to consumers. [10]

A wide range of power-line communication technologies are needed for different applications, ranging from home automation to Internet access which is often called broadband over power lines (BPL). Most PLC technologies limit themselves to one type of wires (such as premises wiring within a single building), but some can cross between two levels (for example, both the distribution network and premises wiring). Typically transformers prevent propagating the signal, which requires multiple technologies to form very large networks. Various data rates and frequencies are used in different situations.

Power-line communications systems operate by adding a modulated carrier signal to the wiring system. Different types of power-line communications use different frequency bands. Since the power distribution system was originally intended for transmission of AC power at typical frequencies of 50 or 60 Hz, power wire circuits have only a limited ability to carry higher frequencies. The propagation problem is a limiting factor for each type of power-line communications. Data rates and distance limits vary widely over many power-line communication standards. Low-frequency (about 100–200 kHz) carriers impressed on high-voltage transmission lines may carry one or two analog voice circuits, or telemetry and control circuits with an equivalent data rate of a few hundred bits per second; however, these circuits may be many miles long. Higher data rates generally imply shorter ranges; a local area network operating at millions of bits per second may only cover one floor of an office building, but eliminates the need for installation of dedicated network cabling.

2.5. Fixed Network

Communication networks that provide the connections to the telecommunication devices at fixed locations constitute the concept of "Fixed Network". There are two types of fixed networks:

2.5.1. Wired fixed network

It usually comprises of the systems that have a fixed wire connection to the telephones (landlines). The wired connection can either be a metal wire (coaxial, twisted pair or an optical fiber cable). Such devices are connected to a wired end and are usually immobile. It provides with a type of Internet access where connections to service providers make use of the cables.

2.5.2. Wireless fixed networks

Wireless fixed network for telephony or fixed wireless refers to the operation of wireless devices or systems in fixed locations such as homes. Fixed wireless devices usually derive their electrical power from the utility mains electricity, unlike mobile wireless or portable wireless, which tend to be battery-powered. Although mobile and portable systems can be used in fixed locations, efficiency and bandwidth are compromised compared with fixed systems. Mobile or portable, battery-powered wireless systems can be used as emergency backups for fixed systems in case of a power blackout or natural disaster. [11]

However, fixed wireless broadband is a type of high-speed Internet access where connections to service providers use radio signals rather than cables. Areas that lack fiber optic cable, DSL, or cable television lines can still enjoy broadband Internet access via a fixed wireless service. [12] Although they tend to offer lower speeds than other types of broadband Internet, fixed wireless services generally support between one and 10 Mbps of network bandwidth.

2.6. RF network

Radio frequency (RF) is a rate of oscillation in the range of around 3 kHz to 300 GHz, which corresponds to the frequency of radio waves, and the alternating currents which carry radio signals. RF usually refers to electrical rather than mechanical oscillations; however, mechanical RF systems do exist. [13]

To receive radio signals an antenna must be used. However, since the antenna will pick up thousands of radio signals at a time, a radio tuner is necessary to tune into a particular frequency (or frequency range).

2.7. Wireless fidelity (Wi-Fi)

Wi-Fi is a technology that allows an electronic device to exchange data or connect to the internet wirelessly using microwaves in the 2.4 GHz and 5 GHz bands. The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network (WLAN) products that are based on the

Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards". However, since most modern WLANs are based on these standards, the term "Wi-Fi" is used in general English as a synonym for "WLAN". [14]

Many devices can use Wi-Fi, e.g., personal computers, video-game consoles, smartphones, some 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. To connect to a Wi-Fi LAN, a computer has to be equipped with a wireless network interface controller. The combination of computer and interface controller is called a station. All stations share a single radio frequency communication channel. Transmissions on this channel are received by all stations within range. The hardware does not signal the user that the transmission was delivered and is therefore called a best-effort delivery mechanism. A carrier wave is used to transmit the data in packets, referred to as "Ethernet frames". Each station is constantly tuned in on the radio frequency communication channel to pick up available transmissions.

CHAPTER NO.3

3. Automation

Automation (automatic control) is the term used for various control systems for the operation of equipment such as machinery, processes in factories, heat treating ovens and boilers, telephone network switch boards, steering of ships and aircrafts and other applications with minimal or reduced human intervention. Some of these process have been automated. The biggest benefit of automation systems is that they save labor, however, it is also used to save energy and materials and for the improvement in quality, accuracy and precision.

Automated home can vary from the simple devices that are providing the control over individual components to the integrated systems for complete house. For example a gadget can be an “X-10” device which can controls the appliance that is plugged into it. Integrated house system is such system that combines all the devices as well as all subsystems of home with one system which is centrally controlled. Automated home can include safety, telecommunications, security and lighting systems.

The industrial automation primarily deals with the automation of manufacturing process, quality control and the material handling processes. Programmable logic controllers and computers are used as general purpose controllers for the industrial processes. One of the increased trends is the usage of machine vision to provide the automatic inspection and robotic guidance. Another trend is the continuous increase in the use of robots. Energy efficiency in industrial processes has become a higher priority. Semiconductor companies like Infineon Technologies are offering 8-bit micro-controller applications for example found in motor controls, general purpose pumps and fans to reduce energy consumption and thus increase efficiency. [15]

3.1. Types of automation

The two common types of automation is feedback control, which is continuous and takes measurements using sensors and making calculated adjustments to keep the measured variables within the set range, and sequence control, where the programmed sequence of discrete operations is performed, often that is based on the systematic logic. Cruise control is example of the former while an elevator or an automated teller machine (ATM) is an example of the latter.

Control theory, which is the theoretical basis for feedback theory, also covers servomechanisms that are often part of an automated system. The feedback control is called “closed loop” while, non-feedback control is called an “open loop”.

3.1.1. Feedback control

This type of automation is accomplished with a controller. A controller must provide correction in a manner that maintains stability, for the feedback control to function properly. Maintaining stability is a principal objective of control theory and all the elements constituting the measurement and control of a single variable are called control loop.

The complexities of this are that the quantities involved are all different physical types. The temperature sensor signal may either be electrical or pressure from an enclosed fluid, while the controller may employ pneumatic, hydraulic, mechanical or electronic techniques to sense the error and send a signal to adjust the air pressure that moves the valve.

The very first controllers used analog methods to perform the calculations. These analog methods were also used in solving various differential equations of control theory. This is the main reason the analog computers were developed to solve control type problems and controllers were also developed. Analog computers were displaced by digital computers when they became widely available. Common applications of feedback control are control of temperature, pressure, flow, and speed.

3.1.2. Sequential control and logical sequence control

This type of control may either be for a fixed sequence or for a logical one that will perform different actions depending on various system states. A development of sequential logic, by which electrical relays engage electrical contacts which either start or interrupt power to a device. Telegraph networks were the first to make use of the relays, before they were developed for controlling the other devices. Using relays for control purposes allowed for the event-driven control, where the actions can be driven out of the sequence in response to the external events, which made them more flexible in their response than the rigid single-sequence cam timers.

3.1.3. Computer control

Computers are capable of performing both the sequential and feedback control. Typically, a single computer is able to do both in an industry. Programmable Logic Controllers (PLCs) are a type of special purpose microprocessor that replaced many types of hardware components such as timers and drum sequencers used inn relay logic. Standalone computers have increasingly been replaced by the general purpose process control computers because a single computer is capable of controlling the operation of hundreds of computers. Process control computers are capable of processing the date from a network of PLCs and controllers in order to implement typical control of many individual variables. In some cases, complex control algorithms are implemented using multiple inputs and mathematical manipulations.

Control of an ATM machine is an example of an interactive process in which a logic derived response to a user that is based on the information that is retrieved from the networked database.

CHAPTER NO.4

4. Home automation

Home automation is the residential extension of building automation. It is automation of the home, housework or household activity. Home automation may include centralized control of lighting, HVAC (heating, ventilation and air conditioning), appliances, security locks of gates and doors and other systems, to provide improved convenience, comfort, energy efficiency and security. Home automation for the elderly and disabled can provide increased quality of life for persons who might otherwise require caregivers or institutional care.

4.1. Types of home automation

There are three types of home automation systems, which are as follows:

4.1.1. Power line system

It is the most economical in all the automated home systems; this system depends on the power lines that exist in the home to transfer signals. These are often based on the X10 technology. In these systems, backward compatibility almost always refer to new gadgets working with older X10 systems. X10 is one of the oldest and most popular home automation protocols (not to be confused with a company by the same name). Many older or legacy products use this protocol. [16]

4.1.2. Wired systems

These systems use cables (Cat-5) to transfer signals. They may be connected into proprietary control system. These types of systems can be installed in new as well as in the existing homes. It is a good choice to hardwire an Automated Home system as the wiring is durable and integration of all the systems together is a simple process as the cables run to a common junction point.

4.1.3. Wireless system

The system is most suitable for present homes because there is no requirement of installing wires in the walls, since it is a very expensive task. Wireless systems are also in progress to integrate the Wi-Fi networks. It means that they can be easily used with any type of home networks at present time. Wireless systems are based on different frequencies, which makes them incompatible with the open networks. It is very important phenomenon so it should have to be checked when installation takes place.

This project is also a form of wireless system, where we will be modifying the traditional wiring system by using Ethernet cables.

4.2. Advantages of automated home

Automated Home is defined across a large number of functions, which include, controlling of the home security system remotely by using the GSM modem to switch on or off the lights in the home. Generally, Automated Home System defined as to access control the home appliances remotely.

Automated Home Systems shower us with convenience by conserving the time and the effort performing domestic tasks; this system can also help in conserving energy as well as save the money by reducing the consumption of energy at home; and gives the client a peace of mind.

Following are a few benefits of an Automated Home system:

4.2.1. Convenience

One can enjoy the convenience of controlling many gadgets within the home remotely, or from any location. We can switch on or off the lights in the home without moving. Turn the home alarm system on/off from the cell phone just by a simple android application.

4.2.2. Energy Saving and Efficiency

Many Automated Home appliances are often called “smart” products, which help us manage our energy consumption. Some “smart” gadgets can also synchronize with the home appliances for communicating the real-time information of energy. This helps the home appliances to know about the right timings to operate.

4.2.3. Home Security

Automated Home also has security benefits. It can be checked any time from any place about the situation of the home. Real-time video feed can be taken via help of the IP cameras and can be observed whenever required on the android device. These IP cameras can also be applied at the entrance of the house. There are such Automated Home systems that will alert you via the android device of the owner of the house in case of emergencies.

4.3. Building Automation

It is described as the functions provided by a control system of the building. The BAS is also a type of strewn control system. This system is a programmed, smart electronic devices network, and it is intended to supervise, manage the motorized and electrical control systems in the building.

BAS core functionality keeps building climate within a specified range, lights rooms based on an occupancy schedule (in the absence of overt switches to the contrary), monitors performance and device failures in all systems, provides malfunction alarms to building engineering/maintenance staff and contractors. BAS reduce building energy and maintenance costs compared to a non-controlled building. Typically they are financed through energy and insurance savings, and other savings associated with pre-emptive maintenance and quick detection of issues. [17]

4.3.1. Lighting control

Both energy usage and comfort makes the lighting control a natural extension of designing BAS. Lighting control, being multi-disciplinary, is very complicated. Lighting design is mostly done by an interior designer while the power, circuiting and switching has to be done by an electrical engineer. Lighting control tends to follow these same paths with the design and installation falling into the electrical scope of work. [18]

Ideally the lighting control system would still be able to be designed and delivered as a part of the electrical system. But this can readily be integrated to BAS. Systems are now available that offer the ability to integrate using a variety of open protocols including XML, BACnet, Modbus and LonTalk. Designing a system to be integrated requires careful coordination between the BAS / Integration design and the lighting control design. This coordination extends through installation, commissioning and operations. Some of the common functions of an integrated lighting control system include:

- Coordinated scheduling of lights and HVAC loads. While this seems simplistic it has a giant potential for energy savings.
- Control and monitoring of advanced lighting control functions such as daylight harvesting. The ability to take advantage of natural light from windows and skylights has a tremendous potential to both save energy and improve occupant comfort. Monitoring this through the BAS allows for improved tuning to get the most out of this often underutilized application.
- Lighting control provides for effective and inexpensive occupancy sensing. Information on occupancy from the lighting system can be readily used for scheduling and control in the BAS.
- For projects that are enrolled in a demand response program the ability to shed lighting load provides an effective solution with little occupant impact.
- New technologies with communicating ballasts offer the potential to gain valuable maintenance information including notification to replace bulbs.

4.3.2. Device control

Generally, the control of mechanical, electrical and plumbing (MEP) systems are deployed first while building automations. For example, the heating, ventilations and air conditioning (HVAC) systems is almost always controlled, including the control of its various pieces of equipment such as: [19]

- Boilers
- Chillers
- Roof-top Units (RTUs)
- Fan Coil Units (FCUs)

- Air Handling Units (AHUs)
- Heat Pump Units (HPUs)
- Variable Air Volume boxes (VAVs)

Lighting control is, likewise, low-hanging fruit for optimizing building performance. Other systems that are often controlled and/or brought under a complete automation system include:

- Power monitoring
- Security
- Close Circuit Video (CCTV)
- Card and Keypad access
- Fire alarm system
- Elevators/escalators
- Plumbing and water monitoring

CHAPTER NO.5

5. Smart Home

The term Smart Home creates the image of an environment in the mind that is capable of reacting “intelligently” by anticipating, predicting and taking decisions with a few signs of autonomy. There is a natural association between expectations from a natural perspective by the use of techniques from Artificial Intelligence (AI). The research related to the Smart Homes has, however, largely focused on the network and hardware oriented solutions. The Artificial Intelligence based techniques that promote intelligent behavior have not been examined to the same extent, although notable exceptions can be found.

Some of the functionalities offered by a Smart Home are related to economic and comfort related aspects for example, lighting and cooling/heating systems are turned automatic i.e., turning them on/off according to the inhabitant’s location/presence within the home. One of the most beneficial elements witnessed from the Smart Homes is the improvement in the independence in living lifestyles that such environments can offer. Most importantly, the people that can benefit from such an environment are those who suffer cognitive impairments. The introduction of such a technological support into the lives of such patients offers the potential for them to perform their daily activities which they would previously have performed with the external support. With such a radical change in the lifestyle, the person can remain within their own home without the need of institutionalization. [20]

The concept of smart home originated from the concept of Home Automation System. A Smart Home concentrates on making the simple home appliances intelligent enough to interact with the human beings through a medium, like a personal computer, laptop or a smart phone/tablet.

The topic of a home network has been discussed and implemented over various media such as power line, Ethernet and phone line. But now, due to the advancement in the wireless technology, several different methods have come into focus to take this revolutionary concept up a notch. These wireless technologies include GSM, WIFI, ZIGBEE and Bluetooth, each having their own unique application and specifications. So, while considering the ease of access to the WIFI technology in every household, it was chosen with its suitable capability. WIFI connection completely encompasses a house it is connected to and therefore, is the most logical choice to serve our purpose effectively.

Figure 4. Smart Home

5.1.Components of Smart Home:

A Smart Home comprises of the devices that can be controlled by the user via remote-control or a smart device. So, basically, all the appliances being controlled and the devices being used to control this system are collectively said to be the components of Smart Home.

It depends on the user as to how many and which devices are they planning on controlling through this automation system. However, in this project, we have opted for following components:

a) Personal Computer (PC)/Laptop

b) Smart Phone (android/iOS)

c) Lights

d) Fans

e) Air Conditioners

f) Uninterruptable Power Supply

5.2. Smart Devices:

According to the traditional definition, a Smart Device is an electronic device connected to other devices or networks through different protocols. These protocols can be NFC, WiFi or 3G, etc. this enables these devices to operate somewhat interactively or autonomously with either the user or other devices. Examples of such devices are smartphones like apple iphone or android phones/tablets. This term can also be used for the devices exhibiting some properties of ubiquitous computing. The devices being used as the Smart devices in this project are typical everyday electrical appliances such as lights, fans, ACs, etc.

Each device is connected to its own individual microcontroller unit module that makes it intelligent enough to compute the data being received from the user communicate as desired by the user. This microcontroller unit groups the appliances on the network as required and thus provides the users with an opportunity to customize their house however they require.

Most of the network enabled appliances can be easily integrated with the network IC and this method is especially popular in the appliances such as PCs, printers, etc. This method is also applicable to the entertainment appliances such as audio and video devices in order to achieve the networking function and thus enabling us to design new services. These devices, however, need to be networked, distributed and transparently accessible to the users. The human-computer interaction with these devices must be hidden to a certain degree and the devices should also exhibit context awareness of their environment and time so that they are capable of optimizing their performance as customized by the users.

The type of environment that is being incorporated in this project is called the “Humans Environment”. Here, the users, either individually or collectively, form a smart environment for the appliances. Users, however, can make use of smart devices such as mobile phones or use the surface mounted devices and embedded devices.

So, according to what has been done in the said project, the user sends a command via smart phone or PC and the device, in response to that command operates as desired by the user. Not only this, but the devices are also capable of communicating with the user by sending the data diagnostics in the real time to the user as per requirement.

5.3. Connectivity of Smart Devices:

The figure given in this section illustrates the overall control function of the system. The said system is directly installed along with the conventional electrical switching and wiring. The wireless connection enabled system communicates with the GUI on either the user's PC or smartphone. The target home appliances are controlled individually via microcontroller units especially designed for this purpose.

The Smart Home has three logical levels, as seen in figure given below. The top-most layer depicts inputs from the user via PC or an Android application. The next layer comprises of the control unit which, after reading this input, performs the actions depending on the input values and the control program as specified. In this project ATMEGA88 microcontroller acts the control layer. Finally, the control unit outputs the commands to the actuator layer. The actuator layer is responsible for forming and sending commands to the real-world systems such as lights, fans, Air conditioners etc. For our system, we will be designing our own control module.

This connectivity from the user to the device will be done wirelessly in order to render the project more applicable and appealing to the users. The input layer may consist of any smart device, the one the respective user finds himself comfortable with to make the project all the more user-friendly.

5.4.Energy saving in Smart Home:

Smart Home not only targets on making lives of its users safer but more convenient and entertaining at the same time. Also, it helps you in saving heaps of money that would have otherwise been spent on unnecessary wastage of electric power. [21]

An average household can cut a third of its energy bill simply by starting the use of energy efficient appliances and lighting. From lighting to Air Conditioning, Smart Home offers a variety of products that clients can purchase and begin saving on the energy costs today.

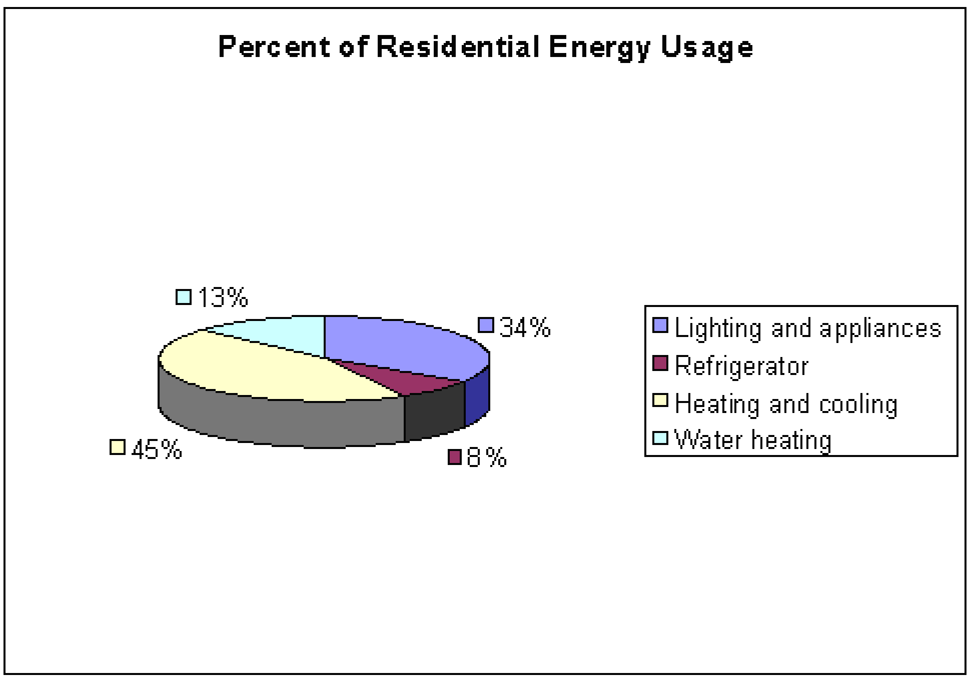


Figure 5. Pie Chart for Residential Energy Usage

As it can be seen from the above figure, almost half of a typical utility bill consists of the cost of energy expended on heating and cooling in a typical house. A Smart Home offers the installation of cooling system that can be managed by the user as per their requirement. They don’t have to get up to check up on the thermostat of an air conditioner working in a different part of the house, they can simply use their smart phone to log in to their home data base, provided by an app specific for this purpose and check on the status of the air conditioner and thus not only switch the devices but also adjust the thermostat as per their wishes. Moreover, the user doesn’t even have to be inside their home to switch the Smart Devices, they simply have to be in the vicinity of an internet connection to just log in to their home data base to work the magic.

While, 34% of a typical household bill goes towards lights and appliances, a full 20% of this is actually spent on just lights. In order to reduce the energy usage, a Smart Home enable the users to not only switch the lights from their android device but also, allows them to customize the automatic switching of the lights. Also, light dimmers can be put to use to reduce the wattage and output to save energy. In the similar way, the speed of the fans and their switching can be controlled accordingly.

5.5. Architecture of Smart Home:

The most fundamental requirement in a Smart Home is a home network. The specifications of the networks for grouping of the appliances can be different as the characteristics of the groups differ from each other. A high speed network is required, due to the ever increasing demand of sharing HD picture quality and voice in entertainment appliances, in order to connect them with the information appliances. The cost of this type of network, however, can be a very critical issue and therefore, it makes it hard for the current available technologies to meet every aspect of the Smart Home network.

The Smart Home network has to cover several application fields, having different characteristics. Therefore, an integrated solution to combine the pros from all the different networks was put forward. This solution will enable us to meet the high speed demands and will also reduce the product development cycle within one system. The figure on the next page emphasizes the solution for the integration of different types of home networks, in addition to the value-added services and corresponding standards.

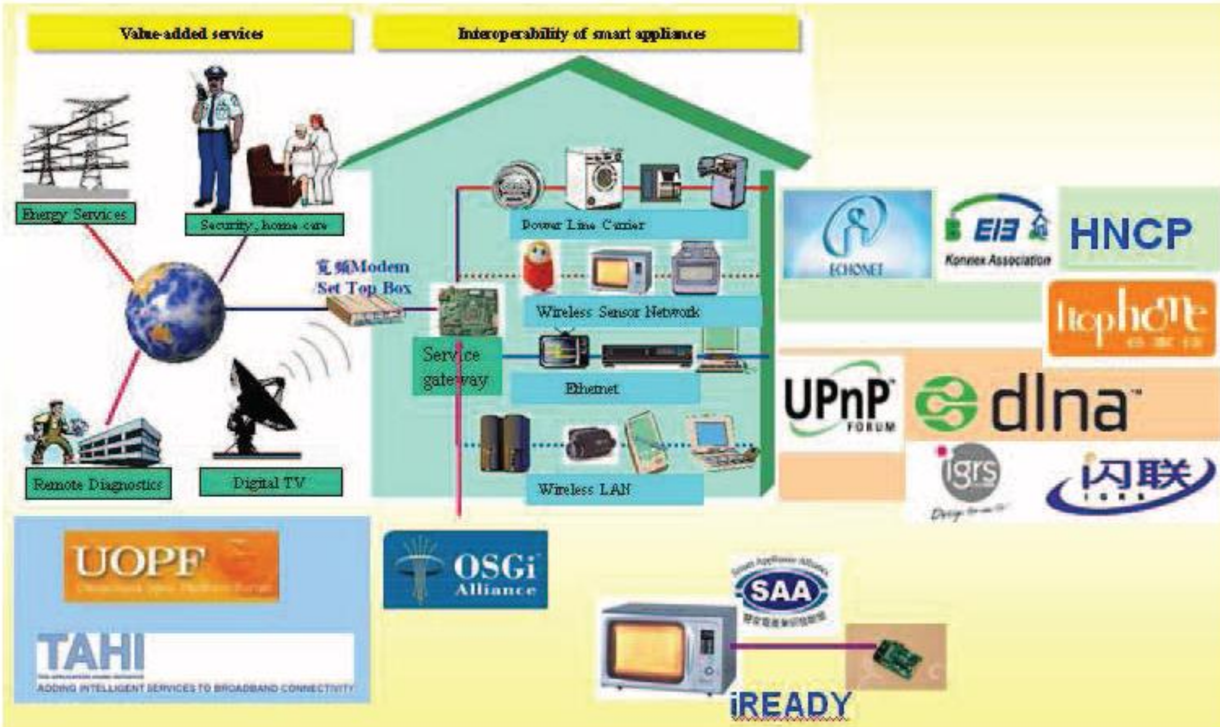


Figure 6. Smart Home Architecture

CHAPTER NO.6

6. Design and Development

What we intend to do is to send a packet over IP network. Microcontroller unit receives the packet and performs the desired functions.

The Home Automation System has three logical levels, as seen in Figure 2. The top-most layer depicts inputs from the user via PC or an Android application. The next layer comprises of the control unit which reads the inputs, performs the actions depending on the values of these inputs and the control program as specified by the user. In our project ATMEGA88 microcontroller acts the control layer. Finally, the control unit outputs the commands to the actuator layer. The actuator layer is responsible for forming and sending commands to the real-world systems such as lights, fans, Air conditioners etc. For our system, we will be designing our own control module. The details and corresponding subcomponents of each of these layers are described in the remainder of this section. [22]

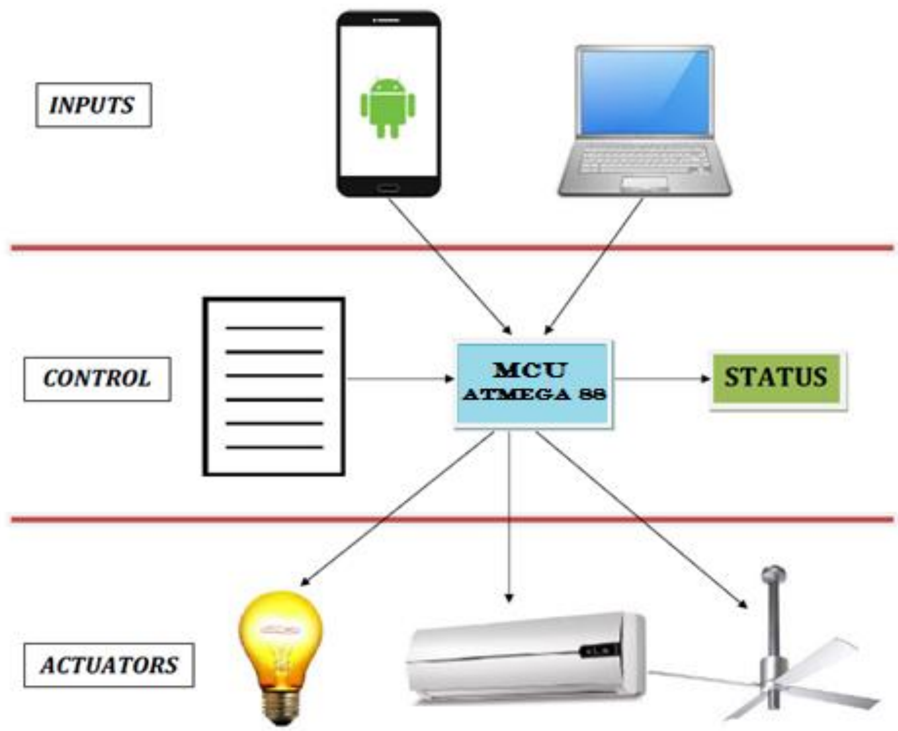


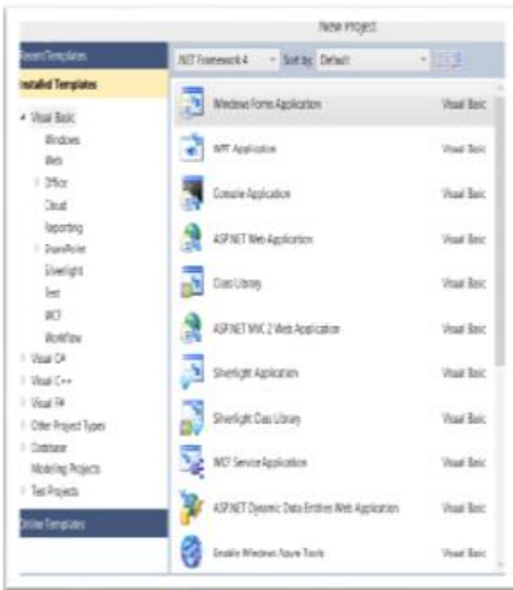
Figure 7. Three Layer Design

6.1. Input layer

Input layer comprises of two software applications which are as follows:

6.1.1. Visual basic application for PC

The first phase of our project is to send the data from a PC to the controller via Ethernet. A user is required to input the data in a computer application which will forward it to the switching module. Therefore we have developed an application on visual basic which serves as a platform that enables a user to interact with the hardware part of the system in the run time and therefore control the devices as desired.



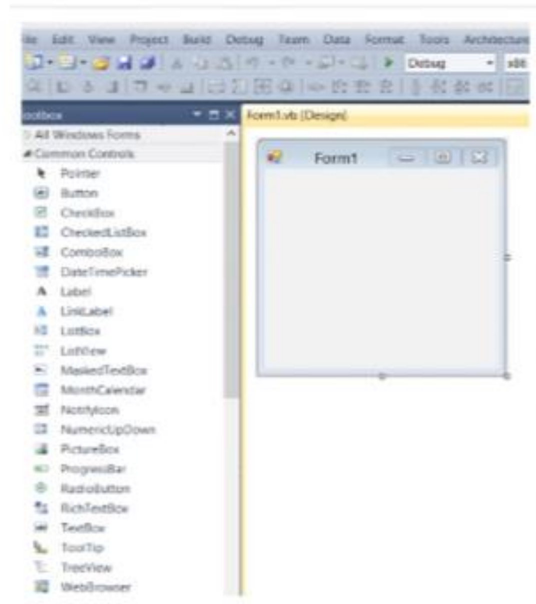


Figure 8. VB app initiation

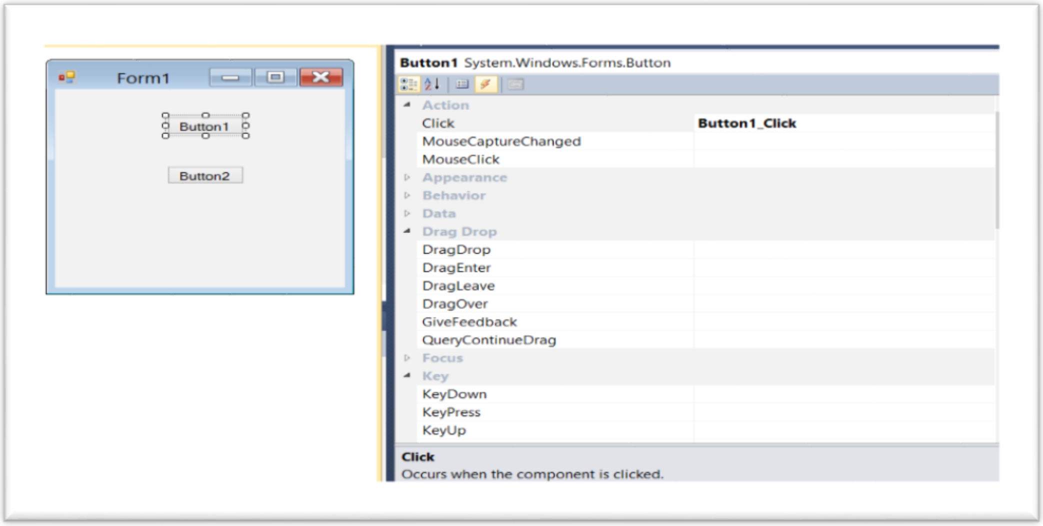


Figure 9. VB Click Button

We started off with a simple application “HELLO WORLD”. This application consists of a simple button that displays a string when the button is pressed/clicked.

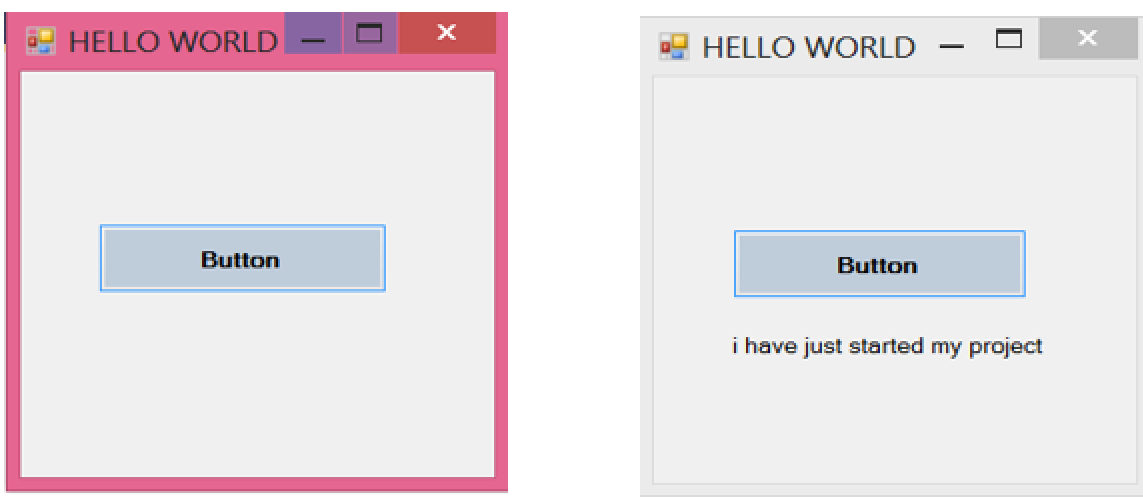


Figure 10. Hello World app

We have seen that the strings can be displayed by clicking the button in the same manner any data information can be written by just clicking the button. Moving towards the communication part, first, we interfaced serial port with our application to which the data can be sent. Interfacing was done by importing the I/O ports. The port is opened before sending any data. For serial communication through VB baud rate needs to be defined before opening the port. The Baud Rate can be defined as “bits per second”. The transferred bits include the start bit, the data bits, the parity bit and the stop bits.

When “send” button is pressed whatever the data is present in the text box is written on the serial port. The application was first tested using HyperTerminal and after that by null modem and then it was interfaced with the controller. Using serial communication, we connected LEDs to the controller which was interfaced to the microcontroller.

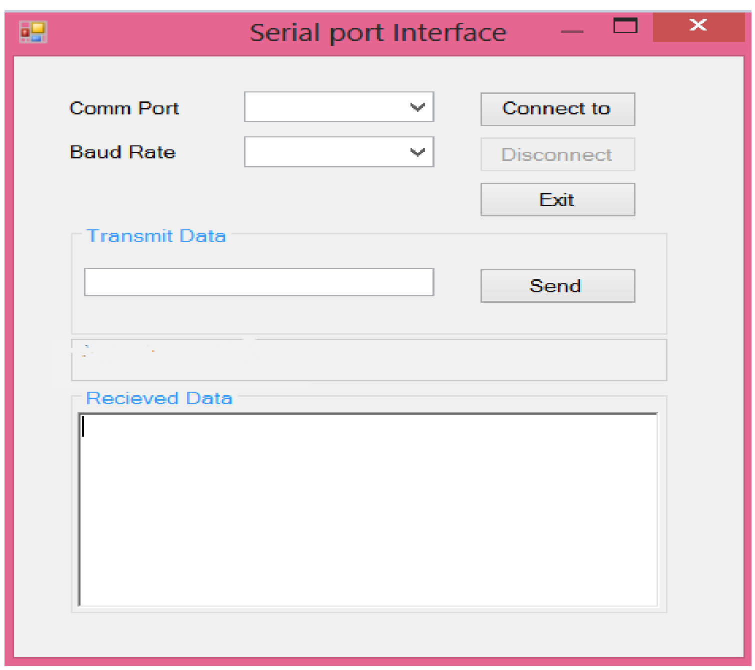


Figure 11. Serial Port Interface

Figure 12. Serial Control

By clicking the "connect" button serial port is opened and is made ready for communication. When LED1 is turned ON, a value or string, say letter 'a' is written to the serial port. Serial port sends this data to the microcontroller. Microcontroller then decodes the information and turns on LED. After successfully achieving serial communication we started working on Ethernet communication. We developed an application that set data on a specific IP address set by the user by establishing UDP client we tested our application using UDP client test tool application on the other end.

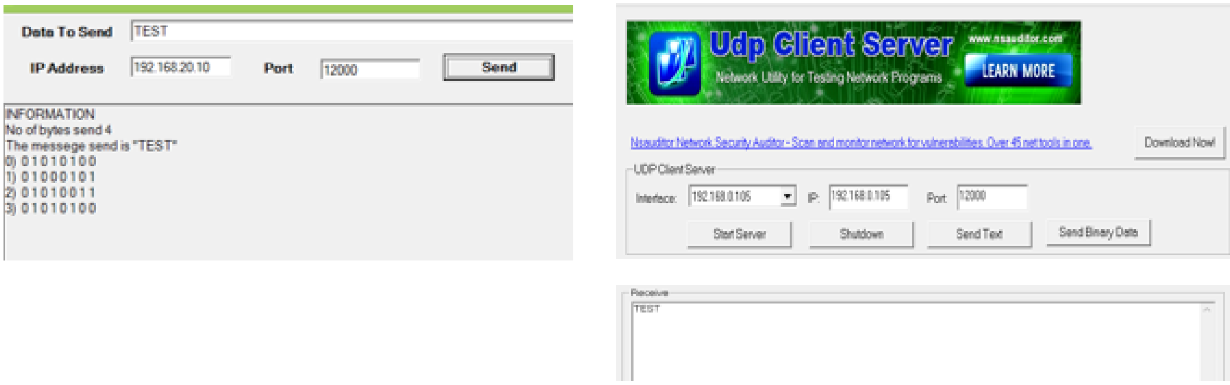


Figure 13. UDP Client Server

The final VB application that we have developed provides with the ability to control the devices on a single click. Each button is coded at the back end and has a unique functionality. Whenever a button is pressed to perform the desired action, UDP packet containing the relevant switching info is sent over the IP network. ENC modules are used at the control layer to receive these packets. The information is then passed to controller which decodes it and passes it to actuator layer. [23]

Figure 14. Final VB Application

6.1.2. Android application development

Following is the GUI application developed by us for an android device for controlling the respective devices.

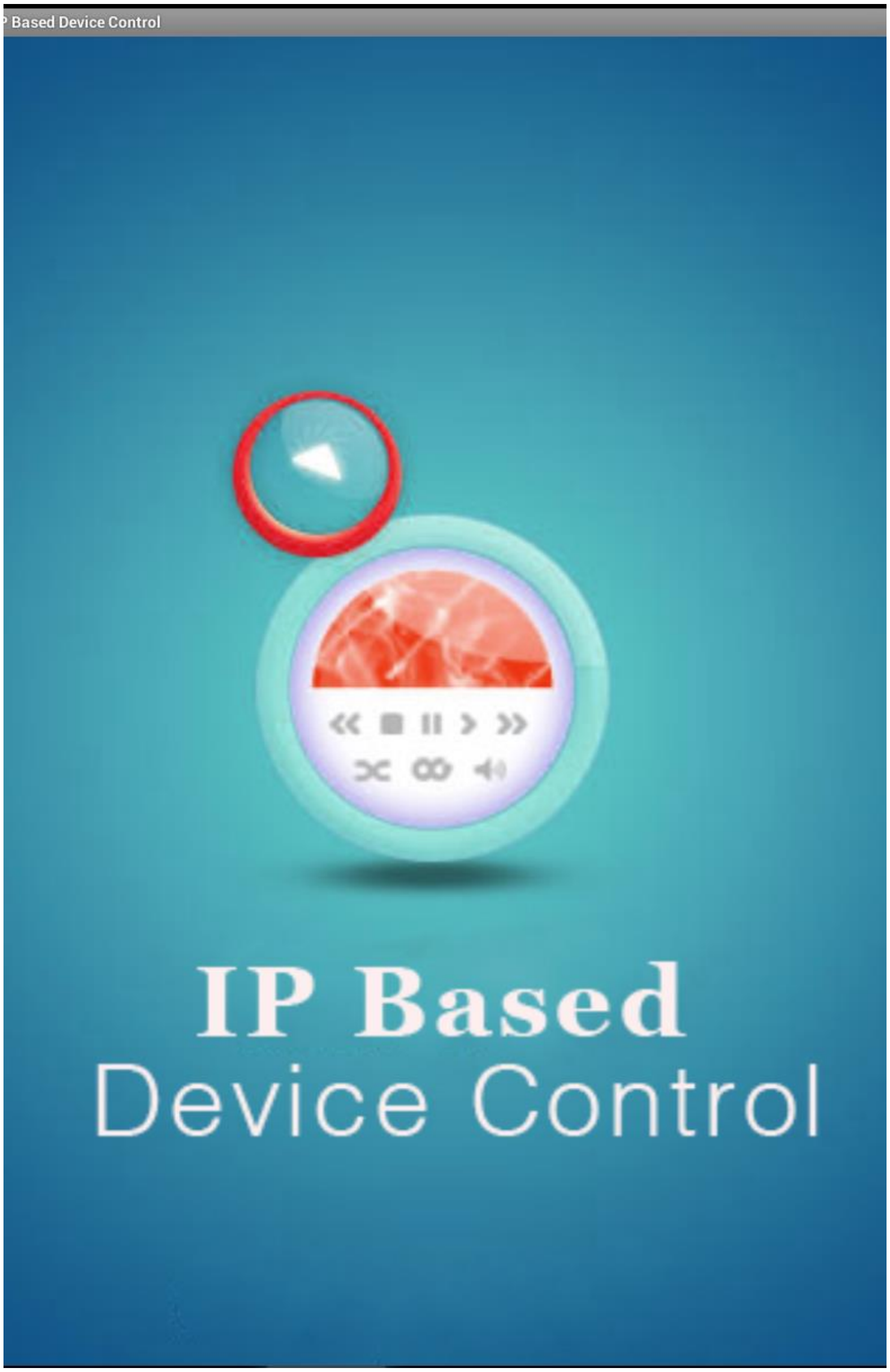


Figure 15. Android App GUI (a)

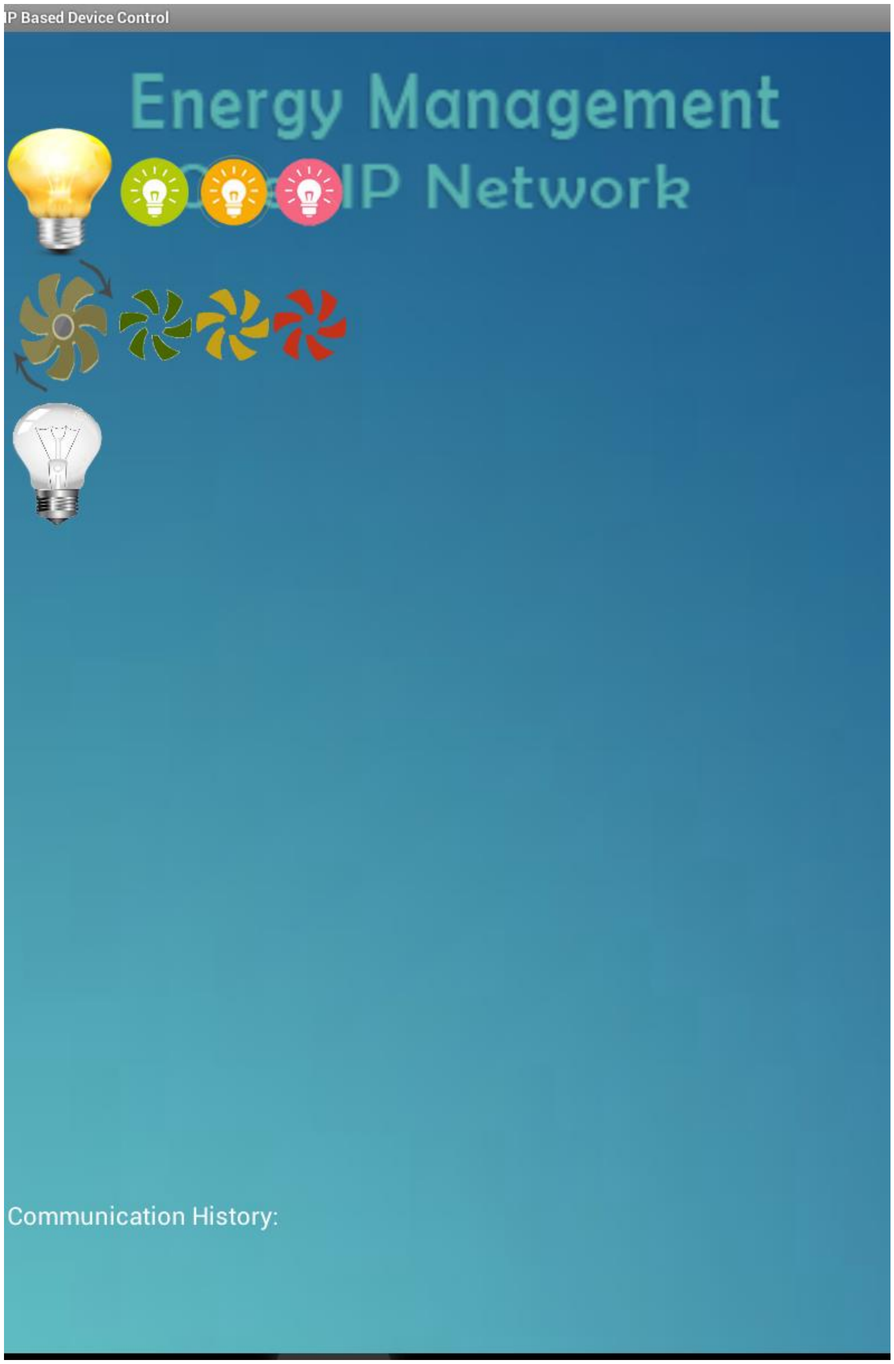


Figure 16. Android App GUI (b)

6.2. Control layer

Here in the control layer, we have to deal with two types of communication protocols, i.e. Ethernet and serial communication. The brief description is as under:

6.2.1. Ethernet communication

Ethernet topology, which is based on bus and bus-star physical configurations, is currently the most frequently configured LAN network architecture. A bus is a common pathway (usually copper wire or fiber cable) between multiple devices such as computers. It is often used as a backbone between devices. It is a technology that has been evolving for more than 25 years and is still evolving to meet the ever increasing and changing needs of the internetworking community. [24]

Ethernet is a passive, contention-based broadcast technology that uses baseband signaling. Baseband signaling uses the entire bandwidth of a cable for a single transmission. Only one signal can be transmitted at a time and every device on the shared network hears broadcast transmissions. Passive technology means that there is no one device controlling the network. Contention-based means that every device must compete with every other device for access to the shared network. In other words, devices take turns. They can transmit only when no other device is transmitting.

6.2.2. Serial communication

In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

Serial communication is used for all long-haul communication and most computer networks, where the cost of cable and synchronization difficulties make parallel communication impractical. Serial computer buses are becoming more common even at shorter distances, as improved signal integrity and transmission speeds in newer serial technologies have begun to outweigh the parallel bus's advantage of simplicity and to outstrip its disadvantages (clock skew, interconnect density). [25]

6.2.3. ATMEGA88

The microcontroller that we have put to use in our project is ATMEGA88. The purpose of the control would be to receive the input from the upper layer, process it and send the desired command the actuator layer. The microcontroller acts as perfect control system for us. [26]

For the first phase of our project we want our controller to receive an IP control packet from a PC or an Android device. The control IP packet will not only be containing the information about which device to turn on but also the state of that device. The microcontroller will be programmed such that it will be able to process the data in the packet and then send the correct information to the lower layer i.e. the actuator layer.

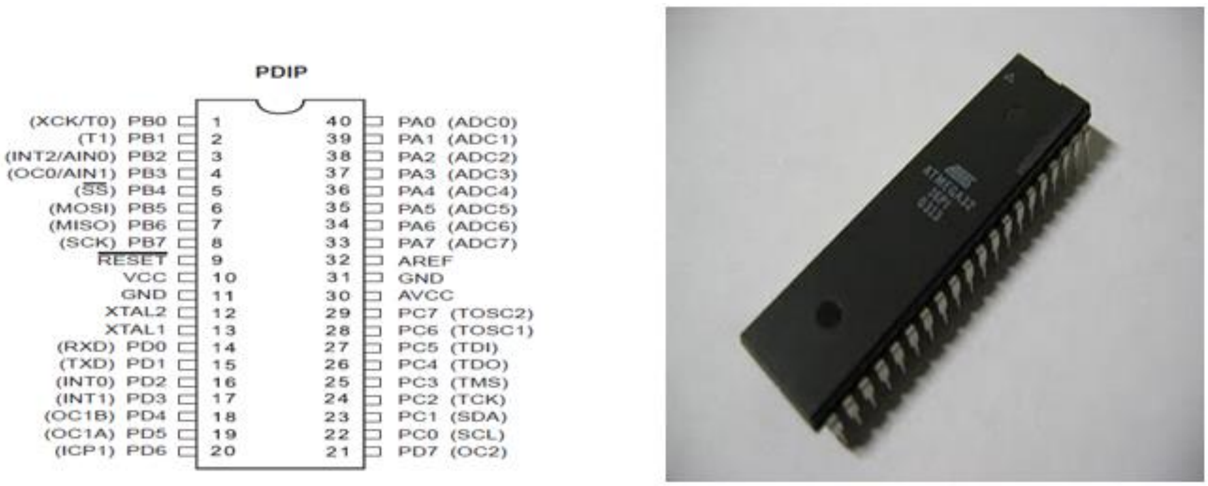


Figure 17 ATMEGA88 Pin Configuration

6.2.4. ENC28J60

Since the problem with this is that the microcontroller cannot be directly interfaced with Ethernet because the microcontrollers can only understand serial communication. Therefore, we need to interface an Ethernet -to-serial convertor chip for interfacing the controller with Ethernet. The IC that we are using for this purpose is ENC28J60, which is developed by Microchip. [27]

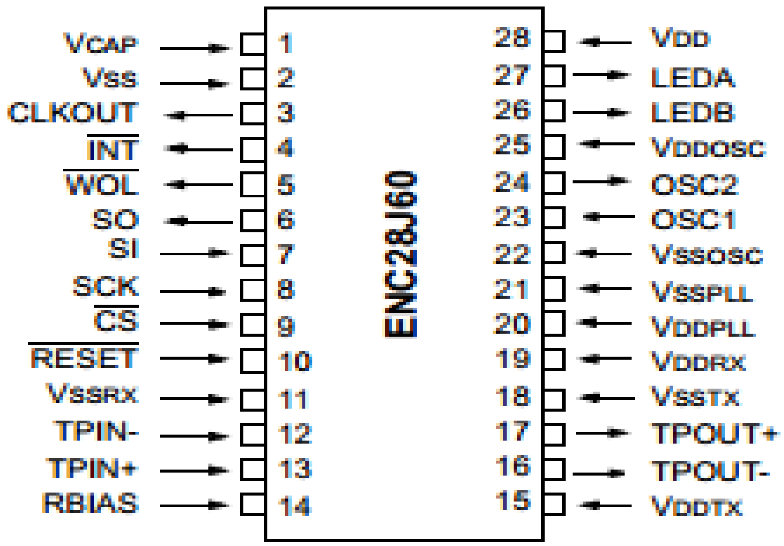


Figure 18. ENC28J60 Pin Configuration

The figure shows the pin configurations of the Ethernet to serial convertor IC. However this needs to be interfaced with the Mag Jack. To accomplish that the circuit shown below in the figure needs be implemented. When the circuit is ready and functioning it is then interfaced with microcontroller using the Serial Peripheral Interface or SPI.

This type of communication requires one chip to act as master while the other as slave.

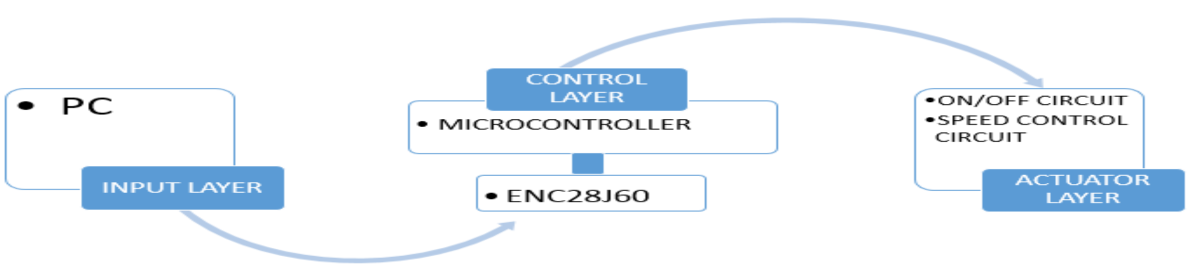


Figure 19. Interfacing Flow Chart

SPI is a standard synchronous high speed interface that operates in full duplex and can operate with one master device and one or more slave devices.

Before ensuring the successful communication through SPI, both the master and slave must agree on some clock signal settings.

In an AVR, four signals (pins) are used for the SPI: MISO, MOSI, SCK and SS' (SS' means SS complemented). Here is the brief description of the function of each signal: [28]

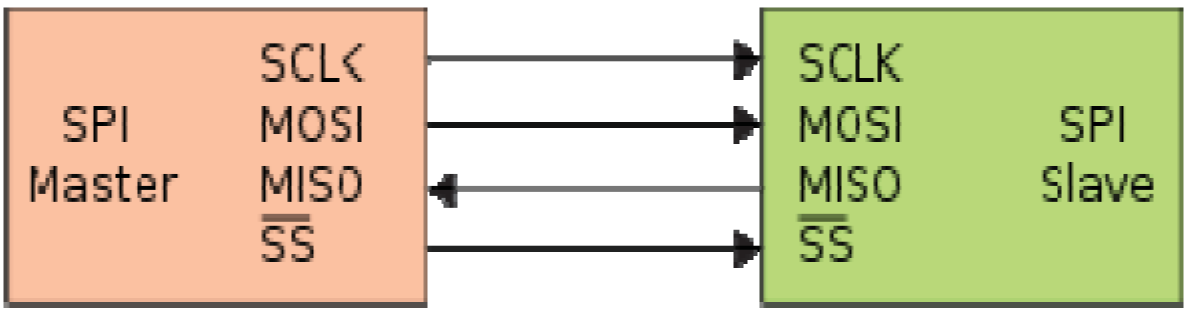


Figure 20. SPI Master Slave Concept

MISO (Master In Slave Out): the input of the Master's shift register, and the output of the Slave's shift register.

MOSI (Master Out Slave In): the output of the Master's shift register, and input of the Slave's shift register

SCK (Serial Clock): In the Master, this is the output of the clock generator. In the Slave, it is the input clock signal.

SS' (Slave Select): since in an SPI setup you can have several slaves at the same time, we need a way to select which slave we want to communicate to, this is what SS' is used for.

If SS’ is held state, all slave SPI pins are normal inputs, and will not receive incoming SPI data. On the other hand, if SS’ is held in a low state, the SPI is activated. The software of the Master must control the SS’ –line of each Slave.

Following is the circuit diagram for the ENC28J60 Mag Jack.

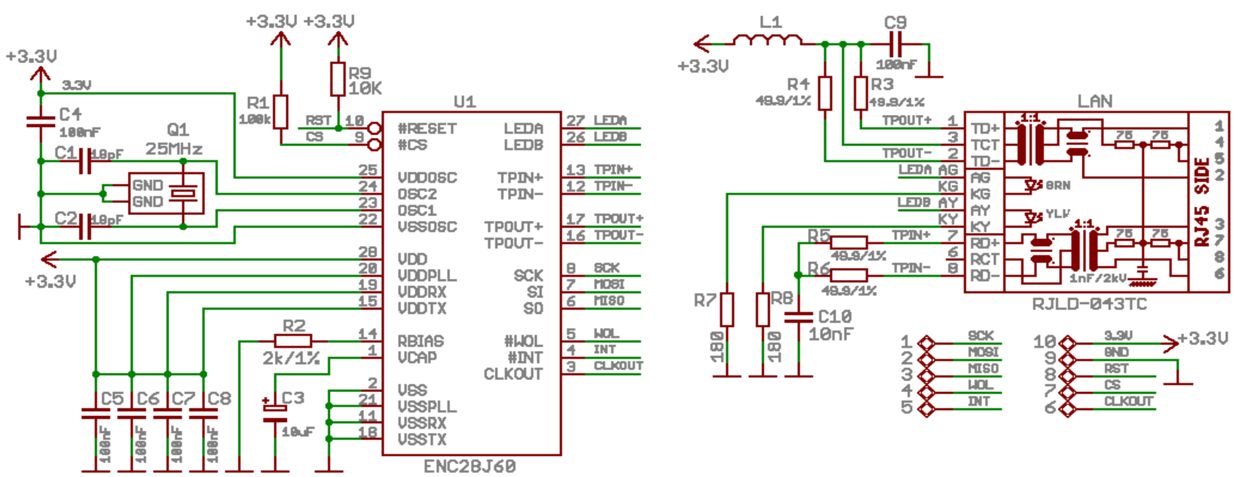


Figure 21. Interfacing via ENC28J60

Our first task after establishing all the above facts and measures, we were able to send a UDP packet from a PC and the microcontroller unit turned on the LED.

The simulation for which is as under:

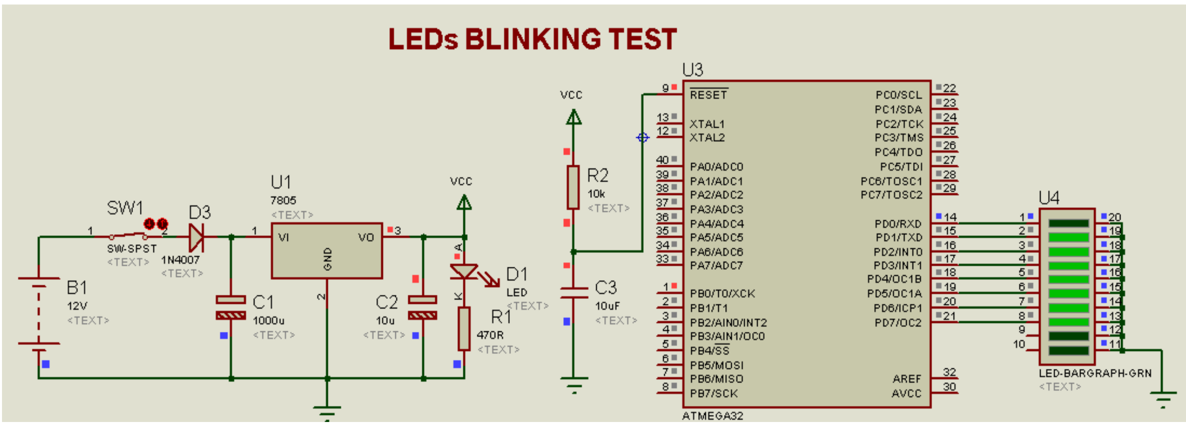


Figure 22. LED Blinking Test

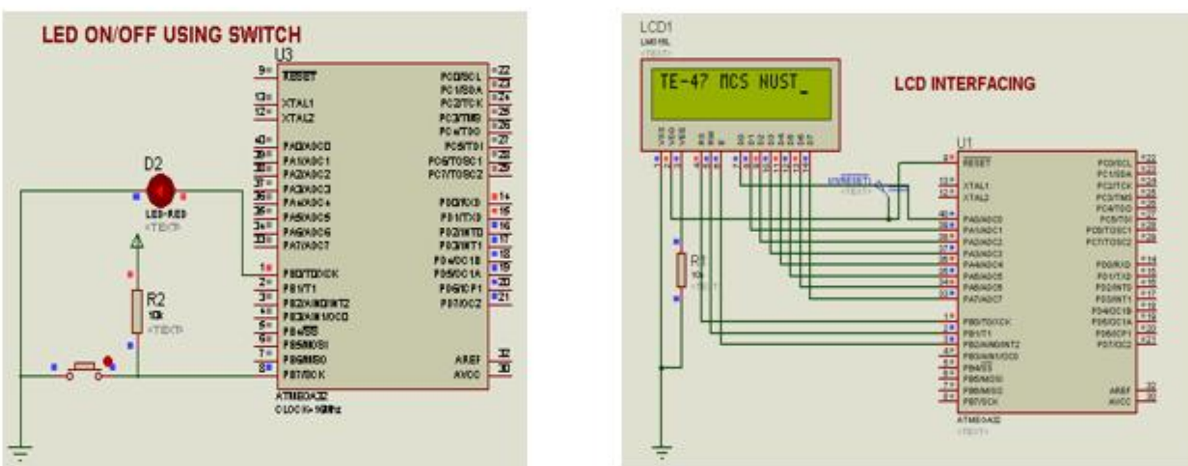


Figure 23. LED On/Off Using Relay

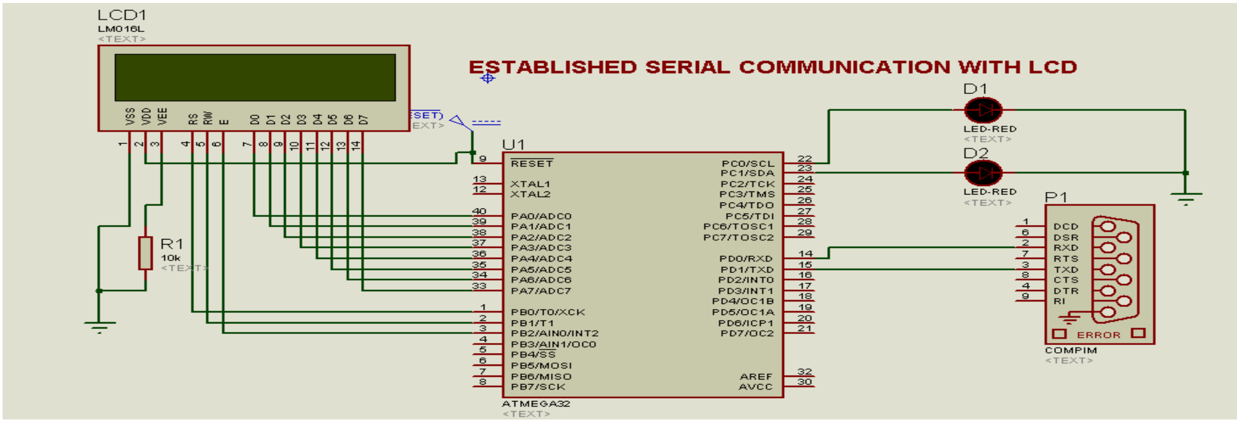


Figure 24. Serial Communication Established

6.3. Actuator Layer

The actuator layer comprises of all the end devices, including light, fan and an air conditioner.

6.3.1. Mechanical Relay for switching AC devices

A relay is used for mechanical switching of the devices. We are using mechanical relay for the switching purpose and electronic relays for speed control. The mechanical relay model, used for switching of devices is shown in the figure below. [29]

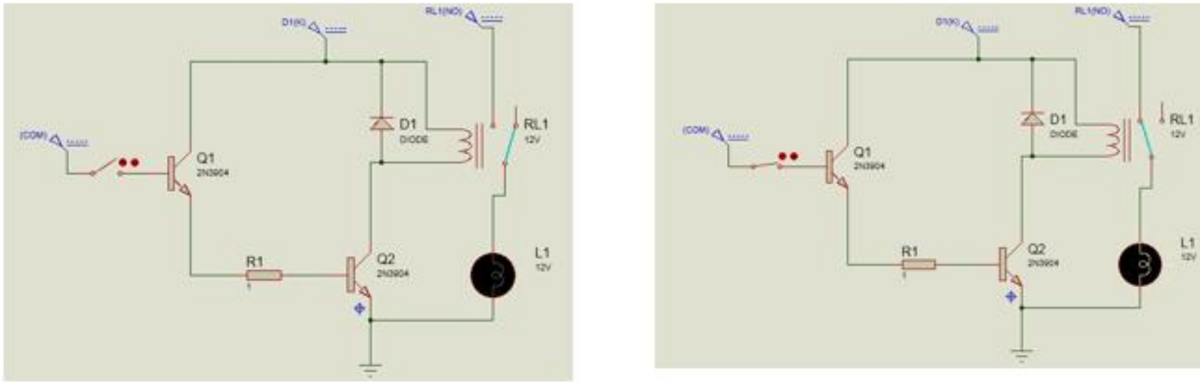


Figure 25. Mechanical Relay for Switching

6.3.2. Electronic Relays for AC speed control

The speed control circuit is similar to an electronic relay. High frequency switching is required here, so we prefer not to use mechanical relay for such application as electronic relays have many advantages over mechanical relays.

6.3.3. Zero crossing detector

To detect the zero crossings of main power supply, we have designed a circuit, including a bridge rectifier and an opto-coupler. This circuit detects the zero crossings and feed in the output to microcontroller. [30] The circuit diagram of the zero crossing detector is shown in figure below.

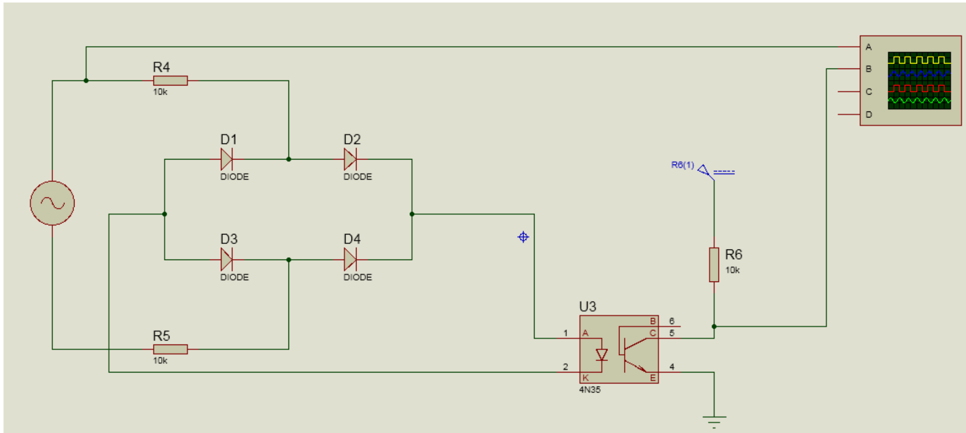


Figure 26. Zero Crossing Detector

6.3.4. TRIAC circuit for AC power control

The speed of the fan is controlled by changing the firing angle or increasing the delay before triggering the TRIAC after a zero crossing is detected at the main power supply. We can vary the delay for triggering the TRIAC to operate it on multiple speeds. For maximum speed, TRIAC is triggered as soon as the zero crossing is detected. And similarly for the minimum speed, TRIAC is triggered just before the next zero crossing is detected. TRIAC, from Triode for Alternating. The circuit is shown in the figure below.

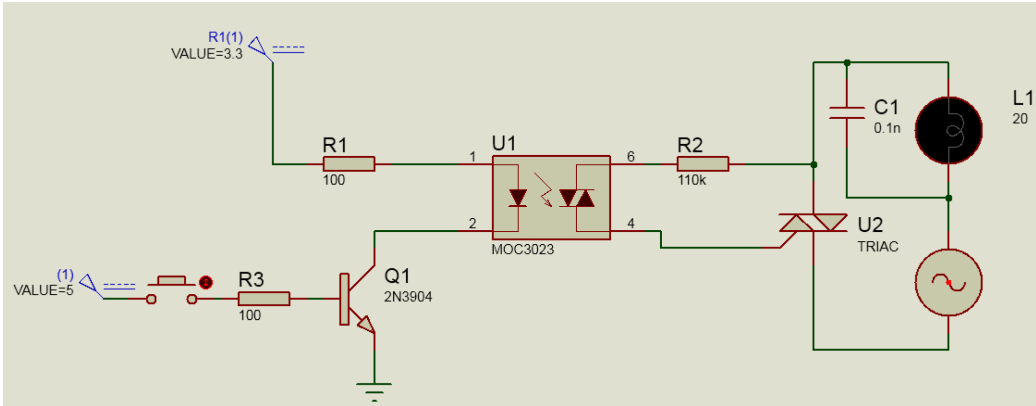


Figure 27. TRIAC Circuit

6.3.5. AC speed control circuit using MOC3023, tested using a bulb

For testing the speed control circuit along with the zero crossing detector, we integrated both the circuits and provided them with the common main supply as in the figure shown on the next page.

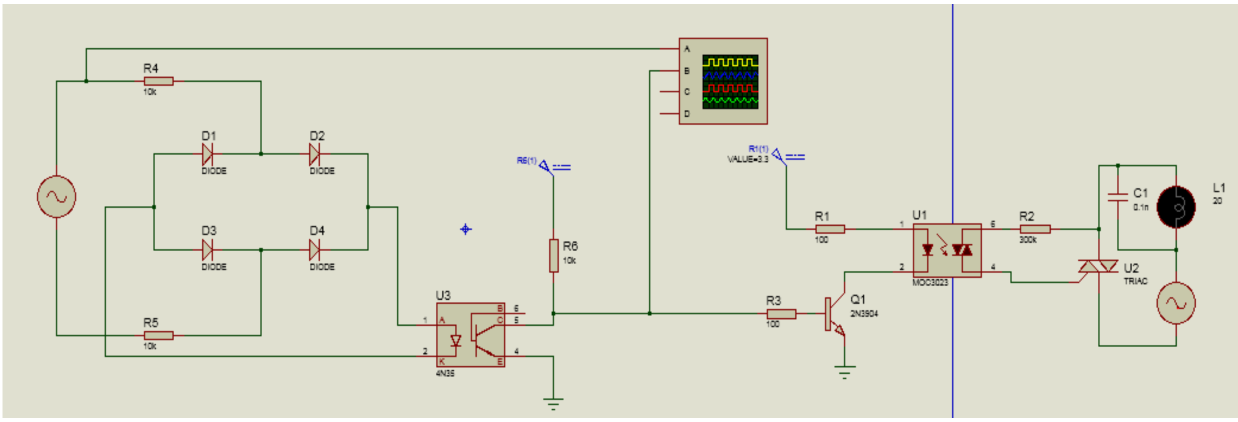


Figure 28. AC Speed Control Circuit

The output of the zero crossing detector is fed in to the speed controller circuit. When the zero crossing is detected, TRIAC is triggered and bulb is turned on.

The input and the output of the circuit is observed from the oscilloscope and shown in the figure below. From the figure, we can observe that at each zero crossing, we get a high value at the output. This output is further used to control the microcontroller and the speed control circuit. In order to control the intensity of the bulb, this zero crossing detector plays a major role.

Since the AC wave crosses the zero twice in one time period, the Zero Crossing Detector gives rise to a signal with twice the frequency as that of the originally fed signal to the circuit.

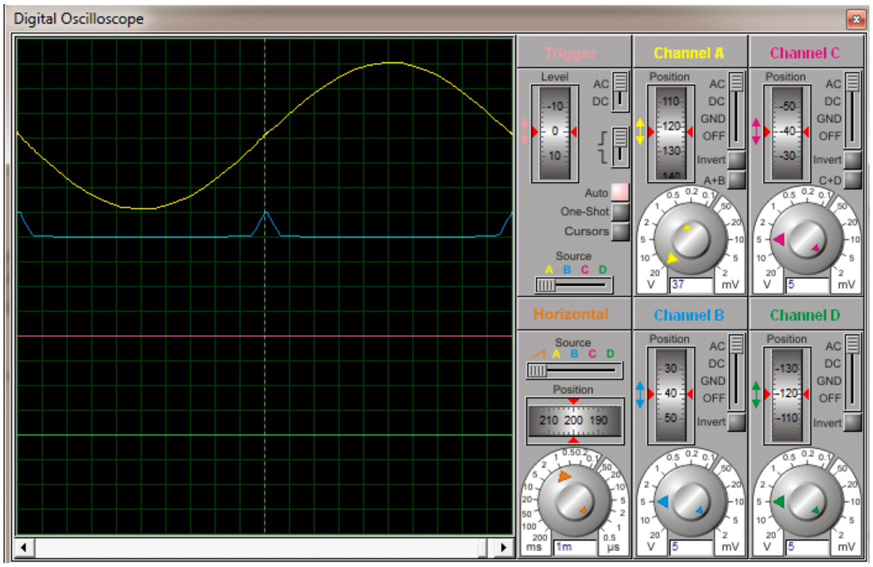


Figure 29. ZCD Response wave

6.3.6. Pulse Width Modulation and intensity control circuit

One of the qualities of the microcontroller is to generate constant waves via few special pins. The frequency and the duty cycle of these waves can be easily controlled. Taking an advantage of this factor, these waves are made to synchronize with the TRIAC signals. The intensity of the bulb and speed of the fan is controlled by the microcontroller unit by giving an appropriate delay to these waves in synchronization with the TRIAC pulses. This phenomenon is termed as PWM (Pulse Width Modulation) and the combination of Zero Crossing Detector and TRIAC circuit is called the intensity control circuit.

6.3.7. WiFi module interfacing of the circuit

After interfacing of the system with the Ethernet module, we move on to the next phase of our project. Here, rather than an Ethernet module, we use a WiFi module and interface it with the system in order to give the users a wireless access to the devices.

The WiFi module that we have used in the project is from SHUNCOM, and the module is shown in the figure.

Figure 30. WiFi Module

The first step is the wireless network choosing, since the module that we have put in to use supports two types of wireless networks, i.e. Infra and ADHOC. Each of these networks has its own technical specifications, the detail of which is as follows: [31]

a) Infra Network:

This is created by an access point (AP) with the access to station (STA) into it. All of the communication is transferred by AP since it is the network center. Following are three features offered by this network:

1. Communication range by the wireless router is expandable
2. Dynamic allocation of the IP addresses by the DHCP of the router
3. In order to protect the data, WPA2 encryption can be set

b) ADHOC Network:

For this type of network, at least two STAs are required, the AP however, don’t exist. Following are the features offered by this network:

1. It does not need a router
2. Multiple modules can be supported
3. It needs to set a static IP

This mode is low cost and easy to use. The module can be directly connected to the laptop or a mobile phone and does not depend on the WiFi router.

This module offers and supports multiple modes wireless encryption in order to make sure that the user data is safe. Moreover, this module is for the UART transparent data transmission. A simple configuration tool is provided for convenience of the users. A chip flash is used to save all the parameters after the configuration until a new configuration takes place. In case the PC and the mobile phone support WiFi and ADHOC, configuration can take place via web. When the power is supplied for the first time, point to point network (SHUNCOM_hoc) appears in the network list. Then after this, the module is configured to the computer and connected to the circuit instead of the Ethernet module and we get the same results as before. The visual concept of this topic is shown in the figure below:

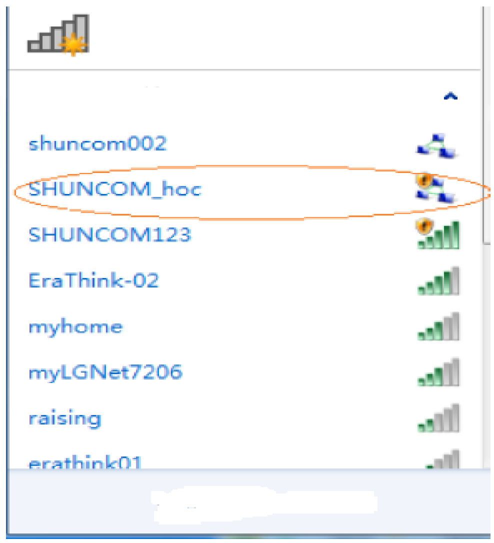


Figure 31. SHUNHOM_hoc

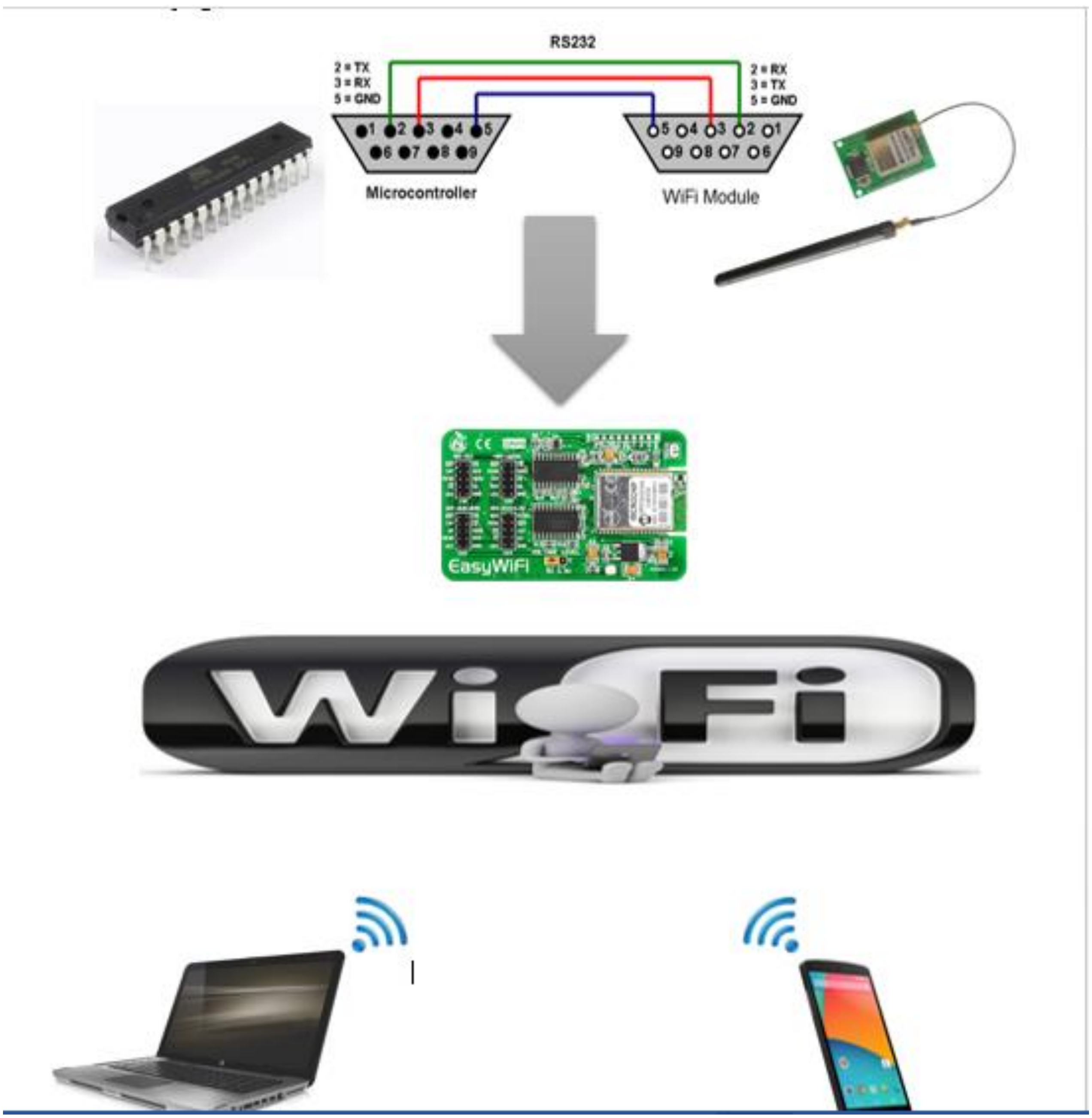


Figure 32. Final view After Interfacing

7. Analysis and evaluation

With the implementation of the above mentioned system, we were successful in controlling not only the switching of the light bulbs ON/OFF but also, controlling the intensity of the light emitted but the light bulb.

The wave produced by the microcontroller is synchronized with that of the output wave of the TRIAC circuit as shown in the following figure.

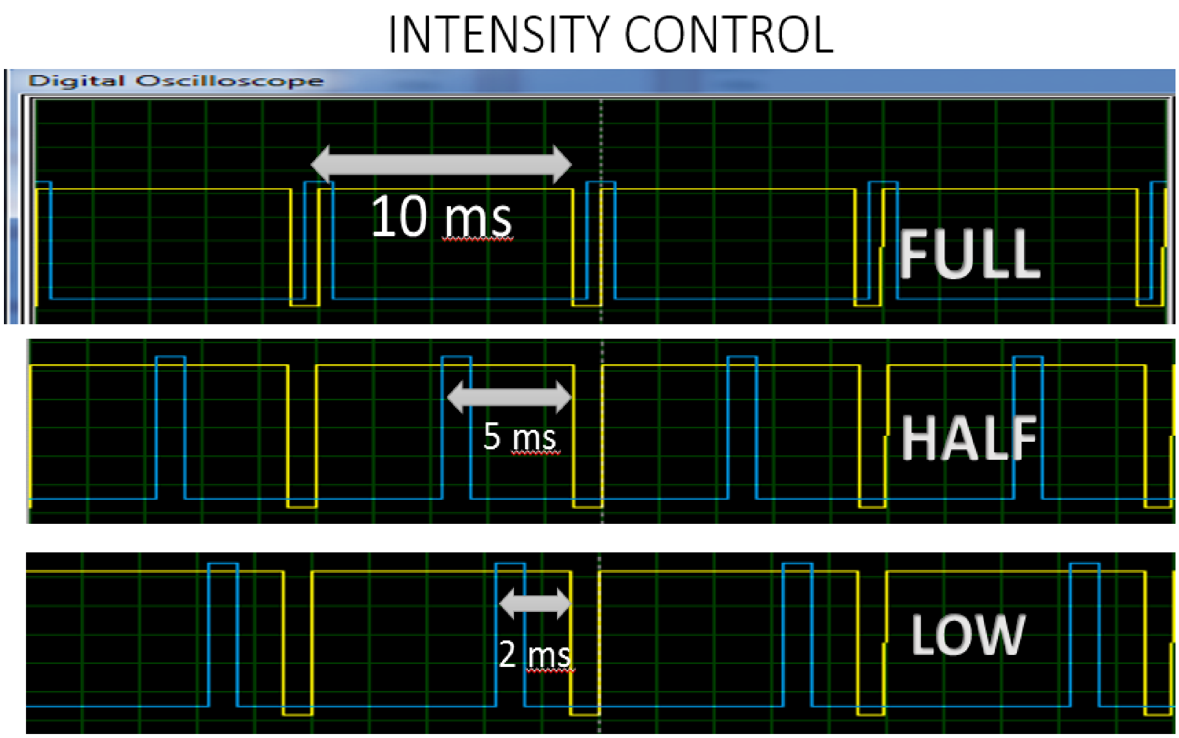


Figure 33. Intensity Control (a)

Since the output wave from the TRIAC circuit is responsible for the intensity of the light bulb, as soon as the pulse drops to zero, the bulb switches OFF and switches back ON as soon as the pulse steps up again, as a result, the bulb keeps on flickering. By synchronizing the signal generated by the microcontroller with this output as soon as the pulse drops to zero, the microcontroller switches the bulb ON again, leaving no chance for the bulb to flicker and thus, we get full intensity from the light bulb.

Similarly, the medium and high intensities of the light bulb are achieved by difference in synchronization of these two output waves as shown in above figure.

The result of the above process is shown as follows:

Light bulb at minimum intensity | Light bulb at 50% intensity | Light bulb at 100% intensity

Figure 34. Intensity Control (b)

Following circuit depicts the circuit we used to interface the microcontroller unit with the intensity control circuit:

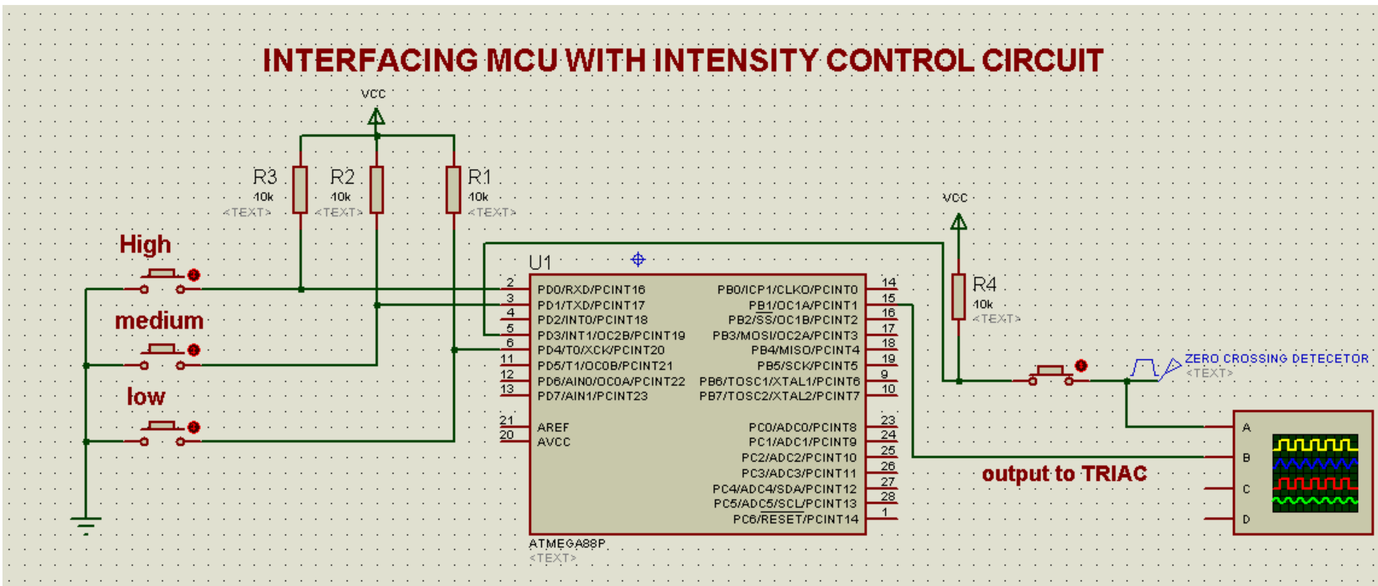


Figure 35. Interfacing MCU with Intensity Control Circuit

Following is a figure depicting the circuit ping of the project through a PC before using the PC app to send a packet of data to the devices, i.e. light bulb and fan. The PC app has already been shown in the figure in section 6.1.3.

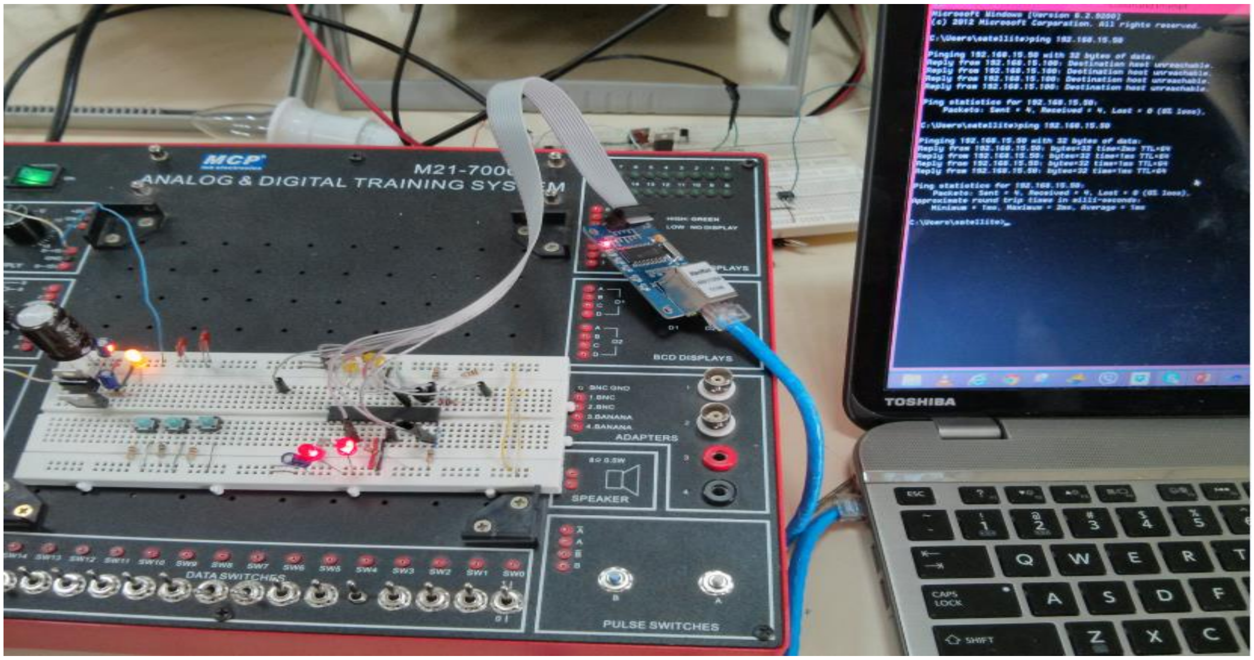


Figure 36. Circuit Ping

After the successful communication with the device using Ethernet module, we tested out project wirelessly via not only the PC but also via an android device, which was, in turn just as successful as the wired system.

CHAPTER NO.8

8. Recommendations for future work

The work on this technology does not end here, this concept is evergreen and ever-developing. Later on, after the deployment of this technology, smart devices, this project can be further extended to the concept of smart UPS, smart home, smart meter, smart grid and many more since this is just the beginning of the application of the word ‘smart’ in this era. A model illustration of this concept is as follows:

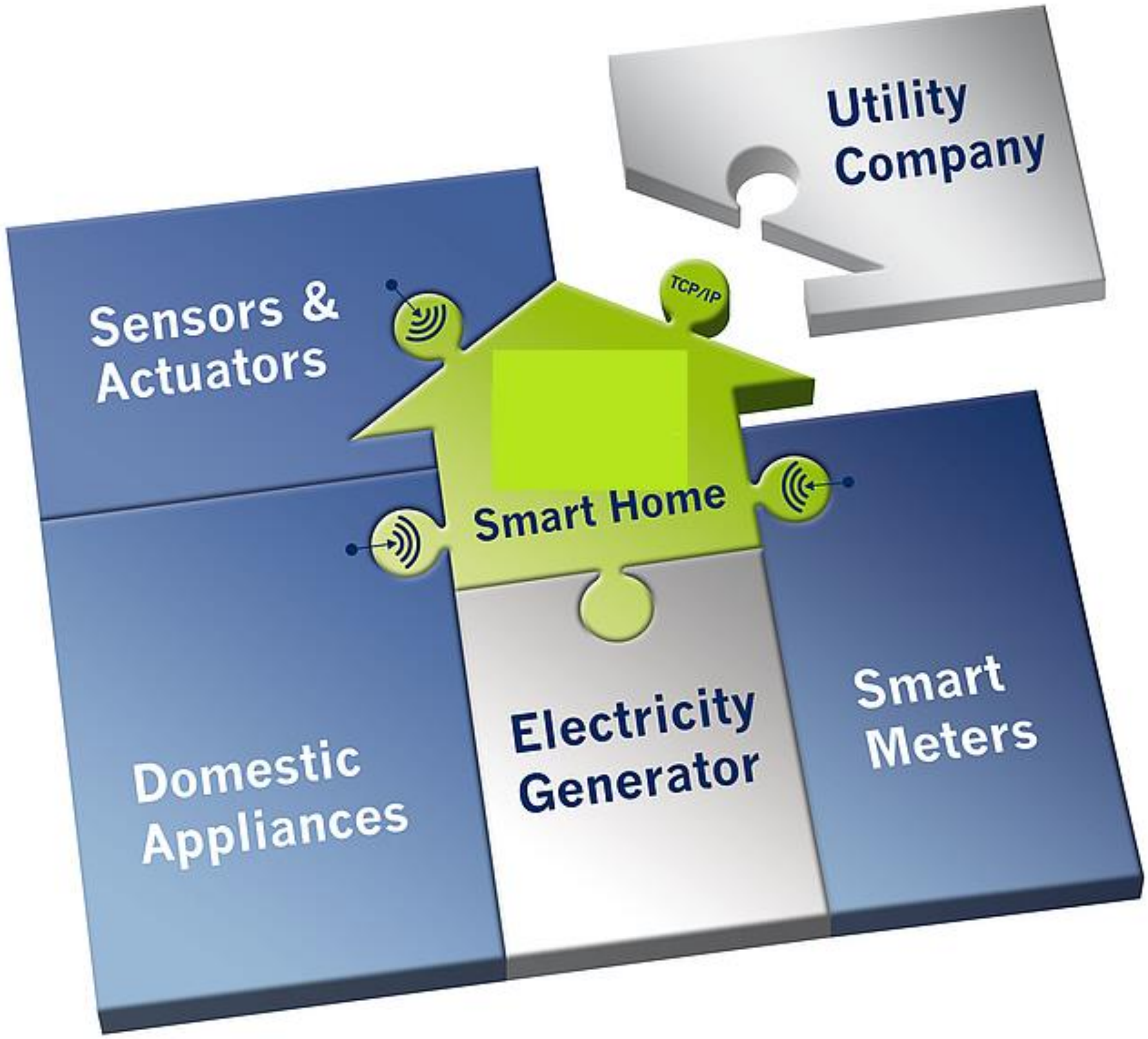


Figure 37. Smart Home Components

A brief description on each of these concepts is given as follows:

a) **Smart home**

A home consisting of all the devices turned smart, i.e. each device consists of its own control module such as designed and deployed by us in our project and making the users capable of not only controlling the lights, fans and ACs as per their own wishes but also controlling the heating thermostat, entertainment system (audio and video), security system (including the application of IP cameras) and other electronic devices. The whole house will turn into a fantasy home, where everything happens at the tap of its occupant's fingers. [32]

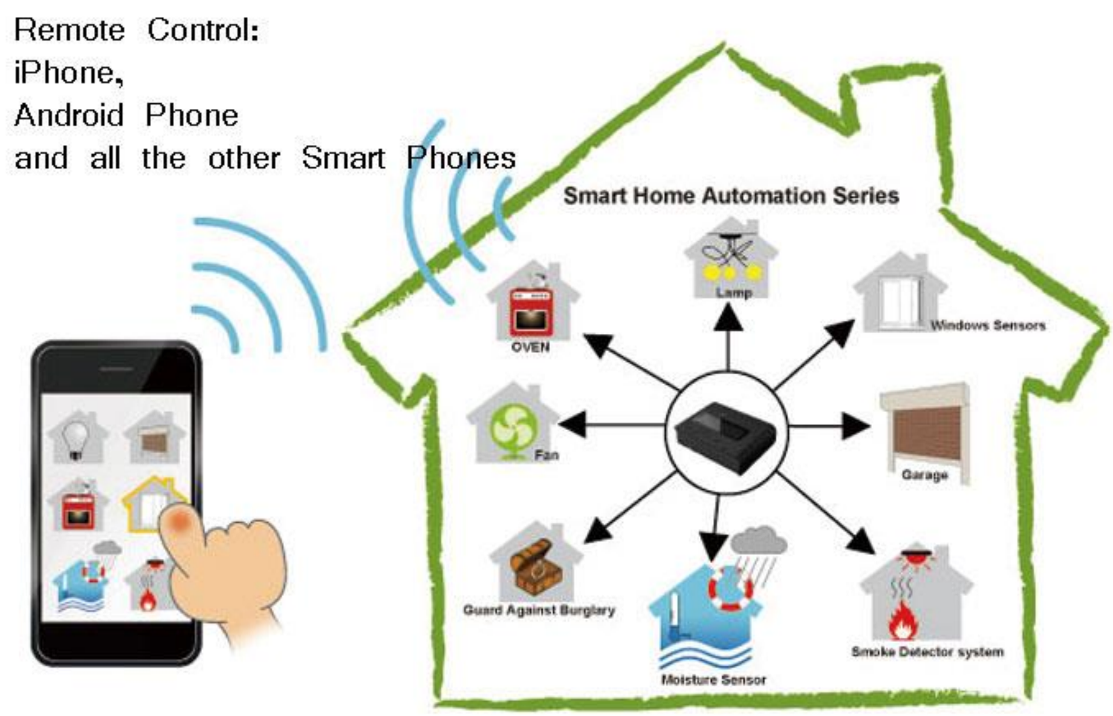


Figure 38. Controlling a Smart Home

b) Smart UPS

The concept of a smart UPS evolved primarily in order to avoid compromising the critical data or the loss of data altogether every time the power fails. Such a UPS is believed to provide with the reliable and efficient supply of energy. As soon as the power fails for one reason or another, this UPS activates just like a regular UPS, but the difference lies in the programming of the UPS to supply energy to only the priority devices in case of power dissipation of the UPS. In this way, the devices having highest priority are switched off at the end, giving the users enough time to save the data and manually switch off the device safely. [33]

c) Smart Grid

A Smart Grid is a technologically revolutionized way for homes and businesses to use, produce and sell electricity. An electrical grid is a huge network of transmission lines, power plants and substations that deliver electrical power to end connections. Statistically, 7% of the power being generated gets lost during transmission and distribution. Since many of the power plants are often dirty, inefficient and expensive and used to generate power for several dozen hours per year as they are specifically designed to meet peat energy demands. These type of plants can drive one bank corrupt and emit unnecessary pollution. Therefore, a newer, more efficient and cleaner version of the conventional power plants is devised, known as the Smart Grid.

The offices and housing systems on the smart grid are always in a constant communication with the grid to devise the most efficient and cheapest way to operate. The old/conventional meters are replaced by the digital Smart Meters that provide with the real time electricity data, through which the users can check their energy consumption via smart phones and computers. [34]

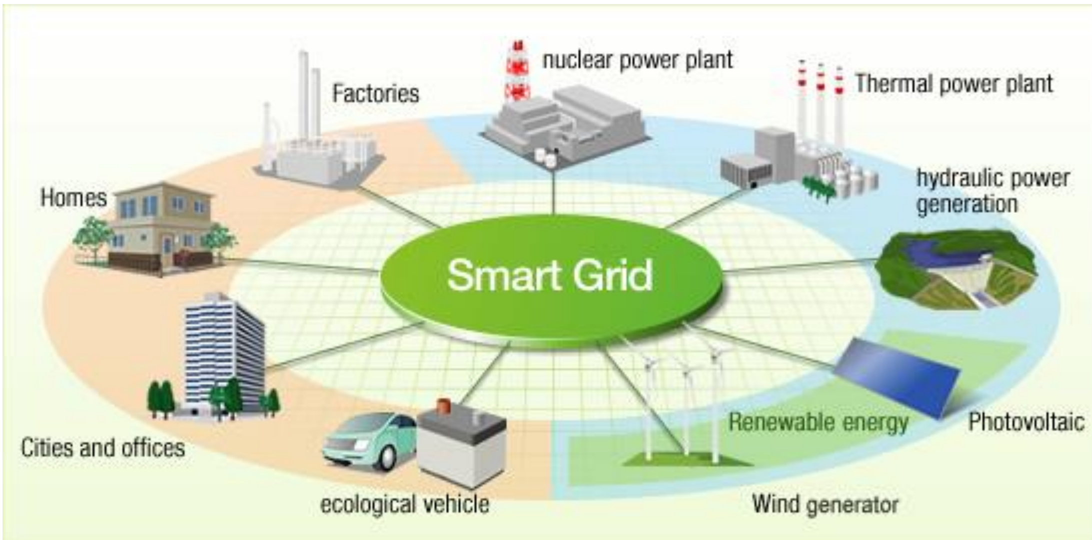


Figure 39. Smart Grid

d) Smart Meter

This is usually and electronic device that records the consumption of electricity after suitable intervals and sends that information on the daily basis to the consumer or utility in order to help monitor the energy consumption and carry out the bills themselves. This type of meter enables a two way communication between the meter and central system. Unlike the typical home electric meters, these smart meters are capable of gathering and sending the data to remote locations as required. [35] Even this is not an end to this direction of technology, this concept has yet to revolutionize out world utterly and completely.

Figure 40. Smart Meter

CHAPTER NO. 9

9. Conclusions

To conclude, we have initiated this revolutionary idea towards a new dimension in technology that not only provides the consumers with utmost luxury but also means of reducing their energy consumption at the same time. Now, the simple devices like a light bulb is not dumb anymore, you can also ask it for diagnostics in real time just by a single tap of your fingertips.

9.1. Overview

We have rendered the project capable enough that it helps a layman with the basic understanding of technology in controlling his whole house not only in the local area but also remotely. All he has to do is to be in the vicinity of internet connection and log in to the domain of his house and thereby, keeping a regular check on his home.

9.2. Applications

- Revolutionized concept of home, office, school and/or industrial automation system
- Control system which also offers the remote access to on network users
- Capable of performing automatic routing of traffic to a backup system

- Most effective way of reducing the power consumption and in turn save a lot of money
- Additional cost savings with reduced cabling, wiring and dedicated PC requirements
- The device simultaneously routs data to multiple destinations
- Data and diagnostics can be handled simultaneously in real time
- Once deployed it may be upgraded with further advancements in the front end operating systems with time while the back end physical system is an evergreen technology
- Future enhancements include: deployment of wireless network which is upgradable to 3G, 4G and further generation technologies
- Represents the concept of one time investment and lifetime saving

APPENDICES

Appendix A

Module circuit designs:

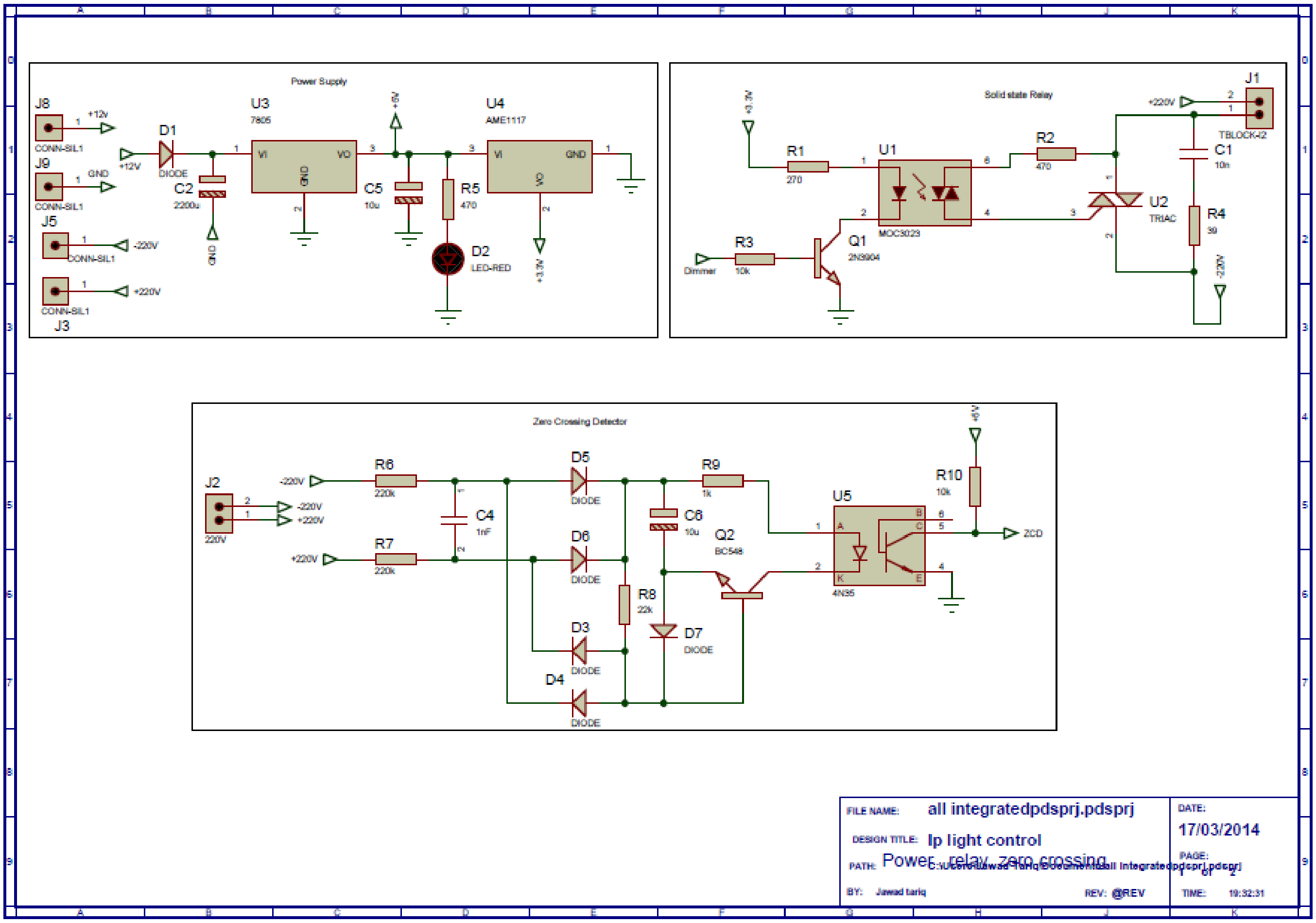


Figure 41. Circuit Module Designs

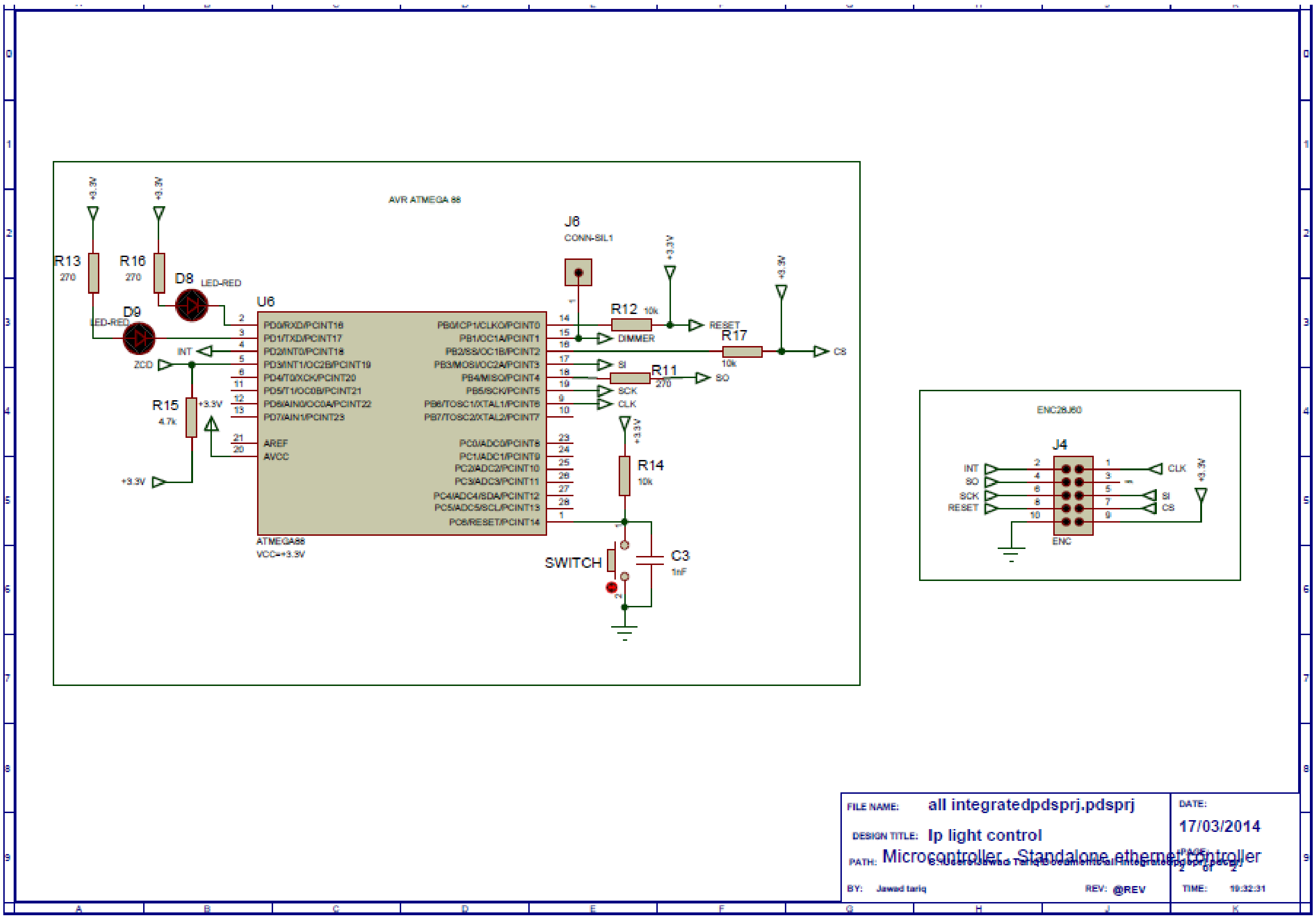


Figure 42. 2D Circuit Design

3-D view of the module

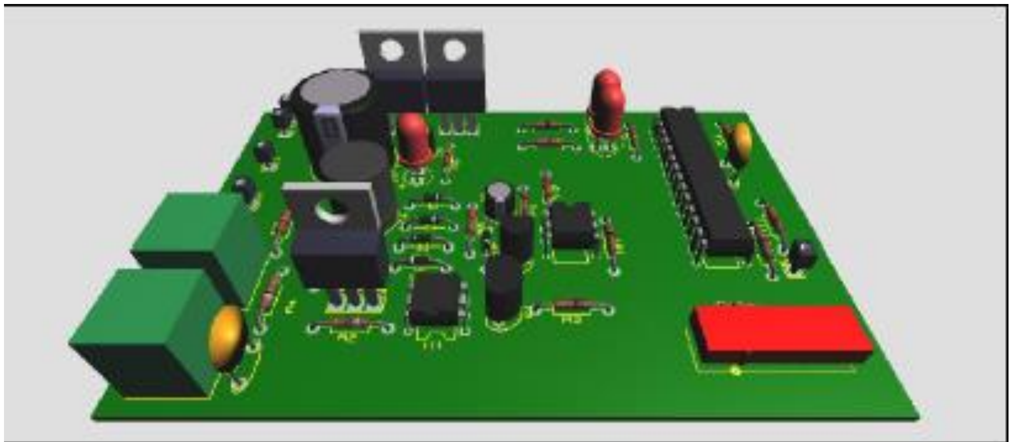


Figure 43. 3D Circuit Design

Appendix B

Serial to Ethernet Interfacing Code:

```
#include <avr/interrupt.h>//

#include <util/delay.h>//

#include "ip_arp_udp.h"

#include "enc28j60.h"

#include "timeout.h"

#include "avr_compat.h"

#include "net.h"



static uint8_t mymac[6] = {0x54,0x55,0x58,0x10,0x00,0x24};// mac address

static uint8_t myip[4] = {192,168,15,55}; //Ip address

static uint16_t myport =1200; // listen port for udp



#define BUFFER_SIZE 250

static uint8_t buf[BUFFER_SIZE+1];


// the password string (only the first 5 char checked):
```

```
static char password[]="secret";


//
uint8_t verify_password(char *str)
{
        // the first characters of the received string are
        // a simple password/cookie:
        if (strncmp(password,str,5)==0){
                return(1);
        }
        return(0);
}


volatile uint8_t a=0,b=0; // global variables




// initialize timer,
void timer1_init()


{
    // set up timer with prescaler = 8 and Frequency and Phase Correct mode
    TCCR1B |= (1 << WGM13)|(1 << CS10)|(1 << CS11);
```

```
    // set up timer OC1A pin in  INVERTING

    TCCR1A |= (1 << COM1A1);


    // initialize counter

    TCNT1 = 0;

    // initialize compare value

    ICR1 = 625;

         OCR1A=63;



}


void timer1_stop()

{

         TCCR1B &= ~((1 << WGM13)|(1 << CS10)|(1 << CS11));


}



int main(void){
```

```
    //Initialize the zero crossing detector INT(1)

EICRA|=(1<<ISC11);    //INT in falling edge
EIMSK = 1<<INT1;            //Enable INT1
sei();



    uint16_t plen;
    uint8_t i=0;
    uint8_t cmd_pos=0;
    uint8_t payloadlen=0;
    char str[30];
    char cmdval;



    CLKPR=(1<<CLKPCE);
    CLKPR=0; // 8 MHZ
    delay_ms(10);

    /* enable PB1, OCR1A as output to dimmer */
    DDRB|= (1<<DDB1);
```

```
/* enable PB0, reset as output */
DDRB|= (1<<DDB0);

/* enable PD2/INT0, as input interrupt for ENC */
DDRD&= ~(1<<DDD2);

        /* enable PD3/INT1, as input interrupt for Zero Crossing Detector */
DDRD&= ~(1<<DDD3);

/* set output to gnd, reset the ethernet chip */
PORTB &= ~(1<<PB0);
delay_ms(10);
/* set output to Vcc, reset inactive */
PORTB|= (1<<PB0);
delay_ms(200);

/*initialize enc28j60*/
enc28j60Init(mymac);
delay_ms(20);

// LED1
```

```
        /* enable PD0 as test LED*/
DDRD|= (1<<DDD0);

        /* set PD0*/
PORTD|= (1<<PD0);

// LED2
/* enable PD1, LED as output */
DDRD|= (1<<DDD1);

/* set output to Vcc, LED off */
PORTD|= (1<<PD1);

// the transistor on PD7
DDRD|= (1<<DDD7);
PORTD &= ~(1<<PD7);// transistor off

/* Magjack leds configuration, see enc28j60 datasheet, page 11 */
// LEDB=yellow LEDA=green
//
// 0x476 is PHLCON LEDA=links status, LEDB=receive/transmit
// enc28j60PhyWrite(PHLCON,0b0000 0100 0111 01 10);
```

```
enc28j60PhyWrite(PHLCON,0x476);
delay_ms(20);

/* set output to GND, red LED on */
PORTD &= ~(1<<PD1);
i=1;

//init the ethernet/ip layer:
init_ip_arp_udp(mymac,myip);

while(1){
    // get the next new packet:
    plen = enc28j60PacketReceive(BUFFER_SIZE, buf);

    /*plen will ne unequal to zero if there is a valid
     * packet (without crc error) */
    if(plen==0){
        continue;
    }
    // led----------
    if (i){
        /* set output to Vcc, LED off */
```

```
        PORTD|= (1<<PD1);
        i=0;
    }else{
        /* set output to GND, LED on */
        PORTD &= ~(1<<PD1);
        i=1;
    }

    // arp is broadcast if unknown but a host may also
    // verify the mac address by sending it to
    // a unicast address.
    if(eth_type_is_arp_and_my_ip(buf,plen)){
        make_arp_answer_from_request(buf,plen);
        continue;
    }
    // check if ip packets (icmp or udp) are for us:
    if(eth_type_is_ip_and_my_ip(buf,plen)==0){
        continue;
    }

    if(buf[IP_PROTO_P]==IP_PROTO_ICMP_V &&
buf[ICMP_TYPE_P]==ICMP_TYPE_ECHOREQUEST_V){
```

```
            // a ping packet, let's send pong
            make_echo_reply_from_request(buf,plen);
            continue;
        }
        // we listen on port 1200=0x4B0
        if
(buf[IP_PROTO_P]==IP_PROTO_UDP_V&&buf[UDP_DST_PORT_H_P]==4
&&buf[UDP_DST_PORT_L_P]==0xb0){
            payloadlen=buf[UDP_LEN_L_P]-UDP_HEADER_LEN;
            // you must sent a string starting with v
            // e.g udpcom version 10.0.0.24
            if (verify_password((char *)&(buf[UDP_DATA_P]))){
                // find the first comma which indicates
                // the start of a command:
                cmd_pos=0;
                while(cmd_pos<payloadlen){
                    cmd_pos++;
                    if (buf[UDP_DATA_P+cmd_pos]==','){
                        cmd_pos++; // put on start of cmd
                        break;
                    }
                }
```

```
// a command is one char and a value. At
// least 3 characters long. It has an '=' on
// position 2:
if (cmd_pos<2 || cmd_pos>payloadlen-3 ||
buf[UDP_DATA_P+cmd_pos+1]!='='){
        strcpy(str,"e=no_cmd");
        goto ANSWER;
}
// supported commands are
// t=1 t=0 t=?
if (buf[UDP_DATA_P+cmd_pos]=='t'){
        cmdval=buf[UDP_DATA_P+cmd_pos+2];

        if(cmdval=='0'){


timer1_stop();

                a=2;// bulb off
                strcpy(str,"t=0");
              //  goto ANSWER;z
```

```
timer1_stop();

                                        }else
if(cmdval=='1'){

        timer1_stop();

        a=1;

        b=0;                // bulb ON with full intensity

                  strcpy(str,"t=1");
                //   goto ANSWER;

                                          }else
if(cmdval=='2'){

        timer1_stop();
```

```
a=1;

b=6;                    // bulb ON with half intensity

strcpy(str,"t=2");
                    //    goto ANSWER;

                                                  }else
if(cmdval=='3'){

timer1_stop();

a=1;

b=8;                    // bulb ON with low intensity

strcpy(str,"t=3");
                    //  goto ANSWER;
```

```
                    }else if(cmdval=='?'){
                          if (PORTD & (1<<PD7)){
                                strcpy(str,"t=1");
                                goto ANSWER;
                          }
                          strcpy(str,"t=0");
                    //   goto ANSWER;
                    }
              }
              strcpy(str,"e=no_such_cmd");
              goto ANSWER;
        }
        strcpy(str,"e=invalid_pw");
ANSWER:
        make_udp_reply_from_request(buf,str,strlen(str),myport);
     }
  }
  return (0);
}

ISR(INT1_vect)
```

```
{
  //if(b==5)
        //b=0;
  if(a==1)
  {
        // PORTC ^= (1<<PC2);
         delay_ms(b);

        /*
         PORTC |= (1<<PC6);
         _delay_ms(1);
         PORTC &= ~(1<<PC6);
/*      for(int i=0;i<500;i++)
        {
         i++;
         i--;
        }
*/       timer1_init();
        // a=0;
        // b++;
        }
```

```
            if(a==2)

            {

             timer1_stop();

      a=0;

            timer1_stop();

            }


}
```

Appendix C

VB Application Code

```
Imports System.IO

Imports System.Net.Sockets.Socket

Imports System

Imports System.Text

Imports System.Net

Imports System.Net.Sockets

Imports Microsoft.VisualBasic

Imports System.Runtime.InteropServices

Public Class form1

    Inherits System.Windows.Forms.Form
```

```
Dim GLOIP As IPAddress

Dim GLOINTPORT As Integer

Dim bytCommand As Byte() = New Byte() {}

Dim udpClient As New UdpClient

Dim pRet As Integer

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load


    light1on.Visible = False

    light1hi.Visible = False

    light1med.Visible = False

    light1low.Visible = False

    light1off.Visible = False

    fanintensity.Visible = False

    fan1on.Visible = False

    fan1hi.Visible = False

    fan1med.Visible = False

    fan1low.Visible = False

    fan1off.Visible = False

    light1intensity.Visible = False

    exit1.Visible = False

    exit2.Visible = False
```

```
        light2on.Visible = False

        light2hi.Visible = False

        light2med.Visible = False

        light2low.Visible = False

        light2off.Visible = False

        exit3.Visible = False

        exit4.Visible = False

        light2intensity.Visible = False

        fan2intensity.Visible = False

        fan2on.Visible = False

        fan2hi.Visible = False

        fan2med.Visible = False

        fan2low.Visible = False

        fan2off.Visible = False

    End Sub


    Private Sub fan1menu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles fan1menu.Click
fan1on.Visible = True

        fan1hi.Visible = True

        fan1med.Visible = True

        fan1low.Visible = True
```

```
        fan1off.Visible = True
        fanintensity.Visible = True
        exit1.Visible = True

    End Sub

    Private Sub fan1off_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan1off.Click


        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next

        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
```

```
        GLOINTPORT = 1200
        udpClient.Connect(GLOIP, GLOINTPORT)
        bytCommand = Encoding.ASCII.GetBytes("secret,t=0")
        pRet = udpClient.Send(bytCommand, bytCommand.Length)
        System.Threading.Thread.Sleep(100)
        Continue For
    Next



    fan1on.BackColor = Color.Gray
    fan1off.BackColor = Color.Black
    fan1hi.BackColor = Color.Gray
    fan1low.BackColor = Color.Gray
    fan1med.BackColor = Color.Gray
    fan1on.Enabled = True
    fan1off.Enabled = True
    fan1low.Enabled = False
    fan1med.Enabled = False
    fan1hi.Enabled = False

End Sub
```

```
   Private Sub fan1on_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan1on.Click
        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next







        fan1on.BackColor = Color.Black
        fan1off.BackColor = Color.LightGray
        fan1on.Enabled = True
```

```
        fan1off.Enabled = True
        fan1low.Enabled = True
        fan1med.Enabled = True
        fan1hi.Enabled = True


    End Sub

    Private Sub fan1low_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan1low.Click
        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=3")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next
```

```
        fan1low.BackColor = Color.Black
        fan1med.BackColor = Color.Gray
        fan1hi.BackColor = Color.Gray
    End Sub

    Private Sub fan1med_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan1med.Click

        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=2")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next

        fan1med.BackColor = Color.Black
```

```
        fan1low.BackColor = Color.Gray
        fan1hi.BackColor = Color.Gray

    End Sub

    Private Sub fan1hi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan1hi.Click
        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.55")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next

        fan1hi.BackColor = Color.Black
        fan1low.BackColor = Color.Gray
```

```
        fan1med.BackColor = Color.Gray
        GLOIP = IPAddress.Parse("192.168.15.55")
        GLOINTPORT = 1200
        udpClient.Connect(GLOIP, GLOINTPORT)
        bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
        pRet = udpClient.Send(bytCommand, bytCommand.Length)
    End Sub

    Private Sub exit1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles exit1.Click
        fanintensity.Visible = False
        fan1on.Visible = False
        fan1hi.Visible = False
        fan1med.Visible = False
        fan1low.Visible = False
        fan1off.Visible = False
        exit1.Visible = False
    End Sub

    Private Sub light1menu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1menu.Click
        light1on.Visible = True
```

```
        light1hi.Visible = True
        light1med.Visible = True
        light1low.Visible = True
        light1hi.Visible = True
        light1intensity.Visible = True
        exit2.Visible = True
        light1off.Visible = True


    End Sub

    Private Sub light1on_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1on.Click
        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.50")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next
```

```
        light1on.BackColor = Color.Black
        light1off.BackColor = Color.LightGray
        light1on.Enabled = True
        light1off.Enabled = True
        light1low.Enabled = True
        light1med.Enabled = True
        light1hi.Enabled = True

    End Sub

    Private Sub light1off_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1off.Click

        For buzo As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.50")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
```

```
    pRet = udpClient.Send(bytCommand, bytCommand.Length)
    System.Threading.Thread.Sleep(100)
    Continue For
Next
For buzo As Integer = 1 To 3
    GLOIP = IPAddress.Parse("192.168.15.50")
    GLOINTPORT = 1200
    udpClient.Connect(GLOIP, GLOINTPORT)
    bytCommand = Encoding.ASCII.GetBytes("secret,t=0")
    pRet = udpClient.Send(bytCommand, bytCommand.Length)
    pRet = udpClient.Send(bytCommand, bytCommand.Length)
    pRet = udpClient.Send(bytCommand, bytCommand.Length)
    System.Threading.Thread.Sleep(100)
    Continue For
Next

light1on.BackColor = Color.Gray
light1off.BackColor = Color.Black
light1hi.BackColor = Color.Gray
light1low.BackColor = Color.Gray
light1med.BackColor = Color.Gray
light1on.Enabled = True
```

```
light1off.Enabled = True

light1low.Enabled = False
light1med.Enabled = False
light1hi.Enabled = False

End Sub

Private Sub light1low_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1low.Click
    For buzo As Integer = 1 To 2
        GLOIP = IPAddress.Parse("192.168.15.50")
        GLOINTPORT = 1200
        udpClient.Connect(GLOIP, GLOINTPORT)
        bytCommand = Encoding.ASCII.GetBytes("secret,t=3")
        pRet = udpClient.Send(bytCommand, bytCommand.Length)
        System.Threading.Thread.Sleep(100)
        Continue For
    Next

    light1low.BackColor = Color.Black
```

```
        light1med.BackColor = Color.Gray
        light1hi.BackColor = Color.Gray

    End Sub
    Private Sub light1med_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1med.Click
        For buzo As Integer = 1 To 2
            GLOIP = IPAddress.Parse("192.168.15.50")
            GLOINTPORT = 1200
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("secret,t=2")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next


        light1med.BackColor = Color.Black
        light1low.BackColor = Color.Gray
        light1hi.BackColor = Color.Gray

    End Sub
```

```
Private Sub light1hi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light1hi.Click
    For buzo As Integer = 1 To 2
        GLOIP = IPAddress.Parse("192.168.15.50")
        GLOINTPORT = 1200
        udpClient.Connect(GLOIP, GLOINTPORT)
        bytCommand = Encoding.ASCII.GetBytes("secret,t=1")
        pRet = udpClient.Send(bytCommand, bytCommand.Length)
        System.Threading.Thread.Sleep(100)
        Continue For
    Next
    light1hi.BackColor = Color.Black
    light1med.BackColor = Color.Gray
    light1low.BackColor = Color.Gray

End Sub
Private Sub exit2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles exit2.Click
    light1intensity.Visible = False
    light1on.Visible = False
    light1hi.Visible = False
```

```
        light1med.Visible = False
        light1low.Visible = False
        light1off.Visible = False
        exit2.Visible = False
    End Sub

    Private Sub light2menu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light2menu.Click
        light2on.Visible = True
        light2hi.Visible = True
        light2med.Visible = True
        light2low.Visible = True
        light2hi.Visible = True
        light2intensity.Visible = True
        exit3.Visible = True
        light2off.Visible = True

    End Sub

    Private Sub exit3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles exit3.Click
        light2intensity.Visible = False
```

```
        light2on.Visible = False
        light2hi.Visible = False
        light2med.Visible = False
        light2low.Visible = False
        light2off.Visible = False
        exit3.Visible = False
    End Sub

    Private Sub light2off_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light2off.Click
        For index As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.60")
            GLOINTPORT = 12345
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("dddddddddddddddddddddd")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next
```

```
        light2on.BackColor = Color.Gray
        light2off.BackColor = Color.Black
        light2hi.BackColor = Color.Gray
        light2low.BackColor = Color.Gray
        light2med.BackColor = Color.Gray
        light2on.Enabled = True
        light2off.Enabled = True
        light2low.Enabled = False
        light2med.Enabled = False
        light2hi.Enabled = False

    End Sub

    Private Sub light2med_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light2med.Click
        For index As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.60")
            GLOINTPORT = 12345
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb")
```

```
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next






        light2med.BackColor = Color.Black
        light2low.BackColor = Color.Gray
        light2hi.BackColor = Color.Gray

    End Sub

    Private Sub light2hi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles light2hi.Click

        For index As Integer = 1 To 5
            GLOIP = IPAddress.Parse("192.168.15.60")
            GLOINTPORT = 12345
            udpClient.Connect(GLOIP, GLOINTPORT)
```

```
            bytCommand = Encoding.ASCII.GetBytes("aaaaaaaaaaaaaaaaaa")

            pRet = udpClient.Send(bytCommand, bytCommand.Length)

            System.Threading.Thread.Sleep(100)

            Continue For

        Next

        light2hi.BackColor = Color.Black

        light2med.BackColor = Color.Gray

        light2low.BackColor = Color.Gray

    End Sub

    Private Sub fan2menu_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles fan2menu.Click
```

```
        fan2on.Visible = True
        fan2hi.Visible = True
        fan2med.Visible = True
        fan2low.Visible = True
        fan2off.Visible = True
        fan2intensity.Visible = True
        exit4.Visible = True

    End Sub

    Private Sub exit4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles exit4.Click
        fan2on.Visible = False
        fan2hi.Visible = False
        fan2med.Visible = False
        fan2low.Visible = False
        fan2off.Visible = False
        fan2intensity.Visible = False
        exit4.Visible = False
    End Sub
```

```
Private Sub light2low_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light2low.Click
    For index As Integer = 1 To 10
        GLOIP = IPAddress.Parse("192.168.15.60")
        GLOINTPORT = 12345
        udpClient.Connect(GLOIP, GLOINTPORT)
        bytCommand = Encoding.ASCII.GetBytes("ccccccccccccccccc")
        pRet = udpClient.Send(bytCommand, bytCommand.Length)
        System.Threading.Thread.Sleep(100)
        Continue For
    Next
    light2hi.BackColor = Color.Gray
    light2med.BackColor = Color.Gray
    light2low.BackColor = Color.Black
End Sub

Private Sub light2on_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles light2on.Click

    For index As Integer = 1 To 5
        GLOIP = IPAddress.Parse("192.168.15.60")
        GLOINTPORT = 12345
```

```
            udpClient.Connect(GLOIP, GLOINTPORT)
            bytCommand = Encoding.ASCII.GetBytes("aaaaaaaaaaaaaaaaaaaa")
            pRet = udpClient.Send(bytCommand, bytCommand.Length)
            System.Threading.Thread.Sleep(100)
            Continue For
        Next
        light2on.BackColor = Color.Black
        light2off.BackColor = Color.LightGray
        light2on.Enabled = True

        light2off.Enabled = True
        light2low.Enabled = True
        light2med.Enabled = True
        light2hi.Enabled = True
    End Sub
End Class
```

BIBLIOGRAPHY

[1] Smart Homes, Background on Smart Homes, http://www.academia.edu/4375055/Smart_Home_System_Analysis#

[2] Students of MCS, “Automated Home using BAS with GSM support”, MCS (NUST), Rawalpindi, 2012

[3] Students of MCS, “Android Based Home Automation System”, MCS (NUST), Rawalpindi

[4] Home Automation Green Energy Solutions, NXP Green Chip, www.-nxp.-com/-campaigns/-greenchip/-smartlighting

[5] Existing Home Automation Systems in the Market, http://techcrunch.com/2012/05/06/att-introduces-digital-life-ip-based-home-automation-and-security-system-with-247-monitoring-centers/

[6] Home automation in Pakistan, http://www.automatedhome.co.uk/reviews/lightwaverf-home-automation-system-in-depth-review.html

[7] Wiki, Understanding About Microcontroller, http://en.wikipedia.org/wiki/Microcontroller

[8] Atmega88 specifications, 8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash, http://www.atmel.com/images/doc2545.pdf

[9] Atmega88, http://www.atmel.com/images/doc2545.pdf

[10] Wikipedia, power line communication, http://en.wikipedia.org/wiki/Power-line_communication

[11] Wikipedia, fixed networks, http://en.wikipedia.org/wiki/Landline

[12] wireless/networking, fixed wireless networks, http://compnetworking.about.com/od/wirelessinternet/f/fixed-wireless.htm

[13] RF wiki, http://n.wikipedia.org/wiki/Radio_frequency

[14] Wikipedia, Wireless fidelity, http://en.wikipedia.org/wiki/Wi-Fi

[15] Wikipedia, automation systems, http://en.wikipedia.org/wiki/Automation#Industrial_automation

[16] Home improvement pages, types of automated home, http://www.homeimprovement pages.com.au/article/Types_Home_Automation

[17] Wikipedia, Building Home Automation, http://en.wikipedia.org/wiki/Building_automation

[18] BAS lighting control, http://www.automatedbuildings.com/news/mar09/columns/090219022129big.htm

[19] BAS device control, http://www.kmccontrols.com/products/Understanding_Building_Automation_and_Control_Systems.aspx

[20] Juan C Agusto, Smart Homes Can be Smarter

[21] Smart home, energy saving, https://www.smarthome.com/sh-solutions-save-energy-the-smarthome-way.html

[22] Javier Castro and James Psota, “The Specification, Design and Implementation of a Home Automation System”, MIT, Massachusetts

[23] Visual Basic tutorials, Visual Basic 6.0, http://www.vb6.us/

[24] Wikipedia, Ethernet Communication, http://en.wikipedia.org/wiki/Ethernet

[25] Wikipedia, Serial Communication, http://en.wikipedia.org/wiki/Serial_communication

[26] ATmega88, Microcontrollers, http://www.atmel.com/devices/ATMEGA88.aspx

[27] Florin, Niculae, Adrian Mircea, “Ethernet Interface for a System of Automation with Microcontrollers”, UOC, Craiova

[28] Wikipedia, SPI, http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[29] Switching with a Relay, Driving a Relay with a Transistor and Opto-isolator, http://electronics.stackexchange.com/questions/59456/driving-a-relay-with-transistor-and-opto-isolator

[30] Zero Crossing Detector Circuit, Zero Crossing Detector, http://www.visionics.a.se/html/curriculum/Experiments/Zero%20Crossing%20Detector/Zero%20Crossing%20Detector1.html

[31] WiFi Module Operations, WiFi Module User Manual, Shanghai Shuncom Slectronic Technology Co., Ltd, http://www.shuncom.com/

[32] Wikipedia, Smart Home, http://en.wikipedia.org/wiki/Home_automation

[33] Smart UPS, APC Smart-UPS, http://www.apc.com/products/family/?id=165

[34] Wikipedia, Smart Grid, http://en.wikipedia.org/wiki/Smart_grid

[35] Wikipedia, Smart Meter, http://en.wikipedia.org/wiki/Smart_meter