EAST WEST UNIVERSITY

Department of Electronics & Communication Engineering

Project on-

Ethernet shield based servo motor control system for entrance security using internet

Prepared by

Sajjad Hossain

ID: 2011-1-58-054

MD. Fakrul Hasan

ID: 2011-1-55-026

Project Supervisor

Dr. Saeed Mahmud Ullah

Department of Applied Physics, Electronics & Communication Engineering

University of Dhaka.

DECLARATION

We certify that the thesis titled “**Ethernet Shield Based Servo Motor Control System for Entrance Security Using Internet**” partial fulfillment of the B.Sc. Engineering degree in the Dept. of Electronics and Communication Engineering under the supervision of **Dr. Saeed Mahmud Ullah**, associate professor, Department of EEE, DU, adjunct faculty of **EWU** and it has not been submitted elsewhere for any other degree.

Signature:

…………………………	………………………………..
Sajjad Hossain	**MD.Fakrul Hasan**
ID: 2011-1-58-054	ID: 2011-1-55-026

Supervisor

Dr. Saeed Mahmud Ullah

Associate Professor, Dept. of Electrical and Electronic

Engineering

University Of Dhaka

APPROVAL

This is to certify that the Project titled as “**Ethernet Shield Based Servo Motor Control System for Entrance Security Using Internet**” submitted to the respected members of the Board of Examiners of the Faculty of Engineering for partial fulfillment of the requirements for the degree of Bachelor of Science in Electronics & Telecommunications Engineering by the following students and has been accepted as satisfactory.

Submitted by

Sajjad Hossain

ID: 2011-1-58-054

MD. Fakrul Hasan

ID: 2011-1-55-026

Dr. Saeed Mahmud Ullah

Associate Professor, Electrical and Electronics Engineering

University of Dhaka

……………………………..

Dr. Gurudas Mandal

Associate Professor and Chairperson,

Dept of Electronics and Communication Engineering

East West University, Aftabnagar, Dhaka

Acknowledgements

First of all we would like to express our heartiest gratitude to Almighty Allah for the successful completion of our undergraduate thesis. We must express our heartfelt gratitude to Associate Professor **Dr. Saeed Mahmud Ullah** under whose supervision we started developing “**Ethernet Shield Based Servo Motor Control System for Entrance Security Using Internet** “as our project goal. Without the guidance and constant support along with continuous encouragement from him, our work would have never been materialized.

Finally, we would like to thank our family, friends and all persons who were involved with this project for their valuable help and professionalism during this project. We are also indebted to the members of our family, especially to our parents for their patients and constant encouragement.

Dedication

To our Parents

ABSTRACT

Home security is something, that is applicable to all of us and a traditional home security system involves having lock on the door or security guards which can prevent burglary in some cases. This project is aimed to develop a microcontroller based control system where a servo motor has been controlled by internet. The servo motor actually controls a security lock for entrance door. The resident can choose to control the door from anywhere which gives security as well as convenience for him. An Ethernet shield has been used along with Arduino Uno microcontroller. Arduino sends appropriate signal to the servo motor which eventually opens or locks the door. The user sends command using particular apps/browser through internet to the Ethernet shield which eventually conveys the signal to the microcontroller. Under this project we have successfully implemented and demonstrated the prototype of this system. This project can also be extended to control other home/office/industrial appliances (using relay).

Table of Content

Chapter 1: Introduction Pages 1-12

Chapter 2: An overview of microcontrollers

Chapter 3: Theory behind this project

Chapter 4: System Design and Analysis

Chapter 5: Implementation and result

Chapter 6: Market analysis

Appendix

References

List of figure:

Chapter 1

Introduction

1.1 What is security system?

The most basic definition of any security system is found in its name. It is literally a means or method by which something is secured through a system of interworking components and devices.

In this instance, we're talking about home security systems, which are networks of integrated electronic devices working together with a central control panel to protect against burglars and other potential home intruders [1].

A typical home security system includes:

- A control panel, which is the primary controller of a home's security system
- Door and window sensors
- Motion sensors, both interior and exterior
- Wired or wireless security cameras
- A high-decibel siren or alarm
- A yard sign and window stickers

Security is a degree of protection that is arranged by an individual or a group of people to protect one's valuable things from being stolen. Theft is the main deterrent of security protection. For a very long time people have been focusing on security system. Trying to establish the best system ever so that they could sleep or pass their vacations without worrying about their house. [2]

Home security systems work on the simple concept of securing entry points into a home with sensors that communicate with a control panel or command center installed in a convenient location somewhere in the home.

The sensors are typically placed in doors that lead to and from a house as well as easily accessible windows, particularly any that open, especially those at ground level. Open spaces inside of homes can be secured with motion sensors.

During vacations security issues become more vulnerable. It is often seen that a family after returning from outing discovers all they have, been robbed! So people become very insecure to leave the vacant house and go for outing. In Bangladesh we have been using door lock or guards at the main gate as mean of security. But these steps are never enough to ensure complete security. Door locks can easily be broken or unlocked. Security guards could be bribed to cheat us. In spite of that it is very common scenario that a guard is napping sitting outside the gate being completely unaware what is happening around him. It proves the security system turned into a great mockery. Should we still rely on manual security system sitting in the 21st century, where the whole world has become automated? In this present world of technology, depending on manual power for security, would nothing but a proof of our ignorance. It is high time we should keep pace with the modern era and digitize our security system. Well, it might sound like, how a developing country like ours could afford such kind of security. It is a good thing to be announced that modern technology has allowed us to create a security system that is fully affordable and can provide a level of security that is more than any manual protection.

1.2 Types of Security:

Today's technical world has made such excellent innovations which the world had never experienced. There are different types of security systems available. All of them have the same purpose, which is to ensure safety. But with the improvement in technology, security systems are available with different features and everybody can choose the one which is best suited to them. Below a brief discussion is given:

- Wireless Systems:

Wireless security system is very simple to install and it can be kept anywhere because it doesn't need to be plugged. We can change its position anytime we want. Generally this type of system use battery powered security systems which have less expensive startup cost. But it is not a very reliable system as it depends on its battery power. It doesn't spoil the beauty of home with tangled wires.

- Door and window sensors:

These sensors are placed on entrance or window to detect intrusion. It consists of two pieces, one is on the door and the other one is in the door frame and same with the window. When a burglar tries to enter, the sensors trigger the control panel and alert the owner.

- Motion sensors:

These sensors are used to detect motion around its coverage area. Some of these sensors can detect the level of infrared radiation and detect the presence of an intruder with the change in infrared radiation. Example of such a motion sensor is passive infrared sensor. Some of these release a high frequency wave to detect intrusion.

- Vibration sensors:

These sensors detect vibration range in areas where they have provision. If it is placed on window and a burglar tries to get in, it detects and gives an alarm.

- Acoustic sensors:

It's working principle based on detection of sounds such as breaking glasses. So these are kept in any entrance such as doors and windows where an intruder may try to get in the house or office.

- Smoke, heat and CO2 sensors:

Smoke detectors detect smoke from a fire, heat sensor detect the rapid rise in heat and CO2 sensors sense level of C02 and a possibility of leakage from gas stove. After detection they sound an alarm.

- Flood sensors:

These are used to detect the level of standing water in industry and home. It consists of two parts. One is on the floor which is the sensor and the other part is on the wall a few inches above from the floor which is the communicator.

- Image sensors:

They have a built in camera which continuously monitor our property and if detect any movement, it takes picture of the intruder.

- Panic buttons:

These can be manually activated if anyone suspects the presence of an intruder. It can be placed near bed and by pressing the button it gives an alarm.

- Microwave sensors:

These sensors emit microwaves using a transmitter and when received by a receiver it detects if there is a reduction in the beam, which ensures the presence of an intruder.

- Digital security system:

These systems work with a code which is burnt in it and if someone tries to enter he must give the code to disarm it. If it is not given, there would be an alarm or the security company monitoring the device sends help. Automatic door lock is also a type of digital security system which can be controlled remotely. Online based security system is now-a-days is being very much popular.

- GSM Based Security System:

GSM based security systems are one of the most reliable systems which are operated with the latest GSM technology. The user can be connected to the security system as long he is under the GSM network and can be alerted in the case of an intrusion.

Energy is the capacity of a physical system to perform work. Energy exists in several forms such as heat, kinetic or mechanical energy, light, potential energy, electrical, or other forms.

1.3 Purpose of this project

Nowadays peoples are concern about their home security system like doors, lockers, industrial entrance etc. This project shows how to set up an arduino with Ethernet shield to control a servo motor through a server or via mobile apps. The web servers in this project are used to serve up web pages that can be accessed from a web browser running from any computer or mobile device. The web server or the software controls (e.g. motor rotationally up and down from the web or apps) and monitors (e.g. whether it works or display in the webpage) the hardware. From this device we can also control other home appliance like fans, lights, TV, refrigerator etc. In this project we designed and implemented a microcontroller based digital door lock system. It can be used at home, factory, industry wherever there an entrance point to lock or unlock the door or window. In this project we use Arduino Uno, Ethernet shield, Servo motor, adapter, pc, Smart phone, Wi-Fi router or internet connectivity. It is secured, easily controllable easily and environment friendly. The main purpose of this project is given below:

- To design a smart home appliance control system
- Construct this controller circuit with breadboard initially
- Make easy to control for entrance security and secure
- Industrialization this system
- To design this system with a low cost components
- To design other home appliance by using relay or servo motor

Chapter 2

AN OVERVIEW OF MICROCONTROLLERS

2.1 Microcontroller

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on asingle integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, 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. 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 mill 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. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

2.2 History

The first 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 Smithsonian Institution says TI engineers Gary Boone and Michael Cochran succeeded in creating the first microcontroller in 1971. The result microcontroller was one of the most successful in the company's history, and expanded the division's budget over 25%. Most microcontrollers at this time had two variants. 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; sometimes this was signified with the designation OTP, standing for "one-time programmable". The PROM was actually exactly the same type of memory as the EPROM, but because there was no way to expose it to ultraviolet light, it could not be erased. The erasable versions required ceramic packages with quartz windows, making them significantly more expensive than the OTP versions, which could be made in lower-cost opaque plastic packages. For the erasable variants, quartz was required, instead of less expensive glass, for its transparency to ultraviolet—glass is largely opaque to UV—but the main cost differentiator was the ceramic package itself. In 1993, the introduction of EEPROM memory allowed microcontrollers (beginning with the Microchip PIC16x84) to be electrically erased quickly without an expensive package as required for EPROM, allowing both rapid prototyping, and In System Programming. (EEPROM technology had been available prior to this time, but the earlier EEPROM was more expensive and less durable, making it unsuitable for low-cost mass-produced microcontrollers.) The same year, Atmel introduced the first microcontroller using Flash memory, a special type of EEPROM. Other companies rapidly followed suit, with both memory types. Cost has plummeted over time, with the cheapest 8-bit microcontrollers being available for under 0.25 USD in quantity (thousands) in 2009, and some 32-bit microcontrollers around 1USD for similar quantities. Nowadays microcontrollers are cheap and readily available for hobbyists, with large online communities around certain processors. In the future, MRAM could potentially be used in microcontrollers as it has infinite endurance and its incremental semiconductor wafer process cost is relatively low of their work was the TMS 1000, which became commercially available in 1974. It combined read-only memory, read/write memory, processor and clock on one chip and was targeted at embedded systems. Partly in response to the existence of the single-chip TMS 1000[7], Intel developed a computer system on a chip optimized for control applications, the Intel 8048, with commercial parts first shipping in 1977[7]. It combined RAM and ROM on the same chip. This chip would find its way into over one billion PC keyboards, and other numerous applications. At that time Intel's President, Luke J. Valenter, stated the invention.

2.3 Types of Microcontrollers:

As of 2008 there are several dozen microcontroller architectures and vendors including:

- ARM core processors (many vendors)
- ARM Cortex-M cores are specifically targeted towards microcontroller applications
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Cypress Semiconductor's M8C Core used in their PSoC (Programmable System-on-Chip)
- Free scale Cold Fire (32-bit) and S08 (8-bit)
- Free scale 68HC11 (8-bit)
- Intel 8051
- Infineon: 8-bit XC800, 16-bit XE166, 32-bit XMC4000 (ARM based Cortex M4F), 32-bit Tricore and, 32-bit AurixTricore Bit microcontrollers

- MIPS
- Microchip Technology PIC, (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24), (32-bit PIC32)
- NXP Semiconductors LPC1000, LPC2000, LPC3000, LPC4000 (32-bit), LPC900,LPC700 (8-bit)
- Parallax Propeller
- PowerPC ISE
- Rabbit 2000 (8-bit)
- Renesas Electronics: RL78 16-bit MCU; RX 32-bit MCU; Super H; V850 32-bit MCU; H8; R8C 16-bit MCU.

2.4 Embedded Design:

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system. The majority of microcontrollers in use today are embedded in other machinery, such as automobiles, telephones, appliances, and peripherals for computer systems. While some embedded systems are very sophisticated, many have minimal requirements for memory and program length, with no operating system, and low software complexity. Typical input and output devices include switches, relays, solenoids, LEDs, small or custom LCD displays, radio frequency devices, and sensors for data such as temperature, humidity, light level etc. Embedded systems usually have no keyboard, screen, disks, printers, or other recognizable I/O devices of a personal computer, and may lack human interaction devices of any kind.

2.5 Interrupts:

Microcontrollers must provide real time (predictable, though not necessarily fast) response to events in the embedded system they are controlling. When certain events occur, an interrupt system can signal the processor to suspend processing the current instruction sequence and to begin an interrupt service routine (ISR, or "interrupt handler"). The ISR will perform any processing required based on the source of the interrupt, before returning to the original instruction sequence. Possible interrupt sources are device dependent, and often include events such as an internal timer overflow, completing an analog to digital conversion. a logic level change on an input such as from a button being pressed, and data received on a communication link. Where power consumption is important as in battery operated devices, interrupts may also wake a microcontroller from a low power sleep state where the processor is halted until required to do something by a peripheral event.

2.6 Advantages of Microcontroller

- Microcontrollers are cheap and very small in size; therefore they can be embedded on any device.
- Programming of Microcontrollers is simple to learn. It’s not much complicated.
- We can use simulators on Computers to see the practical results of our program. Thus we can work on an embedded project without even buying the required Components and Chips. Thus we can virtually see the working of our project or program.
- Low cost and small packaging.
- Simple software able to control it all and which everyone can easily learn about has been developed
- Since microcontroller has faster execution and fully integrated inside the processor, e.g. computer on a chip .These devices operate at faster speeds to execute instructions compare to general purpose microprocessor.

2.7 Application of microcontroller:

Microcontrollers are mostly used in following electronic equipment’s:

- ✓ Home appliance system
- ✓ Auto Mobiles
- ✓ CD/DVD Players
- ✓ Washing Machines
- ✓ Cameras
- ✓ In Computers-Modems and Keyboard Controllers
- ✓ Security Alarms
- ✓ Electronic Measurement Instruments.
- ✓ Microwave Oven.

Chapter 3

THEORY BEHIND THIS PROJECT

3.1 Introduction:

This project includes with arduino uno, Ethernet shield, servo motor, adapter(Dc 5v), Wi-Fi router, PC, smart internet connector device, RJ 45 cable. We connect the Ethernet Shield to an Arduino Uno. At first we developed a circuit without load (e.g. LED light). After that we built up the circuit with load (e.g. Servo Motor) to control it.

3.2 Microcontrollers:

A microcontroller (sometimes abbreviated µC, uC 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 Ferroelectric RAM, 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.

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.

In electronics market different types of microcontroller are availableATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. Here we use Arduino Uno which is developed by Arduin of Ivrea, who was the margrave of Ivrea and king of Italy from 1002 to 1014. [4]

3.3 Arduino Uno:

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

3.4 Architecture of Arduino-Uno

There are different types of Arduino board uses for different purpose. We use Arduino Uno board, where a microcontroller chip built in. Architecture of this board and components is given below:

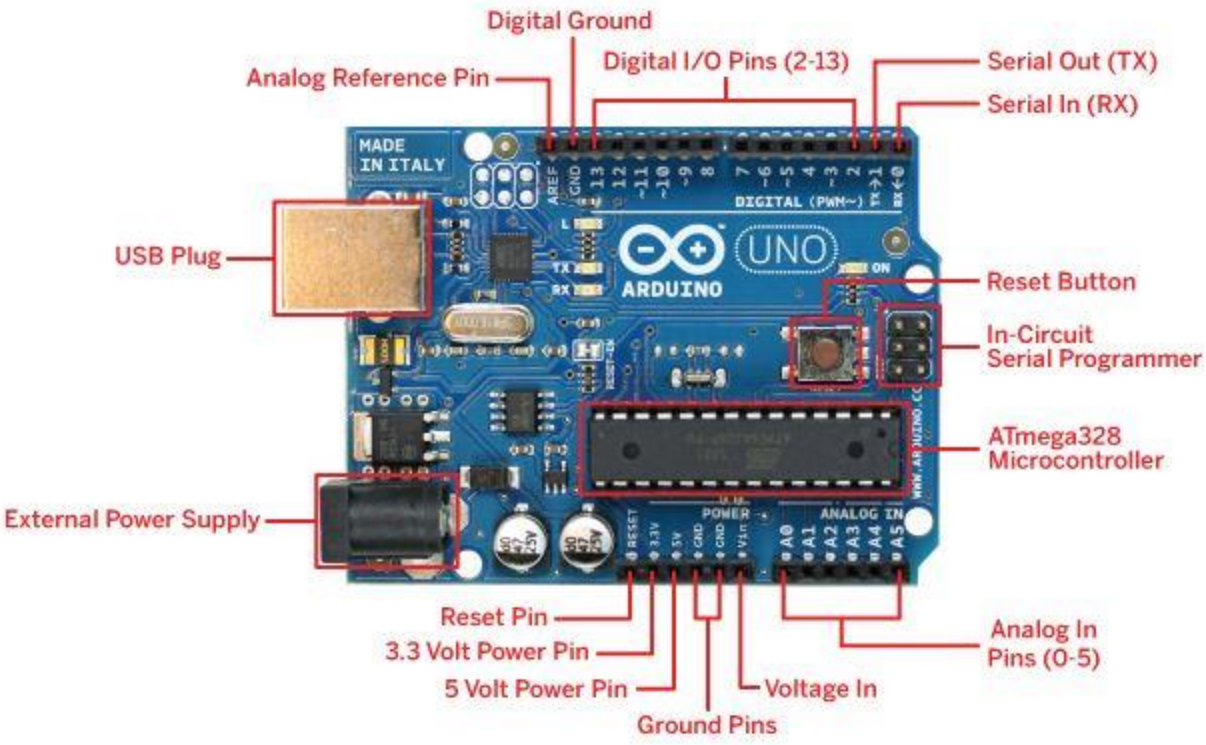

Fig: 3.1Arduino-Uno R3 Board

3.5 Arduino Functionality

All arduino boards have a few key capabilities and functions. To examine the arduino uno take some moment, it will be base configuration. These are some key components that will be concerning with:

- Atmel microcontroller
- Voltage regulator and power connections
- USB programming/ communication interface(s)
- Reset button
- Debug, power
- RX/TX LEDs
- In- circuit serial programmer
- Breakout I/O pins

3.5.1 Overview

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

- 1.0pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that uses the AVR, which operate with 5V and with the Arduino due, which operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

3.5.2 Atmel Microcontroller

In every arduino there is an Atmel microcontroller unit (MCU). Most arduino boards, including the arduino uno, use an AVR atmega microcontroller. The microcontroller is responsible for holding all compiled code and executing the commands specifically. Arduino programming language gives access to microcontroller peripherals, including analog to digital converters (ADC), general purpose input/ output (I/O) pins, communication buses (including I^2C and SPI), and serial interfaces. Most arduino boards come with a debug LED already connected to pin 13, which enables you to run first program (blinking an LED) without connecting any additional circuitry.

3.5.3 Programming Interfaces

The most important characteristics of an arduino are that anyone can program it easily via USB, without using a separate programmer. This functionality is made possible by the arduino boot loader. The boot loader is loaded onto the atmega at the factory (using the ISCP header), which allows a serial USART (Universal Synchronous/Asynchronous receiver/ transmitter) to load program on the arduino without using a separate programmer.

3.5.4Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- Vin: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector

(5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- 3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.5.5 Memory

The ATmega328 has 32 KB (with 0.5 KB used for the boot loader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

3.2.6 TX (Transmitter) and RX (Receiver) LEDs

TX is short for transmit, RX is short for receive. At initial stage both TX and RX blinking. First we try with a LED light. In our project case, there are two places on the Arduino UNO where TX and RX appear once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs. These LEDs will give us some nice visual indications whenever Arduino is receiving or transmitting data.

3.5.7 Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write () function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analog Reference ().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

3.5.8 Reset Button

The Arduino has a reset button. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if code doesn’t repeat, but we want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any problems.

3.5.9 Power Jack

In arduino a power jack uses as AC to DC adapter. It supplies 7-12 volt on the board, and power of VCC and GND supply from arduino board.

3.5.10 Voltage Regulator

The voltage regulator is not actually something interacting with on the Arduino. But it is potentially useful to know that it is there and what it's for. It controls the amount of voltage that is let into the Arduino board. It will turn away an extra voltage that might harm the circuit.

3.5.11 Technical specification

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 Ma
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by boot loader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3.5.12 Schematic Diagram

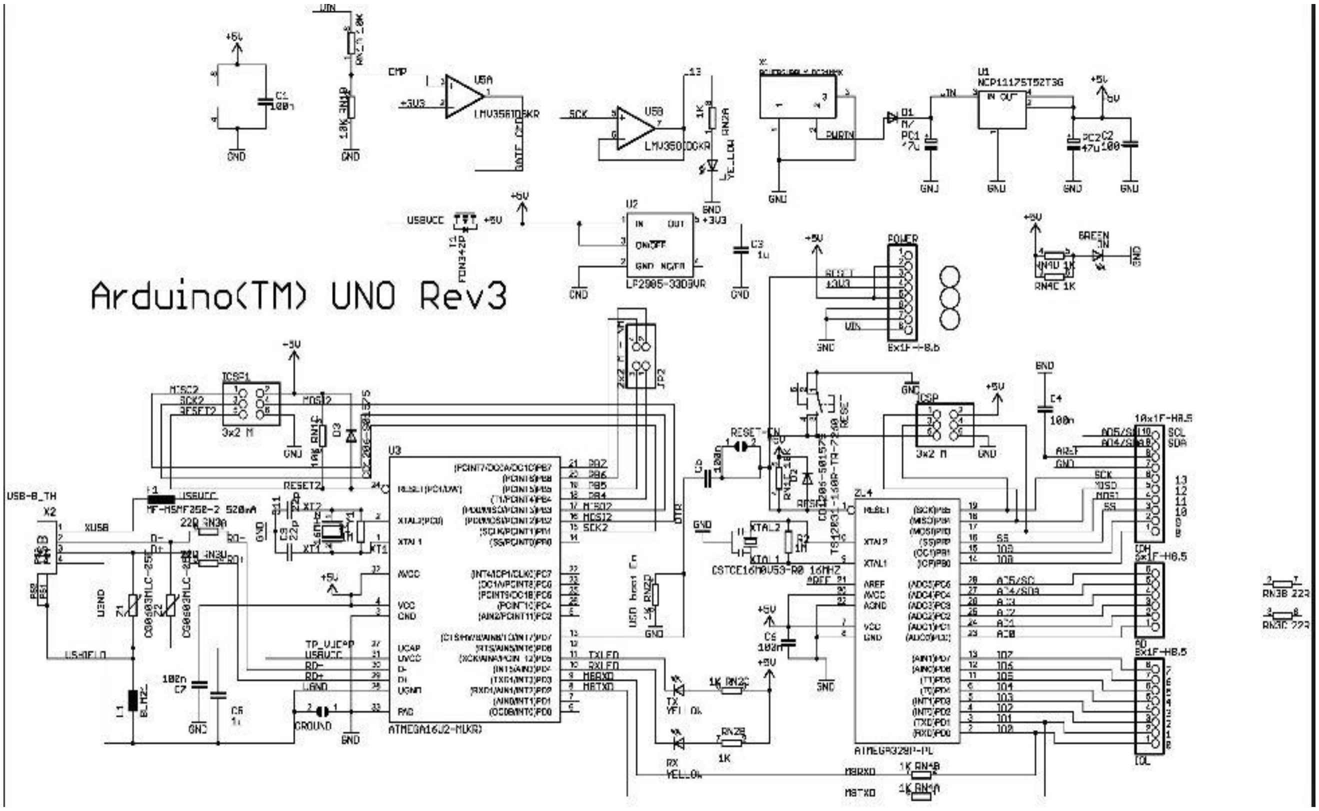

Fig: 3.2 Schematic diagram of Arduino UNO.

3.6 Getting started with Arduino Software

At first download a arduino software from arduino website and then install for widows, Mac, Linux, Android from arduino.cc. Windows users also need to install a driver. Connect your board via USB, launch the Arduino application and select Arduino-Uno from the tools to board menu. Open the sketch File. **For example:** Blinking for different time. Click the toolbar button to upload it to your board and you can change time duration for checking.

3.7.1 The Integrated Development Environment (IDE)

Every microcontroller needs software to be programmed. The Arduino board is not a case apart. It has its own integrated development environment (IDE).It is free and everyone can download it from its official website using either the Windows, Mac OS X or Linux platform. That allows Arduino Board to gain more users and it also helps it to grow.

3.3.2 IDE Parts

- Compile: Before program “code” can be sent to the board, it needs to be converted into instructions that the board understands. This process is called Compiling.
- Stop: This stops the compilation process.
- Create new Sketch: This opens a new window to create news ketch.
- Open Existing Sketch: This loads a sketch from a file on our computer.
- Save Sketch: This saves the changes to the sketch.
- Upload to Board: This compiles and then transmits over the USB cable to our board.
- Serial Monitor: Until this point when our programs (sketches) didn’t work, we just pulled out our hair and tried harder.
- Tab Button: This lets you create multiple files in your sketch. This is for more advanced programming than we will do in this class.
- Sketch Editor: This is where write or edit sketches.
- Text Console: This shows you what the IDE is currently doing and is also where error messages display if make a mistake in typing program. Line Number: This shows what line number your cursor is on.

3.7 Arduino Ethernet Shield

The Arduino Ethernet Shield is capable to easily connect an Arduino to the internet. This shield enables the Arduino to send and receive data from anywhere in the world with an internet connection. You can use it to do fun stuff like control robots remotely from a website, or ring a bell every time you get a new twitter message. This shield opens up endless amounts of possibility by allowing you to connect your project to the internet in no-time flat. This Shield is based on the Wiz net W5100 Ethernet chip (datasheet). The Wiz net W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous sockets connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The new Ethernet shield now includes a micro SD card connector, it is MEGA compatible and an on-board reset controller. This onboard micro-SD card slot can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

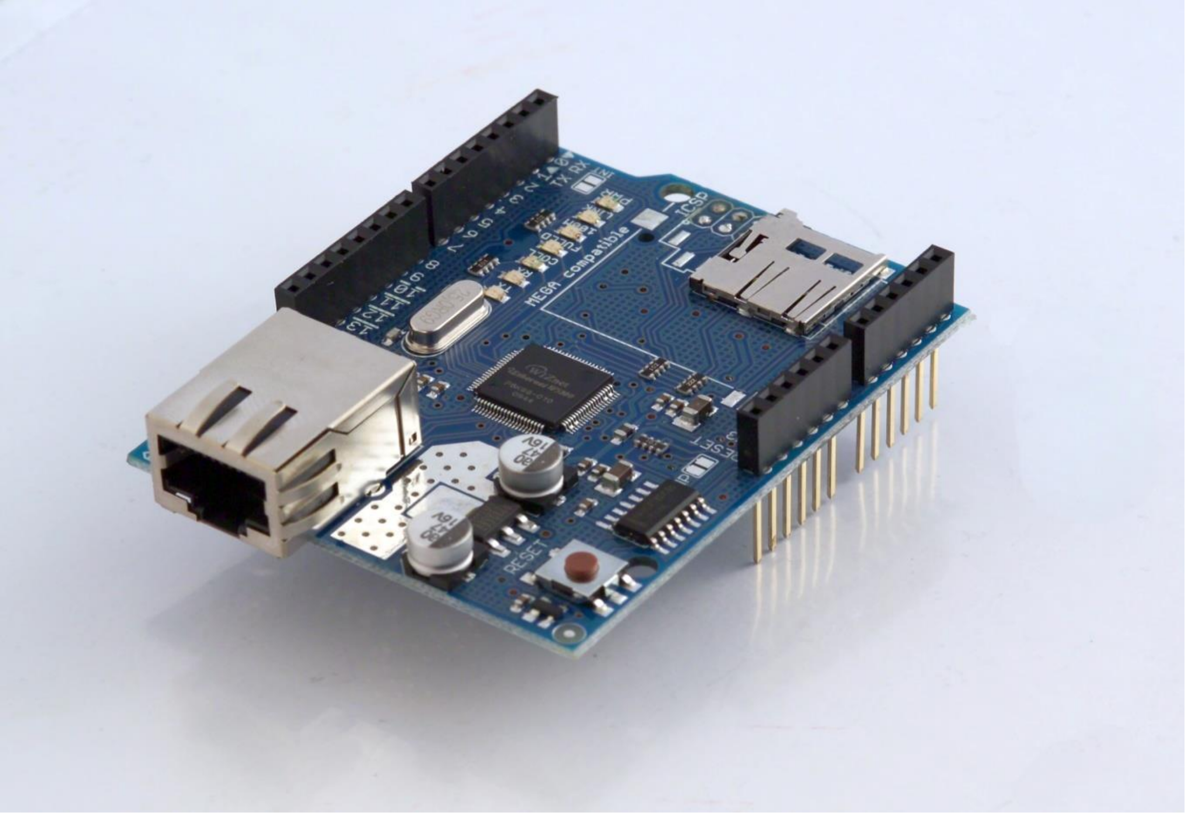

Fig 3.3: Arduino Ethernet Shield.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable.

The Ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

Arduino uses digital pins 10, 11, 12, and 13 (SPI) to communicate with the W5100 on the Ethernet shield. These pins cannot be used for general I/O. The shield provides a standard RJ45 Ethernet jack. An Arduino is necessary to use this shield but is not included.

3.7.1 Architecture of Arduino Ethernet Shield

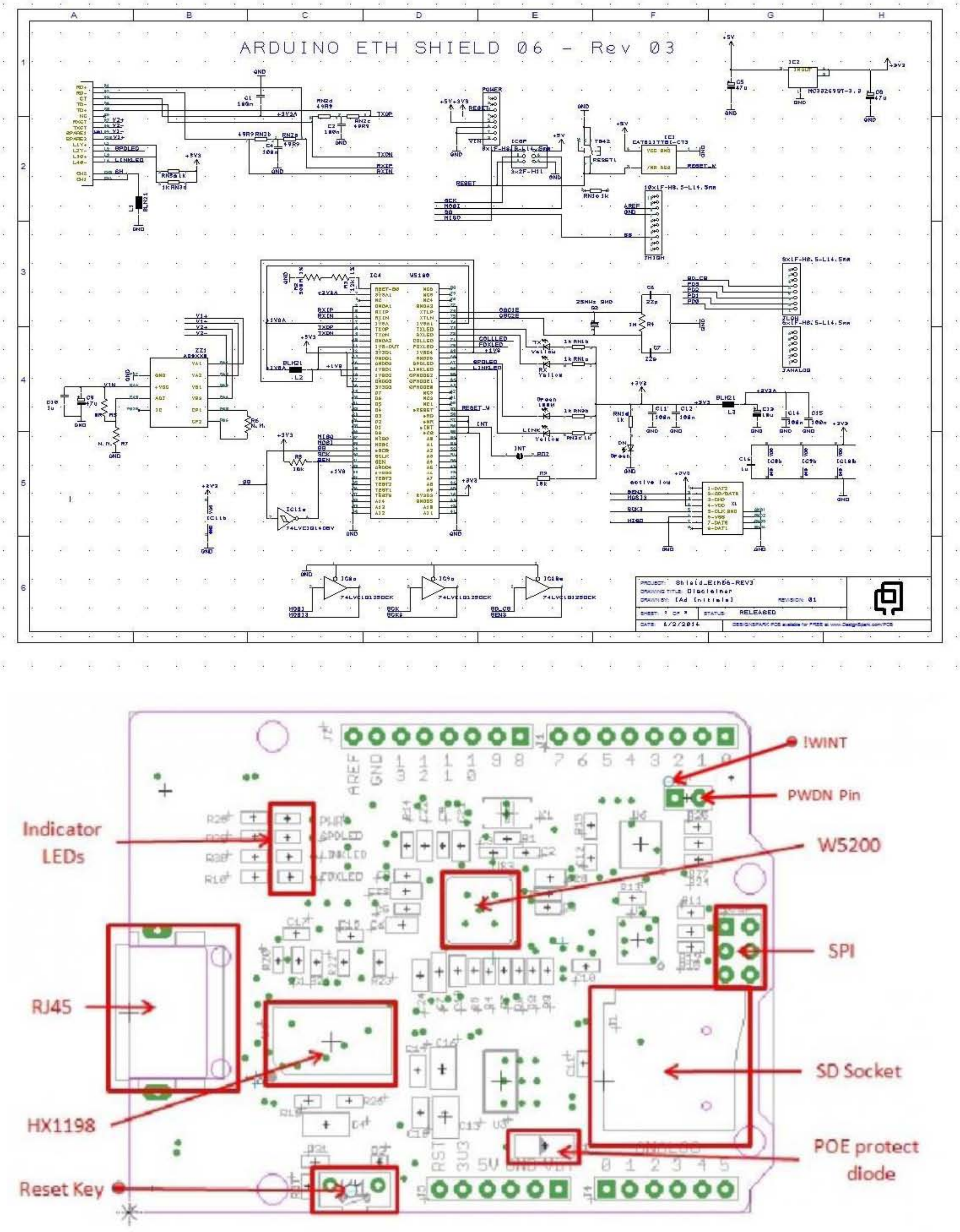

Fig 3.4: Architecture of Ethernet Shield

3.7.2 Description

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiz net W5100 Ethernet chip (datasheet). The Wiz net W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The Ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pin out on rev 3 of the Arduino UNO board.

The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with all the Arduino/Genuino boards. The on-board micro SD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.

The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.

The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: type 75% @ 50% load
- 1500V isolation (input to output)

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 10, 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 Ethernet jack.

The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

3.7.3 Where to use

With the Arduino Ethernet Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones. The library supports up to four concurrent connection (incoming or outgoing or a combination). Arduino communicates with the shield using the SPI bus. This is on digital pins 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used as SS. On the Mega, the hardware SS pin, 53, is not used to select the W5100, but it must be kept as an output or the SPI interface won't work.

Examples

- Chat Server: set up a simple chat server.
- Web Client: make a HTTP request.

- Web Client Repeating: Make repeated HTTP requests.
- Web Server: host a simple HTML page that displays analog sensor values.
- Barometric Pressure Web Server: outputs the values from a barometric pressure sensor as a web page.
- UDP Send Receive String: Send and receive text strings via UDP.
- UDP NTP Client: Query a Network Time Protocol (NTP) server using UDP.
- DNS Web Client: DNS and DHCP-based Web client.
- DHCP Chat Server: A simple DHCP Chat Server
- DHCP Address Printer: Get an IP address via DHCP and print it out
- Telnet Client: A simple Telnet client

3.7.4 Connecting the Shield

To use the shield, mount it on top of an Arduino board (e.g. the Uno). To upload sketches to the board, connect it to your computer with a USB cable as you normally would. Once the sketch has been uploaded, you can disconnect the board from your computer and power it with an external power supply. Connect the shield to your computer or a network hub or router using a standard Ethernet cable (CAT5 or CAT6 withRJ45 connectors). Connecting to a computer may require the use of a cross-over cable (although many computers, including all recent Macs can do the cross-over internally).

3.7.5 Network Settings

The shield must be assigned a MAC address and a fixed IP address using the Ethernet, begin () function. A MAC address is a globally unique identifier for a particular device. Current Ethernet shields come with a sticker indicating the MAC address you should use with them. For older shields without a dedicated MAC address, inventing a random one should work, but don't use the same one for multiple boards. Valid IP addresses depend on the configuration of your network. It is possible to use DHCP to dynamically assign an IP to the shield. Optionally, you can also specify a network gateway and subnet.

3.7.6 SD Card

The latest revision of the Ethernet Shield includes a micro-SD card slot, which can be interfaced with using the SD library.

The new Ethernet shield now includes a micro SD card connector, it is MEGA compatible and an on-board reset controller.

The Ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

Arduino uses digital pins 10, 11, 12, and 13 (SPI) to communicate with the W5100 on the Ethernet shield. These pins cannot be used for general I/O.

The shield provides a standard RJ45 Ethernet jack. An Arduino is necessary to use this shield but is not included.

3.8 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. [5] It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing. The main objectives of servo are given below:

- design a control system
- analyze the transient response
- fine tune the feedback loop
- determine the proper gear ratio for the desired speed or efficiency
- choose a motor
- build the mechanical sections
- build the amplifier and motor driver
- try to make it fit inside whatever you’re trying to control!

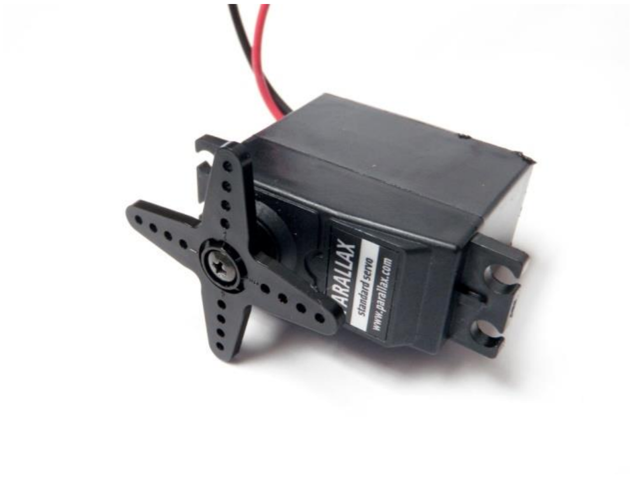

Fig 3.5: Servo Motor.

3.8.1 Advantages over Servo Motor Controller

- No gain tuning (Automatic adjustment of gain in response to a load change.)
- Maintains the stable holding position without oscillation after completing positioning.
- Fast positioning due to the independent control by on-board DSP.
- Continuous operation during rapid short-stroke movement due to instantaneous positioning.

3.8.2 Torque Comparison between Step and Servo Motor

- Torque from Servo Motor is proportional to position Error.
- Torque from Step Motor has no relation to position error.
- Torque from Step Motor has only relation to Speed.

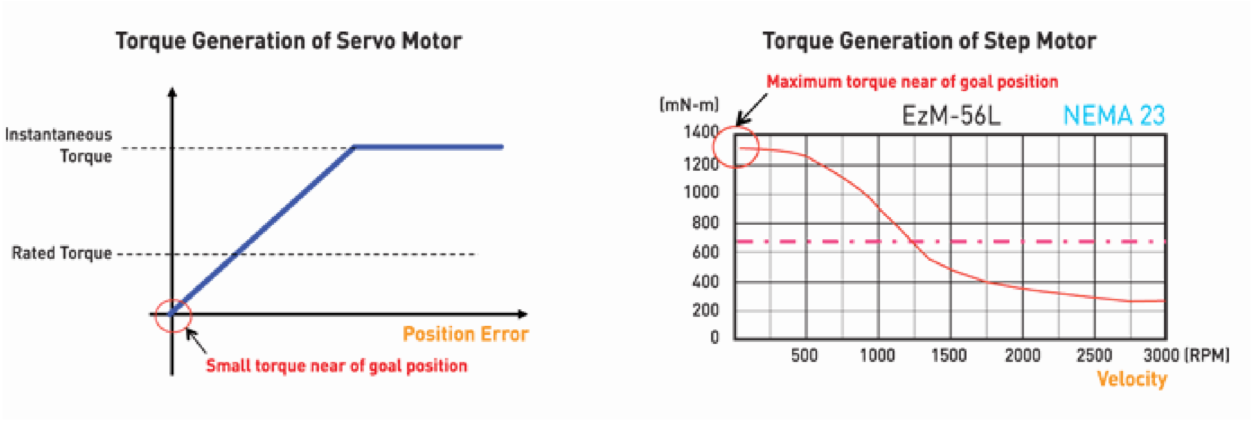

Fig 3.6: Compare between step and servo

3.8.3 No Gain Tuning

Conventional servo systems, to ensure machine performance, smoothness, positional error and low servo noise, require the adjustment of its servo s gains as an initial crucial step. Even systems that employ auto-tuning require manual tweaking after the system is installed, especially if more than one axis are interdependent. SERVO employs the best characteristics of stepper and closed loop motion controls and algorithms to eliminate the need of tedious gain tuning required for conventional closed loop servo systems. This means that SERVO is optimized for the application and ready to work right out of the box.

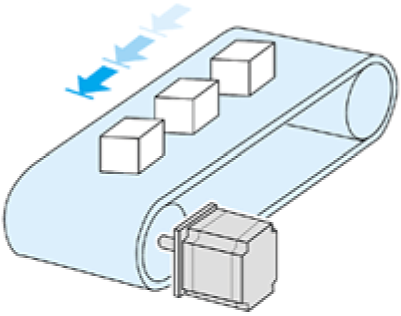

Fig 3.7: Gain Tuning

3.8.4 No Hunting

Traditional servo motor drives overshoot their position and try to correct by overshooting the opposite direction, especially in high gain applications. This is called null hunt and is especially prevalent in systems that the break away or static friction is significantly higher than the running friction. The cure is lowering the gain, which affects accuracy or using SERVO Motion Control System. SERVO utilizes the unique characteristics of stepping motors and locks itself into the desired target position, eliminating Null Hunt. This feature is especially useful in applications such as nanotech manufacturing, semiconductor fabrication, vi-sion systems and inkjet printing in which system oscillation and vibration could be a problem.

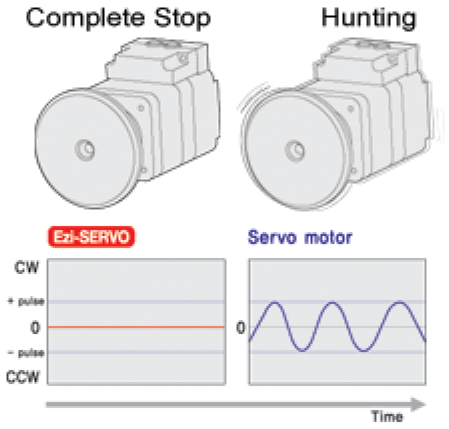

Fig3.8: Hunting position.

3.8.5 Smooth and Accuracy

Ezi-SERVO is a high-precision servo drive, using a high-resolution encoder with32, 000 pulses/ revolution. Unlike a conventional Micro step drive, the on-board high performance DSP (Digital Signal Processor) performs vector control and filtering, producing a smooth rotational control with minimum ripples.

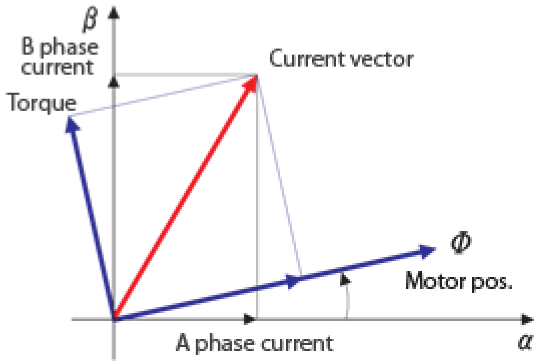

Fig 3.9: Accuracy curve.

3.8.6 Encoders

Modern servomotors use rotary encoders, either absolute or incremental. Absolute encoders can determine their position at power-on, but are more complicated and expensive. Incremental encoders are simpler, cheaper and work at faster speeds. Incremental systems, like stepper motors, often combine their inherent ability to measure intervals of rotation with a simple zero-position sensor to set their position at start-up. Instead of servomotors, sometimes a motor with a separate, external linear encoder is used.[6] These motor + linear encoder systems avoid inaccuracies in the drive train between the motor and linear carriage, but their design is made more complicated as they are no longer a pre-packaged factory-made system.

Most modern servomotors are designed and supplied around a dedicated controller module from the same manufacturer. Controllers may also be developed around microcontrollers in order to reduce cost for large volume applications.

3.8.7 Figure of Servo Connectors

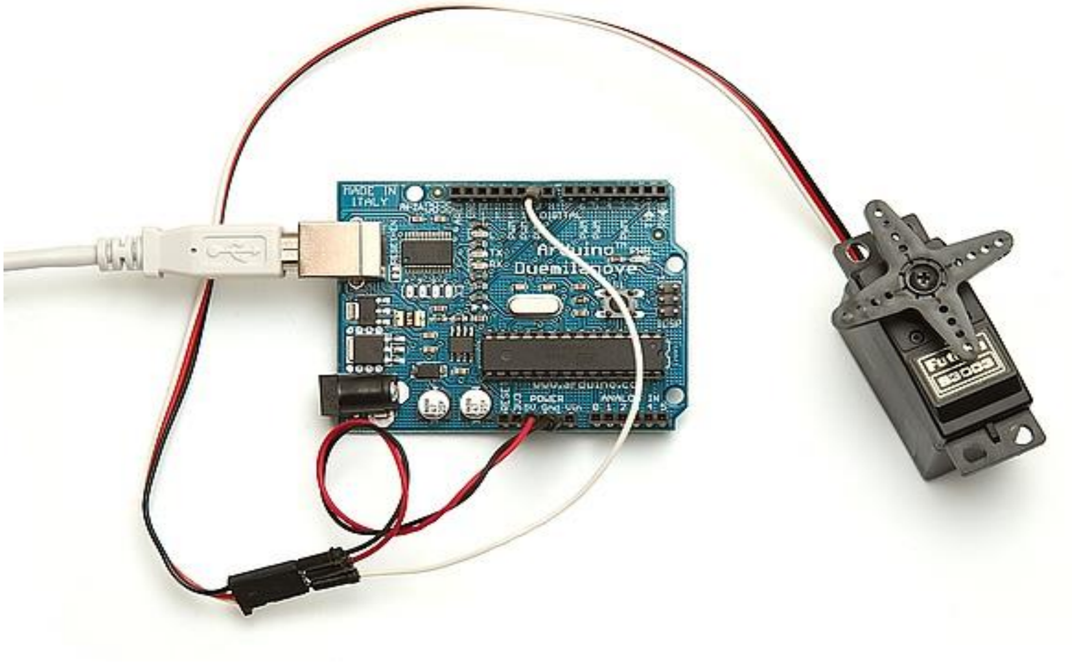

Fig 3.10: Servo with Arduino connect

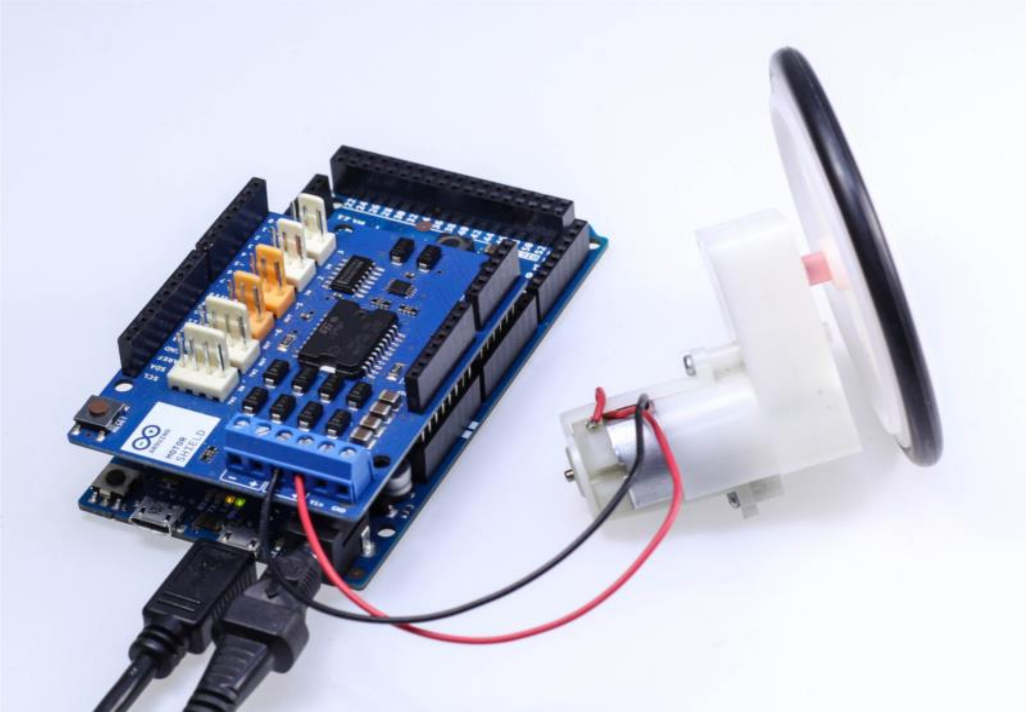

Fig 3.11: Servo with Ethernet Shield connect

3.8.8 Specification of servo motor

Basic Information

Modulation:	Digital
Torque:	**4.8V:** 130.5 oz-in (9.40 kg-cm) **6.0V:** 152.8 oz-in (11.00 kg-cm)
Speed:	**4.8V:** 0.19 sec/60° **6.0V:** 0.15 sec/60°
Weight:	1.94 oz (55.0 g)
Dimensions:	Length: 1.60 in (40.7 mm) Width: 0.78 in (19.7 mm) Height: 1.69 in (42.9 mm)
Motor Type:	3-pole
Gear Type:	Metal
Rotation/Support:	Dual Bearings

Additional Specifications

Rotational Range:	? (add)
Pulse Cycle:	1 ms
Pulse Width:	? (add)
Connector Type:	JR

3.8.9 Connectivity's

Internet

The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. Information that travels over the Internet does so via a variety of languages known as protocols

The Internet, sometimes called simply "the Net," is a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any other computer (and sometimes talk directly to users at other computers). It was conceived by the Advanced Research Projects Agency (ARPA) of the U.S. government in 1969 and was first known as the Arpanet. The original aim was to create a network that would allow users of a research computer at one university to "talk to" research computers at other universities. A side benefit of Arpanet's design was that, because messages could be routed or rerouted in more than one direction, the network could continue to function even if parts of it were destroyed in the event of a military attack or other disaster

WEB

The World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet. The Web uses the HTTP protocol, only one of the languages spoken over the Internet, to transmit data. Web services, which use HTTP to allow applications to communicate in order to exchange business logic, use the Web to share information. The Web also utilizes browsers, such as Internet Explorer or Firefox, to access Web documents called Web pages that are linked to each other via hyperlinks. Web documents also contain graphics, sounds, text and video.

The Web is just one of the ways that information can be disseminated over the Internet. The Internet, not the Web, is also used for e-mail, which relies on SMTP, Usenet news groups, instant messaging and FTP. So the Web is just a portion of the Internet, albeit a large portion, but the two terms are not synonymous and should not be confused.

Wi-Fi

Wi-Fi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. A common misconception is that the term Wi-Fi is short for "wireless fidelity," however this is not the case. Wi-Fi is simply a trademarked phrase that means IEEE 802.11x.

The Wi-Fi Alliance

The Wi-Fi Alliance, the organization that owns the Wi-Fi registered trademark term specifically 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." Initially, Wi-Fi was used in place of only the 2.4GHz 802.11b standard; however the Wi-Fi Alliance has expanded the generic use of the Wi-Fi term to include any type of network or WLAN product based on any of the 802.11 standards, including 802.11b, 802.11a, dual-band, and so on, in an attempt to stop confusion about wireless LAN interoperability.

Wireless Access Points or (WAPs)

Wireless access points (APs or WAPs) are special-purpose communication devices on wireless local area networks (WLANs). Access points act as a central transmitter and receiver of wireless radio signals. Mainstream wireless APs support Wi-Fi and are most commonly used to support public Internet hotspots and other business networks where larger buildings and spaces need wireless coverage. Access points are small hardware devices closely resembling home broadband routers. (Home routers actually integrate an access point into the rest of the device.) AP hardware consists of radio transceivers, antennas and device firmware. Access points enable so-called Wi-Fi infrastructure mode networking. Although Wi-Fi connections do not technically require the use of access points, APs enable Wi-Fi networks to scale to larger distances and numbers of clients. Modern access points support up to 255 clients (while very old ones supported only about 20). APs also provide bridging capability that enables a Wi-Fi network to connect to other wired networks.

Chapter 4

System Design and Analysis

4.1 Introduction

In this project we show that how an entrance system works for security purpose. Those people who are concern about their home security they can interested on it.To design a project includes into two parts, one is hardware design and another part is software design. We use Arduino Ethernet Shield, Servo Motor for the hardware design and we connected these components with microcontroller. Arduino software is downloaded from www.arduino.cc and C/C++ programmable language is used.

4.2 Hardware Design

The whole system design is divided into two parts to design a smart home appliance control system. One is the design the smart system in the breadboard and controls the designed system. Another part is the display part design to count the value in smart system. Finally, the smart home appliance control system is formed a complete integrated system. In this project Arduino development board is more efficient.

4.2.1 Block Diagram

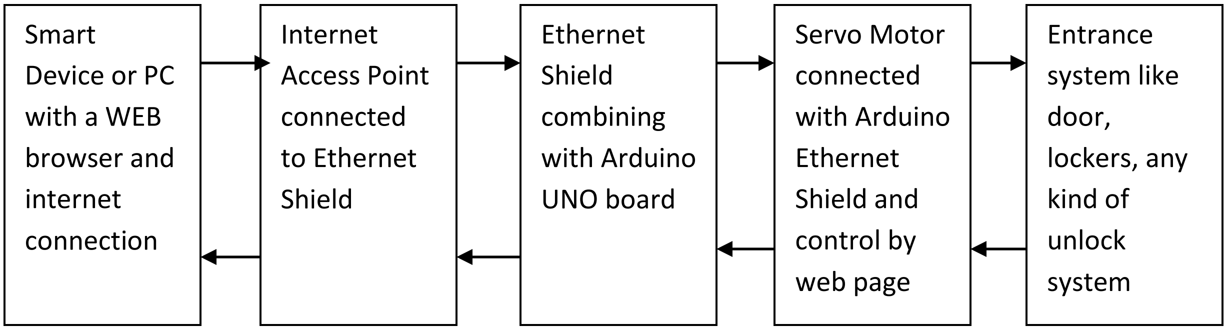
Smart Device or PC with a WEB browser and internet connection
Internet Access Point connected to Ethernet Shield
Ethernet Shield combining with Arduino UNO board
Servo Motor connected with Arduino Ethernet Shield and control by web page
Entrance system like door, lockers, any kind of unlock system

Fig 4.1: Block Diagram of control system

4.2.2 Equipment’s Used in this System

To design the project we use following component:

- **One Arduino Uno Board.**
- **One bread board.**
- **One Arduino Ethernet Shield.**
- **Internet access connector/Wi-Fi router**
- **RJ-45 Connector.**
- **UTP Cable.**
- **Two LED.**
- **Servo motor**
- **Power adapter 2V,1A**
- **1K, 2K, 330Ω Resistance.**
- **Jumper cable**
- **AC to DC power adapter**

4.2.3 Full Circuit Diagram

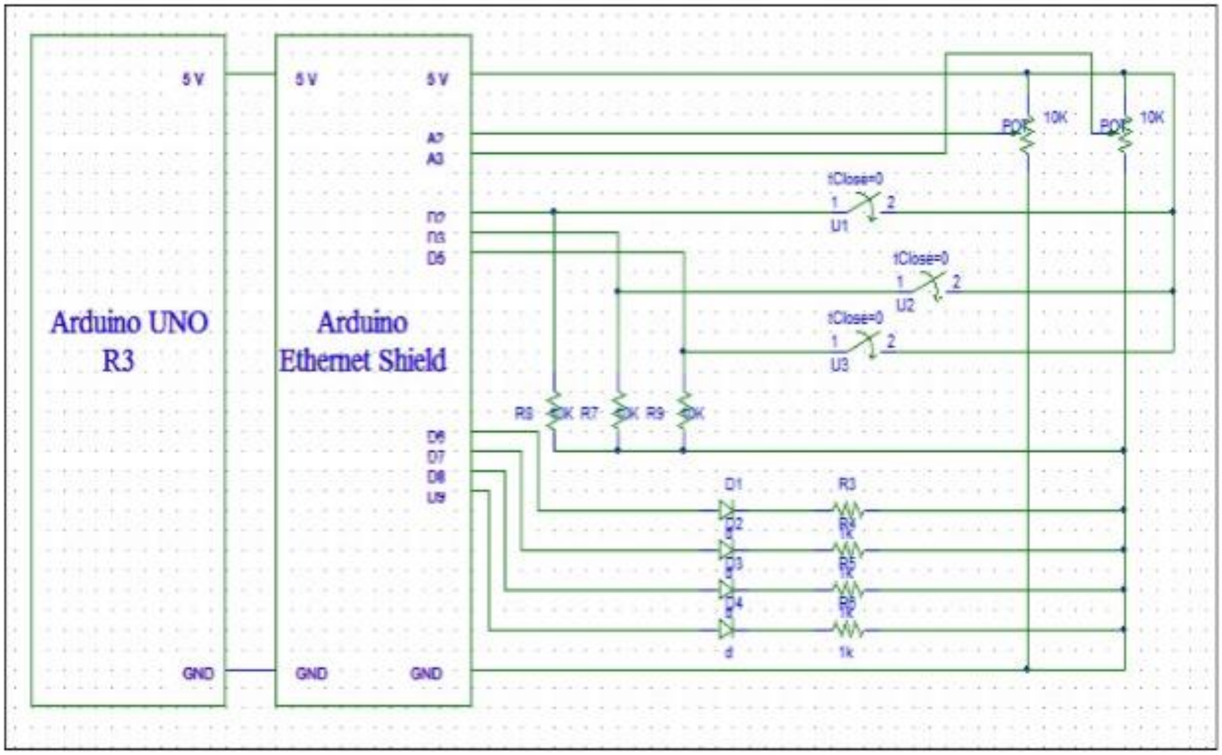

Fig 4.2: Full Circuit Diagram

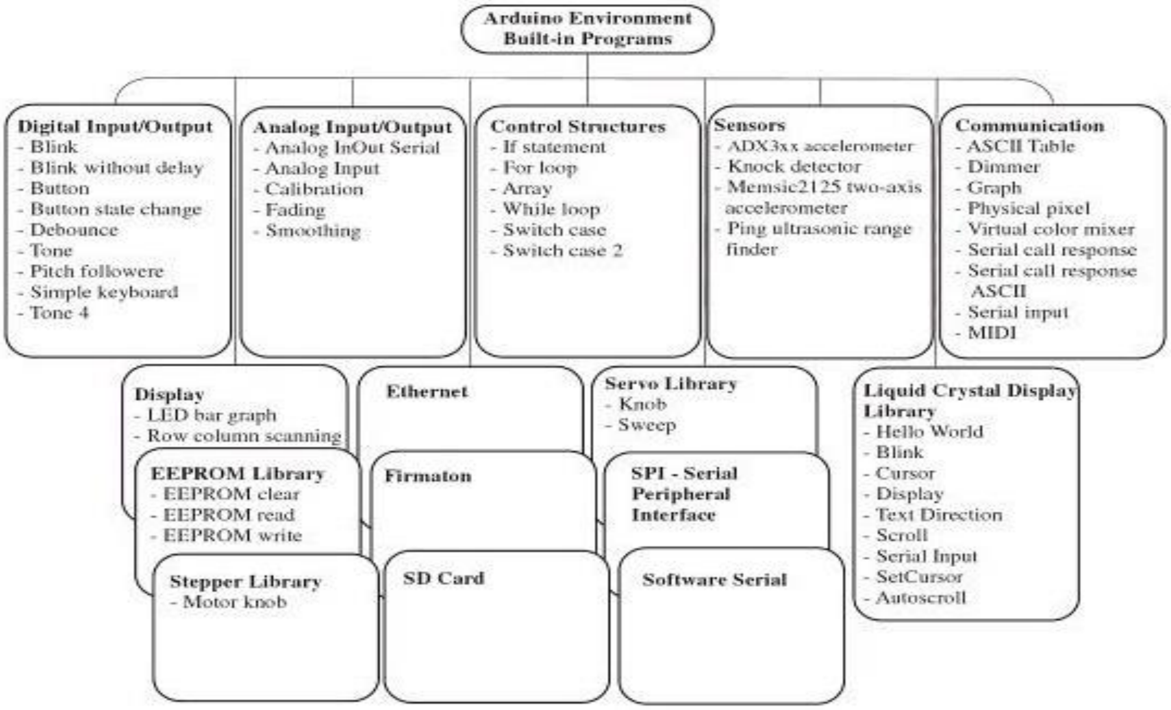

Fig 4.3: Block Diagram of Arduino Environmental System

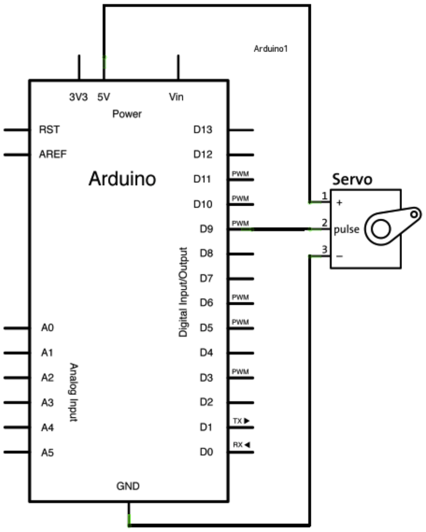

Fig 4.4: Arduino pin diagram with servo motor.

4.2.4 Connecting Arduino board with Ethernet Shield

Simply connect Ethernet Shield pins into Arduino UNO board show in below:

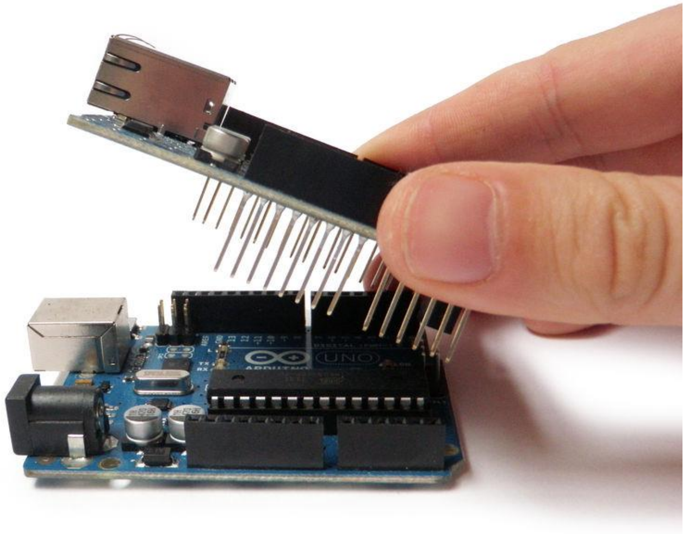

Fig 4.5: Connecting Ethernet Shield with Arduino

4.2.5 Arduino Ethernet Shield with Internet Cable/ Wi-Fi Router

Here we connect the Arduino Ethernet Shield with internet cable or can use Wi-Fi access point via RJ 45connector and UTP cable. A USB port is connected with Arduino whether opposite point of the USB port is connected with PC to configure it and the Ethernet Shield into Wi-Fi access point.

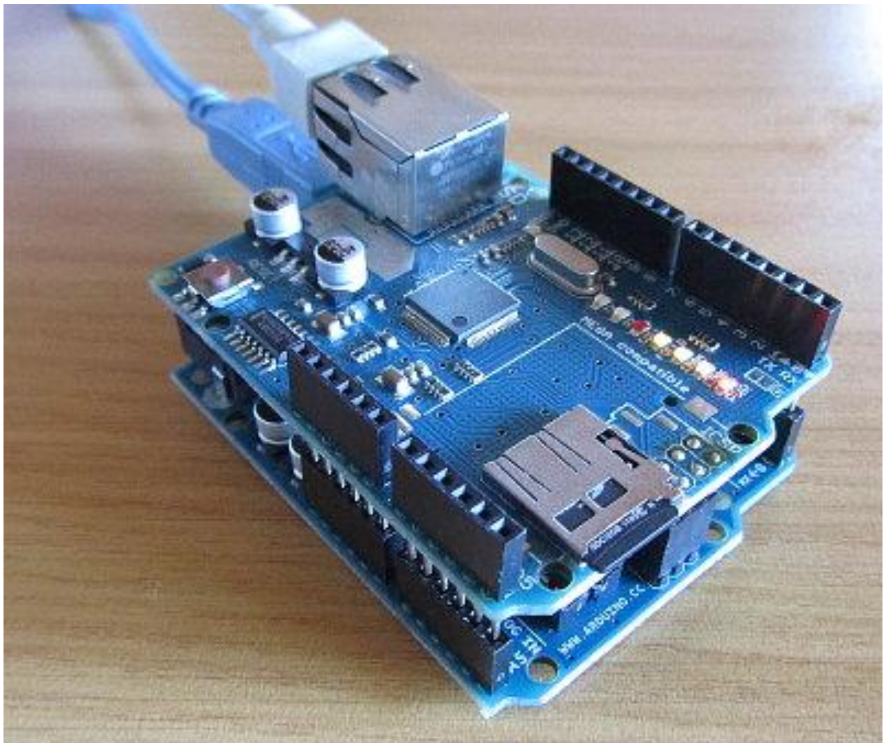

Fig 4.6: Connection of Arduino Ethernet Shield with Internet access point.

4.2.6 Power Source Connection with Servo Motor

In this project we use a Servo Motor to control door or entrance security system. The main operation of servo motor is to follow user instruction; it's mainly rotate in both direction at a fixed angle by user command and can be used to control a lot of circuits. The servo motor can control the door whether it is lock or unlock at a high power maintained by this electric motor.

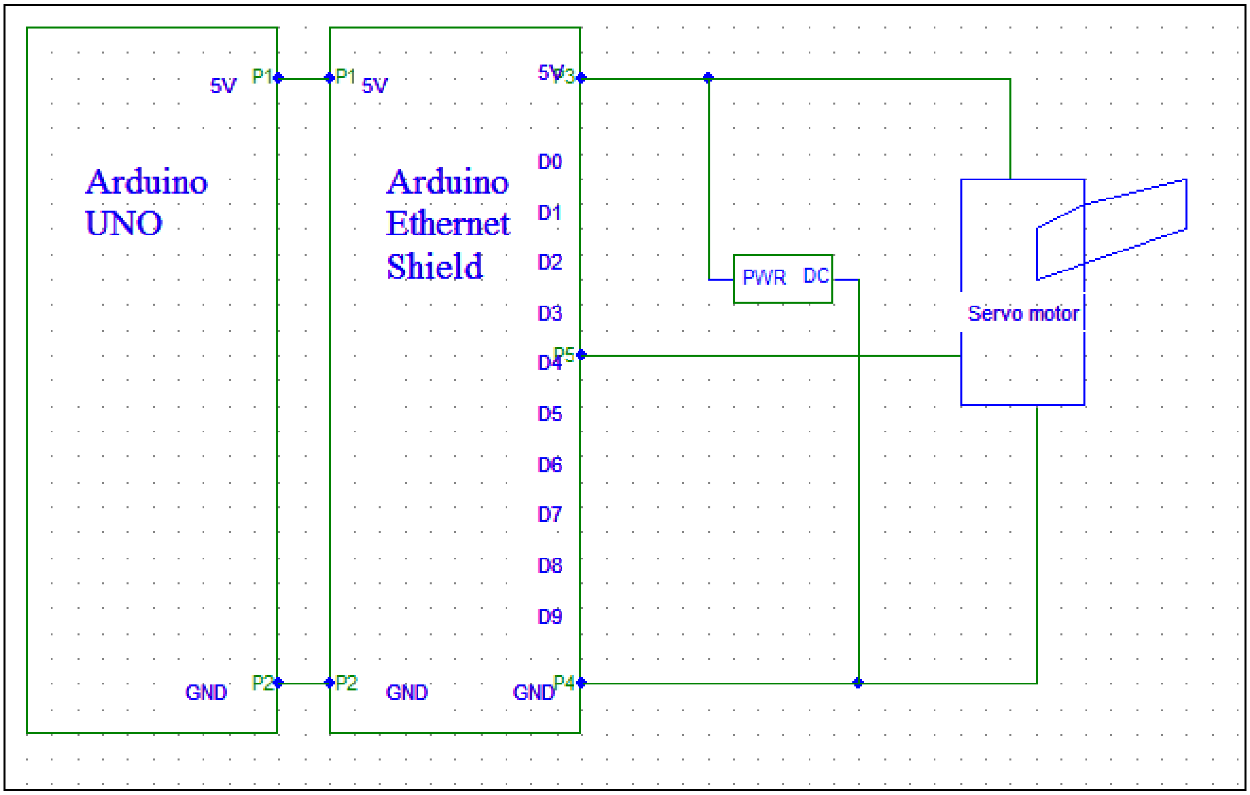

Fig 4.7: Connection to Servo Motor with Power Source

Motors are mainly use for control a system. Here we connect Arduino pin 3 for signaling data send from arduino with Ethernet Shield and send the command to servo motor through the net. The positive and negative (GND) wires are connected into the breadboard. A GND wire is connected from Arduino or Ethernet Shield to the breadboard, so that the board can get the actual voltage. Otherwise the systems cannot response. Servo motor has three pins; tow pins are connected at the positive and negative point on the board. Third pin is connected with shield for signaling the load command. Whenever command will pass from the web the motor works rotationally as user direction. These motors can move into forward direction and also into backward direction.

4.2.7 Motor Characteristic Curve

Using the asynchronous servomotor CV100M4 from SEW-EURODRIVE, important data for this project planning including the motor characteristic curve will be looked at in more detail below. Usually, the following motor data is known:

Motor type	: CV100M4
Rated speed N_{rated}	: 2100 1/m
Rated torque M_{rated}	: 15Nm
Rated current I_{rated}	: 8.1 A
Transition speed n_{trans}	: 1760 1/m (together with a 4-kW servo inverter)

Special attenuation should be paid to the transition speed during project planning. The transition speed is the speed up to which the maximum torque is available for the utilizing the maximum servo inverter peak current. If the motor is operated above the transition speed, the available torque is greatly reduced. This can be clearly seen in the following figure:

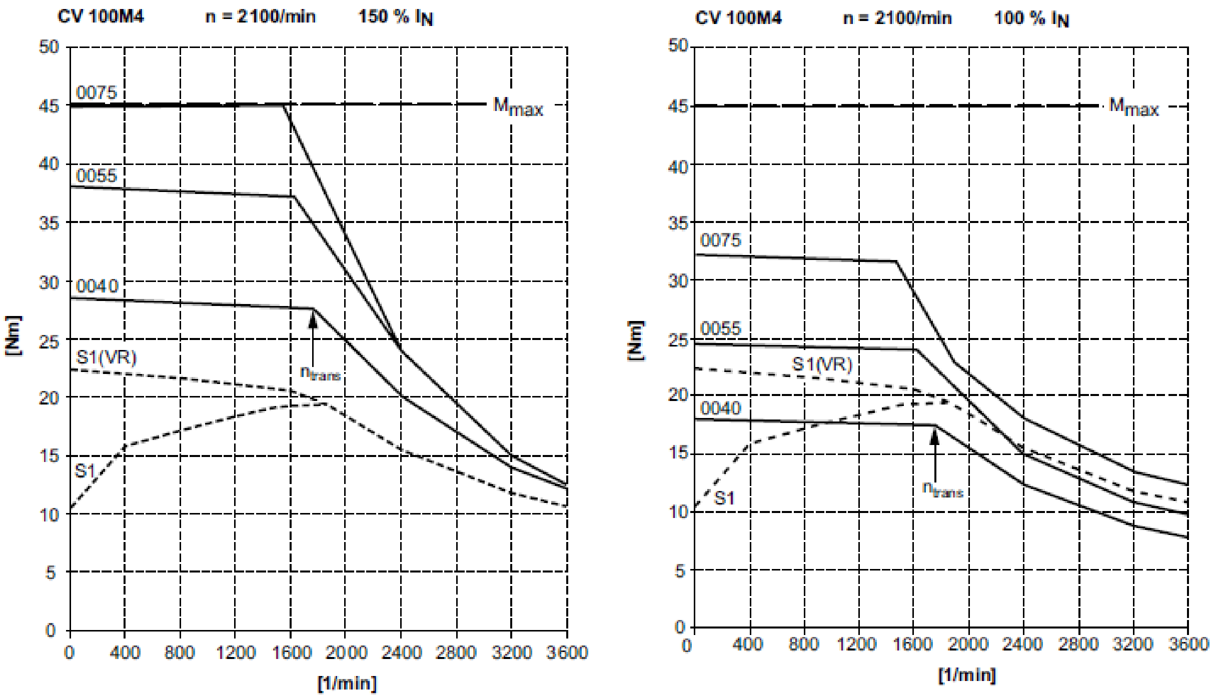

Fig4.8: Characteristic curves of asynchronous servo motor

4.3 Software Design

Software design is divided into two parts. First we write the Arduino program in Arduino software. Then we compile it to the Arduino hardware. This Arduino command is control the Arduino hardware and other circuit connection. For making connection between Arduino and Wi-Fi enable device we need to install Arduino SD card that host a web page that displays to the browser.

4.3.1 Installing Arduino

Arduino runs on Windows. Go to the Arduino software web site at http://arduino.cc/en/Main/Software and download the version of the software compatible with our system. We use Arduino 1.0.5 version.

4.3.2 Verifying the Hardware

Now that we have the Arduino IDE software installed, let’s connect the computer to the Arduino board, load a small program, and verify that all components are working together. First, need to connect the USB cable to our mc board and then plug the other end of the USB cable into our computer.

4.3.3 Arduino Language

The Arduino language is implemented in C/C++ and based in Wiring. When we write an Arduino sketch, we are implicitly making use of the Wiring library, which is included with the Arduino IDE. This allows us to make run able programs by using only two functions: setup () and loop (). As mentioned, the Wiring language is inspired by Processing, and the Arduino language structure is inherited from the Processing language, where the equivalent functions are called setup (). We need to include both functions in every Arduino program, even if we don’t need one of them. Let’s analyze the structure of a simple Arduino sketch using again the Blink example.

Chapter 5

IMPLEMANTATION AND RESULT

5.1 Implementation

All the parts are connected as circuit design. Then we upload the programming code in the Arduino and we get positive result. It works properly according to our design. After connecting the whole system we think about our main project. For testing of the circuit connection we used a LED with 1k resistance. We install a sufficient program for Servo motor rotate. First we tried to give connection without motor for safety purposes. To build the circuit, connect one end of the resistor to Arduino pin 13. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the Arduino GND. Here we represent the photo of this connection.

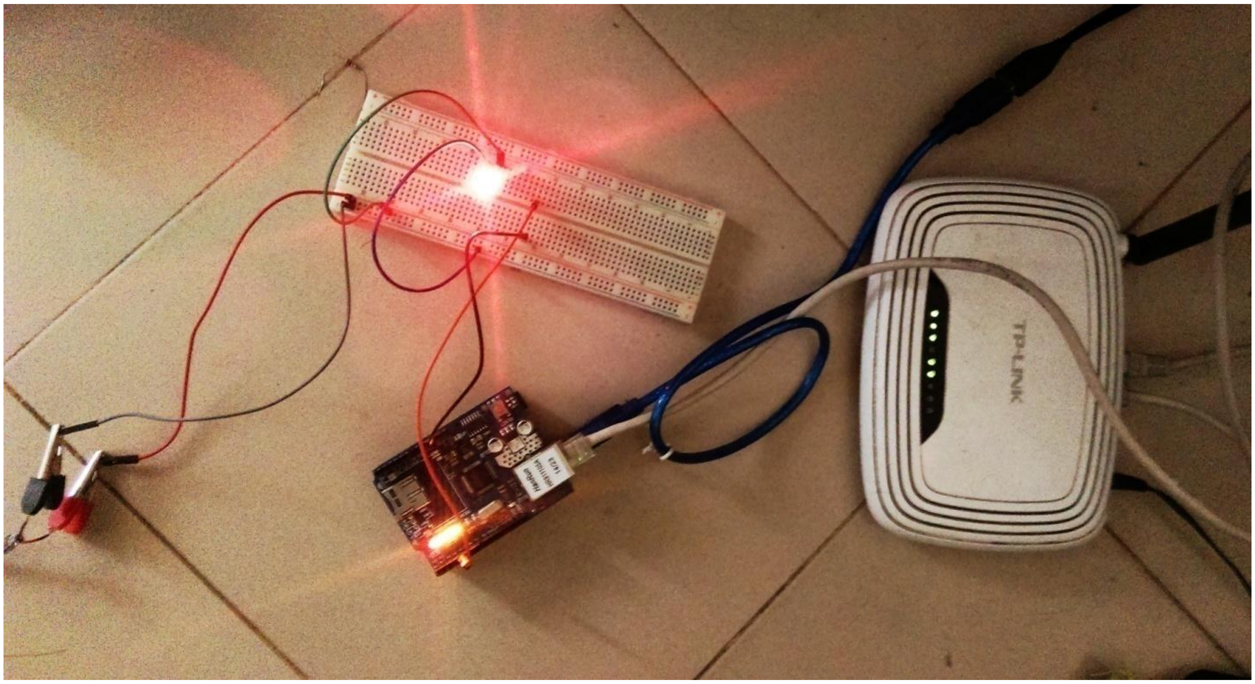

Fig 5.1: Connection without Servo

5.2 Development of the Whole System with load

We developed the main connection of this project. We said as last part about without Servo connection of the motor, now we discuss the connection with fan, lights, AC, refrigerator etc. Here for AC load we use Arduino Relay Module. We also use some sensor for read the analog output. After all successful connection we will get output by controlling every device. We can control fan and lights by using internet connection or Wi-Fi. We represent the photo of this connection in below:

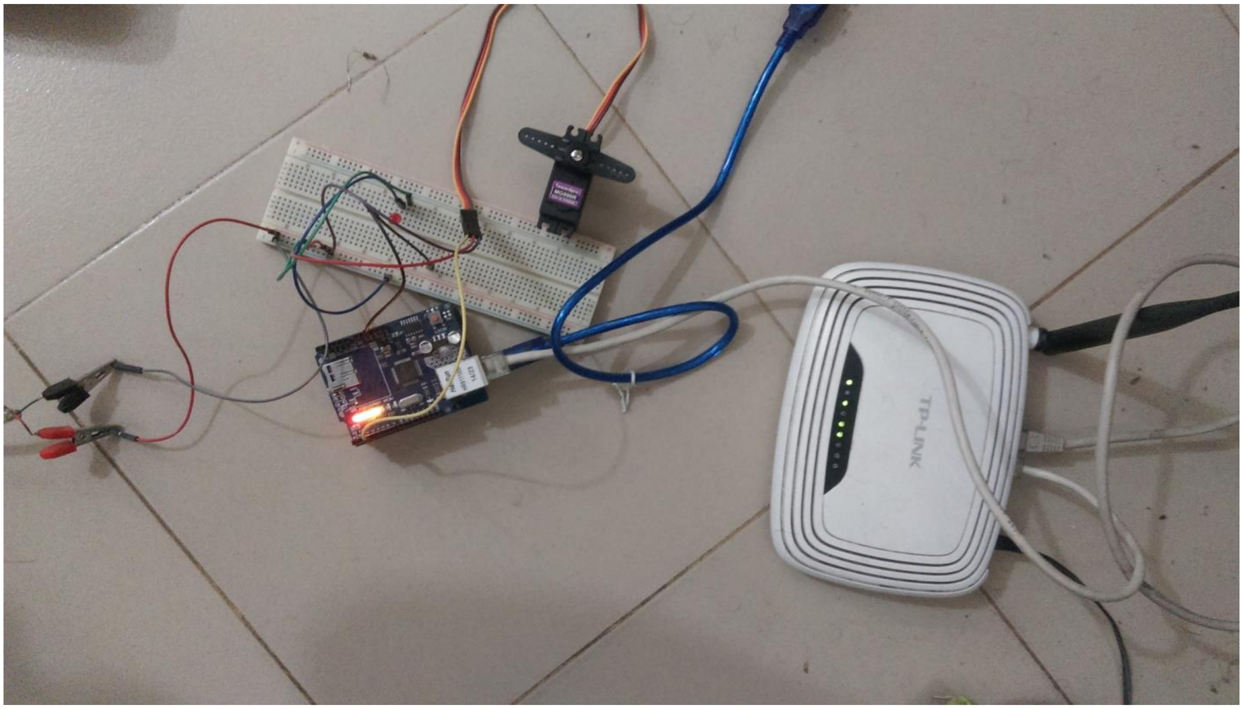

Fig 5.2: Connection with servo (servo on)

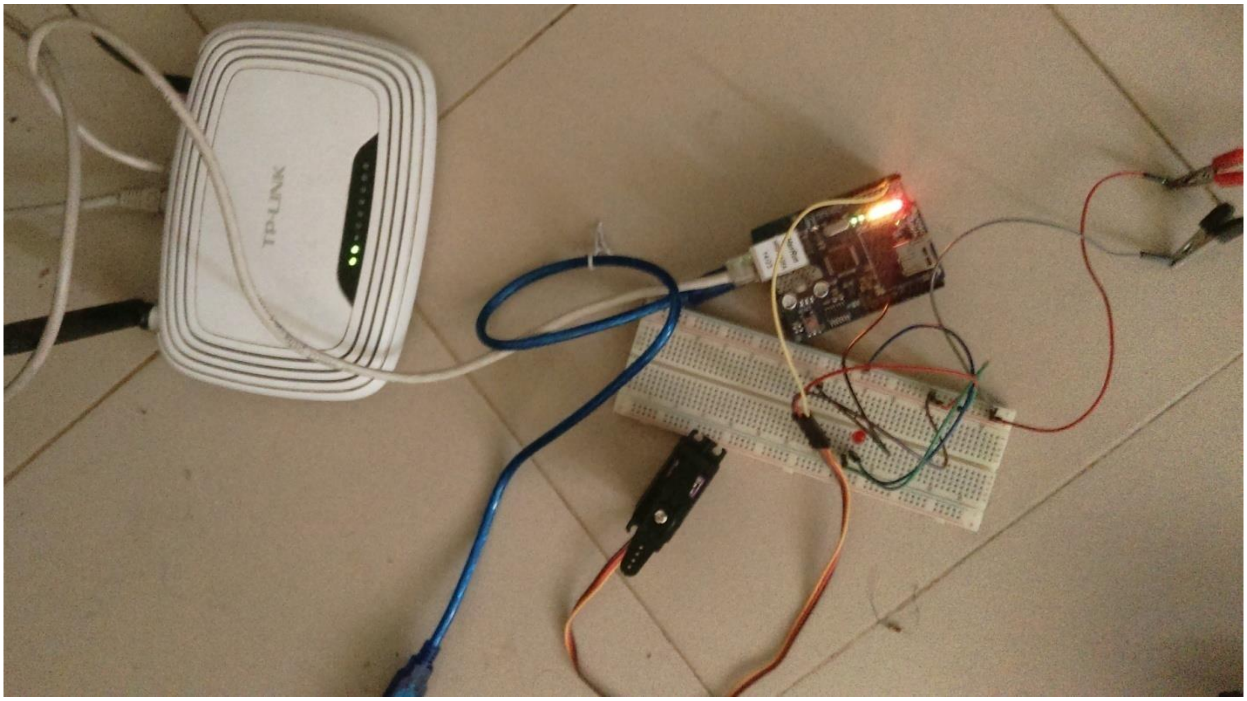

Fig 5.3: Connection with servo (servo off)

Chapter 6

Market Analysis

6.1 Our market in general terms:

Security, as such, is becoming a huge market worldwide; mainly due to the rising crime rates in different parts of the world. As a result, the electronic security equipments are expected to find potential opportunities in this market. The penetration level of security solutions in residential sector is witnessing a rapid growth since the last few years. There are several factors that can be attributed to this growth; such as increase in home burglary rates, attractive insurance policies to residents for installation of security solutions, growth in iPads and tablets markets, and more.

People want to be secured in every place but home is highest priority. Every people want to be safe in their home. Our product ensures people home security. It is quite smart and compatible with the latest market.

Traditional home security system involves having lock on the door or security guards which can prevent burglary in some cases. In most of the case, traditional security systems provides some problem which is not enough secure for the entrance and sometimes we cannot control it at the outside. Our project (Ethernet shield based servo motor control system for entrance security using internet) is Servo motor based control system. It can control by the owner of his/ her house entrance.

So it can be a very reliable and competitive product in the latest market.

6.2 Our business contributes in the local economy system:

Security measures aiming to manage crime risk impose economic effects on society. Urban security has a cost at the point of build or refurbishment, and typically, the benefits come later. There are several types of security measures. Some of these measures involve monetary expenditures (e.g. locks, money safes, alarm systems, security guards, insurances, etc.) .Publics are investing a huge amount of money on these security systems.

Security measures intend to increase the level of security, deterring crime, and/or at least mitigating the negative economic impact of crime. In essence, they increase actual and/or perceived risk to the offender and his or her effort, changing the behavior of the criminal. Put differently, anti-crime security measures positively influence the costs of crime.

Improvements in security have similar indirect (secondary) economic effect as an improvement in urban environment, thereby raising the real wealth of households, companies and public authorities in the affected area. Specific examples of indirect effects are:

- A reduction in insurance premiums
- The opportunity cost of police presence
- An increase in tourist income and tourist employment
- A decrease in long-term health costs (due to the reduction in victim-hood and Increase in psychological well-being)
- A reduction in the carbon cost of crime
- A reduction in costs of securing void areas
- Less people who feel forced to move to another home

6.3 Target Consumer Group:

Here we are going to talk about the customers we are going to target and their trends and include figures and predictions for future growth.

To keep the society violation free security is necessary. As our society reaches an advanced stage of social development and organization, security system should be developed as well. Our product is very modern and can be a great solution for security purpose.

We are targeting customers in urban areas. Because, in urban areas peoples are more conscious about their security rather than the rural areas. They are educated, smart so they want their home smart too. To make a home smart modern security system is must. Our product can be quite attractive to them as it is smart and reliable. And recently shop lifting is a major crime, people are aware of security of their shops too. Our product can be applicable in corporate places, hospitals, and personal locker room also.

With respect to different types of homes wherein the security solutions are utilized, independent homes are observed to hold the highest share, i.e. 80.1% in 2011. However, the adoption rate is expected to grow rapidly in apartments in near future owing to the rise in nuclear families.

6.4 Practical Market on Servo Motor

The global servo motors & drives were valued at USD 8.64 Billion in 2014 and is projected to reach USD 12.56 Billion by 2020, at a CAGR of 6.1% from 2015 to 2020. Factors such as increasing technological advancement, rapid growth in the automation market, growing demand for energy efficient solutions, enhanced production capacity, high level security, increased venture capital funding, and initiatives taken by government bodies for industrial growth led to the development of automation and industrial control systems across the globe are driving the growth of this market.

Servo motor is an efficient, accurate, and flexible motor that helps the system to work efficiently. It increases the entire system performance. It is considered to be one of the highly expensive motors when compared to stepper motors. However, in the recent times there has been a decline in prices of the components (particularly rare earth permanent magnet material) used in the manufacturing of servo motors. This is turn is increasing the growth of this market.

The global servo motors market has been segmented on the basis of type, component, voltage range, application, and geography. Based on type, the market is segmented into AC servo motors/drives and DC servo motors/drives. In terms of supply voltage range, the market is segmented into low and medium. On the basis of component, the market is segmented into sensors, controller modules, encoders, amplifiers. The servo motors market accounted for the largest share of ~55% in 2014.

The market on the basis of geography has been classified into four regions, namely Americas, Europe, Asia-Pacific, and Row. APAC is expected to contribute the maximum revenue towards servomotors and drives market. Europe is the second-fastest growing market after APAC.

Servo motors and drives are used in various applications, such as automotive & transportation, semiconductor & electronics, food processing, textile machines, healthcare, packaging, material handling, machine tools, and many more. Major applications, such as automotive & transportation, semiconductor & electronics, food processing, and textile accounted for a share of ~47% of the global servo drives market.

Some of the major players operating in the global servo motors market include Yaskawa Electric Corporation (Japan), Mitsubishi Electric Corporation (Japan), Siemens AG (Germany), Schneider Electric (France), Rockwell Automation Inc. (U.S.), ABB ltd. (U.S.), KINCO Automation Pvt. Ltd (China), Delta Electronics, Inc. (Taiwan), Emerson Electric Co. (U.S.), and Attach (SHENZHEN) Technology Co. Ltd. (Japan). Most of the companies have adopted various strategies, such as mergers & acquisitions and partnerships, among others to sustain in the servo drives market.

6.5 Customization Options:

Along with the market data, Markets and Markets offer customizations as per the company’s specific requirement. The following customization options are available for the report.

Product Analysis

- Usage pattern (in-depth trend analysis) of products (segment-wise)
- Product matrix, which gives a detailed comparison of product portfolio of each company mapped at country and sub-segment level
- End-user adoption rate analysis of the products (segment-wise and country-wise)
- Comprehensive coverage of product approvals, pipeline products, and product recalls

Technology Data

- Country-specific data of the market segments such as type, component, and application
- Premium insights of the servo motors market

Comparative Analysis

- Market data of top companies
- Key developments of top companies

Appendix

```
#include <EEPROM.h>
#include <Servo.h>
#include <Wire.h>
#include <Teleduino328.h>

#include <SPI.h>
#include <Ethernet.h>

// User configurable variables
byteuseDhcp = true;
byteuseDns = true;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0x02 };
IPAddressdeviceIp(192, 168, 1, 100); // Only if useDhcp is false
IPAddressgatewayIp(192, 168, 1, 1); // Only if useDhcp is false
IPAddressdnsIp(192, 168, 1, 1); // Only if useDhcp is false
IPAddresssubnet(255, 255, 255, 0); // Only if useDhcp is false
IPAddressserverIp(173, 230, 152, 173); // Only if useDns is false
charserverName[] = "us01.proxy.teleduino.org"; // Only if useDns is true
unsignedintserverPort = 5353; // Can be set to either 53 or 5353
bytestatusLedPin = 8;

// User configurable key, this is used to authenticate with the proxy server
// This is checked against the EEPROM on boot and written if necessary
// The proxy server retreives the key from the EEPROM
byte key[] = { 0xAA, 0x5C, 0x69, 0xBF,
               0xEC, 0x14, 0xD8, 0xA5,
               0x4D, 0xCC, 0x48, 0x7B,
               0x56, 0xC5, 0xC2, 0x05 };

// Other required variables
byte data[257];
bytedataLength;
bytehexStage;
unsigned long lastInstruction = 0;
unsigned long lastRefresh = 0;
byte stage = 0;

// Declare client object
EthernetClient Client;

void setup()
{
  // Load presets
Teleduino328.loadPresets();

  // Set status LED pin
Teleduino328.setStatusLedPin(statusLedPin);
```

```
Teleduino328.setStatusLed(1); // Initialisation
  // Check the EEPROM header and check to see if the key is correct
  // This is to ensure the key is not cleared from the EEPROM
if(EEPROM.read(0) != '#')
  {
EEPROM.write(0, '#');
  }
if(EEPROM.read(1) != 0)
  {
EEPROM.write(1, 0);
  }
if(EEPROM.read(2) != '#')
  {
EEPROM.write(2, '#');
  }
if(EEPROM.read(160) != '#')
  {
EEPROM.write(160, '#');
  }
for(byte i = 0; i < 16; i++)
  {
if(EEPROM.read(161 + i) != key[i])
    {
EEPROM.write(161 + i, key[i]);
```

```
    }
  }
if(EEPROM.read(177) != '#')
  {
EEPROM.write(177, '#');
  }

  // Start network and attempt to connect to proxy server
Teleduino328.setStatusLed(2); // Network configuration
if(useDhcp)
  {
if(!Ethernet.begin(mac))
    {
Teleduino328.setStatusLed(2, false, 10000);
Teleduino328.reset();
    }
  }
else
  {
Ethernet.begin(mac, deviceIp, dnsIp, gatewayIp, subnet);
  }
delay(1000);

Teleduino328.setStatusLed(3); // Connect to server
```

```
if((useDns&& !Client.connect(serverName, serverPort)) || (!useDns&& !Client.connect(serverIp,
serverPort)))

  {

Teleduino328.setStatusLed(3, false, 10000);

Teleduino328.reset();

  }

lastInstruction = millis();

}


void loop()

{

if(Client.connected())

  {

    // What we need to do depends on which 'stage' we are at

switch(stage)

    {

case 0: // Wait for start byte

if(Client.available())

        {

char c = Client.read();

if(c == '?')

          {

stage++;

          }

        }
```

```
break;
case 1: // Reset variables
dataLength = 0;
hexStage = 0;
stage++;
break;
case 2: // Instruction byte
if(Client.available())
     {
char c = Client.read();
if(c == '?')
       {
stage = 1;
break;
       }
else if(c == '\r' || c == '\n' || c == '.')
       {
stage = 0;
break;
       }
if(!hexStage)
       {
data[0] = Teleduino328.hexDecode(c) * 16;
       }
else
```

```
        {
data[0] += Teleduino328.hexDecode(c);
        }
hexStage = !hexStage;
if(!hexStage)
        {
stage++;
        }
      }
break;
case 3: // Data length byte
if(Client.available())
      {
char c = Client.read();
if(c == '?')
        {
stage = 1;
break;
        }
else if(c == '\r' || c == '\n' || c == '.')
        {
stage = 0;
break;
        }
if(!hexStage)
```

```
        {
data[1] = Teleduino328.hexDecode(c) * 16;
        }
else
        {
data[1] += Teleduino328.hexDecode(c);
        }
hexStage = !hexStage;
if(!hexStage)
        {
stage++;
        }
      }
break;
case 4: // Data
if(Client.available())
      {
char c = Client.read();
if(c == '?')
        {
stage = 1;
break;
        }
else if(c == '\r' || c == '\n' || c == '.')
        {
```

```
if(dataLength == data[1])
        {
stage++;
break;
        }
else
        {
stage = 0;
break;
        }
       }
if(!hexStage)
       {
data[2 + dataLength] = Teleduino328.hexDecode(c) * 16;
       }
else
       {
data[2 + dataLength] += Teleduino328.hexDecode(c);
       }
hexStage = !hexStage;
if(!hexStage)
       {
dataLength++;
       }
      }
```

```
break;

case 5: // Execute instruction and return result

Teleduino328.instruction(data);

Client.write('!');

for(int i = 0; i < data[1] + 2; i++)

        {

Client.write(Teleduino328.hexEncode(data[i] / 16));

Client.write(Teleduino328.hexEncode(data[i] % 16));

        }

Client.write('\n');

lastInstruction = millis();

stage = 0;

break;

    }

  }

else

  {

Teleduino328.setStatusLed(10);

Teleduino328.reset();

  }



  // Has the instruction timeout been reached?

if(millis() - lastInstruction> 30000)

  {

Client.flush();
Client.stop();
Teleduino328.setStatusLed(9);
Teleduino328.reset();
  }

  // Process refreshes every 50ms
if(millis() - lastRefresh>= 50)
  {
Teleduino328.pinTimers();
Teleduino328.shiftRegisterTimers();
Teleduino328.shiftRegisters();
lastRefresh = millis();
  }

  // Check to see if reset has been requested
Teleduino328.checkReset();
}
```

<u>HTML CODE VIA WEB SERVER:</u>

```
<html>
        <body>
        <h1>Arduino</h1>
                <div>
                        <h2>Servo</h2>
```

```
                              <button><a
href="http://us01.proxy.teleduino.org/api/1.0/328.php?k=AA5C69BFEC14D8A54DCC487B56
C5C205&r=setServo&servo=0&position=15">DOOR OFF</a></button><br /><br />

                              <button><a
href="http://us01.proxy.teleduino.org/api/1.0/328.php?k=AA5C69BFEC14D8A54DCC487B56
C5C205&r=setServo&servo=0&position=75">DOORON</a></button><br /><br />


                    </div>

          </body>

</html>
```

References

1. http://www.safewise.com/home-security-faq/how-do-security-systems-work

2. T.Baranowski, J. Baranowski, K.W. Cullen, T.Marsh, N.Islam, I.Zakeri, L.H.Morreale, and C.deMoor, "Squire's a multimedia game", American Journal of Preventive Medicine, 24(1), 2003.

3. https://www.techopedia.com/definition/28247/internet-of-things-iot.

4. Justin Lahart (27 November 2009). "Taking an Open-Source Approach to Hardware". The Wall Street Journal. Retrieved 7 September 2014.

5. http://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf

6. http://www.epiloglaser.com/legend_linearencoders.htm

7. http://archive.computerhistory.org/resources/access/text/Oral_History/102658328.05.01.acc.pd

8. http://www.engineersgarage.com/articles/servo-motor

9. http://www.farnell.com/datasheets/1682209.pdf

10. http://www.marketsandmarkets.com/Market-Reports/servo-motors-and-drives-market-1127.html

11. http://whatis.techtarget.com/definition/Internet-of-Things

12. https://www.arduino.cc/en/Guide/Introduction

13. https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg

14. https://www.arduino.cc/en/Main/ArduinoBoardUno

15. https://www.arduino.cc/en/Main/ArduinoEthernetShield

16. https://www.arduino.cc/en/Guide/ArduinoEthernetShield