MICROCONTROLLER BASED AUTOMATIC TRAFFIC CONTROL SYSTEM

SUBMITTED BY

MD. AHSANUL HOQUE
ID: 102-33-255

MD. MOSTAFIZUR RAHMAN
ID: 093-33-123

The Project Report submitted in partial fulfillment of the requirements for the Bachelor of Science degree
In
Electrical & Electronic Engineering

SUPERVISED BY

MD. DARA ABDUS SATTER
Assistant professor
Department of EEE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JUNE 2014

DECLARATION

We hereby declare project titled“**Microcontroller Based Automatic Traffic Control system**” has been done by us under the supervision of **Md. Dara Abdus Satter,** Assistant Professor, Department of EEE**,**Daffodil International University. This is our original work and was not submitted elsewhere for the award of any other degree or any other publication.

Date: 24-05-2014

Project Supervisor:

____

Md. Dara Abdus Satter
Assistant Professor
Department of EEE
Daffodil International University

Submitted by:

____

Md. Ahsnul Hoque
Student ID: 102-33-255
Department of EEE
Daffodil International University

____

Md. Mostafizur Rahman
Student ID: 093-33-123
Department of EEE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

We would like to take this opportunity to express our heartfelt gratitude to our supervisor Mr. Md. Dara Abdus Satter sir for his guidance, and instruction that he has given us to complete our thesis project in such a short time. We have been able to learn the skills from him, which have benefitted us immensely and will continue to help us throughout our future endeavors, both academically and professionally. We have been challenged in our education and have grown from it both as a person and as engineers.

We would like to express our heartiest gratitude to Professor Dr. M. Shamsul Alam, Dean, Faculty of Engineering (FE)and Dr.Md.Fayzur Rahman, Head**,** Department of EEE for their kind help to finish our thesis and also to other faculty members and the staffs of EEE department of Daffodil International University. We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant supports and patients of our parents.

ABSTRACT

This project is on “Microcontroller Based Traffic Control System”. This is a kind of Automation system, which helps to control intelligence traffic management by static time scheduling.

The aim of the microcontroller based traffic management system is built up an intelligence management system. After assign a microcontroller generate its code for built a circuit design into simulation software, loading this code into microcontroller and simulate it. If it is work properly than we burn a specific microcontroller and make a module of road and interface with the circuit board. If it is worked appropriately than we can implement this system.

The traffic control is a system which ensures the safety of pedestrians from accident and maintained the direction of vehicles on the road. This is also save the life of general peoples. Bangladesh want an automatic traffic management system because it’s heavy populations and heavy traffic jam. Our proposed work will manage the existing traffic system to follow an automatic real time situation regarding situation based system. We analyzed various traffic places in different times in a whole day and making a daily traffic schedule of that traffic place and also included an imaging system. It is also collects information and stored into a server. It is a very simple to control the traffic light uses schedule timer for each phase but real time image processing management is too difficult but we able to do microcontroller based traffic control system.

After implementation of all functions, the system is tested in different stages and its work successfully as Traffic Control System.

TABLE OF CONTENTS

CONTENTS **PAGES**

CHAPTERS

LIST OF FIGURE

LIST OF TABLE

Chapter 1

Introduction

1.1General Introduction

The population of the modern cities or urban is increase day by day which vehicular travel is increasing and lead to congestion problem. Traffic congestion occurred many critical moment and people also face many complex event on the road. It is the causes of traffic jam, accident and many others complex situation. For this traffic jam the people loss their valuable time, fuel and it are also effect on our environment. The oil is our national asset, we loss this asset for causes traffic jam. This time we have to need an intelligence traffic management system but we have not any existing system now it maintained only situation based system manually using by hues number of police. Our proposed system will control this system with reduce the number of traffic police, controlled automatically.

1.2 Overview

The technique of traffic control system in Bangladesh is not well organized and automated. The traffic police maintained that manually. This is not well suited were as most of the development country uses intelligence automated system and Bangladesh control it manually. We have not any discipline in this existing system. We were not following any rules; the traffic police maintain that given signal over his hand for that some time they fetch complex situation and it causes accident. It had not any information collection system.

1.3 Objective

The proposed system will reduce the traffic jam because we analysis the specific traffic signal various time in a day and make a schedule for that place and it controlled by following that schedule and also with image processing. It will reduce the number of uses traffic police because this mechanism handled the signal atomically so we need only a traffic police into each traffic place and it will also reduce the complexity of the traffic point. This mechanism will save the valuable time of the general public with its traffic jam free road and people finish their daily work with his great satisfaction.

1.4 Background of the Project

This traffic management system will control the traffic signal automatically followed by time base (scheduling) system as well imaging. Help the people to traveling or moving one place to another without traffic jam over road. The system included online traffic signal monitoring, vehicle tracking system and all of the information store into a central database for that any time tracing any one of vehicle. The traffic police or any police officer traces the vehicle which is involved into different types of crime. It is also helpful to our crime branch and they can also help to the general people frequently.

1.5 Methodology of the Project

In this mechanism we use an automated traffic management system that is maintained the traffic signal following the time schedule intelligence by microcontroller. We generate a C-language code including different time schedule in a day and it work as following that time schedule. It has also included an image processing system which collect image by CC Camera processed that and maintained real time situation. This is an intelligence technique so we need only one traffic polices on each traffic signal only for monitor it and maintained any exception event.

1.6 History of Traffic Police in Bangladesh

The traffic police in Bangladesh have a history is a long ago. Bangladesh traffic police establish before 200 years and launching during the British colonial period followed by the Pakistani era and the emergence of independent Bangladesh in 1971. During the rule of Muslim Sultanate in Bengal between 14th and 16th century AD, the police system underwent some changes but it was during the subsequent Mughal reign that it entered a new phase through expansion. Although there were no professional police force similar the British during the Mughal era, an orderly the force was in motion to maintain law and order throughout the land. Radical changes were brought about this administration by the British colonial rulers the police reform act 1782. Then at1857 the ‘sepoy mutiny’ view that the police administration was not very effective. then established a uniform and effective police administration in some provinces of Indian subcontinent including Bengal in line with the Royal Irish Constabulary system.

1.7 Existing Traffic Management System

At present we have not any existing automated system the traffic police maintain it manually given signal over his hand. They have not any existing data stored system. The traffic police maintain only showing his hand so we need many police in one signal people violate traffic rules and some time they fetch complexity moment. It is the causes of accident and creates a traffic jam. If you go to a traffic point you saw that has traffic signal light but it is unused some time the traffic police avoid the traffic light and it is also maintained manually. Now this is a national problem of our country. Some traffic point has many CC Camera but that is disconnected or which connected that is only capture seen of accident.

1.8 Problem Finding

Our existing system does not follow any automated system it following situation based manual system. A traffic police control the traffic signal given signal up to his hand and that was needed hues number of traffic police for a traffic signal. Some they cannot take up the situation and occurred traffic jam and some time they fallen a complex situation (Fig: 1.1). The general peoples are fallen problem and they were loss there much valuable time on the way. For that they were mistake their work schedule.

Fig 1.1: Complex situation of traffic management system.

1.9 Benefit of the Project

This automated project will reduce the traffic jam and also reduce the number of traffic police. It helps to the general people of our country to jam free moving on the street. It helps to our traffic police, elite force and crime branch to catch crime or involve vehicles. The system also stores all the information that we needed to different time of various types of forces for different kinds of investigations. Our country people will be most benefited with this project because traffic jam is a nation problem at present in our country. When it reduces the society people can move one to another with a jam free road. They can also maintain their work schedule frequently and given appropriate value of their valuable time.

1.10 Organization of the Report

This project is organized in chapters. Chapter 2 presents Theoretical background of the project. In Chapter 3 describes the System Design and Development. Chapter 4 provides the Result and Simulation and Chapter 5 represents the conclusion and future work.

1.11 Conclusion

Our country at present going to be digitalized but our traffic management system is not digitalized. Before some days it was maintained by a time based management system but at present it has not any existing automated system. Many of the time the traffic signal lights are totally stopped then it is maintained by traffic police manually. This is not fare for our country.

Chapter Two
Theoretical Background

2.1 System Overview

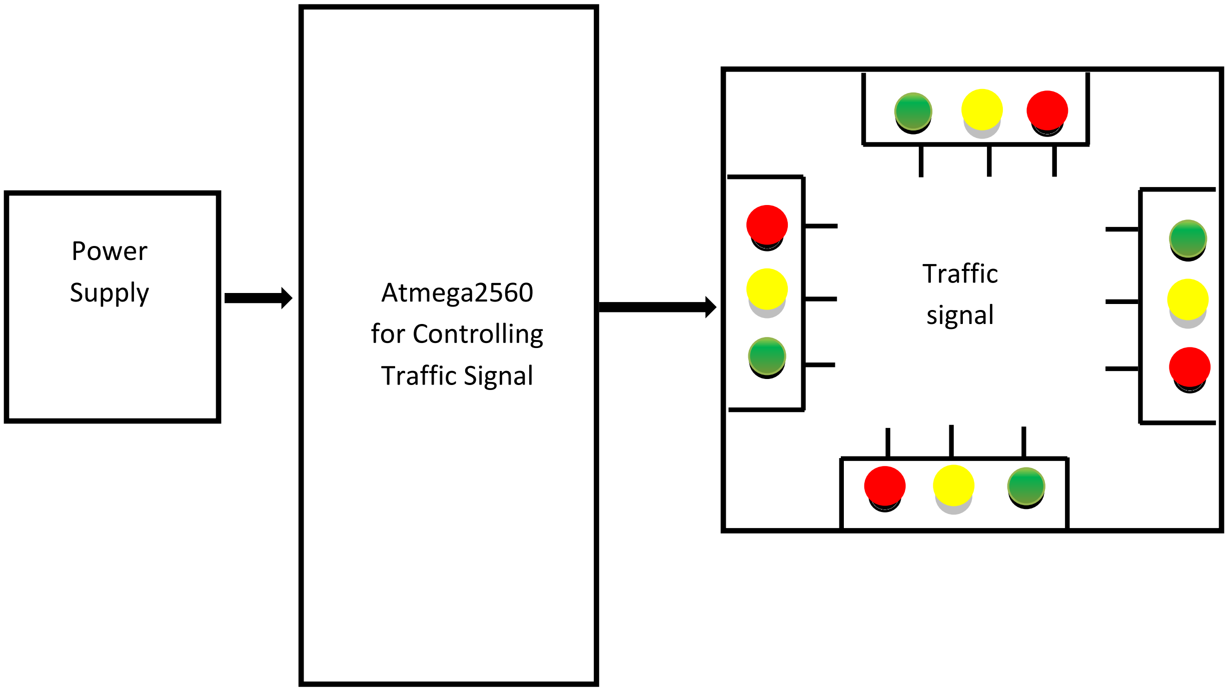


Figure: 2.1: System overview

Figure: 2.1 show the proposed traffic control system overview. Here we usedArduino Mega 2560 Microcontroller for controlling traffic signal. For the proposed traffic control system we used fixed +5V power supply system using a voltage regulator. The microcontroller used for traffic light signal controlling performs the whole operation by means of a program developed for Arduino Mega 2560 Microcontroller using C Programming language.

2.2 LM7805 Voltage Regulator

A voltage regulator is used to produce a constant linear output voltage. It’s generally used with AC to DC power supply. And also it can be used as well as a DC to DC voltage converter. To regulating low voltage, most used device is one single IC. 7805, 7812, 7905 etc. 78xx series are design for positive and 79xx series are for Negative voltage regulator.7805 is a three terminal +5v voltage regulator IC from 78XX chips family. See 7805 pin out below

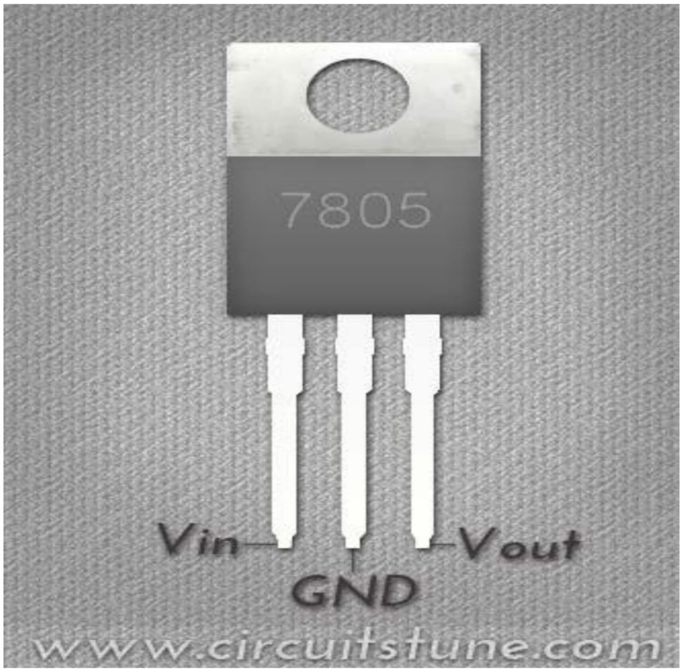


Fig: 2.2: Pin out of 7805

Its output voltage is +5V DC that we need. Any voltage can be supplied in input; the output voltage will be always regulated +5V. But it is recommended to supply more than 18V or less than 8V in input. There used two capacitors in this voltage regulator circuit, they aren’t mandatory to use. But it will be best if you use them. They helped to produce a smooth regulated voltage at output. Use electrolyte capacitor instead of ceramic capacitor. The following figure shows the circuit diagram for fixed +5V output which we need to operate the microcontroller

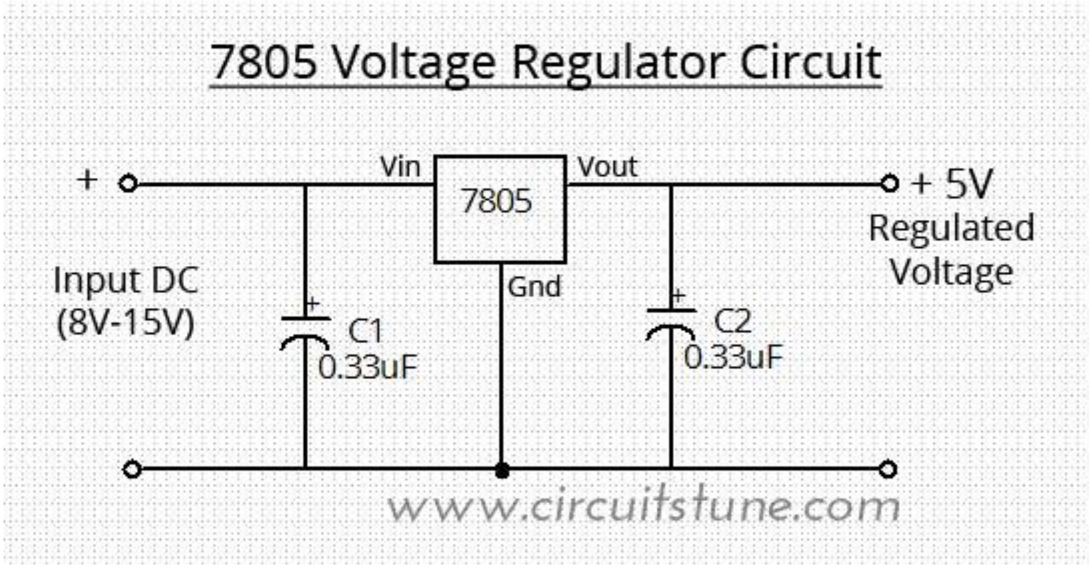


Fig: 2.3: Circuit diagram of 7505 Voltage Regulator for fixed +%V Output

2.3 Arduino Platform

Arduino is an open source electronics prototyping platform based on flexible hardware and software. The Arduino is a simple yet sophisticated device which is based on At-mel'sATmega microcontrollers. The Arduino software is supported by Windows, Macin-tosh OSX and Linux operating systems despite the fact that most microcontrollers are limited to Windows operating system. The software language is based on AVR C programming language and can be expanded through C++ libraries. There are various types of Arduino microcontroller board available in the market including the Arduino kits and Arduino shields.

2.3.1 Arduino Mega 2560 Board

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

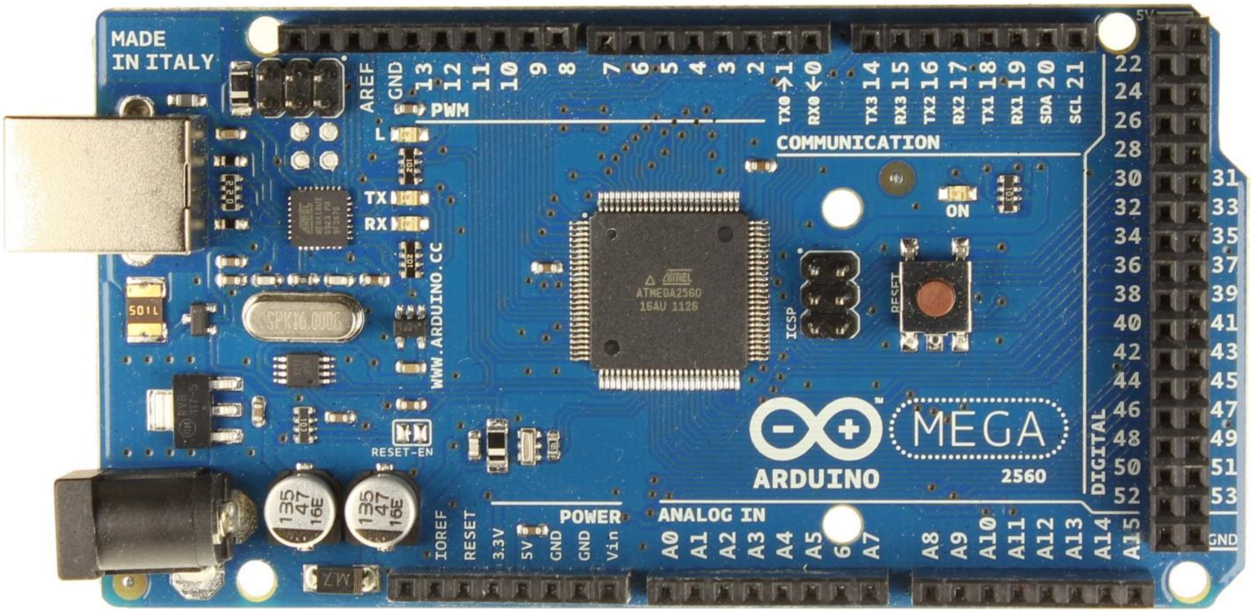


Fig: 2.4: ArduinoMega 2560

The Mega 2560 is an update to the Arduino Mega, which it replaces.The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 boards) programmed as a USB-to-serial converter.

2.3.2 Pin Mapping of Mega 2560

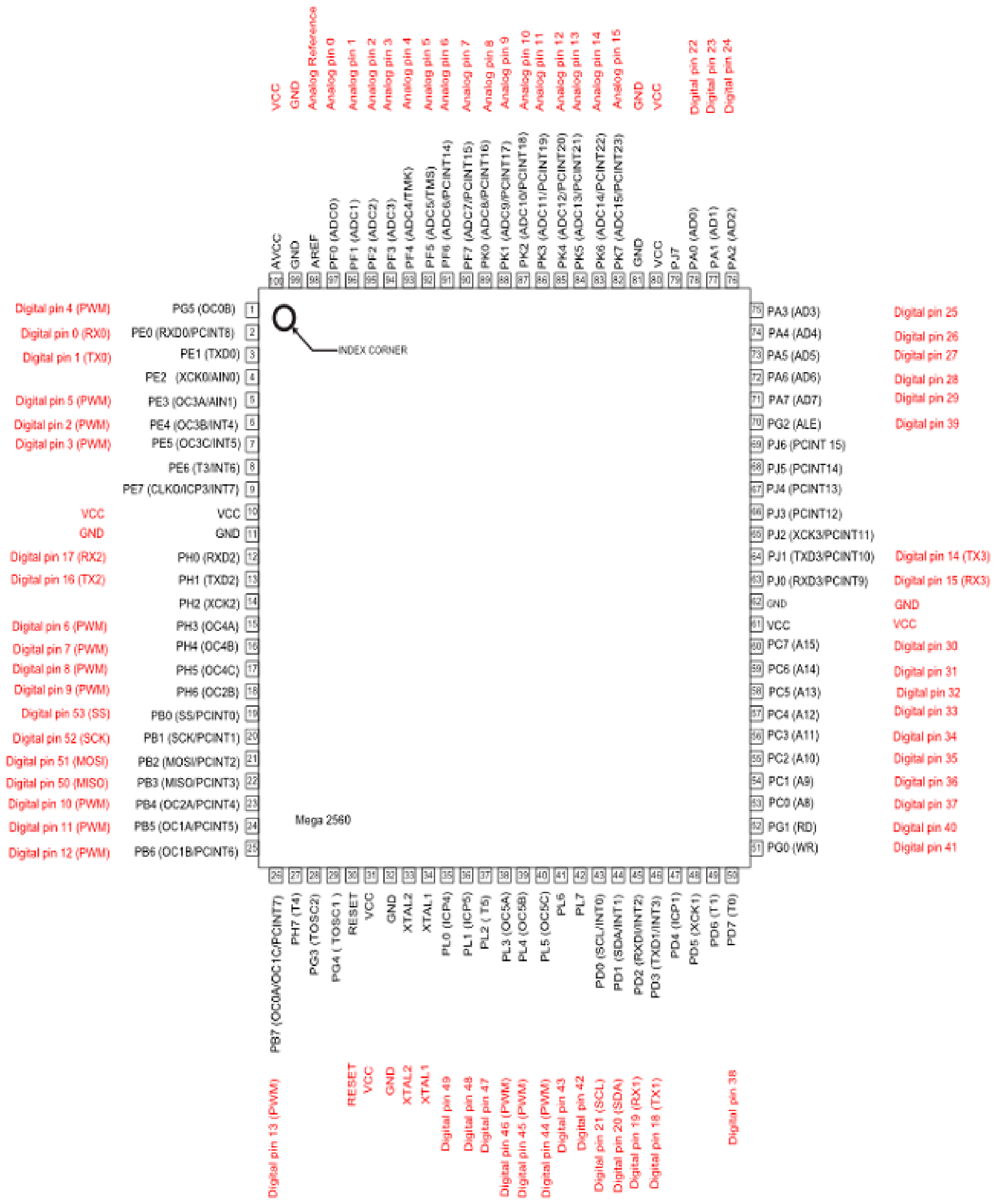


Fig: 2.5: Ardiuno Mega 2560 pin mapping diagram

Table 1: pins summarized as follow:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

2.3.3 Power

The Arduino Mega 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 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 for working with the 5V or 3.3V.

2.3.4 Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM Library).

2.3.5 Input and Output

Each of the 54 digital pins on the Mega 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 kohms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). 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 attachInterrupt() function for details.

PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

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.

TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provides 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 analogReference() function.

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.

2.3.6 Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTsfor TTL (5V) serial communications. An ATmega16U2 (AT mega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .info file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via theATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

2.3.7 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

2.4 ATmega2560 Microcontroller

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

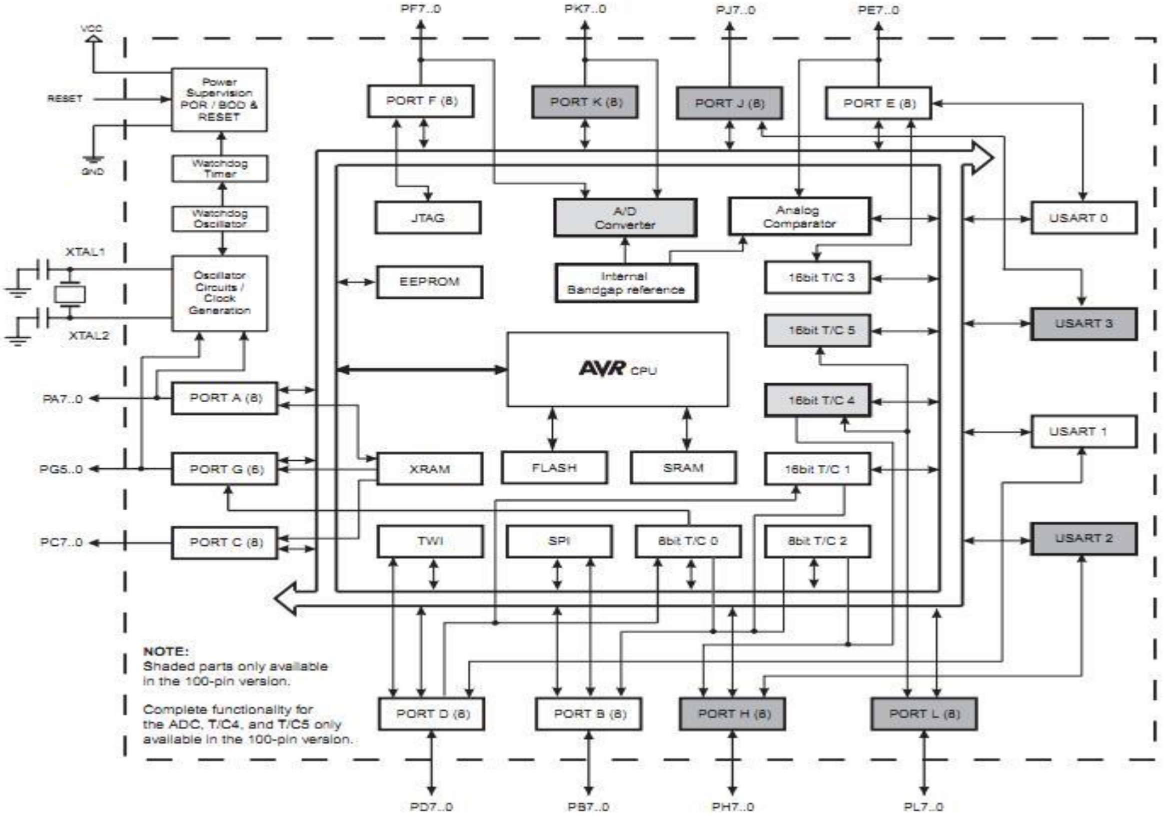
The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Decimal. Please note that I^{2}C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Decimal (analog inputs 4 and 5).

2.5 AVR Atmega 2560 Microcontroller

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing thsystemdesigner to optimize power consumption versus processing speed.

2.5.1 Block Diagram

Figure 2-1. Block Diagram



NOTE:
Shaded parts only available in the 100-pin version.

Complete functionality for the ADC, T/C4, and T/C5 only available in the 100-pin version.

Fig 2.5.1: Block diagram

The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega640/1280/1281/2560/2561 provides the following features: 64K/128K/256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 8Kbytes SRAM, 54/86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters with compare modes and PWM, four USARTs, a byte oriented 2-wire Serial Interface, a 16-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE® std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM; Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the Touch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully denounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The easy-to-use Touch Suite tool chain allows exploring, developing and debugging applications. The device is manufactured using the Atmel high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega640/1280/1281/2560/2561 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications. The ATmega640/1280/1281/2560/2561 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.6 Light Emitting Diode (LED)

A "Light Emitting Diode" or LED as it is more commonly called, is basically just a specialized type of PN junction diode, made from a very thin layer of fairly heavily doped semiconductor material. When the diode is forward biased, electrons from the semiconductors conduction band recombine with holes from the valence band releasing sufficient energy to produce photons which emit a monochromatic (single color) of light. Because of this thin layer a reasonable number of these photons can leave the junction and radiate away producing a colored light output.

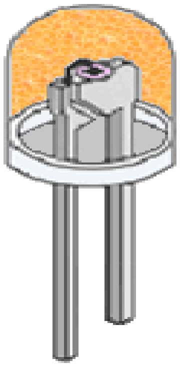


Fig: 2.6: LED Construction

Then we can say that when operated in a forward biased direction **Light Emitting Diodes** are semiconductor devices that convert electrical energy into light energy. The construction of a light emitting diode is very different from that of a normal signal diode. The PN junction of an LED is surrounded by a transparent, hard plastic epoxy resin hemispherical shaped shell or body which protects the LED from both vibration and shock. Surprisingly, an LED junction does not actually emit that much light so the epoxy resin body is constructed in such a way that the photons of light emitted by the junction are reflected away from the surrounding substrate base to which the diode is attached and are focused upwards through the domed top of the LED, which itself acts like a lens concentrating the amount of light. This is why the emitted light appears to be brightest at the top of the LED. However, not all LEDs are made with a hemispherical shaped dome for their epoxy shell. Some indication LEDs have a rectangular or cylindrical shaped construction that has a flat surface on top or their body is shaped into a bar or arrow. Also, nearly all LEDs have their cathode, terminal identified by either a notch or flat spot on the body, or by one of the leads being shorter than the other, (the Anode, A).Unlike normal incandescent lamps and bulbs which generate large amounts of heat when illuminated, the light emitting diode produces a “cold” generation of light which leads to high efficiencies than the normal “light bulb” because most of the generated energy radiates away within the visible spectrum. Because LEDs are solid-state devices, they can be extremely small and durable and provide much longer lamp life than normal light sources.

2.6.1 Types of Light Emitting Diode

- Gallium Arsenide (GaAs) - infra-red
- Gallium Arsenide Phosphate (GaAsP) - red to infra-red, orange
- Aluminum Gallium Arsenide Phosphate (AlGaAsP) - high-brightness red, orange-red, orange, and yellow
- Gallium Phosphate (GaP) - red, yellow and green
- Aluminum Gallium Phosphate (AlGaP) - green
- Gallium Nitride (GaN) - green, emerald green
- Gallium Indium Nitride (GaInN) - near ultraviolet, bluish-green and blue
- Silicon Carbide (SiC) - blue as a substrate
- Zinc Solenoid (ZnSe) - blue
- Aluminum Gallium Nitride (AlGaN) - ultraviolet

Like conventional PN junction diodes, light emitting diodes are current-dependent devices with its forward voltage drop V_F, depending on the semiconductor compound (its light color) and on the forward biased LED current. The point where conduction begins and light is produced is about 1.2V for a standard red LED to about 3.6V for a blue LED. The exact voltage drop will of course depend on the manufacturer because of the different do pant materials and wavelengths used. The voltage drop across the LED at a particular current value, for example 20mA, will also depend on the initial conduction V_F point. As an LED is effectively a diode, its forward current to voltage characteristics curves can be plotted for each diode color as shown below

2.6.2 Light Emitting Diodes I-V Characteristics

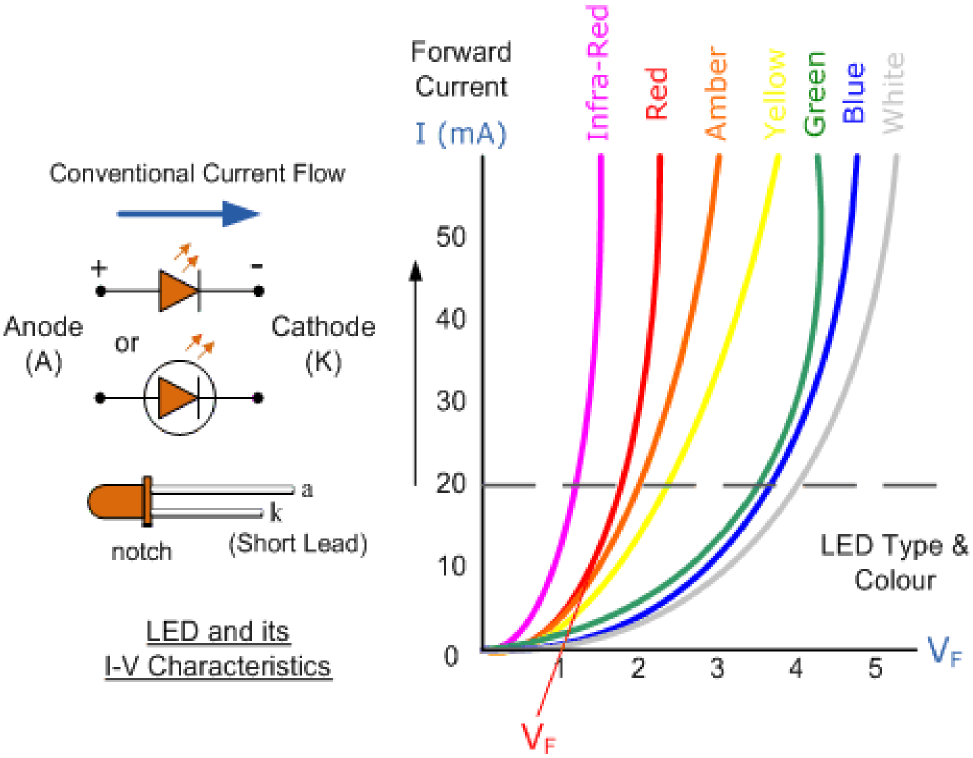


Fig: 2.7: Light Emitting Diode (LED) Schematic symbol and I-V Characteristics Curves showing the different colors available.

Before a light emitting diode can "emit" any form of light it needs a current to flow through it, as it is a current dependant device with their light output intensity being directly proportional to the forward current flowing through the LED.As the LED is to be connected in a forward bias condition across a power supply it should be *current limited* using a series resistor to protect it from excessive current flow. Never connect an LED directly to a battery or power supply as it will be destroyed almost instantly because too much current will pass through and burn it out. From the table above we can see that each LED has its own forward voltage drop across the PN junction and this parameter which is determined by the semiconductor material used, is the forward voltage drop for a specified amount of forward conduction current, typically for a forward current of 20mA. In most cases LEDs are operated from a low voltage DC supply, with a series resistor, R_S used to limit the forward current to a safe value from say 5mA for a simple LED indicator to 30mA or more where a high brightness light output is needed.

2.7 Sealed Lead Acid Battery (12V 7.2 Amp Hour)

Size 150(L) x 65(D) x 93(H) mm

Charge current 720mA for 10-14 hours

Discharge current 20 hr rate 350mA

Capacity:

20hr rate (0.35A) 7.0Ah

10hr rate (0.67A) 6.7Ah

5hr rate (1.19A) 5.95Ah

1hr rate (4.00A) 4.0Ah

Int Resistance:

Fully charged (20°C,68°F) 25m Ohm

Max Discharge Current (5 sec):40A

Constant Voltage Charge:

Initial charging current < 2.1A

Cycle:

Voltage 14.40V~15.00V @ 20°C (68°F)

Temp Coefficient: - 30mV/°C

Standby:

Voltage 13.5V~13.80V @ 20°C (68°F)

Temp Coefficient: - 20mV/°C

Chapter 3

System Design and Development

3.1 System Implementation

In this Chapter we are going to Explain about the system Design construction through Hardware and development of software. In addition, the chapter elaborates the hardware and the software step by step. All the operations of hardware and software are also included in this chapter. The system design of the total project is shown in below Figure3.1 with simple block diagram.

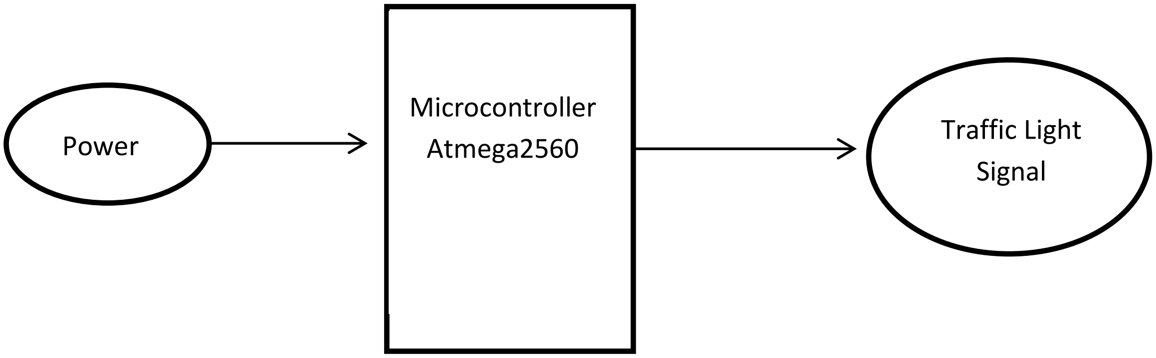


Fig: 3.1: Block Diagram of Simple System Design

3.2 System Architecture

(A) Input Power Appliance Module to operate Microcontroller

(B) Microcontroller Module

(C) Output Appliance Module

3.2.1 Input Power Appliance Module to Operate Microcontroller

To operate the microcontroller we need to supply fixed +5V to the microcontroller so that the microcontroller can give proper output for controlling traffic signal

3.2.1.1 Voltage Regulator

As microcontroller needs constant 5 volts to operate so we need a voltage regulator circuit. We are using IC 7805 along with two polar and two non-polar capacitors for this purpose. 7805 will take input from the supply (adopter) in our case and provide constant 5 volts throughout the circuit. The Vccthat microcontroller gets is coming from output of this IC.

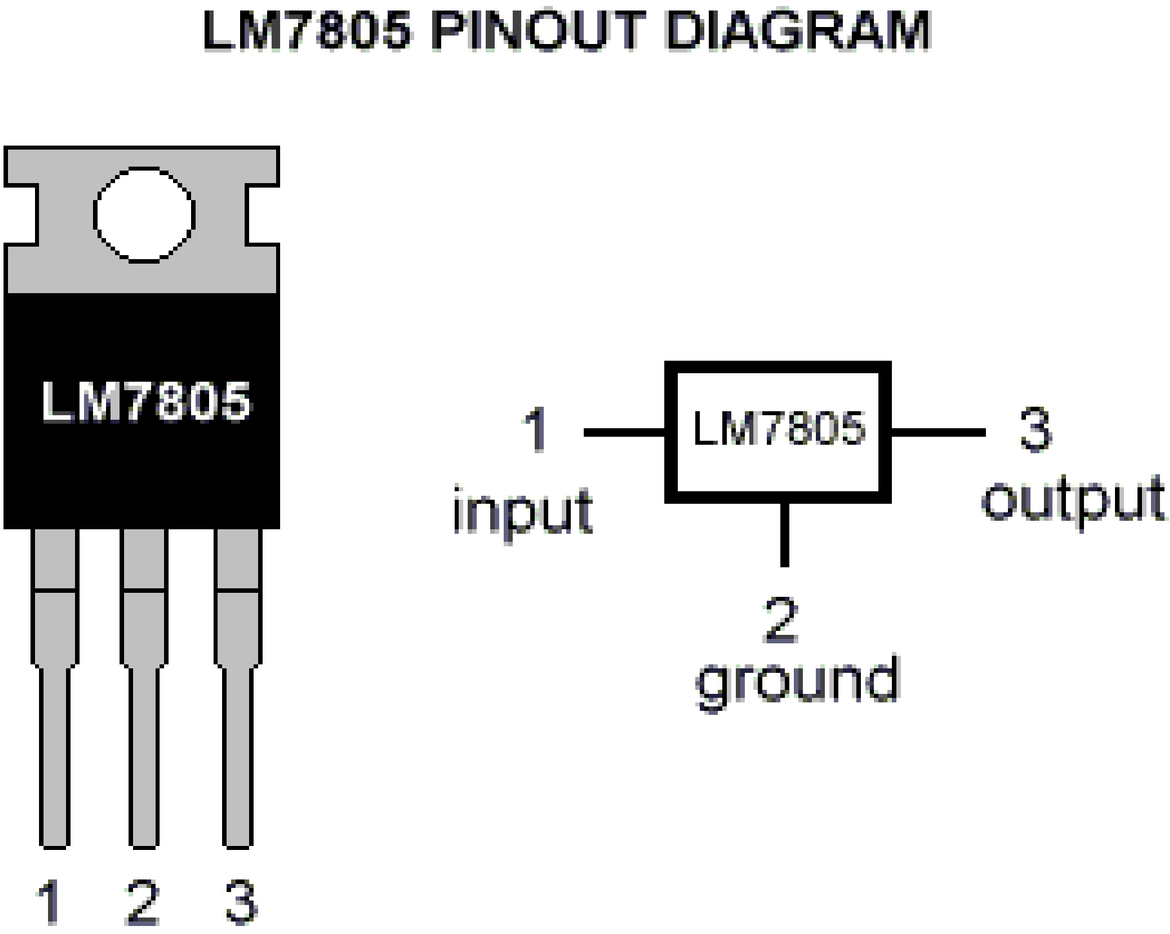


Fig 3.2: LM7805 Voltage Regulator

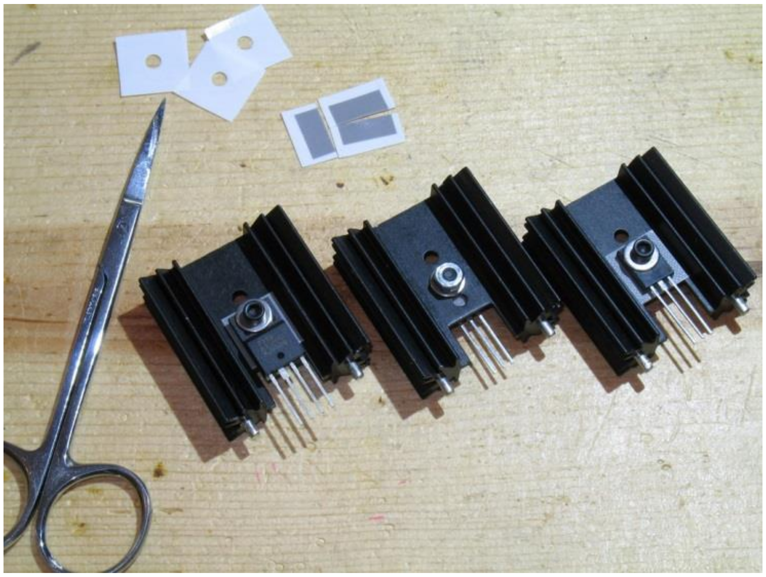


Fig 3.3: Heat sink mounting with LM7805

3.2.1.2 Power Supply

The Arduino 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. In our project we use 12 V 7.2 Ah Lead Acid Battery for power supply.

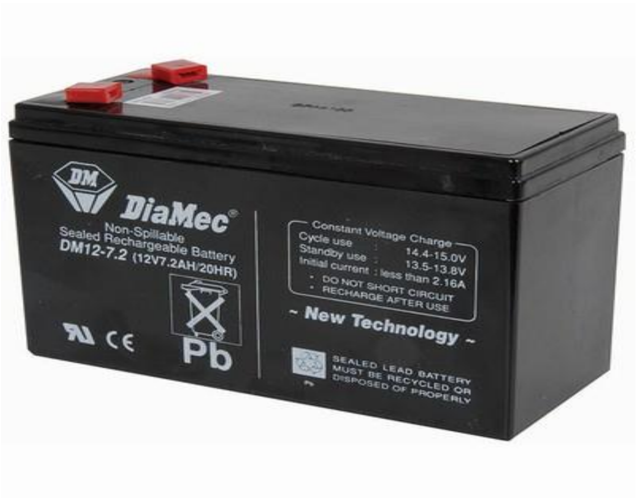


Fig 3.4: DC 12 V Lead Acid Battery

3.2.2 Microcontroller Module

Controller is the main part of the system where all the process flow will be controlled by this hardware accordingly to the embedded programming in it. Microcontroller is chosen for the system as the controller. In other word it is the heart of this device system. The functions of the microcontroller are limited by manufacturers or the types of certain model. Here we use handmade version of Arduino Mega2560 microcontroller.

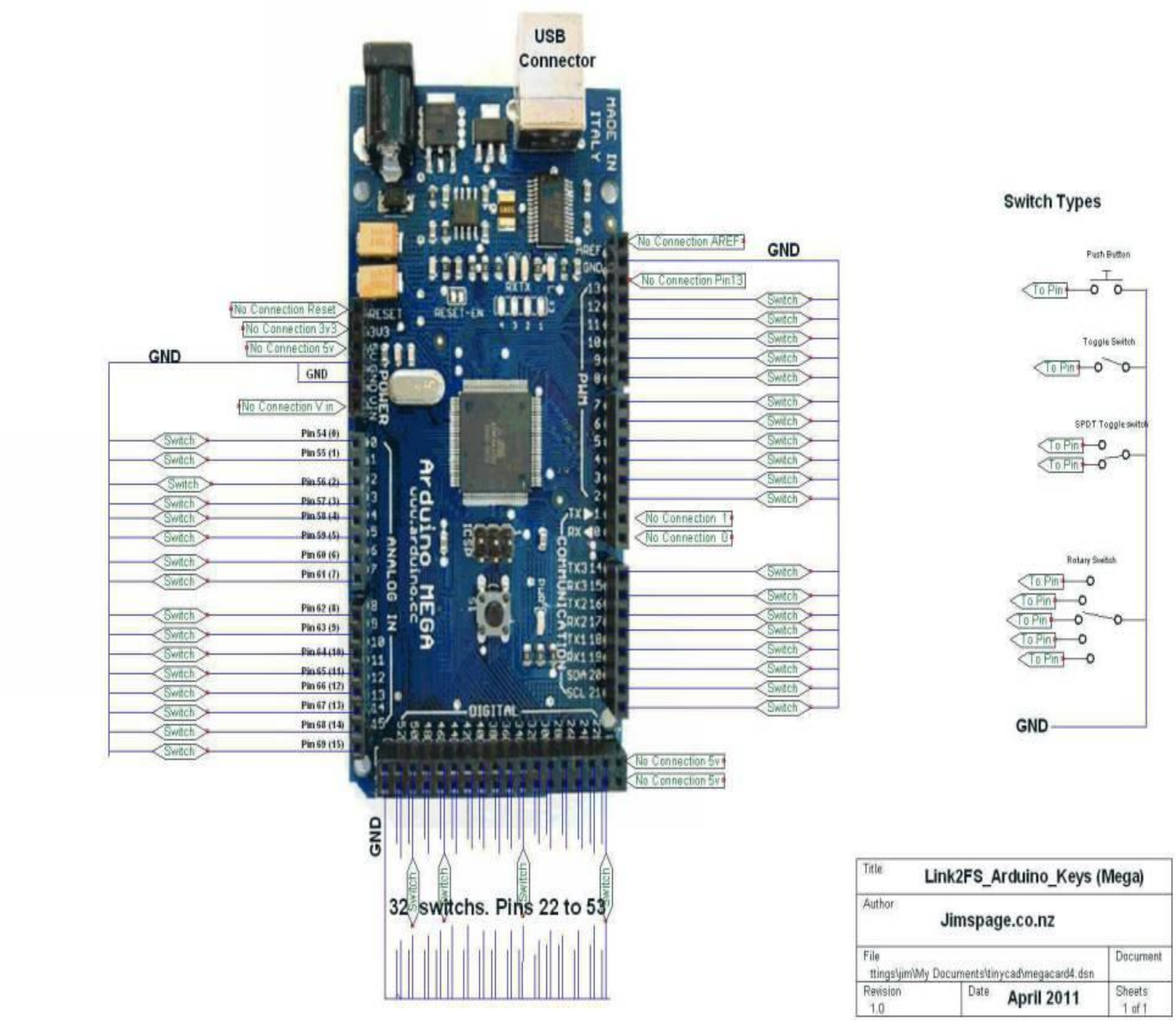


Fig 3.5: Pin diagram of Arduino Mega2560

3.2.2.1 Programming ATmega2560

To program ATMEGA2560 at first we have to burn boot loader of new ATMEGA328P using ARDUINO-mega2560 Programmer. For burn boot loader connect new ATMEGA2560 with programmer. Connection is shown below.

Caution: make sure Programmer is not connected to external power source because it get required power from USB cable connected to computer and circuit with ATMEGA2560 have regulated +12v supply.

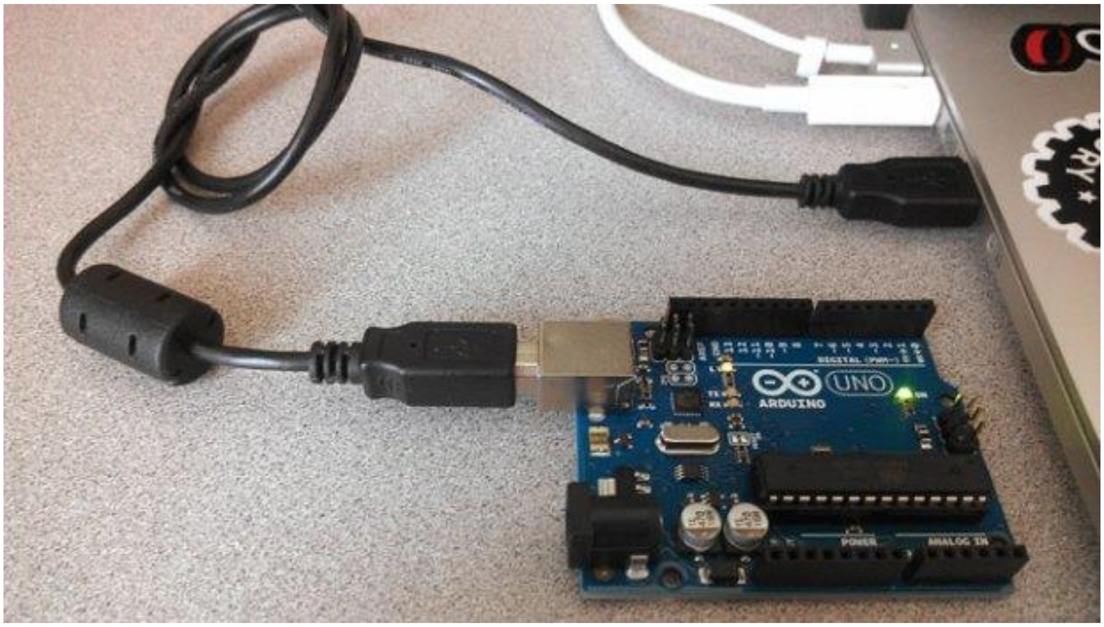
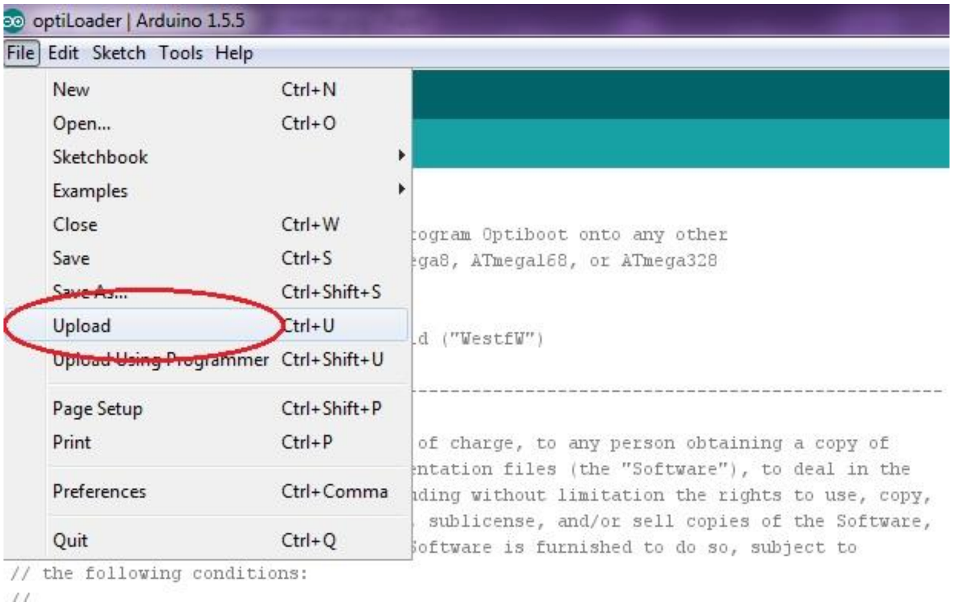


Fig 3.6: Interfacing ARDUINO Programmer with PC

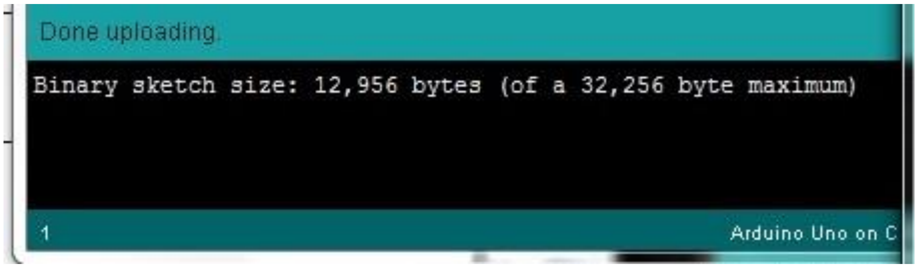
Then this System is ready for burn boot loader, to do this action simply open IDE on computer then upload configuration code which is consists of two different code named “optiLoader.h” and “optiLoader.pde”.Then upload both code to microcontroller, process is shown bellow



```
rogram Optiboot onto any other
ega8, ATmega168, or ATmega328

d ("WestfW")

---------------------------------------------------

 of charge, to any person obtaining a copy of
ntation files (the "Software"), to deal in the
ding without limitation the rights to use, copy,
 sublicense, and/or sell copies of the Software,
Software is furnished to do so, subject to
// the following conditions:
//
```

Fig 3.7: Upload Boot loader to ARDUINO Programmer

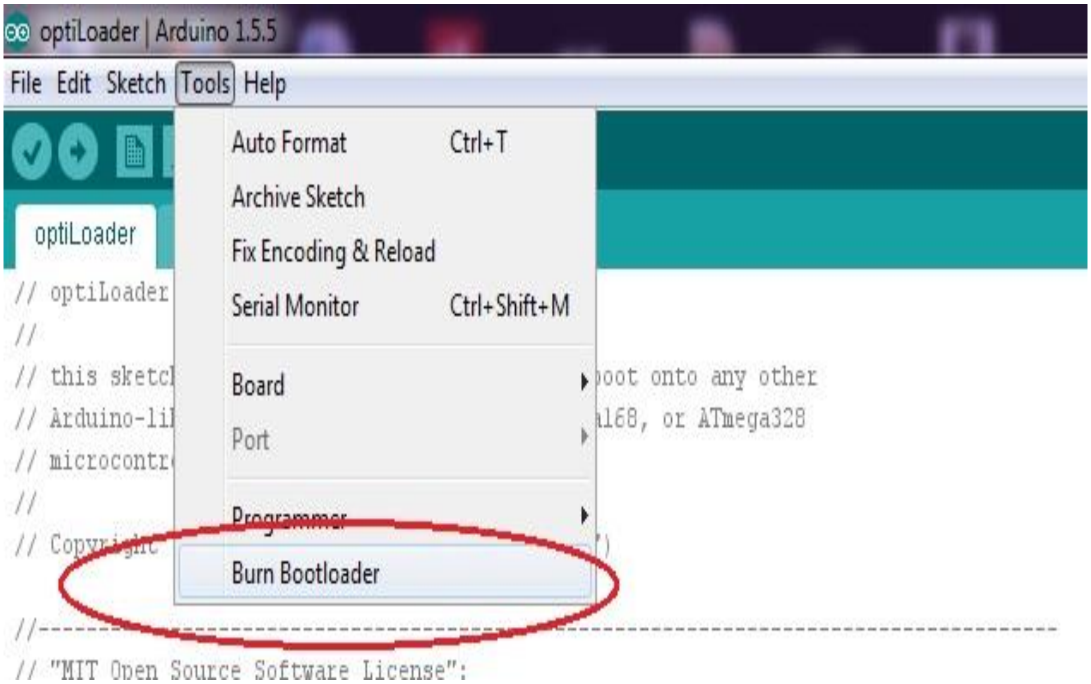
After successful upload a notification will be showed.


Done uploading.

```
Binary sketch size: 12,956 bytes (of a 32,256 byte maximum)
```

Fig 3.8: Confirmation Message for Upload Boot loader

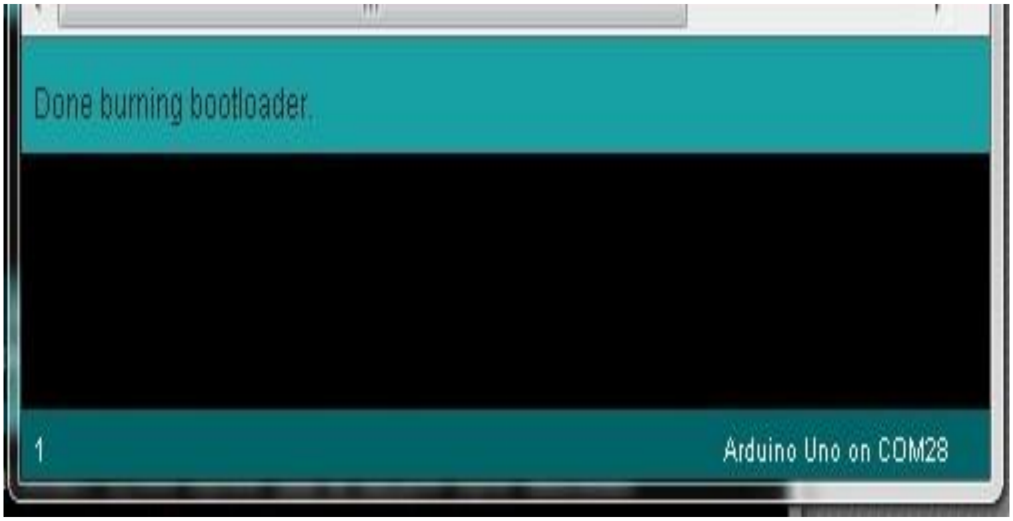
Then just burn it burning process is shown below:



```
// optiLoader
//
// this sketcl                                  boot onto any other
// Arduino-lil                                  a168, or ATmega328
// microcontr
//
// Copyright                                    ')

//------------------------------------------------------------------------------------
// "MIT Open Source Software License":
```

Fig 3.9: Burning Boot loader of ATmega2560

After burning a notification will be shown.



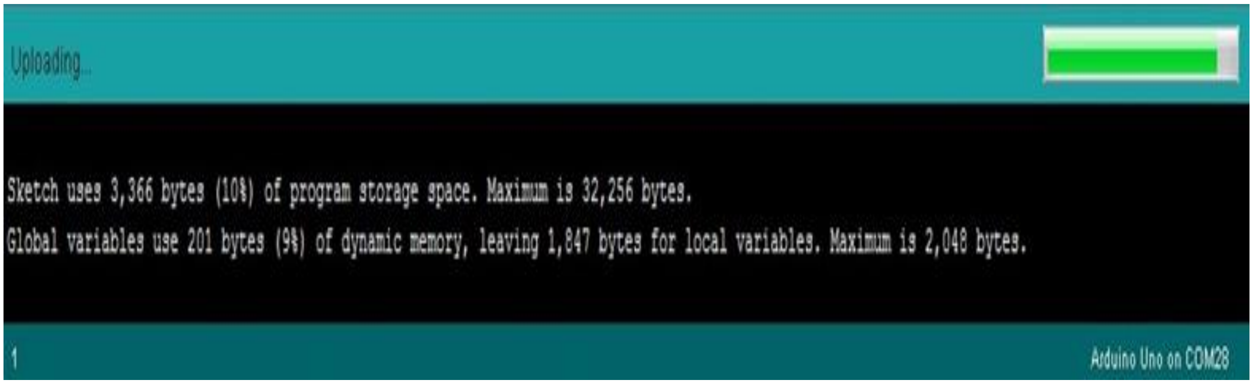
```
Done burning bootloader.
```

```
1                                        Arduino Uno on COM28
```

Fig 3.10: Confirmation message for burn Boot loader

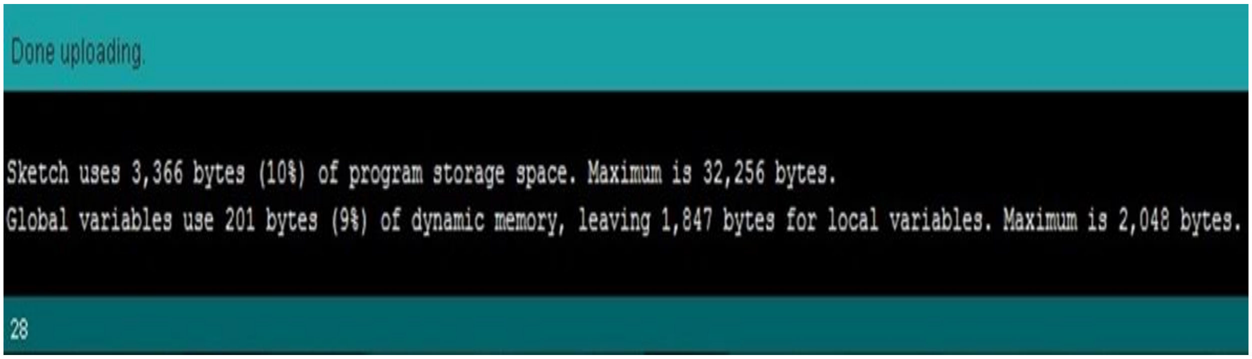
Then new ATMEGA328p is ready for programming.

At finally upload main program to new ATMEGA2560 using IDE



```
Sketch uses 3,366 bytes (10%) of program storage space. Maximum is 32,256 bytes.
Global variables use 201 bytes (9%) of dynamic memory, leaving 1,847 bytes for local variables. Maximum is 2,048 bytes.
```

Fig 3.11: Uploading PID Formula



```
Sketch uses 3,366 bytes (10%) of program storage space. Maximum is 32,256 bytes.
Global variables use 201 bytes (9%) of dynamic memory, leaving 1,847 bytes for local variables. Maximum is 2,048 bytes.
```

Fig 3.12: Message for upload complete

Finally our ATMEGA2560 is ready to use.

3.2.2.2 Oscillator

As microcontroller need a clock so for two capacitors and a crystal is used which will produce 16 MHz the two capacitors are connected to pin number 9 and 10 of microcontroller

Figure 3.13: Crystal Oscillator

3.2.3 Output Appliance Module

In this module we have taken 18 digital output pin of the microcontroller to connect the light signal at respective point. Each three output pins among the 18 output pins is connected to the three lights of red, green and yellow. Thus to implement our proposed project we used 18 output pins of microcontroller as shown in the following figure

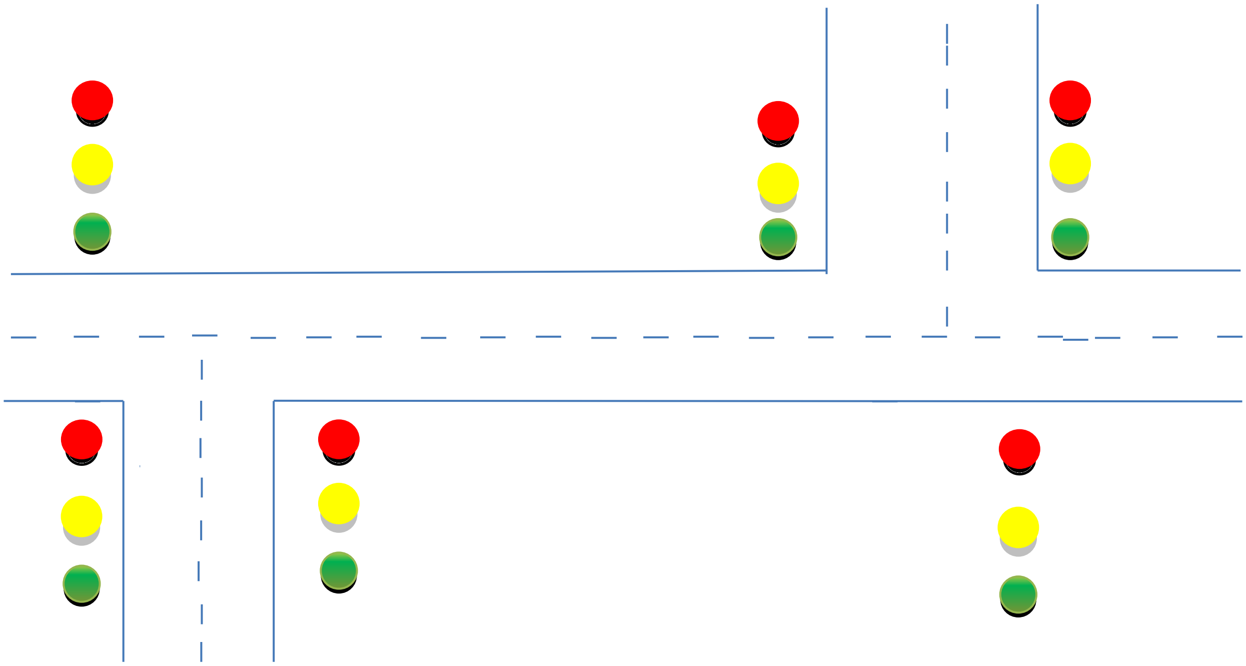


Fig: 3.14: Road Map Diagram

Chapter 4

Result and Simulation

The aim of this chapter is to show the result and simulation of this project for a two point of a three way road traffic signal system based on microcontroller which maintains the signal automatically as well as other important aspects of our proposed project. In our project we basically implemented time scheduling based traffic signal that is after a specific time period on the road side will be open and other will be closed and it will be repeated cyclically, so in that case we had several simulation model for the several traffic signal after a specific time period.

4.1 Simulation Model 1

In this simulation model we have the following situation where the

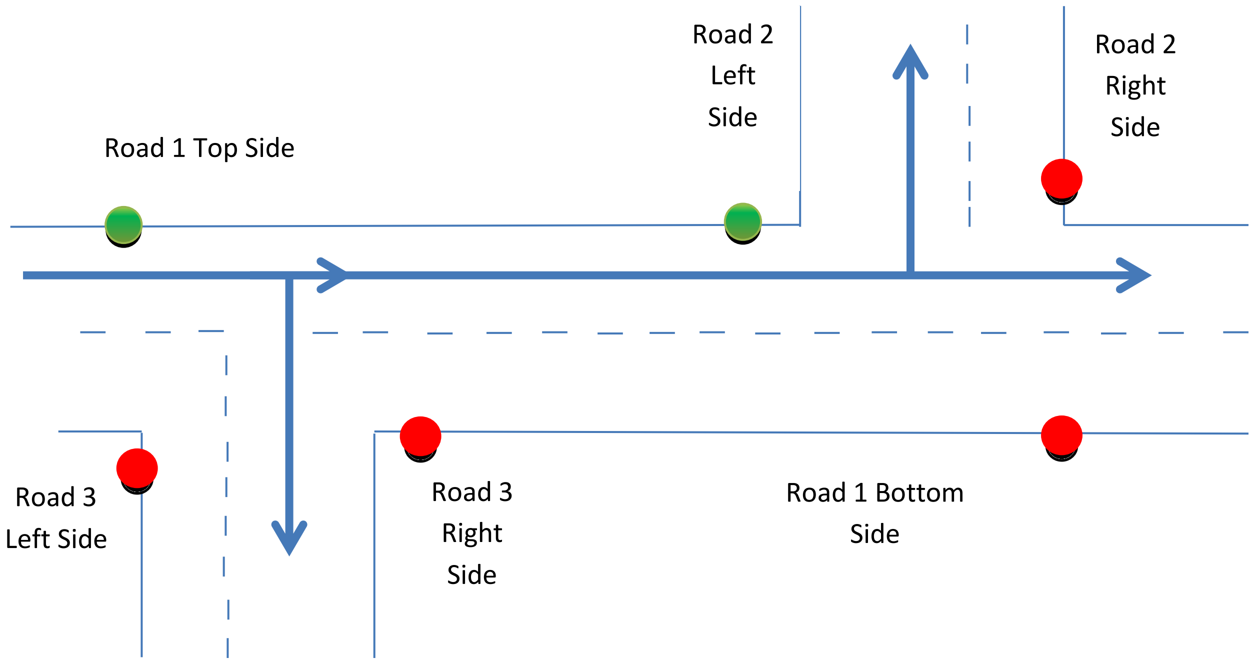


Fig 4.1 Simulation Model 1

Road 1 Top side, Road 2 of Left side and Road 3 Right side remain open and all the other side remains closed as long as the green signal of the Road 1 Top side remain in ON state. The cars are allowed to go through the Road 1 Top side, Road 2 of Left side and Road 3 Right side as shown in the above figure. In our proposed project the roads will be open for 30 seconds.

4.2 Simulation Model 2

In this simulation model we have the following situation where the

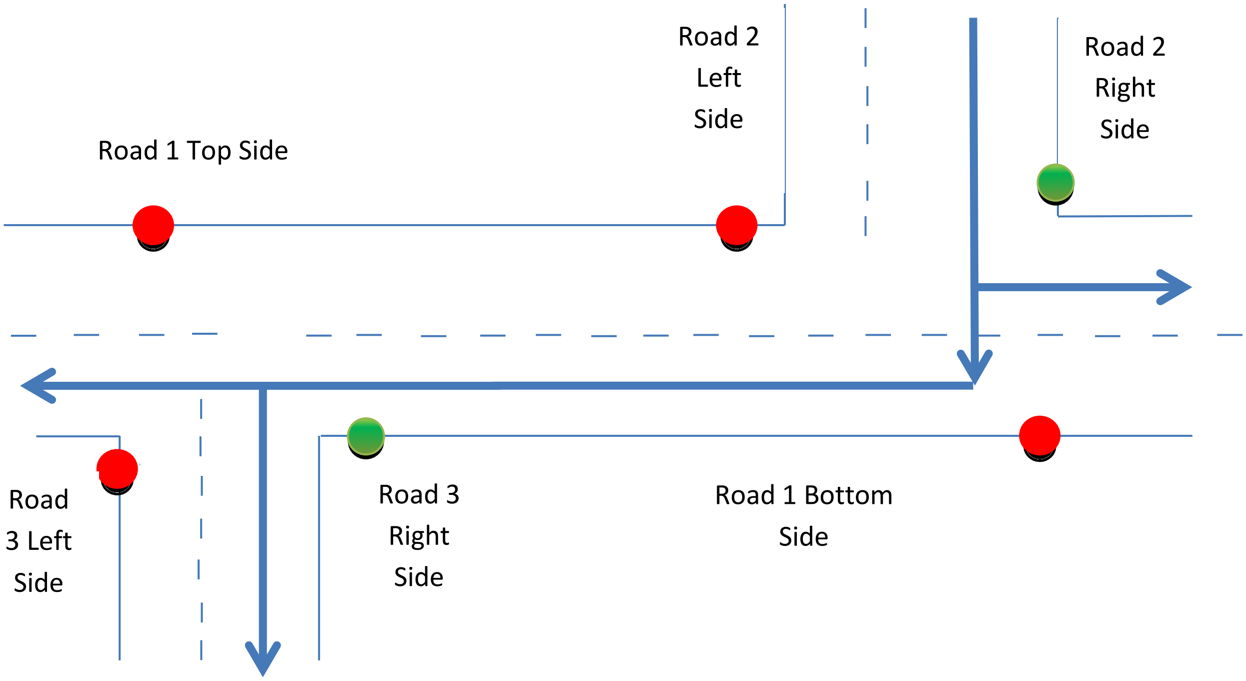


Fig 4.2 Simulation Model 2

Road 2 Right side, Road 1 Bottom side and Road 3 Right side remain open and all the other side remains closed as long as the green signal of the Road 2 Right side remain in ON state. The vehicles are allowed to come from Road 2 Right side as shown in the above figure. It is noted here that Road 1 Bottom side will be kept open only for the vehicles of Road 2 Right side not for the vehicles of Road 1 Bottom side individually.

4.3 Simulation Model 3

In this simulation model we have the following situation where the

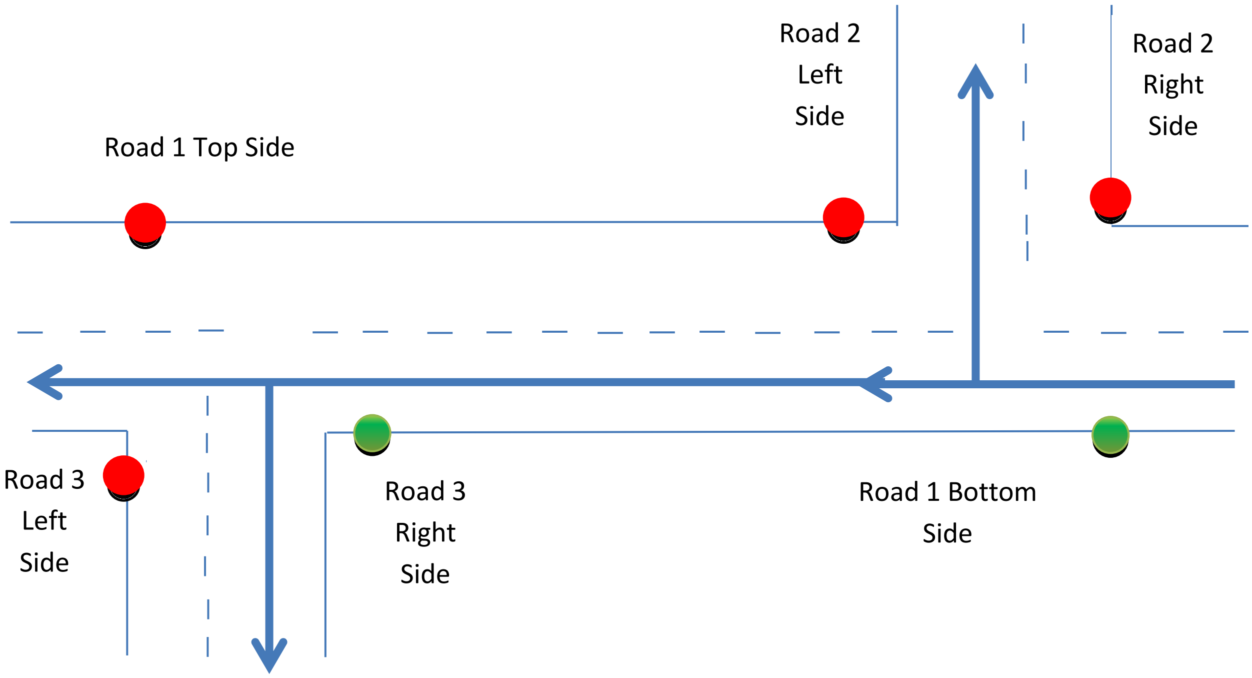


Fig 4.3 Simulation Model 3

Road 1 Bottom side, Road 2 Left side only for Road1 Bottom side vehicles and Road 3 Right side remain open and all the other Road sides remain closed as long as the green signal of the Road 1 Bottom side remain in ON state. The vehicles are allowed to go through Road 2 Left side, Road 1 Bottom side and Road 3 Right as shown in the above figure.

4.4 Simulation Model 4

In this simulation model we have the following situation where the

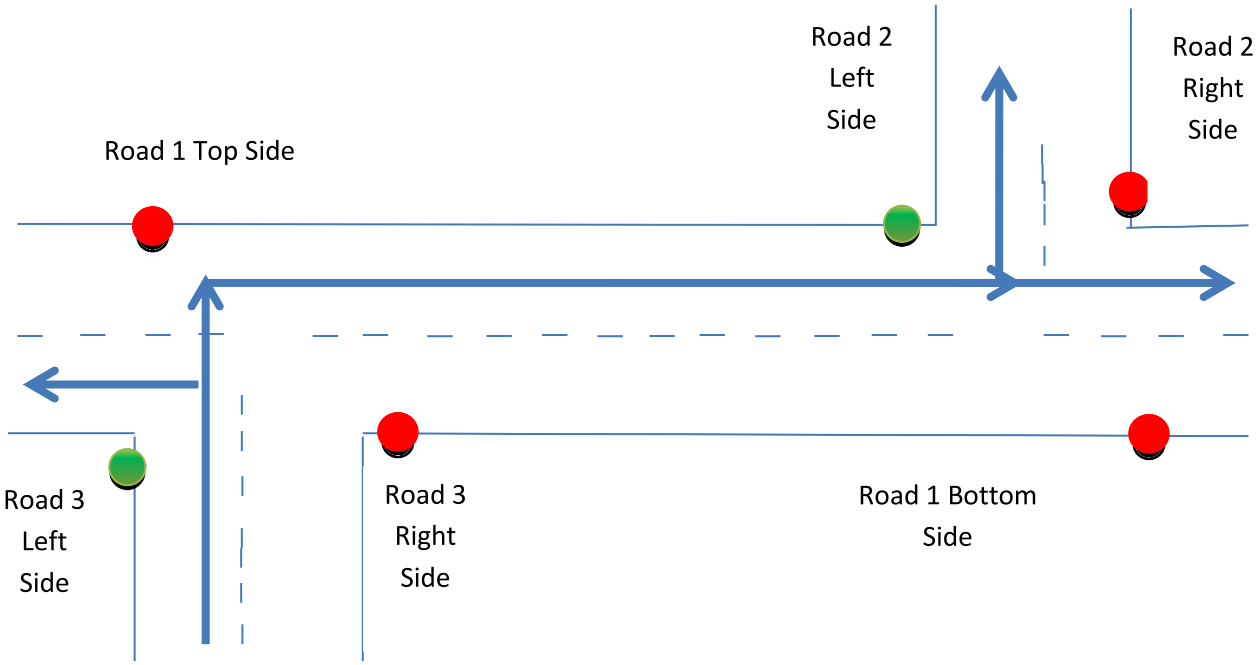


Fig 4.4 Simulation Model 4

Road 3 Left side, Road 1 Top side only for Road 3 Left side vehicles and Road 2 Left side remain open and all the other Road sides remain closed as long as the green signal of the Road 3 Left side remain in ON state. The vehicles are allowed to go through Road 2 Left side, Road 1 Bottom side and Road 3 Right as shown in the above figure.

4.5 Project over view

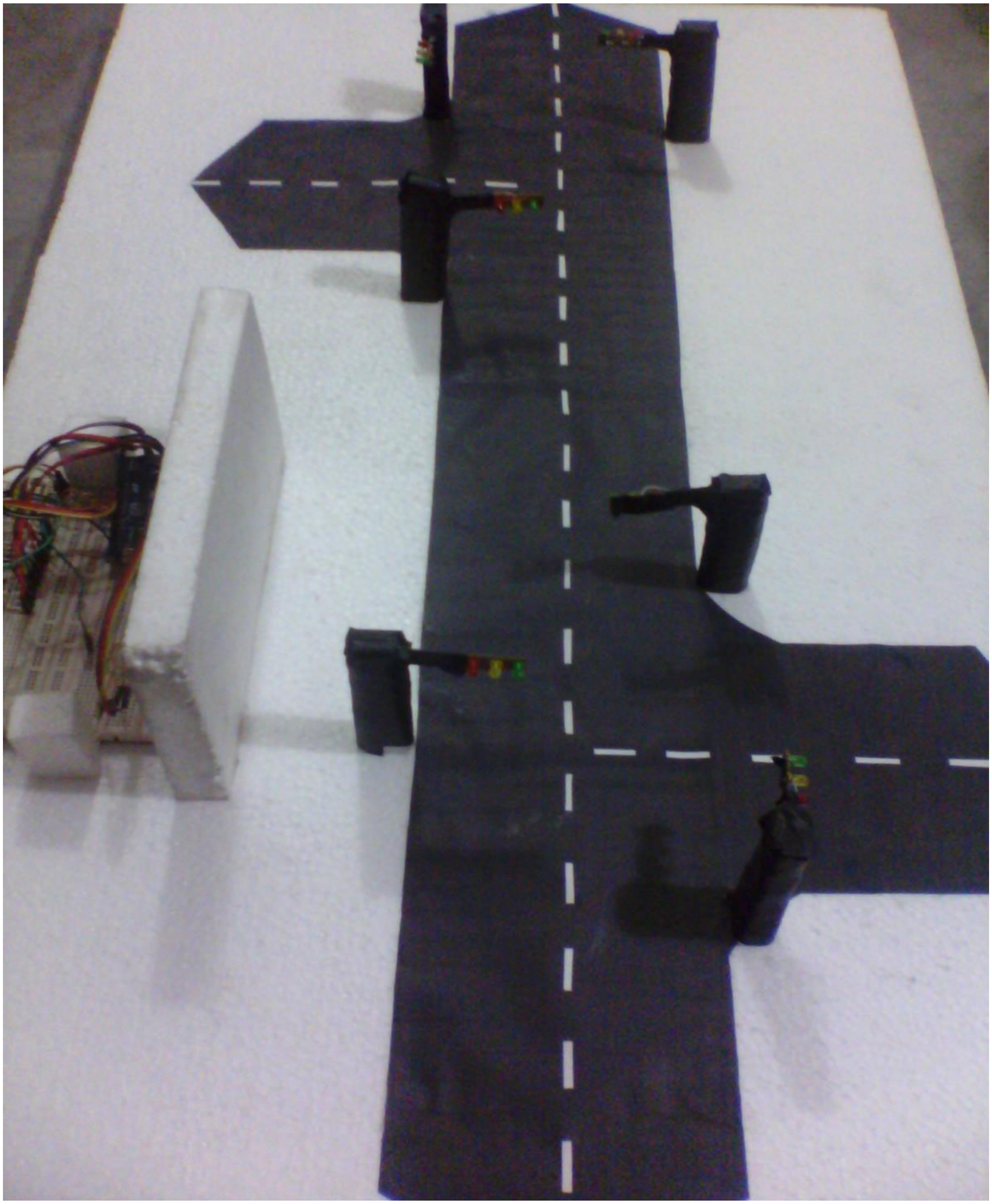


Fig: 4.5 Project Screenshot

4.6 Costing

Costing is always a vital issue to make any project. Price of electronics is not stable for a developing country like Bangladesh, because Bangladesh never produces electronics parts but import from other developed country and during import price depends upon the stock of foreign currency. Average price of parts used in this project is given bellow:

Table 2: Price list of the Hardware used

Sl. No.	Name	Quantity	Price per Quantity	Price(BDT)
1	Arduino mega 2560	1	2480	2480
2	Resistor 45Ω	20	2	40
3	Switch	2	15	30
4	Connecting Rail(Male&Femal)	20	5	100
5	Connecting Wire	25 (ft)	5	125
6	Bread Board	2	240	480
7	12V Lead Acid Battery	1	150	150
8	Voltage Regulator(7805)	1	10	10
9	Others		100	100
			Total Cost = 3515	

Chapter 5

Conclusion & Future Scope

5.1 Conclusion

In this work, an attempt has been done to design of a traffic control system using Micro controller for efficient use of traffic control system. It will help to reduce the wastage of time, save lives, reduce percentage of accident. The program we embedded in the micro controller works according to our wish. According to the study and analysis of various parts of the system, a design has been carried out. The results obtained from the measurement have shown that the system perform well under all the conditions. It is very essential to solve our traffic management problem. Because each time of our life and every movement we move into tie traffic signal and sometime it is a big issue for losing our value able time. Our neatest many country follow many kinds of automation system where as we have not any existing automated system. Now this issue common phenomena and it is also a national problem of our country.

5.2 Future Scope

The Performance of microcontroller based efficient use of electricity in Office appliance system has been found on expected lines. However, there exists a scope for further improvement in its speed, power consumption, and PC interface software for post data analysis. Because there is say that "**Tomorrow is more advanced than Today**".

We proposed a traffic model we used into some cc camera which collects snap continuously and send it to the police box controller machine. Controller machine process that images control traffic signal and also send information to the central database. Those are store into the central database. These databases share this information to connect with others traffic signal. It is also send its information to the central database through the network. So that, all the traffic signals know the real time situation of each the traffic point. It will also included a vehicle tracking system which is trace the vehicle recognize the number plate.

Appendix 01

C-language code of the system

```
constint MRS_1_P1_GREEN =6;
constint MRS_1_P1_YELLOW =7;
constint MRS_1_P1_RED =8;
constint MRS_1_P2_GREEN =9;
constint MRS_1_P2_YELLOW =11;
constint MRS_1_P2_RED =13;
constint R_27S_2_GREEN =22;
constint R_27S_2_YELLOW =23;
constint R_27S_2_RED =24;
constint MRS_2_P2_GREEN =25;
constint MRS_2_P2_YELLOW =26;
constint MRS_2_P2_RED =27;
constint MRS_2_P1_GREEN =27;
constint MRS_2_P1_YELLOW =28;
constint MRS_2_P1_RED =29;
constint PRS_2_GEREEN =30;
constint PRS_2_YELLOW =31;
constint PRS_2_RED =32;


void setup() {

  // put your setup code here, to run once:

pinMode(MRS_1_P1_GREEN,OUTPUT);
pinMode(MRS_1_P1_YELLOW,OUTPUT);
pinMode(MRS_1_P1_RED,OUTPUT);
pinMode(MRS_1_P2_GREEN,OUTPUT);
pinMode(MRS_1_P2_YELLOW,OUTPUT);
pinMode(MRS_1_P2_RED ,OUTPUT);
pinMode(R_27S_2_GREEN,OUTPUT);
pinMode(R_27S_2_YELLOW,OUTPUT);
pinMode(R_27S_2_RED,OUTPUT);
pinMode(MRS_2_P2_GREEN,OUTPUT);
pinMode(MRS_2_P2_YELLOW,OUTPUT);
pinMode(MRS_2_P2_RED,OUTPUT);
pinMode(MRS_2_P1_GREEN,OUTPUT);
pinMode(MRS_2_P1_YELLOW,OUTPUT);
pinMode(MRS_2_P1_RED,OUTPUT);
pinMode(PRS_2_GEREEN,OUTPUT);
pinMode(PRS_2_YELLOW,OUTPUT);
pinMode(PRS_2_RED,OUTPUT);

//initially the outputs are low

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,LOW);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,LOW);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,LOW);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,LOW);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,LOW);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,LOW);

}

void loop() {

  // put your main code here, to run repeatedly:

  show_MRS_1_P1P2_GREEN_ON();
delay(30000);
  show_MRS_1_P1P2_YELLOW_ON();
delay(3000);
  show_R_27S_2_MRS_2_P1_GREEN_ON();
delay(30000);
  show_R_27_S2_YELLOW_MRS_2_P1_GREEN_ON();
delay(3000);
  show_MRS_2_P2P1_GREEN_ON();
delay(30000);
  show_MRS_2_P2P1_YELLOW_ON();
delay(3000);
  show_PPS_2_GREEN_MRS_1_P2_GREEN_ON();
delay(30000);
```

```
  show_PPS_2_YELLOW_MRS_1_P2_GREEN_ON();
delay(3000);

}

//functions

void show_MRS_1_P1P2_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,HIGH);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,LOW);
digitalWrite(MRS_1_P2_GREEN,HIGH);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,LOW);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,HIGH);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show MRS_1_P1P2_YELLOW on

void show_MRS_1_P1P2_YELLOW_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,HIGH);
digitalWrite(MRS_1_P1_RED,LOW);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,HIGH);
digitalWrite(MRS_1_P2_RED ,LOW);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,HIGH);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show R_27_S2 on

void show_R_27S_2_MRS_2_P1_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,HIGH);
digitalWrite(R_27S_2_GREEN,HIGH);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,LOW);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,HIGH);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,LOW);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show R_27_S2_YELLOW_MRS_2_YELLOW on

void show_R_27_S2_YELLOW_MRS_2_P1_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,HIGH);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,HIGH);
digitalWrite(R_27S_2_RED,LOW);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,HIGH);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,LOW);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show MRS_2_P2P1_GREEN_ on

void show_MRS_2_P2P1_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,HIGH);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,HIGH);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,LOW);
digitalWrite(MRS_2_P1_GREEN,HIGH);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,LOW);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show MRS_2_P2P1_YELLOW on

void show_MRS_2_P2P1_YELLOW_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,LOW);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,HIGH);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,HIGH);
digitalWrite(MRS_2_P2_RED,LOW);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,HIGH);
digitalWrite(MRS_2_P1_RED,LOW);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,HIGH);

}

//show PPS_2_GREEN_MRS_1_P2_GREEN on

void show_PPS_2_GREEN_MRS_1_P2_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,HIGH);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,LOW);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,HIGH);
digitalWrite(PRS_2_GEREEN,HIGH);
digitalWrite(PRS_2_YELLOW,LOW);
digitalWrite(PRS_2_RED,LOW);

}

//show PPS_2_GREEN_MRS_1_P2_GREEN on

void show_PPS_2_YELLOW_MRS_1_P2_GREEN_ON(){

digitalWrite(MRS_1_P1_GREEN,LOW);
digitalWrite(MRS_1_P1_YELLOW,LOW);
digitalWrite(MRS_1_P1_RED,HIGH);
digitalWrite(MRS_1_P2_GREEN,HIGH);
digitalWrite(MRS_1_P2_YELLOW,LOW);
digitalWrite(MRS_1_P2_RED ,LOW);
digitalWrite(R_27S_2_GREEN,LOW);
digitalWrite(R_27S_2_YELLOW,LOW);
digitalWrite(R_27S_2_RED,HIGH);
digitalWrite(MRS_2_P2_GREEN,LOW);
digitalWrite(MRS_2_P2_YELLOW,LOW);
digitalWrite(MRS_2_P2_RED,HIGH);
digitalWrite(MRS_2_P1_GREEN,LOW);
digitalWrite(MRS_2_P1_YELLOW,LOW);
digitalWrite(MRS_2_P1_RED,HIGH);
digitalWrite(PRS_2_GEREEN,LOW);
digitalWrite(PRS_2_YELLOW,HIGH);
digitalWrite(PRS_2_RED,LOW);

}
```

References

1. Arduino. Introduction to Arduino [Online]. Italy:
   Arduino URL: http://arduino.cc/en/Guide/Introduction
   Accessed July 15 2013

2 Arduino. Arduino board mga[Online]. Italy:
Arduino URL: http://arduino.cc/en/Main/arduinoBoardMega2560
Accessed 17 July 2013

3 Arduino 1.0.5
Arduino URL: http://arduino.cc/en/main/software#toc2
Accessed 17 July 2013

4 Microcontrollers: Theory and Applications by Ajay V Deshmukh
URL: http://books.google.com.bd/books?id=5PDx2Q9Ea_YC&hl=en&authuser
Accessed 11 July 2013

5 Tech Shop Bd
URL: http://www.techshopbd.com
Accessed 11 July 2013

6 LED:
http://www.electronics-tutorials.ws/diode/diode_8.html

7 *LM7805* Datasheet
URL: *www.fairchildsemi.com/ds/LM/LM7805.pdf*
Accessed 11 July 2013

8 http://www.circuitstune.com/2012/09/7805-voltage-regulator-circuit-7805-pinout.html