Design & Developing of a Microcontroller Based Intelligent Traffic Control System

Prepared By

Salim Bin Islam

ID: 2012-1-55-058

&

Mahmudur Rahman

ID: 2012-1-55-002

Project Supervisor:

Sarwar Jahan

Senior Lecturer
Dept. of Electronics & Communication Engineering
East West University

East West University

Department of Electronics & Communication Engineering

DECLARATION

We hereby declare that this Project is our original work. We also declare that no part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged. Requisite references are quoted to support my work.

Signature:

Salim Bin Islam
ID: 2012-1-55-058

Mahmudur Rahaman
ID: 2012-1-55-002

Signature of Supervisor	**Signature of Dept. Chairperson**
--- **Sarwar Jahan** Senior Lecturer Dept. Physics Electronics & Communication Engineering East West University	------------------------------------ **Dr. Md. Mofazzal Hussain** Professor & Chairperson Dept. of Electronics &Communication Engineering East West University

APPROVAL

The Project titled as “Design and developing of a microcontroller based Intelligent Traffic Control System” has been submitted to the following 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.

Salim Bin Islam
ID: 2012-1-55-058

Mahmudur Rahman
ID: 2012-1-55-002

Supervisor

Sarwar Jahan

Senior Lecturer

Dept. of Electronics & Communication Engineering

East West University

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

ACKNOWLEDGEMENTS

Many people deserve our thanks for their help in completing this project. We would like to thank our department for giving us this chance to do this project. We want to express our thanks and deep appreciation to our advisor Sarwar Jahan as he has exhausted all his knowledge and time by following up our daily progress and encouraging advices and even by sharing on the troubles.

We would like to extend our thanks to the laboratory staff of ECE Dept. for their fast response and cooperation with us to get some materials we need for our case.

We have special acknowledgment for our group members for their understanding each other and hard working from the beginning up to the end.

Finally, we would like to thank the entire person who involve with this projects for their invaluable help and professionalism during the project work.

ABSTRACT

The number of vehicles is gradually increasing day by day all over the world. Vehicles are also increasing in the metropolitan cities of Bangladesh. As a result huge traffic congestions and traffic jam is increasing day by day. For traffic congestions a huge amount of time is being wasted by the citizen of metropolitan area. In this project an intelligent traffic control system is developed by us to minimize the congestions of traffic system. Here the traffic signal will be automatically control by the system. The road will be open for moving forward or close for moving forward is controlled by the system in real time and as necessary. When a certain number of vehicle is stuck in the traffic signal then the other part of the path will be show red signal and the congested path's signal will be Green. After a few time when the first path will reach a certain number of vehicle then the second path's signal will turn into red and the first path's signal will turn into green. Our intelligent traffic control system can detect temperature of the environment also. And the information about signal and temperature will send to the certain vehicle holder by Short Message Service (SMS).

Chapter 1

INTRODUCTION

1.1 Traffic Congestion

Congestions in traffic system occur when there are more vehicles than the road can handle. This situation makes the trip longer than it should be as it increases the queuing of vehicles. This phenomenon is also known as traffic jam. Congestions in traffic can be a result of accident, going through wrong way, for VIP passing, unauthorized parking etc. it can be happened due to bad road layout, misunderstanding traffic rules also.

1.2 Impacts of traffic congestion

Traffic congestion has a number of negative effects.

- Wasting of valuable time.
- Delay, that may result in late arrival for employment, office, school.
- Inability to guess accurate travelling time.
- Increased wasting of fuel and air pollution also.
- Congestion of traffic may block the passage of emergency vehicles travelling.
- Higher chance of collisions due to tight spacing and constant stopping and going.

1.3 Purpose of the project

Various studies and surveys have estimated that, on a daily basis most of the traffic congestion or traffic jams occur because of traffic light system. Traffic congestion is nothing but an additional waste of time from one’s daily routine. It is noticed that most of the traffic congestion is occur during the morning and late afternoon. Basically during that time the students and employers go to school, college, University or office so they also be late for their office or institutions at the traffic light spot. As the number of road users constantly increasing with respect to time, and resources provided by current infrastructures are limited, intelligent control of traffic will become a very important issue in the future. Avoiding unwanted traffic jam would be beneficial to both environment and economy.

Our smart traffic light and congestion control system designed to have a solution of traffic congestion problem of metropolitan cities of the country that will results in minimizing the economic cost and save time also. By decreasing the congestion of cities we can also decrease the extra waste of energy like CNG, Petroleum for special case electricity also. As a densely populated developing country like Bangladesh whose GDP is on an average 6% and whose most of the foreign currency from reserves are spent on importing petroleum and electricity, can't afford to waste such an important resource.

1.4 Our "Intelligent Traffic Control System"

We propose a new intelligent traffic control system that is able to control the traffic system through traffic signals on the basis of current traffic density. The current traffic control system of Bangladesh is controlled by the traffic police manually. Some roads of Dhaka city have a automatic system for controlling the traffic light. It has fixed time interval for passing the traffic from either sides of the road that causes congestion problem. Our intelligent traffic control system will solve this problem on real time basis. This system will observe the current traffic condition then it will take decision that which road will remain open and which road will be remain close.

1.5 Aim of the Project

- To design a smart traffic control system
- To construct a smart traffic controller circuit in the breadboard
- Test for its functionality
- Product commercialization
- To design the control system with low cost components

Chapter 2

AN OVERVIEW OF MICROCONTROLLERS

2.1 Arduino

Arduino is the most popular programmable board that help to complete any communications and electronics based projects. Basically Arduino is an open source platform which has “one click compile or upload” feature. It is an open source electronics prototyping platform that easy to use and flexible both for the software and hardware. Arduino is able to observe its surroundings through the light, controlling motor and other activators. It is very easy to write and upload codes to the I/O board to run any programs due to its open source platform features.

2.2 History

In 2005, Arduino was released by students from the Interaction Design Institute Ivrea(IDII). They developed it for their class project as a modest tool for windows and Mac OSX. From the beginning time, the attractive features of Arduino was developed to attract artist and designers. The open source microcontroller hardware has been designed in a way that it can easily interface of the external components like motors, speakers and LED (reply to the user inputs). Hernando Barragan, who was created the wiring microcontroller that parsing data to electronics devices to contribute Arduino project. Mainly, he wanted it to be used as a prototyping tool. Later, his assistance (Hernando Barragan) created the visual programming language for the prototyping tool.

2.3 Development

Whereas discussing the development of Arduino, it is worth introducing about microcontroller. In the 1960, a revolutionary development was seen in the computer industry after the development of solid state computers (like as IBM 1401), that used transistors to process its operation and its magnetic core memory for its storage (on behalf of vacuum tubes) that enable to increase the solidity of the computer. In 1959, Jack kilby invented integrated circuits, which enable circuits and transistors to be focused into tiny chips of semiconducting materials (like as silicon), which allowed further miniaturization of the compute component. The other most important in the same decade was the high level computer programming languages, written in symbolic language which made computer codes somewhat easy to learn and read than the previous machine languages. FORTRAN (uses in scientific calculator) and COBOL (uses in business application) were the main languages introduced in that time. The microprocessor was one of the greatest innovations in the history of the modern computer in the 1970's. In the beginning, the microprocessor miniaturized all the hardware components of CPU to fit into one, tiny, integrated circuit known as the microchip. The microchip became the main driving component of the microcontrollers (like Arduino), which consists of a microchip, memory storage hardware and input/output hardware for sensors. After that in 1970's and 1980's more languages were also developed (such as C, C++, Java) for applications in science and business.

2.4 Evolution

Before the invention of Arduino, the PIC microcontroller board was introduced by general instruments in 1985 was one of the most used tools for the electronic enthusiasts. The PIC microcontroller board was preferred because of it's the speed and ease of its programming through simple languages including PBASIC. Another reason was that it was able to store programs on a flash memory chip which enable the instructions on the board to be programmed with an infinite number of possibilities. It also support output device like LED and motors as well as sensors. Other popular boards for hobbyists include the BASIC Stamp (Parallax Inc. 1990), and wiring both of which share the benefits of simplicity of programming, and a resulting ease of rapid-prototyping. In fact, the immediate precursor to the Arduino was a custom made wiring microcontroller created by the artist/ designer Hernando Barragan in 2004.

It was in 2005 when the Arduino team was formed in Italy and it consisted of Barragan Massimo, David Cuartilles, Nicholas Zambetti and Dave Mellis. Main goal of this team was to develop an electronic prototyping platform that would simplify the wiring platform and make it accessible to the non-technical users especially in the creative fields. The Arduino, therefore incorporated the following characteristic, a programming environment based on processing language (a programming language conceived by Ben Fry and Casey Reas, also conceived for artists/ designers), the ability to program the board via a standard USB connection and a low price point.

Chapter 3

THEORY BEHIND THE PROJECT

3.1 Introduction

This project is included with Arduino Uno broad, (Light Dependent Resistor), Shift resistor IC 74164, LM35, LED. We want to develop a dark sensor and connect it to an Arduino. And that will be used to provide a pulse.

3.2 Architecture Arduino Uno board

There are various types of Arduino boards that can be used for different purposes. The Arduino UNO components are:

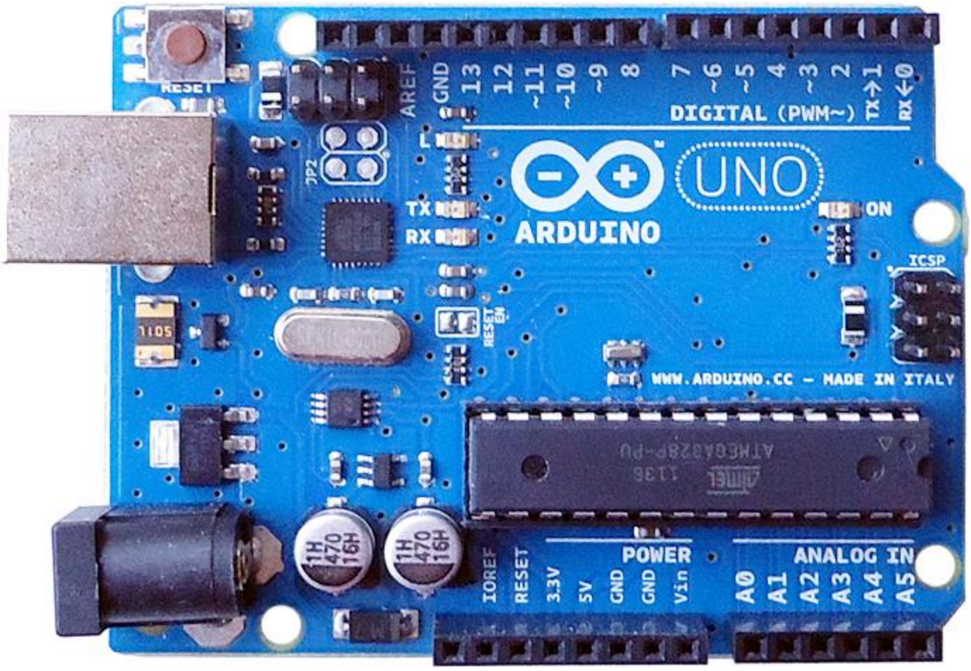

Figure 3.1: Arduino UNO R3 Board

3.2.1 Power -USB / Barrel Jack

Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled and the barrel jack is labeled. The USB connection is also how you will load code onto your Arduino board.

3.2.2 Pins (Analog, Digital, PWM, AREF, 5V, 3.3V, GND)

Arduino pins are the places where connect wires to construct a circuit. The Arduino has several types of pins, each of which is labeled on the board and used with different functions.

- ❖ **Analog**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) is Analog In pins. These pins can read the signal from an analog sensor and convert it into a digital value that we can read.
- ❖ **Digital**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input and digital output (like powering an LED).
- ❖ **PWM**: The digital pins (3, 5, 6, 9, 10, and 11) on the UNO are the PWM (~) pins. These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).
- ❖ **AREF**: Stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.
- ❖ **5V**: The 5V pin supplies 5 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- ❖ **GND**: Full name is Ground. There are several GND pins on the Arduino, any of which can be used to ground circuit.

3.2.3 Power LED Indicator

Just under and to the right of the word “UNO” on circuit board, there’s a tiny LED next to the word ‘ON’. This LED should light up whenever plug Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong.

3.2.4 Reset Button

The Arduino has a reset button. When pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be 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.

3.2.5 TX RX LEDs

RX is short for receive, TX is short for transmit. In our 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.

3.2.6 Main IC

The main IC on the Arduino is slightly different from board type to board type, but is usually from the AT mega line of IC’s from the ATMEL Company. The black thing with all the metal legs is an IC, or Integrated Circuit. It is very important, as may need to know the IC type before loading up a new program from the Arduino software.

3.2.7 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.

3.2.8 Schematic Diagram

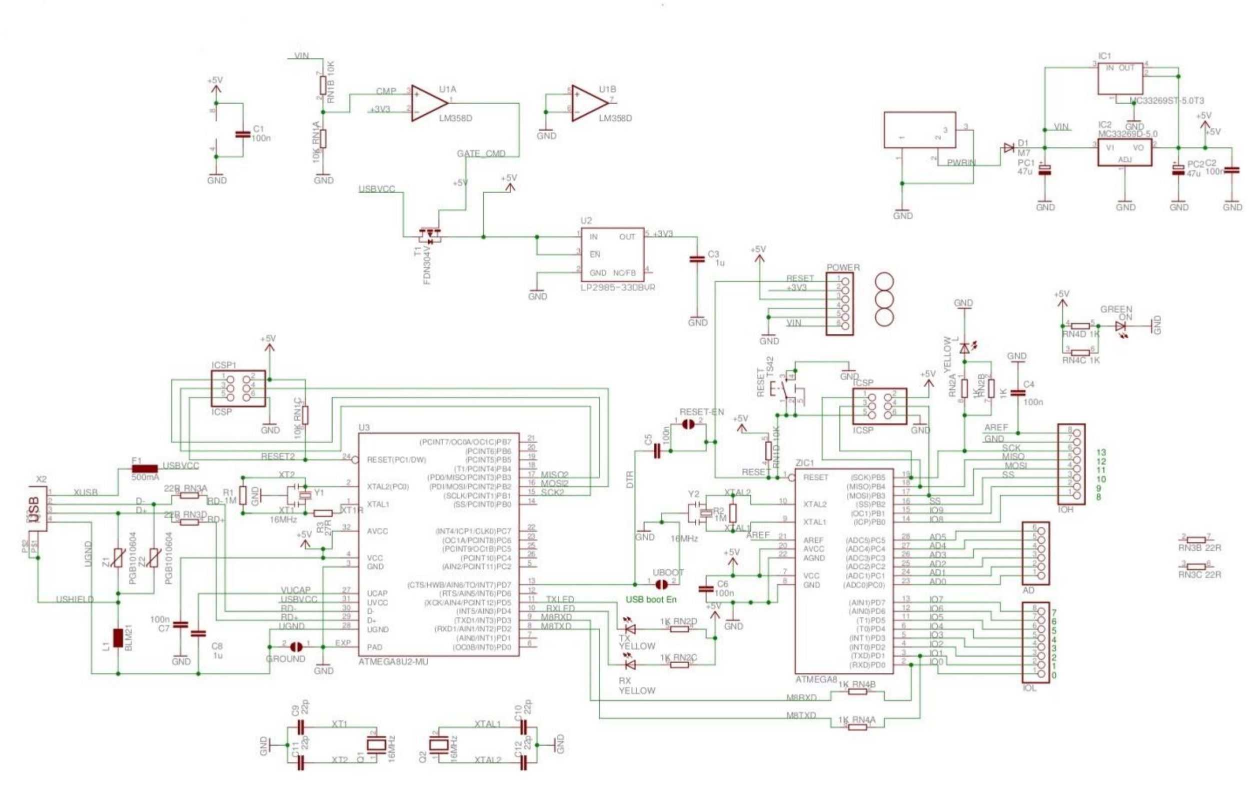

Figure 3.2: Schematic diagram for Arduino Uno Board

3.3 Working with Arduino Software

First download and install the Arduino IDE for Mac, Linux or Windows 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. Open Examples: 01.Basics: Blink. Click the toolbar button to upload it to your board.

3.3.1 The Integrated Development Environment (IDE)

Microcontroller needs software for programming. The Arduino board has its own integrated development environment (IDE).It is free and anyone can download it from its official. 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.

3.4 LDR (Light Dependent Resistor)

LDR is the short form of Light Dependent Resistor. It's also known as photo resistor. It is a light control variable passive resistor. The resistance of a photo resistor decreases when the light incident intensity of light increases. In other words it exhibits the photoconductivity. With the help of this Light Dependent Resistor we have developed sensor that can detect the difference of dark and light.

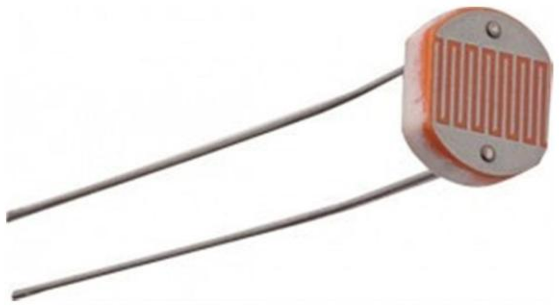

Figure3.3: Light Dependent Resistor

3.5 IC74164

DM74LS164 is a 8-bit shift register featured gated serial inputs with asynchronous clear. A low logical value at either input inhibits entry of the new data, and resets the first flip-flop to the low level at the next clock pulse passed, and starts providing complete control over the incoming data. A high logical value on either input enables the other input and it will then determine the state of the first flip-flop. Inputted data at the serial inputs may be changed while the clock is at HIGH or LOW, but only the information meeting the setup and hold time requirements will be proceed. Clocking occurs on the LOW-to-HIGH level transition to the clock input. All inputs will be diode-clamped to minimize transmission-line effects.

Figure3.4: IC 74164

3.5.1 Specifications

- IC type: 8 bit CMOS SIPO shift resister
- VCC = -0.5 to +7 volts
- V_{in} = -0.5 to +5 volts
- V_{out} = -0.5 to +5 volts
- DC input current = ±2.0 mA
- DC output current = ±2.0 mA
- Power dissipation = 500 (Max) mW
- Storage temperature = -60^{0}C to 150^{0}C
- Lead temperature = 300^{0}C (within 10 seconds)

3.5.2 Features

- Gated (enable/disable) serial inputs
- Fully buffered clock and serial inputs
- Asynchronous clear
- Typical clock frequency 36 MHz
- Typical power dissipation 80 mW

3.5.3 Pin Diagram of IC74164

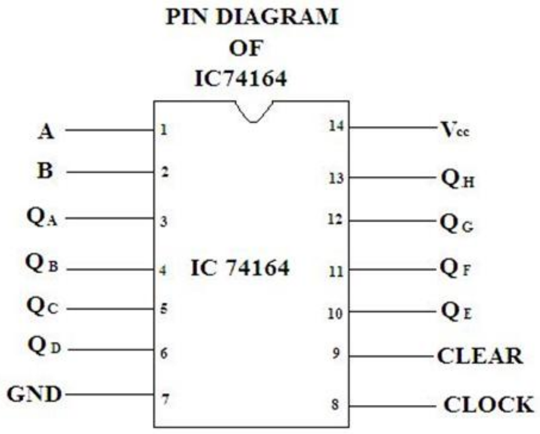

Figure3.5: Pin Diagram of IC74164

3.5.4 Schematic Diagram of IC74264

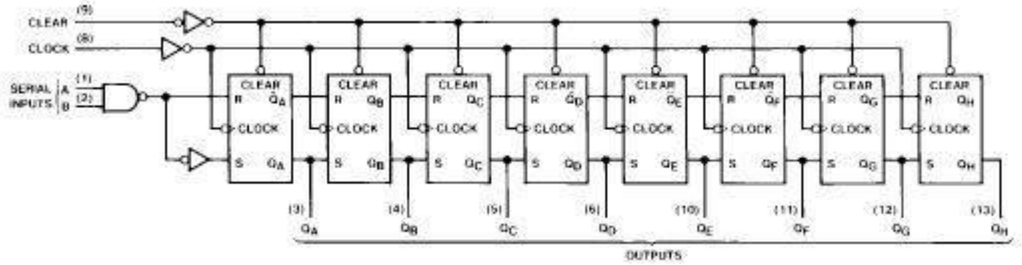

Figure3.6: Schematic Diagram of IC74264

3.6 BC547 Transistor

BC547 is a bi-polar NPN transistor. A transistor is used for transferring resistance, and is commonly used for amplifying current. A small portion of current at it's base provides a large current through it's collector and emitter. BC547 transistor is used in the switching purpose. And the maximum current gain of BC547 is 800. BC548 and BC549 are the equivalent transistors of BC5467.

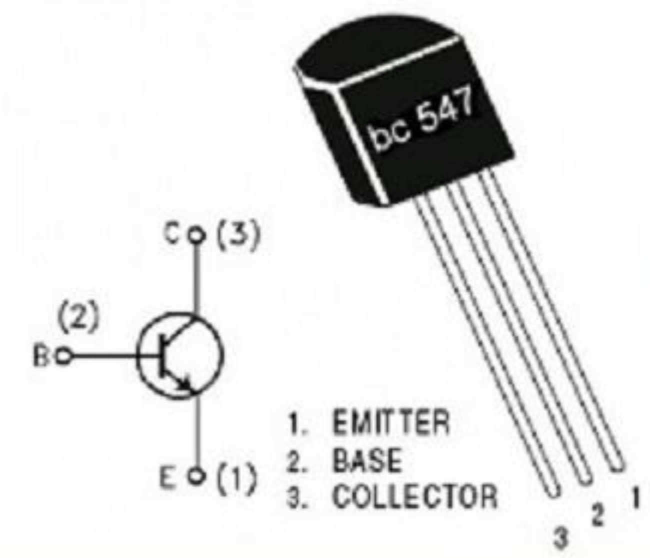

Figure3.7: Transistor BC547

A fixed DC voltage is required to the transistor terminals so that it can works in the desired region of it characteristic curves. This procedure is known as biasing. The transistor is biased so that it is partly on for all the input conditions. BC547 transistor is used via common emitter configuration of the amplifiers. The biasing mood is commonly use the voltage divider. The transistor is biased, as a result it remains fully on if there is a signal at the base for switching applications. If there is no base signal the transistor will get completely switched off.

3.6.1 Absolute Maximum Ratings (T a = 25oC)

Description	Symbol	Value	Unit
Collector Base Voltage	V_{CBO}	50	V
Collector Emitter Voltage (V_{BE} =0v)	V_{CES}	50	V
Collector Emitter Voltage	V_{CEO}	45	V
Emitter Base Voltage	V_{EBO}	6	V
Collector Current	I_C	100	mA
Collector Current Peak	I_{CM}	200	mA
Emitter Current Peak	I_{EM}	200	mA
Storage Temperature Range	T_{Stg}	-55 to +150	^{O}C
Junction Temperature	T_j	150	^{O}C
Base Current Peak	I_{BM}	200	mA

Table 3.2: Absolute Maximum Ratings of BC547 Transistor

3.7 GSM SIM900A

The Sim900A has 2 built in RS232 serial ports. One is for common communications to permit a computer to speak to the module and therefore the other is a "service" RS232 port (debug port) used for upgrading firmware, and doubtless alternative similar tasks.

The SIM900A communicates at TTL voltage levels (0 to supply voltage-0.1V) on these ports, this has an onboard MAX232 chip on the communications port only to translate RS232 at TTL level to the standard voltage levels (-12V to 12V) and might interface on to a computer.

The communications port is accessible at TTL levels if take away the jumpers close to the antennae connection. The debug port is available at TTL levels only, and to use it on the module.

The communications serial port pin out at standard voltage levels if any other versions of the module have slightly different TTL pin outs, so use a multi-meter and locate the ground pin/s. This modem can operate at both TTL 3.3V and 5V Logic Level therefore making it appropriate for controllers at 5V logic level (PIC, AVR). The Network LED indicates the varied standing of GSM module e.g. Power on, Network registration & GPRS connectivity. When the modem is powered up, this Network LED will blink every second. After the Modem registers in the network (takes between 10-60 seconds), this LED will blink in step of 3 seconds. The connection and interfacing of GSM/GPRS TTL SIM900A connections should be like

- TX (modem)-RX (converter TTL)
- RX (modem)-TX (converter TTL)
- Make ground (GND-GND) common

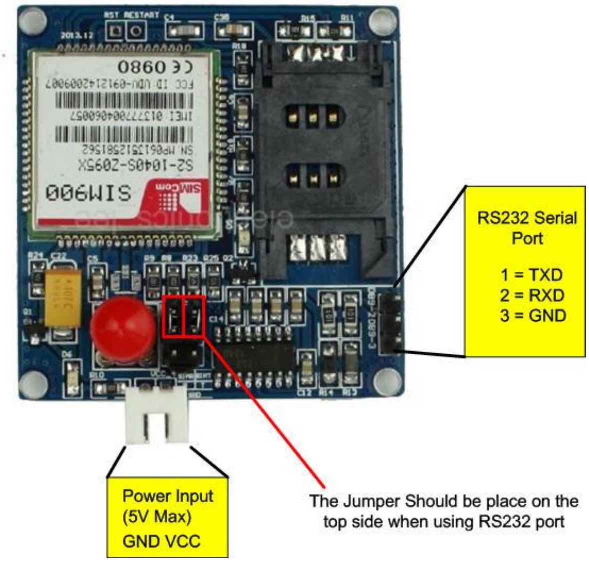

Figure 3.8: GSM SIM900A

3.7.1 Features of GSM Sim900A Module

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- Dual-Band 900/ 1900 MHz
- GPRS multi-slot class 10/8GPRS mobile station class B
- Compliant to GSM phase 2/2+Class 4 (2 W@ 850/ 900 MHz)
- Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- Low power consumption: 1.5mA(sleep mode)
- SMA connector with GSM Antenna.
- SIM Card holder.
- Built in Network Status LED
- Inbuilt Powerful TCP/IP protocol stack for internet data transfer over GPRS.
- Normal operation temperature: -20 °C to +55 °C
- Input Voltage: 4.5V-12V DC

Chapter 4

SYSTEM DESIGN AND ANALYSIS

4.1 Introduction

The project will help peoples who want to do something or make something with the help of Arduino. The project is designed in two parts, these are the software part and another one is the hardware designing. In this project we have used some basic components like LDR (Light Dependent Resistor), Shift resistor IC 74164, LM35, LED and Microcontroller. As a microcontroller we have used Arduino UNO Board here, which is perfect for building a new project or doing anything in the field of robotics or something that is smart to use. To use an Arduino we have to use the Arduino Software which free for all users and can be downloaded from http://www.arduino.cc. The Arduino program is based on C/C++ Programming language. And a huge collection of example is provided in their websites which are also free for all. And the software is one of the easiest software to use.

4.2 Hardware design

The hardware design of this project is very simple and efficient also. First of all we have designed the Dark sensor. The Dark sensor generally works as counter. And counter passes a signal after a significant number of signals or movement is detected. Then we have designed the whole circuit in the breadboard with the help of microcontroller. Arduino UNO board is the perfect choice to make the project more efficient and easy also.

4.2.1Full Circuit Design

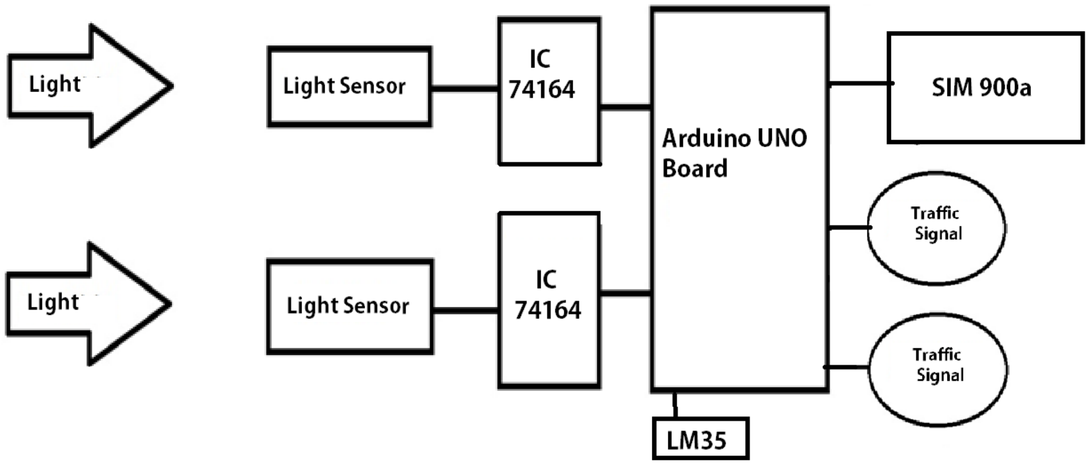

Figure 4.1: Full circuit design

In this control system we have used some basic components to design this circuit. These components are found easily at any electronics shops or markets. For counting we have made a light sensor with some basic components. The output of this circuit is connected to the Yellow, Green and Red lights of Traffic Signal.

4.2.2 Components Used in the System

To design the circuit of the project we have used the following component:

- One Arduino Uno Board
- Two bread board
- LM 35
- IC74164
- LDR

- Resistance
- Capacitor
- Laser Light
- Sim 900a
- LED
- Transistor

4.2.3 Sensing Unit

To detect or sense how many car or vehicle is passed we have designed a light sensor with the help of LDR(Light Dependent Resistor), some resistance, an NPN Transistor BC47. Designing the light sensor is very easy. The sensor can detect shadow. In this project we have used it as counter, that will count how many vehicle or car has passed the through the sensor.

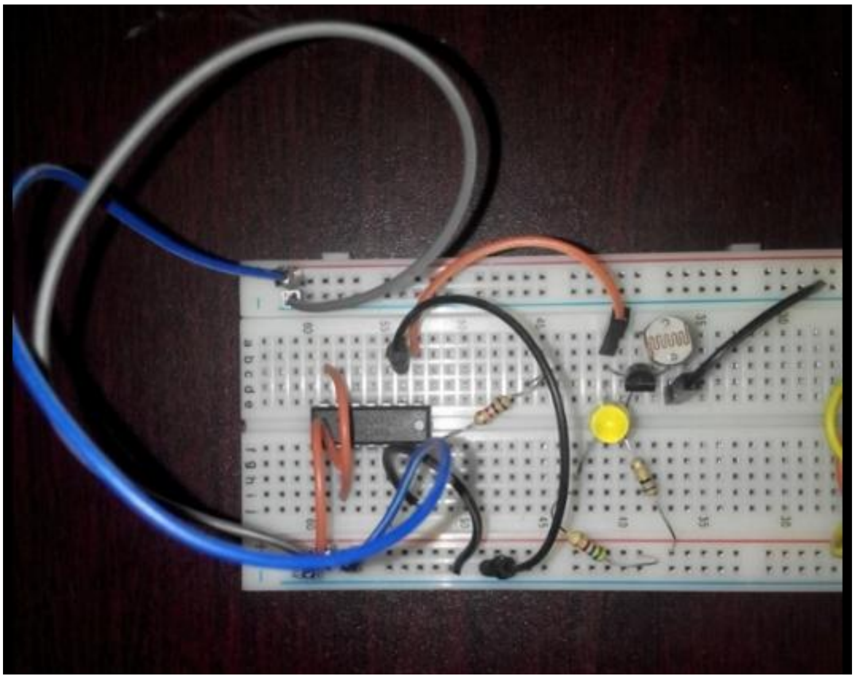

Figure 4.2: Circuit diagram of light sensor.

4.3 Software design

To design the software we have to write the program inside the Arduino software. After writing the program we need to verify the program that the program has any error or not. After verification the software is uploaded to Arduino UNO board from the software with the help of USB cable of Arduino UNO Board.

4.3.1 Installing Arduino

Arduino IDE software can be used in Windows, Mac or Linux operating system. The software can be found at their own website. We have downloaded the software from the following link **http://arduino.cc/en/Main/Software.** From that link you can found the latest version of the Arduino IDE software.

4.3.2 Verifying the Hardware

After installing the Arduino IDE now it's time to check the hardware is in proper condition or not. To check this first of all we have to connect the Arduino UNO Board to the computer with the help of USB cable of Arduino UNO Board. After connecting the hardware it's time to check the hardware with some small programs. In this case we have gone through the built in Blinking LED program. The program ran perfectly both in the IDE and Arduino UNO Board. And we got the confirmation that all the components are working properly and perfectly.

4.3.3 Arduino Language

The Arduino IDE basically based on the C and C++ programming language. It's follows the Wiring library function of C/C++. When we write a program or Arduino Sketch, we are implicitly making the use of Wiring library that is included in the Arduino IDE. This Wiring library is helping us doing program simply and easily. The wiring library is helping use writing program or Sketch of Arduino by using only two functions like the "setup()" and the "loop()" function. Basically the wiring language is inspired by the processing language. And the programming language of Arduino IDE is inherited from the processing language. In this programming language the equivalent functions are known as "setup()". For programming any program with Arduino IDE those two functions "setup()" and "loop()" is must to use. It is mandatory for all programs. Whenever there is no necessity of using one of those functions, the also we have keep those functions in the program to run the program successfully.

Chapter 5

IMPLEMANTATION AND TESTING

5.1 Implementation

After connecting all the parts as the circuit design diagram we have uploaded the programming code in the Arduino UNO Board. After uploading we have got positive result. The circuit works properly as it is designed.

5.1.1 Count Testing

In the count testing unit we have found that it works properly. Still the counter sometimes provides wrong results because of error noise in the IC74164 SIPO shift resister. Here first of all we tested the counter only with an LED, after finding that the counter works properly then we added that with the Arduino UNO Board.

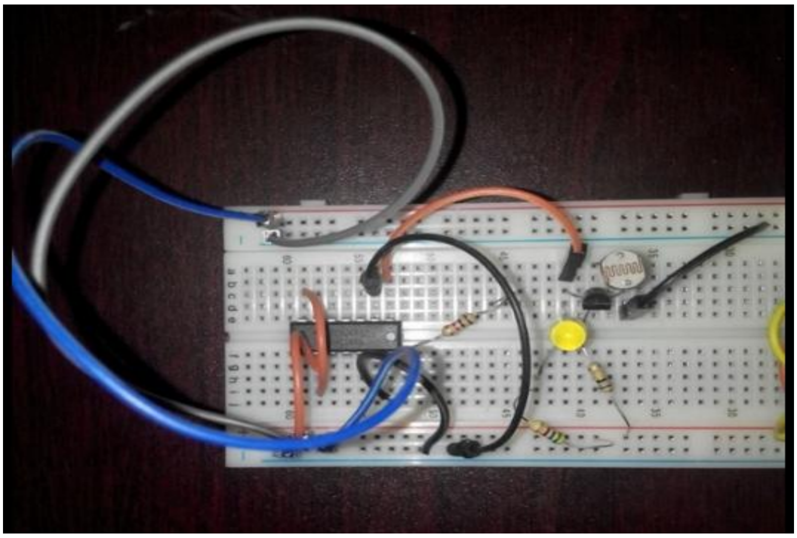

Figure 5.1: Counter Testing

5.1.2 Development of the Whole System

After testing all the components we have combined all the components together as like as the circuit design diagram and found that the application works correctly. When a certain number of vehicle passes through the Dark sensor one side of the road gets stopped and the other road will be free to move forward.

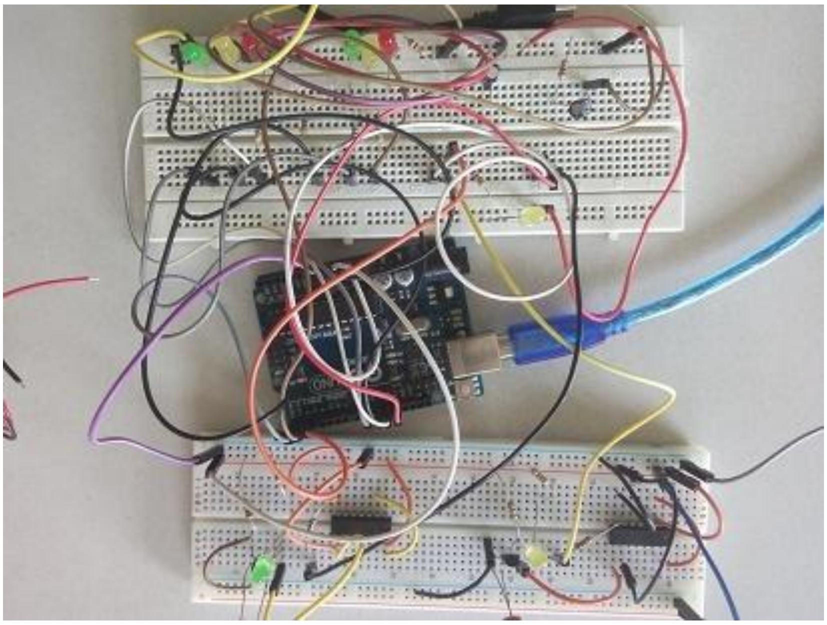

Figure 5.2: Final Development of Circuit

5.2 Product Commercialization

After completing the circuits together and developing the Intelligent Traffic Control System we connecting it into a demo Two way road and it worked correctly. As a result we can say that the Traffic control system we have developed is ready to use commercially.

Chapter 6

CONCLUSION

6.1 Future Work Scope

With the help of this project there is an opportunity of doing a big project in future. The applications those are stated above are some demo applications. But there is a huge possibility of developing this project. Because both the number of vehicles will increase and the roads will decrease proportionally with respect to time. Initially for the limitation of funding and time we have developed an intelligent traffic control system for two way road. To make the project more efficient we can use IR sensor. That will be more efficient than dark sensor. Here we can see a big future work scope in this sector. We have faith that we will be able to complete all the features needed for the ultimate application in near future.

6.2 Conclusion

The Intelligent Traffic Control System was designed and developed to decrease traffic congestions or traffic jam that is occurred by the traffic control system. We have used 5v and 3.3v from Arduino UNO Board. We have used LDR and BC547 NPN transistor to develop the Dark sensor. With the help of Dark Sensor we have counted the number of vehicle passed through the sensor.

At the end we have designed and developed a microcontroller based Intelligent Traffic Control System, and fixed the problem that we had before. Finally we have reached our goal successfully.

In this project we are succeeded to minimize the traffic congestions created by the fixed traffic light system with the help of microcontroller and improved algorithm. That is dependent on real time rather than a fixed time. We have noticed that our intelligent traffic control system is much efficient and the cost of production is very low. As a result "Intelligent Traffic Control System" is suitable enough to use commercially.

APPENDIX

Programming Codes for Testing

.While the intelligent traffic signal will be ON or OFF thus kind of matter is written in this code. The count code is:

```
#include <SoftwareSerial.h>

SoftwareSerialmySerial(9, 10);

int sensor=A1;

floattemp_read,Temp_alert_val,Temp_shut_val;

intsms_count=0,Fire_Set;

void setup()

{

pinMode(sensor,INPUT);

mySerial.begin(9600);

Serial.begin(9600);

delay(500);

}

void loop()

{

CheckFire();

CheckShutDown();
```

```
}

voidCheckFire()

{

Temp_alert_val=CheckTemp();

if(Temp_alert_val>0)

{

SetAlert(); // Function to send SMS Alerts

}

}

floatCheckTemp()

{

temp_read=analogRead(sensor); // reads the sensor output (Vout of LM35)

temp_read=temp_read*5;    // converts the sensor reading to temperature

temp_read=temp_read/10;  // adds the decimal point

returntemp_read; // returns temperature value in degree celsius

}

voidSetAlert()

{

while(sms_count<3) //Number of SMS Alerts to be sent

{

SendTextMessage(); // Function to send AT Commands to GSM module

}

Fire_Set=1;
```

```
}

voidCheckShutDown()

{

if(Fire_Set==1)

{

Temp_shut_val=CheckTemp();

if(Temp_shut_val<20)

{

sms_count=0;

Fire_Set=0;

}}}

voidSendTextMessage()

{

mySerial.println("AT+CMGF=1");    //To send SMS in Text Mode

delay(2000);

mySerial.println("AT+CMGS=\"+8801685549938\"\r"); // change to the phone number
you using

delay(2000);

mySerial.println("Tempurature !");//the content of the message

delay(200);

mySerial.println((char)26);//the stopping character

delay(5000);
```

```
mySerial.println("AT+CMGS=\"+8801685549938\"\r"); // change to the phone number you using

delay(2000);

mySerial.println("Temee !");//the content of the message

delay(200);

mySerial.println((char)26);//the message stopping character

delay(5000);

sms_count++;

}
```

Programming code of LED for Testing

We connected the LED with Arduino and write a program for operates that. Codes are:

```
#define westButton 3
#define eastButton 13
#define westRed 2
#define westYellow 1
#define westGreen 0
#define eastRed 12
#define eastYellow 11
#define eastGreen 6
#define yellowBlinkTime 500
booleantrafficWest = true; // west = true, east = false
intflowTime = 1000; // amount of time to let traffic flow
intchangeDelay = 1000; // amount of time between color changes
void setup()
{
// setup digital I/O pins
pinMode(westButton, INPUT);
pinMode(eastButton, INPUT);
pinMode(westRed, OUTPUT);
pinMode(westYellow, OUTPUT);
pinMode(westGreen, OUTPUT);
pinMode(eastRed, OUTPUT);
pinMode(eastYellow, OUTPUT);
pinMode(eastGreen, OUTPUT);
// set initial state for lights - west side is green first
digitalWrite(westRed, LOW);
digitalWrite(westYellow, LOW);
digitalWrite(westGreen, HIGH);
digitalWrite(eastRed, HIGH);
digitalWrite(eastYellow, LOW);
digitalWrite(eastGreen, LOW);
}
void loop()
{
if ( digitalRead(westButton) == HIGH ) // request west>east traffic flow
{
if ( trafficWest != true )
// only continue if traffic flowing in the opposite (east) direction
{
trafficWest = true; // change traffic flow flag to west>east
delay(flowTime); // give time for traffic to flow
digitalWrite(eastGreen, LOW); // change east-facing lights from green
// to yellow to red
```

```
digitalWrite(eastYellow, HIGH);
delay(changeDelay);
digitalWrite(eastYellow, LOW);
digitalWrite(eastRed, HIGH);
delay(changeDelay);
for ( int a = 0; a < 5; a++ ) // blink yellow light
{
digitalWrite(westYellow, LOW);
delay(yellowBlinkTime);
digitalWrite(westYellow, HIGH);
delay(yellowBlinkTime);
}
digitalWrite(westYellow, LOW);
digitalWrite(westRed, LOW); // change west-facing lights from red to green
digitalWrite(westGreen, HIGH);
}
}
if ( digitalRead(eastButton) == HIGH ) // request east>west traffic flow
{
if ( trafficWest == true )
// only continue if traffic flow is in the opposite (west) direction
{
trafficWest = false; // change traffic flow flag to east>west
```

```
delay(flowTime); // give time for traffic to flow

digitalWrite(westGreen, LOW);

// change west lights from green to yellow to red

digitalWrite(westYellow, HIGH);

delay(changeDelay);

digitalWrite(westYellow, LOW);

digitalWrite(westRed, HIGH);

delay(changeDelay);

for ( int a = 0 ; a < 5 ; a++ ) // blink yellow light

{

digitalWrite(eastYellow, LOW);

delay(yellowBlinkTime);

digitalWrite(eastYellow, HIGH);

delay(yellowBlinkTime);

}

digitalWrite(eastYellow, LOW);

digitalWrite(eastRed, LOW); // change east-facing lights from red to green

digitalWrite(eastGreen, HIGH);

}

}

}
```

References

- http://www.arduino.cc/en/Guide/Introduction
- http://www.arduino.cc/en/Main/Software
- http://www.arduino.cc/en/Tutorial/Blink
- http://www.ijater.com/Files/IJATER_03_06.pdf
- http://www.engineersgarage.com/electronic-components/transistor-bc547-datasheet
- http://www.digi.com/pdf/chart_xbee_rf_features.pdf