DIRE DAWA UNIVERSITY

DIRE DAWA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING (CONTROL)



"DENSITY BASED TRAFFIC LIGHT SYSTEM BY USING PIC MICROCONTROLLER"

For the impartial fulfillment for award of use B.SC Degree in electrical and computer engineering

ADVISOR NAME: PRO.RAJESH

Group Name **ID: NO**

1. MOHAMMED YUSUF ……R/881/04

2. MARSHA FIKADIE ……R/810/04

3.FELEKE ABEBE……R/454/04

4. ABDU ESMAIEL……R/021/04

SUBMISSION DATE 15/06/2016

DIRE DAWA ETHIOPIA

1. ACKNOWLEDGEMENT

First of all we would like to acknowledge the Almighty ALLAH who keeps us by its providence for all our day to day activity. We would also like to say thanks to our advisor assists Pro. RAJESH who supports us in all our work during our project work time.

In addition to this we would like to give thanks to all who helps us on this project Mr Solomon, Mr Yonas and Ato Leykun

2. ABSTRACT

Nowadays, traffic congestion is a main problem in major cities wherein the traffic signal lights are programmed for particular time intervals. However, sometimes the demand for longer green light comes in at the one side of the junction due to huge traffic density. Thus, the traffic signal lights system is enhanced to generate traffic-light signals based on the traffic on roads at that particular instant.

The proposed system works based on a PIC16F877 microcontroller. Proximity sensors are interfaced to the microcontroller. The proximity sensors are used for detecting the density of the vehicles which are placed within a fixed distance.

If the traffic density is high on a particular side, more priority is given for that side. The sensors continuously keep on sensing the density on all sides of the junction. While the sensors detect high density they correspondingly give first priority for a longer green light at that junction. The side with the next priority level follows the first priority level.

The system solves the wasting time and fuel problem and the problem of the emergency car which waits a lot in the road intersection as well as it solves the problem of the congestion that occurs when an accident in the path happens. The system can be used for the traffic at the road intersections.

3. MOTIVATION

The traffic lights that are in widespread use today do not do much intricate reasoning when deciding when to change the lights for the various road users waiting in different lanes.

How long the signal stays green in one lane and red in another is most often determined by simple timing that is calculated when the crossing is designed. Even though today's methods are robust and work well when the traffic load is distributed evenly across the lanes in the intersection, the systems are very inefficient because they are unable to handle various simple situations that arise throughout the day. Unnecessary waiting time in the signal can be avoided by determining in which side the green signal should be large during the traffic.

Contents

1.ACKNOWLEDGEMENT ii

2. ABSTRACT iii

3. MOTIVATION iv

CHAPTER -1 1

1.1.Introduction 1

1.2. Problem statement 2

1.3 .Proposed solution 2

1.4. Objective 2

1.4.1. General objective 2

1.4.2. Specific objective 3

1.5. Project layout 3

1.5.1. Path that we have followed 3

1.6. Future scope 4

1.7.Methodology 5

Chapter -2 6

Literature Review 6

2.1. Background 6

2.2.Benefit of Traffic Light Controller 6

2.3. Typesof Traffic Signals Components and Operations 7

2.4. BLOCK DIAGRAM 8

CHAPTER -3 9

HARDWAREDESCRIPTIONANDIMPLIMENTATION 9

3.1.Introduction 9

3.2.1. Features of PIC16F877 microcontroller 11

3.2.2.Description of the PIC16F877 microcontroller 12

3.3 .Pin configuration 13

3.3.1.Pin Description 13

3.4.LCD (liquid crystal display) 14

3.4.1.Description of LCD 14

3.4.2.LCD Commands and Instruction set 15

3.5.power supply unit 17
3.6.proximity sensor 18
3.7.Microcontroller oscillators 20
3.8.Light emitting diode(LED) 21
CHAPTER -4 23
SOFTWARE DESIGN AND IMPLEMENTATION 23
4.1. Description of Proteus and C compiler environment software 23
4.1.2. Proteus simulation environment 23
4.2. Proteus operating interface 23
4.2.1. Proteus VSM 23
4.2.2. Description of c compiler software 24
4.3. SOFTWARE IMPLEMENTATION 25
4.4. Flow chart 26
CHAPTER -5 28
System design and Result 28
5.1. System design 28
5.2. Result 30
5.3. Prototype 33
CHAPTER -6 34
CONCLUSION AND RECOMMENDATION 34
6.1. Conclusion 34
6.2. Recommendation 35
REFERENCES 36
Appendix 37
Appendix A 37
C code of density based traffic light control 37
Appendix B 46

List of figures and tables

List of figures

Figure 1: Flow chart of project development

Figure 2: block diagram of the system

Figure 3: basic block diagram

Figure 4: Pin diagram of microcontroller system

Figure 5**:** LCD pin diagram

Figure 6: Block diagram of the Power Supply Unit of the Micro Controller

Figure 7: proximity sensor

Figure 8: Proximity Sensor Circuit Block Diagram

Figure 9: crystal oscillator and its circuit diagram

Figure 10: LED

Figure 11: Circuit diagram traffic signal controller at normal case

Figure 12: flow chart

Figure 13: System design

Figure 14: Circuit diagram traffic signal controller

Figure 15: Circuit diagram of more density based traffic signal controller.

Figure 11: Hardware result

List of tables

Table 1: the key feature of 16F877

Table 2: LCD PIN description

Table 3: pin connection of LCD

Table 4:

Table 5:

Table 6:

Abbreviations

PIC……programmable interface controller

LED……..light emitting diode

LCD…….liquid crystal display

RAM……read access memory

ROM……..read only memory

HZ…….hertz

ALU……...arithmetic logic unit

I/O……input output

CPU……Central processing unit

CHAPTER -1

1.1. Introduction

Traffic congestion is one of the major problems, the world is facing today. Traffic monitoring and controlling is a difficult task. The aim of the traffic research is to optimize the flow of vehicular traffic and goods. The flow of the traffic constantly changes depending on the time of the day, day of the week and time of the year. At times, road work and accidents further influence the complexity. Hence, traffic light optimization is a complicated process. Even for single junctions there might be no obvious solution and the problem becomes even more complex for the multiple junctions, as the state of one light in one junction directly influences the flow of traffic towards many other lights. With the ever increasing vehicles on the road and the number of road users, the limited resources provided by current infrastructure lead to ever increasing travelling times. Hence, an density based control of traffic is a important issue to be considered. The Traffic Monitoring Authority need to find new methods of overcoming this problem like construction of new roads, flyovers etc., and also development of sophisticated traffic monitoring and control systems. One way to improve the traffic flow and safety of the current transportation system is to apply automation and intelligent control methods to roadside infrastructure and vehicles.

The goal of the driver is to reach at destination without wasting time and money. But resources provided by current infrastructures are limited. So the Traffic management at road is crucial to reduce waiting and traveling times, save fuel and money. Even though present traffic light controlling system handles the traffic at intersections, many times congestion, accidents happened due to its poor performance and the expansion of road infrastructure is not ultimate solution to the traffic congestion. It requires some smart mechanism that deals with the problems in the present traffic controlling system. The proposed system also provides the map feature, which shows the traffic situation of requested traffic signal.

The Traffic density is increasing at an alarming rate in developing countries which call for the need of Advance intelligent Traffic signals to replace the conventional manual and time based Traffic signal system .that consider the priority the density of traffic on the roads .that could be

done by making a bridge with proximity sensor on the road intersections and make the decision according to the information that the sensor sense. The timings of Red, Green lights at each crossing of road will be intelligently decided based on the total traffic on all adjacent roads. Thus, optimization of traffic light switching increases road capacity and traffic flow, and can prevent traffic congestions.

The primary objective of this proposed project is to identify the road with the higher density of vehicles the proposed system tries to reduce the traffic jams to some extent. The system is based on the microcontroller.PIC 16F877 microcontroller is used in the proposed system.

1.2. Problem statement

As the number of vehicles on road is increasing traffic congestion has increased. The main effect of this matter is the time wasting of the people on the road. Also at certain junctions, sometimes even if there is no traffic, people have to wait. Because the traffic lights remain red for the preset time period, the road users should wait until the light turn to green. Present traffic systems fail to provide traffic information including congested roads and alternate routes available in case of congestion.

1.3 .Proposed solution

The solution for this problem is by developing a system to control the traffic light monitor and a device sends for user on request. The delay for junctions that have high number of vehicles should take longer time than the delay for the junction that has less number of vehicles and developing a system which detects traffic flow on each road and set timing

1.4. Objective

1.4.1. General objective

The main objective of the project is to Density Based Traffic Light control

System by using pic microcontroller

1.4.2. Specific objective

- ✓ To solve and monitor the traffic light system and make it more efficient
- ✓ To solves the wasting time and fuel problem and the problem of the emergency car
- ✓ To solves the problem of the congestion that occurs when an accident in the path happens.
- ✓ To developing a system which detects traffic flow on each road and set timings.
  - ✓ Besides reducing the number of accidents
  - ✓ Provide orderly movement of traffic.
  - ✓ We can save considerable amount of time.

1.5. Project layout

We divide the whole project content into six specific chapters. In the 1st chapter we will simply introduce about the project. It contains introduction, Objectives, project limitation, scope of project and problem statement. In second chapter the discussion of literature review In the 3rd chapter the hardware implementation, the theory and description of the devices will be stated. We also introduce Software design and implementation in the 4rd chapter of the project. Next in the 5th chapter result and discussion of the overall project will be studied. Finally in the 6th chapter conclusion and recommendation of the project will be stated.

1.5.1. Path that we have followed

There are few stages that will be involved in order to work out the aimed objectives in this project. The project was beginning by discussion the project with our advisor. After finding objectives, problem and other related issues, the finding is done doing some literature review. After that the project development is begin. The project is divided in to two parts which are hardware and software development.

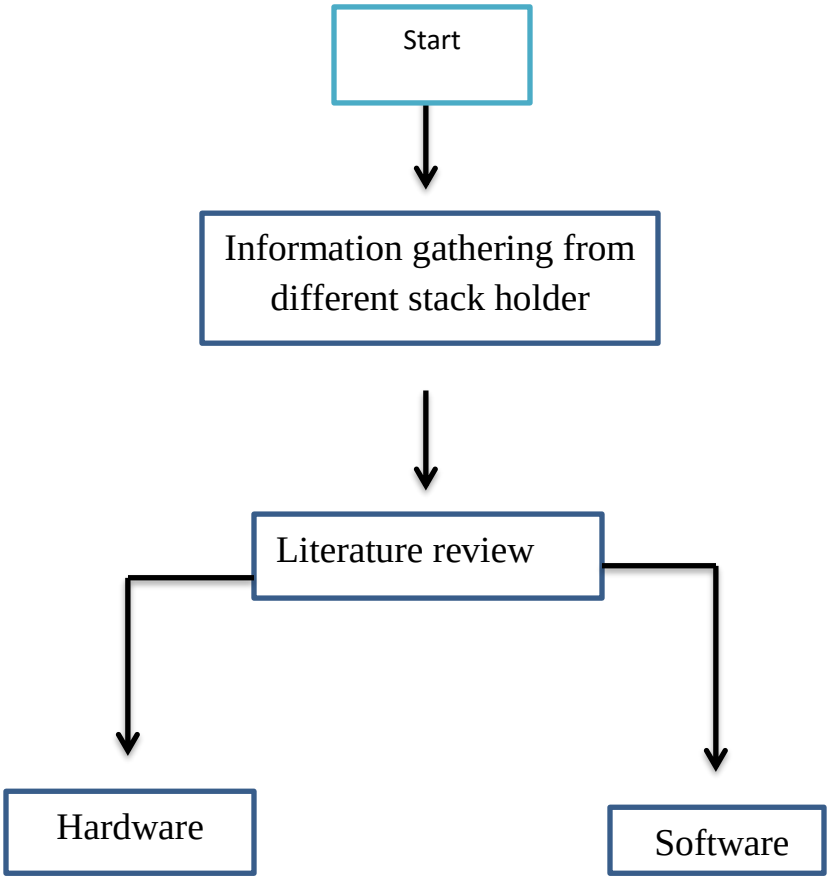


Fig.1.Flow chart of project development

1.6. Future scope

Further enhancements can be done to the prototype by microcontroller and proximity sensors. Currently, we have implemented system by considering one road of the traffic junction.
In future this system can be used to inform people about the condition of traffic at different places. This technique allows the operator to gather the recorded data from a far end to his home computer without going there. Based on the technology studied it is possible to develop cost effective, weather resistant products that have the potential for more sophisticated applications, including vehicle speed measurement and length classification.

1.7. Methodology

The methodology that the system will take to achieve it's goal is as the step bellow

1. The proximity Sensor sends the current traffic information to microcontroller.

2. Microcontroller then analyze the sensor
3. Microcontroller decides the traffic level of each section
4. Then Microcontroller takes the decision of glowing signals.

In order to measure the density of traffic on each side, proximity sensors will be kept on either sides of the road at a specific distance the traffic signal will be tuned with a default timing of 60 seconds of green light.

Should the road be cleared soon, by any chance, the lights will turn red after a short flash of yellow. Each road, however, will get access to its fair chance of 10 seconds of green light. Under conditions where the density is equal on all sides of the junction

The system designed here control the traffic movement for four roads respectively. In this system, we use proximity sensor to find the traffic density. When the proximity cut in any one of the roads, that road is considered as the higher traffic density road. So the road with the highest priority is cleared first. The system designed here control the traffic movement for four roads respectively by giving high priority to the road with high density of vehicles .Four proximity sensors are used in this proposed system, one for each road. Four proximity sensors are connected to 15, 16, 17 and 18 pins of PIC microcontroller respectively

Chapter -2

Literature Review

2.1. Background

A traffic light is a device that generates signals that is posted on road intersection in order to make the driving safe by using a universal color code. In other words a traffic light system is an electronic device that assigns right of way at an intersection or crossing or street crossing by means of displaying the standard red, yellow and green colored indications. An addition, it also works in conjunction with pedestrian displays to assign pedestrian crossing right of way .The traffic lights commonly for vehicles have three main lights, a red light that means stop, a green light that mean go and yellow that means ready to stop. But now days there are only two lights, a red light and a green light that mean stop and go respectively. The traffic lights have given many benefits to all road users. Besides reducing the number of accidents, it made the traffic flow smoothly and possibly could save people time.

The world's first traffic light Installed at an intersection in London in 1868, it was a revolving lantern with red and green signals .The lantern, illuminated by gas, was turned by means of a lever at its base so that the appropriate light faced traffic.

After the invention of automobiles, the situation got even worse. Police Officer William L. Potts of Detroit, Michigan, decided to do something about the problem. What he had in mind was figuring out a way to adapt railroad signals for street use. The railroads were already utilizing automatic controls. But railroad traffic traveled along parallel lines. Street traffic traveled at right angles. Potts used red, amber, and green railroad lights and about thirty-seven dollars' worth of wire and electrical controls to make the world's first 4-way three color traffic light

2.2. Benefit of Traffic Light Controller

When properly used, traffic control signals are important devices for the control of Vehicular in road. They assign the right- of-way to a choice of traffic movements and thereby deeply

influence traffic flow. Traffic control signals that are properly designed, located, operated, and maintained will have one or more of the following advantages**:**

1. Provide orderly movement of traffic.
2. Minimize completing movement.
3. Coordinated for continuous movement.
4. It clear traffic in less time.
5. It decreases waiting time on traffic light while there were no or less traffic on rest sides.
6. It automatically controls the traffic light According to congestion by providing proper time for Green light.

2.3. Types of Traffic Signals Components and Operations

Most traffic signals will have the following components or part:

I. Main display with red, yellow and green lights.

II. Traffic signal cabinet containing the traffic signal controller and Vehicle Detection Systems, either

III. Inductive loops or sensors

In the many cities, traffic signals mainly operate in three modes:

- **Fixed-time mode:** Under this mode, there are no detections for any approach. The signal continuously cycles regardless of actual traffic demand. Pedestrian walk signals are automatic and will cycle concurrently with the vehicular signal indication.
- **Semi-Actuated mode:** Under this mode, the detection system is present only on a minor cross street. When detection is activated, the green light on a major street is interrupted to allow the minor street traffic and pedestrians to safely enter the intersection. Pedestrian walk signals for crossing a minor street are automatic, while those for crossing a major street are not. Pedestrians crossing a major street must push the "pedestrian push button" to get the walk signal
- **Actuated mode:** Under this mode, there are detections for all approaches. The traffic signal is set to provide the green light "on demand "or only in the presence of vehicles

2.4. BLOCK DIAGRAM

The following figure describe the whole idea of the design

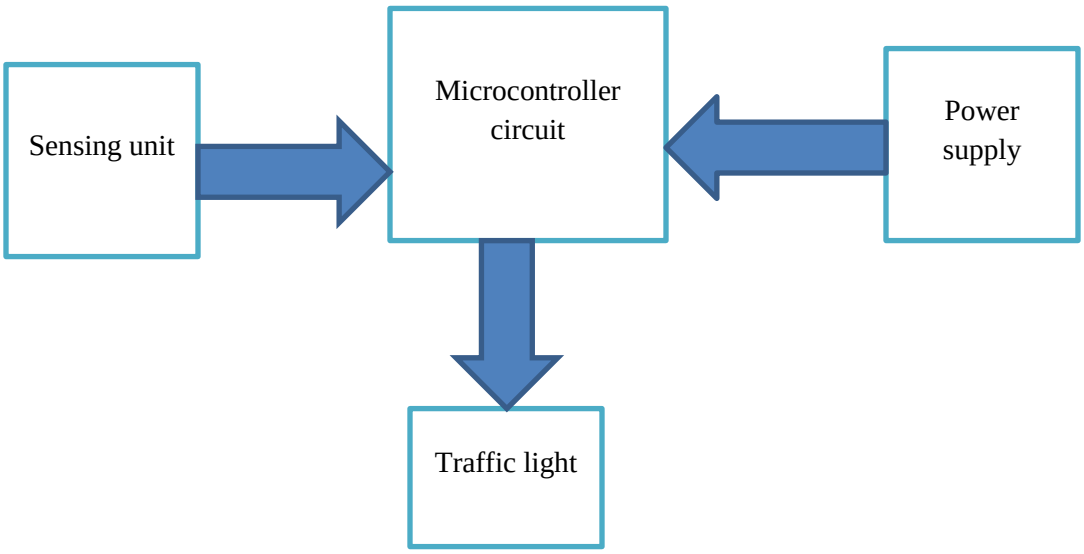


Fig.2. block diagram of the system

A. sensing unit:

The idea of using sensors to regulate the cycle time of a traffic light is starting to be used by the industry, especially in low traffic conditions. In this research the sensing unit that would be used is the Infrared sensor proximity sensor detect the vehicles and then according to the program in microcontroller the traffic light change its situation or remind stable

B. Microcontroller circuit:

Pic microcontroller will be used in the circuit and its pin will connect with the sensor and the traffic light

C. Power supply:

The circuit has to be supplied with voltage not less than 5 voltage and also the circuit must be grounded.

D. Traffic light

In the circuit there is two colors will be used the green means Go and the red means Stop so the LED will be used with those two colors and connect to the microcontroller

CHAPTER -3

HARDWAREDESCRIPTIONANDIMPLIMENTATION

3.1. Introduction

The whole system for hardware design block diagram is illustrated in the following fig. In this paper, the control card for density based advanced traffic light system is constructed for hardware design based on PIC16F877. Micro C programming is considered for software implementation. Related to development and innovation in semiconductor technology, PIC16F877 is introduced for both circuit design scheme and software programming to design and build a controlled device. The main proposed control card is fully controlled by the 8 bit microcontroller PIC16F877 which has an 8 Kbytes of ROM for program memory.

The basic block diagram of the proposed system is illustrated in Fig the heart of the system is the PIC 16F877 micro controller. The proposed system also comprises of a power supply unit, proximity sensor, and light emitting diodes (LEDs) and liquid crystal display (LCD).

The units of the desired hardware which belongs to LCD display with density based advanced light system on PIC16F877 microcontrollers are stated as below respectively.

Material required

- PIC 16F877 microcontroller
- Liquid Crystal Display (LCD)
- Light emitting diodes (LEDs)
- Proximity sensor
- Power supply

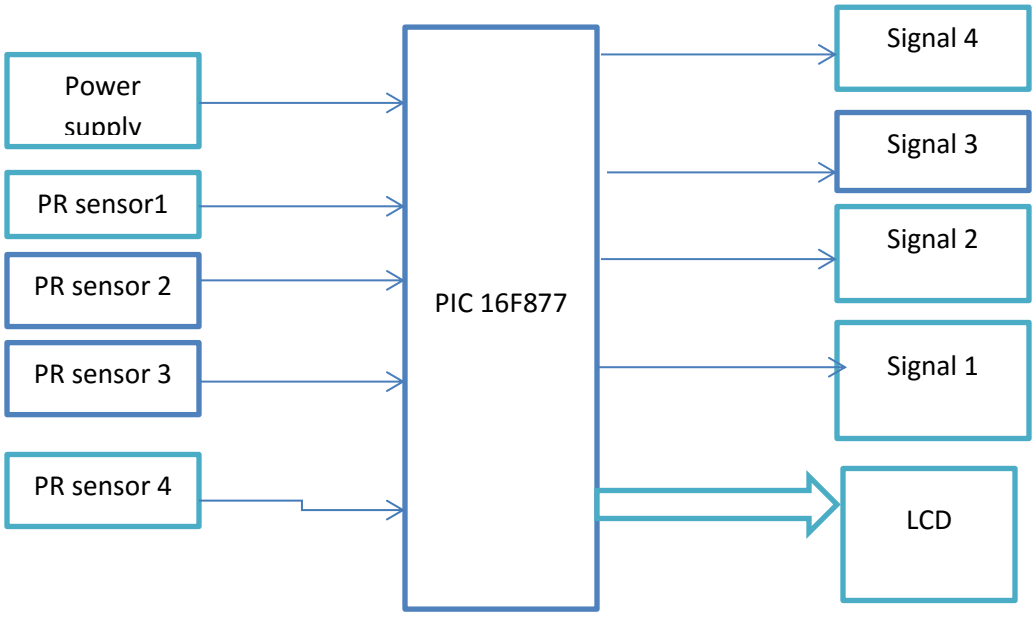


Fig.3. basic block diagram

The units of the designed and constructed hardware are explained as follows:

3.2. What is the PIC16F877 microcontroller?

The PIC microcontroller was designed using Harvard Architecture, with separate address spaces for data (SRAM), Program (FLASH, or EPROM) and EEPROM memory, PIC processor with few exception do not allow for direct access to their program memory space. The integrated circuit used in this paper operates at 8MHz clock frequency. As their structure are based on CMOS technology PICs consume very less energy.

PIC 16F877 is a 40-pin 8-Bit CMOS FLASH Microcontroller from Microchip. The core architecture is high-performance RISC CPU with only 35 single word1 instructions. Since it follows the RISC architecture, all single cycle instructions take only one instruction cycle except for program branches which take two cycles. 16F877 comes with 3 operating speeds with 4, 8, or

20 MHz clock input. Since each instruction cycle takes four operating clock cycles, each instruction takes 0.2 µs when 20MHz oscillator is used.

It has two types of internal memories: program memory and data memory. Program memory is provided by 8K words (or 8K*14 bits) of FLASH Memory, and data memory has two sources. One type of data memory is a 368-byte RAM (random access memory) and the other is 256-byte EEPROM (Electrically erasable programmable ROM).

3.2.1. Features of PIC16F877 microcontroller

The core feature includes interrupt capability up to 14 sources, power saving SLEEP mode, and single 5V In-Circuit Serial Programming (ICSP) capability. The sink/source current, which indicates a driving power from I/O port, is high with 25mA. Power consumption is less than 2mA in 5V operating condition.

The peripheral features include:

- 3 time blocks: Timer0 for 8-bit timer/counter; Timer1 for 16-bit timer/counter; andTimer2: 8-bit timer/counter with 8-bit period register, pre scalar and postscaler.
- Two Capture, Compare, PWM modules for capturing, comparing 16-bit, and PWM Generation with 10-bit resolution.
- 10-bit multi-channel (max 8) Analog-to-Digital converter module.
- Synchronous Serial Port (SSP) with SPI (Master Mode) and I2C2 (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls
- I/O ports.

Table.1.the key feature of 16F877

FLASH Program Memory (14-bit word)	8K Words
Data Memory (RAM)	368 Bytes
Data Memory (EEPROM)	256 Bytes
Interrupt	14
I/O Port	Port A,B,C,D,E
Timers	3
Capture/Compare/PWM Modules	2
Serial Communications	MSSP, USART
Parallel Communications PSP	PSP
10-bit Analog-to-Digital Module	8 channels
Instruction Set	35Nstructions

3.2.2. Description of the PIC16F877 microcontroller

The PIC16F877 provides the following standard features: 14Kbytes or 8k 14 bit-Flash, 368 bytes of RAM, 33 I/O lines, three 16-bit timer/counters,15 interrupt architecture of which 2 of them are external interrupts; a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the PIC16F877A is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

3.3 .Pin configuration

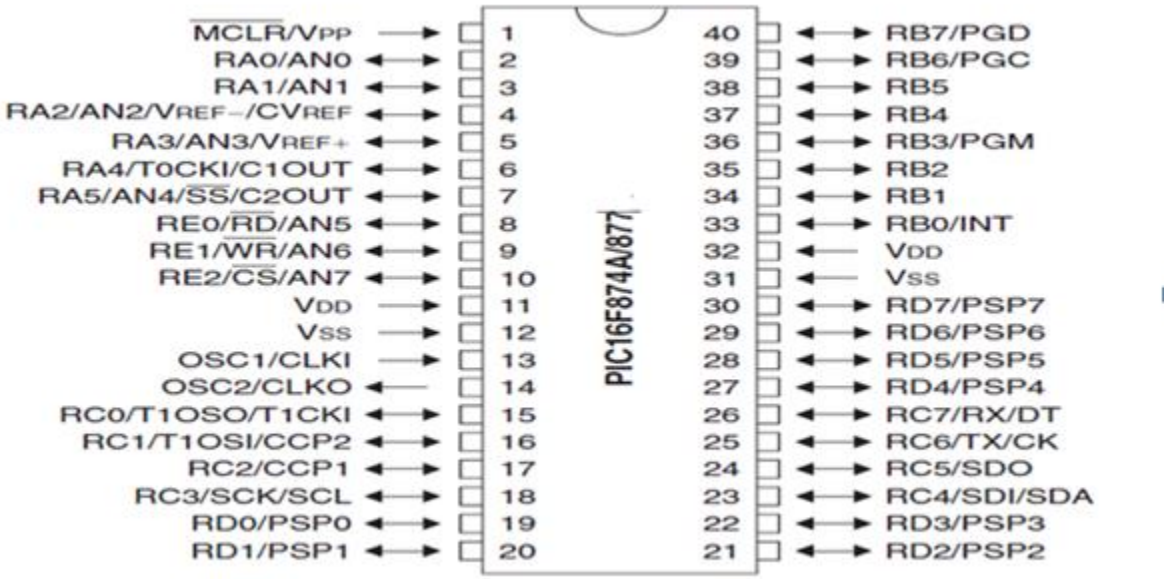


Fig.4. Pin diagram of microcontroller system:

3.3.1. Pin Description

The PIC16F877 micro controller is a 40-pin IC. The 11 th and 32th pins of the controller are VDD pin and the 5V dc supplies are given to these pins. The 12th and 31 th pins of the controller are VSS pin and are grounded. A 20 MHZ crystal oscillator is connected to 13 th and 14th pins of the PIC16F877 micro controller and two 20pf capacitors are connected to ground from 13 th and 14th pins. The 1 st pin is Reset pin.

Special function register:-

RA0-RA5: RA is a bidirectional port. That is, it can be configured as an input or an output. The number following RA is the bit number (0 to 5). So, we have one 5-bit directional port where each bit can be configured as Input or Output.

RB0-RB7: RB is a second bidirectional port. It behaves in exactly the same way as RA, except there are 8 - bits involved.

RC0-RC7 & RD0-RD7: RC & RD are bidirectional ports. They are 8-bit wide pins.

RE0-RE2: 3-bit wide, bidirectional.

VSS and VDD: These are the power supply pins. VDD is the positive supply, and VSS is the negative supply, or 0V. The maximum supply voltage that you can use is 6V, and the minimum is 2V.

OSC1/CLK IN And OSC2/CLKOUT: These pins are where we connect an external clock, so that the microcontroller has some kind of timing.

MCLR: This pin is used to erase the memory locations inside the PIC (i.e. when we want to reprogram it). In normal use it is connected to the positive supply rail.

INT: This is an input pin which can be monitored. If the pin goes high, we can cause the program to restart, stop or any other single function we desire. We won't be using this one much.

T0CK1: This is another clock input, which operates an internal timer. It operates in isolation to the main clock. Again, we won't be using this one much either.

3.4. LCD (liquid crystal display)

Microcontroller controlled LCD are widely used in many application, having replaced most of their LEDs because of their low power consumption and flexible graphics display. This intelligent LCD module can show 160 different characters. The entire system is supplied with a 5V power supply. Data are entered via the data line (D0-D7).

3.4.1. Description of LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. In this project we use the 16x2 LCD display; it‟s a very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD.

A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

3.4.2. LCD Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions.

There are four categories of instructions that:

- Designate LCD functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions



Fig.5. LCD pin diagram

The LCD Pin description is illustrates in the table below

Table.2.LCD PIN description

Pin No	Symbol	Function
1	Vg	Ground (Common Source of Electrons i.e. Ground) 0V
2	VDD	(Common Drain for Electrons i.e. High Voltage) 5V
3	VEE	(Common Emitter – used to control Contrast) Ground via resistor
4	RS	(Register Select) Used to tell LCD controller weather data on Data Bus Is a Command or Text Data. High Signal for Data and Low Signal for Command.
5	R/W	(Read Write) Used to tell LCD controller, if we wish to Read or Write To the LCD. High for Read, Low for Write.
6	EN	(Enable) Used for Enabling and Disabling the LCD
7	DB0	LCD Data Bus
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	Vee	Positive Terminal for Backlight (5V)
16	K	Negative Terminal for Backlight (0V)

Table.3. pin connection of LCD

Pin No	Name	Function
1	VSS	Ground
2	VDD	Positive supply
3	VEE	Contract
4	RS	Register select
5	R/W	Read write
6	E	Enable
7	D0	Data bit0
8	D1	Data bit 1
9	D2	Data bit 2
10	D3	Data bit 3
11	D4	Data bit 4
12	D5	Data bit 5
13	D6	Data bit 6
14	D7	Data bit 7

3.5. Power supply unit

The power supply unit is one of the most important one. This power supply unit shown in Fig 2 converts 230V AC to 5V DC .Power supply unit consists of a transformer, bridge rectifier, voltage regulator and capacitors.

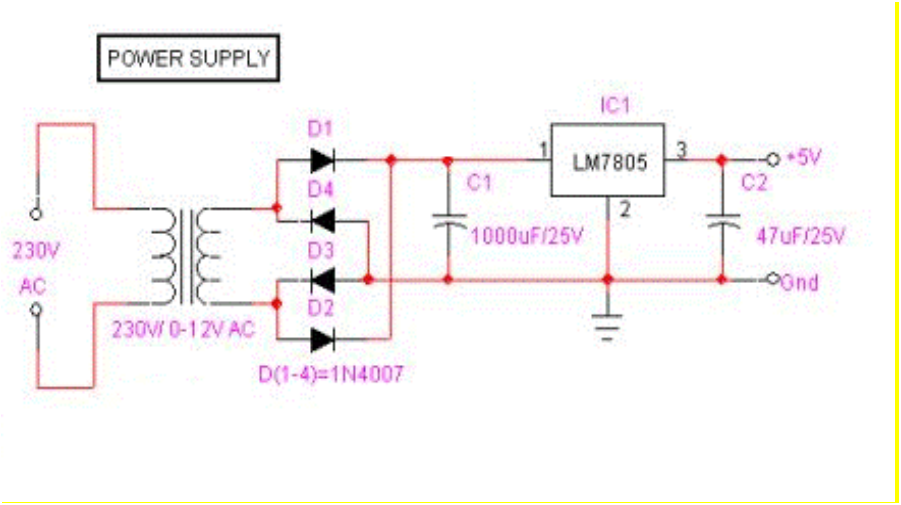


Fig.6. Block diagram of the Power Supply Unit of the Micro Controller

3.6. Proximity sensor

In this system, we use proximity sensor to find the traffic density. When the proximity cut in any one of the roads, that road is considered as the higher traffic density road. So the road with the highest priority is cleared first. The system designed here control the traffic movement for four roads respectively by giving high priority to the road with high density of vehicles .Four proximity sensors are used in this proposed system, one for each road. Four proximity sensors are connected to 15, 16, 17 and 18 pins of PIC microcontroller respectively. These 4 pins of the micro-controller drives the IR transmitter by generating 38KHZ square wave with 50% duty cycle. The outputs of four proximity sensors are connected to 12 different pins of the micro-controller.

The proximity sensor detects the proximity wave .So whenever any vehicle passes through proximity sensor the proximity sensor provides the low output at the corresponding pin. A sensor that can be used for detecting the presence of objects surrounding it without having any physical contact is termed as a proximity sensor. This can be done using the electromagnetic field or electromagnetic radiation beam in which the field or return signal changes in the event of the presence of any object in its surrounding. This object sensed by the proximity sensor is termed as a target.

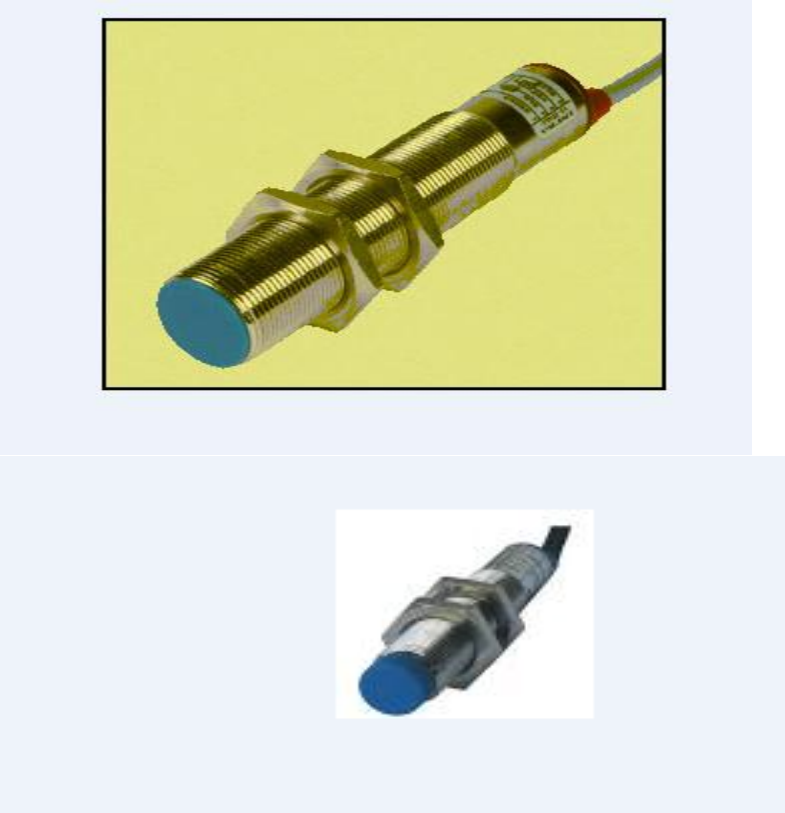


Fig.7. proximity sensor

Thus, if we discuss about different types of targets such as plastic target, metal target, and soon requires different types of proximity sensors such as capacitive proximity sensor or photoelectric proximity sensor, inductive proximity sensor, magnetic proximity sensor and so on. The range in which the proximity sensor is able to detect an object is termed as nominal range.

Unlike the other sensors, proximity sensors can last for long life and have very high reliability as there are no mechanical parts as well as no physical contact exists between the sensor and sensed object.

Proximity Sensor Circuit Diagram

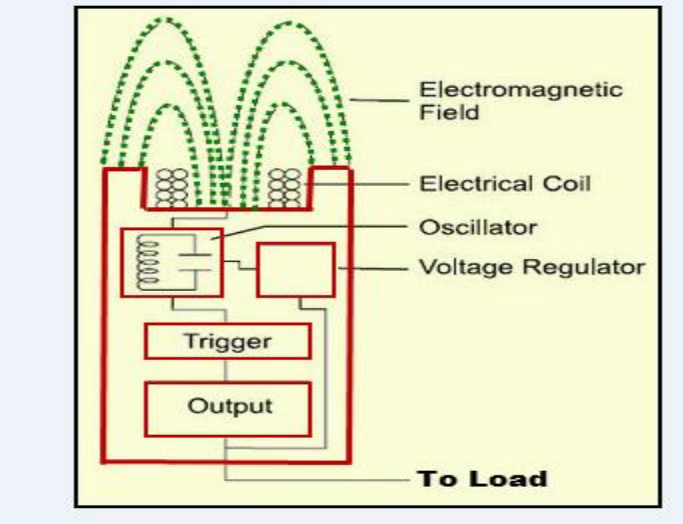


Fig.8. Proximity Sensor Circuit Block Diagram

Let us discuss about the inductive proximity sensor circuit which is most frequently used in many applications. The proximity sensor circuit diagram is shown in the above figure which consists of different blocks such as oscillator block, electrical induction coil, power supply, voltage regulator, etc.

Proximity Sensor Working Principle

The inductive proximity sensor circuit is used for detecting the metal objects and the circuit doesn't detect any objects other than metals. The above proximity sensor circuit diagram represents the field produced by the coil, which is generated by providing a power supply whenever, this field is disturbed by detecting any metal object (as a metal object enters this field), then an eddy current will be generated that circulates within the target.

3.7. Microcontroller oscillators

Oscillators are connected externally with the microcontroller to provide high frequency signal to the oscillator circuit in the micro controller. The oscillator circuit provides the clock signal to the microcontroller. The majority of clock sources for microcontrollers can be grouped into two types: those based on mechanical resonant devices, such as crystals and ceramic resonators, and those based on electrical phase-shift circuits such as RC (resistor, capacitor) oscillators.

Crystal and ceramic resonator-based oscillators (mechanical) typically provide very high initial accuracy and a moderately low temperature coefficient. RC oscillators, in contrast, provide fast startup and low cost, but generally suffer from poor accuracy over temperature and supply voltage. So, the optimal clock source for a particular microcontroller application is determined by a combination of factors including accuracy, cost, power consumption, and environmental requirements.

Crystal oscillator: -Crystal oscillator is a device that produces electrical oscillations at a frequency determined by the physical characteristics of a piezoelectric quartz crystal. This oscillator circuit is used to generate the device clock. The device clock is required for the device to execute instructions and for the peripherals to function. For our project we use 4 MHZ crystal oscillator because it has sufficient stability, high accuracy, and low cost and low power consumption as compared with other microcontroller oscillators

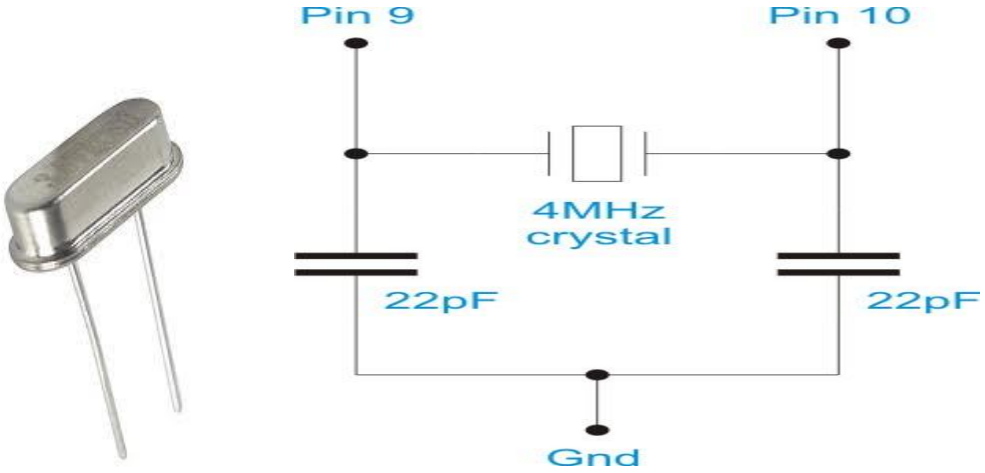


Figure.9. crystal oscillator and its circuit diagram

Light emitting diode (LED)

Short for Light-Emitting Diode, LED is a special semiconductor that illuminates when an electrical charge passes through it. LEDs are commonly green or red; however can be an assortment of other colors. Below are just a few examples of how an LED could be used with a computer. LED falls within the family of P-N junction devices. The light emitting diode (LED)is a diode that will give off visible light when it is energized. In any forward biased P-N junction there is, with in the structure and primarily close to the junction, a recombination of hole and electrons. This recombination requires that the energy possessed by the unbound free electron be transferred to another state. The process of giving off light by applying an electrical source is called electroluminescence.

LEDs produce more light per watt than incandescent bulbs; this is useful in battery powered or energy-saving devices. LEDs can emit light of an intended color without the use of color filters that traditional lighting methods require. This is more efficient and can lower initial costs. The solid package of the LED can be designed to focus its light. Incandescent and fluorescent sources often require an external reflector to collect light and direct it in a usable manner. When used in applications where dimming is required, LEDs do not change their color tint as the current passing through them is lowered, unlike incandescent lamps, which turn yellow. LEDs are ideal for use in applications that are subject to frequent on-off cycling, unlike fluorescent lamps that burn out more quickly when cycled frequently, or High Intensity Discharge (HID) lamps that require a long time before restarting. LEDs are currently more expensive, price per lumen, on an initial capital cost basis, than more conventional lighting technologies. The additional expense partially stems from the relatively low lumen output and the drive circuitry and power supplies needed. However, when considering the total cost of ownership (including energy and maintenance costs), LEDs far surpass incandescent or halogen sources and begin to threaten the future existence of compact fluorescent lamps.



Fig .10. LED

LED performance largely depends on the ambient temperature of the operating environment. Over-driving the LED in high ambient temperatures may result in overheating of the LED package, eventually leading to device failure. Adequate heat-sinking is required to maintain long life.

.

CHAPTER -4

SOFTWARE DESIGN AND IMPLEMENTATION

4.1. Description of Proteus and C compiler environment software

4.1.2. Proteus simulation environment

Proteus 8.10 is best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So it is a handy tool to test programs and embedded designs for electronics hobbyist. We can simulate our programming of microcontroller in Proteus 8.10 Simulation Software.

After simulating our circuit in Proteus 8.10 Software we can directly make PCB design with it so it could be a all in one package for students and hobbyists. So we think now we have a little bit idea about what is Proteus software.

4.2. Proteus operating interface

4.2.1. Proteus VSM

ISIS provides the development environment for PROTEUS VSM. This product combines mixed mode circuit simulation, micro-processor models and interactive component models to allow the simulation of complete micro-controller based designs.

ISIS provides the means to enter the design in the first place, the architecture for real time interactive simulation and a system for managing the source and object code associated with each project. In addition, a number of graph objects can be placed on the schematic to enable conventional time, frequency and swept variable simulation to be performed.

And it provides architecture in which additional animated models may be created by anyone, including end users. Indeed, many types of animated model can be produced without resort to coding. It created a range of simulator models for popular micro-controllers and a set of animated models for related peripheral devices such as LED and LCD displays. It is possible to simulate complete micro-controller systems and thusto develop the software for them without access to a physical prototype.

4.2.2. Description of c compiler software

A compiler is a computer program that transforms human readable source code of another computer program into the machine readable code that a CPU can execute. The act of transforming source code into machine code is called compilation. This is the opposite to the process of interpretation. Compilers are differentiated to many, which are depend on the operating system they are supported, license type and integrated development environment (IDE). Few of many of these compilers are CCS C Compiler, MPLAB, Ch. compiler, Clang compiler, GCC C compiler, Interactive C and etc.

Compiler options perform a variety of functions, such as setting compiler characteristics, describing the object code to be produced, controlling the diagnostic messages emitted, and performing some preprocessor functions. We can specify compiler options on the command line, in a configuration file, in our source code, or any combination of these techniques.

The overall compilation process consists of three phases: preprocessing, translation to object code, and linking. By default, a compiler invocation command invokes all phases of the compilation process to translate a program from source code to executable output. If file names for input and output files are specified when the compiler is invoked, it determines the starting and ending phases from the file name suffix (extension) of the input and output files. We can also create a particular type of output file at any compilation phase by using appropriate compiler options. In our project we select the C compiler to compile our program.

Compiler for programs written in C. C is a general purpose programming language which features economy of expression, modern control flow and data structures, and a rich set of operators. The C compiler is included in the default user path.The software to control the robot is written in C language. For our project, to compile our c code we have used CCS complier.

CCS compiler: - it is a C code compiler software, Proprietary license and compatible for all window operating system. The compiler contains Standard C operators and built-in libraries that are specific to the PIC registers. CCS C compilers contain Standard C operators and Built-in Function libraries that are specific to PIC registers, providing developers with a powerful tool for accessing device hardware features from the C language level. Standard C preprocessors, operators and statements can be combined with hardware specific directives and CCS provided built-in functions and example libraries to quickly develop applications incorporating leading edge technologies such as capacitive touch, wireless and wired communication, motion and motor control and energy management.

4.3. SOFTWARE IMPLEMENTATION

The circuit shown in the Fig. 11 is complete circuit diagram of an density based traffic light control at normal case and monitoring system with some peripherals interfaced to the PIC16F877 micro controller. Proteus software is used to make the circuit. Basically Proteus software is a circuit making and simulation window bade software. Micro controller receives the 20MHz from the crystal oscillator at OSC1 and OSC2 pin. Pin 1 of micro controller provides the manual reset of the micro controller. LCD contains three control signal RS (Register Select), R/W (Read/Write) and E (Enable).These three control signals are connected to the PIC micro controller pins 25,26(port D) respectively. LCD is used to display the status messages for the user. Port B is used to provide data to LCD to display as character

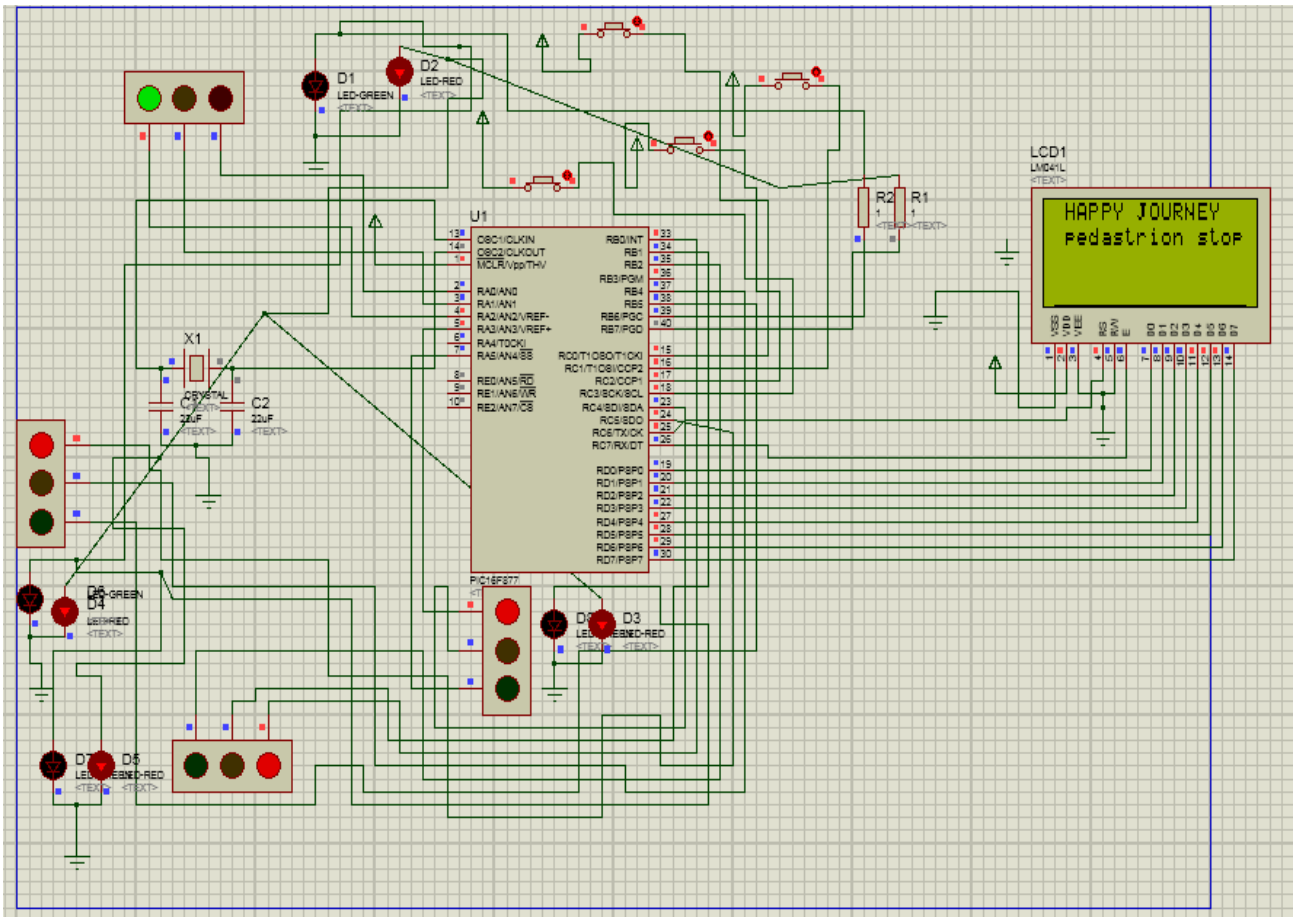


Fig.11. Circuit diagram traffic signal controller at normal case

4.4. Flow chart

To implement the systemPIC16F877A microcontroller was used, because of its simplicity for programming purposes. The program is used to control the motor based on the input comes from the two sensors. The Program can be clearly shown in the flow chart below.

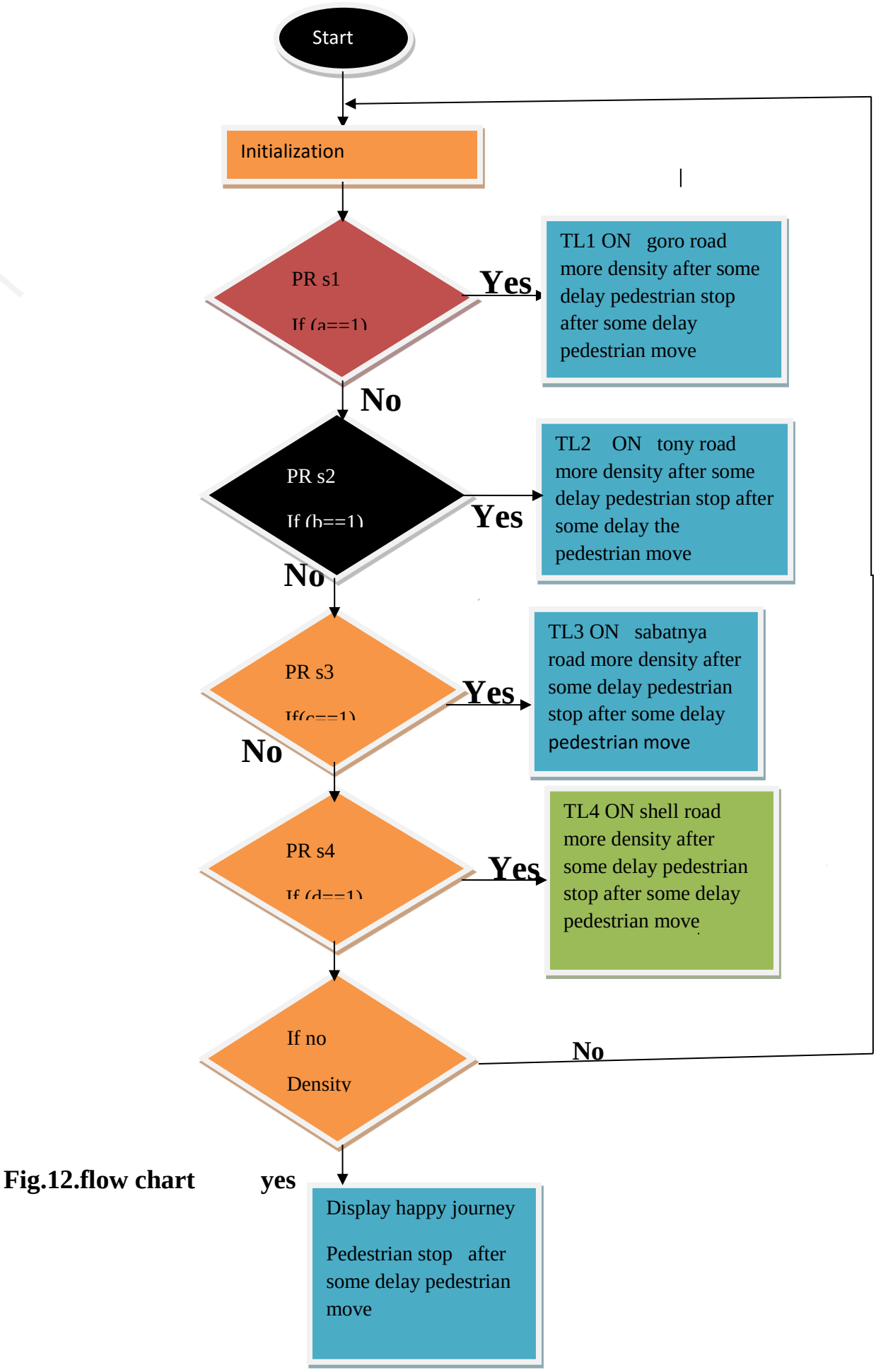


Fig.12.flow chart

CHAPTER -5

System design and Result

5.1. System design

The project uses simple electronic components such as LED as TRAFFIC LIGHT indicator, proximity sensor, LCD display and a MICROCONTROLLER for auto change of signal after a pre-specified time interval.

The proximity sensor should be in the both side of the street so when the car reach it could detect the car and send this information to microcontroller which decided the traffic light mode and which of the four traffic light should be in “GREEN” mode and which should be in “ RED “ mode .

- Microcontroller PIC 16F877 is the brain of the project which initiates the traffic signal at a junction.
- LEDs used are red, yellow and green.
- Red LED indicates “stop driving”
- Yellow LED indicates “start stopping”
- Green LED indicates “drive”.

Figure shows the drawing of the 4-way junction, where each way has its traffic light and counter

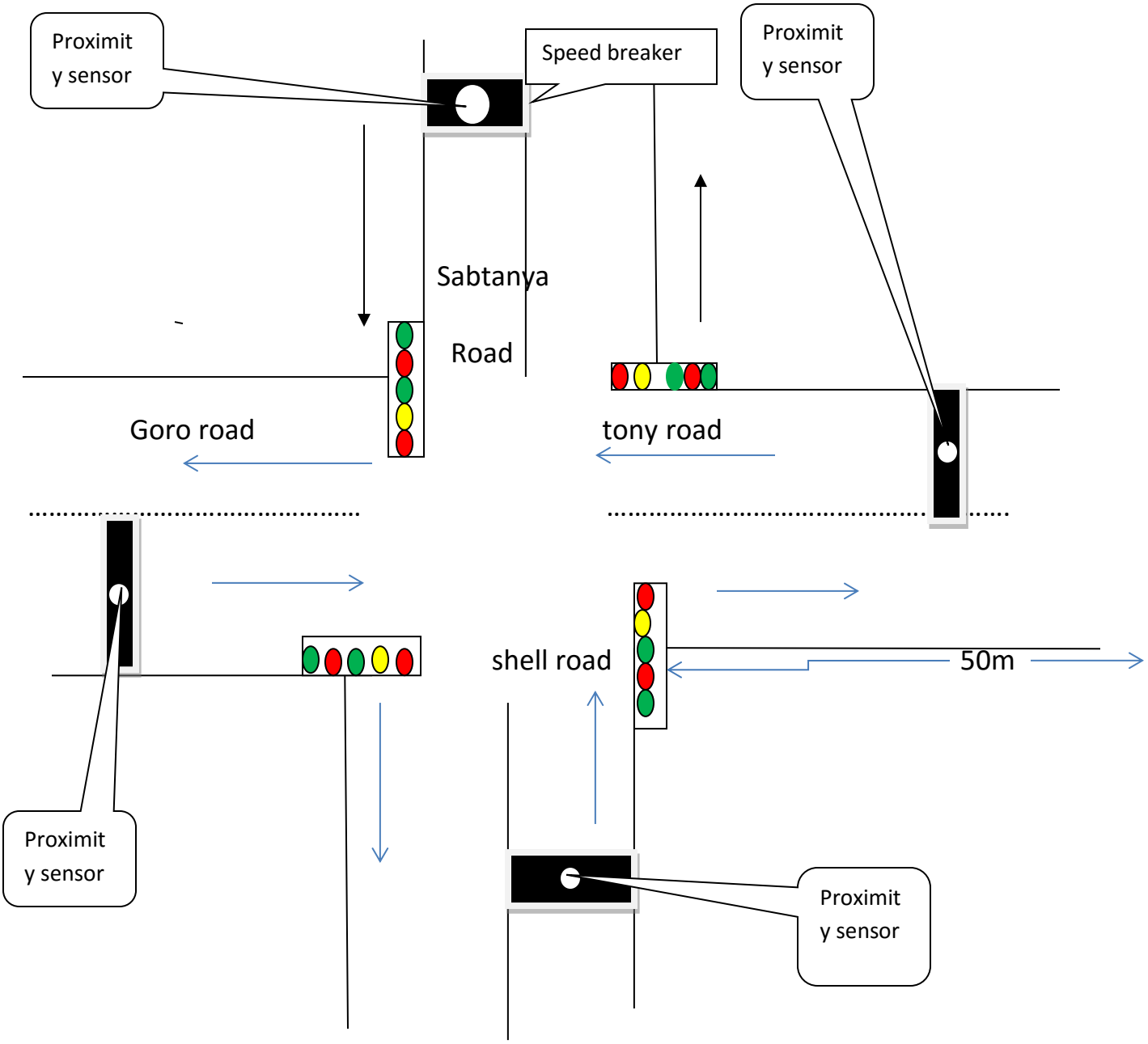


Fig .13. System design

5.2. Result

The simulation of the system has been done on Proteus Professional Software v8.1. PIC16F877 microcontroller based on Modified Harvard architecture is used in the system. Coding of the system has been done in Embedded C language. 16X2 LCD display has been used which is connected to PORT B of the microcontroller.

In our prototype, we have used PIC 16F877A [1] micro controller along with proximity sensors. A 16x2 (Liquid crystal display) LCD is interfaced with the micro controller to display the status of the signal with the traffic light model . The traffic light model is shown by red, yellow and green (Light emitting diode) LEDs. The code written is compiled in High-tech C Compiler and burned on microcontroller.

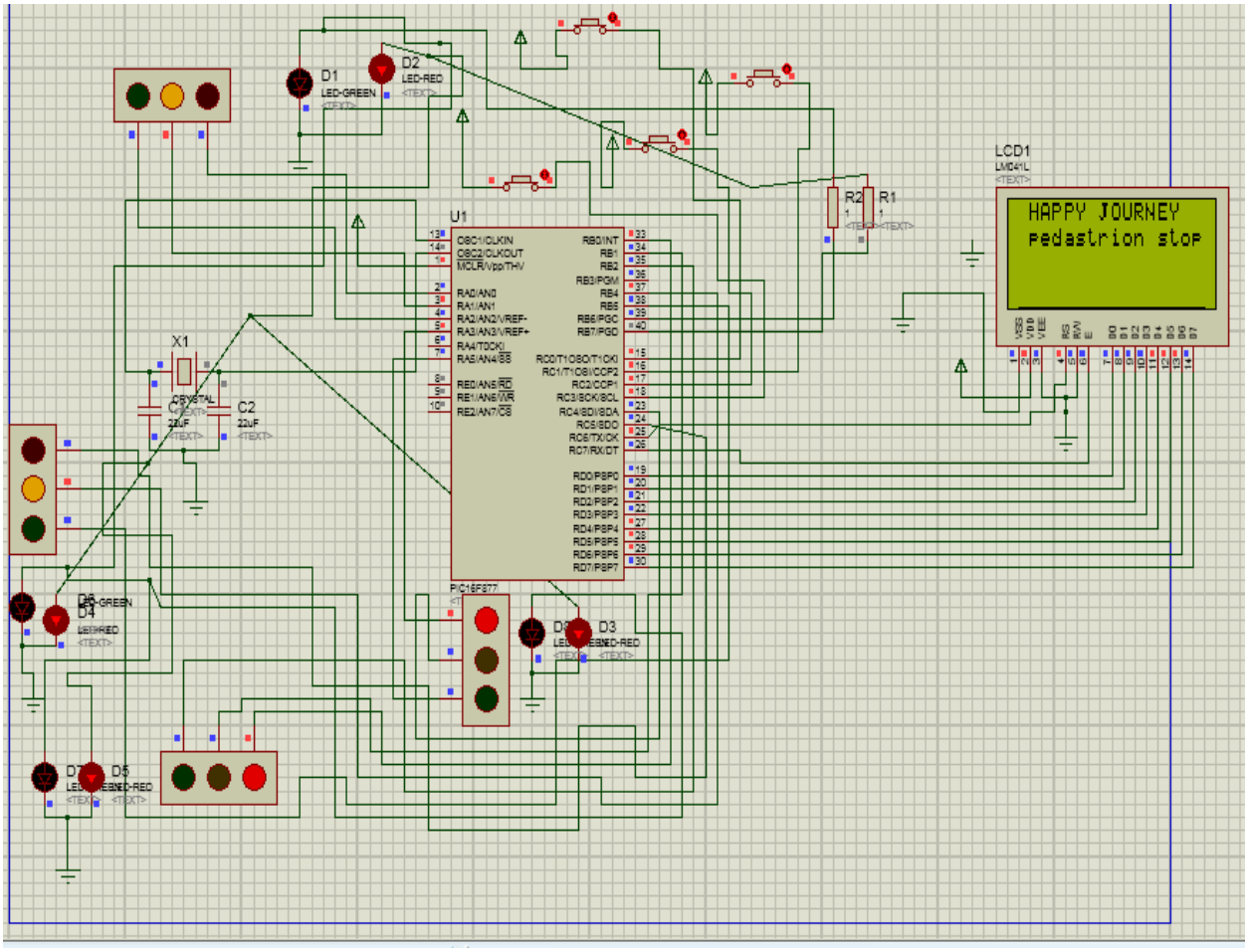


Fig.14. Circuit diagram of density based traffic signal controller.

The figure shows how the real processing happen in the density based traffic light system the proximity sensor from one to four send the information to microcontroller

All the information is process and the microcontroller send the information result to the traffic light from one to four one of them will be in “GREEN” mode and three will be in “RED” mode

The system designed here control the traffic movement for four roads respectively . In this system, we use proximity sensor to find the traffic density. When the proximity cut in any one of the roads, that road is considered as the higher traffic density road. So the road with the highest priority is cleared first. The system designed here control the traffic movement for four roads respectively by giving high priority to the road with high density of vehicles. Four proximity sensors are used in this proposed system, one for each road. Four IR sensors are connected to 15, 16, 17 and1 8 pins of PIC microcontroller respectively. These 4 pins of the micro-controller drives the IR transmitter by generating 38KHZ square wave with 50% duty cycle. The outputs of four proximity sensors are connected to 12 different pins of the micro-controller.

- if proximity sensor1 more density LCD to display goro road is more density after some delay pedestrian stop then after some delay pedestrian move
- if proximity sensor2 more density LCD to display tony road is more density after some delay pedestrian stop after some delay pedestrian move
- if proximity sensor3 more density LCD to display sabatnya road is more density after some delay pedestrian stop after some delay pedestrian move
- if proximity sensor4 more density LCD to display shell road is more density after some delay pedestrian stop after some delay pedestrian move

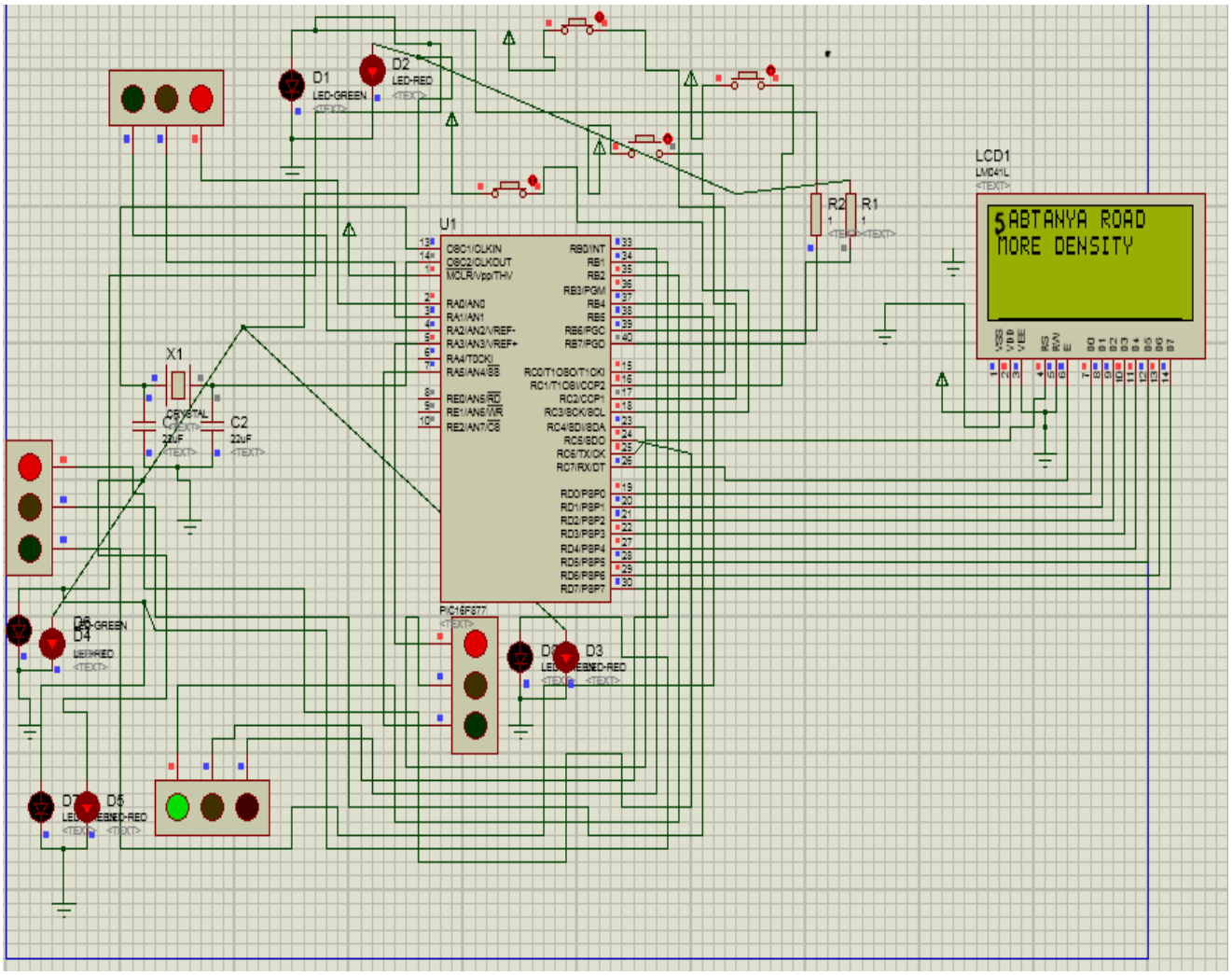


Fig.15. Circuit diagram of more density based traffic signal controller.

5.3. Prototype

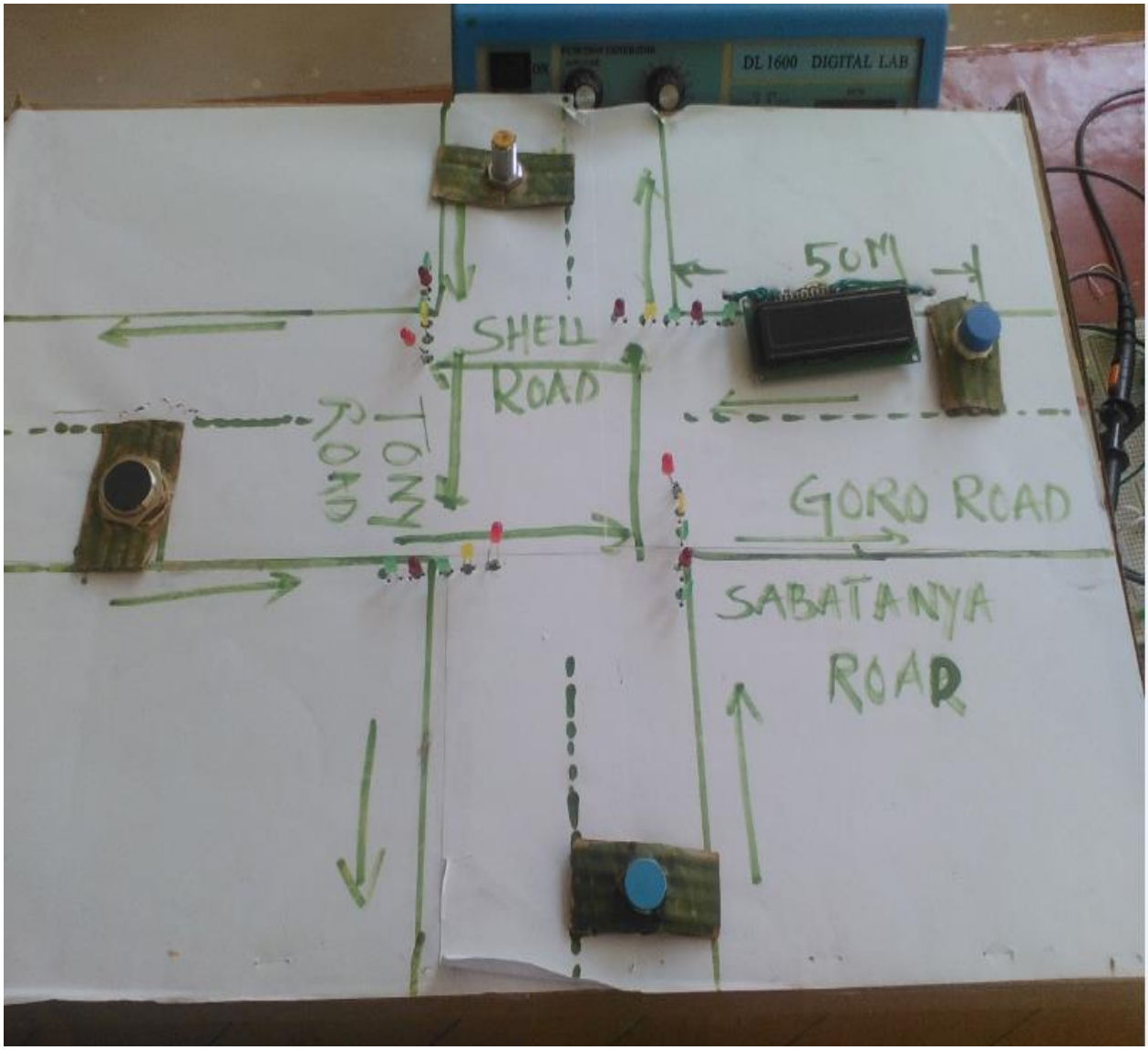


Fig .16.hardware result

CHAPTER -6

CONCLUSION AND RECOMMENDATION

6.1. Conclusion

In this proposed system, we have studied the optimization of traffic light controller in a city using proximity sensors and PIC 16F877 micro controller.Fig.2 shows the basic block diagram of the system and fig.11. Shows the complete circuit diagram of micro controller board. This proposed system reduces the possibilities of traffic jams, caused by high red light delays to an extent and successfully. Here we designed the system with the purpose to clear the traffic in accordance with priority .In this system, we use proximity sensor to find the traffic density. When the proximity cut in any one of the roads, that road is considered the higher traffic density road. So the road with the highest priority is cleared first. The program was successfully burned on the micro-controller using USB programmer. This proposed system is used to build a smart city with less traffic jams and it also helps the emergency vehicle to reach in time to the destination. The proposed system finds applications at toll gates. Further the project may be extended to the synchronization of all the traffic lights in the city.

6.2. Recommendation:

Great effort has been made in order to make this work complete in all aspect .but as with all project, there is always a room for improvement and further enhancements. There are a number of way in which this work can be carried forward

1- In this thesis, it could improve by adding readers to each traffic light.

2- In this thesis ,the sensing efficiency could be improved and the number of vehicles that the system take in consideration could be increased.

REFERENCES

[1] PIC16F877A, 8 bit PIC microcontroller, http://www.microchip.com.

[2] Marco Wiering. “Intelligent Traffic Light Control”. Institute of information and computing sciences, Utrecht University.

[3] W. Wen & C. L. Yang. “A dynamic and automatic traffic light control system for solving the road congestion problem”.

[4] “Automatic Traffic Control System”. SMEU Astana Solutions Automatic Traffic Control System.htmKarmakar, N. , Handbook of Smart Antennas for RFID Systems ,Wiley-IEEE Press, Pages: 13 -56 .

[5] Passive transponders,“http://www.rhydolabz.com/documents/rfid/EM4102_DS.pdf”.

[6] High-Tech C compiler, http://www.microchip.com.

Appendix

Appendix A

C code of density based traffic light control

```
#include "Includes.h"

#define R1 PORTAbits.RA0

#define Y1 PORTAbits.RA1

#define G1 PORTAbits.RA2

#define R2 PORTAbits.RA3

#define Y2 PORTAbits.RA4

#define G2 PORTAbits.RA5

#define R3 PORTBbits.RB0

#define Y3 PORTBbits.RB1

#define G3 PORTBbits.RB2

#define R4 PORTBbits.RB3

#define Y4 PORTBbits.RB4

#define G4 PORTBbits.RB5

#define a PORTCbits.RC0

#define b PORTCbits.RC1

#define c PORTCbits.RC2

#define d PORTCbits.RC3

#define e PORTCbits.RC4

#define f PORTCbits.RC5

#define pg PORTBbits.RB6
```

```
#define pr PORTBbits.RB7
//void main(void)
//{
void main(void)
{
   unsigned int Count,pulserate,count1,i;
            const char s1[30]="B.Sc FINAL PROJECT\0";
            const char s2[30]="DEN Bas TRAF\0";
            const char s3[17]="SHELL ROAD\0";
           const char s4[13]="GORO ROAD\0";
           const char s5[14]="SABTANYA ROAD\0";
            const char s6[13]="TONY ROAD\0";
            const char s7[26]="MORE DENSITY\0";
           const char s8[17]=" HAPPY JOURNEY\0";
           const char s9[25]=" pedastrion move\0";
           const char s10[25]=" pedastrion stop\0";
TRISA = 0x00;
TRISB = 0x00;
   //TRISC = 0xff;
            TRISCbits.TRISC0=1;
            TRISCbits.TRISC1=1;
            TRISCbits.TRISC2=1;
             TRISCbits.TRISC3=1;
   TRISCbits.TRISC4=0;
```

```
   TRISCbits.TRISC3=1;
   TRISCbits.TRISC5=0;
        PORTA = 0;
        PORTB = 0;
   InitLCD();
        ClearLCDScreen();                                   // Clear LCD screen
   WriteStringToLCD(s1);
   WriteCommandToLCD(0xC0);
   WriteStringToLCD(s2);
   //__delay_ms(20000);
   InitLCD();
//e=0;
//f=0;
//       WriteStringToLCD(s2);          // Write Message on LCD
   while(1)
   {
         if(a==0)
                {
                PORTA =0x0C;
        PORTB =0x89;
        f=1;
                  ClearLCDScreen();                    // Clear LCD screen
        WriteStringToLCD(s3);
        WriteCommandToLCD(0xC0);
```

```
WriteStringToLCD(s7);
__delay_ms(10000);
        ClearLCDScreen();
WriteStringToLCD(s10);
 //__delay_ms(5000);
        PORTA =0x09;
PORTB =0x49;
f=1;
ClearLCDScreen();
WriteStringToLCD(s9);
    __delay_ms(5000);
           }
           else if(b==0)
           {
              PORTA =0x21;
              PORTB =0x89;
f=1;
              ClearLCDScreen();                    // Clear LCD screen
WriteStringToLCD(s4);
WriteCommandToLCD(0xC0);
WriteStringToLCD(s7);
__delay_ms(10000);
ClearLCDScreen();
WriteStringToLCD(s10);
```

```
 //__delay_ms(5000);
        PORTA =0x09;
PORTB =0x49;
f=1;
ClearLCDScreen();
WriteStringToLCD(s9);
     __delay_ms(5000);
                                }
            else if(c==0)
            {
              PORTA =0x09;
              PORTB =0x8C;
f=1;
              ClearLCDScreen();                    // Clear LCD screen
WriteStringToLCD(s5);
WriteCommandToLCD(0xC0);
WriteStringToLCD(s7);
__delay_ms(10000);
ClearLCDScreen();
WriteStringToLCD(s10);
    //__delay_ms(5000);
    PORTA =0x09;
PORTB =0x49;
f=1;
```

```
PORTC=0x00100000b;
ClearLCDScreen();
WriteStringToLCD(s9);
   __delay_ms(5000);
                              }
            else if(d==0)
            {
              PORTA =0x09;
              PORTB =0xA1;
              ClearLCDScreen();                    // Clear LCD screen
WriteStringToLCD(s6);
WriteCommandToLCD(0xC0);
WriteStringToLCD(s7);
__delay_ms(10000);
       ClearLCDScreen();
WriteStringToLCD(s10);
       //__delay_ms(5000);
PORTA =0x09;
PORTB =0x49;
 f=1;
ClearLCDScreen();
WriteStringToLCD(s9);
     __delay_ms(5000);
            }
```

```
        else
        {
ClearLCDScreen();                       // Clear LCD screen
WriteStringToLCD(s8);
WriteCommandToLCD(0xC0);
WriteStringToLCD(s10);
            PORTA = 0x00;
             PORTB = 0x00;
             PORTA =0x0C;
             PORTB =0x89;
f=1;
__delay_ms(10000);
             PORTA =0x12;
             PORTB =0x89;
e=1;
f=1;
      __delay_ms(5000);

             PORTA = 0x00;
             PORTB = 0x00;
e=0;
             PORTA =0x21;
             PORTB =0x89;
f=1;
```

```
                __delay_ms(10000);
                PORTA =0x11;
                PORTB =0x89;
        e=1;
        f=1;
        __delay_ms(5000);
PORTA = 0x00;
PORTB = 0x00;
      // f=0;

//pedastrian move
      PORTA =0x09;
       //f=1;
      PORTB =0x49;
       f=1;
       e=0;
       PORTC=0x20;
      ClearLCDScreen();
      WriteStringToLCD(s9);
      __delay_ms(10000);
      ClearLCDScreen();
      WriteStringToLCD(s10);
      ClearLCDScreen();                    // Clear LCD screen
      WriteStringToLCD(s8);
```

```
WriteCommandToLCD(0xC0);
WriteStringToLCD(s10);

PORTA =0x09;
            PORTB =0x8C;
f=1;
      __delay_ms(10000);
            PORTA =0x09;
            PORTB =0x92;
f=0;
e=0;
//f=1;
             __delay_ms(5000);
            PORTA = 0x00;
            PORTB = 0x00;
            PORTA =0x09;
            PORTB =0xA1;
            __delay_ms(10000);
            PORTA =0x0A;
            PORTB =0x91;
e=0;
            __delay_ms(5000);
            PORTA = 0;
            PORTB = 0;
```

```
                }
    }
}
```

Appendix B

Register definition and pin description of PIC16F877 Microcontroller

Port A and TRISA Register

Port A is an 8-bit wide, bidirectional port. Bits of the TRISA and ANSEL control the PORTA pins. All Port A pins act as digital inputs/outputs. Five of them can also be analog inputs (denoted as AN):

	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Port B and TRISB Register

Port B is an 8-bit wide, bidirectional port. Bits of the TRISB register determine the function of its pins.

	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

-	Bit is unimplemented
R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Port C and TRISC Register

Port C is an 8-bit wide, bidirectional port. Bits of the TRISC Register determine the function of its pins. Similar to other ports, a logic one (1) in the TRISC Register configures the appropriate port pin as an input.

	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Port D and TRISD Register

Port D is an 8-bit wide, bidirectional port. Bits of the TRISD register determine the function of its pins. A logic one (1) in the TRISD register configures the appropriate port pin as input.

	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

R/W	Readable/Writable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

Port E and TRISE Register

Port E is a 4-bit wide, bidirectional port. The TRISE register's bits determine the function of its pins. Similar to other ports, a logic one (1) in the TRISE register configures the appropriate port pin as input. The exception is RE3 which is input only and its TRIS bit is always read as '1'.

					R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTE	-	-	-	-	RE3	RE2	RE1	RE0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

					R (1)	R/W (1)	R/W (1)	R/W (1)	Features
TRISE	-	-	-	-	TRISE3	TRISE2	TRISE1	TRISE0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

-	Bit is unimplemented
R/W	Readable/Writable bit
R	Readable bit
(x)	After reset, bit is unknown
(1)	After reset, bit is set

ANSEL and ANSELH Registers

The ANSEL and ANSELH registers are used to configure the input mode of an I/O pin to analog or digital.

	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

			R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	R/W (1)	Features
ANSELH	-	-	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

-	Bit is unimplemented
R/W	Readable/Writable bit
(1)	After reset, bit is set