Visvesvaraya Technological University, Belagavi.

PROJECT REPORT

on

"SMART DUAL SYSTEM OF AUTO GREEN CORRIDOR FOR EMERGECY SERVICES AND AN AGILE TRAFFIC LIGHT INFORMER"

Project Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

For the academic year 2019-20

Submitted by

USN	Name
MONISH GOWDA B R	1CR16EC086
NIDARSHAN K G	1CR16EC097
NIRANJANA N B	1CR16EC101
RAKESH H	1CR16EC125

Under the guidance of

Dr. IMTIYAZ AHMED B K

Associate Professor

Department of ECE

CMRIT, Bengaluru

Department of Electronics and Communication Engineering

CMR Institute of Technology, Bengaluru – 560 037

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to Certify that the dissertation work **"SMART DUAL SYSTEM OF AUTO GREEN CORRIDOR FOR EMERGECY SERVICES AND AN AGILE TRAFFIC LIGHT INFORMER"** carried out by **Monish Gowda B R, Nidharshan K G, Niranjan N B, Rakesh H** USN: **1CR16EC086, 1CR16EC097, 1CR16EC0101, 1CR16EC0125** bonafide students of **CMRIT** in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi,** during the academic year **2019-20**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Signature of Guide	Signature of HOD	Signature of Principal
________________	________________	________________
IMTIYAZ AHMED B K	Dr. R. Elumalai	Dr. Sanjay Jain
Associate Professor	Head of the Department,	Principal,
Dept. of ECE.,	Dept. of ECE.,	CMRIT,
CMRIT, Bengaluru.	CMRIT, Bengaluru.	Bengaluru.

External Viva

Name of Examiners Signature & date

1.

2

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it as our privilege to express the gratitude to all those who guided in the completion of the project.

We express my gratitude to Principal, **Dr. Sanjay Jain,** for having provided me the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank Dr. R. Elumalai**,** HOD**,** Department of Electronics and Communication Engineering, CMR Institute of Technology for the immense support given to me.

We express my gratitude to our project IMTIYAZ AHMED B K, Associate Professor for their support, guidance and suggestions throughout the project work.

Last but not the least, heartfull thanks to our parents and friends for their support.

Above all, We thank the Lord Almighty for His grace on us to succeed in this endeavor.

CONTENTS

CHAPTERS		TITLE	PAGE NO
1		**INTRODUCTION**	01
	1.1	INTRODUCTION	01
2		**LITERATURE SURVEY**	04
3		**METHODOLOGY**	08
	3.1	BLOCK DIAGRAM	08
	3.2	WORKING PRINCIPLE	10
4		**HARDWARE REQUIREMENTS**	14
	4.1	MICROCONTROLLER ARDUINO UNO	14
	4.2	ARDUINO IDE	24
	4.3	IR TRANSMITTER AND RECIEVER	31
	4.4	16*2 LCD DISPLAY	36
	4.5	POWER SUPPLY UNIT	41
	4.6	RELAY DRIVER CIRCUIT USING IC ULN2003	46
	4.7	RFID READ AND WRITE MODULE	49
	4.8	433 MHz RF TRANSMITTER AND RECIEVER	51
	4.9	DRIVER IC ULN2003	57
	4.10	ULTRASONIC SENSOR	58
5		**SOFTWARE REQUIREMENTS**	61
	5.1	EMBEDDED C	61
6		**RESULT**	62

7	**CONCLUSION**	63
	REFERENCES	64

LIST OF FIGURES

1.1 OVERVIEW OF THE MODEL 02

3.1 TRAFFIC SIGNAL UNIT BLOCK DIAGRAM 08

3.2 VEHICLE UNIT 09

3.3 FLOW CHART TRANSMITTER 11

3.4 FLOW CHART RECIVER 12

4.1 PIN DIAGRAM OF ARDUINO UNO 17

4.2 ARDUINO UNO 21

4.3 ARDUINO IDE 24

4.4 IR TX AND RX 31

4.5 WORKING OF IR SENSOR 32

4.6 IR SENSOR CIRCUIT DIAGRAM 32

4.7 IR LED 34

4.8 PHOTO DIODE 34

4.9 PIN DIAGRAM OF LM358 35

4.10 16X2 LCD PINOUT DIAGRAM 36

4.11 16X2 LCD DISPLAY 36

4.12 16X2 LCD PIN DIAGRAM 37

4.13 IC78XX 43

4.14 CIRCUIT DIAGRAM OF +5V AND +12V BRIDGE RECTIFIER REGULATED POWER 43

4.15 PDO IC ULN2003A 47

4.16 ULN2003 INTERNAL SCHEMATIC DIAGRAM 47

4.17 RELAY DRIVER 48

4.18 INTERFACING 4 RALAY DRIVER WITH ULN2003A 49

4.19 RFID RC522 51

4.20 TRANSMITTER CIRCUIT 53

4.21 RECIVER CIRCUIT 54

4.22 HT12E ENCODER 55

4.23 HT12D DECODER 56

4.24 ULN2003 DRIVER IC 57

4.25 ULN2003 INTERNAL SCHEMATIC DIAGRAM 58

4.26 ULTRASONIC SOUND WAVES TRANSMITTER AND RECIEVER 58

4.27 WORKING OF ULTRASONIC SENSOR 59

4.28 ULTRASONIC SENSOR 60

LIST OF TABLES

4.1 IMPORTANT COMMAND CODES FOR LCD 39

4.2 STARTTING ADRESS FOR EACH CHARACTER WITH THEIR PRINTING COMMANDS 41

CHAPTER 1

INTRODUCTION

Road traffic congestion becomes a major issue for highly crowded metropolitan cities like, Chennai, Bangalore. Ambulance service is one of the major services which gets affected by traffic jams and one more problem we face as we drive around town on our daily commute, you may find yourself stuck in traffic and getting fuel or poor gas wastage. During heavy Rain and in the foggy condition we are not able see a traffic signal, addition to this a large truck in front of our vehicle blocks signal view for a driver. To smoothen the ambulance movement and the proper visibility of signal lights we have come up with the solution of “Smart dual system of auto-green corridor for emergency services and agile traffic informer. The proposed system connects both the vehicles and the traffic signal stations using RF network.

The main principle is the identifying the traffic light signals in-vehicle itself at traffic intersections or in unsignalized locations and to make autogreen signal for emergency vehicles. By using this virtual traffic lights in vehicle itself driver can know which signal shows in traffic intersections. This will be useful in heavy rain and fog conditions. Traditional or Conventional traffic lights are vital for keeping up the activity ability and security at intersections. Be that as it may, unsignalized places are as yet visit in genuine rush hour gridlock situations.

The enhancement of movement capacity and security at unsignalized convergences is a basic issue that requires attention. As of late the creative idea of virtual movement lights has been proposed as an approach to determine the issue, in view of vehicle to foundation correspondence advances. Virtual movement lights include the appearance activity light data inside vehicles rather of through ground movement lights. Virtual movement light is a practically identical novel idea construct for the most part with respect to vehicle to vehicle correspondence and that permitting the light activity installations in the vehicle itself traffic lights has to be viewed, for this the LED and the controller has to be placed inside the vehicle dash board.

The installation of traffic light information in vehicles is additionally a testing vision assignment in LED design acknowledgment, with acknowledgment and recognizable proof as its essential advances. on the part of utilization of shading data LED lights are used. We ascribe to virtual activity lights connected in driving condition in-vehicle movement lights, to be striking from those utilized as a part of movement reproductions. For the use of vehicle activity lights at signalized intersections, it was discovered that when the ongoing condition of regular movement lights was anticipated on the LED lights shield to drivers in vehicle. Obstacle identification will be done with the help of ultrasonic sensor. The distance to the obstacle will be displayed on the LCD display.

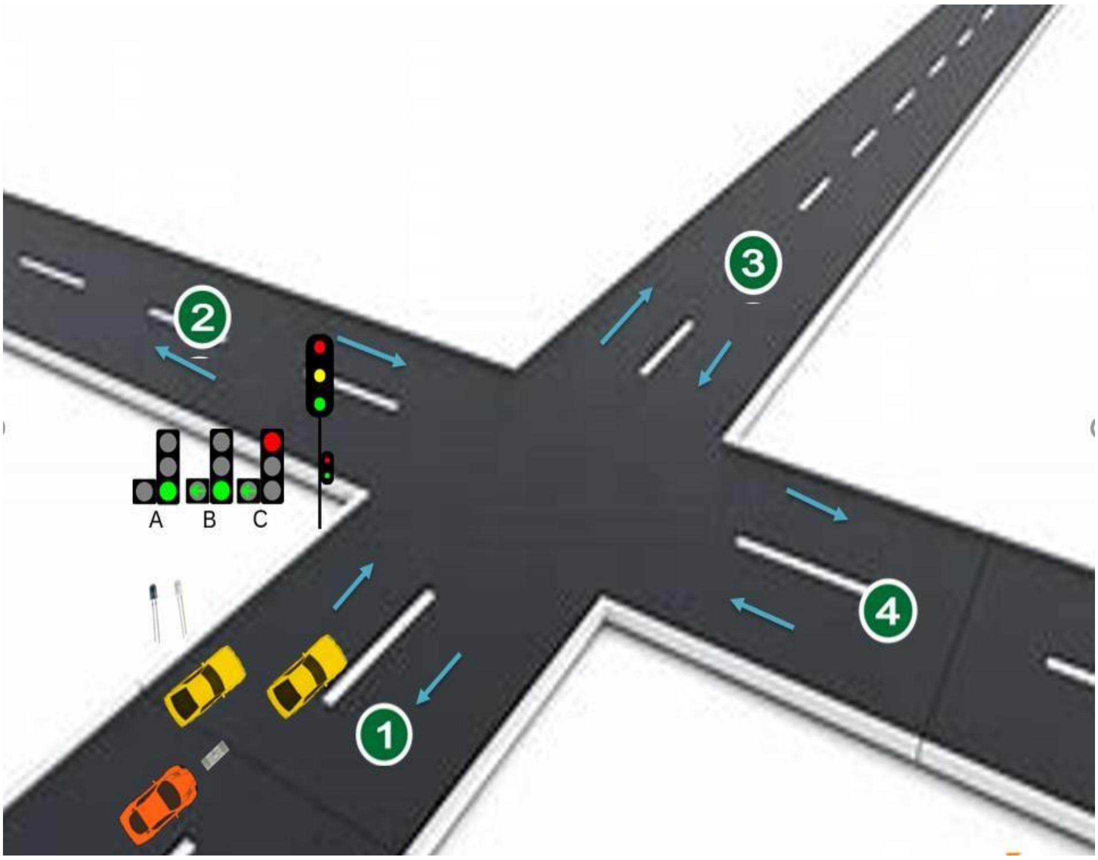

Fig:1.1 Overview of the Model

As shown in the fig 1.1, the proposed model is designed for the junction where 4 roads meet which are named from 1 to 4. The traffic signal monitoring is done manually. There are four conditions that will be shown in the traffic post. They are as follows.

1. ONLY GREEN
2. ONLY RED
3. GREEN WITH FREE LEFT
4. RED WITH FREE LEFT

When the vehicle moves near to the traffic post the status of the current status of traffic light signal will be displayed in the LCD display which will be placed near to the driver inside the vehicle with the help of RF transmitter and Receiver placed in vehicles and traffic junctions which helps for communication between them. The particular signal is made Green for some time and after the ambulance passes by, it regains its original flow of sequence of signaling If, this scheme is fully automated, it finds the ambulance spot, controls the traffic lights. This system controls the traffic lights and save the time in emergency periods. Thus, it acts as a life saver project.

In addition to this, Obstacle identification will be done and the distance of the obstacle will be displayed. Free left will also be displayed in the LCD when the vehicle goes in contact with IR sensor which will be installed at the left side of the road.

CHAPTER 2

LITERATURE SURVEY

- Yang, B., Zheng, R., Shimono, K., Kaizuka, T. and Nakano, K., 2017. Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalized intersections. IET Intelligent Transport Systems, 11(2), pp.76-83.
- Ground traffic lights are essential for maintaining traffic efficiency and safety at intersections. However, unsignalized intersections are still frequent in actual traffic environments. With the development of new forms of vehicular communication, in-vehicle traffic lights can assist drivers at unsignalized intersections.
- The authors proposed two types of in-vehicle traffic lights to assist drivers; these corresponded to two-way and all-way stop-controlled intersections. They adopted gap acceptance theory and a first-come-first-served strategy to determine passing priority for the two types of intersections, respectively. They then conducted a driving simulator experiment involving 23 participants, to investigate driver behaviours elicited by the proposed system. They prepared four experimental conditions with combinations of in-vehicle traffic lights and auditory warnings.
- The authors' experimental results indicated that in-vehicle traffic lights were associated with significantly longer post-encroachment times and a significantly shorter maximum brake stroke. In terms of eye-gaze behaviours, the percentage of gaze concentration to the road centre area and mean glance durations were deemed acceptable for the avoidance of visual distraction, when in-vehicle traffic lights were presented via a head-up display. Therefore, their analysis of driver behaviours indicates that in-vehicle traffic lights can effectively provide driver assistance at unsignalized intersections.

- Zhang, K., Sheng, Y. and Li, J., 2012. Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. IET Intelligent Transport Systems, 6(3), pp.282291.
- Considering the problem of automatic information acquisition in the field of intelligent transportation system (ITS), a new approach for detection of road traffic sign from natural scene images is proposed in this study.

- The adaptive colour segmentation based on pixel vector is firstly used to segment colour image into binary image and stand out traffic sign regions, which can reduce the influence of lighting conditions on image segmentation.
- Secondly, to improve the ability of shape identification during traffic sign detection, central projection transformation (CPT) is used to compute shape feature vectors of different candidate regions, and this shape feature is input to the probabilistic neural networks (PNN) to discriminate true traffic signs from candidates. The proposed approach is applied to many natural images.

➢ Bazzi, A., Zanella, A., Masini, B.M. and Pasolini, G., 2014, June. A distributed algorithm for virtual traffic lights with IEEE 802.11 p. In Networks and Communications (EuCNC), 2014 European Conference on (pp. 1-5). IEEE.

- Recently, 3GPP has added new features to long term evolution (LTE) that allow vehicles to communicate directly with each other and with surrounding objects. These short-range communications will play a key role in the so-called cellular vehicle-to-anything (C-V2X). As a particular service, which will be the basis of most applications for automated and connected cars, each vehicle will periodically broadcast information on its identity, status, and movements through short-range vehicle-to-vehicle (V2V) communications.
- Given the importance of this service, great attention has been given to the associated resource allocation procedures and the number of vehicles that can be simultaneously communicating in the network. However, little attention has been paid to handling messages of different sizes, which is actually foreseen in practice in order to transmit static information with a reduced periodicity. With this in mind, we evaluate the number of vehicles that can be allocated when parameters are optimized for messages of different sizes.
- This objective is pursued taking into account the numerous constraints imposed by the standard, through the formalization and solution of a combinatorial optimization problem. Example results, based on 3GPP indications, show that, with respect to the optimization of the allocation based on the larger packets, it is possible to obtain an

increase that varies between 10% and 30%, depending on the conditions, with peaks above 150% in specific cases.

- Yapp, J. and Kornecki, A.J., 2015, August. Safety analysis of virtual traffic lights. In Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on (pp. 505-510). IEEE.
- Traffic congestion is a daunting problem that is affecting the daily lives of billions of people across the world. Recently, a promising new traffic control scheme known as Virtual Traffic Lights (VTL) has been proposed for mitigating traffic congestion. VTL is an infrastructure free traffic control scheme that leverages the presence of Vehicle-to-Vehicle (V2V) communications.
- Such infrastructure free scheme has several benefits, such as alleviating traffic congestion; reducing the large cost of traffic lights and traffic control systems; reducing carbon emission, etc. This paper reports a DSRC-based prototype design effort on VTL using Dedicated Short-Range Communications (DSRC) technology.
- The experiments performed show the feasibility of implementing VTL with DSRC technology. Preliminary results of the field tests conducted in Pittsburgh with vehicles using VTL equipment indicate that VTL is capable of coordinating traffic at intersections and reducing the commute time of people.

- Yang, Bo, Rencheng Zheng, Keisuke Shimono, Tsutomu Kaizuka, and Kimihiko Nakano. "Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalized intersections." IET Intelligent Transport Systems 11, no. 2 (2017)
- Ground traffic lights are essential for maintaining traffic efficiency and safety at intersections. However, unsignalized intersections are still frequent in actual traffic environments. With the development of new forms of vehicular communication, in-vehicle traffic lights can assist drivers at unsignalized intersections.
- The authors proposed two types of in-vehicle traffic lights to assist drivers; these corresponded to two-way and all-way stop-controlled intersections. They adopted gap acceptance theory and a first-come-first-served strategy to determine passing priority for the two types of intersections, respectively. They then conducted a driving simulator

experiment involving 23 participants, to investigate driver behaviours elicited by the proposed system.

- They prepared four experimental conditions with combinations of in-vehicle traffic lights and auditory warnings. The authors' experimental results indicated that in-vehicle traffic lights were associated with significantly longer post-encroachment times and a significantly shorter maximum brake stroke. In terms of eye-gaze behaviours, the percentage of gaze concentration to the road centre area and mean glance durations were deemed acceptable for the avoidance of visual distraction, when in-vehicle traffic lights were presented via a head-up display. Therefore, their analysis of driver behaviours indicates that in-vehicle traffic lights can effectively provide driver assistance at unsignalized intersections.

➢ Higaki, H., 2014, March. Virtual Traffic Signals by Cooperation among Vehicle-Mounted Mobile Computers. In New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on (pp. 1-6). IEEE.

- A traffic signal at an intersection avoids collisions among vehicles whose lines of flow have a point of intersection. Though it provides a safe driving environment, it requires high configuration and maintenance cost and overhead.
- In order for their avoidance and/or reduction, this paper proposes a novel method for virtual traffic signals realized by cooperation among vehicle-mounted mobile wireless computers. Here, the red signal information is initiated based on mobility information, i.e., velocity, acceleration and location, of vehicles approaching an intersection.
- These vehicle- mounted computers cooperate by adhoc IVC (Inter-Vehicle Communication) for initiation, distribution and update of the red signal information. This paper shows a required wireless transmission range of the vehicle-mounted computers and an ad-hoc wireless communication protocol among them for the virtual traffic signals.

CHAPTER 3

METHODOLOGY

In this project, the basic working of model can be understood by using simple block diagrams. It is divided into 2 sections.

1.Traffic signal section

2.Vehicle section

3.1 BLOCK DIAGRAM:

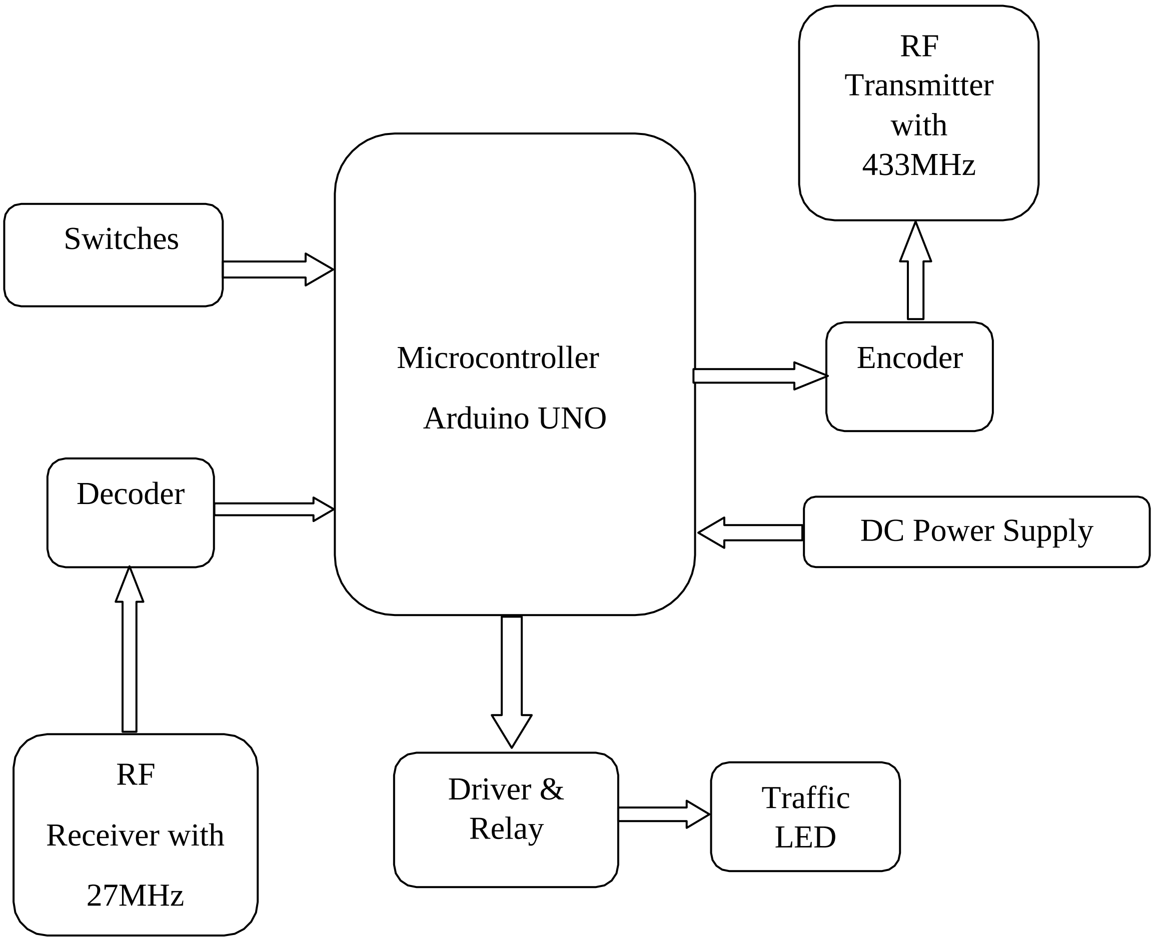

Fig:3.1 Traffic Signal Unit Block Diagram

To power up the circuit 12v power supply is used. Using manual switches traffic light signals are controlled and after that driver and relay circuit is used to switch the signals from green to red and vice versa, to process input and output the 28-pin microcontroller Arduino UNO is used. The data from the controller is encoded using the Encoder and then transmitter by the RF Transmitter. The traffic signal junction consists of RF Transmitter of 433 MHz for sending encoded status of the traffic light and also it consists of 27 MHz RF Receiver to receive the arrival signal from emergency vehicles. Traffic lights are used to indicate the traffic light signal.

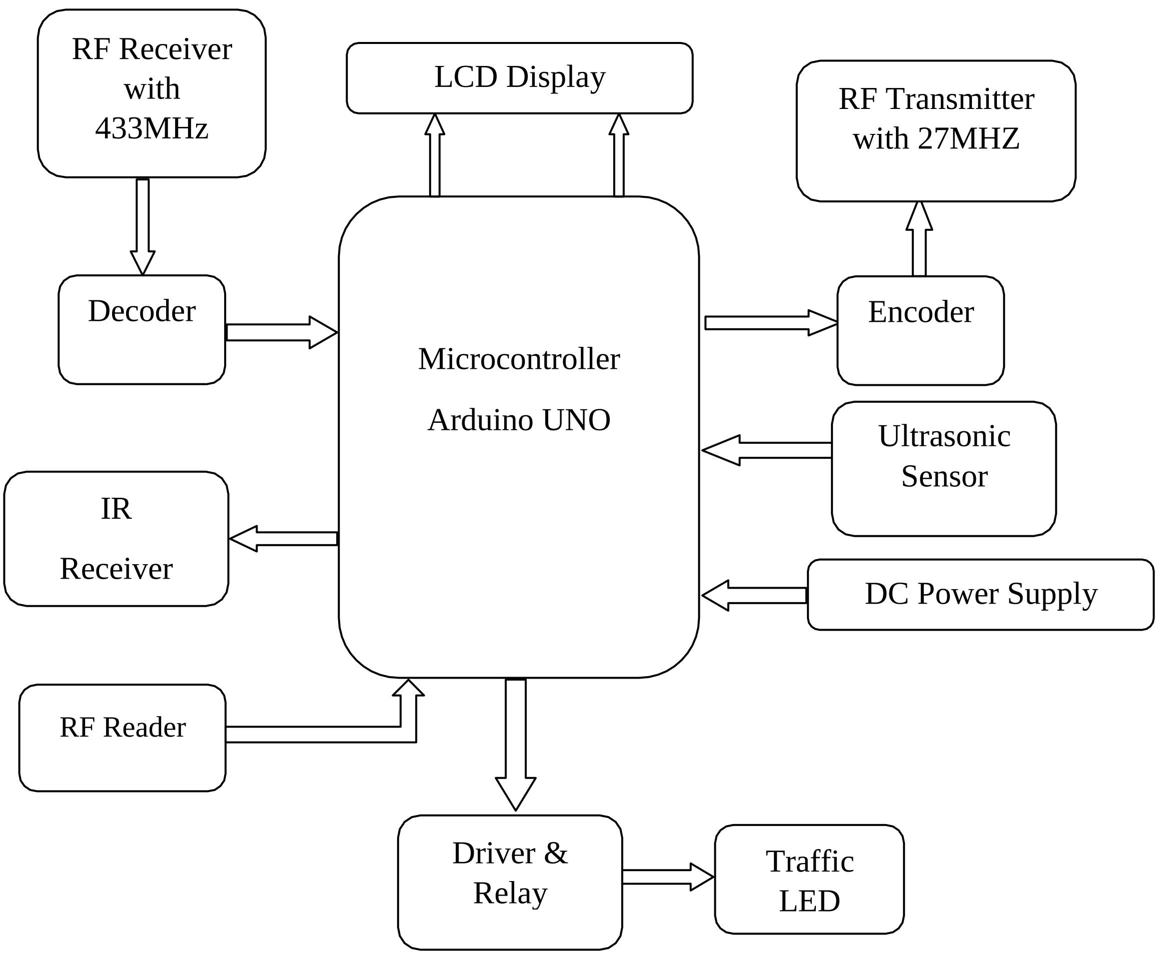

Fig:3.2 Vehicle Unit Block Diagram

RF Receiver receives the data that is being transmitted by the transmitter and is decoded by the decoder and sent to the microcontroller. RF Reader reads the code from the RF tag that will be placed in the road and sends the data to the controller. After processing, the status of the traffic signal will be displayed in the LCD display. The emergency vehicles consist of RF Transmitter of 27 MHz for sending the status of its arrival. Vehicle uses 433 MHz RF Receiver to receive the signal from traffic junctions. LED lights are used to indicate the traffic light signal. We are using different RF frequencies to avoid interference of signals. Ultrasonic sensor is used to indicate the obstacle and measure the distance from the obstacle.IR Receiver receives the signal from the IR transmitter whenever free left is activated.

3.2 WORKING PRINCIPLE:

The hardware components being used are Arduino UNO with microcontroller ATMEGA328 both in traffic post and in-vehicle, RF Transmitter and RF Receiver, IR Transmitter and Receiver, Ultrasonic sensor, displays used are Traffic LED both in traffic post and in-vehicle, LCD display in vehicle, and power supply. Relay switches are used to control LED. This RF module involves a RF Transmitter and a RF Receiver. The transmitter/collector (Tx/Rx) match works at a recurrence of 433 MHz Arduino is a solitary board microcontroller intended to make the application more open which are intuitive object and for autogreen signal for emergency vehicles we use 27MHz for communication.

The IR signal transmitted from the vehicle IR transmitter will be received by the IR receiver fixed on the road to identify the vehicle on that road. Up on reception of signal from the IR receiver the microcontroller sends the status of the traffic lights using RF transmitter. The switches in the traffic signal is used to change the traffic lights.

The RF receiver fitted in the vehicle will receive the signal from the RF transmitter of traffic signal. Up on reception of signal the RF receiver will give the signal to the microcontroller. The microcontroller processes the signal to identify the status of the traffic light. The status of the traffic light will be displayed on the driver dashboard with the help of

LED. The microcontroller uses the driver and relay circuit to control the LED lights to show the status of the traffic lights. The same will be displayed on the LCD display.

The ultrasonic sensor is used to detect obstacles in front of the vehicle. The output of the ultrasonic sensor will be given to the microcontroller. Upon reception of signal from the ultrasonic sensor, the microcontroller calculates the distance to the obstacle and display the same on the LCD display. The DC regulated power supply is used to power the entire system.

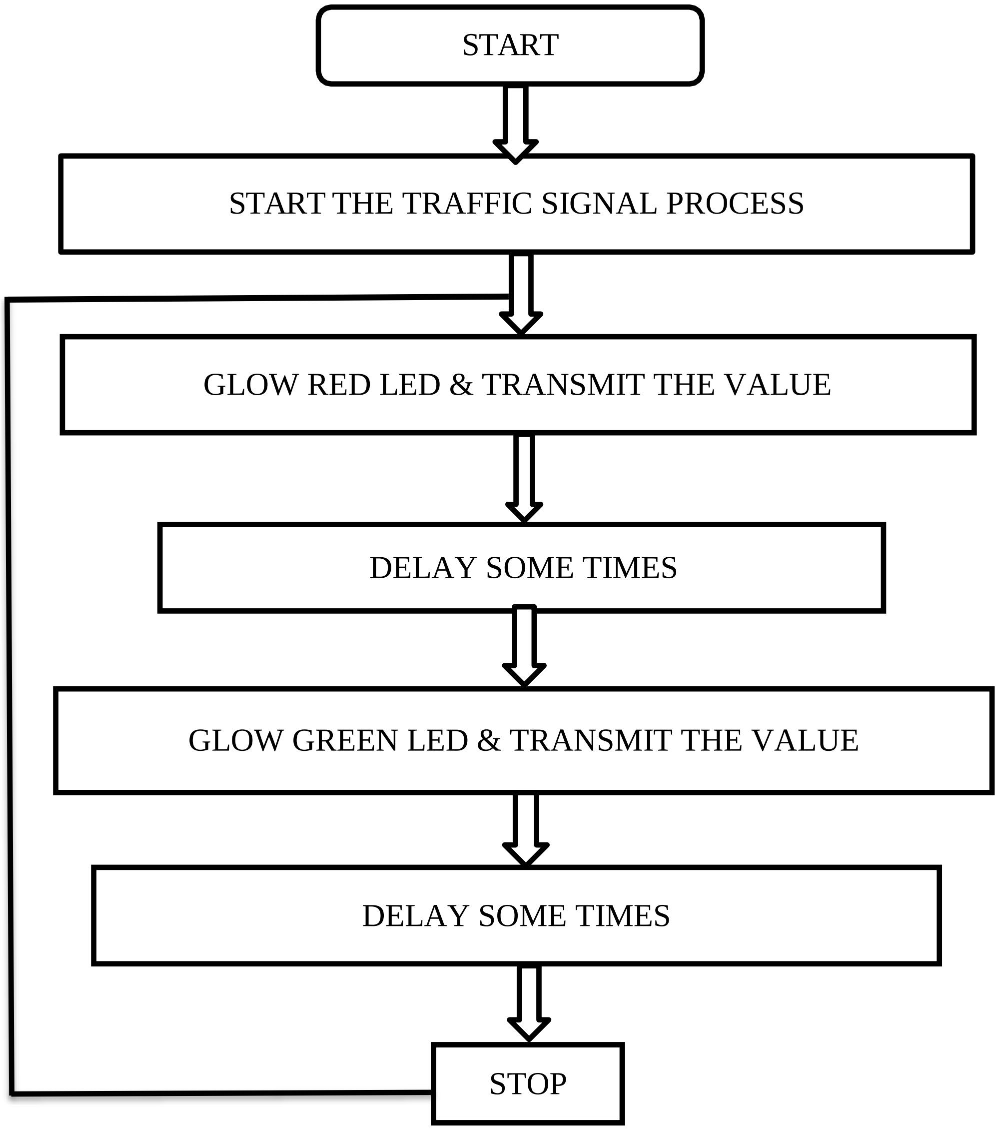

Fig: 3.3 Flow Chart of Transmitter

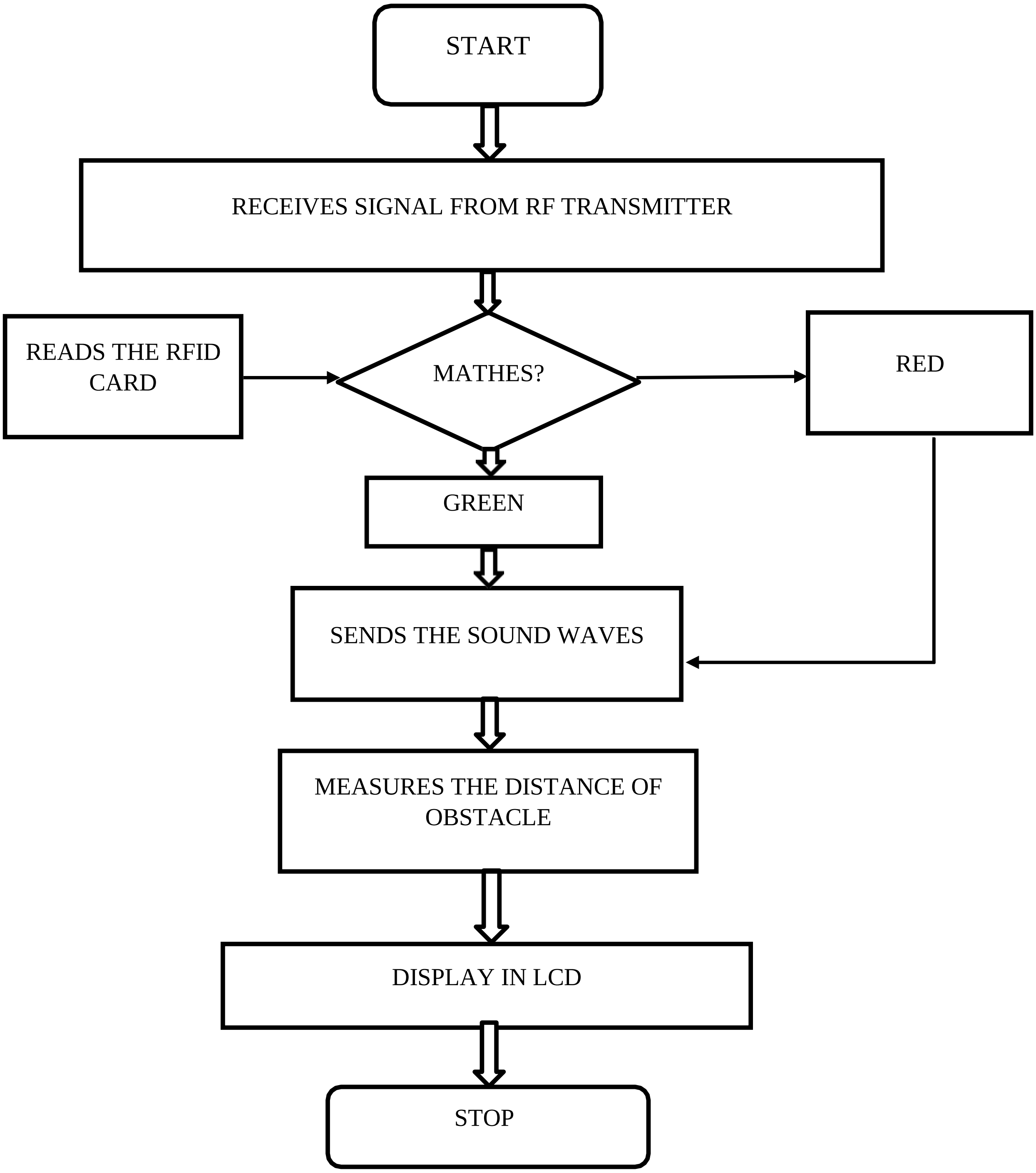

Fig:3.4 Flow Chart of Receiver

As shown in fig:3.3 when the transmission of signal starts, Red signal is transmitted for a while and then the green signal or it can be vice versa too. This controlling of the signal is done manually using four switches. Functions of the switches are as follows:

1. SWITCH1- GREEN SIGNAL FOE ROAD1
2. SWITCH2- GREEN SIGNAL FOR ROAD2
3. SWITCH3- GREEN SIGNAL FOR ROAD3
4. SWITCH4- GREEN SIGNAL FOR ROAD4
5. SWITCH5- SIGNAL FOR FREE LEFT.

The receiver receives the data from the RF transmitter and send it to controller RF card reader reads the data and sends it to the controller. The controller sends compares both the data. If both the data matches then green signal will be indicated or else red signal will be indicated.

If free left switch is activated then the IR transmitter will transmit the signal and the IR receiver on the vehicle receives the signal and the free left is indicated in the LCD display in the vehicle. Ultrasonic sensor continuously monitors to detect the obstacle in front of the vehicle and measures the distance from the obstacle.

CHAPTER 4

HARDWARE REQUIREMENT

In this project, The model consists some Hardware requirements as in following

1.Microcontroller-Arduino UNO

2.Arduino IDE

3.IR Transmitter and Receiver

4.16x2 LCD Display

5.Power Supply unit

6.Relay Driver Circuit using IC ULN2003

7.RFID Read and Write Module

8.Driver IC ULN2003

9.Ultrasonic Sensor

4.1 Microcontroller – Arduino UNO

The Arduino Uno R3 is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

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

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller ATmega328

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Memory

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

Pin Diagram

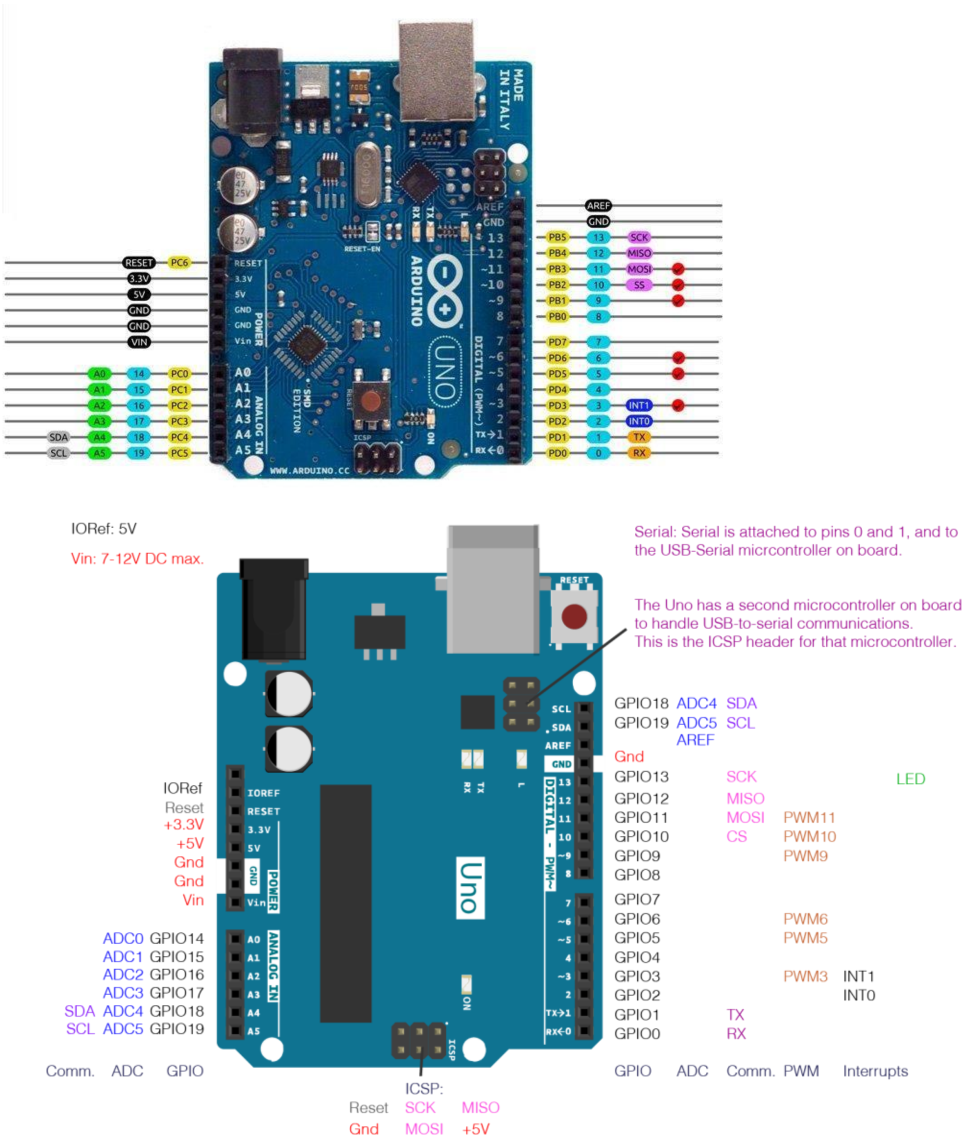

Fig:4.1 Pin Diagram of Arduino UNO

Input and Output

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

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

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

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

There are a couple of other pins on the board:

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

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes reburned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.It can be used as Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than physical press of the reset button before an upload, the Arduino Uno is designed in the way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 Nano farad capacitor. When this line is asserted, the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line; see this forum thread for details.

USB Over current Protection

The Arduino Uno has a resettable poly fuse that protects your computer's USB ports from shorts and over current. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Fig:4.2 Arduino UNO

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy,[2] aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

Hardware

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -Arduino.

An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I^2C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the Opti boot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Board Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Decimal,[a] Demilune,[b] and current Uno[c] provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarding boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

4.2 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

Fig:4.3 Arduino IDE

Sketch

A program written with the Arduino IDE is called a sketch.[58] Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

A minimal Arduino C/C++ program consist of only two functions:

- **setup()**: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
- **loop()**: After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Libraries

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

WRITING SKETCHES

Programs written using Arduino Software (IDE) are called **sketches.** These sketches are written in the text editor and are saved with the file extension. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

- Verifies and checks for errors.
- Uploads, compiles and uploads it to the board.
- Creates a new sketch.

- Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.
- Saves your sketch.
- Opens the serial monitor.

Additional commands are found within the five menus: **File, Edit, Sketch, Tools and Help.** The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

FILE

- **New:** Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- **Open:** Allows loading a sketch file browsing through the computer drives and folders. **Open Recent:** Provides a short list of the most recent sketches, ready to be opened.
- **Sketchbook:** Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- **Examples:** Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- **Close:** Closes the instance of the Arduino Software from which it is clicked.
- **Save:** Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as".
- **Save as:** Allows to save the current sketch with a different name
- **Page Setup:** It shows the Page Setup window for printing.
- **Print:** Sends the current sketch to the printer according to the settings defined in Page Setup.
- **Preferences:** Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- **Quit:** Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

EDIT

- **Undo/Redo:** Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- **Cut:** Removes the selected text from the editor and places it into the clipboard.
- **Copy:** Duplicates the selected text in the editor and places it into the clipboard.
- **Copy for Forum:** Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- **Copy as HTML:** Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- **Paste:** Puts the contents of the clipboard at the cursor position, in the editor.
- **Select All:** Selects and highlights the whole content of the editor.
- Comment/Uncomment: Puts or removes the // comment marker at the beginning of each selected line.
- **Increase/Decrease Indent:** Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- **Find:** Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- **Find Next:** Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- **Find Previous:** Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

SKETCH

- **Verify/Compile:** Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- **Upload:** Compiles and loads the binary file onto the configured board through the configured Port.
- **Upload Using Programmer:** This will overwrite the boot loader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your

sketch. Please note that this command will NOT burn the fuses. To do so a Tool ->Burn Bootloader command must be executed.

- **Export Compiled Binary:** Saves a .hex file that may be kept as archive or sent to the board using other tools.
- **Show Sketch Folder:** Opens the current sketch folder.
- **Include Library:** Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- **Add File:** Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

TOOLS

- **AutoFormat*:*** This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- **Archive Sketch:** Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- **Fix Encoding & Reload:** Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- **Serial Monitor:** Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- **Board*:*** Select the board that you're using. See below for descriptions of the various boards.
- **Port*:*** This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- **Programmer:** For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new microcontroller, you will use this.

- **Burn Bootloader:** The items in this menu allow you to burn a boot loader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new AT mega microcontroller (which normally come without a boot loader). Ensure to select the correct board from the **Boards** menu before burning the boot loader on the target board. This command also set the right fuses.

SKETCH BOOK

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the **File > Sketchbook** menu or from the **Open** button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the **Preference** dialog.

TABS, MULTIPLE FILES AND COMPILATION

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

LIBRARIES

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of your code.

THIRD-PARTY HARDWARE

Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, boot loaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory.

SERIAL MONITOR

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to **Serial.begin** in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

BOARDS

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn boot loader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the boot loader. You can find a comparison table between the various boards here.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows adding support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- **ArduinoYùn:** An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- **Arduino or GenuinoUno:** An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- **Arduino Diecimilaor Duemilanovew or*ATmega168*:** An ATmega168 running at 16 MHz with auto-reset.
- **Arduino Nanowor ATmega328P:** An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs.
- **Arduino or GenuinoMega2560:** An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- **Arduino Mega:** An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- **Arduino Mega ADK:** *An* ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

4.3 IR Transmitter and Receiver

An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which is used in many applications in electronics, like it is used in Remote control system, motion detector, Product counter, Line follower Robots, Alarms etc.

IR sensor basically consist an IR LED and a Photodiode, this pair is generally called IR pair or Photo coupler. IR sensor work on the principal in which IR LED emits IR radiation and Photodiode sense that IR radiation. Photodiode resistance changes according to the amount of IR radiation falling on it, hence the voltage drop across it also changes and by using the voltage comparator (like LM358) we can sense the voltage change and generate the output accordingly.

The placing of IR LED and Photodiode can be done in two ways: Direct and Indirect. In Direct incidence, IR LED and photodiode are kept in front of one another, so that IR radiation can directly falls on photodiode. If we place any object between them, then it stops the falling of IR light on photodiode.

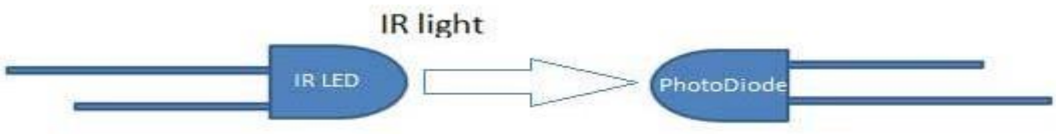

Fig:4.4 IR Tx and Rx

In Indirect Incidence, both the IR LED and Photo diode are placed in parallel (side by side), facing both in same direction. In that fashion, when a object is kept in front of IR pair, the IR light gets reflected by the object and gets absorbed by photodiode. Note that object shouldn't be black as it will absorb all the IR light, instead of reflect. Generally IR pair is placed in this fashion in IR sensor Module.

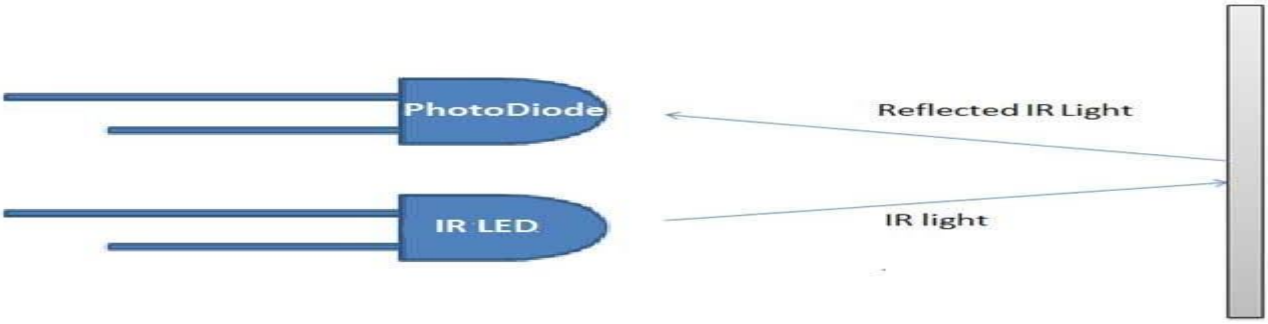

Fig:4.5 Working of IR Sensor

IR Sensor Circuit Diagram:

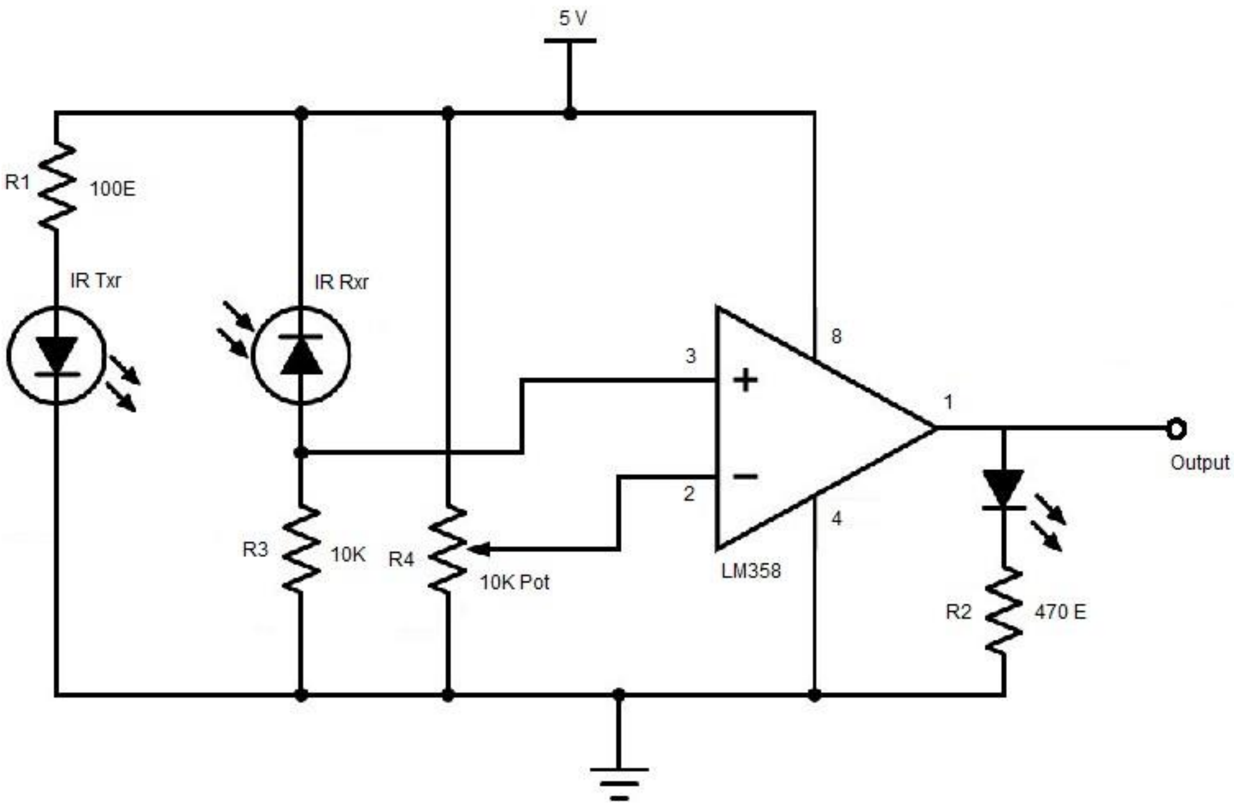

Fig:4.6 IR Sensor Circuit Diagram

- **IR pair (IR LED and Photodiode)**
- **IC LM358**
- **Resistor 100 ohm, 10k, 470 ohm**
- **Variable resistor – 10k**
- **LED**

Circuit Operation:

- You can see the connections in the IR sensor circuit diagram. Photo diode is connected in reverse bias, inverting end of LM358 (PIN 2) is connected to the variable resistor, to adjust the sensitivity of the sensor. And non-inverting end (PIN 3) is connected to the junction of photodiode and a resistor.
- When we turn ON the circuit there is no IR radiation towards photodiode and the Output of the comparator is LOW. When we take some object (not black) in front of IR pair, then IR emitted by IR LED is reflected by the object and absorbed by the photodiode. Now when reflected IR Falls on Photodiode, the voltage across photodiode drops, and the voltage across series resistor R2 increases. When the voltage at Resistor R2 (which is connected to the non-inverting end of comparator) gets higher than the voltage at inverting end, then the output becomes HIGH and LED turns ON.
- Voltage at inverting end, which is also called Threshold Voltage, can be set by rotating the variable resistor's knob. Higher the voltage at inverting end (-), less sensitive the sensor and Lower the voltage at inverting end (-), more sensitive the sensor.

IR LED

IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. Everything which produce heat, emits infrared like for example our human body. Infrared have the same properties as visible light, like it can be focused, reflected and polarised like visible light.

IR LED looks like a normal LED and also operates like a normal LED, it consumes 20mA current and 3vots power. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimetres to several feets, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometres.

Fig:4.7 IR LED

Photodiode

Photodiode is considered as Light dependent Resistor (LDR), means it has very High resistance in absence of light and become low when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection.

Photodiode looks like a LED, with a Black colour coating on its outer side. It is used in reversed biased, as showed in circuit diagram.

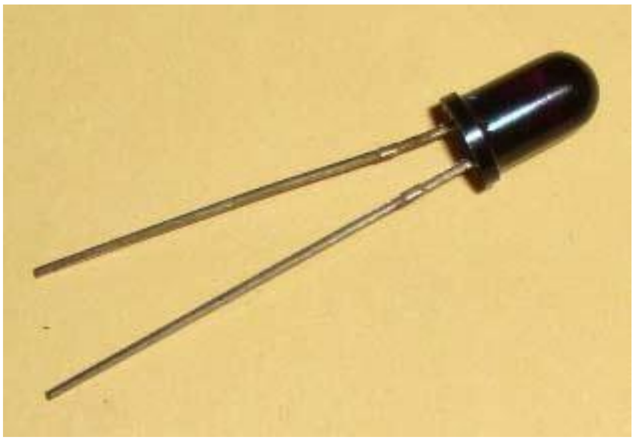

Fig:4.8 Photodiode

LM358

LM358 is an operational amplifier (Op-Amp) and in this circuit we are using it as a voltage comparator. The LM358 has two independent voltage comparators inside it, which can be powered by single PIN, so we can use the single IC to build two IR sensor modules. We have used only one comparator here, which have inputs at PIN 2 & 3 and output at PIN 1. Voltage comparator has two inputs, one is inverting input and second is non-inverting input (PIN 2 and 3 in LM358). When voltage at non-inverting input (+) is higher than the voltage at inverting input (-), then the output of comparator (PIN 1) is High. And if the voltage of inverting input (-) is Higher than non-inverting end (+), then output is LOW.

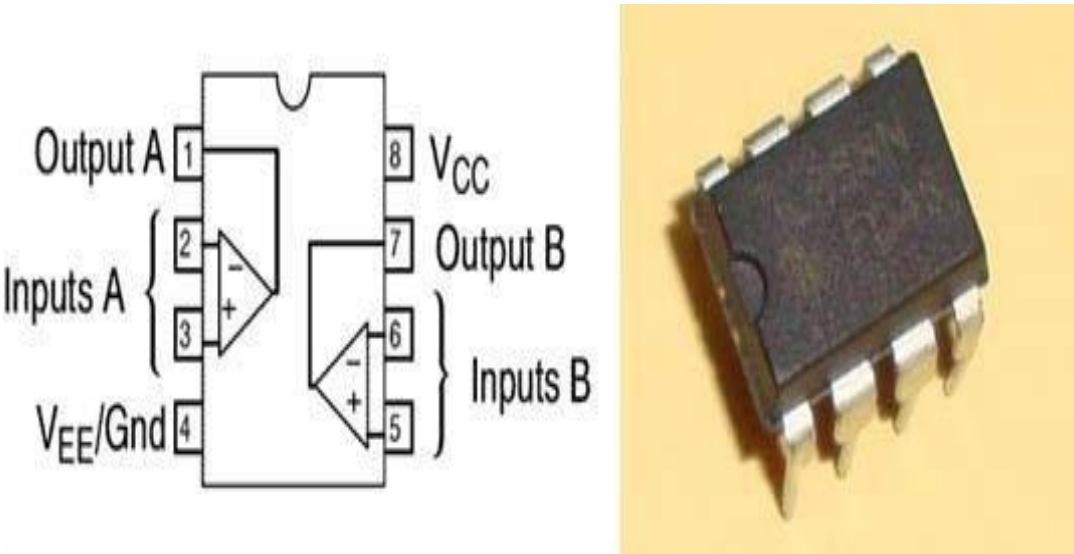

Fig:4.9 Pin Diagram of LM358

4.4 16×2 LCD Display

We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix.

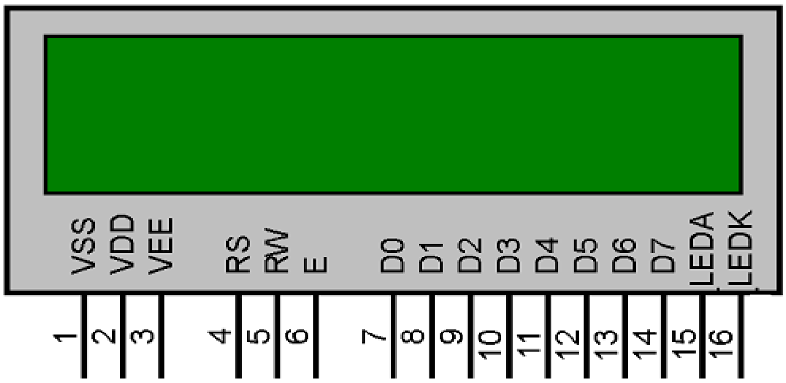

Fig:4.10 16X2 LCD Pinout Diagram

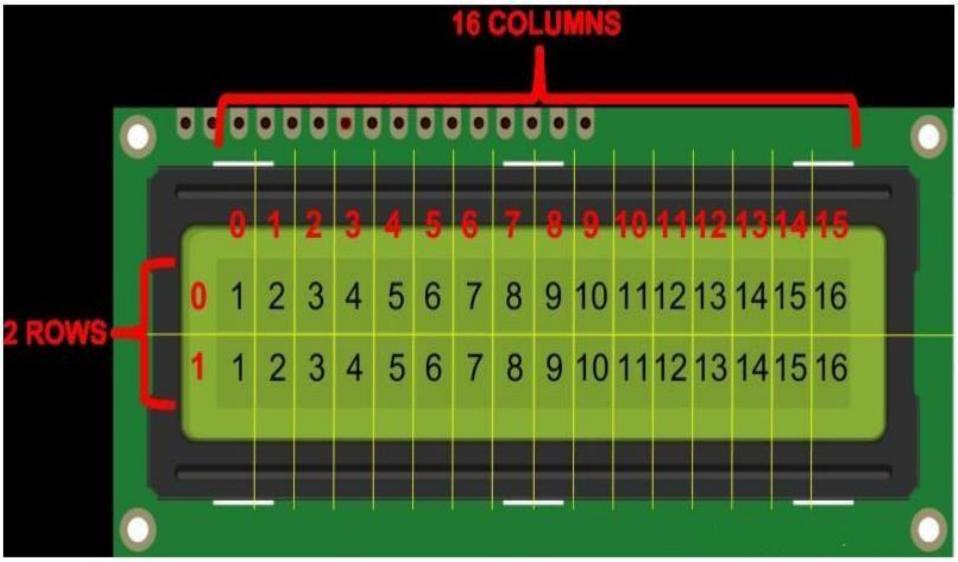

Fig:4.11 16X2 LCD Di

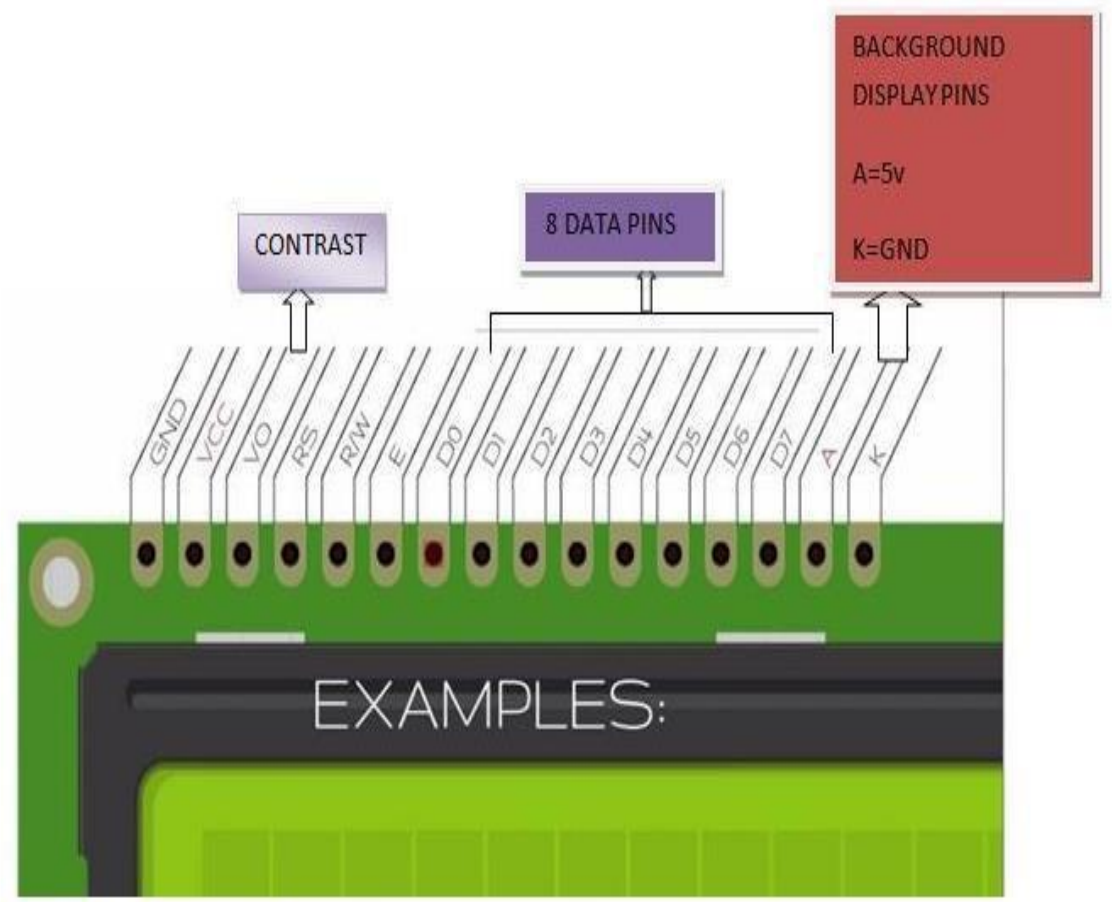

Fig:4.12 Pins used to interface with I2C

FUNCTION OF PIN:

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.

- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

Command Register: 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. Processing for commands happen in the command register.

Data Register: The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected.

Table: 4.1 Important command codes for LCD

Sl.No.	Hex Code	Command to LCD instruction Register
1	01	Clear display screen
2	02	Return home
3	04	Decrement cursor (shift cursor to left)
4	06	Increment cursor (shift cursor to right)
5	05	Shift display right
6	07	Shift display left
7	08	Display off, cursor off
8	0A	Display off, cursor on
9	0C	Display on, cursor off
10	0E	Display on, cursor blinking
11	0F	Display on, cursor blinking
12	10	Shift cursor position to left
13	14	Shift cursor position to right

14	18	Shift the entire display to the left
15	1C	Shift the entire display to the right
16	80	Force cursor to beginning (1st line)
17	C0	Force cursor to beginning (2nd line)
18	38	2 lines and 5×7 matrix

Generating custom characters on LCD is not very hard. It requires the knowledge about custom generated random-access memory (CG-RAM) of LCD and the LCD chip controller. Most LCDs contain Hitachi HD4478 controller. CG-RAM is the main component in making custom characters. It stores the custom characters once declared in the code. CG-RAM size is 64 bytes providing the option of creating eight characters at a time. Each character is eight bytes in size.

CG-RAM address starts from 0x40(Hexadecimal) or 64 in decimal. We can generate custom characters at these addresses. Once we generate our characters at these addresses, now we can print them on the LCD at any time by just sending simple commands to the LCD. Character addresses and printing commands are below.

In the table below, you can see starting addresses for each character with their printing commands. The first character is generated at address 0x40 to 0x47 and is printed on LCD by just sending simple command 0 to the LCD. The second character is generated at address 0x48 to 0x55 and is printed by sending 1 to LCD.

Table: 4.2 Starting Addresses for each Character with Their Printing Commands

CG-RAM CHARACTERS	CG-RAM Address (Hexadecimal)	Commands to display Generated character
1st Character	0x40	0
2nd Character	0x48	1
3rd Character	0x56	2
4th Character	0x64	3
5th Character	0x72	4
6th Character	0x80	5
7th Character	0x88	6
8th Character	0x96	7

4.5 POWER SUPPLY UNIT

The circuit needs two different voltages, +5V & +12V, to work. These dual voltages are supplied by this specially designed power supply.

The power supply, unsung hero of every electronic circuit, plays very important role in smooth running of the connected circuit. The main object of this 'power supply' is, as the name itself implies, to deliver the required amount of stabilized and pure power to the circuit. Every typical power supply contains the following sections:

1. Step-down Transformer: The conventional supply, which is generally available to the user, is 230V AC. It is necessary to step down the mains supply to the desired level. This is achieved by using suitably rated step-down transformer. While designing the power supply, it is necessary to go for little higher rating transformer than the required one. The reason for this is for proper working of the regulator IC (say KIA 7805) it needs at least 2.5V more than the expected output voltage.

2. Rectifier stage*:* Then the step-downed Alternating Current is converted into Direct Current. This rectification is achieved by using passive components such as diodes. If the power supply is designed for low voltage/current drawing loads/circuits (say +5V), it is sufficient to employ full-wave rectifier with centre-tap transformer as a power source. While choosing the diodes the PIV rating is taken into consideration.

3. Filter stage*:* But this rectified output contains some percentage of superimposed AC ripples. So to filter these AC components filter stage is built around the rectifier stage. The cheap, reliable, simple and effective filtering for low current drawing loads (say up to 50 mA) is done by using shunt capacitors. This electrolytic capacitor has polarities, take care while connecting the circuit.

4. Voltage Regulation*:* The filtered DC output is not stable. It varies in accordance with the fluctuations in mains supply or varying load current. This variation of load current is observed due to voltage drop in transformer windings, rectifier and filter circuit. These variations in DC output voltage may cause inaccurate or erratic operation or even malfunctioning of many electronic circuits. For example, the circuit boards which are implanted by CMOS or TTL ICs.

The stabilization of DC output is achieved by using the three terminal voltage regulator IC. This regulator IC comes in two flavors: 78xx for positive voltage output and 79xx for negative voltage output. For example 7805 gives +5V output and 7905 gives -5V stabilized output. These regulator ICs have in-built short-circuit protection and auto-thermal cutout provisions. If the load current is very high the IC needs 'heat sink' to dissipate the internally generated power.

Circuit Description:

A DC power supply which maintains the output voltage constant irrespective of AC mains fluctuations or load variations is known as **regulated DC power supply.** It is also referred as full-wave regulated power supply as it uses four diodes in bridge fashion with the transformer. This laboratory power supply offers excellent line and load regulation and output voltages of +5V & +12 V at output currents up to one amp.

KIA 78xx

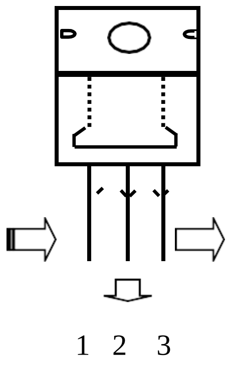

Fig:4.13 IC 78XX

CIRCUIT DIAGRAM OF +5V & +12V BRIDGE RECTIFIER REGULATED POWER

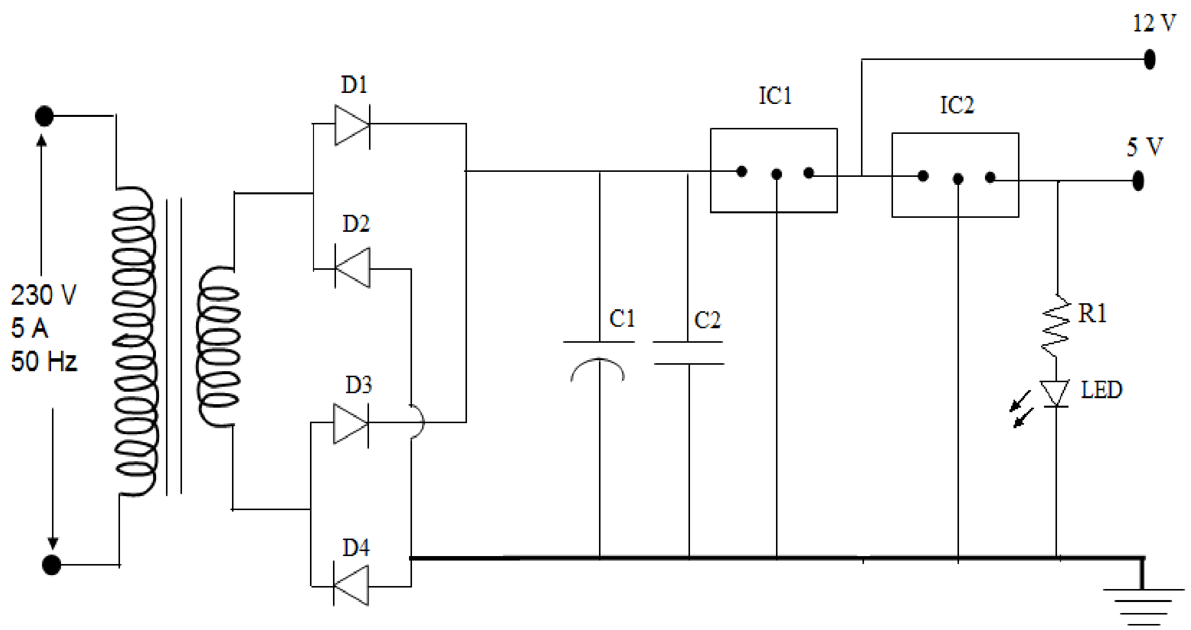

Fig:4.14 Circuit Diagram of +5V & +12V Bridge Rectifier Regulated Power

Parts List:

SEMICONDUCTORS		Total no.
IC1 IC2	7812 Regulator IC 7805 Regulator IC	1 1
D1, D2, D3 & D4	1N4007 Rectifier Diodes	4
CAPACITORS		
C1	1000 µf/25V Electrolytic	1
C2	0.1µF Ceramic Disc type	1
RESISTORS		
R1	470 Ohm	1

1. Step-down Transformer: The transformer rating is 230V AC at Primary and 0 - 12V, 1Ampers across secondary winding. This transformer has a capability to deliver a current of 1Ampere, which is more than enough to drive any electronic circuit or varying load. The 12VAC appearing across the secondary is the RMS value of the waveform and the peak value would be 12 x 1.414 = 16.8 volts. This value limits our choice of rectifier diode as 1N4007, which is having PIV rating more than 16Volts.

2. Rectifier Stage: The two diodes D1, D2, D3 & D4 are connected across the secondary winding of the transformer as a bridge rectifier. During the positive half-cycle of secondary voltage, the end A of the secondary winding becomes positive and end B negative. This makes the diodes D1 & D3 forward biased and diodes D2 & D4 reverse biased. Therefore diodes D1 & D3 conducts while diodes D2 & D4 does not. During the negative half-cycle, end A of the secondary winding becomes negative and end B positive. Therefore diodes D2 & D4 conducts while diodes D1 & D3 does not. Therefore, pulsating DC is obtained at the output of diodes D1 & D3 and ground is obtained at the output of diodes D2 & D4.

3. Filter Stage: Here Capacitor C1 is used for filtering purpose and connected across the rectifier output. It filters the AC components present in the rectified DC and gives steady DC voltage. As the rectifier voltage increases, it charges the capacitor and also supplies current to the load. When capacitor is charged to the peak value of the rectifier voltage, rectifier voltage starts to decrease. As the next voltage peak immediately recharges the capacitor, the discharge period is of very small duration. Due to this continuous charge-discharge-recharge cycle very little ripple is observed in the filtered output. Moreover, output voltage is higher as it remains substantially near the peak value of rectifier output voltage. This phenomenon is also explained in other form as: the shunt capacitor offers a low reactance path to the AC components of current and open circuit to DC component. During positive half cycle the capacitor stores energy in the form of electrostatic field. During negative half cycle, the filter capacitor releases stored energy to the load.

4. Voltage Regulation Stage: Across the point 'D' and Ground there is rectified and filtered DC In the present circuit KIA 7812 three terminal voltage regulator IC is used to get +12V and KIA 7805 voltage regulator IC is used to get +5V regulated DC output. In the three terminals, pin 1 is input i.e., rectified & filtered DC is connected to this pin. Pin 2 is common pin and is grounded. The pin 3 gives the stabilized DC output to the load. The circuit shows two more decoupling capacitors C2 & C3, which provides ground path to the high frequency noise signals. Across the point 'E' and 'F' with respect to ground +5V & +12V stabilized or regulated DC output is measured, which can be connected to the required circuit.

Note: While connecting the diodes and electrolytic capacitors the polarities must be taken into consideration. The transformer's primary winding deals with 230V mains, care should be taken with it.

4.6 Relay Driver Circuit using IC ULN2003

In general, while designing electronics projects the loads are controlled (switched ON or OFF) using microcontroller block. But, for this purpose the circuit requires relays, acting as controlled switches (for different circuits different types of relays are used). Depending on the signals received from the microcontroller or other control circuits the relay controls the load. The relay consists of continuous power supply and whenever it gets driven or gets control signal then the relay gets activated and loads can be turned ON or OFF. But, primarily we must know what is a relay driver circuit.

Relay Driver Circuit

The circuit used for driving a relay can be termed as a relay driver circuit and it can be designed using various integrated circuits. These relays are needed to be driven for activating or to turn ON. So, relays require some driver circuitry to turn ON or OFF (based on the requirement). The relay driver circuit can be realized using different integrated circuits such as ULN2003, CS1107, MAX4896, FAN3240, A2550, and so on. This is about relay driver circuit using ULN2003.

ULN2003 Driver IC

The IC ULN2003A is a Darlington transistor array which deals with high-voltage and high-current. There are various types of relay driver ICs such as a high side toggle switch, low side toggle switch, bipolar NPN transistor, Darlington transistor, N-channel MOSFET, ULN2003 driver IC.

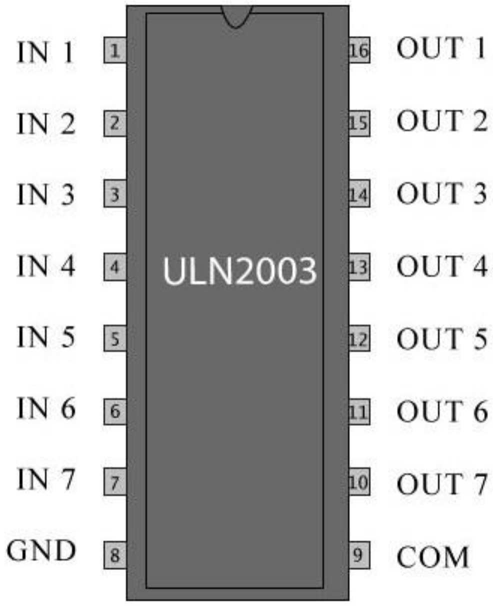

Fig:4.15 pin diagram of IC ULN2003A

The pin diagram of IC ULN2003A is shown in the above figure which consists of 16 pins. The IC ULN2003A comprises of 7-NPN Darlington pairs as shown in the internal schematic diagram and is typically used to switch inductive loads (dissipates voltage spikes if any using suppression diode) and to drive stepper motors.

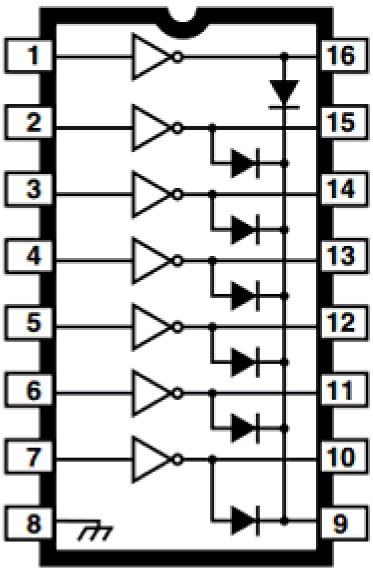

Fig:4.16 ULN2003 Internal Schematic Diagram

It is difficult to use a number of relays with transistors, so, relay driver IC ULN2003A can be used for availing more relays. We can use seven relays with relay driver circuit using ULN2003.

Relay

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal.

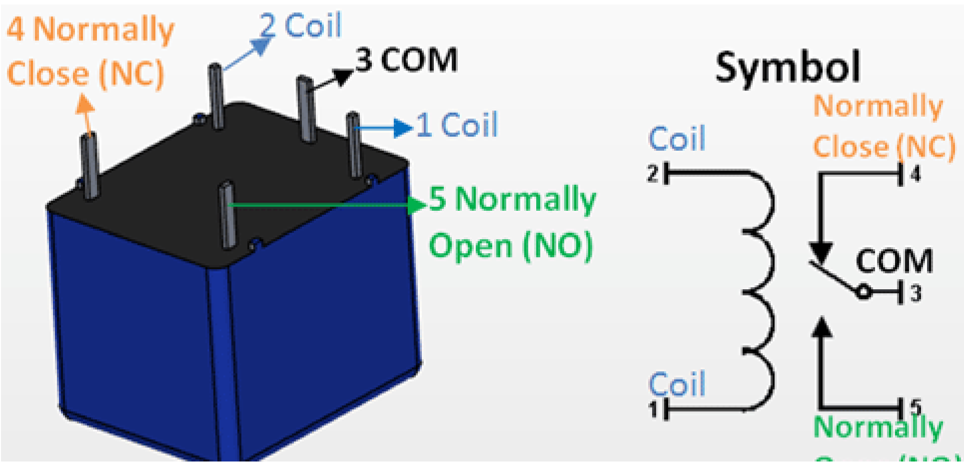

Fig:4.17 Relay Driver

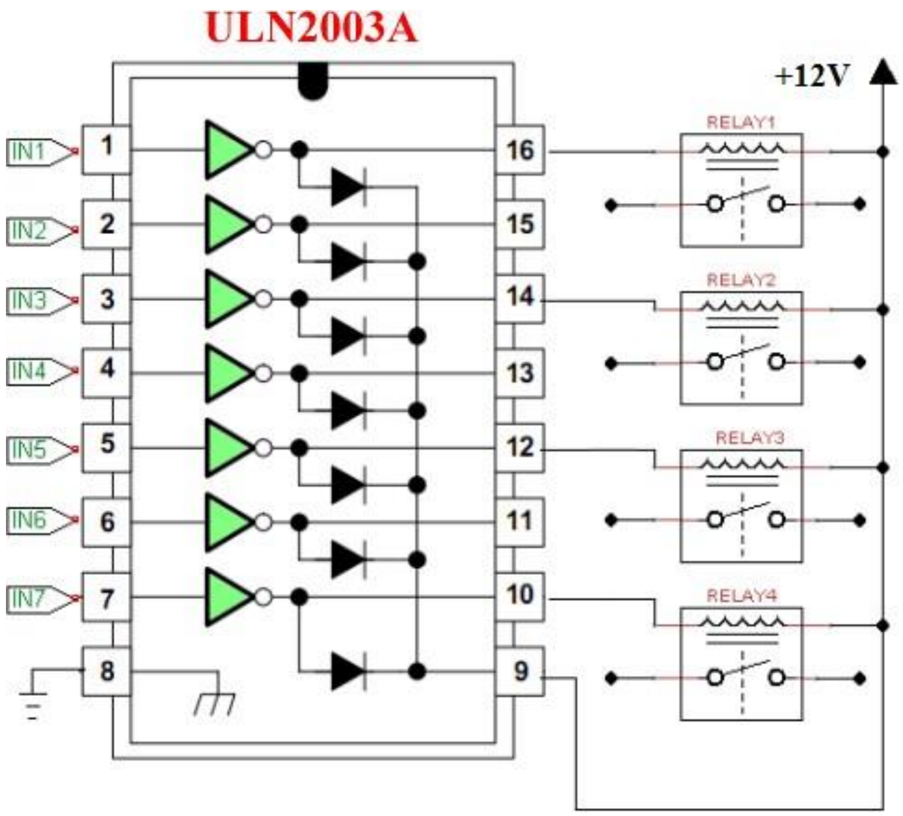

Fig:4.18 Interfacing of 4 Relay Driver with ULN2003A

4.7 RFID Read and Write Module

- MIFARE Series Proximity RFID Read/Write Module has an integrated/external antenna in minimized form factor. It is designed to work on the industry-standard carrier frequency of 13.56MHz. This feature packed module is best suited for Access Control, Time and Attendance, Handheld smart reader and other RFID enabled applications. An integrated antenna in minimized form factor in the module facilitates communication between the MIFARE Read/Write module and the Read/Write transponders. To achieve better Read/Write functionality this module is provided with a pin out for the optional external antenna. The Read/Write module identifies 13.56

MHz range Read/Write type transponders (Tags) in a contact-less mode via Radio Frequency (RF).

- MIFARE Read/Write module communicates with the host system via the wired interface in a protocol of available choice that is selected from the module pin out to transmit the data received from the transponder. The baseband processor decodes the data received from the front end before transmitting it to the host system.
- MIFARE Read/Write module is completely sealed modules in minimized dimensions. The electronics are placed inside a plastic housing that is sealed with a potting agent for mechanical and electrical stability.
- MIFARE RC522 is the high integrated RFID card reader which works on non-contact 13.56mhz communication, is designed by NXP as a low power consumption, low cost and compact size read and write chip, is the best choice in the development of smart meters and portable hand-held devices.
- MF RC522 use the advanced modulation system, fully integrated at 13.56MHz with all kinds of positive non-contact communication protocols. This module can fit directly in hand held devices for mass production. Module use 3.3V power supply, and can communicate directly with any CPU board by connecting through SPI protocol, which ensure reliable work, good reading distance.

Specifications

Voltage	DC 3.3V (Do not use 5V supply)
Operating Current	13-26mA
Operating Frequency	13.56MHz
Module Interface SPI Data Transfer Rate	Max. 10Mbit/s
Card reading distance	0～30mm （Mifare1 card）

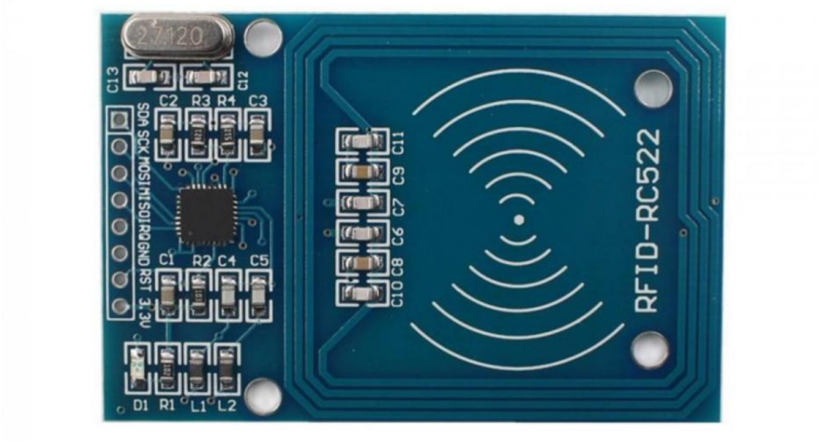

Fig:4.19 RFID-RC522

4.8 433MHz and 27MHz RF Transmitter and Receiver

A wireless radio frequency (RF) transmitter and receiver can be easily made using HT12D Decoder, HT12E Encoder and ASK RF Module. Wireless transmission can be done by using 433Mhz/27MHz ASK RF Transmitter and Receiver modules. In these modules digital data is represented by different amplitudes of the carrier wave, hence this modulation is known as Amplitude Shift Keying (ASK). Radio Frequency (RF) transmission is more strong and reliable than Infrared (IR) transmission due to following reasons:

- Radio Frequency signals can travel longer distances than Infrared.
- Only line of sight communication is possible through Infrared while radio frequency signals can be transmitted even when there is obstacle.
- Infrared signals will get interfered by other IR sources but signals on one frequency band in RF will not interfered by other frequency RF signals.

This circuit utilizes the RF module (Tx/Rx) for making a wireless communication, which could be used to drive an output from a distant place. RF module, as the name suggests, uses radio frequency to send signals. These signals are transmitted at a particular frequency and a baud rate. A receiver can receive these signals only if it is configured for that frequency.

A four channel encoder/decoder pair has also been used in this system. The input signals, at the transmitter side, are taken through four switches.

This radio frequency (RF) transmission system employs Amplitude Shift Keying (ASK) with transmitter/receiver (Tx/Rx) pair operating at 434 MHz. The transmitter module takes serial input and transmits these signals through RF. The transmitted signals are received by the receiver module placed away from the source of transmission. The system allows one way communication between two nodes, namely, transmission and reception. The RF module has been used in conjunction with a set of four channel encoder/decoder ICs. Here HT12E & HT12D have been used as encoder and decoder respectively. The encoder converts the parallel inputs (from the remote switches) into serial set of signals. These signals are serially transferred through RF to the reception point. The decoder is used after the RF receiver to decode the serial format and retrieve the original signals as outputs.

Transmitter Circuit

HT12E Encoder IC will convert the 4 bit parallel data given to pins D0 – D3 to serial data and will be available at DOUT. This output serial data is given to ASK RF Transmitter. Address inputs A0 – A7 can be used to provide data security and can be connected to GND (Logic ZERO) or left open (Logic ONE). Status of these Address pins should match with status of address pins in the receiver for the transmission of the data. Data will be transmitted only when the Transmit Enable pin (TE) is LOW. 1.1MΩ resistor will provide the necessary external resistance for the operation of the internal oscillator of HT12E.

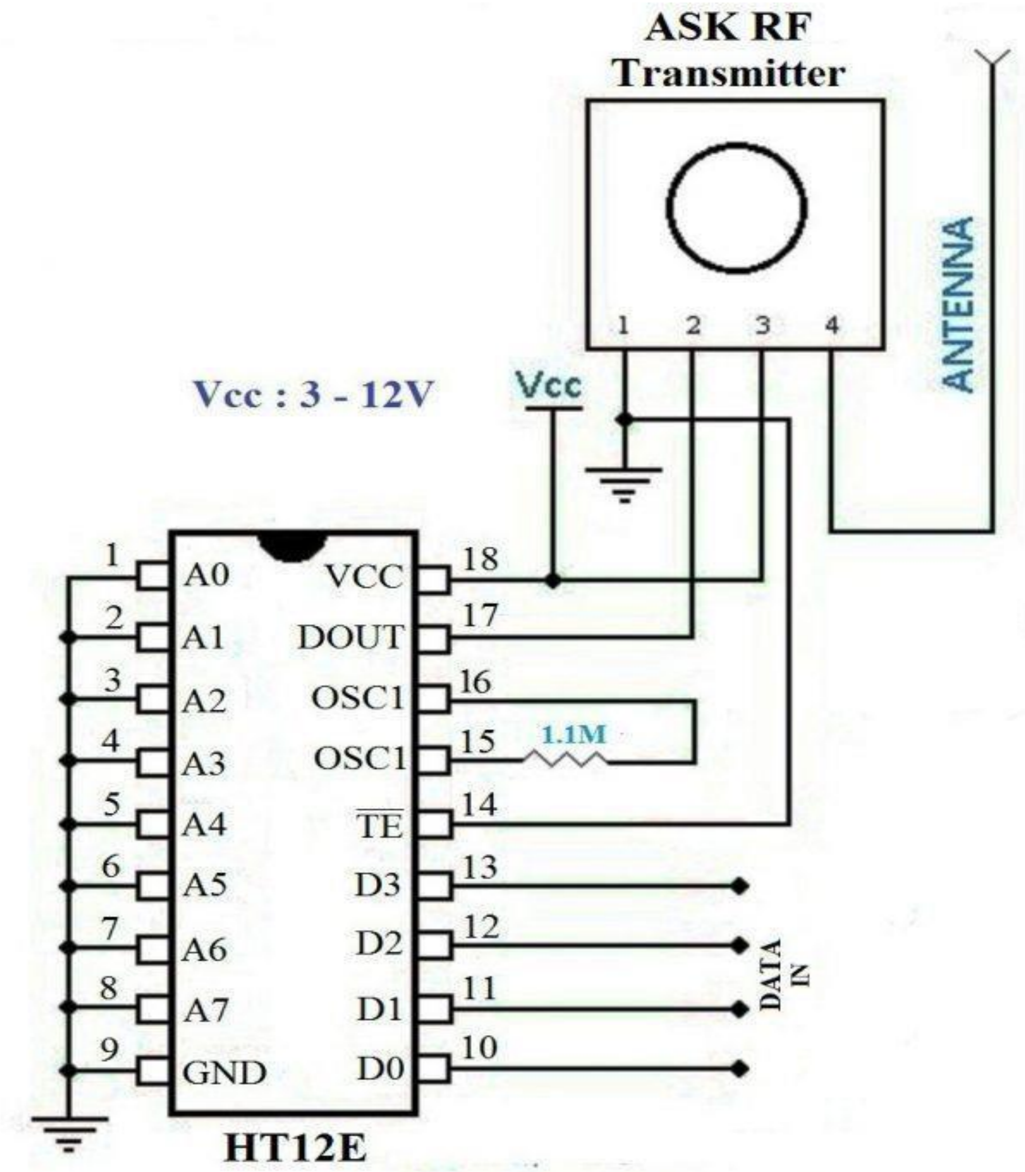

Fig:4.20 Transmitter Circuit

Receiver Circuit

ASK RF Receiver receives the data transmitted using ASK RF Transmitter. HT12D decoder will convert the received serial data to 4 bit parallel data D0 – D3. The status of these address pins A0-A7 should match with status of address pin in the HT12E at the transmitter for the transmission of data. The LED connected to the above circuit glows when valid data transmission occurs from transmitter to receiver. 51KΩ resistor will provide the necessary resistance required for the internal oscillator of the HT12D.

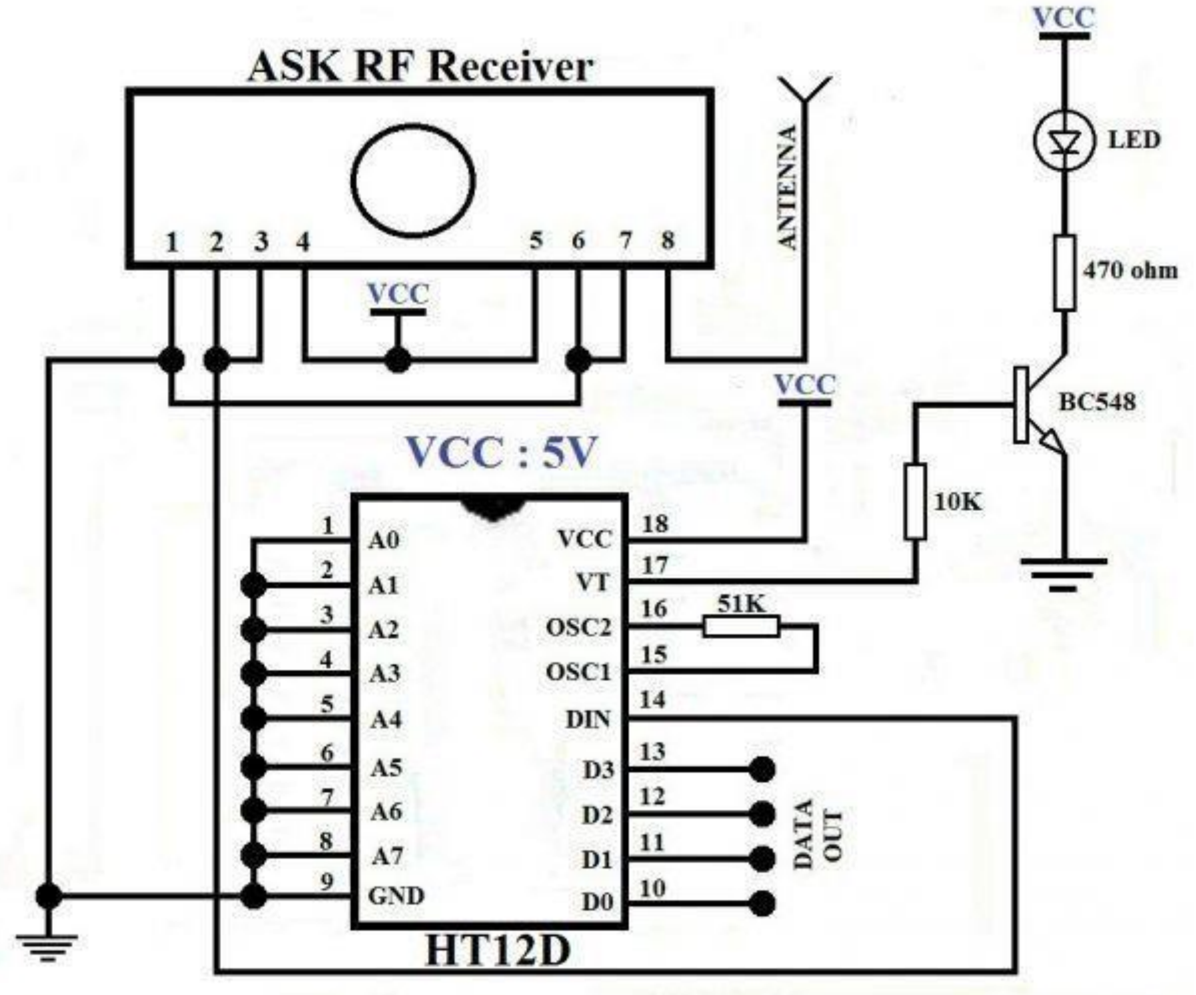

Fig:4.21 Receiver Circuit

HT12E Encoder

Encoder IC (HT12E) receives parallel data in the form of address bits and control bits. The control signals from remote switches along with 8 address bits constitute a set of 12 parallel signals. The encoder HT12E encodes these parallel signals into serial bits. Transmission is enabled by providing ground to pin14 which is active low. The control signals are given at pins 10-13 of HT12E. The serial data is fed to the RF transmitter through pin17 of HT12E.Transmitter, upon receiving serial data from encoder IC (HT12E), transmits it wirelessly to the RF receiver. The receiver, upon receiving these signals, sends them to the decoder IC (HT12D) through pin2. The serial data is received at the data pin (DIN, pin14) of HT12D

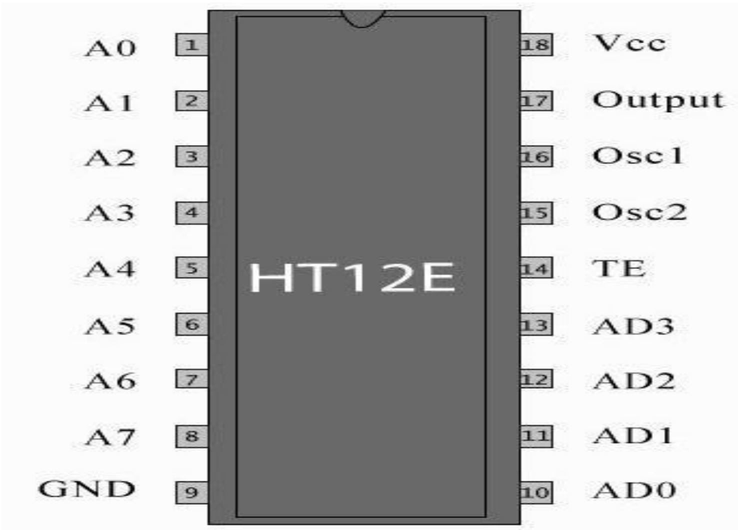

Fig:4.22 HT12E Encoder

The decoder then retrieves the original parallel format from the received serial data. HT12E is an encoder integrated circuit of 212 series of encoders. They are paired with 212 series of decoders for use in remote control system applications. It is mainly used in interfacing RF and infrared circuits. The chosen pair of encoder/decoder should have same number of addresses and data format. Simply put, HT12E converts the parallel inputs into serial output. It encodes the 12 bit parallel data into serial for transmission through an RF transmitter. These 12 bits are divided into 8 address bits and 4 data bits. HT12E has a transmission enable pin which is active low.

When a trigger signal is received on TE pin, the programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium. HT12E begins a 4-word transmission cycle upon receipt of a transmission enable. This cycle is repeated as long as TE is kept low. As soon as TE returns to high, the encoder output completes its final cycle and then stops.

HT12D Decoder

HT12D is a decoder integrated circuit that belongs to 212 series of decoders. This series of decoders are mainly used for remote control System applications, like burglar alarm, car door controller, security system etc. It is mainly provided to interface RF and infrared circuits. They are paired with 212 series of encoders. The chosen pair of encoder/decoder should have same number of addresses and data format. In simple terms, HT12D converts the serial input into parallel outputs. It decodes the serial addresses and data received by, say, an RF receiver, into parallel data and sends them to output data pins. The serial input data is compared with the local addresses three times continuously. The input data code is decoded when no error or unmatched codes are found. A valid transmission in indicated by a high signal at VT pin. HT12D is capable of decoding 12 bits, of which 8 are address bits and 4 are data bits. The data on 4 bit latch type output pins remain unchanged until new is received.

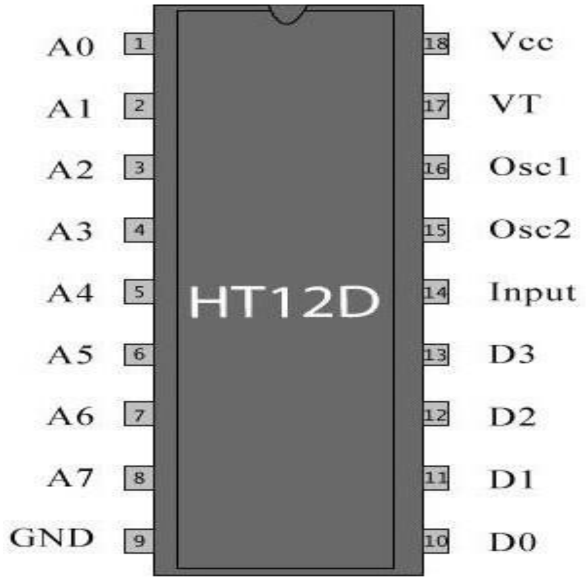

Fig:4.23 HT12D Decoder

4.9 Driver IC ULN2003

The IC ULN2003A is a Darlington transistor array which deals with high-voltage and high-current. There are various types of relay driver ICs such as a high side toggle switch, low side toggle switch, bipolar NPN transistor, Darlington transistor, N-channel MOSFET, ULN2003 driver IC.

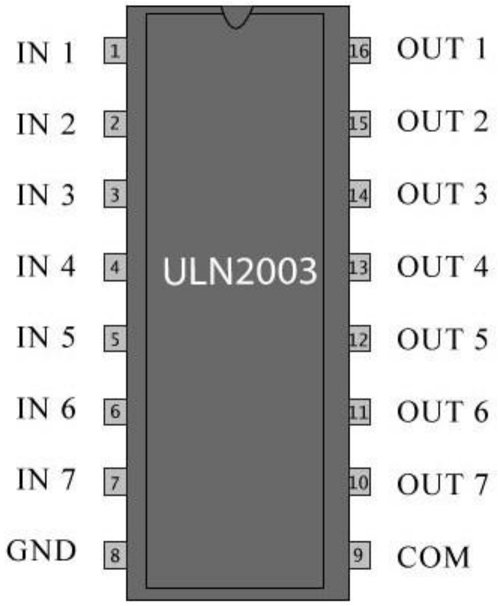

Fig:4.24 ULN2003 Driver IC

The pin diagram of IC ULN2003A is shown in the above figure which consists of 16 pins. The IC ULN2003A comprises of 7-NPN Darlington pairs as shown in the internal schematic diagram and is typically used to switch inductive loads (dissipates voltage spikes if any using suppression diode) and to drive stepper motors.

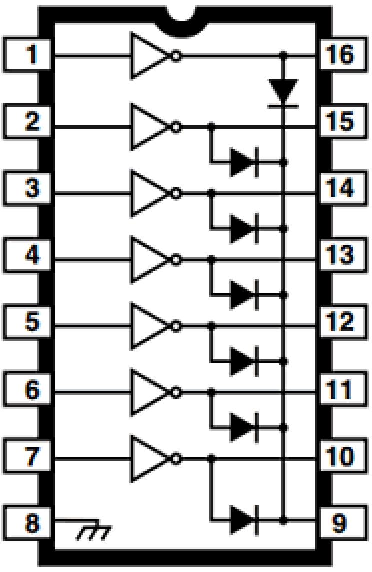

Fig:4.25 ULN2003 Internal Schematic Diagram

It is difficult to use a number of relays with transistors, so, relay driver IC ULN2003A can be used for availing more relays. We can use seven relays with relay driver circuit using ULN2003.

4.10 Ultrasonic Sensor

Fig:4.26 Ultrasonic Sensor

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

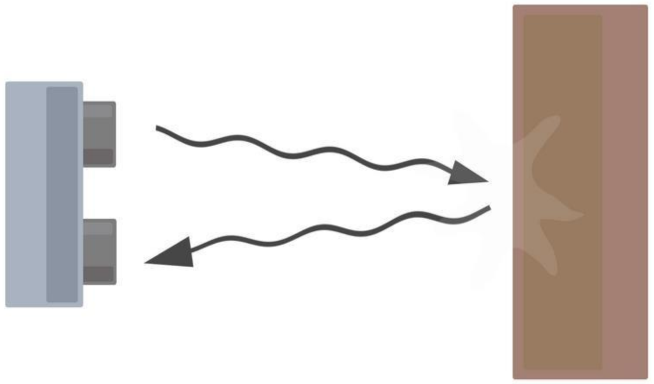

Fig:4.27 Ultrasonic Sound Waves Tx And Rx

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

It is important to understand that some objects might not be detected by ultrasonic sensors. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object, but are deflected away from the Ultrasonic sensor. It is also possible for the object to be too small to reflect enough of the sound wave back to the sensor to be detected. Other objects can absorb the sound wave all together (cloth, carpeting, etc), which means that there is no way for the sensor to detect them accurately. These are important factors to consider when designing and programming a robot using an ultrasonic sensor.

How does an Ultrasonic Distance Sensor work?

The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).

The speed of sound is approximately 341 meters (1100 feet) per second in air. The ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object. It uses the following mathematical equation:

Distance = Time x Speed of Sound divided by 2

Time = the time between when an ultrasonic wave is transmitted and when it is received

You divide this number by 2 because the sound wave has to travel to the object and back.

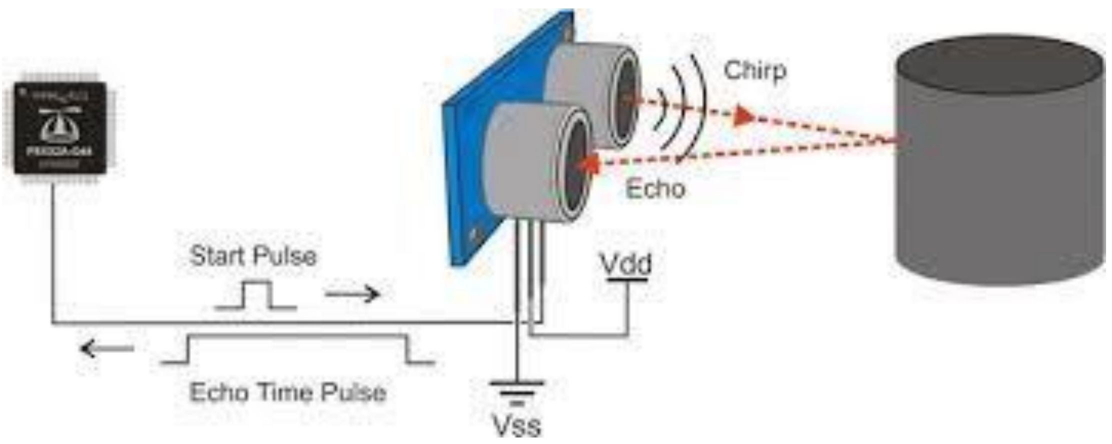

Fig:4.28 Working Of Ultrasonic Sensor

CHAPTER 5

SOFTWARE REQUIREMENT

5.1 Embedded C

Embedded C is a set of language extensions for the C programming language by the C standards Committee to address commonality issues that exit between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct banks, and basic I/O operations.

Examples of properties of typical embedded computers when compared with general-purpose counterparts are low power consumption, small size, rugged operating ranges, and low per-unitcost.

This comes at the price of limited processing resources, which make them significantly more difficult to program and to interact with however, by building intelligence mechanism on top of the hardware, taking advantages of possible existing sensors and the existence of a network of embedded units, one can both optimally manage available resources at the unit and network levels as well as provide augmented functions, well beyond those available. For example, intelligent techniques can be designed to manage power consumption of embedded systems.

CHAPTER 6

RESULT

To give the flag data in vehicle itself these virtual traffic lights are utilized. With these two models we can discover which flag is appearing a traffic convergences in a specific range.

The two models , one will be at traffic crossing points and one more will be in-vehicle to help driver and auto green traffic lights for emergency vehicles.

For the vehicles that take left turn will be indicated by the free left inside the vehicle itself so even if there is red signal we can take the left turn if free left is indicated.

CHAPTER 7

CONCLUSIONS

The paper provides the safety analysis of vehicle drivers by using the virtual traffic lights in-vehicles at the traffic intersections.

These virtual traffic lights assists the driver to identify the traffic signals before a certain range at traffic junctions and obstacle identification is done, then driver can know which signal is going on in traffic junction and he can find any obstacles are coming towards the vehicle when he is moving towards the traffic junction and then virtual traffic light system assists the driver in LCD displays the distance between vehicle and obstacle and traffic signal time will be shown in LCD display. Traffic signals will be identified by LED lights in-vehicle itself. Addition to this auto green signal for emergency vehicles will save the time to reach its destinations.

Virtual Traffic Lights Future tasks would concentrate around significantly more point by point distinguishing proof of flaws and their causes and consequent thorough danger examination. It would be prescribed to investigate car crashes information to improve evaluation of the seriousness and likelihood levels utilized as a part of hazard counts. The good judgment and designing background would be a manual for investigate fitting moderation measures considering likewise the peril of high as well as medium and even perhaps low class. It may be useful to consider unflinching quality data for hardware and point the finger at rates for programming to improve the assessment of the peril levels.

This paper is to reduce the traffic congestion which results in long waiting times to turn signal green, loss of fuel and money. For national development it necessary to reduce traffic.

REFERENCE

[1] Yang, B., Zheng, R., Shimono, K., Kaizuka, T. and Nakano, K., 2017. Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalised intersections. IET Intelligent Transport Systems, 11(2), pp.76-83.

[2] Zhang, K., Sheng, Y. and Li, J., 2012. Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. IET Intelligent Transport Systems, 6(3), pp.282291.

[3] Bazzi, A., Zanella, A., Masini, B.M. and Pasolini, G., 2014, June. A distributed algorithm for virtual traffic lights with IEEE 802.11 p. In Networks and Communications (EuCNC), 2014 European Conference on (pp. 1-5). IEEE.

[4] Yapp, J. and Kornecki, A.J., 2015, August. Safety analysis of virtual traffic lights. In Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on (pp. 505-510). IEEE.

[5] Yang, Bo, Rencheng Zheng, Keisuke Shimono, Tsutomu Kaizuka, and Kimihiko Nakano. "Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalised intersections." IET Intelligent Transport Systems 11, no. 2 (2017)

[6] Higaki, H., 2014, March. Virtual Traffic Signals by Cooperation among Vehicle-Mounted Mobile Computers. In New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on (pp. 1-6). IEEE.

[7] Yang, B., Zheng, R., Shimono, K., Kaizuka, T. and Nakano, K., 2017. Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalised intersections. IET Intelligent Transport Systems, 11(2), pp.76-83.

[8] Zhang, K., Sheng, Y. and Li, J., 2012. Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. IET Intelligent Transport Systems, 6(3), pp.282291.

[9] Bazzi, A., Zanella, A., Masini, B.M. and Pasolini, G., 2014, June. A distributed algorithm for virtual traffic lights with IEEE 802.11 p. In Networks and Communications (EuCNC), 2014 European Conference on (pp. 1-5). IEEE.

[10] Yapp, J. and Kornecki, A.J., 2015, August. Safety analysis of virtual traffic lights. In Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on (pp. 505-510). IEEE.

[11] Yang, Bo, Rencheng Zheng, Keisuke Shimono, Tsutomu Kaizuka, and Kimihiko Nakano. "Evaluation of the effects of in-vehicle traffic lights on driving performances for unsignalised intersections." IET Intelligent Transport Systems 11, no. 2 (2017)

[12] Higaki, H., 2014, March. Virtual Traffic Signals by Cooperation among Vehicle-Mounted Mobile Computers. In New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on (pp. 1-6). IEEE.