

SMART TRANSPORTATION FOR SMART CITY WITH ADAS USING IOT AND AI

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in
Electronics and Communication Engineering

By

ABIRAMI S (38130004)

ESAKKIAMMAL A (38130061)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH 2022



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of ABIRAMI S(Reg. No. 38130004) and ESAKKIAMMAL A (Reg. No.38130004) who carried out the project entitled "SMART TRANSPORTATION FOR SMART CITY WITH ADAS USING IOT AND AI" under our supervision from December 2021 to March 2022.

**Internal Guide
(Dr.G. Jegan)**

**Dr. T . RAVI , M.E ., Ph. D.,
Head of the Department**

Submitted for Viva voce Examination held on _____

Name :

Signature:

Internal Examiner

External Examine

DECLARATION

We **ABIRAMIS** (38130004) and **ESAKKIAMMALA** (38130061) hereby declare that the Project entitled **“SMART TRANSPORTATION OF SMART CITIES WITH ADAS USING IOT AND AI”** done by us under the guidance of **Dr.G.JEGAN M.E ., Ph. D.**, at Sathyabama Institute of Science and Technology , **Chennai-600119** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.

2.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We wish to express our sincere thanks to the people who extended their help during the course of our project work.

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their encouragement in doing this project and for completing it successfully .We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhitha, M.E ., Ph. D.,Dean, School of Electrical and Electronics Engineering** and **Dr. T. RAVI, M.E ., Ph. D., Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would also like to express our sincere and deep sense of gratitude to our internal guide **Dr.G.JEGAN, M.E ., Ph. D.,** for his valuable guidance, constant technical support and stupendous encouragement, which enabled us to complete our project successfully.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

Further the acknowledgement would be incomplete if we would not mention a word of thanks to our most beloved **Parents** whose continuous support and encouragement all the way through the course has led us to pursue the degree and confidently complete the project.

ABSTRACT

Everyone has easy access to bus transportation. These bus transportation facilities are open to everybody without prejudice. It is capable of transporting a large number of people at once. Bus transportation has existed since the dawn of civilization and is often handled by the government or commercial groups. In nations such as India, public transportation is owned and administered by the government, and fares are kept low to make it affordable to the general public.

To advance this to the next step, we've included a smart system that includes a Raspberry Pi and a camera. Using the CNN algorithm, the bus will stop when a human is recognized ahead of the bus. Other sensors, such as the ADXL335 that acts as an accident detection sensor and the IR sensor that acts as a speed breaker detection sensor, transmit data to the Arduino UNO. Additionally, we have installed a bus transport platform that consists of a radio frequency transmitter that transmits a signal continuously that is received by the bus's receiver and automatically stops.

The primary objective of our project is to detect accidents and send emails to the appropriate parties using IOT, to detect people closer to the bus using AI, to make the bus stop at each and every bus stop and then automatically slow down to a safe speed, and to avoid highway accidents using sensor technologies.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
1	INTRODUCTION	1
	1.1 OBJECTIVE	2
2	LITERATURE SURVEY	3
3	METHODOLOGY	6
	3.1 EXISTING SYSTEM	6
	3.2 PROPOSED SYSTEM	7
4	BLOCK DIAGRAM AND DESCRIPTION	7
	4.1 BLOCK DIAGRAM DESCRIPTION	7
	4.2 BLOCK DIAGRAM	8
5	5.1 HARDWARE REQUIREMENT	9
	5.1.1 POWER SUPPLY	10
	5.1.2 RESISTOR	13
	5.1.3 CAPACITOR	16
	5.1.4 CRYSTA OSCILLATOR	17
	5.1.5 ARDUINO UNO	19

	5.1.6 RASPBERRY PI	25
	5.1.7 IR SENSOR	32
	5.1.8 MOTOR DRIVER (L23D)	36
	5.1.9 RF TRANSMITTER RECEIVER MODULE	38
	5.1.10 WEBCAMS	41
	5.1.11 ADXL335	46
	5.2 SOFTWARE REQUIREMENTS	50
	5.2.1 PYTHON	50
	5.2.2 ARDUINO IDE	54
	5.2.3 LINUX OS	61
6	CIRCUIT DIAGRAM	64
7	CONSTRUCTION AND WORKING	65
8	SOFTWARE USED	66
	8.1 RASPBERRY PI CODE	66
	8.2 ARDUINO UNO CODE	71
9	OUTPUT AND HARDWARE VIEW	75
10	FUTURE SCOPE	77
11	CONCLUSION	78
	REFERENCE	79

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
5.1	Power Supply	10
5.2	Transformer winding	10
5.3	Bridge Rectifier	11
5.4	Carbon composition resistors	13
5.5	Carbon Film Resistors	14
5.6	Metal film resistors	15
5.7	Wire wound Resistors	15
5.8	Capacitor	16
5.9	Capacitor Pin Diagram	17
5.10	Quartz crystal	18
5.11	Arduino Uno board	19
5.12	Arduino Uno pin diagram	21
5.13	Block diagram	26
5.14	Raspberry pi	26
5.15	Raspberry pi pin diagram	28
5.16	Raspberry pi top view	30
5.17	Raspberry pi labelled	31
5.18	Raspberry bottom view	32

5.19	IR Sensor	33
5.20	IR Sensor diagonal view	34
5.21	Transmitter and Receiver Block diagram	35
5.22	LED Bulb	35
5.23	Circuit Diagram	36
5.24	Motor Driver IC	36
5.25	Circuit Diagram	37
5.26	Infrared Transmitter	40
5.27	Webcam	42
5.28	Webcam top view	42
5.29	Webcam Internal view	43
5.30	Webcam bottom view	43
5.31	Image to binary bit conversion	44
5.32	ADXL335	46
5.33	Python LOGO	51
5.34	Arduino Uno IDE	54
5.35	LINUX LOGO	62
6	CIRCUIT DIAGRAM	64
9	HARDWARE VIEW	75

CHAPTER 1

INTRODUCTION

In the industrialized world, road deaths are a big worry. According to recent research, a third of fatal or severe accidents are caused by excessive or unsuitable speed, as well as changes in the highway, such as the presence of road work or unanticipated barriers.

Traffic authorities, the car industry, and transportation research organizations are all concerned with reducing the number of accidents and mitigating their repercussions. One critical line of action is the implementation of advanced driver assistance systems (ADAS), which are auditory, tactile, or visual signals generated by the car to alert the driver to the danger of a collision. These technologies are now available in limited quantities in commercial vehicles, and future trends predict that increased safety will be accomplished via autonomous driving controls and an increasing number of sensors on both the road infrastructure and the vehicle itself.

A prominent example of driver assistance systems is cruise control (CC), which is capable of maintaining a steady user-preset speed, and its development, adaptive cruise control (ACC), which adds the capacity of maintaining a safe following distance to CC. A disadvantage of these systems is that they are unable of independently identifying straight and curving sections of road where the speed must be reduced to prevent accidents.

A bus stop is a defined location where buses halt to allow passengers to enter or exit the vehicle. Bus stops are often constructed in accordance with their degree of utilization, with shelters, seats, and perhaps electronic passenger information systems at bus stops in high-traffic areas and a basic pole and flag at bus stops in low-traffic areas. Bus stops are crowded together in some places to form transport hubs, allowing for easy interchange between routes departing from neighboring stops and with other forms of public transport.

Bus stops enhance passenger safety in a number of ways:

Bus stops stop people from trying to get on or off the bus in dangerous places, like at an intersection or when a bus is turning and isn't using the curb lane. A bus driver can't be expected to keep looking for people who want to get on the bus all the time. A bus stop means that the driver only has to look for people who want to get on the bus at the start of each stop. Passengers are forced to form groups before boarding a bus, which speeds up the boarding process. At night, when there aren't as many people on the bus, some rules may be relaxed. Passengers may be able to get off the bus anywhere as long as it's safe.

People who drive buses can stop at bus turnouts or lay-bys so they don't have to slow down traffic on the main road.

1.1 OBJECTIVES

The bus stop should always be located near the junction to maximize efficiency and ease of use for passengers. As a general rule, most bus drivers will not wait at the bus stop. This project proposes the use of IOT and AI to create an automated bus stop system that makes use of RF technology, as well as an automatic slow-down brake to help prevent highway accidents.

CHAPTER 2

LITERATURE SURVEY

1.EARLYWARNING SPEED-BREAKER

**Mohit Jain, Ajeet Pal Singh, Soshant Bali, Sanjit Kaul System IJSRD-
International Journal for Scientific Research&Development 2014 Speed-breaker
early Warning System(SWAS)**

There is evidence that speed breakers may result in accidents and harm. When a vehicle approaches a speed breaker at a rate higher than a predetermined threshold, the chance of an accident or injury is significant. Speed-breakers are almost invisible in low-visibility situations, such as at night or in the presence of fog, rain, or snow. This issue is especially significant in underdeveloped nations, where speed limit signs are not always present. We propose an early warning system based on a smartphone application that notifies the driver when the vehicle is nearing a speed breaker. Additionally, the programme continuously analyses the accelerometer of the smartphone to identify previously undisclosed speed-breakers. The suggested detection technique is simple to implement, since it does not involve reorienting the accelerometer. This is a significant addition of our work, since earlier techniques reoriented the accelerometer using costly calculations. The system was assessed utilizing 678 kilometers of driving data from 22 different drivers, five different kinds of automobiles, and four different types of smart phones. The findings are quite encouraging and might be enhanced further by pooling detection data from many smart phones.

2. NAVIGATION OF BLIND PEOPLE USING PASSENGER BUS ALERT SYSTEM

Swapnil Gholap*, Govind Ekshinge, Parag Naik**,Prof.S.D.Chavan****

**Department of Electronics and Telecommunication,Dr.D.Y.Patil Institute of
Engineering and Technology,Pune,Maharashtra,India**

The primary issue for visually impaired individuals is identifying and avoiding obstacles, as well as adapting to their surroundings. Conventional techniques of navigation used by visually impaired persons include talking signs, guide cans, and echolocations. We have explored some current systems in this arena and offered an approach that may be implemented using wireless sensor networks to assist visually impaired individuals in navigating public transportation. According to the suggested concept, a radio frequency unit is installed at the bus stop where

visually challenged passengers wait and is subsequently detected by another unit on the bus. The visually handicapped will provide information about their position at the bus stop through an RFID tag. The ARM-7 generates the signals, which are then sent to the bus through the RF module via the ARM controller. The bus routes are converted to audio using a speech synthesizer APR9600. The visually challenged are notified by a buzzer that is activated when the bus arrives at the bus stop. The visually impaired individual's request to board a certain route bus is communicated to him or her through a speech synthesizer device. The bus routes will be sent to the visually impaired through the RF transceiver, and the visually impaired person will be informed of the bus routes via the speaker. The secondary objective of this initiative is to assist older citizens and illiterates in navigating independently using public transportation.

3.DRIVER ASSIST SYSTEM(DAS) TO PREVENT ROAD ACCIDENTS

Jayalakshmi .R,Haripriya.K Student,Department of Biomedical Instrumentation Engineering, Avinashilingam Institute of Home science and higher education for women, Coimbatore

Traffic accidents occur on a daily basis, and according to 2015 data, at least ten people perished and another 35 were wounded after a bus collided with a ditch on February 4, 2015, while travelling between the Indian towns of Pune and Satara. Numerous individuals have died as a result of the dangers associated with footboard riding on buses. According to reports, four students riding the footboard of an overloaded bus were killed after they collided with a vehicle on 10th December 2012 at Chennai's Rajiv Gandhi Salai. Footboard travel on buses is harmful, and it must be avoided by the implementation of a system equipped with modern technology that prevents people from travelling in footboard. In ordinary Metropolitan buses, doors are provided to close off the vehicle after the customer has boarded and disembarked. However, this is not tamper-resistant, which allows drivers to simply fake it. The DAS is equipped with a system in which the microcontroller continually analyses the output from sensors mounted on the bus's footboard and automatically stops the bus if somebody stands on the footboard for an extended period of time. Additionally, the system includes an enhanced lane detection system and an infrared-based driver tiredness monitoring system. This system is comprised of a distance reflecting sensor that is capable of preventing accidents by regulating the speed of the vehicle. Additionally, it has an LCD display of the bus's passenger count and an LED indicator of available seats to the computers located outside the vehicle.

4.PROACTIVE ACCIDENT AVOIDANCE SYSTEM USING IOT

Mithilaesh. J, Santhosh Kumer.K, Ashwin Balaji.P,Dr.S.Hemalatha,Dr.V.Subedha
Depatment of Computer Science & Engineering Panimalar Institute of
Technology,Chennai, Tamilnadu,India

Numerous accidents occur as a result of driver mistake. The primary cause of car collisions will be alcohol intake, tiredness, and distracted eyesight. The suggested system is intended to aid in the analysis of these parameters and the development of an accident avoidance system. Here, we identify driver sleepiness by attaching a camera to the steering wheel. The camera's picture of the driver's face is evaluated using the MatLab programme, and the driver is warned if sleepiness is identified. Additionally, we integrate an alcohol detection system and a crash monitoring system. Alcohol sensors are utilised to detect drunk drivers, and SMS notifications are delivered to the concerned individual and police in the event of drunk driving. Ultrasonic sensors are used to determine the distance between automobiles and, in the event of a collision, the vehicle is either halted or slowed down. TSOP Sensor, Alcohol Sensor, Ultrasonic Sensor, Gaze Detection, Eye wink sensor, Microcontroller are some of the key terms.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

Design of RF based speed control system for vehicles

Nowadays, people drive at a high rate of speed, accidents occurs more often, and we lose our priceless lives as a result of a little error while driving (school zone, hills area, and highways). Thus, in order to prevent such accidents and to notify drivers, as well as to help them manage their vehicle's speed in such locations, the highway department has put signboards. Thus, informing the driver of zones and speed limits automatically will be accomplished via the use of RF technology. The primary purpose is to develop a Smart Display controller capable of controlling vehicle speed and monitoring zones through signboards. The controller will operate on an embedded device. The Smart Display & Control (SDC) system may be customized to fit into the dashboard of a car and shows information about the vehicle. The project will be divided into two distinct sections. Once the information from the zones is received, the vehicle's embedded unit immediately alerts the driver to lower speed in accordance with the zone and waits a few seconds; otherwise, the vehicle's SDC unit reduces speed automatically. The purpose of this project is to develop a sensor system for infrastructure-to-vehicle communication that can transfer information generated by active signals placed on the road in order to change the vehicle's speed and avoid crashes. By active signals, we mean conventional traffic signals that combine long-range active RFID tags that contain data. This data will be gathered in real time using RFID sensors mounted on the car, which we have adapted to automatically adjust its speed to the road conditions. We have designed a fuzzy logic control algorithm that operates on the longitudinal speed of the vehicle in conjunction with actuators that control the vehicle's throttle and brake systems in order to achieve and maintain a specified goal speed.

3.2 PROPOSED SYSTEM

To maximize the efficiency and convenience of transit for passengers, the junction will always be the optimal position for the bus stop. Generally, bus drivers will not stop at designated bus stops. This concept proposes an automated bus stop system, automatic slowing of the bus in speed brakes, person detection, sending signals to ambulances in the event of a bus accident, and preventing highway accidents. The bus will automatically stop at the bus stop; the bus unit is equipped with an RF receiver that transmits bus stop information. The RF transmitter will communicate information about the specific bus stops that will be halted. After the bus unit receives information from the RF receiver, the relay automatically stops and starts the bus engine without the need for human involvement. The TSOP sensor is used to slow the buses down while they are in speed brake mode. Another TSOP sensor will be used to initiate braking when it detects cars braking in front of it on highways. The suggested framework focuses on developing smart bus transportation via the use of AI and IoT. We've built a robotic setup utilizing a Raspberry Pi and an Arduino UNO, as well as a smart platform employing a radio frequency transmitter and receiver.

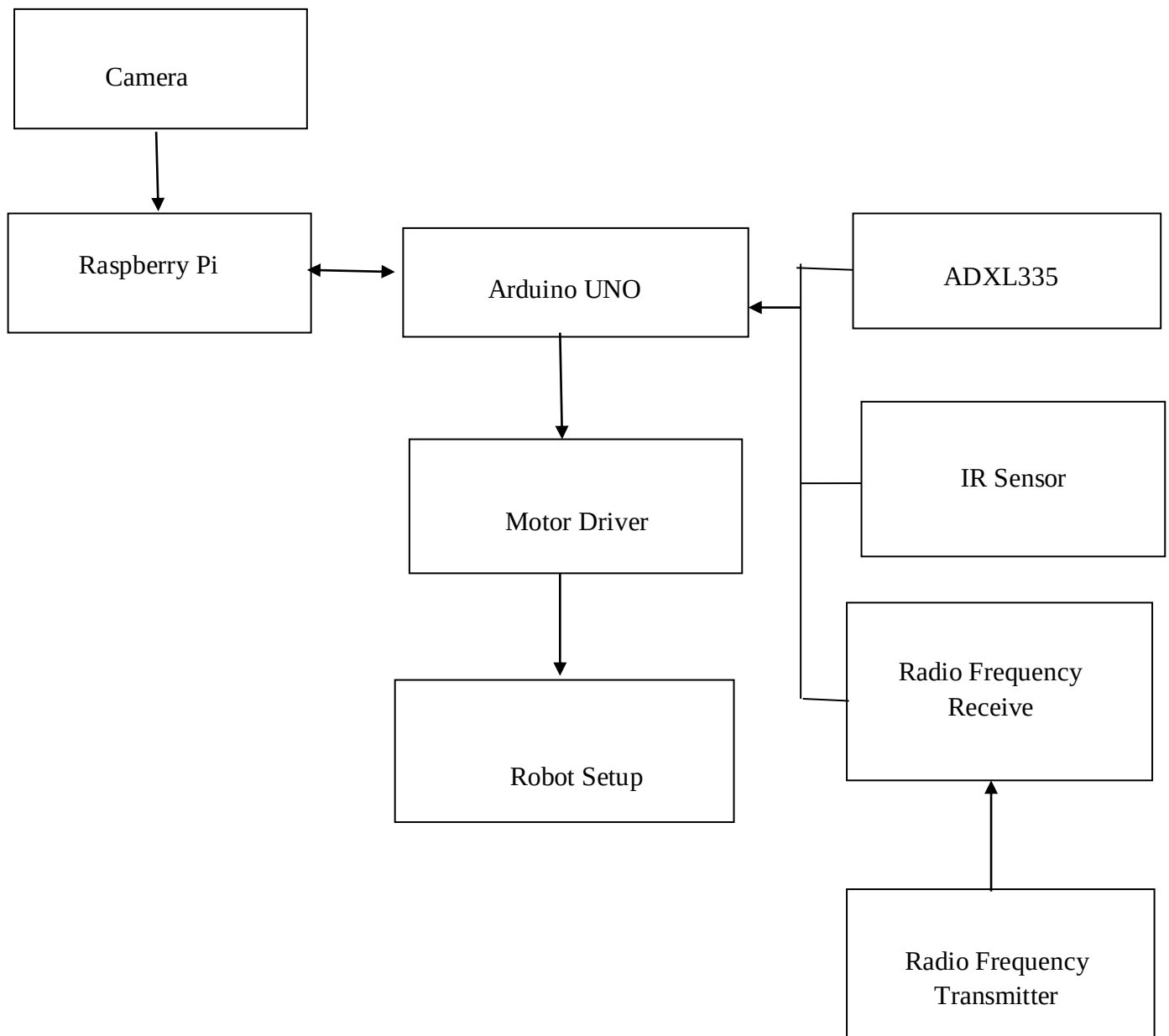
CHAPTER 4

BLOCK DIAGRAM AND DESCRIPTION

4.1. Block Diagram Description:

The arduino and raspberry pi are linked using serial communication, while the raspberry pi and camera are connected by USB. Android apps can connect to the cloud through a Raspberry Pi and use its services. 433 Hz data is sent from additional sensors linked through GPIO or radio frequencies. The robot setup is linked to the motor driver, which is connected to the motor by means of standard GPIO.

4.2 BLOCK DIAGRAM



CHAPTER 5

5.1 Hardware Requirements

5.1.1 Power supply

5.1.2 Resistor

5.1.3 Capacitor

5.1.4 Crystal Oscillator

5.1.5 Arduino Uno

5.1.6 Raspberry pi

5.1.7 IR Sensor

5.1.8 Motor Driver (L23d)

5.1.9 RF Transmitter Receiver Module

5.1.10 Webcams

5.1.11 ADXL335

5.2 Software Requirements

5.2.1 Python

5.2.2 Arduino IDE

5.2.3 Linux OS

5.1.1 Power supply

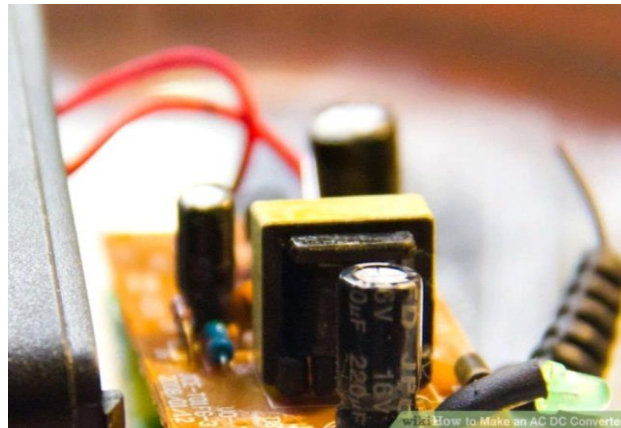


Fig 5.1 Power Supply

Alternating current (AC) is used to transmit electricity via power lines and to operate high-power equipment such as appliances and lights. The features of alternating current make it excellent for transmission over long distances and for supplying huge quantities of power to largely uncontrolled applications such as heat and light generation.

A transformer is made up of two windings of wire that are magnetically linked. The main winding is one of the windings. The primary is powered by the main alternating current source. The secondary winding is the other. The secondary will act as the AC DC converter's power input. This transformer, as well as the other components required to construct an AC DC converter, are widely accessible in electronic and hobby shops.



Fig 5.2 Transformer winding

The alternating current mains provide 120 volts AC. If 120 volts AC is converted straight to DC, the resultant DC voltage is significantly too high to be used by appliances and gadgets. The transformer's main and secondary windings were scaled in order to create a lower voltage on the secondary winding. Alternately, you may use a secondary winding. The secondary winding's AC output should be rated at the same voltage as the DC generated.

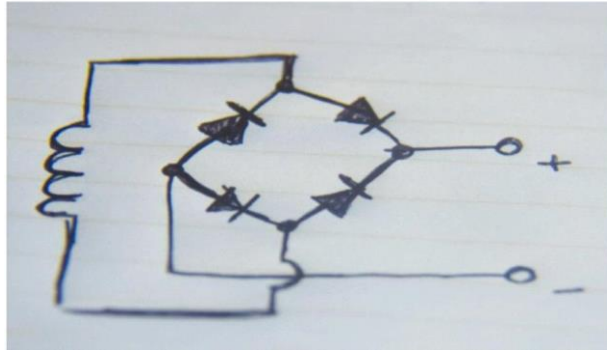


Fig 5.3 Bridge Rectifier

A step down transformer is used to convert a 230 volt alternating current mains supply to 12 volts alternating current. Transformers are used to get power. Bridge rectifiers are used to convert 12 volts alternating current to 12 volts direct current. The voltage regulator IC (integrated circuit) 7805 converts 12 volts direct current to 5 volts direct current. The circuit's initial stage is a transformer of the step-down kind that reduces the amplitude of the input voltage. The majority of electrical projects will use a 230/12V converter to convert the 230V AC mains supply to a 12V AC supply.

The next step is a diode-bridge rectifier, which employs four or more diodes, depending on the bridge rectifier type. Choosing a certain diode or other switching device for a matching rectifier requires some considerations about the device, such as its Peak Inverse Voltage (PIV), forward current, and voltage ratings. It is responsible for generating unidirectional or direct current to the load through a series of diodes for each half cycle of the input signal. Because the output of the diode bridge rectifiers is pulsing, filtering is required to convert it to pure DC. Filtering is often accomplished using one or more capacitors connected across the load, as seen in the following image, where wave smoothing is accomplished. Additionally, the capacitor rating is dependent on the output voltage.

The last element of this controlled DC supply is a voltage regulator that maintains a consistent output voltage. Therefore, to decrease this voltage and keep a consistent level, the transformers are static device of which electric power in our circuit is transformed into electric power of same frequency to another circuit. It can raise or lower the voltage present in the circuit. It works on the principle of mutual induction; in this project we will be using a step down transformer to provide the necessary supply for the circuit. Here we will step-down a 230v AC into 12v AC.

i. Rectifier

Rectification is often accomplished in a power supply unit by the use of diodes. Diodes have the attribute of having a one-way current flow. The project will use a full wave rectifier, which is the most often used kind of rectifier in business. In this scenario, the output has a ripple frequency of 100Hz, since each attenuation results in a ripple frequency that is twice the attenuation

frequency. The output direct current is not a pure direct current; rather, it is a pulsing direct current voltage. As a result, it transforms the required AC to DC, which is utilized to significantly minimize harmonics and ripple in the output.

ii. Filter

To get a 50Hz DC voltage, we must use low pass filters. As a result, a capacitive filter is utilized, in which a capacitor is connected to the rectifier output and a DC voltage is acquired across it. The filtered waveform is basically a direct current with very little ripple, which is sent to the load.

iii. Regulators

The capacitor's output voltage is filtered and then controlled. Voltage regulators are devices that maintain a constant output voltage regardless of supply variances, load variations, or temperature differences. Here, we're going to utilize a fixed voltage regulator, the LM7805.

5.1.2 Resistor

An electric resistor is a passive component with two terminals that is used to oppose and restrict current. A resistor operates according to Ohm's Law, which says that the voltage across a resistor's terminal is proportional to the current flowing through it.

Ohm's Law $\Rightarrow V=IR$

Where V is the voltage applied across resistor,

I is the current flowing through it,

And R is the constant called resistance.

The unit of resistance is ohms.

Types of Resistors:

Resistors may be categorized generically according to the following terms. There are many factors to consider, ranging from the sort of material utilised to the power rating and resistance value.

i. Fixed resistors:

In rare instances, an electrical circuit may need less current than the input value. In these instances, fixed resistors are utilized to restrict the flow of current. In some cases, an electrical circuit may need a lesser amount of current to flow through it than the input value. Fixed resistors are used in these cases to limit the flow of current.

i. **Carbon Composition Resistors:**

These cylindrical rod-shaped resistors are made of a blend of carbon granules and powdered ceramic. The resistor value is determined by the ceramic material's composition. Increased ceramic content results in increased resistance. Because the rod is covered with an insulating substance, there is a risk of damage from the extreme heat generated during soldering. Due to its high cost, this kind of resistor is seldom utilized anymore.

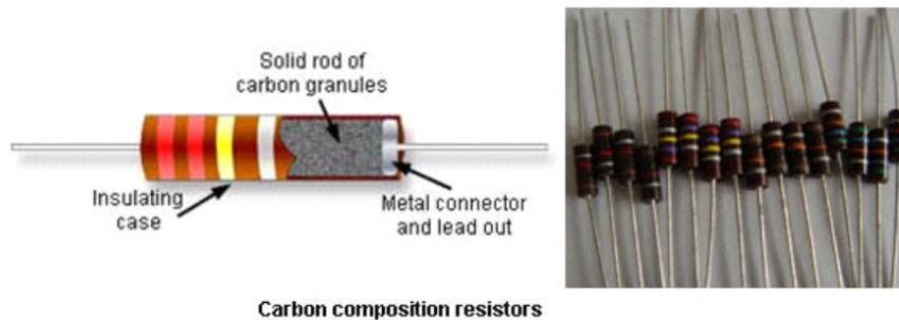


Fig 5.4 Carbon composition resistors

ii. **Carbon film resistors:**

Carbon film resistors are created by depositing a layer of carbon on an insulating substrate. Through the carbon sheet, helical incisions are made to trace a lengthy and helical resistive route. The resistance may be adjusted by varying the resistivity of the carbon material and altering the form of the resistor. These resistors are inductive due to the helical resistive channel and are also employed in RF applications. This resistor's functioning requires a very high level of pulse stability.

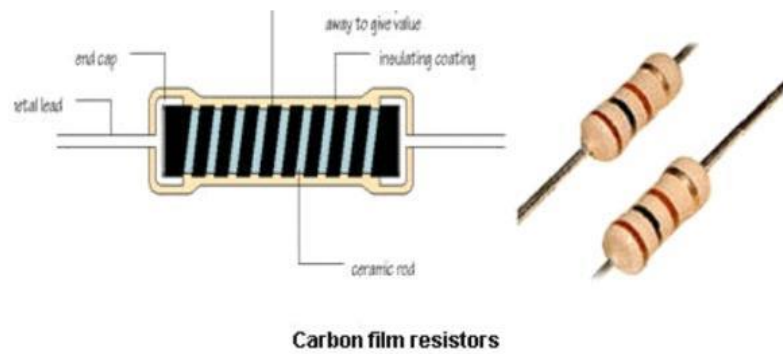


Fig 5.5 Carbon Film Resistors

iii. Metal Film resistors

Ceramic rods covered in metal or metal oxide are used to make these resistors. The thickness of the coating layer will have a significant impact on the resistance value. Tolerance in metal film resistors ranges from a few ohms to hundreds of millions, and the range is quite large. Small carbon, metal, and oxide resistors are available in a variety of colors, such as dark red, brown, blue, green, grey, or white, with a claimed value of 100K Ohm.

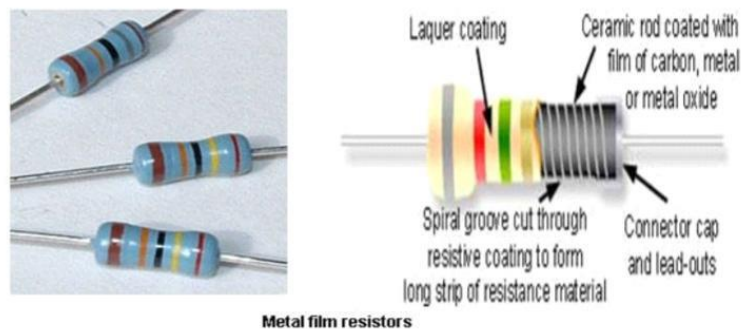


Fig 5.6. Metal film resistors

iv. Wire wound resistor

In terms of size and physical appearance, wire wound resistors may be a wide range of sizes and shapes. A tiny ceramic or glass fiber rod is wrapped in an insulating flame-proof cement film that serves as the wire's resistive element, which is often nickel/chromium or manganin alloy wire. They typically have low resistances, but they are capable of dissipate a great quantity of electricity at the same time.

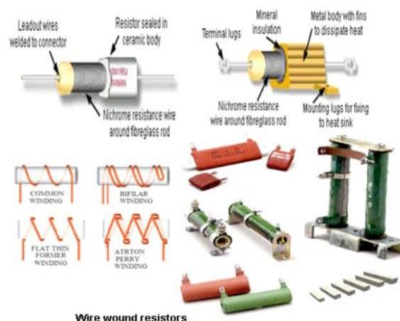


Fig 5.7 Wire wound Resistors

Due to the fact that wire wound resistors are main coils, they have a higher inductance than other forms of resistors, however wrapping the wire in portions with alternately reversed orientations might help reduce inductance. Other ways include bifilar winding to minimize the coil's cross-section area. Resistors with Ayrton-Perry windings are employed in the most demanding circuits.

5.1.3 Capacitor

Capacitors are kind of passive components that are used to store electrical charge. The charge (q) stored in a capacitor is calculated as the product of its capacitance (C) value and the applied voltage (V). Because capacitors have an infinite reactance at zero frequency, they are employed to block DC components or to bypass alternating current signals. In alternating current circuits, capacitors undergo a recursive cycle of charging and discharging, with the voltage and current across them determined by the RC time constant. Capacitors are utilized in this scenario to smooth out power supply fluctuations. Other applications include linking the different stages of an audio system and adjusting radio circuits. These are used to store energy, such as in the flash of a camera.

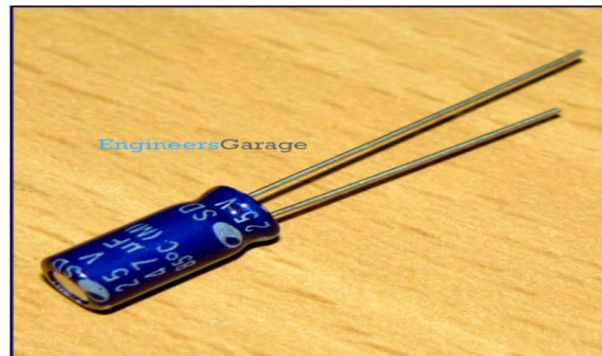


Fig 5.8 Capacitor

i. Pin diagram:

Capacitors are an extremely common kind of electrical component. It accumulates electric charges before discharging them into the circuit. It will conduct direct current but will enable alternating current to flow through it. Depending on the application, many types of capacitors were utilized, including ceramic, electrolytic, and mica. This article will discuss electrolytic capacitors.

Structure of Capacitor: A capacitor consists of two conducting plates, which are often formed of metal, connected by an insulator. Insulators are sometimes referred to as dielectrics and are composed of materials such as paper, plastic, ceramic, or glass. Two thin metal rods, referred to as the capacitor's legs, link the two plates electrically to the external circuit. These two plates are used to store charges between them. One of the plates is connected with positive voltage and other one with negative voltage. A capacitor is characterized by the parameters of capacitance.

Capacitance is the ratio of the charge difference between the plates to the overall voltage drop.

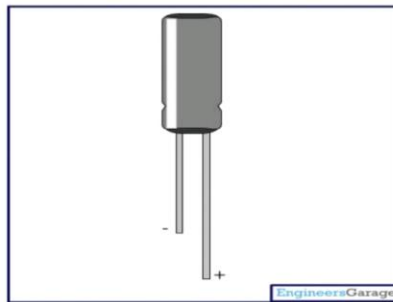


Fig 5.9 Capacitor Pin diagram

$$C=dQ/dV$$

The unit of capacitance is FARAD.

5.1.4 Crystal Oscillator

A quartz crystal resonator is critical to the operation of an electrical oscillator. Occasionally spelled crystal oscillator, it is a critical component of the feedback network. Electronic oscillators are employed in frequency control applications across a wide variety of industries, from semiconductors to aerospace. The heart of this sort of resonator is a quartz crystal. Their advantages over other resonators such as LC circuits, spinning forks, and ceramic resonators include a high quality factor, stability, compact size, and cheap cost.



Fig 5.10 Quartz Cystal

The fundamental principle behind the operation of a quartz crystal oscillator is the reverse piezoelectric effect, which occurs when an electric field is placed across certain materials and they begin to deform mechanically. Mechanical motions are determined by the quartz crystal's basic structure. Quartz is a naturally occurring mineral that exhibits piezoelectric properties; nevertheless, it is artificially created for use as a resonator since processing naturally existing quartz is complex and expensive. The image below depicts a commonly used quartz crystal resonator. It is commonly employed in the circuitry of electronic oscillators found in digital circuits and microcontrollers or processors.

5.1.5 ARDUINO UNO



Fig 5.11 Arduino Uno board

i. Specifications:

The ATmega2560 is a Microcontroller

- The operating voltages of these microcontrollers are 5 volts.
- The recommended input voltage range from 7 volts to 12 volts.
- The input voltage range from 6 volts to 20 volts
- The digital input/output pins are 54 where 15 of these pins will supply PWM output
- Analog input pins are 16
- DC current for each input/output pin 40 mA
- DC current used for 3.3V pin is 50 mA
- Flash memory like 256 KB where 8KB of flash memory is used with the help of boot loader
- The static random access memory (SRAM) is 8KB

v. Arduino UNO Peripherals

Introduction

The Arduino Uno is a single-board computer based on the ATmega1280 microcontroller (datasheet). It has 54 I/O pins, 16 analogue pins, four UARTs, a 16 MHz crystal oscillator, a USB port, a power connector, an ICSP header, and a reset button. It includes everything you need to get started with the microcontroller; just connect it to a computer through USB and power it with an AC-to-DC converter or battery. The Arduino Uno is compatible with the majority of shields produced for the Duemilanove or Diecimila boards.

Summary

Microcontroller	ATmega1280
Operating Voltage	5V
Input voltage(recommended)	7-12V
Input Voltage(limits)	6-20V
Digital I/O pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash memory	128 KB of which 4KB used by bootloader
SRAM	8KB
EEPROM	4KB
Clock Speed	16 MHZ

vi. Power

The Arduino Uno may be powered through USB or an external power source. Automatically, the power source will be picked.

The power connectors are as follows:

Vin: The voltage applied to the Arduino board when it is powered by an external power source. We can provide voltage through these pins or, if we supply voltage via the power jack, we can access it via this pin.

5V: The regulated power source will be utilised to power both the microcontroller and the rest of the board's components. This may be provided by VIN via an on-board regulator or via USB or another regulated 5V source.

3V3: The on-board FTDI chip will create a 3.3 volt supply. The maximum current that may be drawn is 50 Ma.

GND: Pins that have been grounded

Memory

The ATmega1280 includes 128KB of flash memory, 8KB of SRAM, and 4KB of EEPROM for storing code.

Input and Output

Using the pinMode(), digitalWrite(), and digitalRead() routines, each of the uno's 54 digital pins may be utilised as an input or output. They work at a voltage of 5 volts. Each pin may provide or receive a maximum of 40mA and has a 20-50 kOhm internal pull-up resistor. Additionally, several pins perform specific duties.

Serial numbers 0 (RX) and 1 (TX), Serial numbers 1:19 (RX) and 18 (TX), Serial numbers 2:17 (RX) and 16 (TX), Serial numbers 3:15 (RX) and 14 (TX). This device is used to receive and send TTL serial data. Pins 0 and 1 are linked to the FTDI USB-to-TTL Serial chip's equivalent pins.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2), these pins may be programmed to generate an interrupt on a low value, a rising or falling edge, or a value change.

For specifics, we may refer to the attachInterrupt() method.

The Uno features sixteen analogue inputs, each of which has a resolution of ten bits. They measure from ground to 5 volts by default, but the AREF pin and analogue reference() function allow you to adjust the top limit of their range.

Communication

The Arduino Uno has a variety of communication interfaces for connecting to a computer, another Arduino, or any other microcontroller. The ATmega1280 has four hardware UARTs for TTL(5V) serial transmission, as well as 12C and SPI connectivity. The Arduino software contains a wire library that simplifies the usage of the 12C bus; further information is available on the Wiring website.

vii. Programming

Arduino software is required to programme the Arduino Uno. The Arduino Uno's ATmega1280 is preburned with a bootloader that enables us to upload fresh code to it without using an external hardware programmer. It communicates only over the STK500 protocol. Additionally, you may disable the boot loader and programme the microcontroller directly through the ICSP header.

viii. Automatic (Software) Reset

In place of pressing the reset button, the Arduino Unos are meant to be reset by software running on a computer that is linked to the device. The 100 nano farad capacitor connects the ATmega1280's reset line to one of the FT232RL's hardware flow control lines (DTR). The reset line drops long enough to reset these chips when this line is asserted. For example, in the Arduino uno environment, we just hit the "upload" button in order to upload our code. In this case, the timeout for the boot loader will be reduced, since the DTR lowering and the upload start will be perfectly timed.

This arrangement has additional consequences. For each software connection to Mac or Linux, the Uno must be reset (via USB). The boot loader is executing on the Uno for the next half-second or so. The initial few bytes of data transmitted to the board when a connection is established are intercepted by the board, despite the fact that it is configured to reject faulty data. You must wait a second after the connection is opened and before sending data to a sketch running on the board if it needs one-time setup or other information when it initially begins.

The auto-reset feature of the Uno may be disabled by cutting a trace on the board. The trace may be re-enabled by soldering the pads on each side of it together. "RESET-EN" is the label on it. A 110 ohm resistor connected from 5V to the reset line may also be able to deactivate the auto-reset; check this forum topic for details.

USB over current Protection

The USB ports on your computer are guarded from shorts and overcurrent's by the Arduino Uno's resetting polyfuse. The fuse adds an additional degree of security, even though most computers have their own built-in safeguards. The USB port's fuse immediately breaks the connection if more than 500 mA is applied to the USB port until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The USB connection and power jack stretch beyond the Uno PCB's maximum length and width of 4 and 2.1 inches, respectively. The board has three screw holes for attaching it to a surface or case. Digital pins 7 and 8 have a distance of 160 mil (0.16"), which is not an even multiple of the 100 mil spacing for the other pins.

The Uno is intended to work with most Diecimila and Duemilanove shields. Analog inputs 0 to 5, AREF and GND, the power header, and the ICSP header are all in the same place as the digital pins 0 to 13. In addition, the primary UART (serial port) and external interrupts 0 and 1 are also situated on the same pins (0 and 1). (pins 2 and 3 respectively). The ICSP header on both the Uno and Duemilanove provides access to SPI.

5.1.6 Raspberry pi:

Small single-board computers called Raspberry Pis were created in Britain by a non-profit organisation called the Raspberry Pi Foundation in order to improve the teaching of fundamental computer science in schools as well as impoverished nations. Because of this, the original model is now being sold for purposes other than what it was originally designed for (such as robotics). It excludes all accessories and packaging.

The company behind the Raspberry Pi has two divisions: one for marketing and the other for development. The Raspberry Pi foundation created the first two versions. It was Ebenuphton's job to lead a company called Raspberry Pi Trading to build the B+ version of the Pi when it was launched. The technology is being developed by Raspberry Pi Trading, which is a for-profit company, while the foundation is a non-profit organization.

i. Operating systems

Depending on the board and available adapters, several operating systems may be loaded on the MicroSD, MiniSD, or SD card for the Raspberry Pi. The MicroSD slot on the Raspberry Pi 2 board can be viewed here. As well as third-party Ubuntu, Windows 10 IoT Core, RISC OS, and specialist media center versions, the Raspberry Pi foundation makes its own Raspbian Debian-based operating system available for free download. In addition to promoting python and scratch, it also provides support for a wide range of other programming languages. An unofficial open source version of the default firmware is available. The Raspberry Pi is also capable of running a wide variety of alternative operating systems. Ubuntu MATE, Windows 10 IoT Core, RISC OS, and specific editions for the Kodi media centre and classroom administration are among the other third-party operating systems accessible through the official website.

ii. Hardware

The hardware of the Raspberry pi has undergone multiple iterations, each with a different amount of memory and compatibility for various peripherals.

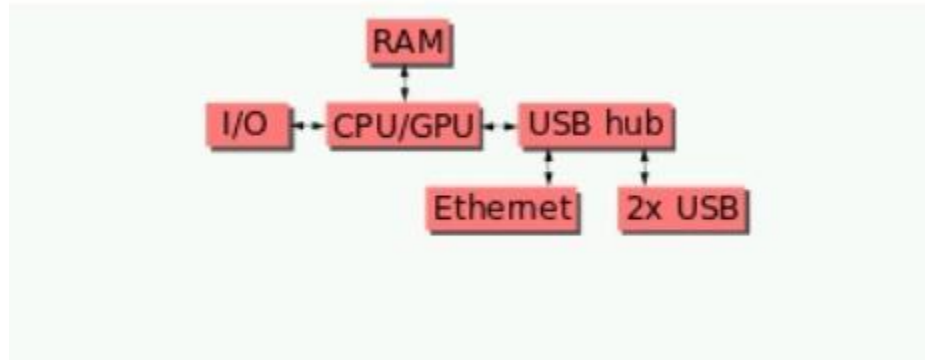


Fig 5.13 Block diagram

There are three different models of Raspberry Pi, however this block diagram only covers models B and B+. An extra USB port is linked to the Ethernet adapter. System-on-a-chip USB ports are found on the Raspberry Pi zero and its predecessors (models A,A+ and 0). When it comes to the USB/Ethernet chip, the pi 1 model B and subsequent versions include a five-point USB hub with four usable ports, whereas the pi 1 model B only has two. Using a tiny USB connector, the Pi zero's USB port is also directly linked to the SoC.

Processor

The 32-bit 900MHZ quad-core ARM Cortex-A7 CPU powers the Raspberry Pi 2B. ARM1176JZF-S CPU, video core IV graphics processing unit, and RAM are all included in the Broadcom BCM2835 SoC used in the first version Raspberry pi.



Fig 5.14 Raspberry pi

Each of the two caches has a size of 128kb, with the smaller of the two being 128kb. The GPU was the primary user of the level 2 cache.

The edge of the SoC may be seen since it is layered on top of the RAM chip. An improved version of the 1176JZ(F)-S, clocked at a higher frequency and coupled with a significantly faster graphics processor, powers the first iPhone 2. Quad-core ARM cortex-A53 processors with 512 kb of shared L2 cache power the Raspberry pi 3+'s Broadcom BCM2837B0 SoC.

iii. Performance

The real-world performance of the first-generation Raspberry Pi may be equated to 0.041 GFLOPS while running at 700 MHz. Performance on the CPU level is comparable to the 300 MHz Pentium II of 1997–1999, which was released in 1997–98. One Gpixel/s of graphics processing power or 24 GFLOPS of general purpose computing power are provided by the GPU. The graphics capabilities of the Raspberry Pi are comparable to those of the Xbox (2001).

The quad-core Cortex-A7 operating at 900 MHz and 1 GB of RAM were included in the Raspberry Pi 2 V1.1. It was said to be anywhere from four to six times as strong as its forebear. In other words, the GPU was the same as before. The Raspberry Pi 2 V1.1 might be up to 14 times quicker than the Raspberry Pi 1 Model B+ in parallelized benchmarks.

Raspberry Pi 3 is said to be 10 times faster than the Raspberry Pi 1, thanks to its quad-core ARM Cortex-A53 CPU. Task threading and instruction sets are thought to have a major role in this. The Raspberry Pi 3 was shown to be 80 percent quicker than the Raspberry Pi 2 in parallelized workloads, according to benchmark results.

iv. The seven over clock presets are:

- none; 700 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 over volting
- modest; 800 MHz ARM, 250 MHz core, 400 MHz SDRAM, 0 over volting,
- medium; 900 MHz ARM, 250 MHz core, 450 MHz SDRAM, 2 over volting,
- high; 950 MHz ARM, 250 MHz core, 450 MHz SDRAM, 6 over volting,
- turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 over volting,
- Pi 2; 1000 MHz ARM, 500 MHz core, 500 MHz SDRAM, 2 over volting,
- Pi 3; 1100 MHz ARM, 550 MHz core, 500 MHz SDRAM, 6 over volting. The CPU speed will be shown as 1200 MHz in the system information. Idling at 600 MHz reduces the speed to a more manageable level.

- After determining that 500 MHz SDRAM clocks were causing SD card damage, the maximum setting was adjusted to 600 MHz. In both high and medium modes, the core clock speed is reduced from 450 to 250 MHz simultaneously. The Raspberry Pi Zero clocks in at 1 GHz.
- There was no need for a heat sink or fan on the original and second generation Raspberry Pi boards while overclocking their CPUs, however the Raspberry Pi 3 may create more heat if it is overclocked.

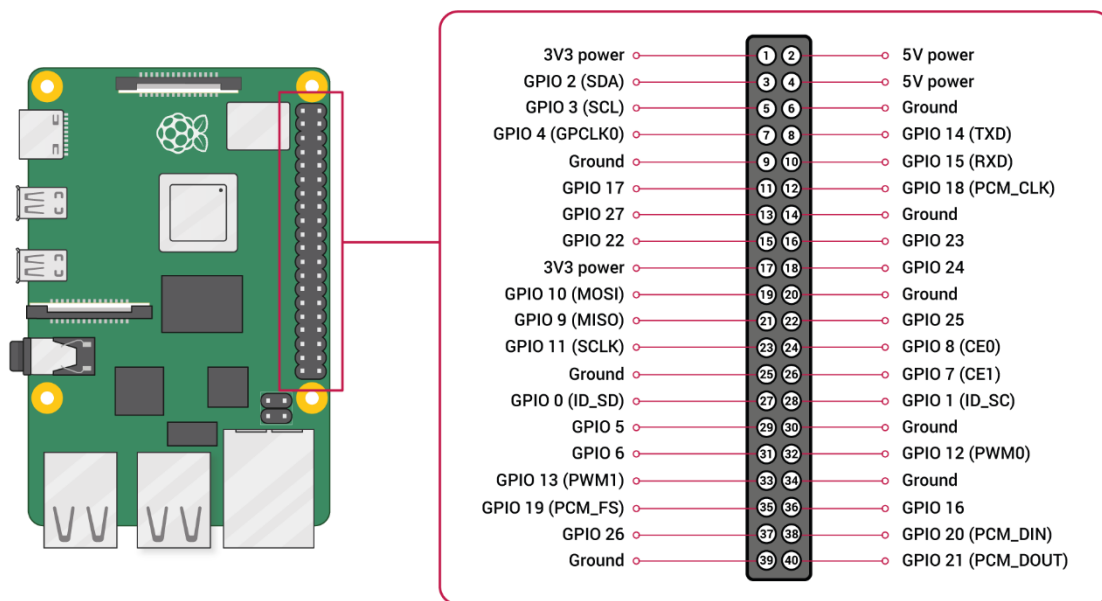


Fig 5.15 Raspberry pi pin diagram

v. RAM:

On earlier beta Model B boards, 128 MB of RAM is reserved for the GPU by default, leaving 128 MB for the Central Processing Unit (CPU). Three such splits are available on the initial 256 MB edition of Model B (and Model A). The default memory allocation was 192 MB (RAM for CPU), which should be adequate for solo 1080p video decoding or for basic 3D, but definitely not for both. 224 MB was allocated for Linux, which had a 1080p frame buffer and was thus unlikely to support any additional video or 3D. 128 MB was used for intensive 3D rendering, maybe with video decoding (e.g. XBMC). On the other hand, the Nokia 701's Broadcom Video Core IV requires 128 MB.

For the later Model B with 512 MB RAM, additional standard memory split files for 256 MB, 384 MB, and 496 MB CPU RAM (as well as 256 MB, 128 MB, and 16 MB visual RAM) are first supplied. However, about one week later, the RPF published a fresh version of `start.elf` that could read a new item in `config.txt` and dynamically allocate RAM (from 16 to 256 MB in 8 MB increments) to the GPU, therefore eliminating the need for memory divides and allowing a single `start.elf` works identically on the 256 MB and 512 MB Raspberry Pi models. The Raspberry Pi 2 and 3 each come with 1 GB of RAM. The Raspberry Pi Zero and Zero W both come equipped with 512 MB of RAM.

vi. Networking:

A USB Ethernet adapter or a Wi-Fi adapter are typically used to connect the Model A, A+, and Pi Zero to a network, since they lack Ethernet hardware. This chip, the SMSC LAN9514, is used to provide an Ethernet port on the Model B and B+. Both the Raspberry Pi Zero W (wireless) and the Raspberry Pi 3 (wireless) include 2.4GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on the Broadcom BCM43438 Full MAC chip, which has no official support for monitor mode but is enabled via unauthorized firmware patching. Gigabit Ethernet, Bluetooth 4.2, and IEEE 802.11ac WiFi are all included in this model's feature set. The USB 2.0 connection connecting the Raspberry Pi 3B+ to the SoC is capped at 300 Mbps.

Peripherals:

A USB keyboard and mouse may be used to control the Raspberry Pi. If you're looking for a solution that can be utilized with all kinds of USB-enabled devices, this is a great option. The surface of the Raspberry Pi has a variety of pins and connections that may be used to connect additional devices.



Fig 5.16 Raspberry pi top view

vii. Video

Modern HD and Full HD TV standards, as well as higher and lower monitor resolutions and older NTSC or PAL standard CRT TV resolutions, may all be generated by the video controller. When it ships, it can display at 640x350 EGA, 640x480 VGA, 800x600 SSVGA, 1024x768 XGA, 1280x720 HDTV and so on. When it is overclocked, it can display at 1024x768 XGA, 1280x720 HDTV and so on.

Up to a 2048x1152 resolution, 3840x2160 at a frame rate of 15 Hz (too low for convincing video) may work. This does not necessarily suggest that the GPU is capable of decoding video codecs at these resolutions, since the Pi is known to have issues with H.265 decoding (most common formats up to Full HD do work).

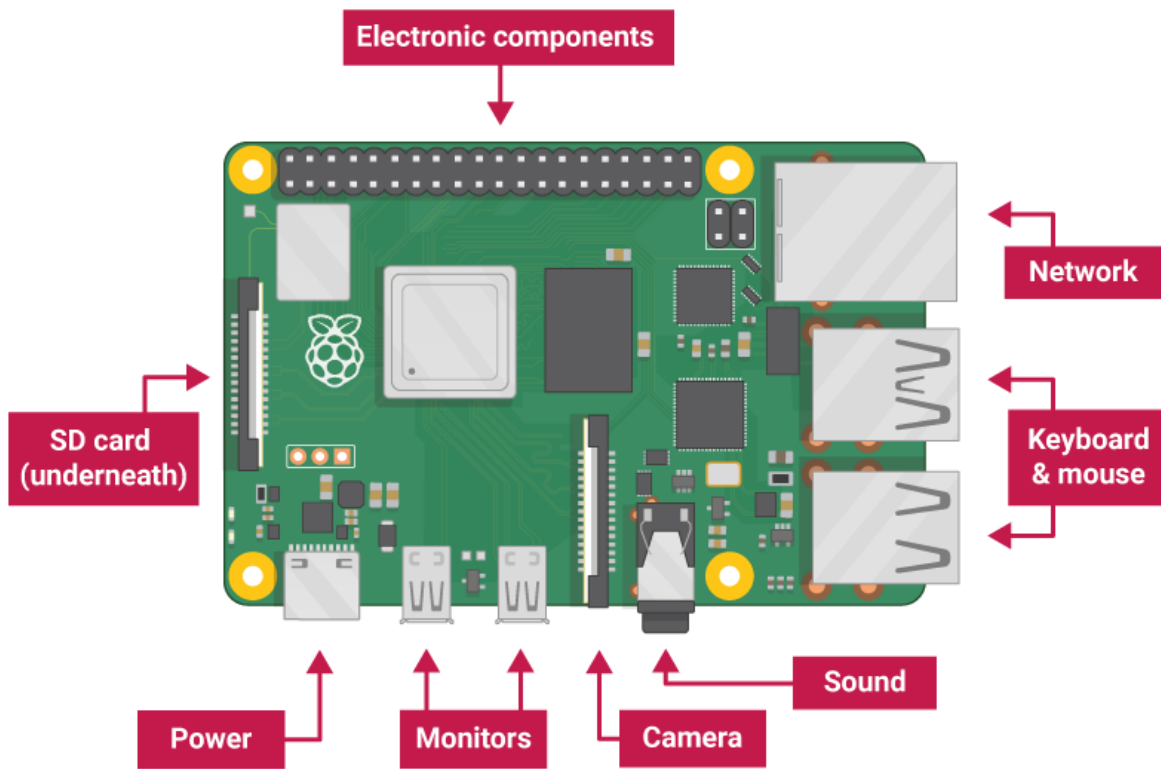


Fig 5.17 Raspberry pi labelled

viii. Real-time clock:

Because there are no built-in real-time clocks in any of the current Raspberry Pi models, they are unable to keep track of the time of day. In order to get around this problem, the time may be obtained via a network time server or from human input when the Pi is switched on. The Pi automatically stores the current system time when it shuts down and reloads that time when it starts up again.

I2C-enabled hardware clocks, such as the DS1307, may also be used to keep track of time.

ix. Operating systems

A MicroSD, MiniSD, or SD card may be used to install several operating systems for the Raspberry Pi, depending on the board and available adapters. The bottom of a Raspberry Pi 2 board has a MicroSD card slot.

There are a variety of Linux variants available for the Raspberry Pi, including Raspbian, a Debian-based operating system, as well as Ubuntu, Windows 10 IoT Core, and specialist media centre versions. Many additional programming languages are supported, although Python and Scratch are promoted as the primary ones. An unofficial open source version of the default firmware is also available.

The Raspberry Pi is capable of running a wide variety of alternative operating systems. Ubuntu MATE, Windows 10 IoT Core, and specific editions for the Kodi media centre and classroom administration are among the other third-party operating systems accessible through the



official website.

Fig 5.18 Raspberry bottom view

5.1.7 IR sensor

A broad range of wireless applications are served by infrared technology. Sensing and remote control are the two most important components. Far, medium, and close are all subcategories of infrared in the electromagnetic spectrum; each has a distinct wavelength range. These areas' spectral characteristics and potential uses are shown in the table below.

- sensors and fiber optics in the 700nm to 1400nm range of the infrared spectrum
- In the mid-infrared range, between 1400 and 3000 nm, heat detection is possible.
- Thermal imaging in the far infrared (3000 nm to 1 mm) range.



Fig 5.19 IR sensor

Infrared has a larger frequency range than microwaves but a lower frequency range than visible light. Photo optics technologies are employed in the near infrared range for optical sensing and optical communication since light is less complicated than RF when applied as a signal source. For short-range applications, IR data transfer is used for optical wireless communication. To gather information about its surroundings, an infrared sensor either produces or detects infrared light. Planck's radiation law, Stephen-Boltzmann law, and Wien's displacement law determine how an infrared sensor works.

As stated by Planck, "any object emits radiation at a temperature that is not equal to 00K.". "At all wavelengths, the total energy released by a black body is proportional to the fourth power of the absolute temperature," says Stephen - Boltzmann law. As stated by Wien's Displacement law, "the black body radiation curve at various temperatures will reach its peak at a wavelength inversely proportional to the temperature. "By its most basic level, the Infrared Sensor that is employed as an obstacle detector works by transmitting infrared signals that bounce off of an object, and these signals are received at an infrared receiver.

A standard infrared detection system consists of an infrared source, a transmission channel, an optical component, infrared detectors or receivers, and signal processing. Many kinds of infrared sources are available, including infrared lasers and LEDs. Infrared transmission occurs mostly via vacuum, environment, and optical fibers. Optical components may be used to focus infrared light or restrict its spectrum response fibers. Infrared radiation may be focused or restricted by using optical components. The infrared radiation is focused using optical lenses constructed of quartz, germanium, and silicon. Infrared receivers include photodiodes, phototransistors, and other similar devices. Photosensitivity, detectability, and noise equivalent

power are some of the most essential infrared receiver parameters. The infrared detector's output is so tiny that signal processing is done using amplifiers.

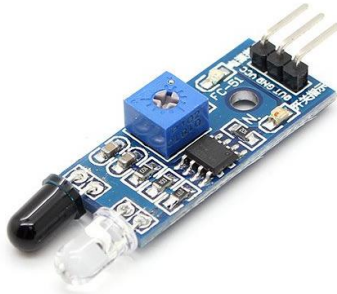


Fig 5.20 IR sensor diagonal view

i. **Types of IR Sensors**

There are two types of infrared sensors: passive and active. An infrared detector is a passive infrared sensor in a nutshell. Sensors that employ passive infrared technology do not use any infrared source, but rather detect the energy generated by obstructions in their range of vision. Quantum and thermal are the two main varieties. Thermal infrared sensors generate heat using infrared radiation and are not reliant on a particular wavelength for operation. Infrared detectors of the thermal kind include thermocouples, pyro electric sensors, and bolometers.

Thermal infrared detectors are slower and less accurate than quantum infrared detectors. Quantum type detector photosensitivity is wavelength dependent. Intrinsic and extrinsic forms of quantum detectors are further differentiated. Intrinsic quantum detectors include photoconductive and photovoltaic cell types.

The infrared source and detector are both components of active infrared sensors. LEDs and infrared laser diodes are examples of infrared sources. Photodiodes and phototransistors are examples of infrared detectors. Reflected infrared energy from an item lands on the infrared detector as a result of its emitting infrared light.

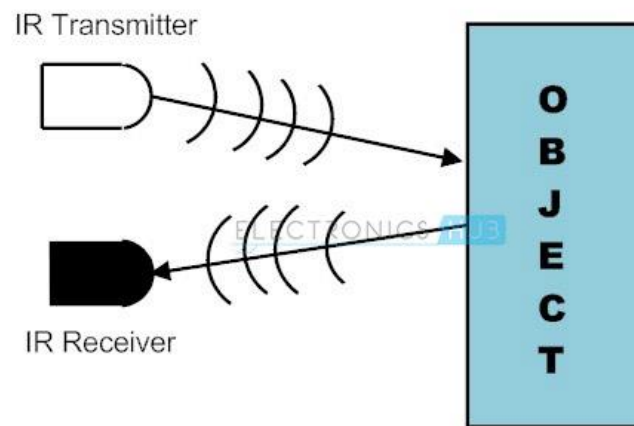


Fig 5.21 Transmitter and Receiver Block diagram

ii. IR Transmitter

Light emitting diode (LED) producing infrared radiation is the Infrared Transmitter. Infrared LEDs are the result. The radiation released by an IR LED is not visible to the human eye, despite the fact that it appears like a standard LED.

Typical images of an Infrared LED are displayed.



Fig 5.22 Led Bulb

Depending on the wavelength, output power, and reaction time, there are many kinds of infrared transmitters.

An infrared LED, a current-limiting resistor, and a power source may be used to build a basic infrared transmitter. An IR transmitter's schematic is illustrated below.

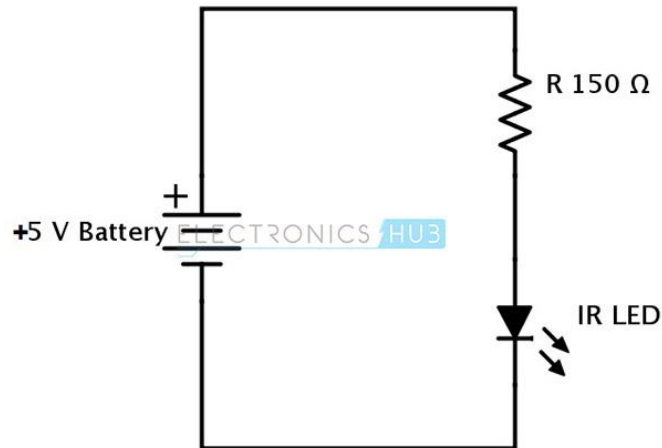


Fig 5.23 Circuit Diagram

The IR transmitter uses 3 to 5 mA of electricity when powered by a 5V source. These transmitters may be programmed to emit a certain frequency of infrared light. 'OOK (ON – OFF – KEYING) modulation is the most frequent kind of modulation. It is possible to use IR transmitters for many purposes. For certain applications, infrared heat is needed, and the best source of infrared heat is an infrared transmitter. Solar cells may be created using infrared emitters and quartz.

5.1.8 MOTOR DRIVER (L23D):

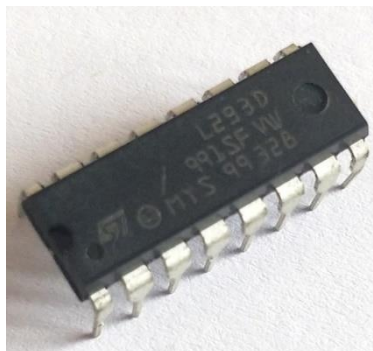


Fig 5.24 Motor driver IC

The L293D is a widely used 16-pin motor driver integrated circuit. As implied by the name, it is mostly used to power motors. A single L293D IC is capable of driving two DC motors simultaneously; moreover, the direction of these two motors may be separately regulated. Therefore, if you have motors that operate at less than 36V and less than 600mA and are to be controlled by digital circuits such as Op-Amps, 555 timers, digital gates, or even microcontrollers such as Arduino, PIC, or ARM, this IC is the correct option for you.

It's really easy to use this L293D motor driver IC. The IC operates on the Half H-Bridge concept; we will not go into detail about what H-Bridge is; suffice it to say that H bridge is a configuration that is used to drive motors in both clockwise and anticlockwise directions. As previously stated, this integrated circuit is capable of simultaneously driving two motors in either direction; the circuit for doing this is depicted in Figure 5.25.

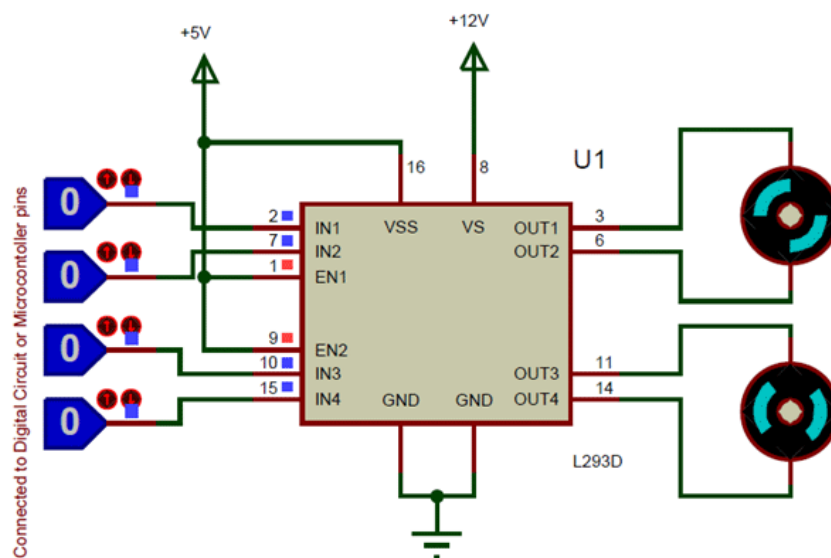


Fig 5.25 Circuit diagram

Ensure that all of the ground pins are connected. This IC has two power pins: Vss (Vcc1), which must be connected to +5V in order for it to operate. Depending on the specifications of your motor, you may connect Vs (Vcc2) from 4.5V to 36V, here I've linked it to +12V, but you can use whatever voltage you choose.

To enable the input pins for Motor 1 and Motor 2, utilize the Enable pins (Enable 1, 2, and Enable 3, 4). In most circumstances, we'll be running both motors, thus connecting them to a +5V supply keeps both pins high by default. It is controlled by input pins 1, 2 and 3 and 4 for

input pins 3, 4 for input pins 1 and 2. A microcontroller or digital circuit may be used to regulate the motor's speed and direction. To operate your motor, you may use the following table to toggle the input pins.

i. Applications

- Used to drive high current Motors using Digital Circuits
- Can be used to drive Stepper motors
- High current LED's can be driven
- Relay Driver module (Latching Relay is possible)

5.1.9 RF Transmitter Receiver Module:

Wireless Communication in any form has become an essential part of human life whether it may be short distance T.V Remote or long distance radio communication. Wireless communication is all about transmission of data wirelessly so that there is no hassle of any wires and no direct contact with the device itself One of the easiest and cheapest ways to implement wireless communication is using RF Module (Radio Frequency Module). By combining the two objects i.e. wireless communication with Arduino, we can create a wide range of applications like remote controlled cars, wirelessly operated robots, home automation, simple data transfer etc.

RF Module:

Whether it's a short-distance T.V. remote or a long-distance radio link, wireless communication has become an indispensable component of modern life. When it comes to wireless communication, there is no need for any cables or direct touch between the gadget and its data. When it comes to implementing wireless communication, RF Modules are one of the simplest and most affordable options (Radio Frequency Module). We can construct a broad variety of applications by integrating wireless communication with Arduino, such as remote-controlled autos, wirelessly operated robotics, home automation, and simple data transmission.

Transmitter

Arduino UNO with the 434 MHz Transmitter module makes up the transmitter. It's possible to utilize an external LED, however an on-board LED would be adequate in this case. Following are the specifications for the Transmitter. Four pins make up the RF Transmitter Module: the VCC and GND power supplies, as well as the data line and the antenna. The 5V and ground pins of the VCC and GND connectors are linked together. A digital input/output pin may be used to link the data pin to the Arduino's data pin. It is attached to Pin 12 in this instance. It is necessary to attach the antenna pin to a coil of wire, which is all that is required. For demonstration purposes, we are utilizing the on-board LED, although an external LED and current limiting resistor may be used.

Receiver

The 434 MHz Receiver module and Arduino UNO form the receiver's core. While an external LED and a current-limiting resistor may be utilized, an on-board LED should be enough. The receiver component is designed as follows. The RF Receiver Module has four pins: VCC, GND, Data and Antenna. The Arduino's 3.3V pin is linked to the VCC and GND pins, respectively. The Arduino's Pin 12 is linked to the data pin. The antenna pin of the 434 MHz Receiver module is attached to an antenna identical to the transmitter module. Even though it is possible to utilize an external LED in this project, the Arduino's 13th pin is used for the LED.

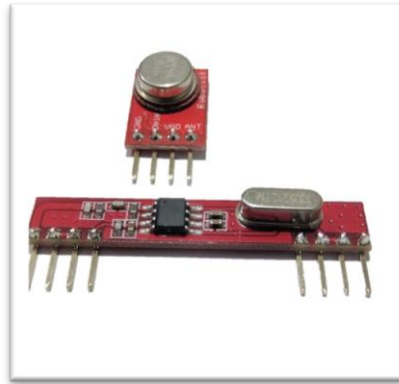


Fig 5.26 Infrared Transmitter

i. Working Process

Arduino UNO boards are used to demonstrate RF communication in this project. Using two Arduino UNO microcontroller boards, the project aims to send data between the RF Transmitter and Receiver modules. Here, the project's workings are detailed. Mike McCauley developed the Arduino library VirtualWire.h. Two Adriano's may communicate with each other using the RF Module (transmitter/receiver pair) in this communication library. Module configuration, data transmission, and data reception are all supported by this library's many methods and APIs. Two letters are all that the transmitter needs to communicate in this project; it transmits the character "1" and a few seconds later, it sends "0". The LED on the transmitting side of the project will be turned on once the "1" is transmitted. The data "1" will be received by the receiver since it is being transferred through RF transmission.

The Arduino on the receiver side of the project will turn on the LED when it gets "1." Similarly, the LED on the transmitter side is turned off when the RF transmitter transmits the value "0". Because of this, the LED on the receiver side has been switched out, and the receiver now receives "0." In this way, the receiver mimics the transmitter's behavior.

ii. Applications

RF modules may be used in a broad variety of applications, including RC Cars, Home Automation and Robotics, as stated in the project introduction.

5.1.10 Webcams:

Webcams are small digital cameras that may be attached to your computer and used to stream live video footage (as they happen). Like a digital camera, it uses an image-sensing microchip (a charge-coupled device (CCD) or, more often these days, a CMOS image sensor) to record light from the front via a small lens. When the image sensor and its electronics are put together, they turn the image in front of the camera into a string of zeros and ones that a computer can process. While digital cameras include memory chips and flash memory cards, a webcam doesn't have any of these since it is meant to collect and transfer photographs to a computer as soon as they are taken. In order to facilitate this, cameras include USB ports on the back. Using a USB connection, the computer provides power to the camera and sends digital information acquired by the webcam's image sensor back to the computer.



Fig 5.27 Webcam

When it comes to webcams, they all do the same thing: They employ an image sensor chip to capture moving pictures and transform them into numbers that are sent over the Internet. How does a webcam's image sensor chip operate, and why is it so important? To discover out, let's disassemble a webcam.

There is little more than a plastic lens linked to a very tiny electrical circuit board when you remove the webcam's outer case. You may fine-tune the camera's focus by varying the focal length of the lens used. When you remove the webcam's exterior casing, you'll see that it's nothing more than a plastic lens attached to a very small electrical circuit board. By adjusting the focal length of the lens, you may fine-tune the camera's focus.

Fig 5.28 Webcam Bottom view



The image sensor (also known as the CCD or CMOS chip) is visible when the lens is removed. It is the square component located centrally inside this circuit. Only the small, green-colored centre region of the chip is light-sensitive; the remainder of the chip is concerned with connecting the light detector to the larger circuit that surrounds it:.

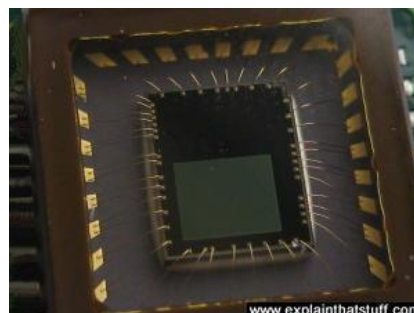


Fig 5.29 Webcam Internal view



Fig 5.30 Webcam bottom view

Webcams versus digital cameras

So the camera's "electronic eye" is its image sensor. There are millions of small, light-sensitive squares organized in a grid pattern on the chip, which is a semiconductor. They're known as pixels. Small sensors with only a few hundred thousand pixels (usually 640 x 480) are used in basic cameras. The number of megapixels (millions of pixels) used to compare cameras is a common measure of the quality of a digital camera's sensor. To put it another way, the basic webcam can only capture roughly 300,000 images per second, but a digital camera with 6 megapixels can capture almost 20 times as many images per second. A 12 megapixel camera with a 4000 x 3000 pixel sensor would be great. For the same size shot, the 12 megapixel camera will have 1000 more horizontal and vertical dots, resulting in a better quality photo with more information. The diameter of a single pixel in a high-quality sensor is around 10 micrometers (10 m).

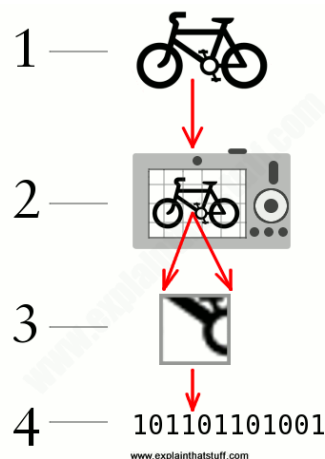


Fig 5.31 Image to binary conversion

How does an image sensor convert a picture into digital form?

To snap a picture or look through your webcam, light travels rapidly through the lens. The image sensor receives this "picture" and splits it down into individual pixels, which are then turned into numbers. The two types of image sensors, CCDs and CMOS chips, do this task in somewhat different ways. Photoelectric cells (used in items like "magic eye" intruder alarms or washroom washbasins that flip on automatically when you put your hands under the faucet) transform incoming light rays into electricity at first. When it comes to the CCD, however, it is simply an analogue optical chip that transforms light into changing electrical signals, which are then sent to one or more additional chips where they are digitalized (turned into numbers). To put it another way: With a CMOS chip, you can catch light rays and convert them into digital signals all on the same chip. As a result, it is a digital device in contrast to a CCD, which is analogue. Because they are more efficient and less expensive to produce in large quantities, CMOS chips are now used in the majority of low-cost camera phones and webcams. However, CCDs are still frequently employed in low-light astronomy and other applications.

An incoming picture is turned into a pattern of digital pixels, regardless of whether it is captured using a CMOS sensor or a CCD and accompanying equipment. Rather of calling it a CMOS sensor or a CCD, we'll simply call it "the image sensor" from now on. To begin with, the image sensor detects the amount of light that reaches each pixel in the picture area. Memory chips within the camera may store this information in the form of a numerical value. Digital photography, on the other hand, turns the image you see into a lengthy series of numbers. Each pixel in the picture is represented by a unique number, which specifies its brightness and colour.

Step by step

1. The camera's lens receives light from the subject (in this example, a bicycle).
2. There are millions of pixels in each picture taken by the camera's image sensor (squares). An LCD screen on the camera's back displays you the picture that the sensor is taking, not an image of the item viewed through a set of lenses (as with a normal

camera), but a redrawn, computerized version of the original thing presented on a screen.

3. Each pixel's color and brightness are measured by the sensor.
4. Four. Binary values (patterns of zeros and ones) are recorded on the camera's memory card for color and brightness. Your computer receives these figures quickly when you attach your camera to it.

5.1.11 ADXL335:

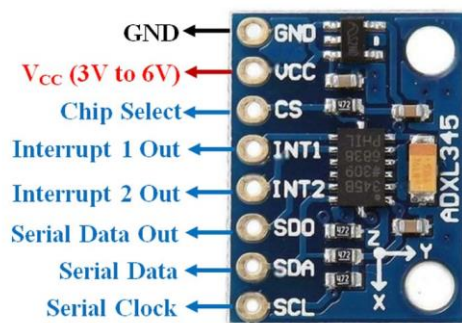


Fig 5.32 ADXL335

The ADXL345 is a three-axis MEMS accelerometer module having I2C and SPI interfaces. The Adafruit Breakout boards for those modules have on-board 3.3 volt voltage regulation and level shifting, making them ideal for use with 5 volt microcontrollers like as the Arduino. The ADXL345 is available in four different sensitivity levels ranging from +/- 2G to +/- 16G.

MEMS are an acronym for Micro Electro-Mechanical Systems.

A micro machined structure on a silicon wafer serves as the sensor's construction. The structure is sustained by polysilicon springs, which enable it to deflect smoothly in any direction when subjected to X, Y, and Z axis acceleration. Deflection results in a change in capacitance between fixed and suspended plates. The change in capacitance along each axis is translated to an output voltage proportionate to that axis's acceleration.

I2C Wiring:

The ADXL 345 Breakout has an I2C address of 0x53 and is capable of sharing the I2C bus with other I2C devices as long as each device has a unique address. I2C communication requires just four connections:

GND->GND

VIN->+5v

SDA->SDA (Analog 4 on "Classic Arduino")

SCL->SCL (Analog 5 on "Classic Arduino")

The Ada fruit breakout has level shifting & regulation circuitry ,so you can power it from 3-5V and use (3- 5)V logic level for I2c

Gravity as a Calibration Reference

Acceleration can be measured in the units of gravitational force or "G", where 1G represents the gravitational pull at the surfaces of the earth. Gravity is a relatively stable force and make convenient and reliable calibration reference for surface-dwelling earthlings.

Calibration Method:

The sensor output for each axis must be determined when it is exactly aligned with the gravitational pull axis in order to calibrate it to the gravitational reference. Laboratory quality calibration makes use of precision positioning jigs. The procedure outlined here is simple, but the effects it produces are astounding for something as simple as a block of wood.

Install Sensor:

The sensor should first be attached to a block box. All that matters is that all the corners be right angles. Material does not matter so much as long as it's reasonably stiff.

DC Motor / Direct Current Motor:

It is a dc motor if it is an electric motor that is powered by dc. DC electrical energy may be converted into mechanical energy by using this apparatus.

Principle of DC Motor

An electric current's inclination to travel is increased when the conductor is put in a magnetic field. When a mechanical field is created by the interaction of a magnetic field with an electric field. This is how a DC motor, often known as a direct current motor, operates. The term for this activity is "motoring." Using Fleming's left hand rule, which states that if the index finger, middle finger, and thumb of your left hand are extended mutually perpendicular to each other, and if the index finger represents the direction of magnetic field, middle finger indicates the direction of current, and then thumb represents the direction of force which is experienced by a shaft of a DC rotor, this motor's direction of rotation is described.

DC motors and DC generators are structurally and constructionally same, yet electrically they are completely different. Unlike a generator, here we send electrical power to the input port and get mechanical power from the output port. The graphic below illustrates how it works. To get the mechanical output (torque " T ") and speed " ω " from a DC motor, the supply voltage E and current I must be supplied to the electrical port / input port. K is a variable that connects the direct current motor's input and output port variables.

We can see from the illustration above that a motor is the polar opposite of a DC generator, and that the same equipment may perform both functions by just reversing the ports. Electrical supply voltage and current I are supplied to a DC motor's electrical port, or input ports, whereas torque T (and hence speed ω) and speed are derived from the mechanical port or output port.

This parameter, known as K , connects the direct current motor's in and out ports. The graphic above shows that a DC generator and a motor are diametrically opposed, and that the same piece of equipment may serve both purposes by just reversing the ports. In-Depth Look at a DC Motor. The DC motor is represented by the central circle. The brushes are drawn on the circle. There are two sets of external terminals that we attach to each brush. Our armature has a shaft that extends from its centre and connects to the mechanical load at the mechanical terminal. The armature resistance R_a is shown in series on the power supply terminals.

Now, apply the input voltage E to the brushes. T_g is the torque generated by the electric current flowing through the rotor armature through brushes in the presence of a magnetic field. The armature of the dc motor will revolve as a result of the torque T_g . It creates an emf E_b

in a way very similar to a generator because the armature conductors carry currents and the armature spins inside the stator magnetic field. The produced Emf E_b is known as the rear Emf because it counters the forward voltage and is directed in the opposite direction of the supply voltage. In generators, the back emf is represented by

Where, P = no of poles

ϕ = flux per pole

Z = No. of conductors

A = No. of parallel paths

N = speed of the DC Motor.

E_b is inversely proportional to the speed ' N ' in the following equation. A direct current motor generates back EMf anytime it turns because of this. The rotor speed is represented in rad/sec by this symbol. So E_b is inversely proportional to the.

As a result, when a load is applied to a motor, the motor's speed lowers. E_b grows as the voltage differential between the supply and the back emf widens. There is a rise in armature current, which causes torque and hence speed to increase as a result. When the load on the DC Motor changes, it is able to maintain the same speed.

Now I_a is a representation of the armature current.

Now speed $\omega = 0$ so at starting $E_b = 0$

With no back Emf and a low armature winding electrical resistance R_a , this motor can start with a large current. To get the DC Motor up and running, we need a starter.

There are two things happening at the same time: The back emf is being created and current is decreasing as the motor speeds up.

Direct motors are classified according to the way the field winding is connected to the armature. It is possible to divide them into three categories:

1. Shunt wound DC motor
2. Series wound DC motor
3. Compound wound DC motor

5.2 SOFTWARE REQUIREMENTS

5.2.1 Introduction to Python

Because of its high degree of versatility and interpreter foundation, Python is a popular choice among programmers of all skill levels. In 1991, Guido van Rossum created it and the Python Software Foundation developed it. Code readability was the primary goal of its development, with its syntax making it possible for programmers to convey ideas in a few lines of code. Python is a programming language that makes it possible to operate quickly, clearly, and more seamlessly across many platforms, web development (server-side), software development, mathematics and System scripting.

i.i. What is Python capable of?

- Python may be used to build web applications on a server.
- Python may be used in conjunction with other tools to construct processes.
- Database systems may be connected to using Python. It's capable of both reading and writing to files.
- Python may be used to manage large amounts of data as well as to do mathematical calculations of a sophisticated nature. When it comes to quick prototyping, Python is a great choice.



Fig 5.33 PYTHON LOGO

i. Why Python?

- Python runs on a wide range of operating systems, including Windows, Mac OS X, Linux, and the Raspberry Pi.
- The syntax of Python is quite close to that of English.
- Programs in Python may be written with less lines of code than in languages like C or Java.
- As soon as you write a line of code in Python, it is immediately available for use. This implies that python prototyping is really fast.
- Programming in Python may be done in a variety of ways, including procedural, object-oriented, and functional approaches.

Good to know

- The most recent major version is Python , which we shall be using in tutorial. However, Python 2, although not being update with anything other than security update, is still quite popular.
- As of December 2008, the most common version of Python was 2.0, which was published in 2000. Version 3.0 was then released, and it had a few very tiny but important changes that were not backward compatible with the 2.x version. Version 3.0 has now been retired. They're fairly similar, and certain features of the newer version of Python 3 have been back ported to the older version. It's still not entirely compatible in general.

- With periodic updates, both Python 2 and 3 are maintained and developed. 2.7.15 and 3.6.5 are the most latest versions available at the time of this writing. Python has been given the official End Of Life date of 01-01-2020, after which it will no longer be maintained.
- The Python community has given Guido the moniker of BDFL (Benevolent Dictator For Life), and he is still in charge of maintaining Python with the help of the Institute's core development team. Guido's favorite British comedy company Monty Python's Flying Circus, of which Guido was and probably still is a fan, is the source of Python's moniker. Python's documentation is littered with Monty Python jokes and allusions to the film.
- Many integrated development environments, such as "Thonny," "Pycharm," "Net Beans," or "Eclipse," allow you to write Python. These are especially helpful when dealing with a large number of Python files.

iv. Python Syntax compares to other programming languages

- Python was developed with mathematical influence for readability and some closeness to the English language.
- Python, unlike other programming languages, does not utilise semicolon or parentheses to conclude a command.
- Python defines scope, such as the scope of a loop, function, or class, utilising indentation and white space. This is a common application of " in other programming languages.

Python is Interpreted

- Many languages are compile-time, which means that the source code you write must be converted to machine code, the language of your computer processor, before it can be executed. Programs written in an interpreted language are sent directly to an interpreter, which executes them.
- This results in a faster development cycle, since you can just enter in code and execute it without performing an intermediary compilation phase.

- One disadvantage of interpreted languages is their slow execution speed. Programs that are compiled into the computer processor's native language often run faster than interpreted programs. This may be a constraint for certain applications that need a high level of computing, such as visual processing or severe number crunching.
- In reality, however, the difference in execution speed for the majority of applications is measured in milliseconds, Or seconds at most, and is hardly visible to a human user. Generally, the convenience of coding in an interpreted language outweighs the disadvantages in most applications.
- Despite its syntactical simplicity, Python supports the majority of high-level language constructs, including complicated dynamic data types, structure and function programs, and OOPS.
- Additionally, a large library of classes and functions is available to extend the language's capabilities, such as database processing and graphical user interface development.

5.2.2 ARDUINO IDE:

• Arduino is an open source electrical platform with simple hardware and software. Arduino boards may receive input from a light sensor, a finger on a button, or a Twitter tweet and output it as a motor, an LED, or a blog post. You can tell your board what to do by sending commands to its microcontroller. Use Arduino programming language (then Wiring) and Arduino IDE (then Processing) to do.

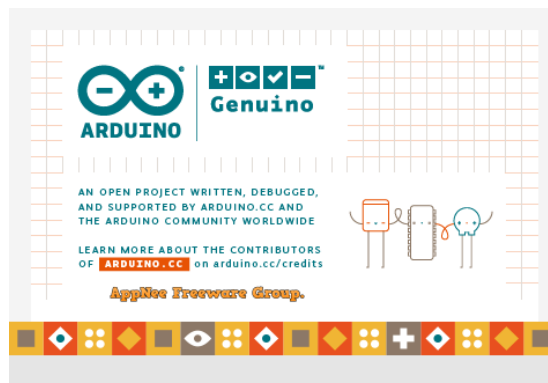


Fig 5.34 Arduino uno IDE

Arduino has grown into a worldwide community of makers, from students to hobbyists to artists, programmers, and professionals, whose contributions have added up to an incredible amount of accessible knowledge that may be of great help to novices and experts alike. This tool was created by Ivrea Interaction Design Institute for students with no prior knowledge of electronics or programming. The Arduino board has evolved from a basic 8 bit board to a device for IoT, wearable, 3D printing, and embedded nature. Arduino boards are entirely open source, allowing users to construct their own and tailor them to their needs. The programme is open source and grows with the help of users worldwide.

WRITING SKETCHES

Sketches created in the Arduino Software (IDE) are known as programmes. Text editor was used to create these sketches, which were then saved as .ino files. Cutting and pasting was possible, as was searching for and replacing text inside the editor. When saving or exporting, the messages box provides feedback and displays errors. The Arduino IDE (software) outputs a full error message and additional information to the console display. It displays the serial port number "7" in the lower right hand corner of the window. Open the serial monitor, generate, open, and save a sketch with the click of a button on the toolbar.

File

- New Create a new instance of the editor with the very minimum sketch structure already in place.
- Open Allows loading of a sketch file by navigating the computer's disc and folders.
- Recent Open Provide a concise summary of the most current sketches that are ready to be opened.
- Sketch book Displays the current sketch inside the drawing book's folder structure; clicking on any name opens the associated sketch in a new editor instance.
- Example This menu option displays any example offered by the Arduino Software (IDE) / library. All examples are organised in a tree format that enables simple navigation by topic/library.
- Exit the Arduino Software instance from whence it clicked.
- The drawing will be saved with the current name. If the file has not been named before, the "Save as..." box will suggest a name.

- **Save as...** Allows you to rename the current drawing.
- **Page Configuration** It displays the Print Page Setup window.
- **Print** Send the current drawing to the printer using the Page Setup parameters.
- **Preference** Open the window where some IDE settings, such as the language of the IDE interface, may be modified.
- **Quit** All IDE windows should be closed. The same sketch that was open when Quit was selected will be reopened automatically the next time you start the IDE.

Edit

- **Redo or Undo** Restore one or more editing steps; once restored, you may proceed using Redo.
- **Cut** Copy chosen text from the editor to the clipboard.
- **Copy** Duplicates text chosen in the editor and saves it to the clipboard.
- **Forum Copy** Copy the sketch's code to the clipboard in a format acceptable for posting to the forum, including the whole syntax colour.
- **Save as HTML** Copy the sketch's code to the clipboard as HTML, which is ideal for embedding in a web page.
- **Paste** the contents of the clipboard in the editor's cursor position.
- **Choose All** Select and highlight the whole editor's content.
- **Add or Remove a Comment** Adds or removes the comment marker at the start of each chosen line.
- **Alternately Increase or Decrease Indent** Add/delete a space at the start of each selected line, repositioning the text one space to the right / omitting a space at the start.
- **Find** Open the Find & replace window, from which you may define text to search for inside the current drawing based on a variety of criteria.
- **Locate the Following** The next occurrence of any string specified as a search item in the Find window is highlighted relative to the cursor point.
- **Find Previous** Highlights the previous occurrence of any string supplied in the Find window as a search item relative to the cursor point.

Sketch

- **Verify or Compile** Verifies and compiles the sketch; it will provide memory utilization for the code and variables in the console area.
- **Compiles and uploads** binary files to the configure board over the configure Port.
- **Upload Programmer** will overwrite the boot loader on the board; you will need to restore it using Tools > Burn Boot loader to be able to upload to the USB serial port again. Nonetheless, it enables you to use the whole capacity of the Flash memory for

sketching. Please keep in mind that this instruction DOES NOT burn fuses. To do so, use the Tools -> Burn Boot loader command.

- Export Binary Compiled Produces a '.hex' file that may be archived or transmitted to the board through other tools.
- Display Sketch Folder Open the current sketch directory.
- Incorporate a library Include the library in your drawing by using #include lines at the beginning of your code. For further information, please visit the library section below. Additionally, you may access the Library Manager from this menu item and import new libraries from '.zip' files.
- Add File a supplementary file to the drawing 'it will be copied from its current place data
- Subdirectory of sketch for materials such as documentation. The document's content data folders are not compiled, and hence are not included in the sketch programme.

a) **Tools**

- Auto Format this formats your code nicely: i.e. indents that open & close '{}' line up, and that the statements inside {} are indented more.
- Archive Sketch Archive a copy of current sketch on .zip format. The archive is place in same directory as sketch.
- Fix Encoding and Reload Fixe possible discrepancy between the editor char map encode & other OS char maps.
- Serial Monitor Open the serial monitor window and initiate the exchange of data with any connected board on currently selected Port. This usually reset the board, if the board support Reset over serial port opens.
- Board Selected that you're using. See below for description of the various boards.
- Port this menu contain all serial devices 'real or virtual' on your machine. It should be automatically refresh every time you open top-level tools menu.
- Programmer for selecting hardware programmer when programming a board / chip & not using the onboard USB serial connection. Normally you won't need this but if you are burning a boot loader to new microcontroller, you will use this. Burn Boot loader items in this menu allow you to burn a boot loader on to microcontroller on Arduino board. This does not required for normal use of Arduino board but it is useful if you purchase new AT mega microcontroller 'which normally come without boot loader'. Ensure that you have select the correct board from the Board menu before burning the boot loader on target board. This command also set right fuses.

b) Help

This section makes it simple to view a variety of documents that come with the Arduino Software (IDE). You have authorization to download and print Get Started, Reference, this guide to the IDE, and other documents locally, even if you do not have an online connection. The document is a local duplicate of the online version and may include a link to the online page. Find in Reference is the Help menu's lone interactive feature: it selects the appropriate page in the local copy of Reference for the function or command now under the cursor.

Sketchbook

The Arduino IDE (Integrated Development Environment) makes use of the sketchbook concept: a standardised location for storing your programmes or drawings. The drawing in your sketch book may be accessed through the File > Sketchbook menu item or using the toolbar's Open button. When you run the Arduino programme for the first time, it will automatically create a directory for your sketch book. The Preferences dialogue allows you to examine or alter the location of the sketchbook.

Files are stored with the .ino file extension beginning with version 1.0. Earlier versions used the .pde file extension. You may continue to open files with the extension .pde in version 1.0; the programme will automatically rename the file to .ino.

c) Tabs, Multiple Files, and Compilation

Allows you to handle sketches that include many files, each of which shows in its own tab. These might be standard Arduino code files without any noticeable extensions. c, and c++ files cp, files with headers h. Prior to compiling the sketch, all of the sketch's usual Arduino code files '.ino,.pde' are concatenated into a single file in the order in which the tabs are shown.

Uploading

Prior to submitting your drawing, make sure that the Tools > Board and Tools > Port menus are set to the right values. The boards are detailed in further detail below. On a Mac, the serial port is generally /dev/tty.usbmodem241 (for a UNO, Mega2560, or Leonardo), /dev/tty.usbserial-1B1 (for a Due over or older USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected using a Key span USB-to-Serial converter). On Windows, it's most likely COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or above (for a USB board) - you may determine this by searching for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/tty ACMX; on Windows, it should be /dev/tty USB. Once

the serial port and board have been selected correctly, click the upload button in the toolbar or pick the Upload item from the Sketch menu. The current Arduino board is immediately reset and the upload process begins. With earlier boards (pre-Die Cimila), which lack auto-reset, you must push the reset button on the board shortly before uploading. On the majority of boards, the RX and TX LEDs will flicker throughout the upload process. When the upload is complete, the Arduino Software (IDE) will display a message indicating success or an error.

When you supply a sketch, you are using the Arduino boot loader, a short software that has been put into the board's microcontroller. It enables the upload of code without the need for extra hardware. When the board resets, the boot loader remains active for a few seconds before starting the most recent sketch uploaded to the microcontroller. When the boot loader starts, it blinks the on-board (pin 13) LED (i.e. when board reset).

Libraries

Drawing libraries give additional functionality, such as working with hardware or manipulating data, for use in the sketch. Simply click on Sketch > Import Library to import the library. In order to build the library with your sketch, you'll need to include one or more `#include` declarations to the start of your sketch. Because your drawing includes a library, it takes up a lot of room on the board. Simply remove the library's `#include` declarations from the top of your code if you no longer require it.

The reference includes a list of available libraries. The Arduino software comes with a number of libraries preinstalled. Various sources including the Library Manager may be used to get more copies. You may now import a library from a zip file and use it in an open drawing in IDE version 1.0.5. This third-party library may be installed by following the instructions in the following section. See this guide on how to create your own library.

Third-Party Hardware

Add a third-party device to the sketchbook's hardware directory. Board definitions, core libraries, boot loaders, and programmer definitions may all be included in the platform installation. The third-party platform must be unzipped into its own directory once the hardware directory is created. (If you call the subdirectory "Arduino," you will override the Arduino platform that is already installed.) Simply delete this directory to remove the software.

i. **Serial Monitor**

In this example, the Arduino board's serial output is shown through the USB / serial port. Using the "send" button or pressing enter, you may transmit data to the board. Using the drop-down option, choose the board rate that matches the rate that was supplied to Serial in your sketch. Note that if you attach the board to a serial monitor on Windows, Mac OS X, or Linux, it will restart your sketch. Serial Monitor does not process control characters; if your sketch demands total control over serial communication, you may use an external terminal software that connects to the COM port allocated to your Arduino board and processes control characters.

v. Preferences:

Configure them in the Preferences dialogue box (found under the Arduino menu on the Mac, or File on Windows & Linux). All of the remainder is in the preferences file, and its location is shown in the preference dialogue.

5.2.3 Linux OS

Linux® is a free and open-source operating system (OS). Hardware resources such as CPU, memory, and storage may be managed directly by an operating system (OS), which is a piece of software. The Operating System (OS) lies between your programs and your hardware and connects them all together.

5.2.3 Linux OS

Linux® is a free and open-source os (OS). Managing a computer's hardware and resources, such as the CPU, memory and storage, requires an operating system. Connecting all your programs to each other and to your computer hardware is done by an operating system (OS).

i. How does Linux work?

Consider OS to be a motor car engine. An engine can operate on its own, but when it's coupled to gearbox, axles, and wheels, it becomes a fully working vehicle. The remainder of the vehicle will not function if the engine malfunctions. As UNIX developed, so did Linux,

which now runs on everything from mobile phones to supercomputers. In every Linux-based OS, there is a Linux kernel that manages hardware resources and a collection of software packages that form the operating system's framework.

The GNU tool is one of several common basic components of the OS. The resources given by the kernel may be managed using these tools, as well as extra applications installed and performance, security, and other parameters customized. The operating system as a whole is made up of these tools. It's possible to have a different flavour of Linux using a different mix of open-source applications.

ii. What's a command line?

The command line is a computer's only means of communication with the user. In this mode, you may instruct software to perform hardware functions that aren't possible using a standard point-and-click graphical user interface (GUI). More commercial and open source operating systems have command-line interfaces accessible. Command lines and open source software, on the other hand, tend to be linked with Linux since they both provide users full control over their machine. It provides consoles that combine these capabilities into easy-to-handle modules that live outside of the command line in our most recent release of Red Hat® Enterprise Linux.

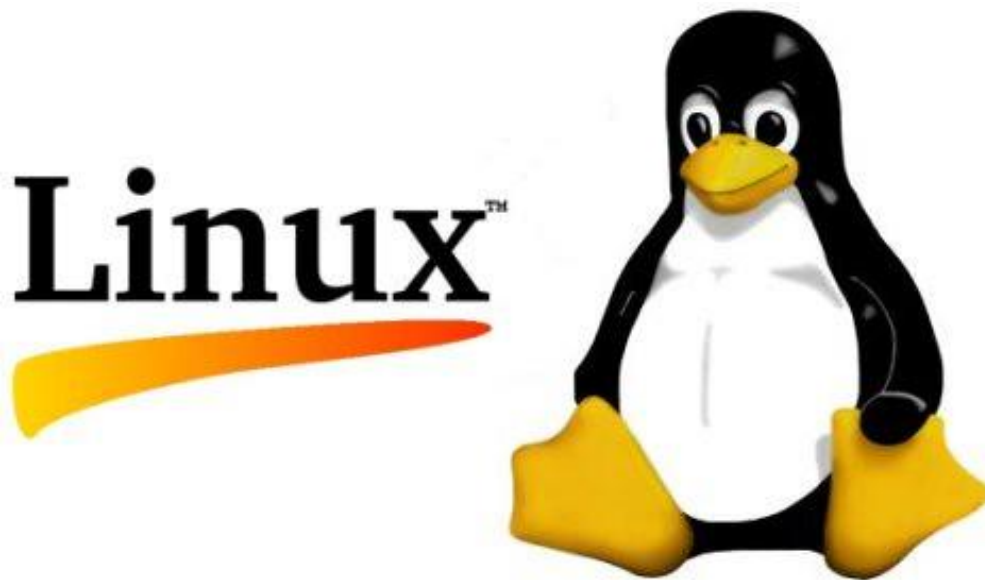


Fig 5.35 LINUX LOGO

iii. What does Linux include?

a .Kernel

The operating system (OS) is the most important component. Software and hardware interact via the kernel. It's in charge of memory, processes, and files.

b. System user space

System-level functions such as configuring and installing software are handled by this layer. This includes the background processes of the desktop environment's shell/command line/daemons.

iv. Application

You may do a job with this sort of software. Everything from desktop utilities and programming languages to multi-user business suites may be found in apps. This feature is common in most Linux distributions, where users may search for and download new software.

Linux and open source

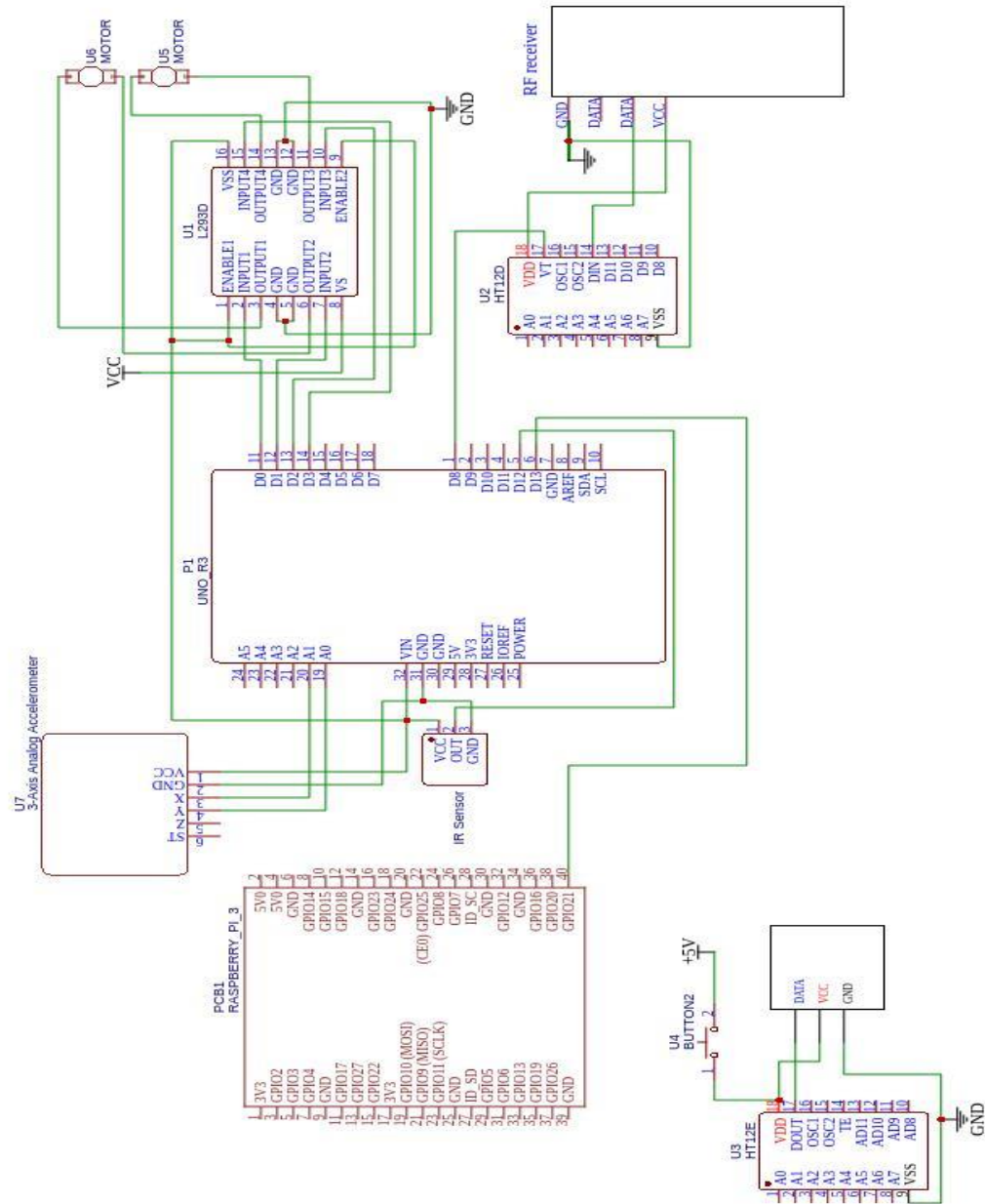
Licensed under the GNU General Public License (GPL), Linux is a free, open-source operating system. Even if you don't have permission to do so, you may run it, study it, alter it, and distribute it under the same licence. The Linux operating system is the world's most widely used free and open-source software. As a result of their efforts, professional and amateur programmers from all over the globe help to improve the Linux kernel by introducing new features as well as detecting and correcting bugs and security problems.

Top 20 Linux Applications

- ✓ Mozilla Firefox
- ✓ Thunderbird
- ✓ LibreOffice
- ✓ VLC Media Player
- ✓ Shortcut
- ✓ GIMP
- ✓ Audacity
- ✓ Visual Studio Code
- ✓ VirtualBox
- ✓ ClamAV

CHAPTER 6

CIRCUIT DIAGRAM



CHAPTER 7

CONSTRUCTION AND WORKING

CONSTRUCTION:

Raspberry pi is used to identify a person from a camera picture in this application. USB is used to connect the camera to the Raspberry Pi. The GPIO pins 11, 12, 13, and 14 are used to link the Arduino Uno and the L293D motor driver input ports. DC motors are linked to the output pins of the L293D. Any time the IR sensor picks up the speed breaker on pin number 5, it will send out a signal. Using pin 1 of the arduino, the decoder (HT12D) is linked to the Arduino. Decoders receive signals from RF receivers via a data pin. Analog pins A0 and A1 link the accelerometer ADXL335 to the Arduino.

WORKING:

The L293D motor driver is used by the Arduino to control the four-wheel robot. L293D receives a signal from an IR sensor when the speed breaker arduino detects the speed breaker arduino. The Raspberry Pi is employed as a slave CPU for the CNN algorithm, which is used to identify a person. Raspberry Pi uses GPIO pins to send the signal to Arduino if it recognises a certain individual. The arduino then shuts off the DC motor. ' Radio-frequency (RF) transmitters are installed at bus stops, and as a bus approaches the stop, the RF receiver on board receives a signal from the transmitter. A stop signal was sent to the motor driver by the arduino.

CHAPTER 8

SOFTWARE USED

8.1 RASPBERRY PI CODE

```
from imutils.video import VideoStream

from imutils.video import FPS

import numpy as np

import argparse

import imutils

import time

import cv2


Buzzer=23

#Ultrasonic sensor setting

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

PIN_TRIGGER = 7

PIN_ECHO = 11

GPIO.setup(PIN_TRIGGER, GPIO.OUT)

GPIO.setup(Buzzer, GPIO.OUT)

GPIO.setup(PIN_ECHO, GPIO.IN)

# construct the argument parse and parse the arguments
```

```

COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

a="MobileNetSSD_deploy.prototxt.txt"

b="MobileNetSSD_deploy.caffemodel"

# load our serialized model from disk

print("[INFO] loading model...")

net = cv2.dnn.readNetFromCaffe(a, b)

# initialize the video stream, allow the cammera sensor to warmup,

# and initialize the FPS counter

print("[INFO] starting video stream...")

vs = VideoStream(src=0).start()

# vs = VideoStream(usePiCamera=True).start()

time.sleep(2.0)

fps = FPS().start()


# loop over the frames from the video stream
while True:

    # grab the frame from the threaded video stream and resize it

    # to have a maximum width of 400 pixels

    frame = vs.read()

    frame = imutils.resize(frame, width=400)


    # grab the frame dimensions and convert it to a blob

```

```

(h, w) = frame.shape[:2]

blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
    0.007843, (300, 300), 127.5)

# pass the blob through the network and obtain the detections and
# predictions
net.setInput(blob)

detections = net.forward()

# loop over the detections
for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with
    # the prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the `confidence` is
    # greater than the minimum confidence
    if confidence > 0.8:
        # extract the index of the class label from the
        # `detections`, then compute the (x, y)-coordinates of
        # the bounding box for the object
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

```

```

# draw the prediction on the frame

label = "{: {:.2f}%".format(CLASSES[idx],
                             confidence * 100)

if(CLASSES[idx]=="person"):
    cv2.rectangle(frame, (startX, startY), (endX, endY),
                  COLORS[idx], 2)
y = startY - 15 if startY - 15 > 15 else startY + 15

    cv2.putText(frame, label, (startX, y),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

cv2.imshow("Frame", frame)

print(CLASSES[idx])

# show the output frame

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop

if key == ord("q"):

    break

```

```
# update the FPS counter

fps.update()

# stop the timer and display FPS information

fps.stop()

#print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))

#print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# do a bit of cleanup

cv2.destroyAllWindows()

vs.stop()

GPIO.cleanup()
```

8.2 ARDUINO UNO CODE

```
const int IN1 = 2;

const int IN2 = 3;

int PersonSen = 4;

int SpeedSen = 5;

int BusPin = 6;

int analogpin1=9;

void setup()

{

    Serial.begin(9600);

    pinMode(IN1, OUTPUT);

    pinMode(IN2, OUTPUT);

    pinMode(analogpin1, OUTPUT);

    pinMode (PersonSen, INPUT);

    pinMode (SpeedSen, INPUT);

    pinMode (BusPin, INPUT);

}

void loop()
```

```

{

int Speed=digitalRead(SpeedSen);

int Person=digitalRead(PersonSen);

int Bus=digitalRead(BusPin);

Serial.println(Speed);

Serial.println(Person);

Serial.println(Bus);

if (Speed == 1) {

    analogWrite(analogpin1, 100);

    Serial.println("Speed Breaker Detected...");

}

if((Person==0) || (Bus==0))

{

    analogWrite(analogpin1, 0);

    //Serial.println("Alert:presence of personals");

}

else

{

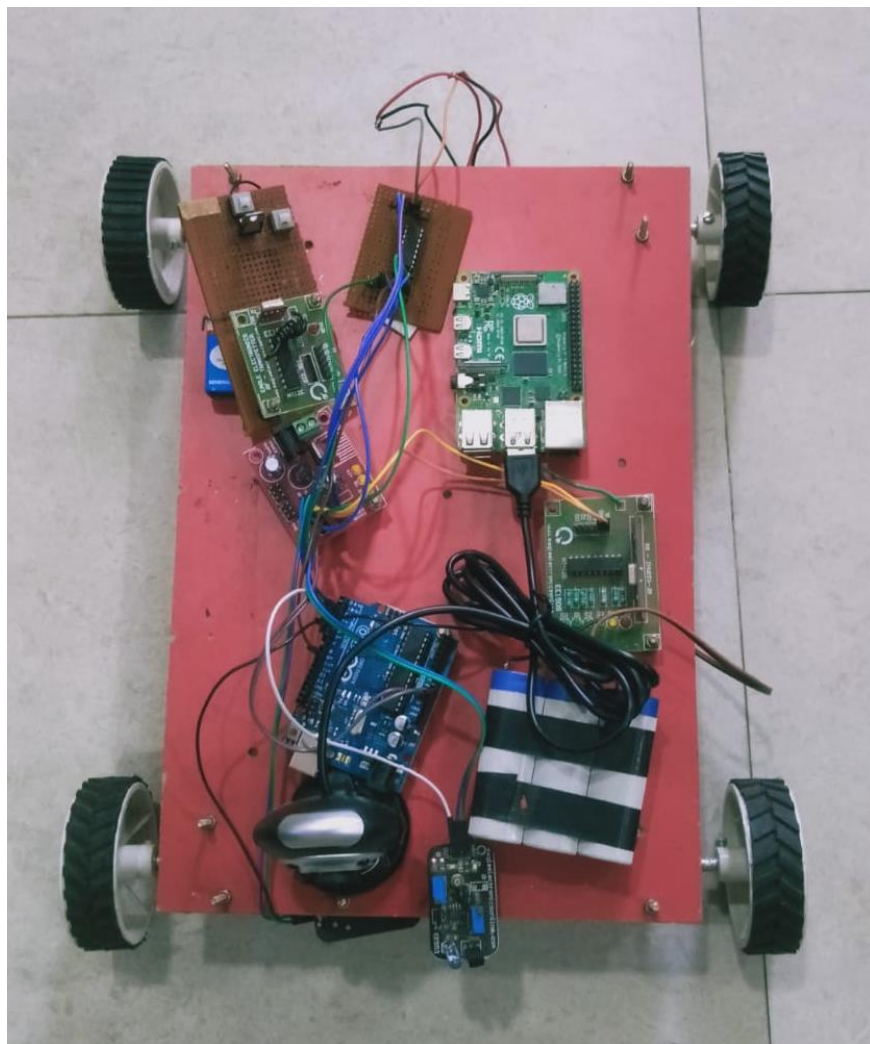
```

```
    analogWrite(analogpin1, 240);  
  
}  
  
digitalWrite(IN1,HIGH);  
  
digitalWrite(IN2,LOW);  
  
delay(1000);  
  
}
```

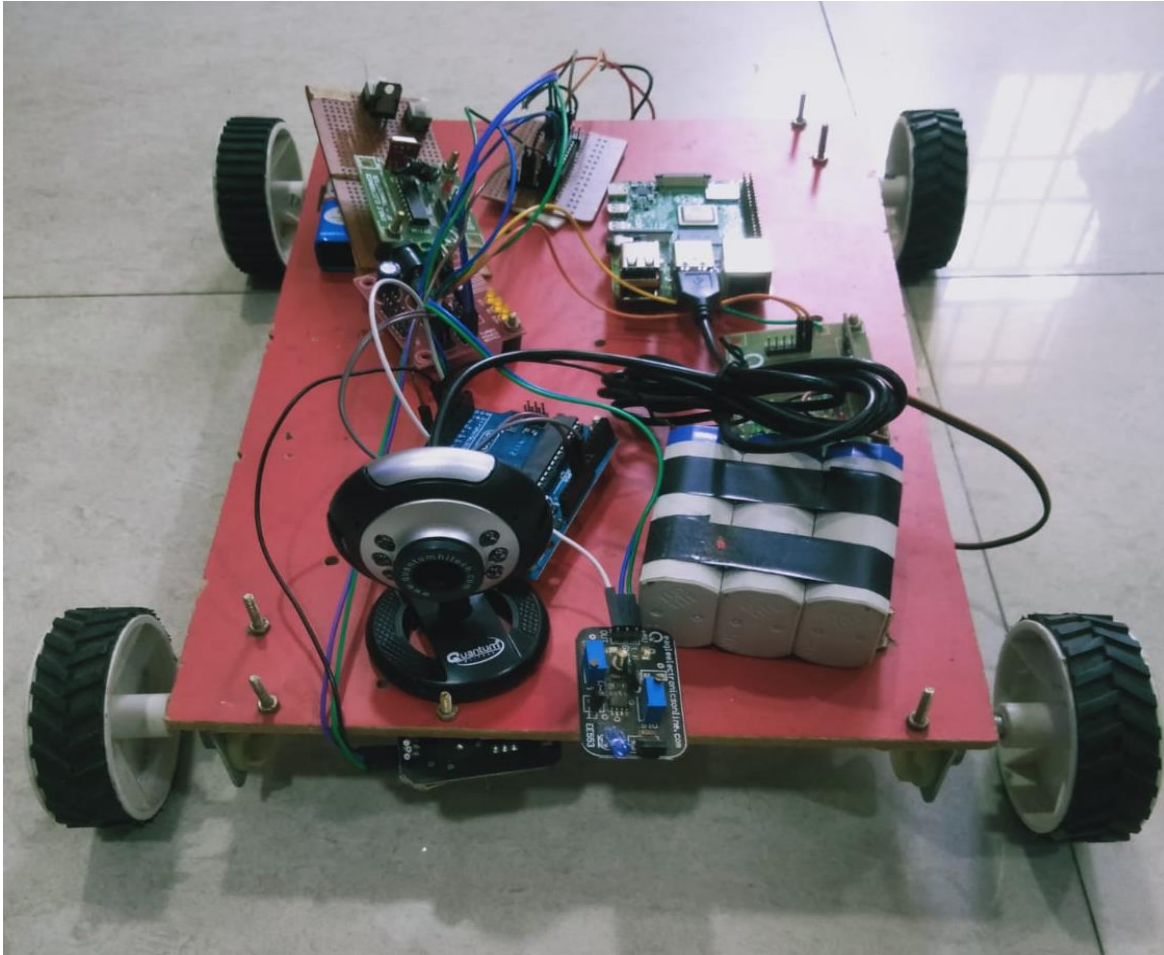
CHAPTER 9

OUTPUT & HARDWARE VIEW

The robotic module's output will directly manage the vehicle at the bus stop and speed controller. Then, the RF communication system may be utilised to operate the bus unit from the bus stop. It may be utilised to transmit the radio frequency signal to the bus unit and regulate the motor.



HARDWARE VIEW



CHAPTER 10

FUTURE SCOPE

- This initiative is very beneficial for future bus transit, as it will enable people to travel comfortably and securely.
- This concept makes use of a speed break controller; this controller helps to minimise vehicle damage and the occurrence of accidents.
- This effort will help to prevent highway accidents. This will result in a reduction of accidents in metropolitan areas.
- Both IoT and AI are critical components of accident detection and prevention strategies. We can use this idea efficiently in the future by using IoT and AI. It is economical.

CHAPTER 11

CONCLUSION

Approximately 60% of individuals go by bus. However, the bus will not stop at the majority of bus stops. As a result, the residents there will endure considerable hardship in order to catch a bus in an emergency situation. To prevent this, we are equipping all vehicles with infrared technology, which ensures that every bus will come to a complete stop at every bus stop.

Accidents are more common in speed breaker lanes. To prevent speed breaker accidents, we use a TSOP sensor in the speed breaker, which slows the vehicle automatically.

When an accident occurs, a notification is transmitted to the control room and to the appropriate person through IOT. When the bus detects a passenger in front of it, it will immediately stop.

As a result of our initiative, the majority of traffic accidents may be averted, hence saving the majority of lives.

REFERENCE

- 1)AFUKAAR , F. Speed control in developing countries : Issues ,challenges and opportunities in reducing road traffic injuries. Injury Control and safety Promotion 10,2(2003),77-81.
- 2)ANGAMI,T .Illegal speed breakers in Nagaland .The Morung Express.
- 3)ASLAN, S., KARCIOGLU , O., KATIRCI, Y ., KANDI, H., EZIRMIK, N., AND BILIR,O. Speed bump induced spinal column injury.
The American Journal of Emergency Medicine 23,4(2005),563-564.
- 4)BHORASKAR ,R., VANKADHARA ,N ., RAMAN , B., AND KULKARNI, P.Wolverine : Traffic and road condition estimation using smartphone sensors. In Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on (jan. 2012) , pp.1-6.
- 5)BOWREY ,D ., THOMAS , R., EVANS , R., AND RICHMOND ,P . Road humps: accident prevention or hazard ? Journal of accidents & emergency medicine 13,4 (07 1996).
- 6)DIGHE,S.Symbiosis student killed in mishap.Daily News and Analysis(03 2012)
- 7)ERIKSSON.J,GIRED.L,HULL.B,NEWTON.R,MADDEN.S AND BALAKRISHNAN.H. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In The Sixth Annual International conference on Mobile
Systems,Applications and Services(MobiSys 2008) (Breckenridge, U.S.A, June 2008)
- 8)FRANCIS K.AFUKAAR,J.D-D.Evaluation of speedhumpson pedestrian injuries in ghana. Injury Prevention 16,1 (2010)
- 9)GENI BRAFMAN BAHAR,I.T.E.C,Guidelines for the design and application of speed humps. Institute of Transportation Enginners(ITE),Washington,DC(2007)
- 10)HABIB, Y.Punjab government flouts the orders on illegal speed breakers.Pakiistan Today(03 2010).