PRACTICAL TRAINING REPORT

AT

BHARAT SANCHAR NIGAM LIMITED

PATNA

(5th JAN 2023-4th FEB 2023)

BSNL

Connecting India

Submitted in the partial fulfilment of bachelor of technology department

Of

ELECTRONICS AND COMMUNICATION ENGINEERING

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

NSIT

SUBMITTED BY

NAME:- SAUMYA KAUSHIK

COURSE:-B.TECH

ROLL NO:-202002

REG NO:-20104103001

Alok Kumar 21/07/23.

Scanned with ACE Scanner

भारत संचार निगम लिमिटेड

(भारत सरकार का उपक्रम)

BHARAT SANCHAR NIGAM LIMITED

(A Govt. of India Enterprises)

Training Cell O/o The Principal General Manager, Patna Telecom District, Patna.

No.: PTD/ Vocational Training/ Cert/ 2022-23/103 Date: 04-02-2023.

This is to certify that ~~Mr.~~/Ms SAUMYA KAUSHIK

S/~~o~~ D/o/ Sri/ ~~Smt~~ MUKESH PANDEY

Student of (College) NETAJI SUBHAS INSTITUTE OF TECHNOLOGY. AMHARA BLHTA, PATNA has undergone 02 (Two) /04 (Four) /06 (Six) Weeks in-plant Vocational Training for B.TECH (E.C.E) Student w.e.f 05.02.2023 at BSNL Patna Telecom District, Patna.

Wishing every success in your career

Asst. General Manager (Admn)
O/o The PGMTD, Patna

For any query contact : Training cell, Patna Telephones.
Ph: 0612-2506096, 2506575
Email : cmsalesptd@gmail.com

Scanned with ACE Scanner

PRACTICAL TRAINING REPORT

AT

BHARAT SANCHAR NIGAM LIMITED
PATNA

(5th JAN 2023-4TH FEB 2023)

BSNL
Connecting India

Submitted in the partial fulfilment of bachelor of technology department

Of

ELECTRONICS AND COMMUNICATION ENGINEERING

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

SUBMITTED BY

NAME:- BASUDEV

COURSE:-B.TECH

ROLL NO:-202003

REG NO:-20104103002

Alok Kumar 21/07/23

Scanned with ACE Scanner

भारत संचार निगम लिमिटेड

(भारत सरकार का उपक्रम)

BHARAT SANCHAR NIGAM LIMITED

(A Govt. of India Enterprises)

Training Cell O/o The Principal General Manager, Patna Telecom District, Patna.

No.: PTD/ Vocational Training/ Cert/ 2022-23/101 Date: 04.02.2023

This is to certify that Mr. Ms BASUDEV

S/o D/o Sri. Smt MANOJ KUMAR

Student of (College) NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, AMHARA, BIHTA, PATNA has undergone 02 (Two) /04 (Four) /06 (Six) Weeks In-plant Vocational Training for B.TECH (E.C.E) Student w.e.f 05.01.2023 at BSNL Patna Telecom District, Patna.

Wishing every success in your career

U.P. Singh 7/2/23

Asst. General Manager (Mktg.)
O/o The PGMTD, Patna

...Telephones.

Scanned with ACE Scanner

PRACTICAL TRAINING REPORT

AT

BHARAT SANCHAR NIGAM LIMITED
PATNA

(5th JAN 2023-4TH FEB 2023)

BSNL
Connecting India

Submitted in the partial fulfilment of bachelor of technology department

Of

ELECTRONICS AND COMMUNICATION ENGINEERING

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

NSIT

SUBMITTED BY

NAME:- TRIPTI RAJ

COURSE:-B.TECH

ROLL NO:-202006

REG NO:-20104103004

Alok Kumar 21/07/23

भारत संचार निगम लिमिटेड
(भारत सरकार का उपक्रम)
BHARAT SANCHAR NIGAM LIMITED
(A Govt. of India Enterprises)

Training Cell O/o The Principal General Manager, Patna Telecom District, Patna.

No.: PTD/ Vocational Training/ Cert/ 2022-23/102. **Date:** 04.02.2023

This is to certify that Mr./Ms TRIPTI RAJ

S/o D/o/ Sri/ Smt SUBODH SINGH

Student of (College) NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, AMHARA, BIHTA, PATNA **has undergone 02 (Two) /04 (Four) /06 (Six) Weeks in-plant Vocational Training for** B.TECH (E.C.E) **Student w.e.f** 05.01.2023.

at BSNL Patna Telecom District, Patna.

Wishing every success in your career

U.P. Singh 7/2/23

Asst. General Manager (Mktg.)
O/o The PGMTD, Patna

For any query contact : Training cell, Patna Telephones.
Ph: 0612-2506096, 2506575
Email : cmsalesptd@gmail.com

BSNL

Connecting India

ACKNOWLEDGEMENT

It is with profound gratitude that I express my deep indebtedness to all the employees of B.S.N.L. Without their support and guidance it would not have been possible for this training to have materialized and taken a concrete shape. I owe my personal thanks to my trainers in charge — for undergoing training at a reputed company like B.S.N.L.

I am also indebted to my professors and friends for their constant encouragement and for helping me in my endeavor. Last, but not the least, I would like to thank everyone who has contributed for the successful completion of my training.

PREFACE

Organizations are made up of people and function through people. Without people, organizations cannot exist. The resources of men, money, material, machinery, and mechanism are connected, coordinated, and utilized through people. Engineers need to concentrate more on mechanisms and the way in which things have been made. The need for training arises for doing things yourself, understanding its way.

Practical exposure for doing things makes a person conversant to the technicalities involved in any job. In view of such benefits, imparting of vocational training has been made an integral part of any academic structure.

In B.S.N.L., training is given to Engineering Aspirants to secure future in the dynamic world of telecommunications. Today telecommunication industry is one of the very fastest growing industries in the world.

In this order I have taken 28 days BSNL training. In my report I try to introduce the Concepts of OFC (optical fiber communication) B.S.N.L employs use of OFC for transmission purposes.

ABOUT BSNL

BHARAT SANCHAR NIGAM LIMITED (BSNL) was incorporated on 15th September 2000. It took over the business of providing of telecom services and network management from the erstwhile Central Government Departments of Telecom Services (DTS) and Telecom Operations (DTO), with effect from 1st October' 2000 on going concern basis. The company provides telecom services throughout the country excluding Delhi & Mumbai. BSNL is a 100% Govt. of India owned Public Sector Undertaking with an authorized share capital of Rs. 40,000 crore and paid up capital of Rs. 12,500 crores comprising of Rs. 5,000 crores of Equity and Rs.

7,500 crores of Preference shares capital. Its total income during FY 2020-21 was Rs.18, 595 crores.

BSNL is a technology-oriented integrated telecom service providing company which provides complete bouquet of telecom services:

- Wire line Services
- GSM Mobile Services including 2G, 3G, 4G & Value added Services (VAS) • Internet and Broadband services including Fiber to the Home (FTTH)
- Wi-Fi services
- Data Center services Enterprise Data services such as Leased circuits, MPLS VPN etc
- National Long Distance services
- International Long Distance Services

BSNL telecom network therefore, is a part of modern global network, providing access to countries around the world for transporting information in the form of voice, data and video.

The company has a vast experience in planning, installation, network integration & maintenance of switches & transmission networks. BSNL has a world class ISO 9000 certified Telecom Training Institute.

The details of Services provided by BSNL as on 30.11.2021 are as under: BSNL has modernized its network by incorporating state of the art technology with 100%
digital new technology switching network. BSNL has a customer base of 1208.35 lakh subscribers.

Wire-line Services: The vast switching network for landlines comprises of 27,329 exchanges having a capacity of 228.62 lakh lines serving 76.48 lakh customers.

Wireless Services: BSNL has covered almost all the cities, and substantial length of National Highways, Rail Routes and State Highways. The Cellular services of BSNL are also providing incidental coverage to the rural areas falling en-route to

National and important State Highways. 1131.86 Lakh mobile connections are working against equipped capacity of 1142.50 lakhs. BSNL has 84,544 BTSs of 2G technologies, 62,408 Node-Bs of 3G technology and 8,637 e Node-Bs of 4G technologies. The 3G mobile facility has been rolled out in 6,272 cities / towns.

Broadband Services: BSNL had launched its Broadband services in January 2005 using ADSL2+ technology. As on 30.11.2021, 23.06 lakh connections are working with the installed capacity of 100.18 lakhs broadband ports. BSNL has covered 665 DHQ, 5,996 BHQ, 4,524 Cities and 1, 71,476 Villages with broadband services.

BSNL is providing Wi-Fi broadband connections. As on 30.11.2021, BSNL has 6.09 lakh Wi-Fi unique users.

In addition, BSNL is providing a host of Value Added Services (VAS) to its landline and mobile customers. VAS is normally a third party service and is provided on franchise model on revenue share basis.

BSNL VISION

- Be the leading telecom service provider in India.
- Be a customer-focused organization with excellence in customer care and marketing.
- Leverage technology to provide affordable and innovative telecom services / products across customer segments.

BSNL MISSION

- Becoming the most trusted preferred and admired telecom brand.
- Providing reliable telecom services that are value for money.
- Generating value for all stakeholders - employees, shareholders, vendors and business associates.
- Excellence in customer services-friendly, reliable, time bound, convenient and courteous service.
- Offering differentiated products / services tailored to different service segments.
- Developing a marketing culture that is responsive to customer needs.

- Maximizing return on existing assets with sustained focus on profitability.

BSNL OBJECTIVES

- Increase sales revenue with focus on subscriber retention & acquisition by way of strengthening marketing, quality of service and customer delivery.
- Accelerate the pace of expansion of mobile & data services with up gradation of technology.
- Adopt policies and processes to enable transparent, quick and efficient decision making.
- Developing marketing team with attitude towards customer care.
- Improve customer care by reducing fault rate, upgrading Customer service and convergent billing.
- Providing a conducive work environment with strong focus on performance to enhance customer delight towards BSNL services.
- Leverage data services to increase BSNL's customer's base & revenues by providing higher bandwidths capabilities for wire line and wireless broadband customers.
- Strengthen company's finances by gainful utilization of its assets through sharing / monetization of existing infrastructure like land, building and sharing of passive infrastructure like towers etc.
- Creating Wi-Fi Hot Spots and replacing Legacy wire line exchanges by Next Generation Network.
- Expanding the reach of fiber network near to the customer premises particularly in apartment complexes through FIT I-I in order to meet the bandwidth requirement for both data & video applications.
- Leverage the existing infrastructure of BSNL thereby contributing towards nation building by facilitating the execution of government programs and initiatives viz. National Optical Fiber Network (NOFN), Network for spectrum (NFS), and Smart City concept.
- Improve productivity by training and skill development and rationalization of manpower.

- Developing knowledge pool exposed to latest technological advancements.
- To explore opportunities in international telecom in developing markets.
- To become preferred service provider to the Government for reliable and secure service Network and to serve National security interests.

BROADBAND

Broadband Internet- broadband indicates a means of connectivity at a very high or 'broad' bandwidth. It is defined as an 'always on' data connection that is able to support interactive services including internet access and has the capability of the minimum download speed of 256 kbps to an individual subscriber from point of presence (POP) of the service provider. Broadband technology will allow for highspeed transmission of voice, video, and data over network like the internet.

- Broad band speed range-256kpbs-8000kpbs.

Services available through broadband:

- High speed internet Access- This will always provide internet access service with high speed ranging from 2mpbs to 8mpbs.
- Bandwidth on demand- This will facilitate customer to change bane as per his/her requirement. For example, a customer with 2mpbs can change to 4mpbs during the IP TV session.
- Multicasting- This is to provide video multicast services for application in distance education, telecom etc.
- Dial VPN service- This service allows remote users to access their private network securely over the NIB-II infrastructure.
- Content based services - Like video on demand, Interactive gaming live and time shifted TV.

Digital subscriber line technology- DSL is the next generation modem like technology that allows for the transmission of voice, video and data over existing copper telephone lines at incredible megabit speed.

The copper telephone lines are often referred to as the local loop or the last mile from the exchange to end user's home or business. DSL provides high speed internet access using regular telephone lines.

Some of its functional elements —

Broadband Remote Access Server (BRAS) - This is a device which acts as a gateway. It offers following services:

- Internet access
- Broadband acces
- VLAN and gigabit internet services
- VPN
- Multicasting
- Content delivery services

Aggregation switch (LAN switch)-the subscriber connection from DSLAMs are aggregated through this before passed onto core router. This again is implemented in two tiers, with a tier-I switch aggregating traffic from multiple tier-2 switches.

DSL Access Multiplexer (DSLAM) - The DSL Access multiplexer, popularly known as DSLAM, is employed by the telecom companies to code the subscriber line with the broadband data content. Once the ADSL copper lines reaches the customer, some customer premises equipment (CPE) has to be employed to separate the voice and data signals.

Broadband Applications

-Personal services

1. High speed internet access
2. Multimedia

-Government Public Services

1. E-governance
2. E-education

-Commercial Services I.E-commerce

1. E-commerce
2. Corporate internet

3. Video conferencing

-Video and entertainment services

1. Broadband TV.
2. Video on demand
3. Interactive Gaming
4. Music on demand
5. Online radio

TELEPHONE EXCHANGE

The main function of an exchange is to process call from a calling subscriber and make connection to the called subscriber. This connection can be direct or via another exchange. This requires all parts of the exchange to work as a unit to ensure this is properly handled.

BRIEF HISTORY:

The Center for Development of Telemetric (C-DOT) is the telecom technology development center of the government. It was established in August 1984 as an autonomous body. It was vested with full authority and total flexibility to develop state of-the-art telecommunication technology to meet the needs of the Indian telecommunication network. The key objective was to build a center for excellence in the area of telecom technology.

ORGANIZATION:

The management of C-DOT has a three-tier structure:

The governing Council: provides policy guidelines and approves the annual budget of the center.

The Steering Committee: has the role of reviewing and monitoring the performance of the center.

The Project Board: is responsible for the implementation of C-DOT's project and the day-to-day function of the center.

ARCHITECTURE

C-DOT switches have distributed architecture .a base module (BM) has 512 port . which can provide non-blocking connectivity and can accept concentrated

subscriber lines. The BM, capable of serving up to 2000 lines or 512 trunks, is used as the basic building block. Each BM is housed in a single Cabinet.

By interconnecting up to 35 such BMs through a central module (CM), the switch can support up to 40,000 lines. The system can support up to 4:1 concentration. Lines and trunks can be intermixed in the same BM. The front-end of the system consists of an input output processor (IOP) connected to the administration processor (IOP) connected to the administration processor (AP).

Each System consist of two IOPS working in duplex mode .10P, based on Motorola 68040 and supporting UNIX environment, provides interface for the main machine communication, system initialization support operator features etc.

OBJECTIVES:

- Work on telecom technology products and services.
- Provide solutions for current and future requirements of telecommunication and converged networks including those required for rural application.
- Provide market orientation to R & D activities and sustain C-DOT as center of excellence.
- Build partnerships and joint alliances with industry, solution provides, telcos and other development organizations to offer cost effective solution.
- Support telcos and service provides in the introduction of new technologies, features and services by optimal utilization of installed network.

ELECTRONIC EXCHANGE:

- All control functions by series of instructions are stored in memory.
- Memories are modifiable and control program can always be rewritten, For each call processing step decision is taken according to class of service.

CARRIER ROOM:

Leased line connectivity is provided in carrier room. This room has two parts:
1. Conventional leased line system

2. MLLN

CONVENTIONAL LEASED LINE SYSTEM:

- It consists of modems and routers that are provided by the company requesting for that network.
- Connectivity of different ATM, banks etc. is provided by BSNL here.
- For this, we have 4 modems (2 in Exchange, 1 at sender and 1 at receiver)
- Modems are used for short distances i.e. trans and receive part are received here and local lead connection is given to the subscriber.
- Local lead faults can be handled here but the trans and receive faults can be handled by the department meant for it.
- Accept 64Kbps or 2 Mbps.
- For long distance communication we have MUXS and data is sent through optical fibers. MUXs are present at both the ends.

MANAGED LEASED LINE NETWORK:

- No open wiring.
- Route can be changed by the computer software. • In Agra Gate Exchange, we have 3 VMUX of type II.

MDF (MAIN DISTRIBUTION FRAME):

M.D.F. is a media between switching network and subscriber's line. It is a termination point within the local telephone exchange where exchange equipment and terminations of local loops are connected by jumper wires.

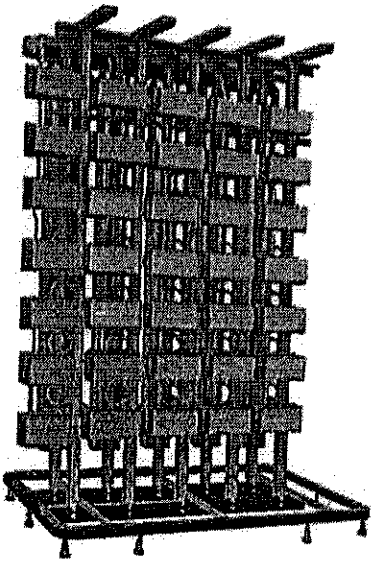


Fig. 1.1 MDF

FUNCTIONS OF MDF:

- All cable copper wires supplying services through user telephone lines are terminated and distributed through MDF.
- The most common kind of large MDF is a long steel rack accessible from both sides. Each jumper is a twisted wire.
- It consists of local connection and broadband connection frames for the main Exchange area.
- The MDF usually holds central office protective devices including heat coils and functions as a test point between a line and the office.
- It provides testing of calls.
- It checks whether fault is indoor or external.
- All lines terminate individually.

POWER PLANT:

- It provides -48V to the switch rooms and 48V to the connections.
- Batteries are artificially discharged once in a year for their maintenance.
- Cooling is provided through fans & AC.
- There is earth region too for protection,

OFC IN BSNL

OFC stands for optical fiber communication. The use and demand for optical fiber has grown tremendously and optical-fiber applications are numerous. Telecommunication applications are widespread, ranging from global networks to desktop computers. These involve the transmission of voice, data, or video over distances of less than a meter to hundreds of kilometers, using one of a few standard fiber designs in one of several cable designs.

Carriers use optical fiber to carry plain old telephone service (POTS) across their nationwide networks. Local exchange carriers (LECs) use fiber to carry this same service between central office switches at local levels, and sometimes as far as the neighborhood or individual home (fiber to the home [FTI-I]).

Optical fiber is used by BSNL for transmission purposes OFC being a safer and faster method of transmission provides efficiency in communication.

The high bandwidth provided by fiber makes it the perfect choice for transmitting broadband signals, such as high-definition television (HDTV) telecasts. Intelligent transportation systems, such as smart highways with intelligent traffic lights, automated tollbooths, and changeable message signs, also use fiber-optic-based telemetry systems.

FIBER OPTIC SYSTEM

Optical Fiber is new medium, in which information (voice, Data or Video) is transmitted through a glass or plastic fiber, in the form of light, following the transmission sequence give below •

(1) Information is encoded into Electrical Signals.

(2) Electrical Signals are converted into light Signals.

(3) Light Travels down the Fiber.

(4) A Detector Changes the Light Signals into Electrical Signals.

(5) Electrical Signals are decoded into Information.

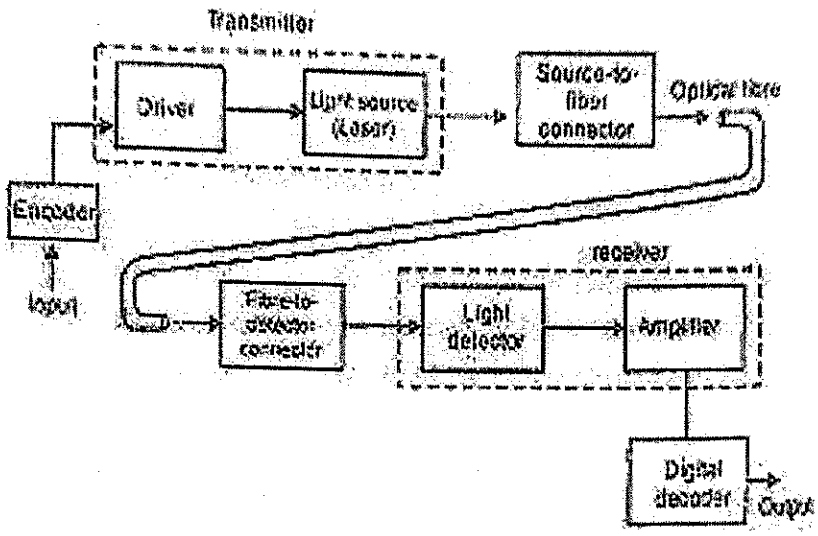


Fig. 1.2

- Inexpensive light sources available.

- Repeater spacing increases along with operating speeds because low loss fibers are used at high data rates.

A Detector Changes the Light Signals into Electrical Signals.

PRINCIPLE OF OPERATION: THEORY

- Total Internal Reflection The Reflection that Occurs when a Light Ray Travelling in One Material Hits a Different Material and Reflects Back into the Original Material without any Loss of Light.

second material. The angle of incidence and reflection are equal and it is called Total Internal Reflection.

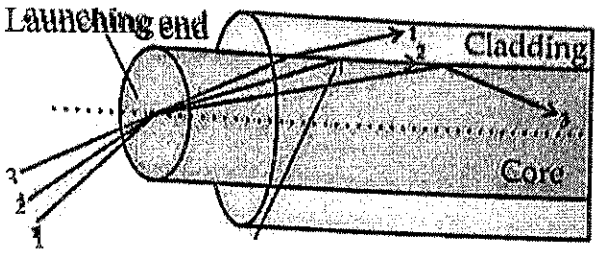
PROPOGATION OF LIGHT THROUGH FIBER

The optical fiber has two concentric layers called the core and the cladding. The inner core is the light carrying part. The surrounding cladding provides the difference refractive index that allows total internal reflection of light through the core. The index of the cladding is less than 1%, lower than that of the core. Typical values for example are a core refractive index of 1.47 and a cladding index of 1.46. Fiber manufacturers control this difference to obtain desired optical fiber characteristics. Most fibers have an additional coating around the cladding. This buffer coating is a shock absorber and has no optical properties affecting the propagation of light within the fiber. Figure shows the idea of light travelling through a fiber. Light injected into the fiber and striking core to cladding interface at greater than the critical angle, reflects back into core, since the angle of incidence and reflection are equal, the reflected light will again be reflected. The light will continue zigzagging down the length of the fiber. Light striking the interface at less than the critical angle passes into the cladding, where here it is lost over distance. The cladding is usually inefficient as a light carrier, and light in the cladding becomes attenuated fairly. Propagation of light through fiber is governed by the indices of the core and cladding by Snell's law.

Such total internal reflection forms the basis of light propagation through a optical fiber. This analysis consider only meridional rays- those that pass through the fiber axis each time, they are reflected. Other rays called Skew rays travel down the fiber without passing through the axis. The path of a skew ray is typically helical wrapping around and around the central axis. Fortunately skew rays are ignored in most fiber optics analysis.

The specific characteristics of light propagation through a fiber depends on many factors, including

- The size of the fiber.
- The composition of the fiber.
- The light injected into the fiber,



Critical angle

Propagation of light through the optical fibre

Fig. 3.1

LOSSES IN OPTICAL FIBER

- There are several points in an optic system where losses occur.
- These are: Coupler, Splices connectors and the fiber itself.
- Losses associated within the fiber classified as under:
- Losses due to absorption: Even the purest glass will absorb heavily within specific wavelength regions. Other major source of loss is impurities like, metal ions and OH ions.
- Losses due to scattering: caused due to localized variations in density called Rayleigh scattering and the loss are: L = 1.7(0.85/•) ^{4} db/km.
- Losses due to geometric effect:
- Micro-bending
- Macro-bending

OPTICAL SOURCES

- The device that actually converts electrical signals to its optical equipment,
- Most common light sources:
- Light emitting diodes (LEDis).

- Light amplification by Stimulated Emission of Radiation (LASER) diodes. It is particularly required in lasers to maintain stable output power by way of feedback mechanism.

- Laser is very sensitive to temperature. Operating characteristics of a semiconductor laser- notably threshold, current, output power, and wavelength change with temperature sensing and control is required to maintain stable temperature.

TRANSMISSION SYSTEM

In telecommunications a transmission system is a system that transmits a signal from one place to another. The signal can be an electrical, optical or radio signal. Some transmission systems contain multipliers, which amplify a signal prior to retransmission, or regenerators, which attempt to reconstruct and re-shape the coded message before re-transmission.

Pulse-code modulation

Pulse-code modulation (PCM) is a digital representation of an analog signal where the magnitude of the signal is sampled regularly at uniform intervals, then Wantized to a series of symbols in a digital (usually binary) code. PCM has been used in digital telephone systems and is also the standard form for digital audio in computers and the compact disc red book format. It is also standard in digital video, for example, using ITU-R BT.601. However, straight PCM is not typically used for video in consumer applications such as DVD or DVR because it requires too high a bit rate (PCM audio is supported by the DVD standard but rarely used). Instead, compressed variants of PCM are normally employed. However, many Bluray Disc movies use uncompressed PCM for audio. Very frequently, PCM encoding facilitates digital transmission from one point to another (within a given system, Or geographically) in serial form.

History of PCM

In retrospect, PCM, like many other great inventions, appears to be simple and obvious. In the history of electrical communications, the earliest reason for

sampling a signal was to interlace samples from different telegraphy sources, and convey them over a single telegraph cable. Telegraph time-division multiplexing (TDM) was conveyed as early as 1853, by the American inventor M.B. Farmer. The electrical engineer W.M. Miner, in 1903, used an electro mechanical commutator for time-division multiplex of multiple telegraph signals, and also applied this technology to telephony. He obtained intelligible speech from channels sampled at a rate above 3500-4300 Hz: below this was unsatisfactory. This was TDM, but pulse-amplitude modulation (PAM) rather than PCM. Paul M. Rainey of Western Electric in 1926 patented a facsimile machine using an optical mechanical analog to digital converter. The machine did not go into production. British engineer Alec Reeves, unaware of previous work, conceived the use of PCM for voice communication in 1937 while working for International Telephone and Telegraph in France. He described the theory and advantages, but no practical use resulted. Reeves filed for a French patent in 1938, and his U.S. patent was granted in 1943.

The first transmission of speech by digital techniques was the SIGSALY vocoder encryption equipment used for high-level Allied communications during World War II from 1943.1t was not until about the middle of 1943 that the,Bell Labs people who designed the SIGSALY system, became aware of the use of PCM binary coding as already proposed by Alec Reeves. PCM in the 1950s used a cathode-ray coding tube with a grid having encoding perforations. As in an Oscilloscope, the beam was swept horizontally at the sample rate while the Vertical deflection was controlled by the input analog signal, causing the beam to pass through higher or lower portions of the perforated grid. The grid interrupted •the beam, producing current variations in binary code. Rather than natural binary, the grid was perforated to produce Gray code lest a sweep along a transition zone produce glitches

Managed Leased Line Network (MLLN)

Leased Line services with flexible access bandwidth. The MLLN is a Managed Leased Line Network system which is proposed to provide Leased line COnnectivity. With the State-of-the-art technology equipment, MLLN is designed mainly for having effective control and monitor on the leased line so that the down time is very much minimized and the circuit efficiency is increased thus achieving

more customer satisfaction. This mainly deals with data circuits ranging from 64 KBPS to 2048 KBPS.

In MLLN network conventional PCM primary MUX and subscriber Modems are replaced by versatile MUX and network terminating units respectively. MLLN mainly consists of Digital Cross Connect (DXC), versatile MUX (V MUX), Network Terminating Units (NTU) and Network Management System (NMS). DXC's and vMUXIs are inter connected via optical fiber links with alternate routing facility in case of any route failure. VMUX is in turn connected to NTU ' s placed at customer premises through 2 wire copper pair. At the top of it NMS is suitably placed at the central location for effective control & monitor.

NTUS are fully managed from NMS. They are programmable for different data speeds ranging from 64 KBPS (n x64 KBPS: nxl to 32) depending upon the customer demand, thus having bandwidth control without changing Modem at his premises. NTUS operate on 230 V AC.

Features of MLLN

- Control, Manage the leased line network.
- Bandwidth management as per the customer demand.
- Pro-active maintenance, without waiting for customer to book a complaint.
- Self Diagnostic/software loops to check El connectivity to DXC, VMUX/software loops for checking copper pair at NTU point for immediately identifying the faulty section for trouble shooting.
- Alternate routing in case of any route failure. Generation of the periodic Performance reports for self-analysis.

Synchronous Digital Hierarchy

SDH (Synchronous Digital Hierarchy) is a standard technology for synchronous data transmission on optical media. It is the international equivalent of synchronOUS Optical Network. Both technologies provide faster and less expensive network interconnection than traditional PDH (Plesiochronous Digital Hierarchy)

equipment. In digital telephone transmission, "synchronous" means the bits from one call are carried within one transmission frame. "Plesiochronous" means "almost (but not) synchronous," or a call that must be extracted from more than one transmission frame.SDH uses the following Synchronous Transport Modules (STM) and rates: STM-l (155 megabits per second), STM-4 (622 Mbps), STM-16 (2.5 gigabits per second), and STM-64 (10 Gbps).

GSM (Global System for Mobile Communications)

GSM (Global System for Mobile Communications, originally Grouped Special Mobile), is a standard set developed by the European Telecommunications standards Institute (ETSI) to describe technologies for second generation (or "26") digital cellular networks. Developed as a replacement for first generation analogue cellular networks, the GSM standard originally described a digital, circuit switched network optimized for full duplex voice telephony. The standard was expanded over time to include first circuit switched data transport, then packet data transport via GPRS. Packet data transmission speeds were later increased via EDGE. The GSM standard is succeeded by the third generation (or "36") UMTS standard developed by the 3GPP. GSM networks will evolve further as they begin to incorporate fourth generation (or "4G") LTE Advanced standards. "GSM" is a trademark owned by the GSM Association. The GSM Association estimates that technologies defined in the GSM standard serve 80% of the global mobile market, encompassing more than 1.5 billion people across more than 212 countries and territories, making GSM the most ubiquitous of the many standards for cellular networks.

Introduction to GSM Networks

The various interface labels are the formal names given to these interfaces. More details about these interfaces are found in GSM TS 03.02 [261.

The GSM network consists mainly of the following functional parts:

i. MSC - the mobile service switching centre (MSC) is the core switching entity in the network. The MSC is connected to the radio access network (RAN); the RAN is formed by the BSCS and BTSS within the Public Land Mobile

Network (PLMN). Users of the GSM network are registered with an MSC; all calls to and from the user are controlled by the MSC. A GSM network has one or more MSCs, geographically distributed.

ii. VLR- the visitor location register (VLR) contains subscriber data for subscribers registered in an MSC. Every MSC contains a VLR. Although MSC and VLR are individually addressable, they are always contained in one integrated node.

iii. GMSC-the gateway MSC (GMSC) is the switching entity that controls mobile terminating calls. When a call is established towards a GSM subscriber, a GMSC contacts the HLR of that subscriber, to obtain the address of the MSC where that subscriber is currently registered. That MSC address is used to route the call to that subscriber.

iv. HLR- the home location register (HLR) is the database that contains a subscription record for each subscriber of the network. A GSM subscriber is normally associated with one particular HLR. The HLR is responsible for the sending of subscription data to the VLR (during registration) or GMSC (during mobile terminating call handling).

v. CN- the core network (CN) consists of, amongst other things, MSC(s), GMSC(s) and HLR(s). These entities are the main components for call handling and subscriber management. Other main entities in the CN are the equipment identification register (EIR) and authentication centre(AUC). CAMEL has no interaction with the EIR and AUC; hence EIR and AUC are not further discussed.

vi. BSS - the base station system (BSS) is composed of one or more base station controllers (BSC) and one or more base transceiver stations (BTS). The BTS contains one or more transceivers.

vii. (TRX)- The TRX is responsible for radio signal transmission and reception. BTS and BSC are connected through the Abis interface. The BSS is connected to the MSC through the A interface.

viii. MS-the mobile station (MS) is the GSM handset. The structure of the MS will be described in more detail in a next section. A GSM network is a public land mobile network (PLMN). Other types of PLMN are the time division multiple access (TDMA) network or code division multiple access (CDMA) network. GSM uses the following sub-division of the PLMN:

ix. Home PLMN (HPLMN) - the HPLMN is the GSM network that a GSM user is a subscriber of. That implies that GSM user's subscription data resides in the

x.

HLR in that PLMN. The HLR may transfer the subscription data to a VLR (during registration in a PLMN) or a GMSC (during mobile terminating call handling). The HPLMN may also contain various service nodes, such as a short message service centre (SMSC), service control point (SCP), etc. Visited PLMN (VPLMN) - the VPLMN is the GSM network where a subscriber is currently registered. The subscriber may be registered in her HPLMN or in another PLMN. In the latter case, the subscriber is outbound roaming (from HPLMN's perspective) and inbound roaming (from VPLMN's perspective). When the subscriber is currently registered in her HPLMN, then the HPLMN is at the same time VPLMN.

xi. Interrogating PLMN (IPLMN) - the IPLMN is the PLMN containing the GMSC that handles mobile terminating (MT) calls. MT calls are always handled by a GMSC in the PLMN, regardless of the origin of the call. For most operators, MT call handling is done by a GMSC in the HPLMN; in that case, the HPLMN is at the same time IPLMN. This implies that calls destined for a GSM subscriber are always routed to the HPLMN of that GSM subscriber. Once the call has arrived in the HPLMN, the HPLMN acts as IPLMN. MT call handling will be described in more detail in subsequent sections. When basic optimal routing (BOR) is applied, the IPLMN is not the same PLMN as the HPLMN. The user of a GSM network is referred to as the served subscriber; the MSC that is serving that subscriber is known as the serving MSC.

xii. Mobile originated call - the MSC that is handling the call is the serving MSC for this call; the calling subscriber is the served subscriber.

xiii. Mobile terminated call - the GMSC that is handling the call is the serving GMSC for this call; the called subscriber is the served subscriber.

Signaling in GSM

The various entities in the GSM network are connected to one another through signaling networks. Signaling is used for example, for subscriber mobility, subscriber registration, call establishment, etc. The connections to the various entities are known as 'reference points', Examples include:

a. An interface-the connection between MSC and BSC;
b. Abis interface - the connection between BSC and BTS;
c. D interface- the connection between MSC and HLR;
d. Um interface - the radio connection between MS and BTS.

various signaling protocols are used over the reference points, Some of these protocols for GSM are the following:

a. mobile application part (MAP) - MAP is used for call control, subscriber registration, short message service, etc.; MAP is used over many of the GSM network interfaces;

b. base station system application part (BSSAP)- BSSAP is used over the A;

c. direct transfer application part (DTAP)-DTAP is used between MS, MSC;

d. ISDN user part (ISUP)-ISUP is the protocol for establishing and releasing

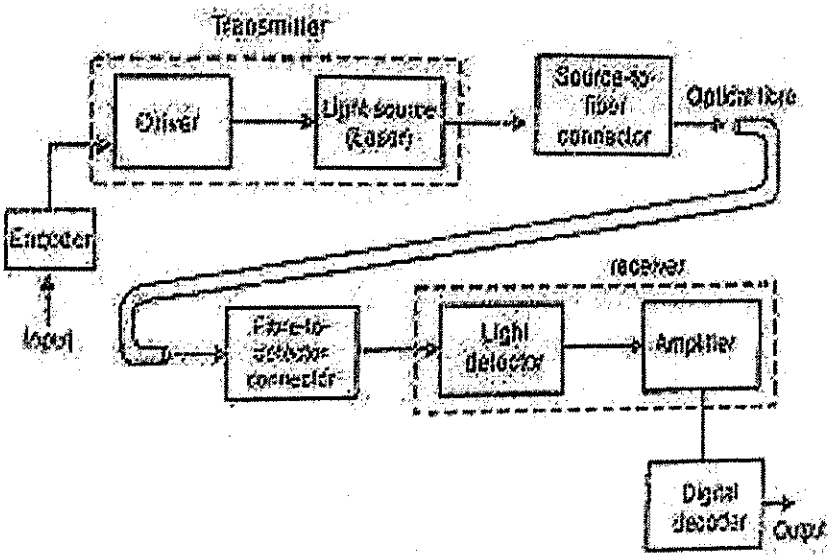


Fig. 3.2

e. When it comes to call establishment, GSM makes a distinction between Signaling and payload. Signaling refers to the exchange of information for call set up; payload refers to the data that is transferred within a call, i.e. voice, video, fax etc. For a mobile terminated GSM call, the signaling consists of exchange of MAP messages between GMSC, HLR and visited MSC (VMSC). The payload is transferred by the ISUP connection between GMSC and VMSC. It is a continual aim to optimize the payload transfer through the network, as payload transfer has a direct cost aspect Associated with it. Some network services are designed to optimize the payload transfer. One Example is optimal routing.

The network is structured into a number of discrete sections:

- The Base Station Subsystem (the base stations and their controllers).
- The Network and Switching Subsystem (the part of the network most similar to a fixed network). This is sometimes also just called the core network.

Network Switching Subsystem

Network switching subsystem (NSS) (or GSM core network) is the component of a GSM system that carries out call switching and mobility management functions for mobile phones roaming on the network of base stations. It is owned and deployed by mobile phone operators and allows mobile devices to communicate with each other and telephones in the wider Public Switched Telephone Network or (PSTN). The architecture contains specific features and functions which are needed because the phones are not fixed in one location.

The NSS originally consisted of the circuit-switched core network, used for traditional GSM services such as voice calls, SMS, and circuit switched data calls. It was extended with an overlay architecture to provide packet-switched data services known as the GPRS core network. This allows mobile phones to have access to services such as WAP, MMS, and the Internet.

All mobile phones manufactured today have both circuit and packet based services, so most operators have a GPRS network in addition to the standard GSM core network.

Mobile switching center (MSC)

The mobile switching center (MSC) is the primary service delivery node for GSM/CDMA, responsible for routing voice calls and SMS as well as other services (such as conference calls, FAX and circuit switched data). The MSC sets up and releases the end-to-end connection, handles mobility and hand-over requirements during the call and takes care of charging and real time pre-paid account monitoring. In the GSM mobile phone system, in contrast with earlier analogUe services, fax and data information is sent directly digitally encoded to the MSC. Only at the MSC is this recoded into an "analogue" signal (although

actually this will almost certainly mean sound encoded digitally as PCM signal in a 64-kbit/s timeslot, known as a DSO in America). There are various different names for MSCs in different contexts which reflects their complex role in the network, all of these terms though could refer to the same MSC, but doing different things at different times.

The Gateway MSC (G-MSC) is the MSC that determines which visited MSC the subscriber who is being called is currently located at. It also interfaces with the PSTN. All mobile to mobile calls and PSTN to mobile calls are routed through a GMSC. The term is only valid in the context of one call since any MSC may provide both the gateway function and the Visited MSC function, however, some manufacturers design dedicated high capacity MSCs which do not have any BSSS connected to them. These MSCs will then be the Gateway MSC for many of the calls they handle.
The visited MSC (V-MSC) is the MSC where a customer is currently located. The VLR associated with this MSC will have the subscriber's data in it. The anchor MSC is the MSC from which a handover has been initiated.

Mobile switching centre server (MSCS)

The mobile switching centre server is a soft-switch variant of the mobile switching centre, which provides circuit-switched calling, mobility management, and GSM services to the mobile phones roaming within the area that it serves. MSS fUnctionality enables split between control (signalling) and user plane (bearer in network element called as media gateway/MG), which guarantees better Placement of network elements within the network. MSS and MGW media gateway makes it possible to cross-connect circuit switched calls switched by using IP, ATM AAL2 as well as TDM. More information is available in 3GPP TS 2205.

The MSC connects to the following elements:

- The home location register (HLR) for obtaining data about the SIM and mobile services ISDN number (MSISDN; i.e., the telephone number).
- The base station subsystem which handles the radio communication with 2G and 2.5G mobile phones.

- The UM TS terrestrial radio access network (UTRAN) which handles the radio communication with 3G mobile phones.
- The visitor location register (VLR) for determining where other mobile subscribers are located.
- Other MSCs for procedures such as handover.

CONCLUSION

The final conclusion is the study of basics of TELECOMMUNICATION system installed and operated in BSNL will help us understand the practical variations in the theoretical concepts studied in our course plan. The study also helps us to have a better understanding of the current technology being implemented and also the scope of further research in the field of communication. As we are moving more towards the software and IT sector, even communication sector is evolving itself into a software based hardware sector. So the further scope for research and development is vast.

INTERNSHIP REPORT

A report submitted in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNITATION ENGINEERING

by
SAUMYA KAUSHIK
Regd. No.: 20104103001
Roll No.: 202002

Under Supervision of

Whole team

OF

E & ICT ACADEMY, IIT KANPUR
(Duration: 6th MAY, 2022 to 9th JUNE, 2022)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
Bihta, Patna, Bihar

BATCH
2020 – 2024

Scanned with ACE Scanner

DEPARTMENT OF ELECTRONICS AND COMMUNITATION ENGINEERING
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

BIHTA, PATNA, BIHAR

CERTIFICATE

This is to certify that the **—Internship report"** submitted by **SAUMYA KAUSHIK (Regd. No.: 20104103001)** is work done by him and submitted during 2021 – 2022 academic year, in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in** ELECTRONICS AND COMMUNITATION **ENGINEERING,** at **E & ICT ACADEMY, IIT KANPUR.**

Head of the Department
ELECTRONICS AND COMMUNITATION Engineering

External Examiner

Date:

Date:

Scanned with ACE Scanner

In association with

TEQIP-3

Technical Education Quality Improvement Program

E & ICT Academy, IIT Kanpur

(A Joint Initiative of MeitY & IIT Kanpur)

This is to certify that

Dr. / Mr. / Ms. **SAUMYA KAUSHIK**

of

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA

has successfully completed 4 weeks Summer Training on

EMBEDDED SYSTEM

from **6 May 2022 to 9 June 2022**

Performance Level: **Excellent**

Prof. B. V. Phani

Prof. Amey Karkare

Chief Investigators, E & ICT Academy, IIT Kanpur

Scanned with ACE Scanner

INTERNSHIP REPORT

A report submitted in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNITATION ENGINEERING

by

SHASHIKANT KUMAR
Regd. No.: 20104103003
Roll No.: 202004

Under Supervision of

Whole team

OF

E & ICT ACADEMY, IIT KANPUR
(Duration: 6th MAY, 2022 to 9th JUNE, 2022)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY
Bihta, Patna, Bihar

BATCH
2020 – 2024

DEPARTMENT OF ELECTRONICS AND COMMUNITATION ENGINEERING
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

BIHTA, PATNA, BIHAR

CERTIFICATE

This is to certify that the —**Internship report"** submitted by **SHASHIKANT KUMAR (Regd. No.: 20104103003)** is work done by him and submitted during 2021 – 2022 academic year, in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in** ELECTRONICS AND COMMUNITATION **ENGINEERING,** at **E & ICT ACADEMY, IIT KANPUR.**

Head of the Department
ELECTRONICS AND COMMUNITATION Engineering

External Examiner

Date:

Date:

In association with

TEQIP-3

Technical Education Quality Improvement Program

E & ICT Academy, IIT Kanpur

(A Joint Initiative of MeitY & IIT Kanpur)

This is to certify that

Dr. / Mr. / Ms. **SHASHIKANT KUMAR**

of

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA

has successfully completed 4 weeks Summer Training on

EMBEDDED SYSTEM

from **6 May 2022 to 9 June 2022**

Performance Level: **Excellent**

Prof. B. V. Phani

Prof. Amey Karkare

Chief Investigators, E & ICT Academy, IIT Kanpur

CHAPTER-1

INTRODUCTION

ThemostbasicdefinitionforEmbeddedSystemscanbewrittenasahardwaresystemalong withasoftwarepartembeddedtoittoperformvarioustasks.AnEmbeddedSystemscanbe definedasanindependententityorasystemthatisapartofamuchlargersystem.Itcanbe consideredasamicrocontrolleroramicroprocessordesignedtoperformvariousapplication basedtasks.

Embedded Systems comprise of three main components -

- Hardwarecomponents
- Softwarecomponents
- RTOStosupervisethefunctions.

mechanicalorelectricalsystem,frequentlywithreal-timecomputingconstraints..mostfrequently afteraddinghardwareandsoftwareitisamedicinecompletedevice.Dailyweareusingembedded systemtocontrolmanydevices.Anembeddedsystemisatypeofcomputersystemwithinscribed functionwithinahugemechanicalorelectricalsystem,frequentlywithreal-timecomputing constraints.

Embedded systems are generally classified into two types. They are

- Generalpurposeembeddedsystems
- Realtimeembeddedsystems

Examples for general purpose embedded systems:

- Laptops
- Mobilephones

Examples for real time embedded systems:

• AirconditionerAnembeddedsystemisacomputersystemwithadedicatedfunction within alarger

• Traffic lightcontrollers

The major difference between these two embedded systems is that the general-purpose system uses microprocessor while the other one uses

MicrocontrollerNowletushavealookonthedifferencesbetweenamicroprocessoranda microcontroller.

MicroprocessorisanICwhichhasonlytheCPUinsidethemi.e.onlytheprocessingpowers suchasIntel'sPentium1,2,3,4,core2duo,i3,i5etc.Thesemicroprocessorsdon'thaveRAM, ROM,andotherperipheralonthechip.Asystemdesignerhastoaddthemexternallytomake themfunctional.ApplicationofmicroprocessorincludesDesktopPC's,Laptops,notepadsetc. ButthisisnotthecasewithMicrocontrollers.MicrocontrollerhasaCPU,inadditionwitha fixedamountofRAM,ROMandotherperipheralsallembeddedonasinglechip.Attimesitis alsotermedasaminicomputeroracomputeronasinglechip.Todaydifferentmanufacturers producemicrocontrollerswithawiderangeoffeaturesavailableindifferentversions.Some manufacturersareATMEL,Microchip,TI,Freescale,Philips,Motorolaetc.

Microcontrollersaredesignedtoperformspecifictasks.Specificmeansapplicationswherethe relationshipofinputandoutputiselucidated.Dependingontheinput,someprocessingneedstobe doneandoutputisdelivered.Forexample,keyboards,mouse,washingmachine,digicam,pendrive, remote,microwave,cars,bikes,telephone,mobiles,watches,etc.Sincetheapplicationsarevery specific,theyneedsmallresourceslikeRAM,ROM,I/Oportsetc.

Microprocessorsusedtofindanapplicationswhereassignmentsareunspecificlikedeveloping software,games,websites,photoediting,creatingdocumentsetc.Insuchsituationstherelationship betweeninputandoutputisnotdescribed.Theyneedmoreamountofresourceslike RAM,ROM,I/O portsetc.

TheclockspeedoftheMicroprocessorisquitehighascomparedtothemicrocontroller. Whereas the microcontrollers operate from a few MHz to 30 to 50 MHz, today's microprocessoroperateabove1GHzastheyperformcomplextasks.

- Examplesofmicroprocessor:8085,8086,Intelcore7.

- Examplesofmicrocontroller:8051,PIC,Arduino.

CHAPTER-2

DESCRIPTION OF THE ORGANIZATION

IndiainfotechanIndianbasedITSoftwaredevelopmentCompanythatspotlightson worthwhile,timelydeliveredandextremelyqualitativeITenabledservices.Withaprosperous andspeckledfamiliarityinsoftwaredevelopmentelucidationthatgivesyourbusinessanew dimensionoveryourcontestants.Ouruniquesoftwaremodelwillbringutmosttargetedresult toyourbusiness.ThisSoftwareDevelopmentCompanywithitsHeadOfficeinChennaihave ateamofprofessionalshavingin-depthexpertiseinlatesttechnologiesprovidesexpert softwaresolutionsforbusinessofallsizeslike-E-commerce,finance,travel&tourism, youthfulentrepreneurs,corporate,ITconsultancyorganizationsetc....Wedeliverthetop levelmobileapplicationsforAndroid,BlackBerry,Windows,Symbian,SmartPhones,iPhone, andiPadwiththetouchofiOS.Getwebsiteapplicationdoneonvariedplatformslike Java, PHP,.NETthatisproductivelydeliveredtovariouspartnersandcustomersglobally.At present,withourknowledgeandexpertise,wehavecreatedaworldwidetrailofclientsand partners.OurultimatecustomizedprofessionalSEOservicesofferedtolotsofentrepreneurs facilitatesthemtoincreaseutmostROIfortheirbusinessandincreasetraffic.

BENIFITS

- AssuredReturnonInvestment
- Faster developmentcycle
- Increased developmentproductivity
- Low developmentcosts
- Reducedtimetomarket
- Quality softwareproducts

DEVELOPMENTPROCESS

- ProjectKick-off
- SoftwareRequirementsAnalysisandManagement
- Software ProjectPlanning
- SoftwareProjectTrackingandmanagement
- Software QualityAssurance

ENGAGEMENTMODELS

- Fixed-Price
- Time andMaterial
- Dedicated DevelopmentTeam
- Offshore DevelopmentCentre
- On siteConsulting

DEVELOPMENT

India Info tech is a software development company which offer Custom Software Developmentservices,includingproductideaanalysis,offshoresoftwaredevelopmentto outsourcingsupport,andenhancement.IndiaInfotechemploysalargepoolofsoftware engineerscomingfromdifferentbackgrounds.WithIndiaInfotechsoftwaredevelopment services,youreduceoutsourcingsoftwaredevelopmentcostsandenhancerevenue growthduetoour:

- Highresponsiveness
- Skilledandorganizedpersonel
- Strongexperienceinprojectsofdifferentcomplexity

- Timetestedsoftwaredevelopmentprocess
- On-timeandwithinbudgetdelivery

DESKTOPAPPLICATION

DesktopApplicationsarerobust,rich-clientapplicationsthatarecompiledandrunusingthe processingpowerofanindividualuser'sdesktopcomputer.IndiaInfotechhasexpertiseina widevarietyprogramminglanguages,tools,andplatformsforclientserverapplication.Our structuredmethodologiesandprovenITprocesseshelpstoreducecomplexity,risksandcosts associatedwithclient/serverdevelopment.Wewillhelpyoudevelop

customized software applications and assist you thorough out the entire software developmentLifeCycle,includingsystemsanalysisanddesign,softwaredevelopment,project management,implementationandtraining.

WEB APPLICATIONDEVELOPMENT

Applicationdevelopmenthasundergoneatransitionoveryears,withthemigrationof applicationsfromtraditionalclient/serverdesignstotheInternet.Weutilizecutting-edge technologiessuchasMicrosoftASP.NET,Java/J2EE,ROR,AjaxandXMLtomeetourcustomers variedrequirements.

SMART CLIENTAPPLICATION

Uniq tech has developed many Smart Client applications which enable Single Click Deployment,EnterpriseSecurity,ExtensibilityandOnlineandOfflineapplicationaccessetc.

CHAPTER 3

DEPARTMENT OF STUDENT INTERNED

ABOUT EMBEDDEDSYSTEM

Thecombinationofhardwareandsoftwareeitherfixedincapabilityorprogrammableisknow Embeddedsystem,thatisdesignedforaspecificfunction.Embeddedsystemsisdeviceused tocontrol,monitortheoperationofequipment's,machineryorplant."Embedded"refersto the internalpart.

These are classified into different steps :

Based on the Performance and Functional Requirement:

1. **REALTIMEEMBEDDEDSYSTEMS:**typeofsystemsaredefinedasthesesystemsin which thetruthfulnessofthesystemdependsnotonlyonthelogicalresultcomputation,but alsotheresultsareproducedontime.
   - Hardreal-timesystems(e.g.,Avioniccontrol).
   - Firmreal-timesystems(e.g.,Banking).
   - Softreal-timesystems(e.g.,Videoondemand).

2. **DETACHED EMBEDDEDSYSTEMS:**
   - Thissystemsdonotrequireahostlikeacomputer,dependsonitself.
   - Ittakestheinputsfromtheeitheranalogordigitalandprocesses,calculatesandmodifiedthe dataandgivestheenddatathroughthelinkdevicewhicheithercontrols,drivesanddisplays the linkeddevices.
   - examples:mp3musicplayers,digitalcameras,videogame,microwaveovensand temperature checkupsystems.

3. **NETWORKED EMBEDDED SYSTEMS:**
   - Thesesystemsarebranchtoanetworktoaccesstheresources.
   - connectednetworkcanbeLAN,WAN,ortheinternet.Theconnectionmaybewirelessor wired.Thissystemishighestgrowingareainembeddedsystemapplications.

4. **MOBILE EMBEDDED SYSTEMS:**
   - Thebasiclimitationsofthesedevicesistheotherresourcesandlimitations.
   - Mobileembeddedsystemsareusedinportableembeddeddeviceslikecellphone,mobiles, digitalcameras,mp3playersandpersonaldigitalassistants,etc

Based on the Performance of the Microcontroller:

1. **SMALL SCALE EMBEDDED SYSTEMS:**

- Thesetypesofsystemsaredesignedwitha8or16-bitsinglemicrocontroller,thatmaybe generatedbyabattery.
- Forthedevelopingembeddedsoftwareforthisembeddedsystems,theimportant programmingtoolsareeditor,assembler,crossassemblerandIDE.

2. **MEDIUM SCALE EMBEDDEDSYSTEMS:**

- Mediumscaleembeddedsystemsaredesignedwitha16or32-bitMC,RISCsorDSPs.
- Bothhardwareandsoftwarecomplexitiesareavailable.
- Forthedevelopingembeddedsoftwareforthissystem,themaintoolsareC,C++,JAVA, Visual,RTOS,debugger,sourcecodeengineeringtool,simulatorandIDE.

3. **WORDLY EMBEDDEDSYSTEMS:**

- Thesesystemsaredesignedwith32-bit.
- typeofembeddedsystemshavelargehardwareandsoftwarecomplexities.
- Whichmayrequiredscalableorconfigurableprocessorandprogramminglogicalarrays.

CHAPTER - 4

GENERAL ROLES AND RESPONSIBILITIES

IwasinternedattheEmbeddedDepartmentwhereiwasprovidedexposuretovarious microcontrollerboardsthatwereavailablefordevelopmentofvariousIOTapplicationareas likeconsumer,homeautomation,security,surveillance,healthcare,etc.Themainagendaof thedepartmenttobondthegapbetweenthecompanyandthestudentsbyprovidingthem withpracticalexperiencebyconsideringthevariousconstraintsthatcomesintoeffectduring thephysicalimplementationofaproject.

Thedepartmentwhereiwasinternedhelpedmebridgethegapbetweentheindustry and academia by providing the complete experience to on board using the various microcontrollerboardslikePICmicrocontroller,Arduinonanomicrocontrolleretc.

ThedepartmenthasanIndustryExcellenceExperienceCentertolearn,innovate,and prototype embedded designs on various Industry standard hardware and microcontroller platforms.AtthesametimebuildaRewardingCareertostudentsinembeddedEngineering Domain.TheLabsetupandtraininghelpedmetobecomeacompetentandproductiveAnalog andDigitaldesignEngineers.Thetrainingenabledmetoacquireknowledge,skillsand practicalexperienceacrosstheentirefrontendandbackendFullCustomFlow(Circuittotape-out).Thetrainingcoverskeyfundamentalconceptsofphysicalbesignmethodologywhichwill enhancetheemployabilityofthestudents.TheSessions,LabexercisesandIndustryStandard Projectsenabledmetogetthroughinstillsconfidenceandtheanalyticalabilitiesrequiredto workoncomplexindustry'schallengesinvariousDeepSub-MicronTechnologyProcess. ExposuretotheuseofPhysicalDesigntoolsfamiliaritywithtimingclosureandrelatedtopics arecovered.

CHAPTER 5

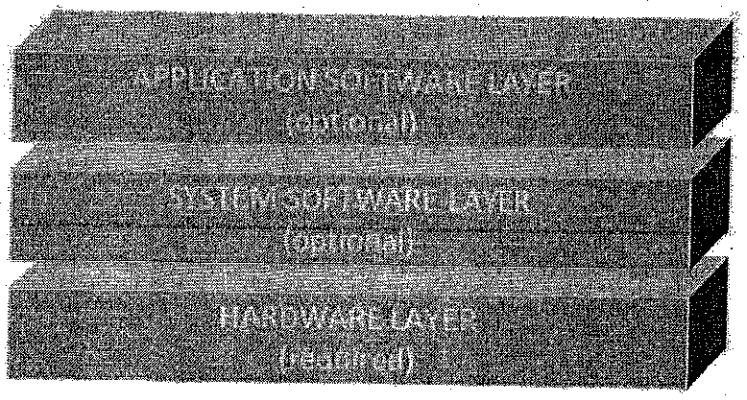
PROFESSIONAL AND TECHNICAL TAKE-AWAY

Embedded SystemArchitecture

Embeddedsystemarchitectureisangeneralizedformofthesystem.Butthisarchitecture doesnotshowadetailedimplementationinformationlikesoftwaresourcecodeorhardware circuitdesign:Asapartofcompositionofinteractingelementsthehardwareandsoftware elementsarepresentinembeddedsystem.hereelementsareusedtorepresentthehardware andsoftware,leavingonlybehavioralandinterrelationshipinformation.theonepossible representationofthearchitectureisastructure.Astructureislikesnapshotofthesystem's hardwareandsoftwareduringdesigntimeoratruntimehavingaparticularenvironmentand asetofelements.Earlierembeddedsystemarchitectureisusedtoresolvechallengesina project.Architectureisthefirsttoolusedtoanalysethesystemwithoutdefiningorknowing theinternallevelofimplementation.architecturecanalsobeusedasahighlevelblueprintin definingtheinfrastructureofadesign.

Designing an EmbeddedSystem

The four cornerstones of embedded system modelling are big-bang, code-and-fix, waterfall



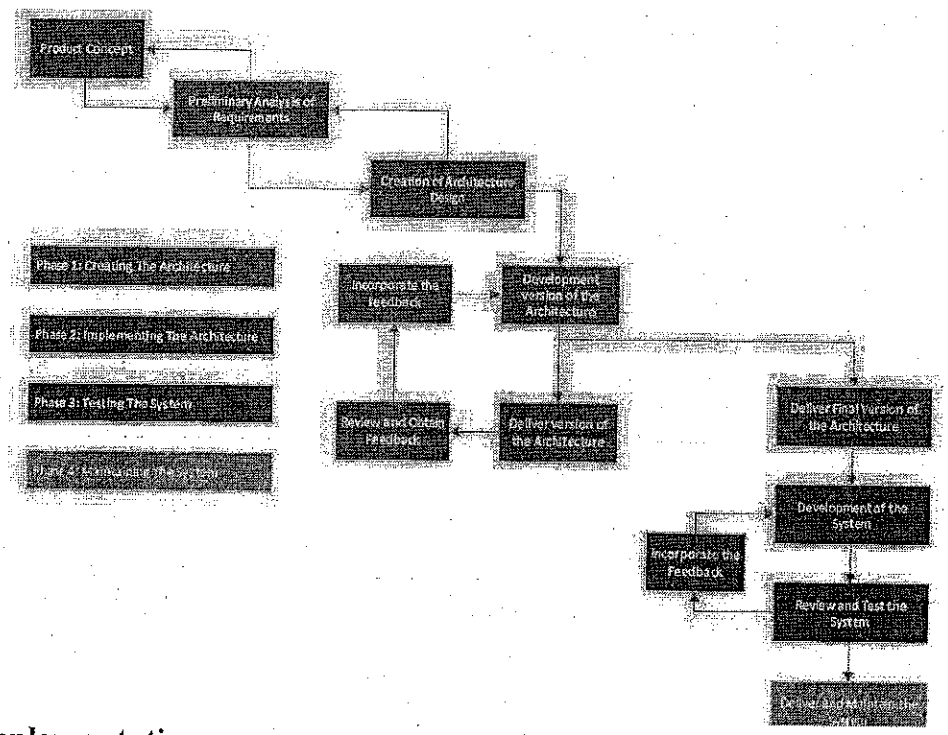
GRAPH 2. Model of the high-level embedded system architecture (adapted from Noergaard 2005).

and spiral model. Most of the many models used in system design are based on single model.

bangmodel,noplanningisexecutedbeforedevelopingthesystem.Thecode-and-fixmodel requirements are described but no processes are prepared before the start of the development.

Thewaterfallmodelusesstrictprocessfordevelopingsysteminsteps,eachpartofthesystemis developedstepbystep.Spiralmodelisasimilarmodeltowaterfallmodel.Betweenstepfeedbackand systematicprogressmodelisshowninGraph3.IntheGraph3waterfallandspiralmodelshavebeen combinedintoonesystemdesignmodel.(Arduino2005).

GRAPH 3. Design and Development Lifecycle Model (adapted from Arduino 2005).



Implementation

Oneofthefinalphasesofembeddedsystemdesignforimplementingthedesign.Basicallythedesign andimplementationofcontrolsystemormorefrequentlyseparatedwhichproducesdevelopmentof embeddedsystemtobemoretimeconsumingandcostly.

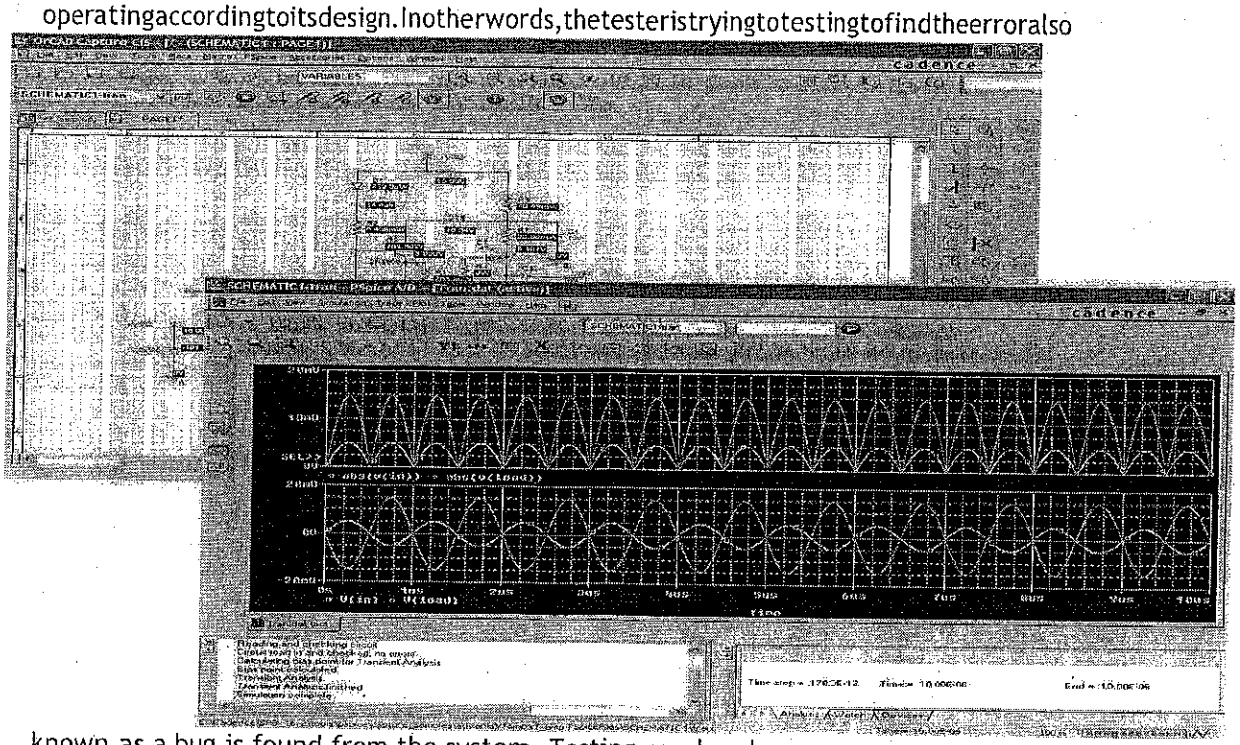
theimplementationphase.Thereareseveraltoolsbuilttoeasetheimplementationofthesystem.The implementationanddevelopmentprocessoftheembeddedsystem'shardwareandsoftwarelayercan bepossiblewithdevelopmenttools.OneofthetoolsusedonthehardwaresideisComputer-Aided Design(CAD).CADisusedforsimulatecircuitsatelectricallevel.Inadditiontostudyacircuit'sbehavior beforethefinalcircuitandsoftwareisimplementedsimulationisusedtotestthehardware.Screen

captureofthesoftwareisshowninGraph4.IntegratedDevelopmentEnvironment(IDE)isusedforaid theimplementationofthesoftwaresidepresentinembeddedsystems.Moreinformationabout IntegratedDevelopmentEnvironmentisfoundinchapter3underArduinoIDE.(ArduinoUNO.)

GRAPH 4. CAD software for circuit simulation

Testing

Thepurposeoftestingistocheckthequalityofasystem.Thetesteristryingtotestifthesystemis operatingaccordingtoitsdesign.Inotherwords,thetesteristryingtotestingtofindtheerroralso



known as a bug is found from the system. Testing can be also

usedtotrackifbugshavebeenfixed.Arduinodescribesinthequotationwhatbugsareandhowthey behaveinasystem.

During testing, bugs usually stem from either the system not adhering to the architectural specifications—i.e.,behavinginawayitshouldn'taccordingtodocumentation,notbe-havingina wayitshouldaccordingtothedocumentation,behavinginawaynotmentionedindocumentation—ortheinabilitytotestthesystem.(Arduino2005,563).

Thetypesofbugsencounteredintestingdependonthetypeoftestingperformedtothe system.Therearefourgenerallyusedmodelstotestthesystem:staticblackboxtesting,static whiteboxtesting,dynamicblackboxtesting,ordynamicwhiteboxtesting.Table1shows techniquesusedfortesting.Blackboxtestingmeansthatatesterhasnoaccesstoschematics orsourcecode.Blackboxtestingisbasedongeneralproductinformationgiventothetester. Whiteboxtestingalsoknownasclearboxorglassboxtestingiswherethetesterhasaccess tothesourcecodeandtheschematicsofthesystem.Statictestingisusedwhenthesystem isnotrunning,whereasdynamictestingisappliedwhenthesystemisrunning.

	Black Box Testing	White Box Testing
Static Testin g	Testing products features by: 1. Expecting for high-level fundamental problems, omissions (i.e., pretending tobe customer, research existing guidelines,review and test similarsoftware,etc.). 2. Low-level specificationtesting by insuring completeness, accuracy,preciseness,	Process of methodically revising hardwareandcodetobugswithout executingit.

Dynamic Testing	relevance,consistency, feasibility, etc. Requirements of definition of what hardware and software does, includes: • *datatesting*,whichis information of user outputandinput. • *boundary condition testing*, which is used to testconditionsatedgeof planned operational limits ofsoftwares • *internal boundarytesting*, which is used to test powers-of- two, ASCII table • *input testing*, which is usedtotestnull,invalid data • *state testing*, which are testing modes and transitionsbetweenmodes soft-wareisinwithstate variables i.e., race conditions, repetition	Testingrunningsystemduringseeing at schematics,code,etc. Directlytestingofhigh-levelandlow-level based on detailed operational knowledge, memory dumps and accessingvariables.Lookingfordata declaration errors, data reference errors, comparison errors, computation errors, control flow errors,subroutineparametererrors, I/O errors,etc.

	testing, stress (starving software = low memory, slow CPU, slow network), load(feed	

ARDUINO

Fordevelopingcomputersembeddedsystemaudinoisisusedasopensourcetoolthatitcan senseandcontrolbemoreofthephysicalworldthatthedesktopcomputers.Ireceivinginput fromvarietyofsensorsitcansensetheenvironment.Discerningsauaffectedbyordernoby itscontrollinglights,motorsandotheractuators.Itcanalsosupportstandaloneapplications l. It can easily integrated with computer / other process.

Brief History of Arduino

In2005,aprojectwasinitiatedtomakeadeviceforcontrollingstudent-builtdesignprojects thatwaslessexpensivethantheotherprototypingsystemavailableatthetime.

FoundersMassimoBanziandDavidCuartiellesnamedtheprojectsafterArduinoofIvreaand beganproducingboardsinasmallfactorylocatedinIvrea.

Features of Arduino UNO:

Microcontroller	: ATmega328
OperatingVoltage	:5Vand3.3V
InputVoltage	:6-20V
DigitalI/OPins	:14
AnalogInputPins	:6
DCCurrentperI/OPin	: 40mA

DCCurrentfor3.3VPin : 50mA

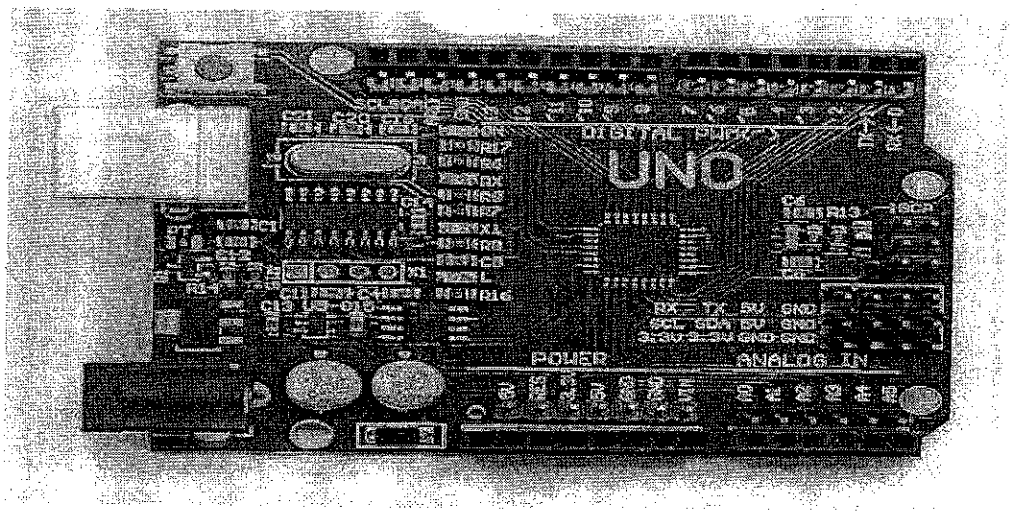
FlashMemory : 32KB(ATmega328)

Boot Loader requirement :512B

SRAM : 2KB(ATmega328)

EEPROM : 1KB(ATmega328)

ClockSpeed : 20MHz

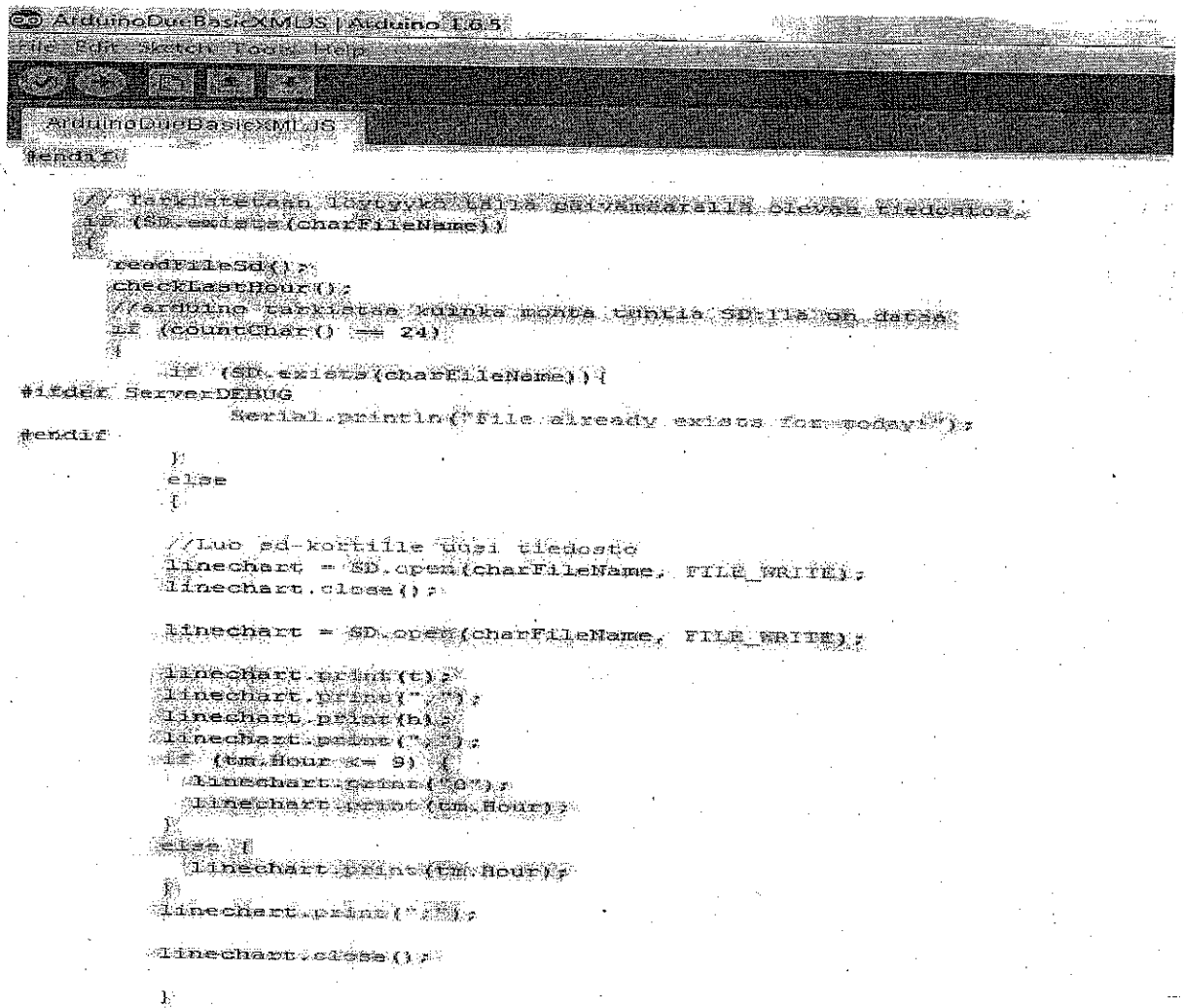


ArduinoIDE

ArduinoIDE(IntegratedDevelopmentEnvironment)isprogrammingprototypewhichcanlet theusertodraftvariouskindofprogramsandloadthemintotheArduinomicrocontroller. Wecanalsobeprogrammedbyusingotheri DEstoo,likeEclipse.ArduinoIDEismoreversatile. ThisArduinoIDEneedsnospecialdriversoradditionalcomponents.Thisisavailablefor Windows,LinuxandMac.Crosscompiler-compilesforadifferenttargetplatformthantheone beingprogrammed.

IDE- SOFTWARE TOOL FOR PROGRAMMING:

- Fileoperationsandothergeneraloptionsontop.
- Buttonsformostcommonlyusedoptions(Verify,Upload,etc.).
- Mainwindow-Texteditorforwritingcode.
- Messagearea-formessagestotheprogrammer.



OPTIONS BUTTONS:

Buttons on the top have the most common, useful operations

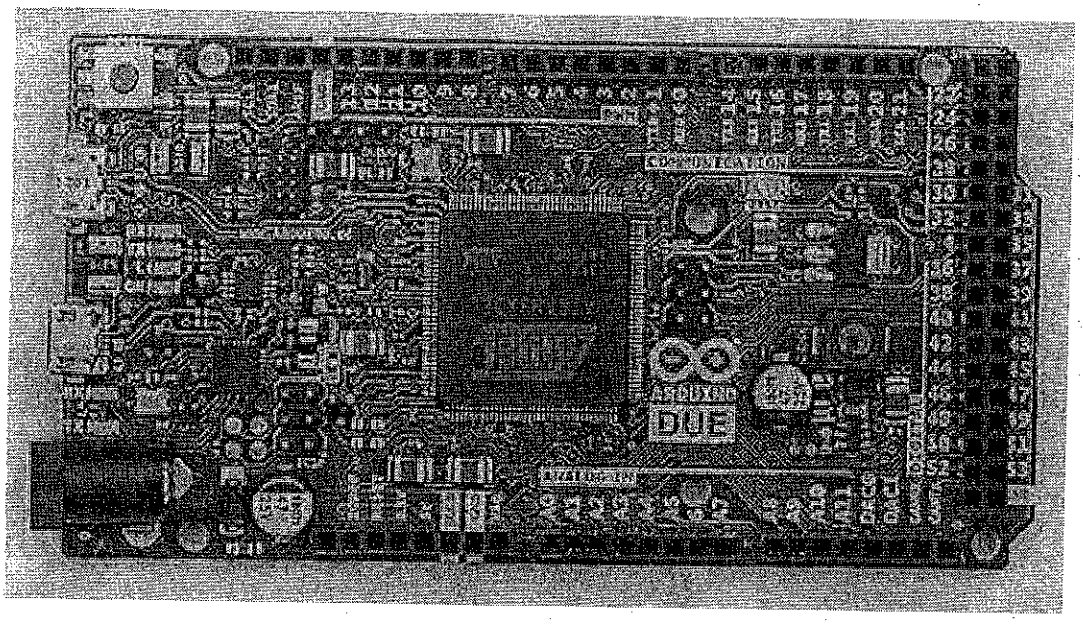
- Verify-Compilesthecodeandchecksforerrors.
- Uploads-Uploadsthecompiledcodetotheboard.Worksonlyiftheboardisconnected.
- New-Createsanewsketch,anewprogram.
- Open-Opensanexistingsketch.
- Save-Savesthecurrentsketchinthedirectoryofyourchoice.
- SerialMonitor-Openswindowtocommunicatewiththeboard.
- RequireJavaRuntimeEnvironment.
- WritecodesingeneralClanguage.
- Setup()-Initiatesthevariablesandsetsupdeviceinstances.
- Loop()-Runsthecodethatcontainsoperationsandmanipulations,iteratesinfinitely.

ArduinoDue

TheArduinoDueisoneofthemicrocontrollerboardrealatedontheAtmelSAM3X8EARM Cortex-M3CPU.ThefirstArduinodependsdona32-bitARMcoremicrocontrollerisArduino due.Itconsistsof54digitali/opinsinthat12CanbeusedforPWMoutputs,12analog inputs,4UARTs,a84MHzclock,anUSBOTGcapableconnection,2DAC,2TWI,apowerjack, anSPIheader,aJTAGheader,aresetbuttonandanerasebutton.

FEATURES OF ARDUINODUE:

Microcontroller	:AT91SAM3X8E
OperatingVoltage	:3.3V
FlashMemory	:96KB(64KBand32KB)
ClockSpeed	: 84MHz
Length	: 101.52mm
Width	: 53.3mm
Weight	:36g



ArduinoShields

AudinoShieldsaretypeofboardsthatisusedtobepluggedontopoftheArduinoPCBused toprotectandprovideadditionalcapabilities.

There are different types of shields. They are

Motor Shield:

Amotorshieldisacircuitboardthatcontainsamotordriverchiptodrivemotors.Itisused tocontroltheworkingspeedanddirectionofthemotor.

Ethernet Shield:

It allows the user to easily connect Arduino to the internet with an internet connection.

Wi-Fi Shield:

It connects the Arduino to the internet wirelessly.

MP3 Player Shield:

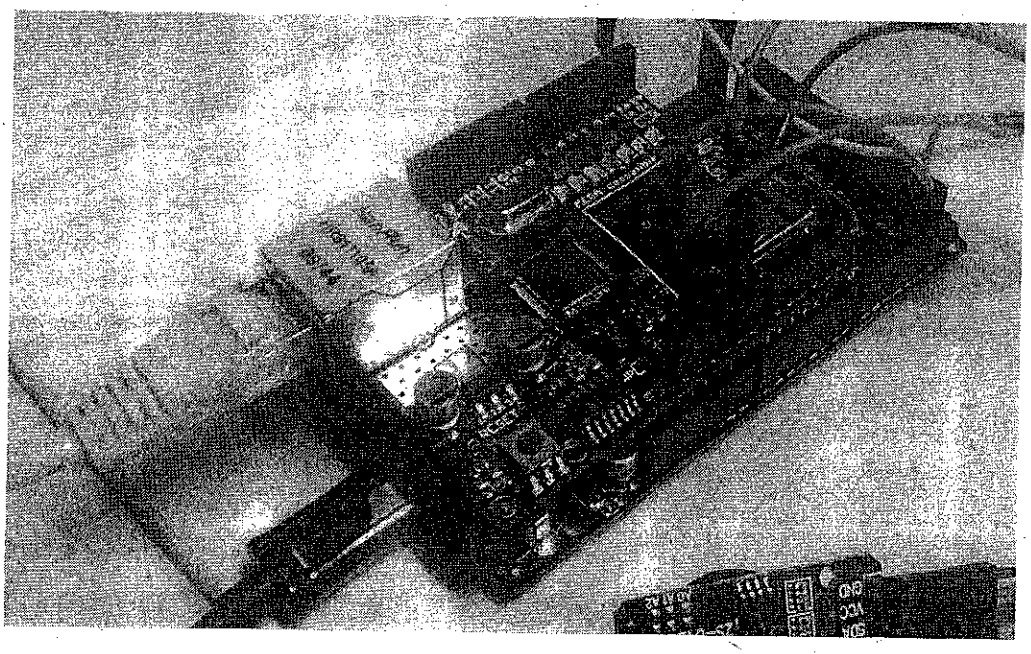
It is used to play any mp3 file from a micro SD card which is mounted on the board.

Features:

- OperatingVoltageis5Vto12V.
- MotorcontrollerL298P,Drives2DCmotorsor1steppermotor.
- Maximumcurrentif2Aperchannel.
- Currentsensingis1.65V/A.
- Freerunningstopandbreakfunction.

Input and Output

- Twoseparatechannels,calledAandB.
- Use4oftheArduinopinstodriveorsensethemotor.
- 8pinsinuseonthisshield.
- EachchannelseparatelytodrivetwoDCmotorsorcombinetodriveonebipolar steppermotor.

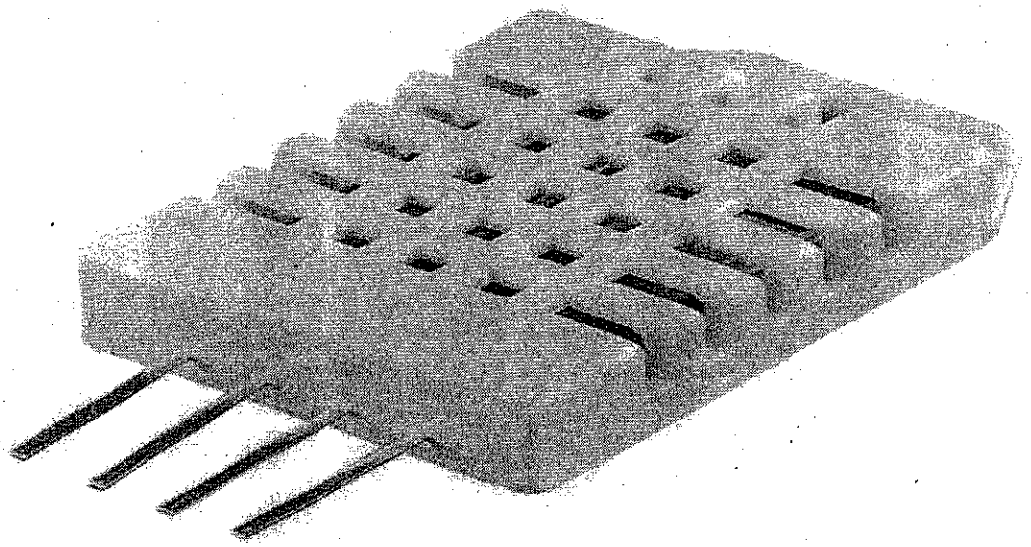


Sensors

Therequirementofasensoristoreactforinputphysicalpropertyandconvertitintoan electricalsignalthatissuitablewithelectroniccircuits(Fraden2010,2).Sensorsareelectronic devicesthatmeasureaphysicalqualitysuchaslightortemperatureandconvertittoa voltage.ExampleofdigitaltemperatureandmoisturesensorispresentedinGraph9.There aretwotypesofsensors:digitalandanalog.

Theoutputofdigitalsensorwhereisbetween0and1whichcantranslatetosensorsvoltage range.Analogsensorcanoutputanyvaluebetweenitsvoltageranges.Accordingtothe readingfromthesensorchangesitsoutput.DigitalsensoroutputisON(1)often5v,orOFF (0),0v.Analogsensorisusedtomeasureprecisenumericalinformationliketemperatureor speed.Analogsensorscanoutputalmostaninfiniterangeofvalues.Outputpinofsensor connectedtoinputpinaforedenimshowthedigitalformisobtainedbytheconversionof data.

Somesensorshaveanalogtodigitalconverterembeddedtothesensorsothedataisoutputted as digital data. After data is processed to digital form, it can be processed on the microcontroller.(Karvinen&Karvinen,2014.)



ULTRASONIC

Theworkingprincipleofultrasonicsensorissameastheworkingprinciplesofradarsystem. Itcanconverttheelectricalenergyintotheacousticandviceversa.Theultrasonicwaveis nothingbut,aacousticwavesignalwhichcantravelatafrequencyabove18kHz.Themost famoussensorHCSR04generatesultrasonicwavesof40kHzfrequency.Forcommunication withanultrasonicsensoramicrocontrollerisused,wheremicrocontrollersendsatrigger signaltoultrasonicsensor.Thedutycycleofthetriggersignalis10uSfortheHC-SR04ultra sonicsensor.Whentriggedtheultrasonicsensorgenerateseightacousticwaveburstsand initiatesatimercounter.Assoonasthereflectedsignalisreceivedthetimerstops.Theoutput oftheultrasonicsensorisahighpulsewiththesamedurationasthetimedifferencebetween transmittedultrasonicburstsandthereceivedechosignal.

Features

Here'salistofsomeoftheHC-SR04ultrasonicsensorfeaturesandspecs:

PowerSupply:+5VDC

QuiescentCurrent:<2mA

WorkingCurrent:15mA

EffectualAngle:<15°

RangingDistance:2cm–400cm/1"–1

Resolution: 0.3cm

Measuring Angle: 30 degree

Trigger Input Pulse width: 10uS

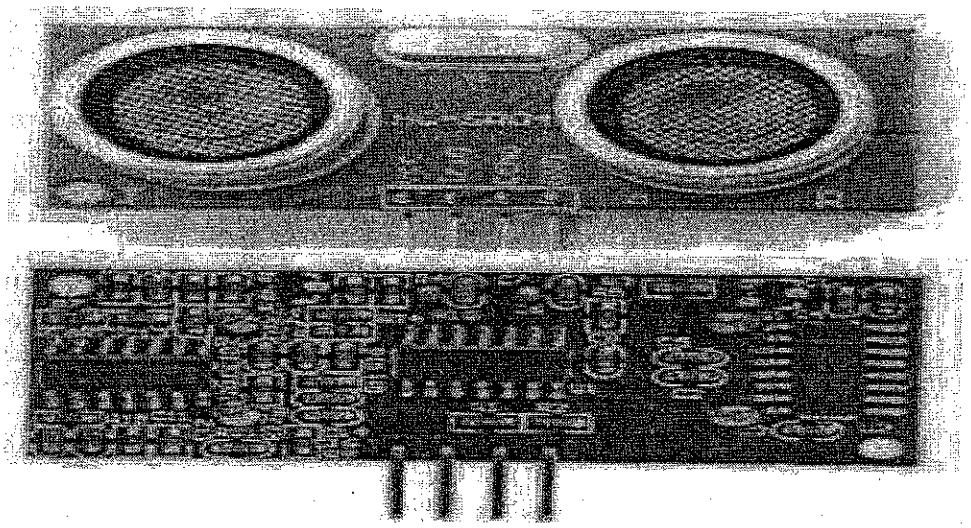
Dimension: 45mm x 20mm x 15mm

How Does it Work?

Theultrasonicsensorusessonartodeterminethedistancetoanobject.Here'swhat happens:

The transmitter (trig pin) sends a signal: a high-frequency sound.

When the signal finds an object, it is reflected and the transmitter (echo pin) receives it.



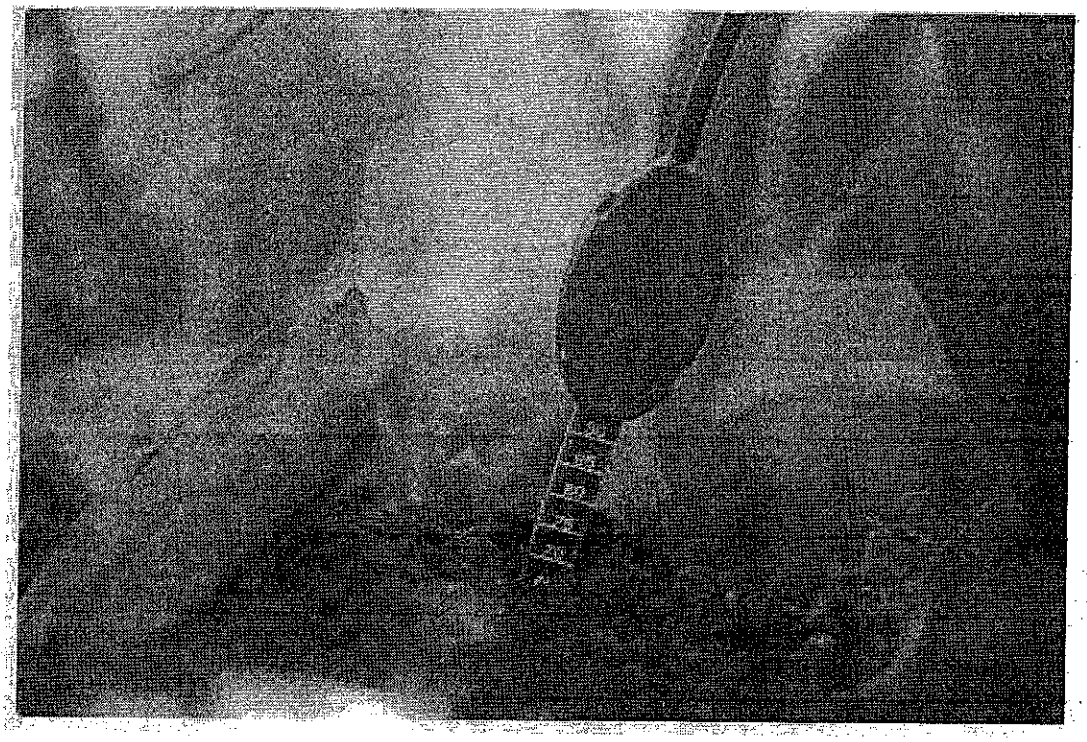
SOIL MOISTURE

.Thevolumetriccontentinsoilismeasuredbythesoilmoisturesensors.Removing,dryingand weightingofasamplehowtherequirementsfordirectgravimetricmeasurementoffreesoil moisture.Byusingsomeotherpropertyofsoil,soilmoisturesensorsmeasurethevolumetric watercontentindirectly.Dependinguponthеenvironmentalfactorssuchassoiltype, temperatureobstacletherelationshipbetweenmeasurepropertyandsoilmoisturemustbe calibratedandmayvary.

Reflectedmicrowaveradiationisaffectedbythesoilmoistureandisusedforremotesensin ginhydrologyandagriculture.Portableprobeinstrumentscanbeusedbyfarmersor gardeners.

Soilmoisturesensorstypicallyrefertosensorsthatestimatevolumetricwatercontent. Anotherclassofsensorsmeasureanotherpropertyofmoistureinsoilscalledwaterpotential; these sensors are usually referred to as soil water potential sensors and include tensiometersandgypsumblocks.Soilmoisturecauseseffecttothedefectmicrowave radiationandthisrefractionmicrowaveradiationusedforremotesensinginhydrologyand agricultural.Farmersandgovernorsareusedtheportableprobeinstruments.Thesensorsthat estimatevolumetricwatercontentisreferredbythesoilmoisturesensors.specialclassof

sensorsmeasuresomespecificpropertyofmoistureinsoilwhichisknownaswaterpotential. Dispenserarealsoknownasandreferredtosoilwaterpotentialsensorswhichinclude tensiometresandgypsumblock.



Rugged and Waterproof VH400 Soil Moistures for Professionals

Heartbeat Sensor

Thedesigningofheartbeatsensorishappenedinsuchawaythatwhenfingerisplacedonit thedigitalvalueoftheheartbeatisdisplayed.Theheartbeatishappenedwhenthebloodis circulatedfromoneparttootherpartinthehumanbody.

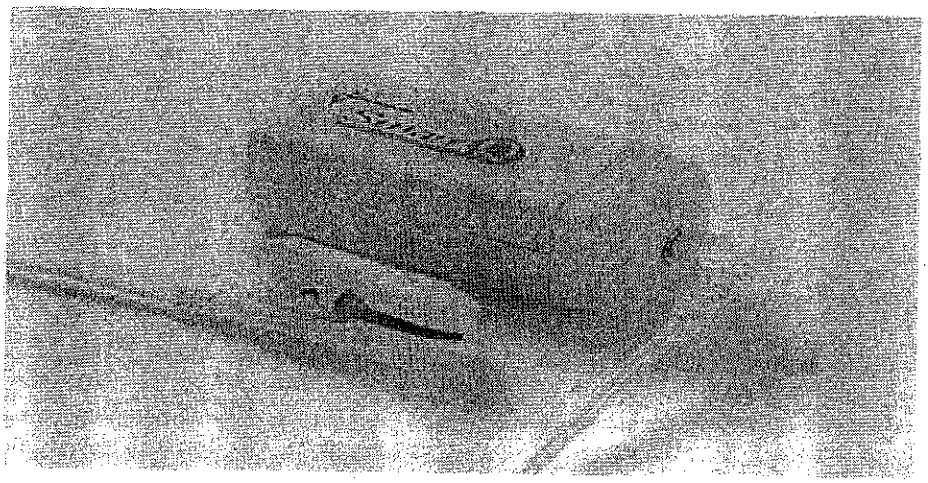
Two Ways to Measure a Heartbeat

ManualWay:Heartbeatcanbecheckedmanuallybycheckingone'spulsesattwolocations- wrist(theradialpulse)andtheneck(carotidpulse).Theprocedureistoplacethetwofingers (indexandmiddlefinger)onthewrist(orneckbelowthewindpipe)andcountthenumberof pulsesfor30secondsandthenmultiplyingthatnumberby2togettheheartbeatrate.How-

everpressureshouldbeappliedminimumandalsofingersshouldbemovedupanddowntill thepulseisfelt.

Principle of Heartbeat Sensor

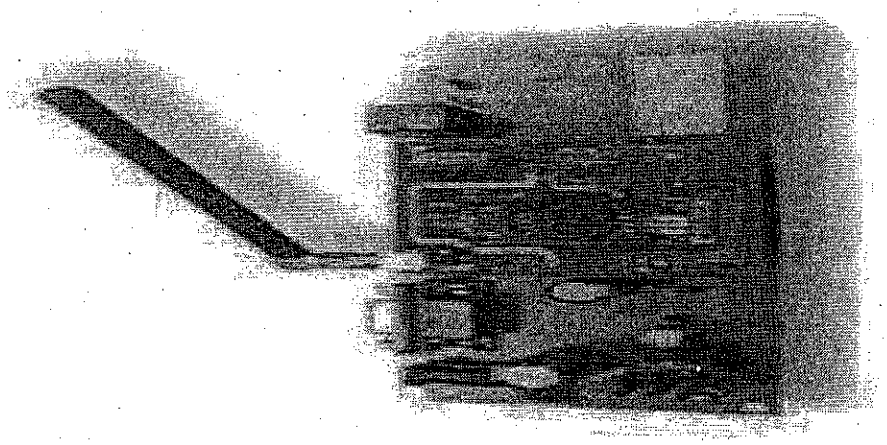
Theprincipleofheartbeatsensorisbasedontheprincipleofphotoplethysmography.With eachheartbeatitcanmeasurethebloodvolumealternationatthefingertip.Thissensor unitconsistsofphotodiodeandIRLEDwhichisplacedsidebyside.Thephotodiodeisusedto sensetheportionoflightwhichisreflectedbackandIRdiodeisusedtotransmitaninfrared lightintotothefingertip.Bloodvolumeinsidethefingertipwillrepresenttheintensityoflight. Eachandeveryheartbeatslightlywhereistheamountofreflectedinfraredlightwhichcan bedetectedbyphotodiode.



GSM

FullformofGSMisGlobalSystemforMobilecommunication.GSMmodemisadevicethatis usedforthecommunication.TheGSMideawasfirstlydevelopedatBellLaboratoriesin1970. Now,intheworldthemostwidelyusedcommunicationsystemisGSM.Thefrequencybands thatareusedinopenanddigitalcellulartechnologiesare850MHz,900MHz,1800MHzand 1900MHzforthetransmittingofmobilevoiceanddataservices.

TheGSMsystemwasdevelopedasadigitalsystemforcommunicationpurposebyusing techniqueoftimedivisionmultipleaccess(tdma).TheGSMisusedtotodigitiseandreduce thedatathenitwillsendreduceddataDon3channelwhichintwodifferentstreamsofclient dataeachonehasitsownparticulartimeslot.thecapabilityofdigitalsystemtocarryadata of64kbpsto120mbpsofdatarates.



GSM Architecture

A GSM network consists of the following components.

- **AMobileStation:**Itisthemobilephonewhichconsistsofthetransceiver,thedisplayand theprocessorandiscontrolledbyaSIMcardoperatingoverthenetwork.
- **BaseStationSubsystem:**Itactsasaninterfacebetweenthemobilestationandthe networksubsystem.ItconsistsoftheBaseTransceiverStationwhichcontainstheradio transceiversandhandlestheprotocolsforcommunicationwithmobiles.
- **NetworkSubsystem:**Itprovidesthebasicnetworkconnectiontothemobilestations.The basicpartoftheNetworkSubsystemistheMobileServiceSwitchingCentrewhichprovides accesstodifferentnetworkslikeISDN,PSTNetc.ItalsoconsistsoftheHomeLocation RegisterandtheVisitorLocationRegisterwhichprovidesthecallroutingandroaming capabilitiesofGSM.ItalsocontainstheEquipmentIdentityRegisterwhichmaintainsan accountofallthemobileequipmentwhereineachmobileisidentifiedbyitsownIMEI number.IMEIstandsforInternationalMobileEquipmentIdentity.

Features of GSM Module:

- Spectral/Frequencyefficiency.
- Supports Internationalroaming.
- Abilitytosupportmultiplehandhelddevices.
- Clear VoiceClarity.
- Lowpoweredhandhelddevices.
- Usesencryptiontomakephonecallsmoresecure.
- EaseofaccessingtheNetwork.

ThemostsecuretelecommunicationstandardofGSMsystemconcurrentlyaccessiblebyusing securitystrategiesstandardises.Theradiochannelcanshoottheconfidentialofthecallon thesecrecyofaGSMsubscriber.Thisishugestepinachievingendtoendsecurity.

GSM Modem

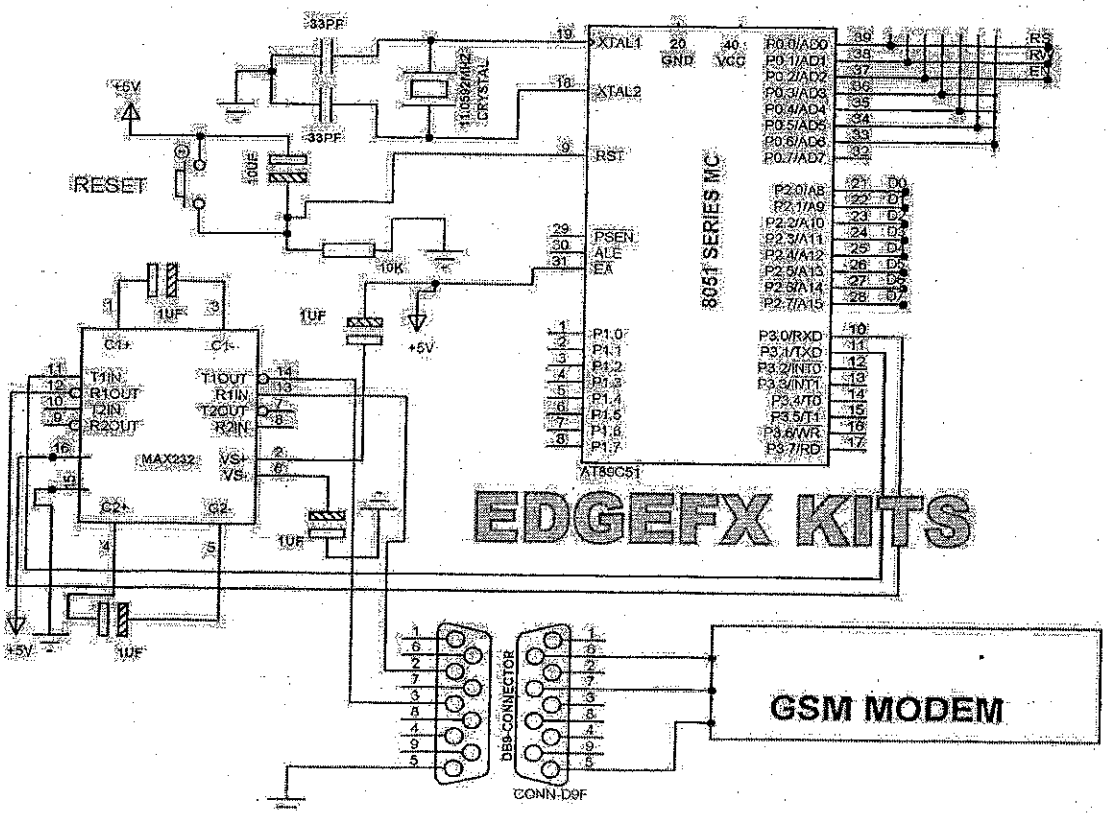
AGSMmodemisadevicewhichisverysimilartomobilephone,whereasthemobilephone isoperatedbyhumanbeingsbuttheGSMmodemisoperatedbyamicrocontrollers.GSM modemisalsousedtomakeapersonalcomputeroranyotherprocessorcommunicateover anetwork.AGSMmodemrequiresaSIMcardtobeinsertedinordertooperatesovera networkrangesubscribedbythenetworkoperator.Itcanbeconnectedtoacomputer throughserial,USBorBluetoothconnection.

AGSMmodemcanalsobeastandardGSMmobilephonewiththeappropriatecableand softwaredrivertoconnecttoaserialportorUSBportonyourcomputer.Apersonalcomputer isconnectedwithUSBtocombconnectorandcompoundDB9connectorisconnectedtoGSM modem.So,thatthesetuphasbeenformedtochecktheGSMmodembyusingcommands thatarecompatibletotheOperatingSystem.TheGSMmodemhaswiderangeofapplications intransactionterminals,supplychainmanagement,securityapplications,weatherstations andGPRSmoderemotedatalogging.

Working of GSM Module:

FromthebelowcircuitaGSMmodemdulyinterfacedtotheMCthroughthelevelshifterIC Max232.TheSIMcardmountedGSMmodemuponreceivingdigitcommandbySMSfromany

cellphonesendthatdatatotheMCthroughserialcommunication.Whiletheprogramis executed,theGSMmodemreceivescommand'STOP'todevelopanoutputattheMC,the contactpointofwhichareusedtodisabletheignitionswitch.Thecommandsosentbythe user is based on an intimation received by him through the GSM modem 'ALERT' a programmedmessageonlyiftheinputisdrivenlow.Thecompleteoperationisdisplayedover 16×2 LCDdisplay.



Intelligent GSM Device for Automation and Security

Inthesedays,theGSMmobileterminalhasbecomeoneoftheitemsthatareconstantlywith us.Justlikeourwallet/purse,keysorwatch,theGSMmobileterminalprovidesusa communicationchannelthatenablesustocommunicatewiththeworld.Therequirementfor apersontobereachableortocallanyoneatanytimeisveryappealing.Inthisproject,

AsthetitleofprojectisdependingontranslationofSMSfromsendertoreceiverbyusingGSM Networktechnologyoftransmission.Ubiquitousactressofapplianceandbridgecontrolat homewhereusedbytheSMSsendingandreceiving.Twosubsystemsareproposedbythe system.Tocontrolthehomeapplianceremotelyusingapplliancecontrolsubsystemandfor automaticsecuritymonitoringyouareusingsecurityalertsystem.

AccordingtotheuserneedsandrequirementsthesystemiscapableenoughtosendinstructionsandconditionsviaSMSfromaparticularcellnumber.Onthedetectionof intrusionthesecretariatcanbeachievedthisisoursecondaspectthoughalertingtheuser againstsecurityriskthesystemallowsautomaticgenerationofSMS

Software

TheArduinoDuemilanovecanbeprogrammedwiththeArduinosoftware.TheArduino IntegratedDevelopment(IDE)isderivedfromtheIDEfortheprocessingprogramming languageandiswritteninJava.Ithascodeeditorfeaturessuchasthesyntaxhighlighting braces matching automatic indentation indentation and is also capable of compiling and uploadingprogramstoaboardwithasingleclick.AprogramorcodewrittenforArduinois calleda"sketch".ArduinoprogramsarewritteninCandC++.

Loop():

- Calledaftercreatingsetupfunction.
- Function loopsconsecutively.
- Allowstheprogramtochangeandrespond.
- ActivelycontrolstheArduino.

Compilercompilesthesketchafterwritingthecodeandchecksthecodeforerrors.Upload theprogramtotheconfiguredArduinoboard.Checktheboardwhetheritwillworkswellor not.Ifanyproblem,thenrevisethecodeorcircuityoubuild.

CODE

Temperature and Humidity:

```
#include <dht.h>

#definedht_apinA0//AnalogPinsensorisconnectedto

dhtDHT;

void setup(){

Serial.begin(9600);

delay(500);//Delay to let system boot

Serial.println("DHT11Humidity&temperatureSensor\n\n");

delay(1000);//WaitbeforeaccessingSensor

}//end

"setup()"void

loop(){

//Start of Program

DHT.read11(dht_apin);

Serial.print("Current humidity = ");

Serial.print(DHT.humidity);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(DHT.temperature);

Serial.println("C "); delay(5000);

end loop()
```

TO BLINK AN LED USING EX-OR OPERATION:

```
void setup() {
```

```
pinMode(2,INPUT);
```

```
pinMode(3,INPUT);

pinMode(4,OUTPUT);
}

void loop() {
int a,b;

a=digitalRead(2);

b=digitalRead(3);

if(a==b)
{

digitalWrite(4,0);

}
else
{

digitalWrite(4,1);

}

}
```

TO BLINK THE WORD DISPLAYED IN THE LCD:

```
#include<LiquidCrystal.h>

LiquidCrystallcd(12,11,10,9,8,7);

void setup() {

lcd.begin(16,2);

lcd.setCursor(0,0);

lcd.print("NHCE");

}
void loop(){


lcd.display();

delay(1000);
```

```
lcd.noDisplay();

delay(1000);
}
```

Alcohol Detection:

```
#include

<LiquidCrystal.h>#includ

e <MQ135.h> #include

"MQ135.h"

void setup() {

Serial.begin(9600);

}

void loop() {

MQ135 gasSensor = MQ135(A0);

float air_quality = gasSensor.getPPM();

Serial.println (air_quality);

delay(3000);

}
```

CHAPTER 6

CONCLUSION

Asystematicapproachindesigningasystemformeasurementandalsoforthecontrolofthree importantrequirementpartsbasedonthemicrocontroller.

Ammetersforplantgrowth,temperature,humidityandsoilmoisture,hasbeenfollowed.The

systemwhichisproposedhassuccessfullyovercometheproblemsoftheexistingsystemsby reducingtheconsumptionofpower,complexityandthecostbyprovidingapreciseformof maintaining theenvironment.

Arduino microcontrollers are constantly evolving development platform. Vastly advancing technologycaneasilybypasstechnologyusedinArduinoDueandmakethetechnology outdated.Growingopen-sourcecommunityisconstantlydeveloping.Moreadvancedsoftware isprogrammedtoworksimilarlywithmonitoringenvironmentvariables.

Environment variable monitoring DIY projects are common in open-source communities. Multiplegreenhouseorhouseholdplantmonitoringprojectscanbefoundonline.Thekey factorthatsetsthisgreenhouse-monitoringprojectapartfromotherDIYmonitoringsystem isthesysteminwhichthedataisaccessedeasilybytheuserthroughthewebinterfaces.

Usually web interfacing monitoring systems require multiple hardware to handle hosting services.Tonarrowdownthecostofhostingandmonitoringsystemwascombinedtoone device.

furtherresearch,thesystemcanbedesignedinsuchawaythattooperateinawireless environment..ToprovideaccesstouserinterfaceweareconsideringtheWi-Fitechnology. Andalsowehavealternateoptionsforwirelessconnectivity3Gor4Gcellularconnectivity. Themonitoringofdataaccessirrespectiveofdistancetolocationofthegreenhousewhich willhaveleftandconductionandmonitoringthedatamoreefficiently.

EmbeddedSystems

ORIGINALITY REPORT

17%	%	17%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	T NOERGAARD. "The Final Phases of Embedded Design: Implementation and Testing", Embedded Systems Architecture, 2005 Publication	3%
2	"Smart Water Dripping System for Agriculture/ Farming", International Journal of Recent Technology and Engineering, 2019 Publication	1%
3	Robinson P. Paul, Ghansyam B. Rathod, Vishwa R. Trivedi, Punit V. Thakkar. "Persistence of Vision control using Arduino", International Journal of Intelligent Systems and Applications, 2014 Publication	1%
4	Lakshmi Boppana, Vishal Jain, Ravi Kishore. "Smart Stick for Elderly", 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart	1%

Netaji Subhas Institute of Technology,
Bihta, Patna

Approved by AICTE, New Delhi; Affiliated to Aryabhatta knowledge University
Patna (Bihar)

"INTERNSHIP REPORT"

BASED ON THE INTERNSHIP DONE AT E & ICT ACADEMY,IIT KANPUR IN THE MONTH OF MAY 2022 TO JUNE 2022 IN EMBEDDED SYSTEMS

In partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLGY

With specialization in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:

BASUDEV (202003)

Submitted to :

Department of Electronics and communication Engineering
NSIT Bihta, Patna

TABLE OF CONTENT

- Abstract
- Declaration
- Acknowledgement
- Certificate from the Institute
- Project Overview
  - Introduction
  - Components used
    - ✓ Power Supply
    - ✓ Entry and Exit sensor circuit
    - ✓ ATMEGA328P micro-controller
    - ✓ L293D Driver
    - ✓ LCD Display
    - ✓ Seven Segment
    - ✓ Buzzer
    - ✓ LEDs
    - ✓ Embedded Systems
  - Introduction
  - Components of Embedded systems
    - ✓ Hardware
    - ✓ Software
    - ✓ RTOS (Real Time Operating Systems)
    - ✓ Working
    - ✓ Characteristics of Embedded Systems
    - ✓ Application
    - ✓ Microcontroller
    - ✓ Code Vision AVR
- conclusion

Netaji Subhas Institute of Technology,
Bihta, Patna

Approved by AICTE, New Delhi; Affiliated to Aryabhatta knowledge University
Patna (Bihar)

CERTIFICATE

This is to certify that Project Report entitled "EMBEDDED SYSTEM" that is submitted by BASUDEV in partial fulfillment of the requirement for the award of the degree Bachelor of Technology in Department of ELECTRONICS & COMMUNICATION ENGINEERING in Netaji Subhas Institute of Technology, Bihta(Patna) is a record of the candidate own work carried out by him under my own supervision. The matter embodies in thesis is original and has not been submitted for the award of any other degree.

MR.Rajni Ranjan
H.O.D
Electronics & Communication Department

Date:
NSIT, Bihta, PATNA

ABSTRACT

In the last decade, the open source community has expanded to make it possible for people to build complex products at home.A modern and efficient controller is the LQR – Linear Quadratic Regulator.

Being a state space feedback controller the model has to be a good representation of reality since the output signal depends on the model. In this thesis, the validation process was performed using a Microcontroller. The results showed that the model is not yet reliable. The reasons for this are discussed and recommendations for future development are listed.

One can seek the help of electronic security systems.

Such a portable security system is described here.

This electronic setup carrying sensor can be auto activated whenever the motion is detected near by the Sensor and can auto activate the landline number and redial the last dialed number from the conventional telephone. All we need is to do minor changes to activate the telephone as it works as to become auto dialer circuit.

Thus whenever the intruder enters to the area, it activates the sensor circuit of either sound activation or infrared light beam obstruction circuit, the redial circuit become active and gives a ring tone to the receiving end. It may be a mobile phone or any landline phone or even police control room.

DECLARATION

I hereby declare that this submission is own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extend has been accepted for the award of the award of any other degree or diploma of the university or other institute of the higher leaning except where due acknowledgement has been made in the text.

Signature:

Name: BASUDEV
Roll No.: 202003
Date:

ACKNOWLEDGEMENT

I express my gratitude to the institute **"E & ICT ACADEMY, IIT KANPUR"** for providing necessary facility to carry out my training successfully.

It is my duty to record my sincere thanks and gratitude towards the institute stiff who helped me in bringing this project to its present form. The valuable guidance and interest taken by them has been a motivator and source of inspiration for me to carry out the necessary proceedings for the project to be completed successfully.

I humbly thank each and every personal who explain me the various task being carried out at E & ICT ACADEMY, IIT KANPUR which really boost my practical knowledge and encouraged me to learn more.

Date: .....

Signature

In association with

TEQIP-3

Technical Education Quality Improvement Program

E & ICT Academy, IIT Kanpur

(A Joint Initiative of MeitY & IIT Kanpur)

This is to certify that

Dr. / Mr. / Ms. **BASUDEV**

of

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA

has successfully completed 4 weeks Summer Training on

EMBEDDED SYSTEM

from **6 May 2022 to 9 June 2022**

Performance Level: **Very Good**

Prof. B. V. Phani

Prof. Amey Karkare

Chief Investigators, E & ICT Academy, IIT Kanpur

Project:

1. Introduction Of Project

Project title is "EMBEDDED SYSTEM ".

The objective of this project is to make a controller based model to control the BO/DC Motor every time when the PIR Sensor detects any obstacle or motion. Moreover LCD, Seven Segment and Buzzer are also connected here with the Arduino Microcontroller by which at every detection Buzzer starts beeping whereas Seven Segment and LCD display the way Motor is moving.

In today's world, there is a continuous need for automatic appliances; there is a sense of urgency for developing circuits that would ease the complexity of life.

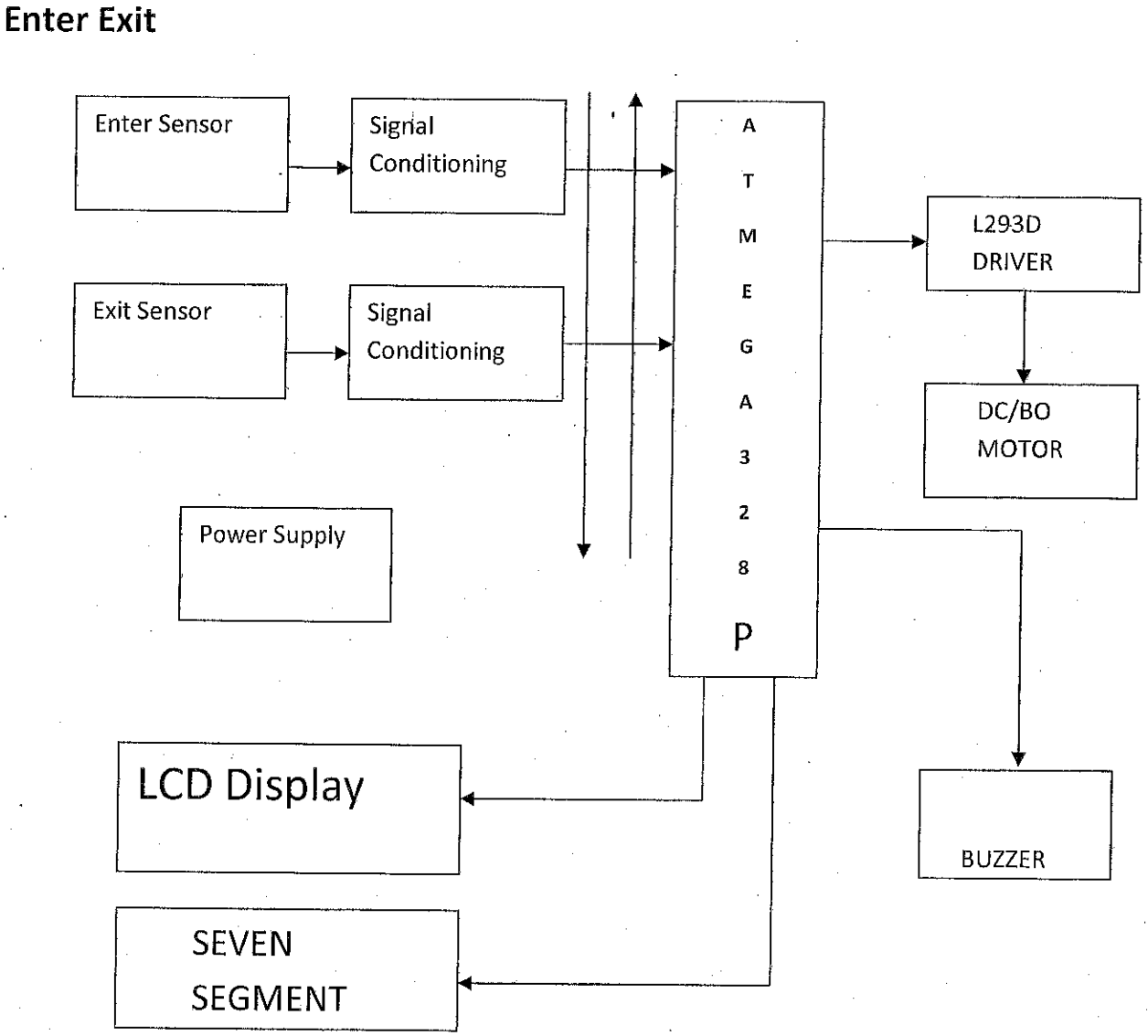
Also if at all one wants to know the number of people present in room so as not to have congestion. This circuit proves to be helpful.

1.1 Project Overview

This Project "EMBEDDED SYSTEM" is a reliable circuit that takes over the task of controlling the DC/BO Motor with the accurate capacity of displaying the way Motor is moving as well as beeping at each detection of obstacles. IR Sensor when detect any obstacles then buzzer starts beeping and the way of movement of Motor changes which is displayed on the LCD and Seven Segment.

The microcontroller does the above job. It receives the signals from the sensors, and this signal is operated under the control of software which is stored in ROM. Microcontroller ATMEGA328P continuously monitor the Infrared Receivers, When any object pass through the IR Receiver's then the IR Rays falling on the receiver are obstructed , this obstruction is sensed by the Microcontroller.

2.1 Basic Block Diagram:-



2.2 Block Diagram Description

The basic block diagram of the "EMBEDDED SYSTEM" is shown in the above figure. Mainly this block diagram consists of the following essential blocks.

- ✓ Power Supply
- ✓ Entry and Exit sensor circuit
- ✓ ATMEGA328P micro-controller
- ✓ L293D Driver
- ✓ LCD Display
- ✓ Seven Segment
- ✓ Buzzer
- ✓ LEDs

1. Power Supply:-

Here we used +12V and +5V dc power supply. The main function of this block is to provide the required amount of voltage to essential circuits. +12 voltagesare given. +12V is given to relay driver. To get the +5V dc power supply we have used here IC 7805, which provides the +5V dc regulated power supply.

2. Enter and Exit Circuits:-

This is one of the main parts of our project. The main intention of this block is to sense the obstacles. For sensing the obstacles and PIR Sensor. By using this sensor and its related circuit diagram we can control and monitor the Motor.

3. ATMEGA328P Microcontroller:-

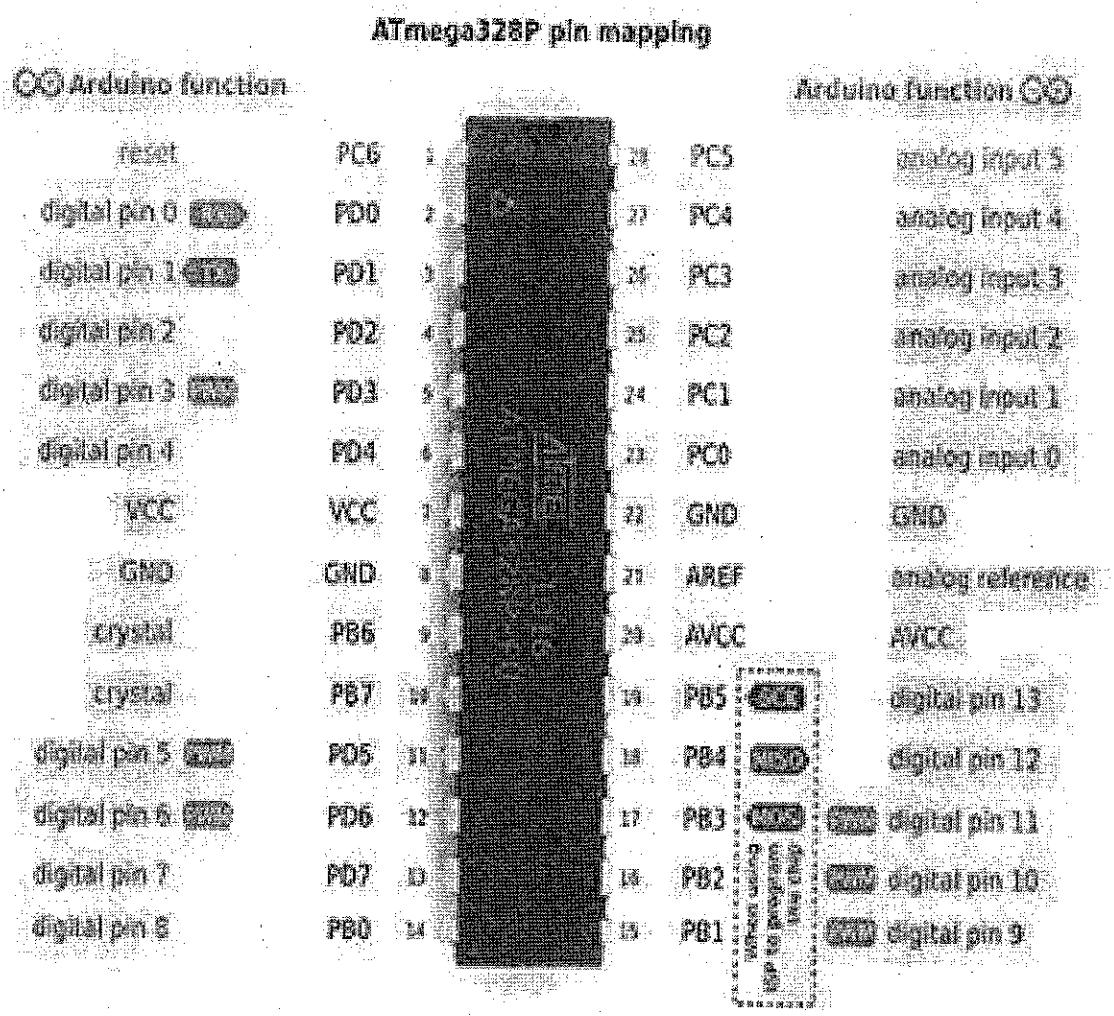
ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on

AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards.

Although we have many controllers ATMEGA328P is most popular of all because of its features and cost. ARDUINO boards are also developed on this controller because of its features.

- With program memory of 32 Kbytes ATMEGA328P applications are many.
- With various POWER SAVING modes it can work on MOBILE EMBEDDED SYSTEMS.
- With Watchdog timer to reset under error it can be used on systems with minimal human interference.
- With advanced RISC architecture, the controller executes programs quickly.
- Also with in chip temperature sensor the controller can be used at extreme temperatures.

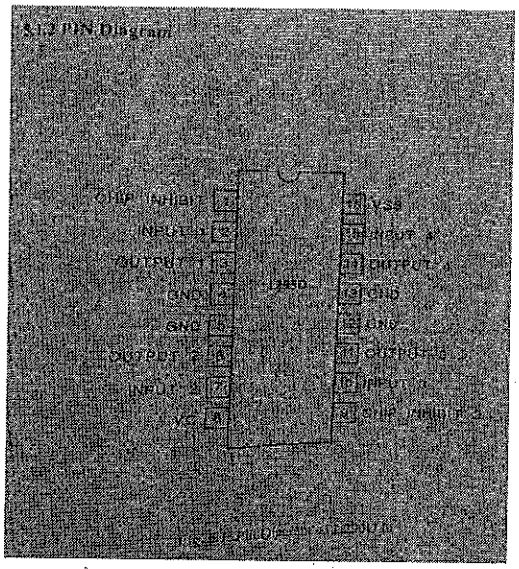
These all features add together promoting ATMEGA328P further.

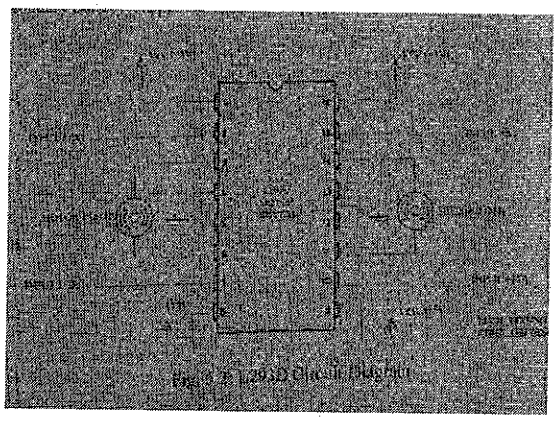


4. L293D MOTOR DRIVER:-

The L293D is a popular 16-Pin **Motor Driver IC**. As the name suggests it is mainly used to drive motors. A single **L293D IC** is capable of running two DC motors at the same time; also the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be

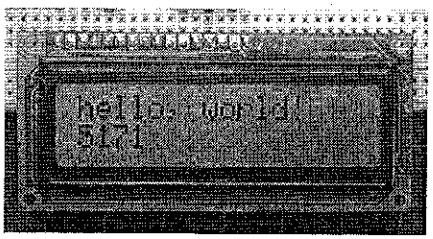
controlled by digital circuits like Op-Amp, 555timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc.. This IC will be the right choice for you.





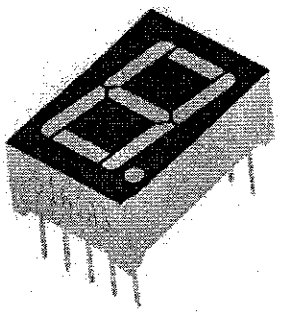
5. LCD DISPLAY:-A **liquid-crystal display** (**LCD**) is a flat-paneldisplay or other electronically modulated optical device that uses the light-modulating properties of liquid

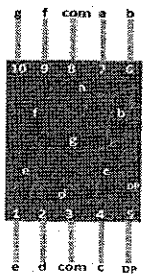
crystals combined with polarizer's. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce

Images in color or monochrome. 

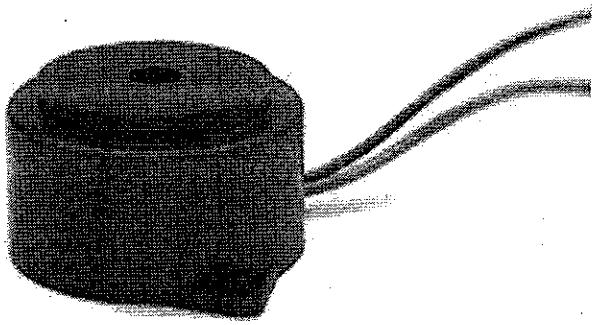
6. **SEVEN SEGMENT**: - A **seven-segment display** is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays.

Seven-segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information.

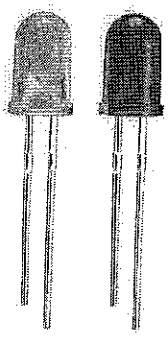


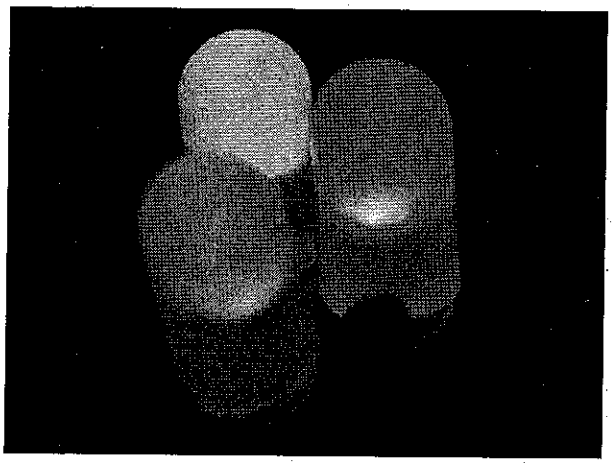


7. BUZZER: -A **buzzer** or **beeper** is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (*piezo* for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



8. LED's:- A **light-emitting diode (LED)** is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.





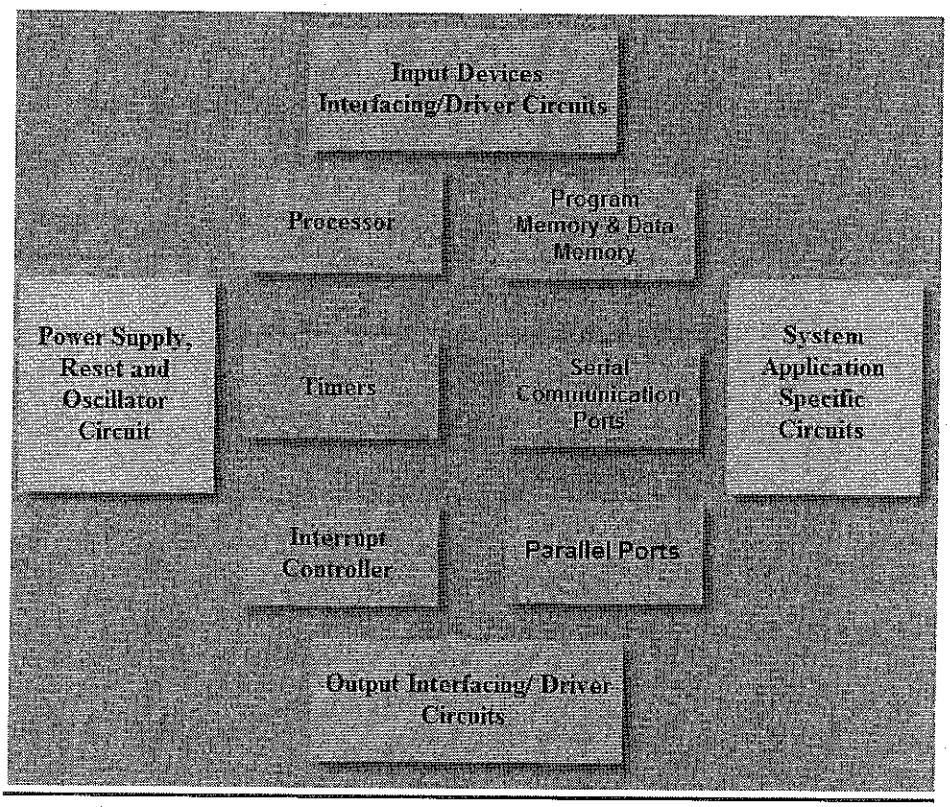
(3.) EMBEDDED SYSTEMS

(1.) WHAT IS EMBEDDED SYSTEM?

- Embedded systems are a combination of hardware and software where software is

usually known as firmware that is embedded into the hardware.

- An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task.
- For example, a fire alarm is an embedded system; it will sense only smoke.



(2.)COMPONENTS OF EMBEDDED SYSTEMS:-

- There are three components of embedded systems:-

(A)Hardware

(B)Software

(C)Real time operating system (RTOS)

(A). **HARDWARE:-**

- An embedded system uses a hardware platform to perform the operation. Hardware of the embedded system is assembled with a microprocessor /microcontroller. It has the elements such as input/ display unit. Generally , an embedded system comprises of the following:
- Power Supply
- Memory
- Processor
- Timers
- Output/Output circuits
- Serial communication ports
- SASC(System application specific circuits)

(B) **SOFTWARE:-**

- The software of an embedded system is written to execute a particular function. It is normally written in a

high – level setup and then compiled down to offer code that can be stuck within a non-volatile memory in the hardware.

- An embedded system software is intended to keep in view of the following three limits;-
  1. Convenience of system memory.
  2. Convenience of processor's speed.
  3. When the embedded system runs constantly, there is a necessity to limit power dissipation for actions like run, stop and wake up.

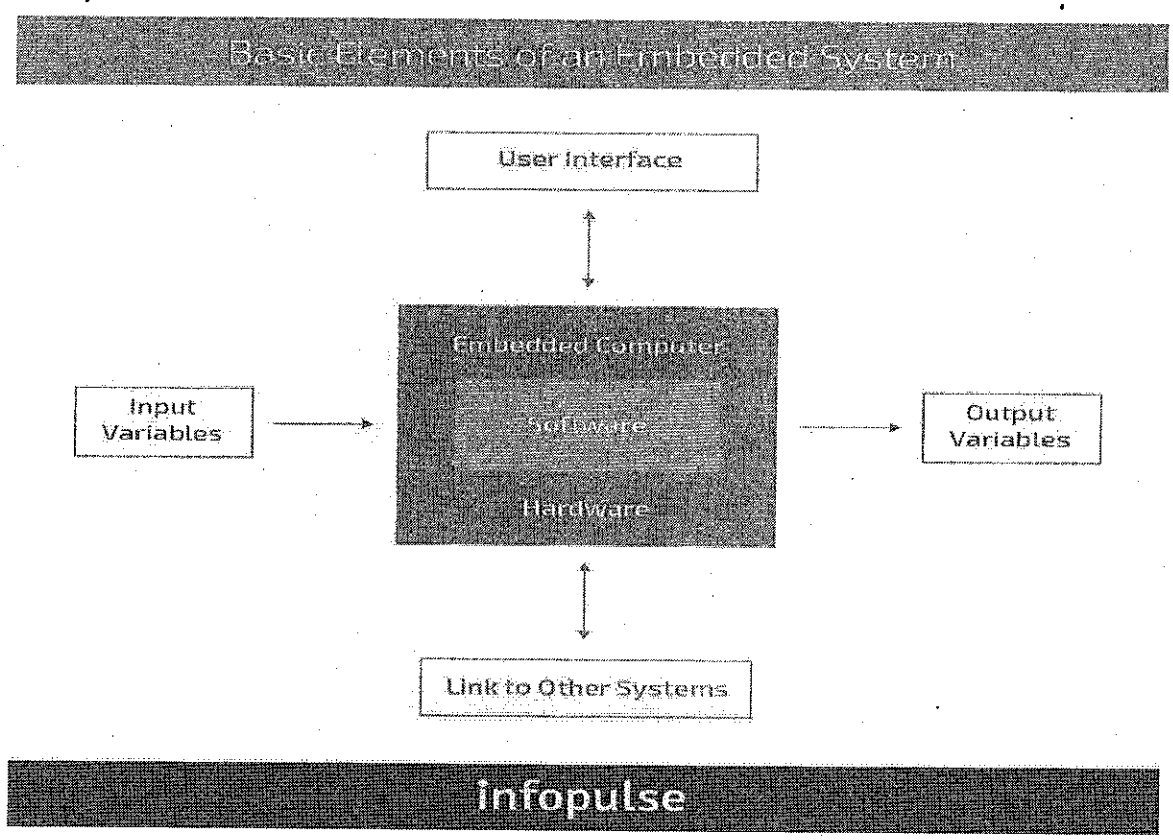
(C) **REAL TIME OPERATING SYSTEM:-**

- A system which is essential to finish its task and send its service on time, then only it said to be a real timeoperating system.
- RTOS controls the application software and affords a device to allow the processor run.
- It is responsible for managing the different hardware resources of a personal computer and also host application which run on the PC.

(3).**WORKING:-**

- Every Embedded computing device may it be your computer or mobile has some inputs and corresponding outputs.

- Here is an example of how an embedded system works.
- It takes the analog/digital input from the user. Examples are push button switch, keyboard, sensor, touch screen.
- Then, it processes the given input. Processing may be calculation or conversion. For examples, an ADC (Analog to Digital converter) converts a sensor analog input to digital output.



(4). **CHARACTERISTICS OF EMBEDDED SYSYTEM:-**

- **Single-functioned-** An embedded system usually performs a specialized operation and does the same repeatedly. For example: A pager always functions as a pager.
- **Tightly constrained-** All computing systems have constraints on design metrics. Design metrics is a measure of an implementation's features such as its cost, size, power, and performance. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- **Reactive and Real time-** Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay.
- **Microprocessors based-** It must be microprocessor or microcontroller based.
- **Memory-** It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- **Connected-** It must have connected peripherals to connect input and output devices.
- **HW**-SW systems- Software is used for more features and flexibility. Hardware is used for performance and security.

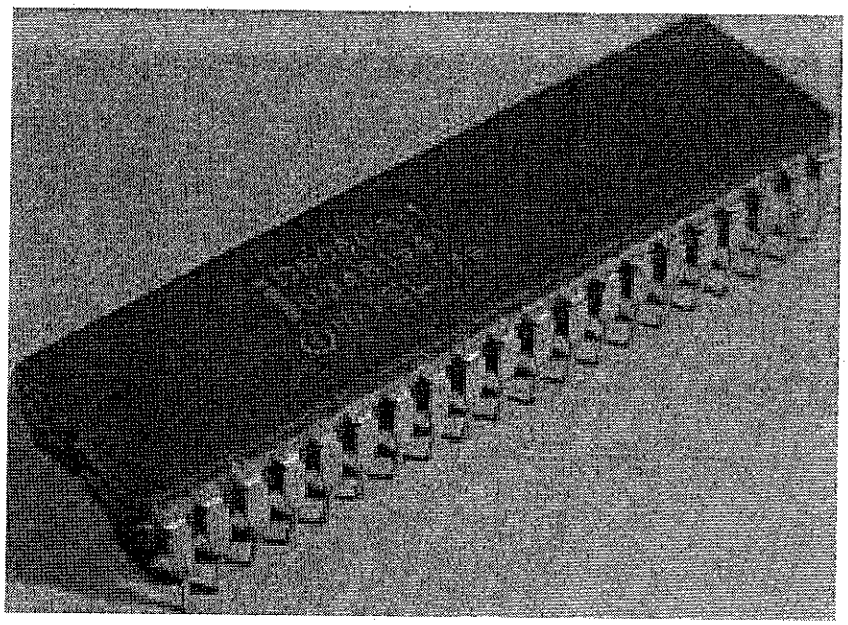
(5). APPLICATIONS:-

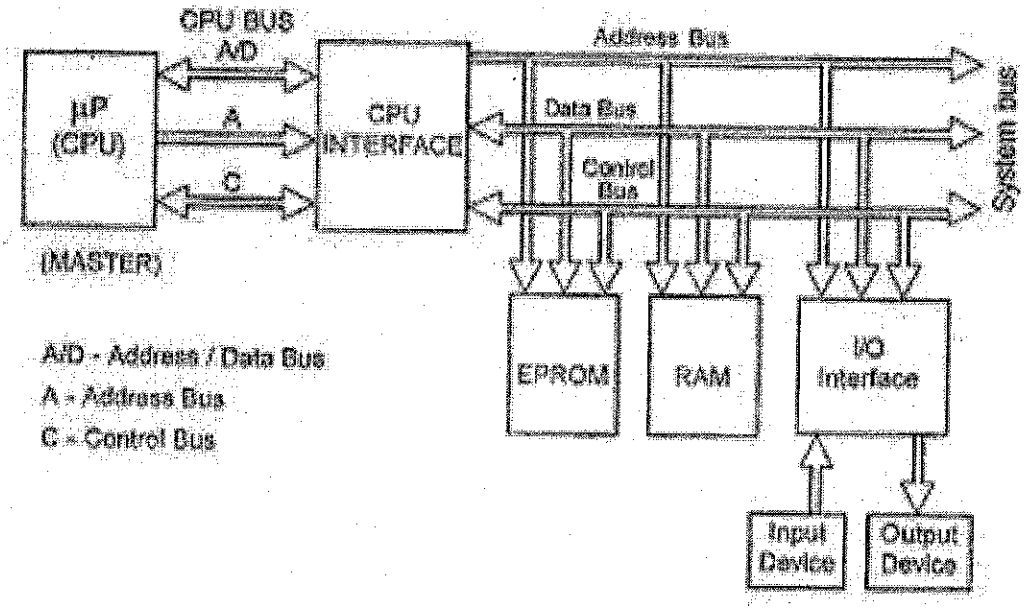
- Embedded systems in automobiles include motor control, cruise control, body safety, engine safety; robotics is an assembly line, car multimedia, car entertainment, E- com access, mobiles etc.
- Embedded systems in telecommunications include networking, mobile, computing, and wireless communication, etc.
- Embedded systems in smart cards include banking, telephone and security systems.
- Embedded system in satellites and missiles include defense, communication, and aerospace
- Embedded systems in computer networking & peripherals include image processing, networking systems, printers, network cards, monitors and displays
- Embedded Systems in digital consumer electronics include set-top boxes, DVDs, high definition TVs and digital cameras.

(6).MICROCONTROLLER:-

- Microcontroller is single chip computer.
- A many more functional block on a chip.

- Microcontroller is a user specific processor.
- It is designed for embedded system.





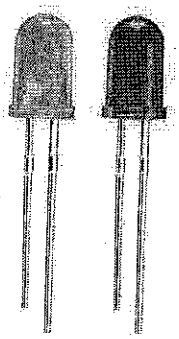
(7).CODE VISION AVR (CVAVR):-

- CVAVR is an IDE (integrated development program).

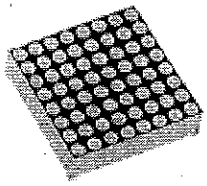
- An IDE is a software having functions like preprocessing, assembly, linking object translation, text editor etc.
- It has limitation of code size.

I/O FUNCTIONS REGISTERS

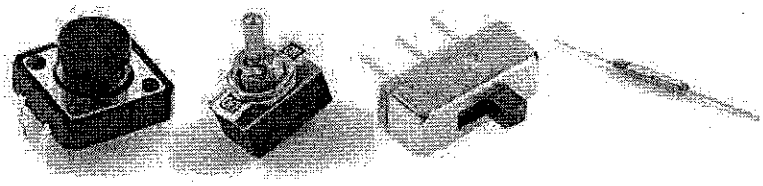
- DDRX----> sets whether a port is input or output of PORTX.
- PORTX---> sets the output value of PORTX. For ex:- PORTA=255(all pins work of port A as output) for LED & LCD
- PINX----> Reads the value of PORTX for ex. PINA, o=o(0th pin of port a working as input) for switches.



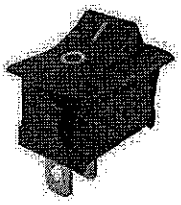
---→**LIGHT EMITTING DIODE(LED)**



--→**LCD DISPLAY (8*8 MATRIX)**



---→ **SWITCHES**



----→**SWITCHES**

CONCLUSION

The world of embedded systems is a dreamer's paradise with unlimited possibilities. Imagine you control all the systems around you just by a simple gesture and the things respond to you as if it was some magic. This could be possible with embedded systems. As microprocessors are becoming smaller and cheaper, more and more products are becoming 'smart' with microprocessors embedded in them. From handheld devices such as personal digital assistants and smart cell phones to automobiles and rocket propellers, embedded systems are at the heart of it all. The demand for embedded systems is, in fact, rising in integrated embedded solutions across various industry verticals. As the world is experiencing groundbreaking research in hardware technology (nano-technology, quantum mechanics, etc), packing more power into single chip will soon be a reality.

INTERNSHIP REPORT

A report submitted in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING

by
SHASHIKANT KUMAR
Regd. No.: 20104103003

Roll No.: 202004

Under Supervision of

Whole team of

INTERNSHALA TRAINING.

(Duration: 28 December, 2022 to 18 February, 2023)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERINGNETAJI SUBHAS INSTITUTE OF TECHNOLOGY
Bihta, Patna, Bihar

BATCH
2020 – 2024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

BIHTA, PATNA, BIHAR

CERTIFICATE

This is to certify that the —Internship report" submitted by SHASHIKANT KUMAR (Regd. **No.: 20104103003) is work done by her and submitted during 2022 – 2023 academic year, in** partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in** ELECTRONICS AND COMMUNICATION **AND ENGINEERING,** at **INTERNSHALA.**

Head of the Department
ELECTRONICS AND COMMUNICATION **ENGINEERING**

Date:

Alok Kumar 21/07/23
External Examiner
Date:

INTERNSHALA TRAININGS

Certificate of Training

Shashikant Kumar

from Netaji subhas institute of technology Bihta has successfully completed a 6-week online training on **MATLAB** . The training consisted of Introduction, Foundations, Mathematics, Functions, Plotting, Programming, Case Studies, and Final Project modules.

In the final assessment, Shashikant scored 100% marks. We wish Shashikant all the best for future endeavours.

Sarvesh Agarwal

FOUNDER & CEO, INTERNSHALA

Date of certification: 2023-02-18

Certificate no. : c27tgbtqkm5

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ACKNOWLEDGEMENT

First I would like to thank my mentor at **Internshala Training** for giving me the opportunity to do an internship within the organization.

I would also like to thank all the people who have worked along with me at **Internshala** for helping me whenever I got stuck. They created an enjoyable working environment with their patience and openness.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am extremely grateful to my friends and my faculties who helped me in successful completion of this internship.

SHASHIKANT KUMAR
(20104103003)

ABSTRACT

The Organization:

Internshala is Having Worldwide Name in IT field. We provide High quality cyber security training and consulting services that full fill Students, government and corporate Requirements.

Internshala is Topmost online Educational Training Institute in India. and provides the careers opportunity also.

Internshala is an educational brand that provide the IT training for all B.Tech and BCA students and it also provides Ethical hacking course.Internshala Training is the best Online institute.

Mission:

To empower people and especially students by educating them on technology and related necessary skills.

To create local opportunity, growth and impact via tech education in every country around the world and to bring diversity in tech domain.

Goal:

1. To educate people in tech
2. Remove myths about tech education
3. Spread awareness about opportunities in tier two & tier three cities
4. Introduce tech content that is helpful for everyone

Contents

1 Tutorial lessons 1 . . . 1

1.1 Introduction . . . 1

1.2 Basic features . . . 2

1.3 A minimum MATLAB session . . . 2

1.3.1 Starting MATLAB . . . 2

1.3.2 Using MATLAB as a calculator . . . 4

1.3.3 Quitting MATLAB . . . 5

1.4 Getting started . . . 5

1.4.1 Creating MATLAB variables . . . 5

1.4.2 Overwriting variable . . . 6

1.4.3 Error messages . . . 6

1.4.4 Making corrections . . . 6

1.4.5 Controlling the hierarchy of operations or precedence . . . 6

1.4.6 Controlling the appearance of floating point number . . . 8

1.4.7 Managing the workspace . . . 8

1.4.8 Keeping track of your work session . . . 9

1.4.9 Entering multiple statements per line . . . 9

1.4.10 Miscellaneous commands . . . 10

1.4.11 Getting help . . . 10

1.5 Exercises . . . 11

2 Tutorial lessons 2 . . . 12

2.1 Mathematical functions . . . 12

2.1.1 Examples . . . 13

2.2 Basic plotting 14
2.2.1 overview 14
2.2.2 Creating simple plots 14
2.2.3 Adding titles, axis labels, and annotations 15
2.2.4 Multiple data sets in one plot 16
2.2.5 Specifying line styles and colors 17
2.3 Exercises 18
2.4 Introduction 19
2.5 Matrix generation 19
2.5.1 Entering a vector 19
2.5.2 Entering a matrix 20
2.5.3 Matrix indexing 21
2.5.4 Colon operator 22
2.5.5 Linear spacing 22
2.5.6 Colon operator in a matrix 22
2.5.7 Creating a sub-matrix 23
2.5.8 Deleting row or column 25
2.5.9 Dimension 25
2.5.10 Continuation 26
2.5.11 Transposing a matrix 26
2.5.12 Concatenating matrices 26
2.5.13 Matrix generators 27
2.5.14 Special matrices 28
2.6 Exercises 29

3 Array operations and Linear equations **30**
3.1 Array operations 30
3.1.1 Matrix arithmetic operations 30
3.1.2 Array arithmetic operations 30
3.2 Solving linear equations 32
3.2.1 Matrix inverse 33

3.2.2 Matrix functions 34
3.3 Exercises 34

4 Introduction to programming in MATLAB 35
4.1 Introduction 35
4.2 M-File Scripts 35
4.2.1 Examples 36
4.2.2 Script side-effects 37
4.3 M-File functions 38
4.3.1 Anatomy of a M-File function 38
4.3.2 Input and output arguments 40
4.4 Input to a script file 40
4.5 Output commands 41
4.6 Exercises 42

5 Control flow and operators 43
5.1 Introduction 43
5.2 Control flow 43
5.2.1 The ```if...end`'' structure 43
5.2.2 Relational and logical operators 45
5.2.3 The ```for...end`'' loop 45
5.2.4 The ```while...end`'' loop 46
5.2.5 Other flow structures 46
5.2.6 Operator precedence 47
5.3 Saving output to a file 47
5.4 Exercises 48

6 Debugging M-files 49
6.1 Introduction 49
6.2 Debugging process 49
6.2.1 Preparing for debugging 50
6.2.2 Setting breakpoints 50

6.2.3 Running with breakpoints 50
6.2.4 Examining values 51
6.2.5 Correcting and ending debugging 51
6.2.6 Ending debugging 51
6.2.7 Correcting an M-file 51

A Summary of commands **53**

B Release notes for Release 14 with Service Pack 2 **58**
B.1 Summary of changes 58
B.2 Other changes 60
B.3 Further details 60

C Main characteristics of MATLAB **62**
C.1 History 62
C.2 Strengths 62
C.3 Weaknesses 63
C.4 Competition 63

List of Tables

1.1 Basic arithmetic operators 5
1.2 Hierarchy of arithmetic operations 7

2.1 Elementary functions 12
2.2 Predefined constant values 13
2.3 Attributes for `plot` 18
2.4 Elementary matrices 27
2.5 Special matrices 28

3.1 Array operators 31
3.2 Summary of matrix and array operations 32
3.3 Matrix functions 34

4.1 Anatomy of a M-File function 38
4.2 Difference between scripts and functions 39
4.3 Example of input and output arguments 40
4.4 `disp` and `fprintf` commands 41

5.1 Relational and logical operators 45
5.2 Operator precedence 47

A.1 Arithmetic operators and special characters 53
A.2 Array operators 54
A.3 Relational and logical operators 54
A.4 Managing workspace and file commands 55
A.5 Predefined variables and math constants 55

A.6 Elementary matrices and arrays 56
A.7 Arrays and Matrices: Basic information 56
A.8 Arrays and Matrices: operations and manipulation 56
A.9 Arrays and Matrices: matrix analysis and linear equations 57

List of Figures

1.1 The graphical interface to the MATLAB workspace 3

2.1 Plot for the vectors x and y 15

2.2 Plot of the Sine function 16

2.3 Typical example of multiple plots 17

Preface

"Introduction to MATLAB for Engineering Students" is a document for an introductory course in MATLAB®[1] and technical computing. It is used for freshmen classes at Northwestern University. This document is not a comprehensive introduction or a reference manual. Instead, it focuses on the specific features of MATLAB that are useful for engineering classes. The lab sessions are used with one main goal: to allow students to become familiar with computer software (e.g., MATLAB) to solve application problems. We assume that the students have no prior experience with MATLAB.

The availability of technical computing environment such as MATLAB is now reshaping the role and applications of computer laboratory projects to involve students in more intense problem-solving experience. This availability also provides an opportunity to easily conduct numerical experiments and to tackle realistic and more complicated problems.

Originally, the manual is divided into computer laboratory sessions (labs). The lab document is designed to be used by the students while working at the computer. The emphasis here is "learning by doing". This quiz-like session is supposed to be fully completed in 50 minutes in class.

The seven lab sessions include not only the basic concepts of MATLAB, but also an introduction to scientific computing, in which they will be useful for the upcoming engineering courses. In addition, engineering students will see MATLAB in their other courses.

The end of this document contains two useful sections: a Glossary which contains the brief summary of the commands and built-in functions as well as a collection of release notes. The release notes, which include several new features of the Release 14 with Service Pack 2, well known as R14SP2, can also be found in Appendix. All of the MATLAB commands have been tested to take advantage with new features of the current version of MATLAB available here at Northwestern (R14SP2). Although, most of the examples and exercises still work with previous releases as well.

This manual reflects the ongoing effort of the McCormick School of Engineering and Applied Science leading by Dean Stephen Carr to institute a significant technical computing in the Engineering First®[2] courses taught at Northwestern University.

Finally, the students - Engineering Analysis (EA) Section - deserve my special gratitude. They were very active participants in class.

David Houcque
Evanston, Illinois
August 2005

[1] MATLAB® is a registered trademark of MathWorks, Inc.

[2] Engineering First® is a registered trademark of McCormick School of Engineering and Applied Science (Northwestern University)

Acknowledgements

I would like to thank Dean Stephen Carr for his constant support. I am grateful to a number of people who offered helpful advice and comments. I want to thank the EA1 instructors (Fall Quarter 2004), in particular Randy Freeman, Jorge Nocedal, and Allen Taflove for their helpful reviews on some specific parts of the document. I also want to thank Malcomb MacIver, EA3 Honors instructor (Spring 2005) for helping me to better understand the *animation* of system dynamics using MATLAB. I am particularly indebted to the many students (340 or so) who have used these materials, and have communicated their comments and suggestions. Finally, I want to thank IT personnel for helping setting up the classes and other computer related work: Rebecca Swierz, Jesse Becker, Rick Mazec, Alan Wolff, Ken Kalan, Mike Vilches, and Daniel Lee.

About the author

David Houcque has more than 25 years' experience in the modeling and simulation of structures and solid continua including 14 years in industry. In industry, he has been working as R&D engineer in the fields of nuclear engineering, oil rig platform offshore design, oil reservoir engineering, and steel industry. All of these include working in different international environments: Germany, France, Norway, and United Arab Emirates. Among other things, he has a combined background experience: scientific computing and engineering expertise. He earned his academic degrees from Europe and the United States.

Here at Northwestern University, he is working under the supervision of Professor Brian Moran, a world-renowned expert in fracture mechanics, to investigate the integrity assessment of the aging highway bridges under severe operating conditions and corrosion.

Chapter 1

Tutorial lessons 1

1.1 Introduction

The tutorials are independent of the rest of the document. The primarily objective is to help you learn *quickly* the first steps. The emphasis here is "learning by doing". Therefore, the best way to learn is by trying it yourself. Working through the examples will give you a feel for the way that MATLAB operates. In this introduction we will describe how MATLAB handles simple numerical expressions and mathematical formulas.

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB [1] is a high-performance language for technical computing. It integrates *computation*, *visualization*, and *programming* environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated *data structures*, contains built-in editing and *debugging tools*, and supports *object-oriented programming*. These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an *array* that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful *built-in* routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as *toolbox*. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

In addition to the MATLAB documentation which is mostly available on-line, we would

recommend the following books: [2], [3], [4], [5], [6], [7], [8], and [9]. They are excellent in their specific applications.

1.2 Basic features

As we mentioned earlier, the following TUTORIAL lessons are designed to get you started quickly in MATLAB. The lessons are intended to make you familiar with the basics of MATLAB. We urge you to complete the EXERCISES given at the end of each lesson.

1.3 A minimum MATLAB session

The goal of this *minimum* session (also called *starting* and *exiting* sessions) is to learn the first steps:

- How to log on
- Invoke MATLAB
- Do a few simple calculations
- How to quit MATLAB

1.3.1 Starting MATLAB

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut *icon* (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains *other* windows. The major tools within or accessible from the desktop are:

- The COMMAND WINDOW
- The COMMAND HISTORY
- The WORKSPACE
- The CURRENT DIRECTORY
- The HELP BROWSER
- The START button

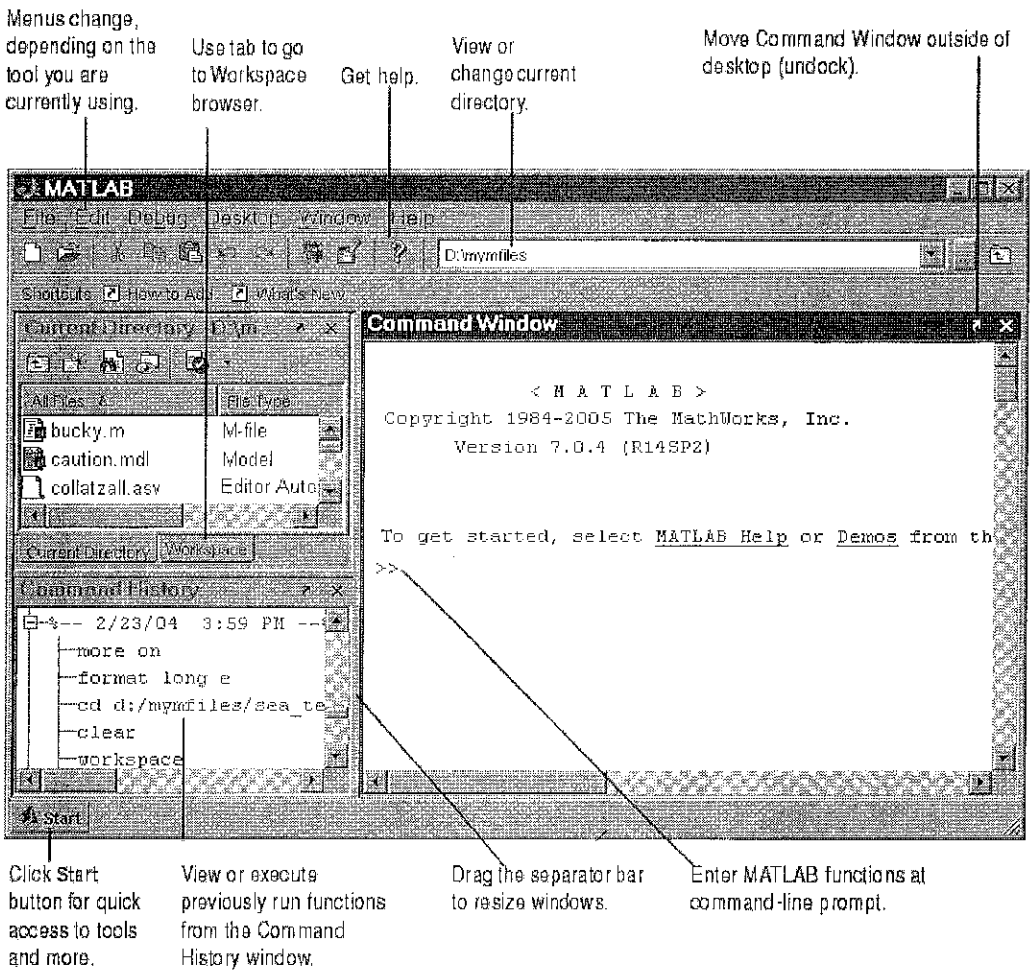


Figure 1.1: The graphical interface to the MATLAB workspace

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure 1.1. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs.

Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run.

You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

```
>>          for full version
EDU>        for educational version
```

NOTE: To simplify the notation, we will use this prompt, >>, as a standard prompt sign, though our MATLAB version is for educational purpose.

1.3.2 Using MATLAB as a calculator

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate the expression, $1 + 2 \times 3$. You type it at the prompt command (>>) as follows,

```
>> 1+2*3
ans =
    7
```

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable ans, short for answer, to store the results of the current calculation. Note that the variable ans is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example,

```
>> x = 1+2*3
x  =
   7
```

will result in x being given the value $1 + 2 \times 3 = 7$. This variable name can always be used to refer to the results of the previous computations. Therefore, computing $4x$ will result in

```
>> 4*x
ans  =
     28.0000
```

Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic operators.

Table 1.1: Basic arithmetic operators

SYMBOL	OPERATION	EXAMPLE
$+$	Addition	$2+3$
$-$	Subtraction	$2-3$
$*$	Multiplication	$2*3$
$/$	Division	$2/3$

1.3.3 Quitting MATLAB

To end your MATLAB session, type **quit** in the Command Window, or select **File** $\longrightarrow$ **Exit MATLAB** in the desktop main menu.

1.4 Getting started

After learning the minimum MATLAB session, we will now learn to use some additional operations.

1.4.1 Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable assignment is

```
variable name = a value (or an expression)
```

For example,

```
>> x = expression
```

where `expression` is a combination of numerical values, mathematical operators, variables, and function calls. On other words, `expression` can involve:

- manual entry
- built-in functions
- user-defined functions

1.4.2 Overwriting variable

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line. Then the sequence of commands looks like this:

```
>> t = 5;
>> t = t+1
t    =
     6
```

1.4.3 Error messages

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, $*$, in the following expression

```
>> x = 10;
>> 5x
??? 5x
     |
Error: Unexpected MATLAB expression.
```

1.4.4 Making corrections

To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key $\uparrow$. When the command is displayed at the command prompt, it can be modified if needed and executed.

1.4.5 Controlling the hierarchy of operations or precedence

Let's consider the previous arithmetic operation, but now we will include *parentheses*. For example, $1 + 2 \times 3$ will become $(1 + 2) \times 3$

```
>> (1+2)*3
ans    =
       9
```

and, from previous example

```
>> 1+2*3
ans    =
       7
```

By adding parentheses, these two expressions give different results: 9 and 7.

The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. *Exponentiations* are done *first*, followed by *multiplications* and *divisions*, and finally by *additions* and *subtractions*. However, the standard order of precedence of arithmetic operations can be changed by inserting *parentheses*. For example, the result of $1+2\times3$ is quite different than the similar expression with parentheses $(1+2)\times3$. The results are 7 and 9 respectively. Parentheses can always be used to overrule *priority*, and their use is recommended in some complex expressions to avoid ambiguity.

Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules. The order in which the arithmetic operations are evaluated is given in Table 1.2. MATLAB arithmetic operators obey the same *precedence* rules as those in

Table 1.2: Hierarchy of arithmetic operations

PRECEDENCE	MATHEMATICAL OPERATIONS
First	The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward.
Second	All exponentials are evaluated, working from left to right
Third	All multiplications and divisions are evaluated, working from left to right
Fourth	All additions and subtractions are evaluated, starting from left to right

most computer programs. For operators of *equal* precedence, evaluation is from *left* to *right*. Now, consider another example:

$$\frac{1}{2+3^2}+\frac{4}{5}\times\frac{6}{7}$$

In MATLAB, it becomes

```
>> 1/(2+3^2)+4/5*6/7
ans  =
     0.7766
```

or, if parentheses are missing,

```
>> 1/2+3^2+4/5*6/7
ans  =
     10.1857
```

So here what we get: two different results. Therefore, we want to emphasize the importance of precedence rule in order to avoid ambiguity.

1.4.6 Controlling the appearance of floating point number

MATLAB by default displays only 4 decimals in the result of the calculations, for example -163.6667, as shown in above examples. However, MATLAB does numerical calculations in *double* precision, which is 15 digits. The command `format` controls how the results of computations are displayed. Here are some examples of the different formats together with the resulting outputs.

```
>> format short
>> x=-163.6667
```

If we want to see all 15 digits, we use the command `format long`

```
>> format long
>> x= -1.636666666666667e+002
```

To return to the standard format, enter `format short`, or simply `format`.

There are several other formats. For more details, see the MATLAB documentation, or type `help format`.

NOTE - Up to now, we have let MATLAB repeat everything that we enter at the prompt (>>). Sometimes this is not quite useful, in particular when the output is pages en length. To prevent MATLAB from echoing what we type, simply enter a semicolon (`;`) at the end of the command. For example,

```
>> x=-163.6667;
```

and then ask about the value of `x` by typing,

```
>> x
x   =
    -163.6667
```

1.4.7 Managing the workspace

The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a `clear` command at the start of each new independent calculation.

```
>> clear
```

The command `clear` or `clear all` removes all variables from the workspace. This frees up system memory. In order to display a list of the variables currently in the memory, type

```
>> who
```

while, `whos` will give more details which include size, space allocation, and class of the variables.

1.4.8 Keeping track of your work session

It is possible to keep track of everything done during a MATLAB session with the `diary` command.

```
>> diary
```

or give a name to a created file,

```
>> diary FileName
```

where `FileName` could be any arbitrary name you choose.

The function `diary` is useful if you want to save a complete MATLAB session. They save all input and output as they appear in the MATLAB window. When you want to stop the recording, enter `diary off`. If you want to start recording again, enter `diary on`. The file that is created is a simple text file. It can be opened by an editor or a word processing program and edited to remove extraneous material, or to add your comments. You can use the function type to view the diary file or you can edit in a text editor or print. This command is useful, for example in the process of preparing a homework or lab submission.

1.4.9 Entering multiple statements per line

It is possible to enter multiple statements per line. Use commas (`,`) or semicolons (`;`) to enter more than one statement at once. Commas (`,`) allow multiple statements per line without suppressing output.

```
>> a=7; b=cos(a), c=cosh(a)
b    =
     0.6570
c    =
     548.3170
```

1.4.10 Miscellaneous commands

Here are few additional useful commands:

- To clear the Command Window, type `clc`
- To abort a MATLAB computation, type `ctrl-c`
- To continue a line, type ...

1.4.11 Getting help

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the *Help Browser*. The Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

```
>> help Command
```

Another way to get help is to use the `lookfor` command. The `lookfor` command differs from the `help` command. The help command searches for an exact function name match, while the `lookfor` command searches the quick summary information in each function for a match. For example, suppose that we were looking for a function to take *the inverse of a matrix*. Since MATLAB does not have a function named `inverse`, the command `help inverse` will produce nothing. On the other hand, the command `lookfor inverse` will produce detailed information, which includes the function of interest, inv.

```
>> lookfor inverse
```

NOTE - At this particular time of our study, it is important to emphasize one main point. Because MATLAB is a huge program; it is impossible to cover *all the details* of each function one by one. However, we will give you information how to get help. Here are some examples:

- Use on-line `help` to request info on a specific function

  ```
  >> help sqrt
  ```

- In the current version (MATLAB version 7), the `doc` function opens the on-line version of the help manual. This is very helpful for more complex commands.

  ```
  >> doc plot
  ```

- Use `lookfor` to find functions by keywords. The general form is

```
>> lookfor FunctionName
```

1.5 Exercises

NOTE: Due to the teaching class during this Fall 2005, the *problems* are *temporarily* removed from this section.

Chapter 2

Tutorial lessons 2

2.1 Mathematical functions

MATLAB offers many predefined mathematical functions for technical computing which contains a large set of mathematical functions.

Typing `help elfun` and `help specfun` calls up full lists of *elementary* and *special* functions respectively.

There is a long list of mathematical functions that are *built* into MATLAB. These functions are called *built-ins*. Many standard mathematical functions, such as $\sin(x)$, $\cos(x)$, $\tan(x)$, e^x, $ln(x)$, are evaluated by the functions `sin`, `cos`, `tan`, `exp`, and `log` respectively in MATLAB.

Table 2.1 lists some commonly used functions, where variables `x` and `y` can be numbers, vectors, or matrices.

Table 2.1: Elementary functions

`cos(x)`	Cosine	`abs(x)`	Absolute value
`sin(x)`	Sine	`sign(x)`	Signum function
`tan(x)`	Tangent	`max(x)`	Maximum value
`acos(x)`	Arc cosine	`min(x)`	Minimum value
`asin(x)`	Arc sine	`ceil(x)`	Round towards $+\infty$
`atan(x)`	Arc tangent	`floor(x)`	Round towards $-\infty$
`exp(x)`	Exponential	`round(x)`	Round to nearest integer
`sqrt(x)`	Square root	`rem(x)`	Remainder after division
`log(x)`	Natural logarithm	`angle(x)`	Phase angle
`log10(x)`	Common logarithm	`conj(x)`	Complex conjugate

In addition to the elementary functions, MATLAB includes a number of predefined

constant values. A list of the most common values is given in Table 2.2.

Table 2.2: Predefined constant values

`pi`	The π number, $\pi = 3.14159\ldots$
`i,j`	The imaginary unit i, $\sqrt{-1}$
`Inf`	The infinity, ∞
`NaN`	Not a number

2.1.1 Examples

We illustrate here some typical examples which related to the elementary functions previously defined.

As a first example, the value of the expression $y = e^{-a}\sin(x) + 10\sqrt{y}$, for $a = 5$, $x = 2$, and $y = 8$ is computed by

```
>> a = 5; x = 2; y = 8;
>> y = exp(-a)*sin(x)+10*sqrt(y)
y  =
   28.2904
```

The subsequent examples are

```
>> log(142)
ans    =
       4.9558

>> log10(142)
ans    =
       2.1523
```

Note the difference between the natural logarithm `log(x)` and the decimal logarithm (base 10) `log10(x)`.

To calculate $\sin(\pi/4)$ and e^{10}, we enter the following commands in MATLAB,

```
>> sin(pi/4)
ans    =
       0.7071

>> exp(10)
ans    =
       2.2026e+004
```

NOTES:

- Only use built-in functions on the right hand side of an expression. Reassigning the value to a built-in function can create problems.
- There are some exceptions. For example, `i` and `j` are pre-assigned to $\sqrt{-1}$. However, one or both of `i` or `j` are often used as loop indices.
- To avoid any possible confusion, it is suggested to use instead `ii` or `jj` as loop indices.

2.2 Basic plotting

2.2.1 overview

MATLAB has an excellent set of graphic tools. Plotting a given data set or the results of computation is possible with very few commands. You are highly encouraged to plot mathematical functions and results of analysis as often as possible. Trying to understand mathematical equations with graphics is an enjoyable and very efficient way of learning mathematics. Being able to plot mathematical functions and data freely is the most important step, and this section is written to assist you to do just that.

2.2.2 Creating simple plots

The basic MATLAB graphing procedure, for example in 2D, is to take a vector of x-coordinates, $\mathtt{x} = (x_1, \ldots, x_N)$, and a vector of y-coordinates, $\mathtt{y} = (y_1, \ldots, y_N)$, locate the points (x_i, y_i), with $i = 1, 2, \ldots, n$ and then join them by straight lines. You need to prepare x and y in an identical array form; namely, x and y are both row arrays or column arrays of the *same* length.

The MATLAB command to plot a graph is `plot(x,y)`. The vectors $\mathtt{x} = (1, 2, 3, 4, 5, 6)$ and $\mathtt{y} = (3, -1, 2, 4, 5, 1)$ produce the picture shown in Figure 2.1.

```
>> x = [1 2 3 4 5 6];
>> y = [3 -1 2 4 5 1];
>> plot(x,y)
```

NOTE: The `plot` functions has different forms depending on the input arguments. If `y` is a vector `plot(y)` produces a piecewise linear graph of the elements of `y` versus the index of the elements of `y`. If we specify two vectors, as mentioned above, `plot(x,y)` produces a graph of `y` versus `x`.

For example, to plot the function $\sin(x)$ on the interval $[0, 2\pi]$, we first create a vector of x values ranging from 0 to 2π, then compute the *sine* of these values, and finally plot the result:

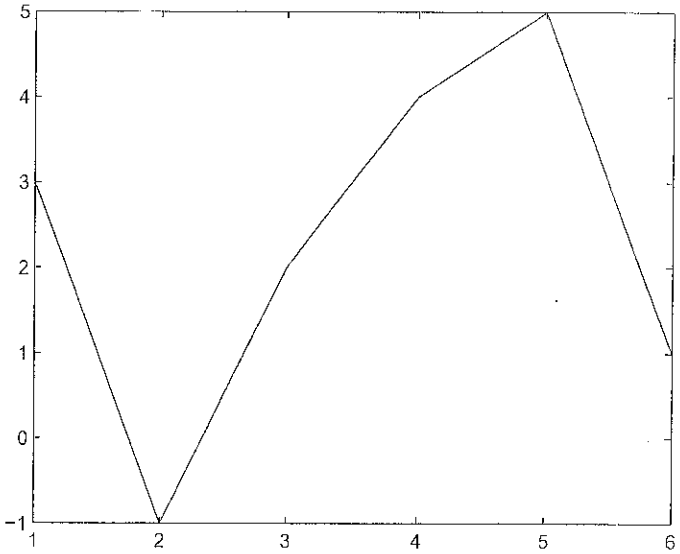


Figure 2.1: Plot for the vectors x and y

```
>> x = 0:pi/100:2*pi;
>> y = sin(x);
>> plot(x,y)
```

NOTES:

- `0:pi/100:2*pi` yields a vector that
  - starts at 0,
  - takes steps (or increments) of $\pi/100$,
  - stops when 2π is reached.
- If you omit the increment, MATLAB automatically increments by 1.

2.2.3 Adding titles, axis labels, and annotations

MATLAB enables you to add axis labels and titles. For example, using the graph from the previous example, add an x- and y-axis labels.

Now label the axes and add a title. The character `\pi` creates the symbol π. An example of 2D plot is shown in Figure 2.2.

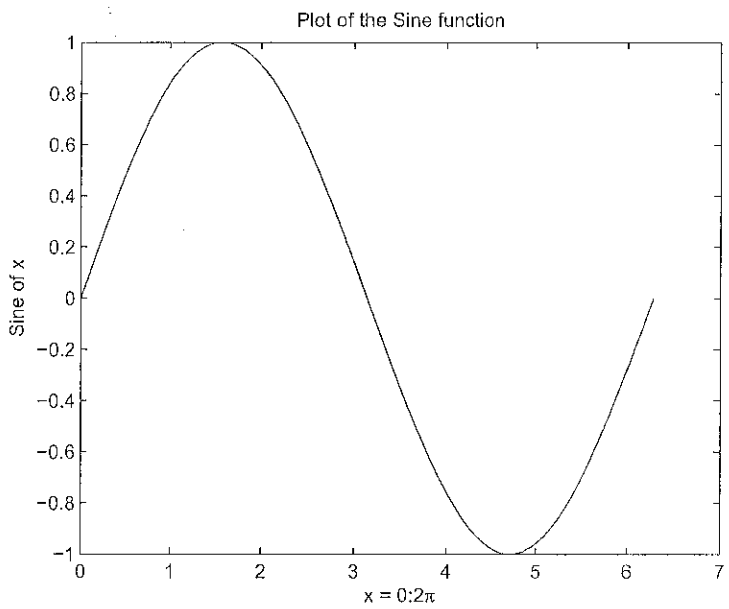


Figure 2.2: Plot of the Sine function

```
>> xlabel('x = 0:2\pi')
>> ylabel('Sine of x')
>> title('Plot of the Sine function')
```

The color of a single curve is, by default, blue, but other colors are possible. The desired color is indicated by a third argument. For example, red is selected by `plot(x,y,'r')`. Note the single quotes, ' ', around `r`.

2.2.4 Multiple data sets in one plot

Multiple (x, y) *pairs* arguments create *multiple* graphs with a single call to *plot*. For example, these statements plot three related functions of x: $y_1 = 2\cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0.5 * \cos(x)$, in the interval $0 \leq x \leq 2\pi$.

```
>> x = 0:pi/100:2*pi;
>> y1 = 2*cos(x);
>> y2 = cos(x);
>> y3 = 0.5*cos(x);
>> plot(x,y1,'--',x,y2,'-',x,y3,':')
>> xlabel('0 \leq x \leq 2\pi')
>> ylabel('Cosine functions')
>> legend('2*cos(x)','cos(x)','0.5*cos(x)')
```

```
>> title('Typical example of multiple plots')
>> axis([0 2*pi -3 3])
```

The result of multiple data sets in one graph plot is shown in Figure 2.3.

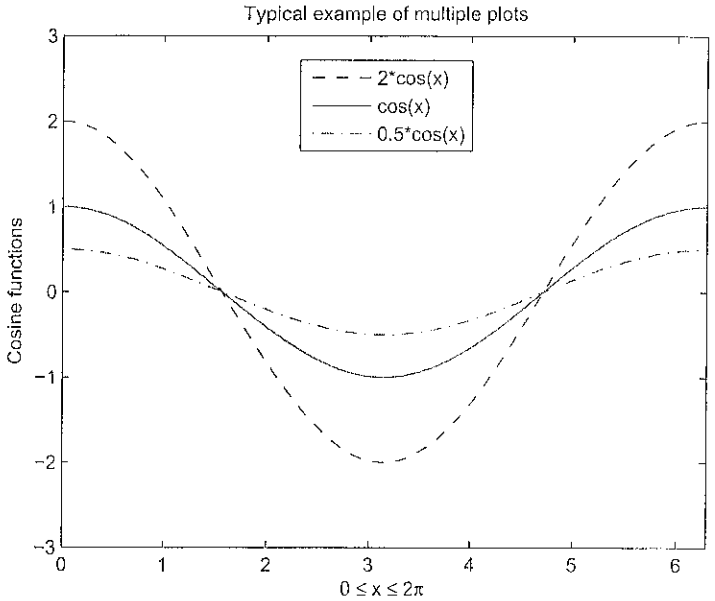


Figure 2.3: Typical example of multiple plots

By default, MATLAB uses *line style* and *color* to distinguish the data sets plotted in the graph. However, you can change the appearance of these graphic components or add annotations to the graph to help explain your data for presentation.

2.2.5 Specifying line styles and colors

It is possible to specify *line styles*, *colors*, and *markers* (e.g., circles, plus signs, ...) using the `plot` command:

```
plot(x,y,'style_color_marker')
```

where `style_color_marker` is a *triplet* of values from Table 2.3.

To find additional information, type `help plot` or `doc plot`.

Table 2.3: Attributes for `plot`

SYMBOL	COLOR	SYMBOL	LINE STYLE	SYMBOL	MARKER
k	Black	$-$	Solid	$+$	Plus sign
r	Red	$--$	Dashed	o	Circle
b	Blue	:	Dotted	$*$	Asterisk
g	Green	$-.$	Dash-dot	.	Point
c	Cyan	none	No line	$\times$	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

2.3 Exercises

NOTE: Due to the teaching class during this Fall Quarter 2005, the *problems* are *temporarily* removed from this section.

2.4 Introduction

Matrices are the basic elements of the MATLAB environment. A matrix is a two-dimensional array consisting of m rows and n columns. Special cases are *column vectors* ($n = 1$) and *row vectors* ($m = 1$).

In this section we will illustrate how to apply different *operations* on matrices. The following topics are discussed: vectors and matrices in MATLAB, the inverse of a matrix, determinants, and matrix manipulation.

MATLAB supports two types of operations, known as *matrix operations* and *array operations*. Matrix operations will be discussed first.

2.5 Matrix generation

Matrices are fundamental to MATLAB. Therefore, we need to become familiar with matrix generation and manipulation. Matrices can be generated in several ways.

2.5.1 Entering a vector

A vector is a special case of a matrix. The purpose of this section is to show how to create vectors and matrices in MATLAB. As discussed earlier, an array of dimension $1 \times n$ is called a *row* vector, whereas an array of dimension $m \times 1$ is called a *column* vector. The elements of vectors in MATLAB are enclosed by square brackets and are separated by spaces or by commas. For example, to enter a row vector, v, type

```
>> v = [1 4 7 10 13]
v  =
     1     4     7     10     13
```

Column vectors are created in a similar way, however, semicolon (;) must separate the components of a column vector,

```
>> w = [1;4;7;10;13]
w  =
     1
     4
     7
     10
     13
```

On the other hand, a *row* vector is converted to a *column* vector using the *transpose* operator. The *transpose* operation is denoted by an apostrophe or a single quote (').

```
>> w = v'
w  =
     1
     4
     7
     10
     13
```

Thus, `v(1)` is the first element of vector `v`, `v(2)` its second element, and so forth.

Furthermore, to access *blocks* of elements, we use MATLAB's colon notation (`:`). For example, to access the first three elements of `v`, we write,

```
>> v(1:3)
ans =
    1    4    7
```

Or, all elements from the third through the last elements,

```
>> v(3,end)
ans =
    7    10    13
```

where `end` signifies the *last* element in the vector. If `v` is a vector, writing

```
>> v(:)
```

produces a column vector, whereas writing

```
>> v(1:end)
```

produces a row vector.

2.5.2 Entering a matrix

A matrix is an array of numbers. To type a matrix into MATLAB you must

- begin with a square bracket, [
- separate elements in a row with spaces or commas (,)
- use a semicolon (;) to separate rows
- end the matrix with another square bracket,].

Here is a typical example. To enter a matrix A, such as,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{2.1}$$

type,

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

MATLAB then displays the 3×3 matrix as follows,

```
A   =
    1   2   3
    4   5   6
    7   8   9
```

Note that the use of semicolons (;) here is different from their use mentioned earlier to suppress output or to write multiple commands in a single line.

Once we have entered the matrix, it is automatically stored and remembered in the *Workspace*. We can refer to it simply as matrix A. We can then view a particular element in a matrix by specifying its location. We write,

```
>> A(2,1)
ans  =
     4
```

A(2,1) is an element located in the second row and first column. Its value is 4.

2.5.3 Matrix indexing

We select elements in a matrix just as we did for vectors, but now we need two indices. The element of row i and column j of the matrix A is denoted by A(i,j). Thus, A(i,j) in MATLAB refers to the element A_{ij} of matrix A. The *first* index is the *row* number and the *second* index is the *column* number. For example, A(1,3) is an element of *first* row and *third* column. Here, A(1,3)=3.

Correcting any entry is easy through indexing. Here we substitute A(3,3)=9 by A(3,3)=0. The result is

```
>> A(3,3) = 0
A   =
    1    2    3
    4    5    6
    7    8    0
```

Single elements of a matrix are accessed as `A(i,j)`, where $i \geq 1$ and $j \geq 1$. Zero or negative subscripts are not supported in MATLAB.

2.5.4 Colon operator

The colon operator will prove very useful and understanding how it works is the key to efficient and convenient usage of MATLAB. It occurs in several different forms.

Often we must deal with matrices or vectors that are too large to enter one element at a time. For example, suppose we want to enter a vector x consisting of points $(0, 0.1, 0.2, 0.3, \cdots, 5)$. We can use the command

```
>> x = 0:0.1:5;
```

The row vector has 51 elements.

2.5.5 Linear spacing

On the other hand, there is a command to generate linearly spaced vectors: `linspace`. It is similar to the colon operator (`:`), but gives direct control over the number of points. For example,

```
y = linspace(a,b)
```

generates a row vector `y` of 100 points linearly spaced between and including `a` and `b`.

```
y = linspace(a,b,n)
```

generates a row vector `y` of `n` points linearly spaced between and including `a` and `b`. This is useful when we want to divide an interval into a number of subintervals of the same length. For example,

```
>> theta = linspace(0,2*pi,101)
```

divides the interval $[0, 2\pi]$ into 100 equal subintervals, then creating a vector of 101 elements.

2.5.6 Colon operator in a matrix

The colon operator can also be used to pick out a certain row or column. For example, the statement `A(m:n,k:l` specifies rows m to n and column k to l. Subscript expressions refer to portions of a matrix. For example,

```
>> A(2,:)
ans  =
     4    5    6
```

is the second row elements of A.

The colon operator can also be used to extract a sub-matrix from a matrix A.

```
>> A(:,2:3)
ans  =
     2    3
     5    6
     8    0
```

A(:,2:3) is a sub-matrix with the last two columns of A.

A row or a column of a matrix can be deleted by setting it to a *null* vector, [ ].

```
>> A(:,2)=[]
ans  =
     1    3
     4    6
     7    0
```

2.5.7 Creating a sub-matrix

To extract a *submatrix* B consisting of rows 2 and 3 and columns 1 and 2 of the matrix A, do the following

```
>> B = A([2 3],[1 2])
B    =
     4     5
     7     8
```

To interchange rows 1 and 2 of A, use the vector of row indices together with the colon operator.

```
>> C = A([2 1 3],:)
C    =
     4     5     6
     1     2     3
     7     8     0
```

It is important to note that the *colon operator* (:) stands for *all columns* or *all rows*. To create a vector version of matrix A, do the following

```
>> A(:)
ans =
     1
     2
     3
     4
     5
     6
     7
     8
     0
```

The submatrix comprising the intersection of rows `p` to `q` and columns `r` to `s` is denoted by `A(p:q,r:s)`.

As a special case, a colon `(:)` as the row or column specifier covers all entries in that row or column: thus

- `A(:,j)` is the `j`th column of `A`, while
- `A(i,:)` is the `i`th row, and
- `A(end,:)` picks out the last row of `A`.

The keyword `end`, used in `A(end,:)`, denotes the last index in the specified dimension. Here are some examples.

```
>> A
A  =
   1    2    3
   4    5    6
   7    8    9

>> A(2:3,2:3)
ans =
     5    6
     8    9

>> A(end:-1:1,end)
ans =
     9
     6
     3
```

```
>> A([1 3],[2 3])
ans =
    2    3
    8    9
```

2.5.8 Deleting row or column

To delete a row or column of a matrix, use the *empty vector* operator, [ ].

```
>> A(3,:) = []
A  =
   1    2    3
   4    5    6
```

Third row of matrix A is now deleted. To restore the third row, we use a technique for creating a matrix

```
>> A = [A(1,:);A(2,:);[7 8 0]]
A  =
   1    2    3
   4    5    6
   7    8    0
```

Matrix A is now restored to its original form.

2.5.9 Dimension

To determine the *dimensions* of a matrix or vector, use the command `size`. For example,

```
>> size(A)
ans    =
       3     3
```

means 3 rows and 3 columns.

Or more explicitly with,

```
>> [m,n]=size(A)
```

2.5.10 Continuation

If it is not possible to type the entire input on the same line, use consecutive periods, called an ellipsis ..., to signal continuation, then continue the input on the next line.

```
B = [4/5        7.23*tan(x)        sqrt(6); ...
     1/x^2      0                  3/(x*log(x)); ...
     x-7        sqrt(3)            x*sin(x)];
```

Note that *blank* spaces around +, −, = signs are optional, but they improve readability.

2.5.11 Transposing a matrix

The *transpose* operation is denoted by an apostrophe or a single quote ('). It flips a matrix about its main diagonal and it turns a row vector into a column vector. Thus,

```
>> A'
ans =
    1   4   7
    2   5   8
    3   6   0
```

By using linear algebra notation, the transpose of $m \times n$ real matrix A is the $n \times m$ matrix that results from interchanging the rows and columns of A. The transpose matrix is denoted A^T.

2.5.12 Concatenating matrices

Matrices can be made up of sub-matrices. Here is an example. First, let's recall our previous matrix A.

```
A   =
    1   2   3
    4   5   6
    7   8   9
```

The new matrix B will be,

```
>> B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]
B   =
    1    2    3    10    20    30
```

```
 4    5    6   40   50   60
 7    8    9   70   80   90
-1   -2   -3    1    0    0
-4   -5   -6    0    1    0
-7   -8   -9    0    0    1
```

2.5.13 Matrix generators

MATLAB provides functions that generates elementary matrices. The matrix of zeros, the matrix of ones, and the identity matrix are returned by the functions `zeros`, `ones`, and `eye`, respectively.

Table 2.4: Elementary matrices

`eye(m,n)`	Returns an m-by-n matrix with 1 on the main diagonal
`eye(n)`	Returns an n-by-n square identity matrix
`zeros(m,n)`	Returns an m-by-n matrix of zeros
`ones(m,n)`	Returns an m-by-n matrix of ones
`diag(A)`	Extracts the diagonal of matrix A
`rand(m,n)`	Returns an m-by-n matrix of random numbers

For a complete list of *elementary matrices* and *matrix manipulations*, type `help elmat` or `doc elmat`. Here are some examples:

1.
```
>> b=ones(3,1)
b    =
     1
     1
     1
```

Equivalently, we can define `b` as `>> b=[1;1;1]`

2.
```
>> eye(3)
ans    =
     1     0     0
     0     1     0
     0     0     1
```

3.
```
>> c=zeros(2,3)
c    =
     0     0     0
```

```
     0     0     0
```

In addition, it is important to remember that the three elementary operations of *addition* (+), *subtraction* (−), and *multiplication* (∗) apply also to matrices whenever the dimensions are *compatible*.

Two other important matrix generation functions are `rand` and `randn`, which generate matrices of (pseudo-)random numbers using the same syntax as `eye`.

In addition, matrices can be constructed in a block form. With `C` defined by `C = [1 2; 3 4]`, we may create a matrix `D` as follows

```
>> D = [C zeros(2); ones(2) eye(2)]
D =
  1     2     0     0
  3     4     0     0
  1     1     1     0
  1     1     0     1
```

2.5.14 Special matrices

MATLAB provides a number of special matrices (see Table 2.5). These matrices have interesting properties that make them useful for constructing examples and for testing algorithms. For more information, see MATLAB documentation.

Table 2.5: Special matrices

`hilb`	Hilbert matrix
`invhilb`	Inverse Hilbert matrix
`magic`	Magic square
`pascal`	Pascal matrix
`toeplitz`	Toeplitz matrix
`vander`	Vandermonde matrix
`wilkinson`	Wilkinson's eigenvalue test matrix

2.6 Exercises

NOTE: Due to the teaching class during this Fall Quarter 2005, the *problems* are *temporarily* removed from this section.

Chapter 3

Array operations and Linear equations

3.1 Array operations

MATLAB has two different types of arithmetic operations: matrix arithmetic operations and array arithmetic operations. We have seen matrix arithmetic operations in the previous lab. Now, we are interested in array operations.

3.1.1 Matrix arithmetic operations

As we mentioned earlier, MATLAB allows arithmetic operations: $+$, $-$, $*$, and ^ to be carried out on matrices. Thus,

`A+B` or `B+A`	is valid if `A` and `B` are of the same size
`A*B`	is valid if `A`'s number of column equals `B`'s number of rows
`A^2`	is valid if `A` is square and equals `A*A`
α`*A` or `A*`α	multiplies each element of `A` by α

3.1.2 Array arithmetic operations

On the other hand, array arithmetic operations or *array operations* for short, are done *element-by-element*. The period character, ., distinguishes the array operations from the matrix operations. However, since the matrix and array operations are the same for addition (+) and subtraction (−), the character pairs (.+) and (.−) are not used. The list of array operators is shown below in Table 3.2. If `A` and `B` are two matrices of the same size with elements $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$, then the command

.*	Element-by-element multiplication
./	Element-by-element division
.^	Element-by-element exponentiation

Table 3.1: Array operators

```
>> C = A.*B
```

produces another matrix C of the same size with elements $c_{ij} = a_{ij}b_{ij}$. For example, using the same 3×3 matrices,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

we have,

```
>> C = A.*B
C   =
     10     40     90
    160    250    360
    490    640    810
```

To raise a scalar to a power, we use for example the command 10^2. If we want the operation to be applied to each element of a matrix, we use .^2. For example, if we want to produce a new matrix whose elements are the square of the elements of the matrix **A**, we enter

```
>> A.^2
ans   =
      1      4      9
     16     25     36
     49     64     81
```

The relations below summarize the above operations. To simplify, let's consider two vectors U and V with elements $U = [u_i]$ and $V = [v_j]$.

$$\begin{array}{lll} U.*V & \text{produces} & [u_1v_1 \; u_2v_2 \ldots \; u_nv_n] \\ U./V & \text{produces} & [u_1/v_1 \; u_2/v_2 \ldots \; u_n/v_n] \\ U.\hat{}V & \text{produces} & [u_1^{v_1} \; u_2^{v_2} \ldots \; u_n^{v_n}] \end{array}$$

Operation	Matrix	Array
Addition	+	+
Subtraction	−	−
Multiplication	*	.*
Division	/	./
Left division	\	.\
Exponentiation	^	.^

Table 3.2: Summary of matrix and array operations

3.2 Solving linear equations

One of the problems encountered most frequently in scientific computation is the solution of systems of simultaneous linear equations. With matrix notation, a system of simultaneous linear equations is written

$$Ax = b \tag{3.1}$$

where there are as many equations as unknown. A is a given square matrix of order n, b is a given column vector of n components, and x is an unknown column vector of n components.

In linear algebra we learn that the solution to $Ax = b$ can be written as $x = A^{-1}b$, where A^{-1} is the inverse of A.

For example, consider the following system of linear equations

$$\begin{cases} x + 2y + 3z &= 1 \\ 4x + 5y + 6z &= 1 \\ 7x + 8y &= 1 \end{cases}$$

The coefficient matrix A is

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{and the vector} \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

With matrix notation, a system of simultaneous linear equations is written

$$Ax = b \tag{3.2}$$

This equation can be solved for x using linear algebra. The result is $x = A^{-1}b$.

There are typically two ways to solve for x in MATLAB:

1. The first one is to use the matrix inverse, `inv`.

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = inv(A)*b
x   =
    -1.0000
     1.0000
    -0.0000
```

2. The second one is to use the *backslash* (\)operator. The numerical algorithm behind this operator is computationally efficient. This is a numerically reliable way of solving system of linear equations by using a well-known process of Gaussian elimination.

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = A\b
x   =
    -1.0000
     1.0000
    -0.0000
```

This problem is at the heart of many problems in scientific computation. Hence it is important that we know how to solve this type of problem efficiently.

Now, we know how to solve a system of linear equations. In addition to this, we will see some additional details which relate to this particular topic.

3.2.1 Matrix inverse

Let's consider the same matrix A.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

Calculating the inverse of A manually is probably not a pleasant work. Here the hand-calculation of A^{-1} gives as a final result:

$$A^{-1} = \frac{1}{9} \begin{bmatrix} -16 & 8 & -1 \\ 14 & -7 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

In MATLAB, however, it becomes as simple as the following commands:

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> inv(A)
ans =
    -1.7778    0.8889   -0.1111
     1.5556   -0.7778    0.2222
    -0.1111    0.2222   -0.1111
```

which is similar to:

$$A^{-1} = \frac{1}{9}\begin{bmatrix} -16 & 8 & -1 \\ 14 & -7 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

and the determinant of A is

```
>> det(A)
ans  =
     27
```

For further details on applied numerical linear algebra, see [10] and [11].

3.2.2 Matrix functions

MATLAB provides many matrix functions for various matrix/vector manipulations; see Table 3.3 for some of these functions. Use the online help of MATLAB to find how to use these functions.

`det`	Determinant
`diag`	Diagonal matrices and diagonals of a matrix
`eig`	Eigenvalues and eigenvectors
`inv`	Matrix inverse
`norm`	Matrix and vector norms
`rank`	Number of linearly independent rows or columns

Table 3.3: Matrix functions

3.3 Exercises

NOTE: Due to the teaching class during this Fall Quarter 2005, the *problems* are *temporarily* removed from this section.

Chapter 4

Introduction to programming in MATLAB

4.1 Introduction

So far in these lab sessions, all the commands were executed in the Command Window. The problem is that the commands entered in the Command Window cannot be saved and executed again for several times. Therefore, a different way of executing repeatedly commands with MATLAB is:

1. to *create* a file with a list of commands,
2. *save* the file, and
3. *run* the file.

If needed, corrections or changes can be made to the commands in the file. The files that are used for this purpose are called script files or *scripts* for short.

This section covers the following topics:

- M-File Scripts
- M-File Functions

4.2 M-File Scripts

A *script file* is an external file that contains a sequence of MATLAB statements. Script files have a filename extension `.m` and are often called M-files. M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that can accept arguments and can produce one or more outputs.

2.1 Examples

Here are two simple scripts.

Example 1

Consider the system of equations:

$$\begin{cases} x + 2y + 3z &= 1 \\ 3x + 3y + 4z &= 1 \\ 2x + 3y + 3z &= 2 \end{cases}$$

Find the solution x to the system of equations.

SOLUTION:

- Use the MATLAB *editor* to create a file: **File** → **New** → **M-file**.
- Enter the following statements in the file:

```
A = [1 2 3; 3 3 4; 2 3 3];
b = [1; 1; 2];
x = A\b
```

- Save the file, for example, `example1.m`.
- Run the file, in the command line, by typing:

```
>> example1
x   =
    -0.5000
     1.5000
    -0.5000
```

When execution completes, the variables (`A`, `b`, and `x`) remain in the workspace. To see a listing of them, enter `whos` at the command prompt.

NOTE: The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The MATLAB editor has numerous menus for tasks such as *saving*, *viewing*, and *debugging*. Because it performs some simple checks and also uses color to differentiate between various elements of codes, this text editor is recommended as the tool of choice for writing and editing M-files.

There is another way to open the editor:

```
>> edit
```

or

```
>> edit filename.m
```

to open `filename.m`.

Example 2

Plot the following cosine functions, $y_1 = 2\cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0.5 * \cos(x)$, in the interval $0 \leq x \leq 2\pi$. This example has been presented in previous Chapter. Here we put the commands in a file.

- Create a file, say `example2.m`, which contains the following commands:

```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
plot(x,y1,'--',x,y2,'-',x,y3,':')
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
legend('2*cos(x)','cos(x)','0.5*cos(x)')
title('Typical example of multiple plots')
axis([0 2*pi -3 3])
```

- Run the file by typing `example2` in the Command Window.

4.2.2 Script side-effects

All variables created in a script file are added to the workspace. This may have undesirable effects, because:

- Variables already existing in the workspace may be overwritten.
- The execution of the script can be affected by the state variables in the workspace.

As a result, because scripts have some undesirable side-effects, it is better to code any complicated applications using rather function M-file.

4.3 M-File functions

As mentioned earlier, functions are programs (or *routines*) that accept *input* arguments and return *output* arguments. Each M-file function (or *function* or *M-file* for short) has its *own* area of workspace, separated from the MATLAB base workspace.

4.3.1 Anatomy of a M-File function

This simple function shows the basic parts of an M-file.

```
function f = factorial(n)                    (1)
% FACTORIAL(N) returns the factorial of N.   (2)
% Compute a factorial value.                 (3)

f = prod(1:n);                               (4)
```

The first line of a function M-file starts with the keyword `function`. It gives the function *name* and order of *arguments*. In the case of function `factorial`, there are up to one output argument and one input argument. Table 4.1 summarizes the M-file function.

As an example, for $n = 5$, the result is,

```
>> f = factorial(5)
f =
  120
```

Table 4.1: Anatomy of a M-File function

Part no.	M-file element	Description
(1)	Function definition line	Define the function name, and the number and order of input and output arguments
(2)	H1 line	A one line summary description of the program, displayed when you request Help
(3)	Help text	A more detailed description of the program
(4)	Function body	Program code that performs the actual computations

Both *functions* and *scripts* can have all of these parts, except for the *function definition line* which applies to *function* only.

In addition, it is important to note that *function name* must begin with a letter, and must be no longer than than the maximum of 63 characters. Furthermore, the name of the text file that you save will consist of the function name with the extension `.m`. Thus, the above example file would be `factorial.m`.

Table 4.2 summarizes the differences between *scripts* and *functions*.

Table 4.2: Difference between scripts and functions

SCRIPTS	FUNCTIONS
- Do not accept input arguments or return output arguments.	- Can accept input arguments and return output arguments.
- Store variables in a workspace that is shared with other scripts	- Store variables in a workspace internal to the function.
- Are useful for automating a series of commands	- Are useful for extending the MATLAB language for your application

4.3.2 Input and output arguments

As mentioned above, the input arguments are listed inside parentheses following the function name. The output arguments are listed inside the brackets on the left side. They are used to transfer the output from the function file. The general form looks like this

```
function [outputs] = function_name(inputs)
```

Function file can have none, one, or several output arguments. Table 4.3 illustrates some possible combinations of input and output arguments.

Table 4.3: Example of input and output arguments

`function C=FtoC(F)`	One input argument and one output argument
`function area=TrapArea(a,b,h)`	Three inputs and one output
`function [h,d]=motion(v,angle)`	Two inputs and two outputs

4.4 Input to a script file

When a script file is executed, the variables that are used in the calculations within the file must have assigned values. The assignment of a value to a variable can be done in three ways.

1. The variable is defined in the script file.
2. The variable is defined in the command prompt.
3. The variable is entered when the script is executed.

We have already seen the two first cases. Here, we will focus our attention on the third one. In this case, the variable is defined in the script file. When the file is executed, the user is *prompted* to assign a value to the variable in the command prompt. This is done by using the `input` command. Here is an example.

```
% This script file calculates the average of points
% scored in three games.
% The point from each game are assigned to a variable
% by using the 'input' command.

game1 = input('Enter the points scored in the first game ');
```

```
game2 = input('Enter the points scored in the second game ');
game3 = input('Enter the points scored in the third game ');
average = (game1+game2+game3)/3
```

The following shows the command prompt when this script file (saved as `example3`) is executed.

```
>> example3
>> Enter the points scored in the first game   15
>> Enter the points scored in the second game  23
>> Enter the points scored in the third game   10

average =
        16
```

The `input` command can also be used to assign *string* to a variable. For more information, see MATLAB documentation.

A typical example of M-file function programming can be found in a recent paper which related to the solution of the ordinary differential equation (ODE) [12].

4.5 Output commands

As discussed before, MATLAB automatically generates a *display* when commands are executed. In addition to this automatic display, MATLAB has several commands that can be used to generate displays or outputs.

Two commands that are frequently used to generate output are: `disp` and `fprintf`. The main differences between these two commands can be summarized as follows (Table 4.4).

Table 4.4: `disp` and `fprintf` commands

`disp`	. Simple to use. . Provide limited control over the appearance of output
`fprintf`	. Slightly more complicated than `disp`. . Provide total control over the appearance of output

4.6 Exercises

1. Liz buys three apples, a dozen bananas, and one cantaloupe for \$2.36. Bob buys a dozen apples and two cantaloupe for \$5.26. Carol buys two bananas and three cantaloupe for \$2.77. How much do single pieces of each fruit cost?

2. Write a function file that converts temperature in degrees Fahrenheit (°F) to degrees Centigrade (°C). Use `input` and `fprintf` commands to display a mix of text and numbers. Recall the conversion formulation, $C = 5/9 * (F - 32)$.

3. Write a user-defined MATLAB function, with two input and two output arguments that determines the height in centimeters (`cm`) and mass in kilograms (`kg`)of a person from his height in inches (`in.`) and weight in pounds (`lb`).

   (a) Determine in SI units the height and mass of a 5 ft.15 in. person who weight 180 lb.

   (b) Determine your own height and weight in SI units.

Chapter 5

Control flow and operators

5.1 Introduction

MATLAB is also a *programming language*. Like other computer programming languages, MATLAB has some decision making structures for control of command execution. These decision making or *control flow* structures include `for` loops, `while` loops, and `if-else-end` constructions. Control flow structures are often used in script M-files and function M-files.

By creating a file with the extension `.m`, we can easily write and run programs. We do not need to *compile* the program since MATLAB is an interpretative (not compiled) language. MATLAB has thousand of *functions*, and you can add your own using m-files.

MATLAB provides several tools that can be used to control the *flow* of a program (*script* or *function*). In a simple program as shown in the previous Chapter, the commands are executed one after the other. Here we introduce the flow control structure that make possible to skip commands or to execute specific group of commands.

5.2 Control flow

MATLAB has four control flow structures: the `if` statement, the `for` loop, the `while` loop, and the `switch` statement.

5.2.1 The ''`if...end`'' structure

MATLAB supports the variants of "`if`" construct.

- `if ... end`
- `if ... else ... end`

- `if ... elseif ... else ... end`

The simplest form of the `if` statement is

```
if expression
   statements
end
```

Here are some examples based on the familiar quadratic formula.

1.
```
discr = b*b - 4*a*c;
if discr < 0
   disp('Warning: discriminant is negative, roots are
   imaginary');
end
```

2.
```
discr = b*b - 4*a*c;
if discr < 0
   disp('Warning: discriminant is negative, roots are
   imaginary');
else
   disp('Roots are real, but may be repeated')
end
```

3.
```
discr = b*b - 4*a*c;
if discr < 0
   disp('Warning: discriminant is negative, roots are
   imaginary');
elseif discr == 0
   disp('Discriminant is zero, roots are repeated')
else
   disp('Roots are real')
end
```

It should be noted that:

- `elseif` has no space between `else` and `if` (one word)
- no semicolon (`;`) is needed at the end of lines containing `if`, `else`, `end`
- indentation of `if` block is not required, but facilitate the reading.
- the `end` statement is required

5.2.2 Relational and logical operators

A relational operator compares two numbers by determining whether a comparison is *true* or *false*. Relational operators are shown in Table 5.1.

Table 5.1: Relational and logical operators

OPERATOR	DESCRIPTION
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
~=	Not equal to
&	AND operator
|	OR operator
~	NOT operator

Note that the "equal to" relational operator consists of two equal signs (==) (with no space between them), since = is reserved for the *assignment* operator.

5.2.3 The ''for...end'' loop

In the `for ... end` loop, the execution of a command is repeated at a fixed and predetermined number of times. The syntax is

```
for variable = expression
    statements
end
```

Usually, `expression` is a vector of the form `i:s:j`. A simple example of `for` loop is

```
for ii=1:5
    x=ii*ii
end
```

It is a good idea to indent the loops for readability, especially when they are nested. Note that MATLAB editor does it automatically.

Multiple `for` loops can be nested, in which case *indentation* helps to improve the readability. The following statements form the 5-by-5 symmetric matrix `A` with (i, j) element i/j for $j \geq i$:

```
n = 5; A = eye(n);
for j=2:n
    for i=1:j-1
        A(i,j)=i/j;
        A(j,i)=i/j;
    end
end
```

5.2.4 The ''while...end'' loop

This loop is used when the number of *passes* is not specified. The looping continues until a stated condition is satisfied. The `while` loop has the form:

```
while expression
      statements
end
```

The `statements` are executed as long as `expression` is true.

```
x = 1
while x <= 10
      x = 3*x
end
```

It is important to note that if the condition inside the looping is not well defined, the looping will continue *indefinitely*. If this happens, we can stop the execution by pressing **Ctrl-C**.

5.2.5 Other flow structures

- The `break` statement. A `while` loop can be terminated with the `break` statement, which passes control to the first statement after the corresponding `end`. The `break` statement can also be used to exit a `for` loop.
- The `continue` statement can also be used to exit a `for` loop to pass immediately to the next iteration of the loop, skipping the remaining statements in the loop.
- Other control statements include `return`, `continue`, `switch`, etc. For more detail about these commands, consul MATLAB documentation.

5.2.6 Operator precedence

We can build expressions that use any combination of *arithmetic*, *relational*, and *logical operators*. Precedence rules determine the order in which MATLAB evaluates an expression. We have already seen this in the "Tutorial Lessons".

Here we add other operators in the list. The precedence rules for MATLAB are shown in this list (Table 5.2), ordered from *highest* (1) to *lowest* (9) precedence level. Operators are evaluated from left to right.

Table 5.2: Operator precedence

PRECEDENCE	OPERATOR
1	Parentheses ()
2	Transpose (.'), power (.^), matrix power (^)
3	Unary plus (+), unary minus (−), logical negation (~)
4	Multiplication (.*), right division (./), left division (.\), matrix multiplication (*), matrix right division (/), matrix left division (\)
5	Addition (+), subtraction (−)
6	Colon operator (:)
7	Less than (<), less than or equal to (≤), greater (>), greater than or equal to (≥), equal to (==), not equal to (~=)
8	Element-wise AND, (&)
9	Element-wise OR, (|)

5.3 Saving output to a file

In addition to displaying output on the screen, the command `fprintf` can be used for writing the output to a *file*. The saved data can subsequently be used by MATLAB or other softwares.

To save the results of some computation to a file in a text format requires the following steps:

1. Open a file using `fopen`
2. Write the output using `fprintf`
3. Close the file using `fclose`

Here is an example (script) of its use.

```
% write some variable length strings to a file
op = fopen('weekdays.txt','wt');
fprintf(op,'Sunday\nMonday\nTuesday\nWednesday\n');
fprintf(op,'Thursday\nFriday\nSaturday\n');
fclose(op);
```

This file (`weekdays.txt`) can be opened with any program that can read `.txt` file.

5.4 Exercises

NOTE: Due to the teaching class during this Fall Quarter 2005, the *problems* are *temporarily* removed from this section.

Chapter 6

Debugging M-files

6.1 Introduction

This section introduces general techniques for finding *errors* in M-files. *Debugging* is the process by which you isolate and fix *errors* in your program or code.

Debugging helps to correct two kind of errors:

- **Syntax errors** - For example omitting a parenthesis or misspelling a function name.
- **Run-time errors** - Run-time errors are usually apparent and difficult to track down. They produce unexpected results.

6.2 Debugging process

We can debug the M-files using the Editor/Debugger as well as using debugging functions from the Command Window. The debugging process consists of

- Preparing for debugging
- Setting breakpoints
- Running an M-file with breakpoints
- Stepping through an M-file
- Examining values
- Correcting problems
- Ending debugging

6.2.1 Preparing for debugging

Here we use the Editor/Debugger for debugging. Do the following to prepare for debugging:

- Open the file
- Save changes
- Be sure the file you run and any files it calls are in the directories that are on the search path.

6.2.2 Setting breakpoints

Set breakpoints *to pause* execution of the function, so we can examine where the problem might be. There are three basic types of breakpoints:

- *A standard breakpoint*, which stops at a specified line.
- *A conditional breakpoint*, which stops at a specified line and under specified conditions.
- *An error breakpoint* that stops when it produces the specified type of *warning*, *error*, *NaN*, or infinite value.

You cannot set breakpoints while MATLAB is busy, for example, running an M-file.

6.2.3 Running with breakpoints

After setting breakpoints, run the M-file from the Editor/Debugger or from the Command Window. Running the M-file results in the following:

- The prompt in the Command Window changes to

  ```
  K>>
  ```

  indicating that MATLAB is in debug mode.
- The program pauses at the *first* breakpoint. This means that line will be executed when you continue. The pause is indicated by the green arrow.
- In breakpoint, we can examine variable, step through programs, and run other calling functions.

6.2.4 Examining values

While the program is paused, we can view the value of any variable currently in the workspace. Examine values when we want to see whether a line of code has produced the expected result or not. If the result is as expected, step to the next line, and continue running. If the result is not as expected, then that line, or the previous line, contains an *error*. When we run a program, the current workspace is shown in the **Stack** field. Use `who` or `whos` to list the variables in the current workspace.

Viewing values as datatips

First, we position the cursor to the left of a variable on that line. Its current value appears. This is called a *datatip*, which is like a *tooltip* for data. If you have trouble getting the datatip to appear, click in the line and then move the cursor next to the variable.

6.2.5 Correcting and ending debugging

While debugging, we can change the value of a variable to see if the *new* value produces expected results. While the program is paused, assign a new value to the variable in the Command Window, Workspace browser, or Array Editor. Then continue running and stepping through the program.

6.2.6 Ending debugging

After identifying a problem, end the debugging session. It is best to quit *debug mode* before editing an M-file. Otherwise, you can get unexpected results when you run the file. To end debugging, select **Exit Debug Mode** from the **Debug** menu.

6.2.7 Correcting an M-file

To correct errors in an M-file,

- Quit debugging
- Do not make changes to an M-file while MATLAB is in debug mode
- Make changes to the M-file
- Save the M-file
- Clear breakpoints

- Run the M-file again to be sure it produces the expected results.

For details on debugging process, see MATLAB documentation.

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA, PATNA

INTERNSHIP REPORT

IN

Linux (Operating System)

At

Spoken Tutorial

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTOF

BACHELOR OF TECHNOLOGY (ELECTRONICS AND COMMUNICATION)

BY

ARYABHATTA KNOWLEDGE UNIVERSITY, PATNA, BIHAR

SUBMITTED BY:

Md Faisal Iqbal (192014)

7th Sem

SESSION 2019-23

DECLARATION

I hereby declare that the Internship Report on **("Linux Operating System")** is an authentic record of my own work as requirements of Internship during the period of 8Week for the award of degree of B.Tech. (ECE), Netaji Subhas Institute of Technology, Bihta, under the guidance of (**Mr. Kannan M Moudgalya**).

Md Faisal Iqbal (192014)

ACKNOWLEDGEMENT

It is with the great sense of satisfaction that we present our reveal venture in training process in the form of project work. We are very fortunate to have Mr. Kannan M Moudgalya in Spoken Tutorial as our trainee and mentor.

While working on this project, their patience and valuable hints helped us in sailing through the rough seas. They were generous enough to lend us their time, experience and expertise.

We also acknowledge, with great thanks, our Head of the Centre and Program Coordinator Mr. Kannan M Moudgalya for their encouragement and suggestions that we were provided at different stages while working on this project and all the staff members of our department for all the help that they have extended to us.

Last but not the least, we are also thankful to all our friends without whose help this process would have been unexpectedly protracted and the meeting of deadlines would have been impossible.

Md Faisal Iqbal (192014)

Netaji Subhas Institute of Technology (NSIT)

Department of Electronic & Communication Engineering

CERTIFICATE

Spoken Tutorial Project at IIT Bombay

Certificate of Internship

This is to certify that **MD FAISAL IQBAL** has participated in **Student Internship Programme** from **2022-12-05** to **2023-01-30** on **Linux** organized by **IIT Bombay - Spoken Tutorial** with course material provided by Spoken Tutorial Project, IIT Bombay.

This training is offered by the Spoken Tutorial Project, IIT Bombay.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

Spoken Tutorial
Project at
IIT Bombay

Certificate for Completion of Training

This is to certify that **MD FAISAL IQBAL** successfully passed a **Linux** test, remotely conducted by the Spoken Tutorial project, IIT Bombay, under an honour invigilation system. Self learning through Spoken Tutorials and passing an online test completes the training programme.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

14647W2PHF

TABLE OF CONTENTS

1. ABSTRACT 07-07

2. INTRODUCTION OF TRAINING 08-08

3. INTRODUCTION OF ORGANIZATION 09-09

4. LINUX INTRODUCTION 10-18

   4.1 :INSTALLATION OF LINUX
   4.2:NORMAL INSTALLATION
   4.3: INSTALLATION THROUGH VIRTUAL BOX

5. LINUX BASIC COMMANDS 19-22

6. WORKING WITH FILES 23-29
   6.1: LISTING FILES
   6.2: GRANTING PERMISSION

7. USER MANAGEMENT 30-32
   7.1: ADDING AND DELETING USER
   7.2: ADDING AND DELETING HOME DIRECTORY

8. PROCESS MANAGEMENT 33-36
   8.1: LIST OF COMMANDS AND THEIR USES
   8.2: SWITCH OPERATION

9. SHELL SCRIPTING 37-42
   9.1: STEPS TO SCRIPT SHELL
   9.2: LIST OF PROGRAM OF SHELL SCRIPTING

10. SUMMARY 43-43

11. REFERENCE 44-44

1. ABSTRACT

This report discusses the training program at SPOKEN TUTORIAL.

It States the concept of Linux .

It shows a brief about Linux Operating System. It mentions somefacts about the department that was responsible for the training program.The report describes about the history, characteristic, overview and Concepts of Linux.

2. INTRODUCTION OF TRAINING

The purpose of industrial Training is to provide exposure for the students on practical engineering fields. Through this exposure, students will have better understanding of engineering practical in general and sense of frequent and possible problems. This training is part of learning process. So the exposure that uplifts the knowledge and experience of a student needs to be properly documented in the form of a report. Through this report, the experience gained can be delivered to their peers. A properly prepared report can facilitate the presentation of the practical experience in an orderly, precise and interesting manner.

I have chosen Linux as my training because it's in core of computer and having a practical knowledge in this Subject will help me to understand in better way how operating system work, the computer architecture which will help me in my carrier growth. During learning this I came to understand much more concepts.

2.1 Learning Outcomes from Training General learning outcomes:

An understanding of the principles and practice of linux.

A competence to design, write, compile, test and execute straight forward programs using a high level language.

An awareness of the need for a professional approach to design and the importance of good documentation to the finished program.

3. Introduction of the Organization

Spoken Tutorial is a multi-award winning educational content portal. Here one can learn various Free and Open Source Software all by oneself. Our self-paced, multi-lingual courses ensure that anybody with a computer and a desire for learning, can learn from any place, at any time and in a language of their choice.

Our courses are simple and easy to follow even for a beginner but they also meet the growing needs of the learner. Our engaging digital content ensures that learning happens at all levels - Basic, Intermediate and Advanced. Our content mandates side-by-side practice thereby ensuring that learners are actively learning. Many of the software taught, are used in various disciplines of Engineering, pure Sciences and several other Under-Grad and Post-Grad studies, and can be extended to Commerce, Arts and Management streams as well. Alongside these, there are some courses relevant at School level, too, which help school students to visualise difficult concepts of Math and Science. These can also be used by Teachers to prepare lesson plans, explain abstract concepts and give digital assignments to students.

Our courses are now available on **SWAYAM**. All our courses on **SWAYAM** are **AICTE approved** courses.
Your contributions:

Spoken Tutorial is a community-based website, and we welcome your contributions. Have a topic, script or video for a spoken tutorial? Get in touch with us and share your ideas. Together we will design and develop the course. Once your tutorials are reviewed by experts, they will be hosted on our site and will become part of the public domain, thus benefiting all who are interested in it.

We are also promoting translations of these tutorials into different languages. This helps those learners who are not fluent with English to learn software in a language of their choice, thereby increasing their concept clarity.
For more details, please write to contact@spoken-tutorial . org

4. INTRODUCTION TO LINUX

Linux is an open source operating system (OS). An operating system is the software that directly manages a system's hardware and resources, like CPU, memory, and storage. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

Think about an OS like a car engine. An engine can run on its own, but it becomes a functional car when it's connected with a transmission, axles, and wheels. Without the engine running properly, the rest of the car won't work.

How does Linux work?

Linux was designed to be similar to UNIX, but has evolved to run on a wide variety of hardware from phones to supercomputers. Every Linux-based OS involves the Linux kernel—which manages hardware resources—and a set of software packages that make up the rest of the operating system.

The OS includes some common core components, like the GNU tools, among others. These tools give the user a way to manage the resources provided by the kernel, install additional software, configure performance and security settings, and more.

5. INSTALLATION OF LINUX

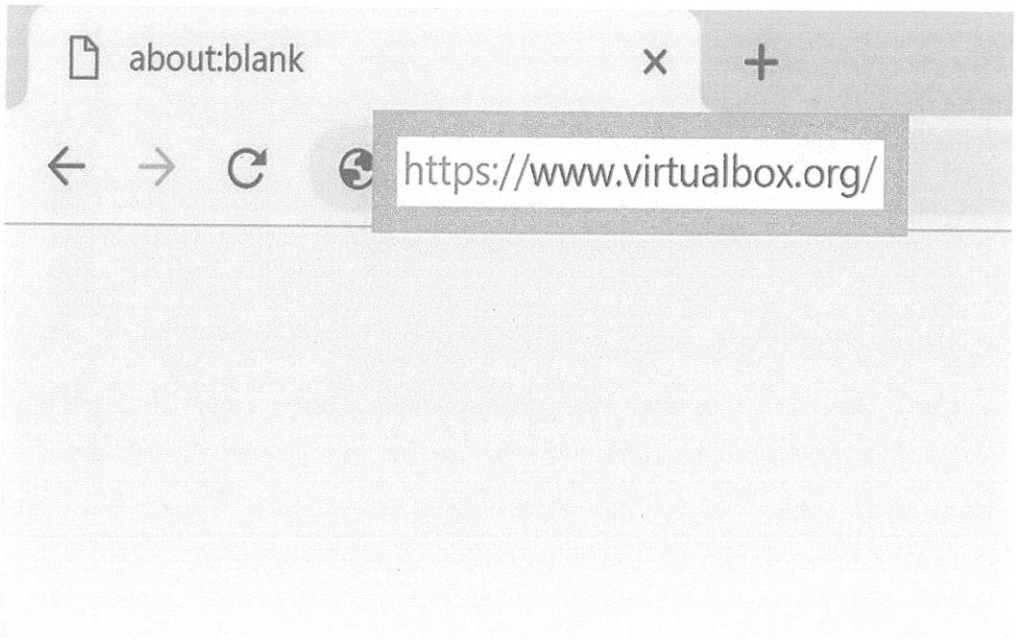
THERE ARE THREE WAYS OF INSTALLING LINUX:-

- **NORMAL INSTALLATION**
- **VIRTUAL BOX**

1. **The first way** is to download the Linux distribution you want and burn it into a DVD or USB stick and boot your machine with it and complete the installation process.

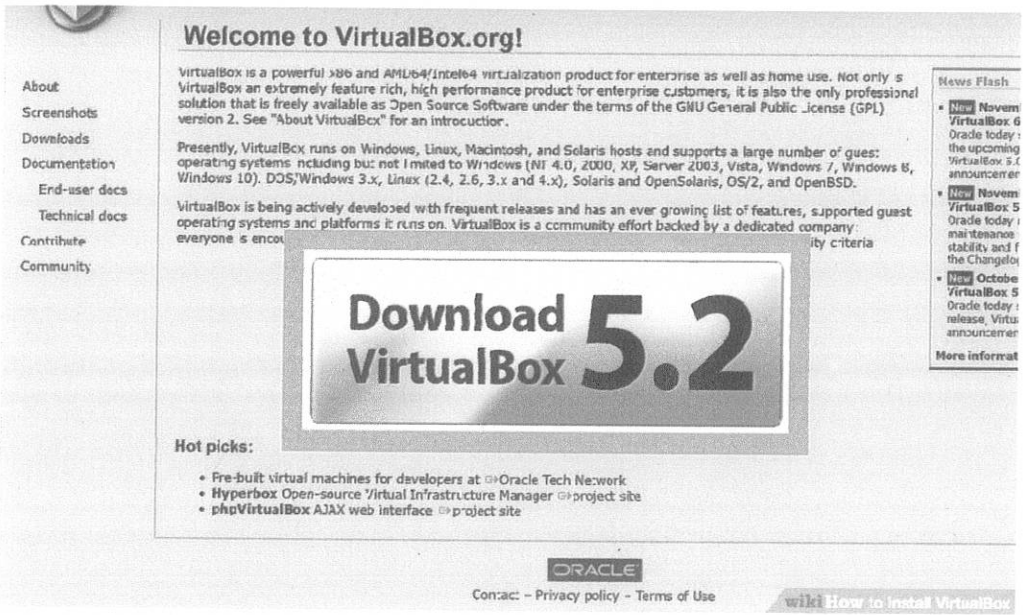
2. **The Second way** is to install it virtually on a virtual machine like Virtual Box or VMware without touching your Windows or Mac system, so your Linux system will be contained in a window you can minimize and continue working on your real system.
   For me, I prefer Virtual Box, it's free and runs very fast on my PC than VMware, and support installing Windows, Linux and Mac OS with all versions.

We can install it in windows in the following ways:-



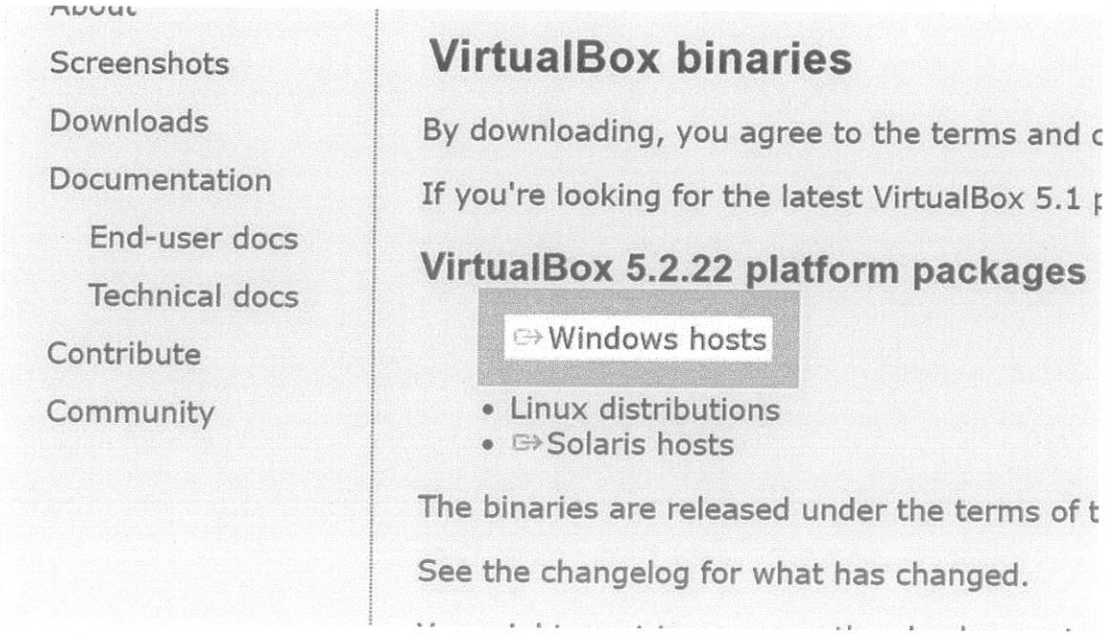
Open the Virtual Box website.

Go to https://www.Virtual Box.org/ in your computer's Internet browser.
This is the website from which you'll download the Virtual Box setup file.



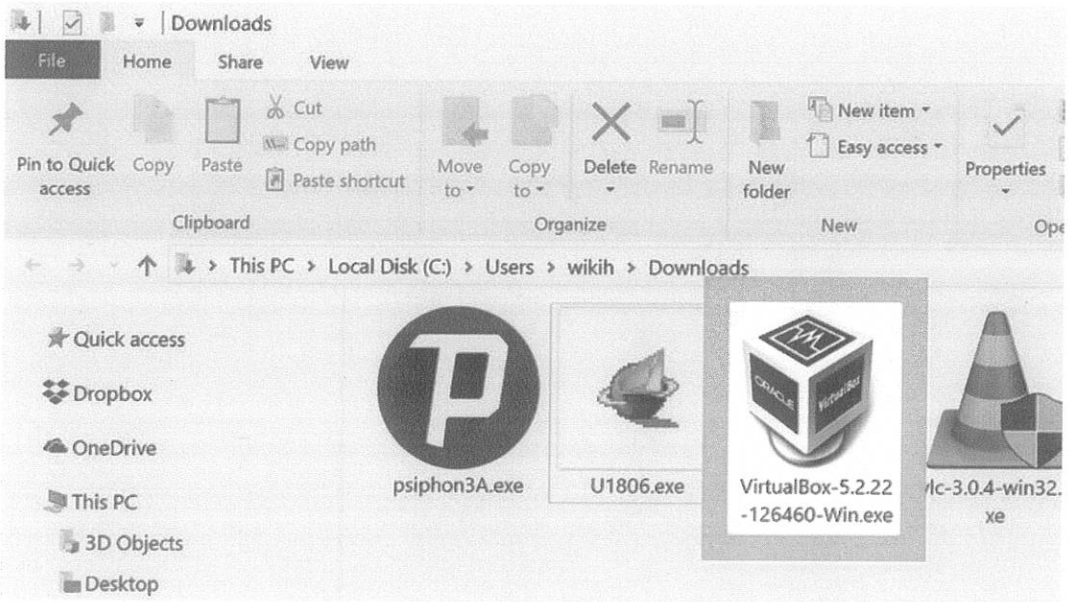
Click Download Virtual Box.

It's a blue button in the middle of the page. Doing so will open the downloads page.



Click Windows hosts.

You'll see this link below the "Virtual Box 5.2.8 platform packages" heading. The Virtual Box EXE file will begin downloading onto your computer.



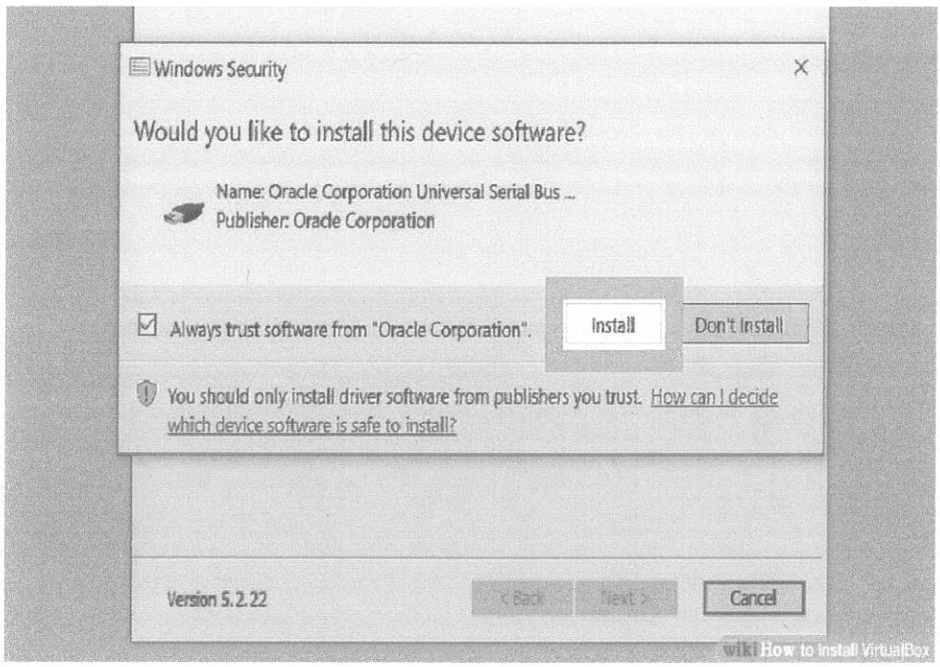
Open the Virtual Box EXE file.

Go to the location to which the EXE file downloaded and double-click the file. Doing so will open the Virtual Box installation window.

Navigate through the installation prompts.

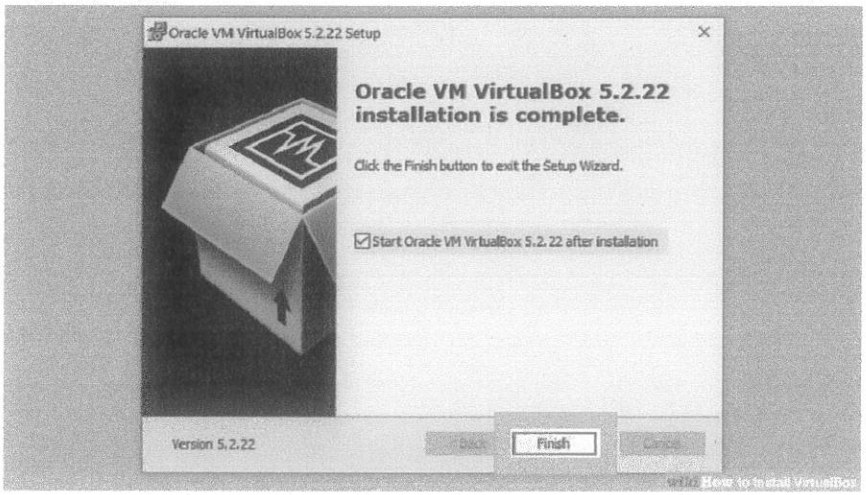
Do the following:

- Click next on the first three pages.
- Click **Yes** when prompted.
- Click Install
- Click **Yes** when prompted.



Click Install when prompted.

Doing so will allow Virtual Box to begin installing on your computer.



Click Finish when prompted.

It's in the lower-right side of the window. Doing so will close the installation window and open Virtual Box. Now that you've installed and opened Virtual Box, you can create a virtual machine in order to run any operating system on your PC.

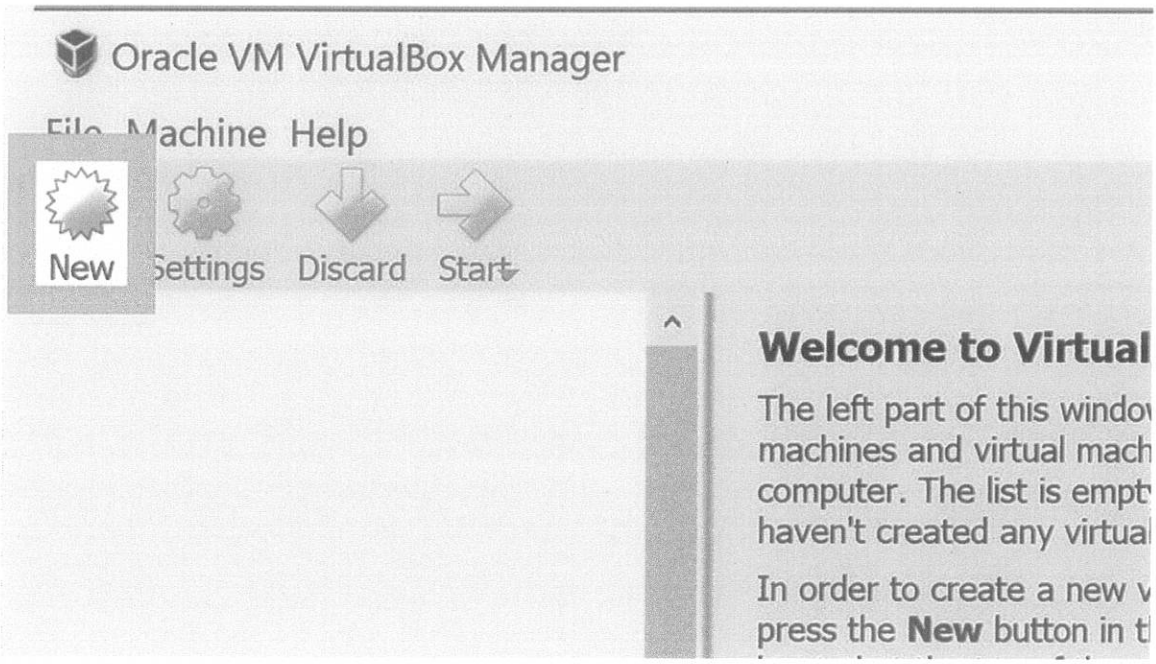
- Make sure that you don't uncheck the "Start" box before doing this.

CREATING A VIRTUAL MACHINE

- **Gather your installation disc(s) or files.**

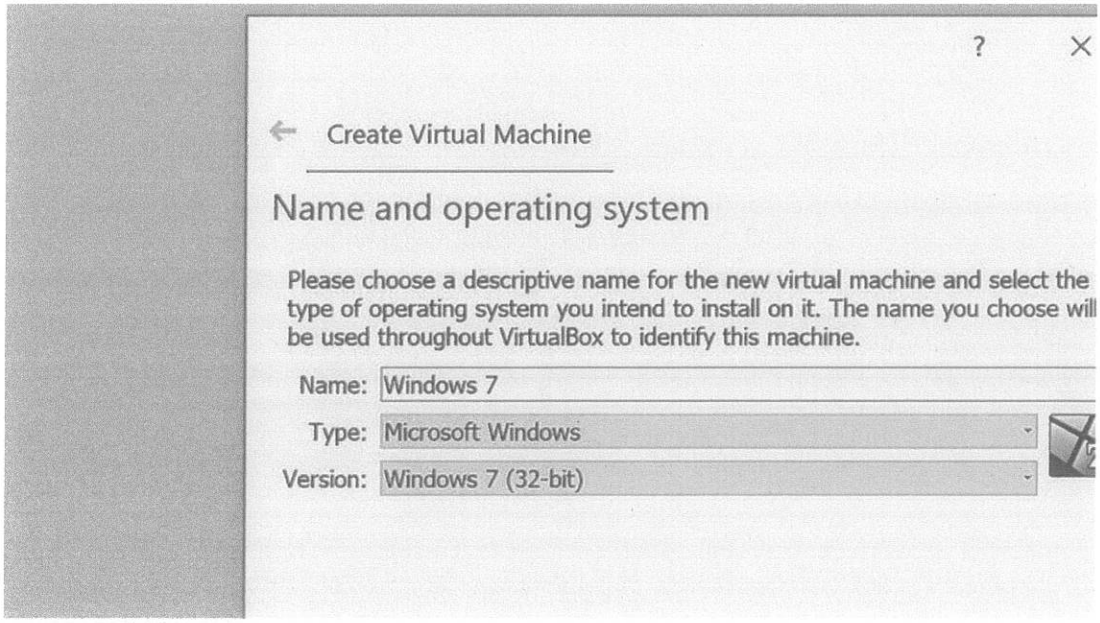
When creating a virtual machine, you will need to install the operating system just like you would on a regular computer. This means that you will need the installation disc(s) for the operating system you want to install on the virtual machine.

You can also install an operating system by using its ISO file.



Click.

New. This will open the wizard that will guide you through the process to create your first virtual machine.

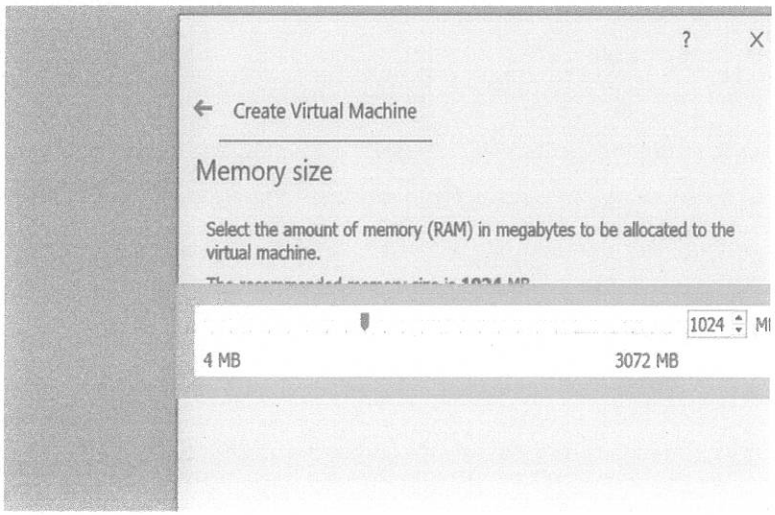


- **Identify the operating system.**

On the first screen of the wizard, you will be asked to give the new virtual machine a name as well as choose what operating system you will be installing. Choose the type of operating system from the "Type" menu, and then choose which version you are installing from the "Version" menu.

Click Next.

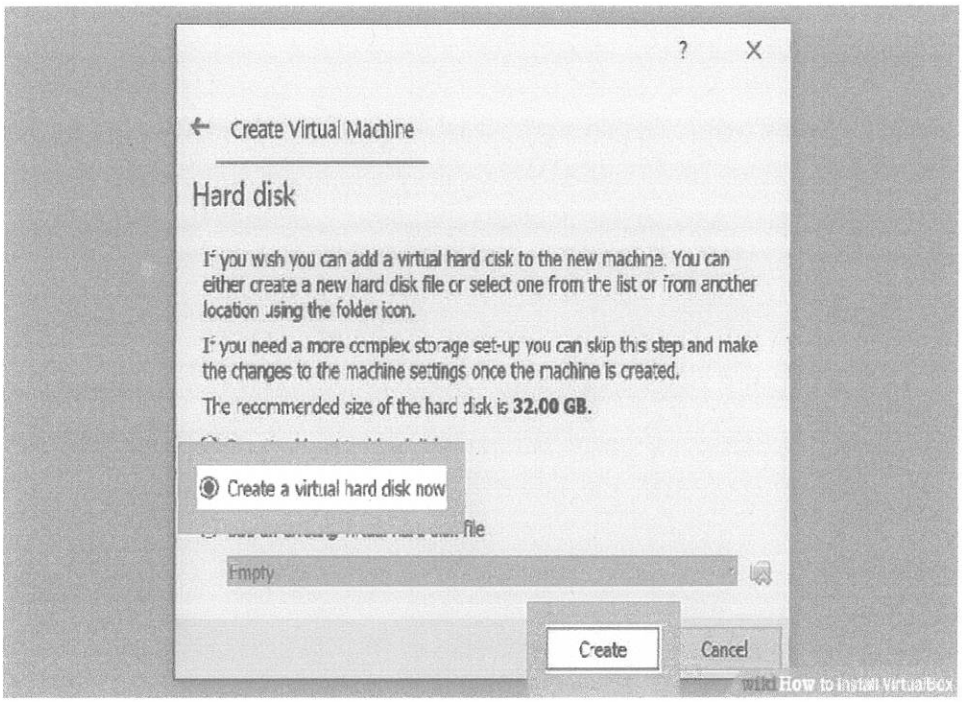
It's at the bottom of the window.



- **Set the amount of RAM.**

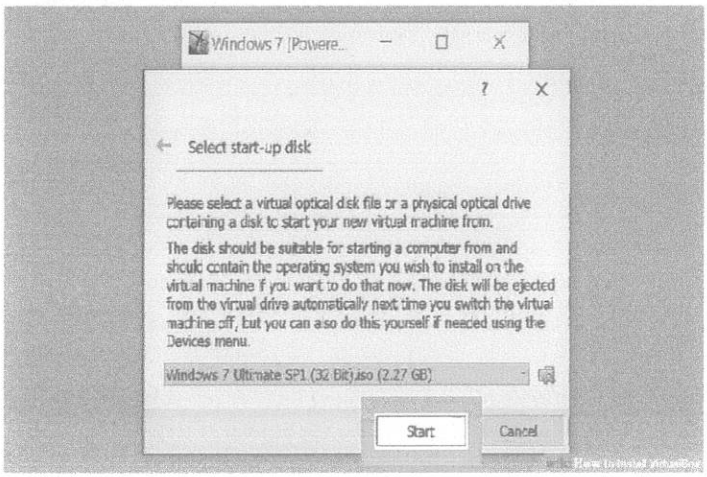
You will need to designate how much of your computer's RAM will be allocated to your virtual machine. You can only go as high as the amount of RAM physically installed in your system.

Click Next.



- **Create a virtual hard drive.**

Select a virtual hard drive option and click Create, then click through the prompts and click Create again. Your virtual machine will need a virtual hard drive in order to install the operating system and any programs.



- **Start the operating system installation.**

Once the virtual machine has been configured, the wizard will close and you will be taken back to the Virtual Box main window. Double-click your new machine in the left menu, then do one of the following:

i. If you are installing from a disc, insert it into your computer, click the "Host drive" drop-down box and click the correct drive letter from the drop-down menu.
ii. If you are installing from an image file, click the folder-shaped icon to browse through your computer for the installation image file.

Click Start.

It's at the bottom of the window. This will prompt Virtual Box to begin reading your disk or file.

- **Install the operating system.**

After selecting the installation media, the operating system installation will begin. Installation proceeds the same way it would as if you were installing the operating system on a regular computer. See the following guides for instructions on installing the operating system of your choice:

- **Boot up your virtual machine.**

Once the operating system is installed, your virtual machine is ready to go. Simply double-click the name of your virtual machine in the left menu of the Virtual Box main page to start it up. The virtual computer will boot and load into the operating system that you installed.

- **Your virtual machine will run in a window.**

Whenever the virtual machine window has focus, any keystrokes or mouse clicks will affect the virtual machine and not your physical computer.

5.Linux Basic Command

Let's start with some simple commands.

1) pwd Command

'pwd' command prints the absolute path to current working directory.

```
$ pwd
/home/raghu
```

2) cal command

Displays the calendar of the current month.

```
$ cal
July 2012
 Su Mo Tu We  Th Fr Sa
1 2  34 5  67
8 9 10 11 12 13 14
15  1617  1819  20 21
22  2324  2526  27 28
29  3031
```

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA, PATNA

INTERNSHIP REPORT

IN

Linux (Operating System)

At

Spoken Tutorial

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTOF

BACHELOR OF TECHNOLOGY (ELECTRONICS AND COMMUNICATION)

BY

ARYABHATTA KNOWLEDGE UNIVERSITY, PATNA, BIHAR

SUBMITTED BY:

Nishant Kumar Singh (192009)

7th Sem

SESSION 2019-23

Scanned with ACE Scanner

Spoken Tutorial
Project at
IIT Bombay

Certificate of Internship

This is to certify that **NISHANT KUMAR SINGH** has participated in **Student Internship Programme** from **2022-12-05** to **2023-01-30** on **Linux** organized by **IIT Bombay - Spoken Tutorial** with course material provided by Spoken Tutorial Project, IIT Bombay.

This training is offered by the Spoken Tutorial Project, IIT Bombay.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

Scanned with ACE Scanner

Certificate for Completion of Training

This is to certify that **NISHANT KUMAR SINGH** successfully passed a **Linux** test, remotely conducted by the Spoken Tutorial project, IIT Bombay, under an honour invigilation system. Self learning through Spoken Tutorials and passing an online test completes the training programme.

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

Scanned with ACE Scanner

INTERNSHIP REPORT

IN

Linux (Operating System)

At

Spoken Tutorial

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT OF

BACHELOR OF TECHNOLOGY (ELECTRONICS AND COMMUNICATION)

BY

ARYABHATTA KNOWLEDGE UNIVERSITY, PATNA, BIHAR

SUBMITTED BY:

Nikita Kumari (192005)

7th Sem

SESSION 2019-23

Scanned with ACE Scanner

Spoken Tutorial
Project at
IIT Bombay

Certificate of Internship

This is to certify that NIKITA KUMARI has participated in Student Internship Programme from 2022-12-05 to 2023-01-30 on Linux organized by IIT Bombay - Spoken Tutorial with course material provided by Spoken Tutorial Project, IIT Bombay.

This training is offered by the Spoken Tutorial Project, IIT Bombay.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutoi ial is a project at IIT Bombay, started with funding fi om the National Mission on Education through ICT
Minist ry DL EdLJca hDn (previously MH RD), GDVt Of And la

Scanned with ACE Scanner

Spoken Tutorial
Project at
IIT Bombay

Certificate of Internship

This is to certify that NIKITA KUMARI has participated in Student Internship Programme from 2022-12-05 to 2023-01-30 on Linux organized by IIT Bombay - Spoken Tutorial with course material provided by Spoken Tutorial Project, IIT Bombay.

This training is offered by the Spoken Tutorial Project, IIT Bombay.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutoi ial is a project at IIT Bombay, started with funding fi om the National Mission on Education through ICT
Minist ry DL EdLJca hDn (previously MH RD), GDVt Of And la

Scanned with ACE Scanner

INTERNSHIP REPORT
IN
Linux (Operating System)
At
Spoken Tutorial

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT OF

BACHELOR OF TECHNOLOGY (ELECTRONICS AND COMMUNICATION)

BY

ARYABHATTA KNOWLEDGE UNIVERSITY, PATNA, BIHAR

SUBMITTED BY:

Bhaskar Kumar (192007)

7th Sem

SESSION 2019-23

Scanned with ACE Scanner

Spoken Tutorial
Project at
IIT Bombay

Certificate of Internship

This is to certify that **BHASKAR KUMAR** has participated in **Student Internship Programme** from **2022-12-05** to **2023-01-30** on **Linux** organized by **IIT Bombay - Spoken Tutorial** with course material provided by Spoken Tutorial Project, IIT Bombay.

This training is offered by the Spoken Tutorial Project, IIT Bombay.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

Scanned with ACE Scanner

Spoken Tutorial
Project at
IIT Bombay

Certificate for Completion of Training

This is to certify that **BHASKAR KUMAR** successfully passed a **Linux** test, remotely conducted by the Spoken Tutorial project, IIT Bombay, under an honour invigilation system. Self learning through Spoken Tutorials and passing an online test completes the training programme.

Prof. Kannan M Moudgalya
IIT Bombay

Spoken Tutorial is a project at IIT Bombay, started with funding from the National Mission on Education through ICT, Ministry of Education (previously MHRD), Govt. of India

Scanned with ACE Scanner

प्रसार भारती

India's Public Service Broadcaster

आकाशवाणी ALL INDIA RADIO

सत्यम् शिवम् सुन्दरम् DOORDARSHAN

A report on "Implant Training" undertaken at DOORDARSHAN KENDRA (Patna)

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

BIHTA, PATNA (BIHAR)

SUBMITTED BY - PRINCE

ROLL NUMBER - 192001

REGISTRATION NUMBER - 19104103005

BRANCH - ELECTRONICS AND COMMUNICATION ENGINEERING

SEMESTER - 7TH

YEAR - 4TH

SESSION: 2019-23

DECLARATION

I, PRINCE, hereby certify that the work which is presented in this report entitled "IN-PLANT TRAINING" at DOORDARSHAN KENDRA by me in partial fulfillment of the requirements for the internship of Bachelor of Technology in Electronics & Communication Engineering from NSIT, Patna is an authentic record of my own work carried under the supervision of Mr. S.K. Jha which are duly listed in the reference section.

Date -

Prince (192001)

सत्यं शिवं सुन्दरम्

प्रसार भारती
PRASAR BHARATI

दूरदर्शन केन्द्र : पटना (बिहार)

प्रमाणित किया जाता है कि

श्री/सुश्री ...प्रिंस.....

जो ...नेताजी सुभाष इंस्टिट्यूट ऑफ टेक्नोलोजी बिहटा के/की विद्यार्थी हैं, ने दूरदर्शन केन्द्र, पटना से टेलीविजन कार्यक्रम निर्माण एवं प्रसारण की मूल तकनीक में दिनांक ...28.06.2022... से ...25.07.2022... तक ...4... सप्ताह का प्रशिक्षण सफलतापूर्वक प्राप्त किया।

This is to certify that

Mr./Ms. PRINCE

a student of NETAJI SUBHAS INSTITUTE OF TECHNOLOGY BIHTA

underwent and successfully completed his/ her Vocational Training in Basic Techniques of Television Production & Transmission at Doordarshan Kendra, Patna for a period from 28.06.2022 to 25.07.2022 of 4 weeks.

हम भविष्य में सभी क्षेत्रों में इनकी सफलता की कामना करते हैं।
We wish him/her every success in all future endeavours.

Serial No. 004

दिनांक : 29/07/2022
Date :

उप महानिदेशक सह केन्द्राध्यक्ष
DDG CUM HOO

ACKNOWLEDGEMENT

At the outset, I would like to express my gratitude to DOORDARSHAN KENDRA, PATNA who gave me this opportunity to learn something new and related to our branch that will be useful in my academics and career.

A very special vote of thanks to Mr. S. K. Jha sir for guiding us through our training period.

CONTENTS

1. History

2. Orbital Abstract of Satellite Communication

3. Working of a Satellite

4. Satellite Link Design

5. Bibliography

HISTORY

A satellite is an object which has been placed into orbit by human endeavor. Such objects are sometimes called artificial satellites to distinguish them from natural satellites such as moon. The world's first artificial satellite, the SPUTNIK 1, was launched by Soviet Union in 1957.

Satellites are used for a large number of purposes. Common types include military and civilian Earth observation satellites, communications satellites, navigation satellites, weather satellites, and research satellites. Space stations and human spacecraft in orbit are also satellites. Satellite orbits vary greatly, depending on the purpose of the satellite, and are classified in a number of ways. Well-known (overlapping) classes include low Earth orbit, polar orbit, and geostationary orbit.

Satellites are usually semi-independent computer-controlled systems. Satellite subsystems attend many tasks, such as power generation, thermal control, telemetry, altitude control and orbit control.

Sputnik 2 was launched on November 3, 1957 and carried the first living passenger into orbit, a dog named Laika.

Order	Country	Year of first launch	Rocket	Satellite
1	Soviet Union	1957	Sputnik-PS	*Sputnik 1*
2	United States	1958	Juno I	*Explorer 1*
3	France	1965	Diamant	*Astérix*
4	Japan	1970	Lambda-4S	Ōsumi
5	China	1970	Long March 1	*DFH I*
6	United Kingdom	1971	Black Arrow	*Prospero X-3*
7	India	1980	SLV	*Rohini*
8	Israel	1988	Shavit	*Ofeq 1*
9	Russia[1]	1992	Soyuz-U	*Kosmos 2175*
10	Ukraine[1]	1992	Tsyklon-3	*Strela*
11	Iran	2009	Safir-2	*Omid*

ORBITAL ASPECT OF SATELLITE COMMUNIATION

Altitude classifications

- Low Earth orbit (LEO): Geocentric orbits ranging in altitude from 0 2000 km (0 1240 miles)
- Medium Earth orbit (MEO): Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 km (22,236 mi). Also known as an intermediate circular orbit.
- High Earth orbit (HEO): Geocentric orbits above the altitude of geosynchronous orbit 35,786 km (22,236 mi).

Inclination classifications

- **Inclined orbit**: An orbit whose inclination in reference to the equatorial plane is not zero degrees.
- **Polar orbit**: An orbit that passes above or nearly above both poles of the planet on each revolution. Therefore, it has an inclination of (or very close to) 90 degrees.
- **Polar sun synchronous orbit**: A nearly polar orbit that passes the equator at the same local time on every pass. Useful for image taking satellites because shadows will be nearly the same on every pass.

Eccentricity classifications

- **Circular orbit**: An orbit that has an eccentricity of 0 and whose path traces a circle.
- **Hohmann transfer orbit**: An orbital maneuver that moves a spacecraft from one circular orbit to another using two engine impulses. This maneuver was named after Walter Hohmann

- **Elliptic orbit**: An orbit with an eccentricity greater than 0 and less than 1 whose orbit traces the path of an ellipse.

- **Geosynchronous transfer orbit**: An elliptic orbit where the perigee is at the altitude of a Low Earth orbit (LEO) and the apogee at the altitude of a geosynchronous orbit.

- **Geostationary transfer orbit**: An elliptic orbit where the perigee is at the altitude of a Low Earth orbit (LEO) and the apogee at the altitude of a geostationary orbit.

- **Molniya orbit**: A highly elliptic orbit with inclination of 63.4° and orbital period of half of a sidereal day (roughly 12 hours). Such a satellite spends most of its time over two designated areas of the planet (specifically Russia and the United States).

- **Tundra orbit**: A highly elliptic orbit with inclination of 63.4° and orbital period of one sidereal day (roughly 24 hours). Such a satellite spends most of its time over a single designated area of the planet.

Look Angle of Satellite Communication

The coordinates to which the earth station antennas must be pointed to communicate with the satellite is called Look angles.
There r 2 types of look angles

1. Azimuth angle: Measured eastward from geographic north to the projection of satellite path on the local horizontal plane at the earth station.

2. Elevation Angle: Measured upward from local horizontal plane at the earth station to the satellite path.

Azimuth and Elevation Calculations

The azimuth and elevation angles for our earth station antenna must be calculated so that the correct satellite can be seen. The azimuth is the horizontal pointing angle of the earth station antenna. The elevation is the angle we look up into the sky to see the satellite. To calculate the azimuth and elevation of a ground station antenna requires that the ground station latitude and longitude be known as well as the longitude of the satellite.
The longitude of the satellite can be obtained from charts such as the WESTSAT. The latitude and longitude of the ground station can be obtained from U.S. Geological Survey maps of the region you are in. More advanced methods of locating the exact ground station are available through the use of the Global Positioning Satellite (GPS). Once the actual location is known the elevation angle of the earth station antenna can be calculated using equation 1.

$$E = \arctan\left[\frac{\cos(G)\cos(L) - .1512}{\sqrt{1-\cos^2(G)\cos^2(L)}}\right]$$

E = elevation of antenna in degrees
S = satellite longitude in degrees
N = site longitude in degrees
L = site latitude in degrees
G = S − N

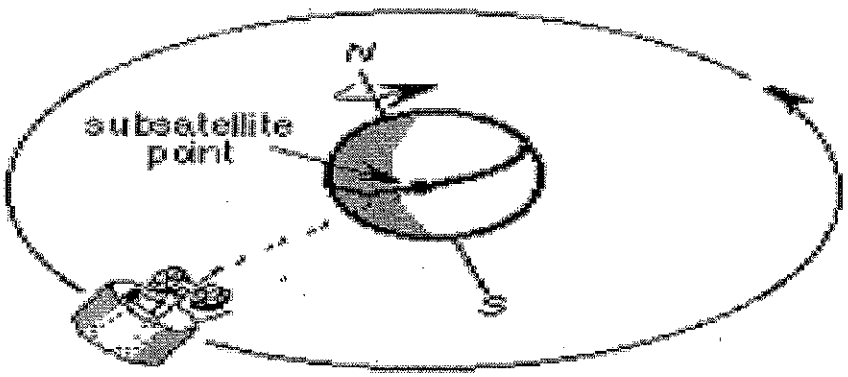
Next the azimuth can be calculated using equation 2.

$$A = 180 + \arctan\left[\frac{\tan(G)}{\sin(L)}\right]$$

A = azimuth of antenna in degrees
S = satellite longitude in degrees
N = site longitude in degrees
L = site latitude in degrees
G = S – N

Parking slot of a satellite

Latitude and longitude of a sub-satellite point are called as parking slot of the satellite where sub-satellite point stands for a point where a straight line drawn from a satellite to the center of the Earth intersects the Earth's surface.



parking slot is the longitudinal co-ordinates in case of geo-stationary satellites

WORKING OF SATELLITES

- Service Types
- Fixed Service Satellites (FSS) Example: Point to Point Communication.
- Broadcast Service Satellites (BSS)

  Example: Satellite Television/Radio Also called Direct Broadcast Service (DBS).
- Mobile Service Satellites (MSS)
  Example: Satellite Phones

Frequency bands:

- ✓ L Band: 1 to 2 GHz, used by MSS
- ✓ S-Band: 2 to 4 GHz, used by MSS, NASA, deep space research
- ✓ C-Band: 4 to 8 GHz, used by FSS
- ✓ X-Band: 8 to 12.5 GHz, used by FSS and in terrestrial imaging, ex: military and meteorological satellites
- ✓ Ku-Band: 12.5 to 18 GHz: used by FSS and BSS (DBS)
- ✓ K-Band: 18 to 26.5 GHz: used by FSS and BSS
- ✓ Ka-Band: 26.5 to 40 GHz: used by FSS

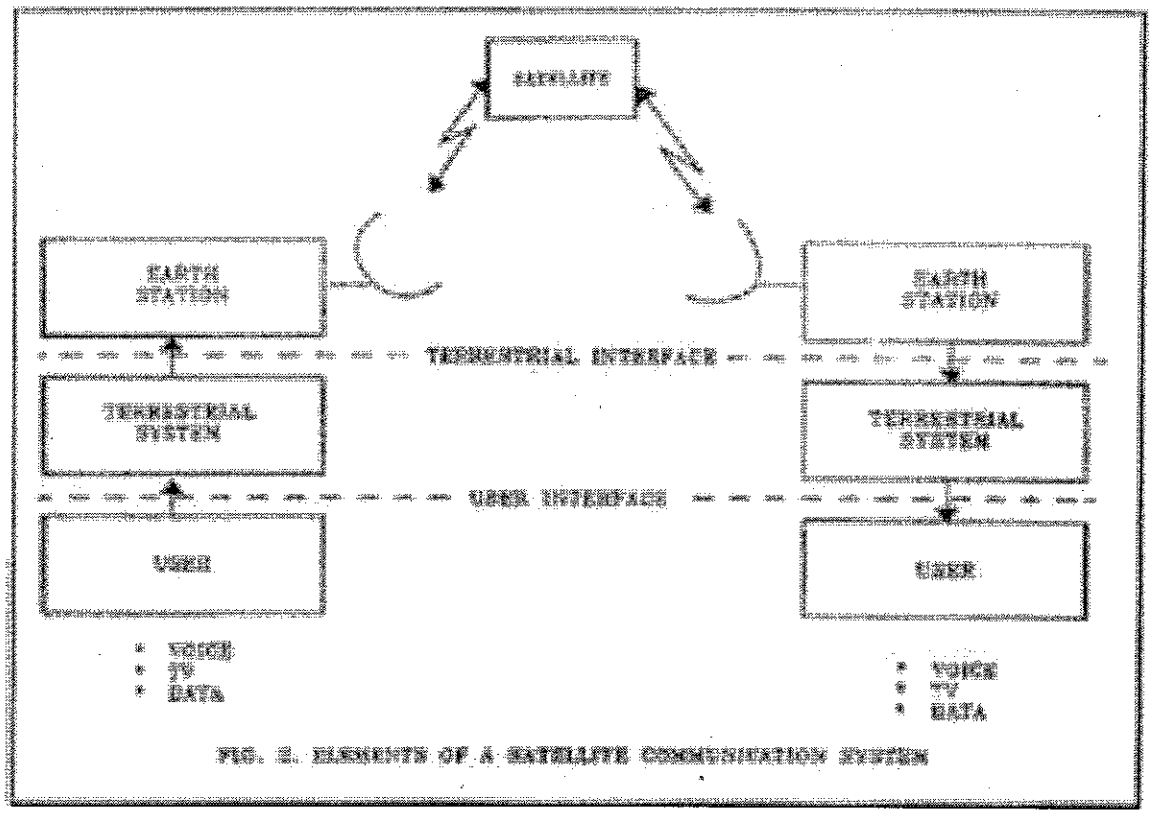


FIG. 2. ELEMENTS OF A SATELLITE COMMUNICATION SYSTEM

Communication Satellite:

Communication satellites are very complex and extremely expensive to procure & launch.

The communication satellites are now designed for 12 to 15 years of life during which the communication capability of the satellite earns revenue, to recover the initial and operating costs. Since the satellite has to operate over a long period out in the space the subsystems of the satellite are required to be very reliable. Major subsystems of a satellite are:

- Satellite Bus Subsystems
  Satellite Payload

Satellite Payloads

- Communication transponders
- Communication Antennas

Satellite Bus subsystems:

- Mechanical structure
- Attitude and orbit control system
- Propulsion System
- Electrical Power System
- Tracking Telemetry and Command System
- Thermal Control System

Since the communications capacity earns revenue, the satellite must carry as many communications channels as possible. However, the large communications channel capacity requires large electrical power from large solar arrays and battery, resulting in large mass and volume. Putting a heavy satellite in geosynchronous orbit being very expensive, it is logical to keep the size and mass of the satellite small. Lightweight material optimally designed to carry the load and withstand vibration & large temperature cycles are selected for the structure of the satellite.

Attitude and orbit control system maintains the orbital location of the satellite and controls the attitude of the satellite by using different sensors and firing small thrusters located in different sides of the satellite. Liquid fuel and oxidizer are carried in the satellite as part of the propulsion system for firing the thrusters in order to maintain the satellite attitude and orbit. The amount of fuel and oxidizer carried by the satellite also determines the effective life of satellite.

The electrical power in the satellite is derived mainly from the solar cells. The power is used by the communications payloads and also by all other electrical subsystems in the satellite for housekeeping. Rechargeable battery is used for supplying electrical power during ellipse of the satellite.

Telemetry, Tracking and Command system of the satellite works along with its counterparts located in the satellite control earth station. The telemetry system collects data from sensors on board the satellite and sends these data via telemetry link to the satellite control center which monitors the health of the satellite. Tracking and ranging system located in the earth station provides the information related to the range and location of the satellite in its orbit. The command system is used for switching on/off of different subsystems in the satellite based on the telemetry and tracking data.

The thermal control system maintains the temperature of different parts of the satellite within the operating temperature limits and thus protects the satellite subsystems from the extreme temperature conditions of the outer space.

The communications subsystems are the major elements of a communication satellite and the rest of the space craft is there solely to support it. Quite often it is only a small part of the mass and volume of the satellite. The communications subsystem consists of one or more antennas and communications receiver - transmitter units known as transponders. Transponders are of two types, Repeater or Bent pipe and processing or regenerative. In Repeater type, communications transponder receives the signals at microwave frequencies and amplifies the RF carrier after frequency conversion, whereas in processing type of transponder in addition to frequency translation and amplification, the RF carrier is demodulated to baseband and the signals are regenerated and modulated in the transponder. Analog communication systems are exclusively repeater type. Digital communication system may use either variety. Fig. 3(a) and 3(b) show the schematic diagrams of repeater type and regenerative type *transponders respectively* regenerative type *transponders respectively.*

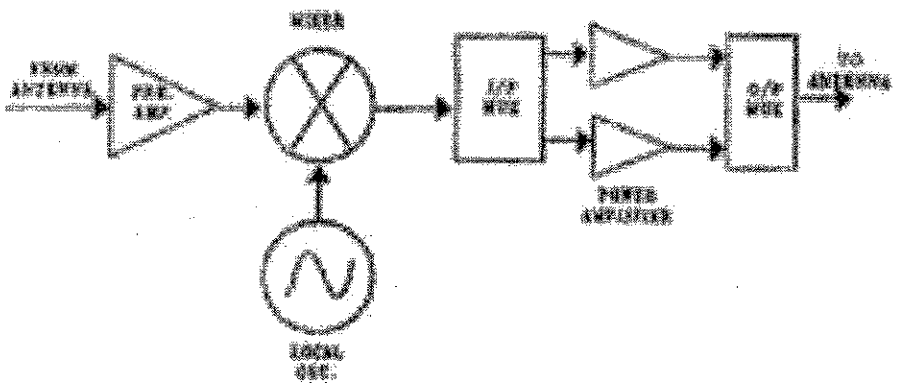


FIG. 3(a). SCHEMATIC DIAGRAM OF REPEATER TYPE OF TRANSPONDER

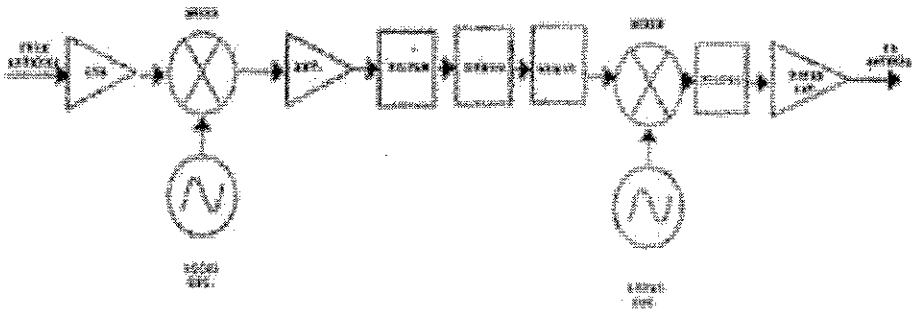


FIG. 3(b) SCHEMATIC DIAGRAM OF REGENERATIVE TYPE OF TRANSPONDER

Ground Segment:

The ground segment of satellite communications system establishes the communications links with the satellite and the user. In large and medium systems the terrestrial microwave link interfaces with the user and the earth station. However, in the case of small systems, this interface is eliminated and the user interface can be located at the earth station. The earth station consists of

- Transmit equipment.
- Receive equipment.
- Antenna system.

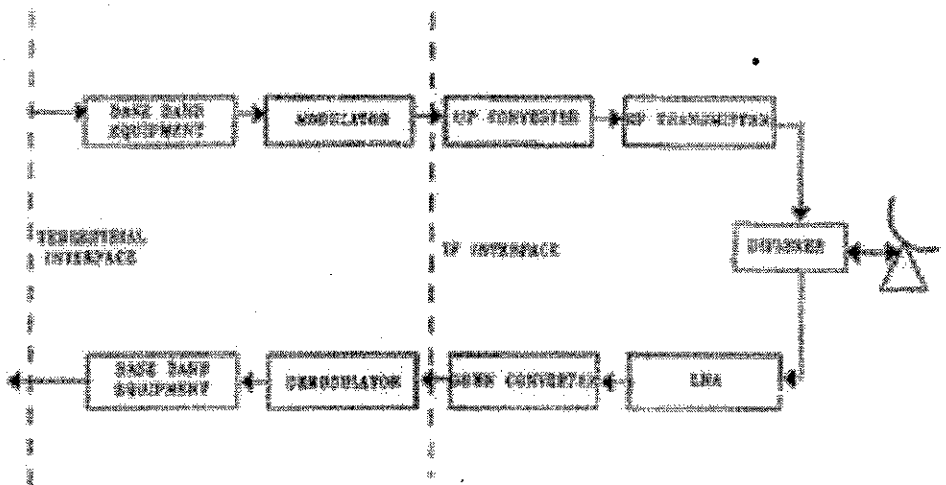


FIG. 4. SCHEMETIC BLOCK DIAGRAM OF AN EARTH STATION

In the earth station the base band signal received directly from users' premises or from terrestrial network are appropriately modulated and then transmitted at RF frequency to the satellite. The receiving earth station after demodulating the carrier transmits the base band signal to the user directly or through the terrestrial link.

The baseband signals received at the earth stations are mostly of the following types.

- Groups of voice band analog or digital signals
- Analog or digital video signals
- Single channel analog or digital signal
- Wide band digital signal.

In satellite communications, in early days FM modulation scheme was most frequently used for analog voice and video signal transmission. However, the trend is now to use digital signals for both voice and video. Various digital modulation schemes like Phase Shift Keying (PSK) and Frequency Shift Keying (FSK) are adopted for transmission of digital signals.

The network operations and control center for the communications network monitors the network operations by different users, distribution of different carriers within a transponder and allocation of bandwidth & EIRP of different carriers. Proper functioning of Network operations and control center is essential where the number of users in the network is large.
Network operations & control center is also responsible for giving clearance to the ground system in respect of antenna radiation pattern, EIRP etc.

SATELLITE LINK DESIGN

Basic Transmission Theory:

A link budget is the accounting of all of the gains and losses from the transmitter, through the medium (free space, cable, waveguide, fiber, etc.) to the receiver in a telecommunication system. It accounts for the attenuation of the transmitted signal due to propagation, as well as the antenna gains, feed line and miscellaneous losses. Randomly varying channel gains such as fading are taken into account by adding some margin depending on the anticipated severity of its effects. The amount of margin required can be reduced by the use of mitigating techniques such as antenna diversity or frequency hopping.

A simple link budget equation looks like this:

Received Power (dBm) = Transmitted Power (dBm) + Gains (dB) – Losses (dB)

Note that decibels are logarithmic measurements, so adding decibels is equivalent to multiplying the actual numeric ratios.

Link Budget Parameters:

- Transmitter power at the antenna
- Antenna gain compared to isotropic radiator
- EIRP
- Flux density at receiver
- Free space path loss
- System noise temperature
- Figure of merit for receiving system
- Carrier to thermal noise ratio
- Carrier to noise density ratio
- Carrier to noise ratio

Isotopic Radiator:

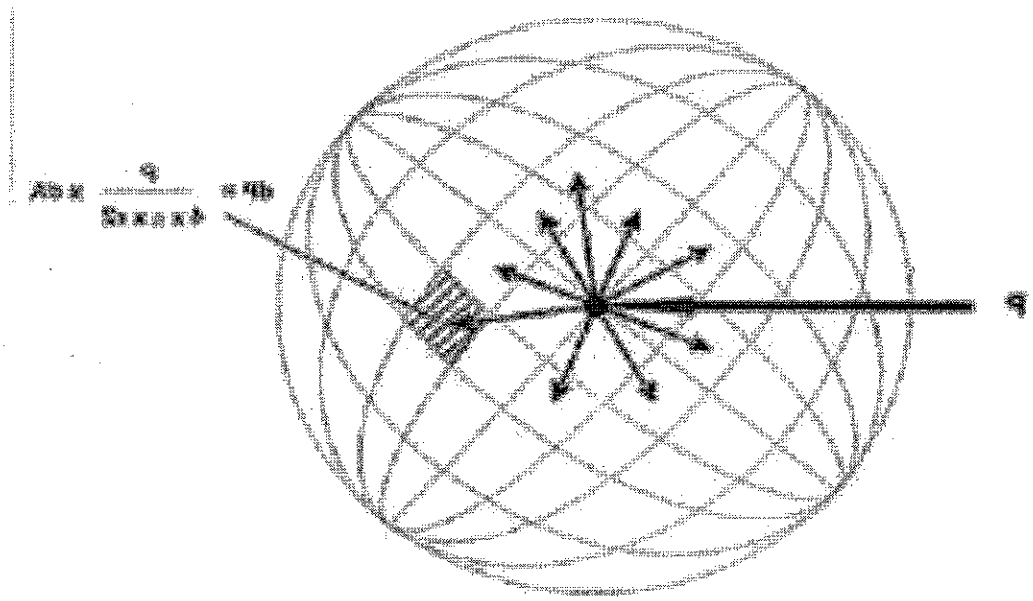
An isotropic radiator is a theoretical point source of electromagnetic or sound waves which radiates the same intensity of radiation in all directions. It has no preferred

direction of radiation. It radiates uniformly in all directions over a sphere centered on the source. Isotropic radiators are used as reference radiators with which other sources are compared.

Whether a radiator is isotropic is independent of whether it obeys Lambert's law· As radiators, a spherical black body is both, a flat black body is Lambertian but not isotropic, a flat chrome sheet is neither, and by symmetry the Sun is isotropic, but not Lambertian on account of limb darkening.

Consider an Isotropic Source (punctual radiator) radiating Pt Watts uniformly into free space. At distance R, the area of the spherical shell with center at the source is 4pR2 Flux density at distance R is given by Eq. 4.1

$$F \square \frac{P_t}{4 Y_2 R} \quad \text{W/m}^2$$



Antenna Gain:

In electromagnetics, an antenna's power gain or simply gain is a key performance figure which combines the antenna's directivity and electrical efficiency. As a transmitting antenna, the figure describes how well the antenna converts input power into radio waves headed in a specified direction. As a receiving antenna, the figure describes how well the antenna converts radio waves arriving from a specified direction into electrical power. When no direction is specified, "gain" is understood to refer to the peak value of the gain. A plot of the gain as a function of direction is called the radiation pattern.

Antenna gain is usually defined as the ratio of the power produced by the antenna from a far-field source on the antenna's beam axis to the power produced by a hypothetical lossless isotropic antenna, which is equally sensitive to signals from all directions. Usually this ratio is expressed in decibels, and these units are referred to as "decibels-isotropic" (dBi).

Antenna has gain in every direction! Term gain may be confusing sometimes. Usually "Gain" denotes the maximum gain of the antenna. The direction of maximum gain is called "boresight".

$G\ [dB] = 10 \log_{10} (G\ ratio)$

Antenna gain is relative to this standard. Antennas are fundamentally passive:

- No additional power is generated
- Gain is realized by focusing power
- Similar to the difference between a lantern and a flashlight

Effective Isotropic Radiated Power (EIRP):

EIRP is the amount of power the transmitter would have to produce if it was radiating to all directions equally

Note that EIRP may vary as a function of direction because of changes in the antenna gain vs. angle

The output power of a transmitter HPA is P_{out} watts.

Some power is lost before the antenna $P_t = P_{out}/L_t$ watts reaches the antenna (P_t = Power into antenna)

The antenna has a gain of G_t relative to an isotropic radiator

This gives an effective isotropic radiated power of EIRP = $P_t\ G_t$ watts relative to a 1 watt isotropic radiator.

Power flux density: We now want to find the power density at the receiver. We know that power is conserved in a lossless medium The power radiated from a transmitter must pass through a spherical shell on the surface of which is the receiver. The area of this spherical shell is $4pR^2$. Therefore spherical spreading loss is $1/4pR^2$.

Power flux density (p.f.d.) is a measure of the power per unit area. This is a regulated parameter of the system. CCIR regulations limit the p.f.d. of any satellite system. CCIR regulations are enforced by signatory nations. Allowable p.f.d. varies w.r.t. elevation angle. It also allows control of interference. There is increasing importance with proliferation of LEO systems.

Effective aperture: Real antennas have effective flux collecting areas which are LESS than the physical aperture area.

Effective Aperture Area Ae is A_e □ A_{phy} x □ where A_{phy} is actual physical aperture area and □ = aperture efficiency.

Aperture antennas: (horns and reflectors) have physical collecting area that can be easily calculated from their dimensions.

$$A_{phy} = \pi r^2 = \pi \frac{D^2}{4}$$

Why db is used as a unit for calculations?

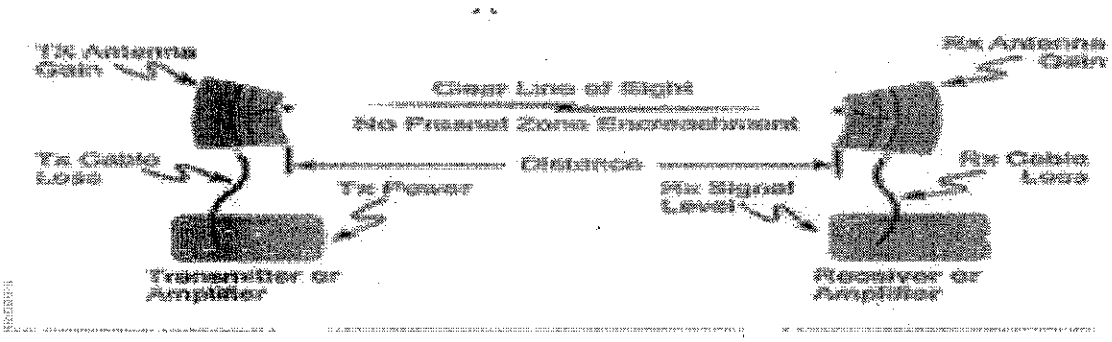
There is a large dynamic range of parameters in satellite communications. A typical satellite antenna has a gain of >500.Received power flux is about one part in 100,000,000,000,000,000,000of the transmitted power.

Wouldn't it be nice to have a better way to write these large numbers? dB also lets many calculations be addition or subtraction! Decibel (dB) is the unit for 10 times the base 10 logarithmic ratio of two powers For instance: gain is defined as P_{out}/P_{in} (where P_{out} is usually greater than P_{in})

In db Gain is expressed as $$G = 10 \cdot \log\left(\frac{P_{out}}{P_{in}}\right) \text{dB}$$

Similarly loss can be expressed as $$L = 10 \log\left(\frac{P_{in}}{P_{out}}\right) \text{dB}$$

Link Budget:



Need for calculation of a Link Budget

System performance is tied to operation thresholds. Operation thresholds C_{min} tell the minimum power that should be received at the demodulator in order for communications to work properly.

Operation thresholds depend on:

- Modulation scheme being used.
- Desired communication quality.
- Coding gain.
- Additional overheads.
- Channel Bandwidth.
- Thermal Noise power.

System figure of merit:

G/T_s: RX antenna gain/system temperature which is also called the System Figure of Merit, G/T_s. It easily describes the sensitivity of a receive system .It must be used with caution:

Some (most) vendors measure G/T_s under ideal conditions only. G/T_s degrades for most systems when rain loss increases. This is caused by the increase in the sky noise component. This is in addition to the loss of received power flux density.

System noise power:

- Performance of system is determined by C/N ratio.
- Most systems require C/N > 10 dB.
- (Remember, in dBs: C - N > 10 dB)
- Hence usually: C > N + 10 dB
- We need to know the noise temperature of our receiver so that we can calculate N, the noise power ($N = P_n$).
- T_n (noise temperature) is in Kelvin's (symbol K):

$$T[K] = T\,^{o}C + 273 \qquad T[K] = (T[^{o}F] - 32)\frac{5}{9} + 273$$

System noise is caused by thermal noise sources.

External to RX system:

- Transmitted noise on link
- Scene noise observed by antenna

Internal to RX system

The power available from thermal noise is:

$$N \quad kT_S B \quad \textbf{(dBW)}$$

where k = Boltzmann's constant

$= 1.38 \times 10^{-23}$ J/K(-228.6 dBW/HzK),

T_s is the effective system noise temperature, and B is the effective system bandwidth.

BIBLIOGRAPHY

- Wikipedia
- Satellite communication book by Dennis Roddy
- www.satcoms.org.uk
- technologyinterface.nmsu.edu
- www.meteor.iastate.edu
- www.cosmic.ucar.edu

प्रसार भारती

India's Public Service Broadcaster

ALL INDIA RADIO

DOORDARSHAN

A report on "Implant Training" undertaken at DOORDARSHAN KENDRA (Patna)

BIHTA, PATNA (BIHAR)

SUBMITTED BY – PRINCE KUMAR

ROLL NUMBER - 192017

REGISTRATION NUMBER - 19104103008

BRANCH - ELECTRONICS AND COMMUNICATION ENGINEERING

SEMESTER - 7TH

YEAR - 4TH

SESSION: 2019-23

Scanned with ACE Scanner

प्रसार भारती

PRASAR BHARATI

दूरदर्शन केन्द्र : पटना (बिहार)

प्रमाणित किया जाता है कि

श्री/सुश्री प्रिंस कुमार

जो नेताजी सुभाष इंस्टिट्यूट ऑफ टेक्नोलोजी बिहटा के/की विद्यार्थी हैं, ने दूरदर्शन केन्द्र, पटना से टेलीविजन कार्यक्रम निर्माण एवं प्रसारण की मूल तकनीक में दिनांक 28.06.2022 से 25.07.2022 तक 4 सप्ताह का प्रशिक्षण सफलतापूर्वक प्राप्त किया।

This is to certify that

Mr./Ms. PRINCE KUMAR

a student of NETAJI SUBHAS INSTITUTE OF TECHNOLOGY BIHTA

underwent and successfully completed his/ her Vocational Training in Basic Techniques of Television Production & Transmission at Doordarshan Kendra, Patna for a period from 28.06.2022 to 25.07.2022 of 4 weeks.

हम भविष्य में सभी क्षेत्रों में इनकी सफलता की कामना करते हैं।

We wish him/her every success in all future endeavours.

Serial No. 004

दिनांक : 29/07/2022

Date :

उप महानिदेशक सह केन्द्राध्यक्ष

DDG CUM HOO

Scanned with ACE Scanner

Internship report on Python

Submitted in partial fulfillment of the requirements for

Bachelor of Engineering Degree in

Electronics & Communication Engineering

DEPARTMENT OF

ELECTRONICS & COMMUNICATION

Netaji Subhas Institute Of Technology, Bihta

PATNA – 801103 . BIHAR.

Submitted By

Name - Piyush Kumar

Reg. No. - 22104103902

Branch - ECE

Sem / year - 3rd Sem /2nd year

Scanned with ACE Scanner

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

It is Affiliated to Bihar Engineering University (BEU) Amraha, Bihta ,

Patna –801103

ELECTRONICS & COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this internship Report is the Bonafide work of **Piyush kumar (22104103902)** who Underwent the internship in **"Python"** from 15th May to 10th june

Head of the Department (HOD)

RAJNI RANJAN

Signature

Scanned with ACE Scanner

Polytropic Services Pvt. Ltd.
(An ISO 9000-2015 & ISO 27001-2013 Certified ICT Company)

Ref #: INT/2306/113

Date: 13 June 2023

Internship Certificate

This is to certify that **Mr. PIYUSH KUMAR, Reg. No 22104103902,** student of **Netaji Subhas Institute of Technology (NSIT), Bihta, Patna,** ***Session 2021 - 2025*** of **Electronics & Communication Engineering** has undergone ***Four Week*** Internship with our Organization in **Python** from ***15 May 2023 to 10 June 2023.***

During the internship duration, he has done a good job in his Project work and is found to be sincere,hardworking with a positive approach.

We wish **Mr. PIYUSH KUMAR** all the best in his future career endeavors.

For, ***Polytropic Services Pvt Ltd.***

Polytropic Services Pvt. Ltd. Patna

Authorized Signatory
(Seal and Sign)

(**NOTE:** You may verify this certificate on - https://training.polytropicservices.com/)

Corporate Office: IT Startup Hub, Dept. of IT, Govt. of Bihar, 13th Floor, Biscomaun Tower, Gandhi Maidan, Patna
Branch Office: Hitio 29, Prayag Niwas, North SK Puri, Boring Road, Patna-800001
Email Us: admin@polytropicservices.com, polytropicservices@gmail.com
Call Us: 9431051799, 9801946661, 9473010949 **Visit Us:** www.polytropicservices.com

Scanned with ACE Scanner

Acknowledgement

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

The seminar on **"Python"** was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the seminar. Their contributions and technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledgement.

Thank you

Scanned with ACE Scanner

DECLARATION

hereby declare that the internship report entitled "Programming Essentials In Python" submitted by me is an authentic record of my own work as requirement of internship training during the period from 15th May 10th june 2023 in partial fulfillment of the requirement forthe award of degree of Bachelor of Technology (Electronics & Communication Engineering), Netaji subhas institute of technology , Bihta Patna under the Supervision of Polytropic Services Pvt. Ltd.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university

Signature of the Student
Piyush KUMAR
(22104103902)

Scanned with ACE Scanner

Internship report on Python

Submitted in partial fulfillment of the requirements for

Bachelor of Engineering Degree in

Electronics & Communication Engineering

DEPARTMENT OF

ELECTRONICS & COMMUNICATION

Netaji Subhas Institute Of Technology, Bihta

PATNA – 801103, BIHAR.

Submitted By

Name - jayshankar thakur

Reg. No. - 21104103001

Branch – ECE

Sem / year - 3rd Sem /2nd year

Scanned with ACE Scanner

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

It is Affiliated to Bihar Engineering University (BEU) Amraha, Bihta ,

Patna –801103

ELECTRONICS & COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this internship Report isthe Bonafide work of jayshankar thakur (21104103001) who Underwent the internship in "Python" from 20th February to 22th March

Head of the Department (HOD)

RAJNI RANJAN

Signature

Scanned with ACE Scanner

Polytropic Services Pvt. Ltd.

(An ISO 9000-2015 & ISO 27001-2013 Certified ICT Company)

Ref #: INT/2303/114

Date: 25 Mar 2023

Internship Certificate

This is to certify that **Mr. JAYSANKAR THAKUR, Reg. No 21104103001**, student of **Netaji Subhas Institute of Technology (NSIT), Bihta, Patna**, ***Session 2021 - 2025*** of **Electronics & Communication Engineering** has undergone ***Four Week*** internship with our Organization in **Python** from ***20 Feb 2023 to 22 Mar 2023.*** During the internship duration, he has done a good job in his Project work and is found to be sincere,hardworking with a positive approach.

We wish **Mr. JAYSANKAR THAKUR** all the best in his future career endeavors.

For, *Polytropic Services Pvt Ltd.*

Polytropic Services Pvt. Ltd. Patna

Authorized Signatory

(Seal and Sign)

(**NOTE:** You may verify this certificate on - https://training.polytropicservices.com/)

Corporate Office: IT Startup Hub, Dept. of IT, Govt. of Bihar, 13th Floor, Biscomaun Tower, Gandhi Maidan, Patna.
Branch Office: H/No 29, Prayag Niwas, North SK Puri, Boring Road, Patna-800001.
Email Us: admin@polytropicservices.com, polytropicservices@gmail.com
Call Us: 9431051799, 9801946661, 9473010949 **Visit Us:** www.polytropicservices.com

Scanned with ACE Scanner

Acknowledgement

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

The seminar on **"Python"** was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the seminar. Their contributions and technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledgement.

Thank you

Scanned with ACE Scanner

Polytropic Services Pvt. Ltd.
(An ISO 9000-2015 & ISO 27001-2013 Certified ICT Company)

Ref #: INT/2303/116

Date: 25 Mar 2023

Internship Certificate

This is to certify that **Ms. TANYA KUMARI, Reg. No 22104103903,** student of **Netaji Subhas Institute of Technology (NSIT), Bihta, Patna,** ***Session 2021 - 2025*** of **Electronics & Communication Engineering** has undergone ***Four Week*** Internship with our Organization in **Python** from ***20 Feb 2023 to 22 Mar 2023.*** During the internship duration, she has done a good job in her Project work and is found to be sincere,hardworking with a positive approach.

We wish **Ms. TANYA KUMARI** all the best in her future career endeavors.

For, ***Polytropic Services Pvt Ltd.***

Polytropic Services Pvt. Ltd. Patna

Authorized Signatory
(Seal and Sign)

(**NOTE:** You may verify this certificate on - https://training.polytropicservices.com/)

Corporate Office: IT Startup Hub, Dept. of IT, Govt. of Bihar, 13th Floor, Biscomaun Tower, Gandhi Maidan, Patna.
Branch Office: H/No 29, Prayag Niwas, North SK Puri, Boring Road, Patna-800001.
Email Us: admin@polytropicservices.com, polytropicservices@gmail.com
Call Us: 9431051799, **9801946661**, 9473010949 **Visit Us:** www.polytropicservices.com

Scanned with ACE Scanner

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

NSIT

AN EFFECTIVE SOURCE OF TECHNOCRATS

NAME Tanya Kumari

SESSION 2021 - 25 SEM 3rd

COLLEGE ROLL NO 212064 BRANCH ECE

SUBJECT Calculus SUB CODE

CHECKED BY

Scanned with ACE Scanner

Acknowledgement

It is our proud privilege and duty to acknowledge the kind of help and guidance receiver from several people in preparation of this report. It would not have been possible to prepar this report in this form without their valuable help, cooperation and guidance.

First and foremost, we wish to record our sincere gratitude to Prof., **Mr** for his constant support and encouragement in preparation of this report and for making available library and laboratory facilities needed to prepare this report.

The seminar on **"Python"** was very helpful to us in giving the necessary backgrou information and inspiration in choosing this topic for the seminar. Their contributions a technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledgement.

Scanned with ACE Scanner

Polytropic Services Pvt. Ltd.

(An ISO 9000-2015 & ISO 27001-2013 Certified ICT Company)

Ref #: INT/2301/161

Date: 25 Mar 2023

Internship Certificate

This is to certify that **Mr. DIVAKAR KUMAR, Reg. No 21104103005,** student of **Netaji Subhas Institute of Technology, Bihta,** ***Session 2021 – 2025*** of **Electrical & Communication Engineering (ECE)** has undergone ***Four Weeks*** Internship with our Organization in **Python** from ***20 Feb 2023 to 22 March 2023.***

During the internship duration, he worked on internal projects for training purposes. He has done a good job in his Project work and is found to be sincere, hardworking with a positive approach.

We wish **Mr. DIVAKAR KUMAR** all the best in his future career endeavors.

For, ***Polytropic Services Pvt Ltd.***

Polytropic Services Pvt. Ltd. Patna

Authorized Signatory

(Seal and Sign)

(**NOTE:** You may verify this certificate on - https://training.polytropicservices.com/)

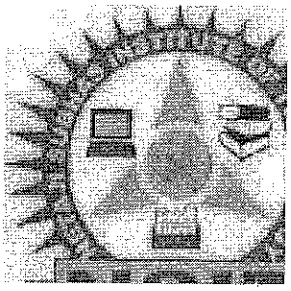
Scanned with ACE Scanner

Internship report on Python

Submitted in partial fulfillment of the requirements for

Bachelor of Engineering Degree in

Electronics & Communication Engineering



DEPARTMENT OF

ELECTRONICS & COMMUNICATION

Netaji Subhas Institute Of Technology, Bihta

PATNA – 801103. BIHAR.

Submitted By

Name - Prince Kumar

Reg. No. - 22104103901

Branch - ECE

Sem / year - 3rd Sem /2nd year

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY

It is Affiliated to Bihar Engineering University (BEU) Amraha, Bihta ,

Patna –801103

ELECTRONICS & COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this internship Report is the Bonafide work of **Prince kumar (22104103901)** who Underwent the internship in **"Python"** from 15th May to 10th june

Head of the Department (HOD)

RAJNI RANJAN

28/7/23

Signature

Polytropic Services Pvt. Ltd.

(An ISO 9000-2015 & ISO 27001-2013 Certified ICT Company)

Ref #: INT/2306/112 ***Date: 12 June 2023***

Internship Certificate

This is to certify that **Mr. PRINCE KUMAR, Reg. No 22104103901,** student of **Netaji Subhas Institute of Technology (NSIT), Bihta, Patna,** ***Session 2021 - 2025*** of **Electronics & Communication Engineering** has undergone ***Four Week*** Internship with our Organization in **Python** from ***15 May 2023 to 10 June 2023.***

During the internship duration, he has done a good job in his Project work and is found to be sincere,hardworking with a positive approach.

We wish **Mr. PRINCE KUMAR** all the best in his future career endeavors.

For, ***Polytropic Services Pvt Ltd.***

Polytropic Services Pvt. Ltd. Patna

Authorized Signatory

(Seal and Sign)

(**NOTE:** You may verify this certificate on - https://training.polytropicservices.com/)

Corporate Office: IT Startup Hub, Dept. of IT, Govt. of Bihar, 13th Floor, Biscomaun Tower, Gandhi Maidan, Patna.
Branch Office: H/No 29, Prayag Niwas, North SK Puri, Boring Road, Patna-800001.
Email Us: admin@polytropicservices.com, polytropicservices@gmail.com
Call Us: 9431051799, 9801946661, 9473010949 **Visit Us:** www.polytropicservices.com

Acknowledgement

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

The seminar on **"Python"** was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the seminar. Their contributions and technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledgement.

Thank you

DECLARATION

I hereby declare that the internship report entitled "Programming Essentials In Python" submitted by me is an authentic record of my own work as requirement of internship training during the period from 15th May 10th june 2023 in partial fulfillment of the requirement forthe award of degree of Bachelor of Technology (Electronics & Communication Engineering), Netaji subhas institute of technology , Bihta Patna under the Supervision of Polytropic Services Pvt. Ltd.

I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university

Signature of the Student
PRINCE KUMAR
(22104103901)

Table Of Contents

Introduction

1.1 Python 1-2

1.2 Scripting Language 2-3

1.3 Object Oriented Programming 3-3

1.4 History of python 3-4

1.5 Behind the Scene of Python 5-5

Downloading & Installing Python

2.1 Downloading Python 6-7

2.2 Installing Python 8-10

2.3 Setup path of variable 10-11

2.4 Running The Python IDE 12-13

2.5 Python code Execution 13-13

Data Types & Operator

3.1 Data Type 14-15

3.2 Variables 15-15

3.3 String 15-16

3.4 Python Operator 16-17

3.4.1 Arithmetic Operator 17-17

3.4.2 Comparison Operator 17-18

Tuple & List

4.1 Tuple 18-20

4.1.1 Accessing Tuple Values 18-18

4.1.2 Built in Operation 18-19

4.1.3 Built in Touple Functions 19-19

4.2 List 20-23

4.2.1 Accessing List Values 20-21

4.2.2 Built in Operation 21-21

4.2.3 Built in Functions 21-21

Loops & Conditional Statements

5.1 Loops 21-26

5.1.1 Loops Definition 21-22

5.1.2 Loops Example 22-23

5.2 Conditional Statement 23-23

5.21Conditional Statement Definition 23-23

5.2.2 Conditional Statement Example 24-25

5.3 Function 25-25

5.3.1 Syntax & Examples 25-26

Uses & Scope of python.

6.1 What can we do With Python? 26-26

6.2 Who Uses Python Today? 27-27

6.3 Why do People use python? 27-27

Python

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems.

Scripting Language

A scripting or script language is a programming language that supports scripts, programs written for a special run-time environment that automate the execution of tasks that could alternatively be executed one-by-one by a human operator.

Scripting languages are often interpreted (rather than compiled). Primitives are usually the elementary tasks or API calls, and the language allows them to be combined into more complex programs. Environments that can be automated through scripting include software applications, web pages within a web browser, the shells of operating systems (OS), embedded systems, as well as numerous games.

A scripting language can be viewed as a domain-specific language for a particular environment; in the case of scripting an application, this is also known as an **extension language**. Scripting languages are also sometimes referred to as very high-level programming languages, as they operate at a high level of abstraction, or as control languages.

Object Oriented Programming Language

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A distinguishing feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self").

In OO programming, computer programs are designed by making them out of objects that interact with one another. There is significant diversity in objectoriented programming, but most popular languages are class-based, meaning that objects are instances of classes, which typically also determines their type.

History

Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, benevolent dictator for life (BDFL).

"Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered."

- Guido van Rossum

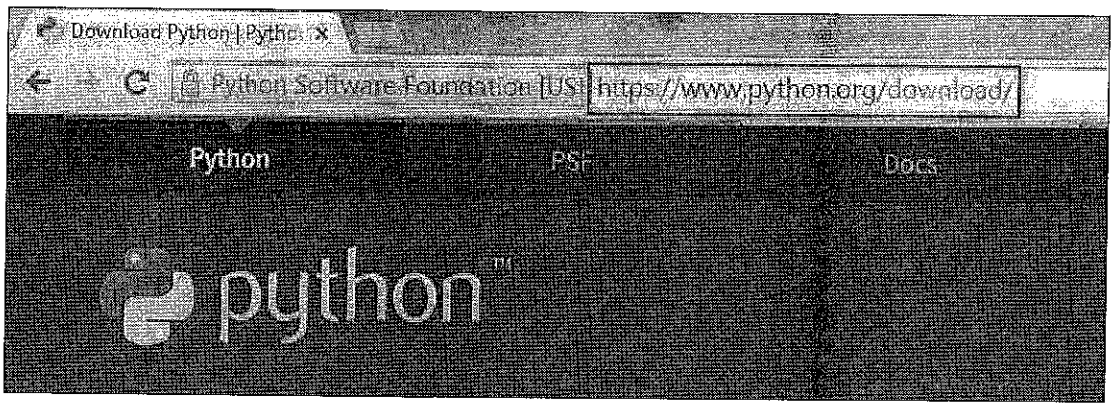
Behind The Scene of Python

About the origin of Python, Van Rossum wrote in 1996:

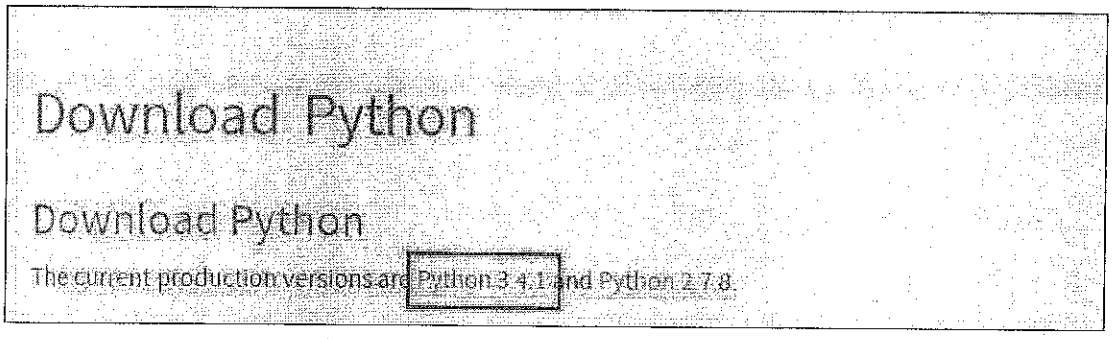
Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home Computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a **descendant of ABC** that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of **Monty Python's Flying Circus**).

Downloading python

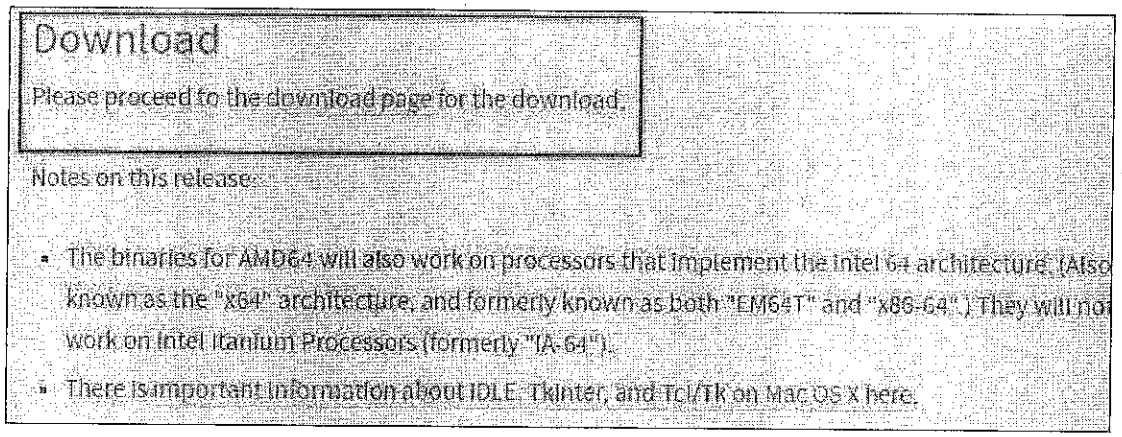
If you don't already have a copy of Python installed on your computer, you will need to open up your Internet browser and go to the Python download page ***(http://www.python.org/download/).***



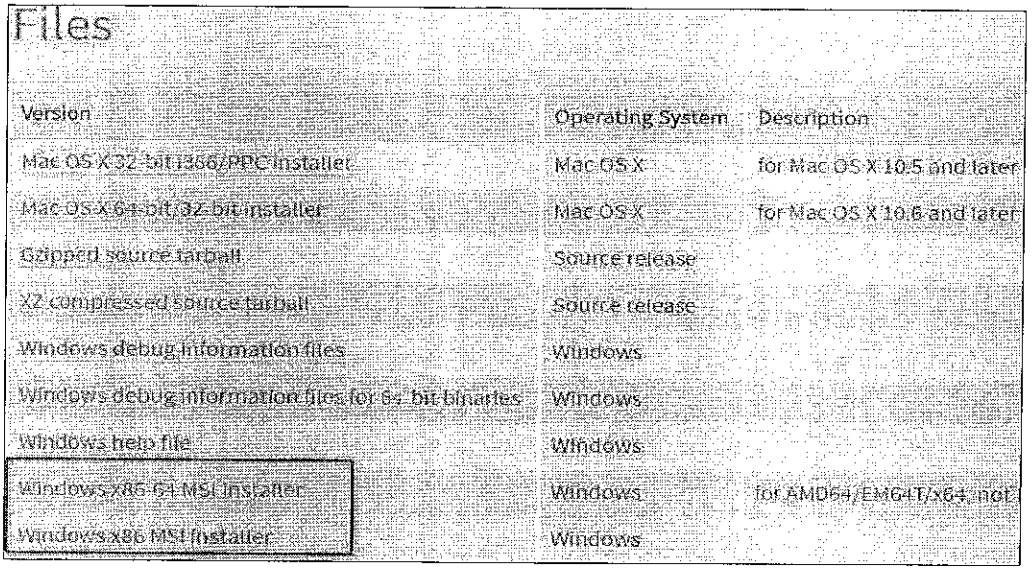
Now that you are on the download page, select which of the software builds you would like to download. For the purposes of this article we will use the most up to date version available (Python 3.4.1).



Once you have clicked on that, you will be taken to a page with a description of all the new updates and features of 3.4.1, however, you can always read that while the download is in process. Scroll to the bottom of the page till you find the "Download" section and click on the link that says "download page."



Now you will scroll all the way to the bottom of the page and find the "Windows x86 MSI installer." If you want to download the 86-64 bit MSI, feel free to do so. We believe that even if you have a 64-bit operating system installed on your computer, the 86-bit MSI is preferable. We say this because it will still run well and sometimes, with the 64- bit architectures, some of the compiled binaries and Python libraries don't work well.

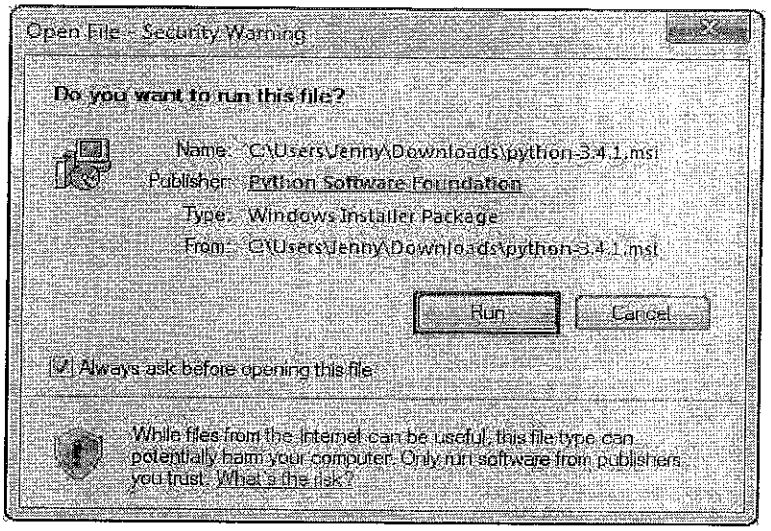


Files

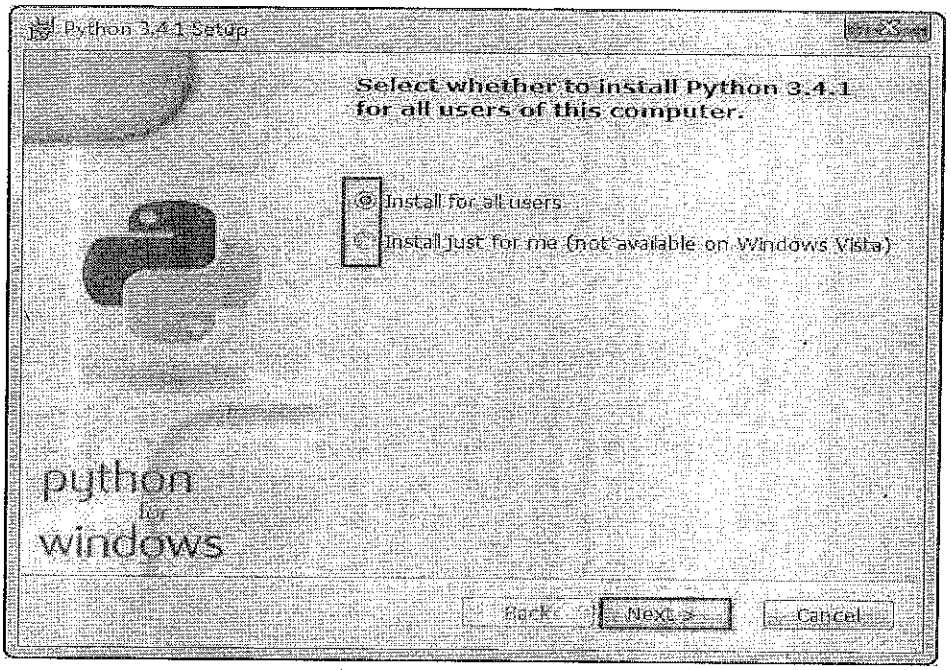
Version	Operating System	Description
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
Windows debug information files	Windows	
Windows debug information files for 64-bit binaries	Windows	
Windows help file	Windows	
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64, not i
Windows x86 MSI installer	Windows	

Installing Python

Once you have downloaded the Python MSI, simply navigate to the download location on your computer, double clicking the file and pressing Run when the dialog box pops up.

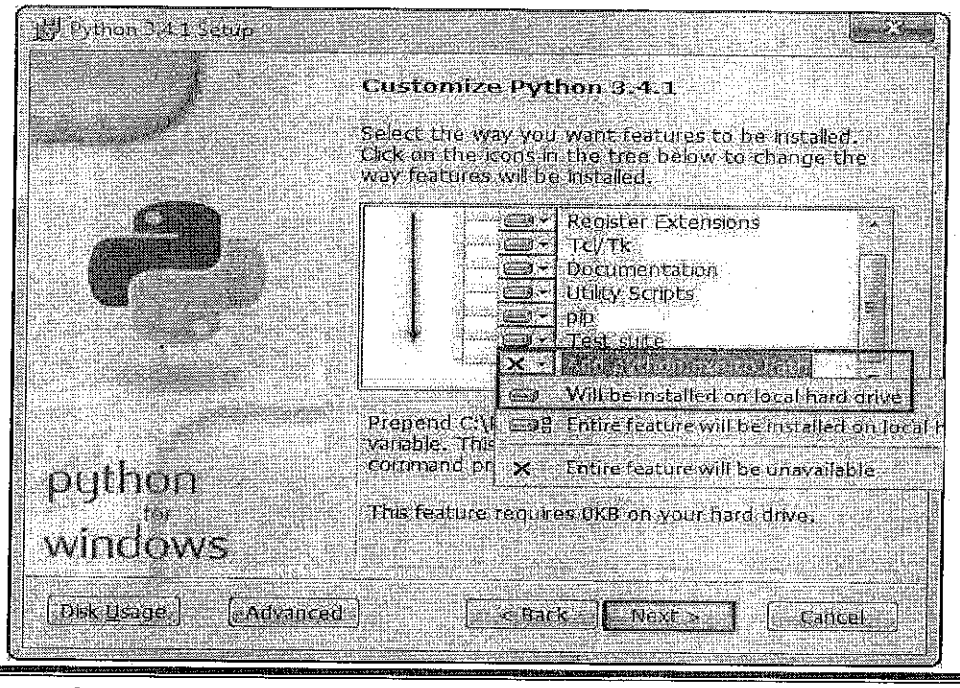


If you are the only person who uses your computer, simply leave the "Install for all users" option selected. If you have multiple accounts on your PC and don't want to install it across all accounts, select the "Install just for me" option then press "Next."

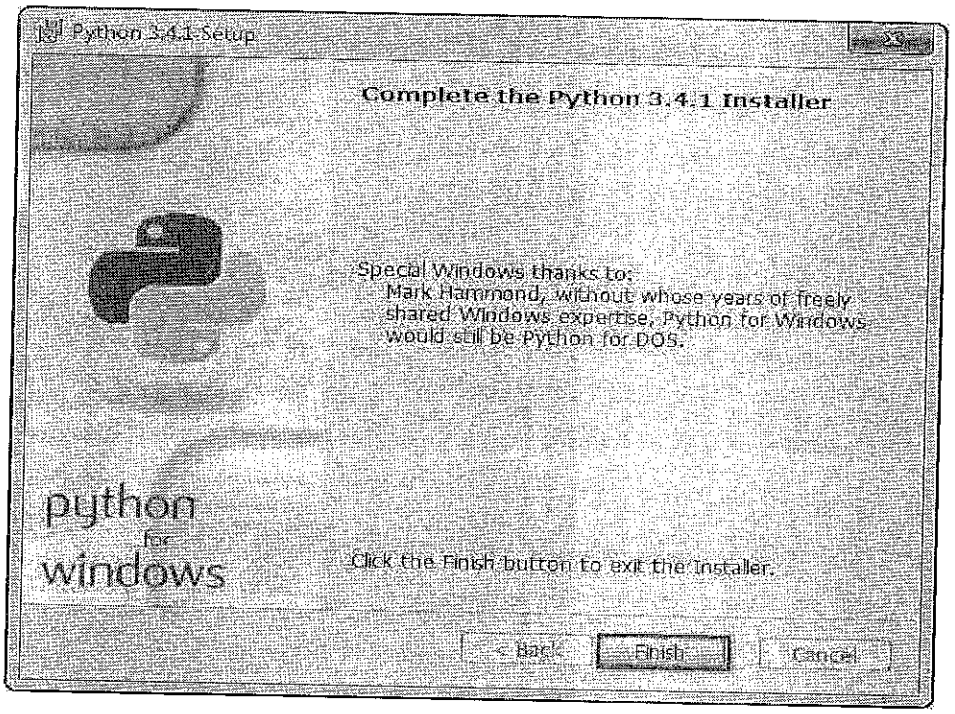


f you want to change the install location, feel free to do so; however, it is best to leave it as is and simply select next, Otherwise...

Scroll down in the window and find the "Add Python.exe to Path" and click on the small red "x." Choose the "Will be installed on local hard drive" option then press "Next."



Now that you have completed



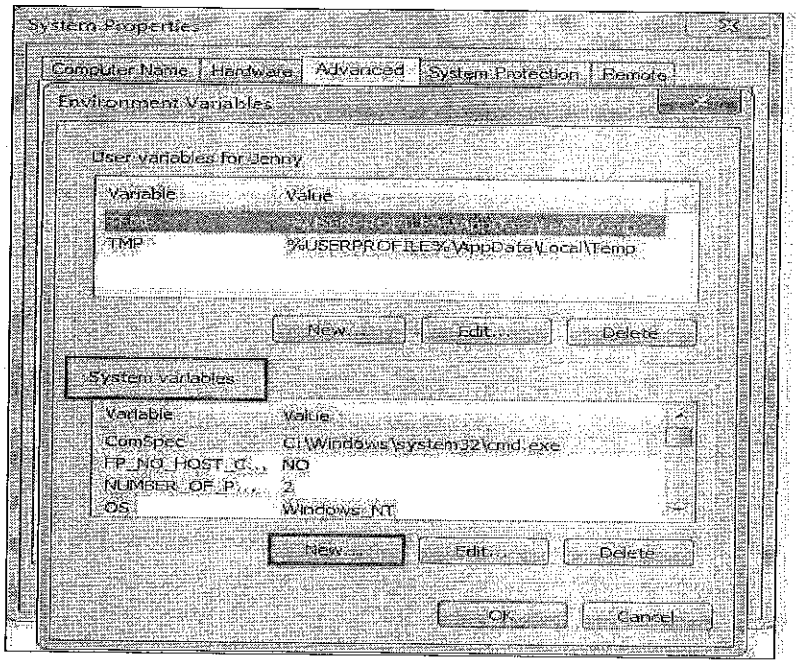
the installation process, click on "Finish.

Setup the Path Variable

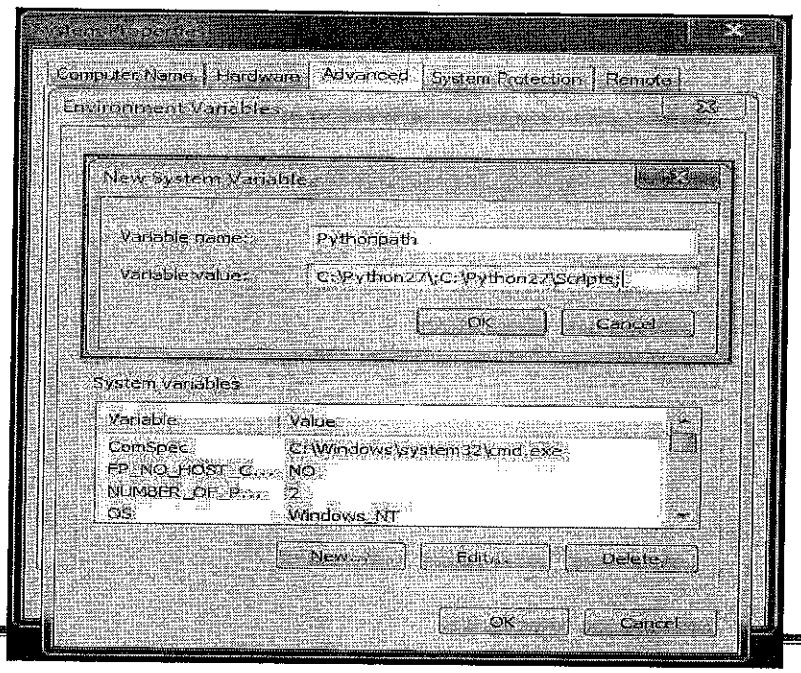
Begin by opening the start menu and typing in "environment" and select the option called "Edit the system environment variables."

When the "System Properties" window appears, click on "Environment Variables..."

Once you have the "Environment Variables" window open, direct your focus to the bottom half. You will notice that it controls all the "System Variables" rather than just this associated with your user. Click on "New..." to create a new variable for Python.

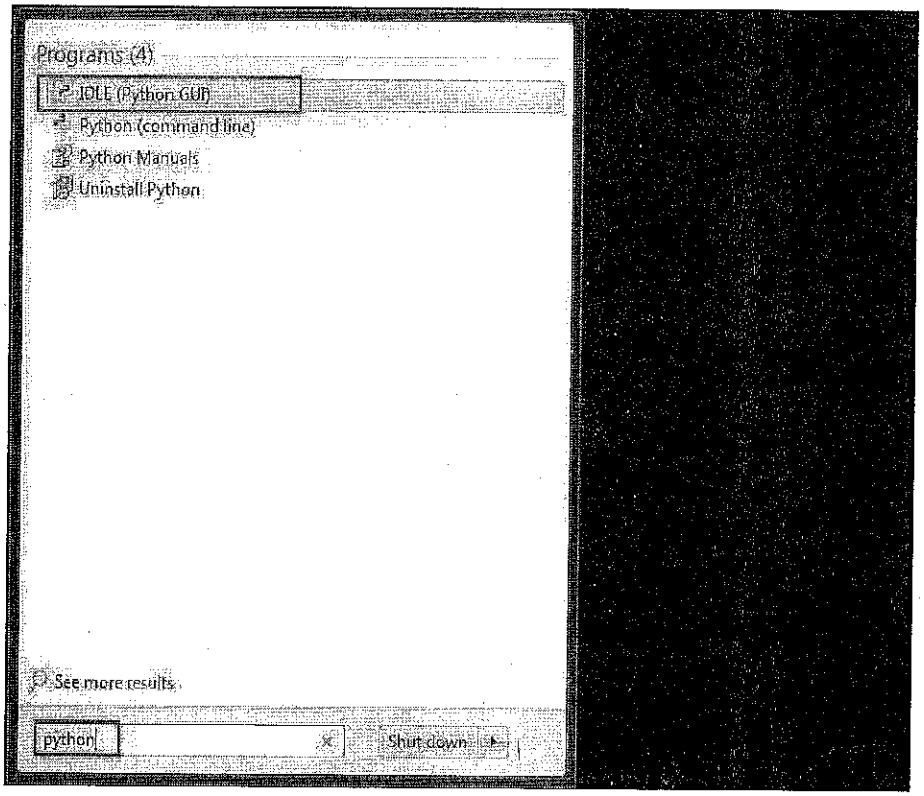


Simply enter a name for your Path and the code shown below. For the purposes of this example we have installed Python 2.7.3, so we will call the path: "Pythonpath." The string that you will need to enter is: "C:\Python27\;C:\Python27\Scripts;"



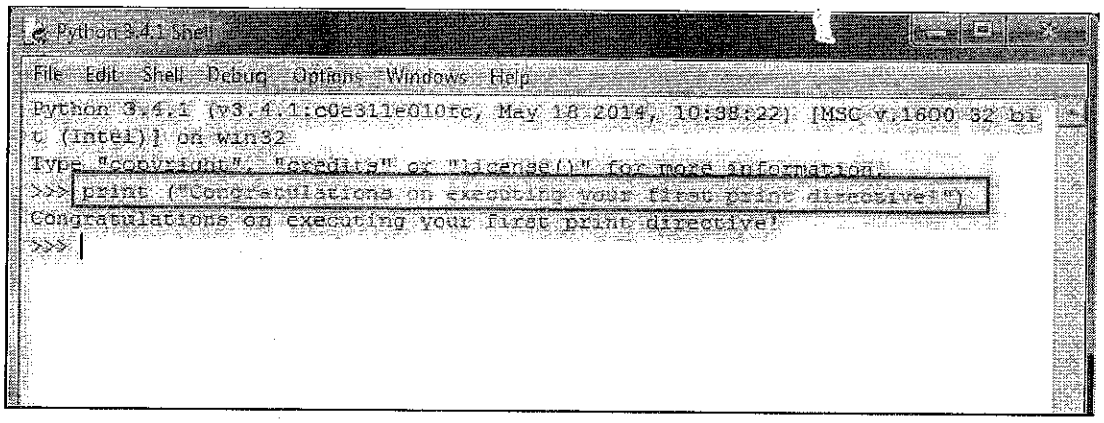
Running The Python IDE

Now that we have successfully completed the installation process and added our "Environment Variable," you are ready to create your first basic Python script. Let's begin by opening Python's GUI by pressing "Start" and typing "Python" and selecting the "IDLE (Python GUI)."



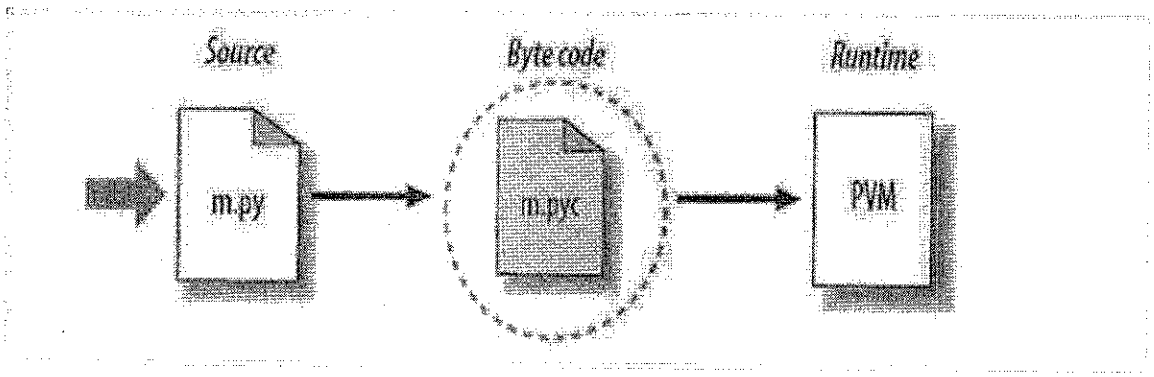
Once the GUI is open, we will begin by using the simplest directive possible. This is the "print" directive which simply prints whatever you tell it to, into a new line. Start by typing a print directive like the one shown in the image below or copy and paste this text then press

"Enter": print ("Congratulations on executing your first print directive!")



Python Code Execution

Python's traditional runtime execution model: source code you type is translated to byte code, which is then run by the Python Virtual Machine. Your code is automatically compiled, but then it is interpreted.



Source code extension is .py

Byte code extension is .pyc (compiled python code)

Data Type

(this is called dynamic typing). Data types determine whether an object can do something, or whether it just would not make sense. Other programming languages often determine whether an operation makes sense for an object by making sure the object can never be stored somewhere where the operation will be performed on the object (this type system is called static typing). Python does not do that. Instead it stores the type of an object with the object, and checks when the operation is performed whether that operation makes sense for that object

Python has many native data types. Here are the important ones:

Booleans are either True or False.

Numbers can be integers (1 and 2), floats (1.1 and 1.2), fractions (1/2 and 2/3), or even complex numbers.

Strings are sequences of Unicode characters, e.g. an HTML document.

Bytes and byte arrays, e.g. a JPEG image file.

Lists are ordered sequences of values.

Tuples are ordered, immutable sequences of values.

Sets are unordered bags of values.

Variable

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Ex: counter = 100	# An integer
assignment miles = 1000.0	# A floating
point name = "John"	# A string

String

In programming terms, we usually call text a string. When you think of a string as a collection of letters, the term makes sense.

All the letters, numbers, and symbols in this book could be a

string. For that matter, your name could be a string, and so could

your address.

Creating Strings

In Python, we create a string by putting quotes around text. For example, we could take our otherwise useless

- "hello"+"world" "helloworld" # concatenation
- "hello"*3 "hellohellohello" # repetition
- "hello"[0] "h" # indexing
- "hello"[-1] "o" # (from end)
- "hello"[1:4] "ell" # slicing
- len("hello") 5 # size
- "hello" < "jello" 1 # comparison
- "e" in "hello" 1 # search

Python Operator

Arithmetic Operator

Operator	Meaning	Example
+	Add two operands or unary plus	x + y +2
-	Subtract right operand from the left or unary minus	x - y -2
*	Multiply two operands	x * y
/	Divide left operand by the right one (always results into float)	x / y

%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Comparison Operator

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Tuples

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses.

Accessing Values in Tuples:

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example – tup1 = ('physics', 'chemistry', 1997, 2000); tup2 = (1, 2, 3, 4, 5, 6, 7); print "tup1[0]: ", tup1[0] print "tup2[1:5]: ", tup2[1:5]

When the above code is executed, it produces the following result – tup1[0]: physics tup2[1:5]: [2, 3, 4, 5]

Basic Tuples Operations.

Tuples respond to the + and * operators much like strings; they mean concatenation and repetition here too, except that the result is a new tuple, not a string. In fact, tuples respond to all of the general sequence operations we used on strings in the prior chapter -

Python Expression	Results	Description
len((1, 2, 3))	3	Length
(1, 2, 3) + (4, 5, 6)	(1, 2, 3, 4, 5, 6)	Concatenation
('Hi!',) * 4	('Hi!', 'Hi!', 'Hi!', 'Hi!')	Repetition
3 in (1, 2, 3)	True	Membership

for x in (1, 2, 3): print x,	1 2 3	Iteration

Built-in Tuple Functions

Python includes the following tuple functions –

SN	Function with Description
1	**cmp(tuple1, tuple2)** Compares elements of both tuples.
2	**len(tuple)** Gives the total length of the tuple.
3	**max(tuple)** Returns item from the tuple with max value.
4	**min(tuple)** Returns item from the tuple with min value.
5	**tuple(seq)** Converts a list into tuple.

List

The list is a most versatile datatype available in Python which can be written as a list of comma- separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets.

For example -

list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1,2, 3, 4, 5]; list3 = ["a", "b", "c", "d"];

Similar to string indices, list indices start at 0, and lists can be sliced, concatenated and so on.

Accessing Values in Lists:

To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example – list1 = ['physics', 'chemistry', 1997, 2000]; list2 = [1, 2, 3, 4, 5, 6, 7]; print "list1[0]: ", list1[0] print "list2[1:5]: ", list2[1:5]

Output:

list1[0]: physics

list2[1:5]: [2, 3, 4, 5]

Update:

list = ['physics', 'chemistry', 1997, 2000]; print "Value available at index 2 : " print list[2] list[2] = 2001; print "New value available at index 2 : " print list[2]

Output:

Value available at index 2 : 1997

New value available at index 2 : 2001

Delete: list1 = ['physics', 'chemistry', 1997, 2000]; print

list1 del list1[2]; print "After deleting value at index 2 : " print

list1

['physics', 'chemistry', 1997, 2000]

Output: After deleting value at index 2 :

['physics', 'chemistry', 2000]

Basic List Operation

Python Expression	Results	Description
len([1, 2, 3])	3	Length
[1, 2, 3] + [4, 5, 6]	[1, 2, 3, 4, 5, 6]	Concatenation
['Hi!'] * 4	['Hi!', 'Hi!', 'Hi!', 'Hi!']	Repetition
3 in [1, 2, 3]	True	Membership
for x in [1, 2, 3]: print x,	1 2 3	Iteration

Built-in List Functions & Methods:

SN	Function with Description
1	cmp(list1, list2) Compares elements of both lists.
2	len(list) Gives the total length of the list.
3	max(list) Returns item from the list with max value.
4	min(list) Returns item from the list with min value.
5	list(seq) Converts a tuple into list.

Python includes following list methods

SN	Methods with Description
1	list.append(obj) Appends object obj to list
2	list.count(obj) Returns count of how many times obj occurs in list
3	list.extend(seq) Appends the contents of seq to list
4	list.index(obj) Returns the lowest index in list that obj appears

5	**list.insert(index, obj)** Inserts object obj into list at offset index
6	**list.pop(obj=list[-1])** Removes and returns last object or obj from list
7	**list.remove(obj)** Removes object obj from list
8	**list.reverse()** Reverses objects of list in place
9	**list.sort([func])** Sorts objects of list, use compare func if given

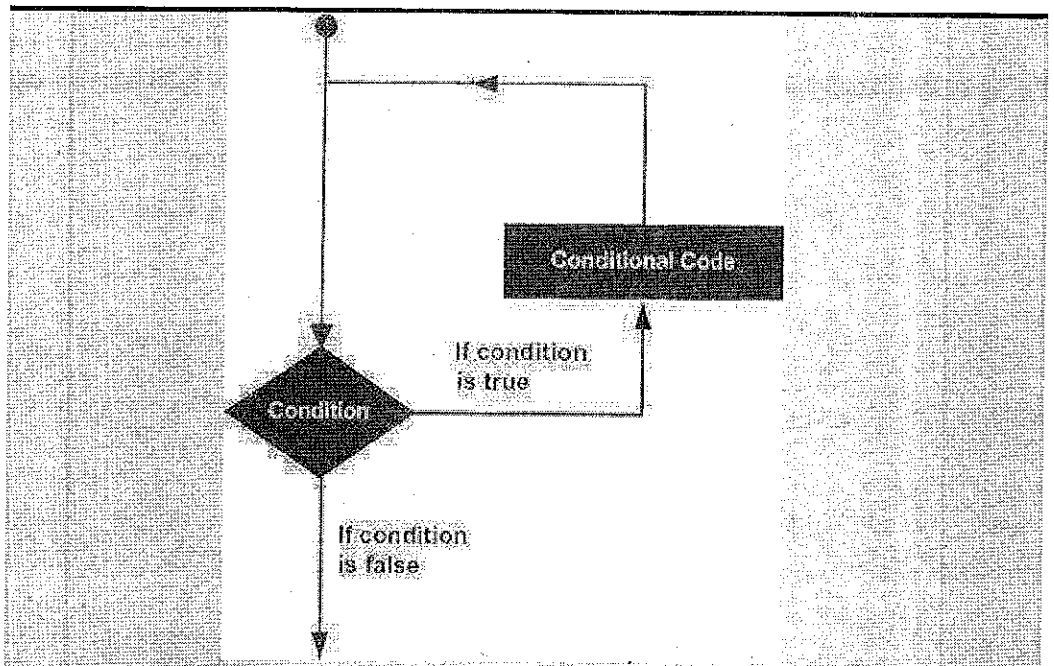
Loop definition

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times.

Thefollowing diagram illustrates a loop statement –

!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to - True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to - True if left operand is less than or equal to the right	+x <= y



>	Greater that - True if left operand is greater than the right	x > y
<	Less that - True if left operand is less than the right	x < y
==	Equal to - True if both operands are equal	x == y

Python programming language provides following types of loops to handle looping requirements.

Loop Type	Description
while loop	Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing theloop body.
for loop	Executes a sequence of statements multiple times andabbreviates the code that manages the loop variable.
nested loops	You can use one or more loop inside any another while, for or do..while loop.

Loop Example:

For Loop:

```
>>> for mynum in [1, 2, 3, 4, 5]:

print ("Hello", mynum )

Hello 1
Hello 2
Hello 3
Hello 4
Hello 5
```

While Loop:

>>> count = 0 >>while(count< 4):

print 'The count is:', count count = count + 1

The count is: 0

The count is: 1

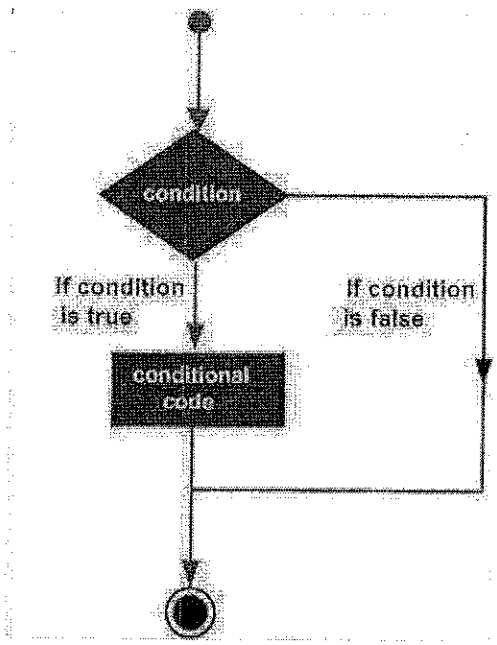
The count is: 2

The count is: 3

Conditional Statements:

Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions.

Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome is TRUE or FALSE otherwise.



Python programming language provides following types of decision making statements. Click the following links to check their detail.

Statement	Description
if statements	An **if statement** consists of a boolean expression followed by one or more statements.
if...else statements	An **if statement** can be followed by an optional **else statement**, which executes when the boolean expression is FALSE.
nested if statements	You can use one **if** or **else if** statement inside another **if** or **else if** statement(s).

Example:

If Statement:

```
a=33
b=200
If b>a:
print("b")
```

If...Else Statement:

```
a=200
b=33
if b>a:
    print("b is greater than a")
else:
  print("a is greater than b")
```

Function

Function blocks begin with the keyword **def** followed by the function name and parentheses (( )).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
The first statement of a function can be an optional statement - the documentation string of the function.

The code block within every function starts with a colon (:) and is indented.

The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Syntex:

Def functionname(parameters):

"function_docstring"

Function_suite

Return[expression]

Example:

Def printme(str):
"this print a passed string into this
function" print str
return

1. # *Function definition is here* def
   printme(str):

   "This prints a passed string into this function" print
   str return;

   # *Now you can call printme function* printme("I'm first call to user defined function!") printme("Again second call to the same function")

SCOPE OF PYTHON

1 - Science

- Bioinformatics

2 - System Administration

- Unix
- Web logic
- Web sphere

3 - Web Application Development

What Can We do With Python?

1 - System programming

2 - Graphical User Interface

Programming 3 - Internet Scripting

4 - Component Integration

5 - Database Programming

6 - Gaming, Images, XML , Robot and more

WHO USES PYTHON TODAY?

- Python is being applied in real revenue-generating products by real companies.
- Google makes extensive use of Python in its web search system, and employs Python's creator.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.
- ESRI uses Python as an end-user customization tool for its popular GIS mapping products.

WHY DO PEOPLE USE PYTHON?

- The YouTube video sharing service is largely written in Python.
- Python is object-oriented ○ Structure supports such concepts as polymorphism, operation overloading, and multiple inheritance.
- Indentation ○ Indentation is one of the greatest future in Python.
- It's free (open source) ○ Downloading and installing Python is free and easy ○ Source code is easily accessible
- It's powerful ○ Dynamic typing ○ Built-in types and tools ○ Library utilities
  - ○ Third party utilities (e.g. Numeric, NumPy, SciPy) ○ Automatic memory management
- It's portable ○ Python runs virtually every major platform used today ○ As long as you have a compatible Python interpreter installed, Python programs will run in exactly the same manner, irrespective of platform.

Conclusion

I believe the trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language:

- It is Free (as in both cost and source code).
- It is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated;
- It is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills;
- It is a real-world programming language that can be and is used in academia and the commercial world;
- It appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied;

and most importantly, its clean syntax offers increased understanding and enjoyment for students

Netaji Subhas Institute Of Technology, Bihta, Patna.

Approved by AICTE, New Delhi, Affiliated to Aryabhatta Knowledge University,
Patna (Bihar)

"Internship Report"

Submitted by:
Ritika Jha
214009

Scanned with ACE Scanner

Feb 1st, 2023

TO WHOM IT MAY CONCERN,

This is to certify that Ms Ritika Jha, D/O Mr Shishir Kumar Jha, a Electronic and Communication student from NSIT, has completed **Intern – Software Development.** The duration of the internship was from **15th December 2022 to 15th Jan 2023.**

Her role was to familiarize herself with all the processes at the organization. She spent a significant amount of time in each department and learnt in-depth what the flow of work is like. Ms Jha is a diligent individual and particularly inquisitive.

We hope her time spent at Pitbit.ai was fruitful. We wish her all the success in her future.

Congratulations!

Regards,

Pibit.ai

Scanned with ACE Scanner

Feb 1st, 2023

TO WHOM IT MAY CONCERN,

This is to certify that Mr **Rishabh Kumar**, D/O Mr Shashank Kumar, an Electronic and Communication student from Netaji Subhas Institute of Technology **(NSIT) Patna**, has completed **Intern – Software Development**. The duration of the internship was from **14th December 2022 to 13th Jan 2023**.

His role was to familiarize himself with all the processes at the organization. He spent a significant amount of time in each department and learnt in-depth what the flow of work is like. Mr. **Rishabh Kumar** is a diligent individual and particularly inquisitive. He helped in developing FastAPI for the product he was assigned to.

We hope his time spent at **Swiftstatsai.com** was fruitful. We wish him all the success in his future.

Congratulations!

Regards,

Swiftstatsai.com

DECLARATION

I hereby declare that this submission is own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extend has been accepted for the award of award of any other degree of the university or other institute of the higher learning except where due acknowledgement has been made in the text.

Name : Rishabh Kumar
Roll No: 212005

Signature: .....
Date: .....

Netaji Subhas Institute Of Technology, Bihta, Patna.

Approved by AICTE, New Delhi, Affiliated to Aryabhatta Knowledge University,
Patna (Bihar)

"Internship Report"

Submitted by:
Rishabh Kumar
212005

Feb 1st, 2023

TO WHOM IT MAY CONCERN,

This is to certify that Mr **Rishabh Kumar**, D/O Mr Shashank Kumar, an Electronic and Communication student from Netaji Subhas Institute of Technology **(NSIT) Patna**, has completed **Intern – Software Development**. The duration of the internship was from **14th December 2022 to 13th Jan 2023**.

His role was to familiarize himself with all the processes at the organization. He spent a significant amount of time in each department and learnt in-depth what the flow of work is like. Mr. **Rishabh Kumar** is a diligent individual and particularly inquisitive. He helped in developing FastAPI for the product he was assigned to.

We hope his time spent at **Swiftstatsai.com** was fruitful. We wish him all the success in his future.

Congratulations!

Regards,

Swiftstatsai.com

Contents

1. Introduction
2. Drawing a line
3. Types of linear regression
4. Creating a linear regression
5. What is the best fit line?
6. Cost function for linear regression
7. Gradient descent for linear regression
8. Coefficient of Determination or (R-2) R-squared
9. Root mean squared error
10. Assumptions of Linear Regression
11. Hypothesis in linear regression
12. Assessing the model fit

Introduction

Linear regression is a statistical method used to model the relationship between two or more variables. It is a widely used technique in machine learning and data science for predicting a continuous output variable based on one or more input variables.

In linear regression, a linear relationship is assumed between the input variables and the output variable. The model tries to fit a straight line through the data to make predictions. The line is defined by the intercept and the slope of the line.

The basic form of linear regression is expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_{\square} x_{\square} + \varepsilon$$

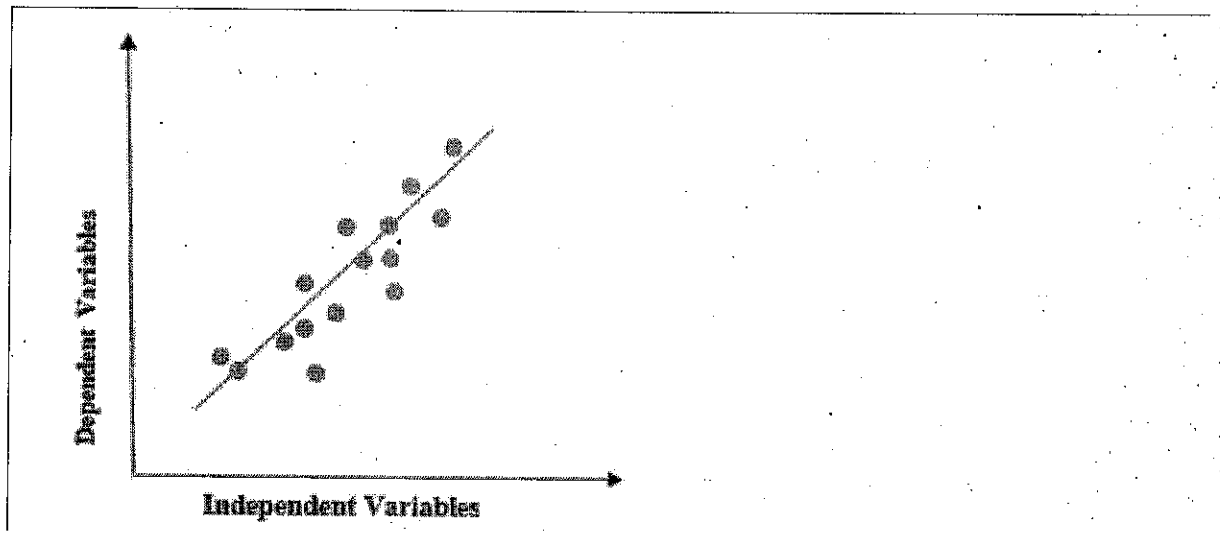
where:

- y is the output or response variable
- β_0 is the intercept
- $\beta_1, \beta_2, \ldots, \beta_{\square}$ are the coefficients or weights of the input variables $x_1, x_2, \ldots, x_{\square}$
- ε is the error or residual term that accounts for unexplained variability in the data

The goal of linear regression is to find the values of the coefficients that minimize the sum of the squared errors between the predicted values and the actual values in the training data.

Linear regression can be used for both simple linear regression (one input variable) and multiple linear regression (multiple input variables). It can also be extended to include more complex relationships between the input and output variables, such as polynomial regression or regression with interaction terms.

Drawing a line/Slope intercept



Types of linear Regression

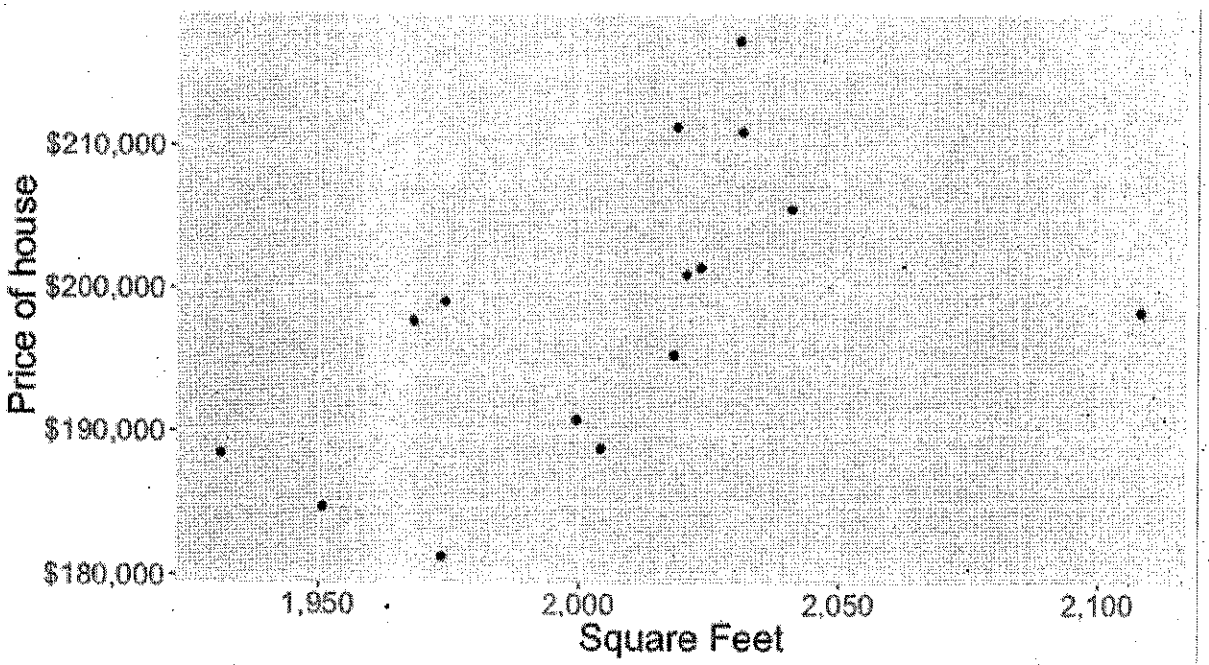
Linear regression is a statistical method used to model the relationship between two variables by fitting a linear equation to the observed data. There are several types of linear regression, including:

1. Simple Linear Regression: This is the most basic form of linear regression that involves two variables, one independent variable and one dependent variable. The goal is to find the best fitting straight line through the data that minimizes the sum of the squared residuals.
2. Multiple Linear Regression: This type of regression involves more than one independent variable and one dependent variable. The goal is to find the best fitting linear equation that explains the relationship between the dependent variable and multiple independent variables.

3. Polynomial Regression: This is a type of regression where the relationship between the independent variable and dependent variable is modeled as an nth degree polynomial. This type of regression is useful when the relationship between the variables is not linear.

Creating a linear regression

Linear regression is a quiet and the simplest statistical regression method used for predictive analysis in machine learning. Linear regression shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear regression. If there is a single input variable X(independent variable), such linear regression is called simple linear regression.



The above graph presents the linear relationship between the output(y) variable and predictor(X) variables, which are price of houses vs square feet of those houses.

To calculate best-fit line linear regression uses a traditional slope-intercept form which is given below -

$$Yi = \beta 0 + \beta 1 Xi$$

where Yi = Dependent variable, $\beta 0$ = constant/Intercept, $\beta 1$ = Slope/Intercept, Xi = Independent variable.

This algorithm explains the linear relationship between the dependent(output) variable y and the independent(predictor) variable X using a straight line Y= B0 + B1 X.

But how the linear regression finds out which is the best fit line?

The goal of the linear regression algorithm is to get the best values for B0 and B1 to find the best fit line. The best fit line is a line that has the least error which means the error between predicted values and actual values should be minimum.

Random Error(Residuals)

In regression, the difference between the observed value of the dependent variable(yi) and the predicted value(predicted) is called the residuals.

εi = y predicted – yi

where y predicted = B0 + B1 Xi

What is the best fit line?

In simple terms, the best fit line is a line that fits the given scatter plot in the best way. Mathematically, the best fit line is obtained by minimizing the Residual Sum of Squares(RSS).

Cost Function for Linear Regression

The cost function helps to work out the optimal values for B0 and B1, which provides the best fit line for the data points.

In Linear Regression, generally Mean Squared Error (MSE) cost function is used, which is the average of squared error that occurred between the y predicted and yi.

We calculate MSE using simple linear equation y=mx+b:

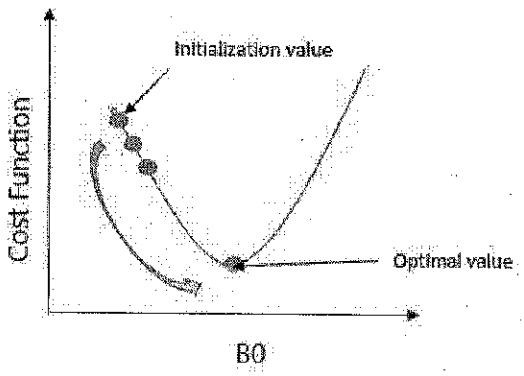
$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (B1x_i + B0))^2$$

Using the MSE function, we'll update the values of B0 and B1 such that the MSE value settles at the minima. These parameters can be determined using the gradient descent method such that the value for the cost function is minimum.

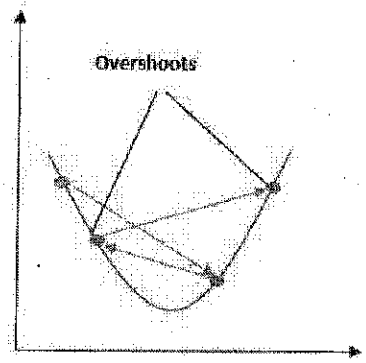
Gradient Descent for Linear Regression

Gradient Descent is one of the optimization algorithms that optimize the cost function(objective function) to reach the optimal minimal solution. To find the optimum solution we need to reduce the cost function(MSE) for all data points. This is done by updating the values of B0 and B1 iteratively until we get an optimal solution.

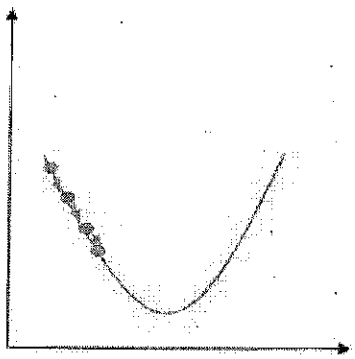
A regression model optimizes the gradient descent algorithm to update the coefficients of the line by reducing the cost function by randomly selecting coefficient values and then iteratively updating the values to reach the minimum cost function.



Let's take an example to understand this. Imagine a U-shaped pit. And you are standing at the uppermost point in the pit, and your motive is to reach the bottom of the pit. Suppose there is a treasure at the bottom of the pit, and you can only take a discrete number of steps to reach the bottom. If you opted to take one step at a time, you would get to the bottom of the pit in the end but, this would take a longer time. If you decide to take larger steps each time, you may achieve the bottom sooner but, there's a probability that you could overshoot the bottom of the pit and not even near the bottom. In the gradient descent algorithm, the number of steps you're taking can be considered as the learning rate, and this decides how fast the algorithm converges to the minima.



High learning rate



Low learning rate

To update B0 and B1, we take gradients from the cost function. To find these gradients, we take partial derivatives for B0 and B1.

$$J = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial \mathrm{B}_0} = \frac{2}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial \mathrm{B}_1} = \frac{2}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i) \cdot x_i$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial \mathrm{B}_0} = \frac{2}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial \mathrm{B}_1} = \frac{2}{n}\sum_{i=1}^{n}(\mathrm{B}_0 + \mathrm{B}_1 \cdot x_i - y_i) \cdot x_i$$

$$\mathrm{B}_0 = \mathrm{B}_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$\mathrm{B}_1 = \mathrm{B}_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

Evaluation Metrics for Linear Regression

The strength of any linear regression model can be assessed using various evaluation metrics. These evaluation metrics usually provide a measure of how well the observed outputs are being generated by the model.

The most used metrics are-

- Coefficient of Determination or R-Squared (R2)
- Root Mean Squared Error (RSME) and Residual Standard Error (RSE)

Coefficient of Determination or R-Squared (R2)

R-Squared is a number that explains the amount of variation that is explained/captured by the developed model. It always ranges between 0 & 1 . Overall, the higher the value of R-squared, the better the model fits the data.

Mathematically it can be represented as,

$$R2 = 1 - (RSS/TSS)$$

Residual sum of Squares (RSS) is defined as the sum of squares of the residual for each data point in the plot/data. It is the measure of the difference between the expected and the actual observed output.

Root Mean Squared Error

The Root Mean Squared Error is the square root of the variance of the residuals. It specifies the absolute fit of the model to the data i.e. how close the observed data points are to the predicted values. Mathematically it can be represented as,

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\sum_{i=1}^{n}\left(y_i^{Actual} - y_i^{Predicted}\right)^2 / n}$$

To make this estimate unbiased, one has to divide the sum of the squared residuals by the degrees of freedom rather than the total number of data points in the model. This term is then called the Residual Standard Error(RSE). Mathematically it can be represented as,

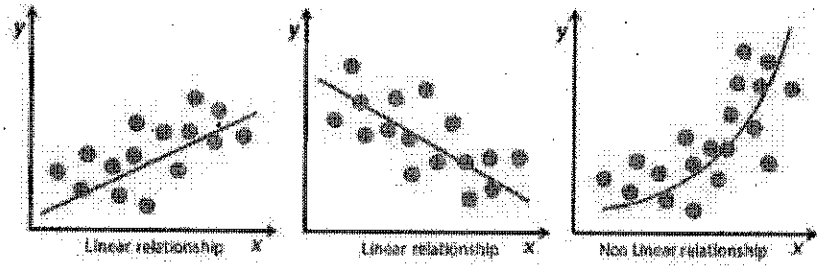
$$RSE = \sqrt{\frac{RSS}{df}} = \sqrt{\sum_{i=1}^{n}\left(y_i^{Actual} - y_i^{Predicted}\right)^2 / (n-2)}$$

R-squared is a better measure than RSME. Because the value of Root Mean Squared Error depends on the units of the variables (i.e. it is not a normalized measure), it can change with the change in the unit of the variables.

Assumptions of Linear Regression

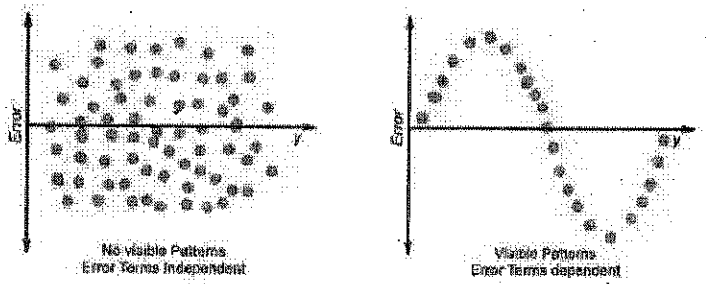
Regression is a parametric approach, which means that it makes assumptions about the data for the purpose of analysis. For successful regression analysis, it's essential to validate the following assumptions.

1. Linearity of residuals: There needs to be a linear relationship between the dependent variable and independent variable(s).



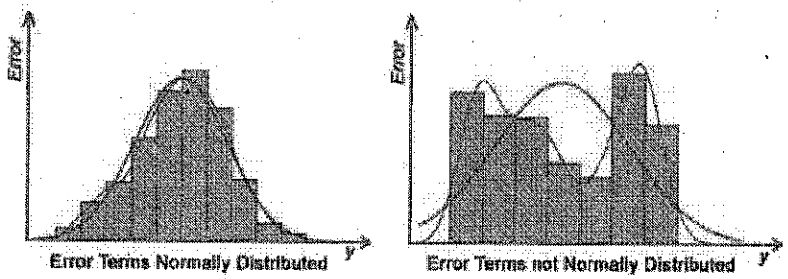
2. Independence of residuals: The error terms should not be dependent on one another (like in time-series data wherein the next value is dependent on the previous one). There should be no correlation between the residual terms. The absence of this phenomenon is known as Autocorrelation.

There should not be any visible patterns in the error terms.



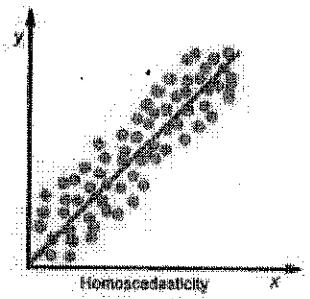
3. Normal distribution of residuals: The mean of residuals should follow a normal distribution with a mean equal to zero or close to zero. This is done in order to check whether the selected line is actually the line of best fit or not.

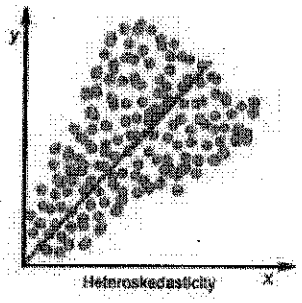
If the error terms are non-normally distributed, suggests that there are a few unusual data points that must be studied closely to make a better model.



4. The equal variance of residuals: The error terms must have constant variance. This phenomenon is known as Homoscedasticity.

The presence of non-constant variance in the error terms is referred to as Heteroscedasticity. Generally, non-constant variance arises in the presence of outliers or extreme leverage values.





Hypothesis in Linear Regression

Once you have fitted a straight line on the data, you need to ask, "Is this straight line a significant fit for the data?" Or "Is the beta coefficient explain the variance in the data plotted?" And here comes the idea of hypothesis testing on the beta coefficient.

The Null and Alternate hypotheses in this case are:

H0: B1 = 0

HA: B1 $\neq$ 0

To test this hypothesis we use a t-test, test statistics for the beta coefficient is given by,

Assessing the model fit

Some other parameters to assess a model are:

t statistic: It is used to determine the p-value and hence, helps in determining whether the coefficient is significant or not

F statistic: It is used to assess whether the overall model fit is significant or not. Generally, the higher the value of the F-statistic, the more significant a model turns out to be.