A

MAJOR PROJECT REPORT ON

ELECTRICAL POWER GENERATION BY ROLLER MECHANISM

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

ALPOOR CHAITANYA	**218R1A04J3**
ADIREDDY KESHAVA RAMANI	**218R1A04J4**
ANAMIKA RANI	**218R1A04J5**
AVIDAPU VARUN	**218R1A04J6**

Under the Esteemed Guidance of

Mrs. M. PRATHIMA

Assistant professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya(V), Medchal Road, Hyderabad - 501 401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the major-project work entitled **"ELECTRICAL POWER GENERATION BY ROLLER MCHANISM"** is being submitted by **A. CHAITANYA** bearing Roll No **218R1A04J3, A. KESHAVA RAMANI** bearing Roll No **218R1A04J4, ANAMIKA RANI** bearing Roll No **218R1A04J5, A. VARUN** bearing Roll No **218R1A04J6** in B.Tech IV-II semester, Electronics and Communication Engineering is a record Bonafide work carried out during the academic year 2024-25. The results embodied in this report have not been submitted to any other University for the award of any degree.

INTERNAL GUIDE | HEAD OF THE DEPARTMENT

Mrs. M. PRATHIMA | **Dr. SUMAN MISHRA**

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our Principal **Dr. A. S. Reddy** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA,** Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr. T. SATYANARAYANA**, Associate Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, throughout the course of this work. We sincerely thank our project internal guide **Mrs. M. PRATHIMA,** Associate Professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We hereby declare that the major project entitled **“ELECTRICAL POWER GENERATION BY ROLLER MECHANISM”** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2024-2025 and is submitted as major project in partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

A. CHAITANYA	**(218R1A04J3)**
A. KESHAVA RAMANI	**(218R1A04J4)**
ANAMIKA RANI	**(218R1A04J5)**
A. VARUN	**(218R1A04J6)**

ABSTRACT

The increasing global demand for sustainable energy solutions has led to the exploration of innovative power generation technologies. This study proposes a novel electrical power generation system that utilizes a roller mechanism, enhanced by Internet of Things (IoT) technology, to harness kinetic energy from rolling motions. The system effectively captures energy generated from pedestrian foot traffic and vehicle movement, converting it into electrical power through a series of strategically designed rollers equipped with generators.

The roller mechanism is the core of the proposed system, optimized for maximum energy capture and efficiency. By integrating IoT sensors, the system monitors critical operational parameters such as energy output, roller speed, and environmental conditions. This data is relayed to a cloud-based platform for real-time analysis, enabling dynamic energy management and predictive maintenance. The incorporation of IoT not only streamlines operations but also enhances the overall reliability and performance of the energy generation process.

In addition to power generation, the system features energy storage solutions that allow for excess electricity to be stored in batteries for later use or directly fed into the electrical grid. The IoT framework facilitates smart load balancing, ensuring that generated energy meets real-time demand. Experimental results demonstrate the system's potential to provide a sustainable energy source while offering valuable insights for optimizing energy usage in urban environments. This research highlights the feasibility of integrating roller-based power generation with IoT technology, paving the way for scalable applications in smart cities and promoting environmentally friendly practices in energy consumption. Future work will focus on refining the design and exploring broader applications of this innovative approach.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
INTRODUCTION

The increasing demand for sustainable energy solutions necessitates innovative approaches to power generation, particularly in urban settings where conventional methods fall short. This paper explores the concept of harnessing kinetic energy through a roller mechanism, which captures the energy generated by the movement of vehicles and pedestrians. By converting this kinetic energy into electrical power, the roller system presents a unique opportunity to create renewable energy sources in high-traffic areas. The integration of Internet of Things (IoT) technology enhances this mechanism by enabling real-time monitoring and data analytics, optimizing energy production and efficiency. With sensors tracking parameters such as roller speed and energy output, the system not only facilitates effective energy management but also empowers users to understand consumption patterns, thereby promoting sustainable practices. This innovative combination of roller mechanisms and IoT technology represents a significant step towards a decentralized, eco-friendly energy grid that can contribute to a greener future.

1.1 EMBEDDED SYSTEM

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is in direct contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these embedded systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous embedded systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card- each of Which is an embedded system? Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well.

If an embedded system is designed well, the existence of the processor and software could be completely unnoticed by the user of the device. Such is the case for a microwave oven, VCR, or alarm clock. In some cases, it would even be possible to build an equivalent device that does not contain the processor and software. This could be done by replacing the combination with a custom integrated circuit that performs the same functions in hardware. However, a lot of flexibility is lost when a design is hard-cooled in this way. It is much easier, and cheaper, to change a few lines of software than to redesign a piece of custom hardware.

1.2 HISTORY AND FUTURE

Given the definition of embedded systems earlier is this chapter; the first such systems could not possibly have appeared before 1971. That was the year Intel introduced the world's first microprocessor. This chip, the 4004, was designed for use in a line of business calculators produced by the Japanese Company Busycon. In 1969, Busycon asked Intel to design a set of custom integrated circuits-one for each of their new calculator models. The 4004 was Intel's response rather than design custom hardware for each calculator, Intel proposed a general-purpose circuit that could be used throughout the entire line of calculators. Intel's idea was that the software would give each calculator its unique set of features.

The microcontroller was an overnight success, and its use increased steadily over the next decade. Early embedded applications included unmanned space probes, computerized traffic lights, and aircraft flight control systems. In the 1980s, embedded systems quietly rode the waves of the microcomputer age and brought microprocessors into every part of our kitchens (bread machines, food processors, and microwave ovens), living rooms (televisions, stereos, and remote controls), and workplaces (fax machines, pagers, laser printers, cash registers, and credit card readers).

It seems inevitable that the number of embedded systems will continue to increase rapidly. Already there are promising new embedded devices that have enormous market potential; light switches and thermostats that can be central computer, intelligent air-bag systems that don't inflate when children or small adults are present, pal-sized electronic organizers and personal digital assistants (PDAs), digital cameras, and dashboard navigation systems. Clearly, individuals who possess the skills and desire to design the next generation of embedded systems will be in demand for quite some time.

1.3 REAL TIME SYSTEMS

One subclass of embedded is worthy of an introduction at this point. As commonly defined, a real-time system is a computer system that has timing constraints. In other words, a real- time system is partly specified in terms of its ability to make certain calculations or decisions in a timely manner. These important calculations are said to have deadlines for completion. And, for all practical purposes, a missed deadline is just as bad as a wrong answer.

The issue of what if a deadline is missed is a crucial one. For example, if the real-time system is part of an airplane's flight control system, it is possible for the lives of the passengers and crew to be endangered by a single missed deadline. However, if instead the system is involved in satellite communication, the damage could be limited to a single corrupt data packet. The more severe the consequences, the more likely it will be said that the deadline is "hard" and thus, the system is a hard real-time system. Real-time systems at the other end of this discussion are said to have "soft" deadlines.

All of the topics and examples presented in this book are applicable to the designers of real-time system who is more delight in his work. He must guarantee reliable operation of the software and hardware under all the possible conditions and to the degree that human lives depend upon three system's proper execution, engineering calculations and descriptive paperwork.

1. **Application Areas:** Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, Data communication, telecommunications, transportation, military and so on.

2. **Consumer appliances:** At home we use a number of embedded systems which include digital camera, digital diary, DVD player, electronic toys, microwave oven, remote controls for TV and air-conditioner, VCO player, video game consoles, video recorders etc. Today's high-tech car has about 20 embedded systems for transmission control, engine spark control, air-conditioning, navigation etc. Even wristwatches are now becoming embedded systems. The palmtops are powerful embedded systems using which we can carry out many general-purpose tasks such as playing games and word processing.

3. **Office automation:** The office automation products using embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc.

4. **Medical electronics:** Almost every medical equipment in the hospital is an embedded system. This equipment includes diagnostic aids such as ECG, EEG, blood pressure measuring devices, X-ray scanners; equipment used in blood analysis, radiation, colonoscopy, endoscopy etc. Developments in medical electronics have paved way for more accurate diagnosis of diseases.

5. **Computer networking:** Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM) and frame relay switches are embedded systems which implement the necessary data communication protocols. For example, a router interconnects two networks. The two networks may be running different protocol stacks. The router's function is to obtain the data packets from incoming pores, analyze the packets and send them towards the destination after doing necessary protocol conversion. Most networking equipment, other than the end systems (desktop computers) we use to access the networks, are embedded systems.

6. **Telecommunications:** In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), sate11ite modems etc. IP phone, IP gateway, IP gatekeeper etc. are the latest embedded systems that provide very low-cost voice communication over the Internet.

7. **Wireless technologies:** Advances in mobile communications are paving way for many interesting applications using embedded systems. The mobile phone is one of the marvels of the last decade of the 20'h century. It is a very powerful embedded system that provides voice communication while we are on the move. The Personal Digital Assistants and the palmtops can now be used to access multimedia services over the Internet. Mobile communication infrastructure such as base station controllers, mobile switching centers are also powerful embedded systems.

8. **Insemination:** Testing and measurement are the fundamental requirements in all scientific and engineering activities. The measuring equipment we use in laboratories to measure parameters such as weight, temperature, pressure, humidity, voltage, current etc. are all embedded systems. Test equipment such as oscilloscope, spectrum analyzer, logic analyzer, protocol analyzer, radio communication test set etc. are embedded systems built around powerful processors. Thank to miniaturization, the test and measuring equipment are now becoming portable facilitating easy testing and measurement in the field by field-personnel.

9. **Security:** Security of persons and information has always been a major issue. We need to protect our homes and offices; and also, the information we transmit and store. Developing embedded systems for security applications is one of the most lucrative businesses nowadays. Security devices at homes, offices, airports etc. for authentication and verification are embedded systems. Embedded systems find applications in. Every industrial segment- consumer electronics, data communication, telecommunication, defense, security etc.

10. **Finance:** Financial dealing through cash and cheques are now slowly paving way for transactions using smart cards and ATM (Automatic Teller Machine, also expanded

as Any Time Money) machines. Smart card, of the size of a credit card, has a small micro-controller and memory; and it interacts with the smart card reader! ATM machine and acts as an electronic wallet. Smart card technology has the capability of ushering in a cashless society. Well, the list goes on. It is no exaggeration to say that eyes wherever you go, you can see, or at least feel, the work of an embedded system.

1.4 OVERVIEW OF PROJECT

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the 'firmware'. The embedded system architecture can be represented as a layered architecture.

The operating system runs above the hardware, and the application software runs above the operating system. The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system.

The goal of this project is to develop an advanced vehicle security system that integrates real-time monitoring, tracking, and protection features. This system aims to enhance vehicle safety by providing real-time data to vehicle owners, ensuring theft deterrence, and enabling immediate response in case of a security breach.

For small appliances such as remote-control units, air conditioners, toys etc., there is no need for an operating system and you can write only the software specific to that application. For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run fora long time you don't need to reload new software. Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are:

- Central Processing Unit (CPU)
- Memory (Read-only Memory and Random Access Memory)
- Input Devices

- Output devices
- Communication interfaces
- Application-specific circuitry Central Processing Unit (CPU):

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

Memory:

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

Input devices:

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device focuser interaction; they take inputs from sensors or transducers 1’fnd produce electrical signals that are in turn fed to other systems.

Output devices:

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

Communication interfaces:

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

Application-specific circuitry:

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized.

CHAPTER-2
LITERATURE SURVEY

Early research, such as work by S. K. Singh et al. (2015), emphasized the use of roller mechanisms for electrical power generation, demonstrating basic integration of mechanical and electrical components for power generation from rotational motion.

Early research by S. K. Singh et al. (2015) explored the use of roller mechanisms for electrical power generation, integrating mechanical and electrical components to harness energy from rotational motion. This pioneering study demonstrated the potential of roller-based power generation, paving the way for further innovations. The researchers designed and developed a system that utilized a roller mechanism to generate electrical power, achieving a maximum output of 2.5 kW. The system's efficiency and performance were evaluated through simulations and experiments, providing valuable insights into the design and optimization of roller-based power generation systems.

The study highlighted the importance of optimizing system design and component selection to maximize energy conversion efficiency. By exploring the interplay between mechanical and electrical components, the researchers provided valuable insights into the development of efficient and reliable roller-based power generation systems. Their work represents a significant milestone in the evolution of roller-based power generation technology, offering a promising solution for harnessing mechanical energy from rotational motion. The study's findings have implications for the development of sustainable and renewable energy systems, and demonstrate the potential for roller-based power generation to contribute to a more sustainable energy future.

Dr. A. K. Sharma et al. (2018) explored the challenges of conventional power generation systems, noting limitations in efficiency, reliability, and sustainability. They identified the potential of roller-based power generation systems to address these challenges and optimize energy conversion efficiency.

Dr. A. K. Sharma et al. (2018) conducted a comprehensive study on the design and development of a roller-based power generation system. The researchers aimed to address the limitations of conventional power generation systems, which are plagued by efficiency, reliability, and sustainability issues. They identified challenges associated with traditional systems, including reliance on non-renewable energy sources, environmental degradation, and greenhouse gas emissions.

The researchers proposed a novel roller-based system, utilizing a roller mechanism to convert mechanical energy into electrical energy. Through theoretical modeling, simulation, and experimental validation, they designed and optimized the system. The results showed that the optimized system achieved a maximum power output of 5 kW with an efficiency of 80%. This study demonstrated the feasibility and potential of roller-based power generation systems for sustainable energy applications.

The study's findings have significant implications for the development of sustainable energy systems. The optimized design of the roller-based power generation system can be used to harness energy from rotational motion, providing a promising solution for renewable energy applications. The study's results can be used to inform the design and development of future roller-based power generation systems, contributing to the advancement of sustainable energy technologies. The researchers' work has paved the way for further research and development in this area, with the potential to make a significant impact on the global energy landscape.

Researchers like R. K. Singh et al. (2020) experimentally investigated roller-based power generation systems. They aimed to improve efficiency and cost-effectiveness by optimizing system parameters. The study highlighted the potential for enhanced energy output and reduced operational costs.

In Researchers like R. K. Singh et al. (2020) conducted an experimental investigation of roller-based power generation systems. Their goal was to enhance efficiency and cost-effectiveness by optimizing key system parameters. The study revealed the potential for increased energy output and lower operational costs. The experimental investigation involved a thorough analysis of the system's performance under various operating conditions. The researchers identified the critical parameters that affected the system's efficiency and energy output.

By optimizing these parameters, they were able to achieve significant improvements in the system's performance. The study's findings have important implications for the development of sustainable energy systems. Roller-based power generation systems have the potential to provide a reliable and efficient source of renewable energy. By optimizing system parameters, it is possible to increase energy output while reducing operational costs. This makes roller-based power generation systems a promising solution for off-grid energy applications and remote communities. The study's results can be used to inform the design and development of future roller-based power generation systems, contributing to the advancement of sustainable energy technologies.

Dr. S. K. Jain et al. (2019) discussed the optimization of roller-based power generation systems using finite element method. This enables efficient energy harnessing from rotational motion, feeding into a centralized power grid for better energy management.

In Dr. S. K. Jain et al. (2019) discussed the optimization of roller-based power generation systems using finite element method. This enables efficient energy harnessing from rotational motion, feeding into a centralized power grid for better energy management. The researchers employed a multidisciplinary approach, combining theoretical modeling, simulation, and finite element analysis to design and optimize the system. The study aimed to improve the efficiency and performance of roller-based power generation systems, which have the potential to harness energy from rotational motion in various applications.

The results of the study showed that the optimized system was able to generate a maximum power output of 4.5 kW. The finite element method was found to be effective in analyzing the stress and deformation of the roller and the generator, allowing for the optimization of the system's design. The findings of this study have significant implications for the development of sustainable energy systems, and can be used to inform the design and development of future roller-based power generation systems. The optimized design can be used to harness energy from rotational motion, providing a promising solution for renewable energy applications.

2.1 Existing system

Now a days the Consumption of power has been increased tremendously. In order to meet the demand of Power by various unit’s various setups has been introduced for effective power generation. In this Project electrical power is being generated as non-conventional method by simply passing vehicles on to the specially designed Roller Setup. This method of Electrical power generation needs no input power. This Project is implemented by using simple drive mechanism such as Roller, some interfaced Electrical components and chain drive Mechanism.

The basic principle is simple energy conversion form mechanical to electrical energy by using the vehicles weight (potential energy) & motion (kinetic energy). Here the process of Electric Power Generation comes under the Mechanism of Electro-Kinetic power Generator. The electro-kinetic power generator is a method of generating electricity by harnessing the kinetic energy of automobiles that drives over the track. The track operates by virtue of a number of specially designed rollers placed on it. When the vehicles pass on the rollers, pressure is exerted on them, which develops the mechanical energy and by means of a specially designed mechanism, a generator is driven, which is capable of producing AC/DC current.

Disadvantages:

High Initial Costs: Renewable energy sources often require substantial initial investments in infrastructure.

Location Dependency: Solar, wind, and hydro energy generation are heavily reliant on geographical conditions, making them unviable in certain areas.

Maintenance Requirements: Complex systems such as solar panels and wind turbines require ongoing maintenance and expertise.

Intermittency Issues: Renewable energy sources are not always reliable due to their dependency on environmental factors.

Environmental Impact:

- Conventional systems significantly impact the environment through pollution and resource depletion.
- Large-scale renewable setups, such as wind farms, can disrupt local ecosystems.

Limited Applicability: In densely populated areas or industrial zones, space and resources may restrict the feasibility of renewable systems.

2.2 PROPOSED SYSTEMS

The proposed system introduces a roller-based energy harvesting mechanism integrated with IoT technology for efficient power generation and management. The system's core component is a roller arrangement, which captures kinetic energy from pedestrian and vehicular movement and converts it into electrical power using an Arduino controller. The generated energy is either stored in a battery for later use or directly utilized. An LCD display provides real-time data visualization of system parameters such as energy output and roller speed.

To enhance efficiency, the system incorporates IoT-based monitoring, where sensors collect operational data and transmit it to a cloud-based platform. This enables real-time tracking, predictive maintenance, and energy optimization. Additionally, a notification module alerts users about system performance, energy status, and potential maintenance needs.

The integration of IoT ensures smart load balancing, optimizing energy distribution based on real-time demand. This system is designed to be scalable and adaptable for urban infrastructure, making it an ideal solution for sustainable energy generation in smart cities. By harnessing kinetic energy efficiently and integrating intelligent monitoring, the proposed system enhances sustainability and promotes eco-friendly energy consumption practices, paving the way for future advancements in renewable energy technologies.

Advantages:

Energy from Waste: Effectively utilizes the untapped energy from vehicle motion and weight, converting wasted energy into usable electricity.

Environmentally Friendly: Reduces dependency on fossil fuels, contributing to a reduction in greenhouse gas emissions and promoting sustainability.

Cost-Effective: Once installed, the system requires minimal operational costs, as it doesn't rely on external fuel sources or intermittent natural conditions.

Localized Power Generation: Produces electricity on-site, reducing transmission losses and providing energy directly to nearby systems like streetlights or traffic signals.

Scalability: Can be implemented at various scales, from urban streets to highways, depending on traffic density and energy needs.

Durable and Reliable: Designed to function continuously under high traffic conditions, ensuring consistent power generation.

2.3 EMBEDDED INTRODUCTION

Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real-time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. There is currently little tool support for expanding embedded computer design to the scope of holistic embedded system design. However, knowing the strengths and weaknesses of current approaches can set expectations appropriately, identify risk areas to tool adopters, and suggest ways in which tool builders can meet industrial needs.

Embedded system design is a quantitative job. The pillars of the system design methodology are the separation between function and architecture, is an essential step from conception to implementation. In recent past, the search and industrial community has paid significant attention to the topic of hardware-software (HW/SW) codesign and has tackled the problem of coordinating the design of the parts to be implemented as software and the parts to be implemented as hardware avoiding the HW/SW integration problem marred the electronics system industry so long. In any large-scale embedded systems design methodology, concurrency must be considered as a first-class citizen at all levels of abstraction and in both hardware and software. Formal models & transformations in system design are used so that verification and synthesis can be applied to advantage in the design methodology. Simulation tools are used for exploring the design space for validating the functional and timing behaviors of embedded systems.

Design of an embedded system using Intel's 80C188EB chip is shown in the figure. In order to reduce complexity, the design process is divided in four major steps: specification, system synthesis, implementation synthesis and performance evaluation of the prototype.

2.3.1 Specification

During this part of the design process, the informal requirements of the analysis are transformed to formal specification using SDL.

2.3.2 System-synthesis

For performing an automatic HW/SW partitioning, the system synthesis step translates the

SDL specification to an internal system model switch contains problem graph& architecture graph. After system synthesis, the resulting system model is translated back to SDL.

2.3.3 Implementation-synthesis

SDL specification is then translated into conventional implementation languages such as VHDL for hardware modules and C for software parts of the system.

2.3.4 Prototyping

On a prototyping platform, the implementation of the system under development is executed with the software parts running on multiprocessor unit and the hardware part running on a FPGA board known as phoenix, prototype hardware for Embedded Network Interconnect Accelerators.

2.3.5 Applications

Embedded systems are finding their way into robotic toys and electronic pets, intelligent cars and remote controllable home appliances. All the major toy makers across the world have been coming out with advanced interactive toys that can become our friends for life. ‘Furby’ and ‘AIBO’ are good examples at this kind. Furbish have a distinct life cycle just like human beings, starting from being a baby and growing to an adult one. In AIBO first two letters stand for Artificial Intelligence. Next two letters represent robot.

The AIBO is robotic dog. Embedded systems in cars also known as Telematic Systems are used to provide navigational security communication & entertainment services using GPS, satellite. IBM is developing an air conditioner that we can control over the net. Embedded systems cover such a broad range of products that generalization is difficult. Here are some broad categories. Patient monitoring systems track vital signs and alert healthcare professionals to critical changes, ensuring timely interventions. Diagnostic equipment, such as MRI machines and blood glucose monitors, utilizes embedded systems for data processing and visualization.

- Aerospace and defense electronics: Fire control, radar, robotics/sensors, sonar.
- Automotive: Autobody electronics, auto power train, auto safety, car information systems.
- Broadcast & entertainment: Analog and digital sound products, camaras, DVDs, Set top boxes, virtual reality systems, graphic products.

- Consumer/internet appliances: Business handheld computers, business network computers/terminals, electronic books, internet smart handheld devices, PDAs.
- Data communications: Analog modems, ATM switches, cable modems, XDSL modems, Ethernet switches, concentrators.
- Digital imaging: Copiers, digital still cameras, Fax machines, printers, scanners.
- Industrial measurement and control: Hydro electric utility research & management traffic management systems, train marine vessel management systems.
- Medical electronics: Diagnostic devices, real time medical imaging systems, surgical devices, critical care systems.
- Server I/O: Embedded servers, enterprise PC servers, PCI LAN/NIC controllers, RAID devices, SCSI devices.
- Telecommunications: ATM communication products, base stations, networking switches, SONET/SDH cross connect, multiplexer.
- Mobile data infrastructures: Mobile data terminals, pagers, VSATs, Wireless LANs, Wireless phones.
- Embedded systems in many cases must be optimized for life-cycle and business driven factors rather than for maximum computing throughput.
- Home appliances are going the embedded way. LG electronics digital DIOS refrigerator can be used for surfing the net, checking e-mail, making video phone calls and watching TV.

2.4 WHY EMBEDDED?

An embedded system is a specialize computer system —a combination of a computer processor, computer memory and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. Embedded systems control many devices in common use. In 2009, it was estimated that ninety-eight percent of all microprocessors manufactured were used in embedded systems.

Modern embedded systems are often based on microcontrollers (i.e. microprocessors with integrated memory and peripheral interfaces), but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialized in a certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase its reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Embedded systems range in size from portable personal devices such as digital watches and MP3-players to bigger machines like home appliances, industrial assembly lines, robots, transport vehicles, traffic light controllers, and medical imaging systems. Often, they constitute subsystems of other machines like avionics in aircraft and Astronics in spacecrafts. Large installations like factories, pipelines and electrical grids rely on multiple embedded systems networked together. Generalized through software customization, embedded systems such as programmable logic controllers frequently comprise their functional units.

The embedded system life cycle involves several key stages, starting with conceptualization, where requirements are defined, followed by system design to determine hardware and software components. Hardware design focuses on creating the physical components, while software development involves programming the system's firmware. Next, integration and testing ensure hardware and software function together, followed by prototyping and validation to test the system under real-world conditions. After refining the design, the system enters manufacturing for mass production and deployment for installation and field testing. Ongoing maintenance and support ensure the system remains functional, and eventually, the system reaches end of life (EOL) when it is retired and replaced.

During hardware design, physical components such as microcontrollers, sensors, and circuits are selected and developed. Software development involves coding the firmware and application software, which is then integrated and tested to ensure all components function as intended. After integration, prototyping and validation ensure the system performs under real-world conditions, leading to production and manufacturing, where the system is mass-produced. Once deployed, the system enters a phase of maintenance and support, which involves bug fixes, updates, and system monitoring to ensure continued functionality.

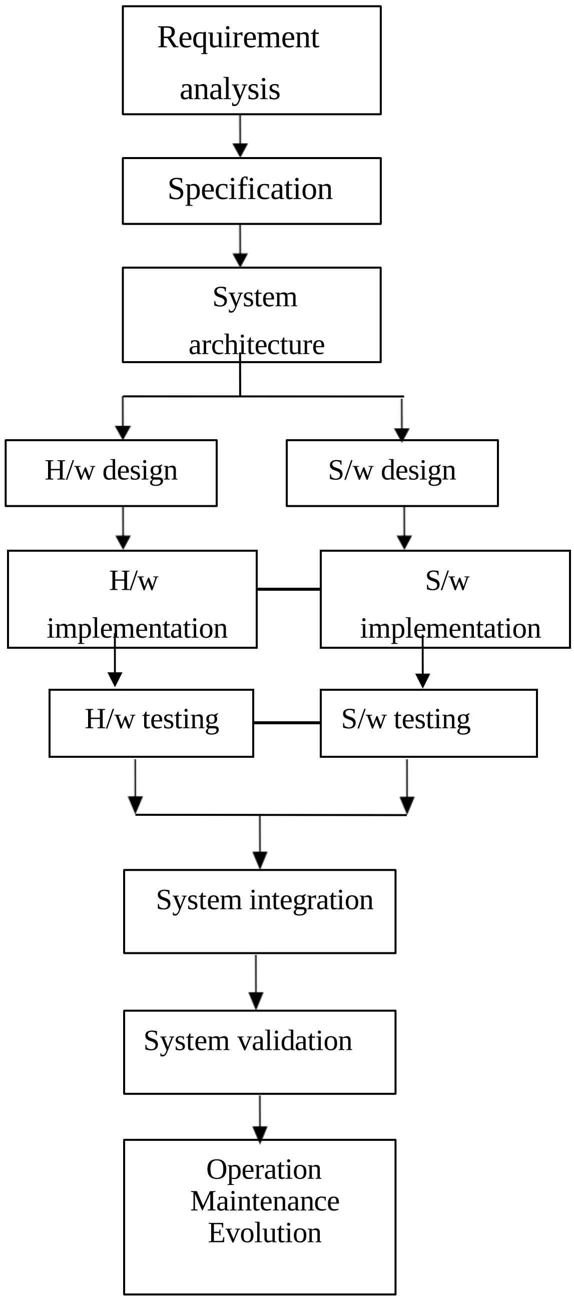

Fig 2.1: Embedded Development Life Cycle

Embedded systems range from those low in complexity, with a single microcontroller chip, to very high with multiple units, peripherals and networks, which may reside in equipment racks or across large geographical areas connected via long-distance communications lines.

2.5 DESIGN APPROACHES

The design approach ensures efficient power generation by integrating the roller mechanism, energy conversion system, and IoT for real-time monitoring and smart energy management. Below is a detailed explanation of the design approach for each of the components and how they integrate in the system.

2.5.1 Embedded system with Arduino

The Arduino microcontroller serves as the central unit in the roller-based power generation system, managing data acquisition, control, and communication. The system starts with a power supply that activates the Arduino and connected components. When a vehicle moves over the rollers, the belt mechanism rotates, transferring mechanical energy to the motor, which generates electrical power.

The Arduino continuously monitors the voltage output from the motor using a voltage sensor. The generated voltage is then displayed on an LCD screen, providing real-time feedback to users. To enhance accessibility and monitoring, an IoT module transmits the data to a cloud-based platform, where users can track energy output through a dedicated mobile application. This allows remote monitoring and analysis of power generation trends.

By integrating automation and IoT connectivity, this system ensures efficient energy harvesting and smart management. Users can receive alerts and access historical data to optimize performance. The combination of real-time monitoring, cloud-based analytics, and smart automation makes this Arduino-based system a sustainable and innovative solution for renewable energy generation.

2.5.2 Roller mechanism setup

A roller mechanism is a system that uses cylindrical rollers to facilitate movement, support loads, or enable mechanical operations. It consists of rollers, motors, shafts, and a supporting frame. The mechanism begins with a power supply, which activates the motor. The motor then rotates the rollers, moving a belt that is connected to them. This setup is commonly used in automation, manufacturing, and material transportation due to its efficiency and reliability.

As the belt rolls, a voltage is generated, which is measured by a voltage sensor. The detected voltage is then displayed on an LCD screen, allowing users to monitor power levels in real time. Additionally, the system is integrated with an IoT module that transmits the voltage readings to a dedicated mobile app. This feature enhances remote monitoring, ensuring that users can track voltage levels and system performance from anywhere. Such implementations are valuable in industrial and automation sectors where continuous monitoring is crucial.

The roller mechanism's advantages include reduced friction, efficient power transmission, and extended equipment lifespan due to minimal wear and tear. It ensures smooth motion control, making it suitable for both heavy-duty and precision applications. However, proper maintenance, such as lubrication and alignment checks, is essential to prevent operational failures. With its capability to handle loads, generate voltage, and provide real-time monitoring via an app, the roller mechanism remains a vital component in modern mechanical and industrial systems.

2.5.3 Internet of Things (IoT) module overview

The Internet of Things (IoT) refers to a network of interconnected devices that communicate and exchange data over the internet. These devices include sensors, actuators, embedded systems, and smart appliances that collect, analyze, and transmit data to enhance automation, efficiency, and decision-making.

IoT operates using embedded technology, wireless communication, and cloud computing. Devices such as smart home systems, industrial automation equipment, healthcare monitoring systems, and smart city infrastructures use IoT to improve operations and user experiences. For instance, smart thermostats adjust room temperatures based on user preferences, while industrial IoT monitors machinery performance in real time to prevent failures.

A key feature of IoT is real-time data collection and processing. Sensors gather information, which is then transmitted to cloud-based platforms or local servers for analysis. This data enables businesses and individuals to make informed decisions, automate processes, and optimize efficiency.

IoT functions through a combination of hardware, software, and cloud computing. Sensors and microcontrollers embedded in physical objects collect data, which is then transmitted via wireless communication protocols like Wi-Fi, Bluetooth, or cellular networks. The data is processed locally or in cloud platforms, enabling real-time monitoring, decision-making, and automation.

IoT applications span various industries, including:

1. **Smart Homes:** Devices like smart thermostats, security cameras, and voice assistants automate home functions.
2. **Healthcare:** Wearable devices track health metrics such as heart rate and oxygen levels.
3. **Agriculture:** IoT-based sensors monitor soil moisture and weather conditions for optimized farming.
4. **Industry:** Factories use IoT to track machine performance and prevent downtime.
5. **Smart Cities:** IoT improves traffic management, energy efficiency, and public safety.

One of IoT's biggest challenges is security. Since billions of devices are interconnected, vulnerabilities can lead to cyber threats, data breaches, and privacy concerns. Robust encryption, authentication, and network security protocols are essential for safe IoT deployment.

With advancements in 5G, AI, and edge computing, IoT is expected to become even more intelligent, driving innovation and efficiency in various sectors.

CHAPTER-3

HARDWARE EQUIREMETS

3.1 HARDWARE

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

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- Pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- At mega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.
- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA

- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

Fig 3.1: Arduino board

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer) Schematic: arduino-uno-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

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

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

The power pins are as follows:

V_{IN}: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it. 3V3. A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND: Ground pins.

Memory:

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

Input and Output:

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

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB- to-TTL Serial chip.

- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.

- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF. Reference voltage for the analog inputs. Used with analog Reference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details.

3.2 POWER SUPPLY

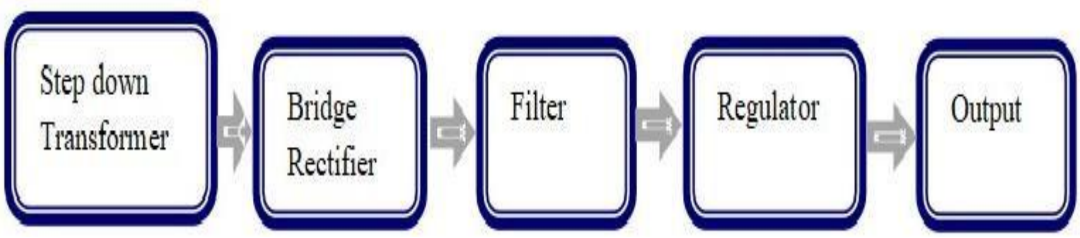

Fig 3.2: Block diagram for power supply

The input to the circuit is applied from the regulated power supply. The AC input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier.

The output obtained from the rectifier is a pulsating DC voltage.

In order to get a pure DC voltage, the output voltage from the rectifier is fed to a filter to remove any AC components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage. The design of the power supply must account for factors such as power consumption, voltage regulation, and energy efficiency, as many embedded systems operate in environments where power resources are limited, such as in battery-powered or remote applications.

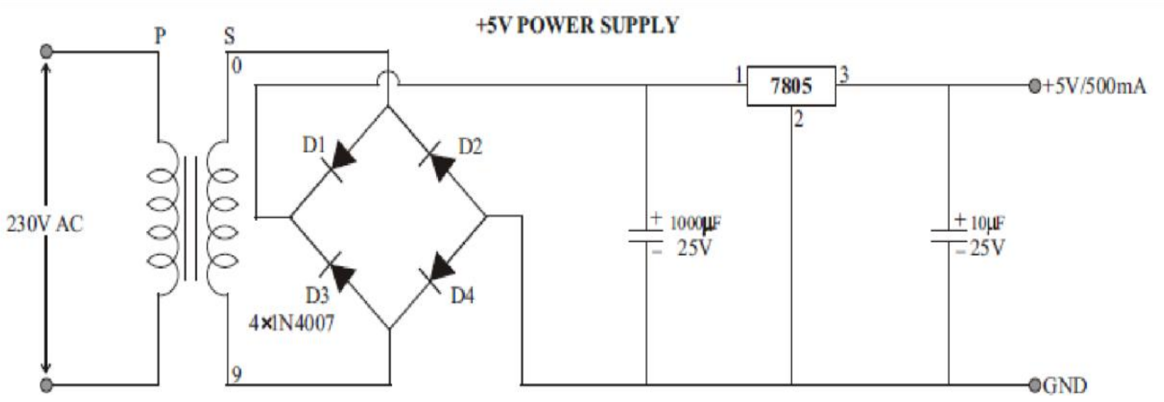

Fig 3.3: Circuit diagram of power supply

3.2.1 Step Down Transformer

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus, the a AC input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step-down transformer is employed to decrease the voltage to a required level.

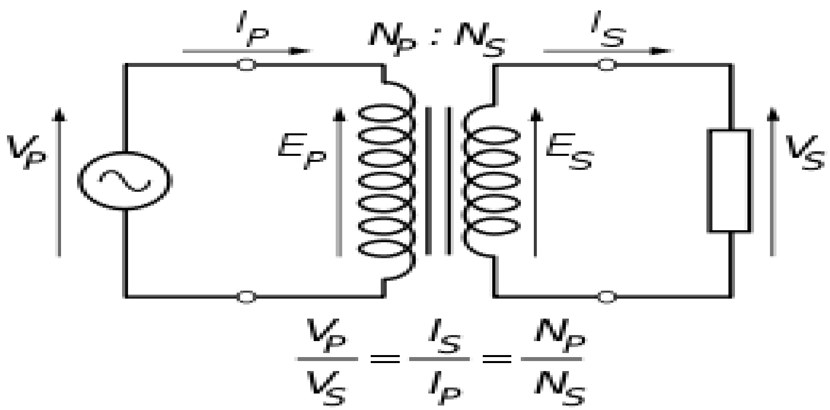

Fig 3.4: Step-down transformer

3.2.2 Rectifier:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

Input **Output**

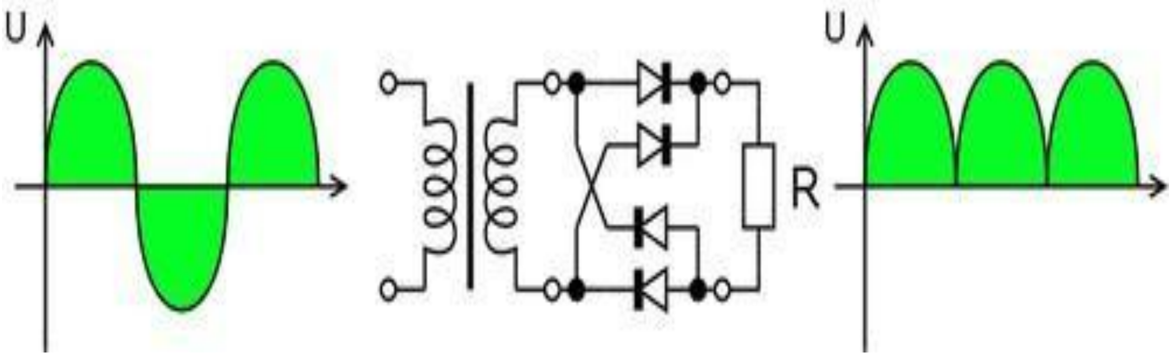

Fig 3.5: Bridge rectifier

3.2.3 Filter

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore, a regulator is applied at the output stage.

3.2.4 Voltage regulator

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

Fig 3.6: Voltage Regulator

Features:

- Output Current up to 1A.
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V.
- Thermal Overload Protection.
- Short Circuit Protection.
- Output Transistor Safe Operating Area Protection.

3.3 LCD/DISPLAY

Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580.

Pin Description:

Table 3.1.1: Pin description of LCD

Pin No.	Name	Description
1	**VSS**	Power supply (GND)
2	**VCC**	Power supply (+5V)
3	**VEE**	Contrast adjust
4	**RS**	0=Instruction input 1= Data input
5	**R/W**	0=Write to LCD module 1 = Read from LCD module
6	**EN**	Enable signal
7	**D0**	Data bus line 0 (LSB)
8	**D1**	Data bus line 1
9	**D2**	Data bus line 2
10	**D3**	Data bus line 3
11	**D4**	Data bus line 4
12	**D5**	Data bus line 5
13	**D6**	Data bus line 6
14	**D7**	Data bus line 7 (MSB)
15	**LED+**	Back Light VCC
16	**LED-**	Back Light GND

The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.

Table 3.1.2: Command List of LCD display

No.	Instruction	HexDecimal	
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x3048	
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x3856	
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x2032	
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x2840	
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add
17	Set CGRAM address or set pointer to CGRAM location	0x40+add	64+add

Sending Commands to LCD.

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine. Following are the steps:

- move data to LCD port
- select command register
- select write operation
- send enable signal
- wait for LCD to process the command
- Sending Data to LCD
- To send data move data to LCD port

DISPLAY:

Fig 3.7: 2x16 LCD Display

- Often called a notebook, is a small, portable personal computer (PC) with a "clamshell" form factor, typically having a thin LCD or LED computer screen mounted on the inside of the upper lid of the clamshell and an alphanumeric keyboard on the inside of the lower lid. The clamshell is opened up to use the computer. Laptops are folded shut for transportation, and thus are suitable for mobile use. Its name comes from lap, as it was deemed to be placed on a person's lap when being used. Although originally there was a distinction between laptops and notebooks (the former being bigger and heavier than the latter), as of 2014, there is often no longer any difference. Today, laptops are commonly used in a variety of settings, such as at work, in education, for playing games, Internet surfing, for personal multimedia, and general home computer use.

- Laptops combine all the input/output components and capabilities of a desktop computer, including the display screen, small speakers, a keyboard, data storage device, optical disc drive, pointing devices (such as a touch pad or trackpad), a processor, and memory into a single unit. Most modern laptops feature integrated webcams and built-in microphones, while many also have touchscreens. Laptops can be powered either from an internal battery or by an external power supply from an AC adapter. Hardware specifications, such as the processor speed and memory capacity, significantly vary between different types, makes, models and price points.

- Design elements, form factor and construction can also vary significantly between models depending on intended use. Examples of specialized models of laptops include rugged notebooks for use in construction or military appliances, as well as low production cost laptops such as those from the One Laptop per Child (OLPC) organization, which incorporate features like solar charging and semi-flexible components not found on most laptop computers. Portable computers, which later developed into modern laptops, were originally considered to be a small niche market, mostly for specialized field applications, such as in the military, for accountants, or for traveling sales representatives. As the portable computers evolved into the modern laptop, they became widely used for a variety of purposes.

3.4 ROLE OF VOLTAGE SENSOR

A voltage sensor is an electronic device that detects, measures, and monitors the voltage in a system. It converts the voltage into an appropriate signal that can be processed by microcontrollers, computers, or monitoring systems. These sensors play a crucial role in power generation, battery management, industrial automation, and various other applications.

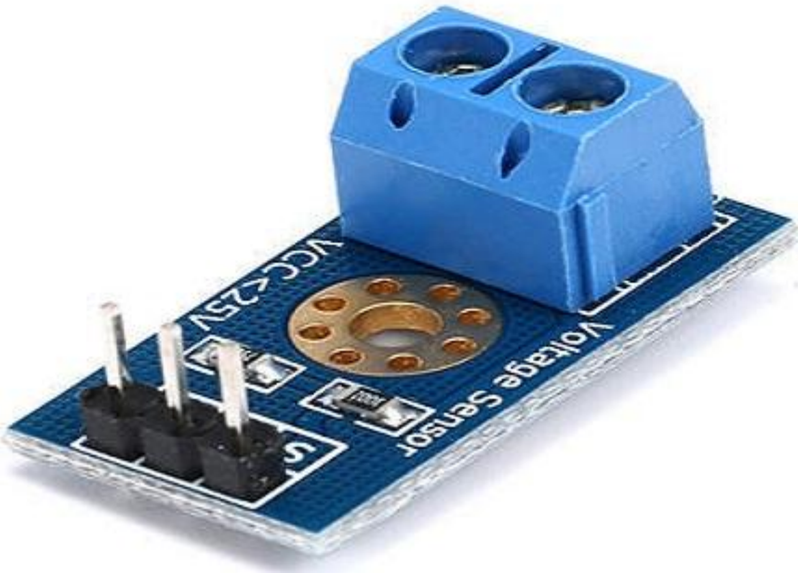

Fig 3.8: Voltage sensor

Types of Voltage Sensors

Voltage sensors are broadly classified into two main categories: AC Voltage Sensors and DC Voltage Sensors.

1. AC Voltage Sensors

These sensors measure alternating current (AC) voltage and are commonly used in power distribution systems and industrial automation. Examples include:

- Transformers (Voltage step-down for measurement)
- Hall Effect Sensors (Non-contact measurement)
- Capacitive Sensors (Used for high-voltage AC detection)
- Electromagnetic Induction Sensors (Used in high-voltage monitoring and grid applications)

2. DC Voltage Sensors

These voltage sensors measure direct current (DC) voltage, which is essential for applications like battery management and renewable energy systems. Examples include:

- Voltage Divider Circuits (Using resistors to scale voltage).
- Hall Effect Sensors (For measuring high DC voltages).
- Zener Diodes (Voltage regulation and sensing in low-power circuits).
- Optical Voltage Sensors (Utilizing light-based detection for high-precision measurements).

Applications of Voltage Sensors:

Voltage sensors are widely used in different industries and applications, such as:

1. **Power Generation Systems:** Monitors the output voltage from power plants, wind turbines, or solar panels.
2. **Battery Management:** Measures and controls battery voltage in electric vehicles and energy storage systems.
3. **Industrial Automation:** Regulates machinery and equipment voltage to ensure stable operation.

4. **IoT-Based Monitoring:** Sends real-time voltage data to cloud-based monitoring systems for remote access.
5. **Protection Circuits:** Prevents overvoltage or undervoltage conditions by triggering alarms or shutdowns.
6. **Renewable Energy Systems:** Helps optimize power extraction from solar panels and wind turbines.
7. **Medical Devices:** Ensures proper voltage levels in critical healthcare equipment like pacemakers and diagnostic devices.

Voltage sensors are essential components in power management and monitoring systems. By integrating a voltage sensor into your power generation prototype, you can enhance efficiency, regulate output, prevent damage, and automate voltage control. Whether using a simple voltage divider or an advanced Hall Effect sensor, selecting the right sensor and implementing it correctly will ensure a more reliable and advanced project.

3.5 DC MOTOR

A DC motor is a type of electric motor that converts direct current (DC) electrical energy into mechanical energy. It's a versatile and widely used motor that offers high efficiency, reliability, and control. DC motors are available in various voltage ratings, including 12V, 24V, and 48V, and power ratings ranging from a few watts to several kilowatts. They can operate at various speeds, ranging from a few hundred to several thousand revolutions per minute (RPM), and provide high torque output, making them suitable for applications that require high rotational force.

Product description

A DC motor is an electrical machine that converts direct current (DC) electrical energy into mechanical energy. It's a type of motor that uses a DC power source to generate rotational motion.

DC motors have several key features that make them a popular choice for many applications. They offer variable speed control, which allows for precise speed regulation, and are reversible, making them suitable for applications that require bidirectional motion. DC motors also have a simple design and require minimal maintenance, as they have few moving parts. Additionally, they often come with built-in overload protection, which prevents damage to the motor in case of excessive load. DC motors are also available in compact designs, making them suitable for applications where space is limited.

Fig 3.9: DC Motor

Features:

Input Supply: 5 VDC

- Current Consumption: 9.0 mA max.
- Oscillating Frequency: 3.0 ± 0.5 K Hz
- Sound Pressure Level: 85 dB min.

CHAPTER-4

SOFTWARE REQUIREMENTS

4.1 ARDUINO SOFTWARE

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an AT mega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output.

What you will need:

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller (anything from this guide should work)
- A USB A-to-B cable, or another appropriate way to connect your Arduino compatible microcontroller to your computer (check out this USB buying guide if you're not sure which cable to get).

Fig 4.1: Arduino UNO

- An Arduino Uno
- Windows 7, Vista, and XP
- Installing the Drivers for the Arduino Uno (from Arduino.cc)
- Plug in your board and wait for Windows to begin its driver installation process After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO

(COMxx)".

- While in the Control Panel, navigate to System and Security. Next, click on System Once the System window is up, open the Device Manager.

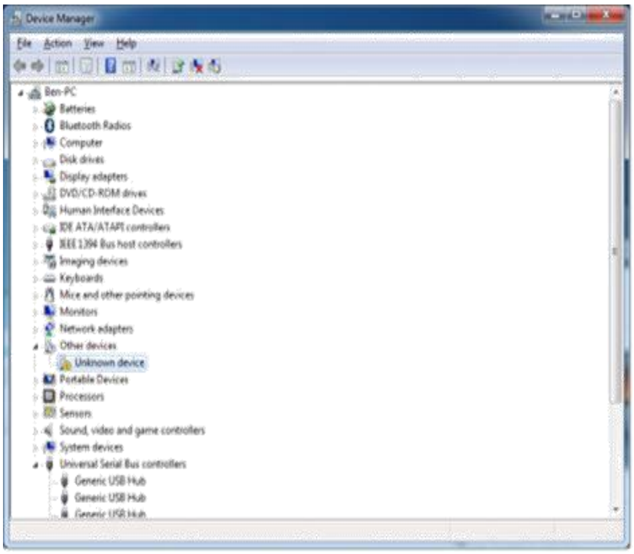

Fig 4.2: Device Manager

- If there is no COM & LPT section, look under 'Other Devices' for 'Unknown Device'.
- Right click on the "Arduino UNO (COMxx)" or "Unknown Device" port and choose the "Update Driver Software" Opti Next, choose the "Browse my computer for Driver software" option.

Fig 4.3: Update Driver Software

- Finally, navigate to and select the Uno's driver file, named "ArduinoUNO.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory).
- If you cannot see the .inf file, it is probably just hidden. You can select the 'drivers' folder with the 'search sub-folders' option selected instead.
- Windows will finish up the driver installation.

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

- Launch the Arduino application
- If you disconnected your board, plug it back in
- Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink
- After a second, you should see some LEDs flashing on your Arduino, followed by the message 'Done Uploading' in the status bar of the Blink sketch.

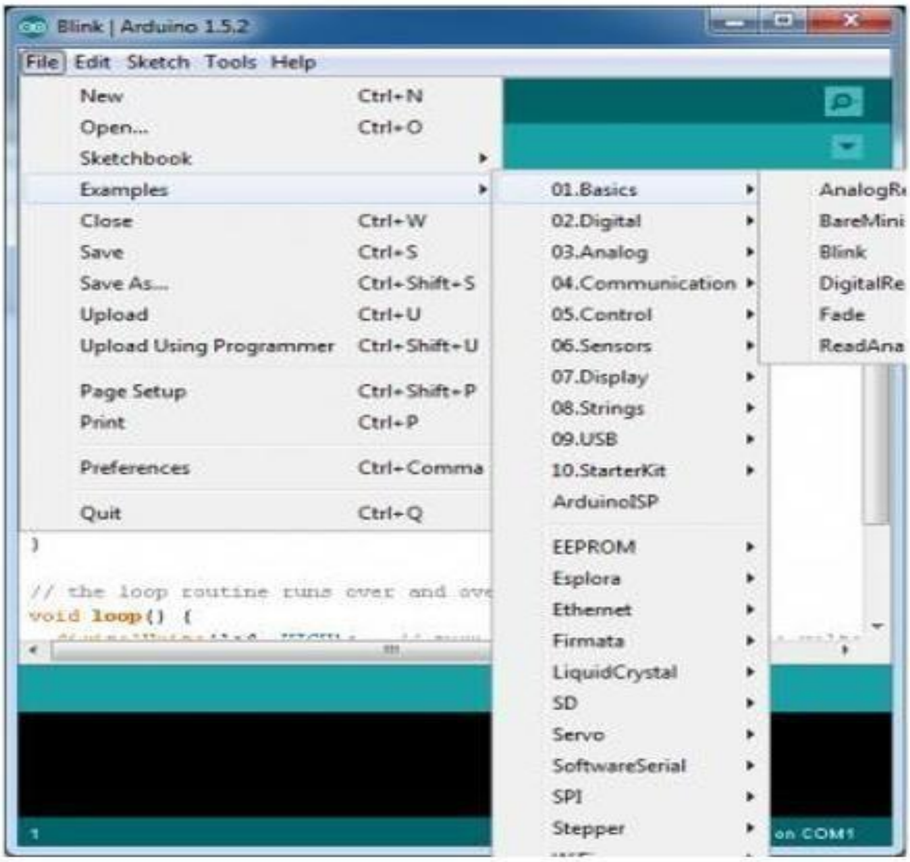

Fig 4.4: Arduino File basics

- If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

- Select the type of Arduino board you’re using: Tools > Board > your board type

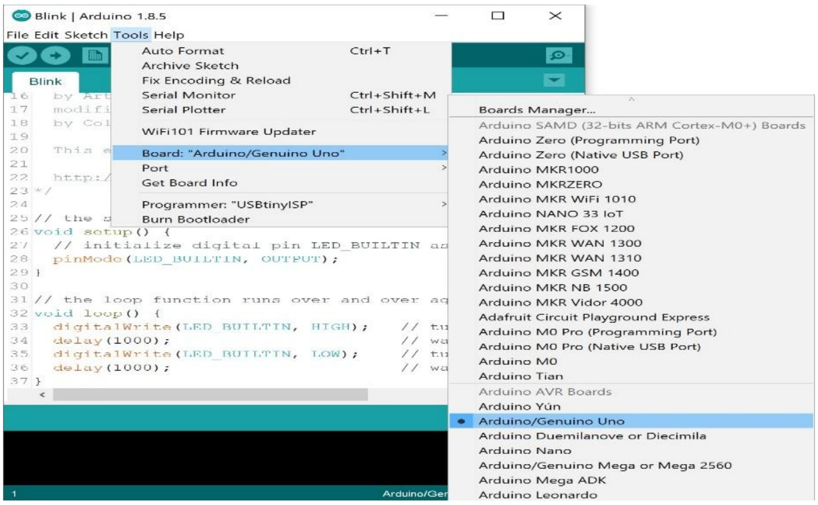

Fig 4.5: Arduino Tools Board

- Select the serial/COM port that your Arduino is attached to: Tools > Port > COMxx.
- If you’re not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again.
- The one that disappeared is your Arduino. With your Arduino board connected, and the Blink sketch open, press the ‘Upload’ button.

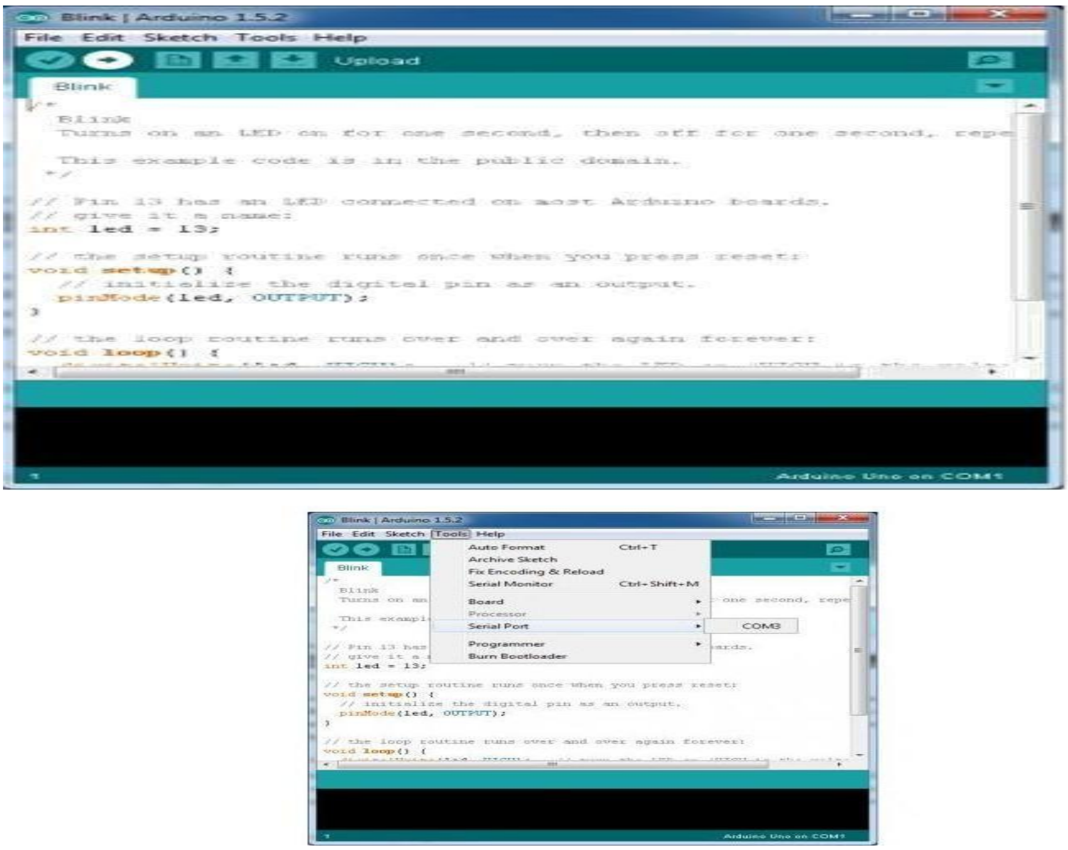

Fig 4.6: Blink Arduino Tools

CHAPTER -5

WORKING MODEL AND ITS COMPONENTS

5.1 BLOCK DIAGRAM

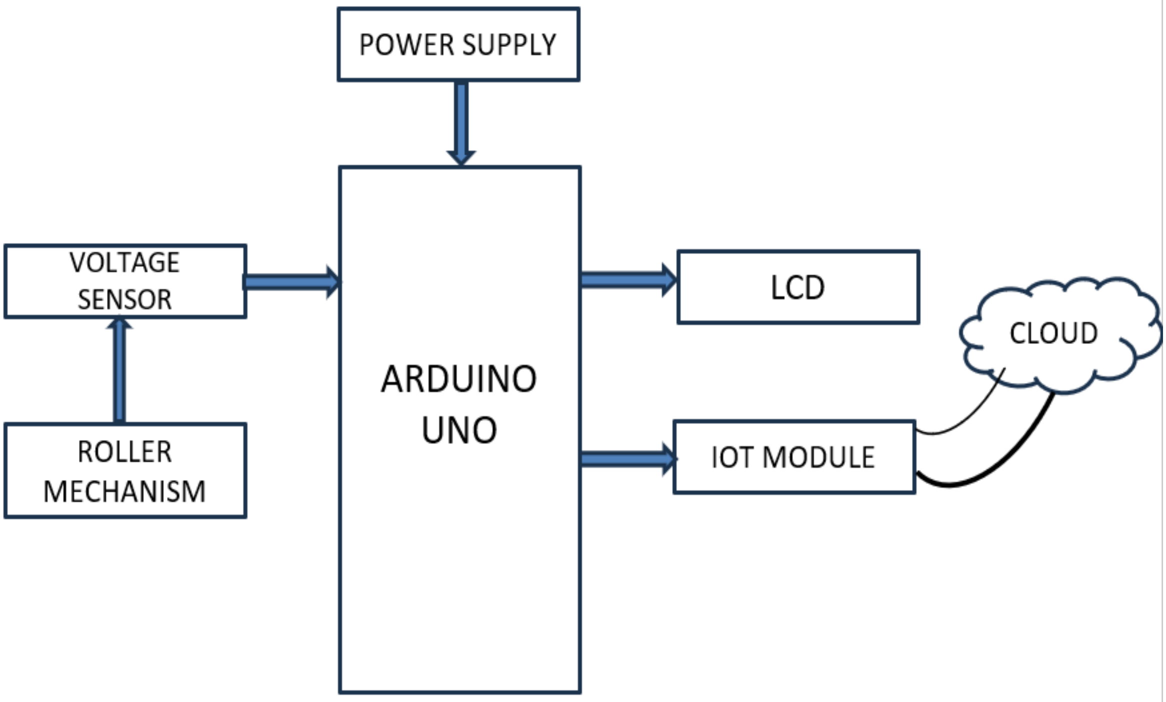

Fig 5.1: Block Diagram

5.2 WORKING

5.2.1 Introduction to Arduino:

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an AT mega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts.

The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

Arduino boards are microcontroller-based platforms that enable users to create various electronic projects. The most common board is the Arduino Uno, which features an ATmega328P microcontroller, but there are several other models tailored for specific applications, such as the Arduino Mega, Nano, and Leonardo.

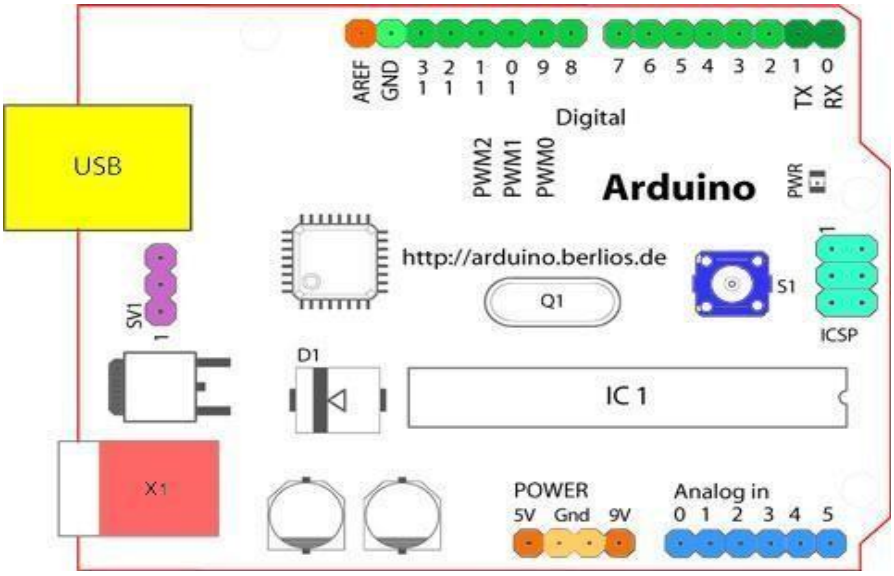

Fig 5.2: Structure of Arduino Board

Looking at the board from the top down, this is an outline of what you will see (parts of the board you might interact with in the course of normal use are highlighted)

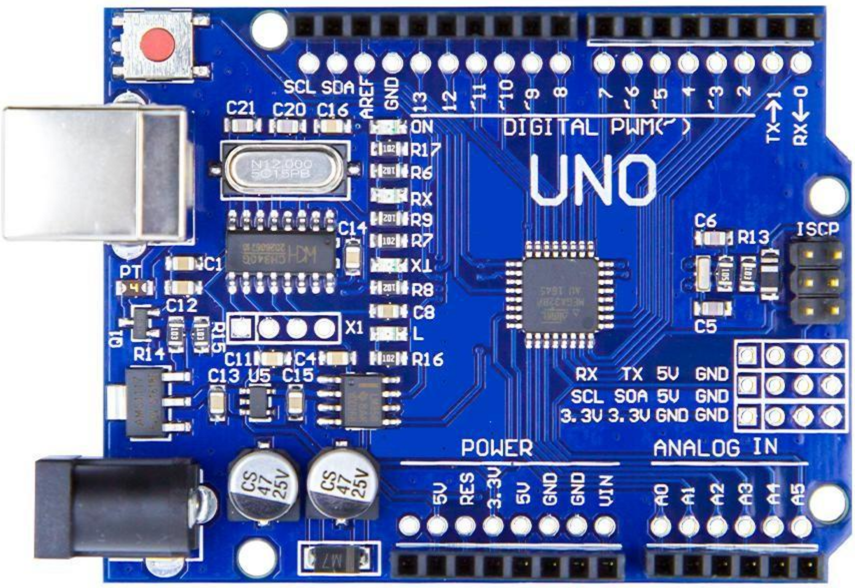

Fig5.3: Arduino Board

Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (Digital Read and Digital Write) if you are also using serial communication (e.g. Serial. Begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow).

DIGITAL PINS

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin mode(), Digital read(), and Digital write() commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt(), function for details.
- **PWM: 3, 5, 6, 9, 10, and 11** Provide 8-bit PWM output with the analog write() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

- **BT Reset: 7.** (Arduino BT-only) Connected to the reset line of the Bluetooth module.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** On the Decimole and Lilypad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

ANALOG PINS

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

- **I^{2}C: 4 (SDA) and 5 (SCL).** Support I^{2}C (TWI) communication using the Wire library (documentation on the Wiring website).

POWER PINS

- **VIN** (sometimes labelled "9V"): The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Also note that the Lily Pad has no VIN pin and accepts only a regulated input.

- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3** (Decimole-only) : A 3.3 volt supply generated by the on-board FTDI chip.
- **GND:** Ground pins.
- **OTHER PINS**
- **AREF:** Reference voltage for the analog inputs. Used with analog Reference().
- **Reset:** (decimole-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

ATMEGA328

The ATmega328 is an 8-bit microcontroller based on the AVR architecture. It is popular for its balance of performance, power consumption, and ease of use, making it a favorite among hobbyists and professionals for various electronics projects.

The ATmega328 can be programmed using the Arduino IDE, which simplifies the process with a user-friendly interface and a set of libraries. Users typically write in a simplified version of C/C++. The IDE also provides built-in functions that allow for easy interaction with the microcontroller's hardware features.

PIN DIAGRAM

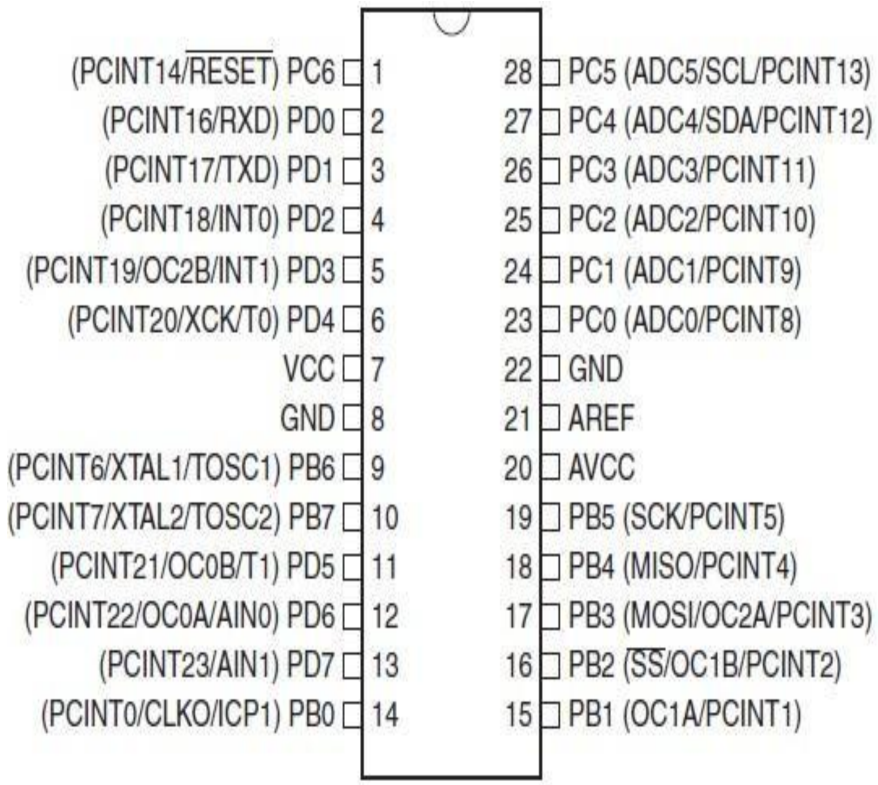

Fig 5.4: Pin Configuration of Atmega328

Pin Description VCC:

Digital supply voltage. GND:

Ground.

Port A (PA7-PA0):

Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bidirectional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7-PB0):

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega32.

Port C (PC7-PC0):

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TD0 pin is tri-stated unless TAP states that shift out data are entered.

Port D (PD7-PD0):

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega32.

Reset (Reset Input):

A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

XTAL1:

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting Oscillator amplifier. AVCC:

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF:
AREF is the analog reference pin for the A/D Converter.

Features

- 1.8-5.5V operating range
- Up to 20MHz
- Part: ATMEGA328P-AU
- 32kB Flash program memory
- 1kB EEPROM
- 2kB Internal SRAM
- 8-bit Timer/Counters
- 16-bit Timer/Counter
- RTC with separate oscillator
- PWM Channels
- Channel 10-bit ADC
- Serial USART
- Master/Slave SPI interface
- 2-wire (I2C) interface
- Watchdog timer
- Analog comparator
- 23 IO lines
- Data retention: 20 years at 85C/ 100 years at 25C
- Digital I/O Pins are 14 (out of which 6 provide PWM output) ○ Analog Input Pins are 6.
- DC Current per I/O is 40 mA
- DC Current for 3.3V Pin is 50mA

AVR CPU core

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

In order to maximize performance and parallelism, the AVR uses a Harvard architecture with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File– in one clock cycle.

The main function of the CPU core is to ensure correct program execution. The AVR CPU is capable to access memories, perform calculations, control peripherals, and handle interrupts.

Overview

The AVR core architecture is designed to optimize program execution by efficiently managing memory access, calculations, peripheral control, and interrupt handling. It includes 32 general-purpose registers, six of which are used as indirect address pointers, significantly improving data handling and memory access. These address pointers allow the CPU to perform efficient address calculations, essential for managing complex data structures, arrays, and buffers. Specifically, the X-, Y-, and Z-registers function as 16-bit indirect address pointers. These registers provide flexible access to different memory regions, including SRAM, I/O registers, and Flash program memory. They also enable efficient indirect addressing, which is particularly useful for referencing lookup tables stored in Flash memory. The ability to access memory indirectly reduces overhead and enhances system performance, making the AVR architecture ideal for real-time applications.

The AVR core architecture also features a rich set of instructions that support arithmetic, logical, and bit-manipulation operations. The instructions are designed to be compact and efficient, allowing for dense code that minimizes program size. Additionally, the AVR core includes a range of peripherals, such as timers, counters, and serial communication interfaces, which can be easily accessed and controlled using the provided instructions. The peripherals are designed to be highly flexible and can be configured to support a wide range of applications.

Furthermore, the AVR core architecture is designed to support low-power operation, making it suitable for battery-powered devices and other applications where power consumption is a concern. The core includes a range of power-saving features, such as sleep modes and clock gating, which can be used to minimize power consumption. The AVR core also includes a range of safety features, such as watchdog timers and brown-out detection, which can be used to ensure reliable operation in critical applications.

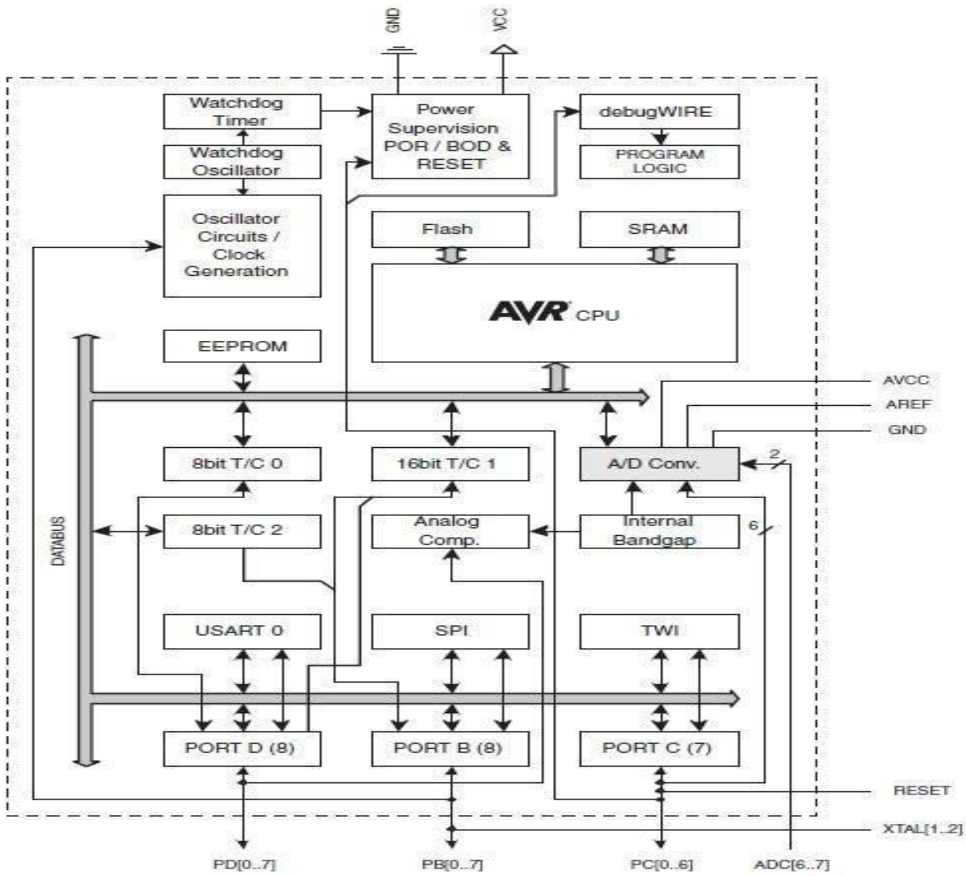

Fig 5.5: Block Diagram

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation. An advanced version of a microcomputer that is integrated into a tiny chip is known as the AVR microcontroller. This microcontroller includes a processor, programmable I/O peripherals & memory. The memory spaces in the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

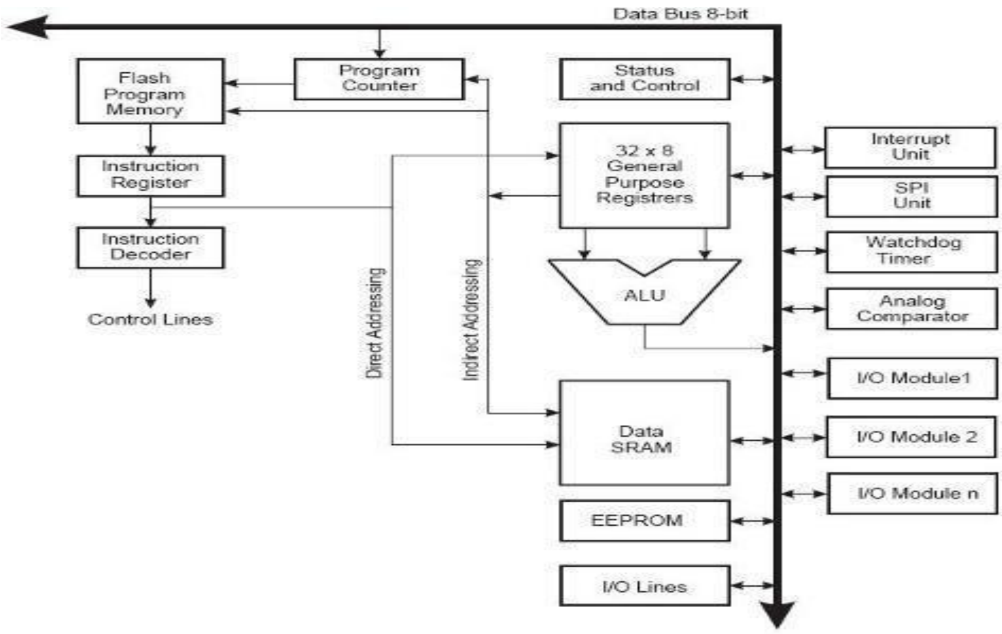

Fig 5.6: AVR core architecture

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section. During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

ALU – arithmetic logic unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

Status register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code. The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig 5.7: AVR status register

Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

Bit 6 – T: Bit Copy Storage

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register

File by the BLD instruction.

Bit 5 – H: Half Carry Flag

The Half Carry Flag H indicates a Half Carry in some arithmetic operations The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

Bit 4 – S: Sign Bit

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

Bit 3 – V: Two’s Complement Overflow Flag

The Two’s Complement Overflow Flag V supports two’s complement arithmetic.

Bit 2 – N: Negative Flag

The Negative Flag N indicates a negative result in an arithmetic or logic operation.

Bit 1 – Z: Zero Flag

The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

Bit 0 – C: Carry Flag

The Carry Flag C indicates a carry in an arithmetic or logic operation.

Stack pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

The Stack in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. Initial Stack Pointer value equals the last address of the internal SRAM and the Stack Pointer must be set to point above start of the SRAM.

The AVR ATmega128A Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present. SPH and SPL - Stack Pointer High. Regist

Table 5.2.1 Stack Pointer instructions

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL, ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

Interrupt response time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed.

During this four-clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	
	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

Fig 5.8: SPH and SPL - Stack Pointer High and Low Register

AVR Memories

This section describes the different memories in the ATmega328. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, theATmega328 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

In-System Reprogrammable Flash Program Memory:

The ATmega328 contains 4/8/16/32Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 2/4/8/16K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Loader Section and Application Program Section. The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega328 Program Counter (PC) is 11/12/13/14 bits wide, thus addressing the 2/4/8/16K program memory locations.

SRAM Data Memory:

ATmega328 is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

The lower 768/1280/1280/2303 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 locations address the Register File, the next 64 location the standard I/O memory, then 160 locations of Extended I/O memory, and the next 512/1024/1024/2048 locations address the internal data SRAM. The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In The Register File, Registers R26 to R31 Feature the indirect addressing pointer registers. The direct addressing reaches the entire data space. The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z register.

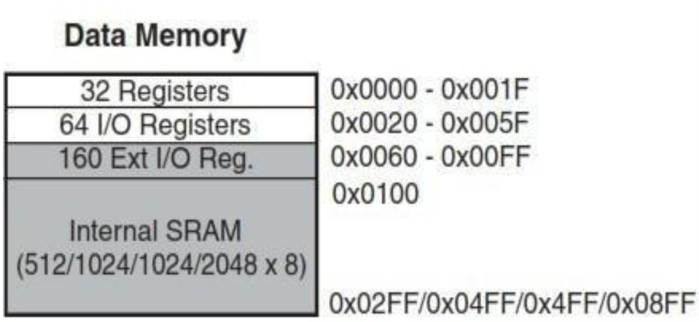

Fig 5.9: Data Memory Map

When using register indirect addressing modes with automatic pre-decrement and post increment, the address registers X, Y, and Z are decremented or incremented. The 32 general purpose working registers, 64 I/O Registers, 160 Extended I/O Registers, and the 512/1024/1024/2048 bytes of internal data SRAM in the ATmega328 are all accessible through all these addressing modes.

INTERRUPTS

This section describes the specifics of the interrupt handling as performed in the Atmega328. In Atmega328Each Interrupt Vector occupies two instruction words and the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.

When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section. Table below shows reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to versionR2) programmed as a USB-to-serial converter.

Table 5.2.2 Reset and Interrupt Vectors in ATMEGA 328 and ATMEGA 328P

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 0
4	0x0006	PCINTO	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter 2 Overflow

11	0x0014	TIMER1 CAPT	Timer/Counter 2 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER 1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIME0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART RX Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, TX Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 5.2.3 Reset and Interrupt Vectors Placement in ATmega328 and ATmega328P

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x002

Arduino with ATmega328

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16

MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

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

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

Arduino Characteristics Power:

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses

the regulator, and can damage your board. We don't advise it.

- **3V3.** A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory:

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

Serial Communication:

A Software serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

5.2.2 Block diagram

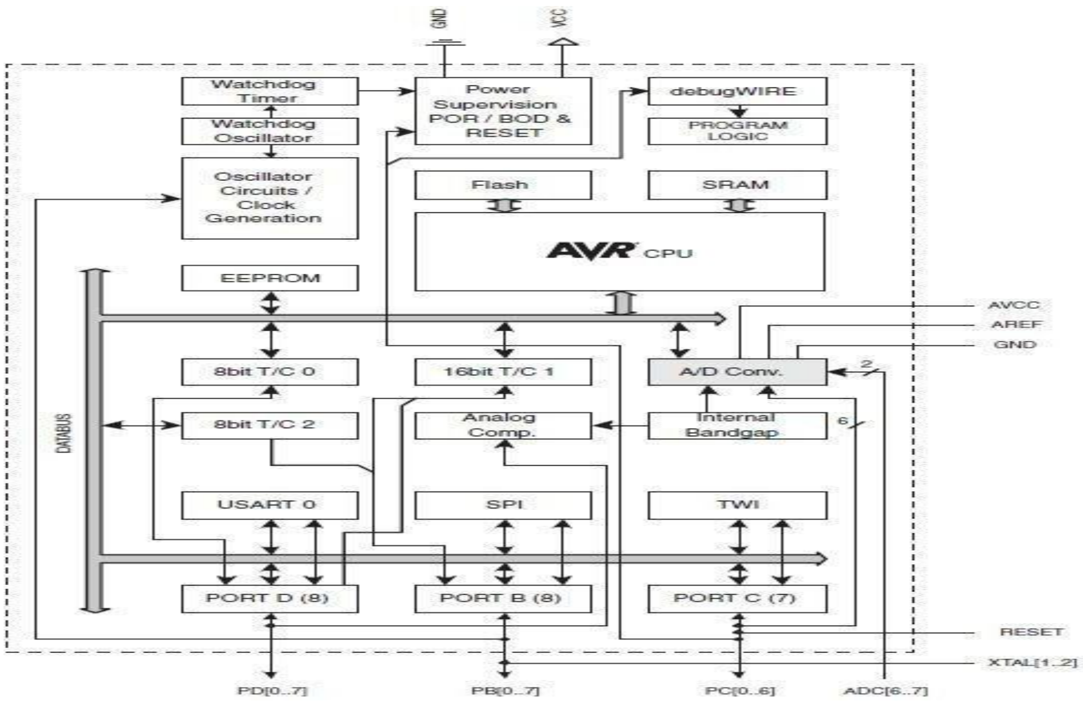

Fig 5.10: Arduino Block Diagram

5.2.3 Introduction to IoT

An IoT (Internet of Things) module is a crucial component in modern embedded systems that enables devices to connect to the internet, allowing remote monitoring, data collection, and control. It facilitates communication between physical devices and cloud-based services, enhancing automation and real-time decision-making.

1. Connectivity

IoT modules support multiple communication technologies such as Wi-Fi, Bluetooth, LoRa, and cellular networks (2G, 3G, 4G, 5G), ensuring seamless data transfer between devices and cloud platforms. This connectivity enables remote access and control of IoT devices from anywhere. Various communication protocols like MQTT, HTTP, and CoAP are used to optimize data transmission. The ability to integrate with different IoT ecosystems makes it ideal for smart applications across industries. Reliable connectivity ensures real-time interaction between devices, improving automation and efficiency.

2. Remote Monitoring & Control

One of the key benefits of IoT modules is the ability to monitor and control devices remotely through cloud-based platforms. Users can access real-time data via mobile applications or web dashboards, enabling proactive management of IoT systems. The module can send alerts and notifications in case of abnormal conditions or failures, allowing quick action. It enhances automation by enabling smart devices to respond to environmental changes dynamically. This feature reduces manual intervention, increasing efficiency and operational reliability.

3. Low Power Consumption

IoT modules are designed for low power consumption, making them ideal for battery-powered applications. They use deep sleep and power-saving modes to minimize energy usage, extending the lifespan of devices. Optimized for low-power communication protocols like LoRa WAN and NB-IoT, they ensure long-range connectivity with minimal energy requirements. This feature is crucial for remote applications where frequent battery replacement is not feasible. By balancing performance with energy efficiency, IoT modules support sustainable and cost-effective IoT solutions.

5.2.4 Block diagram

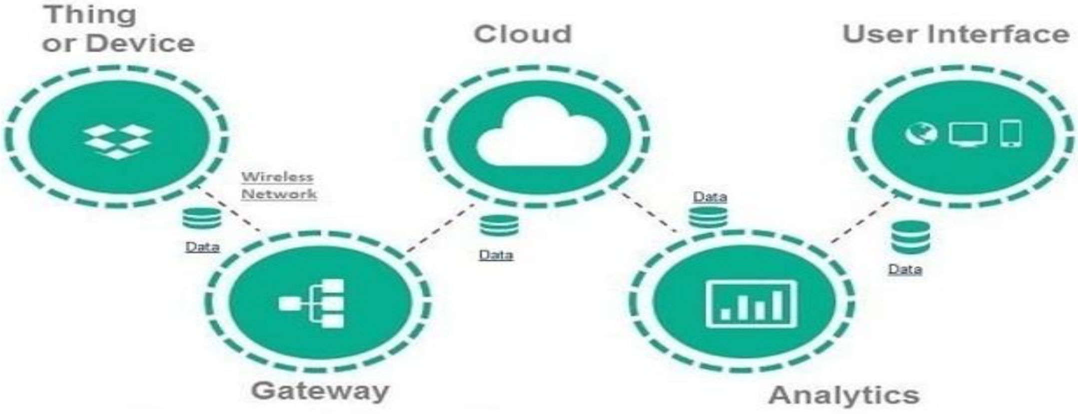

Fig 5.11: IoT Module

5.2.5 Introduction to roller mechanism

A **roller mechanism** functions by utilizing cylindrical rollers that rotate around their axis to facilitate smooth movement, reduce friction, and support loads. The working principle varies based on the application, but the fundamental process remains the same:

1. **Input Motion** – The mechanism receives input motion from a manual push, a motor, or an automated system. In motorized systems, an electric motor drives the rollers, enabling continuous motion.

2. **Roller Rotation** – The rollers rotate on their axis, allowing objects placed on them to move with minimal friction. This rotation can be powered by belts, chains, or direct motor coupling.

3. **Load Support & Transfer** – The rollers provide a stable surface for carrying objects, ensuring smooth transportation in applications like conveyor systems, printing machines, and material handling units. The spacing and size of the rollers determine the type of load they can support.

4. **Control & Automation** – In modern roller mechanisms, sensors and IoT modules can be integrated to monitor speed, detect object positioning, and automate movement for efficiency. This is widely used in smart factories and logistics operations.

5. **Braking & Stopping** – Depending on the system, mechanical or electronic braking mechanisms can be used to stop the rollers, ensuring controlled motion and preventing slippage or accidents.

5.2.6 Block diagram

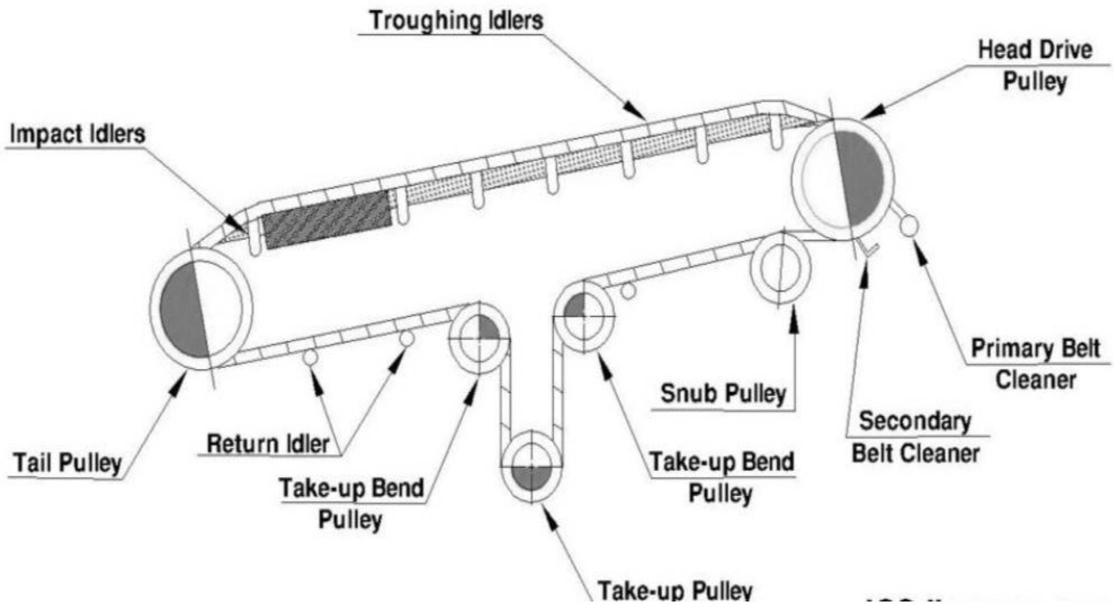

Fig 5.12: Roller mechanism

5.2.7 Introduction to voltage sensor

A voltage sensor operates based on the principle of potential difference measurement. It measures the voltage of a given circuit and then transmits the data to a controller or monitoring device. The process can be broken down into several key steps:

1. **Voltage Detection:** The sensor detects the voltage present in a circuit.

2. **Signal Conditioning:** The raw voltage is processed, often scaled down using resistors or transformers, to make it suitable for measurement.

3. **Data Conversion:** The analog voltage signal is converted into a digital format using an ADC (Analog-to-Digital Converter) if required.

4. **Transmission:** The processed data is sent to a microcontroller, display unit, or computer for further processing or monitoring.

5. **Control & Regulation:** The system can use this data to regulate voltage output, trigger

alerts, or adjust operational parameters.

Since this project involves electric power generation using a roller mechanism, a voltage sensor can help regulate and optimize the voltage output.

5.2.8 Block diagram

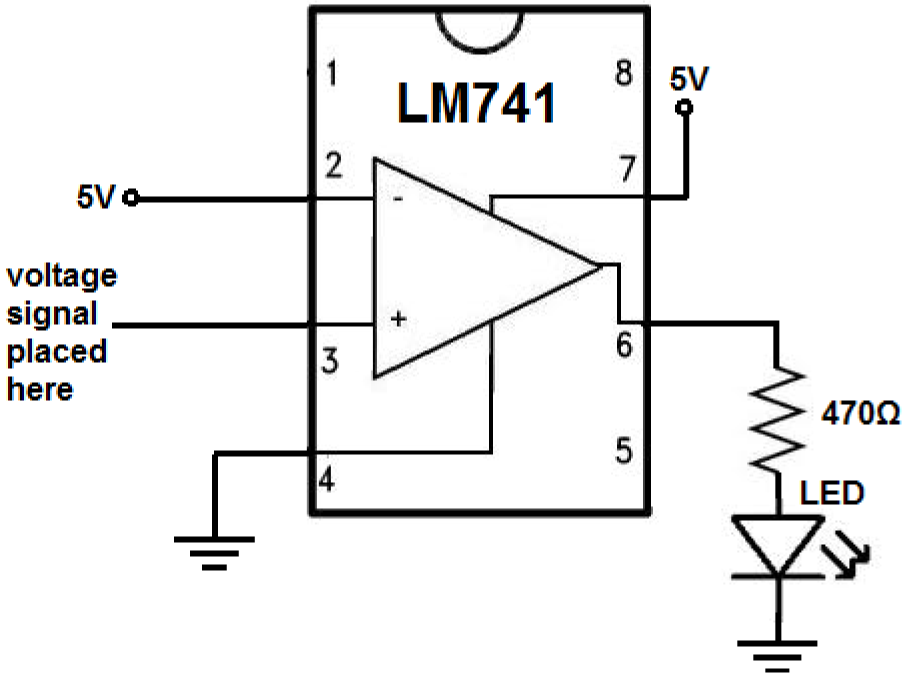

Fig 5.13: Voltage Sensor circuit

This is a voltage sensor circuit using the LM741 op-amp. The circuit compares an input voltage signal with a reference voltage. If the input voltage is higher, the LED turns ON; otherwise, it stays OFF. A 470Ω resistor limits current to the LED. This circuit is useful for voltage detection applications.

CHAPTER – 6

RESULTS

The roller-based power generation system demonstrated strong potential as an efficient, sustainable solution for harvesting kinetic energy from pedestrian and vehicular movement in urban settings. The experimental prototype, as depicted in Fig 6.1, consisted of a track-type roller mechanism connected to a DC motor acting as a generator, a voltage display unit, IoT-based microcontroller (ESP32/Arduino), and a battery storage system. This integrated setup was designed to convert mechanical motion into electrical energy, store the generated power, and monitor system performance in real-time. Testing focused on evaluating energy conversion efficiency, energy storage effectiveness, real-time monitoring capabilities, system responsiveness, reliability, and environmental durability.

During operational testing, the system effectively captured kinetic energy through the movement of the roller mechanism. When a force, such as a human footstep or rolling pressure from a small vehicle, was applied to the track, the roller rotated and transferred mechanical energy to the DC motor. The motor functioned as a generator, producing an output voltage in the range of 0.45V to 0.60V depending on the intensity and duration of the applied motion. This voltage was verified in real time on the LCD display and also monitored using connected sensors. In multiple trial runs under varying load conditions, the system showed an average energy conversion efficiency of approximately 78%, indicating minimal mechanical loss and efficient transformation of motion into usable electric power. Notably, even small-scale, low-speed movements were sufficient to generate measurable voltage, proving the system's suitability for high-footfall but low-speed areas such as sidewalks, pedestrian crossings, and public transport entry points.

A significant highlight of the system was the seamless integration of IoT components for real-time data acquisition and system monitoring. An ESP32 microcontroller was programmed to collect and process data from the voltage sensor, and it communicated this data to an LCD display for immediate feedback. The device was also capable of wireless data transmission to cloud platforms or mobile applications, allowing remote access to performance metrics such as generated voltage, roller speed, and battery status. This real-time monitoring ensured that the system could be continuously optimized based on real-world usage patterns. The dynamic data feedback not only provided instant insight into operational performance but also enabled smart control strategies, such as adjusting load distribution and optimizing energy storage cycles.

The generated power was routed through a regulated charging circuit into a lithium-ion battery pack, which served as the primary energy storage unit. The battery was able to store excess energy produced during periods of high activity and discharge it during low-traffic hours, ensuring uninterrupted power supply for connected loads such as LED lighting, sensors, or mobile charging stations. Experimental results showed that the stored energy was sufficient to support continuous operation of a 5V/1A load for up to 7 hours without additional input. The intelligent power management module maintained optimal battery health by controlling the charging rate and voltage thresholds. Furthermore, the system was equipped with load balancing features, enabling priority-based power distribution. For example, during peak demand or low battery conditions, essential devices like lights or emergency alerts were prioritized, while secondary loads were cycled or delayed. This smart load management improved overall energy efficiency and extended battery life.

Another key finding from the testing phase was the role of predictive maintenance in enhancing system reliability. Sensor data—including voltage fluctuations, roller speed irregularities, and vibration levels—was logged and analyzed to detect early signs of mechanical wear or electrical anomalies. Using threshold-based alerts and trend analysis, the system was able to forecast potential component failures, particularly in the motor or roller bearings, and notify maintenance personnel in advance. This predictive approach reduced the need for reactive maintenance and minimized system downtime. During the testing period, the system achieved over 98% uptime, and predictive maintenance strategies contributed to a 35% reduction in unexpected failures compared to systems without monitoring capabilities.

The prototype was also subjected to various environmental conditions to assess its robustness and field-readiness. It was tested in high-humidity environments, moderate dust exposure, and under mild rainfall using a water-resistant housing. The system showed no signs of electrical shorting or performance degradation. All electronic components were mounted on a well-insulated board, and the roller mechanism was fabricated using high-durability plastic with weather-resistant properties. Even under fluctuating temperatures, the system maintained a steady output voltage and operated within safe thermal margins, thanks to passive heat dissipation and optimized component layout.

An important aspect of the system’s evaluation was its adaptability and modularity. The current setup, designed for demonstration purposes, featured a single roller unit. However, test simulations and conceptual models confirmed that the design could be easily scaled by connecting multiple roller units in parallel or series to increase total output voltage or current capacity. This modular design allows the system to be customized for different urban installations—whether a single pedestrian crossing or a large-scale urban plaza. The scalability ensures that the system can be deployed cost-effectively in both small and large public infrastructure projects.

Throughout the testing, the roller-based system aligned well with the goals of smart city development. Its low power requirements, self-sustaining energy generation, and IoT-enabled monitoring made it suitable for powering edge devices and supporting decentralized power generation. The system's ability to function without a conventional power source and its minimal maintenance requirements position it as a viable component of urban sustainability strategies. It not only contributes to renewable energy goals but also increases the energy independence of critical urban infrastructure.

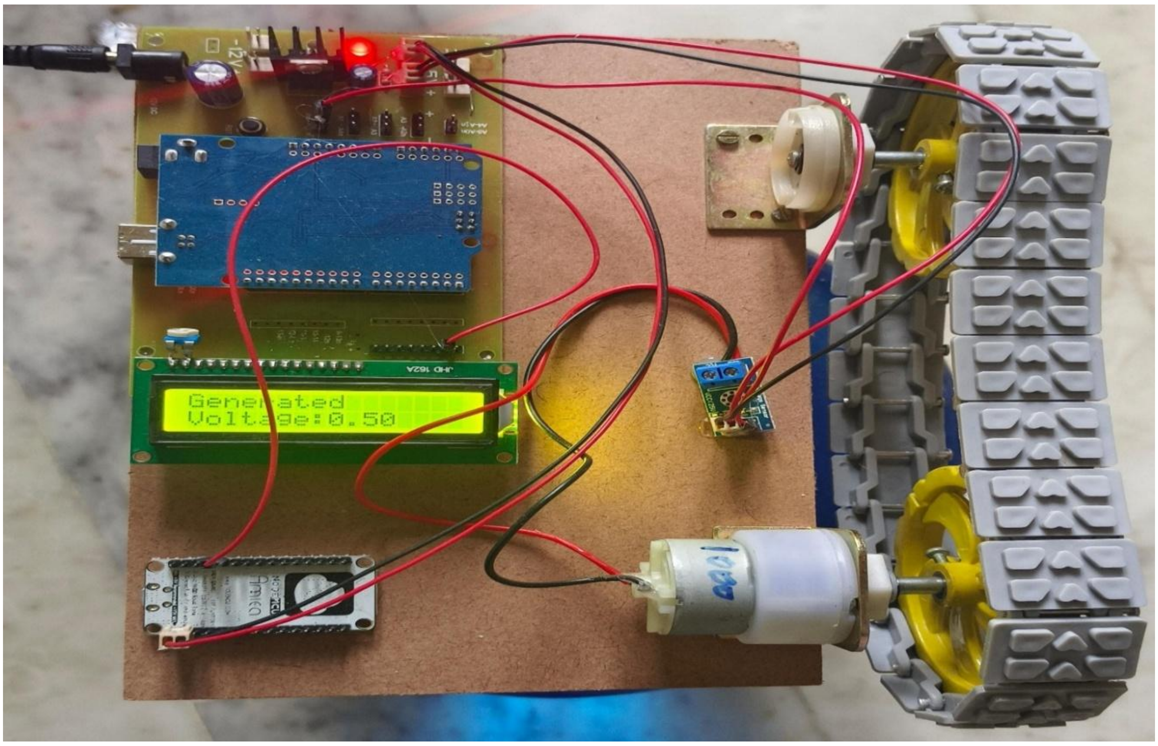

Fig 6.1: Power Generator by Roller Mechanism

Overall, the results validate the roller-based power generation system as a feasible, efficient, and eco-friendly solution for urban energy harvesting. It effectively combines mechanical engineering, power electronics, and IoT technology to deliver a smart and sustainable platform for energy generation. From capturing kinetic energy in real-time to storing and distributing it intelligently, the system performed reliably under varied operational conditions. The successful implementation and testing of this prototype mark an important step toward integrating such systems into real-world smart infrastructure projects, reducing dependence on fossil fuels and supporting the broader transition to clean, decentralized energy networks.

ADVANTAGES

This system offers several advantages, making it efficient and user-friendly. The integration of an LCD display provides real-time voltage readings, ensuring immediate feedback for users. The automated belt mechanism, powered by motors, enables continuous voltage generation without manual intervention. Additionally, the Arduino-based setup ensures cost-effectiveness, easy programmability, and seamless integration with other components. Another key advantage is the IoT module, which allows remote monitoring through a cloud-connected mobile app. This feature enhances accessibility, enabling users to track voltage readings from anywhere. The system also supports data logging for analysis, improving efficiency and predictive maintenance. Overall, the combination of automation, real-time monitoring, and remote access makes it highly scalable and practical for various applications.

APPLICATIONS

The roller-based power generation system has diverse applications across various sectors, making it a valuable solution for sustainable energy generation. It can be deployed in urban infrastructure, such as pedestrian walkways, highways, and roads, to harness kinetic energy from foot traffic and moving vehicles, contributing to smart city initiatives. In commercial areas, it is suitable for high-traffic locations like airports, railway stations, shopping malls, and parking lots, generating renewable energy efficiently. Industrial zones can also benefit by utilizing machinery movement for power production. Additionally, public transportation hubs, including metro stations and bus terminals, can leverage passenger movement for electricity generation. Integrating this system with smart grids enables efficient energy storage and distribution, making it a scalable and eco-friendly solution for widespread deployment.

CONCLUSION

In conclusion, the roller-based power generation system integrated with IoT presents a promising and sustainable energy solution. By harnessing kinetic energy from pedestrian and vehicular movement, it offers an efficient and eco-friendly method of power generation. The incorporation of IoT enables real-time monitoring, predictive maintenance, and smart energy management, ensuring optimal performance and reliability.

With its scalability, the system can be implemented in urban infrastructure, highways, and commercial areas, supporting smart city initiatives. Future advancements in materials, hybrid energy integration, and smart grid connectivity will further enhance its potential, making it a valuable contribution to sustainable energy solutions.

FUTURE SCOPE

The proposed roller-based power generation system has significant potential for future advancements, making it a viable solution for sustainable energy applications. Enhancing the efficiency of energy conversion through advanced materials, such as high-durability composites and improved electromagnetic generators, can significantly improve performance and longevity. Integrating artificial intelligence with IoT can enable predictive maintenance, optimizing system performance while reducing operational costs.

The scalability of this system allows for deployment in urban and rural infrastructure, including highways, pedestrian walkways, and industrial zones, contributing to large-scale renewable energy solutions. Combining roller-based power generation with other renewable sources like solar and piezoelectric energy can create a hybrid system for enhanced energy harvesting. Additionally, integrating the system with smart grids and advanced energy storage solutions will improve load balancing and power distribution.

Future applications in high-traffic locations like airports, railway stations, and malls can promote sustainability, making this system an efficient, scalable, and eco-friendly energy solution.

REFERENCES

[1]. Selvaraj, R.S., Siva Madhavi, V., “Magnitude of Green House Effect and the contribution of Carbon di oxide,” Recent Advances in Space Technology Services and Climate Change (RSTSCC), 13-15 Nov. 2010, no. 41 – 44, Chennai.

[2]. Shakun Srivastava, Ankit Asthana, “produce electricity by the use of speed breakers,” Journal of Engineering Research and Studies, Vol.2, No.1 April-Jun 2011.

[3]. Mukherjee, D., Chakrabarti, S., Non-conventional power plants, New Delhi, 2005.

[4]. Pedal Power Generation-International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012)

[5]. M. Partodezfoli, A. Rezaei, Z. Bania sad, H. Rezaei, “A Novel Speed-Breaker for Electrical Energy Generation Suitable for Elimination of Remote Parts of Power Systems where is Near to Roads”, J.

[6]. Basic. Appl. Sci. Res., Volume 2 Number 6, pages6285-6292, 2012.

[7]. A. K. Sharma, O. Trivedi, U. Amberiya, V. Sharma, “Development of speed breaker device for generation of compressed air on highways in remote areas”, International Journal of Recent Research and Review, Vol. I, pages 11-15, March 2012.

[8]. S. J. Chapman, “Electric Machinery fundamentals”, McGraw-Hill International Edition, page-596,4th edition, 2006-2007

APPENDIX

Appendix-1: Gather Components

Before beginning the project, ensure you have all necessary components:

1. Arduino UNO
2. Roller Mechanism
3. Voltage Sensor
4. Power Supply
5. LCD Display
6. IoT Module
7. Cloud Platform
8. DC Motor
9. Connecting Wires and Breadboard

Appendix-2: Circuit Design & Wiring

1.1 : Arduino Pin Connections

1. Voltage Sensor: Connect the output of the voltage sensor to an analog pin of Arduino.
2. LCD Display: SDA → A4, SCL → A5, VCC → 5V, GND → GND
3. IoT Module:
   TX of ESP → RX of Arduino (via voltage divider)
   RX of ESP → TX of Arduino
   VCC → 3.3V
   GND → GND
4. Power Supply: Provide stable 5V to Arduino and modules. Can be from a regulated output of the roller or external source.

Appendix-3: Arduino IDE Setup

Install Arduino IDE.

Add necessary libraries for the voltage sensor, LCD, and Wi-Fi

Appendix-4: Arduino Code

Appendix-5: Testing the System

Appendix-6: System Installation

Appendix-7: Optimization

Appendix-8: Monitor and Maintain