A

MINI PROJECT

REPORT

ON

"MICRO CONTROLLER BASED BANK LOCKER SECURITY SYSTRM WITH VOICE REPORTING AND RFID."

This dissertation Submitted in partial fulfillment of the requirements For the award of the degree of

BACHELOR OF TECHNOLOGY IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

VALLAB SATISH KUMAR	**19C71A0415**
MOHAMMED AYUB	**18C71A0442**
KARBHARI BAJRANG	**19C71A0401**

Under the guidance of

Mr.D.SHEKAR GOUD

(Asst. Prof)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ELLENKI COLLEGE OF ENGINEERING & TECHNOLOGY

PATELGUDA, AMEENPUR, SANGAREDDY DISTRICT - 502319

Affiliated to JNTU, Hyderabad

2022-2023

ELLENKI COLLEGE OF ENGINEERING & TECHNOLOGY

PATELGUDA, AMEENPUR, SANGAREDDY DISTRICT – 502319

Affiliated to JNTU Hyderabad.

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled **"MICRO CONTROLLER BASED BANK LOCKER SECURITY SYSTRM WITH VOICE REPORTING AND RFID"** being submitted by **VALLAB SATISH KUMAR , MOHAMMED AYUB & KARBHARI BAJRANG** bearing Roll No.**19C71A0415 , 18C71A0442 & 19C71A0401** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in **Electronics And Communication Engineering,** under **Jawaharlal Nehru Technological University,** Hyderabad, is a record of bonafide work carried out by them under by guidance and supervision. The results embodied in this Mini Project report have not been submitted to any other university or institute for the award of any degree.

INTERNAL GUIDE
Mr D.SHEKAR GOUD (Asst.Prof)

HEAD OF THE DEPARTMENT
Dr. K. SRINIVASULU (Prof)

EXTERNAL EXAMINER

PRINCIPAL
Dr. P.JOHN PAUL (Prof)

ACKNOWLEDGEMENT

We are very pleased present this thesis of our research. This period of our student life has been truly rewarding a number of people were of immense help to me during the course of our research and the preparation of our thesis.

We would like to thank **Sri E. SADASHIVA REDDY, Chairman, Ellenki Group of Institutions** for providing us the conducive environment for carrying through our academic schedules and Mini Projects with ease.

We express a whole hearted gratitude to **Prof. M. SAMBHA SHIVA REDDY, Director, Ellenki Group of Institutions and Academic Director Mr E. SAI KIRAN REDDY** for providing us the conducive environment for carrying through our academic schedules and Mini projects with ease.

We would like to thank our **Principal Dr. P. JOHN PAUL ,Ellenki College Of Engineering and Technology.** His insight during course of our research and regular guidance were invaluable to us and also for providing right suggestions at every phase of the development of our Mini Project.

We would like to thank our **Head of the Department Dr. K. SRINIVASULU in Electronics and Communication Engineering, Ellenki College of Engineering and Technology,** for providing seamless support and knowledge over the past one year and also for providing right suggestions at every phase of the development of our Mini Project.

We would like to thank our guide **Mr.D.SHEKAR GOUD, Assistant Professor, Department of Electronics and Communication Engineering, Ellenki College Of Engineering and Technology,** for providing seamless support and knowledge over the past one year and also for providing right suggestions at every phase of the development of our Mini Project.

We would like to thank all the staff members of **Department of Electronics and Communication Engineering.**

There is definitely a need to thank our friends and parents for their patience and support in always leading to successful completion of the Mini Project.

VALLAB SATISH KUMAR **19C71A0415**

MOHAMMED AYUB **18C71A0442**

KARBHARI BAJRANG **19C71A0401**

INDEX

ABSTRACT

This project is focused on effective recognizing and controlling system for Bank locker room which is fully self-determining. In cases of robberies, it's commonly happen that the banned entrance in the locker room area which can be detected by our security system. If the robbery take place the banks are not be capable to recognize the robber due to absence of the proof by using the current human operated security system. Development of various sensors has enabled systems to have preventive and corrective measures in this regard significantly. In order to deliver a concrete security solution for critically important and confidential documents and goods .

Bank security is an important thing as we people grow and achieve many things that are important for each individual, we need a proper security for such things as could be some important papers, jewelery, personal stuffs etc. The world has changed a lot from the old and mechanical stuffs to the modern electronics world. So as to keep up with the advancement many ideas are proposed to develop the security of the banks and one of the best ideas are using biometrics. We can use RFID and voice authentication as a base for this idea and we can develop more advanced security with the development in technology.

List of Figures :

List of Tables

CHAPTER 1

INTRODUCTION OF PROJECT

1.1 INTRODUCTION TO PROJECT

The bank locker system during olden days were using mechanical locks and were not secure. The technology is growing day by day and modern electronic locks were used to avoid robbery and unauthorized access.

This system is proposed to enhance the security of the present bank locker security system. As present system is vulnerable, for the hackers and thieves. The RFID in the system which is presently at use where the locker can be opened by any unauthorized person if they have the RFID tag which could be easily stolen or lost. So, to overcome this we are making use of the face and voice to open the locker in this system. Working of the proposed system takes places, where in the first step the camera which is interfaced with raspberry pi collects the data and stores it in the database by giving it a unique id to it, which should be matched to open the locker. Similarly, the voice recognition is done with google assistant that is interfaced with MCU. Hence, this system is highly secured and consumes low power for its operation, it would be best advancements that could be implemented in the bank lockers so that the security for the client's valuables would get increased.

1.2 INTRODUCTION TO EMBEDDED SYSTEMS

Each day, our lives become more dependent on 'embedded systems', digital information technology that is embedded in our environment. More than 98% of processors applied today are in embedded systems, and are no longer visible to the customer as computers' in the ordinary sense. An Embedded System is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs one or a few pre-defined tasks, usually with very specific requirements. Since the system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass- produced, benefiting from economies of scale. The increasing use of PC hardware is one of the most important developments in high-end embedded systems in recent years. Hardware costs of high-end systems have dropped dramatically as a result of this trend, making feasible some projects which previously would not have been done because of the high cost of non-PC-based embedded hardware. But

software choices for the embedded PC platform are not nearly as attractive as the hardware. Typically, an embedded system is housed on a single microprocessor board with the programs stored in ROM. Virtually all appliances that have a digital interface -- watches, microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system, but many are so specialized that the entire logic can be implemented as a single program. Physically, Embedded Systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip"), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine. As the embedded system is the combination of both software and hardware.

1.3 DEFINATION OF AN EMBEDDED SYSTEM

Embedded system is defined as, for a particular/specific application implementing the software code to interact directly with that particular hardware what we built. Software is used for providing features and flexibility.

Hardware = {Processors, ASICs, Memory...} is used for Performance & sometimes security. There are many definitions of embedded system but all of these can be combined into a single concept. An embedded system is a special purpose computer system that is used for particular task.

1.4 EXAMPLES OF AN EMBEDDED SYSTEMS

Embedded systems are found in wide range of application areas. Originally they were used only for expensive industrial control applications, but as technology brought down the cost of dedicated processors, they began to appear in moderately expensive applications such as automobiles, communication and office equipment and television Today's embedded systems are so inexpensive that they are used in almost every electronic product in our life. Embedded systems are often designed for mass production.

1. Automatic Teller Machines
2. Cellular telephone and telephone switches
3. Computer network equipment
4. Computer printers
5. Disk drives
6. Home automation products
7. Handheld calculators
8. Household appliances
9. Medical equipment
10. Measurement equipment
11. Multifunction wrist watches
12. Multifunction printers

1.5 HISTORY OF AN EMBEDDED SYSTEMS

The first recognizably modern embedded systems was the Apollo Guidance Computer, developed by "Charles Stark Draper" at the MIT Instrumentation Laboratory. At the project inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass produced embedded system was the Automatics D-17 guidance computer for the Minuteman missile, released in 1961.It was built from transistor logic and had a hard disk for main memory .When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high volume use of integrated circuits. This program alone reduced prices on quad and gate ICs from $1000/ each to $3/ each permitting their use in commercial products.

Since these early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality. The first microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required many external memory and support chips. In 1978 National Engineering Manufactures Association released a "standard" for programmable microcontrollers, including almost any computer based controllers, such as single board computers, numerical and event based controllers . Embedded Systems are designed to some specific task, rather than be a general-purpose computer for multitasks. Some also have realtime performances constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirement, allowing the system hardware to be simplified to reduce cost.

1.6 FEATURES OF AN EMBEDDED SYSTEM

The versatility of the embedded computer system lends itself to utility in all kinds of enterprises, from the simplification of deliverable products to a reduction in costs in their development and manufacture. Complex systems with rich functionality employ special Operating systems that take into account major characteristics of embedded systems. Embedded operating systems have minimized footprint and may follow real-time operating system specifics.

The special computers system is usually less powerful than general-purpose systems, although some expectations do exist where embedded systems are very powerful and complicated. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Many embedded systems use very small operating systems; most of these provide very limited operating system capabilities. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and

cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale. Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity. For high volume systems such as portable music players or mobile phones, minimizing cost is usually the primary design consideration. Engineers typically select hardware that is just "good enough" to implement the necessary functions. For low volume or prototype embedded systems, general purpose computers may be adapted by limiting the programs or by replacing the operating system with a real-time operating system.

1.7 CHARACTERSTICS OF AN EMBEDDED SYSTEM

Embedded computing systems generally exhibit rich functionality; complex functionality is usually the reason for introducing CPUs into the design. However, they also exhibit many non-functional requirements that make the task especially challenging:

- Real-time deadlines that will cause system failure if not met
- Multi-rate operation
- In many cases, low power consumption
- Low manufacturing cost, which often means limited code size.

Workstation programmers often concentrate on functionality. They may consider the performance characteristics of a few computational kernels of their software, but rarely analyze the total application. They almost never consider power consumption and manufacturing cost. The need to juggle all these requirements makes embedded system programming very challenging and is the reason why embedded system designers need to understand computer architecture.

1.8 OVERVIEW OF AN EMBEDDED SYSTEM ARCHITECTURE

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 as shown in Fig.

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. For small appliances such as remote-control units, air conditioners, toys etc., there is no need foran operating system and you can write only the software specific to that application.

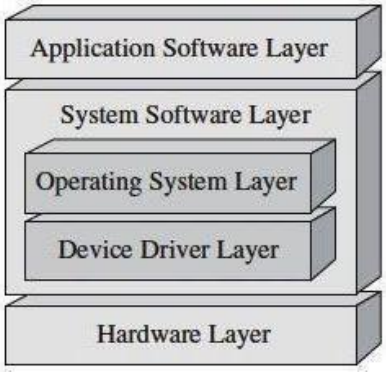


Fig 1.1 Layered Architecture of Embedded System

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 *for* a long time you don't need to reload new software.

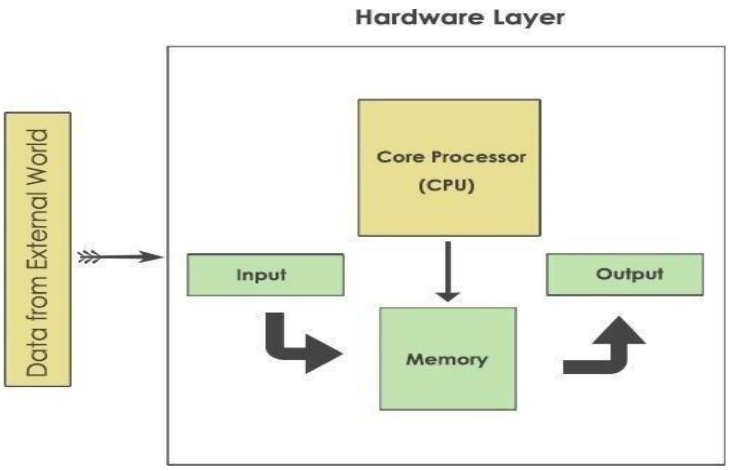


Fig 1.2 Block diagram of Embedded systems

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 interface
- 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 *for* user 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.

1.9 DEBUGGING OF AN EMBEDDED SYSTEMS

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging using the simple shell provided by the embedded operating system(e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operating using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or NEXUS interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in- circuit emulator (ICE) replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.

CHAPTER 2
LITERATURE REVIEW

2.1 EXISTING SYSTEM

Security is a defense against threats which provides an assurance of safety. Now and before security is one of the major concern in places like home, offices, institutions, laboratories etc. in order to keep our data confidentially so that no other unauthorized person could have access on them. In olden days the security mechanisms are less in order to prevent unauthorized access. Nowadays lot of security mechanisms have been introduced for such places and applications. But along with a wide variety of security methods, the techniques of theft are also changing and it's increasing day by day. With the available systems we can protect home and institutions to some extent. But that's not the case for critical places like military offices and scientific laboratory. These places require highly secure systems at every point of time in order to protect the valuable data and money. Varieties of security systems are now available such as password protected ones, RFID card technologies, biometric protected systems, OTP based, cryptography based and many more. Each system is applicable for different application zones depending upon their technical usage. Also there are systems that use a combination of any two techniques for more security, yet they don't provide a complete secure system as there is only single factor authentication. Also these systems can be broken by hackers or burglars. So these systems cannot be taken for the critical places that need more security. Here three techniques; RFID technology, encryptions and OTP which are the best techniques that incorporate with each other and the most effective ones is chosen to build our secure system. It doesn't mean that the other technologies are not worth. Based on the survey of different door lock access control mechanisms, the above three techniques that is used in this system is much better than the other ones.

Also these three methods are compatible with each other to produce a stronger system. Microcontroller PIC16F877A is used here which is a programmable device. High speed performance, programming flexibility and low-cost features are the high lights to choose microcontroller. They consist of 5 GPIO ports. However, only a single task can be executed to control a single system. This consists of several sections. Section 2 gives the literature review about the existing systems. Section 3 describes the methodology of the developed system. Section 4 presents the hardware implementation of the door lock. Section 5 describes the software part of the system which deals with encryption process. Section 6 introduces the android segment with the working of OTP technology and the secure wallet application. Section 7 provides the results and finally Section 8 concludes the overall work.

2.2 DISADVANTAGES OF EXISTING SYSTEM

Introduces the android segment with the working of OTP technology and the secure wallet application.

Here in upgradation we need not to worry about carrying mobile and OTP's

The software part of the system which deals with encryption process. With the new upgrade we reduce the software complexity .

2.3 PROPOSED SYSTEM

In this project some of the related works are done by connecting the RFID and voice with arduino for security purpose are illustrated. The RFID reader reads a data from tag and it also send to the microcontroller. If the card is authentic then display shows the eight digit number of the card read by the microcontroller. Then the authenticated person needs to enter into the locker, or by using the voice process the locker will open the door. Then the person enters into the locker room. This method is simple and more reliable than other system. This system consists of Microcontroller , RFID Reader, voice module, LCD, Servo motor. In this system the RFID reader reads the ID number from the passive tag and sends it to the microcontroller also we use voice module connected to Bluetooth. If the given inputs matches then the servo motor will open. If the given inputs doesn't matches, information is sent to the authenticated person and the system will remain in locked position until a prescribed input is given to the controller.

2.4 MOTIVATION

As an engineering student from electronics communications engineering, I just want to introduce technology in bank locker security system using micro controllers. This might sound simple but I just want to put my knowledge about the embedded systems in an ongoing application. This project mainly focuses on increasing security features in present existing technology. Bank lockers are plays important role in today's life and it is considered the safest place to store jewelry, documents, stock certificates and etc. The traditional methods used for most of banks rely on manual lock and PIN number/ password which are not fully secure. In manual lock systems, whenever a customer uses the locker, he/she should assist by the bank staff. This may lead to waste of time for both the customer and the staff. The major drawback of such manual locker systems are lack of security because the key can be duplicated. This would lead to theft of the entire valuable possessions in the bank lockers. The password, Personal Identification Number (PIN) or smart cards are used for personal identification to access lockers. Anyhow, the smart cards can be stolen, the password and PIN numbers can be forgotten or might be guessed. Still many banks struggle to prevent illegal access, intrusions and stopping secret information disclosure.

CHAPTER 3

BLOCK DIAGARM & SCHEMATIC OF THE PROJECT

3.1 BLOCK DIAGRAM

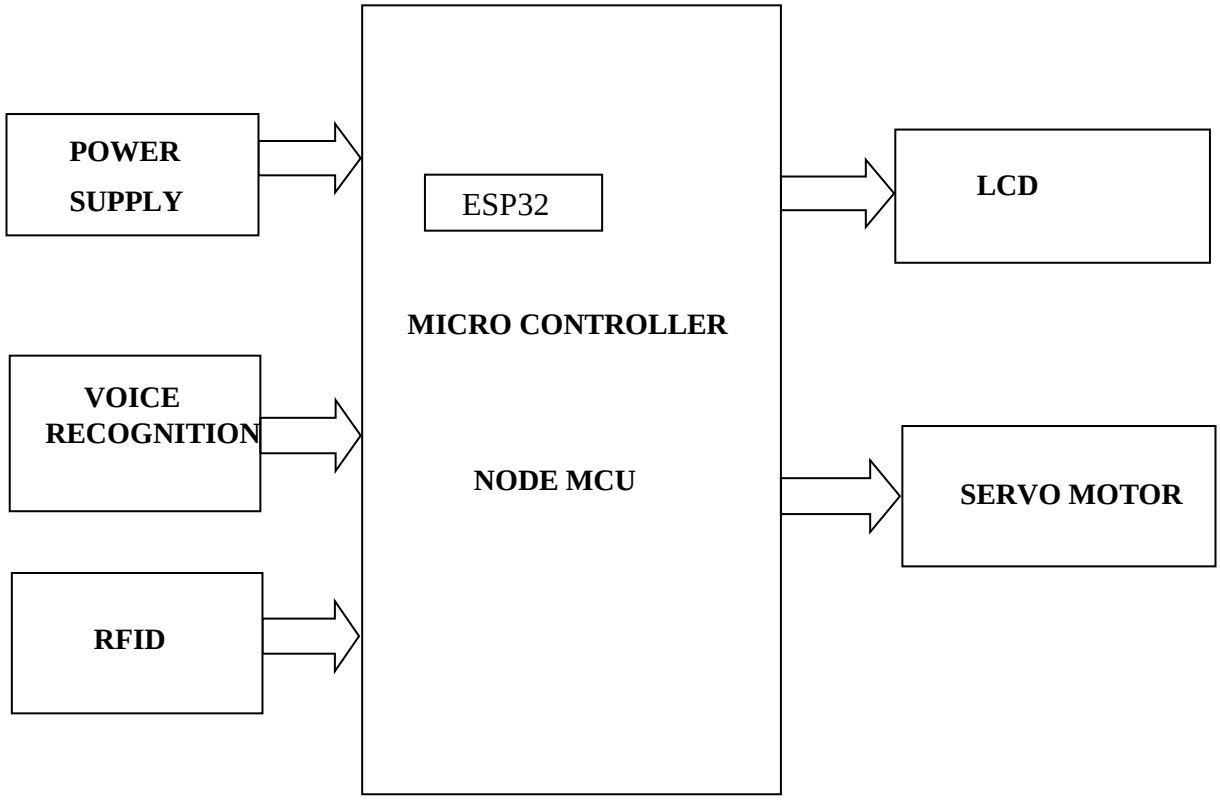


FIG 3.1 Block diagram of Micro Controller based Bank locker Security system with Voice Reporting and RFID

3.2 BLOCK DIAGRAM EXPLANATION

Initially first we do is power on i.e we give a power supply to the whole circuit. Then the power passes through the microcontroller and through that controller the power is supplied to all other components wisely . As we put the RFID Tag infront of the EM18 i.e RFID reader condition is placed. The reference number present in tag matches the number present in RFID reader which is already programmed then the servo motor gets open. if the reference number doesn't matches the number present in reader then the servo motor remains in closed position. The microcontroller (ESP32) is built with the Bluetooth and wifi. As we connect the wireless connection with the microcontroller with the Bluetooth Arduino then we can speak in the Bluetooth Arduino as we speak our voice gets analysed by the controller and what action is prescribed to it, it will get active with the voice we give. Then the servo motor will open as we speak in and gets closed in a given particular of time.

POWER SUPPLY

The DC power supply is practically converted to each and every stage in an electronic system. Thus a common requirement for all these phases will be the DC power supply. All low power system can be run with a battery. But, for a long time operating devices, batteries could prove to be costly and complicated. The best method used is in the form of an unregulated power supply –a combination of a transformer, rectifier and a filter.

VOICE RECIGNITION

ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its I2C / UART interfaces as it is In-Built. ESP32 has Bluetooth Low Energy generally reffered as BLE it is intended to provide considerably reduced power consumption. It's a Bluetooth 4 (LE) with supporting of 100 metre of long range and having frequency from 2.402 – 2.481 GHz. This Bluetooth Protocol is secured because the authentication of the device is done over a secure Bluetooth connection.

LCD

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

*The declining prices of LCDs.

*The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.

*Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.

Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

SERVO MOTOR

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor

IOT MODULE

An IoT module is a small electronic device embedded in objects, machines, and things connected to wireless networks and sends and receives data. Sometimes referred to as a "wireless module", "rf module" or "IoT chip," the IoT module contains the same technology and data circuits found in mobile phones but without features like a display or keypad.

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long - term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.

RFID READER

When RFID tag comes in range of signal transmitted by the reader, transponder in the tag is hit by this signal. A tag draws power from the electromagnetic field created by reader. Then, the transponder converts that radio signal into the usable power. After getting power, transponder sends all the information it has stored in it, such as unique ID to the RFID reader in the form of RF signal. Then, RFID reader puts this unique ID data in the form of byte on serial Tx (transmit) pin. This data can be used or accessed by PC or microcontroller serially using UART communication

3.3 FLOW CHART

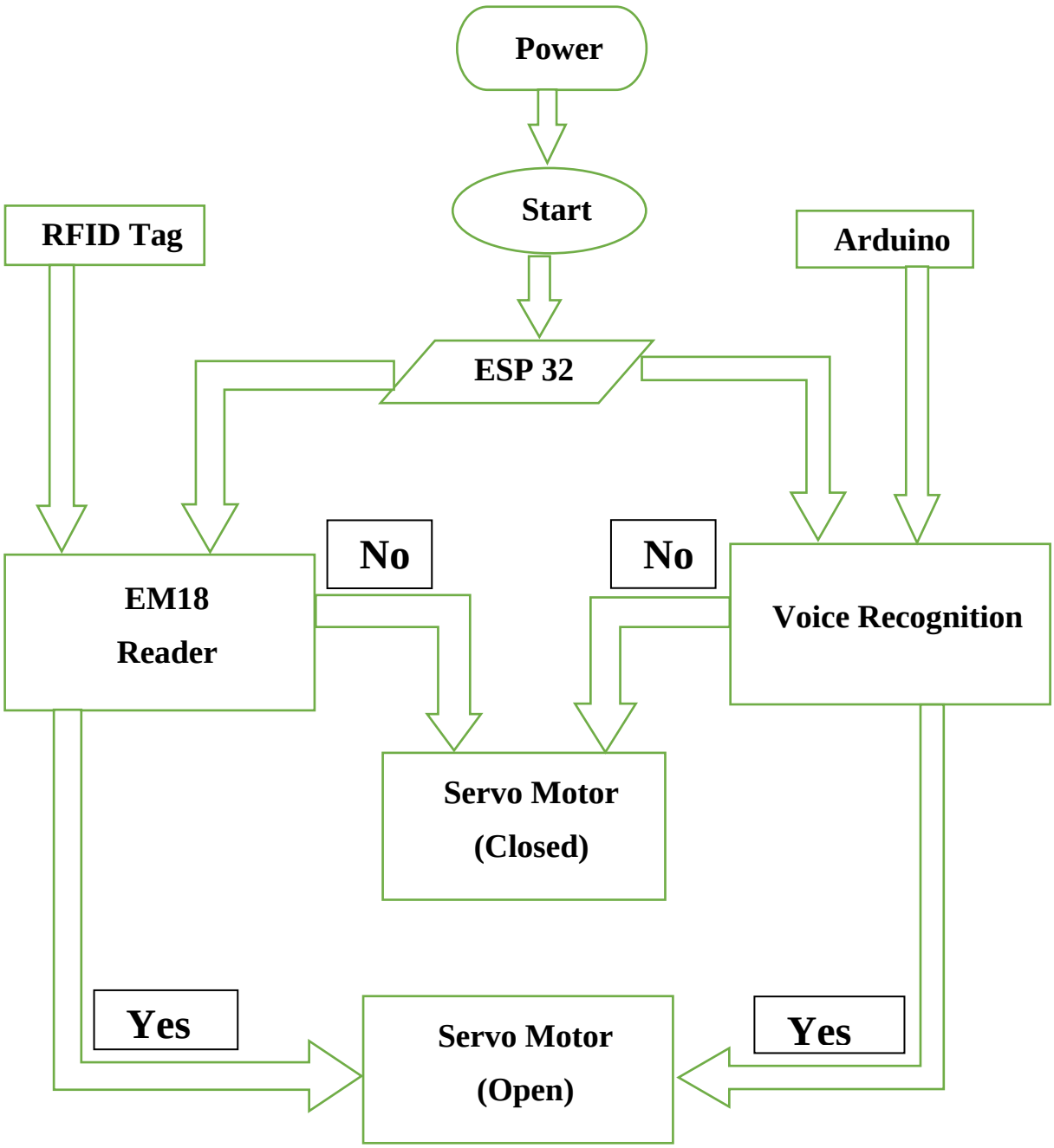


Fig 3.2 Flow Chart

3.4 FLOW CHART EXPLANATION :

Initially first step we do is power on i.e we give a power supply to the whole circuit. Then the power passes through the microcontroller and through that controller the power is supplied to all other components wisely . As we put the RFID Tag infront of the EM18 i.e RFID reader condition is placed. The reference number present in tag matches the number present in RFID reader which is already programmed then the servo motor gets open. if the reference number doesn't matches the number present in reader then the servo motor remains in closed position. The microcontroller (ESP32) is built with the Bluetooth and wifi. As we connect the wireless connection with the microcontroller with the Bluetooth Arduino then we can speak in the Bluetooth Arduino as we speak our voice gets analysed by the controller and what action is prescribed to it, it will get active with the voice we give. Then the servo motor will open as we speak in and gets closed in a given particular of time.

3.5 SCHEMATIC OF THE PROJECT

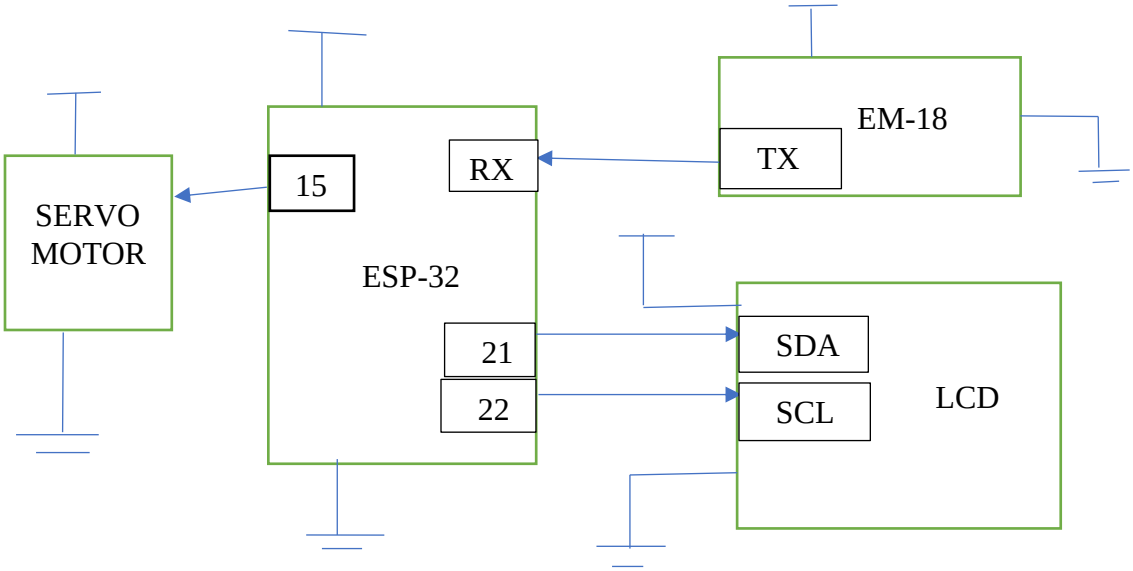


Fig 3.3 Schematic Of The Project

3.6 SCHEMATIC EXPLANATION :

The power suplly is given to the ESP32 Microcontroller, LCD, SERVO MOTOR and EM18 (RFID Reader) to the pins 5v/VCC. The transceiver pin of EM18 is connected to Receiver pin of ESP32. The G15 pin of ESP32 is connected to Servo Motor , G22 pin is connected to the SCL pin of LCD to transmit the data, also G21 is connected to SDA of LCD which is used to control the I2C bus lines. Also the Ground pins are connected to each component of the circuit. Now the Circuit is ready to use. Each component is ready to transmit / receive the data lines from the I2C bus lines pins . As we give the input to the circuit of each component we get a preferred output.

CHAPTER 4

NODE MCU

4.1 INTRODUCTION & FEATURES

The Internet of Things (IoT) has been a trending field in the world of technology. It haschanged the way we work. Physical objects and the digital world are connected now more thanever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) hasreleased an adorable,bite-sizedWiFi enabled micro-controller– ESP-32,atan unbelievableprice! For less than $3, it can monitor and control things from anywhere in the world – perfect forjustaboutanyIoTproject.

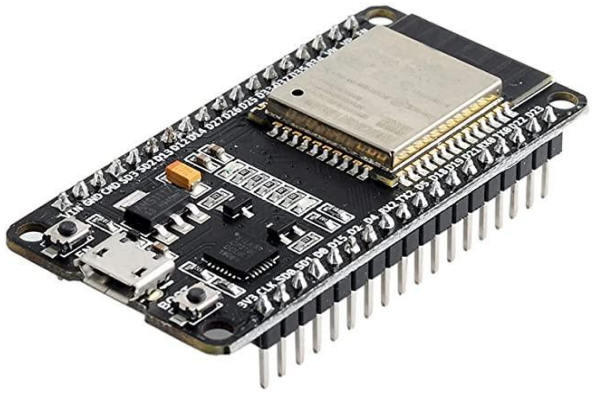


Fig:4.1 NodeMCUESP-32Micro-Controller

ESP-12EModule

The development board equips the ESP-12E module containing ESP-32 chip having TensilicaXtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clockfrequencyandsupportsRTOS.

ESP-32 DEVKIT V1ESP-32Specifications&Features

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
- OperatingVoltage: 3.3V
- InputVoltage:7-12V.

- 34 Programmable GPIO's.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 MBPS.
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- 520 KB of SRAM, 448 KB of ROM, Flash up-to 16 MB and 16 KB of RTC SRAM.
- ClockSpeed: 240MHz.
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- PCBAntenna.
- Built-in CP21XX USB to UART Serial Bridge.
- Wi-Fi 802.11 b/g/n/e/i.
- Bluetooth v4.2 BR/EDR and BLE.
- 2 x 8-bit DAC [D26][D25].
- 9 x touch sensors. Note: Touch Sensor 1 is [D0].
- 36 GPIO pins
- Internal temperature Sensor.

4.2 ARCHITECTURE

ESP-32ECHIP

- TENSILICAXTENSA32 BITLX106
- 240MHzClockFrequency
- 520 KBinternalRAM
- 448 KB of SROM.
- 16 MBExternalFlash
- 802.11/b/g/nWi-Fitransmitter.

Fig 4.2 ESP-32EChip

There's also 520 KB RAM and 16 MB of Flash memory (for program and data storage) just enoughtocopewiththelargestringsthatmakeupwebpages,JSON/XMLdata,andeverythingwethrowatIoTdevicesnowadays.

The ESP-32 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFinetwork and interact with the Internet, but it can also set up a network of its own, allowing otherdevicestoconnectdirectlyto it.ThismakestheESP-32ESP-32even moreversatile.

PowerRequirement

As the operating voltage range of ESP-32 is 3V to 3.6V, the board comes with a LDO voltageregulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should bemore than enough when ESP-32 pulls as much as 80mA during RF transmissions. The output ofthe regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can beusedtosupplypowertoexternalcomponents.

4.3 PINDIAGRAM

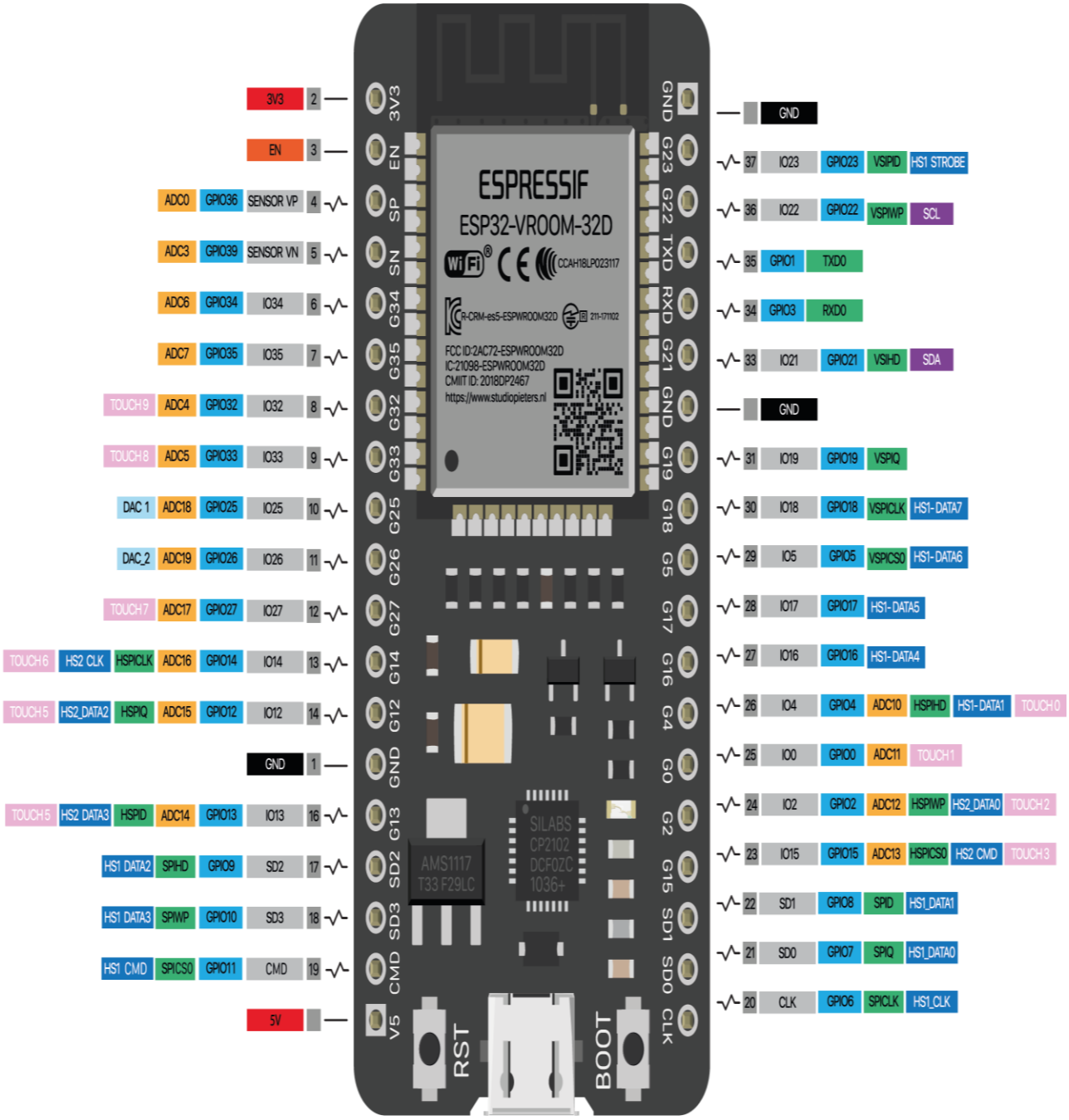


Fig.4.3 PinDiagram ofESP-32

E(

4.4 PIN DESCRIPTION

Input/output pins:

There are totally 39 digital Pins on the ESP32 out of which 34 can be used as GPIO and the remaining are input only pins. The device supports 18-channels for 12-bit ADC and 2-channel for 8-bit DAC. It also has 16 channels for PWM signal generation and 10 GPIO pins supports capacitive touch features. The ESP32 has multiplexing feature, this enables the programmer to configure any GPIO pin for PWM or other serial communication through program. The ESP32 supports 3 SPI Interface, 3 UART interface, 2 I2C interface, 2 I2S interface and also supports CAN protocol.

- **3 UART interface:** The ESP32 supports 3 UART interface for TTL communication. This would require 3 sets of Rx and Tx pins. All the 6 pins are software configurable and hence any GPIO pin can be programmed to be used for UART.
- **External Interrupt:** Again since the ESP32 supports multiplexing any GPIO pin can be programmed to be used as an interrupt pin.
- **GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS):** These pins are used for SPI communication. ESP32 supports two SPI, this is the first set.
- **GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS):** These pins are used for SPI communication. ESP32 supports two SPI, this is the second set.
- **GPIO21(SDA), GPIO22(SCL):** Used for IIC communication using Wire library.
- **Reset Pin:** The reset pin for ESP32 is the Enable (EN) pin. Making this pin LOW, resets the microcontroller.

ESP32 Pin-out Configuration

Pin Category	Pin Name	Details
Power	**Micro-USB, 3.3V, 5V, GND**	**Micro-USB:** ESP32 can be powered through USB port **5V:** Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board. **3.3V:** Regulated 3.3V can be supplied to this pin to power the board. **GND:** Ground pins.
Enable	**En**	The pin and the button resets the microcontroller.
Analog Pins	**ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9**	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC
DAC pins	**DAC1 and DAC2**	Used for Digital to analog Conversion
Input/Output Pins	**GPIO0 to GPIO39**	Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	**T0 to T9**	These 10 pins can be used as touch pins normally used for capacitive pads

RTC GPIO pins	**RTCIO0 to RTCIO17**	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be used to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM through the software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.
HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	**AREF**	To provide a reference voltage for input voltage.

Table: 4.1 ESP-32 Pin Configuration

ESP32 Technical Specifications

Microprocessor	TensilicaXtensa LX6
Maximum Operating Frequency	240MHz

Operating Voltage	3.3V
Analog Input Pins	12-bit, 18 Channel
DAC Pins	8-bit, 2 Channel
Digital I/O Pins	39 (of which 34 is normal GPIO pin)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
SRAM	520 KB
Communication	SPI(4), I2C(2), I2S(2), CAN, UART(3)
Wi-Fi	802.11 b/g/n
Bluetooth	V4.2 – Supports BLE and Classic Bluetooth

Table 4.2 ESP-32 technical Specification

4.5 SERIALCOMMUNICATION

The board includes CP2102 USB to UART Bridge Controller from silicon labs, which convertsUSB signal to serial and allows your computer to program and communicate with the ESP-32chip.

- CP2102USBtoUARTConverter.
- 4.5MBPScommunicationspeed.

- FlowControlSupport.

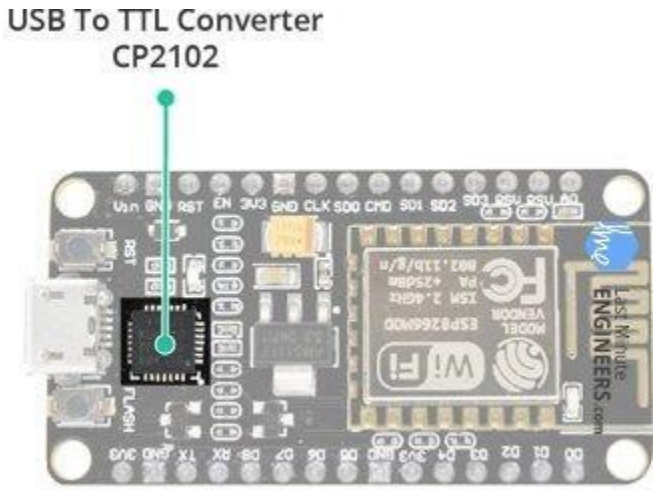


Fig:4.4 USB to TTL Converter

CHAPTER 5
FUNCTIONAL MODULES

5.1 RFID :

Fig 5.1 RFID Reader

When RFID tag comes in range of signal transmitted by the reader, transponder in the tag is hit by this signal. A tag draws power from the electromagnetic field created by reader. Then, the transponder converts that radio signal into the usable power. After getting power, transponder sends all the information it has stored in it, such as unique ID to the RFID reader in the form of RF signal. Then, RFID reader puts this unique ID data in the form of byte on serial Tx (transmit) pin. This data can be used or accessed by PC or microcontroller serially using UART communication.

Fig 5.2 RFID Tag and Tag Inside

RFID Tag:

RFID tag includes microchip with radio antenna mounted on substrate which carries 12 Byte unique Identification number.

RFID Reader:

It is used to read unique ID from RFID tags. Whenever RFID tags comes in range, RFID reader reads its unique ID and transmits it serially to the microcontroller or PC. RFID reader has transceiver and an antenna mounted on it. It is mostly fixed in stationary position.

Basically, RFID systems categorised as active and passive based on how they are powered and their range.

1. Active RFID system

Active RFID tags have their own transmitter and power source (Mostly battery operated). They operate at 455 MHz, 2.45 GHz, or 5.8 GHz, and they typically have a read range of 60 feet to 300 feet (20 meters to 100 meters).

2. Passive RFID system

Passive RFID tags do not have a transmitter, they simply reflect energy (radio waves) back coming from the RFID reader antenna. They operate in Low frequency (~125 KHz) as well as High frequency (~13 MHz) band and have limited read range of up to ~1m.

There are two modes of coupling for communication used in RFID as,

Inductive coupling, in which RFID reader emits magnetic field and whenever RFID tags enters the magnetic field which creates energy response from RFID tags and is detected by RFID reader. As the magnetic field drops sharply with distance, it is used for short distance applications.

Capacitive coupling, where reader emits electromagnetic waves and whenever that waves encounters RFID tags, RFID tag reflects signal containing information of ID.

RFID Reader has transceiver which generates a radio signal and transmits it through antenna. This signal itself is in the form of energy which is used to activate and power the tag.

When RFID tag comes in range of signal transmitted by the reader, transponder in the tag is hit by this signal. A tag draws power from the electromagnetic field created by reader. Then, the transponder converts that radio signal into the usable power. After getting power, transponder sends all the information it has stored in it, such as unique ID to the RFID reader in the form of RF signal. Then, RFID reader puts this unique ID data in the form of byte on serial Tx (transmit) pin. This data can be used or accessed by PC or microcontroller serially using UART communication.

5.2 VOICE REPORTING

Fig 5.3 Voice Reporting ESP-32 Bluetooth

Voice Reporting : ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces as it is In-Built. ESP32 has Bluetooth Low Energy generally reffered as BLE it is intended to provide considerably reduced power consumption. It's a Bluetooth 4 (LE) with supporting of 100 metre of long range and having frequency from 2.402 – 2.481 GHz. This Bluetooth Protocol is secured because the authentication of the device is done over a secure Bluetooth connection.

ESP32 Bluetooth uses :

- Medical devices for monitoring and reporting.
- Sports and fitness devices**.**
- Industrial monitoring sensors.
- Home automation.
- Geo-based, targeted promotions via beacons.
- Public transportation apps.
- Remote controls.
- PC peripheral devices like wireless mouse and keyboard.

5.3 LCD

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because Of the following reasons: The declining prices of LCDs.

Fig 5.4 LCD

The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.

Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.

Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

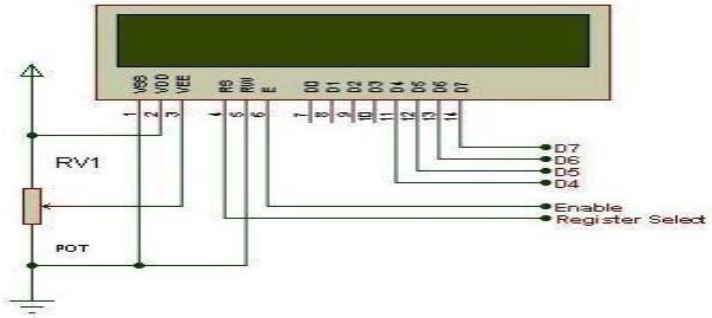


Fig 5.5 Pin Connections of LCD

A model described here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 microcontroller (Hitachi) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical

symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic message on display (shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

LCD CONNECTIONS

Depending on how many lines are used for connection to the microcontroller, there are 8bit and 4bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called initialization. In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected.

Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from microcontroller to LCD) one more I/O pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. EvenEven though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

PIN CONFIGURATION OF LIQUID CRYSTAL DISPLAY

PinNo	Pin Name	Description
1	Vss (Ground)	Ground pin connected to system ground
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shift between command/data register
5	Read/Write	Used to read or write data. Normally grounded to write data to LCD
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement
7	Data Pin 0	Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free .
15	LED Positive	Backlight LED pin positive terminal
16	LEDNegative	Backlight LED pin negative terminal

Table 5.1 Pin description table of 16*2 LCD display

Code (hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Shift cursor to left
5	Shift display right
6	Shift cursor to right
7	Shift display left
8	Display off, Cursor off
A	Display off, Cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Table 5.2 Basic commands of LCD

5.4 SERVO MOTOR

Fig 5.6 Servo Motor

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

5.5 IOT MODULE

Fig 5.7 Node MCU

An IoT module is a small electronic device embedded in objects, machines, and things connected to wireless networks that sends and receives data. Sometimes referred to as a "wireless module", "rf module" or "IoT chip," the IoT module contains the same technology and data circuits found in mobile phones but without features like a display or keypad. The INTERNET OF THINGS **(IoT)** is a set of technologies that uses sensors and actuators to inform us about the status of everyday items such as vehicles, tools and even living beings. It allows us to interact with them, enabling connectivity with platforms in the cloud that receive and process information for posterior analysis. This analyzed data is then used to make decisions.

5.6 POWER SUPPLY

The DC power supply is practically converted to each and every stage in an electronic system. Thus a common requirement for all these phases will be the DC power supply. All low power system can be run with a battery. But, for a long time operating devices, batteries could prove to be costly and complicated. The best method used is in the form of an unregulated power supply –a combination of a transformer, rectifier and a filter. All devices will have a certain power supply limit and the electronic circuits inside these devices must be able to supply a constant DC voltage within this limit. This DC supply is regulated and limited in terms of voltage and current. But the supply provided from mains may be fluctuating and could easily break down the electronic equipment, if not properly limited. This work of converting an unregulated alternating current (AC) or voltage to a limited Direct current (DC) or voltage to make the output constant regardless of the fluctuations in input, is done by a regulated power supply circuit.

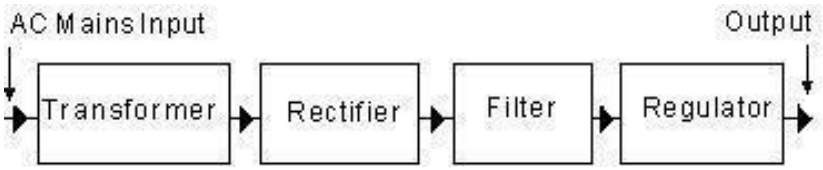


Fig 5.8 Block diagram of Regulated power supply

All the active and passive electronic devices will have a certain DC operating point (Q-point or Quiescent point), and this point must be achieved by the source of DC power. A step down transformer is used to reduce the voltage level to the devices needs. In India, a 1 Ø supply is available at 230 volts. The output of the transformer is a pulsating sinusoidal AC voltage, which is converted to pulsating DC with the help of a rectifier. This output is given to a filter circuit which reduces the AC ripples, and passes the DC components. But here are certain disadvantages in using an unregulated power supply. Regulated power supply is an electronic circuit that is designed to provide a constant dc voltage of predetermined value across load terminals irrespective of ac mains fluctuations or load variations. A regulated power supply essentially consists of an ordinary power supply and a voltage regulating device, as illustrated in the figure. The output from an ordinary power supply is fed to the voltage regulating device that provides the final output. The output voltage remains constant irrespective of variations in the ac input voltage or variations in output (or load) current.

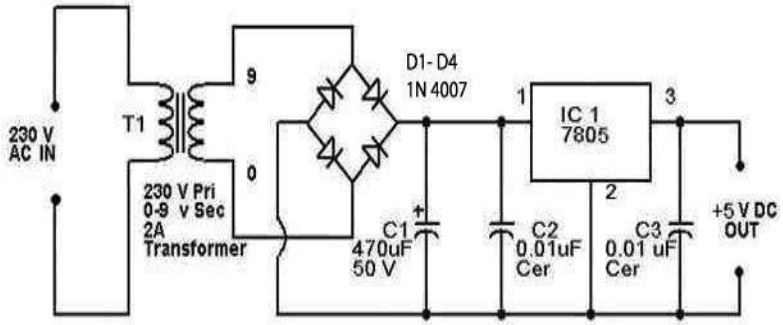


Fig 5.9 RPS Circuit Diagram

CHAPTER 6

SOFTWARE DISCRIPTION AND SOURCE CODE

6.1 INTRODUCTION OF ARDUINO IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

6.2 KEY FEATURES OF ARDUINO IDE

Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.

You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.

Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

Finally, Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program onthe Arduino board.

6.3 ARDUINO DATATYPES

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use During Arduino programming.

Void

The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

Example:

```
Void Loop ( )
{
// rest of the code
}
```

Boolean

A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Example:

Boolean state= false ; // declaration of variable with type boolean and initialize it with false. Boolean state = true ; // declaration of variable with type boolean and initialize it with false.

Char

A data type that takes up one byte of memory that stores a character value. Character literals are written in single quotes like this: 'A' and for multiple characters, strings use double quotes: "ABC".

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in whichthe ASCII value of the character is used. For example, 'A' + 1 has the value 66, since theASCII value of the capital letter A is 65.

Example:

Char chr_a = ‘a’ ;//declaration of variable with type char and initialize it with character a. Char chr_c = 97 ;//declaration of variable with type char and initialize it with character 97

Unsigned char

Unsigned char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

Example:

Unsigned Char chr_y = 121 ; // declaration of variable with type Unsigned char and initialize it with character y

Byte

A byte stores an 8-bit unsigned number, from 0 to 255.

Example: byte m = 25 ;//declaration of variable with type byte and initialize it with 25

int

Integers are the primary data-type for number storage. int stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767

(minimum value of -2^15 and a maximum value of (2^15) - 1).

The int size varies from board to board. On the Arduino Due, for example, an int stores a 32-bit (4-byte) value. This yields a range of -2,147,483,648 to 2,147,483,647 (minimum value of -2^31 and a maximum value of (2^31) - 1).

Example:int counter = 32 ;// declaration of variable with type int and initialize it with 32.

Unsigned int

Unsigned int's (unsigned integers) are the same as int in the way that they store a 2 byte value. Instead of storing negative numbers, however, they only store positive values, yielding a useful range of 0 to 65,535 (2^16) - 1). The Due stores a 4 byte (32-bit) value, ranging from 0 to 4,294,967,295 (2^32 - 1).

Example: Unsigned int counter= 60 ; // declaration of variable with type unsigned int and initialize it with 60.

Word

On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number. On the Due and Zero, it stores a 32-bit unsigned number.

Example word w = 1000 ;//declaration of variable with type word and initialize it with 1000.

Long Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from 2,147,483,648 to 2,147,483,647.

Example : Long velocity= 102346 ;//declaration of variable with type Long and initialize it with 102346

Unsigned long

Unsigned long variables are extended size variables for number storage and store 32 bits (4 bytes). Unlike standard longs, unsigned longs will not store negative numbers, making their range from 0 to 4,294,967,295 (2^32 - 1).

Example:

Unsigned Long velocity = 101006 ;// declaration of variable with type Unsigned Long and initialize it with 101006.

Short

A short is a 16-bit data-type. On all Arduinos (ATMega and ARM based), a short stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

Example: short val= 13 ;//declaration of variable with type short and initialize it with 13

Float

Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values because they have greater resolution than integers.Floating-point numbers can be as large as 3.4028235E+38 and as low as 3.4028235E+38.

They are stored as 32 bits (4 bytes) of information.

Example:

float num = 1.352;//declaration of variable with type float and initialize it with1.352.

Double

On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision

Example:

double num = 45.352 ;// declaration of variable with type double and initialize it with 45.352

.

6.4 STEPS TO UPLOAD THE PROGRAMM IN ARDUINO BOARD

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) anda USB cable.

In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind youwould connect to a USB printer as shown in the following image.

Fig 6.1 USB Cable

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

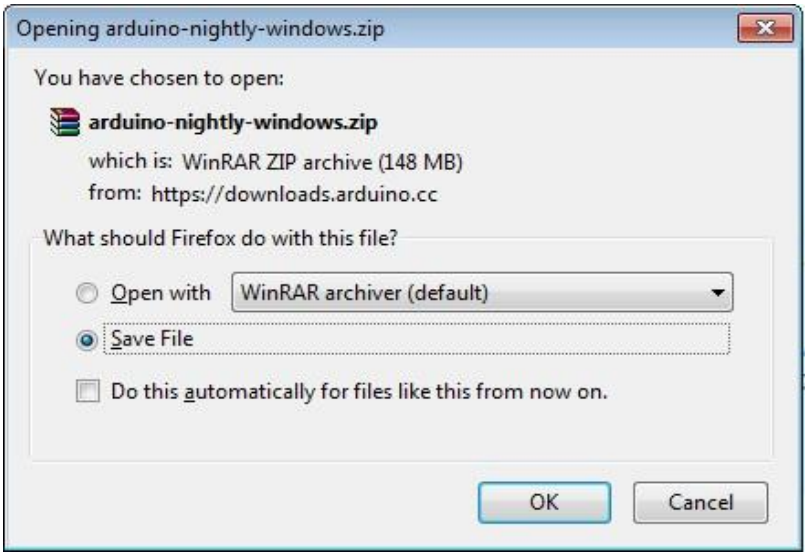


Fig 6.2 Opening Arduino

Step 4: Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Doubleclick the icon to start the IDE.

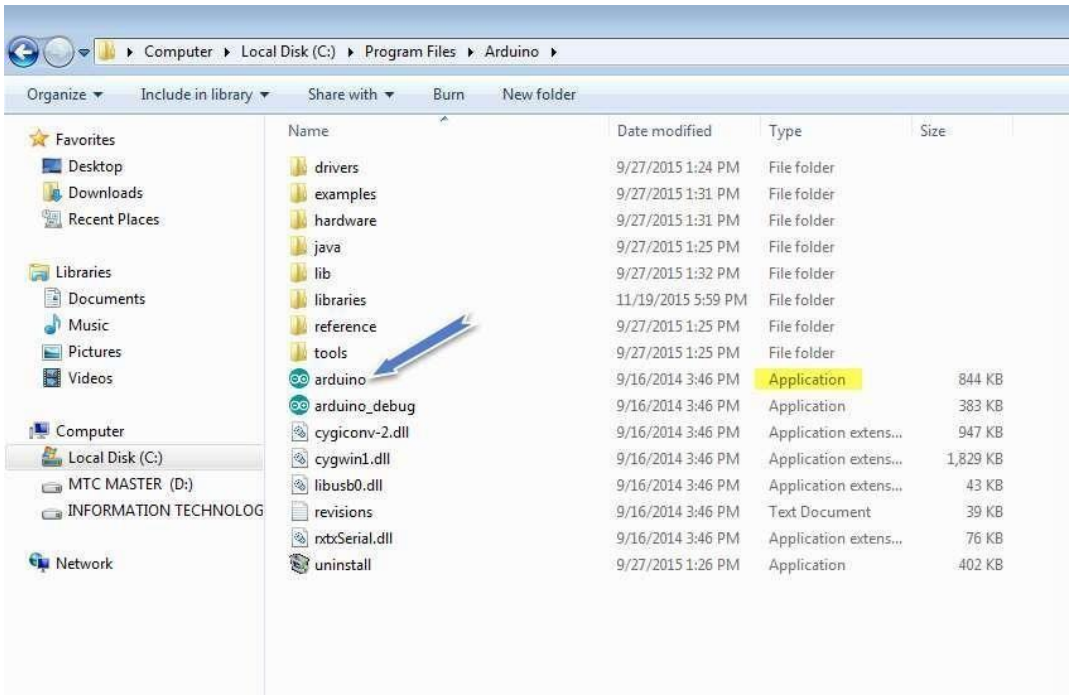


Fig 6.3 LAUNCHING ARDUINO IDE

Step 5: Open your first project.

Once the software starts, you have two options: Create a new project.

Open an existing project example.

To create a new project, select File --> New.To open

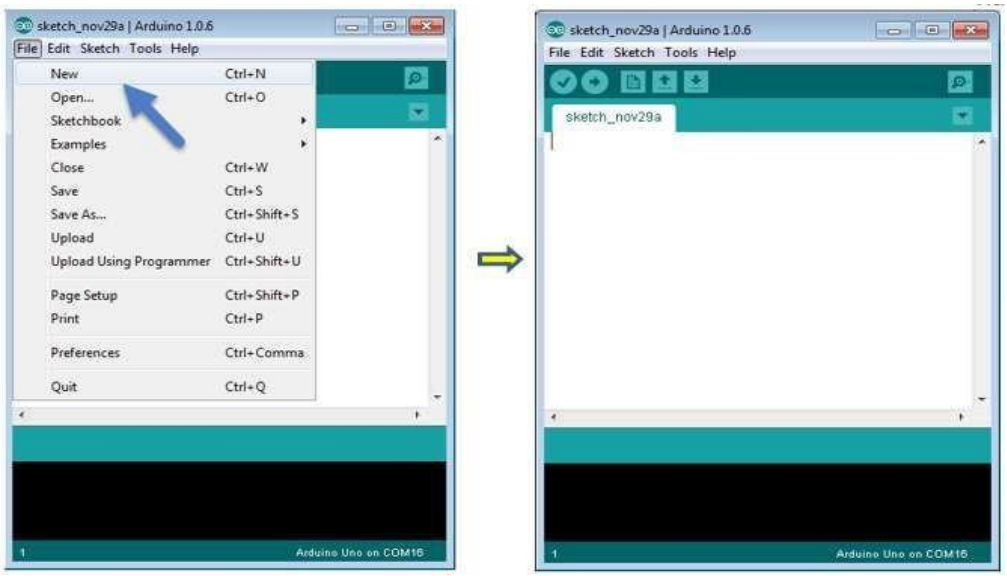


Fig 6.4 CREATING A PROJECT

To open an existing project example, select File -> Example -> Basics -> Blink.

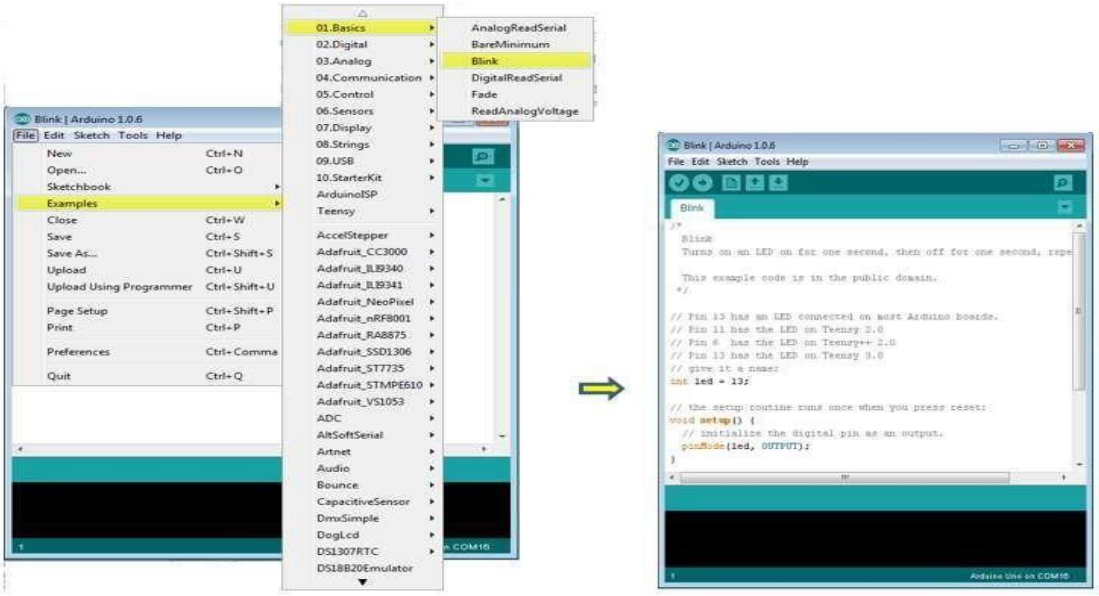


Fig 6.5 OPENING THE PROJECT

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board

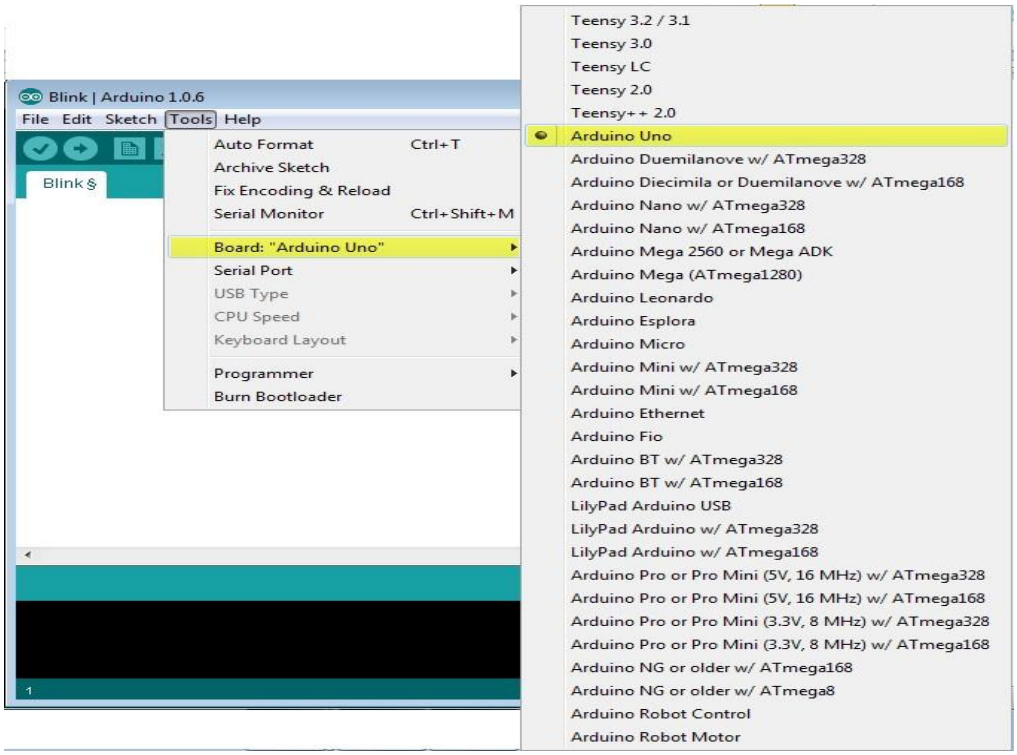


Fig 6.6 SELECTING THE ARDUINO BOARD

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using

Step 7: Select your serial port.

Select the serial device of the Arduino board. Go to Tools ->Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

.

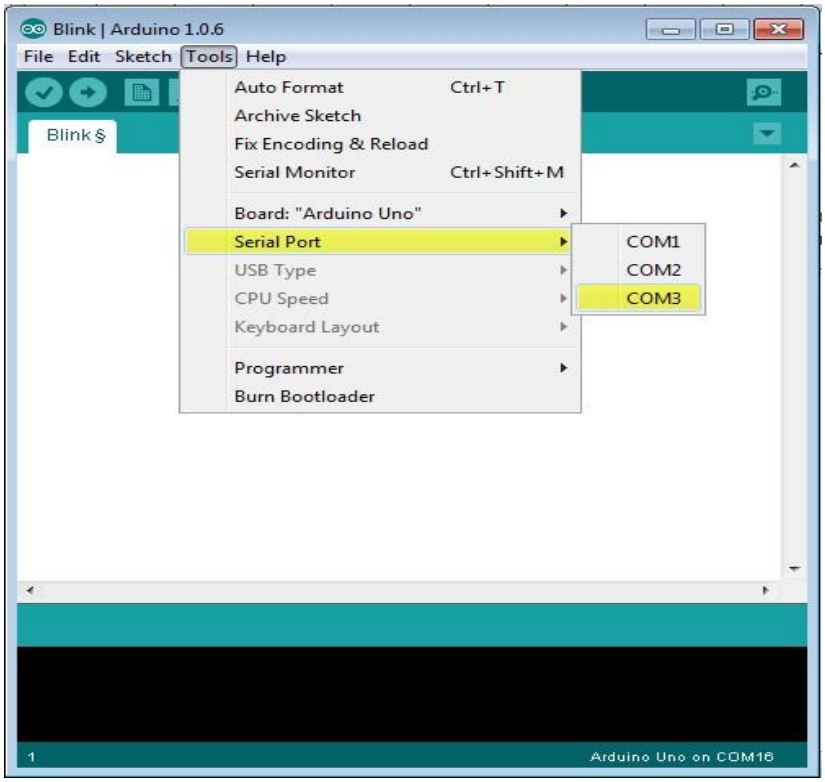


Fig 6.7 SELECTING THE SERIAL PORT

Step 8: Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

Fig 6.8 UPLOADING THE PROGRAM

A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketch.

E- Used to save your sketch.

F- Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

6.5 PROGRAMMING STRUCTURE

In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open- source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch: The first new terminology is the Arduino program called “sketch”. Structure Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions: Setup( ) function

Loop( ) function

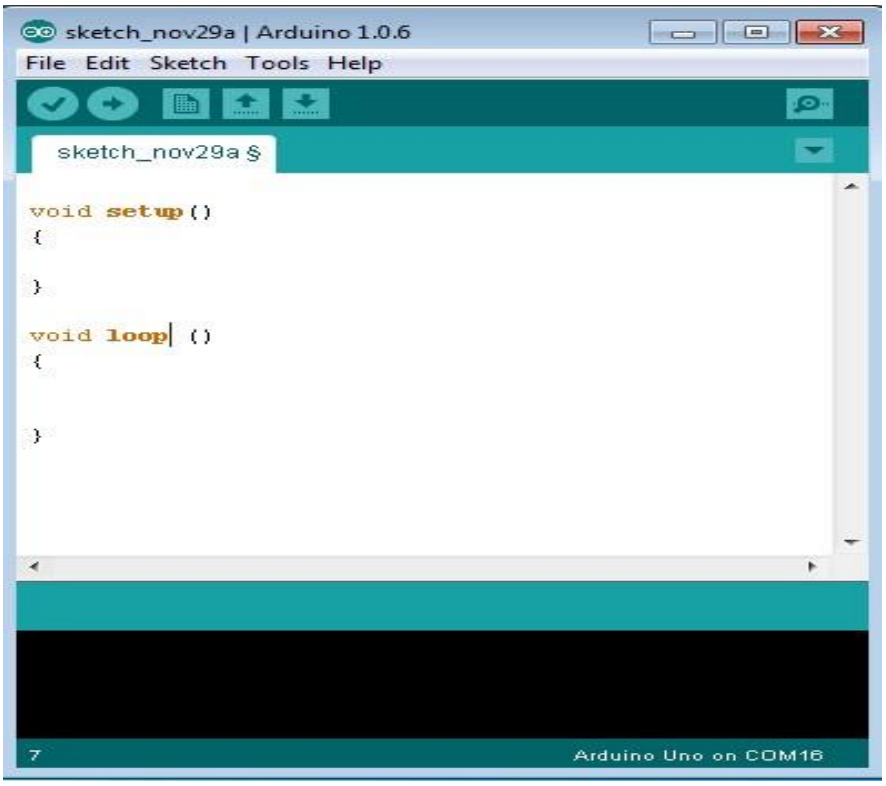


Fig 6.9 ARDUINO PROGRAMMING STRUCTURE

```
Void setup ( )
{
}
```

PURPOSE:

The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

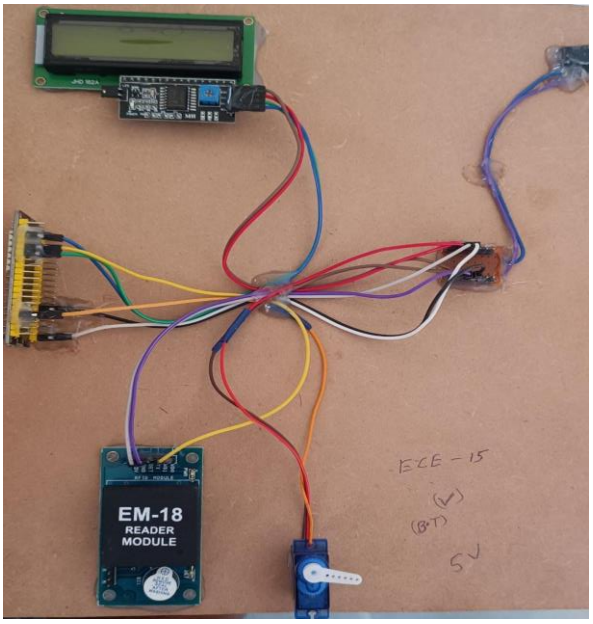
```
INPUT
OUTPUT
RETURN
Void Loop ( )
{
}
```

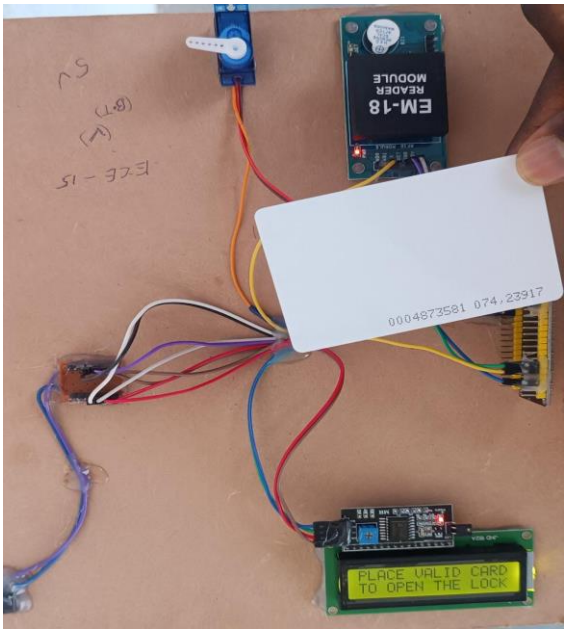
6.6 SOURCE CODE :

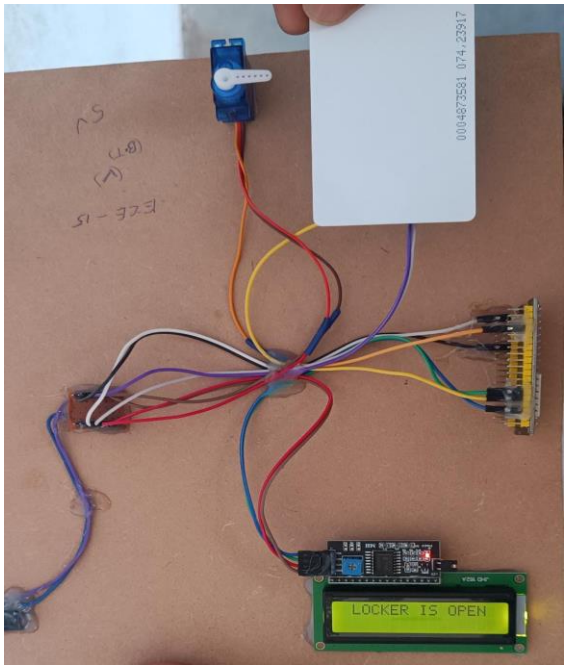
```
#include "BluetoothSerial.h" //Header File for Serial Bluetooth, will be added by default into Arduino
BluetoothSerial ESP_BT; //Object for Bluetooth
String data=" ";
const int ledPin = 16; // 16 corresponds to GPIO16
// setting PWM properties
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;
int dutyCycle;
void setup()
{
 ledcSetup(ledChannel, freq, resolution);
 ledcAttachPin(ledPin, ledChannel);
 Serial.begin(115200);
 ESP_BT.begin("MOTOR SPEED CONTROL"); //Name of your Bluetooth Signal
 Serial.println("Bluetooth Device is Ready to Pair");delay(50);
 Serial.print("Testing DC Motor...");
  }
  void loop()
 {
  if(ESP_BT.available()) //Check if we receive anything from Bluetooth
  {
   data=" ";
   data= ESP_BT.readString(); //Read what we recevive
   dutyCycle=data.toInt();
   Serial.print("Received:"); Serial.println(data);
   if(dutyCycle <= 0)
  {
  dutyCycle=0;
```

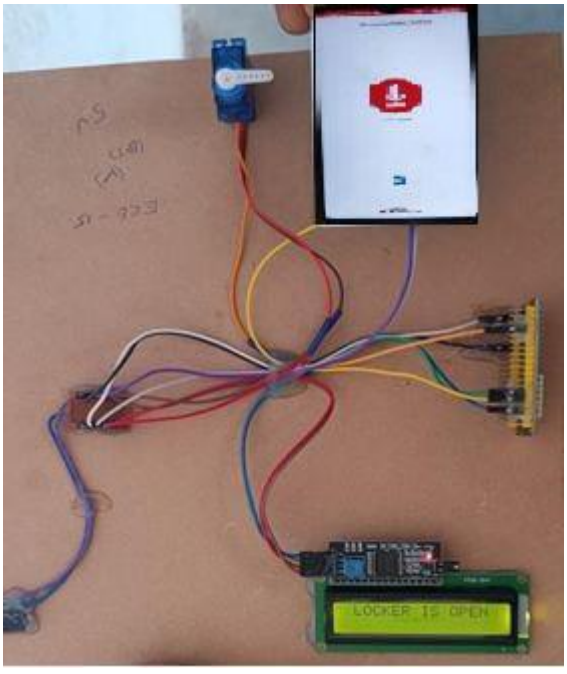
```
 }
 if(dutyCycle >= 255)
{
 dutyCycle=255;
}
}
ledcWrite(ledChannel, dutyCycle);
Serial.println(dutyCycle);
delay(500);
}
```

CHAPTER 7
RESULT AND ANALYSIS REPORT









- Initially 5v of dc supply is given to the esp32 microcontroller then the board gets ON.
- When the rfid tag is brought near to the sensor i.e rfid reader.
- This rfid reader emits radio waves and reads the data coming from the rfid tag and makes sure that whether the data is correct as programmed.
- If yes the servo motor will open, if no the servo motor remains closed and sends an information to the user.
- When we input our voice to the voice to the Arduino voice control it recognises and it sends the information to the microcontroller.
- If the voice command matches then information is sent to servo motor and servo motor will open.

ADVANTAGES

- Two level of authentication.
- Protection levels are more.
- Avoid the unauthorized person to access the locker.
- Speed and sufficient reliable for recognition system.
- Good performance system with complex background.
- Improved security.
- Improved customer experience.
- Cannot be forgotten of lost.
- Reduced operational costs.

APPLICATIONS

- Hotel's Operating Rooms.
- Schools and Universities.
- Product Tracking.
- Locker Systems.

CONCLUSION AND FUTURE SCOPE OF THE PROJECT

CONCLUSION

This project implements a Bank locker security system using Arduino, passive RFID and voice. It is a cheap, low power consumption, compact in size and standalone system. In this project the design and implementation of a prototype of an automated locking system is presented which warrants double layer of security. It ensures the proper user of the vault by securing locker with RFID tags and voice recognition module. It places the conditions of operation which are continuously feeding information to the controller of the proposed system to confirm the rightful access and security of the contents within the vault. The entire system can be easily managed with all the status updates being reeled by the controller to the administrators eliminating the unforced reasons of human errors. The future enhancement to this work could be done by adding some more aspects. Therefore it improved the reliability of bank locker and unauthorized access will be minimized. The enhancement could be further applied to identify the illegal entrance.

FUTURE SCOPE

RFID technology uses radio waves to automatically identify people or objects. After sixty years of development RFID is being used in many fields. , The RFID technology will be a big help to human. . RFID tags will only become cheaper and more powerful with improving technology and design experience .Some standards for RFID system are under development. Also there is improvement in tag life expectancy and durability in past few years. The RFID technology brings new opportunities as well as challenges to the AIDC infrastructure. Although RFID suffers from many limitation but still Demand for RFID systems is increasing day by day. RFID tags can combine with sensors of different kinds. This would allow the tag to report not simply the same information over and over, but identifying information along with current data picked up by sensors. Over times, the proportion of “scan-it-yourself” will increase. RFID technology does not replace barcode.

BIBLOGRAPHY

[1]. Parvathy A, Venkata Rohit Raj, Venumadhav, Manikanta, "RFID Based Exam Hall Maintenance System'', IJCA Special Issue on "Artificial Intelligence Techniques-Novel Approaches& Practical Applications" AIT, 2011.

[2]. Gyanendra K Verma, Pawan Tripathi, "A Digital Security System with Door Lock System Using RFID Technology", International Journal of Computer Applications (IJCA) (0975 – 8887), Volume 5– No.11, August 2010.

[3]. Kumar Chaturvedula .U.P, "RFID Based Embedded System for Vehicle Tracking and Prevention of Road Accidents", International Journal of Engineering Research & Technology (IJERT) , Vol. 1 Issue 6, August – 2012, ISSN: 2278-0181.

[4]. Islam, N.S. Wasi-ur-Rahman, M. "An intelligent SMS-based remote Water Metering System". 12th International Conference on Computers and Information Technology, 2009, 21-23 Dec. 2009, Dhaka, Bangladesh.

[5]. Mohd Helmy Abd Wahab, Siti Zarina Mohd Muji, Fazliza Md. Nazir. "Integrated Billing System through GSM Network". In Proceeding of 3rd International Conference on Robotics, Vision, Information and Signal Processing 2007(ROVISP2007), Penang, 28-30 November 2007.

[6]. Malik Sikandar Hayat Khiyal, Aihab Khan, and Erum Shehzadi. " SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security", Issues in Informing Science and Information Technology. Vol. 9. pp. 887 – 894. 2009.

[7]. Al-Ali, A.R. Rousan, M.A. Mohandes, M. "GSM-Based Wireless Home Appliances Monitoring & Control System", Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications, pp 237-238, 2004.

[8]. Ahmad A.W., Jan N., Iqbal S. and Lee C., "Implementation of ZigBee-GSM based home security monitoring and remote control system", IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 2011, pp. 1-4.