RFID Based Students Attendance and SMS Notification System

SONARGAON UNIVERSITY (SU)

Supervised By

Md. Majharul Islam

Lecturer
Department of EEE
Sonargaon University (SU)

Submitted By

Md. Jamiul Hoque	**ID: EEE 1802014058**
Ujjal Majumder	**ID: EEE 1802014055**
A.K.M Aminul Islam	**ID: EEE 1802014052**
Sagar Mondal	**ID: EEE 1802014066**
Md. Emran Hosson	**ID: EEE 1802014089**
Md. Zahidul Islam	**ID: EEE 1503006010**
Md. Mahafuzur Rahaman	**ID: EEE 1503006240**

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthopath,
Dhaka-1215, Bangladesh.

Date of submission: 05/02/2022

Declaration

It is declared hereby that this project paper or any part of it has not been submitted to anywhere else for the award of any degree.

……	……..
Md. Jamiul Hoque	**Ujjal Majumder**
……	…….
A.K.M Aminul Islam	**Sagar Mondal**
……	…….
Md. Emran Hossen	**Md. Zahidul Islam**

……

Md. Mahafuzur Rahaman

Under Supervision of

Md. Majharul Islam

Lecturer

Department of EEE

Sonargaon University

Certification

This is to certify that this project entitled "**RFID Based Students Attendance and SMS Notification System**" is done by the following students under my direct supervision. This project work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervisor

.....

Md. Majharul Islam

Lecturer

Department of Electrical and Electronic Engineering (EEE)

Sonargaon University (SU)

ACKNOWLEDGEMENT

The report titled as on "**RFID Based Students Attendance and SMS Notification System**" has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast co-operation.

At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable teacher **Md. Majharul Islam**, Lecturer, Department of Electrical & Electronics Engineering, SU–Sonargaon University, Dhaka, for his valuable and patient advice, sympathetic assistance, co-operation, contribution of new idea. Deep theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

Finally, we would like to thanks again to the respected **Vice- Chancellor of SU, Professor Dr. Md. Abul Bashar** also thanks to **Head of Department of SU, Electrical & Electronics Engineering, Professor Dr. M. Bashir Uddin** because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under Bsc in EEE program, and that will be very helpful for our prospective career.

We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report.

Abstract

Main objective of RFID based Attendance System project is to take the attendance of students or employees. Notification sends to parents or authority. To demolish the flaws associated with the manual attendance system.

Radio-frequency identification (RFID) is a technology that uses radio waves to transfer data from an electronic tag, called RFID tag or label, attached to an object, through a reader for the purpose of identifying and tracking the object. RFID chips contain a radio transmitter that emits a coded identification number when queried by a reader device. Some RFID tags can be read from several meters away and beyond the line of sight of the reader. The application of bulk reading enables an almost-parallel reading of tags. This small type is incorporated in consumer products, and even implanted in pets, for identification. in this project we use RFID-522 as rfid card reader, Sim-800L as sms notifier and Arduino Mega as the controller of the project.

TABLE OF CONTENTS

LIST OF THE FIGURE

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 Background

RFID (radio frequency identification) is a new technology that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency (RF) portion of the electromagnetic spectrum to uniquely identify an object, animal, or person. RFID tags are not an "improved bar code" as the proponents of some technologies would like you to believe. An RFID system consists of three components: an antenna and transceiver (often combined into one reader) and a transponder (the tag). The antenna uses radio frequency waves to transmit a signal that activates the transponder. When activated, the tag transmits data back to the antenna. RFID technology differs from bar codes. RFID can read the tag using RF, meaning that the RFID reader can be read from a distance, right through your clothes, wallet or purse. Besides the RFID tag consist of unique ID for each tag. The technology used in RFID has been around since the early 1920s. In our country, this technology is less frequent and the mostly use technology is the biometric systems using finger print. Our government less applies this technology by using RFID as I.C (identification card). In some places of our country, people prefer to use Barcode which is cheaper than RFID. Technology spread very fast and in few years, the possibility that RFID replaces the barcode system will just be a reality.

Nowadays, there are many of universities around our country and each of this university consists of student up to 10 thousand. To handle a large amount of student may be problem specially to get the attendance. Now, process to get attendance in majority universities still used the manual process. The manual process means that when the class (or lecture) starts, lecturer will give a piece of attendance paper and students will check their name and then will sign on it. At the end of class, lecturer will take back the attendance paper and keep it as a record.

Normally, the attendance paper need much time to be signed by all students especially for classes with a lot of student. Students also forget to sign that attendance and they are assumed absent for that class. The problem also will happen when lecturer forget to bring the attendance paper to class. Students need to write their name on a piece of paper and sometimes student

will take this opportunity to cheat during the process of getting the attendance. The suitable solution for this problem is to design a system that will record attendance automatically.
In this project, which is based RFID system is used to record student attendance automatically. This project will use student ID card as RFID tag, a RFID reader and SIM-800L for sent sms to Authority or Parents. This RFID system will be integrated with a microcontroller named Arduino-UNO which is programmed by a software named Arduino-IDE. This method is more effective to prevent problems in the process of getting the attendance manually

.

1.2 Objective

- The main objective of this project is to take attendance smartly.
- To utilize time.
- Notification send to parents or authority

1.3 Methodology

- Creating an idea for design and construction of RFID based attendance system. And designing a block diagram & circuit diagram to know which components need to construct it.
- Collecting the all components and programming for the microcontroller to controll the system.
- Setting all components in a circuit board & soldering. Then assembling the all block in a board and finally run the system & checking.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

The term Radio Frequency Identification (RFID) technology has come a long way both in terms of the multitude of applications and their potential uses. Despite the involvement of RFID in a variety of areas such as logistics, inventory control, aviation security, road, and rail rolling, the potential implementation and use of the technology in libraries is still in its budding stage, particularly in the case of countries with developing economies like India.

As inferred from the fifth law of library science that 'library is a growing organism', librarians are always found interested in adopting latest technologies to provide better and efficient services to the patrons. Indeed, RFID technology has redefined various library related services and made every patron's job easier and efficient, from the patrons to the library professionals, as compared to conventional technologies like barcode, but in a developing country like India, the high cost of implementation and maintenance is still the major barrier in the proliferation of such auspicious technology.

After the perusal of literature based on RFID, it is found that most of the work was carried out on different aspects of RFID technology and none of them systematically covered them keeping focus the availability of technology in libraries.

The aim of the paper is to present the descriptive literature reviews (the abstract section of the source papers) and integrative literature reviews (reference the conclusion) on different facets of RFID technology with emphasis on select libraries of India as revealed in the literature available online.

2.2 Literature Review

The RFID thought isn't new yet rather has been around for an impressive time span; really, it was first observed in WWII by the Air-Force to recognize companion flying planes to that of rival plane using radars (table 1 underneath shows brief of the recorded setting of RFID development). Starting now and into the foreseeable future, this creation has been used for various corner applications, for instance retail industry, monitoring merchandise against robbery structures, equip following in air terminals, electronic tolls, and numerous others. This was the beginning time of RFID innovation time.

History of RFID Systems

1930 to 1940 IFF framework was made by an exploration research center.

1940 to 1950 It was then used to distinguish companion or foe in WWII

1950 to 1960 Present day air traffic additionally utilized the distinguish companion or foe ideas.

Bit by bit RFID ideas were connected to military divisions. It was utilized in research labs.

1960 to 1970 SENSOR frameworks that utilized RFID innovation started in this timeframe.

1970 to 1980 Uninvolved labels were presented for the first run through in this period. RFID was

currently utilized in zoos and national parks to monitor creatures 1980 to 1990

RFID frameworks were currently made in an expansive scale. Programmed

installment at toll doors application started in this period.

1990 to 2000 Between operable hardware with RFID innovation was created.

2003 EPC worldwide was presently MIT's auto-ID focus.

2005 EPC was propelled by Walmart.

2.3 Conclusion

The article presents the reviews mainly on the different theoretical and practical aspects of the implementation and use of RFID technology in libraries, including the benefits, pros, cons, challenges, and problems related to RFID application. In conclusion, a common feature that reappears is the improvement in the library service after its implementation vis a vis shortage of funds for its implementation. The future of the library can be improved many times if generous funding is done to perpetuate the technology in all the potentially active libraries of wikipedia. There will be no exaggeration in saying that RFID is right technology investments for all stakeholders, for long-term security of the library.

The reviews of this paper will not only guide the librarians in effective implementation of RFID technology in their libraries, but also abreast the knowledge of RFID technology and its advantages. The major limitation of the study related to many libraries in asia is using RFID technology, but not reported in the form of an article. Although, there exist many related studies that are covered in foreign studies, but these are excluded due to Geographical limitation.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Block Diagram

Here is the block diagram of the RFID based attendance system with all the essential components.

The Block Diagram of the RFID based attendance system:

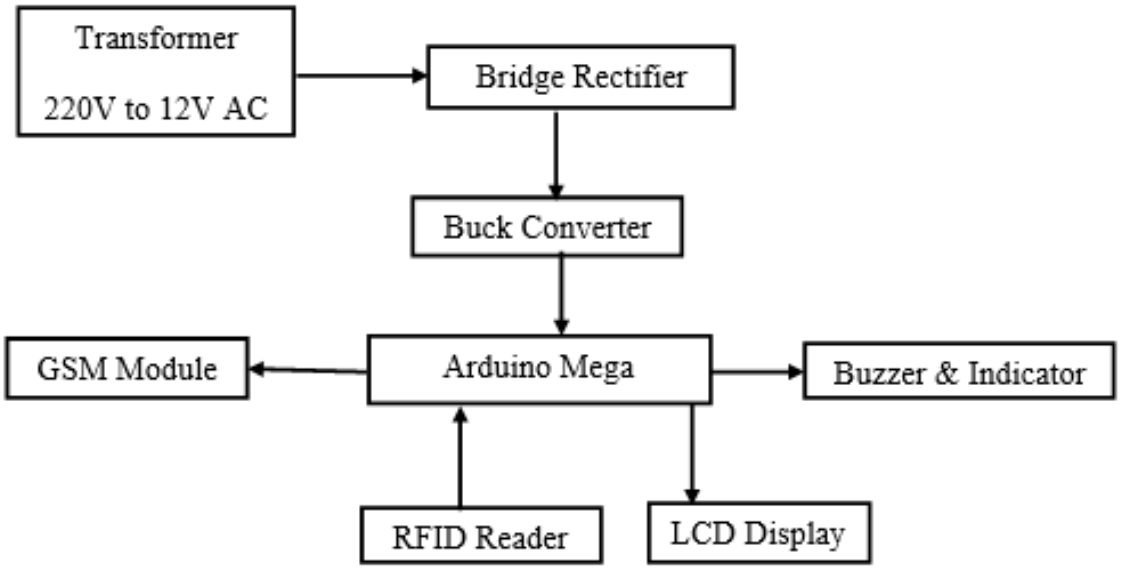


Fig 3.1: Block Diagram of RFID based attendance system.

3.2 Working Principle

This attendance system is implement for reduce wasting of time and inform parents or authority so that they can know the student or employee is attendant or not. In this system, the microcontroller which is very essential component and it always keep sensing the RFID reader whenever it available. When any ID card is comes across the card reader the microcontroller read and analyze the data, if the card is valid microcontroller send the card and related information such as name and id number to sim module to send sms. Microcontroller also send those information to LCD and a dedicated green Led will glow up. if the card information is not match it send error message to LCD and a red led will blink. all valid information also save to computer via usb cable.

3.3 Circuit Diagram

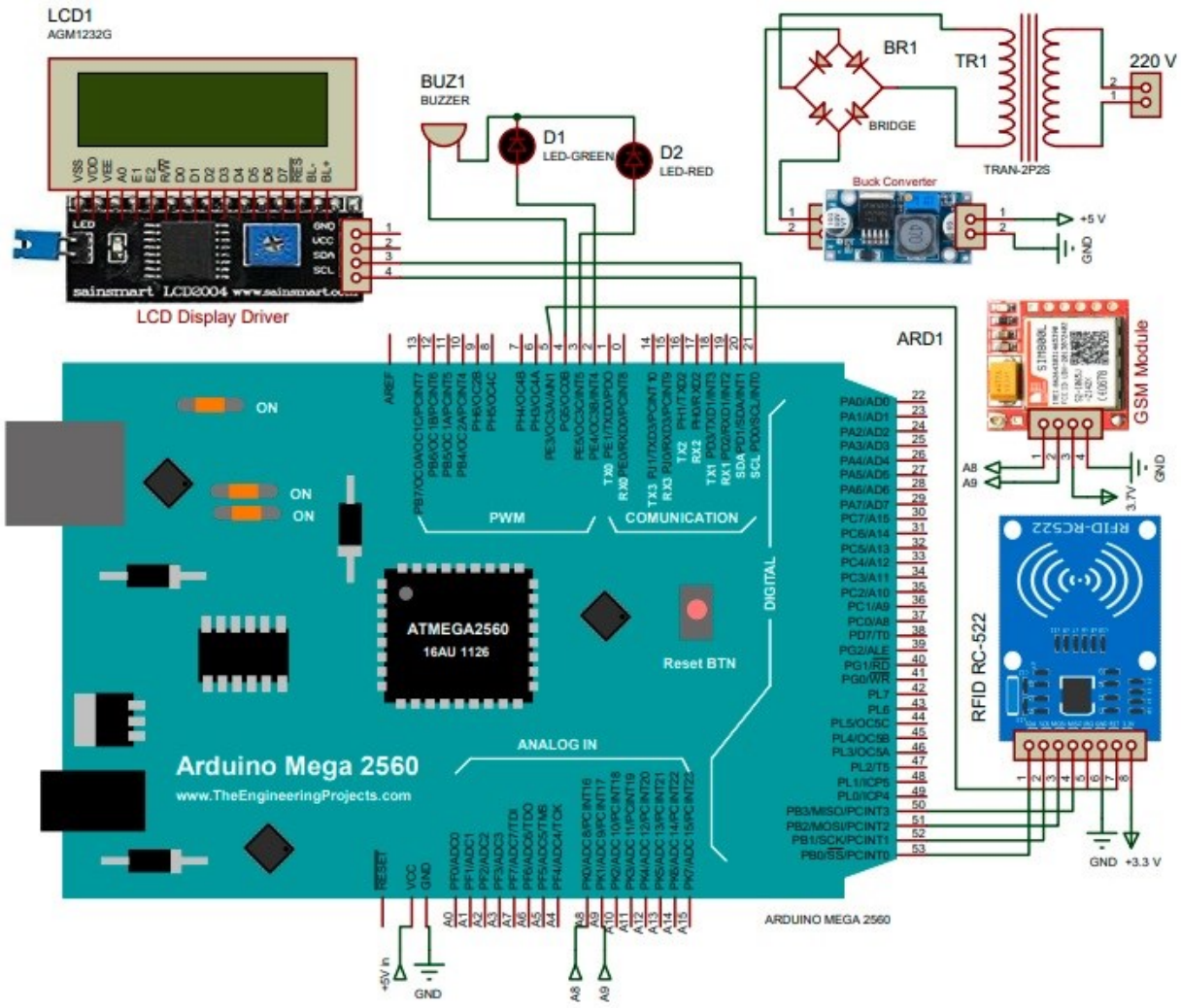


Fig 3.2: Circuit Diagram of RFID card based attendance System.

3.4 Advantage

RFID attendance system is ecumenically benefited.

- This system could be used in that places where has large amount of population.
- This system could be used in industries for counting worker.
- This system could be used in educational institutes and hospitals.
- This system is more compact and reliable as compared to the traditional attendance system.
- This system is less costly as compared to the other attendance systems.
- This System can observe by remote place using GSM.

3.5 Application

- Industry.
- Educational Institute.
- Hospital.
- Restaurant.

3.6 Project Picture

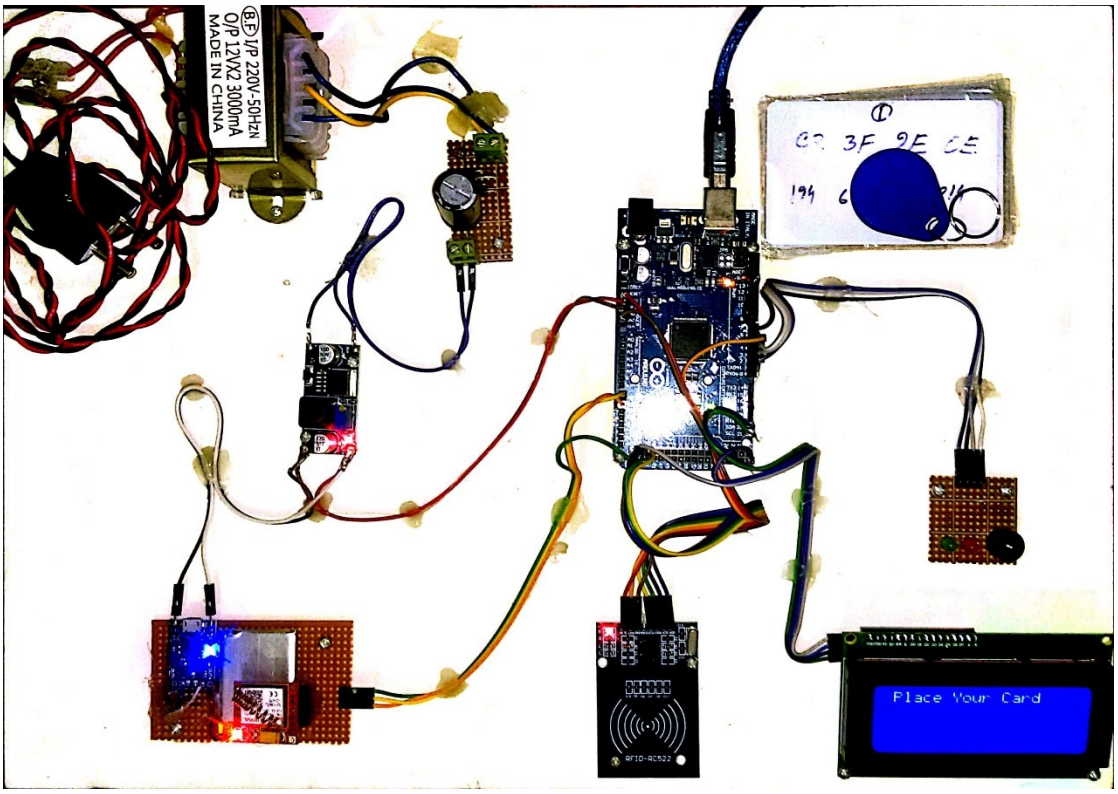


Fig 3.3: Project Picture

CHAPTER 4

HARDWARE ANALYSIS

4.1 Hardware Required

To fulfill our project we use

- Transformer
- Bridge Rectifier
- Buck Converter
- Arduino Mega
- RFID Card Reader
- LED & Buzzer
- LCD Display
- GSM Module
- Camera

4.2 Processing System

This part is Brain of this project. The Arduino Mega is a microcontroller board based on the ATmega2560. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for any types of project to use as well as professionals and Students. Arduino MEGA has great advantage that it has built-in chip which work as a USB-to-serial converter. In this project we use Arduino-Mega which is consist of ATmega2560 Microcontroller with 16MHz crystal. It works upon the data signal which gets from IR obstacle and temperature sensor. The CPU decides what to do at which condition. In this project we use relay module for operate door solenoid.

Features of the Arduino MEGA:

- Microcontroller: ATmega2560
- Operating Voltage: 5V
- Input Voltage (limits): 7-12V
- Digital I/O Pins: 54 (of which 14 provide PWM output)
- Analog Input Pins: 16
- Flash Memory: 256 KB
- SRAM: 4 KB (ATmega2560)
- EEPROM: 4 KB (ATmega2560)

- Clock Speed: 16 MHz

4.2.1 Arduino IDE Platform

Arduino is an open source electronics prototyping platform based on flexible hardware and software. The Arduino is a simple yet sophisticated device which is based on Atmel's ATmega microcontrollers. The Arduino software is supported by Windows and Linux operating systems despite the fact that most microcontrollers are limited to Windows operating system. The software language is based on AVR C programming language and can be expanded through C++ libraries. There are various types of Arduino micro controller board available in the market including the Arduino kits and Arduino shields.

4.2.2 Arduino Mega Board

Arduino Mega is one of the microcontroller boards manufactured by the Arduino and it is a microcontroller board based on Atmel's ATmega2560 microcontroller. Mega board is the latest in a series of USB (Universal Serial Bus) Arduino boards which is the reference model for the Arduino platform. The Arduino Mega board has a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, a reset button, 16 analog inputs and 54 digital input/output pins (of which 15 can be used as PWM outputs). It uses the Atmega16U2 programmed as a USB – to-serial converter instead of FTDI USB – to–serial driverchip which was used in all the pre-ceding boards. The board has 256 KB flash memory of which 0.5 KB is used by boot-loader, 8 KB of SRAM, 4 KB of EEPROM and 16 MHz clock speed. Reprinted from the Arduino Board Mega Figure 1 shows the Arduino-Mega Board manufactured by the Arduino in Italy. It can be powered via a USB connection or with an external power supply. As can be seen in figure 1, pins A0 to A5 are the analog input pins, pins 0 to 13 are 14 digital input/outputpins and the pins with a "~" sign can be used asdigital pins PWM o can be used as input or output pins by selecting the mode by using the function Pin-Mode() and then using the function digitalRead() or digitalWrite() according to the ne-cessity. Pins 0(RX) and 1(TX) are used for serial communication while pins 10(SS), 11(MOSI), 12(MISO) and 13(SCK) are used for SPI(Serial Peripheral Interface) communication. In addition to pin 0 and 1, a SoftwareSerial library allows serial communication on any of the Mega's digital pin.

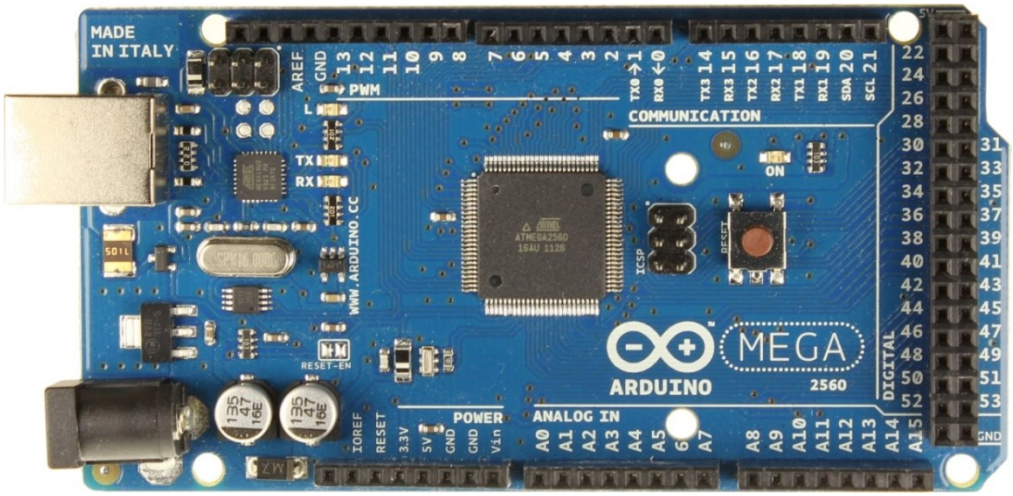


Fig 4.1: Arduino Mega

4.2.3 ATmega 2560 Microcontroller

The microcontroller is a low-power CMOS (Complementary Metal Oxide Semiconductor) 8-bit microcontroller based on the AVR enhanced RISC (Reduced Instruction Set Computer) architecture. The powerful execution of instructions in a single clock cycle leads to the achievement of 1 MIPS per MHz throughputs allowing the designer to optimize power consumption versus processing speed.

Fig 4.2: ATmega2560 Microcontroller

Reprinted from Datasheet of ATmega2560 the internal architecture of the microcontroller is shown in Fig 2.3.3 (02). The central processing unit (CPU) is the brain of the microcontroller which controls the execution of the program. The MCU (Microcontroller unit) consists of 4K/8K bytes of in-system programmable flash with read-while-write capabilities, 256/412/1K

bytes EEPROM along with the 512/1K/2K bytes of SRAM. Along with this, the MCU consists of many other features

- 54 general purpose I/O lines
- flexible timer/counters with compare modes, internal and external interrupts and a serial programmable USART
- A byte-oriented 2-wire serial interface, an SPI serial port,a 15-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable watch-dog timer with an internal oscillator and 5 software-selectable power saving modes.

The five, software selectable, power saving modes are idle mode, Power-down mode, Power-save mode, ADC Noise Reduction mode and theStandby mode. As mentioned in section 2.1.2, the CPU is the brain of the microcontroller which controls the execution of the program. Therefore the CPU is able to access the memories, perform calculations, control peripherals and handle interrupts. The AVR uses the Harvard architecture with separate memories and buses for program and data to maximize the performance as well as the parallelism. The principle of execution of instructions in the program memory is the single-level pipelining. The concept of pre-fetching the next instruction while executing one instruction enables the instructions to be executed in every clock cycle and the program memory is in the System Reprogrammable Flash memory.

Reprinted from Datasheet of ATmega2560, The block diagram of AVR CPU Core architecture is shown in figure 3. The fast-access Register File contains 32 x 8 bit general-purpose working registers with a single cycle access time which results in a single-cycle ALU operation. The arithmetic and logical operations between the registers or between the constant and a register are supported by the ALU. The status register is updated to reflect information about the result of the operation after an arithmetic operation. The boot program section and the application program section are the two main sections of the program flash memory. Stack stores the return address of the program counter during the interrupts and subroutine calls which is allocated in the general data SRAM. The size of the stack is limited by the total size and usage of the SRAM. The data SRAM is accessible through five different addressing modes supported in the AVR architecture while the stack pointer is read/write accessible in the I/O space. The memory spaces in the AVR architecture are all linear and regular memory maps

4.3. LCD

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 20x4 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs.

The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 20 characters per line and there are 4 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction

given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

Fig 4.3: LCD Display

Features of LCD Display

5 x 8 dots with cursor

Built-in controller (KS 0066 or Equivalent) + 5V power supply (Also available for + 3V) 1/16 duty cycle

B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED) N.V. optional for + 3V power supply.

4.4 GSM

A number of signal detecting devices are available in the market. These devices include:

- Ethernet Module
- Wifi Module
- Bluetooth Module
- SIM / GSM Module

All of the devices have their own advantages and disadvantages but we will be targeting that particular device which best suits our requirement. There are certain features that should be considered when choosing the GSM module for use. The features are

- Long Distance Coverage
- Function description
  - Supply voltage: 3.5V - 4.2V
  - Power consumption:
    sleep mode < 2.0mA
    idle mode < 7.0mA
  - GSM transmission (avg): 350 mA
  - GSM transmission (peek): 2000mA
  - Module size: 25 x 23cm
  - SIM card socket: microSIM
  - Antenna connector: IPX
  - Status signaling: LED
  - Working temperature range: -40 do + 85 ° C
- Has high-performance
- Low Cost

This GSM module can easily achieve data. Its operating frequency is among the 900/1800/1900 MHz frequency band. In SIM800L signal transmit time of different devices stands at a 0.5 seconds interval so that the work load of SIM chip can be reduced substantially and more sleeping time can be saved for GSM module. This module is set with serial interface, which is easy to use and simplifies the overall design.

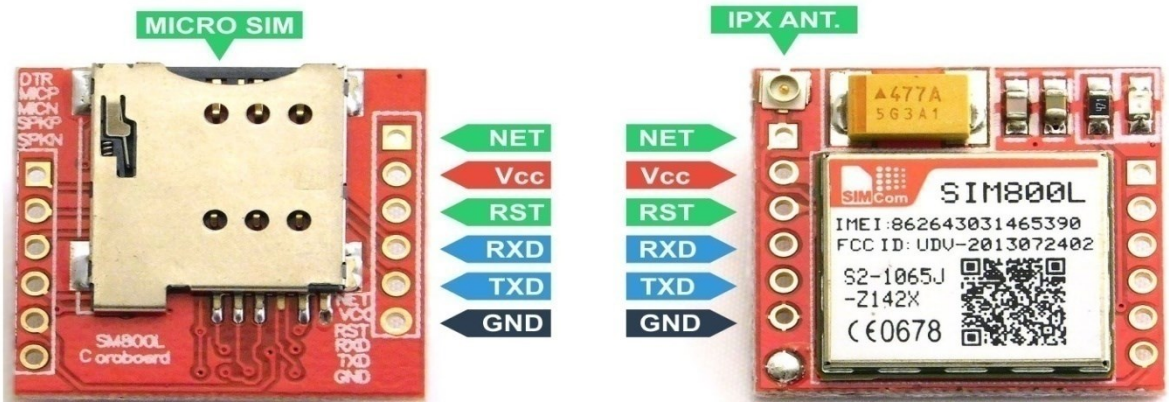


Figure 4.4: SIM800L GSM/GPRS

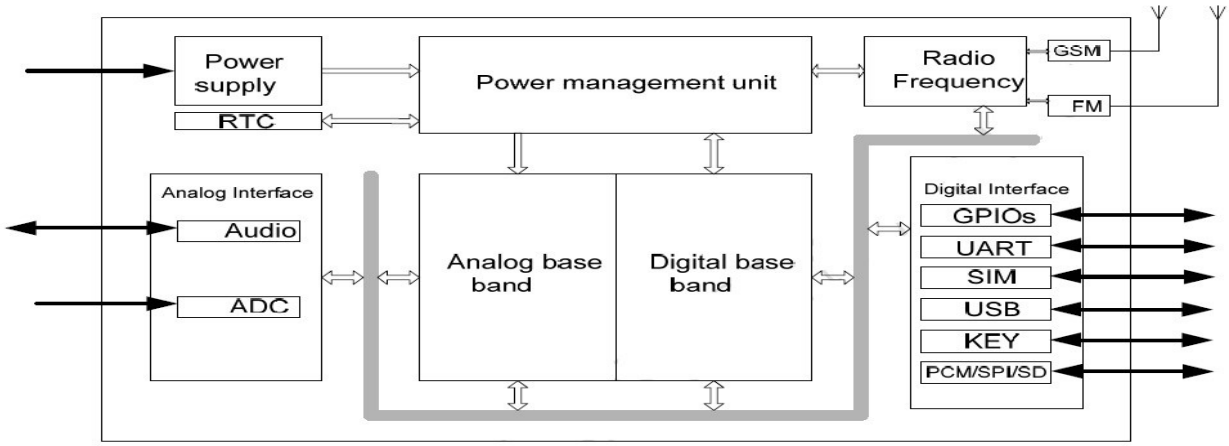


Fig 4.5**:** Schematic Diagram of SIM800L GSM Module

GSM (Global System for Mobile communication) is a digital mobile telephony system that is widely used in Europe and other parts of the world. GSM uses a variation of time division multiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band.

GSM/GPRS module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate

communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1. Receive, send or delete SMS messages in a SIM.
2. Read, add, search phonebook entries of the SIM.
3. Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.

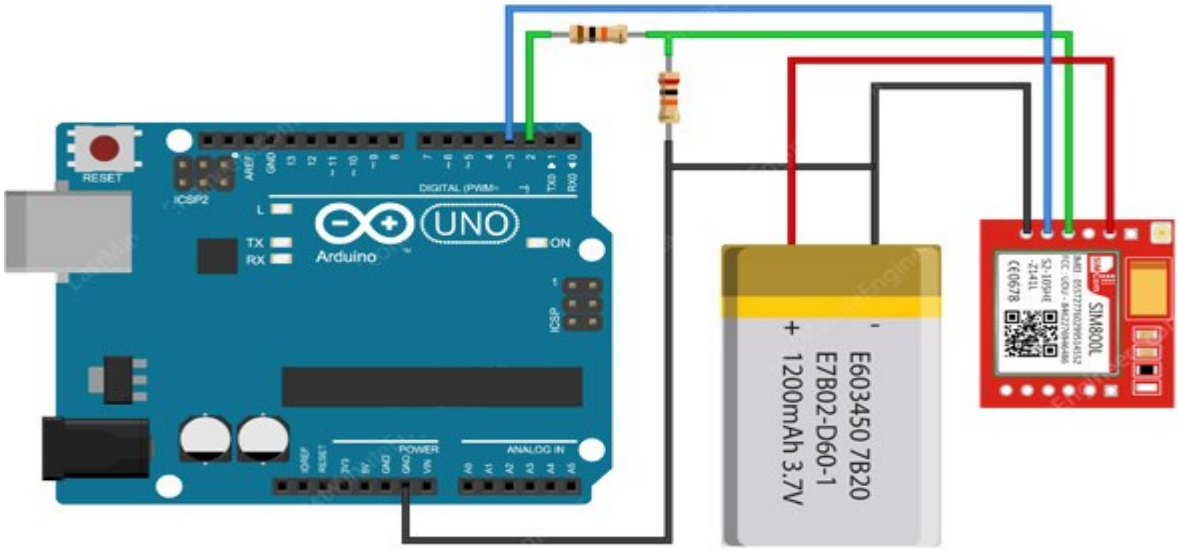


Figure 4.6: Circuit Diagram for Arduino and GSM module Connection.

4.5 RFID Module

In this RFID based attendance system, the RFID reader is a device which is used in collecting the information or data from the RFID card or RFID tag. It is used for individual object and transfers the data from the RFID tag to RFID card reader through radio waves. The RFID reader does not scan all the RFID tag; it's only scanned the tags which is 3 to 300 feet. The RFID technology allows several items to be scanned quickly and fast identification of the specific objects.

RFID or Radio Frequency Identification system consists of two main components, a transponder/tag attached to an object to be identified, and a Transceiver also known as interrogator/Reader.

A Reader consists of a Radio Frequency module and an antenna which generates high frequency electromagnetic field. On the other hand, the tag is usually a passive device, meaning it doesn't contain a battery. Instead it contains a microchip that stores and processes information, and an antenna to receive and transmit a signal.

To read the information encoded on a tag, it is placed in close proximity to the Reader (does not need to be within direct line-of-sight of the reader). A Reader generates an electromagnetic field which causes electrons to move through the tag's antenna and subsequently power the chip.

The powered chip inside the tag then responds by sending its stored information back to the reader in the form of another radio signal. This is called backscatter. The backscatter, or change in the electromagnetic/RF wave, is detected and interpreted by the reader which then sends the data out to a computer or microcontroller.

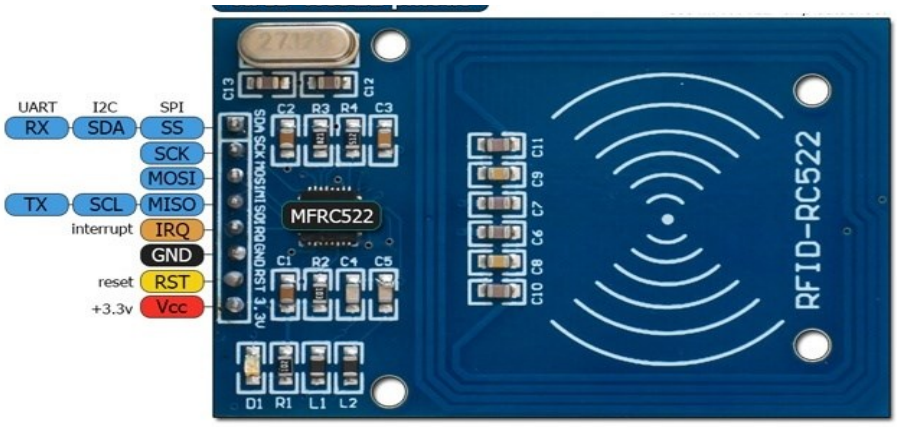


Fig 4.7: RFID Module

The RC522 RFID module based on MFRC522 IC from NXP is one of the most inexpensive RFID options that you can get online for less than four dollars. It usually comes with a RFID card tag and key fob tag having 1KB memory. And best of all, it can write a tag, so you can store your some sort of secret message in it.

The RC522 RFID Reader module is designed to create a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags (ISO 14443A standard tags). The reader can

communicate with a microcontroller over a 4-pin Serial Peripheral Interface (**SPI**) with a maximum data rate of **10Mbps**. It also supports communication over I2C and UART protocols.

Specifications

Frequency Range: 13.56 MHz ISM Band

Host Interface: SPI / I2C / UART

Operating Supply Voltage: 2.5 V to 3.3 V

Max. Operating Current 13-26mA

Min. Current(Power down): 10µA

Logic Inputs: 5V Tolerant

Read Range: 5 cm

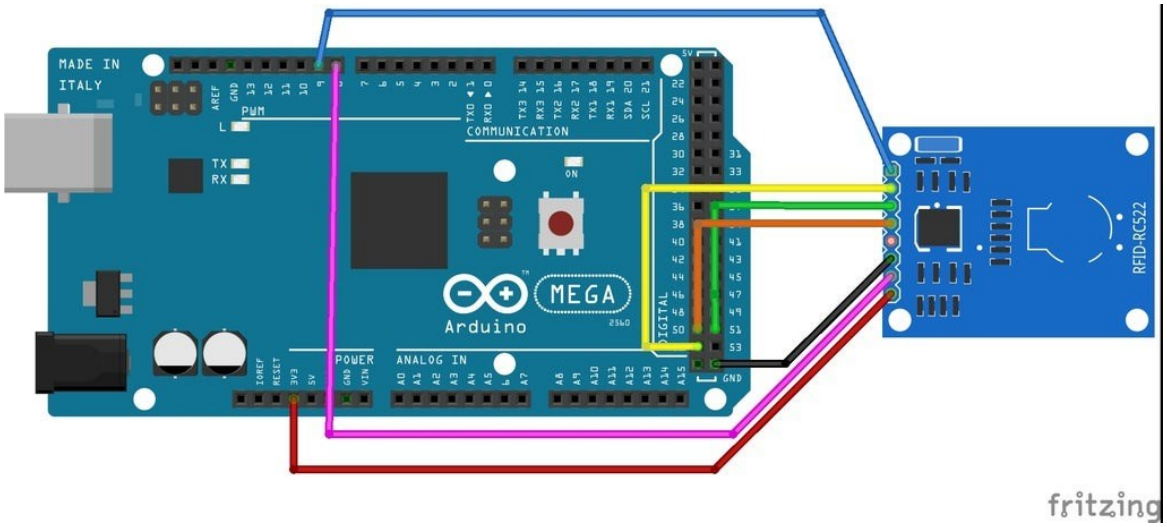


Fig 4.8: Circuit Diagram for Arduino and RFID module Connection.

4.6 Buzzer

A buzzer is an electrical device that makes a buzzing noise and is used for signaling. A buzzer has a piezo disc and an oscillator inside. When the buzzer is powered, the oscillator generates a frequency around 2-4 kHz and the piezo element vibrates accordingly to produce the sound. Assoon as the gas leak is detected, the buzzer automatically rings to alert the users about the gas leak.

4.7 Transformer

A transformer is a passive electrical device that transfers electrical energy between two or more circuits. A varying current in one coil of the transformer produces a varying magnetic flux, which, in turn, induces a varying electromotive force across a second coil wound around the same core. Electrical energy can be transferred between the two coils, without a metallic connection between the two circuits. Faraday's law of induction discovered in 1831 described the induced voltage effect in any coil due to changing magnetic flux encircled by the coil.

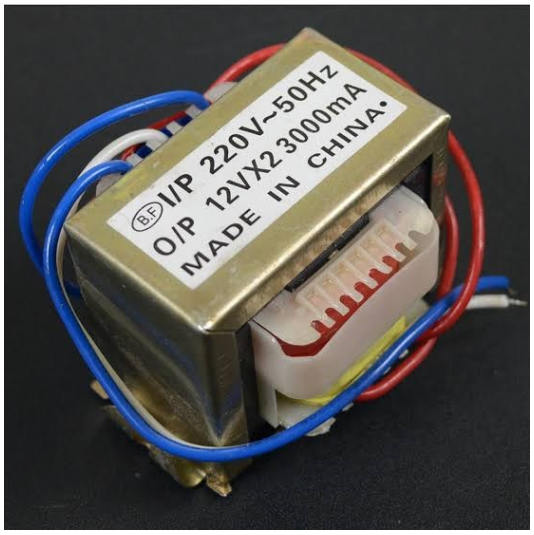


Fig 4.9: 220v to 12v step down transformer.

Transformers are used for increasing or decreasing the alternating voltages in electric power applications, and for coupling the stages of signal processing circuits.

4.8 Diode

A diode is a specialized electronic component with two electrodes called the anode and the cathode. Most diodes are made with semiconductor materials such as silicon, germanium, or selenium. Some diodes are comprised of metal electrodes in a chamber evacuated or filled with a pure elemental gas at low pressure. Here we used converted AC into DC using a bridge-wave rectifier that consists of four diodes

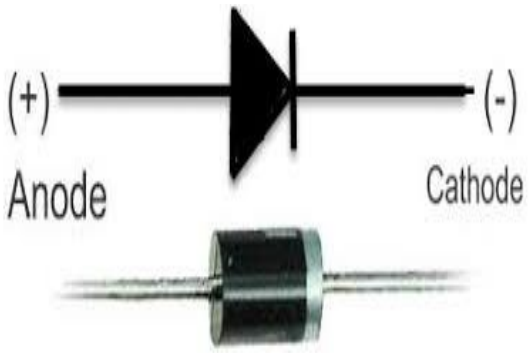


Fig 4.10: Diode

4.9 Full wave bridge rectifier

A Full wave bridge rectifier is a circuit arrangement which makes use of both half cycles of input alternating current (AC) and converts them to direct current (DC). ... This arrangement is known as a Bridge Rectifier. It uses the entire AC wave (Both positive and negative sections). Each diode uses 0.7v when conducting and there are always two diodes conducting.

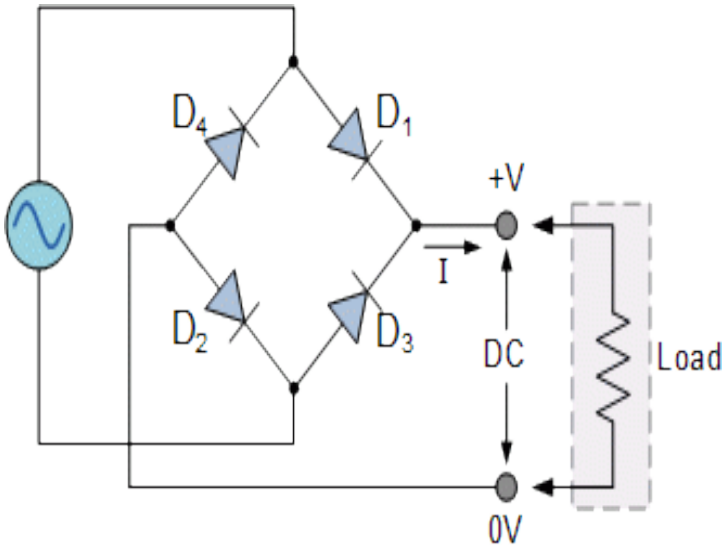


Fig 4.11: Full wave bridge rectifier circuit diagram

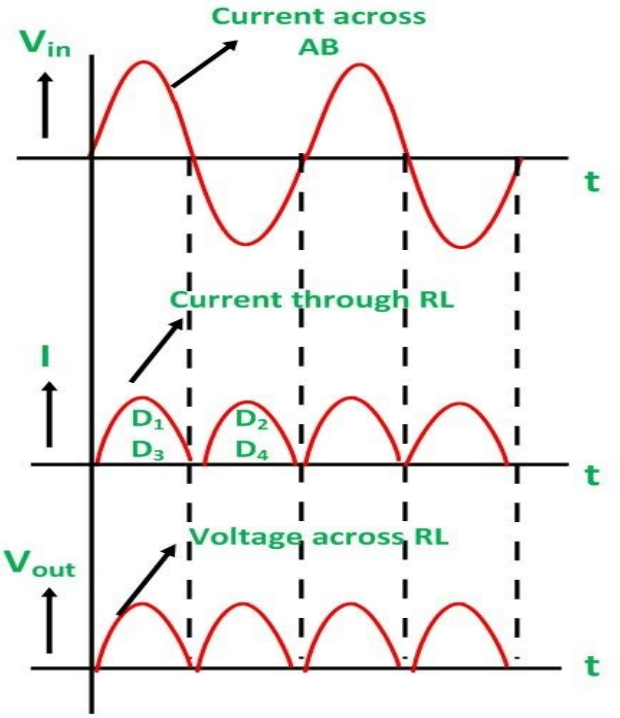


Fig 4.12: Full wave bridge rectifier wave from

4.10 Buck Converter Module

A buck converter (step-down buck converter) is a DC-to-DC power converter, which steps down voltage from its input to its output. The basic operation of the buck converter has the current in an inductor controlled by two switches. In the idealised converter, all the components are considered to be perfect. Specifically, the switch and the diode have zero voltage drop when on and zero current flow when off, and the inductor has zero series resistance. Further, it is assumed that the input and output voltages do not change over the course of a cycle

Fig 4.13: DC-DC Buck Converter

4.11 Indicator (LED)

A light emitting diode (LED) is a semiconductor light source that as shown in Figure 3.5 act as an indicator. The red LED indicates door lock while the green LED indicates door unlock.

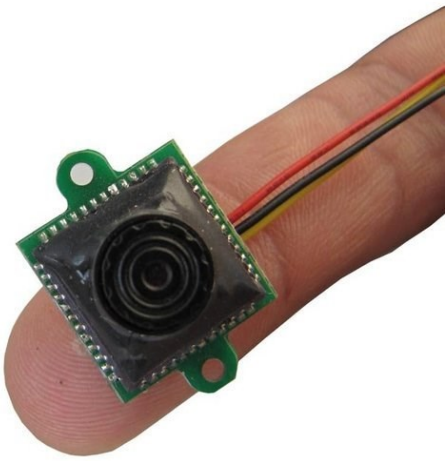
Fig 4.14: LED

4.12 Jumper Wire

Fig 4.15: Jumper Wire

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a bread board or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

4.13 Camera



Hidden cameras can be built into commonly used objects such as television sets, smoke detectors, alarm clocks, motion detectors, ink pen caps, plants, and mobile phones. Hidden cameras may be used for household surveillance devices and may also be used commercially or industrially as spying.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 Arduino Software (IDE):

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

5.2 Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for

searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor. Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

The available tools are:


Verify :Checks your code for errors compiling it.


Upload :Compiles your code and uploads it to the configured board. See uploading below for details. If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"


New :Creates a new sketch.


Open :Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content. Due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the **File | Sketchbook** menu instead.


Save :Saves your sketch.


Serial Monitor :Opens the serial monitor.

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

5.2.1 File

- New :Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open :Allows to load a sketch file browsing through the computer drives and folders.

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

5.2.2 Edit

- Undo/Redo :Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut :Removes the selected text from the editor and places it into the clipboard.
- Copy :Duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum :Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- Copy as HTML :Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste :Puts the contents of the clipboard at the cursor position, in the editor.
- Select All :Selects and highlights the whole content of the editor.
- Comment/Uncomment :Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent :Adds or subtracts a space at the beginning of each

selected line, moving the text one space on the right or eliminating a space at the beginning.

- Find :Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next :Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous :Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

5.2.3 Sketch

- Verify/Compile :Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload :Compiles and loads the binary file onto the configured board through the configured Port.
- Upload using the programmer :This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.
- Export Compiled Binary :Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder :Opens the current sketch folder.
- Include Library :Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File :Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

5.2.4 Tools

- AutoFormat :This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are

indented more.

- ArchiveSketch :Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- Fix Encoding & Reload: Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor :Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board :Select the board that you're using. See below for descriptions of the various boards.
- Port :This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer :For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- BurnBootloader :The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the **Boards** menu before burning the bootloader on the target board. This command also set the right fuses.

5.2.5 Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection.

FindinReference :This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

5.2.6 Sketchbook

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

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino**.**

5.2.7 Tabs, Multiple Files, and Compilation

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

5.2.8 Uploading

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1** (for a Duemilanove or earlier USB board), or

/dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably **COM1** or **COM2** (for a serial board) or **COM4**, **COM5**, **COM7**, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be **/dev/ttyACMx** , **/dev/ttyUSBx** or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

5.2.9 Libraries

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

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library. To write your own library, see this tutorial.

5.2.10 Third-Party Hardware

Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

5.2.11 Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to **Serial.begin** in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor.

5.2.12 Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

5.2.13 Language Support

Since version 1.0.1 , the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.) If you would like to change the language manually, start the Arduino Software (IDE) and open the **Preferences** window. Next to the **Editor Language** there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting **System Default** from the **Editor Language** drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

Fig 5.1: Languages supported by the software.

5.2.14 Boards

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

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

- ArduinoYùn

An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino/GenuinoUno

  An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

- ArduinoDiecimilaor

  Duemilanovew/ATmega168 AnATmega168

  running at 16 MHz with auto-reset.

- Arduino Nanow/ATmega328

  An ATmega328 running at 16 MHz with auto-reset. Has eight analog inputs.

- Arduino/GenuinoMega2560

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Mega

  An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino MegaADK

  An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Leonardo

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino/Genuino Micro

  An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.

- Arduino Esplora

  An ATmega32u4 running at 16 MHz with auto-reset.

- Arduino Miniw/ATmega328

  An ATmega328 running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Ethernet

  Equivalent to Arduino UNO with an Ethernet shield: An ATmega328 running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino Fio

  An ATmega328 running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328, 6 Analog In, 14 Digital I/O and 6 PWM.

5.3 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017.Feature led product releases are typically biannual, while maintenance based service packs are released as required.

CHAPTER 6

DISCUSSION & CONCLUSION

6.1: Discussion

The project has met all the objectives as listed previously. The objective as defined earlier was to reduce time waste, take attendance smartly, Notification send to parents or authority, Store attendance data to server. Some excellent features are also provided in the project that makes the project unique.

6.2 Limitation

- This project is not capable to store data
- It needs computer for add or remove Parents number.

6.3 Future Scope

- Server PC can be added, so that data can store on it.
- Memory can use to store student data.
- IOT can be implement to check status over internet

6.4 Conclusion

For developing countries like Bangladesh, people working in industries and garment factories are not enough educated and conscious about their attendance. Also they have no direct communication with the Admin or top management authorities of the industry. As attendance is vital thing for the development of the industry and so understand the thoughts or reason of being late is also a very important thing. So an attendance management system providing this privilege is crying need for now-a–days. Our attendance system with RFID card reader provides the accurate attendance information of the workers and an interface to communicate with the workers. As all data is uploaded in server, internet connection is a not must during attendance taking but server pc must be open. Our automated attendance management system is user friendly, easy to use and provides a better security and privacy than manual attendance system.

Reference

[1] C. Turcu, **"Current Trends and Challenges in RFID"**, *I-Tech Education and Publishing KG*, ISBN 978-953-307-356-9, pp. 87-93, Rijeka, Croatia, 2011.

2] H. Lehpamer, **"Automatic Identification Systems"**, Chapter in the book titled **"RFID Design Principles"**, *Artech House Microwave Library*, ISBN 13: 978-1-59693-194-7, INC. 685 Canton Street, Norwood, USA, 2008.

[3] S. Ahuja and P. Potti, **"An Introduction to RFID Technology"**, *Journal of Technology Management & Innovation*, DOI: 10.4236/ cn. 2010.23026, Vol.2, Issue 1, pp. 183 - 186, Florida, United States of America, Aug 2010.

[4] T. T. Tanpure, H. S. Sonawane, Ch. R. Sonawane, P. V. Ovhal, and V, B. Maral, **"Online Student Monitoring System Using Passive RFID"**, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 1, Issue 2, pp 409-416, April 2013.

[5] A. S.Tiwar, P.T.Nasre, S.C.Sevalkar, M.B.Giradkar, N. R. Patel, and A. R. M. Khan, **"Review of Student Attendance System Using RFID"**, *Proceedings of Annual International Conference IRAJ*, ISBN: 978-93-82702-54-2, Pune India Jan 2014.

[6] R. Patel, N. Patel and M. Gajjar, **"Online Students' Attendance Monitoring System in Classroom Using Radio Frequency Identification Technology: A Proposed System Framework"**, *International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, vol. 2, Issue 2, Gujarat, India, Feb 2012.

[7] S. Nainan, R. Parekh and T. Shah, **"RFID Technology Based Attendance Management System"**, *International Journal of Computer Science Issues*, ISSN 1694-0814, vol. 10, no. 1, Issue 1, Mumbai, Maharashtra, India, Jan 2013.

[8] https://en.wikipedia.org/wiki/Radio-frequency_identification#Commerce

Program Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include <SPI.h>
#include <MFRC522.h>
#include<SoftwareSerial.h>
SoftwareSerial mySerial(A8,A9);

LiquidCrystal_I2C lcd(0x27,20,4);
#define redLed 3
#define greenLed 2
#define Buzzer 4
#define wipeB 8
#define SS_PIN 53
#define RST_PIN 5
String number_1 ="01705384555";
String number_2 ="01518398162";
String number_3 ="01711233700";
String number_4 ="01718116115";
String number_5 ="01940370935";
String number_6 ="01844083957";
String number_7 ="01677353280";
String number_8 ="01705384555";
int data;
String str="";
char incomingByte;
String inputString;

byte card_ID[4]; //card UID size 4byte
byte RFID1[4]={0xC2,0x3F,0x9E,0x0E};
//first UID card
byte RFID2[4]={0x1B,0x84,0x9C,0x22};
//second UID card
byte RFID3[4]={0xD2,0x72,0x28,0x1B};
byte RFID4[4]={0x61,0x16,0x1D,0x08};
byte RFID5[4]={0x52,0xC8,0xE4,0x0E};
byte RFID6[4]={0x37,0x97,0x59,0xE4};
byte RFID7[4]={0x23,0x4B,0xA3,0x03};
byte RFID8[4]={0x27,0x6B,0x5F,0xE4};

String Name1="Md. Jamiul";
String Id_1="EEE1802014058";
String Name2="Ujjal Majumder";
String Id_2="EEE1802014055";
String Name3="Aminul Islam";
String Id_3="EEE1802014052";
String Name4="Sagar Mondal";
String Id_4="EEE1802014066";
String Name5="Emran Hossen";
String Id_5="EEE1802014089";
String Name6="Zahidul Islam";
String Id_6="EEE1503006010";
String Name7="Mahafuz Rahaman";
String Id_7="EEE1503006240";
String Name8="Majharul Islam";
String Id_8="Lecturer, EEE";

String in_sms=" is in school";
String out_sms=" is out school";
////////////////// Setup ////////////////////////
void setup() {
```

```
  lcd.init();
  lcd.backlight();
  //Arduino Pin Configuration
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  pinMode(wipeB, INPUT_PULLUP);   // Enable pin's pull up resistor
  pinMode(Buzzer, OUTPUT);
  digitalWrite(Buzzer, LOW);    // Make sure door is locked
  digitalWrite(redLed, LOW);  // Make sure led is off
  digitalWrite(greenLed, LOW);  // Make sure led is off
  Serial.begin(9600);  // Initialize serial communications with PC
  mySerial.begin(9600);
//gsmInit();
SPI.begin();          // MFRC522 Hardware uses SPI protocol
 lcd.setCursor(4, 0);
 lcd.print("WELCOME");
 lcd.setCursor(3, 1);
 lcd.print("SU PROJECT");
delay(2000);
lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("RFID Card Based");
   lcd.setCursor(1, 1);
  lcd.print("Attend & SMS");
  delay(4000);
  lcd.clear();
   lcd.setCursor(1, 0);
  lcd.print("Place Your Card");
  delay(1000);
 ShowReaderDetails();  // Show details of PCD - MFRC522 Card Reader details
  digitalWrite(redLed, LOW);  // Make sure red LED is off
  digitalWrite(greenLed, HIGH);   // Make sure green LED is on
  delay(200);
  digitalWrite(redLed, LOW);  // Make sure red LED is off
  digitalWrite(greenLed, LOW);  // Make sure green LED is off
  delay(200);
  digitalWrite(redLed, HIGH);   // Make sure red LED is on
  digitalWrite(greenLed, LOW);  // Make sure green LED is off
  delay(200);}



/////////////// Main Loop ///////////////////////
void loop () {
for (byte i = 0; i < 1; i++) {
card_ID[i]=mfrc522.uid.uidByte[i];
if(card_ID[i]==RFID1[i] && s1f==0)
emp1in();
else if(card_ID[i]==RFID2[i]&& s2f==0)
emp2in();
else if(card_ID[i]==RFID3[i]&& s3f==0)
emp3in();
```

```
else if(card_ID[i]==RFID4[i]&& s4f==0)
emp4in();
else if(card_ID[i]==RFID5[i]&& s5f==0)
emp5in();
else if(card_ID[i]==RFID6[i]&& s6f==0)
emp6in();
else if(card_ID[i]==RFID7[i]&& s7f==0)
emp7in();
else if(card_ID[i]==RFID8[i]&& s8f==0)
emp8in();

else if(card_ID[i]==RFID1[i] && s1f==1)
emp1out();
else if(card_ID[i]==RFID2[i]&& s2f==1)
emp2out();
else if(card_ID[i]==RFID3[i]&& s3f==1)
emp3out();
else if(card_ID[i]==RFID4[i]&& s4f==1)
emp4out();
else if(card_ID[i]==RFID5[i]&& s5f==1)
emp5out();
else if(card_ID[i]==RFID6[i]&& s6f==1)
emp6out();
else if(card_ID[i]==RFID7[i]&& s7f==1)
emp7out();
else if(card_ID[i]==RFID8[i]&& s8f==1)
emp8out();
else if  denied();
}


//////// Access Denied  ////////////////////
void denied() {
 lcd.clear();
 lcd.setCursor(3, 0);
 lcd.print("Wrong Card");
 digitalWrite(redLed, HIGH);  // Make sure
led is off
 digitalWrite(greenLed, LOW);
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 delay(200);
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
delay(1000);
 lcd.clear();
 lcd.setCursor(0, 0);
  lcd.print("Place Your Card");
}

void emp1in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name1);
 lcd.setCursor(0, 2);
```

```
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_1);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_1 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name1);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_1);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s1f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card"); }

 void emp1out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name1);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_1);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_1 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name1);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_1);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s1f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card"); }
 void emp2in(){
```

```
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name2);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_2);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_2 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name2);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_2);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s2f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp2out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name2);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_2);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_2 + "\"\r"); //Mobile phone number
to send message
 delay(50);
```

```
 mySerial.print(Name2);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_2);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s2f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");  }

void emp3in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name3);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_3);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_3 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name3);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_3);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s3f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

void emp3out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
```

```
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name3);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_3);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_3 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name3);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_3);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s3f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}
void emp4in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name4);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_4);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_4 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name4);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_4);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
```

```
 s4f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp4out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name4);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_4);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_4 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name4);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_4);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s4f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

void emp5in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name5);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_5);
 lcd.setCursor(0, 3);
```

```
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_5 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name5);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_5);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s5f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp5out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name5);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_5);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_5 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name5);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_5);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s5f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

void emp6in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
```

```
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name6);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_6);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_6 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name6);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_6);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s6f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp6out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name6);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_6);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_6 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name6);
 delay(20);
```

```
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_6);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s6f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

void emp7in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name7);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_7);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_7 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name7);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_7);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s7f=1;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp7out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
```

```
 lcd.setCursor(6, 1);
 lcd.print(Name7);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_7);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_7 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name7);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_7);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s7f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

 void emp8in(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 //sendSMS1("Name1,Id_1 is in School");
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Welcome");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name8);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_8);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_8 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name8);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_8);
 delay(20);
 mySerial.print(in_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s8f=1;
```

```
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");   }

void emp8out(){
 digitalWrite(Buzzer, HIGH);
 delay(200);
 digitalWrite(Buzzer, LOW);
 digitalWrite(redLed, LOW);  // Make sure
led is off
 digitalWrite(greenLed, HIGH);
 lcd.clear();
 lcd.setCursor(4, 0);
 lcd.print("Thank You");
 lcd.setCursor(0, 1);
 lcd.print("Name: ");
 lcd.setCursor(6, 1);
 lcd.print(Name8);
 lcd.setCursor(0, 2);
 lcd.print("ID:");
 lcd.setCursor(4, 2);
 lcd.print(Id_8);
 lcd.setCursor(0, 3);
 lcd.print("Dept: EEE");
 mySerial.println("AT+CMGF=1");
 delay(50);
 mySerial.println("AT+CMGS=\"" +
number_8 + "\"\r"); //Mobile phone number
to send message
 delay(50);
 mySerial.print(Name8);
 delay(20);
 mySerial.print(", ");
 delay(20);
 mySerial.print(Id_8);
 delay(20);
 mySerial.print(out_sms);
 delay(100);
 mySerial.print((char)26);
 delay(2500);
 s8f=0;
 lcd.clear();
 lcd.setCursor(1, 0);
 lcd.print("Place Your Card");}

////////////GSM Network Communication///////
void gsmInit()
{
lcd.clear();
lcd.print("Finding Module..");
boolean at_g_flag=1;
  while(at_g_flag)
  {
mySerial.println("AT");
    while(mySerial.available()>0)
    {
      if(mySerial.find("OK"))
at_g_flag=0;
    }
    delay(1000);
  }
mySerial.println("ATE0");
lcd.clear();
lcd.print("Searching Network..");
boolean net_g_flag=1;
  while(net_g_flag)
  {
mySerial.println("AT+CPIN?");
```

```
    while(mySerial.available()>0)
    {
      if(mySerial.find("READY"))
net_g_flag=0;
      break;
    }
    delay(1000);
  }
mySerial.println("AT+CNMI=2,2,0,0,0");
   delay(1000);
mySerial.println("AT+CMGF=1");
   delay(1000);
mySerial.println("AT+CSMP=17,167,0,0");
lcd.clear();
mySerial.flush();
}
```