

A paper on

**DESIGN AND CONSTRUCTION OF AN RFID BASED AUTOMATIC ATTENDANCE
MANAGEMENT SYSTEM**

Written by:

AKPAN ISONG ANIEBIET,

Prof. Musa Abdulkadir

Department of Electrical/Electronics Engineering

University of Maiduguri, Borno State.

IJSER

ABSTRACT:

As a result of the challenges of the manual method of taking attendance in schools and colleges in Nigeria, an automated attendance system needs to be adopted. The challenges include difficulty in keeping the attendance list over a long period of time, unnecessary time wastage during writing or signing, improper documentation, students forgetting to write or sign the attendance paper, lecturers forgetting the attendance list in the classroom, students writing or signing illegally for an absentee among others. This paper implements Radio frequency identification (RFID) automatic attendance system in Nigeria educational institutions which provides the functionalities of registering students and recording attendance. This work eradicates the deficiencies associated with the manual attendance system with an automated approach implemented through Radio frequency identification (RFID) technology.

June, 2017

1. INTRODUCTION

BACKGROUND

Radio Frequency Identification (RFID), is an automatic identification technology used for retrieving from or storing data on to RFID Tags without any physical contact. An RFID system primarily comprises of RFID Tags, RFID Reader, Middleware and a Backend database. RFID Tags are uniquely and universally identified by an identification sequence, governed by the rubrics of EPC global Tag Data Standard. A tag can either be passively activated by an RFID reader or it can actively transmit RF signals to the reader. The RFID reader, through its antenna, reads the information stored on these tags when it's in its vicinity. The reader, whose effective range is based on its operational frequency, is designed to operate at a certain frequency. The operational frequency of the reader is 13.56 MHz. The RFID system is interdependent on its core components to achieve maximum efficiency and optimum performance of the application. Due to its high degree of flexibility, the system can be easily adopted for an array of applications ranging from small scale inventory cabinets to multifarious and highly agile supply chain management systems. Although, the cost of incorporating this technology has restricted its outreach, the technology promises to have untapped potential.

MOTIVATION

In most universities, lecturers take attendance by calling out the names of students, and then marking them, while, in others, teachers pass around a sheet of paper, asking students to sign in attendance sheet just next to their surnames. Both practices have their drawbacks. In the

first case, if numerous groups attend the lesson, checking all of these students by name and surname might take about 10 minutes out of each lesson; in the second case, friends of absent students may write down their names and surnames. These practices place university teachers and their institutions at considerable disadvantages when it comes to taking attendance.

PROBLEM STATEMENT

Some students do not come to classroom due to one reason or the other and because of this they do not perform well in their examination, so there is need to monitor student attendance in the classroom to enhance their academic performance. Going by the NUC (Nigerian Universities Commission) standards, students are expected to attend 70 percent of the class before they are allowed to sit for the course examination. Other problems associated with this method vary from unnecessary time wastage to improper documentation, students forgetting to put down their names on the attendance list or students writing on behalf of other students that are absent from the class. To eradicate the deficiencies associated with the manual attendance system, an automated approach is implemented through Radio frequency identification (RFID) technology.

OBJECTIVES

The main aim of this project is to overcome the drawbacks of the manual system of taking attendance by recording the attendance of students using RFID tags and cards. Each student is provided with his/her authorized and unique tag or card to swipe over the reader to record their attendance which will be written to a micro SD card. Thus, the objectives of this project are as follows:

- To design and construct a Radio Frequency based attendance system which will overcome the drawbacks of the manual attendance system.

- To analyze how the system operates using the Radio Frequency Identification techniques.
- To test the system on a few set of tags and cards and observe its limitations.

The implementation of RFID in student management will provide additional capabilities like high efficiency and overall ease in management of student attendance.

SCOPE OF STUDY

The RFID based attendance system is intended for use in schools, colleges and institutions, but could also be easily modified for use in places where attendance taking is required, for example recording of employee attendance in industries, companies, hospitals, banks and so on. The system could also be modified for use in examination halls to replace the conventional system of signing attendance while examination is ongoing. Generally, the system could be used anywhere secure and limited access to a place is required.

2. METHODOLOGY

The RFID based automatic attendance system incorporates the use of a microcontroller (Arduino Mega 2560); RFID devices like an RC522 13.56MHz RFID reader, RFID tags; a 16x2 LCD; a buzzer and a 9v power supply. The block diagram in Fig 1 below shows how the various components are interfaced with the arduino microcontroller.

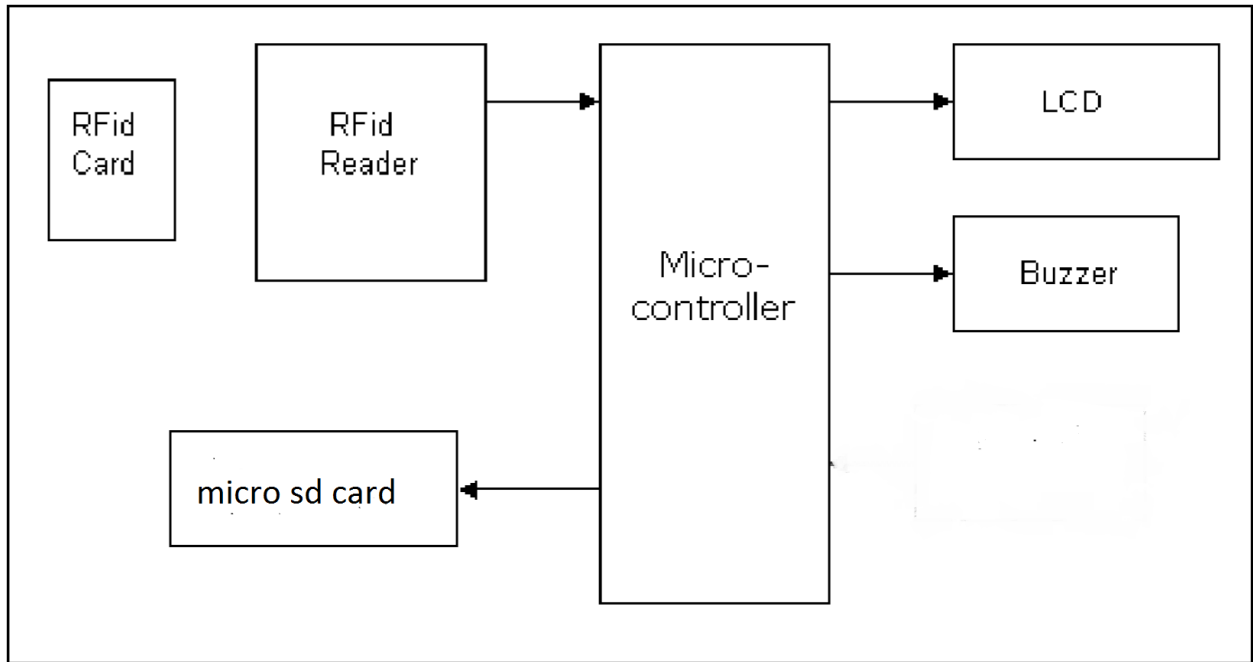


Fig 1 Block diagram description for RFID based attendance system.

The primary purpose of the Radio Frequency Identification system is to register student attendance wirelessly by a mobile device called a tag, which is read by an RFID reader. Tags and cards of 13.56MHz frequency were used. Also, an RC522 Arduino RFID reader which is of same frequency was used.

The circuit simulation for the interfacing of the Liquid Crystal Display (LCD) with the buzzer with the Arduino microcontroller were done in PROTEUS 8 PROFESSIONAL (with coding done in Arduino compiler) before the actual hardware implementation so as to avoid any errors during the complete circuit design and implementation, this simulation is shown in Fig 2 below.

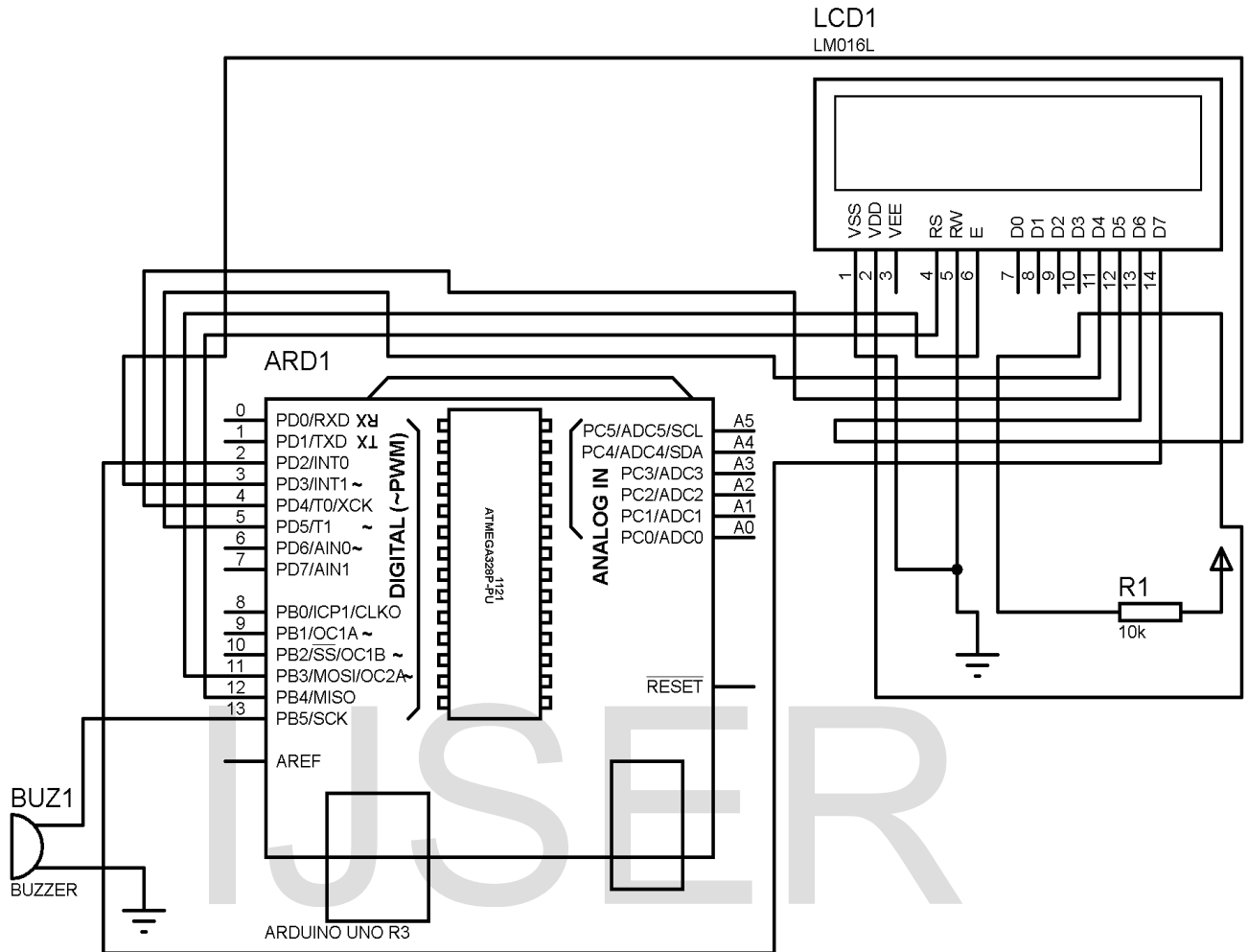


Fig 2 Simulation of LCD and Buzzer with Arduino

UNITS OF THE RFID ATTENDANCE SYSTEM

The system provides solution through a hardware design consisting of four different units with each unit being interfaced with the Arduino Mega 2560 microcontroller. These units are listed below:

- The Radio frequency Identification reader unit
- The Liquid Crystal Display unit.

- The Micro SD reader unit
- The buzzer unit

Each of these units and the models used could be seen in the circuit diagram in Fig 3 shown below. The units will be elaborated upon in subsequent subsections.

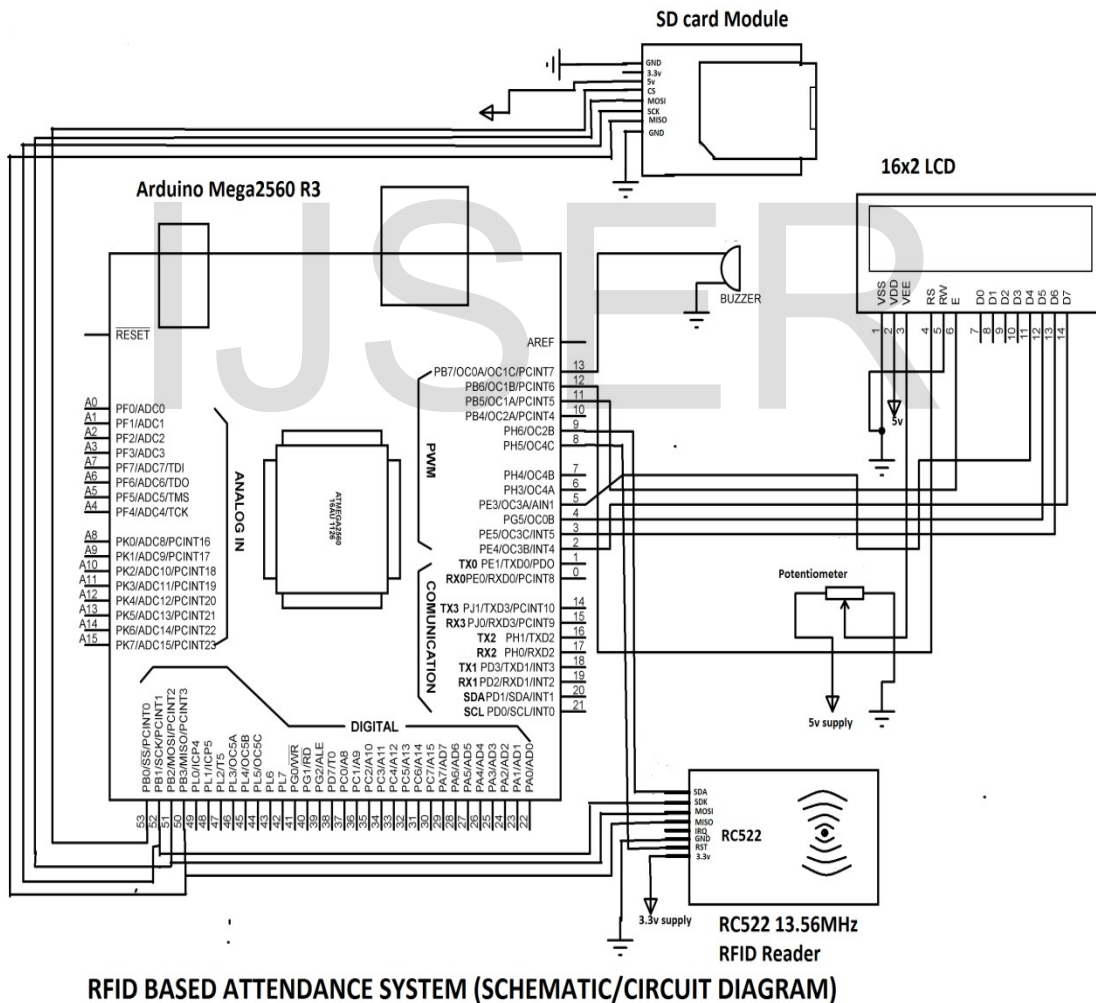


Fig 3 Circuit Diagram for the RFID Based Attendance System

I. The Radio frequency Identification reader unit

This unit consist of the MIFARE RC522 Arduino RFID reader/writer (see Fig 4 below) which can read and write to RFID tags and cards. The reader uses Serial Peripheral Interface (SPI) for communicating with the microcontroller. SPI, is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors and SD cards. The chip of the reader supports Inter-Integrated Circuit (I2C) which is a multi-master, multi-slave, packet-switched Serial bus invented by Philips typically used for attaching lower-speed peripheral ICs (the RC522 IC in this case) to processors and microcontrollers in a short-distance communication. It also supports Universal Asynchronous Transmitter and Receiver (UART) protocols, that is, its data format and transmission speeds are configurable.



Fig 4. MIFARE RC522 RFID reader/writer.

The reader communicates with the cards and tags using a 13.56MHz electromagnetic field frequency, that is the tags and cards are ISO 14443A standard tags.

II. The Liquid Crystal Display (LCD) unit.

This unit is the display unit of the device, it consist of a 16x2 LCD. So named because it has 16 Columns and 2 Rows. Also there are several other combinations available in the market, this size was selected because it is the smallest available size that could accommodate the number of characters intended to be displayed. The LCD is a 16-pin type as seen in Fig 5 below.



Fig 5. Liquid Crystal Display Unit (16x2)

It consists of: two source pins, that is, supply and ground; three control pins for controlling contrast, reading/writing and enabling reading/writing; 8 data pins used to send data to the LCD although not all were used in this attendance system; and two LED pins for illuminating the LCD (supply and ground pins).

III. The Micro SD reader unit

This unit is the memory unit of the system, it consists a 2GB Micro SD card with its Adapter and an Arduino SD Card Module as seen in Fig 6 a and b. Daily attendance is written to

and saved on the SD card. The Micro SD card is inserted into the Adapter before inserting into the Arduino SD card Module. The Arduino SD card module is an 8 pin module which uses a dc supply voltage (Vcc) of 3.3V supplied from the Arduino's 3.3V pin. Although the space required for this project is negligible compared to the size of the Micro SD card itself, the card is still used as sizes available in the market are of similar ranges. In the Micro SD card, the attendance is written to a text file named from within the arduino source code.

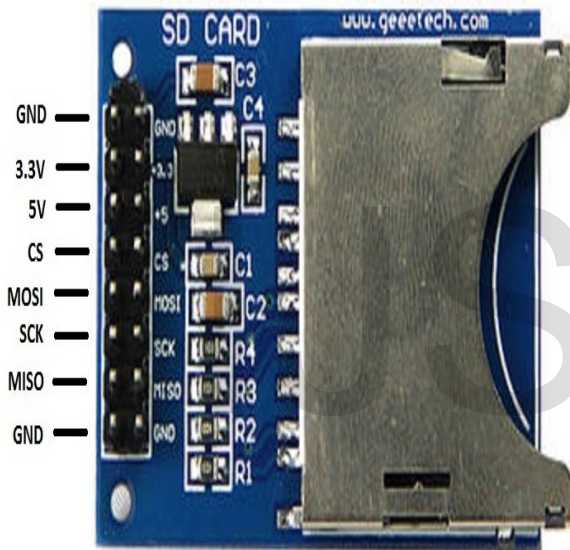


Fig 6 a. SD Card Module



Fig 6 b. Micro SD Card Adapter

IV. The Buzzer Unit

This unit the sound producing section of the arduino which consists of a 5V buzzer (seen in Fig 7 below with its supply coming from one of the analog pins of the arduino, this pin can be seen in the circuit diagram in Fig 3. The aim of including a buzzer in the project is to produce

sound whenever a card is swiped, so the user can know whether or not his card has been seen by the device. It also produced sound at the start up of the system.



Fig 7 Buzzer (5v dc)

Circuit Simulation

Simulation for some of these units was carried out in PROTEUS 8 PROFESSIONAL as a complete simulation of the circuit diagram seen in Fig 3 was not possible as some libraries such as that of the Arduino RC522 RFID reader is not available. Nevertheless, a simulation for interfacing the Liquid Crystal Display (LCD) and the buzzer with the arduino was carried out and this can be seen in Fig 1.

Coding

Complete coding of the system was carried out within the Arduino Compiler precisely version **1.6.5**. The behavior of the system at any point in time was frequently observed in the serial monitor of the arduino compiler. Some of the libraries used for the project such as the LCD library already exist as default libraries within the compiler although some were custom

libraries added to the compiler such as the RC522 RFID library. The libraries used for this project include the:

- i. LiquidCrystal.h library for the LCD.
- ii. SoftwareSerial.h library for the arduino's Serial Monitor.
- iii. SPI.h library for Serial Communication.
- iv. RFID.h library for the RC522 RFID reader.
- v. SD.h library for the SD card.

Most of the functions used in the source code were first written in CodeBlocks (a compiler for the C++ and the C language) before writing them in the arduino compiler. This is because the syntax of the arduino language is similar to C++ although it has some differences. For example using the keyword 'string' (which is valid in the C++ language) in place of 'String' creates a bug when written in the arduino compiler. Some other C++ keywords and functions are changed in the arduino language although having similar names in the arduino language.

The complete source code and complete description of these libraries can be seen in the appendix.

3. DESIGN AND IMPLEMENTATION

This section explains the design and implementation of the various units of this project.

The system consists of three phases:

- The circuit design
- The circuit implementation
- The Algorithm, Flow Chart and coding phase

The circuit diagram as seen in Fig 3 is being implemented on a breadboard/protoboard, (Fig 8 (a)) using jumper wires as seen in Fig 8 (b) below, and is not transferred to a veroboard, hence no permanent soldering of any component, this is done so as to allow for easy modification of the project as the project is a prototype version that is limited to only two students.

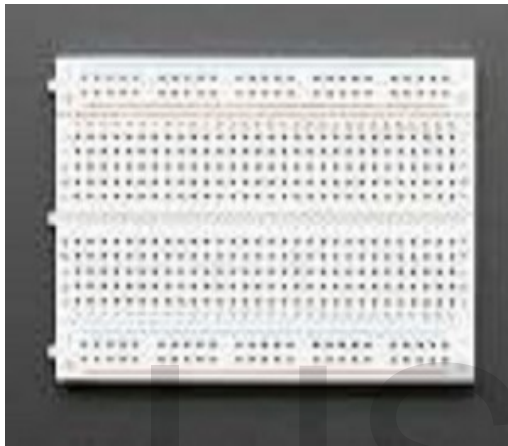


Fig 8 (a) Breadboard/Protoboard



(b) Jumper Wires

The system consists of four individual units (as explained in chapter three) and a microcontroller unit serving as a central unit to the four units. These four units which include the Radio frequency Identification reader unit, the Liquid Crystal Display unit, the Micro SD Reader unit, and the buzzer unit are all interfaced with the Microcontroller unit using connecting/jumper wires. The functionality of each of these units was verified individually before integrating all the units on the breadboard. This was done by interfacing each major component of each unit with the microcontroller so as to ensure that the component is fully functional, so as to avoid unintended results as a result of a faulty component.

Circuit Design Phase

The design of the circuit in Fig 3 was fully drawn inside PROTEUS 8 PROFESSIONAL using its 2D graphics feature. The circuit was not fully simulated due to lack of some Arduino libraries. As in the case of the RC522 RFID Reader unit and the SD card unit. Other units, that is, the Liquid Crystal Display (LCD) unit and the buzzer unit were simulated in PROTEUS before implementation on the protoboard/breadboard.

The four units and the pin configurations of their major components with the Arduino are seen in the project circuit diagram. The schematics of the design of each of these units can be seen in the figures in the following subsections.

LCD Interfacing/Configuration

The LCD used as said earlier is a 16x2 alphanumeric display LCD. The Pinout and Pin Description of the 16x2 LCD Module used can be seen in Table 1 below:

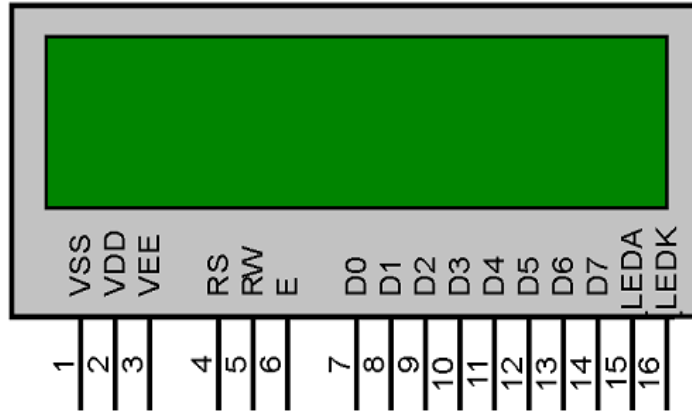


Fig 9 LCD Pinouts

S/N	Pin Number	Pin Name	Pin Type	Pin Description	Pin Connection
1	Pin 1	Ground	Source Pin	This is a ground pin of the LCD	Connected to the ground of the MCU power source
2	Pin 2	Vcc	Source Pin	This is the supply voltage pin of the LCD	Connected to the supply pin of the power source
3	Pin 3	V ₀ /V _{EE}	Control Pin	Adjusts the contrast of the LCD	Connected to a variable pot of source 0-5v
4	Pin 4	Register Select	Control Pin	Toggles between command/data a register	Connected to the MCU and gets 0 or 1: 0 – command mode

					1 – data mode
5	Pin 5	Read/Write	Control Pin	Toggles LCD between Read and Write operation	Connected to the MCU and gets 0 or 1: 0 – Write operation 1 – Read operation
6	Pin 6	Enable	Control Pin	Must be held high to perform R/W operation	Connected to the MCU and always high
7	Pin 7-14	Data Bits (0-7)	Data/Command Pin	Pins used to send command/data to the LCD	<u>In 4wire mode</u> , only four pins (0-3) are connected to the LCD <u>In 8wire mode</u> , only eight pins (0-7) are connected to the LCD
8	Pin 15	LED positive	LED Pin	Normal LED like operation to illuminate the LCD	Connected to +5V
9	Pin 16	LED negative	LED Pin	Normal LED like operation to illuminate	Connected to ground

				the LCD connected with GND	
--	--	--	--	----------------------------------	--

Table 1. Pinouts and Pin Description Of a 16x2 LCD

The LCD configuration with respect to the Arduino is shown in Table 2 below:

S/N	LCD PIN	CONNECTION
1	Pin 1	Ground of Arduino
2	Pin 2	5V from Arduino
3	Pin 3	Grounded with 1.47K resistor
4	Pin 4	Arduino Digital pin 12
5	Pin 5	Ground of Arduino
6	Pin 6	Arduino Digital pin 11
7	Pin 7	NC
8	Pin 8	NC
9	Pin 9	NC
10	Pin 10	NC
11	Pin 11	Arduino Digital pin 5
12	Pin 12	Arduino Digital pin 4
13	Pin 13	Arduino Digital pin 3
14	Pin 14	Arduino Digital pin 2
15	Pin 15	5V from Arduino with 220ohm resistor
16	Pin 16	Ground of Arduino

Table 2 LCD Configuration With Respect To The Arduino

Pin 3 of the LCD is used to regulate the contrast of the display and hence the 1.47K ohm resistor could be replaced by a variable potentiometer so as to toggle the contrast, but this is not necessary in this project, hence, pin 3 was grounded with a 1.47K resistor, that is full contrast. The brightness could also be regulated by similarly replacing the 220ohm resistor at pin 15 of the LCD with a required more suitable value or potentiometer.

RC522 RFID Reader Module Interfacing/Configuration

The low cost MFRC5222 reader module (seen in Fig) is easy to use and can be used in a wide range of applications as already explained in the previous chapter. The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56MHz as stated in previous chapters.

Features of this reader include the following:

- MFRC522 chip based board
- Operating frequency: 13.56MHz
- Supply Voltage: 3.3V
- Current: 13-26mA
- Read Range: Approx 3cm
- Serial Peripheral Interface
- Max Data Transfer Rate: 10Mbps
- Dimensions: 60mm x 39mm

Before the MFRC522 RFID reader with the Arduino, a custom library <RFID.h> was added

into the Arduino compiler's libraries as it is not inbuilt in the compiler at preinstallation. The wiring of the MFRC522 varies depending on the type of Arduino used. In this case, the Arduino Mega2560 was used. Hence, wiring of the MFRC522 RFID reader (all its 8 pins included) with the Arduino (for Arduino Mega only) is shown in Table 3 below.

S/N	MFRC522 Pin	Arduino Mega Pin
1	SDA (SS)	D53
2	SCK	D52
3	MOSI	D51
4	MISO	D50
5	IRQ	Not Connected (NC)
6	GND	GND
7	RST	D9
8	3.3V	3.3

Table 3 Arduino Wiring With MFRC522 RFID Reader

The schematics for the wiring with the MFRC522 reader is shown in Fig 10 overleaf.

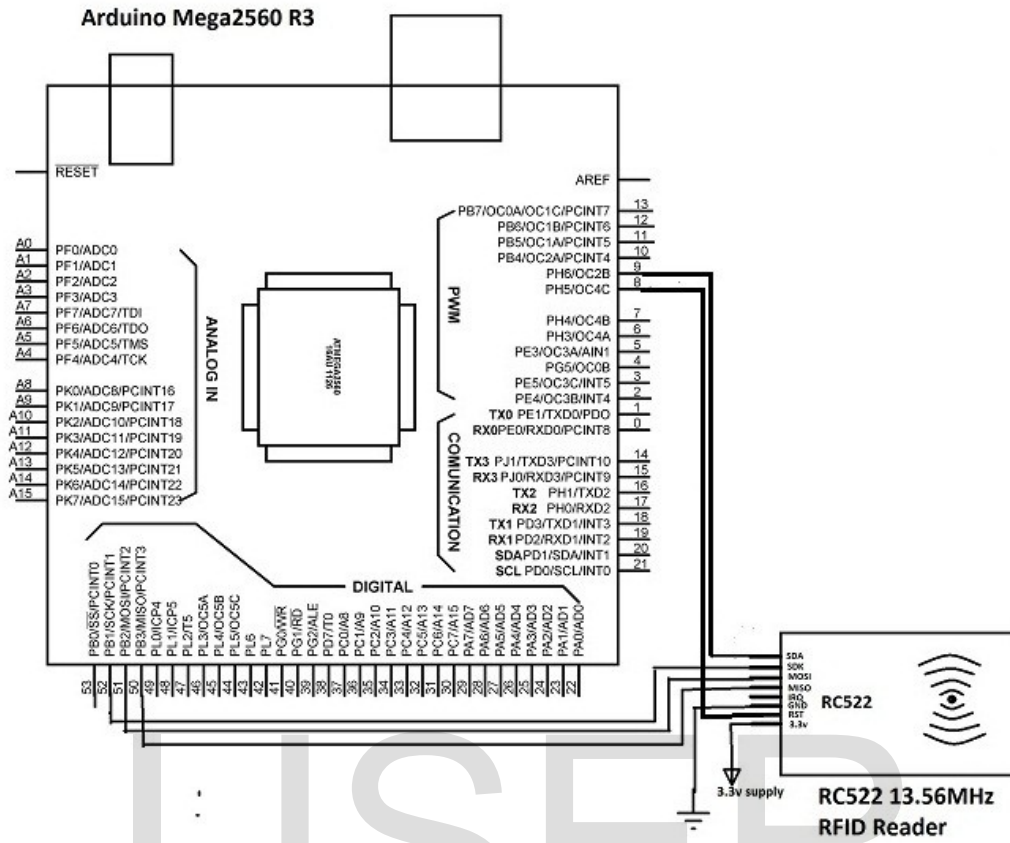


Fig 10 Schematics For Arduino Wiring With MFRC522 RFID Reader

SD Card Module Interfacing/Configuration

The Arduino SD Card Shield (see Fig 5) serves as a simple solution for transferring data to and from an SD Card. The pinout is directly compatible with Arduino but can also be used with other microcontrollers. The module allows for adding mass storage and data logging capabilities to the project. The device is an 8 pin device, 6pins are used in this case and its pin configurations with the Arduino are shown in Table 4 below:

S/N	SD Module Pin	Arduino Mega Pin
-----	---------------	------------------

1	MISO	50
2	MOSI	51
3	SCK	52
4	CS/SS	53
5	5V	5V
6	GND	GND

Table 4 Arduino Pin Configuration with an SD Card Module

The Schematics for interfacing the SD Card Module with the Arduino Mega is shown in Fig 11 below:

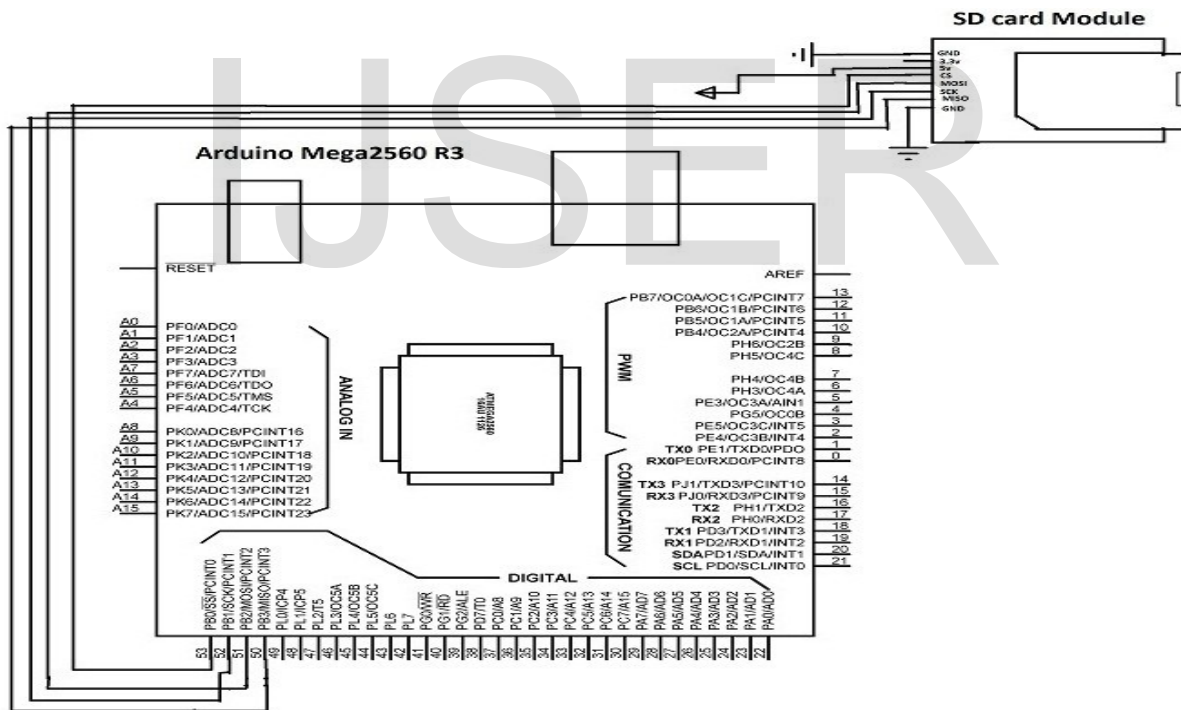


Fig 11 Schematics for interfacing the SD Card Module with the Arduino Mega

Buzzer Interfacing/Configuration

The buzzer used is a two terminal type that uses 5V dc supply gotten from Pin 13 (a PWM pin) of the Arduino. The buzzer is programmed to sound whenever an RFID card or Tag is brought in contact with the RFID reader. The altitude of the sound produced by the buzzer is regulated via the code to a suitable hearing range for the human ear.

The two terminals (Supply and Ground) of the buzzer are connected to Pins 13 and GND of the Arduino respectively. The schematic of the buzzer being interface with the Arduino is shown in Fig 12 below.

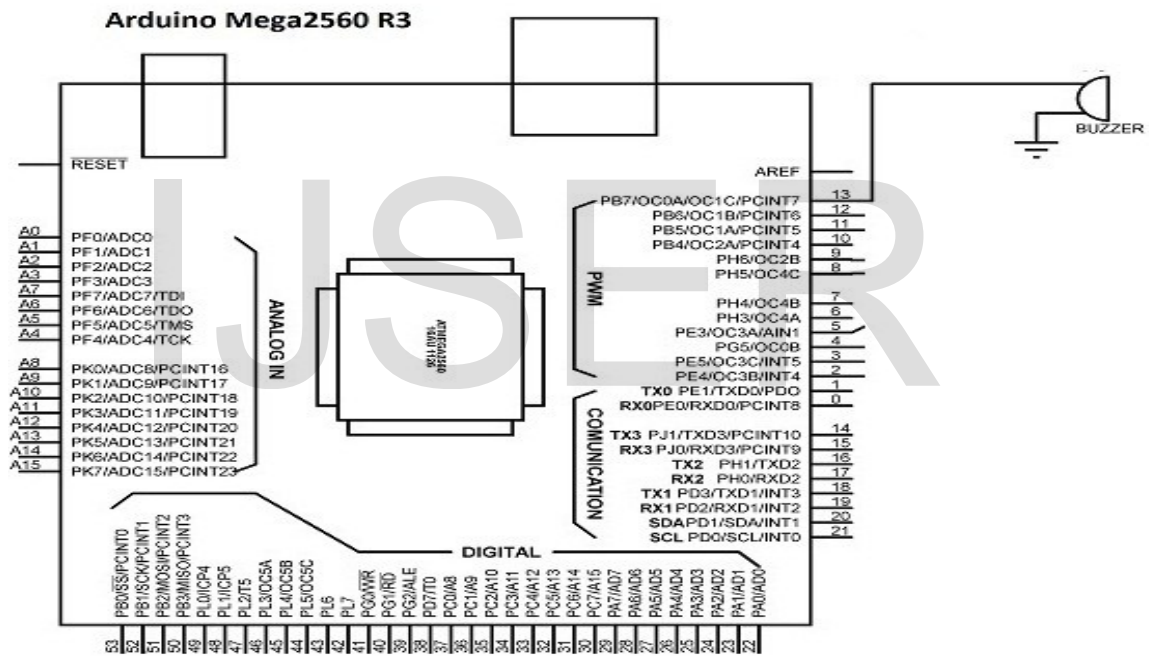


Fig 12 Schematic Of The Buzzer Being Interface With The Arduino

Circuit Implementation Phase

The circuit diagram as seen in the Fig 3 in the previous chapter was fully implemented on a breadboard/protoboard after individually testing all the main components of the various units explained in the section.

The Liquid Crystal Display (LCD) was first placed on the breadboard and interfaced with the microcontroller as it had the most amount of pin configurations, this was done to ease the process of setting up other components of the system so as to avoid mismatching pins of the Arduino Mega and any other component. After placing the LCD, the contrast was varied using the potentiometer attached to pin 3 so as to adjust it to a very suitable value that can be easily read by anyone. The LCD was set to display a welcome message and give some timing information at initialization stage, after which it requests card swipes.

After mounting the LCD and testing out the display using few LCD codes (for example: HelloWorld, and ScrollDisplay.), the MFRC522 RFID Reader was mounted and its various pins connected with the arduino, cards were then used to test the reader to ensure full functionality before mounting anyother component as this was the most vital component in the project.

As soon as the MFRC522 Reader was fully mounted, the next most vital component was the SD Card Module for reading and writing daily attendance to the SD card. The module was mounted with its pins fully configured and a 2GB SD Card inserted using an SD Card Jack. The SD Card and the MFRC522 shared some I²C pins, that is pins 50, 51, 52, and 53 of the Arduino Mega. Hence the I²C pins of both components were linked via the same bus to the mentioned pins of the Arduino Mega. The SD Card was first tested using the CardInfo example in the SD library of the Arduino Compiler to ensure the SD Card was functional, it was then tested with

ListFiles, so as to ensure the SD Card wasn't creating some garbage files on its own. Then reading and writing to the SD Card was tested using some read/write examples in the compiler.

The buzzer was the final component mounted on the breadboard as it had only two pin configurations: a supply and ground connection. It was tested using the analogWrite() and digitalWrite() generic, functions of the arduino compiler.

Algorithm, Flow Chart And Coding Phase

Coding the RFID BASED ATTENDANCE SYSTEM is the most tedious part of the project. The code used for this system is over 400LOC (0.4KLOC). As said earlier, most of the functions used in the code were first of all written in C++ using the CodeBlocks compiler as the Arduino language has a similar syntax to the C++ language although with a lot of syntax differences. This section consists of a three main subsections, which include the algorithm, the flow chart, and the coding phase. The program requirement is first stated in the first subsection.

I. Program Requirement:

The system is required to accept the lecturers card only (called the mastercard) at the beginning and end of the attendance period or class, this is to mark the beginning and end of an attendance period. Registered students only are allowed to take attendance with the system, any invalid card or tag swiped will have no effect on the system. The buzzer should make a buzz each time a card is swiped (valid or invalid/registered or not registered). When a student swipes a card more than once, he or she should see his or her cumulative attendance for the class. Daily student attendance should be written to an SD Card. Attendance should be available for at most twenty classes. A reset Card Should be available for resetting the system at any point in time the device is powered on.

II. Algorithm For The RFID Based Attendance System

The algorithm for the RFID based attendance system is written as follows:

- i. Swipe the reset card if the system should be reset
- ii. Swipe the lecturers card or mastercard to allow students access
- iii. Make a buzz for any card/tag swiped (valid/invalid)
- iv. Do nothing if invalid card/tag is swiped
- v. Give student access to system if RFID card/tag swiped is the lecturers (is a mastercard).
- vi. Swipe students card/tag to register attendance
- vii. Display student cumulative attendance if card/tag is swiped more than once.
- viii. Swipe lecturers card to end attendance for the day.
- ix. Reiterate the process for twenty classes

III. Flow Chart Of The RFID Based Attendance System Program

The flow chart of the system is shown in Fig 13 overleaf.

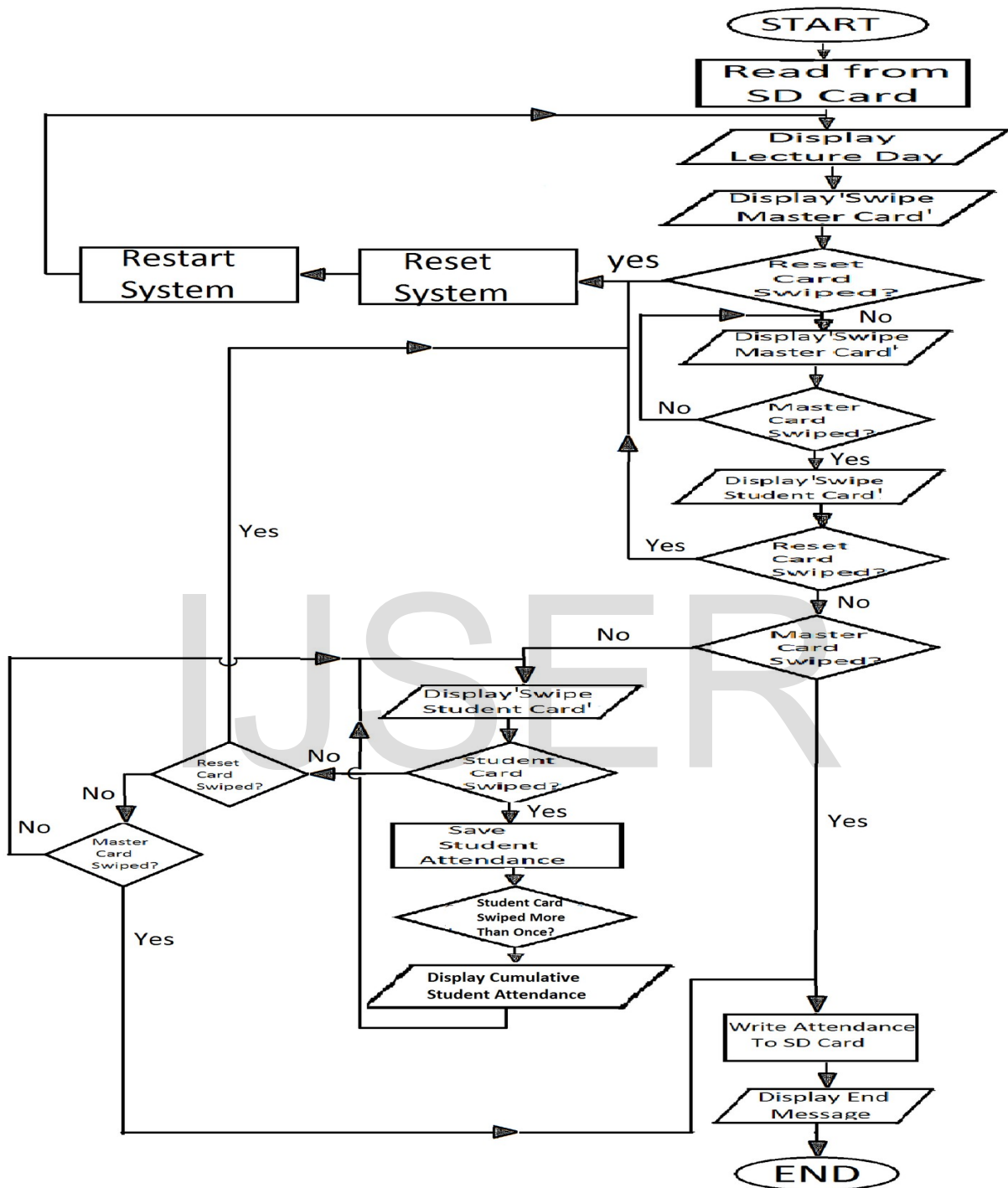


Fig 13 Flow Chart Of the RFID Based Attendance System

IV. Coding Of The RFID Based Attendance System Program.

The arduino Integrated Development Environment (IDE) was the only compiler used for complete programming of the system. Libraries used include some generic libraries which are available in the compiler at pre-installation phase and some Custom libraries downloaded from the internet and added to the compiler's libraries folder.

Generic libraries used in this project include the following:

- i. SoftwareSerial library ("SoftwareSerial.h" for viewing the arduino Serial monitor)
- ii. LCD library ("LiquidCrystal.h" for the LCD interfacing)
- iii. SD Card Library (SD.h) for fully accessing the memory card.

Custom libraries added include:

- i. Serial Peripheral Interface Library (SPI.h) for SPI communication .
- ii. RadioFrequency Identification Library (RFID.h) for the MFRC522.

The Serial monitor was used to observe the behavior of the system during development, it assists in ensuring the correctness of the code and verification of the response of the hardwares to the code, hence it is a very useful tool when developing a system using arduino.

A number of functions were created in this phase, these functions served various vital purposes in the programming of the device. Functions created include:

- `bool check_master_id(String card_num)`: a function which takes in an RFID card number and checks if it is the Master Card (Lecturers Card) by returning a 'true'.
- `bool check_reset_id(String card_num)`: a function which takes in an RFID card number and checks if it is the Reset by returning a 'true'.

- `int check_card(String card_num)`: a function which checks an RFID card for its registered ID number.
- `void set_att()`: a function that sets the daily attendance of a student, 1 for present, 0 for absent.
- `int charToInt(char char1, char char2)`: a function that converts two characters into a single integer number
- `char* intToChar(int number)`: this function returns an array of characters in place of a number, it does the opposite of `charToInt()`.
- `String changeString(String strToChange, char fir, char sec, char value)`: this function will take first two characters and return a string with the next character changed.
- `void init_sd()`: for initializing the SD Card.
- `void get_str()`: reads lines from the SD Card.
- `void get_day_and_totals()` : get the present day and cumulative attendance of students
- `void set_next_day()`: sets the next day in the SD Card. Used after the day's attendance have been recorded.
- `void write_att()`: Write Attendance to the SD Card. Used after the day's attendance have been set.
- `void set_new_total()`: for updating the new total
- `void del_n_rec_file()`: for deleting and recreating the SD Card file
- `void reset_system()`: for resetting the system. It works only with the reset card.

The complete code for RFID Attendance System can be seen in the appendix at the end of the write up.

4. SUMMARY, CONCLUSION AND RECOMMENDATION

SUMMARY

The main aim of carrying out this project is to overcome the drawbacks of the manual system of taking attendance by recording the attendance of students using RFID tags and cards. The system utilizes RFID technology in ensuring that attendance taking in schools, institutions and other settings where attendance taking is required is automated by assigning RFID tags and cards to registered users.

Although other works exist that utilize RFID technology, this system has a very short operating time compared to most works on this technology, in that a single user can get registered within a second as he/she is only required to swipe the card over the high frequency RFID reader (13.56MHz) to get registered.

The System comprises about four units as explained in earlier chapters with the microcontroller unit serving as the a main/control unit to the rest of the units and the buzzer unit set to make a buzz whenever a card is swiped so as to ensure the sensitivity of the reader to the card swiped. The circuit design for the system was drawn in PROTEUS 8 PROFESSIONAL, with some component parts such as the Liquid Crystal Display and Buzzer simulated first.

The circuit diagram seen in chapter three was fully implemented on a protoboard/breadboard as the design is only a laboratory prototype. Complete coding of the system was carried out in the Arduino compiler with the Codeblocks compiler serving as a test compiler although with different syntax from arduino. The full code of the system can be found in the appendix section of this write up.

CONCLUSION

The design and implementation of a RFID based automatic attendance system which is the aim and objective of this paper was successfully implemented and can be seen in Fig 14 below. This system provides an effective and more convenient method of taking attendance when compared to the manual system. The system was tested with a single user tag and a fully observed. The system was programmed for twenty lecture days as stated in the earlier chapter. And can be easily reprogrammed for more lecture days by altering a single line of code. More students can be registered to the system by reprogramming the system to do so .

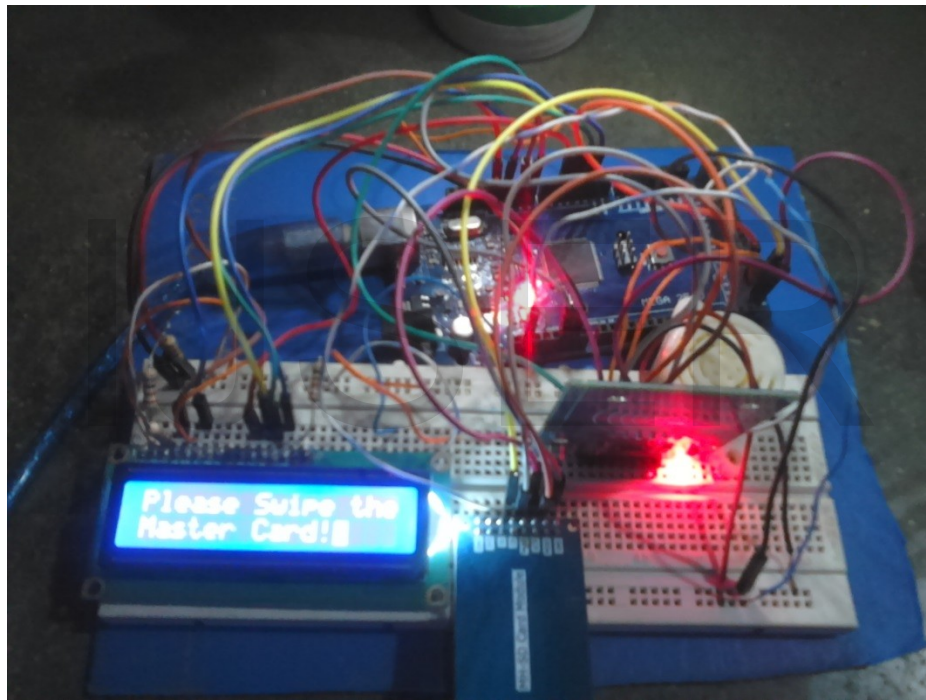


Fig 14 RFID Based Attendance System

As the RFID technology evolves, more sophisticated applications will use the capability of RFID to receive, store and forward data to a remote sink source. RFID has many applications as can be imagined. In this paper, I have utilized the versatility of RFID in implementing functional and automatic student course attendance recording system that allows students to

simply fill their attendance just by swiping or moving their ID cards over the RFID reader which are located at the entrance of lecture halls. I hope that this system can shift the paradigm of students' lecture attendance monitoring in classroom and provide a new, accurate, and less cumbersome way of taking student attendance in Nigerian Higher Institutions.

RECOMMENDATION

Every good engineering design innovation has limitations. This passive RFID based lecture attendance monitoring system is not without limitation as a data collection technology with accurate and timely data entry, the system should be limited to a few hundreds of users as there is no real database incorporated to it. Hence, the limitation of this design would be improved upon in future by considering the following salient recommendations:

- i. By incorporating a facial/biometric recognition application that would serve to further increase the security of the system against impersonation by erring students.
- ii. Incorporation of an online MySQL database to the system so it can be used for institutions/organisations with very large number students/employees.

REFERENCES

1. A. Parvathy, V. R. Raj and M. Reddy, (2011). "RFID based exam hall maintenance system", IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications (AIT).
2. B. M. Stephen, E. S. Johnson and R. W. John, (2012). "RFID Security Technology and Applications", Cambridge University Press.
3. Brian Evans, (2007). Arduino Notebook v1.1.
4. C. E. Geoffrey, (2012). "Automatic Access Control System using Student's Identification Card based on RFID Technology", Unpublished Thesis, Faculty of Electrical Engineering, University of Teknologi, Malaysia.
5. Chitresh S. and Amit K. (2010). "An efficient Automatic Attendance Using Fingerprint Verification Technique", International Journal on Computer Science and Engineering (IJCSE), Vol. 2 No. 2, pp. 264-269.
6. D. Zhang and A. K. Jain, (2004). "RFID BASED PASSPORT SYSTEM", First International Conference on Biometric Authentication (ICBA), Hong Kong.
7. Goankhar M. and Uppala Mamatha, (2009). — RFID Based Device Control And Authentication - Proceedings of ASCNT, CDAC, Noida, India.
8. K. D. Mahajan, P. Pandey and B. K. Pandher, (2010). "Application of RFID Technology in Libraries and Role of Librarian". pp. 109-118.
9. M. P. Mojares, G. A. Litan and J. G. Mojares, (2013). An SMS and RFID-Based Notification System of Lipa City Colleges", Journal of Applied Global Research, vol. 6, no. 18.
10. Maltoni D., D. Maio, A. K. Jain, S. Prabhaker (2003), "Handbook of Fingerprint Recognition", Springer, New York, pp. 13-20.

11. Mohamed A. B., Abdel-Hamid A. and Mohammed K. Y, (2009). "Implementation of an Improved secure system detection for E passport by using EPC RFID tags", World Academy of Science, Engineering and Technology Journal, Volume 6,pp1-5.

IJSER

Appendix

Complete Arduino Source Code For The RFID Based attendance

System

```
/*Source Code For RFID Based Attendance System,  
* written by Aniebiet I. Akpan  
* this is a sample code for two students.  
* and can be modified for more students.  
*/  
  
/* Include the standard Arduino SPI library */  
#include <SPI.h>  
/* Include the RFID library */  
#include <RFID.h>  
  
/* Define the DIO used for the SDA (SS) and RST (reset) pins. */  
#define SDA_DIO 9  
#define RESET_DIO 8  
/* Create an instance of the RFID library */  
RFID RC522(SDA_DIO, RESET_DIO);  
  
/*include the SD lib*/  
#include <SD.h>  
File myFile;  
  
#include <LiquidCrystal.h>  
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
//get string from SD  
char holder; //hold each character  
String line; //hold each line from the file  
int num_lines = 0;  
  
//today  
//get_day_and_totals variables  
char day_c1, day_c2; //get day from file, save chars  
int today;  
//next day  
int next_day;  
String n_day_c;  
  
//totals for each id  
int total[] = {0, 0};  
int total_ct = 0;  
  
//numbers for charToInt  
int num1, num2;  
  
//characters from intToChar  
char numb_ret[2];
```

```
//changeString variables
char a[2]; //conv char to String
char b[2]; //convert char to String
int f_index;
String _s;
String str_ret;

//strings to hold card number from serNUM
unsigned char temp1[5];
String temp2 = "";
String card_num = "";

int i = 0;

//master card id
String master_card = "382263999128";
int m_card_swiped = 0; //m_card swiped
int st_once = 0; //student swiped once

//card numbers are stored here
String card_nums_holder[] = {"532498109169", "1808524411796", "1197522643245"};
String id_1 = "532498109169";
String id_2 = "1808524411796 ";
String res_card = "1197522643245"; //the reset card

//student attendance for today
int attendance[] = {0, 0};

int id_swiped = 0;
int prev_id_swiped;

void setup() {
//buzzer pin
pinMode(13, OUTPUT);

Serial.begin(9600);
SPI.begin(); //enable SPI interface
RC522.init(); //Initialise the RFID reader

//initialise the sd card
init_sd();
get_str(); //get line from file

get_day_and_totals(); //get day and totals
set_next_day(); //set next day

Serial.println(line);

Icd.begin(16, 2);
Icd.print("FINAL YR PROJECT");
Icd.setCursor(5,1);
Icd.print("EEE599");
delay(3000);
Icd.clear();
```

```
lcd.print("ID: 12/05/05/031");
delay(3000);
lcd.clear();
lcd.setCursor(2,0);

for(int k=0; k<2; k++){
lcd.print("Welcme To Dept of EEE Attendance System");
for (int positionCounter = 0; positionCounter < 30; positionCounter++) {
lcd.scrollDisplayLeft();
delay(200); //initially 200
}
lcd.clear();
}
delay(2000);
lcd.print("initialising...");
lcd.setCursor(2,1);
lcd.print("please wait...");
lcd.blink();
delay(3000); //initially 3000

//ready
lcd.clear();
lcd.print("Ready in 5secs>");
for(int ct=5; ct>-1; ct--){
lcd.setCursor(15,1);
lcd.print(ct);
lcd.cursor();
delay(1000); //initially 1000
}

//today
lcd.clear();
lcd.print("today is Lecture ");
lcd.setCursor(3,1);
lcd.print("Day ");
lcd.setCursor(7,1);
lcd.print(day_c1);
lcd.setCursor(8,1);
lcd.print(day_c2);
delay(5000); //initially 5000
lcd.clear();

//delay
lcd.setCursor(2,1);
lcd.print("please wait...");
delay(3000); //initially 3000

lcd.clear();
lcd.print("Please Swipe the");
lcd.setCursor(0,1);
lcd.print("Master Card!");
}

void loop() {
/* Has a card been detected? */
```

```
if (RC522.isCard())
{
/* If so then get its serial number */
RC522.readCardSerial();
Serial.println("Card detected:");
analogWrite(13, 150);
delay(200);
analogWrite(13, 0);
delay(200);
for(i=0;i<5;i++)
{
temp1[i] = RC522.serNum[i];
//Serial.print(RC522.serNum[i]);
//Serial.print(RC522.serNum[i],DEC);
temp2 += temp1[i];
if(i == 4){
card_num = temp2;
temp2 = "";
}
}
Serial.println(card_num);
if(check_master_id(card_num) == true){
m_card_swiped++;
lcd.clear();
if(m_card_swiped == 1){
lcd.print("Access Grant!");
Serial.print("master swiped!");
lcd.setCursor(0,1);
lcd.print("please Wait... ");
delay(3000);
}
lcd.clear();
lcd.print("swipe student");
lcd.setCursor(0, 1);
lcd.print("Card: ");

if(m_card_swiped > 1 && st_once >= 1){
write_att();
set_new_total();
del_n_rec_file();

lcd.clear();
for(int c=0; c<10; c++){
lcd.print("**Thank you and goodbye!!!**");
for (int positionCounter = 0; positionCounter < 24; positionCounter++) {
lcd.scrollDisplayLeft();
delay(500);
}
}
lcd.clear();
}
}
}
else if((check_reset_id(card_num) == true))
reset_system();
```

```
else if((check_card(card_num) != 0) && (m_card_swiped >= 1)){ //not master_card(is a
student card) or invalid
st_once++;
if(st_once == 1){
prev_id_swiped = id_swiped;
set_att();
//prev_id_swiped = id_swiped;
lcd.setCursor(6,1);
Serial.print("saved");
Serial.print(attendance[id_swiped-1]);
lcd.print("saved  ");
delay(1000);
lcd.setCursor(6,1);
lcd.print("  ");
return;
}
else if(st_once > 1 && prev_id_swiped == id_swiped){
lcd.setCursor(6,1);
lcd.print("TOT ATT:");
lcd.setCursor(14,1);
if(id_swiped == 1){
lcd.print(total[0] + 1);
}
else if(id_swiped == 2){
lcd.print(total[1] + 1);
}
delay(1000);
lcd.setCursor(6,1);
lcd.print("  ");
return;
}
else if(st_once > 1 && prev_id_swiped != id_swiped){
st_once = 1;
prev_id_swiped = id_swiped;
set_att();
lcd.setCursor(6,1);
Serial.print("saved");
Serial.print(attendance[id_swiped-1]);
lcd.print("saved  ");
delay(1000);
lcd.setCursor(6,1);
lcd.print("  ");
return;
}

Serial.print("STUDENT Card Swiped");
}
Serial.println();
Serial.println();
}
delay(100);
}
//check if master card is swiped
bool check_master_id(String card_numb){
if(card_numb == master_card)
```

```
return true;
else
return false;
}
//check if reset card is swiped
bool check_reset_id(String card_num){
if(card_num == res_card)
return true;
else
return false;
}
int check_card(String card_num){
for(int count=0; count<2; count++){
if(card_num == card_nums_holder[count]){
id_swiped = count+1;
return id_swiped;
}
}
return 0;
}
void set_att(){
attendance[id_swiped - 1] = 1;
}
int charToInt(char char1, char char2){ //this function converts two characters into a
numbers
num1 = char1 - 48;
num2 = char2 - 48;
return (num1*10+num2);
}
char* intToChar(int number){ //this function returns an array of characters in place of a
number
numb_ret[0] = number/10+48;
numb_ret[1] = number%10+48;
return numb_ret;
}
String changeString(String strToChange, char fir, char sec, char value){ //this function will
take first two characters and
a[0] = fir; //return a string with the next character
changed
a[1] = '\0';
b[0] = sec;
b[1] = '\0';
_s = String(a);
_s += b;
f_index = strToChange.indexOf(_s);
strToChange.setCharAt(f_index+2, value);
str_ret = strToChange;
return str_ret;
}
void init_sd(){
Serial.print("Initializing SD card...");

if (!SD.begin(53)) {
Serial.println("initialization failed!");
return;
}
```

```
}
Serial.println("initialization done.");
}
void get_str(){
myFile = SD.open("att.txt", FILE_READ);

if (myFile) {
Serial.println("att.txt is open");
// read from the file until there's nothing else in it:
while (myFile.available()) {
//Serial.write(myFile.read());
holder = myFile.read();
line += holder;
}
// close the file:
myFile.close();
} else {
// if the file didn't open, print an error:
Serial.println("error opening att.txt");
}
}
void get_day_and_totals(){
//get day
day_c1 = line.charAt(4);
day_c2 = line.charAt(5);
today = charToInt(day_c1, day_c2);
next_day = today+1;

//get totals
total[0] = charToInt(line.charAt(177), line.charAt(178));
total[1] = charToInt(line.charAt(322), line.charAt(323));
}
void set_next_day(){
n_day_c = intToChar(next_day);
line[4] = n_day_c[0];
line[5] = n_day_c[1];
}
void write_att(){
int ctr=0;
for(int w=0; w<line.length(); w++){
if((line[w] == day_c1) && (line[w+1] == day_c2) && (line[w-1] != 'L') && (line[w-2] != 'y'))
{
line[w+3] = char(attendance[ctr] + 48);

//fix the '#' bug at the start of second students attendance
if((today == 1) && (attendance[1] == 0) && (ctr == 2))
line[w+3] = '0';
else if((today == 1) && (attendance[1] == 1) && (ctr == 2))
line[w+3] = '1';

++ctr;
}
}
}
```



```
}  
void set_new_total(){  
int new_tot1, new_tot2;  
String new_t1;  
String new_t2;  
if(attendance[0] == 0){  
new_tot1 = total[0];  
new_t1 = intToChar(new_tot1);  
line[177] = new_t1[0];  
line[178] = new_t1[1];  
}  
else if(attendance[0] == 1){  
new_tot1 = total[0] + 1;  
new_t1 = intToChar(new_tot1);  
line[177] = new_t1[0];  
line[178] = new_t1[1];  
}  
if(attendance[1] == 0){  
new_tot2 = total[1];  
new_t2 = intToChar(new_tot2);  
line[322] = new_t2[0];  
line[323] = new_t2[1];  
}  
else if(attendance[1] == 1){  
new_tot2 = total[1] + 1;  
new_t2 = intToChar(new_tot2);  
line[322] = new_t2[0];  
line[323] = new_t2[1];  
}  
}  
void del_n_rec_file(){  
// delete the file:  
Serial.println("Removing att.txt...");  
SD.remove("att.txt");  
myFile = SD.open("att.txt", FILE_WRITE);  
if (myFile) {  
Serial.print("Writing to att.txt...");  
myFile.print(line);  
// close the file:  
myFile.close();  
Serial.println("done.");  
} else {  
// if the file didn't open, print an error:  
Serial.println("error opening att.txt");  
}  
// close the file:  
myFile.close();  
}  
void reset_system(){ //if reset card is swiped  
lcd.clear();  
lcd.print("resetting");  
lcd.setCursor(4,1);  
lcd.print("System...");  
delay(4000);  
// delete the file:
```

```
Serial.println("Removing att.txt...");
SD.remove("att.txt");
myFile = SD.open("att.txt", FILE_WRITE);
if (myFile) {
  Serial.print("Writing to att.txt...");
  myFile.print("day(01)\nID    NUMBER\tCLASS(ATTENDANCE)\n1.          \
t01(0),02(0),03(0),04(0),05(0),06(0),07(0),08(0),09(0),10(0),11(0),12(0),13(0),14(0),15(0),1
6(0),17(0),18(0),19(0),    a20(0)          TOTAL01(00)\n2.          \
t01(0),02(0),03(0),04(0),05(0),06(0),07(0),08(0),09(0),10(0),11(0),12(0),13(0),14(0),15(0),1
6(0),17(0),18(0),19(0),20(0)  TOTAL02(00)*");
  // close the file:
  myFile.close();
  Serial.println("done.");
} else {
  // if the file didn't open, print an error:
  Serial.println("error opening att.txt");
}
// close the file:
myFile.close();
lcd.clear();
lcd.print("RESET COMPLETE.");
lcd.setCursor(1,1);
lcd.print("PLEASE RESTART!");
delay(10000);
}
```

IJSER