**ATTENDANCE REGISTER USING RFID MODULE AND EXCEL SHEET**

*Mini /Major project report submitted in partial fulfillment of the requirements*

*for the degree of*

**Bachelor of Technology**

in

**Electronics and Communication Engineering**

*Submitted by*

**A Y S HARSHA**

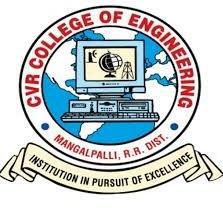
**(16B81A04P9)**

**SRI HARSHA SAGAR**

**(16B81AO4L0)**

**SOWMYA NAIDU**

**(16B81A04K3)**

****

**Department of Electronics & Communication Engineering**

**CVR COLLEGE OF ENGINEERING**

**(An Autonomous Institution & Affiliated to JNTUH)**

**Ibrahimpatnam (M), Ranga Reddy (D), Telangana**

**2016-2020**

**ATTENDANCE REGISTER USING RFID MODULE AND EXCEL SHEET**

*Mini /Major project report submitted in partial fulfillment of the requirements*

*for the degree of*

**Bachelor of Technology**

in

**Electronics and Communication Engineering**

*Submitted by*

**A Y S HARSHA**

**(16B81A04P9)**

**SRI HARSHA SAGAR**

**(16B81AO4L0)**

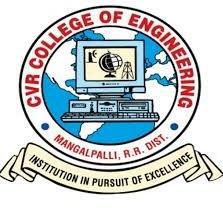
**SOWMYA NAIDU**

**(16B81A04K3)**

Under the Supervision of

**Dr/Mr/Ms. Name of the faculty member**

Designation



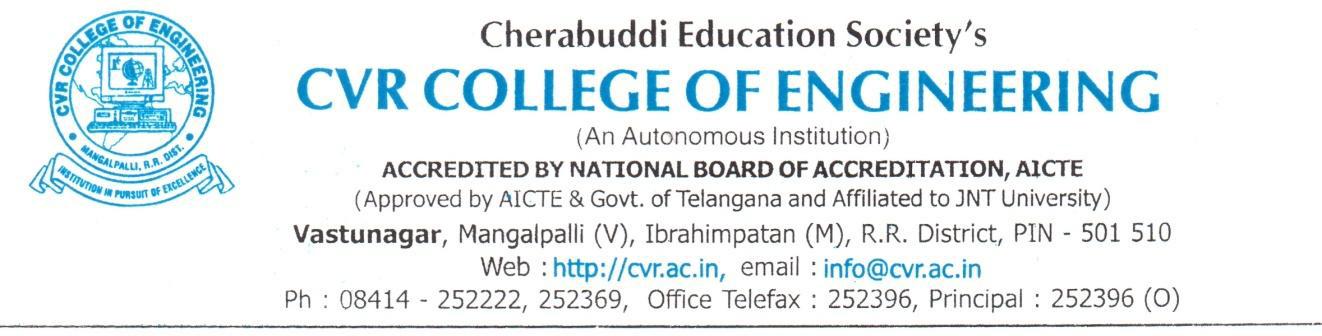
**Department of Electronics & Communication Engineering**

**CVR COLLEGE OF ENGINEERING**

**(An Autonomous Institution & Affiliated to JNTUH)**

**Ibrahimpatnam (M), Ranga Reddy (D), Telangana**

**2016-2020**

****

**CERTIFICATE**

This is to certify that the project titled “**TITLE OF THE PROJECT”** submitted to the **CVR College of Engineering,** affiliated to **JNTU, Hyderabad** by **A Y S HARSHA(16B81A04P9), SRI HARSHA SAGAR(16B81AO4L0), SOWMYA NAIDU(16B81A04K3)**is a bonafide record of the work done by the students towards partialfulfillment of requirements for the award of the degree of **Bachelor of Technology in** **Electronics & Communication Engineering**.

|  |  |
| --- | --- |
| **Supervisor** | **Head of the Department** |
|  |  |
| Dr./Mr./Ms. Name | Dr. K. Lalithendra |
| Designation | Professor & HOD |
| Department of Electronics & | Department of Electronics & |
| Communication Engineering | Communication Engineering |

Place: Hyderabad

Date:



**ACKNOWLEDGEMENT**

Completion of this project and thesis would not have been possible without the help of many people, to whom we are very thankful. First of all, we would like to convey our sincere thanks to **Dr. K. S. Nayanathara**, **Principal**, CVR College of Engineering for her constant support and encouragement.

We would like to express my sincere gratitude to our supervisor, **Name of the faculty member , Designation.** His/ her constant motivation, guidance and support helped us a great deal to achieve this feat.

We would like to thank **Dr. K. Lalithendra, Professor of ECE & HOD**, for guiding and inspiring us in many ways. We are also thankful to other faculty members and staff of Electronics and Communication department for their support.

We also thank our coordinator **Name of the faculty member , Designation** for his/her constant support in guiding us about the academic related work.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank our seniors whose guidance helped us in this regard.

**ABSTRACT**

**Abstract:**

Now a days everything has been modified and updated to modern technological methods by using Embedded systems. But yet we have Old school type of attendance register. There are big books for marking attendance. And also it takes at least 5 to 10 min of every class hour which the lecturer loses.

So, in order to update the attendance, register we propose embedded system using RFID module.

# Prototype Components:

* RFID RC522 module
* Arduino
* Buzzer
* Green and Red LED
* UID Cards
* Excel Sheet

# Technology:

* Arduino IDE
* PLX-DAQ software

# Advantages:

The main advantage is there would be no proxy attendance given. It updates the lengthy and hard work of marking attendance in books and also reduces a lot of burden to the lecturer. Data is stored in spread sheet which makes it even more easy to read and stores it safely.

# Conclusion:

Updating the old school type of attendance marking takes us a step ahead to the modernization and also using of electronic embedded system makes a man’s work a lot easier.

**GUIDE:**

Bhanu Prakash Associate Professor of ECE

**TABLE OF CONTENTS**

**Certificate iii**

**Acknowledgement iv**

**Abstract v**

**Table of Contents vi**

**1 INTRODUCTION 9**

**1.1 Objective 9**

**1.2 Conclusion 9**

**2 EMBEDDED SYSTEM 10**

**2.1 Introduction 10**

**2.2 Need for Embedded System 10**

**2.3 Types of Embedded System 11**

**2.3.1 Small scale Embedded Systems 11**

**2.3.2 Medium scale Embedded Systems 11**

**2.3.3 Sophisticated Embedded Systems 11**

**2.4 Conclusion 12**

**3 BLOCK DIAGRAM AND DESCRIPTION 12**

**3.1 Block Diagram 12**

**3.2 Description 12**

**3.3 Conclusion 12**

**4 HARDWARE DESCRIPTION 13**

**4.1 Hardware Components 13**

**4.2 Arduino UNO 13**

**4.2.1 Pin Description 14**

**4.2.2 Technical Specifications 15**

**4.2.3 How to use Arduino Board 15**

**4.2.4 Communication 16**

**4.2.5 Arduino UNO to ATmega328 Pin Mapping 16**

**4.3 RFID-RC522 17**

**4.3.1 What is RFID? 17**

**4.3.2 Why Use RFID? 18**

**4.3.3 RFID Solutions 18**

**4.3.4 RFID Tags 18**

**4.3.5 RFID Readers 19**

**4.3.6 RFID Frequencies 21**

**4.3.7 RFID Systems 22**

**4.3.8 RFID Applications 23**

**4.4 UID 24**

**4.5 Conclusion 26**

**5 SOFTWARE DESCRIPTION 26**

**5.1 Arduino IDE 26**

**5.1.1 Arduino IDE: Initial Setup 26**

**5.2.2 IDE: Board Setup 27**

**5.2.3 IDE: COM Port Setup 28**

**5.1.4 Testing your Settings: Uploading Blink 28**

**5.1.5 Guide Summary 30**

**5.1.6 Trouble Shooting Uploading Errors 30**

**5.3 PLX-DAQ 30**

**6 CONCLUSION AND FUTURE SCOPE 32**

**6.1 Conclusion 32**

**6.2 Future Scope 32**

**CODE 33**

**REFERENCES 37**

**ATTENDANCE REGISTER USING RFID MODULE AND EXCEL SHEET**

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION**

The use of Radio Frequency Identification (RFID) technology extends historically to the 1970’s, when the first commercial products emerged. The introduction of this technology motivated future research that sought to integrate RFID into various applications ranging from inventory control/monitoring to human/animal tracking. However, the practical significance of this technology was not fully realized, until mass manufacturing of RFID tags and readers, at different scales and for different ranges, was made possible. Currently, RFID-based systems are being proposed as alternatives to previously established technologies and as extensions to others.

The concept of RFID systems has been available for a long time, yet only recently has its potential been manifested in applications of ever-growing demand. The recent popularity of RFID technology is due to the following fundamental properties that characterize the production, deployment, and functionality of RFID systems. (1) The recent ability to acquire low cost and power efficient RFID equipment has supported a novel trend in RFID system research that seeks to replace current technologies. (2) Due to its data storage capacity and the addition of diverse sensors to RFID tags, this technology has the potential of bridging “the growing gap between the digital networked world and the physical world”. This provides a low cost means for ubiquitous sensing of environmental parameters, which is not restricted to measurement alone, but it can also render intelligent monitoring of these parameters and their temporal fluctuations.

In this report, we describe details of an ongoing project that aims at integrating RFID technology into the attendance register. We automatically send the data to database without any intervention of human being.

The main advantage is there would be no proxy attendance given. It updates the lengthy and hard work of marking attendance in books and also reduces a lot of burden to the lecturer. Data is stored in spread sheet which makes it even more easy to read and stores it safely.

* 1. **CONCLUSION**

Updating the old school type of attendance marking takes us a step ahead to the

modernization and also using of electronic embedded system makes a man’s work a lot easier.

**CHAPTER 2**

**EMBEDDED SYSTEM**

**2.1 INTRODUCTION**

Embedded systems are computer systems inside a larger system, with a specific function that it is designed for. It is composed of both hardware and software. They integrate different mechanical, chemical and electrical parts.

These embedded systems are found in most of the electronic systems we use. They are essential to our everyday lives Some examples are: Personal Digital Assistants (PDAs), mobile phones, microwave ovens, washing machines, automobiles, and vital signs monitoring. We can see that embedded systems are used for a wide range of areas.

Firmware is used to refer to the program instructions used for embedded systems. The hardware is very limited, a small memory, and sometimes a screen or keyboard.

Embedded systems are considered to be very reliable, they are supposed work for years without any errors. In the time an error occurs the embedded system should be able to fix the error by itself. A problem with this kind of system is that it cannot be shut down if it repair is needed, lots of money will be lost in that event. An example of such an event is if an error occurs to factory controls.

The processor is the center of the embedded system, it has two main units: a control unit and an execution unit.

Embedded systems are reactive systems, this means they take action depending on the environment around them, they do this using sensors that take in data from the environment, and produce an output using an actuator.

These kinds of systems can be classified into three different types:

* + - small scale embedded systems
    - medium scale embedded systems
    - sophisticated embedded systems

**2.2 NEED FOR EMBEDDED SYSTEM**

The first reason why we need embedded systems is because general-purpose computers, like PCs, would be far too costly for the majority of products that incorporate some form of embedded system technology. Another reason why we need embedded systems is because general-purpose solution might also fail to meet a number of functional or performance requirements such as constraints in power-consumption, size-limitations, reliability or real-time performance etc.

The digital revolution, started decades ago, has reached a stage that we cannot conduct our normal modern daily lives without this technology. Indeed, it is safe to say that we already own at least one piece of equipment, which contains a processor, whether it is a phone, a television, an automatic washing machine or an MP3 player.

The colossal growth of processing power in small packages has fuelled the digital revolution. All sectors of the economy have been influenced by the digital revolution and the industry has experienced tremendous developments in all aspects of engineering disciplines.

**2.3 TYPES OF EMBEDDED SYSTEMS**

The embedded systems classified based on the performance and architecture.

* Small Scale Embedded System
* Medium Scale Embedded Systems
* Sophisticated or Complex Embedded Systems

**2.3.1 Small Scale Embedded System**

* Small-scale embedded systems are an entry-level system in which 8-bit or 16-bit processor is used. The processor has very limited resources like RAM, ROM and processing speed.
* For the development of the small-scale embedded system, we need integrated development environment(IDE) for writing the code. The code is generally written in assembly language or in embedded C language. The compiler is needed if programming in embedded C. Compiler compiles the C code into the Hex code. We need assembler if we are coding in assembly language.

And at the time of writing the code into the processor. We need a hardware device called the programmer. That takes the hex code and write it to the ROM of the processor.

### **Examples of Small Scale Embedded Systems**

* washing machine
* Oven
* Automatic Door Lock
* Motion Controlled Home Security System
* Keyboard controller
* CD Drive
* fax machine
* photocopy machine
* printer

chocolate vending machine, washing machine, cooking system, multitasking toys, keyboard controller, MMI and network access cards, CD drive or hard disk drive controller, Fax machine, photocopy or printer machine, Remote controller of TV, Telephone with memory, display and other sophisticated features,

The small-scale embedded systems are small and can be operated with a small battery.

## **2.3.2 Medium Scale Embedded Systems**

This type of embedded systems consisting of 16-bit or 32-bit microprocessor. There are other options also available like Digital signal processor(DSP), or some Advanced RISC architecture machine. The medium scale embedded systems are generally faster than the small scale due to the number of bits of processor that give a better speed.

At the time of development, we need different types of software tools like assembler, compiler and debugger. An RTOS can be implemented on medium scale embedded systems.

## **2.3.3 Sophisticated or Complex Embedded Systems**

This type of embedded systems is made to perform the complex functions. The systems have hardware and software complexity. In this type of systems, the speed is measure concern. So some of the software functionality is implemented into the hardware. There are different modules are implemented on the CPU like TCP/IP, encryption, decryption, multiplier unit and more. Once the functionality is implemented into hardware the gives the better speed.

This type of systems have some key components like graphical user interface, communication ports like I2C, CAN, RS232, Ethernet or USB. You can see this type of system as network router, IP camera, embedded web server.

**2.4 CONCLUSION**

This the basic definition and the concept of embedded system.

**CHAPTER 3**

**BLOCK DIAGRAM AND DESCRIPTION**

**3.1 BLOCK DIAGRAM**

UID CARDs

Arduino UNO

RFID CARD READER

Computer

With PLX-DAQ

**3.2 DESCRIPTION**

A computer installed with Microsoft windows excel sheet and PLX-DAQ software is connected to Arduino UNO which in turn connected to RFID card reader.

This RFID card reader produces a radio frequency of certain range. When a UID (Unique Identifier) card comes into the radio frequency range of RFID card reader, UID card gets detected. Arduino UNO sends the detected card details to the spreadsheet using the PLX-DAQ.

**3.3 CONCLUSION**

The detailed working of the project has been explained with using the block diagram.

**CHAPTER 4**

**HARDWARE DESCRIPTION**

**4.1 HARDWARE COMPONENTS**

The hardware of the project mainly has 3 components.

* Arduino UNO
* RFID RC522
* UID Cards

It also has basic components like

* Bread board
* LED blubs
* Buzzer
* Connecting Wires

A laptop with windows excel is also important.

**4.2 ARDUINO UNO**

**4.2.1 PIN DESCRIPTION**

|  |  |  |
| --- | --- | --- |
| **Pin Category** | **Pin Name** | **Details** |
| Power | Vin, 3.3V, 5V, GND | Vin: Input voltage to Arduino when using an external power source.  5V: Regulated power supply used to power microcontroller and other components on the board.  3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA.  GND: ground pins. |
| Reset | Reset | Resets the microcontroller. |
| Analog Pins | A0 – A5 | Used to provide analog input in the range of 0-5V |
| Input/Output Pins | Digital Pins 0 - 13 | Can be used as input or output pins. |
| Serial | 0(Rx), 1(Tx) | Used to receive and transmit TTL serial data. |
| External Interrupts | 2, 3 | To trigger an interrupt. |
| PWM | 3, 5, 6, 9, 11 | Provides 8-bit PWM output. |
| SPI | 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) | Used for SPI communication. |
| Inbuilt LED | 13 | To turn on the inbuilt LED. |
| TWI | A4 (SDA), A5 (SCA) | Used for TWI communication. |
| AREF | AREF | To provide reference voltage for input voltage. |

**4.2.2 TECHNICAL SPECIFICATIONS**

|  |  |
| --- | --- |
| Microcontroller | ATmega328P – 8 bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended Input Voltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**4.2.3 HOW TO USE ARDUINO BOARD**

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default.  Out of these 14 pins, some pins have specific functions as listed below:

* **Serial Pins 0 (Rx) and 1 (TX):** RX and TX pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
* **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analogWrite() function.
* **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
* **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference() function.

* Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

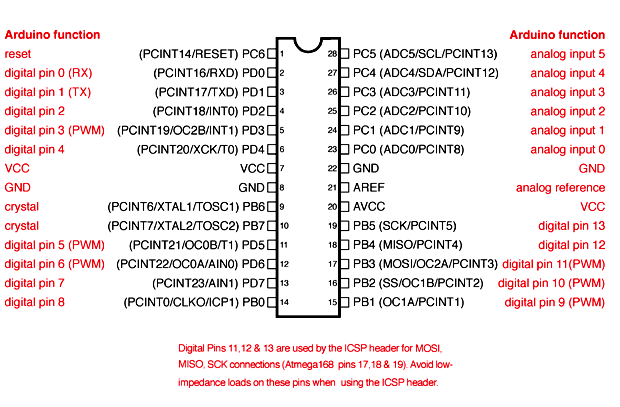
Arduino Uno has a couple of other pins as explained below:

* **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.
* **Reset Pin:**Making this pin LOW, resets the microcontroller.

**4.2.4 COMMUNICATION**

Arduino can be used to communicate with a computer, another Arduino board or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two RX and TX LEDs on the Arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

**4.2.5 ARDUINO UNO TO ATMEGA328 PIN MAPPING**



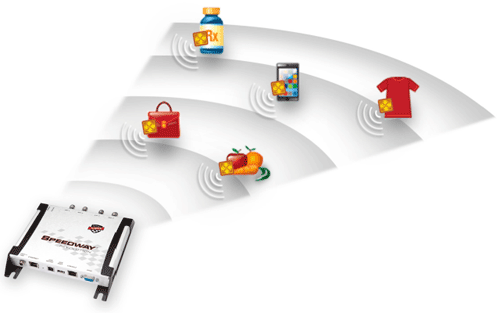
**4.3 RFID – RC522**

****

**4.3.1 WHAT IS RFID?**

**RFID is short for radio frequency identification. In a nutshell, it is a form of a wireless communication that uses radio waves in order to track and identify objects.** Similar to the concept of a barcode scan, RFID digitalizes this into a process where people can:

* locate an individual or an item beyond its product type
* locate products that cannot be directly seen
* locate up to thousands of items simultaneously
* locate and identify items which are at only a few centimetres or several meters away from each other



Every radio frequency identification (RFID) system has readers and tags that communicate with each other through radio. Having in mind that the RFID tags are small and require little power, they don’t need any battery to store information and exchange the data with readers. This eases up the entire process – and makes applying tags and all kinds of things possible in an easy way.

**4.3.2 WHY USE RFID?**

Simply put, the RFID technology is advanced and is able to enhance and protect the life of every consumer, but also set the bar higher when it comes to doing business. They are a progressive, flexible and auto-identifying technology that can be used to track and monitor every item with the utmost accuracy.

When put to practice, the RFID technology can tell you where a particular object is, what is its condition and why it is important. As a technology advancing under the prism of the Internet of Things (IoT), RFID has the exact same focus – connecting every physical object and item in a digital way, so that they become a part of a large information system that informs, shares and responds.

Over the past few years, there is an increase in demand for RFID. This is certainly due to the capability of ‘tagging’ items seamlessly and processing information in a quick way. However, it is also due to the cost-effectiveness of the communications chip and the equipment in general. Aside from that, RFID can be used in a range of industries, from supply chain management to asset tracking, authentication of counterfeited pharmaceuticals etc.

**4.3.3 RFID Solutions**

A basic RFID system has tags attached to all items that need to be tracked. Made from a tiny tag-chip, these tags are also known as integrated circuit (IC) and are connected to an antenna that can be built into various types of tags such as apparel hang tags, labels and security tags – but also industrial asset tags. In a nutshell, every tag chip contains memory that is stored in it and contains the basic product code (EPC) of the product and other necessary information allowing it to be tracked and identified by the UHF RFID scanner or reader everywhere.

On the other hand, the UHF RFID reader is a network-connected device which can either be fixed or mobile – and features an antenna that sends the power and receives signals as commands to the tags. In other words, it is an access point for RFID tagged items so that the tags data is made available to every business application.

**4.3.4 RFID TAGS**

As we said above, an RFID tag is comprised of an integrated circuit called an IC or simply, a chip. This chip is attached to an antenna that is printer, etched, vapor-deposited or stamped onto a mount that is a polyethylene therephtate (PET) or a paper substrate. This combination of a chip and an antenna is called an inlay and is converted and put in between a printed label and its adhesive backing – or inserted in a durable structure.

**Tag Chip**

Also known as the integrated circuit, the tag chip is known for its performance, memory and extended features. It is a device that is pre-programmed with a tag identifier and a unique serial number that is assigned by the chip manufacturer. It also features a memory bank in order to store every item’s unique tracking identifier – known as a product code or EPC.

**Electronic Product Code (EPC)**

The electronic product code (or EPC) is stored inside the chip’s memory – printed by an RFID printer and taking 96-bit strings of data. Its first eight bits consist of a header that identifies the protocol’s version and the next 28 bits identify the organization that manages that tag’s data, assigned by the EPC global consortium. The following 24 bits present the object glass that identifies the type of product and the last 36 bits are reserved for the unique serial number for each particular tag. Usually, the last two fields are set by the organization that issued that tag.

**Tag Antennas**

The tag antenna is a collector of energy that channels it to the chip in order to turn it on. The larger this antenna’s area is, the more energy it is capable of collecting and channeling – hence, the greater read range that the tag will have.

Basically, there isn’t a perfect fit of an antenna that works for all applications. The most important aspect is the application that defines its specifications. And while some tags are optimized for a particular frequency band, others can be tuned for stellar performance when attached to materials that do not work well wirelessly (liquids, metals etc.). An antenna can be printed, etched or stamped with conductive ink, or even vapor deposited onto labels.

Every tag having a single antenna is not as reliable as a tag with multiple antenna. Simply put, a single antenna can result in a ‘dead zone’ – an area where the signal is not strong or easily harvested to power on the chip. That is why multiple antennas are recommended – in order to eliminate any dead zones and increase the readability of the chip.

**4.3.5 RFID READERS**

An UHF RFID reader is also known as an interrogator. It is simply a device that provides connection between the tag data and the enterprise system software needing the information. The UHF RFID scanner (or reader) is capable of communicating with tags that are within the field of operation and can seamlessly take care of the inventory, filter, write or encode selected tags.

The reader uses an antenna that is attached and able to capture data from tags. Afterwards, the data is passed to a computer so that it becomes processed. Similar to the RFID tags, there are multiple makes and models of UHF RFID readers in various sizes and types. They can either be fixed in a store or factory or integrated into a mobile device known as a handheld PDA with UHF RFID – or a handheld computer with UHF RFID.

**Reader Antennas**

As you may know by now, an UHF RFID scanner or reader is comprised of an UHF RFID reader and reader antennas that work together in order to read tags. The process is quite straightforward – the reader antennas convert the electrical current into electromagnetic waves which are radiated into space and from where they can be received by a tag antenna and afterwards converted into electrical current. Similar to the tag antennas, there is a vast array of reader antennas and selecting the best one varies mostly because of the solution’s specific application and environment.

The two most common types of reader antennas are linear and circular polarized antennas. In a nutshell, antennas that radiate linear electric fields have long ranges and high levels of power that enable their signals to penetrate through different materials in order to read tags. The linear antennas are however sensitive to tag orientation and depend on the tag angle or placement. Therefore, they can have a difficult time reading tags. Alternatively, the antennas that radiate circular fields are less sensitive to the orientation, but not able to deliver the same amount of power as linear antennas.

The choice of antenna is also determined by the distance between the tag and the UHF RFID reader. Also known as read range, this distance is operated by reader antennas in either a ‘near-field’ (short range) or ‘far field’ (long range). The near-field situation is less than 30cm where the antenna uses magnetic coupling – and the readability of the tags is not affected by the presence of dielectrics. On the other hand, far-field applications use a range that is greater than 30cm and can extend to tens of meters. This process is known as electromagnetic coupling and dielectrics can weaken its communication.

**Reader Control And Application Software**

The application software and reader control are both aspects of the middleware. The middleware is the platform onto which the UHF RFID readers and applications operate. It is also the backbone that sends the control commands to the reader and receives the tag data from it.

**Creating An RFID Solution**

Deploying an UHF RFID scanner or reader needs a lot of components and multiple actors. Technically, it is a system that requires basic hardware including the previously mentioned tag chips, antennas, readers and reader antennas – as well as the reader control and application software to complete the entire solution.

As soon as these components ‘click’ together, there are endless ways an UHF UHF RFID reader can be used – with a lot of creative applications possible. The RFID system applications are bound to improve the quality of many business operations, inventory and customer experience – and disrupt the traditional way of organizing and handling items.

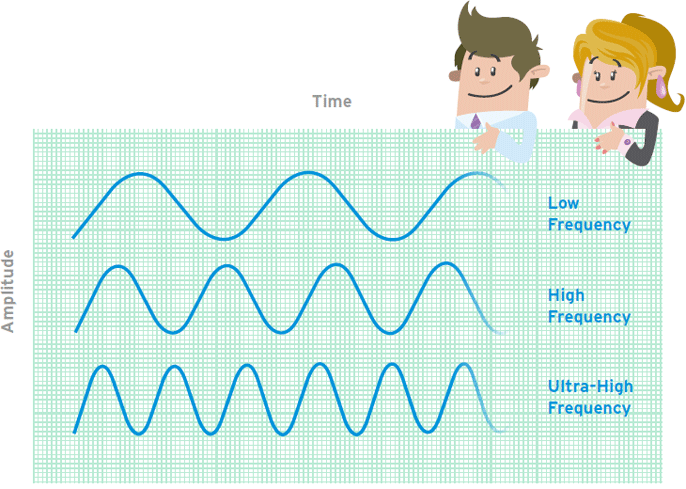
**DIFFERENT TYPES OF RFID SYSTEMS**

Every RFID system can be broken down by the frequency band in which it operates. Whether it is low, high or ultra-high frequency – there are many other categories of RFID systems as well. However, there are two major categories of RFID systems – active and passive, and we are explaining each of their types and frequencies below.

**4.3.6 RFID FREQUENCIES**

The frequency refers to the size of the radio that is used to communicate between each of the RFID system components. Hence the RFID systems are able to operate in low frequency (LF), high frequency (HF) and ultra-high frequency (UHF) bands. The radio waves are able to behave differently in each of these frequencies – and there are pros and cons of each frequency band.

On the off chance that a RFID system works at a lower recurrence, it has a shorter perused range and slower information read rate, yet expanded abilities for perusing close or on metal or fluid surfaces. On the off chance that a system works at a higher recurrence, it for the most part has speedier information exchange rates and more read extents than lower recurrence systems, however more affectability to radio wave obstruction brought on by fluids and metals in the earth.



**LF RFID**

The LF band covers frequencies from 30 KHz to 300 KHz. Regularly LF RFID systems work at 125 KHz, in spite of the fact that there are some that work at 134 KHz. This recurrence band gives a short read scope of 10 cm, and has slower perused speed than the higher frequencies, yet is not exceptionally delicate to radio wave impedance.

LF RFID applications incorporate access control and domesticated animals following.

Gauges for LF creature following systems are characterized in ISO 14223, and ISO/IEC 18000-2. The LF range is not viewed as a really worldwide application on account of slight contrasts in recurrence and power levels all through the world.

**HF RFID**

The HF band ranges from 3 to 30 MHz Most HF RFID systems work at 13.56 MHz with read ranges between 10 cm and 1 m. HF systems experience moderate affectability to impedance.

HF RFID is ordinarily utilized for ticketing, instalment, and information exchange applications.

There are a few HF RFID principles set up, for example, the ISO 15693 standard for following things, and the ECMA-340 and ISO/IEC 18092 gauges for Near Field Communication (NFC), a short-range innovation that is regularly utilized for information trade between devices. Other HF principles incorporate the ISO/IEC 14443 An and ISO/IEC 14443 benchmarks for MIFARE innovation, which utilized as a part of keen cards and closeness cards, and the JIS X 6319-4 for FeliCa, which is a savvy card system usually utilized as a part of electronic cash cards.

**UHF RFID**

The UHF recurrence band covers the reach from 300 MHz to 3 GHz. Systems consenting to the UHF Gen2 standard for RFID utilize the 860 to 960 MHz band. While there is some fluctuation in recurrence from area to locale, UHF Gen2 RFID systems in many nations work somewhere around 900 and 915 MHz

The read scope of detached UHF systems can be the length of 12 m, and UHF RFID has a speedier information exchange rate than LF or HF. UHF RFID is the most touchy to impedance, however numerous UHF item producers have discovered methods for planning labels, antennas, and readers to keep execution high even in troublesome situations. Inactive UHF labels are simpler and less expensive to make than LF and HF labels.

UHF RFID is utilized as a part of a wide assortment of applications, extending from retail stock administration, to pharmaceutical hostile to forging, to wireless device design. The heft of new RFID tasks are utilizing UHF contradicted to LF or HF, making UHF the quickest developing section of the RFID market.

The UHF recurrence band is managed by a solitary worldwide standard called the ECPglobal Gen2 (ISO 18000-6C) UHF standard.

**4.3.7 RFID SYSTEMS**

**Active RFID Systems**

* In active RFID systems, every tag has its own particular transmitter and power source. In most of the cases, the power source is a battery. Active tags show their own sign to transmit the data put away on their microchips.
* Active RFID systems regularly work in the ultra-high recurrence (UHF) band and offer a scope of up to 100 m. By and large, active tags are utilized on expansive items, for example, rail autos, enormous reusable holders, and different resources that should be followed over long separations.
* There are two fundamental sorts of active tags: transponders and reference points. Transponders are “woken up” when they get a radio sign from a reader, and after that power on and react by transmitting a sign back. Since transponders don’t effectively emanate radio waves until they get a reader signal, they moderate battery life.

Reference points are utilized as a part of most ongoing finding systems (RTLS), so as to track the exact area of an advantage persistently. Not at all like transponders, reference points are not fueled on by the reader’s sign. Rather, they discharge signals at pre-set interims. Contingent upon the level of finding exactness required, reference points can be set to transmit flags at regular intervals, or once per day. Every reference point’s sign is gotten by reader antennas that are situated around the border of the region being checked, and imparts the tag’s ID data and position.

**Passive RFID Systems**

In passive RFID systems, the reader and reader antenna send a radio sign to the tag. The RFID tag then uses the transmitted sign to control on, and reflect vitality back to the reader. Aloof RFID systems can work in the low recurrence (LF), high recurrence (HF) or ultra-high recurrence (UHF) radio groups. As passive system extents are restricted by the power of the tag’s backscatter (the radio sign reflected from the tag back to the reader), they are normally under 10 m. Since latent tags don’t require a power source or transmitter, and just require a tag chip and antenna, they are less expensive, littler, and simpler to produce than active tags.

Aloof tags can be bundled in a wide range of routes, contingent upon the particular RFID application necessities. Case in point, they might be mounted on a substrate, or sandwiched between a cement layer and a paper mark to make keen RFID names. Latent tags may likewise be implanted in an assortment of devices or bundles to make the tag impervious to great temperatures or cruel chemicals.

Aloof RFID arrangements are valuable for some applications, and are usually sent to track merchandise in the store network, to stock resources in the retail business, to validate items, for example, pharmaceuticals, and to insert RFID capacity in an assortment of devices. Detached RFID can even be utilized as a part of stockrooms and dispersion focuses, notwithstanding its shorter extent, by setting up readers at stifle focuses to screen resource development.

**Battery-Assisted Passive (BAP) Systems**

A Battery-Assisted Passive RFID tag is a kind of inactive tag that joins a urgent active tag highlight. While most latent RFID tags utilize the vitality from the UHF RFID reader’s sign to control on the tag’s chip and backscatter to the reader, BAP tags utilize a coordinated power source (more often than not a battery) to control on the chip, so the greater part of the caught vitality from the reader can be utilized for backscatter. Not at all like transponders, BAP tags do not consist of their own transmitters.

**4.3.8 RFID APPLICATIONS**

The benefits of radio frequency identification are endless. Basically, RFID can help with:

* automating the inventory and asset-tracking in a lot of industries, from healthcare to manufacturing and retail
* identifying every product, its source and origin
* cutting the usage of counterfeit products in the supply chain
* setting the shopping experience on another level when it comes to consumers
* making the process of returns and fewer out-of-stock items easier
* providing utmost visibility in the supply chain and reducing the business costs through a new efficient distribution model
* preventing any lost to theft or inaccurate accounting of goods
* improving the security of every item and personnel through advanced cargo monitoring
* locking, unlocking and configuring devices wirelessly
* enabling top-shelf access control in certain areas and devices

Basically, RFID is a new form of technology ready to ramp up the efficiency of each operation, improve the traceability and visibility of every product, decrease the reliance and automate the manual processes into a cost-effective operation providing useful data for business analytics.

**4.4 UID**

A unique identifier (UID) is a numeric or alphanumeric string that is associated with a single entity within a given system. UIDs make it possible to address that entity, so that it can be accessed and interacted with.

Unique identifiers can be assigned to anything that needs to be distinguished from other entities, such as individual users, companies, machines or websites. These distinctive values are usually assigned depending on the needs of the specific application, but can either be randomly auto-generated with an algorithm, allocated incrementally or chosen by the user.

### **Uses of UIDs**

The most widely known use of unique identifiers occurs when users register for a website or service. Customers are often provided with a username or user ID that allows the company they are registering with to differentiate them within their user logs. These identifiers are then also used for security and log on purposes.

In a database or spreadsheet, unique identifiers may be designated as a specific column or field to help make sorting and filtering through information easier. This also helps trace information back to a specific user or entity within the system.

Another popular application of UIDs is in a physical supply chain. Manufacturers often mark individual pieces of a larger component, such as computer parts, or an entire product with a serial number. This allows users to trace back the origin of the product in case of a malfunction, defect or recall.

Unique identifiers are also commonly used within the healthcare industry. Instead of reporting medical information with a patient’s name, a personal code is created. This code combats issues of patient privacy as well as eliminates the possibility of duplicate reports.

### **Examples of UIDs**

While UIDs can take a variety of forms depending on the application, a few examples include:

* A Uniform Resource Identifier (URI) is a unique identifier that makes content addressable on the Internet by uniquely targeting items, such as text, video, images and applications.
* A Uniform Resource Locator (URL) is a particular type of URI that targets Web pages so that when a browser requests them, they can be found and served to users.
* A Universal Unique Identifier (UUID) is a 128-bit number used to uniquely identify some object or entity on the Internet.
* A global unique identifier (GUID) is a number that Microsoft programming generates to create a unique identity for an entity such as a Word document.
* A bank identifier code (BIC) is a unique identifier for a specific financial institution.
* A unique device identifier (UDID) is a 40-character string assigned to certain Apple devices including the iPhone, iPad and iPod Touch.
* A service set identifier (SSID) is a sequence of characters that uniquely names a wireless local area network (WLAN).
* A national provider identifier (NPI) is a unique ten-digit identification number required by HIPAA for all health care providers in the United States.
* A MAC address is a computer’s unique hardware number in a local area network (LAN).

Other, more basic ways to uniquely identify an entity or user include by social security number, email address, username and phone number.

### **UID vs IUID vs UII**

Within the Department of Defense (DoD), there are three related acronyms associated with the standard of government property. The first is the Item Unique Identification (IUID) Registry. This registry stores information for the DoD about how to track, maintain and deploy government assets. Items are added to the IUID Registry if their acquisition cost is over five thousand dollars, they are mission critical or they are high risk.

**4.5 CONCLUSION**

The hardware description has been given. Components used are described and its uses are mentioned.

**CHAPTER 5**

**SOFTWARE DESCRIPTION**

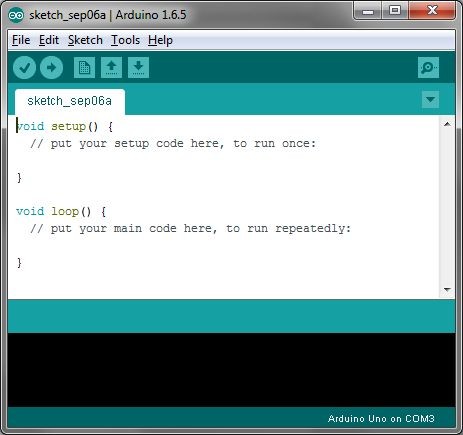
**5.1 ARDUINO IDE**

Arduinofirst and foremost is an open-source computer hardware and software company. The Arduino Communityrefers to the project and user community that designs and utilizes microcontroller-based development boards. These development boards are known as Arduino Modules***,*** which are open-source prototyping platforms. The simplified microcontroller board comes in a variety of development board packages.

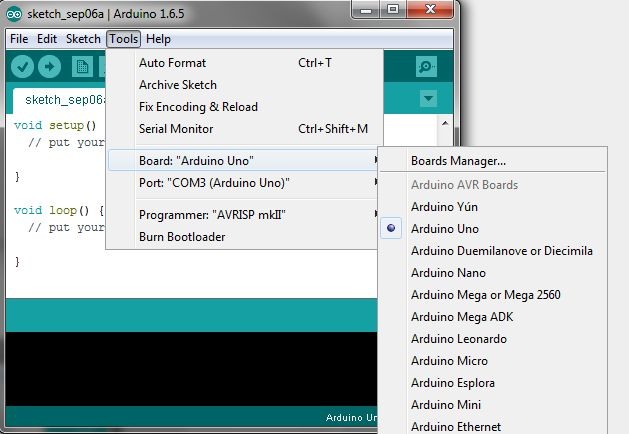
The most common programming approach is to use the Arduino IDE, which utilizes the C programming language. This gives you access to an enormous Arduino Library that is constantly growing thanks to open-source community.

**5.1.1 ARDUINO IDE: INITIAL SETUP**

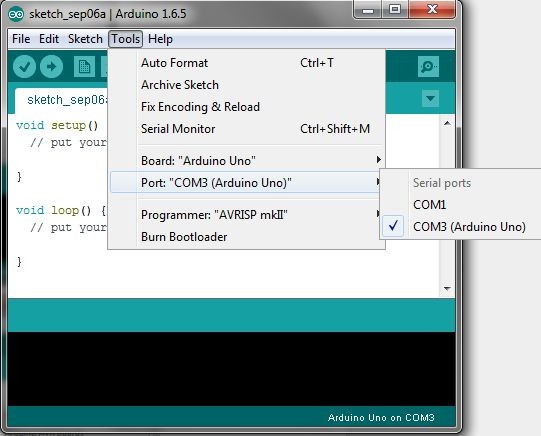
This is the Arduino IDE once it’s been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.



**5.1.2 IDE: BOARD SETUP**

You have to tell the Arduino IDE what board you are uploading to. Select the Toolspulldown menu and go to Board**.** This list is populated by default with the currently available Arduino Boards that are developed by Arduino. If you are using an Uno or an Uno-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If you are using another board/clone, select that board.

**5.1.3 IDE: COM PORT SETUP**

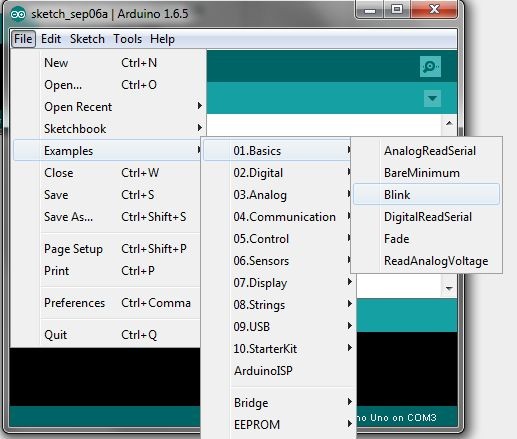
If you downloaded the Arduino IDE before plugging in your Arduino board, when you plugged in the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Toolspulldown menu and then Port**.** Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give it’s name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, you should see the board type and COM number of the board you plan to program. Note: The Arduino Uno occupies the next available COM port; it will not always be COM3.

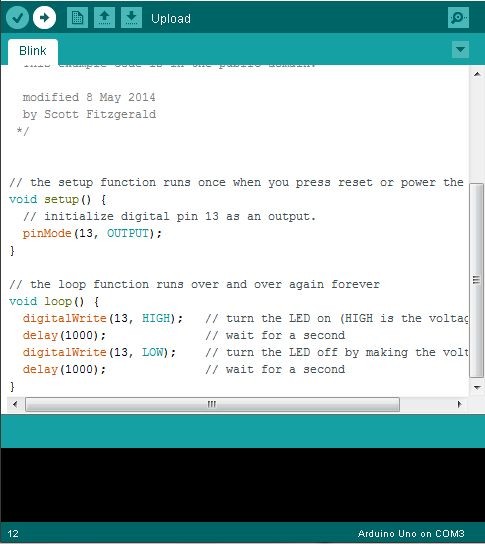
* + 1. **TESTING YOUR SETTINGS: UPLOADING BLINK**

One common procedure to test whether the board you are using is properly set up is to upload the “Blink” sketch. This sketch is included with all Arduino IDE releases and can be accessed by the Filepull-down menu and going to Examples, 01. Basics**,** and then select Blink. Standard Arduino Boards include a surface-mounted LED labeled “L” or “LED” next to the “RX” and “TX” LEDs, that is connected to digital pin

This sketch will blink the LED at a regular interval, and is an easy way to confirm if your board is set up properly and you were successful in uploading code. Open the “Blink” sketch and press the “Upload” button in the upper-left corner to upload “Blink” to the board.

Upload Button: 





* + 1. **GUIDE SUMMARY**
* Download and install Arduino ID
* Plug in your Arduino Board
* Select the proper board in the IDE (Tools>Boards>Arduino Uno)
* Select the proper COM port (Tools>Port>COMx (Arduino Uno))
* Open the “Blink” sketch (File>Examples>Basics>01.Blink)
* Press the Upload button to upload the program to the board
* Confirm that your board is working as expected by observing LED

**5.1.6 TROUBLE SHOOTING UPLOAD ERRORS**

Arduino has lots of community support and documentation. Your best bet when running into unexpected problems is to search online for help. You should be able to find a forum where someone had the same problem you are having, and someone helped them fix it. If you don’t find results, try modifying your search, or post on the Arduino forums.

* **My board isn’t listed under devices and is not recognized by IDE:**
  + Most likely, this means that the ATMega328p chip is not programmed with the Arduino firmware.
  + If you don’t have a separate Arduino available, let me know and I can use an Atmel Programmer to upload the firmware.
  + There may be hardware damage if you had the board plugged into USB and external power at the same time. You may have to replace the chip if this is the case.
* **Error Message: avrdude: stk500\_recv(): programmer is not responding**
  + Double-check that you are using the correct COM port.
  + Make sure that your Arduino Board is plugged into the computer.
* **The IDE says “Uploading…” after pressing the upload button, but nothing is happening.**
  + Double-check that you have the correct board selected in the **Tools** menu.
  + Depending on the size of your program, it may take a few seconds to upload. If you feel like it is taking too long, it may be encountering an error and you can try unplugging and plugging in the Arduino board.

**5.2 PLX-DAQ**

Parallax Data Acquisition tool (PLX-DAQ) software add-in for Microsoft Excel acquires up to 26 channels of data from any Parallax microcontrollers and drops the numbers into columns as they arrive. PLX-DAQ provides easy spreadsheet analysis of data collected in the field, laboratory analysis of sensors and real-time equipment monitoring.

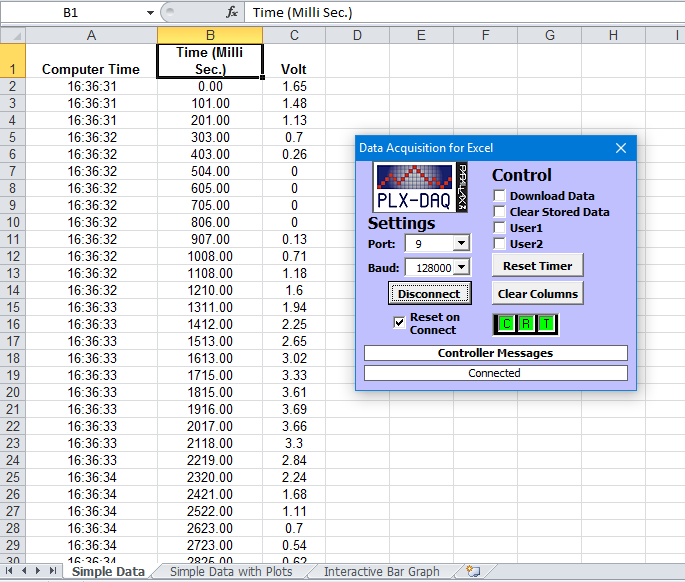
**PLX-DAQ Features**

PLX-DAQ is a Parallax microcontroller data acquisition add-on tool for Microsoft Excel. Any of our microcontrollers connected to any sensor and the serial port of a PC can now send data directly into Excel. PLX-DAQ has the following features:

* Plot or graph data as it arrives in real-time using Microsoft Excel
* Record up to 26 columns of data
* Mark data with real-time (hh:mm:ss) or seconds since reset
* Read/Write any cell on a worksheet
* Read/Set any of 4 checkboxes on control the interface
* Example code for the BS2, SX (SX/B) and Propeller available
* Baud rates up to 128K
* Supports Com1-15

**System Requirements**

* Microsoft Windows 98
* Microsoft Office/Excel 2000 to 2003
* May not work with newer software; no longer supported



**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

**6.1 CONCLUSION**

The RFID based security and attendance system is more secure and fast responded as compared to the other system like biometric. The main advantage of the RFID system is contact-less and works on non-line-of-sight. Tag works in any environmental condition. By using the Arduino platform system becomes more faster in response and while burning the code it is just like plug and play device. User can change the application accordingly by using Arduino. It is very easy to use and very accurate also. Hence this project can be very useful for implementing the real time application for recording the attendance and tracking system as well as providing the security benefits. This project can improve by increasing the range of reader in which the tag read. Further this project can be done by using a method in which the tag encrypts its ID and then send to the reader which can eliminate the capturing of the tag IDs.

**6.2 FUTURE SCOPE**

If we talk about future scopes then it depends upon how innovative one could be to enhance the use of this project. But for us this project is very useful for future uses

e.g. - 1) for tagging animals.

2) Uses in ATM machines.

3) To save students by tracking their attendance.

4) Use in car lock keys and engine start by placing the card in front of reader.

5) Make things environment friendly.

6) Gaming zone and many more future uses.

**ARDUINO**

**CODE**

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

#define LED\_G 6

#define LED\_R 7

#define BUZZER 8

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

int readsuccess;

byte readcard[4];

char str[32] = "";

String StrUID;

void setup() {

Serial.begin(9600); // Initialize serial communications with the PC

SPI.begin(); // Init SPI bus

mfrc522.PCD\_Init(); // Init MFRC522 card

Serial.println("CLEARDATA");

Serial.println("LABEL,Date,Time,RFID UID");

delay(1000);

pinMode(LED\_G, OUTPUT);

pinMode(LED\_R, OUTPUT);

pinMode(BUZZER, OUTPUT);

noTone(BUZZER);

Serial.println("Scan Card");

Serial.println("");

}

// --------------------------------------------------------------------

void loop() {

readsuccess = getid();

if(readsuccess){

if(StrUID == "50FE6056"){

Serial.println ("Access Granted" );

delay(100);

digitalWrite(LED\_G, HIGH);

tone(BUZZER, 1200);

delay(500);

noTone(BUZZER);

digitalWrite(LED\_G, LOW);

delay(500);

Serial.println( (String) "DATA,DATE,TIME,SAGAR,GRANTED," + StrUID );

Serial.println("");

}

else if(StrUID == "3BDBE4A9"){

Serial.println ("Access Granted" );

delay(100);

digitalWrite(LED\_G, HIGH);

tone(BUZZER, 1200);

delay(500);

noTone(BUZZER);

digitalWrite(LED\_G, LOW);

delay(500);

Serial.println( (String) "DATA,DATE,TIME,SOWMYA,GRANTED," + StrUID );

Serial.println("");

}

else if(StrUID == "66C0E4FA"){

Serial.println ("Access Granted" );

delay(100);

digitalWrite(LED\_G, HIGH);

tone(BUZZER, 1200);

delay(500);

noTone(BUZZER);

digitalWrite(LED\_G, LOW);

delay(500);

Serial.println( (String) "DATA,DATE,TIME,KEERTHIKA,GRANTED," + StrUID );

Serial.println("");

}

else

{

Serial.println("Access denied");

delay(100);

int n=0;

while(n<=2){

digitalWrite(LED\_R, HIGH);

tone(BUZZER, 100);

delay(200);

digitalWrite(LED\_R, LOW);

noTone(BUZZER);

delay(200);

n++;

}

Serial.println( (String) "DATA,DATE,TIME,DENIED," + StrUID );

Serial.println("");

}

}

}

// --------------------------------------------------------------------

int getid(){

if(!mfrc522.PICC\_IsNewCardPresent()){

return 0;

}

if(!mfrc522.PICC\_ReadCardSerial()){

return 0;

}

Serial.println("Scanned Card");

for(int i=0;i<4;i++){

readcard[i]=mfrc522.uid.uidByte[i]; //storing the UID of the tag in readcard

array\_to\_string(readcard, 4, str);

StrUID = str;

}

mfrc522.PICC\_HaltA();

Serial.println( (String) "" + StrUID);

return readcard;

}

// --------------------------------------------------------------------

void array\_to\_string(byte array[], unsigned int len, char buffer[])

{

for (unsigned int i = 0; i < len; i++)

{

byte nib1 = (array[i] >> 4) & 0x0F;

byte nib2 = (array[i] >> 0) & 0x0F;

buffer[i\*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;

buffer[i\*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;

}

buffer[len\*2] = '\0';

}

**REFERENCE:**

ARDUINO UNO: <https://store.arduino.cc/usa/arduino-uno-rev3>

PLX-DAQ: <https://www.parallax.com/downloads/plx-daq>

ARDUINO IDE: <https://www.arduino.cc/en/main/software>

RFID-RC522 interface with Arduino : <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>