1. **CHAPTER-1**

**INTRODUCTION**

**1.1 Project Introduction-**

Security systems play an important role to prevent unauthorized personnel entry into a secured environment, which may include physical and intellectual property. Various door locks such as mechanical locks or electrical locks were designed to attain basic security requirements. Basically, these locks can be easily hacked by unwanted people thereby allowing unauthorized personnel into secured premises. Automatic access control system has become necessary to overcome the security threats faced by many organizations in Nigeria. By installing the system at the entrance will only allow the authorized personnel to enter the organization. The system is not restricted to main entrance installation, but can be installed at various entrances within the organization to track personnel movement thereby restricting their access to areas where they not authorized. There are several automatic access control technologies including barcode, magnetic stripe and Radio Frequency Identification (RFID) applied in security system. Radio-Frequency Identification (RFID) is an emerging technology and one of the most rapidly growing segments of today’s access control. RFID technology, offers superior performance over other automatic identification systems and is used in many areas such as public transport, ticketing, animal identification, electronic immobilization, industrial automation, access control, asset tracking, people tracking, inventory detection and many more.



*Fig 1.1****: RFID Based Door Lock System***

Figure 1.1 show two different ways access control system can be accomplished. Use of keys which is old method and by use of RFID technology. This paper discusses the design of an automatic access control system using Arduino microcontroller and RFID system. The aim is granting access to authorized personnel and denying access to unauthorized personnel by using RFID technology instead of keys as shown in figure 1. Each person is issued an authorized tag, which can be used for swiping in front of the RFID reader to have access to a secured environment.

**1.2 Literature Review:**

**Umar et al** proposed an RFID based security and access control system. It is the design of RFID based security and access control system for use in hostels inside Punjab University promises. The system combines RFID technology and biometrics to accomplish the required task. When the RFID reader installed at the entrance of the hostel detects tag UID, the system captures the user’s image and scans the database for a match. If both card UID and captured image matches a registered user, access is granted; otherwise, access is denied and the system turns on alarm to alert the security personals.

The advantage of the system is that it successfully accomplished security and control task by processing information from sub-controllers like; entrance monitoring controller, exit monitoring controller and mess monitoring controller installed at entrance gate, exit gate and mess gate respectively. Although the developed system is useful in reducing security threats to the hostels, there is a room for improvement in the response time of the system.

The response time can be improved by using dedicated processors instead of computer systems capable of processing the images in real time. RFID based access control security system with GSM technology was proposed by **Peter et al** .

The work was archived through the use of RFID system operating on 125 KHz frequency, microcontroller programmed to send control signals, DC motor, relay, buzzer, liquid crystal display (LCD), and GSM/GPRS modem.

Once the RFID tag which contains the user’s unique information is scan by the RFID reader and confirmed match with the information stored in the microcontroller, the microcontroller is instructed to turn on the DC motor, display user number and card number on the LCD and activates the GSM/GPRS modem to send an SMS alert about authorized user card to the security personnel. Else, the DC motor remained off, LCD displays invalid card, buzzer turns on for about 5seconds, and GSM/GPRS modem activated to send unauthorized user card to the security personnel.

**CHAPTER-2**

**BLOCK DIAGRAM & EXPLAINATION**

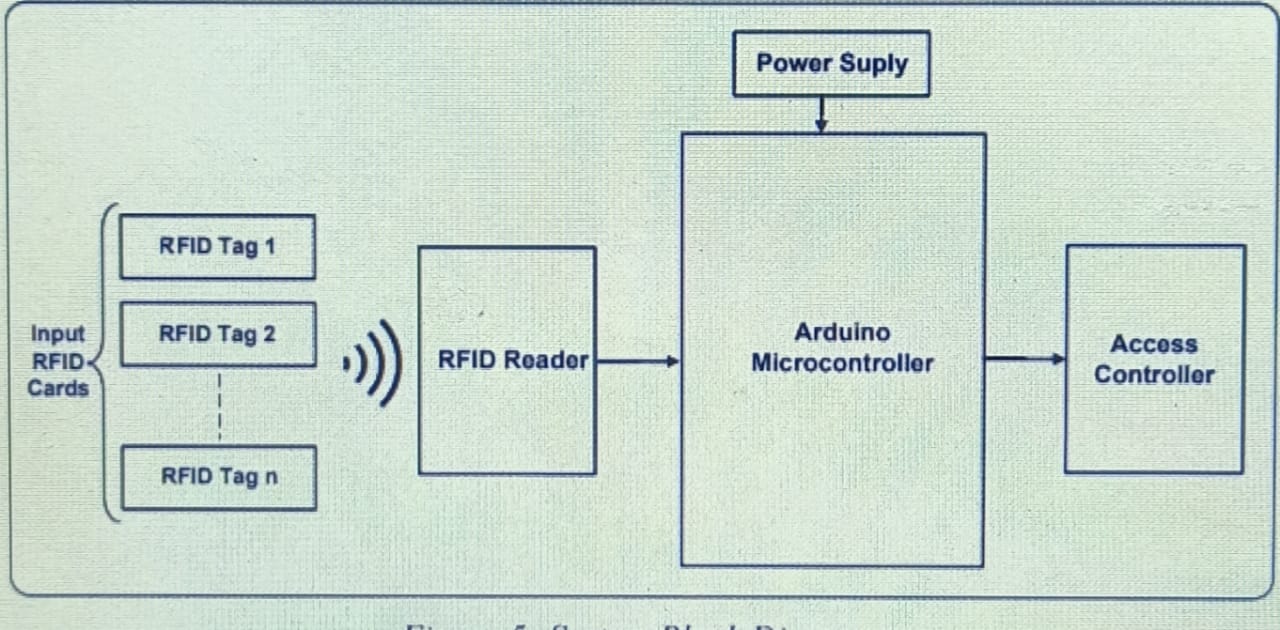


FIG:2.1**: System Block Diagram**

It shows the access control system block diagram using Arduino and RFID. The system has three separate parts, an RFID reader, a microcontroller, and an access controller. The RFID reader reads the RFID tags and the microcontroller accept the data from the reader process it and use the result to either grant or deny access to the user using access controller. The project can be enhanced by connecting an LCD display to display if access is granted or denied instead of serial monitor.

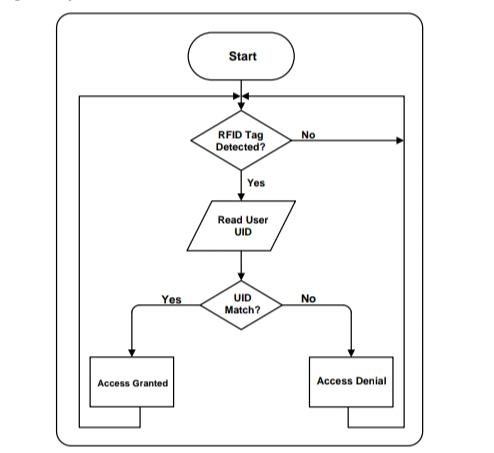
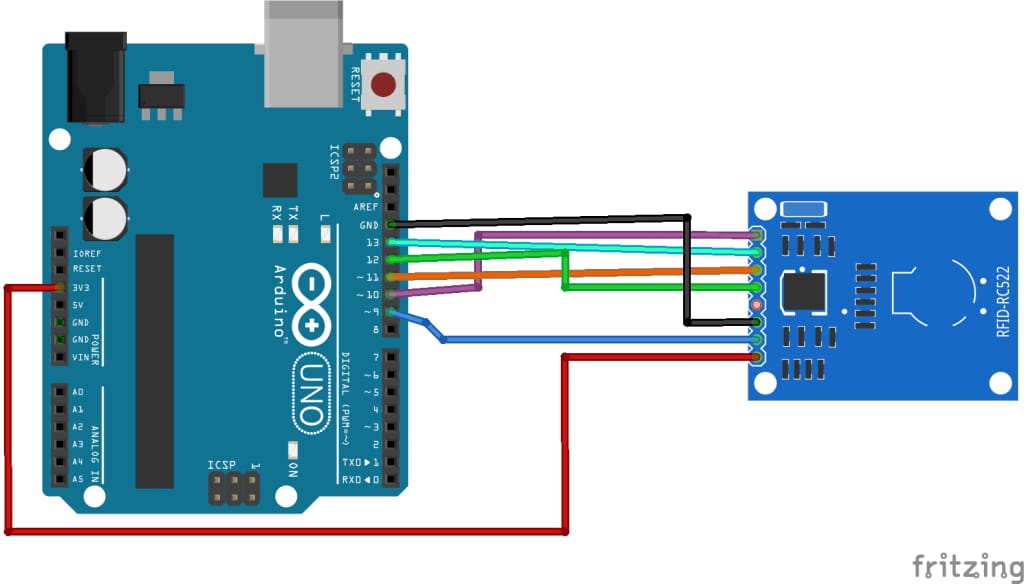


FIG2.2: ***: Control Flow Diagram***

**2.2:CONTROL FLOW**:

All necessary information about all users is stored in the system. In other to add a new user, we must first register the user with the system then, corresponding user information is burn in RFID tag. The new tag will now be accessible through the system. When a user comes to the entry point where the RFID reader is installed and places the RFID tag close (contactless) to the reader, the system checks whether it is a registered user or not. If the user is registered, the tag information is matched with the user information stored in system as shown in figure 2.2. Access is granted to such user while access is denied to unauthorized user see figure 2.2

**3**.**CIRCUIT DIAGRAM**



3.1: Automatic Access Control System using Arduino and RFID

In figure 3.1, when the RFID tag is placed close to the RFID reader, access is granted or denied. The right tag stored on the microcontroller grants access to the secure environment while the wrong tag not stored on the microcontroller will deny access to the card holder. Access granted or denied is displayed on the serial monitor.

**Table1: Connection of the RFID Reader with Arduino Microcontroller**

|  |  |
| --- | --- |
| RFID Module | Arduino |
| SDA | Digital Pin 10 |
| SCK | Digital Pin 13 |
| MOSI | Digital Pin 11 |
| MISQ | Digital Pin 12 |
| IRQ | No Connection |
| GND | GND |
| RST | Digital Pin 9 |
| 3.3V | 3.3v |

Table 1 shows how the Arduino is interfaced with the RFID reader. Also note that 3.3v of the RFID module must be connected to 3.3v on the Arduino board not 5v. With the above connections, the Arduino is ready to take commands and execute accordingly.

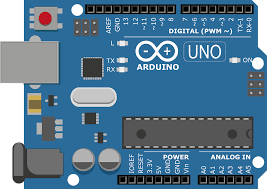
**4. COMPONENT DESCRIPTION**

**4.1:ARDIUNO UNO**

The Arduino Uno is one kind of microcontroller board based on ATmega328, and Uno is an Italian term which means one. Arduino Uno is named for marking the upcoming release of microcontroller board namely Arduino Uno Board 1.0. This board includes digital I/O pins-14, a power jack, analog i/ps-6, ceramic resonator-A16 MHz, a USB connection, an RST button, and an ICSP header. All these can support the microcontroller for further operation by connecting this board to the computer. The power supply of this board can be done with the help of an AC to DC adapter, a USB cable, otherwise a battery. This article discusses what is an Arduino Uno microcontroller, pin configuration, Arduino Uno specifications or features, and applications.

**What is Arduino Uno ATmega328?**

The ATmega328 is one kind of single-chip microcontroller formed with Atmel within the megaAVR family. The architecture of this Arduino Uno is a customized Harvard architecture with 8 bit RISC processor core. Other boards of Arduino Uno include Arduino Pro Mini, Arduino Nano, Arduino Due, Arduino Mega, and Arduino Leonardo



**Fig- 4.1 Arduino Uno**

**Features of Arduino Uno Board**

The features of Arduino Uno ATmega328 include the following.

1. The operating voltage is 5V

2. The recommended input voltage will range from 7v to 12V

3. The input voltage ranges from 6v to 20V

4. Digital input/output pins are 14

5. Analog i/p pins are 6

6. DC Current for each input/output pin is 40 mA

7. DC Current for 3.3V Pin is 50 mA

8. Flash Memory is 32 KB

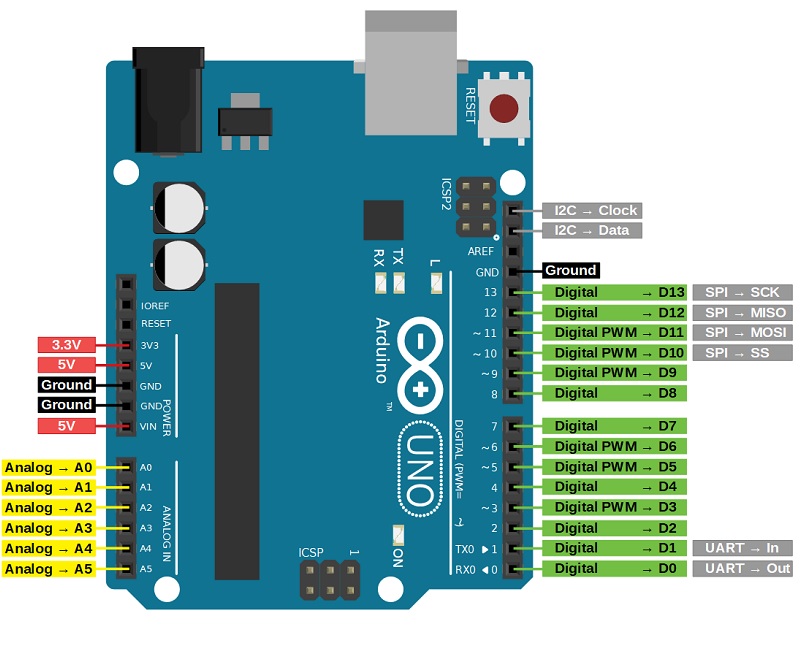
9. SRAM is 2 KB

10. EEPROM is 1 KB

11. CLK Speed is 16 MHz

**Arduino Uno Pin Diagram**

The Arduino Uno board can be built with power pins, analog pins, ATmegs328, ICSP header, Reset button, power LED, digital pins, test led 13, TX/RX pins, USB interface, an external power supply. The Arduino UNO board description is discussed below.



**Fig- 4.2 Arduino Uno Pin Diagram**

|  |  |  |
| --- | --- | --- |
| **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. |

*Pin description Table no.1*

**ARDUINO UNO TECHNICAL SPECIFICATIONS:**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 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 Boot loader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

*Technical specifications Table no.2*

**COMMUNICATING WITH ARDUINO UNO BOARD:**

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 Software Serial 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.

**How to Use an Arduino Uno?**

Arduino Uno can detect the surroundings from the input. Here the input is a variety of sensors and these can affect its surroundings through controlling motors, lights, other actuators, etc. The ATmega328 microcontroller on the Arduino board can be programmed with the help of an Arduino programming language and the IDE (Integrated Development Environment). Arduino projects can communicate by software while running on a PC.

**Arduino Programming**

Once the Arduino IDE tool is installed in the PC, attach the Arduino board to the computer with the help of USB cable. Open the Arduino IDE & select the right board by choosing Tools–>Board.>Arduino Uno, and select the right Port by choosing Tools–>Port. This board can be programmed with the help of an Arduino programming language depends on Wiring.

To activate the Arduino board & flash the LED on the board, dump the program code with the selection of Files–> Examples.>Basics..>Flash. When the programming codes are dumped into the IDE, and then click the button ‘upload’ on the top bar. Once this process is completed, check the LED flash on the board.

**High Voltage Protection of USB**

The Arduino Uno board has a rearrangeable poly fuse that defends the USB port of the PC from the over-voltage. Though most of the PCs have their own inner protection, the fuse gives an additional coating of safety. If above 500mA is given to the USB port, then the fuse will routinely crack the connection until the over-voltage is removed.

**Physical Characteristics**

The physical characteristics of an Arduino board mainly include length and width. The printed circuit board of the Arduino Uno length and width are 2.7 X 2.1 inches, but the power jack and the USB connector will extend beyond the previous measurement. The board can be attached on the surface otherwise case with the screw holes.

**Applications of Arduino Uno ATmega328**

The applications of Arduino Uno include the following.

1. Arduino Uno is used in Do-it-Yourself projects prototyping.

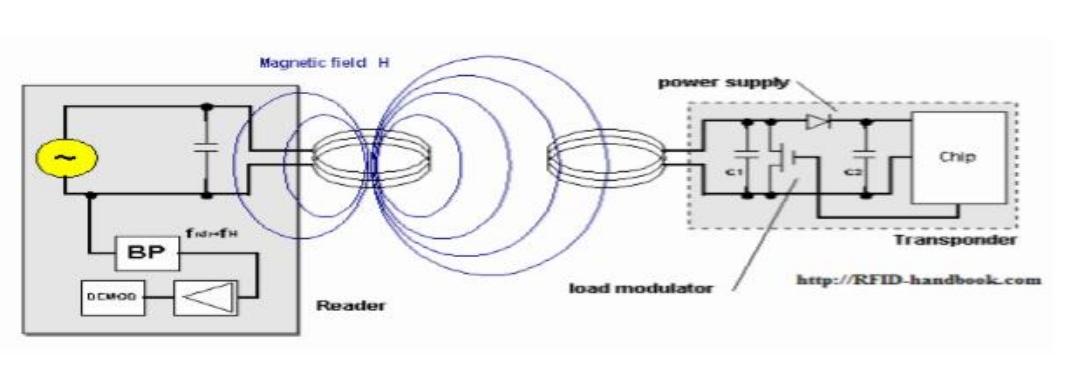
2. In developing projects based on code-based control

3. Development of Automation System

4. Designing of basic circuit designs.

**4.2: RFID(Radio Frequency Identification):**

RFID stands for Radio Frequency Identification. RFID is one member in the family of Automatic Identification and Data Capture (AIDC) technologies and is a fast and reliable means of identifying objects. There are two main components: The Interrogator (RFID Reader) which transmits and receives the signal and the Transponder (tag) that is attached to the object. An RFID tag is composed of a miniscule microchip and antenna. RFID tags can be passive or active and come in a wide variety of sizes, shapes, and forms. Communication between the RFID Reader and tags occurs wirelessly and generally does not require a line of sight between the devices. An RFID Reader can read through most anything with the exception of conductive materials like water and metal, but with modifications and positioning, even these can be overcome.

**Working of RFID FIGURE**

**4.3: Working of RFID**

In its simplest form a product tag, made of a microchip with a tiny antenna, is attached to a product. An associated tag reader puts out electromagnetic waves. The tag antenna receives the waves and the tag itself draws power from the field generated by the reader, powering the chip, and then modulates the reader signal, sending it back where it is converted into digital data. The electromagnetic waves are harmless at the low end of the spectrum and no more dangerous than a car radio.

RFID tags come in a wide range of flavors including the passive type described above, battery powered, multi-frequency and tag-talks-first. RFID antenna types can vary too. Whats more, not all RFID systems use low-frequency EM waves. There are read-only tags and read-write tags. There are tags holding up to 2K of product data and tags that contain only a single product ID. Tags can also be used for more than product IDs; they can be used in environmental monitors, security devices, and product integrity mechanisms

**RFID Frequencies:**

Radio waves are the carriers of data between the reader and tags. The approach generally adopted for RFID communication is to allocate frequencies depending on application. The frequencies used cover a wide spectrum.

These specified bands are

Very long wave 9 - 135 kHz Short wave 13.56MHz

UHF 400-1200 MHz Microwave 2.45 and 5.8 GHz

The allocation of frequencies is regulated by government agencies, requiring care in considering RFID applications in different countries. Efforts at standardization should avert these problems. The many varied applications will work their best at different frequencies; therefore, it is important to understand the requirements before selecting a particular type of RFID system. The most common uses of low frequency systems are in security access, asset tracking and animal identification. They generally have short reading ranges and lower system costs. High-frequency systems are used for such applications as railroad car tracking and automated toll collection. 14 They offer long reading ranges and high reading speeds. This higher performance usually entails higher costs. The power level of the interrogator and the power available within the tag to respond will determine the reading range that can be achieved in an RFID system. Like the restrictions on carrier frequencies there are legislative constraints on power levels. Environmental conditions, particularly at the higher frequencies, can also influence the range of communication.

|  |
| --- |
| **RFID frequency bands** |
| **Band** | **Regulations** | **Range** | **Data speed** | [**ISO/IEC 18000**](https://en.wikipedia.org/wiki/ISO/IEC_18000) **section** | **Remarks** | **Approximate tag cost in volume (2006)** |
| LF: 120–150 kHz | Unregulated | 10 cm | Low | [Part 2](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=46146) | Animal identification, factory data collection | US$1 |
| HF: 13.56 MHz | [ISM band](https://en.wikipedia.org/wiki/ISM_band) worldwide | 10 cm–1 m | Low to moderate | [Part 3](https://en.wikipedia.org/wiki/ISO/IEC_18000-3) | Smart cards ([ISO/IEC 15693](https://en.wikipedia.org/wiki/ISO/IEC_15693), [ISO/IEC 14443](https://en.wikipedia.org/wiki/ISO/IEC_14443) A, B), ISO-non-compliant memory cards ([Mifare](https://en.wikipedia.org/wiki/Mifare" \o "Mifare) Classic, iCLASS, Legic, Felica ...), ISO-compatible microprocessor cards (Desfire EV1, Seos) | US$0.50 to US$5 |
| UHF: 433 MHz | Short range devices | 1–100 m | Moderate | [Part 7](https://www.iso.org/standard/57336.html) | Defense applications, with active tags | US$5 |
| UHF: 865–868 MHz (Europe) 902–928 MHz (North America) | ISM band | 1–12 m | Moderate to high | [Part 6](https://www.iso.org/standard/59644.html) | EAN, various standards; used by railroads[[16]](https://en.wikipedia.org/wiki/Radio-frequency_identification#cite_note-16) | US$0.15 (passive tags) |
| [microwave](https://en.wikipedia.org/wiki/Microwave): 2450–5800 MHz | ISM band | 1–2 m | High | [Part 4](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=62539) | 802.11 WLAN, Bluetooth standards | US$25 (active tags) |
| microwave: 3.1–10 GHz | Ultra wide band | up to 200 m | High | not defined | Requires semi-active or active tags | US$5 projected |

### Signaling

**Types of TAG:**

Most RFID tags contain at least two parts. One is an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal. There are generally two types of RFID tags: active RFID tags, which contain a battery and thus can transmit its signal autonomously, and passive RFID tags, which have no battery and require an external source to initiate signal transmission.

**Passive TAG:**

Passive tags are generally smaller, lighter and less expensive than those that are active and can be applied to objects in harsh environments, are maintenance free and will last for years. These transponders are only activated when within the response range of a reader. The RFID reader emits a low- power radio wave field which is used to power up the tag so as to pass on any information that is contained on the chip.

**Active TAG:**

Active tags differ in that they incorporate their own power source, where as the tag is a transmitter rather than a reflector of radio frequency signals which enables a broader range of functionality like programmable and read/write capability

**Signaling**



**FIG4.4:RFID Tag**

Signaling between the reader and the tag is done in several different incompatible ways, depending on the frequency band used by the tag. Tags operating on LF and HF bands are, in terms of radio wavelength, very close to the reader antenna because they are only a small percentage of a wavelength away. In this [near field](https://en.wikipedia.org/wiki/Near_and_far_field) region, the tag is closely coupled electrically with the transmitter in the reader.

The tag can modulate the field produced by the reader by changing the electrical loading the tag represents. By switching between lower and higher relative loads, the tag produces a change that the reader can detect. At UHF and higher frequencies, the tag is more than one radio wavelength away from the reader, requiring a different approach.

The tag can [backscatter](https://en.wikipedia.org/wiki/Backscatter) a signal. Active tags may contain functionally separated transmitters and receivers, and the tag need not respond on a frequency related to the reader's interrogation signal.

An [Electronic Product Code](https://en.wikipedia.org/wiki/Electronic_Product_Code) (EPC) is one common type of data stored in a tag. When written into the tag by an RFID printer, the tag contains a 96-bit string of data. The first eight bits are a header which identifies the version of the protocol. The next 28 bits identify the organization that manages the data for this tag; the organization number is assigned by the EPC Global consortium.

The next 24 bits are an object class, identifying the kind of product. The last 36 bits are a unique serial number for a particular tag. These last two fields are set by the organization that issued the tag. Rather like a [URL](https://en.wikipedia.org/wiki/Uniform_resource_locator), the total electronic product code number can be used as a key into a global database to uniquely identify a particular product.

Often more than one tag will respond to a tag reader, for example, many individual products with tags may be shipped in a common box or on a common pallet. Collision detection is important to allow reading of data. Two different types of protocols are used to ["singulate"](https://en.wikipedia.org/wiki/Singulation) a particular tag, allowing its data to be read in the midst of many similar tags.

In a [slotted Aloha](https://en.wikipedia.org/wiki/ALOHAnet) system, the reader broadcasts an initialization command and a parameter that the tags individually use to pseudo-randomly delay their responses. When using an "adaptive binary tree" protocol, the reader sends an initialization symbol and then transmits one bit of ID data at a time; only tags with matching bits respond, and eventually only one tag matches the complete ID string.

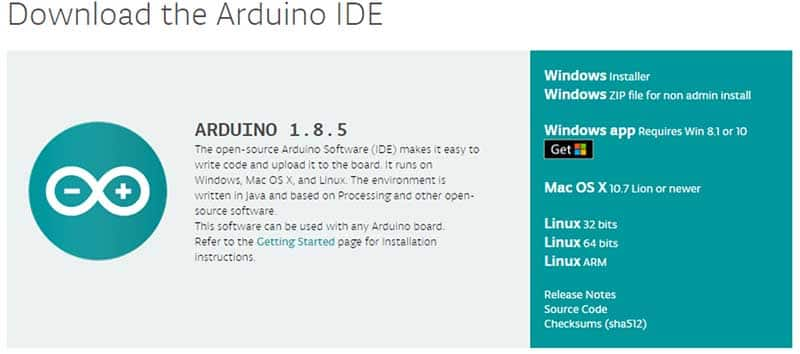
**5.Software**

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

**5.1 STARTING WITH ARDUINO IDE:**

**Step 1**: Download and Install the IDE

You can download the IDE from the official [Arduino website](https://www.arduino.cc/en/Main/Software). Since the Arduino uses a USB to serial converter (which allow it to communicate with the host computer), the Arduino board is compatible with most computers that have a USB port. Of course, you will need the IDE first. Luckily, the Arduino designers have released multiple versions of the IDE for different operating systems, including Windows, Mac, and Linux. In this tutorial, we will use Window 10, so ensure that you download the correct version of the IDE if you do not have Windows 10.



*Arduino ide Figure 5.1*

Once downloaded, install the IDE and ensure that you enable most (if not all) of the options, INCLUDING the drivers.

**Step 2**: Configure the IDE

### *Now that we have determined the COM port that the Arduino is on, it’s time to load the Arduino IDE and configure it to use the same device and port. Start by loading the IDE. Once it’s loaded, navigate to Tools > Board > Arduino Uno. However, if you are using a different board (i.e., not the Arduino Uno), you must select the proper board!*

### Tell the IDE which board you are using

*Arduino board selection Figure5.2*

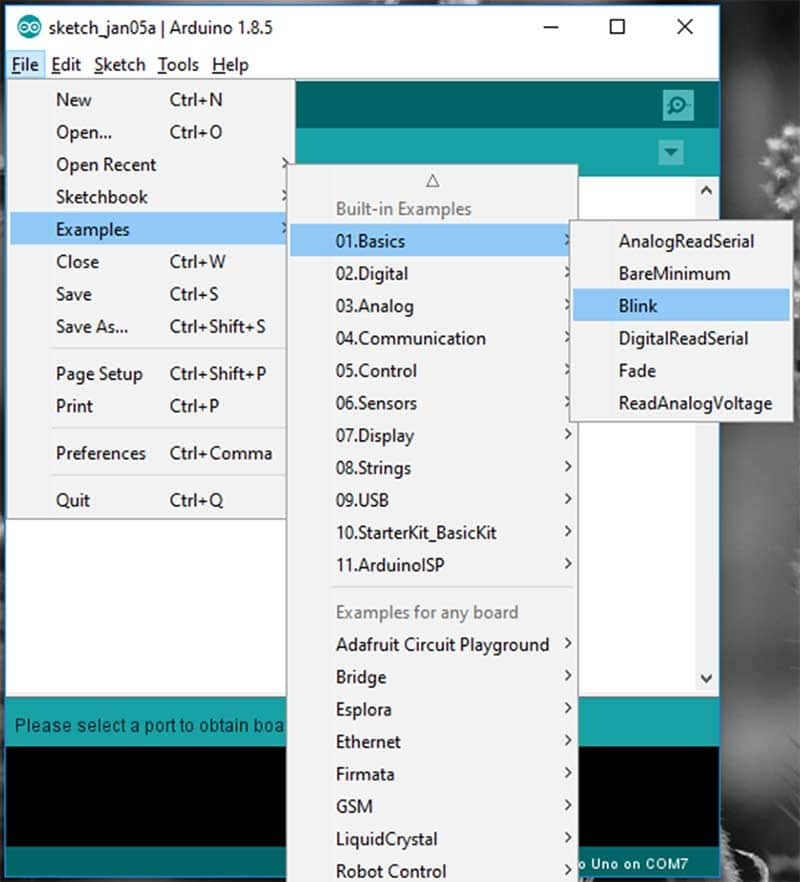
### Next, you must tell the IDE which COM port the Arduino is on. To do this, navigate to Tools > Port > COM7. Obviously, if your Arduino is on a different port, select that port instead.

### If your Arduino is on a different port, select that port instead

*Arduino Port selection Figure 5.3*

### Step 3:Loading a Basic Example

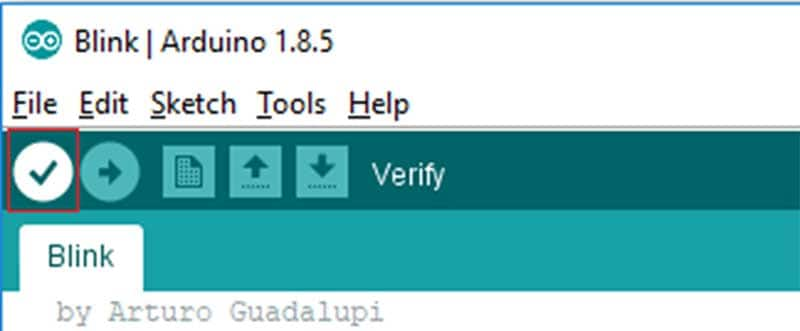
For the sake of simplicity, we will load an example project that the Arduino IDE comes with. This example will make the onboard LED blink for a second continuously. To load this example, click File > Examples > 01.Basics > Blink.



*Arduino basic program load Figure 5.4*

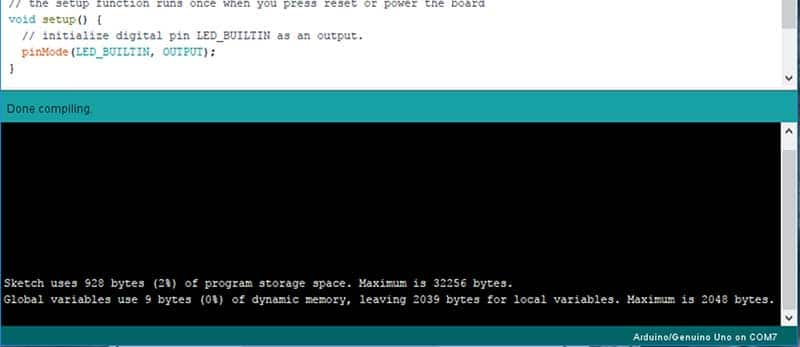
With the example loaded, it’s time to verify and upload the code. The verify stage checks the code for errors, then compiles the ready-for-uploading code to the Arduino. The upload stage actually takes the binary data, which was created from the code, and uploads it to the Arduino via the serial port.

To verify and compile the code, press the check mark button in the upper left window.



*Compiling program Figure 5.5*

If the compilation stage was successful, you should see the following message in the output window at the bottom of the IDE. You might also see a similar message—just it’s one that does not have words like “ERROR” and “WARNING”.



*No error Figure 5.6*

With the code compiled, you must now upload it the Arduino Uno. To do this, click the arrow next to the check mark.

### The “Upload” button will program the Arduino with your code

*Uploading program Figure 5.7*

### The Arduino is a powerful prototyping tool for many reasons, including its lack of a dedicated programmer, its wide range of available libraries, and the simplicity of its IDE. While we only got a light to blink in this project, you can expect much more in the future. Try your hand at interfacing with displays, taking measurements, talking over the internet, and possibly even working with AI!

**5.2: SOURCE CODE OF THE PROJECT:**

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

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

void setup()

{

Serial.begin(9600); // Initiate a serial communication

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

Serial.println("Put your card to the reader...");

Serial.println();

}

void loop()

{

// Look for new cards

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

// Select one of the cards

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

//Show UID on serial monitor

Serial.print("UID tag :");

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

Serial.print(mfrc522.uid.uidByte[i], HEX);

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

Serial.println();

Serial.print("Message : ");

content.toUpperCase();

if (content.substring(1) == "67 1E 50 34" ) //change here the UID of the card/cards that you want to give access

{

Serial.println("Authorized access");

Serial.println();

delay(500);

}

else

{

Serial.println(" Access denied");

}

}

Once the program sees the correct combination of tag UID it will grant access to the user. While access will be denied to incorrect tag UID or tags that are not store in the system. Tag UID can be added or remove by admin through the IDE and re-uploaded back the Arduino board for effective operation.

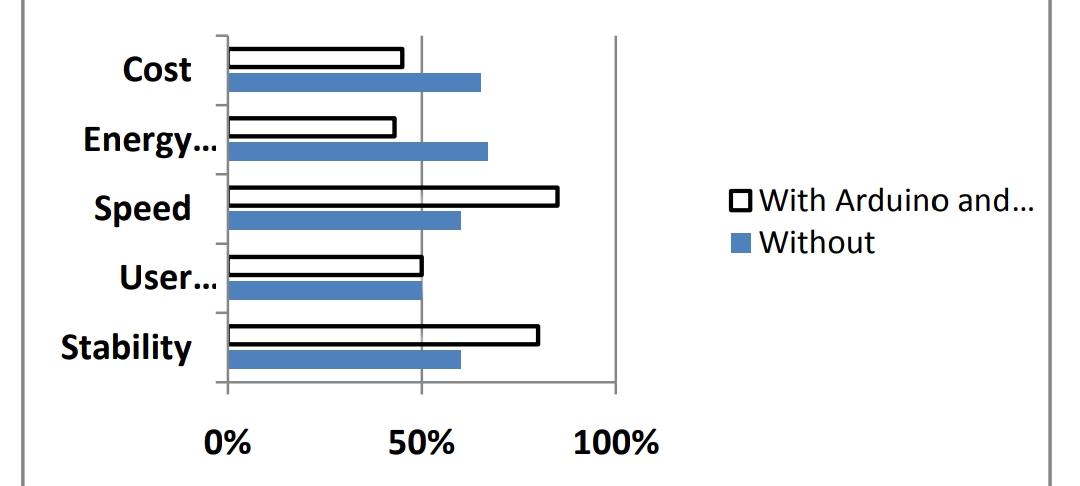
**5.3: Results-**

Results The project has the following workflow: on arriving at the door where the access control is installed, one is asked to approximate their RFID tag to the reader as show on the output window . The reader reads the tag and the microcontroller compare the tag’s UID for match and grant access if there is a match and deny access if there is no match

**5.3.1: Output through serial monitor**

An RFID tag can be added or removed through the Arduino IDE or any other programming language that Arduino understands. For changes made on the sketch (i.e. adding or removing a tag) to be effective on our system, the sketch must be re-uploaded to the Arduino board to override previous sketch.

**5.4: Access Control System Analysis:**

****

**5.4.1:** **Access Control System with Arduino and RFID and without**

Access control system was analyzed using the following criteria: cost, energy consumption, speed, user satisfaction, and stability. The bar chart shows that there are significant enhancements in access control system using Arduino and RFID technology. The enhancements come in the area of cost, energy consumption, speed, and stability. Other access control systems have high energy consumption rate which is a great problem in Nigeria where energy is very expensive, while the users of the system have equal satisfaction in both systems.