

**DEPARTMENT OF ELECTRICAL AND ELCTRONICS ENGINEERING**

***A Mini Project Report***

***On***

**“RFID Based Access Control System”**

*Submitted in the partial fulfilment of the Mini Project*

***Submitted by***

|  |  |
| --- | --- |
| **NAME-NIRANJAN C** | **USN- 1NH18EE731** |
| **NAME- ABHIMANYU IYER** | **USN-1NH18EE700** |
| **NAME- DHANUSH L** | **USN-1NH18EE716** |

Under the Guidance of

***Prof. Anitha***

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

**Academic Year - 2019-2020**

****

**Academic Year – 2019 – 2020**

**DEPARTMENT OF ELECTRICAL AND ELCTRONICS ENGINEERING**

**CERTIFICATE**

**Certified that the mini project work entitled “RFID based Access Control System” carried out by NIRANJAN C, ABHIMANYU IYER and DHANUSH L,bonafied students of New Horizon College of Engineering submitted report in completion of mini project under the guidance of PROF.ANITHA of Department of Electrical and Electronics, New Horizon College of Engineering during the academic year 2019-2020. It is certified that all the corrections and suggestions indicated for Internal Assessment have been approved as it satisfies the academic requirements in respect of internship work prescribed work prescribed for said degree.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head Of the Department**

**Dr.S.Ramkumar**

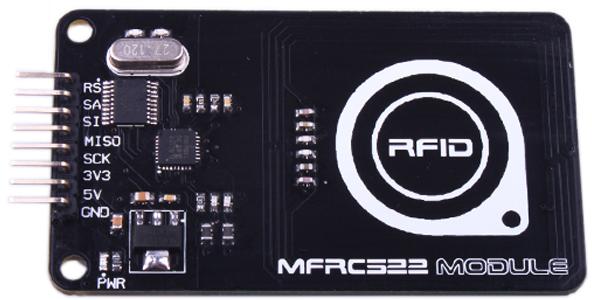
**RFID-RADIO FREQUENCY ID**

-About RFID reader

**Radio-frequency identification** (**RFID**) electromagnetic fields are used by it to identify and match tags attached to objects giving it a unique ID. The electronically-stored information is contained by it.The passive tag collects energy from a nearby energy signals of an rfid reader. Active tags have their own power source (such as a battery) and have a larger range compared to passive tags as it has its own battery.

Many industries use RFID tags, for example, an RFID tag is placed on an automobile during production progress can be tracked through the tags; Warehouses can tag inventory of pharmaceuticals through tags; and RFID microchips in cattle and farm animals allows for their identification and numbers.

Cash, clothing, and possessions, animals and people can be attached or implanted by an RFID tag, the possibility of a privacy breach has raised many alarms.These concerns resulted in the addressing all the issues of leaking personal or sensitive data. ISO/IEC 20248, a digital signature data structure for RFID is specified by it and barcodes, data, source and read method authenticity are provided by it. Tags are also used by shops, this is to prevent theft by customers and employees.



An MFRC522 RFID reader.

-Applications

* Logistics and supply chain visibility.
* Item level inventory tracking
* Race timings
* Attendance system
* Access control
* IT asset tracking
* Library system

And many more…..

**TYPES OF RFID TAGS**

There are mainly two types of RFID tags:

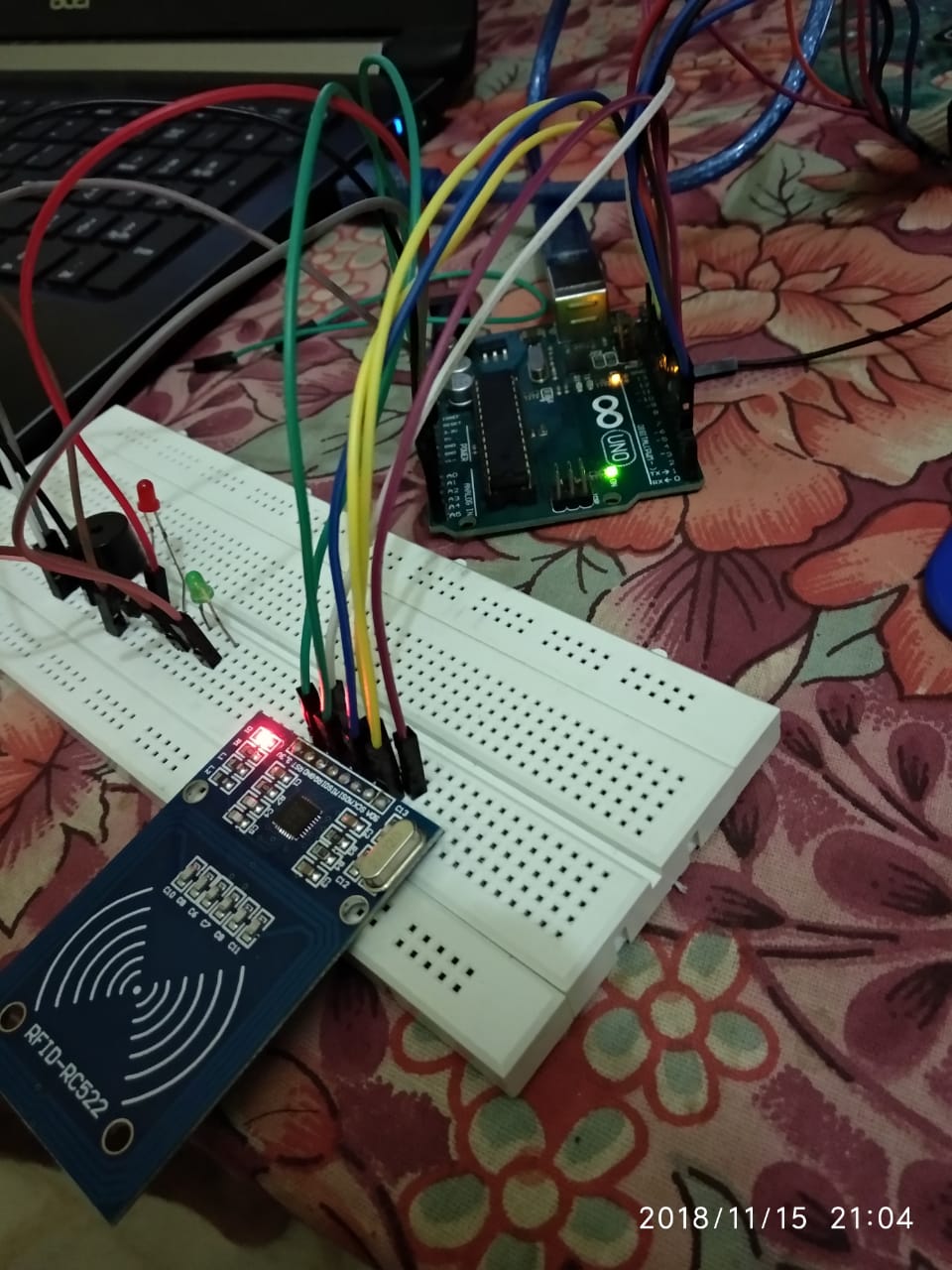
* An **Active Reader Passive Tag** (**ARPT**) this type of tag has an active reader and energy or interrogatory signals are transmitted by it. Also, authentication is received by the passive tags.
* An **Active Reader Active Tag** (**ARAT**) system, in this active tags are used by the system with an energy signal to trigger the signal from the active reader. Another sub type of tags that have its own battery source.



Some RFID tags

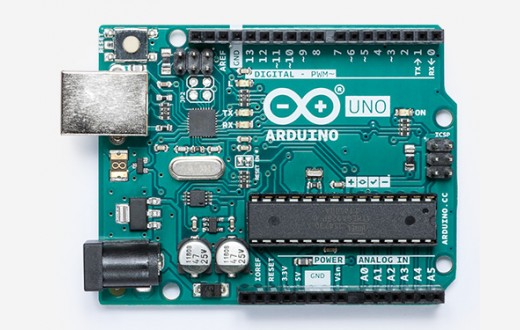
**RFID BASED ACCESS CONTROL SYSTEM**

* Materials
* Circuit diagram
* Connections
* Working
* Block diagram
* Program



**MATERIALS REQUIRED**

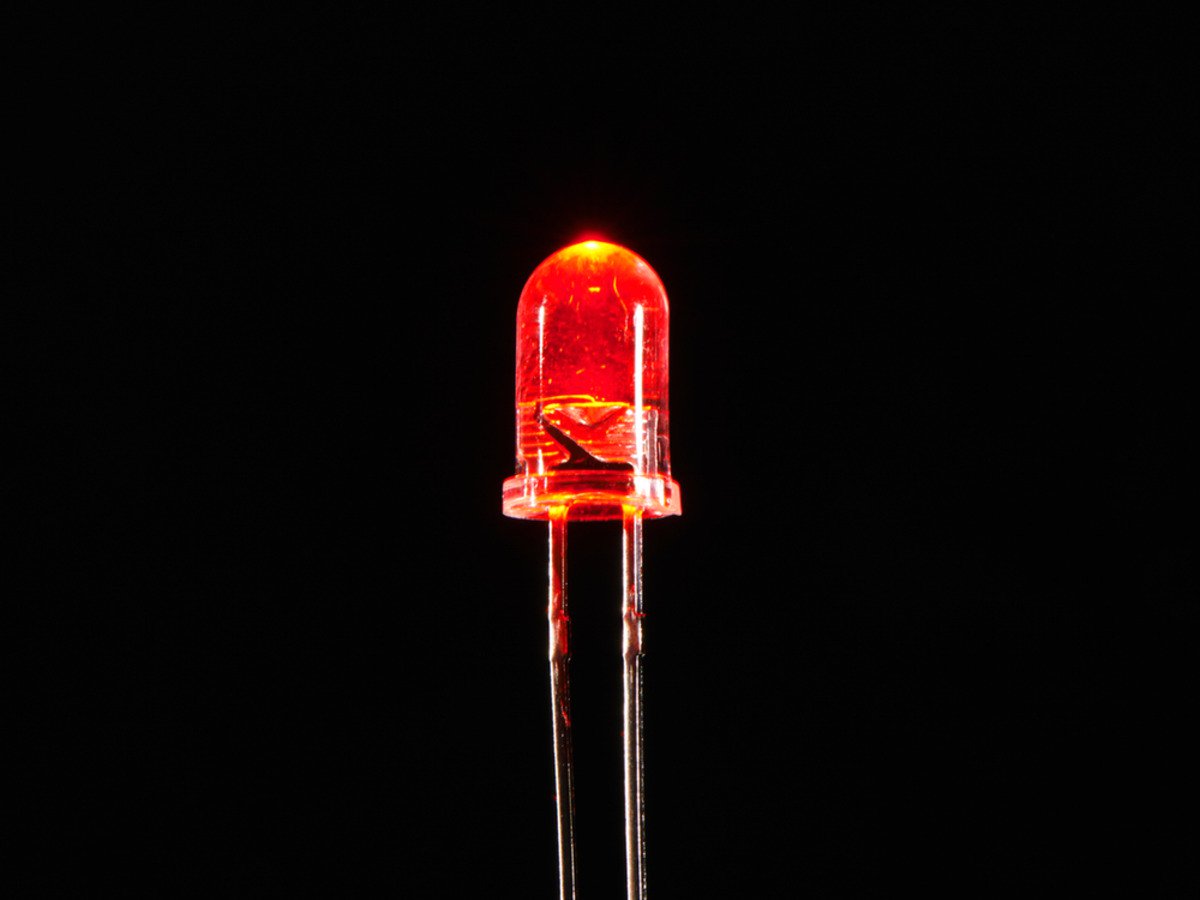
* Arduino Uno



* Keypad 4\*3



* Red LED



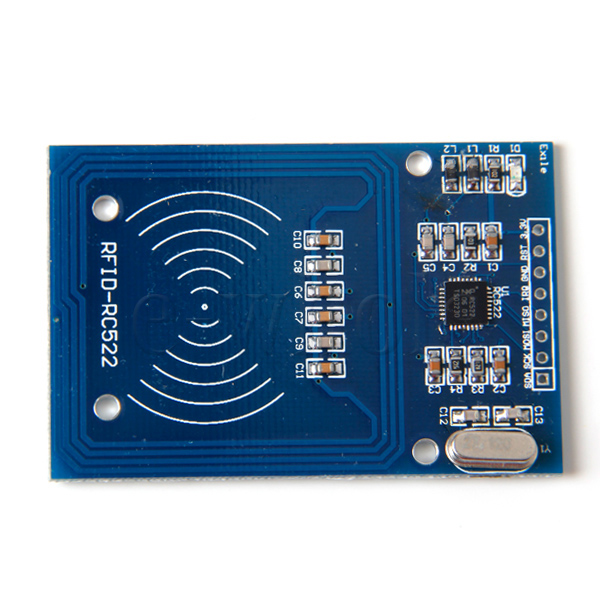
* Green LED



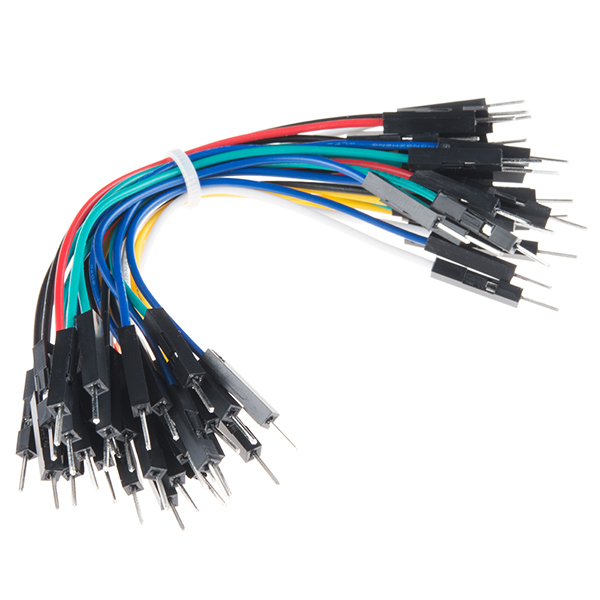
* Buzzer 5V



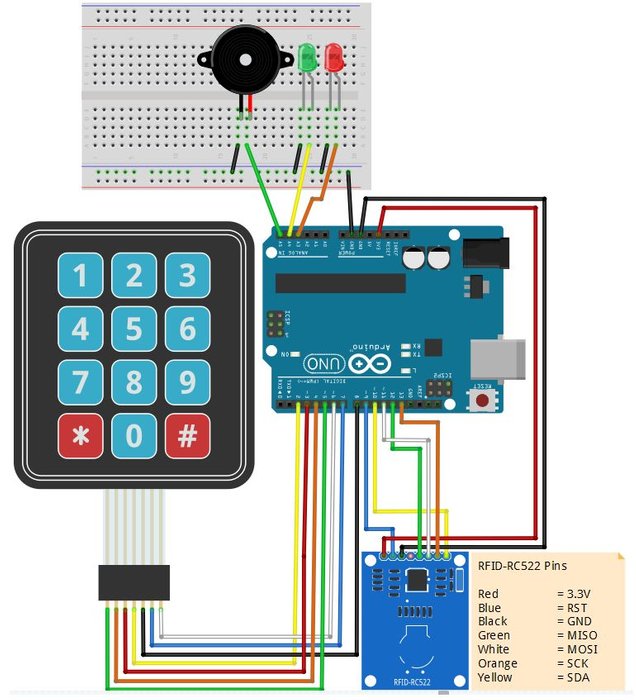
* MFRC522 module



* Jumper cables



**CIRCUIT DIAGRAM**



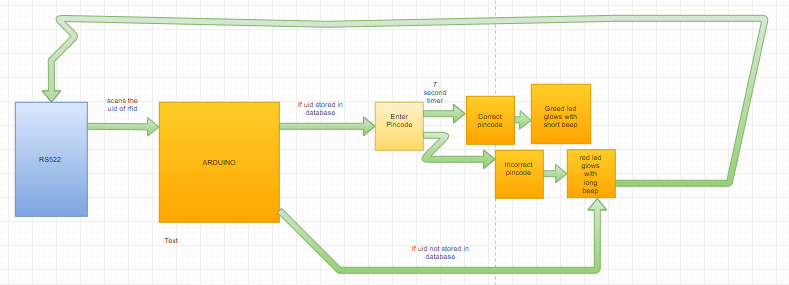
**CONNECTIONS**

* Arduino 3.3V to RC522 3.3V
* Arduino GND to RC522 GND
* Arduino A3(analog) to Red LED (+)
* Arduino A4(analog) to Green LED (+)
* Arduino A5(analog) to Buzzer (+)
* Arduino GND to Breadboard (-)
* Breadboard (-) to Red LED (-)
* Breadboard (-) to Green LED (-)
* Breadboard (-) to Buzzer (-)
* Arduino Pin 13 to RC522 SCK (serial clock)
* Arduino Pin 12 to RC522 MISO (Master in Slave out)
* Arduino Pin 11 to RC522 MOSI (Master out Slave in)
* Arduino Pin 10 to RC522 SDA
* Arduino Pin 9(digital) to RC522 RST (Reset)
* Arduino Pin 8(digital) to Keypad Pin 3 (Black)
* Arduino Pin 7(digital) to Keypad Pin 2 (Blue)
* Arduino Pin 6(digital) to Keypad Pin 1 (White)
* Arduino Pin 5(digital) to Keypad Pin 7 (Green)
* Arduino Pin 4(digital) to Keypad Pin 6 (Orange)
* Arduino Pin 3(digital) to Keypad Pin 5 (Red)
* Arduino Pin 2(digital) to Keypad Pin 4 (Yellow)

**WORKING**

* First you make the connections according to the circuit diagram.
* After uploading the code to the Arduino the project will begin to function.
* Basically, there are two RFID tags. One that has access and the other that does not have access.
* We place the tags over the RFID reader for scanning. If the RFID is authorized. The user has to put in a pincode set for that particular tag.
* The user has about 7 seconds to put in a pincode. If he/she doesn’t then a long beep with red LED will be the output.
* If the user enters in the correct pincode, the green LED glows and the buzzer gives out a short beep.
* And displaying correct pincode on screen.
* The pincode will be typed in through the 4\*3 keypad.

**EXPLANATION OF BLOCK DIAGRAM**



This block diagram is small systematic explanation of our project.

First the Arduino takes in the input from the RS522 and reads the UID of the tags.

If the UID is stored in database there is a second level of id and that is to enter in a pincode.

The outcomes of wrong, correct pincodes are given above in the diagram.

The timer of 7 seconds has also been typed in code to give some amount of time to user.

**EXPLANATION OF PROGRAM**

Before the program is even written due precautions should be taken as to adding the correct libraries to the software.

Another step that has been done before writing the code.

The UID of the RFIDs have already been obtained through running another sketch in the software.

#include <LiquidCrystal.h>

#include <Keypad.h> // This sketch uses the Keypad.h library

#include <SPI.h> // This sketch uses the SPI.h library

#include <MFRC522.h> // This sketch uses the MFRC522.h library

#define SS\_PIN 10 // Define SS\_PIN of the RC522 RFID Reader to digital pin 10 of the Arduino

#define RST\_PIN 9 // Define RST\_PIN of the RC522 RFID Reader to digital pin 9 of the Arduino

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

In this step we are including all the libraries and defining the pins. Essentially telling the Arduino what is input and output.

intRedLed = A3; // Red LED output is analog pin A3

intGreenLed = A4; // Green LED output is analog pin A4

int Buzzer = A5; // Buzzer output is analog pin A5

unsigned long Timer; // Variable for the Timer

const byte ROWS = 4; // Four rows

const byte COLS = 3; // Three columns

char Keys[ROWS][COLS] = { // Define the symbols on the buttons of the keypads

{'1','2','3',},

{'4','5','6',},

{'7','8','9',},

{'\*','0','#',}

};

Initialization of the LEDs and keypad.

int MifareCard1; // We need this specific variable to determine the valid PinCode in combination with the UID nummer of the Mifare Card

constint Code1MifareCard1 = '6'; // Determine here the first digit of your 6 digit PIN

constint Code2MifareCard1 = '6'; // Determine here the second digit of your 6 digit PIN

constint Code3MifareCard1 = '6'; // Determine here the third digit of your 6 digit PIN

constint Code4MifareCard1 = '2'; // Determine here the fourth digit of your 6 digit PIN

constint Code5MifareCard1 = '2'; // Determine here the fifth digit of your 6 digit PIN

constint Code6MifareCard1 = '2'; // Determine here the sixth digit of your 6 digit PIN

Setting the pincode for the RFID tag.

if (Reset == 1) { RightCard = 0;

MifareCard1 = 0;

MifareCard2 = 0;

RightPinCode = 0;

WrongPinCode = 0;

Code1Correct = 0;

Code2Correct = 0;

Code3Correct = 0;

Code4Correct = 0;

Code5Correct = 0;

Code6Correct = 0;

PinCodeCounter = 0; delay(50); reset = 0; }

(PTO)

Above code is for the reset function of the reader. Where it waits for an RFID to be scanned.

if (millis()- Timer > 7000 &&RightCard == 1) -If the Timer will be larger as 7 seconds and a valid Mifare Card is offered, reset value will be high {

digitalWrite (Buzzer, HIGH); -Make a long beep and set the Red LED HIGH for the wrong code or timer expiry

digitalWrite (RedLed, HIGH); delay(1500); - This means that you have after presenting a valid card, 7 seconds to enter a valid code or the loop resets.

Reset = 1;

Serial.println("Scanned card not authorized") } – To print message.

if

(mfrc522.uid.uidByte[0] == 0xAB && // Fill in the first digits of the UID number of your Mifare card

mfrc522.uid.uidByte[1] == 0xD2 && // Fill in the seconds digits of the UID number of your Mifare card

mfrc522.uid.uidByte[2] == 0x63 && // Fill in the third digits of the UID number of your Mifare card

mfrc522.uid.uidByte[3] == 0xA3) // Fill in the fourth digits of the UID number of your Mifare card

This block of code is where we put in the scanned UID of the RFID. The if condition is checking to match the hex numbers.

RightCard = 1; // The offered Mifare Card is known in the database, set variable "RightCard" to 1

MifareCard1 = 1; // This Mifare Card is known as Mifare Card 2 in the Database, so set variable MifareCard2 to value 1.

digitalWrite (Buzzer, HIGH); // Make a beep for the accepted Mifare Card

delay (150); //

digitalWrite (Buzzer, LOW); //

PinCodeCounter = 0; // Reset the PinCodeCounter to 0

Timer = millis(); // Reset the Timer. The 7 seconds limit is running now for a valid PinCode

Serial.println("RFID authorized, please enter pincode"); // Print the text "CardAccesOn" to the serial monitor

delay (200);

This block of code is executed if the RFID is authorized. (above)

if

(mfrc522.uid.uidByte[0] == 0x51 && // Fill in the first digits of the UID number of your Mifare card

mfrc522.uid.uidByte[1] == 0x51 && // Fill in the seconds digits of the UID number of your Mifare card

mfrc522.uid.uidByte[2] == 0x24 && // Fill in the third digits of the UID number of your Mifare card

mfrc522.uid.uidByte[3] == 0x2E) // Fill in the fourth digits of the UID number of your Mifare card {

digitalWrite (Buzzer, HIGH); // Make a long beep and set the Red LED HIGH for the wrong code or timer expiry

digitalWrite (RedLed, HIGH); //

delay(1500); //

digitalWrite (Buzzer, LOW); //

digitalWrite (RedLed, LOW); //

Serial.println("RFID not authorized");

(pto)

This code is for the wrong RFID. ( above) That is setting the red led and long beep.

if (Code6Correct == 1 &&RightCard == 1) // If the PinCode is the correct and you did enter this within 7 seconds

{

RightPinCode = 1; // VariableRightPinCode will be set to 1

digitalWrite (GreenLed, HIGH); // Make a nice beep for the correct Pincode and set the Green LED high for 1 second

delay (150); //

digitalWrite (Buzzer, HIGH); //

delay (150); //

digitalWrite (Buzzer, LOW); //

delay (50); //

digitalWrite (Buzzer, HIGH); //

delay (150); //

digitalWrite (Buzzer, LOW); //

delay (500); //

digitalWrite (GreenLed, LOW); //

Serial.println("Correct PinCode"); // Print the text "Correct PinCode" to the serial monitor

Reset = 1;

The above code is executed when the pincode entered is correct. (above)

if ((Code6Correct == 0) && (PinCodeCounter>= 6) && (RightCard == 1)) // If you offered a valid Mifare Card and the Pincode is not correct and you already entered 6 digitkeys

{

WrongPinCode = 1; // Variable WrongPinCode will be set to 1

Serial.println("Incorrect pincode, please try again.");

// Print the text "WrongKey" to the serial monitor

Reset = 1; // Reset the loop

}

If incorrect pincode is entered. (above)

if ((WrongPinCode == 1) || (millis()- Timer > 7000 &&RightCard == 1)) // If you offered a valid Mifare Card and you entered a wrong PinCode or the timer (7 seconds) expires

{

digitalWrite (Buzzer, HIGH); // Make a long beep and set the Red LED HIGH for the wrong code or timer expiry

digitalWrite (RedLed, HIGH); //

delay(1500); //

digitalWrite (Buzzer, LOW); //

digitalWrite (RedLed, LOW); //

Serial.println("WrongCode or Timer expired"); // Print the text "WrongCode or Timer expired" to the serial monitor

Reset = 1; // Reset the loop

}

If the timer becomes more than 7 seconds or the user does not enter the pincode within 7 seconds. (above)

char KeyDigit = keypad.getKey(); // Get the DigitKey from the keypad

if ((RightCard == 1) && // If you offered a valid Mifare Card and any DigitKey on the keypad is pressed

((KeyDigit == '1') ||

(KeyDigit == '2') ||

(KeyDigit == '3') ||

(KeyDigit == '4') ||

(KeyDigit == '5') ||

(KeyDigit == '6') ||

(KeyDigit == '7') ||

(KeyDigit == '8') ||

(KeyDigit == '9') ||

(KeyDigit == '0') ||

(KeyDigit == '\*') ||

(KeyDigit == '#')))

{

PinCodeCounter++; // PinCodeCounter value +1 for every press on any Digitkey on the keypad

digitalWrite (Buzzer, HIGH); // Make a short beep for a DigitKey press on the keypad

delay (50); //

digitalWrite (Buzzer, LOW); //

}

Initialization of keypad buttons. (above)

if ((KeyDigit == Code1MifareCard1) && (RightCard == 1) && (Code1Correct == 0) && (MifareCard1 == 1))

{

Code1Correct = 1; // Variable Code1Correct is set to 1

return; // Return to begin loop

} (pto)

if ((KeyDigit == Code2MifareCard1) && (Code1Correct == 1) && (Code2Correct == 0) && (MifareCard1 == 1))

Code2Correct = 1; // Variable Code2Correct is set to 1

return; // Return to begin loop

}

if ((KeyDigit == Code3MifareCard1) && (Code2Correct == 1) && (Code3Correct == 0) && (MifareCard1 == 1))

{

Code3Correct = 1; // Variable Code3Correct is set to 1

return; // Return to begin loop

}

if ((KeyDigit == Code4MifareCard1) && (Code3Correct == 1) && (Code4Correct == 0) && (MifareCard1 == 1))

{

Code4Correct = 1; // Variable Code4Correct is set to 1

return; // Return to begin loop

} if ((KeyDigit == Code5MifareCard1) && (Code4Correct == 1) && (Code5Correct == 0) && (MifareCard1 == 1))

{

Code5Correct = 1; // Variable Code5Correct is set to 1

return; // Return to begin loop

}

if ((KeyDigit == Code6MifareCard1) && (Code5Correct == 1) && (Code6Correct == 0) && (MifareCard1 == 1))

{

Code6Correct = 1; // Variable Code6Correct is set to 1

return; // Return to begin loop

}

In the above block of code the keypad is taking the input of the user verifying for the correct pincode.

**OTHER APPLICATIONS**

In this scheme, many other applications are present.

Other projects that can be made are:

* RFID based attendance system

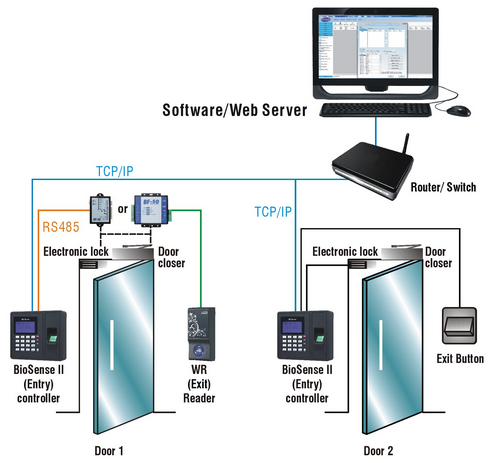
This system is used in many colleges and schools. Where rfid is used to keep track of the attendance of teachers and students. So, when one person scans his/her RFID her attendance is recorded for that day. Reducing the task of manually taking attendance which could be time consuming.

* RFID based home automation system

This system scans the RFID of the person and then the electrical appliances start working in the room. This system is mostly widely used in hotels. Where you put in your room key into the provided slot and then it supplies electricity to the room. This helps in saving electricity. It also provides security.

**REAL LIFE APPLICATIONS OF RFID BASED ACCESS CONTROL SYSTEM**

The most common use of RFID is the access control system. Keeping away unauthorized personnel away and maintaining security in a more safe and efficient manner. These are most widely used in buildings that hold very sensitive information like banks, labs product based companies. It helps to control plagiarism and theft. It can also be used as an identification process reducing the burden of manually checking and thus make work faster. The figure below is a more modern and wireless way of an access control system



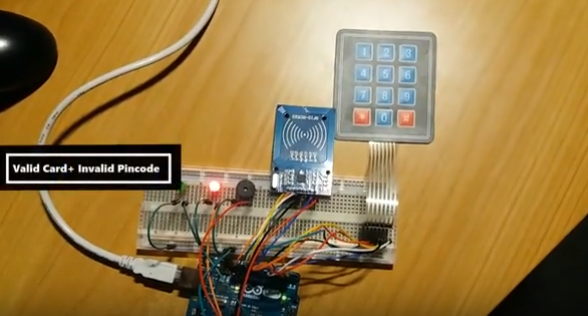
**MERITS**

* Easy to use
* Lucrative
* More safe
* Provides security
* Variety of uses
* Fairly small in size
* Automated
* Longevity is high
* Rugged
* Highly reliable
* Fast scanning
* Easy inventory management
* Make work less

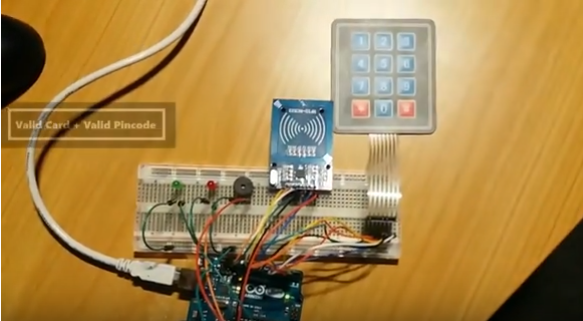
**DEMERITS**

* Expensive
* Each UID has to scanned which is tedious process
* Can be hacked
* Security can be breached
* Personal or sensitive information could be leaked due to errors
* Humanitarian questioning due to implantation in animals
* Cannot be completely dependent on it.
* Implementation in questioning due to privacy.

**PICTURES**



For valid card and invalid pincode.



For valid card and valid pincode.