**CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING & TECHNOLOGY**

**COMPUTER SCIENCE ENGINEERING**

**Project Report**

**On**

***SMART DOOR LOCK***

**Course Name – Internet of Things**

**Course Code – CS201**

**Submitted By: Submitted To:**

**SAHIL GOYAL (1910990023) MRS. SONAM AGGARWAL**

**YASHASVI KESARWANI (1910990009)**

**PARV DEWAN (1910990029)**

**RAGHAV SHARMA (1910990033)**

**YASH GOEL (1910990037)**

**CHITKARA UNIVERSITY**

NH-7, CHANDIGARH PATIALA NATIONAL HIGHWAY

RAJPURA, PUNJAB 140401.

***INTRODUCTION***

A smart lock is an electronic and mechanical locking device that opens wirelessly with an authorized users’ authentication,

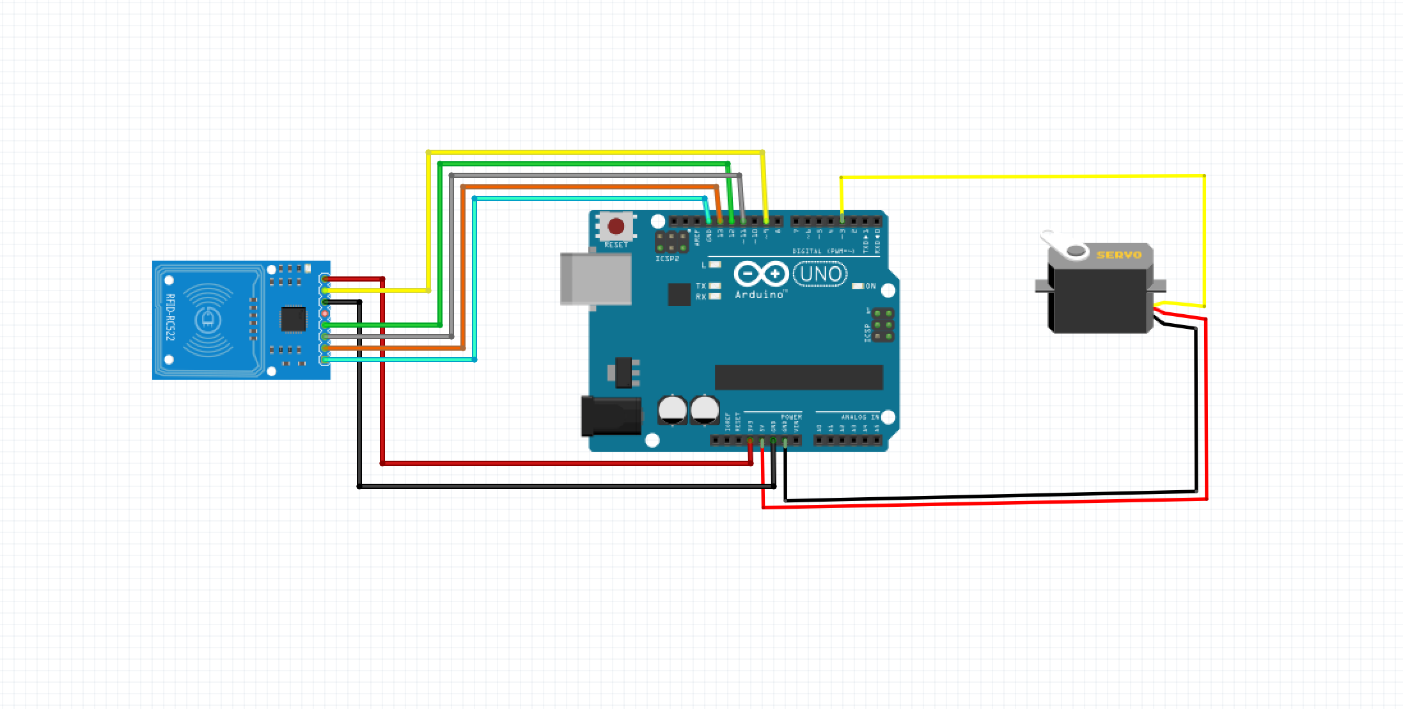
In a smart home, smart locks allow a homeowner to enter their home or provide others access without requiring a traditional key. Instead, the user uses their smart phone or a key fob to wirelessly verify and mechanically unlock the door. Smart locks are an extension of home automation into home security.  As a connected device, smart locks can be considered a part of the internet of things (IoT).

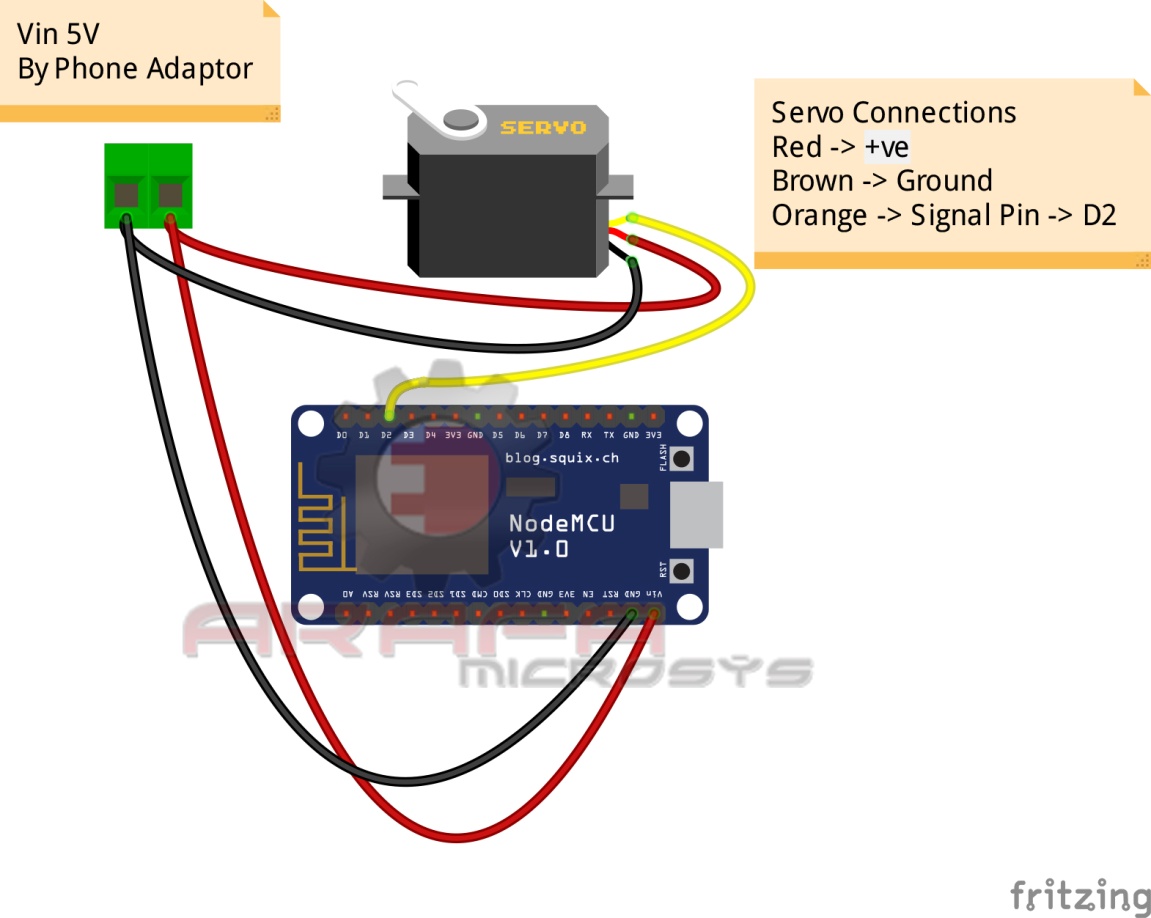
Many smart lock systems use mobile apps or websites to allow homeowners to grant access to third parties by sending a virtual key. Virtual keys can be sent by SMS text message or email, enabling access to guests or service personnel.

Once received, these encrypted digital keys allow access to the smart lock for a preset period of time. On top of regulating access, many smart locks log access, providing the means to monitor use of a given secured door. Some smart locks feature a camera, which provides a picture of those accessing the door and makes for an easily referenced photo log.

Potential vulnerabilities in smart locks do create security concerns. Security advisors recommend that smart locks should not necessarily be thought of as more secure than a conventional lock and key and recommend that users think carefully about settings and options.

***CIRCUIT DIAGRAM***





***WORKING OF PROJECT***

Smart door lock is operated by RFID card and using blink app wirelessly. When we show RFID card to the sensor then the servo rotates and allow the door to open and automatically after 7 seconds it reaches to its original position. We can also control it by pressing on button in the blink app to rotate the servo and press the off button to bring it back to its original position (to lock the door).Here the components used in the project are nodeMCU , RFID sensor ,arduino UNO, jumper wires ,servo motors and jumper wires. By the use of blink app we connect nodeMCU to the server of the blynk app and we can operate our smart door lock from our smartphone from anywhere.

***SOFTWARE PROGRAM OF THE PROJECT***

//Arafa Microsys

//www.youtube.com/arafamicrosystems

//www.fb.com/arafa.microsys

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DNSServer.h>

#include <ESP8266WebServer.h>

#include <WiFiManager.h> //https://github.com/tzapu/WiFiManager

#include <EEPROM.h>

#include <Servo.h>

Servo myservo;

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "YQV3sYJGGdDfcSDhHtEz5XA\_LE4f-A\_l";

void setup()

{

EEPROM.begin(512);

Serial.begin(9600);

myservo.attach(4);

myservo.write(180);

WiFiManager wifiManager;

//wifiManager.resetSettings(); //Uncomment this to wipe WiFi settings from EEPROM on boot. Comment out and recompile/upload after 1 boot cycle.

wifiManager.autoConnect("NodeMCU");

//if you get here you have connected to the WiFi

Serial.println("connected...:)");

pinMode(LED\_BUILTIN, OUTPUT); // Initialize the BUILTIN\_LED pin as an output, I like blinkies.

Blynk.begin(auth, WiFi.SSID().c\_str(), WiFi.psk().c\_str());

}

BLYNK\_WRITE(V2)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

Serial.print("V1 Slider value is: ");

Serial.println(pinValue);

Blynk.virtualWrite(V0, "Door State");

if(pinValue)

{

Blynk.virtualWrite(V1, "Open");

servo();

}else

{

Blynk.virtualWrite(V1, "Closed");

myservo.write(180);

}

}

void loop()

{

Blynk.run(); // Commented out for a short time.

}

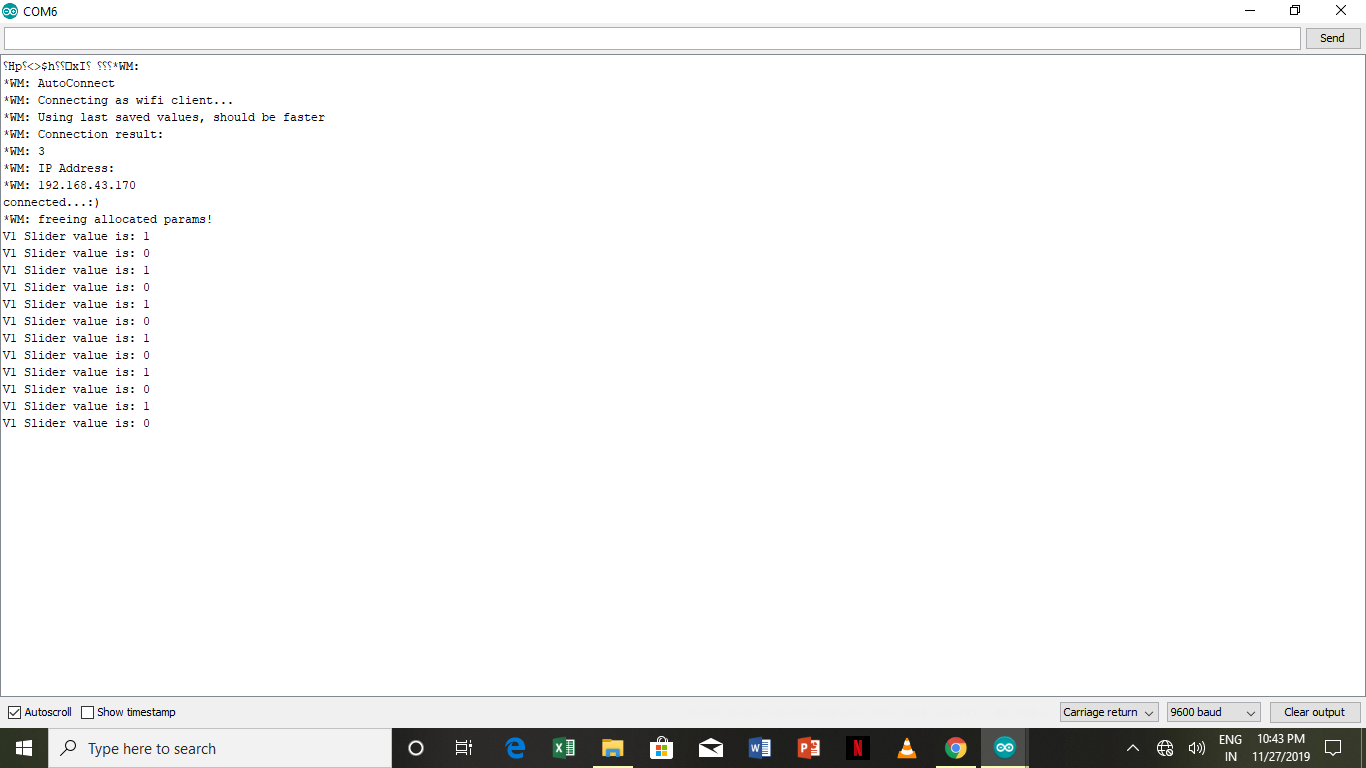
void servo()

{

myservo.write(90);

}

***SCREENSHOT OF SERIAL MONITOR***

******