HOME AUTOMATION FOR A SMART LIFE

Introduction

- Home automation is the ability to control domestic appliances by electronically controlled, internet-connected systems. It may include setting complex heating and lighting systems in advance and setting alarms and home security control connected by a central hub and remote-controlled by a mobile app.
- The data is collected and used to monitor, control, and transfer information to other devices via the internet. This allows specific actions to be automatically activated whenever certain situations arise.

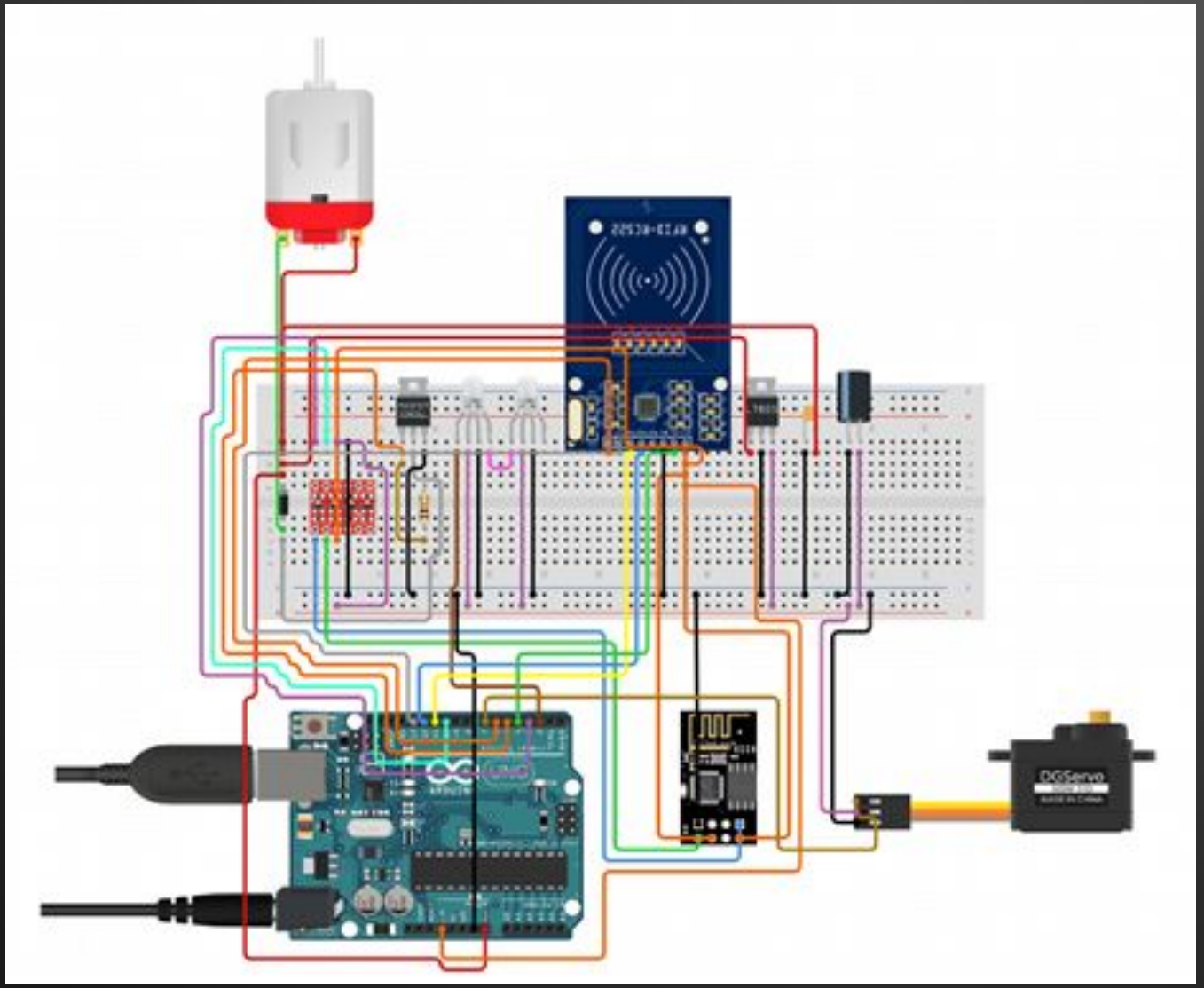
Problem Statement

To create a system for handling and controlling home appliances by using a microcontroller. A system enables remote control of lighting and other appliances through complex micro-controller or computer-based networks with various intelligence and automation degrees. Home automation provides security, energy efficiency, and ease of use. It also provides a remote interface to home appliances to provide control and monitoring.

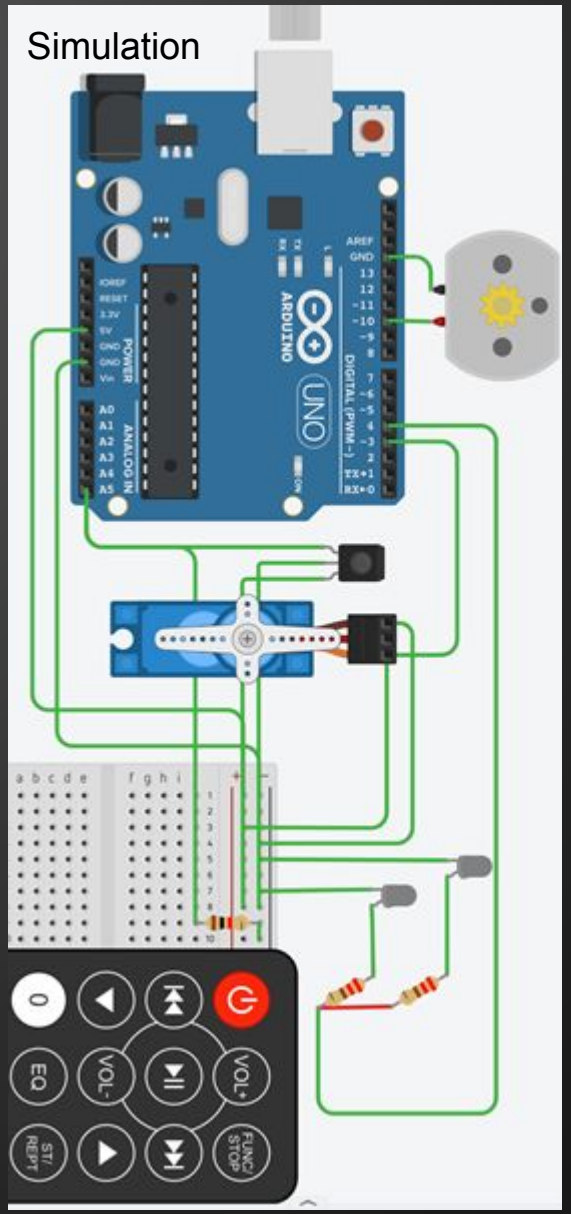
Solution Provided

Home Automation using smart sensors and IoT. This project depicts how RFID and Wi-Fi modules can be used for home automation applications. An RFID entry tag is used as an access card to enter the room, and once the sensor senses it as a correct tag, the door is opened using a door lock system controlled by a servo motor. The lights and fans are turned on (LEDs and Small fan fixed to DC Motor is used for this purpose), they can also be controlled with an app and google assistant using a Wi-Fi module esp 01 integrated with the project. When the person leaves the room, if the RFID exit tag is shown at the exit, the lights and fans will automatically get turned off, and the door gets locked. By this, we can save energy and automate the process. This improves the power management, safety and security of the homes.

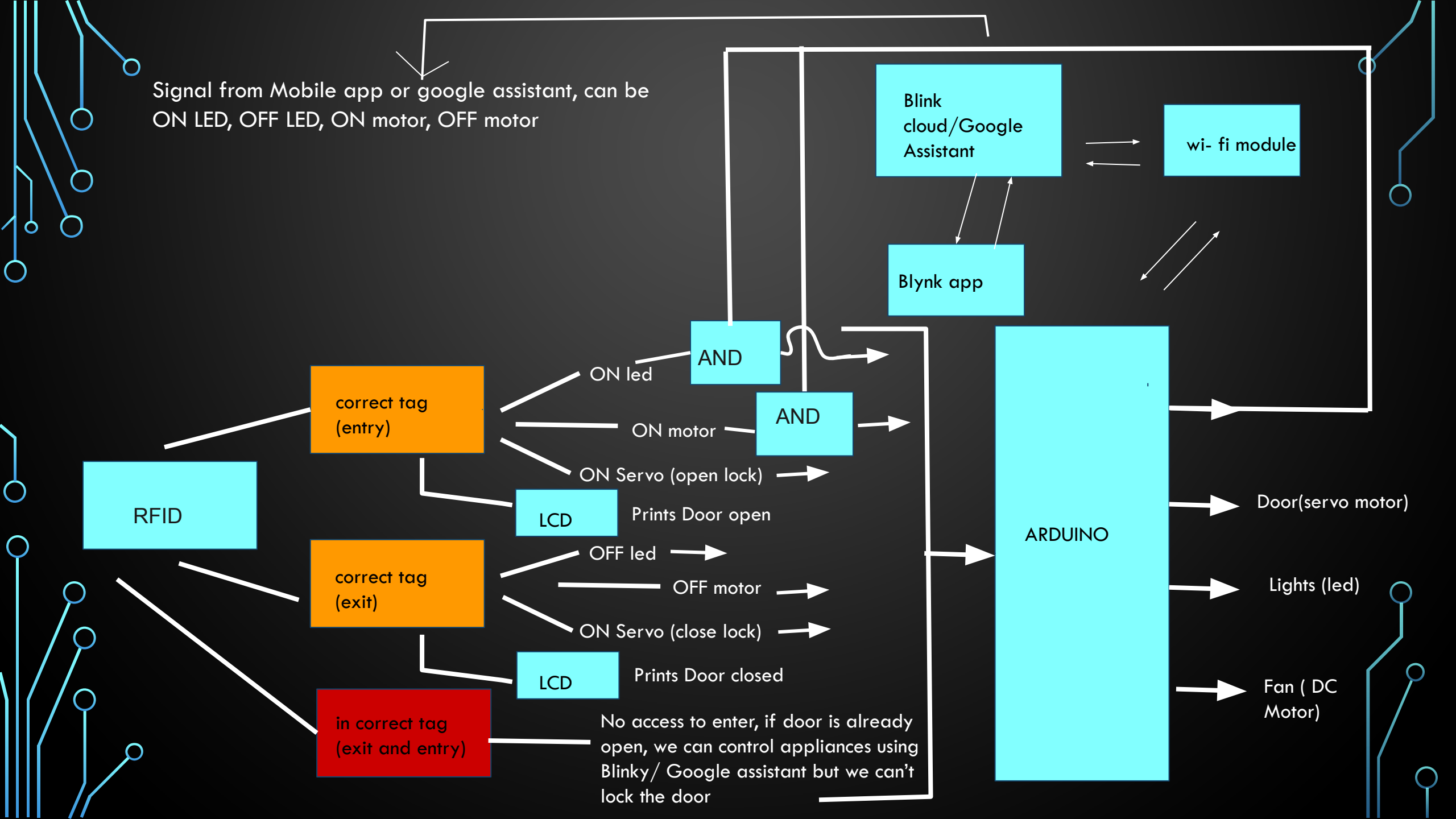
Circuit Diagram



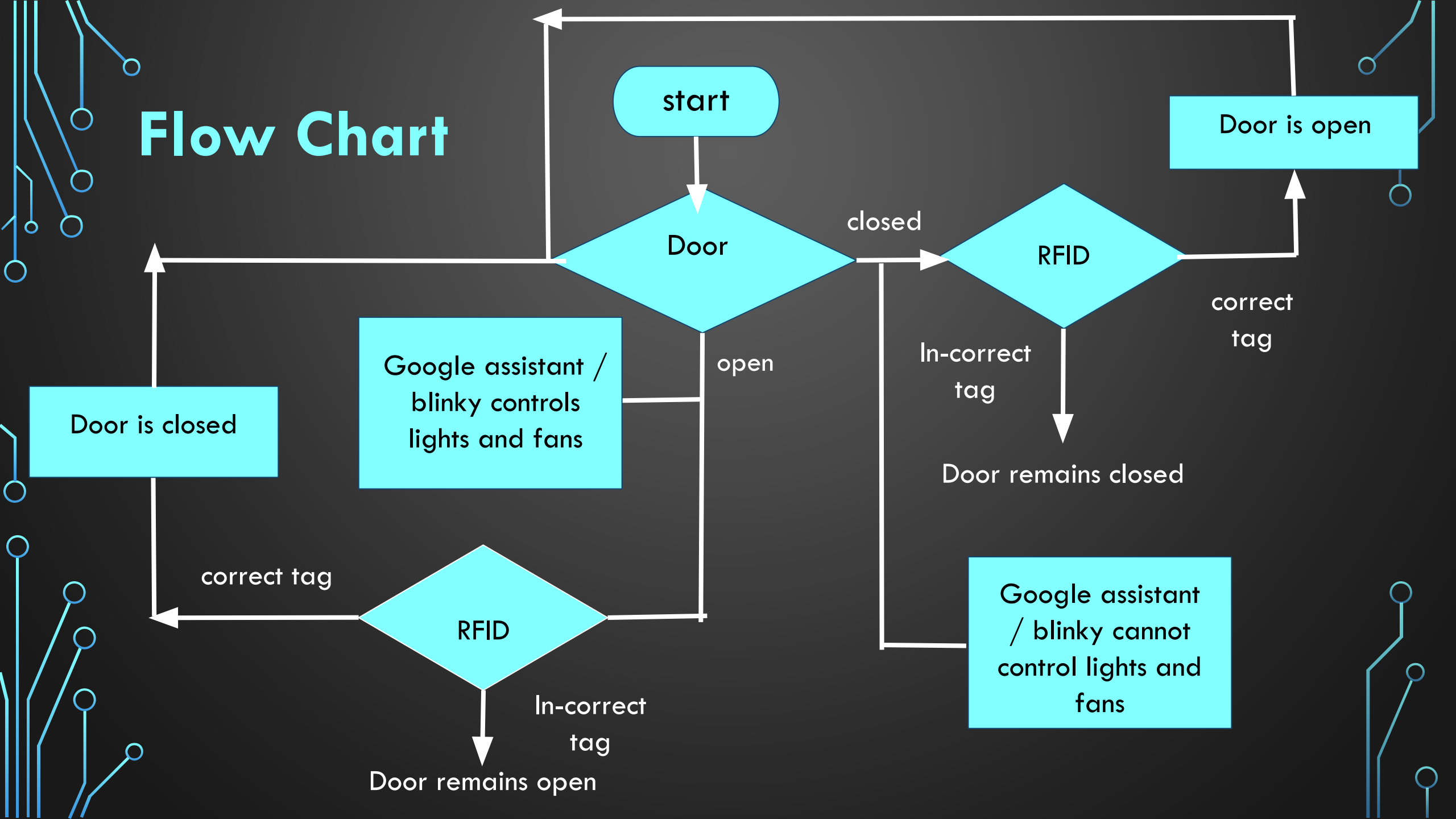
Simulation



Block Diagram


Signal from Mobile app or google assistant, can be ON LED, OFF LED, ON motor, OFF motor
Blink cloud/Google Assistant
wi- fi module
Blynk app
AND
AND
ON led
ON motor
ON Servo (open lock)
correct tag (entry)
RFID
LCD
Prints Door open
OFF led
OFF motor
ON Servo (close lock)
correct tag (exit)
LCD
Prints Door closed
in correct tag (exit and entry)
No access to enter, if door is already open, we can control appliances using Blinky/ Google assistant but we can't lock the door
ARDUINO
Door(servo motor)
Lights (led)
Fan (DC Motor)

Flow Chart

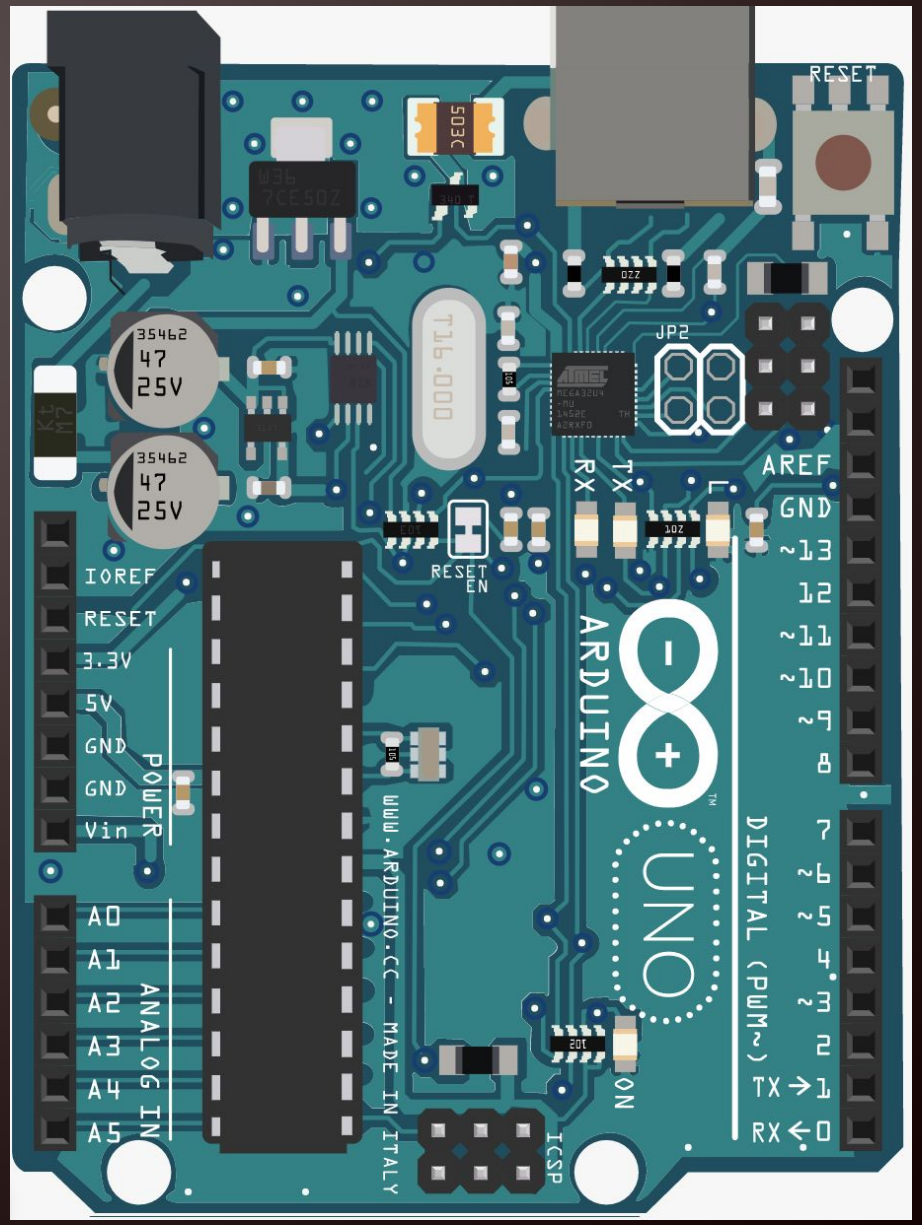

start
Door
closed
RFID
Door is open
correct tag
In-correct tag
Door remains closed
open
Google assistant / blinky controls lights and fans
Door is closed
correct tag
RFID
In-correct tag
Door remains open
Google assistant / blinky cannot control lights and fans

COMPONENTS USED

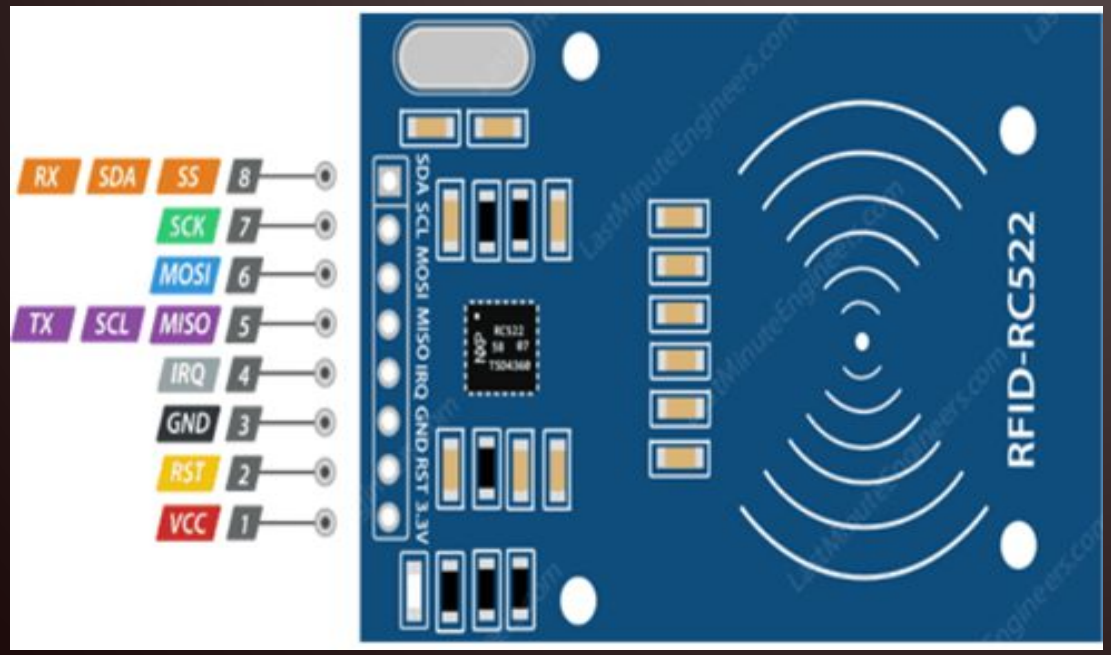
- Arduino
- RFID card and reader
- Wi-fi module
- Servo motor
- DC motor
- LCD display
- PCB or bread board
- LEDs and resistors
- Battery and connectors, wires

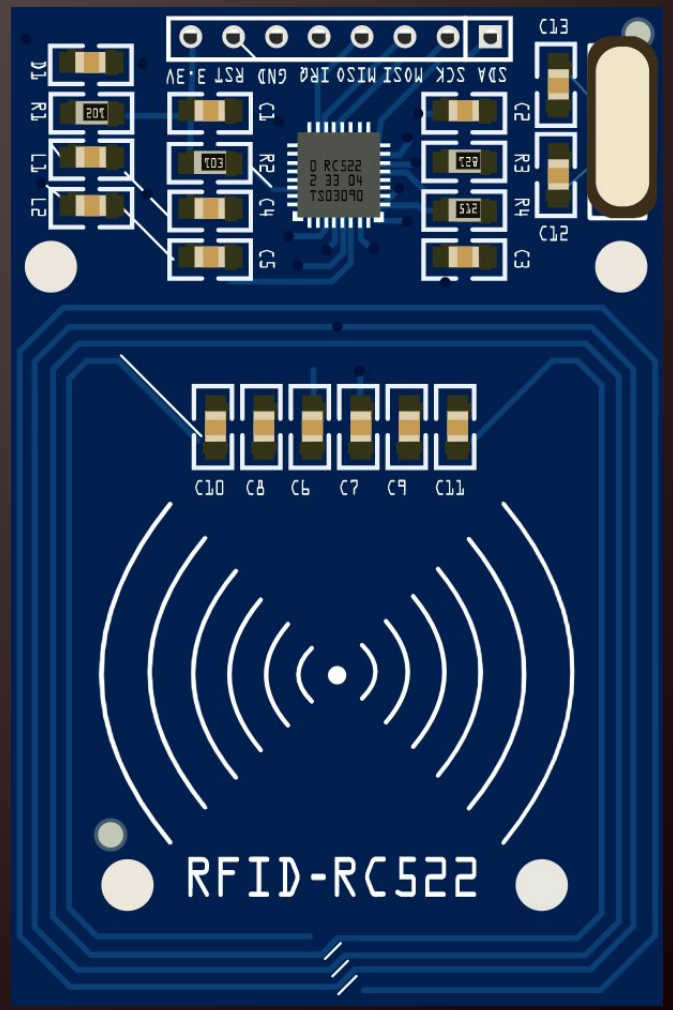
Arduino

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.



RFID MODULE

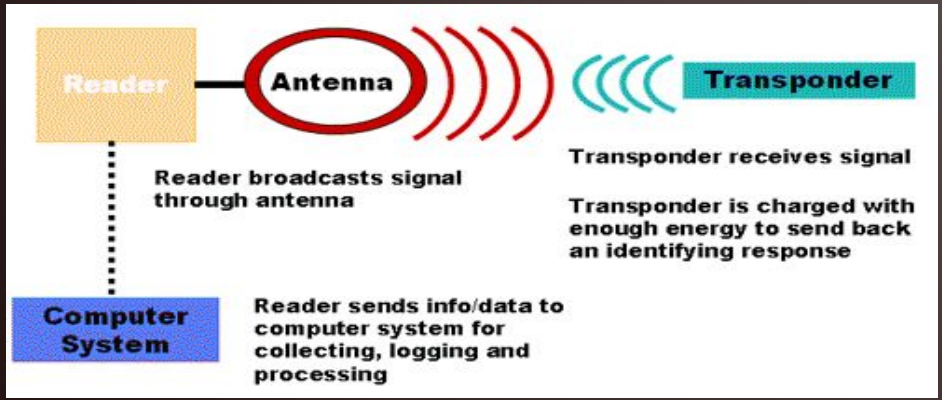

RX
SDA
SS
8
SCK
7
MOSI
6
TX
SCL
MISO
5
IRQ
4
GND
3
RST
2
VCC
1
SDA SCL MOSI MISO IRQ GND RST 3.3V
RFID-RC522

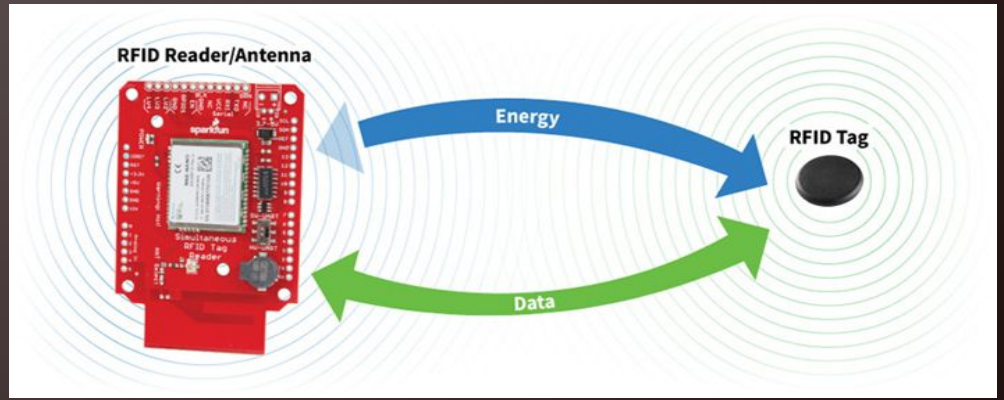

SDA SCK MOSI MISO IRQ GND RST 3.3V
C13
C12
C10 C8 C6 C7 C9 C11
RFID-RC522

Pin number	Pin name	Pin function
1.	**VCC**	2.5 to 3.3 volts input, power supply
2.	**RST**	when low - hard power-down, when high - reset
3.	**GND**	GND
4.	**IRQ**	an interrupt pin, alerts the microcontroller when RFID tag comes into its vicinity.
5.	**MISO / SCL / Tx**	Master-In-Slave-Out when SPI interface is enabled serial clock when I2C interface is enabled serial data output when UART interface is enabled.
6.	**MOSI (Master Out Slave In)**	SPI input
7.	**SCK (Serial Clock)**	accepts clock pulses provided by the SPI bus Master
8.	**SS / SDA / Rx pin**	Signal input when SPI interface is enabled serial data when I2C interface is enabled and serial data input when UART interface is enabled.

RFID system

- Radio Frequency Identification





The RFID reader generates a high frequency electromagnetic field and when the tag comes near it, a voltage is induced in tags antenna coil due to induction. This induced voltage acts as power for the tag. The tag in return converts the signal in power and responds to the reader.

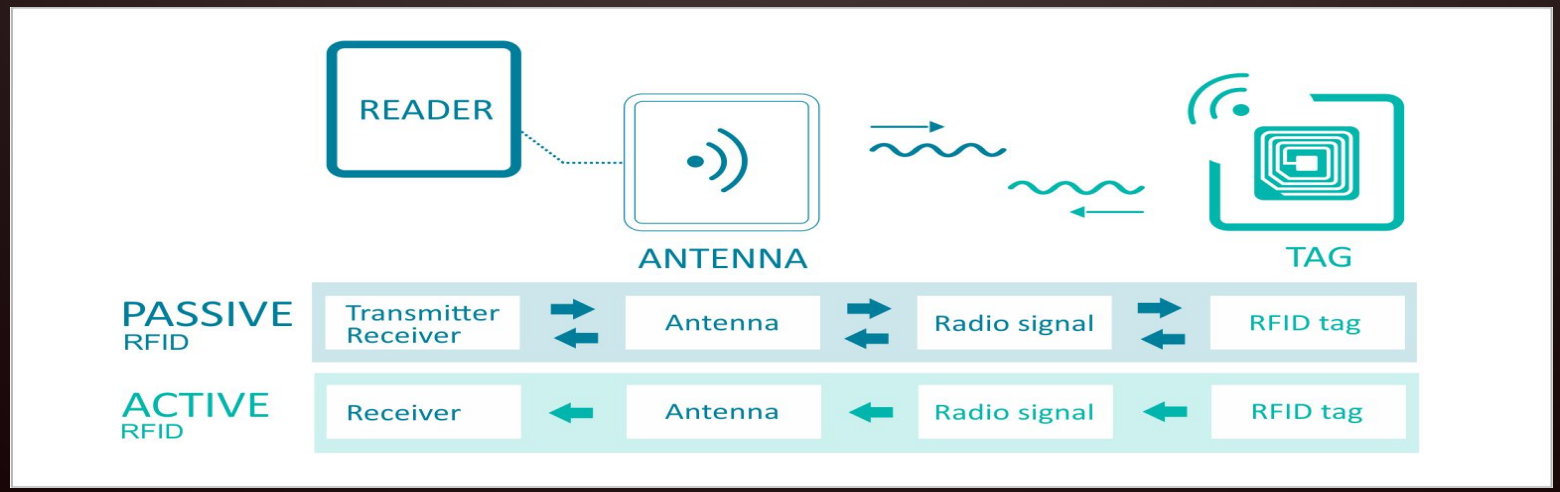
RFID FREQUENCIES

Depending on the **frequency band** they operate at we have different RFID systems:

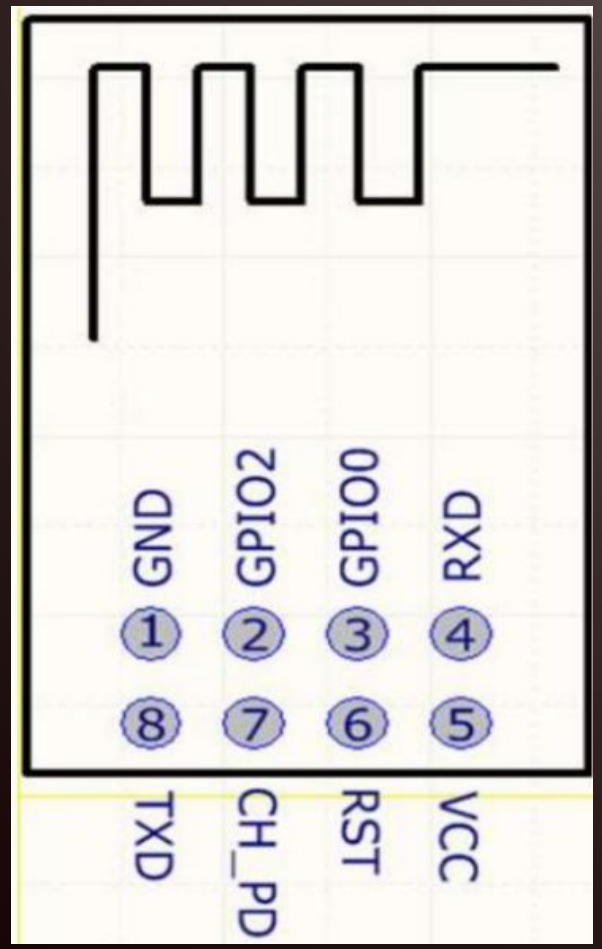
- Low frequency (**LF**) - 30 KHz to 300 KHz range, read range upto 10 cm
- High frequency (**HF / NFC**) - 3 MHz to 30 MHz range, read range 10 cm to 1m
- Ultra-high frequency (**UHF**) - 300 MHz to 3 GHz range, read range upto 12 m

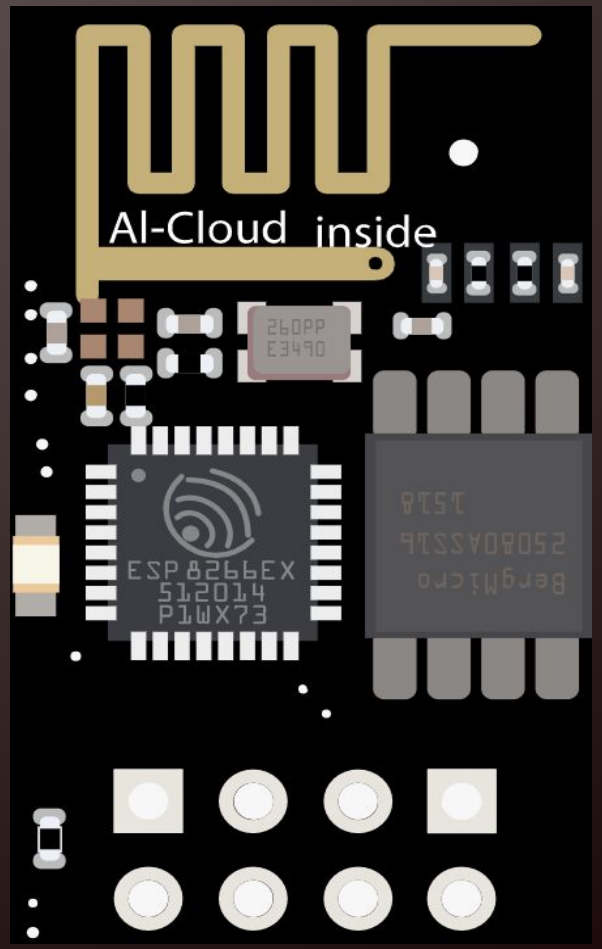
LF and HF systems use Inductive coupling, load modulation

UF systems use Far field coupling, backscatter modulation technique



Wi-Fi Module


GND
GPIO2
GPIO0
RXD
1
2
3
4
8
7
6
5
TXD
CH_PD
RST
VCC


AI-Cloud inside

Pin number	Pin name	Pin function
1.	**GND**	GND
2.	**GPIO2**	GPIO, internal pull - up
3.	**GPIO0**	GPIO, internal pull - up
4.	**RXD / GPIO3**	UART0, Data received pin
5.	**VCC**	3.3 power supply (VDD)
6.	**RST**	External reset pin, active low Can loft or external MCU
7.	**CH_PD**	Chip enable pin, active high
8.	**TXD / GPIO1**	UART0, data send pin RXD

Wi-fi Module

This module is a self-contained **SOC** (System On a Chip) that doesn't necessarily need a microcontroller.

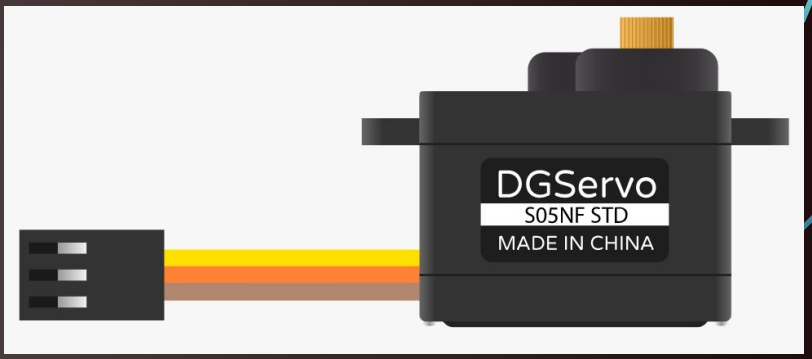
The **ESP8266 ESP-01**, comes with a pre-installed **AT firmware, which** is compatible with the **Arduino IDE,** so we are using this firmware.

The ESP8266 ESP-01 module has three operation modes:

1. **Access Point (AP)** (can create hotspot)
2. **Station (STA)** (can connect to wifi)
3. **Both**

GPIO – 0	GPIO – 2	Mode	Used for
High	High	Flash Mode	Run the program that is already uploaded to the module
low	High	UART Mode	Programming mode- to program using Arduino or any serial communication

Servo Motors

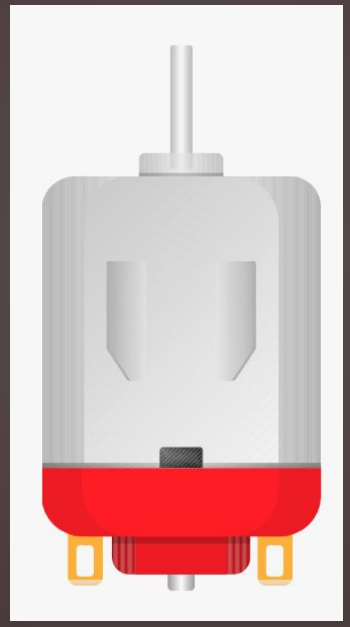


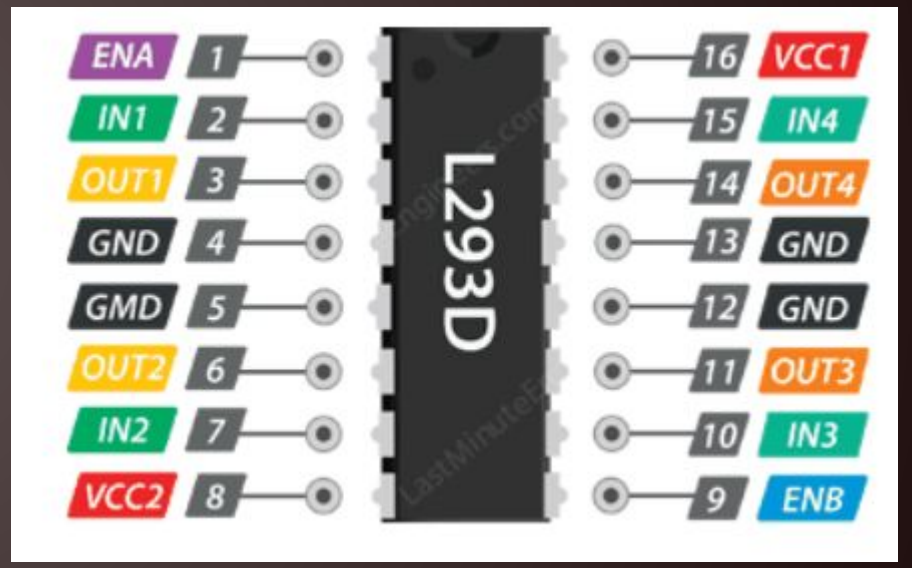
1. The servo motor has a female connector with three pins. The darkest or even black one is usually the ground. Connect this to the Arduino GND.
2. Connect the power cable that in all standards should be red to 5V on the Arduino.
3. Connect the remaining line on the servo connector to a digital pin on the Arduino.

- Servo motor works on **PWM** principle.
- Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears.**
- High speed force of DC motor is converted into torque by Gears.
- The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

DC Motor

Electric motors turn electricity into motion by exploiting electromagnetic induction. The motor features a permanent horseshoe magnet (called the stator because it's fixed in place) and an turning coil of wire called an armature (or rotor, because it rotates).

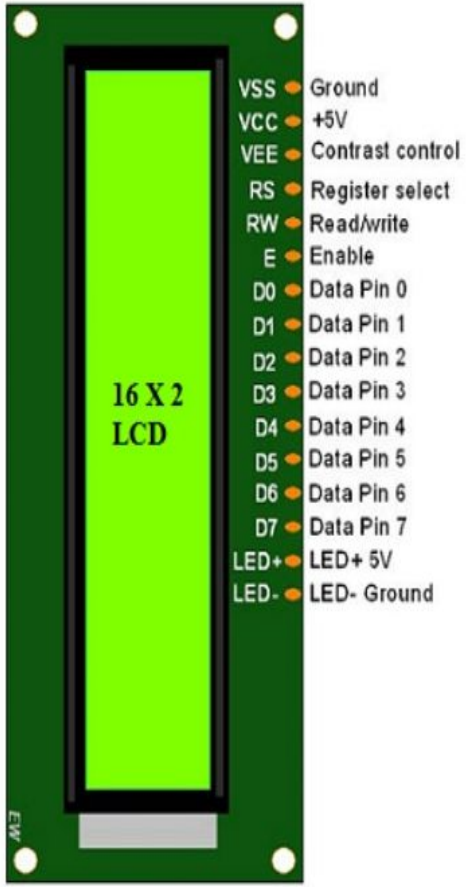




IN1	IN2	Spinning Direction
Low(0)	Low(0)	Motor OFF
High(1)	Low(0)	Forward
Low(0)	High(1)	Backward
High(1)	High(1)	Motor OFF

LCD

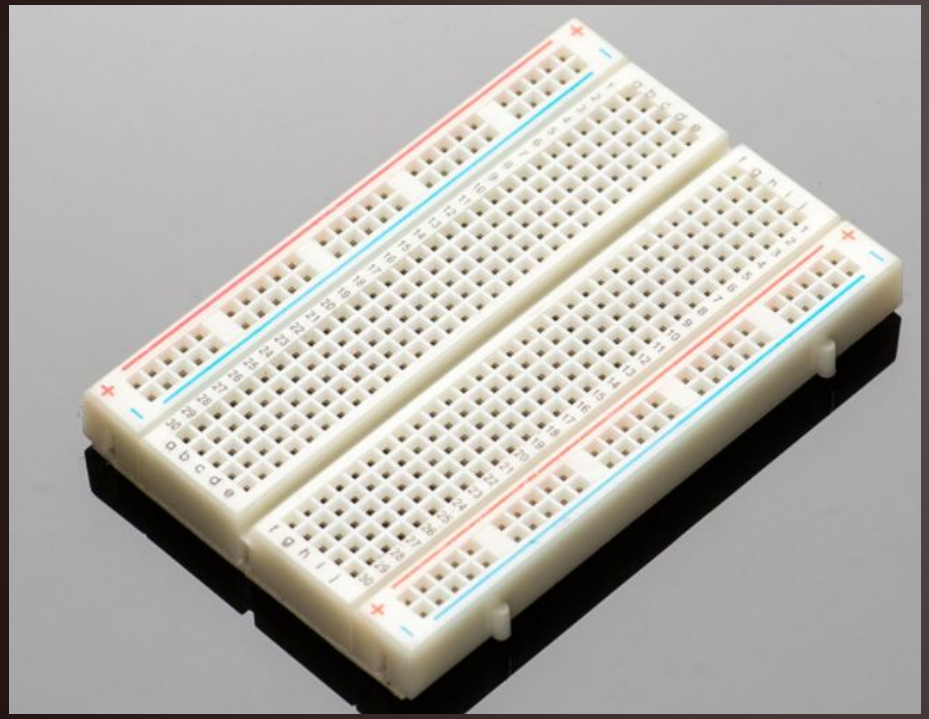
Pin No:	Pin Name:	Description
1	Vss (Ground)	Ground pin connected to system ground
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shift between command/data register
5	Read/write	Used to read or write data. Normally grounded to write data to LCD
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement
7 - 14	Data pin 0 - 7	Data pins 0 to 7 forms a 8-bit data line, can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.
15	LED positive	Backlight led pin positive terminal
16	LED negative	Backlight led pin negative terminal



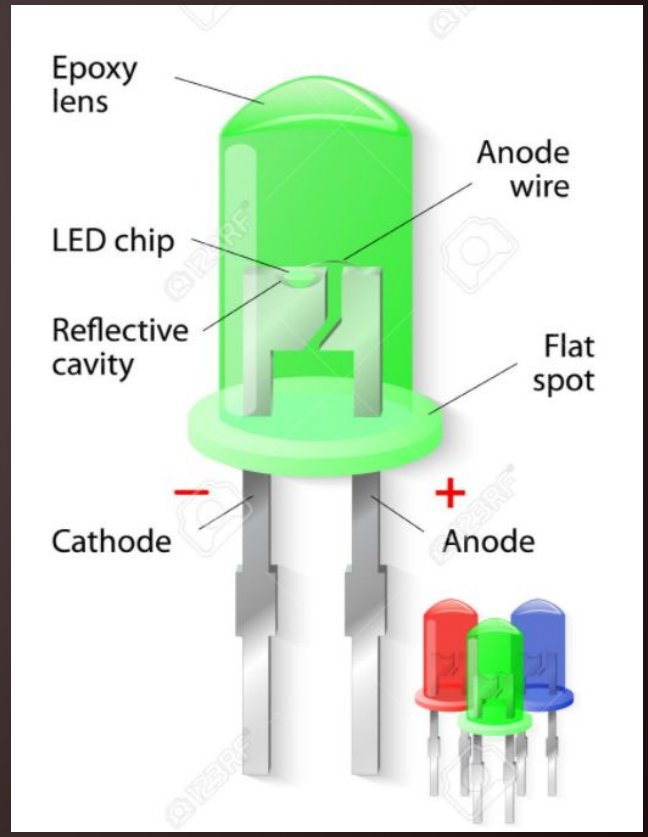
LCD-16×2-pin-diagram

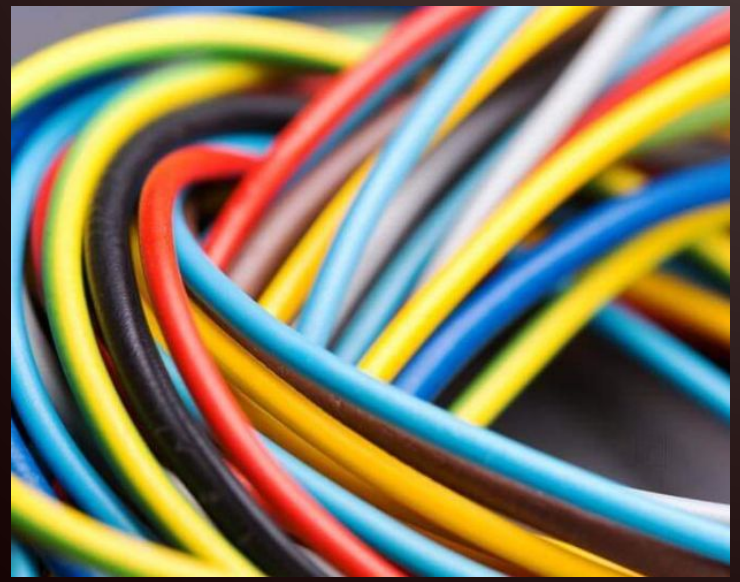
Bread board, LEDs, resistors, Battery and wires

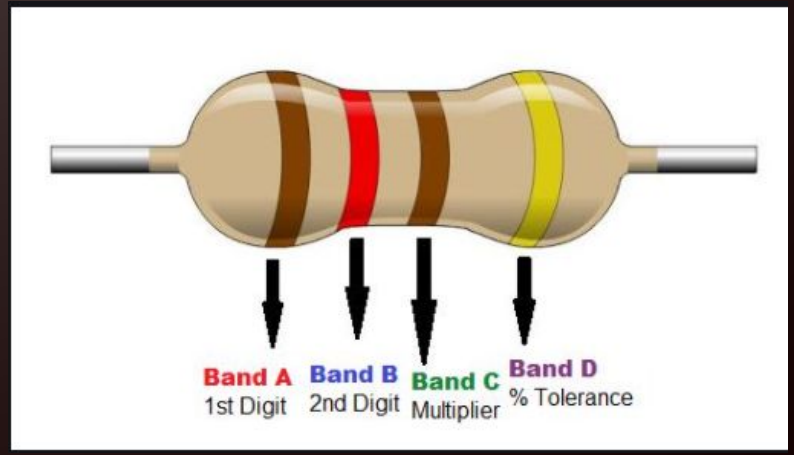
- A breadboard is a construction base for prototyping of electronics.
- A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it.
- A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.
- A battery is a device consisting of one or more electrochemical cells with external connections for powering electrical devices.
- A wire is a single usually cylindrical, flexible strand or rod of metal. Wires are used to bear mechanical loads or electricity and telecommunications signals.




HW
Hi-Watt
GENERAL PURPOSE


Epoxy lens
Anode wire
LED chip
Reflective cavity
Flat spot
Cathode
Anode




Band A
1st Digit
Band B
2nd Digit
Band C
Multiplier
Band D
% Tolerance

Software Code

ESP	Arduino
TX	TX pin 0
RX	RX pin 1
VCC	3.3V
GND	GNH

RFID	Arduino
SDA	10
SCK	13
MOSI	11
MISO	12
IRQ	not connected
GND	GND
RST	not connected
VCC	3.3V

/*

* Firstly, A Brief About our Project

* A Home Automation System For a Smart Life

* What do we do?

* 1) A Door Lock System Based on RFID

* 2) When Correct Entry Card is shown Lock Opens,Lights and fans gets turned on

* 3) Don’t want lights or fans to run? No problemo. Control With the app Blinky

* 4) Lazy? Use Google Assistant

* 5) Finally While Exiting show Exit Card All lights, fans are turned off, Door is Locked

*

* Pin Connections:
*
* ESP 01 -Arduino Uno
* TX -TX(1)
* RX -RX(0)
* 3.3V -3.3V
* GND -GND
*
* RFID -Arduino Uno
* SDA-10
* SCK-13
* MOS1-11
* MIS0-12
* IRQ-Not Connected
* GND-gnd
* RST-A0(Not Connected though)
* 3.3V-3.3V
*

```
*
*  DC Motor- Arduino Uno
*   enA - A3
*   in1 -A5
*   in2 -A4
*
*   Servo- Arduino Uno
*   Attach pin -A2
*   5v-5v
*   gnd-gnd
*
*   pin9- LED on Pin when RFID on
*   Pin7-LED Communication with Blynk
*   pin6- Motor Communication with Blynk
*   PinA0-Wifi with Blynk
*
*   And of Pin7,pin9 =>LED
*   And of Pin6,A4=>Motor Driver pin
*
```

* Basic Steps Included are

* 1) Libraries Installation

* 2) Interfacing Pins

* 3) Pin Name Declarations

* 4) Authorization Key For Blinky

* Void Setup:

* 4) Begin Communications

* 5) Initial Positions/Status/Configurations

* 6) Message To be printed On Serial Monitor

* Void Loop:

* 7) Blink Reading

* 8) Tag there or not , Same or different

* 9) If Entering Correct Card => On Lights,Fans,Open Lock

* 10) If Exiting Correct Card => Off All

* 11) Else Unauthorized

* Functions:

* 12) On Motors

* 13) Off Motors

*/

```
#include <SPI.h>                    //Library For RFID
#include <MFRC522.h>                //Library For RFID
#include <Servo.h>                  //Library For Servo
#include<LiquidCrystal.h>           //Library For LCD
#include <SoftwareSerial.h>         //Library For Serial Communication With Wifimodule+PC
#include <BlynkSimpleStream.h>      //Library For Blinky

MFRC522 mfrc522(10,A2);             //Interfacing of RFID  -> SDA,RESET
#define BLYNK_PRINT DebugSerial     //The Interfacing Data Type of BLYNK_PRINT
SoftwareSerial DebugSerial(0,1);    //Interfacing of Blynk -> RX,TX
LiquidCrystal lcd(6,7,5,4,3,2);     //register select,enable,db4,db5,db6,db7
Servo myServo;                      //Servo Variable
```

```
int light=9;

int enA = A3;

int in1 = A5;

int in2 = A4;

int Lock_Open_Angle=40;

int Lock_Close_Angle=145;

char auth[] = "xhTgiE44-fo6YUGRGSjAL6s3ZiBj0Xfa";
```

```
void setup()
{
  Serial.begin(9600);                  //Communication With Serial Monitor
  SPI.begin();                         //begin
  mfrc522.PCD_Init();                  //Initialize RFID module
  DebugSerial.begin(9600);             //Communication with Wifi Module+pc
  Blynk.begin(Serial, auth);           //Communication With Blynk
  lcd.begin(16,2);                     // Communication with LCD

  myServo.attach(A0);                   // Servo Pin Initialising

  pinMode(9,OUTPUT);                    //LED        -Lights in ROOM
  pinMode(enA, OUTPUT);                 //DC MOTOR    -PWM speed Pin
  pinMode(in1, OUTPUT);                 //DC MOTOR    -One pin of H-Bridge gate
  pinMode(in2, OUTPUT);                 //DC MOTOR    -Second pin of the Pair of the Bridge
  lcd.clear();

  digitalWrite(light,LOW);              //Initial Lights Status
  turn_off_motors();                    //Initial Motor Status
  myServo.write(Lock_Close_Angle);          //Initial Servo Position
}
```

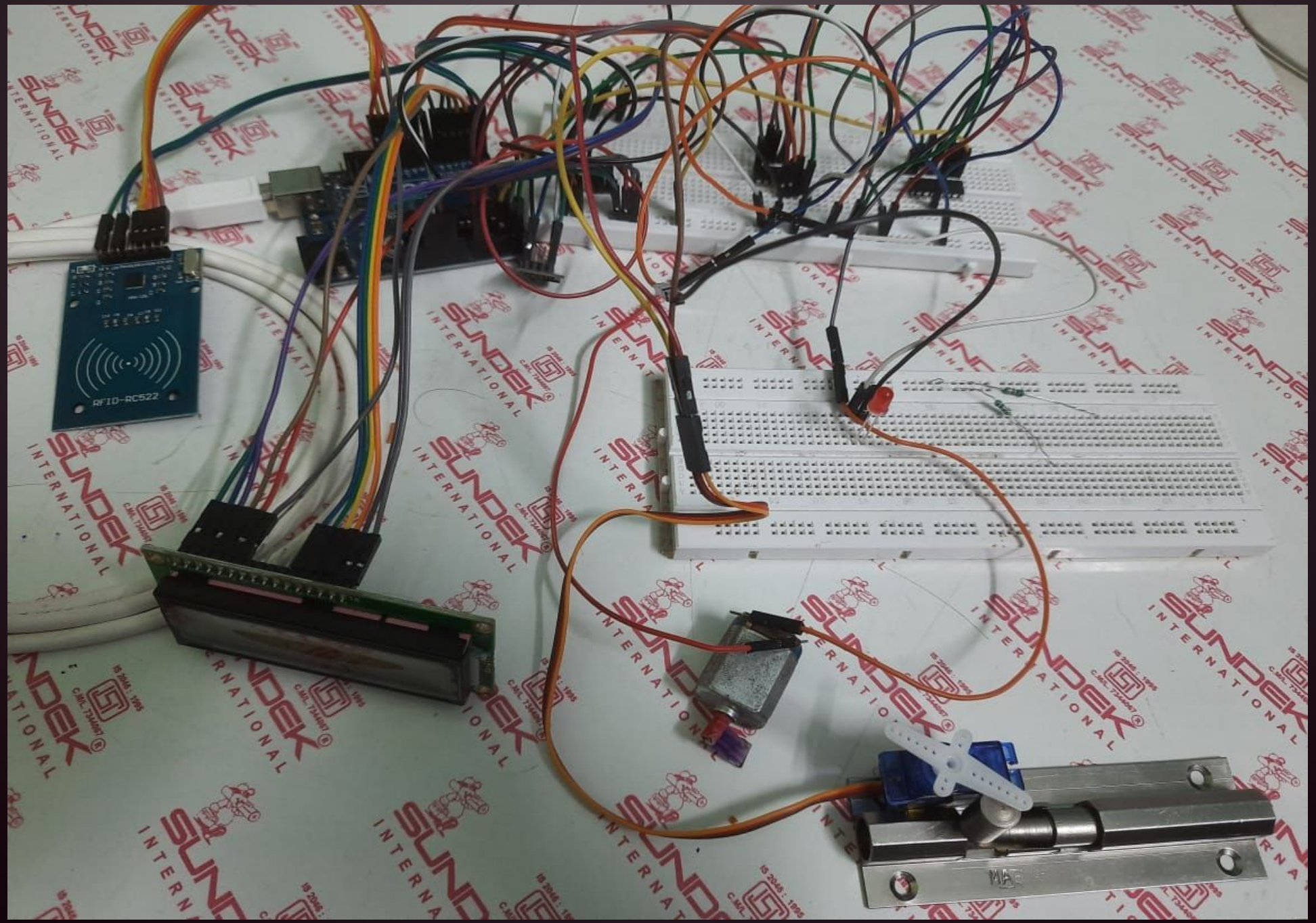
```
void loop()
{
  Blynk.run();
  if ( ! mfrc522.PICC_IsNewCardPresent())       // Look for new cards.No New card = Return Loop
  {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial())         // Select one of the cards.No card = Return Loop
  {
    return;
  }
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
```

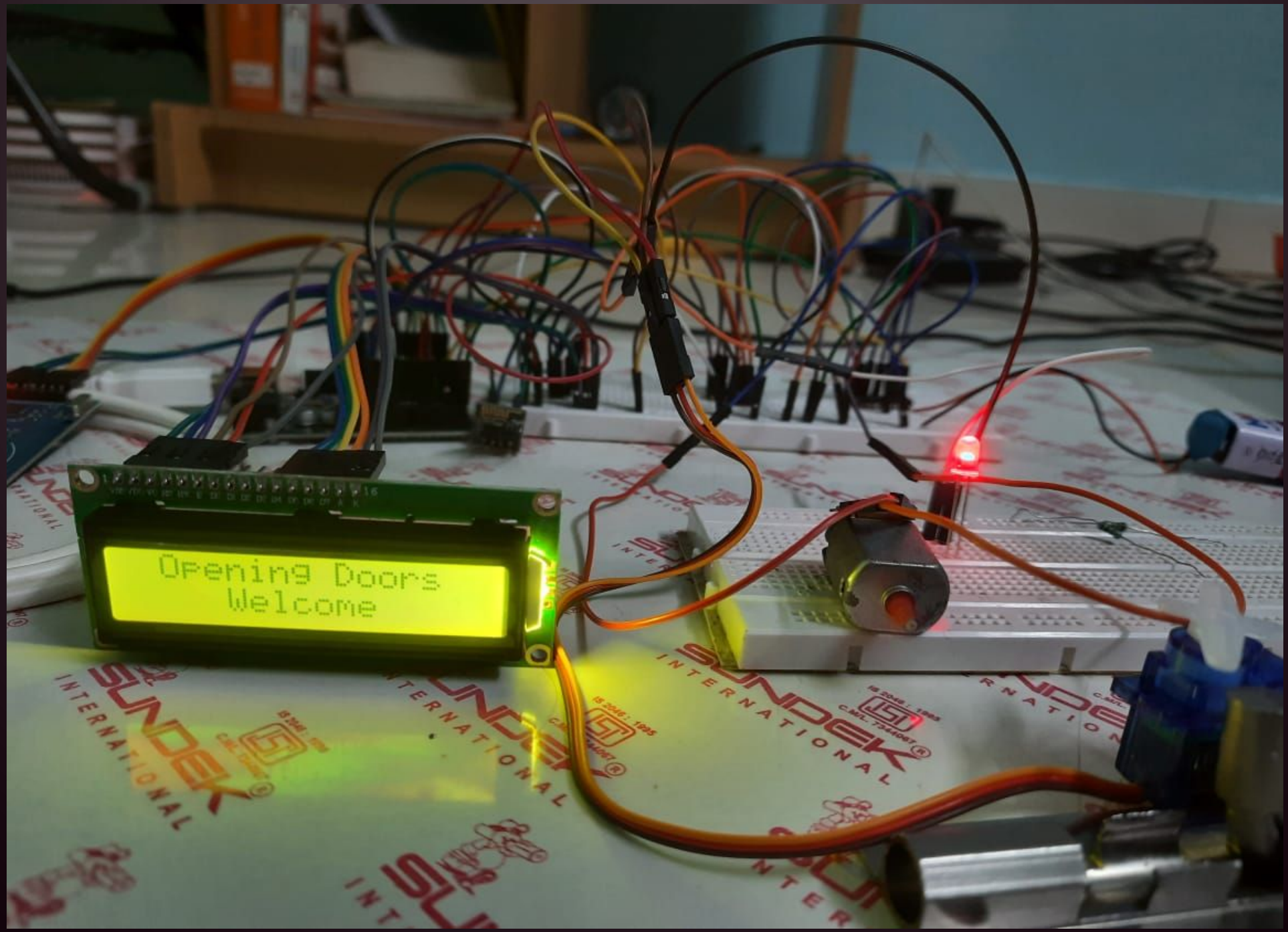
```
Serial.print("Message : ");


content.toUpperCase();
if (content.substring(1) == "53 9A 6D 0C")      //change here the UID of the card/cards that you want to give access
  {


    lcd.setCursor(1,0);
    lcd.print("Opening Doors");
    lcd.setCursor(4,1);
    lcd.print("Welcome");
    myServo.write(Lock_Open_Angle);            //Opening Lock
    digitalWrite(light, HIGH);                 //Turning on Lights(LED)
    turn_on_motors();                          //Turning on Fans(DC MOTOR)


  }
```

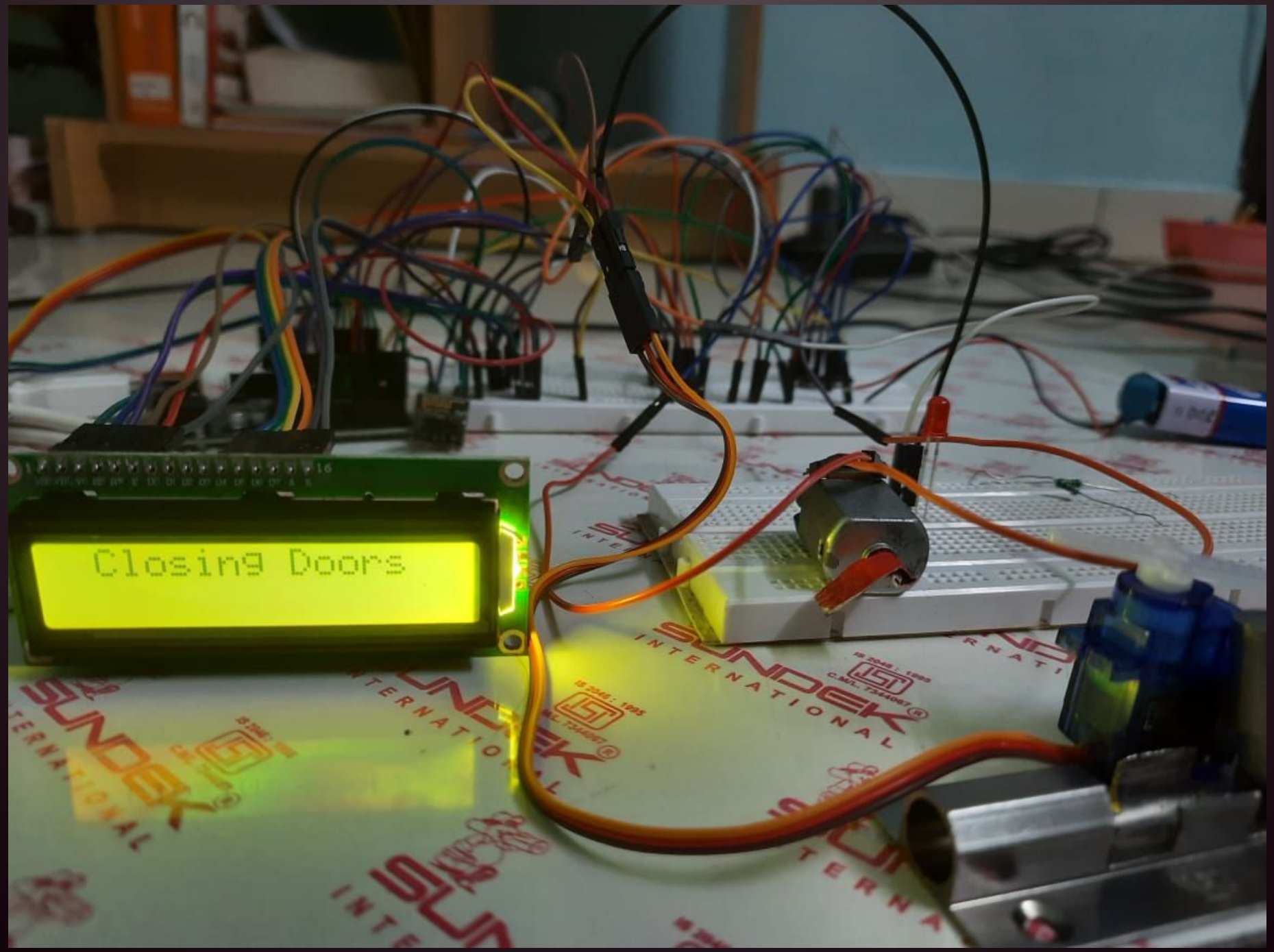
```
else if(content.substring(1) == "57 29 9C 60")
 {
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Closing Doors");
    myServo.write(Lock_Close_Angle);          //Opening Lock
    digitalWrite(light, LOW);                 //Turning Off Lights(LED)
    turn_off_motors();                        //Turning Off Fans(DC MOTOR)
 }
 else
  {
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Access Denied");
  }
}
```

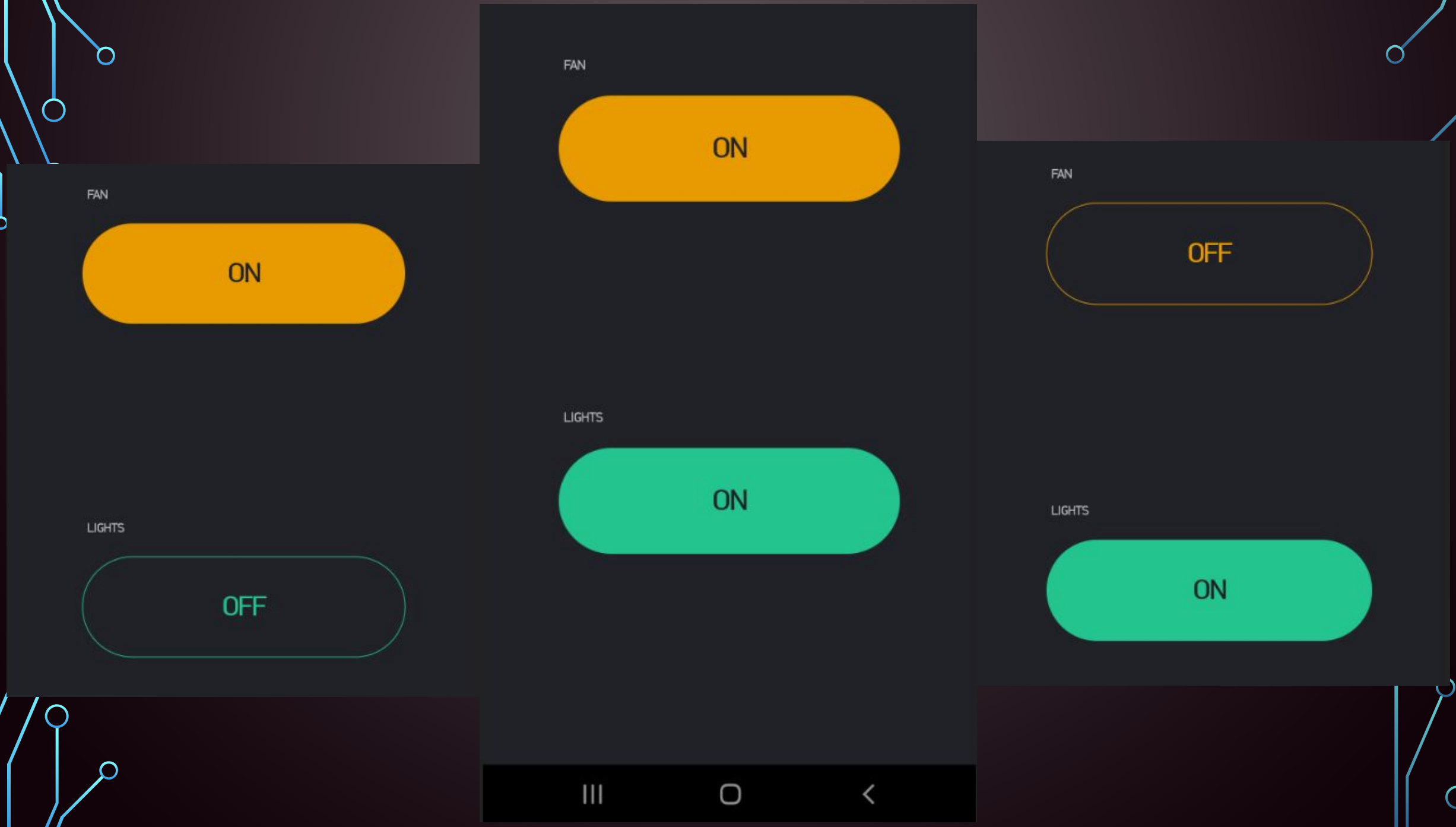
```
void turn_on_motors()
{
  analogWrite(enA, 255);                //RPM controller using PWM
  analogWrite(in1, 0);                  //Opposite Signs of these open the switch and will work
  analogWrite(in2, 255);                //To Reverse Directions Make 0->255 and 255->0
};
void turn_off_motors()
{
  analogWrite(in1, 0);  //Keeping Same Voltage on Both Switches of H-Bridge => Turning OFF
  analogWrite(in2, 0);
};
//end
```

Hardware Model


RFID-RC522


Opening Doors
Welcome


Closing Doors


FAN
ON
LIGHTS
OFF
FAN
ON
LIGHTS
ON
FAN
OFF
LIGHTS
ON

Advantages

- Managing all of our home devices from one place. Enormous **convenience** factor.
- They Maximize home **security**. There are tons of options, only a few dozen of which are currently being explored. For example, home automation systems can connect motion detectors, surveillance cameras, automated door locks, and other tangible security measures throughout the home to activate them from one mobile device monitor activities in real-time whether we are in the house or halfway around the globe.
- **Remote control** of home functions.
- Increased **energy efficiency**. Lights and fans can be programmed to switch to an evening mode as the sun sets, or they can be turned on and off automatically when we enter or leave the room, so we never have to worry about wasting energy.
- Improved appliance **functionality**.
- **Home management** insights. There's also something to be said for our ability to tap into insights on how our home operates.

Future Scope

Due to a lack of resources and time, we could not implement more features into the project. So, in the future, we can work on top of this by adding PIR for the motion sensor, and we can implement occupancy-based lighting and other appliances control, LDR for control of the brightness and temperature sensor to control the thermostat or AC in homes. Furthermore, adding a Cloud-based networking system involves storage and maintenance of data over the Internet location. This gives users the flexibility to have access to the data from any location on the planet. To enable the system to detect power surges, recognize leakages using water or gas sensors, warn about dangerous air pollution, and send timely notifications to prevent complications and much more.

References

https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet

https://components101.com/16x2-lcd-pinout-datasheet

http://www.microchip.ua/wireless/esp01.pdf

https://www.instructables.com/Arduino-Servo-Motors

https://www.instructables.com/Getting-Started-With-the-ESP8266-ESP-01/

Thank You!