VEHICLE ACCESS CONTROL SYSTEM

22AIE113 - Introduction to Communication & IoT

End-Sem Project

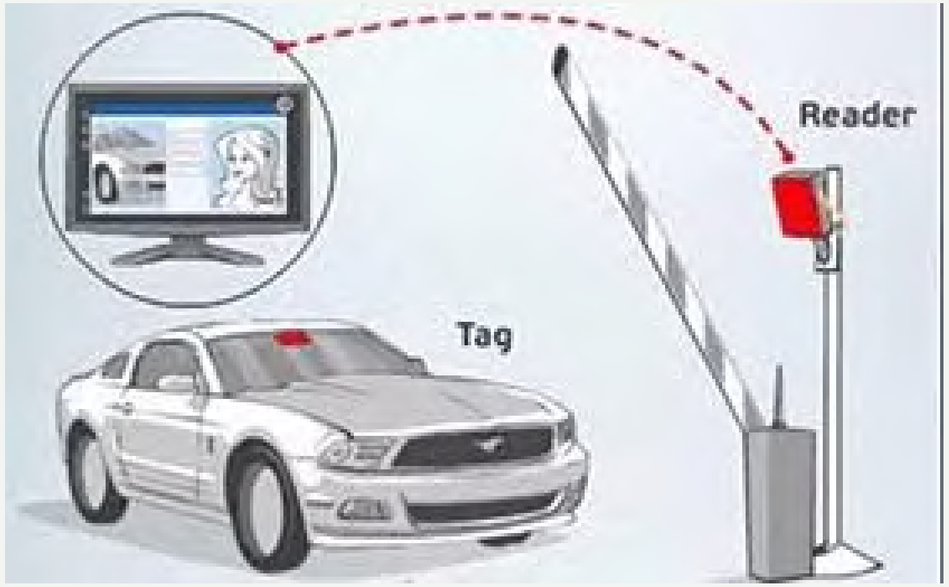
TEAM-10

Introduction

- In today's fast-paced world, managing vehicle access efficiently is crucial for smooth traffic flow and security, especially in parking lots, toll booths, and restricted areas.
- Traditional methods of vehicle access control are often labor-intensive, error-prone, and lack real-time monitoring capabilities, leading to inefficiencies and security concerns.
- This presentation explores an innovative solution: an Automated Vehicle Access Control System that leverages the power of the Internet of Things (IoT) to streamline access management and enhance security.

Functionality

- RFID tags on vehicles enable contactless identification upon approach.
- Strategically placed IR sensors accurately detect arriving and departing vehicles
- Based on authorized RFID tag and IR sensor confirmation
- Servo motor grants access by opening the gate.
- LED indicators provide real-time access status (authorized/unauthorized) through color coding patterns.
- Upon authorized entry, captures data like date, time, and vehicle ID (from RFID tag).



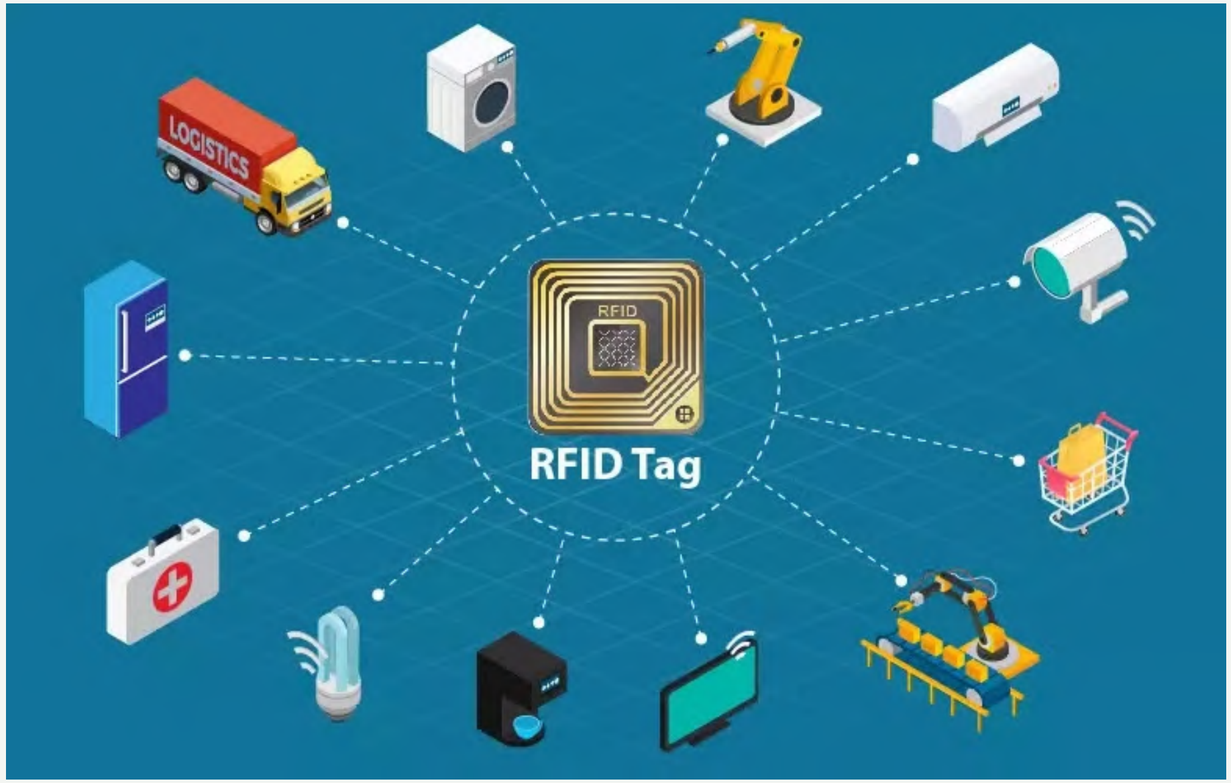
Benifits

- Eliminates manual gate operation, reducing reliance on personnel and wait times.
- Restricts access to authorized vehicles only, preventing unauthorized entry.
- Provides a record of vehicle entries for improved security monitoring and potential identification of suspicious activity.
- Saves time and resources compared to manual systems.
- , the system generates a record of vehicle entries.

RFID in IOT

- RFID (Radio-Frequency Identification) technology in IoT (Internet of Things) refers to the use of RFID tags and readers to communicate information wirelessly within the IoT ecosystem. RFID enables unique identification and tracking of objects, assets, or individuals by utilizing electromagnetic fields for automatic identification and data capture.

- In IoT applications, RFID plays a significant role in providing real-time visibility and traceability of assets throughout various processes.



Problem Statement

To develop an Automated Vehicle Access Control System that integrates RFID technology and IR sensors to efficiently manage vehicle access at designated points like parking lots, toll booths, or restricted areas. The system should be able to verify authorized vehicles, record entry data (such as date, time, and vehicle ID) for easy tracking and analysis. The goal is to enhance security, streamline access management, and improve operational efficiency in various applications that require controlled vehicle entry."

Procedure

System Initialization:

- When the system is powered on, the Arduino initializes all parts (RFID reader, IR sensors, servo motor, LEDs).

Detecting Vehicle at Entry:

- The entrance IR sensor looks for a vehicle. When it detects one, it tells the Arduino to start scanning for an RFID tag.

Scanning RFID Tag:

- The RFID reader scans for a tag. If it finds one, it reads the unique ID (UID) from the tag.

Checking Authorization:

- The Arduino checks if the UID is on a list of authorized IDs.
- If the UID is authorized, the system allows entry.
- If the UID is not authorized, entry is denied.

Procedure

Granting Access:

- The green LED turns on to show access is granted.
- The servo motor opens the gate.
- The system waits until the vehicle passes through the entrance, detected by the exit IR sensor.

Denying Access:

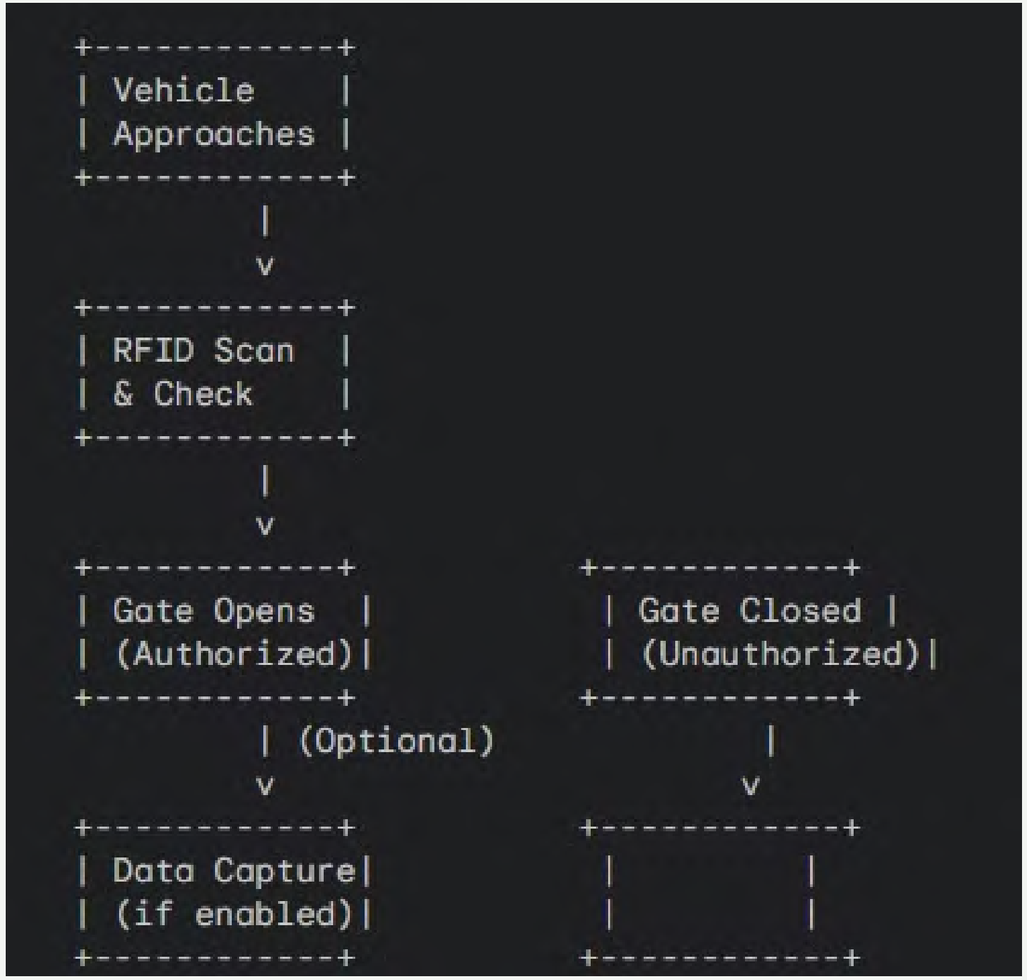
- The red LED turns on to show access is denied.
- After a short wait, the red LED turn off.
- The system resets to check the next vehicle.

Detecting Vehicle at Exit:

- The exit IR sensor watches the vehicle leave.
- When the vehicle passes through, the servo motor closes the gate.
- The system resets, ready for the next vehicle.

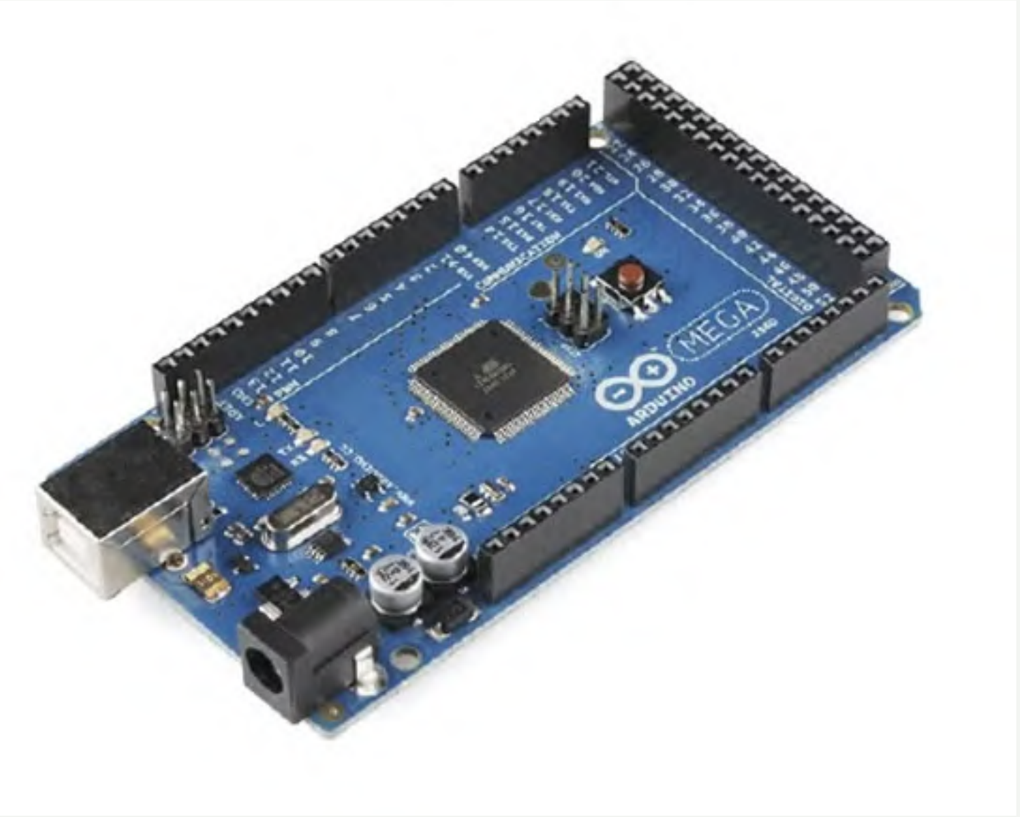
Flow Diagram

This simplified flow diagram captures the core functionalities of your system: RFID identification, gate control, and the optional data logging process.



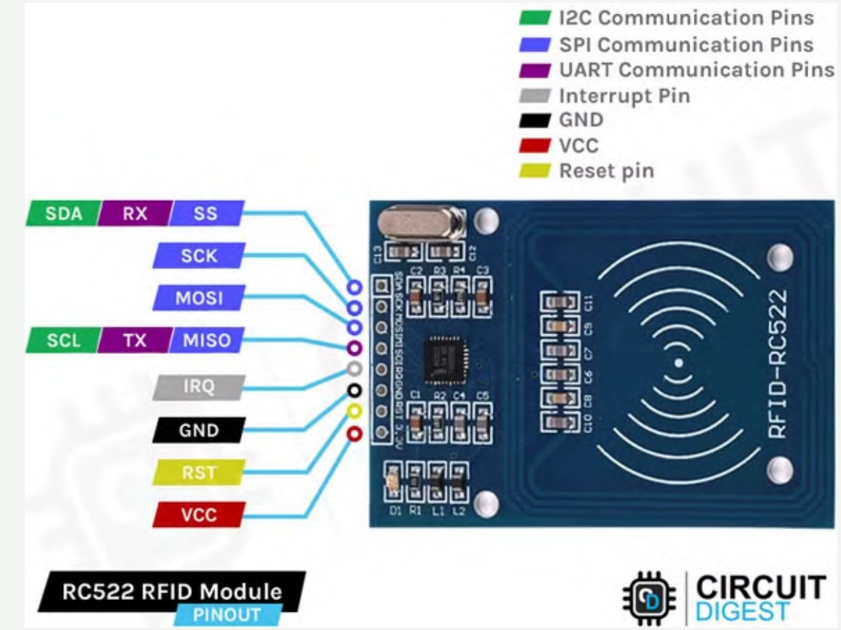
Components

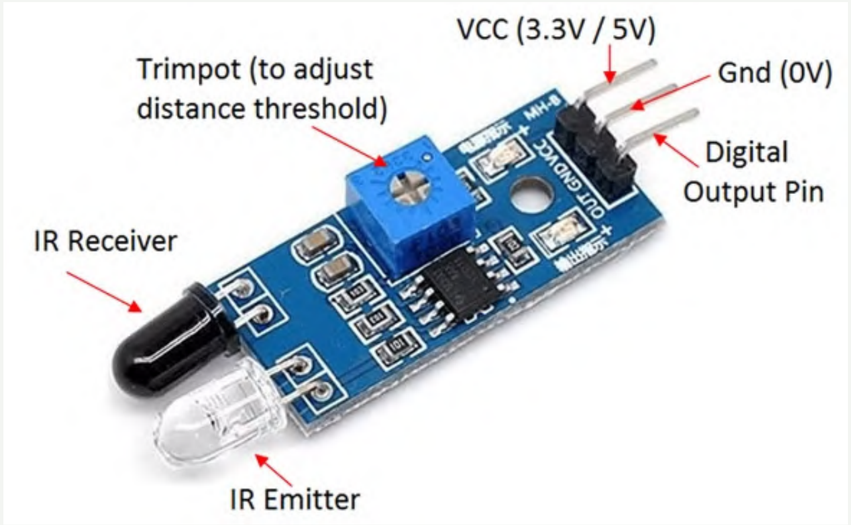
Arduino Mega



Componenets (Sensors)

RFID Reader Module (RC522)

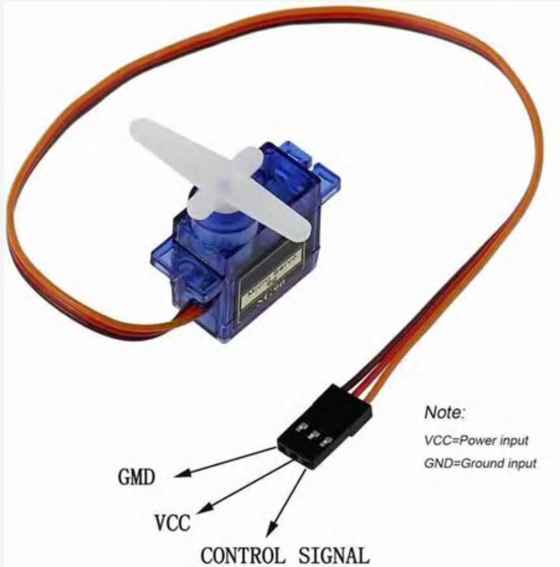




IR Sensor

Componenets (Atuators)

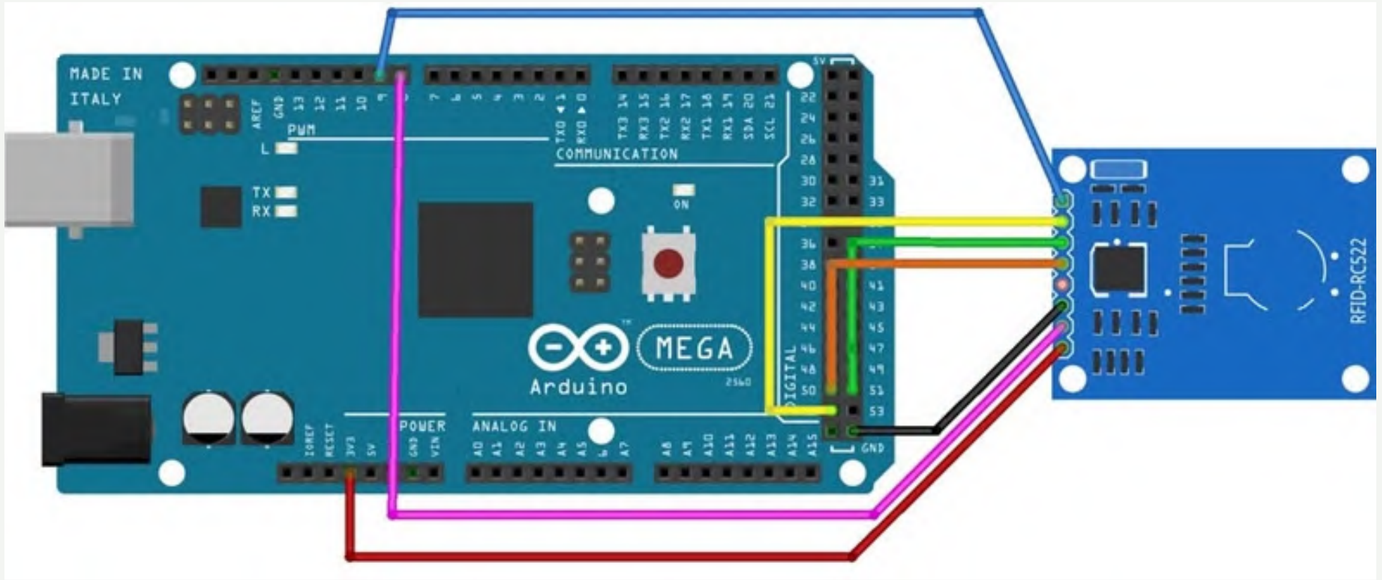
Servo Motor



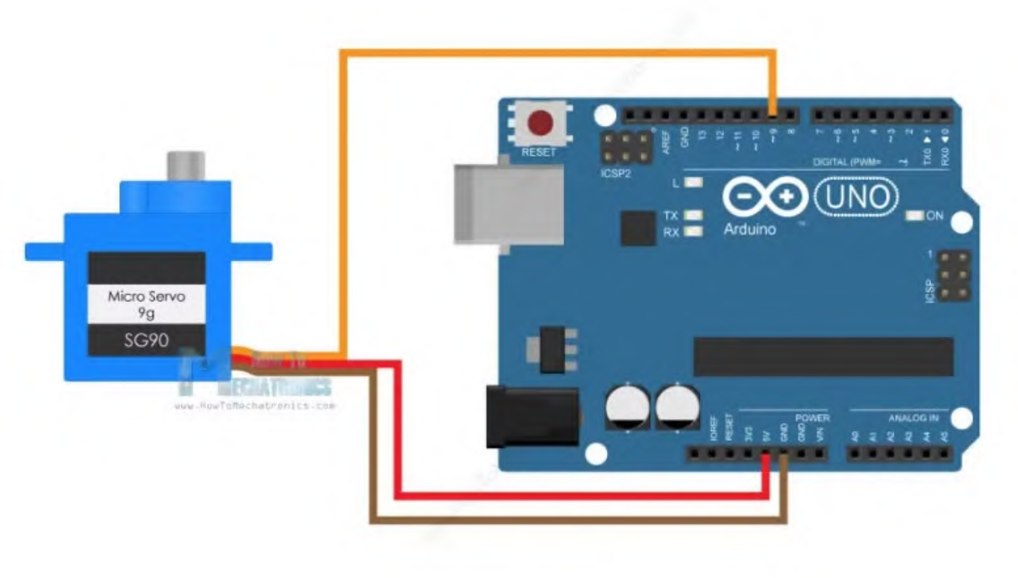
LED Indicators

Connections

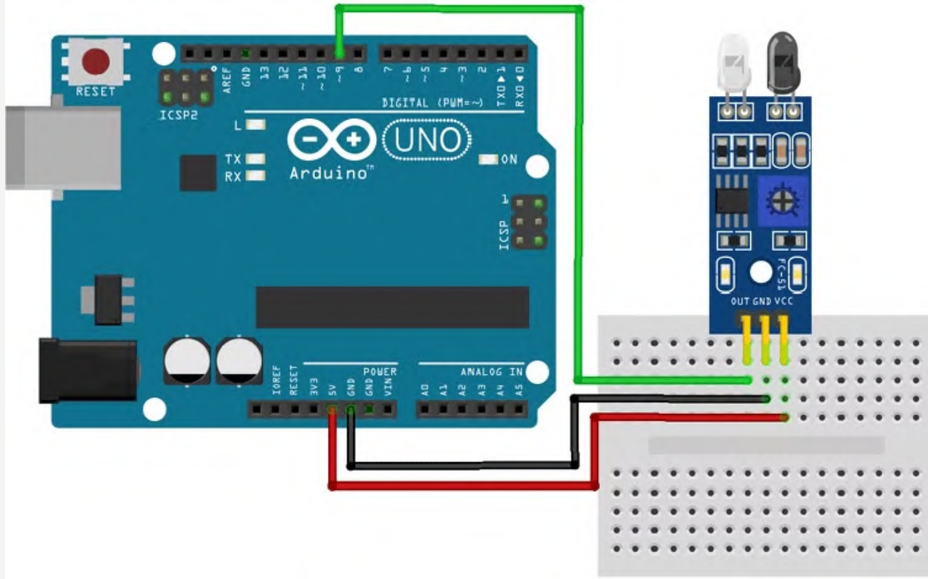
1. **RFID Module:**
   - SDA -> D9
   - SCK -> D52
   - MOSI -> D51
   - MISO -> D50
   - GND -> GND
   - RST -> D8
   - 3.3V -> 3.3V



Connections



Signal -> D4, VCC -> 5V, GND -> GND

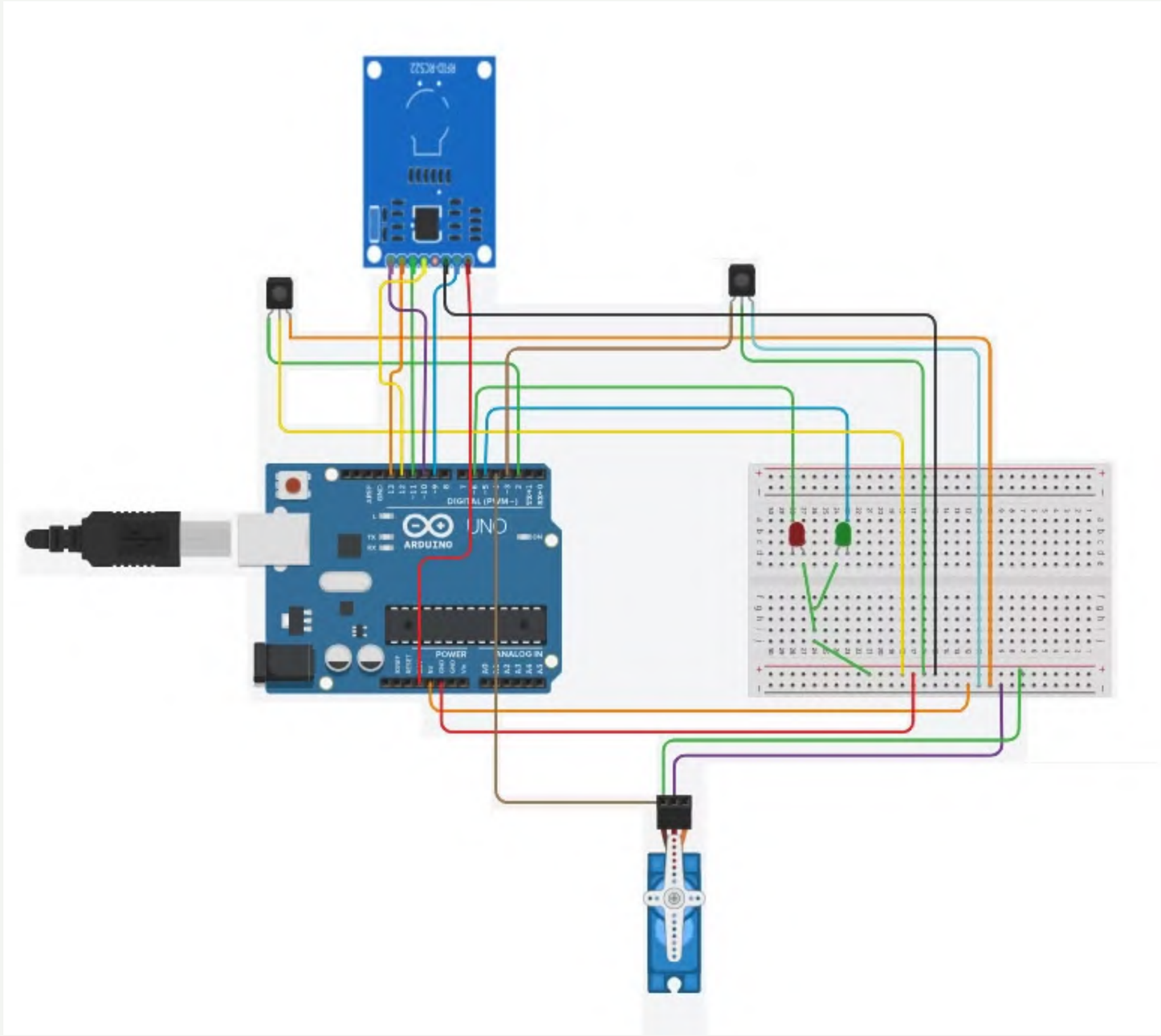


VCC -> 5V, GND -> GND, OUT -> D2 For Entrence

VCC -> 5V, GND -> GND, OUT -> D3 For EXIT

Circuit diagram

(Instead of Arduino UNO we used Arduino Mega 2560)



Code

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

// Pin Definitions
#define SS_PIN 9
#define RST_PIN 8
#define ENTRANCE_SENSOR_PIN 2
#define EXIT_SENSOR_PIN 3
#define SERVO_PIN 4
#define GREEN_LED_PIN 5
#define RED_LED_PIN 6

// RFID Module
MFRC522 mfrc522(SS_PIN, RST_PIN);

// Servo Motor
Servo gateServo;
```

```
// Create a dictionary to store authorized UIDs and their respective names
struct AuthorizedCard {
  String uid;
  String name;
};

AuthorizedCard authorizedCards[] = {
  { "0E43257D", "Alice" },
  { "63BFBB34", "Bob" }
};
const int numAuthorizedCards = sizeof(authorizedCards) / sizeof(AuthorizedCard);

// Variables
bool vehicleDetected = false;
bool gateOpen = false;
unsigned long lastEntranceSensorTriggerTime = 0;
unsigned long lastExitSensorTriggerTime = 0;
```

Code

```
void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();

  pinMode(ENTRANCE_SENSOR_PIN, INPUT);
  pinMode(EXIT_SENSOR_PIN, INPUT);
  pinMode(GREEN_LED_PIN, OUTPUT);
  pinMode(RED_LED_PIN, OUTPUT);

  gateServo.attach(SERVO_PIN);
  gateServo.write(0); // Initial position (gate closed)

  Serial.println("System Initialized. Waiting for vehicle...");
}
```

```
  // Check the exit sensor with debounce
  if (digitalRead(EXIT_SENSOR_PIN) == HIGH && digitalRead(ENTRANCE_SENSOR_PIN) == LOW) {
    lastExitSensorTriggerTime = currentMillis;
    if (vehicleDetected == true) {
      vehicleDetected = false;
      Serial.println("Vehicle detected at the exit");
      closeGate(); // Close gate when vehicle reaches exit sensor
    }
  }
}
```

Code

```
int scanRFID() {
  if (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
    return -1; // No card present
  }

  String uid = "";
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    uid += String(mfrc522.uid.uidByte[i], HEX);
  }
  uid.toUpperCase();

  Serial.print("Card UID: ");
  Serial.println(uid);

  mfrc522.PICC_HaltA();

  return isAuthorized(uid) ? 1 : 0;
}

bool isAuthorized(String uid) {
  for (int i = 0; i < numAuthorizedCards; i++) {
    if (uid == authorizedCards[i].uid) {
      Serial.print("Welcome ");
      Serial.println(authorizedCards[i].name);
      return true;
    }
  }
  return false;
}
```

```
void openGate() {
  Serial.println("Access granted. Opening gate...");
  digitalWrite(GREEN_LED_PIN, HIGH);
  digitalWrite(RED_LED_PIN, LOW);
  gateServo.write(180); // Open gate
  gateOpen = true;

  delay(2000); // Keep the red LED on for 2 seconds

  digitalWrite(GREEN_LED_PIN, LOW);
}

void denyAccess() {
  Serial.println("Access denied.");
  digitalWrite(RED_LED_PIN, HIGH);
  digitalWrite(GREEN_LED_PIN, LOW);
  delay(2000); // Keep the red LED on for 2 seconds
  digitalWrite(RED_LED_PIN, LOW);
}

void closeGate() {
  if (gateOpen) {
    Serial.println("Closing gate...");
    gateServo.write(-20); // Close gate
    gateOpen = false;
  }
}
```

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)