Project Title- Radar on Roads

1. Project Plan and Flow

1. **Vehicle Detection**:
   Use the ultrasonic sensor to detect the presence of a vehicle at a certain distance from the toll booth.
2. **Authentication with RFID**:
   After vehicle detection, prompt the user to place the RFID card near the reader for authentication. If the RFID is valid, proceed to toll deduction.
3. **Barrier Control**:
   Use the servo motor to open the barrier when the RFID is authenticated successfully.
4. **Notifications**:
   - Use LEDs to indicate system status:
     - Green LED: Authentication successful, barrier opening.
     - Red LED: Authentication failed.
   - Use the buzzer for alerting invalid RFID or errors.
5. **System Reset**:
   After the vehicle passes, reset the system to wait for the next vehicle.

2. Required Components

- **Arduino Uno**: The microcontroller to control the system.
- **Ultrasonic Sensor (HC-SR04)**: To detect the presence of vehicles.
- **RFID Module (RC522)**: To read RFID tags.
- **Servo Motor**: To control the toll barrier.
- **LEDs**: To indicate system status.
- **Buzzer**: For notifications.
- **Jumper Wires**: For connections.
- **Breadboard**: For prototyping.

3. Arduino Code

```
#include <Servo.h>

#include <SPI.h>

#include <MFRC522.h>

// Pin Definitions

#define TRIG_PIN 9

#define ECHO_PIN 10
```

```
#define GREEN_LED 7
#define RED_LED 8
#define BUZZER 3
#define SERVO_PIN 6
#define RST_PIN 5
#define SS_PIN 4
// Objects
Servo barrierServo;
MFRC522 rfid(SS_PIN, RST_PIN);
// Variables
long duration;
int distance;
void setup() {
  // Initialize components
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  barrierServo.attach(SERVO_PIN);
  barrierServo.write(0); // Gate closed position
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  // Initial State
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
  Serial.println("System Ready. Waiting for vehicle...");
}
void loop() {
  distance = getDistance();
  if (distance < 20) { // Vehicle detected
    Serial.println("Vehicle detected. Waiting for RFID...");
    if (authenticateRFID()) {
      grantAccess();
    } else {
      denyAccess();
    }
    delay(3000); // Reset delay
  }
}
int getDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2; // Distance in cm
}
bool authenticateRFID() {
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
    return false;
  }
  String uid = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    uid += String(rfid.uid.uidByte[i], HEX);
  }
  uid.toUpperCase();
  Serial.println("RFID UID: " + uid);
  return (uid == "A1B2C3D4"); // Example valid UID
}
void grantAccess() {
  Serial.println("Access Granted.");
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
  barrierServo.write(90); // Open gate
  delay(5000);            // Keep gate open for 5 seconds
  barrierServo.write(0);  // Close gate
  digitalWrite(GREEN_LED, LOW);
}
void denyAccess() {
  Serial.println("Access Denied.");
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(BUZZER, HIGH);
  delay(1000); // Notification duration
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
}
```

4. LEDs and Buzzer for Notifications

```
#define GREEN_LED 7
#define RED_LED 8
#define BUZZER 3
void setup() {
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
}
void notifyAccessGranted() {
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
  delay(5000); // Notification duration
  digitalWrite(GREEN_LED, LOW);
}
void notifyAccessDenied() {
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(BUZZER, HIGH);
  delay(1000); // Notification duration
  digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);
}
```

6. RFID Authentication

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 5
#define SS_PIN 4
MFRC522 rfid(SS_PIN, RST_PIN);
```

```
void setup() {
  SPI.begin();
  rfid.PCD_Init();
  Serial.begin(9600);
}
bool authenticateRFID() {
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
    return false;
  }
  String uid = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    uid += String(rfid.uid.uidByte[i], HEX);
  }
  uid.toUpperCase();
  Serial.println("RFID UID: " + uid);
  return (uid == "A1B2C3D4"); // Example valid UID
}
```

7. Ultrasonic Sensor for Vehicle Detection

```
#define TRIG_PIN 9
#define ECHO_PIN 10
long duration;
int distance;
```

```
void setup() {

  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);

  Serial.begin(9600);

}

int getDistance() {

  digitalWrite(TRIG_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG_PIN, LOW);

  duration = pulseIn(ECHO_PIN, HIGH);

  return duration * 0.034 / 2; // Distance in cm

}
```

8. Testing

1. **Vehicle Detection**: Ensure the ultrasonic sensor detects vehicles correctly.
2. **RFID Authentication**: Test with both valid and invalid RFID tags.
3. **Servo Motor**: Confirm the barrier opens and closes smoothly.
4. **LEDs and Buzzer**: Check the status indicators.