**Table of Contents**

[**Introduction** 2](#_Toc197165348)

[Software Components Requirements 2](#_Toc197165349)

[**Arduino IDE** 3](#_Toc197165350)

[**Hardware Description** 4](#_Toc197165351)

[**Arduino Nano:** 4](#_Toc197165352)

[**Keypad :** 4](#_Toc197165353)

[**Jumper Wires:** 4](#_Toc197165354)

[**Types of Jumper Wires** 4](#_Toc197165355)

[1. Male-to-Male: 4](#_Toc197165356)

[2. Male-to-Female 5](#_Toc197165357)

[3. Female-to-Female: 5](#_Toc197165358)

[**Micro Servo:** 5](#_Toc197165359)

[**LCD Display** 5](#_Toc197165360)

[**IR Sensor** 5](#_Toc197165361)

[**Buzzer** 6](#_Toc197165362)

[**Radio Frequency Identification (RFID)** 6](#_Toc197165363)

[**Circuit Diagram** 7](#_Toc197165364)

[**Code For Arduino IDE** 8](#_Toc197165365)

**Toll System Using Arduino Nano**

# **Introduction**

Design and development of an RFID-based automatic toll collecting system that detects vehicles, uses RFID cards to identify authorized users, automatically deducts toll charges, and uses sensors to control a servo-operated gate, so lowering manual intervention, saving time, and improving efficiency at toll booths.

Beside this you need to download software that Arduino IDE.  
Hardware Components Requirements:

1. Arduino Nano
2. Keypad
3. Jumper wire
4. Motor
5. Servo
6. LCD
7. IR sensor
8. Buzzer
9. RDID

## Software Components Requirements

1. Arduino IDE

# **Arduino IDE**

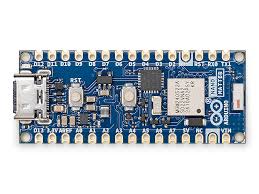
Arduino IDE is the open source platform to write the code which can understood by the Arduino. The process to download the Arduino IDE are listed below and you can follow the same steps to download in your computer as well. Here we go,

**Process to Download the Arduino IDE:**

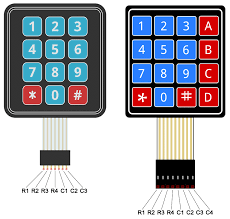
1. Open your favorite browser such as: Google Chrome, Fire Fox, Edge
2. In the URL box type Download Arduino IDE
3. Click on the first option or you can directly go this URL: <https://www.arduino.cc/en/software>
4. Go to the DOWNLOAD OPTION and click on as per your OS.
5. After that click on Just download
6. Lastly again click on just download
7. After being download install it in your PC

# **Hardware Description**

# **Arduino Nano:**

The Arduino Nano is a compact and versatile microcontroller board based on the ATmega328P microcontroller. It is a breadboard-friendly board that retains the functionalities of the larger Arduino boards but in a smaller form factor. The Nano is particularly popular in prototyping, educational contexts, and embedded system development due to its ease of use and extensive support community. Common applications include DIY electronics projects, robotics, sensor interfacing, and interactive artworks.

## **Keypad :**

The 4X4 Membrane Matrix Keypad is a user input device manufactured by Parallax Inc., part number 27899. This keypad consists of 16 buttons arranged in a 4x4 grid, allowing for a compact and versatile method of capturing user input. It is commonly used in electronic projects such as security systems, telephone dial pads, and various control panels.

## **Jumper Wires:**

Jumper wires are simple electrical wires with connector pins at both ends, designed to connect various components in a circuit without soldering. They are widely used in prototyping and breadboarding, allowing hobbyists and engineers to quickly and efficiently build and test circuits.

## **Types of Jumper Wires**

1. Male-to-Male: Both ends have pins that fit into female connectors or breadboard holes.
2. Male-to-Female: One end has a pin (male), and the other has a socket (female), making them suitable for connecting breadboards to components or modules with male headers.
3. Female-to-Female: Both ends have sockets, ideal for connecting components or modules with male pins.

Jumper wires are usually color-coded (e.g., red for power, black for ground) to help organize and distinguish connections.

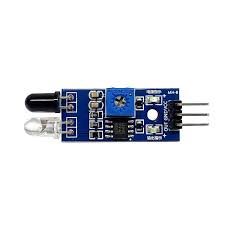
# **Micro Servo:**

The UNO MG995 Servo Motor is a high-torque, high-speed rotary actuator designed for precise control of angular position, velocity, and acceleration. It integrates a motor with a position feedback sensor, making it ideal for applications requiring accurate and repeatable movements. Common applications include robotics, RC vehicles, and automated machinery.

# **LCD Display**

The LCD 16x2 with I2C interface is a versatile and widely used display module capable of showing 16 characters per line across 2 lines. It is equipped with an I2C (Inter-Integrated Circuit) interface, which simplifies communication with microcontrollers by reducing the number of required pins. This makes it an excellent choice for projects where pin availability is limited or where simplicity is desired.

# **IR Sensor**

An infrared (IR) sensor, manufactured by Arduino (Part ID: G), is a versatile electronic component designed to detect infrared radiation. It is commonly used in applications such as motion detection, proximity sensing, and remote control systems. The sensor operates by emitting or detecting IR light, making it ideal for non-contact sensing tasks.

IR sensors are widely used in robotics, home automation, and security systems due to their reliability and ease of integration into various circuits.

# **Buzzer**

A buzzer is an audio signaling device that produces sound when an electric current passes through it. It is widely used in various electronic applications to provide audible alerts or notifications. Buzzers are commonly found in alarms, timers, household appliances, and embedded systems. They are available in two main types: active buzzers, which generate sound when powered, and passive buzzers, which require an external signal to produce sound.

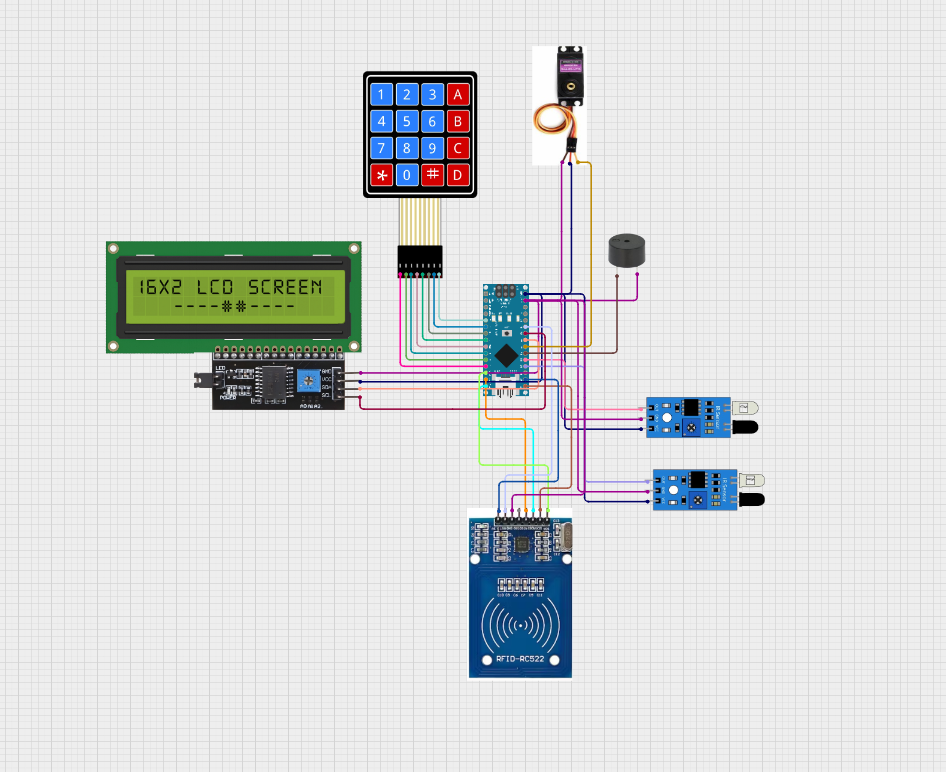
# **Radio Frequency Identification (RFID)**

Radio Frequency Identification (RFID) is a technology that uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID system consists of two main components: a reader and tags. Tags can be either passive (powered by the reader's signal) or active (equipped with their own power source).

RFID is widely used in various applications, including:

* Inventory management
* Access control systems
* Asset tracking and monitoring
* Contactless payment systems
* Library book management

# **Circuit Diagram**



cirkitstudio

# **Code For Arduino IDE**

#include <SPI.h>

#include <MFRC522.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Keypad.h>

#include <Servo.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

Servo servo;

int servoPos = 0;

#define sensorPin1 A0

#define sensorPin2 A1

#define buzzerPin A2

int senVal1 = 0;

int senVal2 = 0;

#define RST\_PIN A6

#define SS\_PIN 10

int card1Balance = 5000;

int card2Balance = 300;

#define num 7

char Data[num];

byte data\_count = 0;

String num1, num2, card;

int a, b;

char Key;

bool recharge = true;

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

int state = 0;

// Keypad setup

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

  {'1','2','3','A'},

  {'4','5','6','B'},

  {'7','8','9','C'},

  {'\*','0','#','D'}

};

byte rowPins[ROWS] = {9, 8, 7, 6};

byte colPins[COLS] = {5, 4, 3, 2};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// ---------- Function Prototypes ----------

void lcdPrint();

void LcdPrint();

void clearData();

void reCharge();

void sensorRead();

void rfid();

void KeyPad();

void servoDown();

void servoUp();

void setup() {

  lcd.init();

  lcd.backlight();

  Serial.begin(9600);

  servo.attach(A3);

  servo.write(0); // Start closed

  pinMode(sensorPin1, INPUT);

  pinMode(sensorPin2, INPUT);

  pinMode(buzzerPin, OUTPUT);

  SPI.begin();

  mfrc522.PCD\_Init();

  lcd.setCursor(0, 0);

  lcd.print(" Automatic toll");

  lcd.setCursor(0, 1);

  lcd.print("collection system");

  delay(3000);

  lcd.clear();

}

void loop() {

  if (!recharge) {

    reCharge();

  } else {

    lcd.setCursor(0, 0);

    lcd.print("   Welcome!!!");

    sensorRead();

    rfid();

    KeyPad();

    if (senVal1 == 0) {

      servoDown();

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Vehicle detected");

      delay(1000);

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Put your card to");

      lcd.setCursor(0, 1);

      lcd.print("the reader......");

      delay(2000);

      lcd.clear();

    } else if (senVal2 == 0 && state == 1) {

      servoUp();

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Have a safe");

      lcd.setCursor(0, 1);

      lcd.print("journey");

      delay(1000);

      lcd.clear();

      state = 0;

    }

  }

}

void servoDown() {

  for (servoPos = 0; servoPos <= 90; servoPos++) {

    servo.write(servoPos);

    delay(10);

  }

}

void servoUp() {

  for (servoPos = 90; servoPos >= 0; servoPos--) {

    servo.write(servoPos);

    delay(10);

  }

}

void sensorRead() {

  senVal1 = digitalRead(sensorPin1);

  senVal2 = digitalRead(sensorPin2);

}

void rfid() {

  if (!mfrc522.PICC\_IsNewCardPresent() || !mfrc522.PICC\_ReadCardSerial()) {

    return;

  }

  String content = "";

  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));

  }

  content.toUpperCase();

  if (content.substring(1) == "24 EA 29 03") {

    if (card1Balance >= 500) {

      lcdPrint();

      card1Balance -= 500;

      lcd.setCursor(9, 1);

      lcd.print(card1Balance);

      delay(2000);

      lcd.clear();

      state = 1;

    } else {

      card = content.substring(1);

      LcdPrint();

      lcd.setCursor(9, 1);

      lcd.print(card1Balance);

      lcd.print(" Tk");

      delay(2000);

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Please Recharge");

      delay(1000);

      lcd.clear();

      state = 0;

    }

  } else if (content.substring(1) == "D3 DC C1 01") {

    if (card2Balance >= 500) {

      lcdPrint();

      card2Balance -= 500;

      lcd.setCursor(9, 1);

      lcd.print(card2Balance);

      delay(2000);

      lcd.clear();

      state = 1;

    } else {

      card = content.substring(1);

      LcdPrint();

      lcd.setCursor(9, 1);

      lcd.print(card2Balance);

      lcd.print(" Tk");

      delay(2000);

      lcd.clear();

      lcd.setCursor(0, 0);

      lcd.print("Please Recharge");

      delay(1000);

      lcd.clear();

      state = 0;

    }

  } else {

    digitalWrite(buzzerPin, HIGH);

    lcd.setCursor(0, 0);

    lcd.print("Unknown Vehicle");

    lcd.setCursor(0, 1);

    lcd.print("Access denied");

    delay(1500);

    lcd.clear();

    digitalWrite(buzzerPin, LOW);

  }

}

void KeyPad() {

  char key = keypad.getKey();

  if (key && key == 'A') {

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Recharging Mode.");

    lcd.setCursor(0, 1);

    lcd.print("................");

    delay(1500);

    lcd.clear();

    recharge = false;

  }

}

void clearData() {

  while (data\_count != 0) {

    Data[data\_count--] = 0;

  }

}

void reCharge() {

  lcd.setCursor(0, 0);

  lcd.print("Enter the amount");

  char key = keypad.getKey();

  if (key) {

    if (key == 'D') {

      if (card == "24 EA 29 03") {

        num1 = Data;

        card1Balance += num1.toInt();

        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("Your current");

        lcd.setCursor(0, 1);

        lcd.print("balance: ");

        lcd.print(card1Balance);

        lcd.print(" Tk");

        delay(3000);

        clearData();

        lcd.clear();

        recharge = true;

      } else if (card == "D3 DC C1 01") {

        num2 = Data;

        card2Balance += num2.toInt();

        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("Your current");

        lcd.setCursor(0, 1);

        lcd.print("balance: ");

        lcd.print(card2Balance);

        lcd.print(" Tk");

        delay(3000);

        clearData();

        lcd.clear();

        recharge = true;

      }

    } else if (data\_count < num - 1) {

      Data[data\_count] = key;

      lcd.setCursor(data\_count, 1);

      lcd.print(Data[data\_count]);

      data\_count++;

    }

  }

}

void lcdPrint() {

  digitalWrite(buzzerPin, HIGH);

  delay(200);

  digitalWrite(buzzerPin, LOW);

  delay(100);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("  Successfully");

  lcd.setCursor(0, 1);

  lcd.print(" paid your bill");

  delay(1500);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Your Remaining");

  lcd.setCursor(0, 1);

  lcd.print("balance: ");

}

void LcdPrint() {

  digitalWrite(buzzerPin, HIGH);

  delay(200);

  digitalWrite(buzzerPin, LOW);

  delay(100);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("  Your balance");

  lcd.setCursor(0, 1);

  lcd.print(" is insufficient");

  delay(1500);

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Your Remaining");

  lcd.setCursor(0, 1);

  lcd.print("balance: ");

}