Course Title: Microprocessor And Interfacing Lab	Course Code: CSE-368

Shahjalal University of Science & Technology, Sylhet

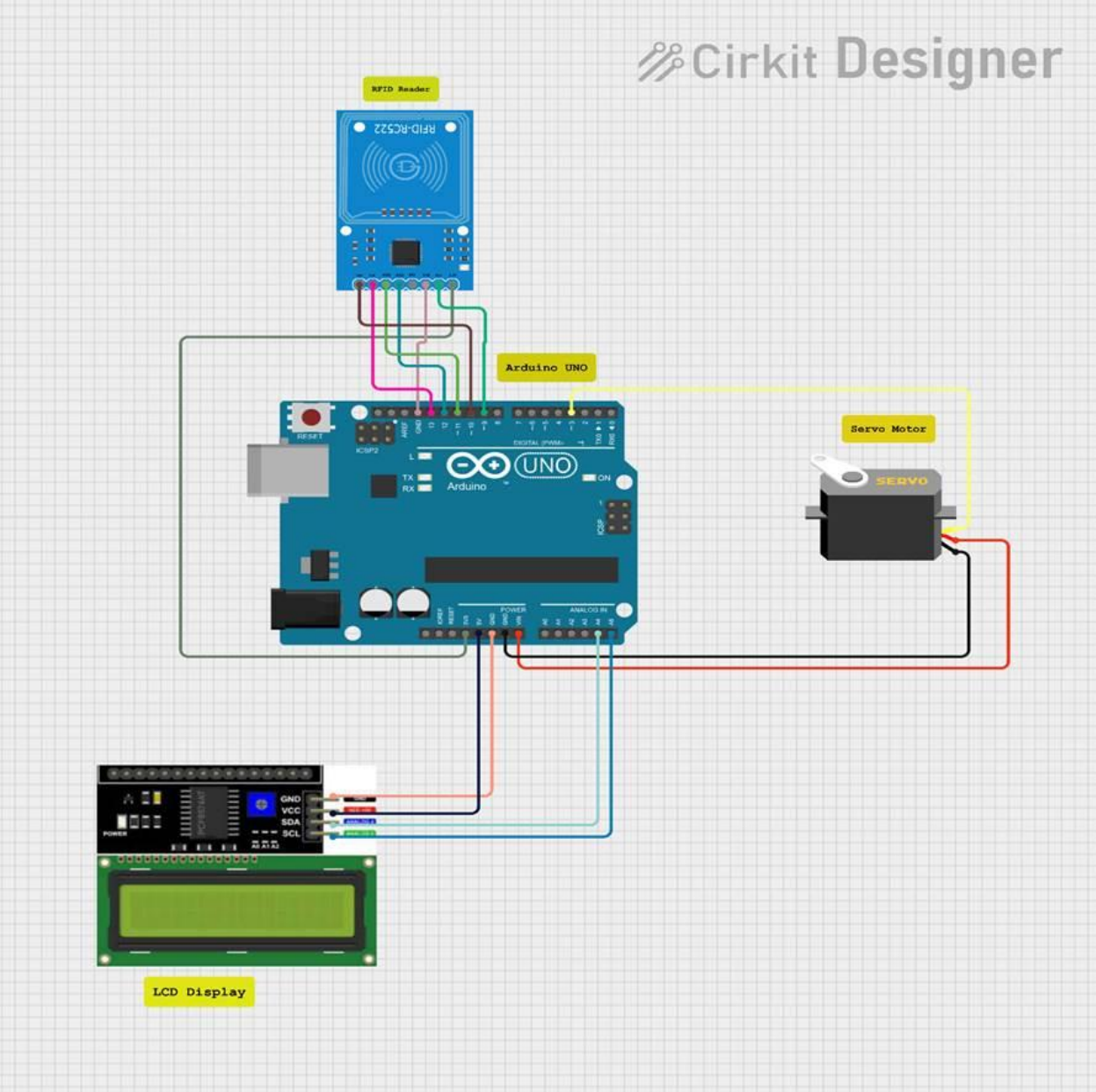
Submitted By (Group 16)	Submitted to
Md. Nazmul Hossen - 2019331047 **Samim Ahmed - 2020331065** **Imran Bin Azad Siyam - 2020331101** **Ajoad Islam - 2020331104)** **Muntasir Mamun - 2020331110**	**Abdullah Al Noman** **Lecturer,** **Department of Computer Science and Engineering,** **Shahjalal University of Science and Technology, Sylhet**

Smart Door Security: An IOT-Based RFID Lock System

Github Link:

https://github.com/NazmulRahul/Group_16

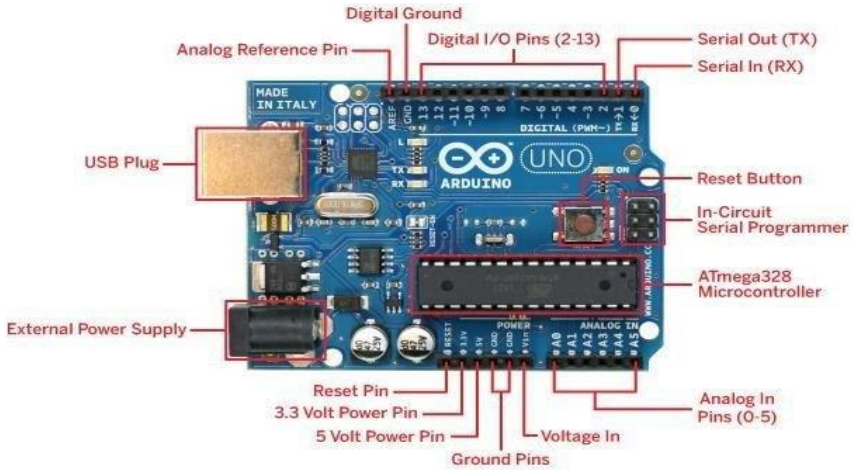
Diagram:



Components Needed:

1. **Arduino Uno**
2. **RFID Reader (MFRC522)**
3. **Servo Motor** (e.g., SG90)
4. **LED Display (16x2 LCD with I2C interface)** (optional, for visual feedback)
5. **Connecting Wires**
6. **Door Lock**

1. Arduino Uno



The Arduino Uno microcontroller is a popular choice for many projects due to its simplicity and versatility. It features several digital and analog input/output pins that allow interfacing with various components like sensors, displays, motors, and more. Here's an overview of the digital pins (D0 to D13) on the Arduino Uno:

Digital Pins (D0-D13)

1. **D0 (RX)**: Used for receiving serial data. Typically connected to the TX pin of a serial device (not used in this example).
2. **D1 (TX)**: Used for transmitting serial data. Typically connected to the RX pin of a serial device (not used in this example).
3. **D2**: Connect to the IN pin of the relay module to control the lock.
4. **D3**: Connect to the control pin of the buzzer for sound feedback.
5. **D4**: Connect to the control pin of an LED for visual feedback (optional).
6. **D5-D8**: Available for connecting a keypad (optional).
7. **D9**: Connect to the RST pin of the RFID reader.
8. **D10**: Connect to the SDA/SS pin of the RFID reader (Slave Select for SPI communication).
9. **D11**: Connect to the MOSI pin of the RFID reader (Master-Out-Slave-In for SPI communication).
10. **D12**: Connect to the MISO pin of the RFID reader (Master-In-Slave-Out for SPI communication).

11. **D13**: Connect to the SCK pin of the RFID reader (Serial Clock for SPI communication).

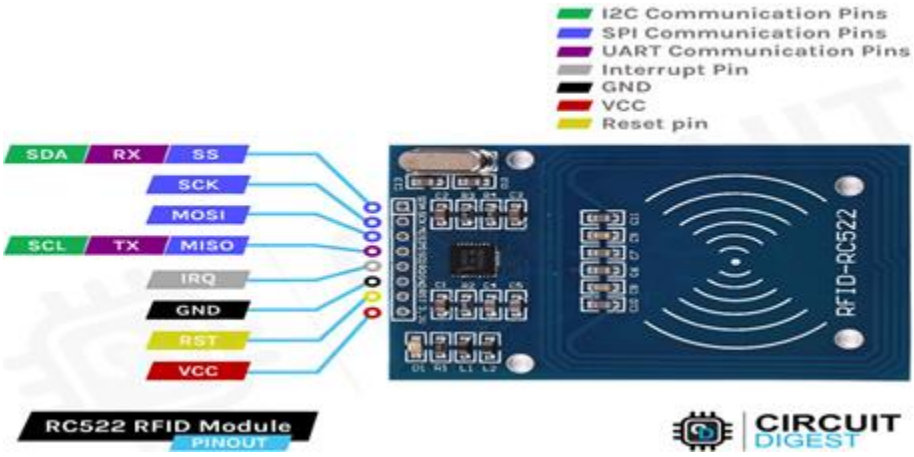
Analog Pins (A0-A5)

1. **A0-A3**: Available for connecting additional sensors or keypad columns (optional).
2. **A4 (SDA)**: Connect to the SDA pin of an RTC module for I2C communication (optional).
3. **A5 (SCL)**: Connect to the SCL pin of an RTC module for I2C communication (optional).

Power Pins

1. **VIN**: The input voltage to the Arduino board when using an external power source (6-12V). Can be used to power the board.
2. **5V**: Provides a regulated 5V output. Can be used to power other components like the relay module, buzzer, and optional components.
3. **3.3V**: Provides a regulated 3.3V output. Typically used to power the RFID reader.
4. **GND**: Ground pins. Common ground for all components.
5. **IOREF**: Provides the voltage reference with which the microcontroller operates.

2. RFID Reader (MFRC522)

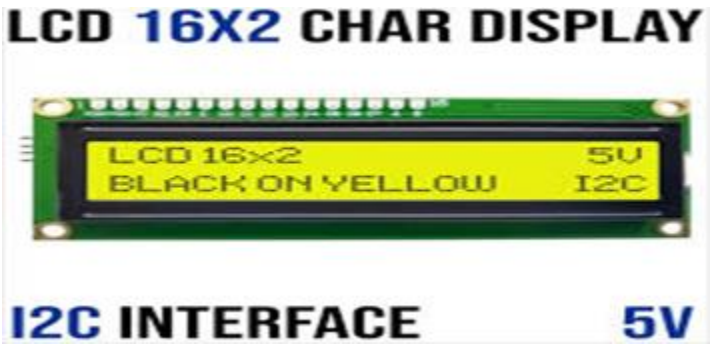


Components and Pins

1. **MFRC522 Module**: Contains the RFID reader IC (MFRC522) and associated components.
2. **Antenna**: Used for transmitting and receiving RF signals to/from RFID tags.
3. **Power Pins**:
   - **VCC**: Typically connects to 3.3V (some can handle 5V).
   - **GND**: Connects to the ground of the power supply.
4. **SPI Communication Pins**:
   - **RST (Reset)**: Used to reset the MFRC522 module. Connect to a digital pin on the Arduino (e.g., D9).
   - **SDA (Serial Data)**: Also known as NSS or SS (Slave Select), used for SPI communication. Connect to D10 on Arduino.
   - **SCK (Serial Clock)**: Clock signal for SPI communication. Connect to D13 on Arduino.

- **MOSI (Master Out Slave In)**: Data from Arduino to MFRC522. Connect to D11 on Arduino.
- **MISO (Master In Slave Out)**: Data from MFRC522 to Arduino. Connect to D12 on Arduino.

3. LED Display (16x2 LCD with I2C Interface)



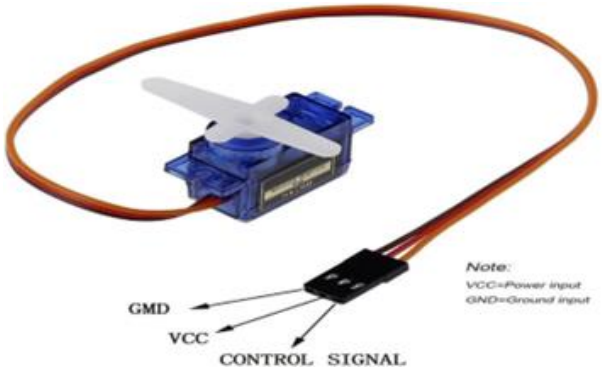
Components and Their Pins

1. **16x2 LCD Display**: This display can show 16 characters per line, and it has 2 lines.
2. **I2C Interface Module**: This module connects to the LCD display and provides an I2C interface, which simplifies wiring and communication.

I2C Interface Pins

1. **VCC**: Connect to 5V on the Arduino to power the module.
2. **GND**: Connect to GND on the Arduino for the common ground.
3. **SDA**: Connect to A4 (SDA) on the Arduino for I2C data line.
4. **SCL**: Connect to A5 (SCL) on the Arduino for I2C clock line.

4. Servo motor



Integrating a servo motor into an RFID door lock system can allow you to control the locking mechanism more precisely. A servo motor can be used to move a latch or bolt to lock and unlock the door. Here's a comprehensive guide on how to integrate a servo motor with an Arduino Uno in an RFID door lock system.

5. RFID Cards

Connection Overview:

1. **RFID Reader**: Connected to the Arduino for reading RFID tags.
2. **Servo Motor**: Controls the locking mechanism.
3. **LED Display**: Displays messages like "Access Granted" or "Access Denied".

Detailed Connections:

RFID Reader (MFRC522)

- **VCC**: Connect to 3.3V on the Arduino.
- **GND**: Connect to GND on the Arduino.
- **RST**: Connect to D9.
- **MISO**: Connect to D12.
- **MOSI**: Connect to D11.
- **SCK**: Connect to D13.
- **SDA/SS**: Connect to D10.

Servo Motor

- **VCC**: Connect to 5V on the Arduino.
- **GND**: Connect to GND on the Arduino.

- **Signal**: Connect to D3.

LED Display

- **VCC**: Connect to 5V on the Arduino.
- **GND**: Connect to GND on the Arduino.
- **SDA**: Connect to A4 on the Arduino.
- **SCL**: Connect to A5 on the Arduino.
- **VCC** (Red wire) -> Arduino VIN

Steps:

1. **Initialization:**
   - The Arduino initializes the RFID reader, I2C LCD, and servo motor.
2. **RFID Tag Detection:**
   - When an RFID tag is brought near the reader, it reads the tag's unique ID.
3. **ID Verification:**
   - The Arduino checks if the ID matches a predefined list of authorized IDs.
4. **Display Status:**
   - The status (lock/unlock) is displayed on the LCD.
5. **Lock/Unlock Door:**
   - If the ID is authorized, the servo motor rotates to unlock the door.
   - If unauthorized, the door remains locked.

This setup provides a secure, automated door lock/unlock system using RFID technology, controlled by an Arduino, with visual feedback through an I2C LCD display.

CODE:

##New Card Scanner

```
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN  10
byte readCard[4];
byte a = 0;

LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  lcd.begin();


  lcd.backlight();
  while (!Serial);
  SPI.begin();
  mfrc522.PCD_Init();
  delay(4);
  mfrc522.PCD_DumpVersionToSerial();
  lcd.setCursor(2, 0);
  lcd.print("Put your card");
}

void loop() {
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return 0;
  }
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return 0;
  }

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanned UID");
  a = 0;
  Serial.println(F("Scanned PICC's UID:"));
  for ( uint8_t i = 0; i < 4; i++) {  //
    readCard[i] = mfrc522.uid.uidByte[i];
```

```
    Serial.print(readCard[i], HEX);
    Serial.print(" ");
    lcd.setCursor(a, 1);
    lcd.print(readCard[i], HEX);
    lcd.print(" ");
    delay(500);
    a += 3;
  }
  Serial.println("");
  mfrc522.PICC_HaltA();
  return 1;
}
```

##Main

```
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
String UID = "F3 A1 01 10";
byte lock = 0;

Servo servo;
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 rfid(SS_PIN, RST_PIN);


void setup() {
  Serial.begin(9600);
  servo.write(120);
  lcd.begin();
  lcd.backlight();
  servo.attach(3);
  SPI.begin();
  rfid.PCD_Init();
}

void loop() {
  lcd.setCursor(4, 0);
  lcd.print("Welcome!");
  lcd.setCursor(1, 1);
```

```
  lcd.print("Put your card");

  if (!rfid.PICC_IsNewCardPresent())
    return;
  if (!rfid.PICC_ReadCardSerial())
    return;

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanning");
  Serial.print("NUID tag is :");
  String ID = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    lcd.print(".");
    ID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
    ID.concat(String(rfid.uid.uidByte[i], HEX));
    delay(300);
  }
  ID.toUpperCase();

  if (ID.substring(1) == UID && lock == 0) {

    servo.write(70);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door is locked");
    delay(1500);
    lcd.clear();
    lock = 1;
  } else if (ID.substring(1) == UID && lock == 1) {
    servo.write(160);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Door is open");
    delay(1500);
    lcd.clear();
    lock = 0;
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Wrong card!");
    delay(1500);
    lcd.clear();
  }
}
```