Department of Computer Science

Lahore Garrison University

Lahore

Internet of Things Project

DIGITAL VOTING SYSTEM

By

S#	Name	Roll#	Section
1.	Imran butt	410	3
2.	Talha Ahmed	438	3

Submitted to:

Sir. Waheed ul Hasan _____ **(**Signature**)**

Introduction

This project presents a smart and secure Digital Voting System based on IoT technologies. The system is designed to modernize the traditional voting process by introducing automation, identity verification, and real-time display of results. It uses components like RFID for voter authentication, PIR sensors to detect user presence, and push buttons for casting votes. The entire system is managed by an Arduino Uno, and the voting outcome is displayed on a 16x2 LCD using the I2C interface to reduce wiring complexity. This project not only enhances voting transparency but also prevents multiple votes from the same user, thereby improving the integrity of the election process.

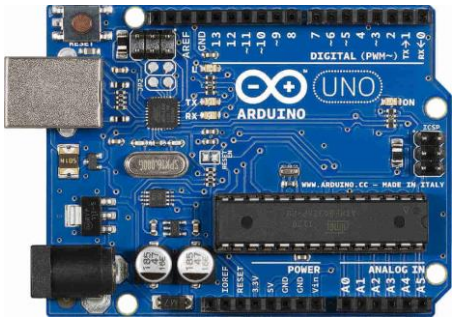
Objectives

The main objective of this project is to design a secure, reliable, and user-friendly digital voting system using IoT principles. It aims to ensure that only authorized individuals are allowed to vote, that each voter can vote only once, and that votes are recorded and displayed in real-time. The project also aims to minimize human errors and reduce the resources required for managing and counting votes manually.

Components

1. **Arduino Uno:**

- **Description:** The main microcontroller board that controls the entire system.
- **Role:** Processes inputs from the motion sensor, RFID, and buttons, and sends output to the LCD. It runs the program to manage the voting logic.
- **Connection:** Powered by 5V and GND from the breadboard, with digital/analog pins connected to other components (e.g., D2 for motion, D10-D13 for RFID).



2. Motion Sensor (IR Obstacle Avoidance Sensor Module)

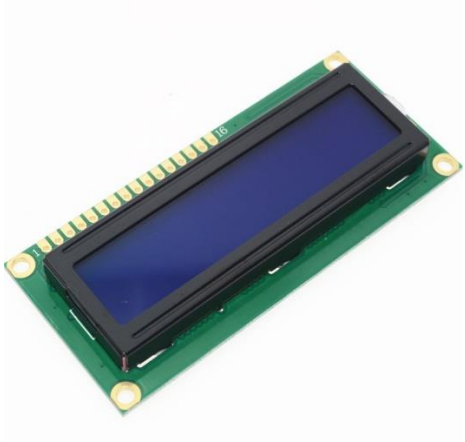
- **Description:** An active IR sensor with an IR LED and photodiode, used to detect objects or presence.
- **Working:** Detects a voter's presence by emitting IR light and sensing its reflection when the voter enters the range (2cm to 30cm, adjustable). It sends a signal to the Arduino to start the voting process.
- **Role:** Serves as the initial trigger to activate the system when a voter is detected.

3. RFID Reader Module (RC522):

- **Description:** A blue module with an antenna coil that reads RFID cards.
- **Role:** Authenticates the voter by reading the RFID card's Unique ID (UID) and sending it to the Arduino for verification.
- **Connection:** VCC to 3.3V/5V, GND to ground, and digital pins (e.g., D10 for SDA, D13 for SCK, D11 for MOSI, D12 for MISO, D9 for RST).

4. LCD Display (16x2) with I2C Module:

- **Description:** A green 16x2 character display with an I2C module attached for easier connection.
- **Role:** Displays messages like “Scan your ID” and “Thanks for voting” to guide the voter.
- **Connection:** VCC to 5V, GND to ground, SDA to Arduino’s A4 pin, and SCL to A5 pin.

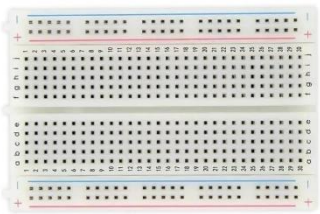


5. Push Buttons:

- **Description:** Small tactile switches mounted on the breadboard.
- **Role:** Allows the voter to cast their vote by pressing the appropriate button, which the Arduino registers.
- **Connection:** One pin to a digital pin (e.g., D3, D4), and the other to GND or 5V with a pull-up/pull-down resistor.

6. Breadboard:

- **Description:** A white perforated board for prototyping the circuit.
- **Role:** Provides a platform to connect all components and wires without soldering.
- Connection: Holds power rails (5V, GND) and component pins with jumper wires.

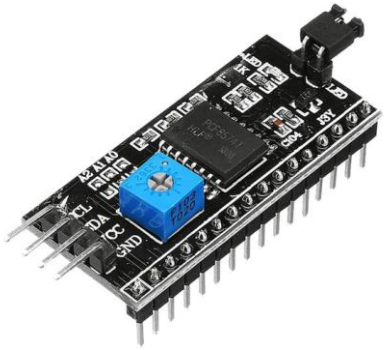


7. Jumper Wires:

- **Description:** Colorful wires used to connect components.
- **Role:** Transfers power (5V, GND) and data signals between the Arduino, breadboard, and other modules.
- **Connection:** Red for 5V, black/brown for GND, and others for digital pins.

8. I2C (Inter-Integrated Circuit):

- **Description**: A communication protocol integrated into the I2C module of the LCD.
- **Role:** Simplifies communication between the Arduino and LCD using only two wires (SDA and SCL), reducing the number of pins needed.
- **Connection**: SDA to A4, SCL to A5 on Arduino, with VCC and GND for power.



Working of the Project

The digital voting system operates as follows:

1. System Activation:

• The IR Obstacle Avoidance Sensor detects a voter when they enter its range (e.g., within 10-30cm), sending a signal to the Arduino to start.

2. ID Scan Prompt:

• The Arduino instructs the LCD (via I2C) to display “Scan your ID”.

3. RFID Authentication:

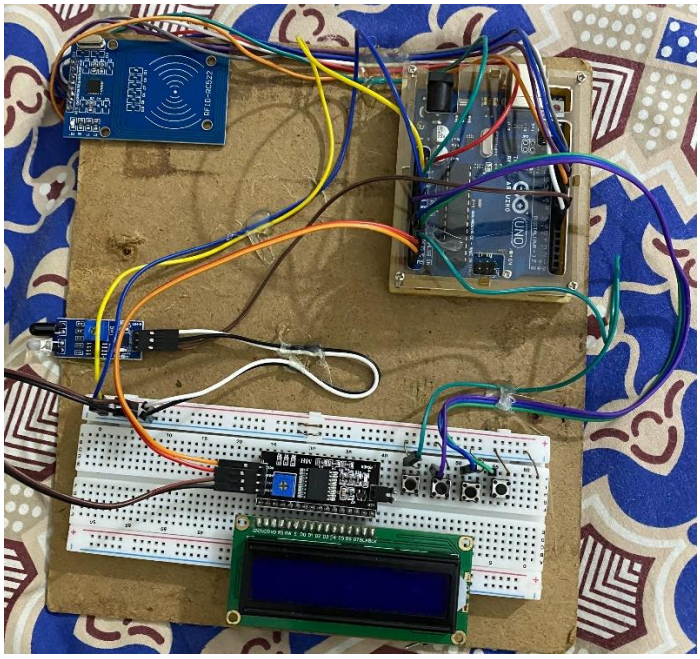
• The RFID Reader Module (RC522) scans the card’s UID, and Arduino verifies it.

4. Voting Process:

• The voter presses a push button to vote, recorded by Arduino.

5. Confirmation:

• The LCD shows “Thanks for voting”, blocking re-voting with the same card.



Tables of Connections:

MFRC522 RFID Reader		
Component	**Arduino Pin**	**Details**
- SDA	6 (SS_PIN)	Slave Select pin
- SCK	13	SPI Clock
- MOSI	11	SPI Master Out Slave In
- MISO	12	SPI Master In Slave Out
- RST	12 (RST_PIN)	Reset pin
- GND	GND	Ground
- VCC	3.3V	Power (Use 3.3V, not 5V, for MFRC522)

LCD I2C Pin	Arduino Pin	Description
VCC	5V	Power supply
GND	GND	Ground
SDA	A4	I2C Data line
SCL	A5	I2C Clock line

Button Function	Arduino Pin	Symbol in Code	Description
Vote for PMLN	A0	`sw1`	Increments `vote1`
Vote for PTI	A1	`sw2`	Increments `vote2`
Vote for TLP	A2	`sw3`	Increments `vote3`
Vote for PPP	A3	`sw4`	Increments `vote4`
Show Result	D4	`sw5`	Displays winning party

PIR Sensor Pin	Arduino Pin	Description
VCC	5V	Power supply
GND	GND	Ground
OUT	D7	`pirPin` - Detects motion (LOW = Motion Detected)

Code Overview

The code begins by initializing all the components, including the RFID module, LCD display (via I2C), PIR sensor, and push buttons. In the main loop, the Arduino continuously checks for motion through the PIR sensor. When motion is detected, it reads the RFID tag. If the tag is valid and not previously used, it waits for a button press. Based on the button pressed, it records the vote, displays confirmation on the LCD, and stores the RFID tag in memory to prevent further voting.

Testing and Results

The system was tested under various conditions. In the first test, an invalid or unauthorized RFID tag was used, and the system correctly denied voting access. In the second test, a valid RFID was scanned, and the system allowed a successful vote, displaying the result on the LCD. In the third test, the same RFID was used again, and the system blocked further voting, proving its efficiency in preventing duplicate votes. The system performed reliably in all scenarios.

Applications

This IoT-based digital voting system can be applied in various settings such as school or college elections, club and society polling, secure feedback systems, or small-scale public elections. It offers a low-cost, efficient, and tamper-proof method for collecting and managing votes.

Future Improvements

In future versions, the system could be enhanced with biometric verification such as fingerprint or facial recognition to further increase security. Data could also be logged to the cloud using Wi-Fi modules like the ESP8266, allowing real-time remote monitoring. Additionally, a buzzer or audio feedback system could be added to improve the user experience, and a more advanced display could be used to show vote counts or percentages.

Conclusion

In conclusion, this Digital Voting System based on IoT is a practical and scalable solution that can revolutionize small to medium-scale elections. By using RFID for voter verification and real-time feedback through an LCD, it improves both the efficiency and security of the voting process. The system eliminates manual counting, prevents unauthorized access, and promotes fair and transparent elections.

Code:

```
#include <Wire.h>

#include <SPI.h>

#include <MFRC522.h>

#include <LiquidCrystal_I2C.h>


#define SS_PIN 6

#define RST_PIN 12

#define sw1 A0

#define sw2 A1

#define sw3 A2

#define sw4 A3

#define sw5 4

#define pirPin 7


int vote1 = 0, vote2 = 0, vote3 = 0, vote4 = 0;

bool scanned = false;


MFRC522 mfrc522(SS_PIN, RST_PIN);

LiquidCrystal_I2C lcd(0x27, 16, 2);


void setup() {

  Wire.begin();
```

```
  SPI.begin();
  mfrc522.PCD_Init();
  lcd.begin(16, 2);

  lcd.backlight(); // Turn on LCD light
  lcd.clear();
  lcd.print("System Ready");
  delay(5000);      // Show for 5 seconds
  lcd.noBacklight(); // Turn off backlight

  pinMode(sw1, INPUT_PULLUP);
  pinMode(sw2, INPUT_PULLUP);
  pinMode(sw3, INPUT_PULLUP);
  pinMode(sw4, INPUT_PULLUP);
  pinMode(sw5, INPUT_PULLUP);
  pinMode(pirPin, INPUT);

  Serial.begin(9600);
}

void loop() {
  int motionDetected = digitalRead(pirPin);
```

```
if (motionDetected == LOW && !scanned) {
  lcd.backlight(); // Turn on LCD when motion is detected

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Motion Detected");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scan your ID...");

  while (!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
    // Wait for RFID scan
  }

  Serial.print("UID: ");
  String content = "";
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
```

```
  Serial.println();


  content.toUpperCase();
  if (content.substring(1) == "99 3E 3D 02") { // Replace with your UID
    lcd.clear();
    lcd.print("Authorized Access");
    delay(2000);
    scanned = true;
    lcd.clear();
    lcd.print("PMLN PTI TLP PPP");
  } else {
    lcd.clear();
    lcd.print("Access Denied");
    delay(3000);
    lcd.clear();
    lcd.noBacklight(); // Turn off LCD if unauthorized
    scanned = false;
  }
}
else if (motionDetected == HIGH && !scanned) {
  lcd.setCursor(0, 0);
  lcd.print("Waiting for ");
  lcd.setCursor(0, 1);
```

```
  lcd.print("motion...");
  delay(500);
 }

if (scanned) {
  if (digitalRead(sw1) == LOW) {
    vote1++;
    thankYou();
    scanned = false;
  } else if (digitalRead(sw2) == LOW) {
    vote2++;
    thankYou();
    scanned = false;
  } else if (digitalRead(sw3) == LOW) {
    vote3++;
    thankYou();
    scanned = false;
  } else if (digitalRead(sw4) == LOW) {
    vote4++;
    thankYou();
    scanned = false;
  } else if (digitalRead(sw5) == LOW) {
    showResult();
```

```
      scanned = false;
    }
  }
}

void thankYou() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Thanks for");
  lcd.setCursor(0, 1);
  lcd.print("voting!");
  delay(3000);
  lcd.clear();
  lcd.noBacklight(); // Turn off LCD after thank you
}

void showResult() {
  lcd.clear();
  int total = vote1 + vote2 + vote3 + vote4;

  if (total == 0) {
    lcd.print("No votes yet");
  } else {
```

```
    if (vote1 > vote2 && vote1 > vote3 && vote1 > vote4) lcd.print("PMLN Wins");
    else if (vote2 > vote1 && vote2 > vote3 && vote2 > vote4) lcd.print("PTI Wins");
    else if (vote3 > vote1 && vote3 > vote2 && vote3 > vote4) lcd.print("TLP Wins");
    else if (vote4 > vote1 && vote4 > vote2 && vote4 > vote3) lcd.print("PPP Wins");
    else lcd.print("Tie / No winner");
  }
  delay(2000);
  lcd.clear();
  lcd.noBacklight(); // Turn off LCD after showing result
}
```

this code