

CREATIVE EVM [ELECTRONIC VOTING MACHINE] MINI-PROJECT

MEET OUR TEAM

**CHAKSHU
BANSAL
22104031**

**DEVANSH
22104035**

**HIMANI
VERMA
22104046**

**KAJAL
KUMARI
22104056**

OUR PROJECT



1. Definition: Electronic Voting Machine (EVM)

An EVM is an electronic device used for casting votes in an election. It replaces traditional paper ballots with electronic ballots and is designed to be secure, efficient, and user-friendly.

2. Importance of EVMs in modern elections:

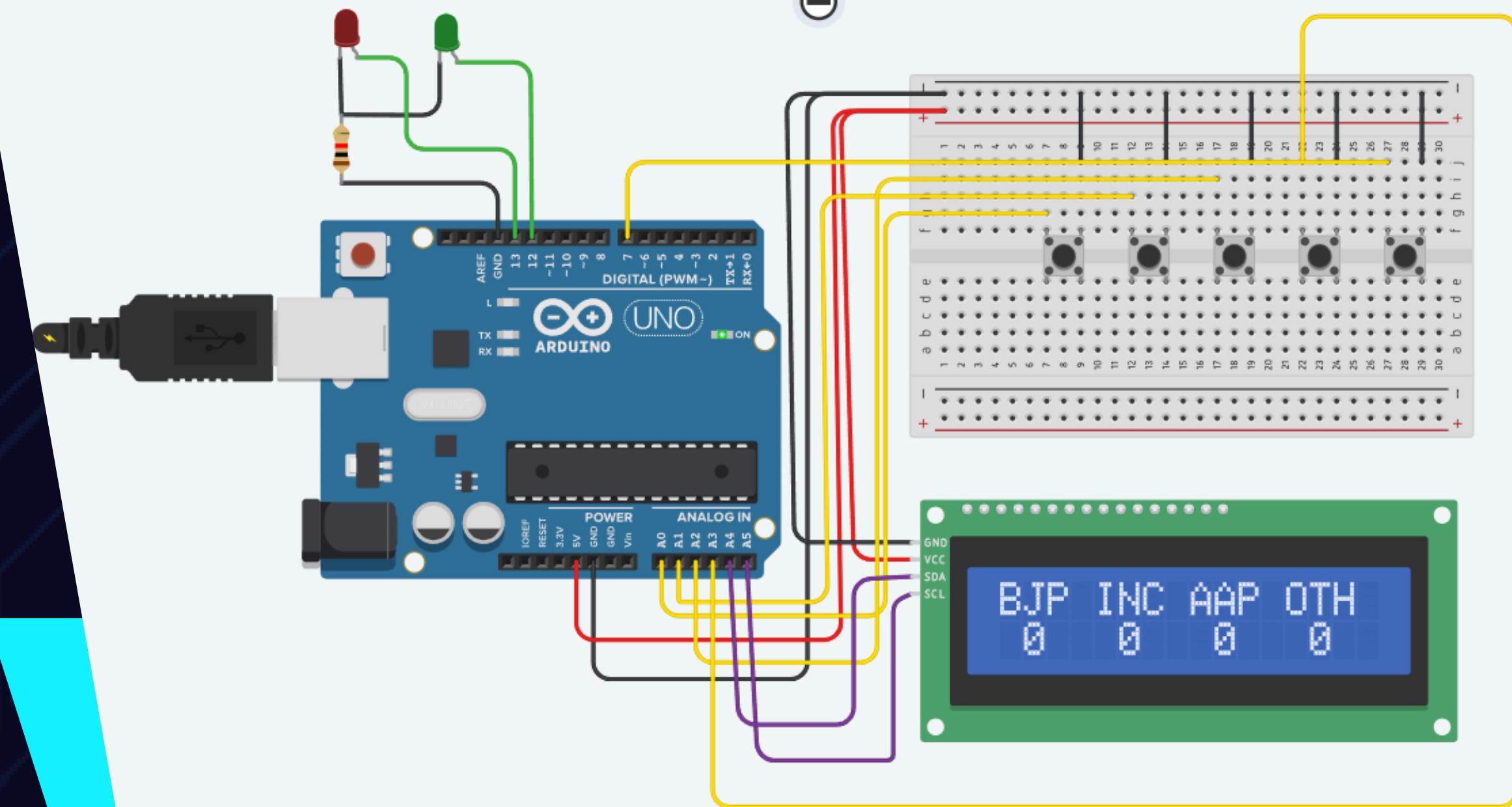
Enhances the speed and accuracy of vote counting
Reduces the likelihood of human errors and fraud
Increases voter turnout by providing a convenient voting experience.

3. Brief history and evolution of EVMs:

Introduction of EVMs in the 1960s
Advancements in technology leading to the development of more sophisticated EVMs

ABOUT TECHNICAL

EVM (VOTING MACHINE)



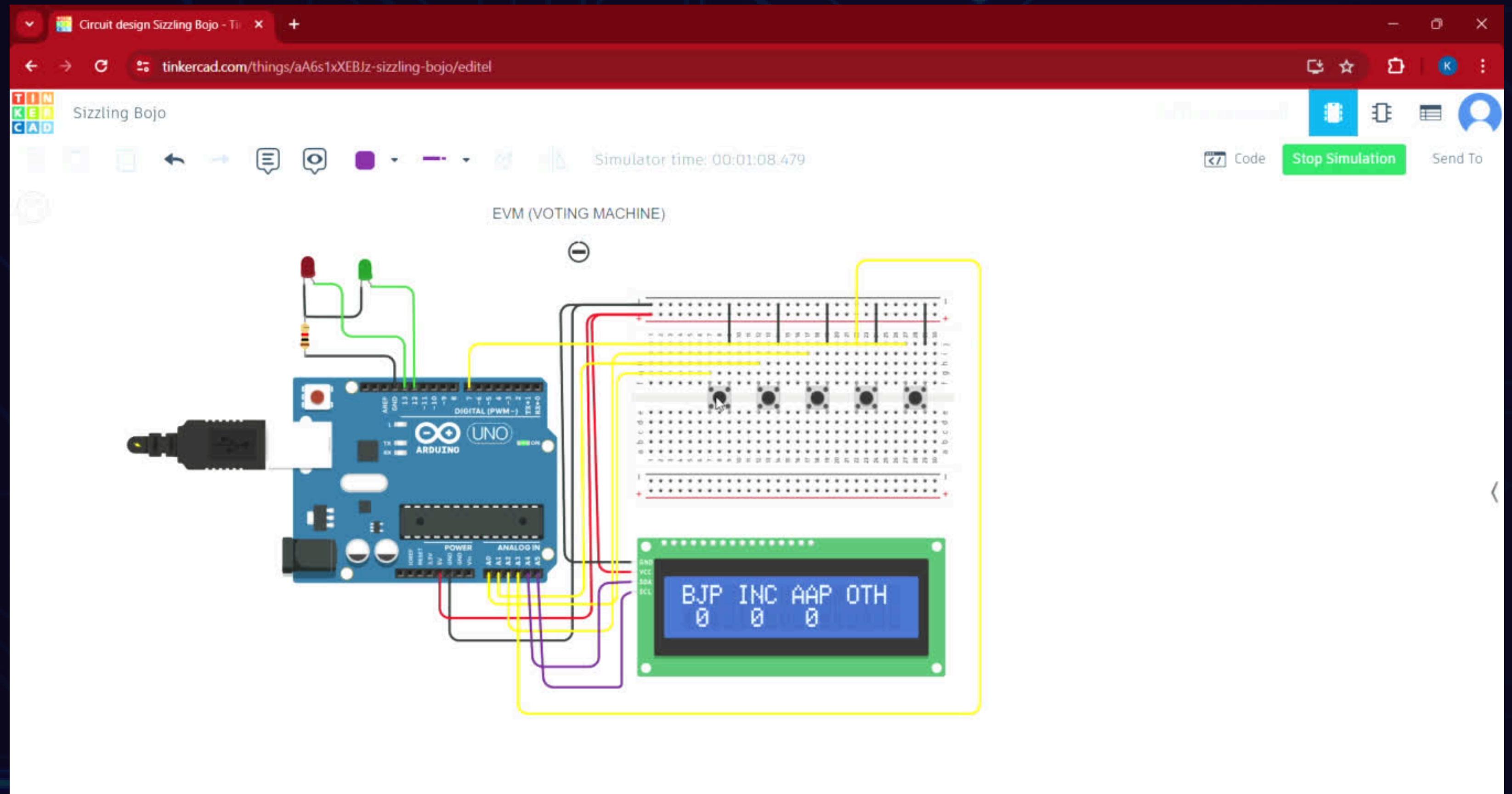
OUR OBJECTIVE

01

- Purpose of the project:
 - To design and simulate an EVM on Tinkercad
 - Gain hands-on experience in circuit design and programming
 - Understand the functionality of an electronic voting system
- Learning outcomes:
 - Understanding of circuit design, simulation, and functionality of an EVM
 - Learn how to design and simulate circuits using Tinkercad
 - Gain proficiency in programming Arduino microcontrollers
 - Understand the components and operation of an electronic voting machine

[LINK FOR TINKERCAD](#)
[SIMULATION](#)

DEMO VIDEO OF WORKING PROJECT [EVM]



ARDUINO UNO CODE FOR CREATING EVM

The Arduino IDE interface shows the code for the first half of the EVM. The code includes the Adafruit LiquidCrystal library, pin definitions for five buttons (sw1 to sw5) and two LEDs (pin 13 for Red LED, pin 12 for Green LED), and the setup function which initializes the pins as inputs and sets the LCD to 16x2 characters.

```
1 // Include Library
2 // https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library
3
4 #include <Adafruit_LiquidCrystal.h>
5
6 Adafruit_LiquidCrystal lcd_1(0);
7 #define sw1 A0 // Button 1
8 #define sw2 A1 // Button 2
9 #define sw3 A2 // Button 3
10 #define sw4 A3 // Button 4
11 #define sw5 7 // Button 5 for result
12 int vote1=0;
13 int vote2=0;
14 int vote3=0;
15 int vote4=0;
16 void setup()
17 {
18 pinMode(sw1, INPUT);
19 pinMode(sw2, INPUT);
20 pinMode(sw3, INPUT);
21 pinMode(sw4, INPUT);
22 pinMode(sw5, INPUT);
23 pinMode(13, OUTPUT); // Red LED
24 pinMode(12, OUTPUT); // Green LED
```

The Arduino IDE interface shows the code for the second half of the EVM. This part handles button presses, updates the LCD display, and controls two LEDs (Red and Green). It prints "EVM" and "Circuit design" to the LCD, then displays "BJP" and "INC".

```
18 pinMode(sw1, INPUT);
19 pinMode(sw2, INPUT);
20 pinMode(sw3, INPUT);
21 pinMode(sw4, INPUT);
22 pinMode(sw5, INPUT);
23 pinMode(13, OUTPUT); // Red LED
24 pinMode(12, OUTPUT); // Green LED
25
26 lcd_1.begin(16, 2);
27 lcd_1.setCursor(0,0);
28 lcd_1.print("    EVM ");
29 lcd_1.setCursor(0,1);
30 lcd_1.print("Circuit design ");
31 delay(3000);
32 digitalWrite(sw1, HIGH);
33 digitalWrite(sw2, HIGH);
34 digitalWrite(sw3, HIGH);
35 digitalWrite(sw4, HIGH);
36 digitalWrite(sw5, HIGH);
37 lcd_1.clear();
38 lcd_1.setCursor(0,0);
39 lcd_1.print("BJP");
40 lcd_1.setCursor(4,0);
41 lcd_1.print("INC");
```

For full code

ARDUINO UNO CODE FOR CREATING EVM



Text ▼ ▼ ▼ 1 (Arduino Uno R3) ▼

```
65 if(digitalRead(sw1)==0)
66 {
67   vote1++;
68   digitalWrite(12,HIGH);
69   delay(500);
70   while(digitalRead(sw1)==0);
71   digitalWrite(12,LOW);
72   delay(1000);
73 }
74
75 if(digitalRead(sw2)==0)
76 {
77   vote2++;
78   digitalWrite(12,HIGH);
79   delay(500);
80   while(digitalRead(sw2)==0);
81   digitalWrite(12,LOW);
82   delay(1000);
83 }
84
85 }
```

Serial Monitor

Text ▼ ▼ ▼ 1 (Arduino Uno R3) ▼

```
89 if(digitalRead(sw3)==0)
90 {
91   vote3++;
92   digitalWrite(12,HIGH);
93   delay(500);
94   while(digitalRead(sw3)==0);
95   digitalWrite(12,LOW);
96   delay(1000);
97 }
98
99 if(digitalRead(sw4)==0)
100 {
101   vote4++;
102   digitalWrite(12,HIGH);
103   delay(500);
104   while(digitalRead(sw4)==0);
105   digitalWrite(12,LOW);
106   delay(1000 );
107 }
108
109 if(digitalRead(sw5)==0)
110 {
111 }
```

Serial Monitor

[For full code](#)

ARDUINO UNO CODE FOR CREATING EVM



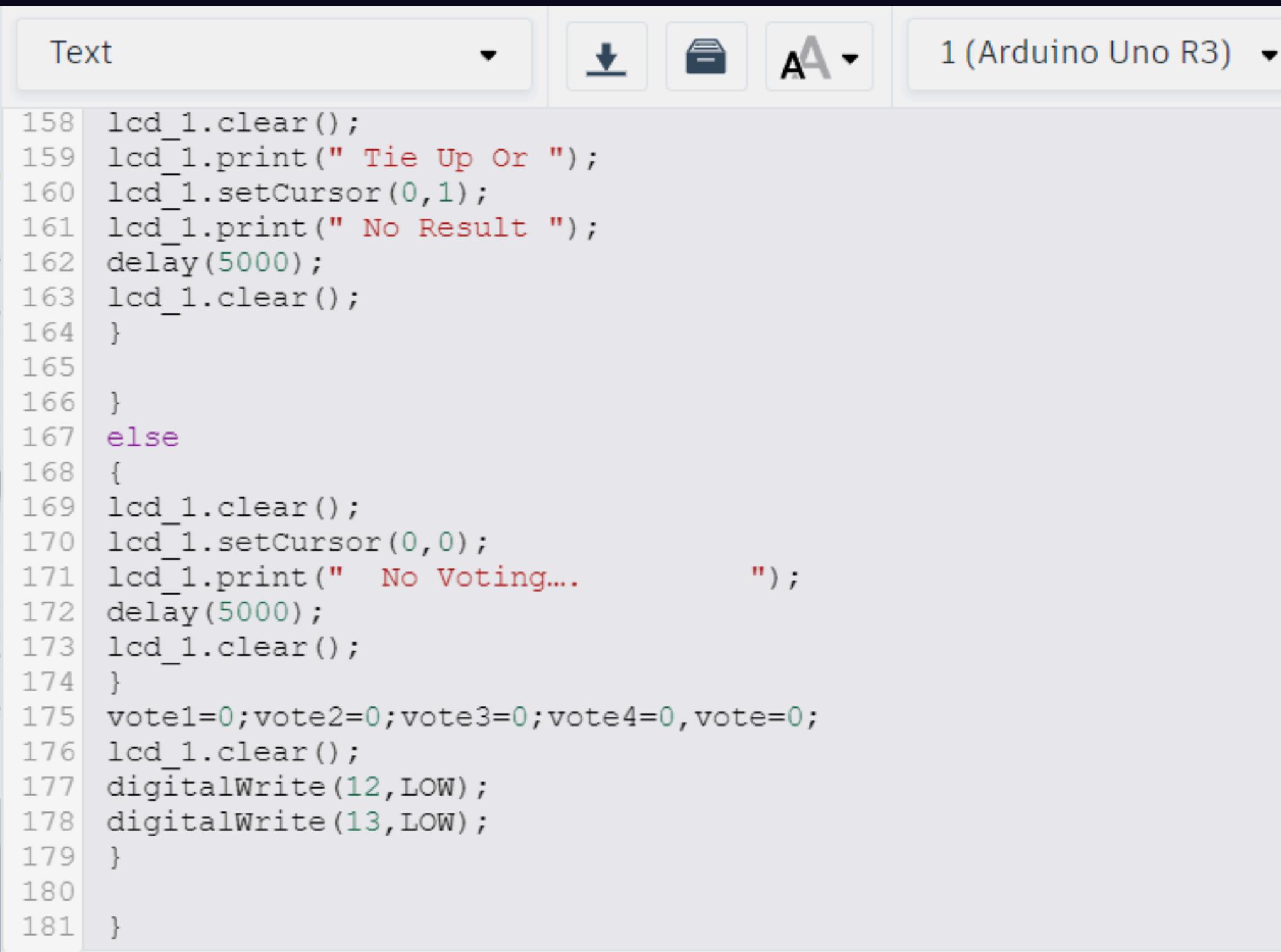
```
Text ▾    1 (Arduino Uno R3) ▾  
112 {  
113 digitalWrite(13,HIGH);  
114 int vote=vote1+vote2+vote3+vote4;  
115 if(vote)  
116 {  
117 if((vote1 > vote2 && vote1 > vote3 && vote1 > vote4))  
118 {  
119 lcd_1.clear();  
120 lcd_1.print("BJP Wins");  
121 delay(5000);  
122 lcd_1.clear();  
123 }  
124 else if((vote2 > vote1 && vote2 > vote3 && vote2 > vote4))  
125 {  
126 lcd_1.clear();  
127 lcd_1.print("INC Wins");  
128 delay(5000);  
129 lcd_1.clear();  
130 }  
131 else if((vote3 > vote1 && vote3 > vote2 && vote3 > vote4))  
132 {  
133 lcd_1.clear();  
134 lcd_1.print("AAP Wins");  
135 delay(5000);  
Serial Monitor
```

```
Text ▾    1 (Arduino Uno R3) ▾  
135 delay(5000);  
136 lcd_1.clear();  
137 }  
138 else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)  
139 {  
140 lcd_1.setCursor(0,0);  
141 lcd_1.clear();  
142 lcd_1.print("OTH Wins");  
143 delay(5000);  
144 lcd_1.clear();  
145 }  
146  
147 else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)  
148 {  
149 lcd_1.setCursor(0,0);  
150 lcd_1.clear();  
151 lcd_1.print("OTH Wins");  
152 delay(2000);  
153 lcd_1.clear();  
154 }  
155  
156 else  
157 {  
158 lcd_1.clear();  
Serial Monitor
```

[For full code](#)

ARDUINO UNO CODE FOR CREATING EVM

>>>



The image shows a screenshot of the Arduino IDE interface. At the top, there's a toolbar with a 'Text' dropdown, a download icon, a sketch folder icon, a font size dropdown, and a board dropdown set to '1 (Arduino Uno R3)'. Below the toolbar is a code editor containing the following Arduino code:

```
158 lcd_1.clear();
159 lcd_1.print(" Tie Up Or ");
160 lcd_1.setCursor(0,1);
161 lcd_1.print(" No Result ");
162 delay(5000);
163 lcd_1.clear();
164 }
165
166 }
167 else
168 {
169 lcd_1.clear();
170 lcd_1.setCursor(0,0);
171 lcd_1.print(" No Voting....      ");
172 delay(5000);
173 lcd_1.clear();
174 }
175 vote1=0;vote2=0;vote3=0;vote4=0,vote=0;
176 lcd_1.clear();
177 digitalWrite(12,LOW);
178 digitalWrite(13,LOW);
179 }
180 }
181 }
```

[For full code](#)

COMPONENTS NEEDED

List of components required for the project:

- Arduino Uno: Microcontroller board used as the brain of the EVM.
- Pushbuttons for candidate selection: Input devices for voters to make their choices
- LED display for vote count:
Output device to display the total number of votes.
- Resistors, wires, breadboard, etc.:
Additional components for circuit connections and stability

COMPONENTS NEEDED

Name	Quantity	Component
S1		
S2		
S3	5	Pushbutton
S4		
S5		
U1	1	Arduino Uno R3
D1	1	Red LED
D2	1	Green LED
R1	1	1 kΩ Resistor
U2	1	MCP23008-based, 32 (0x20) LCD 16 x 2 (I2C)

CONCLUSIONS



In summary, our exploration of building an Electronic Voting Machine (EVM) on Tinkercad has been a rich learning experience.

We've gained hands-on knowledge in circuit design, programming, and the functioning of an EVM. This project underscores the value of practical, project-based learning in STEM education and opens doors for further innovation. Our journey doesn't end here; it's a stepping stone for future endeavors, inspiring us to continue exploring and contributing to technology and beyond.

THANK YOU