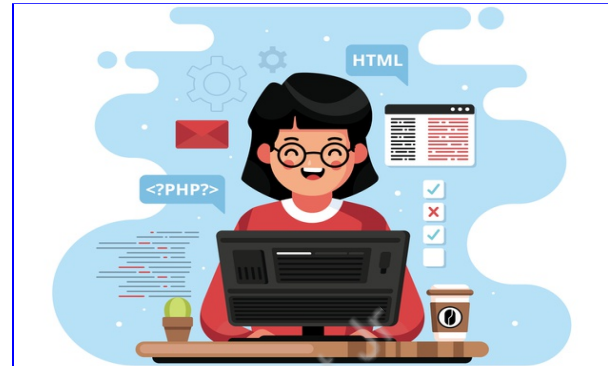


Electronic Voting Machine-2



What is our GOAL for this CLASS?

In this class, we learned **how to design electronic voting Machine on OLED and how calculations and results can be displayed on OLED by Arduino Programming**

What did we ACHIEVE in the class TODAY?

- We learned Algorithm of EVM
- We learned how to display result

Which CONCEPTS/ CODING BLOCKS did we cover today?

- We learned about algorithm of the EVM machine, and then we will display results at last after writing the calculation algorithm.
- Logic of the Electronic Voting Machine
 - When we press the button which is already in the HIGH stage, i.e. it's already receiving 5 volts from ESP32. But as soon as the push-button is pressed the **HIGH** state convert into **the LOW** state because it will break the contact.
 - As when the button state becomes LOW it will increment the vote count of a particular party
 - Start with push button 1 i.e. for the S1 button.
 - To read the stage of a button we have a function called **digitalRead()**

How did we DO the activities?

1. Gather the material from the IoT kit Collect the material
 - 1 x ESP32
 - 1 x USB Cable
 - 4 x Breadboard
 - 2 x Jumper wires
 - 1 x OLED
 - 5 x PushButtons
 - 15 x Jumper Wires -9 Female to Male, 6 Male to Male
2. Do connections:
 - The circuit of this project consists of an ESP32 Controller, pushbuttons, and an OLED screen. Complete processes are controlled by ESP32 Controller, including reading buttons, incrementing vote values, generating results, and sending votes and results to an OLED.
 - Take five buttons in which the first button is for A Party second for B Party third is for C Party fourth is for D Party others and the last button is used for calculating or displaying results.
 - Take **ESP32** from the kit.
 - Take five pushbuttons
 - Insert five push buttons on the breadboard one by one keep spacing between buttons.
 - The five push button's **positive parts** are directly connected with ESP32 **pin no. 13, 33, 14, 27, 26** respectively.
 - Supply negative supply from ESP32 GND to the power rail of the breadboard. Take one **Female to Male** jumper wire, Insert the **Female TERMINAL** into the **ESP32 GND** terminal and drag it to the breadboard and insert **MALE TERMINAL** into the **negative terminal** of the Power rail.
 - The **negative part** of all the **pushbuttons** will be directly connected to the **negative power rail** of the **breadboard**.
 - Take an **OLED** from the kit.
 - Take **Female to Male** jumper wire. Use **Female part at ESP32 terminal** and **Male part** on the breadboard
 - Connect OLED PIN VCC to ESP32 PIN 3.3V
 - Connect OLED PIN GND to ESP32 PIN GND
 - Connect OLED PIN SCL to ESP32 PIN GPIO22
 - Connect OLED PIN SDA to ESP32 PIN GPIO21
3. To read the stage of a button we have a function called **digitalRead()**
 - **digitalRead(S1)** is equal to **LOW**, then increment the value of vote1
 - **++** is used to add the value in variable **vote1**.

- **Serial.print** is used to print the vote number.
- **oled.clearDisplay()** is used to clear the display.

```
if (digitalRead(S1) == LOW)
{
    vote1++;
    Serial.print(vote1);
    delay(500);
    oled.clearDisplay();
}
```

4. Repeat the same step for start push button 2 i.e. for the S2 button.
- To read the stage of a button we have a function called **digitalRead()**
 - **digitalRead(S2)** is equal to **LOW**, then increment the value of vote1
 - ++ is used to add the value in variable **vote2**.
 - **Serial.print** is used to print the vote number.
 - **oled.clearDisplay()** is used to clear the display.

```
if (digitalRead(S2) == LOW)
{
    vote2++;
    Serial.print(vote2);
    delay(500);
    oled.clearDisplay();
}
```

5. Repeat the same step for start push button 3 i.e. for the S3 button.
- To read the stage of a button we have a function called **digitalRead()**
 - **digitalRead(S3)** is equal to **LOW**, then increment the value of vote1
 - ++ is used to add the value in variable **vote3**.
 - **Serial.print** is used to print the vote number.
 - **oled.clearDisplay()** is used to clear the display.

```
if (digitalRead(S3) == LOW)
{
    vote3++;
    Serial.print(vote3);
    delay(500);
    oled.clearDisplay();
}
```

6. Repeat the same step for start push button 4 i.e. for the S3 button.
- To read the stage of a button we have a function called **digitalRead()**
 - **digitalRead(S4)** is equal to **LOW**, then increment the value of vote1
 - ++ is used to add the value in variable **vote4**.
 - **Serial.print** is used to print the vote number.
 - **oled.clearDisplay()** is used to clear the display.

```
if (digitalRead(S4) == LOW)
{
    vote4++;
    Serial.print(vote4);
    delay(500);
    oled.clearDisplay();
}
```

7. Do the calculations and display the results.

8. Gather the material from the IoT kit:

- 1 x ESP32
- 1 x USB Cable
- 4 x Breadboard
- 1 x OLED
- 5 x Push Buttons
- 15 x Jumper Wires -9 Female to Male, 6 Male to Male

9. Do connections:

- The circuit of this project consists of an ESP32 Controller, pushbuttons, and an OLED screen. Complete processes are controlled by ESP32 Controller, including reading buttons, incrementing vote values, generating results, and sending votes and results to an OLED.
- Take five buttons in which the first button is for A Party second for B Party third is for C Party fourth is for D Party others and the last button is used for calculating or displaying results.
- Take **ESP32** from the kit.
- Take five pushbuttons
- Insert five push buttons on the breadboard one by one keep spacing between buttons.
- The five push button's **positive parts** are directly connected with ESP32 **pin no. 13, 33, 14, 27, 26** respectively.
- Supply negative supply from ESP32 GND to the power rail of the breadboard. Take one **Female to Male** jumper wire, Insert the **Female TERMINAL** into the **ESP32 GND** terminal and drag it to the breadboard and insert **MALE TERMINAL** into the **negative terminal** of the Power rail.
- The **negative part** of all the **pushbuttons** will be directly connected to the **negative power rail** of the **breadboard**.

- Take an **OLED** from the kit.
- Take **Female to Male** jumper wire. Use **Female part at ESP32 terminal** and **Male part** on the breadboard
- Press the last button i.e **S5** use to calculate the result.
- On pressing the S5 button it will sum all the votes of the all parties like **Party A, Party B, Party C, Party D** . To save all the votes we will use another variable i.e **vote**.
- **Serial.print(vote)** will display the total count of votes.
- Set up a delay of **500ms** to display the total.

```
if (digitalRead(S5) == 0)
{

    int vote = vote1 + vote2 + vote3 + vote4;
    Serial.print(vote);
    delay(500);
}
```

10. Display the winner.

- Compare the total vote count with all the party votes.
- Start with **vote 1**, **vote 1** will compare with **vote 2** , **vote 3**, and **vote 4**.
- Use **And** condition to compare the vote results. If all **&&(and)** conditions should be true on comparing with vote1 then display **A is winner**
- To **write** text, set the font size, color, and location where the text will be displayed in the OLED and data which need to be printed.
 - Set the font size using the **setTextSize()** method.
 - Set the font color using the **setTextColor()** method.**WHITE** sets white font and black background.
 - Using the **setCursor(x,y)** method, specify the starting point of the text.
 - As a final step, display data using the **println()** method.Print (**"A"**). Here **A** refers to **Party A** and print the total **vote** for **party A**.
 - At last, we need to call the **display()** method to actually display the text on the screen.

```

if (vote > 0)
{
    if ((vote1 > vote2 && vote1 > vote3 && vote1 > vote4))
    {
        oled.clearDisplay();
        delay(1000);
        oled.setTextSize(1);
        oled.setTextColor(WHITE);
        oled.setCursor(1, 0);
        oled.print("A is Winner");
        oled.setCursor(1, 100);
        oled.print(vote1);
        oled.display();
        delay(5000);
    }
}

```

- Use **And** condition to compare the vote results. If all **&&(and)** conditions should be true on comparing with vote1 then display **B is winner**
- To **write** text, set the font size, color, and location where the text will be displayed in the OLED and data which need to be printed.
 - Set the font size using the **setFontSize()** method.
 - Set the font color using the **setTextColor()** method. **WHITE** sets white font and black background.
 - Using the **setCursor(x,y)** method, specify the starting point of the text.
 - As a final step, display data using the **println()** method. Print ("**B**"). Here **B** refers to **Party B** and print the total **vote** for **party B**.
 - At last, we need to call the **display()** method to actually display the text on the screen.

```

else if ((vote2 > vote1 && vote2 > vote3 && vote2 > vote4))
{
    oled.clearDisplay();
    delay(1000);
    oled.setTextSize(1);
    oled.setTextColor(WHITE);
    oled.setCursor(1, 0);
    oled.print("B is Winner");
    oled.setCursor(1, 100);
    oled.print(vote2);
    oled.display();
    delay(5000);
}

```

- Use **And** condition to compare the vote results. If all **&&(and)** conditions should be true on comparing with vote1 then display **C is winner**
- To **write** text, set the font size, color, and location where the text will be displayed in the OLED and data which need to be printed.

- Set the font size using the **setTextSize()** method.
- Set the font color using the **setTextColor()** method. **WHITE** sets white font and black background.
- Using the **setCursor(x,y)** method, specify the starting point of the text.
- As a final step, display data using the **println()** method. Print ("**C**"). Here **C** refers to **Party C** and print the total **vote** for **party C**.
- At last, we need to call the **display()** method to actually display the text on the screen

```

else if ((vote3 > vote1 && vote3 > vote2 && vote3 > vote4))
{
    oled.clearDisplay();
    delay(1000);
    oled.setTextSize(1);
    oled.setTextColor(WHITE);
    oled.setCursor(1, 0);
    oled.print("C is Winner");
    oled.setCursor(1, 100);
    oled.print(vote1);
    oled.display();
    delay(5000);
}

```

- Use **And** condition to compare the vote results. If all **&&(and)** conditions should be true on comparing with vote1 then display **D is winner**
- To **write** text, set the font size, color, and location where the text will be displayed in the OLED and data which need to be printed.
 - Set the font size using the **setTextSize()** method.
 - Set the font color using the **setTextColor()** method. **WHITE** sets white font and black background.
 - Using the **setCursor(x,y)** method, specify the starting point of the text.
 - As a final step, display data using the **println()** method. Print ("**D**"). Here **D** refers to **Party D** and print the total **vote** for **party D**.
 - At last, we need to call the **display()** method to actually display the text on the screen

```
else if (vote4 > vote1 && vote4 > vote2 && vote4 > vote3)
{
    oled.clearDisplay();
    delay(1000);
    oled.setTextSize(1);
    oled.setTextColor(WHITE);
    oled.setCursor(1, 0);
    oled.print("D is Winner");
    oled.setCursor(1, 100);
    oled.print(vote1);
    oled.display();
    delay(5000);
}
```

- Sometimes conditions can come where all the parties get the same number of votes.
- So that time there should be conditions where we display **tie** or **no result**.

```
else
{
    oled.clearDisplay();
    delay(1000);
    oled.setTextSize(1);
    oled.setTextColor(WHITE);
    oled.setCursor(1, 0);
    oled.print(" Tie Up Or ");
    oled.setCursor(1, 100);
    oled.print(" No Result ");
    oled.print(vote1);
    oled.display();
    delay(3000);
    oled.clearDisplay();
}
```

- If there is a condition where no party gets any vote then print **No Voting**


```
else
{
    oled.clearDisplay();
    oled.print("No Voting....");
    delay(3000);
    oled.clearDisplay();
}
vote1 = 0; vote2 = 0; vote3 = 0; vote4 = 0, vote = 0;
oled.clearDisplay();
}

}
```

11. Output:

- Compile and upload the program to the ESP32 board using Arduino IDE
- Verify the program by clicking the Tick option.
- Upload the program by clicking the arrow option.

What's NEXT?

In the **next class**, we will learn about **web servers**.

Expand Your Knowledge

To know more about **different categories of OLED** [click here](#).