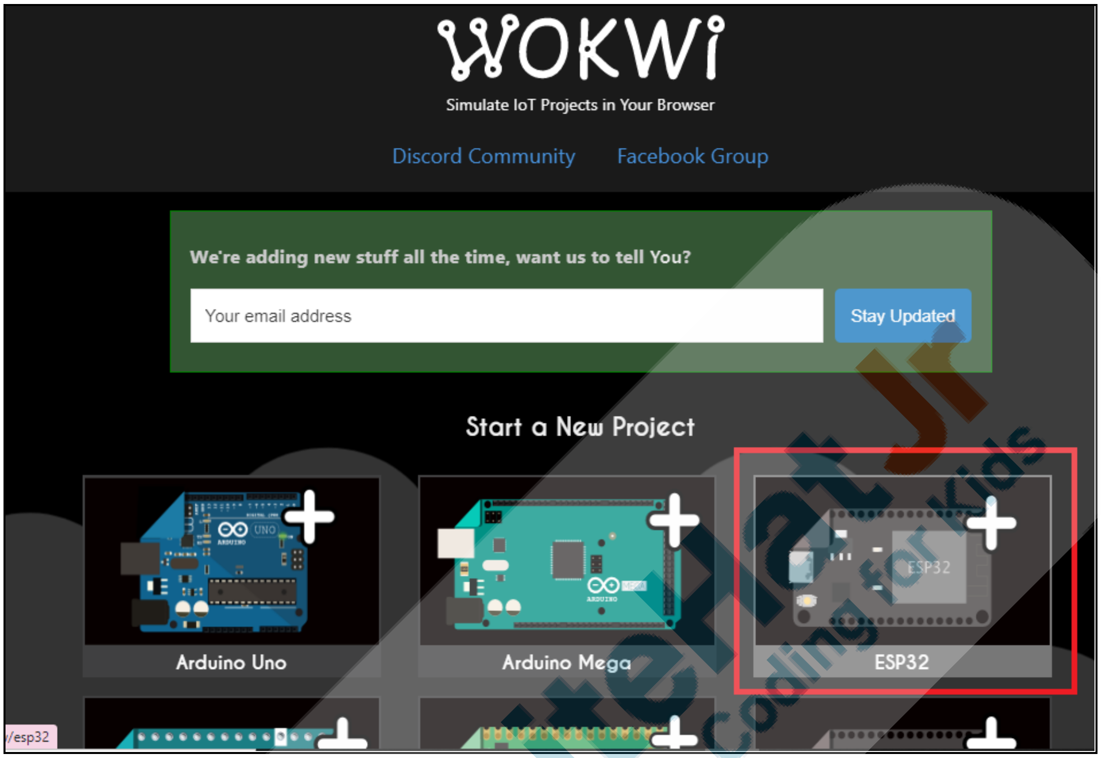
Topic	ELECTRONIC VOTING MACHINE -1	
Class Description	Students will learn to design an electronic voting machine on an OLED display. They will also learn to program a push button easily using a library.	
Class	PRO C257	
Class time	50 mins	
Goal	● Design EVM on an OLED display ● Display data on OLED display	
Resources Required	● Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone ● Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	Warm-Up Teacher-Led Activity-1 Student-Led Activity- 2 Wrap-Up	10 mins 15 mins 15 mins 10 mins

WARM-UP SESSION - 10 mins

Teacher Action	Student Action
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?	**ESR**: Hi, thanks! Yes, I am excited about it!

Following are the WARM-UP session deliverables: ● Greet the student. ● Revision of previous class activities. ● Quizzes.	Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
Activity Details **Following are the session deliverables:** ● Appreciate the student. ● Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	
TEACHER -LED ACTIVITY-1 - 15mins	
Student Initiates Screen Share	
● **OLED setup**	
Teacher Action	**Student Action**
In the last class, we learned about **TONE GENERATORS.** *If the student has any doubt, clarify the doubts* Are you interested in politics and want to become a leader? How do people choose their minister or leader? Right, Voting needs to be done to choose our favorite one.	**ESR:** varied. **ESR:** By Voting

But do you know how real voting can be done?	**ESR:** With the help of a Voting Machine
Are you allowed to vote? No, because you must be 18 to cast the vote.	**ESR:** No!
But you must be curious to cast your first vote? No worries at all! As you are so interested in voting, we can develop our own voting machine. Now you must be wondering what ma'am is saying? How is it possible to develop your own voting machine? Indeed, it's possible. So today, we will develop an electronic voting machine. *Teacher opens the* *Teacher Activity 1*. We will work with ESP 32. So, under **Start a New Project,** click **ESP 32.**	**ESR:** Yes!



WOKWi

Simulate IoT Projects in Your Browser

We're adding new stuff all the time, want us to tell You?

Start a New Project

Step -1: Select the material from the Simulator

(Click on + Sign to add a new part.)

- 1 x **ESP32** (It will be already added in the simulator)
- 1 x **SSD1306 OLED display**
- 5 x **Pushbuttons** (4 push buttons for different vote parties and 1 push button for Result.)

Step -2: Let's 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.

Here we are taking five buttons where -

- the first button is for A Party,
- the second for B Party,
- the third is for C Party,
- the fourth is for D Party and
- the last button is used for calculating or displaying results.

Once we have added 5 push buttons, connect the button to the **ESP 32** board.

- We will connect the push button's **1.l** pin directly to a digital pin. In this case,

Button id	**ESP 32 pin**
btn1	13
btn2	33
btn3	14
btn4	27
btn5	26

- Then, connect the **2.r** pin to the **GND**. Change the color of the wires connected to **GND** to black.

  Click on the wire to change its color.

  Watch this reference video to learn how to change the color of the wires.

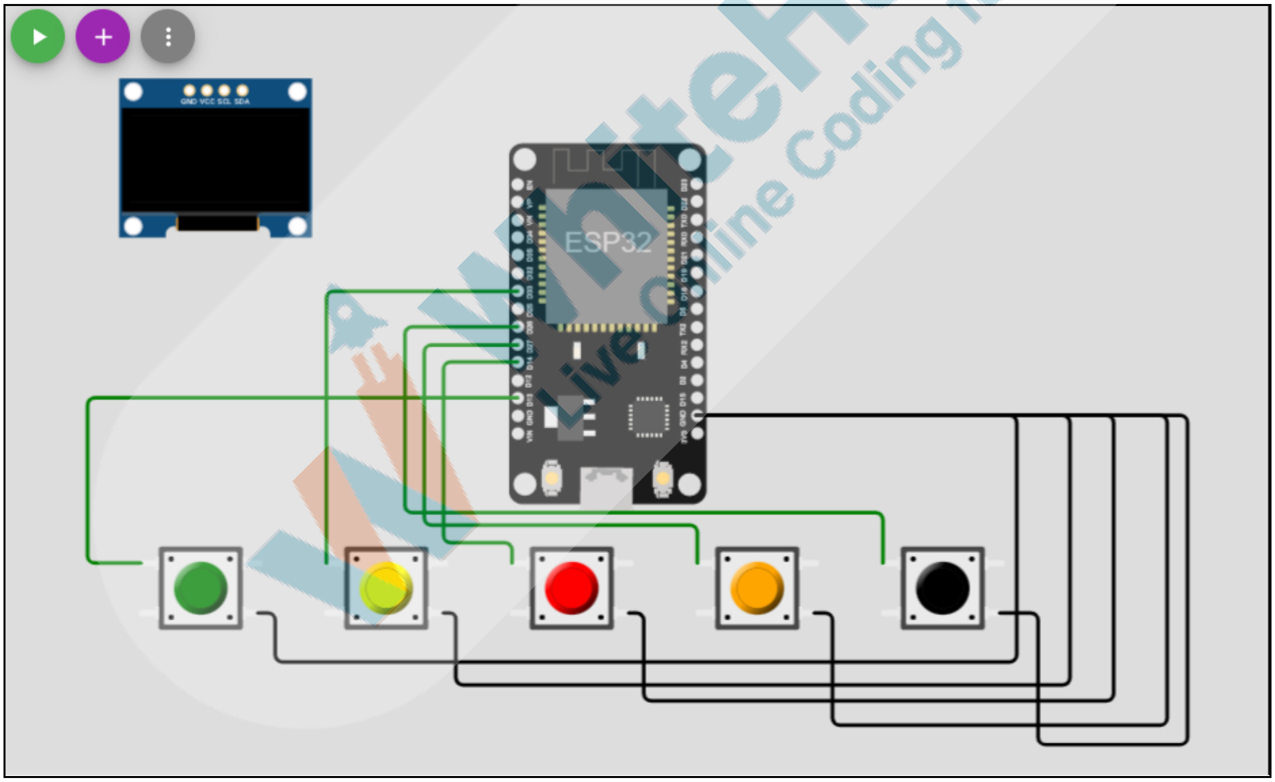
- Click the push buttons and change the color of the buttons as following-

  Click on the button to change its color.

  Watch this reference video to learn how to change the color of the buttons.

Button id	Color
btn1	green
btn2	yellow
btn3	red
btn4	orange
btn5	black

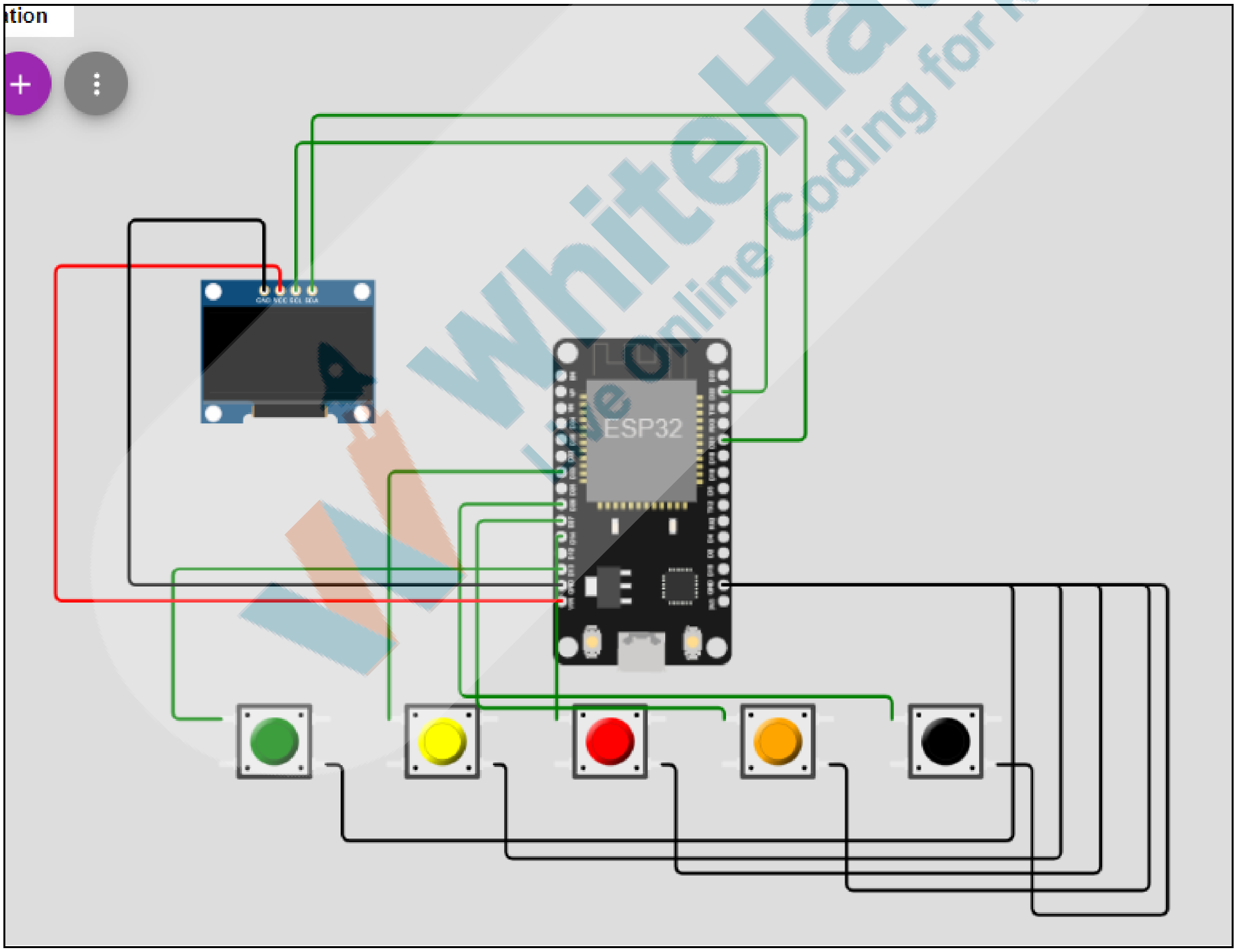
Reference Diagram:



- Connect the **OLED display** with the **ESP 32** board as shown below:

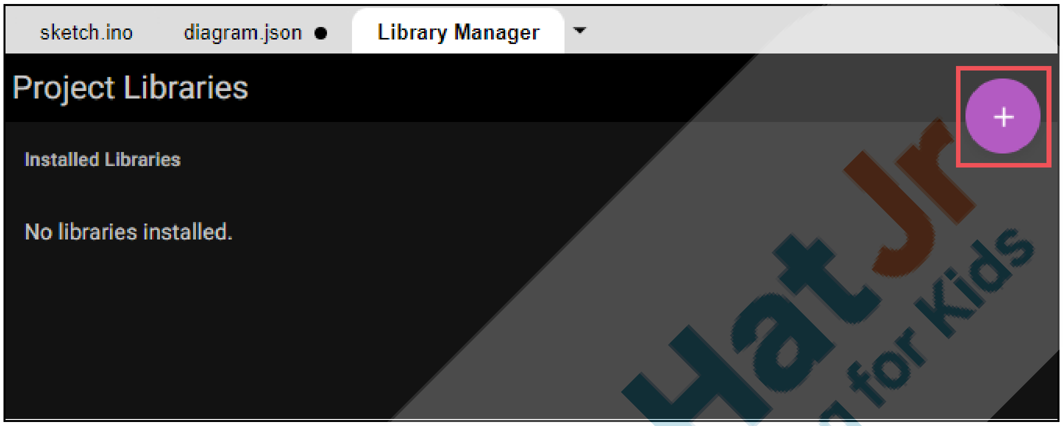
OLED pin	ESP32 pin
VCC	VIN
GND	GND
SCL (CLK)	GPIO 22
SDA (DATA)	GPIO 21

Reference Diagram:



To control the **OLED** display, we need to install libraries

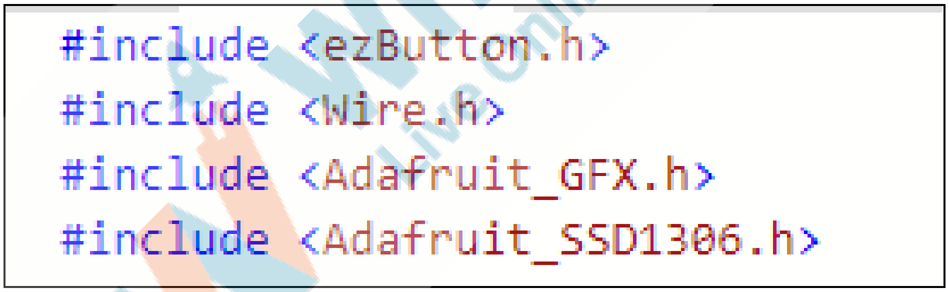
1. Open the **Library Manager**.
2. Click on the ⊕ icon.



No libraries installed.

3. Install **Adafruit SSD1306** and **ezButton** library.

Once, this is done. Go to the **sketch.ino** file and include these header files in the **sketch.ino** file.



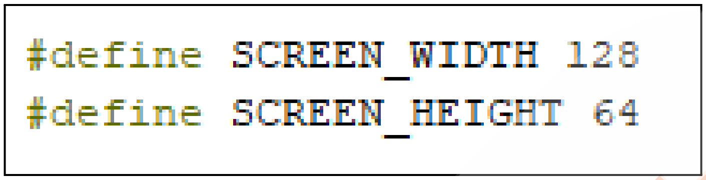
```
#include <ezButton.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

We need these header files for the following reasons:

- **Wire.h:** This **library** allows you to communicate with I2C / devices. I2C is **a serial communication protocol**, so data is transferred bit by bit along a single wire.
- **Adafruit_GFX.h:** This library offers a common graphical syntax and set of functions for all LCD displays, OLED displays, and LED matrices.
- **Adafruit_SSD1306:** This library takes care of low-level communication with the hardware.
- **ezButton.h:** We will use this library to program our push button. Using this library,

it will be easier to debounce the push buttons.

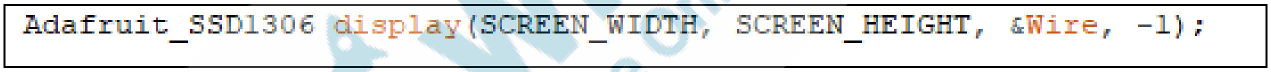
Define **SCREEN_WIDTH & SCREEN_HEIGHT** for OLED
Our **OLED** size is a **128×64**



```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

Declaration of an **SSD1306** display that connects to **I2C** communication using **Wire** Library

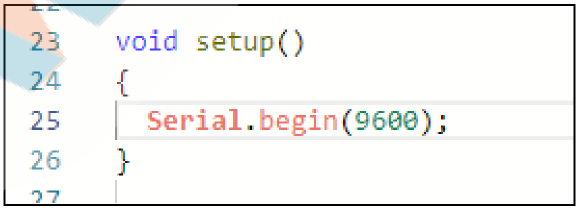
- Initialize a **display** object with the **SCREEN_WIDTH** & **SCREEN_HEIGHT** defined earlier with the I2C communication protocol.
- A value of **(-1)** indicates that our OLED display does not have a **RESET** pin. Sometimes OLED displays have a RESET pin on the OLED, in that case, we should connect it to a GPIO and should include the GPIO number as a parameter.



```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

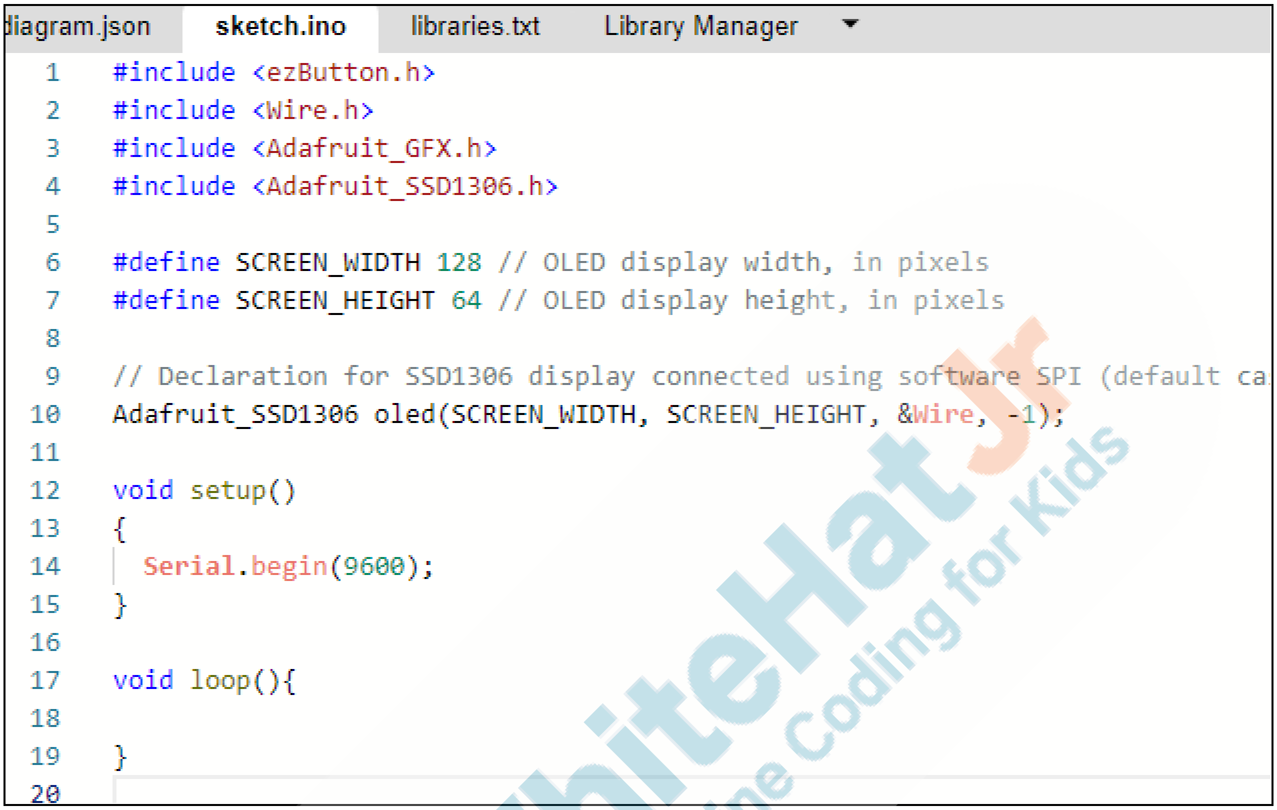
After this, we will define the **setup()** method-

- First, use **Serial.begin()** to set the data rate.



```
void setup()
{
  Serial.begin(9600);
}
```

<u>Reference Code:</u>



```
#include <ezButton.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for SSD1306 display connected using software SPI (default ca
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

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

void loop(){

}

```

So we have set up our **OLED** design. The next task is to display parties names on the voting machine.

Student Stops Screen Share

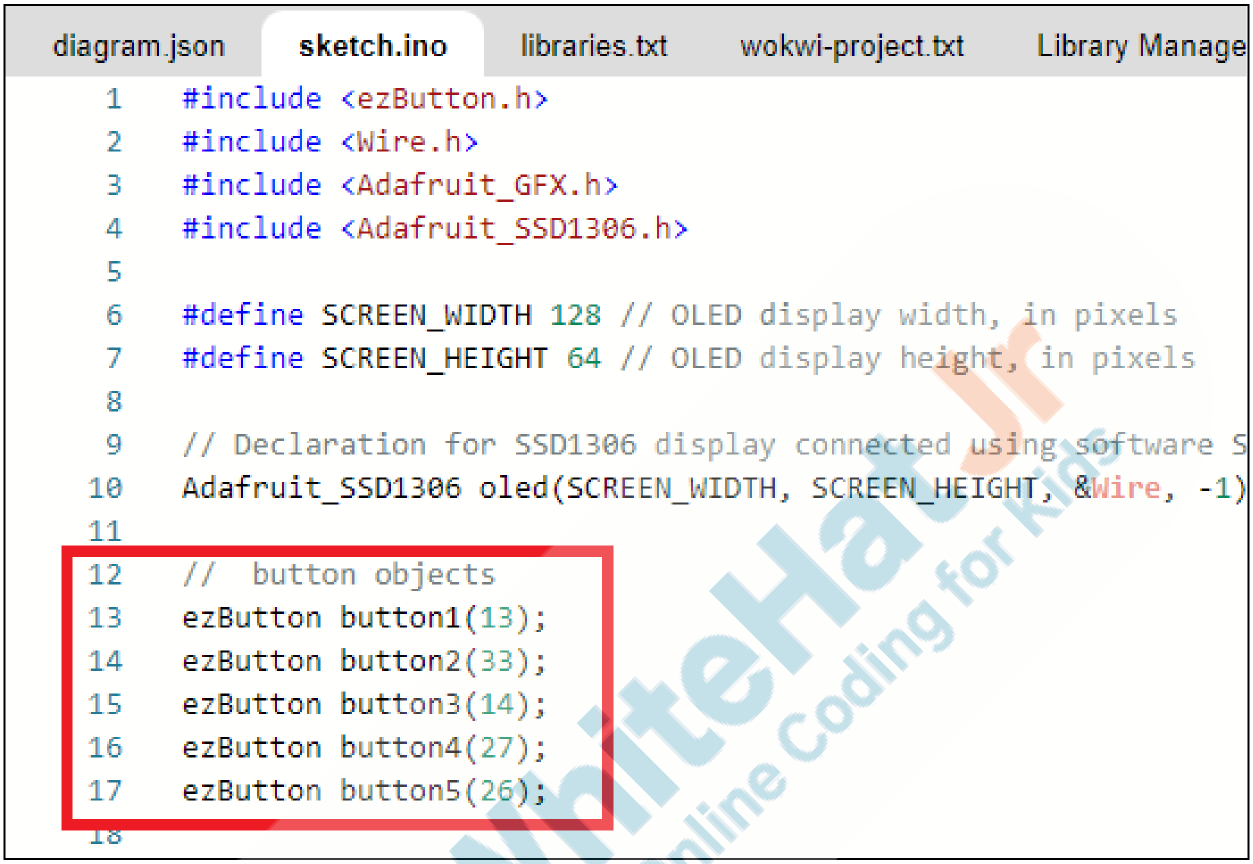
So it's time to set the design of **OLED**.
Please share your screen with me.

STUDENT-LED ACTIVITY - 15 mins

- **Ask the student to press the ESC key to come back to the panel.**
- **Guide the student to start Screen Share.**
- **The teacher gets into Full Screen.**

Student Initiates Screen Share

<table>
<tr><th colspan="2"><u>ACTIVITY</u><ul><li><b>Display parties name and messages on OLED</b></li></ul></th></tr>
<tr><th>Teacher Action</th><th>Student Action</th></tr>
<tr><td><i>The teacher guides the student to download boilerPlate code from <u>Student Activity 3</u>.</i></td><td><i>Student opens the <u>Student Activity 3</u> and downloads the boilerplate code.</i></td></tr>
<tr><td>Let’s start coding now-</td><td></td></tr>
<tr><td><ol><li>We have connected our push buttons to the following pins-</li></ol>
<table>
<tr><th>Button id</th><th>ESP 32 pin</th></tr>
<tr><td>btn1</td><td>13</td></tr>
<tr><td>btn2</td><td>33</td></tr>
<tr><td>btn3</td><td>14</td></tr>
<tr><td>btn4</td><td>27</td></tr>
<tr><td>btn5</td><td>26</td></tr>
</table>
Now, let’s initiate new instances of these buttons-</td><td></td></tr>
</table>

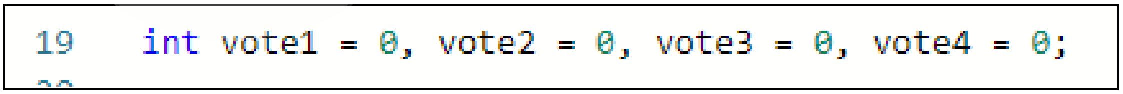


```
#include <ezButton.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for SSD1306 display connected using software S
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1)

//  button objects
ezButton button1(13);
ezButton button2(33);
ezButton button3(14);
ezButton button4(27);
ezButton button5(26);
```

2. Let's initiate the variables for **vote1, vote2, vote3, vote4**.

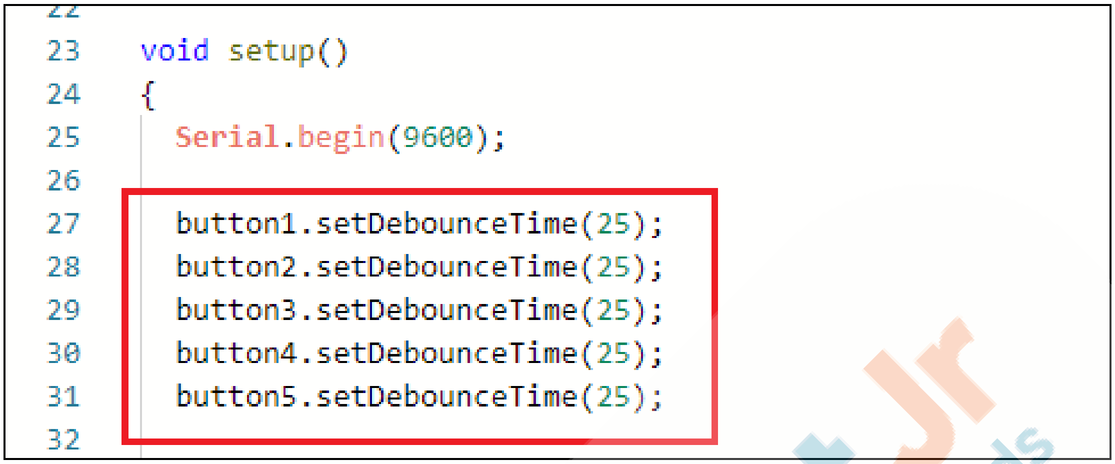
   The number of votes will be integers. So, we will use **int** datatype to declare these variables.

   We will store votes of party **A**, **B**, **C** and **D** in the variables **vote1**, **vote2**, **vote3** and **vote4** respectively.



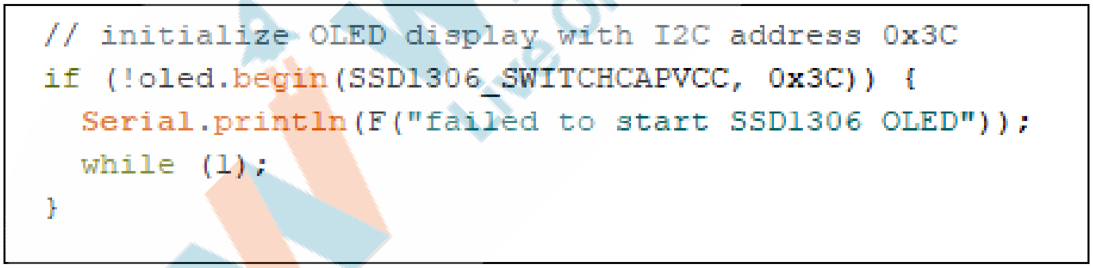
```
int vote1 = 0, vote2 = 0, vote3 = 0, vote4 = 0;
```

3. Now, in the **setup()** method,
   - we need to set the debounce time for each button.



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

  button1.setDebounceTime(25);
  button2.setDebounceTime(25);
  button3.setDebounceTime(25);
  button4.setDebounceTime(25);
  button5.setDebounceTime(25);

```

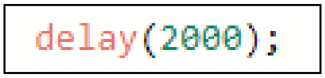
- Initialize the OLED display with the **begin()** method.
- If the OLED displays nothing, check the OLED address at **0x3C**. In our case, the address is 0x3C.
- If we are not able to connect to the display, it prints a message on the Serial Monitor.If something fails, don't proceed further, try to repeat the process using a **for()** loop.



```
// initialize OLED display with I2C address 0x3C
if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("failed to start SSD1306 OLED"));
  while (1);
}
```

4. <u>**Print data on OLED:**</u>

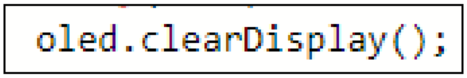
- In order to allow the OLED to initialize, add a **two-second delay** before writing text.



```
delay(2000);
```

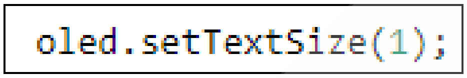
- Clear the display buffer with the

clearDisplay() method after initializing the display.



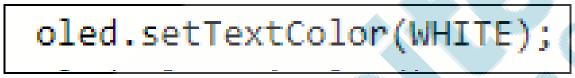
```
oled.clearDisplay();
```

- To **write** text, you must first set the font size, color, and location where the text will be displayed in the OLED and the data which needs to be printed.
- Set the font size using the **setTextSize()** method.



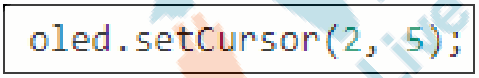
```
oled.setTextSize(1);
```

- Set the font color using the **setTextColor()** method.**WHITE** sets white font and black background.



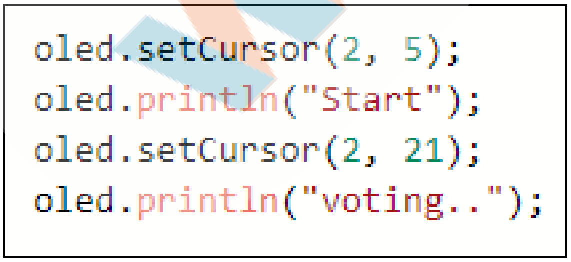
```
oled.setTextColor(WHITE);
```

- Using the **setCursor(x,y)** method, specify the starting point of the text. In this case, the text will be started at **(2,5).**



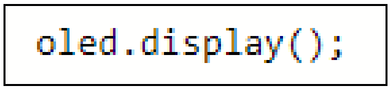
```
oled.setCursor(2, 5);
```

- The next step is to display data using the **println()** method.



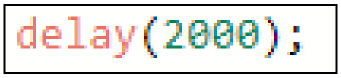
```
oled.setCursor(2, 5);
oled.println("Start");
oled.setCursor(2, 21);
oled.println("voting..");
```

- Finally, we need to call the **display()** method to actually display the text on the screen.



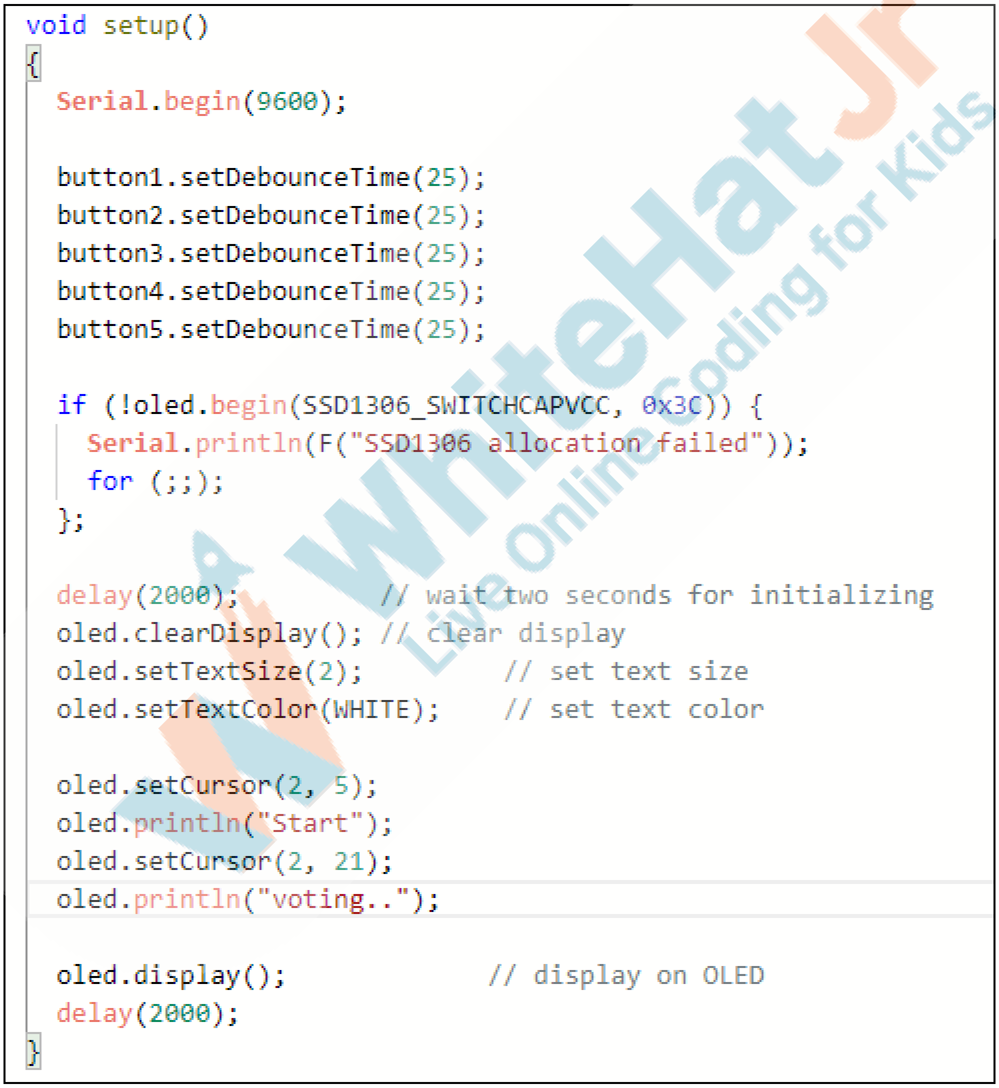
```
oled.display();
```

- Add a 2 seconds delay after this.



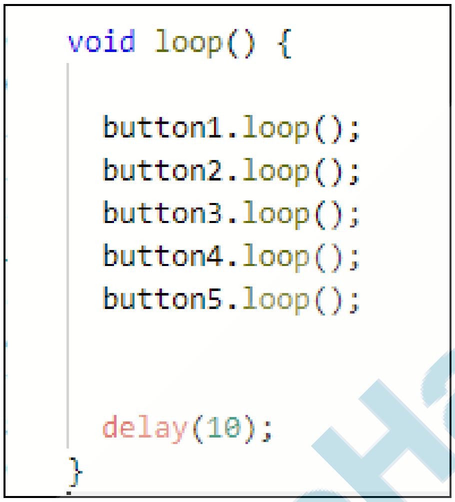
```
delay(2000);
```

Reference Code:



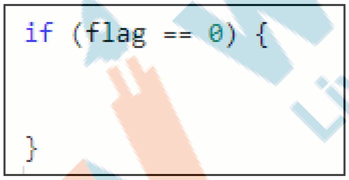
```
void setup()
{
  Serial.begin(9600);

  button1.setDebounceTime(25);
  button2.setDebounceTime(25);
  button3.setDebounceTime(25);
  button4.setDebounceTime(25);
  button5.setDebounceTime(25);

  if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  };

  delay(2000);          // wait two seconds for initializing
  oled.clearDisplay(); // clear display
  oled.setTextSize(2);          // set text size
  oled.setTextColor(WHITE);     // set text color

  oled.setCursor(2, 5);
  oled.println("Start");
  oled.setCursor(2, 21);
  oled.println("voting..");

  oled.display();              // display on OLED
  delay(2000);
}
```

5. Also, we need to call the **loop()** method for each button to make them work.



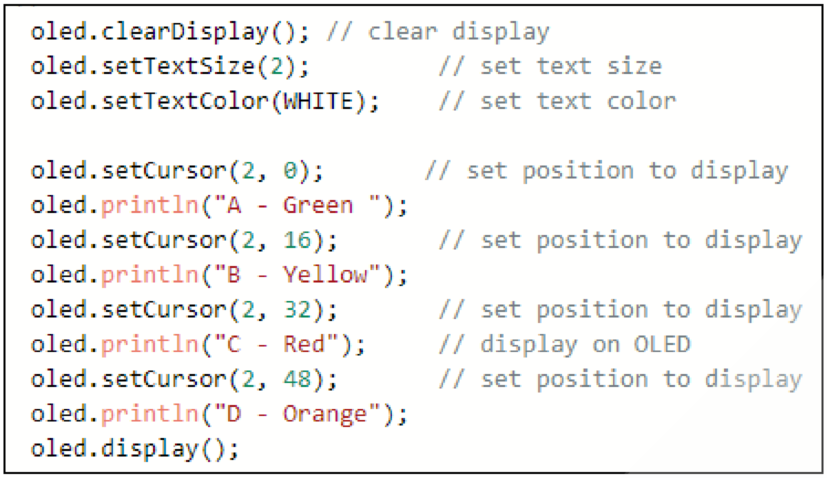
```
void loop() {

  button1.loop();
  button2.loop();
  button3.loop();
  button4.loop();
  button5.loop();


  delay(10);
}
```

6. Now, we need to display the information about which button represents which team.
    a. First, initiate a variable named **flag** to 0.
    b. Write an if condition which will run only when flag is equals 0.



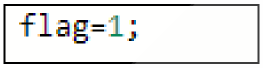
```
if (flag == 0) {


}
```

    c. Now, write code to show the button & team information.



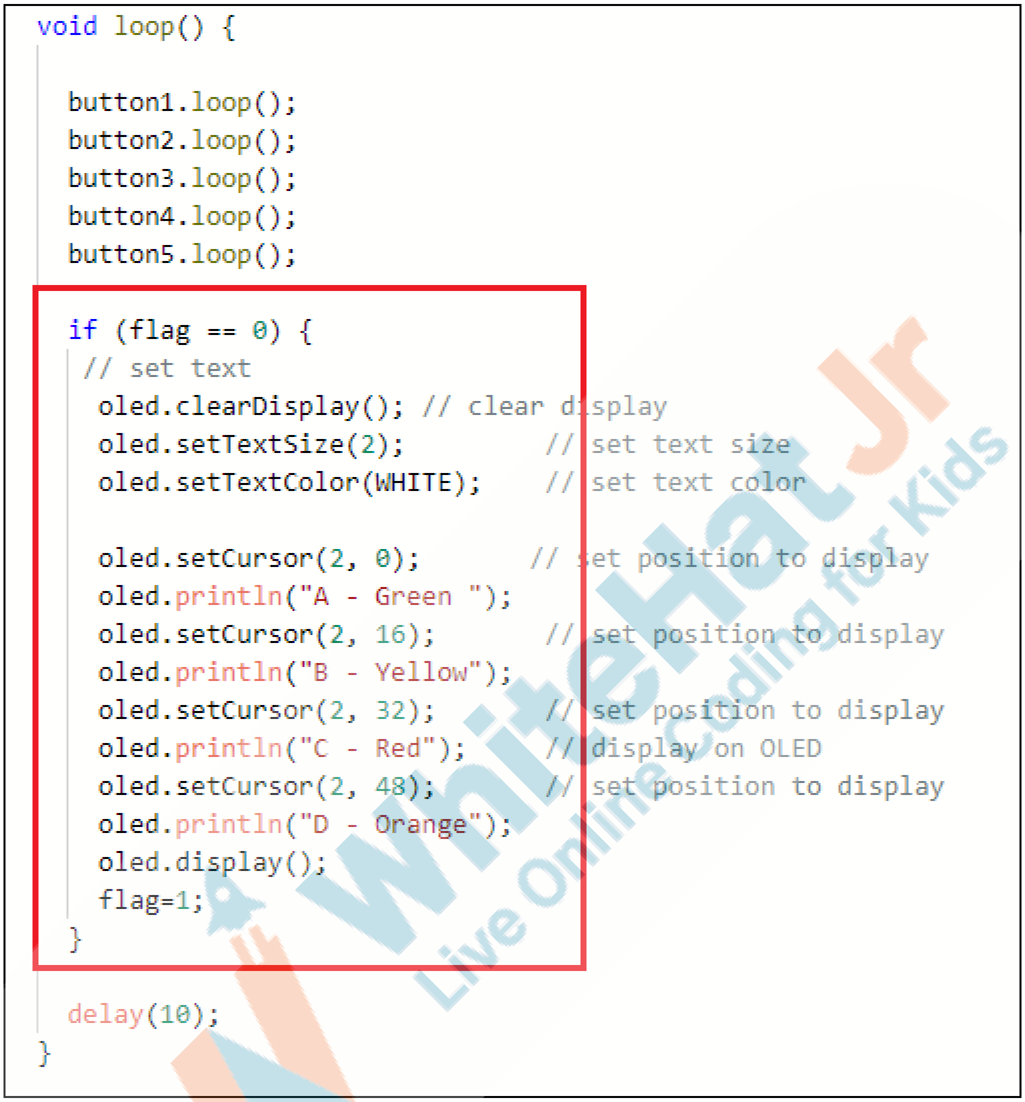
```
oled.clearDisplay(); // clear display
oled.setTextSize(2);        // set text size
oled.setTextColor(WHITE);   // set text color

oled.setCursor(2, 0);       // set position to display
oled.println("A - Green ");
oled.setCursor(2, 16);      // set position to display
oled.println("B - Yellow");
oled.setCursor(2, 32);      // set position to display
oled.println("C - Red");    // display on OLED
oled.setCursor(2, 48);      // set position to display
oled.println("D - Orange");
oled.display();
```

d. Change the flag to 1 now. This will make sure that this portion of code will only run once when flag is 0.



```
flag=1;
```

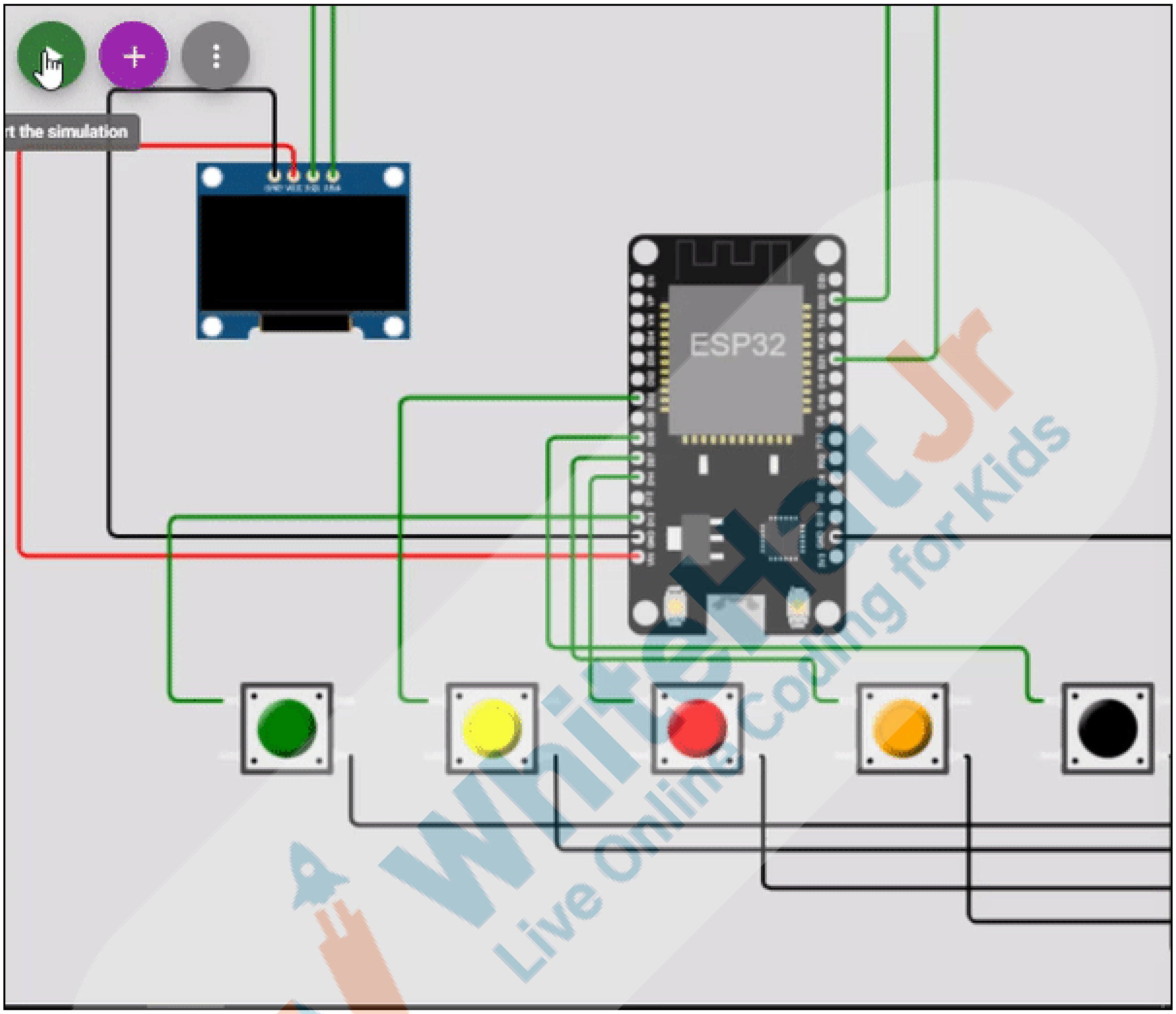
Reference Code:



```
void loop() {

  button1.loop();
  button2.loop();
  button3.loop();
  button4.loop();
  button5.loop();

  if (flag == 0) {
   // set text
    oled.clearDisplay(); // clear display
    oled.setTextSize(2);         // set text size
    oled.setTextColor(WHITE);    // set text color

    oled.setCursor(2, 0);       // set position to display
    oled.println("A - Green ");
    oled.setCursor(2, 16);       // set position to display
    oled.println("B - Yellow");
    oled.setCursor(2, 32);       // set position to display
    oled.println("C - Red");     // display on OLED
    oled.setCursor(2, 48);       // set position to display
    oled.println("D - Orange");
    oled.display();
    flag=1;
  }

  delay(10);
}
```

Reference Output:

Click on the save button and then Click on restart the simulation. If there is any error, resolve it.



[Click here](#) to view the reference video.

Great, so now we can see parties' names on the OLED, but still, it's not completed. Next, we will write logic to count the votes.	

Teacher Guides Student to Stop Screen Share

WRAP-UP SESSION - 05 mins

Activity details

Following are the WRAP-UP session deliverables:

- Appreciate the student.
- Revise the current class activities.
- Discuss the quizzes.

WRAP-UP QUIZ
Click on In-Class Quiz

Activity Details

Following are the session deliverables:

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

FEEDBACK

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get "hats-off" for your excellent work! In the next class, we will complete the **Electronic Voting Machine** project.	*Make sure you have given at least 2 hats-off during the class for:*

PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections
Teacher Clicks ✖ End Class

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Simulator	https://wokwi.com/
Teacher Activity 2	Wokwi push button Reference	https://docs.wokwi.com/parts/wokwi-pushbutton
Teacher Reference 1	Reference Code	https://github.com/procodingclass/PRO-C257-Reference-Code
Teacher Reference 2	Project	https://s3-whjr-curriculum-uploads.whjr.online/4af13c40-0848-48a3-9583-487e560a9dba.pdf
Teacher Reference 3	Project Solution	https://wokwi.com/projects/339608931890365010
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/81177453-baab-4f61-b155-97f0f0896dd3.docx
Student Activity 1	Simulator	https://wokwi.com/
Student Activity 2	Wokwi push button Reference	https://docs.wokwi.com/parts/wokwi-pushbutton
Student Activity 3	Boilerplate Code	https://github.com/procodingclass/PRO-C257-Student-Boilerplate