Topic	ELECTRONIC VOTING MACHINE-2	
Class Description	**Students will write an algorithm for an electronic voting machine on an OLED display. Additionally, students will program to display the results on the OLED display.**	
Class	**PRO C258**	
Class time	**50 mins**	
Goal	• Algorithm to calculate vote. • Display calculations and results.	

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** **Student-Led Activity** **Wrap-Up**	**10 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 15mins	
Student Initiates Screen Share	
• **Algorithm to calculate vote count**	
Teacher Action	**Student Action**
The teacher opens the code from the last class by clicking on Teacher Activity 1.	
In the last class, we started with our Electronic Voting Machine. Do you have any doubts? *If the student has any doubts, clarify the doubts.* In today's class, our focus will be on the EVM machine's algorithm. We will also display the results after that.	**ESR:** Varied.

Let's start!	
1. Firstly, let's initiate a new variable named **voting_completed.** `int voting_completed = 0;` We will use this variable to store the state of voting completion. If it's value is 0, the voting procedure is still going on. We will change this variable to 1 as soon as the black push button is pressed. This will mean the voting procedure is completed.	

2. In the main loop() method, we will write code to check if **button1** is pressed or not.
 	
But users should be able to vote only when the initial texts have been displayed i.e. after **flag** is 1 and when the **voting_completed** variable is 0.
 	
`if (voting_completed == 0 && flag==1) {`
 	
`}`	

3. Now, in the **loop()** method, we will write code for **button1** or the green button. Do you remember which library we were using to program the buttons? What code did we write for the buttons till now?	**ESR:** Yes! **ezButton** library **ESR:** We have written code to create an instance of each button. Then, we used the **setDebounceTime()**

Great! Now, we need to write the code to check if the button1 is pressed or not. To do this, we will use the **isPressed()** method. The **isPressed()** method returns true only when the button is pressed.

If the **button1** is pressed, we will increase the **vote1** variable by 1 and print the value of **vote1** on the **serial monitor**.

```
if (voting_completed == 0 && flag==1) {
  if (button1.isPressed()) {
    vote1++;
    Serial.println(vote1);
  }
}
```

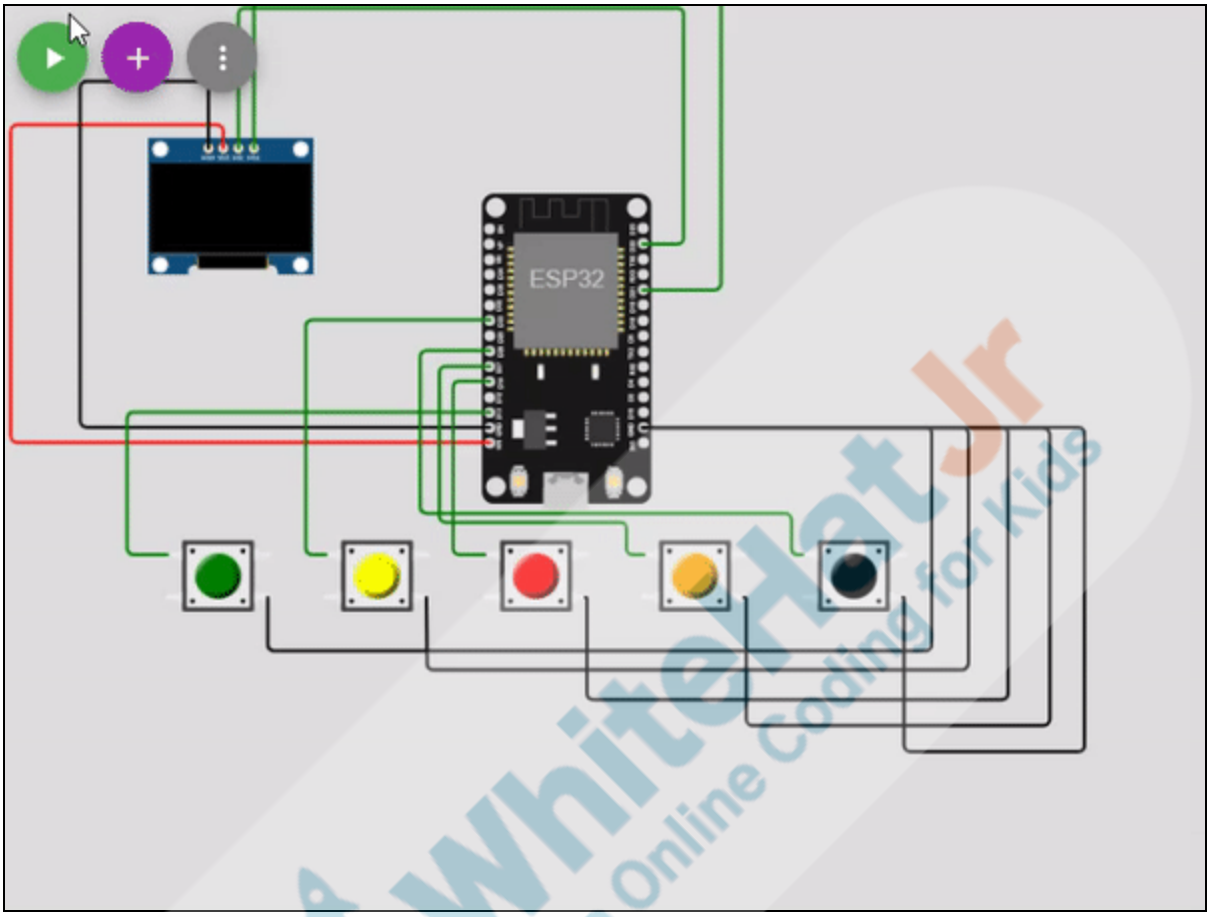
method to set the button debounce interval.

After that, we used the **loop()** method for each button.

Reference code:

```
72        oled.setCursor(2, 32);        // set position to display
73        oled.println("C - Red");      // display on OLED
74        oled.setCursor(2, 48);        // set position to display
75        oled.println("D - Orange");
76        oled.display();
77        flag=1;
78      }
79
80      if (voting_completed == 0 && flag==1) {
81        if (button1.isPressed()) {
82          vote1++;
83          Serial.println(vote1);
84        }
85      }
86
87      delay(10);
88    }
```

Reference Output:



Click here to view the reference video.

So we have understood how to write the code for when a button is pressed. Now, you will write code for the rest of the buttons.	
Student Stops Screen Share	
We have one more class challenge for you. Can you solve it? Let's try. I will guide you through it.	
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

ACTIVITY

- **Algorithm for calculations of the votes**
- **Determine the winner and display result**

Teacher Action	Student Action
Teacher guides the student to open the code from Student Activity 1	*Student downloads the code from Student Activity 1*
Now, what do we need to do? Exactly. So, 1. when **button2** is pressed, **vote2** will increase, when **button3** is pressed, **vote3** will increase, when **button4** is pressed, **vote4** will increase. Finally, when **button5** is pressed, the voting will be completed and we.will increase **voting_completed** value to 1.	**ESR:** We need to write the code for the rest of the buttons.

```
if (voting_completed == 0 && flag==1) {
  if (button1.isPressed()) {
    vote1++;
  }
  else if (button2.isPressed()) {
    vote2++;
  }
  else if (button3.isPressed()) {
    vote3++;
  }
  else if (button4.isPressed()) {
    vote4++;
  }
  else if (button5.isPressed()) {
    voting_completed = 1;
  }
}
```

2. Now, as we have the number of votes for each team. We will define a new method named **show_votes()** and call it when **button5** is pressed.

```
    else if (button3.isPressed()) {
      vote3++;
    }
    else if (button4.isPressed()) {
      vote4++;
    }
    else if (button5.isPressed()) {
      voting_completed = 1;
      show_votes();
    }
  }
  delay(10);
}

void show_votes() {


}
```

3. Let's write the **show_votes()** method now.
   Here, we will display each team's name and their corresponding number of votes.

```
void show_votes() {
  oled.clearDisplay();
  oled.setTextSize(2);
  oled.setTextColor(WHITE);

  oled.setCursor(2, 0);
  oled.print("A - ");
  oled.setCursor(50, 0);
  oled.print(vote1);

  oled.setCursor(2, 16);
  oled.println("B - ");
  oled.setCursor(50, 16);
  oled.print(vote2);

  oled.setCursor(2, 32);
  oled.println("C - ");
  oled.setCursor(50, 32);
  oled.print(vote3);

  oled.setCursor(2, 48);
  oled.println("D - ");
  oled.setCursor(50, 48);
  oled.print(vote4);
  oled.display();
}
```

4. Now, we define the **determine_winner()** method. From the name it is evident that this method will determine the winner and display it on the **OLED display.**
    - As we want to print messages when the winner is determined, let's set the textSize, textColor and setCursor first.

```
oled.clearDisplay();
oled.setTextSize(3);
oled.setTextColor(WHITE);
oled.setCursor(1, 10);
```

- For each team, we will check if this team's number of votes is greater than the other 3 teams. To write this, we will define an if-else ladder.

```
if (vote1 > vote2 && vote1 > vote3 && vote1 > vote4)
  oled.print("A won!");
else if (vote2 > vote1 && vote2 > vote3 && vote2 > vote4)
  oled.print("B won!");
else if (vote3 > vote1 && vote3 > vote2 && vote3 > vote4)
  oled.print("C won!");
else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)
  oled.print("D won!");
```

- If none of these conditions are true, that means some of these teams have the same number of votes. We can print "tie" in this case.

```
else
  oled.print("Tie!");
```

Reference Code:

```
void determine_winner() {
  oled.clearDisplay();
  oled.setTextSize(3);
  oled.setTextColor(WHITE);
  oled.setCursor(1, 10);

  if ((vote1 > vote2 && vote1 > vote3 && vote1 > vote4))
    oled.print("A won!");
  else if (vote2 > vote1 && vote2 > vote3 && vote2 > vote4)
    oled.print("B won!");
  else if (vote3 > vote1 && vote3 > vote2 && vote3 > vote4)
    oled.print("C won!");
  else if(vote4 > vote1 && vote4 > vote2 && vote4 > vote3)
    oled.print("D won!");
  else
    oled.print("Tie!");

  oled.display();
  delay(1000);
}
```

Click on the save button and then Click on restart the simulation

- If there is any error resolve it

Note:

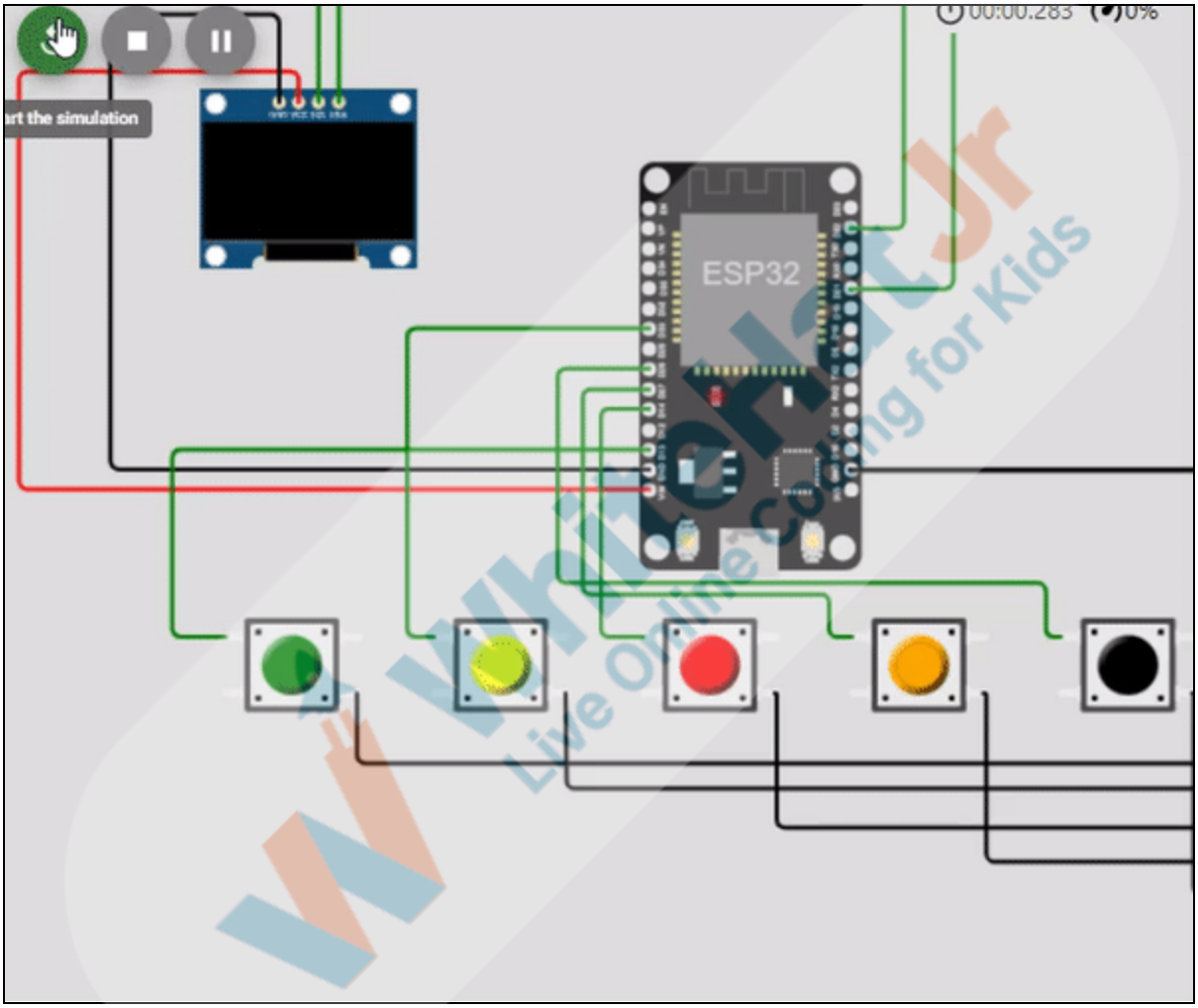
If your OLED display is not showing anything:

Check that the OLED display is properly wired. Check all connections should be tight.

How the code works:
First, wait until the teams page shows and then you can

vote. *Once voting is completed, click the last button i.e the black button to display the winner.*	

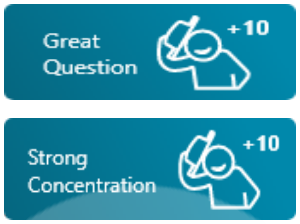
Reference Output:



Click here to view the reference code.

So, we have completed our Electronic Voting Machine. That's fun!	

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 learn about keypads.	*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 	
ADDITIONAL ACTIVITIES (Optional)	
Additional Activities	
Teacher Action	**Student Action**
We have written the code for winner determination. Also, we have the code which checks if there was a tie. You can try writing the code which can determine which teams have tied at the winning position and print teams' names on the screen.	

ACTIVITY LINKS

Activity Name	Description	Links
Teacher Activity 1	Simulator	https://wokwi.com/
Teacher Activity 2	PushButton wokwi	https://docs.wokwi.com/parts/wokwi-pushbutton
Teacher Activity 3	Teacher BoilerPlate code	https://github.com/procodingclass/PRO-C257-Reference-Code
Teacher Reference 1	Reference Code	https://github.com/procodingclass/PRO-C258-Reference-Code
Teacher Reference 2	Project	https://s3-whjr-curriculum-uploads.whjr.online/7aba23c6-237c-4919-a97e-59e0122340e8.pdf
Teacher Reference 3	Project Solution	https://wokwi.com/projects/339615154517836371
Teacher Reference 4	In-Class -Quiz	https://s3-whjr-curriculum-uploads.whjr.online/eb73dce6-ac3c-4d0a-a4d6-292d54ff2a99.pdf
Student Activity 1	Student Boilerplate Code	https://wokwi.com/projects/340148243302187602
Student Activity 2	PushButton wokwi	https://docs.wokwi.com/parts/wokwi-pushbutton