OLED BIT MAP ARRAY

What is our GOAL for this CLASS?

In this class, we learned about OLED

What did we ACHIEVE in the class TODAY?

- We learned about OLED
- We displayed BIT MAP Array on OLED

Which CONCEPTS/ CODING BLOCKS did we cover today?

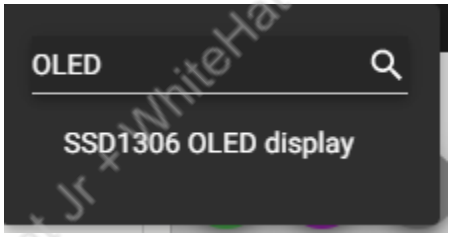
- OLED:
    - An OLED stands for an organic light-emitting diode. An OLED display is made up of pixels that glow when electricity is applied to them. It's like the heating elements in a toaster, but with less heat and a better resolution. This effect is called **electroluminescence**
    - It is called organic because it is made up of organic substances, such as carbon.
    - **BITMAP** is an array of binary data representing the values of pixels in an image or display. A bitmap is a file format or memory organization of rows and columns of bits (or pixels) that collectively display a graphical representation. Most graphic images contain thousands of bits. **Pixels** are larger squares made up of bits.

We can draw a bitmap in this OLED display with some steps:

1. Resize your image to fit the OLED display.
2. Convert image to monochrome
3. Convert monochrome image to array
4. Copy the array to Arduino code.

How did we DO the activities?

1. Display **WHITEHATJR** on the OLED.
   - **Collect the material**
     - 1 x ESP32
     - **1 x OLED**



   - **Connections:**
     - Insert OLED into the breadboard
     - Take four jumper wires.
     - OLED VCC to ESP32 PIN VCC
     - OLED GND to ESP32 PIN GND
     - OLED Clk to ESP32 PIN GPIO22
     - OLED Data to ESP32 PIN GPIO21

To control the **OLED** display, install libraries

- Click on the small triangle icon next to Library Manager
- Select New File
- Name the file libraries.txt
- Write down **Adafruit SSD1306**
- Write down **Adafruit GFX Library**

diagram.json | **libraries.txt** | OLED SA.ino | wokwi-project.txt | Library Manager

```
# Wokwi Library List
# See https://docs.wokwi.com/guides/libraries

Adafruit SSD1306
Adafruit GFX Library
```

1. Write the program:
   - **SPI.h Serial Peripheral Interface (SPI)** is a synchronous serial communication protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances.When using SPI, there is always one master device (usually a microcontroller) that controls all peripheral devices.
   - **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.

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

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

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

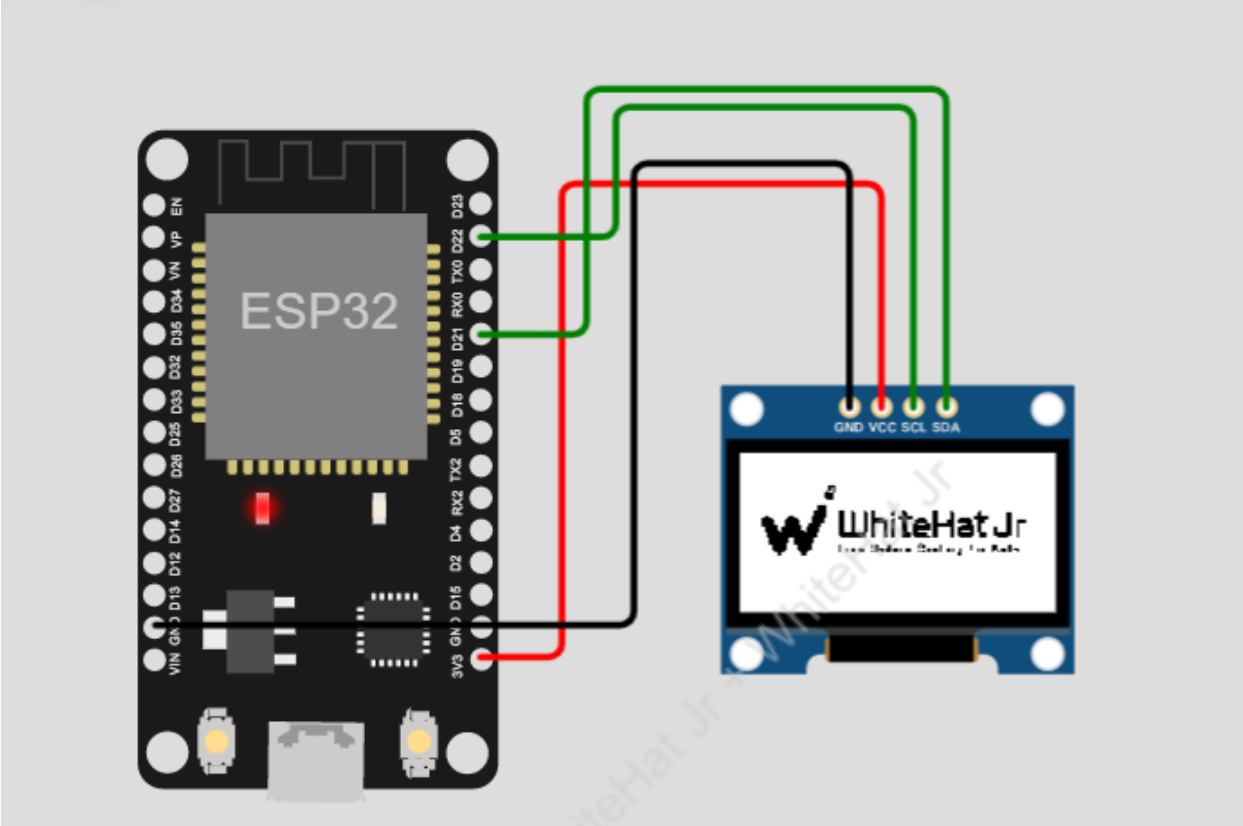
   - Define **SCREEN_WIDTH & SCREEN_HEIGHT** for OLED
   - **OLED** size is a **128×64**

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

   - Initialize using **void setup()** function

- Serial.begin(115200): Sets the **data rate** in bits per second (baud) for **serial** data transmission.
- 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 **for() l**oop
- display.drawBitmap is used to display the picture
  Write down the exact name of the BitMAP array.
  Here in this case it is **bitmap_14v9q**

2. **Output:**
   - **Output:**
   - Click on the Save button and then click on the simulation button



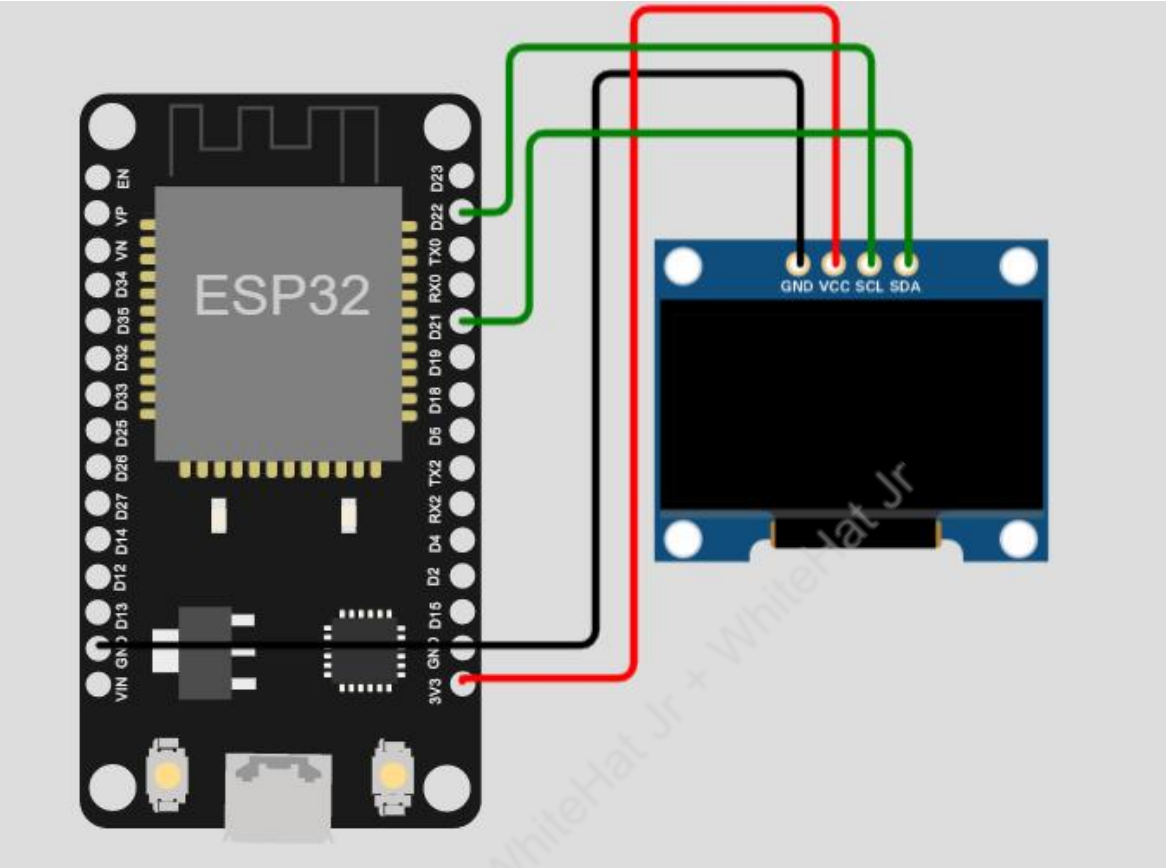
3. **Select the material from the Simulator**

   - 1 x ESP32
   - 1 x OLED Click on + Sign and selects OLED

4. **Let's do connections:**

   - Insert OLED into the breadboard
   - Take four jumper wires.

OLED PIN	ESP32 PIN
VCC	**3.3 V**
GND	**GND**
Data	GPIO **21**
CLK	GPIO **22**



5. Initialize using **void setup()** function

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

- Click on the small triangle icon next to Library Manager
- Select New File
- Name the file libraries.txt
- Write down **Adafruit SSD1306**
- Write down **Adafruit GFX Library**

diagram.json | **libraries.txt** | OLED SA.ino | wokwi-project.txt | Library Manager

```
# Wokwi Library List
# See https://docs.wokwi.com/guides/libraries

Adafruit SSD1306
Adafruit GFX Library
```

7. **Set the MELODY Length, tempo, pause time and rest time**

- **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.
- Define **SCREEN_WIDTH & SCREEN_HEIGHT** for OLED .Our **OLED** size is 1**28×64.**
- 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.

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

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

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

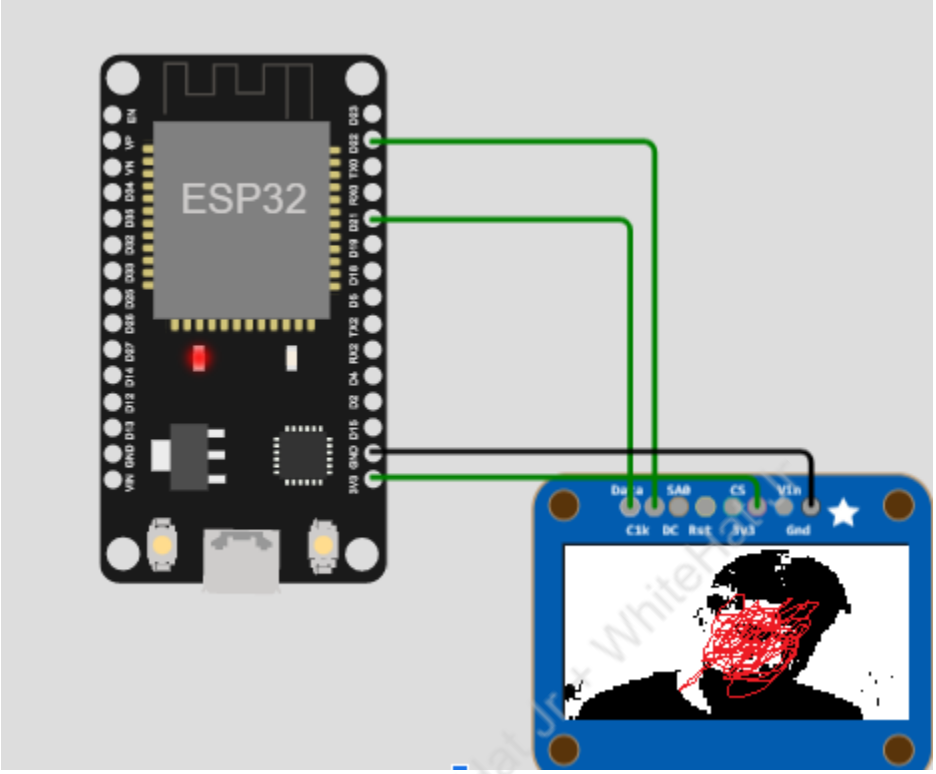
8. Convert Picture into Bit Map Array
9. Initialize using **void setup()** function

- Serial.begin(115200): Sets the **data rate** in bits per second (baud) for **serial** data transmission.

- 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 **for() l**oop
- display.drawBitmap is used to display the picture
  Write down the exact name of the BitMAP array.
  Here in this case it is **bitmap_14v9q**

```
void setup() {
  Serial.begin(115200);

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

  delay(2000);         // wait two seconds for initializing
  display.clearDisplay(); // clear display

  display.drawBitmap(0, 0, bitmap_14v9pq, 128, 64, 1);

  display.display();
}
```

10. Output
    - Click on the save button and then Click on restart the simulation



What's NEXT?

In the **next class**, we will learn about **Electronic Voting machines.**

Expand Your Knowledge

To know more about **Bitmap** click here.