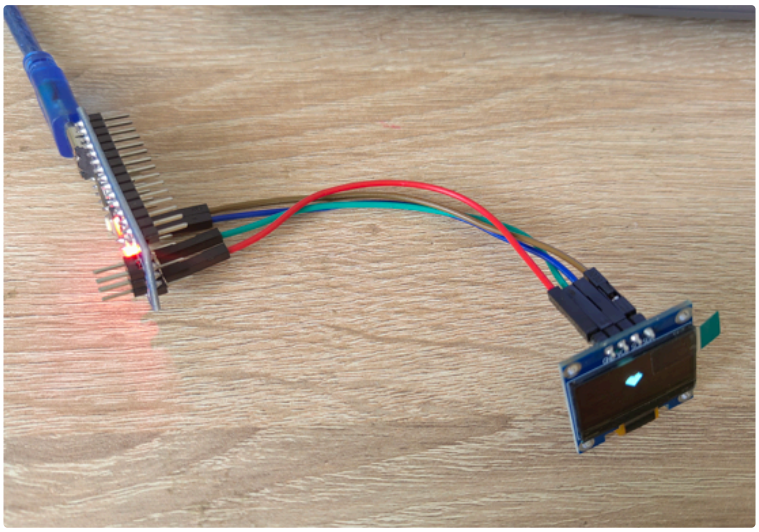
Libraries Adafruit_GFX.h and Adafruit_SSD1306.h



Adafruit_SSD1306 Library: Features and Functionality

The **Adafruit_SSD1306** library is a software tool developed by Adafruit to simplify the use of OLED displays that use the **SSD1306** controller. This chip is widely used in small displays such as 0.96" modules with typical resolutions of **128x64 pixels**, using **I2C or SPI** communication.

This library works in combination with two other essential libraries:

```
#include <Wire.h>                  // I2C communication library
#include <Adafruit_GFX.h>          // Core graphics library
#include <Adafruit_SSD1306.h>      // OLED driver for SSD1306 displays
```

- **Wire.h**: Handles I2C communication between the microcontroller and the display.
- **Adafruit_GFX.h**: A graphics base library that provides functions to draw text, shapes , bitmap images, and more.

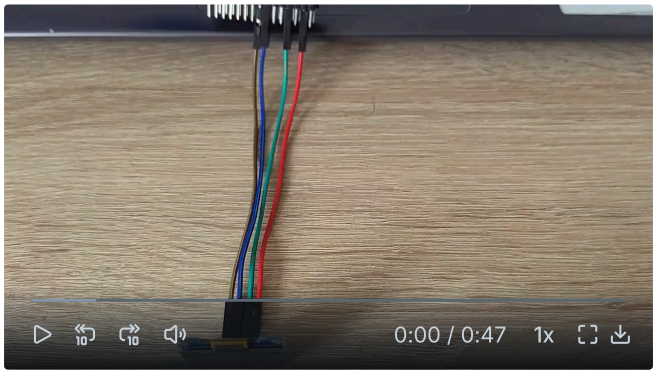
Main Features of Adafruit_SSD1306:

1. **Display Initialization**
   The function `display.begin(...)` configures the OLED display with its I2C address and power settings. It also initializes the display buffer in memory that acts as a virtual screen before pushing content to the actual display.
2. **Clearing and Updating the Display**
   - `display.clearDisplay()` clears the display buffer, basically turning off all pixels.
   - `display.display()` sends the content of the buffer to the physical screen, making it visible.
3. **Text Rendering**
   Using functions inherited from `Adafruit_GFX`, such as:
   - `setCursor(x, y)` to position text.
   - `setTextSize(size)` to scale the text.
   - `setTextColor(color)` to define the text color (white or black on monochrome displays).
   - `print()` to write text directly to the screen.

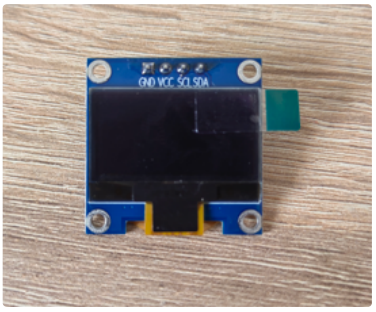
4. **Drawing Graphics**
   Provides functions for drawing:
   - Lines: `drawLine(x0, y0, x1, y1, color)`, you have to write two points, then the color
   - Diferent figures:
     - Rectangles: `drawRect()` or `fillRect()`
     - Circles: `drawCircle()` or `fillCircle()`
     - Triangles, individual pixels, and more.
5. **Displaying Images**
   You can load **bitmaps** as binary arrays and display them with `drawBitmap(x, y, arrayName, width, height, color)`. This is ideal for icons, logos, or simple black-and-white animations. We are going to use it later.
6. **Support for Multiple Resolutions**
   No matter the type of screen resolution, it supports both 128x64 and 128x32.
7. **It daes not support emojis**

   If you want to include emojis, you can't use this library directly.
   However, if you want to display something like a smiley face, you can create it using a custom bitmap.

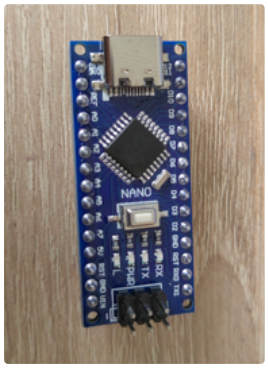
Animated Beating Heart on OLED Display
This program displays a heart icon on an OLED screen that simulates a beating effect by alternating between two bitmap images of different sizes. It creates a simple animation by rapidly switching between them, giving the illusion of a heartbeat.



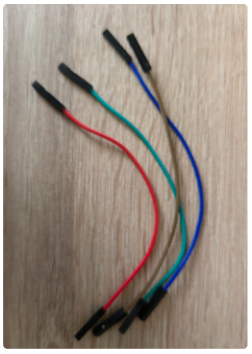
Components



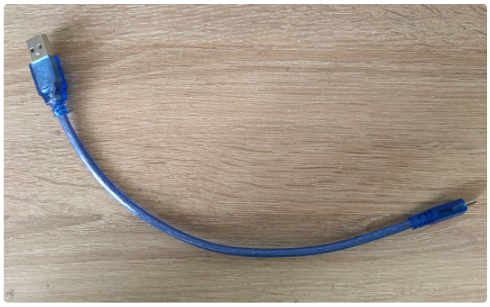
OLED screen 128x64



Arduino Nano



Cables

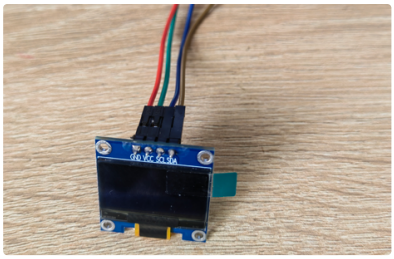


USB to USB-C cable

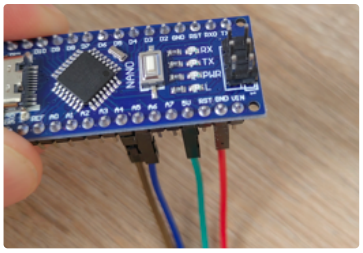
Steps to Build

Connections

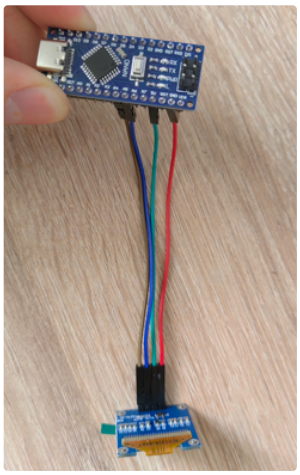
Arduino Nano	Cables	Screen
5V		VCC
GND		GND
SCL		A5
SDA		A4



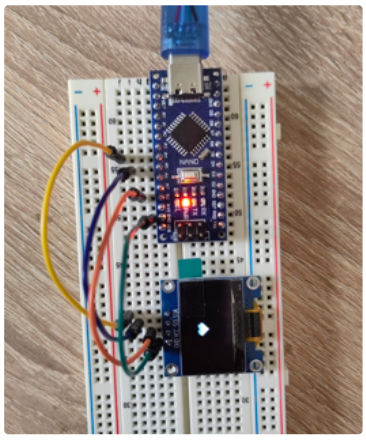
1



2



Result 1



Result 2

Code

```
#include <Wire.h>                    // I2C communication library
#include <Adafruit_GFX.h>           // Core graphics library
#include <Adafruit_SSD1306.h>       // OLED driver for SSD1306 displays

//OLED screen size
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// create an SSD1306 display object connected
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// small heart bitmap (24x21 pixels)
const unsigned char heart_small[] PROGMEM = {
  0b00000110, 0b00000011, 0b00000000, // each row is 24 pixels (3 bytes of 8 bits)
  0b00011111, 0b10011111, 0b00000000,
  0b00111111, 0b11011111, 0b10000000,
  0b01111111, 0b11111111, 0b11000000,
  0b01111111, 0b11111111, 0b11000000,
  0b01111111, 0b11111111, 0b11000000,
  0b00111111, 0b11111111, 0b10000000,
  0b00011111, 0b11111111, 0b00000000,
  0b00001111, 0b11111110, 0b00000000,
  0b00000111, 0b11111100, 0b00000000,
  0b00000011, 0b11111000, 0b00000000,
  0b00000001, 0b11110000, 0b00000000,
  0b00000000, 0b11100000, 0b00000000,
  0b00000000, 0b01000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000, // Empty rows for padding
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000
};

// large heart bitmap (24x21 pixels)
const unsigned char heart_large[] PROGMEM = {
  0b00000110, 0b00000011, 0b00000000, // each row is 24 pixels (3 bytes of 8 bits)
  0b00011111, 0b10011111, 0b00000000,
  0b00111111, 0b11011111, 0b10000000,
  0b01111111, 0b11111111, 0b11000000,
  0b11111111, 0b11111111, 0b11100000, // more pixels here to make it look "larger"
  0b11111111, 0b11111111, 0b11100000,
  0b11111111, 0b11111111, 0b11100000,
  0b01111111, 0b11111111, 0b11000000,
  0b00111111, 0b11111111, 0b10000000,
  0b00011111, 0b11111111, 0b00000000,
  0b00001111, 0b11111110, 0b00000000,
  0b00000111, 0b11111100, 0b00000000,
  0b00000011, 0b11111000, 0b00000000,
  0b00000001, 0b11110000, 0b00000000,
  0b00000000, 0b11100000, 0b00000000,
  0b00000000, 0b01000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
```

```
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000,
  0b00000000, 0b00000000, 0b00000000
};

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // initialize OLED
  display.clearDisplay();// clear display buffer
}

void loop() {
  // draw large heart
  display.clearDisplay(); // clear screen
  display.drawBitmap(52, 20, heart_large, 24, 21, SSD1306_WHITE); // x=52, y=20: center position; 24x21 is bitma
  display.display();
  delay(400); // wait

  // draw small heart
  display.clearDisplay();
  display.drawBitmap(52, 20, heart_small, 24, 21, SSD1306_WHITE); // draw smaller version at same position
  display.display();
  delay(400); // repeat every 400 milliseconds (simulate "beating")
}
```

Explanation of the Code: Beating Heart on OLED

This Arduino program uses the **Adafruit SSD1306** and **Adafruit GFX** libraries to display a simple animation on a 128x64 OLED screen. The animation shows a **heart icon that appears to beat**, giving a visual pulse effect.

Screen Setup

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

These lines define the width and height of the OLED display in pixels. These are standard dimensions for small SSD1306 OLED displays.

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

This creates a display object named `display`, which is used to interact with the OLED screen. It uses I2C communication (`&Wire`) and does not use a reset pin (indicated by `-1`).

Heart Bitmaps

There are two `const unsigned char` arrays stored in PROGMEM:

- `heart_small[]`: A smaller version of the heart shape.
- `heart_large[]`: A slightly larger version of the heart.

These arrays contain binary values that represent the pixels of the image, where 1 is a white pixel and 0 is black.

Each heart is **24 pixels wide and 21 pixels tall**.

Setup Function

```
void setup() {   display.begin(SSD1306_SWITCHCAPVCC, 0x3C);   display.clearDisplay(); }
```

- `display.begin(...)`: Initializes the OLED screen with the I2C address `0x3C` and enables internal voltage conversion (`SSD1306_SWITCHCAPVCC`).
- `display.clearDisplay()`: Clears any previous data from the screen buffer.

Loop Function – Creating the Animation

The loop creates the animation by alternating between the two heart images:

```
display.clearDisplay();
  display.drawBitmap(52, 20, heart_large, 24, 21, SSD1306_WHITE);
  display.display();
  delay(400);
```

- Clears the display.
- Draws the large heart at position `(x=52, y=20)` using `drawBitmap(...)`.
- `display.display()` pushes the buffer to the screen so the image is shown.
- Waits for 400 milliseconds to hold the image.

Then the same is done for the smaller heart:

```
  display.clearDisplay();
  display.drawBitmap(52, 20, heart_small, 24, 21, SSD1306_WHITE);
  display.display();
  delay(400);
```

This repeated switching between the small and large bitmap images **simulates a "beating" heart effect**.