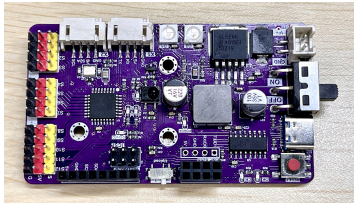


Lesson 5 OLED Display

5.1 Overview

This course mainly introduces the application of OLED display screen on Adeept Pixie Board, covering the required components, working principle, hardware connection, code writing and demonstration, to help master the combination of OLED display screen and Arduino, and achieve the display function of information on OLED screen.

5.2 Required Components

Components	Quantity	Picture
Adeept Pixie Board	1	
Type-C USB Cable	1	
OLED Screen	1	

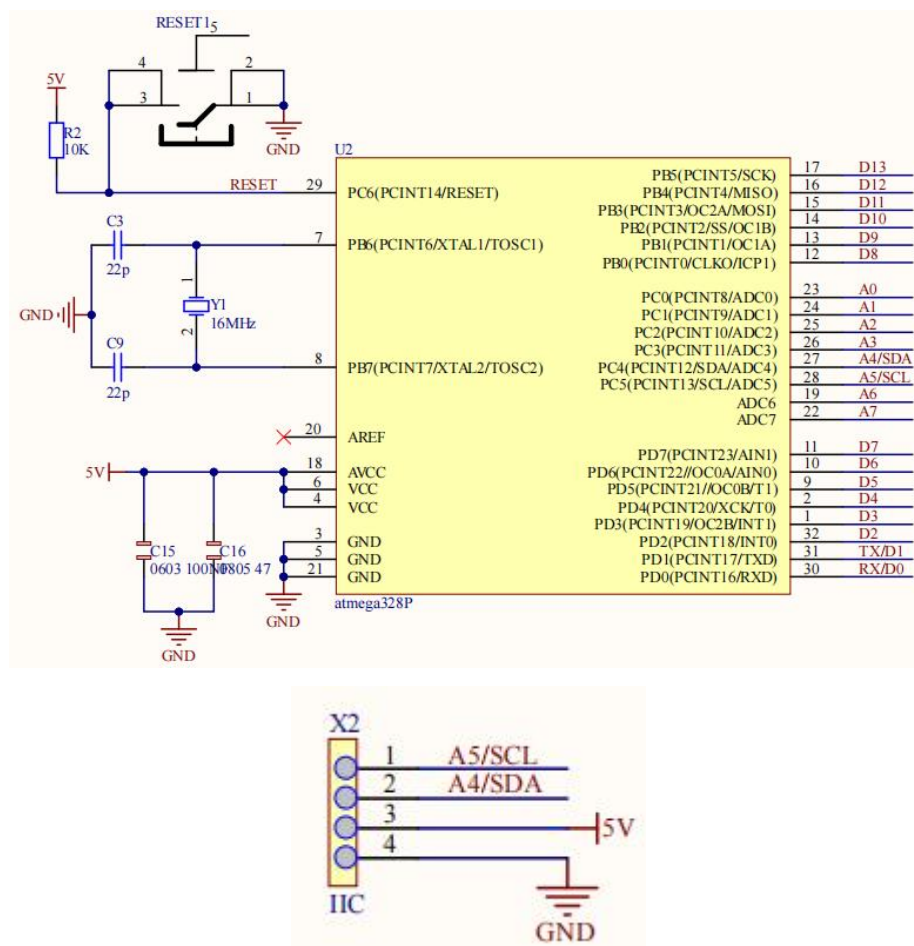
5.3 Principle Introduction

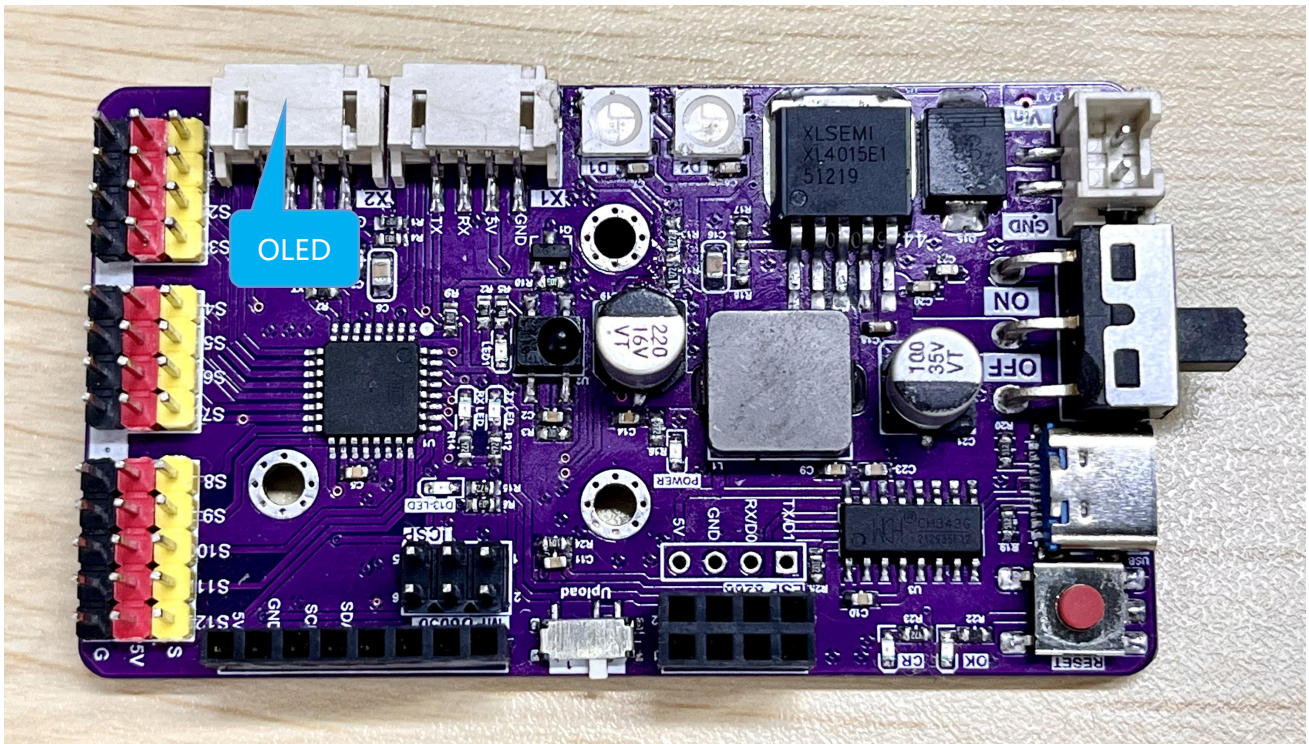
OLED (Organic Light-Emitting Diode), also known as organic electroluminescence display or organic light-emitting semiconductor, belongs to a current-type organic light-emitting device. It

emits light through the injection and recombination of carriers, and the luminous intensity is proportional to the injected current, with relatively low power consumption.

Due to their small size, light weight, and low power consumption, OLED screens are increasingly used in embedded electronic devices. There are various types of OLED screens, which display different colors. Commonly used ones include white display, blue display, and yellow-blue two-color display. Additionally, they come in different screen sizes and are equipped with various built-in driver chips. The commonly used driver interfaces are SPI and IIC. This article introduces a 0.96-inch blue IIC-driven screen, and its built-in driver chip is SSD1305.

5.4 Wiring Diagram

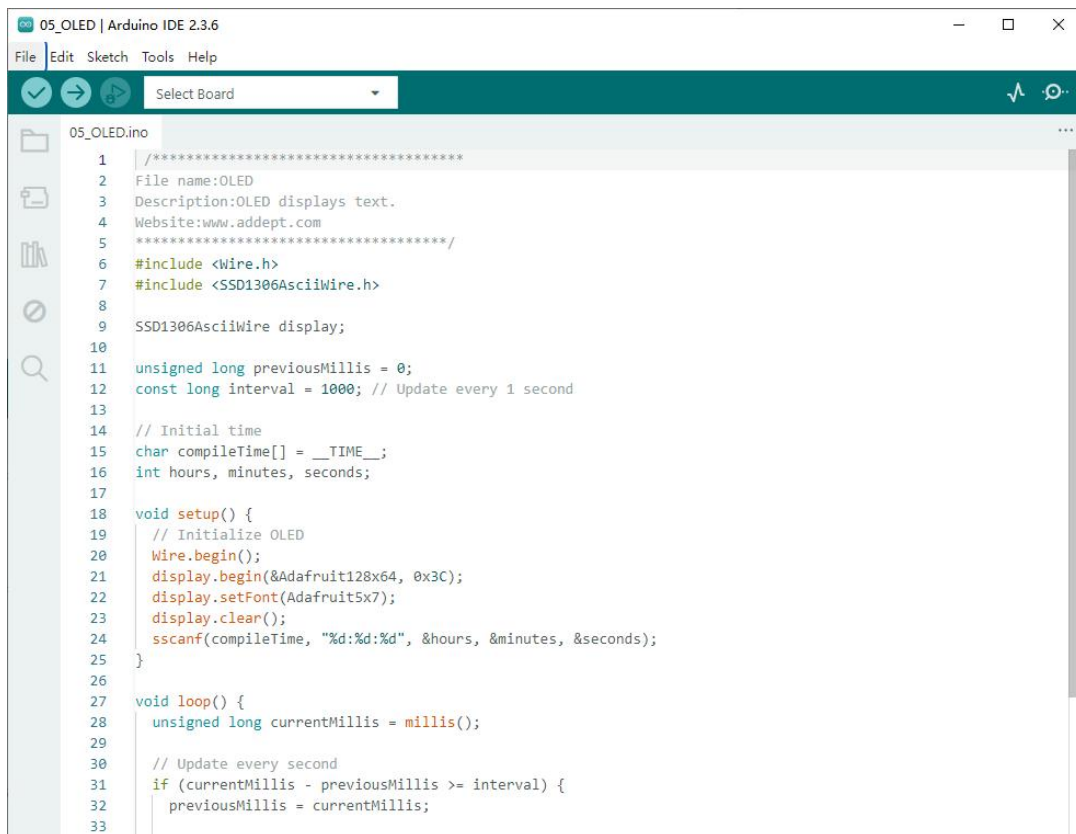




PINS of Adeept Pixie Board	OLED
A4	SDA
A5	SCL
GND	GND
5V	VCC

5.5 Demonstration

1. Connect your computer and Adeept Pixie Board with a USB cable.
2. Open **"05_OLED"** folder in **"/Code"** , double-click **"05_OLED.ino"** .



```
05_OLED | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board

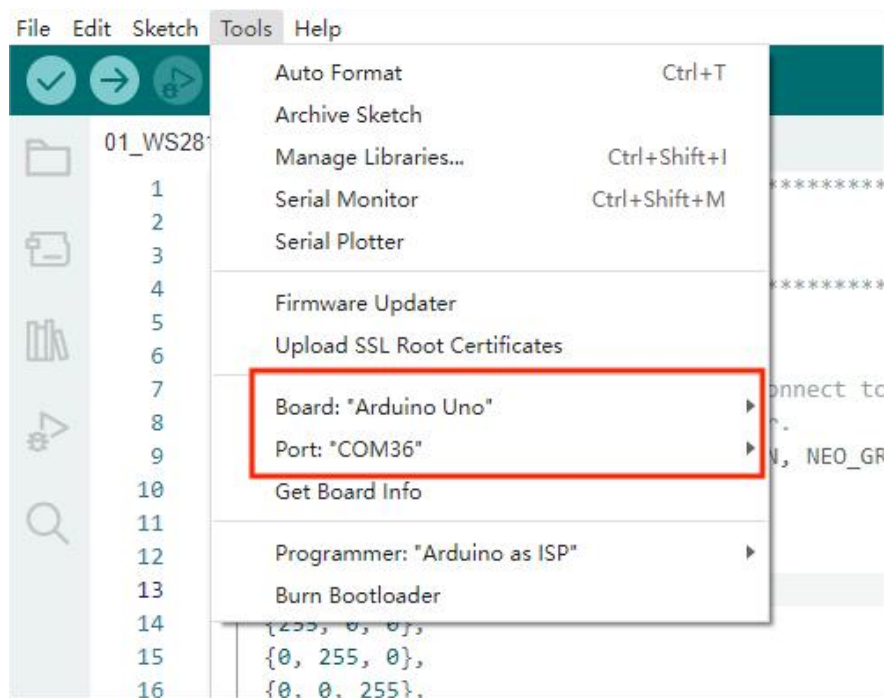
05_OLED.ino
1  /**
2  File name:OLED
3  Description:OLED displays text.
4  Website:www.adeept.com
5  */
6  #include <Wire.h>
7  #include <SSD1306AsciiWire.h>
8
9  SSD1306AsciiWire display;
10
11 unsigned long previousMillis = 0;
12 const long interval = 1000; // Update every 1 second
13
14 // Initial time
15 char compileTime[] = __TIME__;
16 int hours, minutes, seconds;
17
18 void setup() {
19   // Initialize OLED
20   Wire.begin();
21   display.begin(&Adafruit128x64, 0x3C);
22   display.setFont(Adafruit5x7);
23   display.clear();
24   sscanf(compileTime, "%d:%d:%d", &hours, &minutes, &seconds);
25 }
26
27 void loop() {
28   unsigned long currentMillis = millis();
29
30   // Update every second
31   if (currentMillis - previousMillis >= interval) {
32     previousMillis = currentMillis;
33   }
```

3. Select development board and serial port.

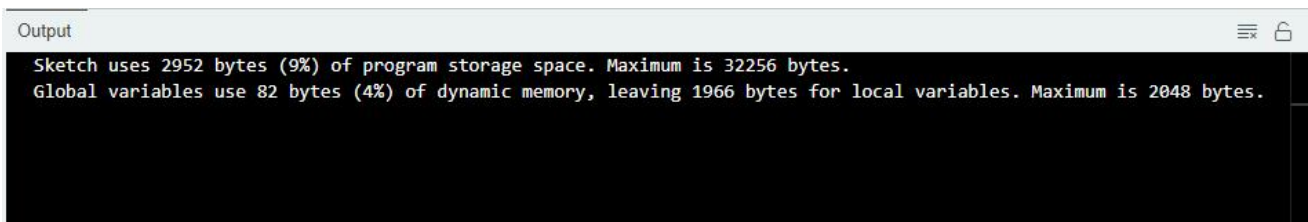
Board: **Tools--->Board--->Arduino AVR Boards--->Arduino Uno**

Port: **Tools --->Port--->COMx**

Note: The port number will be different in different computers.



4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



5. When the program runs successfully, you will clearly see the OLED displays time like a clock.

5.6 Code

```

01  /*****
02  File name:OLED
03  Description:OLED displays text.
04  Website:www.adeept.com
05  *****/
06  #include <Wire.h>
07  #include <SSD1306AsciiWire.h>
08
09  SSD1306AsciiWire display;
10

```

```
11 unsigned long previousMillis = 0;
12 const long interval = 1000; // Update every 1 second
13
14 // Initial time
15 char compileTime[] = __TIME__;
16 int hours, minutes, seconds;
17
18 void setup() {
19     // Initialize OLED
20     Wire.begin();
21     display.begin(&Adafruit128x64, 0x3C);
22     display.setFont(Adafruit5x7);
23     display.clear();
24     sscanf(compileTime, "%d:%d:%d", &hours, &minutes, &seconds);
25 }
26
27 void loop() {
28     unsigned long currentMillis = millis();
29
30     // Update every second
31     if (currentMillis - previousMillis >= interval) {
32         previousMillis = currentMillis;
33
34         // Update time
35         seconds++;
36         if (seconds >= 60) {
37             seconds = 0;
38             minutes++;
39             if (minutes >= 60) {
40                 minutes = 0;
41                 hours++;
42                 if (hours >= 24) {
43                     hours = 0;
44                 }
45             }
46         }
47
48         // Update display
49         updateDisplay();
50     }
51 }
52
53 void updateDisplay() {
54     // Display large time
55     display.set2X();
56     display.setCursor(20, 0); // Center display
57
58     // Format display HH:MM:SS
59     if (hours < 10) display.print("0");
60     display.print(hours);
61     display.print(":");
62     if (minutes < 10) display.print("0");
63     display.print(minutes);
64     display.print(":");
65     if (seconds < 10) display.print("0");
66     display.print(seconds);
```

```
67 }
```

Code explanation

Initialization Stage:

In the `setup ()` function, first initialize I2C communication with `Wire.begin()`, then initialize the Adafruit 128x64 type OLED screen with address `0x3C`, set the font to Adafruit 5x7 and clear the screen; Next, use `display.clear()` to clear the screen. Use `sscanf(compileTime, "%d:%d:%d", &hours, &minutes, &seconds)` to Get the current time.

Loop Control Process:

The time is updated every second and redisplayed on the OLED. Use `display.set2X()` to enlarge the font, locate it with `display.setCursor (20,0)`, and finally use `display.println` to display the time.