

# arduino液晶屏

## 设备

使用0.96寸OLED显示屏 与uno板  
这个显示屏使用IIC协议

我们的显示屏是Adafruit公司开发的，应用他们专用的库

## IIC是什么

IIC通信协议简单，易于实现，适合于低速、短距离的应用场景  
设备默认的地址是0X3C

[Arduino驱动OLED屏幕 – 太极创客 \(taichi-maker.com\)](http://taichi-maker.com)

## 最终成果

### display.display()

1. **缓冲区操作**：Adafruit SSD1306 库使用一个缓冲区来存储所有待显示的图像和文本。你可以多次调用绘图和打印函数，这些操作都会修改缓冲区中的内容
2. **更新显示**：只有当你调用 `display.display()`；时，缓冲区中的内容才会被写入到 OLED 显示屏，从而让你看到更新后的内容。如果不调用这个函数，屏幕上将显示之前的内容，或者什么都不显示。
3. **效率**：这种设计的一个好处是，你可以一次性设置多个显示内容，然后一次性刷新显示，减少了直接写入屏幕的频率，这样可以提高效率。

代码使用了 **Adafruit SSD1306** 库来控制 OLED 显示屏，显示了宿舍号、学号后两位以及它们的不同进制表示。

```
// 引入必要的库
#include <Wire.h>           // 引入 I2C 通信库
#include <Adafruit_GFX.h>   // 引入 Adafruit 图形库（用于绘图）
#include <Adafruit_SSD1306.h> // 引入 Adafruit OLED 库

// OLED显示屏的尺寸，设置显示屏宽度和高度
#define SCREEN_WIDTH 128    // OLED 屏幕宽度为 128 像素
#define SCREEN_HEIGHT 64    // OLED 屏幕高度为 64 像素
#define SSD1306_I2C_ADDRESS 0x3C // OLED 显示屏的 I2C 地址，0x3C 是常见的地址
```

```

// 创建一个 Adafruit_SSD1306 类的对象，初始化 OLED 显示屏
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// 设置宿舍号和学号
const int dormNumber = 617; // 宿舍号设置为 617
const int studentId = 79; // 学号的后两位设置为 79

void setup() {
  // 初始化串口通讯，用于调试输出
  Serial.begin(9600); // 设置串口波特率为 9600

  // 初始化 OLED 显示屏
  // 该函数会通过 I2C 初始化屏幕，显示屏电压由电池供电
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

  // 如果显示屏初始化失败，输出错误信息并进入死循环
  if (!display.begin(SSD1306_SWITCHCAPVCC, SSD1306_I2C_ADDRESS)) {
    Serial.println(F("OLED初始化失败!")); // 串口输出“OLED初始化失败！”
    for (;;); // 停止程序，进入死循环
  }

  // 清空显示屏，确保不会显示旧内容
  display.clearDisplay();

  // 显示宿舍号及其二进制和十六进制表示
  display.setTextSize(1); // 设置文本大小为 1
  display.setTextColor(SSD1306_WHITE); // 设置文本颜色为白色
  display.setCursor(0, 0); // 设置光标位置到屏幕的左上角

  // 打印宿舍号及其二进制和十六进制表示
  display.print("dormitory: "); // 打印文字 "dormitory: "
  display.print(dormNumber); // 打印宿舍号
  display.print(" BIN"); // 打印文字 "BIN"
  display.print(dormNumber, BIN); // 打印宿舍号的二进制表示
  display.print(" HEX: "); // 打印文字 "HEX: "
  display.print(dormNumber, HEX); // 打印宿舍号的十六进制表示

  // 将光标移动到下一行
  display.setCursor(0, 30); // 将光标移动到 Y 轴坐标为 30 的位置
  display.print("BUPT: BUPT "); // 打印文字 "BUPT: BUPT "
  display.print(" ASCII: "); // 打印文字 "ASCII: "
  display.print(char('B'), BIN); // 打印字符 'B' 的 ASCII 值的二进制表示

  // 将光标移动到下一行
  display.setCursor(0, 50); // 将光标移动到 Y 轴坐标为 50 的位置
  display.print("studentId"); // 打印文字 "studentId"
  display.print(studentId); // 打印学号后两位
  display.print(" OCT "); // 打印文字 "OCT"
  display.print(studentId, OCT); // 打印学号后两位的八进制表示
}

```

```

// 刷新 OLED 显示屏，将以上内容显示出来
display.display();           // 刷新屏幕，显示缓冲区的内容
}

void loop() {
    // 在此处可以添加其他代码
    // 目前没有循环任务，所以此函数为空
}

```

## 代码总结

1. `#include`：引入了三个库，分别是 `Wire.h`（I2C 通信库）、`Adafruit_GFX.h`（基本图形绘制库）、`Adafruit_SSD1306.h`（专门控制 OLED 屏幕的库）。
2. `#define`：定义了屏幕的宽度、屏幕的高度以及 OLED 显示屏的 I2C 地址。
3. `Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);`：创建了一个 OLED 显示屏对象，用来控制 OLED 显示屏。
4. `setup()`：初始化了显示屏并进行一些基本的配置：
  - 初始化串口通信（用于调试输出）。
  - 使用 `display.begin()` 初始化 OLED 显示屏。
  - 使用 `display.clearDisplay()` 清空显示屏。
  - 通过 `display.setTextSize()`、`display.setTextColor()` 和 `display.setCursor()` 设置文本显示的大小、颜色和光标位置。
  - 打印宿舍号、学号后两位的不同进制表示：二进制、十六进制和八进制。
  - 使用 `display.display()` 刷新显示屏，确保所有内容在屏幕上显示。
5. `loop()`：在这个例子中，`loop()` 函数为空。通常这里可以放入需要循环执行的代码。

## 各个函数的功能：

- `Serial.begin(9600)`：初始化串口通讯，波特率为 9600。
- `display.begin()`：初始化 OLED 显示屏，并设置电源管理。
- `display.clearDisplay()`：清空 OLED 显示屏，准备显示新的内容。
- `display.setTextSize()`：设置显示文本的大小。
- `display.setTextColor()`：设置显示文本的颜色。
- `display.setCursor()`：设置显示文本时光标的位置。
- `display.print()`：在显示屏上打印文本、数字或其他字符。
- `display.display()`：将缓冲区中的内容刷新到 OLED 屏幕上，确保显示内容更新。

## 打印出很多乱码

## 破案：

1. 因为不能直接打印出汉字，一旦打印汉字就会出现“海嗣文字”的样子，如果一定要打印，应该使用取模软件
2. 这个是有自动换行的，但是`display.setCursor()`又会在选定的位置打印，因此如果选择的位置不合适就会打印在一起

## 1. 初始化问题

确保 OLED 显示屏的初始化成功。如果 OLED 没有正确初始化，它可能不会显示任何内容或显示乱码。请在 `setup()` 中添加一段代码，检查是否初始化成功：

```
if (!display.begin(SSD1306_SWITCHCAPVCC, SSD1306_I2C_ADDRESS)) {  
    Serial.println(F("OLED初始化失败!"));  
    for (;;) // 停止程序  
}
```

## 2. 字体和文本大小

在调用 `display.print()` 前，确保设置的文本大小适合显示屏的尺寸。可以尝试设置文本大小为 1，确保文字不会太大超出显示区域。

```
display.setTextSize(1); // 确保文本大小为1
```

## 3. 清空显示

在显示新内容之前，确保调用 `display.clearDisplay();`，以清空屏幕上的旧内容。

显示adafruit标志：因为 `display.begin()`会显示这个标志，所以在这个之后需要  
`display.clearDisplay();`