

 search

Controller >

DFR0182 Wirless GamePad V2.0

DFR0100 DFRduino Beginner Kit For Arduino V3

DFR0267 Bluno

DFR0282 Beetle

DFR0283 Dreamer Maple V1.0

DFR0296 Bluno Nano

DFR0302 MiniQ 2WD Plus

DFR0304 BLE Wireless Gamepad V2

DFR0305 RoMeo BLE

DFR0351 Romeo BLE mini V2.0

DFR0306 Bluno Mega 1280

DFR0321 Wido-WIFI IoT Node

DFR0323 Bluno Mega 2560

DFR0329 Bluno M3

DFR0339 Bluno Beetle

DFR0343 UHex Low-power Controller

DFR0355 SIM808 with Leonardo mainboard

DFR0392 DFRduino M0 Mainboard Arduino Compatible

DFR0398 Romeo BLE Quad Robot Controller

DFR0416 Bluno M0 Mainboard

DFR0575 Beetle ESP32

DFR0133 X-Board

DFR0162 X-Board V2

DFR0428 3.5 inches TFT Touchscreen for Raspberry Pi

DFR0494 Raspberry Pi UPS HAT

DFR0514 DFR0603 IIC 16X2 RGB LCD KeyPad HAT V1.0

DFR0524 5.5 HDMI OLED-Display with Capacitive Touchscreen V2.0

DFR0550 5" TFT-Display with Touchscreen V1.0

Introduction

Features

Application

Specification

Board Overview

Description of pin function

Tutorial

FAQ

More Documents

SKU:DFR0934



Introduction

This is a 1.51"SSD1309 transparent monochrome OLED display(light blue) with a converter,full-view and blue display. For convenience, the small converter comes with two available connections: GDI and SPI. And there are all kinds of display code samples for meeting the basic requirements of most users.

Features

- Transparent OLED display
- Full-view display
- High resolution
- Blue display

Application

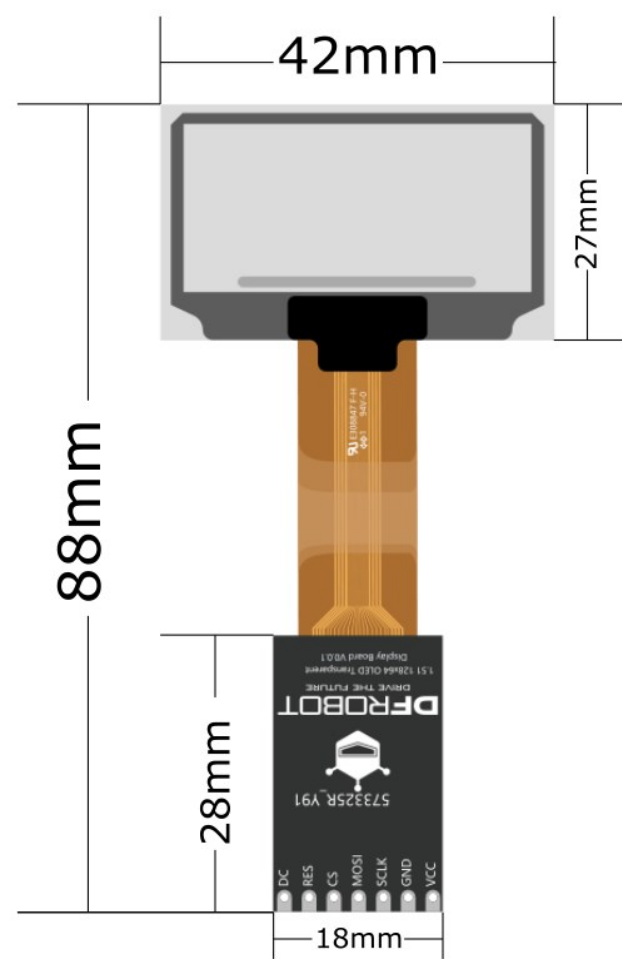
- Smart home transformation
- DIY Desktop display
- Interactive display for booth
- 3D Display

Specification

- Screen Interface: 8-bit 68xx/80xx Parallel, 3-/4-wire SPI, 12C
- Adapter Interface: SPI, GDI
- Adapter Size: 18 x 28mm/0.71 x 1.10inch
- Working Voltage: 3.3V
- View Angle: full view
- Driver Chip: SSD1309
- Display Color: blue
- Display Area: 35.05 x 15.32 (mm)/1.40 x 0.60inch
- Pixel Pitch: 0.274 x 0.274mm
- Pixel Size: 0.254 x 0.254mm
- Screen Size: 41.92x27.08mm/1.65 x 1.07inch
- The fully transparent region resolution :128*56
- Working Temperature: -40~70 °C

- DFR0591 raspberry pi e-ink display module V1.0
- DFR0592 DC Motor Driver HAT
- DFR0604 I O Expansion HAT for Pi zero V1.0
- DFR0566 IO Expansion HAT for Raspberry Pi
- DFR0528 UPS HAT for Raspberry Pi Zero
- DFR0331 Romeo for Edison Controller
- DFR0453 DFRobot CurieNano - A mini Genuino Arduino 101 Board
- TEL0110 CurieCore intel® Curie Neuron Module
- DFR0478 FireBeetle ESP32 IOT Microcontroller(V3.0) Supports Wi-Fi & Bluetooth
- DFR0483 FireBeetle Covers-Gravity I O Expansion Shield
- FireBeetle Covers-24×8 LED Matrix
- TEL0121 FireBeetle Covers-LoRa Radio 433MHz
- TEL0122 FireBeetle Covers-LoRa Radio 915MHz
- TEL0125 FireBeetle Covers LoRa Radio 868MHz
- DFR0489 FireBeetle ESP8266 IOT Microcontroller
- DFR0492 FireBeetle Board-328P with BLE4.1
- DFR0498 FireBeetle Covers-Camera&Audio Media Board
- DFR0507 FireBeetle Covers-OLED12864 Display
- DFR0508 FireBeetle Covers-DC Motor & Stepper Driver
- DFR0511 FireBeetle Covers-ePaper Black&White Display Module
- DFR0531 FireBeetle Covers-ePaper Black&White&Red Display Module
- DFR0536 Micro bit Gamepad Expansion Board
- DFR0548 Micro bit Driver Expansion Board
- ROB0148 micro: Maqueen for micro:bit
- ROB0150 Micro bit Circular RGB LED Expansion Board

Board Overview



Description of pin function

	Pin Name	Funtions
1	VCC	Power(3.3V-5V)
2	GND	Negative pole
3	SCLK	Clock
4	MOSI	Data (sent by host and received by slave)
5	CS	Screen piece selected
6	RES	Reset
7	DC	Data/Command

Tutorial

The product adopts SPI communication with onboard GDI interface, which greatly reduces the wiring complexity.

Requirements

- Hardware
 - ESP32-E (or similar) x 1
 - 1.51" Transparent OLED Display Module x1
 - Dupont Wires

MBT0005 micro IO-BOX

SEN0159 CO2 Sensor

DFR0049 DFRobot Gas Sensor

TOY0058 Barometric Pressure Sensor

SEN0220 Infrared CO2 Sensor 0-50000ppm

SEN0219 Gravity Infrared CO2 Sensor For Arduino

SEN0226 Gravity I2C BMP280 Barometer Sensor

SEN0231 Gravity HCHO Sensor

SEN0251 Gravity BMP280 Barometric Pressure Sensors

SEN0132 Carbon Monoxide Gas Sensor MQ7

SEN0032 Triple Axis Accelerometer Breakout - ADXL345

DFR0143 Triple Axis Accelerometer MMA7361

Triple Axis Accelerometer FXLN83XX Series

SEN0072 CMPS09 - Tilt Compensated Magnetic Compass

SEN0073 9 Degrees of Freedom - Razor IMU

DFR0188 Flymaple V1.1

SEN0224 Gravity I2C Triple Axis Accelerometer - LIS2DH

SEN0140 10 DOF Mems IMU Sensor V2.0

SEN0250 Gravity BMI160 6-Axis Inertial Motion Sensor

SEN0253 Gravity BNO055 + BMP280 intelligent 10DOF AHRS

SEN0001 URM37 V5.0 Ultrasonic Sensor

SEN0002 URM04 V2.0

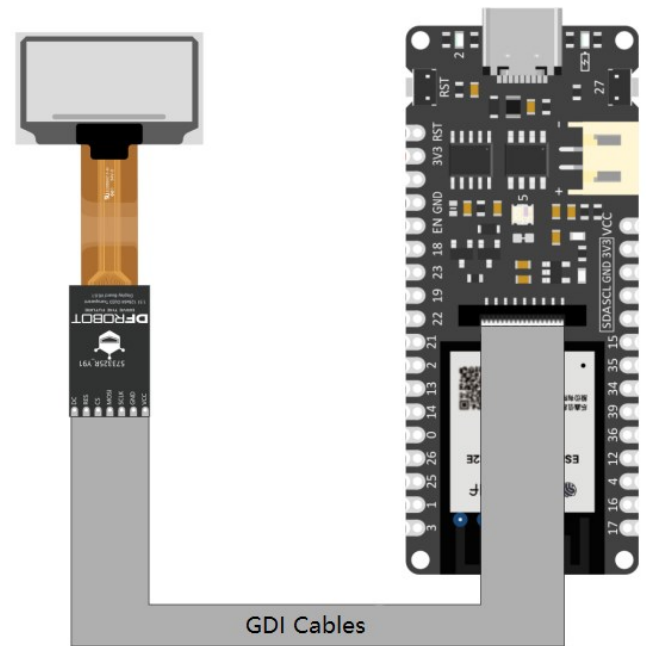
SEN0004 SRF01 Ultrasonic sensor

SEN0005 SRF02 Ultrasonic sensor

- Software

- [Arduino IDE](#)
- Download and install the [U8g2 Arduino Library](#) ([About how to install the library?](#))

Connection Diagram



- Sample Code:

- [Sample Code 1-CommonIcon.ino](#)
- [Sample Code 2-Cube.ino](#)
- [Sample Code 3-Font.ino](#)
- [Sample Code 4-FontDirection.ino](#)
- [Sample Code 5-GraphicsTest.ino](#)
- [Sample Code 6-HelloWorld.ino](#)
- [Sample Code 7-Language.ino](#)

Note:

1.The demos of this product are all stored in the file U8g2_Arduino-master> DFRobot_Demo> 1.51 inch OLED12864-SSD1309.

2.Before burning the Demo, please open the corresponding materialization function (U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2).

```
//U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2 (/* rotation=0 */ U8G2_R0, /* reset=1 */ U8G2_F128X64) // M0/ESP32/ESP8266/mega2560/Pro/Leonardo
//if defined ARDUINO_SAM_ZERO
#define OLED_DC 7
#define OLED_CS 5
#define OLED_RST 6
//ESP8266 and ESP32
#elif defined(ESP32) || defined(ESP8266)
#define OLED_DC D0
#define OLED_CS D4
#define OLED_RST D3
//AVR: see the board!
#else
#define OLED_DC 2
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2 (/* rotation=0 */ U8G2_R0, /* cs=1 */ OLED_CS, /* dc=1 */ OLED_DC, /* reset=1 */ OLED_RST);
```

Sample Code 1-CommonIcon.ino

SEN0006 SRF05 Ultrasonic sensor

SEN0007 SRF08 Ultrasonic Sensor

SEN0008 SRF10 Ultrasonic sensor

SEN0149 URM06-RS485 Ultrasonic

SEN0150 URM06-UART Ultrasonic

SEN0151 URM06-PULSE Ultrasonic

SEN0152 URM06-ANALOG
Ultrasonic

SEN0153 URM07-UART Ultrasonic
Sensor

SEN0246 URM08-RS485 Waterproof
Sonar Range Finder

SEN0304 URM09 Ultrasonic Sensor
(Gravity-I2C) (V1.0)

SEN0304 URM09 Ultrasonic Sensor
(Gravity-I2C) (V1.0)

SEN0300 Water-proof Ultrasonic
Sensor ULS

SEN0301 Water-proof Ultrasonic
Sensor ULA

SEN0307 URM09 Ultrasonic Sensor
Gravity Analog

SEN0311 A02YYUW Waterproof
Ultrasonic Sensor

SEN0312 ME007YS Waterproof
Ultrasonic Sensor

SEN0313 A01NYUB Waterproof
Ultrasonic Sensor

DFR0066 SHT1x Humidity and
Temperature Sensor

DFR0067 DHT11 Temperature and
Humidity Sensor

SEN0137 DHT22 Temperature and
humidity module

DFR0023 DFRobot LM35 Linear
Temperature Sensor

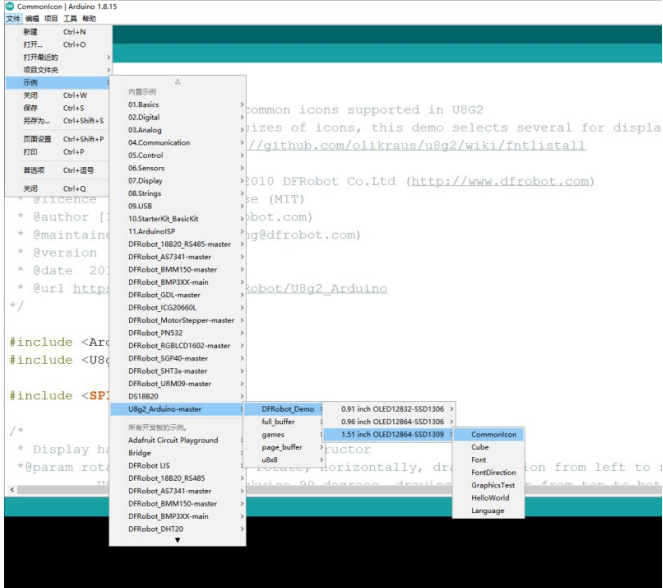
DFR0024 Gravity DS18B20
Temperature Sensor Arduino
Compatible V2

DFR0024 Gravity DS18B20
Temperature Sensor Arduino
Compatible V2

SEN0114 Moisture Sensor

TOY0045 TMP100 Temperature
Sensor

TOY0054 SI7021 Temperature and



```
/*!  
 * @file CommonIcon.ino  
 * @brief Display of several common icons supported  
 * @n U8G2 supports multiple sizes of icons, this d  
 * @n U8G2 GitHub Link:https://github.com/olikraus/  
 *  
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (h  
 * @licence The MIT License (MIT)  
 * @author [Ivey](Ivey.lu@dfrobot.com)  
 * @maintainer [Fary](feng.yang@dfrobot.com)  
 * @version V1.0  
 * @date 2019-10-15  
 * @url https://github.com/DFRobot/U8g2_Arduino  
 */
```

```
#include <Arduino.h>  
#include <U8g2lib.h>
```

```
#include <SPI.h>
```

```
/*  
 * Display hardware IIC interface constructor  
 * @param rotation:U8G2_R0 Not rotate, horizontally,  
 * U8G2_R1 Rotate clockwise 90 degrees, dra  
 * U8G2_R2 Rotate 180 degrees clockwise, dr  
 * U8G2_R3 Rotate clockwise 270 degrees, dr  
 * U8G2_MIRROR Normal display of mirror con  
 * Note: U8G2_MIRROR need to be used with s  
 * @param reset:U8x8_PIN_NONE Indicates that the pin  
 * Display hardware SPI interface constructor  
 * @param Just connect the CS pin (pins are optiona  
 * @param Just connect the DC pin (pins are optiona  
 *  
 */
```

```
//U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(/* rotat  
#if defined ARDUINO_SAM_ZERO  
#define OLED_DC 7  
#define OLED_CS 5  
#define OLED_RST 6  
/*ESP32 */  
#elif defined(ESP32)  
#define OLED_DC D2  
#define OLED_CS D6  
#define OLED_RST D3  
/*ESP8266*/  
#elif defined(ESP8266)  
#define OLED_DC D4  
#define OLED_CS D6  
#define OLED_RST D5  
/*AVR series board*/  
#else  
#define OLED_DC 2
```

SEN0206 IR Thermometer Sensor
MLX90614

SEN0227 SHT20 I2C Temperature &
Humidity Sensor Waterproof Probe

SEN0236 Gravity I2C BME280
Environmental Sensor Temperature,
Humidity, Barometer

SEN0248 Gravity I2C BME680
Environmental Sensor VOC,
Temperature, Humidity, Barometer

DFR0558 Gravity Digital High
Temperature Sensor K-type

SEN0308 Waterproof Capacitive Soil
Moisture Sensor

SEN0019 Adjustable Infrared Sensor
Switch

SEN0042 DFRobot Infrared sensor
breakout

SEN0143 SHARP GP2Y0A41SK0F
IR ranger sensor 4-30cm

SEN0013 Sharp GP2Y0A02YK IR
ranger sensor 150cm

SEN0014 Sharp GP2Y0A21 Distance
Sensor 10-80cm

SEN0085 Sharp GP2Y0A710K
Distance Sensor 100-550cm

DFR0094 Digital IR Receiver Module

DFR0095 DIGITAL IR Transmitter
Module

SEN0018 Digital Infrared motion sensor

DFR0107 IR Kit

SEN0264 TS01 IR Thermal Sensor (4-
20mA)

SEN0169 Analog pH Meter Pro

DFR0300-H Gravity: Analog Electrical
Conductivity Sensor(K=10)

DFR0300 Gravity Analog Electrical
Conductivity Sensor Meter V2 K=1

SEN0165 Analog ORP Meter

SEN0161-V2 Gravity Analog pH
Sensor Meter Kit V2

SEN0161 PH meter

SEN0237 Gravity Analog Dissolved
Oxygen Sensor

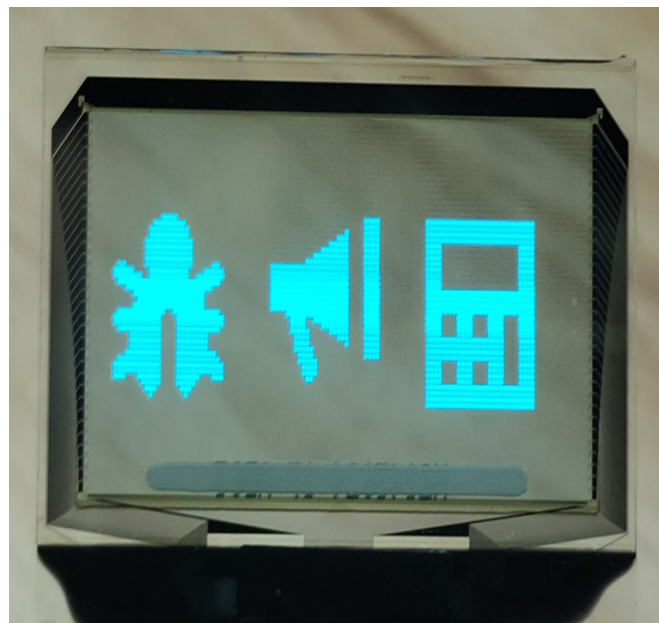
SEN0204 Non-contact Liquid Level

```
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot

void setup(void)
{
    u8g2.begin(); //init
    u8g2.setFontPosTop(); /**When you use drawStr to
    * coordinates of the characters.The function can
    * corner of the display string as the coordinate
}

void loop(void)
{
    /*
    * firstPage will change the current page num
    * When modifications are between firstpage a
    * This method consumes less ram space than s
    */
    u8g2.firstPage();
    for(int i = 64 ; i <287; i++){
        u8g2.clear();
        do
        {
            u8g2.setFont(u8g2_font_open_iconic_all_4x_t )
            /*
            * Draw a single character. The character is
            * U8g2 supports the lower 16 bit of the unic
            * The encoding can be any value from 0 to 65
            */
            u8g2.drawGlyph(/* x=*/4, /* y=*/16, /* encodi
            u8g2.drawGlyph(/* x=*/44, /* y=*/16, /* encod
            u8g2.drawGlyph(/* x=*/84, /* y=*/16, /* encod
        } while ( u8g2.nextPage() );
        i = i+3;
        delay(2000);
    }
}
```

Result(There are many demo screens in this example. The following are some pictures in the demo screen):



Sample Code 2-Cube.ino

Sensor XKC-Y25-T12V

SEN0205 Liquid Level Sensor-FS-IR02

SEN0244 Gravity Analog TDS Sensor
Meter For Arduino

SEN0249 Gravity Analog Spear Tip pH
Sensor Meter Kit For Soil And Food
Applications

SEN0121 Steam Sensor

SEN0097 Light Sensor

DFR0026 DFRobot Ambient Light
Sensor

TOY0044 UV Sensor

SEN0172 LX1972 ambient light sensor

SEN0043 TEMENT6000 ambient light
sensor

SEN0175 UV Sensor v1.0-ML8511

SEN0228 Gravity I2C VEML7700
Ambient Light Sensor

SEN0101 TCS3200 Color Sensor

DFR0022 DFRobot Grayscale Sensor

SEN0017 Line Tracking Sensor for
Arduino V4

SEN0147 Smart Grayscale sensor

SEN0212 TCS34725 I2C Color Sensor
For Arduino

SEN0245 Gravity VL53L0X ToF Laser
Range Finder

SEN0259 TF Mini LiDAR ToF Laser
Range Sensor

SEN0214 20A Current Sensor

SEN0262 Gravity Analog Current to
Voltage Converter for 4~20mA
Application

SEN0291 Gravity: I2C Digital
Wattmeter

DFR0027 DFRobot Digital Vibration
Sensor V2

DFR0028 DFRobot Tilt Sensor

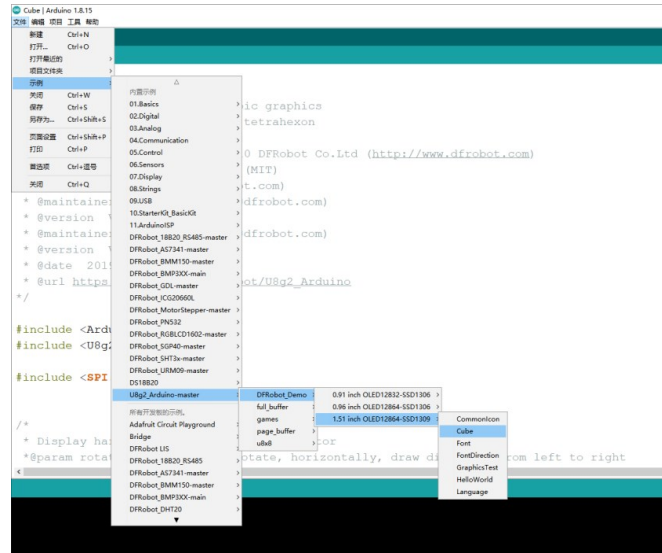
DFR0029 DFRobot Digital Push Button

DFR0030 DFRobot Capacitive Touch
Sensor

DFR0032 Digital Buzzer Module

DFR0033 Digital magnetic sensor

DFR0034 Analog Sound Sensor



```
/*!  
 * @file Cube.ino  
 * @brief Rotating 3D stereoscopic graphics  
 * @n This is a simple rotating tetrahexon  
 *  
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)  
 * @licence The MIT License (MIT)  
 * @author [Ivey](Ivey.lu@dfrobot.com)  
 * @maintainer [Fary](feng.yang@dfrobot.com)  
 * @version V1.0  
 * @maintainer [Fary](feng.yang@dfrobot.com)  
 * @version V1.0  
 * @date 2019-10-15  
 * @url https://github.com/DFRobot/U8g2_Arduino  
 */
```

```
#include <Arduino.h>  
#include <U8g2lib.h>
```

```
#include <SPI.h>
```

```
/*  
 * Display hardware IIC interface constructor  
 * @param rotation:U8G2_R0 Not rotate, horizontally,  
 * U8G2_R1 Rotate clockwise 90 degrees, draw  
 * U8G2_R2 Rotate 180 degrees clockwise, draw  
 * U8G2_R3 Rotate clockwise 270 degrees, draw  
 * U8G2_MIRROR Normal display of mirror con  
 * Note: U8G2_MIRROR need to be used with s  
 * @param reset:U8x8_PIN_NONE Indicates that the pin  
 * Display hardware SPI interface constructor  
 * @param Just connect the CS pin (pins are optiona  
 * @param Just connect the DC pin (pins are optiona  
 *  
 */
```

```
#if defined ARDUINO_SAM_ZERO
```

```
#define OLED_DC 7
```

```
#define OLED_CS 5
```

```
#define OLED_RST 6
```

```
/*ESP32 */
```

```
#elif defined(ESP32)
```

```
#define OLED_DC D2
```

```
#define OLED_CS D6
```

```
#define OLED_RST D3
```

```
/*ESP8266*/
```

```
#elif defined(ESP8266)
```

```
#define OLED_DC D4
```

```
#define OLED_CS D6
```

```
#define OLED_RST D5
```

```
/*AVR series board*/
```

```
#else
```

```
#define OLED_DC 2
```

SEN0038 Wheel Encoders for
DFRobot 3PA and 4WD Rovers

DFR0051 Analog Voltage Divider

DFR0052 Analog Piezo Disk Vibration
Sensor

DFR0076 Flame sensor

DFR0053 Analog Slide Position Sensor

DFR0054 Analog Rotation Sensor V1

DFR0058 Analog Rotation Sensor V2

DFR0061 Joystick Module For Arduino

DFR0075 ADKeyboard Module

DFR0332 Fan Module

SEN0177 PM2.5 laser dust sensor

SEN0160 Weight Sensor Module

SEN0170 Wind Speed Sensor Voltage
Type 0-5V

TOY0048 High Accuracy Dual Axis
Inclinometer Sensor Arduino Gadgeteer
Compatible

SEN0187 RGB and Gesture Sensor

SEN0186 Weather Station with
Anemometer Wind vane Rain bucket

SEN0192 MicroWave Sensor

SEN0185 Hall sensor

FIT0449 DFRobot Speaker v1.0

SEN0203 Heart Rate Sensor

DFR0423 Self-Locking Swicth

SEN0213 Heart Rate Monitor Sensor

SEN0221 Gravity Hall Angle Sensor

SEN0223 Conductivity Switch Sensor

SEN0230 Incremental Photoelectric
Rotary Encoder - 400P R

SEN0235 EC11 Rotary Encoder
Module

SEN0240 Analog EMG Sensor by
OYMotion

SEN0232 Gravity Analog Sound Level
Meter

SEN0233 Air Quality Monitor PM 2.5,
Formaldehyde, Temperature & Humidity
Sensor

```
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot

//2D array: The coordinates of all vertices of the
double tetrahedron[4][3] = {{0,20,-20},{-20,-20,-20},
void setup(void) {
    u8g2.begin();
}

void loop(void) {
    /*
        * firstPage will change the current page num
        * When modifications are between firstpage a
        * This method consumes less ram space than s
    */
    u8g2.firstPage();
    do {
        //Connect the corresponding points inside the tet
        u8g2.drawLine(OxyzToOu(tetrahedron[0][0], tetrahe
        u8g2.drawLine(OxyzToOu(tetrahedron[1][0], tetrahe
        u8g2.drawLine(OxyzToOu(tetrahedron[0][0], tetrahe
        u8g2.drawLine(OxyzToOu(tetrahedron[0][0], tetrahe
        u8g2.drawLine(OxyzToOu(tetrahedron[1][0], tetrahe
        u8g2.drawLine(OxyzToOu(tetrahedron[2][0], tetrahe
        // Rotate 0.1°
        rotate(0.1);

    } while ( u8g2.nextPage() );
    //delay(50);
}
/* !
 * @brief Convert xz in the three-dimensional coord
 * into the u coordinate inside the two-dimensional
 * @param x in Oxyz
 * @param z in Oxyz
 * @return u in Ouv
 */
int OxyzToOu(double x,double z){

    return (int)((x + 64) - z*0.35);
}

/* !
 * @brief Convert the yz in the three-dimensional c
 * the two-dimensional coordinate system Ouv
 * @param y in Oxyz
 * @param z in Oxyz
 * @return v in Ouv
 */
int OxyzToOv(double y,double z){
    return (int)((y + 26) - z*0.35);
}

/* !
 * @brief Rotate the coordinates of all points of
 * @param angle represents the angle to rotate
 *
 * z rotation (z unchanged)
    x3 = x2 * cosb - y1 * sinb
    y3 = y1 * cosb + x2 * sinb
    z3 = z2
 */
void rotate(double angle)
{
    double rad, cosa, sina, Xn, Yn;
```


DFR0327 Arduino Shield for Raspberry Pi 2B and 3B

DFR0371 IO Expansion Shield for Bluno M3

DFR0356 Bluno Beetle Shield

DFR0412 Gravity IO Expansion Shield For DFRduino M0

DFR0375 Cookie I O Expansion Shield V2

DFR0334 GPIO Shield for Arduino V1.0

DFR0502 Gravity IO Expansion & Motor Driver Shield V1.1

DFR0518 Micro Mate- A Mini Expansion Board for micro bit

DFR0578 Gravity I O Expansion Shield for OpenMV Cam M7

DFR0577 Gravity I O Expansion Shield for Pyboard

DFR0626 MCP23017 IIC to 16 digital IO expansion module

DFR0287 LCD12864 Shield

DFR0009 LCD KeyPad Shield For Arduino

DFR0063 I2C TWI LCD1602 Module Gadgeteer Compatible

DFR0154 I2C TWI LCD2004 Module Arduino Gadgeteer Compatible

DFR0202 RGB LED Matrix

DFR0090 3-Wire LED Module

TOY0005 OLED 2828 color display module .NET Gadgeteer Compatible

TOY0006 OLED 9664 RGB Display module

TOY0007 OLED 2864 display module

FIT0328 2.7 OLED 12864 display module

DFR0091 3-wire Serial LCD Module Arduino Compatible

DFR0347 2.8 TFT Touch Shield with 4MB Flash for Arduino and mbed

DFR0348 3.5 TFT Touch Shield with 4MB Flash for Arduino and mbed

DFR0374 LCD Keypad Shield V2.0

DFR0382 LED Keypad Shield V1.0

```
* @date 2019-10-15
* @url https://github.com/DFRobot/U8g2_Arduino
*/

#include <Arduino.h>
#include <U8g2lib.h>

#include <SPI.h>

/*
 * Display hardware IIC interface constructor
 *@param rotation:U8G2_R0 Not rotate, horizontally,
                U8G2_R1 Rotate clockwise 90 degrees, dra
                U8G2_R2 Rotate 180 degrees clockwise, dr
                U8G2_R3 Rotate clockwise 270 degrees, dr
                U8G2_MIRROR Normal display of mirror con
                Note: U8G2_MIRROR need to be used with s
 *@param reset:U8x8_PIN_NONE Indicates that the pin
 * Display hardware SPI interface constructor
 *@param Just connect the CS pin (pins are optiona
 *@param Just connect the DC pin (pins are optiona
 */
*/
#if defined ARDUINO_SAM_ZERO
#define OLED_DC 7
#define OLED_CS 5
#define OLED_RST 6
/*ESP32 */
#elif defined(ESP32)
#define OLED_DC D2
#define OLED_CS D6
#define OLED_RST D3
/*ESP8266*/
#elif defined(ESP8266)
#define OLED_DC D4
#define OLED_CS D6
#define OLED_RST D5
/*AVR series board*/
#else
#define OLED_DC 2
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot

void setup(void)
{
    u8g2.begin();           //Initialize the function
    u8g2.setFontPosTop();   /**When you use drawStr to
 * coordinates of the characters.The function can
 * corner of the display string as the coordinate
}

void draw(int a )
{
    switch(a)
    {
        case 0:
            u8g2.setFont(u8g2_font_bubble_tr ); //Set
            u8g2.drawStr(/* x=*/0,/* y=*/0, "DFR123");
            u8g2.setFont(u8g2_font_lucasarts_scumm_subtitl
            u8g2.drawStr(0, 25, "DFR123");
            u8g2.setFont(u8g2_font_HelvetiPixelOutline_tr
            u8g2.drawStr(0, 45, "DFR123");
            break;
        case 1:
            u8g2.setFont(u8g2_font_tenstamps_mr );
            u8g2.drawStr(0,0, "DFR123");
            u8g2.setFont(u8g2_font_jinxedwizards_tr )
            u8g2.drawStr(56, 16, "DFR123");
```

DFR0387 TELEMATICS 3.5 TFT Touch LCD Shield

DFR0459 8x8 RGB LED Matrix

DFR0460 64x32 RGB LED Matrix -
4mm Pitch/64x32 Flexible RGB LED
Matrix-4mm Pitch/64x32 Flexible RGB
LED Matrix-5mm Pitch

DFR0461 Gravity Flexible 8x8 RGB LED Matrix

DFR0462 Gravity Flexible 8x32 RGB LED Matrix

DFR0463 Gravity Flexible 16x16 RGB LED Matrix

DFR0471 32x16 RGB LED Matrix -
6mm pitchDFR0472 32x32 RGB LED Matrix -
4mm pitch

DFR0464 Gravity I2C 16x2 Arduino LCD with RGB Backlight Display

DFR0499 64x64 RGB LED Matrix - 3mm pitch

DFR0506 7" HDMI Display with Capacitive Touchscreen

DFR0555\DF0556\DFR0557 Gravity I2C LCD1602 Arduino LCD Display Module

DFR0529 2.2 inches TFT LCD Display V1.0 (SPI Interface)

DFR0605 Gravity: Digital RGB LED Module

FIT0352 Digital RGB LED
Weatherproof Strip 60LED m*3m

DFR0645-G DFR0645-R 4-Digital LED Segment Display Module

SKU DFR0646-G DFR0646-R 8-Digital LED Segment Display Module

DFR0597 7x71 Flexible RGB LED Matrix

DFR0231 NFC Module for Arduino

TEL0005 APC220 Radio Data Module

TEL0023 BLUETOOTH BEE

TEL0026 DF-BluetoothV3 Bluetooth
module

TEL0037 Wireless Programming Module For Arduino

TEL0044 DFRduino GPS Shield-LEA-5H

```

    u8g2.setFont(u8g2_font_secretaryhand_tr
    u8g2.drawStr(0, 32, "DFR123");
    u8g2.setFont(u8g2_font_freedoomr10_mu
    u8g2.drawStr(56, 48, "DFR123");
    break;
}
}

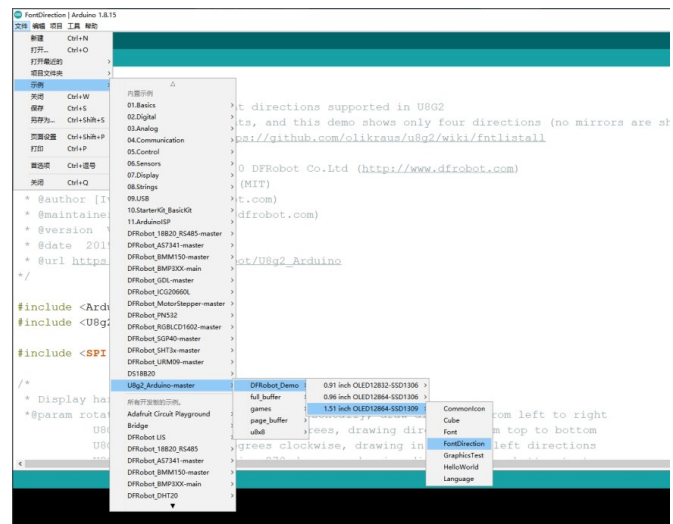
void loop()
{
    for( int i = 0; i <2 ; i++)
    {
        /*
        * firstPage will change the current page num
        * When modifications are between firstpage a
        * This method consumes less ram space than s
        */
        u8g2.firstPage();
        do
        {
            draw(i);
        } while( u8g2.nextPage() );
        delay(2000);
    }
}
}

```

Result:



Sample Code 4-FontDirection.ino



TEL0047 WiFi Shield V2.1 For Arduino

TEL0051 GPS GPRS GSM Module
V2.0

TEL0067 WiFi Bee V1.0

TEL0073 BLE-Link

TEL0075 RF Shield 315MHz

TEL0078 WIFI Shield V3 PCB
Antenna

TEL0079 WIFI Shield V3 RPSMA

TEL0084 BLEmicro

TEL0086 DF-Beacon EVB

TEL0087 USBBLE-LINK Bluno
Wireless Programming Adapter

TEL0080 UHF RFID MODULE-USB

TEL0081 UHF RFID MODULE-
RS485

TEL0082 UHF RFID MODULE-
UART

TEL0083-A GPS Receiver for Arduino
Model A

TEL0092 WiFi Bee-ESP8266 Wirelss
module

TEL0094 GPS Module With Enclosure

TEL0097 SIM808 GPS GPRS GSM
Shield

DFR0342 W5500 Ethernet with POE
Mainboard

DFR0015 Xbee Shield For Arduino no
Xbee

TEL0107 WiFiBee-MT7681 Arduino
WiFi Wireless Programming

TEL0089 SIM800C GSM GPRS Shield
V2.0

TEL0112 Gravity 315MHZ RF Receiver
Module

TEL0113 Gravity UART A6 GSM &
GPRS Module

TEL0118 Gravity UART OBLOQ IoT
Module

TEL0120 DFRobot BLE4.1 Module

TEL0002 Bluetooth Adapter

TEL0108 Bluetooth Audio Receiver
Module

```
/*!  
 * @file FontDirection.ino  
 * @brief Display of several font directions support  
 * @n U8G2 supports multiple fonts, and this demo s  
 * @n U8G2 Font GitHub Link:https://github.com/olik  
 *  
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (h  
 * @licence The MIT License (MIT)  
 * @author [Ivey](Ivey.lu@dfrobot.com)  
 * @maintainer [Fary](feng.yang@dfrobot.com)  
 * @version V1.0  
 * @date 2019-10-15  
 * @url https://github.com/DFRobot/U8g2_Arduino  
 */
```

```
#include <Arduino.h>  
#include <U8g2lib.h>
```

```
#include <SPI.h>
```

```
/*  
 * Display hardware IIC interface constructor  
 * @param rotation:U8G2_R0 Not rotate, horizontally,  
 U8G2_R1 Rotate clockwise 90 degrees, dra  
 U8G2_R2 Rotate 180 degrees clockwise, dr  
 U8G2_R3 Rotate clockwise 270 degrees, dr  
 U8G2_MIRROR Normal display of mirror con  
 Note: U8G2_MIRROR need to be used with s  
 * @param reset:U8x8_PIN_NONE Indicates that the pin  
 * Display hardware SPI interface constructor  
 * @param Just connect the CS pin (pins are optiona  
 * @param Just connect the DC pin (pins are optiona  
 *  
 */
```

```
#if defined ARDUINO_SAM_ZERO
```

```
#define OLED_DC 7
```

```
#define OLED_CS 5
```

```
#define OLED_RST 6
```

```
/*ESP32 */
```

```
#elif defined(ESP32)
```

```
#define OLED_DC D2
```

```
#define OLED_CS D6
```

```
#define OLED_RST D3
```

```
/*ESP8266*/
```

```
#elif defined(ESP8266)
```

```
#define OLED_DC D4
```

```
#define OLED_CS D6
```

```
#define OLED_RST D5
```

```
/*AVR series board*/
```

```
#else
```

```
#define OLED_DC 2
```

```
#define OLED_CS 3
```

```
#define OLED_RST 4
```

```
#endif
```

```
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot
```

```
//width:30,height:30
```

```
const uint8_t col[] U8X8_PROGMEM= {0x00,0xc0,0x00,0  
0xc0,0x00,  
0x01,0x00,  
0x38,0xdf,  
0xe0,0xff,  
0xff,0xff,  
0x60,0x01,  
0x01,0xe0,  
0x00,0xfe,  
0xc0,0x00,
```

```
void setup() {
```

TEL0124 SIM7600CE-T 4G(LTE)
Shield V1.0

DFR0505 SIM7000C Arduino NB-IoT
LTE GPRS Expansion Shield

DFR0013 IIC to GPIO Shield V2.0

DFR0057 Sensor Motor Drive Board -
Version 2.2

DFR0062 WiiChuck Adapter

DFR0233 RS485 Sensor Node V1.0

DFR0259 Arduino RS485 Shield

DFR0370 CAN-BUS Shield V2

DFR0627 IIC to dual UART module

TEL0070 Multi USB RS232 RS485
TTL Converter

DFR0064 386AMP audio amplifier
Module

DFR0273 Speech Synthesis Shield

DFR0299 DFPlayer Mini

TOY0008 DFRduino Player MP3

SEN0197 Voice Recorder-ISD1820

DFR0420 Audio Shield For DFRduino
M0

DFR0534 Voice Module

SD2403 Real time clock Module SKU
TOY0020

TOY0021 SD2405 Real time clock
Module

DFR0151 Gravity I2C DS1307 RTC
Module

DFR0469 Gravity I2C SD2405 RTC
Module

DFR0316 MCP3424 18-Bit ADC-4
Channel with Programmable Gain
Amplifier

DFR0552 Gravity 12-Bit I2C DAC
Module

DFR0553 Gravity I2C ADS1115 16-Bit
ADC Module Arduino & Raspberry Pi
Compatible

DFR0117 Gravity I2C EEPROM Data
Storage Module

DFR0071 SD Module

```
u8g2.begin();
u8g2.setFontPosTop();/**When you use drawStr to d
* coordinates of the characters.The function can
* corner of the display string as the coordinate
}

void Rotation() {
  u8g2.setFont(u8g2_font_bracketedbabies_tr);
  u8g2.firstPage();
  do {
    u8g2.drawXBMP( /* x=*/0 , /* y=*/0 , /* width=*/
/*@brief Set the drawing direction of all stri
*@param dir=0
            dir=1, rotate 0°
            dir=2, rotate 180°
            dir=3, rotate 270°

    /*
    u8g2.setFontDirection(0);
    u8g2.drawStr( /* x=*/64,/* y=*/32, " DFR");
    u8g2.setFontDirection(1);
    u8g2.drawStr(64,32, " DFR");
    u8g2.setFontDirection(2);
    u8g2.drawStr(64,32, " DFR");
    u8g2.setFontDirection(3);
    u8g2.drawStr(64,32, " DFR");
  }while(u8g2.nextPage());
  delay(1500);
}

void loop(void)
{
  u8g2.setFont( u8g2_font_sticker_mel_tr);
  for(int i = 0 ; i < 4 ; i++ )
  {
    switch(i)
    {
      case 0:
        u8g2.setFontDirection(0);
        break;
      case 1:
        u8g2.setFontDirection(1);
        break;
      case 2:
        u8g2.setFontDirection(2);
        break;
      case 3:
        u8g2.setFontDirection(3);
        break;
    }
    /*
    * firstPage will change the current page num
    * When modifications are between firstpage a
    * This method consumes less ram space than s
    */
    u8g2.firstPage();
    do
    {
      u8g2.drawStr(64, 32, "DFR");
      u8g2.drawXBMP( 0 , 0 , 30 , 30 , col );
    }while(u8g2.nextPage());
    delay(500);
  }
  Rotation();
}
```

Result:

DFR0057 Sensor Motor Drive Board
Version 2.2

DFR0360 XSP - Arduino Programmer

DFR0411 Gravity 130 DC Motor

DFR0438 Bright LED Module

DFR0439 LED String Lights Colorful

DFR0440 The Micro Vibration Module

DFR0448 LED String Lights Warm
White

DFR0503 Embedded Thermal Printer -
TTL Serial

DFR0504 Gravity Analog Signal Isolator

DFR0520 Dual Digital Pot 100K

DFR0565 Gravity Digital Signal Isolator

DFR0563 Gravity 3.7V Li Battery Fuel
Gauge

DFR0576 Gravity Digital 1-to-8 I2C
Multiplexer

DFR0117 Gravity I2C EEPROM Data
Storage Module

DRI0001 Arduino Motor Shield L293

DRI0002 MD1.3 2A Dual Motor
Controller

DRI0009 Arduino Motor Shield L298N

DRI0021 Veyron 2x25A Brush DC
Motor Driver

DRI0017 2A Motor Shield For Arduino
Twin

DRI0018 DC Motor Driver 2x15A Lite

FIT0450 Micro DC Motor with
Encoder-SJ01

FIT0458 Micro DC Motor with
Encoder-SJ02

DFR0399 DC Micro Metal Gear Motor
75 1 w Driver

DRI0039 Quad Motor Driver Shield for
Arduino

DRI0040 Dual 1.5A Motor Driver -
HR8833

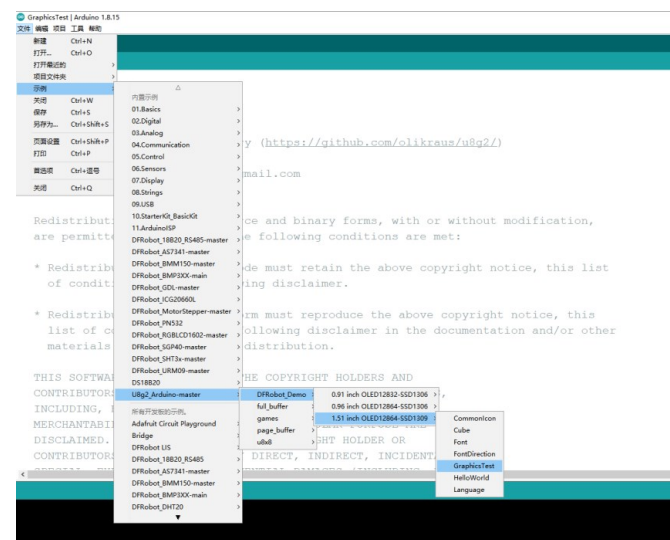
DRI0044 2x1.2A DC Motor Driver
TB6612FNG

DFR0513 PPM 2x3A DC Motor Driver

DFR0523 Gravity Digital Peristaltic
Pump



Sample Code 5-GraphicsTest.ino



```
/*
```

GraphicsTest.ino

Universal 8bit Graphics Library (<https://github.com>)

Copyright (c) 2016, olikraus@gmail.com
All rights reserved.

Redistribution and use in source and binary forms
are permitted provided that the following conditions

* Redistributions of source code must retain the
of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the
list of conditions and the following disclaimer
materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION;
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

DR10027 Digital Servo Shield for Arduino

DR10029 Veyron Servo Driver 24-Channel

SER0044 DSS-M15S 270° 15KG DF Metal Servo with Analog Feedback

DR10023 Stepper Motor Shield For Arduino DRV8825

DR10035 TMC260 Stepper Motor Driver Shield

DFR0105 Power Shield

DFR0205 Power Module

DFR0457 Gravity MOSFET Power Controller

DFR0564 USB Charger for 7.4V LiPo Battery

DFR0535 Solar Power Manager

DFR0559 Sunflower Solar Power Manager 5V

DFR0559 Solar Power Manager 5V

DFR0580 Solar Power Manager For 12V Lead-Acid Battery

DFR0222 X-Board Relay

DFR0017 Relay Module Arduino Compatible

DFR0289 RLY-8-POE Relay Controller

DFR0290 RLY-8-RS485 8 Relay Controller

DFR0144 Relay Shield for Arduino V2.1

DFR0473 Gravity Digital Relay Module Arduino & Raspberry Pi Compatible

KIT0003 EcoDuino - An Auto Plant Kit

KIT0071 MiniQ Discovery Kit

KIT0098 Breadboard Plugin Components Pack

SKU DFR0748 Kitty Flower

SEN0305 Gravity: HUSKYLENS - An Easy-to-use AI Machine Vision Sensor

Sensor connection to Raspberry Pi

DFR0677 ONPOWER UPS HAT for Raspberry Pi

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

#include <Arduino.h>

#include <U8g2lib.h>

#include <SPI.h>

/*

U8g2lib Example Overview:

Frame Buffer Examples: clearBuffer/sendBuffer.

Page Buffer Examples: firstPage/nextPage. Less

U8x8 Text Only Example: No RAM usage, direct co

This is a page buffer example.

*/

// Please UNCOMMENT one of the constructor lines bel

// U8g2 Constructor List (Picture Loop Page Buffer)

// The complete list is available here: <https://git>

// Please update the pin numbers according to your

#if defined ARDUINO_SAM_ZERO

#define OLED_DC 7

#define OLED_CS 5

#define OLED_RST 6

/*ESP32 */

#elif defined(ESP32)

#define OLED_DC D2

#define OLED_CS D6

#define OLED_RST D3

/*ESP8266*/

#elif defined(ESP8266)

#define OLED_DC D4

#define OLED_CS D6

#define OLED_RST D5

/*AVR series board*/

#else

#define OLED_DC 2

#define OLED_CS 3

#define OLED_RST 4

#endif

U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot

// End of constructor list

void u8g2_prepare(void) {

u8g2.setFont(u8g2_font_6x10_tf); //Set the fon

u8g2.setFontRefHeightExtendedText(); //Ascent

u8g2.setDrawColor(1); //Ascent will be the lar

u8g2.setFontPosTop(); //When you use drawStr

XXXX */

u8g2.setFontDirection(0); //Set the screen ori

}

void u8g2_box_title(uint8_t a) {

u8g2.drawStr(10+a*2, 5, "U8g2"); //Draw string "

u8g2.drawStr(10, 20, "GraphicsTest");

u8g2.drawFrame(0,0,u8g2.getDisplayWidth(),u8g2.ge

}

/*

* Draw solid squares and hollow squares

*/

void u8g2_box_frame(uint8_t a) {

```

u8g2.drawStr( 0, 0, "drawBox");
u8g2.drawBox(5,10,20,10);    //Start drawing a so
u8g2.drawBox(10+a,15,30,7);
u8g2.drawStr( 0, 30, "drawFrame");
u8g2.drawFrame(5,10+30,20,10);
u8g2.drawFrame(10+a,15+30,30,7);
}

/*
 * Draw solid and hollow circles
 */
void u8g2_disc_circle(uint8_t a) {
    u8g2.drawStr( 0, 0, "drawDisc");
    u8g2.drawDisc(10,18,9);    //Draw a solid circle
    u8g2.drawDisc(24+a,16,7);
    u8g2.drawStr( 0, 30, "drawCircle");
    u8g2.drawCircle(10,18+30,9);    //Draw a hollow c
    u8g2.drawCircle(24+a,16+30,7);
}

/*
 * Draw solid and hollow boxes
 */
void u8g2_r_frame(uint8_t a) {
    u8g2.drawStr( 0, 0, "drawRFrame/Box");
    u8g2.drawRFrame(5, 10,40,30, a+1);    //At the po
    u8g2.drawRBox(50, 10,25,40, a+1);    //At the
}

void u8g2_string(uint8_t a) {
    /*@brief Set the drawing direction of all strings c
        *@param dir=0
                dir=1, rotate 0°
                dir=2, rotate 180°
                dir=3, rotate 270°
        */
    u8g2.setFontDirection(0);
    u8g2.drawStr(30+a,31, " 0");
    u8g2.setFontDirection(1);
    u8g2.drawStr(30,31+a, " 90");
    u8g2.setFontDirection(2);
    u8g2.drawStr(30-a,31, " 180");
    u8g2.setFontDirection(3);
    u8g2.drawStr(30,31-a, " 270");
}

/*
 * Draw segments
 */
void u8g2_line(uint8_t a) {
    u8g2.drawStr( 0, 0, "drawLine");
    u8g2.drawLine(7+a, 10, 40, 55);    //Draw a line
    u8g2.drawLine(7+a*2, 10, 60, 55);
    u8g2.drawLine(7+a*3, 10, 80, 55);
    u8g2.drawLine(7+a*4, 10, 100, 55);
}

/*
 * Draw solid triangles and hollow triangles
 */
void u8g2_triangle(uint8_t a) {
    uint16_t offset = a;
    u8g2.drawStr( 0, 0, "drawTriangle");
    u8g2.drawTriangle(14,7, 45,30, 10,40);    //Draw
    u8g2.drawTriangle(14+offset,7+offset, 45+offset,3
    u8g2.drawTriangle(57+offset*2,10, 45+offset*2,30,
    u8g2.drawTriangle(10+offset,40+offset, 45+offset,
}

/*
 * Show characters in the ASCII code table
 */

```

```

void u8g2_ascii_1() {
    char s[2] = " ";
    uint8_t x, y;
    u8g2.drawStr( 0, 0, "ASCII page 1");
    for( y = 0; y < 6; y++ ) {
        for( x = 0; x < 16; x++ ) {
            s[0] = y*16 + x + 32;
            u8g2.drawStr(x*7, y*10+10, s);
        }
    }
}

void u8g2_ascii_2() {
    char s[2] = " ";
    uint8_t x, y;
    u8g2.drawStr( 0, 0, "ASCII page 2");
    for( y = 0; y < 6; y++ ) {
        for( x = 0; x < 16; x++ ) {
            s[0] = y*16 + x + 160;
            u8g2.drawStr(x*7, y*10+10, s);
        }
    }
}

/*
 * Draw a string icon in a UTF-8 encoding
 */
void u8g2_extra_page(uint8_t a)
{
    u8g2.drawStr( 0, 0, "Unicode");
    u8g2.setFont(u8g2_font_unifont_t_symbols);
    u8g2.setFontPosTop();
    u8g2.drawUTF8(0, 24, "☀️ 🍄"); //Start drawing
    switch(a) {
        case 0:
        case 1:
        case 2:
        case 3:
            u8g2.drawUTF8(a*3, 36, "☂️");
            break;
        case 4:
        case 5:
        case 6:
        case 7:
            u8g2.drawUTF8(a*3, 36, "");
            break;
    }
}

/*
 * Show the reverse display of font. Which means th
 * the background color. (For example, it would hav
 * on black)
 */
void u8g2_xor(uint8_t a) {
    uint8_t i;
    u8g2.drawStr( 0, 0, "XOR");
    u8g2.setFontMode(1);
    u8g2.setDrawColor(2);
    for( i = 0; i < 5; i++ )
    {
        u8g2.drawBox(10+i*16, 18 + (i&1)*4, 21,31);
    }
    u8g2.drawStr( 5+a, 19, "XOR XOR XOR XOR");
    u8g2.setDrawColor(0);
    u8g2.drawStr( 5+a, 29, "CLR CLR CLR CLR");
    u8g2.setDrawColor(1);
    u8g2.drawStr( 5+a, 39, "SET SET SET SET");
    u8g2.setFontMode(0);
}

```

```

/*
 * This is the data of converting a graph to bitmap
 */
#define cross_width 24
#define cross_height 24
static const unsigned char cross_bits[] U8X8_PROGMEM
    0x00, 0x18, 0x00, 0x00, 0x24, 0x00, 0x00, 0x24, 0
    0x00, 0x42, 0x00, 0x00, 0x42, 0x00, 0x00, 0x81, 0
    0xC0, 0x00, 0x03, 0x38, 0x3C, 0x1C, 0x06, 0x42, 0
    0x01, 0x42, 0x80, 0x06, 0x42, 0x60, 0x38, 0x3C, 0
    0x00, 0x81, 0x00, 0x00, 0x81, 0x00, 0x00, 0x42, 0
    0x00, 0x42, 0x00, 0x00, 0x24, 0x00, 0x00, 0x24, 0

#define cross_fill_width 24
#define cross_fill_height 24
static const unsigned char cross_fill_bits[] U8X8_PROGMEM
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x00, 0
    0x84, 0x00, 0x21, 0x08, 0x81, 0x10, 0x08, 0x42, 0
    0x20, 0x00, 0x04, 0x40, 0x00, 0x02, 0x80, 0x00, 0
    0x80, 0x18, 0x01, 0x80, 0x00, 0x01, 0x40, 0x00, 0
    0x10, 0x3C, 0x08, 0x08, 0x42, 0x10, 0x08, 0x81, 0
    0x64, 0x00, 0x26, 0x18, 0x00, 0x18, 0x00, 0x00, 0

#define cross_block_width 14
#define cross_block_height 14
static const unsigned char cross_block_bits[] U8X8_PROGMEM
    0xFF, 0x3F, 0x01, 0x20, 0x01, 0x20, 0x01, 0x20, 0
    0xC1, 0x20, 0xC1, 0x20, 0x01, 0x20, 0x01, 0x20, 0
    0x01, 0x20, 0xFF, 0x3F, };

/*
 * Draw a bitmap
 */
void u8g2_bitmap_overlay(uint8_t a) {
    uint8_t frame_size = 28;

    u8g2.drawStr(0, 0, "Bitmap overlay");

    u8g2.drawStr(0, frame_size + 12, "Solid / transparent");
    u8g2.setBitmapMode(false /* solid */);
    u8g2.drawFrame(0, 10, frame_size, frame_size);
    u8g2.drawXBMP(2, 12, cross_width, cross_height, cross_bits);
    if(a & 4)
        u8g2.drawXBMP(7, 17, cross_block_width, cross_block_height, cross_block_bits);
    /*
     * Set the pattern of the bitmap to define whether
     * (Mode0/solid, is_transparent = 0).
     * Or not write the background color to the bitmap
     * The default mode is 0(fixed mode).
     */
    u8g2.setBitmapMode(true /* transparent */);
    u8g2.drawFrame(frame_size + 5, 10, frame_size, frame_size);
    /*
     * The position (x,y) is the upper-left corner of the frame
     * The current color index is used for drawing (ref to u8g2.h)
     */
    u8g2.drawXBMP(frame_size + 7, 12, cross_width, cross_height, cross_bits);
    if(a & 4)
        u8g2.drawXBMP(frame_size + 12, 17, cross_block_width, cross_block_height, cross_block_bits);
}

void u8g2_bitmap_modes(uint8_t transparent) {
    const uint8_t frame_size = 24;

    u8g2.drawBox(0, frame_size * 0.5, frame_size * 0.5, frame_size * 0.5, "Black");
    u8g2.drawStr(frame_size * 0.5, 50, "Black");
    u8g2.drawStr(frame_size * 2, 50, "White");
    u8g2.drawStr(frame_size * 3.5, 50, "XOR");

    if(!transparent) {
        u8g2.setBitmapMode(false /* solid */);
        u8g2.drawStr(0, 0, "Solid bitmap");
    }
}

```

```

    } else {
        u8g2.setBitmapMode(true /* transparent*/);
        u8g2.drawStr(0, 0, "Transparent bitmap");
    }
    u8g2.setDrawColor(0); // Black
    u8g2.drawXBMP(frame_size * 0.5, 24, cross_width,
    u8g2.setDrawColor(1); // White
    u8g2.drawXBMP(frame_size * 2, 24, cross_width, cr
    u8g2.setDrawColor(2); // XOR
    u8g2.drawXBMP(frame_size * 3.5, 24, cross_width,
    }

//Define the initial variable for the drawing state
uint8_t draw_state = 0;

/*
 * Draw functions: call other functions in an order
 */
void draw(void) {
    u8g2_prepare();
    switch(draw_state >> 3) {
        case 0: u8g2_box_title(draw_state&7); break;
        case 1: u8g2_box_frame(draw_state&7); break;
        case 2: u8g2_disc_circle(draw_state&7); break;
        case 3: u8g2_r_frame(draw_state&7); break;
        case 4: u8g2_string(draw_state&7); break;
        case 5: u8g2_line(draw_state&7); break;
        case 6: u8g2_triangle(draw_state&7); break;
        case 7: u8g2_ascii_1(); break;
        case 8: u8g2_ascii_2(); break;
        case 9: u8g2_extra_page(draw_state&7); break;
        case 10: u8g2_xor(draw_state&7); break;
        case 11: u8g2_bitmap_modes(0); break;
        case 12: u8g2_bitmap_modes(1); break;
        case 13: u8g2_bitmap_overlay(draw_state&7); bre
    }
}

void setup(void) {
    u8g2.begin(); //Initialize the function
}

void loop(void) {
    // picture loop
    u8g2.firstPage();
    do {
        draw();
    } while( u8g2.nextPage() );

    // increase the state
    draw_state++;
    if ( draw_state >= 14*8 )
        draw_state = 0;

    // delay between each page
    delay(150);

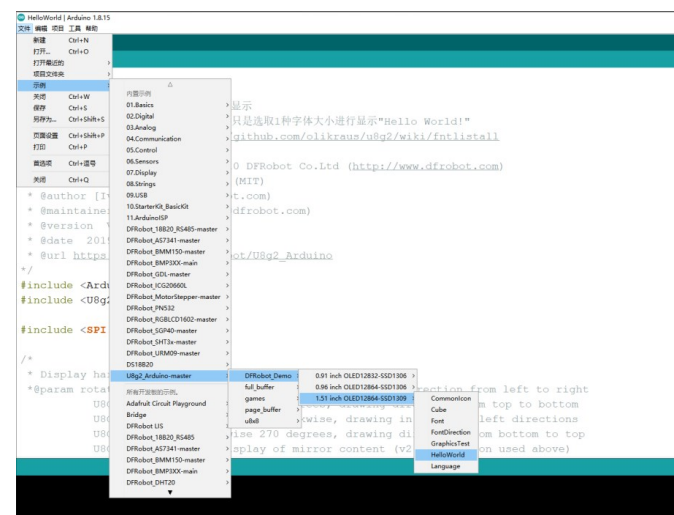
}

```

Result(There are many demo screens in this example. The following are some pictures in the demo screen):



Sample Code 6-HelloWorld.ino



```

/*!
 * @file HelloWorld.ino
 * @brief Display Hello World!
 * @n U8G2 supports fonts of various sizes. This de
 * @n U8G2 Font GitHub link:https://github.com/olik
 *
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (h
 * @licence The MIT License (MIT)
 * @author [Ivey](Ivey.lu@dfrobot.com)
 * @maintainer [Fary](feng.yang@dfrobot.com)
 * @version V1.0

```

```

* @date 2019-10-15
* @url https://github.com/DFRobot/U8g2_Arduino
*/
#include <Arduino.h>
#include <U8g2lib.h>

#include <SPI.h>

/*
 * Display hardware IIC interface constructor
 * @param rotation:U8G2_R0 Not rotate, horizontally,
                U8G2_R1 Rotate clockwise 90 degrees, dra
                U8G2_R2 Rotate 180 degrees clockwise, dr
                U8G2_R3 Rotate clockwise 270 degrees, dr
                U8G2_MIRROR Normal display of mirror con
                Note: U8G2_MIRROR need to be used with s
 * @param reset:U8x8_PIN_NONE Indicates that the pin
 * Display hardware SPI interface constructor
 * @param Just connect the CS pin (pins are optiona
 * @param Just connect the DC pin (pins are optiona
 *
 */
#if defined ARDUINO_SAM_ZERO
#define OLED_DC 7
#define OLED_CS 5
#define OLED_RST 6
/*ESP32 */
#elif defined(ESP32)
#define OLED_DC D2
#define OLED_CS D6
#define OLED_RST D3
/*ESP8266*/
#elif defined(ESP8266)
#define OLED_DC D4
#define OLED_CS D6
#define OLED_RST D5
/*AVR series board*/
#else
#define OLED_DC 2
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rot

void setup(void) {
    u8g2.begin();
    u8g2.setFontPosTop();
}

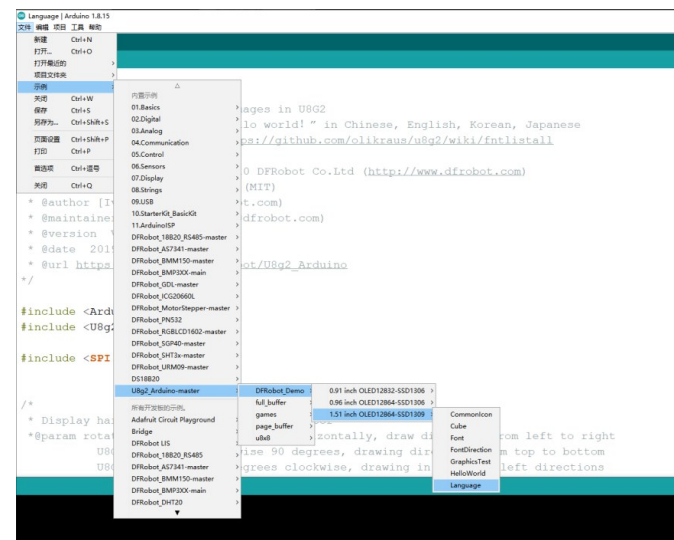
void loop(void) {
    u8g2.firstPage();
    do
    {
        u8g2.clearBuffer();           // clear the inter
        u8g2.setFont(u8g2_font_t0_17b_tr ); // choose
        u8g2.drawStr(0,5,"Hello World!"); // write som
        u8g2.sendBuffer();           // transfer interna
    } while( u8g2.nextPage() );
    delay(2000);
}

```

Result:



Sample Code 7-Language.ino



```
/*!  
 * @file Language.ino  
 * @brief Display multiple languages in U8G2  
 * @n A demo for displaying "hello world!" in Chinese, English, Korean, Japanese  
 * @n U8G2 Font GitHub Link:https://github.com/olikraus/u8g2/wiki/fntlistall  
 *  
 * @copyright Copyright (c) 2010 DFRobot Co.Ltd (http://www.dfrobot.com)  
 * @licence The MIT License (MIT)  
 * @author [Ivey](Ivey.lu@dfrobot.com)  
 * @maintainer [Fary](feng.yang@dfrobot.com)  
 * @version V1.0  
 * @date 2019-10-15  
 * @url https://github.com/DFRobot/U8g2_Arduino  
 */  
  
#include <Arduino.h>  
#include <U8g2lib.h>  
  
#include <SPI.h>  
  
/*  
 * Display hardware IIC interface constructor  
 * @param rotation:U8G2_R0 Not rotate, horizontally,  
 * U8G2_R1 Rotate clockwise 90 degrees, drawing direction from left to right  
 * U8G2_R2 Rotate 180 degrees clockwise, drawing direction from top to bottom  
 * U8G2_R3 Rotate clockwise 270 degrees, drawing direction from left to right  
 * U8G2_MIRROR Normal display of mirror construction  
 * Note: U8G2_MIRROR need to be used with software IIC interface  
 * @param reset:U8x8_PIN_NONE Indicates that the pin
```

```

* Display hardware SPI interface constructor
*@param Just connect the CS pin (pins are optional)
*@param Just connect the DC pin (pins are optional)
*
*/
#if defined ARDUINO_SAM_ZERO
#define OLED_DC 7
#define OLED_CS 5
#define OLED_RST 6
/*ESP32 */
#elif defined(ESP32)
#define OLED_DC D2
#define OLED_CS D6
#define OLED_RST D3
/*ESP8266*/
#elif defined(ESP8266)
#define OLED_DC D4
#define OLED_CS D6
#define OLED_RST D5
/*AVR series board*/
#else
#define OLED_DC 2
#define OLED_CS 3
#define OLED_RST 4
#endif
U8G2_SSD1309_128X64_NONAME2_1_4W_HW_SPI u8g2(/* rotate 0 degrees */

void setup(void) {
    u8g2.begin(); //init
    u8g2.enableUTF8Print(); // Enable UTF8 support
}

void loop(void)
{
    /*@brief Set font direction of all strings setFontDirection
    *@param dir=0, rotate 0 degree
           dir=1, rotate 90 degrees
           dir=2, rotate 180 degrees
           dir=3, rotate 270 degrees
    *@param When completed font setting, re-set the font direction
    */
    u8g2.setFontDirection(0);
    /*
    * firstPage Change the current page number position
    * Revise content in firstPage and nextPage, re-render
    * This method consumes less ram space than sendBuffer
    */
    u8g2.firstPage();
    do {
        /*
        *The font takes up a lot of memory, so please use a controller with larger memory
        *Display by drawXBM or use controller with larger memory
        *Chinese Font:require a controller with larger memory
        *Japanese Font:require a controller with larger memory
        *Korean Font:Arduino INO files of the current directory
        */
        u8g2.setFont(u8g2_font_HelvetiPixelOutline_tr);
        u8g2.setCursor(/* x=0, y=15); //Define cursor position
        u8g2.print("Hello World!");
        u8g2.setFont(u8g2_font_unifont_t_chinese2);
        u8g2.setCursor(0, 35);
        u8g2.print("你好世界"); // Chinese "Hello World"
        u8g2.setFont(u8g2_font_b10_t_japanese1);
        u8g2.setCursor(0, 50);
        u8g2.print("こんにちは世界"); // Japanese "Hello World"
    } while ( u8g2.nextPage() );
    delay(1000);
}

```

Result:



FAQ

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#).

More Documents

- [DFR0934-Schematics.PDF](#)
- [\[DFR0934\]2D.rar](#)
- [\[DFR0934\]3D.rar](#)
- [DFR0934-Datasheet.pdf](#)



Get [Fermion_1.51Inch](#)

[128x64_OLED_Transparent_Display_with_Converter_Breakout](#) from DFRobot Store or [DFRobot Distributor](#).

[Turn to the Top](#)