



Transparent Graphical OLED Breakout Hookup Guide

Introduction

The future is here! You asked and we delivered - our Qwiic Transparent Graphical OLED Breakout allows you to display custom images on a transparent screen using either I²C or SPI connections.

With Qwiic connectors it's quick (ha ha) and easy to get started with your own images. However, we still have broken out 0.1"-spaced pins in case you prefer to use a breadboard. Brilliantly lit in the dark and still visible by daylight, this OLED sports a display area of 128x64 pixels, 128x56 of which are completely transparent. Control of the OLED is based on our new HyperDisplay library.



SparkFun Transparent Graphical OLED Breakout (Qwiic)






○ LCD-15173

Product Showcase: SparkFun Transparent Graphical OLED Brea...

The Arduino sketch required to drive this display requires quite a bit of dynamic memory, meaning that it is not going to fit on a smaller controller like an ATmega328. Any controller with larger RAM should have no problem. In this tutorial, we're using the ESP32 Thing Plus.

Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything, depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary.

Transparent Graphical OLED Breakout Hookup Guide Wish List SparkFun Wish List	
	<p>Qwiic Cable - 200mm PRT-14428</p> <p>This 28AWG wire is terminated with two male Qwiic JST connectors at each end. Each wire is capabl...</p>
	<p>SparkFun Thing Plus - ESP32 WROOM WRL-14689</p>
	<p>SparkFun Transparent Graphical OLED Breakout (Qwiic) LCD-15173</p>
	<p>USB Wall Charger - 5V, 1A (Black) TOL-11456</p> <p>USB is being implemented as a power connection standard more and more these days, but you don't...</p>
	<p>USB Micro-B Cable - 6" CAB-13244</p> <p>This is a USB 2.0 type B to Micro-B 5-pin black cable. You know, the mini-B connector that usually co...</p>

Tools

If you want to use the broken out GPIO pins along the bottom of the board, you'll need to solder some headers to them so you'll need a soldering iron, solder, and general soldering accessories.



Solder Lead Free - 100-gram Spool
● TOL-09325



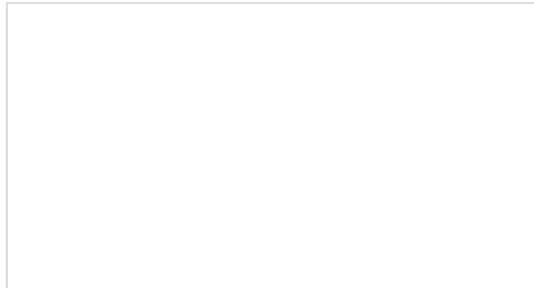
Weller WLC100 Soldering Station
● TOL-14228

Suggested Reading

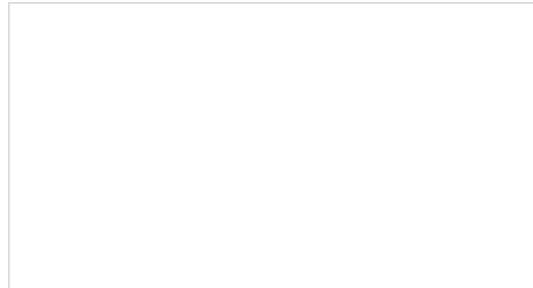
If you aren't familiar with the Qwiic system, we recommend reading here for an overview.



We would also recommend taking a look at the following tutorials if you aren't familiar with them.

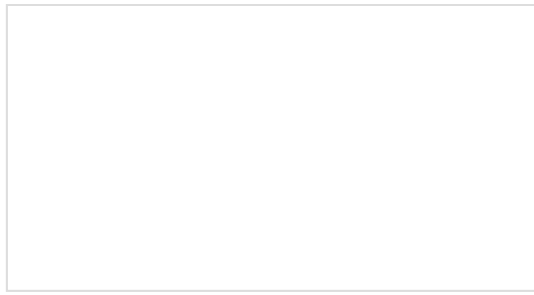


Serial Communication
Asynchronous serial communication concepts: packets, signal levels, baud rates, UARTs and more!



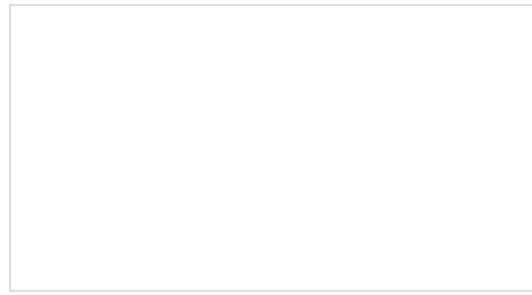
I2C
An introduction to I2C, one of the main embedded communications protocols in use today.

New!



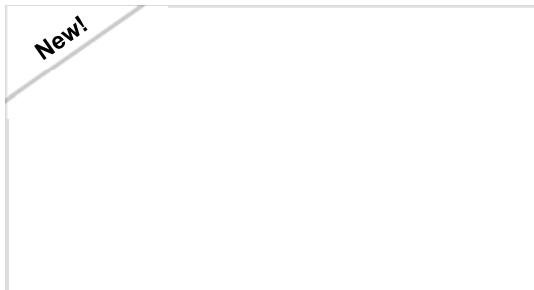
Serial Terminal Basics

This tutorial will show you how to communicate with your serial devices using a variety of terminal emulator applications.



ESP32 Thing Plus Hookup Guide

Hookup guide for the ESP32 Thing Plus using the ESP32 WROOM's WiFi/Bluetooth system-on-chip in Arduino.



Everything You Should Know About HyperDisplay

This is a tutorial to go in-depth about the SparkFun HyperDisplay Arduino Library.

Hardware Overview

Let's check out some of the characteristics of the Qwiic we're dealing with, so we know what to expect out of the board.

Characteristic	Range
Operating Voltage	1.65V-3.3V
Supply Current	400 mA
I ² C Addresses	0x30 (Default), 0x31

Graphical Display

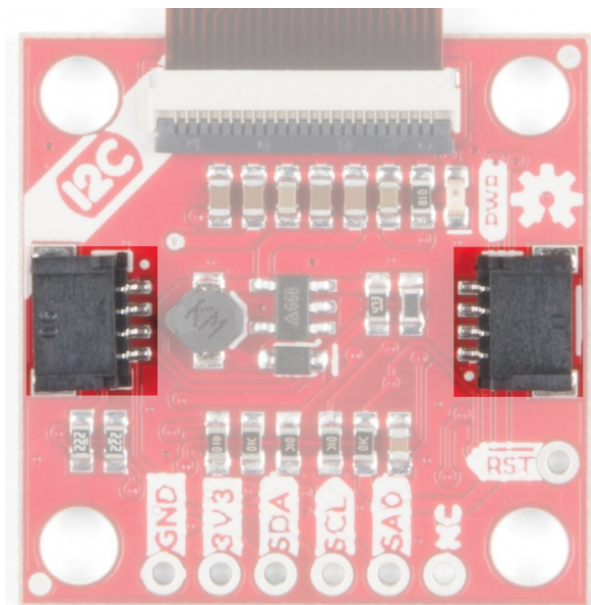
The graphical display is where all the fun stuff happens. The glass itself measures 42mm x 27.16mm, with a pixel display that is 35.5 x 18mm. It houses 128x64 pixels, 128x56 of which are transparent.



Graphical Display

Qwiic Connectors

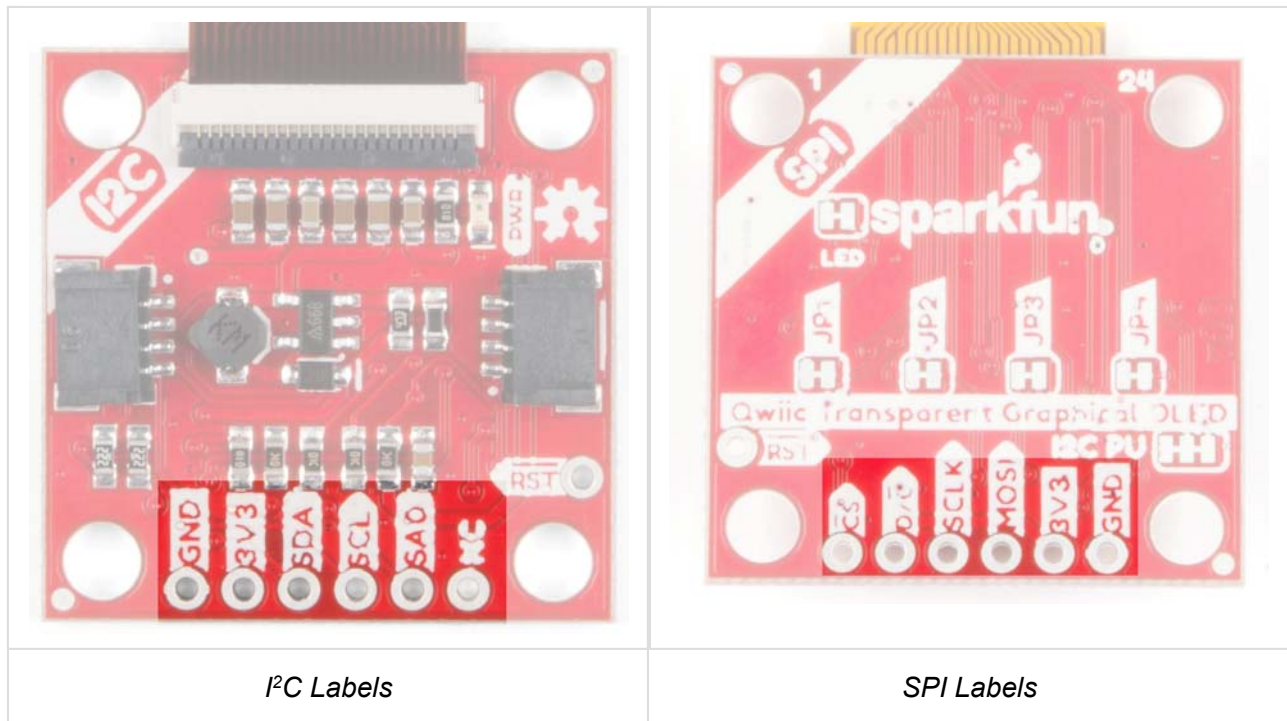
There are two Qwiic connectors on the board such that you can daisy-chain the boards should you choose to do so. If you're unfamiliar with our Qwiic system, head on over to our Qwiic page to see the advantages!



Qwiic Connectors

GPIO Pins

When you look at the GPIO pins, you'll notice that the labels are different from one side to the other. One side is labeled for I²C, the other side is labeled for SPI.

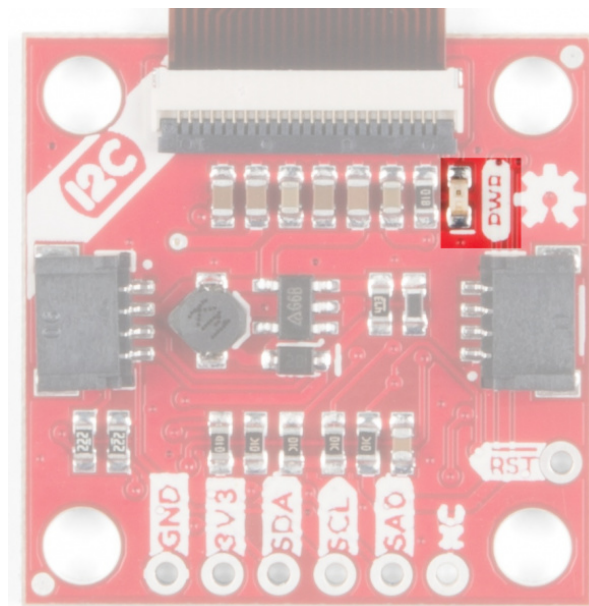


I2C Labels

SPI Labels

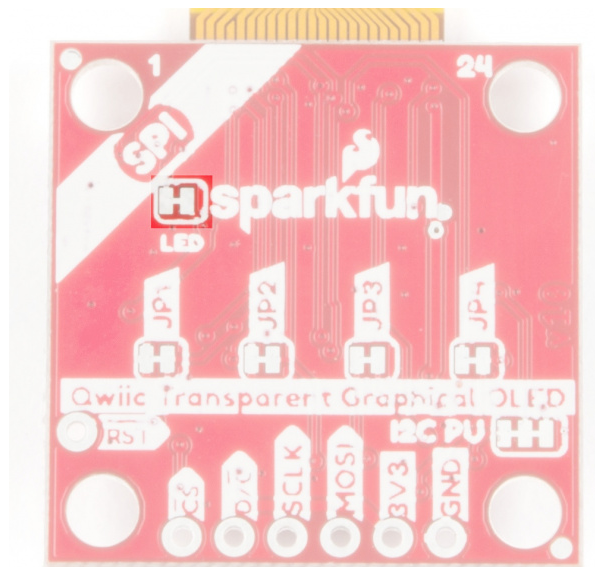
Power LED

This bad boy will light up when the board is powered up correctly.



Power LED

You can disable the power LED by cutting the LED jumpers on the back of the board.

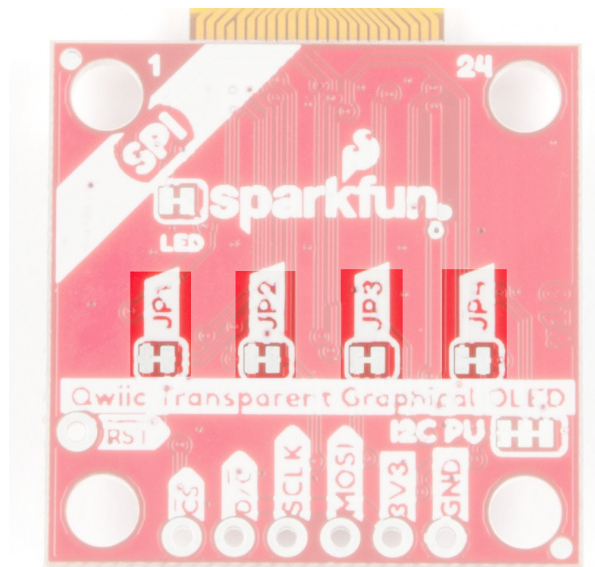


Power LED Jumpers

JPX Jumpers

The JPX jumpers are used to either change the I²C address or configure the board to use SPI communications. The other two jumpers allow you to disconnect the power LED and to disconnect the I²C pull-up resistors when chaining several Qwiic devices.

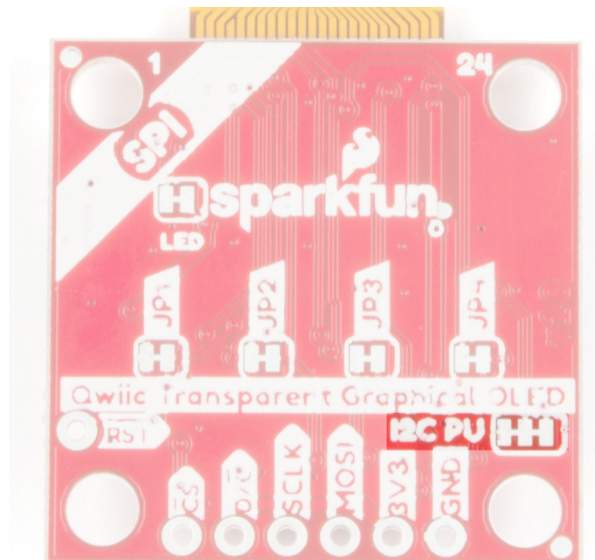
Jumper	Function
JP1	Holds the Chip Select line low when closed. Close for I ² C, open for SPI
JP2	Selects the address in I ² C mode. Closed for 0x30 by default and open for 0x31. Open for SPI mode to release the D/C pin
JP3	Used to select I ² C or SPI mode. Close for I ² C, open for SPI
JP4	This jumper should be closed for I ² C and open for SPI. This connection allows SDA to be bi-directional



JPX Jumpers

I²C Pull-Up Jumper

I²C devices contain open drains so we include resistors on our boards to allow these devices to pull pins high. This becomes a problem if you have a large number of I²C devices chained together. If you plan to daisy chain more than a few Qwiic boards together, you'll need to cut this I²C pull-up jumper.



I²C PU Jumpers

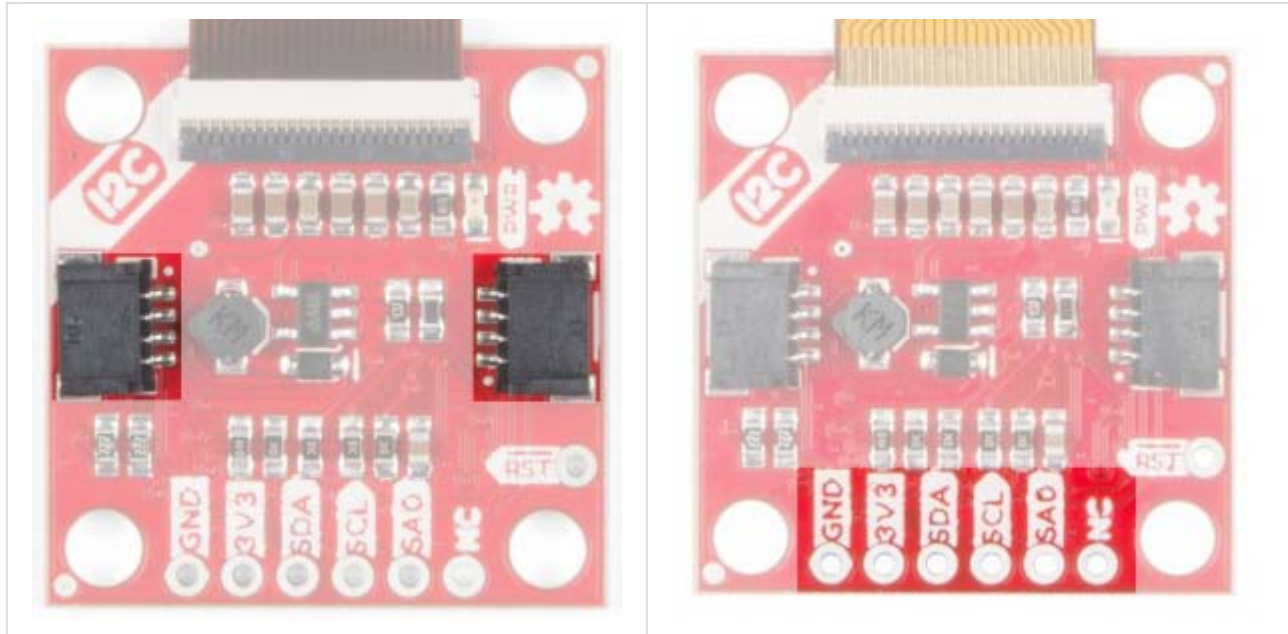
Hardware Hookup

Now that you know what's available on your breakout board we can check out the options for connecting it to the brains of your project. There are two options to use - either I²C or SPI - and they each have their own advantages and drawbacks. Read on to choose the best option for your setup.

✂ **Reminder!** This breakout can only handle up to **3.3V** on the pins, so make sure to do some **level shifting** if you're using a 5V microcontroller.

I²C (Qwiic)

The easiest way to start using the Transparent Graphical OLED is to use a Qwiic Cable along with a Qwiic compatible microcontroller (such as the ESP32 Thing Plus). You can also use the Qwiic Breadboard Cable to attach any I²C capable microcontroller, or take the scenic route and solder in all the I²C wires to the plated-through connections on the board.

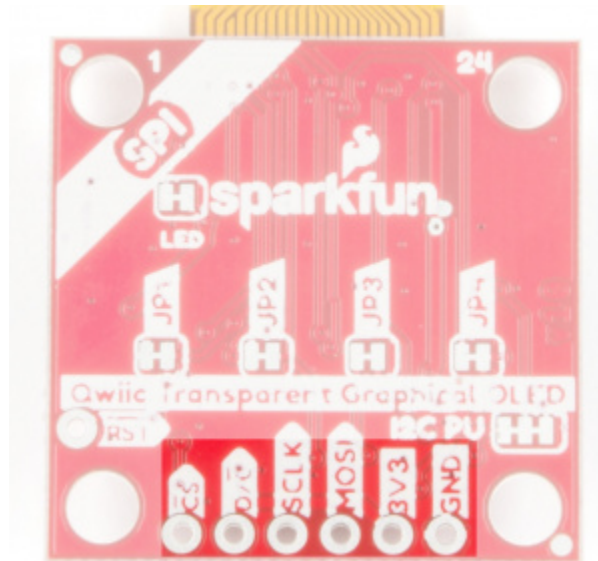


So why use I²C? It's easy to connect with the Qwiic system, and you can put up to two of the Transparent Graphical Breakouts on the same bus without using any more microcontroller pins. That simplicity comes at a cost to performance though. The maximum clock speed of the I²C bus is 400 kHz, and there is additional overhead in data transmission to indicate which bytes are data and which are commands. This means that the I²C connection is best for showing static images.

Breakout Pin	Microcontroller Pin Requirements
GND	Ground pin. Connect these so the two devices agree on voltages
3V3	3.3V supply pin, capable of up to 400 mA output
SDA	SDA - the bi-directional data line of your chosen I2C port
SCL	SCL - the clock line of your chosen I2C port
SA0	<i>Optional</i> : change the I2C address of the breakout. Make sure to cut JP2
RST	<i>Optional</i> : reset the breakout to a known state by pulsing this low

SPI

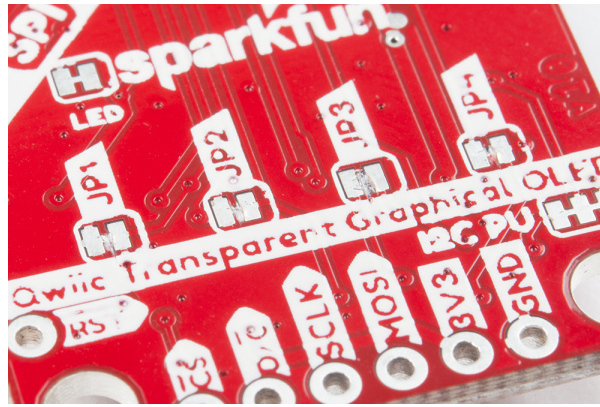
SPI solves the I²C speed problems. With SPI there is a control signal that indicates data or command and the maximum clock speed is 10 MHz -- giving SPI 50x more speed! However, it doesn't have the same conveniences of the polarized Qwiic connector and low pin usage. You'll need to solder to the pins.



You can use SPI to connect as many breakouts as you want. For N displays you will need to use at least N + 3 data pins. That's because the MOSI, SCLK, and D/C pins can be shared between displays but each breakout needs its own dedicated Chip Select (CS) pin.

Breakout Pin	Microcontroller Pin Requirements
CS	A GPIO pin, set low when talking to the breakout
D/C	A GPIO pin, indicates if bytes are data or commands
SCLK	The clock output of your chosen SPI port
MOSI	The data output of your chosen SPI port
3V3	3.3V supply pin, capable of up to 400 mA output
GND	Ground pin. Connect these so the two devices agree on voltages

⚠ Make sure to cut jumpers JP1, JP2, JP3, and JP4 when using SPI mode!

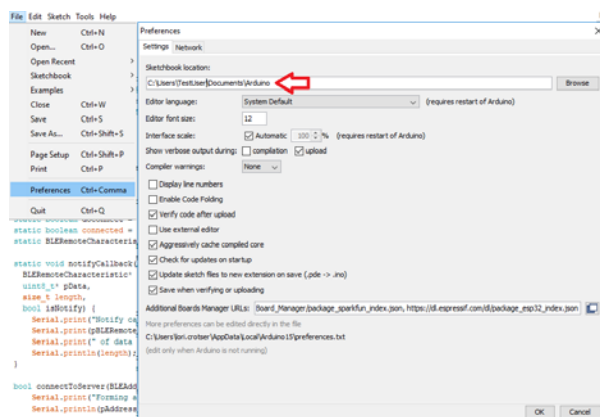


Software Setup and Programming

Note: This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you have not previously installed an Arduino library, please check out our installation guide.

Drivers and Board Add-Ons

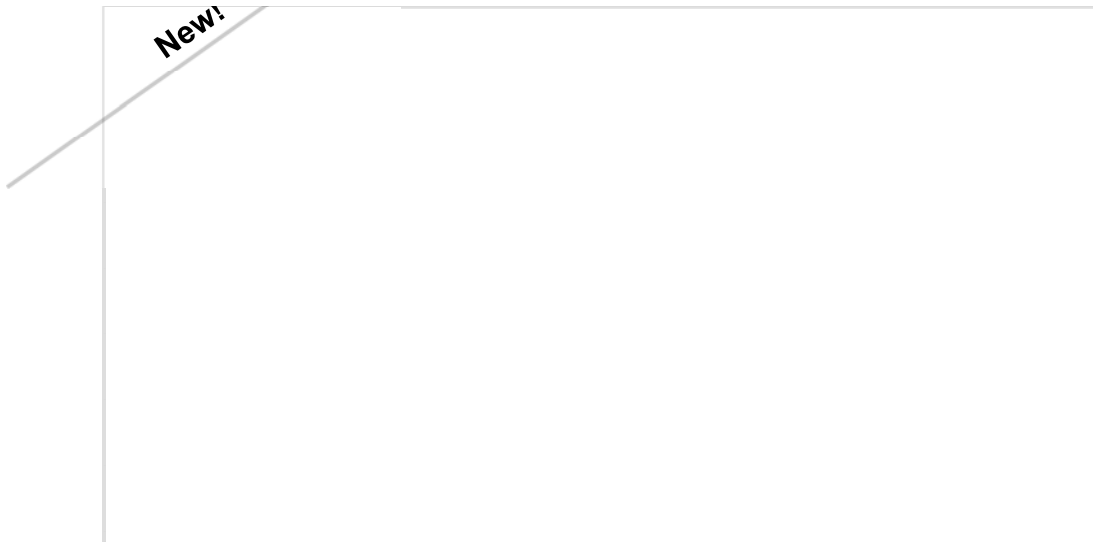
Depending on your development board, you may need to install drivers and a board add-on. For the scope of this tutorial, we will be using the ESP32 Thing Plus. If you have not already, make sure that you install the drivers and ESP32 Arduino core files as explained in our ESP32 Thing Plus Hookup Guide.



Software Setup for the ESP32

Arduino Libraries

The software for the Transparent Graphical OLED Breakout is built on the HyperDisplay libraries developed by SparkFun. If you are unfamiliar with how this abstracted library works, head over to our Everything You Should Know About HyperDisplay tutorial.



Everything You Should Know About HyperDisplay

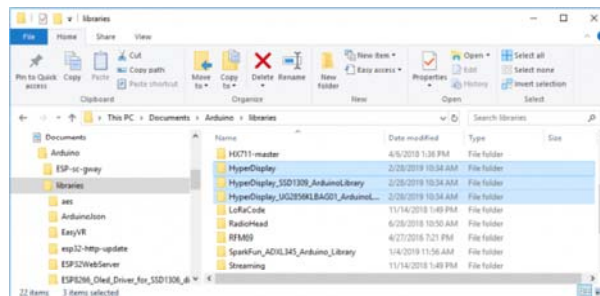
FEBRUARY 20, 2019

This is a tutorial to go in-depth about the SparkFun HyperDisplay Arduino Library.

HyperDisplay is an abstracted library that requires multiple layers in order to function correctly. For the Transparent Graphical OLED to work, you'll need our base HyperDisplay Library, the HyperDisplay SSD1309 Arduino Library, and the somewhat painfully but informatively named HyperDisplay UG2856KLBAG01 Arduino Library. You can obtain these libraries through the Arduino Library Manager by searching for "**HyperDisplay Library**", "**HyperDisplay SSD1309**", "**HyperDisplay Transparent Graphical OLED**", respectively. You can download the required libraries from their individual GitHub pages or download the following ***.zip** file that contains all three libraries to manually install:

SPARKFUN TRANSPARENT GRAPHICAL OLED BREAKOUT LIBRARIES (ZIP)

If you are manually downloading the libraries, unzip them and copy all three libraries to your **Arduino > Libraries** folder, wherever that may be located on your machine.



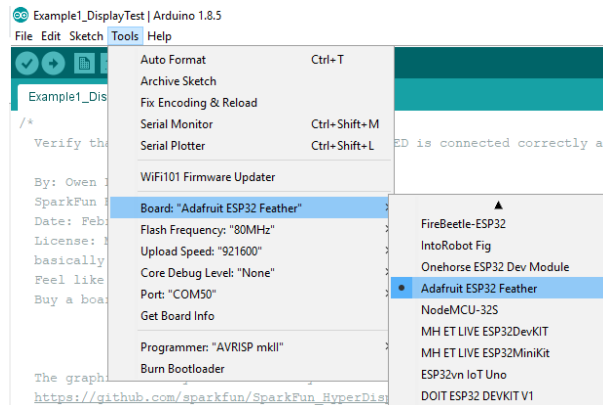
Having a hard time seeing? Click the image for a closer look.

Examples

Included in the HyperDisplay_UG2856KLBAG01_ArduinoLibrary are three examples to help you get started using the Transparent Graphical OLED Breakout. If you haven't already downloaded the libraries from the afore-mentioned locations, click on the button below to download the library bundle. Unzip and copy it into your **Arduino > Libraries** folder.

SPARKFUN TRANSPARENT GRAPHICAL OLED BREAKOUT LIBRARIES (ZIP)

If you are following along with this tutorial using an ESP32 Thing Plus board (as we are), make sure you set your board definition to **Adafruit ESP32 Feather**.



Example 1: DisplayTest

The purpose of this example is to act like the "Hello World" message for your display. Uploading this sketch will help confirm that you have followed the hardware hookup correctly. The sketch runs through a couple of different display test, beginning with a custom SparkFun logo display.

To show the logo, Owen whipped up a quick python script that looks at a jpg image and writes out, pixel by pixel, a function to make the display. We've included his script and our image here--have a look! With just a little modification it could be useful in your own application.

PYTHON SCRIPT TO CONVERT IMAGE TO PIXEL (ZIP)

To get started, either select **File > Examples > SparkFun HyperDisplay Transparent Graphical OLED Library > Example1_DisplayTest** or copy and paste the code below into a new Arduino window.

```

/*
  Verify that your Qwiic Transparent Graphical OLED is connected correctly and working.

  By: Owen Lyke
  SparkFun Electronics
  Date: February 26, 2019
  License: MIT. See license file for more information but you can
  basically do whatever you want with this code.
  Feel like supporting open source hardware?
  Buy a board from SparkFun! https://www.sparkfun.com/products/15173

  The graphics library is like a 3-layer cake. Here they are from top-down
  https://github.com/sparkfun/SparkFun\_HyperDisplay
  https://github.com/sparkfun/HyperDisplay\_SSD1309\_ArduinoLibrary
  https://github.com/sparkfun/HyperDisplay\_UG2856KLBAG01\_ArduinoLibrary

  Hardware Compatibility
  - The IO pins on this board are designed for use with 3.3V so if you are using a 5
  V microcontroller
    please use a level shifter. Note: Qwiic connectors on SparkFun dev boards are al
  ready at 3.3V
  - This display relies on a copy of graphics data in your microcontroller, a total
  of 1024 bytes.
    That is half the RAM available on an Uno so it is easy to run into sinister low-
  memory related
    bugs. We reccomend using a micro with more memory like a SAMD21, Esp32, Teensy,
  etc.

  Hardware Connections:
  Option 1 (I2C):
    Connect using a Qwiic jumper if you have a Qwiic compatible board and you plan to
  use I2C

  Option 2 (SPI):
    Connect SCLK and MOSI to the SPI port of your choice (13 and 11 for SPI on Uno-lik
  e boards)
    Also connect D/C and CS to two unused GPIO pins of your choice (and set the prope
  r pin definitions below)
    Don't forget power - connect 3.3V and GND
*/

#include "HyperDisplay_UG2856KLBAG01.h" // Your library can be installed here: htt
p://librarymanager/All#SparkFun_Transparent_Graphical_OLED
// The rest of the Layer Cake:      htt
p://librarymanager/All#SparkFun_HyperDisplay_SSD1309
//                                  htt

```

```

p://librarymanager/All#SparkFun_HyperDisplay
////////////////////////////////////
//      User Setup      //
////////////////////////////////////
#define SERIAL_PORT Serial
#define WIRE_PORT Wire      // Used if USE_SPI == 0
#define SPI_PORT SPI        // Used if USE_SPI == 1

#define RES_PIN 2          // Optional
#define CS_PIN 4           // Used only if USE_SPI == 1
#define DC_PIN 5           // Used only if USE_SPI == 1

#define USE_SPI 1          // Choose your interface. 0 = I2C, 1 = SPI

// END USER SETUP

// Object Declaration. A class exists for each interface option
#if USE_SPI
  UG2856KLBAG01_SPI myTOLED; // Declare a SPI-based Transparent OLED object called my
  TOLED
#else
  UG2856KLBAG01_I2C myTOLED; // Declare a I2C-based Transparent OLED object called my
  TOLED
#endif /* USE_SPI */

void setup() {
  Serial.begin(9600);

  Serial.println(F("Example1_DisplayTest: Transparent Graphical OLED"));

#if USE_SPI
  SPI_PORT.begin();
  myTOLED.begin(CS_PIN, DC_PIN, SPI_PORT); // Begin for SPI requires
that you provide the CS and DC pin numbers
#else
  WIRE_PORT.begin();
  myTOLED.begin(WIRE_PORT, false, SSD1309_ARD_UNUSED_PIN); // Begin for I2C has defau
lt values for every argument
  Wire.setClock(400000);
#endif /* USE_SPI */

// Don't show the logo on boards with small memory
#if !defined(__AVR_ATmega328P__) && !defined(__AVR_ATmega168__)
  showLogo( ); // The showLogo function is a hacky way to get a large bitmap int
o program space without using <avr/pgspace.h>
#endif
}

```

```
void loop() {

    lineTest();
    delay(500);

    rectTest();
    delay(500);

    circleTest();
    delay(500);
}

void lineTest( void )
{
    myTOLED.clearDisplay();

    for(hd_hw_extent_t indi = 0; indi < myTOLED.xExt; indi+=5)
    {
        myTOLED.lineSet(0,0,indi,myTOLED.yExt-1,1);
        delay(10);
    }

    for(hd_hw_extent_t indi = 0; indi < myTOLED.yExt; indi+=5)
    {
        myTOLED.lineSet(0,myTOLED.yExt-1,myTOLED.xExt-1,myTOLED.yExt-indi-1,1);
        delay(10);
    }

    for(hd_hw_extent_t indi = 0; indi < myTOLED.xExt; indi+=5)
    {
        myTOLED.lineSet(myTOLED.xExt-1,myTOLED.yExt-1,myTOLED.xExt-indi-1,0,1);
        delay(10);
    }

    for(hd_hw_extent_t indi = 0; indi < myTOLED.yExt; indi+=5)
    {
        myTOLED.lineSet(myTOLED.xExt-1,0,0,indi,1);
    }
}

void rectTest( void )
{
    myTOLED.clearDisplay();

    for(uint8_t indi = 0; indi < myTOLED.yExt/2; indi+=1)
    {
        myTOLED.rectangleSet(myTOLED.xExt/2-1-indi, myTOLED.yExt/2-1-indi, myTOLED.xExt/2+
1+indi, myTOLED.yExt/2+1+indi, false);
    }
}
```



```

    delay(50);
}

for(uint8_t indi = 0; indi < myTOLED.yExt/2; indi+=1)
{
    myTOLED.rectangleClear(myTOLED.xExt/2-1-indi, myTOLED.yExt/2-1-indi, myTOLED.xExt/
2+1+indi, myTOLED.yExt/2+1+indi, false);
    delay(50);
}

for(uint8_t indi = 0; indi < myTOLED.yExt/2; indi+=1)
{
    myTOLED.rectangleSet(myTOLED.xExt/2-1-indi, myTOLED.yExt/2-1-indi, myTOLED.xExt/2+
1+indi, myTOLED.yExt/2+1+indi, false);
    delay(50);
}
}

void circleTest( void )
{
    myTOLED.clearDisplay();

    for(uint8_t indi = 0; indi < (myTOLED.xExt/2 - 1); indi++)
    {
        myTOLED.circleSet((myTOLED.xExt/2 - 1),(myTOLED.yExt/2 - 1), indi, false);
        delay(10);
    }
    myTOLED.circleSet((myTOLED.xExt/2 - 1),(myTOLED.yExt/2 - 1), myTOLED.xExt/2, true);
}

#if !defined(__AVR_ATmega328P__) && !defined(__AVR_ATmega168__)
void showLogo( void ){
    myTOLED.setContrastControl(0);

    myTOLED.pixelSet(84, 3); myTOLED.pixelSet(85, 3); myTOLED.pixelSet(86, 3); myTOLED.p
ixelSet(87, 3); myTOLED.pixelSet(88, 3);
    myTOLED.pixelSet(83, 4); myTOLED.pixelSet(84, 4); myTOLED.pixelSet(85, 4); myTOLED.p
ixelSet(86, 4); myTOLED.pixelSet(87, 4);
    myTOLED.pixelSet(82, 5); myTOLED.pixelSet(83, 5); myTOLED.pixelSet(84, 5); myTOLED.p
ixelSet(85, 5); myTOLED.pixelSet(86, 5);
    myTOLED.pixelSet(82, 6); myTOLED.pixelSet(83, 6); myTOLED.pixelSet(84, 6); myTOLED.p
ixelSet(85, 6); myTOLED.pixelSet(86, 6);
    myTOLED.pixelSet(82, 7); myTOLED.pixelSet(83, 7); myTOLED.pixelSet(84, 7); myTOLED.p
ixelSet(85, 7); myTOLED.pixelSet(86, 7);
    myTOLED.pixelSet(82, 8); myTOLED.pixelSet(83, 8); myTOLED.pixelSet(84, 8); myTOLED.p
ixelSet(85, 8); myTOLED.pixelSet(86, 8); myTOLED.pixelSet(87, 8); myTOLED.pixelSet(9
2, 8);
    myTOLED.pixelSet(83, 9); myTOLED.pixelSet(84, 9); myTOLED.pixelSet(85, 9); myTOLED.p
ixelSet(86, 9); myTOLED.pixelSet(87, 9); myTOLED.pixelSet(88, 9); myTOLED.pixelSet(8

```

```
9, 9); myTOLED.pixelSet(91, 9); myTOLED.pixelSet(92, 9); myTOLED.pixelSet(93, 9);
  myTOLED.pixelSet(84, 10); myTOLED.pixelSet(85, 10); myTOLED.pixelSet(86, 10); myTOLE
D.pixelSet(87, 10); myTOLED.pixelSet(88, 10); myTOLED.pixelSet(89, 10); myTOLED.pixels
et(90, 10); myTOLED.pixelSet(91, 10); myTOLED.pixelSet(92, 10); myTOLED.pixelSet(93, 1
0);
  myTOLED.pixelSet(85, 11); myTOLED.pixelSet(86, 11); myTOLED.pixelSet(87, 11); myTOLE
D.pixelSet(88, 11); myTOLED.pixelSet(89, 11); myTOLED.pixelSet(90, 11); myTOLED.pixels
et(91, 11); myTOLED.pixelSet(92, 11); myTOLED.pixelSet(93, 11);
  myTOLED.pixelSet(78, 12); myTOLED.pixelSet(79, 12); myTOLED.pixelSet(85, 12); myTOLE
D.pixelSet(86, 12); myTOLED.pixelSet(87, 12); myTOLED.pixelSet(88, 12); myTOLED.pixels
et(89, 12); myTOLED.pixelSet(90, 12); myTOLED.pixelSet(91, 12); myTOLED.pixelSet(92, 1
2); myTOLED.pixelSet(93, 12); myTOLED.pixelSet(94, 12);
  myTOLED.pixelSet(77, 13); myTOLED.pixelSet(78, 13); myTOLED.pixelSet(79, 13); myTOLE
D.pixelSet(85, 13); myTOLED.pixelSet(86, 13); myTOLED.pixelSet(87, 13); myTOLED.pixels
et(88, 13); myTOLED.pixelSet(89, 13); myTOLED.pixelSet(90, 13); myTOLED.pixelSet(91, 1
3); myTOLED.pixelSet(92, 13); myTOLED.pixelSet(93, 13); myTOLED.pixelSet(94, 13);
  myTOLED.pixelSet(77, 14); myTOLED.pixelSet(78, 14); myTOLED.pixelSet(79, 14); myTOLE
D.pixelSet(85, 14); myTOLED.pixelSet(86, 14); myTOLED.pixelSet(87, 14); myTOLED.pixels
et(88, 14); myTOLED.pixelSet(89, 14); myTOLED.pixelSet(90, 14); myTOLED.pixelSet(91, 1
4); myTOLED.pixelSet(92, 14); myTOLED.pixelSet(93, 14); myTOLED.pixelSet(94, 14);
  myTOLED.pixelSet(76, 15); myTOLED.pixelSet(77, 15); myTOLED.pixelSet(78, 15); myTOLE
D.pixelSet(79, 15); myTOLED.pixelSet(80, 15); myTOLED.pixelSet(84, 15); myTOLED.pixels
et(85, 15); myTOLED.pixelSet(86, 15); myTOLED.pixelSet(87, 15); myTOLED.pixelSet(88, 1
5); myTOLED.pixelSet(89, 15); myTOLED.pixelSet(90, 15); myTOLED.pixelSet(91, 15); myTO
LED.pixelSet(92, 15); myTOLED.pixelSet(93, 15); myTOLED.pixelSet(94, 15);
  myTOLED.pixelSet(76, 16); myTOLED.pixelSet(77, 16); myTOLED.pixelSet(78, 16); myTOLE
D.pixelSet(79, 16); myTOLED.pixelSet(80, 16); myTOLED.pixelSet(81, 16); myTOLED.pixels
et(82, 16); myTOLED.pixelSet(83, 16); myTOLED.pixelSet(84, 16); myTOLED.pixelSet(85, 1
6); myTOLED.pixelSet(86, 16); myTOLED.pixelSet(87, 16); myTOLED.pixelSet(88, 16); myTO
LED.pixelSet(89, 16); myTOLED.pixelSet(90, 16); myTOLED.pixelSet(91, 16); myTOLED.pixe
lSet(92, 16); myTOLED.pixelSet(93, 16); myTOLED.pixelSet(94, 16);
  myTOLED.pixelSet(76, 17); myTOLED.pixelSet(77, 17); myTOLED.pixelSet(78, 17); myTOLE
D.pixelSet(79, 17); myTOLED.pixelSet(80, 17); myTOLED.pixelSet(81, 17); myTOLED.pixels
et(82, 17); myTOLED.pixelSet(83, 17); myTOLED.pixelSet(84, 17); myTOLED.pixelSet(85, 1
7); myTOLED.pixelSet(86, 17); myTOLED.pixelSet(87, 17); myTOLED.pixelSet(88, 17); myTO
LED.pixelSet(89, 17); myTOLED.pixelSet(90, 17); myTOLED.pixelSet(91, 17); myTOLED.pixe
lSet(92, 17); myTOLED.pixelSet(93, 17);
  myTOLED.pixelSet(76, 18); myTOLED.pixelSet(77, 18); myTOLED.pixelSet(78, 18); myTOLE
D.pixelSet(79, 18); myTOLED.pixelSet(80, 18); myTOLED.pixelSet(81, 18); myTOLED.pixels
et(82, 18); myTOLED.pixelSet(83, 18); myTOLED.pixelSet(84, 18); myTOLED.pixelSet(85, 1
8); myTOLED.pixelSet(86, 18); myTOLED.pixelSet(87, 18); myTOLED.pixelSet(88, 18); myTO
LED.pixelSet(89, 18); myTOLED.pixelSet(90, 18); myTOLED.pixelSet(91, 18); myTOLED.pixe
lSet(92, 18); myTOLED.pixelSet(93, 18);
  myTOLED.pixelSet(76, 19); myTOLED.pixelSet(77, 19); myTOLED.pixelSet(78, 19); myTOLE
D.pixelSet(79, 19); myTOLED.pixelSet(80, 19); myTOLED.pixelSet(81, 19); myTOLED.pixels
et(82, 19); myTOLED.pixelSet(83, 19); myTOLED.pixelSet(84, 19); myTOLED.pixelSet(85, 1
9); myTOLED.pixelSet(86, 19); myTOLED.pixelSet(87, 19); myTOLED.pixelSet(88, 19); myTO
LED.pixelSet(89, 19); myTOLED.pixelSet(90, 19); myTOLED.pixelSet(91, 19); myTOLED.pixe
lSet(92, 19);
```

```
myTOLED.pixelSet(76, 20); myTOLED.pixelSet(77, 20); myTOLED.pixelSet(78, 20); myTOLED.pixelSet(79, 20); myTOLED.pixelSet(80, 20); myTOLED.pixelSet(81, 20); myTOLED.pixelSet(82, 20); myTOLED.pixelSet(83, 20); myTOLED.pixelSet(84, 20); myTOLED.pixelSet(85, 20); myTOLED.pixelSet(86, 20); myTOLED.pixelSet(87, 20); myTOLED.pixelSet(88, 20); myTOLED.pixelSet(89, 20); myTOLED.pixelSet(90, 20); myTOLED.pixelSet(91, 20); myTOLED.pixelSet(92, 20);
myTOLED.pixelSet(76, 21); myTOLED.pixelSet(77, 21); myTOLED.pixelSet(78, 21); myTOLED.pixelSet(79, 21); myTOLED.pixelSet(80, 21); myTOLED.pixelSet(81, 21); myTOLED.pixelSet(82, 21); myTOLED.pixelSet(83, 21); myTOLED.pixelSet(84, 21); myTOLED.pixelSet(85, 21); myTOLED.pixelSet(86, 21); myTOLED.pixelSet(87, 21); myTOLED.pixelSet(88, 21); myTOLED.pixelSet(89, 21); myTOLED.pixelSet(90, 21); myTOLED.pixelSet(91, 21);
myTOLED.pixelSet(76, 22); myTOLED.pixelSet(77, 22); myTOLED.pixelSet(78, 22); myTOLED.pixelSet(79, 22); myTOLED.pixelSet(80, 22); myTOLED.pixelSet(81, 22); myTOLED.pixelSet(82, 22); myTOLED.pixelSet(83, 22); myTOLED.pixelSet(84, 22); myTOLED.pixelSet(85, 22); myTOLED.pixelSet(86, 22); myTOLED.pixelSet(87, 22); myTOLED.pixelSet(88, 22); myTOLED.pixelSet(89, 22);
myTOLED.pixelSet(76, 23); myTOLED.pixelSet(77, 23); myTOLED.pixelSet(78, 23); myTOLED.pixelSet(79, 23); myTOLED.pixelSet(80, 23); myTOLED.pixelSet(81, 23); myTOLED.pixelSet(82, 23); myTOLED.pixelSet(83, 23); myTOLED.pixelSet(84, 23); myTOLED.pixelSet(85, 23); myTOLED.pixelSet(86, 23); myTOLED.pixelSet(87, 23);
myTOLED.pixelSet(76, 24); myTOLED.pixelSet(77, 24); myTOLED.pixelSet(78, 24); myTOLED.pixelSet(79, 24); myTOLED.pixelSet(80, 24); myTOLED.pixelSet(81, 24);
myTOLED.pixelSet(76, 25); myTOLED.pixelSet(77, 25); myTOLED.pixelSet(78, 25); myTOLED.pixelSet(79, 25); myTOLED.pixelSet(80, 25);
myTOLED.pixelSet(76, 26); myTOLED.pixelSet(77, 26); myTOLED.pixelSet(78, 26); myTOLED.pixelSet(79, 26);
myTOLED.pixelSet(76, 27); myTOLED.pixelSet(77, 27); myTOLED.pixelSet(78, 27);
myTOLED.pixelSet(76, 28); myTOLED.pixelSet(77, 28);
myTOLED.pixelSet(76, 29); myTOLED.pixelSet(88, 29); myTOLED.pixelSet(89, 29); myTOLED.pixelSet(90, 29); myTOLED.pixelSet(91, 29); myTOLED.pixelSet(92, 29);
myTOLED.pixelSet(69, 30); myTOLED.pixelSet(70, 30); myTOLED.pixelSet(76, 30); myTOLED.pixelSet(87, 30); myTOLED.pixelSet(88, 30); myTOLED.pixelSet(89, 30); myTOLED.pixelSet(90, 30); myTOLED.pixelSet(91, 30); myTOLED.pixelSet(92, 30);
myTOLED.pixelSet(67, 31); myTOLED.pixelSet(68, 31); myTOLED.pixelSet(69, 31); myTOLED.pixelSet(70, 31); myTOLED.pixelSet(86, 31); myTOLED.pixelSet(87, 31); myTOLED.pixelSet(88, 31); myTOLED.pixelSet(89, 31); myTOLED.pixelSet(90, 31); myTOLED.pixelSet(91, 31); myTOLED.pixelSet(92, 31);
myTOLED.pixelSet(67, 32); myTOLED.pixelSet(68, 32); myTOLED.pixelSet(69, 32); myTOLED.pixelSet(70, 32); myTOLED.pixelSet(86, 32); myTOLED.pixelSet(87, 32); myTOLED.pixelSet(88, 32); myTOLED.pixelSet(89, 32);
myTOLED.pixelSet(67, 33); myTOLED.pixelSet(68, 33); myTOLED.pixelSet(69, 33); myTOLED.pixelSet(70, 33); myTOLED.pixelSet(86, 33); myTOLED.pixelSet(87, 33); myTOLED.pixelSet(88, 33);
myTOLED.pixelSet(67, 34); myTOLED.pixelSet(68, 34); myTOLED.pixelSet(69, 34); myTOLED.pixelSet(70, 34); myTOLED.pixelSet(86, 34); myTOLED.pixelSet(87, 34); myTOLED.pixelSet(88, 34);
myTOLED.pixelSet(6, 35); myTOLED.pixelSet(7, 35); myTOLED.pixelSet(8, 35); myTOLED.pixelSet(9, 35); myTOLED.pixelSet(10, 35); myTOLED.pixelSet(11, 35); myTOLED.pixelSet(12, 35); myTOLED.pixelSet(13, 35); myTOLED.pixelSet(22, 35); myTOLED.pixelSet(26, 35);
```

```
myTOLED.pixelSet(27, 35); myTOLED.pixelSet(28, 35); myTOLED.pixelSet(29, 35); myTOLED.
pixelSet(30, 35); myTOLED.pixelSet(31, 35); myTOLED.pixelSet(41, 35); myTOLED.pixelSet
(42, 35); myTOLED.pixelSet(43, 35); myTOLED.pixelSet(44, 35); myTOLED.pixelSet(45, 3
5); myTOLED.pixelSet(46, 35); myTOLED.pixelSet(47, 35); myTOLED.pixelSet(48, 35); myTO
LED.pixelSet(61, 35); myTOLED.pixelSet(62, 35); myTOLED.pixelSet(63, 35); myTOLED.pixe
lSet(64, 35); myTOLED.pixelSet(67, 35); myTOLED.pixelSet(68, 35); myTOLED.pixelSet(6
9, 35); myTOLED.pixelSet(70, 35); myTOLED.pixelSet(77, 35); myTOLED.pixelSet(78, 35);
myTOLED.pixelSet(79, 35); myTOLED.pixelSet(80, 35); myTOLED.pixelSet(84, 35); myTOLED.
pixelSet(85, 35); myTOLED.pixelSet(86, 35); myTOLED.pixelSet(87, 35); myTOLED.pixelSet
(88, 35); myTOLED.pixelSet(89, 35); myTOLED.pixelSet(90, 35); myTOLED.pixelSet(91, 3
5); myTOLED.pixelSet(92, 35); myTOLED.pixelSet(94, 35); myTOLED.pixelSet(95, 35); myTO
LED.pixelSet(96, 35); myTOLED.pixelSet(97, 35); myTOLED.pixelSet(105, 35); myTOLED.pixe
lSet(106, 35); myTOLED.pixelSet(107, 35); myTOLED.pixelSet(108, 35); myTOLED.pixelSet
(112, 35); myTOLED.pixelSet(113, 35); myTOLED.pixelSet(114, 35); myTOLED.pixelSet(11
8, 35); myTOLED.pixelSet(119, 35); myTOLED.pixelSet(120, 35); myTOLED.pixelSet(121, 3
5); myTOLED.pixelSet(122, 35); myTOLED.pixelSet(123, 35);
```

```
myTOLED.pixelSet(5, 36); myTOLED.pixelSet(6, 36); myTOLED.pixelSet(7, 36); myTOLED.p
ixelSet(8, 36); myTOLED.pixelSet(9, 36); myTOLED.pixelSet(10, 36); myTOLED.pixelSet(1
1, 36); myTOLED.pixelSet(12, 36); myTOLED.pixelSet(13, 36); myTOLED.pixelSet(14, 36);
myTOLED.pixelSet(20, 36); myTOLED.pixelSet(21, 36); myTOLED.pixelSet(22, 36); myTOLED.
pixelSet(24, 36); myTOLED.pixelSet(25, 36); myTOLED.pixelSet(26, 36); myTOLED.pixelSet
(27, 36); myTOLED.pixelSet(28, 36); myTOLED.pixelSet(29, 36); myTOLED.pixelSet(30, 3
6); myTOLED.pixelSet(31, 36); myTOLED.pixelSet(32, 36); myTOLED.pixelSet(39, 36); myTO
LED.pixelSet(40, 36); myTOLED.pixelSet(41, 36); myTOLED.pixelSet(42, 36); myTOLED.pixe
lSet(43, 36); myTOLED.pixelSet(44, 36); myTOLED.pixelSet(45, 36); myTOLED.pixelSet(4
6, 36); myTOLED.pixelSet(47, 36); myTOLED.pixelSet(48, 36); myTOLED.pixelSet(49, 36);
myTOLED.pixelSet(50, 36); myTOLED.pixelSet(55, 36); myTOLED.pixelSet(56, 36); myTOLED.
pixelSet(57, 36); myTOLED.pixelSet(58, 36); myTOLED.pixelSet(60, 36); myTOLED.pixelSet
(61, 36); myTOLED.pixelSet(62, 36); myTOLED.pixelSet(63, 36); myTOLED.pixelSet(64, 3
6); myTOLED.pixelSet(67, 36); myTOLED.pixelSet(68, 36); myTOLED.pixelSet(69, 36); myTO
LED.pixelSet(70, 36); myTOLED.pixelSet(76, 36); myTOLED.pixelSet(77, 36); myTOLED.pixe
lSet(78, 36); myTOLED.pixelSet(79, 36); myTOLED.pixelSet(83, 36); myTOLED.pixelSet(8
4, 36); myTOLED.pixelSet(85, 36); myTOLED.pixelSet(86, 36); myTOLED.pixelSet(87, 36);
myTOLED.pixelSet(88, 36); myTOLED.pixelSet(89, 36); myTOLED.pixelSet(90, 36); myTOLED.
pixelSet(91, 36); myTOLED.pixelSet(92, 36); myTOLED.pixelSet(94, 36); myTOLED.pixelSet
(95, 36); myTOLED.pixelSet(96, 36); myTOLED.pixelSet(97, 36); myTOLED.pixelSet(105, 3
6); myTOLED.pixelSet(106, 36); myTOLED.pixelSet(107, 36); myTOLED.pixelSet(108, 36); m
yTOLED.pixelSet(112, 36); myTOLED.pixelSet(113, 36); myTOLED.pixelSet(114, 36); myTOLE
D.pixelSet(117, 36); myTOLED.pixelSet(118, 36); myTOLED.pixelSet(119, 36); myTOLED.pixe
lSet(120, 36); myTOLED.pixelSet(121, 36); myTOLED.pixelSet(122, 36); myTOLED.pixelSet
(123, 36); myTOLED.pixelSet(124, 36);
```

```
myTOLED.pixelSet(4, 37); myTOLED.pixelSet(5, 37); myTOLED.pixelSet(6, 37); myTOLED.p
ixelSet(7, 37); myTOLED.pixelSet(12, 37); myTOLED.pixelSet(13, 37); myTOLED.pixelSet(1
4, 37); myTOLED.pixelSet(15, 37); myTOLED.pixelSet(20, 37); myTOLED.pixelSet(21, 37);
myTOLED.pixelSet(22, 37); myTOLED.pixelSet(23, 37); myTOLED.pixelSet(24, 37); myTOLED.
pixelSet(25, 37); myTOLED.pixelSet(26, 37); myTOLED.pixelSet(27, 37); myTOLED.pixelSet
(28, 37); myTOLED.pixelSet(29, 37); myTOLED.pixelSet(30, 37); myTOLED.pixelSet(31, 3
7); myTOLED.pixelSet(32, 37); myTOLED.pixelSet(33, 37); myTOLED.pixelSet(38, 37); myTO
LED.pixelSet(39, 37); myTOLED.pixelSet(40, 37); myTOLED.pixelSet(41, 37); myTOLED.pixe
```



```
myTOLED.pixelSet(58, 45); myTOLED.pixelSet(67, 45); myTOLED.pixelSet(68, 45); myTOLED.
pixelSet(69, 45); myTOLED.pixelSet(70, 45); myTOLED.pixelSet(74, 45); myTOLED.pixelSet
(75, 45); myTOLED.pixelSet(76, 45); myTOLED.pixelSet(77, 45); myTOLED.pixelSet(78, 4
5); myTOLED.pixelSet(86, 45); myTOLED.pixelSet(87, 45); myTOLED.pixelSet(88, 45); myTO
LED.pixelSet(94, 45); myTOLED.pixelSet(95, 45); myTOLED.pixelSet(96, 45); myTOLED.pixe
lSet(97, 45); myTOLED.pixelSet(105, 45); myTOLED.pixelSet(106, 45); myTOLED.pixelSet(1
07, 45); myTOLED.pixelSet(108, 45); myTOLED.pixelSet(112, 45); myTOLED.pixelSet(113, 4
5); myTOLED.pixelSet(114, 45); myTOLED.pixelSet(115, 45); myTOLED.pixelSet(122, 45); m
yTOLED.pixelSet(123, 45); myTOLED.pixelSet(124, 45); myTOLED.pixelSet(125, 45);
```

```
myTOLED.pixelSet(3, 46); myTOLED.pixelSet(4, 46); myTOLED.pixelSet(5, 46); myTOLED.p
ixelSet(6, 46); myTOLED.pixelSet(13, 46); myTOLED.pixelSet(14, 46); myTOLED.pixelSet(1
5, 46); myTOLED.pixelSet(16, 46); myTOLED.pixelSet(20, 46); myTOLED.pixelSet(21, 46);
myTOLED.pixelSet(22, 46); myTOLED.pixelSet(23, 46); myTOLED.pixelSet(31, 46); myTOLED.
pixelSet(32, 46); myTOLED.pixelSet(33, 46); myTOLED.pixelSet(34, 46); myTOLED.pixelSet
(37, 46); myTOLED.pixelSet(38, 46); myTOLED.pixelSet(39, 46); myTOLED.pixelSet(40, 4
6); myTOLED.pixelSet(48, 46); myTOLED.pixelSet(49, 46); myTOLED.pixelSet(50, 46); myTO
LED.pixelSet(51, 46); myTOLED.pixelSet(55, 46); myTOLED.pixelSet(56, 46); myTOLED.pixe
lSet(57, 46); myTOLED.pixelSet(58, 46); myTOLED.pixelSet(67, 46); myTOLED.pixelSet(6
8, 46); myTOLED.pixelSet(69, 46); myTOLED.pixelSet(70, 46); myTOLED.pixelSet(75, 46);
myTOLED.pixelSet(76, 46); myTOLED.pixelSet(77, 46); myTOLED.pixelSet(78, 46); myTOLED.
pixelSet(86, 46); myTOLED.pixelSet(87, 46); myTOLED.pixelSet(88, 46); myTOLED.pixelSet
(94, 46); myTOLED.pixelSet(95, 46); myTOLED.pixelSet(96, 46); myTOLED.pixelSet(97, 4
6); myTOLED.pixelSet(98, 46); myTOLED.pixelSet(104, 46); myTOLED.pixelSet(105, 46); my
TOLED.pixelSet(106, 46); myTOLED.pixelSet(107, 46); myTOLED.pixelSet(108, 46); myTOLE
D.pixelSet(112, 46); myTOLED.pixelSet(113, 46); myTOLED.pixelSet(114, 46); myTOLED.pix
elSet(115, 46); myTOLED.pixelSet(122, 46); myTOLED.pixelSet(123, 46); myTOLED.pixelSet
(124, 46); myTOLED.pixelSet(125, 46);
```

```
myTOLED.pixelSet(3, 47); myTOLED.pixelSet(4, 47); myTOLED.pixelSet(5, 47); myTOLED.p
ixelSet(6, 47); myTOLED.pixelSet(13, 47); myTOLED.pixelSet(14, 47); myTOLED.pixelSet(1
5, 47); myTOLED.pixelSet(16, 47); myTOLED.pixelSet(20, 47); myTOLED.pixelSet(21, 47);
myTOLED.pixelSet(22, 47); myTOLED.pixelSet(23, 47); myTOLED.pixelSet(24, 47); myTOLED.
pixelSet(30, 47); myTOLED.pixelSet(31, 47); myTOLED.pixelSet(32, 47); myTOLED.pixelSet
(33, 47); myTOLED.pixelSet(34, 47); myTOLED.pixelSet(37, 47); myTOLED.pixelSet(38, 4
7); myTOLED.pixelSet(39, 47); myTOLED.pixelSet(40, 47); myTOLED.pixelSet(47, 47); myTO
LED.pixelSet(48, 47); myTOLED.pixelSet(49, 47); myTOLED.pixelSet(50, 47); myTOLED.pixe
lSet(51, 47); myTOLED.pixelSet(55, 47); myTOLED.pixelSet(56, 47); myTOLED.pixelSet(5
7, 47); myTOLED.pixelSet(58, 47); myTOLED.pixelSet(67, 47); myTOLED.pixelSet(68, 47);
myTOLED.pixelSet(69, 47); myTOLED.pixelSet(70, 47); myTOLED.pixelSet(75, 47); myTOLED.
pixelSet(76, 47); myTOLED.pixelSet(77, 47); myTOLED.pixelSet(78, 47); myTOLED.pixelSet
(79, 47); myTOLED.pixelSet(86, 47); myTOLED.pixelSet(87, 47); myTOLED.pixelSet(88, 4
7); myTOLED.pixelSet(95, 47); myTOLED.pixelSet(96, 47); myTOLED.pixelSet(97, 47); myTO
LED.pixelSet(98, 47); myTOLED.pixelSet(104, 47); myTOLED.pixelSet(105, 47); myTOLED.pi
xelSet(106, 47); myTOLED.pixelSet(107, 47); myTOLED.pixelSet(108, 47); myTOLED.pixelSe
t(112, 47); myTOLED.pixelSet(113, 47); myTOLED.pixelSet(114, 47); myTOLED.pixelSet(11
5, 47); myTOLED.pixelSet(122, 47); myTOLED.pixelSet(123, 47); myTOLED.pixelSet(124, 4
7); myTOLED.pixelSet(125, 47);
```

```
myTOLED.pixelSet(4, 48); myTOLED.pixelSet(5, 48); myTOLED.pixelSet(6, 48); myTOLED.p
ixelSet(7, 48); myTOLED.pixelSet(12, 48); myTOLED.pixelSet(13, 48); myTOLED.pixelSet(1
4, 48); myTOLED.pixelSet(15, 48); myTOLED.pixelSet(16, 48); myTOLED.pixelSet(20, 48);
```



```
myTOLED.pixelSet(21, 48); myTOLED.pixelSet(22, 48); myTOLED.pixelSet(23, 48); myTOLED.
pixelSet(24, 48); myTOLED.pixelSet(25, 48); myTOLED.pixelSet(26, 48); myTOLED.pixelSet
(27, 48); myTOLED.pixelSet(28, 48); myTOLED.pixelSet(29, 48); myTOLED.pixelSet(30, 4
8); myTOLED.pixelSet(31, 48); myTOLED.pixelSet(32, 48); myTOLED.pixelSet(33, 48); myTO
LED.pixelSet(38, 48); myTOLED.pixelSet(39, 48); myTOLED.pixelSet(40, 48); myTOLED.pixe
lSet(41, 48); myTOLED.pixelSet(46, 48); myTOLED.pixelSet(47, 48); myTOLED.pixelSet(4
8, 48); myTOLED.pixelSet(49, 48); myTOLED.pixelSet(50, 48); myTOLED.pixelSet(51, 48);
myTOLED.pixelSet(55, 48); myTOLED.pixelSet(56, 48); myTOLED.pixelSet(57, 48); myTOLED.
pixelSet(58, 48); myTOLED.pixelSet(67, 48); myTOLED.pixelSet(68, 48); myTOLED.pixelSet
(69, 48); myTOLED.pixelSet(70, 48); myTOLED.pixelSet(76, 48); myTOLED.pixelSet(77, 4
8); myTOLED.pixelSet(78, 48); myTOLED.pixelSet(79, 48); myTOLED.pixelSet(80, 48); myTO
LED.pixelSet(86, 48); myTOLED.pixelSet(87, 48); myTOLED.pixelSet(88, 48); myTOLED.pixe
lSet(95, 48); myTOLED.pixelSet(96, 48); myTOLED.pixelSet(97, 48); myTOLED.pixelSet(9
8, 48); myTOLED.pixelSet(99, 48); myTOLED.pixelSet(100, 48); myTOLED.pixelSet(101, 4
8); myTOLED.pixelSet(102, 48); myTOLED.pixelSet(103, 48); myTOLED.pixelSet(104, 48); m
yTOLED.pixelSet(105, 48); myTOLED.pixelSet(106, 48); myTOLED.pixelSet(107, 48); myTOLE
D.pixelSet(108, 48); myTOLED.pixelSet(112, 48); myTOLED.pixelSet(113, 48); myTOLED.pix
elSet(114, 48); myTOLED.pixelSet(115, 48); myTOLED.pixelSet(122, 48); myTOLED.pixelSet
(123, 48); myTOLED.pixelSet(124, 48); myTOLED.pixelSet(125, 48);
```

```
myTOLED.pixelSet(4, 49); myTOLED.pixelSet(5, 49); myTOLED.pixelSet(6, 49); myTOLED.p
ixelSet(7, 49); myTOLED.pixelSet(8, 49); myTOLED.pixelSet(9, 49); myTOLED.pixelSet(1
0, 49); myTOLED.pixelSet(11, 49); myTOLED.pixelSet(12, 49); myTOLED.pixelSet(13, 49);
myTOLED.pixelSet(14, 49); myTOLED.pixelSet(15, 49); myTOLED.pixelSet(20, 49); myTOLED.
pixelSet(21, 49); myTOLED.pixelSet(22, 49); myTOLED.pixelSet(23, 49); myTOLED.pixelSet
(24, 49); myTOLED.pixelSet(25, 49); myTOLED.pixelSet(26, 49); myTOLED.pixelSet(27, 4
9); myTOLED.pixelSet(28, 49); myTOLED.pixelSet(29, 49); myTOLED.pixelSet(30, 49); myTO
LED.pixelSet(31, 49); myTOLED.pixelSet(32, 49); myTOLED.pixelSet(38, 49); myTOLED.pixe
lSet(39, 49); myTOLED.pixelSet(40, 49); myTOLED.pixelSet(41, 49); myTOLED.pixelSet(4
2, 49); myTOLED.pixelSet(43, 49); myTOLED.pixelSet(44, 49); myTOLED.pixelSet(45, 49);
myTOLED.pixelSet(46, 49); myTOLED.pixelSet(47, 49); myTOLED.pixelSet(48, 49); myTOLED.
pixelSet(49, 49); myTOLED.pixelSet(50, 49); myTOLED.pixelSet(51, 49); myTOLED.pixelSet
(55, 49); myTOLED.pixelSet(56, 49); myTOLED.pixelSet(57, 49); myTOLED.pixelSet(58, 4
9); myTOLED.pixelSet(67, 49); myTOLED.pixelSet(68, 49); myTOLED.pixelSet(69, 49); myTO
LED.pixelSet(70, 49); myTOLED.pixelSet(77, 49); myTOLED.pixelSet(78, 49); myTOLED.pixe
lSet(79, 49); myTOLED.pixelSet(80, 49); myTOLED.pixelSet(86, 49); myTOLED.pixelSet(8
7, 49); myTOLED.pixelSet(88, 49); myTOLED.pixelSet(95, 49); myTOLED.pixelSet(96, 49);
myTOLED.pixelSet(97, 49); myTOLED.pixelSet(98, 49); myTOLED.pixelSet(99, 49); myTOLED.
pixelSet(100, 49); myTOLED.pixelSet(101, 49); myTOLED.pixelSet(102, 49); myTOLED.pixel
Set(103, 49); myTOLED.pixelSet(105, 49); myTOLED.pixelSet(106, 49); myTOLED.pixelSet(1
07, 49); myTOLED.pixelSet(108, 49); myTOLED.pixelSet(112, 49); myTOLED.pixelSet(113, 4
9); myTOLED.pixelSet(114, 49); myTOLED.pixelSet(115, 49); myTOLED.pixelSet(122, 49); m
yTOLED.pixelSet(123, 49); myTOLED.pixelSet(124, 49); myTOLED.pixelSet(125, 49);
```

```
myTOLED.pixelSet(6, 50); myTOLED.pixelSet(7, 50); myTOLED.pixelSet(8, 50); myTOLED.p
ixelSet(9, 50); myTOLED.pixelSet(10, 50); myTOLED.pixelSet(11, 50); myTOLED.pixelSet(1
2, 50); myTOLED.pixelSet(13, 50); myTOLED.pixelSet(20, 50); myTOLED.pixelSet(21, 50);
myTOLED.pixelSet(22, 50); myTOLED.pixelSet(23, 50); myTOLED.pixelSet(26, 50); myTOLED.
pixelSet(27, 50); myTOLED.pixelSet(28, 50); myTOLED.pixelSet(29, 50); myTOLED.pixelSet
(30, 50); myTOLED.pixelSet(31, 50); myTOLED.pixelSet(39, 50); myTOLED.pixelSet(40, 5
0); myTOLED.pixelSet(41, 50); myTOLED.pixelSet(42, 50); myTOLED.pixelSet(43, 50); myTO
```

```

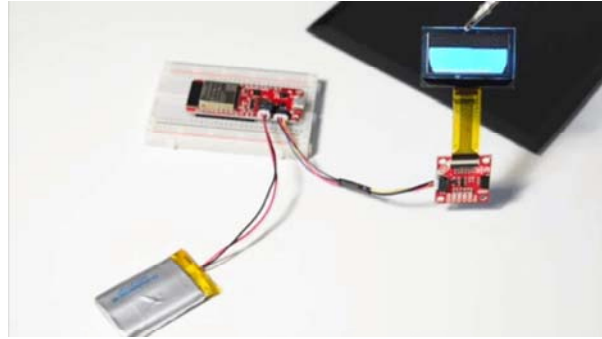
LED.pixelSet(44, 50); myTOLED.pixelSet(45, 50); myTOLED.pixelSet(48, 50); myTOLED.pixe
lSet(49, 50); myTOLED.pixelSet(50, 50); myTOLED.pixelSet(51, 50); myTOLED.pixelSet(5
5, 50); myTOLED.pixelSet(56, 50); myTOLED.pixelSet(57, 50); myTOLED.pixelSet(58, 50);
myTOLED.pixelSet(67, 50); myTOLED.pixelSet(68, 50); myTOLED.pixelSet(69, 50); myTOLED.
pixelSet(70, 50); myTOLED.pixelSet(77, 50); myTOLED.pixelSet(78, 50); myTOLED.pixelSet
(79, 50); myTOLED.pixelSet(80, 50); myTOLED.pixelSet(81, 50); myTOLED.pixelSet(86, 5
0); myTOLED.pixelSet(87, 50); myTOLED.pixelSet(88, 50); myTOLED.pixelSet(96, 50); myTO
LED.pixelSet(97, 50); myTOLED.pixelSet(98, 50); myTOLED.pixelSet(99, 50); myTOLED.pixe
lSet(100, 50); myTOLED.pixelSet(101, 50); myTOLED.pixelSet(102, 50); myTOLED.pixelSet
(105, 50); myTOLED.pixelSet(106, 50); myTOLED.pixelSet(107, 50); myTOLED.pixelSet(10
8, 50); myTOLED.pixelSet(112, 50); myTOLED.pixelSet(113, 50); myTOLED.pixelSet(114, 5
0); myTOLED.pixelSet(115, 50); myTOLED.pixelSet(122, 50); myTOLED.pixelSet(123, 50); m
yTOLED.pixelSet(124, 50); myTOLED.pixelSet(125, 50);
  myTOLED.pixelSet(20, 51); myTOLED.pixelSet(21, 51); myTOLED.pixelSet(22, 51); myTOLE
D.pixelSet(23, 51);
  myTOLED.pixelSet(20, 52); myTOLED.pixelSet(21, 52); myTOLED.pixelSet(22, 52); myTOLE
D.pixelSet(23, 52);
  myTOLED.pixelSet(20, 53); myTOLED.pixelSet(21, 53); myTOLED.pixelSet(22, 53); myTOLE
D.pixelSet(23, 53);
  myTOLED.pixelSet(20, 54); myTOLED.pixelSet(21, 54); myTOLED.pixelSet(22, 54); myTOLE
D.pixelSet(23, 54);
  myTOLED.pixelSet(20, 55); myTOLED.pixelSet(21, 55); myTOLED.pixelSet(22, 55); myTOLE
D.pixelSet(23, 55);
  myTOLED.pixelSet(20, 56); myTOLED.pixelSet(21, 56); myTOLED.pixelSet(22, 56); myTOLE
D.pixelSet(23, 56);
  myTOLED.pixelSet(21, 57); myTOLED.pixelSet(22, 57); myTOLED.pixelSet(23, 57);
  myTOLED.pixelSet(22, 58); myTOLED.pixelSet(23, 58);
  myTOLED.pixelSet(23, 59);

for(uint8_t indi = 0; indi < 254; indi++)
{
  myTOLED.setContrastControl(indi);
  delay(5);
}
delay(5000);
for(uint8_t indi = 255; indi > 1; indi--)
{
  myTOLED.setContrastControl(indi);
  delay(5);
}
myTOLED.setContrastControl(0);
myTOLED.clearDisplay();
myTOLED.setContrastControl(128);
}
#endif

```

Once you've uploaded the code to your board successfully, you should see something like the GIF

below:



If you have a look at the `showLogo` function, you'll see that the SparkFun logo is calculated and drawn out pixel by pixel. This was calculated using the `image_to_pixel` python script above. The image is drawn this way due to memory limitations in some Arduino boards. Perhaps not the most efficient way to do this, but it does the trick.

You'll also notice that the comments in this example are sparse - this example is intended to get you up and running and show you that the OLED works. The next example is where we will more thoroughly demonstrate how to make your own application with the drawing functions.

Example 2: DrawingBasics

This example shows some of the basic functions for drawing shapes, including line, rectangle, circle, and polygon. Once uploaded, you should notice different shapes showing up on your transparent OLED. Either select **File > Examples > SparkFun HyperDisplay Transparent Graphical OLED Library > Example2_DrawingBasics** or copy and paste the code below into a new Arduino window.

```

/*
  Take a guided tour of the basic capabilities of the Transparent Graphical OLED

  By: Owen Lyke
  SparkFun Electronics
  Date: February 27, 2019
  License: MIT. See license file for more information but you can
  basically do whatever you want with this code.
  Feel like supporting open source hardware?
  Buy a board from SparkFun! https://www.sparkfun.com/products/15173

  The graphics library is like a 3-layer cake. Here they are from top-down
  https://github.com/sparkfun/SparkFun\_HyperDisplay
  https://github.com/sparkfun/HyperDisplay\_SSD1309\_ArduinoLibrary
  https://github.com/sparkfun/HyperDisplay\_UG2856KLBAG01\_ArduinoLibrary

  Hardware Compatibility
  - The IO pins on this board are designed for use with 3.3V so if you are using a 5
  V microcontroller
    please use a level shifter. Note: Qwiic connectors on SparkFun dev boards are al
  ready at 3.3V
  - This display relies on a copy of graphics data in your microcontroller, a total
  of 1024 bytes.
    That is half the RAM available on an Uno so it is easy to run into sinister low-
  memory related
    bugs. We reccomend using a micro with more memory like a SAMD21, Esp32, Teensy,
  etc.

  Hardware Connections:
  Option 1 (I2C):
    Connect using a Qwiic jumper if you have a Qwiic compatible board and you plan to
  use I2C

  Option 2 (SPI):
    Connect SCLK and MOSI to the SPI port of your choice (13 and 11 for SPI on Uno-lik
  e boards)
    Also connect D/C and CS to two unused GPIO pins of your choice (and set the prope
  r pin definitions below)
    Don't forget power - connect 3.3V and GND
*/

#include "HyperDisplay_UG2856KLBAG01.h" // Your library can be installed here: htt
p://librarymanager/All#SparkFun_Transparent_Graphical_OLED
// The rest of the Layer Cake:      htt
p://librarymanager/All#SparkFun_HyperDisplay_SSD1309
//                                  htt

```

```

p://librarymanager/All#SparkFun_HyperDisplay
////////////////////////////////////
//      User Setup      //
////////////////////////////////////
#define SERIAL_PORT Serial
#define WIRE_PORT Wire    // Used if USE_SPI == 0
#define SPI_PORT SPI      // Used if USE_SPI == 1

#define RES_PIN 2        // Optional
#define CS_PIN 4         // Used only if USE_SPI == 1
#define DC_PIN 5         // Used only if USE_SPI == 1

#define USE_SPI 0        // Choose your interface. 0 = I2C, 1 = SPI

// END USER SETUP

// Object Declaration. A class exists for each interface option
#if USE_SPI
  UG2856KLBAG01_SPI myTOLED; // Declare a SPI-based Transparent OLED object called my
  TOLED
#else
  UG2856KLBAG01_I2C myTOLED; // Declare a I2C-based Transparent OLED object called my
  TOLED
#endif /* USE_SPI */

void setup() {
  Serial.begin(9600);

  Serial.println("Example2_DrawingBasics: Transparent Graphical OLED");

#if USE_SPI
  SPI_PORT.begin();
  myTOLED.begin(CS_PIN, DC_PIN, SPI_PORT); // Begin for SPI requires
  that you provide the CS and DC pin numbers
#else
  WIRE_PORT.begin();
  myTOLED.begin(WIRE_PORT, false, SSD1309_ARD_UNUSED_PIN); // Begin for I2C has defau
  lt values for every argument
  Wire.setClock(400000);
#endif /* USE_SPI */

  /*
   * Drawing with the Transparent Graphical OLED
   *
   * The TOLED display is a 1-bit per pixel display, meaning that each pixel can eith
   er be on or off. It is also possible to
   * raise/lower the contrast of the entire display, but not any pixel individually.
   *
  */

```

```

* This simplicity allows the TOLED HyperDisplay driver not to worry about color, i
nstead each HyperDisplay function is
* given two variants - one to 'Set' the pixels and one to 'Clear' the pixels. In t
h following code block we will try
* out a few of those functions. Note, all the 'Set' functions have a 'Clear' count
erpart that turns off pixels, but we
* will only show those in comments to avoid redundancy
*/

uint8_t x0      = 0;
uint8_t y0      = 0;
uint8_t x1      = 0;
uint8_t y1      = 0;
uint8_t r       = 0;

x0 = 5;
y0 = 5;
myTOLED.pixelSet(x0, y0);          // Provide the X and Y locations of the singl
e pixel to set or clear
// myTOLED.pixelClear(x0, y0);    //

x0 = 10;
y0 = 10;
x1 = 65;
y1 = 35;
myTOLED.rectangleSet(x0, y0, x1, y1);          // Draw a rectangle outline fro
m (x0, y0) to (x1, y1)
myTOLED.rectangleSet(x0+2, y0+2, x1-2, y1-2, true); // Supply the 'filled' paramete
r as true to fill in the rectangle
// myTOLED.rectangleClear(x0, y0, x1, y1);
// myTOLED.rectangleClear(x0+2, y0+2, x1-2, y1-2, true);

x0 = 70;
y0 = 5;
x1 = 85;
y1 = 50;
myTOLED.lineSet(x0, y0, x1, y1);          // Draw a simple line from (x0, y0) to (x
1, y1)
myTOLED.lineSet(x0+5, y0, x1+5, y1, 3); // Draw a line with width 3
// myTOLED.lineSet(x0, y0, x1, y1);
// myTOLED.lineSet(x0+5, y0, x1+5, y1, 3);

x0 = 108;
y1 = 24;
r = 14;
myTOLED.circleSet(x0, y1, r);          // Outline circle centered at (x0, y0) wit
h radius r
myTOLED.circleSet(x0, y1, r-4, true); // Filled in
// myTOLED.circleSet(x0, y1, r);

```

```

// myTOLED.circleSet(x0, y1, r-4, true);

hd_extent_t x[] = {5, 42, 52};
hd_extent_t y[] = {48, 40, 60};
uint8_t numSides = sizeof(x)/sizeof(hd_extent_t);
myTOLED.polygonSet(x, y, numSides);           // Make a polygon with numSides a
t the x, y coordinate pairs
// myTOLED.polygonClear(x, y, numSides);
}

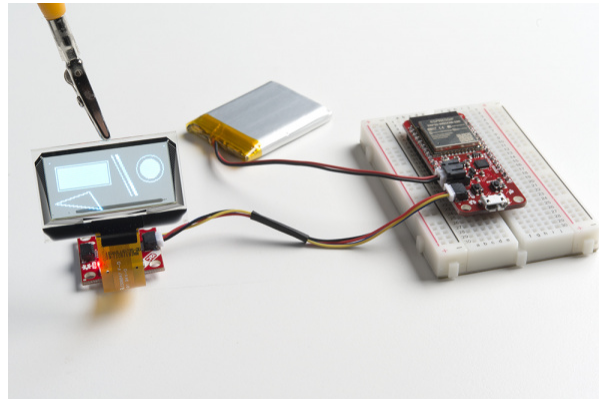
void loop() {

  myTOLED.pixelSet(126,2);
  delay(1000);

  myTOLED.pixelClear(126,2);
  delay(1000);

}

```



Although **Example2** is detailed in how to use the drawing functions, it falls short in explaining one thing; specifically the options you have when creating and beginning the `myTOLED` object. When creating the object you can choose either the `_I2C` or the `_SPI` suffix to the name of the class (UG2856KLBAG01). That of course changes the intended communication protocol.

If you're using Qwiic, choose I²C. The interface decision also impacts your use of the `begin()` function by changing which parameters are expected. For I²C, you can specify an Arduino 'Wire' object to use, which I²C address to use, and a pin to use to select the I²C address.

For I²C, all parameters have defaults so it works to just use `myTOLED.begin()`; For SPI, the first two arguments are required but the third (which allows you to choose which SPI port to use) can be left out (e.g. `myTOLED.begin(CS_PIN, DC_PIN);`).

Example 3: AdvancedFeatures

This example has some serious Mario goalz. When you upload this, you'll see Mario jumping up and down and eventually landing on a pipe. Either select **File > Examples > SparkFun HyperDisplay Transparent Graphical OLED Library > Example3_AdvancedFeatures** or copy and paste the code below into a new Arduino window.


```

/*
  Take a guided tour of the basic capabilities of the Transparent Graphical OLED

  By: Owen Lyke
  SparkFun Electronics
  Date: February 28, 2019
  License: MIT. See license file for more information but you can
  basically do whatever you want with this code.
  Feel like supporting open source hardware?
  Buy a board from SparkFun! https://www.sparkfun.com/products/15173

  The graphics library is like a 3-layer cake. Here they are from top-down
  https://github.com/sparkfun/SparkFun\_HyperDisplay
  https://github.com/sparkfun/HyperDisplay\_SSD1309\_ArduinoLibrary
  https://github.com/sparkfun/HyperDisplay\_UG2856KLBAG01\_ArduinoLibrary

  Hardware Compatibility
  - The IO pins on this board are designed for use with 3.3V so if you are using a 5
  V microcontroller
    please use a level shifter. Note: Qwiic connectors on SparkFun dev boards are al
  ready at 3.3V
  - This display relies on a copy of graphics data in your microcontroller, a total
  of 1024 bytes.
    That is half the RAM available on an Uno so it is easy to run into sinister low-
  memory related
    bugs. We reccomend using a micro with more memory like a SAMD21, Esp32, Teensy,
  etc.

  Hardware Connections:
  Option 1 (I2C):
    Connect using a Qwiic jumper if you have a Qwiic compatible board and you plan to
  use I2C

  Option 2 (SPI):
    Connect SCLK and MOSI to the SPI port of your choice (13 and 11 for SPI on Uno-lik
  e boards)
    Also connect D/C and CS to two unused GPIO pins of your choice (and set the prope
  r pin definitions below)
    Don't forget power - connect 3.3V and GND
*/

#include "HyperDisplay_UG2856KLBAG01.h" // Your library can be installed here: htt
p://librarymanager/All#SparkFun_Transparent_Graphical_OLED
// The rest of the Layer Cake:      htt
p://librarymanager/All#SparkFun_HyperDisplay_SSD1309
//                                  htt

```

```

p://librarymanager/All#SparkFun_HyperDisplay
////////////////////////////////////
//      User Setup      //
////////////////////////////////////
#define SERIAL_PORT Serial
#define WIRE_PORT Wire    // Used if USE_SPI == 0
#define SPI_PORT SPI      // Used if USE_SPI == 1

#define RES_PIN 2         // Optional
#define CS_PIN 4          // Used only if USE_SPI == 1
#define DC_PIN 5          // Used only if USE_SPI == 1

#define USE_SPI 0         // Choose your interface. 0 = I2C, 1 = SPI

// END USER SETUP

// Object Declaration. A class exists for each interface option
#if USE_SPI
  UG2856KLBAG01_SPI myTOLED; // Declare a SPI-based Transparent OLED object called my
  TOLED
#else
  UG2856KLBAG01_I2C myTOLED; // Declare a I2C-based Transparent OLED object called my
  TOLED
#endif /* USE_SPI */

void setup() {
  Serial.begin(9600);

  Serial.println("Example3_AdvancedFeatures: Transparent Graphical OLED");

#if USE_SPI
  SPI_PORT.begin();
  myTOLED.begin(CS_PIN, DC_PIN, SPI_PORT); // Begin for SPI requires
  that you provide the CS and DC pin numbers
#else
  WIRE_PORT.begin();
  myTOLED.begin(WIRE_PORT, false, SSD1309_ARD_UNUSED_PIN); // Begin for I2C has defau
  lt values for every argument
  Wire.setClock(400000);
#endif /* USE_SPI */

  wind_info_t defaultWindow, myWindow; // Create some window objects

  // Initialize the windows to default settings
  // (this is a pretty important step unless you are extra
  // careful to manually initialize each and every paramter)
  myTOLED.setWindowDefaults(&defaultWindow);

```

```

myTOLED.setWindowDefaults(&myWindow);

// We can set the rectangular active area of the window
myWindow.xMin = 25;
myWindow.yMin = 05;
myWindow.xMax = 73;
myWindow.yMax = 48;

// All drawing functions use window-relative coordinates
// This makes it easy to reposition whole compositions

// First let's draw a line on the default window (which is equivalent to the whole s
creen)
myTOLED.lineSet(0, 0, 15, 43);

// Now let's change the active window and draw the same line
delay(1000);
myTOLED.pCurrentWindow = &myWindow;
myTOLED.lineSet(0, 0, 15, 43);

delay(2000);

// To better illustrate where the window is let's fill it in..
myTOLED.windowSet(); // Calling windowSet without arguments operates on the curren
t window.
// myTOLED.windowClear();

delay(2000);

// Remember, drawing functions are relative to the current window,
// so you don't have to do extra math!
myTOLED.circleClear(25, 20, 10); // Add some pizzazz!

delay(2000);

// You can also pass a window to windowSet/Clear to fill it without changing the cur
rent window
myTOLED.windowClear(&defaultWindow); // The default initialization will make this cl
ear the whole screen

}

void loop() {

#if !defined(__AVR_ATmega328P__) && !defined(__AVR_ATmega168__)
// Now for an example of what you might do with a window...

wind_info_t mariosWindow, pipeWindow;
myTOLED.setWindowDefaults(&mariosWindow);

```

```
myTOLED.setWindowDefaults(&pipeWindow);

// Make the window just big enough for Mario
mariosWindow.xMax = 21;
mariosWindow.yMax = 28;

pipeWindow.xMax = 24;
pipeWindow.yMax = 24;

// Move the windows into the starting position
moveWindow(0, 34, &mariosWindow);
moveWindow(103, 63-25, &pipeWindow);

// Draw Pipe
myTOLED.pCurrentWindow = &pipeWindow;
showPipe();

// Draw Mario at the starting position
myTOLED.pCurrentWindow = &mariosWindow;
showMario();

// Move mario to his home
myTOLED.windowClear();
moveWindow(15, -15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(15, 15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(13, -15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(13, 15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(13, -15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(13, 15, &mariosWindow);
showMario();

myTOLED.windowClear();
moveWindow(5, -23, &mariosWindow);
```

```

showMario();

myTOLED.windowClear();
moveWindow(17, 0, &mariosWindow);
showMario();

delay(5000);

myTOLED.windowClear();

#endif
}

void moveWindow( int8_t x, int8_t y, wind_info_t* pwind){
    pwind->xMin += x;
    pwind->yMin += y;
    pwind->xMax += x;
    pwind->yMax += y;
}

#if !defined(__AVR_ATmega328P__) && !defined(__AVR_ATmega168__)
void showMario( void ){
    myTOLED.pixelSet(6, 2); myTOLED.pixelSet(7, 2); myTOLED.pixelSet(8, 2); myTOLED.pixelSet(9, 2); myTOLED.pixelSet(10, 2); myTOLED.pixelSet(11, 2); myTOLED.pixelSet(12, 2); myTOLED.pixelSet(13, 2);
    myTOLED.pixelSet(5, 3); myTOLED.pixelSet(6, 3); myTOLED.pixelSet(7, 3); myTOLED.pixelSet(8, 3); myTOLED.pixelSet(9, 3); myTOLED.pixelSet(10, 3); myTOLED.pixelSet(11, 3); myTOLED.pixelSet(12, 3); myTOLED.pixelSet(13, 3); myTOLED.pixelSet(14, 3); myTOLED.pixelSet(15, 3); myTOLED.pixelSet(16, 3); myTOLED.pixelSet(17, 3); myTOLED.pixelSet(18, 3);
    myTOLED.pixelSet(4, 4); myTOLED.pixelSet(5, 4); myTOLED.pixelSet(6, 4); myTOLED.pixelSet(7, 4); myTOLED.pixelSet(8, 4); myTOLED.pixelSet(9, 4); myTOLED.pixelSet(10, 4); myTOLED.pixelSet(11, 4); myTOLED.pixelSet(12, 4); myTOLED.pixelSet(13, 4); myTOLED.pixelSet(14, 4); myTOLED.pixelSet(15, 4); myTOLED.pixelSet(16, 4); myTOLED.pixelSet(17, 4); myTOLED.pixelSet(18, 4);
    myTOLED.pixelSet(4, 5); myTOLED.pixelSet(5, 5); myTOLED.pixelSet(6, 5); myTOLED.pixelSet(7, 5); myTOLED.pixelSet(8, 5); myTOLED.pixelSet(9, 5); myTOLED.pixelSet(10, 5); myTOLED.pixelSet(11, 5); myTOLED.pixelSet(12, 5); myTOLED.pixelSet(13, 5); myTOLED.pixelSet(14, 5); myTOLED.pixelSet(15, 5);
    myTOLED.pixelSet(4, 6); myTOLED.pixelSet(5, 6); myTOLED.pixelSet(6, 6); myTOLED.pixelSet(7, 6); myTOLED.pixelSet(8, 6); myTOLED.pixelSet(9, 6); myTOLED.pixelSet(10, 6); myTOLED.pixelSet(11, 6); myTOLED.pixelSet(12, 6); myTOLED.pixelSet(13, 6); myTOLED.pixelSet(14, 6); myTOLED.pixelSet(15, 6);
    myTOLED.pixelSet(3, 7); myTOLED.pixelSet(4, 7); myTOLED.pixelSet(5, 7); myTOLED.pixelSet(6, 7); myTOLED.pixelSet(7, 7); myTOLED.pixelSet(8, 7); myTOLED.pixelSet(9, 7); myTOLED.pixelSet(10, 7); myTOLED.pixelSet(11, 7); myTOLED.pixelSet(12, 7); myTOLED.pixel

```

```
Set(13, 7); myTOLED.pixelSet(14, 7); myTOLED.pixelSet(15, 7); myTOLED.pixelSet(16,
7); myTOLED.pixelSet(17, 7); myTOLED.pixelSet(18, 7);
    myTOLED.pixelSet(3, 8); myTOLED.pixelSet(4, 8); myTOLED.pixelSet(5, 8); myTOLED.pixe
lSet(6, 8); myTOLED.pixelSet(7, 8); myTOLED.pixelSet(8, 8); myTOLED.pixelSet(9, 8); my
TOLED.pixelSet(10, 8); myTOLED.pixelSet(11, 8); myTOLED.pixelSet(12, 8); myTOLED.pixel
Set(13, 8); myTOLED.pixelSet(14, 8); myTOLED.pixelSet(15, 8); myTOLED.pixelSet(16,
8); myTOLED.pixelSet(17, 8); myTOLED.pixelSet(18, 8);
    myTOLED.pixelSet(3, 9); myTOLED.pixelSet(4, 9); myTOLED.pixelSet(5, 9); myTOLED.pixe
lSet(6, 9); myTOLED.pixelSet(7, 9); myTOLED.pixelSet(8, 9); myTOLED.pixelSet(9, 9); my
TOLED.pixelSet(10, 9); myTOLED.pixelSet(11, 9); myTOLED.pixelSet(12, 9); myTOLED.pixel
Set(13, 9); myTOLED.pixelSet(14, 9); myTOLED.pixelSet(15, 9); myTOLED.pixelSet(16,
9); myTOLED.pixelSet(17, 9); myTOLED.pixelSet(18, 9); myTOLED.pixelSet(19, 9);
    myTOLED.pixelSet(3, 10); myTOLED.pixelSet(4, 10); myTOLED.pixelSet(5, 10); myTOLED.p
ixelSet(6, 10); myTOLED.pixelSet(7, 10); myTOLED.pixelSet(8, 10); myTOLED.pixelSet(9,
10); myTOLED.pixelSet(10, 10); myTOLED.pixelSet(11, 10); myTOLED.pixelSet(12, 10); myT
OLED.pixelSet(13, 10); myTOLED.pixelSet(14, 10); myTOLED.pixelSet(15, 10); myTOLED.pix
elSet(16, 10); myTOLED.pixelSet(17, 10); myTOLED.pixelSet(18, 10);
    myTOLED.pixelSet(3, 11); myTOLED.pixelSet(4, 11); myTOLED.pixelSet(5, 11); myTOLED.p
ixelSet(6, 11); myTOLED.pixelSet(7, 11); myTOLED.pixelSet(8, 11); myTOLED.pixelSet(9,
11); myTOLED.pixelSet(10, 11); myTOLED.pixelSet(11, 11); myTOLED.pixelSet(12, 11); myT
OLED.pixelSet(13, 11); myTOLED.pixelSet(14, 11); myTOLED.pixelSet(15, 11); myTOLED.pix
elSet(16, 11); myTOLED.pixelSet(17, 11); myTOLED.pixelSet(18, 11);
    myTOLED.pixelSet(6, 12); myTOLED.pixelSet(7, 12); myTOLED.pixelSet(8, 12); myTOLED.p
ixelSet(9, 12); myTOLED.pixelSet(10, 12); myTOLED.pixelSet(11, 12); myTOLED.pixelSet(1
2, 12); myTOLED.pixelSet(13, 12); myTOLED.pixelSet(14, 12); myTOLED.pixelSet(15, 12);
myTOLED.pixelSet(16, 12);
    myTOLED.pixelSet(5, 13); myTOLED.pixelSet(6, 13); myTOLED.pixelSet(7, 13); myTOLED.p
ixelSet(8, 13); myTOLED.pixelSet(9, 13); myTOLED.pixelSet(10, 13); myTOLED.pixelSet(1
1, 13); myTOLED.pixelSet(12, 13); myTOLED.pixelSet(13, 13);
    myTOLED.pixelSet(4, 14); myTOLED.pixelSet(5, 14); myTOLED.pixelSet(6, 14); myTOLED.p
ixelSet(7, 14); myTOLED.pixelSet(8, 14); myTOLED.pixelSet(9, 14); myTOLED.pixelSet(1
0, 14); myTOLED.pixelSet(11, 14); myTOLED.pixelSet(12, 14); myTOLED.pixelSet(13, 14);
    myTOLED.pixelSet(3, 15); myTOLED.pixelSet(4, 15); myTOLED.pixelSet(5, 15); myTOLED.p
ixelSet(6, 15); myTOLED.pixelSet(7, 15); myTOLED.pixelSet(8, 15); myTOLED.pixelSet(9,
15); myTOLED.pixelSet(10, 15); myTOLED.pixelSet(11, 15); myTOLED.pixelSet(12, 15); myT
OLED.pixelSet(13, 15); myTOLED.pixelSet(14, 15); myTOLED.pixelSet(15, 15); myTOLED.pix
elSet(16, 15); myTOLED.pixelSet(17, 15); myTOLED.pixelSet(18, 15);
    myTOLED.pixelSet(2, 16); myTOLED.pixelSet(3, 16); myTOLED.pixelSet(4, 16); myTOLED.p
ixelSet(5, 16); myTOLED.pixelSet(6, 16); myTOLED.pixelSet(7, 16); myTOLED.pixelSet(8,
16); myTOLED.pixelSet(9, 16); myTOLED.pixelSet(10, 16); myTOLED.pixelSet(11, 16); myTO
LED.pixelSet(12, 16); myTOLED.pixelSet(13, 16); myTOLED.pixelSet(14, 16); myTOLED.pixe
lSet(15, 16); myTOLED.pixelSet(16, 16); myTOLED.pixelSet(17, 16); myTOLED.pixelSet(1
8, 16);
    myTOLED.pixelSet(1, 17); myTOLED.pixelSet(2, 17); myTOLED.pixelSet(3, 17); myTOLED.p
ixelSet(4, 17); myTOLED.pixelSet(5, 17); myTOLED.pixelSet(6, 17); myTOLED.pixelSet(7,
17); myTOLED.pixelSet(8, 17); myTOLED.pixelSet(9, 17); myTOLED.pixelSet(10, 17); myTOL
ED.pixelSet(11, 17); myTOLED.pixelSet(12, 17); myTOLED.pixelSet(13, 17); myTOLED.pixel
Set(14, 17); myTOLED.pixelSet(15, 17); myTOLED.pixelSet(16, 17); myTOLED.pixelSet(17,
17); myTOLED.pixelSet(18, 17); myTOLED.pixelSet(19, 17);
```

```
myTOLED.pixelSet(1, 18); myTOLED.pixelSet(2, 18); myTOLED.pixelSet(3, 18); myTOLED.p
ixelSet(4, 18); myTOLED.pixelSet(5, 18); myTOLED.pixelSet(6, 18); myTOLED.pixelSet(7,
18); myTOLED.pixelSet(8, 18); myTOLED.pixelSet(9, 18); myTOLED.pixelSet(10, 18); myTOL
ED.pixelSet(11, 18); myTOLED.pixelSet(12, 18); myTOLED.pixelSet(13, 18); myTOLED.pixel
Set(14, 18); myTOLED.pixelSet(15, 18); myTOLED.pixelSet(16, 18); myTOLED.pixelSet(17,
18); myTOLED.pixelSet(18, 18); myTOLED.pixelSet(19, 18);
myTOLED.pixelSet(1, 19); myTOLED.pixelSet(2, 19); myTOLED.pixelSet(3, 19); myTOLED.p
ixelSet(4, 19); myTOLED.pixelSet(5, 19); myTOLED.pixelSet(6, 19); myTOLED.pixelSet(7,
19); myTOLED.pixelSet(8, 19); myTOLED.pixelSet(9, 19); myTOLED.pixelSet(10, 19); myTOL
ED.pixelSet(11, 19); myTOLED.pixelSet(12, 19); myTOLED.pixelSet(13, 19); myTOLED.pixel
Set(14, 19); myTOLED.pixelSet(15, 19); myTOLED.pixelSet(16, 19); myTOLED.pixelSet(17,
19); myTOLED.pixelSet(18, 19); myTOLED.pixelSet(19, 19);
myTOLED.pixelSet(1, 20); myTOLED.pixelSet(2, 20); myTOLED.pixelSet(3, 20); myTOLED.p
ixelSet(4, 20); myTOLED.pixelSet(5, 20); myTOLED.pixelSet(6, 20); myTOLED.pixelSet(7,
20); myTOLED.pixelSet(8, 20); myTOLED.pixelSet(9, 20); myTOLED.pixelSet(10, 20); myTOL
ED.pixelSet(11, 20); myTOLED.pixelSet(12, 20); myTOLED.pixelSet(13, 20); myTOLED.pixel
Set(14, 20); myTOLED.pixelSet(15, 20); myTOLED.pixelSet(16, 20); myTOLED.pixelSet(17,
20); myTOLED.pixelSet(18, 20); myTOLED.pixelSet(19, 20);
myTOLED.pixelSet(1, 21); myTOLED.pixelSet(2, 21); myTOLED.pixelSet(3, 21); myTOLED.p
ixelSet(4, 21); myTOLED.pixelSet(5, 21); myTOLED.pixelSet(6, 21); myTOLED.pixelSet(7,
21); myTOLED.pixelSet(8, 21); myTOLED.pixelSet(9, 21); myTOLED.pixelSet(10, 21); myTOL
ED.pixelSet(11, 21); myTOLED.pixelSet(12, 21); myTOLED.pixelSet(13, 21); myTOLED.pixel
Set(14, 21); myTOLED.pixelSet(15, 21); myTOLED.pixelSet(16, 21); myTOLED.pixelSet(17,
21); myTOLED.pixelSet(18, 21); myTOLED.pixelSet(19, 21);
myTOLED.pixelSet(1, 22); myTOLED.pixelSet(2, 22); myTOLED.pixelSet(3, 22); myTOLED.p
ixelSet(4, 22); myTOLED.pixelSet(5, 22); myTOLED.pixelSet(6, 22); myTOLED.pixelSet(7,
22); myTOLED.pixelSet(8, 22); myTOLED.pixelSet(9, 22); myTOLED.pixelSet(10, 22); myTOL
ED.pixelSet(11, 22); myTOLED.pixelSet(12, 22); myTOLED.pixelSet(13, 22); myTOLED.pixel
Set(14, 22); myTOLED.pixelSet(15, 22); myTOLED.pixelSet(16, 22); myTOLED.pixelSet(17,
22); myTOLED.pixelSet(18, 22); myTOLED.pixelSet(19, 22);
myTOLED.pixelSet(4, 23); myTOLED.pixelSet(5, 23); myTOLED.pixelSet(6, 23); myTOLED.p
ixelSet(7, 23); myTOLED.pixelSet(8, 23); myTOLED.pixelSet(12, 23); myTOLED.pixelSet(1
3, 23); myTOLED.pixelSet(14, 23); myTOLED.pixelSet(15, 23); myTOLED.pixelSet(16, 23);
myTOLED.pixelSet(4, 24); myTOLED.pixelSet(5, 24); myTOLED.pixelSet(6, 24); myTOLED.p
ixelSet(7, 24); myTOLED.pixelSet(13, 24); myTOLED.pixelSet(14, 24); myTOLED.pixelSet(1
5, 24); myTOLED.pixelSet(16, 24);
myTOLED.pixelSet(3, 25); myTOLED.pixelSet(4, 25); myTOLED.pixelSet(5, 25); myTOLED.p
ixelSet(6, 25); myTOLED.pixelSet(14, 25); myTOLED.pixelSet(15, 25); myTOLED.pixelSet(1
6, 25); myTOLED.pixelSet(17, 25); myTOLED.pixelSet(18, 25);
myTOLED.pixelSet(2, 26); myTOLED.pixelSet(3, 26); myTOLED.pixelSet(4, 26); myTOLED.p
ixelSet(5, 26); myTOLED.pixelSet(6, 26); myTOLED.pixelSet(14, 26); myTOLED.pixelSet(1
5, 26); myTOLED.pixelSet(16, 26); myTOLED.pixelSet(17, 26); myTOLED.pixelSet(18, 26);
myTOLED.pixelSet(19, 26);
myTOLED.pixelSet(1, 27); myTOLED.pixelSet(2, 27); myTOLED.pixelSet(3, 27); myTOLED.p
ixelSet(4, 27); myTOLED.pixelSet(5, 27); myTOLED.pixelSet(6, 27); myTOLED.pixelSet(1
4, 27); myTOLED.pixelSet(15, 27); myTOLED.pixelSet(16, 27); myTOLED.pixelSet(17, 27);
myTOLED.pixelSet(18, 27); myTOLED.pixelSet(19, 27); myTOLED.pixelSet(20, 27);
}
```

```
void showPipe( void ){
    myTOLED.pixelSet(2, 2); myTOLED.pixelSet(3, 2); myTOLED.pixelSet(4, 2); myTOLED.pixelSet(5, 2); myTOLED.pixelSet(6, 2); myTOLED.pixelSet(7, 2); myTOLED.pixelSet(8, 2); myTOLED.pixelSet(9, 2); myTOLED.pixelSet(10, 2); myTOLED.pixelSet(11, 2); myTOLED.pixelSet(12, 2); myTOLED.pixelSet(13, 2); myTOLED.pixelSet(14, 2); myTOLED.pixelSet(15, 2); myTOLED.pixelSet(16, 2); myTOLED.pixelSet(17, 2); myTOLED.pixelSet(18, 2); myTOLED.pixelSet(19, 2); myTOLED.pixelSet(20, 2); myTOLED.pixelSet(21, 2); myTOLED.pixelSet(22, 2);
    myTOLED.pixelSet(2, 3); myTOLED.pixelSet(22, 3);
    myTOLED.pixelSet(2, 4); myTOLED.pixelSet(12, 4); myTOLED.pixelSet(13, 4); myTOLED.pixelSet(14, 4); myTOLED.pixelSet(15, 4); myTOLED.pixelSet(16, 4); myTOLED.pixelSet(22, 4);
    myTOLED.pixelSet(2, 5); myTOLED.pixelSet(12, 5); myTOLED.pixelSet(13, 5); myTOLED.pixelSet(14, 5); myTOLED.pixelSet(15, 5); myTOLED.pixelSet(16, 5); myTOLED.pixelSet(17, 5); myTOLED.pixelSet(22, 5);
    myTOLED.pixelSet(2, 6); myTOLED.pixelSet(4, 6); myTOLED.pixelSet(5, 6); myTOLED.pixelSet(9, 6); myTOLED.pixelSet(12, 6); myTOLED.pixelSet(13, 6); myTOLED.pixelSet(14, 6); myTOLED.pixelSet(15, 6); myTOLED.pixelSet(16, 6); myTOLED.pixelSet(22, 6);
    myTOLED.pixelSet(2, 7); myTOLED.pixelSet(12, 7); myTOLED.pixelSet(13, 7); myTOLED.pixelSet(14, 7); myTOLED.pixelSet(15, 7); myTOLED.pixelSet(16, 7); myTOLED.pixelSet(17, 7); myTOLED.pixelSet(22, 7);
    myTOLED.pixelSet(2, 8); myTOLED.pixelSet(12, 8); myTOLED.pixelSet(13, 8); myTOLED.pixelSet(14, 8); myTOLED.pixelSet(15, 8); myTOLED.pixelSet(16, 8); myTOLED.pixelSet(22, 8);
    myTOLED.pixelSet(2, 9); myTOLED.pixelSet(12, 9); myTOLED.pixelSet(13, 9); myTOLED.pixelSet(14, 9); myTOLED.pixelSet(15, 9); myTOLED.pixelSet(16, 9); myTOLED.pixelSet(22, 9);
    myTOLED.pixelSet(2, 10); myTOLED.pixelSet(12, 10); myTOLED.pixelSet(13, 10); myTOLED.pixelSet(14, 10); myTOLED.pixelSet(15, 10); myTOLED.pixelSet(16, 10); myTOLED.pixelSet(17, 10); myTOLED.pixelSet(22, 10);
    myTOLED.pixelSet(2, 11); myTOLED.pixelSet(4, 11); myTOLED.pixelSet(5, 11); myTOLED.pixelSet(9, 11); myTOLED.pixelSet(12, 11); myTOLED.pixelSet(13, 11); myTOLED.pixelSet(14, 11); myTOLED.pixelSet(15, 11); myTOLED.pixelSet(16, 11); myTOLED.pixelSet(17, 11); myTOLED.pixelSet(18, 11); myTOLED.pixelSet(22, 11);
    myTOLED.pixelSet(2, 12); myTOLED.pixelSet(3, 12); myTOLED.pixelSet(4, 12); myTOLED.pixelSet(5, 12); myTOLED.pixelSet(6, 12); myTOLED.pixelSet(7, 12); myTOLED.pixelSet(8, 12); myTOLED.pixelSet(9, 12); myTOLED.pixelSet(10, 12); myTOLED.pixelSet(11, 12); myTOLED.pixelSet(12, 12); myTOLED.pixelSet(13, 12); myTOLED.pixelSet(14, 12); myTOLED.pixelSet(15, 12); myTOLED.pixelSet(16, 12); myTOLED.pixelSet(17, 12); myTOLED.pixelSet(18, 12); myTOLED.pixelSet(19, 12); myTOLED.pixelSet(20, 12); myTOLED.pixelSet(21, 12); myTOLED.pixelSet(22, 12);
    myTOLED.pixelSet(3, 13); myTOLED.pixelSet(6, 13); myTOLED.pixelSet(7, 13); myTOLED.pixelSet(11, 13); myTOLED.pixelSet(13, 13); myTOLED.pixelSet(14, 13); myTOLED.pixelSet(15, 13); myTOLED.pixelSet(16, 13); myTOLED.pixelSet(17, 13); myTOLED.pixelSet(20, 13); myTOLED.pixelSet(21, 13);
    myTOLED.pixelSet(3, 14); myTOLED.pixelSet(6, 14); myTOLED.pixelSet(14, 14); myTOLED.pixelSet(20, 14); myTOLED.pixelSet(21, 14);
    myTOLED.pixelSet(3, 15); myTOLED.pixelSet(6, 15); myTOLED.pixelSet(14, 15); myTOLED.
```



```

pixelSet(15, 15); myTOLED.pixelSet(20, 15); myTOLED.pixelSet(21, 15);
  myTOLED.pixelSet(3, 16); myTOLED.pixelSet(6, 16); myTOLED.pixelSet(14, 16); myTOLED.
pixelSet(15, 16); myTOLED.pixelSet(20, 16); myTOLED.pixelSet(21, 16);
  myTOLED.pixelSet(3, 17); myTOLED.pixelSet(6, 17); myTOLED.pixelSet(14, 17); myTOLED.
pixelSet(15, 17); myTOLED.pixelSet(20, 17); myTOLED.pixelSet(21, 17);
  myTOLED.pixelSet(3, 18); myTOLED.pixelSet(6, 18); myTOLED.pixelSet(14, 18); myTOLED.
pixelSet(15, 18); myTOLED.pixelSet(20, 18); myTOLED.pixelSet(21, 18);
  myTOLED.pixelSet(3, 19); myTOLED.pixelSet(6, 19); myTOLED.pixelSet(14, 19); myTOLED.
pixelSet(20, 19); myTOLED.pixelSet(21, 19);
  myTOLED.pixelSet(3, 20); myTOLED.pixelSet(6, 20); myTOLED.pixelSet(14, 20); myTOLED.
pixelSet(15, 20); myTOLED.pixelSet(20, 20); myTOLED.pixelSet(21, 20);
  myTOLED.pixelSet(3, 21); myTOLED.pixelSet(6, 21); myTOLED.pixelSet(14, 21); myTOLED.
pixelSet(15, 21); myTOLED.pixelSet(20, 21); myTOLED.pixelSet(21, 21);
  myTOLED.pixelSet(3, 22); myTOLED.pixelSet(6, 22); myTOLED.pixelSet(14, 22); myTOLED.
pixelSet(15, 22); myTOLED.pixelSet(20, 22); myTOLED.pixelSet(21, 22);
  myTOLED.pixelSet(3, 23); myTOLED.pixelSet(6, 23); myTOLED.pixelSet(14, 23); myTOLED.
pixelSet(15, 23); myTOLED.pixelSet(20, 23); myTOLED.pixelSet(21, 23);
  myTOLED.pixelSet(3, 24); myTOLED.pixelSet(6, 24); myTOLED.pixelSet(14, 24); myTOLED.
pixelSet(20, 24); myTOLED.pixelSet(21, 24);
}

#else
void showMario( void ){
  Serial.println(F("Sorry, Uno and other low-memory microcontrollers may not be able t
o run the Mario code due to program memory size constraints."))
}

void showPipe( void ){
  Serial.println(F("Sorry, Uno and other low-memory microcontrollers may not be able t
o run the Mario code due to program memory size constraints."))
}
#endif

```

Once you've uploaded the code, you should see a bouncing Mario heading towards the finish line.



The 'Advanced Feature' referenced in this example is the ability to draw in relation to pre-defined 'windows.' Windows allow you to easily constrain drawing to a particular area or to group related objects. To highlight this fact we made a Mario character with a bunch of hard-coded `pixelSet(x,y)` calls but were able to move him simply by moving the active window and re-drawing. In that case, all the individual pixels that make up Mario were the related objects that we wanted to move together.

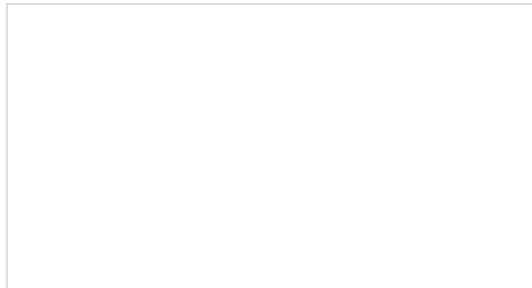
One feature not shown in this example is the built-in HyperDisplay text printing. The reason for this is that on, ESP32 boards the `<avr/pgmspace.h>` header is not supported so we can't use the default font. However, the printing functions should work on microcontrollers that both support `<avr/pgmspace.h>` and have enough RAM to be stable. Check out the HyperDisplay tutorial for more information on this.

Resources and Going Further

For more information on the Transparent Graphical OLED Breakout, check out some of the links here:

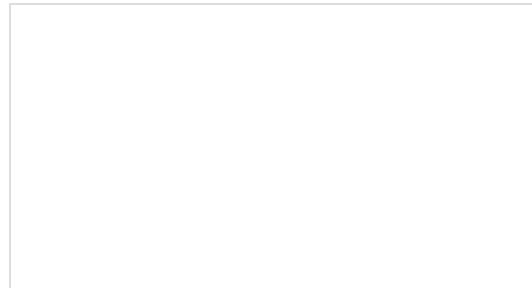
- Schematic (PDF)
- Eagle Files (ZIP)
- Arduino Libraries
 - HyperDisplay Graphics Library GitHub
 - SSD1309 HyperDisplay Mid Level Library GitHub
 - Transparent Graphical Bottom Level Library GitHub
- GitHub Repo
- SFE Product Showcase

Need some inspiration for your next project? Check out some of these other display related tutorials!



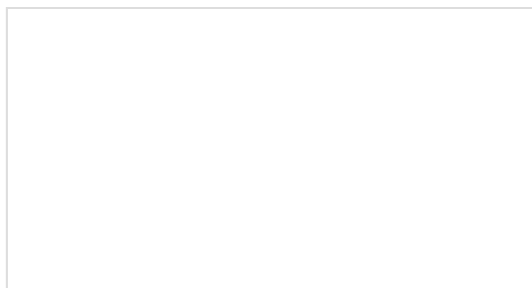
LED Light Bar Hookup

A quick overview of SparkFun's LED light bars, and some examples to show how to hook them up.



Graphic LCD Hookup Guide

How to add some flashy graphics to your project with a 84x48 monochrome graphic LCD.



RGB Panel Jumbotron



Qwiic Transparent OLED HUD Hookup

This tutorial will show you how to combine a webcam, a 32x32 RGB LED panel, and a Teensy 3.1 to stream video from the webcam, pixelate it, and display it on the LED panel - LIVE.

Guide

Learn how to connect that sweet transparent OLED.