



# macsbug

## ESP32-8048S043

ESP32-8048S043

2022.11.29

4.3 inch 800×480 IPS RGB Display with Capacitive Touch and ESP32-S3



プライバシーと Cookies:このサイトは Cookies を使用しています。このサイトの使用を続けると、Cookie の使用に同意したとみなされます。

Cookie の管理方法を含め、詳細についてはこちらをご覧ください:  
[Cookie ポリシー](#)

閉じて承認



ESP32で 最初の RGB Display Board です。

● RGB Display 方式は 内部にBufferがありません。

その為に ESP32のPSRAMを使用し低速になります。

速度の低下や 表示のチラツキが発生する場合があります。

WebRadio：高速表示が必要なWebRadioは受信を妨げ使用できません。

理由：Loopで 受信に必要な 25msec以内に処理する必要があります。

ESP32-8048S043 + PSRAM は 25msecを超えてしまいます。

よって ESP32-8048S043は 高速を必要とする物には使用できません。

● 画像がStackingする為に 私は使用していません。

理想の基板が無く 以下のボードを製作しました。

1. [Making a board of MAR4018 ESP32 S3](#)

2. [Making a board of MRB3973 ESP32 S3](#)

4種類あり、ESP32-8048S043C を解説します。

---

4.3 inch ESP32 module：販売：[Sunton store](#)

---

2614円	ESP32-4827S043N	480×272	TN Without touch
2823円	ESP32-4827S043R	480×272	TN Resistive touch

---

3322円	ESP32-8048S043R	800×480	TN Resistive touch
-------	-----------------	---------	--------------------

3790円	ESP32-8048S043C	800×480	IPS Capacitive touch
-------	-----------------	---------	----------------------

仕様：

サイズ：12.3 x 7.4 cm

Chip Type：ESP32-S3

ESP32-S3-WROOM-1-**N16R8**

ROM：384KB, SRAM：512KB, **SPI Flash 16MB, PSRAM (OctalSPI)：8MB**

TELEC No.：201-220052：技適対応品

Display：

仕様：[JC8048B070N](#)

販売店：[Sunton Store](#)：1971円。

販売店：[4.3 inch 800×480 IPS RGP Display with Capacitive Touch](#)：\$27.99

資料：[ESP32-4827S043 download link](#)

資料：[4.3inch Capacitive Touch LCD](#)

参考：[ESP32-2432S028](#)

参考：[ESP32-3248S035](#)

メモ：LovyanGFX、Arduino GFX 共に複雑な画像を表示しますと横の同期が崩れます。

広告



Gentle Care for Your Hair

**Discover the soothing shampoo for sensitive scalps with La Roche Posay's Thermal Spring Water.**

[Shop Now](#)

## 開発環境 1 : Arduino IDE 設定

```
1 // HARD : ESP32_8048S043C ( ESP32-S3 )
2 // Display : 4.3" 800x480 IPS RGB LCD Touch GT911
3 // Dev environment : Arduino IDE 1.8.19
4 // Board Manager : arduino-esp32 2.0.5
5 // Board : "ESP32S3 Dev Module"
6 // Upload Speed : "921600"
7 // USB Mode : "Hardware CDC and JTAG"
8 // USB CDC On Boot : "Disable"
9 // USB Firmware MSC On Boot : "Disable"
10 // USB DFU On Boot : "Disable"
11 // Upload Mode : "UART0 / Hardware CDC"
12 // CPU Frequency : "240MHz (WiFi/BT)"
13 // Flash Mode : "QIO 120MHz"
14 // Flash Size : "16MB (128Mb)"
15 // Partition Scheme : "16MB Flash (2MB APP/12.5MB FATFS)"
```

```
16 // Core Degug Level : "None"
17 // PSRAM : "OPI PSRAM"
18 // Arduino Runs On : "Core 1"
19 // Events Run On : "Core 1"
20 // Erase All Flash Before Sketch Upload : "Disable"
21 // Pord : "dev/cu.wchusbserial14240"
```

開発環境2 : [LovyanGFX](#)。尚、暫定設定です。

謝辞 : らびやん氏に ESP32\_8048S043Cを送付し

RGB Driver (Bus\_RGB,Panel\_RGB)を作成して頂きました。

設定 : 記事の下に示します。

機能 : Graphics命令が多彩、綺麗な漢字も使用できます。

Down Load : DL後 pdf(↓.pdf)を削除しzipを解凍します。

 Down Load : [PDQgraphicstest\\_LGFX.zip](#)

 Down Load : [MovingCircles\\_8043S043\\_LGFX.zip](#)

開発環境3 : [Arduino\\_GFX](#) ( [GFX Library for Arduino](#) ) 。

陳亮氏が [Adafruit-GFX-Library](#) を改良したものです。

gfx-> の書式に慣れる必要があります。

LovyanGFXで利用できる Graphics命令が無し(setAddrWindow, pushImage)

綺麗な漢字表示のLibrary無し。

Graphics命令の不足は 移植等に難があります。

設定 : 記事の下に示します。

Down Load : DL後 pdf(↓.pdf)を削除しzipを解凍します。

 Down Load : [PDQgraphicstest\\_AGFX.zip](#)

 Down Load : [Clock\\_AGFX.zip](#)

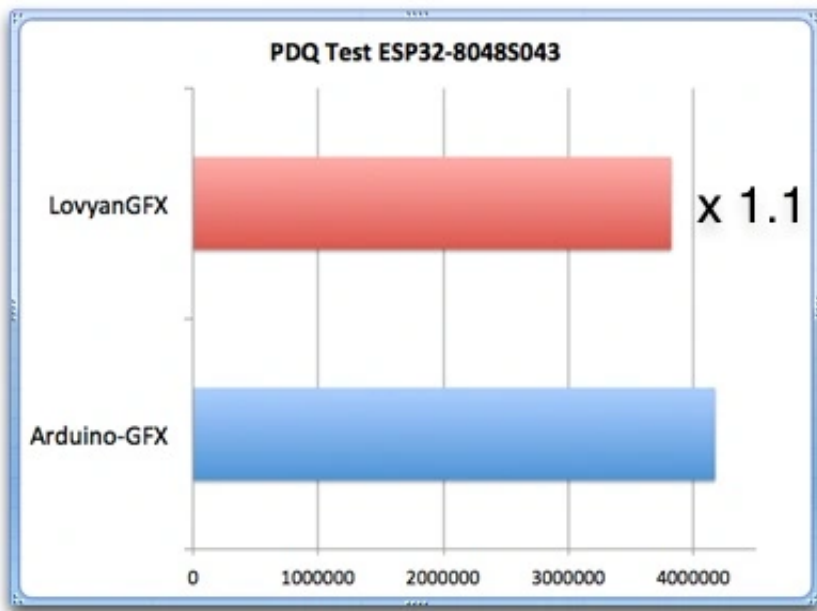
---

速度の比較 : LovyanGFX と Arduino GFX

ESP32-4827S043C, 800×480

RGB Display : バッファに ESP32のPSRAMを使用する為に速度は遅くなります。

setAddrWindow命令を使用できない為 Arduino GFXのPDQgraphicstestを使用。

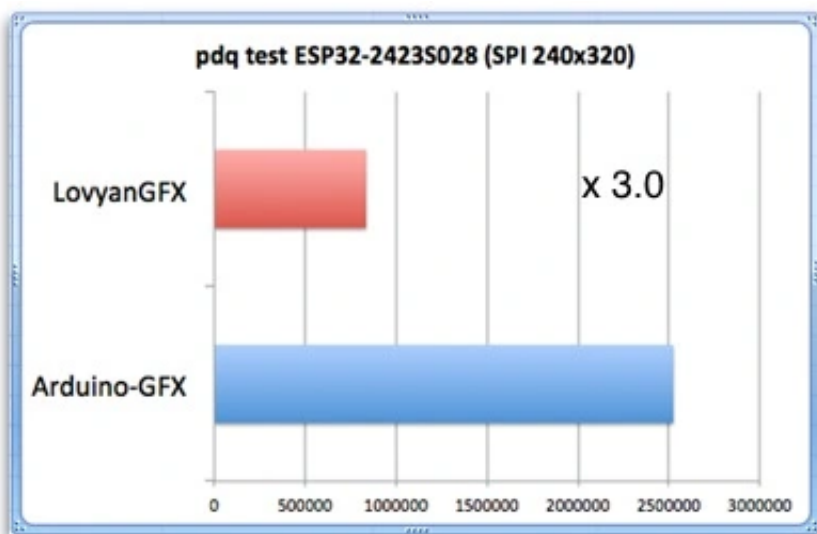


他の ボードに於ける Arduino GFX と LovyanGFX : 速度の比較  
LovyanGFX は 3 倍 速い。

2.8 inch [ESP32-2432S028](#) と PDQ graphics test を実施。

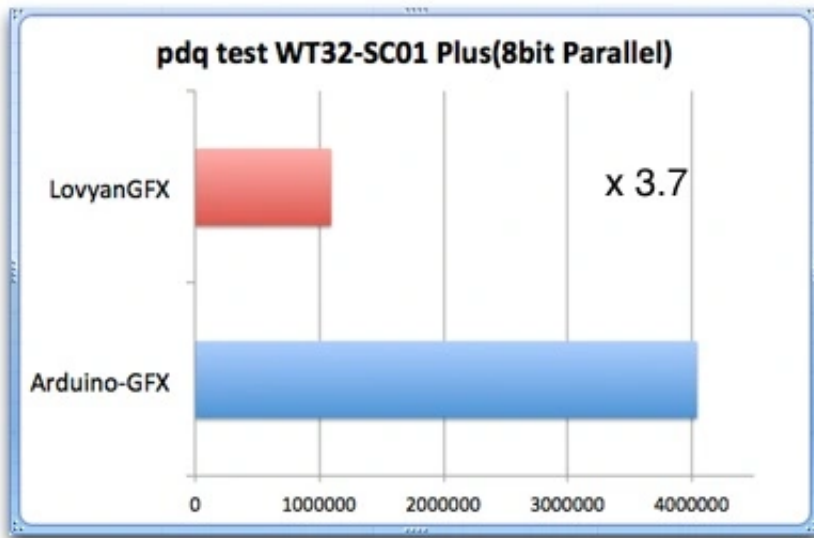
LovyanGFX の設定は [ESP32-2432S028](#) に記載。

Arduino GFX の設定は examples / [PDQgraphicstest](#) を使用。



That Project : [WT32\\_SC01\\_Plus \(8bit Parallel\)](#) の速度比較。

LovyanGFX は 3.7倍 速い。



部品レイアウト：配置、信号、ピンは色分けしてあります。

ESP32-S3-WROOM-1-[N16R8](#) , with TELEC ( [201-220052](#) )

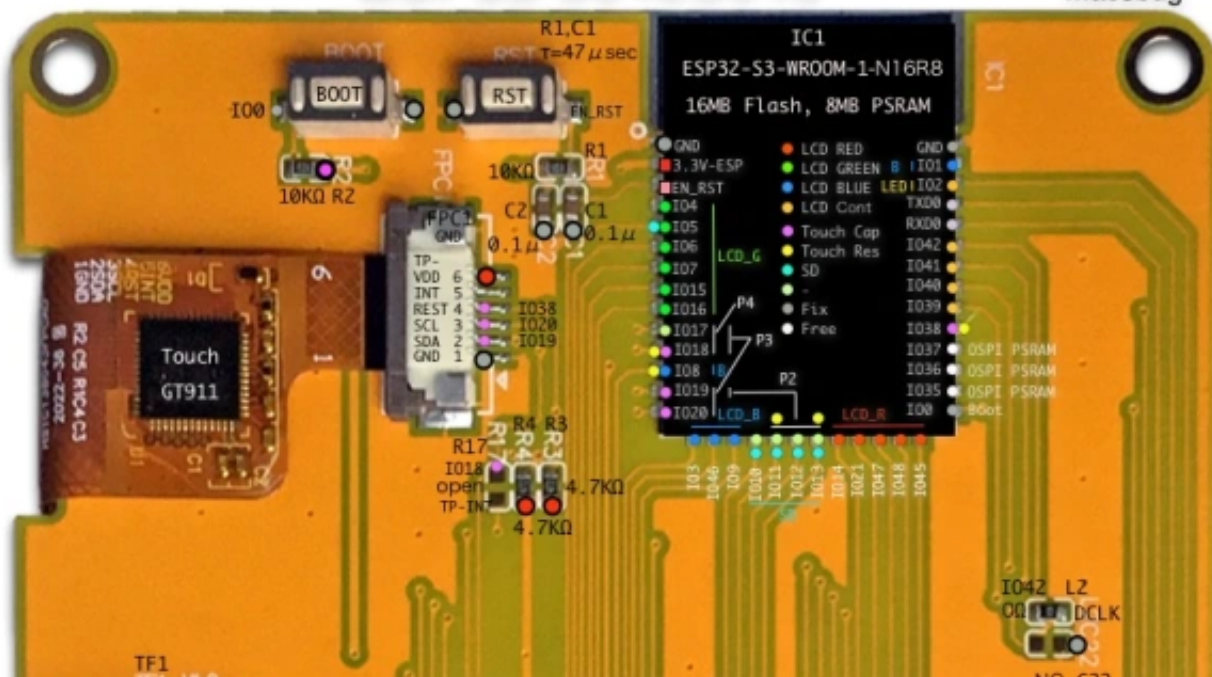




ESP32-8048S043 IPS Capacitive Touch

## ESP32-8048S043

2022.10.18  
macsbug



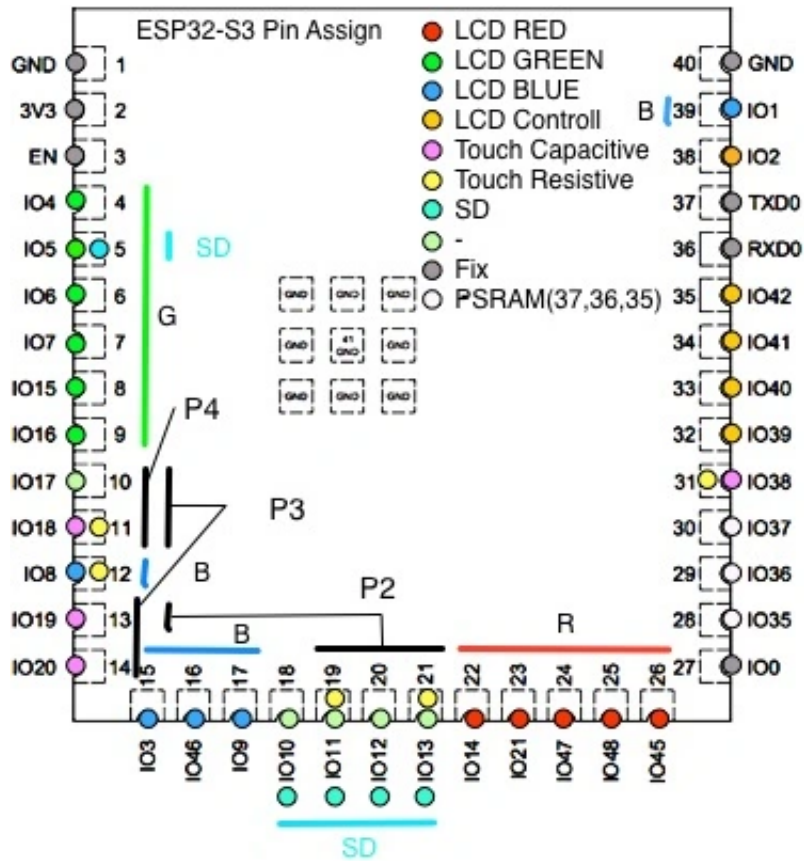




以下のPDF図面は 拡大表示が可能と文字検索が可能です。配線や設計、改造時に便利です。

[4280S043\\_Layout](#)

ESP32-S3 Pin Assign :



GPIO Pin Assign : ESP32-8048S043 ( Resistive or Capacitive touch )

薄緑は GPIO。濃い緑は GPIO共通

Green	Green
GPIO	共通 (Common)

Open Pad ( オープンになっているパッド )

3.3VESP	3.3VTFT	LED	INT	U1	U1	U1	U1	DCLK
JP1	JP2	R10	R17	C3	C4	C5	C6	C22

Extended IO ( P1, P2, P3, P4 )

共通 (Common)の SD と Touch に注意

<b>P1</b>	1	2	3	4
Name	VIN	TXD2	RXD2	GND

<b>P2</b>	1	2	3	4
GPIO	19	11	12	13
GPIO	–	SD	SD	SD

<b>P3</b>	1	2	3	4
GPIO	17	18	19	20
GPIO	–	–	Touch	Touch

<b>P4</b>	1	2	3	4
GPIO	GND	3.3VESP	17	18

GPIO		–		–		–		–			
LCD1											
Touch		X1		Y1		X2		Y2			
U1		X+		Y+		X-		Y-			
LED		VDD		GND		VLED+		VLED-			
LED		3.3V		GND		LEDA		LEDK			
GPIO		–		–		2		2			
LCD1		DCLK		DISP		HSYNC		VSYNC		DE	
GPIO		42		3v3,R16		39		41		40	
RGB_R	R0	R1	R2	R3	R4	R5	R6	R7			
GPIO	GND	GND	GND	45	48	47	21	14			
cfg_pin	–	–	–	d11	d12	d13	d14	d15			
RGB_G	G0	G1	G2	G3	G4	G5	G6	G7			
GPIO	GND	GND	5	6	7	15	16	4			

cfg_pin	–	–	d5	d6	d7	d8	d9	d10
RGB_B	B0	B1	B2	B3	B4	B5	B6	B7
GPIO	GND	GND	GND	8	3	46	9	1
cfg_pin	–	–	–	d0	d1	d2	d3	d4

#### FPC1 : Touch I2C Capasitive : GT911

Pin No.	1	2	3	4	5	6
3.3V	–	R4	R3	–	R17( $\infty$ )	–
Name	GND	SDA	SCL	REST	INT	VDD
GPIO	–	19	20	38	(18)	–

#### Touch SPI Resistive : XPT2046

3.3V	R5	–	–	–	–	–	–
Touch	IRQ	CS	DIN	–	CLK	–	OUT
GPIO	18	38	11	–	12	–	13

SD , TF1										
Pin No	1	2	3	4	5	6	7	8	9	10
Name	DAT2	DAT3	CMD	VDD	CLX	VSS	DAT0	DAT1	CD	GND
3.3V	R15	RN1	RN1	–	–	–	RN1	RN1	–	–
TF1	3V3	CS	MOSI	3V3	CLK	GND	MISO	3.3V	open	GND
GPIO	–	10	11	–	12	–	13	5	–	–

lovyanGFX : LCD1															
cfg.pin	d0	d1	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
RGB	B0	B1	B2	B3	B4	G0	G1	G2	G3	G4	R0	R1	R2	R3	R4
cfg.pin	henable				vsync				hsync				pclk		bl
GPIO	40				41				39				42		2

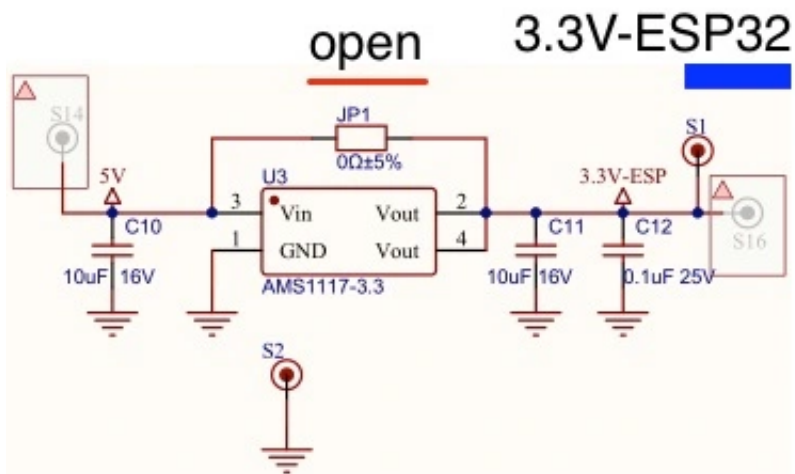
分析：

1. ESP32-S3 未使用端子

GOIO 35, 36, 37 は PSRAM用で GPIOは 使用できません。

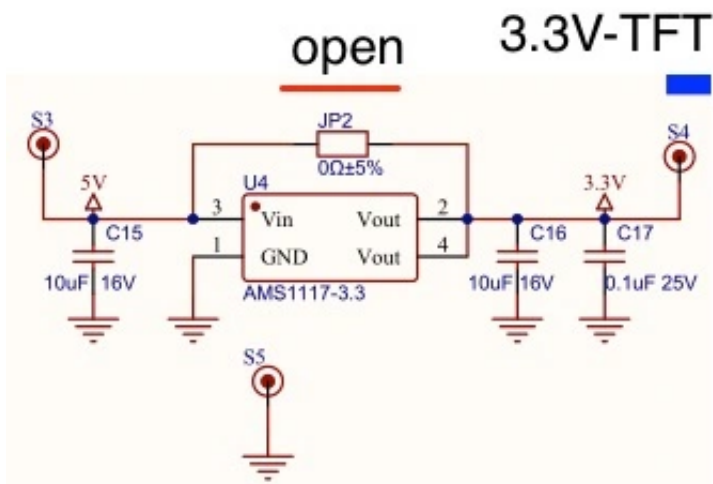
2. 3.3V-ESP32 Power Supply

JP1 : open



3. 3.3V-TFT Power Supply

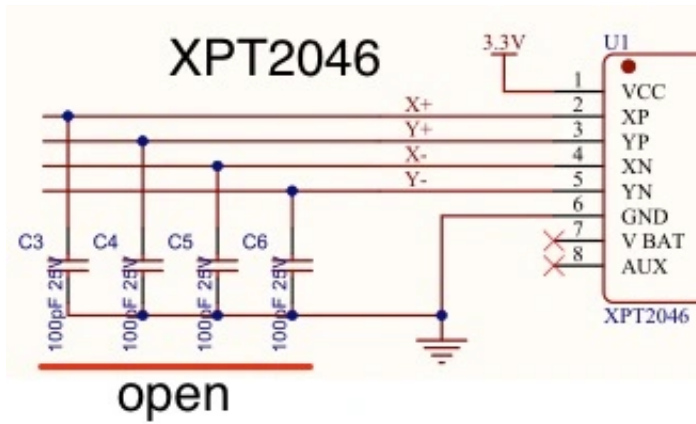
JP2 は open



4. XPT2046 Touch ( Resistive Touch )

C3, C4, C5, C6 は open





5. LCD : WAVESHARE : SKU: 16249

[4.3inch Capacitive Touch LCD, 800×480](#)

[4.3inch Capacitive Touch LCD](#)

[User Manual](#)

[Schematic](#)

[GT911](#)

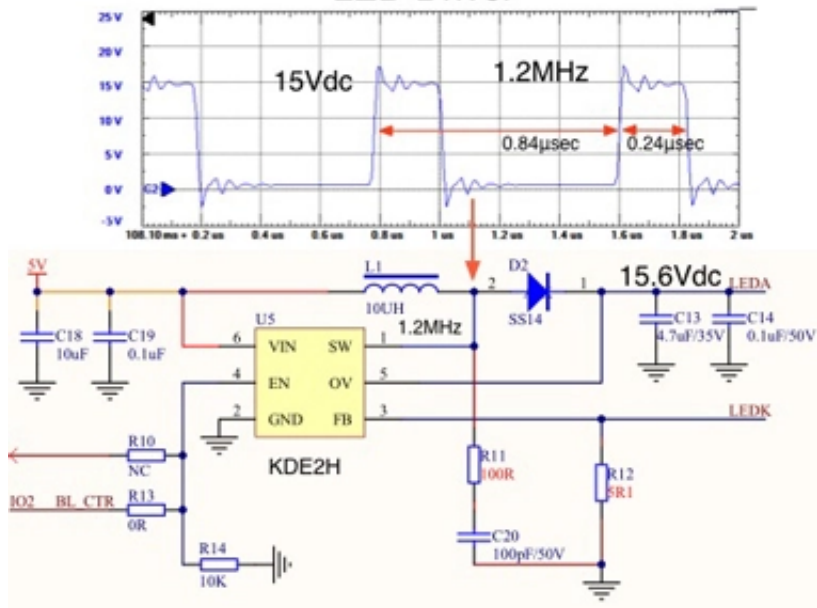
6. LED Driver :

U5 dv-dc Converter : Code=KDE2H : [LT1930](#) ?。

Frequency = 1.2MHz。 Peak +15Vdc。

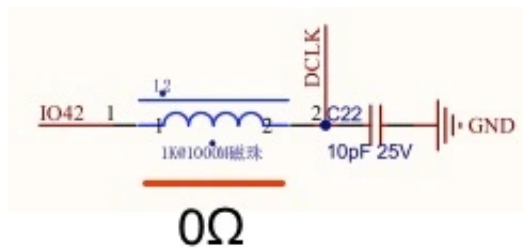
LEDA = +15.6Vdc。

## LED Driver



### 7. LCD L2

L2 は 0Ω



### 8. Capacitive Touch : GT911

Address は 0x14 ( 0x5D ) です。

### 9. ESP32-S3 : EN 時定数

R1(10KΩ) と C1(0.1μ) で 時定数 τ(タウ) は 47μsec です。

47μsecは ESP32 の起動には 仕様を満たした値です。

## LovyanGFX：設定

Library：[lgfx\\_user/LGFX\\_ESP32S3\\_RGB\\_ESP32-8048S043.h](#) を使用します。

Libraryは 暫定処置です。Touchを追加しています。

```
1 //=====
2 // https://github.com/lovyan03/LovyanGFX/tree/develop/src/lgfx
3 // Panel_RGB , Bus_RGB
4 //=====
5 #define LGFX_USE_V1
6 #include <LovyanGFX.hpp>
7 #include <lgfx/v1/platforms/esp32s3/Panel_RGB.hpp>
8 #include <lgfx/v1/platforms/esp32s3/Bus_RGB.hpp>
9
10 class LGFX : public lgfx::LGFX_Device{
11     lgfx::Bus_RGB      _bus_instance;
12     lgfx::Panel_RGB    _panel_instance;
13     lgfx::Light_PWM    _light_instance;
14     lgfx::Touch_GT911  _touch_instance;
15
16 public:LGFX(void){
17     auto cfg = _bus_instance.config();
18     cfg.pin_d0 = GPIO_NUM_8; // B0
19     cfg.pin_d1 = GPIO_NUM_3; // B1
20     cfg.pin_d2 = GPIO_NUM_46; // B2
21     cfg.pin_d3 = GPIO_NUM_9; // B3
22     cfg.pin_d4 = GPIO_NUM_1; // B4
23     cfg.pin_d5 = GPIO_NUM_5; // G0
24     cfg.pin_d6 = GPIO_NUM_6; // G1
25     cfg.pin_d7 = GPIO_NUM_7; // G2
26     cfg.pin_d8 = GPIO_NUM_15; // G3
27     cfg.pin_d9 = GPIO_NUM_16; // G4
28     cfg.pin_d10 = GPIO_NUM_4; // G5
29     cfg.pin_d11 = GPIO_NUM_45; // R0
30     cfg.pin_d12 = GPIO_NUM_48; // R1
31     cfg.pin_d13 = GPIO_NUM_47; // R2
32     cfg.pin_d14 = GPIO_NUM_21; // R3
33     cfg.pin_d15 = GPIO_NUM_14; // R4
34     cfg.pin_henable = GPIO_NUM_40;
35     cfg.pin_vsync = GPIO_NUM_41;
36     cfg.pin_hsync = GPIO_NUM_39;
37     cfg.pin_pclk = GPIO_NUM_42;
38     cfg.freq_write = 16000000;
39     cfg.hsync_polarity = 0;
40     cfg.hsync_front_porch = 8;
41     cfg.hsync_pulse_width = 4;
42     cfg.hsync_back_porch = 16;
43     cfg.vsync_polarity = 0;
```

```

44     cfg.vsync_front_porch = 4;
45     cfg.vsync_pulse_width = 4;
46     cfg.vsync_back_porch  = 4;
47     cfg.pclk_idle_high    = 1;
48     _bus_instance.config(cfg);
49     _panel_instance.setBus(&_bus_instance);
50
51     {
52     auto cfg = _panel_instance.config();
53     cfg.memory_width  = 800;
54     cfg.memory_height = 480;
55     cfg.panel_width   = 800;
56     cfg.panel_height  = 480;
57     cfg.offset_x      = 0;
58     cfg.offset_y      = 0;
59     _panel_instance.config(cfg);
60     }
61
62     {
63     auto cfg = _panel_instance.config_detail();
64     cfg.use_psram = 1;
65     _panel_instance.config_detail(cfg);
66     }
67
68     { auto cfg = _light_instance.config();
69     cfg.pin_bl = GPIO_NUM_2;
70     _light_instance.config(cfg);
71     }
72     _panel_instance.light(&_light_instance);
73
74     { auto cfg = _touch_instance.config();
75     cfg.x_min    = 0;          // タッチスクリーンから得られる最小のX値
76     cfg.x_max    = 800;        // タッチスクリーンから得られる最大のX値
77     cfg.y_min    = 0;          // タッチスクリーンから得られる最小のY値
78     cfg.y_max    = 480;        // タッチスクリーンから得られる最大のY値
79     cfg.pin_int   = 18;        // INTが接続されているピン番号 18
80     cfg.bus_shared = false;    // 画面と共通のバスを使用している場合 t:
81     cfg.offset_rotation = 0;  // 表示とタッチの向きの調整 0~7の値で設
82     // I2C接続
83     cfg.i2c_port  = 1;          // 使用するI2Cを選択 (0 or 1)
84     cfg.pin_sda   = 19;        // SDAが接続されているピン番号
85     cfg.pin_scl   = 20;        // SCLが接続されているピン番号
86     cfg.freq      = 400000;    // I2Cクロックを設定
87     cfg.i2c_addr  = 0x14;      // I2Cデバイスアドレス番号 or 0x5D ,
88     _touch_instance.config(cfg);
89     _panel_instance.setTouch(&_touch_instance); //タッチスクリーン;
90     }
91
92     setPanel(&_panel_instance); // 使用するパネルをセットします。

```

```

93     }
94 };
95 LGFX tft; // 準備したクラスのインスタンスを作成します。
96 //=====

```

---

## Arduino GFX：設定

```

1 // ESP32_4827S043
2 #include <Arduino_GFX_Library.h>
3 #define TFT_BL 2 // LED K
4 // CS, SCK, SDA, DE, VSYNC, HSYNC, PCLK,
5 // R0, R1, R2, R3, R4, G0, G1, G2, G3, G4, G5, B0, B1, B2, B3, B4
6 Arduino_ESP32_RGBPanel *bus = new Arduino_ESP32_RGBPanel(
7     GFX_NOT_DEFINED, GFX_NOT_DEFINED, GFX_NOT_DEFINED, 40, 41, 39, 42,
8     45, 48, 47, 21, 14, 5, 6, 7, 15, 16, 4, 8, 3, 46, 9, 1
9 );
10
11 // width , hsync_polarity, hsync_front_porch, hsync_pulse_width,
12 // height, vsync_polarity, vsync_front_porch, vsync_pulse_width,
13 // pclk_active_neg, prefer_speed, true, auto_flush
14 Arduino_RPi_DPI_RGBPanel *gfx = new Arduino_RPi_DPI_RGBPanel(
15     bus, 800, 0, 8, 4, 8, 480, 0, 8, 4, 8, 1, 16000000, true);
16
17 void setup() {
18     pinMode(TFT_BL, OUTPUT); digitalWrite(TFT_BL, HIGH);

```

---

## 感想：

4.3 inch 800×480 は ゆとりある解像度、多彩で自由なデザインが可能です。

7.0 inch 800×480 の使用経験から 4.3" は 小型で持ちやすく見やすいです。

Capacitive Touch は Resistive Touch よりも反応と操作性が良いです。

広告

Search for

### 1. **Vitamins to Improve Memory**




---

**Kitchen Cabinet Colors for**

## 2. **Kitchen Cabinet Colors for 2023** >

## 3. **Best Stocks to Invest in 2023** >



この広告についてレポート

問題点：

一般的なLCDはDriver ICがあり内部にメモリーが確保されています。

今回のRGB Displayは Driver ICはありません。他のRGBシリーズも同様です。

表示の為にメモリー PSRAM(OPI PSRAM)の容量を使用します。

PSRAM : OPI PSRAM の設定が必要で 結果的に速度の向上ができません。

つまり設計者は安価を目指しRGBを採用したのですが高速表示はできませんでした。

発売時に設計者にお聞きするとDisplay Librayは用意していません。

これでは使い物にならないです。よって この LCDボードは 今後使用する事はないでしょう。

その為に 800×480 サイズのDisplay及び基板の新たな設計を想像しています。

ESP32-4827S043N, ESP32-4827S043R : 480×272

低価格の構成としては 良いですが 480×320 より縦が少なく未購入。

ESP32-8048S043R : 800×480

解像度は 良いですが Resistive Touch で未購入。

ESP32-8048S043C : 800×480

理想的なサイズと解像度、Capasitive Touch の操作性で購入。

---

2023.06.24

hello @aivaras

This is an example of broken image display.

It's a pdf file. Delete the pdf and unzip the zip.

Enter WiFi SSID and PASS.

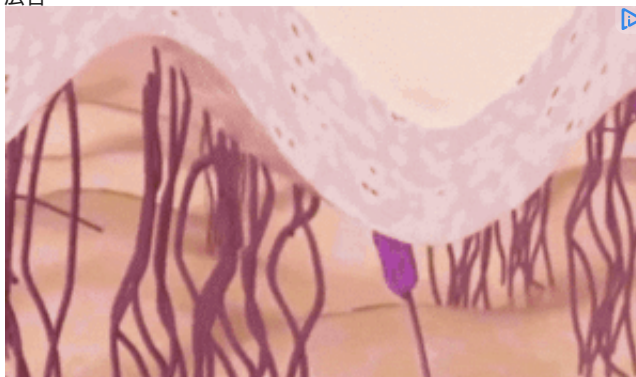
[Web\\_Radio\\_8048S043C\\_LGFX.zip](#)

.

.

4.3 inch , 800×480 , IPS , ESP32-S3 , GT911 ,

広告



**I Stopped Retinol and Used One Household Item that Removes Wrinkles In No Time!**

この広告についてレポート

11月 29, 2022

 12件のフィードバック

---

« 前へ

次へ »

コメントを残す

---



コメントをどうぞ

コメント



12月 26, 2022 12:52 pm に genvex より

I am looking forward to seeing you use lovyangFX to driver the lvgl on this screen.

excellent works .

[返信](#)



12月 31, 2022 9:07 am に macsbug より

Hello genvex

Thank you for visiting our site.

;

The latest version of LVGL is 8.3.4.

This is incomplete.

Therefore, we are preparing for LVGL8.3.

Previously done with LVGL8.2.

Compatibility with LVGL8.2 and LVGL8.3 is low.

You should start with the basics of LVGL8.3.

It is good to use the LVGL8.3 sample.

It is currently in preparation.



6月 24, 2023 4:17 pm に [macsbug](#) より

Thank you for visiting our site.

situation:

WiFi connects in other programs but

No connection in this sketch.

Touch operation is also not possible.

I'm looking into it, but I don't understand.

Problem with 8048S043C.

The display speed is slow and cannot be displayed at high speed.

Example: Added a sketch at the bottom below. 2023.06.24

## ESP32-8048S043

ESP32-  
8048S043  
2022.11.29 4.3 inch 800×480 IPS  
RGB Display with Capacitive Touch  
and ... 続きを読む



macsbug

 12 

A normal LCD display has a built-in driver and memory.

8048S043C uses R.G.B display.

RGB display has no memory.

For that purpose, we use PSRAM memory of ESP32.  
The display speed will be slow due to the use of PSRAM.  
The display is disturbed as if the horizontal synchronization is off.  
For this reason I am not using his 8048S043C.

The article below has LVGL demo running on Arduino\_GFX\_Library.

<https://www.makerfabs.com/sunton-esp32-s3-4-3-inch-ips-with-touch.html>

[https://wiki.makerfabs.com/Sunton\\_ESP32\\_S3\\_4.3\\_inch\\_800x400\\_IPS\\_with\\_Touch.html](https://wiki.makerfabs.com/Sunton_ESP32_S3_4.3_inch_800x400_IPS_with_Touch.html)

<https://drive.google.com/drive/folders/1vtLdvFJK6pgmHYFAGXMqaPkhuZtFDwjY>

6月 24, 2023 4:26 pm に macsbug より

Driver for 8048S043C was developed by Mr. Lovyan03.

<https://github.com/lovyano3/LovyanGFX>

As a result, the image is unstable for sketches that require display speed.  
I am making the board for the article below without using his 8048S043C.

MAR4018 and ESP32 S3

## MAR4018 and ESP32 S3

MAR4018 and ESP32  
S3

2022.12.31 SKU MAR4018 Display は  
4inch size , 解像度は 800×480です。  
Driver ... 続きを読む



macsbug



0



## MRB3973 and ESP32 S3

## MRB3973 and ESP32 S3

MRB3973 and ESP32  
S3

2023.01.11      SKU MRB3973  
LCD Display のサイズは 3.97 inch で  
す。 解像度は 800×480 ... 続きを読む



macsbug



2



[返信](#)

6月 21, 2023 1:29 am に [aivaras](#) より

Here's working code for the 8048S043 with LovyanGFX and LVGL but the

display glitches when calling WiFi.begin(). Any ideas?

<https://pastebin.com/YQ1YVJJH>

↳ 返信

5月 7, 2024 7:58 pm に wombat より

お世話になっております。微気圧計をDPS310とこの基板で作ろうとしています。  
ESP32-3248S035では簡単にi2cで接続し、データが取得できているのですが、この  
基板だと、どうしてもi2cのピン19番と20番につないでも応答がFAIL! ret = -2で返っ  
てきます。ピン番号に誤りか、設定方法になにか特別な処理が必要なののでしょうか？ど  
うぞよろしくお願いします。

↳ 返信

5月 7, 2024 9:09 pm に macsbug より

ESP32-3248S035での実績は役に立ちます。

> この基板だと

ESP32-8048S043 は ESP32-8048S043R と ESP32-8048S043C の 2 種類  
があります。

どちらのお話でしょうか？

ESP32-8048S043C は touch で GPIO\_19, GPIO\_20 を使用しています。

ESP32-8048S043R は GPIO\_19, GPIO\_20 を使用できそうです。

ただし、ESP32-S3-WROOM-1 は Arduino IDE の設定により

GPIO\_19, GPIO\_20 の動作が複数あります。

GPIO\_19, GPIO\_20 は USB にもなる GPIOです。

> ピン19番と20番

番の意味がわかりませんが、GPIO\_19, GPIO\_20 でしょうか？

GPIO\_19.GPIO\_20 は USB にもなる GPIOです。

Arduino IDE の設定により 動作が変化します。

多数の組み合わせがあり、解りづらいです。

> 応答がFAIL! ret = -2

1. 恐らく、GPIO\_19.GPIO\_20 の設定を行っていない為と想像します。

2. GPIOを I2C で使用する場合、pullup抵抗が必要な場合があります。

I2C で 4.7Kや10KΩで pullupさrている回路を参照ください。

この基板の設計は GPIO\_19, GPIO\_20 で USB接続できるのですが

UART IC CH340C を使用し 書き込みを行っています。

設計者は ESP32-S3-WROOM-1 の GPIO\_19, GPIO\_20 を

USBとして使用できる事を 理解していない様です。

 返信

---

5月 7, 2024 9:21 pm に yossiokamoto より

早速の返信ありがとうございます。 Aliの購入歴だとCだと思われます。

Sunton Storeからです。 19番20番はご指摘のようにGPIO\_19.GPIO\_20

です。 これi2cの設定にするにはなにか必要なのでしょうか？ ネットには

ESP32-S3単体の情報はかなりあるのですが、 この基板の情報はこちらの

サイトくらいなので、 とても困っています。 defineとでほかのピンも試し

てみたのですが、

```
define PIN_SDA 19 define PIN_SCL 20
```

```
void setup()
```

```
{
```

```
Wire.setPins(PIN_SDA, PIN_SCL);
```

でピン番号を替えてみたりもしたのですが、だめなようでした（全部のピンは試していません）。元々外部に出力されているピンの数が限られているので、そのあたりも情報を探しているのですが見つかることができませんでした。よろしくお願いします。



5月 7, 2024 10:21 pm に macsbug より

ブログの画像と同じ基板で ESP32-8048S043C としてお話しします。  
C は Capacitive touch で GPIO19, 20 は Touch I2C Capacitive : GT911 で使用されています。

空いている GPIOは 10,11,12,13 です。

ただし SDと共通ですので SDは使用しない事が条件です。

この GPIOは全て 10KΩで PULLUP されています。

接続するI2Cにもよりますが、この抵抗値なら動作条件は良さそうです。

接続するI2C両方の抵抗値を確認した方が良いです。

抵抗値が正しくないと波形が崩れて通信できません。

I2Cの制御：

ESP32のWire.begin(); は以下のパラメーターです。

```
bool begin(int sda=-1, int scl=-1, uint32_t frequency=0);
```

ですので、

例として Wire.begin(10,11); とかで、良いのではと考えています。

ESP32-8048S043Cは 空きのGPOIが少ないですが なんとか使いこなしたいですね。

5月 7, 2024 10:36 pm に yossiokamoto より

的確なご回答ありがとうございます。明日以降その線で頑張ってみます。また結果は数



日中にご報告します。今日は遅くまで色々とお教えいただきありがとうございました。

[返信](#)

5月 12, 2024 7:32 am に yossiokamoto より

先日のご回答ありがとうございます。結果からいうと

```
Wire.begin(12,13);
```

でデータの取得はできました。

ただこれを従来の3.5インチ基板用のプログラムに組み込むとなぜかデータ取得できません。SD関係の命令は//しているのですが、ただ私の目的としてはSD使用はマストなので、この基板は使用には適さないようです。あとご教示いただいた10番ピンは基板図には18と書かれているのですが、このことでしょうか？よろしくお願いします。

[返信](#)

5月 12, 2024 12:43 pm に yossiokamoto より

追伸です。今IO19とIO20でデータ取得できました。この線で少し頑張ってみます。

[返信](#)

検索

最近の投稿

[DRUM with CYD](#)  
[DRUM with M5Core2](#)  
[DRUM with M5Stack](#)  
[Power up the ESP32-2432S032C](#)  
[SPRESENSE Animation](#)

---

## アーカイブ

[2025年2月](#)  
[2025年1月](#)  
[2024年12月](#)  
[2024年11月](#)  
[2024年10月](#)  
[2024年9月](#)  
[2024年8月](#)  
[2024年7月](#)  
[2024年6月](#)  
[2024年5月](#)  
[2024年4月](#)  
[2024年3月](#)  
[2024年2月](#)  
[2024年1月](#)  
[2023年12月](#)  
[2023年11月](#)  
[2023年10月](#)  
[2023年9月](#)  
[2023年8月](#)  
[2023年7月](#)  
[2023年6月](#)  
[2023年5月](#)  
[2023年4月](#)  
[2023年3月](#)  
[2023年2月](#)  
[2023年1月](#)  
[2022年12月](#)  
[2022年11月](#)  
[2022年10月](#)

2022年8月  
2022年7月  
2022年6月  
2022年5月  
2022年4月  
2022年3月  
2022年2月  
2022年1月  
2021年12月  
2021年11月  
2021年10月  
2021年9月  
2021年7月  
2021年6月  
2021年5月  
2021年4月  
2021年2月  
2021年1月  
2020年12月  
2020年7月  
2020年6月  
2020年5月  
2020年4月  
2020年3月  
2020年2月  
2020年1月  
2019年12月  
2019年9月  
2019年8月  
2019年7月  
2019年6月  
2019年5月  
2019年4月  
2019年3月  
2019年2月  
2019年1月  
2018年12月  
2018年10月

2018年9月  
2018年8月  
2018年7月  
2018年6月  
2018年5月  
2018年4月  
2018年3月  
2018年1月  
2017年12月  
2017年11月  
2017年9月  
2017年8月  
2017年7月  
2017年6月  
2017年5月  
2017年4月  
2017年3月  
2017年2月  
2017年1月  
2016年12月  
2016年11月  
2016年10月  
2016年9月  
2016年8月  
2016年7月  
2016年6月  
2016年5月  
2016年4月  
2016年3月  
2016年2月  
2016年1月  
2015年12月  
2015年11月  
2015年10月  
2015年9月  
2015年8月  
2015年7月  
2015年6月

[2015年5月](#)  
[2015年4月](#)  
[2015年3月](#)  
[2015年2月](#)  
[2015年1月](#)  
[2014年12月](#)  
[2014年11月](#)  
[2014年10月](#)  
[2014年9月](#)  
[2012年1月](#)  
[2011年12月](#)

---

## カテゴリー

[Apple](#)  
[Arduino](#)  
[EAGLE](#)  
[ESP32](#)  
[ESP8266](#)  
[Gadget](#)  
[Kinoma Create](#)  
[Knowledge](#)  
[M5STACK](#)  
[Make](#)  
[New Infomation](#)  
[PcDuino](#)  
[Processing](#)  
[PSoC](#)  
[Pythonista](#)  
[Radio](#)  
[SPRESENSE](#)  
[SwiftIO](#)  
[Travel](#)  
[Uncategorized](#)  
[Wio Terminal](#)

---

## メタ情報

[登録](#)

[ログイン](#)

[投稿フィード](#)

[コメントフィード](#)

[WordPress.com](#)

[フルサイズ表示](#)

[WordPress.com Blog.](#)