

480x320ドット 3.5インチ LCD301-TC7796U-25SP

容量型タッチパネル搭載
カラーTFT液晶
コントローラ：ST7796U

表示例



☆色鮮やかな表示の得られる480x320ドットの
小型カラーTFT液晶ディスプレイ
(外形寸法：約98x56mm)

☆容量型タッチパネル搭載(コントローラ：FT6336U)

☆Raspberry Pi Pico/Pico2で動作可能

主な仕様

- ◎表示コントローラ：ST7796U
- ◎表示ドット数：480x320ドット
- ◎表示色数：65k色(16ビットカラー)
- ◎インターフェイス：SPI
- ◎容量型タッチパネル搭載
(コントローラ：FT6336U I2Cバス接続)
- ◎電源電圧：DC3.3～5V 約90mA
※I/Oは3.3V系です
- ◎その他：マイクロSDカードスロット搭載

☆コネクタピン配置

	信号名	概要
1	VDD	電源 DC3.3V～5V
2	GND	グラウンド
3	CS	ST7796 #CS(Lアクティブ)
4	RST	ST7796 #RST(Lアクティブ)
5	D/C	ST7796 D/C
6	SDI	ST7796 データ入力(MOSI)
7	SCK	ST7796 データ取り込みクロック
8	BL	バックライトLED制御(Hで点灯)
9	SDO	ST7796 データ出力(MISO)
10	CTP_SCL	タッチパネル SCL
11	CTP_RST	タッチパネル #RST(Lアクティブ)
12	CTP_SDA	タッチパネル SDA
13	CTP_INT	タッチパネル #INT(Lアクティブ)
14	SD_CS	SDカード #CS

※SDカードのSPIバスは(#CSを除き)液晶コントローラ
と共用になります

電子パーツ・マイコン・メカトロ・オーディオ

デジネット
〒556-0005 大阪市浪速区日本橋5-8-26
(2F)コンテナ2F

(月～土)10:30～19:30 (日・祝)10:00～19:00

TEL:06-6644-4555 / FAX:06-6644-1744

E-MAIL:info@diginet.co.jp

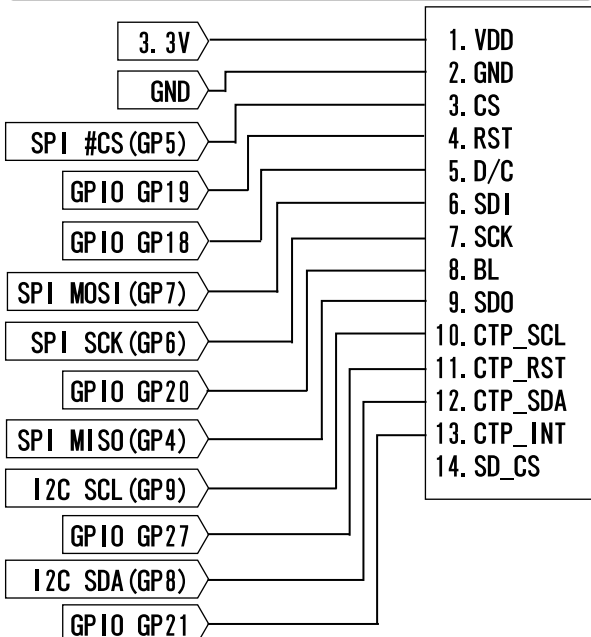
Blog:https://blog.digiparts.com/ Twitter: @0666444555

電子工作向けの学習、実験、開発向けであり
資料等は参考用です。目安程度のもので差異や誤りがある場合があり
商品の性能等を保証するものではありません。
各種設定、使用については自己責任でお願いいたします。
いかなる事故、損失においても製造者、流通者、販売者は
一切の責任を負いかねます。返品、交換、保証等の対応はしていません。

☆接続のしかた

Raspberry Pi Pico/Pico2との接続例

信号名は「ラズパイピコ実験基板」(PICO-BOARD-V1)のシルク印刷で表記しています



バックライトLEDを制御しない場合(常時点灯でよい
場合)は、液晶の8番ピン(BL)は3.3Vに接続します。

2025年 5月

☆プログラムのしかた (Arduino IDE)

Raspberry Pi Pico/Pico2をArduino IDE環境下で
プログラムするしかたについて説明します

1. ブートローダを書き込む

Arduino IDEでスケッチを書き込む際のブートローダと
なる、CircuitPythonのファームウェアを書き込みます。

下記URLからファームウェアのUF2ファイルを
ダウンロードします。

◎Raspberry Pi Picoの場合

[https://circuitpython.org/board/
raspberry_pi_pico/](https://circuitpython.org/board/raspberry_pi_pico/)

◎Raspberry Pi Pico2の場合

[https://circuitpython.org/board/
raspberry_pi_pico2/](https://circuitpython.org/board/raspberry_pi_pico2/)

**Raspberry Pi PicoとPico2で書き込むファーム
ウェアが異なります。注意してください。**

Raspberry Pi Picoモジュール上の「BOOTSEL」ボタンを
押しながらUSBケーブルを接続すると、モジュールが
USBドライブとしてPCに認識されます。

ダウンロードしたファームウェアのUF2ファイルを
このUSBドライブにコピーして貼り付けると
ファームウェアが書き込まれます。

2. Arduino IDEにマイコンボードとライブラリを追加する

Arduino IDEにRaspberry Pi Pico/Pico2用の
ツールチェーン一式をインストールします
カラーTFT液晶用ライブラリ「TFT_eSPI」を
インストールします

(1)追加のボードマネージャURLを追加する
「ファイル」メニューの「基本設定」を開き、「追加の
ボードマネージャ」に下記URLを追加します。

```
https://github.com/earlephilhower/  
arduino-pico/releases/download/  
global/package_rp2040_index.json
```

追加したらArduino IDEを再起動します。

(2)Raspberry Pi Picoのツールチェーン一式を
インストールする

「ツール」の「ボード」から「ボードマネージャ」を開き、
「RP2040」を検索します。

「Raspberry Pi Pico/RP2040/RP2350」を選択し、
インストールします。

インストールが完了したら、Arduinoのサンプルスケッチ
「Blink」を書き込み、Pico/Pico2モジュール上のLEDが
点滅することを確認します。

(3)「TFT_eSPI」ライブラリをインストールする

「ライブラリマネージャ」で「TFT_eSPI」ライブラリを
検索し、インストールします。

あとでライブラリの設定を変更する際に必要に
なるので、ライブラリのインストール先を確認して
おきます
Windowsの場合通常
C:¥Users¥<ユーザ名>¥Documents¥Arduino¥
libraries
に保存されます

3. TFT-eSPIライブラリの設定を行う

使用する液晶とI/Oポート割り付けに合わせて、
ライブラリのソースファイルを編集します
ソースファイルの行の頭に「//」を入力すると
コメントアウトされます(コンパイル時無視されます)
【例】 // #define ILI9341_DRIVER
「//」を外すとその行が有効になります

(1)「TFT_eSPI」の「User_Setups」フォルダにある、
「Setup21_ILI9488.h」を「Setup399_ST7796_RP2040.h」
にコピーします。

コピーしたファイルを次のように編集します。

```
#define USER_SETUP_ID 399  
  
#define ST7796_DRIVER — ST7796を使用します  
  
#define TFT_INVERSION_ON — 反転表示にします  
  
#define TFT_MISO 4  
#define TFT_MOSI 7  
#define TFT_SCLK 6  
#define TFT_CS 5  
#define TFT_DC 18  
#define TFT_RST 19
```

(途中省略：変更しません)

```
//#define SPI_FREQUENCY 20000000  
//#define SPI_FREQUENCY 27000000  
//#define SPI_FREQUENCY 40000000  
#define SPI_FREQUENCY 80000000  
:  
(以下省略：変更しません)
```

「SPI_FREQUENCY」のdefine文は表示データ転送用
SPIクロック(SCLK)の周波数を選択するものです。
本液晶の場合、80MHzを選択します。
(動作不安定な場合は40MHzを選択してください)

(2)「TFT_eSPI」フォルダ直下にある「User_Setup_

```
Select.h」を編集する  
  
#include <User_Setup.h>  
  
#include <User_Setups/Setup399_ST7796_RP2040.h>
```

(1)で編集したファイルを
追加します

(この間の#include文をすべてコメントアウトします)

```
#endif // USER_SETUP_LOADED
```

(以下省略：変更しません)



(3)(2)と同じフォルダにある「User_Setup.h」を編集する

このファイルは4つのセクションからなります。

◎セクション1

```
// #####  
//  
// Section 1. Call up the right driver...  
//  
// #####
```

(この間の#define文をすべてコメントアウトします)

```
//#define R61581_DRIVER  
//#define RM68140_DRIVER  
#define ST7796_DRIVER  
//#define SSD1351_DRIVER
```

ST7796を使用します

(以下Section1終わりまで省略：#define文をコメントアウトします)

◎セクション4

Raspberry Pi Pico/Pico2でSPI転送を行うための設定項目があります。

```
//#define RP2040_PIO_SPI  
//の行のコメント記号を外して有効にします。  
#define RP2040_PIO_SPI
```

SPI転送のクロック周波数は80MHzを選択します。
(表示が不安定な場合は40MHzを選択します)
(1)での設定に合わせてください。

```
// #define SPI_FREQUENCY 1000000  
// #define SPI_FREQUENCY 5000000  
// #define SPI_FREQUENCY 10000000  
// #define SPI_FREQUENCY 20000000  
// #define SPI_FREQUENCY 27000000  
// #define SPI_FREQUENCY 40000000  
// #define SPI_FREQUENCY 55000000  
// STM32 SPI1 only (SPI2 maximum is 27MHz)  
#define SPI_FREQUENCY 80000000
```

◎セクション2

ここにはESP8266/ESP32/STM32用のポート定義があります。

Raspberry Pi Pico/Pico2では使用しません。
#define文をすべてコメントアウトします。

◎セクション3

ここでは表示で使用するフォントを選択できます。
使用しないフォントの#define文をコメントアウトすればプログラムサイズを節約できますが、そのままにしておいてかまいません。

4. 液晶のサンプルスケッチを書き込んでみる

ライブラリと同時にインストールされたサンプルスケッチを書き込んで液晶部の動作を確認します。

動作確認は「TFT_Print_test」で行います。次の1箇所のみ変更してから書き込みます。

```
void setup(void) {  
    tft.init();  
    tft.setRotation(1); //引数を1に修正  
    pinMode(20, OUTPUT);  
    digitalWrite(20, HIGH);  
}
```

正常に表示されていれば液晶ライブラリ(TFT_eSPI)の設定はOKです。
次ページでタッチパネル部の使用方法について説明します。

5. タッチパネルの使用手法

タッチパネル用ライブラリ「Arduino-FT6336U」をインストールすることで本液晶のタッチパネルを使用することができます。

(1) タッチパネル用ライブラリ(Arduino-FT6336U)をインストールする

タッチパネル用ライブラリ「Arduino-FT6336U」はArduino IDEのライブラリマネージャでは見つけることができないため、ライブラリのZIPファイルを手動でダウンロードしてインストールします。

ライブラリのZIPファイルは下記URLからダウンロードできます。

<https://github.com/aselectroworks/Arduino-FT6336U.git>

ダウンロードしたライブラリのZIPファイルをArduino IDEでインストールします。

ライブラリのZIPファイルをインストールするには、Arduino IDEの「スケッチ」メニューから「ライブラリをインクルード」を開きます。
「ZIP形式のライブラリをインストール」をクリックするとファイル選択ウィンドウが出ますので、ダウンロードしたZIPファイルを選択してインストールします。

◎setup()関数での処理

ラズパイピコ実験基板のI2Cバスの割り当てがWireライブラリのデフォルトと異なるため、明示的にポートの番号を指定する必要があります

```
void setup()
```

```
{  
  Wire.setSDA(8);  
  Wire.setSCL(9);  
  ft6336u.begin();  
  (以下省略)
```

I2CバスのSCLとSDAのポート番号を指定します

◎検出したときのみ値を表示するよう変更 loop()関数内のif文

```
  if(digitalRead(INT_N_PIN) != -1)  
  {  
    if(digitalRead(INT_N_PIN) != HIGH)  
    {  
      変更します。    }  
  }  
}
```

スケッチを書き込んでパネルにタッチすると、シリアルモニタに

FT6336U Touch Position 1: (164 , 2)

のようにタッチ位置が表示されます。

(2) タッチパネルの動作を確認する

「Arduino-FT6336U」付属のサンプルスケッチを実行してタッチパネルの動作を確認します。

使用するサンプルスケッチは「ReadTouchParam」です。このスケッチは「スケッチ例」の「FT6336U CTP Controller」に収められています。

スケッチを実行する前に、スケッチを一部変更する必要があります。

◎ヘッダファイルの取り込みとポートの定義

```
#include <FT6336U.h>  
#include <Wire.h>
```

Wire.hを追加

```
#define I2C_SDA 8  
#define I2C_SCL 9  
#define RST_N_PIN 27  
#define INT_N_PIN 21
```

タッチコントローラの制御線の接続先ポート番号を指定

```
FT6336U ft6336u(RST_N_PIN, INT_N_PIN);
```

リセット信号と割り込み信号のピン番号を引数に指定

(3) タッチ位置の座標を知るには

タッチパネルコントローラIC「FT6336U」は、タッチを検出すると割り込み信号(本液晶の場合13番ピンの「CTP_INT」を「L」レベルにしてマイコンに通知します。

マイコン側のプログラム(スケッチ)では、割り込み信号の接続されているポートが「L」になったのを検出して、次の関数

```
ft6336u.read_touch1_x()  
ft6336u.read_touch1_y()
```

でタッチ位置のX座標とY座標を取得します。(この関数はタッチ位置を16ビット符号なし整数(uint16_t)で返します)

なお、同時に2箇所までのタッチを検出可能です。2箇所目のタッチ位置は次の関数

```
ft6336u.read_touch2_x()  
ft6336u.read_touch2_y()
```

で取得できます。タッチが1箇所の場合には上記2つの関数は4095を返します。