

## STK430C-02 と ESP32(開発環境 Arduino IDE を使用した画像表示)

スタータキット「STK430C-02」と「ESP32-DevKitC」を接続し、ソフトウェアで液晶とタッチパネルを制御する方法を紹介します。

お手持ちの PC に開発環境「Arduino IDE」及び ESP32 用パッケージ「Arduono Core for ESP32」をインストールし、「Arduino IDE」で「ESP32-DevKitC」を開発可能であることを前提に説明を行います。

(「Arduino IDE」及び「ESP32 用パッケージ」のインストール及び準備方法は、「Arduino ESP32」等とウェブを検索していただくと参考情報があります。)

この資料ではまず、「Arduino IDE」を使用して「ESP32-DevKitC」にソフトウェアを書き込み、「STK430C-02」に画像とタッチパネルの位置情報を表示させるまでの事を説明します。

その後、「STK430C-02」のタッチパネルの位置情報を使った制御ソフトウェアを動作させてみます。

### 1. 用意するもの

#### ①STK430C-02 (オプション: 「KS-ELKIT」)

BMP 画像登録のため、下の写真とは別に「D-sub9pin オスの USB-シリアルコンバータ (市販品)」、「microSD カード」又は「microSDHC カード」(市販品)が必要です。

#### ②ESP32-DevKitC 秋月電子通商にて購入 ¥1,480

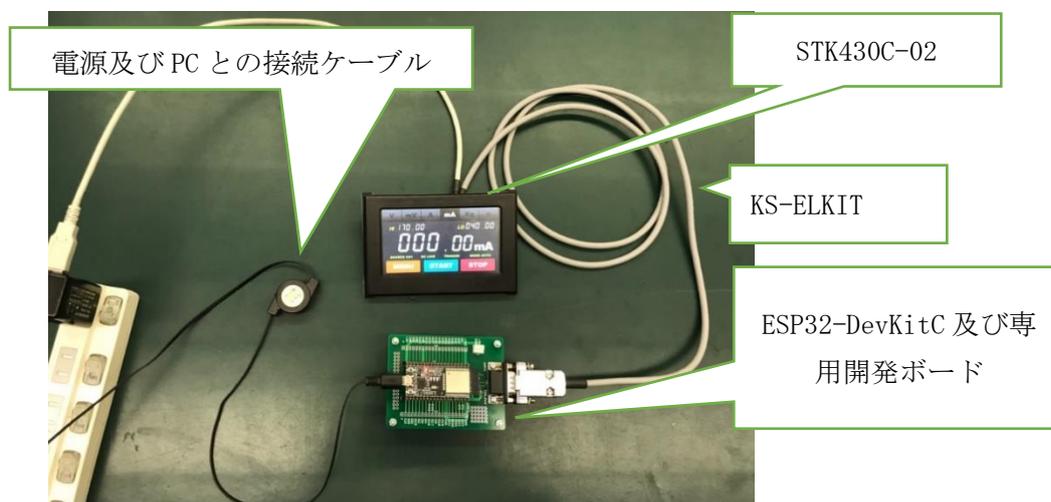
電源及び PC との接続ケーブルとして「A to microB の USB ケーブル」が必要です。

また、スマートホン用の一般的な 5V 電源か PC 等、ESP32-DevKitC 用の電源を用意してください。

#### ③ESP32 専用開発ボード

接続を簡単にするために、「ESP32-DevKitC」を挿しこんで使用できる「ESP32 専用開発ボード」を用意しました。「ESP32 専用開発ボード」の回路図はホームページ上からダウンロード可能です。また、「ESP32 専用開発ボード」は、メール等でのお申し込みにより、¥1,000-(税別)でご購入頂けます。

### 【写真: 接続イメージ図】



## 2. それぞれの準備

「STK430C-02」と「ESP32-DevKitC」をそれぞれに準備します。

### ● STK430C-02 側

ダウンロードした BMP 画像のシリアルフラッシュROMへの登録を行います。

(注：詳細はスタータキットマニュアル「6章 BMP 画像データの登録と描画 (18p)」をご参照下さい)

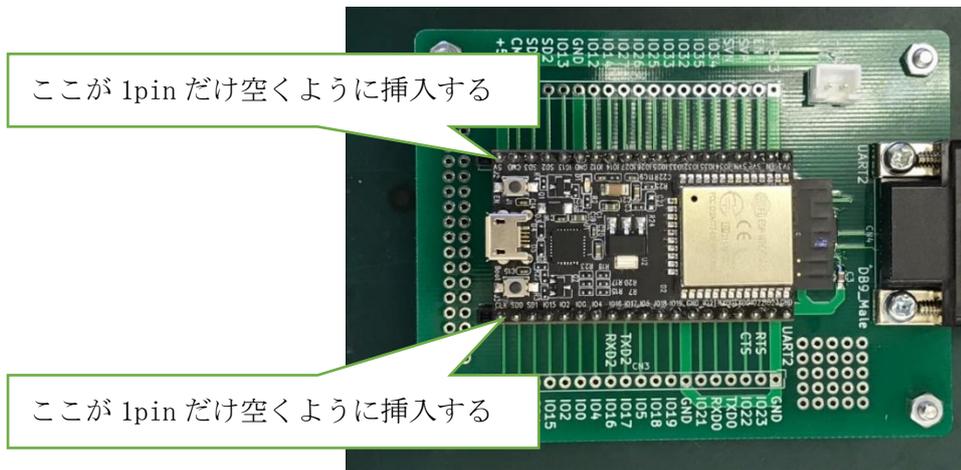
### ● ESP32-DevKitC 側

「ESP32-DevKitC と専用開発ボードの接続」、「ESP32-DevKitC へのソフトウェア」の書き込みを行います。

#### ①専用開発ボードに ESP32 を挿入

「ESP32-DevKitC」は両側に 19pin ずつの DIP ピンを持っており、「専用開発ボード」には 20pin のコネクタが 2 つ実装されています。

「ESP32-DevKitC」を「専用開発ボード」の 20pin コネクタに、D-sub9pin コネクタ側に詰めて挿入してください。



### 3. 「STK430C-02」と「ESP32-DevKitC」の通信

準備した「STK430C-02」と「ESP32-DevKitC」を接続し、実際に通信をさせていただきます。

#### ①デモソフトの実行

「1.用意するもの」の写真を参考に「STK430C-02」と「ESP32-DevKitC」を「KS-ELKIT」で接続し、「STK430C-02」の電源を入れてください。

その後「ESP32-DevKitC」の電源を入れると、「STK430C-02」に画面が表示されます。

さらに、「STK430C-02」のタッチパネルを押すと、押された位置を示すデータが画面左上に表示されます。



【写真：表示例】

「STK430C-02」はシリアル通信でコマンドを送られることによって、画面上に様々な表示をしたり、タッチパネルの押された位置を送信したりしています。

次の②、③ではどのようなコマンドが使われたかと、位置を示すデータの詳細を紹介します。

#### ②デモソフトの内容、画像の表示

デモソフトのセットアップ関数内で、「P コマンド」が送られます。

「P コマンド」は、「STK430C-02」に登録した画像を表示します。

今回は、以下の例のコマンドを送ります。

例「P000000000000」

このコマンドで、X=000, Y=000 の位置 (画面左上の原点) に、シリアルフラッシュロムに登録した「画像 0000 番」を表示させています。

(注：各コマンドの詳細は、SmartLCDC のコマンドマニュアルをご参照ください。)



【図：STK430C-02 に登録した画像 00000. bmp】

### ③デモソフトの内容、座標の確認

デモソフトのセットアップ関数内で、コマンド「t3」が送られます。

このコマンドは、「タッチパネルを押したその時のみタッチパネルの返送データを送る」というコマンドです。

コマンドを受けた「STK430C-02」はタッチパネルを押されると、下の例のようなデータを返します。

例：「t113321F」

t の直後の「1」は、タッチパネルが押されている状態を示します。

続く「133」は X 方向 AD 値、「21F」は Y 方向 AD 値です。

(注：各コマンドの詳細は、SmartLCDC のコマンドマニュアルをご参照ください。)

その後、デモソフトは文字表示のコマンドを送信して、帰ってきたデータをそのまま画面の左上に表示します。

文字表示には「F」コマンドが使われます。

今回の場合、「16 ドットフォント、縦倍角」で、「X 座標 00A」、「Y 座標 028」に対し、「黒文字を白背景」で表示しています。

(注：各コマンドの詳細は、SmartLCDC のコマンドマニュアルをご参照ください。)

#### 4. タッチパネル制御ソフトウェアの実行

ここまでの内容を応用して、ESP32 からシリアル通信で画面やタッチパネルを制御して、「STK430C-02」の画面に動きを持たせてみます。

今回は、下の写真のような動作をさせてみました。



【写真：それぞれのボタンの動作】

「ESP32-DevKitC」にあらためてソフトを書き込み、デモソフトを動作させてみます。

「2. それぞれの準備」と同様に、「ESP32-DevKitC」にホームページからダウンロードしたデモソフト「serial\_test2\_esp32.ino」を書き込んでください。

「STK430C-02」の電源を入れた後「ESP32-DevKitC」の電源を入れると、「STK430C-02」に画面が表示されます。さらに、画面の各種スイッチを押すと上の写真の動作を行います。

この制御動作は、「P」コマンドでの画像描画と、「t」コマンドでのタッチパネルデータ取得を組み合わせで作りました。

「STK430C-02」にはこれらの他にも線を引く、円を描くなど、様々なコマンドがあります。色々なソフトウェアを試してみてください。

また、今回のデモソフトは「STK430C-02」のRTS制御機能を使っています。次項では最後に補足として、RTS制御機能について紹介します。

## 5. 補足：タッチパネル制御ソフトウェアの RTS 制御

ここでは補足として、デモソフトで利用している RTS 制御機能について紹介します。

今回のデモソフト「serial\_test2\_esp32.ino」では、数字のカウントアップの度に数字の画像を追加で表示して画面表示を行っています。

画像の表示は「3. STK430C-02 と ESP32-DevKitC の通信」のようにコマンドのシリアル通信で行います。

しかし、シリアル通信で画像の表示を高速で繰り返すと、画像の表示がシリアル通信に間に合わなくなってしまいます。

そこで今回のデモソフトでは、「STK430C-02」の RTS 制御を利用しています。

RTS は「STK430C-02」の CN1 の pin4 に出力されており、「STK430C-02」がコマンド受信可能状態である場合、' L ' になります。

「STK430C-02」の RTS が ' L ' になるまではデモソフトを while ループして、シリアル通信を待機するようにしています。

なお、「Arduino IDE」でデモソフト「serial\_test2\_esp32.ino」を開き、コードの最も下の方にある「fnLCDC\_RTSset」関数の引数を(0)にすると、「STK430C-02」の RTS を無視してひたすらシリアル通信を行うソフトウェアに改造できます。

改造位置は下の図の通りです。

```
425 void setup() {
426   // put your setup code here, to run once:
427   fnInit();
428   fnLCDC_RTSset(1);           //RTSを使用(1) 使用しない(0)
429   fnLCDCt(0);
430   fnLCDCt(2);
431 }
```



ここを(1)から(0)にすると、RTS を無視するソフトウェアになります。

【図：デモソフト「serial\_test2\_esp32.ino」の改造位置】

改造したソフトウェアを書き込んで実行すると、数値の上昇スピードは速いものの、「STK430C-02」が多すぎるデータを受け取りきれず、画面の表示が間に合わなくなることが確認できるはずですが、この様子をご確認いただいて、このような表示でも問題のないアプリケーションであれば、RTS 制御を省略できるでしょう。