PCB MILLING & CREATION



text by sessakukiban.com NON PERIODICAL WEB MAGAZINE FOR PCB CREATION

CADを使って作る簡単! 切削基板 第12回

KiCadを使用して製作する

OV7670 カメラセンサー・モジュールを利用した Arduino Pro Mini 実験基板

使用するもの

- ●切削加工基板(150mm x 100mm / 1 枚/両面加工)
- ●パーツー覧にある電子部品
- ●Arduino Pro Mini (ATmega328p [3.3V•8MHz] Amazon.co.jp 販売 2個)
- ●SDHC/HC メモリーカード(2GB / Arduino の規格に準ずる)
- ●メモリーカードリーダー/ライター (ELECOM)
- ●OV7670 カメラモジュール(AL422 FIFO 付き/CSun 販売)
- ●TFT ディスプレイ液晶モジュール (2.2 インチ/ LCD / SPI 通信
- IL19341ドライブIC / Amazon.co.jp 販売)
- USB シリアル変換ケーブル (VE488 /秋月電子販売)
- 電源 5V (基板用)

※テスト時・使用言語:Arduino 言語 テスト方法: 本文参照。 PC は Windows 7 を使用。



Arduino Pro Mini (3.3V・8MHz) を2個利用し、 OV7670 カメラセンサー・モジュールから 取得した画像を 2.2 インチ TFT 液晶パネルや PC に表示するテスト、SD カードに バイナリデータとして保存するテスト、 タイマーによる連続撮影テストなど試みます。 本切削基板を発注の方には KiCad データが付属します。 そのデータを使用した オリジナル基板の制作も可能です。

JUL 2020

※本稿使用 KiGad のヴァージョンは **BZR4022** です。 ※切削加工ならびにそれを使用して完成された基板は 本誌撮影用に製作されたものです。基板の仕様,内容は、 検証を重ね、予告なく随時変更しますのであらかじめ ご了承ください。

こ了承くたさい。 米実験は自己責任により行うものとし、いかなる損害 も切削基板屋ではその責任を負いません。実験開始前 にPCなどデータのバックアップをしてください。 本本実験は、ビクセルの欠けや面像の色かぶりなどが 起こる可能性があります。写真データの質は追求して いません。あくまで実験的な使用ですのであらかじめ こ了承ください。

- 」ふてたここ。 ※本誌表示画像ははめ込みによるものではありません。

図1■液晶パネル側(写真上)、カメラ側(写真下)より各々撮影



●は、OV7670カメラ・モジュール FIFO付きを使用したテスト基板を作 成してみます。テストでは画像の取得・表 示やデータとして保存することを試みます。 最終的にはタイマーで時間ごとに画像を取 得し撮影データを SD カードに保存するこ とを目標に実験を行いました。

カメラ・モジュールや TFT 液晶モジュー ルのデータのやりとりが 3.3V 信号で行われ るため、Arduino Pro Mini は 3.3V、8MHz で動作するタイプを2個使用しました。ひ とつは主にカメラ・モジュールの設定を行 い、撮影データを取得するカメラ・モジュー ル用 Pro Mini、ひとつはその撮影データを TFT 液晶パネルに画像を表示し SD カード に保存するための TFT 液晶モジュール用 Pro Mini となります。両者はシリアル通信 でデータの送受信を行います [図2・5]。

レイアウトは表面に OV7670 カメラセン サー・モジュール、裏面に TFT 液晶モジュー

図3■パターン図(カメラモジュール用基板) (基板裏面から見て、左は表面の配線、右は裏面の配線)

◎本品は切削基板で製作しました。CADを使用してわずらわしい手配線を解消できます。



図4-テスト段階で使用した切削基板(TFT液晶パネル側) ※本基板でテスト確認後さらに修正しています。 他の箇所で使用されていますのは、修正後新たに加工した基板です。

ルを配置しました。両面加工で基板製作を 行います [図 3]。

画像の表示は PC もしくは TFT 液晶パネ ルで行います。動画として表示するために は計算速度が遅いため、ゆっくり更新され る静止画像のようになります。表示に際し、 右端 (もしくは左右両方) に緑の線が入るなど の現象が確認されましたが、画像そのもの への影響は小幅と考え無視しました。テス トで取得した画像を後ろのページに掲載し ていますのでご確認ください。カメラのレ ジスタ設定やアルゴリズム、ハード的な基 板特有の問題などが考えれますが、原因は 特定できませんでした。

カメラ・モジュール用 Pro Mini では OV7670 カメラセンサーのレジスター設定 などを I²C 通信で行い、その後パルス入出 力でタイミングを取りながらデータを取得 する処理を行います。一方、TFT 液晶モ ジュール用 Pro Mini では SPI 通信で TFT 液晶パネルに撮影データを表示する処理や そのデータを SD カードに保存する処理を 行います [図 5]。

Arduino でコードを書くにあたり参照し



図5 概略図1:取得画像のTFT液晶パネル表示やSDカードに保存実験時



図6-概略図2:取得画像のPC表示実験時やPro Miniへのデータ書き込み時

たのは、GITHUB で配布されている David Sanz Kirbis さんによる daseki/arduivision な らびに Marek Buriak さんによる ILI9341_ due です。前者はカメラモジュールの設定 およびデータ取得、後者は IL19341 を使用 した SPI 方式 TFT 液晶モジュールへの画像 の表示に利用しています。

その他、OV7670 のレジスターの説明は 『カメラ・モジュールの動かし方と応用製作』 (CQ出版)など参考資料を参照しております。 それでは製作過程と実験の詳細を見てみ ます。



● KiCad を使用し、回路図を作成後、切削加工し、切削加工基板を作成します 図2・3・
4]。回路を作成するにあたり以下の点を反映させます。

1― カメラ・モジュール用 Pro Mini が OV7670 カメラモジュール FIFO 付きから データを取得する回路。また I²C 通信でそ の設定を行う回路。

2——TFT 液晶モジュール用 Pro Mini が SPI 通信で TFT 液晶パネルに画像を表示 するための回路。TFT 液晶モジュール付属 SD カードスロットルに撮影データを保存す るための回路。

3——カメラ・モジュール用 Pro Mini と

TFT 液晶モジュール用 Pro Mini 両者間にて シリアル送受信可能にする回路。

4―カメラ・モジュール用 Pro Mini お よび TFT 液晶モジュール用 Pro Mini を ADM3202ANZ を経由して PC とシリアル 通信で書き込みを行うための回路 (カメラ・ モジュール用 Pro Mini から PC に撮影データをシリ アル送信することで PC での表示も可能)。

上記 1-3 については図5の概念図を、4 については図6の概念図をご確認ください [図5・6]。シリアルポートの併用は、ジャン パーピンで対応しています。

また、OV7670 カメラ・モジュールから データ取得の際に使われる下位 2 ビットで 使われる入力端子は Arduinoの RXI ピン TXO ピンと兼用になります。PC でデータ 表示の場合や Arduino にデータを書き込む 場合はジャンパーピンの接続をオープンに してこの入力端子は使用しません。

TFT 液晶モジュールに 3.3V を入力した ところ、うまく機能しませんでしたので Arduinoの RAW 端子に入力した 5V をこち らへも入力しました。それ以外は、TFT 液 晶モジュール、OV7670 カメラ・モジュー ルともに端子の H レベルはすべて 3.3V と なります。

過去から現在にかけて販売された OV7670 FIFO 付きカメラ・モジュールで



パーツ名称	個数
抵抗 1K	3
コンデンサ 0.1μF	5
タクトスイッチ基板用	1
トグルスイッチ基板用	2
TFT LCD ディスプレイ液晶モジュール	
2.2 インチ IL19341 SPI 通信(Amazon)	1
Arduino Pro Mini (ATmega328p 3.3V 8MHz Amazon)	2
LED 5mm	1
カメラモジュール OV-7670	1
DC JACK MJ-179PH	1
DB9 コネクター	1
16PIN ソケット	1
IC ADM3202ANZ	4
ピンヘッダー シングル 12	4
ピンヘッダー シングル 9	1
ピンヘッダー シングル 6	1
ピンヘッダー シングル 5	1
ピンヘッダー シングル 2	1
ピンヘッダー ダブル 11x2	1
ピンヘッダー ダブル(オス)2x2	2
ピンヘッダー ダブル(オス)1x2	10
ジャンパーピンセット(2.54 ミリピッチ)	1

表1■パーツ一覧

は、最低でも2種類の異なる回路図が確認 できました。詳細は省きますが、一つはカ メラ・モジュール内の配線で、OV7670の VSYNC 信号が AL422B(FIFO)の WRST へも伝達されるタイプのもので、一つはそ のような伝達がなされません。今回入手し たのは後者のものでしたので、カメラ・モ ジュール WRST ピンに、規定のタイミング で Arduino から Write Reset 信号を出す必要 がありそのような回路設計になっています。 ②パーツリストを参照の上、必要な部材を 装着し基板を完成させます [表1・図7・8]。

● Arduino開発環境でコードを開発します。カメラにつきましては前述の通り、 GITHUB配布のdaseki/arduivisionを利用しています。カメラ・モジュールのレジスター設定などほぼ流用している箇所もありますが、VSYNC信号による割り込みを使用したアルゴリズムはやめ、切削基板屋ではVSYNC信号のポーリングによるかたちに変更しています。

TFT 液晶パネルへの表示につきましては 前述の通り、GITHUB 配布の ILI9341_due を利用しています。TFT パネル初期化や TFT パネルで画像を表示するのに必要な部 分だけを抽出・変更をして使用しています。 カメラ・モジュールの設定は、daseki/ arduivisionでは、QQVGAと 80x60サイズを YUYV 形式のカラーデータで読み込むこと を前提にしています。切削基板屋ではあわ せて QVGA と 40x30 のテストも行いまし た。また RGB444 形式が画質が荒いので検 証はせず、カメラ・モジュールのレジスタ 設定による RGB565 形式での画像取得でも 思うような画質が得られなかったので検証 はしてません。

今回のテストでは、画質的に見て、YUYV 形式のカラーデータを Arduino 内でアルゴ リズムで RGB565 形式に変換するのが良い と判断しました。この RGB565 形式は TFT 液晶パネルにに1ピクセルごと画素に色 データをセットするために必要なデータ形 式ですので、必ずアルゴリズムによる変換 が必要になります。

カメラ・モジュール用 Pro Mini で取得し た YUYV 形式のデータを RGB565 形式に変 換し、そのデータをソフトウェアシリアル で TFT 液晶モジュール用 Pro Mini 側に取 得画像の1水平ラインごとに送信していま す。TFT 液晶モジュール用 Pro Mini はその データをその都度 TFT 液晶パネルに表示し たり、SD カードにデータ保存しています。



図10-テスト2:PCでの表示 (Processingで受信し、画像として表示)







0

図11-テスト3:SDカードに保存された バイナリーデータをProcessingの ソフトで変換したBitmap画像。 上から取得サイズは320x240ピクセル、 160x120ピクセル、80x60ピクセル、 40x30ピクセル。 なお、今回の実験に関して、結果がうま くいかいない場合は、実情に即したコーディ ングを行っています。くわしくは本文末尾 の注をお読みください。

● PC 側のシリアル受信用および画像保存 用して Processing2.21を使用したソフトを 開発します。PC 側のシリアル受信用ソフト は全データ受信してから表示します。画像 保存用ソフトは、SD カードに保存された バイナリデータを PC で確認、Bitmap 形式 で書き出すためのソフトです。

画像編集ソフトとして GIMP を使用しま す。Processing も GIMP もそれぞれ無料で ダウンロードできますので、あらかじめ用 意します。

PC で表示するための Processing のコード は David Sanz Kirbis さんによるものは使用 せず切削基板屋で開発しています。いずれ にしましても実験をしながら修正を加えて います。

●外部 5V 電源挿入で電源投入をします。 実験に際し、それぞれジャンパーピンで設 定を変え、カメラ・モジュール用 Pro Mini、 TFT 液晶モジュール用 Pro Mini に書き込 みを行いました。書き込み時にはカメラ・ モジュールと TFT 液晶パネルを外して行い ます。その後、これらを再装着し、実験用 にジャンパーピンをセットし直します。

◆テスト1

TFT 液晶パネルに画面表示します。Arduino Pro Mini の動作周波数、シリアル通信の速 度、さらには SPI 通信も必要なため表示速 度はとても遅いです。TFT 液晶モジュール 用基板のトグルスイッチ(SW1)を切り替 えることでフレームサイズを変更できます。 今回開発したコードでは連続撮影やタイ マー撮影などの撮影期間中のフレームサイ ズの変更はできません[図1・9]。

◆テスト2

画像をとりこみ PC に転送して表示させま す [図 10]。なお Arudino 上ではデータを保 存せず、YUYV は 1 バイトごとに送信し、 YUYV を RGB565 に Arduino で変換した場 合は 4 バイトごとに送信します。

YUYV を RGB565 に変換しているのはプ ログラミングによってです。両者とも1 画 素につき、2バイトずつのデータとなります。

PCとの通信は仮想 COM ポートにシリア ル通信による送受信になります。受信側は Processing を使用しています。Processingの ソフトでは画像データ送信前に Arduino と やりとりして YUYV や RGB565 という設定 やフレームサイズの設定を行っています。 なおフレームレートは成り行きです。 Arduino Pro Miniの動作周波数が遅い上、 シリアル通信のボーレートも高く設定でき



図12■テスト3: SDカードに保存されたデータを Processingで表示

ませんでしたので動画としてはとても遅い と考えてください。

TFT 液晶パネルに表示せず、PC に転送す る場合は、トグルスイッチによるフレーム サイズの変更は考慮しませんでした。デフォ ルトのフレームサイズのみで表示可能です。

◆テスト3

画像をとりこみ SD カードに保存します[図 11・12]。SD カードは 2GB の新規の SDHC/ HC メモリーカードを使用し、撮影終了後 は、メモリーカードリーダー/ライターで PC で読み込みました。予期せぬトラブルが 起こるかもしれませんので重要なデータが 入っている SD カードは使用しないでくだ さい。なお SD カード内のデータを消去す る場合はメモリーカードライターの機能が 必要と思いますので両機能が使えるものが 必要です。

実験では、TFT モジュール用基板の押し ボタンを一回押すごとに指定枚数分の画像 をフレームごとに取得します。

取得画像はトグルスイッチにより2サ イズ選べますが、大きい順から320x240 ピクセル(QVGA)、160x120ピクセル (QQVGA)、80x60ピクセル、40x30ピク セルの計4サイズから選択可能となってい ます。 図13-タイマー設定による撮影画像(320x240ピクセル)。Processingのソフトでの画面表示をキャプチャ。 電源投入からの時間経過を表示。タイマー設定はかけたものの、設定時間が短かったため、最小表示周期 (51秒) ごとに画像取得され、時間が表示にされています。このサイズの画像の場合は約1分以上のタイマー設定を考える必要があります。

◆テスト4

設定時間ごとに画像をSDカードに保存し ます。Arduinoで取得枚数、時間間隔を設 定をしてTFTモジュール用基板側の押しボ タンを押し、取得開始としました。時間間 隔についてはジャンパーピンで4つの間隔 が選べるようにしました。

◎データサイズ 160x120 約 38.4KB320x240 約 153.6KB

なお SD カードに保存されるファイルはす べて同じ時刻になりますので、Arduinoの 電源投入してからの経過時刻をファイル ヘッダーとしてバイナリファイルに含めてい ます。Processingのソフトで読み込んだ場合、 経過時刻が画像の下に付されます [図 13]。

●注 YUYV 形式のデータの並びは、レジスターで設定できるはずですが、MVFP レジスタの水平反転設定などの影響をうけたり、サイズにより HREF レジスターなどの設定の影響も受けているのではと考えられました。16MHzのAruduino Mega で OV7670 カメラ・モジュールのテストした際、たとえば水平反転設定をした場合、赤い色が青い色に表示されるなどの現象も確認されました。

YUYV 形式のデータ配列 Y0、U、Y1、V から、思っ ているような画像が得られるようにするにはどうすれば いいかから逆に順番を予想して適用してます(たとえば UとVが逆になっているのでは?と思えた場合は、結



0003.BIN----0:3:21



0004.BIN----0:4:12



0005.BIN----0:5:3

果が正しくなるようにアルゴリズムを調整しています。) 結果から推測してデータの並びを把握を理解すること につきましては、今回はあくまで実験レベルでなのでご 容赦ください。使用する基板に起因するものかアルゴリ ズムによるものかは不明です。

注意事項/免責事項

◎電子工作は適切な知識のもと、安全面に十分ご配慮し て行なってください。

◎本 PDF マガジンの内容を利用する場合は、使用者の 自己責任において行うものとします。

その際、使用者にいかなる損害、被害が生じましても、 発行者、執筆者、PDF 制作関係者は一切の責任を負いま せん。あらかじめご了承ください。

KiCad の使用方法については PCB MILLING & CREATION 01 号~ 03 号を参照下さい。

- 01号:回路図を手書きで直ぐパターン図作成
- 02号:回路図を KiCad で作成しリンクしてパターン図を作成 03号:ライブラリー作成方法