

ディスプレイ描画プログラミング

～図形や文字をマイコンで描画する～

操田浩之

第2章では、ディスプレイに何らかの図形などを表示するCRTCをCPLDで設計した。ここでは、CRTCに送り込む文字や点、線などといったビデオ・データの設計方法について説明する。
(編集部)

英数字や外字および直線、円をディスプレイに描画する方法について解説します。また、描画用マイコン(ADuC7026 米国 Analog Devices 社)で取得したセンサからのアナログ・データを、リアルタイムにグラフ表示する機能、写真などのフルカラーの画像を8色に減色して表示する機能(誤差拡散アルゴリズム)についても説明します(写真1)。

1. 描画用マイコンとCRTC用CPLDとの接続

基板面積を節約するために描画用マイコンと第2章で設計したCRTC用CPLD(MAX II)との接続は、最短距離を必要最小限の信号線で接続することにしました。I/O端子の接続関係を表1に示します。描画用マイコンとSRAMとの接続は、すべてCRTC用CPLDを経由して行っています。

● MAX II は内部で自由に配線できるためマイコンの端子設定の自由度は高い

描画用マイコンであるADuC7026のポートは、A-DポートおよびD-Aポートが一般のI/Oポート(GPIO)として使えません。また、I/Oポートもポート番号順に連続で並んでいません。例えば、今回使用したポート3(P3)の場合、

P3.0からP3.3までは端子番号29から32に割り当てられていますが、P3.4とP3.5は38、39ピン、P3.6とP3.7は46、47ピンといったように、連続して並んでいません。しかし、CRTC用CPLDに接続すれば、CPLDの内部で自由に配線を組み替えることができるので、1組の8ビット分のデータ・ポートさえ選べば(今回はP3)、残りの端子は別々のポートからでも構いません。

● CRTC用CPLDがディスプレイにデータを送出している間はアクセスできない

第2章で紹介したCRTCは、ディスプレイへの表示を優先して、表示のちらつきを抑えるようにしています。そのため表示データの更新は、ディスプレイ表示のためにCRTC用CPLDがSRAMにアクセスしていないタイミングで行う必要があります。

SRAMのデータの書き換えは、8ドット×3色分のデータをバッファ(Dフリップフロップ)に転送した後から、このデータがシフト・レジスタにセットされるまでの間に行っています。つまり、CRTC用CPLDはSRAMへアクセスする際に、ディスプレイ表示期間とSRAMデータ更新期間に分けてアクセスします。そのため、画面が乱れることなく表示内容を書き換えられるのです。

● ADuC7026 はあらかじめダウンローダ・プログラムが書き込まれている

ADuC7026の端子はアドレス・バスおよびデータ・バスとして使用できますが、プログラムの際にはUARTを除いて

Keyword

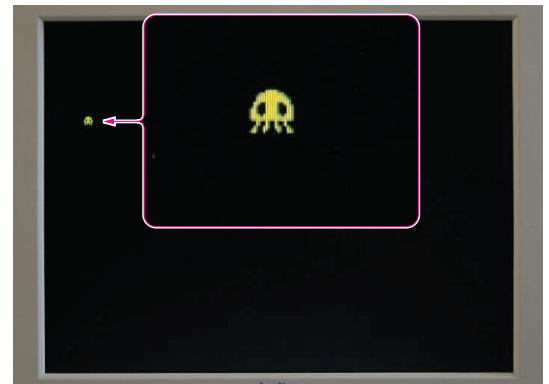
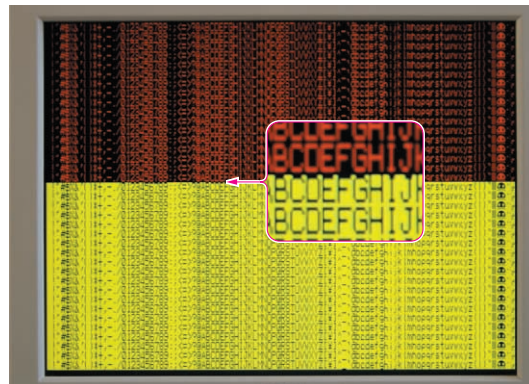
ADuC7026, グラフィックス表示, 文字の表示, 文字色, 背景色, 点と直線の表示, 4則演算, Michener, デジタル・オシロスコープ, 加速度センサ, グラフ表示

GPIOとして扱っています。なお、ADuC7026は、あらかじめダウンロード・プログラムが書き込まれています。JTAGは省略し、プログラムのダウンロードとデバッグはシリアル・ポート(USB)を介して行うことにしました。

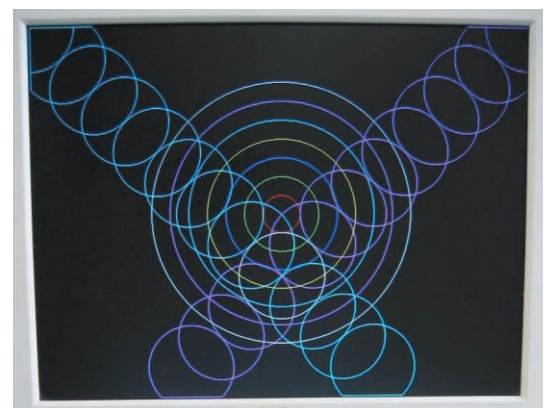
プログラムの開発ツールは米国 Keil 社の統合開発環境 μ Vision3 を使用し、ダウンロード・プログラムは米国 Analog Devices 社の ARMWSD を使用しました。どちらもフリーでダウンロードして、使用できます(図1)。

● 描画用マイコンの動作クロックはCRTC用CPLDのピクセル・クロックと同期していなくてもよい

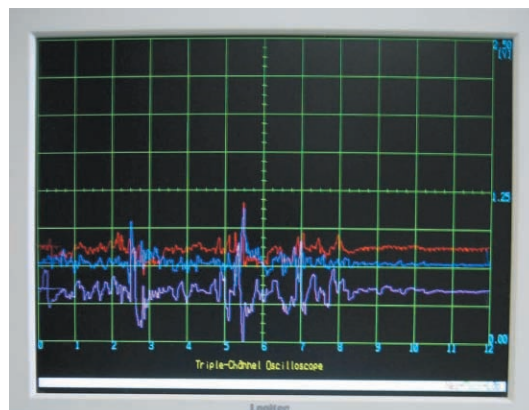
マイコンの動作クロックは、内部のPLL回路で発生するクロックを利用して、41.78MHzで駆動させています。今回、データのSRAMへの書き換えはすべてCRTC用CPLDが行っています。そのため、描画用マイコンの動作クロックがピクセル・クロックの40MHzとは同期しない41.78MHzで動作していても、画面がちらつくことはありません。



(a) 文字(外字)表示



(b) 図形表示(直線, 円)



(c) グラフ表示



(d) 誤差拡散による画像表示

写真1 制作したプログラムによる表示