

タダで使えて高機能!

8-1 オープン・ソースの画像処理ライブラリ

山本 隆一郎

従来の単純なセグメント型液晶であれば、ライブラリを使わずにマイコンやロジック IC で直接的に簡単に制御できます。しかし、JPEG や PNG で圧縮された画像ファイルを、マトリクス型液晶に表示するのは少々面倒です。そこで活躍するのが、各種の画像処理ライブラリです。

本稿では、主に組み込み機器の開発に利用できる画像処理ライブラリを紹介します。画像処理は主に画像の表示と認識に大別されるため、ここでは主に表示系について述べます。



オープン・ソースの メリット/デメリット

画像表示のライブラリを選択する際には選考する基準として、機能面での差異はもちろん、オープン・ソースか商用かで判断が分かります。製品用途では、以前は主に商用のものが広く採用されていましたが、Linux や Android などのオープン・ソースが認知されるようになり、それらで使われている

オープン・ソースの画像処理ライブラリも積極的に利用されてきています。目的や環境で異なりますが、オープン・ソースと商用ライブラリの採用判断の目安となるそれぞれの特徴を表1に示します。

▶オープン・ソースは入手コスト0で自由度が高い

大学の研究室や個人的な開発ならば、入手コスト面で圧倒的にオープン・ソースに分があります。また、試作用途や、製品サイクルが長くターゲット CPU をずっと使うか不確定な場合、またはライブラリそのものの永続性が不安な場合に適します。ただし、何かあった場合には、自分でソース・コードを読むか、周りにそれを代替してくれる人が団体を見つける必要性があり、オープン・ソースならではの自助努力が求められます。逆に言えば、それさえ何とかできれば、これほど自由度の高いものはありません。

▶スムーズに開発するなら商用ライブラリ

製品開発では、商用ライブラリがターゲット CPU に対応しており、かつメーカーやサードパーティが積極的に推奨している場合には、これを利用すればスムーズに開発できます。

表1 オープン・ソースと商用ライブラリのメリット/デメリット

メリット	<ul style="list-style-type: none"> • ソース・コードが公開されている。 • 最初の開発が途絶えてもメンテナンスを引き継がれる可能性が高い。 • 自分でもメンテナンスできる。 • 自分の使いたい CPU に移植ができる。 • バグや不具合が出て自分で修正できる。 • インターネット上で様々なノウハウが公開されている。 • 機能改善やバージョン・アップのスピードが速い^注。 • ライセンス費用が不要
デメリット	<ul style="list-style-type: none"> • ソース・コードを読んだり変更ができないと不具合に対応できない。 • 非開発部門からの信頼が低め。

注：プロジェクトによっては遅いものもある

(a) オープン・ソース

メリット	<ul style="list-style-type: none"> • 導入から開発までサポートが受けられる。 • 非開発部門からの信頼が高い
デメリット	<ul style="list-style-type: none"> • ライセンス費用がかかる。 • 特定の CPU しかサポートしない場合が多い。 • 代理店が頻繁に変わったり開発元が無くなる可能性がある。 • 機能改善やバージョンアップのスピードが遅い^注。

注：製品によっては速いものもある

(b) 商用ライブラリ



代表的なオープン・ソースの ライブラリ

オープン・ソースの画像処理ライブラリは、著名なものからマイナーなものまでさまざまあります。よく利用されているライブラリを表2に示します。特定の CPU や環境によらずに利用できるオープン・ソースの画像処理ライブラリです。

一覧にあげたライブラリは、役割や規模も異なります。たとえば、libpng は主に PNG 画像のデコードを行うだけのシンプルなライブラリですが、GTK+ はそれらを含む、画面を構成する部品を取り扱う UI ライブラリです。Android にいたっては、さらに上位の単なるライブラリ群ではなく、むしろ OS という別のカテゴリになります。

各ライブラリは目的に応じて、JPEG や PNG のようなスタ・データのデコードを担うものや、freetype のように