

QtでキュートなGUIを作る 初歩をマスター！

池田 誠

Qtは、クロス・プラットフォームのアプリケーションとGUI開発フレームワークとして知られる。組み込みで使う事例もよく目にする。本章では、ARM向けの開発環境を構築し、Qtによるアプリケーションの特徴であるシグナル・スロットとレイアウトを解説する。

(編集部)

1. Qtとは

Qt (キュート) は、Nokia社の一部門であるQt Development Frameworksが開発を行っているクロス・プラットフォームのアプリケーション用GUI開発フレームワークです。Linux/X11, Mac OS X, Windows, Embedded Linuxといったプラットフォーム^{注1}で共通のC++インターフェースを持ちます。一度作成したソース・コードをそれぞれの環境でコンパイルすることにより、各環境でネイティブに動作する実行可能ファイルを作成できます。

Qtアプリケーション動作時のレイヤ構成を図1に示します。どのプラットフォームで開発する場合でも、アプリケーションから利用するQtのAPIは共通化されています。上位層のほとんどはQt APIを使って記述されており、その多くはプラットフォームに依存しない共通化された実装です。一方、下位層では、各プラットフォームのネイティブな

APIを使用して実装されています。例えば、Qt/WindowsではWin32 APIやGDI+API、Qt/X11ではPOSIXやXlib、Qt/Mac OS XではCocoaといったAPIが使われています。

組み込みLinuxでX11ウィンドウ・システムを使わない場合には、ウィンドウ・システムがありません。そのため、Qt/Embedded Linuxでは、Qt自身でQWS (Qt Window System) というウィンドウ・システムを持ちます。QWSでは、Qtがフレーム・バッファに直接描画してウィンドウ・マネージャの機能を提供します。また、キーボードやマウス、タッチ・パネルのデバイス・ファイルにも直接アクセスしてデバイスからの入力を扱えます。デバイスを扱う部分はプラグイン形式となっており、新たな機器に対応させたい場合、ユーザが自由にプラグインを作成できます。

Qtのソース・コードは全て公開されているので、開発者は必要に応じて確認できます。特定のバージョンだけでなく、開発中の最新の状態のQtも公開されたgitリポジトリ (qt.gitorious.org) から参照できます。さらに、将

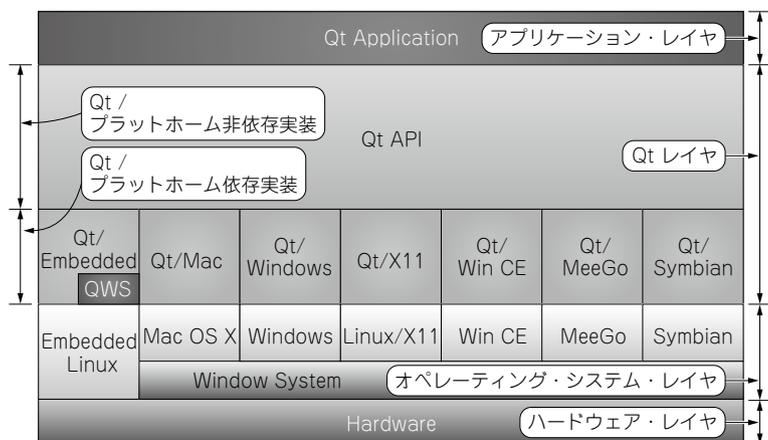


図1
アプリケーション/
Qt/OSのレイヤ構成

注1: Qt対応プラットフォームは、Nokia社のWebサイト (<http://doc.trolltech.com/4.7/supported-platforms.html>) で確認できる。Tier 1, 2は特に重視され、自動テストの対象となる。それ以外はTier 3と呼び、自動テストの対象にならないが、Qtが動作するプラットフォームは多い。組み込み向けでは、QNX, VxWorks, Integrity, eT-Kernelなどで動作するほか、iOSやAndroid向けのQtも開発されている。