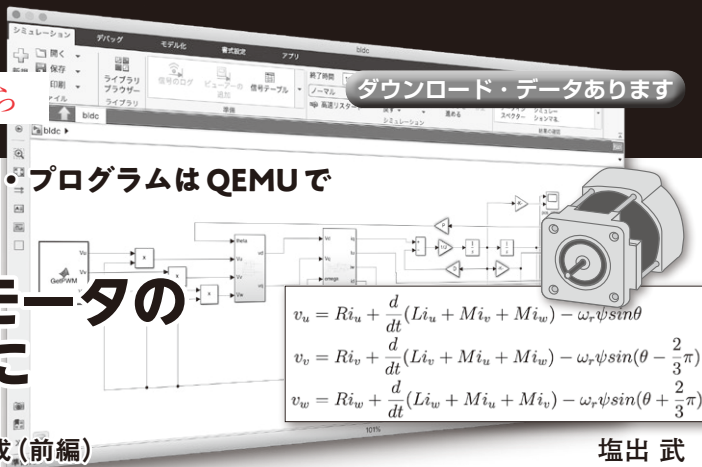


モータ制御はSimulink, マイコン・プログラムはQEMUで

## TRY 仮想開発… DC ブラシレス・モータの ベクトル制御を例に

第4回 QEMUによるRX62T  
マイコン・エミュレータの作成(前編)

塩出 武



```
qemu-6.0.0
├── build      make用に作成
├── hw
│   ├── adc   A-Dセンサ追加
│   ├── arm   Armアーキテクチャ
│   ├── gpio  PORT機能追加
│   ├── rx    RX62Tマイコン追加
│   ├── timer MTU3タイマ追加
│   └── include 主にhw内.cファイルに対応した.hファイル
└── rx-bin    makeしたエミュレータの置き場
```

図1 qemuのディレクトリ構成(一部)

SimulinkでDCブラシレス・モータをモデル化したのに対して、RX62TのマイコンのエミュレータをQEMUで作成します。QEMUはバージョン5.1よりRenesas RXとしてRX62Nが追加されています。これをベースに今回のベクトル制御に必要なRX62Tの機能を追加します。

前編ではQEMUの基本的な使い方とRXマイコンへの対応、後編ではMATLABとQEMUをつなぐ方法を解説します。

### エミュレータQEMUを使う

#### ● QEMUでさまざまなCPUをエミュレート

QEMU<sup>注1</sup>は、x86やPowerPC、ArmなどのさまざまなCPUアーキテクチャの命令をエミュレートできるオープンソースのエミュレータです。ISS(命令セット・シミュレータ)の意味合いが強いですが、タイマやI/O、シリアル通信、A-DコンバータなどCPUの周辺機能もエミュレーション可能です。そのため対応モデルにはCPUだけでなく基板そのもののモデルも出てくることもあります。またソースコードが公開されておりカスタマイズ可能なため、CPUが対応されていれば、SoC(System on a Chip)のような場合であっても、機能を追加して専用エミュレータを構築できます。

注1:「キューイーエムユー」と読みますが、「キューエム」と呼ぶ場合もあるようです。

#### リスト1 MSYS2のターミナル上でパッケージを追加する

```
pacman -Syu # パッケージを最新に更新
pacman -Su # 残パッケージを更新(必要に応じて)
pacman -S base-devel mingw-w64-x86_64-toolchain git
python ninja # git, python, ninjaをインストール
pacman -S mingw-w64-x86_64-glib2 mingw64/
mingw-w64-x86_64-gtk3 mingw64/mingw-w64-x86_64-SDL
python-setuptools # glib2, gtk3, SDL2をインストール
```

今回は安定して動作確認されているQEMUバージョン6.0.0を使っていますが、原稿執筆時点での最新版は8.0.2となっています。

#### ● QEMUのディレクトリ構成

QEMUのディレクトリ構成の一部を図1に示します。マイコンのレジスタ機能は主にhwディレクトリ以下のファイルにて管理されており、タイマやポートといった機能ごとのフォルダとarmやrxといったマイコン機能を取りまとめるフォルダに分けられています。

### QEMUの開発環境とmake

QEMUはgcc環境でmake(ビルド)できます。QEMUのサイトから必要な情報を得ることが可能です。筆者はQEMUのWiki<sup>(1)</sup>を参考にMSYS2の環境でmakeしました。以下にmakeに当たって必要な環境準備とコマンドを記載します。

#### ● 開発環境の準備

##### ▶ パッケージの追加

MSYS2のターミナル上でリスト1の手順でパッケージを追加します。

##### ▶ ソースコードの準備

公式ウェブ・サイトではgit clone <https://www.gitlab.com/qemu/qemu>としてgitクローンする方法を記載していますが、ダウンロード・ページ、

<https://download.qemu.org/>から入手しても構いません。筆者はこの方法としました。

