



## 第11章 サンプル・プログラムでGPUのパフォーマンスをチェック

# CUDA Cプログラミング入門② スレッドの生成と行列の加算/乗算

本章ではJetson NanoのGPUを直接動かす並列処理プログラムをいくつか作成してみます。また、それらの性能評価や、GPUの使用効率を測定します。  
〈編集部〉

### 11-1 CUDA Cプログラミングの準備から実行まで

#### ● Jetson NanoのCUDA Cプログラミング環境を準備

Jetson Nanoでは、SDカードを作成した時点でCUDA Cプログラミング環境はすでにそろっています。ただし、環境変数\$PATHと\$LD\_LIBRARYを設定するため、ファイル~/bashrcの末尾にリスト1の記述を追加してください。

編集後、以下のコマンドを入力して、CUDA Cコンパイラであるnvccコマンドのバージョンが表示されればOKです。

```
$ source ~/.bashrc
$ nvcc --version
```

#### ● サンプル・プログラムを入手

本稿で使うサンプル・プログラムを下記のようにGitHubからダウンロードしてください。jetson\_nano

リスト1 CUDA Cプログラミングのための~/bashrcへの設定追加

```
export PATH=$PATH:/usr/local/cuda/bin:/usr/sbin
export LD_LIBRARY_PATH=/usr/local/cuda/lib
```

ディレクトリ内に8個のディレクトリができていることを確認しておきます。このうちGPIOは第4章で説明したものです。

```
$ cd ~/Documents
$ git clone https://github.com/munetomo-maruyama/jetson_nano.git
$ cd jetson_nano
$ ls
GPIO GPUDEV GPUFP32 GPUMUL README.md
GPUADD GPUFP16 GPUFP64 GPUPI
```

#### ● ディレクトリ構成

本稿における、CUDA Cプログラム開発用ディレクトリ構成を図1に示します。

これは本稿で使用するプログラムのビルド用Makefileが前提としている構成であり、ソース・ファイルと、コンパイル&ビルド結果のバイナリ・ファイルを格納するディレクトリを分離してスッキリさせています。

最初にユーザがプログラムを開発するときは、作業用ディレクトリを作成し、その下に、プログラムのビルド用Makefileと、ソース・ファイル格納用ディレクトリsrcと、そのsrcの下にソース・ファイル(ヘッダ・ファイル\*.hと本体ファイル\*.cu)を作成してください。

Makefileでビルドすると、オブジェクト格納用ディ

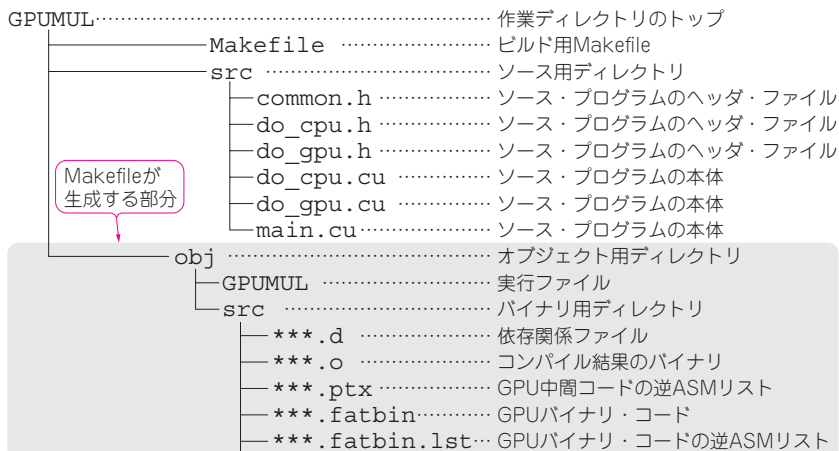


図1 CUDA Cプログラム開発用ディレクトリ構成

【セミナー案内】 [講師実演] [ビギナー向け] 初めてのアナログ回路設計講座：センサ信号処理の徹底強化(その2)  
——センサ信号処理と直線性補正から、校正用疑似信号発生器の製作までをカバー  
【講師】 中村 黄三 氏, 9/5(木) 24,000円(税込) <https://seminar.cqpub.co.jp/>