

ARM達人への道!

定番ARMプロセッサ・
プログラミング入門

第2回

まずCortex-Aの元祖から!
Cortex-A8実験用ハード&ソフト

ご購入はこちら

第1回では、本連載でプログラミング方法を説明するARM Cortex-Aプロセッサや実験の構成などを紹介しました。今回からはARM Cortex-Aプロセッサ・コアの元祖Cortex-A8の裸(Baremetal)プログラミングを紹介します。

Cortex-A8は、最初のCortex-Aプロセッサ・コアです。超定番のCortex-A9コアのように、マルチコアに対応できるわけではありませんが、シングル・コアの元祖から始めるのが分かりやすいのではないかと思います。(編集部)

ターゲットCPU&ボード

今回、元祖Cortex-A8プログラミングを試す実験ボードには、約7000円から入手できるBeagleBone Blackを選びました。AM3359(TI, テキサス・インスツルメンツ)という型名の、Cortex-A8内蔵SoC(System on a Chip)を搭載しています。

通常はLinuxボードとして使うことを想定しているのですが、ラズベリー・パイなどと異なり、SoCの情報が公開されているため、Cortex-Aプロセッサの裸のBaremetalプログラムを始めるにはちょうどよいのではないかと思います。

BeagleBone Blackは最高動作周波数が1GHzのSoCの他、512MバイトRAM(DDR3)や4Gバイト・フラッシュ・メモリ(eMMC)を搭載しています。

BeagleBone Blackには、BeagleBone Black System Reference Manual Rev A5.2というマニュアルがありますが、Baremetalプログラミングに必要な情報は、限られています。TIのサイトから入手できるAM335x ARM Cortex-A8 Microprocessorsには、チップの詳細情報が載っています。

利用するソフトウェア

BeagleBone Blackは、Baremetalプログラミングのためのソフトウェア・フレームワークStarterWareが用意されています。このため、割り込み処理の詳細や

I/Oマップを全く気にしなくても、Baremetalプログラムを作成できます。しかし、これでは、とてもCPUを理解した上で、Baremetalプログラムを作成したことにはなりません。

そこで今回、プログラム方法を理解するために、最低限として、I/O処理はもともとCPUに依存する部分なので、StarterWareを使うことにします(図1)。

それ以外のスタートアップ・ルーチンや割り込み処理は、ARM共通で使える枠組みに書き換えます。

StarterWareには、ARM Cortex-A8を使う上で必要なアセンブラ処理などが、多数含まれています。

今回は、ARM共通部分の処理方法しか説明しませんが、Cortex-Aの詳細処理を知るには、とても良い例題になっていますので、必要に応じて対応するプログラムを説明していきます。

ターゲット・ボードBeagle Bone Blackのメモリマップ

512Mバイト、0x8000 0000から外部メモリとなっているのが必要な情報です。

I/Oの情報は、AM335xのマニュアルにあります。今回は、StarterWareを基本に使うのでほとんど、そのリンク情報に記されています。

```
MEMORY {
    ram (rxw) :    ORIGIN = 0x80000000,
    LENGTH = 0x10000000    /*check */
}
```

I/O処理(CPU固有)

AM3359は、32ビットのGPIOモジュール4つで128ピンのGPIOを制御します。このGPIOは、設定を変更すれば、さまざまな用途に使えるようになっています。

今回は、StarterWareを、そのまま使っていますので、GPIOの設定はこの中で行います。かなり巨大なライブラリ・ソフトウェアで、基本的にStarterWareに任せてしまいますので、ここではGPIOの詳細な設定の説明は省略します。

◆参考・引用*文献◆