

コンポーネント解説

Ver 1.0.3

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、厳重な取り扱い、管理を行ってください。

目次

1	コンポーネントの概要	4
1.1	TWL-SDKにおけるコンポーネント	4
1.2	制御するデバイス	5
1.2.1	デジタルボタン	5
1.2.2	サウンド回路	5
1.2.3	SPIデバイス	5
1.2.4	リアルタイムクロック	5
1.2.5	ワイヤレス通信モジュール	5
1.2.6	カメラモジュール	5
1.2.7	SDCard I/F	6
1.2.8	AESモジュール	6
2	コンポーネントの種類	7
2.1	mongoose	7
2.2	ichneumon	7
2.3	mongoose.TWL	7
2.4	raccoon.TWL	7
2.5	ferret.TWL	8
3	コンポーネントの選択	9
4	コンポーネントの判定	10
5	注意点	11

改訂履歴

版	改訂日	改 訂 内 容
1.0.3	2008-12-05	・ TWL-SDK に対応
1.0.2	2005-09-01	・ コンポーネントはプログラム実行中に動的に切り替えることはできない事を概要説明に明記
1.0.1	2005-06-09	・ SND ライブラリが SDK に組み込まれたことによる記述修正 ・ ARM7 側プログラムの配置位置の記述を 27e0000 番地に修正
1.0.0	2005-03-11	初版

1 コンポーネントの概要

1.1 TWL-SDKにおけるコンポーネント

TWL-SDK において、コンポーネントとは「ARM7 プロセッサ(以下 ARM7)にて実行されるプログラム」を意味します。アプリケーションは ARM9 プロセッサ(以下 ARM9)にて実行されますので、コンポーネントは主にデバイスの制御を担当します。

ARM9 と ARM7 間の FIFO を介して通信することで ARM9 はコンポーネントの動作を制御します。TWL-SDK では各デバイス毎にコンポーネントを制御する機能をまとめたライブラリ群を提供しておりますので、アプリケーションは直接 FIFO を介した ARM7 との通信を行う必要はありません。

TWL-SDK では複数のコンポーネントを提供しておりますが、アプリケーションを作成する際にどれか一つを選択して頂く必要があります。ARM7 では選択したコンポーネント1つが動作し、アプリケーション実行中に切り替えることはできません。

1.2 制御するデバイス

この節では、コンポーネントが制御するデバイスと、そのデバイスを操作するために提供されている ARM9 側のライブラリについて説明します。

1.2.1 デジタルボタン

TWL ハードウェアでは、X ボタン、Y ボタン、開閉検知ボタンは ARM9 から直接状態を読み出すことができません。このため、これらのボタンの状態監視をコンポーネントが行います。PAD ライブラリによってこれらのボタンの状態読み出し機能が提供されています。

1.2.2 サウンド回路

TWL ハードウェアでは、ARM9 から直接サウンド出力回路にアクセスすることができません。このため、サウンド回路の制御はコンポーネントが行います。ARM9 側のライブラリは SND ライブラリとして提供されています。

1.2.3 SPIデバイス

TWL ハードウェアでは、ARM9 から直接 SPI デバイスにアクセスすることができません。また、複数の SPI デバイスは同時にアクセスすることができません。コンポーネントは複数 SPI デバイスへのアクセス管理を行います。

SPI デバイスには、マイク、タッチパネル、パワーマネジメント IC、及び内蔵フラッシュメモリの 4 つが含まれています。マイクについては、コンポーネントが連続したサンプリングの正確な周期管理を行います。MIC ライブラリによってマイクのサンプリング機能が提供されています。

タッチパネルについては、コンポーネントが連続したサンプリングの周期管理を行います。TP ライブラリによってタッチパネルのサンプリング機能が提供されています。

パワーマネジメント IC によって実現可能なさまざまな機能は、PM ライブラリにまとめられています。

内蔵フラッシュメモリについては、読み書きする機能がコンポーネントに含まれていますが、ARM9 からこの機能を制御する API は用意されておりません。オーナー情報など一部の保存データのみ、OS ライブラリによって読み出しがサポートされています。

1.2.4 リアルタイムクロック

TWL ハードウェアでは、リアルタイムクロックに ARM9 から直接アクセスすることができません。このため、リアルタイムクロックの制御はコンポーネントが行います。RTC ライブラリによってリアルタイムクロックを制御する機能が提供されています。

1.2.5 ワイヤレス通信モジュール

TWL ハードウェアでは、ワイヤレス通信モジュールに ARM9 から直接アクセスすることができません。コンポーネントはワイヤレス通信モジュールの制御や通信プロトコル上必要なタイミング管理、通信相手との自動 V ブランク同期など、ワイヤレス通信に関わるあらゆる制御の根幹部分を担います。ARM9 では NITRO 無線に関しては WM ライブラリ等によってワイヤレス通信を制御する機能が、TWL 無線に関しては TWL-SDK Wi-Fi ライブラリによってインフラストラクチャ通信を制御する機能が提供されています。

1.2.6 カメラモジュール

TWL ハードウェアでは、カメラモジュールに ARM9 から直接アクセスすることができません。このためカメラモジュールの制御はコンポーネントが行います。ただし一部 ARM9 からしかアクセスできない機能もあります。ARM9 側のラ

イブラリは CAMERA ライブラリとして提供されています。

1.2.7 SDCard I/F

TWL ハードウェアでは、SDCard I/F に ARM9 から直接アクセスすることができません。

1.2.8 AESモジュール

TWL ハードウェアでは、AES モジュールに ARM9 から直接アクセスすることができません。このため AES モジュールの制御はコンポーネントが行います。ARM9 側のライブラリは AES ライブラリとして提供されています。

2 コンポーネントの種類

TWL-SDK では、mongoose、ichneumon、mongoose.TWL、raccoonTWL 及び ferret.TWL の 5 種類のコンポーネントを提供しています。以下に各コンポーネントの詳細を示します。

2.1 mongoose

[1.2.1]から[1.2.5]で挙げた 5 つのデバイスの制御を行う NITRO ROM 用のデフォルトコンポーネントです。

ワイヤレス通信に関する機能の実行コードの大半がメインメモリに配置されています。このため、通信に関する処理を行う際には頻繁にメインメモリにアクセスします。ARM7 がメインメモリにアクセス中には ARM9 からのアクセス及び DMA によるアクセスは待たされますので(メインメモリへのアクセス優先権設定がデフォルトである場合)、通信頻度や通信データ量が増えれば増えるほど ARM9 がメインメモリにアクセスできる時間は減少します。接続処理時など連続した特殊な動作を行う場合には、最悪の場合数ピクチャーフレームに渡って ARM9 がほとんどメインメモリにアクセスできないことがあります。

2.2 ichneumon

[1.2.1]から[1.2.5]で挙げた 5 つのデバイスの制御を行います。ワイヤレス通信に関する部分が特殊な NITRO ROM 用コンポーネントです。

mongoose コンポーネントとは異なり、ワイヤレス通信に関する機能の実行コードの大半が VRAM に配置されています。このことにより通信に関する処理を行った場合に ARM7 がメインメモリへアクセスする頻度が mongoose コンポーネントと比べると大幅に少なくなります。

ichneumon コンポーネントを用いている場合には、ワイヤレス通信機能を使用するにあたって、まず VRAM-C もしくは VRAM-D の少なくとも片方を ARM7 に割り当てる必要があります。どちらかの VRAM が ARM7 に割り当てられていない時には、VRAM に割り当てべきプログラムのイメージがメインメモリのシステム予約領域に退避されています。このため、ワイヤレス通信機能を使用しない時には ARM9 はこれらの VRAM を別の用途に使用することができます。

VRAM の割り当て状態を変更することでワイヤレス通信機能の有効・無効を切り換えるためのライブラリとして、無線ドライバ駆動制御(WVR)ライブラリが提供されています。

2.3 mongoose.TWL

[1.2]で挙げた全てのデバイスの制御を行う HYBRID ROM 用のデフォルトコンポーネントです。mongoose コンポーネントを HYBRID ROM 用に拡張したものです。ただし NITRO ハードウェアで動作する場合は、[1.2.1]から[1.2.5]で挙げた 5 つのデバイス制御しか行うことができません。

2.4 racoon.TWL

[1.2]で挙げた全てのデバイスの制御を行う LIMITED ROM 用のデフォルトコンポーネントです。TWL ハードウェア専用コンポーネントであり、NITRO ハードウェアでは動作しません。

2.5 ferret.TWL

[1.2]で挙げた中からワイヤレス通信機能とカメラを除いたデバイスの制御を行う LIMITED ROM 用の特殊なコンポーネントです。TWL ハードウェア専用コンポーネントであり、NITRO ハードウェアでは動作しません。

ワイヤレス通信機能とカメラ機能が含まれていませんので、**raccoon.TWL** コンポーネントに比べてサイズが小さくなります。

3 コンポーネントの選択

ARM9 側実行イメージ(アプリケーション)と ARM7 側実行イメージ(コンポーネント)を一つのバイナリファイルにまとめる `makerom` ツールに、どのコンポーネントを使用するかを指定することができます。

具体的には、`makerom.exe` または `makerom.TWL.exe` に与える `rsf` ファイルの `Arm7` セクションにて指定します。詳しくはリファレンスマニュアルの `makerom` についての説明を参照して下さい。`rsf` ファイルがデフォルトの状態でありコマンドラインビルド環境をお使いの場合は、アプリケーションの `Makefile` 内の `commondefs` 参照前で `DEFAULT_COMP_ARM7` 変数にコンポーネントの種類を明示的に設定することで使用するコンポーネントを選択することができます。何も指定されていない場合は、NITRO ROM なら `mongoose` コンポーネントが、HYBRID ROM なら `mongoose.TWL` コンポーネントが、LIMITED ROM なら `racoon.TWL` コンポーネントがデフォルトで設定されます。

4 コンポーネントの判定

アプリケーションにおいて実行時に `ichneumon` コンポーネントであるかを判定する方法を示します。

1. `PXI_Init` 関数呼び出し後、`PXI_IsArm7CallbackReady(PXI_FIFO_TAG_WVR)` 呼び出しに `TRUE` が返された場合は `ichneumon` コンポーネントです。
2. 1.で `FALSE` が返された場合はその他のコンポーネントです。

5 注意点

コンポーネントの実行には、プログラムそのものが配置されるメモリ空間が必要です。TWL ハードウェアには ARM7 専用ワーク RAM が存在しますが、このワーク RAM だけではプログラムを配置するには不十分です。TWL-SDK では ARM9ARM7 共有ワーク RAM(16KB×2)を ARM7 側に割り当ててプログラムを配置しています。この共有ワーク RAM を ARM9 側に割り当て直さないようにして下さい。

更に、このワーク RAM の割り当てだけではワイヤレス通信の制御プログラムを配置するにはまだ不十分です。TWL-SDK ではメインメモリのシステム予約領域として確保されているメモリ空間を ARM7 側のプログラムを配置する領域として用いています。このシステム予約領域(NITRO モードなら 0x027e0000～0x02800000、TWL モードなら 0x02F80000～0x02FE40000)を別の用途に使用しないようにして下さい。なお、TWL-SDK では NITRO モードなら ARM9 側メモリ空間の 0x027e0000 に、TWL モードなら 0x02FE0000 に、デフォルトでデータ TCM が配置されます。

以上

© 2008 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。