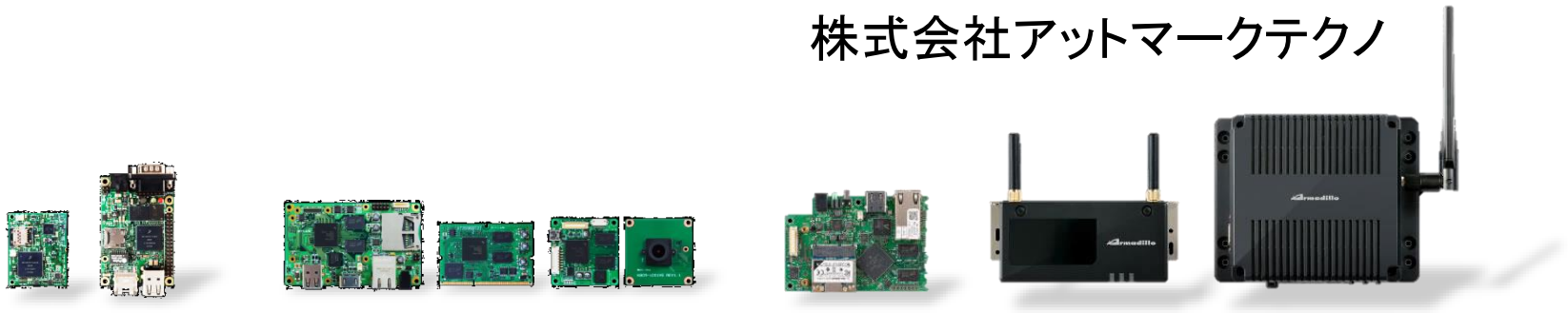
Armadillo-IoT G3/G3L
Armadillo-X1
開発体験セミナー

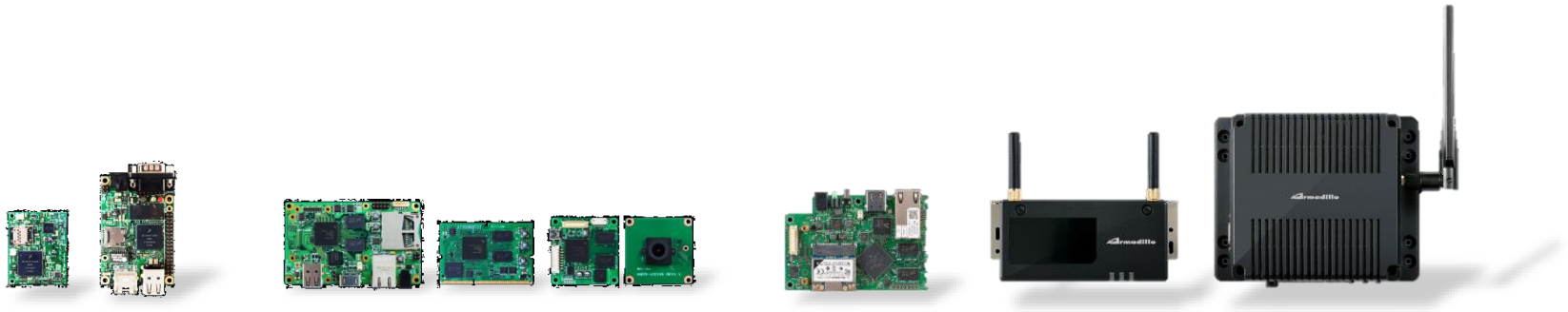
第2部 Armadilloが動作する仕組み

株式会社アットマークテクノ

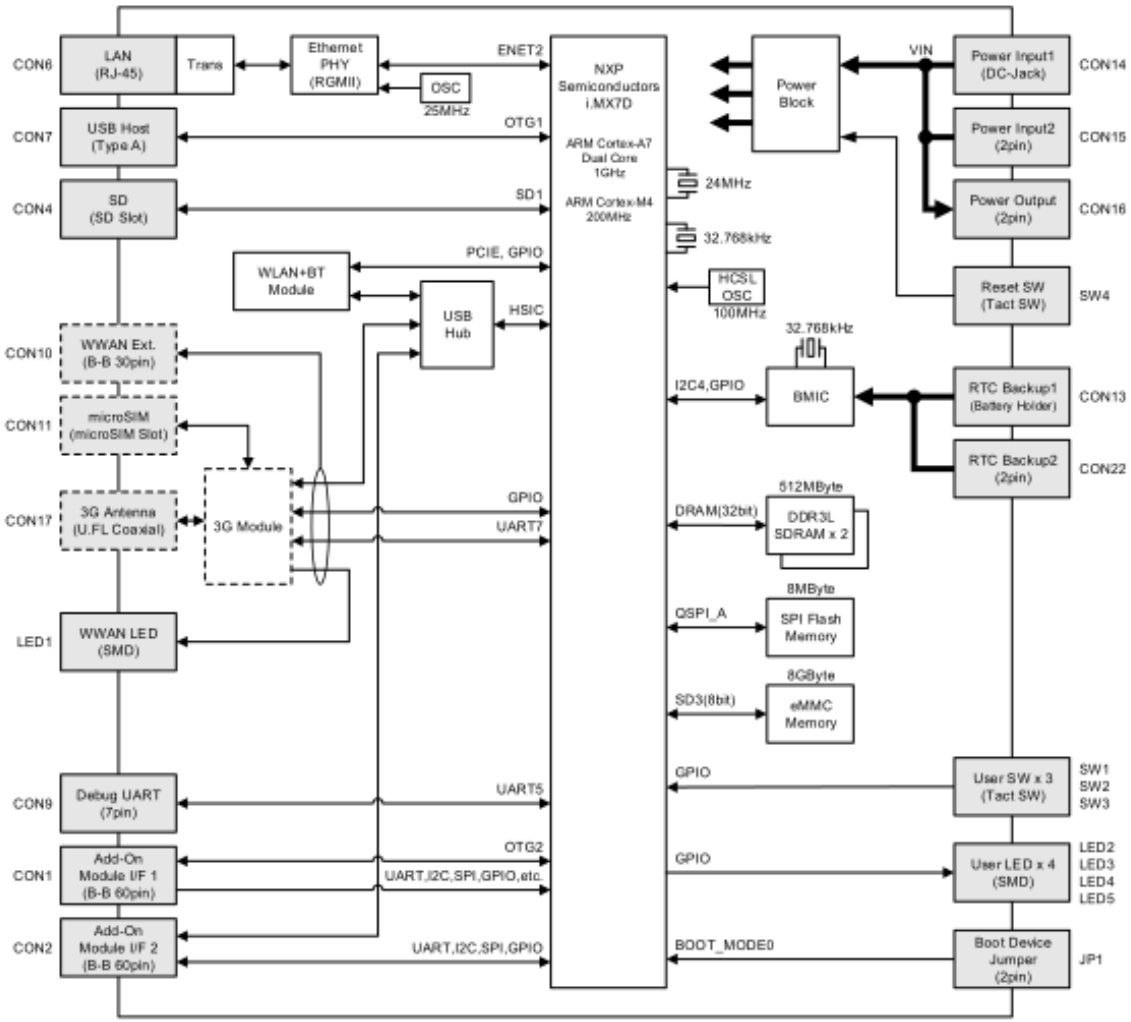


目次

Armadillo-IoT G3/G3L
Armadillo-X1の
ソフトウェア構成

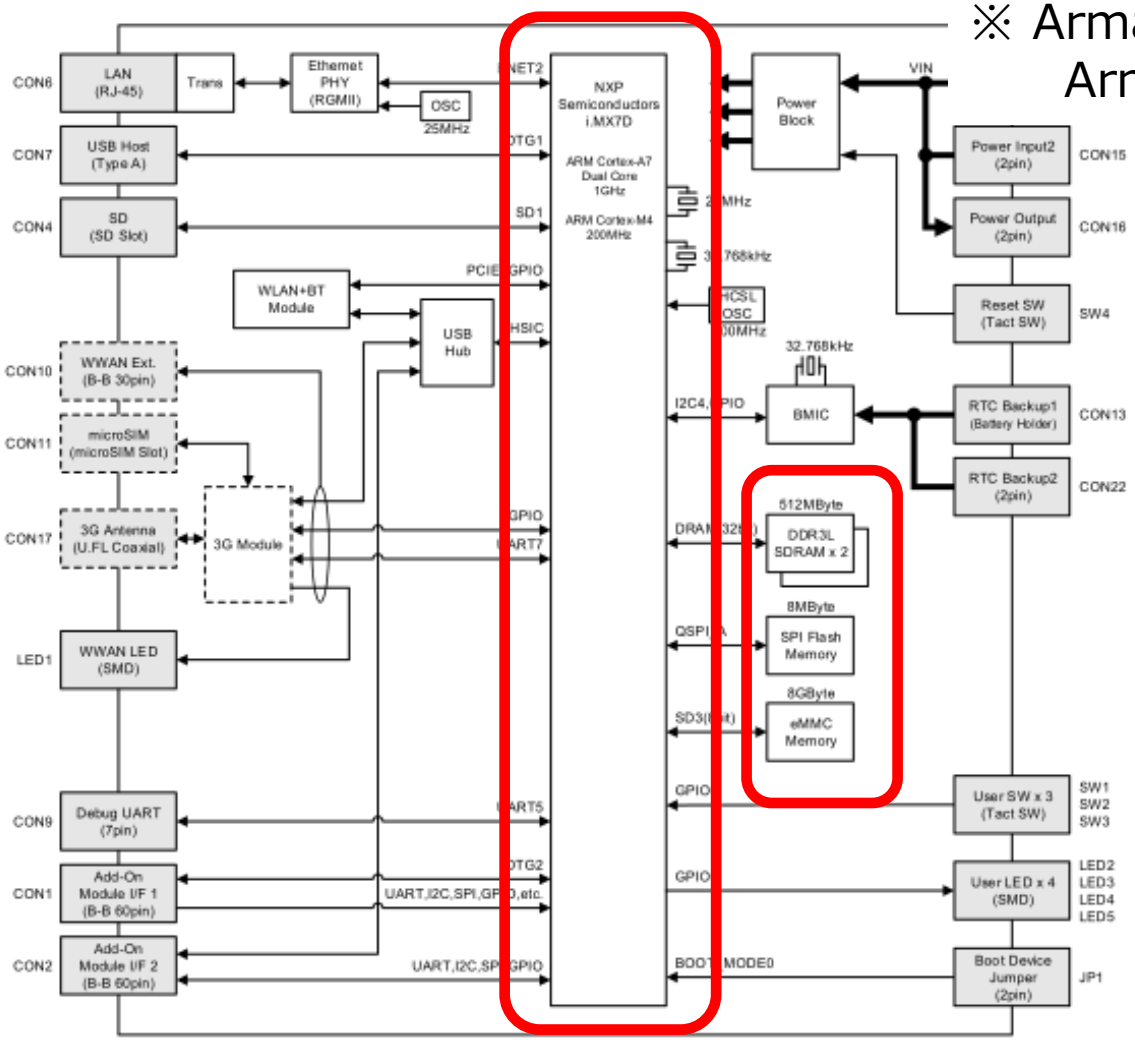


Armadillo-IoT G3のブロック図



Armadillo-IoT G3のコンピューターとしての中心部分

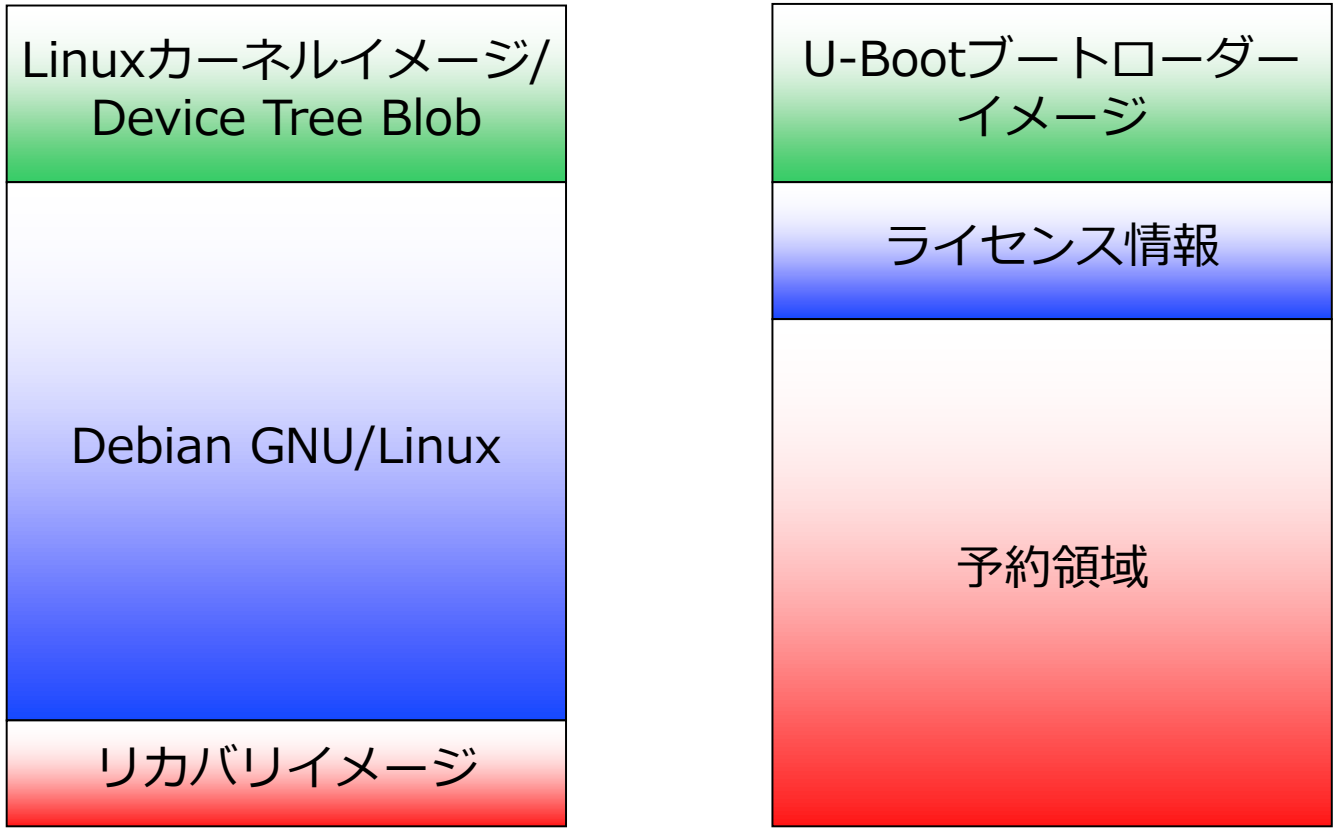
※ Armadillo-IoT G3L
Armadillo-X1も同じ



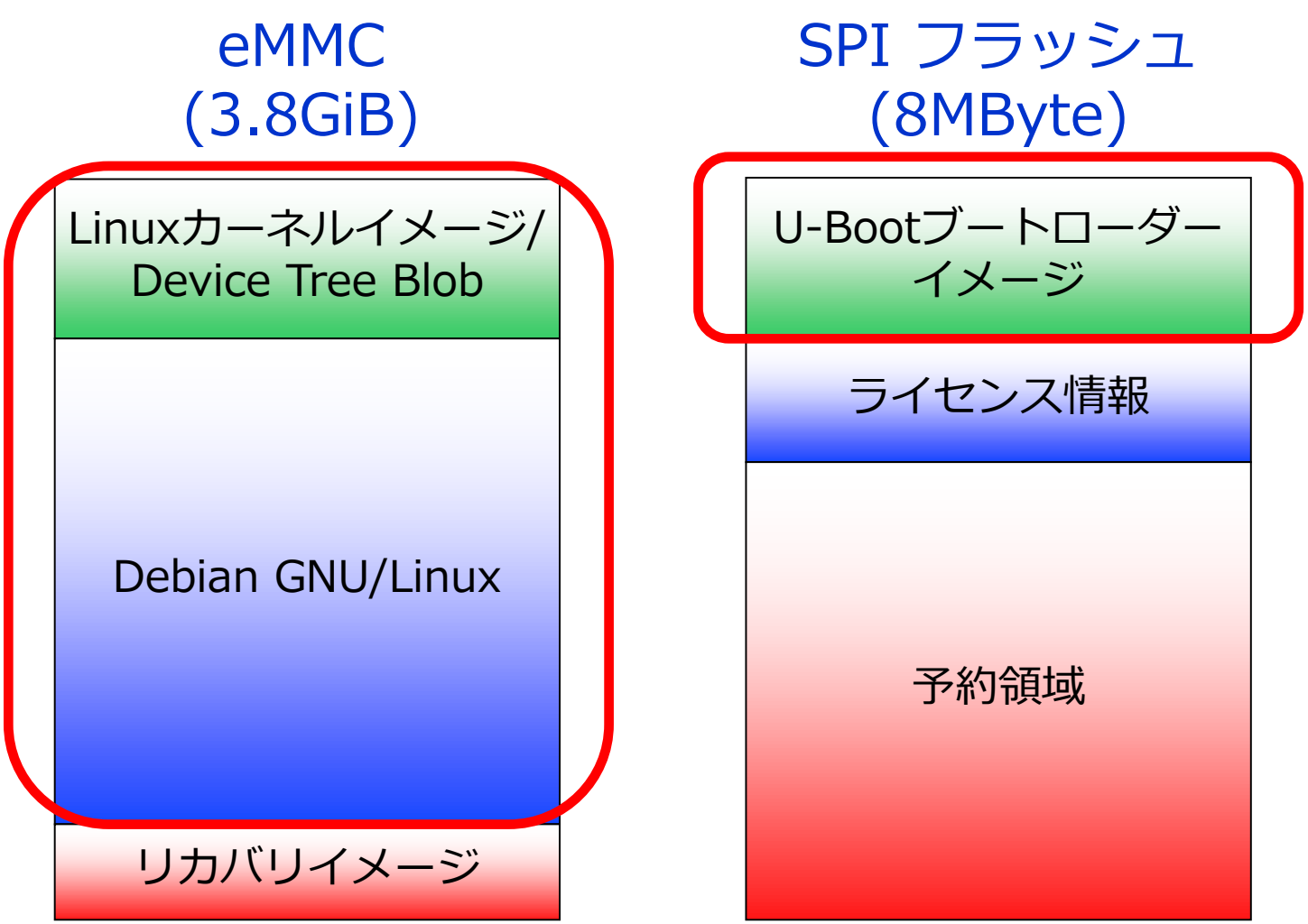
各部の説明

- NXP Semiconductors i.MX7D(CPU)
  - ARMアーキテクチャのCPU
  - メモリに配置されたバイナリを実行
- DDR3L SDRAM x 2
  - 書き換え可能な記憶領域
  - 実行中のプログラムや、ファイルが配置される
- SPI Flash Memory
  - PCでいうとBIOSと似たような役割
  - Linuxを起動させるためのブートローダーが配置される
- eMMC Memory
  - PCでいうとHDD/SSDの役割
  - ルートファイルシステムが配置される

Armadillo-IoT G3のメモリマップ



Armadillo-IoT G3のメモリマップ (ソフトウェア格納場所)



各ソフトウェアの説明

Atmark Techno

- U-Bootブートローダー
  - Linuxシステムを起動するために必要なソフトウェア
  - 電源投入後、最初に起動するソフトウェア
- Linuxカーネル
  - デバイスなどを扱うOSそのもの
  - WindowsのようにGUI等の機能は備えていない
- Device Tree Blob
  - 基板固有の情報が格納されている
- Debian GNU/Linux
  - Linuxシステムを動かすためのライブラリ/アプリケーション/設定等のファイルが配置される
  - ルートファイルシステム・ユーザーランドとも言われる

ブートローダー

- ブートローダー(U-boot)はLinuxカーネルを起動するためのソフトウェアになります
- システムの動作に直接影響がないため、カスタマイズされることは少ないため、アットマークテクノから配布されている物を使用されることも多いです
  - ただしハードウェアに依存する機能が多いため、**部品変更等でバージョンアップが必要となることが比較的多い**ソフトウェアになります
- Armadillo出荷時の物を使用すると自動的に新しいバージョンに変わりますので、システムへの影響が懸念される場合には、**インストールディスクにてバージョンを固定**することをお勧めします

※：ブートローダーの作成方法は、製品マニュアルの「ブートローダーをビルドする」を参照してください

Linuxカーネル/DTB(Device Tree Blob)

- Linuxカーネル、DTB(Device Tree Blob)はOSそのものと、OSを動作させるハードウェア情報です
- 新規ドライバーの有効化や、ハードウェア構成を変更する場合にカスタマイズします
  - Armadillo-IoT G3/G3Lは変更しない場合もあります
  - Armadillo-X1ではハードウェア構成が変化するため、ほぼ変更されます
- システムの基本となるソフトウェアのため、インストールディスクでバージョンを固定することをお勧めします

※：Linuxカーネルの作成方法は、製品マニュアルの「Linux カーネルをビルドする」を参照してください

Device Tree

Atmark Techno

- ARM向けLinuxで使用されているハードウェア固有の情報を記述した設定ファイルのこと
  - ドライバーにハードウェア固有の処理を記述すべきではないという思想
- Armadilloではマルチプレクスの設定や、ドライバーにて扱うするI/Fの設定を行う

- そのため、拡張ボードなどを作成する場合は、ほぼDevice Treeを変更する必要があります

Device Treeの設定

- **Device Treeの設定はLinuxカーネル内部のDTS (Device Tree Source)(拡張子.dtsi, .dts)ファイルに記述します**
  - **「.dtsi」はCPU固有の情報が記述**
  - **「.dts」はボード固有の情報が記述**
    - 内部で「.dtsi」をインクルード
    - 「.dtsi」を基本に、ボード固有の情報を「.dts」で上書き

- **Linuxカーネルをビルド後にDTB(Device Tree Blob)ファイルにコンパイルされます**
  - **実際にArmadilloに書き込むファイル**

Device Treeの設定ファイルの確認

- DTSは以下のパスになります
  - arch/arm/boot/dts/imx7d.dts
  - X1: arch/arm/boot/dts/armadillo_x1.dts
  - G3: arch/arm/boot/dts/armadillo_iotg_g3.dts
  - G3L: arch/arm/boot/dts/armadillo_iotg_g3l.dts

- その他、Armadilloのアドオンモジュール用のDTSは以下のディレクトリに置かれています
  - arch/arm/mach-imx/armadillo_iotg_addon/

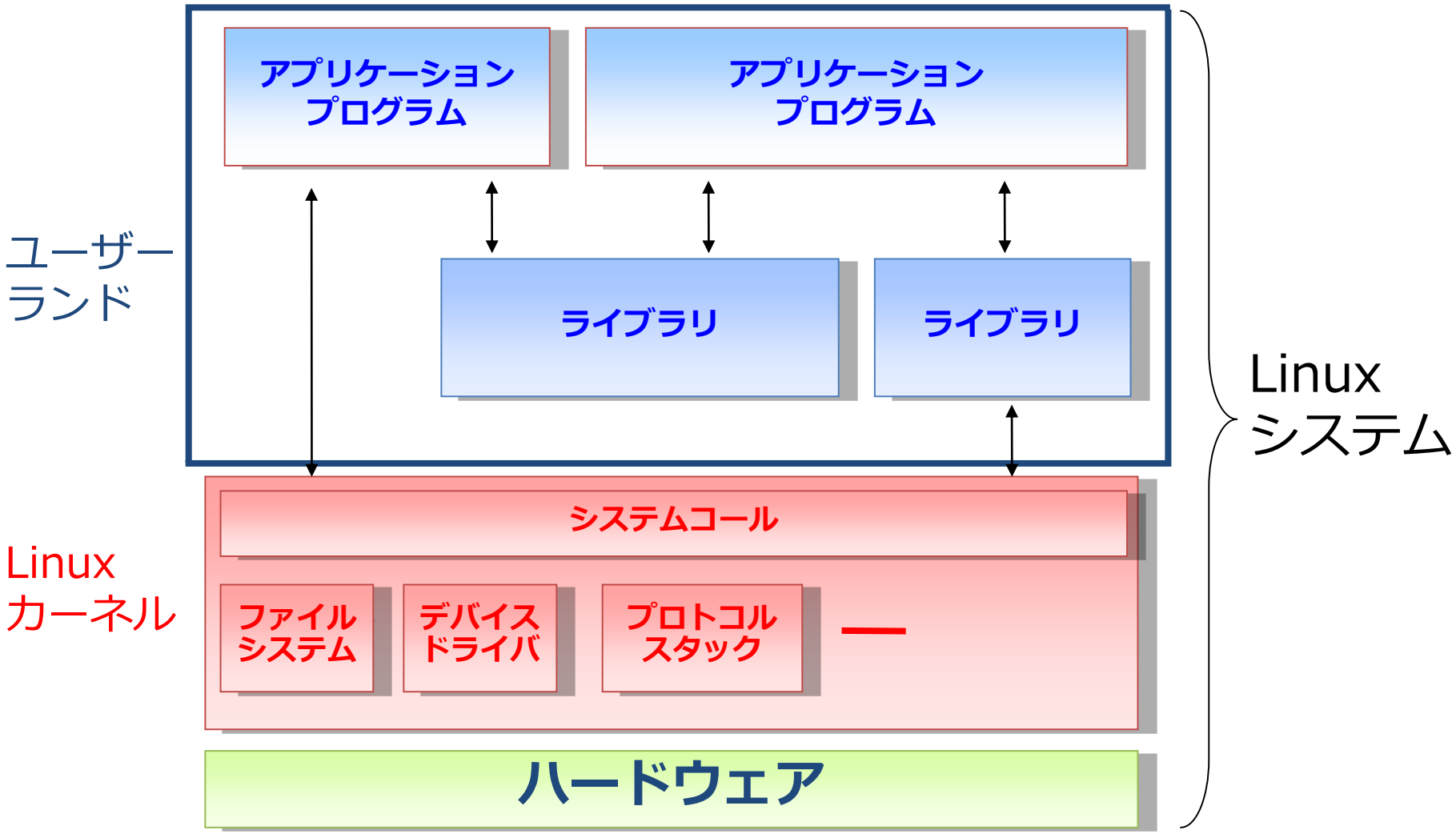
Device Treeの参考情報

- LinuxカーネルのDocumentation/devicetree/にあるファイル
- CPU(i.mx7)のdtsiファイル
  - 基本的な設定はここに記述
- Device Treeの仕様書
  - http://www.devicetree.org/specifications-pdf
- Armadillo-X1製品マニュアルの「拡張インターフェースを使う」
  - 各種インターフェースを使用する場合の設定例

ルートファイルシステム

- ルートファイルシステムは、カスタマイズしたアプリケーションや、ライブラリ、設定等のファイルの集合です
- システムを作る際には、**必ず変更が発生する箇所**になります

Linuxシステムの構成



Linuxディストリビューション

Atmark Techno

- カーネル、ツール類、アプリケーション、ライブラリなどLinuxシステムを構築するのに必要なもの全てを収めたもの
  - PC/サーバー用途
    - **Debian GNU/Linux**・Ubuntu・Fedora・Red Hat Enterprise Linux
  - 組み込み用途
    - MontaVista/Hard Hat Linux・Wind River Linux・emdebian・uCLinux-dist・Atmark Dist

Debian GNU/Linux

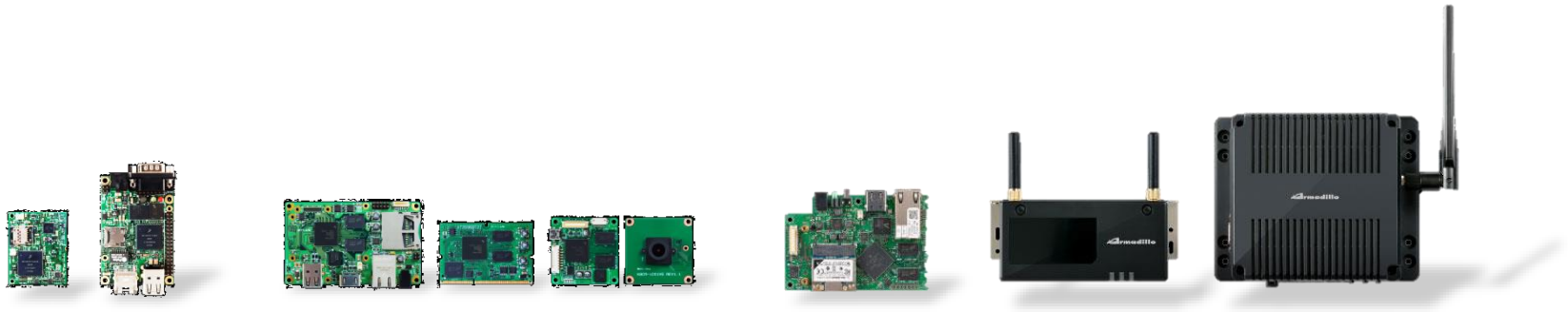
- Linuxをベースとした完全フリーなOS
- ソフトウェアはビルド済みパッケージで管理
  - パッケージは40,000個以上
  - ARM用のパッケージも存在
- Armadillo-IoT G3はDebian GNU/Linux 9.0
  - コードネーム “stretch”
- Armadilloの開発環境でも使用
  - ATDE (Atmark Techno Development Environment)

組み込みLinux

Atmark Techno

- 組み込み機器で動作するLinux
  - デスクトップと異なる部分がある
- デスクトップ上で動作するLinuxとの違い
  - アーキテクチャ(x86/AMD64→ARM)が異なる
    - PC用のアプリケーションは動作しない
  - リソースが少ない
    - RAM/ROMの容量
  - 接続するデバイスが多種多様
    - ドライバーが用意されていないことも

コマンド実行の流れ



組み込みLinuxでのコマンド実行

Atmark Techno

- 組み込みLinuxでは端末からCUIでコマンドを実行することが多い
  - ディスプレイ等がない場合も多いため
- シェルと呼ばれるプログラムに、コマンドを入力することでコマンドを実行する
- ここではコマンド入力から、コマンド実行までの流れを説明します

コマンド実行例(psコマンド)

Atmark Techno

```
[armadillo ~]# ps
  PID  Uid     VmSize Stat Command
    1 root        680 S    init
    2 root            SW<  [kthreadd]
    3 root            SW<  [ksoftirqd/0]
    (中略)
  164 root        420 S <  udevd --daemon
  321 root        484 S    syslogd -L
  354 root        420 S    klogd
 1187 root        264 S    udhcpc -b -p /var/run/udhcpc.e
 1199 root        632 S    inetd
 1223 www-data    932 S    lighttpd -f /etc/lighttpd.conf
 1316 root        924 S    -ash
 1332 root        752 R    ps
```

コマンド実行のフロー

- コマンドをシェルというソフトウェアが受け取る
- 実行すべきファイルを探す
  - コマンドが”/”を含んでいる場合
    - コマンドをパスとして解釈しパスにあるファイル

```
[armadillo ~]# /bin/ls
```

  - コマンドが”/”を含んでいない場合
    - 所定のディレクトリ内にあるコマンド名と同じ名前のファイル

```
[armadillo ~]# ls
```

- いずれも”/bin/ls”が実行される
  - ファイル”ls”は”/bin”に配置されている
  - 環境変数PATHで”/bin/”が指定されている

環境変数PATH

Atmark Techno

- 実行すべきファイルがあるディレクトリのリストが格納されている
  - “:”区切り
- シェルはコマンド名が入力されたら、実行すべきファイルがあるか、PATHで指定されたディレクトリを順番に探す
- Armadilloのrootユーザーの場合
  - /bin・/usr/bin・/sbin・/usr/sbin の順番

```
[armadillo ~]# echo $PATH
/bin:/usr/bin:/sbin:/usr/sbin
```

自作で作成したプログラムを実行するには？

Atmark Techno

- “/”を付けないでコマンド実行する場合には、環境変数PATHに指定したディレクトリに配置する必要があります
- しかし現在のディレクトリにあるファイルを実行したい時もあります
- その場合は、以下のようにファイルを実行します

```
[armadillo ~]# ./command
```

※: “.”は現在のディレクトリを示し、”/”はディレクトリを示します。