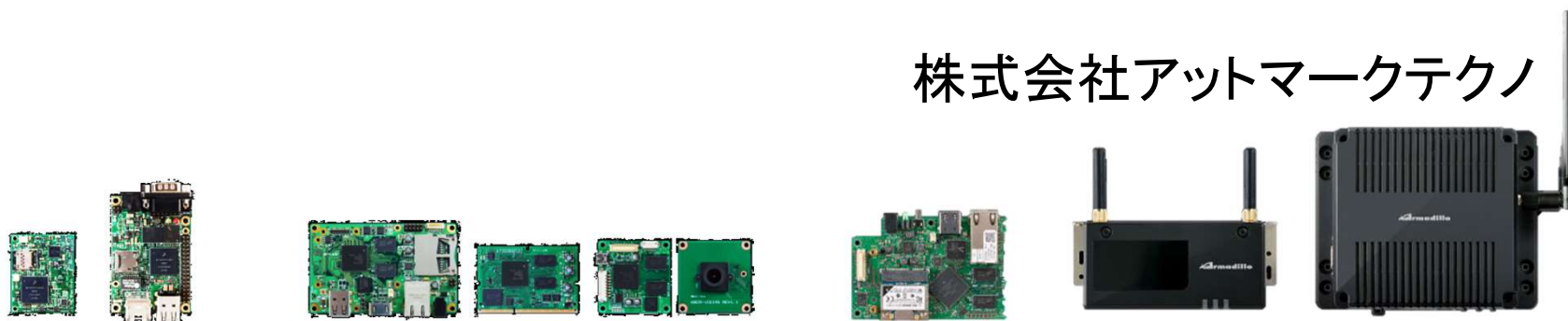


# Armadillo-IoT A6 開発体験セミナー

## 第2部 Armadilloが動作する仕組み

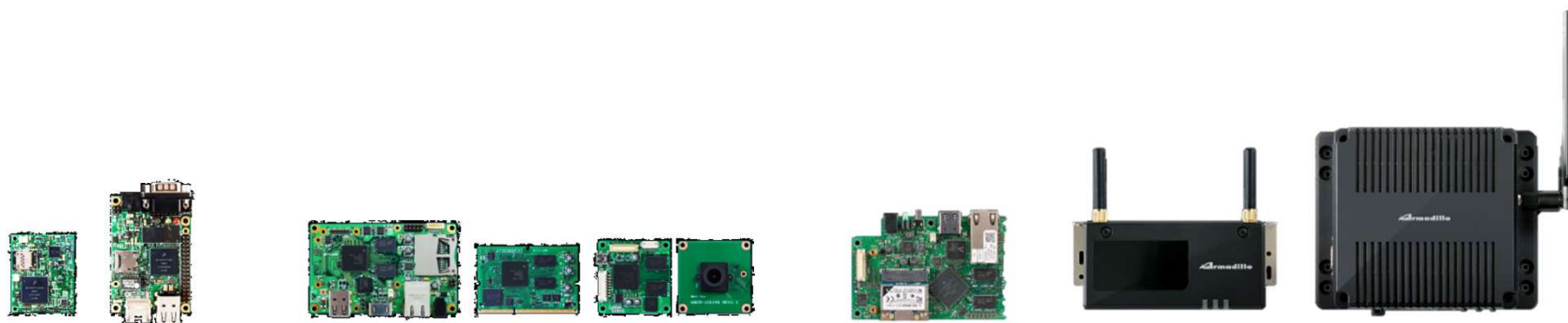
株式会社アットマークテクノ



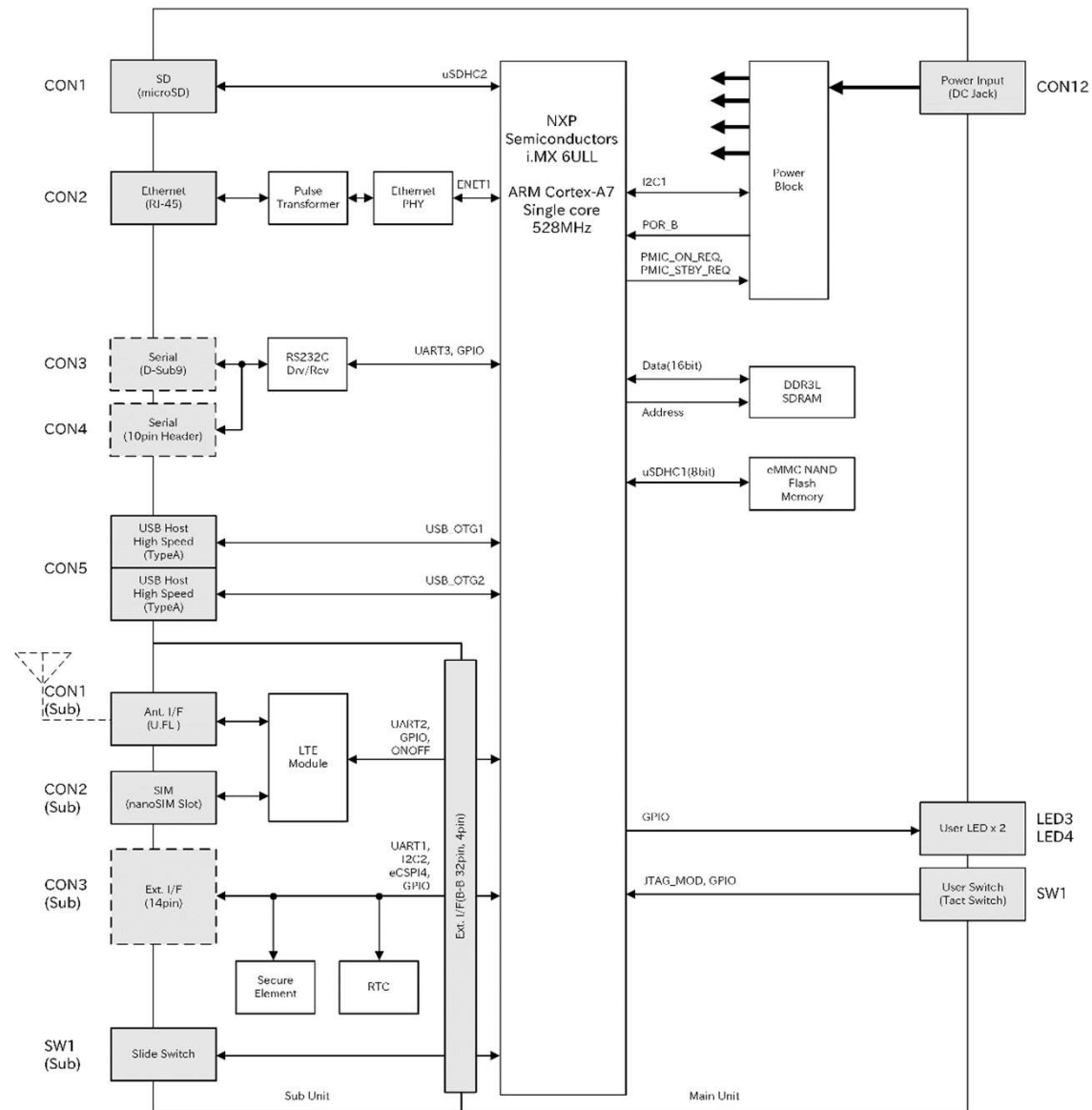
- 
- 第1部 Armadilloとは
  - **第2部 Armadilloが動作する仕組み**
  - 第3部 Armadilloを使用する
  - 第4部 アプリケーションを作成する
  - 第5部 外部機器との連携
  - 第6部 クラウドとの連携
  - 第7部 LTE接続、省電力モードの使用方法
  - 第8部 製品運用に向けての設定
  - 第9部 量産に向けて
  - 第10部 参考情報

# Armadillo-IoT A6の ソフトウェア構成

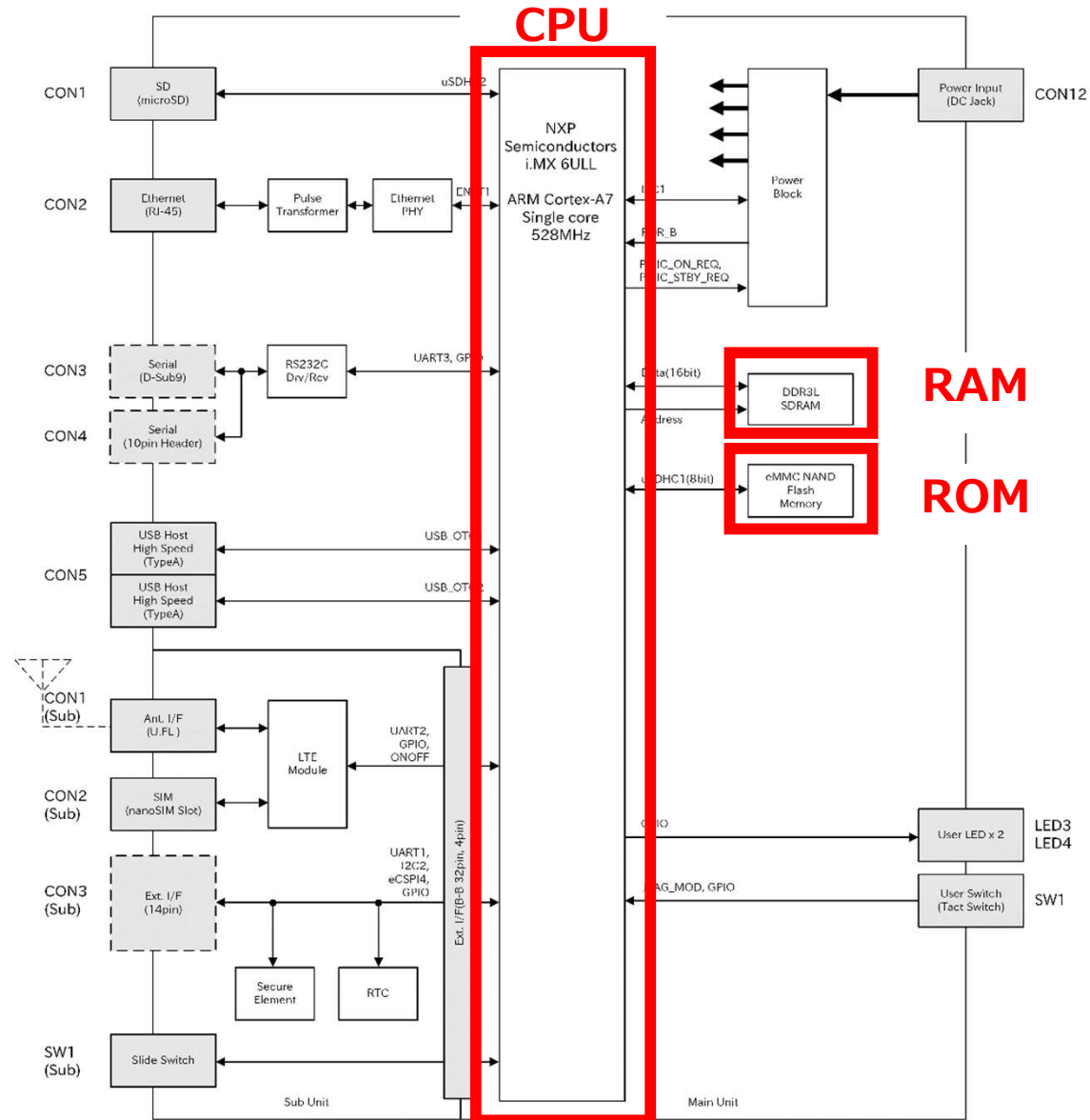
---



# Armadillo-IoT A6のブロック図



# Armadillo-IoT A6のブロック図



---

## ■ CPU: NXP Semiconductors i.MX6ULL

- 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

## ■ RAM: DDR3L SDRAM

- 書き換え可能な記憶領域
- 実行中のプログラム、データが配置される

## ■ ROM: 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

- U-Bootブートローダー（セミナーでは、ブートローダーと呼称）
  - 電源投入後、最初に起動するソフトウェア
  - Linuxシステムを起動するために必要なソフトウェア
  
- Linuxカーネル（セミナーでは、カーネル、と呼称）
  - デバイスなどを扱うOSそのもの
  - カーネルのみではWindowsのようにGUI等の機能は備えていない
  
- Device Tree Blob（セミナーでは、デバイスツリー、と呼称）
  - レジスタの設定等、基板固有の情報
  
- Debian GNU/Linux（セミナーでは、ユーザーランド、と呼称）
  - 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(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 A6(C1モデル)は変更しない場合もあります
  - Armadillo-IoT A6(U1モデル)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

- 
- 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

- 
- 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

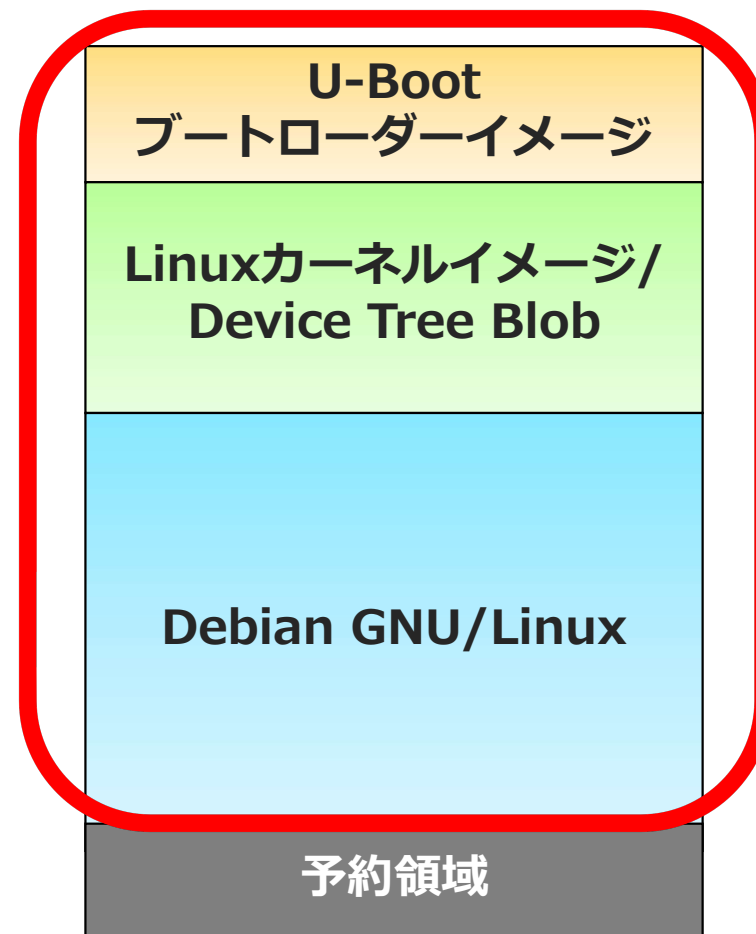
# Armadillo-IoT A6のメモリアップ

eMMC  
(3.8GiB)

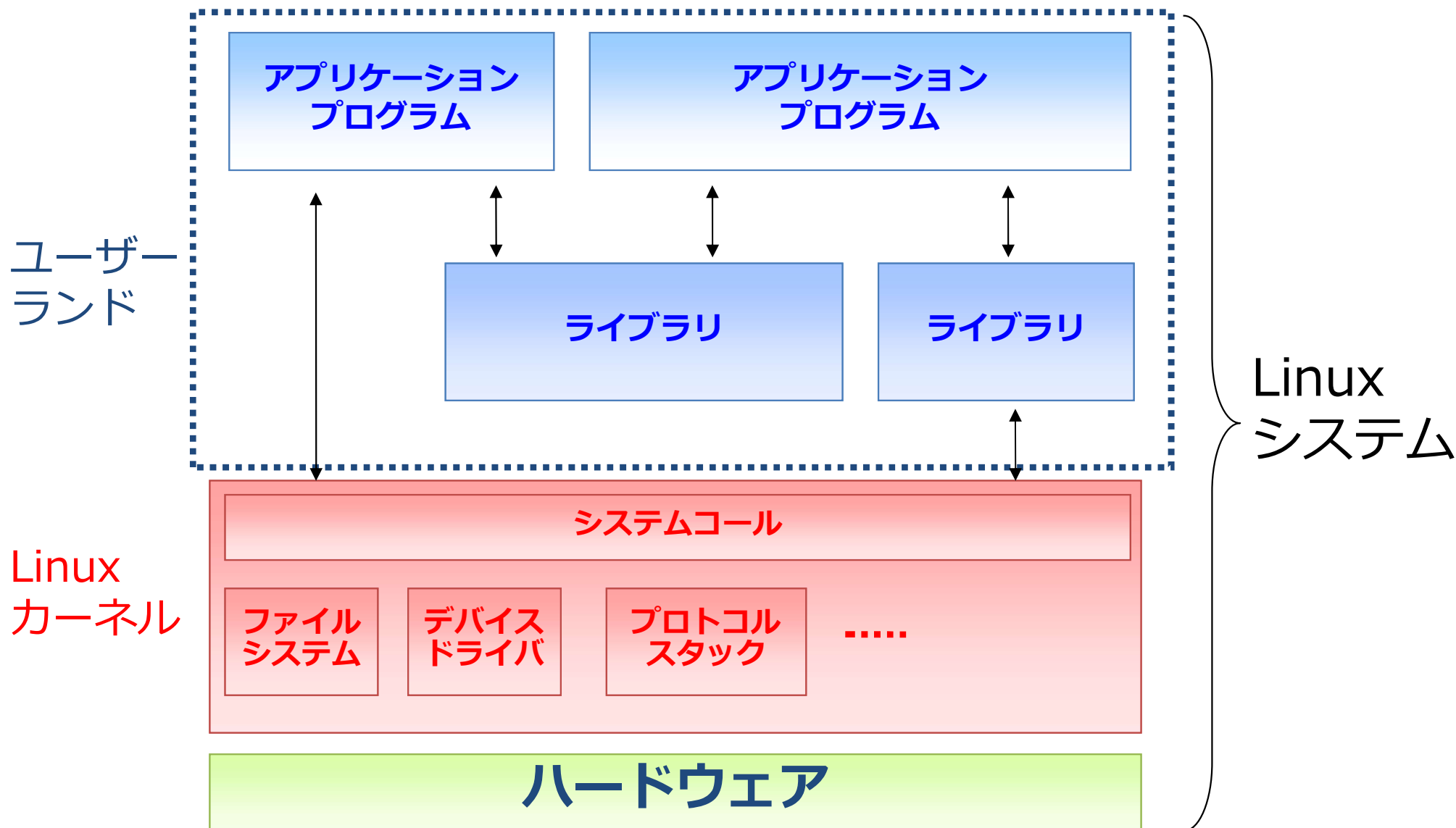


# Armadillo-IoT A6のメモリマップ

eMMC  
(3.8GiB)



# Linuxシステムの構成



- 
- 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

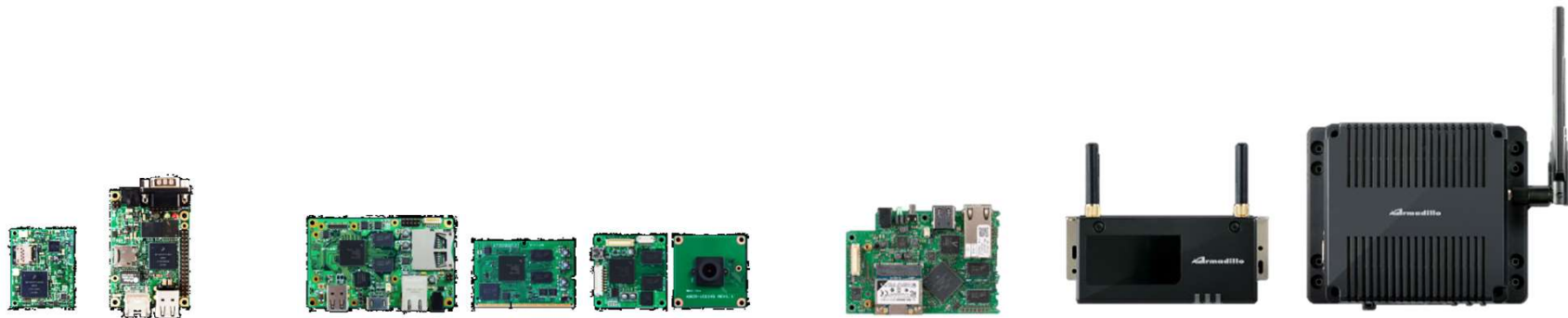
- Linuxをベースとした完全フリーなOS
- ソフトウェアはビルド済みパッケージで管理
  - パッケージは40,000個以上
  - ARM用のパッケージも存在
- Armadillo-IoT A6はDebian GNU/Linux 10.0
  - コードネーム “Buster”
- Armadilloの開発環境でも使用
  - ATDE (Atmark Techno Development Environment)



- 
- **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)

```
[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

- 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？

- “/”を付けずに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、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