

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

## 第3部 Armadilloを使用する

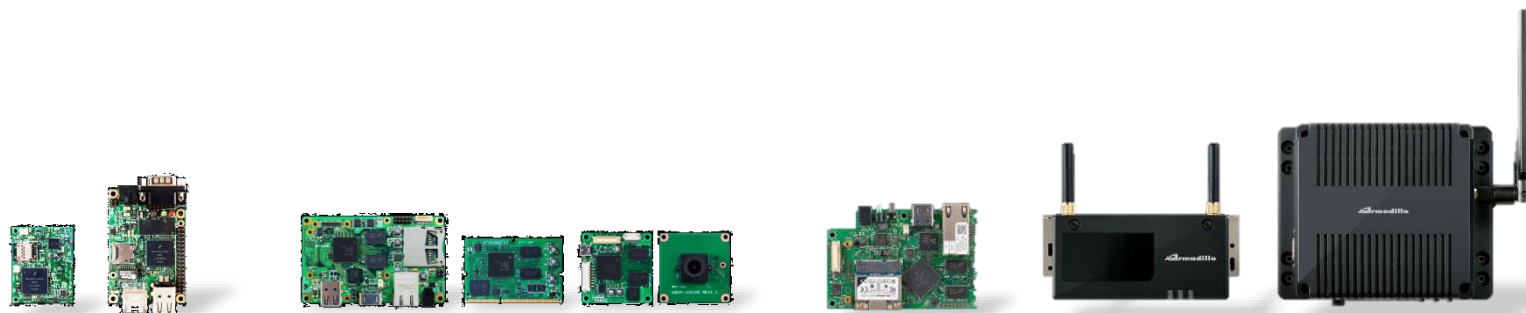
株式会社アットマークテクノ



- 
- 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 製品運用に向けての設定
  - 第8部 量産に向けて
  - 第9部 参考情報

# Armadilloの接続、起動

---

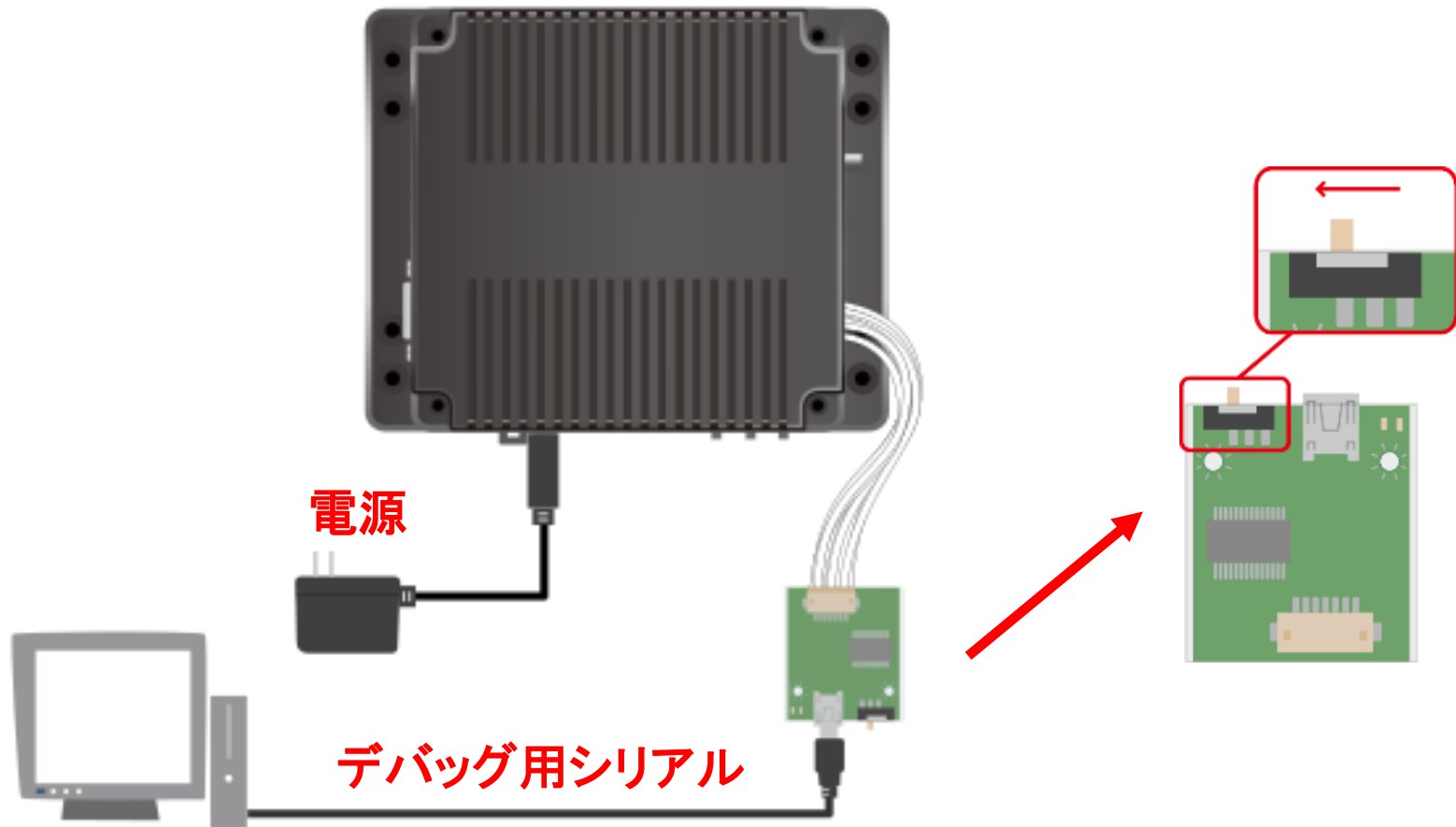


# Armadilloの接続、起動

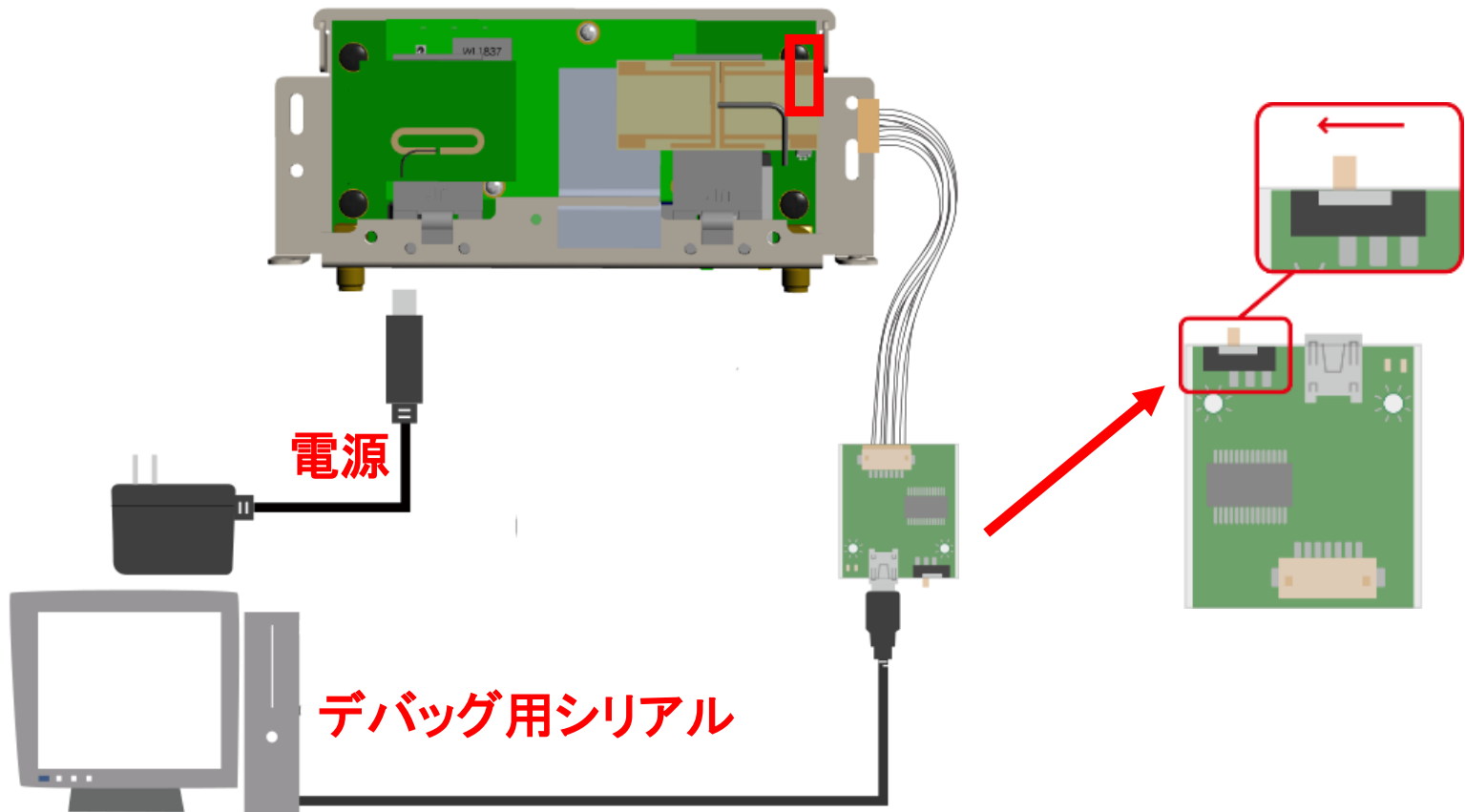
---

- まずはArmadilloを扱うために環境設定を行います
- 以下の設定を行います
  1. Armadilloを接続
  2. コンソールの接続
  3. 電源の投入
  4. Armadilloの起動
  5. ログイン

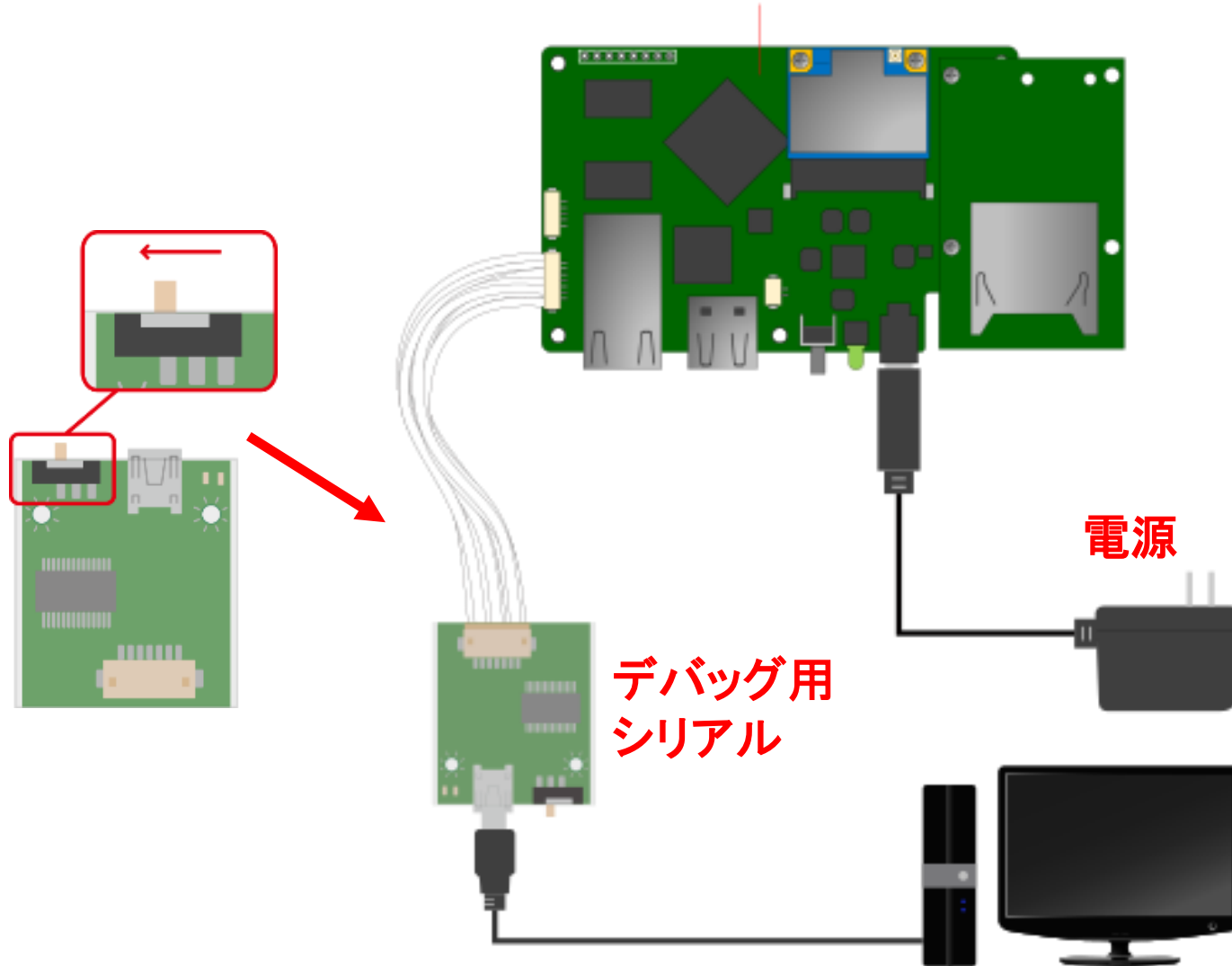
# Armadilloを接続(G3)



# Armadilloを接続(G3L)



# Armadilloを接続(X1)



# コンソールの接続

- TeraTermを起動
- シリアルポートに接続



環境によってシリアルポートの  
番号が変わります



# コンソールの接続

- メニュー「設定」-「シリアルポート」の設定

Tera Term: シリアルポート 設定

ポート(P): COM3

ボー・レート(B): 115200

データ(D): 8 bit

パリティ(A): none

ストップ(S): 1 bit

フロー制御(F): none

送信遅延

0 ミリ秒/字(C) 0 ミリ秒/行(L)

OK

キャンセル

ヘルプ(H)

# 電源投入

- 電源を投入してください  
(ACアダプタをコンセントに接続)
- TeraTermに以下の文字が表示されることを確認して下さい

```
U-Boot 2014.04-at1 (Mar 22 2016 - 16:50:23)
```

```
CPU: Freescale i.MX7D rev1.1 at 792 MHz
```

```
(途中省略)
```

```
=>
```

※:バージョンは違っていても問題ありません

# Armadilloの起動

- 「=>」という表示は、ブートローダーのプロンプトです
- Armadillo-IoT G3のブートローダーはU-Bootです
- Linux システムを起動するには、以下のように「boot」コマンドを実行してください

```
=> boot
```

# 起動メッセージ

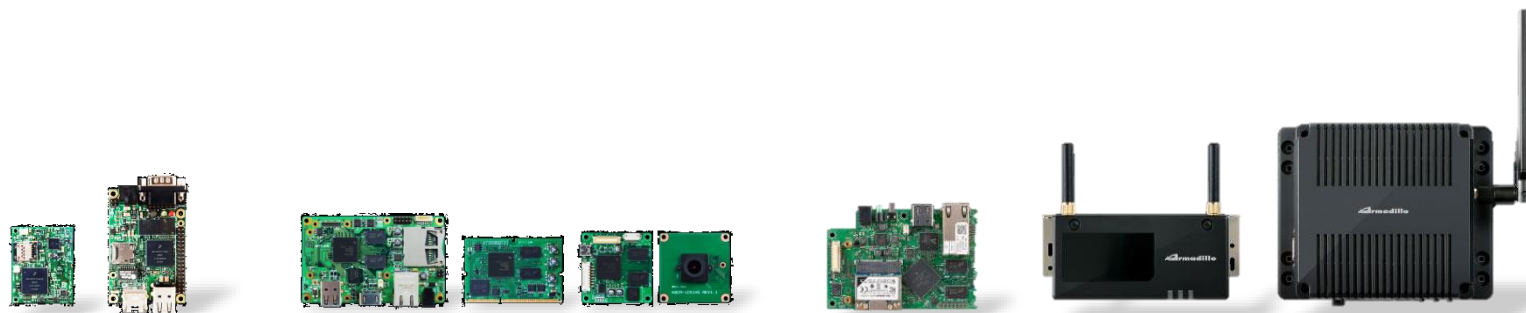
- Linuxが起動すると、以下のように起動時のメッセージが表示されます

```
mmc1(part 0) is current device
mmc1(part 0) is current device
reading boot.scr
** Unable to read file boot.scr **
reading uImage
9331688 bytes read in 229 ms (38.9 MiB/s)
Booting from mmc ...
(省略)
Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 3.14.38-at2 (atmark@atde6) (gcc version 4.9.2 ( 4.9.2-10) ) #15
SMP PREEMPT Thu Sep 15 10:29:57 JST 2016
CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7), cr=10c53c7d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
(続く)
```

# Armadilloでコマンド実行

---



# ログイン

- しばらくするとログインプロンプトが表示されます
- 以下のようにユーザー名「root」、パスワード「root」でログインしてください

```
Debian GNU/Linux 8 armadillo-iotg ttymsc4
```

```
armadillo-iotg login: root // “root” と入力後改行
```

```
Password: root // “root” と入力後改行 (表示されない)
```

# プロンプト/コマンドの入力

## ・標準のプロンプト

入力するコマンドは  
黄色の文字で表記

```
root@armadillo-iotg:~# command
```

ユーザー名

ホスト名

rootユーザーの場合#  
一般ユーザーの場合\$

カレントディレクトリ  
~ はホームディレクトリを意味する

- 実際の表示

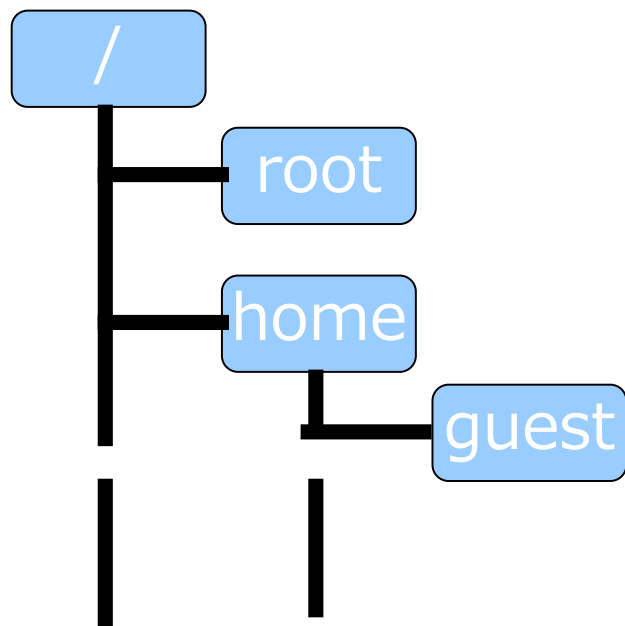
```
root@armadillo-iotg:~# pwd  
/root
```

- 簡略表示

```
[armadillo ~]# pwd  
/root
```



# 参考: ルートファイルシステム



パス	説明
/	ルートディレクトリ
/root	rootユーザーのホームディレクトリ
/home	ユーザー用のディレクトリ
/home/guest	guestユーザーのホームディレクトリ

## ■ ディレクトリとファイルの階層的な構造

- ディレクトリツリー
- ディレクトリ=フォルダ

## ■ “/”から始まる単一のツリー構造

- ドライブという概念はない

# 参考: Armadilloのディレクトリ構造

- 実行ファイル
  - /bin・/usr/bin・/sbin・/usr/sbin
- ライブラリファイル
  - /lib・/usr/lib
- ホームディレクトリ
  - /root・/home/(username)
- デバイスファイル
  - /dev
- 設定ファイル
  - /etc
- プロセス・システムの状態
  - /proc・/sys
- ログファイル
  - /var/log/messages

※: 主にFilesystem Hierarchy Standard (FHS、ファイルシステム階層標準) に従っている。

# ファイル操作等、よく使うコマンド

コマンド	説明
pwd	現在のディレクトリを表示
cd [dir]	ディレクトリの移動
mkdir <dir>	ディレクトリの作成
rmdir <dir>	空のディレクトリを削除
ls [dir]	指定したディレクトリの内容を表示
echo <文字列>	指定した文字列や変数の中身を表示
cp <from> <to>	ファイルまたはディレクトリをコピー
mv <from> <to>	ファイルまたはディレクトリを移動
rm <file>	ファイルまたはディレクトリを削除

- 各コマンドには、オプション等のマニュアルが用意されています
- マニュアルはmanコマンドを実行することで、確認できます
- 以下のコマンドを実行し、pwdコマンドの説明を確認してみてください

```
[armadillo ~]# man pwd
```

# コマンド実行例

- 以下の黄色の文字で記載されたコマンドを実行してみてください

```
[armadillo ~]# ls
[armadillo ~]# pwd
/root
[armadillo ~]# cd /
[armadillo /]# ls
bin  dev  home  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  lib  media      opt  root  sbin  sys  usr
```

# 温度データを取得するコマンド実行

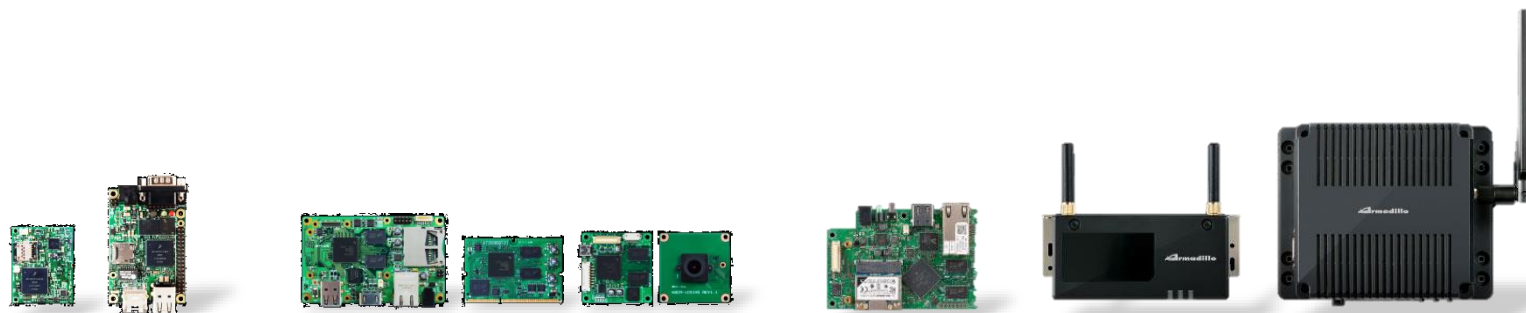
- コマンド例として、CPU内部の温度データを取得してみます
- 以下のコマンドを実行し、CPUの温度データを取得してください

```
[armadillo ~]# cat /sys/class/thermal/thermal_zone1/temp  
50000
```

- 上記の場合、温度に変換すると「50°C」になります

# アプリケーションの追加

---



- ArmadilloにはDebian GNU/Linuxが搭載されています
- Debian GNU/Linuxにはソフトウェアのビルド済みパッケージが用意されています
- ここではそのパッケージをArmadilloにインストールします



- Debianでパッケージをインストールする場合、標準ではインターネットからパッケージをダウンロードしてインストールします
- そのため、ここでは以下の内容を実施します
  - インターネットへの接続
  - パッケージリストの更新
  - パッケージの検索
  - パッケージのインストール

- 
- Armadillo-IoT G3開発セットが接続できるネットワーク
    - 有線LAN
    - 無線LAN
    - 3Gネットワーク
  - このセミナーでは、実習のために無線LANに接続します
  - Armadillo-IoT G3のネットワーク設定はnmcliコマンドを使って行います

# nmcliでネットワークの設定を行う

- 以下のコマンドを実行して、無線LANアクセスポイントに接続してください

```
[armadillo ~]# nmcli device wifi connect [ssid] ¥  
> password [passphrase]
```

※: [ssid]とpassphraseは、別資料に記載の設定に置き換えてください

※: “[と]”は入力不要です。

# IPアドレスの確認

- 以下のコマンドを実行し、IPアドレスが設定されていることを確認して下さい

```
[armadillo ~]# ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr 44:c3:06:30:c3:6b
           inet addr:192.168.11.105  Bcast:192.168.11.255  Mask:255.255.255.0
           inet6 addr: fe80::46c3:6ff:fe30:c36b/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:2197 errors:0 dropped:0 overruns:0 frame:0
           TX packets:497 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:389015 (379.8 KiB)  TX bytes:73601 (71.8 KiB)
```

IPアドレス

# インターネットへの接続確認

- 以下のコマンドを実行し、インターネットに接続されていることを確認してください

```
[armadillo ~]# ping 8.8.8.8
PING 8.8.8.8: 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=49 time=59.8 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=49 time=85.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=49 time=77.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=49 time=78.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=49 time=77.8 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 59.8/75.8/85.4 ms
```

※: pingコマンドを終了するにはCtrl+Cキーを押下してください

# パッケージリストの更新

- Debianパッケージはインターネット上のサーバーからインストールします
- パッケージがバージョンアップしている場合もありますので、まずはパッケージリストを更新しましょう
- 以下のコマンドを実行して、パッケージリストを更新してください

```
[armadillo ~]# apt-get update
```

# パッケージを検索してみる

- パッケージリストを更新すると、Armadillo内にパッケージの情報がダウンロードされます
- どのようなパッケージがあるかの検索は「apt-cache search」コマンドで実施できます
- 例として、以下のコマンドを実行し、coreutilsパッケージがあるか確認してください

```
[armadillo ~]# apt-cache search coreutils
```

- 
- Armadilloでは、ファイル編集はテキストエディタを使用します
  - 標準の状態ではviエディタが入っています。
  - ここでは、より簡単に使用できるnanoエディタをインストールします



# nanoエディタのインストール

---

- 以下のコマンドを実行して、nanoエディタをインストールしてください

```
[armadillo ~]# apt-get install nano
```

# nanoエディタの使用方法

---

- nanoエディタがインストールできたら、試しに起動してみましよう
- 以下のコマンドを実行し、nanoエディタを起動してください

```
[armadillo ~]# nano
```

# nanoエディタの使用方法

```
GNU nano 2.2.6          New Buffer

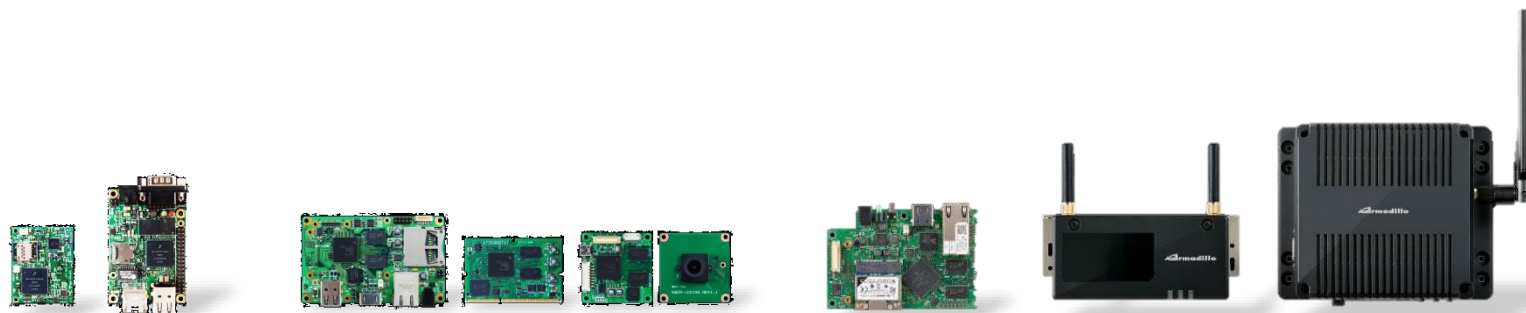
} ヘルプ

^G Get Help      ^C WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

- **^X (Ctrl + x キー): 終了**

# シェルスクリプトの作成

---



# シェルスクリプトを作成

---

- シェルが解釈し実行できるスクリプト
- 以下のコマンドを実行してシェルスクリプトを記述するファイルを開いて下さい

```
[armadillo ~]# nano gettemp
```

# シェルスクリプトを作成

- さきほどのコマンド実行で開いたgettempファイルを以下のように編集して保存・終了してください

```
#!/bin/sh  
  
cat /sys/class/thermal/thermal_zone1/temp
```

# シェルスクリプトの実行

- 以下のようにコマンドを実行してgettempに実行権限を追加してください

```
[armadillo /etc/config]# chmod +x gettemp
```

- 以下のようにコマンドを実行して、シェルスクリプトを実行してください

```
[armadillo /etc/config]# ./gettemp
```