

Armadillo-IoT A6 開発体験セミナー

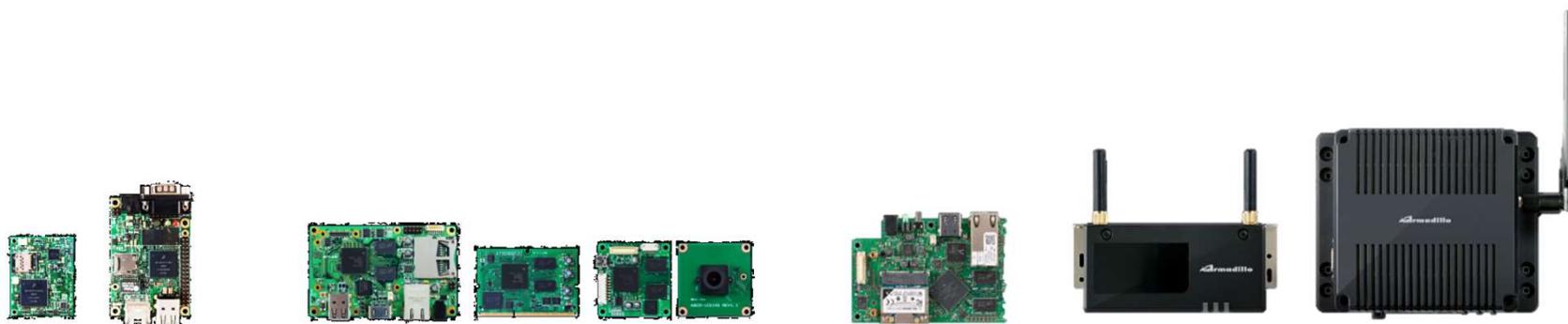
第4部 アプリケーションを作成する

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 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

アプリケーション開発環境



Armadillo-IoT A6 アプリケーション開発環境

- Armadillo-IoT A6のアプリケーション開発は Armadillo上で行います
 - 組み込み開発でよくあるクロス開発ではなく、セルフ開発が基本となります

- Armadillo-IoT A6では各種言語が使用可能
 - C/C++
 - JAVA
 - Ruby, Python等

-
- Armadilloでは、ファイル編集はエディタを使用します。
 - 殆どのLinuxシステムに標準でviエディタがインストールされています。
 - Windowsで一般的なエディタ(メモ帳など)とは異なり、モードを持っている事が大きな特徴です。
 - コマンドモード：入力した文字は全てコマンドとして扱います
 - 入力モード：通常のエディタと同様に、文字の入力ができます

viエディタ基本操作

- Vi エディタ"test"というファイルを開きます。

```
root@armadillo:~# vi test
```

- 起動直後はコマンドモードです。
- 入力モードに切り替える為には以下のいずれかのキーを入力します。

入力キー	動作
i	カーソルのある位置から入力開始
a	カーソルのある後ろから入力開始

- 再度コマンドモードに戻る際は「Esc」キーを入力します。
- コマンドモード時に、「x」でカーソル上の文字、「dd」コマンドでカーソルのある行を削除できます。
- 終了したい場合は、以下のいずれかのコマンドを実行します。

コマンド	動作
:q!	変更を保存せずに終了
:w [ファイル名]	ファイル名を指定して保存
:wq	ファイルを上書き保存して終了

【参考情報】 nanoエディタの使用 方法1

- 普段メモ帳等を使用している方は、viエディタは慣れるまで操作が難しいかと思います。
- メモ帳に近い感覚で使用できるnanoエディタもインストールして使用することができます。

```
root@armadillo:~# apt-get install nano
```

- ご自身が使いやすいエディタをご利用ください。



【参考情報】 nanoエディタの使用 方法2

- nanoエディタでtestというファイルを作成してみます

```
root@armadillo:~# nano test
```

- エディタ画面の末尾の2行はヘルプです。

```
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos  
^X Exit      ^R Read File  ^¥ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

-  ^X (Ctrl + x キー)で終了を選択します。
- Saveの確認では、保存したい場合は y キーを選択します。

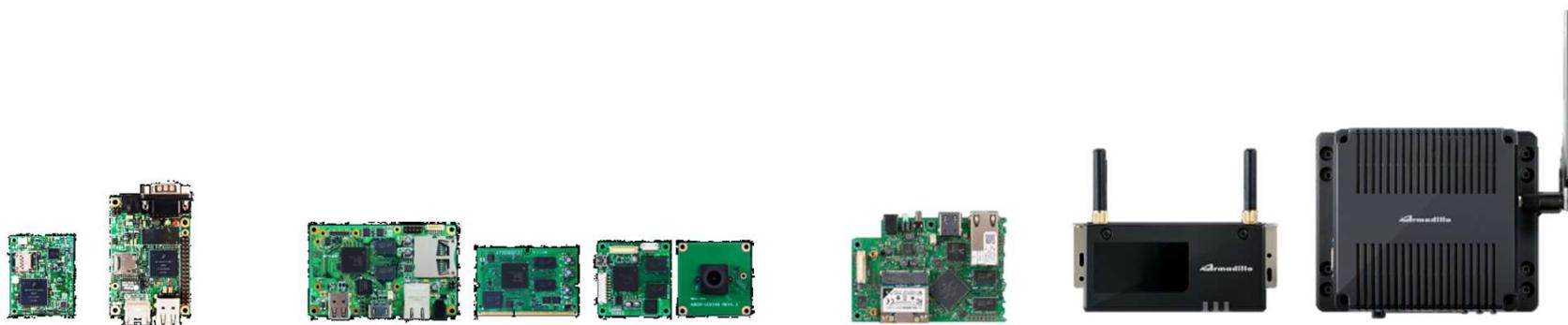
```
Save modified buffer? (Answering "No" will DISCARD changes.)  
Y Yes  
N No          ^C Cancel
```

- ファイル名の確認では、Enter キーを入力します。

```
File Name to Write: temp
```

以上で、最初に指定した"test"というファイルが作成されます。

C言語でHello Worldサンプル の作成、実行



-
- **まずは組み込み開発で最も使用されているCで、サンプル (Hello World)を作成してみましよう**

 - **以下の流れで作成します**
 - C言語開発環境の構築
 - サンプルソースコードの作成
 - Makefileの作成
 - ビルド
 - サンプルの実行

- C言語の開発には、ソースコードを元に実行ファイルを作成するための、ツールチェーンが必要になります
- 以下のコマンドを実行し、ツールチェーンをインストールしてください

```
[armadillo ~]# apt-get install build-essential
```

※すでにインストールされている場合は不要です

サンプルソースコードの作成

- 以下のコマンドを実行して、hello.cを開いてください

```
[armadillo ~]# nano hello.c
```

- 以下のように編集してファイルを保存・終了してください

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Hello World¥n");
    return 0;
}
```

Makefileの作成

- 以下のコマンドを実行してMakefileを開いてください

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

- 以下のように編集してファイルを保存・終了してください

```
all: hello  
  
hello: hello.c  
    gcc -o hello hello.c
```

必ずタブでインデントすること

- 以下のコマンドを実行し、hello.cをビルドしてhelloを作成してください

```
[armadillo ~]# make  
[armadillo ~]# ls hello  
hello
```

- 参考のため、以下のコマンドを実行し、helloがARM用に作成されていることを確認してください

```
[armadillo ~]# file hello  
hello: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically  
linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 2.6.32,  
BuildID[sha1]=ef14f3cd7b8c10ddf6ff6498e3d7c7dc962c68e2, not stripped
```

ARMになっていることを確認

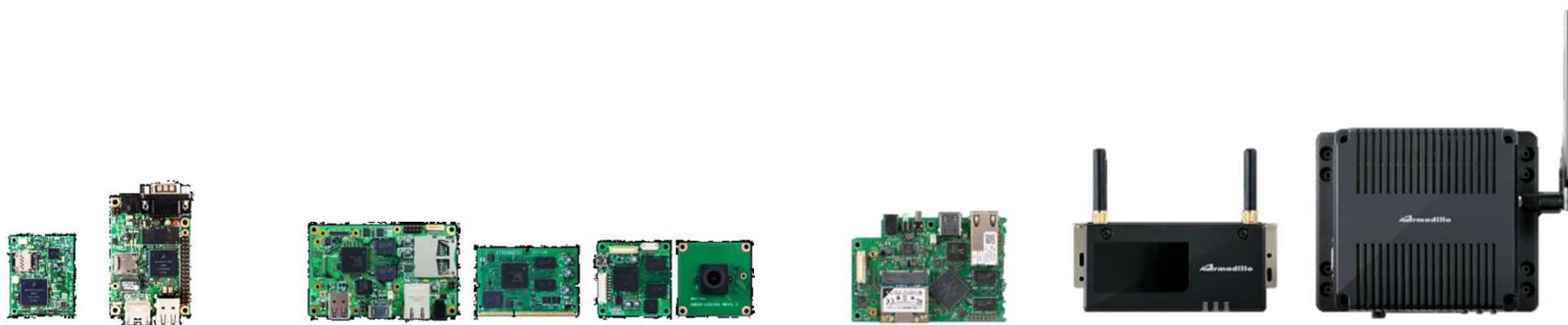


サンプルの実行

- 以下のコマンドを実行し、helloを実行してください

```
[armadillo ~]# ./hello  
Hello World
```

PythonでHello Worldサンプルの作成、実行



PythonでHello World

-
- Pythonは、Webアプリケーションでよく使われるスクリプト言語
 - 例としてPythonを使用
 - 作業内容
 - Python開発環境の構築
 - サンプルソースコードの作成
 - サンプルの実行

- 以下のコマンドを実行して、Python実行環境 (Python3)をインストールしてください

```
[armadillo ~]# apt-get install python3
```

※すでにインストールされている場合は不要です

サンプルソースコードの作成

- 以下のコマンドを実行して、hello.py ファイルを開いてください

```
[armadillo ~]# nano hello.py
```

- 以下のように編集して、ファイルを保存・終了してください

```
print("Hello World")
```

サンプルの実行

- 以下のコマンドを実行し、helloを実行してください

```
[armadillo ~]# python3 hello.py  
Hello World
```