

Armadillo-640 開発基礎セミナー

第3部 Armadilloを動かしてみる

株式会社アットマークテクノ

http://www.atmark-techno.com/





- 第1部 Armadilloについて
- 第2部 Armadilloが動作する仕組み
- 第3部 Armadilloを動かしてみる
- 第4部 開発環境の構築
- 第5部 アプリケーションの作成
- 第6部 拡張I/Fを使った開発の流れ その1) ハードウェア、カーネル編 その2) アプリケーション編
- 第7部 イメージの作成
- 第8部 イメージの書き込み
- 第9部 製品運用に向けて
- 第10部 参考情報



この章の概要

- ArmadilloとPCの接続
- 電源投入
- ログイン
- コマンド演習



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-640とPCを接続









コンソールの接続

TeraTermを起動シリアルポートに接続

Tera Term: 新しい接続		×
O TCP/ <u>I</u> P	ホスト(<u>T</u>): 172.16.0.13 ビビストリ(<u>O</u>) サービス: O Telnet の <u>S</u> SH SSHバージョン(<u>V</u>): SSH2 O その他 プロトコル(<u>C</u>): UNSPEC	>
● <u>57771/(E)</u>	ポート(<u>R</u>): COMB: USB Serial Port (COMB)	~
	OK キャンセル ヘルプ(H)	



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):	СОМВ	~	ОК		
ボー•レート(B):	115200	~			
データ(D):	8 bit	\sim	キャンセル		
バリティ(A):	none	\sim			
ストップ(s):	1 bit	\sim	ヘルプ(H)		
フロー制御(F):	none	\sim			
送信遅延 0 ミリ秒/字(C) 0 ミリ秒/行(L)					



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 2018.03-at1 (Apr 04 2018 - 12:12:16 +0900)

CPU: Freescale i.MX6ULL rev1.0 at 396 MHz

(略)

= >



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-640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。

 \Rightarrow 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。

3380744 bytes read in 128 ms (25.2 MiB/s) 23426 bytes read in 54 ms (422.9 KiB/s) ## Booting kernel from Legacy Image at 82000000 Image Name: Linux-4.14-at2 Image Type: ARM Linux Kernel Image (uncompressed) Data Size: 3380680 Bytes = 3.2 MiB Load Address: 82000000 Entry Point: 82000000 Verifying Checksum ... OK ## Flattened Device Tree blob at 83000000 Booting using the fdt blob at 0x83000000 Loading Kernel Image ... OK Loading Device Tree to 9eeff000, end 9ef07b81 ... OK Starting kernel (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 9 armadillo ttymxc0

armadillo login: root // "root"と入力後改行 Password: root // "root"と入力後改行(表示されない)







シェル (コマンドを実行するソフトウェア)

- Linuxでは、コンソールからコマンドを実行した場合、シェルと呼ばれるソフトウェアが入力された文字を解釈し、ソフトウェアを実行します。
- Windowsのコマンドプロンプトのようなもの
- スクリプト言語としても使用できます。
- シェルにはいくつかの種類があります。
 - bash
 - zsh
 - dash



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></dir>	ディレクトリの移動
mkdir <dir></dir>	ディレクトリの作成
rmdir <dir></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></to></from>	ファイルまたはディレクトリをコピー
<pre>mv <from> <to></to></from></pre>	ファイルまたはディレクトリを移動
rm <file></file>	ファイルまたはディレクトリを削除
cat <file></file>	ファイルの内容を表示



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。

echo: "hello"という文字列を"temp"というファイルに出力

root@armadillo:~# echo hello > temp

ls: (引数無しで)現在のディレクトリ内のファイル、ディレクトリを表示

root@armadillo:~# <mark>|s</mark> temp

cat: "temp"ファイルの中身を表示

root@armadillo:~# cat temp
hello

rm: "temp"ファイルを削除

root@armadillo:~# rm temp



コマンドのマニュアル

- 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。

root@armadillo:~# man pwd



TIPS)入力の省略

コマンドやパスの補完(TABキー)



■ コマンド履歴(上下キー)

root@armadillo:~# ↑+root@armadillo:~# Is



TIPS)リダイレクト

- 「>」リダイレクト
 - 標準出力(コンソール)以外に出力を切り替える機能
 - 以下の例はファイル名"log"にhello1を書込みしています

root@armadillo:~# echo hello1 > log

■ 以下の例はファイル名"log"にhello2を追記しています

root@armadillo:~# echo hello2 >> log

上記コマンドを実行した結果は以下のようになります

root@armadillo:~# cat log hello1 hello2



参考)ルートファイルシステム



- 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

プロセス・システムの状態

/proc · /sys

■ 設定ファイル

• /etc

ログファイル

/var/log

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