

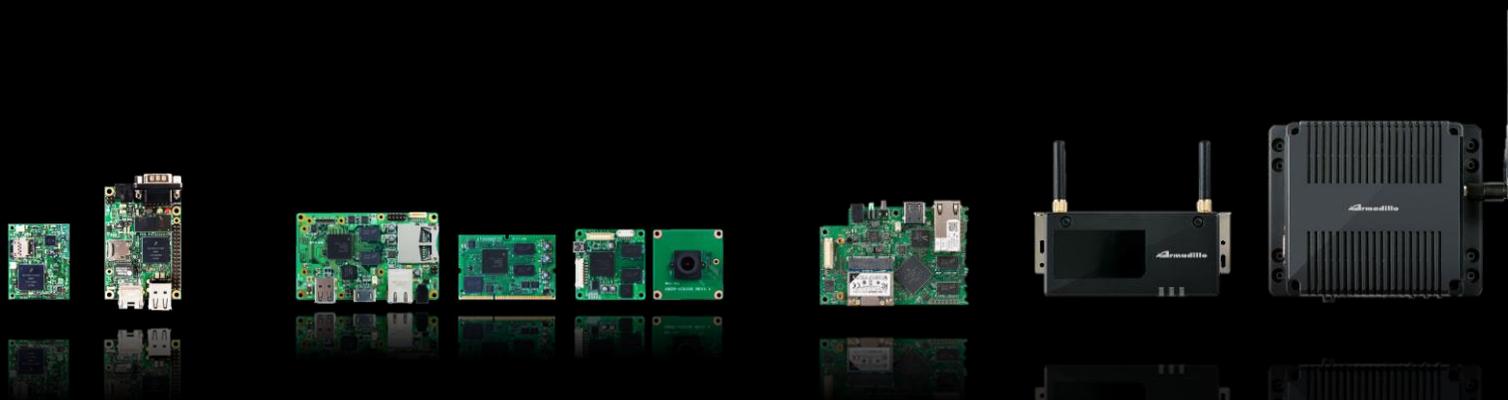


Armadillo-IoT G4 発売記念！ Edge AI作品開発コンテスト meet up！

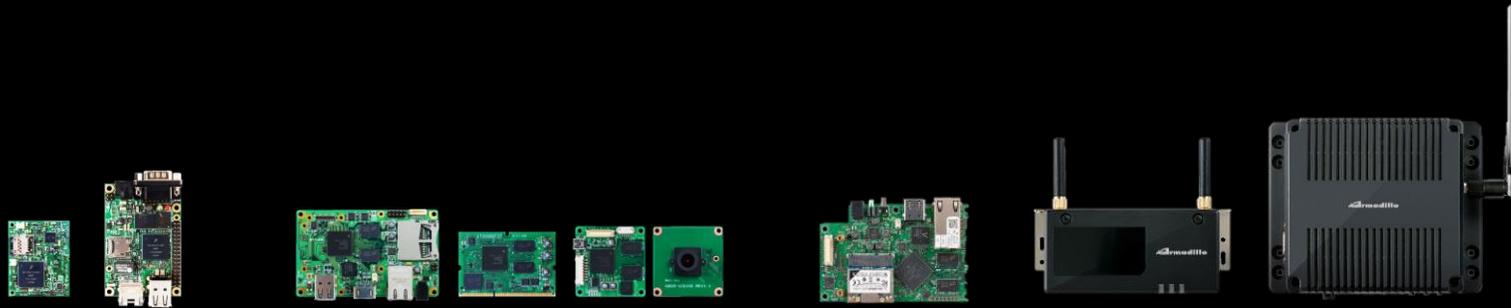
コンテストに向けて

株式会社アットマークテクノ

www.atmark-techno.com

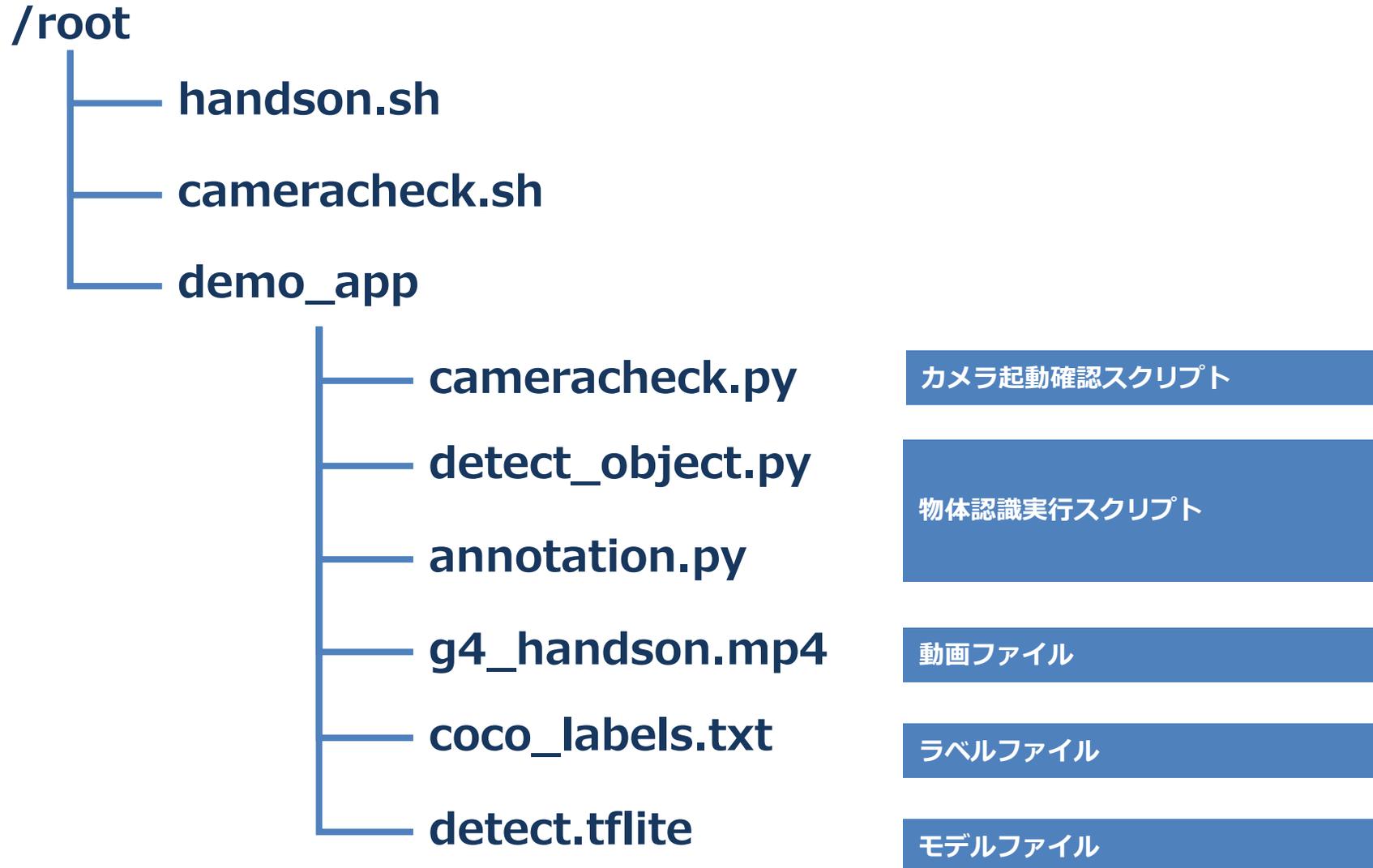


コンテストに向けて



ハンズオン使用コンテナ詳細

物体認識コンテナ仕様



物体認識コンテナ仕様

handson.sh パラメータ

```
#!/bin/bash

app="/root/demo_app/detect_object.py"   実行するスクリプト
model="/root/demo_app/detect.tflite"    モデルファイル
label="/root/demo_app/coco_labels.txt"  ラベルファイル
address="192.168.100.20" } Azureコンテナのアドレス・ポート番号
port=1111
interval=5                             データ送信間隔 [秒]
threshold=0.45                          物体認識スコア閾値
cameraid=3                              カメラデバイスID
(省略)
```

ラベル・モデルファイルの差し替え

■ handson.sh の下記を修正

```
model="/root/demo_app/detect.tflite"      モデルファイル  
label="/root/demo_app/coco_labels.txt"     ラベルファイル
```

■ 転移学習については後述の開発ガイドを参照してください

Azure コンテナとの連携

■ handson.sh の下記の設定の通り動作

```
address="192.168.100.20"
```

```
port=1111
```

```
interval=5
```



Azureコンテナのアドレス・ポート番号

データ送信間隔 [秒]

Azure コンテナとの連携

- IoT Central のデバイステンプレートで定義したテレメトリの形式で Azure コンテナへデータ送信

Most detected object	detectObject	Telemetry	なし	×	▽
Total number of detected objects	totalNumDetectObject	Telemetry	なし	×	▽

表示名	名前*	スキーマ*	
object name	objectName	String	×
number of detections	numObject	Integer	×
+ 追加			



```
detect_object.py send_socket()
```

```
telemetry_data = {"totalNumDetectObject": total,  
                  "detectObject": {"objectName": label, "numObject": num}}
```

Azure コンテナとの連携

- Azure IoT への送信部分を Azure コンテナに任せず、物体認識コンテナに Azure IoT device SDK for Python をインストールして 1つのコンテナだけで実現するパターンもあります

```
root@f6107fd74be7:/# pip3 install azure-iot-device
```

Azureコンテナ仕様

Azure-IoT-samples

└─ Armadillo-IoT_GW



物体認識 コンテナとの連携

- 物体認識コンテナから送信されてきた json データをそのまま Azure IoT に送信

例

```
{"totalNumDetectObject": 6, "detectObject": {"objectName": "tv", "numObject": 2}}
```

Azure IoT Hubへの接続

- Azure IoT Hub への接続も可能です。
Azure Portal の DPS 及び IoT Hub の設定画面から下記を取得、
g4_basic_config.json に設定してください。
 - ◆ IDスコープ
 - ◆ デバイス主キー

Writable property / コマンドについて

- 配布したコンテナのソースでは未対応です。
 - ◆ サンプルを作成しました。詳細はREADMEに記載しています。

■ ダウンロードURL

https://download.atmark-techno.com/sample/algyan/220108_g4_meetup/src/iotc_writable_property/

ドキュメント

■ Armadillo-IoT ゲートウェイ G4 製品マニュアル

<https://armadillo.atmark-techno.com/resources/documents/armadillo-iot-g4/manuals>

記載内容

- ◆ Armadillo-IoT ゲートウェイ G4の使い方
- ◆ 製品仕様(ソフトウェアおよびハードウェア)
- ◆ オリジナルの製品を開発するために必要となる情報
- ◆ 開発に有用な情報
- ◆ その他注意事項

- Armadillo Base OSを搭載した製品を用いたシステムを設計・開発する方に向けた開発ガイドを提供しています

- Armadillo Base OS 開発ガイド

<https://armadillo.atmark-techno.com/resources/documents/armadillo-iot-g4/manuals>

記載内容

- ◆ Armadillo Base OS 上での開発手法
 - podman イメージ及びコンテナの作成と運用
 - ATDE を用いた各種開発手法
- ◆ 機械学習による物体検出アプリケーションを作る際の手法
 - TensorFlowを用いた既存の物体検出モデルの転移学習の手法
 - TensorFlowのモデルを TensorFlowLiteのTFLite形式のモデルに変換する方法
 - システムへの物体検出モデル組み込み

開発

Atmark Techno Development Environment(ATDE)

- Armadilloの開発環境として提供
- VMWare など仮想マシン向けのデータイメージ
- Linux デスクトップ環境をベースに、GNUクロス開発ツール等必要なツールを事前にインストール



開発用PCの用意やツールのインストールなどの開発環境を整える手間を軽減することができます

- ATDEアーカイブはArmadillo サイトから取得可能です

<http://armadillo.atmark-techno.com/>

コンテナに付与する権限

- コンテナを podman run コマンドで作成する際、権限やデバイスへのアクセス許可をオプションで指定します。

例) `--device=/dev/video3:/dev/video3`

- **--privileged** オプションを使用するとコンテナに全権限と全てのデバイスへのアクセスを許可することができます。コンテナに与える最小の権限を洗い出す必要がないため開発時に有用です。



全権限と全てのデバイスへのアクセスを許可するとセキュリティー上問題があるため運用時にはコンテナに最低限の権限のみ付与し、**--privileged** オプションは開発時にのみご利用ください。

製品マニュアル逆引き

- コンテナ操作方法を知りたい
 - ◆ 9.1. アプリケーションをコンテナで実行する
- デバイスを扱いたい
 - ◆ 9.1.4. 入出力デバイスを扱う

GPIO, I2C, SPI, CAN, PWM, シリアル, USB, RTC
音声出力, ユーザースイッチ, LED

- 近距離通信デバイスを扱いたい
 - ◆ 9.1.5. 近距離通信を行う

Bluetooth, Wi-SUN, EnOcean
※ 別途デバイスが必要

製品マニュアル逆引き

■ サーバーを動かしたい

◆ 9.1.7. サーバーを構築する

HTTP, FTP, Samba, SQL

■ タッチパネルを使いたい

◆ 9.1.9. 画面表示を行う

■ 色々試してみたい

◆ 9.10. デモアプリケーションを実行する

mediaplayer, video recoder, led switch tester, rtc tester
object detection demo

運用

コンテナ・アプリケーションの自動実行

- Armadillo の起動時に、作成した podman コンテナ及びコンテナの自動実行を行うことができます。
 - ◆ マニュアル 9.2.1. コンテナの自動起動
 - ◆ 開発ガイド 6.5.6. podman コンテナとアプリケーションの自動実行

コンテナの連携

- ハンズオンの手順のほか、 pod 機能を使用することでも実現できます。

Pod : 複数の関連するコンテナをまとめて構成する機能

- ◆ マニュアル 9.2.2. pod の作成、9.2.3. network の作成

ファイルの永続化

- ハンズオンではファイルの永続化に `persist_file` コマンドを使用しましたが、OSのアップデートで削除されてしまいます。
- これを避けるため、所定のファイルに保存したいファイル名を列挙する必要があります。
 - ◆ マニュアル 9.7.5.1 `swupdate_preserve_files` について

コンテナイメージの配布

- ハンズオンで配布したイメージ形式の他、**SWUpdate** 機能のアップデートイメージ (**SWUイメージ**) を作成し配布することができます。

SWUpdate機能とは：

デバイス上で実行され、ネットワーク/ストレージ経由でデバイスのソフトウェアを更新することができます。オープンソースで開発されており、Armadillo-IoT ゲートウェイ G4では、A/B アップデート・リカバリーモード・ソフトウェアの圧縮/暗号化/署名付与などに使用しています。

SWUイメージとは：

SWUpdate独自のソフトウェア配布フォーマット。Armadillo Base OS ではソフトウェアアップデート向けのOSやコンテナを格納するためにこのイメージ形式を用いています。

コンテナイメージの配布

- ハンズオンで使用したコンテナイメージを SWUイメージ化する場合にベースとして用いることができるファイルを以下の場所にアップロードしています。

■ ダウンロードURL

https://download.atmark-techno.com/sample/algyan/220108_g4_meetup/swu/

- 基本的なSWUイメージ化・書き込み手順は以下を参照してください。
 - ◆ マニュアル 9.7.2. SWUイメージの作成
 - ◆ 開発ガイド 7.1.2. SWUpdate でコンテナイメージを組み込む

コンテナイメージの配布

- swu/ 配下の構成は下記の通りです。3パターンの設定を配置しています。適宜リネームしてご利用ください。

swu/

├── algyan_containers.desc

├── containers

│ ├── azure.conf

│ ├── detect.conf

│ └── my_network.conf

├── containers_detect_only

│ ├── azure.conf

│ ├── detect.conf

│ └── my_network.conf

└── containers_weston_split

│ ├── azure.conf

│ ├── detect.conf

│ ├── my_network.conf

│ └── weston.conf

物体認識結果Azure送信
(物体認識のみ自動起動)

物体認識のみ実行
(自動起動)

westonのみ自動起動し、
物体認識を手動起動

swu配下に 物体認識およびAzureコンテナの
コンテナイメージを配置します。

Azureへの接続を行う場合は、Azure接続設定を
実施した後のコンテナを podman commit / podman
save コマンドで新たに書きだし、そのイメージを使用
してください。

コンテナイメージの配布

- コンテナインストール&Armadillo再起動後、自動起動設定されたコンテナが起動します。
- 手動起動については、下記のコマンドを実行してください。

```
armadillo:~# podman_start azure コンテナ名
```

注意事項

教師データに関して

- 機械学習に用いる教師データを**データセット**という形で公開/販売していることもあります。使用上のライセンスをよく確認した上で利用してください。
- 教師データのもととなる画像について、第三者がアップロードした画像を学習に使用する際は、著作権などの法律を十分に確認した上で、ご自身の責任において画像の収集・利用を行ってください。

質疑応答

- 当日までの質問については、下記 Armadillo フォーラムをご活用ください

- Armadillo フォーラム

<https://armadillo.atmark-techno.com/forum/armadillo>



フォーラム投稿の際、対象製品には「Armadillo-IoT G4」を選択してください。

