

先進テクノロジーで未来を切り開く

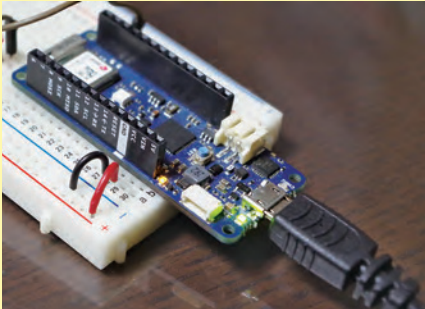
エレキジャックIoT

特集 マイコンでAI顔認証

No. 8

Sipeed M1 Dockで判断/BLEで結果送信

— OpenCV DepthAIカメラをOpenVINOで試す



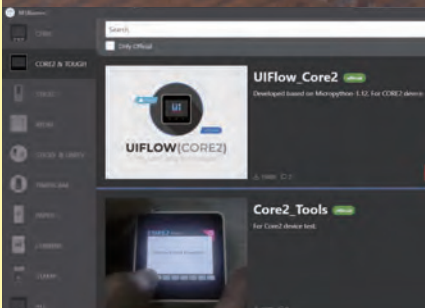
低消費電力で安定動作可能なWiFi機能搭載のArduino MKR WiFi 1010のプログラム開発環境を構築する



AIエンジンとRISC-V搭載したSipeed M1 Dock付属カメラを使ってリアルタイム画像認識プログラミング



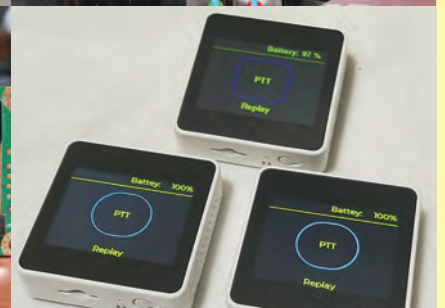
CNN(畳み込みニューラル・ネットワーク)の学習済みモデルで、人間の姿勢を検出するモデルを試す



ファームウェア書き込みソフトM5Burnerを使ってM5Stack Core2のUIFLOW用ファームウェアを書き込む



LuxonisのOpenCV AI KITをMyriad Xと豊富なAIライブラリのOpenVINOを組み合わせて画像認識を体験



マイコン独自のESP-Now通信を利用して、短い音声ファイルを送受信して疑似トランシーバ機能を実現



このPDFは、CQ出版社発売の書籍「エレキジャックIoT No.8」の一部見本です。内容・購入方法については、下記のWebサイトをご覧ください。

内容: <https://shop.cqpub.co.jp/hanbai/books/MTR/MTRZ202209.htm>

小型ディスプレイ付きマイコンM5Stackで始めるプログラミング、GPSで位置情報をクラウドにアップ

さくらのセキュアモバイルコネク特閉域ネットワークを使ってIoTのデータを流すプログラムを紹介



MicroPythonによるプログラム開発環境MatixPyで顔認証プログラムを作成、I/O端子で外部機器を制御

見本

特集 マイコンでAI顔認証

Bluetoothで
結果送信!!

AIエンジン+ 64ビット・マイコンRISC-V搭載 Sipeed M1 Dockで リアルタイム顔認識 プログラミング

使用機材

- Sipeed M1 Dock (付属カメラとLCDを含む)
- BluetoothモジュールRN4020
- ラズベリー・パイ 4

国野 亘

AIアクセラレータ付きのマイコン・ボードSipeed M1 Dockは、1インチ角(25.4mm×25.4mm)で厚さ3.3mmのAIモジュールSipeed M1を搭載したマイコン・ボードです。

顔や物体のリアルタイム認識に役立つ汎用ニューラル・ネットワーク・プロセッサKPU (Knowledge Processing Unit)を搭載しています。カメラやLCDモジュールが付属しているので、画像認識の実験を簡単にすることができます。

Sipeed M1 Dockは、金属製シールドで覆われた1インチ角のAIモジュールSipeed M1を搭載した64ビット・マイコン・ボードです。Sipeed M1には、リアルタイムに画像認識が可能なAIエンジン搭載マイコンK210を内蔵しています。カメラやLCDモジュールも付属しているので、画像認識をすぐに試すことや、画像認識を応用したシステムを簡

単に製作することができます。

本稿では、AIエンジン搭載マイコンK210を使ったカメラ応用システムのプログラム作成方法について解説します。また、ワイヤレス通信BluetoothモジュールRN4020を接続し、ラズベリー・パイ経由でインターネットに画像認識結果を送信するプログラムも説明します。

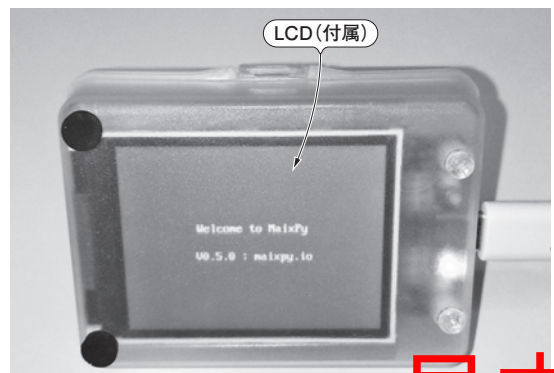
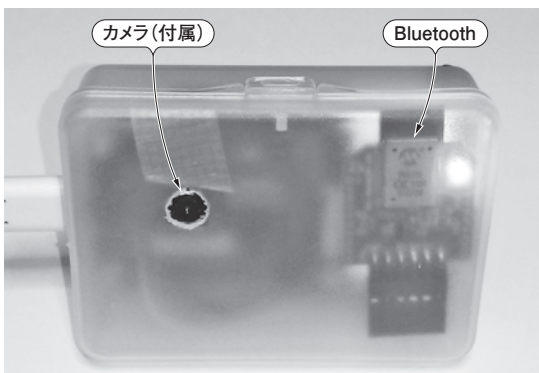


写真1 Sipeed M1 Dockに付属のカメラとLCD、別売のBluetoothモジュールRN4020を自分で加工したケースに収容し、AIエンジン、カメラ、LCD、ワイヤレス通信モジュールをクレジットカード・サイズ(厚みを除く)のケースに組み込んで実験を行った。

見本

第1章 K210マイコンの特徴と開発環境の準備

本章では、Sipeed M1 Dockおよび、K210マイコンの特徴と、開発環境MaixPyのセットアップ方法を説明します。

● Sipeed M1 モジュールとK210マイコン

Sipeed M1 Dock ボードの中央に実装されている1インチ角の金属製シールドで覆われたモジュールがSipeed M1 モジュールです。金属製シールドの中には、K210マイコンと16MBのフラッシュ・メモリ、電源回路を内蔵しています。

● AIエンジン+64ビット・マイコンK210

K210は、64ビットRISC-Vデュアルコア・プロセッサのAI用マイコンです。K210マイコン内には、KPU(Neural Network Processing Unit)と呼んでいるAI演算用ニューラル・ネットワーク・プロセッサや、最大8ch入力、最大192kHzサンプル、FFT演算機能を含むオーディオ用プロセッサAPU(Audio Processing Unit)、FPU(浮動小数点演算ユニット)などを内蔵しています。SRAMは8Mバイトを内蔵し、うち2MバイトはAI処理専用です。

また、Sipeed M1 Dockボード上には、マイクロSDカードを使用するためのTFカード・スロット、オーディオ用マイクロホン、D-Aコンバータ、D級オーディオ・アンプを装備しています。表1-1にK210マイコンを搭載したSipeed M1 Dockの概要仕様を示します。

● MicroPythonによるソフトウェア開発環境MaixPy

Sipeed M1 Dockには、あらかじめMicroPython用ファームウェアが書き込まれています。本稿では、Sipeed M1 Dockと連携可能なソフトウェア開発環境MaixPy(図1-1)を使ったAIカメラ応用MicroPythonプログラムの作成方法について説明します。

● MaixPyの特徴

MaixPyは、Sipeed M1 Dockや、M5Stack製AIカメラM5StickVなどのK210マイコン搭載製品に対応したソフトウェア開発環境です。

MicroPythonでプログラムの作成ができるほか、カメラ画像のリアルタイム表示機能やヒストグラ

表1-1 K210マイコン搭載Sipeed M1 Dock概要仕様表

項目	機能・性能(筆者による実測値を含む)
マイコン	K210 64ビットRISC-Vデュアルコア・プロセッサ(動作周波数400M~500MHz)
マイコン内蔵メモリ	8Mバイト(うち2MバイトはAI処理用)
フラッシュ・メモリ	16Mバイト、TFカード・スロット(マイクロSDXC拡張用・128Gバイトまで)
画像認識速度	60fps(QVGA時)、30fps(VGA時)
ユーザ・インターフェース	押しボタン×1、リセット・ボタン×1、フルカラーLED×1
映像入力	カメラ(OV2640/GC0328)付属
映像出力	2.4インチLCD(320×240ピクセル)
オーディオ入力	MEMSマイクロホン(MSM261S4030H0)
オーディオ出力	D-Aコンバータ(TM8211)、D級オーディオ・アンプ(PAM8403)
PC接続インターフェース	Type C、USBシリアル変換IC(CH340)
電源電圧・電流	5±0.2V・600mA以上、実動作時150mA~180mA程度(カメラ、LCDを含む・実測値)
サイズ	52.33mm×37.34mm(Sipeed M1モジュール部は25.4mm×25.4mm×3.3mm)

参考資料：Sipeed MaixDock Datasheet V1.0

<https://dl.sipeed.com/shareURL/MAIX/HDK/Sipeed-Maix-Dock/Specifications>
https://wiki.sipeed.com/soft/maixpy/en/develop_kit_board/maix_dock.html

見本

第2章 AIカメラSipeed M1 Dock用Pythonプログラムで顔検出&I/O制御

本章では、AIカメラで顔検出するSipeed M1 Dock用の基本サンプル・プログラムと、I/O制御用の基本サンプル・プログラムについて説明します。

プログラム1 AIカメラによる顔検出の基本機能

はじめに、AIカメラで顔検出した画像位置と大きさをシリアル・ターミナル(画面・左下)にログ出力するサンプル・プログラムについて説明します。プログラム2以降で顔検出結果を電子工作に応用するための基礎となるプログラムです。

● プログラムの実行方法と実行結果

プログラムの実行方法について説明します。プログラムは、前章でダウンロードしたZIPファイ

ル含まれるex1_kao.pyです。MaixPy開発環境の[ファイル]メニューからプログラムを読み込み、画面左下にある[接続]ボタンと[実行]ボタンを操作して、プログラムを起動してください。

実行後、人の顔や人のイラストにカメラを向けると、図2-1のように画面右上にカメラ画像と顔検出位置が長方形で表示されます。また、画面左下の[シリアルターミナル]をクリックすると、顔検出した座標と顔の大きさが数字で表示されます。

例えば、(55, 89, 65, 86)は、カメラ画像上のX座標55, Y座標89の位置に、幅65, 高さ86の顔を検出したことを示しています。また、複数人の顔を検出した場合は、同じ行に人数分の値を出力します。本例では2人の顔を検出し、(55, 89, 65, 86)、(209, 88, 64, 86)のように表示されました。

なお、顔検出用の学習モデルkmodelを、あら



図2-1 AIカメラによる顔検出の基本機能ex1_kao.pyを実行したときのようす
AIカメラが顔検出した画像位置と大きさをシリアル・ターミナル(画面・左下)にログ出力する

見本

はじめSipeed M1に書き込んでおく必要があります(第1章の準備2を参照)。一度、書き込んでおけば、以降のプログラムでも利用できます。

● プログラム ex1_kao.ino の内容

サンプル・プログラム ex1_kao.ino の処理内容について、リスト2-1を用いて説明します。

- ① カメラ用ライブラリ sensor と、液晶用ライブラリ lcd を本プログラムに組み込みます。
- ② AIエンジン KPU用ライブラリを本プログラムに組み込みます。
- ③ 配列変数 anchors(アンカ)に10個の数値を代入する処理部です。丸括弧を用いた配列は、タプル型と呼び、要素となる各値を個別に変更しないときに用います。anchorsは、学習モデル kmodelのサイズ比情報です。ここでは2値のアンカ値を5組、計10値を定義します。これらの値や値の数は、モデルによって異なり、使用するモデル用の値をそのまま定義します。

- ④ kpu.loadでkmodelの読み込みを行います。丸括弧内の値はフラッシュ・メモリの相対アドレスです。マイクロSDに保存した学習モデルを読み取りたいときは、kpu.loadの引き数にファイル名を"/sd/xxx.kmodel"のように渡します。
- ⑤ kpu.init_yolo2コマンドで画像認識エンジンの初期化を行います。丸括弧内の引き数は、学習モデル、出力閾値レベル(0~1)、重複出力の閾値レベル(0~1)、アンカ数、アンカ値です。
- ⑥ 処理⑤の第2引き数の出力閾値レベルは、認識結果の信頼度(確からしさ)の低いものを出力しないための設定です。値が大きいくほど信頼度の高い認識結果しか出力しなくなります。
- ⑦ 第3引き数の重複出力の閾値レベルは、認識対象の重なりに対する閾値レベルです。値が大きいくほど重なった結果が出力されやすくなります。
- ⑧ アンカ数を演算で求める処理部です。lenは要素数を得るための関数です。ここではanchorsの配列数10を得ます。演算子//は整数除算です。

リスト2-1 プログラム ex1_kao.py

付属カメラで撮影を行い⑨、AIエンジンKPU用の画像認識機能②③④⑤を用いて、顔検出を実行⑩し、撮影した画像に結果を追記⑬、LCDに表示する⑭

```

import sensor, lcd ←①
import KPU as kpu ←②

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_vflip(True)
sensor.set_hmirror(True) ←※対面撮影時はFalse

anchors = (1.889, 2.525, 2.947, 3.941, 4.0, 5.366, 5.155, 6.923, 6.718, 9.01) ←③
task = kpu.load(0x300000) ←④
kpu.init_yolo2(task, 0.3, 0.1, len(anchors)//2, anchors) ←⑤

while(True):
    img = sensor.snapshot() ←⑨
    objects = kpu.run_yolo2(task, img) ←⑩
    n = 0
    if objects: ←⑪
        n = len(objects)
        for obj in objects: ←⑫
            {
                img.draw_rectangle(obj.rect())
                ⑬ {img.draw_string(obj.x(), obj.y(), str(obj.value)) # 文字列を追記
                    print(obj.rect(), end=', ')
                }
            }
        print()
        img.draw_string(0, 210, 'n=' + str(n), scale=3)
        lcd.display(img) ←⑭
    # カメラ sensor, 液晶 lcd の組み込み
    # AI 演算ユニット KPU の組み込み
    # LCD の初期化
    # カメラの初期化
    # 色設定 (白黒時 GRAYSCALE)
    # 解像度設定 (QVGA: 320x240)
    # カメラ画像の上下反転設定
    # カメラ画像の左右反転設定
    # 永久ループ
    # 撮影した写真を img に代入
    # 写真 img 内の顔検出を実行
    # 検出件数を保持する変数 n
    # 1件以上検出したとき
    # 検出件数を数値変数 n に代入
    # 個々の検出結果ごとの処理
    # 検出範囲を img に追記
    # 文字列を追記
    # 検出結果をログ出力
    # 改行をログ出力
    # 検出件数を img に追記する
    # img を LCD に表示

```

見本

● プログラム ex3_kao_uart.py の内容

UARTの出力方法について、リスト2-3のプログラム ex3_kao_uart.py を使って説明します。

- ① Maix ライブラリ内のUART機能をプログラムに組み込みます。
- ② FPIOA管理機能を組み込みます。
- ③ FPIOAを使って、IO7ピンをUART1のTX(送信)に割り当てます。
- ④ UART1の通信パラメータの設定を行い、オブジェクトuartを生成します。丸括弧内の第1引き数にはUARTチャンネル1を、第2引き数にはビット・レート、以降、ビット数、パリティ有無、ストップ・ビットを渡します。これらは受信側(パソコン側)の設定と同じにします。
- ⑤ 起動したことを示すために文字列Hello!をUART送信します。\\nは改行を示します。
- ⑥ 顔検出した結果を文字列に変換し、文字列変数sに追記します。検出結果は、X座標x、Y座標y、幅w、高さhの4値を含む図2-6(2)のようなタプル型の配列です。

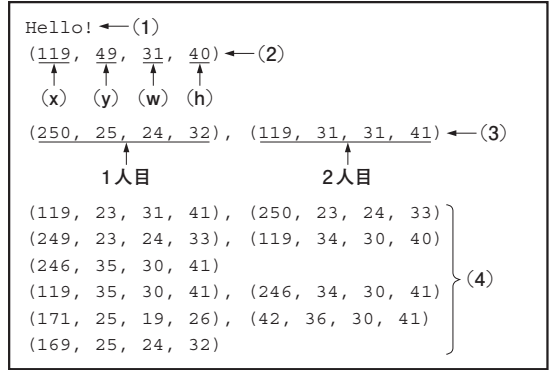


図2-6 顔検出結果のUART出力例
AIカメラの認識結果(X座標, Y座標, 幅, 高さ)の4値を、検出した顔の人数分だけ出力する

- ⑦ 変数sの内容をUART送信します。図2-6(4)のように顔を検出するたびに繰り返します。

第3章 AIカメラ Sipeed M1 Dock 応用例：来場者カウンタ

本章では、AIカメラで顔検出したデータの応用例として、来場者カウンタを製作してみます。さらに、Bluetoothによる送信方法についても紹介します。図3-1にAIカメラによる来場者カウンタ(ex4_kao_count.py)の構成を示します。

プログラム4 AIカメラで来場者をカウントする

AIカメラの顔認識で検出した顔の人数を累計し、来場者数をカウントするプログラムを作成します。検出した人数をそのまま累計すると、映り続けている顔を重複して数えてしまうので、同じ位置で検出した顔を累計値に加算しないようにします。

● 来場者カウンタのシステム構成

必要なハードウェアは、Sipeed M1 Dockと、付属品のカメラ、付属LCDです。ソフトウェアは、ex4_kao_count.pyを使用します。前章同様、顔検出用の学習モデルkmodelをSipeed M1に書き込んでおく必要があります(第1章の準備2を参照)。すでに書き込んであれば、再書き込みは不要です。

● AIカメラ設置時の注意点

AIカメラを設置して、許可なく撮影することは、プライバシーの侵害に抵触する可能性があります。とくに本機は小型なので、撮影していることを張り紙などで知らせる必要があります。映像を保持しない場合であっても、撮影していることにより、

見本

● プログラムex4_kao_count.pyの内容

AIカメラが検出した顔をそのまま累計すると、映り続けている顔を重複カウントしてしまうので、新たに映った顔なのか、以前から映っていた顔なのかを判断する必要があります。本プログラムでは、過去10フレーム分の結果から、直近3フレームの同じ位置に顔を検出しつつ、残る7フレームの同じ位置に顔を検出していなかった場合に、来場者カウンタ(累計値)を加算します(図3-2)。

加算の判定閾値は、リスト3-1の①で定義する6つのパラメータで行います。BufHist_N = 10は、映り続けている顔が同一人物かどうかを判断するために保持する認識結果数です。ここでは過去10フレーム分の認識結果をバッファに保持します。Det_N = 3とDet_Thresh = 0.2は、顔検出を判定するためのフレーム数と判定閾値です。ErrorFact

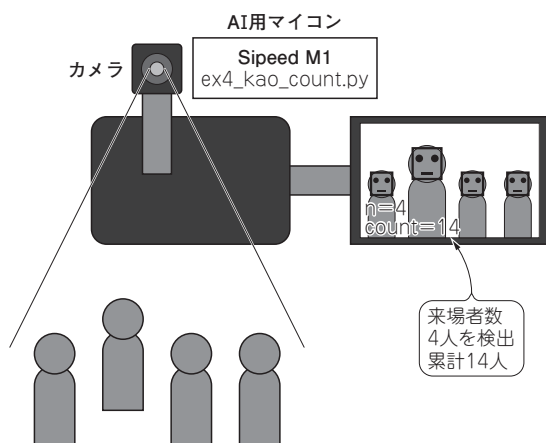


図3-1 AIカメラによる来場者カウンタex4_kao_count.pyを実行したときの様子

AIカメラの認識結果を利用し、新たに検出した人数をカウントする

= 0.2は、10フレーム中の残り7フレームに顔が映っていないことを判定するための閾値です。MotionFact = 2.0とExtentFact = 2.0は、認識対象の座標の変化や、認識対象の大きさの変化に対する許容値です。歩行などで顔位置が変化することを許容します。

ここでは、画像認識エンジンから得られた直近3フレームの信頼度がDet_Thresh = 0.2以上のとき、かつ残り7フレームの信頼度がErrorFact = 0.2以下のときに、来場者カウンタ(累計値)に1を加算します。以上の来場者のカウント方法を組み込んだ、リスト3-1のプログラムex4_kao_count.pyのおもな動作について説明します。

- ① 来場者カウンタ(累計値)countに加算するかどうかを判断するための6つのパラメータを定義します。環境によって誤検出がある場合は、調整してください。
- ② 認識結果に含まれる顔の位置座標、顔の大きさから、検出した顔が新たに映った顔である度合(0~1)と、過去に映っていた度合(0~1)を算出する関数です。引き数は今回の認識結果と、バッファに保存した過去の認識結果です。戻り値は、直近3フレーム分の顔認識の信頼度と、残る7フレーム分の信頼度です。以下③~⑦の処理を行います。
- ③ バッファ数が10に満たないときは、処理を実行せずに0を応答します。
- ④ 過去の認識位置の中から、今回の検出位置と同じ位置に検出していたかどうかを確認します。
- ⑤ 同じ位置に顔を検出していた場合に、画像認識

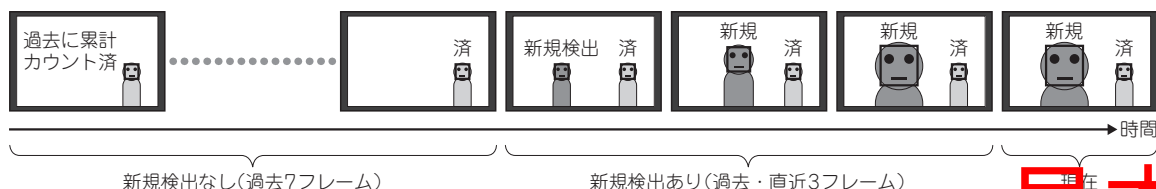


図3-2 来場者数の累計方法

同じ位置で検出した顔を累計値に加算しない

見本

第4章 GPSデータをクラウドに送信

M5Stack純正のGPSユニットで取得した位置情報を元に現在地を地図上に表示するプログラムです。クラウド・サービス Ambient に送信すれば、より詳細な地図の表示や、移動履歴の表示、位置情報の公開ができるようになります(図4-1)。

プログラム7(ex07_gps.ino) 現在位置の表示

使用機材：M5Stack, GPSユニット

● GNSS送信機の製作

ハードウェアは、M5Stack純正のGPSユニットをM5Stack Coreの Grove 互換端子に接続します。本ユニットは、米国のGPSはもちろん、GLONASSや、中国のBDSにも対応しています。出力形式は一般的なNMEA形式です。

ソフトウェアは、第1章でダウンロードしたm5-masterフォルダ→coreフォルダ→ex07_gpsフォルダ内のプログラム7(ex07_gps.ino)を使用します。Core2用はcore2フォルダ内に収録しました。

Arduino IDEでプログラムを読み込み、プログラム内のSSIDとPASSの部分を使用するWi-Fi無

線LAN内蔵ゲートウェイ、またはモバイル・ルータのSSIDとパスワードに書き換え、プログラム内のAmb_IdとAmb_Keyに Ambient用のチャンネルIDとライト・キーを設定してください。

● GNSS送信機の使用法

プログラムを起動すると日本地図を表示し、32秒から数分後に図4-2(a)のような赤色の丸印で現在地を表示します。また、位置情報の値をLAN内にUDP送信します(プログラム4の受信機で表示可能)。

衛星からの電波を受けにくい建物内や地下などでは、位置情報が得られないことがあります。M5Stack用GPSユニットは、回線の中継基地局の情報を併用するなど工夫されている身近なGPS利用機器のスマートフォンと比べると、座標情報の取得まで所要時間が長く精度もやや劣ります。

中央ボタンを押すと、図4-2(b)のような相対座標情報が表示されます。図の中心はボタン押下時の現在地で、画面の上方向が北方向です。本機を移動すると、その軌跡を画面上に残します。

右ボタンを押すと、GPSユニットのログ情報を

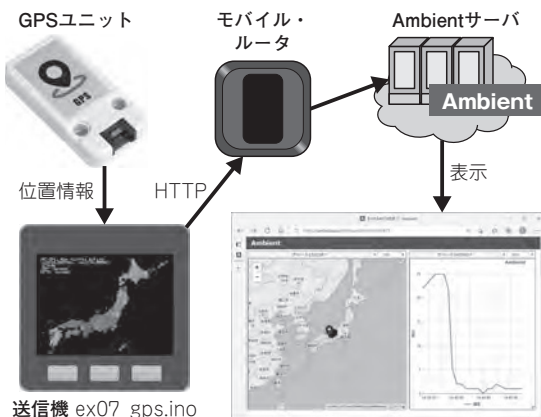


図4-1 現在位置をLCDに表示&クラウドに送信するGNSS送信機

GPSユニットから得られた位置情報をモバイル・ルータで送信し、Ambientのクラウド・サービスを経由してWebブラウザで表示する

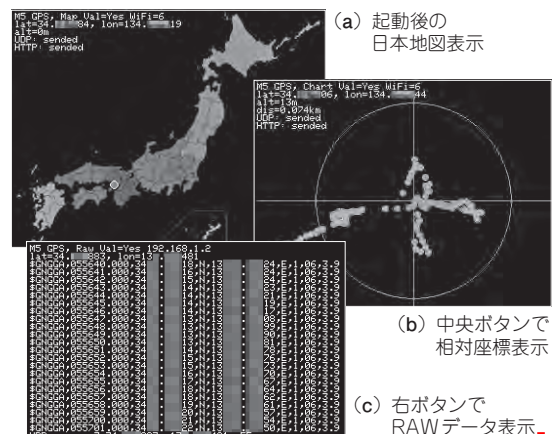


図4-2 GNSS送信機の表示例

M5Stack上のボタン操作で、日本地図、相対座標、RAWデータ表示の切り替えができる

見本

グループでも通話可能!

マイコンで作る 疑似トランシーバ2種!

—インターネット版& ESP-Now通信版—

M5Stack Core2で
音声データを
相互送受信

使用機材

• M5Stack Core2

生田 朋広
(株式会社 WorldLink&Company)

M5Stack Core2(ESP32)で疑似トランシーバを2種類作ります。1つ目は、M5Stack用プログラミング・ツールUIFLOWを使ったWi-Fiインターネット経由で音声データ通信をする方法です。2つ目は、ESP-Nowのローカル無線通信で音声データを送る方法です。Wi-FiやESP-Nowでの無線データ通信と、音声データを扱うプログラミングを体験します。

システムの構成

● 全体のイメージ

1つ目のUIFLOW版は、図1のように、送信時は、M5StackCore2内蔵のマイクで録音した音声データをM5Stackサーバにアップロードします。

受信時は、M5StackCore2でM5StackサーバからWi-Fi経由で音声データをダウンロードして、

再生をすることで送信音を再生します。

この仕組みは、M5Stackサーバを使用しているため、ローカル・エリアのネットワークでは使用できませんが、インターネット環境があれば音声通話できます。

M5Stackサーバを使用するために、プログラミング・ツールUIFLOWのMediaTrans・Audioブロックを使用しました。

2つ目のArduino ESP-Now版は、図2のように、

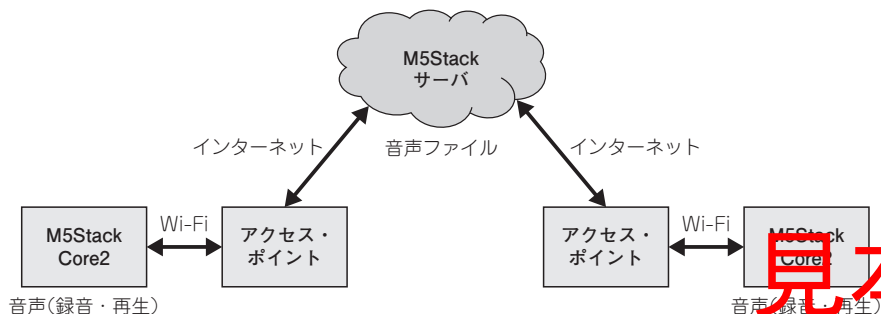


図1
UIFLOW版疑似トランシーバの構成図

音声(録音・再生)

音声(録音・再生)

見本

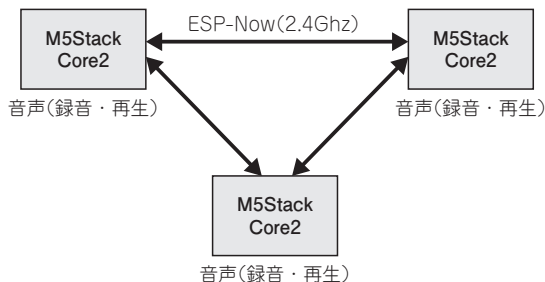


図2 Arduino ESP-Now版疑似トランシーバの構成図

M5Stack Core2内蔵のマイクで録音した音声データをESP-Now通信でブロードキャストし、そのデータを直接受信したM5Stack Core2で音声データを再生することで通信できる疑似トランシーバの機能を実現しました。

ESP-Now通信とは、ESP32系マイコンに備わっている機能で、ESP32系マイコン同士なら、無線LANのIEEE 802.11を使ったデータ通信が可能です。Espressif開発のオリジナルのプロトコルが使われています。

このESP-Now通信は、無線LANが届く範囲でのみ使用できますが、ブロードキャストを使用しているので複数台のM5Stack Core2で音声データのやり取りができます。

● 使用する機材について

今回使用する機材を表1に示します。M5Stack Core2は、M5Stack BasicなどCore機能を改良した第2世代シリーズで、ESP32(ESP32-D0WD-V3)を搭載し、さらに、IPS-LCD(2.0インチ@320×240 ILI9342C)、タッチ・スクリーン、390mAhバッテリー、microSDカード・スロット、マイク(SPM1423)、スピーカ、I²Sパワー・アンプ(NS4168)、6軸IMU、振動モータ、ボタン、PMIC(AXP192)等の豊富なハードウェアが内蔵されたデバイスです。BluetoothとWi-Fi通信も可能です。また、M5Stack用拡張モジュールやUnitファミリ製品が利用可能です。

表1 機材表

部品・機材	品名	数	購入先
M5Stack Core2本体	M5Stack Core2 IoT開発キット	2個以上	https://www.switch-science.com/catalog/6530/

表2 サンプル・プログラム構成

No.	ファイル名	内容
1	M5WalkieTalkie_core2_v1.m5f	UIFLOW用サンプル・プログラム M5Stack UIFlowで書き込む
2	M5WalkieTalkie_core2_v1.ino	Arduino ESP-Now用サンプル・プログラム Arduino IDEで書き込む

プログラム構成

作成したプログラム(表2)は、Webページ(<https://eleki-jack-iot.com>)で公開しています。このプログラムをダウンロードして、必要箇所を修正してM5Stack Core2に書き込んで使用してください。このあと解説するように、UIFLOW用サンプル・プログラムは、M5Stack社から提供されているビジュアル・プログラミング・ツールUIFLOWで書き込みをしてください。また、Arduino ESP-Now用サンプル・プログラムは、Arduino IDE環境で書き込みをしてください。

これらサンプル・プログラムを作成するときに参考文献(1)～(6)を参考にしました。

M5Stack疑似トランシーバを作ってみる(UIFLOW版)

- M5Stack Core2にサンプル・プログラムを書き込んで、動かしてみる(UIFLOW版)
- ① UIFLOWに対応したファームウェアをM5Stack Core2に書き込みます。M5Stack社のページ(https://docs.m5stack.com/en/quick_start/core2/uiflow)より、Windows用のファームウェア書き込みソフトM5Burnerをダウンロードし

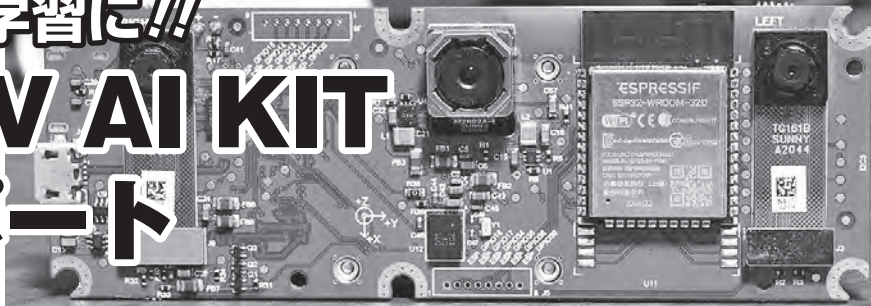
見本

OpenCV
DepthAIカメラを
使ってみよう

特別
企画

カメラを使ったAI開発や
実験/学習に!!

OpenCV AI KIT 試用レポート



使用機材

• OAK-D-IoT-75 (Luxonis)

中村 文隆

(東京大学情報基盤センター)

はじめて OpenCV DepthAI カメラ (OAK-D) を使うための基礎知識として、DepthAI のインストールからデモ・スクリプトの試用、簡単な顔検出のスクリプトについて説明します。

高性能 VPU の Myriad X と豊富な AI ライブラリの OpenVINO の組み合わせで、強力な画像認識を体験してみましょう。

1. OpenCV AI KIT

1.1 OpenCV AI KIT (OAK)

近年、コンピュータによる画像解析 (Computer Vision : CV) と人工知能 (Artificial Intelligence : AI) の技術は急速な発展を遂げ、エンドユーザ・レベルでも高度な機能を手軽に使うことができるようになってきています。中でもインテルが開発した OpenCV は世界中で広く使われている画像解析ライブラリですが、実際に使いこなすには解析に使う AI 用の学習データを用意し、それなりのコーディングが必要という点では、まだ敷居の高いところもありました。

本記事で試用した Luxonis の OAK (OpenCV AI KIT) シリーズは、インテルのビジョン・プロセッシング・ユニット (VPU) Myriad X と 3 つのカメラを組み合わせ、同社の提供する DepthAI API と学習済みデータを利用して、さまざまな AI 画像解

析を手軽に利用できる画期的な製品です。

1.2 試用機について

今回試用した製品は、OAK-D-IoT-75 (以下、OAK-D) です。図1のように3つのカメラが並んで実装されていて、左右のカメラがグレー・スケール・カメラで奥行き認識などの処理のためにステレオのペアとして使われ、中央のカメラは、レンズがカラー・カメラで色情報の取得に使われます。

グレー・スケールのステレオ・カメラは 1200 × 800px の解像度で最高 120fps のフレーム・レートに対応し、シャッターはグローバル・シャッターのため、高速で移動する物体に対する処理も可能な優れた機能を持っています。

中央のカラー・カメラは、60fps でローリング・シャッターですが、解像度は 4056 × 3040px でメイン・カメラとして使われます。

接続端子は図2のように USB Type-C と Type-C のマイクロ、外部電源用となっている Type-C

見本

試用レポート

センサ情報をクラウドに CSV×UDP送信してみた!

さくらの セキュアモバイルコネク

使用機材

- さくらのセキュアモバイルコネク用 SIMカード (要さくらインターネットの法人契約)
- モバイル・ルータ
- M5StickC または M5StickC Plus
- DLIGHT ハット, または ENV Ⅲ ハット

国野 亘

インターネットを経由しない閉域網モバイル通信が特徴のさくらのセキュアモバイルコネク (図1) を使って、クラウド・サーバに接続してみます。

マイコンでCSV形式にしたセンサ値をクラウド・サーバにUDPで送信して、IoTシステムへの応用を考えます。

さくらのセキュア モバイルコネクの特徴

さくらのセキュアモバイルコネクは、モバイル回線をインターネット非経由でクラウド・サーバに接続する閉域網モバイル通信サービスです。IoT/M2M(データ)用のSIMカード(図1)を使って、さくらインターネットが運営するクラウド・サーバに接続することができます。

最大の特徴は、モバイル網がまるでLANのような閉域ネットワークであることです。インターネットを経由しないので、閉域網内での認証や暗号化を行わなくてもセキュアな通信が行えます。もちろん、通常のデータ用SIMカードと同じようにインターネット接続可能な構成にすることもできます。

ほかにもメリットがあり、SIM 1枚あたりの基本利用料が月額13円と低価格な点、ソフトバンク、KDDI、DoCoMoの3キャリアに対応している点、

通信速度制限がない点など、IoT機器に適したモバイル通信サービスになっています。システム全体のサービス総額(月額)は、基本利用料やデータ通信料のほか、各種のクラウド・サービス利用料が必要なので、はじめに試算しておくといでしょう。

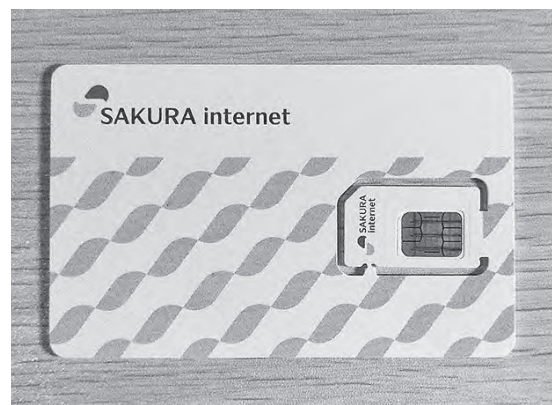


図1 さくらのセキュアモバイルコネク用SIMカード。nano SIM、micro SIM、mini SIM(標準SIM)に対応したマルチサイズのSIMカード

見本

サービス総額(月額)の試算

さくらのセキュアモバイルコネクトをさくらインターネットのクラウド・サーバに接続するのに必要なサービスの総額(月額)を試算してみます。

はじめに、データ通信料(ソフトバンク利用の場合6円/MB)です。UDPでセンサ値(40バイト以内)を30秒ごとに送信すると、1カ月当たり約87,660パケットとなり、約11.2MB、67円に相当します(課金は1MB単位、1パケット128バイトで換算)。また、ソフトバンク網を利用する場合は500MB(月額3,000円相当)までのデータ通信料が後述のモバイル・ゲートウェイの利用料に含まれているので、SIMカード40枚くらいまでであれば、

データ通信料は不要でしょう。

閉域網をさくらのクラウド・サーバに接続するには、図2の(a)モバイル・ゲートウェイの利用料(月額4,400円)と、(b)スイッチの利用料(月額2,200円)が必要です。また、(c)クラウド・サーバの利用料(月額2,585円～)も必要です。さくらのVPSなどにも接続できますが、ブリッジ利用料(月額2,750円)が別途必要なので、事情がない限りクラウド・サーバを使ったほうが良いでしょう。

クラウド・サーバの利用料は、東京よりも石狩データ・センタのほうが安価です。しかも、北海道の冷涼な気候を活用した省エネ化や、地震などのリスクが比較的低いといった特徴もあります。

以上より、最小のサービス総額は月額9,198円で、SIMカード40枚で使うのであれば、月額9,705円、

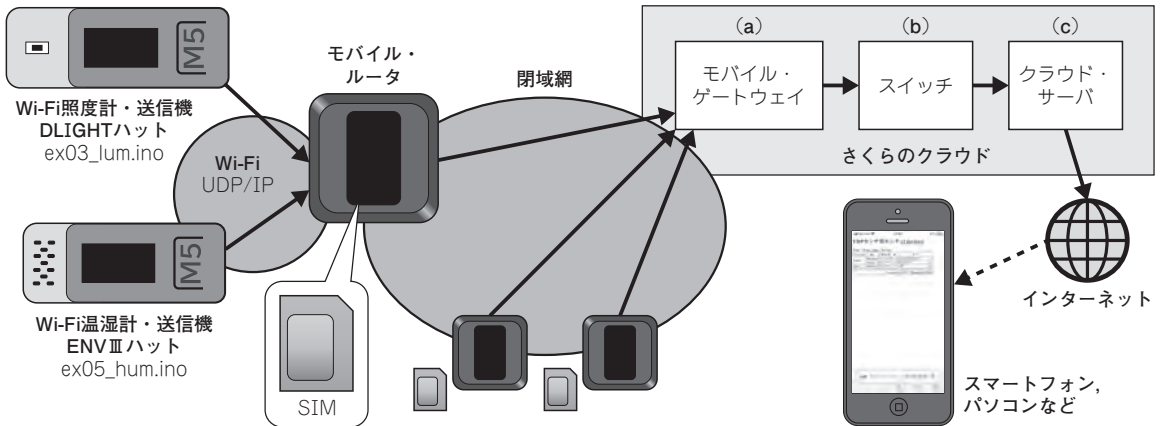


図2 さくらのセキュアモバイルコネクトを使ったシステム例
M5StickCが送信するセンサ値データを閉域網でクラウドに送信する

表1 30秒間隔でUDP送信するシステムのサービス総額(月額)の試算例(初期費用・ハードウェアを除く)

項目	SIM 1枚	SIM 40枚	SIM 100枚	システム条件、試算条件など
基本利用料	13円	520円	1,300円	SIM 1枚あたり13円 最大10,000枚
データ通信料	0円	0円	3,732円	SIM 1枚あたり11.2MB、ソフトバンク利用時
モバイル・ゲートウェイ利用料		4,400円		500MBまでの無料データ通信料を含む
スイッチ利用料		2,200円		クラウド・サーバとの接続用
クラウド・サーバ利用料		2,585円		コア1、RAM 2G、SSD 20GB、石狩
合計(カッコ内は1枚当たり)	9,198円(9,198円)	9,705円(243円)	14,217円(142円)	SIM枚数が少ないと、クラウド・サーバ利用料の合計9,185円の負担割合が大きくなる

価格は執筆時点(参考文献：<https://iot.sakura.ad.jp/sim/>)