

# NX-N947

## AI サンプルプログラム解説

1.1 版 2025年07月18日

<b>1. 概要</b> .....	<b>2</b>
1.1 概要 .....	2
1.2 接続概要 .....	2
1.3 本サンプルプログラムについて .....	3
1.4 開発環境について .....	3
1.5 ワークスペースについて .....	3
<b>2. サンプルプログラムの構成</b> .....	<b>4</b>
2.1 フォルダ構成 .....	4
2.2 ファイルの構成 .....	5
<b>3. NX-N947 サンプルプログラム</b> .....	<b>7</b>
3.1 Debug Console 機能を用いたシリアルターミナルの入出力 .....	8
3.2 Python のインストール .....	10
3.3 動作説明 .....	11
3.4 MCUXpresso IDE を用いたプロジェクトのビルド・デバッグと外部 ROM への書込み .....	15

# 1. 概要

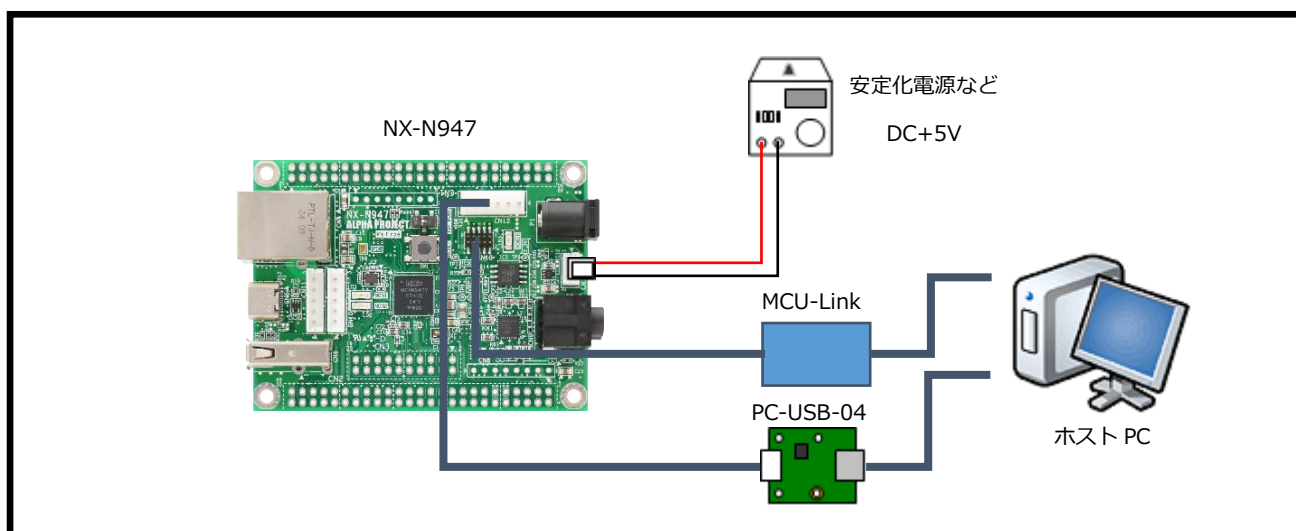
## 1.1 概要

本アプリケーションノートでは、NX-N947 に付属するサンプルプログラムについて解説します。  
 解説するサンプルプログラムは下記のものになります。

デバイス	機能
NX-N947	・ 画像分類

## 1.2 接続概要

本サンプルプログラムの動作を確認する上で必要な CPU ボードの接続例を以下に示します。  
 詳細な接続に関しては後述の「3.2 動作説明」を参照してください。



### 1.3 本サンプルプログラムについて

本サンプルプログラムおよび本書含むアプリケーションノートは、弊社 Web サイトのボード紹介ページで公開されています。

株式会社アルファプロジェクト

NX-N947 製品ページ <https://www.apnet.co.jp/product/nx/nx-n947.html>

### 1.4 開発環境について

本サンプルプログラムは統合開発環境「MCUXpresso IDE」と「MCUXpresso SDK」を用いて開発されています。本サンプルプログラムに対応する開発環境、FSP、コンパイラ、デバッガのバージョンは次の通りです。

ソフトウェア	バージョン	備考
MCUXpresso IDE	V24.12[Build 148]	–
MCUXpresso SDK	V24.12.00	SDK_2.x_MCXN947 を使用
Python	Ver.3.13.3	–

デバッガ	ハードウェアバージョン	備考
MCU-Link	V3.146	–

### 1.5 ワークスペースについて

本サンプルプログラムのプロジェクトファイルは次のフォルダに格納されています。

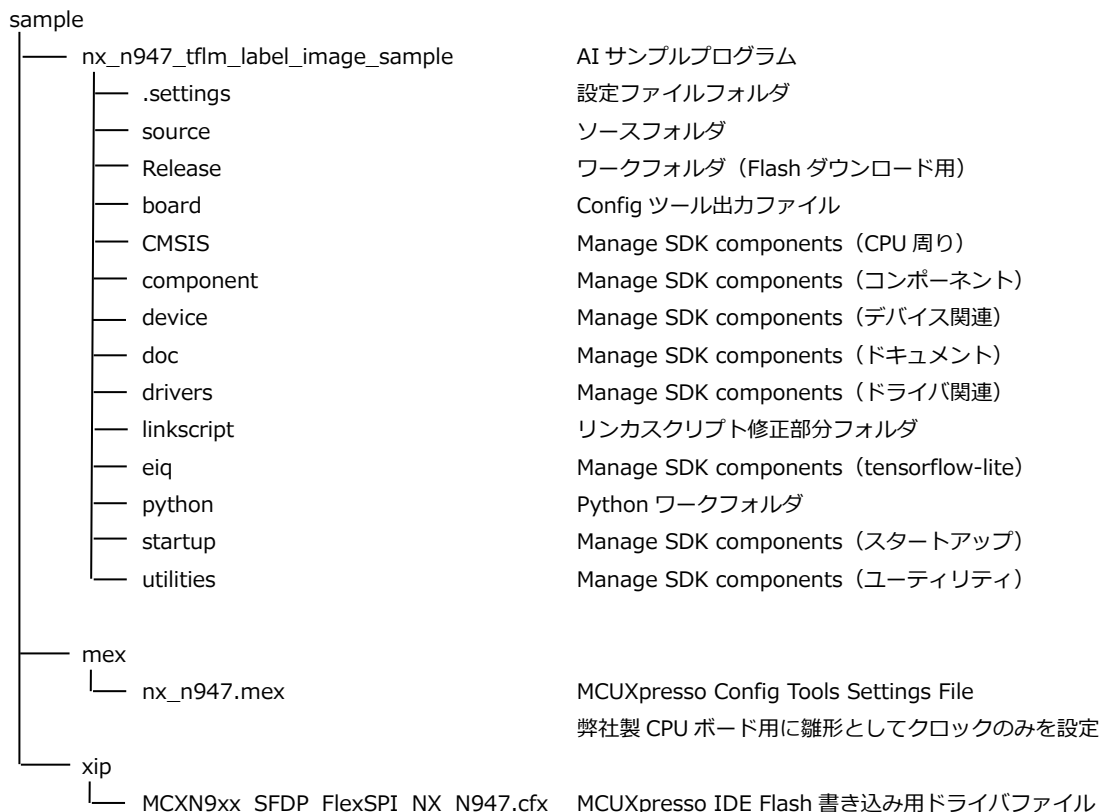
サンプルプログラム	フォルダ
AI サンプルプログラム	¥sample¥nx_n947_tflm_label_image_sample

## 2. サンプルプログラムの構成

### 2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。

本サンプルプログラムは、RAM のサイズが足りず、RAM 上にコード領域とデータ領域の両方を配置することができません。そのため、FlashROM へダウンロードを行う Release 構成のみとなります。



## 2.2 ファイルの構成

サンプルプログラムは以下のファイルで構成されています。

本節では、サンプルプログラムの作成にあたって追加したファイルについてのみ記述し、自動生成ファイルなどに関しては説明を省略します。

### ・ AI サンプルプログラム

<¥sample¥nx\_n947\_tflm\_label\_image\_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_tflm_label_image_sa	...	NX-N947 AI サンプルプログラム
mple.mex		カスタムボードファイル

<¥sample¥nx\_n947\_tflm\_label\_image\_sample¥python フォルダ内>

image_resize.py	...	画像変換用スクリプト
-----------------	-----	------------

<¥sample¥nx\_n947\_tflm\_label\_image\_sample¥source フォルダ内>

demo_config.h	...	コンフィグ用ヘッダファイル
demo_info.cpp	...	モデル information ソースファイル
demo_info.h	...	モデル information ヘッダファイル
labels.h	...	ラベル用ヘッダファイル
main.cpp	...	アプリケーションソースファイル
timer.c	...	timer 関数ソースファイル
timer.h	...	timer 関数ヘッダファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル
image.h	...	image_load 関数ヘッダファイル
image_data1.h	...	image_data1 ヘッダファイル
image_data2.h	...	image_data2 ヘッダファイル
image_data3.h	...	image_data3 ヘッダファイル
image_decode_raw.c	...	image_decode 関数ソースファイル
image_load.c	...	image_load 関数ソースファイル
image_utils.h	...	image_utilities ヘッダファイル
get_top_n.cpp	...	予測値算出関数ソースファイル
get_top_n.h	...	予測値算出関数ヘッダファイル
model.cpp	...	model 用ソースファイル
model.h	...	model 用ヘッダファイル
model_data.h	...	model ヘッダファイル
model_mobilenet_ops_npu.cpp	...	operator 関数ソースファイル
output_postproc.cpp	...	出力処理関数ソースファイル
output_postproc.h	...	出力処理関数ヘッダファイル
board_init.h	...	ボード初期化ヘッダファイル
mcux_config.h	...	MCU コンフィギュレーションヘッダファイル
semihost_hardfault.c	...	HardFault ハンドラソースファイル

**注意**

サンプルプログラムをインポートした後で、クロックを変更するなどをしてコードを再出力する際、操作を誤るとサンプルプログラムの「Config ツール出力ファイル」フォルダ以下および「Manage SDK components」が含まれる複数のファイルが上書き(サンプルプログラム向けの修正箇所が変更前の状態に戻されてしまう)されてしまいます。

この結果、プログラムが動作しなくなることがありますので、取り扱いには注意してください。

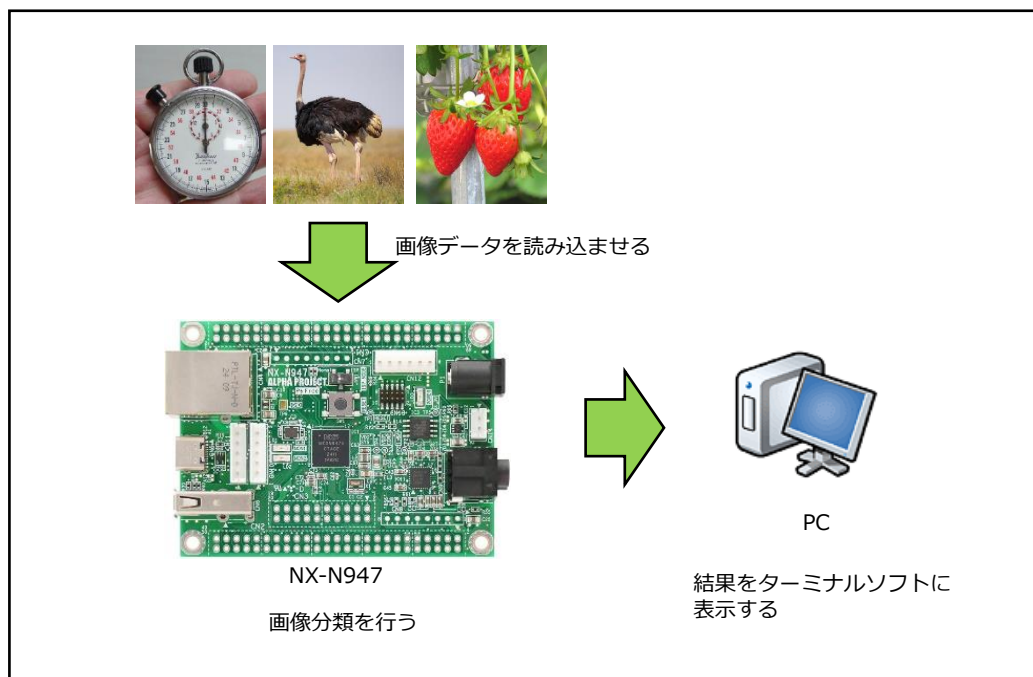
別書 AN2301 NX シリーズ 開発チュートリアル 3.2.項を参照ください。

### 3. NX-N947 サンプルプログラム

本サンプルプログラムでは、N947 に内蔵されている統合型 eIQ Neutron Neural Processing Unit (NPU) ※と NXP 社から提供されている学習データを使い、3 種類の画像の画像分類を行います。

サンプルプログラムを実行しますとそれぞれの画像の分類結果と画像分類の解析にかかった時間を出力します。

学習データは、TensorFlow Lite を用いて作成されており、金魚、コーヒーポット、消防車など 1000 種類の画像分類が可能なため、本サンプルプログラムで使用した画像以外の画像分類も行えます。



サンプルプログラムの処理のイメージ図

※ NPU は、AI の推論処理を高速化するために設計されたプロセッサです。

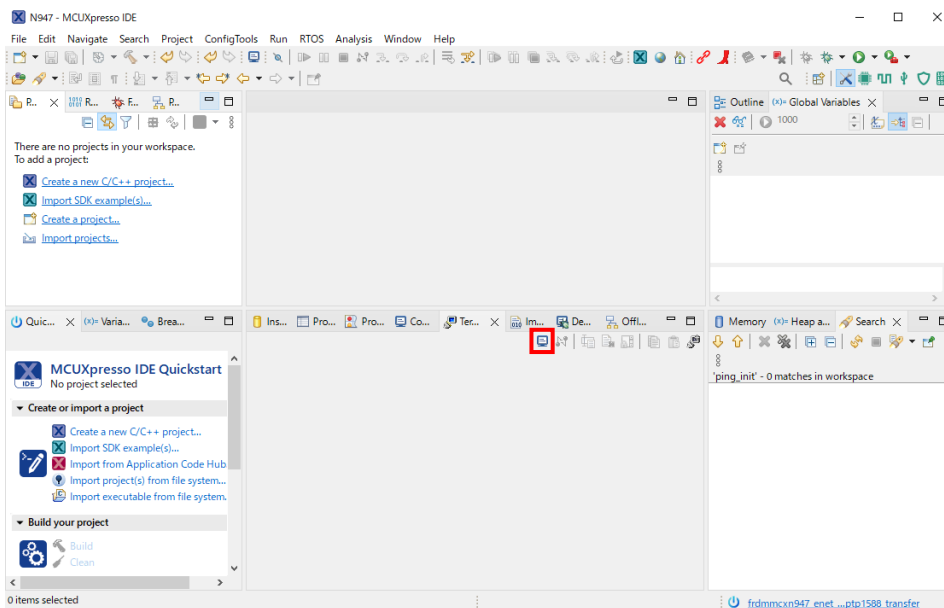
消費電力を抑え、ニューラルネットワークの処理を効率化することで、画像認識や言語処理などの AI 処理の機能を向上させます。

### 3.1 Debug Console 機能を用いたシリアルターミナルの入出力

サンプルプログラムは、MCUXpresso IDE が提供している Debug Console 機能を使用し、動作状況やプログラムの情報が表示される動作があります。

Debug Console 機能を使用した接続は以下の手順に従ってください。

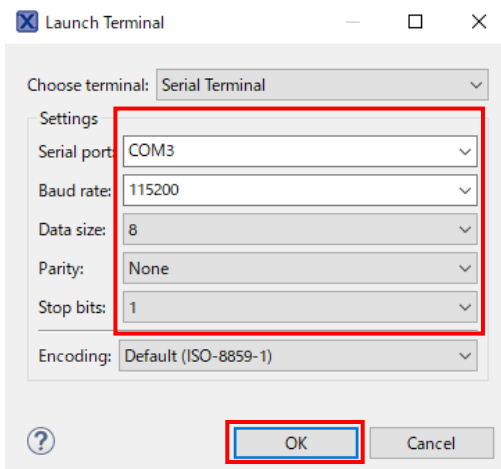
- ① CPU ボードとホスト PC を、UART コネクタ経由で接続します。
- ② ホスト PC 上で MCUXpresso IDE のターミナルソフトの設定画面を起動します。



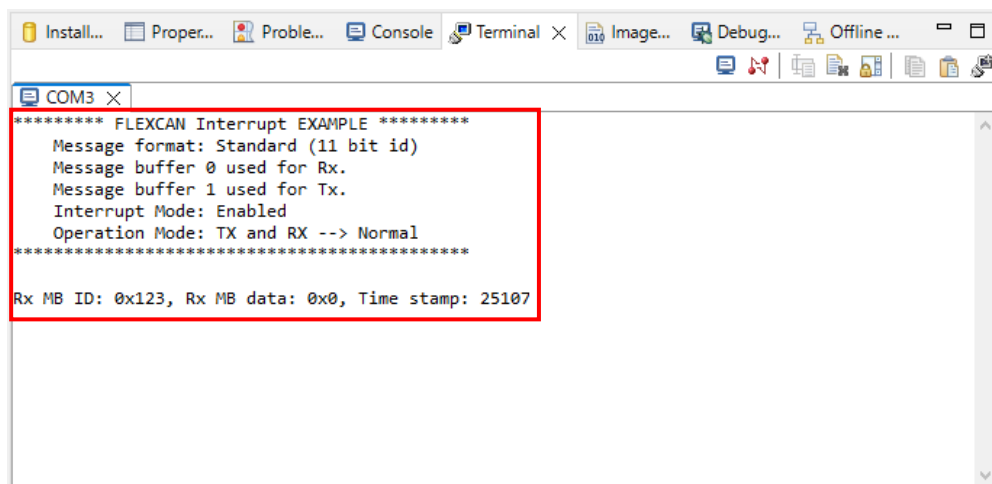
- ③ UART の設定を、115200bps、ビット長 8、パリティなし、ストップビット 1 に設定し、[OK]ボタンを押下し、ターミナルを起動してください。

※ただし[Serial port]に関しては環境によって異なります。

デバイスマネージャー等で使用されている PORT を確認の上、設定してください

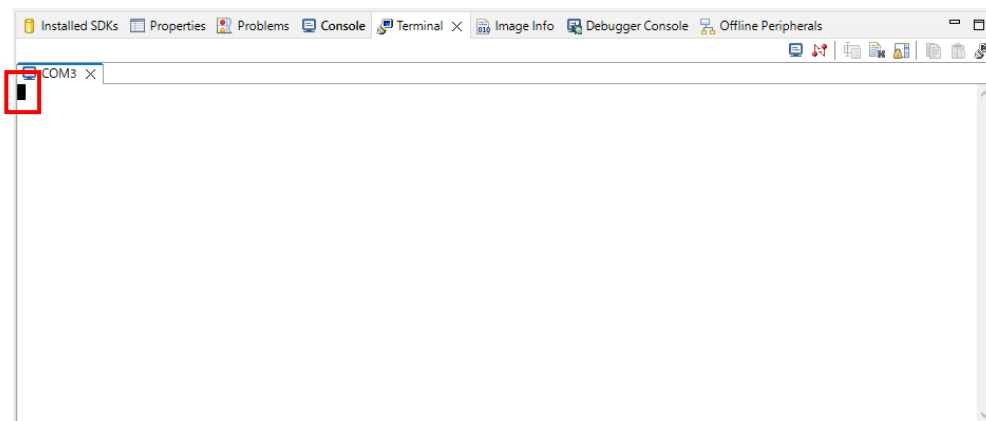


- ④ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ⑤ 接続が完了しますと、CPU ボードからの出力内容が表示されます。



```
***** FLEXCAN Interrupt EXAMPLE *****  
Message format: Standard (11 bit id)  
Message buffer 0 used for Rx.  
Message buffer 1 used for Tx.  
Interrupt Mode: Enabled  
Operation Mode: TX and RX --> Normal  
*****  
Rx MB ID: 0x123, Rx MB data: 0x0, Time stamp: 25107
```

- ⑥ CPU ボードに入力する場合はカーソルが表示されていればいつでも入力することができます。



```
***** FLEXCAN Interrupt EXAMPLE *****  
Message format: Standard (11 bit id)  
Message buffer 0 used for Rx.  
Message buffer 1 used for Tx.  
Interrupt Mode: Enabled  
Operation Mode: TX and RX --> Normal  
*****  
Rx MB ID: 0x123, Rx MB data: 0x0, Time stamp: 25107  
  
|
```

- ⑦ 動作確認が完了したら、ターミナルソフトを終了し、CPU ボードの電源を落とします。

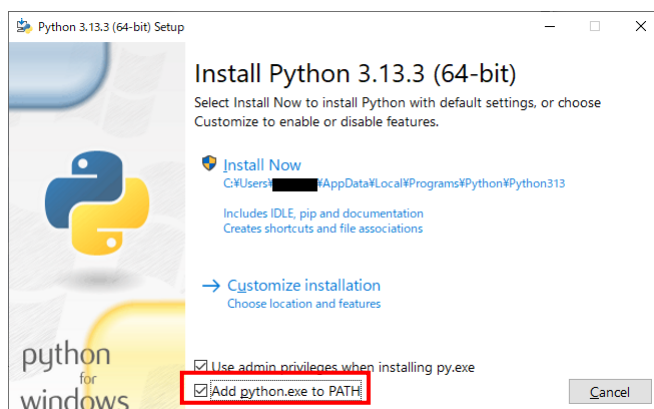
## 3.2 Python のインストール

本サンプルでは、ホスト PC 上で Python が動作する必要があります。

- ① Python の Web ページからインストーラをダウンロードします。

Python Web ページ : <https://www.python.org/>

- ② インストーラを起動して、PC に Python をインストールしてください。  
なお、インストールの際には環境変数へ Python のパスを追加するようにしてください。  
以降、環境変数に Python のパスを追加した状態で説明を行います。



- ③ Python インストール後、コマンドプロンプトを立ち上げ、以下のコマンドを実行します。

"pip install opencv-python numpy"

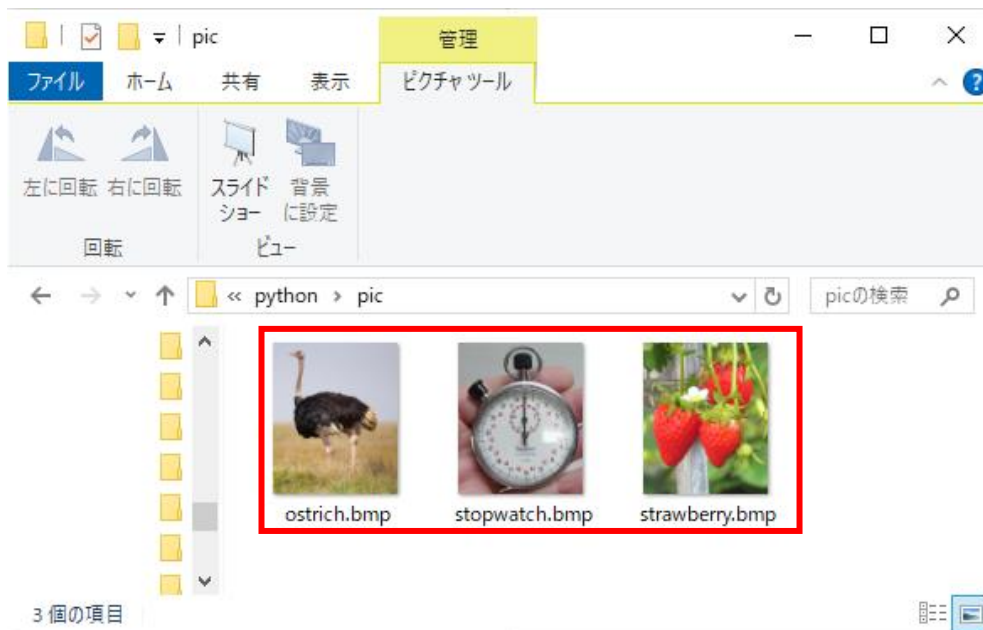
```
コマンドプロンプト
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.
C:\Users\%USER% > pip install opencv-python numpy
```

以上で Python の準備が終了です。

### 3.3 動作説明

本章では、画像分類に使用する画像のプロジェクトへの組み込み方や実行後の結果の確認方法について説明します。  
なお本サンプルプログラムでは、あらかじめ Python をインストールしておく必要があります。  
インストール方法につきましては「3.2 Python のインストール」を参照してください。

- ① 分類したい画像を「Python¥pic」フォルダに保存します。



- ② コマンドプロンプトを開き、カレントフォルダをサンプルプログラムの「Python」フォルダへ移動します。
- ③ 保存した画像から Python スクリプト[image\_resize.py]を実行し、「image\_data\*.h」※を生成します。  
[Please enter the file path of the photo] では保存した画像の Path を、  
[Please enter the image name of the photo] では任意の文字列（画像の名前など）を、  
[Please enter the image number of the photo] ではデータの番号（1~3）を入力してください。  
以下は、ストップウォッチを 1 枚目のデータとして変換した場合の内容です。

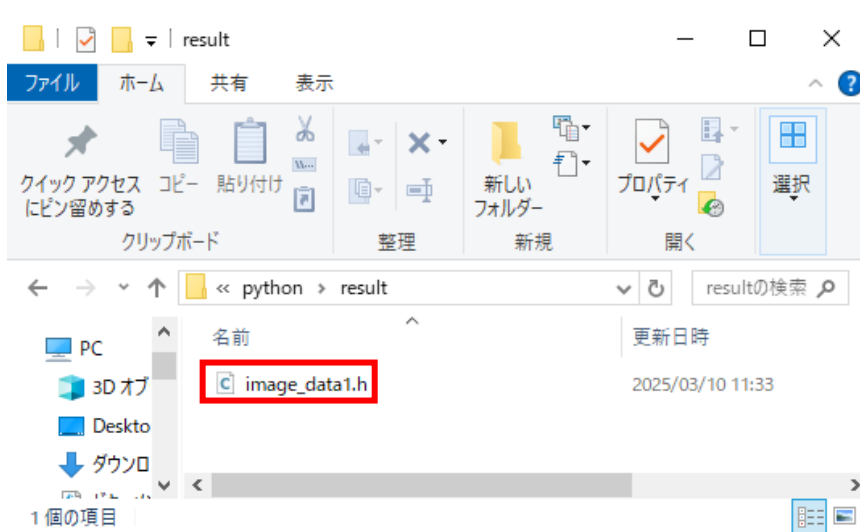
```

C:\> 選択コマンドプロンプト
C:\> cd %Workspace%\nx_n947_tflm_label_image_sample\python
C:\> python image_resize.py
Please enter the file path of the photo
pic/stopwatch.bmp
Please enter the image name of the photo
stopwatch
Please enter the image number of the photo
1
  
```

※ 「image\_data\*.h」の'\*'には、データ番号が入ります。

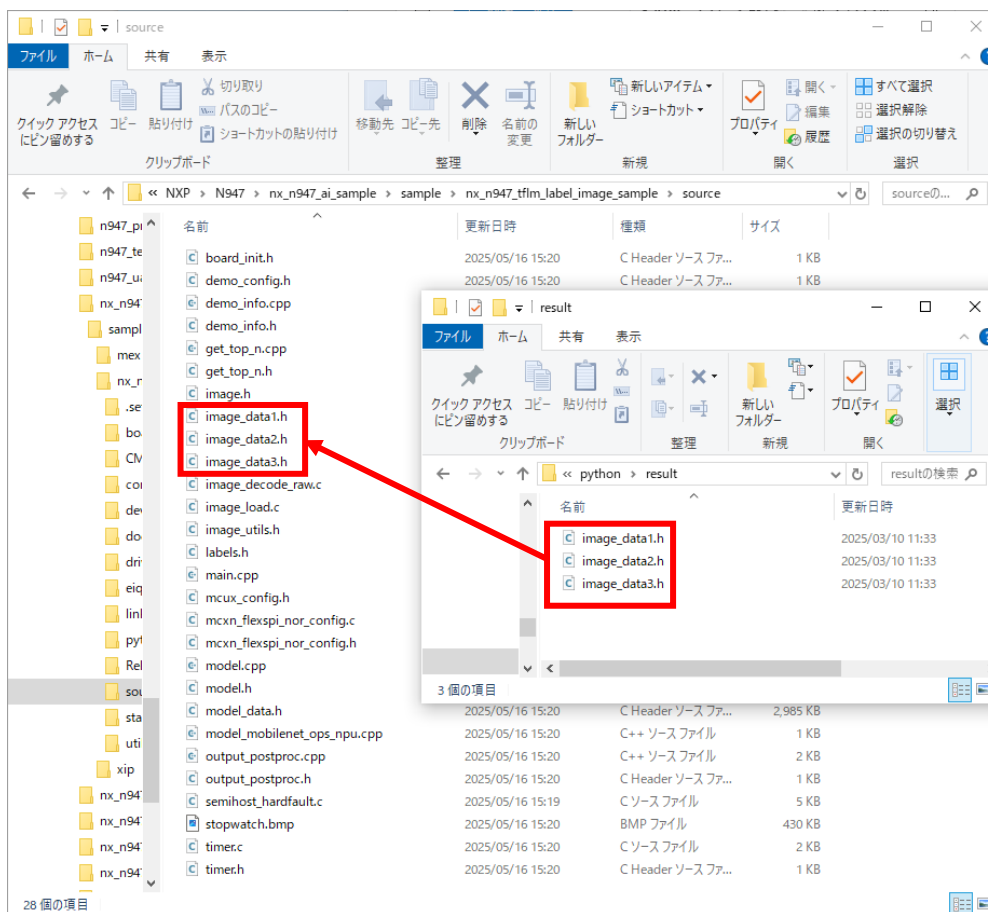
- ④ 実行が完了すると、[Write complete]のメッセージとともに「Python¥result」に「image\_data\*.h」が生成されます。

```
選択コマンドプロンプト
C:¥Workspace¥nx_n947_tflm_label_image_sample¥python>python image_resize.py
Please enter the file path of the photo
pic/stopwatch.bmp
Please enter the image name of the photo
stopwatch
Please enter the image number of the photo
1
Write complete
```



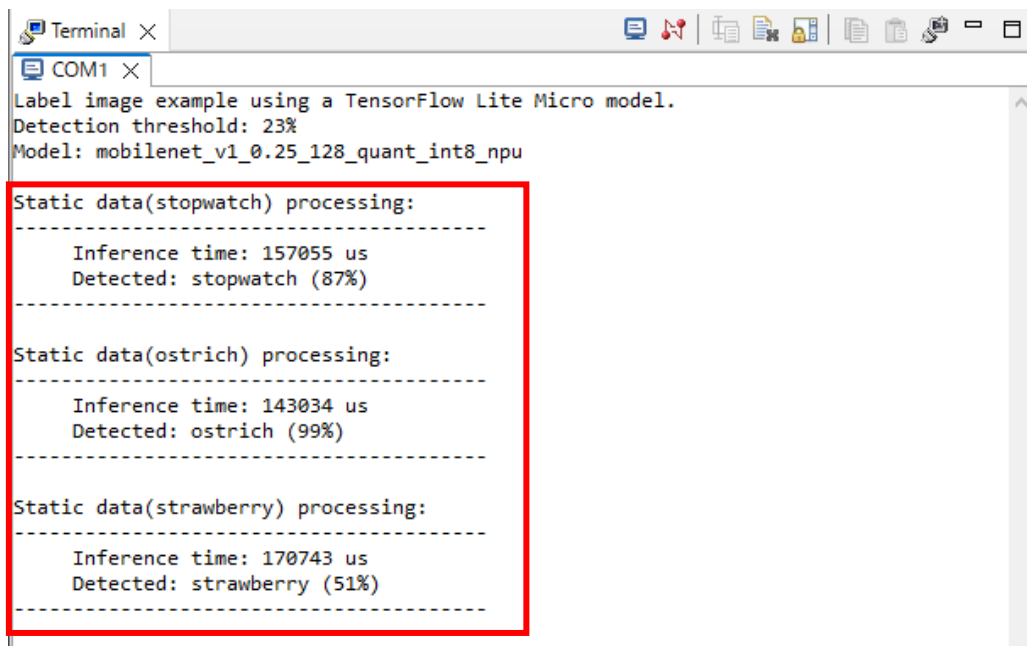
- ⑤ ③ ~ ④の手順で他の 2 枚の画像の変換も行います。

- ⑥ 生成された「image\_data\*.h」を「source¥image」フォルダにある「image\_data\*.h」に上書きします。



- ⑦ 一度サンプルプログラムを[Build]します。  
 ⑧ UART 経由で、CPU ボードと PC を接続し、PC 上でターミナルソフトを起動します。  
 ⑨ CPU ボードに電源を投入、プログラムをダウンロードし、サンプルプログラムを動作させます。

- ⑩ サンプルプログラムが正常に動作した場合は、以下のように 3 種類画像の分類にかかった時間および何が検出されたかが出力されます。



```
Terminal X
COM1 X
Label image example using a TensorFlow Lite Micro model.
Detection threshold: 23%
Model: mobilenet_v1_0.25_128_quant_int8_npu

Static data(stopwatch) processing:
-----
Inference time: 157055 us
Detected: stopwatch (87%)
-----

Static data(ostrich) processing:
-----
Inference time: 143034 us
Detected: ostrich (99%)
-----

Static data(strawberry) processing:
-----
Inference time: 170743 us
Detected: strawberry (51%)
-----
```

### 3.4 MCUXpresso IDE を用いたプロジェクトのビルド・デバッグと外部 ROM への書込み

サンプルプログラムを CPU 上で実行するためには、MCUXpresso IDE 上にサンプルプログラムをインポートし、ビルドを行う必要があります。

MCUXpresso IDE へのサンプルプログラムのインポート、ビルド、デバッグ方法につきましては、

別書 **AN2301 NX シリーズ 開発チュートリアル** に以下の詳細な手順が記されています。

- ・ 3.1.2 インポート方法
- ・ 3.1.3 ビルド構成の変更方法
- ・ 3.1.4 ビルド方法
- ・ 3.1.5 内蔵 RAM を用いたデバッグ、ダウンロード方法
- ・ 3.1.6 外部 ROM を用いたデバッグ、ダウンロード方法
- ・ 5.1 USB インターフェースを使用した書込み方法

## ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本サンプルプログラムで使用されているミドルウェアおよびドライバの著作権は NXP セミコンダクターズ社が保有します。
- ・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

## 商標について

- ・MCX N947 は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・Arm®は Arm Ltd.の登録商標です。
- ・Python は、Python Software Foundation の登録商標または商標です。
- ・MCUXpresso IDE は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・MCUXpresso SDK は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・MCUXpresso Secure Provisioning は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
- ・Windows®10、Windows®11 は、米国 Microsoft Corporation.の商品名称です。  
本文書では下記のように省略して記載している場合がございます。ご了承ください。  
Windows®10 は Windows 10 もしくは Win10  
Windows®11 は Windows 11 もしくは Win11
  
- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト  
〒431-3114  
静岡県浜松市中央区積志町 834  
<https://www.apnet.co.jp>  
E-Mail: [query@apnet.co.jp](mailto:query@apnet.co.jp)