

## 第1章

開発環境のセットアップからマイコン向けサンプルの  
試し方まで

# フレームワーク TensorFlowの準備

大沢 健太郎, 谷本 和俊

オープンソースのマイコン用機械学習フレームワーク TensorFlow Lite for Microcontrollers (以下、TFLM) を、Picoで動かす手順について解説します。TFLMのビルド環境にはセットアップが簡単なラズベリー・パイ4を利用します。

## Pico用のTensorFlow リポジトリがある

Pico用のTFLMは、既にGitHub上に公開されています。ただし、グーグルが運営しているTensorFlowのリポジトリではなく、ラズベリーパイ財団が運営している公式リポジトリ (<https://github.com/raspberrypi/pico-tflmicro>) にあります。

READMEを見ると、このリポジトリは自動生成されたもので、リード・オンリとあり、issueやpull requestは、TensorFlow側のリポジトリにファイルされるとあります。最新版を反映するには、TensorFlowのリポジトリにあるスクリプト `generate.py` (`tensorflow/lite/micro/tools/project/generate.py`) を使って生成するとの記載がありますが、現在、TFLMのリポジトリはTensorFlow本体から独立 (<https://github.com/tensorflow/tflite-micro>) しました。独立後のプロジェクト生成ツール (`tflite-micro/tensorflow/lite/micro/tools/project_generation/create_tflm_tree.py`) もそのままでは動作しませんので、今回はラズベリー・パイ側のリポジトリをそのまま利用します。

Pico用のTFLMには、表1に記載した `hello_world`、`micro_speech`、`magic_wand`、`person_detection` の4つのサンプル・アプリケーションがあります。これらのうち、`hello_world` だけはPicoのLEDで動作が確認できるように実装されています。`hello_world` 以外のサンプルは、Picoにセンサが搭載されていないため、マイク、加速度、カメラなどのセンサからデータを取得するコードは自分で記述する必要があります。まずはビルド環境構築の確認の意味で、Pico単体で動作確認が可能な `hello_world` を動かしてみます。

表1 TFLMのサンプル・アプリケーション一覧

サンプル名	内容
<code>hello_world</code>	入力値に対してsin波(サイン関数)の予測値を返す。PicoのLEDに対して予測値をPWM出力することでPicoのみで動作確認が可能
<code>micro_speech</code>	センサにマイク入力を使用したシンプルなスピーチ(キーワード)認識
<code>magic_wand</code>	加速度センサを使用したジェスチャ推定
<code>person_detection</code>	カメラ(画像入力)を使用した特定の人検出。make時に指定するサンプル名は <code>person_detection_int8</code>

## ビルド環境として利用する ラズパイ4のセットアップ

ビルド環境にラズベリー・パイ4(または3)を利用する場合、Pico用の公式SDK環境をセットアップし、TFLMのリポジトリをクローンするだけで環境を立ち上げることが可能です。以下の手順は、ラズベリー・パイ4の `/home/pi` で実行していきます。

### ● Pico用の公式SDK環境セットアップ

Picoの公式ドキュメント<sup>(1)</sup>の通りにスクリプトをダウンロードし、実行するだけでセットアップが完了します。

```
$ wget https://raw.githubusercontent.com/raspberrypi/pico-setup/master/pico_setup.sh
$ chmod +x pico_setup.sh
$ ./pico_setup.sh
```

### ● TFLMのクローン

ラズベリー・パイ公式のリポジトリから、Pico用のリポジトリをクローンします。

```
$ git clone https://github.com/raspberrypi/pico-tflmicro.git
```

Pico用の環境セットアップ・スクリプトが用意されているため、簡単に環境構築できます。ラズベリー・パイ4上でスクリプトを実行して20分程度(ラズベ