

「まびっどくん」

マニュアル

毎日定時帰宅チーム

2022 年 1 月 7 日

目次

1.	はじめに.....	2
1.1.	目的	2
1.2.	本書を読むにあたって	2
1.3.	開発環境.....	2
2.	実機操作の準備.....	3
2.1.	MICROSD へのデータ保存.....	3
2.2.	スクリプト (MANAGER. PY) の編集	3
2.3.	スクリプト・JSON ファイルの配置	4
2.4.	ハードウェア接続	4
3.	実機を用いたシステムの操作方法	5

1. はじめに

1.1. 目的

本書の目的は、「まぴっどくん」の実行および操作方法を解説することである。

1.2. 本書を読むにあたって

本書は Spresense や Raspberry Pi に関する基礎知識を有した利用者を対象としている。また、各マイコンでのプログラムの詳細な機能等については同ファイル内の各機能解説マニュアルを参考にする。

1.3. 開発環境

本システムの開発環境を以下に示す。

・姿勢推定 + Manager 機能

マイコン	Raspberry Pi 4
Web カメラ	Logicool Web cam C922n
アクセラレータ	Google Coral USB Accelerator
OS	Raspberry Pi OS(32-bit,Released:2021-10-30)
Python	Python-3.7
TensorFlow Lite runtime	2.5.0.post1
OpenCV	4.1.0.25

・表情推定機能

マイコン	Raspberry Pi 4
Web カメラ	Logicool Web cam C922n
OS	Raspberry Pi OS(32-bit,Released:2021-10-30)
Python	Python-3.7
OpenCV	4.1.0.25

・LCD ディスプレイ表示 + 音楽・音声再生機能

マイコン	Spresense メインボード + 拡張ボード
LCD ディスプレイ	ILI9341

2. 実機操作の準備

2.1. microSD へのデータ保存

Spresense の拡張ボードには microSD を使用でき、今回のシステムではハイレゾ音源、アナウンス用音声データ、LCD ディスプレイ表示用データを保存している。

(アナウンス用音声データ)

アナウンス用音声データは GitHub ページの SoundData ディレクトリにある。音声データをダウンロード後 microSD 内に「Sound」というフォルダを作成し、その中にアナウンス用音声データを保存する。

(ハイレゾ音源データ)

ハイレゾ音源データは“happy または neutral”, “angry または surprise”, “sad または fear”の感情に合わせた 3 つのハイレゾ音源データを用意すること。ハイレゾ音源の条件として、ビット数：24-bit, サンプリング周波数：192kHz, ファイル形式：.wav であること。次にファイル名は“happy または neutral”の場合「happy_neutral.wav」, “angry または surprise”の場合「angry_surprise.wav」, “sad または fear”の場合「sad_fear.wav」とすること。最後に用意したデータを microSD 内にあるアナウンス用音声データと同じ「Sound」のフォルダ内に保存する。

(LCD ディスプレイ表示データ)

LCD ディスプレイ表示データは GitHub ページの ImageData ディレクトリにある。表示データをダウンロード後 microSD 内に「image」というフォルダを作成し、その中に LCD ディスプレイ表示データを保存する。

2.2. スクリプト(manager.py)の編集

Google スプレッドシートへデータ送信を行うために RaspberryPi_POSE ディレクトリ内のスクリプトを編集する必要がある。ダウンロードしたスクリプト(manager.py)を開き、15 行目にある“xxxx”の部分各自 GoogleCloudPlatform の設定後に得られる認証情報の入った json ファイル名に変更する。また 17 行目にある“xxxx”の部分には共有設定したスプレッドシートキーへ変更する。そして 18 行目の“xxxx”では共有設定したスプレッドシートのシート名へ変更する。

2.3. スクリプト・json ファイルの配置

上記のスクリプトの編集が完了した後に、姿勢推定機能と表情推定機能のそれぞれの機能を持つ Raspberry Pi の指定ディレクトリへファイルを配置する。

(姿勢推定機能を持つ Raspberry Pi)

事前に Google Coral USB Accelerator を使用するための環境構築が完了しているものとする。詳しい環境構築方法に関しては「RaspberryPi_POSE」のマニュアルを参照のこと。

そして Google Coral の環境構築後に「\$~/coral/project-posenet」のディレクトリに編集をしたスクリプト「manager.py」と Google スプレッドシートの認証情報が含まれる json ファイル、そして姿勢推定を行うスクリプト「org_pose_camera.py」の3つ配置する。

(表情推定機能を持つ Raspberry Pi)

事前に表情推定を行うため TensorFlow, Keras, OpenCV 等の環境構築が完了しているものとする。詳しい動作環境に関しては「RaspberryPi_FACE」のマニュアルを参照のこと。

そして環境構築後に GitHub ページの RaspberryPi_FACE の「emotion_detect」ファイルをダウンロードし、好きな場所へ保存すること。

2.4. ハードウェア接続

RaspberryPi や Spresense, ボタンや LCD 等の結線図を以下に示す。

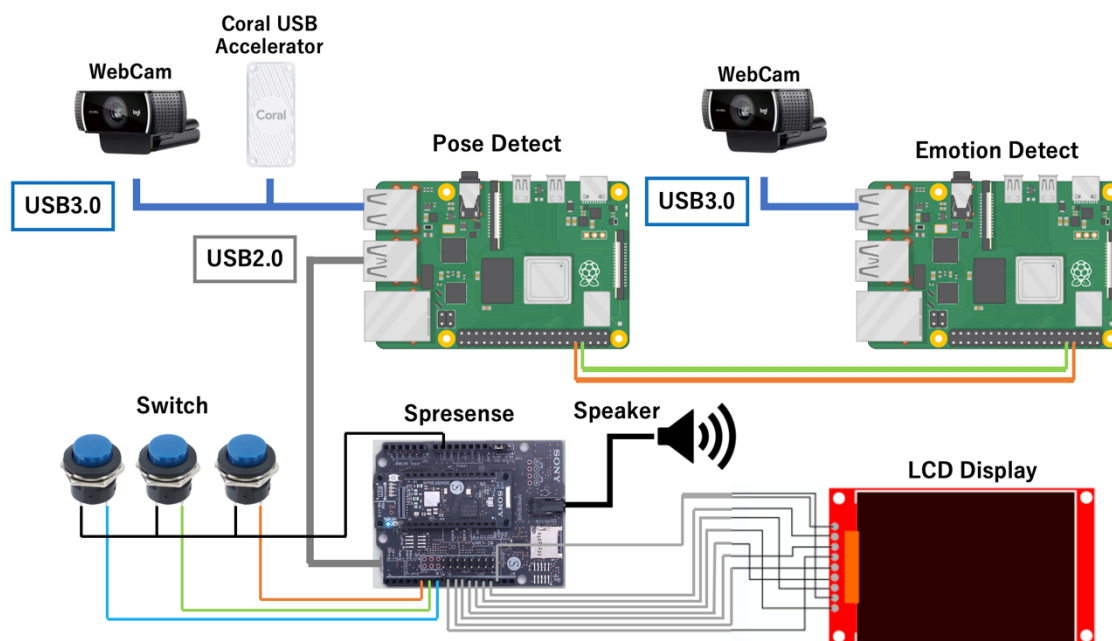


図1 結線図

3. 実機を用いたシステムの操作方法

実機でのシステムの操作を行う前に、上記の「実機操作の準備」が完了していることを前提とする。

- ① 表情推定を行う RaspberryPi にて「send_cap.py」のスク립トを実行する。
- ② 姿勢推定を行う RaspberryPi にて「manager.py」のスク립トを実行する。

上記のスク립トが実行されると、「まぴっどくん」から操作案内のアナウンスが始まるため、アナウンスに沿って操作を行う。詳しいボタン操作に関してはプレゼンテーション資料を参照のこと。