



振動計測システム モニタリングアプリ ユーザーズガイド

1. 本評価ボード・キット、開発ツールは、お客様での技術的評価、動作の確認および開発のみに用いられることを想定し設計されています。それらの技術評価・開発等の目的以外には使用しないでください。本品は、完成品に対する設計品質に適合していません。
2. 本評価ボード・キット、開発ツールは、電子エンジニア向けであり、消費者向け製品ではありません。お客様において、適切な使用と安全に配慮願います。弊社は、本品を用いることで発生する損害や火災に対し、いかなる責も負いかねます。通常の使用においても、異常がある場合は使用を中止してください。
3. 本評価ボード・キット、開発ツールに用いられる部品は、予告なく変更されることがあります。

● 本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告なく変更することがあります。

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りします。
2. 弊社製品のご購入およびご使用にあたりましては、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページなどを通じて公開される最新情報に常にご注意ください。
3. 本資料に掲載されている応用回路、プログラム、使用方法などはあくまでも参考情報です。お客様の機器・システムの設計において、応用回路、プログラム、使用方法などを使用する場合には、お客様の責任において行ってください。これらに起因する第三者の知的財産権およびその他の権利侵害ならびに損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の知的財産権およびその他の権利の実施権の許諾を行うものではありません。
4. 弊社は常に品質、信頼性の向上に努めていますが、一般的に半導体製品は誤動作または故障する場合があります。弊社製品のご使用にあたりましては、弊社製品の誤動作や故障により生命・身体に危害を及ぼすこと又は財産が侵害されることのないように、お客様の責任において、お客様のハードウェア、ソフトウェア、システムに必要な安全設計を行うようお願いいたします。なお、設計および使用に際しては、弊社製品に関する最新の情報(本資料、仕様書、データシート、マニュアル、弊社ホームページなど)をご確認いただき、それに従ってください。また、上記資料などに掲載されている製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価を行い、お客様の責任において適用可否の判断をお願いいたします。
5. 弊社は、正確さを期すために慎重に本資料およびプログラムを作成しておりますが、本資料およびプログラムに掲載されている情報に誤りがないことを保証するものではありません。万一、本資料およびプログラムに掲載されている情報の誤りによってお客様に損害が生じた場合においても、弊社は一切その責任を負いかねます。
6. 弊社製品の分解、解析、リバースエンジニアリング、改造、改変、翻案、複製などは堅くお断りします。
7. 弊社製品は、一般的な電子機器(事務機器、通信機器、計測機器、家電製品など)および本資料に個別に掲載されている用途に使用されることを意図して設計、開発、製造されています(一般用途)。特別な品質、信頼性が要求され、その誤動作や故障により生命・身体に危害を及ぼす恐れ、膨大な財産侵害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある以下の特定用途に使用されることを意図していません。お客様に置かれましては、弊社製品を一般用途に使用されることを推奨いたします。もし一般用途以外の用途で弊社製品のご使用およびご購入を希望される場合、弊社はお客様の特定用途に弊社製品を使用されることへの商品性、適合性、安全性について、明示的・黙示的に関わらずいかなる保証を行うものではありません。

【特定用途】

宇宙機器(人工衛星・ロケットなど) / 輸送車両並びにその制御機器(自動車・航空機・列車・船舶など)
医療機器(本資料に個別に掲載されている用途を除く) / 海底中継機器 / 発電所制御機器 / 防災・防犯装置
交通用機器 / 金融関連機器
上記と同等の信頼性を必要とする用途

8. 本資料に掲載されている弊社製品および当該技術を国内外の法令および規制により製造・使用・販売が禁止されている機器・システムに使用することはできません。また、弊社製品および当該技術を大量破壊兵器等の開発および軍事利用の目的その他軍事事務等に使用しないでください。弊社製品または当該技術を輸出または海外に提供する場合、「外国為替及び外国為替法」、「米国輸出管理規則(EAR)」、その他輸出入関連法令を遵守し、係る法令の定めるところにより必要な手続きを行ってください。
9. お客様が本資料に掲載されている諸条件に反したことに起因して生じたいかなる損害(直接・間接を問わず)に関して、弊社は一切その責任を負いかねます。
10. お客様が弊社製品を第三者に譲渡、貸与などをしたことにより、損害が発生した場合、弊社は一切その責任を負いかねます。
11. 本資料についての詳細に関するお問合せ、その他お気付きの点などがありましたら、弊社営業窓口までご連絡ください。
12. 本資料に掲載されている会社名、商品名は、各社の商標または登録商標です。

2022.08

©Seiko Epson Corporation 2023, All rights reserved.

商標

- Raspberry Pi is a trademark of Raspberry Pi Ltd.
- マイクロソフト、Windows は、マイクロソフト グループの企業の商標です。
- EPSON はセイコーエプソン株式会社の登録商標です。
- その他の製品名は各社の商標または登録商標です。

目次

商標.....	3
改訂履歴.....	5
1. 関連文書	6
2. はじめに	7
3. 仕様	8
3.1. 動作確認環境	8
3.2. 入力仕様	8
3.3. 出力仕様	8
3.3.1. ホーム画面	8
3.3.2. グラフ画面	9
3.4. 設定項目	9
4. Raspberry Pi へのセットアップ.....	10
4.1. ファイル・フォルダ構成	10
4.2. 事前準備	10
4.3. Raspberry Pi へプログラムを転送	10
4.4. Raspberry Pi 上に実行環境を構築	10
4.4.1. Python 仮想環境の作成	11
4.4.2. パッケージ インストール.....	11
4.4.3. アプリケーションの設定	11
4.4.4. サービス登録ファイルのインストール.....	11
5. アプリケーションの実行	12
5.1. 事前準備	12
5.2. アプリケーションの起動	12
5.3. 振動の計測	12
5.4. ハードウェア状態モニターツールの実行.....	12
6. Appendix : 開発者向けガイド	13
6.1. プログラム構成	13
6.1.1. Subscriber クラス	13
6.1.2. DataManager クラス.....	13
6.1.3. Dash アプリケーション.....	13
6.1.4. DataSet クラス	13
6.2. ローカルでの実行.....	14
6.2.1. 動作環境	14
6.2.2. ローカル環境構築.....	14
6.2.3. プログラムの実行.....	15
6.3. 参考リンク	15
7. お問い合わせ.....	16

改訂履歴

Rev. No.	改訂日	Page	改訂内容
20250331	2025/3/31	ALL	初版制定 MSG005-001a_v1.0.0 リリースに対応
20250731	2025/7/31	---	MSG005-001a_v1.0.1 リリースに対応
		10	転送ファイルが不足していたため記載を修正
		11	OS 起動時にサービス起動する設定が不足していたため、サービス 定義ファイルを修正

1. 関連文書

1. 『Raspberry Pi 社製品を利用した振動計測システム セットアップマニュアル』 Rev.20250331
2. 『Raspberry Pi 社製品を利用した振動計測システム オペレーションマニュアル』 Rev.20250331
3. 『VC 判定 振動計測システム ユーザーズガイド』 Rev.20250331

2. はじめに

本ユーザーズガイドは、「振動計測システム モニタリングアプリ」について、以下の項目を説明するものです。

- 仕様
- セットアップ方法
- 実行方法
- 開発者向けガイド

「振動計測システム モニタリングアプリ」（以下「本アプリ」と記載します）は、「Raspberry Pi 社製品を利用した振動計測システム」や「VC 判定 振動計測システム」（以下この二つをまとめて「計測アプリ」と記載します）がインストールされた Raspberry Pi 上で動作する Web アプリケーションです。

計測アプリや計測アプリに付属の「ハードウェア状態モニターツール」から、MQTT のメッセージを媒介する「MQTT ブローカー」を通じてメッセージを受け取り、計測の状態、接続されている振動センサーの状態、VC 判定結果等を Web 画面に表示します。

PC から Web ブラウザで Raspberry Pi にアクセスすることで、Web アプリケーションを表示できます。

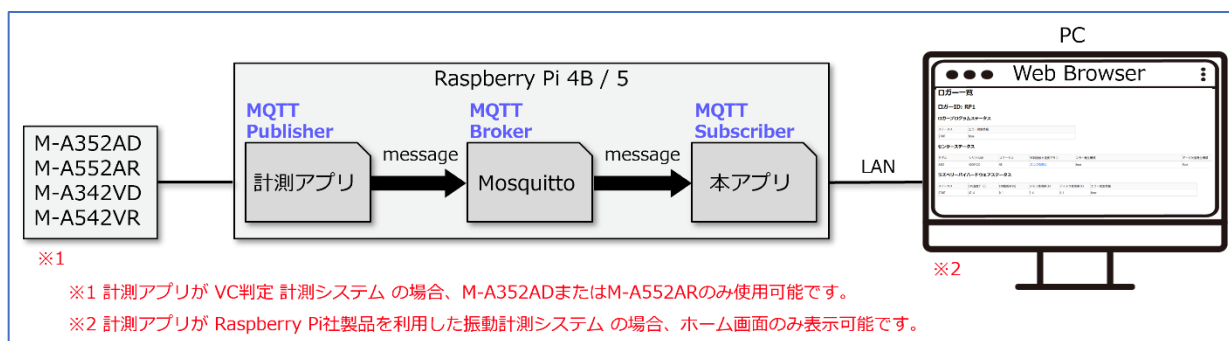


図 2-1 アプリケーション イメージ図

3. 仕様

3.1. 動作確認環境

本アプリは以下の Raspberry Pi で動作を確認しています。

- Raspberry Pi 4B
- Raspberry Pi 5

また、以下の Web ブラウザで動作を確認しています。

- Google Chrome
- Microsoft Edge

3.2. 入力仕様

本アプリは以下のツールに対応し、メッセージ送信された情報を受信します。

- 「Raspberry Pi 社製品を利用した振動計測システム」
- 「Raspberry Pi 社製品を利用した振動計測システム」に付属の「ハードウェア状態モニター」
- 「VC 判定 振動計測システム」

各々のメッセージ仕様については、関連文書 2, 3 を参照ください。

3.3. 出力仕様

本アプリは Web アプリケーションのため、Web 画面の仕様を記載します。

本アプリは 2 つの Web 画面から構成されています。

いずれの Web 画面も、計測アプリから送信されたメッセージを 1 秒サイクルで画面に反映します。

3.3.1. ホーム画面

MQTT ブローカーに接続している計測アプリの情報のうち、ロガーの状態や振動センサーの状態を表示する画面です。

Logger List

Logger ID: RP1

Logger Program Status

Status	Error
START	None

Sensor Status

Model	Serial number	Status	VC judgment value and velocity graph	Error	Data loss	Element abnormality
A352	00004020	OK	Disc. 9.2a0b	None	None	None

Raspberry Pi Hardware Status

Status	CPU temperature (degC)	CPU usage (%)	Memory usage (%)	Disk usage (%)	Error
START	52.35	1.6	5.3	0.6	None

図 3-1 ホーム画面

1. 接続しているロガーの ID を表示します
2. 接続しているロガーの状態を表示します
3. ロガーに接続している振動センサーの状態を表示します
「グラフを開く」を押下すると、後述のグラフ画面を開きます
4. ラズベリーパイの CPU 使用率等を表示します

3.3.2. グラフ画面

接続している振動センサーの VC 判定値と速度グラフを表示する画面です。

※ 注意： この画面の内容は VC 判定 振動計測システムを実行している場合のみ表示されます。

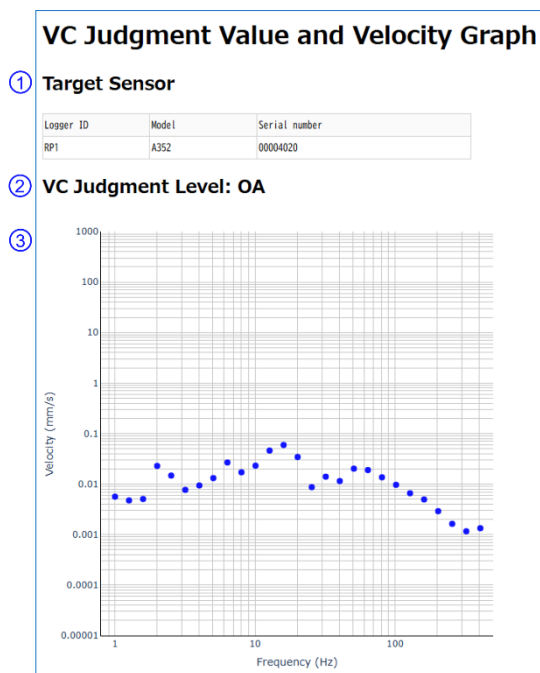


図 3-2 グラフ画面

1. グラフを表示している振動センサーの情報を表示します
2. 表示している振動センサーが計測した振動データの VC 判定値を表示します
3. 表示している振動センサーが計測した振動データの速度グラフを表示します

3.4. 設定項目

本アプリの設定は、本アプリをセットアップしたディレクトリの「.env」ファイルから読み込みます。設定ファイルに設定可能な項目は以下のとおりです。

表 3-1

設定項目名	説明	デフォルト設定値	補足
LOG_LEVEL	出力するログレベル	INFO	
APP_HOST	本アプリにアクセスするホスト名	192.168.1.52	Web ブラウザからアクセスするホスト名を指定する
APP_PORT	本アプリにアクセスするポート番号	8050	
MAX_MESSAGE_COUNT	複数の値を保持できる項目の保持件数	3	エラー等の発生を保持できる件数
MQTT_BROKER_HOST	MQTT ブローカーのホスト名	localhost	
MQTT_BROKER_PORT	MQTT ブローカーのポート番号	1883	

4. Raspberry Pi へのセットアップ

本アプリを Raspberry Pi 上にセットアップする手順を説明します。

4.1. ファイル・フォルダ構成

ダウンロードした ZIP ファイルを展開したファイル・フォルダ構成は以下のとおりです。

展開フォルダ	
└── pyproject.toml	Python のプロジェクト定義ファイル
└── sandbox	動作確認や調査用のプログラムを置くフォルダ
└── src	本アプリ本体のソースコードを置くフォルダ
└── common	アプリ内で共通的に使うプログラムのコード
└── data	アプリ内で扱うデータを定義したコード
└── infra	MQTT ブローカーから受信したメッセージを処理するコード
└── ui	アプリケーションの Web UI を作成するコード
└── utils	その他複数箇所で利用するコード
└── tests	本アプリのテストコードを置くフォルダ

図 4-1 ファイル・フォルダ構成

4.2. 事前準備

本アプリのセットアップを開始する前に、関連文書 1 および 3 を参照し、以下の準備を行ってください。

- Raspberry Pi と PC をネットワークで接続する
- Raspberry Pi からインターネットに接続する
- Raspberry Pi に計測アプリをインストールする
- Raspberry Pi に MQTT ブローカー（Mosquitto）をインストールする
- 計測アプリの設定で、MQTT ブローカーへのメッセージ送信を有効にする

本ガイドの以下の手順では、次の想定で記載しています：

- Raspberry Pi に作成したユーザー名：`pi`
- Raspberry Pi に設定した固定 IP アドレス：`192.168.1.52`

ユーザー名・固定 IP アドレスに異なるものを設定した場合には、適宜読み替えて実行してください。

4.3. Raspberry Pi へプログラムを転送

以下のコマンドを実行して、本アプリの実行に必要なファイルを Raspberry Pi に転送します。

1. Raspberry Pi 内の /app に MSG005-001a ディレクトリを作成します

▶ `ssh pi@192.168.1.52 "mkdir -p /app/MSG005-001a"`

2. 作成した /app/MSG005-001a へプログラムを転送します

▶ `scp -r bin src pyproject.toml .env.default pi@192.168.1.52:/app/MSG005-001a`

4.4. Raspberry Pi 上に実行環境を構築

以降は Raspberry Pi 上で操作するため、Raspberry Pi にログインします。

- ▶ `ssh pi@192.168.1.52`

4.4.1. Python 仮想環境の作成

本アプリが使用するパッケージをインストールするため、以下のコマンドを入力し Python 仮想環境を作成します。

- ▶ `cd /app/MSG005-001a`
- ▶ `python -m venv venv`
- ▶ `source venv/bin/activate`
- ▶ `pip install -U pip`

4.4.2. パッケージ インストール

以下のコマンドを実行し、仮想環境に本アプリが使用するパッケージをインストールします。

- ▶ `pip install .`

4.4.3. アプリケーションの設定

以下のコマンドで設定ファイルのテンプレート `.env.default` をコピーし、設定ファイルを作成します。

- ▶ `cp .env.default .env`

設定ファイルをエディタで開き、設定値を変更します。

設定項目は「3.4 設定項目」を参照してください。

4.4.4. サービス登録ファイルのインストール

以下のコマンドでサービス登録ファイルをシステムにインストールします。

- ▶ `sudo cp bin/logger_monitor.service /etc/systemd/system`

※ 注意： サービス登録ファイルは Python 仮想環境が `/app/MSG005-001a/venv` に作成されていることを前提にしています。異なるパスに仮想環境を作成した場合は、サービス登録ファイルの以下の行を適切なパスに修正してください。

- ▶ `ExecStart=/app/MSG005-001a/venv/bin/python -m raspi_web`

Raspberry Pi の起動時に本アプリを自動的に起動するには、以下のコマンドを入力します。

- ▶ `sudo systemctl enable logger_monitor`

5. アプリケーションの実行

5.1. 事前準備

本アプリを実行する前に、関連文書 1 および 3 を参照し、以下の準備を行ってください。

- Raspberry Pi と PC をネットワークで接続する
- Raspberry Pi に振動センサーを接続する

5.2. アプリケーションの起動

「4.4.4 サービス登録ファイルのインストール」の手順で OS 起動時に本アプリが自動的に起動するよう設定している場合、設定ファイルの APP_HOST, APP_PORT で指定したホスト名とポート番号で Web アプリが起動します。

▶ 例：192.168.1.52:8050

OS 起動時に起動するよう設定していない場合、以下のコマンドで本アプリを起動します。

▶ `sudo systemctl start logger_monitor`

※ 注意：アプリケーションの起動時、以下の警告がログに出力されます。これは内部で使用する Web サーバーは開発用途での使用を想定したもので、公開環境での使用を想定していないことを意味しています。外部に公開されるネットワーク環境での使用は避けてください。

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

PC で Web ブラウザを起動し、URL バーに以下を入力することで本アプリにアクセスできます。

▶ `http://192.168.1.52:8050/`

※ 注意：本アプリにアクセスできない場合、問題の解析のために Raspberry Pi にログインし、以下のコマンドを入力してログ出力を確認してください。

`systemctl status logger_monitor`

5.3. 振動の計測

この状態で計測アプリを実行して振動を計測することで、計測アプリの状態や振動センサーの状態などを本アプリで確認することができます。

計測方法については関連文書の 2 または 3 を参照してください。

5.4. ハードウェア状態モニターツールの実行

この状態でハードウェア状態モニターツールを実行することで、Raspberry Pi の CPU 使用率やメモリ使用率などを本アプリで確認することができます。

実行方法については 2 を参照してください。

6. Appendix : 開発者向けガイド

この章は開発者向けガイドとして、本アプリに機能追加や改修を行う際に必要となる情報を記載します。

- プログラム構成
- PC ローカルへの環境構築
- PC ローカルでのプログラム実行

6.1. プログラム構成

本アプリのプログラムは、大きく以下の3つのモジュールから構成されています。

- Subscriber クラス (./src/raspi_web/infra/subscriber.py)
- DataManager クラス (./src/raspi_web/infra/data_manager.py)
- Dash アプリケーション (./src/raspi_web/__main__.py)

また、プログラム内で扱うデータを定義したクラスが1つあります。

- DataSet クラス (./src/raspi_web/data/data_set.py)

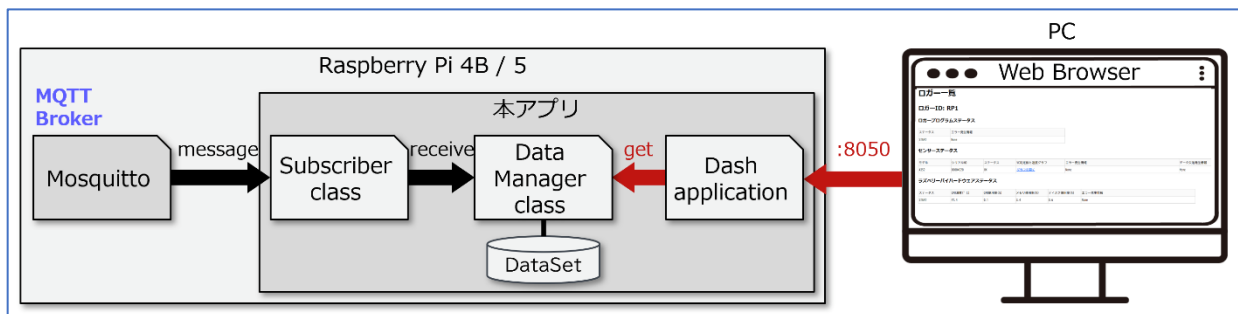


図 6-1 プログラム構成 イメージ図

各項目について、以下に説明します。

6.1.1. Subscriber クラス

MQTT ブローカーからメッセージを受け付け、DataManager にメッセージを渡します。

6.1.2. DataManager クラス

Subscriber から受け取ったメッセージに対して、トピックに応じたメッセージの解析を行い、Dash 画面で表示するデータを蓄積します。また、Dash 画面にデータを渡すデータ取得メソッドを備えています。

新たなトピックを追加する場合は、ソースコード内の `_TOPIC_PATTERN_LIST` にトピックパターンとハンドラーを追加してください。

6.1.3. Dash アプリケーション

Web アプリケーションの UI を構築しています。DataManager から 1 秒おきにデータを取得し、画面に反映させています。

UI を追加・変更する場合は、こちらを確認してください。

6.1.4. DataSet クラス

DataManager や Dash アプリケーションで扱うデータの型を定義しています。

6.2. ローカルでの実行

本アプリに機能追加や修正を行う場合は、ローカル PC にて作業を行います。以下の内容を説明します。

- 本アプリをローカルにセットアップする手順
- ローカルで MQTT ブローカーを起動する手順

6.2.1. 動作環境

ローカルでの実行には以下の環境が必要になります。

- Python 3.11
- Docker Desktop など Docker コンテナを実行できる環境
- Docker Compose v2 （v2.31.0 で動作することを確認しています）

6.2.2. ローカル環境構築

ダウンロードした ZIP ファイルをローカル PC の任意のフォルダに展開します。そのフォルダで以下のステップに従い環境を構築します。

6.2.2.1. Python 仮想環境作成

Windows の場合：

- PowerShell を起動し、以下のコマンドを実行します。
 - `py -3.11 -m venv venv`
 - `venv\scripts\activate.ps1`

Mac や Linux の場合：

- ターミナルを起動し、以下のコマンドを実行します。
 - `python3.11 -m venv venv`
 - `source venv/bin/activate`

6.2.2.2. パッケージ インストール

本アプリが使用するパッケージをインストールします。

Windows の場合：

- `python -m pip install -U pip`
- `pip install -e .`

Mac や Linux の場合：

- `pip install -U pip`
- `pip install -e .`

6.2.2.3. MQTT ブローカー (mosquitto) 起動

この手順では MQTT ブローカーを Docker コンテナとして起動します。./sandbox/mqtt/compose.yml を使用します。

Windows の場合：

- `cd sandbox\mqtt`
- `docker compose up -d`
- `cd ..\..`

Mac や Linux の場合：

- `cd sandbox/mqtt`
- `docker compose up -d`

▶ `cd ../../`

6.2.2.4. 設定ファイルの作成

以下のコマンドで `.env.default` をコピーし、設定ファイル `.env` を作成します。

▶ `cp .env.default .env`

作成された設定ファイルをエディタで開き、`APP_HOST` を `localhost` に変更します。

その他、必要に応じて設定を変更します。

6.2.3. プログラムの実行

6.2.3.1. アプリケーションの実行

以下のコマンドを実行し、Web アプリケーションを起動します。

▶ `python -m raspi_web`

Web ブラウザから `http://localhost:8050/` にアクセスすることで、Web 画面を確認できます。

6.2.3.2. サンプルメッセージの送信

サンプルのメッセージを送信するプログラムを `sandbox/mqtt/publish.py` に用意しています。

プログラム下部の `logger_id`, `model`, `serial`, `topic_type` の項目を送りたいメッセージに合わせて変更し、以下のコマンドで実行します。

▶ `python ./sandbox/mqtt/publish.py`

6.2.3.3. サンプルメッセージの削除

サンプルメッセージは MQTT ブローカーに対してメッセージを「保持 (retain)」する設定で送信しています。保持されたメッセージを削除するプログラムを `sandbox/mqtt/none_publish.py` に用意しています。

プログラム下部の `logger_id`, `model`, `serial`, `topic_type` の項目を削除したいメッセージに合わせて変更し、以下のコマンドで実行します。

▶ `python ./sandbox/mqtt/none_publish.py`

6.3. 参考リンク

- Dash Python User Guider (<https://dash.plotly.com/>)
 - ▶ Web アプリケーションフレームワーク Dash の公式サイト

7. お問い合わせ

セイコーエプソン株式会社

営業本部 MD 営業部

インターネットによるお問い合わせ先

https://www.epson.jp/prod/sensing_system/contact/