

EPSON

**ISO 20816 に基づく
機械状態監視システム
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用用途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は、「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	9
3.1. 動作確認環境	9
3.2. 対応センサー	9
3.3. 入力仕様	9
3.4. 集計仕様	10
3.5. 出力仕様	10
3.5.1. 集計サマリーファイル	10
3.5.2. 集計データメッセージ	11
3.5.3. 集計警告メッセージ	14
3.6. 設定項目	15
4. Raspberry Pi へのセットアップ	17
4.1. ファイル・フォルダ構成	17
4.2. 事前準備	17
4.3. Raspberry Pi へ ZIP ファイルを転送	17
4.4. Raspberry Pi にアプリケーションをインストール	17
4.5. アプリケーション設定	18
5. アプリケーション実行	19
5.1. アプリケーションの手動実行	19
5.2. アプリケーションのスケジュール実行	19
6. Appendix : 開発者向けガイド	20
6.1. 開発環境構築	20
6.1.1. Python 仮想環境作成	20
6.1.2. パッケージ インストール	20
6.2. プログラム構成	21
7. お問い合わせ	22

改訂履歴

Rev. No.	改訂日	Page	改訂内容
20251130	2025/11/30	ALL	初版制定 MSG007-001a_v1.0.0 リリースに対応

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.20250731
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.20250731
3. 『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』 Rev.20251130

2. はじめに

本ユーザーズガイドは、「ISO 20816 に基づく機械状態監視システム」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

「ISO 20816 に基づく機械状態監視システム」（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」（以下「ラズパイロガー」と記載します）がインストールされた Raspberry Pi 上で動作するプログラムで、ラズパイロガーが計測・保存したデータファイルを集計するアプリケーションです。

ラズパイロガーによる計測実行を1時点として各センサーの計測値を統計処理し、複数の計測実行にわたる各センサーの統計値を時系列の変化として集計します。また、あらかじめ各センサーの計測軸ごとに閾値を設定することで、閾値を超過した場合に警告を出力することが可能です。

データファイルの集計結果や警告結果は、MQTT メッセージとして「MQTT ブローカー」に送信します。MQTT ブローカーに送信されたメッセージは、「振動計測システム モニタリングアプリ」（以下「モニタリングアプリ」と記載します）を使用して、PC から Web ブラウザで確認することができます。

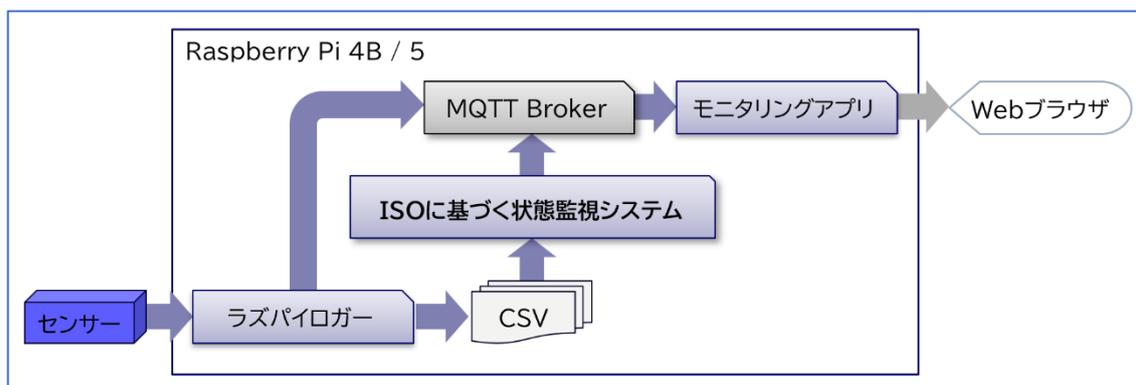


図 2-1 全体構成図

本アプリを用いることで、ISO 20816 で定められた回転機器の状態監視が可能になります。ISO 20816 では振動を用いた機械状態監視の方法が定められています。具体的には、対象機器の振動を定期的に計測し、振動の大きさやその変化量、変化の速さを監視します。予め設定した閾値を超過した場合には、対象装置の緊急停止や保守作業を行います。

本アプリでは以下の基準値・閾値を用いて状態監視を行います。これらの値は事前の振動計測結果および ISO 規格や現場の経験則などをもとに、予め決定する必要があります。

表 2-1 振動による状態監視における基準値・閾値

基準値・閾値	説明
ベースライン値	正常状態の振動の大きさを示す基準値。
アラーム値	振動の大きさ／変化量に対する閾値。アラーム値を超過した場合、慎重に装置運転を続行し、その間に原因特定・対策を講じる必要がある。
トリップ値	振動の大きさ／変化量に対する閾値。トリップ値を超過した場合、それ以上の運転は危険であると判断される。

なお本アプリでは、振動の大きさとして RMS（二乗平均平方根）を用います。本アプリをカスタマイズすることで、RMS 以外の指標を用いた状態監視も可能です。

参考文献：振動技術研究会, ISO 基準に基づく機械設備の状態監視と診断（振動 カテゴリー II）【第 5 版】, 2019 年 9 月 30 日発行

3. 仕様

3.1. 動作確認環境

- Raspberry Pi 4B
- Raspberry Pi 5

3.2. 対応センサー

- M-A342VD / M-A542VR
- M-A352AD / M-A552AR
- M-A370AD

3.3. 入力仕様

本アプリは、ラズパイロガーの「データ保存フォルダ」に作成された「計測フォルダ」と、その中に保存された「計測情報ファイル」と「計測データファイル」を入力とします。

- 「計測フォルダ」が示す「計測開始日時」を、その計測の時点情報として使用します。
- 「計測フォルダ」内の「計測情報ファイル」を手掛かりに、対象のセンサーを識別します。
- 「計測情報ファイル」に関連する「計測データファイル」のうち、最初に作成されたものを使用します。

※ 注意： 「計測フォルダ」「計測情報ファイル」「計測データファイル」がリネームされている場合、ターミナルに警告が出力されてそれらが使用されないか、処理対象として検出されません。

※ 注意： 「計測データファイル」は最初に作成されたもののみ使用するため、設定項目「SKIP_SECONDS」と「SUMMARY_SECONDS」で指定した秒数の合計を十分に上回るデータが記録されている必要があります。合計秒数にデータが満たない場合、ターミナルに警告が出力されその計測データは使用されません。

※ 注意： センサーの出力物理量が Raw モードでない場合、警告が出力されその計測データは使用されません。

参考までに、ラズパイロガーのデータ保存フォルダと計測データファイル類の例を以下に示します。

/home/pi/measure	データ保存フォルダ
├── 20250801_091011	計測フォルダ (計測開始日時)
├── A342_00000100_info.csv	計測情報ファイル (センサー1)
├── A342_00000100_R1_0_250801_091012.csv	計測データファイル 1 (センサー1)
├── A342_00000100_R1_0_250801_092012.csv	計測データファイル 2 (センサー1)
├── A352_00000200_info.csv	計測情報ファイル (センサー2)
├── A352_00000200_R1_1_250801_091012.csv	計測データファイル 1 (センサー2)
├── A352_00000200_R1_1_250801_092012.csv	計測データファイル 2 (センサー2)
└── measure.log	ログファイル
├── 20250801_101011	次の計測フォルダ
├── A342_00000100_info.csv	(同上)
├── A342_00000100_R1_0_250801_101012.csv	
├── A342_00000100_R1_0_250801_102012.csv	
├── A352_00000200_info.csv	
├── A352_00000200_R1_1_250801_101012.csv	
├── A352_00000200_R1_1_250801_102012.csv	
└── measure.log	

図 3-1 ラズパイロガー データ保存フォルダ 例

3.4. 集計仕様

本アプリは、「3.3 入力仕様」に記載のように、ラズパイロガーの「データ保存フォルダ」を以下の詳細に従い使用します。

- 「データ保存フォルダ」から「計測フォルダ」をフォルダ名の昇順で検索する。
 - 設定項目「MAX_MEASUREMENTS」に正の数が指定されている場合、最新の計測からその個数分を集計対象として絞り込む。
- 見つかった「計測フォルダ」のフォルダ名から「計測開始日時」を取得し、集計の時点情報とする。
- 「計測フォルダ」の中の「計測情報ファイル」を検索し、計測に使用したセンサー情報とする。
 - 使用したセンサーが設定項目「SENSOR_MODEL」に該当しない場合、スキップする。
- 「計測情報ファイル」に関連する最初の「計測データファイル」を検索し、入力データとして処理する。
- すべての「計測情報ファイル」を処理したのち、次の「計測フォルダ」に移る。

※ 注意： 「データ保存フォルダ」に「計測フォルダ」がない場合、何も集計されません。

本アプリは、上記により検索したファイルに対して、以下の集計処理を行います。

- 「計測情報ファイル」から「SPS」を取得し、設定項目「SKIP_SECONDS」と「SUMMARY_SECONDS」に必要なデータ件数を算出する。
- 「計測データファイル」の先頭から「SKIP_SECONDS」のデータ件数を読み飛ばし、それ以降の「SUMMARY_SECONDS」のデータ件数について、X, Y, Z の各軸データと、XYZ から算出した三軸合成値 (C) のRMS (二乗平均平方根) を算出する。
- この方法で集計したデータについて、センサーごとに各時点の集計値として時系列に蓄積する。

※ 注意： 計測に使用した個々のセンサーについて、「計測情報ファイル」から得られる物理量 (PHYSICAL) の値が途中で変わっている場合、ターミナルに警告が出力されその計測データは集計値として使用されません。

本アプリは、すべての「計測フォルダ」を検索した後、集計したデータを時系列にたどり変化量の計算を行います。

- 個々のセンサーの軸ごとのデータについて、前回値と今回値の差の絶対値 (前回値がない初回は0) を変化量として保持する。

3.5. 出力仕様

本アプリは、集計し蓄積したデータについて、以下の種類の情報を出力します。

1. 集計サマリーファイル
2. 集計データメッセージ
3. 集計警告メッセージ

3.5.1. 集計サマリーファイル

本アプリは、「3.4 集計仕様」に記載の集計処理の過程で、各「計測フォルダ」内に集計したセンサーごとに「集計サマリーファイル」を保存します。

「計測フォルダ」内に「集計サマリーファイル」がある場合、本アプリは集計処理をスキップし「集計サマリーファイル」の内容を集計値として使用します。

- ファイル名：「モデル名_シリアル NO_summary.csv」

- ファイル形式：CSV

```
"measure","model","serial","physical","X","Y","Z","C"
"20250801_091011","A352","00000200","Acceleration",0.019273,0.002433,0.998745,0.998934
```

図 3-2 集計サマリーファイル 例

表 3-1 集計サマリーファイル 項目説明

項目名	説明
measure	計測フォルダ名（計測開始日時の時点情報）
model	センサーのモデル名
serial	センサーのシリアル NO
physical	計測した物理量（計測情報ファイルから取得）
X	X データに対する RMS 値
Y	Y データに対する RMS 値
Z	Z データに対する RMS 値
C	XYZ の三軸合成値に対する RMS 値

3.5.2. 集計データメッセージ

本アプリは、「計測フォルダ」に対する集計を行った後、集計したデータを「集計データメッセージ」として MQTT で送信します。

- トピック名：summary/\$LOG/\$MDL/\$SNO/data
 - トピック名の中の \$ 要素は、それぞれ以下の値が設定されます：
 - \$LOG：「LOGGER_ID」設定項目、\$MDL：センサーモデル名、\$SNO：センサーシリアル NO
- メッセージ形式：JSON 文字列

```
{
  "timestamp": "2025/08/10 09:10:11,100",
  "datetime": ["20250801_091011", ...],
  "physical": "Velocity",
  "trend": {
    "X": {
      "value": [9.999, ...],
      "baseline": 9.999,
      "limit_alarm": 9.999,
      "limit_trip": null,
    },
    "Y": { ... },
    "Z": { ... },
    "C": { ... }
  },
  "change": {
    "X": {
      "value": [9.999, ...],
      "limit_alarm": 9.999,
      "limit_trip": 9.999,
    },
    "Y": { ... },
    "Z": { ... },
    "C": { ... }
  },
  "alert": "Trip"
}
```

図 3-3 集計データメッセージ 例

表 3-2 集計データメッセージ 項目説明

項目名	データ型	必須	説明
"timestamp"	文字列	Y	メッセージを送信したタイムスタンプ "yyyy-mm-dd hh:mm:ss,mmm" 形式
"datetime"	文字列のリスト	Y	集計データに含まれる時点数分の文字列を含むリスト 計測フォルダ名の "yyyymmdd_hhmmss" 形式
"physical"	文字列	Y	そのセンサーが計測した物理量の種別
"trend"	辞書	Y	集計データの傾向値データを保持する辞書
"X"	辞書	Y	傾向値データのうち X 軸のデータを保持する辞書
"value"	数値のリスト	Y	X 軸の傾向値のリスト データの個数は "datetime" と一致する
"baseline"	数値	N	設定項目でこのセンサーに対する TREND_BASE が設定されている場合の値、未設定の場合は null
"limit_alarm"	数値	N	設定項目でこのセンサーに対する TREND_ALARM が設定されている場合の値、未設定の場合は null
"limit_trip"	数値	N	設定項目でこのセンサーに対する TREND_TRIP が設定されている場合の値、未設定の場合は null
"Y"	辞書	Y	傾向値データのうち Y 軸のデータを保持する辞書 (下位の要素は X と共通のため省略)
"Z"	辞書	Y	傾向値データのうち Z 軸のデータを保持する辞書 (下位の要素は X と共通のため省略)
"C"	辞書	Y	傾向値データのうち三軸合成のデータを保持する辞書 (下位の要素は X と共通のため省略)
"change"	辞書	Y	集計データの変化量データを保持する辞書
"X"	辞書	Y	変化量データのうち X 軸のデータを保持する辞書
"value"	数値のリスト	Y	X 軸の変化量のリスト データの個数は "datetime" と一致する
"limit_alarm"	数値	N	設定項目でこのセンサーに対する CHANGE_ALARM が設定されている場合の値、未設定の場合は null
"limit_trip"	数値	N	設定項目でこのセンサーに対する CHANGE_TRIP が設定されている場合の値、未設定の場合は null
"Y"	辞書	Y	変化量データのうち Y 軸のデータを保持する辞書 (下位の要素は X と共通のため省略)
"Z"	辞書	Y	変化量データのうち Z 軸のデータを保持する辞書 (下位の要素は X と共通のため省略)
"C"	辞書	Y	変化量データのうち三軸合成のデータを保持する辞書 (下位の要素は X と共通のため省略)
"alert"	文字列	Y	集計した全体の傾向値・変化量データのうち、設定された閾値を超過するものがある場合、超過した種別の最大を示す文字列。"Trip" > "Alarm" > ""。

※ 注意：「集計データメッセージ」の「alert」項目は集計データ全体に対して判定されます。警告の発生後、対処を行うことで警告状態から回復した場合でも、再度集計を行うことで過去の状態から警告と判定します。それらを警告としたいくない場合、必要に応じて警告状態以前の「計測フォルダ」、または「計測情報ファイル」と「計測データファイル」を、集計対象外のフォルダに退避してください。

3.5.3. 集計警告メッセージ

本アプリは、「集計データメッセージ」を送信した後、最新時点において閾値を超えた警告データがある場合に「集計警告メッセージ」を MQTT で送信します。警告データがない場合は「集計警告メッセージ」は送信されません。

- トピック名：summary/\$LOG/\$MDL/\$SNO/alert
- メッセージ形式：JSON 文字列

```
{
  "timestamp": "2025/08/10 09:10:11,200",
  "datetime": "20250801_101011",
  "physical": "Velocity",
  "level": "Trip",
  "trend": [
    {
      "axis": "Y",
      "type": "Trip",
      "value": 0.003843,
      "baseline": 0.001,
      "limit_alarm": 0.002,
      "limit_trip": 0.003
    }
  ],
  "change": []
}
```

図 3-4 集計警告メッセージ 例

表 3-3 集計警告メッセージ 項目説明

項目名	データ型	必須	説明
"timestamp"	文字列	Y	メッセージを送信したタイムスタンプ "yyyy-mm-dd hh:mm:ss,mmm" 形式
"datetime"	文字列	Y	警告を検出した最新時点を示す文字列 計測フォルダ名の "yyyymmdd_hhmmss" 形式
"physical"	文字列	Y	そのセンサーが計測した物理量の種別
"level"	文字列	Y	検出した警告のうち最大の種別, "Trip" > "Alarm"
"trend"	辞書のリスト	Y	傾向値データから検出した警告データを格納した辞書のリスト、検出がない場合は空のリスト
--	辞書	N	(リストから取得された辞書要素)
"axis"	文字列	Y	警告が検出された軸、"X" "Y" "Z" "C"
"type"	文字列	Y	警告の種別、"Trip" "Alarm"
"value"	数値	Y	集計値
"baseline"	数値	N	設定項目でこのセンサーに対する TREND_BASE が設定されている場合の値、未設定の場合は null
"limit_alarm"	数値	N	設定項目でこのセンサーに対する TREND_ALARM が設定されている場合の値、未設定の場合は null
"limit_trip"	数値	N	設定項目でこのセンサーに対する TREND_TRIP が設定されている場合の値、未設定の場合は null
"change"	辞書のリスト	Y	変化量データから検出した警告データを格納した辞書のリスト、検出がない場合は空のリスト (下位の要素は"trend"に準じるため省略) (ただし "baseline" 要素は含まず)

3.6. 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 3-4 ラズパイロガーと共通する設定項目

設定項目名	説明	設定可能な値 [初期設定値]	必須	補足
LOGGER_ID	どのラズパイロガーで計測したかを識別する ID	1～8 文字程度の英数字 [RP1]	Y	
OUTPUT_PATH	ラズパイロガーのデータ保存フォルダ	存在するディレクトリパス [/home/pi/measure]	Y	
MESSAGE_HOST	MQTT メッセージ送信先ホスト	ホスト名として有効な文字列 [localhost]	N	localhost=ラズパイ本体
MESSAGE_PORT	MQTT メッセージ送信先ポート番号	0 から 65535 の整数 [1883]	N	
LOG_LEVEL	ログ出力する最低レベル	ERROR, WARN, INFO, DEBUG [INFO]	N	設定可能な値以外の場合 INFO を適用

表 3-5 本アプリの設定項目

設定項目名	説明	設定可能な値 [初期設定値]	必須	補足
MESSAGE_CLID	MQTT 接続クライアント ID	任意の英数字文字列 []	N	
MAX_MEASUREMENTS	集計の対象とする計測フォルダ数	0 以上の整数、0=全フォルダ [0]	Y	最新から指定数のフォルダを絞り込む
SENSOR_MODEL	集計の対象にするセンサーモデル名	モデル名を含んだ文字列、または * [A342,A352,A370]		* はすべてにマッチ
SKIP_SECONDS	計測データの先頭から読み飛ばす秒数	0 以上の整数 [0]		
SUMMARY_SECONDS	データファイルから集計する秒数	1 以上の整数 [10]		

表 3-6 本アプリでセンサー毎に定義する設定項目

設定項目名	説明	設定可能な値	必須	補足
MODEL_SERIAL_AXIS_TREND_BASE	傾向値集計にてベースラインとして使用する値	正の数値	N	MODEL, SERIAL, AXIS をセンサーに応じて置き換えて定義する *1
MODEL_SERIAL_AXIS_TREND_ALARM	傾向値集計にてアラーム警告として使用する閾値			
MODEL_SERIAL_AXIS_TREND_TRIP	傾向値集計にてトリップ警告として使用する閾値			
MODEL_SERIAL_AXIS_CHANGE_ALARM	変化量集計にてアラーム警告として使用する閾値			
MODEL_SERIAL_AXIS_CHANGE_TRIP	変化量集計にてトリップ警告として使用する閾値			

- *1 : 設定項目のテンプレート行をコピーし、MODEL, SERIAL, AXIS を使用するセンサーのモデル名、シリアル NO、計測軸に置き換えて定義してください。
 - 例) A370_00000001_X_TREND_BASE=0.000001
 - 計測軸は X, Y, Z 以外に三軸合成値の C が使用可能です。

4. Raspberry Pi へのセットアップ

本アプリを Raspberry Pi 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	
├── LICENSE	本アプリのライセンス条項を記載したファイル
├── pyproject.toml	Python プロジェクト設定ファイル
├── src	本アプリの Python ソースコード
├── tests	本アプリのユニットテストソースコード
└── .env.default	本アプリの設定ファイルのテンプレート

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」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 MQTT ブロカー（mosquitto）をインストールする

本ガイドの以下の手順では、以下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 ZIP ファイルを転送

以下のコマンドを実行して、ダウンロードした ZIP ファイルを Raspberry Pi に転送します。

```
▸ scp MSG007-001a.zip pi@192.168.1.52:.
```

`pi` ユーザーのホームディレクトリにファイルが転送されます。

4.4. Raspberry Pi にアプリケーションをインストール

以下の手順に従い、Raspberry Pi にアプリケーションをインストールします。

1. Raspberry Pi にログインします

```
▸ ssh pi@192.168.1.52
```

2. アプリケーションをインストールするディレクトリを作成します

```
▸ sudo mkdir /app (※ /app フォルダがない場合)
▸ sudo chown pi:pi /app (※ /app フォルダがない場合)
▸ mkdir -p /app/MSG007-001a
```

3. アプリケーションの ZIP ファイルを展開します

- ▶ `cd /app/MSG007-001a`
- ▶ `unzip ~/MSG007-001a.zip`

4. アプリケーション実行用に Python の仮想環境を作成します

- ▶ `python -m venv --upgrade-deps venv`
- ▶ `source venv/bin/activate`

5. アプリケーションが使用するパッケージをインストールします

- ▶ `pip install .`

4.5. アプリケーション設定

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`

必要に応じてテキストエディタで設定を変更します。

設定項目は「3.6 設定項目」を参照ください。

5. アプリケーション実行

Raspberry Pi 上でラズパイロガーを使って計測が実行された状態から、本アプリを実行する方法を説明します。

5.1. アプリケーションの手動実行

以下の手順で本アプリをコマンドラインから手動実行します。

1. Raspberry Pi にログインします
 - ▶ `ssh pi@192.168.1.52`
2. アプリケーションをインストールしたディレクトリに移動します
 - ▶ `cd /app/MSG007-001a`
3. Python 仮想環境を有効化し、本アプリを手動実行します
 - ▶ `source venv/bin/activate`
 - ▶ `python -m raspi_summary`
4. 3 の手順の代わりに、以下でも本アプリを手動実行できます
 - ▶ `venv/bin/python -m raspi_summary`

5.2. アプリケーションのスケジュール実行

本アプリは OS の機能を使うことで定期的に行うことが可能です。

「関連文書 2」に記載のラズパイロガーの「スケジュール・サイクル実行」を応用して、ラズパイロガーと本アプリを定時実行する例を以下に記載します。

例) 毎日 8:00 に 5 分間ラズパイロガーで計測し、その後 8:10 に本アプリを実行する

1. Raspberry Pi にログインします
 - ▶ `ssh pi@192.168.1.52`
2. スケジュールを設定するためのコマンドを実行します (エディタが開きます)
 - ▶ `crontab -e`
3. 開いたファイルに次のようにスケジュール設定を記載します

```
# 8:00 から 5 分間計測する
0 8 * * * sudo systemctl start logger@300.service
# 8:10 に集計を実行する
10 8 * * * /app/MSG007-001a/venv/bin/python -m raspi_summary
```

図 5-1 スケジュール設定 例

- ▶ スケジュール設定書式の詳細については「関連文書 2」を参照ください
4. エディタを保存し終了します

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、本アプリの機能追加や改修に必要となる開発環境の構築方法を説明します。

6.1. 開発環境構築

ローカル PC に Python3.11 がインストールされ、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た前提で説明します。

6.1.1. Python 仮想環境作成

ターミナルを起動し、ZIP ファイルを展開したフォルダに Python 仮想環境を作成します。

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 --upgrade-deps venv`
 - `venv\scripts\activate.ps1`
- `py` コマンドがインストールされていない場合は、代わりに以下を実行します。
 - `python3.11 -m venv --upgrade-deps 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 --upgrade-deps venv`
 - `source venv/bin/activate`

6.1.2. 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。

OS 共通 :

- `pip install -e .`

本アプリのソースコードは「editable mode」で仮想環境にインストールします。

また、開発やテストでのみ使用するパッケージをインストールするには、以下を実行します。

Windows の場合 :

- `pip install -e .[dev,test]`

Mac や Linux の場合 :

- `pip install -e ".[dev,test]"`

6.2. プログラム構成

本アプリのプログラム構成を以下に説明します。

表 6-1 プログラム構成

プログラムファイル	説明
src/raspi_summary	本アプリのプログラムのルートパッケージ。
__main__.py	本アプリのメインモジュール。アプリの全体的な処理が記載されている。
config.py	設定ファイルから設定項目を取得する関数群。簡易なチェックを実施している。
const.py	プログラムで共通利用する定数を定義したモジュール。
domain/	本アプリが扱うデータ構造やビジネスロジックを扱うパッケージ。
logger.py	ラズパイロガーの計測フォルダに関するデータ構造を扱うモジュール。
summary.py	計測データを集計したデータや集計ロジックを扱うモジュール。
util/	本アプリで共通的に扱うユーティリティを扱うパッケージ。
env.py	環境変数を扱うモジュール。
logging.py	ログ出力を扱うモジュール。
message.py	メッセージ送信を扱うモジュール。

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.epsondevice.com/sensing/ja/contact/>