EPSON

Machine Condition Monitoring System based on ISO 20816 - User's Guide

Notice of Document

Evaluation board/kit and development tool important notice

1. This evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the requirements of design for finished products.
2. This evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by the use of it. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. The part used for this evaluation board/kit or development tool may be changed without any notice.

NOTICE : PLEASE READ CAREFULLY BELOW BEFORE THE USE OF THIS DOCUMENT

The content of this document is subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purposes, in whole or in part, without the consent of Seiko Epson Corporation("Epson").
2. Before purchasing or using Epson products, please contact with our sales representative for the latest information and be always sure to check the latest information published on Epson's official web sites and sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purpose only. Please use the application circuits, programs, usage, etc. in the design of your equipment or systems at your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties' intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. In using Epson products, you shall be responsible for safe design in your products; your hardware, software and systems are designed enough to prevent any harm or damages to life, health or property even if any malfunction or failure might be caused by Epson products. In designing of your products with using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson's web site, etc.). When using the information included in the above materials such as product data, chart, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both in stand-alone basis and within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incurred by due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) and applications individually listed in this document ("General Purpose"). Epson products are NOT intended for any use beyond the General Purpose that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to harm to life, health or serious property damage or severe impact on society, including, but not limited to listed below. Therefore, you are advised to use Epson products only for the General Purpose. Should you desire to buy and use Epson products for the particular purpose other than the General Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any particular purpose.
   [Particular purpose]
   Space equipment (artificial satellites, rockets, etc.)
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
   Medical equipment (other than applications individually listed in this document) / Relay equipment to be placed on sea floor
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment
   Other applications requiring similar levels of reliability as the above
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing military weapons of mass destruction, military purpose use, or any other military applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

2022.08

Trademark

- Raspberry Pi is a trademark of Raspberry Pi Ltd.
- Microsoft, Windows are trademarks of the Microsoft group of companies.
- EPSON is a registered trademark of Seiko Epson Corporation.
- Other product names are trademarks or registered trademarks of their respective owners.

Table of contents

Revision History

Rev. No.	Rev. Date	Page	Rev. Contents
20251130	2025/11/30	ALL	First edition Compatible with MSG007-001a_v1.0.0 release

1. Related Document

1. "Setup Manual for Vibration Measurement System Using Raspberry Pi Products " Rev.20250731
2. "Operation Manual for Vibration Measurement System Using Raspberry Pi Products" Rev.20250731
3. "Monitoring Application for Vibration Measurement System - User's Guide" Rev.20251130

2. Introduction

This User's Guide describes the following aspects of the “Machine Condition Monitoring System Based on ISO 20816”.

- Specification
- Setup Procedures
- Execution Method
- Developer Guide

"Machine Condition Monitoring System based on ISO 20816" (hereinafter referred to as "this application") is a program that runs on a Raspberry Pi where the “Vibration Measurement System Using Raspberry Pi Products” (hereinafter referred to as "RasPi Logger") is installed. The application summarizes data files measured and saved by RasPi Logger.

Each measurement session performed by RasPi Logger is treated as a single time point, and the sensor data is statistically processed. The statistical values from multiple measurement sessions are then aggregates as time-series changes for each sensor. In addition, by configuring threshold values for each sensor axis in advance, the system can output warnings when the thresholds are exceeded.

The results of the data summarization and warnings are sent as MQTT messages to MQTT broker. Messages sent to the MQTT broker can be viewed on a PC web browser using the "Vibration Measurement System Monitoring App" (hereinafter referred to as the "Monitoring App").

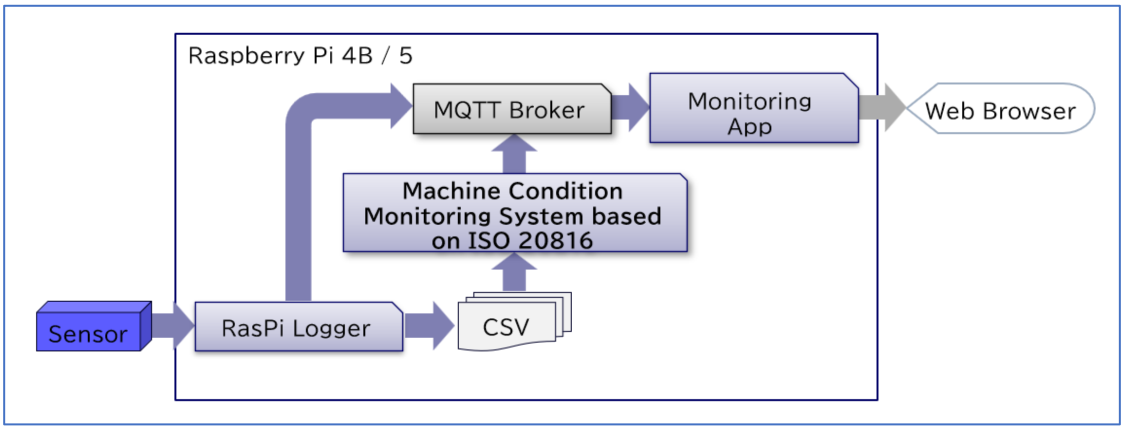


Figure 2-1 Overall Configuration Diagram

By using this application, it becomes possible to monitor the condition of rotating equipment in accordance with ISO 20816.

ISO 20816 defines methods for machine condition monitoring using vibration. Specifically, it involves periodically measuring the vibration of the target equipment and monitoring the magnitude, variation, and rate of change of the vibration. If the measured values exceed predefined thresholds, emergency shutdowns or maintenance actions are carried out on the target equipment.

This application performs condition monitoring using the following reference values and thresholds. These values must be determined in advance based on prior vibration measurements, ISO standards, and practical experience from the field.

Table 2-1 Reference Values and Thresholds for Vibration-Based Condition Monitoring

Reference value / Threshold	Description
Baseline value	A reference value that indicates the magnitude of the vibration under normal operating conditions.
Alarm value	A threshold for vibration magnitude or variation. If exceeded, the equipment should continue operating with caution while investigating the cause and implementing countermeasures.
Trip value	A threshold for vibration magnitude or variation. If exceeded, continued operation is considered hazardous and should be stopped immediately.

In this application, RMS (root mean squared) is used as the magnitude of the vibration. By customizing this application, it is also possible to monitor the condition using indicators other than RMS.

References: *v_TECH*, Condition Monitoring and Diagnosis of Machine and Equipment Based on ISO Standard (Vibration Category II : Version 5.0), September 30, 2019

3. Specification

3.1. Verified Operating Environment

- Raspberry Pi 4B
- Raspberry Pi 5

3.2. Compatible Sensors

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

3.3. Input Specifications

This application inputs the "Measurement Folder" created in the "Data Storage Folder" of the RasPi Logger, and the "Measurement Information File" and "Measurement Data File" stored in it.

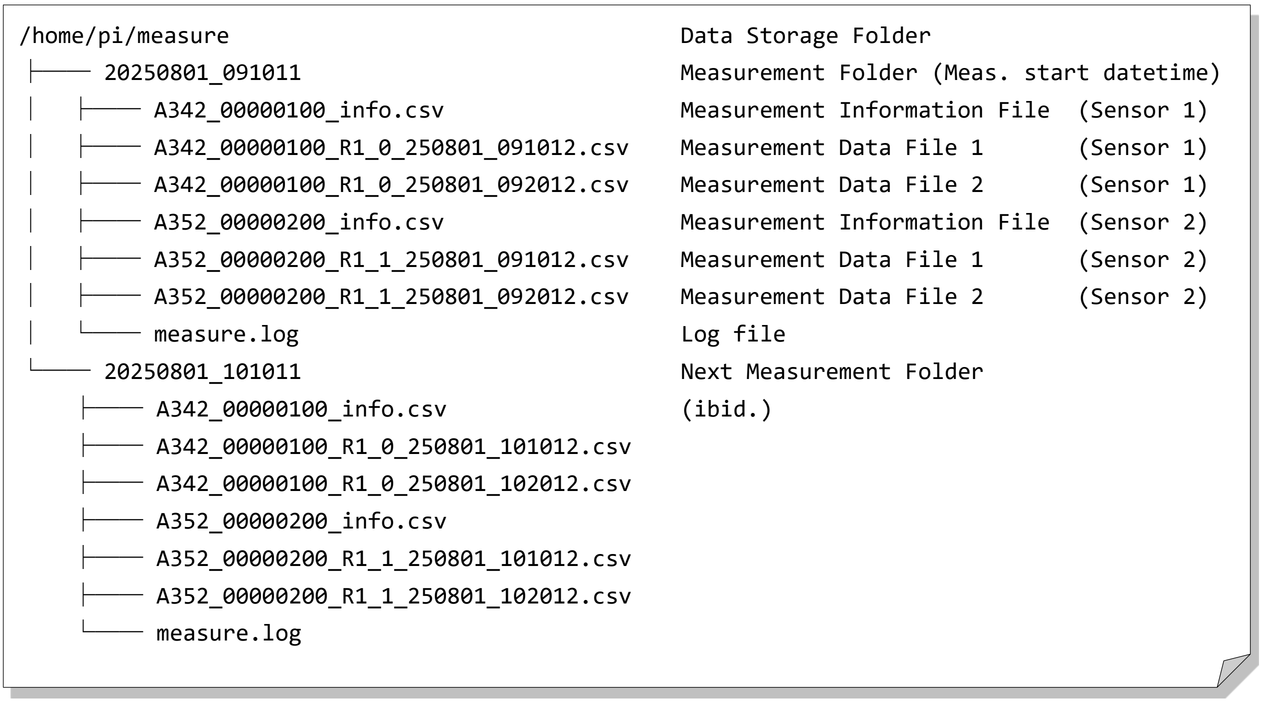
- The "Measurement Start Date and Time" indicated by the "Measurement Folder" is used as the time-in-time information for the measurement.
- Use the "Measurement Information File" in the "Measurement Folder" as a clue to identify the target sensor.
- The first “Measurement Data File” related to the “Measurement Information File” is used.

Note: If "Measurement Folder", "Measurement Information File", and "Measurement Data File" are renamed, warnings are output in the terminal and they are not used, or they are not detected as being processed.

Note: “SKIP_SECOND” and “SUMMARY_SECONDS” can be defined as the configuration item. Since this application uses only the first data file, the data in this file must be longer than the sum of ”SKIP_SECONDS” and "SUMMARY_SECONDS". If there is not enough data to the required number of seconds, a warning is output in the terminal and the data is not used.

Note: If the sensor output is not in Raw mode, a warning is output and the measurement data is not used.

For reference, the following is an example of the RasPi Logger's data storage folder and measurement data files.



```
/home/pi/measure                                 Data Storage Folder
├──── 20250801_091011                            Measurement Folder (Meas. start datetime)
│    ├──── A342_00000100_info.csv                Measurement Information File  (Sensor 1)
│    ├──── A342_00000100_R1_0_250801_091012.csv  Measurement Data File 1       (Sensor 1)
│    ├──── A342_00000100_R1_0_250801_092012.csv  Measurement Data File 2       (Sensor 1)
│    ├──── A352_00000200_info.csv                Measurement Information File  (Sensor 2)
│    ├──── A352_00000200_R1_1_250801_091012.csv  Measurement Data File 1       (Sensor 2)
│    ├──── A352_00000200_R1_1_250801_092012.csv  Measurement Data File 2       (Sensor 2)
│    └──── measure.log                           Log file
└──── 20250801_101011                            Next Measurement Folder
     ├──── A342_00000100_info.csv                (ibid.)
     ├──── 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
```

Figure 3-1 RasPi Logger Data Storage Folder Example

3.4. Summary Specifications

This application uses "Data Storage Folder" of the RasPi Logger as described in 3.3 Input Specifications. Details of data search and summarizing methods are shown below.

- Search for "Measurement Folder" from "Data Storage Folder" in ascending order of folder name.
  - If a positive number is specified in the configuration item "MAX_MEASUREMENTS", the number of items is narrowed down from the latest measurement as the summary target.
- The "Measurement start date and time" is obtained from the name of found "Measurement Folder" and used as the information at the time of summarization.
- Search for the "Measurement Information File" in the "Measurement Folder" and select the information of sensor used for measurement.
  - If the sensor does not correspond to the configuration item "SENSOR_MODEL", the file is skipped.
- Search for the first "Measurement Data File" related to the "Measurement Information File" and process it as input data.
- After processing all the "Measurement Information Files", move on to the next "Measurement Folder".

Note: If there are no "Measurement Folders" in "Data Storage Folder", nothing is summarized.

Based on the above, this application performs the following summary operations on the files found through the search.

- Retrieve "SPS" from the "Measurement Information File" and calculate the required number of data points for the configuration parameters "SKIP_SECONDS" and "SUMMARY_SECONDS".
- Skip the number of data points corresponding to "SKIP_SECONDS" from the beginning of the "Measurement Data File", then extract the number of data points corresponding to "SUMMARY_SECONDS". For these data points, calculate the RMS (root mean squared) for each axis (X, Y, and Z) and the triaxial composite value (C) calculated from XYZ.

- The data summarized in this method is stored in time series as summary values for each sensor at each point in time.

Note: If the value of the physical quantity obtained from the "Measurement Information File" for any individual sensor changes, a warning will be output to the terminal, and the corresponding measurement data will not be used in the summary.

After searching all "Measurement Folders", this application traces the summarized data in chronological order and calculates the amount of change.

- For each axis of each sensor, the absolute difference between the previous value and the current value is stored as the amount of change (for the first entry where no previous value exist, the change is set to zero).

3.5. Output Specifications

This application outputs the following types of information about the summarized and accumulated data.

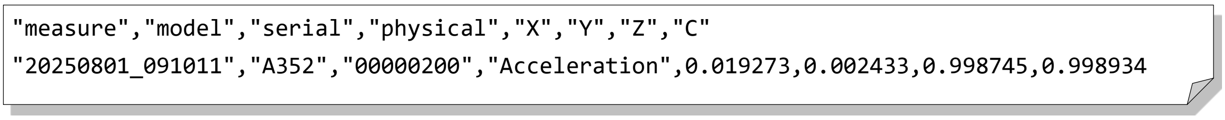
1. Summary File
2. Summary Data Messages
3. Summary Warning Messages

3.5.1. Summary File

During the summary process described in section 3.4 Summary Specifications, this application saves a Summary File for each sensor within each "Measurement Folder".

If there is a "Summary File" in the "Measurement Folder", this application will skips the summary process and use the contents of the "Summary File" as the summary value.

- File Name: "Model Name_Serial No_summary.csv
- File format: CSV



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

Figure3-2 Summary File Example

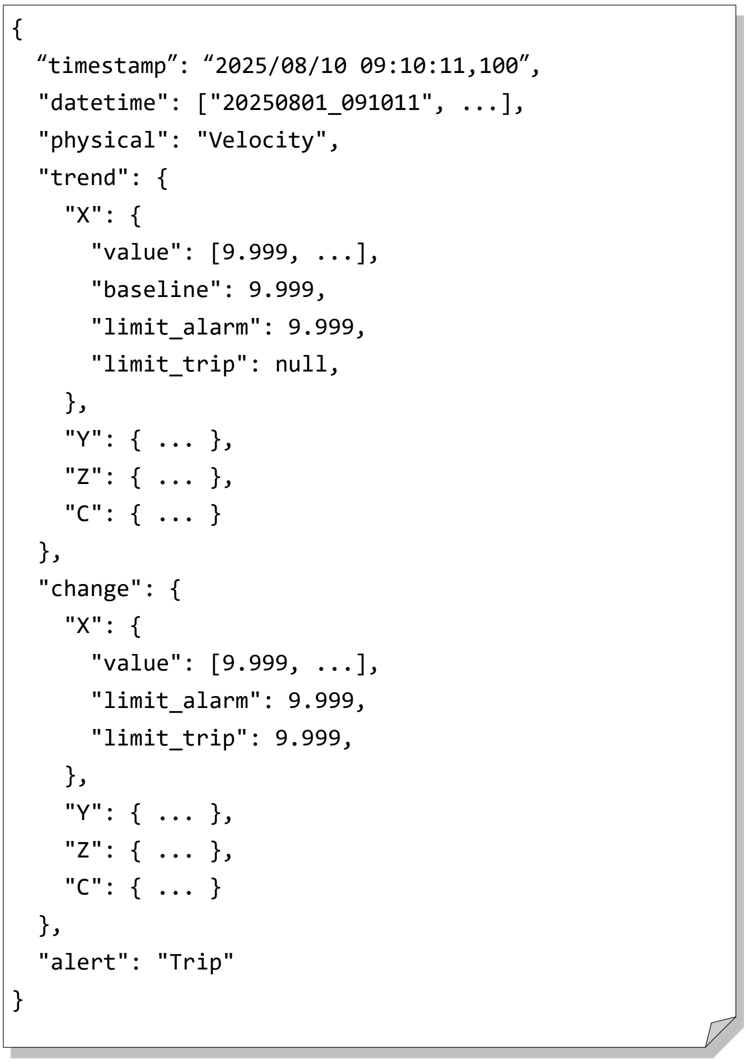
Table 3-1 Summary File Item Description

Item	Description
measure	Measurement folder name (current information of measurement start date and time)
model	Sensor model name
serial	Sensor serial No
physical	Measured physical quantities (obtained from measurement information files)
X	RMS value for X data
Y	RMS value for Y data
Z	RMS value for Z data
C	RMS values for triaxial synthesis values of XYZ

3.5.2. Summary Data Messages

After performing the summary process for all "Measurement Folder", this application sends the summarized data as an "Summary Data Message" via MQTT.

- Topic Name: `summary/$LOG/$MDL/$SNO/data`
  - Each $ element in the topic name is replaced with the following values:
    `$LOG`: "LOGGER_ID" configuration item, `$MDL`: Sensor model name, `$SNO`: Sensor serial number
- Message Format: JSON String



```
{
  “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"
}
```

Figure 3-3 Summary Data Message Example

Table 3-2 Summary Data Message Item Descriptions

Field Name	Data type	Required	Description
“timestamp”	`string`	Y	Timestamp when the message was sent. Format: "`yyyy-mm-dd hh:mm:ss,mmm`"
“datetime”	`List of strings`	Y	List containing strings for the number of points in the summary data. Format: "`yyyymmdd_hhmmss`" (same as the measurement folder name)

“physical”	`string`	Y	The type of physical quantity measured by the sensor.
“trend”	`dictionary`	Y	Dictionary that holds trend data from the summary.
“X”	`dictionary`	Y	Dictionary that holds the trend data for the X-axis.
“value”	`List of numbers`	Y	List of trend values for the X-axis. The number of entries matches "datetime" list.
“baseline”	`numeric value`	N	Value of TREND_BASE for this sensor if configured; otherwise `null`.
“limit_alarm”	`numeric value`	N	Value of TREND_ALARM for this sensor if configured; otherwise `null`.
“limit_trip”	`numeric value`	N	Value of TREND_TRIP for this sensor if configured; otherwise `null`.
“Y”	`dictionary`	Y	Dictionary that holds the trend data for the Y-axis (same structure as X).
“Z”	`dictionary`	Y	Dictionary that holds the trend data for the Z axis (same structure as X).
“C”	`dictionary`	Y	Dictionary that holds the trend data for the synthesized three-axis value (same structure as X).
“change”	`dictionary`	Y	Dictionary that holds change data derived variation data
“X”	`dictionary`	Y	Dictionary that holds change for the X-axis of variable data.
“value”	`List of numbers`	Y	List of change values for the X-axis. The number of data matches "datetime"
“limit_alarm”	`numeric value`	N	Value of CHANGE_ALARM for this sensor if configured; otherwise `null`.
“limit_trip”	`numeric value`	N	Value of CHANGE_TRIP for this sensor if configured; otherwise `null`.
“Y”	`dictionary`	Y	Dictionary that holds change data for the Y-axis (same structure as X).
“Z”	`dictionary`	Y	Dictionary that holds change data for the Z axis (same structure as X).
“C”	`dictionary`	Y	Dictionary that holds change data for the synthesized three-axis value (same structure as X).
“alert”	`string`	Y	If any trend or change value exceeds the configured thresholds, this field indicates the highest severity among the exceeded types: `"Trip" > "Alarm" > ""` (empty string if no threshold is exceeded).

Note: The alert field in the “Summary Data Messages” is evaluated based on the entire set of summarized data. Even if a warning condition has been resolved through corrective actions, re-running the summary process may still result in a warning being triggered based on past data.
If you wish to exclude such data from triggering warnings, please move the corresponding

“Measurement Folders”, or the “Measurement Info File” and “Measurement Data File”, to a folder that is not included in the summary process.

3.5.3. Summary Warning Messages

After sending a "Summary Data Message", this application sends a "Summary Warning Message" via MQTT only if there is warning data that exceeds the threshold at the latest time point. If there is no warning data, no "Summary Warning Message" is sent.

- Topic Name: `summary/$LOG/$MDL/$SNO/alert`
- Message Format: JSON String

```
{
  “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": []
}
```

Figure 3-4 Example of a Summary Warning Message

Table 3-3 Summary Warning Messages Field Description

Field Name	Data type	Required	Description
“timestamp”	`string`	Y	Timestamp when the message was sent. Format:`"yyyy-mm-dd hh:mm:ss,mmm"`
“datetime”	`string`	Y	Timestamp of the latest data point where the warning was detected. Format:`"yyyymmdd_hhmmss"` (same as measurement folder names)
“physical”	`string`	Y	Type of physical quantity measured by the sensor
“level”	`string`	Y	Highest severity level among detected warnings: `"Trip" > "Alarm"`
“trend”	`List of dictionaries`	Y	List of dictionaries that store warning data detected from trend data, or an empty list if there are no detections
--	`dictionary`	N	(Dictionary elements retrieved from the list)
“axis”	`string`	Y	Axis on which the warning was detected, `"X"｜”Y”｜”Z”｜”C”`
“type”	`string`	Y	Warning Type, `"Trip"｜”Alarm”`
“value”	`numeric value`	Y	Summary value that triggered the warning
“baseline”	`numeric value`	N	Value of TREND_BASE for this sensor if configured; otherwise `null`
“limit_alarm”	`numeric value`	N	Value of TREND_ALARM for this sensor if configured; otherwise, `null`
“limit_trip”	`numeric value`	N	Value of TREND_TRIP for this sensor if configured; otherwise `null`
“change”	`List of dictionaries`	Y	List of dictionaries that store warning data detected from change values. If no warnings are detected, this list is empty. (Structure follows "trend" but does not include the "baseline" field)

3.6. Configuration Items

This application reads its configuration from the "`.env`" file located in the setup directory.

The following items can be set in the configuration file:

Table 3-4 Common Configuration Parameters (Shared with RasPi Logger)

Parameter Name	Description	Allowed Values [Default]	Required	Notes
LOGGER_ID	Identifier for the RasPi Logger used for measurement	Alphanumeric characters of about 1~8 characters [RP1]	Y	
OUTPUT_PATH	Directory path where the RasPi logger saves measurement data	Valid existing directory path [/home/pi/measure]	Y	

MESSAGE_HOST	Host for MQTT message transmission	Valid hostname [localhost]	N	`localhost` refers to the Raspberry Pi itself
MESSAGE_PORT	Port number for MQTT message transmission	Integers from 0 to 65535 [1883]	N	
LOG_LEVEL	Minimum log level to output	ERROR, WARN, INFO, DEBUG [INFO]	N	Apply INFO if not configurable

Table 3-5 Application Specific Configuration Parameters

Parameter Name	Description	Allowed Values [Default]	Required	Note
MESSAGE_CLID	MQTT client ID used for connection	Any alphanumeric string []	N	
MAX_MEASUREMENTS	Number of Metering Folders to Target	Integer greater than 0, 0 = all folders [0]	Y	Filters folders starting from the most recent
SENSOR_MODEL	Sensor model name to be aggregated	A string containing the model's name, or * [A342,A352,A370]		* matches all models
SKIP_SECONDS	Number of seconds to skip from the beginning of the measurement data	Integer greater than 0 [0]		
SUMMARY_SECONDS	Number of seconds to include in the summary	Integer numbers greater than 1 [10]		

Table 3-6 Sensor-Specific Configuration Parameters in this Application

Parameter Name	Description	Allowed Values	Required	Note
*MODEL_SERIAL_AXIS***_TREND_BASE**	Baseline value used in trend summary	Positive number	N	Replace *MODEL, SERIAL, AXIS* with actual sensor info *1
*MODEL_SERIAL_AXIS***_TREND_ALARM**	Threshold for alarm warning in trend summary			
*MODEL_SERIAL_AXIS***_TREND_TRIP**	Threshold for trip warning in trend summary			
*MODEL_SERIAL_AXIS***_CHANGE_ALARM**	Threshold for alarm warning in change summary			

*MODEL_SERIAL_AXIS***_CHANGE_TRIP**	Threshold for trip warning in change summary			

- *1: Copy the template line and replace MODEL, SERIAL, and AXIS with the actual sensor model name, serial number, and measurement axis.
  - Example: A370_00000001_X_TREND_BASE=0.000001
  - In addition to X, Y, and Z, the measurement axes can be used with the synthesized three-axis value C.

4. Setup on a Raspberry Pi

This section describes the steps to set up the application on a Raspberry Pi.

4.1. File and Folder Structure

After extracting the downloaded ZIP file, the file and folder structure is as follows.

```
Expand Folder
├─── LICENSE              Files containing the license terms of the app
├─── pyproject.toml       Python project configuration file
├─── src                  Python source code of this app
├─── tests                Unit test source code for this app
└─── .env.default         Template of the configuration file of this
```

Figure 4-1 File and Folder Structure

4.2. Preparation

Before starting the setup this application, please refer to Related Documentation 1 and complete the following preparations.

- Connect the Raspberry Pi and PC to the same network
- Ensure the Raspberry Pi has internet access
- Install the MQTT broker (mosquitto) on the Raspberry Pi

The following steps in this guide assume:

- Username created on Raspberry Pi: `pi`
- Static IP address assigned to Raspberry Pi: `192.168.1.52`

If you have configured a different username or IP address, adjust the commands accordingly.

4.3. Transfer ZIP Files to Raspberry Pi

Run the following command to transfer the downloaded ZIP file to the Raspberry Pi.

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

The file will be transferred to the `pi` user's home directory.

4.4. Install the Application on the Raspberry Pi

Follow the steps below to install the application on your Raspberry Pi.

1. Log in to the Raspberry Pi

   - ‣ `ssh pi@192.168.1.52`

2. Create the directory to install the application

   - ‣ `sudo mkdir /app` (* If the /app folder does not exist)
   - ‣ `sudo chown pi:pi /app` (* If the /app folder does not exist)
   - ‣ `mkdir -p /app/MSG007-001a`

3. Extract the ZIP file of the application

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

4. Create a Python virtual environment for running the application

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

5. Install the required packages for this application

   - `pip install .`

4.5. Application Configuration

Use the following command to create the configuration file by copying the template `.env.default`:

- `cp .env.default .env`

Edit the `.env` file using a text editor as needed.

For details on configurable parameters, refer to Section 3.6 Configuration Items .

5. Running the Application

This section describes how to run this application after measurements have been performed using the RasPi Logger.

5.1. Manual Execution of the Application

Follow these steps below to manually run the application from the command line:

1. Log in to the Raspberry Pi
    - `ssh pi@192.168.1.52`
2. Move to the application installation directory
    - `cd /app/MSG007-001a`
3. Activate the Python virtual environment and run the application manually
    - `source venv/bin/activate`
    - `python -m raspi_summary`
4. Instead of step 3, you can also run the application using the following command
    - `venv/bin/python -m raspi_summary`

5.2. Scheduled Execution of Applications

This application can be executed periodically using the operating system’s scheduling features.

The following example shows how to schedule both the RasPi Logger and this application by adapting the "Scheduled Cycle Execution" method described in Related Document 2.

Example: Measure data using RasPi Logger for 5 minutes at 8:00 every day and then run this application at 8:10.

1. Log in to your Raspberry Pi
    - `ssh pi@192.168.1.52`
2. Open the scheduler configuration editor
    - `crontab -e`
3. Add the schedule settings to the opened file as follows:

```
# Measure for 5 minutes from 8:00
0  8 * * *  sudo systemctl start logger@300.service
# Run the summary at 8:10
10 8 * * *  /app/MSG007-001a/venv/bin/python -m raspi_summary
```

Figure 5-1 Schedule Setting Example

- Refer to Related document 2 for details on the cron schedule format.

4. Save and exit the editor

6. Appendix: Developer's Guide

This chapter provides guidance for developers on how to set up the development environment required to add new features or make modifications to the application.

6.1. Setting Up the Development Environment

This section assumes that Python 3.11 is installed on the local PC and that the downloaded ZIP file has been extracted to a folder of your choice.

6.1.1. Creating a Python Virtual Environment

Launch the terminal and create a Python virtual environment in the folder where you extracted the ZIP file.

On Windows:

- Launch PowerShell and run the following commands:
  - `py -3.11 -m venv --upgrade-deps venv`
  - `venv\scripts\activate.ps1`
- If `py` command is not installed, run the following commands instead:
  - `python3.11 -m venv --upgrade-deps venv`
  - `venv\scripts\activate.ps1`

On Mac or Linux:

- Open a terminal and run the following commands:
  - `python3.11 -m venv --upgrade-deps venv`
  - `source venv/bin/activate`

6.1.2. Installing Packages

Install the package required by the application.

For all operating systems:

- `pip install -e .`

The application’s source code is installed into the virtual environment in "editable mode".

To install packages used only for development and testing, run the following:

On Windows:

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

On Mac and Linux:

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

6.2. Program Structure

The program configuration of this application is described below.

Table 6-1 Program Structure

Program File/Directory	Description
`src/raspi_summary`	Root package of the program in this application.
`__main__.py`	Main module of this application. Contains the overall execution logic.
`config.py`	Functions for retrieving configuration parameters from the `.env` file. Includes basic validation.
`const.py`	Module defining constants used throughout the application.
`domain/`	Package for handling data structures and business logic used by this application.
`logger.py`	Module for handling data structures related to measurement folders from the RasPi Logger.
`summary.py`	Module for handling summarized data and summary logic.
`util/`	Package for common utility functions used across the application.
`env.py`	Module for handling environment variables.
`logging.py`	Module for managing log output.
`message.py`	Module for handling message transmission.

7. Contact Information

Seiko Epson Corporation

Sales Division MD Sales Department

Contact Information via the Internet

https://www.epsondevice.com/sensing/en/contact/