



Operation Manual for Vibration Measurement System Using Raspberry Pi Products

Notice of the 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

©Seiko Epson Corporation 2023, All rights reserved.

Trademark

Raspberry Pi is a trademark of Raspberry Pi Foundation.

Microsoft and 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 the respective companies.

Table of contents

Notice of the Document	2
Trademark	3
Revision History	6
1. Related Documents	7
2. Introduction	8
3. Specification	9
3.1. Compatible Sensor	9
3.2. Input Specification	9
3.2.1. M-A342VD/M-A542VR	9
3.2.2. M-A352AD/M-A552AR	9
3.2.3. M-A370AD	9
3.3. Output Specification	10
3.3.1. Output Folder	10
3.3.2. Measurement Data File	10
3.3.3. Measurement Information File	11
3.3.4. Status Messages	14
3.3.5. Log File	15
3.4. Configuration Items	15
3.5. Measurement Specifications	16
3.5.1. Measurement Duration	16
3.5.2. Measurement Transient Response Time	16
3.5.3. Abnormality detection function for sensor data	17
4. Conducting Measurements	18
4.1. Preparation	18
4.1.1. Operation from PC	18
4.1.2. Checking the System Time	18
4.2. Measurement Execution	18
4.2.1. Startup Execution	18
4.2.2. Scheduled Execution	19
4.2.3. Manual Execution	19
4.2.4. Notes on Measurement Execution	19
4.3. Measurement Confirmation	20
4.3.1. Checking Measurement Status	20
4.3.2. Checking Logs	20
4.3.3. Stopping the Measurement	20
4.4. Acquiring Measurement Data	20
5. Appendix tool : Sensor Self Test	21
5.1. General	21
5.2. Input Specification	21
5.3. Output Specification	21
5.4. Execution Specification	22
6. Appendix tool : Hardware Status Monitor	23
6.1. General	23
6.2. Output Specification	23

6.2.1.	Hardware Status Monitor – Measurement Data File.....	23
6.2.2.	Hardware Status Monitor – Status Messages	23
6.3.	Execution Specification	24
6.3.1.	Terminal Execution	24
6.3.2.	OS Service Registration	24
7.	Appendix : Log message.....	25
8.	Contact Information.....	29

Revision History

Rev. No.	Rev. Date	Page	Rev. Contents
20240315	2024/3/15	ALL	First Edition
20240927	2024/9/27	---	Revisions corresponding to the release of MSG002-001a_v1.10
		ALL	Minor corrections to the description, addition of notes, and addition and update of illustrations
		11, 3.415, 17	Addition of sensor data anomaly detection function, and associated corrections
		18	Addition of a procedure to check the time before starting measurement
		20	Addition of a procedure to obtain measurement data
		21	Addition of a sensor self-test tool
20250331	2025/3/31	---	Addition of a hardware status monitor tool
		14, 23	- Revised to support MSG002-001a_v1.2.0 release - Added message specifications for new MQTT message transmission feature
		11, 15	Added configuration items and measurement info file items for same reason
		21	Revised notes for sensor self-test tool
		25	Added log messages due to new features
20250731	2025/7/31	---	Revised to support MSG002-001a_v1.3.0 release
		7, 9, 15, 16, 21	Added support for M-A370AD
		8	Added system overview diagram
		10	Added handling when folder with same name already exists during output folder creation
		15	- Moved "Configuration Items" section to be part of specifications - Added default values for configuration items
		16, 18, 20	- Revised measurement specifications, separated explanations for "Measurement Time" and "Measurement Execution" - Updated execution method names in "Measurement Execution" section - Removed "manual" from startup arguments - Reorganized post-execution operations under "Measurement Confirmation"
		18, 20	Simplified manual by assuming standard username and IP address specified by user

1. Related Documents

- “Setup Manual for Vibration Measurement System Using Raspberry Pi products” Rev.20250731
- “Monitoring Application for Vibration Measurement System – User’s Guide” Rev.20250731
- M-A370AD Datasheet
- M-A352AD Datasheet
- M-A342VD Datasheet
- M-A552AR Datasheet
- M-A542VR Datasheet

2. Introduction

This manual explains the following components of the measurement system (hereafter referred to as "the system") built using the procedures described in the *Setup Manual for Vibration Measurement System Using Raspberry Pi Products*:

- Specifications
- Measurement Execution
- Included Tool: Sensor Self-Test
- Included Tool: Hardware Status Monitor

In this manual, the term "logger" refers to a configuration in which necessary equipment such as sensors is connected to a Raspberry Pi to measure vibrations.

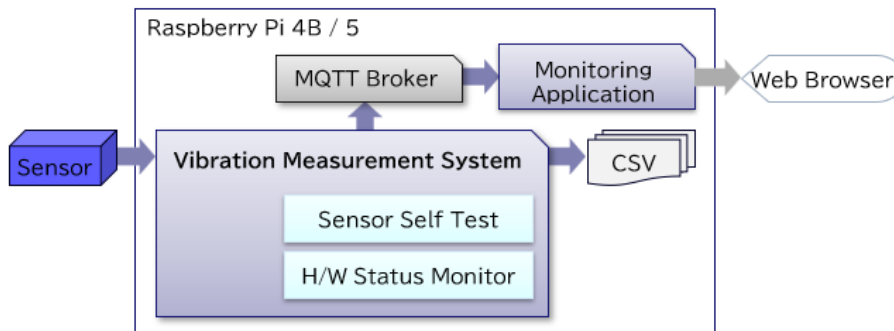


Figure 2-1 Overall System Configuration

3. Specification

3.1. Compatible Sensor

The following sensors can be measured using the logger.

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

3.2. Input Specification

This logger is possible to connect multiple sensors via USB and perform measurements. The sensor output mode uses UART automatic sampling mode.

Below are the input specifications from the sensor for each sensor model.

3.2.1. M-A342VD/M-A542VR

Table 3-1

Communication Speed	912.6kbps or 460.8kbps
Measurement Data	Velocity or Displacement
Output Data	RAW Data, RMS, Peak to Peak
Sampling Rate	Velocity: 3000sps, Displacement: 300sps

3.2.2. M-A352AD/M-A552AR

Table 3-2

Communication Speed	460.8kbps or 230.4kbps
Measurement Data	Acceleration
Output Data	RAW Data
Sampling Rate	1000sps, 500sps, 200sps, 100sps, 50sps
Filter Setting	1000sps: FIR Kaiser TAP512 fc=460 500sps: FIR Kaiser TAP512 fc=210 200sps: FIR Kaiser TAP512 fc=60 100sps: FIR Kaiser TAP512 fc=16 50sps: FIR Kaiser TAP512 fc=9

3.2.3. M-A370AD

Table 3-3

Communication Speed	460.8kbps or 230.4kbps
Measurement Data	Acceleration
Output Data	RAW Data
Sampling Rate	1000sps, 500sps, 200sps, 100sps, 50sps
Filter Setting	1000sps: FIR Kaiser TAP512 fc=210 500sps: FIR Kaiser TAP512 fc=210 200sps: FIR Kaiser TAP512 fc=60

100sps: FIR Kaiser TAP512 fc=16
50sps: FIR Kaiser TAP512 fc=9

3.3. Output Specification

3.3.1. Output Folder

A "Measurement start date and time" folder will be created in the folder specified as the data save folder in the settings file, and the following three types of files will be output.

- Measurement data file
- Measurement information file
- Log file

/home/pi/measure	Data storage folder
└ 20240909_081406	"Measurement start data and time" folder
└ A342_00000100_R4_0_240909_081407.csv	Measurement data file (Sensor1)
└ A342_00000100_info.csv	Measurement information file (Sensor2)
└ A342_00000200_R4_1_240909_081407.csv	Measurement data file (Sensor2)
└ A342_00000200_info.csv	Measurement information file (Sensor2)
└ measure.log	Log file

Figure 3–1 Example of Output to Folder

Note: If a folder with the same name already exists (e.g., due to inaccurate RTC), a new folder with a suffix (e.g., "-1") is created to avoid overwriting.

3.3.2. Measurement Data File

- The file name format is:
[SensorName_SerialNumber_LoggerID_ConnectionSerialNumber_YYMMDD_hhmmss.csv](#)
- "LoggerID" is specified in the setting file.
- The measurement data is output in the following format in csv format.
- There is no header, and the delimiter for the csv file is a comma (,), and the decimal point is a period (.).
- Depending on the settings in "3.4. Setting Items", a new measurement data file is created at intervals of 1 to 60 minutes.
- The data received from the sensor is recorded as is, except for the index. The index is added by this logger.

```
index,count,temperature,x output,youtput,z output,flag (Information for description)
0,1,33.045598400,-0.000008106,0.000001669,-0.000015497,0b00000000
1,2,33.045598400,-0.000009298,-0.000016928,-0.000018597,0b00000000
```

Figure 3–2 Example of Measurement Data File Output

Table 3-4 Measurement Data File Item Description

Item	Description
index	Record number from the start of measurement. The number is always incremented from 0 to 1. If measurement data is saved in multiple files, the index of the last record in the previous file and the index of the first record in the next file will be consecutive numbers.
count	Sensor output COUNT value. Data loss is detected by looking at this value.
temperature	TEMP value of the sensor output.
X output	X-axis measurement value of the sensor output. The physical quantity changes depending on the setting.

Y output	Y-axis measurement value of the sensor output. The physical quantity changes depending on the setting.
Z output	Z-axis measurement value of the sensor output. The physical quantity changes depending on the setting.
flag	FLAG (ND/EA) value of the sensor output. See Table 3-5 Explanation of Digits in the “Flag” Item for the meaning of each digit.

Table 3-5 Explanation of Digits in the “Flag” Item

Number of digits	M-A342VD	M-A352AD
8	X_EXI_ERR Displayed when the X-axis measurement value becomes abnormal due to structural resonance within the sensor. 1: Measurements are abnormal 0: Measurements are within normal range	Unused
7	Y_EXI_ERR Displayed when the Y-axis measurement value becomes abnormal due to structural resonance within the sensor. 1: Measurements are abnormal 0: Measurements are within normal range	Unused
6	Z_EXI_ERR Displayed when the Z-axis measurement value becomes abnormal due to structural resonance within the sensor. 1: Measurements are abnormal 0: Measurements are within normal range	Unused
5	0 (*1)	0 (*1)
4	0 (*1)	0 (*1)
3	0 (*1)	0 (*1)
2	Unused	0 (*2)
1	0 (*3)	0 (*3)

- *1: The ALARM_ERR bit. This logger does not allow you to set alarm values, so this is always 0.
- *2: The ALIASI_ERR bit. This logger automatically sets the output rate and filter cutoff frequency to normal, so this is always 0.
- *3: The EA bit. This logger does not allow you to perform self-diagnosis, so this is always 0.

3.3.3. Measurement Information File

- The filename of the measurement information file is "[SensorName_SerialNumber_info.csv](#)".
- This file is output in CSV format with the following structure.
- The measurement information file is generated at the start of the measurement, and the measurement time is updated at the end of the measurement.

```

OS,Linux-6.6.51+rpt-rpi-2712-aarch64-with-glibc2.36
LOGGER_VERSION,1.2.0
LOGGER_ID,R5
START_TYPE>manual
TIME_ZONE,UTC+0900
DATE,2024/12/13 08:14:06
MEASURE_TIME_SEC,60
INITIAL_WAIT,1
FILE_ROTATE_MIN,10
BAUD_RATE,460800
SENSOR_DATA_DIAG,True
SENSOR_DATA_DIAG_SEC,1.000000
MESSAGE_SEND,True
MESSAGE_HOST,localhost
MESSAGE_PORT,1883
PORT,/dev/ttyUSB0
SENSOR,A342
PRODUCT_ID,A342VD10
SERIAL_NO,00000100
FIRM_VER,0x0280
PHYSICAL,Velocity
OUTPUT_TYPE,Raw
SPS,3000
FILTER,
RMS_PP_INTERVAL,

```

Figure 3-3 Example Output of the Measurement Information File

Table 3-6 Description of Items in the Measurement Information File

Item	Description	Supplementary Information
OS	Operating system running on the logger	
LOGGER_VERSION	Logger version	Fixed value in the program
LOGGER_ID	String identifying the Raspberry Pi specified in the configuration file	
START_TYPE	Method used to specify measurement duration	auto: specified in config file manual: specified via startup argument
TIME_ZONE	Time zone set on the Raspberry Pi performing the measurement	
DATE	Measurement start date and time	Local time, no time zone information
MEASURE_TIME_SEC	Measurement duration (seconds)	If the measurement is stopped midway, the actual measurement time is recorded

INITIAL_WAIT	Initial waiting time (seconds)	Always 0 for scheduled measurements
FILE_ROTATE_MIN	Measurement file rotation interval (minutes)	
BAUD_RATE	Communication speed with the sensor	
SENSOR_DATA_DIAG	Status of the sensor data anomaly detection function	True: enabled, False: disabled
SENSOR_DATA_DIAG_SEC	Duration (seconds) used for anomaly detection	
MESSAGE_SEND	Status of message sending feature	True: enabled, False: disabled
MESSAGE_HOST	Hostname for message sending	
MESSAGE_PORT	Port number for message sending	
PORT	Port to which the sensor is connected on the Raspberry Pi	
SENSOR	Sensor used for measurement	
PRODUCT_ID	Product ID of the sensor used for measurement	
SERIAL_NO	Serial number of the sensor used for measurement	
FIRM_VER	Firmware version of the sensor used for measurement	
PHYSICAL	Physical quantity measured	
OUTPUT_TYPE	Output format of the measurement	
SPS	Sampling rate during measurement	
FILTER	Filter used during measurement	Not applicable for M-A342VD
RMS_PP_INTERVAL	RMS and P-P measurement interval for M-A342VD	Only applicable for M-A342VD Output even when Raw data is selected, but not used in measurement

3.3.4. Status Messages

This system sends messages regarding measurement status, sensor conditions, and error occurrences using the MQTT protocol.

These messages can be viewed via a web browser using Seiko Epson's "Vibration Measurement System Monitoring App."

For details, refer to the Monitoring App User Guide.

The specifications of the system's status messages are shown below:

Table 3-7 Status Message Specifications

Message Type	Topic Name	Message Content
Logger Measurement Status	logger/\$LOG	status: "START" "STOP"
Logger Error Status	logger/\$LOG/error	level: "CRITICAL" "ERROR" "WARNING", message: "Actual error message"
Sensor Measurement Status	logger/\$LOG/sensor/\$MDL/\$SNO	status: "OK" "NG"
Sensor Error Status	logger/\$LOG/sensor/\$MDL/\$SNO/error	level: "CRITICAL" "ERROR" "WARNING", message: "Actual error message"
Sensor Data Loss Status	logger/\$LOG/sensor/\$MDL/\$SNO/lost	message: "Message when data loss occurs"
Sensor Abnormal Detection	logger/\$LOG/sensor/\$MDL/\$SNO/abnormal	message: "Message when abnormality is detected"

- In the topic name:
 - \$LOG is the value of LoggerID from the config file
 - \$MDL is the sensor model name
 - \$SNO is the sensor serial number
- All messages are formatted as JSON strings.
- Each message includes a timestamp field:
timestamp: "yyyy/mm/dd hh:mm:ss.nnnnnn"
- Example of a Logger Measurement Status Messages:

```
{"timestamp": "2025/01/01 00:00:00.000000", "status": "START"}
```

Figure 3-4 Example of a Logger Measurement Status Messages

- For status messages from the Hardware Status Monitor Tool, refer to section [6.2.2 Hardware Status Monitor – Status Messages].

3.3.5. Log File

- The log file is named:
measure.log
- It is output in plain text format.
- During measurement, the log is temporarily saved as
_logger_tmp_<MeasurementStartTime>.log in the folder "/app/MSG002-001a/".
- Upon completion of the measurement, it is saved as a "measure.log" file within the "[measurement start date time](#)" folder in the data storage directory.
- For details on log message types and meanings, refer to "7. Appendix: Log Messages".

```
LOG_LEVEL |Event Occurrence Time(YYYY-MM-DD hh:mm:ss,xxx) MODULE_NAME: MESSAGE (Information for
description)
INFO |2024-09-09 08:13:52,366 controller: Start initialization
INFO |2024-09-09 08:13:52,477 controller: Found product: A342VD10 port=/dev/ttyUSB0 baud=460800
INFO |2024-09-09 08:13:53,863 controller: Found product: A342VD10 port=/dev/ttyUSB1 baud=460800
INFO |2024-09-09 08:14:05,279 controller: Initial wait for 1 sec
INFO |2024-09-09 08:14:06,286 controller: Start measurement
INFO |2024-09-09 08:14:07,182 A342.00000100.writer: Initial output to
/home/pi/measure/20240909_081406/A342_00000100_R4_0_240909_081407.csv
INFO |2024-09-09 08:14:07,396 A342.00000200.writer: Initial output to
```

Figure 3-5 Example of Log Data

3.4. Configuration Items

In this system, variout operations including input/output can be specified by describing the settings in a text-based configuration file.

- The configuration file name is: /app/MSG002-001a/.env

Settings are written in the format: KEY=VALUE

Regardless of the sensor model used for measurement, all configuration items (except LOG_LEVEL) must be assigned valid values.

The program checks the configuration at startup, and if any value is invalid, it will terminate with an error.

Note: Do not insert spaces around the = sign, and do not change the setting key names. Doing so will cause errors.

Table 3-8 List of Configurable Items

Key (Setting Name)	Description	Allowed Values [Default]	Notes
LOGGER_ID	Identifier for the Raspberry Pi used for measurement	Alphanumeric string, 1-63 characters [RP1]	
OUTPUT_PATH	Data storage folder	Valid directory path [/home/pi/measure]	
INITIAL_WAIT	Initial wait time in seconds	Integer from 0 to 3600 [30]	
MEASURE_TIME_SEC	Measurement duration (in seconds) for auto/manual start	Integer from 0 to $2^{63}-1$ [600]	0 = endless measurement. Consider max data interval (255 sec) per config.
BAUD_RATE	Baud rate for sensor communication (baud)	921600, 460800, 230400 [460800]	Must match sensor's baud rate setting

FILE_ROTATE_MIN	Duration per measurement file (in minutes)	Integer from 1 to 60 [10]	
A342_PHYSICAL	Physical quantity for M-A342VD/M-A542VR	Velocity, Displacement [Velocity]	
A342_MODE	Output data type for M-A342VD/M-A542VR	Raw, RMS, P-P [Raw]	
A342_RMSPP_OUTPUT_INTERVAL	Output interval (seconds) for RMS/P-P mode on M-A342VD/M-A542VR	Integer from 1 to 255 [1]	For Velocity, actual interval is 1/10 of this value
A352_SPS	Sampling rate (SPS) for M-A352AD/M-A552AR	1000, 500, 200, 100, 50 [1000]	See section 3.2.2 M-A352AD/M-A552AR
A370_SPS	Sampling rate (SPS) for M-A370AD	1000, 500, 200, 100, 50 [1000]	See section 3.2.3 M-A370AD
SENSOR_DATA_DIAG	Enable sensor data anomaly detection	True, False [True]	Enabled when set to True
SENSOR_DATA_DIAG_SEC	Duration (seconds) for anomaly detection	Decimal > 0 and ≤ 10 [1.0]	
MESSAGE_SEND	Enable status message sending	True, False [False]	
MESSAGE_HOST	Hostname for message sending	Valid hostname string[localhost]	Localhost = Raspberry Pi itself
MESSAGE_PORT	Port number for message sending	Integer from 0 to 65535 [1883]	1883 = standard MQTT port
LOG_LEVEL	Minimum log level form output	CRITICAL, ERROR, WARN, INFO, DEBUG [INFO]	Defaults to INFO if invalid value is set

3.5. Measurement Specifications

3.5.1. Measurement Duration

The measurement duration is determined by the startup argument and the configuration file settings.

Table 3-9

Argument	Measurement Duration	Initial Wait Time Setting	Notes
auto	Value of MEASURE_TIME_SEC in the config file (seconds)	Applied	If MEASURE_TIME_SEC set to 0, endless measurement is performed.
Integer from 0 to 2 ⁶³ -1	Specified duration in seconds	Not applied	If the argument is 0, endless measurement performed.

3.5.2. Measurement Transient Response Time

After the measurement start command is sent to the sensor, there is a delay before data is actually received. This delay is referred to as the transient response time, and the number of output data rows will be shorter than the configured measurement duration by this amount.

Table 3-10

Sensor & Measurement Settings	Transient Response Time Until Data Reception
M-A342VD Velocity Measurement	177ms
M-A342VD Displacement Measurement	1,736ms
M-A370AD / M-A352AD 1000SPS Measurement	127ms
M-A370AD / M-A352AD 500SPS Measurement	126ms
M-A370AD / M-A352AD 200SPS Measurement	155ms
M-A370AD / M-A352AD 100SPS Measurement	150ms
M-A370AD / M-A352AD 50SPS Measurement	140ms

3.5.3. Abnormality detection function for sensor data

The sensor data anomaly detection function determines an anomaly if the same value is obtained continuously from the sensor for a certain period. It outputs a message at the WARN level to the log file. The value is evaluated for each of the XYZ axes.

Additionally, if a different value is obtained while an anomaly is being detected, it determines that the anomaly state has been resolved and outputs a message at the INFO level to the log file.

In the configuration file, you can set the ON/OFF status of the function and the continuous evaluation time (seconds).

4. Conducting Measurements

4.1. Preparation

4.1.1. Operation from PC

If the "5.2. PC Remote Connection Settings Using a Wired LAN Cable" section of the "Vibration Measurement System Setup Manual Using Raspberry Pi Products" is completed, you can operate the Raspberry Pi from a Windows PC.

Steps:

1. Launch PowerShell or Command Prompt on the Windows PC.
2. Enter `ssh pi @ 192.168.1.52`
3. When prompted for a password, enter the password set during the initial Raspberry Pi setup.
4. To disconnect from the Raspberry Pi, enter `exit` .

Assumptions used in this manual:

- Raspberry Pi username: `pi`
- Fixed IP address: `192.168.1.52`

If you use different values, please adjust accordingly.

4.1.2. Checking the System Time

Before starting the measurement, if possible, use the `date` command to ensure that the Raspberry Pi's time is correctly maintained. If the time is inaccurate, you can synchronize it using NTP by connecting to the internet. You can also connect to the internet using your smartphone's Wi-Fi tethering feature. If you suspect a malfunction of the RTC-HAT, check its settings and installation status.

For detailed instructions, refer to the "Setup Manual for Vibration Measurement System Using Raspberry Pi Products"

4.2. Measurement Execution

By combining the logger service with Linux commands, you can execute measurements in various ways.

This manual explains the following methods:

- Startup Execution
- Scheduled Execution
 - Scheduled Cycle Execution
 - Scheduled One-Time Execution
- Manual Execution

In all cases, the measurement duration is specified using the method described in section [3.5.1 Measurement Duration].

4.2.1. Startup Execution

This method starts measurement automatically when the Raspberry Pi is powered on and the OS boots up. There is approximately a 30-second delay from power-on to measurement start.

- To enable automatic measurement at startup from the next power-on of the Raspberry Pi, run:

```
sudo systemctl enable logger@<MeasurementDuration>.service
```

- To disable it, execute:

```
sudo systemctl disable logger@<MeasurementDuration>.service
```

(Use the same measurement duration argument as when enabling.)

4.2.2. Scheduled Execution

This method starts measurement at a specified time while the Raspberry Pi is running.

There is a 5–15 second delay from the scheduled time to actual measurement start.

There are two types depending on the command used:

- Scheduled Cycle Execution
- Scheduled One-Time Execution

4.2.2.1. Scheduled Cycle Execution

This method starts measurement periodically based on a registered time pattern (month, day, weekday, hour, minute).

- Execute: `crontab -e` to open the configuration screen for scheduled measurements.
- Enter commands in the following format:

```
Minute Hour Day Month Weekday sudo systemctl start logger@ <MeasurementDuration>.service
```

Figure 4-1

- Weekday: 1 = Monday, 2 = Tuesday, ..., 7 = Sunday
- Use `*` to indicate “any value”
- Each line can specify a separate measurement schedule
- Save and close the file after editing

Example: To perform a 60-second measurement every day at 8:00 AM and 4:00 PM

```
0 8 * * * sudo systemctl start logger@60.service
0 16 * * * sudo systemctl start logger@60.service
```

Figure 4-2

4.2.2.2. Scheduled One-Time Execution

This method starts measurement once at a specified date and time.

- Execute: `at <time> <date>` to open the configuration screen for a one-time execution.
- Enter: `sudo systemctl start logger@<MeasurementDuration>.service`
- Exit the configuration screen with `Ctrl+D`.

Example: To perform a 60-second measurement on April 15, 2024, at 18:15.

```
at 18:15 2024-04-15
sudo systemctl start logger@60.service
Ctrl+D
```

Figure 4-3

4.2.3. Manual Execution

This method starts measurement manually after logging into the Raspberry Pi. There is a 5-10 second delay from command execution to measurement start.

- Execute: `sudo systemctl start logger@<MeasurementDuration>.service`

4.2.4. Notes on Measurement Execution

- Only one measurement can be executed at a time.
- If you try to run the same logger service in parallel, the second one will not start.
 - Example: If the start and end times of the two scheduled measurements ``logger@3600.service`` overlap

- If you try to run a different logger service (e.g., logger@auto.service and logger@3600.service) in parallel, the second one will log an ERROR and terminate without measuring.

4.3. Measurement Confirmation

4.3.1. Checking Measurement Status

While measurement is running, you can check its status by entering the following command:

```
systemctl status logger@<MeasurementDuration>.service
```

If the output shows 'active (running)' in green txt, the measurement is active.

Note: For scheduled execution, the status will now show 'active (running)' until the scheduled start time is reached.

4.3.2. Checking Logs

During measurement, you can monitor log output in real time using the following command:

```
journalctl -f -u logger@<MeasurementDuration>.service.
```

- This displays log messages continuously.
- To stop the log output, press **Ctrl+C**.

4.3.3. Stopping the Measurement

If you want to stop measurement immediately regardless of the configured duration, use the following command:

```
sudo systemctl stop logger@<MeasurementDuration>.service
```

To confirm that measurement has stopped, run:

```
systemctl status logger@<MeasurementDuration>.service
```

If it shows 'inactive (dead)', the measurement has ended.

Note: For RMS and P-P measurements with M-A342VD/M-A542VR, the measurement may not end for up to the number of seconds set in A342_RMSP_OUTPUT_INTERVAL.

Note: The measurement will also automatically stop if you execute the shutdown or reboot command for the Raspberry Pi without executing the above command. However, in such cases, the log file may not be saved in the data storage folder, and the measurement time may not be recorded correctly. Therefore, please execute the above command to stop the measurement.

4.4. Acquiring Measurement Data

If you have set up the system according to the 'Vibration Measurement System Setup Manual Using Raspberry Pi Products,' the measurement data is saved on the internal SD card. The command to copy all measurement data saved in the home directory of the Raspberry Pi user to the current directory on a PC using PowerShell is as follows:

- `scp -r pi@192.168.1.52:measure .`

To copy only a specific measurement date folder, use:

- `scp -r pi@192.168.1.52:measure/<MeasurementStartDateFolder> .`

5. Appendix tool : Sensor Self Test

5.1. General

The sensor self-test tool operates independently of the logger program. It executes the self-test function built into the sensor and outputs the test results to the terminal.

Note: If environmental vibration levels change suddenly during the test, or if they fall below the sensor's floor noise level, the diagnosis may be inconclusive or incorrect.
For details, refer to the datasheet of each sensor.

The self-test tool references the same configuration file as the logger program. All configuration items are checked at startup, just like the logger.

5.2. Input Specification

The sensor self-test tool supports the following sensors :

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

If multiple sensors are connected to the Raspberry Pi, each sensor will be tested individually.

5.3. Output Specification

For each sensor connected to the Raspberry Pi, the self-test results are output to the terminal in the following format.

```
Model: A342
SerialNo.: XXXXXX
Acceleration: OK
Sensitivity: Not Available
Temperature: OK
Power Voltage: OK
Flash Memory: OK
Structural Resonance: X:OK,Y:OK,Z:OK
```

Figure 5-1 Example of Self-Test Output

Table 5-1 Description of Self-Test Items

Item	Description	Type of Value	Remark
Model	Sensor Model	A370, A352, A342	For A552, it is output as A352 For A542, it is output as A342
Serial No	Sensor Serial Number	---	
Acceleration	Abnormality in Acceleration Sensor Values	OK, NG	
Sensitivity	Abnormality in Sensitivity (For each of X,Y,Z axes)	OK, NG, Not Available	Not available for A342

Temperature	Abnormality in Temperature Sensor	OK, NG	
Power Voltage	Abnormality in Power Voltage Level	OK, NG	
Flash Memory	Abnormality in Non-Volatile Memory	OK, NG	
Structural Resonance	Abnormality in Structural Resonance Level (For each of X,Y,Z axes)	OK, NG, Not Available	Not available for A352 and A370

5.4. Execution Specification

1. Adjust the measurement settings to ensure that no measurements are conducted during the self-test execution.
2. Connect the sensor to the Raspberry Pi
3. Log in to the Raspberry Pi from your PC and navigate to the "/app/MSG-002-001a" folder.
4. Enter the following command to execute the self-test tool: : `python -m logger.tool.selftest`
5. The self-test results will be output to the terminal.

6. Appendix tool : Hardware Status Monitor

6.1. General

The hardware status monitor tool operates independently of the logger program. It measures the hardware status at one-minute intervals and outputs the results to a CSV file.

Like the logger program, this tool references the same configuration file. All configuration items are checked at startup to ensure consistency.

6.2. Output Specification

6.2.1. Hardware Status Monitor – Measurement Data File

- A folder named hardware is created under the configured data storage folder.
- Measurement results are saved in a file named:
`<MonitorStartDateTime>_hwmonitor.csv`
- Due to the small amount of data, file rotation is not performed.

```
Time,CPU Temperature(deg C),CPU Usage(%),Memory Usage(%),Disk Usage(%)
2024-08-19 12:22:48,37.485,0.2,2.5,0.0
2024-08-19 12:23:48,36.998,0.1,2.5,0.0
2024-08-19 12:24:48,39.433,0.1,2.5,0.0
```

Figure 6-1 Example of Hardware Status Monitor Output

Table 6-1 Description of Hardware Status Monitor Items

Items	Description
Time	Measured Date and Time (Year, Month, Day, Hour, Minute, Second)
CPU Temperature(deg C)	CPU Temperature (°C)
CPU Usage(%)	CPU Usage Rate
Memory Usage(%)	Memory Usage Rate
Disk Usage(%)	Disk Usage Rate of the Storage Folder

6.2.2. Hardware Status Monitor – Status Messages

The tool also sends hardware status messages using the MQTT protocol, like the logger program.

Table 6-2 Hardware Status Monitor – Messages Specifications

Message Type	Topic Name	Message Content
H/W Measurement Status	hwmonitor/\$LOG	status: "START" "STOP"
H/W Measurement Data	hwmonitor/\$LOG/data	cpu_temperature: 99.999, cpu_usage: 99.9, memory_usage: 99.9, disk_usage: 99.9
H/W Error Status	hwmonitor/\$LOG/error	level: "CRITICAL" "ERROR" "WARNING", message: "Actual error message"

- For details on message format, refer to section [3.3.4 Status Messages].

6.3. Execution Specification

The hardware status monitoring tool can be executed in two different ways, depending on the usage scenario.

6.3.1. Terminal Execution

This method involves executing commands from the terminal on a Raspberry Pi.

1. Log in to the Raspberry Pi from your PC and navigate to the “/app/MSG-002-001a”.
2. Enter the following command to execute the hardware status monitor : `python -m logger.tool.hwmonitor`
3. The hardware status will be output to the terminal at 1-minute intervals and recorded in a CSV file.
4. To terminate the hardware status monitor, enter `Ctrl+C`.

6.3.2. OS Service Registration

This method involves registering the logger program as a service on the Raspberry Pi OS and running it in the background.

1. Edit the service definition file: `nano /app/MSG002-001a/bin/hwmonitor.service`
 - If your Raspberry Pi username is not `pi`, changing line 13 from “User=`pi`” to your username created.
 - If your virtual environment is named something other than `venv`, update line 10 “ExecStart=`.../venv/...`” accordingly.
2. Install the service file:
`sudo cp /app/MSG002-001a/bin/hwmonitor.service /etc/systemd/system/`
3. Reload the service configuration: `sudo systemctl daemon-reload`
4. Check the service status: `systemctl status hwmonitor.service`
If it shows “Loaded: loaded”, registration was successful.

After registering the service, you can execute it in two ways:

- To run immediately: Execute `sudo systemctl start hwmonitor.service`
 - ✧ To stop execution, run `sudo systemctl stop hwmonitor.service`
 - ✧ After rebooting the OS, the service will remain stopped.
- To run always after OS startup: Execute: `sudo systemctl enable hwmonitor.service`
 - ✧ To disable continuous execution, run: `sudo systemctl disable hwmonitor.service`

When registered as a service, you can check the output of the hardware status monitor with the following command:

- `journalctl -f -u hwmonitor.service`
 - ✧ To stop the output, enter `Ctrl+C`

7. Appendix : Log message

Log output messages are categorized into five levels based on their importance. The levels and their meanings are listed in the table below.

Table 7-1

Level	Meaning
CRITICAL	Errors requiring a Raspberry Pi reboot
ERROR	Measurement cannot start or is forcibly terminated due to configuration errors, sensor failures, or connection issues
WARN	Occurrence of invalid packets or data loss
INFO	Measurement start, measurement end, and measurement data file rotation
DEBUG	Arbitrary items output for verification during development

The following is a list of log messages categorized by level (DEBUG level messages are omitted).

The Japanese text in blue represents variables, and the actual messages output will vary depending on the situation.

Table 7-2 INFO Message Level

Output Message	Meaning and Countermeasure
Start initialization	Sensor initialization has started.
Set measurement configuration	Measurement configuration has been set on the sensor.
Found product: ProductModel port= ConnectedUSBPort baud= ConfiguratedBaudRate	The sensor was found on the USB port with the specified baud rate.
Sensor initialized : ProductModel # SensorSerial Number @ ConnectedUSBPort	Sensor initialization is complete.
Initial wait for InitialWaitTime sec	Initial measurement wait has started.
Start measurement	Measurement has started
Initial output to OutputFileName	Notification of the initial output file name.
Output file is rotated to OutputFileName	The output file has been changed
Sensor on axis: Axis is fixed	The sensor data anomaly detection function has resolved the abnormal state.
Terminate measurement	Measurement was interrupted by user operation or error.
Finish measurement	Measurement has ended as the set measurement time has been completed.

Table 7-3 WARN Message Level

Output Message	Meaning and Countermeasure
Failed to connect message broker: rc= ReasonCode , messaging is disabled and measurement will continue.	MQTT connection failed; measurement continues. Check broker settings or disable MQTT.

Invalid packet boundary: beg= InvalidPacketLeadingByte , end= InvalidPacketTrailingByte . Fix packet.	Data corruption has occurred, resulting in an invalid packet that cannot be converted to data. If this message appears frequently, the sensor may be malfunctioning.
Missing NumberOfMissingData data from index: StartingIndex . Complement them.	Data loss was detected by checking the count value, and a complement was performed. If this message appears frequently, the sensor may be malfunctioning.
Sensor on Axis is possibly broken	The sensor data anomaly detection function has detected that the axis data has remained the same value for a certain period, indicating a possible malfunction.

Table 7-4 ERROR Message

Output Message	Meaning and Countermeasure
Another measurement is running, can't start measurement	Measurement could not be executed because the program was started while another measurement was running. Wait for the other measurement to finish or adjust the settings to avoid overlapping measurement times.
Invalid argument length; Argument must be "auto" or an integer.	Incorrect number of arguments. Check startup configuration.
Measurement time argument must be an integer	Measurement could not start because the argument for the measurement time in the scheduled measurement was not an integer. Check the schedule measurement settings for errors.
Config file has wrong value, can't start measurement	Measurement could not start due to an error in the configuration file. Check the configuration file for errors.
ConfigurationKeyName must be defined	The configuration value is not set. Check the configuration file for errors.
ConfigurationKeyName must be one of ListOfPossibleValues	The configuration value is not one of the possible values. Check the configuration file for errors.
ConfigurationKeyName must be between LowerLimit and UpperLimit	The configuration value does not fall within the acceptable range. Check the configuration file for errors.
ConfigurationKeyName must be less than or equal to MaximumOfConfiguration	The configuration value exceeds maximum. Check the configuration file for errors.
ConfigurationKeyName must match pattern RegularExpressionPattern	The configuration value does not match the regular expression. Check the configuration file for errors.
ConfigurationKeyName must satisfy constraint: ConfigurationConstraint	Value violates a constraint. Check the configuration file for errors.
ConfigurationKeyName must be an existing directory: NameOfDirectory	The folder with the given name does not exist. Check the configuration file for errors.
Failed to connect: ConnectionPort	Failed to connect on the port that should be available. The sensor may be in a state where it cannot communicate. Stop sharing power to the sensor and reconnect.
Sensor device: NOT_READY SensorOutputValue	The sensor is not in a usable state. Wait a while and then try the measurement again.
Sensor device: HARD_ERR SensorOoutputValue	The sensor is malfunctioning.
Sensor: ProductModel # SensorSerialNumber @ ConnectedUSBPort failed to start measurement	Failed to start measurement on the sensor. The sensor may be malfunctioning or the USB cable may be disconnected.

Sensor: ProductModel # SensorSerialNumber @ ConnectedUSBPort got error during measurement	An error occurred during measurement, causing the sensor's measurement to stop. The sensor may be malfunctioning, the USB cable may be disconnected, or there may be an issue with the storage disk.
Sensor: ProductModel # SensorSerialNumber @ ConnectedUSBPort failed to stop measurement	Failed to stop measurement on the sensor. The sensor may be malfunctioning or the USB cable may be disconnected. Restart the sensor's power.
No sensor found	No sensor was found. The sensor may be malfunctioning or the USB cable may be disconnected.
Failed to initialize: ConnectionPort	Failed to initialize the sensor connected to that port. The sensor may be malfunctioning or the USB cable may be disconnected.
Failed to initialize measurement	Failed to initialize the sensor. The sensor may be malfunctioning or the USB cable may be disconnected.
All sensor failed to start measurement	Failed to start measurement on all sensors. The sensors may be malfunctioning or the USB cables may be disconnected.
All sensor failed to measure	Measurement stopped due to failure to start measurement or an error during measurement. The sensors may be malfunctioning or the USB cables may be disconnected.
Failed to measure	An error occurred during measurement. Detailed error logs are output simultaneously, so check the contents.
Unknown error occurred	An unexpected error occurred during program execution. Detailed error logs are output simultaneously, so check the contents.
An error occurred during stop measurement	An error occurred during the measurement stop process. The sensor may be malfunctioning or the USB cable may be disconnected. Restart the sensor's power.
An error occurred while reading data	An error occurred while reading data. The sensor may be malfunctioning or the USB cable may be disconnected.
An error occurred while writing data	An error occurred while writing data. There may be an issue with the storage disk or it may be full.
Failed to send stop command	Failed to send the stop measurement command. The sensor may be malfunctioning or the USB cable may be disconnected. Restart the sensor's power.
Failed to disconnect device connection	Failed to disconnect the sensor. Restart the Raspberry Pi's power.
Can't start measurement	Measurement could not start. Detailed error logs are output simultaneously, so check the contents.
Can't continue measurement	Measurement could not continue. Detailed error logs are output simultaneously, so check the contents.
An error occurred while initializing logging	An error occurred within listener process.
An error occurred within listener process	Error occurred in the internal listener process. Check the detailed error log.

Table 7-5 CRITICAL Message Level

Output Message	Meaning and Countermeasure
Failed to notify finishing writer process	Failed to finish the writing process. Restart the Raspberry Pi's power.
Failed to get reader process finish message	Failed to finish the reading process. Restart the Raspberry Pi's power.

8. Contact Information

Seiko Epson Corporation

Sales Headquarters MD Sales Department

Contact Information via the Internet

https://global.epson.com/products_and_drivers/sensing_system/privacy/area_select_inquiry_contact.html