EPSON

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 dose not assume any responsibility or liability of any kind of damage and/or fire coursed 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 BEFOR USE 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 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 contens

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
		12, 15, 17	• Addition of sensor data anomaly detection function, and associated corrections
		19	• 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
		23	• Addition of a hardware status monitor tool

1. Related Documents

『Vibration Measurement System Setup Manual using Raspberry Pi products』

『M-A352AD Datasheet』

『M-A342VD Datasheet』

『M-A552AR Datasheet』

『M-A542VR Datasheet』

2. Introduction

This manual is for using the measurement system created using the "Vibration Measurement System Setup Manual Using Raspberry Pi Products." This manual describes various specifications and measurement methods.

The "logger" described in this manual refers to a configuration in which sensors and other necessary equipment are connected to a Raspberry Pi in order to measure vibrations using sensors.

3. Specification

3.1. Compatible Sensor

The following sensors can be measured using the logger.

M-A352AD

M-A342VD

M-A552AR

M-A542VR

3.2. General

This logger allows you to specify input and output settings by writing the settings in a text-format configuration file.

For information on how to write the configuration file and the items that can be set, please refer to "5.1. Configuration Items".

3.3. 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.

3.3.1. 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

3.3.2. 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

3.4. Output Specification

3.4.1. 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 Output Folder

3.4.2. Measurement Data File

The file name of the measurement data file is "Sensor name_Serial number_LOGGER ID (specified in the setting file)_Connection serial number_YYMMDD_hhmmss.csv". 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 "5.1. 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 outpit,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 3-3 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 Tables 3 4 for the meaning of each digit.

Table 3-4 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.

3.4.3. Measurement Information File

The filename of the measurement information file is "sensor name _ serial number_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-6.1.21-v8+-aarch64-with-glibc2.31
LOGGER_VERSION,1.1.0
LOGGER_ID,R4
START_TYPE,manual
TIME_ZONE,UTC+0900
DATE,2024/09/09 08:14:06
MEASURE_TIME_SEC,86401
INITIAL_WAIT,1
FILE_ROTATE_MIN,10
BAUD_RATE,460800
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,1
SENSOR_DATA_DIAG,True
SENSOR_DATA_DIAG_SEC,1.000000
```

Figure 3-3: Example Output of the Measurement Information File

Table 3-5 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	Measurement start methods. One of the following values. auto：automatic start schedule：scheduled measurement manual：manual measurement	
TIME_ZONE	Time zone set on the Raspberry Pi performing the measurement	
DATE	Measurement start day and time	Local time without 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	
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	Only output for M-A352AD, empty string for M-A342VD
RMS_PP_INTERVAL	RMS and P-P measurement interval for M-A342VD	Empty string for M-A352AD. Also output for raw data in M-A342VD, but not used in measurement
SENSOR_DATA_DIAG	Status of the sensor data anomaly detection function	True: enabled
SENSOR_DATA_DIAG_SEC	Anomaly detection decision time for the same function	

3.4.4. Log File

The log file is named "measure.log" and is output in text format. During measurement, it is output as a "measurement start date time.log" file in the "/app/MSG002-001a/tmp_log" folder. 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 the types and descriptions of log messages, please refer to "8. 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 3-4 Example of Log Data

3.5. Measurement Specification

The following three types of measurements are possible:

- Automatic measurement
- Scheduled measurement
- Manual measurement

The method for specifying the measurement time varies depending on the measurement method.

3.5.1. Automatic Measurement

This method automatically starts the measurement when the Raspberry Pi is powered on. The measurement time is specified in the configuration file. There is approximately a 30-second delay from power-on to the start of the measurement.

3.5.2. Schedule Measurement

This method starts the measurement by setting the start time using Linux commands. For detailed command input instructions, please refer to "6.2.2. Scheduled Measurement". The measurement time is specified when setting the start time. There is approximately a 10-second delay from the specified measurement start time to the actual start of the measurement.

3.5.3. Manual Measurement

This method starts the measurement manually using Linux commands. The measurement time is specified in the configuration file. There is approximately a 5-second delay from executing the command to the start of the measurement.

3.5.4. Notes on Measurement Time

There is a delay (transient response time) from when the measurement start command is sent to the sensor until data can actually be received. The number of data rows output will be shorter than the specified measurement time by the duration of this delay.

Table 3-6

Sensor Measurement Settings	Transient Response Time Until Data Reception
M-A342VD Velocity Measurement	177ms
M-A342VD Displacement Measurement	1736ms
M-A352AD 1000SPS Measurement	127ms
M-A352AD 500SPS Measurement	126ms
M-A352AD 200SPS Measurement	155ms
M-A352AD 100SPS Measurement	150ms
M-A352AD 50SPS Measurement	140ms

3.5.5. 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. Operation from PC

If the "4.2. 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.

1. Launch PowerShell or Command Prompt on the Windows PC.
2. Enter `ssh Raspberry Pi Username Specified During Initial Setup @ Assigned IP`
3. When prompted for a password, enter the password set during the initial Raspberry Pi setup.
4. Following these steps, you can start or stop measurements and check logs from the Windows PC as described in the subsequent sections.
5. To disconnect from the Raspberry Pi, enter `exit` .

5. Carrying out measurements

5.1. Setting Item

Settings are specified in the "/app/MSG002-001a/.env" file in the "KEY=VALUE" format. Regardless of the product model of the sensor used for measurement, all items except for LOG_LEVEL must be specified with valid values. The settings are checked when the program starts, and if any value is invalid, an error occurs, and the program terminates.

Note: An error will occur if there are spaces before or after the "=" or if the setting KEY name is changed.

The configurable items are as follows:

Table 5-1

Setting Key Names	Description	Configurable Values (VALUE)	Supplementary Information
LOGGER_ID	Identifier for the Raspberry Pi used for measurement	Combination of uppercase and lowercase alphanumeric characters, 1 to 63 characters long	
OUTPUT_PATH	Data storage folder	Path to an existing directory	
INITIAL_WAIT	Initial waiting time (seconds)	Integer value from 0 to 3600 seconds	
MEASURE_TIME_SEC	Measurement time for automatic and manual start (seconds) *Not used for scheduled measurements	Integer value from 0 to (2^{63}-1)	Setting 0 results in endless measurement. Measurement data interval can be up to 255 seconds depending on the measurement settings. Consider the measurement data interval for each setting.
BAUD_RATE	Baud rate used for communication with the sensor	921600,460800, 230400	Must match the sensor's baud rate setting
FILE_ROTATE_MIN	Time to output to one file (minutes)	Integer value from 1 to 60	
A342_PHYSICAL	Measurement physical quantity for M-A342VD/M-A542VR	Velocity, Displacement	
A342_MODE	Output data type for M-A342VD/M-A542VR	Raw, RMS, P-P	
A342_RMSPP_OUTPUT_INTERVAL	Output interval seconds when RMS or P-P output is selected for M-A342VD/M-A542VR	Integer value from 1 to 255	If the physical quantity is Velocity, it is output at 1/10 of the set value interval
A352_SPS	Sampling rate (SPS) for M-A352AD/M-A552AR communication	1000, 500, 200, 100, 50	Refer to "3.3.2. M-A352AD/M-A552AR" for filter settings for each sampling rate.
SENSOR_DATA_DIAG	Enable sensor data anomaly detection function	True, False	Enabled if set to True
SENSOR_DATA_DIAG_SEC	Number of seconds for continuous same value to be considered an anomaly for the same function	Decimal value greater than 0 and less than or equal to 10	
LOG_LEVEL	Minimum log output level	CRITICAL, ERROR, WARN, INFO, DEBUG	If a value other than the configurable ones is set, INFO will be applied

5.2. How to set up measurement

5.2.1. Automatic Measurement

By executing `sudo systemctl enable logger@auto.service` command, the measurement will automatically start from the next power-on of the Raspberry Pi.

To disable automatic startup, execute `sudo systemctl disable logger@auto.service`.

5.2.2. Schedule Measurement

5.2.2.1. Period Measurement

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

To set up repeated measurements, enter `*` . You can set individual periodic measurements on each line.

```
Minute Hour Day Month Week sudo systemctl start logger@ Measurement time.service
```

Figure 5-1

Specify the measurement time as a number between 1 and (2^{63} - 1), similar to the MEASURE_TIME_SEC setting in the configuration file.

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 5-2

5.2.2.2. One measurement

Execute `at time date` command to open the configuration screen for a one-time execution.

Enter `sudo systemctl start logger@measurement_time.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 04152024
sudo systemctl start logger@60.service
Ctrl+D
```

Figure 5-3

5.2.3. Manual Measurement

Execute `sudo systemctl start logger@manual.service`

5.2.4. Notes on Measurement

Only one measurement can be executed at a time. If you attempt to run another measurement service with a different name simultaneously, it will log an ERROR level message and terminate without performing the measurement.

(Example: When the automatic measurement service `logger@auto.service` is running and the scheduled measurement service `logger@3600.service` is started)

If you attempt to run a measurement service with the same name simultaneously, the later measurement service will not be executed.

(Example: If the start and end times of the scheduled measurement `logger@3600.service` overlap)

5.3. Measurement

5.3.1. Check the 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 WiFi 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"

5.3.2. Starting Measurement

Please initiate the measurement by entering the command according to the aforementioned measurement settings

Execute `systemctl status logger@ values suitable for the initialization method.service` and if 'active (running)' is displayed in green text, the measurement has commenced.

Alternatively, by executing `systemctl status "logger*"` you can verify the status regardless of the initialization method.

Please note that in the case of scheduled measurements, 'active (running)' will not be displayed until the specified measurement start time.

5.3.3. During Measurement

You can verify the log output during measurement by executing `journalctl -f -u logger@ value-suitable-for-initialization-method.service.`

Example：

In the Case of Automatic Startup : `journalctl -f -u logger@auto.service`

In the Case of Schedule Startup : `journalctl -f -u logger@ specified measurement time.service`

In the Case of Manual Startup : `journalctl -f -u logger@manual.service`

To stop the log output, press `Ctrl+C`

5.3.4. Stopping the Measurement

Regardless of the measurement initiation method, you can stop the measurement by executing

`sudo systemctl stop logger@ values suitable for the initialization method.service`

If you execute `systemctl status logger@ values suitable for the initialization method.service` and 'inactive (dead)' is displayed, the measurement has ended. Please note that when performing 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_RMSPP_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.

5.4. 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 Raspberry Pi username specified during initial setup @assigned IP : measure .
```

6. Appendix tool : Sensor Self Test

6.1. 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.

6.2. Input Specification

The sensor self-test tool supports the following sensors：

M-A352AD

M-A342VD

M-A552AR

M-A542VR

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

Additionally, the sensor self-test tool operates by referencing the configuration items of the logger program. It refers to all items in the configuration file, and the same level of configuration content checks as the logger program are performed when the tool is initiated.

6.3. 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 6-1　Example of Self-Test Output

Table 6-1 Description of Self-Test Items

Item	Description	Type of Value	Remark
Model	Sensor Model	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

6.4. 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.

Note: Please execute this function in a stationary state.

7. Appendix tool : Hardware Status Monitor

7.1. 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

7.2. Output Specification

The hardware status monitor tool operates by referencing the configuration items of the logger program (reference items: OUTPUT_PATH, LOG_LEVEL).

It creates a 'hardware' folder under the 'data storage folder' and outputs the measurement results to a file named 'monitor_start_date_time_hwmonitor.csv'. Due to the small amount of output data, file switching 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 7-1 Example of Hardware Status Monitor Output

Table 7-1 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

7.3. Execution Specification

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

7.3.1. Run Terminal

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`.

7.3.2. 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. Modify line 12 of the “/app/MSG002-001a/bin/hwmonitor.service” file, changing “User=pi” to the username created during the initial setup of the Raspberry Pi.
2. Execute `sudo cp /app/MSG002-001a/bin/hwmonitor.service /etc/systemd/system/` to install the service configuration file.
3. Execute `sudo systemctl daemon-reload` to reload the service configuration

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`

8. 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 8-1

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 8-2 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: product model port=connected USB port baud=configurated baud rate	The sensor was found on the USB port with the specified baud rate.
Sensor initialized : product model #sensor serial number @connected USB port	Sensor initialization is complete.
Initial wait for initial wait time sec	Initial measurement wait has started.
Start measurement	Measurement has started
Initial output to output file name	Notification of the initial output file name.
Output file is rotated to output file name	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 8-3 WARN Message Level

Output Message	Meaning and Countermeasure
Invalid packet boundary: beg=invalid packet leading byte], end= invalid packet trailing byte. 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 number of missing data data from index: starting index. 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 8-4 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.
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.
Configuration key name must be defined	The configuration value is not set. Check the configuration file for errors.
Configuration key name must be one of list of possible values	The configuration value is not one of the possible values. Check the configuration file for errors.
Configuration key name must be between lower limit and upper limit	The configuration value does not fall within the acceptable range. Check the configuration file for errors.
Configuration key name must match pattern 正 regular expression pattern	The configuration value does not match the regular expression. Check the configuration file for errors.
Configuration key name(folder name) must be an existing directory	The folder with the given name does not exist. Check the configuration file for errors.
Failed to connect: connection port	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 sensor output value	The sensor is not in a usable state. Wait a while and then try the measurement again.
Sensor device: HARD_ERR sensor output value	The sensor is malfunctioning.
Sensor: product model #sensor serial number @connected USB port failed to start measurement	Failed to start measurement on the sensor. The sensor may be malfunctioning or the USB cable may be disconnected.
Sensor: product model #sensor serial number @connected USB port 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: product model #sensor serial number @connected USB port 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: connection port	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.

Table 8-5 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.

9. 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