HardwareX 7 (2020) e00109

Contents lists available at ScienceDirect

HardwareX

journal homepage: www.elsevier.com/locate/ohx

Hardware Article

IoT connected device for vibration analysis and measurement

Ivar Koene *, Ville Klar, Raine Viitala

Aalto University School of Engineering, Espoo, Finland

ARTICLE INFO

Keywords:
MEMS
IoT
Monitoring
Accelerometer
Vibration

ABSTRACT

Accelerometers are widely used in applications, such as condition measurement, motion tracking, and vehicle monitoring. Most handheld smart devices, such as smart phones and tablets, employ accelerometers for motion tracking. The size constraints of such devices has provided the impetus to develop precise, affordable, compact, and power-efficient accelerometers. Microelectromechanical systems (MEMS) accelerometers meet these strict requirements, thus explaining their wide adoption in the smart device market. The increasing cost-performance ratio, particularly regarding accuracy and bandwidth, of modern MEMS accelerometers enables their use in demanding measurement applications. This research introduces an open source battery-powered Internet of Things MEMS accelerometer called Memsio. Memsio is a versatile wireless sensor unit that can be deployed to measure a wide range of different motions. It is operated from a web browser, thus rendering it remotely accessible via smart phone, tablet, or computer. Furthermore, Memsio offers high-speed motion data acquisition capabilities and can store extended measurement periods lasting dozens of minutes; in addition, its wireless and portable nature eases integration in condition monitoring applications. The study presented reports on the construction and functionality of Memsio. Additionally, the functionality of Memsio is demonstrated in the measurement of translational movement of a linear actuator and the vibrations of large rotating machinery.

Specifications table:

Hardware name	*Memsio*
Subject area	• *Engineering* • *Educational Tools and Open Source Alternatives to Existing Infrastructure*
Hardware type	• *Measuring physical properties and in-lab sensors* • *Field measurements and sensors* • *Electrical engineering and computer science* • *Mechanical engineering*
Open source license	*GNU GPL v3*
Cost of hardware	*80 US$*
Source file repository	*https://doi.org/10.17605/OSF.IO/YX687*

* Corresponding author.
E-mail address: ivar.koene@aalto.fi (I. Koene).

https://doi.org/10.1016/j.ohx.2020.e00109
2468-0672/

1. Hardware in context

Mechanical systems are characterized by the motion of different trajectories, amplitudes, and frequencies. A quantitative measurement and subsequent analysis of this motion are crucial for understanding the interrelations of machine elements as well as improving their performance. Among the most common sensors for such measurements are accelerometers, which are widely used in both industry and science to measure translational, rotational, and oscillating motion.

A key advantage of accelerometers is their versatility. They are integral in navigation systems, orientation sensing, condition monitoring as well as stabilization and motion control of closed-loop actuators. While conventional piezoelectric accelerometer (PA) based systems provide high measurement accuracy and frequency, their high cost impedes deployment in embedded measurement applications.

In the embedded domain, microelectromechanical systems (MEMS) accelerometers are an attractive alternative to PAs due to their small size, low cost, and low power consumption. MEMS accelerometers have been intensively researched and developed for use in mass-produced handheld devices, such as smart phones, tablets and smart watches. This development has not only greatly decreased the size and unit costs, but also stimulated the development of MEMS accelerometers with increasingly higher precision and bandwidth [1].

The accuracy and bandwidth of a MEMS sensor depend on the structure of its sensing elements. On one hand, a high natural frequency of the sensing elements results in a broad bandwidth. On the other hand, the sensitivity is inversely proportional to the square of the natural frequency [2]. In other words, the design of MEMS based sensor systems revolves around the trade-off between accuracy and bandwidth. Hence, research on the dynamic behavior of the sensing elements is crucial. The dynamic and static behavior of such elements has been extensively studied in the context of various types of sandwich-structured composites [3–5], nano- and microstructures [6–10], as well as carbon nanotubes [4,11].

The small size of MEMS accelerometers allows them to be easily attached to different objects without influencing the mechanical behavior of the device under test (DUT). Additionally, the data acquisition using a MEMS accelerometer requires fewer components compared to PA based systems. Most modern MEMS accelerometers can directly interface with inexpensive micocontrollers over a serial data bus. Furthermore, MEMS accelerometers are typically capable of concurrently capturing acceleration data along multiple axes, enabling acquisition of three-dimensional motion. MEMS accelerometers have been employed in a wide range of measurement systems including impact quantification [12,13], vehicle monitoring [14], vibration analysis of rotating machinery [15–18], as well as gesture and motion tracking [19–21].

The well-known downside of MEMS based accelerometers is their higher noise density compared to PAs. The high noise density has restricted the adoption of MEMS based measurement systems, such as in condition monitoring. Nonetheless, the advantages achievable with MEMS accelerometers have incentivized research in this area. For example, Albarbar et al. [16] compared three different MEMS accelerometers to a piezoelectric accelerometer in a sinusoidal, random, and impulse excitation. Two of the three tested MEMS accelerometers performed adequately and were deemed suitable for condition monitoring.

Recently, Koene et al. [15] investigated the suitability of MEMS accelerometers for measuring vibrations in large rotating machinery. Large rotating machinery, defined here as machinery in which the mass of the rotating object exceeds 500 kg, is often operated below the critical speed (first mode natural frequency). In such machinery, the vibrations occur at low frequencies below 50 Hz [15]. This frequency range is within the bandwidth specification of common MEMS accelerometers. Koene et al. successfully demonstrated that despite a higher noise density compared to a PA based measurement system, the determination of the main harmonic components in the subcritical rotor speed range was possible using MEMS accelerometers.

In many applications, compact and low-powered MEMS sensors are combined with wireless communication [15,22–25]. This allows deployment into remote places and demanding environments in which installing wired communication is excessively laborious or costly. Wind turbines [26,22] and paper machines [15] are examples of machines which can benefit from the wireless communication. Not only are wind turbines typically located in remote areas, but many sections of the turbine are also difficult to access; thus, wireless data acquisition and facile instrument installation are highly appealing features in such applications. Similarly, in paper machines, the wireless sensors have advantages over wired sensors in the installation process. Retrofitting paper machines with measurement instruments is an expensive process as production has to be halted during installation. With wireless sensors, the installation time is shorter compared to wired sensors resulting in cost-savings due to decreased downtime. Faster installation also expedites troubleshooting. Efficient troubleshooting is useful when unexpected behavior occurs in a machine and the cause cannot be found with the mounted sensors. In addition to the measurement of the static parts of the machinery, the rotating parts can also be outfitted with sensors. The behavior of wind turbine blades [22] and rotating shafts [27,24,28,17] has been investigated using such a sensor configuration. Measurement in these scenarios typically requires wireless data transmission and battery-powered sensor nodes.

In the past decades, the development of free and open source hardware (OSHW) has increased accessibility as well as improved the transparency of research [29,30]. OSHW development has a high return on investment and can accelerate research across a wide range of scientific fields. The versatility, cost-effectiveness, and ease of integration present MEMS accelerometers as an ideal candidate for OSHW motion measurement systems. Consequently, Memsio was developed.

Memsio is a cost-effective Internet of Things (IoT)-enabled MEMS accelerometer measurement system. It is portable, battery-powered and wireless, allowing a wide range of motion analysis. The entire system can be manufactured for less

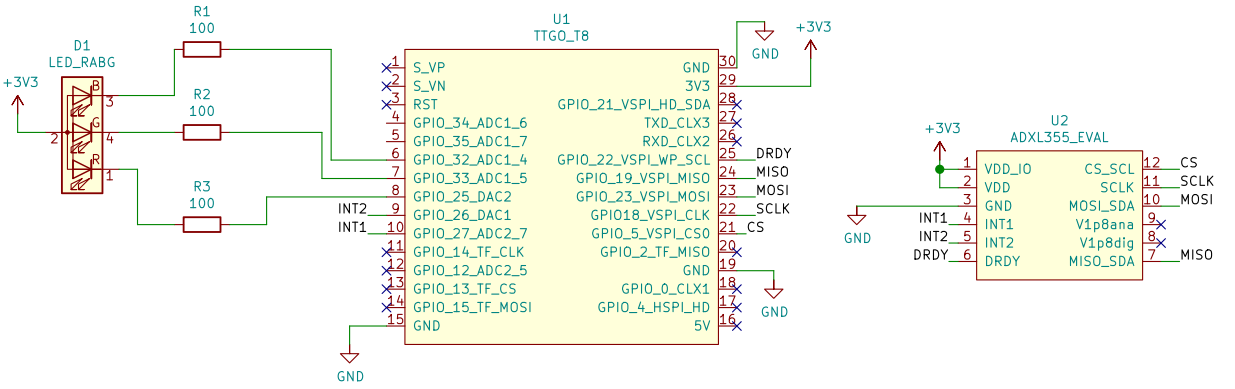

Fig. 1. Schematic of the Memsio circuit.

than 80 USD and can be mounted on a wide range of objects using either fasteners or magnetic feet. Additionally, Memsio has two novel features which distinguish it from prior IoT-enabled MEMS accelerometer measurement systems. First, Memsio utilizes high-capacity onboard flash memory to enable acquiring and storing samples at the highest rate of the utilized MEMS accelerometer for measurement periods lasting dozens of minutes. Secondly, Memsio is operated via a browser interface contained within its own firmware. Consequently, measuring with Memsio does not require installation of any extraneous software apart from a web browser. Herein, we demonstrate the ways Memsio can be used to analyze both vibrations of large rotating machinery and translational motion of a linear actuator.

2. Hardware description

Memsio's main components are a 3-axis MEMS accelerometer, system on chip (SoC) microcontroller unit (MCU), secure digital (SD) memory card, battery and red-green–blue light emitting diode (RGB LED). The data from MEMS accelerometer is read by the MCU which writes the data onto the SD card. The stored measurement data can be accessed either by reading the SD card with a computer or via a browser-based graphical user interface (GUI). The GUI also allows remote control of Memsio, which includes initiating measurements and specifying measurement parameters. A rechargeable lithium-ion (Li-Ion) battery powers the system and the RGB LED indicates the program state. A schematic diagram of Memsio circuit is shown in Fig. 1.

Memsio is based on the ADXL355 (Analog Devices) 3-axis MEMS accelerometer with a 20-bit digital output[1]. Its output data rate (ODR) can be set between 3.906 and 4000 Hz. The low-pass filter corner is determined by the ODR, thereby establishing the bandwidth. In addition to the low-pass filter, ADXL355 has a high-pass filter, which can be set from 0.0095 to 10 Hz when the ODR is 4000 Hz. Memsio's circuit board is designed around the EVAL-ADXL355Z[2] evaluation board of the sensor. The specifications of the ADXL355 are summarized in Table 1.

The ADXL355 communicates with the ESP32-WROVER SoC MCU (Espressif)[3] over a serial peripheral interface (SPI) data bus. The ESP32 has an integrated 2.4 GHz radio module, which enables Wi-Fi and Bluetooth connectivity. It is widely used in various OSHW projects, such as web radios [31] as well as microfluidics [32] and syringe pump control systems [33]. The specifications of the ESP32-WROVER microcontroller are summarized in Table 2.

The popularity of the ESP32 among electronics hobbyists has resulted in the availability of a wide range of different development boards with various peripheral devices. Memsio utilizes the TTGO T8 V1.7[4] development board which features an SD card slot, battery connection, battery management circuit, and an external antenna.

2.1. Firmware overview

The firmware of Memsio was written using the Arduino abstraction layer developed by Espressif.[5] This abstraction layer is built on top of the Espressif IoT Development Framework (ESP-IDF),[6] which is based on the FreeRTOS kernel.[7] This software architecture enables use of the established FreeRTOS application programming interface (API) types, functions, and macros within the Arduino development environment for actions, such as task scheduling and management.

[1] https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_355.pdf
[2] https://www.analog.com/media/en/technical-documentation/user-guides/EVAL-ADXL354-355-UG-1030.pdf.
[3] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
[4] https://github.com/LilyGO/TTGO-T8-ESP32.
[5] https://github.com/espressif/arduino-esp32.
[6] https://docs.espressif.com/projects/esp-idf/en/latest/.
[7] https://freertos.org/.

Table 1
Specifications of ADXL355.

Measurement range (g)	±2.048, ±4.096, ±8.192
Sensitivity (μ g/digit)	3.9, 7.8, 15.6
Noise density ($\mu g/\sqrt{Hz}$)	25
Output data rate (Hz)	3.906–4000
Bandwidth (Hz)	1–1000
High-pass filter for 4000 Hz ODR (Hz)	0.0095–10

Table 2
Specifications of ESP32-WROVER.

Microprocessor unit	Tensilica Xtensa dual-core 32-bit LX6
Clock frequency(MHz)	240
Wi-Fi	802.11 b/g/n (2.4–2.5 GHz)
Bluetooth	v4.2 BR/EDR and Bluetooth Low Energy
SRAM (KB)	520
Integrated PSRAM (MB)	8
External SPI Flash (MB)	4

Unlike conventional Arduino based firmware, the Memsio firmware was written relying on the FreeRTOS task management features. The performant dual core processor of the ESP32 enables parallelization of tasks. This parallelization enables high speed acquisition and storage of the accelerometer samples. The program execution is divided into three different tasks termed GUI task, sensor task, and storage task. The task structure and functionality within them are illustrated in Fig. 2.

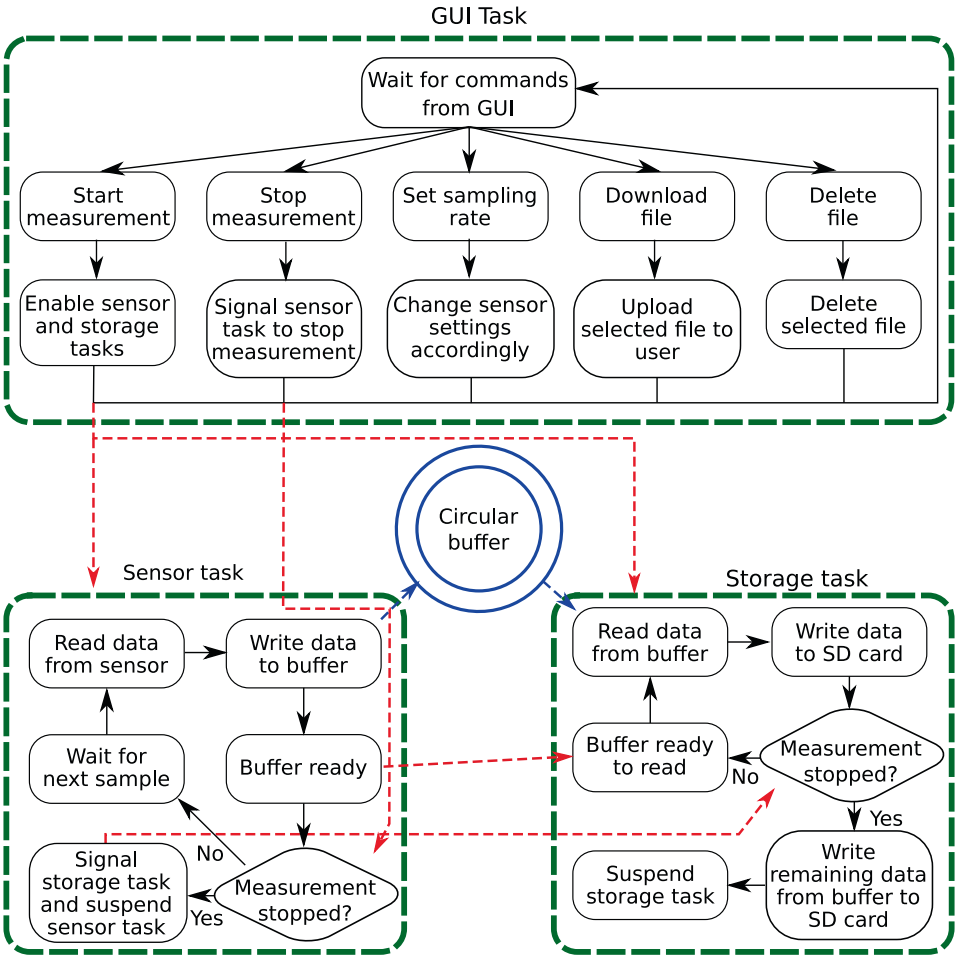

Fig. 2. State diagram of the Memsio firmware in which functions are visualized in boxes, arrows indicate transitions, and dashed red arrows indicate intertask communication. The GUI task processes commands from the GUI, whereas the sensor and storage tasks manages the measurement and storage.

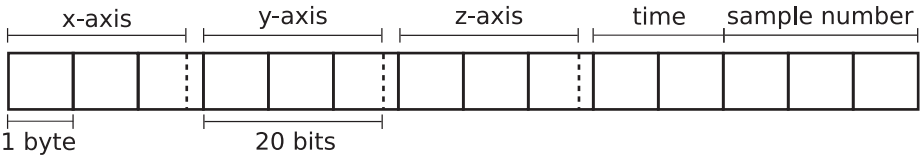

Fig. 3. Structure of sample byte array. The first 9 bytes are the sensor data, the next 2 bytes are occupied for a time elapse from last sample, and the 3 last bytes indicate the sample number.

As the name suggests, the GUI task awaits commands from the browser-based GUI using a websocket based communication. When a measurement procedure is initiated via the GUI, the sensor and storage tasks are concurrently initiated. The sensor task is executed on core 0 and the storage task is executed on core 1. These two tasks run in parallel to ensure that all the samples from the sensor are acquired and stored on the SD card.

Measurement starts when the GUI task receives a "start measurement" message and initiates the sensor and storage tasks. The sensor task starts to read samples from the accelerometer and saves them to a circular first in first out (FIFO) buffer. When a sufficient number of samples are saved to the buffer (between 62 and 2000 samples depending of the ODR), the sensor task informs the storage task that the buffer is ready to be written on the SD card. The storage task does as suggested and the loop continues until a "stop measurements" message comes from the GUI. When this message arrives, the GUI task passes it to the sensor task, which will read and save one more sample before passing the message to the storage task. At this point, the sensor task suspends itself. After receiving the message, the storage task will save the rest of the data from the buffer onto the SD card as well as suspend itself. To avoid errors in the buffer, the message for stopping the measurement is not passed straight from the GUI task to the storage task.

The timing of the acquisition of accelerometer data is performed using the data ready (DRDY) signal of the ADXL355. This pin is connected to an external interrupt of the ESP32. Thus, the sample rate is determined by the ODR of the ADXL355. Sample data is formatted into byte arrays. Each sample byte array has a length of 14 bytes. The structure of the byte array is depicted in Fig. 3. The raw data from the sensor is directly written into the first 9 bytes of the array with data from each axis occupying 3 bytes. The acceleration value of each axis is formatted in a two's complement 20-bit value. Bytes 10 and 11 contain the timestamp of the sample. The timestamp data is delta encoded in microseconds. The final 3 bytes store the sample index.

Memsio utilizes the combination of a circular buffer and SD card for sample data storage. If data is stored on the internal memory of the ESP32, the memory capacity is exceeded in measurements lasting only a few minutes. Buffered writing to the SD card and using two RTOS tasks, one for data acquisition from the sensor and one for writing to memory, enable substantially longer measurements without sacrificing sample rate.

3. Design files

3.1. Design files summary

See Table 3.

4. Bill of materials

See Table 4.

5. Build instructions

Tools required

- M3 Allen key
- Hand drill
- Soldering iron and solder

Table 3
Design files.

Design filename	File type	Open source license	Location of the file
Enclosure top	STL and STP	GNU GPL v3	https://doi.org/10.17605/OSF.IO/YX687
Enclosure bottom	STL and STP	GNU GPL v3	https://doi.org/10.17605/OSF.IO/YX687
PCB	KiCad project and Gerber	GNU GPL v3	https://doi.org/10.17605/OSF.IO/YX687
Firmware	.cpp and.h	GNU GPL v3	https://doi.org/10.17605/OSF.IO/YX687
Binary to ASCII script	.py	GNU GPL v3	https://doi.org/10.17605/OSF.IO/YX687

Table 4
Bill of material.

Designator	Component	Number	Cost per unit currency (US$)	Total cost (US$)	Source of materials	Material type
MCU	TTGO T8 V1.7 (ESP32)	1	7.70	7.70	Aliexpress	Electronics
ACC	EVAL-ADXL355	1	35	35	DigiKey	Electronics
LED	OSTAMA 5B31A	1	0.99	0.99	Starlec OY	Electronics
Battery	16340 (700mAh)	1–2	6.78	6.78–13.56	Starlec OY	Electronics
Battery holder	16340 battery holder	1–2	0.65	0.65–1.3	Aliexpress	Electronics
SD-card	SD-card	1	14.34	14.34	Dustin	Electronics
Resistor	Resistor (approximately 100 ohm)	3	–	–	–	Electronics
PCB	Printed circuit board	1	–	–	–	Electronics
2 × 3 male pin header	2 × 3 male pin header	2	–	–	–	Electronics
1x15 male pin header	1 × 15 male pin header	2	–	–	–	Electronics
2 × 3 female pin header	2 × 3 female pin header	2	–	–	–	Electronics
Magnets	Pot magnet with threaded stem Ø 13 mm	4	2.38	11.85 (min order 5 pcs)	Super magnete	Magnet
M3 nut	M3 nut	8	–	–	–	Fastener
Long M3 nut	M3 nut (14 mm)	4	–	–	–	Fastener
M3x16 bolt	M3 bolt (16 mm)	8	–	–	–	Fastener
M3x25 bolt	M3 bolt (25 mm)	4	–	–	–	Fastener
Top casing	Top casing	1	–	–	–	3D-printed part
Bottom casing	Bottom casing	1	–	–	–	3D-printed part

5.1. Assembly

1. Solder the 2 × 3 female headers and 100 Ω resistors to the PCB.
2. Solder the 1 × 15 male headers to the MCU (Fig. 4a)).
3. Solder the MCU to the PCB (Fig. 4b)).

   - Fig. 4c) present the finished PCB.

4. Solder the 2 × 3 male headers to the ACC (Fig. 4d)).
5. 3D print parts listed in Table 3.
6. Insert the M3 nuts into the top and bottom enclosure (Fig. 5a) and b)).

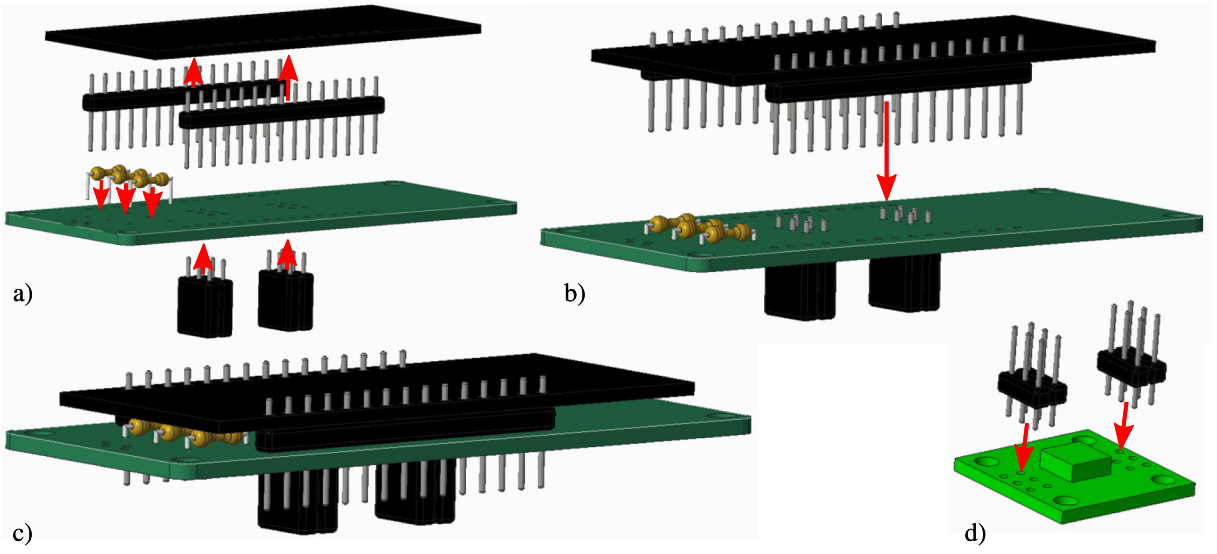

Fig. 4. PCB assembly. a) Resistors and 2 × 3 female headers are soldered to the PCB, and 1X15 male headers are soldered to the MCU. b) MCU is soldered to the PCB. c) Finished PCB. d) 2 × 3 male headers are soldered to the ACC.

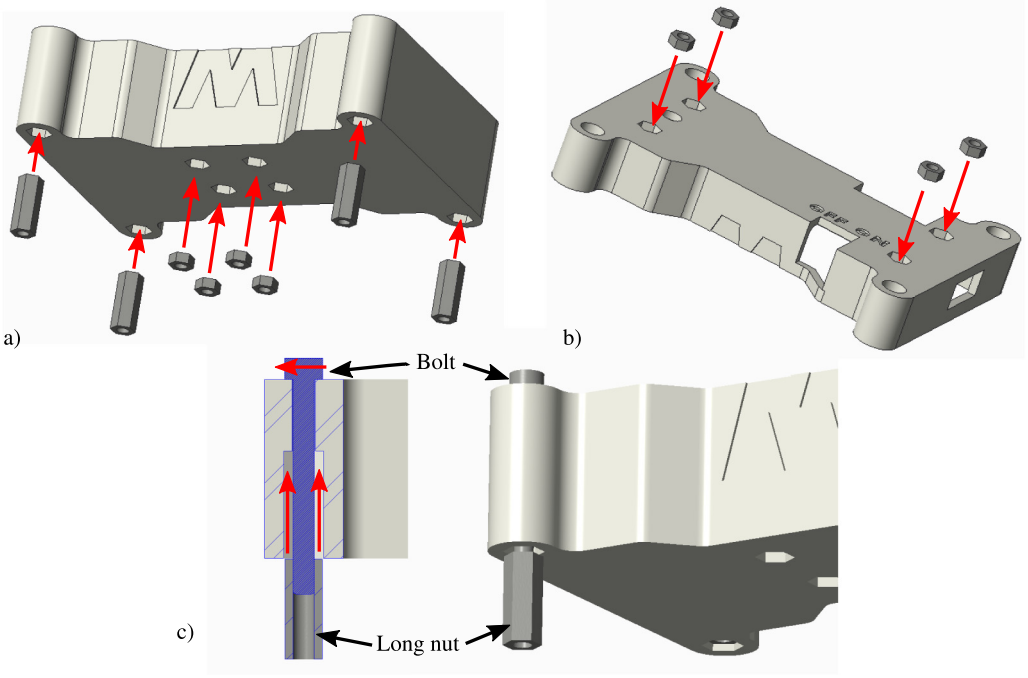

Fig. 5. Nuts that needs to be press-fitted to the enclosure: a) Bottom enclosure and b) top enclosure. c) Indicates the way the nuts are pressed into the enclosure with the bolt.

- The M3 nuts have an interference fit with the top and bottom parts of the enclosure. They can be inserted using a bolt threaded onto them from the opposing side of the enclosure. As the bolt is tightened, it pulls the nut into the enclosure. Fig. 5c) illustrates this insertion procedure. Alternatively, the nuts can be press-fit using a vice or clamp. When using press-fitting, caution should be taken to avoid breaking the enclosure due to misalignment of the nut.

7. Attach the soldered ACC to the bottom enclosure.

- Fig. 6a) presents the correct orientation of the ACC in respect to the enclosure.

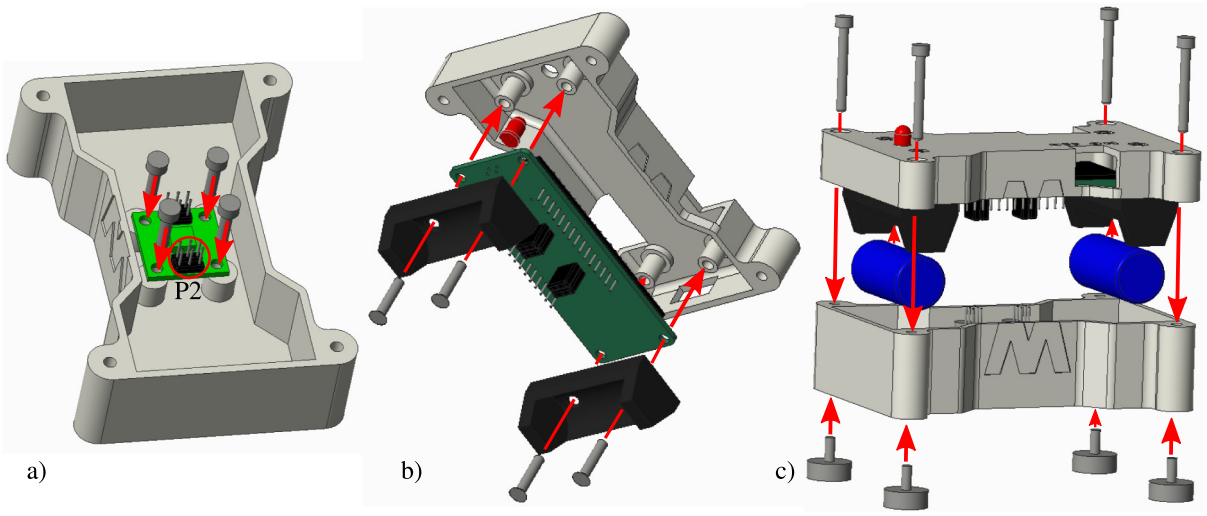

Fig. 6. Final assembly: a) Mount the ACC. b) Mount the PCB and battery holder/holders. c) Mounting the battery/ batteries and magnets, as well as closing the enclosure.

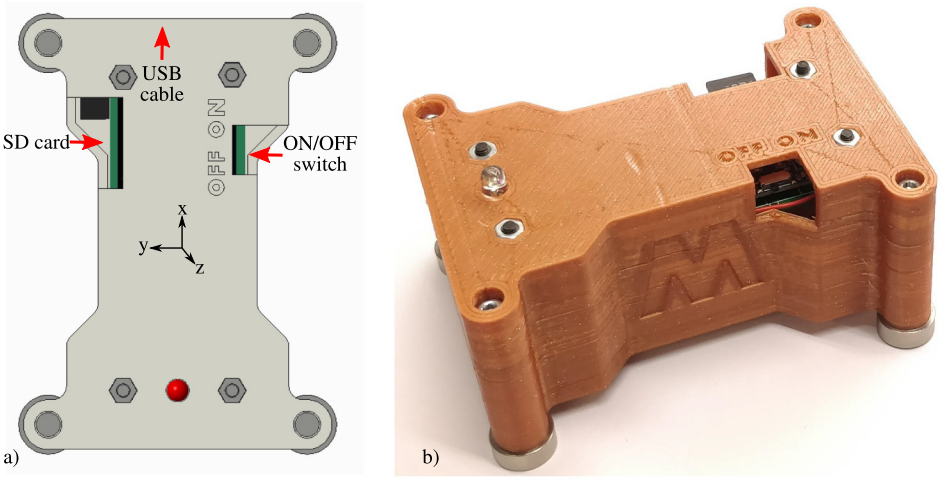

Fig. 7. Finished assembly of Memsio. a) Slots for SD card and USB cable, ON/OFF switch and orientation of the sensor axis. b) Photo of Memsio built and used for the validation measurements.

8. Connect the wires from battery holder/holders to the MCU.
9. Attach the assembled PCB and battery holder/holders to the top enclosure (Fig. 6b)).

- The enclosure is designed to fit up to two 16340 Li-Ion battery cells. Only one cell can be used at a time.
- For correct spacing of the LED in relation to the top half of the enclosure, insert it into the top half before soldering it on the PCB.

10. Mount the battery/batteries onto the holders, then fasten the top and bottom part of the enclosure (Fig. 6c)).

- Ensure that the top and bottom sections of the engraved letter M align with each other when mounting the two halves of the enclosure.

11. Thread the magnets onto the bottom enclosure (Fig. 6c)).

- Fig. 7 shows the SD card slot as well as the location of the USB connector and ON/OFF switch.

5.2. Flashing the firmware

Tools required

- Raspberry Pi single board computer, alternatively a computer with a Debian-based distribution installed.
- Micro USB cable

These instructions are based on using the Raspberry Pi (RPi) single board computer (SBC) to flash the firmware.

1. Install the Raspbian distribution.[8]
2. Log in and connect to the Internet.
3. Open a terminal window and update the package repository and upgrade to the latest software using the command: `sudo apt-get upgrade && sudo apt-get update`
4. Install python pip[9] using: `sudo apt-get install python-pip`
5. Install platformio with `sudo pip install platformio`
6. Download the firmware files from OSF and unzip them. `wget -O memsio_fw.zip https://osf.io/qjfzp/download unzip memsio_fw.zip`
7. Change to the firmware folder and install the required library `cd memsio-fw-v1.0 pio lib install 306`
8. Upload the filesystem and firmware to the ESP32 using make: `make uploadfs && make upload`

[8] https://www.raspberrypi.org/documentation/installation/installing-images/README.md.
[9] https://pypi.org/project/pip/

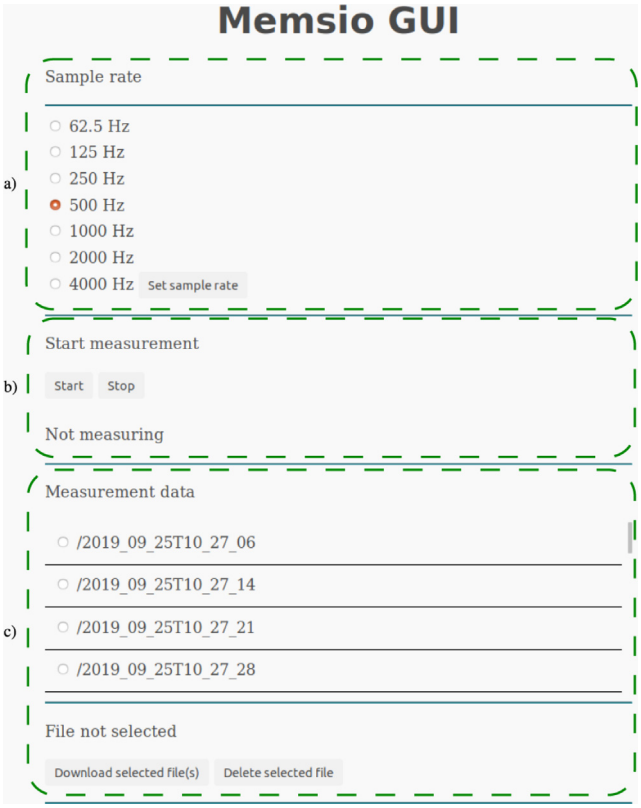

Fig. 8. Memsio GUI. a) set sample rate. b) start and stop measurements. c) download and delete files.

6. Operation instructions

First, ensure that the ESP32 is connected to the wireless network set in the firmware (main.ccp file). During the boot, Memsio will print its IP address onto a serial monitor. Therefore, during the initial setup, it is necessary to connect to Memsio via USB. Once Memsio has successfully connected to a wireless network, the dynamic host configuration protocol will typically allocate the same internet protocol (IP) address to it. A static IP may be allocated to the ESP32 using the application programming interface provided by Espressif.

Memsio is operated from the browser-based GUI, which can be accessed by connecting a computer, smartphone, or tablet to the same wireless network as Memsio and typing the IP address into the browser. Fig. 8 presents the GUI and its three segments, which are marked as a), b), and c). Segment a) allows the user to select which sampling rate will be used. Possible options are 62.5, 125, 250, 500, 1000, 2000, or 4000 Hz. The next segment, Segment b), has two buttons, "Start" and "Stop" buttons. The "Start" button will start a measurement and the "Stop" button will stop the measurement. The measurement files will be printed to Segment c), and the files can be downloaded or deleted with the buttons below.

Each measurement is saved to a separate binary file which is named according to the time and date the measurement began. For example, 2019_9_28T14_56_34 (year_month_dateThour_minutes_seconds) could be one of the filenames. When the files are downloaded from the GUI, they need to be changed to an ASCII format. This can be carried out by using the "binary_to_ascii.py" from the "python_scripts" folder, which can be downloaded from the OSF. It will read the binary file changing it into a comma separated values (CSV) file.

The timestamp and sample number stored to the measurement file can be used to distinguish nonlinear sampling rate or data loss during the measurement or data storing. The timestamp indicates the length of time that has lapsed since the last measurement. For example, if the sampling rate is 1000 Hz, the time between samples is approximately 1000 μ s. If the time between samples varies substantially, it suggests that the sample rate is not steady and there may be problems with the measurement. With a higher sampling rate, there is a higher probability of such erroneous readings since there is less execution time available for other processes,e.g., checking the wireless connection or processing websocket messages. If other processes are running and the sample needs to be read from the sensor, the measurement data acquisition may become nondeterministic. The data loss might not be visible in the timestamp data as they are calculated before storing the data to the buffer. Consequently, the sample index data is stored to facilitate checking the measurements. If the sample index does not increment linearly by one, it implies data loss.

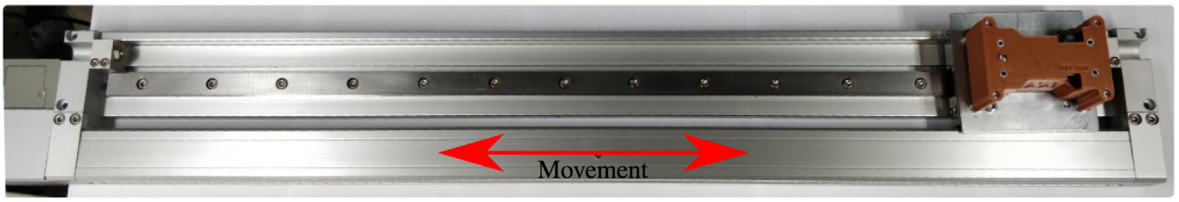

Fig. 9. Photograph of linear actuator measurement configuration.

The LED signals when a command is received and the state of Memsio. When the LED signals blink blue, this indicates that Memsio is ready for a measurement, one green blink that a message has been received from the GUI, and a steady red light that Memsio is measuring.

Due to the portability and wireless functionality of Memsio, it can be placed on a wide variety of DUTs. The magnetic attachments enable facile securing onto a ferromagnetic surface. Other types of attachments can be designed using the M3 threads on Memsio.

7. Validation and characterization

Memsio was validated and characterized by measuring two types of motion: translational and vibration. The translational measurements were carried out with a linear actuator, whereas the oscillating measurements were conducted by measuring the vibration from the bearing housing of a large rotor. In addition, the power consumption was measured.

7.1. Translational motion measurements

The movement of a linear actuator (SMC, Model E-MY2H25-700DBPA-XB11) was measured using Memsio. The linear actuator has programmable acceleration and velocity. Memsio was mounted on the linear actuator with magnets as shown in Fig. 9.

The linear actuator tests were conducted with two different acceleration settings. All tests were conducted at a maximum speed of 0.9 m/s. All translational motion measurements were conducted with a sampling rate of 1000 Hz. Each measurement was repeated a minimum of 10 times. The data was low pass filtered in software with a cutoff frequency of 10 Hz. All the data analysis was performed using python and the scipy library.[10]

Figs. 10 and 11 show the motion data from test runs conducted with 1.47 m/s^2 and 3.92 m/s^2, respectively. The figures show the low pass filtered acceleration (a) data as well as the velocity and distance (b and c). The velocity and distance are computed from the acceleration data by cumulative integration using the composite trapezoidal rule. The composite trapezoidal rule calculates the integral of the curve by dividing the area below the curve into several trapezoids and adding up the sum of their area together. The measurement data is shown in blue, and the reference signal derived from values reported in the linear actuator data sheet is shown in a dashed red color.

As evidenced by Figs. 10 and 11, the motion accurately corresponds to the reference curve. Some oscillation can be observed from Figs. 10a) and 11b). However, the authors consider it likely that this oscillation is caused by the mechanical behavior of the linear actuator rather than inaccuracies or noise in the acceleration measurements. The accuracy and precision of Memsio on translational movement was evaluated from the distance computed by twofold cumulative integration from each of the 10 parallel measurements. The relative error compared to the linear actuator nominal movement range of 700 mm was compared to the measurements and relative error was 0.8% $\pm$ 0.08% and 0.71% $\pm$ 0.16% with an acceleration of 3.92 m/s^2 and 1.47 m/s^2, respectively. The results indicate that Memsio has high accuracy and precision when measuring translational movement.

7.2. Vibration measurements

The vibrations of a rotating paper machine roll (ARotor Laboratory, Aalto University) were measured using Memsio. The paper machine roll was rotated in a computer numerical controlled (CNC) roll grinding machine, which enables dynamic testing of roll and bearing behavior at different rotational speeds. The vibrations at the tending side bearing housing were measured at two different rotational speeds, 600 rpm and 800 rpm. The roll vibration test setup is shown in Fig. 12b.

In order to provide a reference signal, a single axis Integrated Electronics Piezo-Electric (IEPE) accelerometer was mounted next to Memsio. The IEPE accelerometer (Hansford Sensors, H1005005001) has a measurement range of $\pm$16 g, sensitivity of 500 mV/g and a maximum sampling rate of 18 kHz. The data acquisition of the IEPE accelerometers was conducted with a National instruments' data acquisition unit (cDAQ-9174) as well as a sound and vibration input module (NI-9324). Both sensors were mounted onto the same bearing housing, Memsio at the top and the IEPE accelerometer on the side.

[10] https://www.scipy.org/.

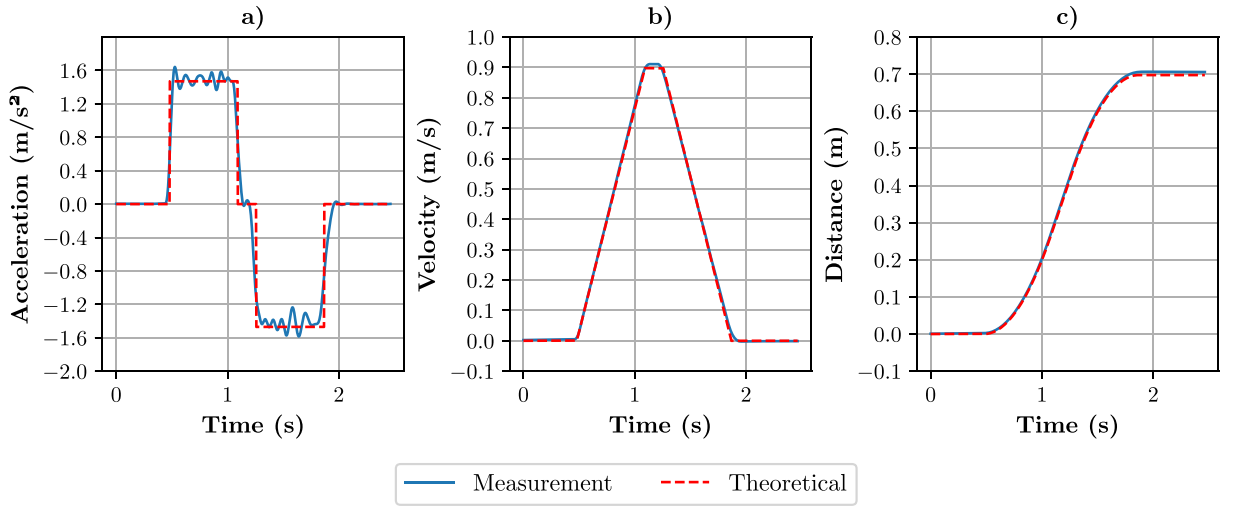

Fig. 10. Linear motor acceleration measurements with 1.47 m/s^2 acceleration and maximum speed of 0.9 m/s. a) acceleration, b) velocity, and c) distance. b) and c) are integrated from acceleration data. The blue line is the actual measured data, and the dashed read line is the calculated ideal acceleration curve.

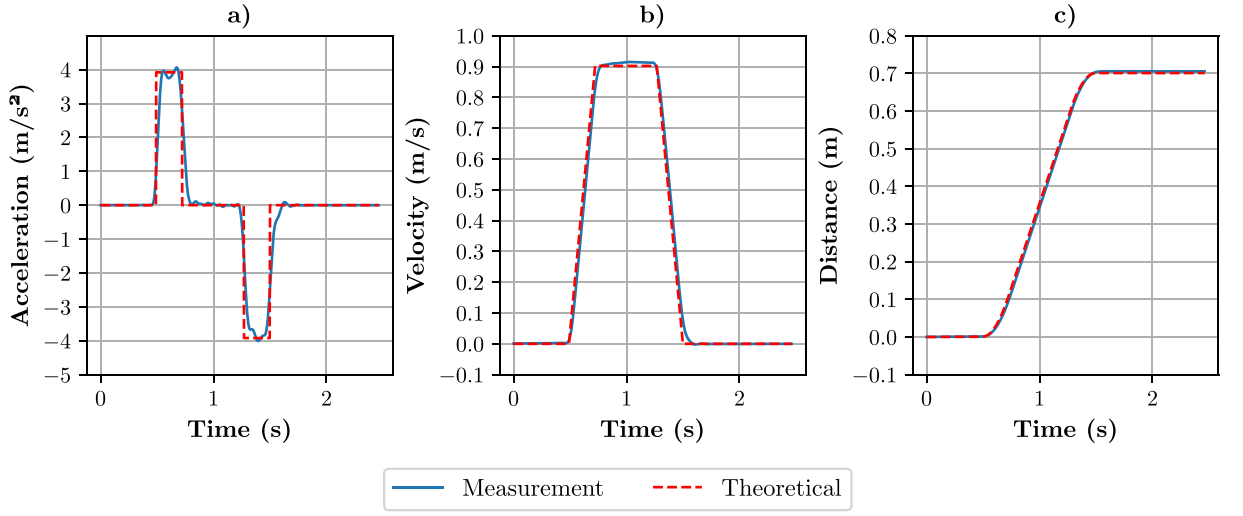

Fig. 11. Linear motor acceleration measurements with 3.92 m/s^2 acceleration and maximum speed of 0.9 m/s. a) acceleration, b) velocity, and c) distance. b) and c) are integrated from acceleration data. The blue line is the actual measured data, and the dashed red line is the calculated ideal acceleration curve.

The sensor locations are shown in Fig. 12a. Vibration data was captured for approximately 18 s with a sampling rates of 4000 Hz and 4266 Hz for Memsio and the IEPE sensor, respectively. The measured data was horizontal vibration, which denotes that the x axis data was analyzed from the Memsio's data.

Table 5 and Fig. 13 present the results of the measurement with 800 rpm; Table 6 and Fig. 14 display the results with 600 rpm. Figs. 13 and 14 present the measured data in a frequency domain. The transform from a time domain to the frequency domain was performed using a Fast Fourier transform (FFT). The data was windowed with a flat top window for the amplitude comparison between the sensors. Frequency comparison was carried out without the windowing.

As demonstrated by the data in Tables 5 and 6 as well as Figs. 13 and 14, the output of Memsio accurately corresponds with the output of the IEPE sensor. Despite some discrepancies in amplitudes and frequencies, the harmonic frequency peaks can be easily distinguished from the FFT data. These results indicate that Memsio functions adequately in detecting vibrations of large rotating machinery.

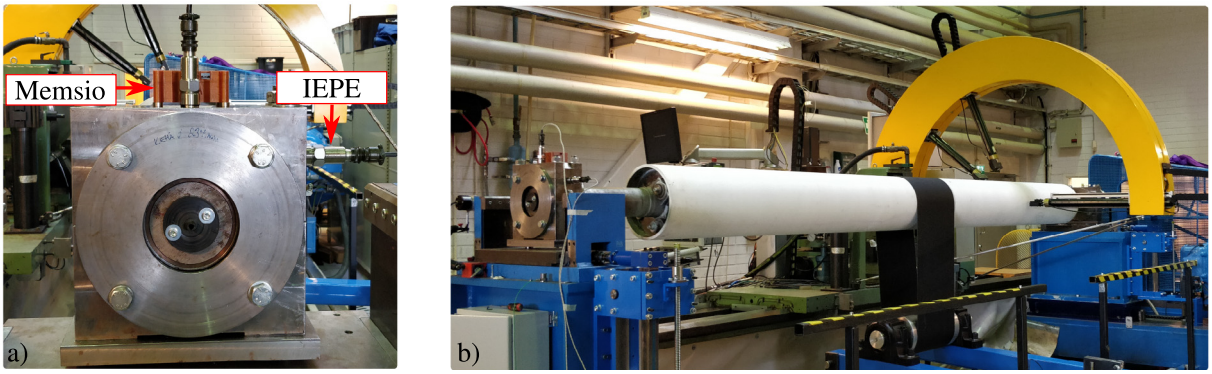

Fig. 12. The roll test setup. a) Memsio and IEPE accelerometer on the tending side of the bearing housing. b) The paper machine roll mounted to a CNC grinding machine. The sensors are on the left-hand side bearing housing.

Table 5
Frequencies and amplituders of the roll measurement at 800 rpm. The peaks are marked in Fig. 13.

Peak	Frequency (Hz)		Amplitude (m/s^2)	
	Memsio	IEPE	Memsio	IEPE
1	13.5	13.5	0.040	0.043
2	17.6	17.5	0.015	0.016
3	22.0	22.0	0.030	0.024
4	23.5	23.6	0.045	0.043
5	35.3	35.3	0.149	0.158

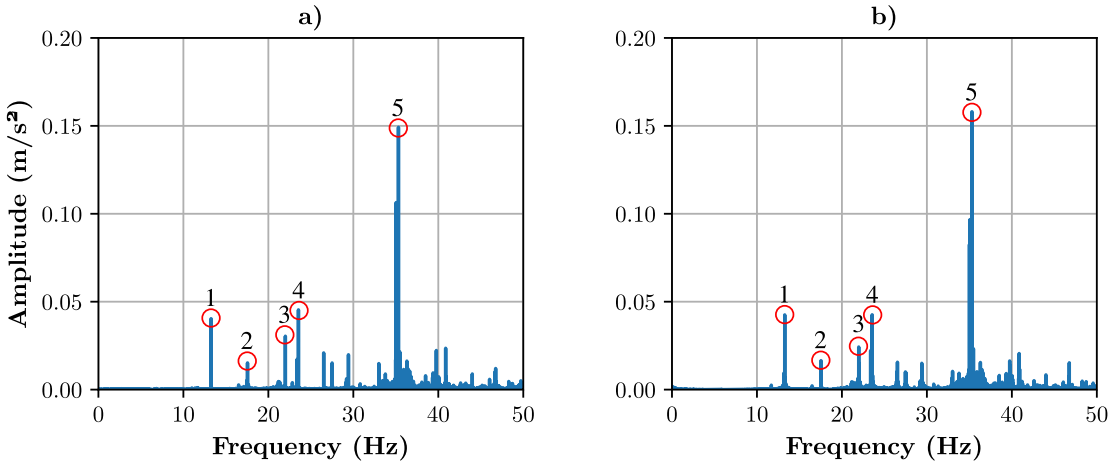

Fig. 13. FFT plots of the paper roll vibration test with 800 rpm. Peaks marked in the figure are presented in Table 5. a) Plot from Memsio's data. b) Plot from IEPE accelerometer data.

Table 6
Frequencies and amplitudes of the roll measurement at 600 rpm. The peaks are marked in Fig. 14.

Peak	Frequency (Hz)		Amplitude (m/s^2)	
	Memsio	IEPE	Memsio	IEPE
1	10.0	9.9	0.020	0.024
2	20.6	20.6	0.029	0.027
3	21.9	21.9	0.054	0.056
4	35.0	35.0	0.138	0.132
5	49.7	49.7	0.072	0.053

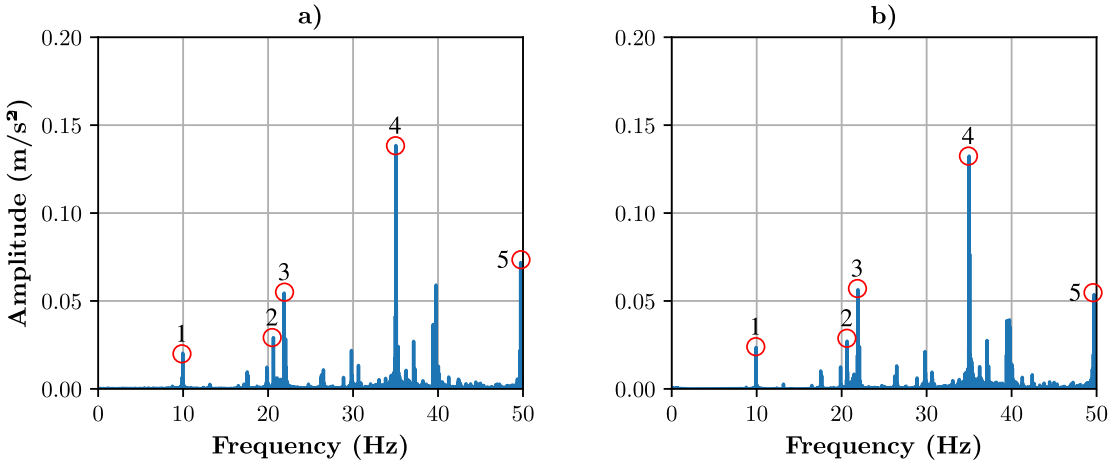

Fig. 14. FFT plots of the paper roll vibration test with 600 rpm. Peaks marked in the figure are presented in Table 6. a) Plot from Memsio's data. b) Plot from IEPE accelerometer data.

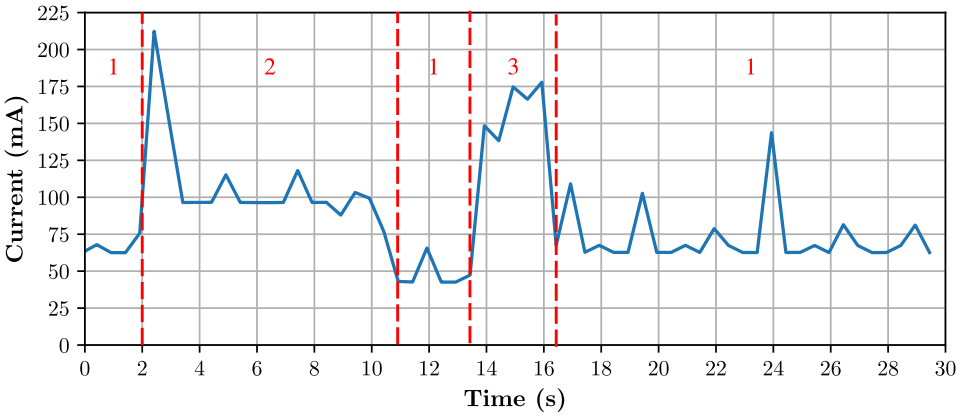

Fig. 15. Measurement cycle with circa 9 s measurement. Different states are separated with dashed red lines. Three different states are presented in the plot: idle (sections marked with 1), measure (section marked with 2), and download (section marked with 3). The measurement was conducted with 4000 Hz sampling rate.

7.3. Current consumption and battery life

The current consumption was measured with a test cycle presented in Fig. 15. The measurement was conducted from the battery with an Agilent multimeter (model 34410A). Different states of the test cycle are marked on Fig. 15. Idle states are marked as 1, the measure state is marked as 2, and the download state is marked as 3. The measurement state of the test cycle presented in Fig. 15 lasted approximately 9 s and was conducted with a 4000 Hz sampling rate. Table 7 presents

Table 7
Average current consumptions in different states presented in Fig. 15.

State	Current consumption (mA)
Idle	50–80
Measure	90–110
Download	150–180

the average current consumptions in different states of the measurement cycle. The high peak at the beginning of the measure state is caused by the "start measurement" command sent with the websocket from the GUI.

The measurement cycle presented in Fig. 15 could be repeated up to 960 times before the 16340 700 mAh battery would need to be recharged. With this estimation, the run time of Memsio would be approximately 8 h.

8. Conclusion

In this research, a novel IoT accelerometer called Memsio was introduced. Memsio utilizes a compact and low-powered MEMS accelerometer and microcontroller with wireless connectivity. Memsio has a large embedded memory in which data is stored during measurements, thus enabling longer measurement without the risk of losing data with wireless data transfer. It has a web browser-based GUI in which settings can be applied and measurements controlled.

The sensor was validated with two types of measurements: lateral movement and vibration. Results indicated that Memsio can accurately and reliably measure in these scenarios. The run time of Memsio is approximately one work day depending on the length of time used to carry out the measurements as well the frequency of these measurements. The battery life and easy mounting of Memsio indicates that it can be utilized for temporary measurements or so called "trouble-shooting" measurements. For example, Memsio could be used for a paper machine with a vibrations problems to localise the source of the vibration.

With IoT devices, it is vital that the battery consumption is suitable for the use case. Memsio was designed for a laboratory environment or temporary measurement needs; however, if more permanent measurements would be required in which measurements are conducted for a several days, months, or even years, the measurement process would need to be better optimized for a longer battery life. The measurement can be optimized by making the measurement process work with cycles. For example, measurements can be conducted once per hour or once per day, and when Memsio is not measuring, it could shift into a deep sleep stage, during which the microcontroller shuts down most of its processes and minimizes power consumption. By utilizing the cyclic measurement with deep sleep, a longer operation time for Memsio is enabled and new possibilities are offered with more permanent measurement needs. This could be the next development step for Memsio.

Human and animal rights

This work does not involve the use of any human or animal subjects.

Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported by Academy of Finland (Digital Twin of Rotor System, under Grant 313675) and Business Finland (Reboot IoT Factory, under Grant 4356/31/2019).

References

[1] R. Bogue, Recent developments in MEMS sensors: a review of applications, markets and technologies, Sensor Rev. 33 (4) (2013) 300–304.
[2] S. Kavitha, R. Joseph Daniel, K. Sumangala, Design and analysis of MEMS comb drive capacitive accelerometer for SHM and seismic applications, Measurement 93 (2016) 327–339.
[3] M. Medani, A. Benahmed, M. Zidour, H. Heireche, A. Tounsi, A.A. Bousahla, A. Tounsi, S.R. Mahmoud, Static and dynamic behavior of (FG-CNT) reinforced porous sandwich plate, Steel Compos. Struct. 32 (5) (2019) 595–610.
[4] A. Draoui, M. Zidour, A. Tounsi, B. Adim, Static and dynamic behavior of nanotubes-reinforced sandwich plates using (FSDT), J. Nano Res. 57 (2019) 117–135.
[5] B. Karami, D. Shahsavaria, M. Janghorbana, A. Tounsi, Resonance behavior of functionally graded polymer composite nanoplates reinforced with grapheme nanoplatelets, Int. J. Mech. Sci. 156 (2019) 94–105.
[6] Y. Tlidji, M. Zidour, K. Draiche, A. Safa, M. Bourada, A. Tounsi, A.A. Bousahla, S.R. Mahmoud, Vibration analysis of different material distributions of functionally graded microbeam, Struct. Eng. Mech. 69 (6) (2019) 637–649.
[7] H. Berghouti, E.A. Adda Bedia, A. Benkhedda, A. Tounsi, Vibration analysis of nonlocal porous nanobeams made of functionally graded material, Adv. Nano Res. 7 (5) (2019) 351–364.
[8] S. Alimirzaei, M. Mohammadimehr, A. Tounsi, Nonlinear analysis of viscoelastic micro-composite beam with geometrical imperfection using FEM: MSGT electro-magneto-elastic bending, buckling and vibration solutions, Struct. Eng. Mech. 71 (5) (2019) 485–502.
[9] M. Balubaid, A. Tounsi, B. Dakhel, S.R. Mahmoud, Free vibration investigation of FG nanoscale plate using nonlocal two variables integral refined plate theory, Comput. Concr. 24 (6) (2019) 579–586.
[10] S. Boutaleb, K.H. Benrahou, A. Bakora, A. Algarni, A.A. Bousahla, A. Tounsi, A. Tounsi, S.R. Mahmoud, Dynamic analysis of nanosize FG rectangular plates based on simple nonlocal quasi 3D HSDT, Adv. Nano Res. 7 (3) (2019) 189–206.
[11] M. Hussain, M.N. Naeem, A. Tounsi, M. Taj, Nonlocal effect on the vibration of armchair and zigzag SWCNTs with bending rigidity, Adv. Nano Res. 7 (6) (2019) 431–442.
[12] A.Y. Alhaddad, J.J. Cabibihan, A. Hayek, A. Bonarini, A low-cost test rig for impact experiments on a dummy head, HardwareX 6 (2019) e00068.
[13] J. Lund, A. Paris, J. Brock, Mouthguard-based wireless high-bandwidth helmet-mounted inertial measurement system, HardwareX 4 (2018) e00041.

[14] T. Lei, A.A. Mohamed, C. Claudel, An IMU-based traffic and road condition monitoring system, HardwareX 4 (2018) e00045.
[15] I. Koene, R. Viitala, P. Kuosmanen, Internet of things based monitoring of large rotor vibration with a microelectromechanical systems accelerometer, IEEE Access 7 (2019) 92210–92219.
[16] A. Albarbar, S. Mekid, A. Starr, R. Pietruszkiewicz, Suitability of MEMS accelerometers for condition monitoring: an experimental study, Sensors 8 (2008) 784–799.
[17] S. Jiménez, M.O. Cole, P.S. Keogh, Vibration sensing in smart machine rotors using internal MEMS accelerometers, J. Sound Vib. 377 (2016) 58–75.
[18] S. Lu, P. Zhou, X. Wang, Y. Liu, F. Liu, J. Zhao, Condition monitoring and fault diagnosis of motor bearings using undersampled vibration signals from a wireless sensor network, J. Sound Vib. 414 (2018) 81–96.
[19] P. Muhammad, S.A. Devi, Hand gesture user interface for smart devices based on mems sensors, Proc. Comput. Sci. 93 (September) (2016) 940–946.
[20] A. Weiskopf, F. Weichert, E. Dommes, A low cost embedded solution for detection and analysis of three-dimensional gestures through wireless communication, vol. 45, IFAC, 2012..
[21] Z. Chen, W. Ren, Y. Ren, K.K.R. Choo, LiReK: a lightweight and real-time key establishment scheme for wearable embedded devices by gestures or motions, Future Gen. Comput. Syst. 84 (2018) 126–138.
[22] R. Simon Carbajo, E. Simon Carbajo, B. Basu, C. Mc Goldrick, Routing in wireless sensor networks for wind turbine monitoring, Pervasive Mobile Comput. 39 (2017) 1–35.
[23] Q. Huang, B. Tang, L. Deng, J. Wang, A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks, Measurement: J. Int. Meas. Confeder. 67 (2015) 51–60.
[24] M.E. Elnady, J.K. Sinha, S.O. Oyadiji, Identification of critical speeds of rotating machines using on-shaft wireless vibration measurement, J. Phys: Conf. Ser. 364 (2012) 012142.
[25] J.P. Lynch, A. Partridge, K.H. Law, T.W. Kenny, A.S. Kiremidjian, E. Carryer, Design of Piezoresistive MEMS-based accelerometer for integration with wireless sensing unit for structural monitoring, J. Aerospace Eng. 16 (3) (2003) 108–114.
[26] G. Kilic, M.S. Unluturk, Testing of wind turbine towers using wireless sensor network and accelerometer, Renew. Energy 75 (2015) 318–325.
[27] M.E. Elnady, A. Abdelbary, J.K. Sinha, S.O. Oyadiji, FE and experimental modeling of on-shaft vibration measurement, in: Proceedings of the 15th International Conference on Aerospace Sciences & Aviation Technology, No. May, 2013, pp. 1–18..
[28] G. Feng, N. Hu, Z. Mones, F. Gu, A. Ball, An investigation of the orthogonal outputs from an on-rotor MEMS accelerometer for reciprocating compressor condition monitoring, Mech. Syst. Signal Process. 76–77 (2016) 228–241.
[29] J.M. Pearce, Building research equipment with free, open-source hardware, Science 337 (6100) (2012) 1303–1304.
[30] J.M. Pearce, Return on investment for open source scientific hardware development, Sci. Public Policy 43 (2) (2015) 192–195.
[31] T. Kodera, Adaptive antenna system by ESP32-PICO-D4 and its application to web radio system, HardwareX 3 (March) (2018) 91–99.
[32] C. Watson, S. Senyo, All-in-one automated microfluidics control system, HardwareX 5 (2019) e00063.
[33] V. Klar, J.M. Pearce, P. Kärki, P. Kuosmanen, Ystruder: open source multifunction extruder with sensing and monitoring capabilities, HardwareX 3 (2019) e00080.