

AN15229

Gesture Classification and Vibration Monitoring Demo for FRDM-MCXE32B

Rev. 1.0 — 27 October 2026

Application note

Document information

Information	Content
Keywords	AN15229, MCX E32B, FRDM-MCXE32B, predictive maintenance, anti-tamper, tilt detection, accelerometer, FXLS8974CF, dual-core, Cortex-M7
Abstract	This application note describes a gesture classification and vibration monitoring demo for the FRDM-MCXE32B development board. The demo uses the onboard FXLS8974CF three-axis accelerometer and the dual-core MCX E32B MCU to demonstrate industrial anti-tamper and tilt detection, as well as predictive maintenance through vibration monitoring.



1 Introduction

The Gesture Classification and Vibration Monitoring demo showcases how the FRDM-MCXE32B can address two common industrial use cases: anti-tamper and tilt detection, and predictive maintenance through vibration monitoring. Using the onboard FXLS8974CF three-axis accelerometer, the application detects motion events and analyzes vibration behavior to identify abnormal operating conditions.

The dual Arm Cortex-M7 architecture of MCX E32B enables sensing, signal processing, and application control tasks to run concurrently, allowing real-time event detection and condition monitoring on a single device. This demonstrates how dual-core microcontrollers can efficiently implement industrial monitoring solutions at the edge.

For anti-tamper and tilt-detection applications, the demo detects orientation changes, unexpected movement, and free-fall events. For predictive-maintenance applications, it establishes a baseline vibration profile and identifies deviations that may indicate developing equipment faults.

This application note describes the system architecture, hardware and software setup, application workflow, and the signal-processing techniques used to implement these monitoring functions on the MCX E32B.

2 Demo overview

This section describes the overall software architecture of the Gesture Classification and Vibration Monitoring demo and the distribution of the application workload across the MCX E32B dual-core architecture.

The MCX E32B integrates two Arm Cortex-M7 cores. In this application, core 0 operates as the primary core and core 1 operates as the secondary core. The application is implemented as separate primary-core and secondary-core projects, with the processing workload divided between the two cores.

Core 1 is responsible for data acquisition and signal processing. It samples the onboard FXLS8974CF three-axis accelerometer, extracts motion features, and performs Fast Fourier Transform (FFT) processing using the CMSIS-DSP library. Core 0 executes the gesture-classification and vibration-monitoring algorithms and manages the application outputs, including the RGB LED, serial console, and optional LCD-PAR-S035 dashboard.

Communication between the two cores is implemented using RMPMsg-Lite over shared memory. Core 1 transfers the processed motion and vibration data to core 0, where the data is evaluated by the gesture-classification and vibration-monitoring algorithms.

The gesture-classification algorithm running on core 0 classifies the received motion data into five states: Idle, Tilt Left, Tilt Right, Motion, and Drop. The Motion state includes events such as tapping, shaking, and dragging, while the Drop state indicates a detected free-fall condition. In parallel, the vibration-monitoring algorithm reports operating conditions such as Learning, Normal, WARN, or ALARM.

[Figure 1](#) illustrates the software architecture, the distribution of processing tasks, and the interaction between the two cores.

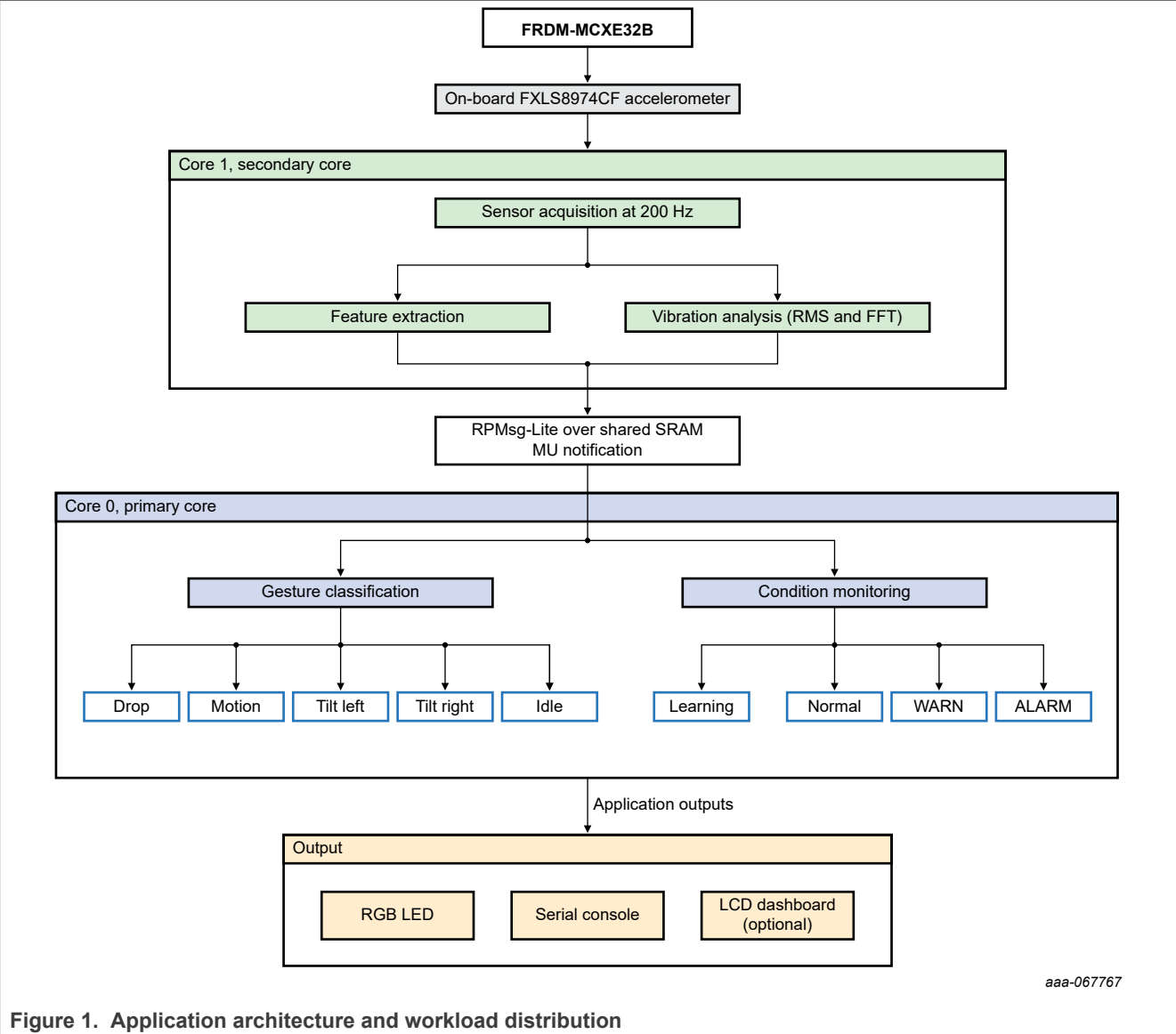


Figure 1. Application architecture and workload distribution

3 MCX E32B dual-core architecture

MCX E32B integrates two Arm Cortex-M7 processor cores and provides memory, system-interconnect, and interprocessor communication resources for multicore operation. This section provides an overview of the processor cores, memory architecture, and hardware communication resources.

3.1 Cortex-M7 cores

MCX E32B includes two Arm Cortex-M7 processor cores, each capable of operating at frequencies up to 160 MHz. Each core includes a single-precision Floating Point Unit (FPU), instruction and data caches, and instruction and data tightly coupled memory resources.

After initialization, the two cores can execute concurrently and access shared device resources through the system interconnect. This architecture supports the distribution of processing tasks across the cores in multicore software implementations.

3.2 Memory architecture

MCX E32B provides on-chip flash memory and SRAM connected through the system interconnect. The system interconnect allows the processor cores and other bus initiators to access the applicable memory and peripheral resources.

Each Cortex-M7 core also includes instruction and data cache support and interfaces to instruction and data tightly coupled memory. The tightly coupled memory resources provide low-latency access to instructions and data.

3.3 Inter-core communication

MCX E32B provides hardware resources for communication and synchronization between the two processor cores. Shared memory can be used to exchange data between the cores, while the Messaging Unit (MU) provides hardware mechanisms for interprocessor communication and signaling.

When data is exchanged through cacheable shared-memory regions, software must perform the required cache-maintenance operations to ensure that data written by one core is visible to the other core.

4 Demo setup

This section provides an overview of the demo setup.

4.1 Hardware requirement

The following hardware is required to run the demo:

- FRDM-MCXE32B board.
- USB Type-C cable for board power and debug.
- Host computer with a serial terminal configured for 115,200 baud.
- LCD-PAR-S035 display panel (optional).

The demo uses the on-board FXLS8974CF three-axis accelerometer and RGB LED, so no external sensor connections are required. The LCD-PAR-S035 provides a graphical dashboard when connected to header J8. The application can also be built and operated in console-only mode without the display.

4.2 Software prerequisites

The software prerequisites are as follows:

- Visual Studio Code with the MCUXpresso for VS Code extension (for more information, refer to [Import Repository — MCUXpresso for VS Code 26.09 documentation](#) and the Getting Started Page for FRDM-MCXE32B).
- MCUXpresso SDK with FRDM-MCXE32B support (version 26.09.00 or later).
- Arm GNU (armgcc) toolchain registered with the extension.
- Demo source code from GitHub repo: <https://github.com/nxp-appcodehub/dm-gesture-and-vibration-demo-for-mcxe32b>

4.3 Import, build, and program

The downloaded demo code is a dual-core application. It is organized as two sub-projects, primary (core 0) and secondary (core 1), built together with the MCUXpresso SDK `sysbuild` flow. The `sysbuild` automatically builds the secondary application, converts it to a binary image, and embeds that binary inside the primary image, which releases the secondary core at startup.

- 1. In the **Repositories** view of the MCUXpresso extension, make sure that the FRDM-MCXE32B MCUXpresso SDK (version 26.09.00) is installed and registered. The SDK may be installed in any folder.
- 2. In the **Projects** view or from the Quick Start Panel, choose **Import Project** (local) and select the TOP folder (`dm-gesture-and-vibration-demo-for-mcxe32b`) of the project, not the primary subfolder. The board (`frdm-mcxe32b`), core (`core0`), toolchain (`armgcc`), and `sysbuild` settings are read from `.vscode/mcuxpresso-tools.json`.
- 3. If prompted, associate the project with your registered MCX E32B SDK.
- 4. Build the project by right-clicking on the top project name and selecting **Build Project**. The `sysbuild` builds the secondary core first and embeds its `core1_image.bin` into the primary image.
- 5. Click **Debug** to program the primary image to the board and run it. The programmed image includes the embedded secondary-core application.
- 6. Alternatively, to flash without starting a debug session, right-click the project and choose **Flash the Selected Target**, then select the core0 ELF image. Once the programming completes, press the reset button (**SW1**) to start the application.

5 Demo operation

This section provides an overview of the demo operation.

5.1 Starting the application

Connect the FRDM-MCXE32B to the host computer using the USB cable and open a serial terminal configured for 115,200 baud. Reset the board to start the application. The primary core starts the secondary core during the application initialization. After the inter-core communication handshake is complete, the secondary core initializes the onboard accelerometer and begins sensor acquisition.

5.2 Demo output

This section introduces the demo output.

5.2.1 Gesture output

The primary core evaluates the gesture-classification logic for each received feature set and updates the RGB LED according to the detected gesture. The serial console reports an event only when the gesture class changes, avoiding continuous per-sample output.

Table 1. Gesture output

Class	Description	RGB LED
Idle	Board is still and roughly level	Off
Tilt Left	Sustained lean with the left side raised	Blue
Tilt Right	Sustained lean with the right side raised	Blue
Motion	Tap, shake, or drag	Red
Drop	Free-fall	Green

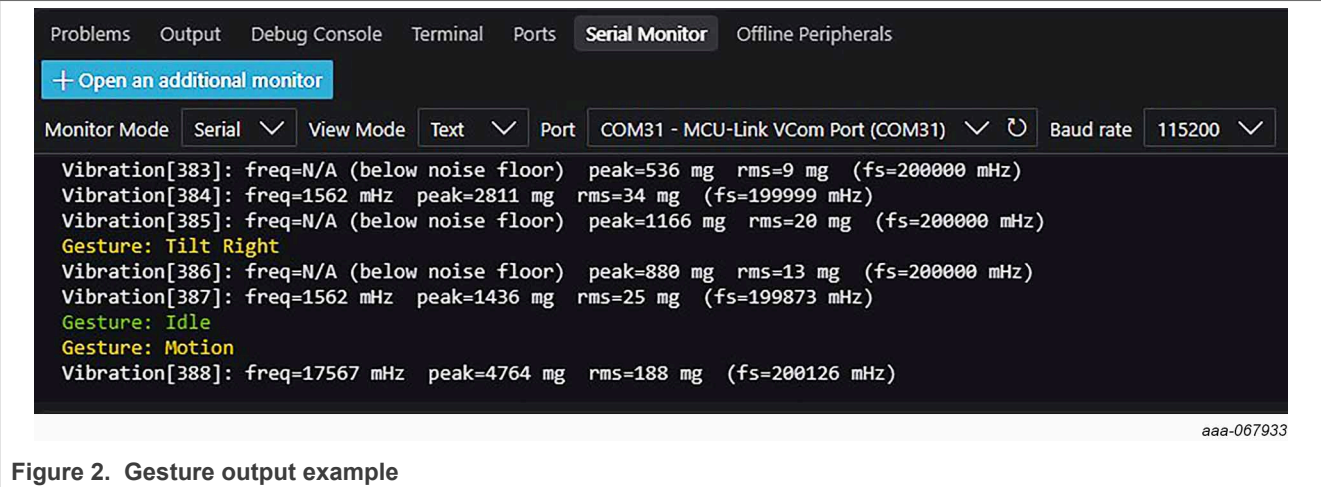


Figure 2. Gesture output example

5.2.2 Vibration condition states

For each completed FFT analysis block, the primary core evaluates the broadband Root Mean Square (RMS) value reported by the secondary core and updates the vibration condition:

- Learning: The application is establishing the vibration baseline.
- Normal: The measured vibration remains below the configured warning threshold.
- WARN: The measured vibration exceeds the configured warning threshold.
- ALARM: The measured vibration exceeds the configured alarm threshold.

The serial console reports an event when the vibration condition changes.

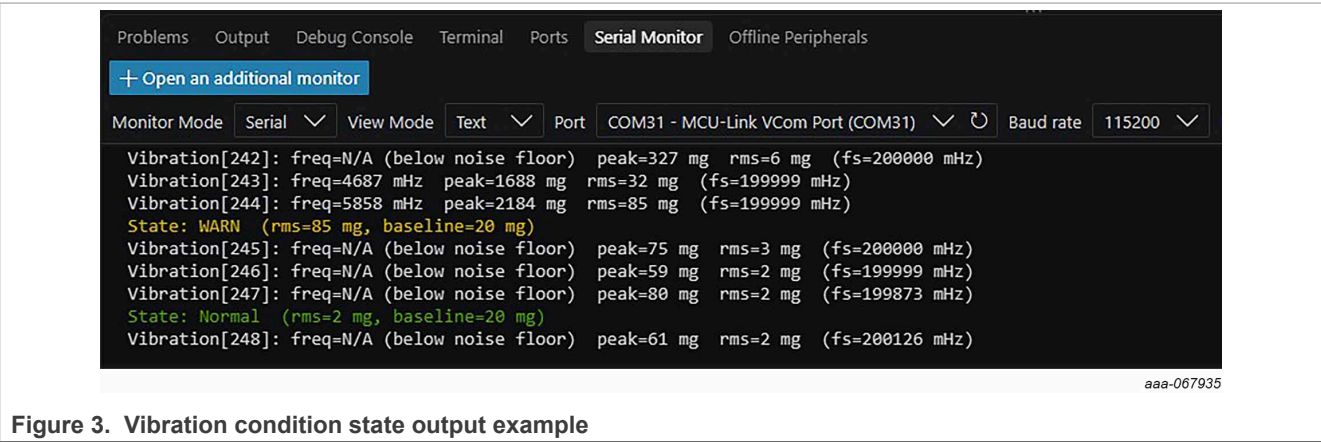


Figure 3. Vibration condition state output example

5.2.3 LCD dashboard

If an LCD-PAR-S035 panel is fitted, the dashboard displays the current gesture, vibration condition and status indicator, RMS bar, dominant-frequency readout, and a real-time vibration chart. The serial console and dashboard use the same application data.



Figure 4. LCD output example

The following build-time switches configure the application output:

- BOARD_LCD_S035: Enables the LCD dashboard. When disabled, the application can be built without the display dependency.
- APP_CONSOLE_COLOR: Enables ANSI color coding for serial-console event messages. Disable this option for terminals that do not support ANSI escape sequences.

6 Application design

This section describes how the application processes accelerometer data to classify motion events and monitor vibration conditions. The processing stages include sensor acquisition and feature extraction, gesture classification, FFT-based vibration analysis, condition monitoring, and dominant-frequency stabilization.

[Figure 5](#) shows the processing flow used to derive gesture and vibration information from the accelerometer data. The figure identifies the processing stages performed by core 1, the secondary core, and core 0, the primary core.

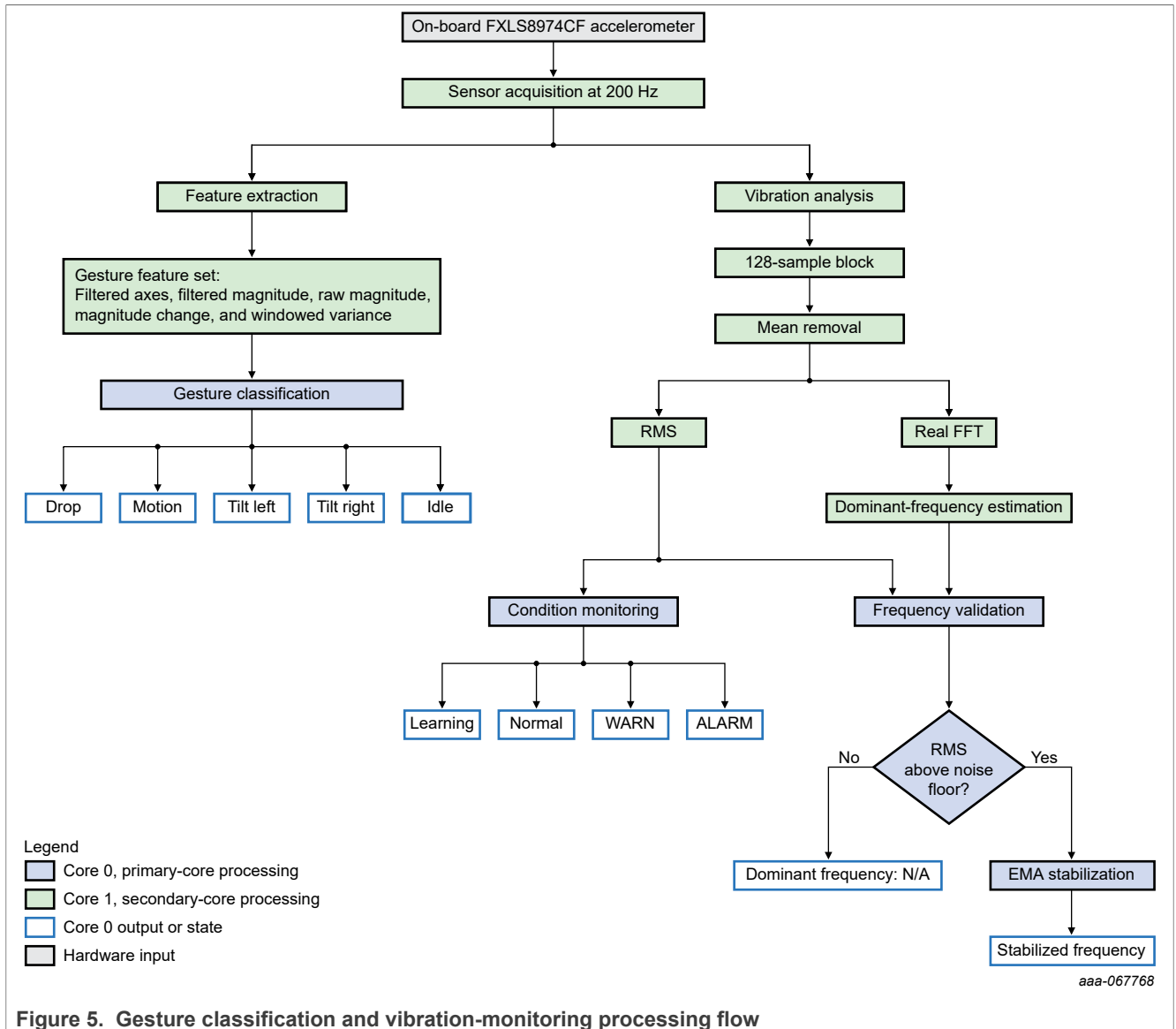


Figure 5. Gesture classification and vibration-monitoring processing flow

6.1 Sensor acquisition and feature extraction

Core 1 samples the onboard FXLS8974CF three-axis accelerometer at a nominal rate of 200 Hz. Sampling is paced using the Cortex-M7 cycle counter. Core 0 manages the system clock and peripheral initialization, while core 1 uses its core-local cycle counter to schedule sensor acquisition.

The accelerometer measures acceleration along three orthogonal axes, X, Y, and Z, in units of g, where 1 g represents the acceleration due to the Earth's gravity. The application combines the three axis measurements into a single acceleration magnitude as follows:

$$magnitude = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

Core 1 derives the following features from the accelerometer samples:

- **Low-pass-filtered axes:** An Exponential Moving Average (EMA) is applied independently to the X, Y, and Z acceleration values. The filtered values provide a stable estimate of orientation for tilt classification.

- **Filtered magnitude:** Calculated from the filtered axis values and used to verify that the acceleration magnitude remains near 1 g during tilt detection.
- **Raw magnitude:** Calculated from the unfiltered axis values and used as the low-acceleration feature for free-fall detection.
- **Magnitude change:** The absolute difference between consecutive filtered-magnitude samples. The implementation names this feature jerk; however, it represents a sample-to-sample acceleration-magnitude change rather than physical jerk because it is not divided by the sampling interval.
- **Windowed variance:** Calculated from the filtered magnitude over a sliding sample window and used to indicate sustained motion.

The extracted features are transferred to core 0 for gesture classification. In parallel, raw acceleration-magnitude samples are collected into blocks for vibration analysis.

6.2 Gesture classification

Core 0 classifies each received feature set using a prioritized sequence of decision rules. The rules are evaluated in the order Drop, Motion, Tilt, and Idle, as shown in [Figure 6](#). The classification thresholds are defined as configurable constants in the application source.

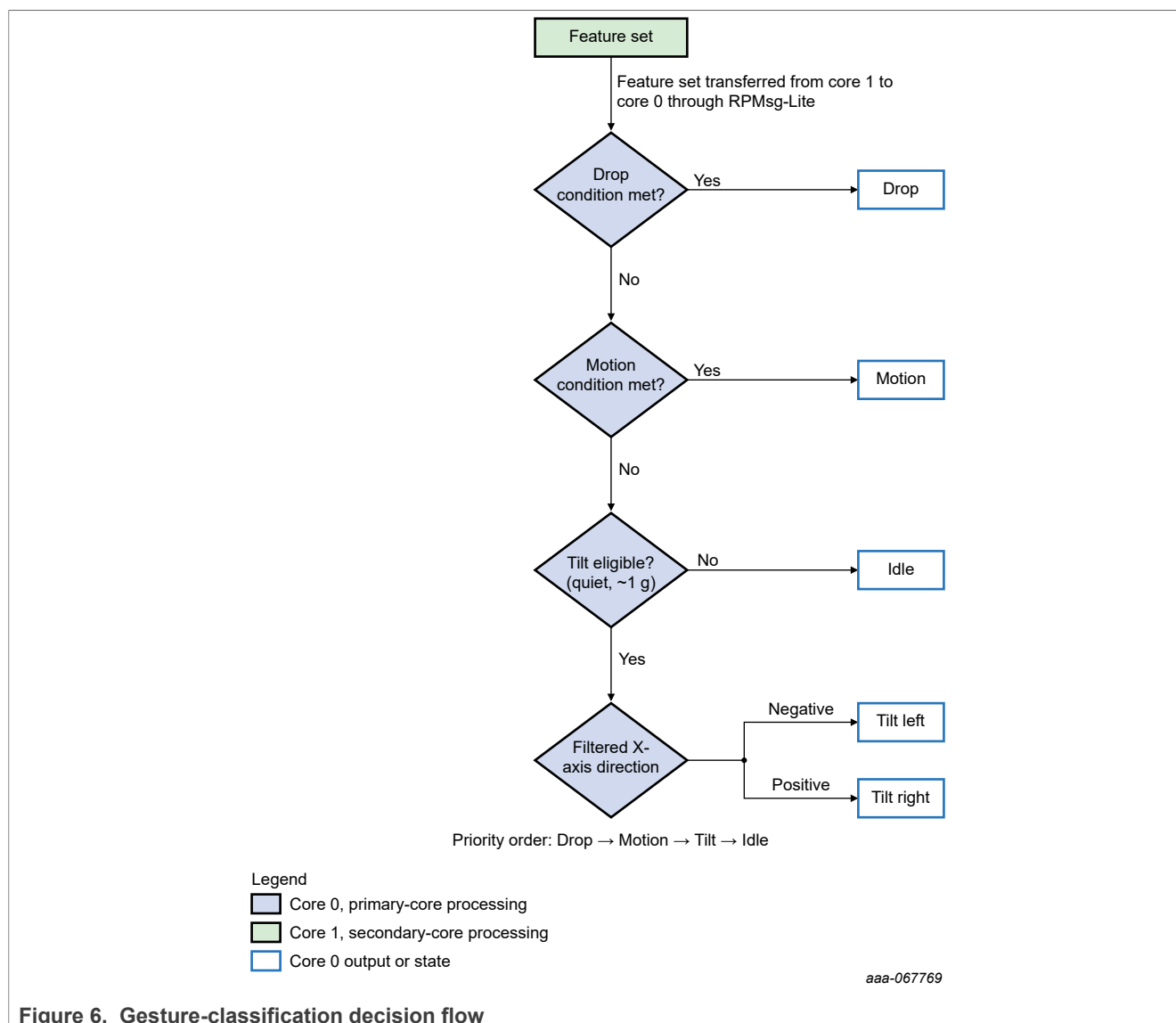


Figure 6. Gesture-classification decision flow

The gesture states are determined as follows:

- **Drop:** During free-fall, the measured specific force approaches zero, causing the raw acceleration magnitude to approach 0 g. A Drop condition is detected when the raw magnitude remains below the configured threshold for a specified number of consecutive samples. After detection, the Drop state is temporarily latched to prevent the landing impact from being immediately classified as Motion. Drop detection is re-armed after the board returns to a stable condition.
- **Motion:** A Motion condition is detected when either the sample-to-sample magnitude change or the windowed variance exceeds the corresponding configured threshold. Tap, shake, and drag events are represented by a single Motion class. The detected state is held for a short interval so that brief motion events remain visible to the user.
- **Tilt:** For a stationary board, tilting redistributes the gravity component among the three axes while the total acceleration magnitude remains close to 1 g. Tilt is evaluated after the measured motion remains below the configured motion thresholds for several samples. The filtered X-axis acceleration then determines whether the board is classified as Tilt Left or Tilt Right.
- **Idle:** The Idle state is reported when none of the preceding conditions is detected.

6.3 FFT vibration analysis

As shown in the vibration-monitoring path in [Figure 5](#), core 1 uses a Fast Fourier Transform (FFT) to analyze the frequency content of the measured vibration. The application processes 128-sample blocks using the CMSIS-DSP `arm_rfft_fast_f32` real FFT function.

At a nominal sampling rate of 200 Hz, each 128-sample block represents approximately 0.64 seconds of data and provides a nominal frequency-bin spacing of:

$$\Delta f = \frac{200}{128} = 1.5625 \text{ Hz} \quad (2)$$

Each block is processed as follows:

1. Collect 128 raw acceleration-magnitude samples.
2. Calculate and subtract the mean value from the samples to remove the DC component.
3. Calculate the RMS of the mean-removed signal to obtain a broadband measure of vibration amplitude.
4. Perform the real FFT.
5. Search the applicable FFT bins, excluding the DC and Nyquist bins, for the largest spectral component.
6. Use the location of the largest spectral component to estimate the dominant vibration frequency.

The reported dominant frequency is an estimate limited by the FFT frequency resolution. Because the application does not apply a window function before the FFT, spectral leakage can distribute energy across adjacent bins when the input frequency does not align with an FFT-bin frequency. Therefore, the reported dominant frequency and spectral peak are relative metrics rather than calibrated vibration measurements.

Core 1 also measures the effective sampling rate using the Cortex-M7 cycle counter. The measured sampling rate is used for the bin-to-frequency conversion so that the reported frequency reflects variations in the actual acquisition rate.

6.4 Condition monitoring

Core 0 evaluates the broadband RMS value calculated for each FFT block to determine the vibration condition. This condition-monitoring method provides a foundation that can be adapted for predictive-maintenance applications.

Mechanical systems can develop faults gradually. For example, worn motor bearings can increase vibration, while a loose mounting component can alter the vibration behavior of a system. The condition-monitoring algorithm compares the measured vibration level with a learned baseline to identify deviations that may indicate abnormal operating behavior. The algorithm can be adapted using application-specific thresholds, frequency ranges, and decision logic for a particular machine or operating environment. In its current form, the demo identifies changes in vibration conditions that may indicate abnormal behavior. The demo does not predict failures or estimate remaining useful life.

The condition-monitoring algorithm operates in two stages:

- **Baseline learning:** During the initial learning period, the application records the lowest measured RMS value, subject to a configured minimum floor, and uses this value as the vibration baseline. Selecting the lowest observed value reduces the effect of temporary handling or movement during startup.
- **Condition evaluation:** After baseline learning is complete, the application compares each RMS measurement with configurable thresholds relative to the learned baseline and assigns a Normal, WARN, or ALARM condition. These relative thresholds are specific to the demonstration and do not represent standardized vibration-severity limits.

The condition state responds quickly to increasing vibration but requires several consecutive lower-vibration blocks before transitioning to a less severe state. This behavior reduces rapid state changes when the measured vibration fluctuates around a threshold.

6.5 Dominant-frequency stabilization

When the RMS is at or below the configured vibration noise floor, the dominant FFT component is typically determined by measurement noise rather than meaningful vibration. Core 0 therefore applies an amplitude gate and reports the dominant frequency as N/A, indicating that a reliable dominant-frequency measurement is not available.

When the RMS value exceeds the noise floor, the dominant-frequency result is filtered using an Exponential Moving Average (EMA) to reduce block-to-block fluctuations in the displayed frequency. The stabilized frequency is then reported through the serial console and, when enabled, displayed on the LCD dashboard. The stabilization logic is implemented using integer arithmetic and is available in both LCD and console-only builds.

7 Conclusion

The Gesture Classification and Vibration Monitoring demo demonstrates how the dual-core architecture of the MCX E32B can be used to implement industrial monitoring applications on a single device. By distributing sensor acquisition, signal processing, and application-control tasks between two Cortex-M7 cores, the demo enables real-time motion-event detection and vibration analysis using the onboard FXLS8974CF three-axis accelerometer.

The demonstrated processing techniques provide a foundation for a range of industrial applications, including anti-tamper and tilt detection and predictive maintenance. Motion-event detection can be used to identify orientation changes, unexpected movement, and free-fall conditions, while the vibration-monitoring workflow can be adapted with application-specific thresholds and decision logic to detect abnormal operating conditions and emerging equipment faults.

8 Acronyms

Table 2 lists the acronyms used in this document.

Table 2. Acronyms

Acronym	Meaning
ANSI	American National Standards Institute
CMSIS	Common Microcontroller Software Interface Standard
DC	Direct Current
DSP	Digital Signal Processing
ELF	Executable and Linkable Format
EMA	Exponential Moving Average
FFT	Fast Fourier Transform
FPU	Floating-Point Unit
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MU	Messaging Unit
RMS	Root Mean Square
SRAM	Static Random Access Memory
USB	Universal Serial Bus

9 Revision history

[Table 3](#) summarizes the revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN15229 v.1.0	27 October 2026	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This evaluation product is intended solely for technically qualified professionals, specifically for use in research and development environments to facilitate evaluation purposes. It is not a finished product, nor is it intended to be a part of a finished product. Any software or software tools provided with an evaluation product are subject to the applicable license terms that accompany such software or software tools.

This evaluation product is provided on an "as is" and "with all faults" basis for evaluation purposes only and is not to be used for product qualification or production. If you choose to use these evaluation products, you do so at your risk and hereby agree to release, defend and indemnify NXP (and all of its affiliates) for any claims or damages resulting from your use. NXP, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this evaluation product remains with user.

In no event shall NXP, its affiliates or their suppliers be liable to user for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out of the use of or inability to use the evaluation product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that user might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP, its affiliates and their suppliers and user's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by user based on reasonable reliance up to the greater of the amount actually paid by user for the evaluation product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose and shall not apply in case of willful misconduct.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Gesture Classification and Vibration Monitoring Demo for FRDM-MCXE32B

Suitability for use in industrial applications (functional safety) — This NXP product has been qualified for use in industrial applications. It has been developed in accordance with IEC 61508, and has been SIL-classified accordingly. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as “Critical Applications”), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys’ fees) that NXP may incur related to customer’s incorporation of any product in a Critical Application.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Tables

Tab. 1.	Gesture output	5	Tab. 3.	Revision history	13
Tab. 2.	Acronyms	12			

Figures

Fig. 1.	Application architecture and workload distribution	3	Fig. 4.	LCD output example	7
Fig. 2.	Gesture output example	6	Fig. 5.	Gesture classification and vibration- monitoring processing flow	8
Fig. 3.	Vibration condition state output example	6	Fig. 6.	Gesture-classification decision flow	10

Contents

1	Introduction	2
2	Demo overview	2
3	MCX E32B dual-core architecture	3
3.1	Cortex-M7 cores	3
3.2	Memory architecture	4
3.3	Inter-core communication	4
4	Demo setup	4
4.1	Hardware requirement	4
4.2	Software prerequisites	4
4.3	Import, build, and program	4
5	Demo operation	5
5.1	Starting the application	5
5.2	Demo output	5
5.2.1	Gesture output	5
5.2.2	Vibration condition states	6
5.2.3	LCD dashboard	6
6	Application design	7
6.1	Sensor acquisition and feature extraction	8
6.2	Gesture classification	9
6.3	FFT vibration analysis	11
6.4	Condition monitoring	11
6.5	Dominant-frequency stabilization	12
7	Conclusion	12
8	Acronyms	12
9	Revision history	13
	Legal information	14

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.
