AN15054

Flash-Based EEPROM Emulation on KE17Z

Rev. 1.0 — 18 June 2026 **Application note**

Document information

Information	Content
Keywords	AN15054, KE17Z, FRDM-KE17Z512, Flash, EEPROM emulation, P-Flash, FTFE
Abstract	This application note describes a Flash-based EEPROM emulation implementation for the KE17Z microcontroller. It explains the Flash constraints, software architecture, demo operation, reserved-memory integration, and endurance estimation used by the FRDM-KE17Z512 demo project.

1 Introduction

Many embedded applications store calibration values, counters, user settings, and production data that must survive a reset or power removal. A conventional EEPROM is convenient for this kind of data because it can be updated at byte or word granularity. The KE17Z devices do not provide a separate EEPROM array. The same user model can be implemented in the internal program Flash with a software-managed emulation layer.

This application note describes the Flash-based EEPROM emulation demo for FRDM-KE17Z512. The implementation uses a reserved area at the end of the internal P-Flash and presents a small record-based API to the application. The document focuses on demo configuration, how records are written and recovered, and required integration checks before using the method in a product.

2 Flash and EEPROM constraints

This section summarizes the Flash characteristics that drive the EEPROM emulation design. It explains why P-Flash cannot be updated in-place like EEPROM. It also explains how erase granularity, phrase programming, and endurance limits affect the record-based implementation used in this application note.

2.1 Program Flash behavior

Program Flash and EEPROM are both non-volatile memories, but they have different update rules. In NOR Flash, an erased cell reads as logic 1. A program operation can only change programmed bits from 1 to 0. Returning any bit from 0 to 1 requires an erase operation over a complete erase unit, which is a Flash sector for this demo. Therefore, writing a small variable directly back to the same Flash address quickly consumes sector erase-cycles and does not support in-place updates.

For the KE17Z512 configuration used here, the project treats the P-Flash sector size as 2 kB and programs data in 8-byte phrases. The wrapper uses the MCUXpresso Flash driver to erase sectors, program phrases, and verify the result with the configured read margin.

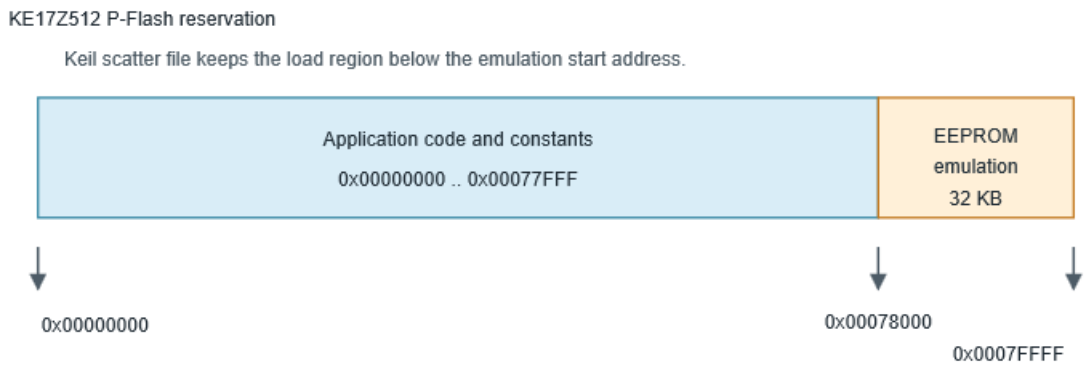

Figure 1. Reserved P-Flash area used for EEPROM emulation

2.2 Emulation approach

The emulation layer avoids overwriting old values. Each logical parameter uses a record ID for identification, and each update appends a new record to the current active sector. A read operation searches active sectors for the newest valid record with the requested ID. Older records remain in Flash until the sector swap operation reclaims them.

BLANK sectors contain erased Flash only.

ALTERNATIVE sectors have a valid erase-cycle value and can be used as swap targets.

ACTIVE sectors contain valid records and receive appended updates.

DEAD or INVALID sectors are excluded from normal allocation after a programming or erase failure.

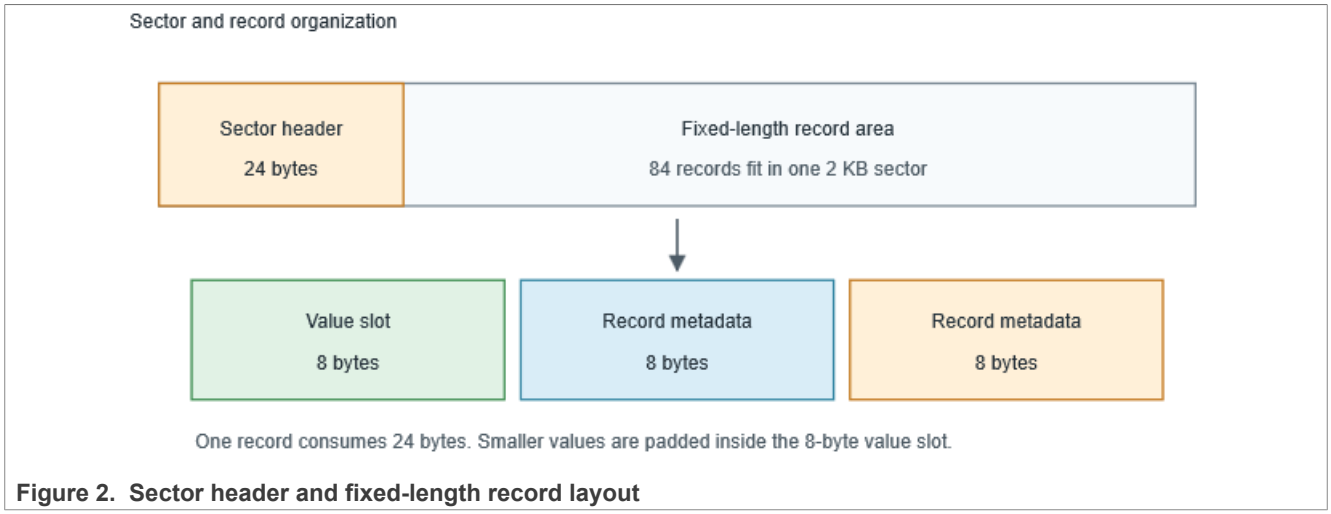

Figure 2. Sector header and fixed-length record layout

3 Software architecture

This section describes the software layers that the demo uses, from the application wrapper API to the generic EEPROM emulation driver and the KE17Z Flash driver. It also explains how initialization, record lookup, append writes, and sector swap are distributed across these layers.

3.1 Project layering

The demo separates the customer-facing API from the generic EEPROM emulation driver. The application calls the small wrapper in source/eeprom_demo_port.c, while the record management, sector state machine, search logic, and swap flow are implemented in component/eeprom_emulation/fsl_eeprom_emulation.c.

Table 1. Project layering

Layer	Main files	Responsibility
Application demo	source/main.c	• Initializes the board • Reads three parameters • Increments bootCount • Writes changed values • Prints the result
KE17Z wrapper	source/eeprom_demo_port.c/.h	• Sets KE17Z Flash parameters • Validates buffers and record IDs • Skips unchanged writes • Retries updates • Verifies readback
Emulation driver	component/eeprom_emulation/fsl_eeprom_emulation.c/.h	• Maintains sector states • Appends records • Searches latest records • Swaps sectors • Recovers after reset or interrupted update
Flash driver	drivers/fsl_ftfx_flash.c	• Provides sector erase, phrase program, erase verify, and program verify operations for the P-Flash.

3.2 Current configuration

The project reserves 32 kB at the end of the 512 kB internal P-Flash. The public API supports records from 1 to 100, and each record stores an application value from 1 byte to 8 bytes. Values smaller than 8 bytes are copied into an 8-byte slot and padded before programming.

Table 2. Current configuration

Parameter	Value
EEPROM emulation start address	0x00078000
EEPROM emulation end address	0x0007FFFF
Reserved Flash size	32 kB
P-Flash sector size	2 kB
Flash program unit	8 bytes
Program check unit	4 bytes
Maximum logical records	100
Application value size	1 byte to 8 bytes
Internal value slot	8 bytes
Sector header size	24 bytes
Record length	24 bytes
Records per sector	84
Required active sectors	2
Extra active sectors	10
Ready sectors	4
Total allocated sectors	16

3.3 Initialization and recovery

EepromDemo_Init() first reads Flash properties through the Flash driver. It verifies that the reserved region is inside the P-Flash range and aligned to the detected sector size. It then fills eeprom_emulation_config_t and calls EE_SetEepromEmulationInfo() to calculate the record length, sector capacity, active-sector count, ready-sector count, and the total allocated Flash range.

EE_Init() scans the allocated sectors at boot time. The driver:

- Validates active indicators, dead indicators, and erase-cycle fields
- Converts blank sectors into alternative sectors when needed
- Creates enough active sectors for the configured logical EEPROM size
- Recovers sectors affected by an interrupted update

This startup scan is the main protection against reset or power removal during a write or swap sequence.

3.4 Read, write, and sector swap

EepromDemo_ReadValue() checks the record ID and buffer size, calls EE_ReadData(), and copies the requested number of bytes from the newest record. A missing record is not treated as a reason to format the entire emulation area; the application keeps its compiled default value.

EepromDemo_WriteValue() prepares an 8-byte record buffer, compares it with the current stored value, and skips the Flash update when the value is unchanged. When a write is required, EE_WriteData() appends the record to the current active sector. If active sectors are full, the driver moves to the next active sector or performs a swap through an alternative sector. The wrapper retries after EE_ERR_UPDATE by re-running initialization and then verifies the newly written value by reading it back.

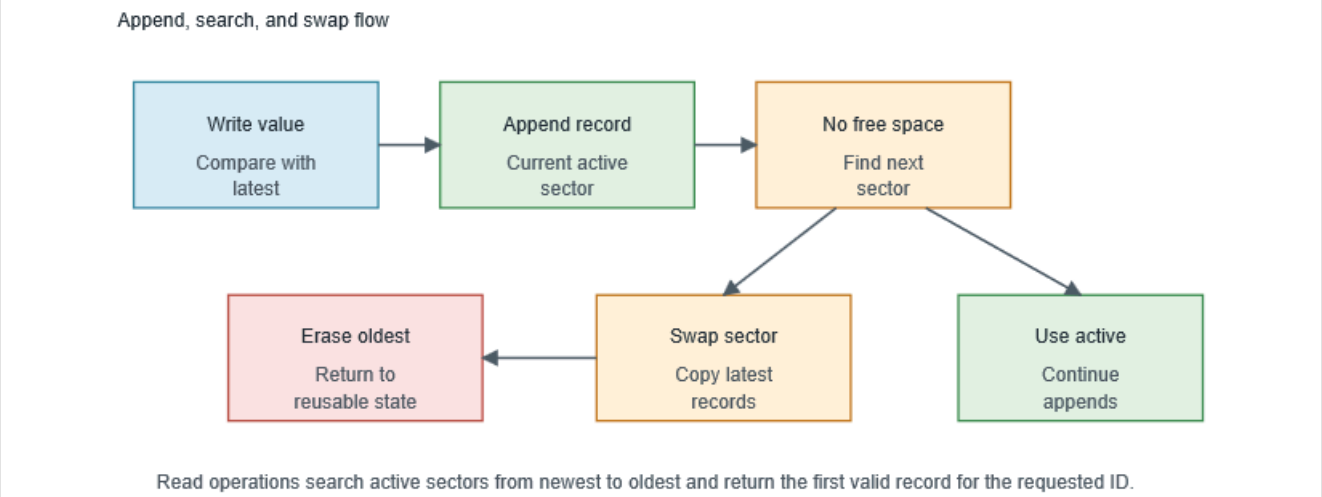

Figure 3. Append, read, and sector-swap flow

4 Running the demo

The demo in source/main.c stores three example variables. On the first run, no records exist, so the compiled default values are used and written to Flash. On subsequent resets, bootCount is read back, incremented, and written again, demonstrating that the data survives the reset.

Table 3. Demo variables

Record ID	Type	Variable	Default value
1	int8_t	temperatureOffset	-5
2	uint32_t	bootCount	0
3	uint16_t	startDelayMs	1000

```
bool EepromDemo_Init(void);
bool EepromDemo_Format(void);
bool EepromDemo_ReadValue(uint16_t recordId, void *value, uint32_t valueSize);
bool EepromDemo_WriteValue(uint16_t recordId, const void *value, uint32_t
 valueSize);
```

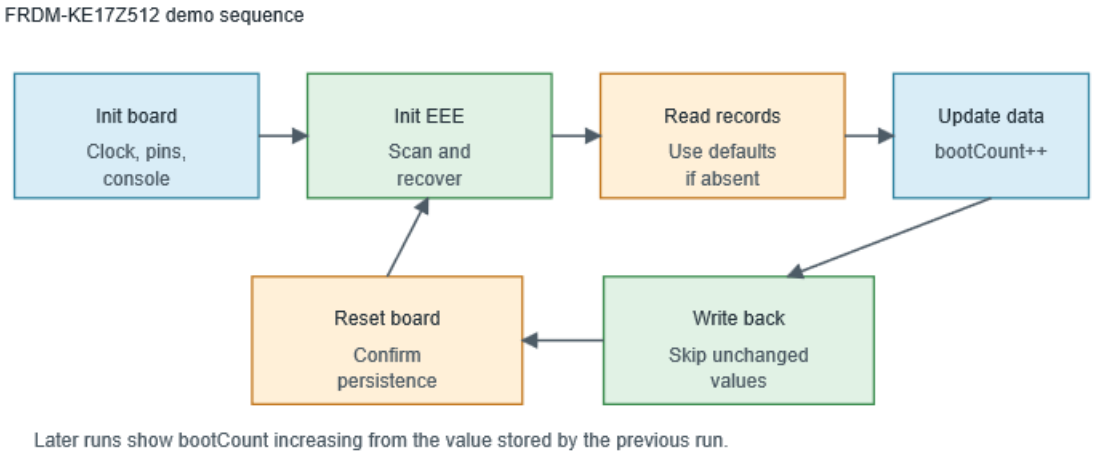

Figure 4. Demo execution flow

5 Integration considerations

This section lists the integration checks required when re-using the demo in an application. It includes Flash reservation, linker configuration, API limits, formatting behavior, and endurance planning.

5.1 Reserve the Flash area

The linker or scatter file must keep application code and constant data out of the emulation region. In this Keil project, MKE17Z512xxx9_flash.scf defines m_text_start as 0x00000410 and m_text_size as 0x00077BF0, so the load region ends at 0x00078000. This boundary matches the wrapper start address and prevents application code from occupying the EEPROM emulation area.

If the reserved size, record count, or value size is changed, update both the wrapper constants and the linker memory layout. Check the build map to confirm that the load region maximum does not exceed the reserved Flash range.

5.2 API limits

Before using the wrapper API in an application, understand the record ID range, value size, write-skip behavior, and format behavior. These rules prevent accidental Flash erases and unnecessary program operations.

Table 4. API limits

API rule	Behavior
recordId	The valid range is 1 to 1000. Record ID 0 is rejected.
valueSize	The valid range is 1 byte to 8 bytes for the current wrapper.
Unchanged value	Write is skipped to avoid unnecessary Flash program operations.
Missing record	Read returns false; the application can keep its default value.
Format	EepromDemo_Format() erases the allocated emulation area and must be used only for explicit initialization or parameter clearing.
Update error	The wrapper reinitializes the emulation state and retries the write.

5.3 Endurance estimate

The *Kinetis KE17Z/13Z/12Z with up to 512 KB Flash Data Sheet* (KE1XZP100M96SF0) specifies program Flash cycling endurance as 10 kcycle minimum and 50 kcycle typical. Estimate the endurance of the emulated EEPROM in appended records, not in API calls alone. Each changed logical value consumes one new record and sector swaps copy live records.

records_per_sector = floor((2048 - 24) / 24) = 84 records

total_record_appends_at_10k = 10000 cycles * 16 sectors * 84 records = 13,440,000 record append

Table 5. Endurance estimate by changed variables per save

Changed variables per save	Records consumed per save	Theoretical saves at 10 kcycle	Suggested 2x-margin budget
1	1	13,440,000	6,720,000
5	5	2,688,000	1,344,000
10	10	1,344,000	672,000

These values are planning estimates. A product design must reserve extra margin for copied records during swaps, failed write retries, low-voltage events, temperature profile, and application-specific write distribution.

5.4 Build evidence

The current project output records successful debug and release builds with zero errors and zero warnings. The map file shows a load region maximum of 0x00078000 and total ROM usage of approximately 16.71 kB. This memory usage leaves the 32 kB reserved emulation area untouched.

6 Conclusion

The KE17Z Flash EEPROM emulation demo provides a compact non-volatile parameter store without adding external EEPROM hardware. The implementation relies on the following:

- Append-only fixed-length records
- Latest-record search
- Round-robin active sectors
- Alternative sectors for swap
- Boot-time recovery

For product integration, the following are the most important checks:

- Reserve the Flash range in the linker file
- Keep record sizes within the configured 8-byte slot
- Avoid unnecessary writes
- Verify endurance against the expected update rate

7 Acronyms

Table 6 lists the acronyms used in this document.

Table 6. Acronyms

Acronym	Description
API	Application Programming Interface

Table 6. Acronyms...*continued*

Acronym	Description
EEPROM	Electrically Erasable Programmable Read-Only Memory
FTFE	The flash memory module that the KE17Z P-Flash driver uses
P-Flash	Program Flash, the internal non-volatile memory used for application code and the reserved emulation area
Phrase	The 8-byte Flash programming unit that the KE17Z512 configuration uses
Record	A fixed-length Flash entry containing one logical data value and metadata
Sector swap	The process that copies the latest valid records into a reusable sector and erases the oldest active sector

8 References

- EEPROM Emulation Driver for the Kinetis E Series Microcontrollers, AN4903
- EEPROM Emulation with Qorivva MPC55xx, MPC56xx, and MPC57xx Microcontrollers, AN4868
- Kinetis KE17Z/13Z/12Z with up to 512 KB Flash Data Sheet, KE1XZP100M96SF0
- Kinetis KE17Z/13Z/12Z with up to 512 KB Flash Reference Manual, KE1XZP100M96SF0RM
- FRDM Development Board for 96 MHz KE17Z/KE13Z/KE12Z with 512 KB Flash MCUs FRDM-KE17Z512

9 Revision history

Table 7 summarizes the revisions to this document.

Table 7. Revision history

Revision number	Release date	Description
AN15054 v.1.0	18 June 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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

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.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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.

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.

Kinetis — is a trademark of NXP B.V.

Tables

Figures

Contents