

AN15124

How to Drive a Memory-In-Pixel (MIP) Panel Using SmartDMA on MCX A Series MCUs

Rev. 1.0 — 12 October 2026

Application note

Document information

Information	Content
Keywords	AN15124, MIP, panel, display, LCD, SmartDMA, MCX A, FRDM-MCXA174, low power
Abstract	This document describes a SmartDMA-based solution for driving Memory-In-Pixel (MIP) LCD panels on MCX A series MCUs.



1 Introduction

Memory-In-Pixel (MIP) displays combine ultralow static power consumption with high reflectivity, making them ideal for wearable and always-on embedded devices. However, MIP panels require strict timing sequences for address and data operations. Meeting these timing requirements can be challenging using CPU-driven GPIO toggling, especially when the system is under interrupt or multitasking load.

This application note describes an efficient method for driving a MIP panel using SmartDMA available on MCX A series microcontrollers. The MCX A174 is used as the reference platform throughout this document. SmartDMA provides deterministic GPIO control that operates independently of CPU execution. This approach enables the precise waveform generation required by MIP displays without the need for a dedicated LCD controller, while minimizing overall system power consumption.

The architecture presented in this document can be reused across MCX A series devices that integrate SmartDMA and adapted to various MIP panel resolutions and interfaces.

2 System overview

MIP displays are widely used in wearable, portable, and always-on embedded systems. They offer ultralow static power consumption while retaining pixel data internally, making them well suited for low-power applications. Unlike conventional LCDs, MIP panels require precise, repeatable timing sequences for address setup, data loading, and pixel-memory update operations. These interface characteristics place strict requirements on the timing of GPIO waveforms, such as Serial Clock (SCLK), Data Enable (DEN), and ENBG/ENBS (Vertical/Horizontal Write Enable signals).

MCX A series MCUs that integrate the SmartDMA peripheral are well suited for driving MIP panels. SmartDMA acts as a lightweight execution engine that autonomously generates protocol-accurate GPIO timing autonomously. As a result, it eliminates the need for CPU-driven bit toggling and significantly reduces processor load significantly. This capability improves energy efficiency and allows the system to remain in low-power modes during display update cycles.

This application note uses the MCX A174 as the reference MCU. It provides a balanced set of resources, including a Cortex-M33 core, a flexible GPIO subsystem, and SmartDMA for driving MIP displays efficiently through software-defined parallel interface. Although the accompanying example uses a 176 × 176 Kyocera TN0104ANVAANN-GN00 MIP LCD, the described architecture applies to other MIP panels and GPIO-based display interfaces that require deterministic timing.

The subsections that follow provide an overview of the MCX A174 reference platform and the target MIP display module. They also describe the overall system architecture used to construct a SmartDMA-driven MIP display pipeline.

2.1 MCX A174 overview

The MCX A174 integrates the following key features:

- Arm Cortex-M33 core (without TrustZone support)
- 64 kB SRAM / 256 kB on-chip flash memory
- SmartDMA execution engine for deterministic GPIO edge timing
- Low-power operating modes suitable for always-on User Interfaces (UIs)

The SmartDMA peripheral includes a compact EZH micro-execution core, which can run short instruction sequences with fixed timing. This capability makes it ideal for controlling MIP display interfaces.

2.2 SmartDMA overview

SmartDMA is a dedicated hardware execution engine integrated into selected MCX A series MCUs. It is suited for implementing timing-critical GPIO-based display interfaces, such as MIP panels.

Traditional Direct Memory Access (DMA) controllers only transfer data between memory and peripherals. In contrast, SmartDMA includes an EZH microexecution core capable of running compact instruction sequences with fixed execution timing. These instruction sequences allow SmartDMA to generate precise, cycle-accurate GPIO waveforms autonomously. This capability makes it especially suitable for interfaces that require deterministic timing, such as MIP LCD panels.

By delegating timing-sensitive operations to SmartDMA, the system eliminates software-driven bit toggling and reduces CPU load significantly. This capability ensures that display interfaces operate with consistent and repeatable timing, independent of system interrupts or CPU activity.

2.3 MIP panel overview

The reference design uses the Kyocera TN0104ANVAANN-GN00, a 176×176 monochrome MIP LCD as a representative panel. Its key attributes include:

- Reflective display technology, eliminating the need for a backlight
- 1-bit internal pixel memory for static image retention
- Pixel updates are performed in 8-pixel horizontal units
- Serial-input/parallel-data architecture comprising SI[7:0], SCLK, DEN, ENBG/ENBS
- Strict timing requirements for address and write strobe phases

These characteristics make the panel well suited for a deterministic timing engine such as SmartDMA.

3 System architecture

This section describes the overall hardware and software architecture used to drive a MIP panel using SmartDMA on the MCX A series. The reference design is centered on the following three major components:

- The MCX A174 MCU, which provides the CPU, GPIO subsystem, and SmartDMA engine
- The TN0104 176×176 MIP LCD panel, which is used as the display target
- Software-defined display pipeline that includes render logic, frame buffer management, and SmartDMA firmware

Together, these elements form a flexible, low-power display control system capable of meeting the strict timing requirements of MIP displays.

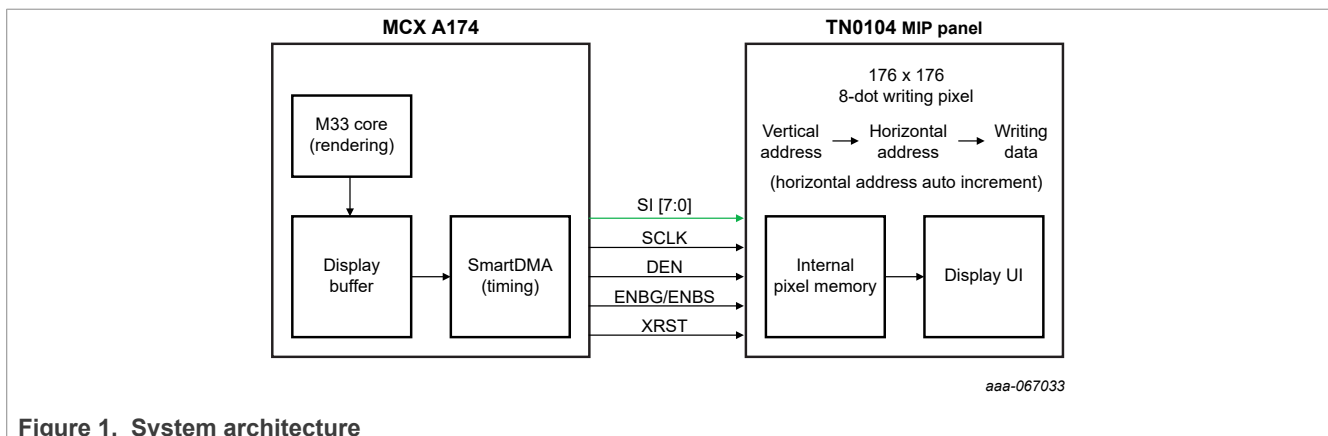


Figure 1. System architecture

3.1 Hardware architecture

The hardware architecture consists of the MCX A174 MCU connected to the TN0104 LCD through GPIO pins. These GPIO pins are configured as an 8-bit address/data bus together with several control signals. The SmartDMA manipulates these GPIO pins to generate the appropriate display protocol sequences.

3.1.1 Hardware connections

[Table 1](#) summarizes the GPIO pin assignment used to interface the MCX A174 with the TN0104 MIP panel in the reference design. All display interface signals are generated through GPIO pins controlled by the SmartDMA peripheral.

Table 1. GPIO connections between the MCX A174 and TN0104 MIP panel

Signal	Direction	MCX A174 GPIO	FRDM board connector	Description
SI[0]	Output	P0_14	J8-13	Data/address bus bit 0 input to the MIP panel
SI[1]	Output	P0_15	J8-14	Data/address bus bit 1 input to the MIP panel
SI[2]	Output	P0_16	J8-15	Data/address bus bit 2 input to the MIP panel
SI[3]	Output	P0_17	J8-16	Data/address bus bit 3 input to the MIP panel
SI[4]	Output	P0_18	J8-17	Data/address bus bit 4 input to the MIP panel
SI[5]	Output	P0_19	J8-18	Data/address bus bit 5 input to the MIP panel
SI[6]	Output	P0_20	J8-19	Data/address bus bit 6 input to the MIP panel
SI[7]	Output	P0_21	J8-20	Data/address bus bit 7 input to the MIP panel
SCLK	Output	P0_22	J8-21	Serial clock signal for address and display data transfer
ENBG/ENBS	Output	P0_23	J8-22	Write enable strobe signals for vertical and horizontal pixel updates
XRST	Output	P2_10	J8-23	Active-low hardware reset signal for the MIP panel
DEN	Output	P2_11	J8-24	Address/data selection control signal

All display interface signals are generated through GPIO pins controlled exclusively by SmartDMA during display updates. As a result, the CPU does not directly toggle these pins, preventing timing jitter.

3.2 Software architecture

The software architecture for driving the TN0104 MIP panel on MCX A174 is designed to clearly separate high-level rendering logic from timing-critical display signaling. This separation allows the system to achieve deterministic display control, reduced CPU load, and optimized power consumption.

The architecture is divided into the following two primary execution domains:

- CPU domain (Cortex-M33), responsible for user interface rendering and high-level control.

- SmartDMA domain, responsible for autonomous generation of the GPIO timing and display update sequences.

The CPU and SmartDMA interact through a shared frame buffer and a parameter block. This interaction enables a clean hand-off between rendering and display refresh operations.

3.2.1 Software function allocation

The responsibilities of the CPU and SmartDMA are explicitly partitioned to ensure predictable timing behavior and efficient system operation.

Following are the responsibilities of CPU (Cortex-M33):

- Render graphical content into a 1-bit frame buffer
- Manage application logic and display update conditions
- Prepare the SmartDMA parameter block, including frame buffer address, row count, and transfer length
- Trigger SmartDMA execution and optionally enter a low-power state

Following are the responsibilities of SmartDMA:

- Generate all timing-critical GPIO waveforms required by the MIP panel
- Manage vertical and horizontal address sequencing
- Stream pixel data with deterministic timing
- Generate write strobes and panel reset sequences
- Operate autonomously without CPU intervention during display refresh

This division of responsibilities ensures that the CPU is not involved in GPIO toggling or timing-sensitive operations during display updates.

3.2.2 Software execution flow

The overall software execution flow follows a producer-consumer model, where the CPU produces display content and SmartDMA consumes it to update the panel.

The overall execution flow is shown below:

1. System initialization
 - a. Initialize the GPIO pins, SmartDMA, and frame buffer.
 - b. Perform the panel reset and stabilization sequence.
2. Frame buffer update
 - a. The CPU renders UI content into the frame buffer.
 - b. All rendering operations are performed in SRAM using a 1-bit pixel.
3. SmartDMA configuration
 - a. The CPU fills a parameter structure describing:
 - Frame buffer base address
 - Start row
 - Number of rows to update
 - Bytes per row (22 bytes for TN0104)
4. Display refresh
 - a. The CPU triggers SmartDMA execution.
 - b. SmartDMA updates the MIP panel autonomously through GPIO.
5. Completion and idle state
 - a. SmartDMA clears the execution flag upon completion.
 - b. The CPU can enter low-power mode or continue application processing.

This execution flow allows display updates to be performed with minimal CPU involvement and highly deterministic timing.

3.2.3 SmartDMA display update processing

Within the SmartDMA domain, display updates follow a fixed and repeatable sequence that matches with the requirements of the TN0104 MIP panel.

For each display update, SmartDMA performs the following steps:

1. Panel control setup
 - a. Configure GPIO output states.
 - b. Generate a panel reset pulse (XRST), if required.
2. Row-based update loop
 - a. Output the vertical (row) address.
 - b. Output the horizontal start address.
 - c. Enable the horizontal address auto-increment mode.
 - d. Stream the pixel data bytes from the frame buffer.
 - e. Generate the ENBG/ENBS write strobe to commit row data.
3. Row increment
 - a. Advance the frame buffer pointer to the next row.
 - b. Increment the row index.
4. Completion
 - a. Clear the SmartDMA execution flag.
 - b. Optionally trigger the completion interrupt.

3.2.4 MIP panel interface timing

The TN0104 panel uses a structured write format consisting of a vertical address, a horizontal address, and pixel data.

- Vertical address → Horizontal address → Pixel data
- Pixel data is written in 8-dot (1-byte) horizontal units
- Horizontal address auto-increment reduces control overhead and improves transfer efficiency

SmartDMA generates all address, data, and write-strobe signals according to the MIP panel protocol. The display update follows a vertical-address, horizontal-address, and data-write sequence with horizontal auto-increment.

[Figure 2](#) illustrates the corresponding address, data, and timing relationships:

How to Drive a Memory-In-Pixel (MIP) Panel Using SmartDMA on MCX A Series MCUs

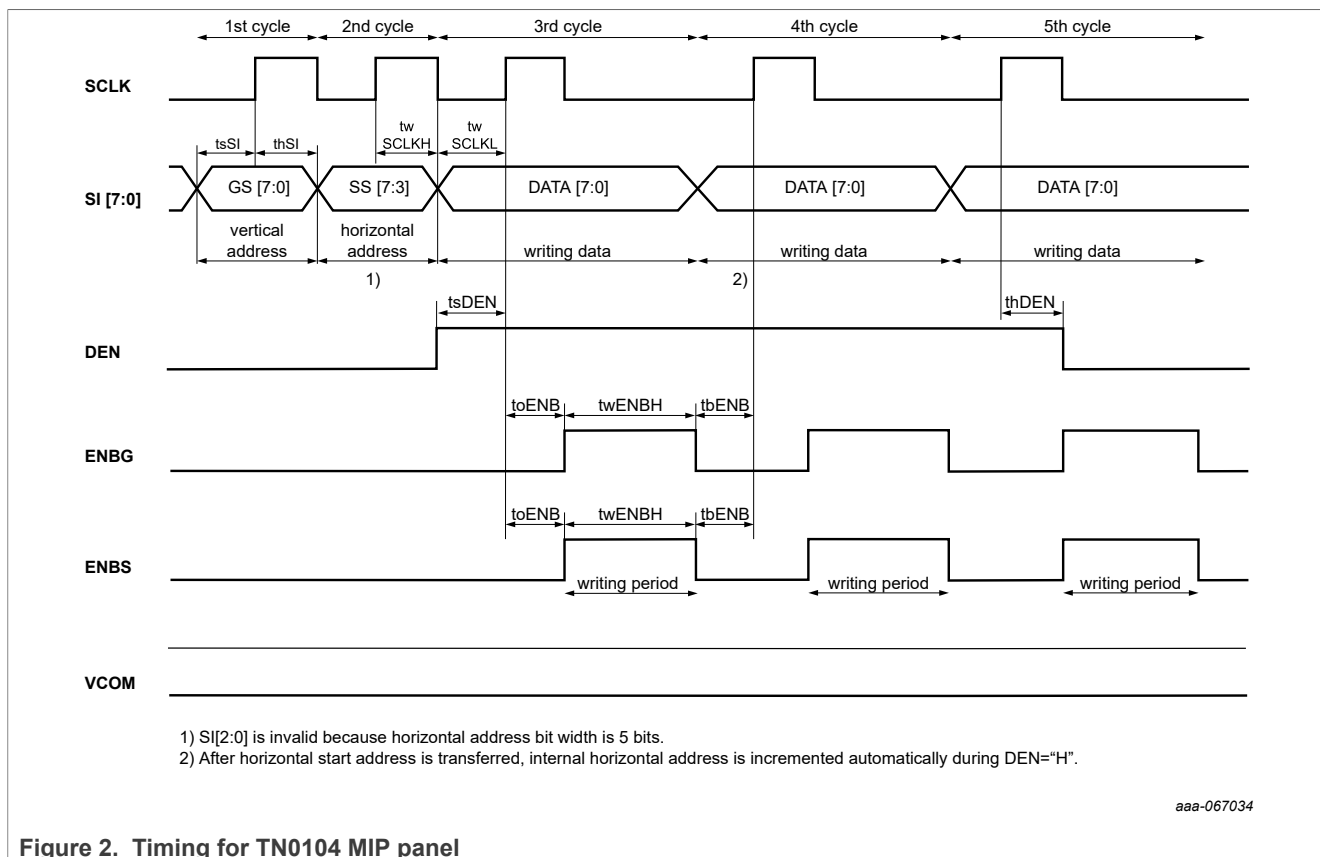


Figure 2. Timing for TN0104 MIP panel

[Table 2](#) summarizes the key timing requirements extracted from the panel data sheet along with the timing values implemented in the demo configuration:

Table 2. Key timing requirements for TN0104 MIP panel interface

Signal group	Timing requirement	Minimum requirement	Implemented value (Demo)	Unit
SCLK	High/low pulse width (twSCLKH/twSCLKL)	0.9	1.0	μs
SI[7:0]	Data set up and hold (tsSI/thSI)	0.9	1.0	μs
DEN	Setup and hold (tsDEN/thDEN)	0.9	1.0	μs
ENBG/ENBS	Write strobe high width (twENBH)	4.5	4.5	μs
ENBG/ENBS	SCLK to strobe delay (toENB/tbENB)	1.8	2.0	μs

Together, this processing model allows SmartDMA to function as a lightweight, software-defined display timing controller. As a result, the CPU remains fully decoupled from all real-time signaling requirements.

4 Running the demo

This section provides the procedures for running the demo project on the FRDM-MCXA174 board and presents the observed results.

4.1 Hardware setup

The following hardware is required to run the demo:

- FRDM-MCXA174 board
- TN0104 MIP panel
- TN0104 adapter board

As described in [Section 3.1.1](#) and summarized in [Table 1](#), connect the TN0104 MIP panel to FRDM-MCXA174 as shown in [Figure 3](#).

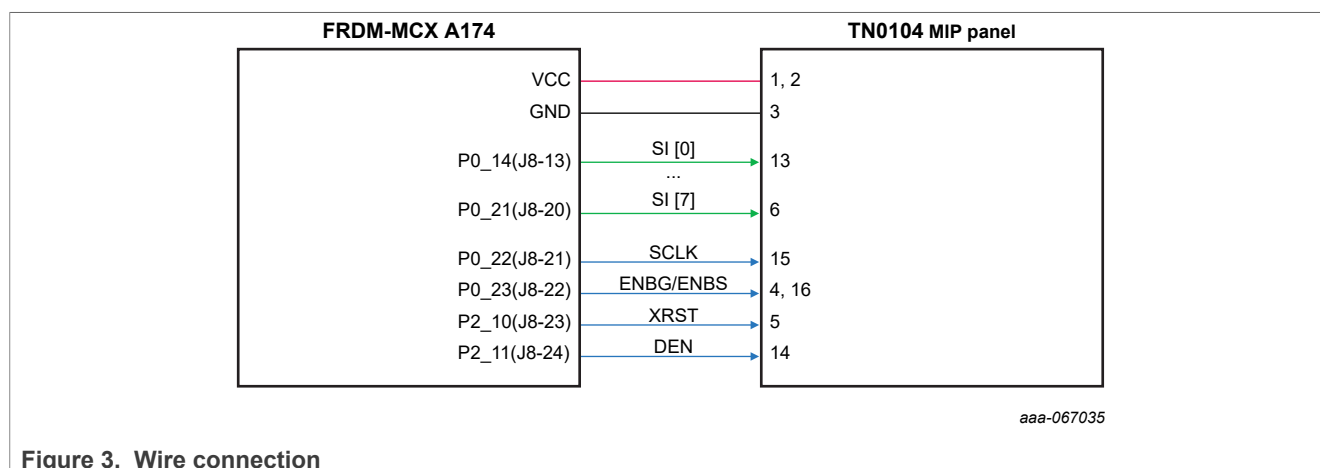


Figure 3. Wire connection

4.2 Software setup

The following software is required to run the demo:

- MCUXpresso for VS Code extension (26.7.52)
- SDK_26_03_00_FRDM-MCXA174

To run the demo, follow the steps below:

1. Download the project from [Application Code Hub](#).
2. Connect the FRDM-MCXA174 board to the host PC using a USB-Type C cable.
3. Download the program onto the target board.
4. To start running the demo, press the 'reset' button on your board or launch the debugger in the IDE to begin execution.

4.3 Demo result

[Figure 4](#) shows the UI content displayed on the TN0104 MIP panel, including the date, day of the week, time, and the NXP logo.



4.4 Power consumption measurement

In addition to functional verification, the power consumption of the system is measured to evaluate the low-power performance of the SmartDMA-based display update solution.

The system operates with a periodic 1-second update cycle. During each cycle, the MCU enters Active mode briefly to update the display and then transitions to Power-Down mode for the remaining time.

To obtain accurate MCU current measurements on the FRDM-MCXA174 board, the following hardware modifications are applied:

- Remove resistors R43, R20, R21, and R27
- Measure current from the IDD_MCU header (JP1)

These modifications ensure that only the MCU current is captured, excluding other onboard components.

The measured timing characteristics are:

- Active mode duration: 4 ms (0.4 %)
- Power Down duration: 996 ms (99.6 %)

The measured current for each mode and the calculated average current are summarized in [Table 3](#).

Table 3. Power consumption measurement results

Mode	Duration (ms)	Duty cycle (%)	Current (μA)	Weighted current (μA)
Active	4	0.4	1306	5.2
Power Down	996	99.6	17.7	17.6
Average	1000	100	-	22.8

During Active mode, the CPU prepares display data and triggers the SmartDMA engine to perform the display update. The SmartDMA autonomously generates all required GPIO timing, minimizing CPU involvement. After the update is complete, the system immediately enters Power-Down mode.

The average current consumption is calculated as:

$I_{avg} = (I_{active} \times D_{active}) + (I_{power_down} \times D_{power_down})$

Substituting the measured values:

$I_{avg} = (1306 \mu A \times 0.004) + (17.7 \mu A \times 0.996) \approx 22.8 \mu A$

The calculated result matches with the measured average current, validating the correctness of the measurement setup and methodology.

The system achieves an ultralow average power consumption, making it well suited for battery-powered and always-on applications, such as wearable devices and low-power IoT systems.

5 Conclusion

This application note has presented a complete and efficient method for driving MIP panels using the SmartDMA peripheral on MCX A series MCUs. SmartDMA provides a deterministic execution model and GPIO-driven timing capability. As a result, MCX A devices can reliably meet the strict signaling requirements of MIP displays without requiring dedicated hardware display controllers.

The MCX A174 serves as an effective reference platform due to its balanced memory footprint, low-power architecture, and integrated SmartDMA engine. In this design, the CPU is responsible for frame buffer rendering and application logic, while SmartDMA autonomously handles timing-critical LCD update sequences. This clean division of responsibilities significantly reduces CPU workload, enables aggressive use of low-power modes, and provides a predictable and scalable display-update mechanism.

The presented architecture demonstrates how SmartDMA can operate as a programmable display timing controller, generating SCLK, DEN, ENBG/ENBS, and SI[7:0] waveforms with cycle-accurate precision. This capability eliminates the timing jitter normally associated with CPU-driven GPIO toggling and ensures reliable pixel-memory updates across all rows and columns of the MIP panel.

The techniques and structure described in this document can be applied broadly to:

- MIP panels with different resolutions and interface variants
- Partial or region-specific updates
- Always-on, battery-powered wearable or IoT devices
- Any GPIO-driven display interface that requires deterministic signal timing

By pairing the energy efficiency of MIP technology with the autonomous execution capability of SmartDMA, MCX A series MCUs deliver a robust and highly optimized platform for next-generation low-power display designs.

6 Acronyms

Table 4 lists the acronyms used in this document along with their description.

Table 4. Acronyms

Term	Description
DEN	Data Enable
DMA	Direct Memory Access
ENBG/ENBS	Vertical Write Enable/Horizontal Write Enable Signals
MIP	Memory-In-Pixel
SCLK	Serial Clock
UIs	User Interfaces

7 Reference

Following is the list of the references used to supplement this document:

- *MCX A173 and MCX A174 Reference Manual* ([MCXAP100M180F41RM](#))
- *SmartDMA Cookbook* ([AN14650](#))
- [TN0104ANVAANN-GN00 LCD Module Datasheet](#)

8 Revision history

[Table 5](#) summarizes the revisions to this document.

Table 5. Revision history

Document ID	Release date	Description
AN15124 v.1.0	12 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.

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.

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.

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.

How to Drive a Memory-In-Pixel (MIP) Panel Using SmartDMA on MCX A Series MCUs

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, μ Vision, 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.

Contents

1	Introduction	2
2	System overview	2
2.1	MCX A174 overview	2
2.2	SmartDMA overview	3
2.3	MIP panel overview	3
3	System architecture	3
3.1	Hardware architecture	4
3.1.1	Hardware connections	4
3.2	Software architecture	4
3.2.1	Software function allocation	5
3.2.2	Software execution flow	5
3.2.3	SmartDMA display update processing	6
3.2.4	MIP panel interface timing	6
4	Running the demo	7
4.1	Hardware setup	8
4.2	Software setup	8
4.3	Demo result	8
4.4	Power consumption measurement	9
5	Conclusion	10
6	Acronyms	10
7	Reference	10
8	Revision history	11
	Legal information	12

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