AN14175

Using FlexIO to emulate Quad SPI master

Rev. 1.0 — 20 January 2024

Application note

Document information

Information	Content
Keywords	AN14175, FlexIO module, board interfaces, MCX-N947-EVK, i.MX RT595-EVK hardware platforms, simplex Quad SPI master, LCD driver
Abstract	This application note describes the implementation of Flex IO peripheral as a simplex Quad SPI master for LCD driver on MCX-N947-EVK or i.MX RT595-EVK hardware platforms.

1 Introduction

FlexIO is an on-chip peripheral available on the Kinetis, S32K, i.MX RT and MCX microcontroller families. It is highly configurable and capable of emulating a wide range of communication protocols, such as UART, I2C, SPI, I2S, and LIN. These protocols are described in the application note, "*Using FlexIO to emulate communications and timing peripherals*" (AN12174) on nxp.com. FlexIO can also be used to emulate other protocols such as J1850, I3C, and Manchester.

The standalone peripheral module FlexIO is used as an additional peripheral module of the microcontroller and is not a replacement of any communication peripheral. The key feature of FlexIO is that it enables users to build their own peripherals depending on their requirements.

This application note describes the implementation of simplex Quad SPI master for LCD driver on NXP provided MCX-N947-EVK and i.MX RT595-EVK hardware platforms. Half-duplex or full-duplex QSPI might also be possible but are not within the scope of this application note.

2 Overview of the FlexIO module

FlexIO module has the following main hardware resources:

- Shifter
- Timer
- Pin

Figure 1 shows a high-level overview of the FlexIO module.

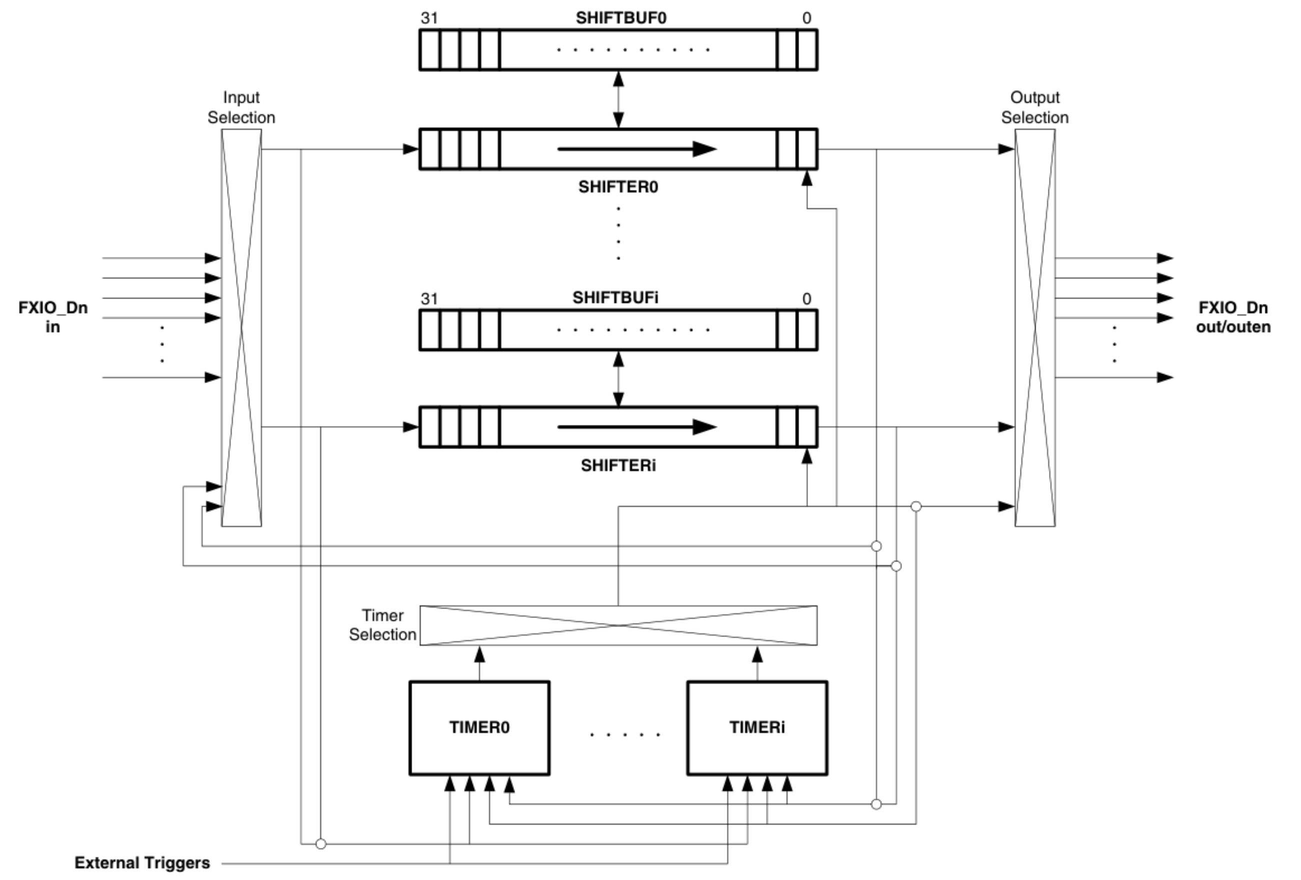


Figure 1. FlexIO block diagram

The following key features are provided:

- 32-bit shifters with transmit, receive, and data match modes
- Double buffered shifter operation
- 16-bit timers with high flexibility support for various internal or external triggers and reset/enable/disable/ decrement conditions
- Automatic start/stop bit generation/check
- Interrupt, DMA, or polling mode operation
- Shifters, timers, pins, and triggers can be flexibly combined to operate

Transmit and receive are two basic modes of the shifters. If one shifter is configured to Transmit mode, it loads data from its buffer register and shifts data out to its assigned pin bit by bit. If one shifter is configured to Receive mode, it shifts data in from its assigned pin and stores data in its buffer register. The shifter's assigned timer controls all the load, store, and shift operations. The timers can also be configured in different operational modes as per your requirement. These include the dual 8- bit counter baud/bit mode, dual 8-bit counter PWM mode, and single 16-bit counter mode.

For more details, refer to the application note AN12174 and the Reference Manual of the respective device on nxp.com.

3 Emulating Quad SPI master using FlexIO

3.1 Requirements

Figure 2 shows an expected waveform for a project.

The communication starts from the command signal (one byte) and address signals (three bytes). A dummy cycle (one byte) is inserted at the head and foot of the data.

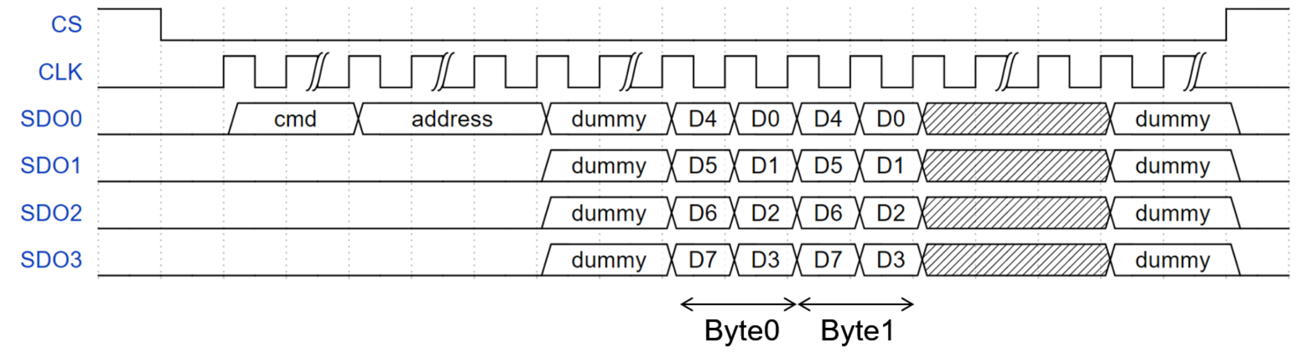


Figure 2. FlexIO waveform

3.2 Simplex Quad SPI configuration

The basic concept is same as the concept for the ordinal SPI. (AN12174). Shifter 0 is used as the Quad SPI master transmitter. Table 1 lists the configurations.

Table 1. Configurations for Shifter 0

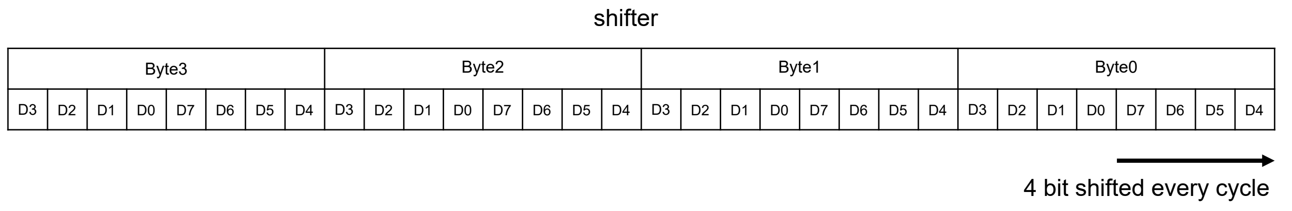
Items	Configurations
shifter mode	transmit
timer selection	timer 0
timer polarity	on negative of shift clock
pin configuration	pin output
pin polarity	active high
pin width	**3**
input source	from pin
start bit	disabled, transmitter loads data on enable
stop bit	disabled
buffer used	nibble byte swapped register

The key difference is the PWIDTH (pin width) register. By configuring the PWIDTH register value to 3, 4 bits are shifted in each cycle, which realizes parallel quad outputs. A sample code to set the width is mentioned below. Also refer Figure 3.

```
/* Configure the shifter 0 for tx. */
            shifterConfig.timerSelect = kFLEXIO_QSPI_TIMER0;
            shifterConfig.pinConfig   = kFLEXIO_PinConfigOutput;
            shifterConfig.pinSelect   = base->SDOPinIndex;
            shifterConfig.pinPolarity = kFLEXIO_PinActiveHigh;
```

```
           shifterConfig.shifterMode = kFLEXIO_ShifterModeTransmit;
           shifterConfig.inputSource = kFLEXIO_ShifterInputFromPin;
           shifterConfig.parallelWidth = 3;
           if (masterConfig->phase == kFLEXIO_SPI_ClockPhaseFirstEdge)
           {
           shifterConfig.timerPolarity =
kFLEXIO_ShifterTimerPolarityOnNegitive;
           shifterConfig.shifterStop   = kFLEXIO_ShifterStopBitDisable;
           shifterConfig.shifterStart  =
kFLEXIO_ShifterStartBitDisabledLoadDataOnEnable;
           }
           else
           {
           shifterConfig.timerPolarity =
kFLEXIO_ShifterTimerPolarityOnPositive;
           shifterConfig.shifterStop   = kFLEXIO_ShifterStopBitLow;
           shifterConfig.shifterStart  =
kFLEXIO_ShifterStartBitDisabledLoadDataOnShift;
           }

           FLEXIO_SetShifterConfig(base->flexioBase, kFLEXIO_QSPI_SHIFTBUF0,
&shifterConfig);
```



Byte3								Byte2								Byte1								Byte0							
D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4

Figure 3. PWIDTH register

To store nibble swapped data to shift buffer, SHIFTBUFNBS (Shifter Buffer N Nibble Byte Swapped) register is used. Refer Table 2.

Table 2. SHIFBUFNBS register

Field	Description
31-0 SHIFTBUFNBS	Shift Buffer Alias to SHIFTBUF register, except reads/writes to this register are nibble swapped within each byter. Reads return {SHIFTBUF[27:24], SHIFTBUF[31:28], SHIFTBUF[19:16], SHIFTBUF[23:20], SHIFTBUF[11:8], SHIFTBUF[15:12], SHIFTBUF[3:0], SHIFTBUF[7:4] }.

- Timer 0 is used by the Quad SPI master to generate the clock output and control load/store/shift of the shifter.
- Timer 1 is used to generate the chip select output.

Refer to the application note AN12174 on nxp.com for more information about the timer 0 or timer 1 configuration.

4 Software implementation overview

i.MX RT595-EVK and MCX-N947-EVK boards were used to test the driver.

The i.MX RT595 software example supports the SmartDMA implementation while the MCX-N947 software example supports the eDMA implementation. SmartDMA needs a custom firmware for moving data from a buffer to SHIFTBUFNBS on demand, which is included in `fsl_smartdma.h` as a binary.

As a result, **i.MX RT595 uses**:

- `fsl_flexio_qspi.c/fsl_flexio_qspi.h`
- `fsl_flexio_qspi_smartdma.c/fsl_flexio_qspi_smartdma.h/fsl_samrtdma.c/fsl_smartdma.h`

MCX N947 uses:

- `fsl_flexio_qspi.c/fsl_flexio_qspi.h`
- `fsl_flexio_qspi_edma.c/fsl_flexio_qspi_edma.h`

Note: *`fsl_flexio_qspi.c/fsl_flexio_qspi.h` are compatible with both of RT595 and MCX-N947.*

4.1 Function description

The functions available in the drivers of the FlexIO QSPI examples are presented in the Table 3, Table 4 and Table 5.

Table 3. fsl_flexio_qspi.c or fsl_flexio_qspi.h

Function	Description
FLEXIO_QSPI_MasterGetDefaultConfig	Gets default configuration for FlexIO QSPI master
FLEXIO_QSPI_MasterInit	Initializes FlexIO module for FlexIO QSPI master
FLEXIO_QSPI_MasterTransferCreateHandle	Initializes the FlexIO QSPI master handle, which is used in Interrupt mode
FLEXIO_QSPI_MasterTransferNonBlocking	Starts transfer in Interrupt mode

Table 4. fsl_flexio_qspi_smartdma.c or fsl_flexio_qspi_smartdma.h

Function	Description
FLEXIO_QSPI_TransferCreateHandleSMARTDMA	Initializes the FlexIO QSPI master handle, which is used in SmartDMA mode
FLEXIO_QSPI_TransferSMARTDMA	Starts transfer in SmartDMA mode

Table 5. fsl_flexio_qspi_edma.c or fsl_flexio_qspi_edma.h

Function	Description
FLEXIO_QSPI_MasterTransferCreateHandleEDMA	Initializes the FlexIO QSPI master handle, which is used in eDMA mode
FLEXIO_QSPI_MasterTransferEDMA	Starts transfer in eDMA mode

4.2 Running the demos

This demo runs on i.MX RT595-EVK and MCX-N947-EVK. See Table 6.

Note:

- *The pinout used for the MCX-N947-EVK in this example is fully compatible with the MCX-N5XX-EVK and the FRDM-MCX-N947. Therefore, the example should run as-is, on any of the mentioned MCX evaluation platforms.*

Before downloading the program image to the MCU via J-link or CMSIS-DAP, ensure to connect the Quad SPI master and Flexcomm SPI slave signals on the same board. The connections must be done as shown in Figure 4.

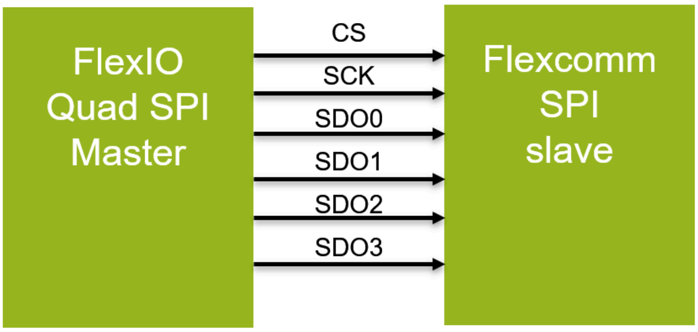


Figure 4. Wire connection

The FlexIO pins assignment for CS, SCK, SDO0, SDO1, SDO2, and SDO3 are shown in Table 6 and Table 7.

Table 6. Pin assignment for QuadSPI master

Pin assignment	i.MX-RT595-EVK	MCX-N947-EVK
CS	J28-2	J20-24
SCK	J28-1	J20-23
SDO0	J28-3	J20-25
SDO1	J28-4	J20-26
SDO2	J28-5	J20-27
SDO3	J28-6	J20-28

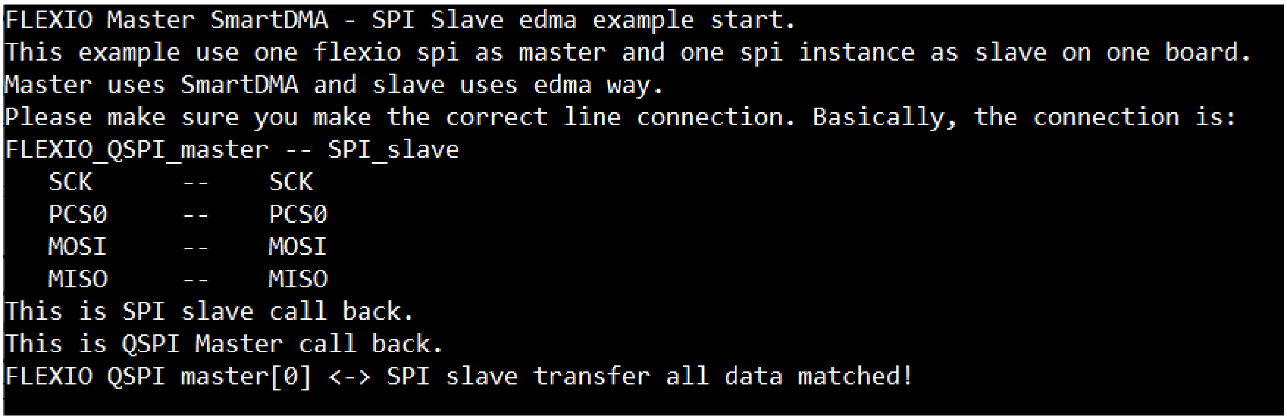
Table 7. Pin assignment for Flexcomm SPI slave

Pin assignment	i.MX RT595-EVK	MCX-N947-EVK
CS	JP26-1	J2-6
SCK	JP26-4	J2-12
MOSI	JP26-2	J2-10

Note: *In case the i.MX RT595-EVK board is used, you must disconnect JS23 1-2, and connect JS23-2 to JP23-3 to provide 1.8 V to VDDIO_3.*

SPI cannot receive all the bits SDO0-SDO3 at once, but can receive the SDOx bits one by one and validate them.

The validated result received on the debug console is shown in Figure 5. In this example, SDO0 is connected to SPI slave. It can be seen that SPI slave properly receives the data matching with SDO0.



```
FLEXIO Master SmartDMA - SPI Slave edma example start.
This example use one flexio spi as master and one spi instance as slave on one board.
Master uses SmartDMA and slave uses edma way.
Please make sure you make the correct line connection. Basically, the connection is:
FLEXIO_QSPI_master -- SPI_slave
   SCK      --    SCK
   PCS0     --    PCS0
   MOSI     --    MOSI
   MISO     --    MISO
This is SPI slave call back.
This is QSPI Master call back.
FLEXIO QSPI master[0] <-> SPI slave transfer all data matched!
```

Figure 5. Debug console when MOSI is connected to SDO0

5 Measurement by logic analyzer

Figure 6 and Figure 7 show the signal output measured by a logic analyzer. The output perfectly matches the requirement for the project. The communication protocol can be customized depending on the connected device used.

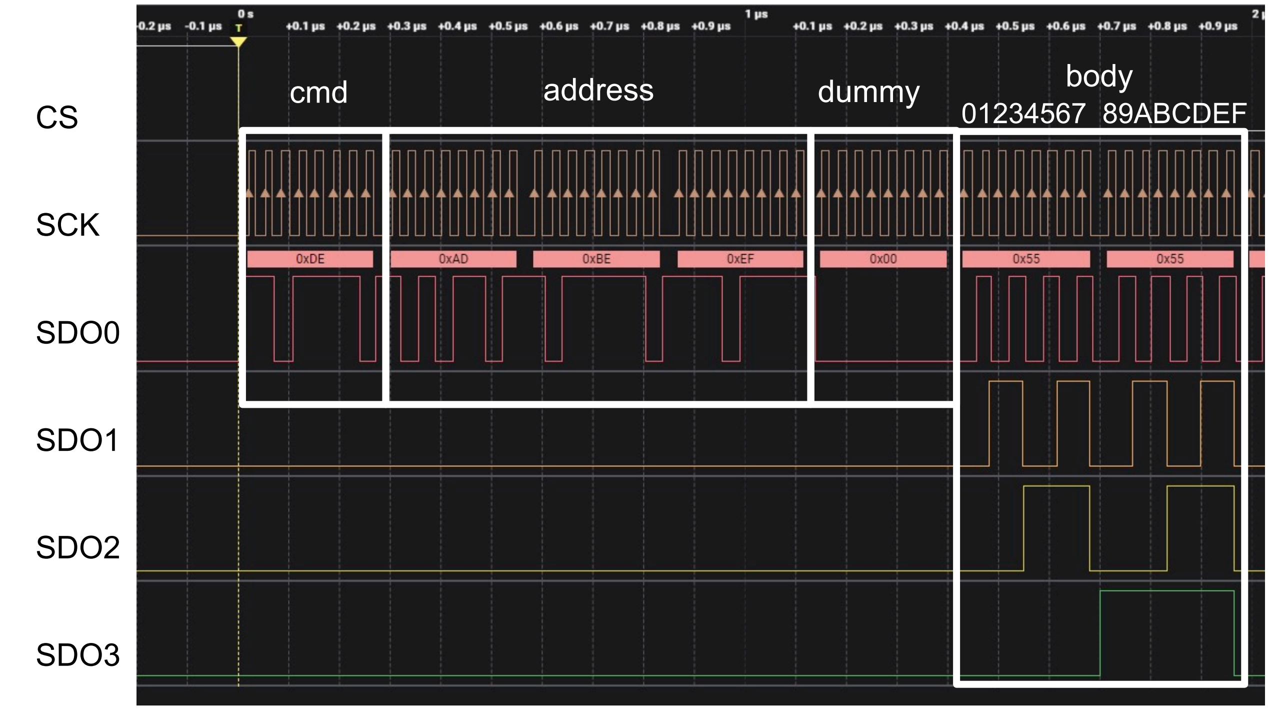


Figure 6. cmd + address + dummy + body

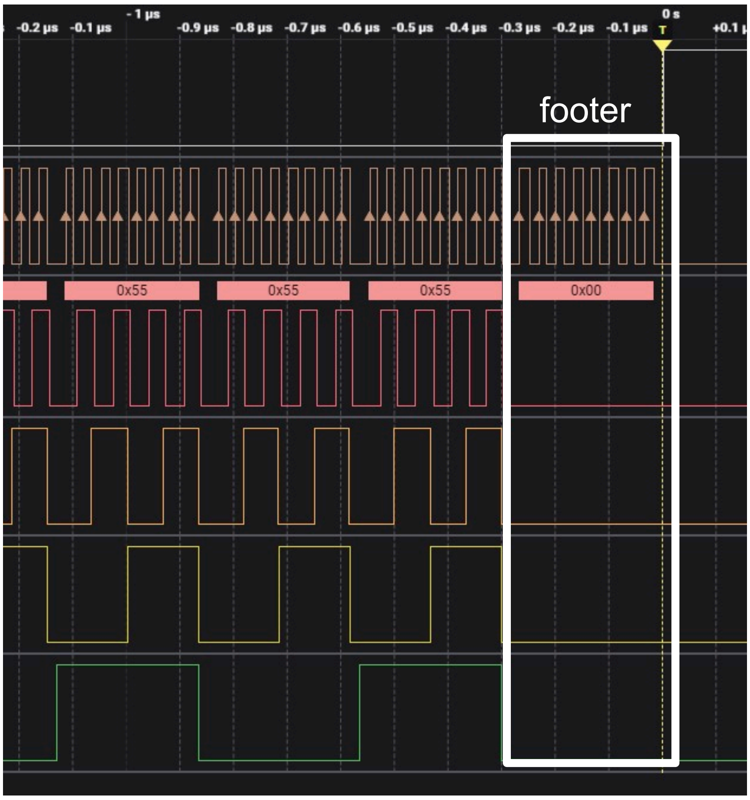


Figure 7. footer

6 Conclusion

This application note describes the implementation of simplex Quad SPI master on i.MX RT and MCX platforms. FlexIO is a flexible module that can be used to design not only common interfaces such as the SPI or I2C, but also proprietary interfaces by combining shifters and timers.

7 Related documentation

For additional information, refer to the documents available on the following URLs:

- https://www.nxp.com/design/design-center/development-boards/i-mx-evaluation-and-development-boards/i-mx-rt595-evaluation-kit:MIMXRT595-EVK#documentation
- https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/mcx-arm-cortex-m/mcx-n94x-and-n54x-mcus-with-dual-core-arm-cortex-m33-eiq-neutron-npu-and-edgelock-secure-enclave-core-profile:MCX-N94X-N54X#documentation
- https://community.nxp.com/t5/Kinetis-Microcontrollers/Understanding-FlexIO/ta-p/1115419
- AN12174 on nxp.com.
- Reference Manual of the respective device on nxp.com.

8 Acronyms

Table 8 lists the acronyms used in this document.

Table 8. Acronyms

Acronym	Description
DMA	Direct memory access
PWM	Pulse width modulation
SPI	Serial peripheral interface
QSPI	Quad serial peripheral interface

9 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

10 Revision history

Table 9 summarizes the revisions to this document.

Revision history

Document ID	Release date	Description
AN14175 v.1.0	20 January 2024	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.

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.

i.MX — is a trademark of NXP B.V.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

Kinetis — is a trademark of NXP B.V.

MCX — is a trademark of NXP B.V.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 20 January 2024
Document identifier: AN14175