AN15149

EnDat 2.2 Interface Implemented by FlexIO on MCX A366

Rev. 1.0 — 25 August 2026 Application note

Document information

Information	Content
Keywords	AN15149, MCX A366, EnDat 2.2, FlexIO, Encoder
Abstract	This application note describes a FlexIO-based EnDat 2.2 controller implemented on the FRDM-MCXA366.

1 Introduction

This application note describes a FlexIO-based EnDat 2.2 controller implemented on the FRDM-MCXA366. The project generates the encoder clock, sends a position command, controls the external RS-485 data transceiver, and captures the returned serial frame. It then decodes the multi-turn and single-turn position and validates the received CRC5.

The default target is encoder EQI1331_810662-03. The current profile uses a 60 MHz FlexIO source clock and a 15 MHz serial bit rate. It uses a 12-bit Multi-Turn (MT) field, a 19-bit single-turn field, one encoder error bit, and a 5-bit CRC. The implementation supports polling by default and also contains an interrupt-driven receive path.

1.1 Key features

- FlexIO0 generates TXD, CLK, and DIR while sampling RXD through a two-shifter receive chain.
- The selected EQI1331_810662-03 profile uses MT=12, ST=19, one ERR1 bit, and CRC5.
- The serial bit rate is configured for 15 MHz from a 60 MHz FlexIO root clock.
- RX capture is armed by the first RXD rising edge after DIR becomes low, matching the encoder Start transition.
- The parser searches the normalized 64-bit raw buffer for Start, extracts position fields, and checks CRC5.
- Serial output reports MT, ST, ERR1, and CRC status without RAW-buffer debug noise.

1.2 Scope

This note documents the current MCX A366 project and the validated receive-alignment method. It is not a substitute for the encoder or RS-485 transceiver data sheets. Encoder supply, differential termination, maximum clock rate, cable length, and setup/hold margin must be checked against the actual hardware.

2 Fundamentals

This chapter briefly introduces the basic building blocks of the protocol.

2.1 EnDat 2.2 physical interface

The MCU side of the interface uses four single-ended logic signals. An external RS-485 stage converts CLK to a differential clock pair and implements the bidirectional differential DATA pair. DIR controls the transition between controller command drive and encoder response receive.

Table 1. Logic signals

Signal	Function	State/Direction
TXD	Serial command data	MCX A366 to RS-485 interface
RXD	Receiver output	RS-485 interface to MCX A366
CLK	Continuous transaction clock	MCX A366 to differential clock driver
DIR	Data transceiver direction	High for command drive; low for response receives

2.2 Command and position transaction

A software write to SHIFTBUFBIS loads command 0x07 and triggers the FlexIO TX timer. The TX and DIR shifters emit the command phase while CLK runs. After the command window, DIR returns low, releasing the

data driver and enabling the receiver. The encoder then raises RXD for Start and shifts the position response under the continuing clock.

2.3 Position frame and CRC5

For the selected encoder profile, the normalized receive buffer contains one Start marker followed by one error bit, 31 position bits, and five CRC bits. Table 2 describes Least Significant Bit (LSB) oriented internal layout of parser this layout reflects FlexIO buffer normalization rather than connector-level bit drawing conventions.

Table 2. Protocol bits field

Normalized field	Width	Parser treatment
Start	1 bit	First set bit in the low 32 bits; removed before CRC processing
ERR1	1 bit	Included in CRC calculation and reported independently
ST	19 bits	Extracted first from the shifted LSB-oriented buffer
MT	12 bits	Extracted after ST
CRC5	5 bits	Bit-reversed after capture and compared with calculated CRC

CRC polynomial: P(x) = x^5 + x^3 + x + 1; initial value = 0x1F; calculated result = bitwise-complemented 5-bit remainder.

2.4 Role of FlexIO

FlexIO acts as the complete synchronous serial engine. It supplies the clock divider, command shifter, direction shifter, edge-triggered receive timer, and cascaded receive shifters. The Cortex-M33 only starts a transaction, waits for completion, reads two 32-bit words, and parses the completed frame.

2.5 EnDat 2.2 reference notes

The supplied EnDat 2.2 reference material confirms that the physical layer is RS-485 compatible: CLOCK is unidirectional from the controller to the encoder, while DATA is bidirectional. The protocol supports multiple mode commands, latches the position value, and transfers the response after direction turnaround.

- The reference timing notes list tst = 2 µs to 10 µs, tCAL <= 9 µs, and tm = 10 µs to 30 µs. Treat these as protocol-level guidance and verify the selected encoder's current data sheet for the final design.
- Position data is LSB-first while the received CRC field is Most Significant Bit (MSB) first. The CRC field ordering agrees with the driver's captured-CRC bit reversal before comparison.
- The 5-bit CRC uses P(x) = x^5 + x^3 + x + 1, an all-one preset, and a complemented final remainder.
- The FlexIO implementation uses four shifters (one TXD, one DIR, and two RXD) and three timers (TX/CLK, DIR, and RX), as summarized in Section 4.3.
- Cable length, clock rate, propagation-delay compensation, termination, and fail-safe bias remain system-level checks. Do not infer a qualified 15 MHz cable limit from the reference graph alone.

3 Hardware pinout and wiring

The following figure shows the intended signal path. The exact RS-485 transceiver part and connector pin numbers depend on the interface board. Connect by signal name and verify the interface board schematic before applying encoder power.

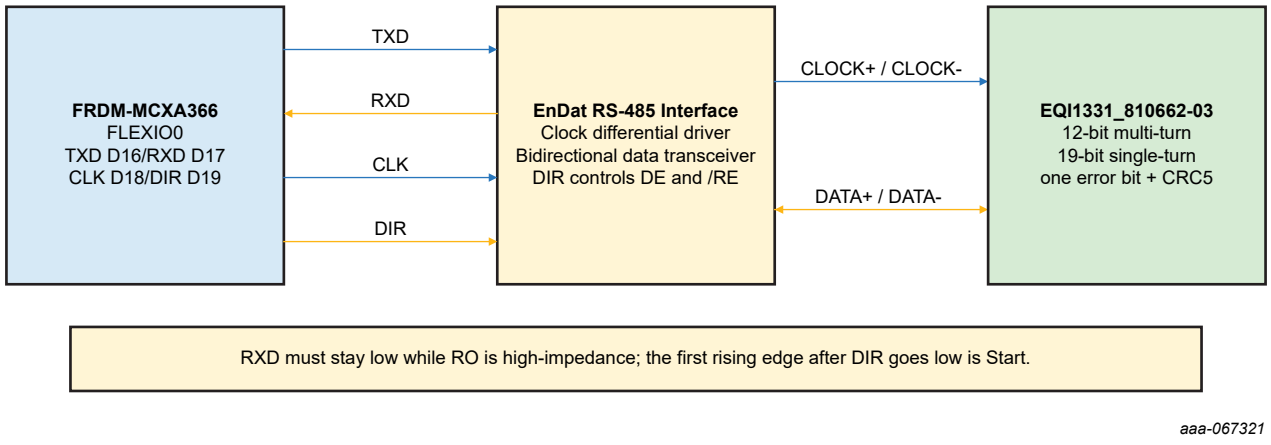

Figure 1. FRDM-MCXA366, EnDat 2.2 RS-485 interface, and encoder signal mapping

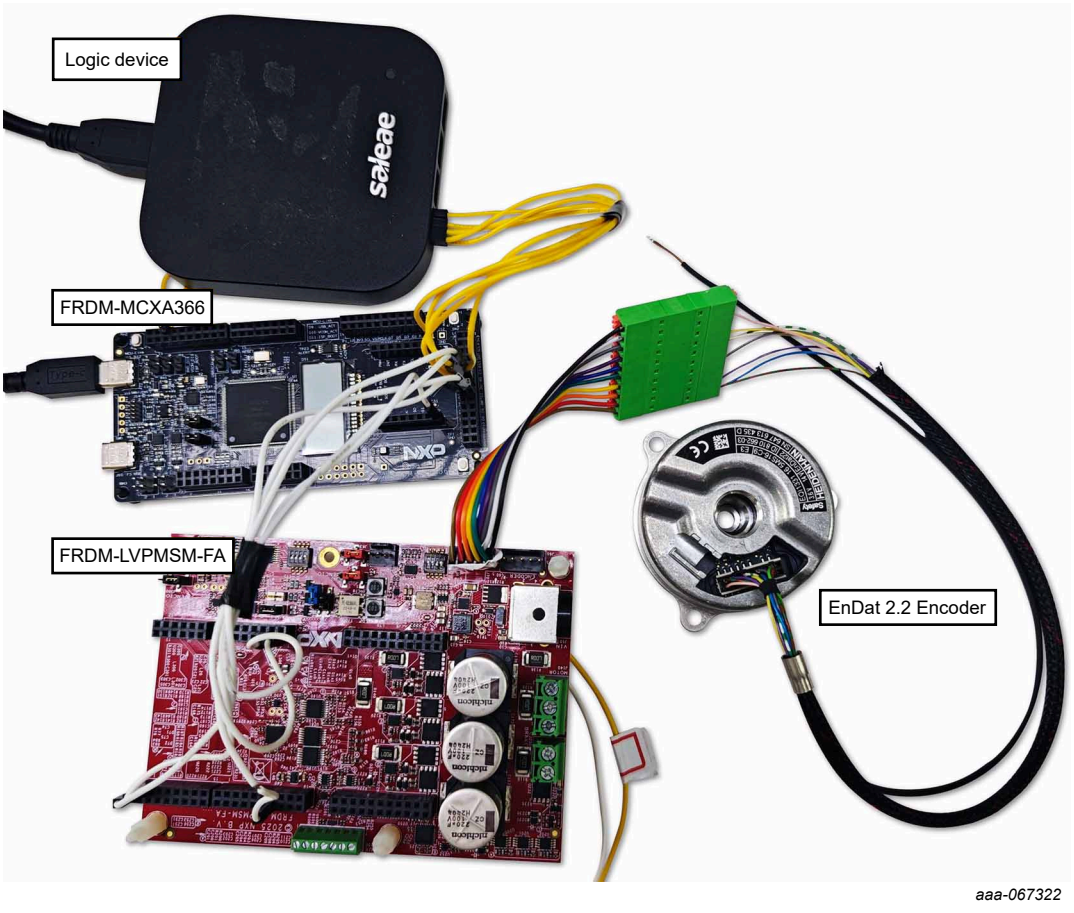

Figure 2. Hardware test setup with the FRDM-MCXA366, FRDM-LVPMSM-FA, EnDat 2.2 encoder, and logic analyzer

3.1 MCU-side signal mapping

Table 3 describes the MCU-side signal mapping.

Table 3. Signals mapping

Project signal	FlexIO signal	MCX A366 pad	Direction	Purpose
TXD	FLEXIO0_D16	P2_8	Output	Position command data
RXD	FLEXIO0_D17	P2_9	Input	Encoder response data
CLK	FLEXIO0_D18	P2_10	Output	15 MHz encoder clock
DIR	FLEXIO0_D19	P2_11	Output	RS-485 drive/receive control
GND	-	GND	Reference	Common logic ground

3.2 RS-485 direction and RX idle state

During the command phase, DIR is high. After the programmed command clocks, DIR becomes low and the interface releases the data driver so that the encoder can respond. When the receiver is disabled, its RO output is high-impedance; therefore, the logic-level RXD net must have a deterministic idle state.

- Remove the external 10 kΩ pull-up resistor from RXD on the validated interface hardware.
- Keep the MCX A366 internal pulldown enabled on P2_9/FLEXIO0_D17.
- Verify that RXD remains low while DIR is high and the receiver output is high-impedance.
- Treat the first RXD rising edge after DIR goes low as the encoder Start transition.

Pre-power check: Verify DIR polarity and confirm with a logic analyzer that RXD is low before the encoder Start edge. A high RXD idle level arms the RX timer too early and shifts every parsed field.

3.3 Power, ground, and differential wiring

- Connect encoder CLOCK+ and CLOCK- to the differential clock output of the interface board.
- Connect encoder DATA+ and DATA- to the bidirectional differential data channel.
- Use the encoder-rated supply and verify logic-side voltage compatibility with MCX A366 I/O.
- Use a low-impedance common ground between the FRDM board and interface logic side.
- Apply termination and fail-safe bias according to the encoder, transceiver, cable, and interface-board requirements.

3.4 Encoder profile and source traceability

Table 4 summarizes the encoder parameters used in this application and their corresponding code definitions.

Table 4. Encoder parameters

Item	Current record	Source/required check
Encoder	EQI1331_810662-03	board/app.h; verify nameplate and encoder data sheet
Bit rate	15,000,000 bit/s	DEMO_ENCODER_BIT_RATE; verify cable and transceiver margin
FlexIO clock	60,000,000 Hz	Runtime CLOCK_GetFlexioClkFreq() output
Multi-turn field	12 bits	DEMO_ENCODER_MT_LEN
Single-turn field	19 bits	DEMO_ENCODER_ST_LEN
Error field	One ERR1 bit	Current sensor behavior and parser implementation
CRC	5 bits	FLEXIO_ENDAT2_2_CRC_LEN and CRC routine
Termination/supply	Hardware-dependent	Verify against the encoder and interface data sheets

4 Software architecture

Figure 3 and Figure 4 summarize the current software-triggered implementation. They are derived from source/main.c, source/flexio_endat2_2.c, and board/app.h; they distinguish the default polling completion path from the optional RX interrupt path.

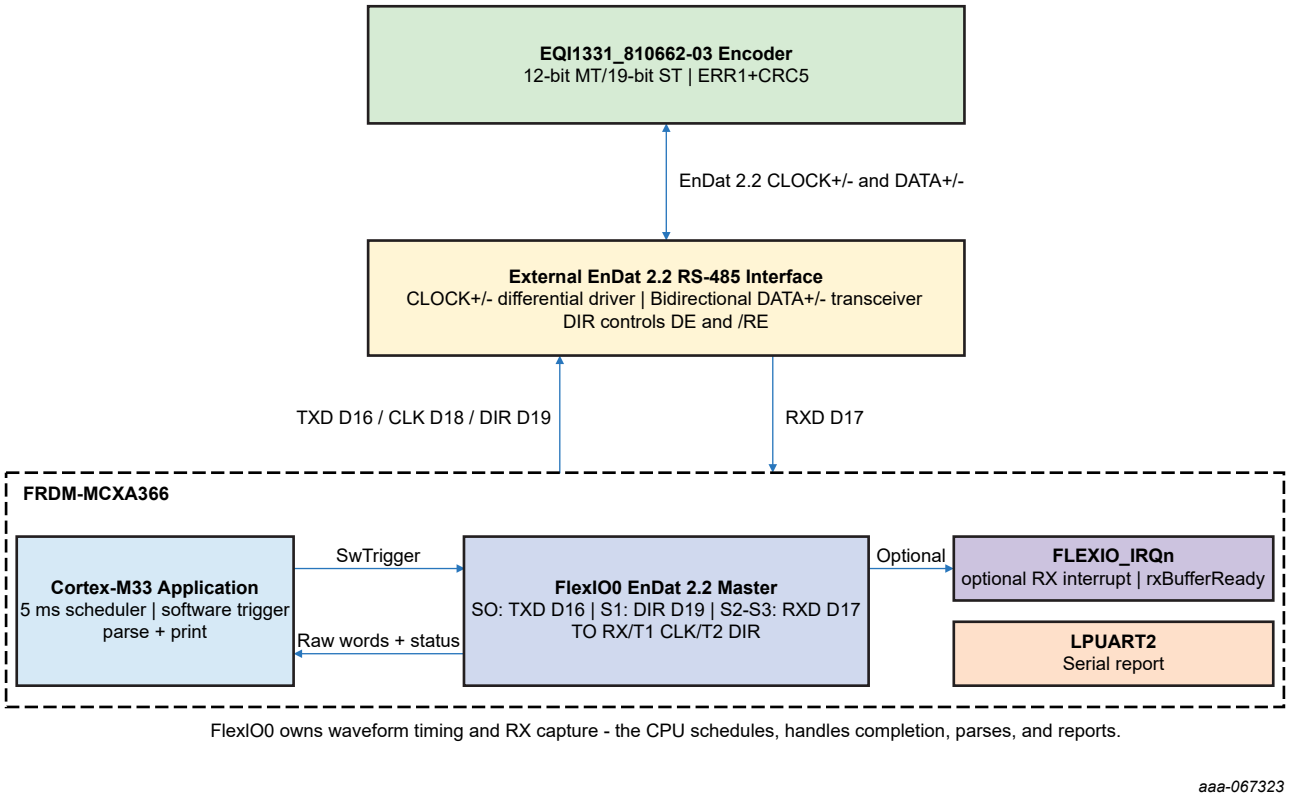

Figure 3. MCX A366 FlexIO EnDat 2.2 system block diagram

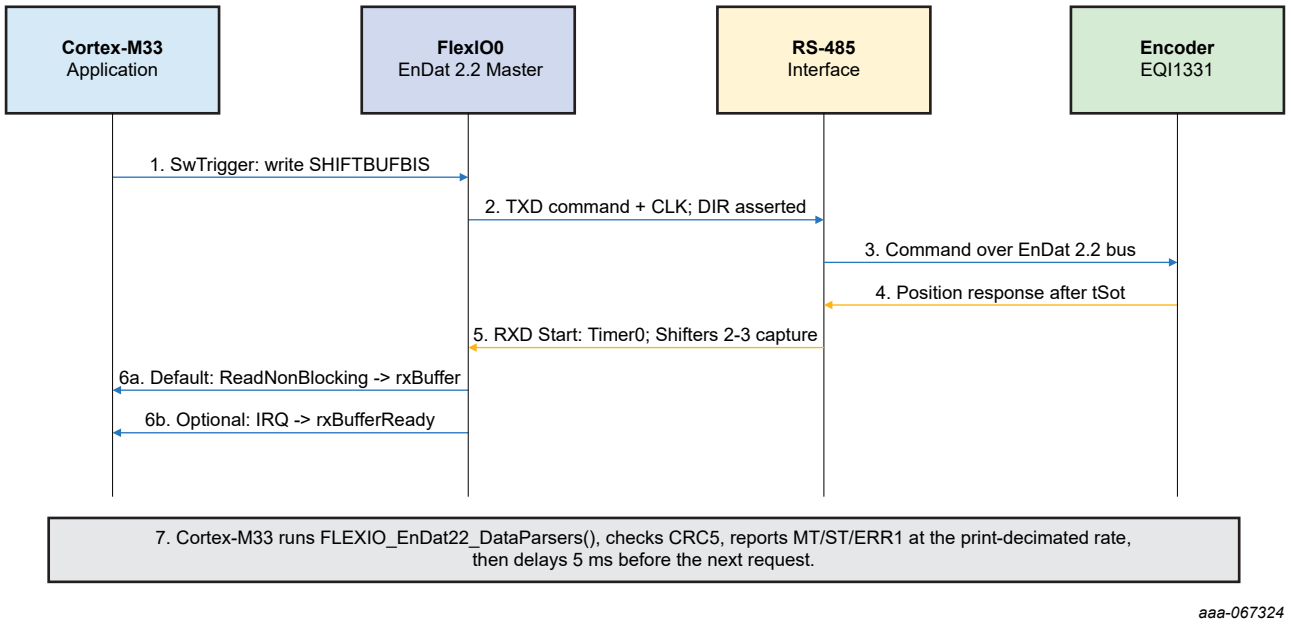

Figure 4. Position request, response capture, and completion sequence

4.1 Software modules

Table 5 describes the software modules and their responsibilities.

Table 5. Files description

File/module	Responsibility
source/main.c	Board and FlexIO clock initialization, command trigger loop, polling/interrupt selection, parser call, and serial output
source/flexio_endat2_2.c	FlexIO shifter/timer configuration, command presets, raw capture, frame parsing, CRC5, timeout, and ISR
source/flexio_endat2_2.h	Command constants, frame definitions, handle structure, and public driver API
board/app.h	Encoder profiles, selected bit rate and field lengths, FlexIO channels, sample period, and print decimation
board/pin_mux.c	P2_8 through P2_11 FlexIO muxing and RXD internal pulldown configuration
board/clock_config.*	The system clock tree used by the board initialization
drivers/fsl_flexio.*	MCUXpresso SDK FlexIO register-level configuration helpers
drivers/fsl_debug_console.*	115,200 bit/s diagnostic console output

4.2 Initialization and runtime flow

1. BOARD_InitHardware() initializes clocks, pins, the debug console, and board resources.
2. DEMO_InitFlexioClock() attaches the PLL1 divider clock, sets divider 1, enables FLEXIO0, and releases reset.
3. FLEXIO_Reset() clears the peripheral before protocol configuration.
4. FLEXIO_ENDAT2_2_Init() computes baudDiv, configures the receive-shifter count, and sets up three timers plus the TXD, DIR, and RXD shifters.
5. FLEXIO_ENDAT2_2_CmdPreset() selects the position command and its RX/DIR timer widths.

6. The main loop triggers a command, waits for completion, parses two raw words, rate-limits output, and delays 5 ms before the next request.

4.3 CPU and FlexIO resource allocation

The design assigns all bit-critical work to FlexIO0. The Cortex-M33 controls the transaction boundary and processes a frame only after FlexIO has captured it. This division keeps CPU instruction timing out of the 15 MHz serial clock path.

Table 6. Task for FlexIO and CPU

Execution owner	Assigned resource/phase	Responsibility	CPU loading implication
FlexIO0 hardware	Shifter 0 + Timer 1	Shift the 9-clock command and generate CLK	No CPU work is required for individual serial bits
FlexIO0 hardware	Shifter 1 + Timer 2	Drive DIR only during the command phase	Turnaround timing remains hardware-deterministic
FlexIO0 hardware	Timer 0 + Shifters 2-3	Start-edge aligned RXD capture into two raw words	The CPU is outside the sampling loop
Cortex-M33	main loop/trigger	Keep the 5 ms request schedule and start the transaction	Small control cost before each frame
Cortex-M33	ReadBlocking or optional IRQ	Default path polls shifter completion; IRQ path copies words and sets ready	Polling occupies the CPU while a frame is in flight; IRQ can reduce that busy-wait
Cortex-M33	DataParser + CRC5	Normalize Start, extract MT/ST/ERR1, and verify CRC	Short post-capture processing cost per request
Cortex-M33 + LPUART2	Rate-limited PRINTF	Emit decoded diagnostics every 100 samples	The largest variable software cost when a line is printed

CPU utilization is not instrumented in the current example, so this application note does not claim a percentage. At the configured 5 ms interval, the nominal request rate is 200 transactions per second. The FlexIO hardware handles the approximately 3.07 µs clocked transfer without requiring CPU bit servicing. However, the default ReadBlocking path busy-waits until the receive shifter reports completion. Encoder turnaround, Start latency, parser work, and serial output must be included in any measured CPU-load result.

5 Key implementation details

This section outlines some key details of the protocol implementation.

5.1 FlexIO clock and 15 MHz bit rate

The FlexIO dual-8-bit baud/bit timer uses the low compare byte as a half-period divider. With a 60 MHz source and a 15 MHz target, the driver calculates baudDiv as follows:

baudDiv = ((60,000,000 / 15,000,000) >> 1) - 1 = 1 bit rate = 60,000,000 / (2 x (baudDiv + 1)) = 15,000,000 bit/s

Table 7. Timer compare registers

Timer/field	High compare byte	Low compare byte	Result
TX timer	0xFF	0x01	The clock runs until the command/response transaction stops it
DIR timer	0x11	0x01	9 command clocks: (9 x 2 - 1) = 17
RX timer	0x49	0x01	37 post-Start bits: (37 x 2 - 1) = 73

5.1.1 Theoretical transaction waveform

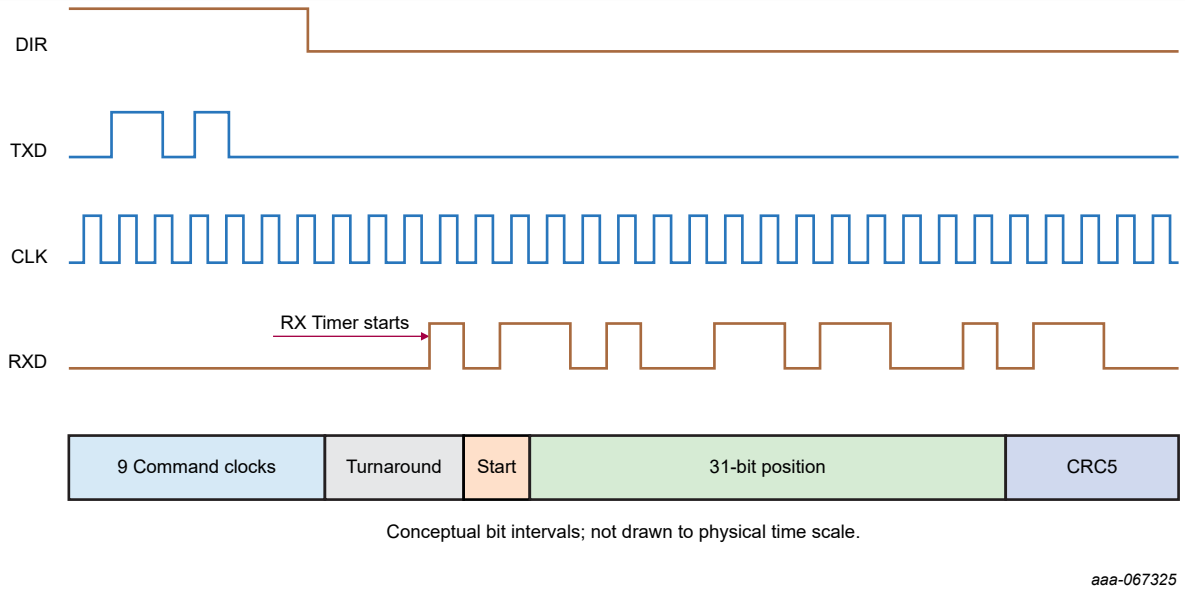

Figure 5. Conceptual FlexIO EnDat 2.2 command and position-response timing

5.1.2 FlexIO Register Configuration

The following field-level register records are distilled from the supplied configuration document. They describe the selected EnDat 2.2 position-command preset at a 60 MHz FlexIO source and a 15 MHz serial bit rate. Values that depend on the command preset must be rechecked when the encoder profile, frame length, or command changes.

Table 8. TXD command shifter: SHIFTER0

Register	Field	Bits	Value	Description
SHIFTCTL0	TIMSEL	[26:24]	1	Timer source is TIMER1
	TIMPOL	[23]	0	Shift on the configured timer edge
	PINCFG	[17:16]	3	Shifter pin configured as output
	PINSEL	[12:8]	16	D16/TXD
	PINPOL	[7]	0	Active-high polarity
	SMOD	[2:0]	2	Transmit mode
SHIFTCFG0	PWIDTH	[20:16]	0	Parallel width = 1
	INSRC	[8]	0	Input source configuration
	SSTOP	[5:4]	2	Stop bit configuration
	SSTART	[1:0]	1	Start bit configuration

Table 9. DIR control shifter: SHIFTER1

Register	Field	Bits	Value	Description
SHIFTCTL1	TIMSEL	[26:24]	2	Timer source is TIMER2
	TIMPOL	[23]	0	Shift on the configured timer edge

Table 9. DIR control shifter: SHIFTER1*...continued*

Register	Field	Bits	Value	Description
	PINCFG	[17:16]	3	Shifter pin configured as output
	PINSEL	[12:8]	19	D19/DIR
	PINPOL	[7]	0	Active-high polarity
	SMOD	[2:0]	2	Transmit mode
SHIFTCFG1	PWIDTH	[20:16]	0	Parallel width = 1
	INSRC	[8]	0	Input source configuration
	SSTOP	[5:4]	2	Stop bit configuration
	SSTART	[1:0]	0	No start bit

Table 10. RX receive-shifter chain: SHIFTER2

Register	Field	Bits	Value	Description
SHIFTCTL2	TIMSEL	[26:24]	0	Timer source is TIMER0
	TIMPOL	[23]	1	Sample on falling edge
	PINCFG	[17:16]	0	Shifter pin disabled
	PINSEL	[12:8]	17	D17/RXD, used as receive input
	PINPOL	[7]	0	Active-high polarity
	SMOD	[2:0]	1	Receive mode
SHIFTCFG2	PWIDTH	[20:16]	0	Parallel width = 1
	INSRC	[8]	1	Chained input from SHIFTER3
	SSTOP	[5:4]	0	Stop bit configuration
	SSTART	[1:0]	3	Start bit enabled

Table 11. RX receive-shifter chain: SHIFTER3

Register	Field	Bits	Value	Description
SHIFTCTL3	TIMSEL	[26:24]	0	Timer source is TIMER0
	TIMPOL	[23]	1	Sample on falling edge
	PINCFG	[17:16]	0	Shifter pin disabled
	PINSEL	[12:8]	17	D17/RXD, direct receive sample
	PINPOL	[7]	0	Active-high polarity
	SMOD	[2:0]	1	Receive mode
SHIFTCFG3	PWIDTH	[20:16]	0	Parallel width = 1
	INSRC	[8]	0	Direct input from D17
	SSTOP	[5:4]	0	Stop bit configuration
	SSTART	[1:0]	3	Start bit enabled

Table 12. RX timer: TIMER0, aligned by the RXD Start edge

Register	Field	Bits	Value	Description
TIMCTL0	TRGSEL	[29:24]	1	Internal trigger selection recorded as SHIFTER0
	TRGPOL	[23]	0/1	Trigger polarity is selected for the active receive path
	TRGSRC	[22]	1	Internal trigger source
	PINCFG	[17:16]	0	Timer pin disabled
	PINSEL	[12:8]	17	D17/RXD input
	TIMOD	[1:0]	1	Dual 8-bit baud/bit mode
TIMCFG0	TIMOUT	[27:24]	0	Output logic 1
	TIMDEC	[21:20]	0	Decrement on FlexIO clock
	TIMRST	[19:16]	Preset dependent	Timer reset condition
	TIMDIS	[11:8]	2	Disable on timer compare
	TIMENA	[10:8]	4	Enable on RXD pin rising edge
TIMCMP0	Baud Div	[7:0]	0x01	(1 + 1) x 2 = divide-by-4 from 60 MHz
	Bit Count	[15:8]	0x49	(73 + 1)/2 = 37 post-Start bits

Table 13. CLK generation timer: TIMER1

Register	Field	Bits	Value	Description
TIMCTL1	TRGSEL	[29:24]	1	Internal trigger selection recorded as SHIFTER0
	TRGPOL	[23]	0	Active trigger polarity
	TRGSRC	[22]	1	Internal trigger source
	PINCFG	[17:16]	3	Timer pin configured as output on D18
	PINSEL	[12:8]	18	D18/CLK
	TIMOD	[1:0]	1	Dual 8-bit baud/bit mode
TIMCFG1	TIMDEC	[21:20]	0	Decrement on FlexIO clock
	TIMENA	[10:8]	1	Enable on trigger
	TIMDIS	[3:0]	2	Disable after the TIMER0 stop signal
TIMCMP1	Baud Div	[7:0]	0x01	Divide-by-4 from 60 MHz
	Bit Count	[15:8]	0xFF	(255 + 1)/2 = 128 bits; effectively unlimited here

Table 14. DIR duration timer: TIMER2

Register	Field	Bits	Value	Description
TIMCTL2	TRGSEL	[29:24]	1	Internal trigger selection recorded as SHIFTER0
	TRGPOL	[23]	0	Active trigger polarity
	TRGSRC	[22]	1	Internal trigger source
	PINCFG	[17:16]	3	Timer pin configured as output on D19
	PINSEL	[12:8]	19	D19/DIR

Table 14. DIR duration timer: TIMER2*...continued*

Register	Field	Bits	Value	Description
	TIMOD	[1:0]	1	Dual 8-bit baud/bit mode
TIMCFG2	TIMDEC	[21:20]	0	Decrement on FlexIO clock
	TIMENA	[10:8]	1	Enable on trigger, synchronized with the transaction
	TIMDIS	[3:0]	2	Disable on timer compare
TIMCMP2	Baud Div	[7:0]	0x01	Divide-by-4 from 60 MHz
	Bit Count	[15:8]	0x11	(17 + 1)/2 = 9 command bits

5.2 RX start-edge synchronization

The RXD pin rising edge drives both timerEnable and timerReset for RX timer 0. This configuration makes the Start transition the receive timing reference. The hardware prerequisite is strict: RXD must stay low throughout the command phase and receiver-disable interval.

The original external 10 kΩ RXD pullup kept the MCU input near the high level when the transceiver RO output was high-impedance. That condition triggered receive timing before the encoder response. Removing the pullup and enabling the MCX A366 internal pulldown produced the required low idle state.

5.3 Receive width and shifter capture

The selected frame has 38 captured bits including Start: 1 Start + 1 ERR1 + 12 MT + 19 ST + 5 CRC. The RX timer counts the 37 bits after Start because the Start edge enables and aligns the timer. Two 32-bit receive shifters are chained to hold the response and any alignment padding.

rxShifterNum = ((MT + ST + 9) >> 5) + 1 = ((12 + 19 + 9) >> 5) + 1 = 2

5.4 Frame alignment, field parsing, and CRC5

FLEXIO_ENDAT2_2_DataParser() combines rxdBuffer[1:0] into one uint64_t, searches the low 32 bits for the first set bit, and treats it as Start. After shifting past Start, it calculates CRC across ERR1 plus the 31 position bits, extracts ST and MT from the normalized LSB-oriented buffer, reverses the captured CRC bit order with the Arm RBIT instruction, and compares both CRC values.

Table 15. CRC input and output

Parser step	Input	Output/failure mode
Find Start	Combined 64-bit raw buffer	First set bit in positions 0-31; otherwise return -1
CRC calculation	ERR1 + ST + MT, 32 bits	Calculated 5-bit complemented remainder
Field extraction	Shifted normalized buffer	ERR1, 19-bit ST, and 12-bit MT
CRC normalization	Captured 5-bit CRC	RBIT corrects captured bit order
Final status	Calculated and received CRC	crcMatch drives CRC=OK or CRC=FAIL

5.5 Timeout and diagnostic output

Polling mode waits for RX shifter status using a finite loop counter initialized to SystemCoreClock/200. On timeout, it prints RX timeout and stops the application, preventing a silent permanent wait when wiring, power, or timing is wrong. A completed frame that contains no Start prints No Endat2_2 start bit detected at the configured print-decimation rate.

Table 16.

Console result	Meaning	Primary checks
CRC=OK, ERR1=0	Valid frame and normal encoder status	Observe MT/ST behavior
CRC=OK, ERR1=1	Valid frame; encoder asserted ERR1	Inspect encoder operating conditions
CRC=FAIL	Frame captured but CRC disagrees	Bit rate, field lengths, polarity, signal integrity
No Endat2_2 start bit detected	Parser did not find Start	RX idle, DIR, receiver enable, DATA wiring
RX timeout	Receive shifter never completed	Encoder power, clock, Start edge, differential link

6 Main function reference

Table 17 provides the main functions and their descriptions.

Table 17. Main functions description

Function	File	Description
main()	source/main.c	Initializes the board and FlexIO, triggers periodic position requests, parses frames, and prints status
DEMO_InitFlexioClock()	source/main.c	Selects the PLL1 divider source, divider 1, FLEXIO0 gate, and reset release
FLEXIO_ENDAT2_2_Init()	flexio_endat2_2.c	Computes divider and resource count; configures shifters, timers, command presets, and ISR handle
FLEXIO_ENDAT2_2_CmdPreset()	flexio_endat2_2.c	Loads RX and DIR timer widths for the selected command
FLEXIO_ENDAT2_2_SwTrigger()	flexio_endat2_2.c	Writes command data to SHIFTBUFBIS and starts the transaction
FLEXIO_ENDAT2_2_ReadBlocking()	flexio_endat2_2.c	Waits with timeout, then copies the configured RX shifter words
FLEXIO_ENDAT2_2_DataParser()	flexio_endat2_2.c	Finds Start, extracts ERR1/ST/MT, calculates CRC5, and records crcMatch
FLEXIO_ENDAT2_2_EnableRx Interrupt()	flexio_endat2_2.c	Enables or disables the first receive shifter interrupt
FLEXIO_ENDAT2_2_IRQHandler()	flexio_endat2_2.c	Copies RX words in interrupt mode and sets rxdBuffer Ready
BOARD_InitPins()	board/pin_mux.c	Muxes D16-D19 and enables the P2_9 RXD internal pulldown

The public handle keeps protocol configuration and results together. When another encoder is selected, update the profile in board/app.h and verify that the total frame still fits the two-word receive buffer and available FlexIO shifters.

7 Build, flash, and run

This section provides hands-on instructions.

7.1 Development environment

- MCUXpresso IDE 25.6.136.

- Project: frdmmcxa366_flexio_endat2_2.
- Board: FRDM-MCXA366 with MCU-Link debug and serial connection.
- Encoder interface: external EnDat-compatible RS-485 stage and EQI1331_810662-03.

7.2 Build

To build the application and generate the axf file, perform the following steps:

1. Import or select frdmmcxa366_flexio_endat2_2 in MCUXpresso IDE.
2. Select the debug configuration and run build Project.
3. Confirm that Debug/frdmmcxa366_flexio_endat2_2.axf is generated without build errors.

7.3 Download and run

To set up the system and verify proper functionality, perform the following steps:

1. Connect the FRDM-MCXA366 MCU-Link USB port and open a 115200, 8-N-1 serial terminal with no flow control.
2. Verify the RS-485 interface wiring, RXD idle-low state, encoder supply, and differential clock/data connections.
3. Start a debug session or download the axf, then reset and run the target.
4. Confirm that the startup banner reports 15,000,000 bit/s, 60,000,000 Hz, MT=12, and ST=19.

7.4 Expected serial output

After the MCU is reset, the following output must appear on the serial console.

```
MCXA366 FlexIO Endat2_2 demo Encoder: EQI1331_810662-03, MT=12, ST=19,
 bitrate=15000000 bps FlexIO clock: 60000000 Hz Pins: TXD=D16, RXD=D17, CLK=D18,
 DIR=D19 MT=2750 ST=519896 ERR1=0 CRC=OK MT=2751 ST=201629 ERR1=0 CRC=OK
```

8 Results and observations

During receive-alignment bring-up, removing the RXD pullup and enabling the internal pulldown allowed the RX timer to start on the actual encoder Start transition. The parser, configured for 12-bit MT, 19-bit ST, one ERR1 bit, and CRC5, then produced stable CRC=OK frames. MT changed across multi-turn motion, while ST represented the within-turn position.

CRC=OK with ERR1=1 is a valid frame that carries an encoder error indication. The source is configured and build-verified for 15 MHz. Figure 6 displays an observed 15.625 MHz clock; reconcile this measured value with the active clock configuration before treating the 15 MHz timing target as validated.

The logic-analyzer capture in Figure 6 records D0 TX, D1 DIR, D2 CLK, and D3 RXD during a position transaction. The displayed CLK measurement is 15.625 MHz with a 50 % duty cycle; the trace also shows command transmission, DIR release, and the encoder RXD response.

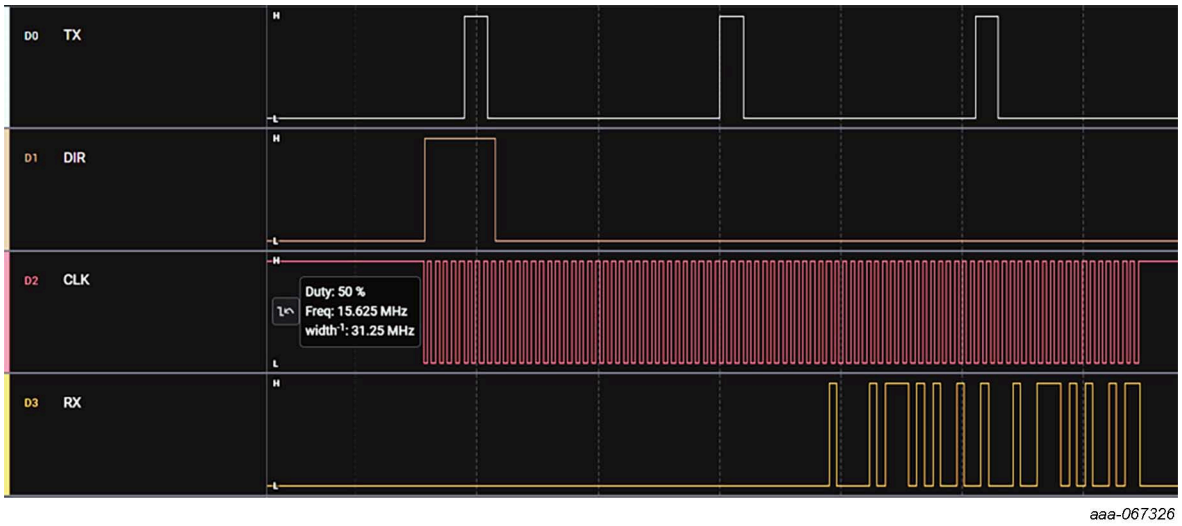

Figure 6. Logic-analyzer measurement of TX, DIR, CLK, and RXD during an EnDat 2.2 transaction

9 Conclusion

The frdmmcxa366_flexio_endat2_2 demo implements an EnDat 2.2 position-acquisition path using FlexIO resources and a small Cortex-M33 parser. FlexIO generates the command, clock, and direction timing and starts receive sampling on the encoder response edge. It then captures the frame in two shifters and supports polling or interrupt completion.

The decisive bring-up detail is the RXD idle state. Because the RS-485 receiver output becomes high-impedance while disabled, the MCU input must be held low so that the encoder Start transition is the first rising edge after DIR goes low. pullupWith the external pullup removed and the internal pulldown enabled, the parser can align the start bit and decode the 12-bit MT and 19-bit ST fields. It can also report the single ERR1 bit and validate CRC5.

This project is a practical baseline for MCXA366 EnDat 2.2 controller development. update theWhen changing the encoder, transceiver, cable, clock rate, or frame profile, update the *board/app.h* and recompute the timer and receive width. Verify the normalized field order and capture the physical waveform before accepting the CRC and position output as fully qualified.

10 Reference implementation

This MCX A366 application references the FlexIO-based EnDat 2.2 implementation for the NXP i.MX RT1180 platform. It is used as a functional reference for the protocol transaction sequence, FlexIO shifter and timer roles, Start-edge receive synchronization, frame handling, and CRC processing.

The MCX A366 project is adapted to FLEXIO0 and the FRDM-MCXA366 hardware. Its D16-D19 pin mapping, clock setup, board initialization, RS-485 interface control, and validation results are specific to this implementation.

11 Acronyms

Table 18 describes the acronyms used in this document.

Table 18. Acronyms

Acronym	Description
CRC	Cyclic Redundancy Check
FlexIO	Flexible Input/Output
IDE	Integrated Development Environment
IR	Interrupt Request
ISR	Interrupt Service Routine
LSB	Least Significant Bit
MSB	Most Significant Bit
MT	Multi-Turn
SDK	Software Development Kit

12 Note about the source code in the document

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

Copyright 2026 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

13 Revision history

Table 19 summarizes revisions to this document.

Table 19. Revision history

Document ID	Release date	Description
AN15149 v.1.0	25 August 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.

Tables

Figures

Contents