AN13005

ISO7816-3 FlexIO Driver Implementation on i.MXRT1060/ i.MXRT1064

Rev. 0 — 12/2020 Application Note

Contents

1 Introduction

ISO/IEC 7816 is a series of standards specifying integrated circuit cards and the use of such cards for interchange. These cards are identification cards intended for information exchange negotiated between the outside world and the integrated circuit in the card. As a result of an information exchange, the card delivers information, and/or modifies its content.

- ISO/IEC 7816-1 specifies physical characteristics for cards with contacts.
- ISO/IEC 7816-2 specifies dimensions and location of the contacts.
- ISO/IEC 7816-3 specifies electrical interface and transmission protocols for asynchronous cards.
- ISO/IEC 7816-4 specifies organization, security and commands for interchange.
- ISO/IEC 7816-5 specifies registration of application providers.
- ISO/IEC 7816-6 specifies interindustry data elements for interchange.
- ISO/IEC 7816-7 specifies commands for structured card query language.
- ISO/IEC 7816-8 specifies commands for security operations.
- ISO/IEC 7816-9 specifies commands for card management.
- ISO/IEC 7816-10 specifies electrical interface and answer to reset for synchronous cards.
- ISO/IEC 7816-11 specifies personal verification through biometric methods.
- ISO/IEC 7816-12 specifies electrical interface and operating procedures for USB cards.
- ISO/IEC 7816-13 specifies commands for application management in multi-application environment.
- ISO/IEC 7816-15 specifies cryptographic information application

This document covers the implementation of smart card interface based on FLEXIO module which follows the ISO/IEC 7816-3 standard. Driver covers *linklayer* of ISO/IEC 7816-3 only. There is no ISO7816 stack software support or other parts of ISO/IEC 7816-X cover. FLEXIO ISO7816-3 driver uses just one FLEXIO module. It is possible to run it on FLEXIO1, FLEXIO2, or FLEXIO3 module.

2 Interface

2.1 Signals description

Table 1. Signals description

Signal name	Pin SO28	Pin TSSOP16	Supply	Type	Description
AUX1C	27	1	VDD	I/O	Auxiliary data line to/from the host (internal 10 k pull-up resistor to VDD)

Table continues on the next page...

Table 1. Signals description (continued)

Signal name	Pin SO28	Pin TSSOP16	Supply	Type	Description
AUX2C	28	2	VDD	I/O	Auxiliary data line to/from the host (internal 10 k pull-up resistor to VDD)
VDD	6	3	VDD	Supply	Supply voltage
PRESN	9	4	VDD	I	Card presence contact input (active LOW); if PRESN is true, then the card is considered as present. A debouncing feature of 4.05 ms typ. is built in
I/O	11	5	VCC	I/O	Data line to/from the card (C7)(internal 10 k pull-up resistor to VCC)
AUX2	12	6	VCC	I/O	Auxiliary data line to/from the card (C8) (internal 10 k pull-up resistor to VCC)
AUX1	13	7	VCC	I/O	Auxiliary data line to/from the card (C4) (internal 10 k pull-up resistor to VCC)
GND	14	8	-	Supply	Ground
CLK	15	9	VCC	O	Clock to the card (C3)
RST	16	10	VCC	O	Card reset (C2)
VCC	17	11	VCC	O	Supply for the card (C1) (decouple to GND with 2x 220 nF capacitors with low ESR)
CMDVCCN	19	12	VDD	I	Start activation sequence input from the host (active LOW)
RSTIN	20	13	VDD	I	Card reset input from the host (active HIGH)
OFFN	23	14	VDD	O	NMOS interrupt to the host (active LOW) with 10 k internal pull-up resistor to VDD.
CLKIN	24	15	VDD	I	External clock
I/OUC	26	16	VDD	I/O	Host data I/O line (internal 10k pull-up resistor to VDD)
CLKDIV	1	NC	VDD	I	Control for choosing CLK frequency
TEST	4	NC	VDD	I	Test mode
PORADJ	18	NC	VDD	I	Input for VDD host supervisor. PORADJ threshold can be changed with an external R bridge
CS	21	NC	VDD	I	Chip select input from the host (active high)

2.2 FLEXIO clock

FLEXIO clock is provided by CCM to FLEXIO module. Using MCUXpresso Clock tool is recommended to get better clock distribution overview.

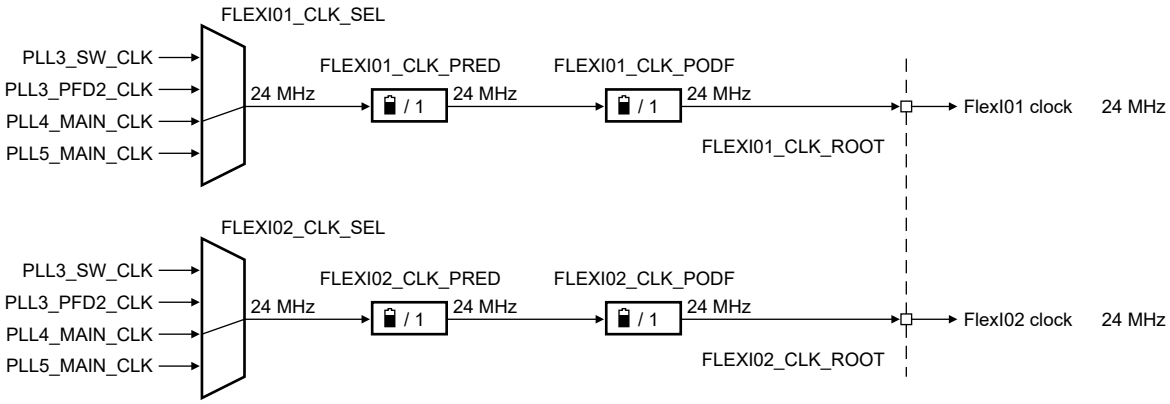

Figure 1. Clock assignments for FLEXIO1, FLEXIO2, and FLEXIO3 modules

NOTE

FLEXIO3 shares clock with FLEXIO2 module.

Interface function:

```
/*
* Mux 0 - PLL4_MAIN_CLK
* Mux 1 - PLL3_PFD2_CLK
* Mux 2 - PLL5_MAIN_CLK
* Mux 3 - PLL3_SW_CLK
*
* PreDiv 0 - divide by 1
* PreDiv 1 - divide by 2
* PreDiv 2 - divide by 3
* PreDiv 3 - divide by 4
* PreDiv 4 - divide by 5
* PreDiv 5 - divide by 6
* PreDiv 6 - divide by 7
* PreDiv 7 - divide by 8
*
* Div 0 - divide by 1
* Div 1 - divide by 2
* Div 2 - divide by 3
* Div 3 - divide by 4
* Div 4 - divide by 5
* Div 5 - divide by 6
* Div 6 - divide by 7
* Div 7 - divide by 8
*
* Return value == 0 - Err
* Return value != 0 - New Flexio freq
*/
uint32_t FLEXIO_UART7816_SetFlexioFreq(FLEXIO_UART7816_Type *base, uint32_t Mux, uint32_t PreDiv,
uint32_t Div)
```

2.3 Output control module

Output control module is responsible for processing of input signal OFFN and also for the timing of output signals CLK_OUT, RSTIN, and CMDVCCN.

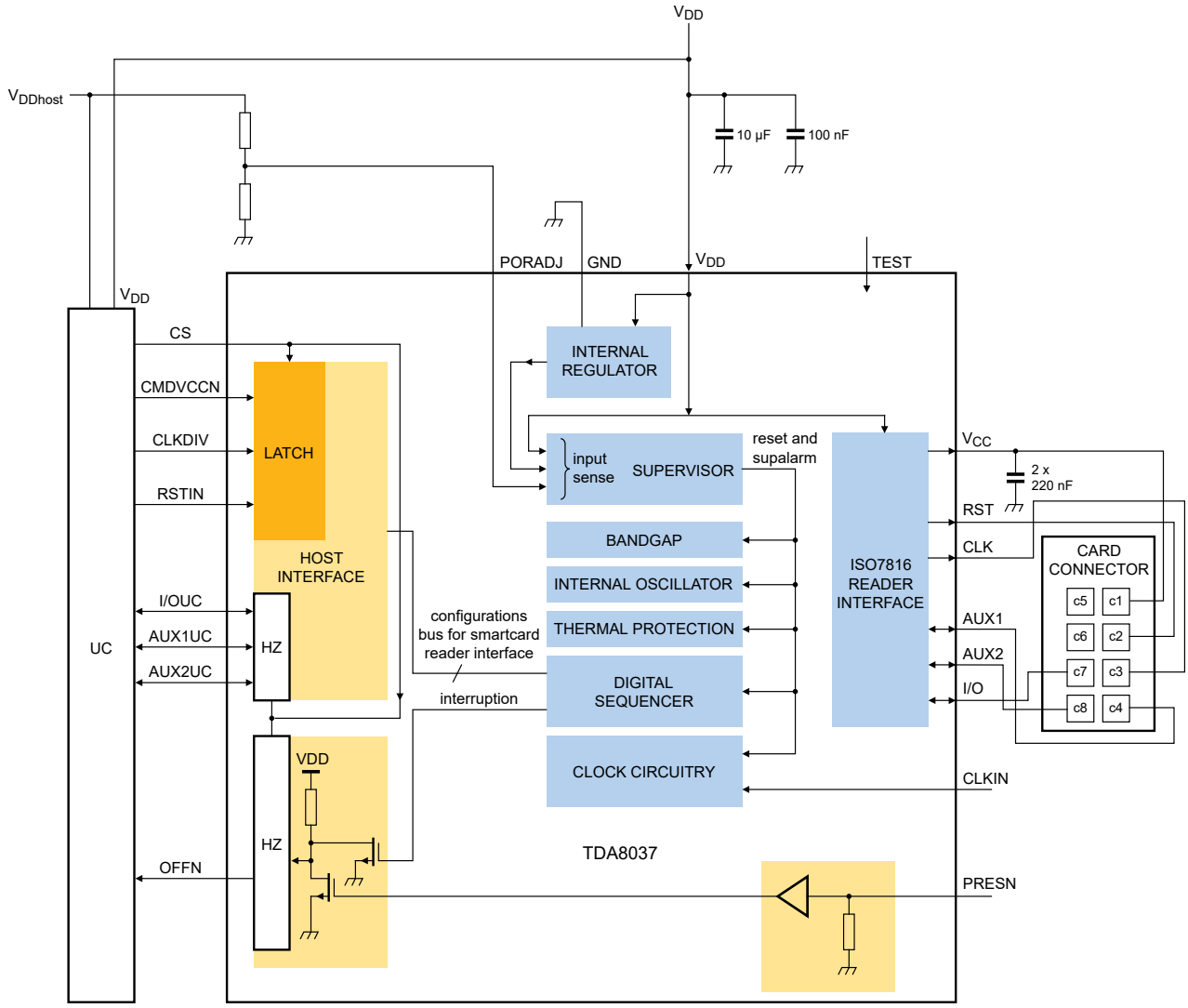

Figure 2. Block diagram of interface TDA8037TT

2.3.1 CLK_OUT

Set the smart card clock CLK_OUT for TDA8037:

```
uint32_t FLEXIO_UART7816_Card_CLK_START( FLEXIO_UART7816_Type *base, flexio_uart7816_handle_t
*handle, uint32_t clk)
```

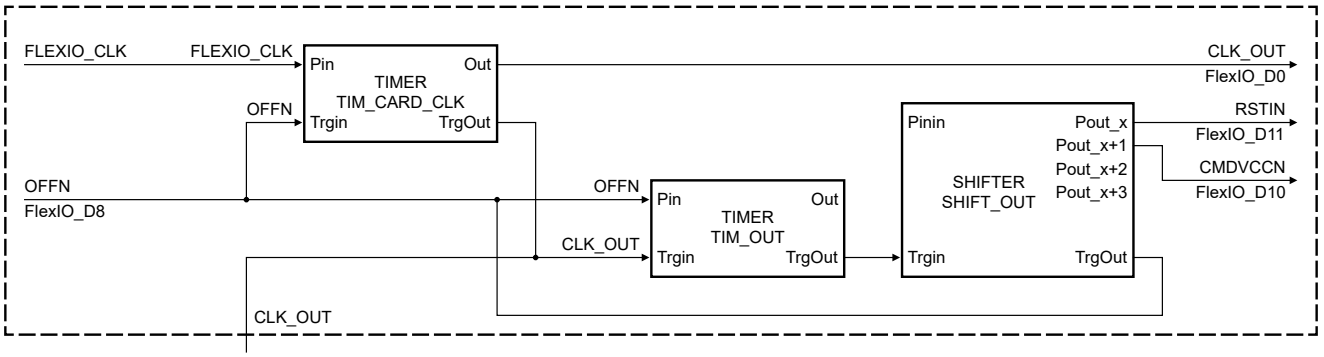

Figure 3. Output control module

TIM_CARD_CLK controls CLK_OUT. According to ISO7816-3 5.2.3 chapter CLK specification, the duty cycle of the clock signal must be between 40 % to 60 % during stable operation. The minimum value should be 1 MHz. And during activation and cold reset, the maximum value should be 5 MHz.

2.3.2 Default state of RSTIN and CMDVCCN

RSTIN and CMDVCCN are FLEXIO signals (not GPIOs). The dedicated function used to set the default value is:

```
void FLEXIO_UART7816_OUT(FLEXIO_UART7816_Type *base, uint32_t val)
```

- val bit 0 - CMDVCCN
- val bit 1 - RSTIN

2.3.3 Reset and activation procedure

To assure correct signal timing for reset and activation procedure, timer TIM_OUT and shifter SHIFT_OUT are used.

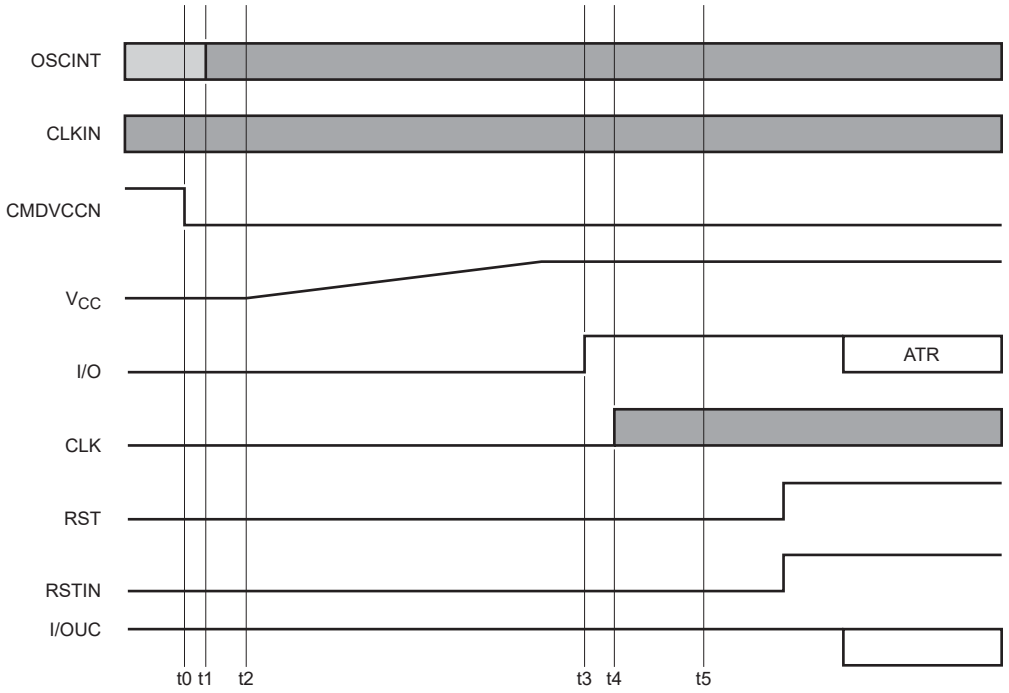

Figure 4. Activation procedure as defined in ISO7816-3

NOTE

TDA8037 handles the VCC signal.

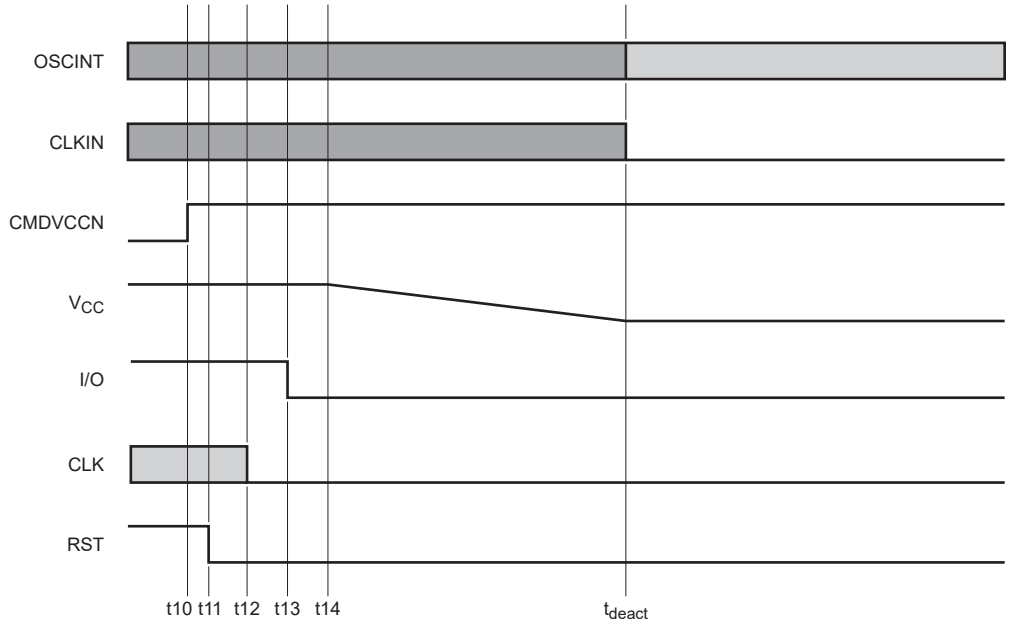

Figure 5. Deactivation procedure as defined in ISO7816-3

NOTE

TDA8037 handles the VCC signal.

You can set any output signals sequence as reaction to OFFN input signal in timeslots controlled by TIM_OUT. CLK_OUT/64 defines the TIM_OUT timeslots.

```
void FLEXIO_UART7816_OUT_ON_OFFN( FLEXIO_UART7816_Type *base, uint8_t pinpol, uint32_t val)
```

- val bit 0 - CMDVCCN
- val bit 1 – RSTIN

```
FLEXIO_UART7816_OUT_ON_OFFN( &uartDev, 1, 0x20000001);
```

This function sets the TIM_OUT to wait for OFFN to go high (card is inserted) and start counting in CLK_OUT/64 timeslots. CMDVCCN goes to low and after 7 timeslots RSTIN goes high. Answer to Reset data block (ATR) can be received.

After, 7 timeslots TIM_OUT interrupt is executed and new sequence is prepared for card insertion or removal. More details are available in the TransferHandleIRQ interrupt handler.

2.4 Timeouts and RX/TX module

Timeouts and RX/TX module is responsible for transmitting and receiving characters. Timeout events are issued according to Character Guard Time (CGT), Block Guard Time (BGT) or Character Waiting Time (CWT), Block Waiting Time (BWT) timeouts setting.

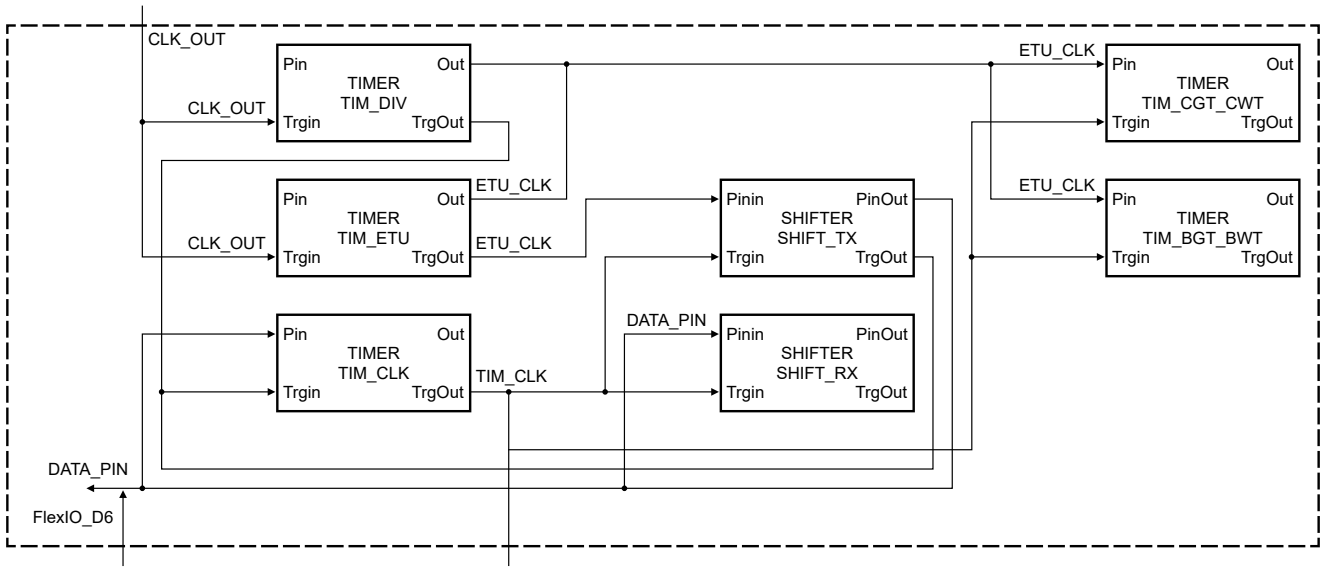

Figure 6. Timeouts and RX/TX module

2.4.1 Receive data

```
void FLEXIO_UART7816_SwitchToRx(FLEXIO_UART7816_Type *base, flexio_uart7816_handle_t *handle)
```

RX mode is set according to handle->state.

```
If handle->state==kState_UART7816_T0_RX then the T0 RX mode is set.
If handle->state==kState_UART7816_T1_RX then the T1 RX mode is set.
```

Received character is passed to the application code by callbacks:

```
handle->call_RxDataT0 = FLEXIO_UART7816_RxData_UserCallback;
handle->call_RxDataT1 = FLEXIO_UART7816_RxData_UserCallback;
```

In case of T0 receive mode parity bit is checked in parity evaluation module and ACK/NACK is issued according to ISO7816-3 T0 mode error signal specification.

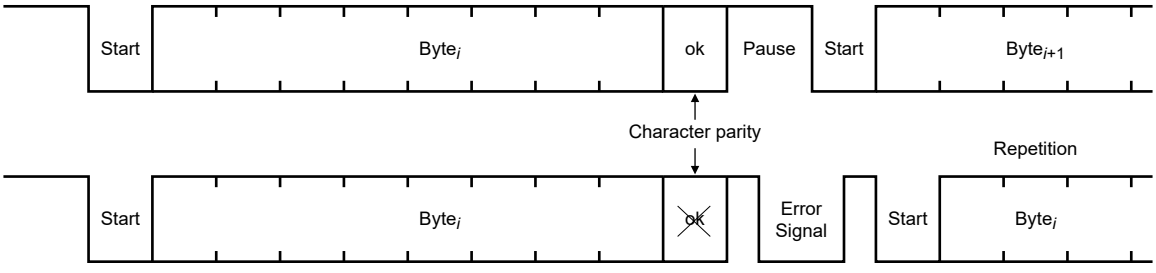

Figure 7. Character transmission and repetition in T0 mode

In case of process handle->call_RxDataT0 callback the parameters handle->ch, handle->par, handle->ack can be used. The received character is located in handle->ch. The information if ACK or NACK was transmitted by parity evaluation module is located in handle->ack parameter. In case of handle->ack==0 means that NACK was sent and character should be discarded because it is repeated by card (in T0 RX mode).

In T1 RX mode parity evaluation module is disabled. Application code is responsible for either parity bit or Longitudinal Redundancy Code (LRC)/Cyclic Redundancy Code (CRC) checking.

2.4.2 Transmit data

To send block of data in T0 mode use:

```
uint8_t blockData[]="123";
g_uartHandle.pDataTX=blockData;
g_uartHandle.dataSizeTX=sizeof(blockData)-1;
result = FLEXIO_UART7816_Transfer_T0_Block( &uartDev, &g_uartHandle);
```

After each character is sent, the error signal is checked. If error signal (NACK) is received, then character is repeated as specified in handle->pConfig->nrRepeats parameter. The driver is responsible for character repetition.

When character is transmitted, the CGT timeout is restarted. Next character is sent when CGT timeout is expired. When last character is sent, driver switches to T0 RX mode.

To send block of data in T1 mode use:

```
uint8_t blockData[]="123";
g_uartHandle.pDataTX=blockData;
g_uartHandle.dataSizeTX=sizeof(blockData)-1;
result = FLEXIO_UART7816_Transfer_T0_Block( &uartDev, &g_uartHandle);
```

When character is transmitted, the CGT timeout is restarted. Next character is sent when CGT timeout is expired. When last character is sent, driver switches to T1 RX mode.

2.4.3 Timeouts handling

The default values of all timeouts are specified in FLEXIO_UART7816_GetDefaultConfig function. Default values are set according to ISO7816-3:

```
userConfig->CGT_ETU = 12;
userConfig->CWT_ETU = 9600;
userConfig->BGT_ETU = 22;
userConfig->BWT_ETU = 25920;
```

All timeouts are specified in Elementary Time Unit (ETU). Timeouts are restarted when FLEXIO_UART7816_SwitchToRx or FLEXIO_UART7816_SwitchToTx function is executed. Timeout signals are passed to application code using callbacks.

```
flexio_uart7816_transfer_callback_t call_CGT_T0_expired;
flexio_uart7816_transfer_callback_t call_CWT_T0_expired;
flexio_uart7816_transfer_callback_t call_CGT_T1_expired;
flexio_uart7816_transfer_callback_t call_CWT_T1_expired;
flexio_uart7816_transfer_callback_t call_BGT_T1_expired;
flexio_uart7816_transfer_callback_t call_BWT_T1_expired;
```

The application code is responsible for appropriate reaction to any timeouts. The description of timeouts is conformed to ISO7816-3.

All callbacks are executed from interrupt handler. Actions executed in callbacks must be quick and re-entrant.

2.5 Parity evaluation module

This module is responsible for parity evaluation in RX character. Module is initialized in:

```
status_t FLEXIO_UART7816_Init(FLEXIO_UART7816_Type *base, flexio_uart7816_handle_t *handle, uint32_t
srcClock_Hz)
```

There are no further actions needed by application code.

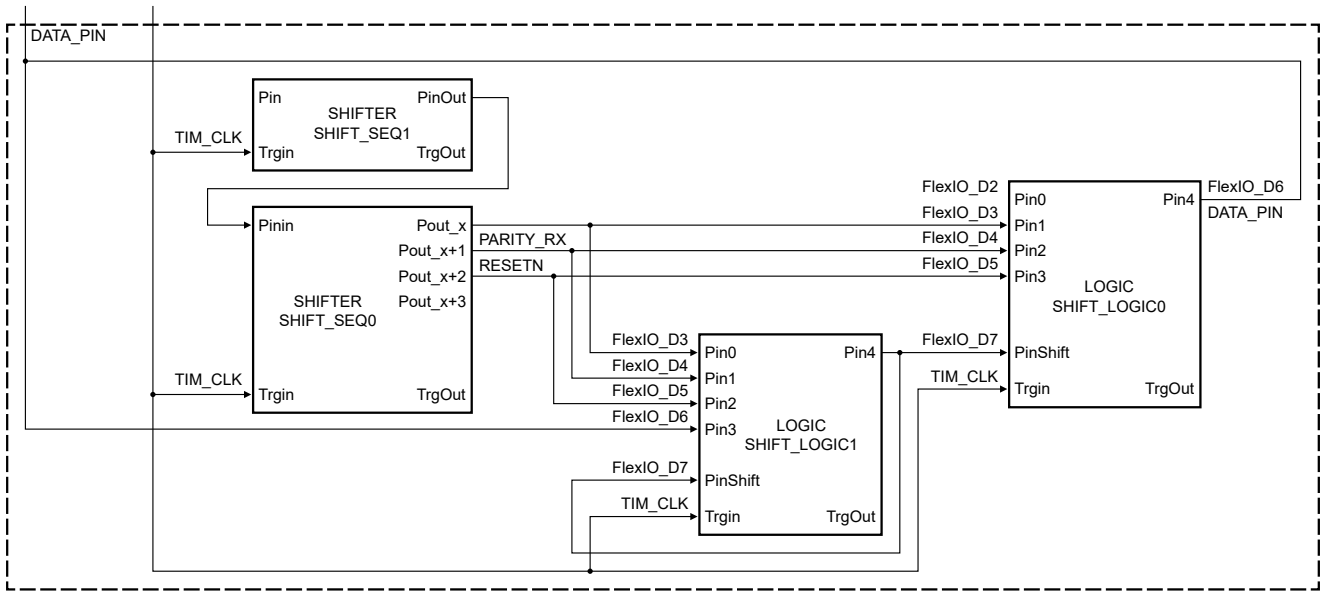

Figure 8. Parity evaluation module

Table 2. SHIFT_LOGIC1

Line	LastState FLEXIO_D5	DATA_PIN	RESETN	Par_RXN		Out Parity EVEN	Out Parity OD	Description
	PinShift	FLEXIO_D6	FLEXIO_D5	FLEXIO_D4	FLEXIO_D3	FLEXIO_D7	FLEXIO_D7	
0	0	0	0	0	0	1	0	RESETN
1	0	0	0	0	1	1	0	RESETN
2	0	0	0	1	0	1	0	RESETN
3	0	0	0	1	1	1	0	RESETN
4	0	0	1	0	0	0	0	PAR_RX request => Out=LS to block change
5	0	0	1	0	1	0	0	PAR_RX request => Out=LS to block change
6	0	0	1	1	0	0	0	LS=0 ^ DATA=0 => Out=0
7	0	0	1	1	1	0	0	LS=0 ^ DATA=0 => Out=0
8	0	1	0	0	0	1	0	RESETN
9	0	1	0	0	1	1	0	RESETN
10	0	1	0	1	0	1	0	RESETN
11	0	1	0	1	1	1	0	RESETN
12	0	1	1	0	0	0	0	Par_RX request => Out=LS to block change

Table continues on the next page...

Table 2. SHIFT_LOGIC1 (continued)

Line	LastState FLEXIO_D5	DATA_PIN	RESETN	Par_RXN		Out Parity EVEN	Out Parity OD	Description
13	0	1	1	0	1	0	0	Par_RX request => Out=LS to block change
14	0	1	1	1	0	1	1	LS=0 ^ DATA=1 => Out=1
15	0	1	1	1	1	1	1	LS=0 ^ DATA=1 => Out=1
16	1	0	0	0	0	1	0	RESETN
17	1	0	0	0	1	1	0	RESETN
18	1	0	0	1	0	1	0	RESETN
19	1	0	0	1	1	1	0	RESETN
20	1	0	1	0	0	1	1	Par_RX request => Out=LS to block change
21	1	0	1	0	1	1	1	Par_RX request => Out=LS to block change
22	1	0	1	1	0	1	1	LS=1 ^ DATA=1 => Out=1
23	1	0	1	1	1	1	1	LS=1 ^ DATA=1 => Out=1
24	1	1	0	0	0	1	0	RESETN
25	1	1	0	0	1	1	0	RESETN
26	1	1	0	1	0	1	0	RESETN
27	1	1	0	1	1	1	0	RESETN
28	1	1	1	0	0	1	1	Par_RX request => Out=LS to block change
29	1	1	1	0	1	1	1	Par_RX request => Out=LS to block change
30	1	1	1	1	0	0	0	LS=1 ^ DATA=1 => Out=0
31	1	1	1	1	1	0	0	LS=1 ^ DATA=1 => Out=0

Table 3. SHIFT_LOGIC0

Line	LastState FLEXIO_D5	RESETN	Par_RXN	Par_TXN	TX/RXN	DATA_PN	Description
	PinShift	FLEXIO_D5	FLEXIO_D4	FLEXIO_D3	FLEXIO_D2	FLEXIO_D4	
0	0	0	0	0	0	1	RESETN
1	0	0	0	0	1	1	RESETN
2	0	0	0	1	0	1	RESETN
3	0	0	0	1	1	1	RESETN
4	0	0	1	0	0	1	RESETN

Table continues on the next page...

Table 3. SHIFT_LOGIC0 (continued)

Line	LastState FLEXIO_D5	RESETN	Par_RXN	Par_TXN	TX/RXN	DATA_PN	Description
5	0	0	1	0	1	1	RESETN
6	0	0	1	1	0	1	RESETN
7	0	0	1	1	1	1	RESETN
8	0	1	0	0	0	1	ERROR
9	0	1	0	0	1	1	ERROR
10	0	1	0	1	0	0	Par_RX request for RX => Out=Parity
11	0	1	0	1	1	0	Par_RX request for RX => Out=Parity
12	0	1	1	0	0	1	ERROR
13	0	1	1	0	1	1	ERROR
14	0	1	1	1	0	1	No Parity request
15	0	1	1	1	1	1	No Parity request
16	1	0	0	0	0	1	RESETN
17	1	0	0	0	1	1	RESETN
18	1	0	0	1	0	1	RESETN
19	1	0	0	1	1	1	RESETN
20	1	0	1	0	0	1	RESETN
21	1	0	1	0	1	1	RESETN
22	1	0	1	1	0	1	RESETN
23	1	0	1	1	1	1	RESETN
24	1	1	0	0	0	1	ERROR
25	1	1	0	0	1	1	ERROR
26	1	1	0	1	0	1	Par_RX request for RX => Out=Parity
27	1	1	0	1	1	1	Par_RX request for RX => Out=Parity
28	1	1	1	0	0	1	ERROR
29	1	1	1	0	1	1	ERROR
30	1	1	1	1	0	1	No Parity request
31	1	1	1	1	1	1	No Parity request

3 SDK integration

3.1 Description

FLEXIO ISO7816-3 driver consists of two files:

- *fsl_flexio_7816uart.h*
- *fsl_flexio_7816uart.c*

Driver is built on top of standard SDK FLEXIO driver:

- *fsl_flexio.h*
- *fsl_flexio.c*

The file *flexio_uart_interrupt_transfer.c* is considered to be a simple example of usage.

3.2 Installation

Driver was developed using MCUXpresso IDE v11.0.1 with installed SDK_2.x_MIMXRT1064xxxA 2.6.1.

The recommended way to use FLEXIO ISO7816-3 driver is to download any FLEXIO SDK (for example flexio_uart_interrupt_transfer) and put FLEXIO ISO7816 driver files into the board, source, and driver directory.

4 References

- INTERNATIONAL STANDARD ISO/IEC 7816-3, Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols
- *Low power 3V smart card interface* (document TDA8037)
- *Design guideline for TDA8037* (document AN11458)

5 Definitions and acronyms

- APDU - Application Protocol Data Unit
- ATR - Answer to Reset data block
- CWT - Character Waiting Time
- CGT - Character Guard Time
- BWT - Block Waiting Time
- BGT - Block Guard Time
- CRC - Cyclic Redundancy Code
- LRC - Longitudinal Redundancy Code
- ETU - Elementary Time Unit
- T=0 - Half duplex transmission of characters
- T=1 - Half duplex transmission of blocks

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 12/2020

Document identifier: AN13005

arm