AN12780
Emulating SPI with the FlexIO on I.MXRT Series MCU

Rev. 0 — 13 March 2020

Application Note

Contents

1 Introduction

This application note illustrates how to use FlexIO module to emulate both SPI master and SPI slave mode based on I.MXRT series platform.

FlexIO is an on-chip peripheral available on NXP I.MXRT series. It is a highly configurable module that is capable of emulating a wide range of communication protocols, such as UART, I2C, SPI, I2S, and so on.

This application creates a simple software demo based on the I.MXRT1010 platform for users to use FlexIO module emulating SPI Master and Slave with related configurations.

2 FlexIO Overview

The FlexIO module of the i.MX RT1010 provides the following key features:

- Array of 32-bit shift registers with transmit, receive, and data match modes.
- Double buffered shifter operation for continuous data transfer.
- Automatic start/stop bit generation.
- Interrupt, DMA, or polled transmit/receive operation.
- Programmable baud rates independent of bus clock frequency, with support for asynchronous operation during stop modes.
- Highly flexible 6-bit timers with support for various internal or external triggers, reset, enable, and disable conditions.
- Programmable logic mode for integrating external digital logic functions on-chip or combining pin/shifter/timer functions to generate complex outputs.
- Programmable state machine for offloading basic system control functions from CPU with support for up to 8 states, 8 outputs, and 3 selectable inputs per state.

Figure 1 gives a high-level overview of the configuration of FlexIO timers and shifters.

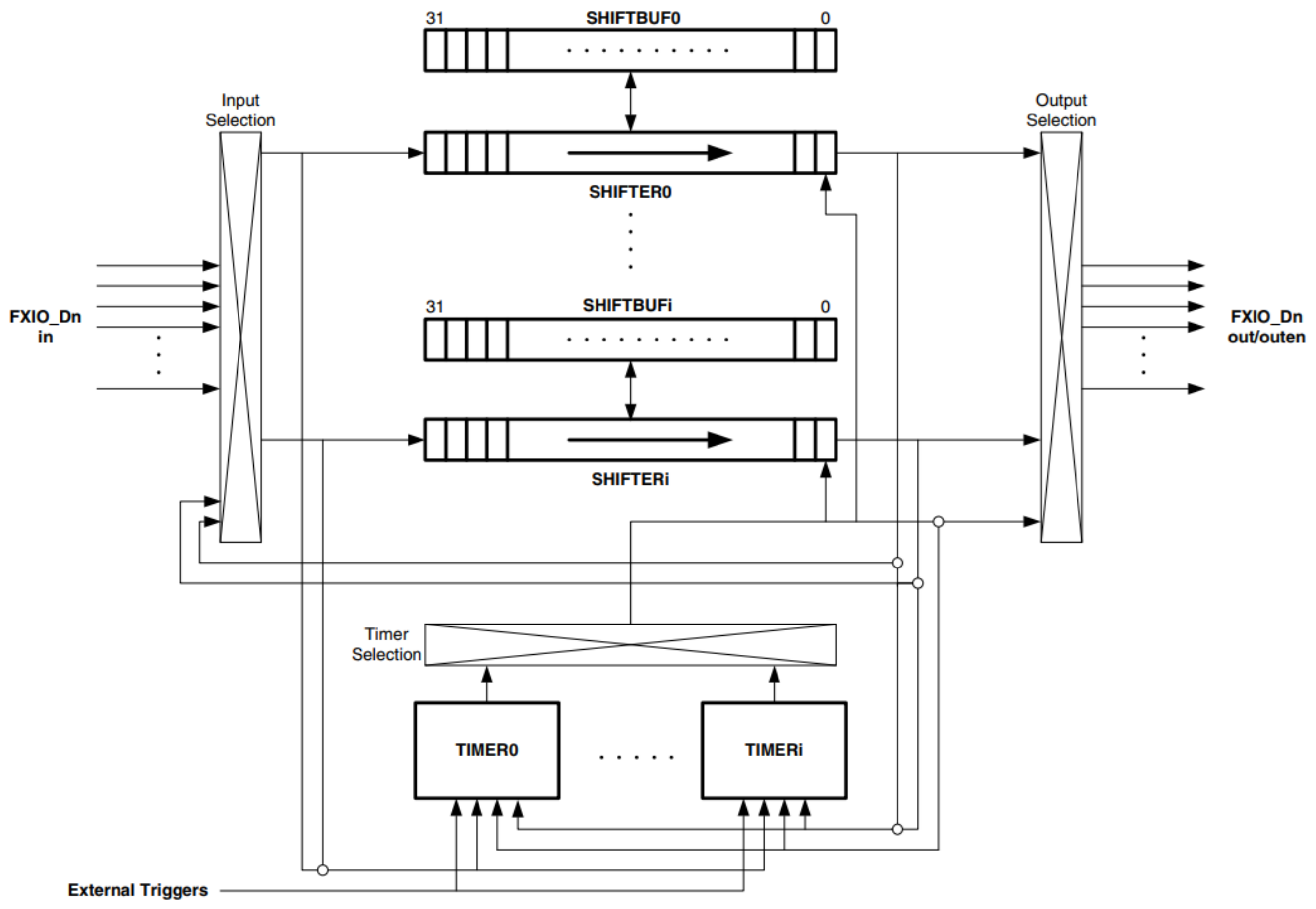


Figure 1. FlexIO Block Diagram

From the FLEXIO_PARAM register, users can read the amount of these resources like shifter, timer, pin, and trigger. For instance, there are eight shifters, eight timers, 32 pins, and two external triggers in I.MXRT1010 (In this device, limited by the number of pins, FlexIO only has 27 pins).

3 Emulating SPI

This chapter mainly introduces how to emulate SPI by using FlexIO module. It describes the configuration of SPI master and slave mode in detail.

3.1 SPI master configuration

To emulate SPI master, following resources are needed:

- Two Timers – one for SPI_CS output generation and the other for the load/store/shift control of the two shifters and SPI_SCK generation.
- Two Shifters – one for data transmitter and the other for receiver.
- Four Pins – separately connect to the two Timers and two Shifters used as SPI_CS, SPI_SCK, SPI_MOSI and SPI_MISO.

Figure 2shows the FlexIO SPI master configuration diagram.

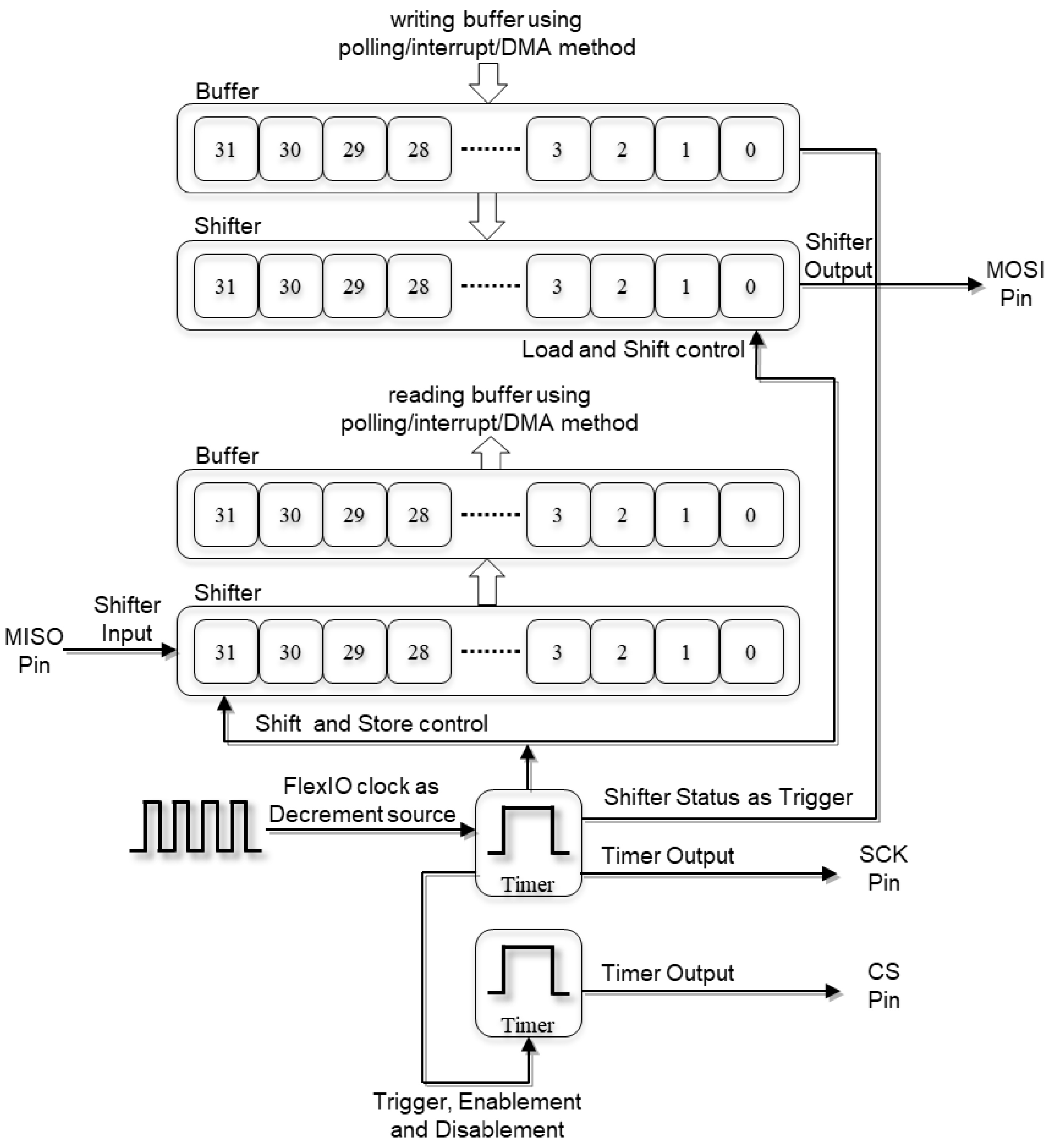


Figure 2. FlexIO SPI master block diagram

In this application note, Timer 0 generates SPI_SCK signal and outputs to FlexIO_D26, Timer 1 generates SPI_CS signal and outputs to FlexIO_D0, Shifter 0 connects to FlexIO_D21 to transmit the data on each rising edge of SPI_SCK, Shifter 1 connects to FlexIO_D22 to receive the data on each falling edge of SPI_SCK. Table 1 shows the configuration for Timer 0.

Table 1. Configurations for Timer 0

Items	Configurations

Table continues on the next page...

Table 1. Configurations for Timer 0 (continued)

Trigger Select	Shifter 0 status flag
Trigger Polarity	active low
Trigger Source	internal trigger
Pin Config	output
Pin Select	FlexIO_D26
Pin Polarity	active high
Timer mode	dual 8-bit counters baud mode
Timer Output	Timer output is logic zero when enabled and timer reset does not affect it
Timer Decrement	decrement counter on FlexIO clock, shift clock equals Timer output
Timer Reset	Timer never resets
Timer Disable	Timer disabled on Timer compare
Timer Enable	Timer enabled on Trigger high
Timer Stop Bit	enabled on timer disable
Timer Start Bit	enable
Timer Compare	((bitCountPerChar[1] * 2 - 1) << 8) | (baudrate_divider[2] / 2 - 1))

[1] bitCountPerChar is the number of bits of each data.

[2] baudrate_divider is calculated from FlexIO clock divided by SPI baud rate.

In 8-bit Baud Counter Mode, the 16-bit counter is divided into two 8-bit counters. The lower 8 bits are used to configure the baud rate of the shift clock, and the upper 8 bits are used to configure the number of shift clock edges in the transfer. When the lower 8 bits decrease to zero, the timer output toggles, and the lower 8 bits reload from the compare register. The upper 8 bits decrease when the lower 8 bits equals zero.

Table 2 shows the configuration for Timer 1.

Table 2. Configurations for Timer 1

Items	Configurations
Trigger Select	trigger from Timer 0
Trigger Polarity	active high
Trigger Source	internal trigger
Pin Config	output
Pin Select	FlexIO_D0

Table continues on the next page...

Table 2. Configurations for Timer 1 (continued)

Pin Polarity	active low
Timer mode	single 16-bit counter mode
Timer Output	Timer output is logic one when enabled and timer reset does not affect it.
Timer Decrement	decrement counter on FlexIO clock, shift clock equals Timer output
Timer Reset	Timer never resets
Timer Disable	Timer disabled on Timer 0 disable
Timer Enable	Timer enabled on Timer 0 enable
Timer Stop Bit	disable
Timer Start Bit	disable
Timer Compare	0xFFFF

Timer 1 enables when the timer 0 enables. The compare register is configured to the 16-bit counter and set to 0xFFFF. With this value, the timer never compares and is always active when the timer is enabled.

Table 3 shows the configuration for Shifter 0.

Table 3. Configurations for Shifter 0

Items	Configurations
Timer Select	Timer 0
Timer Polarity	shift on negedge of shift clock
Pin Config	Shifter pin output
Pin Select	FlexIO_D21
Pin Polarity	active high
Shifter Mode	transmit mode
Input Source	input from pin
Shifter Stop Bit	disable
Shifter Start Bit	disable, transmitter loads data on enable

Table 4 shows the configuration for Shifter 1.

Table 4. Configurations for Shifter 1

Items	Configurations

Table continues on the next page...

Table 4. Configurations for Shifter 1 (continued)

Timer Select	Timer 0
Timer Polarity	shift on posedge of shift clock
Pin Config	output disable
Pin Select	FlexIO_D22
Pin Polarity	active high
Shifter Mode	receive mode
Input Source	input from pin
Shifter Stop Bit	disable
Shifter Start Bit	disable, transmitter loads data on enable

3.2 SPI slave configuration

To emulate SPI slave, following resources are needed:

- One Timer – for the load/store/shift control of the two shifters.
- Two Shifters – one for data transmitter and the other for receiver.
- Four Pins – used as SPI_CS, SPI_SCK, SPI_MOSI and SPI_MISO.

Figure 3 shows the FlexIO SPI slave configuration diagram.

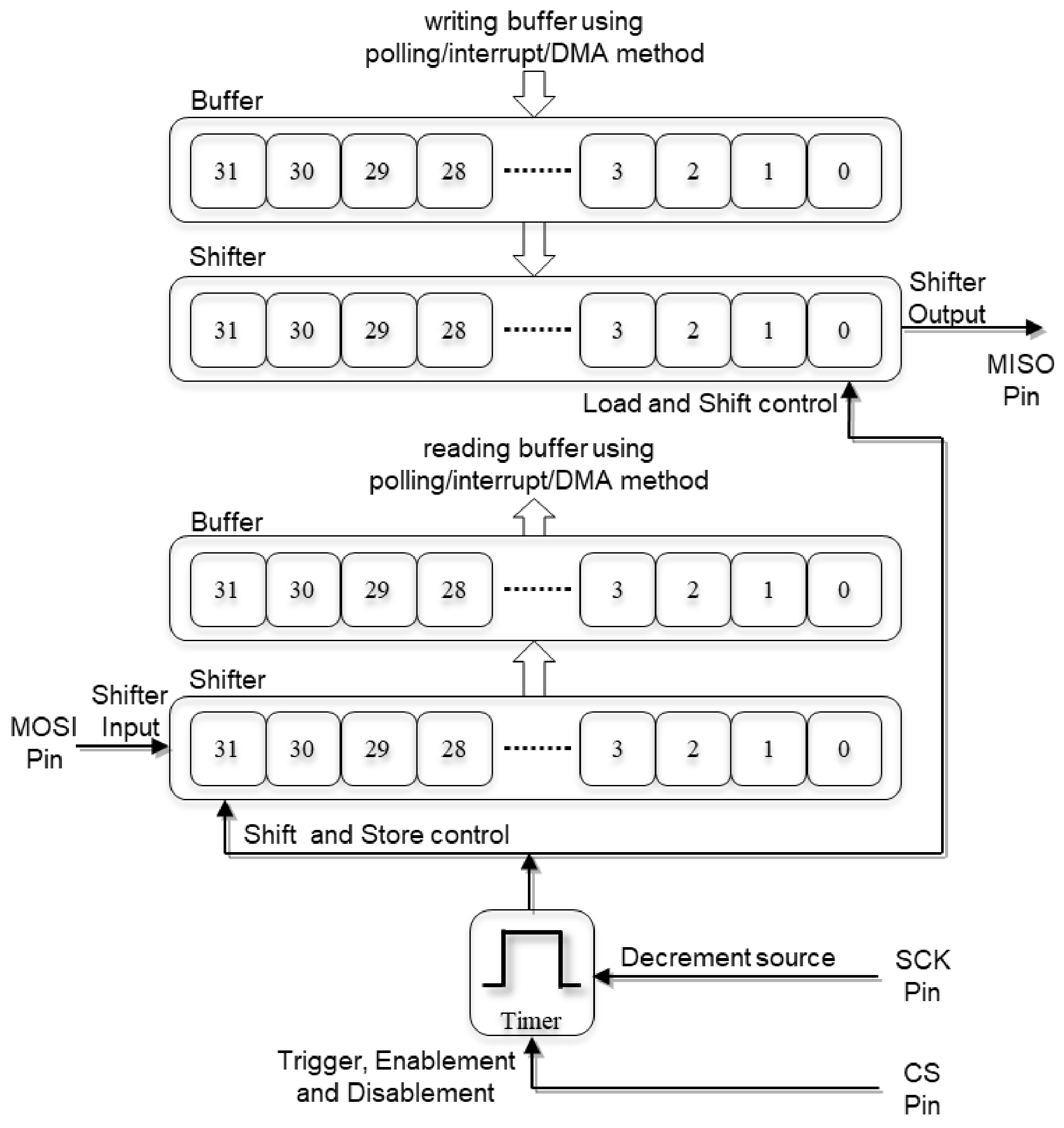


Figure 3. FlexIO SPI slave block diagram

In slave mode, SPI slave uses Timer 0 to acquire SPI_SCK signal on FlexIO_D26 pin from master to load/store/shift control of the two shifters. The SPI_SCK and SPI_CS signals are configured as inputs and SPI bus master drives them. The transmit data is transferred at every SPI_SCK clock edge of each frame to the shift register when the SPI_CS signal is asserted. As a result, select pin FlexIO_D0 of SPI_CS as the trigger input to Timer 0. Shifter 0 is used as SPI slave transmitter on pin FlexIO_D21, Shifter 1 is used as SPI slave receiver on pin FlexIO_D22.

Table 5 shows the configuration for Timer 0.

Table 5. Configurations for Timer 0

Items	Configurations
Trigger Select	trigger from FlexIO_D0 input
Trigger Polarity	active low
Trigger Source	internal trigger
Pin Config	output disable
Pin Select	FlexIO_D26
Pin Polarity	active high
Timer mode	single 16-bit counter mode
Timer Output	Timer output is logic zero when enabled and timer reset does not affect it
Timer Decrement	decrement counter on pin input, shift clock equals pin input
Timer Reset	Timer never resets
Timer Disable	Timer disabled on Timer compare
Timer Enable	Timer enabled on trigger rising edge
Timer Stop Bit	disable
Timer Start Bit	disable
Timer Compare	((bitCountPerChar * 2 - 1)

Table 6 shows the configuration for Shifter 0.

Table 6. Configurations for Shifter 0

Items	Configurations
Timer Select	Timer 0
Timer Polarity	shift on negedge of shift clock
Pin Config	Shifter pin output
Pin Select	FlexIO_D21
Pin Polarity	active high
Shifter Mode	transmit mode
Input Source	input from pin

Table continues on the next page...

Table 6. Configurations for Shifter 0 (continued)

Shifter Stop Bit	disable
Shifter Start Bit	disable, transmitter loads data on enable

Table 7 shows the configuration for Shifter 1.

Table 7. Configurations for Shifter 1

Items	Configurations
Timer Select	Timer 0
Timer Polarity	shift on posedge of shift clock
Pin Config	output disable
Pin Select	FlexIO_D22
Pin Polarity	active high
Shifter Mode	receive mode
Input Source	input from pin
Shifter Stop Bit	disable
Shifter Start Bit	disable, transmitter loads data on enable

4 Run the example

4.1 Development Platform

This document describes the example of application based on the I.MXRT1010-EVK board as shown in Figure 4. Users can also easily enable this application on other I.MXRT series EVK board.

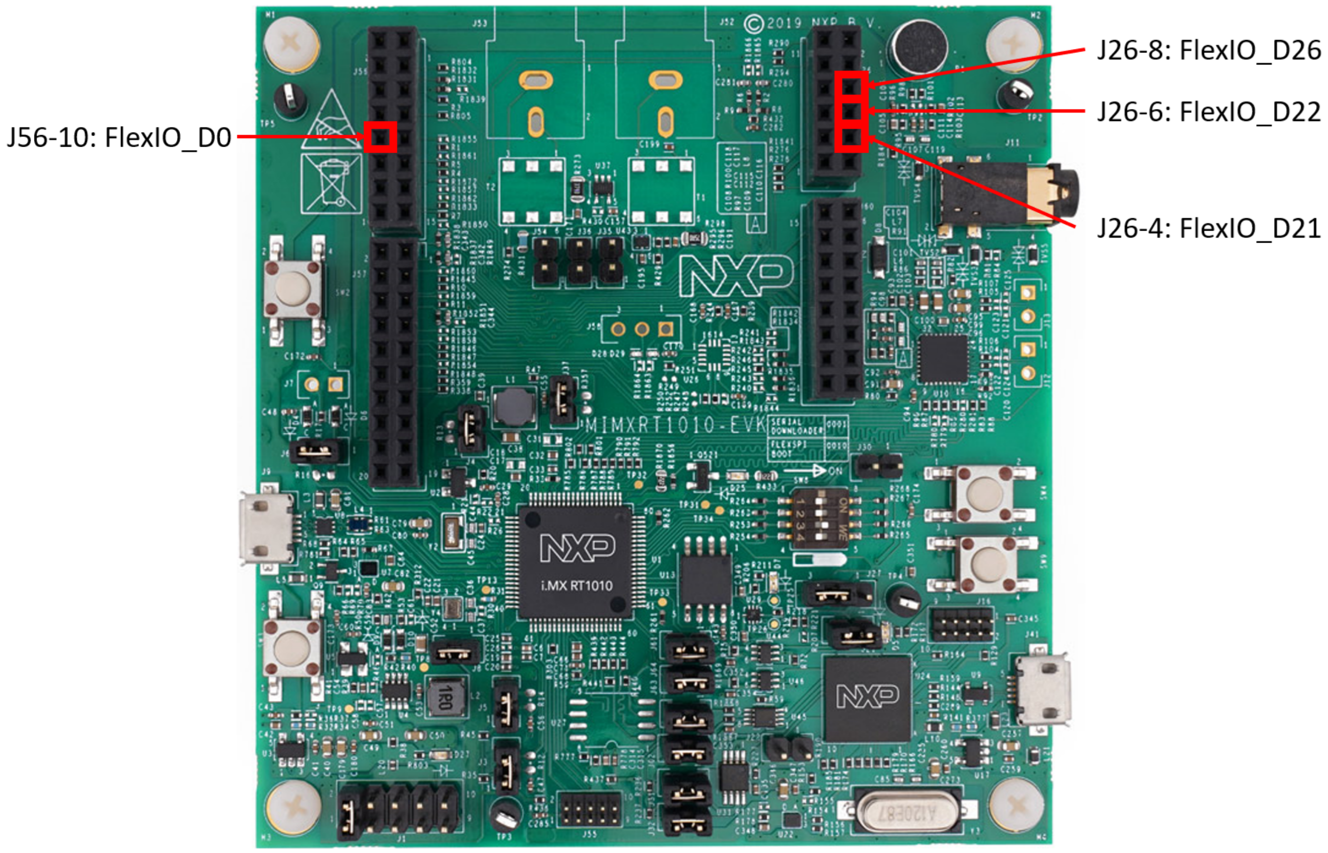
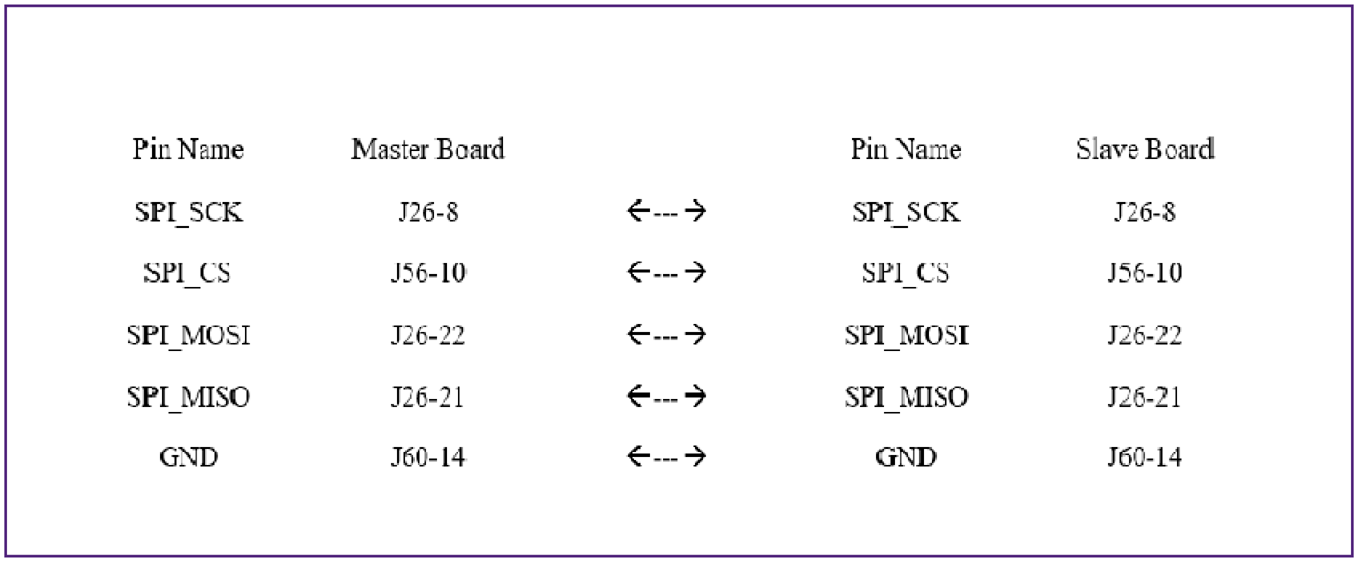


Figure 4. I.MXRT1010-EVK Board

In this application, FlexIO uses FlexIO_D21 pin as data output pin, FlexIO_D22 pin as the data input pin, FlexIO_D26 pin as the SPI_SCK, and FlexIO_D0 pin as SPI_CS.

This application uses two boards, one board as master and one board as slave. The connection between SPI master board and SPI slave board is as follows:



Pin Name	Master Board		Pin Name	Slave Board
SPI_SCK	J26-8	←---→	SPI_SCK	J26-8
SPI_CS	J56-10	←---→	SPI_CS	J56-10
SPI_MOSI	J26-22	←---→	SPI_MOSI	J26-22
SPI_MISO	J26-21	←---→	SPI_MISO	J26-21
GND	J60-14	←---→	GND	J60-14

4.2 Run the demo

Users can download the software in nxp.com. Find the two IAR projects *flexio_spi_master* and *flexio_spi_slave*. Users can separately download these two codes to two boards, connect two boards as above and run the two demos. Figure 5 shows the waveform captured by an oscilloscope.

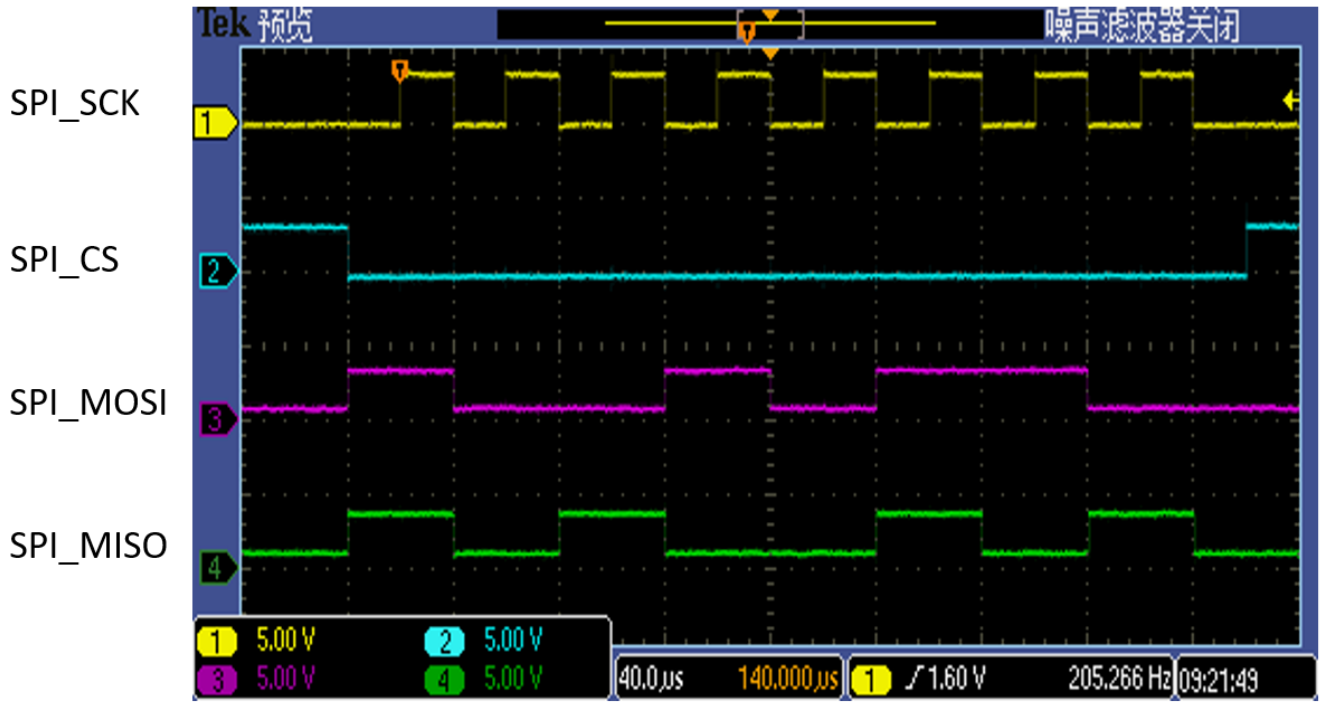
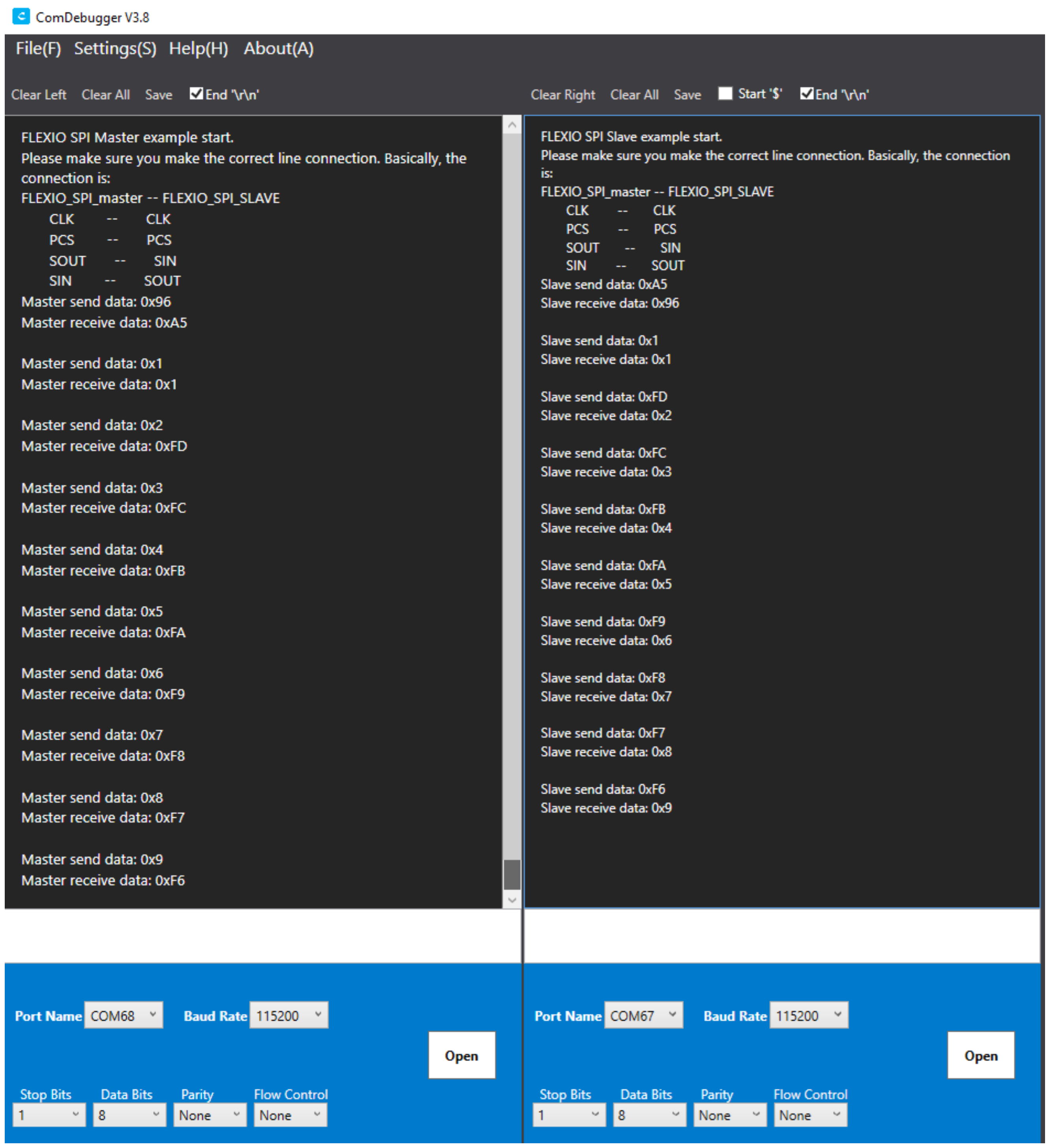


Figure 5. Waveform of the SPI signal

The SPI master board sends 0x96 to the SPI slave board and receives 0xa5 at the same time. Figure 6shows the communication data between SPI master and slave.


ComDebugger V3.8
File(F) Settings(S) Help(H) About(A)
Clear Left Clear All Save ✓End '\r\n'
Clear Right Clear All Save Start '$' ✓End '\r\n'
Open
Open

```
FLEXIO SPI Master example start.
Please make sure you make the correct line connection. Basically, the
connection is:
FLEXIO_SPI_master -- FLEXIO_SPI_SLAVE
      CLK      --      CLK
      PCS      --      PCS
      SOUT     --      SIN
      SIN      --      SOUT
Master send data: 0x96
Master receive data: 0xA5

Master send data: 0x1
Master receive data: 0x1

Master send data: 0x2
Master receive data: 0xFD

Master send data: 0x3
Master receive data: 0xFC

Master send data: 0x4
Master receive data: 0xFB

Master send data: 0x5
Master receive data: 0xFA

Master send data: 0x6
Master receive data: 0xF9

Master send data: 0x7
Master receive data: 0xF8

Master send data: 0x8
Master receive data: 0xF7

Master send data: 0x9
Master receive data: 0xF6
```

```
FLEXIO SPI Slave example start.
Please make sure you make the correct line connection. Basically, the connection
is:
FLEXIO_SPI_master -- FLEXIO_SPI_SLAVE
      CLK      --      CLK
      PCS      --      PCS
      SOUT     --      SIN
      SIN      --      SOUT
Slave send data: 0xA5
Slave receive data: 0x96

Slave send data: 0x1
Slave receive data: 0x1

Slave send data: 0xFD
Slave receive data: 0x2

Slave send data: 0xFC
Slave receive data: 0x3

Slave send data: 0xFB
Slave receive data: 0x4

Slave send data: 0xFA
Slave receive data: 0x5

Slave send data: 0xF9
Slave receive data: 0x6

Slave send data: 0xF8
Slave receive data: 0x7

Slave send data: 0xF7
Slave receive data: 0x8

Slave send data: 0xF6
Slave receive data: 0x9
```

Port Name	Baud Rate	Stop Bits	Data Bits	Parity	Flow Control
COM68	115200	1	8	None	None
COM67	115200	1	8	None	None

Figure 6. Transceiver data printed by debug console

5 References

1. I.MX RT1010 Processor Reference Manual (document I.MXRT1010RM)
2. MCUXpresso SDK: Software Development Kit for NXP MCUs

https://mcuxpresso.nxp.com/en/welcome

6 Revision history

Table 8. Revision history

Revision number	Date	Substantive changes
0	01/2020	Initial release

Date of release: 13 March 2020

Document identifier: AN12780