ASCLIN_SPI_Master_1

SPI master communication via ASCLIN module

AURIX™ TC2xx Microcontroller Training
V1.0.0

Scope of work

An ASCLIN module configured as SPI master sends a two bytes message.

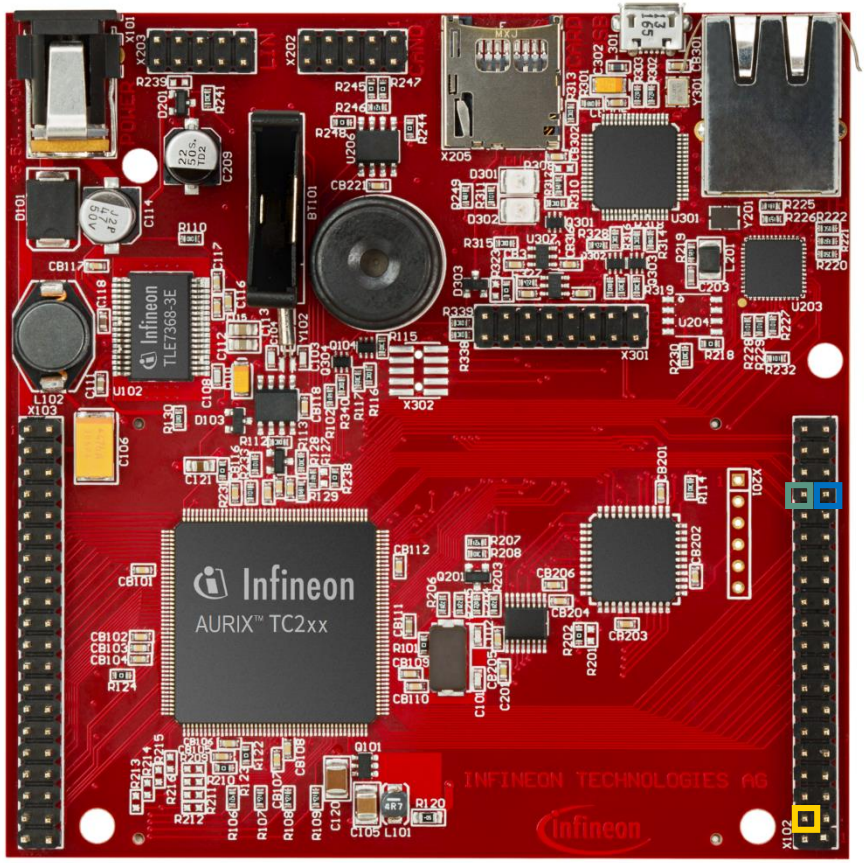
The two bytes message is sent through MTSR (MOSI) port pin P15.4 in loopback mode. This signal can be visualized on the oscilloscope screen.

Introduction

- The Asynchronous/Synchronous Interface (ASCLIN) module provides synchronous serial communication like SPI with external devices, using data-in and data-out signals only.

- The ASCLIN module in SPI configuration can support master mode only with four-wire or three-wire (without slave select output signal) and up to 16-bit data width.

Hardware setup



This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.
The port pin P15.4 (SPI-MTSR) should be connected to the port pin P15.5 (SPI-MRST) in order to form an internal loopback.
Those pins can also be connected to an oscilloscope probe for observing the SPI signal.

		X102			
	P14.5	40	39	P14.4	
	P20.10	38	37	P20.9	
	P15.7	36	35	P15.6	
MRST	P15.5	34	33	P15.4	MTSR
	P15.3	32	31	P15.2	
	P22.3	30	29	P22.2	
	P22.1	28	27	P22.0	
	P33.11	26	25	P23.4	
	P23.3	24	23	P23.2	
	P23.1	22	21	P23.0	
	P33.6	20	19	P33.8	
	P33.12	18	17	P33.1	
	P33.2	16	15	P33.3	
	P33.4	14	13	P33.5	
	AN0	12	11	AN8	
	AN2	10	9	AN3	
	AN32	8	7	AN33	
	AN20	6	5	AN21	
Ground	GND	4	3	GND	
	V_UC(+5V)	2	1	VCC_IN	

Implementation

Configuration of the ASCLIN module:

Configuration of the ASCLIN module for SPI communication is done in the setup phase by initializing an instance of the ***IfxAsclin_Spi_Config*** structure with the following parameters:

- ***baudrate*** – structure to set the actual communication speed in bit/s
- ***interrupt*** – structure to set:
  - transmit and receive interrupt priorities (***txPriority***, ***rxPriority***)
  - ***typeOfService*** – defines which service provider is responsible for handling the interrupt, which can be any of the available CPUs, or the DMA
- ***pins*** – structure to set which GPIO port pins are used for the communication
- ***rxBuffer***, ***rxBufferSize***, ***txBuffer***, ***txBufferSize*** – to configure the buffers that will hold the incoming/outgoing data

The function ***IfxAsclin_Spi_initModuleConfig()*** fills the configuration structure with default values and ***IfxAsclin_Spi_initModule()*** initializes the module with the user configuration.

All the above functions can be found in the iLLD header ***IfxAsclin_Spi.h***.

Implementation

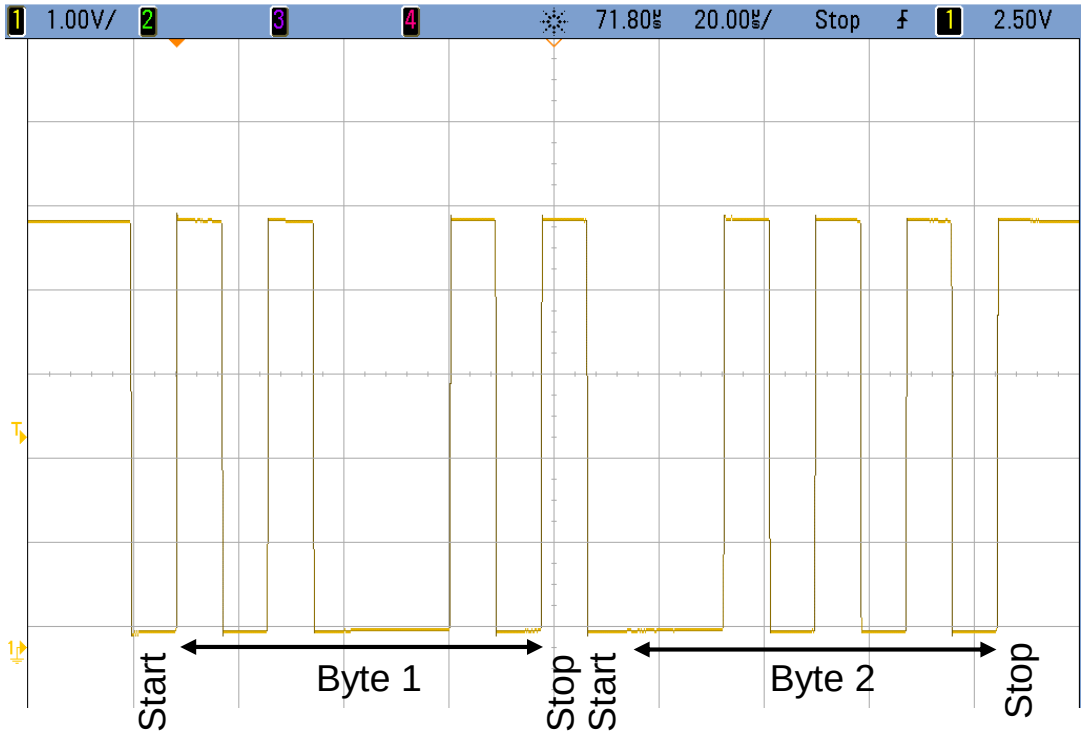
The SPI message exchange function:

- The data-out (MTSR/MOSI) is connected via internal loopback to the data-in (MRST/MISO).
- The two bytes message is sent via the function ***exchange_ASCLIN_SPI_message()*** which is called once after initialization of the ASCLIN module.
- The two bytes message is sent from the ***g_spiTxBuffer*** to the ***g_spiRxBuffer*** using the function ***IfxAsclin_Spi_exchange()*** from the ***IfxAsclin_Spi.h*** header file.

Run and Test

After code compilation and flashing the device, perform the following steps:

- Connect the oscilloscope probe to the MTSR pin (P15.4)
- Reset and run the program by pressing the PORST push button
- Check the oscilloscope for the SPI signal:



Run and Test

An additional test without using an oscilloscope can be performed with the debugger.

- Before transmission, the buffer ***g_spiTxBuffer*** is filled with a two bytes message and the buffer ***g_spiRxBuffer*** is empty.

- After transmission, both buffers should hold the same message:
  - By using the debugger, you can watch the content of both buffers before and after transmission by setting a breakpoint to ***exchange_ASCLIN_SPI_message()***.
  - When reaching this breakpoint, check the content of both buffers (it should be different).
  - After stepping over this function, the content of the buffers must be equal.

Note: The code should run for a few seconds in order to grant enough time for the transmission to be done.

References

- AURIX™ Development Studio is available online:
- https://www.infineon.com/aurixdevelopmentstudio
- Use the *„Import...“* function to get access to more code examples.

- More code examples can be found on the GIT repository:
- https://github.com/Infineon/AURIX_code_examples

- For additional trainings, visit our webpage:
- https://www.infineon.com/aurix-expert-training

- For questions and support, use the AURIX™ Forum:
- https://www.infineonforums.com/forums/13-Aurix-Forum

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-10
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
ASCLIN_SPI_Master_1

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.