



SPC572L64 DSPI example

Quick Guide

Table of Contents

- 1. Terms & Abbreviations 1**
- 2. Introduction 2**
 - 2.1. Prerequisites 2
 - 2.2. HW resources 2
- 3. Example overview 3**
 - 3.1. System Description 3
 - 3.2. Application Flow 3
 - 3.3. Interrupt Handling 4
 - 3.4. DSPI module configuration 5
- 4. Example Import & Build 7**
- Appendix A: Document References 8**
- Appendix B: Release Notes 9**

1. Terms & Abbreviations

DSPI	Deserial Serial Peripheral Interface.
PIT	Periodic Interrupt Timer
IOP	I/O processor
BSP	Board Startup Package
GPIO	General Purpose Input Output
crt0	'C' run-time environment initialization code
HAL	Hardware Abstraction Layer
EVB	Evaluation Board
NC	Not Configured
uC	Microcontroller
ISR	Interrupt Service Routine

2. Introduction

This example is a small functional project designed to simplify an evaluation phase of the certain peripheral on the microcontroller. It is built on the c-startup example providing necessary low-level functions like a startup code, a minimalistic hardware abstraction or predefined memory partitioning. On top of this low-level implementation, it provides the peripheral example code running on single core.

2.1. Prerequisites

- SPC572L64 device
- An evaluation board (SPC57xxMB + SPC572L64)
- HighTec Development Suite, version: 4.9.3.0

2.2. HW resources

Hardware resources used by this example:

HW unit	channel / output pin	function
DSPI_0 SCK	PD[4]	Core [0] DSPI 0 Serial Clock
DSPI_0 SIN	PD[5]	Core [0] DSPI 0 Serial Data Input
DSPI_0 SOUT	PE[9]	Core [0] DSPI 0 Serial Data Output
DSPI_0 CS[0]	PE[5]	Core [0] DSPI 0 Chip Select 0
default resources from the c-startup example		
GPIO	PA [0]	Core [0] LED control
PIT 0	Channel [0]	Core [0] periodic interrupt

Tab. 1. HW resources

NOTE

For correct behavior, the example expects interconnection of pin PE[9] and PD[5] on the SPC57xxMB evaluation board.

3.3. Interrupt Handling

The DSPI initialization function assigns the DSPI_0 Rx FIFO drain request flag interrupt vector and sets the interrupt vector priority.

The example uses the SW interrupt mode.

3.4. DSPI module configuration

The DSPI configuration sets these parameters:

- DSPI in master mode
- TX/RX FIFO not used
- DSPI use SPI protocol
- Continuous clock mode disabled
- $f_{\text{DSPI}} = 80\text{MHz}$

Parameter	Value
Baud rate	1MHz
Frame size	16-bit
Clock polarity	The inactive state value of SCK is low.
Chip select	PCS0
Chipset active state	The inactive state of PCS0 is high.
Data order	Data is transferred MSB first.

Tab. 2. Transfer attributes

for more details see `dspi_Init` function snippet below.

Code 2. `/app/dspi.c`

```

1 void dspi_Init(void)
2 {
3     /* Configure DSPI0 */
4     DSPI_0.MCR.B.MDIS = 0x0;           /* Enable DSPI clocks.*/
5     DSPI_0.MCR.B.MSTR = 0x1;           /* Set DSPI0 as a master */
6     DSPI_0.MCR.B.CONT_SCKE = 0x0;      /* Disable continuous clock */
7     DSPI_0.MCR.B.PCSIS = 0x1;          /* Inactive state of CS = 1 */
8     DSPI_0.MCR.B.DCONF = 0x0;         /* Set SPI mode*/
9     DSPI_0.MCR.B.DIS_TXF = 0x1;        /* Disable TX fifo */
10    DSPI_0.MCR.B.DIS_RXF = 0x1;        /* Disable RX fifo */
11    DSPI_0.CTAR[0].B.FMSZ = DSPI_CTAR_FMSZ; /* Frame size */
12    DSPI_0.CTAR[0].B.PBR = DSPI_CTAR_PBR; /* Baud Rate Prescaler */
13    DSPI_0.CTAR[0].B.BR = DSPI_CTAR_BR; /* Baud rate scaler */
14    DSPI_0.CTAR[0].B.DBR = DSPI_CTAR_DBR; /* Double Baud Rate */
15    DSPI_0.CTAR[0].B.PCSSCK = 0x3;      /* PSC to SCK prescaler set to 7 */
16    DSPI_0.CTAR[0].B.DT = 0x8;         /* Delay after transfer */
17
18    DSPI_0.RSER.B.RFDF_RE = 1;          /* Enable interrupt reqs */
19
20    DSPI_0.SR.B.RFDF = 0x1;             /* Clear flag */
21    DSPI_0.MCR.B.HALT = 0x0;           /* Start transmit/receive */
22    ...

```

23 }

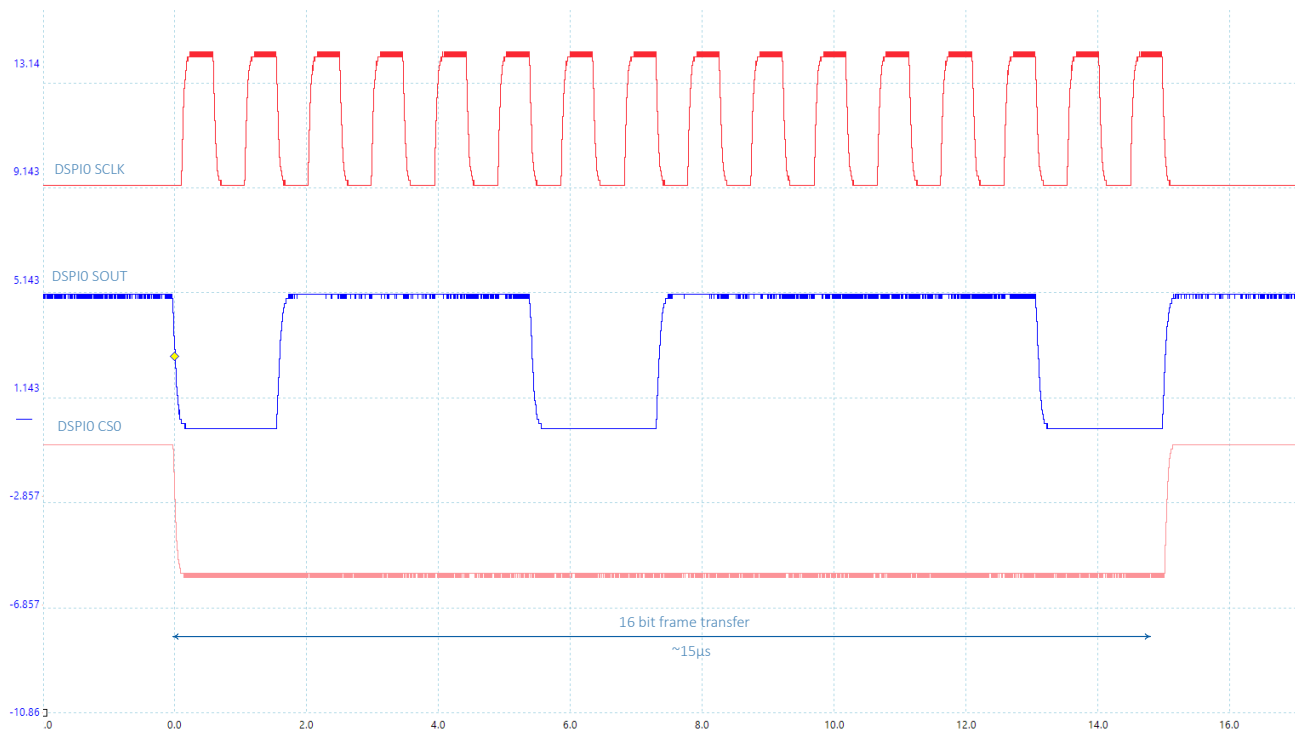


Fig. 3. DSPI transfer timing diagram

4. Example Import & Build

Follow these steps to import an example project to the HighTec IDE environment:

1. From menu **File** → **Import** → **General** choose an option Existing Projects into Workspace
2. Browse for your project location
3. Select project
4. Leave the copy to the workspace option empty
5. Click Finish

Activate the project from menu **Project** → **Set Active Project**.

Build the project from the menu **Project** → **Build Project**.

The output binary file is located under the `_irom_build` folder.

Appendix A: Document References

- [1] "SPC5x c-startup - 'C' run-time initialization", HighTec EDV Systeme GmbH, 2018
- [2] "SPC5x c-startup - Linker file template", HighTec EDV Systeme GmbH, 2018
- [3] "SPC5x c-startup - Hardware Abstraction Layer", HighTec EDV Systeme GmbH, 2018

Appendix B: Release Notes

Version	Date	Changes to the previous version
1.0	March 2018	Example first version Example BSP based on SPC572L64 c-startup BSP version V1.2



HighTec EDV-Systeme GmbH
Feldmannstrasse 98, D-66119 Saarbrücken
info@hightec-rt.com
+49-681-92613-16
www.hightec-rt.com