RTL9310

SPI

Application Note

Rev. 1.0
27 Apr 2018

Realtek Semiconductor Corp.
No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan
Tel.: +886-3-578-0211 Fax: +886-3-577-6047
www.realtek.com

Realtek confidential files
The document authorized to
SuperHome Telecommunication Technologies
2018-09-20 11:06:04

COPYRIGHT
©2018 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

TRADEMARKS
Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

DISCLAIMER
Realtek provides this document "as is", without warranty of any kind, neither expressed nor implied, including, but not limited to, the particular purpose. Realtek may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

USING THIS DOCUMENT
This document is intended for use by the system engineer when integrating with Realtek switch products. Though every effort has been made to assure that this document is current and accurate, more information may have become available subsequent to the production of this guide. In that event, please contact your Realtek representative for additional information that may help in the development process.

Revision	Release Date	Summary
1.0	2018/4/27	Initial

Realtek confidential files
The document authorized to
SuperHome Telecommunication Technologies
2018-09-20 11:06:04

Contents

Realtek confidential files
The document authorized to
SuperHome Telecommunication Technologies
2018-09-20 11:06:04

Table List

Realtek confidential files
The document authorized to
PowerHome Telecommunication Technologies
2018-09-20 11:06:04

1 Overview

RTL9310 supports 2 set master SPI interfaces, which share the SCK pin, MOSI pin and MISO pin, but have different CS# pin. RTL9310 master SPI interface will be used to connect with slave SPI devices, for example, the PSE IC.

2 SPI Master

GPIO[12:8] can be reused as master SPI interface, GPIO8 for SCK, GPIO9 for MISO, GPIO10 for MOSI, GPIO11 for CS#0, and GPIO12 for CS#1.

SPIO Clock Rate Select

SPI operating clock rate selection is determined by *SPI_CLK_SEL*. Please refer to Table 2-3 錯誤! 找不到參照來源。 for details.

Setup and Hold Time Control

The setup time and hold time of the CS# active can be adjusted by *SPI_TSLCH* and *SPI_TCHSH*. Please refer to Table 2-3 for details.

CPOL and CPHA Select

The CPOL can be chosen by *SPI_CPOL*, and the CPHA can be chosen by *SPI_CPHA*. Please refer to Table 2-3 for details.

MOSI/SCK/MISO Delay

The delay of MOSI, SCK and MISO is used to adjust the timing, defined by *SPI_TX_DLY*, *SPI_CLK_DLY* and *SPI_RX_DLY* respectively. Please refer to Table 2-4 for details.

On the application, RTL9310 master SPI could access all kinds of slave SPI devices. The timing between the SPI master and the SPI slave must match with each other. The following is an application case.

Application case: RTL9310 accesses POE device POE-A.

Default register configuration:

SPI_CLK_SEL = 0x0; (SPI Clock is 250MHz/4=62.5MHz)

SPI_CPHA = 0x0; (The data is sampled on the leading first clock edge)

SPI_CPOL = 0x0; (SCLK low while idle)

SPI_TSLCH = 0x4; (SPI CS# active setup time 16ns*5)

Realtek confidential files
The document authorized to
PowerHome Telecommunication Technologies
2018-09-20 11:06:04

SPI_TCHSH = 0x4; (SPI CS# active hold time 16ns*5)

SPI_RX_DLY = 0x0; (SPI RX data no delay)

SPI_CLK_DLY = 0x0; (SPI Clock data no delay)

SPI_TX_DLY = 0x0; (SPI TX data no delay)

Table 2-1 POE-A SPI write format

Byte 1					**Byte 2**	**Byte 3**	**Byte 4**
0	0	0	IC_ADDR[3:0]	Write command=1	Reg. ADDR[7:0]	Data Byte 0	Data Byte 1

Register configuration：

GPIO_SPI_SEL = 1; (Select GPIO works as SPI interface)
GPIO12_CSB1_SEL = 1; (Select GPIO12 works as GPIO or CS#1)

CSB_OUT_SEL = 1; (Select CS#1 output)

SPI_CMD = Byte1;

SPI_CMD_TYPE = 0x1; (Select write command)

ADDR_WIDTH = 0x0; (External SPI Slave device memory address width 1Byte)

DATA_WIDTH = 0x1; (SPI Command read/write data width 2Byte)

SPI_ADDR = Byte2;

SPI_DATA = Byte3+Byte4;

SPI_TRIG = 0x1; (Trigger SPI Master command)
Wait for *SPI_TRIG* = 0, then write success.

Table 2-2 POE-A SPI read format

Byte 1					Byte 2	Byte 3	+	Depend on value of Byte 3
0	0	0	IC_ADDR[3:0]	Read command=0	Reg. ADDR[7:0]	Read Word Number		Slave output data0,1,2…

Register configuration:

GPIO_SPI_SEL = 1; (Select GPIO works as SPI interface)
GPIO12_CSB1_SEL = 1; (Select GPIO12 works as GPIO or CS#1)

CSB_OUT_SEL = 1; (Select CS#1 output)

SPI_CMD = Byte1;

SPI_CMD_TYPE = 0x0; (Select read command)

ADDR_WIDTH = 0x1; (External SPI Slave device memory address width 2Byte)

Realtek confidential files
The document authorized to
PowerHome Telecommunication Technologies
2018-09-20 11:06:04

DATA_WIDTH = (Byte3)* 4 - 1; (SPI Command read/write data width (Byte3)* 4 Byte)

SPI_ADDR = Byte2 + Byte3;

SPI_TRIG = 0x1; (Trigger SPI Master command)

Wait for *SPI_TRIG* = 0, then read success;

Read *SPI_DATA* according to *DATA_WIDTH*;

2.1 Register Setting

Table 2-3 SPI_CTRL0 Register

Bits	Field	Description	Type	Default
31:26	RESERVED			
25:17	DATA_WIDTH	SPI Command read/write data width = (Data_Width+1) byte. Maximum support 512 bytes.	RW	0x003
16:14	ADDR_WIDTH	External SPI Slave device memory address width. 0x0: 1-Byte 0x1: 2-Byte 0x2: 3-Byte 0x3: 4-Byte 0x4:No Address Byte Others reserved	RW	0x2
13:11	SPI_CLK_SEL	SPI operating clock rate selection. The value defines the divisor to generate SPI clock. SPI Clock = Interface clock(250MHz)/ (SPI_CLK_SEL) 0x0: DIV=4 0x1: DIV=6 0x2: DIV=8 0x3: DIV=16 0x4: DIV=125 0x5: DIV=160 0x6: DIV=200 0x7: DIV=250	RW	0x0
10	GPIO_SPI_SEL	Select GPIO8-10 works as GPIO or SCK/ MISO/ MOSI. 0b0: works as GPIO8-10; 0b1: GPIO8 works as SCK, GPIO9 works as MISO, GPIO10 works as MOSI.	RW	0x0
9	GPIO12_CSB1_SEL	Select GPIO12 works as GPIO or CS#1. 0b0: GPIO12; 0b1: CS#1	RW	0x0
8	GPIO11_CSB0_SEL	Select GPIO11 works as GPIO or CS#0. 0b0: GPIO11; 0b1: CS#0	RW	0x0

Realtek confidential files The document authorized to The Telecommunication Technologies 2018-09-20 11:06:04 PerHome

7	SPI_CPHA	Clock phase control bit; CPHA parameter is used to shift the sampling phase 0b0: the data are sampled on the leading (first) clock edge 0b1: the data are sampled on the trailing (second) clock edge	RW	0x0
6	SPI_CPOL	Clock phase control bit; SPI clock may be inverted or non-inverted transmitting 0b0: SCLK low while idle 0b1: SCLK high while idle	RW	0x0
5:3	SPI_TSLCH	SPI CS# active setup time adjustment(relative to SCK) Basic unit = 16ns(Interface clock(250MHz)/4) 0x0 means 1 unit, 0x1 means 2 units, etc.	RW	0x4
2:0	SPI_TCHSH	SPI CS# active hold time adjustment. Basic unit = 16ns(Interface clock(250MHz)/4) 0x0 means 1 unit, 0x1 means 2 units, etc.	RW	0x4

Table 2-4 SPI_CTRL1 Register

Bits	Field	Description	Type	Default
31:9	RESERVED			
8:6	SPI_RX_DLY	SPI RX data delay control Basic unit = 4ns (Interface clock 250MHz) 0x0 means no delay , 0x1 means 1 unit, 0x2 means 2 units, etc.	RW	0x0
5:3	SPI_CLK_DLY	SPI clock delay control Basic unit = 4ns (Interface clock 250MHz) 0x0 means no delay , 0x1 means 1 unit, 0x2 means 2 units, etc.	RW	0x0
2:0	SPI_TX_DLY	SPI TX data delay control Basic unit = 4ns (Interface clock 250MHz) 0x0 means no delay , 0x1 means 1 unit, 0x2 means 2 units, etc.	RW	0x0

Table 2-5 SPI_CTRL2 Register

Bits	Field	Description	Type	Default
31:11	RESERVED			
10	CSB_OUT_SEL	Select CS# output from which pin; 0b0: CS#0 0b1: CS#1 Note: before setting CSB_OUT_SEL, CPU should first set GPIO pin as CS# function; otherwise, CSB_OUT_SEL will not work as expected; if CS#[x] is not selected as CSB output, it should output a high to avoid any mistake.	RW	0x0

Realtek confidential files
The document authorized to
SuperHome Telecommunication Technologies
2018-09-20 11:06:04

9	SPI_CMD_TYPE	Select read command: After read command output, SPI master must store the data from SPI slave output. Select write command: SPI master just only output write command. 0b0: Read 0b1: Write	RW	0x0
8:1	SPI_CMD	8:1 SPI_CMD SPI command code of a command transaction. Ex: 'Read data' is 0x3, 'write data' is 0x2	RW	0x0
0	SPI_TRIG	Trigger SPI Master command. 0: idle; 1: trigger, self cleared to 0 after executed.	RWA C	0x0

Table 2-6 SPI_ADDR Register

Bits	Field	Description	Type	Default
31:0	SPI_ADDR	External SPI slave device memory address for access. Note: When "ADDR_WIDTH selected as 1-Byte mode, bit [7: 0] is valid, bit [31: 8] is invalid. When "ADDR_WIDTH selected as 2-Byte mode, bit [15: 0] is valid, bit [31: 16] is invalid. When "ADDR_WIDTH selected as 3-Byte mode, bit [23: 0] is valid, bit [31: 24] is invalid.	RW	0x0

Table 2-7 SPI_DATA Register

Bits	Field	Description	Type	Default
4095:0	SPI_DATA	For read operation, data is read from slave SPI device; For write operation, data will be write to slave SPI device. Note: only the bits assigned at Data_Width field are valid. For example, set Data_Width to 5, so only bit[47:0] are valid.	RW	0x0

2.2 SDK Setting

SDK3

SDK3 has already supported 4 SPI device drivers, dev1,dev2,dev3 and dev4. User can select which one of them to compile (only one).

The SDK menuconfig path to select is: SDK Configuration -> BSP Option -> SPI device.

Maybe the SPI device driver already implemented does not match user's requirement, user can add a new SPI_DEV_x driver code or modify the current one of the 4 device driver.

Each SPI device driver has an unique file (spi_devx.c) including its own initialization function: spi_devx_init() which is called in function: drv_spi_init.

If user wants to modify based on the current SPI device driver, suggest basing on the SPI4 device driver.

Realtek confidential files
The document authorized to
PowerHome Telecommunication Technologies
2018-09-20 11:06:04

rtk API

SPI write/read API will call specified SPI device write/read by CONFIG_SDK_SPI_DEV_x.

int32 drv_spi_write(uint32 unit, uint32 mAddrs, uint32 *pBuff)

int32 drv_spi_read(uint32 unit, uint32 mAddrs, uint32 *pBuff)

Realtek confidential files
The document authorized to
PowerHome Telecommunication Technologies
2018-09-20 11:06:04