WL865E4-P SPI Host Interface Application Note

80613NT11811A Rev. 1–2020-09-09

TELIT
TECHNICAL
DOCUMENTATION

[04.2016]

Mod. 0809 2016-08 Rev.7

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

NOTICE

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

USAGE AND DISCLOSURE RESTRICTIONS

I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

V. Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY “OTHER CODE”), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

APPLICABILITY TABLE

PRODUCTS

- WL865E4-P

CONTENTS

FIGURE LIST

TABLE LIST

1. INTRODUCTION

1.1. Scope

This document provides an overview of the SPI communication interfaces available in the WL865 module.

1.2. Audience

This document is intended for Telit customers, who are integrators, about to implement their applications using our WL865E4-P modules.

1.3. Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com

Alternatively, use:

https://www.telit.com/contact-us/

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Conventions

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

2. SPI INTERFACE

2.1. Overview

The WL865E4-P operates as a SPI slave device through a 5-pin interface. The SPI slave uses the standard 4-wire Motorola-SPI protocol (see Figure 2-1). The additional pin, SDIO_D1 is an interrupt signal (level triggered) from the WL865E4-P slave device to host, indicating a pending interrupt. Upon receiving interrupt, host must check slave by reading slave registers; if slave is not ready, host must wait for any write operation. But read operation can continue and its independent of write.

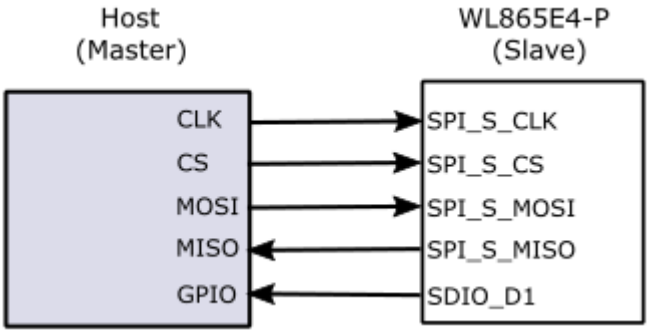

Figure 2-1 SPI Interface

It is important for the SPI host to choose the SPI clock polarity (CPOL) and phase (CPHA) format that matches the SPI slave. WL865E4-P slave SPI peripheral by default operates in SPI mode 3 (Inactive state of serial clock is high, samples the incoming data (MOSI) on the second edge of the serial clock (clock phase 180), after the falling edge of CS signal). See Figure 2-2.

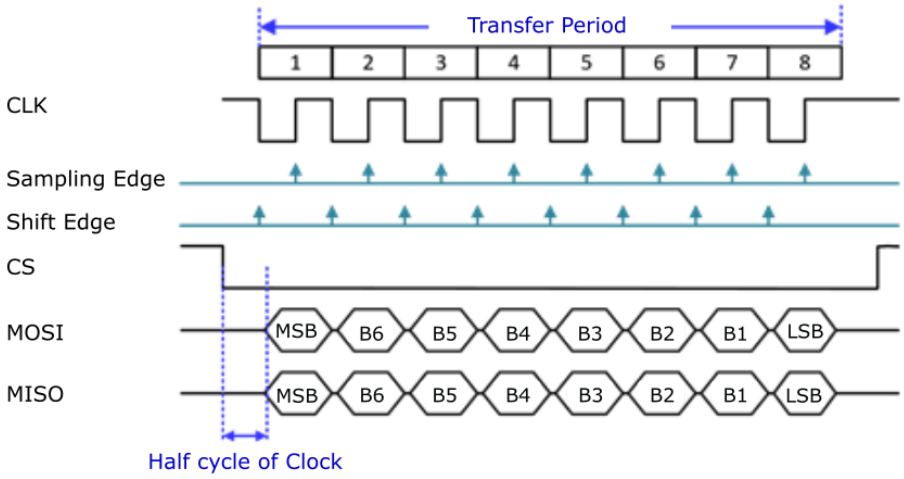

Figure 2-2 Transfer Format 8-Bit

3. HOST REQUIREMENTS

- Data framing 8-Bit and 16-bit.
- Clock phase and polarity control must be configured to SPI mode 3 before any read/write to slave.
- One GPIO capable of detecting level triggered interrupt. This signal must be maskable by software.
- For optimal performance, host SPI peripheral with DMA support for large SPI data frame sequences.

4. SPI BUS PROTOCOL

A SPI bus transaction consists of a command phase followed by a data phase. The command phase operates in the master-to-slave direction (bytes appear on MOSI line). The data phase occurs on either the MOSI line (for writes) or MISO line (for reads). See Figure 4-1.

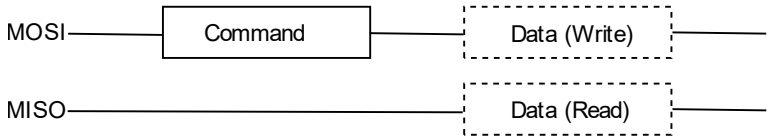

Figure 4-1 SPI Protocol

SPI slave configuration registers are 16 bits wide and can only be accessed in 16-bit mode. The read or write data after the SPI command needs to be clocked within a single CS assertion window, as shown in Figure 4-2 and Figure 4-3.

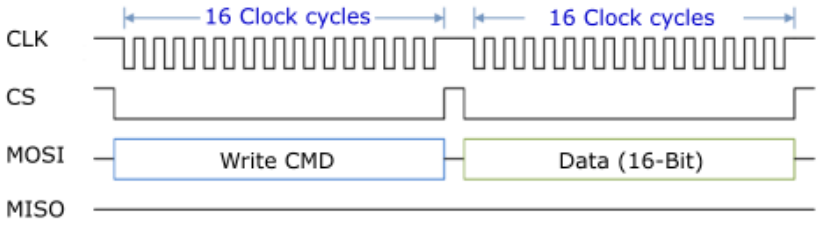

Figure 4-2 SPI Write 16 Bit

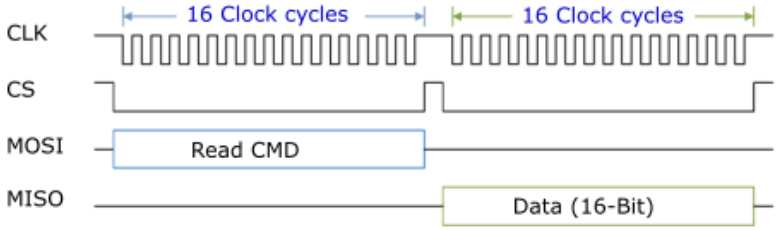

Figure 4-3 SPI Read 16 Bit

Once the read command is sent, MOSI is ignored by the slave during data phase.

4.1. SPI Command

All SPI commands are 16-bits wide and include a read/write bit and addressing mode bit followed by a 14-bit register address. Addressing mode bit is for selecting internal registers and external registers.

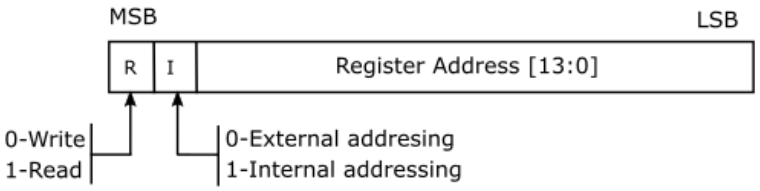

Figure 4-4 SPI Command Field

4.2. SPI Message

Length	Payload
N	n-Bytes

Table 4-1 SPI Message Format

Where payload consists of actual data plus padding bytes. Length field (2-bytes) is the actual data size (excluding 2 bytes of length and padding bytes). If length plus actual data size is not in multiples of 256, padding is necessary. Max data size transfer per transaction is 1534 bytes. See the examples shown below.

a) Actual data size is 254 bytes:

No padding required because length plus data size is 256.

Message size = Length + Payload

Message size: 256 bytes	
Length	**Payload**
N	254 Bytes

b) Actual data < 254 bytes:

Padding size = 254 – Actual data size

Message size: 256 bytes	
Length	**Payload**
N	N-bytes + (254-N) bytes padding

c) Actual data is greater than 254 and <= 510 bytes:

Padding size = 510 – Actual data size

Message size: 512 bytes	
Length	**Payload**
N	N-bytes + (510-N) bytes padding

d) Actual data size is greater than 510 and <= 766 bytes:

Padding size = 766 – Actual data size

Message size: 768 bytes	
Length	**Payload**
N	N-bytes + (766-N) bytes padding

e) Actual data size is greater than 766 and <= 1022 bytes:

Padding size = 1022 – Actual data size

Message size: 1024 bytes	
Length	**Payload**
N	N-bytes + (1022-N) bytes padding

f) Actual data size is greater than 1022 and <= 1278 bytes:

Padding size = 1278 – Actual data size

Message size: 1280 bytes	
Length	**Payload**
N	N-bytes + (1278-N) bytes padding

g) Actual data size is greater than 1278 and <= 1534 bytes:

Padding size = 1534 – Actual data size

Message size: 1536 bytes	
Length	**Payload**
N	N-bytes + (1534-N) bytes padding

h) Actual data size > 1534 bytes:

Since max data size per transaction supported is 1534 bytes, data should be split into two messages. Below example shows how to do this if the data size is 1600 bytes.

Message#1 (1536 bytes)		Message#2 (256 bytes)	
Length	**Payload**	**Length**	**Payload**
1534	1534 bytes	66	66 + (254-66) bytes padding

4.3. SPI Configuration and Data Communication

4.3.1. Host SPI Initialization

1. Configure host SPI peripheral to SPI mode 3 and set SPI clock <=24MHz. Set data width to 16-bit.
2. Configure a GPIO to receive level triggered interrupt from WL865E4-P.
3. Mask the interrupt.

4.3.2. SPI Slave Configuration

1. Reset the slave SPI block.
   a. This can be done by a write operation. Write value 0x8000 to SPI_CONFIG register (0x400). This bit is auto-clear.
   b. Read the SPI_CONFIG register to see if the write was successful.
2. Enable slave I/O block.

a. Write value 0x0080 to SPI_CONFIG register (0x400).
b. Write value 0x0081 to SPI_CONFIG register (0x400).
c. Read the SPI_CONFIG register, the read value should be 0x0081.

3. Select slave interrupts for data set indication and credit counter increment events. These are 8-bit registers, so the host writes 8-bit value.
   a. Write value 0x91 to INT_STATUS_ENABLE register (0x0418).
   b. Write value 0x01 to CPU_INT_STATUS_ENABLE register (0x0419).
   c. Write value 0x00 to ERROR_STATUS_ENABLE register (0x041A).
   d. Write value 0x10 to COUNTER_INT_STATUS_ENABLE register (0x041B).

Read all these registers to see the write was successful.

4. Interrupt the slave.
   a. Write value 0x01 to INT_WLAN register (0x0472).
5. Host now should register a callback for receive interrupt and unmask level triggered interrupt (slave to host interrupt).
6. Enable slave interrupts by writing 0x0021 into INTR_ENABLE register (0x0D00).

4.3.3. Data Write Operation (DMA)

Cmd phase: Issue read command to read the register WRBUF_SPC_AVA (0x0200).

cmd_16_bit = (SPI_WRITE_16BIT_MODE) | (SPI_INTERNAL_16BIT_MODE) | register_address

cmd_16_bit = (0x1000) | (0x4000) | 0x0200

Data phase: Read16-bit value from slave. This gives us available write buffer space in bytes before writing the actual data.

Cmd phase: Issue write command to configure the slave DMA_SIZE register (0x0100) based on the available space. If the incoming write data overflows the buffer (DMA_SIZE > WRBUF_SPC_AVA), the SPI slave will drop the incoming data entirely. A write buffer error interrupt is issued to the host. The host must insure that write data never overflows the buffer.

Data phase: Write data length value. This value should be < WRBUF_SPC_AVA.

Cmd phase: Start the DMA write command.

address = 0xFFF - (write_size - 1)

cmd_16_bit = (SPI_WRITE_16BIT_MODE) | SPI_EXTERNAL_16BIT_MODE| address

= (0x0000) | (0x0000) | address

Data phase: configure host data width to 8-bit. Write data to slave (length should be same as DMA size).

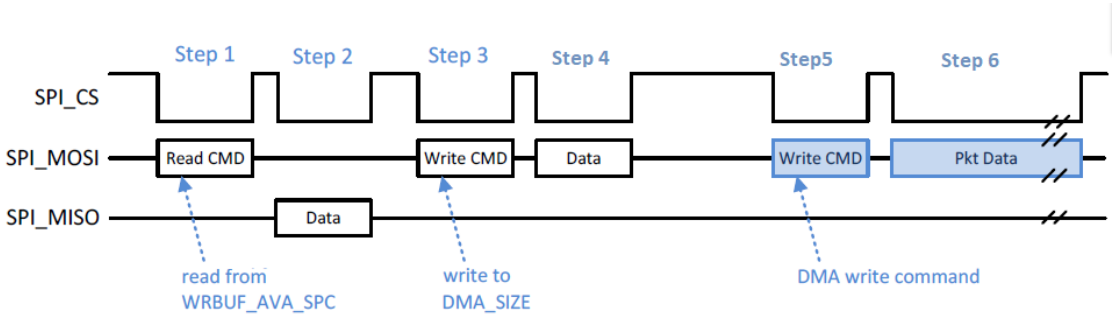

Figure 4-5 Data Write Operation

4.3.4. Data Read Operation (DMA)

1. Host receives level triggered interrupt from slave interrupt pin SDIO_D1. Masks the host interrupt. Configure host data width to 16-bit.
2. Read internal INTR_CAUSE register (0x0C00). This register holds all the pending SPI interrupts.
    a. Check if any error interrupt
    b. Check if it's credit interrupt
    c. Check if the packet_available bit is set
3. Read internal RDBUF_BYTE_AVA register (0x0300). This gives us length of available data in the read buffer. If pending size is greater than 256 bytes, then read the data in chunks of 256 bytes
4. Issue write command to configure the slave DMA_SIZE register (0x0100) with the packet size as 256.
5. Issue read command (0x8000) to read data (should be same as DMA size) by de-asserting chip select pin.
6. The packet_available bit will be cleared by slave SPI peripheral internally at the end of the DMA read. Slave interrupt pin, SDIO_D1 goes high.

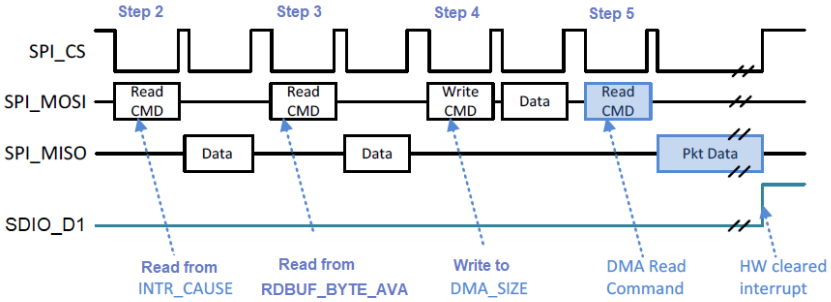

Figure 4-6 Data Read Operation

To read data, perform the following steps:

1. SPI rx processing triggered by data ready interrupt/wake up interrupt.

   This requires additional GPIO (Slave --> Master)

   Its level triggered interrupt
2. Disable the GPIO interrupt (data ready interrupt)
3. Read SPI status register

   This involves 2 bytes register read.

   |CMD 2 byte | Read DATA 2 byte |

   Therefore, this results in (4 * 8 bit) SPI transactions.
4. Read host interrupt status register

Its indirect register and requires multiple transactions.

- Program the transfer size

|CMD 2 byte | Write DATA 2 byte |

- Initiate READ operation by writing into host control register

|CMD 2 byte | Write DATA 2 byte |

- Check if the read operation is completed and data available to fetch. i.e. read status register.

If data is not ready, then repeat this operation in loop till data available or timeout.

|CMD 2 byte | Read DATA 2 byte |

- Read data from host control data register

|CMD 2 byte | Read DATA 2 byte |

5. Check the status register.

- If host interrupt status indicates flow control interrupt, then process flow control
- If interrupt status is packet available indication, then execute following steps:
- Read the register (SPI_SLV_REG_RDBUF_BYTE_AVA) to get pending bytes

|CMD 2 byte | Read DATA 2 byte |

- If pending bytes > 256 then read 256 bytes else return.

To read 256 bytes of data, perform the following steps:

1. Program the DMA transfer size as 256 bytes

|CMD 2 byte | Write DATA 2 byte |

2. Initiate read operation

|CMD 2 byte | Read 256 bytes data |

Therefore, typical read operation of 256 bytes chunk of data involves following sequence (286 * 8 clock cycles):

|CMD 2 byte | Read DATA 2 byte |CMD 2 byte | Write DATA 2 byte |CMD 2 byte | Write DATA 2 byte |CMD 2 byte | Read DATA 2 byte |CMD 2 byte | Read DATA 2 byte |CMD 2 byte | Read DATA 2 byte |CMD 2 byte | Write DATA 2 byte |CMD 2 byte | Read 256 byte data |

4.4. SPI Registers

4.4.1. Internal Registers

Table 4-2 summarizes the internal registers of WL865E4-P SPI peripheral. These registers can be accessed by SPI host at any point in time.

Register Name	Address [13:0]	Description
DMA_SIZE	0x0100	DMA Size
WRBUF_SPC_AVA	0x0200	Write buffer space available
RDBUF_BYTE_AVA	0x0300	Read buffer byte available
SPI_CONFIG	0x0400	SPI configuration
SPI_STATUS	0x0500	SPI Status
HOST_CTRL_BYTE_SIZE	0x0600	Host control register access byte size
HOST_CTRL_CONFIG	0x0700	Host control register configure
HOST_CTRL_RD_PORT	0x0800	Host control register read port

Register Name	Address [13:0]	Description
HOST_CTRL_WR_PORT	0x0A00	Host control register write port
INTR_CAUSE	0x0C00	Interrupt cause
INTR_ENABLE	0x0D00	Interrupt enable
WRBUF_WRPTR	0x0E00	Holds the write pointer for the write buffer

Table 4-2 WL865E4-P SPI Internal Registers

4.4.1.1. DMA Size (DMA_SIZE)

Address:0x0100

This register holds the DMA transfer size in terms of bytes.

Field Name	Bits	Access	Reset Value	Description
dma_size	15:12	RW	0x0	Reserved
	11:0	RW	0x1	Indicates DMA transfer size (in bytes). Programming of zero is not allowed.

4.4.1.2. Write Buffer Space Available (WRBUF_SPC_AVA)

Address:0x0200

This register holds the total number of empty space within the write buffer in terms of bytes.

Field Name	Bits	Access	Reset Value	Description
space_ava	15:12	RO	0x0	Reserved
	11:0	RO	0x0	Indicates write buffer empty space (in bytes).

4.4.1.3. Read Buffer Byte Available (RDBUF_BYTE_AVA)

Address:0x0300

This register holds the total number of data bytes within the read buffer.

Field Name	Bits	Access	Reset Value	Description
byte_ava	15:12	RO	0x0	Reserved

Field Name	Bits	Access	Reset Value	Description
	11:0	RO	0x0	Indicates the total number of data bytes within read buffer.

4.4.1.4. SPI Configuration (SPI_CONFIG)

Address: 0x0400

This register is used to configure the SPI core.

Field Name	Bits	Access	Reset Value	Description	
spi_reset	15	RW	0x0	Asserting this bit will reset the SPI core. All data within the read or write buffer will be lost. This bit is auto-clear.	
	14:13	RW	0x0	Reserved	
prefetch_priority	12:11	RW	0x0	Mailbox priority:	
				2’d0	0, 1, 2, 3
				2’d1	1, 0, 2, 3
				2’d2	2, 0, 1, 3
				3’d3	3, 0, 1, 2
miso_muxsel	10:9	RW	0x0	Select between the negative edge (default) MISO flop and delay version of positive edge MISO flop.	
spi2mbox_intr_en	8	RW	0x0	Enable interrupt going from SPI core to mailbox	
				0	No SPI interrupt will be generated to mailbox
				1	Interrupt will be generated for error conditions and packet available within read buffer
spi_io_enable	7	RW	0x0	This is the SPI version of SDIO-CCCR.*IOE1* bit. This bit should be reset to 0 after power on reset.	

Field Name	Bits	Access	Reset Value	Description	
				The SPI host can program this bit to enable the mailbox and the rest of the chip.	
	6:5	RW	0x0	Reserved	
keep_awake_for_intr	4	RW	0x0	0x0	
				0	WL865E4-P devices may enter SLEEP when a host interrupt is pending
				1	WL865E4-P devices may never enter SLEEP when a host interrupt is pending
keep_awake_en	3	RW	0x0	0	Will allow the WL865E4-P devices to sleep
				1	Wakes up the WL865E4-P (if asleep) and prevents it from going back to sleep. This programmable bit keeps the WL865E4-P awake and continues to drain power until it is eventually deasserted by the host
Swap	2	RW	0x0	0	No swapping
				1	Byte swap
16bit_mode	1	RW	0x0	If swap bit is zero, this bit has no effect. If swap bit is asserted, and:	
				0	16bit_mode = Swap every 2 bytes
				1	16bit_mode = Swap every 4 bytes

Field Name	Bits	Access	Reset Value	Description	
prefetch_mode	0	RW	0x0	Mailbox Rx FIFO prefetch scheme	
				0	Strict priority (MBOX FIFO0 has highest priority, MBOX FIFO3 has lowest priority)
				1	Round Robin

4.4.1.5. SPI Status (SPI_STATUS)

Address: 0x0500

This register indicates the current status of the SPI core.

Field Name	Bits	Access	Reset Value	Description	
mbox_flow_ctrl	15:6	RO	0x0	Reserved	
	5	RO	0x1	This bit represents whether or the mailbox is ready to accept data. Mainly used for debug.	
wrbuf_error	4	RW, W1C	0x0	Indicates an error has occurred while accessing the write buffer. A write of one clears this bit. Mainly used for debug.	
rdbuf_error	3	RW, W1C	0x0	Indicates an error has occurred while accessing the read buffer. A write of one clears this bit. Mainly used for debug.	
rtc_state	2:1	RO	0x0	Indicates the current state of the chip	
				00	Shutdown
				01	On
				10	Sleep
				11	Wakeup
host_access_done	0	RO	0x0	Indicates the host control register access (read or write) has finished. This bit will be set to zero when	

Field Name	Bits	Access	Reset Value	Description
				HOST_CTRL_CONFIG.*enable* is asserted.

4.4.1.6. Host Control Register Access Byte Size (HOST_CTRL_BYTE_SIZE)

Address: 0x0600

This register contains the size of the host control register access in terms of bytes.

Field Name	Bits	Access	Reset Value	Description	
no_addr_increment	15:7	RW	0x0	Reserved	
	6	RW	0x0	0x0	
				0	Increment the host control address by 1 for every byte
				1	Address to the host control register stays the same for the entire transfer
byte_size	0:5	RW	0x1	Indicates the size of the host control register access. The maximum value is 32. Programming of zero is not allowed.	

4.4.1.7. Host Control Register Configure (HOST_CTRL_CONFIG)

Address: 0x0700

This register holds the starting read address, direction, and enable bit for host control register access.

Field Name	Bits	Access	Reset Value	Description	
enable	15	RW	0x0	Once this bit is asserted, the SPI core will read from or write to the host control register inside mailbox. This bit is auto-clear.	
direction	14	RW	0x0	0	Host control register read

Field Name	Bits	Access	Reset Value	Description	
				1	Host control register write
address	13:0	RW	0x0400	The starting address of the host control register access. This address should range from 0x0400 to 0x07FF. Programming anything outside this range (which maps directly to mailbox FIFOs) and will cause undefined results.	

4.4.1.8. Host Control Register Read Port (HOST_CTRL_RD_PORT)

Address: 0x0800

This register provides INTERNAL access to the host control register read port. SPI host can issue a read command, with the internal bit set and the address of this register, for retrieving host control read data.

4.4.1.9. Host Control Register Write Port (HOST_CTRL_WR_PORT)

Address: 0x0A00

This register provides INTERNAL access to the host control register write port. This register (port) is write-only. SPI host can issue a write command, with internal bit asserted and the address of this register, for sending host control write data.

4.4.1.10. Interrupt Cause (INTR_CAUSE)

Address: 0x0C00

This register holds all the pending SPI interrupts. All interrupt bits can be cleared by programming a value of one, except mbox_interrupt, counter_interrupt, and local_cpu_interrupt.

- 0: No interrupt
- 1: Interrupt is pending

Field Name	Bits	Access	Reset Value	Description
wrbuf_below_watermark	15:11	RW, W1C	0x0	Reserved
	10	RW, W1C	0x0	Indicates the number of data bytes within the write buffer is below the water mark level.

Field Name	Bits	Access	Reset Value	Description
host_ctrl_rd_done	9	RW, W1C	0x0	Indicates the host control register read has completed. The read data can now be accessed through the HOST_CTRL_RD_PORT. A write of one will clear the interrupt. A write of zero has no effect.
host_ctrl_wr_done	8	RW, W1C	0x0	Indicates the host control register write has completed. A write of one will clear the interrupt. A write of zero has no effect.
all_cpu_interrupt	7	RO	0x0	This is a shadow copy of the mailbox to host Interrupt. The sum of all the mailbox internal interrupts. Please refer to the mailbox specification. This will be cleared when the mailbox interrupt line is de-asserted.
cpu_on	6	RW, W1C	0x0	The CPU has been wakened. A write of one will clear the interrupt. A write of zero has no effect.
counter_interrupt	5	RO	0x0	This is a shadow copy of the mailbox credit counter interrupt. One or more of the 8 mailbox credit counters transition from 0 to 1. This will be cleared when mailbox credit counters transition from 1 to 0.
local_cpu_interrupt	4	RO	0x0	This is a shadow copy of the mailbox CPU interrupt line. One or more of the local CPU interrupt line has been asserted. This will be cleared when the host control register (CPU_INT_STATUS) is cleared.
address_error	3	RW, W1C	0x0	A DMA access with the out of bound address has occurred. A

Field Name	Bits	Access	Reset Value	Description
				write of one will clear the interrupt. A write of zero has no effect.
wrbuf_error	2	RW, W1C	0x0	Indicates an error has occurred while accessing the write buffer. It basically means that the host is trying to send a DMA write which is larger than the available room within the write buffer. A write of one will clear the interrupt. A write of zero has no effect.
rdbuf_error	1	RW, W1C	0x0	Indicates an error has occurred while accessing the read buffer, meaning the host is sending a DMA read in which the size is larger the total available bytes within the read buffer. A write of one clears the interrupt. A write of zero has no effect.
packet_available	0	RW, W1C	0x0	Read buffer has at least one packet or the total number of available bytes has reached or exceeded the read buffer watermark. At the end of a DMA read, this interrupt will be cleared by HW. A write of one clears the interrupt. A write of zero has no effect.

4.4.1.11. Interrupt Enable (INTR_ENABLE)

Address: 0x0D00

This register is used to mask/enable interrupts going to the SPI host.

- 0: Interrupt is disabled
- 1: Interrupt is enabled

Field Name	Bits	Access	Reset Value	Description
wrbuf_below_watermark	15:11	RW	0x0	Reserved
	10	RW	0x0	Enable the write buffer below watermark interrupt.

Field Name	Bits	Access	Reset Value	Description
host_ctrl_rd_done	9	RW	0x0	Enable the host control read done interrupt.
host_ctrl_wr_done	8	RW	0x0	Enable the host control write done interrupt.
all_cpu_interrupt	7	RW	0x0	Enable the all the CPU interrupts coming from mailbox.
cpu_on	6	RW	0x0	Enable the CPU on interrupt.
counter_interrupt	5	RW	0x0	Enable the mailbox credit counter interrupt.
local_cpu_interrupt	4	RW	0x0	Enable the local CPU sourced interrupt.
address_error	3	RW	0x0	Enable DMA address error interrupt.
wrbuf_error	2	RW	0x0	Enable the write buffer error interrupt.
rdbuf_error	1	RW	0x0	Enable the read buffer error interrupt.
packet_available	0	RW	0x0	Enable the packet available interrupt.

4.4.1.12. Write Buffer Write Pointer (WRBUF_WRPTR)

Address: 0x0E00

This register holds the write pointer for the write buffer.

Field Name	Bits	Access	Reset Value	Description
write_pointer	15:11	RO	0x0	Reserved
	10:0	RO	0x0	The write pointer of the write buffer. This is used for debug only

4.4.1.13. Read Buffer Write Pointer (RDBUF_WRPTR)

Address: 0x1000

This register holds the write pointer for the read buffer.

Field Name	Bits	Access	Reset Value	Description
write_pointer	15:12	RO	0x0	Reserved
	11:0	RO	0x0	The read pointer of the write buffer; used for debug only

4.4.2. External Registers

Table 3 summarizes the host interface register. The address space is in bytes, not words.

Name	Address	Description
HOST_INT_STATUS	0x400	Address for AHB Read Access
CPU_INT_STATUS	0x401	CPU Sourced Interrupt Status
ERROR_INT_STATUS	0x402	Error or Wakeup Interrupt Status
COUNTER_INT_STATUS	0x403	Host Interface Credit Counter Interrupt
INT_STATUS_ENABLE	0x418	HOST_INT_STATUS Enable Bits
CPU_INT_STATUS_ENABLE	0x419	CPU-Sourced Interrupt Status
ERROR_STATUS_ENABLE	0x41A	Error Interrupt Status
COUNTER_INT_STATUS_ENABLE	0x41B	Credit Counter Interrupt Status
COUNT	0x420	Credit Counter Direct Access
COUNT_DEC	0x440	Credit Counter Atomic Increment
SCRATCH	0x460	Interface Scratch
INT_WLAN	0x472	Interrupt the CPU
SPI_CONFIG	0x480	SPI Slave Interface
SPI_STATUS	0x481	SPI Status

Table 4-3 WL865E4-P Host Control Registers

4.4.2.1. Pending Interrupt Status (HOST_INT_STATUS)

Offset: 0x400

Reset Value: 0x0

Access: Read-Only

Reads to this register return pending host interrupt bits. Writes to this register clear interrupt bits.

Note: Write a 1 to each bit to be cleared. All bits written as 0 do not update the interrupt status for that bit.

Field Name	Bits	Description
Reserved	7:5	Reserved
COUNTER	4	Interrupt from software controlled credit counters. Read only, see the COUNTER_INT_STATUS register for details.
MBOX_DATA	3:0	Rx Data Pending in the corresponding MBOX (FIFO is not empty). This will be cleared when the FIFO is no longer empty.

4.4.2.2. Error or Wakeup Interrupt Status (ERROR_INT_STATUS)

Offset: 0x402

Reset Value: 0x0

Access: Read/Write

Indicates a wakeup or error condition which caused the Error Interrupt in the register Pending interrupt status (HOST_INT_STATUS).

Field Name	Bits	Description
RES	7:4	Reserved
SPI	3	SPI Error Interrupt. This error can only be masked or cleared be accessing the SPI-specific registers from the SPI host
WAKEUP	2	The client transitioning to the ON state. Set as the client enters ON and can immediately accept host transactions. Writing a 1 clears the register field. Writing a 0 does not change the bit value.

Field Name	Bits	Description
RXUNDERFLOW	1	The host attempted to read a mailbox which did not contain data, and no data was available for a timeout period. This implies a software flow control error. Writing a 1 clears the register field. Writing a 0 does not change the bit value.
TXOVERFLOW	0	The host attempted to write a mailbox which was full and had no available buffer space for a timeout period. This implies a software flow control error. Writing a 1 to clears the register field. Writing a 0 does not change the bit value.

4.4.2.3. Host IF Credit Counter Interrupt (COUNTER_INT_STATUS)

Offset: 0x403

Reset Value: 0x0

Access: Read-Only

Read-only register to return counter interrupt status.

Field Name	Bits	Description
COUNTER	7:0	Each counter sets and clears its interrupt bit as follows: ▪ Set: Counter transitions from 0 > 1 ▪ Clear: Counter transitions from 1 > 0 Bit mapping is as follows: Bit [7] Counter 7 interrupt Bit [6] Counter 6 interrupt Bit [5] Counter 5 interrupt Bit [4] Counter 4 interrupt Bit [3] Counter 3 interrupt Bit [2] Counter 2 interrupt Bit [1] Counter 1 interrupt Bit [0] Counter 0 interrupt

4.4.2.4. HOST_INT_STATUS Enable Bits (INT_STATUS_ENABLE)

Offset: 0x418

Reset Value: 0x1

Access: Read/Write

Enable bits for the HOST_INT_STATUS register. Each bit enables the corresponding bit in the HOST_INT_STATUS register. Bit values:

- 0 =Interrupt is disabled
- 1 = Interrupt is enabled

Field Name	Bits	Description
ERROR	7	Enable Error Interrupt
	5:6	Reserved
COUNTER	4	Enable counter interrupt
MBOX_DATA	3:0	Enable Rx Data Pending Interrupt in the corresponding MBOX

4.4.2.5. Error Interrupt Status (ERROR_STATUS_ENABLE)

Offset: 0x41A

Reset Value: 0x0

Access: Read/Write

Enable bits for the ERROR_INT_STATUS register. Each bit enables the corresponding bit in the ERROR_STATUS register. Bit values:

- 0 = Interrupt is disabled
- 1 = Interrupt is enabled

Field Name	Bits	Description
RES	7:3	Reserved
WAKEUP	2	Wakeup interrupt enable
RXUNDERFLOW	1	RXUNDERFLOW interrupt enable
TXOVERFLOW	0	TXOVERFLOW interrupt enable

4.4.2.6. Credit Counter Interrupt Status (COUNTER_INT_STATUS_ENABLE)

Offset: 0x41B

Reset Value: 0x0

Access: Read/Write

Enable bits for the COUNTER_INT_STATUS register. Each bit enables the corresponding bit in the COUNTER_INT_STATUS register. Bit values:

- 0 = Interrupt is disabled

- 1 = Interrupt is enabled

Field Name	Bits	Description
BIT	7:0	Enable / disable bits Bit [7] Enable interrupt #7 Bit [6] Enable interrupt #6 Bit [5] Enable interrupt #5 Bit [4] Enable interrupt #4 Bit [3] Enable interrupt #3 Bit [2] Enable interrupt #2 Bit [1] Enable interrupt #1 Bit [0] Enable interrupt #0

4.4.2.7. Credit Counter Atomic Decrement (COUNT_DEC)

Offset: 0x440

Reset Value: 0x0

Access: Read/Write

Reading or writing to this register causes a unit decrement. Reads return the old value, then decrement. If the value of COUNT is 0, the decrement does not occur. If a read returns a 0, software knows the decrement does not occur. Write data is ignored.

Registers are word aligned to allow 16-bit or 32-bit accesses from the host to read or write a single atomic register. Reads or writes to non-word aligned addresses have no effect.

Reset value for all counters is 0.

Field Name	Bits	Description
RES	7:5	Reserved
DEC	4:0	[4:0] = 0x1C Decrements COUNT7 [4:0] = 0x18 Decrements COUNT6 [4:0] = 0x14 Decrements COUNT5 [4:0] = 0x10 Decrements COUNT4 [4:0] = 0xC Decrements COUNT3 [4:0] = 0x8 Decrements COUNT2 [4:0] = 0x4 Decrements COUNT1

		[4:0] = 0x0 Decrements COUNT0

4.4.2.8. Interface Scratch (SCRATCH)

Offset: 0x460

Reset Value: 0x0

Access: Read/Write

Eight scratch registers are available for host and local CPU read/ write. These registers are not atomic, data is always from the last writer.

Reset value for all scratch registers is 0.

Field Name	Bits	Description
RES	7:5	Reserved
SCRATCH	4:0	[4:0] = 0x1C Scratch register 7 [4:0] = 0x18 Scratch register 6 [4:0] = 0x14 Scratch register 5 [4:0] = 0x10 Scratch register 4 [4:0] = 0xC Scratch register 3 [4:0] = 0x8 Scratch register 2 [4:0] = 0x4 Scratch register 1 [4:0] = 0x0 Scratch register 0

4.4.2.9. SPI Slave Interface Configuration (SPI_CONFIG)

Offset: 0x480

Reset Value: See field descriptions

Access: Read/Write

Field Name	Bits	Access	Reset	Description
RES	7:5			Reserved

Field Name	Bits	Access	Reset	Description	
SPI_RESET	4	RW	0x0	Controls the reset state of SPI interface; The assert active low reset mode results in the reset of all state machines of the SPI slave. All operation registers (configuration, status, address, and count registers) retain the last value. This bit is auto-clearing.	
				0	Normal operational mode
				1	Reset SPI core
INTERRUPT_ENABLE	3	RW	0x0	Enables the SPI interface interrupt to propagate to WL865E4-P interrupt logic Interrupts are enabled, resulting in any error conditions (for example, IF error, ADDR Error, RD Error, and WR error) to assert the INTR output of SPI_SLV	
				0	SPI interrupt disabled
				1	SPI interrupt enabled
TEST_MODE	2	RW	0x0	For test mode (Loopback) operation. When set, data received is transmitted back (echo) after 1 transaction delay. Enter Test (Loop back) mode This mode is for debug purposes only. This bit should be reset for normal operation. If set, all received bits on the SPI interface (on SPI_MOSI) are sent back to the Host on the SPI Interface (SPI_MISO).	
				0	SPI normal mode
				1	SPI test mode
DATA_SIZE	1:0	RW	0x2	Selects the data size for SPI. Note: The address phase is always 16-bit. • DATA8 All data phases are 8 bits in size • DATA16 All data phases are maximum 16 bits in size. Allowable data sizes are 16 bits for internal registers access, 8 and 16bits for mailbox single reads and writes, and 16 bits for DMA transfers. 8 bits are also allowed for the last data phase of a DMA transaction • DATA32 All data phases are maximum 32 bits in size. Allowable data sizes are 16-bit for internal registers access, 8, 16, 24, and 32-bit for	

Field Name	Bits	Access	Reset	Description	
				mailbox single reads and writes, and 32-bit for DMA transfers. 8,16, and 24-bit is allowed for the last data phase of a DMA transaction	
				0	8-bit data
				1	16-bit data
				2	32-bit data (Default)
				3	Reserved

4.4.2.10. SPI Status (SPI_STATUS)

Offset: 0x481

Reset Value: See field descriptions

Access: See field descriptions

Field Name	Bits	Access	Reset	Description	
RES	7:4			Reserved	
ADDR_ERR	3	RW1C	0x0	0	No error
				1	Non-existent internal register address received or An 8- or 32-bit address/command phase received
RD_ERR	2	RW1C	0x0	Can be cleared by writing to this register. A read error is indicated by a 16 or 32-SPI_CLK data phase occurring in a DATA8 mode read, or by a 32- SPI_CLK data phase occurring in DATA16 mode.	
				0	No error
				1	Indicates read error occurred
WR_ERR	1	RW1C	0x0	0	Indicates no write error
				1	Indicates write error occurred; Writing a 1 to this register clears the bit

READY	0	RO	0x1	0	Indicates current command pending
				1	Indicates current request completed. Ready to accept SPI transaction

5. GLOSSARY AND ACRONYMS

	Description
CLK	Clock
CS	Chip Select
DMA	Direct Memory Access
GPIO	General Purpose Input Output
I/O	Input Output
MOSI	Master Output – Slave Input
MISO	Master Input – Slave Output
RTC	Real Time Clock
SPI	Serial Peripheral Interface

6. DOCUMENT HISTORY

Revision	Date	Changes
0	2019-12-17	First issue
1	2020-09-09	Correction in the Overview section of Chapter SPI Interface

SUPPORT INQUIRIES

Link to **www.telit.com** and contact our technical support team for any questions related to technical issues.

www.telit.com

Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Co., Ltd.
8th Fl., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit Wireless Solutions
Technologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil

Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit www.telit.com

Copyright © 2016, Telit

[04.2016]

Mod. 0809 2016-08 Rev.7