AN1999

USB to SPI Bridging with Microchip USB 3.1 Gen 1 Hubs

Author: *Andrew Rogers*
Microchip Technology, Inc.

INTRODUCTION

The USB to SPI bridging feature of Microchip hubs provides system designers with expanded system control and potential BOM reductions. When using Microchip's USB hubs, a separate USB to SPI device is no longer required and a downstream USB port is not lost, as occurs when a standalone USB to SPI device is implemented. This feature is available on the Microchip USB5734 and USB58xx/USB59xx USB3.1 Gen 1 Hubs.

Commands may be sent from the USB Host to the internal Hub Feature Controller device in the Microchip hub to perform the following functions:

- Enable/Disable SPI Pass-Through Interface
- SPI Write
- SPI Read

SECTIONS

REFERENCES

Consult the following documents for details on the specific parts referred to in this document:

- *Microchip USB5734 Data Sheet*
- *Microchip USB5806 Data Sheet*
- *Microchip USB5816 Data Sheet*
- *Microchip USB5826 Data Sheet*
- *Microchip USB5906 Data Sheet*
- *Microchip USB5916 Data Sheet*
- *Microchip USB5926 Data Sheet*
- *Microchip AN1903 Configuration Options for the USB5734 and USB5744*
- *Microchip AN2316 Configuration Options for the USB58xx and USB59xx*

1.0 GENERAL INFORMATION

Microchip hub USB Bridging features in Microchip hubs work via host commands sent to a Hub Feature Controller embedded within the hub located on an additional internal USB port. In order for the bridging features to work correctly, this internal Hub Feature Controller must be enabled by default. Table 1 provides details on default Hub Feature Controller setters by device.

TABLE 1: DEFAULT SETTINGS FOR THE HUB FEATURE CONTROLLER ENABLE

Part Number	Part Summary	Hub Feature Controller Default Setting
USB5734	4-Port USB3.1 Gen 1 Hub	Enabled by default
USB5806	6-Port USB3.1 Gen1 Hub	Enabled by default
USB5816	6-Port USB3.1 Gen1 Hub with Type-C™ Support on 1 Downstream Port	Enabled by default
USB5826	6-Port USB3.1 Gen1 Hub with Type-C Support on 2 Downstream Ports	Enabled by default
USB5906	6-Port USB3.1 Gen1 Hub with Type-C Support on the Upstream Port	Enabled by default
USB5916	6-Port USB3.1 Gen1 Hub with Type-C Support on the Upstream Port and 1 Downstream Port	Enabled by default
USB5926	6-Port USB3.1 Gen1 Hub with Type-C Support on the Upstream Port and 2 Downstream Ports	Enabled by default

The Hub Feature Controller is a USB2.0 WinUSB class device connected to an extra internal USB2.0 port in the hub. For example, in a four port hub, the Hub Controller is connected to port 5 of the USB2.0 portion of the hub. The Product ID (PID) for the Hub Controller is 0x2740. All bridging commands are addressed to the Hub Controller, not the Hub.

FIGURE 1-1: MICROCHIP HUB CONTROLLER EXAMPLE

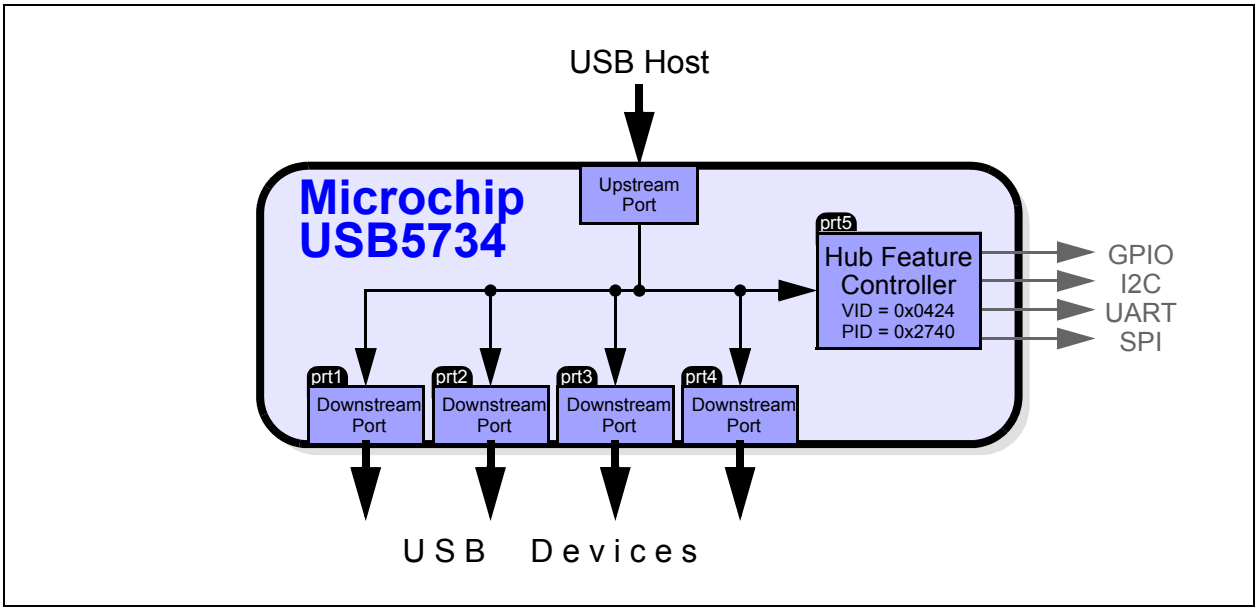
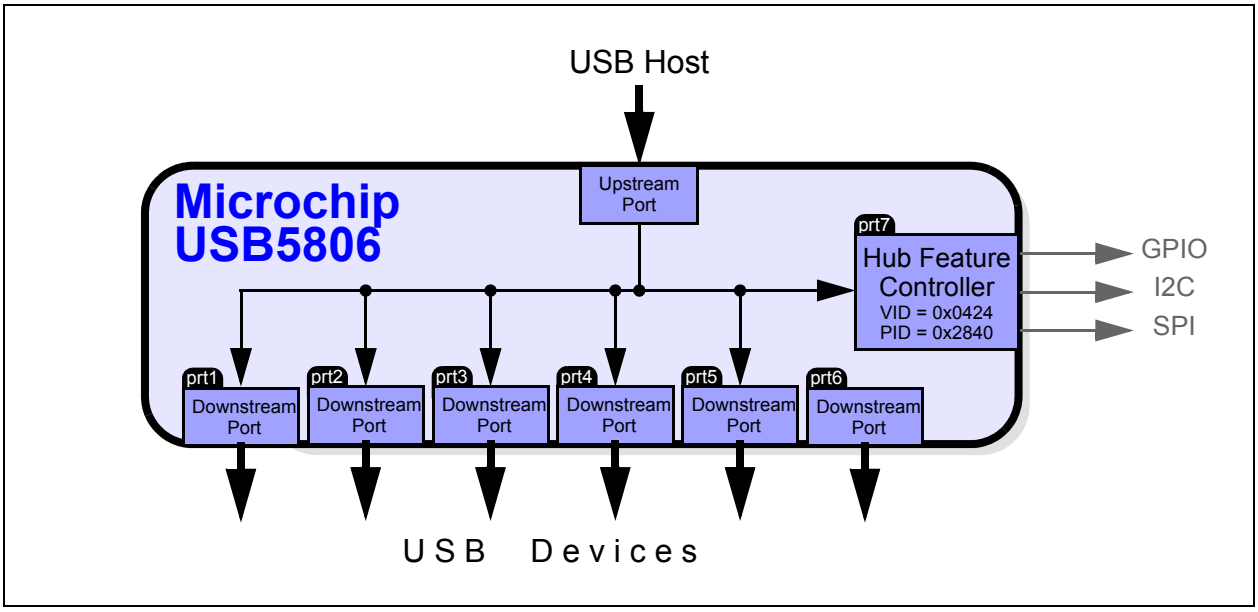


FIGURE 1-2: MICROCHIP HUB CONTROLLER EXAMPLE



1.1 SPI Bridging Commands

The following SPI Functions are supported:

- Enable/Disable SPI Pass-Through Interface
- SPI Write
- SPI Read

1.1.1 ENABLE/DISABLE THE SPI PASS-THROUGH INTERFACE

A single command to enable the SPI interface is required before performing any SPI Write or Read commands. The SPI interface operates at 60 Hz. The SPI interface may be disabled after writing/reading to the device is complete.

1.1.2 SPI WRITE/(READ)

The SPI interface works as a complete pass-through. This means that the host must properly arrange data payloads in the appropriate SPI compatible format and bit order, including the SPI slave device address. Up to 256 Bytes of data payload may be sent per SPI write command sequence.

Data may also be read from a SPI device via the SPI Write command. Up to 512 bytes of data may be read from the SPI device per read. The read data is stored in the internal registers of the hub starting at register 0x4A10. To retrieve the data, you must use a USB to Register Read command. Further details can be found in *AN 26.18 Configuration of the USB253x / USB3x13 / USB46x4.*

1.2 SPI Interface Setup Requirements

1.2.1 SPI MASTER INTERFACE

The SPI Interface always acts as a SPI master.

1.2.2 SPI MODES OF OPERATION

Both SPI Modes 0 and 3 are supported:

- Mode 0: Clock Polarity = 0, Clock Edge = 1
- Mode 3: Clock Polarity = 1, Clock Edge = 0

Dual Output Enable mode is also supported.

The default mode of operation is Mode 0 with Dual Output Enable mode disabled. If the mode of operation is to be modified, a register write to the SPI_CTL register must be performed. See *Microchip AN1903 Configuration Options for the USB5734 and USB5744* or *Microchip AN2316 Configuration Options for the USB58xx and USB59xx* for details on how to modify that register.

2.0 PART NUMBER SPECIFIC INFORMATION

2.1 Part Summary

The following tables display the SPI interface pins by part number as well as any notes on those pins.

2.2 USB5734

TABLE 2: USB5734 SPI INTERFACE PINS

Pin #	Name	Notes
42	SPI_CLK/(I2C_SLCV_CFG0)	SPI Clock (60 MHz) I2C_SLV_CFG0 strap is sampled at power on and it must have no pull-up/down resistors detected at power on to enter SPI Mode
43	SPI_DO/(I2C_SLCV_CFG1)	SPI Data Out I2C_SLV_CFG1 strap is sampled at power on and it must have no pull-up/down resistors detected at power on to enter SPI Mode
44	SPI_DI	SPI Data In
45	SPI_CE_N	SPI Chip Enable

2.3 USB58xx/USB59xx

TABLE 3: USB58XX/USB59XX I2C INTERFACE PINS

Pin #	Name	Notes
65	SPI_CLK/C_ATTACH3/GPIO4	SPI Clock (60 MHz) When implementing the SPI bridging feature, the hub should not be configured for Type-C operation on Downstream Ports 2 or 3.
66	SPI_DO/C_ATTACH2/GPIO5	SPI Data Out When implementing the SPI bridging feature, the hub should not be configured for Type-C operation on Downstream Ports 2 or 3.
67	SPI_DI/GPIO9/**CFG_BC_EN**	SPI Data In This pin is also used as the Battery Charging configuration strap. It is generally not recommended to use this configuration strap when implementing the SPI bridging feature.
68	SPI_CE_N/GPIO7/**CFG_NON_REM**	SPI Chip Enable This pin is also used as the Port Non-Removable configuration strap. It is generally not recommended to use this configuration strap when implementing the SPI bridging feature.

3.0 PROTOUCH2 DLL IMPLEMENTATION

The simplest method for implementing the USB to SPI bridging functions is to use the publicly available ProTouch2 DLL library. The PT2 DLL library is available for Windows operating systems. Visit the product page for any of the hubs listed in this document on microchip.com to download the ProTouch2 package. Using the libraries available in the ProTouch2 DLL library, the bridging features can be implemented in C-code.

The PT2 DLL library package contains the following:

- Protouch2 DLL User Guide: Detail description of how to use the SDK and call each function
- Protouch2 Release Notes:
- Library Files:
  - For Windows: A “.dll” and a “.lib” file
- Example code

3.1 Commands included in the DLL Library

- **MchpUsbSpiSetConfig:** Enables or Disables the SPI interface.
- **MchpUsbSpiFlashWrite:** Write up to 255 Bytes of data to an SPI slave device.
- **MchpUsbSpiFlashRead:** Read up to 255 Bytes of data from an SPI slave device.
- **MchpUsbSpiFlashRead:** Read/Write from a SPI slave device.

For additional details on how to use the PT2 DLL library to implement USB to SPI bridging, download the ProTouch2 package and read the User’s Manual.

4.0 MANUAL IMPLEMENTATION

The USB to SPI bridging features may be implemented at the lowest level if you have the ability to build USB packets. This approach is required if you are not using a Windows or Linux host system and cannot use the ProTouch2 DLL library.

The details of the SPI pass-through control packets are shown below.

4.1 Enable SPI Pass-Through Interface Command

The follow SETUP packet command is required to enable the SPI pass-through interface. The interface must be enabled before any Write or Read commands may be performed. Note that there is no data phase to this USB transaction,

TABLE 4: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x60	Register read command: CMD_SPI_ENTER_PASSTHRU
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x00	No data stage

4.2 SPI Write Command

This command is used to write data to or read data from a SPI peripheral connected to the USB hub.

TABLE 5: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x61	Register read command: CMD_SPI_WRITE
wValue	0xXXXX	The total length of data to be sent to the SPI peripheral (the size of the data following the SETUP packet).
wIndex	0x0000	Reserved
wLength	0xNN	The number of bytes the SPI interface will return for the command sent

The maximum amount of data that can be read from one USB command is 512 Bytes by specifying wValue = 517 and wLength = 5.

The maximum amount of data that can be written to a SPI peripheral is 256 Bytes by specifying wValue = wLength = 260.

4.2.1 SPI WRITE USB TRANSACTION SEQUENCE:

1. SETUP PACKET: To send ‘Write Enable’ OpCode to SPI ROM (wValue = wLength = 1)
2. DATA: 0x06 (opcode for ‘Write Enable).
3. STATUS: An IN-Zero Length Packet is sent from hub.
4. SETUP PACKET: To send data payload
5. DATA: EP0 Data to SPI ROM with 0x02 + 24bit SPI address + Data Stream
6. STATUS: If an IN-Zero Length Packet is sent from hub, transfer was a success. If an IN-STALL packet is sent from hub, there was an error during transfer.

4.2.2 SPI READ USB TRANSACTION SEQUENCE:

1. SETUP PACKET: As defined above.
2. DATA: EP0 OUT Data to SPI ROM with 0x0B + 24Bit SPI Address + 0x00 (dummy byte).
3. STATUS: If an IN-Zero Length Packet is sent from hub, transfer was a success. If an IN-STALL packet is sent from hub, there was an error during transfer, likely due to missing ACK from the SPI slave.
4. Perform Configuration Register Read on hub starting at register 0x4A10 to retrieve read data. See *AN 26.18 Configuration of the USB253x / USB3x13 / USB46x4*

4.3 Disable SPI Pass-Through Interface Command

The follow SETUP packet command is required to enable the SPI pass-through interface. Note that there is no data phase to this USB transaction,

TABLE 6: USB SETUP COMMAND

Setup Parameter	Value	Description
bmRequestType	0x41	Vendor specific command, Host-to-device data transfer
bRequest	0x62	Register read command: CMD_SPI_ENTER_PASSTHRU
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x00	No data stage

5.0 EXAMPLES

5.1 Write 512 Bytes to a SPI ROM

1. Enable the SPI Pass-Through Interface

TABLE 7: ENABLE SPI INTERFACE SETUP COMMAND

Setup Parameter	Value
bmRequestType	0x41
bRequest	0x60
wValue	0x0000
wIndex	0x0000
wLength	0x00

2. Send a SPI Write/Read Command to read 512 Bytes of data.

TABLE 8: SPI WRITE SETUP COMMAND

Setup Parameter	Value
bmRequestType	0x41
bRequest	0x61
wValue	0x0205 (517)
wIndex	0x0000
wLength	0x0005

3. EP0 OUT data = 0x0B, 0xXX, 0xYY, 0xZZ, 0x00, 0xXX, 0xYY, 0xZZ.

Note: 0xXX, 0xYY, 0xZZ is the 24 Bit physical SPI address of the SPI peripheral.

4. Read response via USB to Configuration Register Read from register 0x4A10. Further details can be found in *AN 26.18 Configuration of the USB253x / USB3x13 / USB46x4.*
5. Do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through Command
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

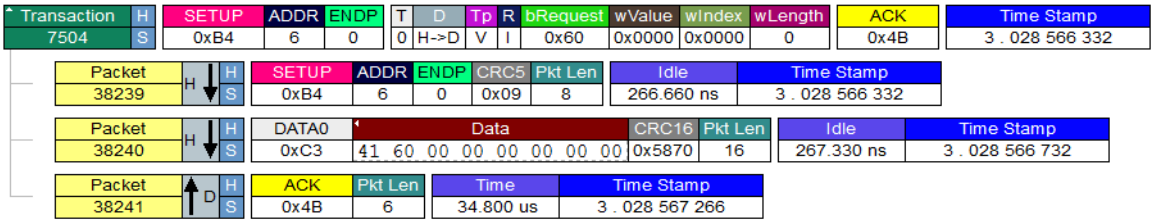
5.2 Enable the SPI Pass-Through Interface

1. **Command Phase (SETUP Transaction):** Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to enable the SPI pass-through interface:

TABLE 9: EXAMPLE I2C WRITE SETUP PACKET

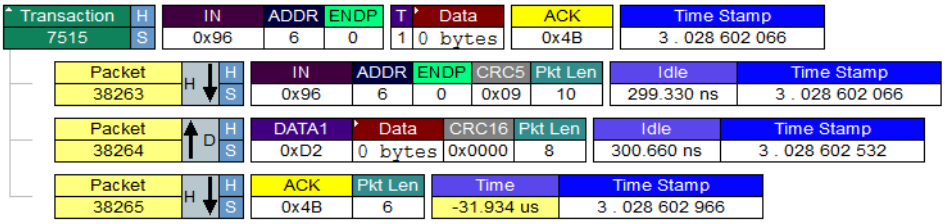
Field	Value	Note
bmRequestType	0x41	
bRequest	0x60	
wValue	0x0000	
wIndex	0x0000	
wLength	0x0000	

FIGURE 1: SETUP TRANSACTION EXAMPLE



2. **Status (IN Transaction):** The host sends an IN packet to the Hub Feature Controller, to which the Hub Feature controller replies with a zero data length packet. The host ACKs to complete the bridging command.

FIGURE 2: IN TRANSACTION EXAMPLE



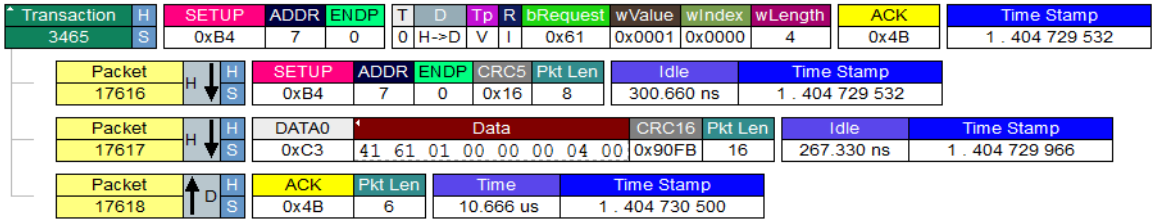
5.3 Read the JEDEC ID from an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** The JEDEC ID gives manufacturer information and memory information. This example shows how to read the JEDEC ID from an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller send a SPI Write command to the attached SPI device.

TABLE 10: SPI JEDEC ID READ SETUP PACKET EXAMPLE

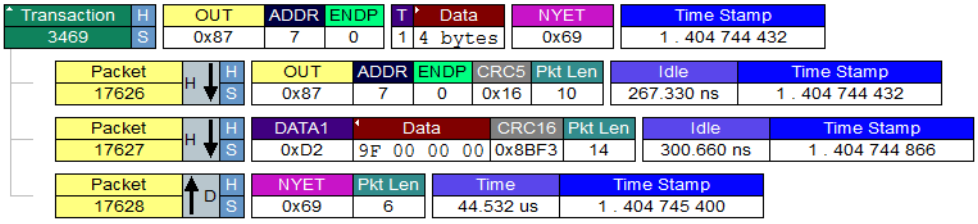
Field	Value	Note
bmRequestType	0x41	
bRequest	0x61	
wValue	0x0001	
wIndex	0x0000	
wLength	0x0004	

FIGURE 3: SPI JEDEC ID READ SETUP TRANSACTION EXAMPLE



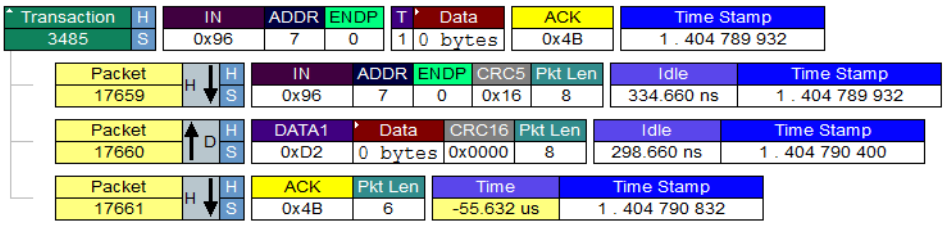
2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by the data bytes of length wLength. In this example, the 0x9F data is the opcode for reading a JEDEC ID in a SPI device. Hub Feature Controller responds with a NYET after receiving the data.

FIGURE 4: SPI JEDEC ID IN TRANSACTION EXAMPLE



3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 5: SPI JEDEC ID OUT TRANSACTION EXAMPLE

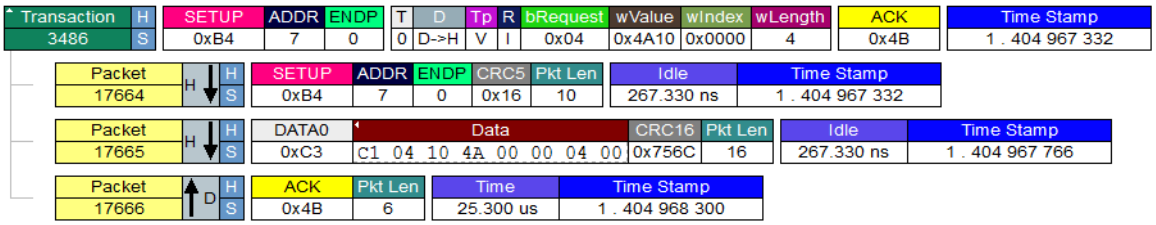


4. **Retrieve the Returned Data (SETUP Transaction 2):** The SPI device will respond to the opcode command and the returned data will be stored in the hub's internal register starting at address 0x4A10. A USB to Register Read command can retrieve the data. This command is setup as:

TABLE 11: SPI JEDEC ID REGISTER READ SETUP PACKET EXAMPLE

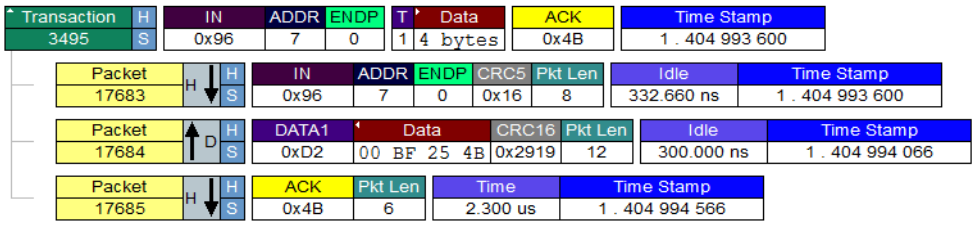
Field	Value	Note
bmRequestType	0xC1	
bRequest	0x04	
wValue	0x4A10	The hub's internal register address
wIndex	0x0000	
wLength	0x0004	A JEDEC ID request will return 4 bytes. The first byte will be a dummy 0x00.

FIGURE 6: SPI JEDEC ID REGISTER READ SETUP TRANSACTION EXAMPLE



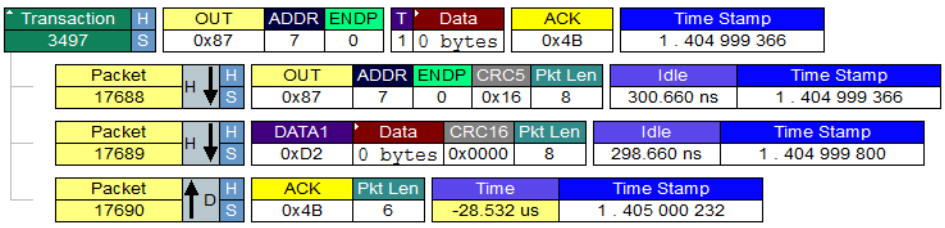
5. **Data Phase 2 (IN Transaction 2):** Host sends an IN packet to retrieve the data from the 0x4A10 register. In this example, the JEDEC ID that is returned is 0xBF, 0x25, 0x4B. The host replies with an ACK after receiving the data.

FIGURE 7: SPI JEDEC ID REGISTER READ IN TRANSACTION EXAMPLE



6. **Status Phase 2 (OUT Transaction 2):** Host sends an OUT packet followed by a zero length data packet. The Hub Feature Controller ACKs to complete the bridging command.

FIGURE 8: SPI JEDEC ID REGISTER READ OUT TRANSACTION EXAMPLE



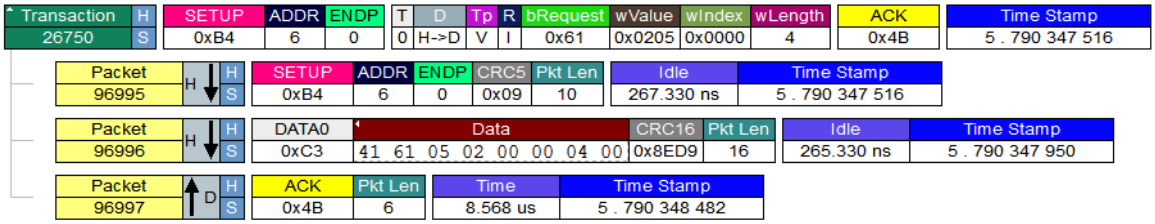
5.4 Read 512 Bytes from an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** This example shows how to perform a block read of 512 bytes (the maximum number of bytes per command) from an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller send a SPI Write command to the attached SPI device

TABLE 12: SPI BLOCK READ COMMAND SETUP PACKET EXAMPLE

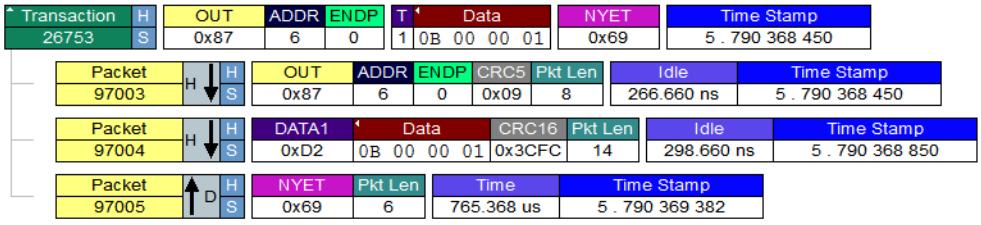
Field	Value	Note
bmRequestType	0x41	
bRequest	0x61	
wValue	0x0205 (517)	The first 5 Bytes of any SPI read must be ignored; Therefore, 5 must be added to the number of Bytes to be read.
wIndex	0x0000	
wLength	0x0004	

FIGURE 9: SPI BLOCK READ COMMAND SETUP TRANSACTION EXAMPLE



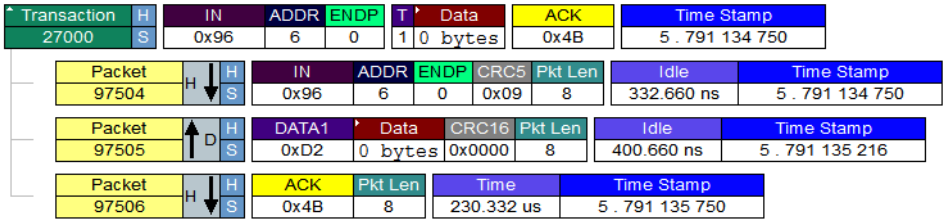
2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by the data bytes of length wLength. In this example, the 0x0B is the opcode for a block read in this specific SPI device, the next 0x00,0x00,0x01 is the register address to begin reading from the SPI device. The Hub Feature Controller responds with a NYET after receiving the data.

FIGURE 10: SPI BLOCK READ COMMAND IN TRANSACTION EXAMPLE



3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 11: SPI BLOCK READ COMMAND OUT TRANSACTION EXAMPLE

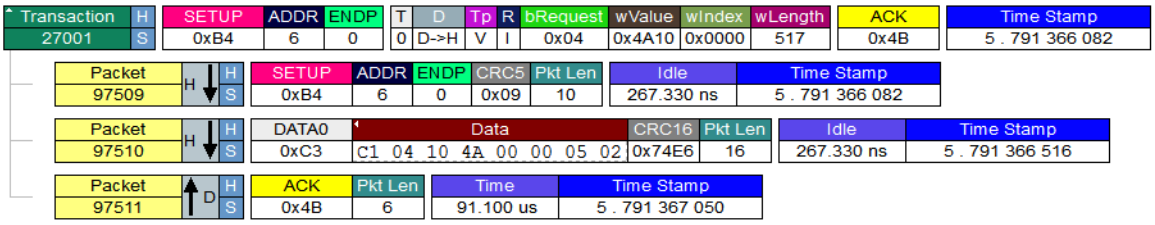


4. **Retrieve the Returned Data (SETUP Transaction 2):** The SPI device will respond to the opcode command and the returned data will be stored in the hub's internal register starting at address 0x4A10. For a SPI Block Read, the first 5 bytes must be ignored. A USB to Register Read command can retrieve the data. This command is setup as:

TABLE 13: REGISTER READ SETUP PACKET EXAMPLE

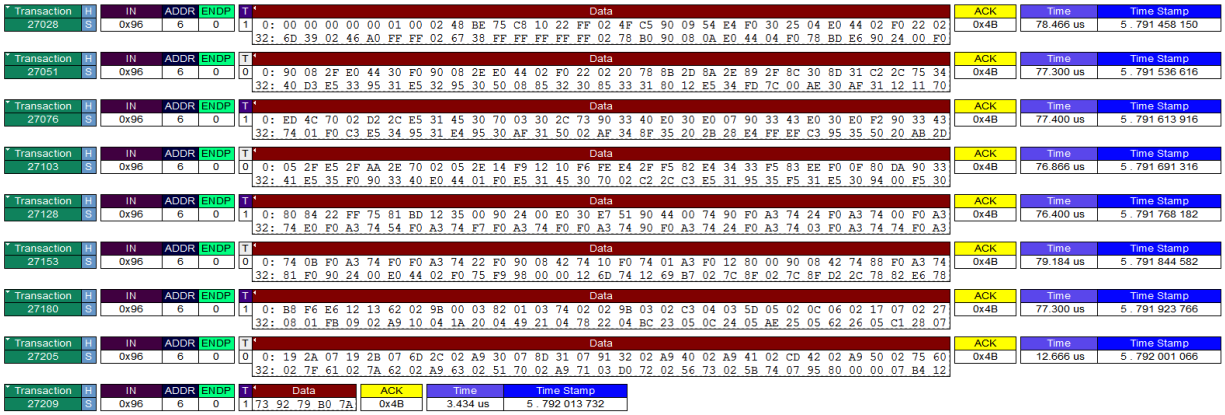
Field	Value	Note
bmRequestType	0xC1	
bRequest	0x04	
wValue	0x4A10	The hub's internal register address
wIndex	0x0000	
wLength	0x0205	517 Bytes will be read (512 + 5 dummy bytes)

FIGURE 12: REGISTER READ SETUP TRANSACTION EXAMPLE



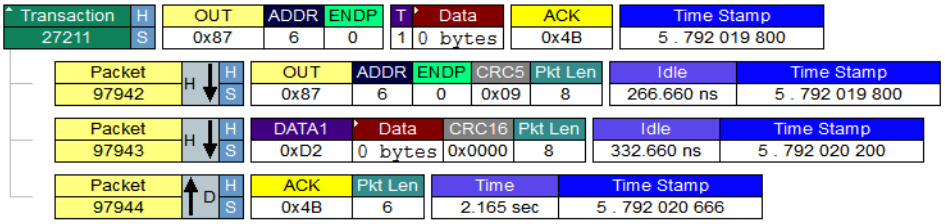
5. **Data Phase 2 (IN Data Payload Transaction):** Hub sends a series of IN packets (64 Bytes per packet) until all 517 bytes are read. The first 5 bytes should always be ignored.

FIGURE 13: REGISTER READ IN TRANSACTION EXAMPLE



6. **Status Phase 2 (OUT Transaction 2):** Host sends an OUT packet followed by a zero length data packet. The Hub Feature Controller ACKs to complete the bridging command.

FIGURE 14: REGISTER READ OUT TRANSACTION EXAMPLE



7. After completing the SPI Read, do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through command.
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

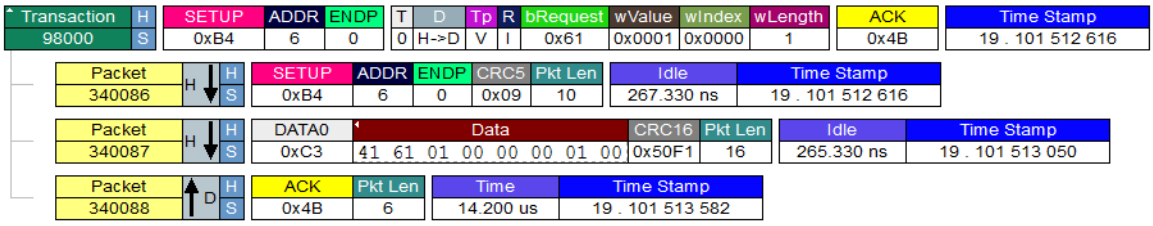
5.5 Write 256 Bytes to an attached SPI Device

1. **Command Phase 1 (SETUP Transaction 1):** This example shows how to perform a block write of 256 bytes to an attached SPI device. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to enable the SPI write feature.

TABLE 14: SPI WRITE ENABLE SETUP PACKET EXAMPLE

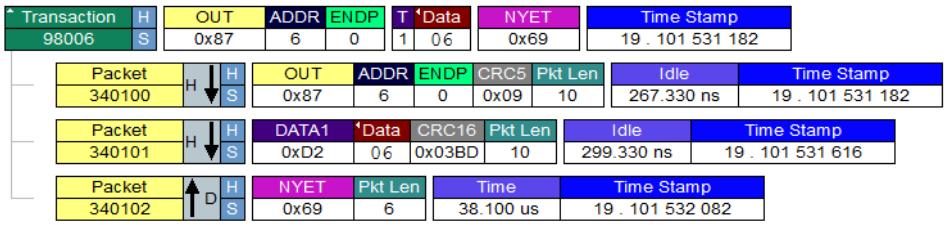
Field	Value	Note
bmRequestType	0x41	
bRequest	0x61	
wValue	0x0001	
wIndex	0x0000	
wLength	0x0001	

FIGURE 15: SPI WRITE ENABLE SETUP TRANSACTION EXAMPLE



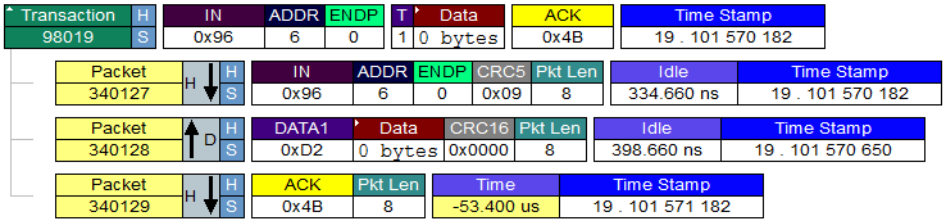
2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by a single byte data payload of 0x06. 0x06 is the SPI write enable command. The Hub Feature Controller responds with a NYET after receiving the data.

FIGURE 16: SPI WRITE ENABLE TRANSACTION EXAMPLE



3. **Status Phase 1 (IN Transaction 1):** Host sends an IN packet to complete the USB Transfer. Hub Feature Controller responds with a zero length data packet. The host ACKs to complete the bridging command.

FIGURE 17: SPI WRITE DATA OUT TRANSACTION EXAMPLE

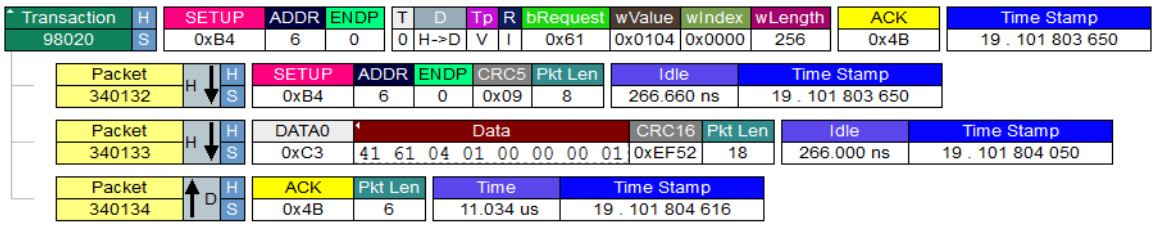


4. **Send the SPI Write Data Payload (SETUP Transaction 2):** The SPI device is now ready to receive the data payload. For a SPI Block Write of 256 bytes, the SETUP command is:

TABLE 15: SPI WRITE DATA SETUP PACKET EXAMPLE

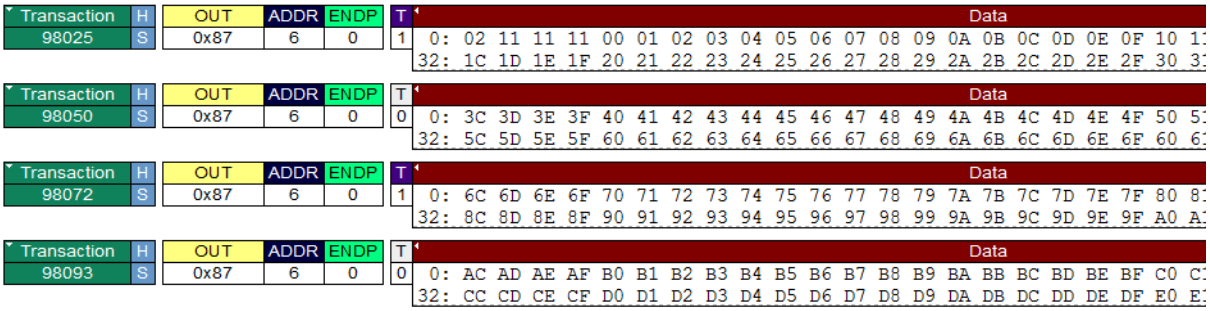
Field	Value	Note
bmRequestType	0x41	
bRequest	0x61	
wValue	0x0104 (260)	The 256 byte data payload + 4 extra command bytes
wIndex	0x0000	
wLength	0x0100 (256)	The 256 byte data payload

FIGURE 18: SPI WRITE DATA TRANSACTION EXAMPLE



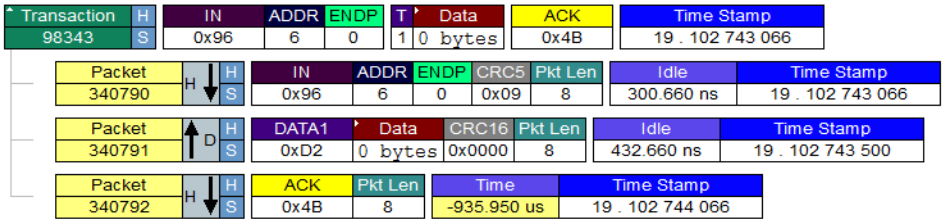
5. **Data Phase 2 (OUT Data Payload Transaction):** Host sends a series of OUT packets (64 Bytes per packet) until all 256 + 4 command bytes are sent. The first 4 bytes must be 0x02, 0xXX, 0xYY, 0xZZ where 0xXX, 0xYY, 0xZZ is the 24 bit physical address of the SPI Flash. In this example, the SPI address is 0x111111.

FIGURE 19: SPI WRITE DATA OUT TRANSACTIONS EXAMPLE



6. **Status Phase 2 (IN Transaction 2):** Host sends an IN packet and the hub responds with a zero length data packet. The Host ACKs to complete the bridging command.

FIGURE 20: SPI WRITE DATA IN TRANSACTION EXAMPLE



7. After completing the SPI Write, do one of the following:
   - Close the SPI Interface with the Disable SPI Pass-Through command.
   - Wait for the manufacturer specified time before performing another Read/Write command.
   - Send RDSR commands until the BUSY field is cleared before performing another Read/Write command.

APPENDIX A: APPLICATION NOTE REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00001999B (12-07-16)	All	Fixed references to *AN2316 Configuration Options for the USB58xx and USB59xx* throughout.
		Trademark and Sales Listing pages updated.
		Updated minor formatting and grammar issues throughout.
DS00001999A (09-11-15)	All	Initial release.

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

ISBN: 978-1-5224-1171-0

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

France - Saint Cloud
Tel: 33-1-30-60-70-00

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820