AN2430

USB-to-SPI Bridging with Microchip USB471x and USB49xx Hubs

Author: *Shiva Balasubramanian*
Microchip Technology Inc.

INTRODUCTION

The USB-to-SPI Bridging feature gives system designers using Microchip hubs expanded system control and potential BOM reduction. The use of 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 Microchip hubs which contain the internal Hub Feature Controller and an SPI interface. These hubs include USB4712, USB4714, USB4715, USB4912, USB4914, USB4916, USB4925, and USB4927.

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

- Get hub information
- Reset the hub
- Force boot from internal ROM
- Enable SPI pass-through interface
- Disable SPI pass-through interface
- SPI pass-through write/read

SECTIONS

REFERENCES

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

- *USB4712 Data Sheet*
- *USB4714 Data Sheet*
- *USB4715 Data Sheet*
- *USB4912 Data Sheet*
- *USB4914 Data Sheet*
- *USB4916 Data Sheet*
- *USB4925 Data Sheet*
- *USB4927 Data Sheet*
- *SST26VF016B Data Sheet*
- *Configuration of the USB471x and USB49xx Application Note*

GENERAL INFORMATION

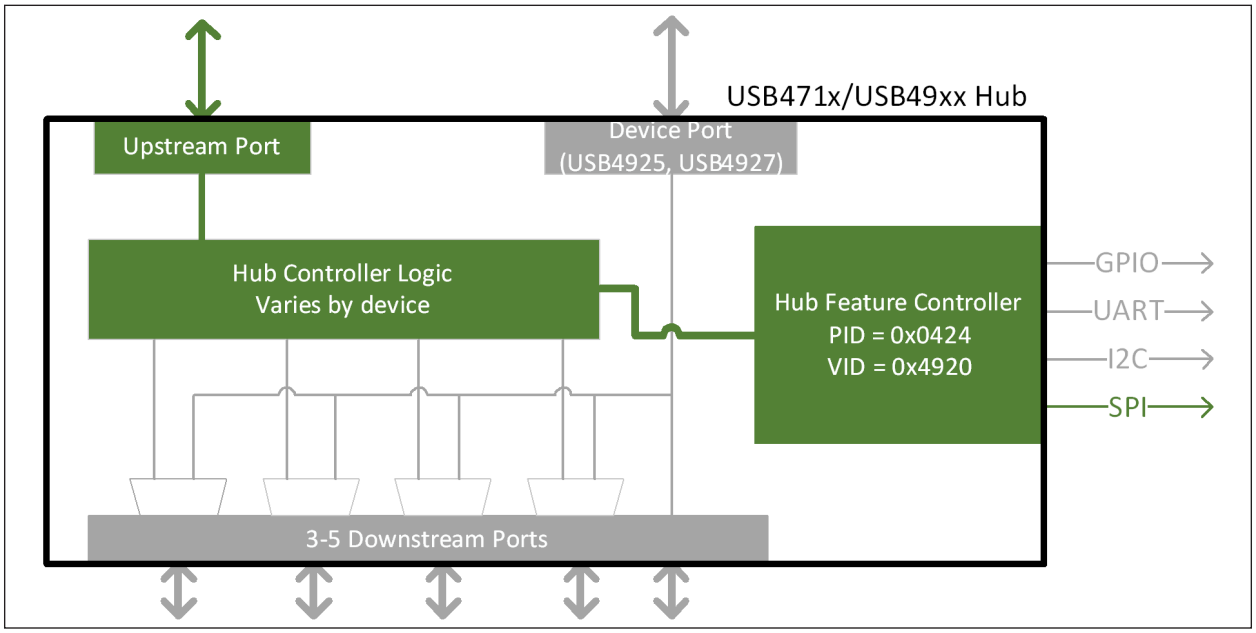
The USB-to-SPI Bridging features in Microchip hubs work via host commands sent to a Hub Feature Controller embedded within the hub located on an additional internal USB port. In order for the bridging features to work correctly, this Internal Hub Feature Controller must be enabled by default. The SPI interface is always enabled after reset. It can be disabled by setting the SPI_MASTER_DIS bit in the PAD_MUX_CTL register. See Table 1 below for details on default Hub Feature Controller settings by part number.

TABLE 1: DEFAULT SETTINGS FOR HUB FEATURE CONTROLLER ENABLE

Part Number	Part Summary	Hub Controller Default Setting
USB4712	One USB upstream port and one USB Flex port	Enabled
USB4715	One USB upstream port and four USB Flex ports	Enabled
USB4912	One USB upstream port and two USB CarPlay ports	Enabled
USB4914	One USB upstream port, two USB CarPlay ports, and one non-removable standard USB port	Enabled
USB4916	One USB upstream port, four USB CarPlay ports, and one non-removable standard USB port	Enabled
USB4925	One USB upstream port, one secondary USB downstream port, two USB CarPlay ports, and one non-removable standard USB port	Enabled
USB4927	One USB upstream port, one secondary USB downstream port, four USB CarPlay ports, and one non-removable standard USB port	Enabled

The Hub Feature Controller is connected to an extra internal port in the hub. (See Figure 1.) For example, in a four-port hub, the Hub Feature Controller is connected to port 5. The Product ID (PID) varies as per the SKU selected. All bridging commands are addressed to the Hub Feature Controller and not the hub.

FIGURE 1: MICROCHIP HUB FEATURE CONTROLLER BLOCK DIAGRAM



SPI Bridging Commands

The following SPI functions are supported:

- Get Hub Information
- Reset the Hub
- Force Boot from Internal ROM
- Disable the SPI Pass-Through Interface
- Enable the SPI Pass-Through Interface
- SPI Pass-Through Write/Read

GET HUB INFORMATION

The host can get information about the hub by issuing the GET_HUB_INFO command. In response, the hub sends a packet which contains information about the device revision, firmware version, and boot mode.

RESET THE HUB

The host can soft reset the hub externally by issuing CMD_DEV_RESET command. This forces the hub firmware to start execution from 0x000000 and go through the boot sequence again.

FORCE BOOT FROM INTERNAL ROM

In cases where the hub is executing out of an external SPI ROM and the host wants to perform SPI pass-through transfers with the SPI ROM, this command sequence can be used to force the hub to boot and execute from the internal ROM.

ENABLE THE SPI PASS-THROUGH INTERFACE

To acquire the SPI interface, the host must send a CMD_SPI_ENTER_PASSTHRU SETUP packet before performing any SPI Write/Read commands. The SPI interface may operate at either 30 MHz or 60 MHz (based on pin strapping).

DISABLE THE SPI PASS-THROUGH INTERFACE

The SPI pass-through interface can be disabled after read/write operations by sending a CMD_SPI_EXIT_PASSTHRU SETUP packet.

SPI PASS-THROUGH WRITE/READ

The SPI pass-through interface allows single/multi-byte write access and read access. In case of these operations, the SPI interface functions as a complete pass-through, which means any SPI data sent as a payload in the USB transfer gets transferred on to the SPI lines directly. Therefore, 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 can be written to an SPI peripheral using an SPI Write command sequence.

Data can also be read from an SPI peripheral using a combination of SPI write/read pass-through transfers. The host first needs to send a SETUP packet which informs the hub about the number of bytes to be read. Following this command, the requested data bytes are stored by the hub in an internal register at 0xBFD22310. The SPI Read command sequence can then be initiated by the host to retrieve the data. Up to 512 bytes of data can be read per SPI Read command sequence.

SPI Interface SETUP Requirements

SPI MASTER INTERFACE

The SPI interface always acts as an SPI master.

SELECTING SPI FREQUENCY

The SPI interface may operate at either 30 MHz or 60 MHz. The speed is selected by pin strapping the SPI_SD_SEL pin (which is the SPI_DO pin during runtime) and is detected at power-on or at the end of reset. The strapping options are:

- GND (Logical `0`) = 30 MHz
- 3.3V (Logical `1`) = 60 MHz

When running in 30 MHz mode, the SPI_DO/SPI_SPD_SEL pin drives to 0V during a Suspend state. When running in 60 MHz mode, the SPI_DO/SPI_SPD_SEL pin tristates during a Suspend state.

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 the Dual Output Enable mode disabled. If the mode of operation is to be modified, a register write to the SPI_CTL (0xBF802400) register must be performed.

PART NUMBER-SPECIFIC INFORMATION

Part Summary

In USB471x/USB49xx devices, SPI interface signals are associated with dedicated pins. See Table 2 for information on the pins.

TABLE 2: USB471X AND USB49XX SPI INTERFACE PIN NUMBERS

Device	SPI_DI	SPI_CE_EN	SPI_DO (Note 1)	SPI_CLK	SPI_D2	SPI_D3
USB4712	23	24	22	21	26	27
USB4715	23	24	22	21	26	27
USB4912	23	24	22	21	26	27
USB4914	23	24	22	21	26	27
USB4916	31	32	30	29	34	35
USB4925	23	24	22	21	26	27
USB4927	31	32	30	29	34	35

Note 1: This pin is also the SPI_SPD_SEL strap which selects between 30 MHz and 60 MHz speed of operation.

MICROCHIP SOFTWARE SOLUTIONS

Microchip currently offers two publicly available software solutions to facilitate USB-to-SPI Bridging in a USB471x/USB49xx series hub on Windows and Linux.

MPLAB Connect Configurator Package (For Windows)

The MPLAB Connect Configurator (MPLABCC) package consists of both GUI-based and CLI-based tools which support USB-to-SPI Bridging in a standalone form. In addition, it contains a Dynamically Linked Library (DLL) for Windows which can be used for implementing USB-to-SPI Bridging feature in custom applications using C programming language. The MPLABCC DLL consists of the following:

- User's guide: A detailed description of how to use the DLL API to call each function
- Release notes
- Library files: A .dll and a .lib file
- Example code

Application Code Examples (For Linux)

For implementing USB-to-SPI Bridging on Linux, you can use one of the following USB471x/USB49xx Linux Application Code Examples (ACEs):

- ACE002 USB Pseudo-OTP Programmer: This ACE demonstrates how to program the pseudo-OTP and read it back using USB-to-SPI Bridging commands.
- ACE003 USB SPI Firmware Programmer: This ACE demonstrates how to program the hub firmware to different types of Microchip SPI flash memories using USB-to-SPI Bridging commands.
- ACE004 USB-to-SPI Bridging: This ACE demonstrates how to transfer data to and from the SPI interface in USB471x/USB49xx using USB-to-SPI Bridging commands.

These application examples use libusb library for Linux to build and send USB packets as described in Low-Level Implementation. Each ACE is a full-feature code example that consists of:

- Example code with minimal abstraction and in-line comments describing the various steps involved
- A Makefile
- README

These ACEs can be used as standalone applications or can be integrated into existing applications.

Note:	Visit the product page on www.microchip.com for any of the hubs listed in this document to download the software solution for the desired operating system.

LOW-LEVEL 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 solutions described in Microchip Software Solutions.

The details of these low-level USB packets are shown below.

Get Hub Information

The following GET_HUB_INFO SETUP packet needs to be sent to get the hub-related information.(See Figure 2.) Refer to Table 3 for the USB SETUP command details.

TABLE 3: USB SETUP COMMAND: GET HUB INFORMATION

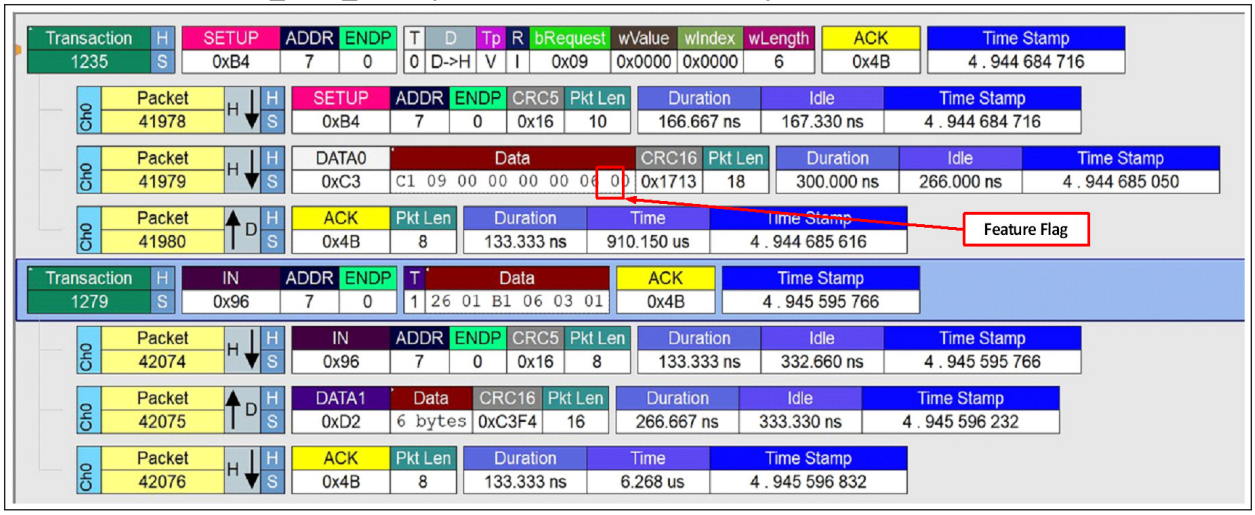
SETUP Packet	Value	Description
bmRequestType	0xC0	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x09	GET_HUB_INFO command
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x0006	Size of the HUB_INFO structure

The hub responds with a Hub Information Structure (HUB_INFO) during the DATA phase. (See Figure 2.) The details on the hub information structure (DATA phase) are in Table 4.

TABLE 4: HUB INFORMATION STRUCTURE (DATA PHASE)

Offset	Size	Field	Description
0x00	2	FW Revision	Firmware revision
0x02	1	Device Revision	Silicon mask revision
0x03	2	ASIC Type	ASIC type: 0xA0 0xB0
0x05	1	Feature Flag	Bit 0: Current code execution area, 0: ROM, 1: SPI

FIGURE 2: GET_HUB_INFO (SETUP AND DATA PHASE)



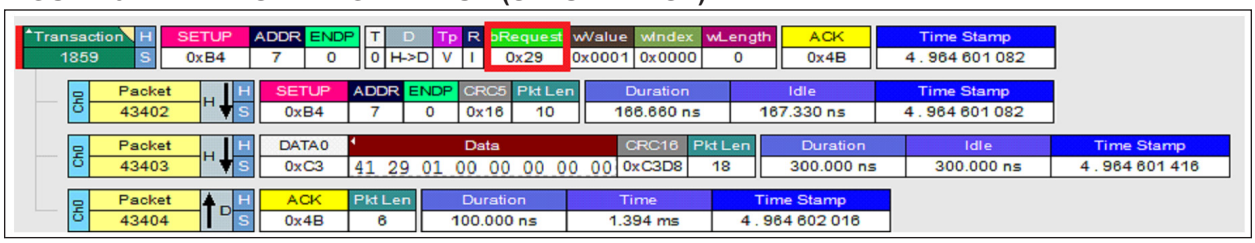
Reset the Hub

The hub can be soft reset externally from the host by issuing the CMD_DEV_RESET command. This command does not have a DATA phase. (See Figure 3.) The existing device handle becomes invalid when the hub is reset. The host must acquire a new device handle by reopening the hub. See Table 5 for the USB SETUP command information.

TABLE 5: USB SETUP COMMAND: RESETTING THE HUB

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x29	CMD_DEV_RESET Command
wValue	0x0001	Reserved
wIndex	0x0000	Reserved
wLength	0x0000	No DATA phase

FIGURE 3: RESETTING THE HUB (SETUP PHASE)



Force Boot from Internal ROM

If the hub is executing out of external SPI ROM and the host needs to perform SPI pass-through operations on the SPI ROM, then the hub needs to be forced to execute from internal ROM for the operations to succeed. This can be achieved by the following command sequence:

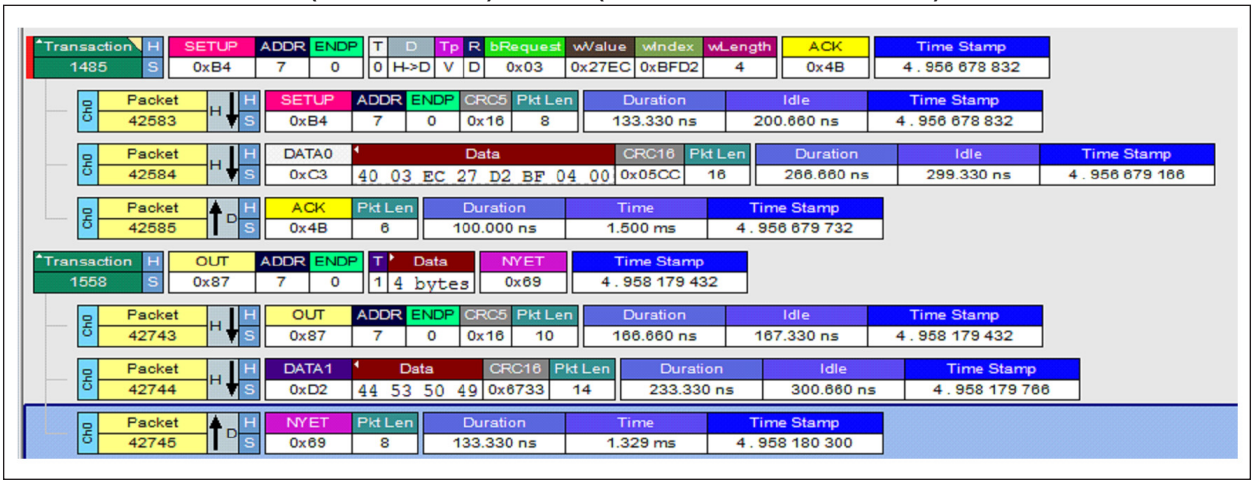
1. **STEP 1**: The disable SPI signature “DSPI” needs to be written to the XDATA memory location 0xBFD227EC. (See Figure 4.) The hub checks for this signature during the boot sequence. Refer to Table 6 for the USB SETUP command.

TABLE 6: USB SETUP COMMAND: WRITING DSPI SIGNATURE

SETUP Packet	Value	Description
bmRequestType	0x40	Vendor-specific command with host-to-device data transfer
bRequest	0x03	CMD_MEMORY_WRITE
wValue	0x27EC	ADDR_LO
wIndex	0xBFD2	ADDR_HI
wLength	0x0004	Number of data bytes to write

During the DATA phase of this transaction, the signature “DSPI” is sent to the hub. EP0 OUT Data = 0x44, 0x53, 0x50, 0x49 (‘D’, ‘S’, 'P', ‘I’).

FIGURE 4: XDATA (0XBFD227EC) WRITE (SETUP AND DATA PHASE)



2. **STEP 2:** Set bit 3 of XDATA memory address 0xBFD1DA1C. (Refer to Figure 5.) See Table 7 for the USB SETUP command information.

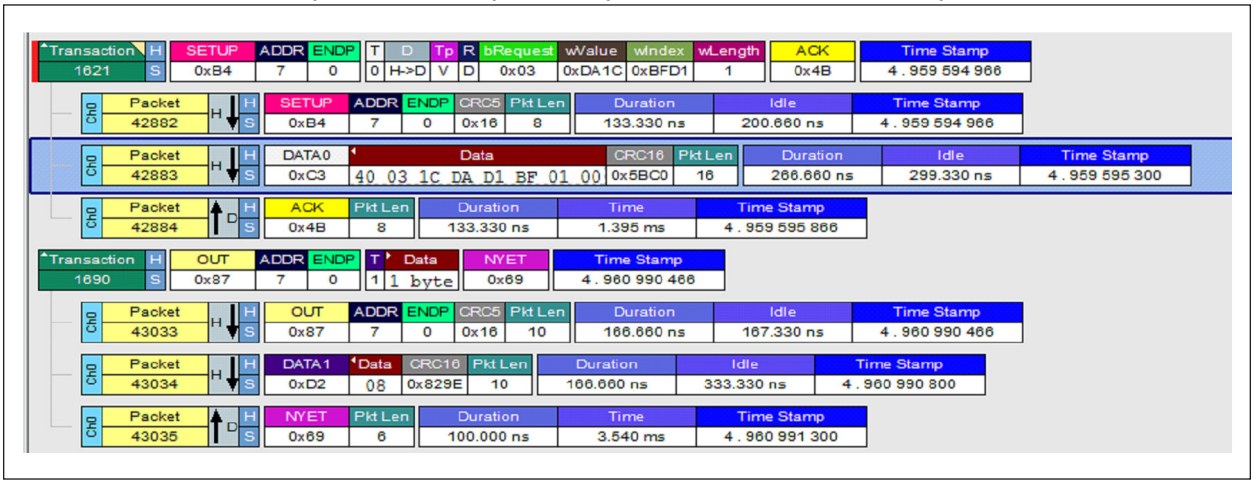
TABLE 7: USB SETUP COMMAND: SETTING BIT 3 AT 0XBFD1DA1C

SETUP Packet	Value	Description
bmRequestType	0x40	Vendor-specific command with host-to-device data transfer
bRequest	0x03	CMD_MEMORY_WRITE
wValue	0xDA1C	ADDR_LO
wIndex	0xBFD1	ADDR_HI
wLength	0x0001	Number of data bytes to write

During the DATA phase of this transaction, the signature 0x08 is sent to the hub. (See Figure 5.)

EP0 OUT Data = 0x08

FIGURE 5: XDATA (0XBFD1DA1C) WRITE (SETUP AND DATA PHASE)



3. **STEP 3:** Reset the hub as specified in Reset the Hub.

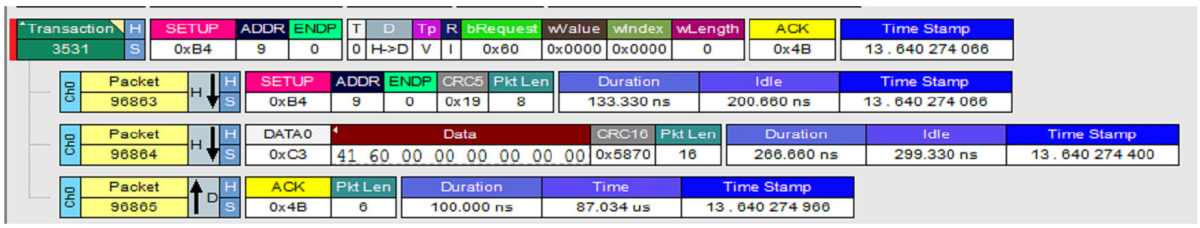
Enable SPI Pass-Through Interface

The following SETUP packet command is required to enable the SPI pass-through interface. (See Figure 6.) The interface must be enabled before any Write/Read commands could be issued by the host. Note that there is no DATA phase to this USB transaction. Refer to Table 8 for the USB SETUP command.

TABLE 8: USB SETUP COMMAND: ENABLING PASS-THROUGH INTERFACE

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x60	CMD_SPI_ENTER_PASSTHRU
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x0000	No DATA phase

FIGURE 6: ENABLE SPI PASS-THROUGH INTERFACE (SETUP PHASE)



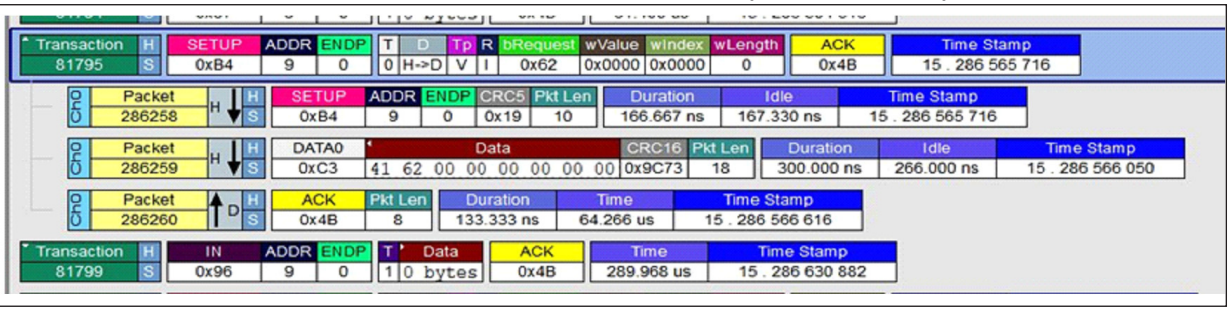
Disable Pass-Through Interface

The following SETUP Packet command is required to disable the SPI pass-through interface. (See Figure 7.) Note that there is no DATA phase to this USB transaction. Refer to Table 9 for the USB SETUP command.

TABLE 9: USB SETUP COMMAND: DISABLING SPI PASS-THROUGH INTERFACE

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x62	CMD_SPI_EXIT_PASSTHRU
wValue	0x0000	Reserved
wIndex	0x0000	Reserved
wLength	0x0000	No DATA phase

FIGURE 7: DISABLE SPI PASS-THROUGH INTERFACE (SETUP PHASE)



SPI Pass-Through Write/Read

The SPI Pass-through Write/Read command sequence consists of the following phases:

CMD_SPI_ACCESS PHASE

The host sends the following SETUP Packet command during the CMD_SPI_ACCESS phase. The non-zero wIndex in this case holds configurable fields which instruct the hub firmware to poll any Status bits (like Busy bits in case of an SPI flash) during an SPI Read/Write access. Refer to Table 10 for the USB SETUP command.

TABLE 10: USB SETUP COMMAND: CMD_SPI_ACCESS

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x61	CMD_SPI_PASS_THRU_WRITE
wValue	0xNNNN	Total length is the total number of data bytes to be sent to the SPI peripheral during the DATA phase that follows this SETUP packet. Total length = Signature length + Configurable field number + Command length + Response length (“SPID” Signature (4 bytes) + Configurable fields (4 bytes) + Number of bytes to transmit in the Command buffer + Number of bytes to receive in Response buffer)
wIndex	0xNN00	This value holds the number of configurable fields required for polling the Status bit. NN is the number of configurable fields. The configurable parameters are: 1. Status (Busy) bit location to Poll 2. Opcode for Status Read 3. Number of dummy bytes in the response 4. Index of the Response buffer to fetch the Status register value being polled Here NN is 04.
wLength	N	Number of bytes the SPI interface will transmit in the following data stage. wLength = Total length – Response length

DATA PHASE

During this phase, the host sends an OUT DATA packet to the SPI peripheral. This packet is formatted as follows:

EP0 DATA =

- Index 0: S (0x53)
- Index 1: P (0x50)
- Index 2: I (0x49)
- Index 3: D (0x44)
- Index 4: Busy bit (0x01)
- Index 5: Read Status opcode
- Index 6: Number of dummy bytes in response
- Index 7: Index of the Response buffer to fetch the Status register value being polled
- Index 8 to Index n: Command buffer data (command byte, address byte, dummy bytes, and up to 256 data bytes)

RETRIEVING THE DATA (APPLIES TO BLOCK READ ONLY)

These additional SETUP and DATA phases are applicable only in case of SPI Block Reads. During the previous DATA phase (See Data Phase.), based on the opcode, the hub retrieves the requested number of data bytes from the SPI peripheral and stores it internally in a register at location 0xBFD22310.

During this phase, the hub first sends the following SETUP packet to read the hub's internal register which contains the requested data. See Table 11 for the USB SETUP command.

TABLE 11: USB SETUP COMMAND: SPI BLOCK READ

SETUP Packet	Value	Description
bmRequestType	0xC0	Vendor-specific command with device-to-host to data transfer
bRequest	0x04	CMD_MEMORY_READ
wValue	0x2310	ADDR_LO
wIndex	0xBFD2	ADDR_HI
wLength	0xNNNN	Number of data bytes to be retrieved

This is followed by a DATA phase where the hub responds to the host with an IN packet containing the requested number of bytes retrieved from the SPI peripheral.

Single-Byte Instruction Write Example

Figure 8 shows an example where a single-byte instruction (WREN) is written to enable the Write latch in case of a SST26VF016B SPI flash memory. Refer to Table 12 for the USB SETUP command information.

TABLE 12: USB SETUP COMMAND: SPI SINGLE-BYTE WRITE

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x61	CMD_SPI_PASS_THRU_WRITE
wValue	0x0009	Total length = Signature length (4 bytes) + Configurable field number (4 bytes) + Command length (1 byte) + Response length (0 byte)
wIndex	0x0400	Number of configurable fields for polling status The configurable parameters are: 1. Busy bit location to Poll (0x01) 2. RDSR opcode (0x05) 3. Number of dummy bytes in the response (0x00) 4. Index of the Response buffer to fetch the Status register value that was returned (0x00)
wLength	0x0009	Number of bytes the SPI interface will transmit in the following data stage wLength = Total length – Response length wLength = 9 – 0 = 9 bytes

RDSR(0x05) is the Read Status register instruction and WREN (0x06) is the Write Latch Enable instruction for SST26VF016B.

During the DATA phase of this command sequence, the host sends an OUT packet containing 9 bytes to the SPI Flash as shown in Figure 8.

EP0 OUT Data =

Index 0: S (0x53)

Index 1: P (0x50)

Index 2: I (0x49)

Index 3: D (0x44)

Index 4: Busy Bit (0x01)

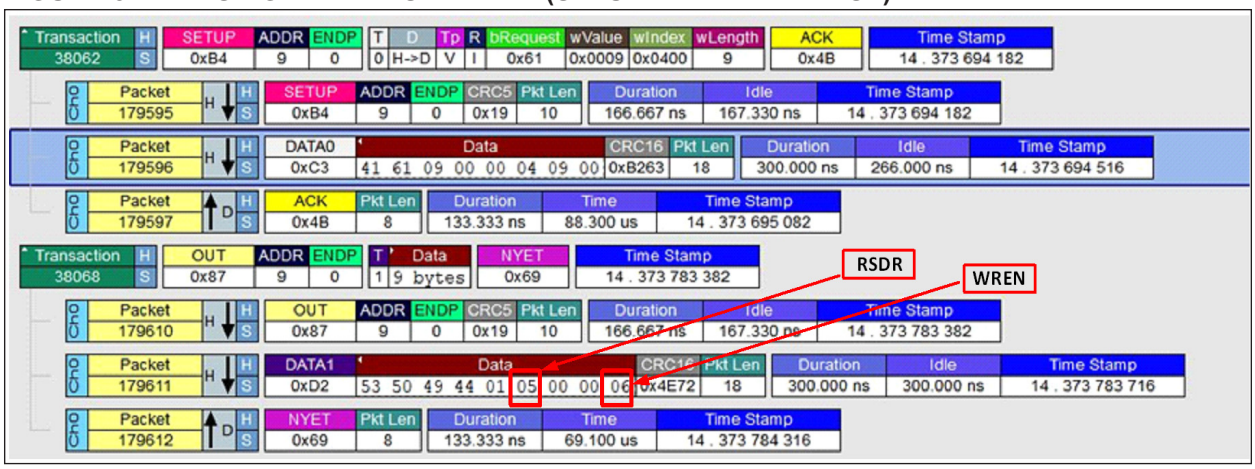
Index 5: Read Status opcode (RDSR:0x05)

Index 6: Number of dummy bytes in response (0x00)

Index 7: Index of the Response buffer to fetch the read status byte (0x00)

Index 8 to Index 9: Command buffer data {Command byte: WREN (0x06)}

FIGURE 8: SINGLE-BYTE SPI WRITE (SETUP AND DATA PHASE)



SPI Block Write Example

The SPI pass-through interface allows the host to perform a block write of up to 256 bytes. Figure 9 shows the SETUP packet for performing a block write to a SST26VF016B SPI flash memory. See Table 13 for the USB SETUP command details.

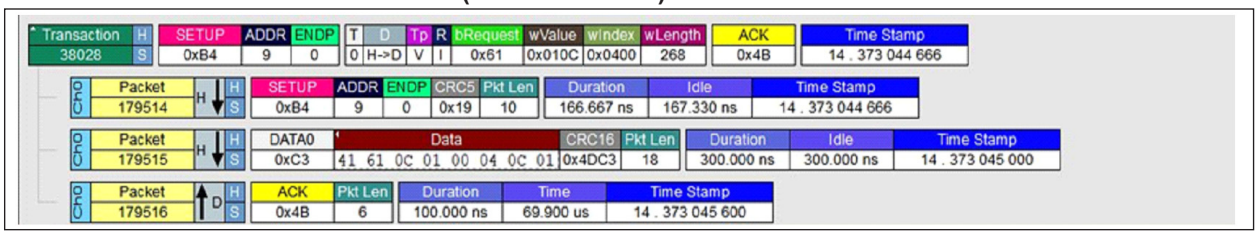
TABLE 13: USB SETUP COMMAND: SPI BLOCK WRITE

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x61	CMD_SPI_PASS_THRU_WRITE
wValue	0x010C	Total length = Signature length (4 bytes) + Configurable field number (4 bytes) + Command length (1-byte opcode + 3 bytes of address + 256 data bytes) + Response length (0 byte)
wIndex	0x0400	Number of configurable fields for polling status The configurable parameters are: 1. Busy bit location to Poll (0x01) 2. RDSR opcode (0x05) 3. Number of dummy bytes in the response (0x00) 4. Index of the Response buffer to fetch the Status register value that was returned (0x00)

TABLE 13: USB SETUP COMMAND: SPI BLOCK WRITE (CONTINUED)

SETUP Packet	Value	Description
wLength	0x010C	Number of bytes the SPI interface will transmit in the following data stage wLength = Total length – Response length wLength = 268 – 0 = 268 bytes (0x010C)

FIGURE 9: SPI BLOCK WRITE (SETUP PHASE)



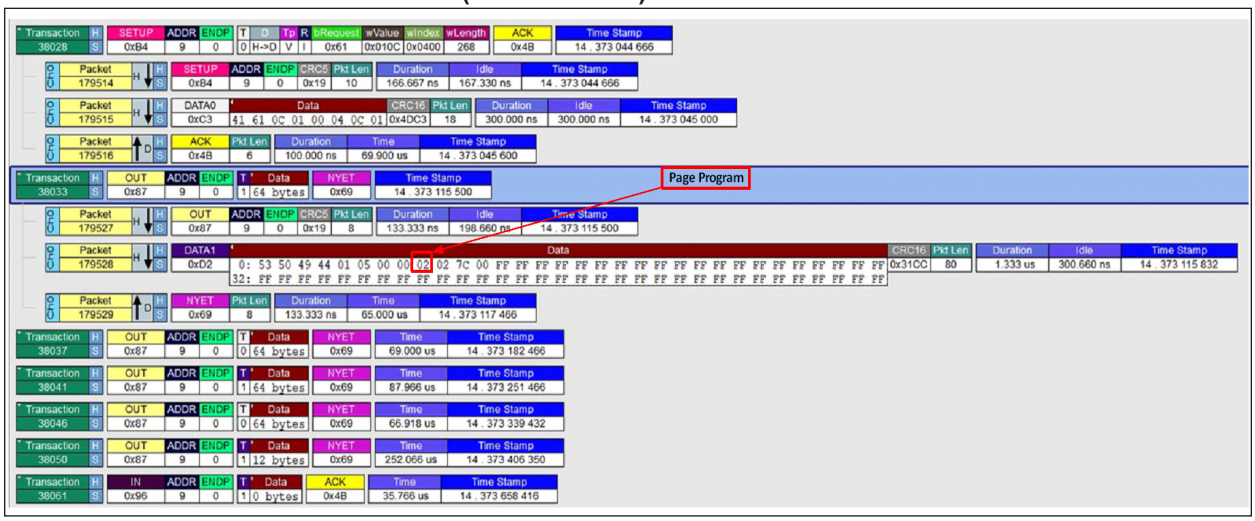
During the DATA phase (See Figure 10.) of this command sequence, 268 bytes are transferred to the SPI flash via the hubs pass-through interface. The DATA Packet consists of the following:

EP0 OUT Data =

- Index 0: S (0x53)
- Index 1: P (0x50)
- Index 2: I (0x49)
- Index 3: D (0x44)
- Index 4: Busy bit (0x01)
- Index 5: Read Status opcode (RDSR:0x05)
- Index 6: Number of dummy bytes in response (0x00)
- Index 7: Index of the Response buffer to fetch the read status byte (0x00)
- Index 8 to Index 267: Command buffer data {Command byte: PAGE PROGRAM (0x02), address byte (0x02, 0x7C, 0x00), dummy byte (none), and up to 256 data bytes}

In this case, PAGE PROGRAM (0x02) instruction is specific to SST26VF016B.

FIGURE 10: SPI BLOCK WRITE (DATA PHASE)



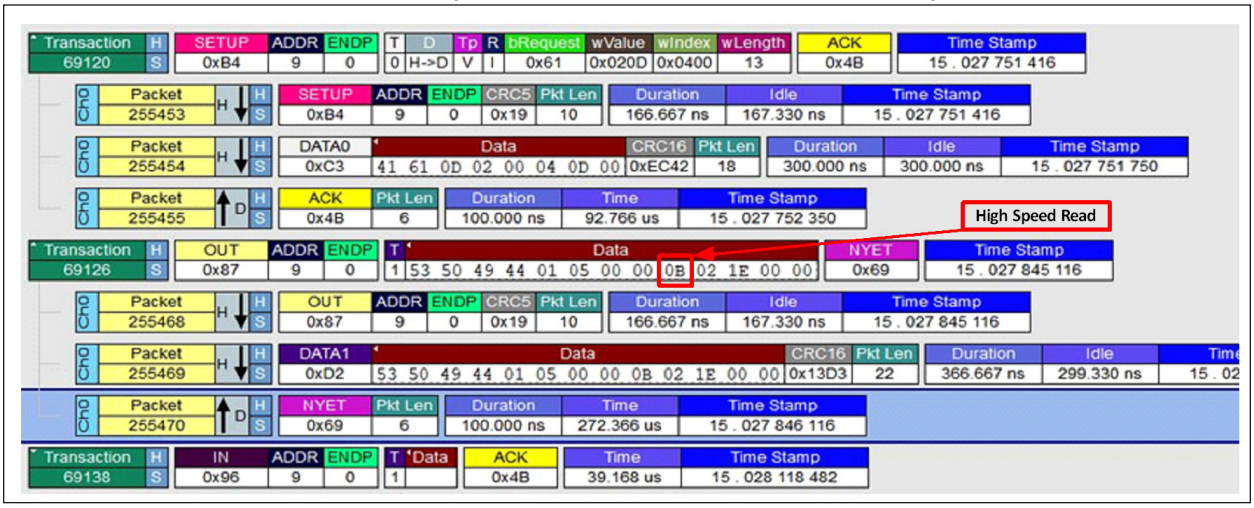
SPI Block Read Example

The SPI pass-through interface allows the host to perform a block read of up to 512 bytes. Figure 11 shows the SETUP packet for this transaction in case of a SST26VF016B SPI flash memory. See Table 14 for the USB SETUP command information.

TABLE 14: USB SETUP COMMAND: SPI BLOCK READ

SETUP Packet	Value	Description
bmRequestType	0x41	Host-to-device data transfer with a vendor-specific command that targets the device
bRequest	0x61	CMD_SPI_PASS_THRU_WRITE
wValue	0x020D	Total length = Signature length (4 bytes) + Configurable field number (4 bytes) + Command length (1-byte opcode + 3 bytes of address +1 dummy byte) + Response length (512 data bytes)
wIndex	0x0400	Number of configurable fields for polling status The configurable parameters are: 1. Busy bit location to Poll (0x01) 2. RDSR opcode (0x05) 3. Number of dummy bytes in the response (0x00) 4. Index of the Response buffer to fetch the Status register value that was returned (0x00)
wLength	0x000D	Number of bytes the SPI interface will transmit in the following data stage wLength = Total length – Response length wLength = 525 – 512 = 13 bytes (0x000D)

FIGURE 11: SPI BLOCK READ (SETUP PHASE 1 AND DATA PHASE 1)



During the DATA phase (See Figure 11.) that follows, 13 bytes are transferred to the hub. The DATA packet consists of the following:

EP0 OUT Data =

- Index 0: S (0x53)
- Index 1: P (0x50)
- Index 2: I (0x49)
- Index 3: D (0x44)
- Index 4: Busy bit (0x01)
- Index 5: Read Status opcode (RDSR:0x05)
- Index 6: Number of dummy bytes in response (0x00)

- Index 7: Index of the Response buffer to fetch the read status byte (0x00)
- Index 8 to Index 12: Command buffer data {Command byte: HIGH SPEED READ (0x0B), address byte (0x02, 0x1E, 0x00), dummy byte (0x00)}

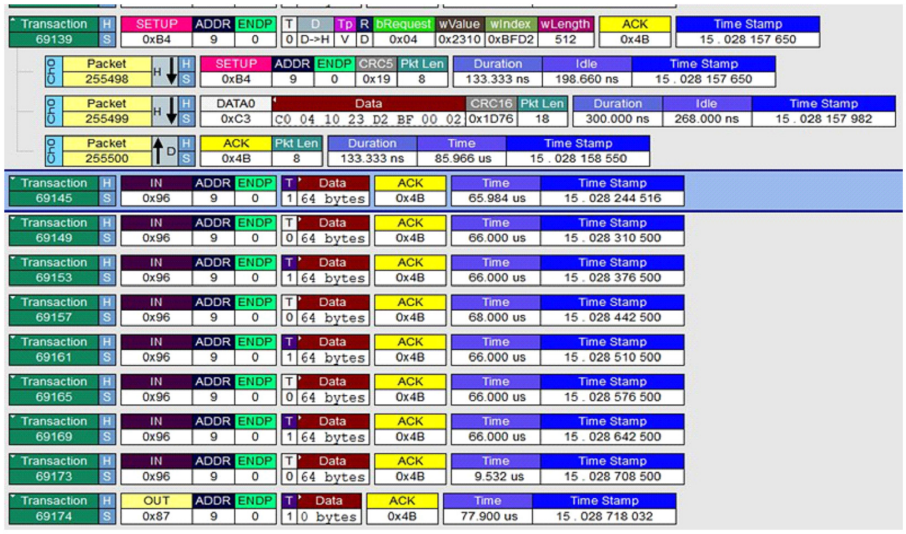
In this case, HIGH-SPEED READ (0x0B) instruction is specific to SST26VF016B.

The host then initiates the second SETUP phase by sending a SETUP packet for reading the hub's internal register at 0xBFD22310 containing the data bytes to be read. Figure 12 shows this SETUP packet. Refer to Table 15 for the USB SETUP command details.

TABLE 15: USB SETUP COMMAND: SPI BLOCK READ

SETUP Packet	Value	Description
bmRequestType	0xC0	Vendor-specific command with device-to-host-to-data transfer
bRequest	0x04	CMD_MEMORY_READ
wValue	0x2310	ADDR_LO
wIndex	0xBFD2	ADDR_HI
wLength	0x0200	Number of data bytes to be retrieved (512 bytes)

FIGURE 12: SPI BLOCK READ (SETUP PHASE 2 AND DATA PHASE 2)



During the DATA phase that follows this, the Hub responds with IN packets (See Figure 12.) containing a total of 512 bytes retrieved from the SPI flash via the pass-through interface.

APPENDIX A: APPLICATION NOTE REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS-00002430C (09-26-18)	Introduction	Added USB4912, USB4712, and USB4714 hubs
	References	Added USB4912, USB4712, and USB4714 Data Sheets to the list of references and changed *"AN2651: Configuration of the USB49x4 Application Note" to "Configuration of the USB47xx and USB49xx Application Note"*
	Microchip Software Solutions	Replaced the "3.0 MPLABCC DLL Implementation" section with the Microchip Software Solutions section
	Table 1	Added settings for USB4712 and USB4912
	Table 2	Added SPI interface pin numbers for USB4712 and USB4912
	All	Made formatting changes throughout the document
DS-00002430B (05-30-18)	USB-to-SPI Bridging with Microchip USB471x and USB49xx Hubs	Changed "USB4715" to "USB471x"
	Introduction	Specified new functions
	References	Added *"SST26VF016B Data Sheet"* and *"AN2651: Configuration of the USB49x4 Application Note"*
	Figure 1	Changed "USB4904 Hub" to "USB49xx/471x Hub"
	Table 2	Changed "USB3613 AND USB3813" to "USB49xx AND USB471x" in the title
	Low-Level Implementation	Changed "Manual Implementation" to "Low-level Implementation" and merged Sections 5 and 6 with this section
	All	Made minor text and formatting changes throughout the document
DS00002430A (04-25-17)	All	Initial release

THE MICROCHIP WEBSITE

Microchip provides Online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 website 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 website 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: 978-1-5224-3560-0

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

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

China - Beijing
Tel: 86-10-8569-7000

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

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

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

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

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

China - Hong Kong SAR
Tel: 852-2943-5100

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

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

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

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

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

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

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

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

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

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

India - Pune
Tel: 91-20-4121-0141

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

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

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

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

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

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

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

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

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

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

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

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

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: 47-7288-4388

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