AN3240

USB-to-I^{2}C Bridging with Microchip USB720x and USB725x Hubs

Authors: Mick Davis and Shiva Balasubramanian
Microchip Technology Inc.

INTRODUCTION

The USB-to-I^{2}C Bridging feature gives system designers using Microchip hubs expanded system control and potential BOM reduction. The use of a separate USB-to-I^{2}C device is no longer required and a downstream USB port is not lost as occurs when a standalone USB-to-I^{2}C device is implemented. This feature is available on the Microchip USB7202, USB7206, USB7250, USB7251, USB7252, and USB7256 hubs.

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:

- Configure I^{2}C Pass-Through Interface
- I^{2}C Write
- I^{2}C Read

SECTIONS

- General Information
- Part Number-Specific Information
- Microchip Software Solutions
- Low-Level Implementation
- Examples

REFERENCES

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

- *Microchip USB7202 Data Sheet*
- *Microchip USB7206 Data Sheet*
- *Microchip USB7250 Data Sheet*
- *Microchip USB7251 Data Sheet*
- *Microchip USB7252 Data Sheet*
- *Microchip USB7256 Data Sheet*
- *Microchip AN2935 Configuration of USB7202/USB7206/USB725x Application Note*

GENERAL INFORMATION

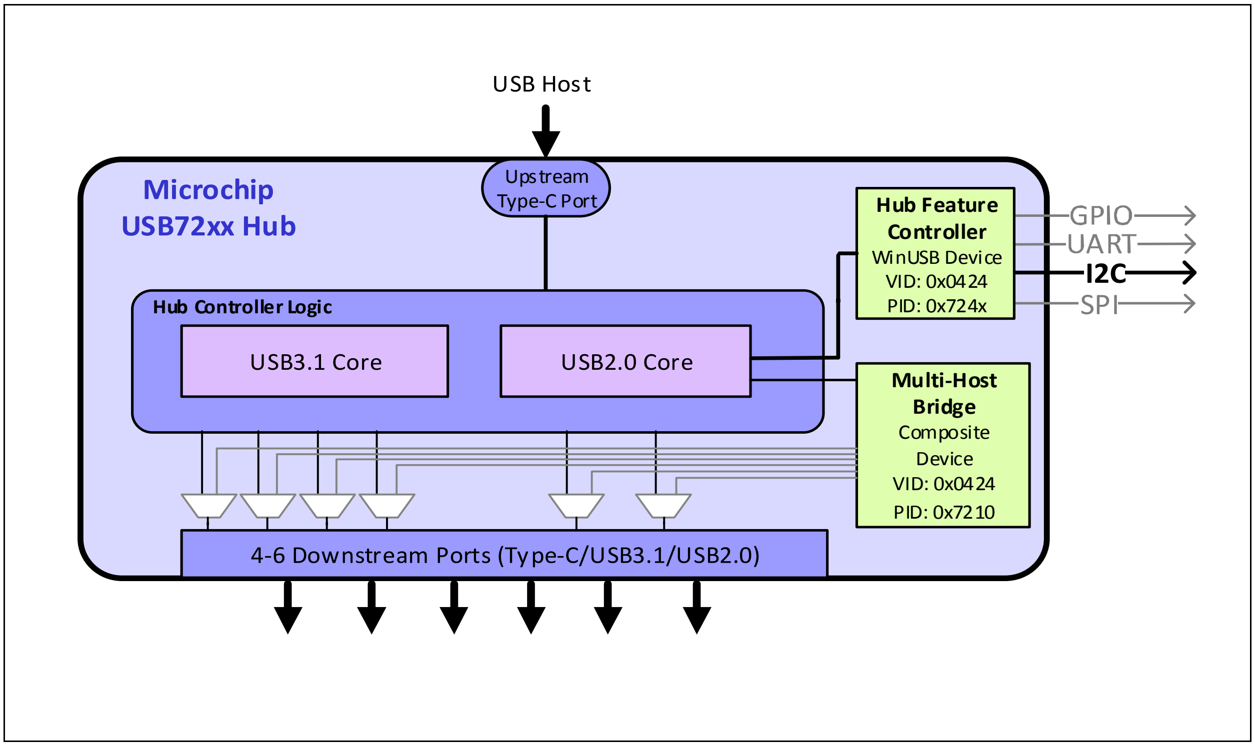
Microchip hub USB Bridging features work via host commands sent to a Hub Feature Controller embedded within the hub located on an additional internal USB port. In order for the bridging features to work correctly, this internal Hub Feature Controller must be enabled by default. Table 1 provides details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

TABLE 1: DEFAULT SETTINGS FOR THE HUB FEATURE CONTROLLER ENABLE

Part Number	Part Summary	Hub Feature Controller Default Setting	USB3 PID	USB2 PID
USB7202	4-Port USB3.1 Gen2 Hub	Enabled on Port 6	0x7202	0x4202
USB7206	6-Port USB3.1 Gen2 Hub	Enabled on Port 8	0x7206	0x4206
USB7250	4-Port USB3.1 Gen2 Hub with power delivery on 3 ports	Enabled on Port 6	0x7250	0x4250
USB7251	4-Port USB3.1 Gen2 Hub with power delivery on 2 ports	Enabled on Port 6	0x7251	0x4251
USB7252	4-Port USB3.1 Gen2 Hub with power delivery on 1 port	Enabled on Port 6	0x7252	0x4252
USB7256	6-Port USB3.1 Gen2 Hub with power delivery on 1 port	Enabled on Port 8	0x7256	0x4256

The Hub Feature Controller is a USB 2.0 WinUSB class device connected to an internal USB 2.0 port in the hub. For example, in a four-port hub, the Hub Controller is connected to port 6 of the USB 2.0 portion of the hub. The base Product ID (PID) for the Hub Feature Controller is 0x7240. All bridging commands are addressed to the Hub Feature Controller and not the Hub. (See Figure 1.)

FIGURE 1: MICROCHIP HUB CONTROLLER EXAMPLE



I²C Bridging Commands

The following I²C functions are supported:

- I2C Write
- I2C Read

I²C WRITE

The I²C interface works as a complete pass-through. This means that the Host must properly arrange data payloads in the appropriate I²C-compatible format and bit order, including the I²C slave device address. Up to 128 bytes of data payload may be sent per I²C Write command sequence.

I²C READ

The I²C interface works as a complete pass-through. This means that the Host must properly arrange data payloads in the appropriate I²C-compatible format and bit order, including the I²C slave device address. Up to 128 bytes of data payload may be received from the slave device per I²C Read command sequence.

I²C Interface Setup Requirements

The I²C interface operates at 100 kHz clock speed by default. Refer to Clock Configuration for other supported speeds. The I²C interface is supported in all configuration options.

PART NUMBER-SPECIFIC INFORMATION

Part Summary

Table 2 displays the I^2C interface pins by part number.

TABLE 2: USB72XX I^2C PINS

Part Number	I^2C Pin Function	Option 1 (I^2C)	Option 2 (I^2S™)	Option 3 (UART) (Note 1)	Option 4 (FLEX)	Option 5
USB7202	MSTR_I2C_CLK	PF18	PF30	—	—	—
	MSTR_I2C_DATA	PF19	PF31	—	—	—
USB7206	MSTR_I2C_CLK	—	—	PF18	—	—
	MSTR_I2C_DATA	—	—	PF31	—	—
USB7250	MSTR_I2C_CLK	PF11	PF11	PF11	PF11	PF11
	MSTR_I2C_DATA	PF10	PF10	PF10	PF10	PF10
USB7251	MSTR_I2C_CLK	PF11	PF11	PF11	PF11	—
	MSTR_I2C_DATA	PF10	PF10	PF10	PF10	—
USB7252	MSTR_I2C_CLK	PF30	PF30	PF30	PF30	—
	MSTR_I2C_DATA	PF31	PF31	PF31	PF31	—
USB7256	MSTR_I2C_CLK	PF30	PF30	—	—	—
	MSTR_I2C_DATA	PF31	PF31	—	—	—

Note 1: In Option 3, USB7206 is configured for I^2S™ without power delivery.

MICROCHIP SOFTWARE SOLUTIONS

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-I^{2}C Bridging in a standalone form. In addition, it contains a Dynamically Linked Library (DLL) for Windows® which can be used for implementing USB-to-I^{2}C 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-I^{2}C Bridging on Linux®, you can use one of the following USB72xx Linux Application Code Examples (ACEs):

- ***ACE006 USB-to-I^{2}C Bridging*: This ACE demonstrates how to read and write data through the I^{2}C Master interface in USB720x/USB72xx using USB-to-I^{2}C 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 solutions for the desired operating system.

LOW-LEVEL IMPLEMENTATION

The USB-to-I^2C 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 the I^2C pass-through control packets are shown below. All USB to I^2C Bridging commands must be sent directly to Endpoint 0 of the Hub Feature Controller connected to the last downstream port of the Microchip hub (i.e. located on the Port 5 of a four-port hub).

I^2C Enter Pass-Through Command

The I^2C Enter Pass-Through command is required to enable the I^2C bridge. This command only needs to be issued one time for every reset or power cycle. The I^2C clock frequency is also configured within the wValue of this command. (See Clock Configuration for details on the possible values.)

I^2C Control Flags

Both the Read and Write commands have a special control flag parameter defined in Table 3.

TABLE 3: I^2C CONTROL FLAGS

Bits	Control	Usage
2 to 7	Reserved	N/A
2	SEND_NACK	If asserted, NACK the last byte in the transfer.
1	SEND_START	If asserted, send a Start condition as the first step in the I^2C command.
0	SEND_STOP	If asserted, send a Stop condition as the last step of this command.

I^2C Write Command

This command is used to send data to an I^2C peripheral connected to the USB hub. Both the I^2C Control flags (defined in I2C Control Flags) and the I^2C slave address are bundled into the wValue field. See Table 4 for more details on the command.

TABLE 4: USB SETUP COMMAND

SETUP Parameter	Value	Description
bmRequestType	`0x41`	Vendor-specific command; Host-to-device data transfer
bRequest	`0x71`	Register read command: CMD_I2C_WRITE
wValue	`0xXXYY`	MSB (XX): I^2C Control flags (See I2C Control Flags.) LSB (YY): I^2C Slave device address
wIndex	`0x0000`	Reserved
wLength	`0xNN`	N bytes of data to be sent in the data stage (in the OUT EP0 control transfer packets)

I^2C WRITE USB TRANSACTION SEQUENCE

Command Phase: The Hub Feature Controller receives the setup packet with the parameters specified above.

Data Phase: The Host sends multiple EP0 OUT packets of 64 bytes each with a total length of N bytes.

Status Phase: If an IN-Zero Length packet is sent from Hub Feature Controller, it would mean that the transfer was a success. If an IN-STALL packet is sent from Hub Feature Controller, there was an error during the transfer potentially due to missing ACK from the I^2C slave.

I^2C Read Command

This command is used to read data from an I^2C peripheral connected to the USB hub. Both the I^2C Control flags (defined in I2C Control Flags) and the I^2C slave address are bundled into the wValue field. (See Table 5.)

TABLE 5: USB SETUP COMMAND

SETUP Parameter	Value	Description
bmRequestType	0xC1	Vendor-specific command; Device-to-Host data transfer
bRequest	0x72	Register read command: CMD_I2C_READ
wValue	0xXXYY	MSB (XX): I^2C Control flags (See I2C Control Flags.) LSB (YY): I^2C Slave device address
wIndex	0x0000	Reserved
wLength	0xNN	N bytes of data to be sent in the data stage (in the OUT EP0 control transfer packets)

I^2C READ USB TRANSACTION SEQUENCE

Command Phase: The Hub Feature Controller receives the SETUP packet with the parameters specified above.

Data Phase: The Hub Feature Controller sends Multiple EP0 IN packets of 64 bytes, each with a total length of N bytes.

Status Phase: The Host sends an OUT-Zero Length ACK packet to acknowledge receipt of data.

EXAMPLES

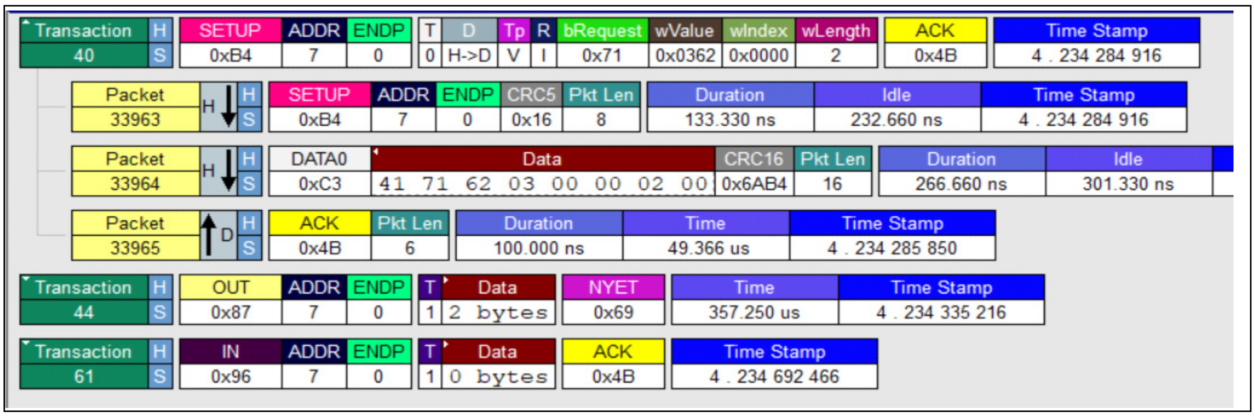
Send an I²C Write to an Attached Device

1. **Command Phase (SETUP Transaction):** I²C Address 0x62: Write a value of 0x12 to Register 0x15. Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to send an I²C Write command to the attached I²C device with the I²C Address as defined in the wValue field. (See Table 6 and Figure 2.)

TABLE 6: I²C WRITE SETUP PACKET EXAMPLE

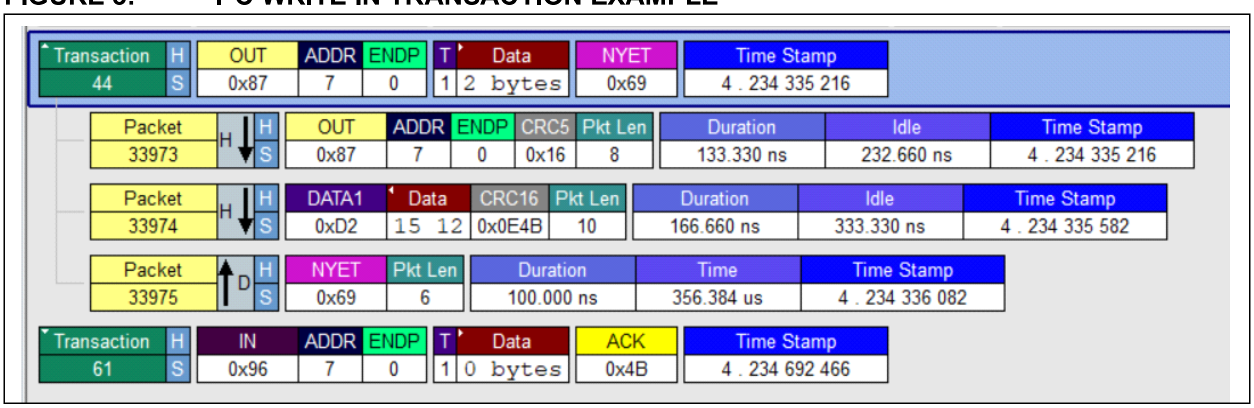
Field	Value	Note
bmRequestType	`0x41`	—
bRequest	`0x71`	—
wValue	`0x0362`	I²C Control flag `0x03`, I²C Address `0x62` (0110 0010b)
wIndex	`0x0000`	—
wLength	`0x0002`	2 bytes of data (Register Address + 1 byte of data)

FIGURE 2: I²C WRITE SETUP TRANSACTION EXAMPLE



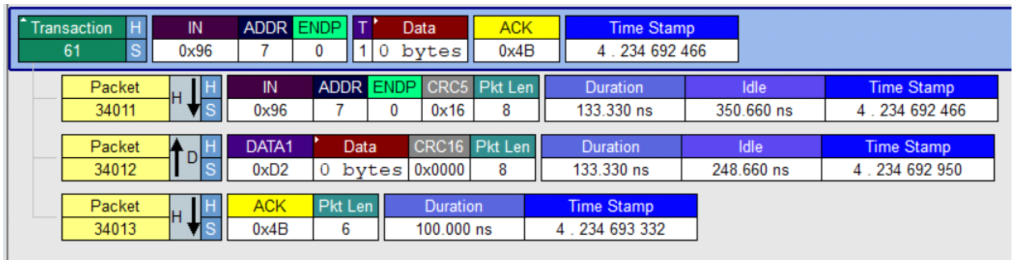
2. **Data Phase (OUT Transaction):** Host sends an OUT packet followed by the data bytes of length wLength starting from the specified address after receiving an IN packet. In this instance, Register `0x12` is being written to Register `0x15` (Data = `0x15`, `0x12`). Hub Feature Controller responds with a NYET after receiving the data. (See Figure 3.)

FIGURE 3: I²C WRITE IN TRANSACTION EXAMPLE



3. **Status Phase (IN Transaction):** 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. (See Figure 4.)

FIGURE 4: I²C WRITE OUT TRANSACTION EXAMPLE



Send an I²C Read to an Attached Device

A read requires two operations:

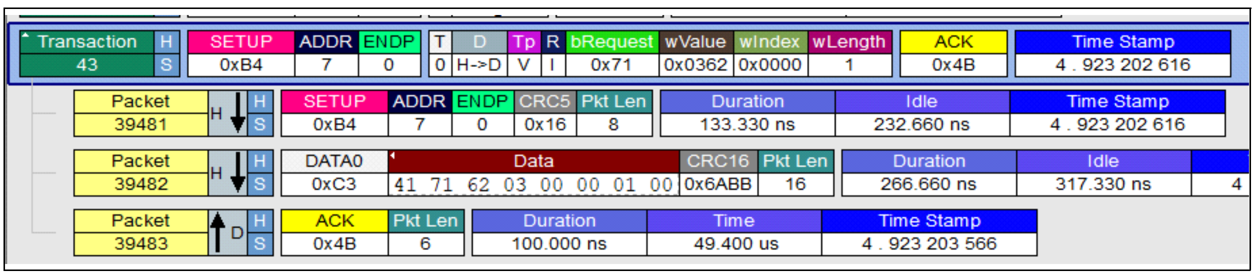
- Transaction 1: Write the address of the register to be read using the I²C Write example above.
- Transaction 2: Read the register content(s), depending on length.

1. **Command Phase 1 (SETUP Transaction 1):** I²C Address 0x62: Read Register 0x15. Send the following SETUP Register Read Command in to Endpoint 0 of the Hub Feature Controller to prepare the I²C device to return data. (See Table 7 and Figure 5.)

TABLE 7: I²C READ SETUP COMMAND 1 EXAMPLE

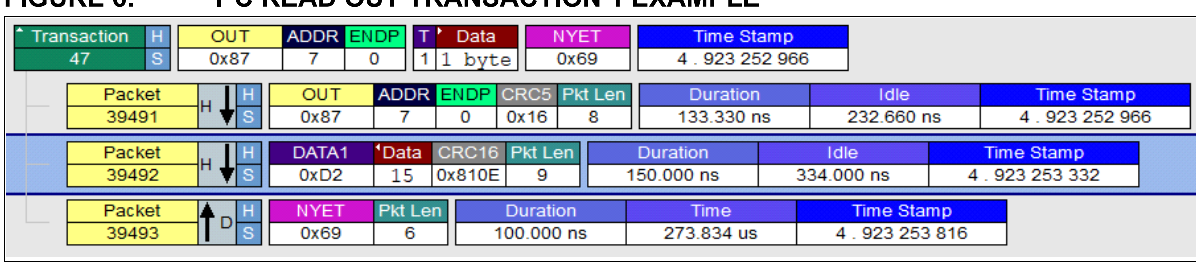
SETUP Parameter	Value	Note
bmRequestType	`0x41`	—
bRequest	`0x71`	—
wValue	`0x0362`	Control flag = `0x03`, I²C Address = `0x62` (01100010b)
wIndex	`0x0000`	—
wLength	`0x0001`	—

FIGURE 5: I²C READ SETUP TRANSACTION 1 EXAMPLE



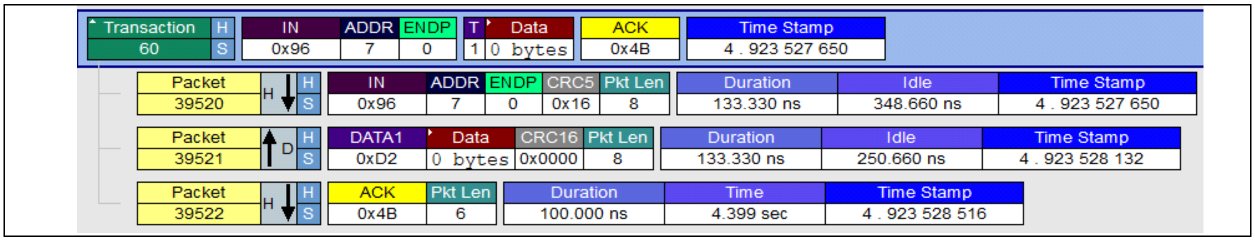
2. **Data Phase 1 (OUT Transaction 1):** Host sends an OUT packet followed by the data. The data in this instance is the register address `0x15`. Hub Feature Controller responds withe a NYET. (See Figure 6.)

FIGURE 6: I²C READ OUT TRANSACTION 1 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. Host sends an ACK. (See Figure 7.)

FIGURE 7: I²C READ IN TRANSACTION 1 EXAMPLE

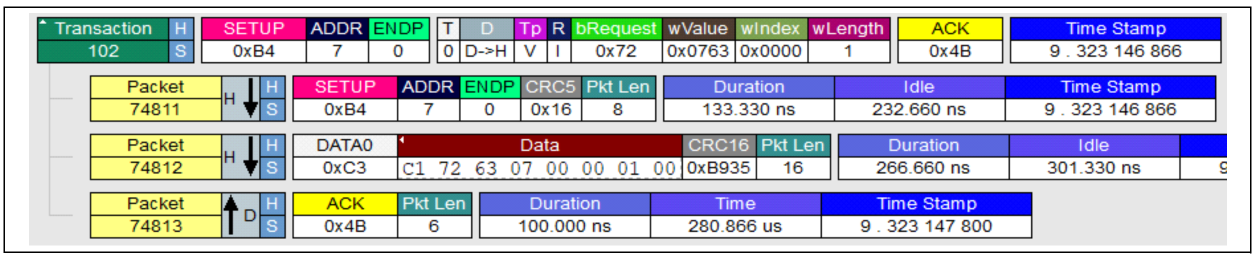


4. **Command Phase 2 (SETUP Transaction 2):** Send the following SETUP Register Read Command to Endpoint 0 of the Hub Feature Controller to retrieve the requested data. (See Table 8 and Figure 8.)

TABLE 8: I²C READ SETUP COMMAND 2 EXAMPLE

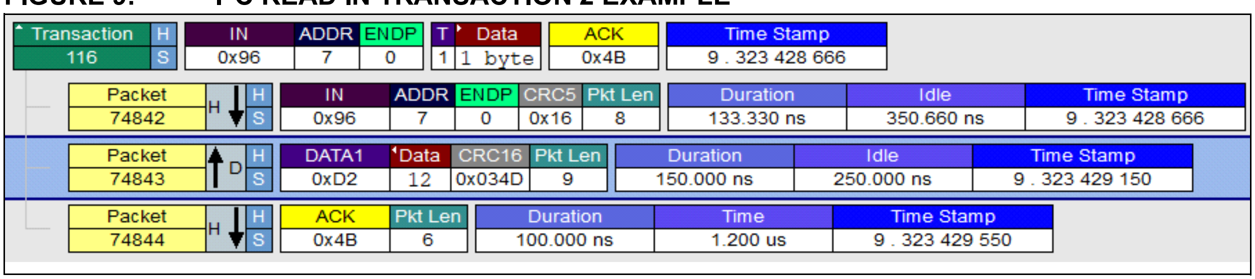
Setup Parameter	Value	Note
bmRequestType	`0xC1`	—
bRequest	`0x72`	—
wValue	`0x0763`	Control Flag = `0x07`, I²C Address = `0x63` (01100011b)
wIndex	`0x0000`	—
wLength	`0x0001`	—

FIGURE 8: I²C ADDRESS DATA PHASE BYTE 3 TRANSACTION 2 EXAMPLE



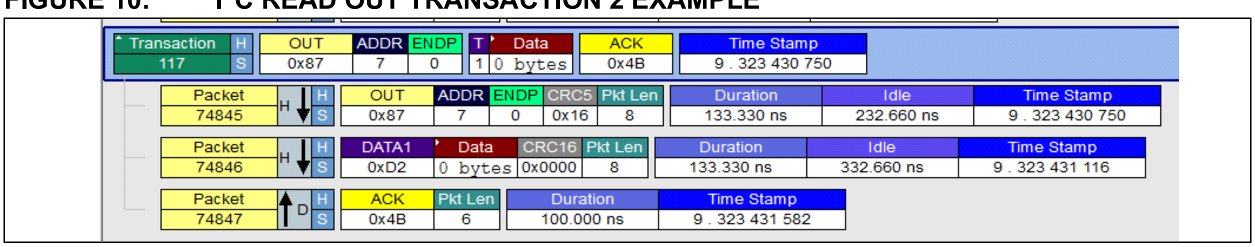
5. **Data Phase 2 (IN Transaction 2):** Host sends an IN packet, and Hub Feature Controller responds with the register contents (`0x12`). Host responds with an ACK. (See Figure 9.)

FIGURE 9: I²C READ IN TRANSACTION 2 EXAMPLE



6. **Status Phase 2 (OUT Transaction 2):** Host sends an OUT packet followed by a zero-data length packet. Hub Feature Controller responds with an ACK to complete the bridging command. (See Figure 10.)

FIGURE 10: I²C READ OUT TRANSACTION 2 EXAMPLE



CLOCK CONFIGURATION

There is a register to control I^2C clock frequency, named bI2CInter128Delay located at address 0xBFD23410. If the DLL API is used, register bI2CInter128Delay is written automatically. The value of bI2CInter128Delay is determined using the formula in Equation 1.

EQUATION 1:

bI2CInter128Delay = 2 x Time period of the I^2C bus clock in microseconds

The default value is 0x14 for 100 kHz clock. A value of 0x5A creates a delay of 900 µs. This value is multiplied by 10 in the firmware to have some buffer time in order not to miss any byte when operating at a lower speed, thereby ensuring data integrity.

The maximum value that can be programmed in bI2CInter128Delay is 0x63. (i.e. a maximum of 99 x 10 = 990 µs can be added as the maximum Inter-128Byte delay)

To configure the USB-I^2C bridge for 40 kHz clock operation, it is only necessary to write a value of 0x32 to bI2CInter128Delay after any other I^2C bridge setup has been made. The bI2CInter128Delay and Bus Frequency Control register values are provided for various supported clock frequencies in Table 9.

The method for writing to registers (including bI2CInter128Delay) through the SMBus (slave) is explained in Section 4.0 of ***AN2935 Configuration of USB7202/USB7206/USB725x Application Note***. An example clock configuration is provided in the Clock Configuration Example.

TABLE 9: BUS FREQUENCY CONTROL AND B12CINTER128DELAY REGISTER VALUES FOR COMMON 12C CLOCK FREQUENCIES

Frequency (kHz)	Bus Frequency Register Value (hex)	bI2CInter128Delay Value	
		Decimal	Hexadecimal
400	0A00	5	05
250	081B	8	08
200	1818	10	0A
100 (default)	3131	20	14
80	3D3E	25	19
50	6363	40	28
40	7C7C	50	32
25	C7C7	80	50
20	F9F9	100	64

Clock Configuration Example

A clock configuration example for 40 kHz operation is provided below. (Refer to Table 10 to Table 13.)

1. Write bI2CInter128Delay located at 0xBFD23410 with a value of 0x32 (40 kHz per Table 9).

TABLE 10: CLOCK CONFIGURATION COMMAND 1 EXAMPLE

Setup Parameter	Value	Note
bmRequestType	`0x40`	Host-to-device data transfer
bRequest	`0x03`	CMD_MEMORY_WRITE
wValue	`0x3410`	Least Significant 16-bits of memory address in little-endian format
wIndex	`0xBFD2`	Most Significant 16-bits of memory address in little-endian format
wLength	`0x0001`	Number of data bytes to write

Note 1: Data to be written: 0x32

2. Enable I^2C pass-through and set frequency.

TABLE 11: CLOCK CONFIGURATION COMMAND 2 EXAMPLE

Setup Parameter	Value	Note
bmRequestType	0x41	Host-to-device data transfer
bRequest	0x70	CMD_I2C_ENTER_PASSTHRU
wValue	0x7C7C	I^2C Clock Frequency: 40 kHz
wIndex	0x0000	—
wLength	0x0000	—

3. Write the start address from which data needs to be read.

TABLE 12: CLOCK CONFIGURATION COMMAND 3 EXAMPLE

Setup Parameter	Value	Note
bmRequestType	0x41	Host-to-device data transfer
bRequest	0x71	CMD_I2C_WRITE
wValue	0x03A0	03: I^2C Control flags (START, STOP) A0: Slave Address
wIndex	0x0000	—
wLength	0x0001	1 byte of data

Note 1: Data to be written: 0x00

4. Read 2 bytes of data.

TABLE 13: CLOCK CONFIGURATION COMMAND 4 EXAMPLE

Setup Parameter	Value	Note
bmRequestType	0xC1	Host-to-device data transfer
bRequest	0x72	CMD_I2C_READ
wValue	0x07A1	07: I^2C Control flags (NACK, START, STOP) A1: Slave Address
wIndex	0x0000	—
wLength	0x0002	2 bytes of data

APPENDIX A: APPLICATION NOTE REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/ Entry	Correction
DS00003240A (10-14-19)	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 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 website at: http://microchip.com/support

ISBN: 978-1-5224-5165-5

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

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: 49-7131-72400

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