



ANX74xx Programming Guide

USB Type-C™ Switches

AA-005329-PG-4D

13 Dec 2018

Analogix Semiconductor, Inc.

3211 Scott Blvd., Suite 100
Santa Clara, CA 95054, USA
+1 (408) 988-8848
www.analogix.com

Contents

1	Introduction.....	7
2	I2C Bus Interface	9
2.1	I2C Slave.....	9
2.2	I2C Master	9
3	Flash Programming	12
3.1	Flash Blocks for Different Versions of Firmware	12
3.2	Firmware Image Area Updating through I2C.....	13
3.3	Firmware Integrity Checking.....	16
3.3.1	Firmware CRC	16
3.3.2	Generating the Internal MCU Firmware Image with CRC.....	16
3.4	Customer-defined Zone	16
3.5	Erasing Flash Memory.....	17
4	ANX74xx Driver on AP/EC Side	19
4.1	ANX74xx Driver Overview	19
4.2	AP/EC Driver Structure	19
4.3	Initialization Settings	19
4.3.1	Firmware Control.....	19
4.3.2	Auto USB PD Negotiation Registers.....	21
4.4	Interrupt Registers.....	22
4.5	VBUS Control	24
4.6	CC/VCONN/Data Role Status	25
4.7	USB PD Interface.....	25
4.7.1	Interface Registers Introduction.....	25
4.7.2	Interface Message Construction.....	25
4.7.3	Interface Message Communication.....	26
4.8	Customer VDM Mode.....	29
4.9	PPS Function on Sink.....	30
5	Automatic Power Negotiation	32
5.1	Accepting/Rejecting the Request Data Object	32
5.2	Responding to the Source Capabilities	32
6	UCSI Passthrough Mode Implementation	36
6.1	Overview.....	36
6.2	Sending a Command	36
6.3	Sending an Interrupt.....	36
6.4	Command List	36
6.5	PPM Initialization.....	37
7	Code Examples.....	39
7.1	Power Status/Cable Status/Messages Received Interrupt Handling.....	39
8	Appendix A: PD Messaging Interface Code	41
8.1	Message Transaction Method	41

8.1.1	Communication between AP/EC and ANX74xx OCM Firmware	41
9	Appendix B: Specific Applications from Customer	43
9.1	Not Handle Interrupt Source	43
9.1.1	Application Overview	43
9.1.2	Work Flow (Recommended)	43
10	Appendix C: Firmware Update (PDFU)	44
10.1	Power Delivery Firmware Update	44
10.1.1	Port Controller Application	44
10.1.2	Cable Application	45
11	Revision History	48

ANALOGIX CONFIDENTIAL

List of Figures

Figure 1-1 ANX74xx Hardware Block Diagram (with I2C Interface)	7
Figure 1-2 Typical ANX74xx Application	8
Figure 3-1 ANX74xx Flash Space Allocation.....	12
Figure 3-2 Firmware Image Updating.....	13
Figure 3-3 Customer-defined Data Structure	17
Figure 4-1 AP/EC ANX74xx Driver Structure.....	19
Figure 4-2 Message Structure	25
Figure 4-3 Customer VDM Flow	30
Figure 4-4 PPS Control Flow	31
Figure 5-1 Accepting/Rejecting the Request Data Object.....	32
Figure 5-2 Selecting an Object Position.....	34
Figure 5-3 Calculating RDO Operating Current	35
Figure 8-1 Communication between AP Interface Code and ANX74xx Firmware	41
Figure 8-2 Communication between ANX74xx Firmware and AP Interface Code	42

List of Tables

Table 2-1 I2C_ADR_0/I2C_ADR_1 for I2C Slave Address Selection	9
Table 2-2 I2C_CTRL_DEV_ADDR (I2C Address: 0x7E; Offset: 0x5E)	9
Table 2-3 I2C_CTRL_OFFSET (I2C Address: 0x7E; Offset: 0x5F)	9
Table 2-4 I2C_CTRL (I2C Address: 0x7E; Offset: 0x60)	9
Table 2-5 I2C_CTRL_ACCESS_NUM_L8 (I2C Address: 0x7E; Offset: 0x61)	10
Table 2-6 CONFIG_X_CTRL_4 (I2C Address: 0x7E; Offset: 0x62)	10
Table 2-7 CONFIG_X_CTRL_5 (I2C Address: 0x7E; Offset: 0x63)	10
Table 2-8 CONFIG_X_CTRL_5 (I2C Address: 0x7E; Offset: 0x64)	10
Table 2-9 CONFIG_X_ACCESS_FIFO (I2C Address: 0x7E; Offset: 0x65)	10
Table 2-10 CONFIG_X_FIFO_CLEAR (I2C Address: 0x7E; Offset: 0x66)	10
Table 2-11 ADDR_INTP_SOURCE_0 (I2C Address: 0x7E; Offset: 0x67)	11
Table 2-12 ADDR_INTP_SOURCE_1 (I2C Address: 0x7E; Offset: 0x68)	11
Table 3-1 FLASH_ADDR_H (I2C Address: 0x7E; Offset: 0x0C)	13
Table 3-2 FLASH_ADDR_L (I2C Address: 0x7E; Offset: 0x0D)	13
Table 3-3 FLASH_WRITE_BUFFER_0~31 (I2C Address: 0x7E; Offset: 0x0E~0x2D)	14
Table 3-4 FLASH_LEN_H (I2C Address: 0x7E; Offset: 0x2E)	14
Table 3-5 FLASH_LEN_L (I2C Address: 0x7E; Offset: 0x2F)	14
Table 3-6 FLASH_RW_CTRL (I2C Address: 0x7E; Offset: 0x30)	14
Table 3-7 FLASH_STATUS_0 (I2C Address: 0x7E; Offset: 0x31)	14
Table 3-8 FLASH_STATUS_1 (I2C Address: 0x7E; Offset: 0x32)	15
Table 3-9 FLASH_STATUS_2 (I2C Address: 0x7E; Offset: 0x33)	15
Table 3-10 FLASH_STATUS_3 (I2C Address: 0x7E; Offset: 0x34)	15
Table 3-11 FLASH_STATUS_4 (I2C Address: 0x7E; Offset: 0x35)	15
Table 3-12 FLASH_STATUS_5 (I2C Address: 0x7E; Offset: 0x36)	15
Table 3-13 FLASH_STATUS_6 (I2C Address: 0x7E; Offset: 0x37)	15
Table 3-14 FLASH_STATUS_7 (I2C Address: 0x7E; Offset: 0x38)	15
Table 3-15 FLASH_STATUS_8 (I2C Address: 0x7E; Offset: 0x39)	15
Table 3-16 FLASH_STATUS_9 (I2C Address: 0x7E; Offset: 0x3A)	15
Table 3-17 FLASH_STATUS_10 (I2C Address: 0x7E; Offset: 0x3B)	16
Table 3-18 FLASH_READ_D0~D31 (I2C Address: 0x7E; Offset: 0x3C~0x5B)	16
Table 3-19 Firmware Version (I2C Address: 0x7E; Offset: 0xB4 and 0xB5)	16
Table 3-20 CRC Location	16
Table 4-1 Firmware Control 0 (I2C Address: 0x58; Offset: 0xB2)	19
Table 4-2 VBUS Over Current Protection Threshold Low Byte (I2C Address: 0x58; Offset: 0xDD)	20
Table 4-3 VBUS Over Current Protection Threshold High Byte (I2C Address: 0x58; Offset: 0xDE)	20
Table 4-4 Firmware Control 1 (I2C Address: 0x58; Offset: 0xF0)	20
Table 4-5 Firmware Control 2 (I2C Address: 0x58; Offset: 0xF1)	21
Table 4-6 Firmware Control 3 (I2C Address: 0x58; Offset: 0xF4)	21
Table 4-7 Maximum Voltage for Request Data Object (I2C Address: 0x7e; Offset: 0xAC)	21
Table 4-8 Maximum System Power (I2C Address: 0x7e; Offset: 0xAD)	21
Table 4-9 Minimum System Power (I2C Address: 0x7e; Offset: 0xAE)	21
Table 4-10 Maximum Voltage for Request Data Object (I2C Address: 0x7e; Offset: 0xAF)	22
Table 4-11 Maximum Power for Request Data Object (I2C Address: 0x7e; Offset: 0xB0)	22
Table 4-12 Maximum Current for Request Data Object (I2C Address: 0x7e; Offset: 0xB1)	22
Table 4-13 PD Specification Revision Configuration (I2C Address: 0x7e; Offset: 0xBA)	22
Table 4-11 Interface and Status Interrupt Mask (I2C Address: 0x7e; Offset: 0xB6)	22
Table 4-12 Internal Alert_0 Interrupt Mask (I2C Address: 0x58; Offset: 0xC9)	23

Table 4-13 Internal Alert_1 Interrupt Mask (I2C Address: 0x58; Offset: 0xCA)	23
Table 4-14 Interface and Status Interrupt (I2C Address: 0x7e; Offset: 0xB7)	23
Table 4-15 Internal Alert_0 Interrupt (I2C Address: 0x58; Offset: 0xCB)	23
Table 4-16 Internal Alert_1 Interrupt (I2C Address: 0x58; Offset: 0xCC)	23
Table 4-17 Interface and Status (I2C Address: 0x7e; Offset: 0xB8).....	24
Table 4-18 CC Status (I2C Address: 0x7e; Offset: 0xB9)	24
Table 4-19 Interface Registers.....	25
Table 4-20 Length Field of an Interface Message.....	26
Table 4-21 Messages Exchanged between AP System PD Policy Code and ANX74xx OCM Firmware	26
Table 5-1 Fixed and Variable Request Data Object	33
Table 6-1 Commands.....	36
Table 10-1 Document Revision History	48

1 Introduction

Analogix USB Type-C™ (USB-C™) crosspoint switch chips, as listed below, support switching at a data rate of up to 10 Gbps. Because most of them begin with **ANX74**, they are referred to as ANX74xx in this document.

- **ANX7447 (TX)**, USB 3.1 (10G) 4-lane DisplayPort switch with cable detection (CD) and power delivery(PD) PHY functions
- **ANX7411 (TX)**, CD + PD PHY
- **ANX7446 (TX)**, USB 3.1 (10G) & 4L DP Switch + CD + PD PHY (Pin compatible with ANX7428)
- **ANX7437 (TX)**, USB 3.1 (10G) Switch + CD + PD PHY
- **ANX7412 (RX)**, CD +PD PHY+ USB billboard
- **ANX7327 (RX)**, USB 3.1 (10G) & 4L DP Switch + CD + PD PHY +USB billboard

This document provides ANX74xx programming information, including information about I2C master/slave interface, Flash programming, and the interface communication with PD2.0 and PD3.0. Before ANX74XX is used as a Type-C Port Controller (TCPC) in a Type-C Port Controller Interface (TCPCI) implementation, the on-chip Flash memory must be erased by the Type-C Port Manager EC (TCPM/EC). Erasing the Flash memory is necessary to prevent the on-chip microcontroller (OCM) from interfering with the TCPM when it access ANX74XX. See Section 3.5 for related information.

Unless otherwise noted, all information in this guide pertains to all parts, that is, ANX7447, ANX7411, ANX7446, ANX7437, ANX7412, and ANX7327.

Figure 1-1 shows ANX74xx hardware block diagram.

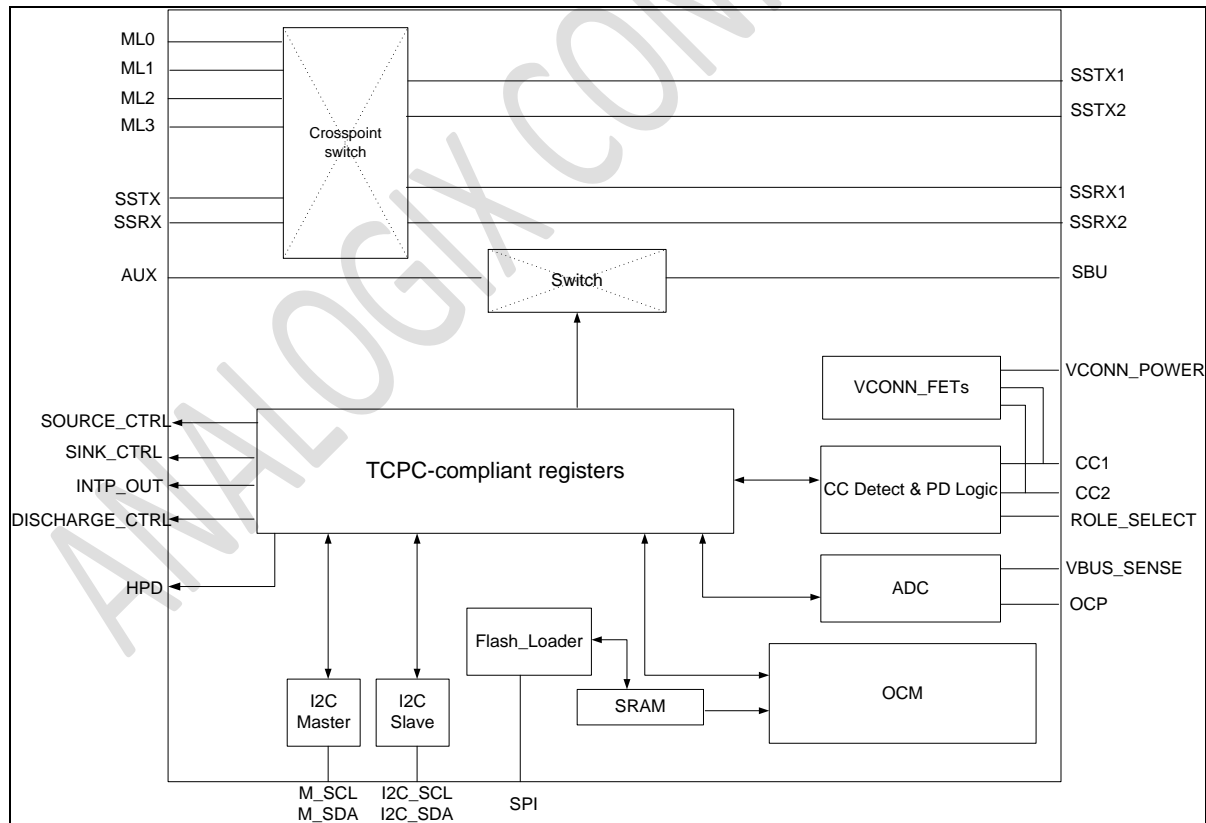


Figure 1-1 ANX74xx Hardware Block Diagram (with I2C Interface)

Figure 1-2 shows a typical ANX74xx application.

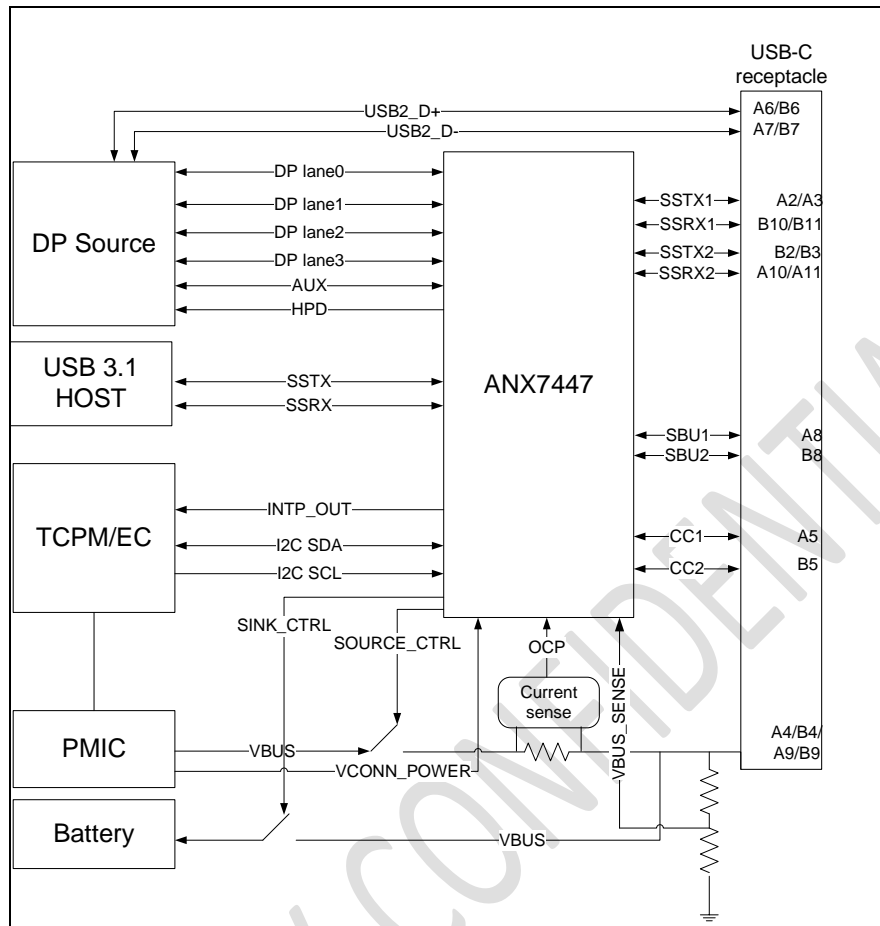


Figure 1-2 Typical ANX74xx Application

2 I2C Bus Interface

The ANX74xx has two I2C interfaces: I2C master interface and I2C slave interface.

2.1 I2C Slave

The slave I2C device address is selected by external control pins I2C_ADR_1 and I2C_ADR_0. See Table 2-1 for related information.

Table 2-1 I2C_ADR_0/I2C_ADR_1 for I2C Slave Address Selection

I2C_ADR_1	I2C_ADR_0	TCPC Slave Address (Wakeup address)	SPI Slave Address	EMTB Slave Address	EMRB Slave Address	Billboard Slave Address
0	0	8'h58	8'h7e	8'h7a	8'h84	8'h72
0	1	8'h56	8'h6e	8'h78	8'h86	8'h76
1	0	8'h54	8'h64	8'h68	8'h6c	8'h5C
1	1	8'h52	8'h62	8'h6a	8'h74	8'h82

2.2 I2C Master

ANX74xx I2C master is used for accessing peripheral devices. The registers described in this section are used to perform I2C master read/write operations and call ANX74xx I2C master functions.

Table 2-2 I2C_CTRL_DEV_ADDR (I2C Address: 0x7E; Offset: 0x5E)

Bit	Name	Type	Default	Description
7:0	I2C_CTRL_DEV_ADDR	R/W	0x0	I2C master device address

Table 2-3 I2C_CTRL_OFFSET (I2C Address: 0x7E; Offset: 0x5F)

Bit	Name	Type	Default	Description
7:0	I2C_CTRL_OFFSET	R/W	0x0	I2C master offset address

Table 2-4 I2C_CTRL (I2C Address: 0x7E; Offset: 0x60)

Bit	Name	Type	Default	Description
7	CONFIG_X_GLH_SEL	R/W	0x0	I2C master glitch select. 1: 4-cycle deglitch 0: 2-cycle deglitch
6:4	CONFIG_X_CMD	R/W	0x0	I2C master command. 000: BYTE_READ 001: BYTE_WRITE 100: I2C_RESET.
3:2	CONFIG_X_SPEED	R/W	0x0	I2C master speed. There are eight speeds.

Bit	Name	Type	Default	Description
1	CONFIG_X_NO_STOP	R/W	0x0	I2C master no stop. 1: no stop 0: stop
0	CONFIG_X_NO_ACK	R/W	0x0	I2C master ACK. 1: no ACK 0: ACK

Table 2-5 I2C_CTRL_ACCESS_NUM_L8 (I2C Address: 0x7E; Offset: 0x61)

Bit	Name	Type	Default	Description
7:0	CONFIG_X_ACCESS_NUM_LOW	R/W	0x0	I2C master access number (low byte)

Table 2-6 CONFIG_X_CTRL_4 (I2C Address: 0x7E; Offset: 0x62)

Bit	Name	Type	Default	Description
7:3	CONFIG_X_DDC_STATE	R/O	0x0	I2C master DDC FSM state
2	CONFIG_X_MODE	R/W	0x0	I2C master mode. See the CONFIG_X_SPEED register for related information.
1:0	CONFIG_X_ACCESS_NUM_HIGH	R/W	0x0	I2C master access number (high byte)

Table 2-7 CONFIG_X_CTRL_5 (I2C Address: 0x7E; Offset: 0x63)

Bit	Name	Type	Default	Description
7:3	CONFIG_X_TX_STATE	R/O	0x0	config_x tx FSM state
2:0	CONFIG_X_RC_STATE	R/O	0x0	config_x rc FSM state

Table 2-8 CONFIG_X_CTRL_5 (I2C Address: 0x7E; Offset: 0x64)

Bit	Name	Type	Default	Description
7:3	CONFIG_X_FIFO_COUNT	R/O	0x0	config_x FIFO count. The FIFO depth is 16.
2:0	CONFIG_X_FIFO_STATE	R/O	0x0	config_x FIFO state

Table 2-9 CONFIG_X_ACCESS_FIFO (I2C Address: 0x7E; Offset: 0x65)

Bit	Name	Type	Default	Description
7:0	CONFIG_X_ACCESS_FIFO	R/W	0x0	config_x FIFO RW interface.

Table 2-10 CONFIG_X_FIFO_CLEAR (I2C Address: 0x7E; Offset: 0x66)

Bit	Name	Type	Default	Description
7	CONFIG_X_FIFO_FULL	R/O	0x0	config_x FIFO.
6	CONFIG_X_FIFO_EMPTY	R/O	0x0	config_x FIFO.
5:1	Reserved			
0	CONFIG_X_FIFO_CLEAR	R/W	0x0	config_x FIFO clear. Write 1 or write 0 to clear.

Table 2-11 ADDR_INTP_SOURCE_0 (I2C Address: 0x7E; Offset: 0x67)

Bit	Name	Type	Default	Description
7:0	INTP_SOURCE_0	W/C	0x0	BIT7: reserved. BIT6: int_hpd_unplug. BIT5: int_hpd_plug. BIT4: int_hpd_irq. BIT3: int_config_x_fifo_full. BIT2: int_config_x_fifo_empty. BIT1: int_config_x_fifo_half. BIT0: int_config_x_acc_done.

Table 2-12 ADDR_INTP_SOURCE_1 (I2C Address: 0x7E; Offset: 0x68)

Bit	Name	Type	Default	Description
7:0	INTP_SOURCE_1	W/C	0x0	BIT7: r_soft_intp[7]. BIT6: r_soft_intp[6]. BIT5: r_soft_intp[5]. BIT4: r_soft_intp[4]. BIT3: r_soft_intp[3]. BIT2: r_soft_intp[2]. BIT1: r_soft_intp[1]. BIT0: r_soft_intp[0].

The following gives the steps to perform I2C master write/read operations to access peripheral devices through I2C master interface.

To perform an I2C master write operation:

1. Set the device address by setting the **I2C_CTRL_DEV_ADDR** register.
2. Set the device offset address by setting the **I2C_CTRL_OFFSET** register.
3. Set the length (in bytes) by setting the **I2C_CTRL_ACCESS_NUM_L8** and **CONFIG_X_CTRL_4** registers.
4. Write the desired data to the **CONFIG_X_ACCESS_FIFO** register.
5. Set the parameters of the write operation such as speed by setting the **I2C_CTRL** register.
6. Wait till bit 0 of the **ADDR_INTP_SOURCE_0** register becomes 1, which means that the I2C write operation finishes.
7. Clear bit 0 of the **ADDR_INTP_SOURCE_0** register for the next I2C operation.

To perform an I2C master read operation:

1. Set the device address by setting the **I2C_CTRL_DEV_ADDR** register.
2. Set the device offset address by setting the **I2C_CTRL_OFFSET** register.
3. Set length (in bytes) by setting the **I2C_CTRL_ACCESS_NUM_L8** and **CONFIG_X_CTRL_4** registers.
4. Set the parameters of the read operation such as speed by setting the **I2C_CTRL** register.
5. Wait till bit 0 of the **ADDR_INTP_SOURCE_0** register becomes 0, which means that the read operation finishes.
6. Clear bit 0 of the **ADDR_INTP_SOURCE_0** register for next I2C operation.
7. Read data from the **CONFIG_X_ACCESS_FIFO** register.

3 Flash Programming

ANX74xx has a 128KB Flash to store Firmware images. The Flash comprises two blocks for Firmware images (56KB x 2), 1 byte for firmware pointer, and a 4KB space for customers, as shown in Figure 3-1.

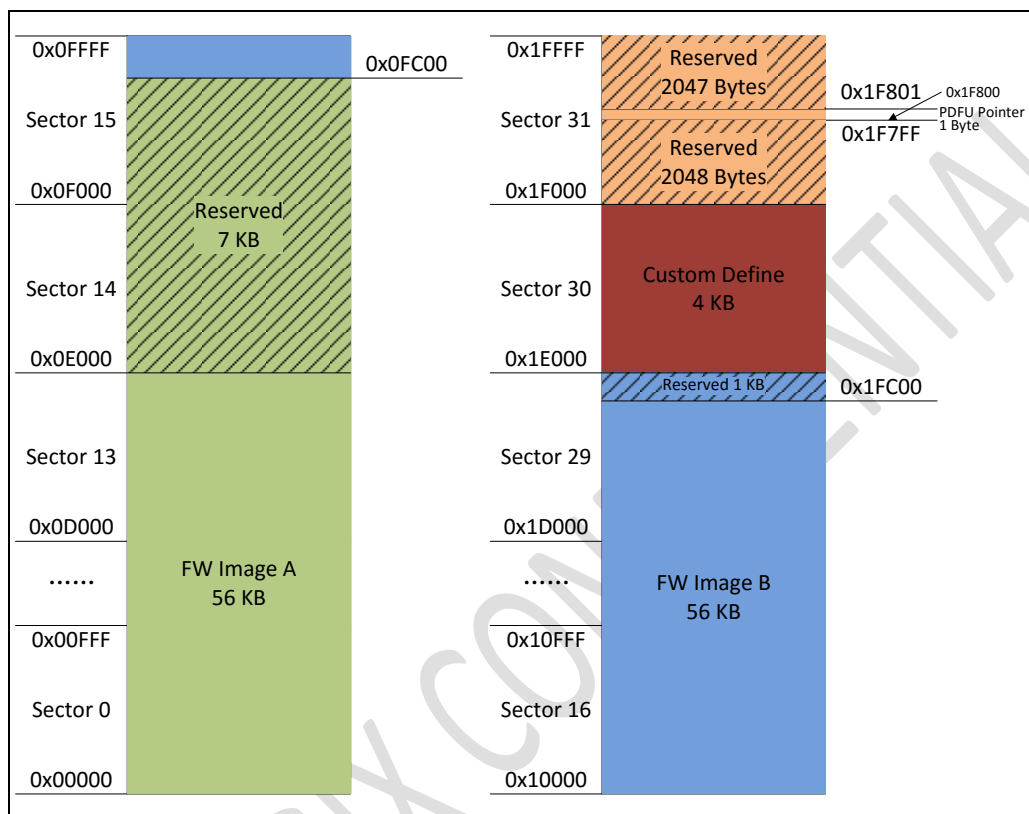


Figure 3-1 ANX74xx Flash Space Allocation

3.1 Flash Blocks for Different Versions of Firmware

Two blocks are allocated in Flash for Firmware images, each of which can hold a specific version of an ANX74xx Firmware image. The firmware pointer (0x1F800) indicates the block that holds the most up-to-date version of the firmware.

After power-on, ANX74xx loads the firmware image from the block the firmware pointer points to OCM SRAM and the ANX74xx firmware starts.

The space for customer is for customization, such as USB PD, USB Type-C, ANX74xx control, and so on.

3.2 Firmware Image Area Updating through I2C

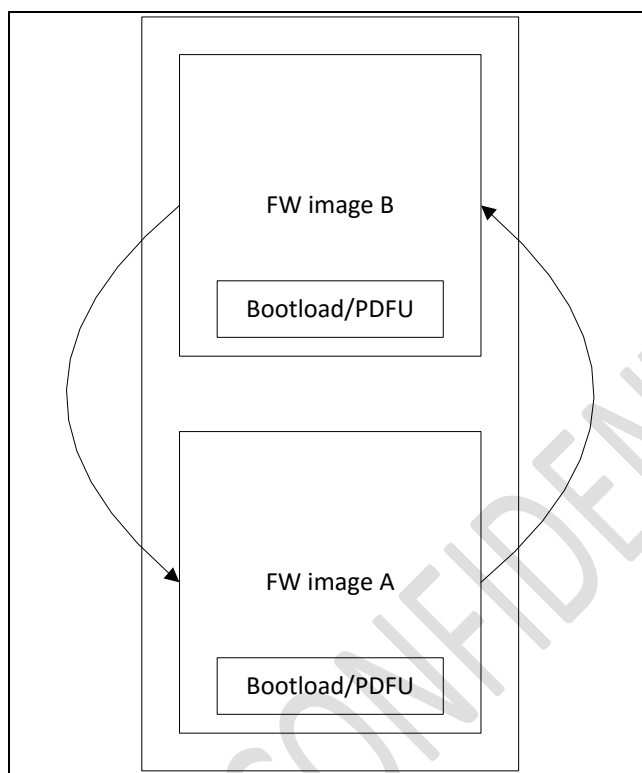


Figure 3-2 Firmware Image Updating

Currently, the firmware images stored in ANX74xx Flash can be updated in two ways: loading Firmware image from UART port and updating firmware through common SPI interface programming; PDFU specification for firmware updating through USB Type-C CC channels and PDFU standard procedures. For information about PDFU state machine, see PDFU specification.

As Figure 3-2 shows, ANX74xx was designed to update the firmware in one of the two ways: the common SPI Flash update procedures or following PDFU specification to update the two different firmware images. After flash programming was done, all blocks should be read-only to protect Flash content from inadvertent erasure or programming. Firmware checks the Flash again to make sure all blocks are protected in Flash initialization in case the firmware fails to do this during the last firmware updating.

ANX74xx can access on-chip Flash through external AP/EC I2C interfaces and a programmer can use the I2C interface to access and control ANX74xx Flash-related registers for Flash read and write features. The registers described in this section are used for ANX74xx OCM Flash memory access.

Table 3-1 FLASH_ADDR_H (I2C Address: 0x7E; Offset: 0x0C)

Bit	Name	Type	Default	Description
7:0	FLASH_ADDR_H	R/W	0x0	Flash address 15:8 for reading/writing specific location of Flash. This byte needs to be configured.

Table 3-2 FLASH_ADDR_L (I2C Address: 0x7E; Offset: 0x0D)

Bit	Name	Type	Default	Description
7:0	FLASH_ADDR_L	R/W	0x0	Flash address 7:0 for reading/writing specific location

Bit	Name	Type	Default	Description
				of Flash. This byte needs to be configured.

Table 3-3 FLASH_WRITE_BUFFER_0~31 (I2C Address: 0x7E; Offset: 0x0E~0x2D)

Offset	Name	Type	Default	Description
0xe	FLASH_WRITE_DATA_0	R/W	0x0	Flash write data byte 0
...	FLASH_WRITE_DATA_m	R/W	0x0	Flash write data byte m
0x2d	FLASH_WRITE_DATA_31	R/W	0x0	Flash write data byte 31

Table 3-4 FLASH_LEN_H (I2C Address: 0x7E; Offset: 0x2E)

Bit	Name	Type	Default	Description
7:0	FLASH_LEN_H	R/W	0x0	Flash read or write length high byte

Table 3-5 FLASH_LEN_L (I2C Address: 0x7E; Offset: 0x2F)

Bit	Name	Type	Default	Description
7:0	FLASH_LEN_L	R/W	0x0	Flash read or write length low byte

Table 3-6 FLASH_RW_CTRL (I2C Address: 0x7E; Offset: 0x30)

Bit	Name	Type	Default	Description
7	READ_DELAY_SELECT	R/W	0x0	Flash SPI controller debugging register. The default value is recommended.
6	GENERAL_INSTRUCTION_EN	S/C	0x0	Enables generating instruction sending. A value of 1 makes the Flash controller to send a one-byte instruction.
5	FLASH_ERASE_EN	S/C	0x0	Enables Flash content erasing. A value of 1 sends a Flash erasing command.
4	RDID_READ_EN	S/C	0x0	Enables RDID instruction reading. A value of 1 sends an RDID read command to the Flash (depending on Flash vendor definition.)
3	REMS_READ_EN	S/C	0x0	Enables REMS instruction reading. A value of 1 sends an REMS read command to the Flash (depending on Flash vendor definition.)
2	WRITE_STATUS_EN	S/C	0x0	Enables Flash status register writing. A value of 1 sends a writing status register command to the Flash, status register related to 0x31.
1	FLASH_READ	S/C	0x0	Enables Flash read. Pulse signal, self-cleared.
0	FLASH_WRITE	S/C	0x0	Enables Flash writing. Pulse signal, self-cleared.

Table 3-7 FLASH_STATUS_0 (I2C Address: 0x7E; Offset: 0x31)

Bit	Name	Type	Default	Description
7:0	STATUS_REGISTER_IN	R/W	0x0	Flash status register. When using write status register instruction, configure this byte as register value.

Table 3-8 FLASH_STATUS_1 (I2C Address: 0x7E; Offset: 0x32)

Bit	Name	Type	Default	Description
7:0	REMS_READ_ADDR	R/W	0x0	REMS read address parameter

Table 3-9 FLASH_STATUS_2 (I2C Address: 0x7E; Offset: 0x33)

Bit	Name	Type	Default	Description
7:0	GENERAL_INSTRUCTION_TYPE	R/W	0x0	General instruction parameter

Table 3-10 FLASH_STATUS_3 (I2C Address: 0x7E; Offset: 0x34)

Bit	Name	Type	Default	Description
7:0	FLASH_ERASE_TYPE	R/W	0x0	When using an erasing instruction with address parameter, this register defines the instruction type.

Table 3-11 FLASH_STATUS_4 (I2C Address: 0x7E; Offset: 0x35)

Bit	Name	Type	Default	Description
7:0	STATUS_REGISTER_LOW_7_0	R/O	0x0	Flash status register read out

Table 3-12 FLASH_STATUS_5 (I2C Address: 0x7E; Offset: 0x36)

Bit	Name	Type	Default	Description
7:0	MANUFACTURE_ID	R/O	0x0	After executing an REMS read instruction, this register restores Flash manufacture ID.

Table 3-13 FLASH_STATUS_6 (I2C Address: 0x7E; Offset: 0x37)

Bit	Name	Type	Default	Description
7:0	DEVICE_ID	R/O	0x0	After executing an REMS read instruction, this register restores Flash device ID.

Table 3-14 FLASH_STATUS_7 (I2C Address: 0x7E; Offset: 0x38)

Bit	Name	Type	Default	Description
7:0	MEM_TYPE	R/O	0x0	After executing an REMS read instruction (instruction 0x90), this register restores Flash MEM type.

Table 3-15 FLASH_STATUS_8 (I2C Address: 0x7E; Offset: 0x39)

Bit	Name	Type	Default	Description
7:0	CAPACITY	R/O	0x0	After executing an RDID read instruction (instruction 0x9f), this register restore flash capacity.

Table 3-16 FLASH_STATUS_9 (I2C Address: 0x7E; Offset: 0x3A)

Bit	Name	Type	Default	Description
7:0	READ_DATA_OUT_OCM_CLK	R/O	0x0	The last byte read from Flash by Flash read access

Table 3-17 FLASH_STATUS_10 (I2C Address: 0x7E; Offset: 0x3B)

Bit	Name	Type	Default	Description
7:4	SPI_LINK_CURRENT_STATE_READ	R/O	0x0	Flash controller read states debug
3:0	SPI_LINK_CURRENT_STATE	R/O	0x0	Flash controller main states debug

Table 3-18 FLASH_READ_D0~D31 (I2C Address: 0x7E; Offset: 0x3C~0x5B)

Offset	Name	Type	Default	Description
0x3c	FLASH_READ_DATA_0	R/O	0x0	Flash read data 0
...	FLASH_READ_DATA_m	R/O	0x0	Flash read data m
0x5b	FLASH_READ_DATA_31	R/O	0x0	Flash read data 31

Table 3-19 Firmware Version (I2C Address: 0x7E; Offset: 0xB4 and 0xB5)

Offset	Name	Type	Default	Description
0xB4	Firmware main version number	R/O	0x0	
0xB5	Firmware build version	R/O	0x0	

3.3 Firmware Integrity Checking

3.3.1 Firmware CRC

As shown in Figure 3-1, each firmware image area (FW image A or B) contains a 4-byte CRC checksum located in the last four bytes of the firmware image area (as listed in the below table. When Firmware image was load from Flash to RAM during power on stage, ANX74xx hardware will check the CRC, only if CRC is correct this firmware can be loaded into OCM RAM.

Table 3-20 CRC Location

Name	CRC Location in Flash			
FW image A	0xDFFC	0xDFFD	0xDFFE	0xDFFF
FW image B	0x1BFC	0x1BFD	0x1BFE	0x1BFF

3.3.2 Generating the Internal MCU Firmware Image with CRC

An MPEG tool provided by Analogix can generate the OCM firmware after OCM firmware is built successfully.

3.4 Customer-defined Zone

The customer-defined zone stores system parameters for the chip to communicate with the USB charger when the system is in the dead battery mode. These parameters are loaded by OCM firmware automatically after chip power-on. They are also used as default USB PD parameters by OCM firmware in systems without Interface driver or without EC/SOC.

There are two data zones in the space ranging from 0x1E000 to 0x1E7b0, each of which includes data content and CRC (address offset: 32 bytes.) The data content is the same as interface message described in section 4.7.2. Figure 3-3 shows customer-defined data structure.

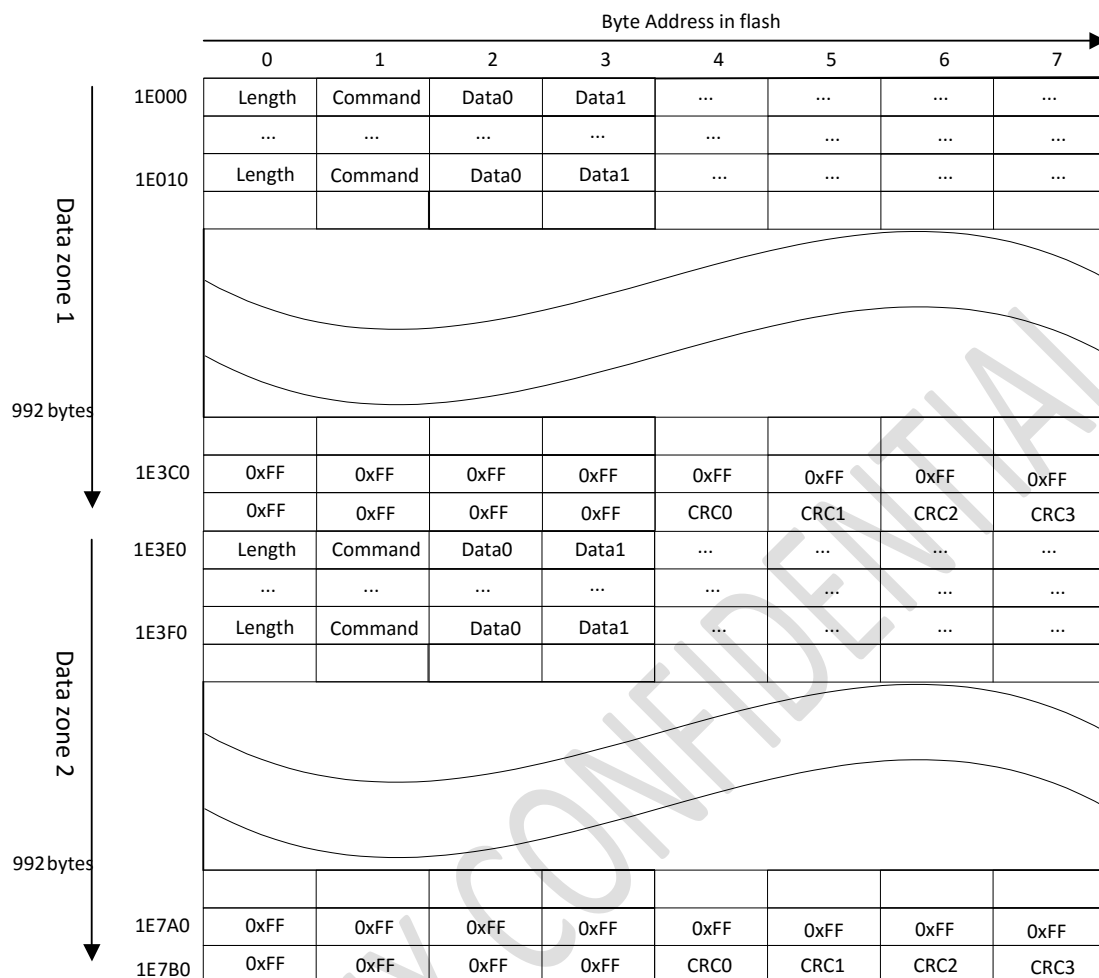


Figure 3-3 Customer-defined Data Structure

Customer-specific data can be generated using Customer Data Tool. See *Analogix Customer Code Generate Tool User Guide* for related information.

3.5 Erasing Flash Memory

When ANX74XX is used as a Type-C Port Controller (TCPC) in a Type-C Port Controller Interface (TCPCI) implementation, the Flash memory needs to be erased. The following gives the register operation needed.

```
/* Reset OCM */

write_byte_or(0x7e, 0x6e, 0x40)

/* disable hardware protected mode (HPM) */

write_byte_or(0x7e, 0x88, 0x80);

/* disable flash write protection */

write_byte_and(0x7e, 0x31, 0x43);

write_byte_or(0x7e, 0x30, 0x04);
```

```
/*wait for flash write protection disable done*/  
  
while(!(read_byte(0x7e, 0x05) & 0x80));  
  
/* flash write enable */  
  
    write_byte(0x7e, 0x33 , 0x06);  
  
    write_byte _or(0x7e, 0x30, 0x40);  
  
        /* wait for write enable sequence done */  
  
        while(!(read_byte(0x7e, 0x05) & 0x80));  
  
/* write flash chip erase instruction */  
  
    write_byte(0x7e, 0x34 , 0x60);  
  
    delay_ms(15);  
  
/* flash chip erase enable */  
  
    write_byte _or(0x7e, 0x30 , 0x20);  
  
/* wait for chip erase sequence done */  
  
    while(!(read_byte(0x7e, 0x05) & 0x80));
```

4 ANX74xx Driver on AP/EC Side

This section describes the events occurring on the AP/EC side when a typical USB Type-C device is plugged in.

The ANX74xx has an interrupt pin for the ANX74xx to inform the driver. When AP/EC receives an interrupt signal, it checks the related status registers to know what was happen.

4.1 ANX74xx Driver Overview

The ANX74xx enters active mode when a USB Type-C device is connected to it. When the ANX74xx receives an interrupt signal through the Alter pin, the driver behaviors according to the related status registers.

When a USB Type-C device is disconnected from the ANX74xx, the ANX74xx driver receives a power-off interrupt signal and then enters standby mode.

4.2 AP/EC Driver Structure

The ANX74xx informs AP/EC of events through interrupts and AP/EC driver handles an interrupt signal receive according to the status of related registers, as shown in Figure 4-1.

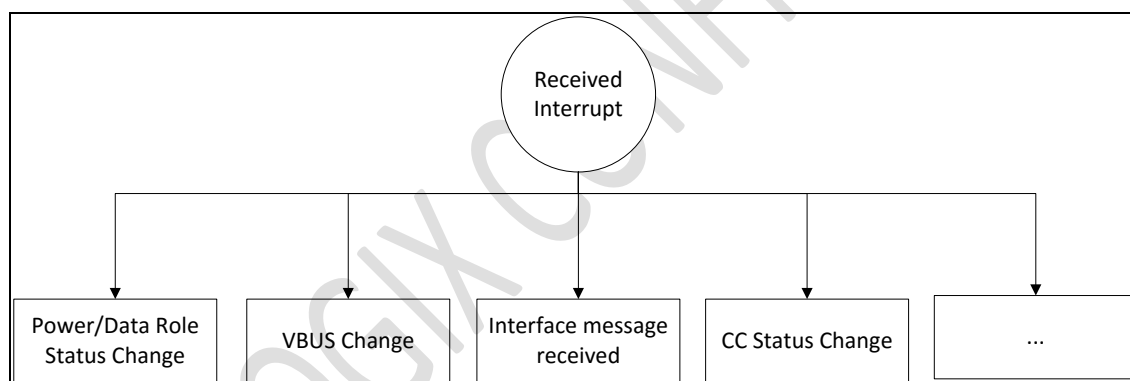


Figure 4-1 AP/EC ANX74xx Driver Structure

4.3 Initialization Settings

4.3.1 Firmware Control

The firmware control register (I2C Address 0x58 and Offset 0xB2) is designed by firmware option function settings and handling. You can control this register for specific firmware function.

Table 4-1 Firmware Control 0 (I2C Address: 0x58; Offset: 0xB2)

Bit	Name	Type	Default	Description
7	low_voltage_for_same_power	R/W	0x0	Selects voltages for the same power case. 0: Selects high voltage. 1: Selects low voltage.
6	force_send_rdo	R/W	0x0	0: Does not force to send RDO objects.

Bit	Name	Type	Default	Description
				1: Forces to send RDO objects.
5	snk_remove_refer_cc	R/W	0x0	0: Disables sink cable remove status will check CC status. 1: Enable sink cable remove status will check CC status.
4	support_goto_min_power	R/W	0x0	0: Disables goto_min_power under automatic PD mode; 1: Enables goto_min_power under automatic PD mode. If Minimum_Power device requirement is not met, the bit means that the OCM will choose the closest or safe 5V.
3	Try_Snk_enable	R/W	0x0	0: Disables try.Snk feature; 1: Enables try.Snk feature.
2	Try_Src_enable	R/W	0x0	0: Disables try.Src feature; 1: Enables try.Src feature.
1	AUTO_PD_EN	R/W	0x0	0: Disables automatic PD mode; 1: Enables automatic PD mode.
0	disable_usb30	R/W	0x0	0: Disables USB 3.0 super send token support; 1: Enables USB 3.0 super speed token support.

Table 4-2 VBUS Over Current Protection Threshold Low Byte (I2C Address: 0x58; Offset: 0xDD)

Bit	Name	Type	Default	Description
7:0	VBUS_OCP_HI_THRESHOLD_BIT7_0	R/W	0x0	10-bit for vbus current threshold.

Table 4-3 VBUS Over Current Protection Threshold High Byte (I2C Address: 0x58; Offset: 0xDE)

Bit	Name	Type	Default	Description
7:2	Reserved			
1:0	VBUS_OCP_HI_THRESHOLD_BIT9_8	R/W	0x0	10-bit for vbus current threshold.

Table 4-4 Firmware Control 1 (I2C Address: 0x58; Offset: 0xf0)

Bit	Name	Type	Default	Description
7:5	Reserved	R/W	0x00	Reserved
4	LEXINGTON_DRIVER_ENABLE	R/W	0x00	0: Disables Lexington driver on OCM firmware. 1: Enables Lexington driver on OCM firmware.
3:2	VSAFE0V_OFFSET_VOLTAGE	R/W	0x00	Increase vsafe0 value with 1V unit.
1	PD_TASK_RUN_ENABLE	R/W	0x00	0: 200ms delay after OCM boot to enter PD task. 1: enter PD task without delay.
0	UNSTRUC_VDM_ENABLE	R/W	0x00	0: Disables unstructured VDM PD message. 1: Enables unstructured VDM PD message.

Table 4-5 Firmware Control 2 (I2C Address: 0x58; Offset: 0xf1)

Bit	Name	Type	Default	Description
7:6	Reserved	R/W	0x00	Reserved
5:3	FRSWAP_VSAFE_OFFSET	R/W	0x0	New source output vbus when vbus voltage drop below vsfave5v_max (5.5V); setting will boost the point with unit 500mV.
2	PPS_FUNC_EN	R/W	0x0	0: Disables PPS function; 1: Enables PPS function for sink.
1	FR_SWAP_TRIGGER_EN	R/W	0x0	0: no action; 1: Enables fast role swap signal trigger.
0	FR_SWAP_CTL_MODE	R/W	0x0	0: Fast role swap signal triggered by OCM firmware; 1: Fast role swap signal triggered by bit set of FR_SWAP_TRIGGER_EN.

Table 4-6 Firmware Control 3 (I2C Address: 0x58; Offset: 0xf4)

Bit	Name	Type	Default	Description
7:3	Reserved	R/W	0x0	Reserved
2	UFP_DR_SWAP_TRIGGER	R/W	0x0	0: Disables PPS function; 1: data role swap signal triggered if it is as initial UFP.
1	DFP_DR_SWAP_TRIGGER	R/W	0x0	0: no action; 1: data role swap signal triggered if it is as initial DFP.
0	PREFER_FUNC_DISABLE	R/W	0x0	0: Vconn swap & data role swap triggered by OCM firmware; 1: no action.

4.3.2 Auto USB PD Negotiation Registers

The ANX74xx can implement USB PD negotiation automatically based on customer requirements. The following describes the related registers. See Section 5 for related information.

Table 4-7 Maximum Voltage for Request Data Object (I2C Address: 0x7e; Offset: 0xAC)

Bit	Name	Type	Default	Description
7:0	Maximum_Voltage	R/W	0x32	The maximum voltage (in 100mV). The default is 5V.

Table 4-8 Maximum System Power (I2C Address: 0x7e; Offset: 0xAD)

Bit	Name	Type	Default	Description
7:0	Maximum_Power	R/W	0x0F	The maximum system power of system (in 500mW). The default is 7.5W.

Table 4-9 Minimum System Power (I2C Address: 0x7e; Offset: 0xAE)

Bit	Name	Type	Default	Description
7:0	Minimum_Power	R/W	0x02	The minimum system power (in 500mW). The default is 1W.

Table 4-10 Maximum Voltage for Request Data Object (I2C Address: 0x7e; Offset: 0xAF)

Bit	Name	Type	Default	Description
7:0	RDO_Max_Voltage	R/O	0x00	The maximum voltage of RDO. In power consumer mode, this value means that the AP got the maximum voltage.

Table 4-11 Maximum Power for Request Data Object (I2C Address: 0x7e; Offset: 0xB0)

Bit	Name	Type	Default	Description
7:0	RDO_Max_Power	R/O	0x00	The maximum power of RDO. In power consumer mode, this value means that the AP got the maximum power.

Table 4-12 Maximum Current for Request Data Object (I2C Address: 0x7e; Offset: 0xB1)

Bit	Name	Type	Default	Description
7:0	RDO_Max_Current	R/O	0x00	The maximum current of RDO. In power consumer mode, this value means that the AP got the maximum current.

Table 4-13 PD Specification Revision Configuration (I2C Address: 0x7e; Offset: 0xBA)

Bit	Name	Type	Default	Description
7:3	Reserved	R/W	0x0	Reserved
5:4	PD_REV_USED	R/O	0x0	Port to Port operating PD Revision. 0: The same as Firmware initiation PD Revision. 1: PD 2.0.
3:2	Reserved	R/W	0x0	Reserved
1:0	PD_REV_INIT	R/W	0x0	Firmware initiation PD Revision. 0: PD3.0. 1:PD2.0.

4.4 Interrupt Registers

Table 4-14 Interface and Status Interrupt Mask (I2C Address: 0x7e; Offset: 0xB6)

Bit	Name	Type	Default	Description
7	Reserved	R/W	0x0	Reserved
6	PR_C_GOT_POWER_MASK	R/W	0x0	Reserved
5	DATA_ROLE_CHG_INT_MASK	R/W	0x0	The mask for data role changed interrupt.
4	CC_STATUS_CHG_INT_MASK	R/W	0x0	The mask for CC status changed interrupt
3	VBUS_CHG_INT_MASK	R/W	0x0	The mask for VBUS changed interrupt.
2	VCONN_CHG_INT_MASK	R/W	0x0	The mask for VCONN changed interrupt.
1	RECEIVED_ACK	R/W	0x0	Reserved
0	Reserved	R/W	0x0	Reserved

Table 4-15 Internal Alert_0 Interrupt Mask (I2C Address: 0x58; Offset: 0xC9)

Bit	Name	Type	Default	Description
7	INTR_RECEIVED_MSG	R/W	0x00	The mask for ANX74xx received SOPx message interrupt.
6	INTR_SOFTWARE_INT	R/W	0x00	The mask for ANX74xx got software interrupt.
5:0	INTR_ALERT_MASK_0	R/W	0x00	Mask register for INTR_ALERT_0 0: Interrupt masked, 1: Interrupt unmasked

Table 4-16 Internal Alert_1 Interrupt Mask (I2C Address: 0x58; Offset: 0xCA)

Bit	Name	Type	Default	Description
7	INTR_INTP_POW_ON	R/W	0x00	The mask for ANX74xx Power ON interrupt.
6	INTR_INTP_POW_OFF	R/W	0x00	The mask for ANX74xx power-off interrupt.
5:0	INTR_ALERT_MASK_1	R/W	0x00	Mask register for INTR_ALERT_1. 0: Interrupt masked; 1: Interrupt unmasked

Table 4-17 Interface and Status Interrupt (I2C Address: 0x7e; Offset: 0xB7)

Bit	Name	Type	Default	Description
7	Reserved	R/W	0x0	Reserved
6	PR_C_GOT_POWER	R/W	0x0	Reserved.
5	DATA_ROLE_CHG_INT	R/W	0x0	Data role changed interrupt.
4	CC_STATUS_CHG_INT	R/W	0x0	CC status changed interrupt
3	VBUS_CHG_INT	R/W	0x0	VBUS changed interrupt.
2	VCONN_CHG_INT	R/W	0x0	VCONN changed interrupt.
1	RECEIVED_ACK	R/W	0x0	Reserved
0	Reserved	R/W	0x0	Reserved.

Table 4-18 Internal Alert_0 Interrupt (I2C Address: 0x58; Offset: 0xCB)

Bit	Name	Type	Default	Description
7	INTR_RECEIVED_MSG	R/W	0x00	ANX74xx received SOPx message interrupt.
6	INTR_SOFTWARE_INT	R/W	0x00	ANX74xx got software interrupt.
5:0	INTR_ALERT_0_OUT	R/W	0x00	Please refer ANX74xx register specification for details.

Table 4-19 Internal Alert_1 Interrupt (I2C Address: 0x58; Offset: 0xCC)

Bit	Name	Type	Default	Description
7	INTR_INTP_POW_ON	R/W	0x00	ANX74xx Power ON interrupt
6	INTR_INTP_POW_OFF	R/W	0x00	ANX74xx power-off interrupt
5:0	INTR_ALERT_1_OUT	R/W	0x00	Refer to ANX74xx register specification for details.

Table 4-20 Interface and Status (I2C Address: 0x7e; Offset: 0xB8)

Bit	Name	Type	Default	Description
7	HPD_STATUS	R/O	0x0	0: HPD Low. 1: HPD High
6	SUPPORT_UNCHUNKING	R/O	0x0	0: Chunking. 1: Chunked.
5	Data Role	R/O	0x0	Data Role Status. 1: DFP; 0: UFP;
4	Reserved	R/O	0x0	Reserved
3	VBUS_STATUS	R/O	0x0	VBUS status. 1: power provider; 0: power consumer;
2	VCONN_STATUS	R/O	0x0	VCONN status 1: VCONN On 0: VCONN Off
1:0	Reserved	R/O	0x0	Reserved

Table 4-21 CC Status (I2C Address: 0x7e; Offset: 0xB9)

Bit	Name	Type	Default	Description
7:4	CC2 Status	R/O	0x0	0000b: SRC.Open 0001b: SRC.Rd 0010b: SRC.Ra 0100b: SNK.Default 1000b: SNK.Power1.5 1100b: SNK.Power3.0 All other values are reserved
3:0	CC1 Status	R/O	0x0	0000b: SRC.Open 0001b: SRC.Rd 0010b: SRC.Ra 0100b: SNK.Default 1000b: SNK.Power1.5 1100b: SNK.Power3.0 All other values are reserved

4.5 VBUS Control

ANX74xx VBUS is controlled by OCM and followed the power role is DFP or UFP to control **SOURCE_CTRL** and **SINK_CTRL** pins for external power management circuits. AP/EC will get VBUS change interrupt and programmer can check or control some status for different power role.

If the power role is DFP, the ANX74xx will control **SOURCE_CTRL** pin to high and **SINK_CTRL** pin to low; similarly, if the power role is UFP, ANX74xx will control **SINK_CTRL** pin to high and **SOURCE_CTRL** to low.

As listed below, the AP/EC will get VBUS change and a programmer can control own features for different status.

1. Read Alert_0 interrupt INTR_SOFTWARE_INT status listed in Table 4-18.
2. Confirm and read VBUS_CHG_INT status set to 1 in Table 4-17.
3. Read VBUS_STATUS in Table 4-20.
4. Remove VBUS_CHG_INT bit.
5. Remove INTR_SOFTWARE_INT bit.
6. The AP/EC check VBUS status based on VBUS_STATUS bit and process related features.

4.6 CC/VCONN/Data Role Status

The AP/EC gets this status through INTP_OUT interrupt and CC/VCON/Data Role status.

1. Remove the related MASK bit in Table 4-14.
2. INTP_OUT pin triggers an ISR.
3. Read the related change status in Table 4-17.
4. Read the related status in Table 4-20 and Table 4-21.
5. Remove the related status change bit in Table 4-17.

4.7 USB PD Interface

4.7.1 Interface Registers Introduction

The PD messaging interface is implemented with two software pre-defined RX/TX data register buffers and software interrupt for communication between ANX74xx OCM F/W and AP/EC F/W. Table 4-22 shows the two data registers for RX and TX and length for two data registers are 32 bytes.

Table 4-22 Interface Registers

Offset	Name	Type	Default	Description
0x7EC0~07ExDF	INTERFACE_RX_BUF	RO	0x00	RX data buffer for receive message from OCM.
0x7EE0~0x7EFF	INTERFACE_TX_BUF	R/W	0x00	TX data buffer for transfer message to OCM
0x58CD	TCPC_CTRL_2	R/W	0x00	Bit 3: SOFT_INTP_3 for inform AP/EC for TX data register is ready to transfer.

4.7.2 Interface Message Construction

An interface message comprises these fields: length (1 byte), command type (1 byte), raw data (up to 29 bytes), and checksum (1 byte). The length field holds the sum of the bytes occupied by the command type and raw data fields; and the checksum field holds the sum of the length, command type, and raw data fields.



Figure 4-2 Message Structure

4.7.2.1 Interface Message Length

The length field of an interface message holds the total length and SOP type of the message. See Table 4-23 for related information.

Table 4-23 Length Field of an Interface Message

Bit(s)	Field Name	Description
4:0	Message Length	Total Message Length for TX or RX
7:5	Message SOP Type	SOP type for SOP, SOP', SOP'' etc.

4.7.2.2 Command Type

All **TYPE_XXXXX_XXXX** commands are supported. The commands need to be changed to the **INTERFACE_CMD_XXXXX_XXXX** format for being transferred or processed after received. For example, the **TYPE_FR_SWAP** command needs to be changed to **INTERFACE_CMD_FR_SWAP** for TX/RX interface communication. See Table 4-24 for related information.

4.7.2.3 Raw Data

PD message raw data exchanged between AP/EC and OCM. The maximum length of the raw data field is 29 bytes.

4.7.2.4 Message Checksum

Length + Command Type + Raw Data 0 + Raw Data 1 + ... + Raw Data N + Checksum = 0

4.7.3 Interface Message Communication

Table 4-24 summarizes the messages exchanged between the AP System PD Policy Code and the ANX74xx OCM FW.

Table 4-24 Messages Exchanged between AP System PD Policy Code and ANX74xx OCM Firmware

Message Type	From/To	Description	Notes
Power source capability object <i>TYPE_PWR_SRC_CAP (0x00)</i>	System to ANX74xx	The system is sending its power sourcing capability objects.	(PD 2.0)
	ANX74xx to system	The system is getting the power source's capability objects from the connected device.	(PD 2.0)
Power sink capability object <i>TYPE_PWR_SNK_CAP (0x01)</i>	System to ANX74xx	The system is sending its power sinking capability objects.	(PD 2.0)
	ANX74xx to system	The system is getting the power sink's capability objects from the connected device.	(PD 2.0)
<i>TYPE_DP_SNK_IDENTITY (0x02)</i>	System to ANX74xx	The system is sending its DP sink's identity.	(PD 2.0)
	ANX74xx to system	The system is getting the DP source's capability objects.	(PD 2.0)
SVID <i>TYPE_SVID (0x03)</i>	ANX74xx to system	The ANX74xx reports SVID of the connected device.	(PD 2.0)
<i>TYPE_GET_DP_SNK_CFG (0x04)</i>	System to ANX74xx	The system wants to get DP sink configuration. When ANX74xx receives it, DP sink configuration sends to System from ANX74xx.	(PD 2.0)
Accept a request <i>TYPE_ACCEPT (0x05)</i>	System to ANX74xx	The system accepts a request.	(PD 2.0)

Message Type	From/To	Description	Notes
Reject a request <i>TYPE_REJECT (0x06)</i>	System to ANX74xx	The system rejects a request.	(PD 2.0)
<i>TYPE_SET_SNK_DP_CAP (0x08)</i>	System to ANX74xx	The system sets ANX74xx DP sink capability.	(PD 2.0)
<i>TYPE_GET_SRC_CAP (0x09)</i>	System to ANX74xx	The system sets ANX74xx DP sink capability.	(PD 2.0)
<i>TYPE_GET_RDO (0x0A)</i>	System to ANX74xx	The system gets ANX74xx current RDO.	(PD 2.0)
Power role swap request <i>TYPE_PSWAP_REQ (0x10)</i>	System to ANX74xx	The system is requesting to swap the power role	(PD 2.0)
	ANX74xx to system	The system is being requested to swap power role and ANX74xx OCM FW has accepted it.	(PD 2.0)
Data role swap request <i>TYPE_DSWAP_REQ (0x11)</i>	System to ANX74xx	The system is requesting to swap the data role	(PD 2.0)
	ANX74xx to system	The system is being requested to swap data role and ANX74xx OCM FW has accepted it.	(PD 2.0)
GoTo_Min request <i>TYPE_GOTO_MIN_REQ (0x12)</i>	System to ANX74xx	The system is requesting to go back to vSafe5V operating mode	(PD 2.0)
	ANX74xx to system	The system is being requested to go back to vSafe5V operating mode	(PD 2.0)
VCONN swap request <i>TYPE_VCONN_SWAP_REQ (0x13)</i>	System to ANX74xx	The system is requesting to swap the VCONN	(PD 2.0)
	ANX74xx to system	The system is being requested to swap VCONN	(PD 2.0)
VDM <i>TYPE_VDM (0x14)</i>	System to ANX74xx	The system is sending the unstructured VDM.	(PD 2.0)
	ANX74xx to system	The system is getting the unstructured VDM.	(PD 2.0)
DP pin assignment <i>TYPE_DP_SNK_CFG (0x15)</i>	System to ANX74xx	The system is sending its DP sink's configuration.	(PD 2.0)
	ANX74xx to system	The system is getting the DP sink's configuration.	(PD 2.0)
Power object position request <i>TYPE_PWR_OBJ_REQ (0x16)</i>	System to ANX74xx	When as a power sink, system is requesting an expected position of power source's capability objects from the connected device.	(PD 2.0)
	ANX74xx to system	When as a power source, system is being requested a desired position of its power capability objects	(PD 2.0)
<i>TYPE_GET_CC_STATUS (0x17)</i>	System to ANX74xx	The system wants to get current cc status. When ANX74xx receives it, it is sent to system from ANX74xx.	(PD 2.0)
<i>TYPE_DP_ALT_ENTER (0x19)</i>	ANX74xx to system	Notifies system that ANX74xx entered DP Alt Mode.	(PD 2.0)
<i>TYPE_DP_ALT_EXIT (0x1A)</i>	ANX74xx to system	Notifies system that ANX74xx exited DP Alt Mode.	(PD 2.0)
<i>TYPE_GET_SINK_CAP (0x1B)</i>	System to ANX74xx	The system wants to get sink capability. When ANX74xx receives it, it is sent to system from ANX74xx.	(PD 2.0)
<i>TYPE_NOT_SUPPORTED (0x1C)</i>	System to ANX74xx	The ANX74xx informs last message is not support.	(PD 3.0)
<i>TYPE_GET_SRC_CAP_EXT (0x1D)</i>	System to ANX74xx	The system is requesting get Extended Source Capabilities.	(PD 3.0)
<i>TYPE_GET_STS (0x1E)</i>	System to ANX74xx	The system is requesting get source status.	(PD 3.0)
<i>TYPE_FR_SWAP (0x1F)</i>	System to ANX74xx	The system is requesting to fast swap.	(PD 3.0)

Message Type	From/To	Description	Notes
TYPE_FR_SWAP_SIGNAL (0x20)	System to ANX74xx	The system informs ANX74xx to trigger fast swap signal.	(PD 3.0)
TYPE_GET_PPS_STS (0x21)	System to ANX74xx	The system is requesting get PPS status.	(PD 3.0)
TYPE_GET_COUNTRY_CODE (0x22)	System to ANX74xx	The system is requesting get Country code	(PD 3.0)
TYPE_GET_SINK_CAP_EXT (0x23)	System to ANX74xx	The system is requesting get extended sink capability.	(PD 3.0)
TYPE_BATT_STS (0x24)	System to ANX74xx	ANX74xx is requesting get battery status.	(PD 3.0)
TYPE_ALERT (0x25)	System to ANX74xx	The system is sending Alert message to ANX74xx.	(PD 3.0)
TYPE_GET_COUNTRY_INFO (0x26)	System to ANX74xx	ANX74xx is requesting get country information.	(PD 3.0)
TYPE_GET_SNK_IDENTITY (0x28)	System to ANX74xx	ANX74xx is requesting get sink identity ACK.	(PD 2.0)
TYPE_GET_DP_CONFIGURE (0x29)	System to ANX74xx	ANX74xx is requesting get DP configuration.	(PD 2.0)
TYPE_DP_CONFIGURE (0x2a)	System to ANX74xx	The system is sending DP configuration to ANX74xx.	(PD 2.0)
TYPE_AMS_PROCESS (0x2b)	System to ANX74xx	The system is Processing AMS (SinkTxNG/SinkTxOK function)	(PD 3.0)
TYPE_PPS_REQUEST (0x2d)	System to ANX74xx	The system is requesting PPS voltage & current.	(PD 3.0)
TYPE_EXT_SRC_CAP_EXT (0x30)	ANX74xx to system	ANX74xx is sending extended source capability.	(PD 3.0)
TYPE_EXT_STS (0x31)	ANX74xx to system	ANX74xx is sending extended Status.	(PD 3.0)
TYPE_EXT_GET_BATT_CAP (0x32)	System to ANX74xx	The system is requesting get extended battery capabilities.	(PD 3.0)
TYPE_EXT_GET_BATT_STS (0x33)	System to ANX74xx	The system is requesting get extended battery status.	(PD 3.0)
TYPE_EXT_BATT_CAP (0x34)	ANX74xx to system	ANX74xx is sending extended Battery capability.	(PD 3.0)
TYPE_EXT_GET_MFR_INFO (0x35)	System to ANX74xx	The system is requesting get manufacture information.	(PD 3.0)
TYPE_EXT_MFR_INFO (0x36)	ANX74xx to system	ANX74xx is sending manufacture information.	(PD 3.0)
BTYPE_EXT_PPS_STS (0x37)	ANX74xx to system	ANX74xx is sending PPS status.	(PD 3.0)
TYPE_EXT_COUNTRY_INFO (0x38)	ANX74xx to system	ANX74xx is sending country information.	(PD 3.0)
TYPE_EXT_COUNTRY_CODE (0x39)	ANX74xx to system	ANX74xx is sending country code.	(PD 3.0)
TYPE_EXT_SINK_CAP_EXT (0x3A)	ANX74xx to system	ANX74xx is sending extended sink capability.	(PD 3.0)
TYPE_EXT_PDFU_REQUEST (0x3B)	System to ANX74xx	The system is sending pd firmware update request.	(PD 3.0)
TYPE_EXT_PDFU_RESPONSE (0x3C)	ANX74xx to system	ANX74xx is sending pd firmware update response.	(PD 3.0)
TYPE_EXT_WriteReg (0x3D)	System to ANX74xx	The system is sending write request for ANX74xx related registers.	

Message Type	From/To	Description	Notes
TYPE_EXT_I2C_Write (0x3E)	System to ANX74xx	The system is sending I2C write request for ANX74xx master I2C interface to access I2C devices.	
Response to a prior request TYPE_RESPONSE_TO_REQ (0xF0)	ANX74xx to system	The system checks if its request is accepted or rejected.	(PD 2.0)
Soft reset TYPE_SOFT_RST (0xF1)	ANX74xx to system	Notifies system that a soft reset is being triggered.	(PD 2.0)
Hard reset TYPE_HARD_RST (0xF2)	ANX74xx to system	Notifies system that a hard reset is being triggered.	(PD 2.0)
TYPE_GET_VAR (0xFC)	System to ANX74xx	The system is requesting variable from ANX74xx	
TYPE_SET_VAR (0xFD)	System to ANX74xx	The system is writing variable to ANX74xx.	
TYPE_BILLBOARD_VENDOR_STR (0x40)	Customer-defined	Stored in customer-defined zone for Billboard function.	
TYPE_BILLBOARD_PRODUCT_STR (0x41)	Customer-defined	Stored in customer-defined zone for Billboard function.	
TYPE_BILLBOARD_SERIAL_STR (0x42)	Customer-defined	Stored in customer-defined zone for Billboard function.	

4.8 Customer VDM Mode

To enter customer VDM mode, the AP must initialize SVID as a customer SVID by interface command TYPE_SVID (0x03); the OCM firmware will discover and enter customer VDM mode if the downstream supports this customer SVID on Ack to Discover SVIDs. After entering customer VDM mode, the OCM transfers customer specific VDM by interface command TYPE_VDM (0x14). Figure 4-3 shows the customer VDM transition flow.

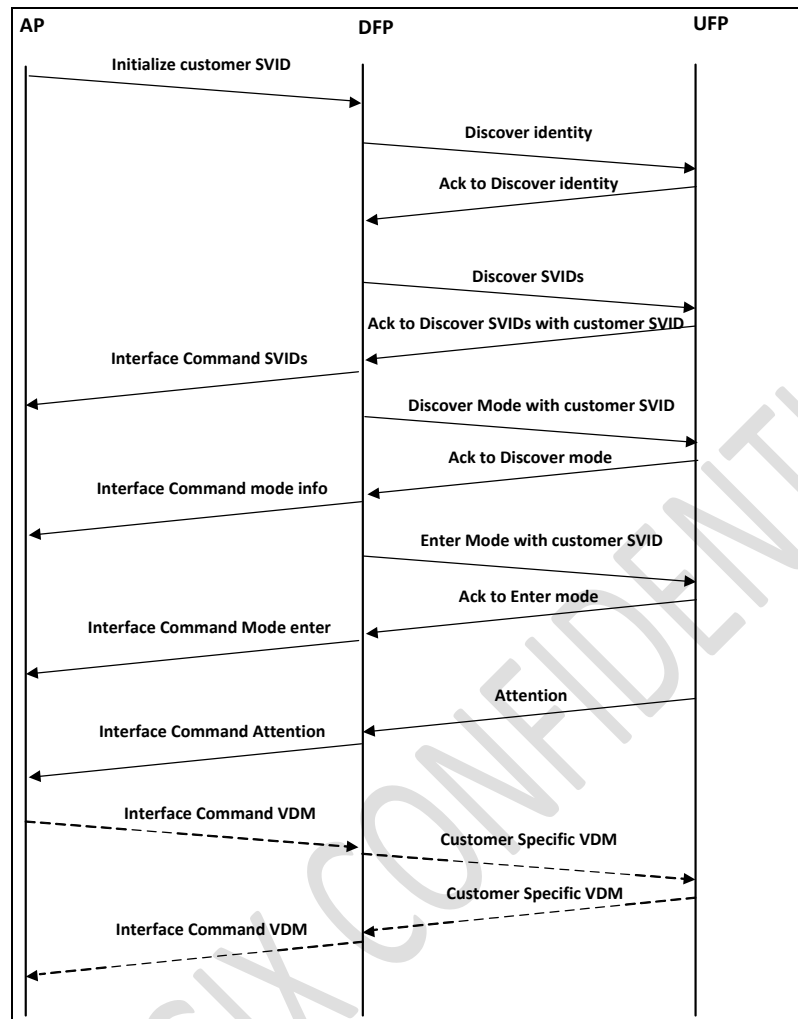


Figure 4-3 Customer VDM Flow

4.9 PPS Function on Sink

ANX74xx only supports the PPS function on the sink side. The AP must enable the PPS function by setting PPS_FUNC_EN (0x58:f1 bit2) and updating request voltage and current by interface command TYPE_PPS_REQUEST (0x2d). Figure 4-4 shows the PPS control flow.

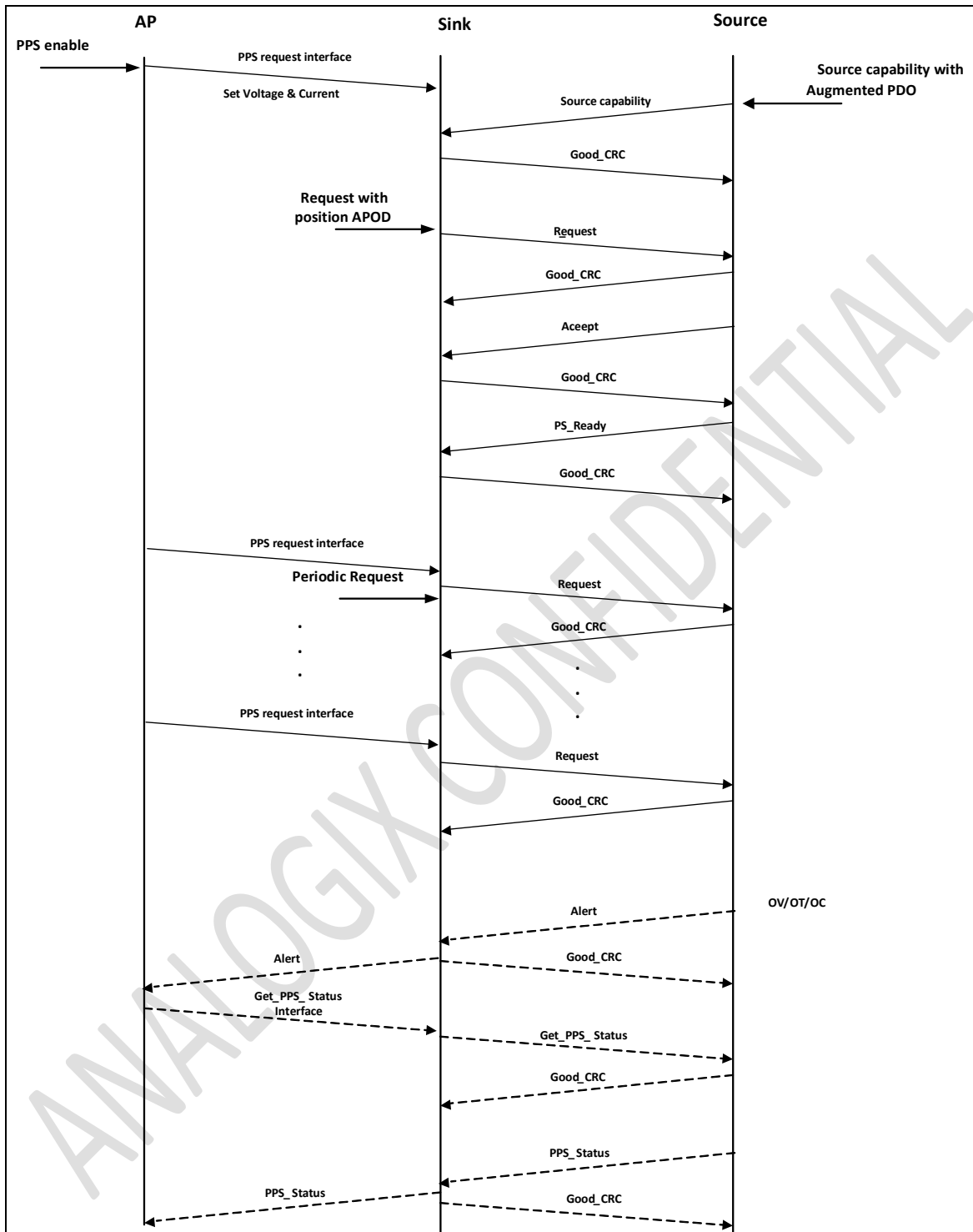


Figure 4-4 PPS Control Flow

5 Automatic Power Negotiation

Per the USB-PD Spec, power negotiation has the following requirements on timing.

- After the Source_Capabilities Message is successfully sent, the power provider triggers SenderResponseTimer (24ms to 30ms) to wait for the Request message.
- After the Request message is successfully sent, the power consumer triggers SenderResponseTimer (24ms to 30ms) to wait for the Accept message.

The ANX74xx provides a method to handle the power negotiation flow through the ANX74xx firmware. If this method is adopted, the ANX74xx driver on the AP/EC is used only to initialize certain settings so that the ANX74xx firmware controls PD and DisplayPort Alternate mode messages. There is no interface message communication as described in section 8. The following section describes this method in greater detail.

Note the following when adopting this method:

- As a power provider, the USB Type-C device must only support 5V power supply.
- As a power consumer, the USB Type-C device must follow the mechanism described below.

5.1 Accepting/Rejecting the Request Data Object

When ANX74xx firmware enters power source mode and receives a Request Data Object (power consumer sent), it checks the Request Data Object with source capabilities (only supports 5V) and then responds with an ACCEPT/REJECT message.

Note: Products with ANX74xx only support 5V power supply if the automatic Power Negotiation method is adopted.

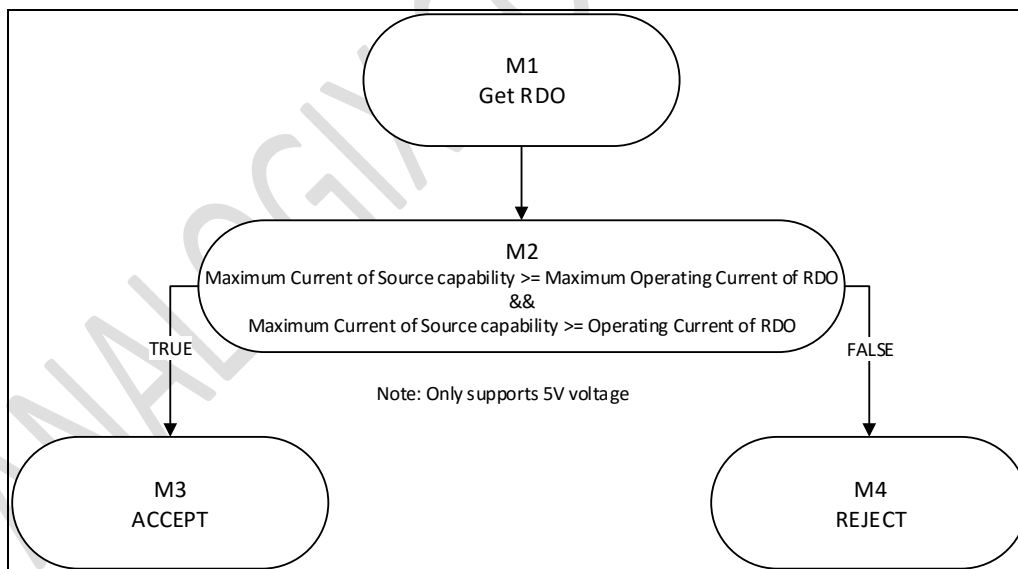


Figure 5-1 Accepting/Rejecting the Request Data Object

5.2 Responding to the Source Capabilities

When ANX74xx enters power consumer mode, the Request Data Object only supports a fixed and variable Request Data Object, as shown in Table 5-1. It does not support a battery Request Data Object.

Table 5-1 Fixed and Variable Request Data Object

Bit	Description
B31	Reserved. Set to zero.
B30..28	Object position (000b is reserved and is not used.)
B27	GiveBack flag = 0
B26	Capability mismatch
B25	USB communication capable
B24	No USB suspend
B23	Unchucked extend messages supported
B22..20	Reserved. Set to zero.
B19..10	Operating current in 10mA units
B9..0	Maximum operating current (in 10mA)

- Object Position

The AP/EC first initializes the maximum voltage and maximum/minimum power before power negotiation. Then, according to the source capabilities the power provider sends, the ANX74xx firmware can determine the object position chosen, as shown in Figure 5-2.

The following give the workflow for choosing the object position.

1. (M1) Initialization. AP/EC initializes maximum voltage (init_maximum_voltage), maximum power (init_maximum_power), and minimum power (init_minimum_voltage) before power negotiation.
2. (M2) Traverse the PDOs of source capabilities. After ANX74xx receives the source capabilities (PDOs), it traverses the PDOs.
3. (M3) Get the PDO status. That is, the maximum power and voltage of the current PDO.
4. (M4) Compare voltages. When the maximum voltage of a PDO is larger than init_maximum_voltage, ANX74xx firmware checks the next PDO; otherwise, go to M5.
5. (M5) Check power. If the previous power (stored value) is less than the current power and init_max_power or, the previous voltage (stored value) equals the current power or init_max_power, and the current voltage is more than the prior voltage, go to M6; otherwise, go to M2.
6. (M6) Store the PDO status. Store the current PDO index, voltage, and power.
7. (M7) Power mismatch. Compare power with the current PDO.
8. (M8) Object position. Maximum power of source capability provided is less than Minimum_Power.
9. (M9) Enable goto_safe5v through register (0x6E:bit4).
10. (M10) Object position =1: Choose Safe5V.

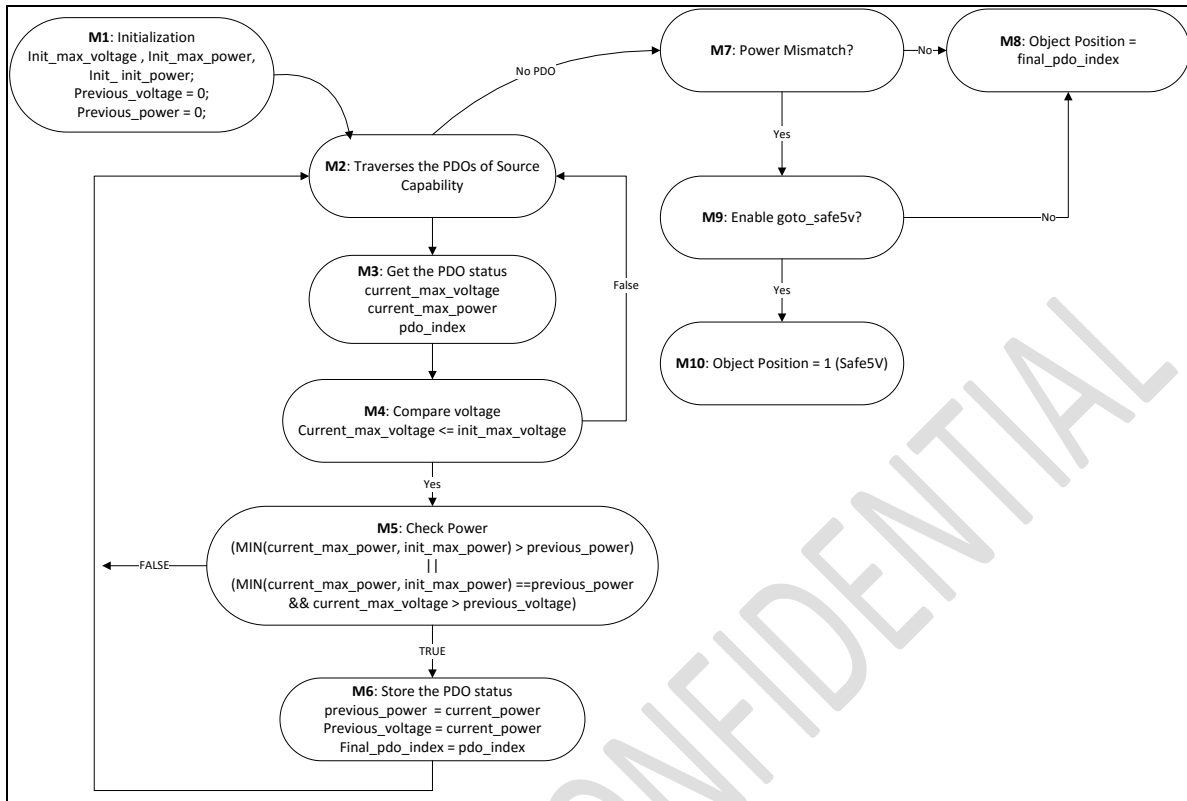


Figure 5-2 Selecting an Object Position

Note: $MIN(x, y)$, compares x and y and returns the smaller one.
Current is less than 3A.

The following is an example.

Initialization: maxW = 20W and maxV = 20V.

- Source capability: 5V@3A, 12V@1A, 20V@1A, Object Position = 3 (20V@1A)
- Source capability: 5V@3A, 12V@3A, Object Position = 2 (12@3A)
- Source capability: 5V@3A, 15V@3A, 16V@1A, 20V@0.5A, Object Position = 2 (15V@3A)
- GiveBack Flag

Can be configured by API.

- USB Communication Capable

Can be configured by API.

- No USB Suspend

Can be configured by API.

- Operating Current (in 10mA)

As shown in Figure 5-3, the X-axis means voltage and the Y-axis means current. When ANX74xx selects a voltage based on the source capability of the power provider, ANX74xx gets a definite value, which is determined by the following formula. Operating current (in 10mA) can also be determined by comparing this value and the source capability.

$$Y = \begin{cases} \max W \frac{1}{X} & (5 \leq X \leq \max V) \\ 0 & (\text{Others}) \end{cases}$$

maxW: Maximum Power

maxV: Maximum Voltage

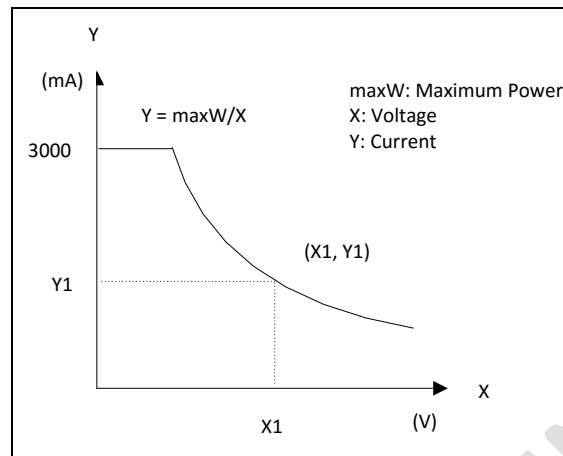


Figure 5-3 Calculating RDO Operating Current

The following is an example.

Initialization: maxW = 20W and maxV = 20V, and maximum current of source capability = 1.8A

- When $X1 = 10V$, Operating current = $\text{MIN}(Y1/10, 1.8) = \text{MIN}(20/10, 1.8) = 1.8A$
- When $X1 = 15V$, Operating current = $\text{MIN}(Y1/15, 1.8) = \text{MIN}(20/15, 1.8) = 1.33A$
- Maximum Operating Current (in 10mA)

Maximum operating current (in 10mA) equals operating current (in 10mA).

6 UCSI Passthrough Mode Implementation

6.1 Overview

A PPM (Platform Policy Manager) is implemented in ANX74xx firmware based on UCSI (USB Type-C Connector System Software Interface) Revision 1.1. The memory of UCSI data structure is 48 bytes in size and is in the I2C slave device address space ranging from 0x7E:0xC0 to 0x7E: 0xEF. PPM will generate an interrupt on INTP pin when asynchronous event occurs to notify OPM (OS Policy Manager). On finishing carrying out the required operation, OPM clears the interrupt request writing the corresponding register. OPM also sends commands to PPM to notify PPM that a command has been issued; an auxiliary (AUX) register is defined. OPM writes a non-zero value to this register to indicate that a command is ready to be processed. PPM clears this register before processing it.

Note that due to the restriction of specification or firmware limitation, not all UCSI commands are implemented or fully implemented.

Refer to the UCSI Spec at: <http://www.intel.com/content/www/us/en/io/universal-serial-bus/usb-type-c-ucsi-spec.html>.

6.2 Sending a Command

The workflow for sending a command:

1. OPM writes the content of the command into the CONTROL register.
2. OPM writes 0x01 to AUX register (0x7E:0xF0.)
3. PPM periodically polls the status of AUX register and clears it when it gets a non-zero value.
4. PPM executes the command.

6.3 Sending an Interrupt

The workflow for sending an interrupt:

1. PPM writes CCI register.
2. PPM generates a low pulse on INTP pin.
3. On reception of an interrupt, OPM clears 0x7E:0xCB bit 7.
4. OPM reads CCI register to determine the cause for the interrupt.

6.4 Command List

Table 6-1 Commands

Command	Value	Comments
Reserved	0x00	Support
PPM_RESET	0x01	Support

Command	Value	Comments
CANCEL	0x02	Not support
CONNECTOR_RESET	0x03	Not support disconnect-connect sequence
ACK_CC_CI	0x04	Support
SET_NOTIFICATION_ENABLE	0x05	Support
GET_CAPABILITY	0x06	Support
GET_CONNECTOR_CAPABILITY	0x07	Support
SET_UOM	0x08	Not support
SET_UOR	0x09	Support
SET_PDM (obsolete)	0x0A	Not support
SET_PDR	0x0B	Support
GET_ALTERNATE_MODES	0x0C	Not support when Recipient is SOP' or SOP''
GET_CAM_SUPPORTED	0x0D	Only support DP_SID
GET_CURRENT_CAM	0x0E	Support
SET_NEW_CAM	0x0F	Support
GET_PDOS	0x10	Support
GET_CABLE_PROPERTY	0x11	Support
GET_CONNECTOR_STATUS	0x12	Not support Battery Charging Capability Status and Provider Capabilities Limited Reason
GET_ERROR_STATUS	0x13	Support
SET_CCOM		Not support (UCSI spec V1.1 does not define Message type)

6.5 PPM Initialization

PPM initialization workflow:

1. Send a PPM_RESET (optional).
2. Enable the "Command Completed" notification by sending SET_NOTIFICATION_ENABLE command.
3. Determine platform capability by sending GET_CAPABILITY command.
4. Enable as many of the notifications supported by the PPM as needed by sending a SET_NOTIFICATION_ENABLE command.
5. OPM sends GET_CONNECTOR_CAPABILITY command to determine the capabilities of the connector.
6. OPM sends GET_ALTERNATE_MODES on the connector to determine the Alternate Modes supported by the PPM. Then, the OPM sends GET_CAM_SUPPORTED to determine the Alternate Modes supported by the connector.

7. If there is a device connected (indicated by the Connector Change Indicator) on the connector, the OPM sends GET_ALTERNATE_MODES (Recipient field equal to SOP) to determine the Alternate Modes supported by the connected device.
8. The OPM can use the GET_CURRENT_CAM command to determine the current Alternate Mode that the connector is operating in.

ANALOGIX CONFIDENTIAL

7 Code Examples

7.1 Power Status/Cable Status/Messages Received Interrupt Handling

```
void get_tpc_alert(int port)
{
    /***/

    change_int = lbt_rdtype(port, RX_P0, CHANGE_INT);
    change_status = lbt_rdtype(port, RX_P0, SYSTEM_STSTATUS);
    intr_alert_0 = lbt_rdtype(port, TCPC_INTERFACE, INTR_ALERT_0);
    intr_alert_1 = lbt_rdtype(port, TCPC_INTERFACE, INTR_ALERT_1);
    /*Clear Interrupt Status*/
    lbt_wrbyte(port, RX_P0, CHANGE_INT, 0x00);
    lbt_wrbyte(port, TCPC_INTERFACE, INTR_ALERT_0, 0xFF);
    lbt_wrbyte(port, TCPC_INTERFACE, INTR_ALERT_1, 0xFF);

    /* Power ON/OFF Process */
    if(intr_alert_1 & INTR_INTP_POW_ON) {
        /*ANX74xx Power ON*/
        usb_pd_function_init(port);
        pd_policy[port].power_on= 1;
    }
    if (intr_alert_1 & INTR_INTP_POW_OFF) {
        /*ANX74xx Power ON*/
        chip_power_down(port);
        pd_policy[port].power_on= 0;
        return;
    }
}
```

```
/*Received software interrupt*/

if (intr_alert_0 & INTR_SOFTWARE_INT) {

    if (change_int & VBUS_CHANGE) {

        if (change_status & VBUS_CHANGE)
        {

            /*ANX74xx SOURCE_CTRL set to high*/

        }

        else

        {

            /*ANX74xx SINK_CTRL set to high*/

        }

    }

    if (change_int & CC_STATUS_CHANGE) {

        /*ANX74xx CC1/CC2 Status Change Handling*/

    }

}

/*Received interface message*/

if (intr_alert_0 & INTR_RECEIVED_MSG) {

/*AP/EC got message from ANX74xx and Handling*/

    interface_recvd_msg_handler(port);

}

}
```


8 Appendix A: PD Messaging Interface Code

This section describes the low-level AP PD Message Interface code.

8.1 Message Transaction Method

The communication between the AP interface code and ANX74xx OCM firmware is based on the Request-Response or Request-Reply protocol. The requestor sends a request message to the replier, which receives and processes the request and returns a message in response. The AP interface code and/or ANX74xx OCM firmware can be either the requestor or the replier. Note that they cannot be the requestor at the same time.

8.1.1 Communication between AP/EC and ANX74xx OCM Firmware

When the AP interface code (external MCU/EC) needs to communicate with the ANX74xx OCM FW, it must follow the sample sequence shown in Figure 8-1, for example, when the AP Interface Code sends source capabilities to ANX74xx OCM FW.

8.1.1.1 Successful Communication from AP Interface Code to ANX74xx FW

External MCU MUST writes new command, data and CRC to RX buffer registers firstly, then write data length to first register of RX buffer; OCM firmware will poll length register and read RX buffer on value of length register greater than zero.

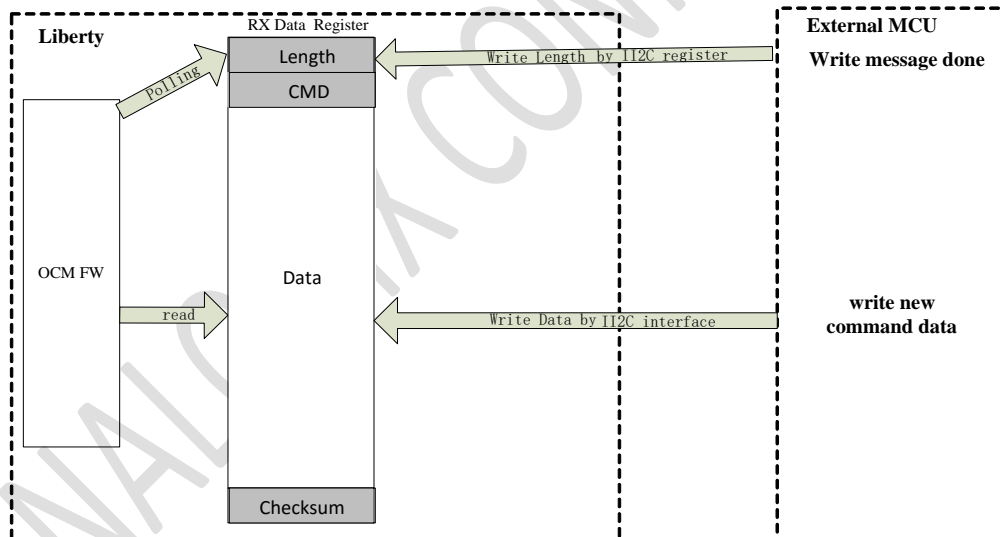


Figure 8-1 Communication between AP Interface Code and ANX74xx Firmware

8.1.1.2 Successful Communication from ANX74xx FW to AP Interface Code

OCM firmware MUST writes length, command, data and CRC to TX buffer registers firstly and then write register to trigger interrupt pin to inform external MCU, which in turn reads data on detect Alert pin assert.

To avoid TX data register overridden by external MCU (AP/EC), do not respond Alert request in time. ANX74xx transfers message data to buffer when last TX data is not read out and then transfers them after TX register is empty.

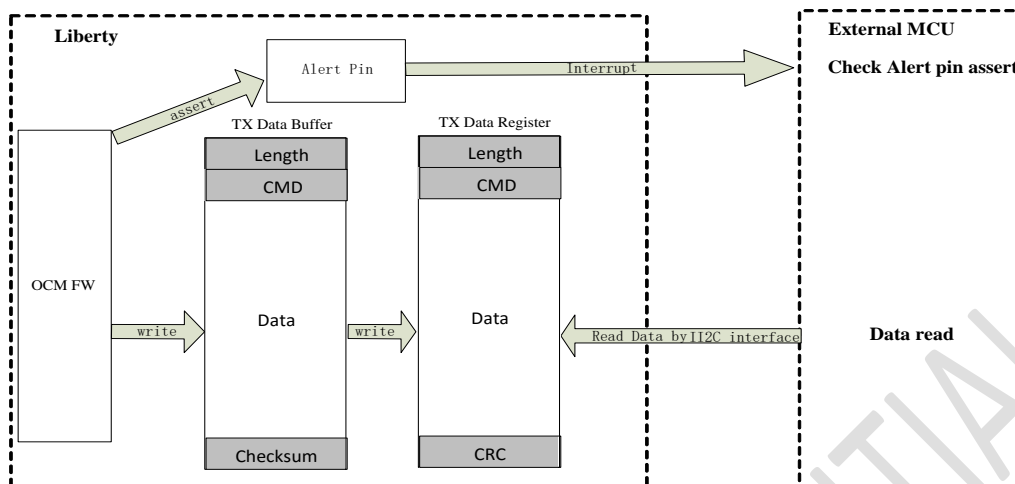


Figure 8-2 Communication between ANX74xx Firmware and AP Interface Code

8.1.1.3 Error Handling

- Checksum error

Before a message can be received from the sender, the checksum must be checked. If a checksum error occurs, the receiver must ACK an error status and the sender must start the retry method.

- ACK error

After the sender (AP interface code or ANX74xx OCM FW) sends a message and a no-ACK message or a failure ACK message is received, the sender starts the retry method.

9 Appendix B: Specific Applications from Customer

9.1 Not Handle Interrupt Source

9.1.1 Application Overview

In some applications, AP/EC cannot handle Alter pin interrupt, AP/EC can only use polling registers mode to trigger charger event.

9.1.2 Work Flow (Recommended)

9.1.2.1 AP/EC Can Read Alter GPIO Value

1. AP/EC reads Alter pin GPIO value;
2. If GPIO value is low (low active), go to Step 3, otherwise, go to Step 1;
3. AP/EC calls `get_tpc_alert()` to handle ANX7447 event;
4. Go to Step 1;

9.1.2.2 AP/EC Can Only Use I2C Polling Registers

1. AP/EC always polling Firmware version registers. If I2C operation is success, goto Step 2;
2. Step 3 AP/EC always call `get_tpc_alert()` to handle ANX7447 event.
3. If AP/EC get Power Down event, Goto Step 1, otherwise go to Step 2.

10 Appendix C: Firmware Update (PDFU)

10.1 Power Delivery Firmware Update

The ANX74xx OCM firmware support USB Power Delivery Firmware Update (PDFU).

Refer to the following documents for related information.

<https://www.usb.org/document-library/usb-power-delivery-firmware-update-specification>

AnaVis (Analogix Debug Tool) support PDFU function for firmware update.

It need a PC/NB with AnaVis software, Type-C cable and two ANX74xx board (one as Initiator, another as Responder).

Refer to the following documents for related information.

AA-005990-SW AnaVis (Analogix Debug Tool) (V1.0.5)

AA-005402-SW ANX7447/7411/7412/7327/3411-AC OCM Firmware HEX (V1.5)

AA-005502-SW Liberty External MCU Souce Code (V1.4)

10.1.1 Port Controller Application

The ANX74xx port controller (Source/Sink, DFP/UFP) application OCM firmware support PDFU as Initiator or Responder.

10.1.1.1 Use AnaVis update port application firmware

The reference environment setup as below.

Connect platform:

PC/NB with AnaVis + ANX74xx EVB (Initiator) + Type-C cable + Target ANX74xx board (Responder).

Steps:

1. Connect COM port with ANX74xx EVB
AnaVis Settings->Port Configuration
Select Port and open it.
2. Set SOP type (default is 0, if not changed, ignore this step)
\sop 0 ; select 0 (SOP) for Port Partner
Please send the command in AnaVis Message box.
3. PDFU
AnaVis Command->PDFU
Select file then send it.

Will see "validate success!" print message in AnaVis (if fail, please retry or adjust delay for 256 bytes and retry)

10.1.2 Cable Application

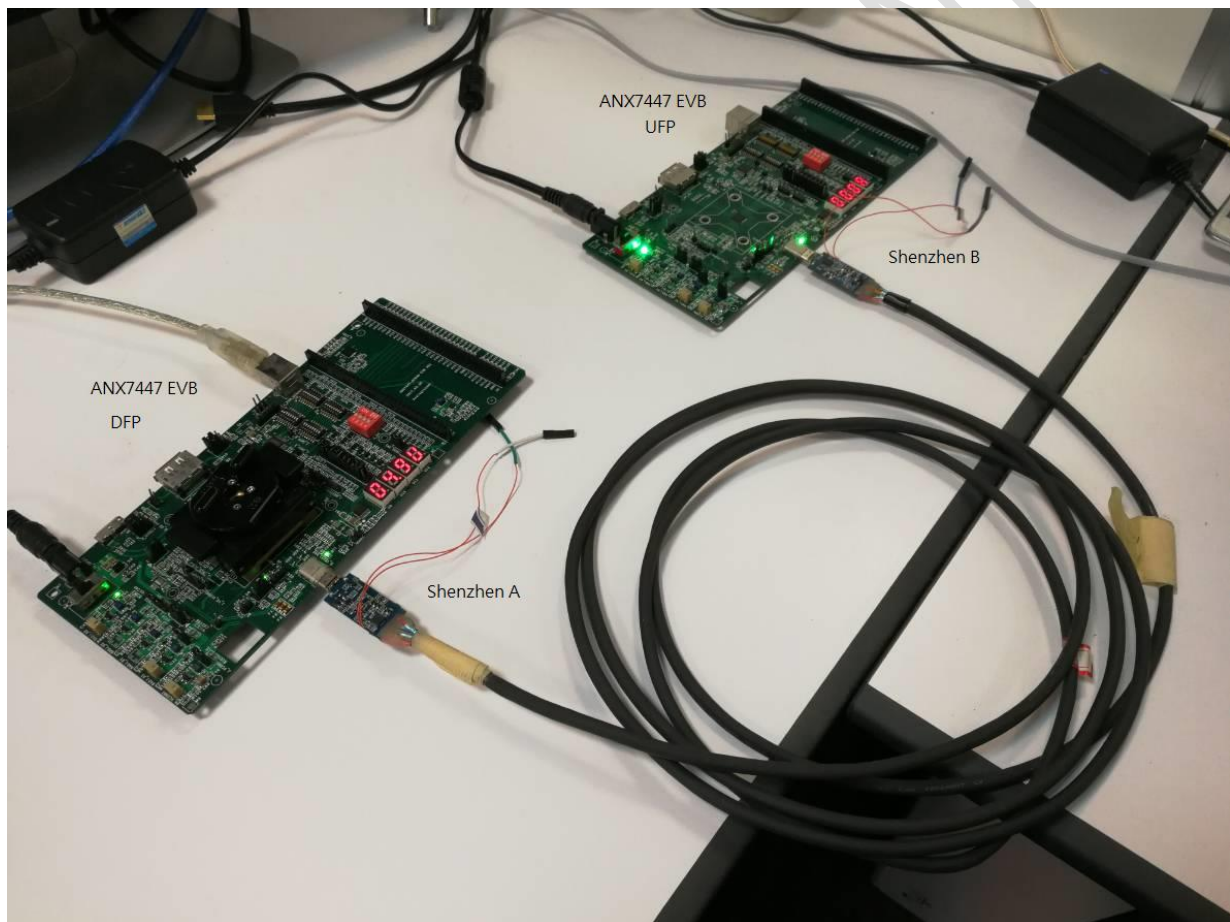
The ANX74xx cable application OCM firmware support PDFU as Responder.

10.1.2.1 Use AnaVis update cable application firmware

The reference environment setup as below.

Connect platform:

PC/NB with AnaVis + ANX7447 EVB (DFP) + Shenzhen A – Shenzhen B (Cable) + ANX7447EVBUFP)



Shenzhen A is cable source side with ANX7410 cable application.

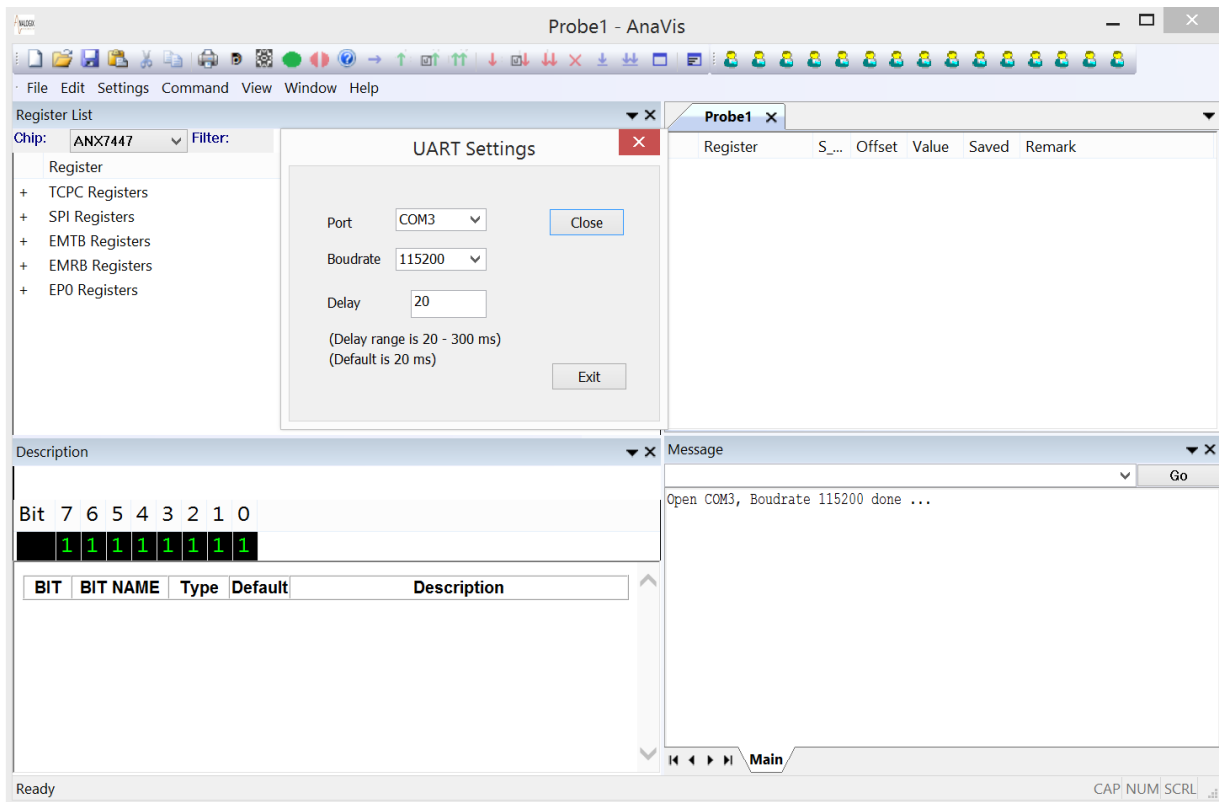
Shenzhen B is cable sink side with ANX7410 cable application.

Update Shenzhen A then Shenzhen B steps as below.

1. Connect COM port with ANX7447 EVB (DFP)

AnaVis Settings->Port Configuration

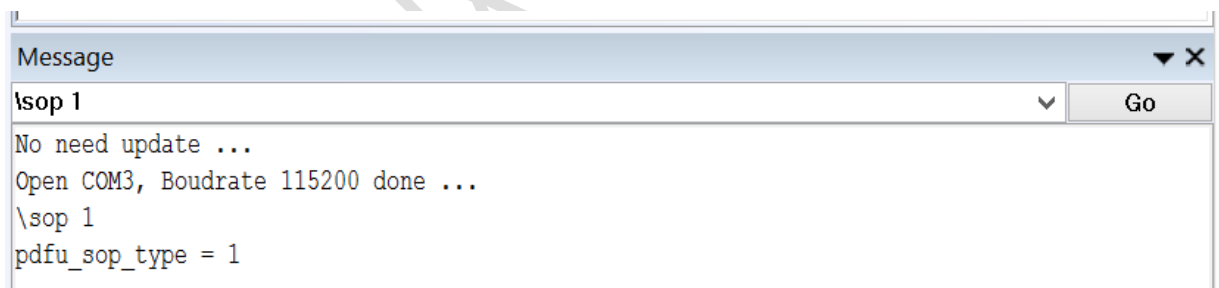
Select Port and open it.



2. Set SOP type

\sop 1 ; select 1 (SOP') for Shenzhen A

Please send the command in AnaVis Message box.

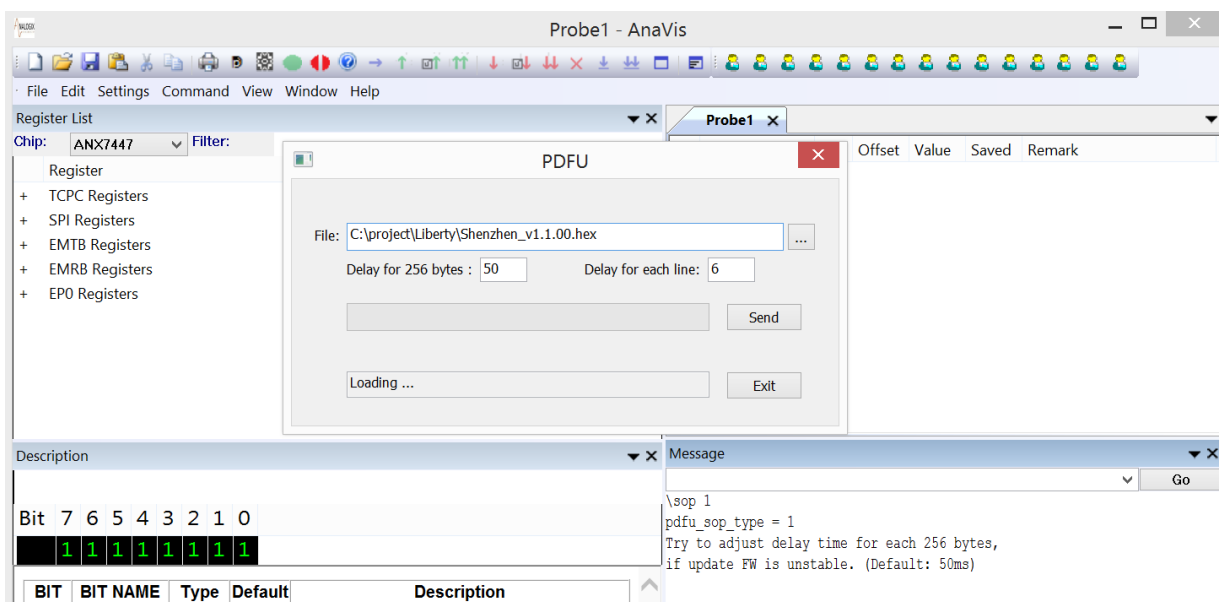


Note: SOP' is near Vconn source side (DFP), SOP" is another side (UFP).

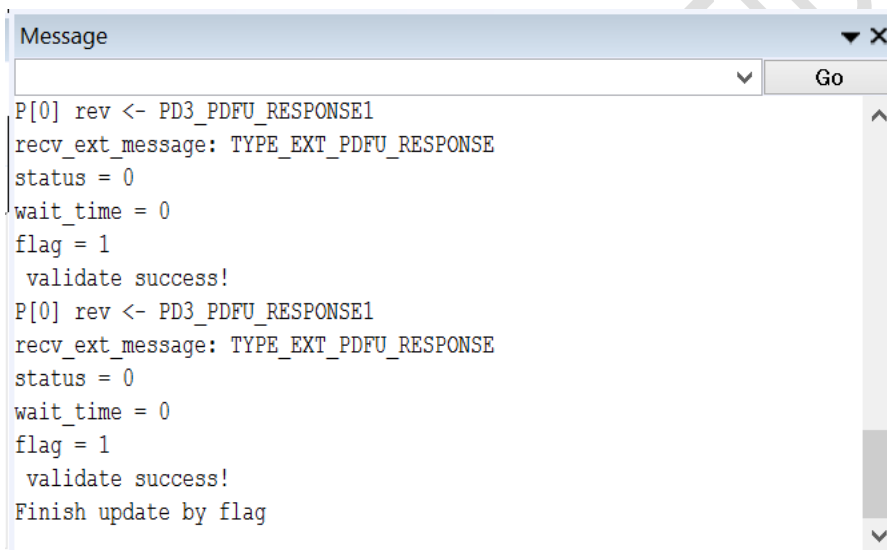
3. PDFU

AnaVis Command->PDFU

Select file then send it.



Will see "validate success!" print message in AnaVis (if fail, please retry or adjust delay for 256 bytes and retry)



4. Set SOP type

\sop 2 ; select 2 (SOP") for Shenzhen B

5. PDFU

AnaVis Command->PDFU

Select file then send it.

(Will see "validate success!" print message in AnaVis)

6. Check FW version (just for verify, not must do it)

Unplug Shenzhen cable and plug it again.

\cbl_rd 1 7e b4 ; select SOP', check for Shenzhen A FW version (if fail, please retry once)

\cbl_rd 1 7e b5 ; select SOP', check for Shenzhen A FW build number

\cbl_rd 2 7e b4 ; select SOP", check for Shenzhen B FW version (if fail, please retry once)

\cbl_rd 2 7e b5 ; select SOP", check for Shenzhen B FW build number

11 Revision History

Table 11-1 Document Revision History

Revision	Date	Changes
Release 1	Aug 2017	Initial version
Release2	Aug 2017	Added Table3-19: Firmware Version (I2C Address: 0x7E; Offset: 0xB4 and 0xB5)
Release2	Oct 2017	Added UCSI description Added subsection PPM Initialization Clear the document format
Release 3	Apr 2018	Updated Section 1: Introduction Added Section 3.3: Firmware Integrity Checking Added Section 3.4: Customer-defined Zone Added Section 3.5: Erasing Flash Memory Updated Table 4-19: Messages Exchanged between AP System PD Policy Code and ANX74xx OCM Firmware Added Section 6: UCSI Passthrough Mode Implementation Added Appendix B: Specific Applications from Customer
Release 4	Aug 2018	Tables 4-1, 4-4, 4-5, 4-6, 4-7, 4-8, 4-9, 4-10, 4-11, 4-14, 4-17, 4-18: Updated title Added Tables 4-2 (Firmware Control 1), 4-3 (Firmware Control 2) Table 4-21: Updated TYPE_DP_CONFIGURE, TYPE_PPS_REQUEST; added TYPE_AMS_PROCESS Added sections 4.8 Customer VDM Mode, 4.9 PPS Function on Sink
Release 5	Nov 2018	Added Appendix C: Firmware Update (PDFU)

Copyright ©2018 Analogix Semiconductor, Inc.
3211 Scott Blvd., Suite 100
Santa Clara, CA 95054, USA

+1 (408) 988-8848

<http://www.analogix.com/>

©2018 Analogix Semiconductor, Inc. All Rights reserved.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY EXPRESS REPRESENTATIONS OF WARRANTIES. IN ADDITION, ANALOGIX SEMICONDUCTOR INC. DISCLAIMS ALL IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS. SPECIFICATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

This document contains proprietary information of Analogix Semiconductor, Inc. or under license from third parties. No part of this document may be reproduced in any form or by any means or transferred to any third party without the prior written consent of Analogix Semiconductor, Inc.

The information contained in this document is not designed or intended for use in on-line control of aircraft, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Analogix disclaims any express or implied warranty of fitness for such uses.

Analogix Semiconductor, Inc., the Analogix Logo, and WideEye™ SerDes, CoolHD™, and SlimPort® are trademarks of Analogix Semiconductor, Inc., in the United States and other countries.

HDMI, the HDMI logo and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

DisplayPort and the DisplayPort logo are trademarks or registered trademarks of the Video Electronics Standards Association, VESA®.

USB and the USB logo are trademarks or registered trademarks of USB Implementers Forum, Inc., creators of USB technology.

All other trademarks and registered trademarks are the property of their respective owners.