MAX77958
Customization Script and OPCode Command Guide

Rev 0; 7/20

Abstract

The MAX77958 Customization Script and OPCode Command Guide explains how system designers can use the MAX77958 and describes the control registers that can be configured by OPCode commands.

Table of Contents

Standard Firmware

The standard firmware controls all USB Type-C and power delivery related functions. This part of the firmware is expected to be familiar to for all system designers as it follows the USB Type-C and PD 3.0 specification.

Another portion of the MTP is reserved for the customization scripts; system designers can program this area to make the MAX77958 satisfy the system definition.

If the standard firmware is updated, the customization script will be automatically removed. When updating the standard firmware and customization script, standard firmware needs to be updated first.

The MAX77958 can be customized to operate with specific applications using customization scripts and OPCode commands.

Customization scripts are recommended for autonomous systems without an application processor and OPCode commands are recommended for systems with an application processor.

Detailed descriptions of each method are illustrated in the following sections.

Customization Script

The customization script is the core of the configurability of the MAX77958 when used in the autonomous configuration. It allows the user to configure either a GPIO state or perform an I²C action when an event is detected on the USB Type-C interface. The customization script is written in the graphical user interface (GUI). The software translates the customization script to hexadecimal format and writes it to the IC configuration area. The configuration by GUI must be stored in the MTP area of the IC to start operating as per the customization commands.

Figure 1 shows the simple implementation of the customization script. Developers can define the functions and the sequences for each Event Id listed on the Event Item List using the MAX77958 GUI. The GUI provides users not only the interface to create the customized code but also the functionality to detect syntax errors in the script usage.

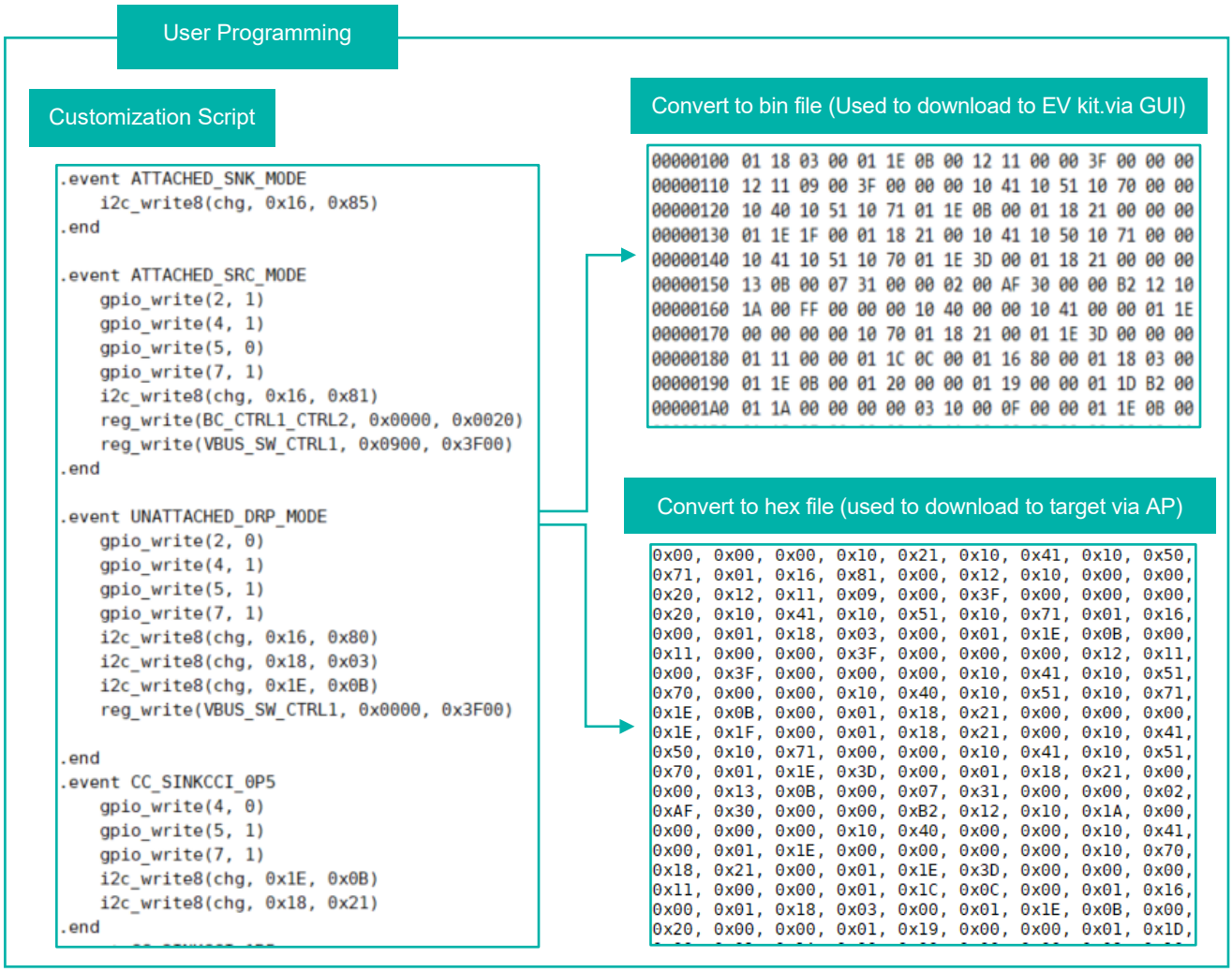

Figure 1. User Programming of Customization Script.

Customization Commands

Table 1 lists the syntax operation of customization commands recognized by the GUI.

Table 1. Commands for peripheral device

TYPE	COMMAND	DESCRIPTION
Comment	;	Any character after ";" is ignored
Directives	.event [ID] or .event [name]	The directive is to indicate the start point of action. See the Event Item List regarding the number/name of Event. ID: Event ID, name: Name of Event List
	.end	The directive is to indicate the end point of action. One Event should be started ".event_[ ]" and ended ".end".
	.dev_map [name] [id]	Mapping device name to device id. available to use name or id in the script. name: the name mapped to the id. id: I²C slave address declared in Custom Config (0x56 OPCode)
	.label [name]	The directive is to indicate the destination of goto_if_equal(….), goto_if_greater, goto_if_less command and goto(...) command.

Table 2 lists the commands recognized by the GUI for customizing the behavior of the IC.

Table 2. Commands for operation

TYPE	COMMAND	DESCRIPTION
I²C Command	i2c_write8(dev_id, address, data)	Function to write 8-bit hex data
GPIO Command	gpio_write(gpio no, state)	Function to write values to a specific GPIO
Register Command	reg_write(register_name, data, mask)	Function to write values to a specific internal register
	reg_read(register_name, mask)	Function to read values from a specific internal register
Common	delay_ms(time)	Time delay during (time) ms. Time = 1~5 (min: 1, max: 5) The delay may cause a timeout issue with PD communication, so use this API with care.
Flow Control Command	goto(label_name)	Jump statement used to go to to a specific label, cannot be used to jump to another event.
	goto_if_equal_r(var_read, data_b, label_name)	Jump statement used to go to a specific label if the read value var_read and data_b are equal, cannot be used to jump to another event.
	goto_if_greater_r(var_read, data_b, label_name)	Jump statement used to go to a specific label if the read value var_read is greater than or equal to data_b cannot be used to jump to another event.
	goto_if_less_r(var_read, data_b, label_name)	Jump statement used to go to a specific label if the read value var_read is less than or equal to data_b, cannot be used to jump to another event.

Event List

The events related to changes on the physical pins on the USB Type-C connector, MAX77958 pins, or registers as well as requirements according to PD messages are summarized in Table 3.

Table 3. Event List

ID	NAME	TRIGGERED BY	TYPICAL USE	SAMPLE CUSTOMIZATION COMMANDS
0	REQ_TURN_ON_VBUS	Hard_reset PD message in SOURCE Mode or PR_Swap PD message when V_{BUS} is lower than vSafe0V	1. Hard Reset: When MAX77958 is under the SOURCE Mode and is required to turn on V_{BUS} 2. PR_Swap: MAX77958 is requested to become the new SOURCE and is required to turn on V_{BUS}	*.dev_map chg 0* *.event REQ_TURN_ON_VBUS* *i2c_write8(chg, 0x16, 0x8A); I²C Mode Enable, TURN ON V_{BUS}* *.end*
1	REQ_TURN_OFF_VBUS	Hard_reset PD message in SOURCE Mode or PR_Swap PD message when V_{BUS} is higher than vSafe5V, USB port is disconnected, or enters contract as SINK	1. Hard Reset: When MAX77958 is under the SOURCE Mode and is required to turn off V_{BUS} 2. PR_Swap or USB port disconnect or enter contract as SINK: MAX77958 is the old SOURCE and is required to turn off V_{BUS}	*.dev_map chg 0* *.event REQ_TURN_OFF_VBUS* *i2c_write8(chg, 0x16, 0x82); I²C Mode Enable, TURN OFF V_{BUS}* *.end*
4	DET_ATTACHED_SNK	Entering the Attached.SNK state, Rp is detected on one of the CCx	SINK Mode: System can pull current from V_{BUS}	*.dev_map chg 0* *.event DET_ATTACHED_SNK* *i2c_write8(chg, 0x16, 0x85)); I²C Mode Enable, Charger on, DCDC on* *.end*
5	DET_ATTACHED_SRC	Entering the Attached.SRC state, Rd is detected on one of the CCx	SOURCE Mode: External device requests power from V_{BUS}	*.dev_map chg 0* *.event DET_ATTACHED_SRC* *i2c_write8(chg, 0x16, 0x8A); I²C Mode Enable, OTG on* *.end*
6	DET_UNATTACHED_DRP	Entering the unattached DRP mode, CCx is open	DRP.Mode (UnAttached state) MAX77958 is waiting for an attached port.	*.event DET_UNATTACHED_DRP* *gpio_write(4, 1)* *.end*
7	DET_AUDIO_ACCESSORY	Detected Audio Accessory, Ra is detected on both CC lines	Audio signal path enable: DN to DN1, DP to DP2 switches	*.event DET_AUDIO_ACCESSORY* *reg_write(BC_CTRL1_CTRL2, 0x0900, 0x3F00)*

				.end
8	DET_DEBUG_SRC	Detected Debug Accessory, Rd is detected on both CC lines	Debug path enable: DN to DN1, DP to DP2 switches	*.event DET_DEBUG_SRC* *reg_write(BC_CTRL1_CTRL2, 0x0900, 0x3F00)* *.end*
9	DET_DEBUG_SNK	Detected Debug Accessory, Rp is detected on both CC lines	Debug path enable: DN to DN1, DP to DP2 switches	*.event DET_DEBUG_SNK* *reg_write(BC_CTRL1_CTRL2, 0x0900, 0x3F00)* *.end*
11	DET_CC1_ACTIVE	Detected Rp or Rd on CC1	Notify CC1 is active to external MUX	*.event DET_CC1_ACTIVE* *gpio_write(1, 1)* *.end*
12	DET_CC2_ACTIVE	Detected Rp or Rd on CC2	Notify CC2 is active to external MUX	*.event DET_CC2_ACTIVE* *gpio_write(1, 0)* *.end*
16	DET_SINKCCI_0P5	MAX77958 is under SINK mode, detected 56kΩ Rp (500mA source)	Set charger input current limit to 500mA	*.dev_map chg 0* *.event DET_SINKCCI_0P5* *i2c_write8(chg, 0x1E, 0x0B); 0.5A* *.end*
17	DET_SINKCCI_1P5	MAX77958 is under SINK mode, detected 22kΩ Rp (1.5A source)	Set charger input current limit to 1.5A	*.dev_map chg 0* *.event DET_SINKCCI_1P5* *i2c_write8(chg, 0x1E, 0x1F)* *.end*
18	DET_SINKCCI_3P0	MAX77958 is under SINK mode, detected 10kΩ Rp (3.0A source)	Set charger input current limit to 3.0A	*.dev_map chg 0* *.event DET_SINKCCI_3P0* *i2c_write8(chg, 0x1E, 0x3D) ; CHGIN_ILIM 3A* *.end*
24	DET_VSAFE5V	Detected V_{BUS} is higher than vSafe5V	Indication V_{BUS} is present	*.event DET_VSAFE5V* *gpio_write(3, 1)* *.end*
25	DET_VSAFE0V	Detected V_{BUS} is lower than vSafe0V	Indication V_{BUS} is not present and Enable External discharge circuit on the V_{BUS} Path	*.event DET_VSAFE0V* *gpio_write(3, 0)* *.end*
37	DET_MOISTURE	Detected Moisture	Notify moisture is present on the Type-C connector	*.event DET_MOISTURE* *gpio_write(8, 1)* *.end*
38	DET_DRY	Detected Dry	Notify no moisture is present on the Type-C connector	*.event DET_DRY* *gpio_write(8, 0)* *.end*

40	REC_PD_HARDRESET_SNK	In sink mode, received hard-reset PD Message	Required charger input current limit to minimum value the sink shall not draw more than iSafe0mA when V_{BUS} is driven to vSafe0V	*.dev_map chg 0* *.event REC_PD_HARDRESET_SNK* *i2c_write8(chg, 0x1E, 0x00); 100mA* *.end*
41	DONE_PD_HARDRESET_SNK	In sink mode, hard-reset sequence completed	Required charger current limit to acquired value, the sink can draw current expected	*.dev_map chg 0* *.event DONE_PD_HARDRESET_SNK* *i2c_write8(chg, 0x1E, 0x3D); CHGIN_ILIM 3A* *.end*
50	DONE_POR	MAX77958 Power on sequence completed	Configure external device register to be initial setting	*.dev_map chg 0* *.event DONE_POR* *i2c_write8(chg, 0x16, 0x80); I^2C Mode Enable, Charger=OTG=DCDC off* *.end*
51~59	DET_GPIO#_LOW	Detected GPIO# of MAX77958 going from high to low	Indication for GPIO# changes from high to low	*.dev_map chg 0* *.event DET_GPIO8_LOW* *gpio_write(7, 1)* *.end*
60	USBPD_IDLE	MAX77958 Power on sequence completed, PD communication changes to the idle state, waiting for PD message.	Set the external device to the initial state	*.dev_map chg 0* *.event USBPD_IDLE* *i2c_write8(chg, 0x16, 0x80); I^2C Mode Enable, Charger=OTG=DCDC off* *.end*
66~71	SNK_REQ_POS#	When MAX77958 is configured source mode, MAX77958 advertises the PDO options.	Set external charger OTG voltage and current limit based on the system definition.	*.event SNK_REQ_POS0* *i2c_write8(chg, 0x1F, 0x8F)* *.end*
73	REQ_SWITCH_CONTROL_SDP_CDP	Detected SDP or CDP	Set USB Switch to close	*.event REQ_SWITCH_CONTROL_SDP_CDP* *reg_write(BC_CTRL1_CTRL2, 0x0900, 0x3F00)* *.end*
74	REQ_SWITCH_CONTROL_DCP	Detected DCP	Set USB Switch to open	*.event REQ_SWITCH_CONTROL_DCP* *reg_write(BC_CTRL1_CTRL2, 0x0020, 0x0020)* *.end*
75	REQ_SWITCH_CONTROL_DCDTO	Detected Data Contact Detection Timeout	Set USB Switch to close	*.event REQ_SWITCH_CONTROL_DCDTO*

				reg_write(BC_CTRL1_CTRL2, 0x0900, 0x3F00) *.end*
83	DET_SINKPDI_0P0_TO_0P5	When SrcCap Current is detected 0<SrcCur<0.5A	Set input current limit to the minimum value	*.event DET_SINKPDI_0P0_TO_0P5* *i2c_write8(chg, 0x1E, 0x02)* *.end*
84	DET_SINKPDI_0P5_TO_1P0	When SrcCap Current is detected 0.5A<=SrcCur<1A	Set input current limit to 0.5A	*.event DET_SINKPDI_0P5_TO_1P0* *i2c_write8(chg, 0x1E, 0x0B)* *.end*
85	DET_SINKPDI_1P0_TO_1P5	When SrcCap Current is detected 1A<=SrcCur<1.5A	Set input current limit to 1.0A	*.event DET_SINKPDI_1P0_TO_1P5* *i2c_write8(chg, 0x1E, 0x15)* *.end*
86	DET_SINKPDI_1P5_TO_2P0	When SrcCap Current is detected 1.5A<=SrcCur<2A	Set input current limit to 1.5A	*.event DET_SINKPDI_1P5_TO_2P0* *i2c_write8(chg, 0x1E, 0x1F)* *.end*
87	DET_SINKPDI_2P0_TO_2P5	When SrcCap Current is detected 2A<=SrcCur<2.5A	Set input current limit to 2.0A	*.event DET_SINKPDI_2P0_TO_2P5* *i2c_write8(chg, 0x1E, 0x29)* *.end*
88	DET_SINKPDI_2P5_TO_3P0	When SrcCap Current is detected 2.5A<=SrcCur<3A	Set input current limit to 2.5A	*.event DET_SINKPDI_2P5_TO_3P0* *i2c_write8(chg, 0x1E, 0x33)* *.end*
89	DET_SINKPDI_3P0_TO_3P5	When SrcCap Current is detected 3A<=SrcCur<3.5A	Set input current limit to 3.0A	*.event DET_SINKPDI_3P0_TO_3P5* *i2c_write8(chg, 0x1E, 0x3D)* *.end*
90	DET_SINKPDI_3P5_TO_4P0	When SrcCap Current is detected 3.5A<=SrcCur<4A	Set input current limit to 3.5A	*.event DET_SINKPDI_3P5_TO_4P0* *i2c_write8(chg, 0x1E, 0x47)* *.end*
91	DET_SINKPDI_UPPER_4P0	When SrcCap Current is detected SrcCur>=4A	For the input current limit higher than 4A, set it according to the customer definition	*.event DET_SINKPDI_UPPER_4P0* *i2c_write8(chg, 0x1E, 0x51)* *.end*

Customization Script GUI Interface

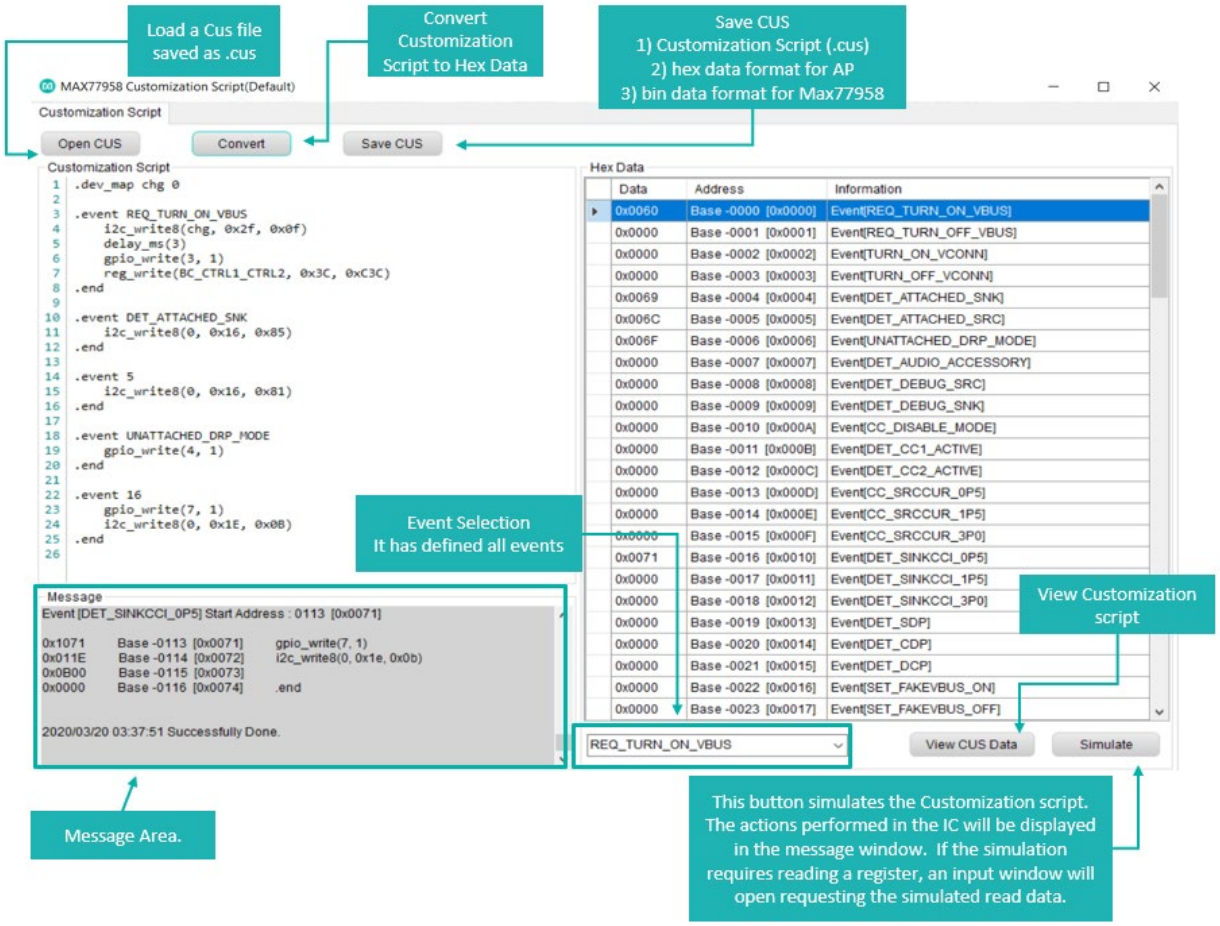

Figure 2. Description of the Customization Script Generation Interface

Customization Script Download Flowchart

The following flowchart describes the process of using the GUI to edit or create a customization script and convert the script to programmable data in the IC.

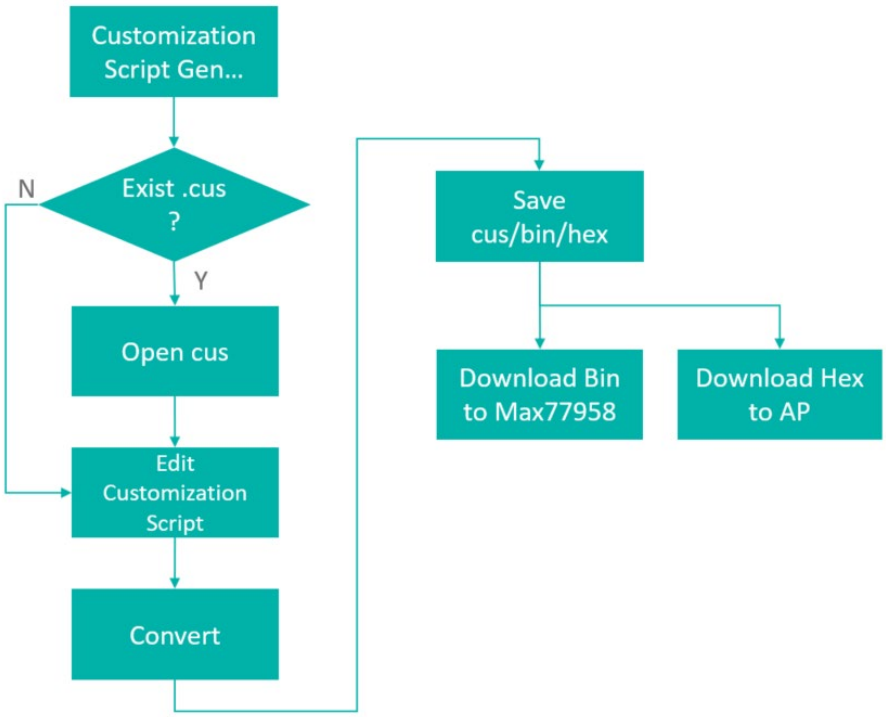

Figure 3. Downloading Customization Script

OPCode Commands

All configuration and control commands to the MAX77958 are sent and received as a packet using an OPCode to identify the packet. The MAX77958 contains a 32-byte buffers for reading and writing OPCode commands.

Simplified Block Diagram of the OPCode Command Process

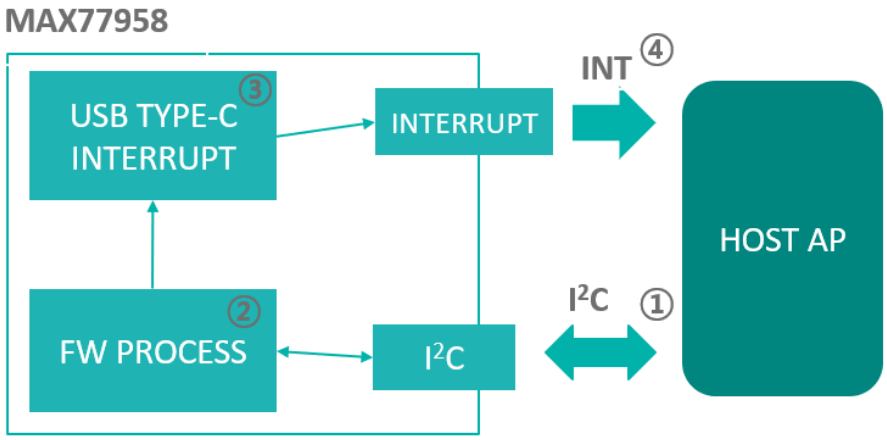

Figure 4. OPCode Command Process

1) The host AP sends the OPCode Command.
2) The MAX77958 FW processes the received command.
3) When the process of the command by the firmware is finished, a USB Type-C Interrupt occurs.
4) MAX77958 sends INT to notify the host AP that the processing of the command was completed.

OPCode Command Format

Messages sent to the MAX77958

The 0x21 register requires an OPCode command. 0x22 to 0x41 registers contain optional messages written by the user, which should be between 0 and 32 bytes in length. These registers are not cleared automatically, the values remain until they are overwritten with new messages by the application processor. The 0x41 register should contain a last message that is recognizable by the MAX77958.

Example 1: OPCode command that does not have a message register

Write 0x21 register: OPCode command

Write 0x41 register: by default, 0 is recognized by the MAX77958

Example 2: OPCode command that has one message register

Write 0x21 register: OPCode command

Write 0x22 register: optional message

Write 0x41 register: by default, 0 is recognized by the MAX77958

Example 3: OPCode command with two message registers

Write 0x21 register: OPCode command

Write 0x22 register: first message

Write 0x23 register: second message

Write 0x41 register: by default, 0 is recognized by the MAX77958

Messages received from MAX77958

The 0x51 register contains the OPCode command identifying the message type. 0x52 to 0x71 registers contain messages returned by the MAX77958. These registers are not cleared automatically, values remain until overwritten with new messages by the MAX77958. After APCmdRes Interrupt occurs, the message returned by the MAX77958 can be read in a flexible from 0 to 32 bytes.

Example 1: OPCode command that has a returned message register

Read 0x51 register: OPCode command returned by MAX77958

Read 0x52 register: message returned by MAX77958

Example 2: OPCode command that has two returned message registers

Read 0x51 register: OPCode command returned by MAX77958

Read 0x52 register: first message returned by MAX77958

Read 0x53 register: second message returned by MAX77958

ADDR [HEX]	NAME	DEFAULT [HEX]	USB-C TYPE	AP TYPE	B7	B6	B5	B4	B3	B2	B1	B0
0x20	Reserved	0x00	RO	RW	0x00							
0x21	AP_DATAOUT0	0x00	RO	RW	AP Request Opcode							
0x22	AP_DATAOUT1	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x23	AP_DATAOUT2	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x24	AP_DATAOUT3	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x25	AP_DATAOUT4	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x26	AP_DATAOUT5	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x27	AP_DATAOUT6	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x28	AP_DATAOUT7	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x29	AP_DATAOUT8	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2A	AP_DATAOUT9	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2B	AP_DATAOUT10	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2C	AP_DATAOUT11	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2D	AP_DATAOUT12	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2E	AP_DATAOUT13	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x2F	AP_DATAOUT14	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x30	AP_DATAOUT15	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x31	AP_DATAOUT16	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x32	AP_DATAOUT17	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x33	AP_DATAOUT18	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x34	AP_DATAOUT19	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x35	AP_DATAOUT20	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x36	AP_DATAOUT21	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x37	AP_DATAOUT22	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x38	AP_DATAOUT23	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x39	AP_DATAOUT24	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x3A	AP_DATAOUT25	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0

0x3B	AP_DATAOUT26	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x3C	AP_DATAOUT27	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x3D	AP_DATAOUT28	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x3E	AP_DATAOUT29	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x3F	AP_DATAOUT30	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x40	AP_DATAOUT31	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x41	AP_DATAOUT32	0x00	RO	RW	B7	B6	B5	B4	B3	B2	B1	B0
0x42-0x50	Not used	0x00	RO	RO	0x00							
0x51	AP_DATAIN0	0x00	RW	RO	MAX77958 response Opcode							
0x52	AP_DATAIN1	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x53	AP_DATAIN2	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x54	AP_DATAIN3	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x55	AP_DATAIN4	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x56	AP_DATAIN5	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x57	AP_DATAIN6	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x58	AP_DATAIN7	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x59	AP_DATAIN8	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5A	AP_DATAIN9	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5B	AP_DATAIN10	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5C	AP_DATAIN11	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5D	AP_DATAIN12	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5E	AP_DATAIN13	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x5F	AP_DATAIN14	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x60	AP_DATAIN15	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x61	AP_DATAIN16	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x62	AP_DATAIN17	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x63	AP_DATAIN18	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x64	AP_DATAIN19	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x65	AP_DATAIN20	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x66	AP_DATAIN21	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x67	AP_DATAIN22	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x68	AP_DATAIN23	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x69	AP_DATAIN24	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6A	AP_DATAIN25	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6B	AP_DATAIN26	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6C	AP_DATAIN27	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6D	AP_DATAIN28	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6E	AP_DATAIN29	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x6F	AP_DATAIN30	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x70	AP_DATAIN31	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0
0x71	AP_DATAIN32	0x00	RW	RO	B7	B6	B5	B4	B3	B2	B1	B0

OPCode Example

This is an example of using OPCode in AP and how to write OPCode.

- A DRP shall transition to either Unattached.SNK or Unattached.SRC.
- Request Sink Mode setting from AP to MAX77958 via Customer Configuration OPCode(0x56)

  // I²C address, Register Address, OPCode Command

  I2C_WRITE(0x4A, 0x21, 0x56)

// I²C address, Register Address, Sink mode only

I2C_WRITE(0x4A, 0x22, 0x50)

// I²C address, Register Address, USB VID 0x0B6A

I2C_WRITE(0x4A, 0x23, 0x6A)

I2C_WRITE(0x4A, 0x24, 0x0B)

// I²C address, Register Address, USB PID 0x6860

I2C_WRITE(0x4A, 0x25, 0x60)

I2C_WRITE(0x4A, 0x26, 0x68)

// I²C address, Register Address, SRC PDO voltage 5000mV

I2C_WRITE(0x4A, 0x27, 0x00)

I2C_WRITE(0x4A, 0x28, 0x64)

// I²C address, Register Address, SRC PDO Max Current 1500mV

I2C_WRITE(0x4A, 0x29, 0x00)

I2C_WRITE(0x4A, 0x2A, 0x96)

// I²C address, Register Address , End of command

// End of command shall be written to be recognized by MAX77958

I2C_WRITE(0x4A, 0x41, 0x00)

- MAX77958's firmware sets the Sink Mode only
- TYPE-C FSM of MAX77958 shall set Unattached.SNK
- When APCmdRes Interrupt occurs, AP can check the setting result from 0x52 to 0x71 register.

  // Slave address, Register Address, Respond Data

  I2C_READ(0x4A, 0x51, 0x56)

OPCode Register Information

This is read/write register information of OPCode.

0x01: BC CTRL1 Config Read

ADDR	BITFIELD	BC_CTRL1			RESET	0X81		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x01							
0x51	0x01							

0x52	DCDCpl	RSVD	RSVD	RSVD	NikonDet	RSVD	CHGDetMan	CHGDetEn

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
DCDCpl	7	1	Data Contact Detection Wait Time	0 = 2000ms 1 = 900ms
RSVD	6:4	0	Reserved	
Nikon Detection	3	0	Nikon Charger Detection	0 = Not enabled 1 = Enabled
RSVD	2	0	Reserved	
CHGDetMan	1	0	Force Manual Run of Charger Detection, Bit Auto Resets to 0	0 = Not enabled 1 = Request manual run of charger detection
CHGDetEn	0	1	Enable Charger Detection	0 = Not enabled 1 = Enabled, charger detection runs every time V_{BUS} > V_{VBDET} and DetAbrt = 0

0x02: BC CTRL1 Config Write

ADDR	BITFIELD	BC_CTRL1			RESET	0X81		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x02							
0x22	DCDCpl	RSVD	RSVD	RSVD	NikonDet	RSVD	CHGDet Man	CHGDet En
0x51	0x02							

0x05: Control1 Read

ADDR	BITFIELD	CONTROL1			RESET	0X00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x05							
0x51	0x05							
0x52	RSVD	RSVD	COMP2Sw			COMN1Sw		

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RSVD	7:6	0	Reserved	
COMP2Sw	5:3	0	Control of COMP2 Switches	000 = Open 001 = COMP2 connected to DN2(USB) 010 to 111 = Open

COMN1Sw	2:0	0	Control of COMN1 Switches	000 = Open
				001 = COMN1 connected to DN1(USB)
				010 to 111 = Open

0x06: Control1 Write

ADDR	BITFIELD	CONTROL1			RESET	0X00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x06							
0x22	RSVD	RSVD	COMP2Sw			COMN1Sw		
0x51	0x06							

0x0B: CC Control1 Read

ADDR	BITFIELD	CC_CONTROL1			RESET	0X81		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x0B							
0x51	0x0B							
0x52	CCVcnEn	CCTrySnkEn	RSVD	CCDbgSrcEn	CCDbgSnkEn	CCAudEn	CCSrcEn	CCSnkEn

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
CCVcnEn	7	1	Force State of V_{CONN}	0 = Force V_{CONN} off (both external boost converter and V_{CONN} switch)
				1 = Automatic operation based on State Machine
CCTrySnkEn	6	0	Allow Transition to TrySnk States	0 = Try SINK is disabled
				1 = Try SINK is enabled
RSVD	5	0	Reserved	
CCDbgSrcEn	4	0	Enable Detection of Type-C Debug Source Adapter	0 = Disabled
				1 = Enabled
CCDbgSnkEn	3	0	Enable Detection of Type-C Debug Sink Adapter	0 = Disabled
				1 = Enabled
CCAudEn	2	0	Enable Detection of Type-C Audio Adapter	0 = Disabled
				1 = Enabled
CCSrcEn	1	0	Enable Detection of Type-C Source Adapter	0 = Disabled
				1 = Enabled
CCSnkEn	0	1		0 = Disabled

			Enable Detection of Type-C Sink Adapter	1 = Enabled

0x0C: CC Control1 Write

ADDR	BITFIELD	CC_CONTROL1			RESET	0X81		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x0C							
0x22	CCVcnEn	CCTrySnkEn	RSVD	CCDbgSrcEn	CCDbgSnkEn	CCAudEn	CCSrcEn	CCSnkEn
0x51	0x0C							

0x11: CC Control4 Read

ADDR	BITFIELD	CC_CONTROL4			RESET	0X00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x11							
0x51	0x11							
0x52	CCVcnOcpEn	RSVD	RSVD	RSVD	RSVD	RSVD	CCDrpPhase	

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
CCVcnOcpEn	7	0	V_{CONN} OCP enable	0 = V_{CONN} OCP does have impact on V_{CONN} SW and BOOST
				1 = V_{CONN} OCP turn-off V_{CONN} SW and BOOST after 12ms
RSVD	6:1	0	Reserved	
CCDrpPhase	1:0	0	Percent of time device is acting as Unattached.SRC when CCSNKSRC=1 and CCSRCSNK=1	00 = 35%
				01 = 40%
				10 = 45%
				11 = 50%

0x12: CC Control4 Write

ADDR	BITFIELD	CC_CONTROL4			RESET	0X00		
	b7	b6	b5	b4	b3	b2	b1	b0

0x21	0x12						
0x22	CCVcnOcpEn	RSVD	RSVD	RSVD	RSVD	RSVD	CCDrpPhase
0x51	0x12						

0x23: GPIO Control Read

ADDR	NAME	GPIO_READ			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x23							
0x51	0x23							
0x52	GPIO3 Output	GPIO3 Direction	GPIO2 Output	GPIO2 Direction	GPIO1 Output	GPIO1 Direction	GPIO0 Output	GPIO0 Direction
0x53	GPIO7 Output	GPIO7 Direction	GPIO6 Output	GPIO6 Direction	GPIO5 Output	GPIO5 Direction	GPIO4 Output	GPIO4 Direction
0x54	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	GPIO8 Output	GPIO8 Direction

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
GPIO3_OUT	7	0	GPIO3 Output	0 = Low 1 = High
GPIO3_Direction	6	0	GPIO3 Direction	0 = Input 1 = Output
GPIO2_OUT	5	0	GPIO2 Output	0 = Low 1 = High
GPIO2_Direction	4	0	GPIO2 Direction	0 = Input 1 = Output
GPIO1_OUT	3	0	GPIO1 Output	0 = Low 1 = High
GPIO1_Direction	2	0	GPIO1 Direction	0 = Input 1 = Output
GPIO0_OUT	1	0	GPIO0 Output	0 = Low 1 = High
GPIO0_Direction	0	0	GPIO0 Direction	0 = Input 1 = Output
GPIO7_OUT	7	0	GPIO7 Output	0 = Low 1 = High

GPIO7_Direction	6	0	GPIO7 Direction	0 = Input 1 = Output
GPIO6_OUT	5	0	GPIO6 Output	0 = Low 1 = High
GPIO6_Direction	4	0	GPIO6 Direction	0 = Input 1 = Output
GPIO5_OUT	3	0	GPIO5 Output	0 = Low 1 = High
GPIO5_Direction	2	0	GPIO5 Direction	0 = Input 1 = Output
GPIO4_OUT	1	0	GPIO4 Output	0 = Low 1 = High
GPIO4_Direction	0	0	GPIO4 Direction	0 = Input 1 = Output
GPIO8_OUT	1	0	GPIO8 Output	0 = Low 1 = High
GPIO8_Direction	0	0	GPIO8 Direction	0 = Input 1 = Output

0x24: GPIO Control Write

ADDR	NAME	GPIO_WRITE			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x24							
0x22	GPIO3 Output	GPIO3 Direction	GPIO2 Output	GPIO2 Direction	GPIO1 Output	GPIO1 Direction	GPIO0 Output	GPIO0 Direction
0x23	GPIO7 Output	GPIO7 Direction	GPIO6 Output	GPIO6 Direction	GPIO5 Output	GPIO5 Direction	GPIO4 Output	GPIO4 Direction
0x24	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	GPIO8 Output	GPIO8 Direction
0x51	0x24							

0x27: GPIO0 GPIO1 ADC Read

ADDR	NAME	SBU_READ			RESET	0X00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x27							
0x51	0x27							
0x52	SBU1							

0x53	SBU2

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
SBU1	7:0	0	Indicates value on V_{BUS} Input of SBU1	The value of SBU1 ADC range
SBU2	7:0	0	Indicates value on V_{BUS} Input of SBU2	The value of SBU2 ADC range

0x2F: Get Sink Cap

The Get_Sink_Cap (get sink capabilities) message can be sent by a port to request the sink capabilities and dual-role power capability of its port partner. The port responds by returning a Sink_Capabilities message.

ADDR	BITFIELD	GET_SRC_CAP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x2F							
0x51	0x2F							
0x52	NumOfPDO			PwrRole	DataRole	RSVD	RSVD	RSVD
0x53	PDO1[7:0]							
0x54	PDO1[15:8]							
0x55	PDO1[23:16]							
0x56	PDO1[31:24]							
•••	••••••••••••							
	PDOx[7:0]							
	PDOx[15:8]							
	PDOx[23:16]							
	PDOx[31:24]							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
NumOfPDO	7:5		Number of Power Data Objects	
PwrRole	4		Power Role of Current Source	0 = Sink
				1 = Source
DataRole	3		Data Role of Current Source	0 = UFP
				1 = DFP
RSVD	2:0		Reserved	

0x30: Current Src Cap

The PDO is requested by sink and accepted by source among source capabilities.

ADDR	BITFIELD	CUR_SEL_SRC_CAP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x30							
0x51	0x30							
0x52	RSVD	RSVD	SEL_PDO_POS			NumOfPDO		
0x53	PDO1[7:0]							
0x54	PDO1[15:8]							
0x55	PDO1[23:16]							
0x56	PDO1[31:24]							
•••	••••••••••••							
	PDOx[7:0]							
	PDOx[15:8]							
	PDOx[23:16]							
	PDOx[31:24]							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RSVD	7:6		Reserved	
SEL_PDO_POS	5:3		Selected Position of POD	0b : Unselected – Should send request message using OPCODE 0x32, SrcCap Request 1b – 7b : Selected PDO number
NumOfPDO	2:0		Number of Power Data Objects	

0x31: Get Source Cap

The Get_Source_Cap (get source capabilities) message can be sent by a port to request the source capabilities and dual-role power capability of its port partner. The port responds by returning a Source_Capabilities message. The port responds by returning a Source_Capabilities Message Addr.

ADDR	BITFIELD	GET_SRC_CAP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x31							
0x51	0x31							
0x52	NumOfPDO			PwrRole	DataRole	RSVD	RSVD	RSVD
0x53	PDO1[7:0]							
0x54	PDO1[15:8]							

0x55	PDO1[23:16]
0x56	PDO1[31:24]
•••	••••••••••••
	PDOx[7:0]
	PDOx[15:8]
	PDOx[23:16]
	PDOx[31:24]

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
NumOfPDO	7:5		Number of Power Data Objects	
PwrRole	4		Power Role of Current Source	0 = Sink
				1 = Source
DataRole	3		Data Role of Current Source	0 = UFP
				1 = DFP
RSVD	2:0		Reserved	

0x32: Src Cap Request

Send the request message as the response of Source_Capablities message to port partner.

ADDR	BITFIELD	SRC_CAP_REQ			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x32							
0x22	RSVD	RSVD	RSVD	RSVD	RSVD	Req_PDO_Pos		
0x51	0x32							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RSVD	7:3		Reserved	
Req_PDO_Pos	2:0		Request Position of POD	

0x33: Set Src Cap

Set the current device's source capabilities.

ADDR	BITFIELD	SET_SRC_CAP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x33							
0x22	RSVD	RSVD	RSVD	RSVD	RSVD	NumOfPDO		
0x23	PDO1[7:0]							
0x24	PDO1[15:8]							
0x25	PDO1[23:16]							
0x26	PDO1[31:24]							

•••	••••••••••••••
	PDOx[7:0]
	PDOx[15:8]
	PDOx[23:16]
	PDOx[31:24]
0x51	0x33

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RSVD	7:3		Reserved	
NumOfPDO	2:0		Number of Power Data Objects	

0x35: Read the Response for Get Request

Read the response for the Get Request message by the port partner.

This OPCode should be used after PDMsg (0x35–0x39) interrupt happens.

ADDR	BITFIELD	READ_GET_REQ_RESP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x35							
0x51	0x35							
0x52	RespMsgType			MsgDataLen				
0x53	Data Objcet(s) or Extended Message Header[7:0]							
0x54	Data Object(s) or Extended Message Header[15:8]							
0x55	Data Object(s) or Data							
0x56	Data Object(s) or Data							
0x57	Data Object(s) or Data							
0x58	Data Object(s) or Data							
•••	••••••••••••••							
	Data Object(s) or Data							
	Data Object(s) or Data							
	Data Object(s) or Data							
	Data Object(s) or Data							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RespMsgType	7:5		Response Message Type	0x00 – Source_Cap_Extended
				0x01 – Status
				0x02 – Battery_Cap
				0x03 – Battery_Status
				0x04 – Manufacturer_Info

MsgDataLen	4:0		Message Data Length Bytes except Message Header	

0x36: Send Get Response

Send the response for Get Request message to port partner

ADDR	BITFIELD	SEND_GET_RESP			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x36							
0x22	RespMsgType			MsgDataLen				
0x23	Data Objcet(s) or Extended Message Header[7:0]							
0x24	Data Object(s) or Extended Message Header[15:8]							
0x25	Data Object(s) or Data							
0x26	Data Object(s) or Data							
0x27	Data Object(s) or Data							
0x28	Data Object(s) or Data							
0x29	••••••••••••							
0x2A	Data Object(s) or Data							
	Data Object(s) or Data							
	Data Object(s) or Data							
	Data Object(s) or Data							
0x51	0x36							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RespMsgType	7:5		Response Message Type	0x00 – Source_Cap_Extended
				0x01 – Status
				0x02 – Battery_Cap
				0x03 – Battery_Status
				0x04 – Manufacturer_Info
MsgDataLen	4:0		Message Data Length Bytes except Message Header	

0x37: Send Swap Request

Send the Swap Request message to port partner

ADDR	NAME	SWAP_REQ			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x37							

0x22							Swap_Name
0x51	0x37						
0x52	Result						

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
Swap_Name	1:0		Swap	0x00 : N/A
				0x01 : DR SWAP
				0x02 : PR SWAP
				0x03 : V_{CONN} SWAP
Result	7:0		Result of Swap	0x00 : Wait
				0x01 : Accepted from port partner
				0x02 : Rejected from port partner
				0xFA : Not Support
				0xFC : No Connection
				0xFD : Already Running
				0xFE : Moisture detection is enabled
				0xFF : Fail to send request

0x38: Send Swap Response

Set the Swap Response for the Swap Request message

ADDR	NAME	SWAP_REQ_RESPONSE			RESET	0x00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x38							
0x22			PR_SWP_Resp		DR_SWP_Resp		V_{CONN}_SWP_Resp	
0x51	0x38							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
PR_SWP_Resp	5:4		Response of PR Swap	0x00 : Accept Sink Role, Reject Source Role
				0x01 : Accept Source Role, Reject Sink Role
				0x02 : Accept Dual Role(Sink or Source)
				0x03 : Wait
DR_SWP_Resp	3:2		Response of DR Swap	0x00 : Accept UFP Role, Reject DFP Role
				0x01 : Accept DFP Role, Reject UFP Role
				0x02 : Accept Dual Role(UFP or DFP)
				0x03 : Wait

V_{CONN}_SWP_Resp	1:0		Response of V_{CONN} Swap	0x00 : Accept Turn Off V_{CONN}, Reject Turn On V_{CONN}
				0x01 : Accept Turn On V_{CONN}, Reject Turn Off V_{CONN}
				0x02 : Accept Turn Off/On V_{CONN}
				0x03 : Wait

0x3A: APDO SrcCap Request

ADDR	NAME	APDO_SRCCAP_REQUEST			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x3A							
0x22	REQ_APDO_POS							
0x23	OUTPUT_VOLTAGE_LOW							
0x24	OUTPUT_VOLTAGE_HIGH							
0x25	RSVD	OPERATING_CURRENT						
0x51	0x3A							
0x52	Result							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
REQ_APDO_POS	7:0		Request Position of PDO	PDO Position
OUTPUT_VOLTAGE_LOW	7:0		Low bit of Output Voltage	0x0000 = Min Output Voltage, 0x0001 = 20mV (Low 0x01, High 0x00), 0x0002 = 40mV (Low 0x02, High 0x00), 0x0003 = 60mV, 0x0004 = 80mV, 0x0005 = 100mV, … 0x00FA = 5000mV (Low 0xFA, High 0x00), …
OUTPUT_VOLTAGE_HIGH	7:0		High bit of Output Voltage	0x01C2 = 9000mV (Low 0xC2, High 0x01), …

				0x03E8 = 20000mV (Low 0xE8, High 0x03)
OPERATING_CURRENT	7:0		Operate Current	0x00 = Max Operating Current, 0x01 = 50mA 0x02 = 100mA 0x03 = 150mA … 0x1E = 1500mA … 0x28 = 2000mA … 0x3C = 3000mA … 0x7C = 6200mA 0x7D – 0xFF = Reserved
Result	7:0		Result	0x00 = Sent APDO Request Message 0x01 = Error, Invalid APDO position 0x02 = Error, Invalid Output Voltage 0x03 = Error, Invalid Operating Current 0x04 = Error, PPS Function Off 0x05 = Error, Not in SNK Ready State 0x06 = Error, PD 2.0 Contract 0x07 = Error, SinkTxNg

0x3C: Set PPS

ADDR	NAME	SET_PPS			RESET	0x00		
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x3C							
0x22	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	PPS On
0x23	DEFAULT_OUTPUT_VOLTAGE_LOW							
0x24	DEFAULT_OUTPUT_VOLTAGE_HIGH							
0x25	RSVD	DEFAULT_OPERATING_CURRENT						
0x51	0x3C							
0x52	Result							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
DEFAULT_OUTPUT_VOLTAGE_LOW	7:0		Low bit of Default Output Voltage	0x0000 = Min Output Voltage, 0x0001 = 20mV (Low 0x01, High 0x00), 0x0002 = 40mV (Low 0x02, High 0x00), 0x0003 = 60mV, 0x0004 = 80mV, 0x0005 = 100mV, … 0x00FA = 5000mV (Low 0xFA, High 0x00), … 0x01C2 = 9000mV (Low 0xC2, High 0x01), … 0x03E8 = 20000mV (Low 0xE8, High 0x03)
DEFAULT_OUTPUT_VOLTAGE_HIGH	7:0		High bit of Default Output Voltage	
DEFAULT_OPERATING_CURRENT	7:0	0	Default Operating Current	0x00 = Max Operating Current, 0x01 = 50mA, 0x02 = 100mA, 0x03 = 150mA, … 0x1E = 1500mA, … 0x28 = 2000mA, … 0x3C = 3000mA, … 0x7C = 6200mA, 0x7D – 0xFF = Reserved
Result	7:0	0	Result	0x00 = PPS Off, 0x01 = PPS On, 0x06 = DP Configured State

0x3E: SNK PDO Request

ADDR	BITFIELD	SNK_PDO_REQUEST_READ			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x3E							
0x22	Read SNK PDO	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
0x51	0x3E							
0x52	Number of PDOs							
0x53	PDO1[7:0]							
0x54	PDO1[15:8]							
0x55	PDO1[23:16]							
0x56	PDO1[31:24]							
•••	••••••••							
	PDOx[7:0]							
	PDOx[15:8]							
	PDOx[23:16]							
	PDOx[31:24]							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RSVD	6:0		Reserved	
Read SNK PDO	7		Read SNK PDO	0x0 = RAM
				0x1 = MTP

0x3F: SNK PDO Set

The maximum SNK PDO number is 5.

ADDR	BITFIELD	SNK_PDO_SET_SET			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x3F							
0x22	Write SNK PDO	RSVD	RSVD	RSVD	RSVD	NumOfPDO		
0x23	PDO1[7:0]							
0x23	PDO1[15:8]							
0x24	PDO1[23:16]							
0x25	PDO1[31:24]							

0x26	••••••••
•••	PDOx[7:0]
	PDOx[15:8]
	PDOx[23:16]
	PDOx[31:24]
0x51	0x3F

0x52	Write SNK PDO	RSVD	RSVD	RSVD	RSVD	NumOfPDO

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
Write SNK PDO	7		Write SNK PDO	0x0 = RAM
				0x1 = MTP
RSVD	6:3		Reserved	
NumOfPDO	2:0		Number of Power Data Objects	

0x4A: Get PD Message

Get sent or received PD message.

ADDR	BITFIELD	GETPDMSG			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x4A							
0x22								RX/TX
0x51	0x4A							
0x52	RSVD	MsgType		MsgDataLen				
0x53	MsgHeader[7:0]							
0x54	MsgHeader[15:8]							
0x55	Data Objcet(s) or Extended Message Header[7:0]							
0x56	Data Object(s) or Extended Message Header[15:8]							
0x57	Data Object(s) or Data							
0x58	Data Object(s) or Data							
0x59	Data Object(s) or Data							
0x5A	Data Object(s) or Data							
•••	••••••••••••							
	Data Object(s) or Data							
	Data Object(s) or Data							
	Data Object(s) or Data							

	Data Object(s) or Data

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
RX/TX	0		RX or TX Message	0b = RX Message
				1b = TX Message
MsgType	6:5		Message Type	00b = Control Message
				01b = Data Message
				10b = Extended Message (for PD version 3.0)
MsgDataLen	4:0		Message Data Length Bytes except Message Header	

0x55: Customer Configuration Read

ADDR	BITFIELD	CUSTOM_CONFIG_INFO			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x55							
0x51	0x55							
0x52	Moisture Detection	Memory Update	TypeC_State		TrySNKMode	Audio Acc	DbgTargetSNK	DbgTargetSRC
0x53	VID[7:0]							
0x54	VID[15:8]							
0x55	PID[7:0]							
0x56	PID[15:8]							
0x57	RSVD							
0x58	SRC_PDO_V[7:0]							
0x59	SRC_PDO_V[15:8]							
0x5A	SRC_PDO_MaxI[7:0]							
0x5B	SRC_PDO_MaxI[15:8]							
0x5C	RSVD							
0x5D	RSVD							
0x5E	RSVD							
0x5F	RSVD							
0x60	RSVD							
0x61	RSVD							
0x62	RSVD							
0x63	RSVD							
0x64	RSVD							
0x65	SID1[7:0]							
0x66	SID2[7:0]							

0x67	SID3[7:0]
0x68	SID4[7:0]

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
DbgTargetSRC	0	0	Debug Target Source Mode	0 = Disable
				1 = Enable
DbgTargetSNK	1	0	Debug Target Sink Mode	0 = Disable
				1 = Enable
AudioAcc	2	0	Audio Accesory Mode	0 = Disable
				1 = Enable
TrySNKMode	3	0	CC Try SNK Mode	0 = Disable
				1 = Enable
TypeC_State	5:4	2	TypeC State Machine	0 = SRC
				1 = SNK
				2 = DRP
Memory Update	6	0	Apply MTP Memory	0 = RAM
				1 = Update Customer Configuration Area of Memory
Moisture Detection	7	0	Enable Moisture Detection	0 = Disable
				1 = Enable
VID	15:0	0B6A	Custom VID	
PID	15:0	6860	Custom PID	
SRC_PDO_V	15:0	64	SRC PDO Voltage Output voltage in units of 50 mV. Valid values are 0-400 (0-20000 mV).	Valid range is 0~5000mV. (in 50mV step)
SRC_PDO_MaxI	15:0	96	SRC PDO Max Current PDO Type is set to 0 (fixed), or 2 (variable), then this field represents the maximum operating current in units of 10mA. If PDO Type is not set to 0 (fixed), 2 (variable), or 3 (PPS), then this field shall be ignored by testers.	Valid range is 0~5000mA. (in 10mA step)

SID1	7:0	69	I²C Slave Address 1 Used when defining dev_map in customization command on GUI. (.dev_map chg 0)	
SID2	7:0	69	I²C Slave Address 2 Used when defining dev_map in customization command on GUI. (.dev_map chg 1)	
SID3	7:0	35	I²C Slave Address 3 Used when defining dev_map in customization command on GUI. (.dev_map chg 2)	
SID4	7:0	28	I²C Slave Address 4 Used when defining dev_map in customization command on GUI. (.dev_map chg 3)	

0x56: Customer Configuration Write

ADDR	BITFIELD	CUSTOM_CONFIG_INFO			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x56							
0x22	Moisture Detection	Memory Update	TypeC_State		TrySNK Mode	AudioAcc	DbgTargetSNK	DbgTargetSRC
0x23	VID[7:0]							
0x24	VID[15:8]							
0x25	PID[7:0]							
0x26	PID[15:8]							
0x27	RSVD							
0x28	SRC_PDO_V[7:0]							
0x29	SRC_PDO_V[15:8]							
0x2A	SRC_PDO_MaxI[7:0]							

0x2B	SRC_PDO_MaxI[15:8]
0x2C	RSVD
0x2D	RSVD
0x2E	RSVD
0x2F	RSVD
0x30	RSVD
0x31	RSVD
0x32	RSVD
0x33	RSVD
0x34	RSVD
0x35	SID1[7:0]
0x36	SID2[7:0]
0x37	SID3[7:0]
0x38	SID4[7:0]
0x51	0x55

0x85: Master I2C Control Read

ADDR	NAME	MASTER_I2C_READ			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x85							
0x51	0x85							
0x52	SID[7:0]							
0x53	REG[7:0]							
0x54	LEN[7:0]							
0x55	Data0[7:0]							
0x56	Data1[7:0]							
0x57	Data2[7:0]							
0x58	Data3[7:0]							
•••	••••••••••••							
0x63	Data14[7:0]							
0x64	Data15[7:0]							
0x65	Data16[7:0]							

BITFIELD	BIT	RESET	DESCRIPTION	DECODE
SID	7:0		Slave Address	Slave Address
REG	7:0		Register	Indicate the read register
LEN	7:0		Length	Length for the read register
Data0	7:0		Read Data	1st Read Data

Data1	7:0		Read Data	2nd Read Data
Data2	7:0		Read Data	3rd Read Data
Data3	7:0		Read Data	4th Read Data
•••			•••	•••
Data14	7:0		Read Data	15th Read Data
Data15	7:0		Read Data	16th Read Data
Data16	7:0		Read Data	17th Read Data

0x86: Master I2C Control Write

ADDR	NAME	MASTER_I2C_WRITE			RESET			
	b7	b6	b5	b4	b3	b2	b1	b0
0x21	0x86							
0x22	SID[7:0]							
0x23	REG[7:0]							
0x24	LEN[7:0]							
0x25	Data0[7:0]							
0x26	Data1[7:0]							
0x27	Data2[7:0]							
0x28	Data3[7:0]							
•••	••••••••••••							
0x33	Data14[7:0]							
0x34	Data15[7:0]							
0x35	Data16[7:0]							
0x51	0x86							

Firmware Update

The firmware can be updated through the AP, MAX77958 GUI, or MAX77958 dongle board.

Firmware update through AP

At boot time, the AP compares the firmware version within the hex file to the target to determine if the firmware needs to be updated.

Firmware Update Flowchart

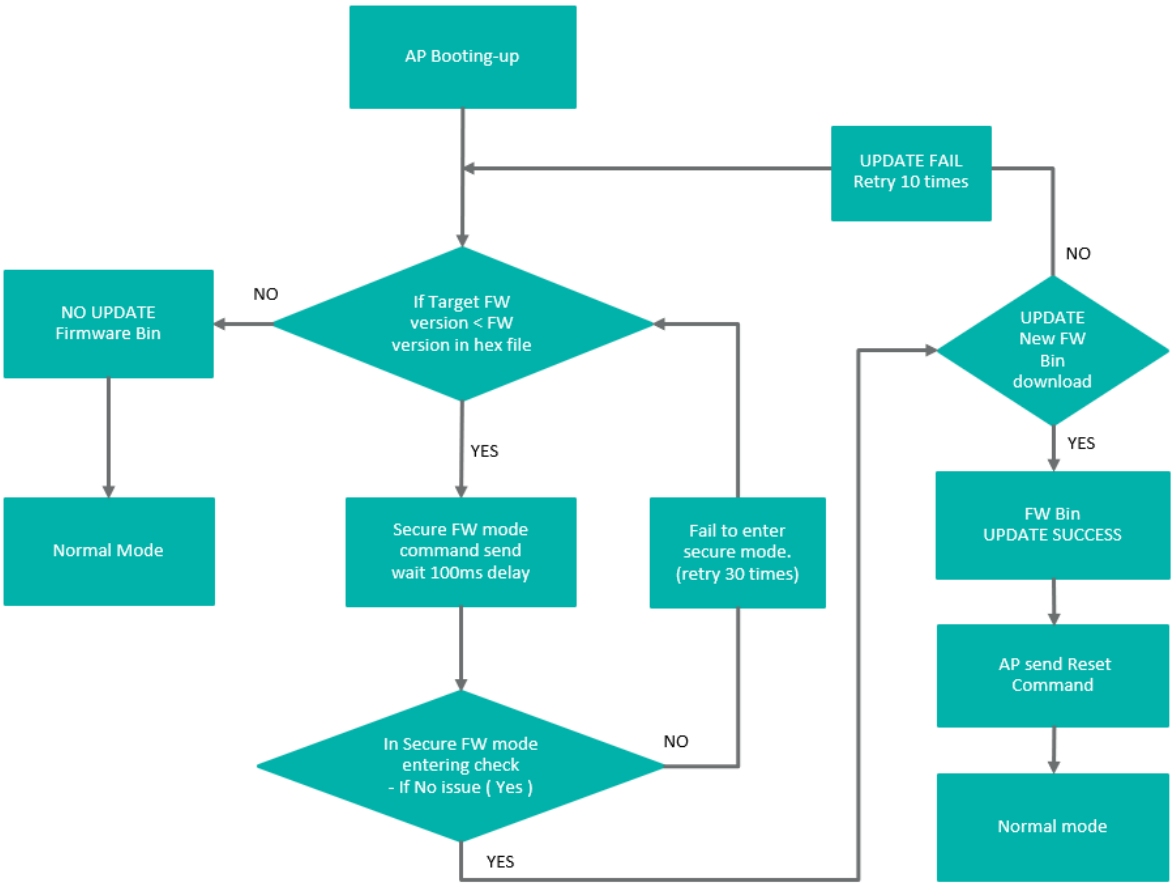

Figure 5. Flowchart of Firmware Update

Issues with RAM, decrypting data, or MTP read/write operations can prevent Secure FW mode from proceeding.

Firmware Update through GUI Interface

1. Select Firmware Update Menu on GUI

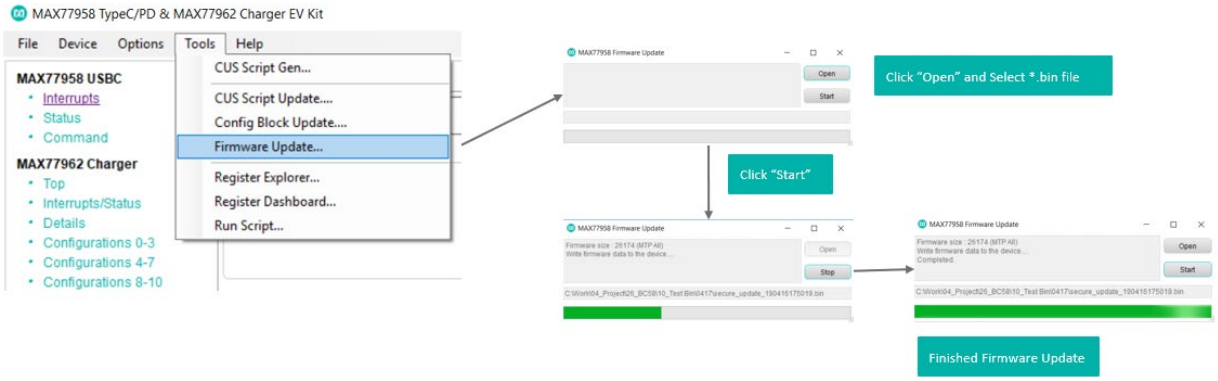

Figure 6. Firmware Update Procedure

2. Firmware Version verification.

The firmware version updated from 06.41 to 06.46

Figure 7. Firmware Version Readback

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	7/20	Initial release	—