How to Decode UCD90xxx Fault Log with PMBus

Yihe Hu

ABSTRACT

The UCD90xxx family devices are flexible and powerful enough to meet sequencing and monitoring needs. This application report addresses how to read and decode the fault logs with PMBus™ Commands. This document does not apply to the UCD9080 and UCD9081 devices. The example discussed in this document strongly focuses on the UCD9090A, but the concepts discussed apply to any particular system in which an UCD90xxx device is embedded.

Contents

List of Figures

List of Tables

Trademarks

PMBus is a trademark of SMIF, Inc..
All other trademarks are the property of their respective owners.

1 Introduction

The UCD90xxx family of digital sequencers, also known as system health monitors, are flexible and powerful enough to meet user sequencing, monitoring, margining and other needs. The entire families of devices are designed to have similar behaviors, but with a different number of rails or some other minor features. Users only need to learn how to use the device once, and can then seamlessly switch to other devices within the family that best fit their future designs. This document is to help applications to read the fault log data from the devices. This document does not apply to the UCD9080 and UCD9081 devices. All commands listed in the document can be found in the literatures listed in Section 7.

All byte values are represented in hexadecimal format. These are the codes to understand all I^2C communications that occur:

- [St] - This is the I^2C Start bit.
- [Sr] - This is the I^2C Restart bit. It is identical to the Start bit.
- [Sp] - This is the I^2C Stop bit.
- [A] - This is the I^2C Acknowledge bit.
- [N] - This is I^2C No Acknowledge bit or NACK.
- [AddrW] - This is the I^2C device address with the Write bit.
- [AddrR] - This is the I^2C device address with the Read bit.
- [W:x55] - This is an example of a write byte for value 55 hexadecimal.
- [W/R:Data_^{n}] - This is to indicate that a byte is being write/read by the I^2C master. The n subscript is an ordered integer use to distinguish multiple bytes read back. Data_1 is the MSB, and Data_2 is the LSB for two bytes read.

(1)

2 Fault Log Capacity

UCD90xxx families of digital sequencers have different fault log capacity due to the number of rails, features. Please refer to Table 1 for the details.

Table 1. Fault Log Capacity

Devices	UCD9090	UCD9090A	UCD90120A	UCD90124A	UCD90160/ UCD90160A	UCD90240/ UCD90320/ UCD90320U
# Fault Log	30	26	16	12	18	100

3 Command to Enable Fault Log Details

In order to log the fault details into non-volatile memory (NVM), the particular fault detail must be enabled first. By default, all fault details are enabled to log. The application can choose which fault detail to be logged by PMBus command LOGGED_FAULT_DETAIL_ENABLES (0xEFh). The command format follows the Table 2. The bits in each byte select if a fault detail is logged (1) or not (0).

Table 2. LOGGED_FAULT_DETAIL_ENABLES (0xEF) Command Format

Description	UCD9090/ UCD9090A	UCD90120A	UCD90124A	UCD90160/ UCD90160A	UCD90240	UCD90320/ UCD90320U
BYTE_COUNT	12	14	15	18	28	37
Non-Paged Faults (See Table 4)	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte

Table 2. LOGGED_FAULT_DETAIL_ENABLES (0xEF) Command Format (continued)

Description	UCD9090/ UCD9090A	UCD90120A	UCD90124A	UCD90160/ UCD90160A	UCD90240	UCD90320/ UCD90320U
GPI Faults (see Table 5)	1 Byte	1 Byte	1 Byte	1 Byte	3 Bytes	4 Bytes
Fan Faults (See Table 6)	NA	NA	1 Bytes	NA	NA	NA
Page Faults (See Table 7)	10 Bytes	12 Bytes	12 Bytes	16 Bytes	24 Bytes	32 Bytes

Table 3. LOGGED_FAULT_DETAIL_ENABLES I2C Code

DEVICE	PMBus DATA FORMAT
UCD9090/UCD9090A	[St] [AddrW] [A] [W:0xEF] [A] [W:0xC] [A] [W:NON_PAGED] [A] [W:GPI_FAULT] [A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #10][N] [Sp]
UCD90120A	[St] [AddrW] [A] [W:0xEF] [A] [W:0xE] [A] [W:NON_PAGED] [A] [W:GPI_FAULT] [A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #12][N] [Sp]
UCD90124A	[St] [AddrW] [A] [W:0xEF] [A] [W:0xF] [A] [W:NON_PAGED] [A][W:GPI_FAULT] [A][W:FAN_FAULT] [A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #12][N] [Sp]
UCD90160/UCD90160A	[St] [AddrW] [A] [W:0xEF] [A] [W:0x12] [A] [W:NON_PAGED] [A] [W:GPI_FAULT] [A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #16][N] [Sp]
UCD90240	[St] [AddrW] [A] [W:0xEF] [A] [W:0x1C] [A] [W:NON_PAGED] [A] [W:GPI1_FAULT] [A][W:GPI2_FAULT[A][W:GPI3_FAULT][A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #24][N] [Sp]
UCD90320/UCD90320U	[St] [AddrW] [A] [W:0xEF] [A] [W:0x25] [A] [W:NON_PAGED] [A] [W:GPI1_FAULT] [A][W:GPI2_FAULT[A][W:GPI3_FAULT][A][W:GPI4_FAULT][A][W:RAIL #1][A] [W:RAIL #2][A]…. [W:RAIL #32][N][Sp]

Table 4. Non-Paged Fault Log Bit Definition

BIT	7	6	5	4	3	2	1	0
Description	Reserved	Reserved	Reserved	Reserved	Watchdog Timeout	Reserved	System Watchdog Timeout	Reserved

Table 5. GPI Fault Log Bit Definitions

BIT	7	6	5	4	3	2	1	0
Description	GPI8	GPI7	GPI6	GPI5	GPI4	GPI4	GPI2	GPI1

If the device has more than eight GPIs, the order is BYTE1(GPI8–1), BYTE2(GPI16–9) ... BYTE4(GPI32–GPI25). Each byte follows Table 5.

Table 6. Fan Fault Log Bit Definitions

BIT	7	6	5	4	3	2	1	0
Description	Reserved	Reserved	Reserved	Reserved	FAN4	FAN3	FAN2	FAN1

Table 7. Paged Fault Log Bit Definitions

BIT	7	6	5	4	3	2	1	0
Description	SEQUENCE OFF TIMEOUT	SEQUENCE R ON TIMEOUT	OVER TEMPERAT URE	UNDER CURRENT	OVER CURRENT	TON MAX	UNDER VOLTAGE	OVER VOLTAGE

Each rail has a unique byte to define the fault log details, as shown in Table 7. The order is BYTE1(rail #1), BYTE2(rail #2), … BYTEN(rail #N).

3.1 Example to Enable Fault Log Details

Assume that the application only wants to log the UNDER_VOLTAGE and OVER_VOLTAGE faults of rail #2 and rail #4 for UCD9090A. The following data is sent to achieve the requirement.

[St] [AddrW] [A] [W:0xEF] [A] [W:0xC] [A] [W:00] [A] [W:00] [A] [W:00][A][W:03][A][W:00][A][W:03][A][W:00][A] [W:00][A] [W:00][A] [W:00][A] [W:00][A] [W:00][N] [Sp]

4 Commands to Read Fault Log

All UCD90xxx devices follow the same command sets to retrieve the fault log stored inside the non-volatile memory.

4.1 (EBh) LOGGED_FAULT_DETAIL_INDEX (MFR_SPECIFIC_27)

This command has the same format across whole UCD90xxx families of digital sequencers. The read form of this command is to report the total number of fault log entries, and the current value of the index into those entries.

Table 8. LOGGED_FAULT_DETAIL_INDEX COMMAND FORMAT

Byte Number	Description
1 [Data_2]	Fault Index (R/W)
2 [Data_1]	Total Number of LOGGED_FAULT_DETAIL entries (Read Only)

4.1.1 Read Total Fault Log Entries

Read the total fault log entries from the device:

[St] [AddrW] [A] [W:0xEB] [A] [Sr] [AddrR] [A] [R:Data_1] [A] [R:Data_2] [N] [Sp]

Assume that the data read back was Data_1 = 0x5h, Data_2 = 0x0h.This means that there are five fault logs available inside the device, and the first fault log is read. The next section discusses how to access the other fault log.

4.1.2 Access the Specific Fault Log

In order to access the fault log stored inside the device, the application needs to write the 0xEB command with proper fault index and the byte of the total number of entries, which is the same as the value returned from the read operation. Sending a fault index greater than or equal to the total number of fault long entries is prohibited.

The same data from the previous section is reused as example. Since device has, in total, five fault logs, the fault index is valid from 0–4. The following example shows that the fifth fault log is selected.

[St] [AddrW] [A] [W:0xEB] [A] [W:05] [A] [W:04] [N] [Sp]

4.2 (ECh) LOGGED_FAULT_DETAIL (MFR_SPECIFIC_28)

This command is to retrieve the raw data of the fault log selected with the LOGGED_FAULT_DETAIL_INDEX(0xEB) command. The command format varies upon device, but the decoding methodology is the same. Please refer to section 10.28 of the *UCD90xxx PMBus Command Reference Guide*, and section 10.24 of the *UCD90320 PMBus Command Reference Guide*. Here, the UCD9090A is used as example, and the rest of the devices follow the same methodology.

The single fault log is composed of three main parts:

- Milliseconds: tell milliseconds time and convert to the format of hour:minute:second
- Fault ID + Days: tell fault type and the years when the fault is present
- Fault Value: tell the exact fault value when the fault is present

Table 9 shows an example of the fault format

Table 9. LOGGED_FAULT_DETAIL Command Format (UCD9090A)

BYTE NUMBER (Read)	DESCRIPTION
0	BYTE_COUNT = 10
1	Milliseconds (high byte)
2	Milliseconds
3	Milliseconds
4	Milliseconds (low byte)
5	Fault ID + Days (high byte) [(1)]
6	Fault ID + Days
7	Fault ID + Days
8	Fault ID + Days (low byte)
9	Fault Value (low byte)
10	Fault Value (high byte)

(1) Days since 0001-01-01 AD

In the following paragraphs, an example demonstrates how to decode the raw data into a readable fault information. Once the target fault log is selected by the LOGGED_FAULT_DETAIL_INDEX (as described in Section 4.1.2), the LOGGED_FAULT_DETAIL(0xECh) command can be issued to retrieve the raw fault data as shown below.

[St] [AddrW] [A] [W:0xEC] [A] [Sr] [AddrR] [A] [R:0XA][A][R:Data_1] [A] [R:Data_2] [A]..[R:DATA_10][N] [Sp]

4.3 Procedures to Read Complete Fault Log

This section describes the steps to read the complete fault log from UCD9090A. The rest of the UCD90xxx devices follow the same procedures.

1. Read the total available fault log entries with the following command:
   [St] [AddrW] [A] [W:0xEB] [A] [Sr] [AddrR] [A] [R:Data_1] [A] [R:Data_2] [N] [Sp]
2. If the Data_1(total fault log) from step 1 is bigger than 0, the devices contain a valid fault log. Select the first fault log with the following command:
   [St] [AddrW] [A] [W:0xEB] [A] [W:Data_1] [A] [W: fault index = 0] [N] [Sp]
3. Read the fault log detail of the fault log selected in Step 2.
   [St] [AddrW] [A] [W:0xEC] [A] [Sr] [AddrR] [A] [R:0XA][A][R:Data_1] [A] [R:Data_2] [A]..[R:DATA_10][N] [Sp]
4. Repeat Steps 2 and 3 with different fault log indexes until the last fault log is read.

Sending a fault index greater than or equal to the total number of fault long entries is prohibited.

5 Decode the Fault Log Raw Data

Assume that the return data from the LOGGED_FAULT_DETAIL(0xECh) command is 0x0A0368946A8B8B3FF13A26. Fusion GUI reports the following information for the given fault data, as shown in Figure 1.

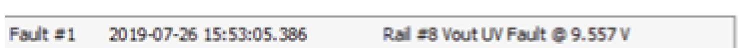

Figure 1. Fault Log Decoded by Fusion GUI

The following sections demonstrate how to decode the raw data into meaningful fault information.

The first byte is fixed at 0x0Ah to tell that there is a total of 10 bytes of data. The remain 10 bytes data can be categorized as shown in Table 10.

Table 10. Fault Log Data Example

Description	Data (Hex)
Milliseconds	0x0368946Ah
Fault ID + Days	0x8B8B3FF1h
Fault Value	0x263Ah

5.1 *Milliseconds Decoding*

Milliseconds are decoded into the following format as shown in Table 11.

Table 11. Fault Log Data Example

Description	Data (Hex)	Data (decimal, ms)	Time Format (HH:MM:SS)
Milliseconds	0x0368946Ah	57185386	15:53:05.386

5.2 *Fault ID + Days Decoding*

The Fault ID + Days has the following data format as shown in Table 12

Table 12. Fault ID + Days (UCD9090A)

Bits	Description
31	Page Specific (1-yes, 0 - No)
30-27	Fault Type
26-23	Page Number
22-0	Days

Table 13. Fault Value

Fault Type	Paged?	Description	Fault Value Units	Data Format
0	No	Reserved		
1	No	System Watchdog Timeout	Not Valid	N/A
2	No	Resequence Error	Not Valid	N/A
3	No	Watchdog Timeout	Not Valid	N/A
4	No	Reserved		
5	No	Reserved		
6	No	Reserved		
7	No	Reserved		
8	No	Fan Fault(1)	RPM	LINEAR11
9	No	GPI Fault(1)	Not Valid	N/A
0	Yes	VOUT_OV Fault	Voltage	LINEAR16
1	Yes	VOUT_UV Fault	Voltage	LINEAR16
2	Yes	TON_MAX Fault	Voltage	LINEAR16
3	Yes	IOUT_OC Fault	Current	LINEAR11
4	Yes	IOUT_UC Fault	Current	LINEAR11
5	Yes	TEMPERATURE_OT Fault	Temperature	LINEAR11
6	Yes	Sequence On Timeout	N/A	Bit Mask(2)
7	Yes	Sequence Off Timeout	N/A	Bit Mask(2)

(1) The Page Number is used to encode which Fan or GPI that the fault information applies to.
(2) Any bit set to 1 corresponds to a page dependency that is not met. The GPI dependencies that are not met are OR'ed into the top 8 bits. So, for example, if page dependencies 2 and 5 are not met and GPI W (bit 4) dependency is not met, the fault value is 0x1024.

Table 14. Relationship Between Page and Rail

Page Number	Output Rail
0	1
1	2
2	3
3	4
…	…

Based on the Table 12, Table 13, and Table 14, the Fault ID + Days can be decoded as shown in Table 15.

Table 15. Fault ID + Days Example

Fields	Data[MSB: LSB]	Descriptions
Fault ID + Days	0x8B8B3FF1	1000 1011 1000 1011 0011 1111 1111 0001b
Paged (1-yes, 0 - No)	Bit[31] = 1b	The fault is a paged related fault.
Fault Type	Bit[30:27] = 0001b	This is a VOUT_UV fault since this is a paged fault, and the fault type is 0x01 based on Table 13.
Page Number	Bit[26:23] = 0111b	The rail #8 has a fault.
Days	Bit[22:0] = 000 1011 0011 1111 1111b = 0x0B3FF1h	The event is detected at 737265 days since 0001-01-01 AD, which is 07/26/2019 Combining the days and milliseconds together, the fault was present at 2019-07-26 15:53:05.386

For the UCD90240/320/320U, the day starts from 2000-01-01 instead of 0001-01-01 AD. Please make adjustments accordingly.

5.3 Fault Value Decoding

The decoding of the fault value is up to the fault type. The following section explains how to decode the fault value based on the according fault type.

5.3.1 Voltage Fault Value Decoding

Table 16. Fault Value Example

Description	Data(Hex)	Data (Decimal)
Fault Value	0x263Ah	9786

For voltage-related faults, the fault voltage is calculated as

$$\text{Voltage} = V \times 2^{X} \quad (2)$$

where

- V is fault value
- X is the 5-bit signed two’s complement integer obtained from the VOUT_MODE command for the fault rail.

The VOUT_MODE(0x20h) is a paged command.

Reading the exponent value from the VOUT_MODE command (0x20) for a given page uses the following commands.

[St][Addrw][A][W:00][A][W:page_index][N][Sp]

[St] [AddrW] [A] [W:0x20] [A] [Sr] [AddrR] [A] [R:Data_1] [N] [Sp]

In this example, the VOUT_MODE of rail #8 is 0x16, therefore the X is -10

Fault Voltage = $9786 \times 2^{-10} = 9.557$ V

5.3.2 Current/Temperature Fault Value Decoding

If the fault type is current or temperature,

the decoded fault = N × 2^x

where

- N is the 11 least significant bits of the fault value
- X is the 5-bit signed two's complement integer obtained from the five most significant bits of the fault value. (3)

For example, if the fault value is 0xEAD1, it is an OVER_TEMPERATURE fault. The fifth most significant bit is 1 1101b, therefore the X is -3 and the eleventh (least significant) bit is 010 1101 0001b = 0x2D1h. N is 0x2D1, so the temperature is 0x2D1 × 2^{-3} = 90.1C°

6 Command to Clear Fault Log

The application can issue 0xEAh (LOGGED_FAULTS) command to clear the fault log. The method is to write a block whose data bytes are all 0x00. The number of writing data bytes vary upon device. Refer to Table 17.

Table 17. Logged Fault Command Size

Device	UCD9090/UCD9090A	UCD90120A	UCD90124A	UCD90160/UCD90160A	UCD90240	UCD90320/UCD90320U
Size	12	14	15	18	28	37

Take UCD9090A as an example to clear the fault log.

[St] [AddrW] [A] [W:0xEA] [A] [W:0XC][A][W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00] [A] [W:00][N] [Sp]

7 References

- Texas Instruments, *UCD90xxx Sequencer and System Health Controller PMBus Command Reference User's Guide*
- Texas Instruments, *UCD90320 Sequencer and System Health Controller PMBus Command Reference User's Guide*
- Texas Instruments, *UCD90320U Sequencer and System Health Controller PMBus Command Reference User's Guide*
- *The PMBus Power System Management Protocol Specification Part II - Command Language, Revision 1.1,* 5 February 2007, available from www.pmbus.org.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (September 2019) to A Revision	**Page**
• Corrected typo that read '727265'. Now reads '737265'.	7