Application Report
SLVAE90–March 2019

UCD90160A/UCD9090A PMBus Command Migrating Guide

1 Trademarks

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

2 Introduction

TI's UCD9090A and UCD90160A power-supply sequencer and monitor with Advanced Configuration and Power Interface (ACPI) support can control up to 10 and 16 voltage rails (respectively), ensure correct power sequences during normal and fault conditions, and include a dedicated fault pin to easily cascade multiple devices. The A revisions of the devices are an upgrade to the UCD9090 and UCD90160 (listed as not recommended for new designs [NRND]). This document describes the PMBus™ Command differences between the UCD9090/UCD90160 and UCD9090A/UCD90160A to help easily migrate the application.

3 Changes of PMBus Commands

In the UCD9090A/UCD90160A, three existing PMBUS Commands are modified to support new features: SEQ_CONFIG(0xF6h), GPI_CONFIG(0xF9h), and MISC_CONFIG(0xFCh) in comparison with UCD9090/UCD90160 respectively. The rest commands have the same definitions as their predecessors defined in the *UCD90xxx Sequencer and System Health Controller PMBus™ Command Reference User's Guide*.

3.1 *(0xF6h)SEQ_CONFIG(MFR_SPECIFIC_38)*

In order to support long sequence timeout period and GPO sequence on/off dependencies, the SEQ_CONFIG command definitions are changed and indicated with the highlights shown in Table 1. The byte count of the command is increased from 12 bytes to 16 bytes. The sequence on/off timeout is incremented by one byte respectively. New fields (GPO sequence on/off dependency Mask) are added with 1 byte each.

Table 1. SEQ_CONFIG

BYTE NUMBER (WRITE)	BYTE NUMBER (READ)	PAYLOAD INDEX	DESCRIPTION
0			CMD = F9
1	0		BYTE_COUNT = 16
2	1	0	Enable pin configuration
3	2	1	GPI sequence on dependency mask
4	3	2	GPI sequence off dependency mask
5	4	3	Sequencing timeout configuration
6	5	4	Sequencing on timeout (high byte)
7	6	5	Sequencing on timeout (low byte)
8	7	6	Sequencing off timeout (high byte)
9	8	7	Sequencing off timeout (low byte)
10	9	8	Page sequencing on dependency mask (high byte)
11	10	9	Page sequencing on dependency (low byte)
12	11	10	Page sequencing off dependency mask (high byte)

Table 1. SEQ_CONFIG (continued)

BYTE NUMBER (WRITE)	BYTE NUMBER (READ)	PAYLOAD INDEX	DESCRIPTION
13	12	11	Page sequencing off dependency mask (low byte)
14	13	12	Fault slaves mask (high byte)
15	14	13	Fault slaves mask (low byte)
16	15	14	GPO sequence on dependency mask
17	16	15	GPO sequence off dependency mask

3.1.1 Sequence On/Off Timeout

The 16 bits are separated into two fields as shown in Table 2. Bits 14 and 15 are the increment field and bits 0 to 13 are the multiplier (0x00 to 0x3FFF). The two are multiplied together to derive the time. A multiplier of 0 results in a time of 0 regardless of the increment value.

Table 2. 16-bit Time Encoding

INCREMENT ENCODING	INCREMENT (ms)	RESULTING RANGE
b'00	1	1 to 16384 ms
b'01	8	8 to 131.72 s
b'10	64	64 ms to 1048.576 s
b'11	512	512 ms to 8388.608 s

3.1.2 GPO Sequence On/Off Dependency Mask

Each page has its own GPO Sequence On/Off Dependency Mask, whose bits are defined as the following:

BIT	7	6	5	4	3	2	1	0
PURPOSE	GPO8	GPO7	GPO6	GPO5	GPO4	GPO3	GPO2	GPO1

The logical state is selected by the GPO, not the actual pin output of the GPO used as dependency. Only the logical state of the first eight GPO can be used as Sequence On/Off Dependency Mask. The enable pin for the page is not de-asserted until all of the logical states selected by the GPO bits are FALSE, vice versa. Once the enable pin is either asserted or de-asserted, the logical state of these GPO has no effect on the state of the enable pin.

3.2 (0xFCh)GPI_CONFIG(MFR_SPECIFIC_41)

In order to support GPI fault response, fault pin, and GPI debug, the GPI_CONFIG command definitions are changed and indicated with the highlights shown in Table 3. The byte count of the command is increased from 13 bytes to 31 bytes. The GPI fault response takes 6 bytes, the fault pin takes 8 bytes, the GPI debug takes 1 bytes, and the cold boot takes 3 bytes.

Table 3. GPI_CONFIG

BYTE NUMBER (WRITE)	BYTE NUMBER (READ)	PAYLOAD INDEX	DESCRIPTION
0			CMD = F9
1	0		BYTE_COUNT = 31
2	1	0	Input Pin S Configuration
3	2	1	Input Pin T Configuration
4	3	2	Input Pin U Configuration
5	4	3	Input Pin V Configuration
6	5	4	Input Pin W Configuration

Table 3. GPI_CONFIG (continued)

BYTE NUMBER (WRITE)	BYTE NUMBER (READ)	PAYLOAD INDEX	DESCRIPTION
7	6	5	Input Pin X Configuration
8	7	6	Input Pin Y Configuration
9	8	7	Input Pin Z Configuration
10	9	8	Fault Enable Flags
11	10	9	Latched Statuses Clear Pin Selection
12	11	10	“Margin Enable” (MRG_EN) Pin Selection
13	12	11	“Margin Low/Not-High” (MRG_LOW_nHIGH) Pin Selection
14	13	12	Fans Installed Pin Selection
15	14	13	GPI Fault response Pin Selection
16	15	14	Fault Responses Byte
17	16	15	PAGE Mask for GPI fault (high Byte)
18	17	16	PAGE Mask for GPI fault (low Byte)
19	18	17	Max Glitch time for GPI (high byte)
20	19	18	Max Glitch time for GPI (low byte)
21	20	19	Fault Pin Function Pin Selection
22	21	20	Fault Responses Byte for Fault Pin
23	22	21	PAGE Mask for Fault pin response(High Byte)
24	23	22	PAGE Mask for Fault pin response(Low Byte)
25	24	23	Max Glitch time for Fault pin (high byte)
26	25	24	Max Glitch time for Fault pin (low byte)
27	26	25	Page Mask for Fault pin output (high Byte)
28	27	26	Page Mask for Fault pin output (low Byte)
29	28	27	Debug Mode Pin Selection
30	29	28	Cold Boot Mode Pin Selection
31	30	29	Cold Boot Timeout
32	31	30	Normal Boot Start Delay

3.2.1 GPI Fault Response

3.2.1.1 GPI Fault Response Pin Selection

The GPI fault response can be enabled with the input pin identified by this byte if the corresponding fault enable bit is also set. The pin is ignored if the value of this byte is 0. When this pin is changed from asserted to de-asserted, the corresponding fault response is performed.

3.2.1.2 Fault Response Byte

This byte defines the fault response for the given GPI; the definition is the same as command FAULT_RESPONSES (0xE9h). It shares the same retry interval defined in the FAULT_RESPONSES command.

3.2.1.3 Page Mask for GPI Fault

The page mask is made up of two bytes whose bits are defined as follows:

BIT[1]	15	14	13	12	11	10	9	8
PURPOSE	PAGE15	PAGE14	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8
BIT	7	6	5	4	3	2	1	0

[1] This table assumes that the device supports 16 rails (0 to 15). For a given device, this may not be the case. The interpretation of this information should be adjusted for the correct number of rails.

BIT[1]	15	14	13	12	11	10	9	8
PURPOSE	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

Each bit selects which pages respond to GPI fault. Setting the bit associated with a given page to 1 enables fault response for that page. If the bit associated with a page is set to 0, the page does not respond to the GPI fault.

3.2.1.4 Max Glitch Time for GPI

The value in these bytes is multiplied by 500 µs to get the max glitch of GPI input if the GPI is used for fault response.

3.2.2 Fault Pin Config with Fault Response

3.2.2.1 Fault Pin Function Pin Selection

The fault pin can be enabled with the input pin identified by this byte if the corresponding fault enable bit is also set. The pin is ignored if the value of the byte is 0. When the fault pin is on, the pin is active as an input to monitor the external event. If there are any internal rail faults, the device pulls the pin low to notify the other devices. When the fault is gone, the device releases the pin and continues the monitoring function on the assigned pin.

3.2.2.2 Fault Reponse Byte for Fault Pin

This byte defines the fault response for the given GPI, the definition is the same as command FAULT_RESPONSES(0xE9h). It shares the same retry interval defined in the FAULT_RESPONSES command.

3.2.2.3 PAGE Mask for Fault Pin Response

This page mask has the exact same function and format as the Page Mask for GPI defined in Section 3.2.1.3.

3.2.2.4 Max Glitch Time for Fault Pin

This glitch time has the exact same function and format as the Max Glitch for GPI defined in Section 3.2.1.4.

3.2.2.5 Page Mask for Fault Pin Output

The page mask is made up of two bytes whose bits are defined as follows:

BIT[1]	15	14	13	12	11	10	9	8
PURPOSE	PAGE15	PAGE14	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PURPOSE	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

[1] This table assumes that the device supports 16 rails (0 to 15). For a given device, this may not be the case. The interpretation of this information should be adjusted for the correct number of rails.

Each bit selects which pages pull the fault pin low when it has fault. Setting the bit associated with a given page to 1 enables fault pin output for that page. If the bit associated with a page is set to 0, the page does not pull the fault pin low when it has fault.

The following faults have impacts on fault pin output:

RESEQUENCE_ ERROR	SEQ_ON_TIMEOUT	SEQ_OFF_TIMEOUT	OT_FAULT	IOUT_OC_FAULT
IOUT_UC_FAULT	VOUT_UV_FAULT	VOUT_OV_FAULT	TON_MAX_FAULT	

3.2.3 Debug Mode Pin Selection

The sequence debug function can be enabled with the input pin identified by this byte. The pin is ignored if the value of the byte is 0. Under Debug Mode, the device does not activate PMBus alert pin for any faults or warnings, respond for any fault responses, or log any faults. This function is mainly designed for debug purpose and it is not recommended in the final production.

The following faults/warnings are impacted by debug mode:

VOUT_OV_FAULT	TON_MAX	IOUT_UC	SYSTEM_WATCHDOG_TIMEOUT
VOUT_OV_WARNING	TOFF_MAX Warning	OT_FAULT	INVALID_LOG
VOUT_UV_FAULT	IOUT_OC_FAULT	OT_WARNING	LOG_FAULT_DETAIL_FULL
VOUT_UV_WARNING	IOUT_OC_WARNING	PKGID_MISMATCH	RESEQUENCE_ERROR
SEQ_ON_TIMEOUT	SEQ_OFF_TIMEOUT	SLAVE_FAULT	HARDCODE_PARMS
All GPI de-asserted			

When the debug mode is on, the rail sequence on/off dependency conditions are ignored. As soon as the sequence on/off timeout is expired, the rails are sequenced on or off accordingly, regardless of the timeout action. If the sequence on/off timeout value is set to 0, the rails are sequenced on or off immediately.

3.2.4 Cold Boot Mode Pin Selection

Cold boot mode is a feature used to heat up the system by turning on some particular rails for certain amounts of time when the system is under an extreme code temperature. The UCD device is communicated with the system via particular GPI (thermal state GPI), which is output from a thermal device. Cold boot-up mode is only entered once per UCD reset. There is no system watchdog reset during the cold boot-up mode. For the rest functions, the device behaves the same when under cold boot-up mode.

Cold Boot mode can be enabled with the pin identified by this byte. The pin is ignored if the value of the byte is 0. When this byte is set to non-zero, the assigned GPI is defined as thermal state GPI. During the UCD9090A boot-up, it reads the thermal state GPI to determine whether it should start cold boot-up or not. When the input of thermal state GPI is DE-ASSERTED, the device enters cold boot mode, otherwise the device enters normal mode. The device logs a GPI fault if the thermal state GPI is at DE_ASSERTED state after the device is out of reset if the GPI fault log enable bit is set. The following changes on the thermal state GPI shall not introduce any logging. Only one GPI can be assigned for this function and once it is assigned, it cannot be used for any other GPI functions.

The rails used in the cold boot-up mode are configurable. For those rails with Sequence On Dependency on the thermal state GPI, they (non-cold boot-up rails) are not powered-up during the cold boot-up since the dependency is not met. But non-cold boot-up rails are powered-on under normal mode since thermal state GPI is treated as ASSERTED when cold boot-up mode is over. For those rails without sequence on dependency on the thermal state GPI, they (cold boot-up rails) are powered-on on both cold boot-up and normal mode. It is the responsibility of the application to set the proper ON_OFF_CONFIG for those cold boot-up rails. Cold boot-up rails are not powered-on if the ON_OFF_CONFIG settings are not met under cold boot mode.

Once the cold boot-up is over, the device stops monitoring the thermal state GPI, which is treated as ASSERTED afterward. The device shuts down all cold boot-up rails which have the EN pin associated immediately. For those cold boot-up rails without the EN pin associated, the device does not shut them down. When all cold boot-up rails with EN pin are below power good off threshold, the device waits for the programmable delay to re-power on all rails. The On-Off-Config conditions and turn on dependencies must be met for the rails to be on properly. The following logic applies to cold boot mode operation:

```
If system temperature is < threshold deg C (Thermal State GPI)
    If yes (DE_ASSERTED), then:
        Log GPI fault
        Start Cold Boot Timeout
        No System Watchdog output
        Ramp up the power supplies based on on_off_config
        Wait for thermal state GPI ASSERTED OR "Cod boot Timeout expire"
        Disable the thermostat input listening mode
        Force to shutdown down all cold boot rails with EN control immediately
        Wait for all cold boot rails with EN control below POWER_GOOD_OFF
        Start and wait Start Delay after cold boot
Disable the thermostat input listening mode
Treat Thermal State GPI ASSERTED
Ramp up power supplies based on on_off_config
```

3.2.5 Cold Boot Mode Timeout

Cold boot mode timeout is only valid if the cold boot mode pin selection is set to a valid number. This byte is used to tell how long the device shall stay at the cold boot-up before it stops listening to the thermal state GPI and shuts down all rails. If this byte is set to 0, the device stays at the cold boot-up mode until the thermal state GPI is ASSERTED. The range is from 0–255 minutes with 1 minute steps. Cold boot mode timeout should never be reset unless the UCD is reset.

3.2.6 Normal Boot Start Delay

This byte is only valid if the cold boot mode pin selection is set to a valid number. NORMAL_BOOT_START_DELAY is used to tell how long the UCD9090A should wait to ramp up the powers after the cold boot up is over. The start delay does not start until all cold-boot rails with EN pins power down during the cold boot mode are at POWER_GOOD_OFF. If NORMAL_BOOT_START_DELAY is set to 0, the UCD should turn on rails as fast as it can after all rails are at POWER_GOOD_OFF state. This byte is formatted according the 8-bit time encoding defined in Section 2.5.

3.3 *(0xFCh)MISC_CONFIG(MFR_SPECIFIC_44)*

In order to enhance the resequencing feature, MISC_CONFIG command definitions are changed and indicated with the highlights shown in Table 4. The byte count of the command is increased from 2 bytes to 4 bytes.

In the MISC_CONFIG shown in Table 5, the reserved bit is defined as new function: Flash_Log_Disable.

Table 4. MISC_CONFIG

BYTE NUMBER (WRITE)	BYTE NUMBER (READ)	PAYLOAD INDEX	DESCRIPTION
0			CMD = FC
1	0		BYTE_COUNT = 4
2	1	0	Miscellaneous Configuration Byte
3	2	1	Time between resequences
4	3	2	Resequences_rail_mask (high byte)
5	4	3	Resequences_rail_mask (low byte)

Table 5. Miscellaneous Configuration Byte

BIT(S)	NAME	DESCRIPTION
7	Resequence Continuously	When this bit is set, there is no limit to the number of times that the device attempts to resequence. The “Max Resequences” value does not apply.
6	Resequence Abort	If a rail fails to turn off during resequencing, stop the resequencing operation.
5:4	Max Resequences	The maximum number of times to attempt to resequence: b’00 – 1 time b’01 – 2 times b’10 – 3 times b’11 – 4 times
3	Flash_Log_Disable	When this bit is set, log(Fault, Peak and RTC) is only stored into the volatile memory and the non-volatile memory is not updated. When this bit is clear, log is stored into both volatile and non-volatile memory. Device Reset is required after the bit is toggled.
2	Enable Log FIFO	When this bit is set, all or part of the LOGGED_FAULT_DETAIL is treated as a FIFO, depending on the “FIFO Entire Log” bit.

Table 5. Miscellaneous Configuration Byte (continued)

BIT(S)	NAME	DESCRIPTION
1	FIFO Entire Log	When this bit is set, all or part of the LOGGED_FAULT_DETAIL is treated as a FIFO, depending on the “FIFO Entire Log” bit.
0	Brownout Enable	Enables the brownout function which allows information (faults, peaks, run-time clock, etc) to be saved to flash only when the device is powered down. When this is not enabled, firmware must periodically update this information in flash. See the data sheet for details regarding the external circuitry required to support this feature.

3.3.1 Resquences_rail_mask

The page mask is made up of two bytes whose bits are defined as follows:

BIT[1]	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PURPOSE	PAGE15	PAGE14	PAGE13	PAGE12	PAGE11	PAGE10	PAGE9	PAGE8
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PURPOSE	PAGE7	PAGE6	PAGE5	PAGE4	PAGE3	PAGE2	PAGE1	PAGE0

[1] This table assumes that the device supports 16 rails (0 to 15). For a given device, this may not be the case. The interpretation of this information should be adjusted for the correct number of rails.

Each bit selects which pages should be not checked its POWER_GOOD_OFF and TOFF_WARN status when performing resequences. When the corresponding page bit is set to 1, the resequencing engine does not check its power_not_good and TOFF_WARN status. When the corresponding page bit is set to 0, the resequence engine check its power_good_off and TOFF_WARN status before starting the resequencing.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265