IR38060/2/3/4 PMBUS COMMAND SET

OBJECTIVE

This document is used to list and describe the PMBus commands supported by the Manhattan platform in general (IR38060, IR38062, IR38063, IR38064). The intention of the document is to provide a blueprint for the datasheet and to serve as a guide for system engineers, FPGA engineers and test engineers to do system validation and test.

SCOPE

This document describes, in a very detailed manner, the PMBus commands supported. Each command is associated with a command code, a description and a set/range of supported/legal values. Also included is a description of the actions to be taken in the event of illegal/unsupported data values.

REVISION

Date	Version	Comment
5/9/2014	1.0	1st release
10/2/2014	1.1	Reformat, correct typos and errors in some command descriptions
5/27/2015	1.2	Added timing diagram for Ton_Max_Fault_Limit
10/21/2015	1.3	Marked TOFF_FALL as not supported, updated FREQUENCY_SWITCH to show finer resolution; Corrected MFR_ID order
02/14/2020	1.4	Updated the IOUT_OC_FAULT Response, table 14, with the supported items

PMBus COMMANDS

Command Code	Command Name	SMBus transaction	No. of bytes	Store/Restore in MTP
01h	OPERATION	R/W Byte	1	✓
02h	ON_OFF_CONFIG	R/W Byte	1	✓
03h	CLEAR_FAULTS	Send Byte	0	n/a
10h	WRITE_PROTECT	R/W Byte	1	
15h	STORE_USER_ALL	Send Byte	0	n/a
16h	RESTORE_USER_ALL	Send Byte	0	n/a
19h	CAPABILITY	Read Byte	1	n/a
1Bh	SMBALERT_MASK	Write word/Block read Process call	2	✓
21h	VOUT_COMMAND	R/W Word	2	✓
22h	VOUT_TRIM	R/W Word	2	✓
24h	VOUT_MAX	R/W Word	2	✓
25h	VOUT_MARGIN_HIGH	R/W Word	2	✓
26h	VOUT_MARGIN_LOW	R/W Word	2	✓
27h	VOUT_TRANSITION_RATE	R/W Word	2	✓
29h	VOUT_SCALE_LOOP	R/W Word	2	✓
33h	FREQUENCY_SWITCH	R/W Word	2	✓
35h	VIN_ON	R/W Word	2	✓
36h	VIN_OFF	R/W Word	2	✓
39h	IOUT_CAL_OFFSET	R/W Word	2	✓
40h	VOUT_OV_FAULT_LIMIT	R/W Word	2	✓
41h	VOUT_OV_FAULT_RESPONSE	R/W Byte	1	✓
42h	VOUT_OV_WARN_LIMIT	R/W Word	2	✓
43h	VOUT_UV_WARN_LIMIT	R/W Word	2	✓
44h	VOUT_UV_FAULT_LIMIT	R/W Word	2	✓
45h	VOUT_UV_FAULT_RESPONSE	R/W Byte	1	✓
46h	IOUT_OC_FAULT_LIMIT	R/W Word	2	✓
47h	IOUT_OC_FAULT_RESPONSE	R/W Byte	1	✓
4Ah	IOUT_OC_WARN_LIMIT	R/W Word	2	✓
4Fh	OT_FAULT_LIMIT	R/W Word	2	✓
50h	OT_FAULT_RESPONSE	R/W Byte	1	✓
51h	OT_WARN_LIMIT	R/W Word	2	✓
55h	VIN_OV_FAULT_LIMIT	R/W Word	2	✓
56h	VIN_OV_FAULT_RESPONSE	R/W Byte	1	✓
58h	VIN_UV_WARN_LIMIT	R/W Word	2	✓
5Eh	POWER_GOOD_ON	R/W Word	2	✓

5Fh	POWER_GOOD_OFF	R/W Word	2	✓
60h	TON_DELAY	R/W Word	2	✓
61h	TON_RISE	R/W Word	2	✓
62h	TON_MAX_FAULT_LIMIT	R/W Word	2	✓
63h	TON_MAX_FAULT_RESPONSE	R/W Byte	1	✓
64h	TOFF_DELAY	R/W Word	2	✓
65h	TOFF_FALL (not supported)	R/W Word	2	✓
78h	STATUS BYTE	Read Byte	1	n/a
79h	STATUS WORD	Read Word	2	n/a
7Ah	STATUS_VOUT	Read Byte	1	n/a
7Bh	STATUS_IOUT	Read Byte	1	n/a
7Ch	STATUS_INPUT	Read Byte	1	n/a
7Dh	STATUS_TEMPERATURE	Read Byte	1	n/a
7Eh	STATUS_CML	Read Byte	1	n/a
88h	READ_VIN	Read Word	2	n/a
8Bh	READ_VOUT	Read Word	2	n/a
8Ch	READ_IOUT	Read Word	2	n/a
8Dh	READ_TEMPERATURE_1	Read Word	2	n/a
96h	READ_POUT	Read Word	2	n/a
98h	PMBUS_REVISION	Read Byte	1	n/a
99h	MFR_ID	Block Read/Write	Variable	✓
9Ah	MFR_MODEL	Block Read/Write	Variable	✓
9Bh	MFR_REVISION	Block Read/Write	Variable	✓
ADh	IC_DEVICE_ID	Block Read	Variable	n/a
AEh	IC_DEVICE_REV	Block Read	Variable	n/a
D0h	MFR_READ_REG	Custom	2	n/a
D1h	MFR_WRITE_REG	Custom	2	n/a
D6h	MFR_I2C_address	R/W Word	1	✓
D8h	MFR_TPGDLY	R/W Word	2	✓
D9h	MFR_FCCM	R/W Byte	1	✓
DBh	MFR_VOUT_PEAK	Read Word	2	n/a
DCh	MFR_IOUT_PEAK	Read Word	2	n/a
DDh	MFR_TEMPERATURE_PEAK	Read Word	2	n/a

NON_STANDARD COMMANDS/REGISTERS

Register Address	Function	SMBus transaction	No. of bytes	Store/Restore in MTP
20	PMBus address	Process Call	1	✓
72	Offset for Vout Reporting	Process Call	1	✓

OPERATION (01H)

The OPERATION command is used to turn the device on and off in conjunction with the input from the Enable pin. It is also used to cause the device to set the output voltage to the upper or lower MARGIN VOLTAGEs. The device stays in the commanded operating mode until a subsequent OPERATION command or change in the state of the Enable pin instructs the device to change to another mode.

The contents of the data byte are shown below in Table 1.

Bits [7:6] determine how the device responds when commanded to turn the output off via the OPERATION command. If bits [7:6] are 00b, then the device turns off immediately and ignores any programmed turn-off delay and fall time. If bits [7:6] are 01b, the device powers down according to the programmed turn-off delay and fall time.

Any value not shown in the table is invalid or unsupported data, and will cause a CML fault. Bits [1:0] are reserved.

In the table below, “Act On Fault” means that if an output overvoltage warning or output overvoltage fault is detected when the output is margined high, the unit treats this as a warning or fault and responds as programmed by the warning limit or fault response command. Similarly, if an output undervoltage warning or output undervoltage fault is detected when the output is margined low, the unit treats this as a warning or fault and responds as programmed by the warning limit or fault response command.

“Ignore Fault” means that if an output overvoltage warning or output overvoltage fault is detected when the output is margined high, the unit ignores the condition and continues to operate without interruption or notification that a fault or warning condition has occurred. Similarly, if an output under-voltage warning or output under-voltage fault is detected when the output is margined low, the unit ignores the condition and continues to operate without interruption or notification that a fault or warning condition has occurred.

Command	**OPERATION (01h)**							
Format	**Unsigned Binary**							
Bit Position	7	6	5	4	3	2	1	0
Default	1	0	0	0	0	0	0	0

TABLE 1

Bits [7:6]	Bits [5:4]	Bits [3:2]	Bits [1:0]	Unit On Or Off	Margin State
00	XX	XX	XX	Immediate Off (No Sequencing)	N/A
01	XX	XX	XX	Soft Off (With Sequencing)	N/A
10	00	XX	XX	On	Off
10	01	01	XX	On	Margin Low (Ignore Fault)
10	01	10	XX	On	Margin Low (Act On Fault)
10	10	01	XX	On	Margin High (Ignore Fault)
10	10	10	XX	On	Margin High (Act On Fault)

For ready reference, some data bytes for the OPERATION command are given below in Table 2

TABLE 2

Desired operation	Data byte
ON without margining (Default)	80h
ON with value loaded from VOUT_MARGIN_LOW, ignoring fault	94h
ON with value loaded from VOUT_MARGIN_LOW, acting on fault	98h
ON with value loaded from VOUT_MARGIN_HIGH, ignoring fault	A4h
ON with value loaded from VOUT_MARGIN_HIGH, acting on fault	A8h
Immediate OFF	00h
Soft OFF	40h

If the device receives unsupported data not listed in Table 1, the device will

- NACK the unsupported data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register and
- Notify the host by pulling down SALERT#

ON_OFF_CONFIG (02H)

The ON_OFF_CONFIG command configures the combination of Enable pin input and serial bus commands needed to turn the device on and off. This includes how the device responds when power is applied.

The default is 1Fh and is shown below:

Command	ON_OFF_CONFIG (02h)							
Format	Unsigned Binary							
Bit Position	7	6	5	4	3	2	1	0
Default	0	0	0	1	1	1	1	1

The details of the ON_OFF_CONFIG data byte are shown below in Table 3

TABLE 3

Bit Number	Purpose	Bit Value	Meaning
[7:5]		000	Reserved For Future Use
4	Sets the default to either operate any time power is present or for the on/off to be controlled by Enable pin and serial bus commands	0	Power conversion will start anytime PVin and Vcc are up without needing the Enable pin to be active and without being commanded to do so by the Operation command.
		1	Power conversion does not start until commanded by the Enable pin and OPERATION command (as programmed in bits [3:0]).

3	Controls how the unit responds to commands received via the serial bus	0	The device ignores the on/off portion (bit[7]) of the OPERATION command from serial bus
		1	To start, the device requires that that the on/off portion (bit[7]) of the OPERATION command is instructing the unit to run. Depending on bit [2], the device may also require the Enable pin to be asserted for the unit to start power conversion
2	Controls how the unit responds to the Enable pin	0	Device ignores the Enable pin (on/off controlled only the OPERATION command)
		1	Device requires the Enable pin to be asserted to start power conversion. Depending on bit [3], the OPERATION command may also be required to instruct the device to start before the output is energized.
1	Polarity of the Enable pin	1	Active high (Pull high to start the unit)
0	Enable pin action when commanding the unit to turn off	0	Use the programmed turn off delay TOFF_DELAY and fall time TOFF_FALL
		1	Turn off the output and stop transferring energy to the output as fast as possible.

If the device receives unsupported data (bits [7:5] NOT 000 or bit 1=0) not listed in Table 3, the device will

- NACK the unsupported data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

CLEAR_FAULTS (03H)

The CLEAR_FAULTS command is used to clear any fault bits that have been set. This command clears all bits in all status registers simultaneously. At the same time, the device de-asserts its SALERT# signal output if asserted.

The CLEAR_FAULTS does not cause a device that has latched off for a fault condition to restart. To restart a device that has latched OFF because of a fault

- The device must be commanded to turn off and on again by the action of the Enable Pin and/or the OPERATION command
- Vcc is cycled.

If the fault is still present when the bit is cleared, the fault bit will immediately be set again and the host notified by the usual means of asserting SALERT#.

This command is write only. There is no data byte for this command.

WRITE_PROTECT (10H)

The WRITE_PROTECT command is used to control writing to the PMBus device. The intent of this command is to provide protection against accidental changes. This command is not intended to provide protection against deliberate or malicious changes to a device's configuration or operation.

All supported commands may have their parameters read, regardless of the WRITE_PROTECT settings.

This command has one data byte as shown in table below: The default is 00h.

TABLE 4

Data Byte	Action
80	Disable all writes except to the WRITE_PROTECT command
40	Disable all writes except to the WRITE_PROTECT, OPERATION and PAGE commands
20	Disable all writes except to the WRITE_PROTECT, OPERATION, ON_OFF_CONFIG and VOUT_COMMAND commands
00	Enable writes to all commands.

Any attempt to set a value other than that in the table above will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

Note that if an attempt is made to write invalid or unsupported data to a Write Protected command, it will still be rejected and result in an invalid data CML fault.

CAPABILITY (19H)

This command provides a way for a host system to determine some key capabilities of a PMBus device. There is one data byte formatted as shown in the table below. This command is read only.

TABLE 5

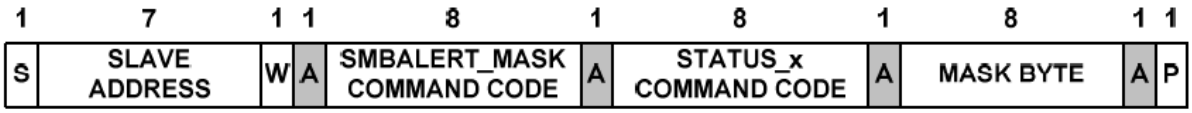
Bits	Description	Value	Meaning
7	Packet Error Checking	0	Packet Error Checking not supported
		1	Packet Error Checking is supported
6:5	Maximum Bus Speed	00	Maximum supported bus speed is 100 kHz
		01	Maximum supported bus speed is 400 kHz
		10	Reserved
		11	Reserved
4	SMBALERT#	0	The device does not have a SMBALERT# pin and does not support the SMBus Alert Response protocol
		1	The device does have a SMBALERT# pin and does support the SMBus Alert Response protocol
3:0	Reserved	X	Reserved

For the Manhattan platform, this command will return B0, indicating that Manhattan supports PEC, a maximum bus speed of 400 kHz and supports the SMBus Alert Response protocol through the SALERT# pin.

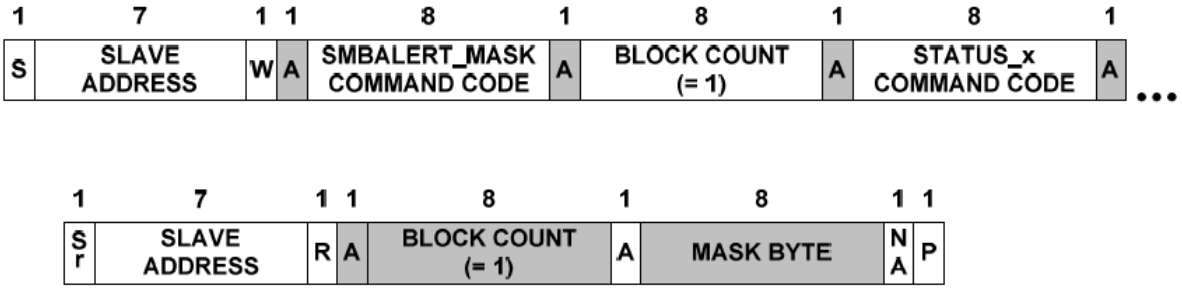
SMBALERT_MASK (1BH)

The SMBALERT_MASK command may be used to prevent a warning or fault condition from asserting the SALERT# signal.

The command format used to block a status bit or bits from causing the SALERT# signal to be asserted uses a Write Word transaction and is shown below. The bits in the mask byte align with the bits in the corresponding status register. For example if the STATUS_VOUT register reads C0, indicating a Vout OV warning as well as fault, the SMBALERT_MASK command code would have to be sent with the mask byte C0 also, in order to block the assertion of the SALERT# signal. If 80h is used as the mask byte, it will prevent an OV fault from asserting SALERT# but an OV warning can still cause SALERT# assertion. Conversely, using 40 h as the mask byte will prevent an OV warning from asserting SALERT#, an OV Fault can still cause SALERT# assertion). The bit maps for each warning or fault condition are found in the individual STATUS descriptions, commands 78h through 80h. The STATUS_X_COMMAND code is sent in the low byte, and the bits to be masked are sent with the high byte.



In order to retrieve the SMBALERT_MASK setting for a given status register, the Block Write Block Read Process call transaction is used with 2 bytes and is shown below.



The default value of all masks are inactive. i.e. faults are not masked.

STORE_USER_ALL (15H)

The STORE_USER_ALL command allows the user to copy the entire contents of the user section of the Operating Memory registers to the matching locations in the non-volatile multi-time programmable memory (MTP). Any items in Operating Memory that do not have matching locations in the MTP User Store are ignored.

It is permitted to use the STORE_USER_ALL command while the device is operating.

However, the device may be unresponsive during the copy operation; in such an event, the BUSY bit in STATUS_BYTE will be set, and it is strongly recommended that the user should pause reading from or writing to the device until the BUSY bit is reset, indicating completion of the copy operation.

This command is Write only and has no data bytes.

RESTORE_USER_ALL (16H)

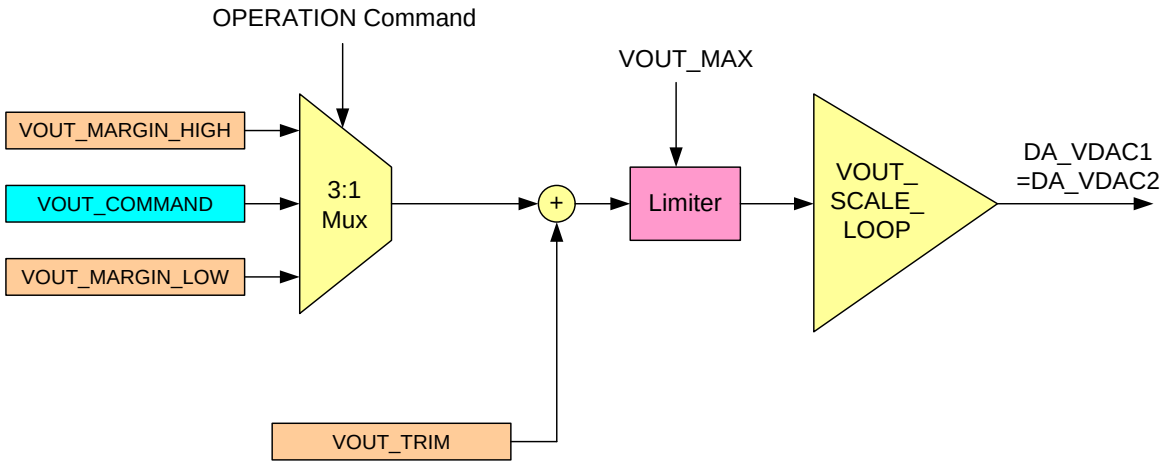
The RESTORE_USER_ALL command allows the user to load the entire contents of the user section of the non-volatile multi-time programmable memory (MTP) into the corresponding locations in the user section of the Operating memory. Any items in MTP User Store that do not have matching locations in in Operating Memory are ignored.

It is permitted, but not recommended to use the RESTORE_USER_ALL command while the device is operating. The device may be unresponsive during the copy operation; in such an event, the BUSY bit in STATUS_BYTE will be set, and it is strongly recommended that the user should pause reading from or writing to the device until the BUSY bit is reset, indicating completion of the copy operation.

This command is Write only and has no data bytes.

VOUT RELATED COMMANDS

The figure below shows a conceptual diagram of how Vout related commands are applied. In this section we will examine the details of each of these commands. It must be noted that the Manhattan platform does not support the VOUT_MODE command; the only format it supports is Linear format, and for Vout related commands the exponent is fixed at -8. Also, it must be noted that while an exponent of -8 implies a resolution of 3.9mV for Vout related commands, the finest native resolution for the Manhattan platform is 5 mV, meaning that intermediate commanded values are truncated to the closest lower 5mV.



VOUT_COMMAND (21H)

This is a command that is used to set the output voltage when the Operation command is set to ON without margining. It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, an exponent of -8 is implied. The default is 0080h, which is 0.5V. The range and resolution depend on VOUT_SCALE_LOOP as follows

TABLE 6

VOUT_SCALE_LOOP	Resolution	Range
E808=1	5 mV	0-2.555V
E804=0.5	10 mV	0-5.11V
E802=0.25	20 mV	0-10.22V
E801=0.125	40 mV	0-20.44V

Command	VOUT_COMMAND (21h)
Format	Unsigned Binary

Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

During startup and shutdown the output voltage ramps up and ramps down to and from the voltage defined by VOUT_COMMAND in a time defined by TON_RISE and TOFF_FALL respectively. However, if this command is issued when the output is up and in steady state, the time to transition to the new commanded voltage is dictated by VOUT_TRANSITION_RATE.

VOUT_MARGIN_HIGH (25H)

This is a command that is used to set the output voltage when the Operation command is set to Margin High. It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 008Dh, which is 0.55V.

The range and resolution depend on VOUT_SCALE_LOOP as follows

TABLE 7

VOUT_SCALE_LOOP	Resolution	Range
E808=1	5 mV	0-2.555V
E804=0.5	10 mV	0-5.11V
E802=0.25	20 mV	0-10.22V
E801=0.125	40 mV	0-20.44V

Command	VOUT_MARGIN_HIGH (25h)															
Format	Unsigned Binary															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1

During startup and shutdown the output voltage ramps up and ramps down to and from the voltage defined by VOUT_MARGIN_HIGH in a time defined by (VOUT_MARGIN_HIGH*TON_RISE) /VOUT_COMMAND and (VOUT_MARGIN_HIGH*TON_FALL)/VOUT_COMMAND respectively. However, if this command is issued when the output is up and in steady state, the time to transition to the new commanded voltage (provided the OPERATION command is set to Margin High) is dictated by VOUT_TRANSITION_RATE.

VOUT_MARGIN_LOW (26H)

This is a command that is used to set the output voltage when the Operation command is set to Margin low. It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0074h, which is 0.45V. The range and resolution depend on VOUT_SCALE_LOOP as follows

TABLE 8

VOUT_SCALE_LOOP	Resolution	Range
E808=1	5 mV	0-2.555V
E804=0.5	10 mV	0-5.11V
E802=0.25	20 mV	0-10.22V
E801=0.125	40 mV	0-20.44V

Command	VOUT_MARGIN_LOW (25h)															
Format	Unsigned Binary															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0

During startup and shutdown the output voltage ramps up and ramps down to and from the voltage defined by VOUT_MARGIN_LOW in a time defined by VOUT_MARGIN_LOW*TON_RISE) /VOUT_COMMAND and (VOUT_MARGIN_LOW*TON_FALL)/VOUT_COMMAND respectively. However, if this command is issued when the output is up and in steady state, the time to transition to the new commanded voltage (provided the OPERATION command is set to Margin Low) is dictated by VOUT_TRANSITION_RATE.

VOUT_TRIM (22H)

The VOUT_TRIM command is used to apply a fixed offset voltage (positive or negative) to the output voltage command value. It is most typically used by the end user to trim the output voltage at the time the device is assembled into the end user's system. This might be done, for example, to adjust the voltage at the pins of a critical IC to optimize its performance.

The exponent is implied to be -8.

The VOUT_TRIM has two data bytes formatted as a two's complement binary integer.

The default value is 0000h. The range is -128 V to 127 V.

Command	**VOUT_TRIM(22h)**															
Format	2's complement Binary															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VOUT_MAX (24H)

This command is used to limit the maximum output voltage, irrespective of whatever voltage is commanded by a combination of VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) and VOUT_TRIM. The intent of this command is to provide a safeguard against a user accidentally setting the output voltage to a possibly destructive level rather than to be the primary output overprotection. It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0600h, which is 6.0V.

Command	**VOUT_MAX (24h)**															
Format	Unsigned Binary															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

If an attempt is made to program the output voltage higher than the limit set by this command, this will flag a WARNING condition, but NOT a fault. The device will respond as follows:

- The commanded output voltage will be set to VOUT_MAX
- The NONE OF THE ABOVE bit will be set in the STATUS_BYTE
- The VOUT bit will be set in the STATUS_WORD
- The VOUT_MAX Warning bit will be set in the STATUS_VOUT register
- The device notifies the host by pulling down SALERT#

The PMBus specification also allows the PMBus device manufacturer to decide the criteria for an out of range data fault. Manhattan establishes 4 criteria for the commanded data to be considered out of range, depending upon the VOUT_SCALE_LOOP setting. It must be noted that in addition to the VOUT_MAX command, there is an internal variable called max_vout, which is 2.56V for VOUT_SCALE_LOOP=1, 5.12 V for VOUT_SCALE_LOOP=0.5, 10.25 V for VOUT_SCALE_LOOP=0.25 and 20.5V for VOUT_SCALE_LOOP=0.125. Attempting to set a VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) higher than max_vout, but lower than VOUT_MAX will not flag a VOUT_MAX warning. Instead, it will be rejected and flagged as a data out of range fault and the following will be done:

- NACK the unsupported data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

It should be noted, however, that if a combination of VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) and VOUT_TRIM is commanded higher than max_vout, but with VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) lower than max_vout, the command is not rejected and no CML fault is flagged. Instead, the output voltage is merely clamped at max_vout.

VOUT_SCALE_LOOP (29H)

This command has two data bytes formatted in the Linear Data format. The value is dimensionless.

The default value is 1. Only an exponent of -3 is supported. The default data, byte, therefore, is E808h.

Command	VOUT_SCALE_LOOP(29h)															
Format	Linear, 2's complement binary															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0

Only 4 values are accepted as valid inputs:

1, ½, ¼ and 1/8.

The devices of the Manhattan platform offer the user the flexibility to set the output voltage with or without a resistive divider at the feedback and sense pins. The resistive divider reduces, or scales, the output voltage so that when the output voltage is correct, the value supplied to the control circuit is equal to the reference voltage. However, commands sent over the PMBus command the output voltage, not the reference voltage. Hence, if a resistive divider is used, one must make sure that the reference voltage VREF to the error amplifier is obtained by scaling down the output voltage command by the same amount.

To allow PMBus devices to map between the commanded voltage (such as 4 V), and the voltage at the feedback pin (e.g. 4 V divided down to match a reference voltage of 0.5 V), the VOUT_SCALE_LOOP command is used.

In general, if the output voltage, VOUT, is processed through a resistive divider which scales it down by a factor K_R, such that $K_R = \frac{V_{ref}}{V_{out}}$ for proper operation at the desired output voltage,

therefore, VOUT_SCALE_LOOP must be set=K_R.

Suppose, for example, the output voltage was 4 V and that the desired voltage at the feedback pin to the PMBus device is 0.5 V. Then K_R is calculated as $K_R = \frac{V_{ref}}{V_{out}}$ =1/8 and VOUT_SCALE_LOOP must be set to E808h=1/8.

Manhattan devices do not provide a separate VOUT_SCALE_MONITOR command for output voltage sensing for telemetry and output UV and OV protection. Therefore, when generating the value reported in response to the READ_VOUT command, the sensed value should be divided by the value of VOUT_SCALE_LOOP. For example, using the same resistor divider ratio as above (0.125, a voltage at the monitoring pin of 0.5 V would result in the value 4 being returned in response to a READ_VOUT command. Similarly in calculating the OV threshold for the output voltage, VOUT_SCALE_LOOP is used as the scale factor.

VOUT_TRANSITION_RATE (27H)

When the device receives either a VOUT_COMMAND or OPERATION (Margin High, Margin Low) that causes the output voltage to change, this command sets the rate in mV/µs at which the output should change voltage. This commanded rate of change does not apply when the unit is commanded to turn on or to turn off.

This command has two data bytes formatted in the Linear Data format. The range of values supported by Manhattan is shown in the table below:

TABLE 8

Exponent	Range	Resolution
0	0-127 mV/us	1 mV/us
-1	0-127.5 mV/us	0.5 mV/us
-2	0-127.75 mV/us	0.25 mV/us
-3	0-127.875 mV/us	0.125 mV/us
-4	0-63.9375 mV/us	0.0625 mV/us

A transition rate of 0 tells the device to complete the transition as fast as possible.

The default value is 0.125 mV/us with 0.125 mV/us resolution. The default data byte, therefore, is E801h, as shown below

Command	VOUT_TRANSITION_RATE (33h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1

Any attempt to set an exponent other than listed above or to set the transition to a negative value will be rejected and result in an invalid data CML fault as follows:

NACK the unsupported/invalid data bytes received before the next STOP condition,

- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

FREQUENCY_SWITCH (33H)

The FREQUENCY_SWITCH command configures the switching frequency (in kHz). This command has two data bytes formatted in the Linear Data format. The default is 600kHz (0258h) and is shown below:

Command	FREQUENCY_SWITCH (33h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0

The supported frequencies are shown in the table below

Manhattan supports frequencies 150 kHz to 2000 kHz given by 48000/N (kHz), where N=24 to 320.

Any intermediate frequency commands are rounded to the closest valid frequency. However, any attempts to set the frequency lower than 150 kHz will result in a switching frequency of 150 kHz. Any attempts to set the frequency higher than 2000 kHz will result in a switching frequency of 2000 kHz.

Note, from the table above, that only exponents of 0 or 1 are allowed. Also, obviously, "negative" frequencies are not allowed. Any attempt to set an exponent that is not 0 or 1, or to set a negative frequency will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

VIN_ON (35H)

The VIN_ON command sets the value of the input voltage PVin, in volts, at which the device is enabled to start power conversion. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.5V for this command, implying that the data word must have an exponent of -1.

The default is F802h (1 V) and is shown below:

Command	**VIN_ON (35h)**															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0

Thresholds from 0 V to 16.5 V are supported with 0.5V resolution. Any values higher than 16.5V will be rejected and result in an invalid data CML fault. Any attempt to set an exponent other than -1 or to set a negative threshold will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition.
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

VIN_OFF (36H)

The VIN_OFF command sets the value of the input voltage PVin, in volts, at which the device is disabled to stop power conversion. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.5V for this command, implying that the data word must have an exponent of -1.

The default is F801h (0.5 V) and is shown below:

Command	VIN_OFF (36h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1

Thresholds from 0V to 16.5 V are supported with 0.5V resolution. Any attempt to set the threshold higher than 16.5V will result in an invalid data CML fault. Any attempt to set an exponent other than -1 or to set a negative threshold will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

IOUT_CAL_OFFSET (39H)

The IOUT_CAL_OFFSET command is used to null out any offsets in the output current measurement. For example, if READ_IOUT returns 10A, where the actual current is known to be 10.5A, this needs an IOUT_CAL_OFFSET of 0.5A to null out the -0.5A offset and make READ_IOUT return 10.5A. This command has two data bytes formatted in the Linear Data Format. Manhattan supports a resolution of 0.5A for this command, implying that the data word must have an exponent of -1. Also, the value set by this command is subtracted from IOUT_OC_FAULT_LIMIT to set the effective overcurrent limit. Manhattan supports values from -128A to 127.5A; however, if IOUT_OC_FAULT_LIMIT-IOUT_CAL_OFFSET< 3, higher values of IOUT_CAL_OFFSET have no impact on the overcurrent threshold. Conversely, if IOUT_OC_FAULT_LIMIT-IOUT_CAL_OFFSET > current_max (37.5A for IR38063), more negative values of IOUT_CAL_OFFSET are rejected and an invalid data CML fault is flagged.

The default is F800 (0A) and is shown below

Command	IOUT_CAL_OFFSET (39h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

VIN_OV_FAULT_LIMIT (55H)

The VIN_OV_FAULT_LIMIT command sets the value of the input voltage PVin, in volts, that causes an input overvoltage fault.. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.25V for this command, implying that the data word must have an exponent of -2.

The default is F060h (24 V) and is shown below:

Command	VIN_OV_FAULT_LIMIT (55h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0

Only 4 distinct values are supported as shown in the table below.

TABLE 9

VIN_OV_FAULT_LIMIT	Threshold (V)
F019	6.25
F03C	15
F050	20
F060	24

Any attempt to set an exponent other than -2 or to set a negative threshold will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

Any intermediate commands (still with 0.25V resolution or -2 exponent) are rounded up to nearest higher supported value. Thresholds higher than 24V are saturated at 24V.

VIN_OV_FAULT_RESPONSE (56H)

The VIN_OV_FAULT_RESPONSE command instructs the device on what action to take in response to an input overvoltage fault. This command has a single data byte. Manhattan supports only 2 types of responses to an overvoltage fault on the input voltage.

TABLE 10

VIN_OV_FAULT_RESPONSE	Response
00	Ignore
80	Shutdown

The default response is to shutdown (80h)

Irrespective of the response type, the device also:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Set the INPUT bit in the upper byte of the STATUS_WORD,
- Sets the VIN_OV_FAULT bit in the STATUS_INPUT register, and
- Notify the host by pulling down SALERT#

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

VIN_UV_WARN_LIMIT (58H)

The VIN_UV_WARN_LIMIT command sets the value of the input voltage that causes aninput voltage low warning.

This alarm is masked until the input exceeds the value set by the VIN_ON command and the unit has been enabled. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.5V for this command, implying that the data word must have an exponent of -1.

The default is F801h (0.5 V) and is shown below:

Command	**VIN_UV_WARN_LIMIT (58h)**															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1

Thresholds from 0 V to 16 V are supported with 0.5V resolution. Any attempt to set an exponent other than -1 or to set a negative threshold will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

In response to the VIN_UV_WARN_LIMIT being exceeded, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Sets the INPUT bit is the upper byte of the STATUS_WORD,
- Sets the VIN_UV_WARNING bit in the STATUS_INPUT register, and
- Notify the host by pulling down SALERT#

VOUT_OV_FAULT_LIMIT (40H)

The VOUT_OV_FAULT_LIMIT command sets the value of the output voltage that causes an output overvoltage fault.

It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 009Bh, which is 0.605V. The range and resolution depend on VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) and VOUT_SCALE_LOOP as follows

TABLE 11

VOUT_SCALE_LOOP	Resolution of VOUT_OV_FAULT_LIMIT-VOUT_COMMAND	Min VOUT_OV_FAULT_LIMIT-VOUT_COMMAND	Max VOUT_OV_FAULT_LIMIT-VOUT_COMMAND
E808=1	10 mV	25mV	655 mV
E804=0.5	20 mV	50mV	1.31V
E802=0.25	40 mV	100mV	2.62V
E801=0.125	80 mV	200mV	5.24V

For a given VOUT_SCALE_LOOP and VOUT_COMMAND, any VOUT_OV_FAULT_LIMIT command that would cause VOUT_OV_FAULT_LIMIT-VOUT_COMMAND < 25 mV/(VOUT_SCALE_LOOP) causes VOUT_OV_FAULT_LIMIT-VOUT_COMMAND to be saturated to 25 mV/(VOUT_SCALE_LOOP). Similarly any VOUT_OV_FAULT_LIMIT command that would cause VOUT_OV_FAULT_LIMIT-VOUT_COMMAND > 655 mV/(VOUT_SCALE_LOOP) causes VOUT_OV_FAULT_LIMIT-VOUT_COMMAND to be saturated to 655 mV/(VOUT_SCALE_LOOP).

VOUT_OV_FAULT_LIMIT-VOUT_COMMAND has a resolution of 10mV

The effective OV threshold is Vout_Command + ROUND((VOUT_OV_FAULT_THRESHOLD-Vout_Command-25mV), resolution) + 25mV/(VOUT_SCALE_LOOP) .

VOUT_OV_FAULT_RESPONSE (41H)

This command controls how the device responds to an overvoltage fault. Manhattan supports only 2 types of responses to an overvoltage fault on the input voltage.

TABLE 12

VOUT_OV_FAULT_RESPONSE	Response
00	Ignore
80	Shutdown

The default response is set to shutdown (80h)

The VOUT_OV_FAULT_RESPONSE command has one data byte.

Irrespective of the response mode, the device:

- Sets the VOUT_OV_FAULT bit (bit 5) in the STATUS_BYTE,
- Sets the VOUT bit (bit 7) in the STATUS_WORD,
- Sets the VOUT_OV_FAULT bit (bit 7) in the STATUS_VOUT register, and
- Notifies the host by issuing SALERT#
- Pulls PGood low

These bits can be cleared only if one or more of the following occur, irrespective of whether the fault condition persists:

- CLEAR_FAULTS command is issued
- Vcc, ENABLE or Operation is cycled

In the Ignore mode, the converter keeps running without interruption. In the shutdown mode, the converter is disabled. If the overvoltage was caused because Vout exceeded VOUT_OV_FAULT_LIMIT, the sync FET is kept on until Vout drops below VOUT_OV_FAULT_LIMIT, following which the sync FET is turned off as well. However, if the overvoltage condition was flagged because VOUT_COMMAND is more than VOUT_OV_FAULT_LIMIT, the converter is disabled without turning on the sync FET.

In the shutdown mode, the converter can be enabled only by cycling Vcc, Enable or the OPERATION command.

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

NACK the unsupported/invalid data bytes received before the next STOP condition,

- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

VOUT_OV_WARN_LIMIT (42H)

The VOUT_OV_WARN_LIMIT command sets the value of the output voltage that causes an output overvoltage warning.

It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0090h, which is 0.56V.

This command has an effective resolution of 1/256 V.

In response to the VOUT_OV_WARN_LIMIT being exceeded, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT_OV_WARNING bit in the STATUS_VOUT register, and
- Notifies the host by issuing SALERT#

VOUT_UV_WARN_LIMIT (43H)

The VOUT_UV_WARN_LIMIT command sets the value of the output voltage that causes an output undervoltage warning. It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0071h, which is 0.44V.

This warning is masked until the device reaches the programmed output voltage and until the Ton_Max_Fault_limit expires. This fault is also masked when the device is disabled. If a Ton_Max_fault occurs, it will not allow a Vout_UV_Warning to be asserted until Vout rises above max(Vout_UV_FAULT_LIMIT, Vout_UV_Warn_limit) at least once before dipping below Vout_UV_WARN_LIMIT again.

This command has an effective resolution of 1/256 V.

In response to the VOUT_UV_WARN_LIMIT being exceeded, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT_UV_WARNING bit in the STATUS_VOUT register, and
- Notifies the host by issuing SALERT#

VOUT_UV_FAULT_LIMIT (44H)

The VOUT_UV_FAULT_LIMIT command sets the value of the output voltage that causes an output undervoltage fault.

It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0065h, which is 0.395V.

This fault is masked until the device reaches the programmed output voltage and until the Ton_Max_Fault_limit expires. This fault is also masked when the device is disabled. If a Ton_Max_fault occurs, it will not allow a Vout_UV_Fault to be asserted until Vout rises above max(Vout_UV_Fault_limit, Vout_UV_WARN_LIMIT) at least once before dipping below VOUT_UV_FAULT_LIMIT again.

This command has an effective resolution of 1/256 V.

In response to the VOUT_UV_FAULT_LIMIT being exceeded, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD,
- Sets the VOUT_UV_FAULT bit in the STATUS_VOUT register, and
- Notifies the host by issuing SALERT#

These bits can be cleared only if one or more of the following occur, irrespective of whether the fault condition persists:

- CLEAR_FAULTS command is issued
- Vcc, ENABLE or Operation is cycled

VOUT_UV_FAULT_RESPONSE (45H)

This command controls how the device responds to an undervoltage fault. Manhattan supports only 2 types of responses to an overvoltage fault on the output voltage.

TABLE 13

VOUT_UV_FAULT_RESPONSE	Response
00	Ignore
80	Shutdown

The default response is set to ignore (00h)

The VOUT_UV_FAULT_RESPONSE command has one data byte.

Irrespective of the response mode, the device:

- Sets the NONE_OF_THE_ABOVE bit (bit 0) in the STATUS_BYTE,
- Sets the VOUT bit (bit 7) in the STATUS_WORD,
- Sets the VOUT_UV_FAULT bit (bit 7) in the STATUS_VOUT register, and
- Notifies the host by issuing SALERT#

In the shutdown mode, the converter can be enabled only by cycling Vcc, Enable or the OPERATION command.

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

IOUT_OC_FAULT_LIMIT (46H)

The IOUT_OC_FAULT_LIMIT command sets the value of the output current, in amperes, that causes the overcurrent detector to indicate an overcurrent fault condition. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.5A for this command, implying that the data word must have an exponent of -1.

The defaults are shown below:

Command	IOUT_OC_FAULT_LIMIT (46h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
IR38064 Default	1	1	1	1	1	0	0	0	0	1	0	1	1	1	0	0

F85Ch (46A)																
IR38063 Default F842h (33A)	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0
IR38062 Default F828h (20A)	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0
IR38060 Default F812h (9A)	1	1	1	1	1	0	0	0	0	0	0	1	0	0	1	0

Any attempt to set an exponent other than -1 or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

For the IR38063, IOUT_OC_FAULT_LIMIT can range from 3A to 37.5A. Any attempts to program this limit lower than 3A is legal, but will result in the limit getting saturated at 3A. However, any attempt to program the limit higher than 37.5A will be rejected and result in an invalid data command fault.

IOUT_OC_FAULT_RESPONSE (47H)

This command controls how the device responds to an overcurrent fault. Manhattan supports the following types of responses:

TABLE 14

IOUT_OC_FAULT_RESPONSE	Response
80	Pulse by pulse for 8 cycles, followed by latched off
B8	Pulse by pulse for 8 cycles, followed by retry after 20 ms blanking

Default response is B8h, i.e, pulse by pulse for 8 cycles, followed by retry after 20 ms blanking. The following actions must be taken for all supported responses, irrespective of the type of response selected:

- Sets the IOUT_OC bit (bit 4) in the STATUS_BYTE,
- Sets the IOUT bit (bit 6) in the STATUS_WORD,
- Sets the IOUT_OC_FAULT bit (bit 7) in the STATUS_IOUT register, and
- Notifies the host by issuing SALERT#
- Pulls PGood low

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

IOUT_OC_WARN_LIMIT (4AH)

The IOUT_OC_FAULT_LIMIT command sets the value of the output current, in amperes, that causes an overcurrent warning. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 0.5A for this command, implying that the data word must have an exponent of -1. The defaults are shown below:

Command	IOUT_OC_WARN_LIMIT (4Ah)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
IR38064 Default F84Eh (39A)	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0
IR38063 Default F838h (28A)	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0
IR38062 Default F823h (17.5A)	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1
IR38060 Default F80Fh (7.5A)	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1

Any attempt to set an exponent other than -1 or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

In response to the IOUT_OC_WARN_LIMIT being exceeded, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE
- Sets the IOUT bit in the STATUS_WORD,
- Sets the IOUT_OC_WARNING bit in the STATUS_IOUT register, and
- Notify the host by pulling down SALERT#

OT_FAULT_LIMIT (4FH)

The OT_FAULT_LIMIT command set the temperature, in degrees Celsius, of the device at which it should indicate an Over-temperature Fault. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 1 deg. C, for this command, implying that the data word must have an exponent of 0.

The default is 0091h (145 C) and is shown below:

Command	OT_FAULT_LIMIT (4Fh)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1

Any attempt to set an exponent other than 0 or to set the threshold to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

OT_FAULT_RESPONSE (50H)

This command controls how the device responds to an over-temperature fault. Manhattan supports 3 types of responses to an over-temperature fault.

TABLE 15

OT_FAULT_RESPONSE	Response
00	Ignore
80	Shutdown
C0	Inhibit

The default response is set to inhibit (C0h)

The OT_FAULT_RESPONSE command has one data byte.

Irrespective of the response mode, the device:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the OT_FAULT bit in the STATUS_TEMPERATURE register
- Notifies the host by issuing SALERT#

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

OT_WARN_LIMIT (51H)

The OT_WARN_LIMIT command set the temperature, in degrees Celsius, of the device at which it should indicate an Over-temperature Warning Alarm. This command has two data bytes formatted in the Linear Data format. Manhattan supports a resolution of 1 deg. C, for this command, implying that the data word must have an exponent of 0.

The default is 007Dh (125 C) and is shown below:

Command	OT_WARN_LIMIT (51h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1

In response to the OT_WARN_LIMIT being exceeded, the device:

- Sets the TEMPERATURE bit in the STATUS_BYTE,
- Sets the OT_WARNING bit in the STATUS_TEMPERATURE register, and
- Notifies the host by pulling down SALERT#

Any attempt to set an exponent other than 0 or to set the threshold to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

POWER_GOOD_ON (5EH)

The POWER_GOOD_ON command sets the value of the output voltage at which the PGood signal is asserted.

It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 0074h, which is 0.45V. The range and resolution depend on VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) and VOUT_SCALE_LOOP as follows

TABLE 16

VOUT_SCALE_LOOP	Resolution of VOUT_COMMAND-POWER_GOOD_ON	Min VOUT_COMMAND-POWER_GOOD_ON	Max VOUT_COMMAND-POWER_GOOD_ON
E808=1	10 mV	0mV	630 mV
E804=0.5	20 mV	0mV	1.26V
E802=0.25	40 mV	0mV	2.52V
E801=0.125	80 mV	0mV	5.04V

For a given VOUT_SCALE_LOOP and VOUT_COMMAND, any POWER_GOOD_ON command that would cause the offset VOUT_COMMAND-POWER_GOOD_ON < 0 mV causes VOUT_COMMAND-POWER_GOOD_ON to be saturated to 0 mV, also causes the PGood signal to be de-asserted. Similarly any POWER_GOOD_ON command that would cause VOUT_COMMAND-POWER_GOOD_ON > 630 mV/(VOUT_SCALE_LOOP) causes VOUT_COMMAND-POWER_GOOD_ON to be saturated to 630 mV/(VOUT_SCALE_LOOP). The offset is truncated to the nearest lower 10 mV/VOUT_SCALE_LOOP

POWER_GOOD_OFF (5FH)

The POWER_GOOD_OFF command sets the value of the output voltage at which the PGood signal is de-asserted.

It has 2 data bytes in unsigned binary format. Manhattan does not support VOUT_MODE, and an exponent of -8 is implied. The default is 006Ch, which is 0.42V. The range and resolution depend on VOUT_COMMAND (or VOUT_MARGIN_HIGH or VOUT_MARGIN_LOW) and VOUT_SCALE_LOOP as follows

TABLE 17

VOUT_SCALE_LOOP	Resolution of VOUT_COMMAND-POWER_GOOD_OFF	Min VOUT_COMMAND-POWER_GOOD_OFF	Max VOUT_COMMAND-POWER_GOOD_OFF
E808=1	10 mV	0mV	630 mV
E804=0.5	20 mV	0mV	1.26V
E802=0.25	40 mV	0mV	2.52V
E801=0.125	80 mV	0mV	5.04V

For a given VOUT_SCALE_LOOP and VOUT_COMMAND, any POWER_GOOD_OFF command that would cause the offset VOUT_COMMAND-POWER_GOOD_OFF < 0 mV causes VOUT_COMMAND-POWER_GOOD_OFF to be saturated to 0 mV. Similarly any POWER_GOOD_OFF command that would cause VOUT_COMMAND-POWER_GOOD_OFF > 630 mV/(VOUT_SCALE_LOOP) causes VOUT_COMMAND-POWER_GOOD_OFF to be saturated to 630 mV/(VOUT_SCALE_LOOP). The offset is truncated to the nearest lower 10 mV/VOUT_SCALE_LOOP.

TON_DELAY (60H)

The TON_DELAY sets the time, in milliseconds, from when a start condition is received (as programmed by the ON_OFF_CONFIG command) until the output voltage starts to rise.

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 127 ms.

The default is 0000h, or 0 ms.

Command	TON_DELAY (60h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Any attempt to set an exponent other than 0 or to set the delay more than 127 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

When TON_DELAY is set to 00h, Manhattan provides the user with a knob zero_delay_start_value[1:0] in the form of two bits of register 6Dh, using the I2C/SMBus interface.

This knob allows the user to achieve a delay as close to 0 as possible by compensating for the ramp offset of the PWM comparator. This is done by making the reference voltage “instantaneously” jump to the value set by zero_delay_start_value[1:0] as shown in the table below:

TABLE 18

zero_delay_start_value[1:0]	Reference voltage start value (mV)
00	0 (Default)
01	50
10	100
11	150

TON_RISE (61H)

The TON_RISE sets the time, in milliseconds, from when the output starts to rise until the voltage has entered the regulation band. Strictly speaking, for Manhattan, it sets the time in which the reference voltage ramps from 0 to the value set by VOUT_COMMAND.

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 127 ms.

The default is 0002h, or 2 ms.

Command	**TON_RISE (61h)**															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Any attempt to set an exponent other than 0 or to set the rise time to more than 127 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

TON_MAX_FAULT_LIMIT (62H)

The TON_MAX_FAULT_LIMIT command sets an upper limit, in milliseconds, on how long the device can attempt to power up the output without reaching the output undervoltage fault limit

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 127 ms.

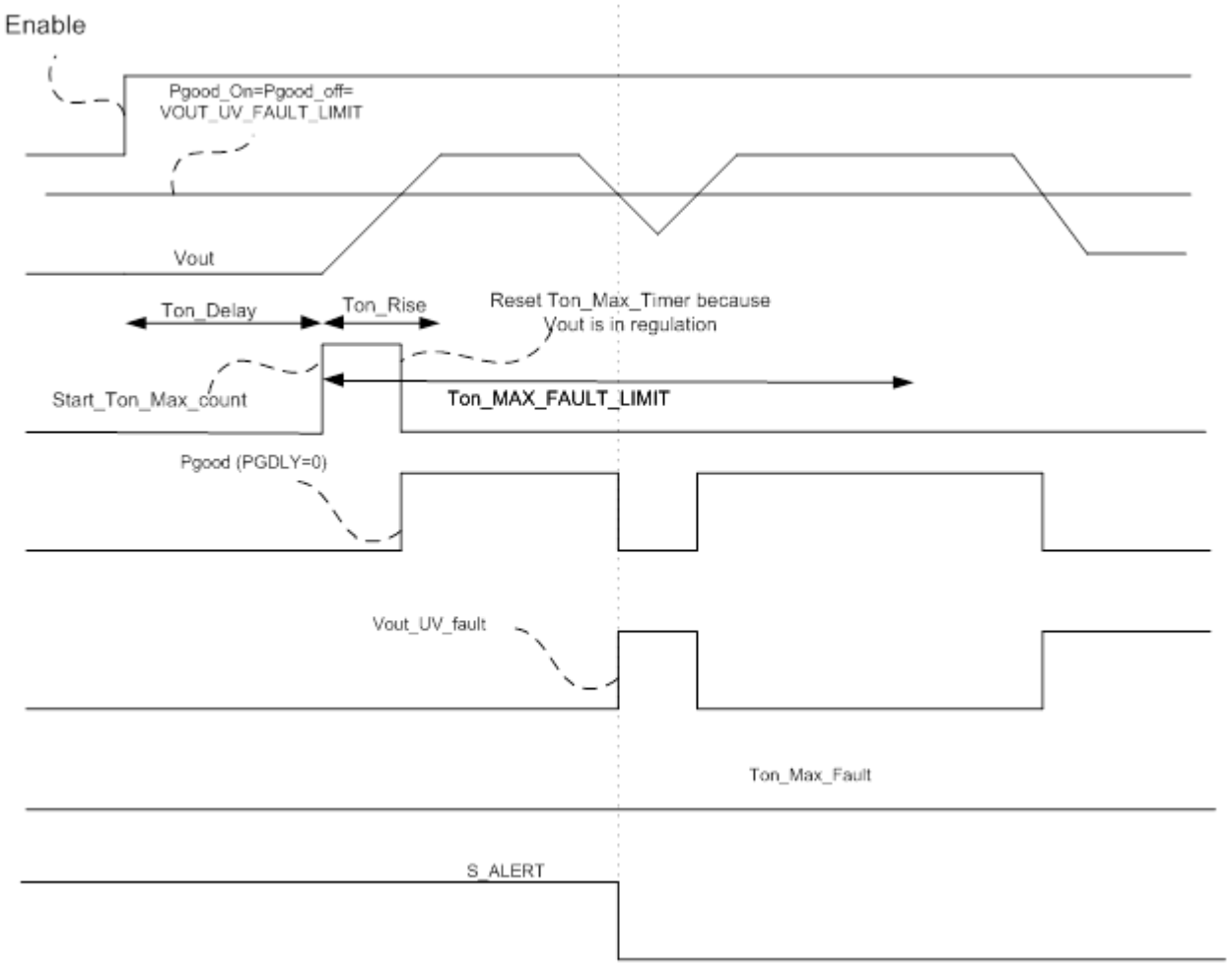
The default is 0000h, which means “no limit” and the device attempts to bring up the voltage indefinitely

Command	**TON_MAX_FAULT_LIMIT (61h)**															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Any attempt to set an exponent other than 0 or to set the rise time to more than 127 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

The timing diagram for Ton_Max_Fault_Limit is shown below.



TON_MAX_FAULT_RESPONSE (63H)

This command controls how the device responds to a Ton_Max fault. Manhattan supports 2 types of responses to a Ton_Max fault.

TABLE 19

TON_MAX_FAULT_RESPONSE	Response
00	Ignore
80	Shutdown

The default response is set to ignore (00h)

The TON_MAX_FAULT_RESPONSE command has one data byte.

Irrespective of the response mode, the device:

- Sets the NONE OF THE ABOVE bit in the STATUS_BYTE,
- Sets the VOUT bit in the STATUS_WORD register
- Sets the Ton Max Fault bit in the STATUS_VOUT register.
- Notifies the host by issuing SALERT#

Any attempt to command a response type other than those defined in the table above will be rejected and will result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

TOFF_DELAY (64H)

The TOFF_DELAY sets the time, in milliseconds, from when a stop condition is received (if a soft off is allowed by the configuration of the ON_OFF_CONFIG command) until the unit stops transferring energy to the output.

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 127 ms.

The default is 0000h, or 0 ms.

Command	**TOFF_DELAY (64h)**															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Any attempt to set an exponent other than 0 or to set the delay more than 127 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

TOFF_FALL (65H) – NOT SUPPORTED

Note, the device will acknowledge this command, but ignore it.

The TOFF_FALL sets the time, in milliseconds, in which the reference voltage ramps down to zero (If a soft off is allowed by the configuration of the ON_OFF_CONFIG command). This time starts from the end of the turn-off delay time

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 127 ms.

The default is 0001h, or 1 ms.

Command	TOFF_FALL (65h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Any attempt to set an exponent other than 0 or to set the delay more than 127 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

STATUS_BYTE (78H)

The STATUS_BYTE command returns one byte of information with a summary of the most critical faults. The format is indicated in the table below:

TABLE 20

Bit	Status Bit Name	Description
7	BUSY	A fault was declared because the device was busy and unable to respond. This bit is hardcoded to 0
6	OFF	This bit is asserted if the device is OFF because power conversion is disabled
5	VOUT_OV_FAULT	An output overvoltage fault has occurred
4	IOUT_OC_FAULT	An output overcurrent fault has occurred
3	VIN_UV_FAULT	An input undervoltage fault has occurred; this is hardcoded to 0.
2	TEMPERATURE	A temperature fault or warning has occurred
1	CML	A communications, memory or logic fault has Occurred
0	NONE OF THE ABOVE	A fault or warning not listed in bits [7:1] has Occurred

STATUS_WORD (79H)

The STATUS_WORD command returns two bytes that summarize the device fault information. Based on the information in these bytes, the host can get more information by reading the appropriate lower level status registers.

The low byte of the STATUS_WORD is the same register as the STATUS_BYTE command. The format for the high byte is indicated in the table below

TABLE 21

Bit	Status Bit Name	Description
7	VOUT	An output voltage fault or warning has occurred
6	IOUT/POUT	An output current or output power fault or warning has occurred.
5	INPUT	An input voltage, input current, or input power fault or warning has occurred
4	MFR_SPECIFIC	A manufacturer specific fault or warning has Occurred
3	POWER_ GOOD#	When this is 1, it indicates that the output power is not good.
2	FANS	A fan or airflow fault or warning has occurred. In Manhattan, this is hardcoded to 0.
1	OTHER	A bit in STATUS_OTHER is set. In Manhattan, this is hardcoded to 0.
0	UNKNOWN	A fault type not given in bits [15:1] of the STATUS_WORD has been detected. In Manhattan, this is hardcoded to 0.

Since bits [2:0] are hardcoded in Manhattan, STATUS_WORD is effectively read only, and any other bit in STATUS_WORD can be cleared only by clearing the bit or all of the bits from the corresponding lower level register that cause it to be set.

STATUS_VOUT (7AH)

The STATUS_VOUT command returns one data byte with contents as follows:

TABLE 22

Bit	Description
7	VOUT_OV_FAULT (Output Overvoltage Fault)
6	VOUT_OV_WARNING (Output Overvoltage Warning)
5	VOUT_UV_WARNING (Output Undervoltage Warning)
4	VOUT_UV_FAULT (Output Undervoltage Fault)
3	VOUT_MAX Warning (An attempt has been made to set the output voltage to value higher than allowed by the VOUT_MAX command)
2	TON_MAX_FAULT
1	TOFF_MAX_WARNING; In Manhattan, this is hardcoded to 0
0	VOUT Tracking Error [1]; In Manhattan, this is hardcoded to 0

Assuming that the fault or warning condition that causes a STATUS_VOUT bit to be set disappears, the individual bit clearing mechanism can be used to clear that bit. For instance, if STATUS_VOUT is 30h,

indicating that bits 5 and 4 (VOUT_UV_WARNING as well as VOUT_UV_FAULT) are set, writing 30h to STATUS_VOUT will clear bits [5:4] if the fault and warning conditions no longer exist.

STATUS_IOUT (7BH)

The STATUS_IOUT command returns one data byte with contents as follows:

TABLE 23

Bit	Description
7	IOUT_OC_FAULT (Output Overcurrent Fault)
6	IOUT_OC_LV_FAULT (Output Overcurrent And Low Voltage Fault); In Manhattan this is hardcoded to 0.
5	IOUT_OC_WARNING (Output Overcurrent Warning)
4	IOUT_UC_FAULT (Output Undercurrent Fault); In Manhattan this is hardcoded to 0.
3	Current Share Fault [1]; In Manhattan this is hardcoded to 0.
2	In Power Limiting Mode [2] ; In Manhattan this is hardcoded to 0.
1	POUT_OP_FAULT (Output Overpower Fault) ; In Manhattan this is hardcoded to 0.
0	POUT_OP_WARNING (Output Overpower Warning) ; In Manhattan this is hardcoded to 0.

Assuming that the fault or warning condition that causes a STATUS_IOUT bit to be set disappears, the individual bit clearing mechanism can be used to clear that bit. For instance, if STATUS_IOUT is A0h, indicating that bits 7 and 5 (IOUT_OC_FAULT as well as IOUT_OC_WARNING) are set, writing A0h to STATUS_IOUT will clear bits 7 and 5 if the fault and warning conditions no longer exist.

STATUS_INPUT (7CH)

The STATUS_INPUT command returns one data byte with contents as follows:

TABLE 24

Bit	Description
7	VIN_OV_FAULT (Input Overvoltage Fault)
6	VIN_OV_WARNING (Input Overvoltage Warning); In Manhattan this is hardcoded to 0.
5	VIN_UV_WARNING (Input Undervoltage Warning)
4	VIN_UV_FAULT (Input Undervoltage Fault); In Manhattan this is hardcoded to 0.
3	Unit Off For Insufficient Input Voltage
2	IIN_OC_FAULT (Input Overcurrent Fault); In Manhattan this is hardcoded to 0.
1	IIN_OC_WARNING (Input Overcurrent Warning); In Manhattan this is hardcoded to 0.
0	PIN_OP_WARNING (Input Overpower Warning); In Manhattan this is hardcoded to 0.

Assuming that the fault or warning condition that causes a STATUS_INPUT bit to be set disappears, the individual bit clearing mechanism can be used to clear that bit. For instance, if STATUS_INPUT is 28h, indicating that bits 5 and 3 (VIN_UV_WARNING as well as Unit off for Insufficient Input Voltage) are set, writing 28h to STATUS_INPUT will clear bits 7 and 5 if the fault and warning conditions no longer exist.

STATUS_TEMPERATURE (7DH)

The STATUS_TEMPERATURE command returns one data byte with contents as follows:

TABLE 25

Bit	Description
7	OT_FAULT (Overtemperature Fault)
6	OT_WARNING (Overtemperature Warning)
5	UT_WARNING (Undertemperature Warning); In Manhattan this is hardcoded to 0.
4	UT_FAULT (Undertemperature Fault); In Manhattan this is hardcoded to 0.
3	Reserved ; In Manhattan this is hardcoded to 0.
2	Reserved ; In Manhattan this is hardcoded to 0.
1	Reserved ; In Manhattan this is hardcoded to 0.
0	Reserved ; In Manhattan this is hardcoded to 0.

Assuming that the fault or warning condition that causes a STATUS_TEMPERATURE bit to be set disappears, the individual bit clearing mechanism can be used to clear that bit. For instance, if STATUS_TEMPERATURE is C0h, indicating that bits 7 and 6 (OT_FAULT as well as OT_WARNING) are set, writing C0h to STATUS_TEMPERATURE will clear bits 7 and 6 if the fault and warning conditions no longer exist.

STATUS_CML (7EH)

The STATUS_CML command returns one data byte with contents as follows:

TABLE 26

Bit	Description
7	Invalid Or Unsupported Command Received; set to 1 if PMBus command is invalid or unsupported
6	Invalid Or Unsupported Data Received; set to 1 if invalid data is received or if any attempt is made to send too many bytes to the Manhattan device. An example of the latter is when the host tries to write a PEC byte to the Manhattan device when it is operating in PEC unsupported mode.
5	Packet Error Check Failed; set to 1 by a PEC error
4	Memory Fault Detected
3	Processor Fault Detected ; hard coded to 0
2	Reserved ; hard coded to 0
1	A communication fault other than the ones listed in this table has occurred; set to 1 if any of the conditions listed are true: a) Improperly set read bit in the address. b) Host sends too many bits c) Host reads too few bits d) Host reads too many bytes e) Command too short
0	Other Memory Or Logic Fault has occurred. ; hard coded to 0.

Assuming that the fault or warning condition that causes a STATUS_CML bit to be set disappears, the individual bit clearing mechanism can be used to clear that bit. For instance, if STATUS_CML is 40h, indicating that bit 6 (Invalid or Unsupported Data received) are set, writing 40h to STATUS_CML will clear bit 6.

READ_VIN (88H)

The READ_VIN command returns the input voltage in volts, with an effective resolution of 1/32V.

The two data bytes are formatted in the Linear Data format.

READ_VOUT (8BH)

The READ_VOUT command returns the output voltage in volts, with an effective resolution of 1/256 V.

The two data bytes are formatted in the Linear Data format, 16 bit unsigned binary with the exponent fixed at -8.

This voltage is calculated in Manhattan by multiplying the voltage at the sense pin by 1/Vout_Scale_Loop.

READ_IOUT (8CH)

The READ_IOUT command returns the output current in amps, with an effective resolution of 1/16A.

The two data bytes are formatted in the Linear Data format.

This reported current is calculated in Manhattan by adding IOUT_CAL_OFFSET to the value of the sensed current.

This is also the value that is compared with IOUT_OC_WARN_LIMIT to generate Iout_OC_Warning alarm.

READ_TEMPERATURE_1 (8DH)

The READ_TEMPERATURE_1 command returns the die temperature, in degrees C, with an effective resolution of 1 C. The two data bytes are formatted in the Linear Data format.

READ_POUT (96H)

The READ_POUT command returns the output power, in watts, of the device. The two data bytes are formatted in the Linear Data format.

PMBUS_REVISION (98H)

PMBUS_REVISION command stores or reads the revision of the PMBus to which the

device is compliant. The command has one data byte. Bits [7:5] indicate the revision of PMBus specification Part I to which the device is compliant. Bits [3:0] indicate the revision of PMBus specification Part II to which the device is compliant.

Manhattan implements this command as a read only command and will return 22h, indicating that is compliant to Rev 1.2 of PMBus specification Part I and compliant to Rev 1.2 of PMBus specification Part 2.

MFR_ID (99H)

The MFR_ID command is used to either set or read the manufacturer's ID (name, abbreviation or symbol that identifies the unit's manufacturer). MFR_ID is typically only set once, at the time of manufacture.

Manhattan will return 3 bytes in response to this command as 02 52 49. The 1st byte (read only) indicates that 2 further bytes will follow. Remaining 2 bytes are ASCII equivalent of "IR". However the user may over-write with any 2 bytes of data.

MFR_MODEL (9AH)

The MFR_MODEL command is used to either set or read the manufacturer's model number. MFR_MODEL is typically set once, at the time of manufacture.

Manhattan implements this command with 2 bytes. The 1st byte (read only) is 01h indicating that one more byte will follow. The 2nd byte can be written to with values not equal to 00h. Executing a read on the command will then return 01h followed by the value written to the 2nd byte. If an attempt is made to write 00h to the 2nd byte, and then a read is executed on this command, it will return the default values (stored in product_id_otp[7:0]) in the 2nd byte:

Device	Read back Value (if write_value=00h)	Read back Value (if write_value=XXh)
IR38064	34h	XX (defined by user)
IR38063	33h	XX (defined by user)
IR38062	32h	XX (defined by user)
IR38060	30h	XX (defined by user)

MFR_REVISION (9BH)

The MFR_REVISION command is used to either set or read the manufacturer's revision number. Each manufacturer uses the format of their choice for the revision number. MFR_REVISION is typically set at the time of manufacture or if the device is updated to a later revision.

Manhattan implements this command with 2 bytes. The 1st byte (read only) is 01h indicating that one more byte will follow. The 2nd byte can be written to with values not equal to 00h. Executing a read on the command will then return 01h followed by the value written to the 2nd byte. If an attempt is made to write 00h to the 2nd byte, and a read is executed on this command, it will return the value of IC_DEVICE_REV in the 2nd byte.

IC_DEVICE_ID (ADH)

The IC_DEVICE_ID command is used to either set or read the type or part number of an IC embedded within a PMBus that is used for the PMBus interface. Each manufacturer uses the format of their choice

for the IC device identification. IC_DEVICE_ID is typically only set once, at the time of manufacture, and is read only.

In Manhattan, this command will return two bytes. The 1st byte is 01h indicating that one more byte will follow. The 2nd byte is the value in product_id_otp [7:0].

IC_DEVICE_REV (AEH)

The IC_DEVICE_REV command is used to either set or read the revision of the IC whose type or part number is set or read with the IC_DEVICE_ID command. Each manufacturer uses the format of their choice for the IC device revision. IC_DEVICE_REV is typically only set once, at the time of manufacture. Any new revision in the IC will be accompanied by rewiring the metal to indicate the new revision number and this number gets passed to IC_DEVICE_REV.

MFR_READ_REG (D0H)

The MFR_READ_REG command is a manufacturer specific command (xD0) that utilizes a Custom Protocol to read from the register map.

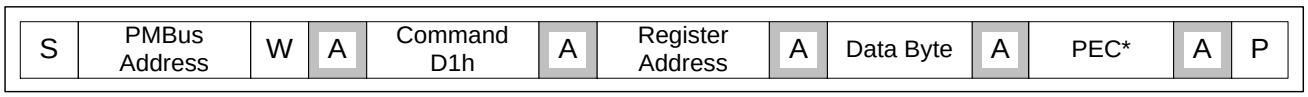
It uses a custom transaction type as shown below:

Thus, this command will allow atomic reads of 16 bit parameters from the register map instead of having to read them in two separate read byte transactions. For instance, with a register address=0x24, this command will allow us to read both the vout_high and vout_low registers, the contents of which together constitute the Vout_Command. The PEC is optional.

MFR_WRITE_REG (D1H)

The MFR_WRITE_REG command is a manufacturer specific command (xD1) that utilizes the Write Word Protocol to write to the register map.

It uses the Write Word transaction type as shown below:



Thus, the register to be written to forms the low byte and the data to be written forms the high byte in this transaction. The PEC is optional.

MFR_I2C_ADDRESS (D6H)

This is a manufacturer specific command, the 7 LSBs of which allow the user to set the 7-bit I2C base address for the device. The device will then respond to I2C commands sent to this address, offset by the offset setting resistor on the ADDR pin. The default value (base address) is 10h.

For example, if MFR_I2C_ADDRESS=10h, and the resistor on the address pin sets an offset of +05h, the device will respond to commands sent to address 15h.

MFR_TPGDLY (D8H)

This is a manufacturer specific command that allows a programmable delay from Vout exceeding the power good threshold to the assertion of the power good signal.

It has 2 data bytes in Linear Format as shown below. Manhattan supports a resolution of 1 ms, implying an exponent of 0. The maximum supported value is 10 ms.

The default is 0000h, or 0 ms.

Command	MFR_TPGDLY (D8h)															
Format	Linear															
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	High Byte								Low Byte							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Any attempt to set an exponent other than 0 or to set the power good assertion delay to more than 10 ms or to a negative value will be rejected and result in an invalid data CML fault as follows:

- NACK the unsupported/invalid data bytes received before the next STOP condition,
- Flush or ignore the received command code and any received data,
- Set the CML bit in the STATUS_BYTE,
- Set the Invalid Or Unsupported Data Received bit in the STATUS_CML register
- Notify the host by pulling down SALERT#

MFR_FCCM (D9H)

This is a manufacturer specific command that allows the user to choose between forced continuous conduction mode and adaptive on-time operation at light load. This command has a single byte in linear format. There are only two values possible

TABLE 27

MFR_FCCM	Operation mode at light load
00h	AOT
01h	CCM

The default is 01h, or CCM.

In the digital mode of operation, this command always overrides the En/FCCM pin settings.

Any attempt to set a mode other than the two given in the table above will be rounded to 00 or 01h, depending upon the LSB.

MFR_VOUT_PEAK (DBH)

This is a manufacturer specific command that continuously records and reports the highest value of Read Vout. It has the same resolution as Read_Vout, i.e., 1/256 V.

It has two data bytes are formatted in the Linear Data format, 16 bit unsigned binary with the exponent fixed at -8.

This value is cleared every time it is read.

The implication of that is that if a new Vout peak is reached, a single read of MFR_VOUT_PEAK will return the correct value.

MFR_IOUT_PEAK (DCH)

This is a manufacturer specific command, which continuously records and reports the highest value of Read Iout. It has the same resolution as Read_Iout, i.e., 1/16A. It has two data bytes are formatted in the Linear Data format.

This value is cleared every time it is read.

The implication of that is that if a new Iout peak is reached, a single read of MFR_IOUT_PEAK will return the correct value.

MFR_TEMPERATURE_PEAK (DDH)

This is a manufacturer specific command, which continuously records and reports the highest value of Read_Temperature_1. It has the same resolution as Read_Temperature_1, i.e., 1C.

It has 2 data bytes are formatted in the Linear Data format.

This value is cleared every time it is read.

The implication of that is that if a new temperature peak is reached, a single read of MFR_TEMPERATURE_PEAK will return the correct value.

NON-STANDARD COMMANDS/REGISTERS

For greater flexibility, Manhattan includes a few user programmable registers. These can be accessed via the PMBus MFR_READ_REG and MFR_WRITE_REG commands. Additionally, they may be accessed via I2C.

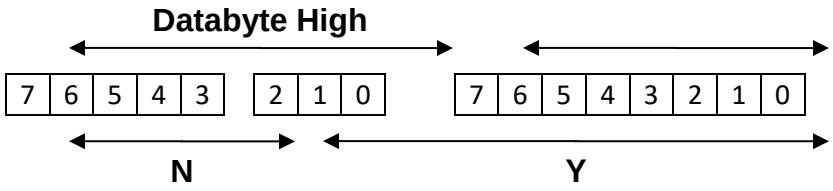
Register Address	Register Bit Meaning
20h	Bit [7]: do not change Bit [6:0]: PMBus address
71h	Bit [7:0] Offset for voltage reporting. 2-s complement format where the LSB represents 4mV

11-BIT LINEAR DATA FORMAT

Monitored parameters use the Linear Data Format encoding into 1 Word (2 bytes), where:

$$Value = Y \times 2^N$$

Note N and Y are “signed” values. The 5 MSBs provide the exponent in 2’s complement form and the 11 LSBs provide the mantissa, also in 2s complement.



16-BIT LINEAR DATA FORMAT

This format is only used for VOUT related commands. All 16 bits provide the mantissa in unsigned binary format, except for Vout_Trim which uses 2s complement form. For Manhattan, the exponent is -8.

$$Value = Y \times 2^N$$