19-5703; Rev 3; 8/12

6-Channel Intelligent Fan Controller

MAX31785

General Description

The MAX31785 is a closed-loop multichannel fan controller. Automatic closed-loop fan control saves system power by operating the fans at the lowest possible speeds. Added benefits of slower fan speeds include lower audible noise, longer fan life, and reduced system maintenance. Based on a user-programmable lookup table (LUT), the device automatically adjusts the speeds of the six independent fans based on one or more of the 11 available temperature sensors. Alternately, an external host can manually command the fan speeds and the device automatically adjusts the fan speeds. The device contains a fan-health-diagnostic function to help users predict impending fan failures. The device can also monitor up to six remote voltages.

Applications

Network Switches/Routers

Base Stations

Servers

Smart Grid Network Systems

Industrial Controls

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX31785ETL+	-40°C to +85°C	40 TQFN-EP*
MAX31785ETL+T	-40°C to +85°C	40 TQFN-EP*

+Denotes a lead(Pb)-free/RoHS-compliant package.
T = Tape and reel.
**EP = Exposed pad.*

Features

- **Six Independent Channels of Fan Control**
  - **Supports 3-Wire and 4-Wire Fans**
  - **Automatic Closed-Loop Fan Speed Control**
  - **RPM- or PWM-Based Control**
  - **Optional Manual Mode Control**
  - **Fast and Slow PWM Frequency Options**
  - **Staggered Fan Spin-Up Eases Power-Supply Stress**
  - **Dual Tachometer (12-Fan Support)**
  - **Fan-Fault Detection**
  - **Fan-Health Status Meter**
  - **Nonvolatile Fan Operating Run Time Meter**
- **Supports Up to 11 Temperature Sensors**
  - **Up to Six External Thermal Diodes with Automatic Series Resistance Cancellation**
  - **One Internal Temperature Sensor**
  - **Up to Four I^2C Digital Temperature Sensors**
  - **Fault Detection on All Temperature Sensors**
- **Available 6-Channel ADC to Measure Remote Voltages**
- **PMBus™-Compliant Command Interface**
- **I^2C/SMBus-Compatible Serial Bus with Bus Timeout Function**
- **On-Board Nonvolatile Fault Logging and Default Configuration Setting**
- **No External Clocking Required**
- **+3.3V Supply Voltage**

PMBus is a trademark of SMIF, Inc.

***Note:** Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, go to: **www.maxim-ic.com/errata**.*

6-Channel Intelligent Fan Controller

TABLE OF CONTENTS

Absolute Maximum Ratings . . . 6
Recommended Operating Conditions . . . 6
DC Electrical Characteristics . . . 6
I2C/SMBus Interface Electrical Characteristics . . . 8
I2C/SMBus Timing . . . 8
Typical Operating Characteristics . . . 9
Pin Configuration . . . 10
Pin Description . . . 10
Block Diagram . . . 12
Multiple Device Connection Diagram . . . 13
Detailed Description . . . 14
- Address Select . . . 16
- SMBus/PMBus Operation . . . 16
  - SMBus/PMBus Communication Examples . . . 16
- Group Command . . . 17
  - Group Command Write Format . . . 17
- Addressing . . . 17
- ALERT and Alert Response Address (ARA) . . . 17
  - Alert Response Address (ARA) Byte Format . . . 18
- Host Sends or Reads Too Few Bits . . . 18
- Host Sends or Reads Too Few Bytes . . . 18
- Host Sends Too Many Bytes or Bits . . . 18
- Host Reads Too Many Bytes or Bits . . . 18
- Host Sends Improperly Set Read Bit in the Slave Address Byte . . . 18
- Unsupported Command Code Received . . . 18
- Invalid Data Received . . . 18
- Host Reads from a Write-Only Command . . . 19
- Host Writes to a Read-Only Command . . . 19
- SMBus Timeout . . . 19
- PMBus Operation . . . 19
- PMBus Protocol Support . . . 19
- Data Format . . . 19
- Interpreting Received DIRECT Format Values . . . 20
- Sending a DIRECT Format Value . . . 20
- Fault Management and Reporting . . . 21
- System Watchdog Timer . . . 21
- Temperature Sensor Operation . . . 21

TABLE OF CONTENTS (continued)

Fan Control Operation 22
Dual Fan Applications 22
Automatic Fan Control Operation 22
Pulse Stretching 24
Fan Spin-Up 24
Fan Power-On Sequencing 25
Fan Health-Meter Function 25
PAGE (00h) 27
CLEAR_FAULTS (03h) 28
WRITE_PROTECT (10h) 28
STORE_DEFAULT_ALL (11h) 28
RESTORE_DEFAULT_ALL (12h) 28
CAPABILITY (19h) 28
VOUT_MODE (20h) 29
VOUT_SCALE_MONITOR (2Ah) 29
FAN_CONFIG_1_2 (3Ah) 29
FAN_COMMAND_1 (3Bh) 30
VOUT_OV_FAULT_LIMIT (40h) 30
VOUT_OV_WARN_LIMIT (42h) 30
VOUT_UV_WARN_LIMIT (43h) 31
VOUT_UV_FAULT_LIMIT (44h) 31
OT_FAULT_LIMIT (4Fh) 31
OT_WARN_LIMIT (51h) 31
STATUS_BYTE (78h) 32
STATUS_WORD (79h) 32
STATUS_VOUT (7Ah) 32
STATUS_CML (7Eh) 33
STATUS_MFR_SPECIFIC (80h) 33
STATUS_FANS_1_2 (81h) 33
READ_VOUT (8Bh) 34
READ_TEMPERATURE_1 (8Dh) 34
READ_FAN_SPEED_1 (90h) 34
PMBUS_REVISION (98h) 34
MFR_ID (99h) 34
MFR_MODEL (9Ah) 34
MFR_REVISION (9Bh) 34
MFR_LOCATION (9Ch) 34

TABLE OF CONTENTS (continued)

MFR_DATE (9Dh) . . . 34
MFR_SERIAL (9Eh) . . . 34
MFR_MODE (D1h) . . . 35
MFR_VOUT_PEAK (D4h) . . . 36
MFR_TEMPERATURE_PEAK (D6h) . . . 36
MFR_VOUT_MIN (D7h) . . . 36
MFR_FAULT_RESPONSE (D9h) . . . 36
MFR_NV_FAULT_LOG (DCh) . . . 37
MFR_TIME_COUNT (DDh) . . . 40
MFR_TEMP_SENSOR_CONFIG (F0h) . . . 40
MFR_FAN_CONFIG (F1h) . . . 41
MFR_FAN_LUT (F2h) . . . 43
TEMPERATURE STEP: Temperature Level Setting . . . 43
FAN SPEED STEP: Fan PWM Duty Cycle or Fan Speed Setting . . . 44
MFR_READ_FAN_PWM (F3h) . . . 45
MFR_FAN_FAULT_LIMIT (F5h) . . . 45
MFR_FAN_WARN_LIMIT (F6h) . . . 45
MFR_FAN_RUN_TIME (F7h) . . . 46
MFR_FAN_PWM_AVG (F8h) . . . 46
MFR_FAN_PWM2RPM (F9h) . . . 46
Applications Information . . . 46
Power-Supply Decoupling . . . 46
Open-Drain Pins . . . 46
Typical Operating Circuit . . . 47
Package Information . . . 47
Revision History . . . 48

LIST OF FIGURES

Figure 1. Automatic Fan Control . . . 23
Figure 2. Fan Speed Example . . . 23
Figure 3. Fan Spin-Up . . . 24
Figure 4. Fan PWM to RPM Example . . . 25
Figure 5. MFR_NV_FAULT_LOG . . . 37
Figure 6. Fan Lookup Table (LUT) Format . . . 44

LIST OF TABLES

Table 1. PMBus Command Codes 14
Table 2. PMBus/SMBus Serial-Port Address 16
Table 3. PMBus Command Code Coefficients 20
Table 4. Coefficients for DIRECT Format Value 20
Table 5. Device Parametric Monitoring States 21
Table 6. DS75LV Address Pin Configurations 21
Table 7. Fan Control Operation Modes 22
Table 8. Fan Health Conditions 26
Table 9. Page Commands 27
Table 10. WRITE_PROTECT Command Byte 28
Table 11. CAPABILITY Command Byte 28
Table 12. VOUT_SCALE_MONITOR 29
Table 13. FAN_CONFIG_1_2 Command Byte 29
Table 14. PWM Fan Mode (FAN_CONFIG_1_2 Bit 6 = 0) 30
Table 15. RPM Fan Mode (FAN_CONFIG_1_2 Bit 6 = 1) 30
Table 16. STATUS_BYTE 32
Table 17. STATUS_WORD 32
Table 18. STATUS_VOUT 32
Table 19. STATUS_CML 33
Table 20. STATUS_MFR_SPECIFIC 33
Table 21. STATUS_FANS_1_2 33
Table 22. MFR_MODE 35
Table 23. MFR_FAULT_RESPONSE 36
Table 24. MFR_NV_FAULT_LOG 38
Table 25. MFR_TEMP_SENSOR_CONFIG 40
Table 26. MFR_FAN_CONFIG 41
Table 27. MFR_FAN_LUT 43
Table 28. Valid Temperature Range 43
Table 29. Monitored Fan Fault and Warning Parameters 45
Table 30. MFR_FAN_PWM2RPM 46

6-Channel Intelligent Fan Controller

ABSOLUTE MAXIMUM RATINGS

V_{DD} to V_{SS} .....-0.3V to +5.5V
RS- to V_{SS} .....-0.3V to +0.3V
All Other Pins Except REG18 and
REG25 Relative to V_{SS} .....-0.3V to (V_{DD} + 0.3V)*
Continuous Power Dissipation (T_A = +70°C)
TQFN (derate 35.7mW/°C above +70°C).....2857.1mW

Operating Temperature Range .....-40°C to +85°C
Storage Temperature Range.....-55°C to +125°C
Lead Temperature (soldering, 10s) .....+300°C
Soldering Temperature (reflow) .....+260°C

**Subject to not exceeding +5.5V.*

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS

(T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{DD} Operating Voltage Range	V_{DD}	(Note 1)	2.7		5.5	V
Input Logic 1	V_{IH}		0.7 x V_{DD}		V_{DD} + 0.3	V
Input Logic 0	V_{IL}		0		0.3 x V_{DD}	V
Input Logic-High: SCL, SDA, MSCL, MSDA	V_{I2C_IH}	2.7V ≤ V_{DD} ≤ 3.6V (Note 1)	2.1		V_{DD} + 0.3	V
Input Logic-Low: SCL, SDA, MSCL, MSDA	V_{I2C_IL}	2.7V ≤ V_{DD} ≤ 3.6V (Note 1)	0		+0.8	V

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current	I_{CPU}	(Note 2)		3		mA
	$I_{PROGRAM}$			8		
Brownout Voltage	V_{BO}	Monitors V_{DD} (Note 1)	2.40	2.46	2.55	V
Brownout Hysteresis	V_{BOH}	Monitors V_{DD} (Note 1)		30		mV
Internal System Clock	f_{MOSC}			4.0		MHz
System Clock Error (Note 3)	$f_{ERR:MOSC}$	+25°C ≤ T_A ≤ +85°C	-3		+2	%
		-40°C ≤ T_A ≤ +25°C	-6.5		+1.6	
Output Logic-Low	V_{OL1}	I_{OL} = 4mA (Note 1)			0.4	V
Output Logic-High	V_{OH1}	I_{OH} = -2mA (Note 1)	V_{DD} - 0.5			V
PWM Pullup Current	I_{PU}	V_{PIN} = V_{SS}, V_{DD} = 3.3V	38	55	107	µA
ADC Internal Reference				1.225		V
ADC Voltage Measurement Error	V_{ERR}		-1		+1	%

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
ADC Internal Reference Temperature Drift			-0.5		+0.5	%
ADC Internal Reference Initial Accuracy (+25°C)			-1		+1	mV
ADC Full-Scale Input Voltage	V_{FS}		1.213	1.225	1.237	V
ADC Measurement Resolution	V_{LSB}			300		µV
ADC Bit Resolution			12			Bits
RS+ Input Resistance	R_{IN}		15			MΩ
ADC Integral Nonlinearity	INL	(Note 4)			±8	LSB
ADC Offset	V_{OFFSET}			±2		LSB
Internal Temperature Measurement Error		T_A = -40°C to +85°C	-3		+3	°C
Remote Temperature Measurement Error (MAX31785 Error Only)		T_A = 0°C to +60°C, T_{DIODE} = +60°C to +120°C	-1.5		+1.5	°C
		T_A = 0°C to +60°C, T_{DIODE} = -45°C to +120°C	-1.75		+1.75	
		T_A = -40°C to +85°C, T_{DIODE} = +60°C to +120°C	-2.75		+2.75	
		T_A = -40°C to +85°C, T_{DIODE} = -45°C to +120°C	-3.0		+3.0	
Store Default All Time				45		ms
Nonvolatile Log Write Time				12		ms
Nonvolatile Log Delete Time				200		ms
Flash Endurance	N_{FLASH}	T_A = +50°C	20,000			Write Cycles
Data Retention		T_A = +50°C	100			Years
Voltage Sample Rate				10		ms
RPM Sample Rate				1000		ms
Temperature Sample Rate				1000		ms
Device Startup Time		Measured from POR until monitoring begins		12		ms
Fan PWM Frequency			30		25,000	Hz
Fan PWM Resolution				7		Bits

I²C/SMBUS INTERFACE ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}		10		100	kHz
Bus Free Time Between STOP and START Conditions	t_{BUF}	(Note 5)	1.4			ms
Hold Time (Repeated) START Condition	$t_{HD:STA}$		4.0			µs
Low Period of SCL	t_{LOW}		4.7			µs
High Period of SCL	t_{HIGH}		4.0			µs
Data Hold Time	$t_{HD:DAT}$	Receive	0			ns
		Transmit	300			
Data Setup Time	$t_{SU:DAT}$		100			ns
START Setup Time	$t_{SU:STA}$		4.7			µs
SDA and SCL Rise Time	t_R				300	ns
SDA and SCL Fall Time	t_F				300	ns
STOP Setup Time	$t_{SU:STO}$		4.0			µs
Clock Low Timeout	t_{TO}		25		35	ms

Note 1: All voltages are referenced to ground (V_{SS}). Currents entering the IC are specified as positive, and currents exiting the IC are negative.

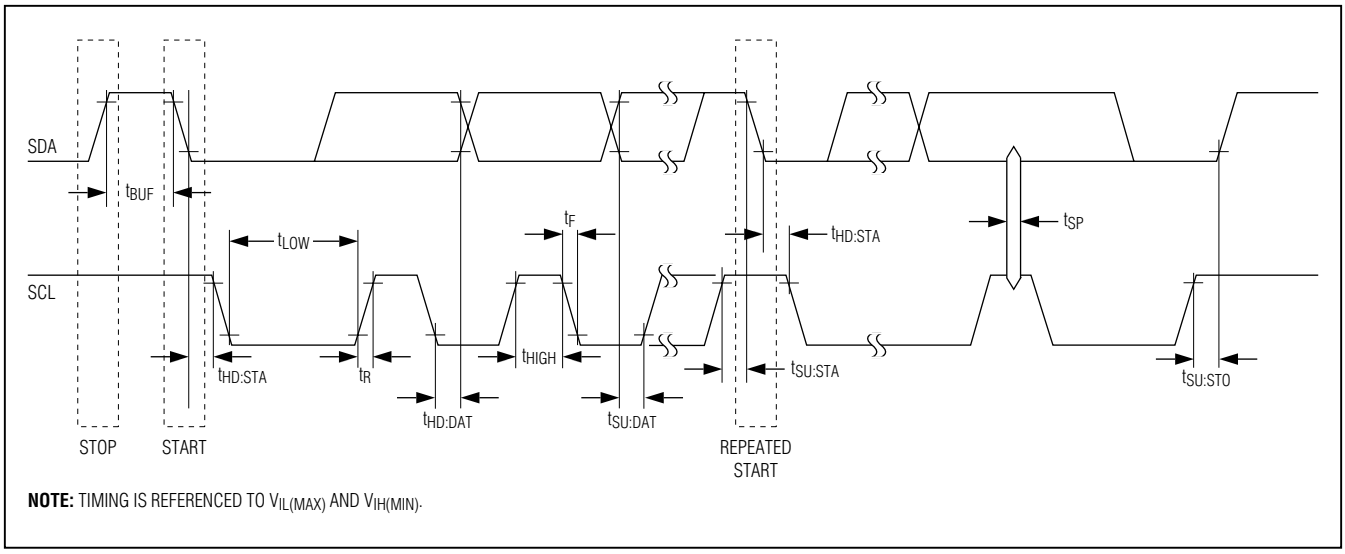
Note 2: This does not include pin input/output currents.

Note 3: Guaranteed by design.

Note 4: ADC has no missing codes.

Note 5: Commands MFR_MODE, STORE_DEFAULT_ALL, and RESTORE_DEFAULT_ALL require a bus-free time of 250ms.

I²C/SMBus Timing

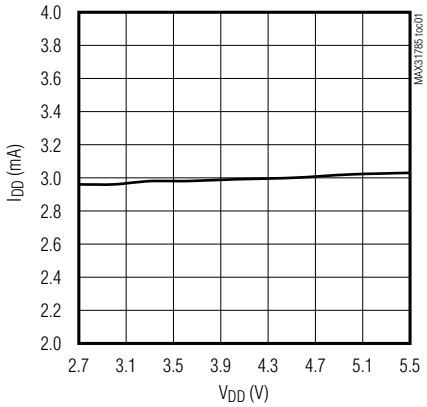


NOTE: TIMING IS REFERENCED TO $V_{IL(MAX)}$ AND $V_{IH(MIN)}$.

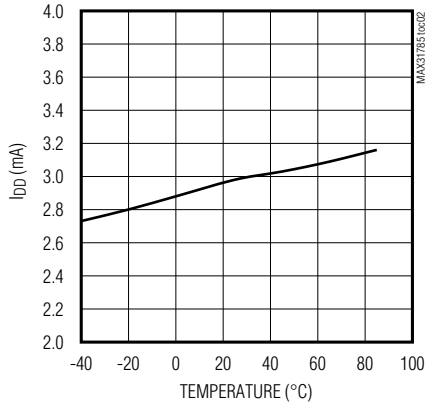
Typical Operating Characteristics

(T_A = +25°C, unless otherwise noted.)

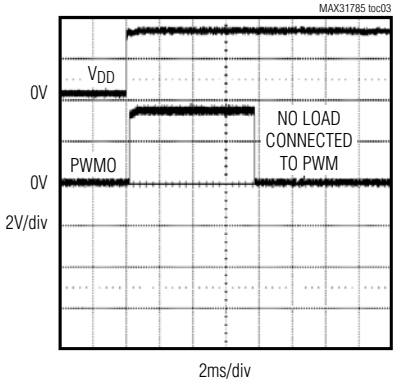
SUPPLY CURRENT vs. SUPPLY VOLTAGE (NO PWM SWITCHING)



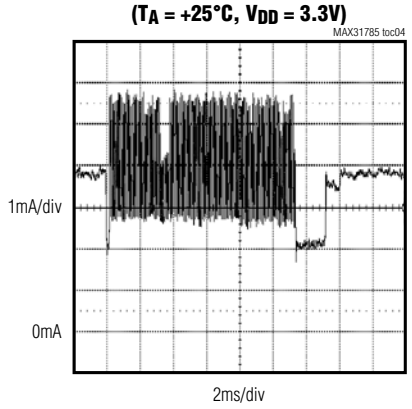
SUPPLY CURRENT vs. TEMPERATURE (V_{DD} = 3.3V, NO PWM SWITCHING)



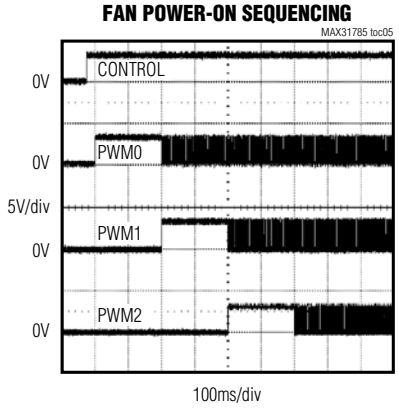
WEAK PULLUP VOLTAGE vs. TIME DURING POWER-UP



I_{DD} vs. TIME DURING A NONVOLATILE LOG WRITE (T_A = +25°C, V_{DD} = 3.3V)

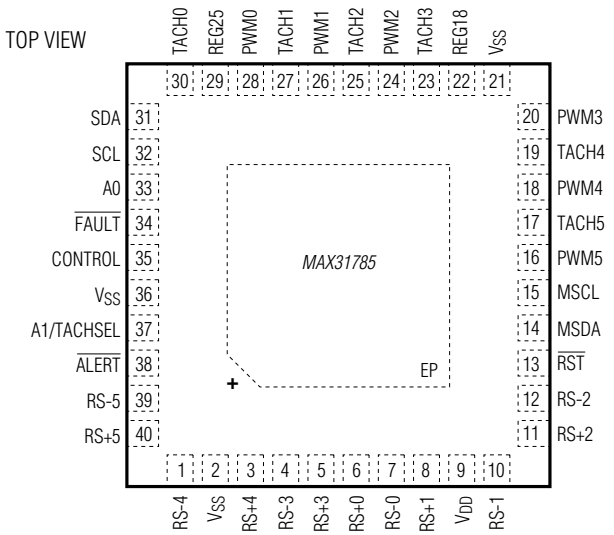


FAN POWER-ON SEQUENCING



6-Channel Intelligent Fan Controller

Pin Configuration



TQFN
(6mm × 6mm × 0.75mm)

Pin Description

PIN	NAME	FUNCTION
1	RS-4	Ground Reference for Thermal Diode or Remote Voltage ADC4 Measurement
2, 21, 36	V_{SS}	Digital-Supply Return Node (Ground)
3	RS+4	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-4
4	RS-3	Ground Reference for Thermal Diode or Remote Voltage ADC3 Measurement
5	RS+3	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-3
6	RS+0	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-0
7	RS-0	Ground Reference for Thermal Diode or Remote Voltage ADC0 Measurement
8	RS+1	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-1
9	V_{DD}	Supply Voltage. Bypass V_{DD} to V_{SS} with a 0.1µF capacitor.
10	RS-1	Ground Reference for Thermal Diode or Remote Voltage ADC1 Measurement
11	RS+2	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-2
12	RS-2	Ground Reference for Thermal Diode or Remote Voltage ADC2 Measurement
13	$\overline{RST}$	Reset Active-Low Input
14	MSDA	Master I^2C Data Input/Output. Open-drain output.
15	MSCL	Master I^2C Clock Input/Output. Open-drain output.
16	PWM5	Fan PWM Output #5. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
17	TACH5	Fan Tachometer Input
18	PWM4	Fan PWM Output #4. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
19	TACH4	Fan Tachometer Input

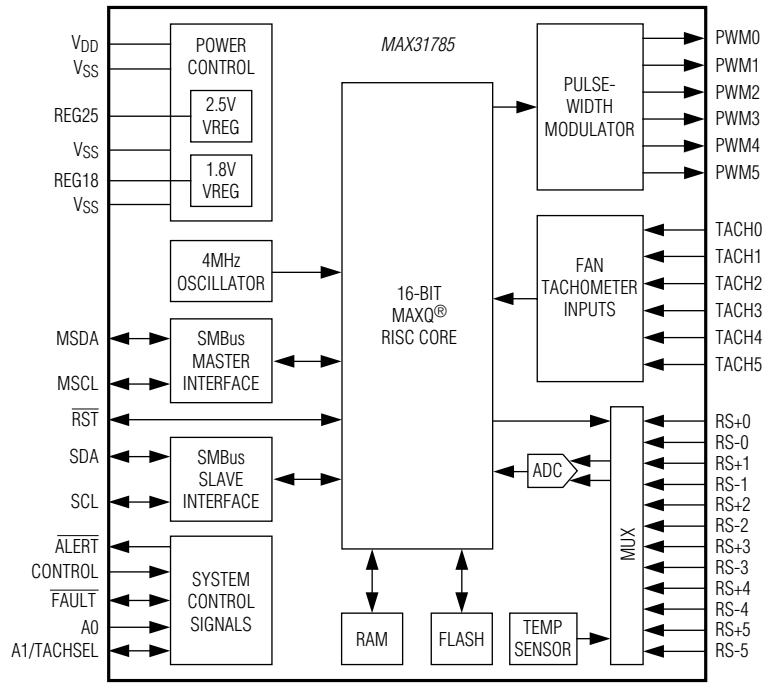
Pin Description (continued)

PIN	NAME	FUNCTION
20	PWM3	Fan PWM Output #3. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
22	REG18	Regulator for Low-Voltage Digital Circuitry. Bypass REG18 to V_{SS} with 1µF and 10nF capacitors. Do not connect other circuitry to this pin.
23	TACH3	Fan Tachometer Input
24	PWM2	Fan PWM Output #2. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
25	TACH2	Fan Tachometer Input
26	PWM1	Fan PWM Output #1. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
27	TACH1	Fan Tachometer Input
28	PWM0	Fan PWM Output #0. CMOS push-pull output. Low when the fan is disabled. A 100% duty cycle implies this pin is continuously high.
29	REG25	Regulator for Analog Circuitry. Bypass REG25 to V_{SS} with 1µF and 10nF capacitors. Do not connect other circuitry to this pin.
30	TACH0	Fan Tachometer Input
31	SDA	I^2C/SMBus-Compatible Input/Output
32	SCL	I^2C/SMBus-Compatible Clock Input
33	A0	SMBus Address 0 Input. This pin is sampled on device power-up to determine the SMBus address; connect a 100kΩ resistor from this pin to either V_{SS} or V_{DD} to set the address.
34	$\overline{FAULT}$	Active-Low, Open-Drain Fault Input/Output. If enabled with the MFR_FAULT_RESPONSE command, this pin is asserted during a fault condition (fan speed, overtemperature, overvoltage, or undervoltage). Also, if enabled with the MFR_FAULT_RESPONSE command, this pin is monitored and when it is asserted the fans can be configured to be forced to 100% PWM duty cycle. This pin is used to provide hardware fault control across multiple devices. This output is unconditionally deasserted when $\overline{RST}$ is asserted or the device is power cycled. This pin has a 50µs deglitch filter.
35	CONTROL	Global Fan-Off Control. When this pin is connected low, all fans are forced off (other functionality remains active). When this pin is connected high (or left open circuit), fans operate normally. This pin has a 50µs deglitch filter and contains a weak pullup.
37	A1/ TACHSEL	SMBus Address 1 Input/Dual Tach-Select Output. This dual-function pin is sampled on device power-up to determine the SMBus address; connect a 100kΩ resistor from this pin to either V_{SS} or V_{DD} to set the address. After device power-up, this pin becomes an output that selects between two tachometers in dual-fan applications.
38	$\overline{ALERT}$	Active-Low, Open-Drain Alert Output
39	RS-5	Ground Reference for Thermal Diode or Remote Voltage ADC5 Measurement
40	RS+5	Thermal Diode or Remote Voltage ADC Input, Measurement Relative to RS-5
—	EP	Exposed Pad (Bottom Side of Package). Connect EP to V_{SS}.

Note: *All pins except V_{DD}, V_{SS}, REG18, REG25, ADC, and the EP are high impedance with a 50µA pullup during device power-up and reset. After device reset, the weak pullup is removed, and the pin is configured as input or output.*

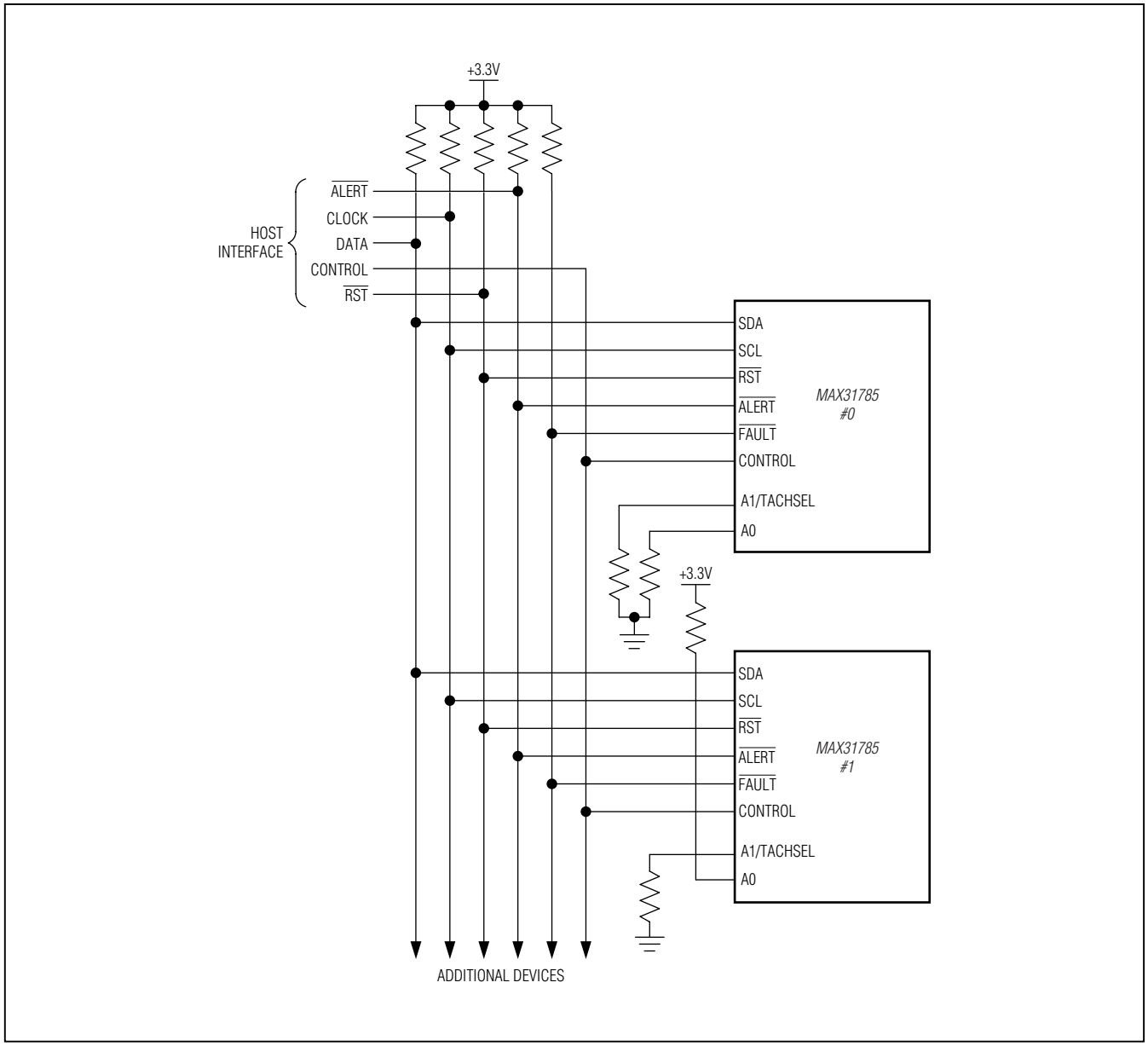
6-Channel Intelligent Fan Controller

Block Diagram



MAXQ is a registered trademark of Maxim Integrated Products, Inc.

Multiple Device Connection Diagram



6-Channel Intelligent Fan Controller

Detailed Description

The MAX31785 is a closed-loop fan controller. Automatic closed-loop fan control saves system power by operating the fans at the lowest possible speeds. Added benefits of slower fan speeds include lower audible noise, longer fan life, and reduced system maintenance. Based on a user-programmable lookup table (LUT), the device automatically adjust the speeds of the six independent fans based on one or more of the 11 available temperature sensors. Temperature offset can be added to individual sensors to compensate for thermal differences in a system. Alternately, an external host can manually command the fan speeds and the device automatically adjusts the fan speeds. The device can also monitor up to six remote voltages.

The device provides $\overline{\text{ALERT}}$ and $\overline{\text{FAULT}}$ output signals. Host communications are conducted through a PMBus™-compatible communications port. Address input connections are also provided to allow up to four MAX31785 devices to reside on the system's I/O bus.

Table 1. PMBus Command Codes

CODE	COMMAND NAME	TYPE	PAGE 0–5	PAGE 6–16	PAGE 17–22	PAGE 255	NO. OF BYTES	FLASH STORED (NOTE 2)	DEFAULT VALUE (NOTE 2)
			(NOTE 1)						
00h	PAGE	R/W Byte	R/W	R/W	R/W	R/W	1	N	00h
03h	CLEAR_FAULTS	Send Byte	W	W	W	W	0	N	—
10h	WRITE_PROTECT	R/W Byte	R/W	R/W	R/W	R/W	1	N	00h
11h	STORE_DEFAULT_ALL	Send Byte	W	W	W	W	0	N	—
12h	RESTORE_DEFAULT_ALL	Send Byte	W	W	W	W	0	N	—
19h	CAPABILITY	Read Byte	R	R	R	R	1	N	00h/10h
20h	VOUT_MODE	Read Byte	R	R	R	R	1	FIXED	40h
2Ah	VOUT_SCALE_MONITOR	R/W Word	—	—	R/W	—	2	Y	7FFFh
3Ah	FAN_CONFIG_1_2	R/W Byte	R/W	—	—	—	1	Y	00h
3Bh	FAN_COMMAND_1	R/W Word	R/W	—	—	—	2	Y	FFFFh
40h	VOUT_OV_FAULT_LIMIT	R/W Word	—	—	R/W	—	2	Y	7FFFh
42h	VOUT_OV_WARN_LIMIT	R/W Word	—	—	R/W	—	2	Y	7FFFh
43h	VOUT_UV_WARN_LIMIT	R/W Word	—	—	R/W	—	2	Y	0000h
44h	VOUT_UV_FAULT_LIMIT	R/W Word	—	—	R/W	—	2	Y	0000h
4Fh	OT_FAULT_LIMIT	R/W Word	—	R/W	—	—	2	Y	7FFFh
51h	OT_WARN_LIMIT	R/W Word	—	R/W	—	—	2	Y	7FFFh
78h	STATUS_BYTE	Read Byte	R	R	R	R	1	N	00h
79h	STATUS_WORD	Read Word	R	R	R	R	2	N	0000h
7Ah	STATUS_VOUT	Read Byte	—	—	R	—	1	N	00h
7Eh	STATUS_CML	Read Byte	R	R	R	R	1	N	00h
80h	STATUS_MFR_SPECIFIC	Read Byte	—	R	—	—	1	N	00h
81h	STATUS_FANS_1_2	Read Byte	R	—	—	—	1	N	00h
8Bh	READ_VOUT	Read Word	—	—	R	—	2	N	0000h
8Dh	READ_TEMPERATURE_1	Read Word	—	R	—	—	2	N	0000h
90h	READ_FAN_SPEED_1	Read Word	R	—	—	—	2	N	0000h
98h	PMBUS_REVISION	Read Byte	R	R	R	R	1	FIXED	11h
99h	MFR_ID	Read Byte	R	R	R	R	1	FIXED	4Dh
9Ah	MFR_MODEL	Read Byte	R	R	R	R	1	FIXED	53h
9Bh	MFR_REVISION	Read Word	R	R	R	R	2	FIXED	3030h
9Ch	MFR_LOCATION	Block R/W	R/W	R/W	R/W	R/W	8	Y	(Note 3)

PMBus is a trademark of SMIF, Inc.

Table 1. PMBus Command Codes (continued)

CODE	COMMAND NAME	TYPE	PAGE 0–5	PAGE 6–16	PAGE 17–22	PAGE 255	NO. OF BYTES	FLASH STORED (NOTE 2)	DEFAULT VALUE (NOTE 2)
			(NOTE 1)						
9Dh	MFR_DATE	Block R/W	R/W	R/W	R/W	R/W	8	Y	(Note 3)
9Eh	MFR_SERIAL	Block R/W	R/W	R/W	R/W	R/W	8	Y	(Note 3)
D1h	MFR_MODE	R/W Word	R/W	R/W	R/W	R/W	2	Y	0000h
D4h	MFR_VOUT_PEAK	R/W Word	—	—	R/W	—	2	N	0000h
D6h	MFR_TEMPERATURE_PEAK	R/W Word	—	R/W	—	—	2	N	8000h
D7h	MFR_VOUT_MIN	R/W Word	—	—	R/W	—	2	N	7FFFh
D9h	MFR_FAULT_RESPONSE	R/W Byte	R/W	R/W	R/W	—	1	Y	00h
DCh	MFR_NV_FAULT_LOG	Block Read	R	R	R	R	255	Y	(Note 4)
DDh	MFR_TIME_COUNT	Block R/W	R/W	R/W	R/W	R/W	4	N	(Note 5)
F0h	MFR_TEMP_SENSOR_CONFIG	R/W Word	—	R/W	—	—	2	Y	0000h
F1h	MFR_FAN_CONFIG	R/W Word	R/W	—	—	—	2	Y	0000h
F2h	MFR_FAN_LUT	Block R/W	R/W	—	—	—	32	Y	(Note 5)
F3h	MFR_READ_FAN_PWM	Read Word	R	—	—	—	2	N	0000h
F5h	MFR_FAN_FAULT_LIMIT	R/W Word	R/W	—	—	—	2	Y	0000h
F6h	MFR_FAN_WARN_LIMIT	R/W Word	R/W	—	—	—	2	Y	0000h
F7h	MFR_FAN_RUN_TIME	R/W Word	R/W	—	—	—	2	Y	0000h
F8h	MFR_FAN_PWM_AVG	R/W Word	R/W	—	—	—	2	Y	0000h
F9h	MFR_FAN_PWM2RPM	Block R/W	R/W	—	—	—	8	Y	(Note 5)

Note 1: Common commands are shaded. Access through any page results in the same device response.

Note 2: In the **Flash Stored** column, an “N” indicates that this parameter is not stored in flash memory when the STORE_DEFAULT_ALL command is executed and the value shown in the **Default Value** column is automatically loaded upon power-on reset or when the $\overline{RST}$ pin is asserted. A “Y” in the **Flash Stored** column indicates that the currently loaded value in this parameter is stored in flash memory when the STORE_DEFAULT_ALL command is executed and is automatically loaded upon power-on reset or when the $\overline{RST}$ pin is asserted and the value shown in the **Default Value** column is the value when shipped from the factory. “FIXED” in the **Flash Stored** column means this value is fixed at the factory and cannot be changed.

Note 3: The factory-set default value for this 8-byte block is 3130313031303130h.

Note 4: The factory-set default value for the complete block of the MFR_NV_FAULT_LOG is FFh.

Note 5: The factory-set default value for the complete block is 00h.

6-Channel Intelligent Fan Controller

Table 2. PMBus/SMBus Serial-Port Address

A1	A0	7-BIT SLAVE ADDRESS
100kΩ to V_{SS}	100kΩ to V_{SS}	1010 010 (A4h)
	100kΩ to V_{DD}	1010 011 (A6h)
100kΩ to V_{DD}	100kΩ to V_{SS}	1010 100 (A8h)
	100kΩ to V_{DD}	1010 101 (AAh)

Address Select

On device power-up, the device samples the A0 and A1 pins to determine the PMBus/SMBus serial-port address.

SMBus/PMBus Operation

The device implements the PMBus command structure using the SMBus format. The structure of the data flow between the host and the slave is shown below for several different types of transactions. Data is sent most significant bit (MSB) first.

SMBus/PMBus Communication Examples

READ WORD FORMAT

1	7	1	1	8	1	1	7	1	1	8	1	8	1	1
S	SLAVE ADDRESS	W	A	COMMAND CODE	A	Sr	SLAVE ADDRESS	R	A	DATA BYTE LOW	A	DATA BYTE HIGH	NA	P

READ BYTE FORMAT

1	7	1	1	8	1	1	7	1	1	8	1	1
S	SLAVE ADDRESS	W	A	COMMAND CODE	A	Sr	SLAVE ADDRESS	R	A	DATA BYTE	NA	P

WRITE WORD FORMAT

1	7	1	1	8	1	8	1	8	1	1
S	SLAVE ADDRESS	W	A	COMMAND CODE	A	DATA BYTE LOW	A	DATA BYTE HIGH	A	P

WRITE BYTE FORMAT

1	7	1	1	8	1	8	1	1
S	SLAVE ADDRESS	W	A	COMMAND CODE	A	DATA BYTE	A	P

SEND BYTE FORMAT

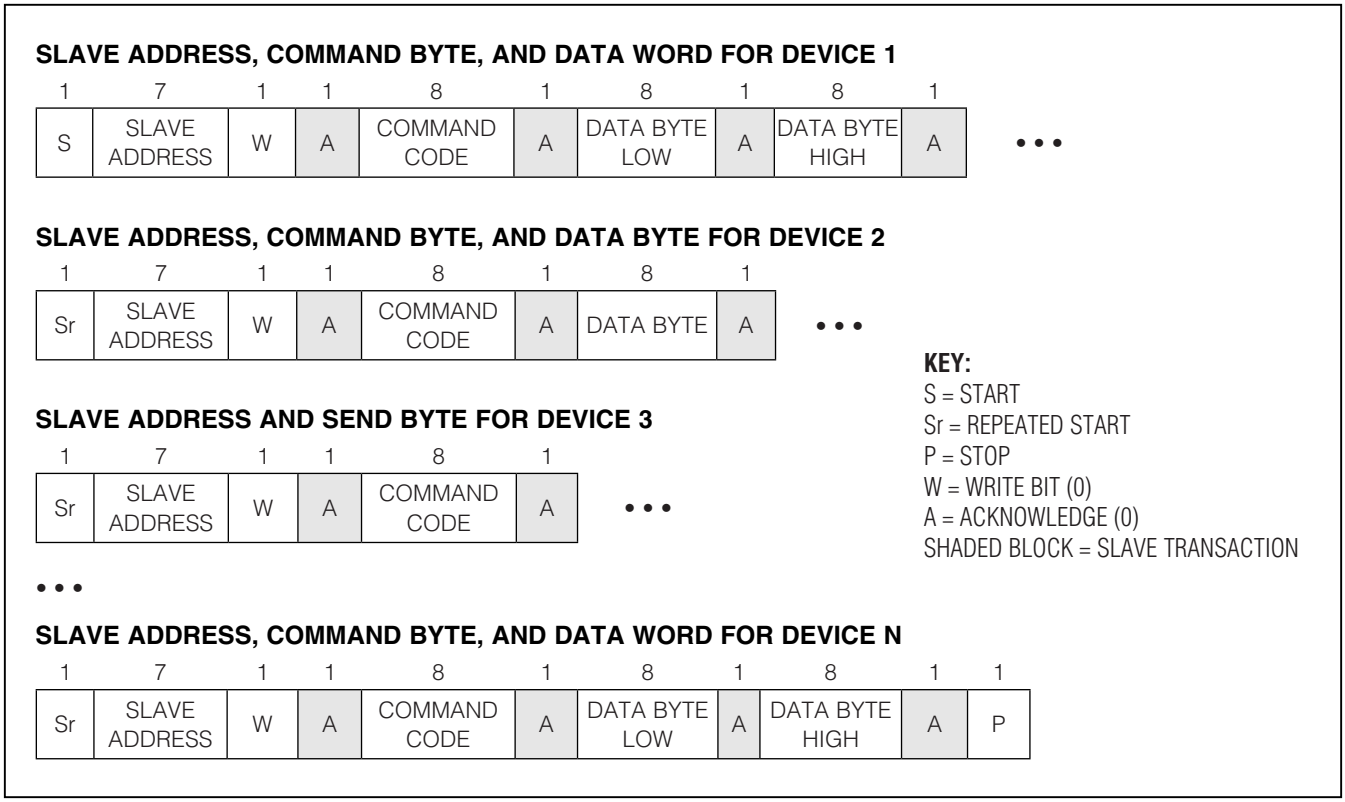
1	7	1	1	8	1	1
S	SLAVE ADDRESS	W	A	COMMAND CODE	A	P

KEY:
S = START
Sr = REPEATED START
P = STOP
W = WRITE BIT (0)
R = READ BIT (1)
A = ACKNOWLEDGE (0)
NA = NOT ACKNOWLEDGE (1)
SHADED BLOCK = SLAVE TRANSACTION

Group Command

The device supports the group command. With the group command, a host can write different data to multiple devices on the same serial bus with one long continuous data stream. All the devices addressed during this transaction wait for the host to issue a STOP before beginning to respond to the command.

Group Command Write Format



Addressing

The device responds to receiving its fixed slave address by asserting an acknowledge (ACK) on the bus. The device does not respond to a general call address; it only responds when it receives its fixed slave address. The only exception to this operation is if the $\overline{\text{ALERT}}$ output is enabled (ALERT bit = 1 in MFR_MODE) and $\overline{\text{ALERT}}$ has been asserted. When this condition occurs, the device only recognizes the alert response address (0001 100, 18h). See the *ALERT and Alert Response Address (ARA)* section for more details.

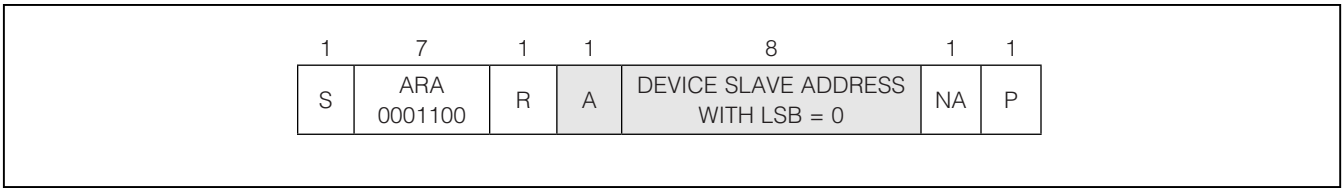
ALERT and Alert Response Address (ARA)

If the $\overline{\text{ALERT}}$ output is enabled (ALERT bit = 1 in MFR_MODE), when a fault occurs the device asserts the $\overline{\text{ALERT}}$ signal and then waits for the host to send the alert response address (ARA) as shown in the *Alert Response Address (ARA) Byte Format* section. **While waiting for the ARA, the device does not respond to its fixed slave address.**

When the ARA is received and the device is asserting $\overline{\text{ALERT}}$, the device ACKs it and then attempts to place its fixed slave address on the bus by arbitrating the bus, since another device could also try to respond to the ARA. The rules of arbitration state that the lowest address device wins. If the device wins the arbitration, it deasserts $\overline{\text{ALERT}}$ and begins to respond to its fixed slave address. If the device loses arbitration, it keeps $\overline{\text{ALERT}}$ asserted and waits for the host to once again send the ARA.

6-Channel Intelligent Fan Controller

Alert Response Address (ARA) Byte Format



Host Sends or Reads Too Few Bits

If for any reason the host does not complete writing a full byte or reading a full byte from the device before a START or STOP is received, the device does the following:

1) Ignores the command.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the DATA_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Host Sends or Reads Too Few Bytes

For each supported command, the device expects a fixed number of bytes to be written or read from the device. If for any reason fewer than the expected number of bytes is written to or read from the device, the device completely ignores the command and takes no action.

Host Sends Too Many Bytes or Bits

For each supported command, the device expects a fixed number of bytes to be written to the device. If for any reason more than the expected number of bytes or bits is written to the device, the device does the following:

1) Ignores the command.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the DATA_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Host Reads Too Many Bytes or Bits

For each supported command, the device expects a fixed number of bytes to be read from the device. If for any reason more than the expected number of bytes or bits is read from the device, the device does the following:

1) Sends all ones (FFh) as long as the host keeps acknowledging.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the DATA_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Host Sends Improperly Set Read Bit in the Slave Address Byte

If the device receives the R/$\overline{\text{W}}$ bit in the slave address set to one immediately preceding the command code, the device does the following (note this does not apply to ARA):

1) ACKs the address byte.
2) Sends all ones (FFh) as long as the host keeps acknowledging.
3) Sets the CML bit in STATUS_BYTE.
4) Sets the CML bit in STATUS_WORD.
5) Sets the DATA_FAULT bit in STATUS_CML.
6) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Unsupported Command Code Received

If the host sends the device a command code that it does not support, or if the host sends a command code that is not supported by the current PAGE setting, the device does the following:

1) Ignores the command.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the COMM_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Invalid Data Received

The device checks the PAGE and WRITE_PROTECT command codes for valid data. If the host writes a data value that is invalid, the device does the following:

1) Ignores the command.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the DATA_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Host Reads from a Write-Only Command

When a read request is issued to a write-only command (CLEAR_FAULTS, STORE_DEFAULT_ALL, RESTORE_DEFAULT_ALL), the device does the following:

1) ACKs the address byte.
2) Ignores the command.
3) Sends all ones (FFh) as long as the host keeps acknowledging.
4) Sets the CML bit in STATUS_BYTE.
5) Sets the CML bit in STATUS_WORD.
6) Sets the DATA_FAULT bit in STATUS_CML.
7) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

Host Writes to a Read-Only Command

When a write request is issued to a read-only command, the device does the following:

1) Ignores the command.
2) Sets the CML bit in STATUS_BYTE.
3) Sets the CML bit in STATUS_WORD.
4) Sets the COMM_FAULT bit in STATUS_CML.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled).

SMBus Timeout

If during an active SMBus communication sequence the SCL signal is held low for greater than the timeout duration (t_{TO}), the device terminates the sequence and resets the serial bus. It takes no other action. No status bits are set.

PMBus Operation

From a software perspective, the device appears as a PMBus device capable of executing a subset of PMBus commands. A PMBus 1.1-compliant device uses the SMBus version 1.1 for transport protocol and responds to the SMBus slave address. In this data sheet, the term SMBus is used to refer to the electrical characteristics of the PMBus communication using the SMBus physical layer. The term PMBus is used to refer to the PMBus command protocol. The device employs a number of standard SMBus protocols to program output voltage and warning/faults thresholds, read monitored data, and provide access to all manufacturer-specific commands.

The device supports the group command. The group command is used to send commands to more than one PMBus device. It is not required that all the devices receive the same command. However, no more than one command can be sent to any one device in one group command packet. The group command must not be used with commands that require receiving devices to respond with data, such as the STATUS_BYTE command. When the device receives a command through this protocol, it immediately begins execution of the received command after detecting the STOP condition.

The device supports the PAGE command and uses it to select which individual channel to access. When a data word is transmitted, the lower order byte is sent first and the higher order byte is sent last. Within any byte, the most significant bit (MSB) is sent first and the least significant bit (LSB) is sent last.

PMBus Protocol Support

The device supports a subset of the commands defined in the *PMBus™ Power System Management Protocol Specification Part II - Command Language, Revision 1.1*. For detailed specifications and the complete list of PMBus commands, refer to Part II of the PMBus specification available at www.PMBus.org. The supported PMBus commands and the corresponding device behavior are described in this document. All data values are represented in DIRECT format, unless otherwise stated. Whenever the PMBus specification refers to the PMBus device, it is referring to the MAX31785 operating in conjunction with a fan. While the command can call for turning on or turning off the PMBus device, the MAX31785 always remains on to continue communicating with the PMBus master, and the MAX31785 transfers the command to the fan accordingly.

Data Format

Voltage data for commanding or reading the output voltage or related parameters (such as the overvoltage threshold) is presented in DIRECT format. DIRECT format data is a 2-byte, two's complement binary value. DIRECT format data can be used with any command that sends or reads a parametric value. The DIRECT format uses an equation and defined coefficients to calculate the desired values. Table 3 shows the coefficients used by the device.

Table 3. PMBus Command Code Coefficients

PARAMETER	COMMANDS	UNITS	RESOLUTION	MAX	m	b	R
Voltage	VOUT_OV_FAULT_LIMIT VOUT_OV_WARN_LIMIT VOUT_UV_WARN_LIMIT VOUT_UV_FAULT_LIMIT READ_VOUT MFR_VOUT_PEAK MFR_VOUT_MIN	mV	1	32,767	1	0	0
Voltage Scaling	VOUT_SCALE_MONITOR	—	1/32,767	1	32,767	0	0
Temperature	OT_FAULT_LIMIT OT_WARN_LIMIT READ_TEMPERATURE_1 MFR_TEMPERATURE_PEAK	°C	0.01	327.67	1	0	2
Fan Speed	READ_FAN_SPEED_1 FAN_COMMAND_1 MFR_FAN_FAULT_LIMIT MFR_FAN_WARN_LIMIT	RPM	1	32,767	1	0	0
	FAN_COMMAND_1 MFR_READ_FAN_PWM MFR_FAN_FAULT_LIMIT MFR_FAN_WARN_LIMIT MFR_FAN_PWM_AVG	%	0.01	327.67	1	0	2

Interpreting Received DIRECT Format Values

The host system uses the following equation to convert the value received from the PMBus device—in this case, the MAX31785—into a reading of volts, degrees Celsius, or other units as appropriate:

$$X = (1/m) \times (Y \times 10^{-R} - b)$$

where X is the calculated, real world value in the appropriate units (V, °C, etc.); m is the slope coefficient; Y is the 2-byte, two's complement integer received from the PMBus device; b is the offset; and R is the exponent.

Sending a DIRECT Format Value

To send a value, the host must use the below equation to solve for Y:

$$Y = (mX + b) \times 10^{R}$$

where Y is the 2-byte, two's complement integer to be sent to the unit; m is the slope coefficient; X is the real world value, in units such as volts, to be converted for transmission; b is the offset; and R is the exponent.

The following example demonstrates how the host can send and retrieve values from the device. Table 4 shows the coefficients used in the following parameters.

Table 4. Coefficients for DIRECT Format Value

COMMAND CODE	COMMAND NAME	m	b	R
8Bh	READ_VOUT	1	0	0

If the host received a value of 0D89h on a READ_VOUT command, this is equivalent to:

$$X = (1/1) \times (0D89h \times 10^{-(-0)} - 0) = 3465mV = 3.465V$$

All voltage-related parameters of PMBus devices are reported as positive values. It is up to the system to know that a particular output is negative if that is of interest to the system. All output voltage-related commands use 2 data bytes.

Table 5. Device Parametric Monitoring States

PARAMETER	REQUIRED CONDITIONS FOR ACTIVE MONITORING	ACTION DURING A FAULT
Overvoltage	Voltage Monitoring Enabled (ADC_ENABLE in MFR_MODE = 1)	Continue Monitoring
Undervoltage	Voltage Monitoring Enabled (ADC_ENABLE in MFR_MODE = 1)	Continue Monitoring
Overtemperature	Temp Sensor Enabled (ENABLE in MFR_TEMP_SENSOR_CONFIG = 1)	Continue Monitoring
Fan Speed	Fan Enabled (Bit 7 in FAN_CONFIG_1_2 = 1)	Continue Monitoring

Fault Management and Reporting

For reporting faults/warnings to the host on a real-time basis, the device asserts the open-drain $\overline{ALERT}$ pin (if enabled in MFR_MODE) and sets the appropriate bit in the various status registers. On recognition of the $\overline{ALERT}$ assertion, the host or system manager is expected to poll the I2C bus to determine the device asserting $\overline{ALERT}$. The host sends the SMBus ARA (0001 100). The device ACKs the SMBus ARA, transmits its slave address, and deasserts $\overline{ALERT}$. The system controller then communicates with PMBus commands to retrieve the fault/warning status information from the device.

See the individual command sections for more details. Faults and warnings that are latched in the status registers are cleared when any one of the following conditions occurs:

- A CLEAR_FAULTS command is received.
- The $\overline{RST}$ pin is toggled.
- Bias power to the device is removed and then reapplied.

The device responds to fault conditions according to the manufacturer fault response command (MFR_FAULT_RESPONSE). This command byte determines how the device should respond to each particular fault. Table 5 illustrates the required conditions and fault actions for specific parameters.

System Watchdog Timer

The device uses an internal watchdog timer that is internally reset every 10ms. In the event that the device is locked up and this watchdog reset does not occur after 500ms, the device automatically resets. After the reset occurs, the device reloads all configuration values that were stored to flash and begins normal operation. After the reset, the device also does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and MFR bits in STATUS_WORD.
3) Sets the WATCHDOG bit in STATUS_MFR_SPECIFIC.
4) Notifies the host through $\overline{ALERT}$ assertion (if enabled in MFR_MODE).

Temperature Sensor Operation

The device can monitor up to 11 different temperature sensors. It can monitor up to four remote I2C-based temperature sensors plus six remote diodes and its own internal temperature sensor. Each of the enabled temperature sensors is measured once a second. The remote diode and internal temperature sensors are averaged eight times to reduce the affect of noise. Each time the device attempts to read a temperature sensor it checks for faults. For the remote diode, a fault is defined as reading greater than +160°C or less than -60°C. For the internal temperature sensor, a fault is defined as reading greater than +130°C or less than -60°C. For the I2C temperature sensors, a fault is defined as a communication access failure. Temperature sensor faults are reported by setting the temperature reading to 7FFFh. A temperature sensor fault results in the setting of the TEMPERATURE bit in STATUS_BYTE and STATUS_WORD and $\overline{ALERT}$ is asserted (if enabled in MFR_MODE). No bits are set in STATUS_MFR_SPECIFIC.

The temperatures do not have to be used to control the fan speed. They can be enabled and used for temperature monitoring only. Reading disabled temperature sensors returns a fixed value of 0000h.

The remote diode temperature sensor can support either npn or pnp transistors. The device automatically cancels the series resistance that can affect remote diodes that are located far from the device.

The device can control up to four DS75LV digital temperature sensors. The A0, A1, and A2 pins on the DS75LV should be configured as shown in Table 6. The thermostat function on the DS75LV is not used and thus the O.S. output should be left open circuit.

Table 6. DS75LV Address Pin Configurations

PAGE	MAX31785 I2C TEMP SENSOR	DS75LV ADDRESS PIN CONFIGURATION		
		A2	A1	A0
13	TEMP SENSOR I2C 0	0	0	0
14	TEMP SENSOR I2C 1	0	0	1
15	TEMP SENSOR I2C 2	0	1	0
16	TEMP SENSOR I2C 3	0	1	1

6-Channel Intelligent Fan Controller

Fan Control Operation

Fan control has four operational modes. The mode is determined by the combination of FAN_COMMAND_1 and bit 6 of FAN_CONFIG_1_2 (see Table 7). Fan control can be disabled by setting bit 7 in FAN_CONFIG_1_2 to zero.

Dual Fan Applications

In dual fan applications operating in RPM mode, the tachometer selected when TACHSEL = 0 is close-loop-controlled to the target RPM. Once PWM ramping is complete, TACHSEL toggles between the two tachometers every 500ms for monitoring purposes. The slower of the two tachometer signals is reported by READ_FAN_SPEED_1 and is used as a comparison for fan faults and warning. If one of the two tachometer signals operate at a slower speed, it is recommended that the slower tachometer be presented to the TACH input when TACHSEL = 0. If more than one fan channel is configured for dual fan operation, these fans must complete ramping to the target fan speed before the device begins to toggle the TACHSEL output and monitor both tachometers.

Automatic Fan Control Operation

In the automatic mode, the fan is controlled in a closed loop based on the controlling temperature (the highest postnormalized temperature reading) and the associated fan control PWM duty cycle (in %) or fan speed (in RPM). These parameters are assigned in the fan lookup table (LUT). See the MFR_FAN_LUT description for configuration details. When a controlling temperature exceeds the temperature level programmed in the LUT, the device outputs a PWM duty cycle or adjusts the fan speed associated with that temperature. See Figure 1 for an example.

One or all of the 11 available temperature sensors can be used to control the fan speed. Each temperature sensor has an offset adjustment that allows monitoring specific temperature zones with different thermal characteristics. In Figure 1, remote diode temperature sensor 0 is monitoring a zone that is 15°C more sensitive than the zone that the I²C temperature sensor 3 is measuring. To keep the audible noise and fan power consumption as low as possible, the device allows each temperature sensor to have a temperature offset added. This allows temperature zones with different thermal profiles to control the fan at the lowest possible speed to maintain the required temperature.

If no temperature sensors are assigned to control the fan, the output fan PWM signal is ramped to 100% duty cycle.

In Figure 2, at temperature sample 1, the required fan speed is at the level associated with temperature level 2 and since temperature sample 1 is above temperature level 3, the fan PWM duty cycle needs to be increased to increase the fan speed. The device increases the fan PWM duty cycle at a rate controlled by the RAMP bits in the MFR_FAN_CONFIG command code. If the PWM duty cycle has not reached the target value before the temperature sample detects that a new PWM target value is needed, the device stops moving toward the old target and starts moving to the new target according to the programmed ramp rate.

Table 7. Fan Control Operation Modes

FAN CONTROL MODE	FAN OPERATIONAL DETAILS	BIT 6 OF FAN_CONFIG_1_2	VALUE IN FAN_COMMAND_1
Manual PWM	External host controls the fan speed by directly setting the fan PWM duty-cycle values.	0	0000h to 7FFFh
Manual RPM	External host controls the fan speed by setting target fan speed values. The device reads the actual fan speed and close loop adjusts the output fan PWM to match the target fan speed.	1	0000h to 7FFFh
Automatic PWM	The device sets the output PWM based on the fan LUT that maps the temperature sensor readings to the required fan PWM duty-cycle values.	0	8000h to FFFFh
Automatic RPM	The device reads the actual fan speed and close loop adjusts the output fan PWM to match the target fan speed based on the fan LUT that maps the temperature sensor readings to the required fan speed.	1	8000h to FFFFh

Note: *The RPM modes should only be used with fans that provide a tachometer output.*

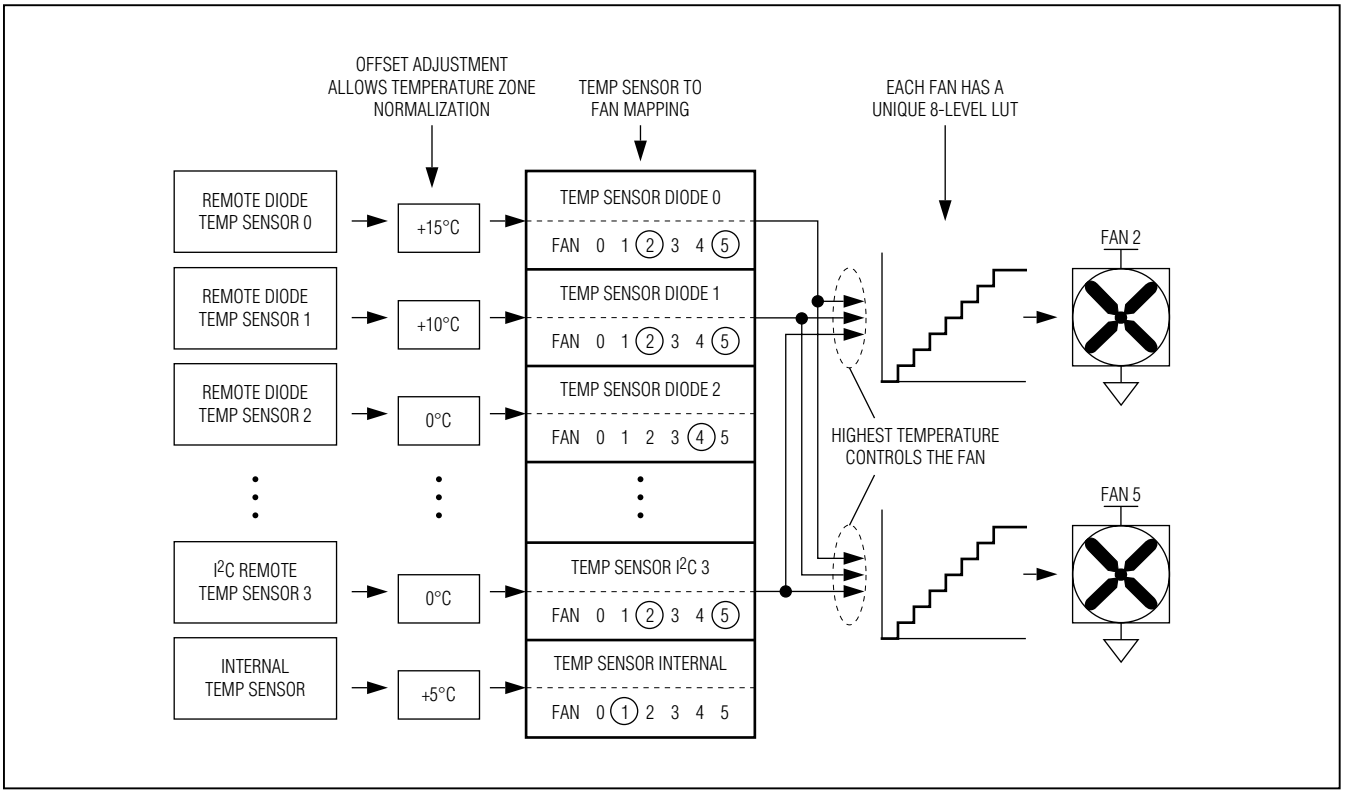


Figure 1. Automatic Fan Control

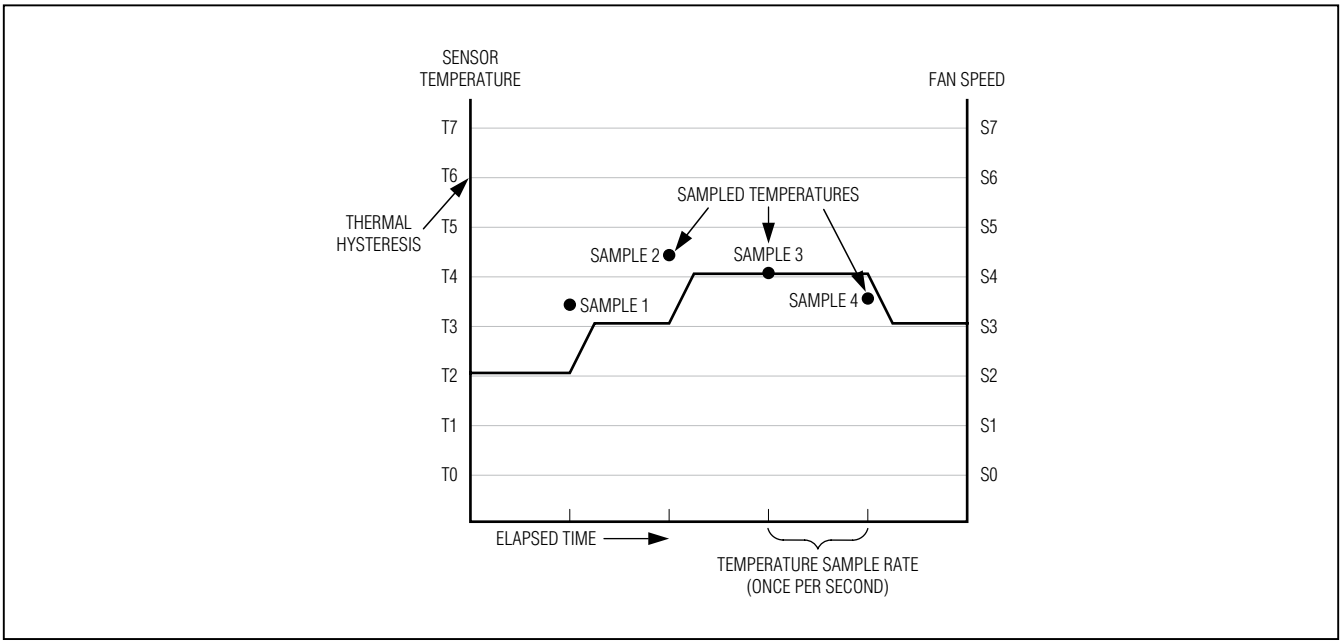


Figure 2. Fan Speed Example

6-Channel Intelligent Fan Controller

At temperature sample 2, the temperature has increased to greater than temperature level 4 so again the PWM duty cycle must be increased. At temperature sample 3, the temperature has dropped but not below the thermal hysteresis level (which is set by the HYS bits in the MFR_FAN_CONFIG command code), so the fan speed remains at level 4. At temperature sample 4, the temperature has dropped below the hysteresis point so the PWM duty cycle is decreased.

Pulse Stretching

In some 3-wire fan applications, when the supply power is interrupted to control fan speed, the tachometer signal is not available. Some fan controllers periodically stretch the PWM signal to allow the tachometer to be accurately detected. Pulse stretching can create audible noise. **This device does not implement pulse stretching.** Hence, the RPM fan modes that require a reliable tachometer signal to be available at all times should not be used in applications that switch power to the fan to control the speed of the fan. For example, low-side switching of a 3-wire fan should not use either of the RPM fan modes.

Fan Spin-Up

Figure 3 shows the fan spin-up process. When the fan is spinning up, the number of revolutions is checked every 200ms for up to 2s. When the number of cumulated revolutions is greater than or equal to the spin-up relaxation criteria, the fan passes spin-up. If the fan has a locked rotor output, fan spin-up passes when the locked rotor signal is no longer asserted.

When spin-up passes, the device forces the fan PWM with a 40% duty cycle. The 40% duty cycle is maintained until the next temperature conversion is completed, which occurs once a second. After the temperature conversion, the device enters either manual mode operation or automatic operation using MFR_FAN_LUT.

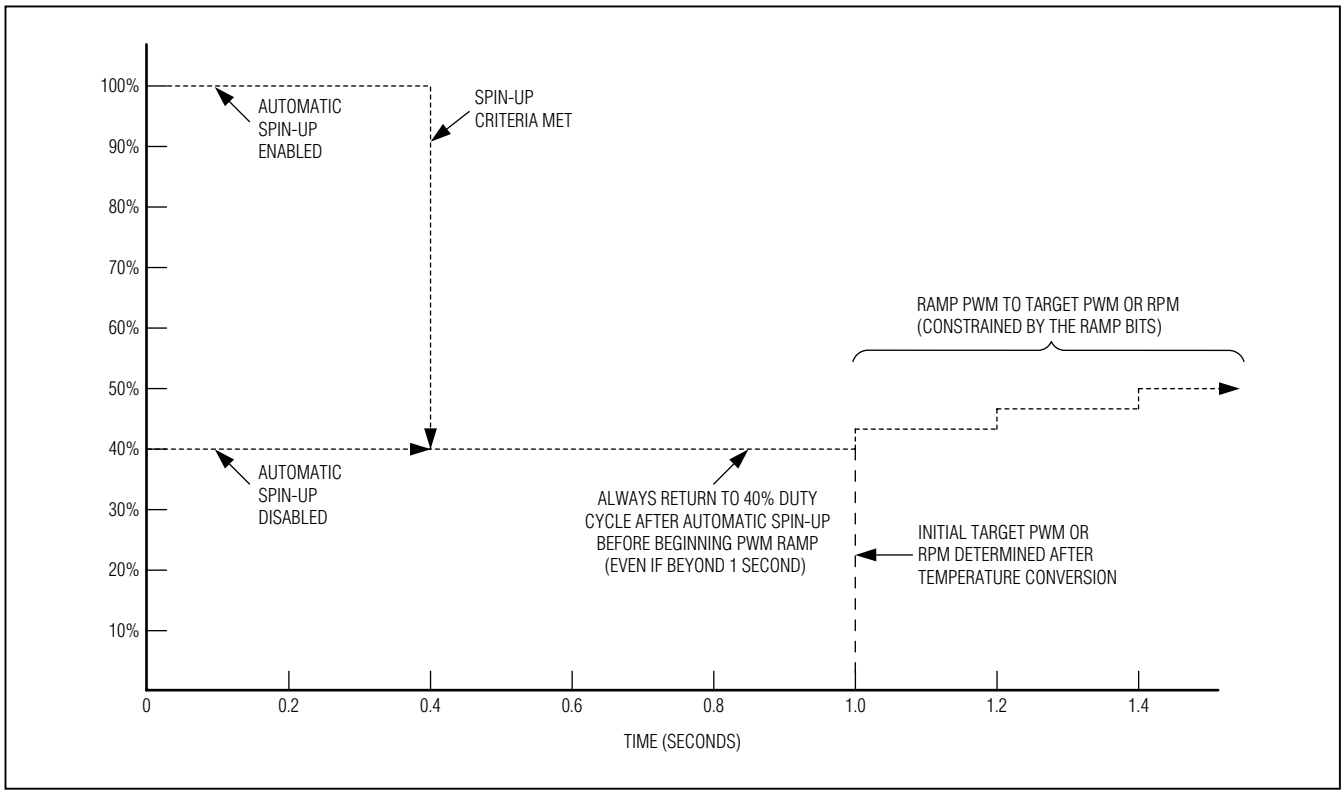


Figure 3. Fan Spin-Up

Fan Power-On Sequencing

Multiple fans are not started at the same time. Fans are started in increments of 200ms to relieve the strain on the power supply. This sequencing occurs both at initial fan power-up and when fans are turned off and on during normal operation.

Fan Health-Meter Function

The device has a fan-health-diagnostic function to help predict the remaining life of the fan. This function compares the measured vs. expected fan speed for the applied PWM duty cycle to determine if the fan is operating normally. This function is enabled with the HEALTH bit (bit 4) in MFR_FAN_CONFIG. When the fan-health-meter function is enabled, it runs once per second for each enabled fan. If the fan's PWM duty cycle has been stable for 30s, the measured RPM is compared to the expected RPM and the correct fan-health-color bit (GREEN, ORANGE, or RED) is set in STATUS_FAN_1_2. If the duty cycle has been unstable for greater than 10 minutes, the YELLOW bit is set in STATUS_FANS_1_2.

The expected fan-speed profile is entered with the MFR_FAN_PWM2RPM command. With this command, a table is filled with four preassigned PWM duty cycle (40%, 60%, 80%, and 100%) vs. expected fan-speed values, as shown in Figure 4. The comparison of the measured RPM to expected RPM is performed when the

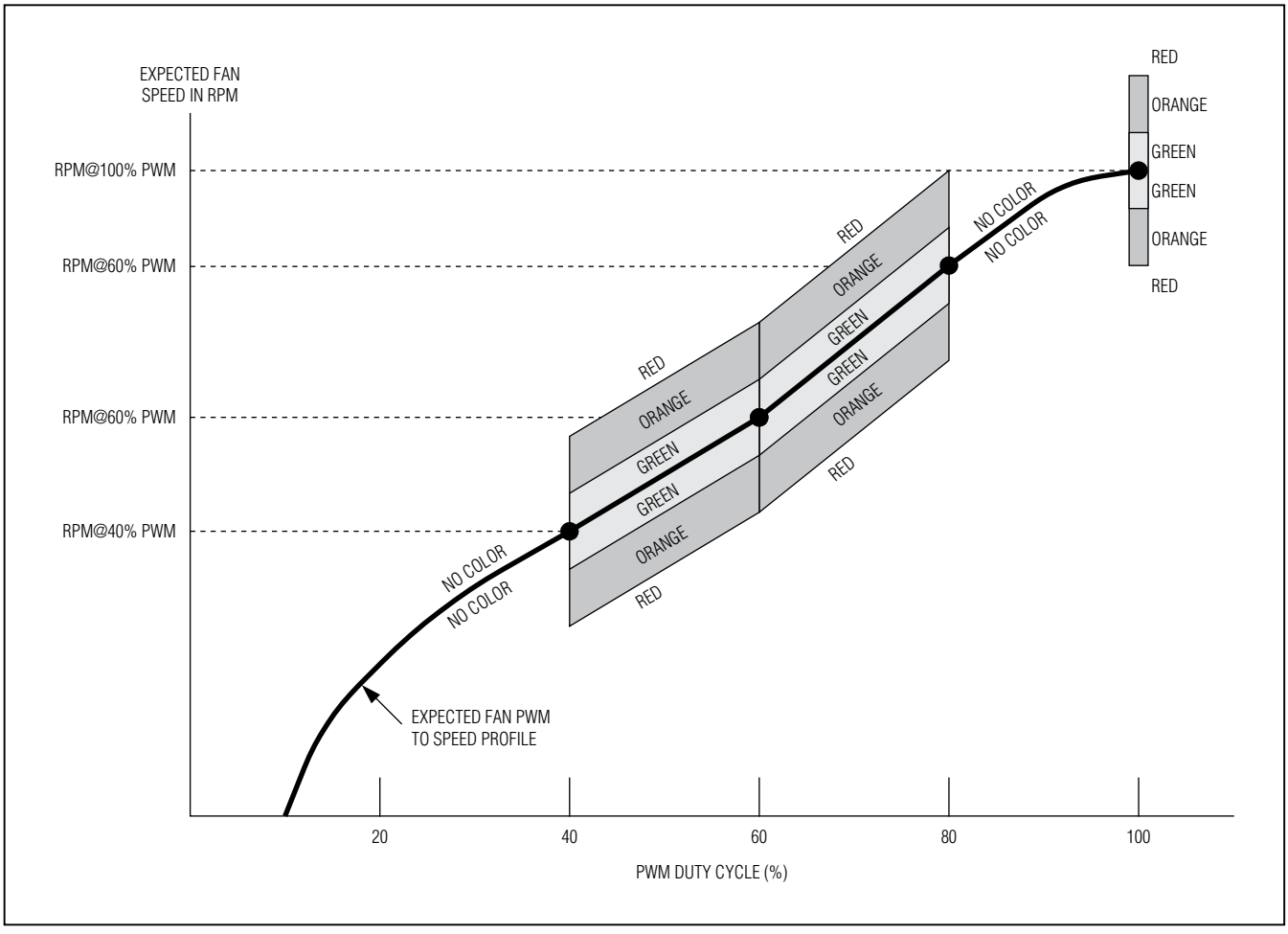


Figure 4. Fan PWM to RPM Example

6-Channel Intelligent Fan Controller

fan's PWM duty cycle is between 40% and 80%. The masured fan speed is also compared to the expected at 100% duty cycle. In the 40% to 80% range, a fan's expected fan-speed profile is approximately linear. At duty cycles outside this range (with the exception of 100%), the shape of the expected fan-speed profile varies too much between fans, and it is not possible to predict an expected speed.

A fan-health-color bit in STATUS_FANS_1_2 is set based upon the difference between the measured and expected fan speed and the FAN_HEALTH_CRITERIA bits in MFR_MODE, as shown in Table 8. If the fan PWM duty cycle is less than 40% or between 80% and 100%, a fan-health color cannot be assigned and all the color bits are cleared. If the fan is unstable, the YELLOW color bit is set.

Table 8. Fan Health Conditions

LOWER NIBBLE OF STATUS_FAN_1_2	FAN_HEALTH_CRITERIA BITS IN MFR_MODE	MONITORED FAN SPEED VS. EXPECTED TARGET
GREEN	00	Within 10%
	01	Within 10%
	10	Within 15%
	11	Within 15%
YELLOW	—	Fan speed is not stable
ORANGE	00	Between 10% and 15%
	01	Between 10% and 20%
	10	Between 15% and 20%
	11	Between 15% and 25%
RED	00	Over 15%
	01	Over 20%
	10	Over 20%
	11	Over 25%

PMBus Commands

A summary of the PMBus commands supported by the device are described in the following sections.

PAGE (00h)

The device can control up to six fans, up to 11 temperature sensors, and measure up to six remote voltages using one PMBus (I^2C) address. Send the PAGE command with data 0 to 22 to select which fan, temperature sensor, or remote voltage fan is affected by all the PMBus commands shown in Table 1. Not all commands are supported within each page. If an unsupported command is received, the CML status bit is set. Some commands are common, meaning that any selected page has the same effect on and the same response from the device.

Set the PAGE to 255 when it is desired that the following PMBus commands should apply to all pages at the same time.

Table 9. Page Commands

PAGE (DEC)	ASSOCIATED CONTROL
0	Fan Connected to PWM 0
1	Fan Connected to PWM 1
2	Fan Connected to PWM 2
3	Fan Connected to PWM 3
4	Fan Connected to PWM 4
5	Fan Connected to PWM 5
6	Remote Thermal Diode Connected to ADC 0
7	Remote Thermal Diode Connected to ADC 1
8	Remote Thermal Diode Connected to ADC 2
9	Remote Thermal Diode Connected to ADC 3
10	Remote Thermal Diode Connected to ADC 4
11	Remote Thermal Diode Connected to ADC 5
12	Internal Temperature Sensor
13	Remote I^2C Temperature Sensor with Address 0
14	Remote I^2C Temperature Sensor with Address 1
15	Remote I^2C Temperature Sensor with Address 2
16	Remote I^2C Temperature Sensor with Address 3
17	Remote Voltage Connected to ADC 0
18	Remote Voltage Connected to ADC 1
19	Remote Voltage Connected to ADC 2
20	Remote Voltage Connected to ADC 3
21	Remote Voltage Connected to ADC 4
22	Remote Voltage Connected to ADC 5
23 to 254	Reserved
255	Applies to All Pages

CLEAR_FAULTS (03h)

The CLEAR_FAULTS command is used to clear any fault or warning bits in the status registers that have been set. This command clears all bits simultaneously. If a fault is still present after the CLEAR_FAULTS command is executed, the fault status bit is set again and the host is also notified by asserting $\overline{\text{ALERT}}$ (if enabled in MFR_MODE). This command is write-only. There is no data byte for this command.

WRITE_PROTECT (10h)

The WRITE_PROTECT command is used to provide protection against accidental changes to the device operating memory. All supported commands can have their parameters read, regardless of the WRITE_PROTECT settings. The WRITE_PROTECT message content is described in Table 10.

Table 10. WRITE_PROTECT Command Byte

COMMAND BYTE	MEANING
80h	Disable all writes except the WRITE_PROTECT command.
40h	Disable all writes except the WRITE_PROTECT and PAGE commands.
00h	Enable writes for all commands (default).

Note: *No fault or error is generated if the host attempts to write to a protected area.*

STORE_DEFAULT_ALL (11h)

The STORE_DEFAULT_ALL command instructs the device to transfer the device configuration information to the internal flash memory array. Not all information is stored. Only configuration data is stored, not any status or operational data. If an error occurs during the transfer, $\overline{\text{ALERT}}$ asserts if enabled and the CML bit in STATUS_BYTE and STATUS_WORD is set to 1. No bits are set in STATUS_CML. **It is NOT recommended to use the STORE_DEFAULT_ALL command while the device is operating fans.** The device is unresponsive to PMBus commands and does not monitor fans while transferring the configuration. After this command is sent, another command should not be sent for at least 250ms. This command is write-only. There is no data byte for this command.

USER NOTE: V_{DD} must be above 2.9V for the device to perform the STORE_DEFAULT_ALL command.

RESTORE_DEFAULT_ALL (12h)

The RESTORE_DEFAULT_ALL command transfers the default configuration information from the internal flash memory array to the user memory registers in the device. The RESTORE_DEFAULT_ALL command should only be executed when the device is not operating the fans. Upon a device reset, this command is automatically executed by the device without PMBus action required. After this command is sent, another command should not be sent for at least 250ms. This command is write-only. There is no data byte for this command.

CAPABILITY (19h)

The CAPABILITY command is used to determine some key capabilities of the device. The CAPABILITY command is read-only. The message content is described in Table 11.

Table 11. CAPABILITY Command Byte

BIT	DESCRIPTION	MEANING
7	Packet-Error Checking	0 = PEC not supported.
6:5	PMBus Speed	00 = Maximum supported bus speed is 100kHz.
4	ALERT	1 = Device supports an ALERT output (if ALERT is enabled in MFR_MODE). 0 = Device does not support ALERT output (ALERT is disabled in MFR_MODE).
3:0	Reserved	Always returns 0000.

VOUT_MODE (20h)

The VOUT_MODE command is used to report the data format of the device. The device uses the DIRECT format for all the voltage-related commands. The value returned is 40h, indicating DIRECT data format. This command is read-only. If a host attempts to write this command, the CML status bit is asserted. See Table 3 for the m, b, and R values for the various commands.

VOUT_SCALE_MONITOR (2Ah)

VOUT_SCALE_MONITOR is used in applications where the measured remote voltage is not equal to the voltage at the ADC input. For example, if the ADC input expects a 1.0V input for a 12V output, VOUT_SCALE_MONITOR = 1.0V/12V = 0.0833. In applications where the remote voltage is greater than the device input range, the voltage of the remote supply is sensed through a resistive voltage-divider. The resistive voltage-divider reduces or scales the remote voltage. The PMBus commands specify the actual remote voltages and not the input voltage to the ADC. To allow the device to map between the remote voltage (such as 12V) and the voltage at the ADC input, the VOUT_SCALE_MONITOR command is used. The 2 data bytes are in DIRECT format. This value is dimensionless. For example, if the required scaling factor is 0.0833, then VOUT_SCALE_MONITOR should be set to 0AABh (2731/32,767 = 0.0833).

Table 12. VOUT_SCALE_MONITOR

NOMINAL VOLTAGE LEVEL MONITORED (V)	NOMINAL ADC INPUT VOLTAGE LEVEL (V) (SEE NOTE)	RESISTIVE VOLTAGE-DIVIDER RATIO	VOUT_SCALE_MONITOR VALUE (HEX)
1.2	1.0	0.833	6AAAh
1.5	1.0	0.667	5555h
1.8	1.0	0.555	470Ah
2.5	1.0	0.4	3333h
3.3	1.0	0.303	26C8h
5	1.0	0.2	1999h
12	1.0	0.0833	0AABh

USER NOTE: On the device, the full-scale ADC voltage is 1.225V. The scaling factor where a 1.0V ADC input represents a nominal 100% voltage level is recommended.

FAN_CONFIG_1_2 (3Ah)

The FAN_CONFIG_1_2 command is used in conjunction with MFR_FAN_CONFIG to configure the fan. See the *MFR_FAN_CONFIG (F1h)* section for more details on this command. The FAN_CONFIG_1_2 command is described in Table 13.

Table 13. FAN_CONFIG_1_2 Command Byte

BIT	NAME	MEANING
7	FAN ENABLE	0 = Fan disabled (PWM forced low). 1 = Fan enabled.
6	RPM/PWM	0 = PWM duty cycle is the fan-controlling parameter. 1 = RPM is the fan-controlling parameter.
5:4	PULSE	00 = 1 Tach pulse per fan revolution. 01 = 2 Tach pulses per fan revolution. 10 = 3 Tach pulses per fan revolution. 11 = 4 Tach pulses per fan revolution.
3:0	0	These bits always return a 0.

6-Channel Intelligent Fan Controller

FAN_COMMAND_1 (3Bh)

The FAN_COMMAND_1 command is used to override the device's automatic fan-control function and force the fan to either a fixed PWM duty-cycle value or a target fan speed (in RPM). The units of FAN_COMMAND_1 are either percent duty cycle (if bit 6 of FAN_CONFIG_1_2 is zero) or RPM (if bit 6 of FAN_CONFIG_1_2 is one). Any value less than 0% duty cycle or 0 RPM causes the device to ignore this command and use the automatic fan-control function. Any value greater than or equal to 0% duty cycle or 0 RPM causes the device to ignore the automatic fan-control function and force the fan to the PWM value or RPM value provided by the FAN_COMMAND_1 command. The 2 data bytes are in DIRECT format.

Table 14. PWM Fan Mode (FAN_CONFIG_1_2 Bit 6 = 0)

FAN_COMMAND_1 VALUE	DEVICE RESPONSE
8000h to FFFFh	Ignore FAN_COMMAND_1 and use automatic fan-control function
0000h to 2710h	0 to 100% fan PWM duty cycle
2711h to 7FFFh	100% fan PWM duty cycle

Table 15. RPM Fan Mode (FAN_CONFIG_1_2 Bit 6 = 1)

FAN_COMMAND_1 VALUE	DEVICE RESPONSE
8000h to FFFFh	Ignore FAN_COMMAND_1 and use automatic fan-control function
0000h to 7FFFh	0 to 32,767 RPM

VOUT_OV_FAULT_LIMIT (40h)

The VOUT_OV_FAULT_LIMIT command sets the value of the output voltage that causes an output overvoltage fault. The monitored voltage must drop by at least 2% below the limit before the fault is allowed to clear. The 2 data bytes are in DIRECT format. In response to the VOUT_OV_FAULT_LIMIT being exceeded, the device does the following:

1) Sets the VOUT_OV bit in STATUS_BYTE.
2) Sets the VOUT_OV and VOUT bits in STATUS_WORD.
3) Sets the VOUT_OV_FAULT bit in STATUS_VOUT.
4) Responds as specified in the MFR_FAULT_RESPONSE.
5) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

VOUT_OV_WARN_LIMIT (42h)

The VOUT_OV_WARN_LIMIT command sets the value of the output voltage that causes an output-voltage high warning. This value is typically less than the output overvoltage threshold in VOUT_OV_FAULT_LIMIT. The monitored voltage must drop by at least 2% below the limit before the warning is allowed to clear. The 2 data bytes are in DIRECT format. In response to the VOUT_OV_WARN_LIMIT being exceeded, the device does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and VOUT bits in STATUS_WORD.
3) Sets the VOUT_OV_WARN bit in STATUS_VOUT.
4) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

VOUT_UV_WARN_LIMIT (43h)

The VOUT_UV_WARN_LIMIT command sets the value of the output voltage that causes an output-voltage low warning. This value is typically greater than the output undervoltage fault threshold in VOUT_UV_FAULT_LIMIT. The monitored voltage must increase by at least 2% above the limit before the warning is allowed to clear. The 2 data bytes are in DIRECT format. In response to violation of the VOUT_UV_WARN_LIMIT, the device does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and VOUT bits in STATUS_WORD.
3) Sets the VOUT_UV_WARN bit in STATUS_VOUT.
4) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

VOUT_UV_FAULT_LIMIT (44h)

The VOUT_UV_FAULT_LIMIT command sets the value of the output voltage that causes an output undervoltage fault. The monitored voltage must increase by at least 2% above the limit before the fault is allowed to clear. The 2 data bytes are in DIRECT format. In response to violation of the VOUT_UV_FAULT_LIMIT, the device does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and VOUT bits in STATUS_WORD.
3) Sets the VOUT_UV_FAULT bit in STATUS_VOUT.
4) Responds as specified in MFR_FAULT_RESPONSE.
5) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

OT_FAULT_LIMIT (4Fh)

The OT_FAULT_LIMIT command sets the temperature, in degrees Celsius, of the selected temperature sensor at which an overtemperature fault is detected. The monitored temperature must drop by at least 4°C below the limit before the fault is allowed to clear. The 2 data bytes are in DIRECT format. In response to the OT_FAULT_LIMIT being exceeded, the device does the following:

1) Sets the TEMPERATURE bit in STATUS_BYTE.
2) Sets the TEMPERATURE and MFR bits in STATUS_WORD.
3) Sets the OT_FAULT bit in STATUS_MFR_SPECIFIC.
4) Responds as specified in MFR_FAULT_RESPONSE.
5) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

OT_WARN_LIMIT (51h)

The OT_WARN_LIMIT command sets the temperature, in degrees Celsius, of the selected temperature sensor at which an overtemperature warning is detected. The monitored temperature must drop by at least 4°C below the limit before the warning is allowed to clear. The 2 data bytes are in DIRECT format. In response to the OT_WARN_LIMIT being exceeded, the device does the following:

1) Sets the TEMPERATURE bit in STATUS_BYTE.
2) Sets the TEMPERATURE and MFR bits in STATUS_WORD.
3) Sets the OT_WARN bit in STATUS_MFR_SPECIFIC.
4) Notifies the host through $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

STATUS_BYTE (78h)

The STATUS_BYTE command returns 1 byte of information with a summary of the most critical faults. A value of 1 indicates that a fault or warning event has occurred and a 0 indicates otherwise. Bits for unsupported features are reported as 0. The STATUS_BYTE cannot be restored by the RESTORE_DEFAULT_ALL command. The STATUS_BYTE message content is described in Table 16. This command is read-only.

Table 16. STATUS_BYTE

BIT	BIT NAME	MEANING
7:6	0	These bits always return a 0.
5	VOUT_OV	An overvoltage fault has occurred.
4:3	0	These bits always return a 0.
2	TEMPERATURE	A temperature fault or warning has occurred.
1	CML	A communication, 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 2 bytes of information with a summary of the reason for a fault. The low byte of the STATUS_WORD is the same data as the STATUS_BYTE. Table 17 describes the STATUS_WORD message content.

Table 17. STATUS_WORD

BIT	BIT NAME	MEANING
15	VOUT	An output voltage fault or warning has occurred.
14:13	0	These bits always return a 0.
12	MFR	A bit in STATUS_MFR_SPECIFIC has been set.
11	0	This bit always returns a 0.
10	FANS	A fan fault or warning has occurred.
9:6	0	These bits always return a 0.
5	VOUT_OV	An overvoltage fault has occurred.
4:3	0	These bits always return a 0.
2	TEMPERATURE	A temperature fault or warning has occurred.
1	CML	A communication, memory, or logic fault has occurred.
0	NONE OF THE ABOVE	A fault or warning not listed in bits [7:1] has occurred.

STATUS_VOUT (7Ah)

The STATUS_VOUT command returns 1 byte of information with contents as described in Table 18.

Table 18. STATUS_VOUT

BIT	BIT NAME	MEANING
7	VOUT_OV_FAULT	VOUT overvoltage fault.
6	VOUT_OV_WARN	VOUT overvoltage warning.
5	VOUT_UV_WARN	VOUT undervoltage warning.
4	VOUT_UV_FAULT	VOUT undervoltage fault.
3:0	0	These bits always return a 0.

STATUS_CML (7Eh)

The STATUS_CML command returns 1 byte of information with contents as described in Table 19.

Table 19. STATUS_CML

BIT	BIT NAME	MEANING
7	COMM_FAULT	An invalid or unsupported command has been received.
6	DATA_FAULT	An invalid or unsupported data has been received.
5:1	0	These bits always return a 0.
0	FAULT_LOG_FULL	MFR_NV_FAULT_LOG is full and needs to be cleared.

STATUS_MFR_SPECIFIC (80h)

The STATUS_MFR_SPECIFIC command returns 1 byte of information with a summary of the reason for a fault. The STATUS_MFR_SPECIFIC message content is described in Table 20.

Table 20. STATUS_MFR_SPECIFIC

BIT	BIT NAME	MEANING
7	0	This bit always returns a 0.
6	OT_WARN	Overtemperature warning.
5	OT_FAULT	Overtemperature fault.
4	WATCHDOG	A watchdog reset has occurred.
3:0	0	These bits always return a 0.

STATUS_FANS_1_2 (81h)

The STATUS_FANS_1_2 command returns 1 byte of information with the status of the fan. The STATUS_FANS_1_2 command is described in Table 21.

Table 21. STATUS_FANS_1_2

BIT	BIT NAME	MEANING
7	FAN_1_FAULT	Fan 1 fault.
6	0	This bit always returns a 0.
5	FAN_1_WARN	Fan 1 warning.
4	0	This bit always returns a 0.
3	RED	The lower nibble reports the estimated health of the fan. Only one of the four bits is set at any one time. If no bits are set, the fan health cannot be estimated. RED is the worst health state and GREEN is the best health state. The CLEAR_FAULTS command does not affect these bits.
2	ORANGE	
1	YELLOW	
0	GREEN	

Note: *The settings of the RED, ORANGE, YELLOW, and GREEN bits does not assert the $\overline{ALERT}$ signal.*

READ_VOUT (8Bh)

The READ_VOUT command returns the actual measured output voltage. READ_VOUT is measured and updated every 10ms. The 2 data bytes are in DIRECT format.

READ_TEMPERATURE_1 (8Dh)

The READ_TEMPERATURE_1 command returns the temperature returned from the temperature sensor. The value of READ_TEMPERATURE_1 does not have offset from MFR_TEMP_SENSOR_CONFIG added. READ_TEMPERATURE_1 returns 7FFFh if the sensor is faulty and 0000h if the sensor is disabled. READ_TEMPERATURE_1 is measured and updated once a second. The 2 data bytes are in DIRECT format.

READ_FAN_SPEED_1 (90h)

The READ_FAN_SPEED_1 command returns the fan speed in RPM. READ_FAN_SPEED_1 is updated once a second, even if the fan is disabled. The PULSE bits in FAN_CONFIG_1_2 must be properly configured to receive the correct fan speed. Fan speeds below 60 RPM (360 RPM for dual mode fans) are reported as 0 RPM. The 2 data bytes are in DIRECT format.

PMBUS_REVISION (98h)

The PMBUS_REVISION command returns the revision of the PMBus specification to which the device is compliant. The command has 1 data byte. Bits [7:4] 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. This command is read-only. The PMBUS_REVISION value returned is always 11h, which indicates that it is compliant with Part I Rev 1.1 and Part II Rev 1.1.

MFR_ID (99h)

The MFR_ID command returns the text (ISO/IEC 8859-1) character of the manufacturer's (Maxim) identification. The default MFR_ID value is 4Dh (M). This command is read-only.

MFR_MODEL (9Ah)

The MFR_MODEL command returns the text (ISO/IEC 8859-1) character of the device model number. The default MFR_MODEL value is 53h (S). This command is read-only.

MFR_REVISION (9Bh)

The MFR_REVISION command returns two text (ISO/IEC 8859-1) characters that contain the device revision numbers for hardware (upper byte) and firmware (lower byte). The default MFR_REVISION value is 3030h (00). This command is read-only.

MFR_LOCATION (9Ch)

The MFR_LOCATION command loads the device with text (ISO/IEC 8859-1) characters that identify the facility that manufactures the equipment. The maximum number of characters is 8. This data is written to internal flash using the STORE_DEFAULT_ALL command. The factory default text string value is 10101010.

MFR_DATE (9Dh)

The MFR_DATE command loads the device with text (ISO/IEC 8859-1) characters that identify the date of manufacture of the equipment. The maximum number of characters is 8. This data is written to internal flash using the STORE_DEFAULT_ALL command. The factory default text string value is 10101010.

MFR_SERIAL (9Eh)

The MFR_SERIAL command loads the device with text (ISO/IEC 8859-1) characters that uniquely identify the equipment. The maximum number of characters is 8. This data is written to internal flash using the STORE_DEFAULT_ALL command. The factory default text string value is 10101010.

MFR_MODE (D1h)

The MFR_MODE command is used to configure the device to support manufacturer specific commands. The MFR_MODE command is described in Table 22. After this command is sent, another command should not be sent for at least 250ms.

Table 22. MFR_MODE

BIT	BIT NAME	MEANING				
15	FORCE_NV_FAULT_LOG	Setting this bit to 1 forces the device to log data into the nonvolatile fault log. Once set, the device clears this bit when the action is completed. The host must set again for subsequent action. If an error occurs during this action, the device sets the CML bit in STATUS_BYTE and STATUS_WORD; no bits are set in STATUS_CML.				
14	CLEAR_NV_FAULT_LOG	Setting this bit to 1 forces the device to clear the nonvolatile fault log by writing FFh to all byte locations. Once set, the device clears this bit when the action is completed. The host must set again for subsequent action. If an error occurs during this action, the device sets the CML bit in STATUS_BYTE and STATUS_WORD; no bits are set in STATUS_CML.				
13	ALERT	0 = $\overline{\text{ALERT}}$ disabled (device does not respond to ARA). 1 = $\overline{\text{ALERT}}$ enabled (device responds to ARA and ARA must be used).				
12	0	This bit always returns a 0.				
11	SOFT_RESET	This bit must be set, then cleared and set again within 8ms for a soft reset to occur.				
10:8	0	These bits always return a 0.				
7:6	FAN_HEALTH_CRITERIA	These bits select the threshold for the expected RPM in the fan health monitor.				
		BIT		**GREEN**	**ORANGE**	**RED**
		0	0	< 10%	< 15%	> 15%
		0	1	< 10%	< 20%	> 20%
		1	0	< 15%	< 20%	> 20%
		1	1	< 15%	< 25%	> 25%
5	ADC5_ENABLE	0 = ADC5 voltage sense disabled. 1 = ADC5 voltage sense enabled (overridden if temp sense enabled)				
4	ADC4_ENABLE	0 = ADC4 voltage sense disabled. 1 = ADC4 voltage sense enabled (overridden if temp sense enabled)				
3	ADC3_ENABLE	0 = ADC3 voltage sense disabled. 1 = ADC3 voltage sense enabled (overridden if temp sense enabled)				
2	ADC2_ENABLE	0 = ADC2 voltage sense disabled. 1 = ADC2 voltage sense enabled (overridden if temp sense enabled)				
1	ADC1_ENABLE	0 = ADC1 voltage sense disabled. 1 = ADC1 voltage sense enabled (overridden if temp sense enabled)				
0	ADC0_ENABLE	0 = ADC0 voltage sense disabled. 1 = ADC0 voltage sense enabled (overridden if temp sense enabled)				

MFR_VOUT_PEAK (D4h)

The MFR_VOUT_PEAK command returns the maximum actual measured output voltage. To reset this value to 0, write to this command with a data value of 0. Any values written to this command are used as a comparison for future peak updates. The 2 data bytes are in DIRECT format.

MFR_TEMPERATURE_PEAK (D6h)

The MFR_TEMPERATURE_PEAK command returns the maximum measured temperature. To reset this value to its lowest value, write to this command with a data value of 8000h. Any other values written by this command are used as a comparison for future peak updates. The 2 data bytes are in DIRECT format.

MFR_VOUT_MIN (D7h)

The MFR_VOUT_MIN command returns the minimum actual measured output voltage. To reset this value, write to this command with a data value of 7FFFh. Any values written to this command are used as a comparison for future minimum updates. The 2 data bytes are in DIRECT format.

MFR_FAULT_RESPONSE (D9h)

The MFR_FAULT_RESPONSE command specifies the response to each fault condition supported by the device. In response to a fault, the device always report the fault in the appropriate status register and asserts the $\overline{\text{ALERT}}$ output (if enabled in MFR_MODE). A CML fault cannot cause any device action other than setting the status bit and asserting the $\overline{\text{ALERT}}$ output. The MFR_FAULT_RESPONSE command is described in Table 23.

Table 23. MFR_FAULT_RESPONSE

BIT	BIT NAME	MEANING
7	NV_LOG	0 = Do not log the fault into MFR_NV_FAULT_LOG. 1 = Log the fault into MFR_NV_FAULT_LOG.
6	NV_LOG_OV	This bit is only valid for pages 17 to 22. Other pages always return a 0. 0 = Bit 7 is ignored for overvoltage faults. 1 = Bit 7 also applies to overvoltage faults.
5	UV_OV_FILTER	This bit is only valid for pages 17 to 22. Other pages always return a 0. 0 = Fault or warning on first voltage sample excursion occurrence. 1 = Requires two consecutive voltage sample excursions before a fault or warning is declared and action taken.
4:3	0	These bits always return a 0.
2	FAULT_PIN_ENABLE_OV	This bit is only valid for pages 17 to 22. Other pages always return a 0. 0 = Bit 1 is ignored for overvoltage faults. 1 = Bit 1 also applies to overvoltage faults.
1	FAULT_PIN_ENABLE	0 = Never assert the $\overline{\text{FAULT}}$ pin. 1 = Assert the $\overline{\text{FAULT}}$ pin during an active fault.
0	FAULT_PIN_MONITOR	This bit is only valid for pages 0 to 5. Other pages always return a 0. 0 = Ignore the $\overline{\text{FAULT}}$ pin. 1 = Force fan to 100% PWM duty cycle when the $\overline{\text{FAULT}}$ pin is asserted.

Note 1: For fault responses, a fault is defined as either a fan fault, an overtemperature fault, or an undervoltage fault. Bits 2 and 6 allow the addition of an overvoltage fault to the fault definition.

Note 2: If FAULT_PIN_MONITOR = 1, the device reacts to both internal and external assertions of the $\overline{\text{FAULT}}$ pin.

MFR_NV_FAULT_LOG (DCh)

Each time the MFR_NV_FAULT_LOG command is executed, the device returns a block of 255 bytes containing one of the 15 nonvolatile fault logs. The MFR_NV_FAULT_LOG command must be executed 15 times to dump the complete nonvolatile fault log. If the returned fault log is all FFs, this indicates that this fault log has not been written by the device. As the device is operating, it is reading the latest operating conditions for fan speed, voltage, and temperature, and it is updating the status registers. All this information is stored in on-board RAM. When a fault is detected (if so enabled in MFR_FAULT_RESPONSE), the device automatically logs this information to one of the 15 nonvolatile fault logs. After 15 faults have been written, bit 0 of STATUS_CML is set and the host must clear the fault log by setting the CLEAR_NV_FAULT_LOG bit in MFR_MODE before any additional faults are logged. All the latest status information is logged as well as eight readings of voltage from the last 800ms in 100ms increments. If a fan is not enabled or if voltage or a temperature sensor is disabled, the associated fault log position returns 0000h.

There is a FAULT_LOG_COUNT (16-bit counter) at the beginning of each fault log that indicates which fault log is the latest. This counter rolls over should more than 65,535 faults be logged. This counter is not cleared when the CLEAR_NV_FAULT_LOG bit in MFR_MODE is toggled. The 255 bytes returned by the MFR_NV_FAULT_LOG command are described in Table 24.

If an error occurs while the device is attempting to write or clear the NV_FAULT_LOG, the device sets the CML bit in STATUS_BYTE and STATUS_WORD; no bits are set in STATUS_CML. $\overline{\text{ALERT}}$ is asserted (if enabled in MFR_MODE). See Figure 5.

USER NOTE: V_{DD} must be above 2.9V for the device to clear or log data into MFR_NV_FAULT_LOG.

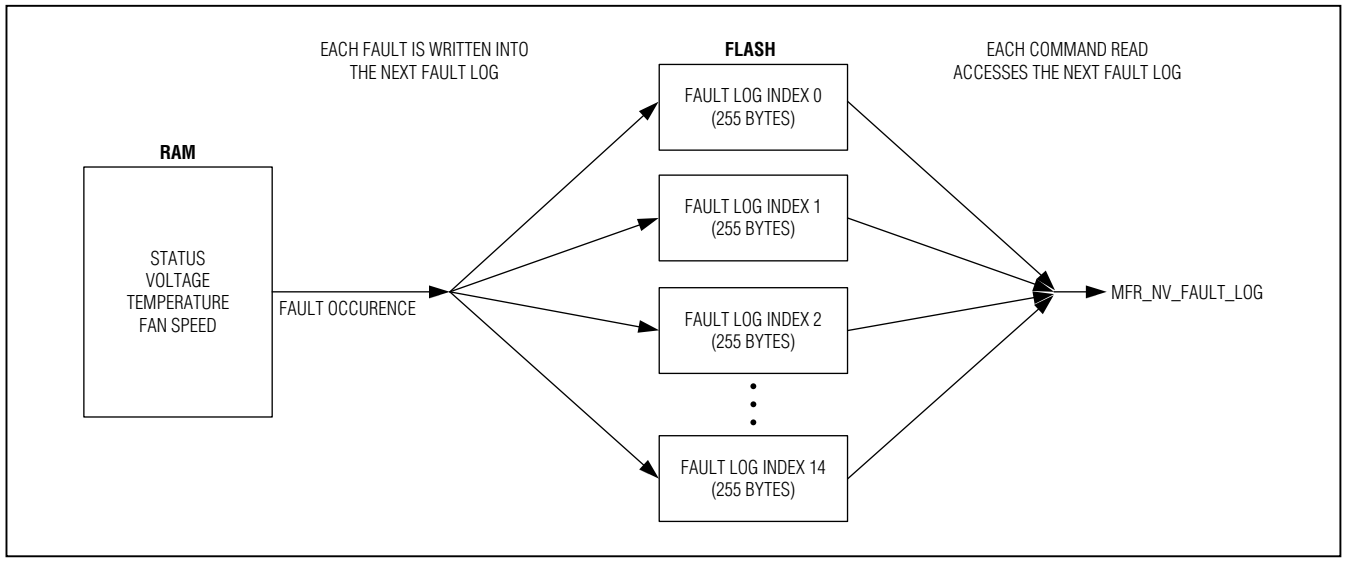


Figure 5. MFR_NV_FAULT_LOG

Table 24. MFR_NV_FAULT_LOG

BYTE	PARAMETER	BYTE	PARAMETER
0	00h/FAULT_LOG_INDEX	128	READ_VOUT Index = 3, Page 19
2	FAULT_LOG_COUNT	130	READ_VOUT Index = 3, Page 20
4	MFR_TIME_COUNT (LSW)	132	READ_VOUT Index = 3, Page 21
6	MFR_TIME_COUNT (MSW)	134	READ_VOUT Index = 3, Page 22
8	STATUS_BYTE/STATUS_CML	136	READ_VOUT Index = 4, Page 17
10	STATUS_WORD	138	READ_VOUT Index = 4, Page 18
12	STATUS_VOUT Pages 17/18	140	READ_VOUT Index = 4, Page 19
14	STATUS_VOUT Pages 19/20	142	READ_VOUT Index = 4, Page 20
16	STATUS_VOUT Pages 21/22	144	READ_VOUT Index = 4, Page 21
18	STATUS_MFR_SPECIFIC Pages 6/7	146	READ_VOUT Index = 4, Page 22
20	STATUS_MFR_SPECIFIC Pages 8/9	148	READ_VOUT Index = 5, Page 17
22	STATUS_MFR_SPECIFIC Pages 10/11	150	READ_VOUT Index = 5, Page 18
24	STATUS_MFR_SPECIFIC Pages 12/13	152	READ_VOUT Index = 5, Page 19
26	STATUS_MFR_SPECIFIC Pages 14/15	154	READ_VOUT Index = 5, Page 20
28	STATUS_MFR_SPECIFIC Pages 16/00h	156	READ_VOUT Index = 5, Page 21
30	STATUS_FANS_1_2 Pages 0/1	158	READ_VOUT Index = 5, Page 22
32	STATUS_FANS_1_2 Pages 2/3	160	READ_VOUT Index = 6, Page 17
34	STATUS_FANS_1_2 Pages 4/5	162	READ_VOUT Index = 6, Page 18
36	MFR_VOUT_PEAK Page 17	164	READ_VOUT Index = 6, Page 19
38	MFR_VOUT_PEAK Page 18	166	READ_VOUT Index = 6, Page 20
40	MFR_VOUT_PEAK Page 19	168	READ_VOUT Index = 6, Page 21
42	MFR_VOUT_PEAK Page 20	170	READ_VOUT Index = 6, Page 22
44	MFR_VOUT_PEAK Page 21	172	READ_VOUT Index = 7, Page 17
46	MFR_VOUT_PEAK Page 22	174	READ_VOUT Index = 7, Page 18
48	MFR_TEMPERATURE_PEAK Page 6	176	READ_VOUT Index = 7, Page 19
50	MFR_TEMPERATURE_PEAK Page 7	178	READ_VOUT Index = 7, Page 20
52	MFR_TEMPERATURE_PEAK Page 8	180	READ_VOUT Index = 7, Page 21
54	MFR_TEMPERATURE_PEAK Page 9	182	READ_VOUT Index = 7, Page 22
56	MFR_TEMPERATURE_PEAK Page 10	184	READ_FAN_SPEED_1 Page 0
58	MFR_TEMPERATURE_PEAK Page 11	186	READ_FAN_SPEED_1 Page 1
60	MFR_TEMPERATURE_PEAK Page 12	188	READ_FAN_SPEED_1 Page 2
62	MFR_TEMPERATURE_PEAK Page 13	190	READ_FAN_SPEED_1 Page 3
64	MFR_TEMPERATURE_PEAK Page 14	192	READ_FAN_SPEED_1 Page 4
66	MFR_TEMPERATURE_PEAK Page 15	194	READ_FAN_SPEED_1 Page 5
68	MFR_TEMPERATURE_PEAK Page 16	196	MFR_READ_FAN_PWM Page 0
70	MFR_VOUT_MIN Page 17	198	MFR_READ_FAN_PWM Page 1
72	MFR_VOUT_MIN Page 18	200	MFR_READ_FAN_PWM Page 2
74	MFR_VOUT_MIN Page 19	202	MFR_READ_FAN_PWM Page 3
76	MFR_VOUT_MIN Page 20	204	MFR_READ_FAN_PWM Page 4
78	MFR_VOUT_MIN Page 21	206	MFR_READ_FAN_PWM Page 5
80	MFR_VOUT_MIN Page 22	208	MFR_FAN_RUN_TIME Page 0
82	RESERVED (0000h)	210	MFR_FAN_RUN_TIME Page 1

Table 24. MFR_NV_FAULT_LOG (continued)

BYTE	PARAMETER	BYTE	PARAMETER
84	RESERVED (0000h)	212	MFR_FAN_RUN_TIME Page 2
86	VOLTAGE_INDEX/00h	214	MFR_FAN_RUN_TIME Page 3
88	READ_VOUT Index = 0, Page 17	216	MFR_FAN_RUN_TIME Page 4
90	READ_VOUT Index = 0, Page 18	218	MFR_FAN_RUN_TIME Page 5
92	READ_VOUT Index = 0, Page 19	220	MFR_FAN_PWM_AVG Page 0
94	READ_VOUT Index = 0, Page 20	222	MFR_FAN_PWM_AVG Page 1
96	READ_VOUT Index = 0, Page 21	224	MFR_FAN_PWM_AVG Page 2
98	READ_VOUT Index = 0, Page 22	226	MFR_FAN_PWM_AVG Page 3
100	READ_VOUT Index = 1, Page 17	228	MFR_FAN_PWM_AVG Page 4
102	READ_VOUT Index = 1, Page 18	230	MFR_FAN_PWM_AVG Page 5
104	READ_VOUT Index = 1, Page 19	232	READ_TEMPERATURE_1 Page 6
106	READ_VOUT Index = 1, Page 20	234	READ_TEMPERATURE_1 Page 7
108	READ_VOUT Index = 1, Page 21	236	READ_TEMPERATURE_1 Page 8
110	READ_VOUT Index = 1, Page 22	238	READ_TEMPERATURE_1 Page 9
112	READ_VOUT Index = 2, Page 17	240	READ_TEMPERATURE_1 Page 10
114	READ_VOUT Index = 2, Page 18	242	READ_TEMPERATURE_1 Page 11
116	READ_VOUT Index = 2, Page 19	244	READ_TEMPERATURE_1 Page 12
118	READ_VOUT Index = 2, Page 20	246	READ_TEMPERATURE_1 Page 13
120	READ_VOUT Index = 2, Page 21	248	READ_TEMPERATURE_1 Page 14
122	READ_VOUT Index = 2, Page 22	250	READ_TEMPERATURE_1 Page 15
124	READ_VOUT Index = 3, Page 17	252	READ_TEMPERATURE_1 Page 16
126	READ_VOUT Index = 3, Page 18	254	LOG_VALID (see note)

Note: *LOG_VALID is set to DDh if the fault log contains valid data.*

MFR_TIME_COUNT (DDh)

The MFR_TIME_COUNT command returns the number of seconds of device lifetime operation. Once every hour, the device automatically stores this counter in flash memory. The counter is a 32-bit value that rolls over. The latest stored value is automatically recalled from flash memory upon device power cycle, $\overline{RST}$ action, or a soft reset. This count can be reset to zero by writing a sequence of all zeros (00000000h), followed by all ones (FFFFFFFFh), followed by all zeros (00000000h) within 8ms.

MFR_TEMP_SENSOR_CONFIG (F0h)

The MFR_TEMP_SENSOR_CONFIG command is used to configure the temperature sensors. Table 25 describes the MFR_TEMP_SENSOR_CONFIG command.

Table 25. MFR_TEMP_SENSOR_CONFIG

BIT	BIT NAME	MEANING	
15	ENABLE	0 = Temperature sensor disabled. 1 = Temperature sensor enabled.	
14:10	OFFSET	The OFFSET setting is used to allow the temperature reading to be normalized among multiple temperature sensors. Values from 00h to 1Eh select the offset value. The valid range is 0°C to +30°C in 1°C steps. If OFFSET is 1Fh, the device automatically uses the value written to the OT_WARN_LIMIT command code for the LUT instead of the digitized measured temperature.	
		OFFSET VALUE	**CONFIGURATION**
		00h	Offset = 0°C
		01h	Offset = +1°C
		02h	Offset = +2°C
		1Dh	Offset = +29°C
		1Eh	Offset = +30°C
		1Fh	Test Mode
9:6	0	These bits always return a 0.	
5	FAN5	0 = Temperature sensor is **not used** to control fan 5 speed. 1 = Temperature sensor is **used** to control fan 5 speed.	
4	FAN4	0 = Temperature sensor is **not used** to control fan 4 speed. 1 = Temperature sensor is **used** to control fan 4 speed.	
3	FAN3	0 = Temperature sensor is **not used** to control fan 3 speed. 1 = Temperature sensor is **used** to control fan 3 speed.	
2	FAN2	0 = Temperature sensor is **not used** to control fan 2 speed. 1 = Temperature sensor is **used** to control fan 2 speed.	
1	FAN1	0 = Temperature sensor is **not used** to control fan 1 speed. 1 = Temperature sensor is **used** to control fan 1 speed.	
0	FAN0	0 = Temperature sensor is **not used** to control fan 0 speed. 1 = Temperature sensor is **used** to control fan 0 speed.	

MFR_FAN_CONFIG (F1h)

The MFR_FAN_CONFIG command is used with FAN_CONFIG_1_2 to configure the fan. See the FAN_CONFIG_1_2 description for more details on this command. Table 26 describes the MFR_FAN_CONFIG command.

Table 26. MFR_FAN_CONFIG

<table>
<tr><th>BIT</th><th>BIT NAME</th><th colspan="4">MEANING</th></tr>
<tr><td rowspan="10">15:13</td><td rowspan="10">FREQ[2:0]</td><td colspan="4">The FREQ bits set the PWM frequency. Note: The device does not support pulse stretching.</td></tr>
<tr><td>FREQ2</td><td>FREQ1</td><td>FREQ0</td><td>PWM FREQUENCY</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>30Hz</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>50Hz</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>100Hz</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>150Hz</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>25kHz</td></tr>
<tr><td>12</td><td>DUAL_TACH</td><td colspan="4">0 = Dual tach functionality is disabled.
1 = Dual tach functionality is enabled.
Note: In dual tachometer applications, it is recommended that the slower tachometer signal be presented to the TACH pin when TACHSEL = 0.</td></tr>
<tr><td rowspan="6">11:10</td><td rowspan="6">HYS[1:0]</td><td colspan="4">The HYS bits determine the amount of hysteresis the device uses to determine how far the temperature must fall below the temperature level threshold programmed in the LUT before switching to the lower PWM/RPM value. The hysteresis should be set lower than the minimum difference between two adjacent temperature steps. These bits are ignored if automatic fan control is disabled.</td></tr>
<tr><td colspan="2">HSY1</td><td>HSY0</td><td>THERMAL HYSTERESIS (°C)</td></tr>
<tr><td colspan="2">0</td><td>0</td><td>2</td></tr>
<tr><td colspan="2">0</td><td>1</td><td>4</td></tr>
<tr><td colspan="2">1</td><td>0</td><td>6</td></tr>
<tr><td colspan="2">1</td><td>1</td><td>8</td></tr>
<tr><td>9</td><td>TSFO</td><td colspan="4">0 = Ramp to 100% PWM duty cycle if temp sensor faults (automatic fan mode) or if no FAN_COMMAND_1 update occurs (manual fan mode) in any 10s period.
1 = Temp sensor fault or update rate to FAN_COMMAND_1 is ignored. Operate at the last updated PWM/RPM value.
Note 1: A temp sensor fault is a faulty temperature sensor reading, not an overtemperature fault.
Note 2: In automatic fan mode, if the TSFO bit is set to 1, the device ignores a sensor fault and uses the remaining assigned temperature sensors (if any) to control the fan PWM duty cycle; or, if the fan has no available temperature sensors to use, it maintains the last updated PWM/RPM fan value before the fault occurred.</td></tr>
<tr><td>8</td><td>TACHO</td><td colspan="4">0 = Ramp fan to 100% PWM duty cycle if fan fault is detected.
1 = Do not ramp fan to 100% PWM duty cycle if fan fault is detected.
Note: If the fan fault is removed after ramping the PWM to 100% duty cycle, normal fan operation is resumed.</td></tr>
</table>

Table 26. MFR_FAN_CONFIG (continued)

<table>
<tr><th>BIT</th><th>BIT NAME</th><th colspan="6">MEANING</th></tr>
<tr><td rowspan="11">7:5</td><td rowspan="11">RAMP[2:0]</td><td colspan="6">The RAMP bits select how fast the device ramps the PWM from one duty cycle to another (either up or down). In PWM mode, the following table always applies. In RPM mode, the fan speed is read either every 200ms or 1000ms and when the reported fan speed is within 20% of the target speed, the maximum allowed PWM duty cycle change is set to 1%. In RPM mode, the PWM duty cycle is not changed as long as the fan is within ±5% of the target speed.</td></tr>
<tr><td>RAMP2</td><td>RAMP1</td><td>RAMP0</td><td>PWM DUTY CYCLE UPDATE RATE (ms)</td><td>MAX PWM DUTY CYCLE CHANGE ALLOWED (%)</td><td>TIME TO RAMP FROM 40% TO 100% PWM DUTY CYCLE (seconds)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1000</td><td>1</td><td>60</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1000</td><td>2</td><td>30</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1000</td><td>3</td><td>20</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>200</td><td>1</td><td>12</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>200</td><td>2</td><td>6</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>200</td><td>3</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>200</td><td>4</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>200</td><td>5</td><td>2.4</td></tr>
<tr><td>4</td><td>HEALTH</td><td colspan="6">The HEALTH bit controls the automatic checking of the fan health. The fan-health diagnostic can be enabled by setting this bit to 1.
0 = Health meter function is disabled.
1 = Health meter function is enabled.</td></tr>
<tr><td>3</td><td>ROTOR_HI_LO</td><td colspan="6">Determines if a locked rotor indication is active low or active high. This bit is ignored if ROTOR = 0.
0 = ROTOR is active low (TACH input is low if the rotor stops).
1 = ROTOR is active high (TACH input is high if the rotor stops).</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">ROTOR</td><td colspan="6">The ROTOR bit selects if the fan does not have a tachometer but rather a stalled (or locked) rotor output.</td></tr>
<tr><td colspan="2">ROTOR</td><td colspan="2">FAN OUTPUT</td><td colspan="2">MAX31785 CONFIGURATION</td></tr>
<tr><td colspan="2">0</td><td colspan="2">Tachometer</td><td colspan="2">TACH input expects fan RPM</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Stalled/locked rotor detect</td><td colspan="2">TACH input expects locked rotor signal. The polarity is selected with the ROTOR_HI_LO bit (also set MFR_FAN_FAULT_LIMIT = 0001h).</td></tr>
<tr><td rowspan="6">1:0</td><td rowspan="6">SPIN[1:0]</td><td colspan="6">The SPIN bits determine how the device spins up (or starts) the fan from a dead stop. To overcome the initial mechanical fan inertia, the device can be programmed to drive the fan at 100% duty cycle until a programmable number of fan revolutions (cumulative count) is detected or a locked rotor signal is negated. The device allows a 2s startup period during which the fan speed monitors are disabled. If after 2s the fan does not respond, the PWM output remains at 100% duty cycle (if TACHO = 0) or goes to 0% duty cycle (if TACHO = 1).</td></tr>
<tr><td colspan="2">SPIN1</td><td colspan="2">SPIN0</td><td colspan="2">SPIN-UP RELAXATION CRITERIA</td></tr>
<tr><td colspan="2">0</td><td colspan="2">0</td><td colspan="2">Automatic spin-up disabled</td></tr>
<tr><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">Two revolutions or locked rotor negated</td></tr>
<tr><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">Four revolutions or locked rotor negated</td></tr>
<tr><td colspan="2">1</td><td colspan="2">1</td><td colspan="2">Eight revolutions or locked rotor negated</td></tr>
</table>

Note: *It is recommended that the fan be disabled prior to changing MFR_FAN_CONFIG.*

MFR_FAN_LUT (F2h)

The MFR_FAN_LUT command is used to configure the LUT that controls the fan. The fan has an LUT that maps eight programmable temperature levels to eight programmable fan PWM duty-cycle levels (bit 6 in FAN_CONFIG_1_2 is zero) or eight programmable target fan speeds (bit 6 in FAN_CONFIG_1_2 is one). The LUT allows a wide variety of profiles to be used.

USER NOTE: The programmable configuration of MFR_FAN_LUT must be monotonic.

Table 27. MFR_FAN_LUT

BYTE NUMBER	WORD NAME	MEANING
0-1	TEMP STEP 0	Temperature for step 0.
2-3	SPEED STEP 0	Fan PWM duty cycle or fan speed for step 0.
4-5	TEMP STEP 1	Temperature for step 1.
6-7	SPEED STEP 1	Fan PWM duty cycle or fan speed for step 1.
8-9	TEMP STEP 2	Temperature for step 2.
10-11	SPEED STEP 2	Fan PWM duty cycle or fan speed for step 2.
12-13	TEMP STEP 3	Temperature for step 3.
14-15	SPEED STEP 3	Fan PWM duty cycle or fan speed for step 3.
16-17	TEMP STEP 4	Temperature for step 4.
18-19	SPEED STEP 4	Fan PWM duty cycle or fan speed for step 4.
20-21	TEMP STEP 5	Temperature for step 5.
22-23	SPEED STEP 5	Fan PWM duty cycle or fan speed for step 5.
24-25	TEMP STEP 6	Temperature for step 6.
26-27	SPEED STEP 6	Fan PWM duty cycle or fan speed for step 6.
28-29	TEMP STEP 7	Temperature for step 7.
30-31	SPEED STEP 7	Fan PWM duty cycle or fan speed for step 7.

TEMPERATURE STEP: Temperature Level Setting

The TEMPERATURE STEP sets the temperature in degrees Celsius and represents a threshold level at which the device updates the fan PWM duty cycle setting. The 2 data bytes are in DIRECT format. The valid temperature range depends on the temperature sensor.

Table 28. Valid Temperature Range

TEMPERATURE SENSOR	VALID RANGE
Page 12: Internal Temp Sensor	-40°C to +85°C
Pages 13 to 16: I^2C Remote Temp Sensor	-55°C to +125°C
Page 6 to 11: Remote Thermal Diode Temp Sensor	-40°C to +120°C

6-Channel Intelligent Fan Controller

FAN SPEED STEP: Fan PWM Duty Cycle or Fan Speed Setting

If bit 6 in FAN_CONFIG_1_2 is set to zero, the FAN SPEED STEP sets the fan PWM duty cycle at each temperature step breakpoint. The valid duty cycle range is 0 to 100 (inclusive). Any values greater than 100 (decimal) result in 100% PWM duty cycle being generated, and any values less than 0 (decimal) result in 0% PWM duty cycle.

If bit 6 in FAN_CONFIG_1_2 is set to 1, the FAN SPEED STEP sets the fan target speed (in RPM) at each temperature step breakpoint. The valid fan speed range is 0 to 32,767 (inclusive).

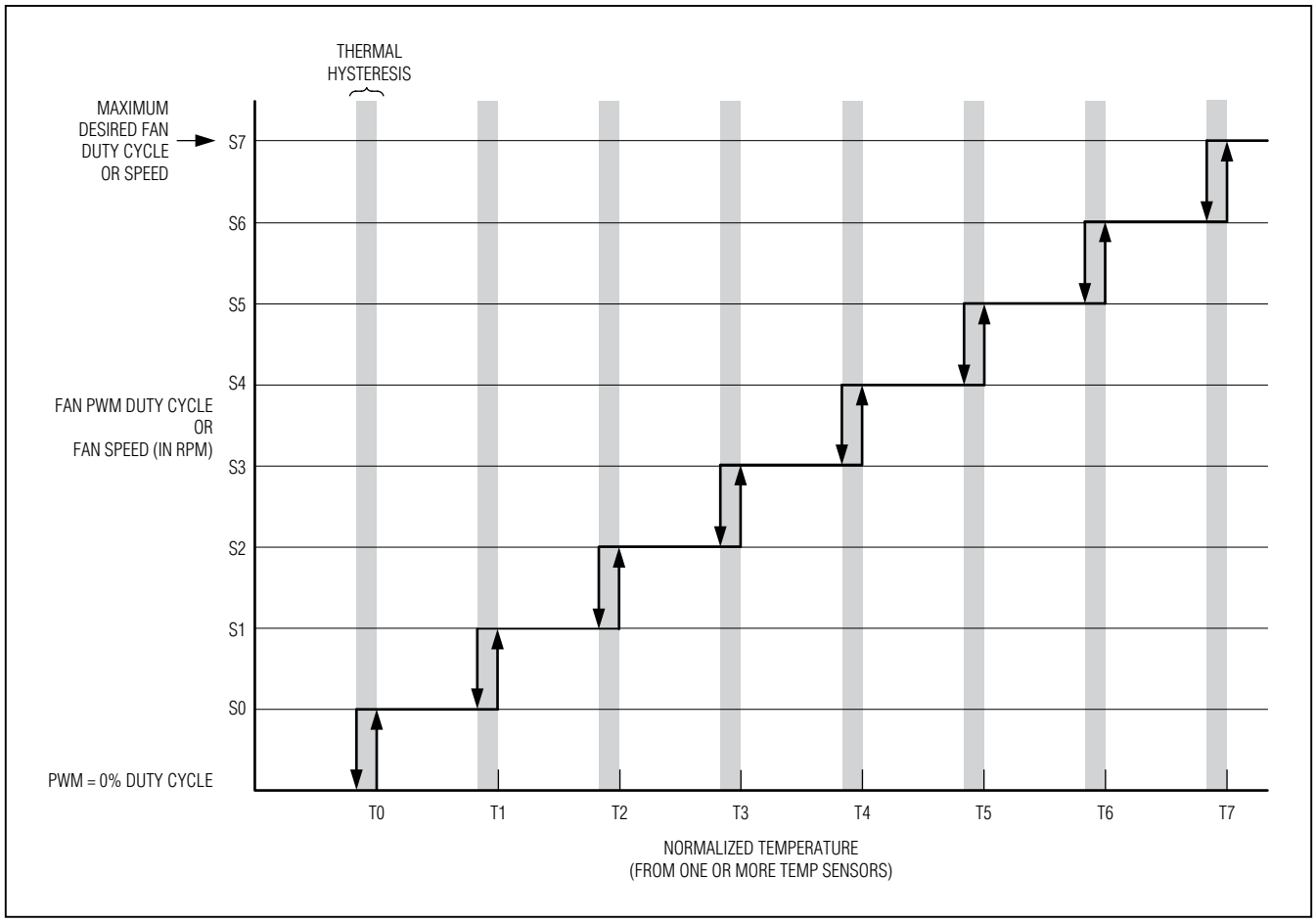


Figure 6. Fan Lookup Table (LUT) Format

MFR_READ_FAN_PWM (F3h)

The MFR_READ_FAN_PWM command returns the latest real-time value for the fan PWM in % duty cycle. MFR_READ_FAN_PWM is updated whenever the PWM duty cycle is updated. This is determined by the RATE bits in MFR_FAN_CONFIG. The 2 data bytes are in DIRECT format.

MFR_FAN_FAULT_LIMIT (F5h)

The MFR_FAN_FAULT_LIMIT command sets the value of the fan speed (in RPM) or percentage of target fan speed that causes a fan fault. Fans operating below these limits for over 10s continuous trip the fault. When the fan is operating in RPM mode, the 10s checking period starts after the device has completed a PWM ramp of the fan speed.

The slowest tachometer signal the device can measure for a fan is 60 RPM. In dual tachometer applications, the slowest RPM is 360. Tachometer signals slower than these minimums are reported as 0 RPM. Fault and warning limits should be set to RPM values greater than these minimum RPM limits.

The 2 data bytes are in DIRECT format. Set to 0000h to disable the limit checking. Set to 0001h to only alarm when the tachometer input is locked for greater than 10s (this mode should be used when the fan only has a locked rotor output). In response to violation of the MFR_FAN_FAULT_LIMIT or if a stalled/locked rotor is detected, the device does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and FANS bits in STATUS_WORD.
3) Sets the FAN_1_FAULT bit in STATUS_FANS_1_2.
4) Responds as specified in the MFR_FAULT_RESPONSE.
5) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

USER NOTE: For proper fan operation, MFR_FAN_FAULT_LIMIT must be configured.

MFR_FAN_WARN_LIMIT (F6h)

The MFR_FAN_WARN_LIMIT command sets the value of the fan speed (in RPM) or percentage of target fan speed that causes a fan speed warning. Fans operating below these limits for over 10s continuous trip the warning. When the fan is operating in RPM mode, the 10s checking period starts after the device has completed a PWM ramp of the fan speed.

Normally, the MFR_FAN_WARN_LIMIT is set higher than the MFR_FAN_FAULT_LIMIT. The 2 data bytes are in DIRECT format. Set to 0000h to disable the limit checking. Set to 0001h to generate a warning when using a locked rotor fan. In response to violation of the MFR_FAN_WARN_LIMIT, the device does the following:

1) Sets the NONE OF THE ABOVE bit in STATUS_BYTE.
2) Sets the NONE OF THE ABOVE and FANS bits in STATUS_WORD.
3) Sets the FAN_1_WARN bit in STATUS_FANS_1_2.
4) Notifies the host using $\overline{\text{ALERT}}$ assertion (if enabled in MFR_MODE).

Table 29. Monitored Fan Fault and Warning Parameters

FAN CONTROL MODE	LIMIT PARAMETER	HYSTERESIS FOR CLEARING THE FAULT/ WARNING	COMPARISON INTERVAL
Manual PWM	Fan speed (in RPM)	> (limit x 110%)	Checked once a second
Manual RPM	Percentage of programmed target fan speed	> (limit + 5%)	Checked once a second
Automatic PWM	Fan speed (in RPM)	> (limit x 110%)	Checked once a second
Automatic RPM	Percentage of LUT target fan speed	> (limit + 5%)	Checked once a second

MFR_FAN_RUN_TIME (F7h)

The MFR_FAN_RUN_TIME command returns the number of hours of fan lifetime operation. Once every hour, the device automatically stores this counter in flash memory. The counter is a 16-bit value that rolls over. The latest stored value is automatically recalled from flash memory upon device power cycle or $\overline{RST}$ action or a soft reset. This count can be reset to zero by writing a sequence of all zeros (0000h), followed by all ones (FFFFh), followed by all zeros (0000h) within 8ms. This value is not updated when the fan is turned off (0% PWM duty cycle).

MFR_FAN_PWM_AVG (F8h)

The MFR_FAN_PWM_AVG command returns the lifetime average of the fan PWM duty cycle in % duty cycle. This average is not updated when the fan is turned off (0% PWM duty cycle). When combined with the MFR_FAN_RUN_TIME information, this duty cycle helps predict the remaining lifetime of the fan. Once every hour, the device automatically stores this value in flash memory. The 2 data bytes are in DIRECT format. The latest stored value is automatically recalled from flash memory upon device power cycle or $\overline{RST}$ action or a soft reset. This count can be reset to zero by writing a sequence of all zeros (0000h), followed by all ones (FFFFh), followed by all zeros (0000h) within 8ms.

MFR_FAN_PWM2RPM (F9h)

The MFR_FAN_PWM2RPM command is used to configure the table the device uses to drive the fan-health status bits in the lower nibble of STATUS_FAN_1_2. The table maps four preassigned PWM duty cycles to the expected fan speed in RPM. Table 30 describes the MFR_FAN_PWM2RPM command.

RPM@PWM: Expected Fan Speed at a Certain PWM Duty Cycle: The expected fan speed at four PWM duty cycles is used to determine the fan health. The fan speed is set in RPM at each PWM duty-cycle step breakpoint. The valid fan-speed range is 0 to 32,767 (inclusive). The 2 data bytes are in DIRECT format.

Table 30. MFR_FAN_PWM2RPM

BYTE NUMBER	WORD NAME	MEANING
6-7	RPM@100% PWM	Expected fan speed for a PWM duty cycle of 100%.
4-5	RPM@80% PWM	Expected fan speed for a PWM duty cycle of 80%.
2-3	RPM@60% PWM	Expected fan speed for a PWM duty cycle of 60%.
0-1	RPM@40% PWM	Expected fan speed for a PWM duty cycle of 40%.

Applications Information

Power-Supply Decoupling

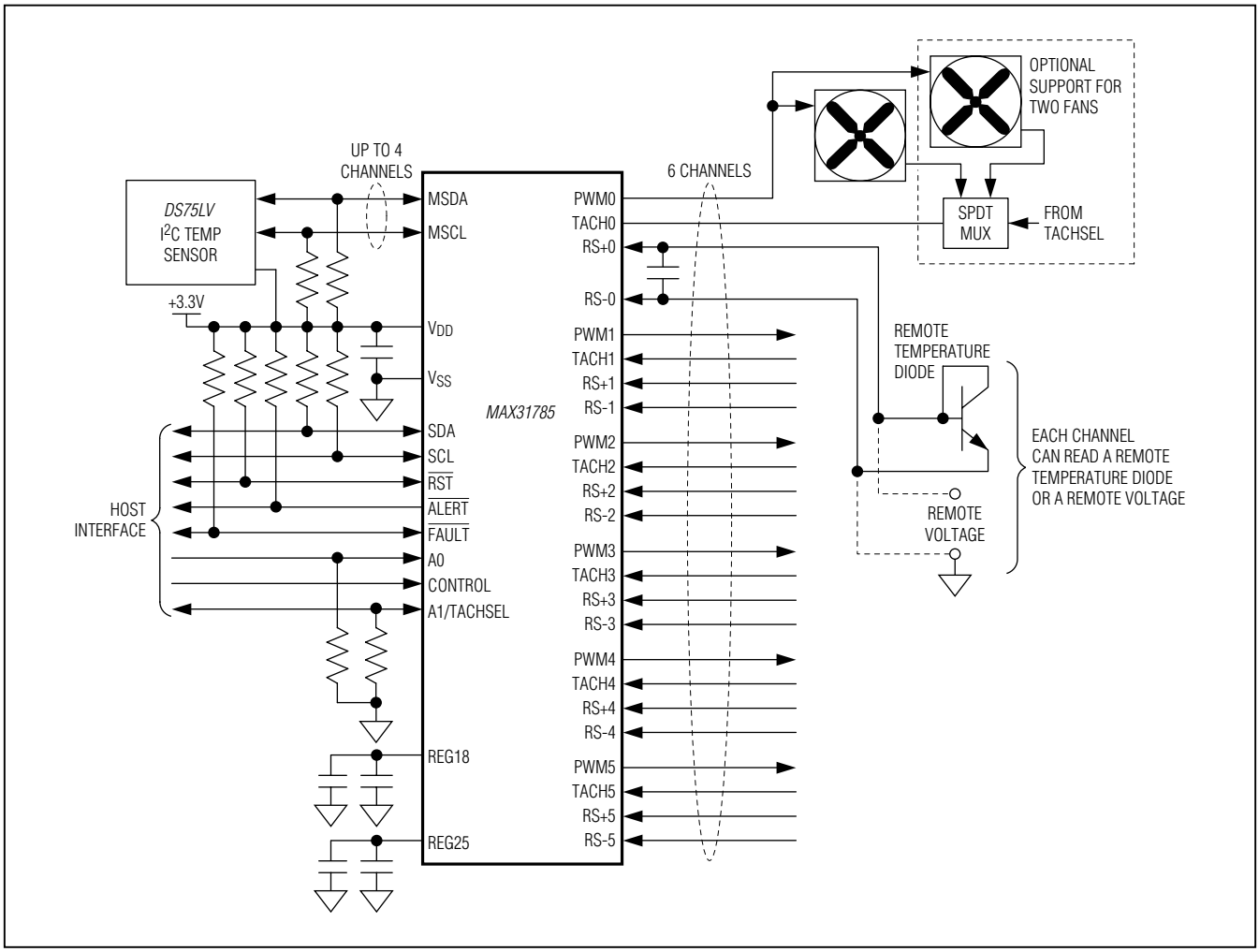
To achieve the best results when using the device, decouple the V_{DD} power supply with a 0.1µF capacitor. Use a high-quality, ceramic, surface-mount capacitor if possible. Surface-mount components minimize lead inductance, which improves performance, and ceramic capacitors tend to have adequate high-frequency response for decoupling applications.

Decouple the REG25 and REG18 regulator outputs using 1µF and 10nF capacitors (one each per output).

Open-Drain Pins

MSDA, MSCL, SCL, SDA, $\overline{FAULT}$, and $\overline{ALERT}$ are open-drain pins and require external pullup resistors connected to V_{DD} to realize high logic levels.

Typical Operating Circuit



Package Information

For the latest package outline information and land patterns (footprints), go to **www.maxim-ic.com/packages.** Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
40 TQFN-EP	T4066+2	21-0141	90-0053

6-Channel Intelligent Fan Controller

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	12/10	Initial release	—
1	11/11	Updated the lead temperature in the *Absolute Maximum Ratings* section, $V_{IL(MIN)}$ and $V_{I2C_IL(MIN)}$ values in the *Recommended Operating Conditions*, INL (MAX) value in the *DC Electrical Characteristics*, and page numbers in Table 6	6, 7, 21
2	1/12	Added Note 4 to the INL parameter and removed the DNL parameter in the *DC Electrical Characteristics*	7, 8
3	8/12	Added Note 5 to the t_{BUF} parameter conditions and changed $t_{BUF(MIN)}$ from 4.7µs to 1.4ms in the *I^2C/SMBus Interface Electrical Characteristics*; added statement about not sending another command for at least 250ms to the *STORE_DEFAULT_ALL (11h)*, *RESTORE_DEFAULT_ALL (12h)*, and *MFR_MODE (D1h)* command sections	8, 28, 35

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

© 2012 Maxim Integrated Products Maxim is a registered trademark of Maxim Integrated Products, Inc.