19-1784; Rev 2; 12/01

MAXIM

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

General Description

The MAX6650/MAX6651 fan controllers use an SMBus™/I²C™-compatible interface to regulate and monitor the speed of 5VDC/12VDC brushless fans with built-in tachometers. They automatically force the fan's tachometer frequency (fan speed) to match a preprogrammed value in the Fan-Speed Register by using an external MOSFET or bipolar transistor to linearly regulate the voltage across the fan. The MAX6650 regulates the speed of a single fan by monitoring its tachometer output. The MAX6651 also regulates the speed of a single fan, but it contains additional tachometer inputs to monitor up to four fans and control them as a single unit when they are used in parallel.

The MAX6650/MAX6651 provide general-purpose input/output (GPIO) pins that serve as digital inputs, digital outputs, or various hardware interfaces. Capable of sinking 10mA, these open-drain inputs/outputs can drive an LED. To add additional hardware control, configure GPIO1 to fully turn on the fan in case of software failure. To generate an interrupt when a fault condition is detected, configure GPIO0 to behave as an active-low alert output. Synchronize multiple devices by setting GPIO2 (MAX6651 only) as an internal clock output or an external clock input.

The MAX6650 is available in a space-saving 10-pin µMAX package, and the MAX6651 is available in a small 16-pin QSOP package.

Applications

RAID	Desktop Computers
Servers	Networking
Workstations	Telecommunications

Features

- **Closed/Open-Loop Fan-Speed Control for 5V/12V Fans**
- **2-Wire SMBus/I²C-Compatible Interface**
- **Monitors Tachometer Output**
  - **Single Tachometer (MAX6650)**
  - **Up to Four Tachometers (MAX6651)**
- **Programmable Alert Output**
- **GPIOs**
- **Hardware Full-On Override**
- **Synchronize Multiple Fans**
- **Four Selectable Slave Addresses**
- **3V to 5.5V Supply Voltage**
- **Small Packages**
  - **10-Pin µMAX (MAX6650)**
  - **16-Pin QSOP (MAX6651)**

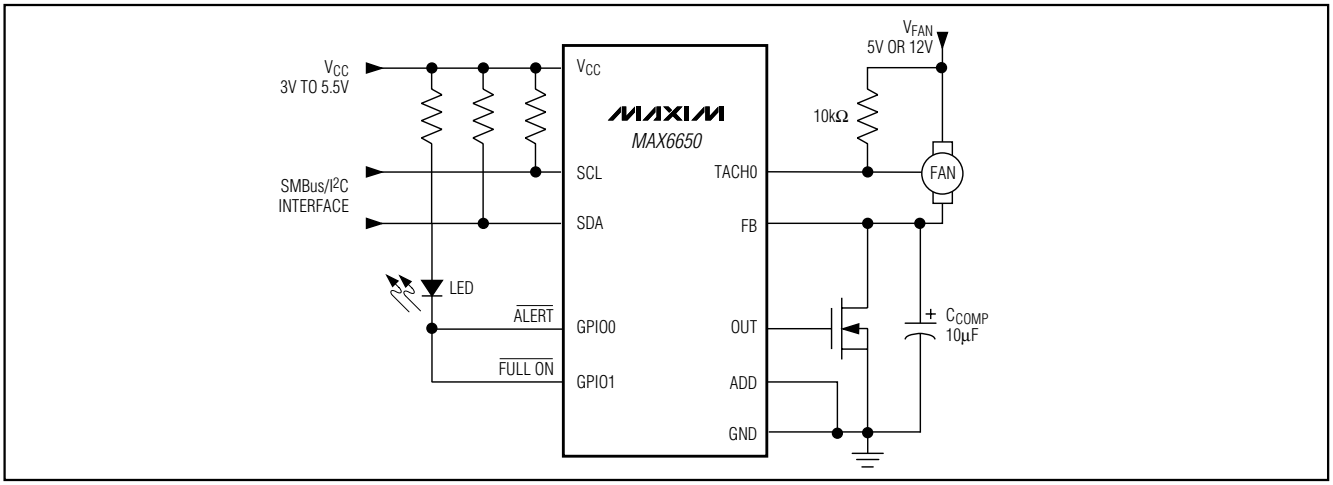
Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX6650EUB	-40°C to +85°C	10 µMAX
MAX6651EEE	-40°C to +85°C	16 QSOP

SMBus is a trademark of Intel Corp.
I²C is a trademark of Philips Corp.

Pin Configurations appear at end of data sheet.

Typical Operating Circuit



Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

ABSOLUTE MAXIMUM RATINGS

V_{CC} to GND .....-0.3V to +6V
FB, TACH_ .....-0.3V to +13.2V
All Other Pins.....-0.3V to (V_{CC} + 0.3V)
Output Voltages.....-0.3V to (V_{CC} + 0.3V)
Maximum Current
  Into V_{CC}, GND, V_{OUT} 100mA
  Into All Other Pins 50mA

Continuous Power Dissipation (T_A = +70°C)
  10-Pin µMAX (derate 5.6mW/°C above +70°C) 444mW
  16-Pin QSOP (derate 8.3mW/°C above +70°C).....667mW
Operating Temperature Range .....-40°C to +85°C
Junction Temperature .....+150°C
Storage Temperature Range .....-65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C

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.

ELECTRICAL CHARACTERISTICS

(V_{CC} = 3.0V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = +25°C and V_{CC} = 5V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY (V_{CC})						
Supply Voltage	V_{CC}		3.0		5.5	V
Supply Current	I_{CC}	Full-on mode, I_{OUT} = 0			10	mA
OUTPUT (OUT)						
Output Voltage Range	V_{OUT}	I_{OUT} = ±100µA	0.3		V_{CC} - 0.3	V
Output Sink Current	I_{SINK}	V_{OUT} = 0.5V	10			mA
Output Source Current	I_{SOURCE}	V_{OUT} = V_{CC} - 1.8V	50			mA
TACHOMETER INPUTS (TACH_)						
Tachometer Threshold	$V_{TACH\_}$	5V fan, 0 < V_{FB} < 4.5V	V_{FB} + 0.5		V_{FB} +1.5	V
		12V fan, 0 < V_{FB} < 9V	V_{FB} + 1.0		V_{FB} +3	
Tachometer Input Impedance	$R_{TACH\_}$	0 < V_{TACH} < 9V	70	100	150	kΩ
FEEDBACK (FB)						
DAC Differential Nonlinearity		Guaranteed monotonicity on FB (Note 1)			5	LSB
Useful DAC Resolution		Measured at FB (Note 1)			8	bits
Feedback Input Impedance	R_{FB}	0 < VFB < 9V	70	100	150	kΩ
GENERAL-PURPOSE INPUTS/OUTPUTS (GPIO_) (Note 2)						
Input Low Voltage	$V_{IL(GPIO\_)}$				0.8	V
Input High Voltage	$V_{IH(GPIO\_)}$	V_{CC} ≤ 3.6V	2			V
		V_{CC} > 3.6V	3			
Input Hysteresis	V_{HYS}			200		mV
Pullup Resistor	$R_{GPIO\_}$			100		kΩ
Output Sink Current	$I_{GPIO\_}$	$V_{GPIO\_}$ = 0.4V	10			mA

ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 3.0V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = +25°C and V_{CC} = 5V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
ADDRESS SELECT (ADD)						
ADD Input Low Voltage	$V_{IL(ADD)}$	Selects slave address 1001 000 (Table 1)			0.1	V
ADD Input High Voltage	$V_{IH(ADD)}$	Selects slave address 1001 011 (Table 1)	V_{CC} - 0.05V			V
ADD External Pulldown Resistor to GND	R_{ADD}	Selects slave address 0011 111 (Table 1)	9.5		10.5	kΩ
Open Resistance	R_{OPEN}	Minimum resistance to GND, selects slave address 0011 011 (Table 1)	5			kΩ
ADD Pullup Current	I_{ADD}	V_{ADD} = 0.5V	40		80	µA
SMBus/I²C INTERFACE (SDA, SCL)						
Data Output Sink Current	I_{SDA}	V_{SDA} = 0.6V	6			mA
Input Leakage Current		0 < V_{IN} < V_{CC}			±1	µA
Input Low Voltage	V_{IL}				0.8	V
Input High Voltage	V_{IH}	V_{CC} ≤ 3.6V	2			V
		V_{CC} > 3.6V	3			
Input Hysteresis	V_{HYS}			200		mV

TIMING CHARACTERISTICS

(V_{CC} = 3.0V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = +25°C and V_{CC} = 5V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
TACHOMETERS						
Glitch Rejection		Minimum pulse duration		500		µs
GPIO2 (Note 2)						
Clock Frequency	f_{CLK}			254		kHz
Clock Frequency Uncertainty	f_{CLK}	V_{CC} = 5V	-10		+10	%
SMBus/I²C INTERFACE (Figures 3, 4)						
SCL Clock Frequency	f_{SCL}		0		400	kHz
Bus Free Time Between Stop and Start Condition	t_{BUF}		1.3			µs
Hold-Time Start Condition	$t_{HD:STA}$		0.6			µs
Low Period of the SCL Clock	t_{LOW}		1.3			µs
High Period of the SCL Clock	t_{HIGH}		0.6			µs
Data Hold Time	$t_{HD:DAT}$	(Note 3)	0		900	µs
Data Setup Time	$t_{SU:DAT}$		100			ns
Rise-Time SDA/SCL Signal (Receiving)	t_R	(Note 4)	20 + 0.1C_B(pf)		300	ns
Fall-Time SDA/SCL Signal (Receiving)	t_F	(Note 4)	20 + 0.1C_B(pf)		300	ns
Fall-Time SDA Signal (Transmitting)	t_F	I_{SINK} < 6mA (Note 4)	20 + 0.1C_B(pf)		250	ns

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

TIMING CHARACTERISTICS (continued)

(V_{CC} = 3.0V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at T_A = +25°C and V_{CC} = 5V.)

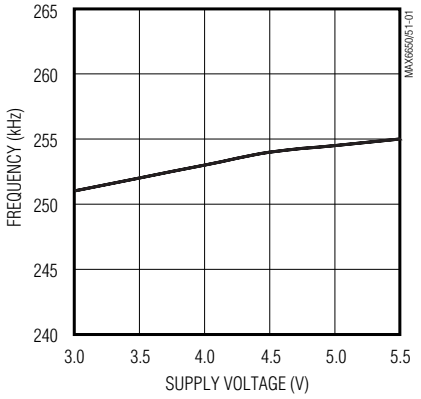
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Setup Time for Stop Condition	$t_{SU:STO}$		0.6			µs
Pulse Width of Spike Suppressed	t_{SPIKE}		0	50		ns

Note 1: For proper measurement of V_{FB}, connect OUT and FB as shown in the *Typical Operating Circuit*.
Note 2: GPIO2, GPIO3, and GPIO4 only in the MAX6651.
Note 3: Note that the transition must internally provide at least a hold time to bridge the undefined region (300ns max) of SCL's falling edge.
Note 4: C_B is the total capacitance of one bus line in pF. Tested with C_B = 400pF. Rise and fall times are measured between 0.3 x V_{CC} and 0.7 x V_{CC}.

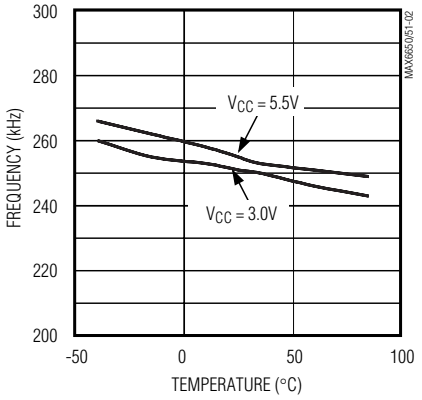
Typical Operating Characteristics

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

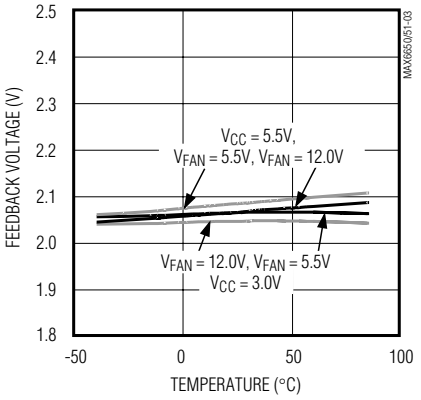
INTERNAL OSCILLATOR FREQUENCY vs. SUPPLY VOLTAGE



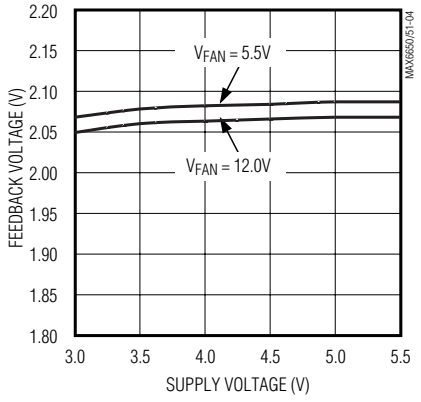
INTERNAL OSCILLATOR FREQUENCY vs. TEMPERATURE



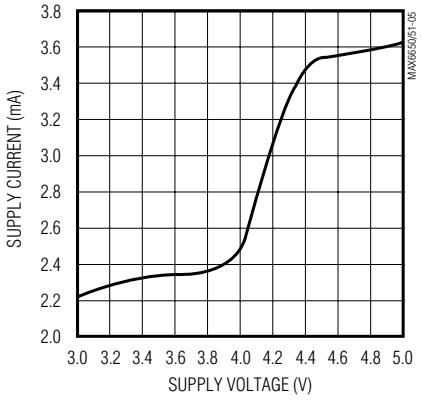
FEEDBACK VOLTAGE vs. TEMPERATURE



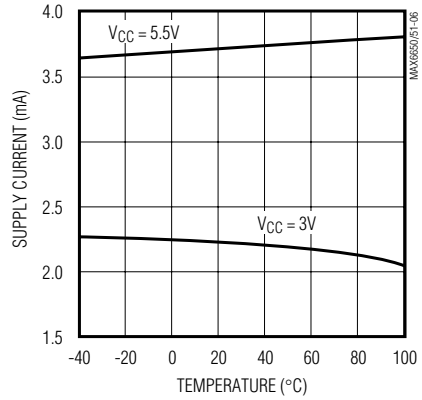
FEEDBACK VOLTAGE vs. SUPPLY VOLTAGE (DAC SET TO 35)



SUPPLY CURRENT vs. SUPPLY VOLTAGE



SUPPLY CURRENT vs. TEMPERATURE



Pin Description

PIN MAX6650	PIN MAX6651	NAME	FUNCTION
1	1	TACH0	Tachometer Input. Used to close the loop around the tachometer.
—	2, 3, 16	TACH2, TACH3, TACH1	Tachometer Inputs. Used to monitor tachometers only.
2	4	GND	Ground
3	5	SDA	2-Wire Serial-Data Input/Output (open drain)
4	6	SCL	2-Wire Serial Clock Input
5	8	ADD	Slave Address Select Input (Table 1)
—	7, 12	GPIO4, GPIO3	General-Purpose Input/Output (open drain)
6	9	GPIO1	General-Purpose Input/Output (open drain). Configurable to act either as an output or as an input ($\overline{\text{FULL ON}}$ or general purpose).
7	10	GPIO0	General-Purpose Input/Output (open drain). Configurable to act as a general input/output line or an active-low $\overline{\text{ALERT}}$ output.
—	11	GPIO2	General-Purpose Input/Output (open drain). Configurable to act as a general input/output line, an internal clock output, or an external clock input.
8	13	OUT	Output. Drives the external MOSFET or bipolar transistor.
9	14	V_{CC}	+3.0V to +5.5V Power Supply
10	15	FB	Feedback Input. Closes the loop around the external MOSFET or bipolar transistor.

Detailed Description

The MAX6650/MAX6651 use an SMBus/I²C-Compatible interface to regulate and monitor the speed of 5VDC/12VDC brush-less fans with built-in open-collector/drain tachometers. Regulating fan speed proportionally with temperature saves power, increases fan life, and reduces acoustic noise. Since fan speed is proportional to the voltage across the fan, the MAX6650/MAX6651 control the speed by regulating the voltage on the low side of the fan with an external MOSFET or bipolar transistor.

The MAX6650/MAX6651 each contain two internal control loops. The first loop controls the voltage across the fan. The internal digital-to-analog converter (DAC) sets the reference voltage for an internal amplifier (Figure 1), which then drives the gate of an external N-channel MOSFET (or the base of a bipolar transistor) to regulate the voltage on the low side of the fan. As the reference voltage provided by the DAC changes, the feedback amplifier automatically adjusts the feedback voltage, which changes the voltage across the fan.

The second control loop consists of the internal digital logic that controls the fan's speed. The MAX6650/MAX6651 control fan speed by forcing the tachometer frequency to equal a reference frequency set by the Fan-Speed Register, the prescaler, and the internal oscillator (see the *Fan-Speed Register* section). When the tachometer frequency is too high, the value of the DAC's input register is increased by the regulator. Once the DAC voltage increases, the analog control loop forces the feedback voltage to rise, which reduces the voltage across the fan. Since fan speed is proportional to the voltage across the fan, the fan slows down.

2-Wire SMBus/I²C-Compatible Digital Interface

From a software perspective, the MAX6650/MAX6651 appear as a set of byte-wide registers that contain speed control, tachometer count, alarm conditions, or configuration bits. These devices use a standard SMBus/I²C-compatible 2-wire serial interface to access the internal registers.

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

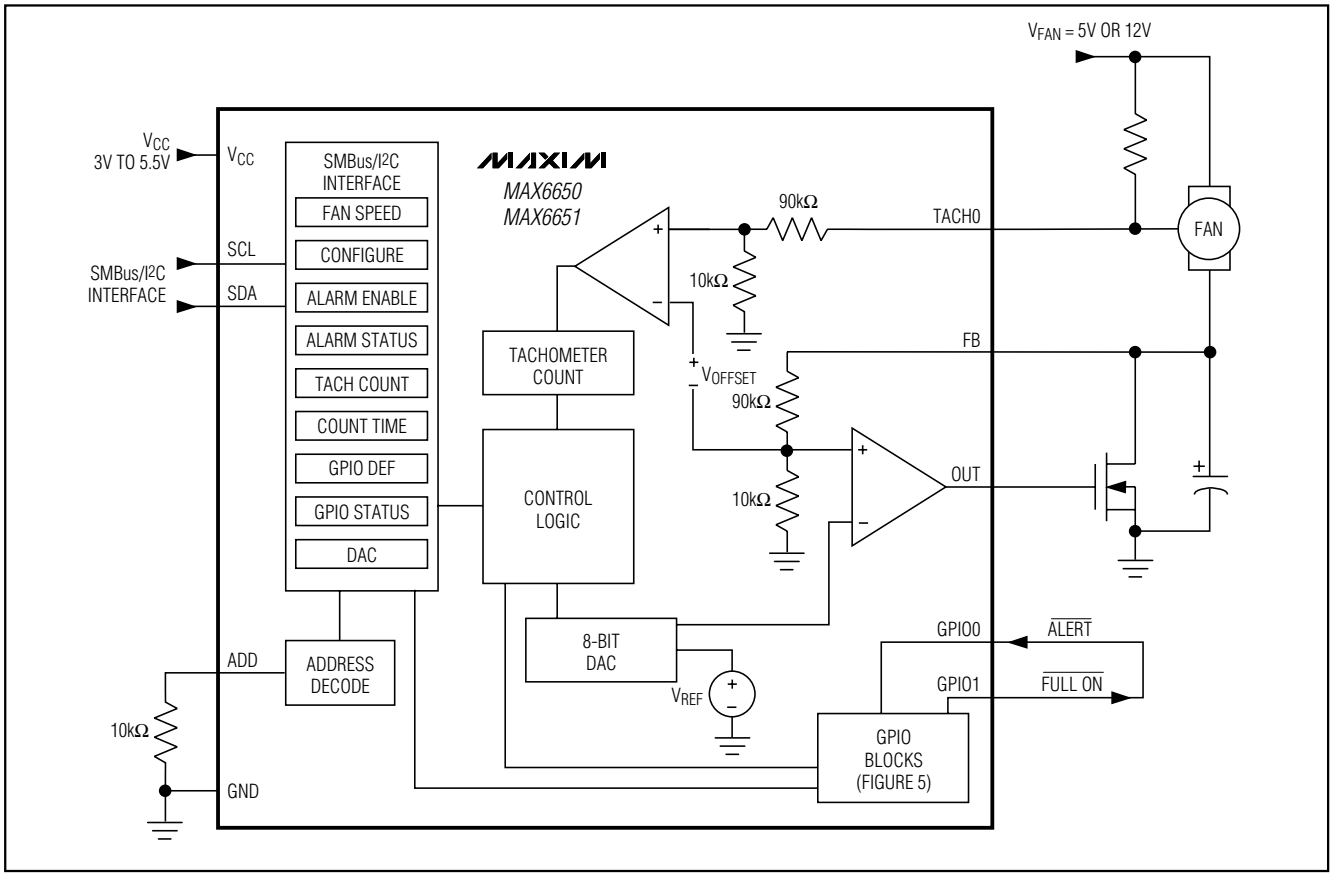


Figure 1. Block Diagram

The MAX6650/MAX6651 employ three standard SMBus protocols: write byte, read byte, and receive byte (Figure 2). The shorter protocol (receive) allows quicker transfers, provided that the correct data register was previously selected by a write or read byte instruction. Use caution with the shorter protocol in multimaster systems, since a second master could overwrite the command byte without informing the first master.

Slave Addresses

The device address can be set to one of four different values. Accomplish this by pin-strapping ADD so that more than one MAX6650/MAX6651 can reside on the same bus without address conflicts (Table 1).

Table 1. Slave Address Decoding (ADD)

ADD	ADDRESS
	BINARY
GND	1001 000
VCC	1001 011
No connection (high-Z)	0011 011
10kΩ resistor to GND	0011 111

S	ADDRESS	$\overline{WR}$	ACK	COMMAND	ACK	DATA	ACK	P
	7 bits	0		8 bits		8 bits		

Slave Address

Command byte: Selects which register you are writing to.

Data byte: Data goes into the register set by the command byte (to set thresholds, configuration masks, and sampling rate).

Figure 2a. SMBus Protocol: Write Byte Format

S	ADDRESS	$\overline{WR}$	ACK	COMMAND	ACK
	7 bits	0		8 bits	

Slave Address

Command byte: Selects which register you are reading from.

S	ADDRESS	RD	ACK	DATA	$\overline{A}$	P
	7 bits	1		8 bits		

Slave Address. Repeated due to change in data-flow direction

Data byte: Reads from the register set by the command byte.

Figure 2b. SMBus Protocol: Read Byte Format

S	ADDRESS	RD	ACK	DATA	$\overline{A}$	P
	7 bits	1		8 bits		

Slave Address

Data byte: Reads data from the register commanded by the last read-byte or write-byte transmission; also used for SMBus alert response return address.

Figure 2c. SMBus Protocol: Receive Byte Format

S = Start condition
P = Stop condition
Shaded = Slave transmission
ACK = Acknowledged = 0
$\overline{A}$ = Not acknowledged = 1
WR = Write = 0
RD = Read =1

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

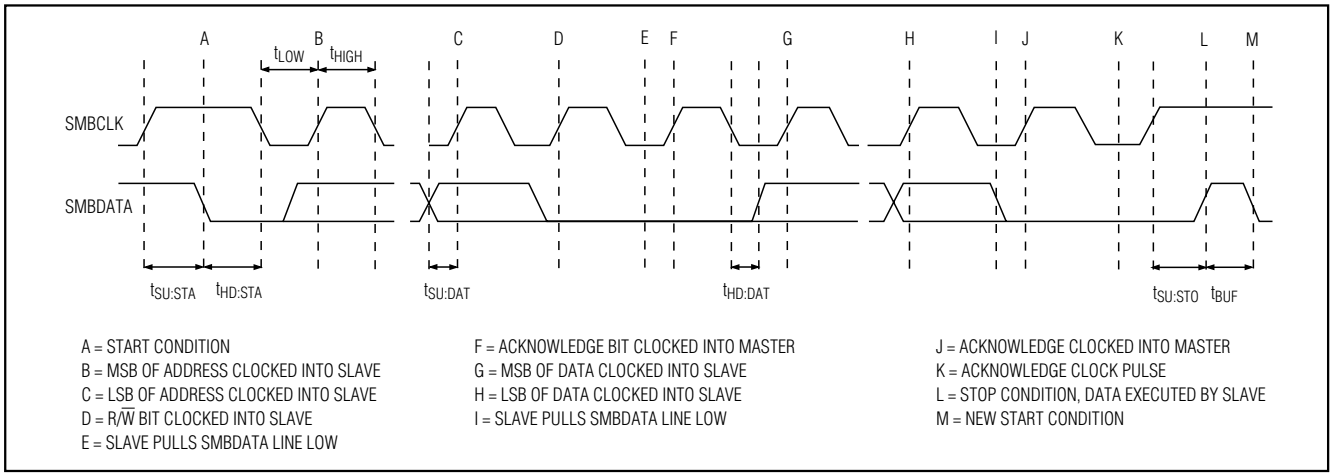


Figure 3. SMBus Write Timing Diagram

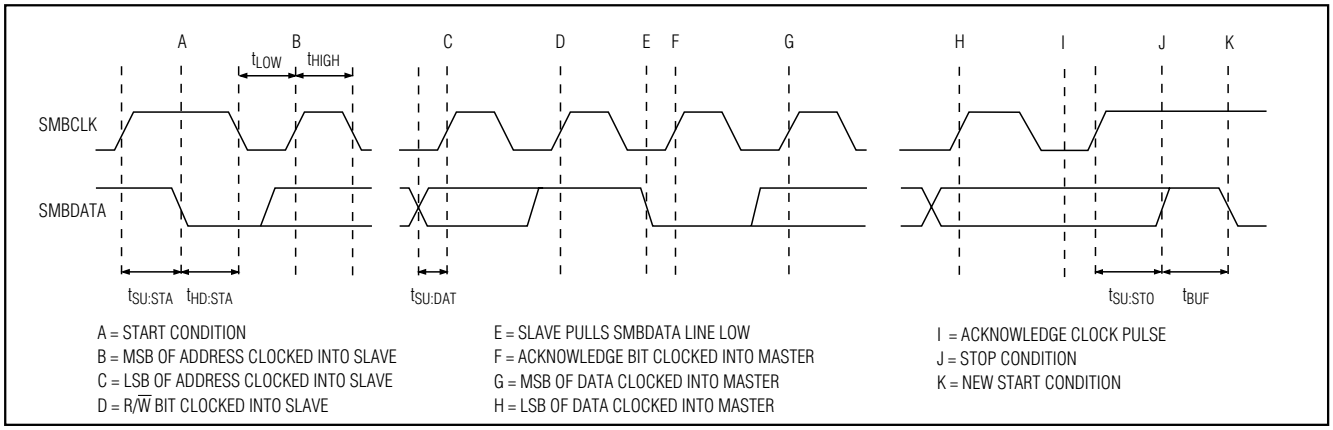


Figure 4. SMBus Read Timing Diagram

Command-Byte Functions

The 8-bit Command-Byte Register (Table 2) is the master index that points to the various other registers within MAX6650/MAX6651. The register's power-on reset (POR) state is 0000 0000, so that a receive-byte transmission (a protocol that lacks the command byte) occurring immediately after POR returns the current speed setting.

Fan-Speed Register

In closed-loop mode, the MAX6650/MAX6651 use the Fan-Speed Register to set the period of the tachometer signal that controls the fan speed. The Fan-Speed Register is ignored in all other modes of operation. The MAX6650/MAX6651 regulate the fan speed by forcing the tachometer period (t_{TACH}) equal to the scaled register value. One revolution of the fan generates two tachometer pulses, so the required Fan-Speed Register value (K_{TACH}) may be calculated as:

$$t_{TACH} = 1 / (2 \times \text{Fan Speed})$$

$$K_{TACH} = [t_{TACH} \times K_{SCALE} \times (f_{CLK} / 128)] - 1$$

where the fan speed is in rotations per second (RPS), t_{TACH} is the period of the tachometer signal, f_{CLK} is the internal oscillator frequency (254kHz ±10%), and K_{SCALE} is the prescaler value (see *Configuration-Byte Register*). Since the fan speed is inversely proportional to the tachometer period, the Fan-Speed Register value (K_{TACH}) does not linearly control the fan speed (Table 3). Select the prescaler value so the fan's full speed is achieved with a register value of approximately 64 (0100 0000) to optimize speed range and resolution. The MAX6651 may be controlled by an external oscilla-

Table 2. Command-Byte Assignments

REGISTER	COMMAND	READ	WRITE	POR (DEFAULT) STATE	FUNCTION
SPEED	0000 0000	x	x	00h	Fan speed
CONFIG	0000 0010	x	x	0Ah	Configuration
GPIO DEF	0000 0100	x	x	FFh	GPIO definition
DAC	0000 0110	x	x	00h	DAC
ALARM ENABLE	0000 1000	x	x	00h	Alarm enable
ALARM	0000 1010	x	—	00h	Alarm status
TACH0	0000 1100	x	—	00h	Tachometer 0 count
TACH1	0000 1110	x	—	00h	Tachometer 1 count
TACH2	0001 0000	x	—	00h	Tachometer 2 count
TACH3	0001 0010	x	—	00h	Tachometer 3 count
GPIO STAT	0001 0100	x	—	h1Fh	GPIO status
COUNT	0001 0110	x	x	02h	Tachometer count time

Table 3. Fan Speed

K_{TACH}	t_{TACH}			FAN SPEED (RPS)			FAN SPEED (RPM)		
	K_{SCALE} (ms)			K_{SCALE}			K_{SCALE}		
	1	4	16	1	4	16	1	4	16
0000 0000	1.0	*	*	500	*	*	30,000	*	*
0000 0001	1.0	*	*	500	*	*	30,000	*	*
0000 0010	1.5	*	*	330	*	*	20,000	*	*
—	—	—	—	—	—	—	—	—	—
0001 1110	16	3.9	*	32	128	*	1900	7700	*
0001 1111	16	4.0	1.0	31	124	500	1900	7400	30,000
0010 0000	17	4.2	1.0	30	120	480	1800	7200	29,000
—	—	—	—	—	—	—	—	—	—
0100 0000	33	8.2	2.1	15.3	61.1	240	910	3700	15,000
—	—	—	—	—	—	—	—	—	—
1111 1000	125	31	7.8	4	15.9	64	240	960	3830

**The minimum allowed tachometer period is 1ms.*

tor that overrides the internal oscillator (see *General-Purpose Input/Output*). When using an external oscillator (f_{OSC}), calculate the Fan-Speed Register value with f_{CLK} equal to f_{OSC}. Codes above F8h (1111 1000) are allowed, but will not significantly decrease the frequency.

Configuration-Byte Register

The Configuration-Byte Register (Table 4) adjusts the prescaler, changes the tachometer threshold voltage, and sets the mode of operation. The three least-significant bits configure the prescaler division used to scale the tachometer period. Select the prescaler value so the fan's full speed is achieved with a register value of approximately 64 (0100 0000) to optimize speed range and resolution (see the *Fan Speed Register* section). The fourth bit selects the fan operating voltage.

The fifth and sixth bits configure the operating mode. The MAX6650/MAX6651 have four modes of operation: full-on, full-off (shutdown), closed-loop, and open-loop. In closed-loop operation, the external microcontroller (µC) sets the desired speed by writing an 8-bit word to the Fan-Speed Register (see the *Fan-Speed Register* section). The MAX6650/MAX6651 monitor the fan's tachometer output and automatically adjust the voltage

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

Table 4. Configuration Byte Register

BIT	NAME	POR (DEFAULT) STATE	FUNCTION
7 (MSB) to 6	—	0	Always 0
5 to 4	MODE	00	Operating Mode: 00 = Software full-on (default) 01 = Software off (shutdown) 10 = Closed-loop operation 11 = Open-loop operation
3	$\overline{5}$/12V	1	Fan/Tachometer Voltage: 0 = 5V 1 = 12V (default)
2 to 0 (LSB)	SCALE	010	Prescaler Division: 000 = Divide by 1 001 = Divide by 2 010 = Divide by 4 (default) 011 = Divide by 8 100 = Divide by 16

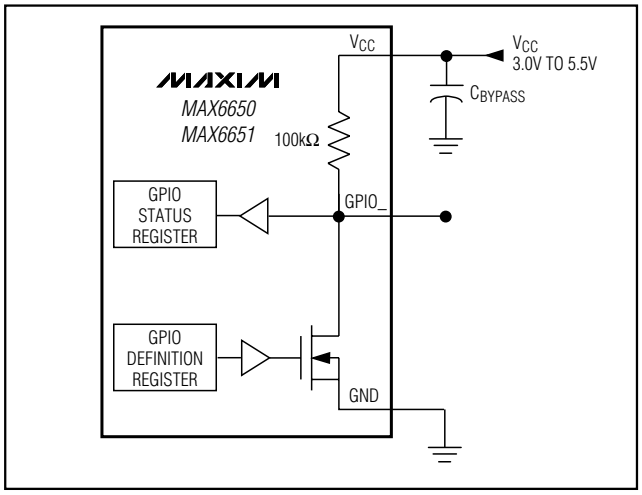


Figure 5. General-Purpose Input/Output Structure

across the fan until the desired speed is reached. Open-loop operation allows the µC to regulate fan speed directly. The µC reads the fan speed from the Tachometer-Count Register. Based on the tachometer count, the µC decides if the fan speed requires adjustment, and changes the voltage across the fan by writing an 8-bit word to the DAC Register. Full-on mode applies the maximum voltage across the fan, forcing it to spin at full speed. Configuring GPIO1 (see the *General-Purpose Input/Output* section) as an active-low input provides additional hardware control that fully turns on the fan and overrides all software commands.

General-Purpose Input/Output

The GPIO pins connect to the drain of the internal N-channel MOSFET and pullup resistor (Figure 5). When the N-channel MOSFET is off (Table 5), the pullup resistor provides a logic-level high output. However, with the MOSFET off, the GPIO may serve as an input pin and its state is read from the GPIO Status Register (Table 6). The MAX6650/MAX6651 power up with the MOSFET off, so input signals may be safely connected to the GPIO pins. When using the GPIO pin as a general-purpose output, change the output by writing to the GPIO Definition Register.

GPIO0 may be configured as an $\overline{\text{ALERT}}$ output that will go low whenever a fault-condition is detected (see the *Alarm-Enable and Status Registers* section). GPIO1 may be configured as a $\overline{\text{FULL ON}}$ input to allow hardware control to fully turn on the fan in case of software or µC failure. GPIO2 (MAX6651 only) may be configured as an internal clock output or as an external clock input to allow synchronization of multiple devices.

Alarm-Enable and Status Registers

The alarms are enabled only when the appropriate bits of the Alarm-Enable Register are set (Table 7). The maximum and minimum output level alarms function only when the device is configured to operate in the closed-loop mode (see the *Configuration-Byte Register* section).

The Alarm Status Register allows the system to determine which alarm caused the alert output (Table 8). The set-alarm and alert outputs clear after reading the

Table 5. GPIO Definition Register

BIT	POR (DEFAULT) STATE	PIN	STATE	FUNCTION
7 (MSB)	1	GPIO4 (MAX6651 only)	0	GPIO4 outputs a logic-level low.
			1	GPIO4 outputs a logic-level high or serves as an input.
6	1	GPIO3 (MAX6651 only)	0	GPIO3 outputs a logic-level low.
			1	GPIO3 outputs a logic-level high or serves as an input.
5 to 4	11	GPIO2 (MAX6651 only)	00	GPIO2 serves as an external clock input.
			01	GPIO2 serves as an internal clock output.
			10	GPIO2 outputs a logic-level low.
			11	GPIO2 outputs a logic-level high or serves as an input.
3 to 2	11	GPIO1	00	GPIO1 outputs a logic-level high or serves as an input.
			01	GPIO1 serves as a $\overline{\text{FULL ON}}$ input.
			10	GPIO1 outputs a logic-level low.
			11	GPIO1 outputs a logic-level high or serves as an input.
—	11	GPIO0	00	GPIO0 outputs a logic-level high or serves as an input.
			01	GPIO0 serves as an $\overline{\text{ALERT}}$ output.
			10	GPIO0 outputs a logic-level low.
			11	GPIO0 outputs a logic-level high or serves as an input.

Table 6. GPIO Status Register

BIT	NAME	POR (DEFAULT) STATE
7 (MSB) to 5	Always 0	0
4	GPIO4 (MAX6651 only)	1
3	GPIO3 (MAX6651 only)	1
2	GPIO2 (MAX6651 only)	1
1	GPIO1	1
0 (LSB)	GPIO0	1

Alarm Status Register if the condition that caused the alarm is removed.

Tachometer

The Tachometer Count Registers record the number of pulses on the corresponding tachometer input during the period defined by the Tachometer Count-Time Register.

The MAX6651 contains three additional tachometer inputs, which may be used to monitor additional fans. For accurate control of multiple fans, use identical fans.

The Tachometer Count-Time Register sets the integration time over which the MAX6650/MAX6651 count tachometer pulses. The devices can count up to 255 (FFh) pulses during the selected count time. If more than 255 pulses occur, the IC sets the overflow alarm and the Tachometer Count Register reports the maximum value of 255. Set the time register so the count register will not overflow under worst-case conditions (maximum fan speed) while maximizing resolution. Calculate the maximum measurable fan speed and minimum resolution with the following equations:

$$\text{Max Fan Speed (in RPS)} = 255 / (2 \times t_{COUNT})$$

$$\text{Min Resolution (in RPS)} = 1 / (2 \times t_{COUNT})$$

where t_{COUNT} is the tachometer count time; 1kHz is the maximum allowable tachometer input frequency for the MAX6650/MAX6651.

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

Table 7. Alarm-Enable Register Bit Masks

BIT	NAME	POR (DEFAULT) STATE	FUNCTION
7 (MSB) to 5	—	0	Always 0
4	GPIO2 (MAX6651 only)	0	GPIO2 Alarm Enable/Disable (MAX6651 only)
3	GPIO1	0	GPIO1 Alarm Enable/Disable
2	TACH	0	Tachometer Overflow Alarm Enable/Disable
1	MIN	0	Minimum Output Level Alarm Enable/Disable
0 (LSB)	MAX	0	Maximum Output Level Alarm Enable/Disable

1 = Enabled

Table 8. Alarm Status Register Bit Assignments

BIT	NAME	POR (DEFAULT) STATE	FUNCTION
7 (MSB) to 5	—	0	Always 0
4	GPIO2 (MAX6651 only)	0	GPIO2 Alarm. Set when GPIO2 is low (MAX6651 only).
3	GPIO1	0	GPIO1 Alarm. Set when GPIO1 is low.
2	TACH	0	Tachometer Overflow Alarm
1	MIN	0	Minimum Output Level Alarm
0 (LSB)	MAX	0	Maximum Output Level Alarm

1 = Alarm condition

Table 9. Tachometer Count-Time Register

(Assumes two pulses per revolution)

REGISTER VALUE (K_{COUNT})	COUNT TIME (s)	MAXIMUM FAN SPEED (RPS)	MINIMUM RESOLUTION (Hz/COUNT)
0000 0000	0.25	512	2
0000 0001	0.5	256	1
0000 0010	1.0	128	0.5
0000 0011	2.0	64	0.25

The first 6 bits of the Tachometer Count-Time Register are always zero, and the last 2 bits set the count time (Table 9). The count time may be determined from the following equation:

$$t_{COUNT} = 0.25s \times 2^{K_{COUNT}}$$

where K_{COUNT} is the numerical value of the two 2LSBs. The 0.25 factor has a ±10% uncertainty.

Upon power-up, the Tachometer Count Registers reset to 00h and the Tachometer Count-Time Register sets a 1s integration time.

Digital-to-Analog Converter

When using the open-loop mode of operation, the DAC Register sets the voltage on the low side of the fan. An internal operational amplifier compares the feedback voltage (V_{FB}) with the reference voltage set by the 8-bit DAC, and adjusts the output voltage (V_{OUT}) until the two input voltages are equal. The voltage at the FB pin may be determined by the following equation:

$$V_{FB} = (10 \times V_{REF} \times K_{DAC}) / 256$$

and the voltage across the fan is:

$$V_{FAN} - \left(\frac{90k}{10k} + 1\right)\left(\frac{K_{DAC}}{256} V_{REF}\right)$$

where K_{DAC} is the numerical value of the DAC Register and V_{REF} = 1.5V. The minimum feedback voltage is limited by the voltage drop across the external MOSFET ($R_{ON} \times I_{FAN}$), and the maximum voltage is limited by the fan's supply voltage (V_{FAN}). For linear opera-

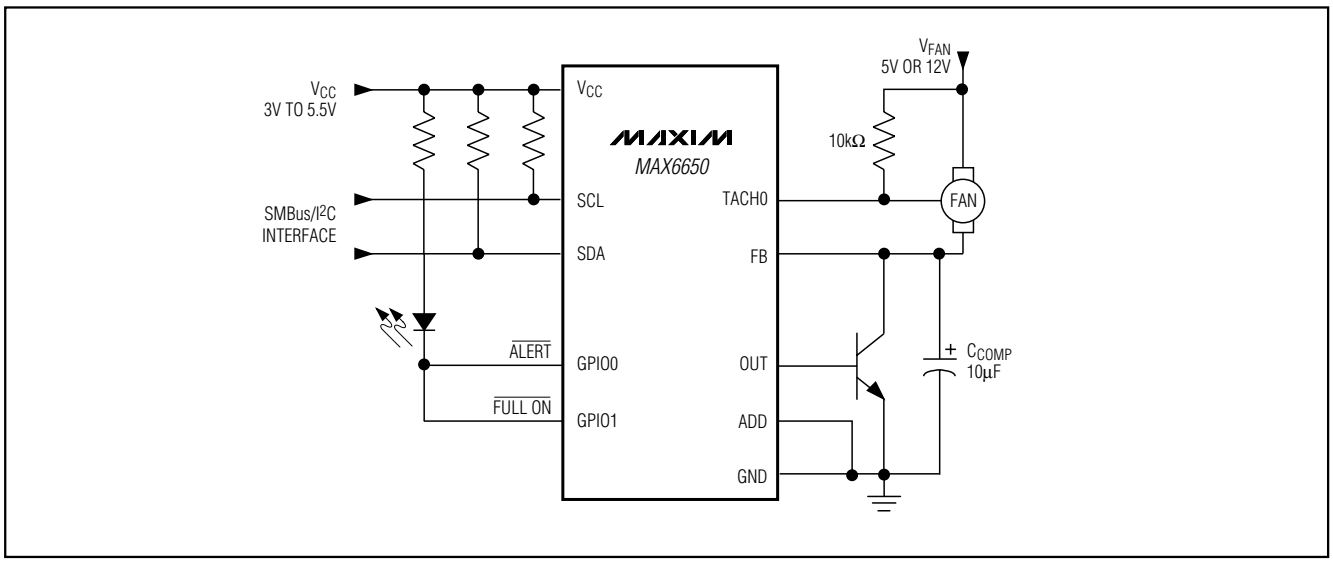


Figure 6. Fan Control with a Bipolar Transistor

tion, use DAC values between 08h and TB0h (see *Typical Operating Characteristics*). When using the closed-loop mode of operation, the contents of the DAC Register are ignored. When writing to the DAC, wait at least 500µs before attempting to read back.

Power-on Reset (POR)

The MAX6650/MAX6651 have volatile memory. To prevent ambiguous power-supply conditions from corrupting the data in the memory and causing erratic behavior, a POR voltage detector monitors V_{CC} and clears the memory if V_{CC} falls below 1.6V. When power is first applied and V_{CC} rises above 1.6V, the logic blocks begin operating (though reads and writes at V_{CC} levels below 3V are not recommended).

Power-up defaults include the following:

- All alarms are disabled.
- Prescale divider is set to 4.
- Fan speed is set in full-on mode.

See Table 2 for the default states of all registers.

Applications Information

MOSFET and Bipolar Transistor Selection

The MAX6650/MAX6651 drive an external N-channel MOSFET that requires five important parameters for proper selection: gate-to-source conduction threshold, maximum gate-to-source voltage, drain-to-source breakdown voltage, current rating, and drain-to-source on-resistance ($R_{DS(ON)}$). Gate-to-source conduction threshold must be compatible with available V_{CC}. The maximum gate-to-source voltage and the drain-to-source breakdown voltage rating should both be at least a few volts higher than the fan supply voltage (V_{FAN}). Choose a MOSFET with a maximum continuous drain current rating higher than the maximum fan current. $R_{DS(ON)}$ should be as low as practical to maximize the feedback voltage range. Maximum power dissipation in the power transistor can be approximated by $P = (V_{FAN} \times I_{FAN(MAX)}) / 4$. Bipolar power transistors are practical for driving small and midsize fans (Figure 6). Very-high-current fans may require output transistor base current greater than the MAX6650's 50mA drive capability. Bipolar Darlington transistors will work but have poor saturation characteristics and could lose up to 2V to 3V of drive voltage.

Resistor Selection

The tachometer input voltages ($V_{TACH\_}$) and feedback voltage (V_{FB}) cannot exceed 13.2V (see *Absolute Maximum Ratings*). When using a fan powered by a 13.2V or greater supply (V_{FAN}), protect these inputs from overvoltage conditions with series resistors. The resistance required to protect these pins may be calculated from the following equation:

$$R_{PROTECT} = [(V_{FAN(MAX)} - 13.2V) \times R_{IN}] / 13.2V$$

where $V_{FAN(MAX)}$ is the worst-case maximum supply voltage used to power the fan and R_{IN} is the input

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

impedance of the tachometer input (150kΩ max) or the feedback input (150kΩ max).

Compensation Capacitor

A compensation capacitor is needed from the fan's low side to ground to stabilize the analog control loop. Typically, this capacitor should be 10µF, but depending on the type of fan being used, a value between 1µF and 100µF may be required. The proper value has been selected when no ringing is present on the voltage at the fan's low side.

Table 10. Fan Manufacturers

MANUFACTURER	FAN MODEL OPTION
Comair Rotron	All DC brushless models can be ordered with optional tachometer output.
EBM-Papst	Tachometer output optional on some models.
NMB	All DC brushless models can be ordered with optional tachometer output.
Panasonic	Panaflo and flat unidirectional miniature fans can be ordered with tachometer output.
Sunon	Tachometer output optional on some models.

Fan Selection

For closed-loop operation and fan monitoring, the MAX6650/MAX6651 require fans with tachometer outputs. A tachometer output is typically specified as an option on many fan models from a variety of manufacturers. Verify the nature of the tachometer output (open collector, totempole) and the resultant levels, and configure the connection to the MAX6650/MAX6651 accordingly. Note how many pulses per revolution are generated by the tachometer output (this varies from model to model and among manufacturers, though two pulses per revolution is the most common).

Table 10 lists the representative fan manufacturers and the models they make available with tachometer outputs.

Low-Speed Operation

Brushless DC fans increase reliability by replacing mechanical commutation with electronic commutation. By lowering the voltage across the fan to reduce its speed, the MAX6650/MAX6651 are also lowering the supply voltage for the electronic commutation and tachometer electronics. If the voltage supplied to the fan is lowered too far, the internal electronics may no longer function properly. Some of the following symptoms are possible:

- The fan may stop spinning.
- The tachometer output may stop generating a signal.
- The tachometer output may generate more than two pulses per revolution.

The problems that occur, and the supply voltages at which they occur, depend on which fan is used. As a

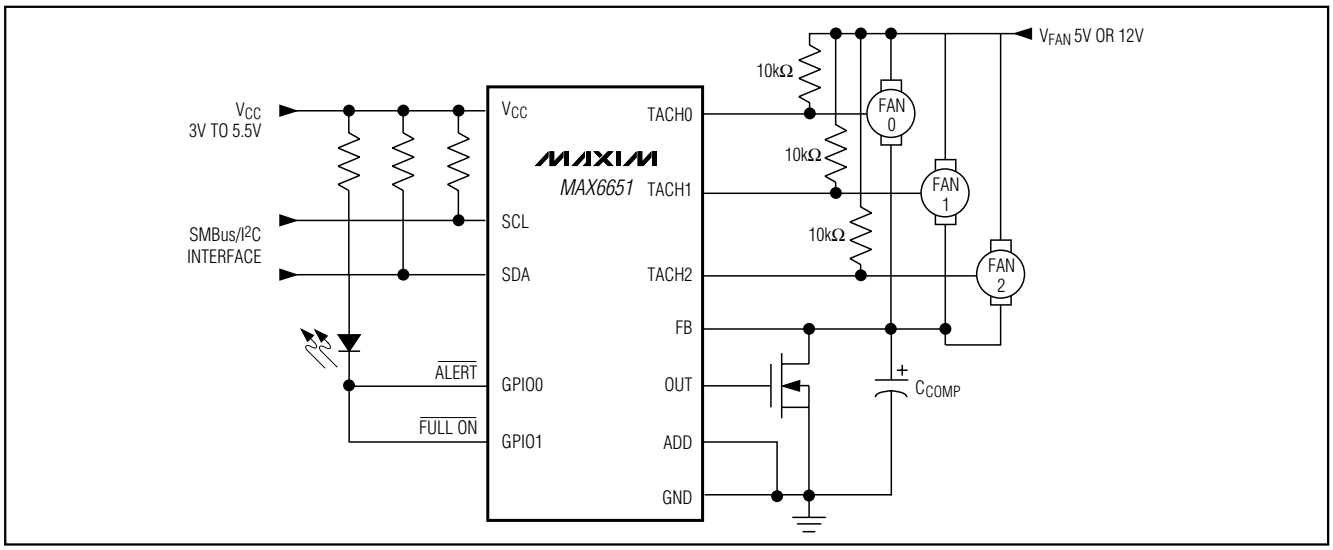


Figure 7. Using the MAX6651 to Control Parallel Fans

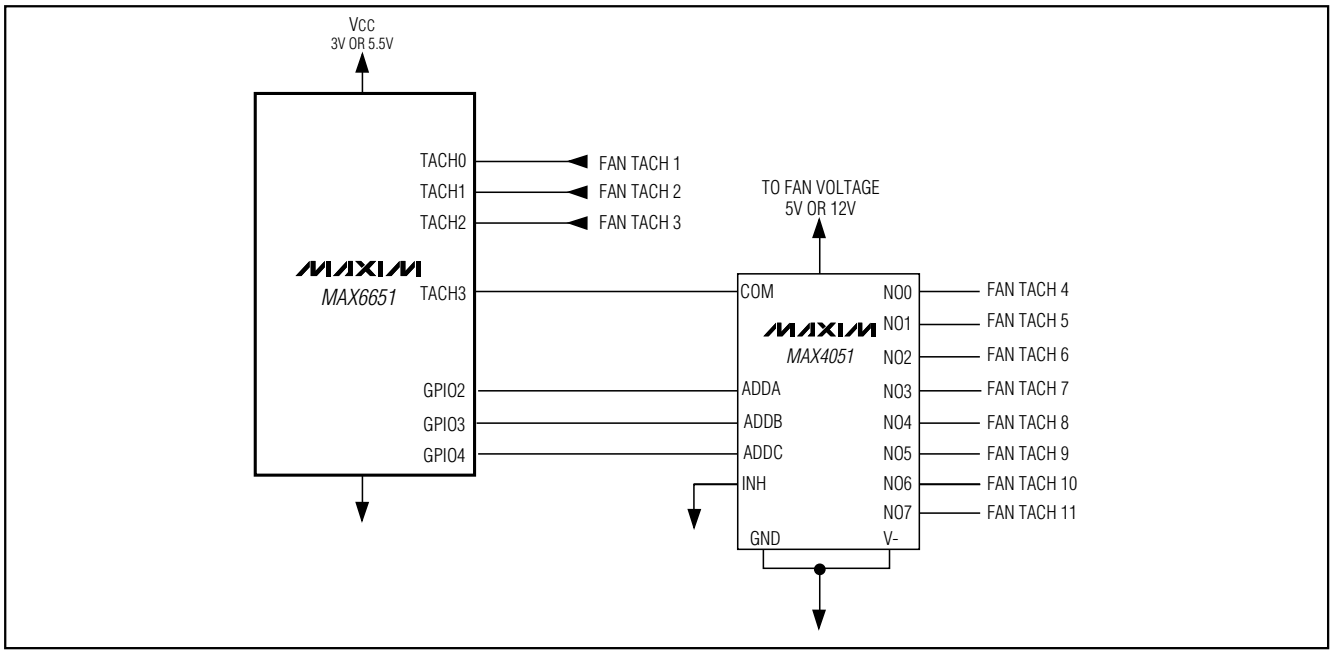


Figure 8. Monitoring Multiple Fans

very rough rule of thumb, 12V fans can be expected to experience problems somewhere around 1/4 to 1/2 their rated speed.

Predicting Future Fan Failure

In systems that require maximum reliability, such as servers and network equipment, it can be advantageous to predict fan failure before it actually happens, to alert the system operator before the fan fails, minimizing down time. The MAX6650 allows the user to monitor the fan's condition through the following modes.

Full-On Mode

By occasionally (over a period of days or weeks) turning the fan on full and measuring the resultant speed, a failing fan can be detected by a trend of decreasing speeds at a given power-supply voltage. Power-up is a convenient time to measure the maximum fan speed.

Open-Loop Mode

The fan's condition can also be monitored using open-loop mode. By characterizing the fan while it is new, fan failure can be determined by writing a predetermined value to the DAC and measuring the resultant fan speed. A decrease over time of the resultant speed may be an indication of future fan failure.

Closed-Loop Mode

The MAX6650 allows the system to read the DAC value used to regulate the fan speed. For a given speed, a significant change in the required DAC value may indicate future fan problems.

Monitoring More than 4 Fans

Use the MAX6651 to monitor up to four fans at a time (Figure 7). For systems requiring more than four fans, Figure 8 shows an application using an analog multiplexer (mux) to monitor 11 fans. GPIO2, GPIO3, and GPIO4 are connected to the mux's address pins. By writing the appropriate value to the GPIO pins, the desired tachometer gets selected and counted by the TACH3 input. Because the TACH inputs are double-buffered, and only sampled every other time slot, it is important to wait at least 4 times the tachometer count time before reading the register after changing the mux address. In the extreme case, a total of 25 fans can be monitored using three multiplexers connected to TACH1, TACH2, and TACH3. Do not connect TACH0 to a mux if the MAX6651 is under closed-loop mode.

N + 1 Fan Application

As shown in Figure 9, if any MAX6650 cannot maintain speed regulation, all other fans will automatically be turned on full. This can be useful in high-reliability systems where any single fan failure should not cause

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

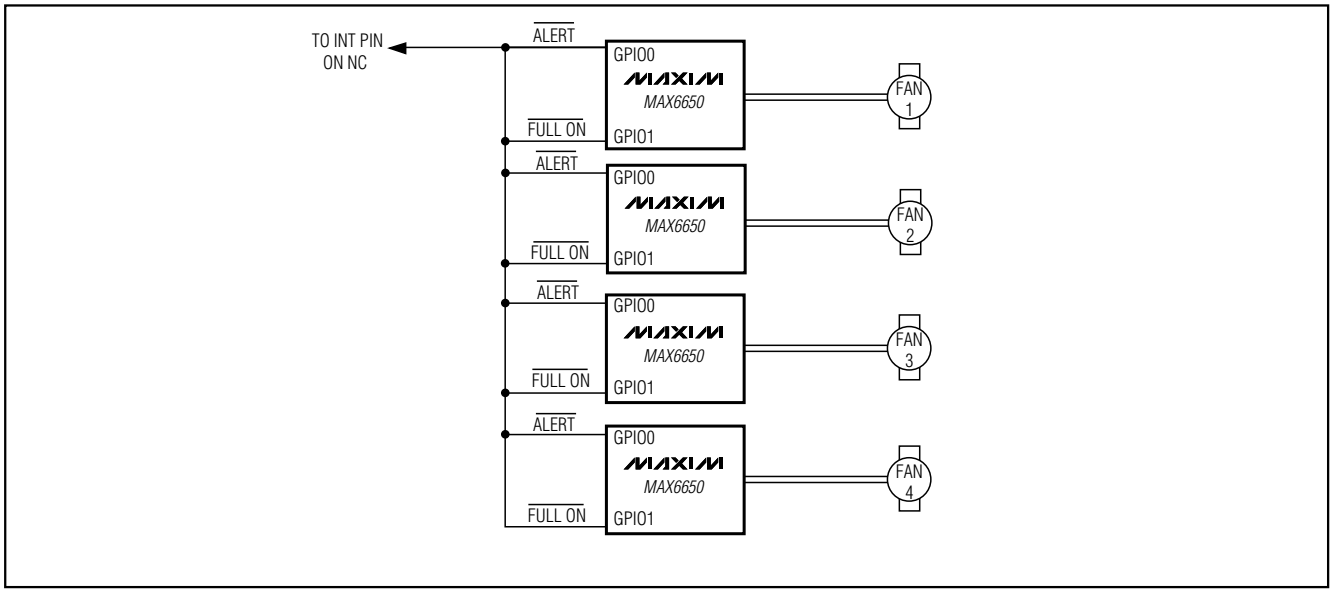


Figure 9. N + 1 Application

downtime. The system should be designed so that the number of fans used is one more than are actually needed. This way, there is sufficient cooling even if a fan fails. With all fans operating correctly, it is unnecessary to run the fans at their maximum speed. Reducing fan speed can reduce noise and increase the life of the fans. However, once a fan fails, it is important that the remaining fans spin at their maximum speed.

In Figure 9, all the GPIO0s are configured as $\overline{\text{ALERT}}$ outputs, and all the GPIO1s are configured as $\overline{\text{FULL ON}}$ inputs. If any MAX6650 generates an $\overline{\text{ALERT}}$ (indicating failure), the remaining MAX6650s will automatically turn their fans on full.

Temperature Monitoring and Fan Control

The circuit shown in Figure 10 provides complete temperature monitoring and fan control. The MAX1617A (a remote/local temperature serial interface with SMBus) monitors temperature with a diode-connected transistor. Based on the temperature readings provided by the MAX1617A, the µC can adjust the fan speed proportionally with temperature. Connecting the $\overline{\text{ALERT}}$ output of the MAX1617A to the $\overline{\text{FULL ON}}$ input of the MAX6650/MAX6651 (see the *General-Purpose Input/Output* section) allows the fan to turn on fully if the MAX1617A detects an overtemperature condition.

MAX6501 Hardware Fail-Safe

Figure 11 shows an application using a MAX6501 as a hardware fail-safe. The MAX6650 has its GPIO1 configured as $\overline{\text{FULL ON}}$ input. The MAX6501 $\overline{\text{TOVER}}$ pin goes low whenever its temperature goes above a preset value. This pulls the $\overline{\text{FULL ON}}$ pin (GPIO1) low, forcing the fan to spin at its maximum speed. Figure 12 shows the use of multiple MAX6501s. The MAX6501 has an open-drain output, allowing multiple devices to be wire ORed to the $\overline{\text{FULL ON}}$ input. This configuration allows fail-safe monitoring of multiple locations around the system.

Hot-Swap Application

Hot swapping of a fan can be detected using the circuit in Figure 13 where GPIO2 is configured to generate an alert whenever it is pulled low. As long as the fan card is connected, GPIO2 is high. However, when the fan card is removed, a 2.2kΩ resistor pulls GPIO2 low, causing an interrupt. This signals to the system that a hot swap is occurring.

Step-by-Step Part Selection and Software Setup

Determining the Fan System Topology

The MAX6650/MAX6651 support three fan system topologies. These are single fan control, parallel fan control, and synchronized fan control.

Single Fan Control

The simplest configuration is a single MAX6650 for each fan. If two or more fans are required per system, then additional MAX6650 controllers are used (one per fan). The advantage of this configuration is the ability to

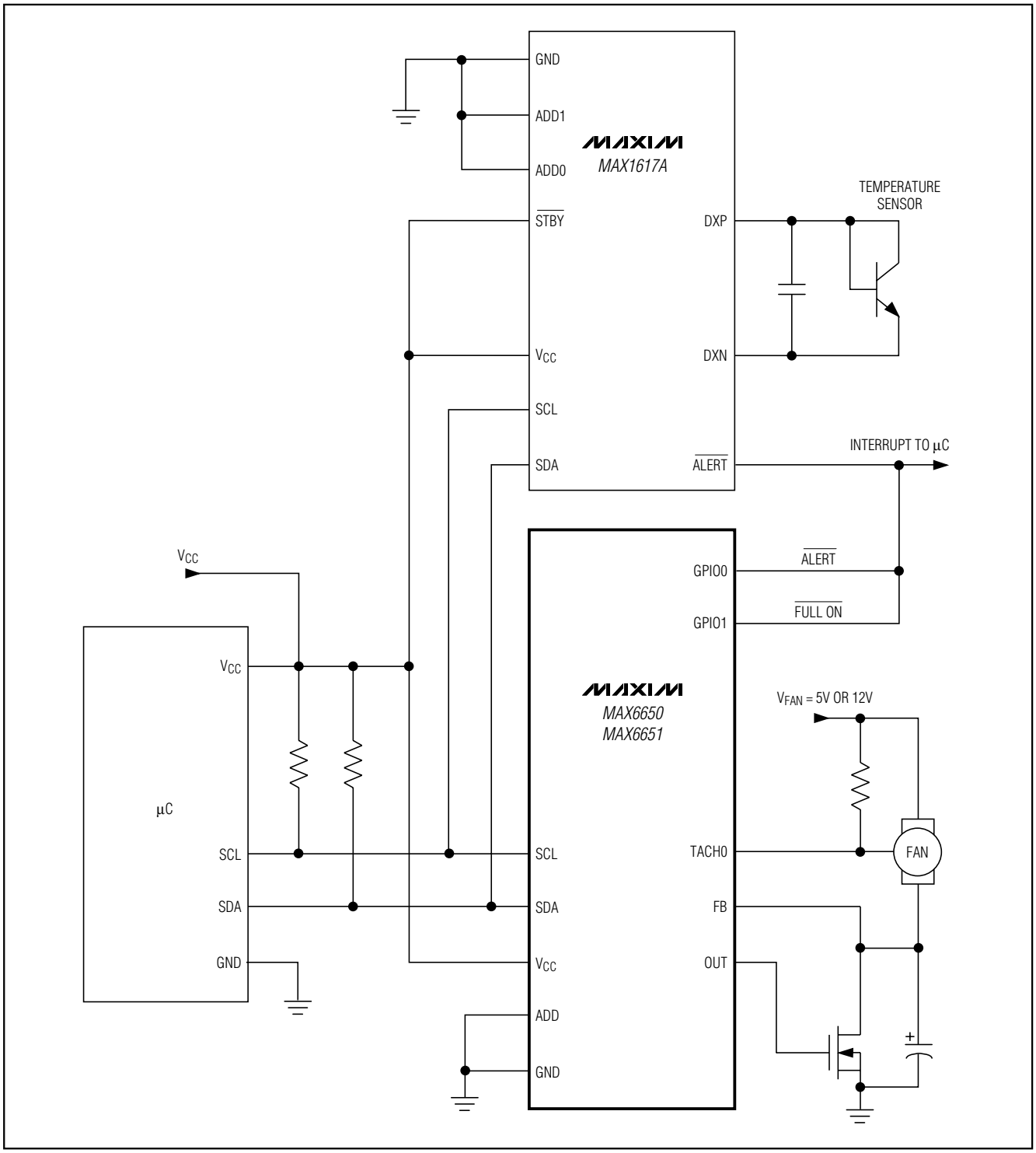


Figure 10. Temperature Monitoring and Fan Control

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

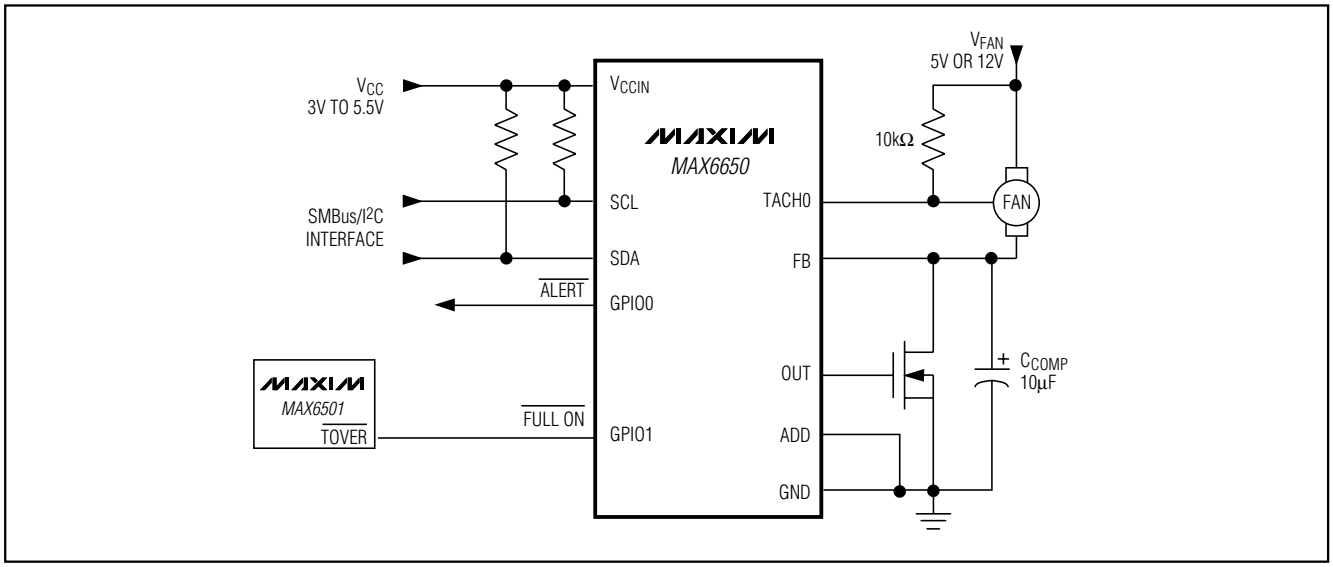


Figure 11. MAX6501 Hardware Fail-Safe

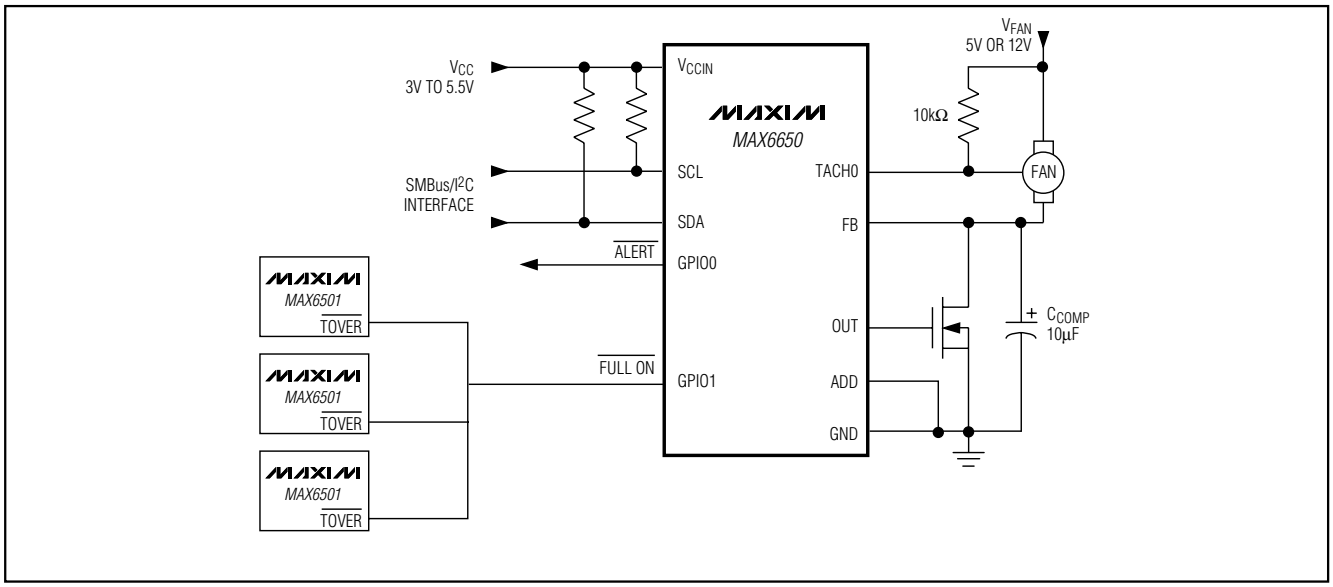


Figure 12. MAX6501 Hardware Fail-Safe

independently control each fan. The disadvantage is cost, size, and complexity.

For single fan control, use the MAX6650 (unless additional GPIOs are needed).

Parallel Fan Control

If multiple fans are required but independent control is not, then a single MAX6650/MAX6651 connected to two or more fans in parallel may make sense (Figure 7). The obvious advantage is simplicity, size, and cost savings. If all the fans connected in parallel are the same type, they will tend to run at similar speeds. However, if one or more of the fans are wearing out, speed mismatches can occur. The MAX6651 allows the system to monitor up to four fans, ensuring any significant speed mismatches can be detected.

For parallel fan control while monitoring up to four fan speeds, select the MAX6651.

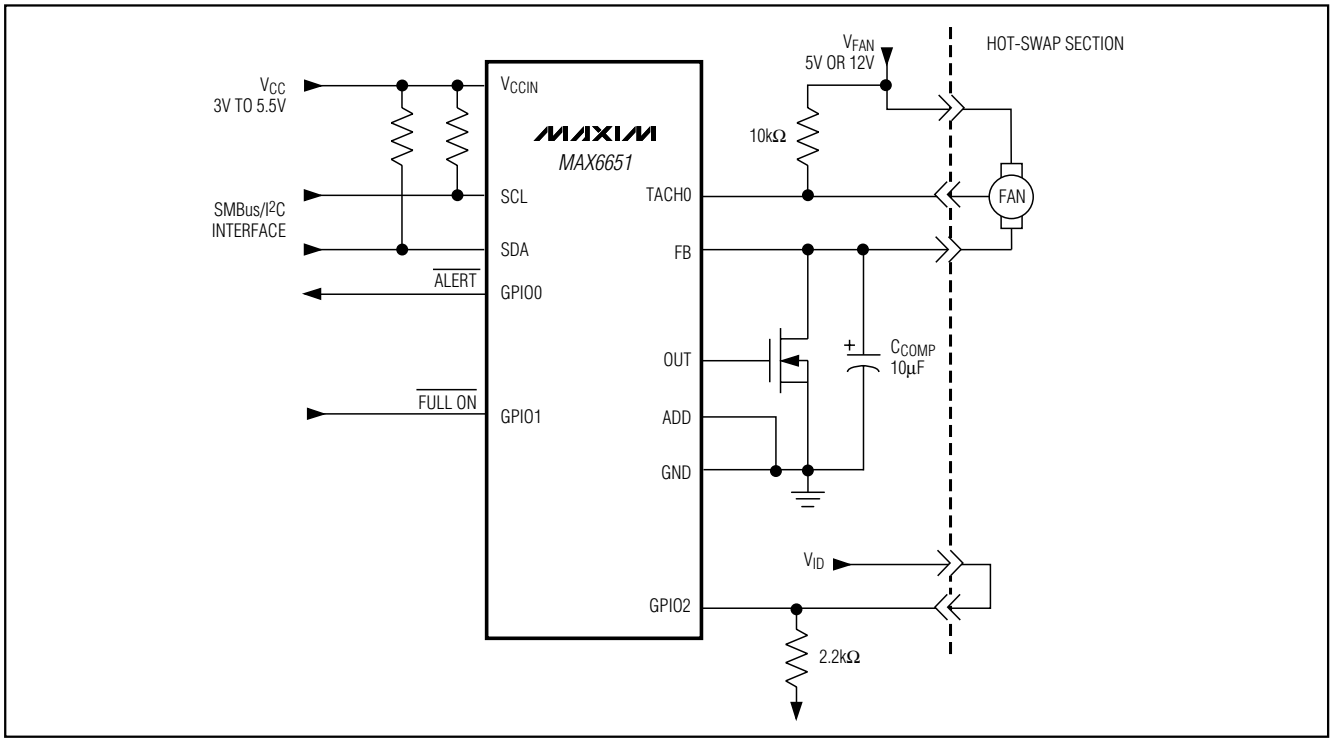


Figure 13. Hot-Swap Application

For parallel fan control while monitoring only a single fan, select the MAX6650.

Synchronized Fan Control (MAX6651 Only)

In systems with multiple fans, an audible beat frequency can sometimes be detected due to fan speed mismatch. This happens in systems where fans are connected in parallel or in systems with a MAX6650 controlling each fan. In parallel fan systems, speed mismatches occur because no two fans are identical. Slight mechanical variations or loading differences can result in enough of a speed mismatch to cause an audible beat.

Even in systems where there is a MAX6650/MAX6651 for each fan, there can still be speed mismatches. This is primarily due to the oscillator tolerance. The MAX6650/MAX6651 oscillator tolerance is specified to be ±10%. In the worst case, this could result in a 20% (one 10% high, one 10% low) speed mismatch.

The solution is to use a single MAX6651 for each fan, and configure the parts to use a shared clock. The shared clock can either be an external system clock or one of the MAX6651's internal clocks. If an external clock is used, its frequency can range from approximately 50kHz to 500kHz.

For synchronized fan control, select the MAX6651.

Combination

In more complex systems, a combination of some or all of the above control types may be needed.

Choosing a Fan

Once the topology is chosen, the next step is to choose a fan. See the appropriate section.

Enter a zero in bit 3 of the configuration register for a 5V fan and 1 for a 12V fan.

Configuring this bit also adjusts the tachometer input threshold voltage. This optimizes operation of the MAX6650/MAX6651 for the operating voltage of the fan being used.

Setting the Mode of Operation

The MAX6650/MAX6651 have four modes of operation as determined by bits 5 and 6 of the configuration register: full on, full off, open loop, and closed loop.

Full-On

The full-on mode applies the maximum available voltage across the fan, guaranteeing maximum cooling. Full-on mode can be entered through software or hard-

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

ware control. To enter full-on mode through hardware, see the *Setting Up the GPIOs* section. Note that a hardware full-on overrides all other modes.

Configure the MAX6650/MAX6651 to run in software full-on mode by entering 00 into bits 5 and 4 of the configuration register.

Full-Off

The full-off mode removes all the voltage across the fan, causing the fan to stop. Because the MAX6650/MAX6651 work by controlling the voltage on the low side of the fan, either 5V or 12V will be on both leads.

Enter full-off mode by entering 01 into bits 5 and 4 of the configuration register.

Open Loop

In open-loop mode, the MAX6650/MAX6651 do not actually regulate the fan speed. Speed regulation requires an external µC. Although open-loop mode allows maximum flexibility, it also requires the most software/processor overhead.

In open-loop mode, the MAX6650/MAX6651 act as an SMB/I²C-controlled voltage regulator. The µC adjusts the voltage across the fan by writing an 8-bit value to the DAC register. This gives the µC direct control of the voltage across the fan. Speed regulation is accomplished by periodically reading the tachometer register(s) and adjusting the DAC register appropriately. The DAC value controls the voltage across the fan according to the following equation:

$$V_{FAN} = V_{FAN\_SUPPLY} - [((R2) / R1) + 1] \times V_{REF} \times K_{DAC} / 256$$

where V_{FAN} = the voltage across the fan, V_{FAN_SUPPLY} = the supply voltage for the fan (5V or 12V), R2 = 90kΩ (typ), R1 = 10kΩ (typ), V_{REF} = 1.5V (typ), and K_{DAC} = the value in the DAC register.

Note several important things in this equation. First, the voltage across the fan moves in the opposite direction of the DAC value. In other words, low DAC values correspond to higher voltages across the fan and therefore higher speeds. Second, DAC values greater than 90kΩ will result in 0V across a 12V fan. Similarly, DAC values greater than 76 will produce 0V across a 5V fan. This limits the useful range of the DAC from 0 to 180 for 12V fans and 0 to 76 for 5V fans.

Remember that device tolerances can cause the output voltage value to vary significantly from unit to unit and over temperature. However, because this voltage is within a closed speed-control loop, such errors are corrected by the loop.

Below is a possible strategy for controlling the fan under open-loop mode:

1) On power-up, put the device in open-loop mode with a DAC value of 00 (full speed).
2) Allow the fan speed to settle.
3) Read the TACH register to determine the speed.
4) Gradually increase the DAC register value (in steps of 1 or 2) until the desired speed is obtained.

In open-loop mode, any one of the four tachometer registers (MAX6651) can be used to measure and regulate the fan's speed. This is especially useful in parallel fan systems where up to four fans will be controlled as one unit.

Care must be taken with this mode to prevent instability, which can be caused by trying to update the fan speed too often or in increments that are too large. Instability can result in the fan speeding up and slowing down repeatedly. Determining the proper update rate, as shown in the following steps, depends largely on the fan's mechanical time constant and the system's loop gain (DAC step sizes):

1) Enter open-loop mode by setting bits 5 and 4 of the control register to 11.
2) Determine the speed of the fan(s) by reading the TACH register(s).
3) Increase or decrease the DAC register to decrease or increase the voltage across the fan, thereby adjusting its speed.

Closed Loop

In closed-loop mode, the SMBus/I²C master (usually a µC) writes a desired fan speed to the MAX6650/MAX6651, and the device automatically adjusts the voltage across the fan to maintain this speed. This operation mode requires less software/processor overhead than the open-loop mode. Once the desired speed has been written, the MAX6650/MAX6651 control the fan's speed independently, with no intervention required from the master. If desired, the MAX6650/MAX6651 can be configured to generate an interrupt if it is unable to regulate the fan's speed at the desired value (see *Setting Up Alarms*). The MAX6650/MAX6651 can regulate only the speed of the fan connected to the TACH0 input. Fans connected in parallel to the TACH0 fan will tend to run at similar speeds (assuming similar fans). When going from full-off to closed-loop-mode, it is recommended following this sequence:

1) Full-off mode
2) Full-on mode (with sufficient pause to initiate movement)
3) Closed-loop mode

The MAX6650 regulates fan speed in the following manner. The output of an internal 254kHz oscillator is divided by 128, generating a roughly 2kHz signal. This signal is divided by 1 plus the value in the speed register and is used as a reference frequency. For example, 02h in the speed register will result in a 667Hz [2kHz / (02h+1)] reference frequency, which is then compared against the frequency at the tachometer input divided by the prescaler value. The MAX6650/MAX6651 attempt to keep the tachometer frequency divided by the prescaler equal to the reference frequency by adjusting the voltage across the fan. If the tachometer frequency divided by the prescaler value is less than the reference frequency, the voltage across the fan is increased. Remember that the tachometer will give two pulses per revolution of the fan. The following equations describe the operation.

When in regulation:

$$[f_{CLK} / (128 \times (K_{TACH} + 1))] = 2 \times FanSpeed / K_{SCALE}$$

where f_{CLK} = oscillator frequency (either the 254kHz internal oscillator or the externally applied clock), K_{TACH} = the value in the speed register, FanSpeed = the speed of the fan in revolutions per second (Hz), K_{SCALE} = the prescaler value (1, 2, 4, 8, or 16).

Solving for all four variables:

$$K_{TACH} = [(f_{CLK} \times K_{SCALE}) / (256 \times FanSpeed)] - 1$$

$$K_{SCALE} = [256 \times FanSpeed \times (K_{TACH} + 1)] / f_{CLK}$$

$$FanSpeed = K_{SCALE} \times f_{CLK} / [256 \times (K_{TACH} + 1)]$$

$$f_{CLK} = 256 \times FanSpeed \times (K_{TACH} + 1) / K_{SCALE}$$

If the internal oscillator is used, setting f_{CLK} to 254kHz can further reduce the equations:

Equation 1: $K_{SCALE} = FanSpeed \times (K_{TACH} + 1) / 992$

Equation 2: $K_{TACH} = (992 \times K_{SCALE} / FanSpeed) - 1$

Equation 3: $FanSpeed = 992 \times K_{SCALE} / (K_{TACH} + 1)$

Enter closed-loop mode by entering 10 into bits 5 and 4 of the configuration register.

Note that in equation 3, the fan speed is inversely proportional to (K_{TACH} + 1). This means the regulated fan speed is a nonlinear function of the value written to the speed register. Low values written to the speed register can result in large relative changes in fan speed. For best results, design the system so that small values (such as 02h) are not needed. This is easily accomplished because an 8-bit speed register is used, and fan-speed control should rarely need more than 16 speeds. A good compromise is to design the system (by selecting the appropriate prescaler value) so that the maximum-rated speed of the fan occurs when the speed register equals approximately 64 (decimal). Although 64 is a good target value, values between 20 and 100 will work fine.

The prescaler value also affects the response time and the stability of the speed-control loop. Adjusting the prescaler value effectively adjusts the loop gain. A larger prescaler value will slow the response time and increase stability, while a smaller prescaler value will yield quicker response time. The optimum prescaler value for response time and stability depends on the fan's mechanical time constant. Small, fast-spinning fans will tend to have small mechanical time constants and can benefit from smaller prescaler values. A good rule of thumb is to try the selected prescaler value in the target system. Set K_{TACH} to around 75% of full scale, and watch for overshoot or oscillation in the fan speed. Also look for overshoot or oscillation when K_{TACH} is changed from one value to another (e.g., from 75% of full-scale speed to 90% of full scale). If there is unacceptable overshoot or if the fan speeds up and slows down with K_{TACH}, set it to a constant value; increase the prescaler value.

Enter the appropriate prescaler value in bits zero to 2 of the configuration register.

Fan speed is a trade-off between cooling requirements, noise, power, and fan wear. In general, it is desirable (within limits) to run the fan at the slowest speed that will accomplish the cooling goals. This will reduce power consumption, increase fan life, and minimize noise. When calculating the desired fan speed, remember that the above equations are written in rotations per second (RPS), where most fans are specified in rotations per minute (RPM).

Write the desired fan speed to the speed register.

Example:

Assume the following:

- 12V fan is rated at 2000RPM at 12V.
- Use the internal oscillator (f_{CLK} = 254kHz).
- Desired fan speed = 1500RPM (25RPS).

First, calculate an appropriate prescaler value (K_{SCALE}) using equation 1. Attempt to get K_{TACH} as close to 64 as possible for the maximum speed of 2000RPM.

- Set FanSpeed = 33.3RPS (2000RPM/60).
- Set K_{TACH} = 64.
- Solving equation 1 gives K_{SCALE} = 2.18.

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

We will start with K_{SCALE} = 2 (to increase stability, a 4 could be tried, or to improve response time, a 1 could be tried).

Second, calculate the appropriate value for the Speed Register (K_{TACH}) using equation 2.

- Set FanSpeed = 25RPS (1500PRM/60).
- Solving for equation 2 gives K_{TACH} = 78 for K_{SCALE} = 2, K_{TACH} = 39 for K_{SCALE} = 1, or K_{TACH} = 158 for K = 4.

Determining the Tachometer Count Time

To monitor the fan speed using the SMBus/I²C, the next step is to determine the tachometer count time. In systems running in open-loop mode, this is necessary. In closed-loop or full-speed mode, reading the tachometer can serve as a valuable check to ensure the fan and the control loop are operating properly.

The MAX6650/MAX6651 use an 8-bit counter to count the tachometer pulses. This means the device can count from 0 to 255 tachometer pulses before overflowing. The MAX6650/MAX6651 can accommodate a large range of fan speeds by allowing the counting interval to be programmed. Smaller/faster fans should use smaller count times. Although larger fans could also use smaller count times, resolution would suffer. Choose the slowest count time that will not overflow under worst-case conditions. Fans are mechanical devices, and their speeds are subject to large tolerance variations. If an overflow does occur, the counter will read 255. The MAX6650/MAX6651 can be configured to generate an alert if an overflow is encountered (see *Setting Up Alarms*). Note that the prescaler value has no effect on the TACH0 register.

Enter the appropriate count-time value in the tachometer count-time register.

Example:

Assume a 12V fan rated at 2000 RPM.

To accommodate large tolerance variations, choose a count time appropriate for a maximum speed of 3000RPM; 3000RPM is 50RPS and generates a 100Hz (2 pulses/revolution) tachometer signal. Table 9 indicates a count time of 2s will optimize resolution. With a 2s count time, speeds as fast as 3825RPM can be monitored without overflow. The minimum resolution will be 15RPM or 0.75% of the rated speed of 2000RPM.

Setting Up the GPIOs

To increase versatility, the MAX6650/MAX6651 have two and five general-purpose digital inputs/outputs, respectively. These GPIOs can be configured through the SMBus/I²C.

Digital Out Low

All GPIOs can be configured to output a logic-level low. The MAX6650/MAX6651 are designed to sink up to 10mA. This high sink current can be especially useful for driving LEDs.

For GPIO3 and GPIO4, write a zero to the appropriate location in the GPIO definition register.

For GPIO0, GPIO1, and GPIO2, write a 10 to the appropriate location in the GPIO definition register.

Digital Out High

All GPIOs can be configured to generate a logic-level high. An output high is generated using an open-drain output stage with an internal pullup resistor of nominally 100kΩ. The MAX6650/MAX6651 power-up default state is with all GPIOs configured as output highs.

For GPIO3 and GPIO4, write a 1 to the appropriate location in the GPIO definition register.

For GPIO0, GPIO1, and GPIO2, write an 11 to the appropriate location in the GPIO definition register.

Digital Input

Since a logic-level high output is open drain with an internal pullup, an external device can actively pull this pin low. The MAX6650/MAX6651 allow the user to read the GPIO value through the GPIO status register.

- Configure the GPIO as an output logic level high (see above).
- Read the state of the GPIO by reading the GPIO status register.

Alert Output

GPIO0 can also serve as an $\overline{\text{ALERT}}$ output. The $\overline{\text{ALERT}}$ output is designed to drive an interrupt on a µC. The $\overline{\text{ALERT}}$ output goes low whenever an enabled alarm condition occurs (see *Setting Up Alarms*).

Configure GPIO0 as an $\overline{\text{ALERT}}$ output by writing a 01 to bits 1 and 0 of the GPIO definition register.

Full-On Input

GPIO1 can also be configured as a full-on input. When the full-on pin is pulled low, the MAX6650/MAX6651 apply the full available voltage across the fan. This happens independently of the software mode of operation. This is a particularly valuable feature in high-reliability systems, designed to prevent software malfunctions from causing system overheating.

Configure GPIO1 as a full-on input by writing a 01 to bits 3 and 2 of the GPIO definition register.

Synchronizing Fans

GPIO2 can be configured to allow multiple MAX6651s to synchronize the speeds of the fans they are driving (Figure 14). Synchronization is accomplished by having one of the MAX6651s (or an external clock) serve as the clock master by configuring one of the GPIO2s in the system as a clock output. The remaining GPIO2s in the system need to be configured as clock inputs:

- Electrically connect all MAX6651 GPIO2s together.
- Configure one of the MAX6651's GPIO2s to be a clock output, using the GPIO Definition Register (set bits 5 and 4 to 01).
- Configure the rest of the GPIO2s as clock inputs, using the GPIO Definition Register (set bits 5 and 4 to 00).
- Configure all MAX6651s in closed-loop mode.
- Configure all prescaler values to be equal.
- Write identical values to all speed registers.

Setting Up Alarms

The MAX6650/MAX6651 can be configured to generate an $\overline{ALERT}$ output on GPIO0 whenever certain events, such as control loop out of regulation, tachometer overflow, or GPI01/GPI02 being driven low, occur. This is designed to enhance the "set and forget" functionality of the fan control system.

Configure GPIO0 to be an $\overline{ALERT}$ output (see above).

Minimum/Maximum Output Level Alarm

The minimum/maximum output level alarms are designed to warn the system when the MAX6650/MAX6651 are unable to maintain speed regulation in closed-loop mode. The MAX6650/MAX6651 maintain speed regulation by adjusting the voltage across the fan. If the desired speed can't be attained, one of these alarms will be generated. Possible causes for failure to attain the desired speed include system programming problems, incipient fan failure, and a programmed speed that the fan cannot support.

The minimum output alarm occurs when the DAC output is 00h. A DAC value of 00h means that the MAX6650/MAX6651 have applied the largest available voltage across the fan. This typically means the fan is unable to spin as fast as the desired speed.

The maximum output alarm occurs when the DAC value is FFh. A DAC value of FFh means the MAX6650/MAX6651 have tried to reduce the voltage across the fan to 0. Although this would seem to indicate the fan is spinning faster than the desired speed, this should rarely happen. If this alarm occurs, it probably indicates some type of system error.

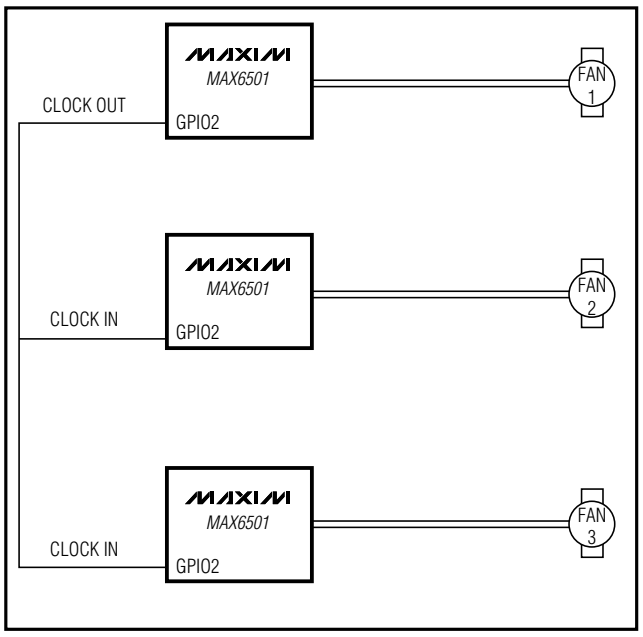


Figure 14. Synchronizing Fans

Enable the minimum/maximum output level alarm by setting bits 0 and 1 of the alarm enable register to 11.

Tachometer Overflow Alarm

If any tachometer counter overflows (reaches a count of 255), this alarm will be set.

Enable the overflow output level alarm by setting bit 2 of the alarm enable register bit to 1.

$\overline{GPIO1/2}$ Pulled Low

Enabling this alarm causes the $\overline{ALERT}$ output to go low whenever GPIO1 or GPIO2 is pulled low. This will occur independent of the configuration of GPIO1 or GPIO2.

Enable the GPIO1/GPIO2 output level alarms by setting bits 3 and/or 4 of the alarm enable register bit to 1.

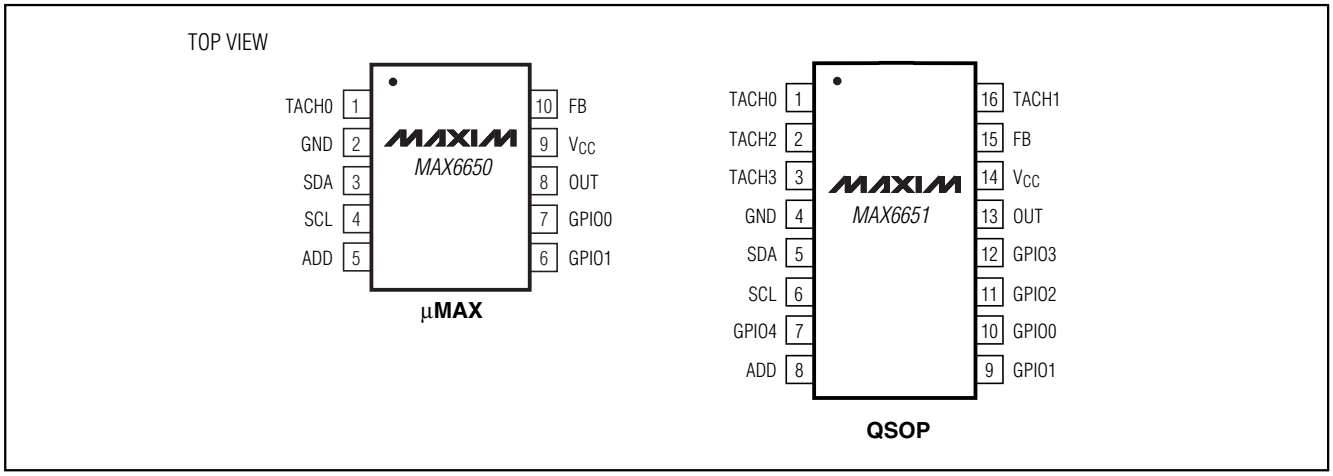
Clearing the $\overline{ALERT}$

Once an $\overline{ALERT}$ is generated, determine which alarm caused the $\overline{ALERT}$ pin to go low. Do this by reading the Alarm Status Register. An $\overline{ALERT}$ output will stay active (low) even if the condition that caused the alert is removed. Reading the Alarm Status Register clears the $\overline{ALERT}$, if the condition that caused the alert is gone. If the condition has not gone away, the $\overline{ALERT}$ will stay active. Disabling the alarm with the Alarm Enable Register will cause the $\overline{ALERT}$output to go inactive.

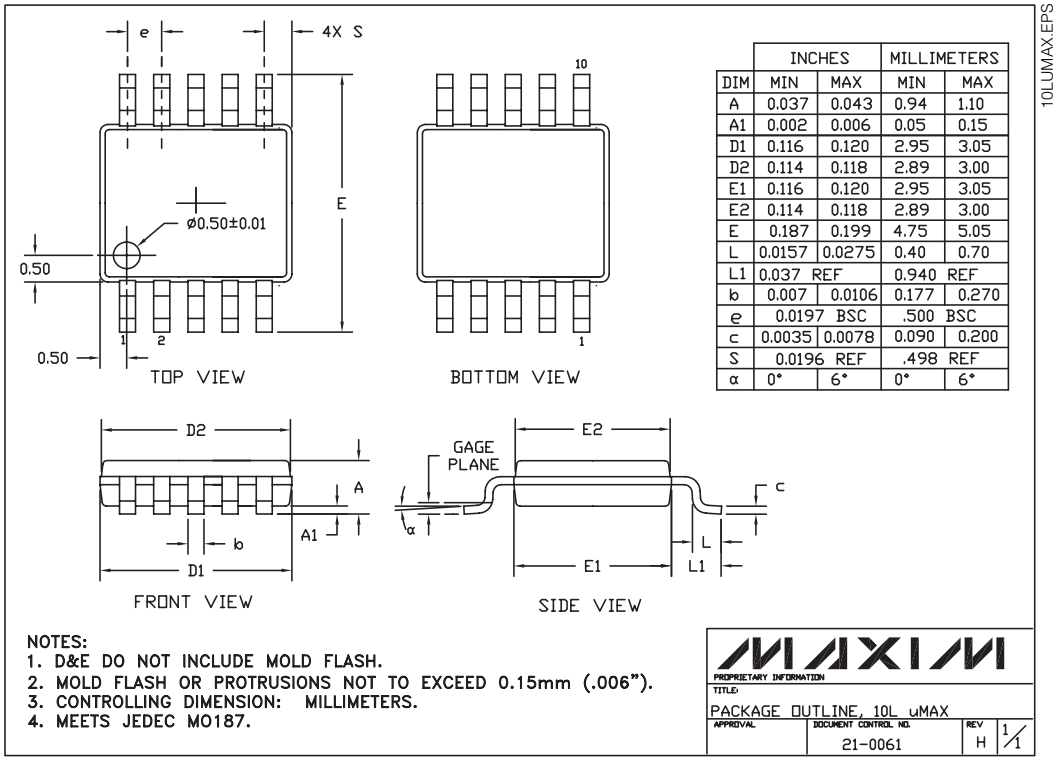
Read the Alarm Status Register.

Fan-Speed Regulators and Monitors with SMBus/I²C-Compatible Interface

Pin Configurations



Package Information



	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.037	0.043	0.94	1.10
A1	0.002	0.006	0.05	0.15
D1	0.116	0.120	2.95	3.05
D2	0.114	0.118	2.89	3.00
E1	0.116	0.120	2.95	3.05
E2	0.114	0.118	2.89	3.00
E	0.187	0.199	4.75	5.05
L	0.0157	0.0275	0.40	0.70
L1	0.037 REF		0.940 REF	
b	0.007	0.0106	0.177	0.270
e	0.0197 BSC		.500 BSC	
c	0.0035	0.0078	0.090	0.200
S	0.0196 REF		.498 REF	
α	0°	6°	0°	6°