Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ADM1031

Intelligent Temperature Monitor and Dual PWM Fan Controller

The ADM1031 is an ACPI-compliant, three-channel digital thermometer and under/overtemperature alarm for use in personal computers and thermal management systems. Optimized for the Pentium®III, the part offers a 1°C higher accuracy, which allows system designers to safely reduce temperature guard-banding and increase system performance.

Two PWM fan control outputs control the speed of two cooling fans by varying output duty cycle. Duty cycle values between 33% and 100% allow smooth control of the fans. The speed of each fan can be monitored via TACH inputs, which can be reprogrammed as analog inputs to allow speeds for 2-wire fans to be measured via sense resistors. The device also detects a stalled fan. A dedicated fan speed control loop provides control without the intervention of CPU software. It also ensures that if the CPU or system locks up, each fan can still be controlled based on temperature measurements, and the fan speed is adjusted to correct any changes in system temperature. Fan speed can also be controlled using existing ACPI software.

Two inputs (4 pins) are dedicated to remote temperature-sensing diodes with an accuracy of ±1°C, and an on-chip temperature sensor allows ambient temperature to be monitored. The device has a programmable $\overline{\text{INT}}$ output to indicate error conditions, and a dedicated $\overline{\text{FAN\_FAULT}}$ output to signal fan failure. The $\overline{\text{THERM}}$ pin is a fail-safe output for overtemperature conditions that can be used to throttle a CPU clock.

Features

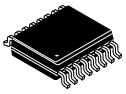
- Optimized for Pentium® III
  - ♦ Reduced Guard-banding Software
  - ♦ Automatic Fan Speed Control, Independent of CPU Intervention After Initial Setup
- 0.125°C Resolution on External Temperature Channels
- Control Loop to Minimal Acoustic Noise and Battery Consumption
- Remote Temperature Measurement Accurate to 1°C Using Remote Diode (Two Channels)
- Local Sensor with 0.25°C Resolution
- Pulse Width Modulation (PWM) Fan Control for 2 Fans
- Programmable PWM Frequency and PWM Duty Cycle
- Tach Fan Speed Measurement (Two Channels)
- Analog Input to Measure Fan Speed of 2-wire Fans (Using Sense Resistor)
- 2-wire System Management Bus (SMBus) with ARA Support
- Overtemperature $\overline{\text{THERM}}$ Output Pin for CPU Throttling
- Programmable $\overline{\text{INT}}$ Output Pin
- Configurable Offsets for Temperature Channels 3.0 V to 5.5 V Supply Range
- Shutdown Mode to Minimize Power Consumption
- Limit Comparison of All Monitored Values
- This is a Pb-Free Device

Applications

- Notebook PCs, Network Servers, and Personal Computers
- Telecommunications Equipment

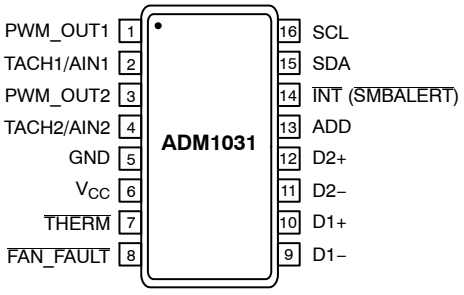
ON Semiconductor®

http://onsemi.com

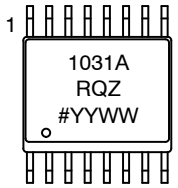


QSOP−16
CASE 492

PIN ASSIGNMENT



MARKING DIAGRAM



1031ARQZ = Specific Device Code
# = Pb-Free Package
YY = Date Code
WW = Work Week

ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 31 of this data sheet.

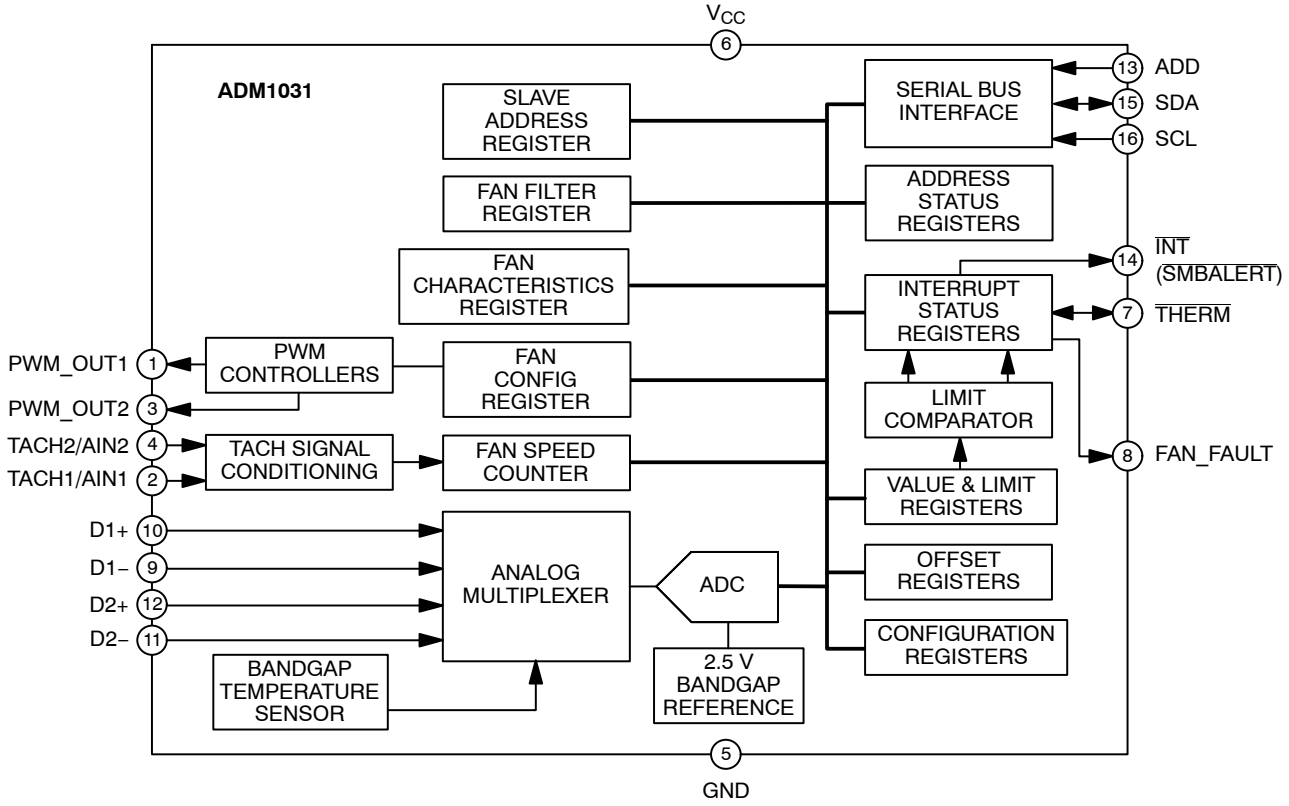


Figure 1. Functional Block Diagram

Table 1. ABSOLUTE MAXIMUM RATINGS

Parameter	Rating	Unit
Positive Supply Voltage (V_{CC})	6.5	V
Voltage on Any Input or Output Pin	−0.3 to +6.5	V
Input Current at Any Pin	±5	mA
Package Input Current	±20	mA
Maximum Junction Temperature ($T_{J\ max}$)	150	°C
Storage Temperature Range	−65 to +150	°C
Lead Temperature, Soldering Vapor Phase (60 sec) Infrared (15 sec)	 215 200	°C
ESD Rating − All Pins	2000	V

Stresses exceeding Maximum Ratings may damage the device. Maximum Ratings are stress ratings only. Functional operation above the Recommended Operating Conditions is not implied. Extended exposure to stresses above the Recommended Operating Conditions may affect device reliability.

NOTE: This device is ESD sensitive. Use standard ESD precautions when handling.

Table 2. THERMAL CHARACTERISTICS

Package Type	θ_{JA}	θ_{JC}	Unit
16-lead QSOP	105	39	°C/W

NOTE: θ_{JA} is specified for the worst-case conditions, that is, a device soldered in a circuit board for surface-mount packages.

Table 3. PIN ASSIGNMENT

Pin No.	Mnemonic	Description
1	PWM_OUT1	Digital Output, Open-Drain. Pulse width modulated output to control fan speed. Requires pullup resistor (10 kΩ typical).
2	TACH1/AIN1	Digital/Analog Input. Fan tachometer input to measure FAN1 fan speed. Can be reprogrammed as an analog input to measure speed of a 2-wire fan via a sense resistor (2 Ω typical).
3	PWM_OUT2	Digital Output, Open-Drain. Pulse width modulated output to control FAN2 fan speed. Requires pullup resistor (10 kΩ typical).
4	TACH2/AIN2	Digital/Analog Input. Fan tachometer input to measure FAN2 fan speed. Can be reprogrammed as an analog input to measure speed of a 2-wire fan via a sense resistor (2 Ω typical).
5	GND	System Ground.
6	VCC	Power. Can be powered by 3.3 V standby power if monitoring in low power states is required.
7	$\overline{\text{THERM}}$	Digital I/O, Open-Drain. An active low thermal overload output that indicates a violation of a temperature set point (overtemperature). Also acts as an input to provide external fan control. When this pin is pulled low by an external signal, a status bit is set, and the fan speed is set to full-on. Requires pullup resistor (10 kΩ).
8	$\overline{\text{FAN\_FAULT}}$	Digital Output, Open-Drain. Can be used to signal a fan fault. Drives second fan to full speed if one fan fails. Requires pullup resistor (typically 10 kΩ).
9	D1−	Analog Input. Connected to cathode of first remote temperature-sensing diode. The temperature-sensing element is either a Pentium III substrate transistor or a general-purpose 2N3904.
10	D1+	Analog Input. Connected to anode of first remote temperature-sensing diode.
11	D2−	Analog Input. Connected to cathode of second remote temperature-sensing diode.
12	D2+	Analog Input. Connected to anode of second remote temperature-sensing diode.
13	ADD	Three-State Logic Input. Sets two lower bits of device SMBus address.
14	$\overline{\text{INT}}$($\overline{\text{SMBALERT}}$)	Digital Output, Open-Drain. Can be programmed as an interrupt (SMBus $\overline{\text{ALERT}}$) output for temperature/fan speed interrupts. Requires pullup resistor (10 kΩ typical).
15	SDA	Digital I/O, Serial Bus Bidirectional Data. Open-drain output. Requires pullup resistor (2.2 kΩ typical).
16	SCL	Digital Input, Serial Bus Clock. Requires pullup resistor (2.2 kΩ typical).

Table 4. ELECTRICAL CHARACTERISTICS ($T_A = T_{MIN}$ to T_{MAX}, $V_{CC} = V_{MIN}$ to V_{MAX}, unless otherwise noted.) (Note 1)

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
POWER SUPPLY					
Supply Voltage, V_{CC}		3.0	3.3	3.6	V
Supply Current, I_{CC}	Interface inactive, ADC active Standby mode	– –	1.4 32	3.0 50	mA μA
TEMPERATURE-TO-DIGITAL CONVERTER					
Local Sensor Accuracy		–	±1.0	±3.0	°C
Resolution		–	0.25	–	°C
Remote Diode1 Sensor Accuracy	60°C ≤ T_D ≤ 100°C	–	±0.5	±1.0	°C
Remote Diode2 Sensor Accuracy	60°C ≤ T_D ≤ 100°C	–	±0.5	±1.75	°C
Resolution		–	0.125	–	°C
Remote Sensor Source Current	High level Low level	– –	180 11	– –	μA
OPEN-DRAIN DIGITAL OUTPUTS ($\overline{\text{THERM}}$, $\overline{\text{INT}}$, $\overline{\text{FAN\_FAULT}}$, PWM_OUT)					
Output Low Voltage, V_{OL}	I_{OUT} = –6.0 mA; V_{CC} = 3.0 V	–	–	0.4	V
High-Level Output Leakage Current, I_{OH}	$V_{OUT} = V_{CC}$; V_{CC} = 3.0 V	–	0.1	1.0	μA

Table 4. ELECTRICAL CHARACTERISTICS ($T_A = T_{MIN}$ to T_{MAX}, $V_{CC} = V_{MIN}$ to V_{MAX}, unless otherwise noted.) (Note 1)

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
OPEN-DRAIN SERIAL DATA BUS OUTPUT (SDA)					
Output Low Voltage, V_{OL}	I_{OUT} = –6.0 mA; V_{CC} = 3.0 V	–	–	0.4	V
High-Level Output Leakage Current, I_{OH}	$V_{OUT} = V_{CC}$	–	0.1	1.0	μA
SERIAL BUS DIGITAL INPUTS (SCL, SDA)					
Input High Voltage, V_{IH}		2.1	–	–	V
Input Low Voltage, V_{IL}		–	–	0.8	V
Hysteresis		–	500	–	mV
DIGITAL INPUT LOGIC LEVELS (ADD, $\overline{\text{THERM}}$, TACH1/2) (Note 2)					
Input High Voltage, V_{IH}		2.1	–	–	V
Input Low Voltage, V_{IL}		–	–	0.8	V
DIGITAL INPUT LEAKAGE CURRENT					
Input High Current, I_{IH}	$V_{IN} = V_{CC}$	–1.0	–	–	μA
Input Low Current, I_{IL}	V_{IN} = 0	–	–	1.0	μA
Input Capacitance, C_{IN}		–	5.0	–	pF
FAN RPM-TO-DIGITAL CONVERTER					
Accuracy	60°C ≤ T_A ≤ 100°C	–	–	±6.0	%
Full-Scale Count		–	–	255	
TACH Nominal Input RPM	Divisor N = 1, Fan Count = 153 Divisor N = 2, Fan Count = 153 Divisor N = 4, Fan Count = 153 Divisor N = 8, Fan Count = 153	– – – –	4400 2200 1100 550	– – – –	RPM
Conversion Cycle Time		–	637	–	ms
SERIAL BUS TIMING (Note 3)					
Clock Frequency, f_{SCLK}	See Figure 2 for All Parameters	10	–	100	kHz
Glitch Immunity, t_{SW}		–	50	–	ns
Bus Free Time, t_{BUF}		4.7	–	–	μs
Start Setup Time, $t_{SU;STA}$		4.7	–	–	μs
Start Hold Time, $t_{HD;STA}$		4.0	–	–	μs
Stop Condition Setup Time, $t_{SU;STO}$		4.0	–	–	μs
SCL Low Time, t_{LOW}		1.3	–	–	μs
SCL High Time, t_{HIGH}		4.0	–	50	μs
SCL, SDA Rise Time, t_R		–	–	1000	ns
SCL, SDA Fall Time, t_F		–	–	300	ns
Data Setup Time, $t_{SU;DAT}$		250	–	–	ns
Data Hold Time, $t_{HD;DAT}$		300	–	–	ns

1. Typicals are at T_A = 25°C and represent most likely parametric norm. Shutdown current typ is measured with V_{CC} = 3.3 V.
2. ADD is a three-state input that can be pulled high, low, or left open-circuit.
3. Timing specifications are tested at logic levels of V_{IL} = 0.8 V for a falling edge and V_{IH} = 2.2 V for a rising edge.

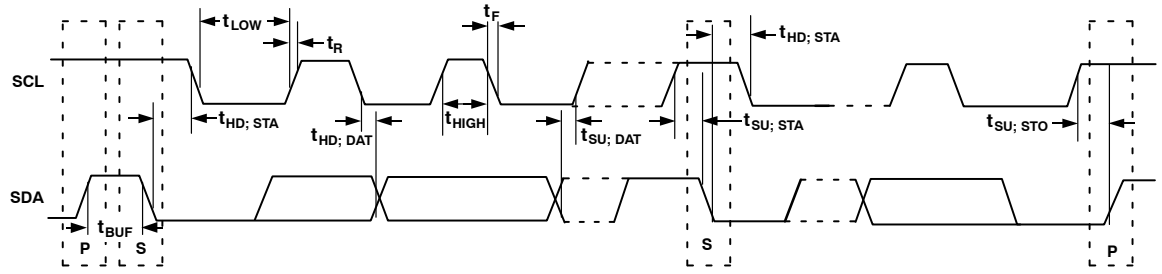


Figure 2. Serial Bus Timing Diagram

TYPICAL PERFORMANCE CHARACTERISTICS

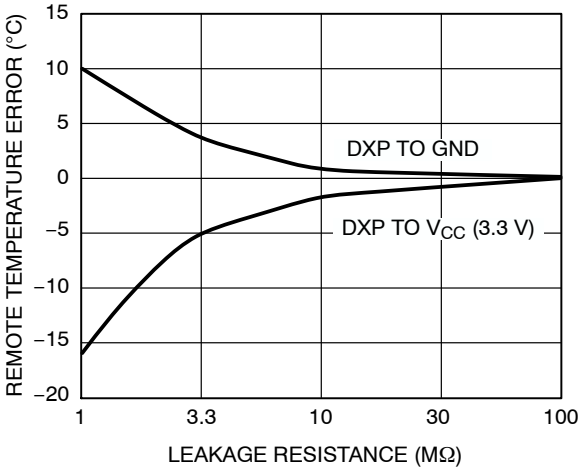


Figure 3. Temperature Error vs. PCB Track Resistance

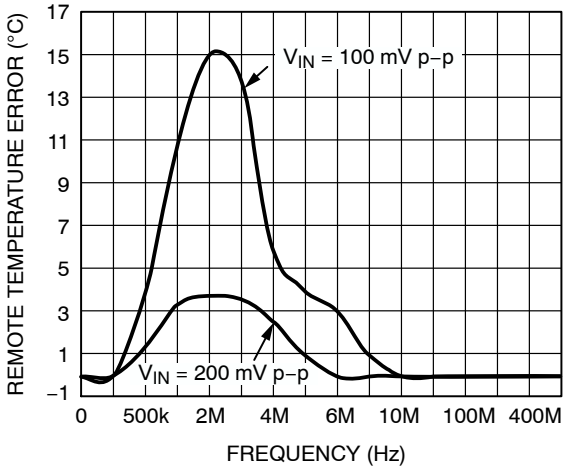


Figure 4. Temperature Error vs. Power Supply Noise Frequency

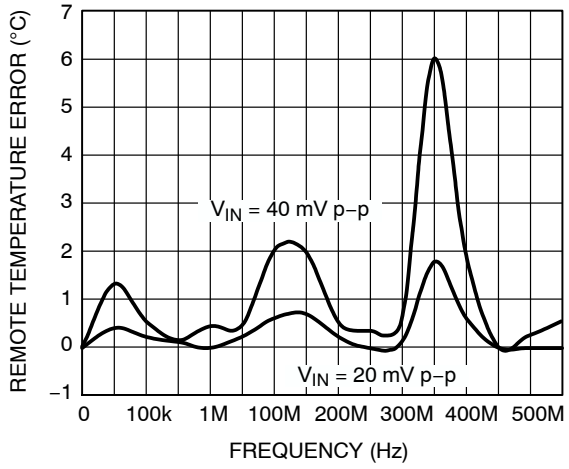


Figure 5. Temperature Error vs. Common-mode Noise Frequency

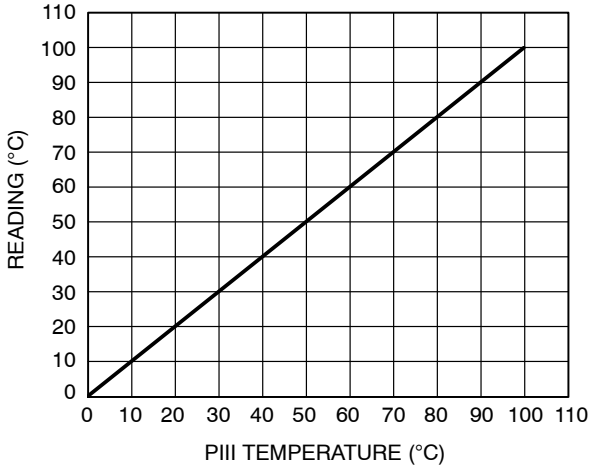


Figure 6. Pentium® III Temperature Measurement vs. ADM1031 Reading

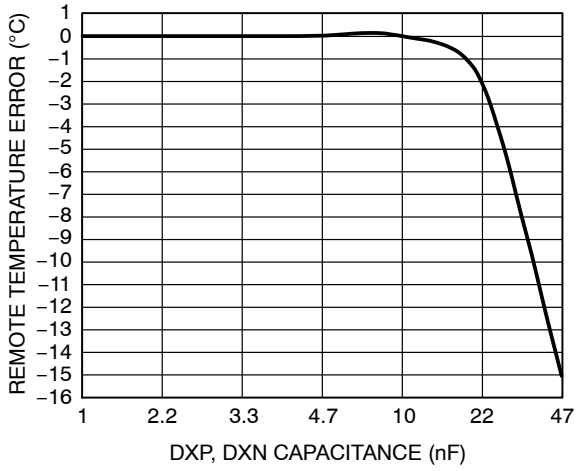


Figure 7. Temperature Error vs. Capacitance between D+ and D–

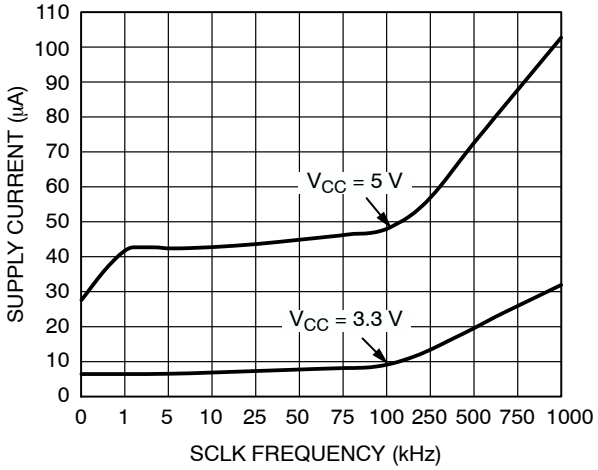


Figure 8. Standby Current vs. Clock Frequency

TYPICAL PERFORMANCE CHARACTERISTICS (Cont'd)

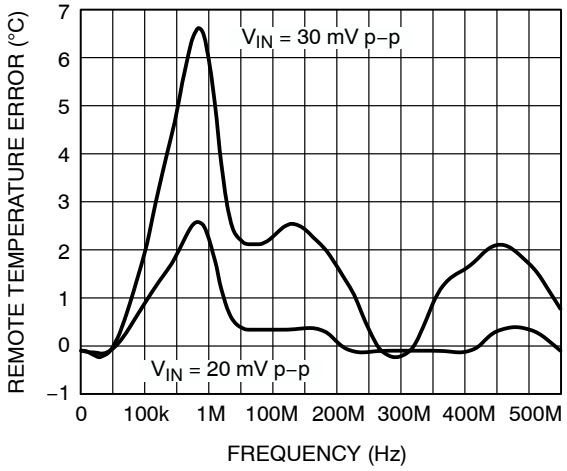


Figure 9. Temperature Error vs. Differential-mode Noise Frequency

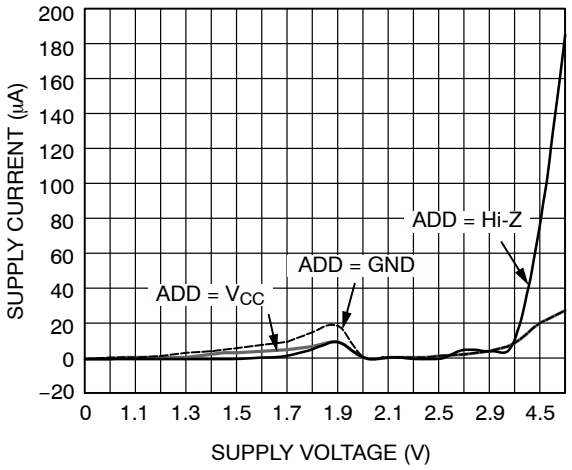


Figure 10. Standby Supply Current vs. Supply Voltage

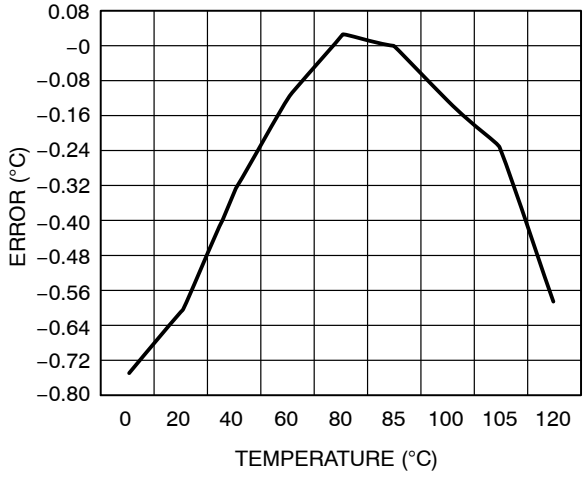


Figure 11. Local Sensor Temperature Error

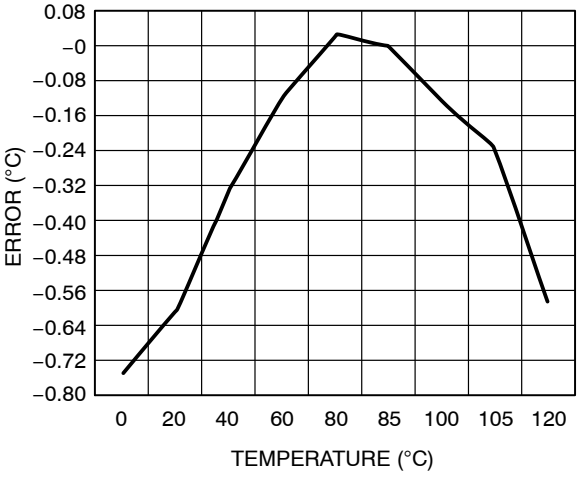


Figure 12. Remote Temperature Sensor Error

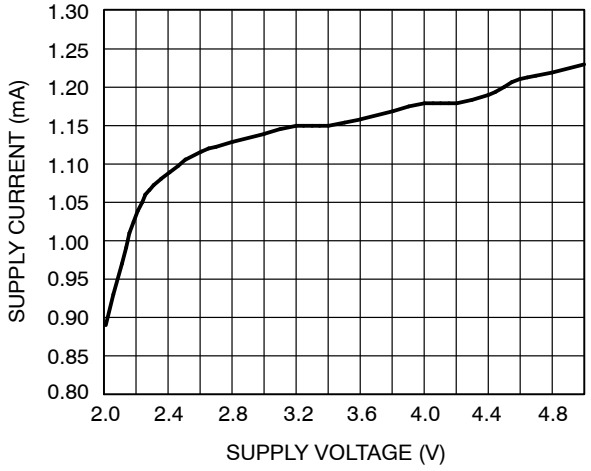


Figure 13. Supply Current vs. Supply Voltage

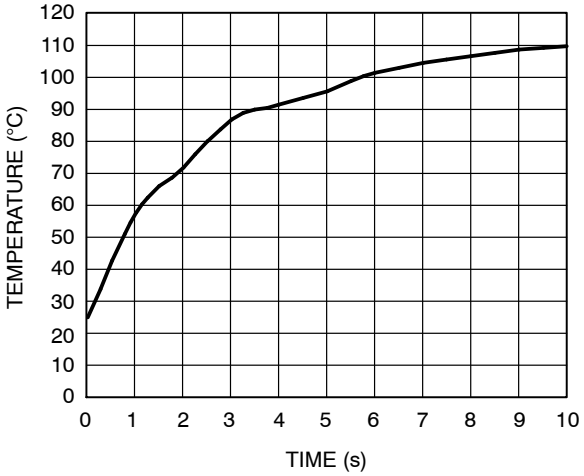


Figure 14. Response to Thermal Shock

Functional Description

The ADM1031 is a temperature monitor and dual PWM fan controller for microprocessor-based systems. The device communicates with the system via a serial System Management Bus (SMBus). The serial bus controller has a hardwired address pin for device selection (Pin 13), a serial data line for reading and writing addresses and data (Pin 15), and an input line for the serial clock (Pin 16). All control and programming functions of the ADM1031 are performed over the serial bus. The device also supports Alert Response Address (ARA).

Internal Registers

Brief descriptions of the ADM1031's principal internal registers are given below. For more detailed information on the function of each register, see Table 18 through Table 33.

Configuration Register

This register controls and configures various functions on the device.

Address Pointer Register

This register contains the address that selects one of the other internal registers. When writing to the ADM1031, the first byte of data is always a register address, which is written to the address pointer register.

Status Registers

These registers provide status of each limit comparison.

Value and Limit Registers

Theses registers store the results of temperature and fan speed measurements, along with their limit values.

Fan Speed Configuration Register

This register is used to program the PWM duty cycle for each fan.

Offset Registers

These registers allow the temperature channel readings to be offset by a 5-bit twos complement value written to these registers. These values are automatically added to the temperature values (or subtracted from if negative). This allows the systems designer to optimize the system if required, by adding or subtracting up to 15°C from a temperature reading.

Fan Characteristics Registers

These registers are used to select the spin-up time, PWM frequency, and speed range for the fans used.

$\overline{\text{THERM}}$ Limit Registers

These registers contain the temperature values at which $\overline{\text{THERM}}$ is asserted.

T_{MIN}/T_{RANGE} Registers

These registers are read/write registers that hold the minimum temperature value below which the fan does not run when the device is in automatic fan speed control mode. These registers also hold the temperature range value that defines the range over which auto fan control is provided, and hence determines the temperature at which the fan is run at full speed.

Serial Bus Interface

Control of the ADM1031 is carried out via the SMBus. The ADM1031 is connected to this bus as a slave device, under the control of a master device, for example, the 810 chipset.

The ADM1031 has a 7-bit serial bus address. When the device is powered up, it does so with a default serial bus address. The five MSBs of the address are set to 01011; the two LSBs are determined by the logical state of Pin 13 (ADD). This is a three-state input that can be grounded, connected to V_{CC}, or left open-circuit to give three different addresses. The state of the ADD pin is only sampled at powerup, so changing ADD with power on has no effect until the device is powered off, then on again.

Table 5. ADD PIN TRUTH TABLE

ADD Pin	A1	A0
GND	0	0
No Connect	1	0
V_{CC}	0	1

If ADD is left open-circuit, then the default address is 0101110. The facility to make hardwired changes at the ADD pin allows the user to avoid conflicts with other devices sharing the same serial bus; for example, if more than one ADM1031 is used in a system.

Serial Bus Protocol

1. The master initiates data transfer by establishing a START condition, defined as a high-to-low transition on the serial data line SDA while the serial clock line SCL remains high. This indicates that an address/data stream follows. All slave peripherals connected to the serial bus respond to the START condition, and shift in the next eight bits, consisting of a 7-bit address (MSB first) plus an R/$\overline{\text{W}}$ bit that determines the direction of the data transfer, that is, whether data is written to or read from the slave device.
   The peripheral whose address corresponds to the transmitted address responds by pulling the data line low during the low period before the ninth clock pulse, known as the Acknowledge Bit. All other devices on the bus now remain idle while the selected device waits for data to be read from or written to it. If the R/$\overline{\text{W}}$ bit is a 0, then the master writes to the slave device. If the R/$\overline{\text{W}}$ bit is a 1, then the master reads from the slave device.
2. Data is sent over the serial bus in sequences of nine clock pulses, eight bits of data, followed by an acknowledge bit from the slave device. Transitions on the data line must occur during the low period of the clock signal and remain stable

during the high period, as a low-to-high transition when the clock is high can be interpreted as a stop signal. The number of data bytes that can be transmitted over the serial bus in a single read or write operation is limited only by what the master and slave devices can handle.

3. When all data bytes have been read or written, stop conditions are established. In write mode, the master pulls the data line high during the tenth clock pulse to assert a stop condition. In read mode, the master device overrides the acknowledge bit by pulling the data line high during the low period before the ninth clock pulse. This is known as No Acknowledge. The master then takes the data line low during the low period before the tenth clock pulse, then high during the tenth clock pulse to assert a stop condition.

Any number of bytes of data can be transferred over the serial bus in one operation, but it is not possible to mix read and write in one operation, because the type of operation is determined at the beginning and cannot subsequently be changed without starting a new operation.

In the case of the ADM1031, write operations contain either one byte or two bytes, and read operations contain one byte, and perform the functions described next.

Writing Data to a Register

To write data to one of the device data registers or read data from it, the address pointer register must be set so that the correct data register is addressed; data can then be written to that register or read from it. The first byte of a write operation always contains an address that is stored in the address pointer register. If data is to be written to the device, the write operation contains a second data byte that is written to the register selected by the address pointer register.

This is illustrated in Figure 15. The device address is sent over the bus followed by R/$\overline{W}$ set to 0. This is followed by two data bytes. The first data byte is the address of the internal data register to be written to, which is stored in the address pointer register. The second data byte is the data to be written to the internal data register.

Reading Data from a Register

When reading data from a register there are two possibilities:

1. If the ADM1031's address pointer register value is unknown or not the desired value, it is first necessary to set it to the correct value before data can be read from the desired data register. This is done by performing a write to the ADM1031 as before, but only the data byte containing the register address is sent, as data is not to be written to the register. This is shown in Figure 16.
   A read operation is then performed consisting of the serial bus address, R/$\overline{W}$ bit set to 1, followed by the data byte read from the data register. This is shown in Figure 17.
2. If the address pointer register is known to be already at the desired address, data can be read from the corresponding data register without first writing to the address pointer register, so Figure 16 can be omitted.

NOTES:

1. Although it is possible to read a data byte from a data register without first writing to the address pointer register, if the address pointer register is already at the correct value, it is not possible to write data to a register without writing to the address pointer register. This is because the first data byte of a write is always written to the address pointer register.
2. In Figure 15, Figure 16, and Figure 17, the serial bus address is shown as the default value 01011(A1)(A0), where A1 and A0 are set by the three-state ADD pin.
3. The ADM1031 also supports the Read Byte protocol, as described in the system management bus specification.

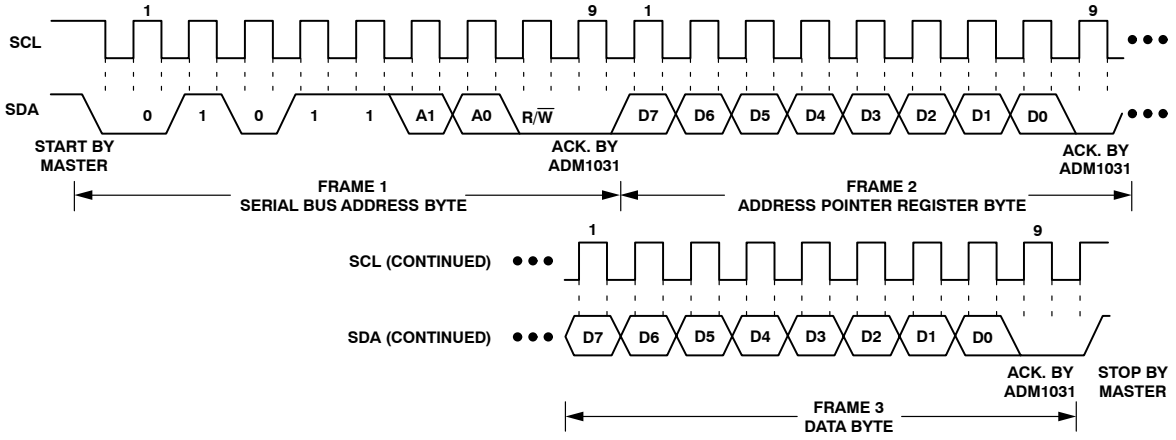


Figure 15. Writing a Register Address to the Address Pointer Register, then Writing Data to the Selected Register

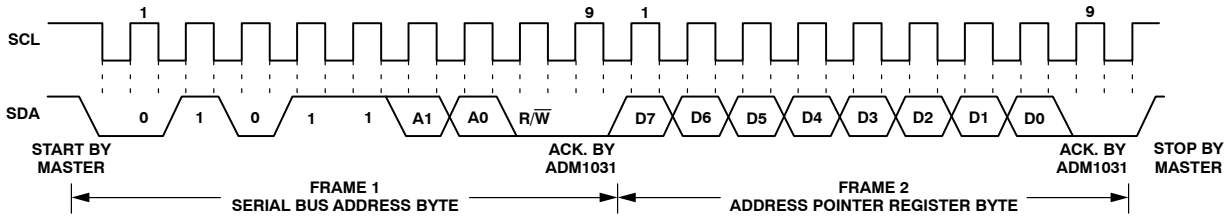


Figure 16. Writing to the Address Pointer Register Only

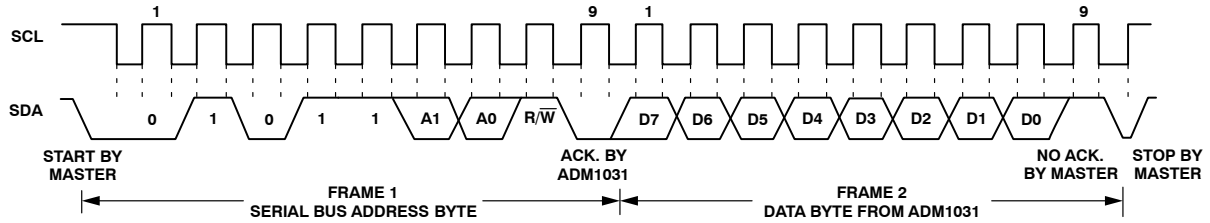


Figure 17. Reading Data from a Previously Selected Register

Alert Response Address

Alert Response Address (ARA) is a feature of SMBus devices that allows an interrupting device to identify itself to the host when multiple devices exist on the same bus.

The $\overline{\text{INT}}$ output can be used as an interrupt output or can be used as an $\overline{\text{SMBALERT}}$. One or more $\overline{\text{INT}}$ outputs can be connected to a common $\overline{\text{SMBALERT}}$ line connected to the master. If a device's $\overline{\text{INT}}$ line goes low, the following procedure occurs:

1. $\overline{\text{SMBALERT}}$ is pulled low.
2. Master initiates a read operation and sends the Alert Response Address (ARA = 0001 100). This is a general call address that must not be used as a specific device address.
3. The device whose $\overline{\text{INT}}$ output is low responds to the alert response address, and the master reads its device address. The address of the device is now known and can be interrogated in the usual way.
4. If more than one device's $\overline{\text{INT}}$ output is low, the one with the lowest device address has priority, in accordance with normal SMBus arbitration.
5. Once the ADM1031 has responded to the alert response address, it resets its $\overline{\text{INT}}$ output. However, if the error condition that caused the interrupt persists, then $\overline{\text{INT}}$ is reasserted on the next monitoring cycle.

Temperature Measurement System

Internal Measurement

The ADM1031 contains an on-chip bandgap temperature sensor. The on-chip ADC performs conversions on the output of this sensor and outputs the temperature data in 10-bit twos complement format. The resolution of the local temperature sensor is 0.25°C. The format of the temperature data is shown in Table 6.

External Measurement

The ADM1031 can measure the temperatures of two external diode sensors or diode-connected transistors, connected to Pins 9 and 10, and Pins 11 and 12.

These pins are dedicated temperature input channels. The function of Pin 7 is as a $\overline{\text{THERM}}$ input/output and is used to flag overtemperature conditions.

The forward voltage of a diode or diode-connected transistor, operated at a constant current, exhibits a negative temperature coefficient of about –2 mV/°C. Unfortunately, the absolute value of V_{BE}, varies from device to device, and individual calibration is required to null this out. As a result, the technique is unsuitable for mass production.

The technique used in the ADM1031 is to measure the change in V_{BE} when the device is operated at two different currents.

This is given by:

$$\Delta V_{BE} = KT/q \times \ln(N) \quad \text{(eq. 1)}$$

where:

K is Boltzmann's constant
q is charge on the carrier
T is absolute temperature in Kelvins
N is ratio of the two currents

Figure 18 shows the input signal conditioning used to measure the output of an external temperature sensor. This figure shows the external sensor as a substrate transistor, provided for temperature monitoring on some microprocessors, but it could equally well be a discrete transistor.

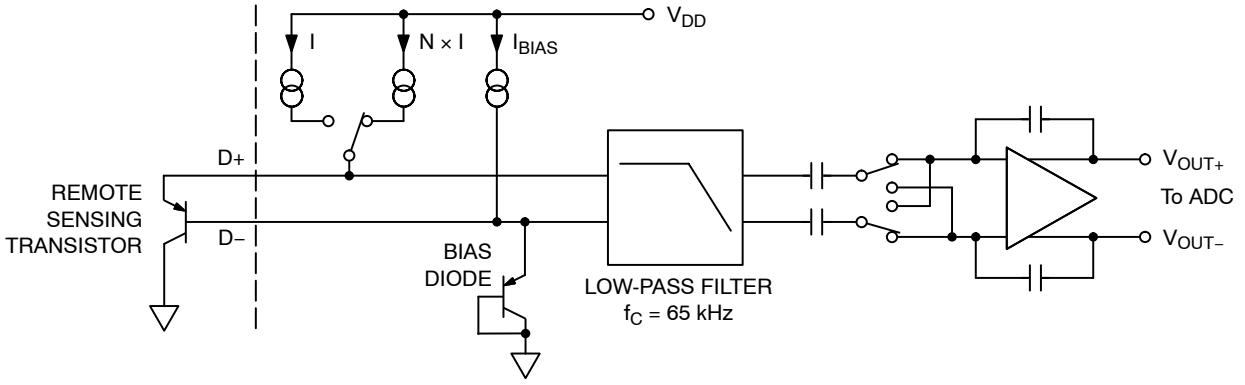


Figure 18. Signal Conditioning

If a discrete transistor is used, then the collector is not grounded, and is linked to the base. If a PNP transistor is used, the base is connected to the D– input and the emitter to the D+ input. If an NPN transistor is used, the emitter is connected to the D– input and the base to the D+ input.

One LSB of the ADC corresponds to 0.125°C, so the ADM1031 can theoretically measure temperatures from –127°C to +127.75°C, although –127°C is outside the operating range for the device. The extended temperature resolution data format is shown in Table 7 and Table 8.

Table 6. TEMPERATURE DATA FORMAT – (LOCAL TEMPERATURE AND REMOTE TEMPERATURE HIGH BYTES)

Temperature (°C)	Digital Output
–128°C	1000 0000
–125°C	1000 0011
–100°C	1001 1100
–75°C	1011 0101
–50°C	1100 1110
–25°C	1110 0111
–1°C	1111 1111
0°C	0000 0000
+1°C	0000 0001
+10°C	0000 1010
+25°C	0001 1001
+50°C	0011 0010
+75°C	0100 1011
+100°C	0110 0100
+125°C	0111 1101
+127°C	0111 1111

Table 7. REMOTE SENSOR EXTENDED TEMPERATURE RESOLUTION

Extended Resolution	Remote Temperature Low Bits
0.000°C	000
0.125°C	001
0.250°C	010
0.375°C	011
0.500°C	100
0.625°C	101
0.750°C	110
0.875°C	111

The extended temperature resolution for the local and remote channels is stored in the extended temperature resolution register (Register 0×06), and is outlined in Table 22.

Table 8. LOCAL SENSOR EXTENDED TEMPERATURE RESOLUTION

Extended Resolution	Local Temperature Low Bits
0.00°C	00
0.25°C	01
0.50°C	10
0.75°C	11

To prevent ground noise interfering with the measurement, the more negative terminal of the sensor is not referenced to ground, but biased above ground by an internal diode at the D– input. If the sensor is used in a very noisy

environment, a capacitor of value up to 1000 pF can be placed between the D+ and D– inputs to filter the noise.

To measure ΔV_{BE}, the sensor is switched between operating currents of I and N × I. The resulting waveform is passed through a 65 kHz low-pass filter to remove noise, then to a chopper-stabilized amplifier that performs the functions of amplification and rectification of the waveform to produce a dc voltage proportional to ΔV_{BE}. This voltage is measured by the ADC to give a temperature output in 11-bit twos complement format. To further reduce the effects of noise, digital filtering is performed by averaging the results of 16 measurement cycles. An external temperature measurement nominally takes 9.6 ms.

Layout Considerations

Digital boards can be electrically noisy environments and care must be taken to protect the analog inputs from noise, particularly when measuring the very small voltages from a remote diode sensor. The following precautions should be taken:

1. Place the ADM1031 as close as possible to the remote sensing diode. Provided that the worst noise sources such as clock generators, data/address buses, and CRTs are avoided, this distance can be 4 to 8 inches.
2. Route the D+ and D– tracks close together, in parallel, with grounded guard tracks on each side. Provide a ground plane under the tracks if possible.
3. Use wide tracks to minimize inductance and reduce noise pickup. Ten mil track minimum width and spacing is recommended.

Figure 19. Arrangement of Signal Tracks

4. Try to minimize the number of copper/solder joints, which can cause thermocouple effects. Where copper/solder joints are used, make sure that they are in both the D+ and D– path and at the same temperature.
   Thermocouple effects should not be a major problem as 1°C corresponds to about 200 μV, and thermocouple voltages are about 3 μV/°C of temperature difference. Unless there are two thermocouples with a big temperature differential between them, thermocouple voltages should be much less than 200 μV.
5. Place a 0.1 μF bypass capacitor close to the ADM1031.
6. If the distance to the remote sensor is more than 8 inches, the use of twisted pair cable is recommended. This works up to about 6 to 12 feet.
7. For extra long distances (up to 100 feet), use a shielded twisted pair cable, such as the Belden #8451 microphone cable. Connect the twisted pair to D+ and D– and the shield to GND close to the ADM1031. Leave the remote end of the shield unconnected to avoid ground loops.

Because the measurement technique uses switched current sources, excessive cable and/or filter capacitance can affect the measurement. When using long cables, the filter capacitor C1 can be reduced or removed. In any case the total shunt capacitance should not exceed 1000 pF.

Cable resistance can also introduce errors. One ohm series resistance introduces about 0.5°C error.

Addressing the Device

ADD (Pin 13) is a three-state input. It is sampled, on powerup to set the lowest two bits of the serial bus address. Up to three addresses are available to the systems designer via this address pin. This reduces the likelihood of conflicts with other devices attached to the system management bus.

The Interrupt System

The ADM1031 has two interrupt outputs, $\overline{\text{INT}}$ and $\overline{\text{THERM}}$. These have different functions. $\overline{\text{INT}}$ responds to violations of software programmed temperature limits and is maskable.

$\overline{\text{THERM}}$ is intended as a “fail-safe” interrupt output that cannot be masked. If the temperature is below the low temperature limit, the $\overline{\text{INT}}$ pin is asserted low to indicate an out-of-limit condition. If the temperature exceeds the high temperature limit, the $\overline{\text{INT}}$ pin is also asserted low. A third limit, $\overline{\text{THERM}}$ limit, can be programmed into the device to set the temperature limit above which the overtemperature $\overline{\text{THERM}}$ pin is asserted low. The behavior of the high limit and $\overline{\text{THERM}}$ limit is as follows:

1. Whenever the temperature measured exceeds the high temperature limit, the $\overline{\text{INT}}$ pin is asserted low.
2. If the temperature exceeds the $\overline{\text{THERM}}$ limit, the $\overline{\text{THERM}}$ output asserts low. This can be used to throttle the CPU clock. If the $\overline{\text{THERM}}$-to-Fan Enable bit (Bit 7 of $\overline{\text{THERM}}$ behavior/revision register) is cleared to 0, then the fans do not run full-speed. The $\overline{\text{THERM}}$ limit can be programmed at a lower temperature than the high temperature limit. This allows the system to run in silent mode, where the CPU can be throttled while the cooling fan is off. If the temperature continues to increase, and exceeds the high temperature limit, an $\overline{\text{INT}}$ is generated. Software can then decide whether the fan should run to cool the CPU. This allows the system to run in silent mode.
3. If the $\overline{\text{THERM}}$-to-Fan Enable bit is set to 1, then the fan runs full-speed whenever $\overline{\text{THERM}}$ is

asserted low. In this case, both throttling and active cooling take place. If the high temperature limit is programmed to a lower value than the $\overline{\text{THERM}}$ limit, exceeding the high temperature limit asserts $\overline{\text{INT}}$ low. Software could change the speed of the fan depending on temperature readings. If the temperature continues to increase and exceeds the $\overline{\text{THERM}}$ limit, $\overline{\text{THERM}}$ asserts low to throttle the CPU and the fan runs full-speed. This allows the system to run in performance mode, where active cooling takes place and the CPU is only throttled at high temperature.

Using the high temperature limit and the $\overline{\text{THERM}}$ limit in this way allows the user to gain maximum performance from the system by only slowing it down, should it be at a critical temperature.

Although the ADM1031 does not have a dedicated interrupt mask register, clearing the appropriate enable bits in Configuration Register 2 clears the appropriate interrupts and masks out future interrupts on that channel. Disabling interrupt bits prevents out-of-limit conditions from generating an interrupt or setting a bit in the status registers.

Using $\overline{\text{THERM}}$ as an Input

The $\overline{\text{THERM}}$ pin is an open-drain input/output pin. When used as an output, it signals overtemperature conditions. When asserted low as an output, the fan is driven full-speed if the $\overline{\text{THERM}}$-to-Fan Enable bit is set to 1 (Bit 7 of Register 0×3F). When $\overline{\text{THERM}}$ is pulled low as an input, the $\overline{\text{THERM}}$ bit (Bit 7) of Status Register 2 is set to 1, and the fans are driven full-speed. Note that the $\overline{\text{THERM}}$-to-Fan Enable bit has no effect whenever $\overline{\text{THERM}}$ is used as an input. If $\overline{\text{THERM}}$ is pulled low as an input, and the $\overline{\text{THERM}}$-to-Fan Enable bit = 0, then the fans are still driven full-speed. The $\overline{\text{THERM}}$-to-Fan Enable bit only affects the behavior of $\overline{\text{THERM}}$ when used as an output.

Status Registers

All out-of-limit conditions are flagged by status bits in Status Register 1 (0×02) and Status Register 2 (0×03). Bit 0 (Alarm Speed) and Bit 1 (Fan Fault) of Status Register 1, once set, can be cleared by reading Status Register 1. Once the alarm speed bit is cleared, this bit is not reasserted on the next monitoring cycle even if the condition still persists. This bit can be reasserted only if the fan is no longer at alarm speed. Bit 1 (Fan Fault) is set whenever a fan tach failure is detected. Once cleared, it reasserts on subsequent fan tach failures.

Bit 2 and Bit 3 of Status Register 1 and Status Register 2 are the Remote 1 and Remote 2 Temperature High and Low status bits. Exceeding the high or low temperature limits for the external channel sets these status bits. Reading the status register clears these bits. However, these bits are reasserted if the out-of-limit condition still exists on the next monitoring cycle. Bit 6 and Bit 7 are the Local Temperature High and Low status bits. These behave exactly the same as the Remote Temperature High and Low status bits. Bit 4 of Status Register 1 indicates that the Remote Temperature $\overline{\text{THERM}}$ limit has been exceeded. This bit gets cleared on a read of Status Register 1 (see Figure 20). Bit 5 indicates a remote diode error. This bit is a 1 if a short or open is detected on the remote temperature channel on powerup. If this bit is set to 1 on powerup, it cannot be cleared. Bit 6 of Status Register 2 (0×03) indicates that the Local $\overline{\text{THERM}}$ limit has been exceeded. This bit is cleared on a read of Status Register 2. Bit 7 indicates that $\overline{\text{THERM}}$ has been pulled low as an input. This bit can also be cleared on a read of Status Register 2.

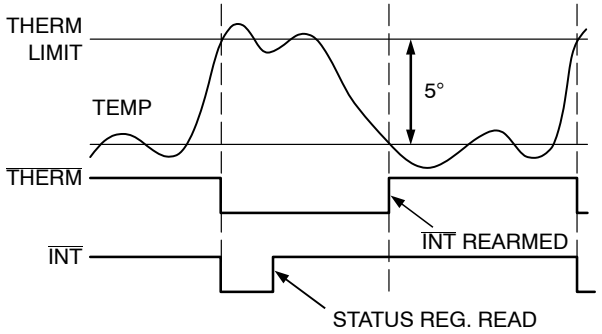


Figure 20. Operation of $\overline{\text{THERM}}$ and $\overline{\text{INT}}$ Signals

Figure 20 shows the interaction between $\overline{\text{INT}}$ and $\overline{\text{THERM}}$. Once a critical temperature $\overline{\text{THERM}}$ limit is exceeded, both $\overline{\text{INT}}$ and $\overline{\text{THERM}}$ assert low. Reading the status registers clears the interrupt and the $\overline{\text{INT}}$ pin goes high. However, the $\overline{\text{THERM}}$ pin remains asserted until the measured temperature falls 5°C below the exceeded $\overline{\text{THERM}}$ limit. This feature can be used to CPU throttle or drive a fan full speed for maximum cooling. Note that the $\overline{\text{INT}}$ pin for that interrupt source is not rearmed until the temperature has fallen below the $\overline{\text{THERM}}$ limit –5°C. This prevents unnecessary interrupts from tying up valuable CPU resources.

Fan Control Modes of Operation

The ADM1031 has four different modes of operation. These modes determine the behavior of the system.

1. Automatic Fan Speed Control Mode.
2. Filtered Automatic Fan Speed Control Mode.
3. PWM Duty Cycle Select Mode (Directly Sets Fan Speed Under Software Control).
4. RPM Feedback Mode.

Automatic Fan Speed Control

The ADM1031 has a local temperature channel and two remote temperature channels, which can be connected to an on-chip diode-connected transistor on a CPU. These three temperature channels can be used as the basis for an automatic fan speed control loop to drive fans using pulse width modulation (PWM).

How Does the Control Loop Work?

The automatic fan speed control loop is shown in Figure 21.

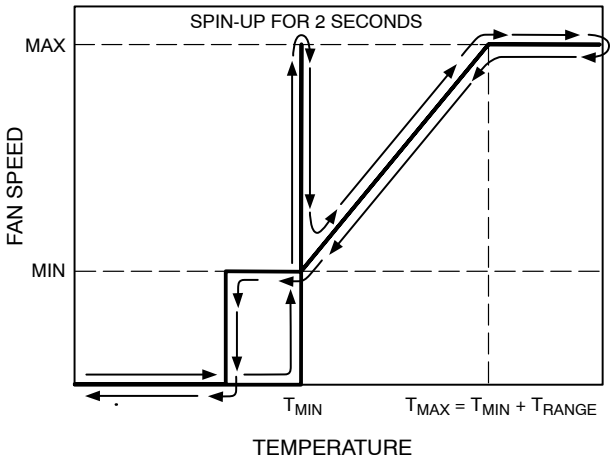


Figure 21. Automatic Fan Speed Control Loop

T_{MIN} is the temperature at which the fan should switch on and run at minimum speed. The fan only turns on once the temperature being measured rises above the T_{MIN} value programmed. The fan spins up for a predetermined time (default = 2 seconds). See the Fan Spin-Up section for more details.

T_{RANGE} is the temperature range over which the ADM1031 automatically adjusts the fan speed. As the temperature increases beyond T_{MIN}, the PWM_OUT duty cycle increases accordingly. The T_{RANGE} parameter actually defines the fan speed vs. temperature slope of the control loop.

T_{MAX} is the temperature at which the fan is at its maximum speed. At this temperature, the PWM duty cycle driving the fan is 100%. T_{MAX} is given by $T_{MIN} + T_{RANGE}$. Since this parameter is the sum of the T_{MIN} and T_{RANGE} parameters, it does not need to be programmed into a register on-chip.

A hysteresis value of 5°C is included in the control loop to prevent the fan continuously switching on and off if the temperature is close to T_{MIN}. The fan continues to run until the temperature drops 5°C below T_{MIN}.

Figure 22 shows the different control slopes determined by the T_{RANGE} value chosen, and programmed into the ADM1031. T_{MIN} is set to 0°C to start all slopes from the same point. The figure shows how changing the T_{RANGE} value affects the PWM duty cycle vs. temperature slope.

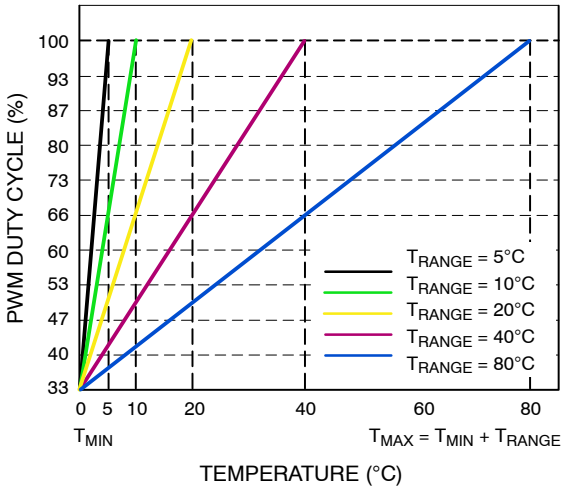


Figure 22. PWM Duty Cycle vs. Temperature Slope (T_{RANGE})

Figure 23 shows how, for a given T_{RANGE}, changing the T_{MIN} value affects the loop. Increasing the T_{MIN} value increases the T_{MAX} (temperature at which the fan runs full speed) value, since $T_{MAX} = T_{MIN} + T_{RANGE}$. Note, however, that the PWM duty cycle vs. temperature slope remains exactly the same. Changing the T_{MIN} value merely shifts the control slope. The T_{MIN} can be changed in increments of 4°C.

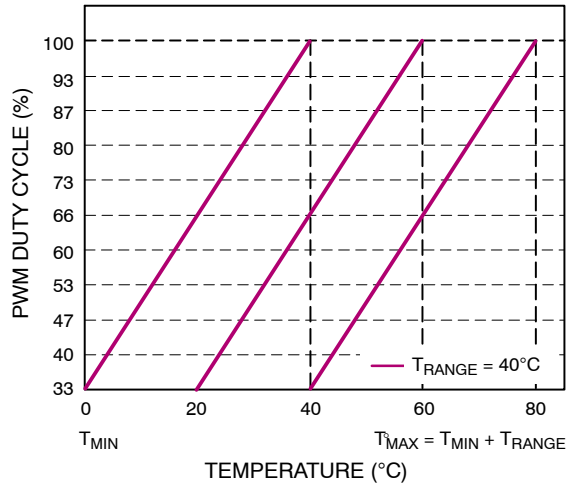


Figure 23. Effect of Increasing T_{MIN} Value on Control Loop

Fan Spin-Up

As mentioned in the How Does the Control Loop Work? section, once the temperature being measured exceeds the T_{MIN} value programmed, the fan turns on at minimum speed (default = 33% duty cycle). However, the problem with fans being driven by PWM is that 33% duty cycle is not enough to reliably start the fan spinning. The solution is to spin the fan up for a predetermined time, and once the fan has spun up, its running speed can be reduced in line with the temperature being measured.

The ADM1031 allows fan spin-up times between 200 ms and 8 seconds. Bits <2:0> of Fan Characteristics Register 1 (Register 0×20) and Fan Characteristic Register 2 (Register 0×21) program the fan spin-up times.

Table 9. FAN SPIN-UP TIMES

Bits 2:0	Spin-Up Time (Fan Characteristics Registers 1, 2)
000	200 ms
001	400 ms
010	600 ms
011	800 ms
100	1 sec
101	2 sec (Default)
110	4 sec
111	8 sec

Once the automatic fan speed control loop parameters have been chosen, the ADM1031 device can be programmed. The ADM1031 is placed into automatic fan speed control mode by setting Bit 7 of Configuration Register 1 (Register 0×00). The device powers up in automatic fan speed control mode by default. The control mode offers further flexibility in that the user can decide which temperature channel/channels control each fan.

Table 10. AUTO MODE FAN BEHAVIOR

Bits 6, 5	Control Operation (Configuration Register 1)
00	Remote Temperature 1 Controls Fan 1 Remote Temperature 2 Controls Fan 2
01	Remote Temperature 1 Controls Fan 1 and 2
10	Remote Temperature 2 Controls Fan 1 and 2
11	Maximum Speed Calculated by Local and Remote Temperature Channels Controls Fans 1 and 2

When Bit 5 and Bit 6 of Configuration Register 1 are both set to 1, increased flexibility is offered. The local and remote temperature channels can have independently programmed control loops with different control parameters. Whichever control loop calculates the fastest fan speed based on the temperature being measured, drives the fans.

Figures 24 and 25 show how the fan's PWM duty cycle is determined by two independent control loops. This is the type of auto mode fan behavior seen when Bit 5 and Bit 6 of Configuration Register 1 are set to 11. Figure 24 shows the control loop for the local temperature channel. Its T_{MIN} value has been programmed to 20°C, and its T_{RANGE} value is 40°C. The local temperature's T_{MAX} is thus 60°C. Figure 25 shows the control loop for the remote temperature channel. Its T_{MIN} value has been set to 0°C, while its T_{RANGE} = 80°C. Therefore, the remote temperature's T_{MAX} value is 80°C.

Consider if both temperature channels measure 40°C. Both control loops calculate a PWM duty cycle of 66%. Therefore, the fan is driven at 66% duty cycle. If both temperature channels measure 20°C, the local channel calculates 33% PWM duty cycle, while the Remote 1 channel calculates 50% PWM duty cycle. Thus, the fans are driven at 50% PWM duty cycle. Consider the local temperature measuring 60°C while the Remote 1 temperature is measuring 70°C. The PWM duty cycle calculated by the local temperature control loop is 100% (because the temperature = T_{MAX}). The PWM duty cycle calculated by the Remote 1 temperature control loop at 70°C is approximately 90%. Therefore, the fan runs full-speed (100% duty cycle). Remember, that the fan speed is based on the fastest speed calculated, and is not necessarily based on the highest temperature measured. Depending on the control loop parameters programmed, a lower temperature on one channel, can actually calculate a faster speed than a higher temperature on the other channel.

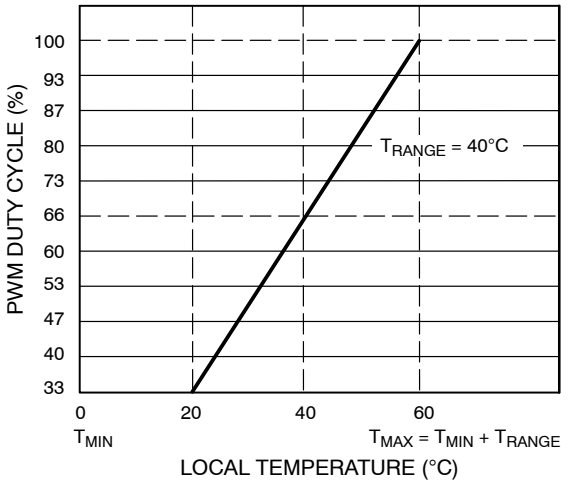


Figure 24. Max Speed Calculated by Local Temperature Control Loop Drives Fan

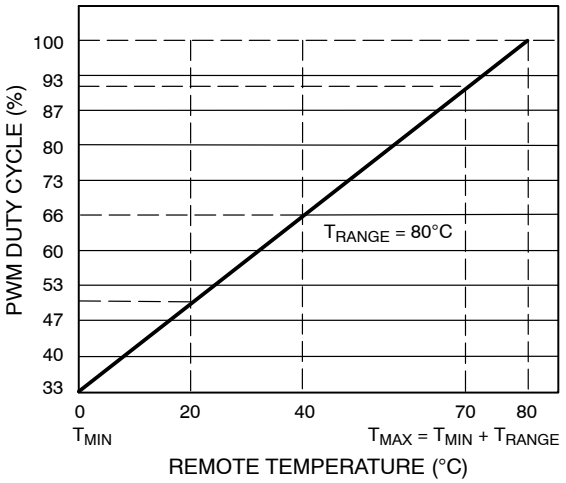


Figure 25. Max Speed Calculated by Remote Temperature Control Loop Drives Fan

Programming the Automatic Fan Speed Control Loop

1. Program a value for T_{MIN}.
2. Program a value for the slope T_{RANGE}.
3. $T_{MAX} = T_{MIN} + T_{RANGE}$.
4. Program a value for fan spin-up time.
5. Program the desired automatic fan speed control mode behavior, that is, which temperature channel controls the fan.
6. Select automatic fan speed control mode by setting Bit 7 of Configuration Register 1.

Other Control Loop Parameters

It should be noted that changing the minimum PWM duty cycle affects the control loop behavior.

Slope 1 of Figure 26 shows T_{MIN} set to 0°C and the T_{RANGE} chosen is 40°C. In this case, the fan's PWM duty cycle varies over the range 33% to 100%. The fan runs full-speed at 40°C. If the minimum PWM duty cycle at which the fan runs at T_{MIN} is changed, its effect can be seen on Slope 2 and Slope 3. Take Case 2, where the minimum PWM duty cycle is reprogrammed from 33% (default) to 53%.

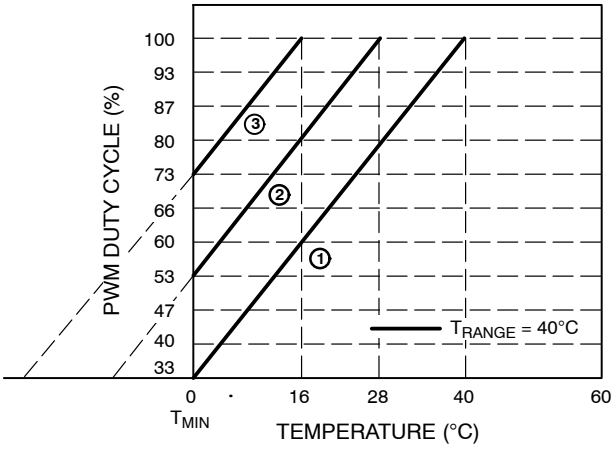


Figure 26. Effect of Changing Minimum Duty Cycle on Control Loop with Fixed T_{MIN} and T_{RANGE} Values

The fan actually reaches full speed at a much lower temperature, 28°C. Case 3 shows that when the minimum PWM duty cycle is increased to 73%, the temperature at which the fan runs full speed is 16°C. Therefore, the effect of increasing the minimum PWM duty cycle, with a fixed T_{MIN} and fixed T_{RANGE}, is that the fan actually reaches full speed (T_{MAX}) at a lower temperature than $T_{MIN} + T_{RANGE}$.

How can T_{MAX} be calculated?

In automatic fan speed control mode, the register that holds the minimum PWM duty cycle at T_{MIN}, is the fan speed configuration register (Register 0×22). Table 11 shows the relationship between the decimal values written to the fan speed configuration register and PWM duty cycle obtained.

Table 11. Programming PWM Duty Cycle

Decimal Value	PWM Duty Cycle
00	0%
01	7%
02	14%
03	20%
04	27%
05	33% (Default)
06	40%
07	47%
08	53%
09	60%
10 (0×0A)	67%
11 (0×0B)	73%
12 (0×0C)	80%
13 (0×0D)	87%
14 (0×0E)	93%
15 (0×0F)	100%

The temperature at which the fan runs full-speed (100% duty cycle) is given by:

$$T_{MAX} = T_{MIN} + ((\text{Max DC} - \text{Min DC}) \times T_{RANGE}/10) \quad \text{(eq. 2)}$$

where:

T_{MAX} = Temperature at which fan runs full-speed
T_{MIN} = Temperature at which fan will turn on
Max DC = Maximum Duty Cycle (100%) = 15 decimal
Min DC = Duty Cycle at T_{MIN}, programmed into Fan Speed Config Register (default = 33% = 5 decimal)
T_{RANGE} = PWM Duty Cycle vs. Temperature Slope

Example 1:
T_{MIN} = 0°C, T_{RANGE} = 40°C
Min DC = 53% = 8 decimal (Table 11)

Calculate T_{MAX}

$$T_{MAX} = T_{MIN} + ((\text{Max DC} - \text{Min DC}) \times T_{RANGE}/10)$$

$$T_{MAX} = 0 + ((100\%\ \text{DC} - 53\%\ \text{DC}) \times 40/10) \quad \text{(eq. 3)}$$

$$T_{MAX} = 0 + ((15 - 8) \times 4) = 28$$

T_{MAX} =28 °C (As seen on Slope 2 of Figure 26)

Example 2:

T_{MIN} = 0°C, T_{RANGE} = 40°C
Min DC = 73% = 11 decimal (Table 11)

Calculate T_{MAX}

$$T_{MAX} = T_{MIN} + ((\text{Max DC} - \text{Min DC}) \times T_{RANGE}/10)$$

$$T_{MAX} = 0 + ((100\%\ \text{DC} - 73\%\ \text{DC}) \times 40/10) \quad \text{(eq. 4)}$$

$$T_{MAX} = 0 + ((15 - 11) \times 4) = 16$$

T_{MAX} =16 °C (As seen on Slope 3 of Figure 26)

Example 3:

T_{MIN} = 0°C, T_{RANGE} = 40°C
Min DC = 33% = 5 decimal (Table 11)

Calculate T_{MAX}

$$T_{MAX} = T_{MIN} + ((\text{Max DC} - \text{Min DC}) \times T_{RANGE}/10)$$

$$T_{MAX} = 0 + ((100\%\ \text{DC} - 33\%\ \text{DC}) \times 40/10) \quad \text{(eq. 5)}$$

$$T_{MAX} = 0 + ((15 - 5) \times 4) = 40$$

T_{MAX} =40 °C (As seen on Slope 1 of Figure 26)

In this case, since the Minimum Duty Cycle is the default 33%, the equation for T_{MAX} reduces to:

$$T_{MAX} = T_{MIN} + ((\text{Max DC} - \text{Min DC}) \times T_{RANGE}/10)$$

$$T_{MAX} = T_{MIN} + ((15 - 5) \times T_{RANGE}/10)$$

$$T_{MAX} = T_{MIN} + (10 \times T_{RANGE}/10) \quad \text{(eq. 6)}$$

$$\mathbf{T_{MAX} = T_{MIN} + T_{RANGE}}$$

Relevant Registers for Automatic Fan Speed Control Mode

Register 0×00 Configuration Register 1

<7> Logic 1 selects automatic fan speed control, Logic 0 selects software control (Default = 1).

<6:5> 01 = Remote Temp 1 controls Fan 1 and Fan 2
10 = Remote Temp 2 controls Fan 1 and Fan 2
11 = Fastest Calculated Speed controls Fan 1 and 2

Register 0×20, 0×21 Fan Characteristics Registers 1, 2

<2:0> Fan X Spin-Up Time.
000 = 200 ms
001 = 400 ms
010 = 600 ms
011 = 800 ms
100 = 1 sec
101 = 2 sec (Default)
110 = 4 sec
111 = 8 sec

<5:3> PWM Frequency Driving the Fan.
000 = 11.7 Hz
001 = 15.6 Hz
010 = 23.4 Hz
011 = 31.25 Hz (Default)
100 = 37.5 Hz
101 = 46.9 Hz
110 = 62.5 Hz
111 = 93.5 Hz

<7:6> Speed Range N; defines the lowest fan speed that can be measured by the device.
00 = 1: Lowest Speed = 2647 RPM
01 = 2: Lowest Speed = 1324 RPM
10 = 4: Lowest Speed = 662 RPM
11 = 8: Lowest Speed = 331 RPM

Register 0×22 Fan Speed Configuration Register

<3:0> Min Speed: This nibble contains the speed at which the fan runs when the temperature is at T_{MIN}. The default is 0×05, meaning that the fan runs at 33% duty cycle when the temperature is at T_{MIN}.

<7:4> Min Speed: Determines the minimum PWM cycle for Fan 2 in automatic fan speed control mode.

Register 0×24 Local Temperature T_{MIN}/T_{RANGE}

<7:3> Local Temperature T_{MIN}. These bits set the temperature at which the fan turns on when under auto fan speed control. T_{MIN} can be programmed in 4°C increments.
00000 = 0°C
00001 = 4°C
00010 = 8°C
00011 = 12°C
|
|
01000 = 32°C (Default)
|
|
11110 = 120°C
11111 = 124°C

<2:0> Local Temperature T_{RANGE}. This nibble sets the temperature range over which automatic fan speed control takes place.
000 = 5°C
001 = 10°C
010 = 20°C
011 = 40°C
100 = 80°C

Register 0×25, 0×26 Remote 1, 2 Temperature T_{MIN}/T_{RANGE}

<7:3> Remote Temperature T_{MIN}. Sets the temperature at which the fan switches on based on Remote X Temperature Readings.
00000 = 0°C
00001 = 4°C
00010 = 8°C
00011 = 12°C
|
|
01100 = 48°C
|
|
11110 = 120°C
11111 = 124°C

<2:0> Remote Temperature T_{RANGE}. This nibble sets the temperature range over which the fan is controlled based on remote temperature readings.
000 = 5°C
001 = 10°C
010 = 20°C
011 = 40°C
100 = 80°C

Filtered Control Mode

The automatic fan speed control loop reacts instantaneously to changes in temperature, that is, the PWM duty cycle responds immediately to temperature change. In certain circumstances, we do not want the PWM output to react instantaneously to temperature changes. If significant variations in temperature are found in a system, the fan speed changes, which could be obvious to someone in close proximity. One way to improve the system's acoustics would be to slow down the loop so that the fan ramps slowly to its newly calculated fan speed. This also ensures that temperature transients are effectively ignored, and the fan's operation is smooth.

There are two means by which to apply filtering to the automatic fan speed control loop. The first method is to ramp the fan speed at a predetermined rate, to its newly calculated value instead of jumping directly to the new fan speed. The second approach involves changing the on-chip ADC sample rate, to change the number of temperature readings taken per second.

The filtered mode on the ADM1031 is invoked by setting Bit 0 of the fan filter register (Register 0×23) for Fan 1 and Bit 1 for Fan 2. Once the fan filter register has been written to, and all other control loop parameters (such as T_{MIN}, T_{RANGE}) have been programmed, the device can be placed into automatic fan speed control mode by setting Bit 7 of Configuration Register 1 (Register 0×00) to 1.

Effect of Ramp Rate on Filtered Mode

Bits <6:5> of the fan filter register determine the ramp rate in filtered mode. The PWM_OUT signal driving the fan has a period, T, given by the PWM_OUT drive frequency, f, since T = 1/f. For a given PWM period, T, the PWM period is subdivided in to 240 equal time slots. One time slot corresponds to the smallest possible increment in PWM duty cycle. A PWM signal of 33% duty cycle is thus high for 1/3 × 240 time slots and low for 2/3 × 240 time slots. Therefore, 33% PWM duty cycle corresponds to a signal that is high for 80 time slots and low for 160 time slots.

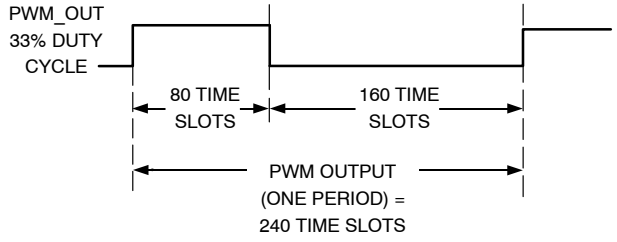


Figure 27. 33% PWM Duty Cycle Represented in Time Slots

The ramp rates in filtered mode are selectable between 1, 2, 4, and 8. The ramp rates are actually discrete time slots. For example, if the ramp rate = 8, then eight time slots are added to the PWM_OUT high duty cycle each time the PWM_OUT duty cycle needs to be increased. Figure 28 shows how the filtered mode algorithm operates.

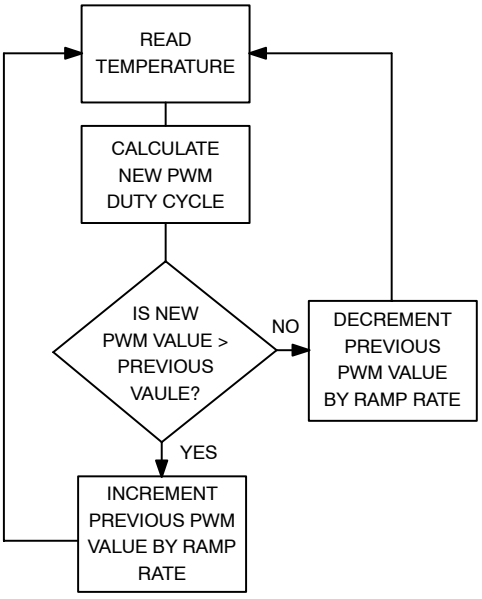


Figure 28. Filtered Mode Algorithm

The filtered mode algorithm calculates a new PWM duty cycle based on the temperature measured. If the new PWM duty cycle value is greater than the previous PWM value, the previous PWM duty cycle value is incremented by either 1, 2, 4, or 8 time slots (depending on the setting of bits <6:5> of the fan filter register). If the new PWM duty cycle value is less than the previous PWM value, the previous PWM duty cycle is decremented by 1, 2, 4, or 8 time slots. Each time the PWM duty cycle is incremented or decremented, it is stored as the previous PWM duty cycle for the next comparison.

What does an increase of 1, 2, 4, or 8 time slots actually mean in terms of PWM duty cycle?

A ramp rate of 1 corresponds to one time slot, which is 1/240 of the PWM period. In filtered auto fan speed control mode, incrementing or decrementing by 1 changes the PWM output duty cycle by 0.416%.

Table 12. EFFECT OF RAMP RATES ON PWM_OUT

Ramp Rate	PWM Duty Cycle Change
1	0.416%
2	0.833%
4	1.66%
8	3.33%

So programming a ramp rate of 1, 2, 4, or 8 simply increases or decreases the PWM duty cycle by the amounts shown in Table 12, depending on whether the temperature is increasing or decreasing.

Figure 29 shows remote temperature plotted against PWM duty cycle for filtered mode. The ADC sample rate is the highest sample rate; 11.25 kHz. The ramp rate is set to 8, which would correspond to the fastest ramp rate. With these settings, it took approximately 12 seconds to go from 0% duty cycle to 100% duty cycle (full-speed). The T_{MIN} value = 32°C and the T_{RANGE} = 80°C. Even though the temperature increased very rapidly, the fan gradually ramps up to full speed.

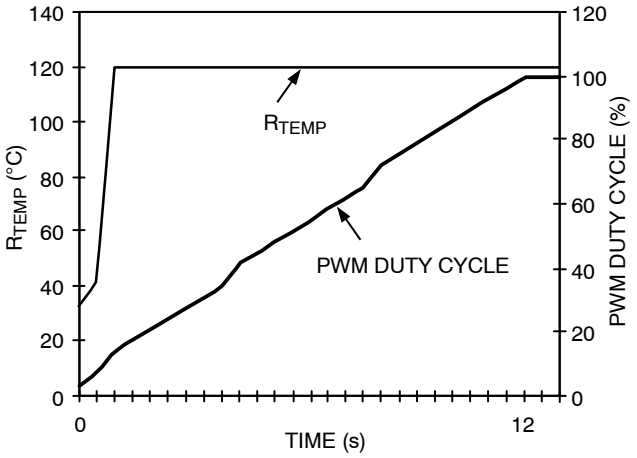


Figure 29. Filtered Mode with Ramp Rate = 8

Figure 30 shows how changing the ramp rate from 8 to 4 affects the control loop. The overall response of the fan is slower. Because the ramp rate is reduced, it takes longer for the fan to achieve full running speed. In this case, it took approximately 22 seconds for the fan to reach full speed.

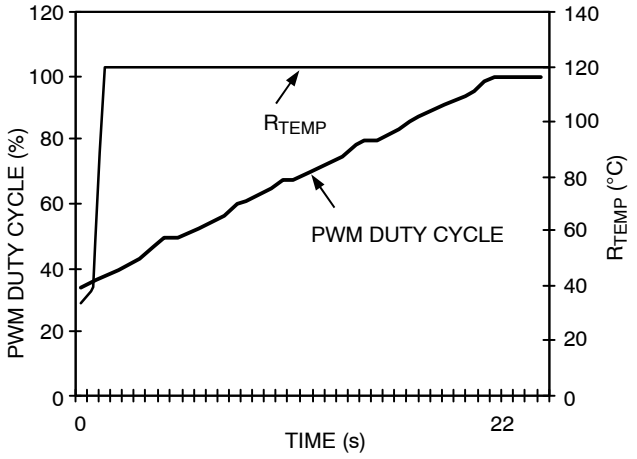


Figure 30. Filtered Mode with Ramp Rate = 4

Figure 31 shows the PWM output response for a ramp rate of 2. In this instance the fan took about 54 seconds to reach full running speed.

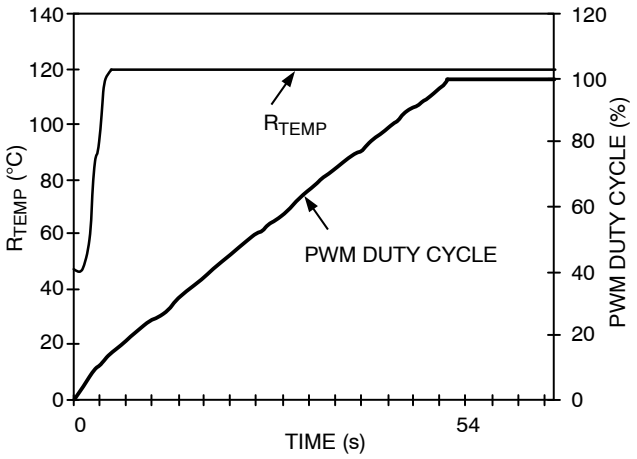


Figure 31. Filtered Mode with Ramp Rate = 2

Finally, Figure 32 shows how the control loop reacts to temperature with the slowest ramp rate. The ramp rate is set to 1, while all other control parameters remain the same. With the slowest ramp rate selected, it took 112 seconds for the fan to reach full speed.

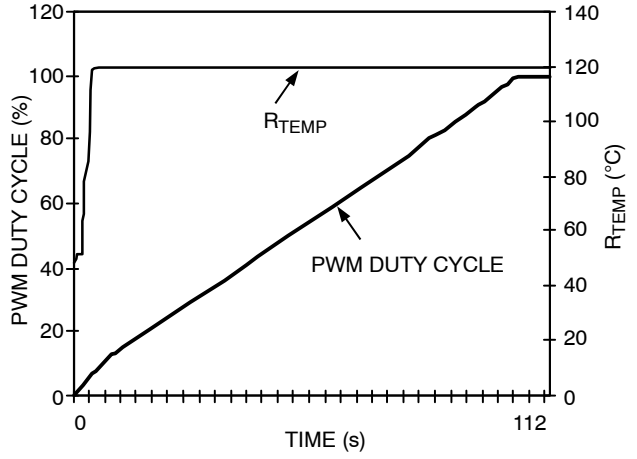


Figure 32. Filtered Mode with Ramp Rate = 1

As can be seen from Figure 29 through Figure 32, the rate at which the fan reacts to temperature change is dependent on the ramp rate selected in the fan filter register. The higher the ramp rate, the faster the fan reaches the newly calculated fan speed.

Figure 33 shows the behavior of the PWM output as temperature varies. As the temperature rises, the fan speed ramps up. Small drops in temperature do not affect the ramp-up function because the newly calculated fan speed is still higher than the previous PWM value. The filtered mode allows the PWM output to be made less sensitive to temperature variations. This is dependent on the ramp rate selected and the ADC sample rate programmed into the fan filter register.

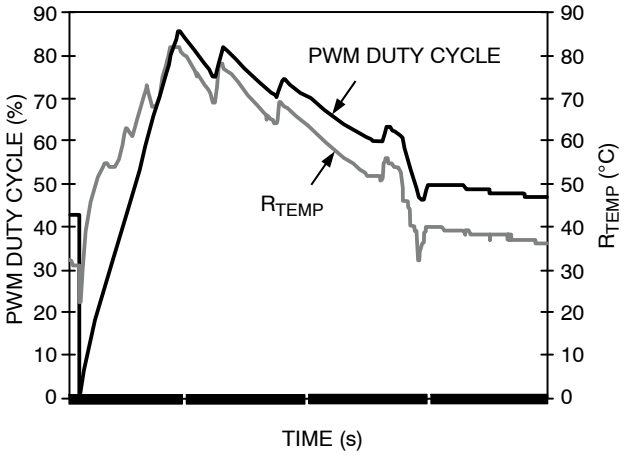


Figure 33. How Fan Reacts to Temperature Variation in Filtered Mode

Effect of ADC Sample Rate on Filtered Mode

The second way to change the filtered mode characteristics is to adjust the ADC sample rate. The faster the ADC sample rate, the more temperature samples are obtained per second. One way to apply filtering to the control loop is to slow down the ADC sampling rate. This means that the number of iterations of the filtered mode algorithm per second is effectively reduced. If the number of temperature measurements per second is reduced, how often the PWM_OUT signal controlling the fan is updated is also reduced.

Bits <4:2> of the fan filter register (Register 0×23) set the ADC sample rate. The default ADC sample rate is 1.4 kHz. The ADC sample rate is selectable from 87.5 Hz to 11.2 kHz. Table 13 shows how many temperature samples are obtained per second, for each of the ADC sample rates.

Table 13. TEMPERATURE UPDATES PER SECOND

ADC Sample Rate	Temperature Updates/Sec
87.5 Hz	0.0625
175 Hz	0.125
350 Hz	0.25
700 kHz	0.5
1.4 kHz	1 (Default)
2.8 kHz	2
5.6 kHz	4
11.2 kHz	8

Relevant Registers for Filtered Automatic Fan Speed Control Mode

In addition to the registers used to program the normal automatic fan speed control mode, the following register needs to be programmed.

Register 0×23 Fan Filter Register

<7> Spin-Up Disable: When this bit is set to 1, fan spin-up is disabled (Default = 0).

<6:5> Ramp Rate: These bits set the ramp rate for filtered mode.
00 = 1 (0.416% Duty Cycle Change)
01 = 2 (0.833% Duty Cycle Change)
10 = 4 (1.66% Duty Cycle Change)
11 = 8 (3.33% Duty Cycle Change)

<4:2> ADC Sample Rate
000 = 87.5 Hz
001 = 175 Hz
010 = 350 Hz
011 = 700 Hz
100 = 1.4 kHz (Default)
101 = 2.8 kHz
110 = 5.6 kHz
111 = 11.2 kHz

<1> Fan 2 Filter Enable: When this bit is set to 1, it enables filtering on Fan 2. Default = 0.

<0> Fan 1 Filter Enable: When this bit is set to 1, it enables filtering on Fan 1. Default = 0.

Programming the Filtered Automatic Fan Speed Control Loop

1. Program a value for $T_{MIN.}$
2. Program a value for the slope $T_{RANGE.}$
3. $T_{MAX} = T_{MIN} + T_{RANGE.}$
4. Program a value for fan spin-up time.
5. Program the desired automatic fan speed control mode behavior, that is, which temperature channel controls the fan.
6. Program a ramp rate for the filtered mode.
7. Program the ADC sample rate in the fan filter register.
8. Set Bit 0 to enable fan filtered mode for Fan 1.
9. Set Bit 1 to enable the fan filtered mode for Fan 2.
10. Select automatic fan speed control mode by setting Bit 7 of Configuration Register 1.

PWM Duty Cycle Select Mode

The ADM1031 can operate under software control by clearing Bit 7 of Configuration Register 1 (Register 0×00). This allows the user to directly control PWM duty cycle for each fan.

Clearing Bit 5 and Bit 6 of Configuration Register 1 allows fan control by varying PWM duty cycle. Values of duty cycle between 0% and 100% can be written to the fan speed configuration register (0×22) to control the speed of each fan. Table 14 shows the relationship between hex values written to the fan speed configuration register and PWM duty cycle obtained.

Table 14. PWM DUTY CYCLE SELECT MODE

Hex Value	PWM Duty Cycle
00	0%
01	7%
02	14%
03	20%
04	27%
05	33%
06	40%
07	47%
08	53%
09	60%
0A	67%
0B	73%
0C	80%
0D	87%
0E	93%
0F	100%

Bits <3:0> set the PWM duty cycle for Fan 1; Bits <7:4> set the PWM duty cycle for Fan 2.

RPM Feedback Mode

The second method of fan speed control under software is RPM feedback mode. This involves programming the desired fan RPM value to the device to set fan speed. The advantages include a very tightly maintained fan RPM over the fan's life, and virtually no acoustic pollution due to fan speed variation.

Fans typically have manufacturing tolerances of ±20%, meaning a wide variation in speed for a typical batch of identical fan models. If it is required that all fans run at exactly 5000 RPM, it can be necessary to specify fans with a nominal fan speed of 6250 RPM. However, many of these fans run too fast and make excess noise. A fan with nominal speed of 6250 RPM could run as fast as 7000 RPM at 100% PWM duty cycle. RPM mode allows all of these fans to be programmed to run at the desired RPM value.

Clearing Bit 7 of Configuration Register 1 (Register 0×00) to 0 places the ADM1031 under software control. Once under software control, the device can be placed into RPM feedback mode by writing to Bit 5 and Bit 6 of Configuration Register 1. Writing a 1 to Bit 5 and Bit 6 selects RPM feedback mode for each fan.

Once RPM feedback mode has been selected, the required fan RPM can be written to the fan tach high limit registers (0×10, 0×11). The RPM feedback mode function allows a fan RPM value to be programmed into the device, and the ADM1031 maintains the selected RPM value by monitoring the fan tach and speeding up the fan as necessary, should the fan start to slow down. Conversely, should the fan start to

speed up due to aging, the RPM feedback slows the fan down to maintain the correct RPM speed. The value to be programmed into each fan tach high limit register is given by:

$$Count = (f \times 60)/R \times N \quad \text{(eq. 7)}$$

where:
f = 11.25 kHz
R = desired RPM value
N = Speed Range; MUST be set to 2

The speed range, N, really determines what the slowest fan speed measured can be before generating an interrupt. The slowest fan speed is measured when the count value reaches 255.

Since N = 2

$$Count = (f \times 60)/R \times N$$
$$R = (f \times 60)/Count \times N$$
$$R = (11250 \times 60)/255 \times 2 \quad \text{(eq. 8)}$$
$$R = (675000)/510$$
$$R = 1324 \text{ RPM, fan fail detect speed.}$$

Programming RPM Values in RPM Feedback Mode

Rather than writing a value such as 5000 to a 16-bit register, an 8-bit count value is programmed instead. The count to be programmed is given by:

$$Count = (f \times 60)/R \times N \quad \text{(eq. 9)}$$

where:
f = 11.25 kHz
R = desired RPM value
N = Speed Range = 2

Example 1:
If the desired value for RPM feedback mode is 5000 RPM, the count to be programmed is:

$$Count = (f \times 60)/R \times N \quad \text{(eq. 10)}$$

Since the desired RPM value, *R* is 5000 RPM, the value for count is:

N = 2:

$$Count = (11250 \times 60)/5000 \times 2$$
$$Count = 675000/10000 \quad \text{(eq. 11)}$$
$$Count = 67 \text{ (assumes 2 tach pulses/rev).}$$

Example 2:
If the desired value for RPM feedback mode is 3650 RPM, the count to be programmed is:

$$Count = (f \times 60)/R \times N \quad \text{(eq. 12)}$$

Since the desired RPM value, *R* is 3650 RPM, the value for count is:

N = 2:

$$Count = (11250 \times 60)/3650 \times 2$$
$$Count = 675000/7300 \quad \text{(eq. 13)}$$
$$Count = 92 \text{ (assumes 2 tach pulses/rev).}$$

Once the count value has been calculated, it should be written to the fan tach high limit register. It should be noted that in RPM feedback mode, there is no high limit register for underspeed detection that can be programmed as there are in the other fan speed control modes. The only time each fan indicates a fan failure condition is whenever the count reaches 255. Since the speed range N = 2, the fan fails if its speed drops below 1324 RPM.

Programming RPM Values

1. Choose the RPM value to be programmed.
2. Set speed range value N = 2.
3. Calculate count value based on RPM and speed range values chosen. Use the count equation to calculate the count value.
4. Clear Bit 7 of Configuration Register 1 (Register 0×00) to place the ADM1031 under software control.
5. Write a 1 to Bit 5 of Configuration Register 1 to place the device in RPM feedback mode.
6. Write the calculated count value to the fan tach high limit register (Register 0×10). The fan speed now goes to the desired RPM value and maintains that fan speed.

RPM Feedback Mode Limitations

RPM feedback mode only controls fan RPM over a limited fan speed range of about 75% to 100%. However, this should be enough range to overcome fan-manufacturing tolerance. In practice, however, the program must not function at too low an RPM value for the fan to run at, or the RPM mode does not operate.

To find the lowest RPM value allowed for a given fan, do the following:

1. Run the fan at 53% PWM duty cycle in software mode. Clear Bit 5 and Bit 7 of Configuration Register 1 (Register 0×00) to enter PWM duty cycle mode. Write 0×08 to the fan speed configuration register (Register 0×22) to set the PWM output to 53% duty cycle.
2. Measure the fan RPM. This represents the fan RPM below which the RPM mode fails to operate. Do not program a lower RPM than this value when using RPM feedback mode.
3. Ensure that speed range N = 2 when using RPM feedback mode.

Fan Drive and Speed Measurement

Fans come in a variety of different options. One distinguishing feature of fans is the number of poles that a fan has internally. The most common fans available have four, six, or eight poles. The number of poles the fan has generally affects the number of pulses per revolution the fan outputs.

If the ADM1031 is used to drive fans other than 4-pole fans that output 2 tach pulses/revolution, then the fan speed measurement equation needs to be adjusted to calculate and

display the correct fan speed, and also to program the correct count value in RPM feedback mode.

Fan Speed Measurement Equations

For a 4-pole fan (2 tach pulses/rev):

$$\text{Fan RPM} = (f \times 60)/\text{Count} \times N \quad \text{(eq. 14)}$$

For a 6-pole fan (3 tach pulses/rev):

$$\text{Fan RPM} = (f \times 60)/(\text{Count} \times N \times 1.5) \quad \text{(eq. 15)}$$

For an 8-pole fan (4 tach pulses/rev):

$$\text{Fan RPM} = (f \times 60)/(\text{Count} \times N \times 2) \quad \text{(eq. 16)}$$

If in doubt as to the number of poles the fans used have, or the number of tach output pulses/rev, consult the fan manufacturer's data sheet, or contact the fan vendor for more information.

Fan Drive Using PWM Control

The external circuitry required to drive a fan using PWM control is extremely simple. A single NMOS FET is the only drive transistor required. The specifications of the MOSFET depend on the maximum current required by the fan being driven. Typical notebook fans draw a nominal 170 mA, and so SOT devices can be used where board space is a constraint. If driving several fans in parallel from a single PWM output, or driving larger server fans, the MOSFET needs to handle the higher current requirements. The only other stipulation is that the MOSFET should have a gate voltage drive, $V_{GS} < 3.3$ V, for direct interfacing to the PWM_OUT pin. The MOSFET should also have a low on-resistance to ensure that there is not significant voltage drop across the FET. This would reduce the maximum operating speed of the fan.

Figure 34 shows how a 3-wire fan can be driven using PWM control.

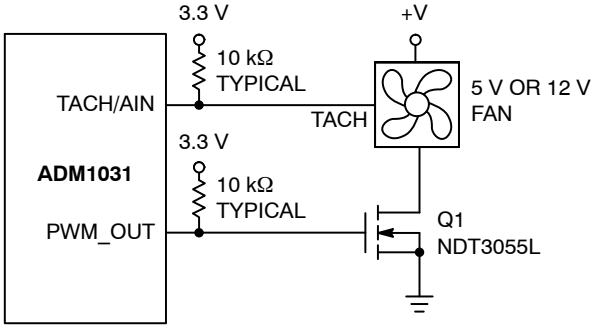


Figure 34. Interfacing the ADM1031 to a 3-wire Fan

The NDT3055L n-type MOSFET was chosen since it has 3.3 V gate drive, low on-resistance, and can handle 3.5 A of current. Other MOSFETs can be substituted based on the system's fan drive requirements.

Figure 35 shows how a 2-wire fan can be connected to the ADM1031. This circuit allows the speed of the 2-wire fan to be measured even though the fan has no dedicated Tach signal. A series R_{SENSE} resistor in the fan circuit converts the fan commutation pulses into a voltage. This is accoupled into the ADM1031 through the 0.01 μF capacitor. On-chip signal conditioning allows accurate monitoring of fan speed. For typical notebook fans drawing approximately 170 mA, a 2 Ω R_{SENSE} value is suitable. For fans such as desktop or server fans that draw more current, R_{SENSE} can be reduced. The smaller R_{SENSE} is, the better, since more voltage is developed across the fan, and the fan then spins faster.

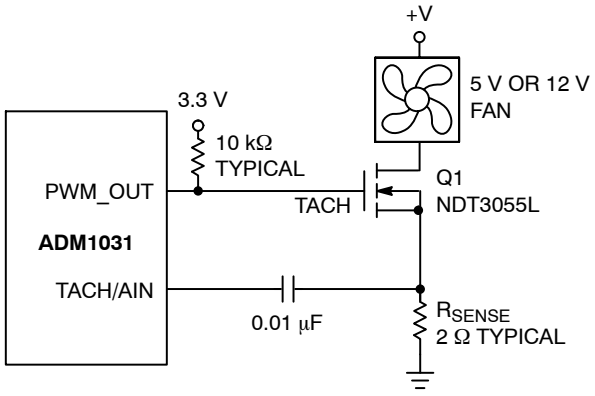


Figure 35. Interfacing the ADM1031 to a 2-wire Fan

Figure 36 shows a typical plot of the sensing waveform at the TACH/AIN pin. The most important thing is that the negative-going spikes are more than 250 mV in amplitude. This is the case for most fans when R_{SENSE} = 2 Ω. The value of R_{SENSE} can be reduced as long as the voltage spikes at the TACH/AIN pin are greater than 250 mV. This allows fan speed to be reliably determined.

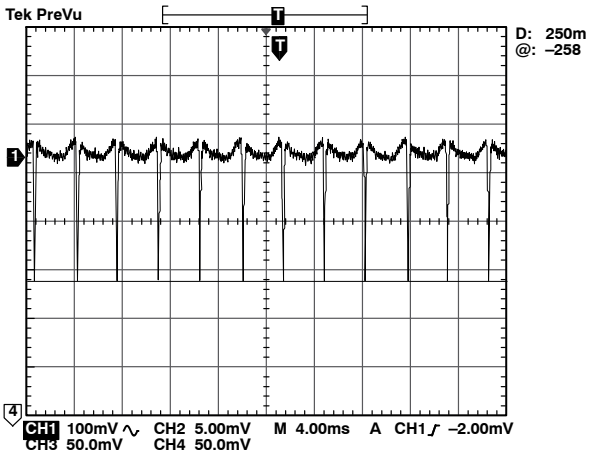


Figure 36. Fan Speed Sensing Waveform at TACH/AIN Pin

Fan Speed Measurement

The fan counter does not count the fan tach output pulses directly, because the fan speed can be less than 1000 RPM and it would take several seconds to accumulate a reasonably large and accurate count. Instead, the period of the fan revolution is measured by gating an on-chip 11.25 kHz oscillator into the input of an 8-bit counter. The fan speed measuring circuit is initialized on the rising edge

of a PWM high output if fan speed measurement is enabled (Bit 2 and Bit 3 of Configuration Register 2 = 1). It then starts counting on the rising edge of the second tach pulse and counts for two fan tach periods, until the rising edge of the fourth tach pulse, or until the counter overranges if the fan tach period is too long. The measurement cycle repeats until monitoring is disabled. The fan speed measurement is stored in the fan speed reading register at address 0×08, 0×09. The fan speed count is given by:

$$Count = (f \times 60)/R \times N \quad \text{(eq. 17)}$$

where:

f = 11.25 kHz

R = Fan Speed in RPM

N = Speed Range (either 1, 2, 4, or 8)

The frequency of the oscillator can be adjusted to suit the expected running speed of the fan by varying N, the speed range. The oscillator frequency is set by Bit 7 and Bit 6 of Fan Characteristics Register 1 (0×20) and Fan Characteristics Register 2 (0×21) as shown in Table 15. Figure 37 shows how the fan measurements relate to the PWM_OUT pulse trains.

Table 15. OSCILLATOR FREQUENCIES

Bit 7	Bit 6	N	Oscillator Frequency (kHz)
0	0	1	11.25
0	1	2	5.625
1	0	4	2.812
1	1	8	1.406

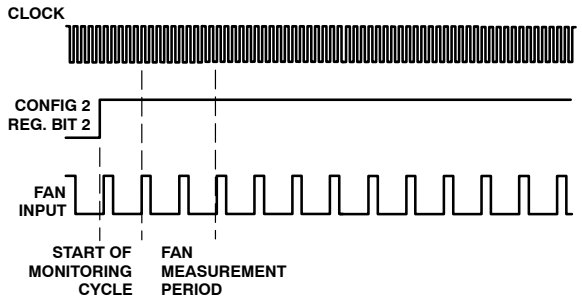


Figure 37. Fan Speed Measurement

In situations where different output drive circuits are used for fan drive, it can be desirable to invert the PWM drive signal. Setting Bit 3 of Configuration Register 1 (0×00) to 1, inverts the PWM_OUT signal. This makes the PWM_OUT pin high for 100% duty cycle. Bit 3 of Configuration Register 1 should generally be set to 1 when using an n-MOS device to drive the fan.

If using a p-MOS device, Bit 3 of Configuration Register 1 should be cleared to 0.

FAN_FAULTs

The $\overline{\text{FAN\_FAULT}}$ output (Pin 8) is an active-low, open-drain output used to signal fan failure to the system processor. Writing a Logic 1 to Bit 4 of Configuration Register 1 (0×00) enables the $\overline{\text{FAN\_FAULT}}$ output pin. The $\overline{\text{FAN\_FAULT}}$ output is enabled by default. The $\overline{\text{FAN\_FAULT}}$ output asserts low only when five consecutive interrupts are generated by the ADM1031 device due to the fan running underspeed, or if the fan is completely stalled. Note that the Fan Tach High Limit must be exceeded by at least one before a $\overline{\text{FAN\_FAULT}}$ can be generated. For example, if we are only interested in getting a $\overline{\text{FAN\_FAULT}}$ if the fan stalls, then the fan speed value is 0×FF for a failed fan. Therefore, we should make the Fan Tach High Limit = 0×FE to allow $\overline{\text{FAN\_FAULT}}$ to be asserted after five consecutive fan tach failures.

Figure 38 shows the relationship between $\overline{\text{INT}}$, $\overline{\text{FAN\_FAULT}}$, and the PWM drive channel. The PWM_OUT channel is driving a fan at some PWM duty cycle, 50% for example, and the fan's tach signal (or fan current for a 2-wire fan) is being monitored at the TACH/AIN pin. Tach pulses are being generated by the fan, during the high time of the PWM duty cycle train. The tach is pulled high during the off time of the PWM train because the fan is connected high-side to the n-MOS device.

Suppose the fan has twice previously failed its fan speed measurement. Looking at Figure 38, PWM_OUT is brought high for two seconds, to restart the fan if it has stalled. Sometime later a third tach failure occurs. This is evident by the tach signal being low during the high time of the PWM pulse, causing the fan speed reading register to reach its maximum count of 255. Since the tach limit has been exceeded, an interrupt is generated on the $\overline{\text{INT}}$ pin. The fan fault bit (Bit 1) of Interrupt Status Register 1 (Register 0×02) is also asserted. Once the processor has acknowledged the $\overline{\text{INT}}$ by reading the status register, the $\overline{\text{INT}}$ is cleared. PWM_OUT is then brought high for another two seconds to restart the fan. Subsequent fan failures cause $\overline{\text{INT}}$ to be reasserted and the PWM_OUT signal is brought high for two seconds (fan spin-up default) each time to restart the fan. Once the fifth tach failure occurs, the failure is deemed to be catastrophic and the $\overline{\text{FAN\_FAULT}}$ pin is asserted low. PWM_OUT is brought high to attempt to restart the fan. The $\overline{\text{INT}}$ pin continues to generate interrupts after the assertion of $\overline{\text{FAN\_FAULT}}$ since tach measurement continues even after fan failure. Should the fan recover from its failure condition, the $\overline{\text{FAN\_FAULT}}$ signal is negated, and the fan returns to its normal operating speed.

Figure 39 shows a typical application circuit for the ADM1031. Temperature monitoring can be based around a CPU diode or discrete transistor measuring thermal hotspots. Either 2- or 3-wire fans can be monitored by the ADM1031, as shown.

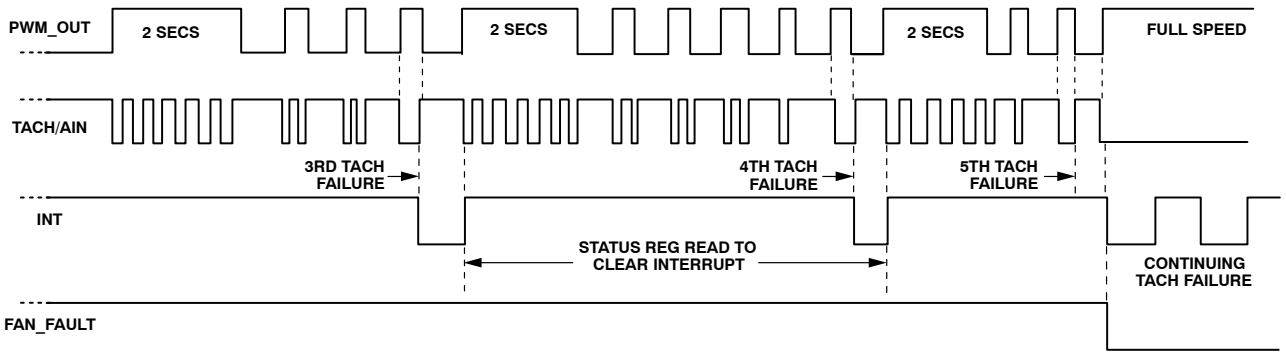


Figure 38. Operation of $\overline{\text{FAN\_FAULT}}$ and Interrupt Pins

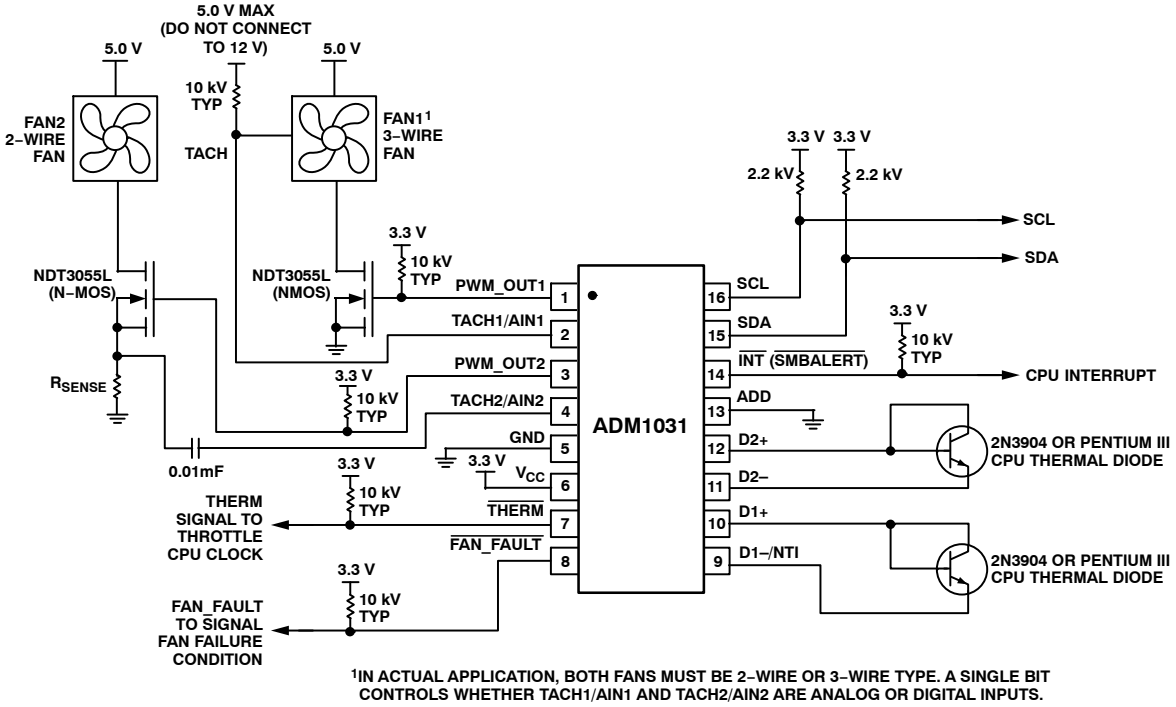


Figure 39. Typical Application Circuit

Table 16. REGISTERS

Register Name	Address A7–A0 in Hex	Comments
Value Registers	0x08–0x1E	See Table 17.
Device ID Register	0x3D	This location contains the device identification number. Since this device is the ADM1031, this register contains 0x31. This register is read only.
Company ID $\overline{\text{THERM}}$	0x3E	This location contains the company identification number (0x41). This register is read only.
Behavior/Revision	0x3F	This location contains the revision number of the device. The lower four bits reflect device revisions [3:0]. Bit 7 of this register is the $\overline{\text{THERM}}$-to-fan enable bit. See Table 30.
Configuration Register 1	0x00	See Table 18. (Power-On Value = 1001 0000)
Configuration Register 2	0x01	See Table 19. (Power-On Value = 0111 1111)
Status Register 1	0x02	See Table 20. (Power-On Value = 0000 0000)
Status Register 2	0x03	See Table 21. (Power-On Value = 0000 0000)
Manufacturer's Test Register	0x07	This register is used by the manufacturer for test purposes only. This register should not be read from or written to in normal operation.
Fan Characteristics Register 1	0x20	See Table 23. (Power-On Value = 0101 1101)
Fan Characteristics Register 2	0x21	See Table 24 . (Power-On Value = 0101 1101)
Fan Speed Configuration Register	0x22	See Table 25. (Power-On Value = 0101 0101)
Fan Filter Register	0x23	See Table 26. (Power-On Value = 0101 0000)
Local Temperature T_{MIN}/T_{RANGE}	0x24	See Table 27. (Power-On Value = 0100 0001)
Remote 1 Temperature T_{MIN}/T_{RANGE}	0x25	See Table 28. (Power-On Value = 0110 0001)
Remote 2 Temperature T_{MIN}/T_{RANGE}	0x26	See Table 29. (Power-On Value = 0110 0001)

Table 17. VALUE REGISTERS

Address	R/$\overline{\text{W}}$	Description
0x06	R	Extended Temperature Resolution (see Table 22).
0x08	R/W	Fan 1 Speed. This register contains the value of the Fan 1 tach measurement.
0x09	R/W	Fan 2 Speed. This register contains the value of the Fan 2 tach measurement.
0x0A	R	Local Temperature Value. This register contains the 8 MSBs of the local temperature measurement.
0x0B	R	Remote 1 Temperature Value. This register contains the 8 MSBs of the Remote 1 temperature reading.
0x0C	R	Remote 2 Temperature Value. This register contains the 8 MSBs of the Remote 2 temperature reading.
0x0D	R/W	Local Temperature Offset. See Table 31. (Power-On Default = 00h)
0x0E	R/W	Remote 1 Temperature Offset. See Table 32. (Power-On Default = 00h)
0x0F	R/W	Remote 2 Temperature Offset. See Table 33. (Power-On Default = 00h)
0x10	R/W	Fan 1 Tach High Limit. This register contains the limit for the Fan 1 tach measurement. Because the tach circuit counts between pulses, a slow fan results in a large measure value, so exceeding the limit is the way to detect a slow or stalled fan. (Power-On Default = FFh)
0x11	R/W	Fan 2 Tach High Limit. This register contains the limit for the Fan 2 tach measurement. Because the tach circuit counts between pulses, a slow fan results in a large measured value, so exceeding the limit is the way to detect a slow or stalled fan. (Power-On Default = FFh)
0x14	R/W	Local Temperature High Limit (Power-On Default 60°C)
0x15	R/W	Local Temperature Low Limit (Power-On Default 0°C)
0x16	R/W	Local Temperature $\overline{\text{THERM}}$ Limit (Power-On Default 70°C)
0x18	R/W	Remote 1 Temperature High Limit (Power-On Default 80°C)
0x19	R/W	Remote 1 Temperature Low Limit (Power-On Default 0°C)
0x1A	R/W	Remote 1 Temperature $\overline{\text{THERM}}$ Limit (Power-On Default 100°C)
0x1C	R/W	Remote 2 Temperature High Limit (Power-On Default 80°C)
0x1D	R/W	Remote 2 Temperature Low Limit (Power-On Default 0°C)
0x1E	R/W	Remote 2 Temperature $\overline{\text{THERM}}$ Limit (Power-On Default 100°C)