# ADM1030

# Intelligent Temperature Monitor and PWM Fan Controller

The ADM1030 is an ACPI-compliant two-channel digital thermometer and under/over temperature alarm, for use in computers and thermal management systems. Optimized for the Pentium® III, the higher 1°C accuracy offered allows systems designers to safely reduce temperature guardbanding and increase system performance. A Pulsewidth Modulated (PWM) Fan Control output controls the speed of a cooling fan by varying output duty cycle. Duty cycle values between 33%–100% allow smooth control of the fan. The speed of the fan can be monitored via a TACH input for a fan with a tach output. The TACH input can be programmed as an analog input, allowing the speed of a 2-wire fan to be determined via a sense resistor. The device will also detect a stalled fan. A dedicated Fan Speed Control Loop provides control even without the intervention of CPU software. It also ensures that if the CPU or system locks up, the fan can still be controlled based on temperature measurements, and the fan speed adjusted to correct any changes in system temperature. Fan Speed may also be controlled using existing ACPI software. One input (two pins) is dedicated to a remote temperaturesensing diode with an accuracy of ±1°C, and a local temperature sensor allows ambient temperature to be monitored. The device has a programmable $\overline{\text{INT}}$ output to indicate error conditions. There is a dedicated $\overline{\text{FAN_FAULT}}$ output to signal fan failure. The $\overline{\text{THERM}}$ pin is a fail-safe output for over-temperature conditions that can be used to throttle a CPU clock.

### Features

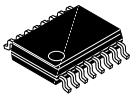
- Optimized for Pentium® III: Allows Reduced Guardbanding Software and Automatic Fan Speed Control
- Automatic Fan Speed Control Allows Control Independent of CPU Intervention after Initial Setup
- Control Loop Minimizes Acoustic Noise and Battery Consumption
- Remote Temperature Measurement Accurate to 1°C Using Remote Diode
- 0.125°C Resolution on Remote Temperature Channel
- Local Temperature Sensor with 0.25°C Resolution
- Pulsewidth Modulation Fan Control (PWM)
- Programmable PWM Frequency
- Programmable PWM Duty Cycle
- Tach Fan Speed Measurement
- 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
- Programmable $\overline{\text{INT}}$ Output Pin
- Configurable Offset for All Temperature Channels
- 3 V to 5.5 V Supply Range
- Shutdown Mode to Minimize Power Consumption
- This is a Pb-Free Device*

### Applications

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

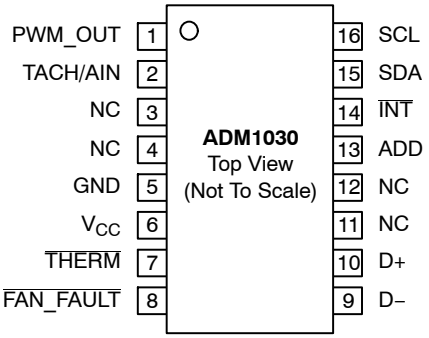
**ON Semiconductor®**

**http://onsemi.com**



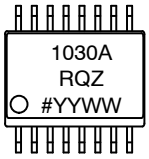
**QSOP–16**
**CASE 492**

### PIN ASSIGNMENT




NC = No Connect

### MARKING DIAGRAM




| | |
|---|---|
| 1029ARQZ | = Special Device Code |
| # | = Pb-Free Package |
| YY | = Year |
| WW | = Work Week |

### ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 29 of this data sheet.

* For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

© Semiconductor Components Industries, LLC, 2012

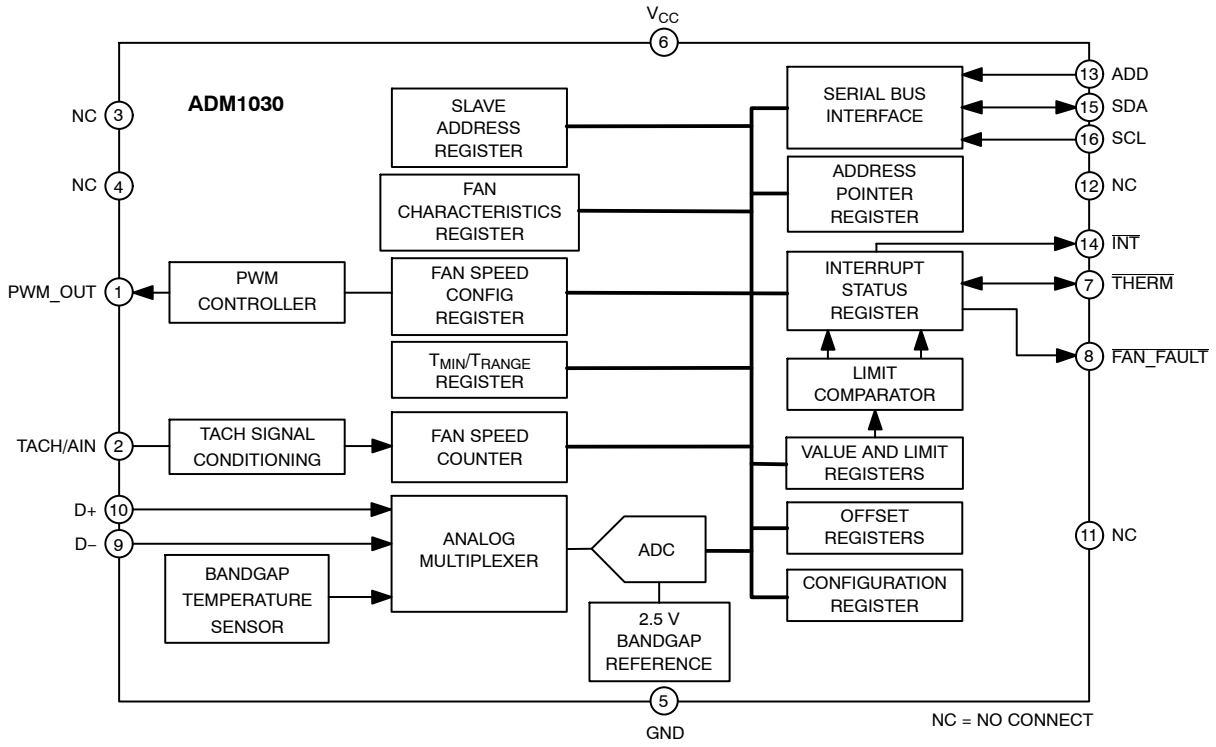



**Figure 1. Functional Block Diagram**

**Table 1. ABSOLUTE MAXIMUM RATINGS**

| Rating | Value | 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 ($T_{JMAX}$) | 150 | °C |
| Storage Temperature Range | −65 to +150 | °C |
| Lead Temperature, Soldering<br>Vapor Phase (60 sec)<br>Infrared (15 sec) | <br>215<br>200 | °C |
| ESD Rating All Pins | 2,000 | 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.

**WARNING:** Electrostatic Sensitive Device − Do not open packages or handle except at a static-free workstation.

**WARNING:** Moisture Sensitive Device − Non-RoHS Compliant − Level 3 MSL; RoHS Compliant − Level 4 MSL. Do not open packages except under controlled conditions.

**Table 2. THERMAL CHARACTERISTICS**

| Package Type | $\theta_{JA}$ | $\theta_{JC}$ | Unit |
|---|---|---|---|
| 16-lead QSOP | 105 | 39 | °C/W |

**Table 3. PIN FUNCTION DESCRIPTIONS**

| Pin No. | Mnemonic | Description |
|---|---|---|
| 1 | PWM_OUT | Digital Output (Open-Drain). Pulsewidth modulated output to control fan speed. Requires pull-up resistor (10 kΩ typical). |
| 2 | TACH/AIN | Digital/Analog Input. Fan tachometer input to measure fan speed. May be reprogrammed as an analog input to measure speed of a 2-wire fan via a sense resistor (2 Ω typical) |
| 3, 4, 11, 12 | NC | Not Connected. |
| 5 | GND | System Ground. |
| 6 | $V_{CC}$ | 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 pull-up resistor (10 kΩ). |
| 8 | $\overline{\text{FAN_FAULT}}$ | Digital Output (Open-Drain). Can be used to signal a fan failure. Requires pull-up resistor (typically 10 kΩ). |
| 9 | D– | Analog Input. Connected to cathode of an external temperature-sensing diode. The temperature-sensing element is either a Pentium® III substrate transistor or a general-purpose 2N3904. |
| 10 | D+ | Analog Input. Connected to anode of the external temperature-sensing diode. |
| 13 | ADD | Three-state Logic Input. Sets two lower bits of device SMBus address. |
| 14 | $\overline{\text{INT}}$ | Digital Output (Open-Drain). Can be programmed as an interrupt output for temperature/fan speed interrupts. Requires pull-up resistor (10 kΩ typical). |
| 15 | SDA | Digital I/O. Serial Bus Bidirectional Data. Open-drain output. Requires pull-up resistor (2.2 kΩ typical). |
| 16 | SCL | Digital Input. Serial Bus Clock. Requires pull-up 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 | 5.5 | V |
| Supply Current, $I_{CC}$ | Interface Inactive, ADC Active<br>Standby Mode | –<br>– | 1.4<br>32 | 3.0<br>50 | mA<br>μA |
| **TEMPERATURE-TO-DIGITAL CONVERTER** | | | | | |
| Internal Sensor Accuracy | | – | ±1.0 | ±3.0 | °C |
| Resolution | | – | 0.25 | – | °C |
| External Diode Sensor Accuracy | 60°C < $T_D$ < 100°C | – | – | ±1.0 | °C |
| Resolution | | – | 0.125 | – | °C |
| Remote Sensor Source Current | High Level<br>Low Level | –<br>– | 180<br>11 | –<br>– | 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 V | – | – | 0.4 | V |
| High Level Output Current, $I_{OH}$ | $V_{OUT} = V_{CC}$, $V_{CC}$ = 3 V | – | 0.1 | 1.0 | μA |
| **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 | – | pF |
| **DIGITAL INPUT LOGIC LEVELS (ADD, $\overline{\text{THERM}}$, TACH)** (Note 2) | | | | | |
| Input High Voltage, $V_{IH}$ | | 2.1 | – | – | V |
| Input Low Voltage, $V_{IL}$ | | – | – | 0.8 | V |
| **OPEN-DRAIN SERIAL DATA BUS OUTPUT (SDA)** | | | | | |
| Output Low Voltage, $V_{OL}$ | $I_{OUT}$ = –6.0 mA, $V_{CC}$ = 3 V | – | – | 0.4 | V |
| High Level Output Leakage Current, $I_{OH}$ | $V_{OUT} = V_{CC}$ | – | 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 |
|---|---|---|---|---|---|
| **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 |
| **FAN RPM-TO-DIGITAL CONVERTER** | | | | | |
| Accuracy | 60°C ≤ $T_A$ ≤ 100°C | – | – | ±6 | % |
| Resolution | | – | – | 8 | Bits |
| TACH Nominal Input RPM | Divisor N = 1, Fan Count = 153<br>Divisor N = 2, Fan Count = 153<br>Divisor N = 4, Fan Count = 153<br>Divisor N = 8, Fan Count = 153 | –<br>–<br>–<br>– | 4400<br>2200<br>1100<br>550 | –<br>–<br>–<br>– | 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, $t_R$ | | – | – | 1,000 | 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 the most likely parametric norm. Shutdown current typ is measured with $V_{CC}$ = 3.3 V.
2. ADD is a three-state input that may 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 $V_{IH}$ = 2.1 V for a rising edge.

NOTE: Specifications subject to change without notice.

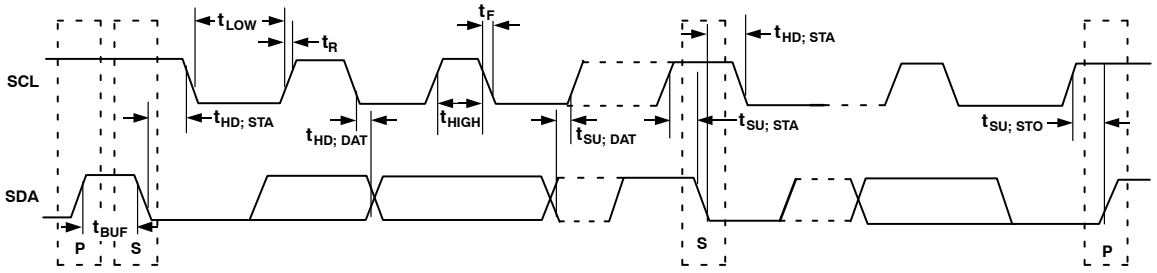



**Figure 2. Serial Bus Timing Diagram**

## TYPICAL PERFORMANCE CHARACTERISTICS

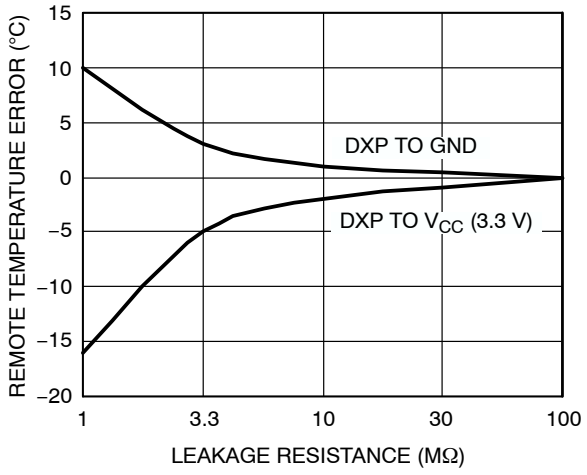



**Figure 3. Temperature Error vs. PCB Track Resistance**

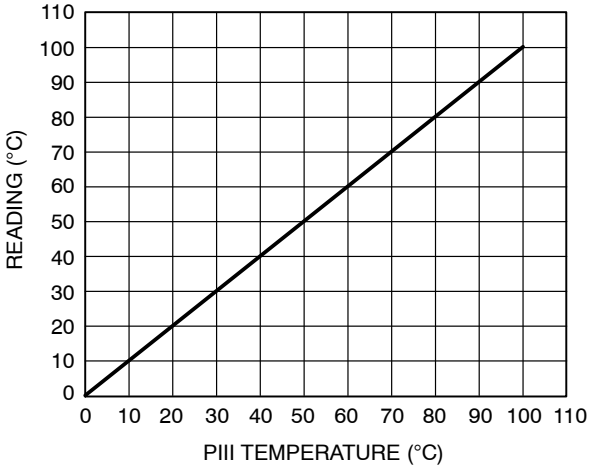



**Figure 4. Pentium® III Temperature Measurement vs. ADM1030 Reading**

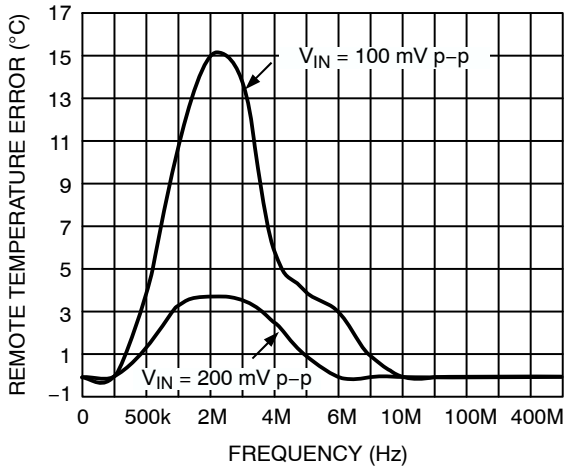



**Figure 5. Temperature Error vs. Power Supply Noise Frequency**

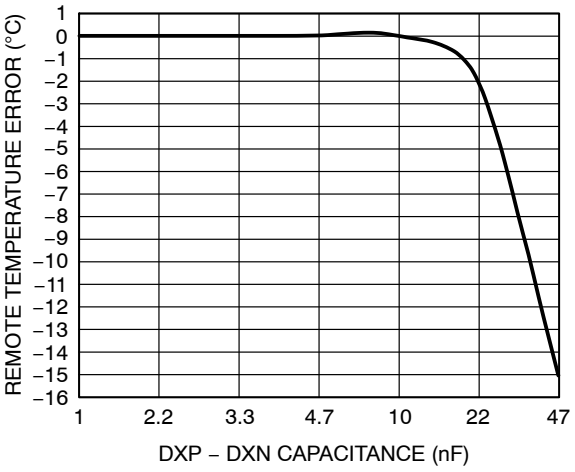



**Figure 6. Temperature Error vs. Capacitance between D+ and D−**

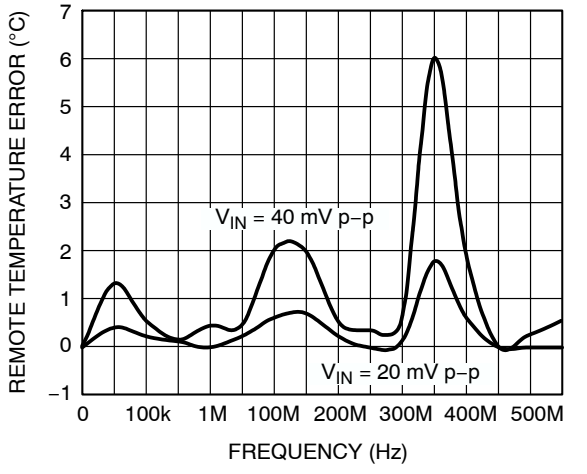



**Figure 7. Temperature Error vs. Common-mode Noise Frequency**

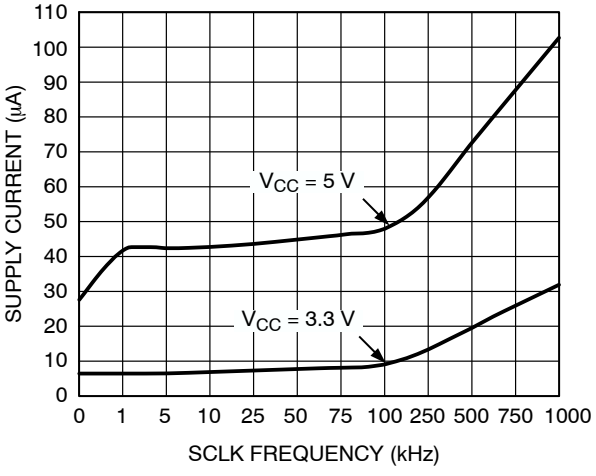



**Figure 8. Standby Current vs. Clock Frequency**

## TYPICAL PERFORMANCE CHARACTERISTICS (Cont'd)

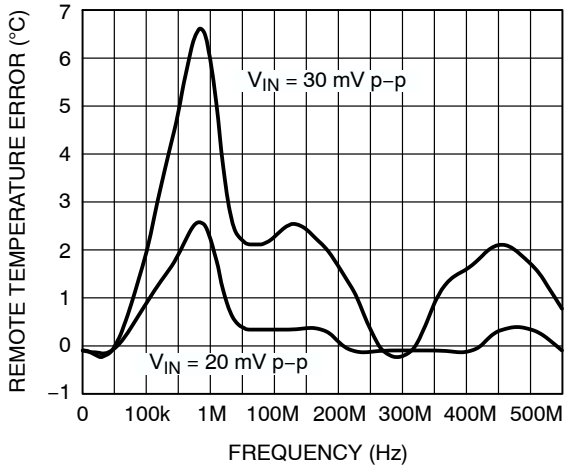



**Figure 9. Temperature Error vs. Differential-mode Noise Frequency**

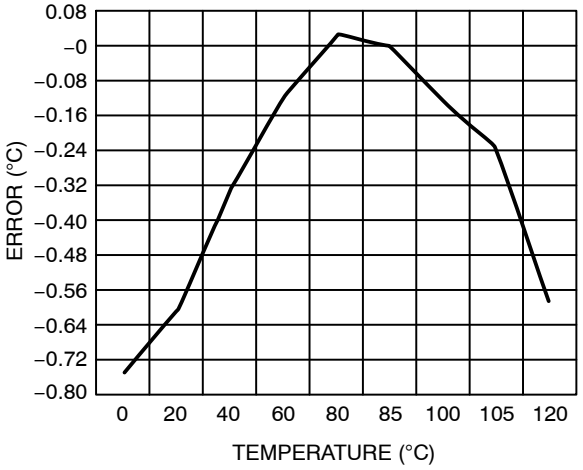



**Figure 10. Remote Sensor Error**

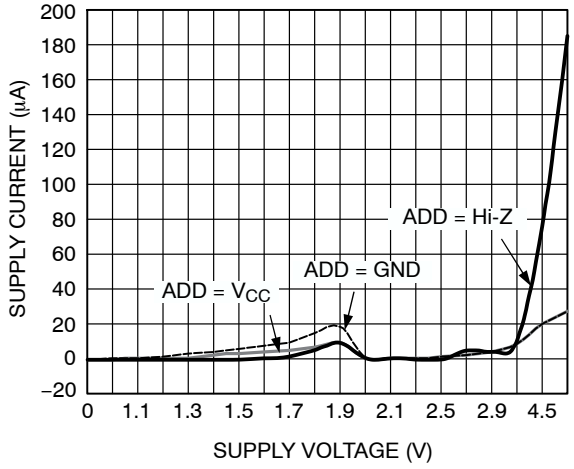



**Figure 11. Standby Supply Current vs. Supply Voltage**

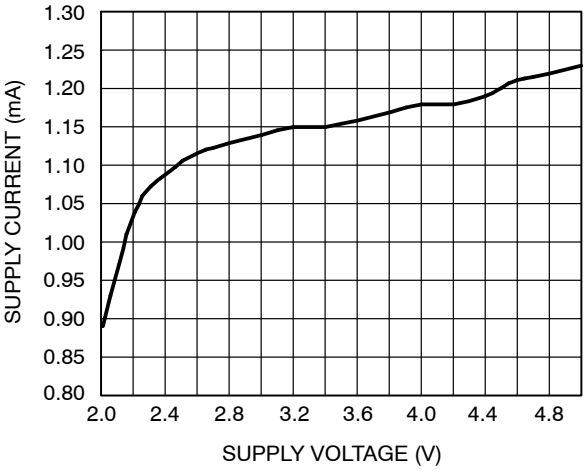



**Figure 12. Supply Current vs. Supply Voltage**

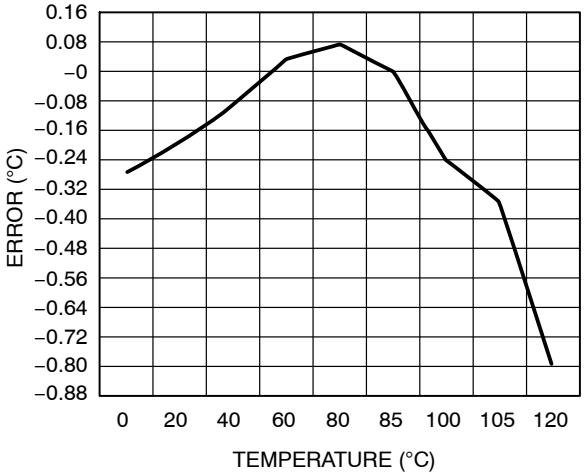



**Figure 13. Local Sensor Error**

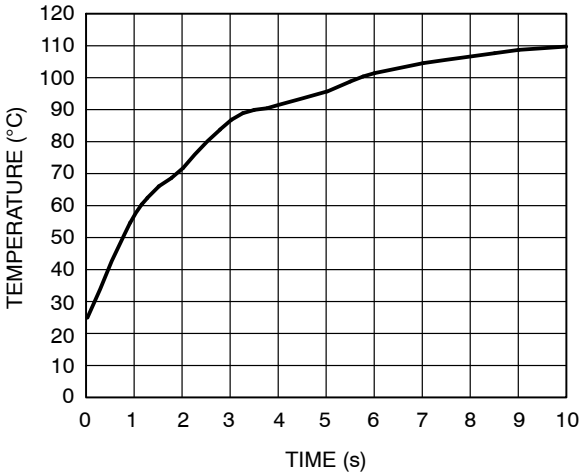



**Figure 14. Response to Thermal Shock**

## General Description

The ADM1030 is a temperature monitor and PWM fan controller for microprocessor-based systems. The device communicates with the system via a serial System Management Bus. 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 ADM1030 are performed over the serial bus. The device also supports the SMBus Alert Response Address (ARA) function.

## Internal Registers of the ADM1030

A brief description of the ADM1030's principal internal registers is given below. More detailed information on the function of each register is given in Table 16 to Table 30.

### Configuration Register

Provides control and configuration of 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 ADM1030, 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

The results of temperature and fan speed measurements are stored in these registers, along with their limit values.

### Fan Speed Config Register

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

### Offset Registers

Allows the temperature channel readings to be offset by a 5-bit two's complement value written to these registers. These values will automatically be 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 Register

This register is used to select the spin-up time, PWM frequency, and speed range for the fan used.

### THERM Limit Registers

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

### $T_{MIN}/T_{RANGE}$ Registers

These registers are read/write registers that hold the minimum temperature value below which the fan will not run when the device is in Automatic Fan Speed Control Mode. These registers also hold the values defining the range over that auto fan control will be provided, and hence determines the temperature at which the fan will run at full speed.

## Serial Bus Interface

Control of the ADM1030 is carried out via the SMBus. The ADM1030 is connected to this bus as a slave device, under the control of a master device, e.g., the 810 chipset. The ADM1030 has a 7-bit serial bus address. When the device is powered up, it will do 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 power-up, so changing ADD with power on will have 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, the default address will be 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 ADM1030 is used in a system.

The serial bus protocol operates as follows:

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 will follow. All slave peripherals connected to the serial bus respond to the START condition, and shift in the next 8 bits, consisting of a 7-bit address (MSB first) plus an R/$\overline{\text{W}}$ bit that determines the direction of the data transfer, i.e., whether data will be 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, the master will write to the slave device. If the R/$\overline{\text{W}}$ bit is a 1, the master will read 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 may 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 will pull the data line high during the tenth clock pulse to assert a STOP condition. In READ mode, the master device will override 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 will then take 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 may 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 ADM1030, write operations contain either one or two bytes, and read operations contain one byte, and perform the following functions.

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 into 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, then 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.

When reading data from a register there are two possibilities:

1. If the ADM1030'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 ADM1030 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, because the first data byte of a write is always written to the Address Pointer Register.
2. In Figures 15 to 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 ADM1030 also supports the Read Byte protocol, as described in the System Management Bus specification.

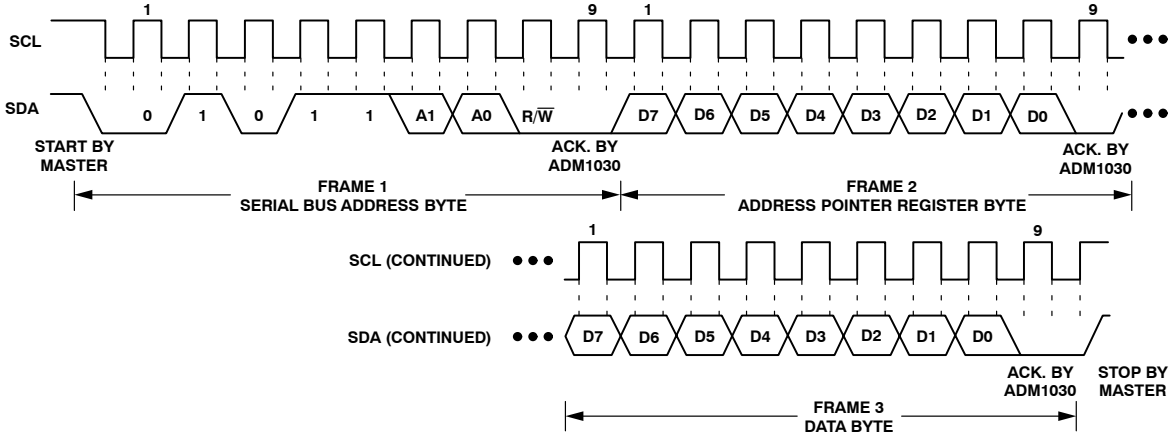



**Figure 15. Writing a Register Address to the Address Pointer Register, then Writing Data to the Selected Register**

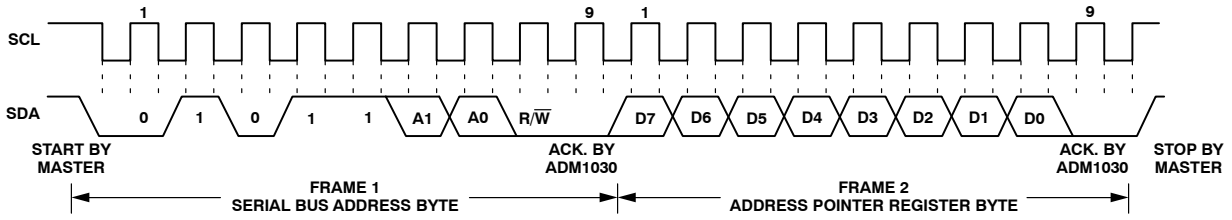



**Figure 16. Writing to the Address Pointer Register Only**

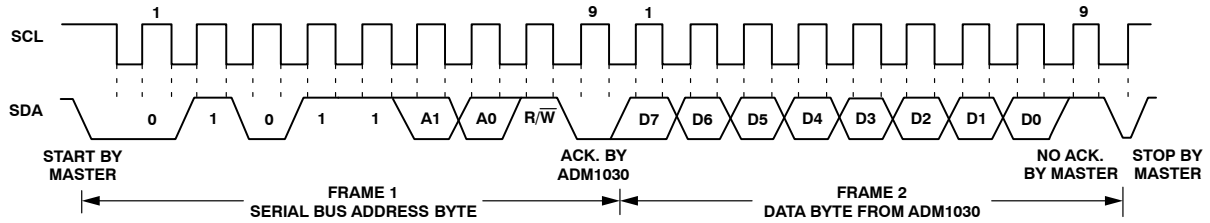



**Figure 17. Reading Data from a Previously Selected Register**

## 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. $\overline{\text{SMBALERT}}$ 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 will have priority, in accordance with normal SMBus arbitration.
5. Once the ADM1030 has responded to the Alert Response Address, it will reset its $\overline{\text{INT}}$ output; however, if the error condition that caused the interrupt persists, $\overline{\text{INT}}$ will be reasserted on the next monitoring cycle.

## Temperature Measurement System

### Internal Temperature Measurement

The ADM1030 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 two's 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 Temperature Measurement

The ADM1030 can measure the temperature of an external diode sensor or diode-connected transistor, connected to Pins 9 and 10.

These pins are a dedicated temperature input channel. 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, so the technique is unsuitable for mass production.

The technique used in the ADM1030 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.

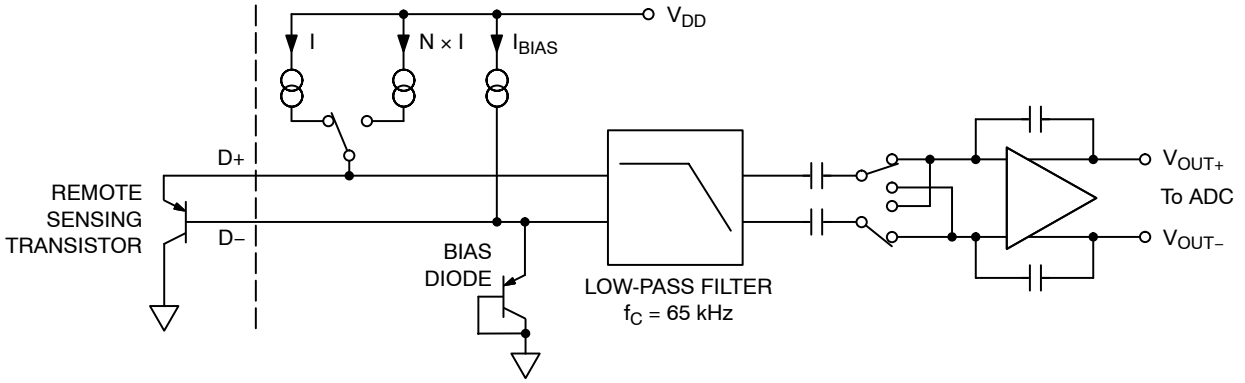



**Figure 18. Signal Conditioning**

If a discrete transistor is used, the collector will not be grounded, and should be 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 ADM1030 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 Tables 7 and 8.

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

| Temperature | 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 (°C) | Remote Temperature Low Bits |
|---|---|
| 0.000 | 000 |
| 0.125 | 001 |
| 0.250 | 010 |
| 0.375 | 011 |
| 0.500 | 100 |
| 0.625 | 101 |
| 0.750 | 110 |
| 0.875 | 111 |

The extended temperature resolution for the local and remote channels is stored in the Extended Temperature Resolution Register (Register 0x06), and is outlined in Table 22.

**Table 8. LOCAL SENSOR EXTENDED TEMPERATURE RESOLUTION**

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

To prevent ground noise interfering with the measurement, the more negative terminal of the sensor is not referenced to ground, but is 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 may be placed between the D+ and D– inputs to filter the noise.

To measure $\Delta V_{BE}$, 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 chopperstabilized amplifier that performs the functions of amplification and rectification of the waveform to produce a dc voltage proportional to $\Delta V_{BE}$. This voltage is measured by the ADC to give a temperature output in 11-bit two's 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 ADM1030 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 pick-up. 10 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 ADM1030.
6. If the distance to the remote sensor is more than 8 inches, the use of twisted pair cable is recommended. This will work up to about 6 to 12 feet.
7. For really long distances (up to 100 feet) use shielded twisted pair such as Belden #8451 microphone cable. Connect the twisted pair to D+ and D– and the shield to GND close to the ADM1030. 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 may be reduced or removed. In any case the total shunt capacitance should not exceed 1000 pF.

Cable resistance can also introduce errors. 1 Ω series resistance introduces about 0.5°C error.

### Addressing the Device

ADD (Pin 13) is a three-state input. It is sampled, on power-up 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 ADM1030 Interrupt System

The ADM1030 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 (described in more detail later).

$\overline{\text{THERM}}$ 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 will be asserted low to indicate an out-of-limit condition. If the temperature exceeds the high temperature limit, the $\overline{\text{INT}}$ pin will also be asserted low. A third limit; $\overline{\text{THERM}}$ limit, may be programmed into the device to set the temperature limit above which the overtemperature $\overline{\text{THERM}}$ pin will be 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, the fan will not run full-speed. The $\overline{\text{THERM}}$ limit may 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 may 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, the fan will run 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 will assert $\overline{\text{INT}}$ 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 ADM1030 does not have a dedicated Interrupt Mask Register, clearing the appropriate enable bits in Configuration Register 2 will clear the appropriate interrupts and mask out future interrupts on that channel. Disabling interrupt bits will prevent 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 over-temperature conditions. When asserted low as an output, the fan will be driven full-speed if the $\overline{\text{THERM}}$-to-Fan Enable bit is set to 1 (Bit 7 of Register 0x3F). 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 fan is 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, the fan will still be 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

Registers 1 and 2 (0x02, 0x03). Bits 0 and 1 (Alarm Speed, Fan Fault) of Status Register 1, once set, may be cleared by reading Status Register 1. Once the Alarm Speed bit is cleared, this bit will not be reasserted on the next monitoring cycle even if the condition still persists. This bit may 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 will reassert on subsequent fan tach failures.

Bits 2 and 3 of Status Register 1 are the Remote 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 will be reasserted if the out-of limit condition still exists on the next monitoring cycle. Bits 6 and 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 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 will be a 1 if a short or open is detected on the Remote Temperature channel on power-up. If this bit is set to 1 on power-up, it cannot be cleared. Bit 6 of Status Register 2 (0x03) indicates that the Local 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.

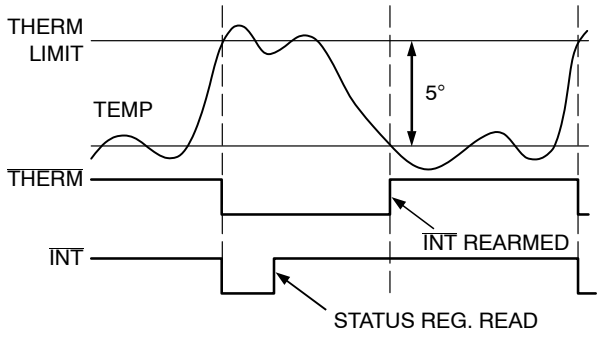



**Figure 20. Operation of $\overline{\text{THERM}}$ and $\overline{\text{INT}}$ Signals**

Figure 20 shows the interaction between $\overline{\text{INT}}$ and $\overline{\text{THERM}}$. Once a critical temperature 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 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 THERM limit –5°C. This prevents unnecessary interrupts from tying up valuable CPU resources.

### Modes of Operation

The ADM1030 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 ADM1030 has a local temperature channel and a remote temperature channel, which may be connected to an on-chip diode-connected transistor on a CPU. These two

temperature channels may be used as the basis for an automatic fan speed control loop to drive a fan using Pulsewidth Modulation (PWM).

### How Does the Control Loop Work?

The Automatic Fan Speed Control Loop is shown in Figure 21 below.

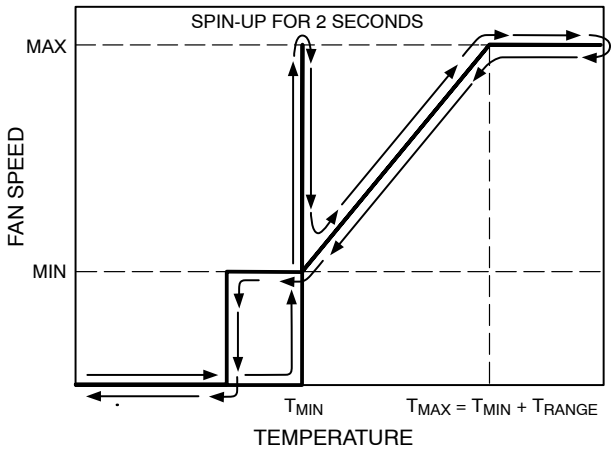



**Figure 21. Automatic Fan Speed Control**

In order for the fan speed control loop to work, certain loop parameters need to be programmed into the device.

1. **$T_{MIN}$.** The temperature at which the fan should switch on and run at minimum speed. The fan will only turn on once the temperature being measured rises above the $T_{MIN}$ value programmed. The fan will spin up for a predetermined time (default = 2 secs). See Fan Spin-up section for more details.
2. **$T_{RANGE}$.** The temperature range over which the ADM1030 will automatically adjust the fan speed. As the temperature increases beyond $T_{MIN}$, the PWM_OUT duty cycle will be increased accordingly. The $T_{RANGE}$ parameter actually defines the fan speed versus temperature slope of the control loop.
3. **$T_{MAX}$.** The temperature at which the fan will be at its maximum speed. At this temperature, the PWM duty cycle driving the fan will be 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.
4. 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 will continue to run until such time as 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 ADM1030. $T_{MIN}$ was set to 0°C to start all slopes from the same point. It can be seen how changing the $T_{RANGE}$ value affects the PWM duty cycle versus temperature slope.

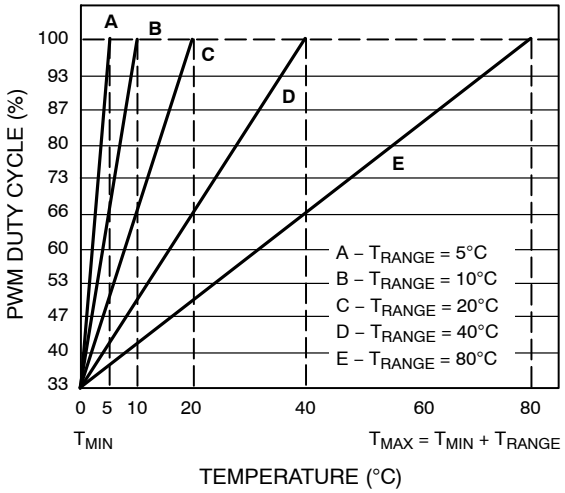



**Figure 22. PWM Duty Cycle vs. Temperature Slopes ($T_{RANGE}$)**

Figure 23 shows how, for a given $T_{RANGE}$, changing the $T_{MIN}$ value affects the loop. Increasing the $T_{MIN}$ value will increase 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 $T_{MIN}$ may be changed in increments of 4°C.

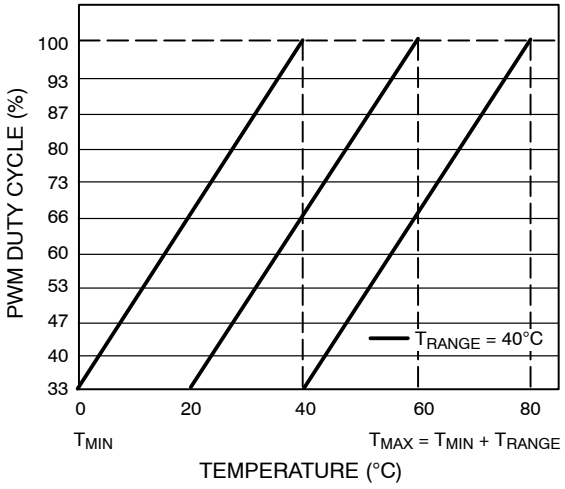



**Figure 23. Effect of Increasing $T_{MIN}$ Value on Control Loop**

### Fan Spin-up

As was previously mentioned, once the temperature being measured exceeds the $T_{MIN}$ value programmed, the fan will turn 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 may be reduced in line with the temperature being measured.

The ADM1030 allows fan spin-up times between 200 ms and 8 seconds. Bits <2:0> of Fan Characteristics Register 1 (Register 0x20) program the fan spin-up time.

**Table 9. FAN SPIN-UP TIMES**

| Bits 2:0 | Spin-up Times (Fan Characteristics Register 1) |
|---|---|
| 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 ADM1030 device may be programmed. The ADM1030 is placed into Automatic Fan Speed Control Mode by setting Bit 7 of Configuration Register 1 (Register 0x00). The device powers up into Automatic Fan Speed Control Mode by default. The control mode offers further flexibility in that the user can decide which temperature channel/channels control the fan.

**Table 10. AUTO MODE FAN BEHAVIOR**

| Bits 6, 5 | Control Operation (Config Register 1) |
|---|---|
| 00 | Remote Temperature Controls the Fan |
| 11 | Maximum Speed Calculated by Local and Remote Temperature Channels Control the Fan |

When Bits 5 and 6 of Config Register 1 are both set to 1, it offers increased flexibility. 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 fan.

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 Bits 5 and 6 of Config 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 $T_{MAX}$ will thus be 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 will be 80°C.

Consider if both temperature channels measure 40°C. Both control loops will calculate a PWM duty cycle of 66%. Therefore, the fan will be driven at 66% duty cycle.

If both temperature channels measure 20°C, the local channel will calculate 33% PWM duty cycle, while the remote channel will calculate 50% PWM duty cycle. Thus, the fan will be driven at 50% PWM duty cycle. Consider the local temperature measuring 60°C while the remote temperature is measuring 70°C. The PWM duty cycle calculated by the local temperature control loop will be 100% (since the temperature = $T_{MAX}$). The PWM duty cycle calculated by the remote temperature control loop at 70°C will be approximately 90%. So the fan will run full-speed (100% duty cycle). Remember, that the fan speed will be 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, may actually calculate a faster speed, than a higher temperature on the other channel.

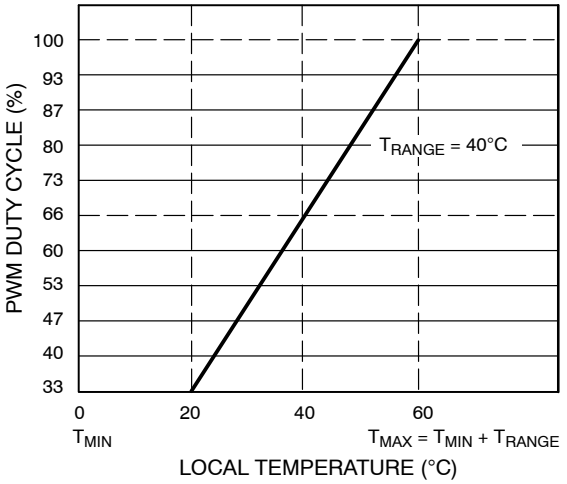



**Figure 24. Max Speed Calculated by Local Temperature Control Loop Drives Fan**

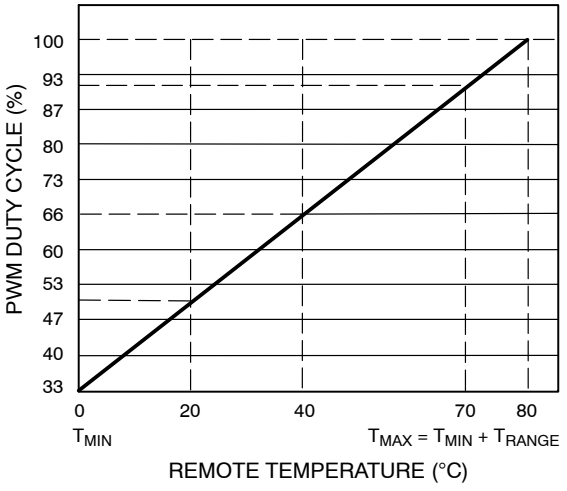



**Figure 25. Max Speed Calculated by Remote Temperature Control Loop Drives Fan**

### 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, i.e., 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

Having programmed all the above loop parameters, are there any other parameters to worry about?

$T_{MIN}$ was defined as being the temperature at which the fan switched on and ran at minimum speed. This minimum speed is 33% duty cycle by default. If the minimum PWM duty cycle is programmed to 33%, the fan control loops will operate as previously described.

It should be noted however, 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 will vary over the range 33% to 100%. The fan will run 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 Slopes 2 and 3. Take Case 2, where the minimum PWM duty cycle is reprogrammed from 33% (default) to 53%.

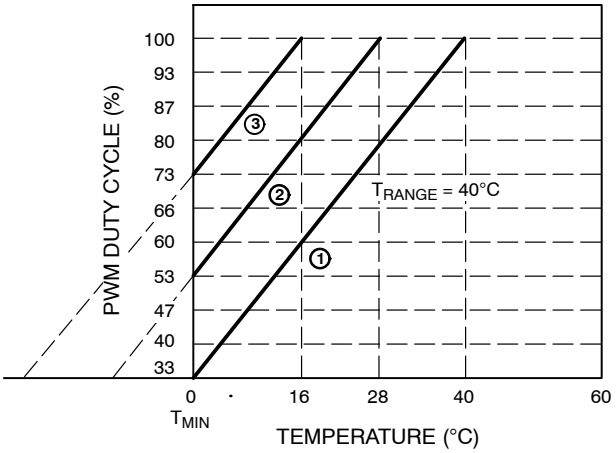



**Figure 26. Effect of Changing Minimum Duty Cycle on Control Loop with Fixed $T_{MIN}$ and $T_{RANGE}$ Values**

The fan will actually reach full-speed at a much lower temperature, 28°C. Case 3 shows that when the minimum PWM duty cycle was increased to 73%, the temperature at which the fan ran full-speed was 16°C. So the effect of increasing the minimum PWM duty cycle, with a fixed $T_{MIN}$ and fixed $T_{RANGE}$, is that the fan will actually reach 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 Config Register (Register 0x22). Table 11 shows the relationship between the decimal values written to the Fan Speed Config 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 (0x0A) | 67% |
| 11 (0x0B) | 73% |
| 12 (0x0C) | 80% |
| 13 (0x0D) | 87% |
| 14 (0x0E) | 93% |
| 15 (0x0F) | 100% |

The temperature at which the fan will run 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 versus 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 0x00 Configuration Register 1

<7> Logic 1 selects Automatic Fan Speed Control, Logic 0 selects software control (Default = 1).

<6:5> 00 = Remote Temperature controls Fan
11 = Fastest Calculated Speed controls the fan when Bit 7 = Logic 1.

### Register 0x20 Fan Characteristics Register 1

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

<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 0x22 Fan Speed Configuration Register

<3:0> Min Speed: This nibble contains the speed at which the fan will run when the temperature is at $T_{MIN}$. The default is 0x05, meaning that the fan will run at 33% duty cycle when the temperature is at $T_{MIN}$.

### Register 0x24 Local Temp $T_{MIN}/T_{RANGE}$

<7:3> Local Temp $T_{MIN}$. These bits set the temperature at which the fan will turn 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 0x25 Remote Temperature $T_{MIN}/T_{RANGE}$

<7:3> Remote Temperature $T_{MIN}$. Sets the temperature at which the fan will switch on based on Remote 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 will be 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, i.e., the PWM duty cycle will respond immediately to temperature change. In certain circumstances, we may not want the PWM output to react instantaneously to temperature changes. If significant variations in temperature were found in a system, it would have the effect of changing the fan speed, 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 will effectively be ignored, and the fan's operation will be 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 ADM1030 is invoked by setting Bit 0 of the Fan Filter Register (Register 0x23). Once the Fan Filter Register has been written to, and all other control loop parameters ($T_{MIN}$, $T_{RANGE}$, etc.) have been programmed, the device may be placed into Automatic Fan Speed Control Mode by setting Bit 7 of Configuration Register 1 (Register 0x00) 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 will have 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 into 240 equal time slots. One time slot corresponds to the smallest possible increment in PWM duty cycle. A PWM signal of 33% duty cycle will thus be high for 1/3 × 240 time slots and low for 2/3 × 240 time slots. Therefore, 33% PWM duty cycle corresponds to a signal which is high for 80 time slots and low for 160 time slots.

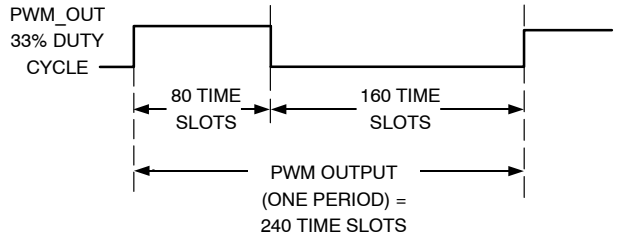



**Figure 27. 33% PWM Duty Cycle Represented in Time Slots**

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 will be 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.

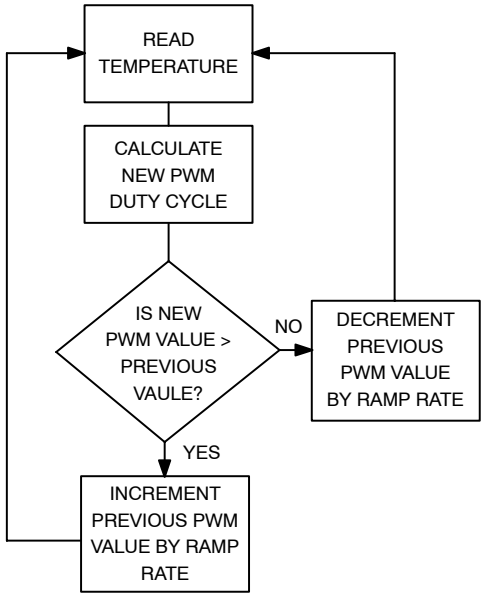



**Figure 28. Filtered Mode Algorithm**

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.

So 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 9, 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. It can be seen that even though the temperature increased very rapidly, the fan gradually ramps up to full speed.

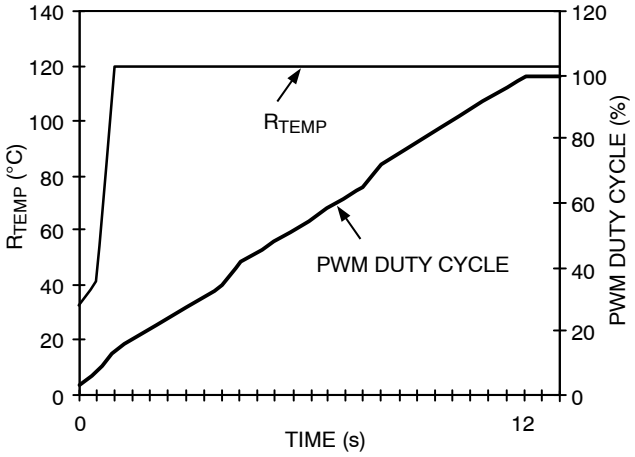



**Figure 29. Filtered Mode with Ramp Rate = 8**

Figure 30 shows how changing the ramp rate from 8 to 4 affects the control loop. The overall response of the fan is slower. Since 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.

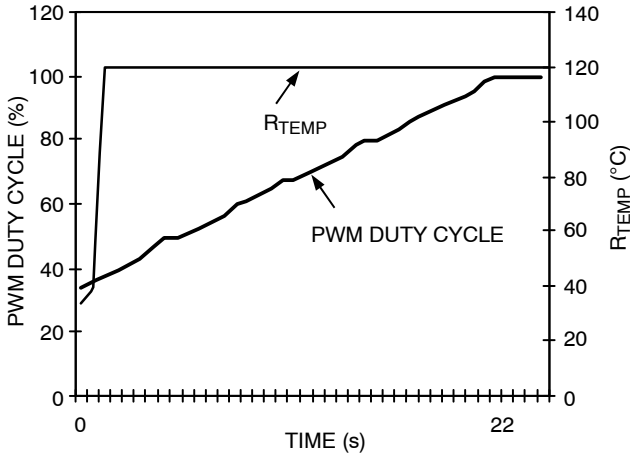



**Figure 30. Filtered Mode with Ramp Rate = 4**

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.

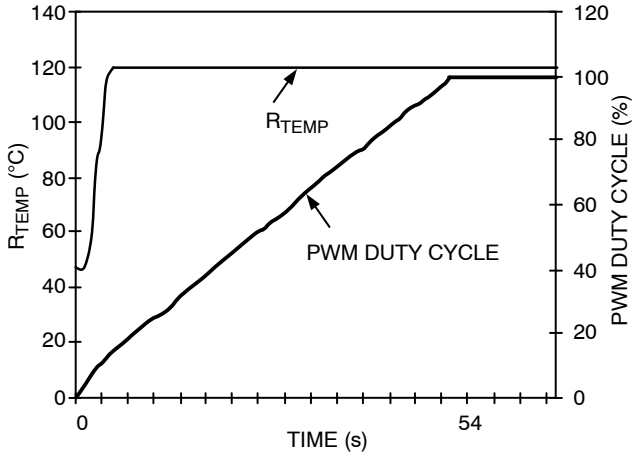



**Figure 31. Filtered Mode with Ramp Rate = 2**

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.

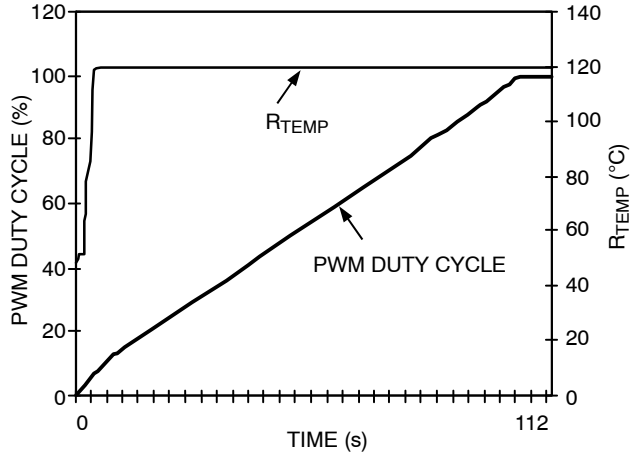



**Figure 32. Filtered Mode with Ramp Rate = 1**

As can be seen from Figures 29 through 32, the rate at which the fan will react to temperature change is dependent on the ramp rate selected in the Fan Filter Register. The higher the ramp rate, the faster the fan will reach the newly calculated fan speed.

Figure 33 shows the behavior of the PWM output as temperature varies. As the temperature is rising, the fan speed will ramp up. Small drops in temperature will not affect the ramp-up function since the newly calculated fan speed will still be higher than the previous PWM value. The

Filtered Mode allows the PWM output to be made less sensitive to temperature variations. This will be dependent on the ramp rate selected and the ADC sample rate programmed into the Fan Filter Register.

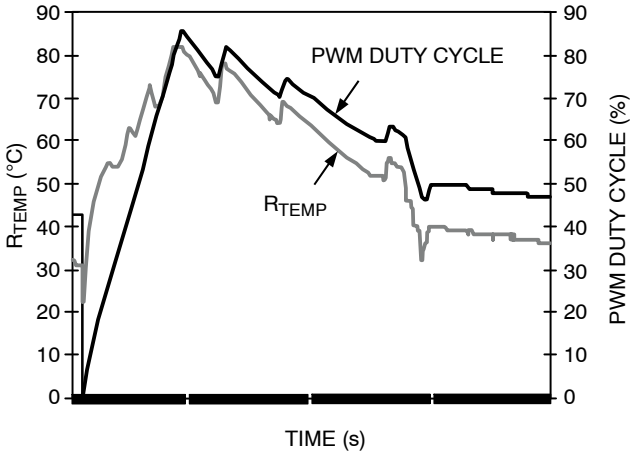



**Figure 33. How Fan Reacts to Temperature Variation in Filtered Mode**

### Effect of ADC Sample Rate on Filtered Mode

The second means by which 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 are effectively reduced. If the number of temperature measurements per second are reduced, how often the PWM_OUT signal controlling the fan is updated is also reduced.

Bits <4:2> of the Fan Filter Register (Reg 0x23) 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 Hz | 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 0x23 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> Unused. 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, i.e., 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 the fan.
9. Select Automatic Fan Speed Control Mode by setting Bit 7 of Configuration Register 1.

## PWM Duty Cycle Select Mode

The ADM1030 may be operated under software control by clearing Bit 7 of Configuration Register 1 (Register 0x00). This allows the user to directly control PWM Duty Cycle.

Clearing Bit 5 of Configuration Register 1 allows fan control by varying PWM duty cycle. Values of duty cycle between 0% to 100% may be written to the Fan Speed Config Register (0x22) to control the speed of the 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% |

## 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 may be necessary to specify fans with a nominal fan speed of 6250 RPM. However, many of these fans will 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 will allow all of these fans to be programmed to run at the desired RPM value.

Clearing Bit 7 of Configuration Register 1 (Reg 0x00) to 0 places the ADM1030 under software control. Once under software control, the device may be placed in to RPM Feedback Mode by writing to Bit 5 of Configuration Register 1. Writing a 1 to Bit 5 selects RPM Feedback Mode for the fan. Once RPM Feedback Mode has been selected, the required fan RPM may be written to the Fan Tach High Limit Register (0x10). The RPM Feedback Mode function allows a fan RPM value to be programmed into the device, and the ADM1030 will maintain 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 will slow the fan down to maintain the correct RPM speed. The value to be programmed into each Fan Tach High Limit Register is given by:

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

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 will be measured when the count value reaches 255.

Since speed range, N, = 2,

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

### 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:

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

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, what value needs to be programmed for Count?

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

Since the desired RPM value, $R$ is 5000 RPM, the value for Count is:

N = 2:

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

***Example 2:***

If the desired value for RPM Feedback Mode is 3650 RPM, what value needs to be programmed for Count?

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

Since the desired RPM value, $R$ is 3650 RPM, the value for Count is:

N = 2:

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

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 will indicate a fan failure condition is whenever the count reaches 255. Since the speed range N = 2, the fan will fail 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 Count Equation to calculate Count Value.
4. Clear Bit 7 of Configuration Register 1 (Reg. 0x00) to place the ADM1030 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 (Reg. 0x10). The fan speed will now go to the desired RPM value and maintain 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 will 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 Bits 5 and 7 of Configuration Register 1 (Reg 0x00) to enter PWM duty cycle mode. Write 0x08 to the Fan Speed Config Register (Reg 0x22) to set the PWM output to 53% duty cycle.
2. Measure the fan RPM. This represents the fan RPM below which the RPM mode will fail 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.

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 ADM1030 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 a 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 will need 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 may be driven using PWM control.

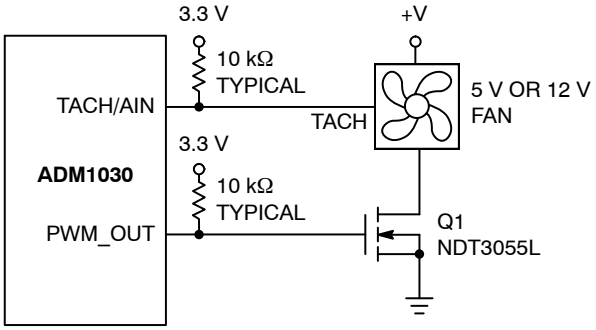



**Figure 34. Interfacing the ADM1030 to a 3-wire Fan**

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 may be substituted based on the system's fan drive requirements.

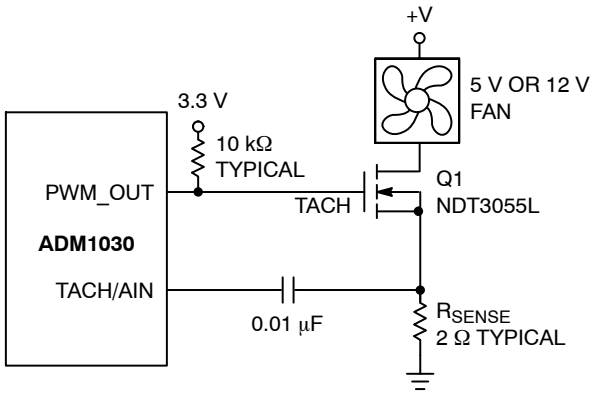



**Figure 35. Interfacing the ADM1030 to a 2-wire Fan**

Figure 35 shows how a 2-wire fan may be connected to the ADM1030. This circuit allows the speed of the 2-wire fan to be measured even though the fan has no dedicated Tach signal. A series RSENSE resistor in the fan circuit converts the fan commutation pulses into a voltage. This is ac-coupled into the ADM1030 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, $R_{SENSE}$ may be reduced. The smaller $R_{SENSE}$ is the better, since more voltage will be developed across the fan, and the fan will spin faster. 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 will be 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.

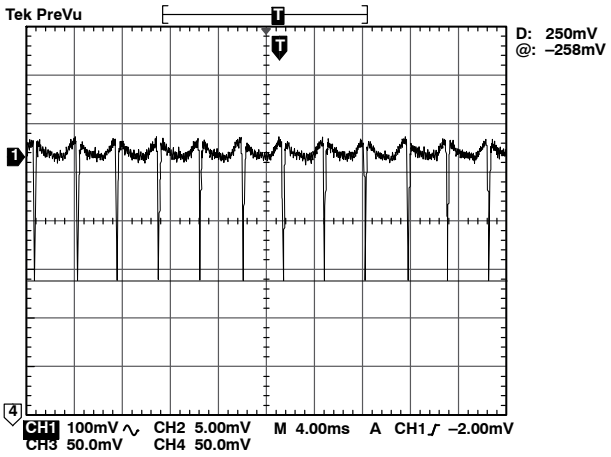



**Figure 36. Fan Speed Sensing Waveform at TACH/AIN Pin**

## Fan Speed Measurement

The fan counter does not count the fan tach output pulses directly, because the fan speed may 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 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 will repeat until monitoring is disabled. The fan speed measurement is stored in the Fan Speed Reading register at address 0x08.

The fan speed count is given by:

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

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 Bits 7 and 6 of Fan Characteristics Register 1 (20h) 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 |

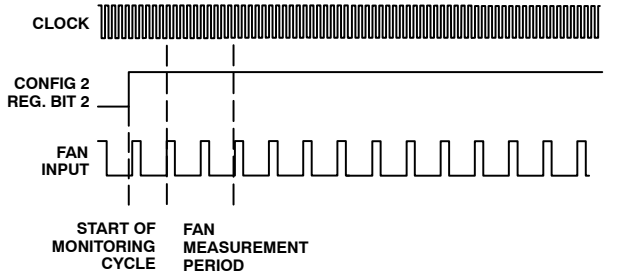



**Figure 37. Fan Speed Measurement**

In situations where different output drive circuits are used for fan drive, it may be desirable to invert the PWM drive

signal. Setting Bit 3 of Configuration Register 1 (0x00) 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 (0x00) 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 ADM1030 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 will be 0xFF for a failed fan. Therefore, we should make the Fan Tach High Limit = 0xFE 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, say 50%, 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 already failed its fan speed measurement twice previously. 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 0x02) will also be 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 2 seconds to restart the fan. Subsequent fan failures cause $\overline{\text{INT}}$ to be reasserted and the PWM_OUT signal is brought high for 2 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 will continue 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 will be negated, and the fan will return to its normal operating speed.

Figure 39 shows a typical application circuit for the ADM1030. Temperature monitoring can be based around a CPU diode or discrete transistor measuring thermal hotspots. Either 2- or 3-wire fans may be monitored by the ADM1030, as shown.

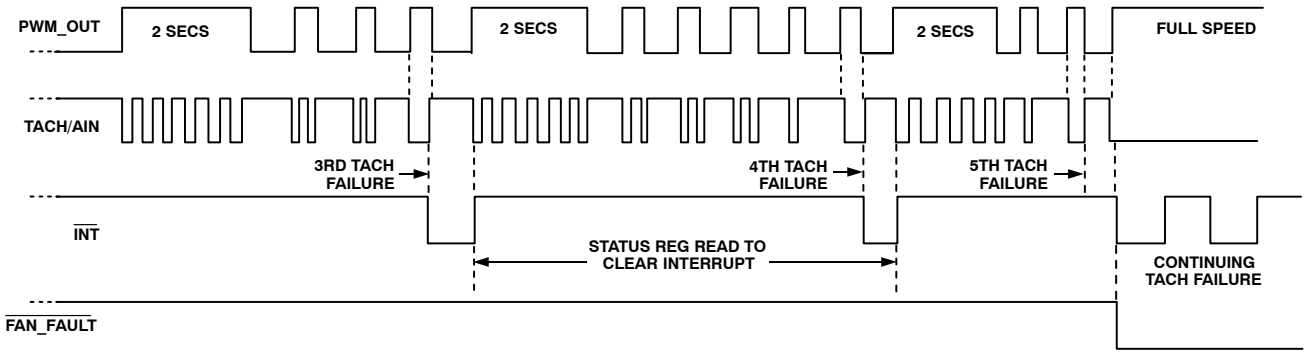



**Figure 38. Operation of $\overline{\text{FAN_FAULT}}$ and Interrupt Pins**

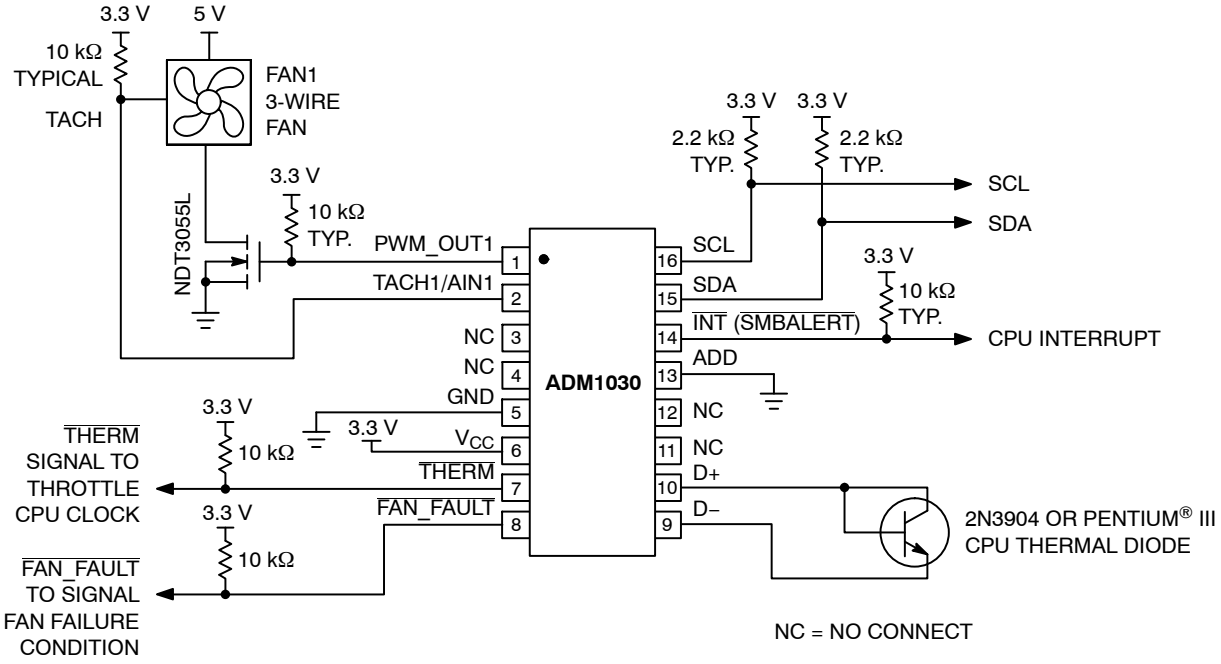



**Figure 39. Typical Application Circuit**

**Table 16. REGISTERS**

| Register Name | Address A7–A0 in Hex | Comments |
|---|---|---|
| Value Registers | 0x06–0x1A | See Table 17. |
| Device ID Register | 0x3D | This location contains the device identification number. Since this device is the ADM1030, this register contains 0x30. This register is read only. |
| Company ID | 0x3E | This location contains the company identification number (0x41). This register is read only. |
| $\overline{\text{THERM}}$ 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 28. |
| 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 Speed Configuration Register | 0x22 | See Table 24. Power-on value = 0101 0101. |
| Fan Filter Register | 0x23 | See Table 25. Power-on value = 0101 0101. |
| Local Temperature $T_{MIN}/T_{RANGE}$ | 0x24 | See Table 26. Power-on value = 0100 0001. |
| Remote Temperature $T_{MIN}/T_{RANGE}$ | 0x25 | See Table 27. Power-on value = 0110 0001. |

**Table 17. VALUE AND LIMIT REGISTERS**

| Address | Read/Write | Description |
|---|---|---|
| 0x06 | Read/Only | Extended Temperature Resolution (see Table 22). |
| 0x08 | Read/Write | Fan Speed Reading – this register contains the fan speed tach measurement. |
| 0x0A | Read/Only | Local Temperature Value – this register contains the 8 MSBs of the local temperature measurement. |
| 0x0B | Read/Only | Remote Temperature Value – this register contains the 8 MSBs of the remote temperature reading. |
| 0x0D | Read/Write | Local Temperature Offset – See Table 29. |
| 0x0E | Read/Write | Remote Temperature Offset – See Table 30. |
| 0x10 | Read/Write | Fan Tach High Limit – this register contains the limit for the fan tach measurement. Since the tach circuit counts between pulses, a slow fan will result in a large measured value, so exceeding the limit by one is the way to detect a slow or stalled fan. (Power-on Default = FFh) |
| 0x14 | Read/Write | Local Temperature High Limit (Power-on Default 60rC). |
| 0x15 | Read/Write | Local Temperature Low Limit (Power-on Default 0rC). |
| 0x16 | Read/Write | Local Temperature Therm Limit (Power-on Default 70rC). |
| 0x18 | Read/Write | Remote Temperature High Limit (Power-on Default 80rC). |
| 0x19 | Read/Write | Remote Temperature Low Limit (Power-on Default 0rC). |
| 0x1A | Read/Write | Remote Temperature Therm Limit (Power-on Default 100rC). |

**Table 18. REGISTER 0X00 CONFIGURATION REGISTER 1 POWER-ON DEFAULT 90H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| 0 | MONITOR | Read/Write | Setting this bit to a “1” enables monitoring of temperature and enables measurement of the fan tach signals. (Power-up Default = 0) |
| 1 | $\overline{INT}$ Enable | Read/Write | Setting this bit to a “1” enables the $\overline{INT}$ output. 1 = Enabled 0 = Disabled (Power-up Default = 0) |
| 2 | TACH/AIN | Read/Write | Clearing this bit to “0” selects digital fan speed measurement via the TACH pins. Setting this bit to “1” configures the TACH pins as analog inputs that can measure the speed of 2-wire fans via a sense resistor. (Power-up Default = 0) |
| 3 | PWM Invert | Read/Write | Setting this bit to “1” inverts the PWM signal on the output pin. (Power-up Default = 0). The power-up default makes the PWM_OUT pin go low for 100% duty cycle (suitable for driving the fan using a PMOS device). Setting this bit to “1” makes the PWM_OUT pin high for 100% duty cycle (intended for driving the fan using an NMOS device). |
| 4 | Fan Fault Enable | Read/Write | Logic 1 enables $\overline{FAN_FAULT}$ pin; Logic 0 disables $\overline{FAN_FAULT}$ output. (Power-up Default = 1.) |
| 6–5 | PWM Mode | Read/Write | These two bits control the behavior of the fan in Auto Fan Speed Control Mode.<br>00 = Remote Temp controls Fan. (Program PWM duty cycle in Software Mode.)<br>11 = Fastest Calculated Speed Controls Fan. (Program RPM speed in Software Mode.) |
| 7 | Auto/SW Ctrl | Read/Write | Logic 1 selects Automatic Fan Speed Control; Logic 0 selects SW control. (Power-up Default = 1) When under software control, PWM duty cycle or RPM values may be programmed for the fan. |

**Table 19. REGISTER 0X01 CONFIGURATION 2 POWER-ON DEFAULT = 7FH**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| 0 | PWM 1 En | Read/Write | Enables fan PWM output when this bit is a “1.” |
| 1 | Unused | Read/Write | Unused |
| 2 | TACH 1 En | Read/Write | Enables Tach input when set to “1.” |
| 3 | Unused | | |
| 4 | Loc Temp En | Read/Write | Enables Interrupts on Local Channel when set to “1.” |
| 5 | Remote Temp En | Read/Write | Enables Interrupts on Remote Channel when set to “1.” Default is normally enabled, except when a diode fault is detected on power-up. |
| 6 | Unused | Read/Write | Unused |
| 7 | SW Reset | Read/Write | When set to “1,” resets the device. Self-clears. Power-up Default = 0. |

**Table 20. REGISTER 0X02 STATUS REGISTER 1 POWER-ON DEFAULT = 00H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| 0 | Alarm Speed | Read Only | This bit is set to “1” when fan is running at alarm speed. Once read, this bit will not reassert on next monitoring cycle, even if the fan is still running at alarm speed. This gives an indication as to when the fan is running full-speed, such as in a $\overline{THERM}$ condition. |
| 1 | Fan Fault | Read Only | This bit is set to “1” if fan becomes stuck or is running under speed. Once read, this bit will reassert on next monitoring cycle, if the fan failure condition persists. |
| 2 | Remote Temp High | Read Only | “1” indicates Remote high temperature limit has been exceeded. If the temperature is still outside the Remote Temp High Limit, this bit will reassert on next monitoring cycle. |
| 3 | Remote Temp Low | Read Only | “1” indicates Remote low temperature limit exceeded (below). If the temperature is still outside the Remote Temp Low Limit, this bit will reassert on next monitoring cycle. |
| 4 | Remote Temp Therm | Read Only | “1” indicates Remote temperature Therm limit has been exceeded. This bit is cleared on a read of Status Register 1. Once cleared, this bit will not get reasserted even if the $\overline{THERM}$ condition persists. |
| 5 | Remote Diode Error | Read Only | This bit is set to “1” if a short or open is detected on the remote temperature channel. This test is only done on power-up, and if set to 1 cannot be cleared by reading the Status Register 1. |
| 6 | Loc Temp High | Read Only | “1” indicates Local Temp High Limit has been exceeded. If the temperature is still outside the Local Temp High Limit, this bit will reassert on next monitoring cycle. |
| 7 | Loc Temp Low | Read Only | “1” indicates Local Temp Low Limit has been exceeded (below). If the temperature is still outside the Local Temp Low Limit, this bit will reassert on next monitoring cycle. |

**Table 21. REGISTER 0X03 STATUS REGISTER 2 POWER-UP DEFAULT = 00H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| 0 | Unused | Read Only | Unused |
| 1 | Unused | Read Only | Unused |
| 2 | Unused | Read Only | Unused |
| 3 | Unused | Read Only | Unused |
| 4 | Unused | Read Only | Unused |
| 5 | Unused | Read Only | Unused |
| 6 | Loc Therm | Read Only | “1” indicates Local temperature Therm limit has been exceeded. This bit clears on a read of Status Register 2. Once cleared, this bit will not be reasserted even if the $\overline{THERM}$ condition persists. |
| 7 | $\overline{THERM}$ | Read Only | Set to “1” when $\overline{THERM}$ is pulled low as an input. This bit clears on a read of Status Register 2. The fan also runs full-speed. |

**Table 22. REGISTER 0X06 EXTENDED TEMPERATURE RESOLUTION POWER-ON DEFAULT = 00H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <2:0> | Remote Temp | Read Only | Holds extended temperature resolution bits for Remote Temperature channel. |
| <5:3> | Reserved | Read Only | Reserved |
| <7:6> | Local Temp | Read Only | Holds extended temperature resolution bits for Local Temperature channel. |

**Table 23. REGISTER 0X20 FAN CHARACTERISTICS REGISTER 1 POWER-ON DEFAULT = 5DH**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <2:0> | Fan 1 Spin-up | Read/Write | These bits contain the Fan Spin-up time to allow the fan to overcome its own inertia.<br>000 = 200 ms<br>001 = 400 ms<br>010 = 600 ms<br>011 = 800 ms<br>100 = 1 sec<br>101 = 2 secs (Default)<br>110 = 4 secs<br>111 = 8 secs |
| <5:3> | PWM 1 Frequency | Read/Write | These bits allow programmability of the nominal PWM output frequency driving the fan (Default = 31 Hz).<br>000 = 11.7 Hz<br>001 = 15.6 Hz<br>010 = 23.4 Hz<br>011 = 31.25 Hz (Default)<br>100 = 37.5 Hz<br>101 = 46.9 Hz<br>110 = 62.5 Hz<br>111 = 93.5 Hz |
| <7:6> | Speed Range | Read/Write | These bits contain the Speed Range, N.<br>00 = 1 (Fail Speed = 2647 RPM)<br>01 = 2 (Fail Speed = 1324 RPM)<br>10 = 4 (Fail Speed = 662 RPM)<br>11 = 8 (Fail Speed = 331 RPM) |

**Table 24. REGISTER 0X22 FAN SPEED CONFIG REGISTER POWER-ON DEFAULT = 05H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <3:0> | Normal/Min Spd 1 | Read/Write | This nibble contains the normal speed value for the fan. When in Automatic Fan Speed Control Mode, this nibble contains the minimum speed at which the fan will run. Default is 0x05 for 33% PWM duty cycle. (See Table 11.) |
| <7:4> | Unused | | Unused |

**Table 25. REGISTER 0X23 FAN FILTER REGISTER POWER-ON DEFAULT = 50H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7> | Spin-up Disable | Read/Write | When set to 1, disables fan spin-up. |
| <6:5> | Ramp Rate | Read/Write | These bits set the ramp rate for the PWM output.<br>00 = 1<br>01 = 2<br>10 = 4<br>11 = 8 |
| <4:2> | ADC Sample Rate | Read/Write | These bits set the sampling rate for the ADC.<br>000 = 87.5 Hz 0.0625 Updates/sec<br>001 = 175 Hz 0.125 Updates/sec<br>010 = 350 Hz 0.25 Updates/sec<br>011 = 700 Hz 0.5 Updates/sec<br>100 = 1.4 kHz (Default) 1 Update/sec<br>101 = 2.8 kHz 2 Updates/sec<br>110 = 5.6 kHz 4 Updates/sec<br>111 = 11.2 kHz 8 Updates/sec |
| <1> | Unused | Read/Write | Unused |
| <0> | Fan Filter En | Read/Write | Setting this bit to 1 enables filtering of the PWM_OUT signal. |

**Table 26. REGISTER 0X24 LOCAL TEMP $T_{MIN}$/$T_{RANGE}$ POWER-ON DEFAULT = 41H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7:3> | Local Temp $T_{MIN}$ | Read/Write | Contains the minimum temperature value for Automatic Fan Speed Control based on Local Temperature Readings. $T_{MIN}$ can be programmed to positive values only in 4°C increments. Default is 32°C.<br>00000 = 0°C<br>00001 = 4°C<br>00010 = 8°C<br>00011 = 12°C<br>⋮<br>01000 = 32°C (Default)<br>⋮<br>11110 = 120°C<br>11111 = 124°C |
| <2:0> | Local Temp $T_{RANGE}$ | Read/Write | This nibble contains the temperature range value for Automatic Fan Speed Control based on the Local Temperature Readings.<br>000 = 5°C<br>001 = 10°C (Default)<br>010 = 20°C<br>011 = 40°C<br>100 = 80°C |

**Table 27. REGISTER 0X25 REMOTE TEMP $T_{MIN}$/$T_{RANGE}$ POWER-ON DEFAULT = 61H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7:3> | Remote Temp $T_{MIN}$ | Read/Write | Contains the minimum temperature value for Automatic Fan Speed Control based on Remote Temperature Readings. $T_{MIN}$ can be programmed to positive values only in 4°C increments. Default is 48°C.<br>00000 = 0°C<br>00001 = 4°C<br>00010 = 8°C<br>00011 = 12°C<br>⋮<br>01100 = 48°C (Default)<br>⋮<br>11110 = 120°C<br>11111 = 124°C |
| <2:0> | Remote Temp $T_{RANGE}$ | Read/Write | This nibble contains the temperature range value for Automatic Fan Speed Control based on the Remote 1 Temperature Readings.<br>000 = 5°C<br>001 = 10°C (Default)<br>010 = 20°C<br>011 = 40°C<br>100 = 80°C |

**Table 28. REGISTER 0X3F THERM BEHAVIOR/REVISION POWER-ON DEFAULT = 80H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7> | Therm-to-Fan Enable | Read/Write | Setting this bit to 1, enables the fan to run full-speed when $\overline{THERM}$ is asserted low. This allows the system to be run in performance mode. Clearing this bit to 0 disables the fan from running full-speed whenever $\overline{THERM}$ is asserted low. This allows the system to run in silent mode. (Power-on Default = 1.) Note that this bit has no effect whenever $\overline{THERM}$ is pulled low as an input. |
| <6:4> | Unused | Read Only | Unused. Read back zeros. |
| <3:0> | Revision | Read Only | This nibble contains the revision number for the ADM1030. |

**Table 29. REGISTER 0X0D LOCAL TEMP OFFSET POWER-ON DEFAULT = 00H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7> | Sign | Read/Write | When this bit is 0, the local offset will be added to the Local Temperature Reading. When this bit is set to 1, the local temperature offset will be subtracted from the Local Temperature Reading. |
| <6:4> | Reserved | Read/Write | Unused. Normally read back zeros. |
| <3:0> | Local Offset | Read/Write | These four bits are used to add a two's complement offset to the Local Temperature Reading, allowing 15°C to be added to or subtracted from the temperature reading. |

**Table 30. REGISTER 0X0E REMOTE TEMP OFFSET POWER-ON DEFAULT = 00H**

| Bit | Name | R/$\overline{W}$ | Description |
|---|---|---|---|
| <7> | Sign | Read/Write | When this bit is 0, the remote offset will be added to the Remote Temperature Reading. When this bit is set to 1, the remote temperature offset will be subtracted from the Remote Temperature Reading. |
| <6:4> | Reserved | Read/Write | Unused. Normally read back zeros. |
| <3:0> | Remote Offset | Read/Write | These four bits are used to add a two's complement offset to the Remote Temperature Reading, allowing 15°C to be added to or subtracted from the temperature reading. |

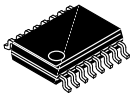
**Table 31. ORDERING INFORMATION TABLE**

| Model* | Temperature Range | Package Type | Package Option | Shipping† |
|---|---|---|---|---|
| ADM1030ARQZ−REEL | 0°C to 100°C | 16-lead QSOP | RQ−16 | 2,500 Tape & Reel |

* Z = Pb-free part.

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

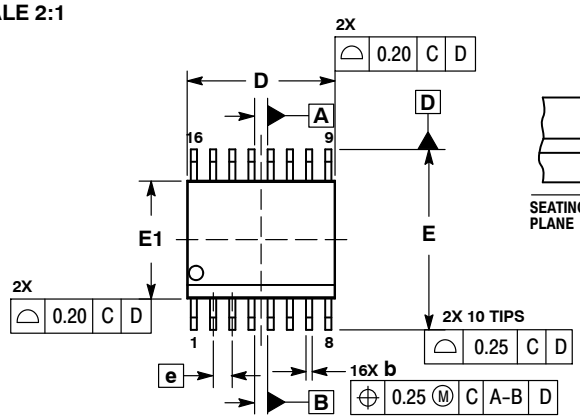
Pentium is a registered trademark of Intel Corporation.

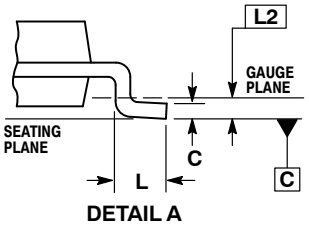


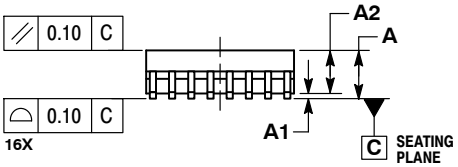
**SCALE 2:1**

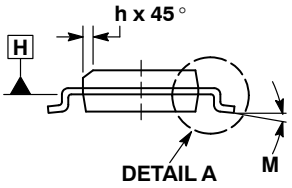
**QSOP16**
CASE 492−01
ISSUE A

DATE 23 MAR 2011



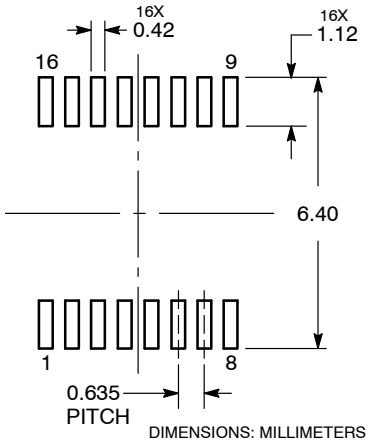




NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: MILLIMETERS.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.
4. DIMENSION D DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS, OR GATE BURRS. MOLD FLASH, PROTRUSIONS, OR GATE BURRS SHALL NOT EXCEED 0.005 PER SIDE. DIMENSION E1 DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.005 PER SIDE. D AND E1 ARE DETERMINED AT DATUM H.
5. DATUMS A AND B ARE DETERMINED AT DATUM H.

| | INCHES | | MILLIMETERS | |
|---|---|---|---|---|
| DIM | MIN | MAX | MIN | MAX |
| A | 0.053 | 0.069 | 1.35 | 1.75 |
| A1 | 0.004 | 0.010 | 0.10 | 0.25 |
| A2 | 0.049 | ---- | 1.24 | ---- |
| b | 0.008 | 0.012 | 0.20 | 0.30 |
| c | 0.007 | 0.010 | 0.19 | 0.25 |
| D | 0.193 BSC | | 4.89 BSC | |
| E | 0.237 BSC | | 6.00 BSC | |
| E1 | 0.154 BSC | | 3.90 BSC | |
| e | 0.025 BSC | | 0.635 BSC | |
| h | 0.009 | 0.020 | 0.22 | 0.50 |
| L | 0.016 | 0.050 | 0.40 | 1.27 |
| L2 | 0.010 BSC | | 0.25 BSC | |
| M | 0 ° | 8 ° | 0 ° | 8 ° |

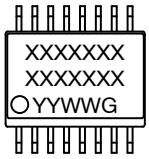
## RECOMMENDED SOLDERING FOOTPRINT*




DIMENSIONS: MILLIMETERS

*For additional information on our Pb−Free strategy and soldering details, please download the **onsemi** Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

## GENERIC MARKING DIAGRAM*




XXXXX = Specific Device Code
YY = Year
WW = Work Week
G = Pb−Free Package

*This information is generic. Please refer to device data sheet for actual part marking. Pb−Free indicator, “G” or microdot “▪”, may or may not be present. Some products may not follow the Generic Marking.

| | | |
|---|---|---|
| **DOCUMENT NUMBER:** | **98AON04472D** | Electronic versions are uncontrolled except when accessed directly from the Document Repository. Printed versions are uncontrolled except when stamped “CONTROLLED COPY” in red. |
| **DESCRIPTION:** | **QSOP16** | **PAGE 1 OF 1** |

**onsemi** and onsemi are trademarks of Semiconductor Components Industries, LLC dba **onsemi** or its subsidiaries in the United States and/or other countries. **onsemi** reserves the right to make changes without further notice to any products herein. **onsemi** makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does **onsemi** assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. **onsemi** does not convey any license under its patent rights nor the rights of others.

© Semiconductor Components Industries, LLC, 2019

**onsemi**, ONSEMI , and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "**onsemi**" or its affiliates and/or subsidiaries in the United States and/or other countries. **onsemi** owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of **onsemi**'s product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent−Marking.pdf. **onsemi** reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as−is" and **onsemi** makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does **onsemi** assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using **onsemi** products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by **onsemi**. "Typical" parameters which may be provided in **onsemi** data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. **onsemi** does not convey any license under any of its intellectual property rights nor the rights of others. **onsemi** products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use **onsemi** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **onsemi** and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that **onsemi** was negligent regarding the design or manufacture of the part. **onsemi** is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

## ADDITIONAL INFORMATION

**TECHNICAL PUBLICATIONS**:
**Technical Library:** www.onsemi.com/design/resources/technical−documentation
**onsemi Website:** www.onsemi.com

**ONLINE SUPPORT**: www.onsemi.com/support
**For additional information, please contact your local Sales Representative at**
www.onsemi.com/support/sales