ADM1030

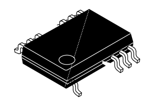
ll
o o d
Co o ll

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

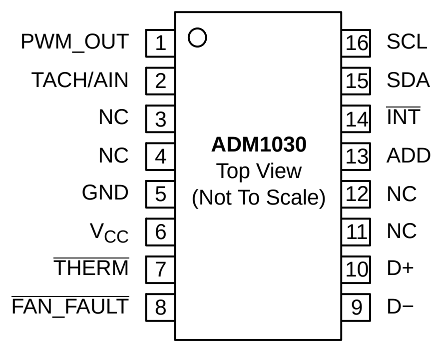
- Optimized for Pentium III: Allows Reduced Guardbanding Software and Automatic Fan Speed Control
- Automatic Fan Speed Control Allows Control Independent of CPU Intervention after Initial Setup
- Control Loop Minimizes Acoustic Noise and Battery Consumption
- Remote Temperature Measurement Accurate 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*

http://onsemi.com



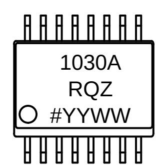
QSOP 16
CASE 492

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.

Applications

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

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

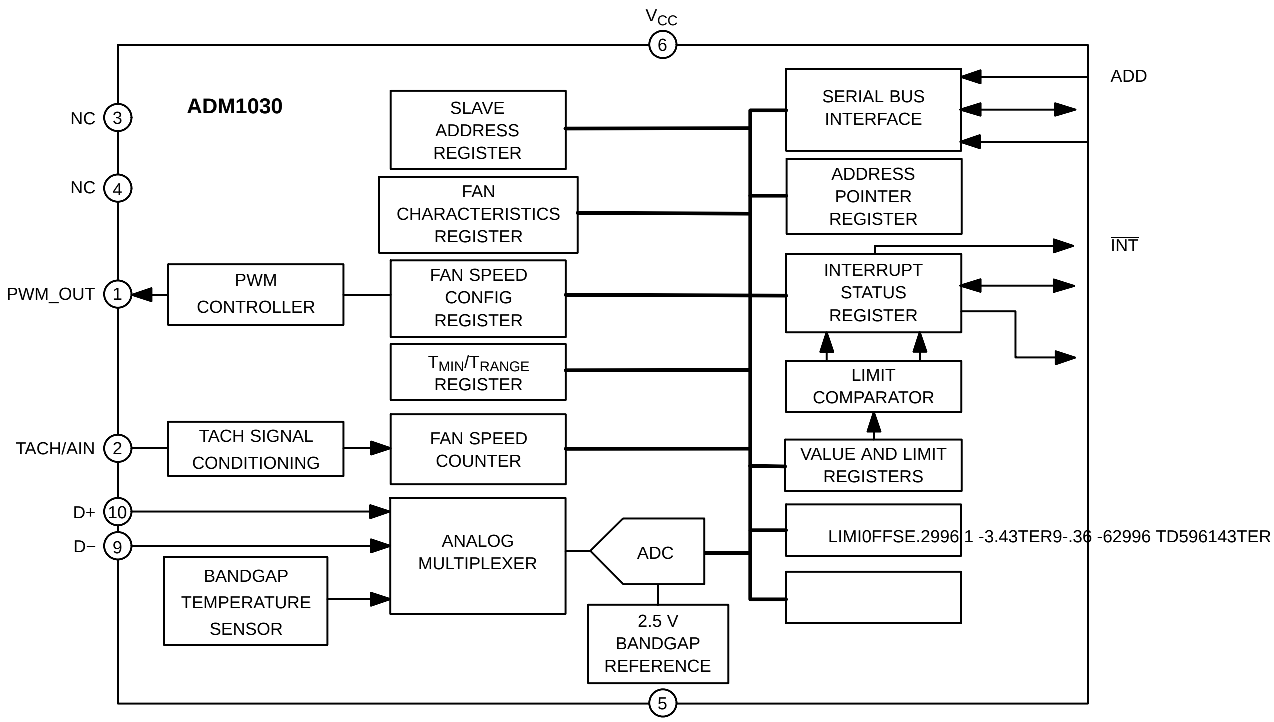


Figure 1. Functional Block Diagram

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	$\overline{THERM}$	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	$\overline{FAN\_FAULT}$	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	$\overline{INT}$	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 Standby Mode	– –	1.4 32	3.0 50	mA 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		

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

Parameter	Max	Unit

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 $R/\overline{W}$ 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 $R/\overline{W}$ bit is a 0, the master will write to the slave device. If the $R/\overline{W}$ 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

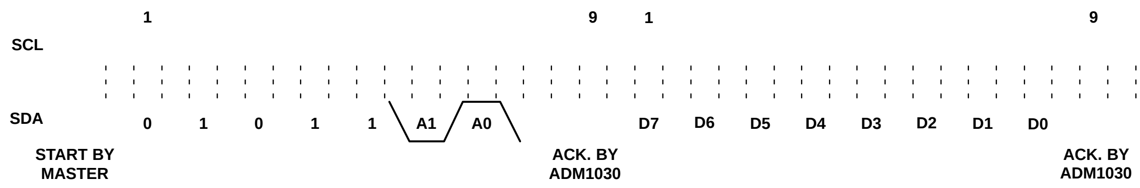


Figure 16. Writing to the Address Pointer Register Only

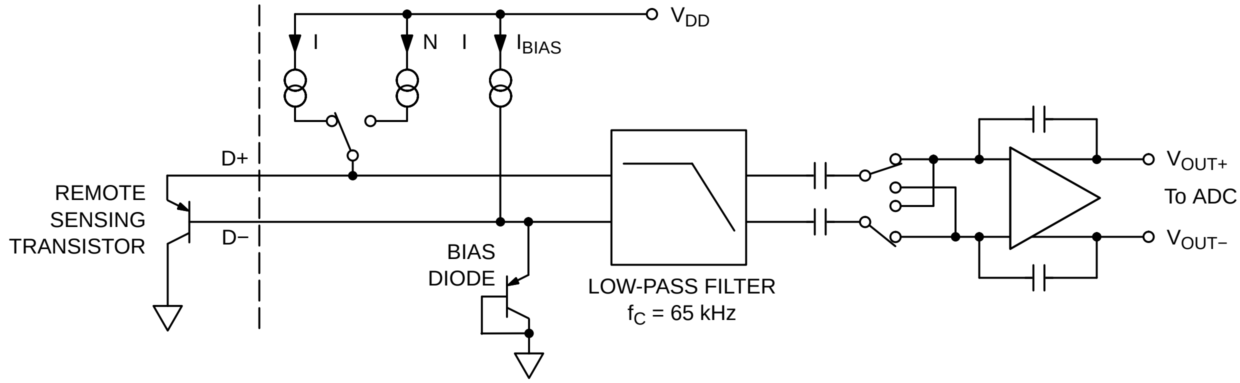


Figure 18. Signal Conditioning

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 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 ΔV_{BE}, the sensor is switched between operating currents of I and N I. The resulting waveform is passed through a 65 kHz low-pass filter to remove noise, then to a chopperstabilized amplifier that performs the functions of amplification and rectification of the waveform to produce a dc voltage proportional to ΔV_{BE}. This voltage is measured by the ADC to give a temperature output in 11-bit 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.

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 f MODE, 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.

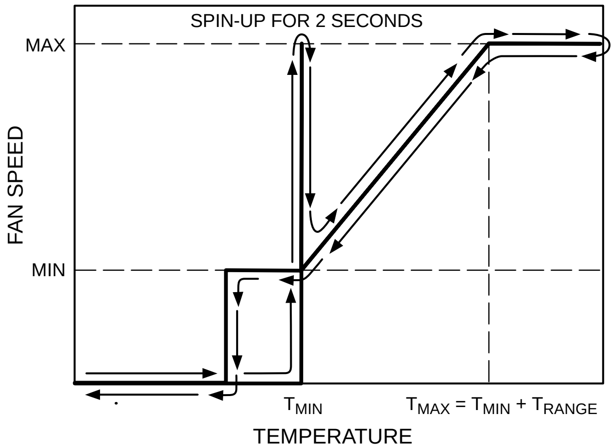


Figure 21. Automatic Fan Speed Control

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

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

Example 3

= 0 C, T_{RANGE} = 40 C
= 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).

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, 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.

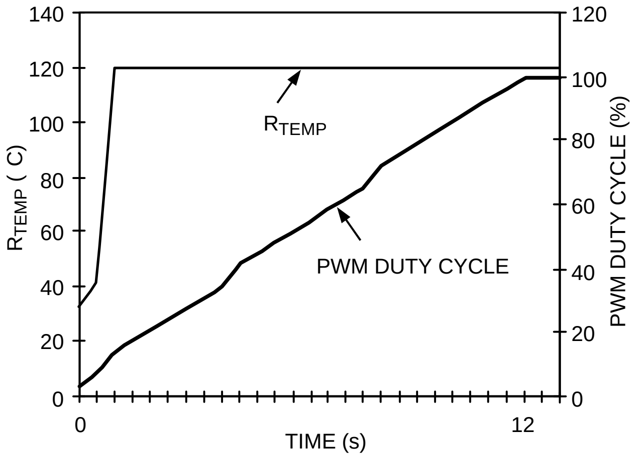


Figure 29. Filtered Mode with Ramp Rate = 8

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.

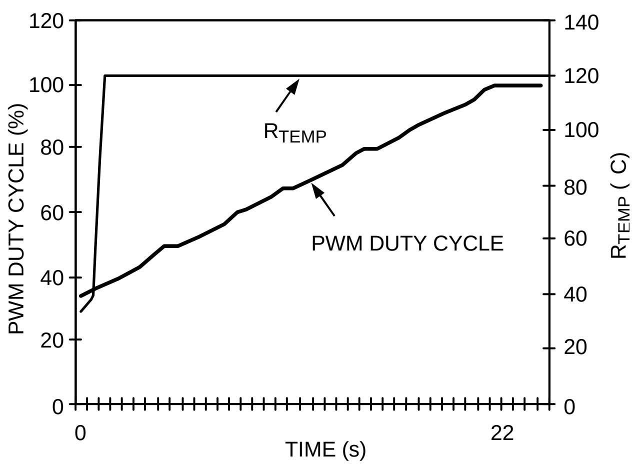


Figure 30. Filtered Mode with Ramp Rate = 4

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.

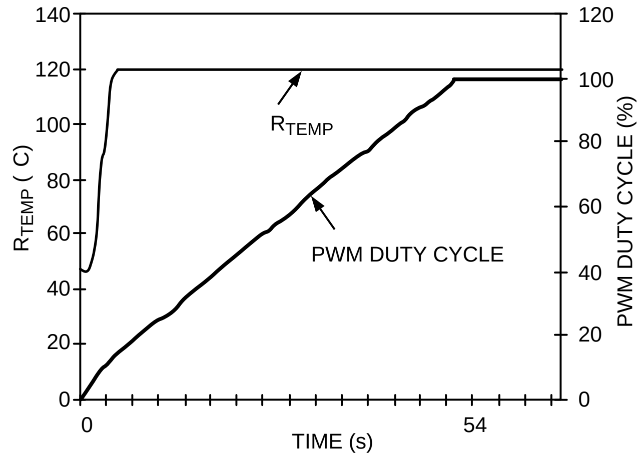


Figure 31. Filtered Mode with Ramp Rate = 2

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.

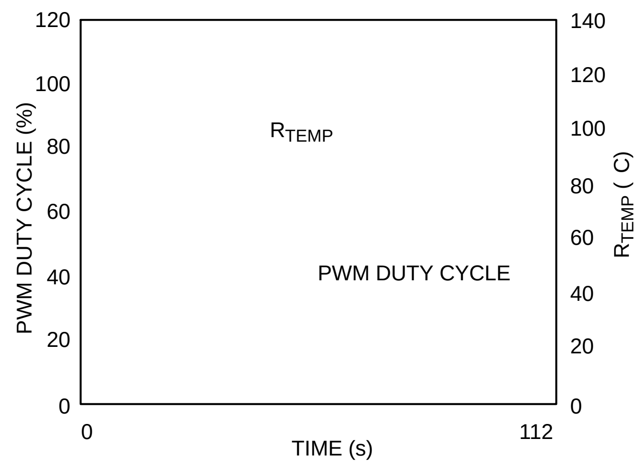


Figure 32. Filtered Mode with Ramp Rate = 1

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.

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:

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

where:

= 11.25 kHz

= desired RPM value

= Speed Range; MUST be set to 2

The speed range, , 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,

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

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:

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

where:

= 11.25 kHz

= desired RPM value

= Speed Range = 2

Example 1:

If the desired value for RPM Feedback Mode is 5000 RPM, what value needs to be programmed for Count?

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

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

N = 2:

$$\text{Count} = (11250 \times 60)/5000 \times 2$$

Count = -.001/TT3 1 T80j/F8v.tai7 0 0 -1.d/F5 1 Tf2.4441 0 TD0 Tc3 Tw (eq. 11)

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):

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

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

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

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

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

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

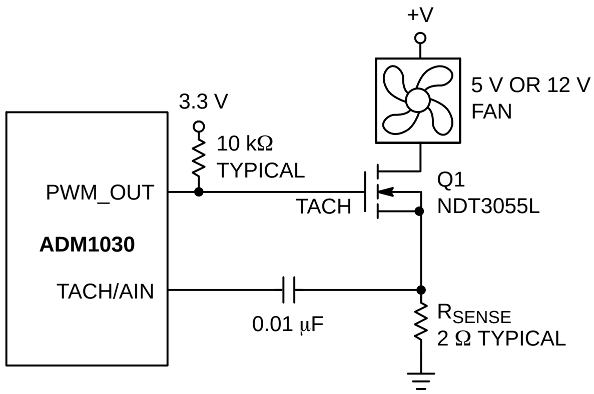
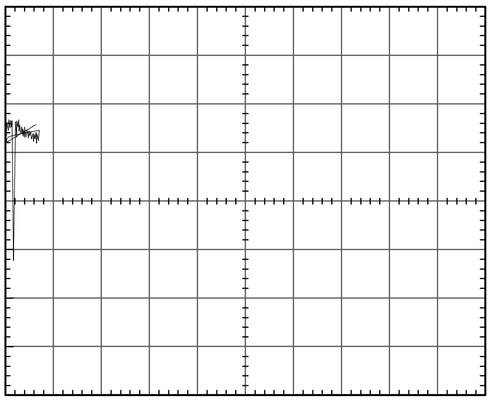


Figure 35. Interfacing the ADM1030 to a 2-wire Fan

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 $R_{SENSE} = 2\ \Omega$. 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.



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 F

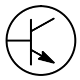


Table 17. VALUE AND LIMIT REGISTERS

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		

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 $\overline{\text{THERM}}$ 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

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

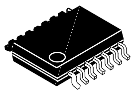
<table>
<tr><th>Bit</th><th>Name</th><th>R/$\overline{W}$</th><th>Description</th></tr>
<tr><td><7:3></td><td>Local Temp T_{MIN}</td><td>Read/Write</td><td>Contains the minimum temperature value for Automatic Fan Speed Control based on Local Temperature Readings. T_{MIN} can be programmed to positive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

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}

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

Bit

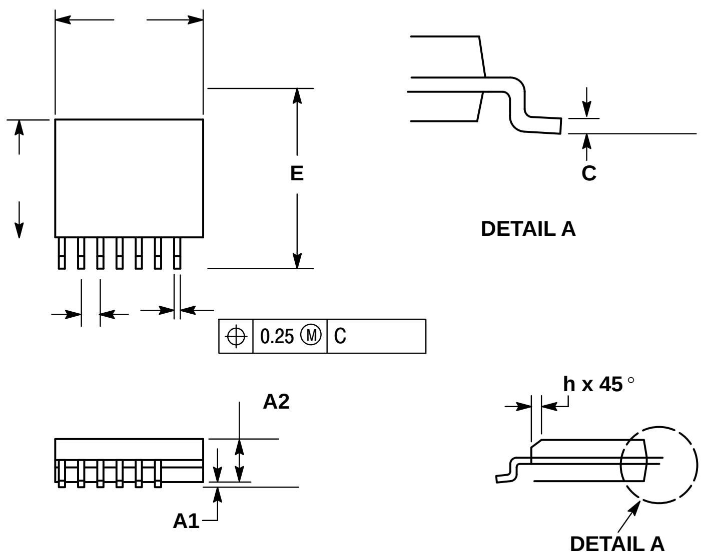


SCALE 2:1

QSOP16
CASE 492−01
ISSUE A

DATE 23 MAR 2011

NOTES:



	INCHES		
DIM	MIN	MA	
A	0.053	0.069	
A1	0.004	0.010	
	0.008	0.012	
ı	0.007	0.010	
	0.025 BSC		
	0.009	0.020	
L	0.016	0.050	
M	0	8	