ANALOG DEVICES

±1°C Local and Remote System Temperature Monitor

ADM1032*

FEATURES

On-chip and remote temperature sensing
Offset Registers for System Calibration
1°C Accuracy and Resolution on Local Channel
0.125°C Resolution/1°C Accuracy on Remote Channel
Fast (up to 32 Conversions/second)
Programmable Over/Under Temperature Limits
Programmable Fault Queue
Programmable THERM Limits
Programmable THERM Hysteresis
2-wire SMBus serial interface
Supports SMBus Alert
160µA Max Operating Current
3µA standby current
3V to 5.5V supply
Small 8-pin SO and uSO package
Optimized for PentiumIII ® - Allows Reduced Guardbanding

Applications

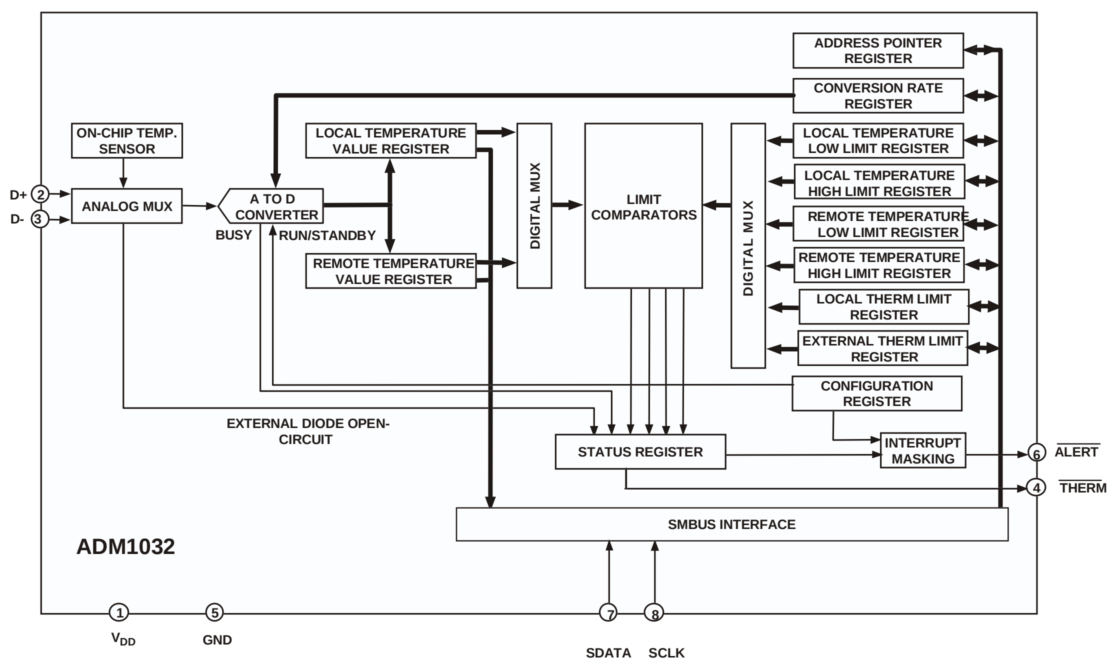
Desktop Computers
Notebook Computers
Smart Batteries
Industrial Controllers
Telecomms Equipment
Instrumentation

PRODUCT DESCRIPTION

The ADM1032 is an ACPI compliant two-channel digital thermometer and under/over temperature alarm, intended for use in personal 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. The device can measure the temperature of a microprocessor using a diode-connected NPN or PNP transistor, which may be provided on-chip in the case of the Pentium III or similar processors, or can be a low-cost discrete device such as the 2N3906. A novel measurement technique cancels out the absolute value of the transistor's base emitter voltage, so that no calibration is required. The second measurement channel measures the output of an on-chip temperature sensor, to monitor the temperature of the device and its environment.

The ADM1032 communicates over a two-wire serial interface compatible with System Management Bus (SMBus™) standards. Under and over temperature limits can be programmed into the device over the serial bus, and an ALERT output signals when the on-chip or remote temperature measurement is out of range. This output can be used as an interrupt, or as an SMBus alert. The THERM output is a comparator output that allows CPU clock throttling or on/off control of a cooling fan.

FUNCTIONAL BLOCK DIAGRAM



REV.PrL 03/2001

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

®PentiumIII is a registered trademark of Intel Corporation
*Patents Pending

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 617/329-4700 World Wide Web Site: http://www.analog.com
Fax: 617/326-8703

ADM1032–SPECIFICATIONS ($T_A = T_{MIN}$ to T_{MAX}, $V_{DD} = V_{MIN}$ to V_{MAX}, unless otherwise noted)

Parameter	Min	Typ	Max	Units	Test Conditions/Comments
POWER SUPPLY					
Supply Voltage, V_{DD}	3.0	3.30	3.6	V	See Note 1
Average Operating Supply Current, I_{CC}		120	160	µA	0.25 Conversions/Sec Rate
		3	10	µA	Standby Mode
Undervoltage Lockout Threshold	2.5	2.7	2.95	V	V_{DD} Input, Disables ADC, Rising Edge
TEMP. -TO-DIGITAL CONVERTER					
Local Sensor Accuracy		±1	±3	°C	
Resolution		1		°C	
Remote Diode Sensor Accuracy			±1	°C	$+60°C \le T_D \le +100°C$
			±3	°C	$+0°C \le T_D \le +100°C$
Resolution		0.125		°C	
Remote Sensor Source Current	120	205	300	µA	High Level, Note 1
	7	12	16	µA	Low Level, Note 1
Conversion Time		TBD		ms	From Stop Bit toConversion Complete (Both Channels)
OPEN-DRAIN DIGITAL OUTPUTS ($\overline{THERM}$, $\overline{ALERT}$)					
Output Low Voltage, V_{OL}			0.4	V	I_{OUT} = -6.0mA, V_{DD} = 3V
High Leval Output Leakage Current, I_{OH}		0.1	1	µA	$V_{OUT} = V_{DD}$
SMBUS INTERFACE					
Logic Input High Voltage, V_{IH} SCLK, SDATA	2.1			V	V_{DD} = 3 V to 5.5 V
Logic Input Low Voltage, V_{IL}			0.8	V	V_{DD} = 3 V to 5.5 V
Hysteresis SCLK, SDATA		500		mV	
SMBus Output Low Sink Current	6			mA	SDATA Forced to 0.6 V
$\overline{ALERT}$ Output Low Sink Current	1			mA	$\overline{ALERT}$ Forced to 0.4 V
Logic Input Current, I_{IH}, I_{IL}	–1		+1	µA	
SMBus Input Capacitance, SCLK, SDATA		5		pF	
SMBus Clock Frequency			100	kHz	
SMBus Timeout			25	ms	Note 2
SMBus Clock Low Time, t_{LOW}	4.7			µs	t_{LOW} Between 10% Points
SMBus Clock High Time, t_{HIGH}	4			µs	t_{HIGH} Between 90% Points
SMBus Start Condition Setup Time, $t_{SU:STA}$	4.7			µs	
SMBus Start Condition Hold Time, $t_{HD:STA}$	4			µs	Time from 10% of SDATA to 90% of SCLK
SMBus Stop Condition Setup Time, $t_{SU:STO}$	4			µs	Time from 90% of SCLK to 10% of SDATA
SMBus Data valid to SCLK Rising Edge Time, $t_{SU:DAT}$	250			ns	Time for 10% or 90% of SDATA to 10% of SCLK
SMBus Data Hold Time, $t_{HD:DAT}$	300			µs	
SMBus Bus Free Time, t_{BUF}	4.7			µs	Between Start/StopCondition
SCLK Falling Edge to SDATA Valid Time, $t_{VD,DAT}$			1	µs	Master Clocking in Data
SCLK, SDATA Rise Time, t_R			1	µs	
SCLK, SDATA Fall Time, t_F			300	ns	

NOTES

1. Guaranteed by Design, not production tested.
2. The SMBus timeout is a programmable feature. By default it is not enabled. Details on how to enable it are available in the SMBus section of this datasheet.

ABSOLUTE MAXIMUM RATINGS*

Positive Supply Voltage (V_{DD}) to GND . . . . . -0.3V, 5.5V
D+ . . . . -0.3V to V_{DD} + 0.3V
D- to GND . . . . -0.3V to +0.6V
SCLK, SDATA, ALERT, ... -0.3V to +5.5V
THERM . . . . -0.3V to V_{DD} + 0.3V
Input Current, SDATA, THERM . . . . -1, +50mA
Input Current, D- . . . . ±1mA
ESD Rating, all pins (Human Body Model) . . . >2000 V
Continuous Power Dissipation
  Up to +70°C 650mW
  Derating above +70°C . . . . 6.7mW/°C
Maximum Junction Temperature ($T_J$max) . . . . . . +150°C
Storage Temperature Range . . . . –65°C to +150°C
IR Reflow Peak Temp . . . . +220°C
Lead Temp (Soldering 10s)..............................+300°C

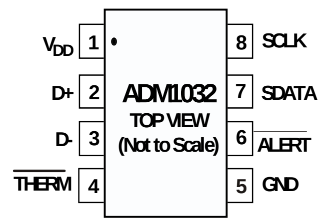
*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTICS

8-Pin SO Package:
θ_{JA} = 150°C/Watt

8-Pin µSOIC Package:
θ_{JA} = 150°C/Watt

PIN CONFIGURATION



ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option	Brand Info	SMBus Addr
ADM1032AR	0°C to +120°C	8-Pin SO Package	SO-8	1032AR	4C
ADM1032ARM	0°C to +120°C	8-Pin µSOIC Package	RM-8	T2A	4C

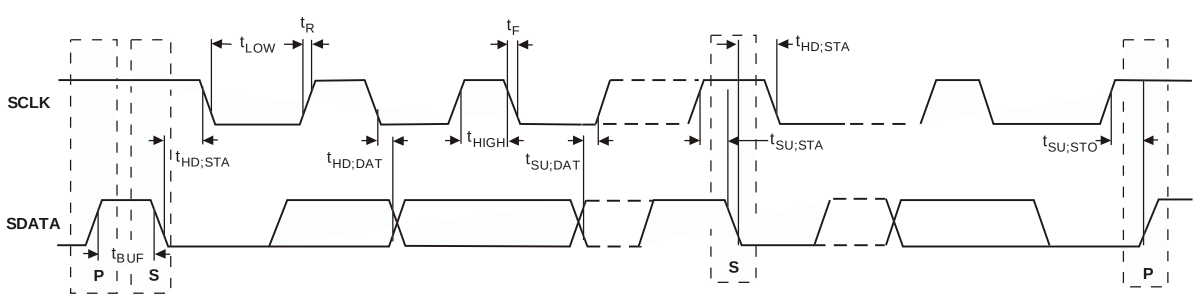


Figure 1. Diagram for Serial Bus Timing

PIN FUNCTION DESCRIPTION

PIN NO.	MNEMONIC	DESCRIPTION
1	V_{DD}	Positive supply, +3V to +3.6V
2	D+	Positive connection to remote temperature sensor
3	D-	Negative connection to remote temperature sensor
4	$\overline{THERM}$	Open-drain output which can be used to turn a fan on/off or throttle a CPU clock in the event of an over-temperature condition. Requires min 2.2kΩ pullup to V_{DD}
5	GND	Supply 0V connection
6	$\overline{ALERT}$	Open-drain logic output used as interrupt or SMBus alert
7	SDATA	Logic input/output, SMBus serial data. Open-drain output. Requires min 2.2kΩ pull-up resistor
8	SCLK	Logic input, SMBus serial clock. Requires min 2.2kΩ pull-up resistor

Typical Performance Characteristics ADM1032

Figure 2. Temperature Error vs. PC Board Track Resistance

Figure 3. Temperature Error vs. Power Supply Noise Frequency

Figure 4. Temperature Error vs. Common Mode Noise Frequency

Figure 5. Temperature Error vs. Pentium III Error

Figure 6. Temperature Error vs. Capacitance between D+ and D-

Figure 7. Standby Supply Current vs. Clock Frequency

Figure 8. Temperature Error vs. Differential-Mode Noise Frequency

Figure 9. Operating Supply Current vs. Conversion Rate

Figure 10. Standby Supply Current vs. Supply Voltage

Figure 11. Response to Thermal Shock

FUNCTIONAL DESCRIPTION

The ADM1032 contains a two-channel A to D converter with special input signal conditioning to enable operation with remote and on-chip diode temperature sensors. When the ADM1032 is operating normally, the A to D converter operates in a free-running mode. The analog input multiplexer alternately selects either the on-chip temperature sensor to measure its local temperature, or the remote temperature sensor. These signals are digitised by the ADC and the results stored in the Local and Remote Temperature Value Registers.

The measurement results are compared with Local and Remote, High, Low and $\overline{\text{THERM}}$ Temperature Limits, stored in nine on-chip registers. Out of limit comparisons generate flags that are stored in the Status Register, and one or more out-of limit results will cause the $\overline{\text{ALERT}}$ output to pull low. Exceeding $\overline{\text{THERM}}$ temperature limits cause the $\overline{\text{THERM}}$ output to assert low.

The limit registers can be programmed, and the device controlled and configured, via the serial System Management Bus (SMBus). The contents of any register can also be read back via the SMBus.

Control and configuration functions consist of:

- switching the device between normal operation and standby mode
- masking or enabling the $\overline{\text{ALERT}}$ output
- selecting the conversion rate

MEASUREMENT METHOD

A simple method of measuring temperature is to exploit the negative temperature coefficient of a diode, or the base-emitter voltage of a transistor, operated at constant current. Unfortunately, this technique requires calibration to null out the effect of the absolute value of V_{be}, which varies from device to device.

The technique used in the ADM1032 is to measure the change in V_{be} when the device is operated at two different currents.

This is given by: $\Delta V_{be} = KT/q \times \ln(N)$

where:

K is Boltzmann's constant

q is charge on the electron (1.6×10^{-19} Coulombs)

T is absolute temperature in Kelvins

N is ratio of the two currents

Figure 2 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. If a discrete transistor is used, the collector will not be grounded, and should be linked to the base. 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 operating in a noisy environment, C1 may optionally be added as a noise filter. Its value is typically 2200pF but should be no more than 3000pF. See the section on layout considerations for more information on C1.

To measure ΔV_{be}, the sensor is switched between operating currents of I and N x I. The resulting waveform is passed through a 65kHz lowpass filter to remove noise, thence to a chopper-stabilized amplifier that performs the functions of amplification and rectification of the waveform to produce a DC voltage proportional to ΔV_{be}. This voltage is measured by the ADC to give a temperature output in two's complement format. To further reduce the effects of noise, digital filtering is performed by averaging the results of 16 measurement cycles.

Signal conditioning and measurement of the internal temperature sensor is performed in a similar manner.

TEMPERATURE DATA FORMAT

One LSB of the ADC corresponds to 0.125°C, so the ADC can measure from 0°C to +127.875°C. The temperature data format is shown in Tables I and II.

Note: The ADM1032 differs from the ADM1021 in that the temperature resolution of the remote channel is improved from 1°C to 0.125°C, but it cannot measure

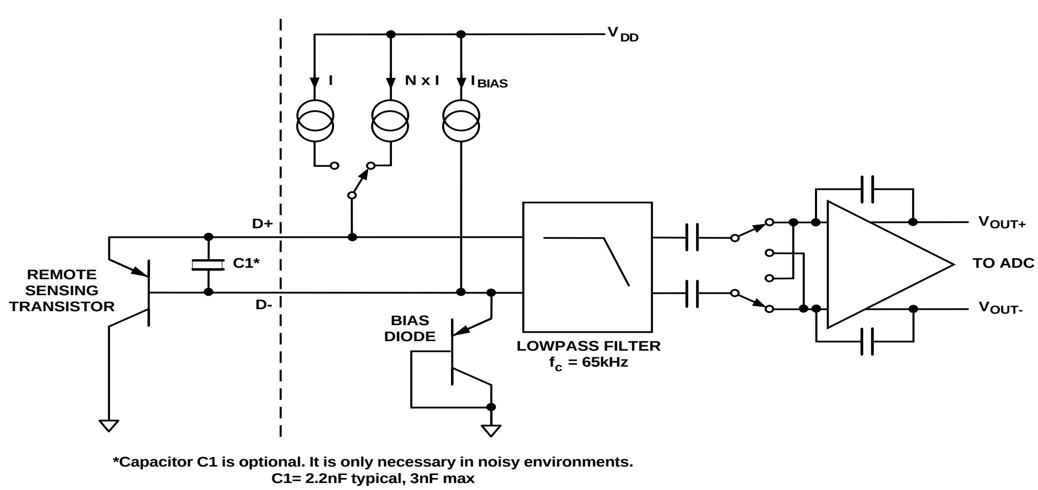


Figure 12. Input Signal Conditioning

temperatures below 0ºC. If negative temperature measurement is required then the ADM1021 should be used.

The results of the local and remote temperature measurements are stored in the Local and Remote temperature value registers, and are compared with limits programmed into the Local and Remote High and Low Limit Registers.

TABLE 1. Temperature Data Format (Local Temp. And Remote Temp. High Byte)

Temperature	Digital Output
0°C	0 000 0000
+1°C	0 000 0001
+10°C	0 000 1010
+25°C	0 001 1001
+50°C	0 011 0010
+75°C	0 100 1011
+100°C	0 110 0100
+125°C	0 111 1101
+127°C	0 111 1111

TABLE II. Extended Temperature Resolution (Remote Temp. Low Byte)

Extended Resolution	Remote Temp. Low Byte
0.000°C	0 000 0000
0.125°C	0 010 0000
0.250°C	0 100 0000
0.375°C	0 110 0000
0.500°C	1 000 0000
0.625°C	1 010 0000
0.750°C	1 100 0000
0.875°C	1 110 0000

ADM1032 REGISTERS

The ADM1032 contains registers that are used to store the results of remote and local temperature measurements, high and low temperature limits, and to configure and control the device. A description of these registers follows, and further details are given in tables III to VII.

Address Pointer Register

The Address Pointer Register itself does not have, or require, an address, as it is the register to which the first data byte of every Write operation is written automatically. This data byte is an address pointer that sets up one of the other registers for the second byte of the Write operation, or for a subsequent read operation.

The power-on default value of the Address Pointer Register is 00h, so if a read operation is performed immediately after power on without first writing to the Address Pointer, the value of the local temperature will be returned, since its register address is 00h.

Value Registers

The ADM1032 has three registers to store the results of Local and Remote temperature measurements. These registers are written to by the ADC only and can be read over the SMBus.

Offset Register

Series resistance on the D+ and D- lines and clock noise can introduce offset errors into the remote temperature measurment. To achieve the specified accuracy on this channel these offsets must be removed.
The offset value is stored as an 11-bit, twos compliment value in registers 11h (high byte) and 12h (low byte, left justified). The value of the offset is negative if the MSB of register 11h is 1 and it is positive if the MSB of register 12h is 0. The value is added to the measured value of remote temperature.
The offset register powers up with a default value of 0°C, and will have no effect if nothing is written to them.

TABLE III. SAMPLE OFFSET REGISTER CODES

Offset Value	11h	12h
-4°C	1 111 1100	0 000 0000
-1°C	1 111 1111	0 000 0000
-0.125°C	1 111 1111	1 110 0000
0°C	0 000 0000	0 000 0000
0.125°C	0 000 0000	0 010 0000
1°C	0 000 0001	0 000 0000
4°C	0 000 0100	0 000 0000

Status Register

Bit 7 of the Status Register indicates that the ADC is busy converting when it is high. Bits 6 to 3, 1 and 0 are flags that indicate the results of the limit comparisons. Bit 2 is set when the remote sensor is open circuit.

If the local and/or remote temperature measurement is above the corresponding high temperature limit or below the corresponding low temperature limit, then one or more of these flags will be set. These 5 flags (bits are NOR'd together, so that if any of them is high, the $\overline{\text{ALERT}}$ interrupt latch will be set and the $\overline{\text{ALERT}}$ output will go low. Reading the Status Register will clear the five flag bits, provided the error conditions that caused the flags to be set have gone away. While a limit comparator is tripped due to a value register containing an out of limit measurement, or the sensor is open circuit, the corresponding flag bit cannot be reset. A flag bit can only be reset of the corresponding value register contains an in-limit measurement or the sensor is good.

The ALERT interrupt latch is not reset by reading the Status Register, but will be reset when the ALERT output has been serviced by the master reading the device address, provided the error condition has gone away and the Status Register flag bits have been reset.

When flags 1 and 0 are set the THERM output goes low to indicate that the temperature measurements are outside the programmed limits. THERM output does not need to be reset unlike the ALERT output. Once the measurements are within the limits the corresponding Status register bits are reset and the THERM output goes high.

TABLE III. STATUS REGISTER BIT ASSIGNMENTS

Bit	Name	Function
7	BUSY	1 When ADC Converting
6	LHIGH*	1 When Local High-Temp Limit Tripped
5	LLOW*	1 When Local Low-Temp Limit Tripped
4	RHIGH*	1 When Remote High-Temp Limit Tripped
3	RLOW*	1 When Remote Low-Temp Limit Tripped
2	OPEN*	1 When Remote Sensor Open-Circuit
1	RTHRM	1 When Remote Therm Limit Tripped
0	LTHRM	1 When Local Therm Limit Tripped

* These flags stay high until the status register is read or they are reset by POR.

Configuration Register

Two bits of the Configuration Register are used. If bit 6 is 0, which is the power-on default, the device is in operating mode with the ADC converting. If bit 6 is set to 1, the device is in standby mode and the ADC does not convert.

Bit 7 of the configuration register is used to mask the ALERT output. If bit 7 is 0, which is the power-on default, the ALERT output is enabled. If bit 7 is set to 1, the ALERT output is disabled.

TABLE V. CONFIG.REGISTER BIT ASSIGNMENTS

Bit	Name	Function	Power-on Default
7	MASK1	0 = ALERT Enabled 1 = ALERT Masked	0
6	RUN/STOP	0 = Run 1 = Standby	0
5-0		Reserved	0

Conversion Rate Register

The lowest 3 bits of this register are used to program the conversion rate by dividing the ADC clock by 1, 2, 4, 8, 16, 32, 64 128, 256 or 512 to give conversion times from 31ms (code 09h) to 16 seconds (code 00h). This register can be written to and read back over the SMBus. The higher five bits of this register are unused and must be set to zero. Use of slower conversion times greatly reduces the device power consumption, as shown in Table VI.

TABLE VI. CONVERSION RATE REGISTER CODES

Data	Conversion/sec	Average Supply Current mA typ. at V_{DD}=3.3V
00h	0.0625	0.17
01h	0.125	0.17
02h	0.25	0.17
03h	0.5	0.17
04h	1	0.17
05h	2	0.17
06h	4	0.35
07h	8	1.1
08h	16	1.7
09h	32	0.73
0A to FFh	Reserved	

Limit Registers

The ADM1032 has nine Limit Registers to store Local and Remote, High, Low and THERM temperature limits. These registers can be written to and read back, over the SMBus.
The high limit registers perform a > comparison while the low limit registers perform a < comparison. For example if the high limit register is programmed with 80°C, then measuring 81°C will result in an alarm condition. Exceeding either the Local or Remote THERM limit asserts THERM low. A default hysteresis value of 10°C is provided, which applies to both channels. This value may be reprogrammed to any value after power up.

One-Shot Register

The One-Shot Register is used to intitiate a single conversion and comparison cycle when the ADM1032 is in standby mode, after which the device returns to standby. This is not a data register as such and it is the write operation that causes the one-shot conversion. The data written to this address is irrelevant and is not stored.

TABLE IV. LIST OF ADM1032 REGISTERS

READ ADDRESS (HEX)	WRITE ADDRESS (HEX)	NAME	POWER-ON DEFAULT
Not Applicable	Not Applicable	Address Pointer	Undefined
00	Not Applicable	Local Temp. Value	0000 0000 (00h)
01	Not Applicable	Ext. Temp. Value High Byte	0000 0000 (00h)
02	Not Applicable	Status	Undefined
03	09	Configuration	0000 0000 (00h)
04	0A	Conversion Rate	0000 1000 (08h)
05	0B	Local Temp. High Limit	0100 0110 (55h) (85°C)
06	0C	Local Temp. Low Limit	0000 0000 (00h) (0°C)
07	0D	Ext. Temp. High Limit High Byte	0100 0110 (55h) (85°C)
08	0E	Ext. Temp. Low Limit High Byte	0000 0000 (00h) (0°C)
Not Applicable	0F	One-Shot	
10	Not Applicable	Ext. Temp. Value Low Byte	0000 0000
11	11	Ext. Temp. Offset High Byte	0000 0000
12	12	Ext. Temp. Offset Low Byte	0000 0000
13	13	Ext. Temp. High Limit Low Byte	0000 0000
14	14	Ext. Temp. Low Limit Low Byte	0000 0000
19	19	External $\overline{\text{THERM}}$ Limit	0101 0101 (55h) (85°C)
20	20	Local $\overline{\text{THERM}}$ Limit	0101 0101 (55h) (85°C)
21	21	$\overline{\text{THERM}}$ Hysteresis	0000 1010 (0Ah) (10°C)
22	22	Consecutive ALERT	0000 0001 (01h)
FE	Not Applicable	Manufacturer ID	0100 0001 (41h)
FF	Not Applicable	Die Revision Code	Undefined

NOTES

[1] Writing to address 0F causes the ADM1032 to perform a single measurement. It is not a data register as such and it does not matter what data is written to it.

Consecutive $\overline{\text{ALERT}}$ Register

This value written to this register determines how many out of limit measurements must occur before an $\overline{\text{ALERT}}$ is generated. The default value is one out of limit measurement generates an ALERT. The max value that can be chosen is 4. The purpose of this register is to allow the user to perform some filtering of the output.

Register Value	Number of "Out of Limit" measurements required
yxxx 000x	1
yxxx 001x	2
yxxx 011x	3
yxxx 111x	4

x = Don't care bit

y = SMBus timeout bit. Default = 0. See SMBus section for more information.

SERIAL BUS INTERFACE

Control of the ADM1032 is carried out via the serial bus. The ADM1032 is connected to this bus as a slave device, under the control of a master device, e.g. the Intel 820 chipset.

There is a programmable SMBus timeout. When this is enabled the SMBus will timeout after 25ms of no activity. However this feature is not enabled by default. To enable it set bit 7 of the Consecutive Alert Register (Addr = 22h)

The ADM1032 supports Packet Error Checking (PEC) and its use is optional. It is triggered by supplying the extra clock for the PEC byte. The PEC byte is calculated using CRC-8. The Frame Check Sequence (FCS) conforms to CRC-8 by the polynomial:-

$C(x) = x^8 + x^2 + x^1 + 1$

Consult SMBus 1.1 specification for more information.

ADDRESSING THE DEVICE

In general, every SMBus device has a 7-bit device address (except for some devices that have extended, 10 bit addresses). When the master device sends a device address over the bus, the slave device with that address will respond. The ADM1032 is available with one device address, which is Hex 4C (1001 100)

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 SDATA whilst the serial clock line SCLK 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 a $R/\overline{W}$ bit, which 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 whilst the selected device waits for data to be read from or written to it. if the $R/\overline{W}$ bit is a 0 then 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 9 clock pulses, 8 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 10th clock pulse to assert a STOP condition. In READ mode, the master device will override the acknowldge bit by pulling the data line high during the low period before the 9th clock pulse. This is known as No Acknowledge. The master will then take the data line low during the low period before the 10th clock pulse, then high during the 10th 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 ADM1032, write operations contain either one or two bytes, while 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, then data can be written into that register or read from it. The first byte of a write operation always contains a valid 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 3a. 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 ADM1032'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 ADM1032 as before, but only the data byte containing the register read address is sent, as data is not to be written to the register. This is shown in figure 3b.

   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 3c.

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 3b 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. Don't forget that some of the ADM1032 registers have different addresses for read and write operations. The write address of a register must be written to the Address Pointer if data is to be written to that register, but it is not possible to read data from that address. The read address of a register must be written to the Address Pointer before data can be read from that register.

$\overline{\text{ALERT}}$ OUTPUT

The $\overline{\text{ALERT}}$ output goes low whenever an out-of limit measurement is detected, or if the remote temperature sensor is open-circuit. It is an open-drain and requires a 10kΩ pullup to V_{DD}. Several $\overline{\text{ALERT}}$ outputs can be wire-ORED together, so that the common line will go low if one or more of the $\overline{\text{ALERT}}$ outputs goes low.

The $\overline{\text{ALERT}}$ output can be used as an interrupt signal to a processor, or it may be used as an $\overline{\text{SMBALERT}}$. Slave devices on the SMBus can normally not signal to the master that they want to talk, but the $\overline{\text{SMBALERT}}$ function allows them to do so.

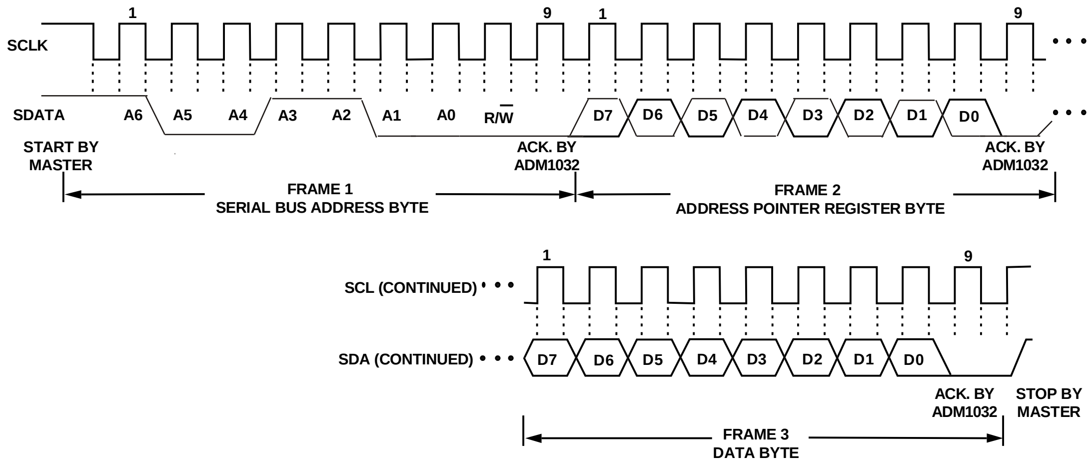


Figure 3a. Writing a Register Address to the Address Pointer Register, then Writing Data to the Selected Register

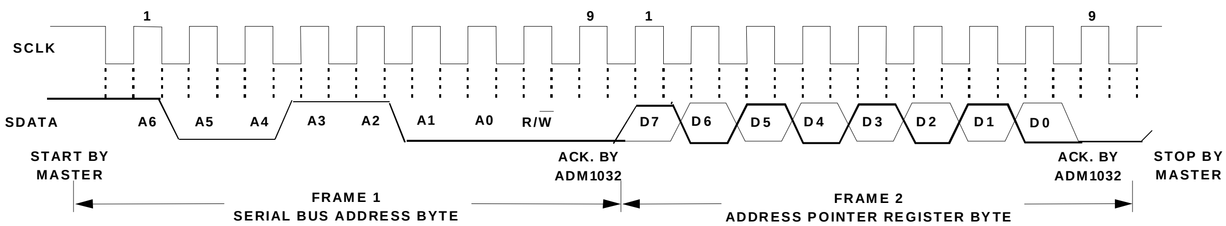


Figure 3b. Writing to the Address Pointer Register only

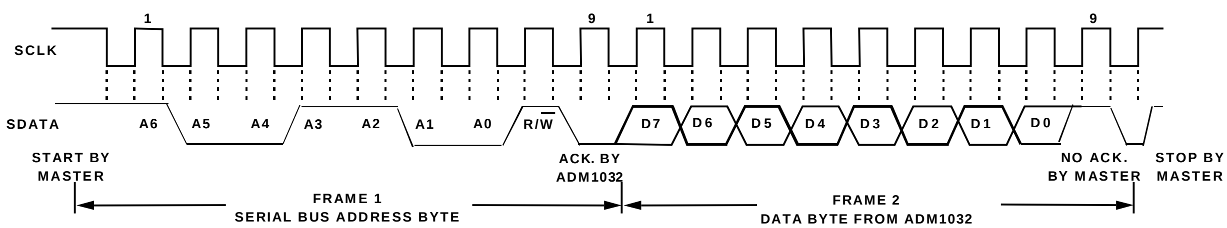


Figure 3c. Reading Data from a Previously Selected Register

One or more ALERT outputs are connected to a common SMBALERT line connected to the master. When the SMBALERT line is pulled low by one of the devices, the following procedure occurs as illustrated in Figure 4.

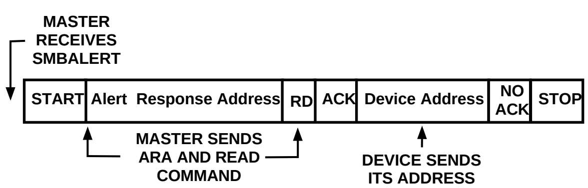


Figure 4. Use of SMBALERT

1. 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 ALERT 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 it can be interrogated in the usual way.
4. If more than one device's ALERT output is low, the one with the lowest device address, will have priority, in accordance with normal SMBus arbitration.
5. Once the ADM1032 has responded to the Alert Response Address, it will reset its ALERT output, provided that the error condition that caused the ALERT no longer exists. If the SMBALERT line remains low, the master will send ARA again, and so on until all devices whose ALERT outputs were low have responded.

LOW-POWER STANDBY MODE

The ADM1032 can be put into a low-power standby mode by setting bit 6 of the Configuration Register. When bit 6 is low the ADM1032 operates normally. When bit 6 is high, the ADC is inhibited, any conversion in progress is terminated without writing the result to the corresponding value register.

The SMBus is still enabled. Power consumption in the standby mode is reduced to less than 10µA if there is no SMBus activity, or 100µA if there are clock and data signals on the bus.

When the devices is in standby mode, it is still possible to initiate a one-shot conversion of both channels by writing XXh to the One-Shot Register (address 0Fh), after which the device will return to standby. It is also possible to write new values to the limit register while it is in standby. If the values stored in the temperature value registers are now outside the new limits then an $\overline{\text{ALERT}}$ is generated even though the ADM1032 is still in standby.

THE ADM1032 INTERRUPT SYSTEM

The ADM1032 has two interrupt outputs, $\overline{\text{ALERT}}$ and $\overline{\text{THERM}}$. These have different functions. $\overline{\text{ALERT}}$ responds to violations of software programmed temperature limits and is maskable. $\overline{\text{THERM}}$ is intended as a "failsafe" interrupt output that cannot be masked.

If the temperature goes below the lower temperature limit, the $\overline{\text{ALERT}}$ pin will be asserted low to indicate an out-of-limit condition. If the temperature is within the programmed low and high temperature limits, no interrupt will be generated.

If the temperature exceeds the high temperature limit, the $\overline{\text{ALERT}}$ pin will be asserted low to indicate an over-temperature condition. A local and remote $\overline{\text{THERM}}$ limit, may be programmed into the device to set the temperature limit above which the over-temperature $\overline{\text{THERM}}$ pin will be asserted low. This temperature limit should be equal to or greater than the high temperature limit programmed.

The behaviour of the high limit and $\overline{\text{THERM}}$ limit is as follows:-

1. If either the temperature measured exceeds the high temperature limit, the $\overline{\text{ALERT}}$ output will assert low.

2. If the local or remote temperature continues to increase and either one exceeds the $\overline{\text{THERM}}$ limit, the $\overline{\text{THERM}}$ output asserts low. This can be used to throttle the CPU clock or switch on a fan.

A $\overline{\text{THERM}}$ Hysteresis Value is provided to prevent the cooling fan cycling on and off. The power on default value is 10°C but this may be reprogrammed to any value after power up.This hysteresis value applies to both the local and remote channels

Using these 2 limits 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.

The THERM signal is open drain and requires a min of a 2.2kΩ pullup to VDD. The THERM signal should always be pulled up to the same power supply as the ADM1032, unlike the SMBus signals (SDATA, SCLK and ALERT) which may be pulled to to a different power rail, usually that of the SMBus controller.

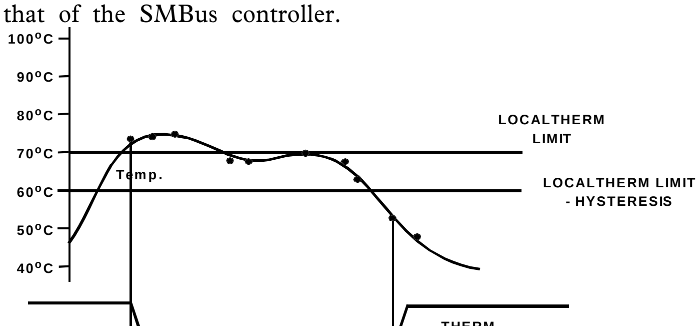


Figure 13. Operation of the THERM Output

TABLE VII. THERM HYSTERESIS SAMPLE VALUES

$\overline{\text{THERM}}$ $\overline{\text{HYSTERESIS}}$	Binary Representation
0°C	0 000 0000
+1°C	0 000 0001
+10°C	0 000 1010

SENSOR FAULT DETECTION

The ADM1032 has a fault detector at the D+ input that detects if the external sensor diode is open-circuit. This is a simple voltage comparator that trips if the voltage at D+ exceeds V_{DD} - 1V (typical). The output of this comparator is checked when a conversion is initiated, and sets bit 2 of the Status Register if a fault is detected.

If the remote sensor voltage falls below the normal measuring range, for example due to the diode being short-circuited, the ADC will output -128 (1000 0000). Since the normal operating temperature range of the device only extends down to 0^{O}C, this output code should never be seen in normal operation, so it can be interpreted as a fault condition. Since it will be outside the power-on default low temperature limit (0^{O}C) and any low limit that would normally be programmed, a short-circuit sensor will cause an SMBus alert.

In this respect the ADM1032 differs from and improves upon, competitive devices that output zero if the external sensor goes short-circuit. These devices can misinterpret a genuine 0°C measurement as a fault condition.

If the external diode channel is not being used and it is shorted out, then the resulting $\overline{\text{ALERT}}$ may be cleared by writing 80h (-128°C) to the low limit register.

APPLICATIONS INFORMATION

FACTORS AFFECTING ACCURACY

REMOTE SENSING DIODE

The ADM1032 is designed to work with substrate transistors built into processors, or with discrete transistors. Substrate transistors will generally be PNP types with the collector connected to the substrate. Discrete types can be either PNP or NPN, connected as a diode (base shorted to collector). If an NPN transistor is used then the collector and base are connected to D+ and the emitter to D-. If a PNP transistor is used then the collector and base are connected to D- and the emitter to D+.

The user has no choice in the case of substrate transistors but if a discrete transistor is used the best accuracy will be obtained by choosing devices according to the following criteria:

1. Base-emitter voltage greater than 0.25V at 6µA, at the highest operating temperature.

2. Base-emitter voltage less than 0.95V at 100µA, at the lowest operating temperature.

3. Base resistance less than 100Ω.

4. Small variation in h_{fe} (say 50 to 150) which indicates tight control of V_{be} characteristics.

Transistors such as 2N3904, 2N3906 or equivalents in SOT23 package are suitable devices to use.

THERMAL INERTIA AND SELF-HEATING

Accuracy depends on the temperature of the remote-sensing diode and/or the internal temperaure sensor being at the same temperature as what is being measured, and a number of factors can affect this. Ideally, the sensor should be in good thermal contact with the part of the system being measured, for example the processor. If it is not, then the thermal inertia caused by the mass of the sensor will cause a lag in the response of the sensor to a temperature change. In the case of the remote sensor this should not be a problem, as it will either be a substrate transistor in the processor, or can be a small package device such as SOT23 placed in close proximity to it.

The on-chip sensor, however, will often be remote from the processor, and will only be monitoring the general ambient temperature around the package. The thermal time constant of the SO-8 package in still air is about 140 seconds, and if the ambient air temperature quickly changed by 100 degrees, it would take about 12 minutes (5 time constants) for the junction temperature of the ADM1032 to settle within 1 degree of this. In practice, the ADM1032 package will be in electrical and hence thermal contact with a printed circuit board, and may also be in a forced airflow. How accurately the temperature of the board and/or the forced airflow reflect the temperature to be measured will also affect the accuracy.

Self-heating due to the power dissipated in the ADM1032 or the the remote sensor causes the chip temperature of the device or remote sensor to rise above ambient. However, the current forced through the remote sensor is so small that self-heating is negligible. In the case of the ADM1032, the worst-case condition occurs when the device is converting at its fastest rate whilst sinking the maximum current of 1mA at the $\overline{\text{ALERT}}$ and $\overline{\text{THERM}}$ output. In this case the total,power dissipation in the device is about 6mW The thermal resistance, θ_{JA} of the SO-8 package is about 150^{O}C/W, so even if the package was in free air with no connections to the leads, the temperature rise should be no more than

$\Delta T = 0.0027 \times 150 = 0.4^{O}C$.

In practice the package will have electrical and hence thermal connection to the printed circuit board, so the temperature rise due to self-heating will be negligible.

LAYOUT CONSIDERATIONS

Digital boards can be electrically noisy environments, and the ADM1032 is measuring very small voltages from the remote sensor, so care must be taken to minimise noise induced at the sensor inputs. The following precautions should be taken:

1. Place the ADM1032 as close as possible to the remote sensing diode. Provided that the worst noise sources such as clock generators, data/address buses and CRTs are avoided, this distance can be 4 to 8 inches.
2. Route the D+ and D- tracks close together, in parallel, with grounded guard tracks on each side. Provide a ground plane under the tracks if possible.
3. Use wide tracks to minimize inductance and reduce noise pickup. 10 mil track minimum width and spacing is recommended.

Figure 5. 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 should not be a major problem as 1°C corresponds to about 200mV, and thermocouple voltages are about 3mV/°C of temperature difference. Unless there are two thermocouples with a big temperature differential between them, thermocouple voltages should be much less than 200mV.
5. Placea 0.1µF bypass capacitor close to the V_{DD} pin and 2200pF input filter capacitors across D+, D- close to the ADM1032.
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 ADM1032. 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 may be reduced or removed.

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

APPLICATION CIRCUIT

Figure 6 shows a typical application circuit for the ADM1032, using a discrete sensor transistor connected via a shielded, twisted pair cable. The pullups on SCLK, SDATA and $\overline{\text{ALERT}}$ are required only if they are not already provided elsewhere in the system.

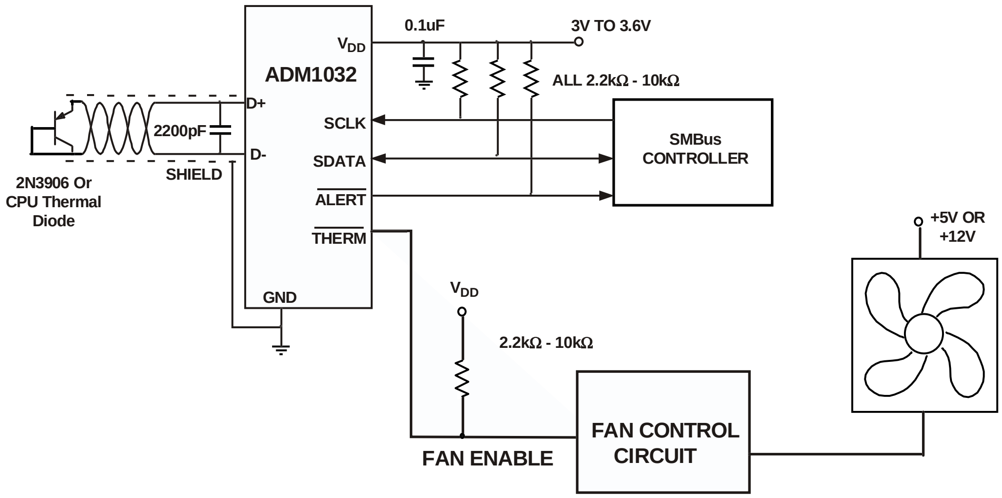


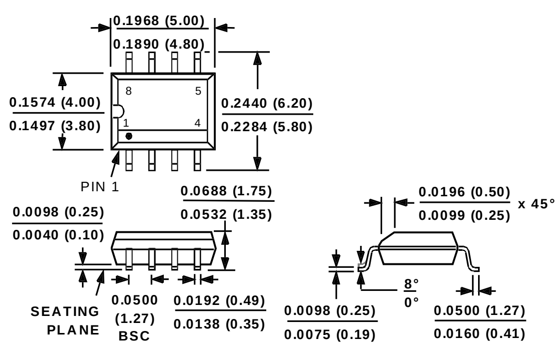
Figure 14. Typical ADM1032 Application Circuit

The SCLK, and SDATA pins of the ADM1032 can be interfaced directly to the SMBus of an I/O controller such as the Intel 820 chipset.

OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

8-Pin SO Package (SO-8)



8-Pin SOIC Package (RM-8)