19-4508; Rev 0; 1/99

EVALUATION KIT MANUAL FOLLOWS DATA SHEET

MAXIM

Remote/Local Temperature Sensor with SMBus Serial Interface

General Description

The MAX1617A (patents pending) is a precise digital thermometer that reports the temperature of both a remote sensor and its own package. The remote sensor is a diode-connected transistor—typically a low-cost, easily mounted 2N3904 NPN type—that replaces conventional thermistors or thermocouples. Remote accuracy is ±3°C for multiple transistor manufacturers, with no calibration needed. The remote channel can also measure the die temperature of other ICs, such as microprocessors, that contain an on-chip, diode-connected transistor.

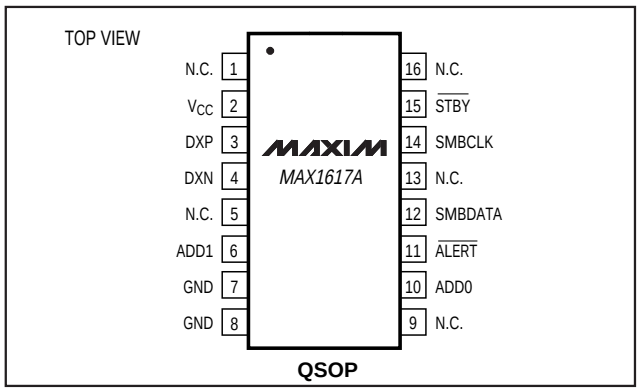
The 2-wire serial interface accepts standard System Management Bus (SMBus®) Write Byte, Read Byte, Send Byte, and Receive Byte commands to program the alarm thresholds and to read temperature data. The data format is 7 bits plus sign, with each bit corresponding to 1°C, in two's complement format. Measurements can be done automatically and autonomously, with the conversion rate programmed by the user or programmed to operate in a single-shot mode. The adjustable rate allows the user to control the supply-current drain.

The MAX1617A is nearly identical to the popular MAX1617, but has improved SMBus timing specifications, improved bus collision immunity, software manufacturer and device identification available via the serial interface, and a power-on reset function that can force a reset of the slave address via the serial interface.

Applications

Desktop and Notebook Computers	Central Office Telecom Equipment
Smart Battery Packs	Test and Measurement
LAN Servers	Multichip Modules
Industrial Controls	

Pin Configuration



SMBus is a registered trademark of Intel Corp.

Features

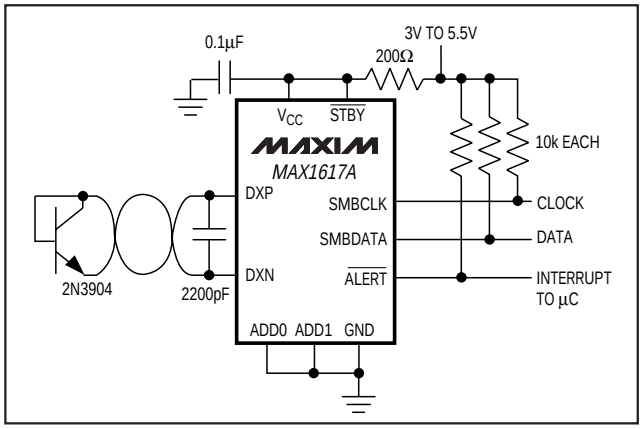
- **Two Channels: Measures Both Remote and Local Temperatures**
- **No Calibration Required**
- **SMBus 2-Wire Serial Interface**
- **Programmable Under/Overtemperature Alarms**
- **Supports SMBus Alert Response**
- **Supports Manufacturer and Device ID Codes**
- **Accuracy**
  - **±2°C (+60°C to +100°C, local)**
  - **±3°C (-40°C to +125°C, local)**
  - **±3°C (+60°C to +100°C, remote)**
- **3µA (typ) Standby Supply Current**
- **70µA (max) Supply Current in Auto-Convert Mode**
- **+3V to +5.5V Supply Range**
- **Small 16-Pin QSOP Package**

Ordering Information

PART*	TEMP. RANGE	PIN-PACKAGE
MAX1617AMEE	-55°C to +125°C	16 QSOP

**U.S. and foreign patents pending.*

Typical Operating Circuit



†Patents Pending

Remote/Local Temperature Sensor with SMBus Serial Interface

ABSOLUTE MAXIMUM RATINGS

V_{CC} to GND .....-0.3V to +6V
DXP, ADD_ to GND .....-0.3V to (V_{CC} + 0.3V)
DXN to GND .....-0.3V to +0.8V
SMBCLK, SMBDATA, $\overline{ALERT}$, $\overline{STBY}$ to GND .....-0.3V to +6V
SMBDATA, $\overline{ALERT}$ Current .....-1mA to +50mA
DXN Current .....±1mA
ESD Protection (SMBCLK, SMBDATA, $\overline{ALERT}$, Human Body Model) 4000V
ESD Protection (other pins, Human Body Model).....2000V

Continuous Power Dissipation (T_A = +70°C)
QSOP (derate 8.30mW/°C above +70°C).....667mW
Operating Temperature Range .....-55°C to +125°C
Junction Temperature.....+150°C
Storage Temperature Range .....-65°C to +165°C
Lead Temperature (soldering, 10sec) .....+300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{CC} = +3.3V, **T_A = 0°C to +85°C**, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
ADC AND POWER SUPPLY						
Temperature Resolution (Note 1)	Monotonicity guaranteed		8			Bits
Initial Temperature Error, Local Diode (Note 2)	T_A = +60°C to +100°C		-2		2	°C
	T_A = 0°C to +85°C		-3		3	
Temperature Error, Remote Diode (Notes 2 and 3)	T_R = +60°C to +100°C		-3		3	°C
	T_R = -55°C to +125°C		-5		5	
Temperature Error, Local Diode (Notes 1 and 2)	Including long-term drift	T_A = +60°C to +100°C	-2.5		2.5	°C
		T_A = 0°C to +85°C	-3.5		3.5	
Supply-Voltage Range			3.0		5.5	V
Undervoltage Lockout Threshold	V_{CC} input, disables A/D conversion, rising edge		2.60	2.80	2.95	V
Undervoltage Lockout Hysteresis				50		mV
Power-On Reset Threshold	V_{CC}, falling edge		1.0	1.7	2.5	V
POR Threshold Hysteresis				50		mV
Standby Supply Current	Logic inputs forced to V_{CC} or GND	SMBus static		3	10	µA
		Hardware or software standby, SMBCLK at 10kHz		4		
Average Operating Supply Current	Auto-convert mode, average measured over 4sec. Logic inputs forced to V_{CC} or GND.	0.25 conv/sec		35	70	µA
		2.0 conv/sec		120	180	
Conversion Time	From stop bit to conversion complete (both channels)		94	125	156	ms
Conversion Rate Timing Error	Auto-convert mode		-25		25	%
Remote-Diode Source Current	DXP forced to 1.5V	High level	80	100	120	µA
		Low level	8	10	12	
DXN Source Voltage				0.7		V
Address Pin Bias Current	ADD0, ADD1; momentary upon power-on reset			160		µA

ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = +3.3V, **T_A = 0°C to +85°C**, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
SMBus INTERFACE					
Logic Input High Voltage	$\overline{STBY}$, SMBCLK, SMBDATA; V_{CC} = 3V to 5.5V	2.2			V
Logic Input Low Voltage	$\overline{STBY}$, SMBCLK, SMBDATA; V_{CC} = 3V to 5.5V			0.8	V
Logic Output Low Sink Current	$\overline{ALERT}$, SMBDATA forced to 0.4V	6			mA
$\overline{ALERT}$ Output High Leakage Current	$\overline{ALERT}$ forced to 5.5V			1	µA
Logic Input Current	Logic inputs forced to V_{CC} or GND	-1		1	µA
SMBus Input Capacitance	SMBCLK, SMBDATA		5		pF
SMBus Clock Frequency	(Note 4)	DC		100	kHz
SMBCLK Clock Low Time	t_{LOW}, 10% to 10% points	4.7			µs
SMBCLK Clock High Time	t_{HIGH}, 90% to 90% points	4			µs
SMBus Start-Condition Setup Time		4.7			µs
SMBus Repeated Start-Condition Setup Time	$t_{SU:STA}$, 90% to 90% points	500			ns
SMBus Start-Condition Hold Time	$t_{HD:STA}$, 10% of SMBDATA to 90% of SMBCLK	4			µs
SMBus Stop-Condition Setup Time	$t_{SU:STO}$, 90% of SMBCLK to 10% of SMBDATA	4			µs
SMBus Data Valid to SMBCLK Rising-Edge Time	$t_{SU:DAT}$, 10% or 90% of SMBDATA to 10% of SMBCLK	250			ns
SMBus Data-Hold Time	$t_{HD:DAT}$ (Note 5)	0			µs
SMBCLK Falling Edge to SMBus Data-Valid Time	Master clocking in data			1	µs

ELECTRICAL CHARACTERISTICS

(V_{CC} = +3.3V, **T_A = -55°C to +125°C**, unless otherwise noted.) (Note 6)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
ADC AND POWER SUPPLY					
Temperature Resolution (Note 1)	Monotonicity guaranteed	8			Bits
Initial Temperature Error, Local Diode (Note 2)	T_A = +60°C to +100°C	-2		2	°C
	T_A = -55°C to +125°C	-3		3	
Temperature Error, Remote Diode (Notes 2 and 3)	T_R = +60°C to +100°C	-3		3	°C
	T_R = -55°C to +125°C	-5		5	
Supply-Voltage Range		3.0		5.5	V
Conversion Time	From stop bit to conversion complete (both channels)	94	125	156	ms
Conversion Rate Timing Error	Auto-convert mode	-25		25	%

Remote/Local Temperature Sensor with SMBus Serial Interface

ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = +3.3V, **T_A = -55°C to +125°C**, unless otherwise noted.) (Note 6)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
SMBus INTERFACE						
Logic Input High Voltage	$\overline{STBY}$, SMBCLK, SMBDATA	V_{CC} = 3V	2.2			V
		V_{CC} = 5.5V	2.4			
Logic Input Low Voltage	$\overline{STBY}$, SMBCLK, SMBDATA; V_{CC} = 3V to 5.5V				0.8	V
Logic Output Low Sink Current	$\overline{ALERT}$, SMBDATA forced to 0.4V		6			mA
$\overline{ALERT}$ Output High Leakage Current	$\overline{ALERT}$ forced to 5.5V				1	µA
Logic Input Current	Logic inputs forced to V_{CC} or GND		-2		2	µA

Note 1: Guaranteed but not 100% tested.

Note 2: Quantization error is not included in specifications for temperature accuracy. For example, if the MAX1617A device temperature is exactly +66.7°C, the ADC may report +66°C, +67°C, or +68°C (due to the quantization error plus the +1/2°C offset used for rounding up) and still be within the guaranteed ±1°C error limits for the +60°C to +100°C temperature range (Table 2).

Note 3: A remote diode is any diode-connected transistor from Table 1. T_R is the junction temperature of the remote diode. See *Remote Diode Selection* for remote diode forward voltage requirements.

Note 4: The SMBus logic block is a static design that works with clock frequencies down to DC. While slow operation is possible, it violates the 10kHz minimum clock frequency and SMBus specifications, and may monopolize the bus.

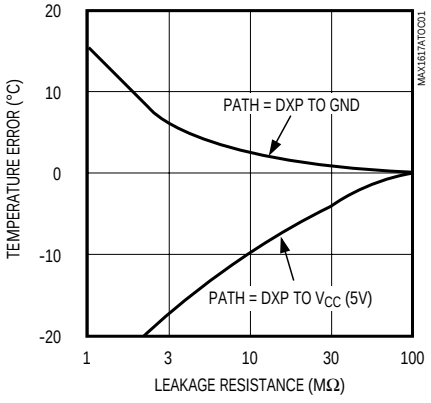
Note 5: Note that a transition must internally provide at least a hold time in order to bridge the undefined region (300ns max) of SMBCLK's falling edge.

Note 6: Specifications from -55°C to +125°C are guaranteed by design, not production tested.

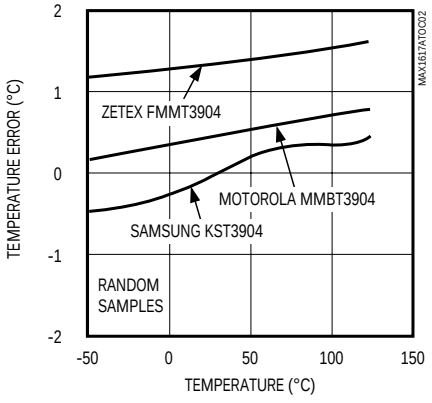
Typical Operating Characteristics

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

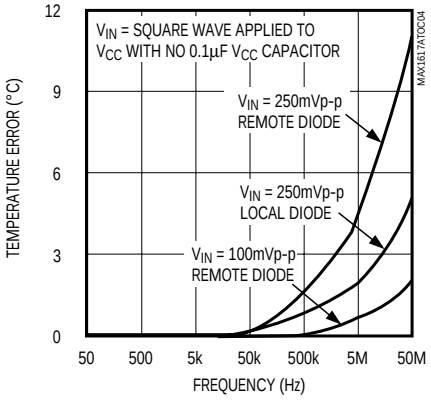
TEMPERATURE ERROR vs. PC BOARD RESISTANCE



TEMPERATURE ERROR vs. REMOTE-DIODE TEMPERATURE



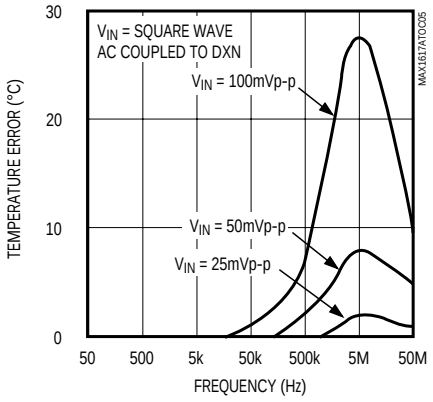
TEMPERATURE ERROR vs. POWER-SUPPLY NOISE FREQUENCY



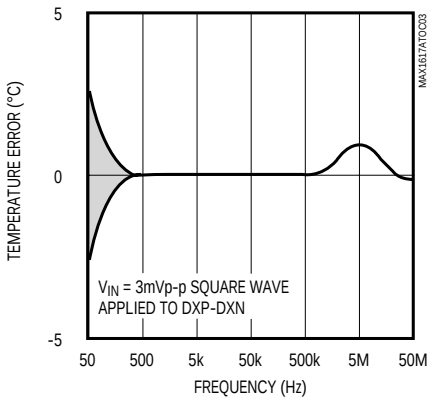
Typical Operating Characteristics (continued)

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

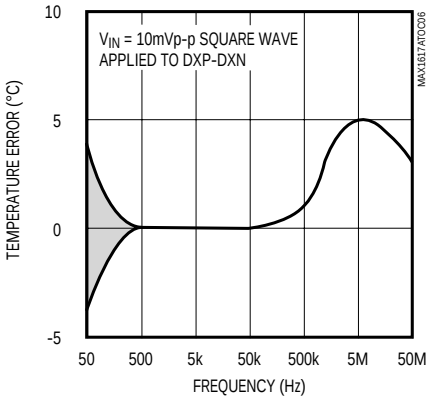
TEMPERATURE ERROR vs. COMMON-MODE NOISE FREQUENCY



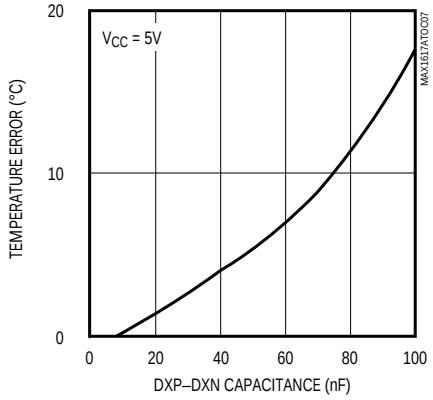
TEMPERATURE ERROR vs. DIFFERENTIAL-MODE NOISE FREQUENCY



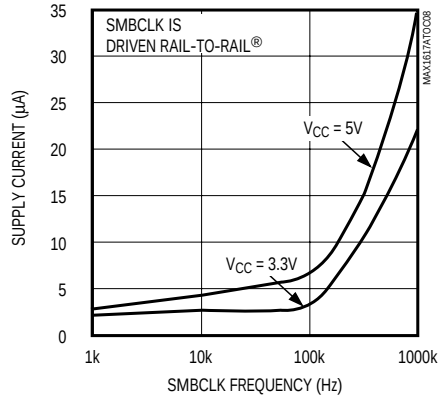
TEMPERATURE ERROR vs. DIFFERENTIAL-MODE NOISE FREQUENCY



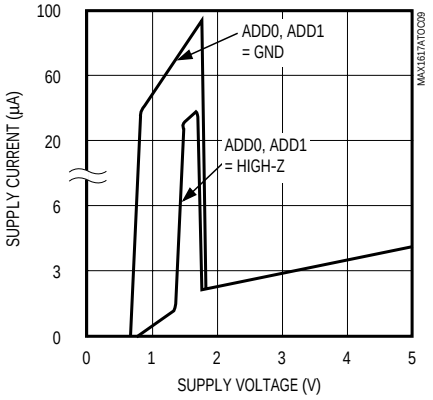
TEMPERATURE ERROR vs. DXP–DXN CAPACITANCE



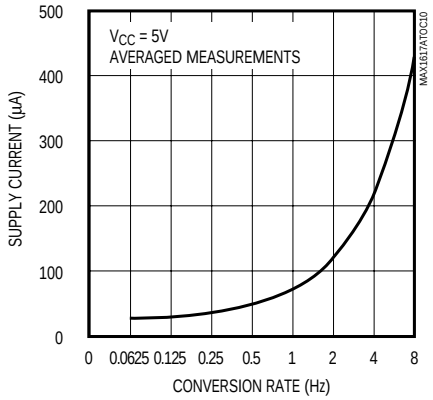
STANDBY SUPPLY CURRENT vs. CLOCK FREQUENCY



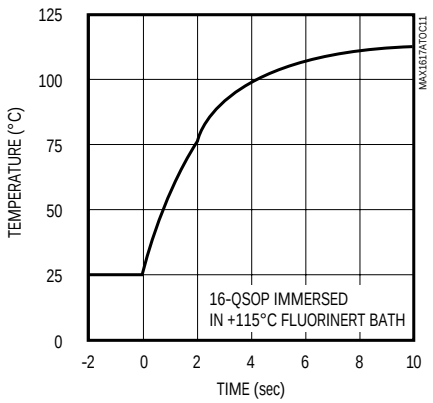
STANDBY SUPPLY CURRENT vs. SUPPLY VOLTAGE



OPERATING SUPPLY CURRENT vs. CONVERSION RATE



RESPONSE TO THERMAL SHOCK



Rail-to Rail is a registered trademark of Nippon Motorola, Ltd.

Remote/Local Temperature Sensor with SMBus Serial Interface

Pin Description

PIN	NAME	FUNCTION
1, 5, 9, 13, 16	N.C.	No Connection. Not internally connected. May be used for PC board trace routing.
2	V_{CC}	Supply Voltage Input, 3V to 5.5V. Bypass to GND with a 0.1µF capacitor. A 200Ω series resistor is recommended but not required for additional noise filtering.
3	DXP	Combined Current Source and A/D Positive Input for Remote-Diode Channel. Do not leave DXP floating; tie DXP to DXN if no remote diode is used. Place a 2200pF capacitor between DXP and DXN for noise filtering.
4	DXN	Combined Current Sink and A/D Negative Input. DXN is normally biased to a diode voltage above ground.
6	ADD1	SMBus Address Select Pin (Table 8). ADD0 and ADD1 are sampled upon power-up. Excess capacitance (>50pF) at the address pins when floating may cause address-recognition problems.
7, 8	GND	Ground
10	ADD0	SMBus Slave Address Select Pin
11	$\overline{ALERT}$	SMBus Alert (interrupt) Output, Open Drain
12	SMBDATA	SMBus Serial-Data Input/Output, Open Drain
14	SMBCLK	SMBus Serial-Clock Input
15	$\overline{STBY}$	Hardware Standby Input. Temperature and comparison threshold data are retained in standby mode. Low = standby mode, high = operate mode.

General Description

The MAX1617A (patents pending) is a temperature sensor designed to work in conjunction with an external microcontroller (µC) or other intelligence in thermostatic, process-control, or monitoring applications. The µC is typically a power-management or keyboard controller, generating SMBus serial commands by "bit-banging" general-purpose input/output (GPIO) pins or via a dedicated SMBus interface block.

Essentially an 8-bit serial analog-to-digital converter (ADC) with a sophisticated front end, the MAX1617A contains a switched current source, a multiplexer, an ADC, an SMBus interface, and associated control logic (Figure 1). Temperature data from the ADC is loaded into two data registers, where it is automatically compared with data previously stored in four over/under-temperature alarm registers.

ADC and Multiplexer

The ADC is an averaging type that integrates over a 60ms period (each channel, typical) with excellent noise rejection.

The multiplexer automatically steers bias currents through the remote and local diodes, measures their forward voltages, and computes their temperatures. Both channels are automatically converted once the conversion process has started, either in free-running or single-shot mode. If one of the two channels is not used, the device still performs both measurements, and the user can simply ignore the results of the unused channel. If the remote diode channel is unused, tie DXP to DXN rather than leaving the pins open.

The DXN input is biased at 0.65V above ground by an internal diode to set up the analog-to-digital (A/D) inputs for a differential measurement. The worst-case DXP–DXN differential input voltage range is 0.25V to 0.95V.

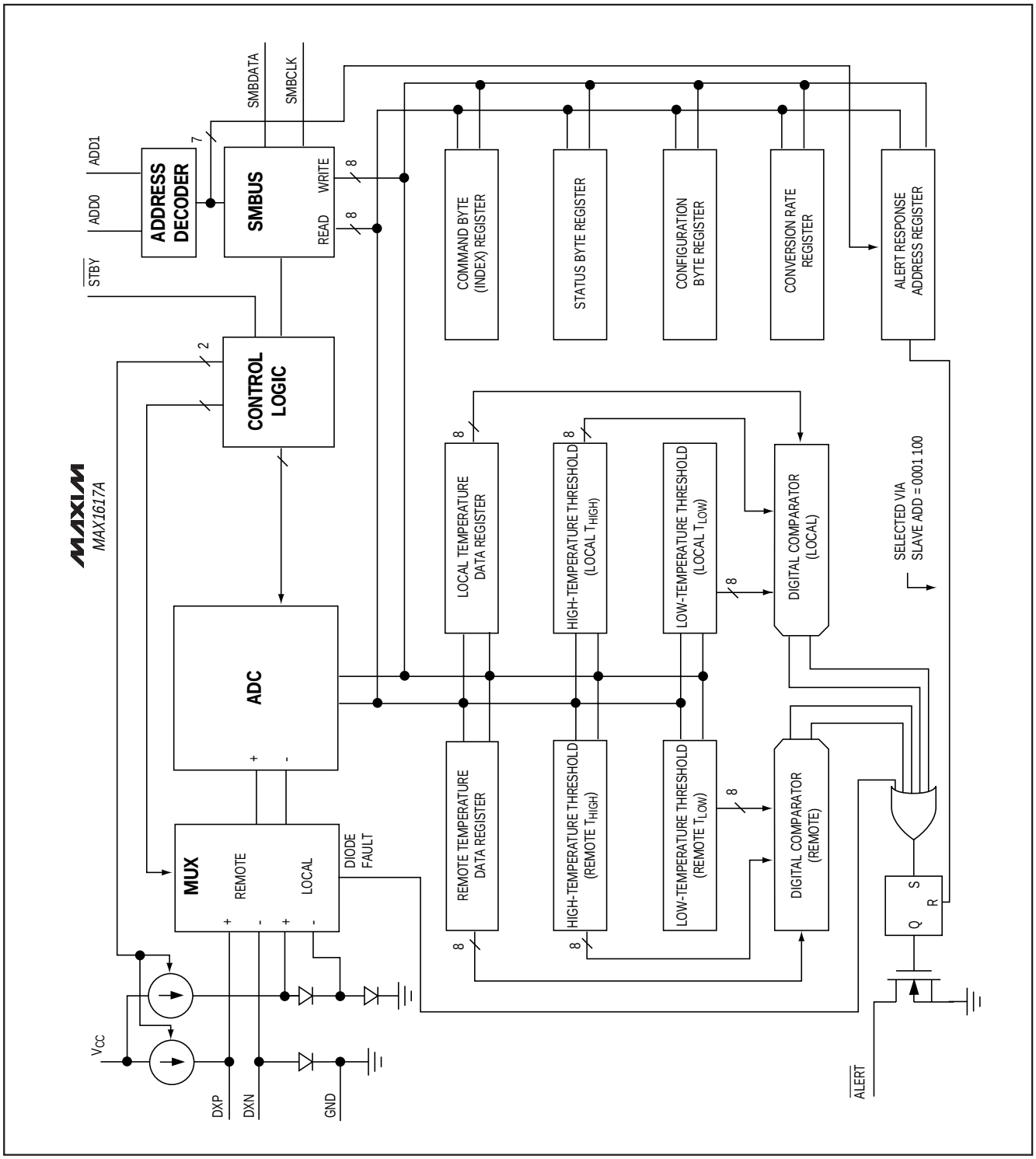


Figure 1. Functional Diagram

Remote/Local Temperature Sensor with SMBus Serial Interface

Excess resistance in series with the remote diode causes about +1/2°C error per ohm. Likewise, 200µV of offset voltage forced on DXP–DXN causes about 1°C error.

A/D Conversion Sequence

If a Start command is written (or generated automatically in the free-running auto-convert mode), both channels are converted, and the results of both measurements are available after the end of conversion. A BUSY status bit in the status byte shows that the device is actually performing a new conversion; however, even if the ADC is busy, the results of the previous conversion are always available.

Remote-Diode Selection

Temperature accuracy depends on having a good-quality, diode-connected small-signal transistor. Accuracy has been experimentally verified for all of the devices listed in Table 1. The MAX1617A can also directly measure the die temperature of CPUs and other integrated circuits having on-board temperature-sensing diodes.

The transistor must be a small-signal type with a relatively high forward voltage; otherwise, the A/D input voltage range can be violated. The forward voltage must be greater than 0.25V at 10µA; check to ensure this is true at the highest expected temperature. The forward voltage must be less than 0.95V at 100µA; check to ensure this is true at the lowest expected temperature. Large power transistors don't work at all. Also ensure that the base resistance is less than 100Ω. Tight specifications for forward-current gain (+50 to +150, for example) indicate that the manufacturer has good process controls and that the devices have consistent VBE characteristics.

For heatsink mounting, the 500-32BT02-000 thermal sensor from Fenwal Electronics is a good choice. This device consists of a diode-connected transistor, an aluminum plate with screw hole, and twisted-pair cable (Fenwal Inc., Milford, MA, 508-478-6000).

Thermal Mass and Self-Heating

Thermal mass can seriously degrade the MAX1617A's effective accuracy. The thermal time constant of the QSOP-16 package is about 140sec in still air. For the MAX1617A junction temperature to settle to within +1°C after a sudden +100°C change requires about five time constants or 12 minutes. The use of smaller packages for remote sensors, such as SOT23s, improves the situation. Take care to account for thermal gradients between the heat source and the sensor, and ensure that stray air currents across the sensor package do not interfere with measurement accuracy.

Table 1. Remote-Sensor Transistor Manufacturers

MANUFACTURER	MODEL NUMBER
Central Semiconductor (USA)	CMPT3904
Motorola (USA)	MMBT3904
National Semiconductor (USA)	MMBT3904
Rohm Semiconductor (Japan)	SST3904
Samsung (Korea)	KST3904-TF
Siemens (Germany)	SMBT3904
Zetex (England)	FMMT3904CT-ND

***Note**: Transistors must be diode-connected (base shorted to collector).*

Self-heating does not significantly affect measurement accuracy. Remote-sensor self-heating due to the diode current source is negligible. For the local diode, the worst-case error occurs when auto-converting at the fastest rate and simultaneously sinking maximum current at the $\overline{ALERT}$ output. For example, at an 8Hz rate and with $\overline{ALERT}$ sinking 1mA, the typical power dissipation is V_{CC} · 450µA plus 0.4V · 1mA. Package theta J-A is about 150°C/W, so with V_{CC} = 5V and no copper PC board heatsinking, the resulting temperature rise is:

$$dT = 2.7mW \cdot 150°C/W = 0.4°C$$

Even with these contrived circumstances, it is difficult to introduce significant self-heating errors.

ADC Noise Filtering

The ADC is an integrating type with inherently good noise rejection, especially of low-frequency signals such as 60Hz/120Hz power-supply hum. Micropower operation places constraints on high-frequency noise rejection; therefore, careful PC board layout and proper external noise filtering are required for high-accuracy remote measurements in electrically noisy environments.

High-frequency EMI is best filtered at DXP and DXN with an external 2200pF capacitor. This value can be increased to about 3300pF (max), including cable capacitance. Higher capacitance than 3300pF introduces errors due to the rise time of the switched current source.

Nearly all noise sources tested cause the ADC measurements to be higher than the actual temperature, typically by +1°C to +10°C, depending on the frequency and amplitude (see *Typical Operating Characteristics*).

PC Board Layout

1) Place the MAX1617A as close as practical to the remote diode. In a noisy environment, such as a computer motherboard, this distance can be 4 in. to 8 in. (typical) or more as long as the worst noise sources (such as CRTs, clock generators, memory buses, and ISA/PCI buses) are avoided.
2) Do not route the DXP–DXN lines next to the deflection coils of a CRT. Also, do not route the traces across a fast memory bus, which can easily introduce +30°C error, even with good filtering. Otherwise, most noise sources are fairly benign.
3) Route the DXP and DXN traces in parallel and in close proximity to each other, away from any high-voltage traces such as $+12V_{DC}$. Leakage currents from PC board contamination must be dealt with carefully, since a 20MΩ leakage path from DXP to ground causes about +1°C error.
4) Connect guard traces to GND on either side of the DXP–DXN traces (Figure 2). With guard traces in place, routing near high-voltage traces is no longer an issue.
5) Route through as few vias and crossunders as possible to minimize copper/solder thermocouple effects.
6) When introducing a thermocouple, make sure that both the DXP and the DXN paths have matching thermocouples. In general, PC board-induced thermocouples are not a serious problem. A copper-solder thermocouple exhibits 3µV/°C, and it takes about 200µV of voltage error at DXP–DXN to cause a +1°C measurement error. So, most parasitic thermocouple errors are swamped out.
7) Use wide traces. Narrow ones are more inductive and tend to pick up radiated noise. The 10 mil widths and spacings recommended in Figure 2 aren't absolutely necessary (as they offer only a minor improvement in leakage and noise), but try to use them where practical.
8) Keep in mind that copper can't be used as an EMI shield, and only ferrous materials, such as steel, work well. Placing a copper ground plane between the DXP-DXN traces and traces carrying high-frequency noise signals does not help reduce EMI.

PC Board Layout Checklist

- Place the MAX1617A close to a remote diode.
- Keep traces away from high voltages (+12V bus).
- Keep traces away from fast data buses and CRTs.
- Use recommended trace widths and spacings.
- Place a ground plane under the traces.

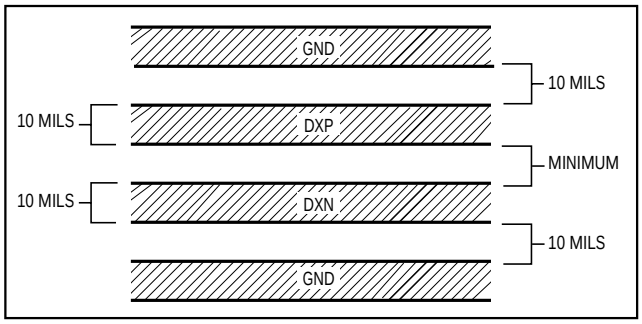


Figure 2. Recommended DXP/DXN PC Traces

- Use guard traces flanking DXP and DXN and connecting to GND.
- Place the noise filter and the 0.1µF V_{CC} bypass capacitors close to the MAX1617A.
- Add a 200Ω resistor in series with V_{CC} for best noise filtering (see *Typical Operating Circuit*).

Twisted Pair and Shielded Cables

For remote-sensor distances longer than 8 in., or in particularly noisy environments, a twisted pair is recommended. Its practical length is 6 feet to 12 feet (typical) before noise becomes a problem, as tested in a noisy electronics laboratory. For longer distances, the best solution is a shielded twisted pair like that used for audio microphones. For example, the Belden 8451 works well for distances up to 100 feet in a noisy environment. Connect the twisted pair to DXP and DXN and the shield to GND, and leave the shield's remote end unterminated.

Excess capacitance at DX_ limits practical remote sensor distances (see *Typical Operating Characteristics*). For very long cable runs, the cable's parasitic capacitance often provides noise filtering, so the 2200pF capacitor can often be removed or reduced in value.

Cable resistance also affects remote-sensor accuracy; 1Ω series resistance introduces about +1/2°C error.

Low-Power Standby Mode

Standby mode disables the ADC and reduces the supply-current drain to less than 10µA. Enter standby mode by forcing the $\overline{STBY}$ pin low or via the RUN/STOP bit in the configuration byte register. Hardware and software standby modes behave almost identically: all data is retained in memory, and the SMB interface is alive and listening for reads and writes. The only difference is that in hardware standby mode, the one-shot command does not initiate a conversion.

Standby mode is not a shutdown mode. With activity on the SMBus, extra supply current is drawn (see *Typical Operating Characteristics*). In software standby mode,

the MAX1617A can be forced to perform A/D conversions via the one-shot command, despite the RUN/STOP bit being high.

Activate hardware standby mode by forcing the $\overline{STBY}$ pin low. In a notebook computer, this line may be connected to the system SUSTAT# suspend-state signal.

The $\overline{STBY}$ pin low state overrides any software conversion command. If a hardware or software standby command is received while a conversion is in progress, the conversion cycle is truncated, and the data from that conversion is not latched into either temperature reading register. The previous data is not changed and remains available.

Supply-current drain during the 125ms conversion period is always about 450µA. Slowing down the conversion rate reduces the average supply current (see *Typical Operating Characteristics*). Between conversions, the instantaneous supply current is about 25µA due to the current consumed by the conversion rate timer. In standby mode, supply current drops to about 3µA. At very low supply voltages (under the power-on-reset threshold), the supply current is higher due to the address pin bias currents. It can be as high as 100µA, depending on ADD0 and ADD1 settings.

SMBus Digital Interface

From a software perspective, the MAX1617A appears as a set of byte-wide registers that contain temperature data, alarm threshold values, or control bits. A standard SMBus 2-wire serial interface is used to read temperature data and write control bits and alarm threshold data. Each A/D channel within the device responds to the same SMBus slave address for normal reads and writes.

The MAX1617A employs four standard SMBus protocols: Write Byte, Read Byte, Send Byte, and Receive Byte (Figure 3). The shorter Receive Byte protocol allows quicker transfers, provided that the correct data register was previously selected by a Read Byte instruction. Use caution with the shorter protocols in multi-master systems, since a second master could overwrite the command byte without informing the first master.

The temperature data format is 7 bits plus sign in two's complement form for each channel, with each data bit representing 1°C (Table 2), transmitted MSB first. Measurements are offset by +1/2°C to minimize internal rounding errors; for example, +99.6°C is reported as +100°C.

Write Byte Format

S	ADDRESS	WR	ACK	COMMAND	ACK	DATA	ACK	P
	7 bits			8 bits		8 bits		1

Slave Address: equivalent to chip-select line of a 3-wire interface

Command Byte: selects which register you are writing to

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

Read Byte Format

S	ADDRESS	WR	ACK	COMMAND	ACK	S	ADDRESS	RD	ACK	DATA	///	P
	7 bits			8 bits			7 bits			8 bits		

Slave Address: equivalent to chip-select line

Command Byte: selects which register you are reading from

Slave Address: repeated due to change in data-flow direction

Data Byte: reads from the register set by the command byte

Send Byte Format

S	ADDRESS	WR	ACK	COMMAND	ACK	P
	7 bits			8 bits		

Command Byte: sends command with no data, usually used for one-shot command

Receive Byte Format

S	ADDRESS	RD	ACK	DATA	///	P
	7 bits			8 bits		

Data Byte: reads data from the register commanded by the last Read Byte or Write Byte transmission; also used for SMBus Alert Response return address

S = Start condition *Shaded = Slave transmission*
P = Stop condition */// = Not acknowledged*

Figure 3. SMBus Protocols

Table 2. Data Format (Two's Complement)

TEMP. (°C)	ROUNDED TEMP. (°C)	DIGITAL OUTPUT DATA BITS		
		SIGN	MSB	LSB
+130.00	+127	0	111	1111
+127.00	+127	0	111	1111
+126.50	+127	0	111	1111
+126.00	+126	0	111	1110
+25.25	+25	0	001	1001
+0.50	+1	0	000	0001
+0.25	0	0	000	0000
0.00	0	0	000	0000
-0.25	0	0	000	0000
-0.50	0	0	000	0000
-0.75	-1	1	111	1111
-1.00	-1	1	111	1111
-25.00	-25	1	110	0111
-25.50	-26	1	110	0110
-54.75	-55	1	100	1001
-55.00	-55	1	100	1001
-65.00	-65	1	011	1111
-70.00	-65	1	011	1111

Alarm Threshold Registers

Four registers store alarm threshold data, with high-temperature (T_{HIGH}) and low-temperature (T_{LOW}) registers for each A/D channel. If either measured temperature equals or exceeds the corresponding alarm threshold value, an $\overline{ALERT}$ interrupt is asserted.

The power-on-reset (POR) state of both T_{HIGH} registers is full scale (0111 1111, or +127°C). The POR state of both T_{LOW} registers is 1100 1001 or -55°C.

Diode Fault Alarm

There is a continuity fault detector at DXP that detects whether the remote diode has an open-circuit condition. At the beginning of each conversion, the diode fault is checked, and the status byte is updated. This fault detector is a simple voltage detector; if DXP rises above V_{CC} - 1V (typical) due to the diode current source, a fault is detected. Note that the diode fault isn't checked until a conversion is initiated, so immediately after power-on reset the status byte indicates no fault is present, even if the diode path is broken.

If the remote channel is shorted (DXP to DXN or DXP to GND), the ADC reads 0000 0000 so as not to trip either the T_{HIGH} or T_{LOW} alarms at their POR settings. In applications that are never subjected to 0°C in normal operation, a 0000 0000 result can be checked to indicate a fault condition in which DXP is accidentally short circuited. Similarly, if DXP is short circuited to V_{CC}, the ADC reads +127°C for both remote and local channels, and the device alarms.

Table 3. Read Format for Alert Response Address (0001100)

BIT	NAME	FUNCTION
7 (MSB)	ADD7	Provide the current MAX1617A slave address that was latched at POR (Table 8)
6	ADD6	
5	ADD5	
4	ADD4	
3	ADD3	
2	ADD2	
1	ADD1	
0 (LSB)	1	Logic 1

ALERT *Interrupts*

The $\overline{ALERT}$ interrupt output signal is latched and can only be cleared by reading the Alert Response address. Interrupts are generated in response to T_{HIGH} and T_{LOW} comparisons and when the remote diode is disconnected (for continuity fault detection). The interrupt does not halt automatic conversions; new temperature data continues to be available over the SMBus interface after $\overline{ALERT}$ is asserted. The interrupt output pin is open-drain so that devices can share a common interrupt line. The interrupt rate can never exceed the conversion rate.

The interface responds to the SMBus Alert Response address, an interrupt pointer return-address feature (see *Alert Response Address* section). Prior to taking corrective action, always check to ensure that an interrupt is valid by reading the current temperature.

Alert Response Address

The SMBus Alert Response interrupt pointer provides quick fault identification for simple slave devices that lack the complex, expensive logic needed to be a bus master. Upon receiving an $\overline{ALERT}$ interrupt signal, the host master can broadcast a Receive Byte transmission to the Alert Response slave address (0001 100). Then any slave device that generated an interrupt attempts to identify itself by putting its own address on the bus (Table 3).

Table 4. Command-Byte Bit Assignments

REGISTER	COMMAND	POR STATE	FUNCTION
RLTS	00h	0000 0000*	Read local temperature: returns latest temperature
RRTE	01h	0000 0000*	Read remote temperature: returns latest temperature
RSL	02h	N/A	Read status byte (flags, busy signal)
RCL	03h	0000 0000	Read configuration byte
RCRA	04h	0000 0010	Read conversion rate byte
RLHN	05h	0111 1111	Read local T_{HIGH} limit
RLLI	06h	1100 1001	Read local T_{LOW} limit
RRHI	07h	0111 1111	Read remote T_{HIGH} limit
RRLS	08h	1100 1001	Read remote T_{LOW} limit
WCA	09h	N/A	Write configuration byte
WCRW	0Ah	N/A	Write conversion rate byte
WLHO	0Bh	N/A	Write local T_{HIGH} limit
WLLM	0Ch	N/A	Write local T_{LOW} limit
WRHA	0Dh	N/A	Write remote T_{HIGH} limit
WRLN	0Eh	N/A	Write remote T_{LOW} limit
OSHT	0Fh	N/A	One-shot command (use send-byte format)
SPOR	FCh	N/A	Write software POR
MFGID	FEh	0100 1101	Read manufacturer ID code
DEVID	FFh	00000001	Read device ID code

**If the device is in hardware standby mode at POR, both temperature registers read 0°C.*

The Alert Response can activate several different slave devices simultaneously, similar to the I²C™ General Call. If more than one slave attempts to respond, bus arbitration rules apply, and the device with the lower address code wins. The losing device does not generate an acknowledge and continues to hold the $\overline{ALERT}$ line low until serviced (implies that the host interrupt input is level-sensitive). Successful reading of the alert response address clears the interrupt latch.

Command Byte Functions

The 8-bit command byte register (Table 4) is the master index that points to the various other registers within the MAX1617A. The register's POR state is 0000 0000, so that a Receive Byte transmission (a protocol that lacks the command byte) that occurs immediately after POR returns the current local temperature data.

The one-shot command immediately forces a new conversion cycle to begin. In software standby mode (RUN/STOP bit = high), a new conversion is begun, after which the device returns to standby mode. If a conversion is in progress when a one-shot command is received, the command is ignored. If a one-shot command is received in auto-convert mode (RUN/STOP bit = low) between conversions, a new conversion begins, the conversion rate timer is reset, and the next automatic conversion takes place after a full delay elapses.

Configuration Byte Functions

The configuration byte register (Table 5) is used to mask (disable) interrupts and to put the device in software standby mode. The lower six bits are internally set to (XX1111), making them "don't care" bits. Write zeros to these bits. This register's contents can be read back over the serial interface.

Status Byte Functions

The status byte register (Table 6) indicates which (if any) temperature thresholds have been exceeded. This byte also indicates whether or not the ADC is converting and whether there is an open circuit in the remote diode DXP–DXN path. After POR, the normal state of all the flag bits is zero, assuming none of the alarm conditions are present. The status byte is cleared by any

I²C is a trademark of Phillips Corp.

Table 5. Configuration-Byte Bit Assignments

BIT	NAME	POR STATE	FUNCTION
7 (MSB)	MASK	0	Masks all $\overline{\text{ALERT}}$ interrupts when high.
6	RUN/ STOP	0	Standby mode control bit. If high, the device immediately stops converting and enters standby mode. If low, the device converts in either one-shot or timer mode.
5–0	RFU	0	Reserved for future use

Table 6. Status-Byte Bit Assignments

BIT	NAME	FUNCTION
7 (MSB)	BUSY	A high indicates that the ADC is busy converting.
6	LHIGH*	A high indicates that the local high-temperature alarm has activated.
5	LLOW*	A high indicates that the local low-temperature alarm has activated.
4	RHIGH*	A high indicates that the remote high-temperature alarm has activated.
3	RLOW*	A high indicates that the remote low-temperature alarm has activated.
2	OPEN*	A high indicates a remote-diode continuity (open-circuit) fault.
1	RFU	Reserved for future use (returns 0)
0 (LSB)	RFU	Reserved for future use (returns 0)

**These flags stay high until cleared by POR, or until the status byte register is read.*

successful read of the status byte, unless the fault persists. Note that the $\overline{\text{ALERT}}$ interrupt latch is not automatically cleared when the status flag bit is cleared.

When auto-converting, if the T_{HIGH} and T_{LOW} limits are close together, it's possible for both high-temp and low-temp status bits to be set, depending on the amount of time between status read operations (especially when converting at the fastest rate). In these circumstances, it's best not to rely on the status bits to indicate reversals in long-term temperature changes and instead use a current temperature reading to establish the trend direction.

Table 7. Conversion-Rate Control Byte

DATA	CONVERSION RATE (Hz)	AVERAGE SUPPLY CURRENT (µA typ, at V_{CC} = 3.3V)
00h	0.0625	30
01h	0.125	33
02h	0.25	35
03h	0.5	48
04h	1	70
05h	2	128
06h	4	225
07h	8	425
08h to FFh	RFU	—

Conversion Rate Byte

The conversion rate register (Table 7) programs the time interval between conversions in free-running auto-convert mode. This variable rate control reduces the supply current in portable-equipment applications. The conversion rate byte's POR state is 02h (0.25Hz). The MAX1617A looks only at the 3 LSB bits of this register, so the upper 5 bits are "don't care" bits, which should be set to zero. The conversion rate tolerance is ±25% at any rate setting.

Valid A/D conversion results for both channels are available one total conversion time (125ms nominal, 156ms maximum) after initiating a conversion, whether conversion is initiated via the RUN/STOP bit, hardware $\overline{\text{STBY}}$ pin, one-shot command, or initial power-up. Changing the conversion rate can also affect the delay until new results are available (Table 8).

Manufacturer and Device ID Codes

Two ROM registers provide manufacturer and device ID codes (Table 4). Reading the manufacturer ID returns 4Dh, which is the ASCII code "M" (for Maxim). Reading the device ID returns 01h, indicating a MAX1617A device. If READ WORD 16-bit SMBus protocol is employed (rather than the 8-bit READ BYTE), the least significant byte contains the data and the most significant byte contains 00h in both cases.

Slave Addresses

The MAX1617A appears to the SMBus as one device having a common address for both ADC channels. The device address can be set to one of nine different values by pin-strapping ADD0 and ADD1 so that more than one MAX1617A can reside on the same bus without address conflicts (Table 9).

Table 8. RLTS and RRTE Temp Register Update Timing Chart

OPERATING MODE	CONVERSION INITIATED BY:	NEW CONVERSION RATE (CHANGED VIA WRITE TO WCRW)	TIME UNTIL RLTS AND RRTE ARE UPDATED
Auto-Convert	Power-on reset	n/a (0.25Hz)	156ms max
Auto-Convert	1-shot command, while idling between automatic conversions	n/a	156ms max
Auto-Convert	1-shot command that occurs during a conversion	n/a	When current conversion is complete (1-shot is ignored)
Auto-Convert	Rate timer	0.0625Hz	20sec
Auto-Convert	Rate timer	0.125Hz	10sec
Auto-Convert	Rate timer	0.25Hz	5sec
Auto-Convert	Rate timer	0.5Hz	2.5sec
Auto-Convert	Rate timer	1Hz	1.25sec
Auto-Convert	Rate timer	2Hz	625ms
Auto-Convert	Rate timer	4Hz	312.5ms
Auto-Convert	Rate timer	8Hz	237.5ms
Hardware Standby	$\overline{STBY}$ pin	n/a	156ms
Software Standby	RUN/STOP bit	n/a	156ms
Software Standby	1-shot command	n/a	156ms

The address pin states are checked at POR and SPOR only, and the address data stays latched to reduce quiescent supply current due to the bias current needed for high-Z state detection.

The MAX1617A also responds to the SMBus Alert Response slave address (see the *Alert Response Address* section).

POR and UVLO

The MAX1617A has a volatile memory. To prevent ambiguous power-supply conditions from corrupting the data in memory and causing erratic behavior, a POR voltage detector monitors V_{CC} and clears the memory if V_{CC} falls below 1.7V (typical, see *Electrical Characteristics* table). When power is first applied and V_{CC} rises above 1.75V (typical), the logic blocks begin operating, although reads and writes at V_{CC} levels below 3V are not recommended. A second V_{CC} comparator, the ADC UVLO comparator, prevents the ADC from converting until there is sufficient headroom (V_{CC} = 2.8V typical).

The SPOR software POR command can force a power-on reset of the MAX1617A registers via the serial interface. Use the SEND BYTE protocol with COMMAND = FCh. This is most commonly used to reconfigure the slave address of the MAX1617A "on the fly," where external hardware has forced new states at the ADD0 and ADD1 address pins prior to the software POR. The new address takes effect less than 100µs after the SPOR transmission stop condition.

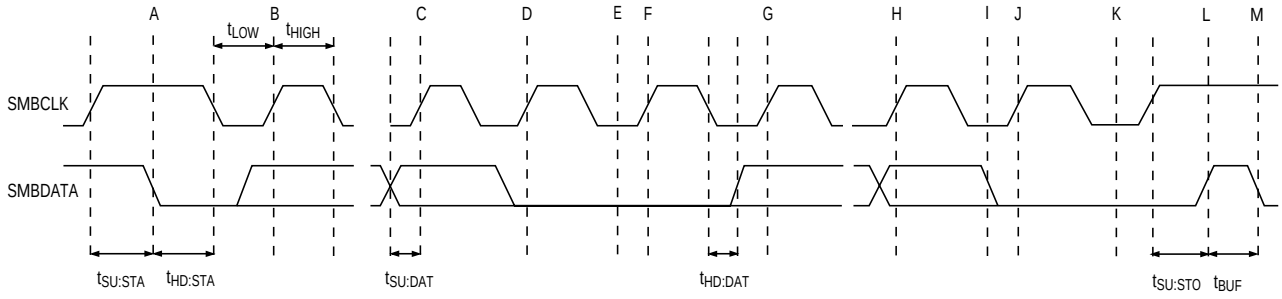
Table 9. Slave Address Decoding (ADD0 and ADD1)

ADD0	ADD1	ADDRESS
GND	GND	0011 000
GND	High-Z	0011 001
GND	V_{CC}	0011 010
High-Z	GND	0101 001
High-Z	High-Z	0101 010
High-Z	V_{CC}	0101 011
V_{CC}	GND	1001 100
V_{CC}	High-Z	1001 101
V_{CC}	V_{CC}	1001 110

Note: High-Z means that the pin is left unconnected and floating.

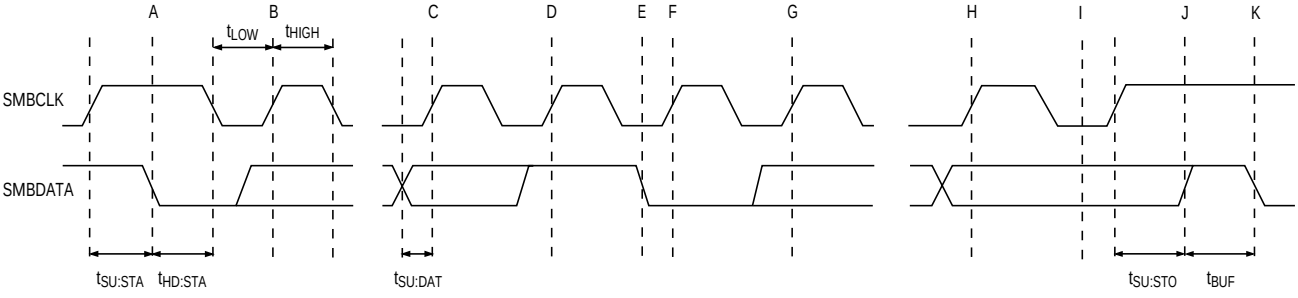
Power-Up Defaults:

- Interrupt latch is cleared.
- Address select pins are sampled.
- ADC begins auto-converting at a 0.25Hz rate.
- Command byte is set to 00h to facilitate quick remote Receive Byte queries.
- T_{HIGH} and T_{LOW} registers are set to max and min limits, respectively.



A = START CONDITION
B = MSB OF ADDRESS CLOCKED INTO SLAVE
C = LSB OF ADDRESS CLOCKED INTO SLAVE
D = R/$\overline{W}$ BIT CLOCKED INTO SLAVE
E = SLAVE PULLS SMBDATA LINE LOW
F = ACKNOWLEDGE BIT CLOCKED INTO MASTER
G = MSB OF DATA CLOCKED INTO SLAVE
H = LSB OF DATA CLOCKED INTO SLAVE
I = SLAVE PULLS SMBDATA LINE LOW
J = ACKNOWLEDGE CLOCKED INTO MASTER
K = ACKNOWLEDGE CLOCK PULSE
L = STOP CONDITION, DATA EXECUTED BY SLAVE
M = NEW START CONDITION

Figure 4. SMBus Write Timing Diagram



A = START CONDITION
B = MSB OF ADDRESS CLOCKED INTO SLAVE
C = LSB OF ADDRESS CLOCKED INTO SLAVE
D = R/$\overline{W}$ BIT CLOCKED INTO SLAVE
E = SLAVE PULLS SMBDATA LINE LOW
F = ACKNOWLEDGE BIT CLOCKED INTO MASTER
G = MSB OF DATA CLOCKED INTO MASTER
H = LSB OF DATA CLOCKED INTO MASTER
I = ACKNOWLEDGE CLOCK PULSE
J = STOP CONDITION
K = NEW START CONDITION

Figure 5. SMBus Read Timing Diagram

```
/* Beginning of the header file which sets the constants */

int     NumStates       = 10;
int     RRTE            = 1;    /* 0x01, command for reading remote temp register */
int     WCA             = 9;    /* 0x09, command for writing configuration register */
int     WCRW            = 10;   /* 0x0A, command for writing conversion rate register */
int     RSL             = 2;    /* 0x02, command for reading status register */
int     WRHA            = 13;   /* 0x0D, command for writing remote THIGH limit register */
int     WRLN            = 14;   /* 0x0E, command for writing remote TLOW limit register */
int     NoError         = 0;
int     Nobody          = 0;
int     MAX1617Addr     = 84;   /* 0x54, default address for MAX1617, ADD0,ADD1=open */
int     InitConfig      = 0;    /* 0x00, configure MAX1617 to MASK=0 and RUN/STOP=0 */
int     InitConv        = 7;    /* 0x07, conversion rate of 8Hz */
int     HighAdder       = 2;    /* 2oC offset for calculating THIGH limit */
int     LowSubtracter   = 4;    /* 4oC offset for calculating TLOW limit */
int     CollisionMask   = 1;    /* 0x01, mask for status bit that indicates collision */
int     DiodeFaultMask  = 4;    /* 0x04, mask for the OPEN diode fault status bit */
int     TempChangeMask  = 24;   /* 0x18, mask for RHIGH and RLOW status bits */

array   State[0..NumStates] of int;

State[0]  = -65 oC      /* At or above this temperature CPU duty cycle is 100% */
State[1]  = 72 oC       /* At or above this temperature CPU duty cycle is 87.5% */
State[2]  = 74 oC       /* At or above this temperature CPU duty cycle is 75% */
State[3]  = 76 oC       /* At or above this temperature CPU duty cycle is 62.5% */
State[4]  = 78 oC       /* At or above this temperature CPU duty cycle is 50% */
State[5]  = 80 oC       /* At or above this temperature CPU duty cycle is 37.5% */
State[6]  = 82 oC       /* At or above this temperature CPU duty cycle is 25% */
State[7]  = 84 oC       /* At or above this temperature CPU duty cycle is 12.5% */
State[8]  = 86 oC       /* At or above this temperature CPU duty cycle is 0.0% */
State[9]  = 88 oC       /* At or above this temperature SHUT SYSTEM OFF! */
State[10] = 127 oC      /* Extra array location so looping is easier */

array   ClockRate[0..NumStates] of real;

ClockRate[0]    = 1.0;
ClockRate[1]    = 0.875;
ClockRate[2]    = 0.75;
ClockRate[3]    = 0.625;
ClockRate[4]    = 0.5;
ClockRate[5]    = 0.375;
ClockRate[6]    = 0.25;
ClockRate[7]    = 0.125;
ClockRate[8]    = 0;
ClockRate[9]    = 0;
ClockRate[10]   = 0;


/* End of the header file */
```

Listing 1. Pseudocode Example

```
int Initialization()
{       int ErrorCode  = NoError;
/* Test the SMBus communications path to the MAX1617 by writing the configuration,
conversion rate and initial temperature limits; if SMBus communication was unsuccessful,
power the system down.  Note that the MAX1617Write procedure takes three parameters: the
command code of the register to be written, the data to write, and a pointer to the the
error code variable.  If the error code variable does not equal NoError before the
execution of MAX1617Write, MAX1617Write does nothing.  If the SMBus communication fails in
MAX1617Write, the error code variable is set to the type of error (for example a NACK,
i.e. MAX1617 did not acknowledge). This code assumes that the BIOS is already in thermal
state 0 (not throttling, i.e. full CPU clock rate) when the initialization routine is
executed. */

        MAX1617Write(WCA, InitConfig, &ErrorCode);    /* MASK=0 and RUN/STOP=0 */
        MAX1617Write(WCRW, InitConv, &ErrorCode);     /* CONV = 8Hz */
        MAX1617Write(WRLN, LowestTemp, &ErrorCode);   /* TLOW = -65oC */
        MAX1617Write(WRHA, State[0] + HighAdder, &ErrorCode) /* THIGH = 72oC */
        if (ErrorCode != NoError) then {
                /* Power off the system */
                }       /* End of if (ErrorCode ... */
        return (ErrorCode);

/* After changing the conversion rate to 8Hz, the MAX1617 temperature register will not
have valid (i.e. current temperature) data for 238 milliseconds. */
}       /* End of Initialization routine */
```

Listing 1. Pseudocode Example (continued)

Programming Example: Clock-Throttling Control for CPUs

Listing 1 gives an untested example of pseudocode for proportional temperature control of Intel mobile CPUs via a power-management microcontroller. This program consists of two main parts: an initialization routine and an interrupt handler. The initialization routine checks for SMBus communications problems and sets up the MAX1617A configuration and conversion rate. The interrupt handler responds to $\overline{\text{ALERT}}$ signals by reading the current temperature and setting a CPU clock duty factor proportional to that temperature. The relationship between clock duty and temperature is fixed in a look-up table contained in the microcontroller code.

Note: Thermal management decisions should be made based on the latest temperature obtained from the MAX1617A rather than the value of the Status Byte. The MAX1617A responds very quickly to changes in its environment due to its sensitivity and its small thermal mass. High and low alarm conditions can exist in the Status Byte due to the MAX1617A correctly reporting environmental changes around it.

```
int ALERT_IntHandler()
{       int ErrorCode  = NoError;
        int WhoDunnit  = Nobody;
        int FoundState = 0;
        int StatusInfo = 0;
        int TempHigh;
        int TempLow;
/* This interrupt handler verifies that the MAX1617 is the source of the interrupt (and
also clears the interrupt) via the SMBus Alert Response address; checks the status byte to
ensure that a temperature change did indeed cause the interrupt; reads the remote
temperature; programs a corresponding clock-throttling duty cycle, and sets up new Thigh
and Tlow limits. */

        ReadAlertResponse(&WhoDunnit, &ErrorCode);
        if (WhoDunnit == MAX1617Addr) then  {

                MAX1617Read(RSL, &StatusInfo, &ErrorCode);

                if (((StatusInfo & CollisionMask) != 0) and (ErrorCode == NoError)) then
                        MAX1617Read(RSL, &StatusInfo, &ErrorCode);

                if (StatusInfo & DiodeFaultMask) != 0) then  {

                        /* Shut down system because thermal diode doesn't work */

                        }
                else if ((StatusInfo & TempChangeMask) != 0) then {

                        MAX1617Read(RRTE, &TempRead, &ErrorCode);
                        while ((TempRead >= State[FoundState + 1]) and
                               (FoundState < (NumStates - 1)) do  FoundState++;
                        if (FoundState == (NumStates - 1)) then  {
                                /* Ahhhhh!!! SHUT SYSTEM OFF!!!! */
                                }
                        else  {
                                /* adjust clock duty cycle */
                                TempHigh = TempRead + HighAdder;
                                TempLow = TempRead - LowSubtracter;
                                MAX1617Write(WRHA, TempHigh, &Error);
                                MAX1617Write(WRLN, TempLow, &Error);
                                }       /* End of if (FoundState ... */

                        }       /* End of if (((StatusInfo .. else if ... */

                /* Handle local temp status bits if set */

                }
        else  {
                /* Handle cases for other interrupt sources */
                }       /* End of if (WhoDunnit ... */

        return(ErrorCode);

}       /* End of Alert_IntHandler interrupt handler routine */
```

Listing 1. Pseudocode Example (continued)