Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

NVT224

dbCOOL™ Remote Thermal Monitor and Fan Controller

The NVT224 dbCOOL controller is a thermal monitor and multiple PWM fan controller for noise–sensitive or power–sensitive applications requiring active system cooling. The NVT224 can drive a fan using either a low or high frequency drive signal, monitor the temperature of up to two remote sensor diodes plus its own internal temperature, and measure and control the speed of up to four fans so that they operate at the lowest possible speed for minimum acoustic noise.

The automatic fan speed control loop optimizes fan speed for a given temperature. The effectiveness of the system's thermal solution can be monitored using the $\overline{\text{THERM}}$ input. The NVT224 also provides critical thermal protection to the system using the bidirectional $\overline{\text{THERM}}$ pin as an output to prevent system or component overheating.

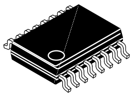
The NVT224 has been through Automotive Qualification according to AEC–Q100 Grade 1 standards.

Features

- Controls and Monitors Up to 4 Fans
- High and Low Frequency Fan Drive Signal
- 1 On–Chip and 2 Remote Temperature Sensors
- Extended Temperature Measurement Range, Up to 191°C
- Automatic Fan Speed Control Mode Controls System Cooling Based on Measured Temperature
- Enhanced Acoustic Mode Dramatically Reduces User Perception of Changing Fan Speeds
- Thermal Protection Feature via $\overline{\text{THERM}}$ Output
- Monitors Performance Impact of Intel Pentium® 4 Processor
- Thermal Control Circuit via $\overline{\text{THERM}}$ Input
- 3–Wire and 4–Wire Fan Speed Measurement
- Limit Comparison of All Monitored Values
- Meets SMBus 2.0 Electrical Specifications (Fully SMBus 1.1 Compliant)
- Automotive Qualification According to AEC–Q100 Grade 1
- These Devices are Pb–Free, Halogen Free/BFR Free and are RoHS Compliant

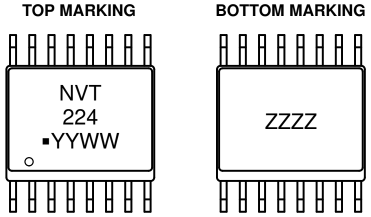
ON Semiconductor®

www.onsemi.com



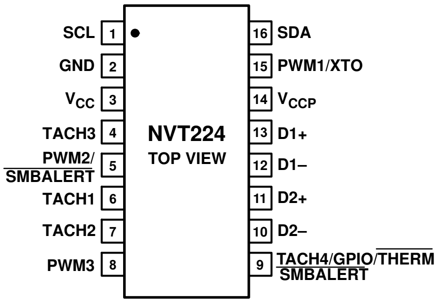
QSOP–16
CASE 492

MARKING DIAGRAMS



NVT224	= Specific Device Code
▪	= Pb–Free Package
YY	= Year
WW	= Work Week
ZZZZ	= Assembly Lot Code

PIN ASSIGNMENT



ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 57 of this data sheet.

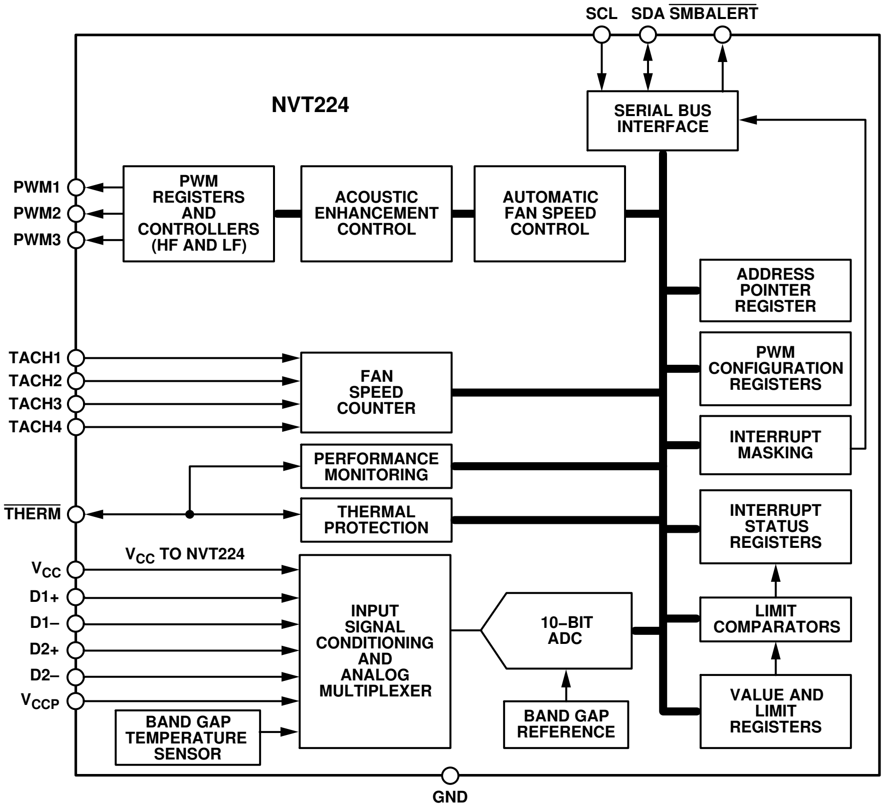


Figure 1. Functional Block Diagram

ABSOLUTE MAXIMUM RATINGS

Parameter	Rating	Unit
Positive Supply Voltage (V_{CC})	3.6	V
Voltage on Any Input or Output Pin	–0.3 to +3.6	V
Input Current at Any Pin	±5	mA
Package Input Current	±20	mA
Maximum Junction Temperature (T_{JMAX})	150	°C
Storage Temperature Range	–65 to +150	°C
Lead Temperature, Soldering IR Reflow Peak Temperature Lead Temperature (Soldering, 10 sec)	 260 300	°C
ESD Rating Human Body Model Machine Model Charged Device Model	 1000 100 1000	V

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

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

THERMAL CHARACTERISTICS

Package Type	θ_{JA}	θ_{JC}	Unit
16–lead QSOP	150	39	°C/W

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

PIN ASSIGNMENT

Pin No.	Mnemonic	Description
1	SCL	Digital Input (Open Drain). SMBus serial clock input. Requires SMBus pullup.
2	GND	Ground Pin.
3	VCC	Power Supply. VCC is also monitored through this pin.
4	TACH3	Digital Input (Open Drain). Fan tachometer input to measure speed of Fan 3.
5	PWM2	PWM2: Digital Output (Open Drain). Requires 10 kΩ typical pullup. Pulse−width modulated output to control Fan 2 speed. Can be configured as a high or low frequency drive.
	$\overline{\text{SMBALERT}}$	$\overline{\text{SMBALERT}}$: Digital Output (Open Drain). This pin can be reconfigured as an $\overline{\text{SMBALERT}}$ interrupt output to signal out−of−limit conditions.
6	TACH1	Digital Input (Open Drain). Fan tachometer input to measure the speed of Fan 1.
7	TACH2	Digital Input (Open Drain). Fan tachometer input to measure the speed of Fan 2.
8	PWM3	Digital I/O (Open Drain). Pulse−width modulated output to control the speed of Fan 3 and Fan 4. Requires 10 kΩ typical pullup. Can be configured as a high or low frequency drive.
9	TACH4 $\overline{\text{THERM}}$	TACH4: Digital Input (Open Drain). Fan tachometer input to measure the speed of Fan 4. $\overline{\text{THERM}}$: Digital I/O (Open Drain). Alternatively, this pin can be reconfigured as a bidirectional $\overline{\text{THERM}}$ pin that can be used to time and monitor assertions on the $\overline{\text{THERM}}$ input. For example, the pin can be connected to the $\overline{\text{PROCHOT}}$ output of an Intel Pentium 4 processor or to the output of a trip point temperature sensor. This pin can be used as an output to signal overtemperature conditions.
	GPIO $\overline{\text{SMBALERT}}$	GPIO: General−Purpose Open Drain Digital I/O. $\overline{\text{SMBALERT}}$: Digital Output (Open Drain). This pin can be reconfigured as an $\overline{\text{SMBALERT}}$ interrupt output to signal out−of−limit conditions.
10	D2−	Cathode Connection to Second Thermal Diode.
11	D2+	Anode Connection to Second Thermal Diode.
12	D1−	Cathode Connection to First Thermal Diode.
13	D1+	Anode Connection to First Thermal Diode.
14	VCCP	Analog Input. Monitors processor core voltage (0 V to 3.0 V).
15	PWM1	Digital Output (Open Drain). Pulse−width modulated output to control Fan 1 speed. Requires 10 kΩ typical pullup.
	XTO	Also functions as the output from the XNOR tree in XNOR test mode.
16	SDA	Digital I/O (Open Drain). SMBus bidirectional serial data. Requires 10 kΩ typical pullup.

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

Parameter	Conditions	Min	Typ	Max	Unit
Power Supply					
Supply Voltage		3.0	3.3	3.6	V
Supply Current, I_{CC}	Interface inactive, ADC active		1.5	3.0	mA
Temperature−to−Digital Converter					
Local Sensor Accuracy	0°C ≤ T_A ≤ 85°C −40°C ≤ T_A ≤ +125°C		±0.5	±1.5 ±2.5	°C
Resolution			0.25		
Remote Diode Sensor Accuracy	0°C ≤ T_A ≤ 85°C −40°C ≤ T_A ≤ +125°C		±0.5	1.5 ±2.5	°C
Resolution			0.25		
Remote Sensor Source Current	High level Low level		180 11		μA
ANALOG−TO−DIGITAL CONVERTER (INCLUDING MUX AND ATTENUATORS)					
Total Unadjusted Error (TUE)				±2	%
Differential Non−linearity (DNL)	8 bits			±1	LSB
Power Supply Sensitivity			±0.1		%/V
Conversion Time (Voltage Input)	Averaging enabled		11		ms
Conversion Time (Local Temperature)	Averaging enabled		12		ms
Conversion Time (Remote Temperature)	Averaging enabled		38		ms
Total Monitoring Cycle Time	Averaging enabled Averaging disabled		145 19		ms
Input Resistance	For V_{CCP} channel	70	120		kΩ
FAN RPM−TO−DIGITAL CONVERTER					
Accuracy	0°C ≤ T_A ≤ 70°C −40°C ≤ T_A ≤ +120°C			±6 ±10	%
Full−Scale Count				65,535	
Nominal Input RPM	Fan count = 0xBFFF Fan count = 0x3FFF Fan count = 0x0438 Fan count = 0x021C		109 329 5000 10,000		RPM
OPEN−DRAIN DIGITAL OUTPUTS (PWM1 TO PWM3, XTO)					
Current Sink, I_{OL}				8.0	mA
Output Low Voltage, V_{OL}	I_{OUT} = −8.0 mA			0.4	V
High Level Output Current, I_{OH}	$V_{OUT} = V_{CC}$		0.1	20	μA
OPEN−DRAIN SERIAL DATA BUS OUTPUT (SDA)					
Output Low Voltage, V_{OL}	I_{OUT} = −4.0 mA			0.4	V
High Level Output Current, I_{OH}	$V_{OUT} = V_{CC}$		0.1	1.0	μA
SMBus DIGITAL INPUTS (SCL, SDA)					
Input High Voltage, V_{IH}		2.0			V
Input Low Voltage, V_{IL}				0.4	V
Hysteresis			500		mV
DIGITAL INPUT LOGIC LEVELS (TACH INPUTS)					
Input High Voltage, V_{IH}		2.0			V
	Maximum input voltage			3.6	
Input Low Voltage, V_{IL}				0.8	V
	Minimum input voltage	−0.3			
Hysteresis			0.5		V p−p

1. All voltages are measured with respect to GND, unless otherwise specified. Typicals are at T_A = 25°C and represent the most likely parametric norm. Logic inputs accept input high voltages of up to V_{MAX}, even when the device is operating down to V_{MIN}. Timing specifications are tested at logic levels of V_{IL} = 0.8 V for a falling edge and V_{IH} = 2.0 V for a rising edge.
2. SMBus timing specifications are guaranteed by design and are not production tested.

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

Parameter	Conditions	Min	Typ	Max	Unit
DIGITAL INPUT LOGIC LEVELS ($\overline{\text{THERM}}$) ADTL+					
Input High Voltage, V_{IH}		0.75 x V_{CC}			V
Input Low Voltage, V_{IL}				0.8	V
DIGITAL INPUT CURRENT					
Input High Current, I_{IH}	$V_{IN} = V_{CC}$		±1		μA
Input Low Current, I_{IL}	$V_{IN} = 0$ V		±1		μA
Input Capacitance, C_{IN}			5		pF
SERIAL BUS TIMING	See Note 2 and Figure 2				
Clock Frequency, f_{SCLK}		10		400	kHz
Glitch Immunity, t_{SW}				50	ns
Bus Free Time, t_{BUF}		4.7			μs
SCL Low Time, t_{LOW}		4.7			μs
SCL High Time, t_{HIGH}		4.0		50	μs
SCL, SDA Rise Time, t_R				1000	ns
SCL, SDA Fall Time, t_F				300	ns
Data Setup Time, $t_{SU:\ DAT}$		250			ns
Detect Clock Low Timeout, $t_{TIMEOUT}$	Can be optionally disabled	15		35	ms

1. All voltages are measured with respect to GND, unless otherwise specified. Typicals are at T_A = 25°C and represent the most likely parametric norm. Logic inputs accept input high voltages of up to V_{MAX}, even when the device is operating down to V_{MIN}. Timing specifications are tested at logic levels of V_{IL} = 0.8 V for a falling edge and V_{IH} = 2.0 V for a rising edge.
2. SMBus timing specifications are guaranteed by design and are not production tested.

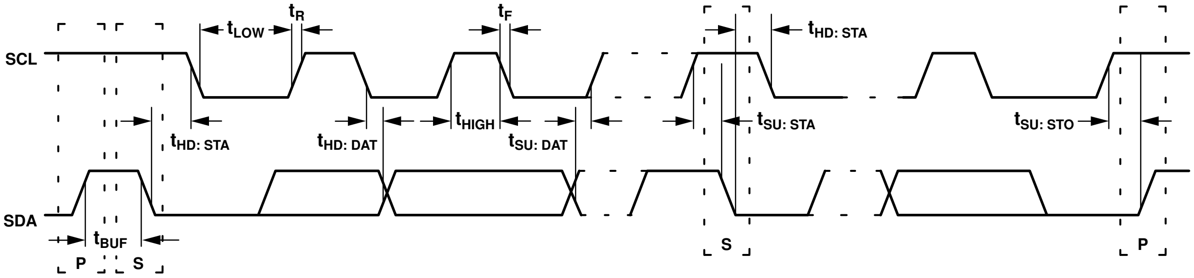


Figure 2. Serial Bus Timing Diagram

TYPICAL PERFORMANCE CHARACTERISTICS

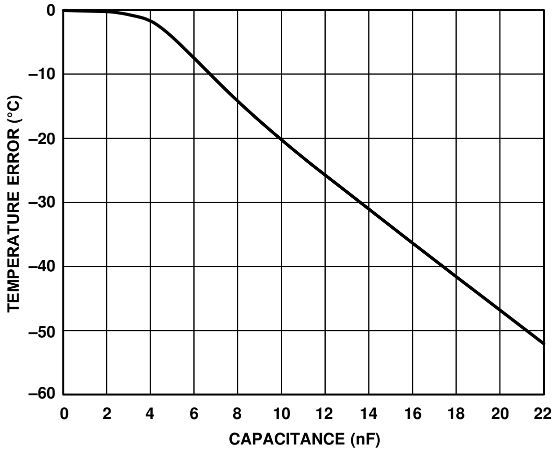


Figure 3. Temperature Error vs. Capacitance Between D+ and D−

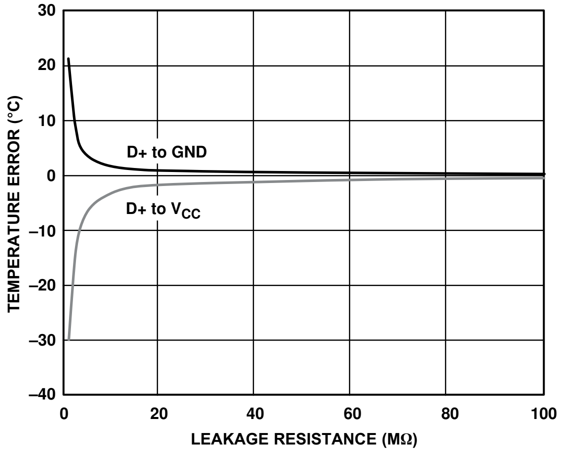


Figure 4. Remote Temperature Error vs. PCB Resistance

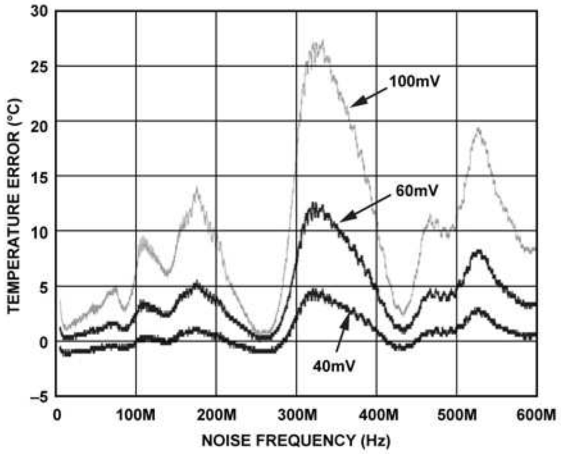


Figure 5. Remote Temperature Error vs. Common−Mode Noise Frequency

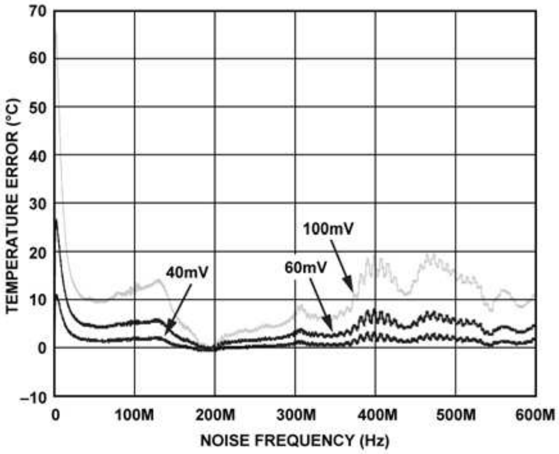


Figure 6. Remote Temperature Error vs. Differential Mode Noise Frequency

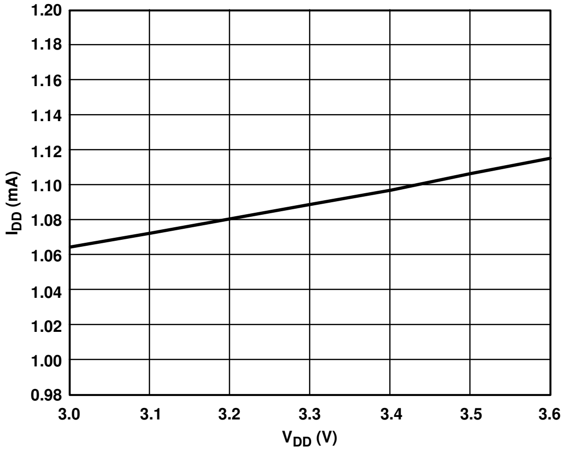


Figure 7. Normal I_{DD} vs. Power Supply

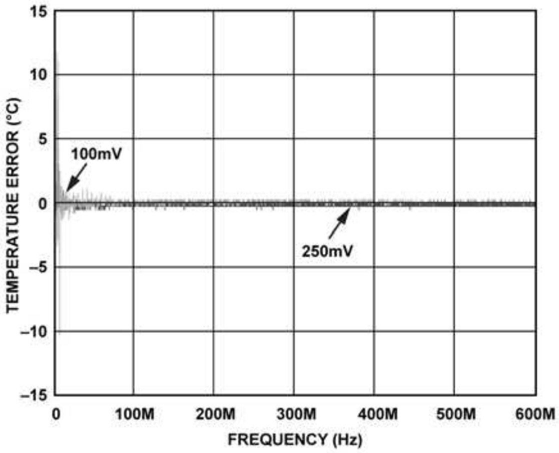


Figure 8. Internal Temperature Error vs. Power Supply Noise

TYPICAL PERFORMANCE CHARACTERISTICS

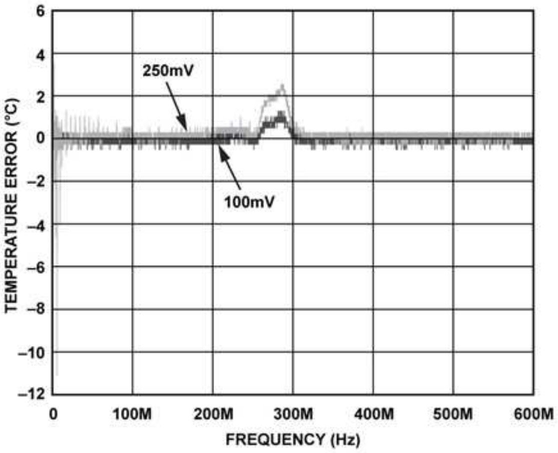


Figure 9. Remote Temperature Error vs. Power Supply Noise Frequency

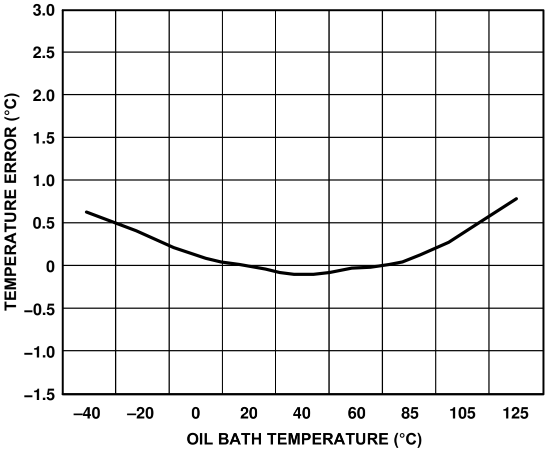


Figure 10. Internal Temperature Error vs. NVT224 Temperature

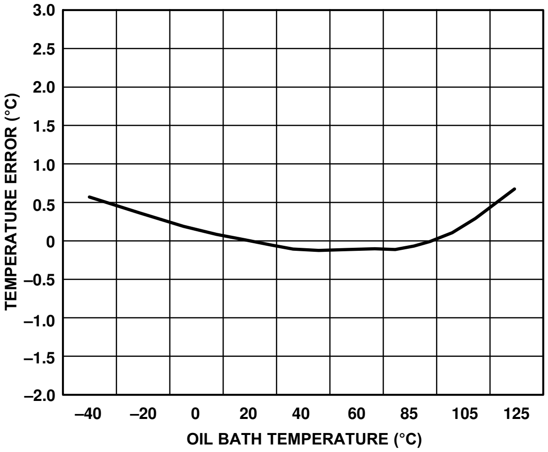


Figure 11. Remote Temperature Error vs. NVT224 Temperature

Product Description

The NVT224 is a complete thermal monitor and multiple fan controller for any system requiring thermal monitoring and cooling. The device communicates with the system via a serial system management bus. The serial bus controller has a serial data line for reading and writing addresses and data (Pin 16), and an input line for the serial clock (Pin 1). All control and programming functions for the NVT224 are performed over the serial bus. In addition, a pin can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out−of−limit conditions.

Quick Comparison Between ADT7473 and NVT224

- The ADT7473 supports advanced dynamic T_{MIN} features while the NVT224 does not.
- Acoustic smoothing is improved on the NVT224.
- $\overline{\text{THERM}}$ can be selected as an output only on the NVT224.
- The NVT224 has two additional configuration registers.
- The NVT224 has other minor register changes.

The NVT224 is similar to the ADT7473 in that it is powered by a supply no greater than 3.6 V. Exceeding this specification results in irreversible damage to the NVT224. Signal pins (TACH/PWM) should be pulled up or clamped to 3.6 V maximum. See the Specifications Section for more information.

Recommended Implementation

Configuring the NVT224 as shown in Figure 12 allows the system designer to use the following features:

- Two PWM outputs for fan control of up−to−three fans (the front and rear chassis fans are connected in parallel).
- Three TACH fan speed measurement inputs.
- V_{CC} measured internally through Pin 3.
- CPU temperature measured using the Remote 1 temperature channel.
- Ambient temperature measured through the Remote 2 temperature channel.
- Bidirectional $\overline{\text{THERM}}$ pin. This feature allows Intel Pentium 4 $\overline{\text{PROCHOT}}$ monitoring and can function as an overtemperature $\overline{\text{THERM}}$ output. The $\overline{\text{THERM}}$ pin can alternatively be programmed as an $\overline{\text{SMBALERT}}$ system interrupt output.

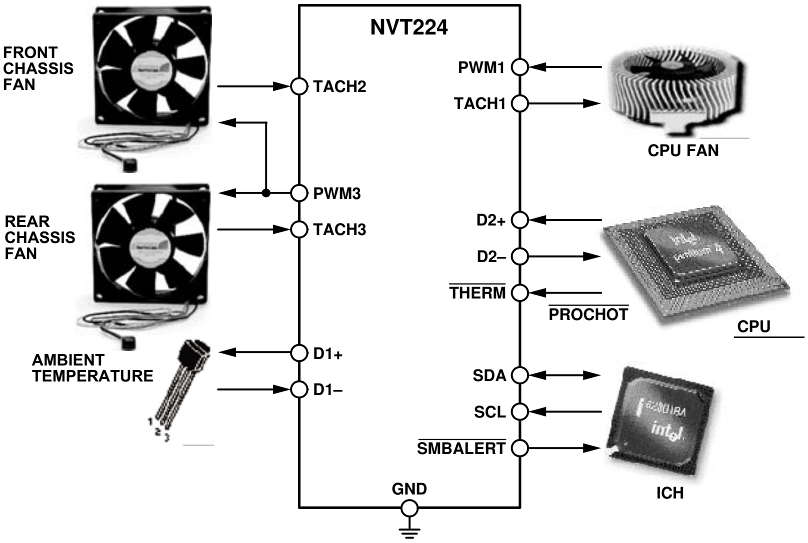


Figure 12. NVT224 Configuration

Serial Bus Interface

On PCs and servers, control of the NVT224 is carried out using the SMBus. The NVT224 is connected to this bus as a slave device under the control of a master controller, which is usually (but not necessarily) the ICH.

The NVT224 has a fixed 7−bit serial bus address of 0101110 or 0x2E. The read/write bit must be added to get the 8−bit address (01011100 or 0x5C). Data is sent over the serial bus in sequences of nine clock pulses, that is, eight bits of data followed by an acknowledge bit from the slave device. Transitions on the data line must occur during the low period of the clock signal and remain stable during the high period because 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.

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

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

In the NVT224, write operations contain either one or two bytes, and read operations contain one byte. 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, and then data can be written to that register or read from it. The first byte of a write operation always contains an address that is stored in the address pointer register. If data is to be written to the device, the write operation contains a second data byte that is written to the register selected by the address pointer register.

This write operation is shown in Figure 13. The device address is sent over the bus, and then R/$\overline{\text{W}}$ is set to 0. This is followed by two data bytes. The first data byte is the address of the internal data register to write to, which is stored in the address pointer register. The second data byte is the data to write to the internal data register.

When reading data from a register, there are two possibilities:

- If the NVT224 address pointer register value is unknown or not the desired value, it must first be set to the correct value before data can be read from the desired data register. This is done by performing a write to the NVT224 as before, but only the data byte containing the register address is sent, because no data is written to the register see Figure 14.
  A read operation is then performed consisting of the serial bus address; the R/$\overline{\text{W}}$ bit set to 1, followed by the data byte read from the data register see Figure 15.
- 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 see Figure 15.

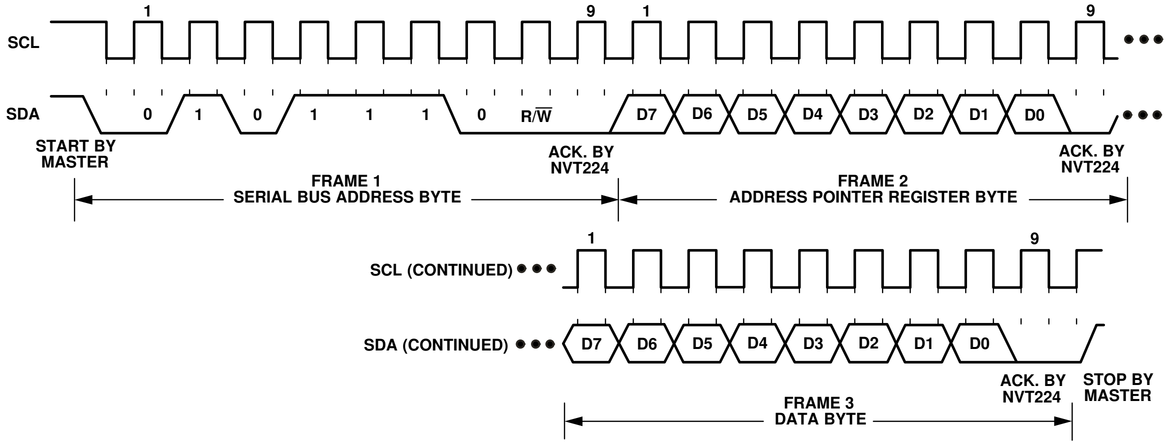


Figure 13. Writing a Register Address to the Address Pointer Register, then Writing Data to the Selected Register

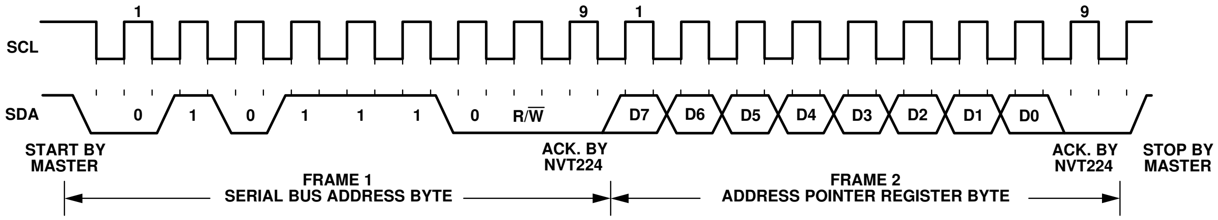


Figure 14. Writing to the Address Pointer Register Only

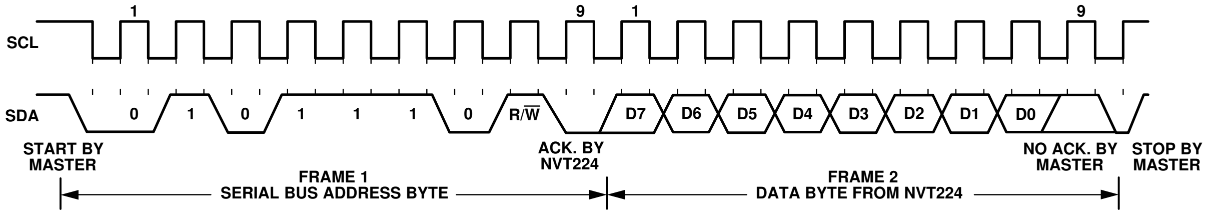


Figure 15. Reading Data from a Previously Selected Register

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. However, 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.

In addition to supporting the send byte and receive byte protocols, the NVT224 also supports the read byte protocol (for more information, see System Management Bus Specifications Rev. 2.0, available from Intel).

If several read or write operations must be performed in succession, the master can send a repeat start condition instead of a stop condition to begin a new operation.

Write Operations

The SMBus specification defines several protocols for different types of read and write operations. The ones used in the NVT224 are discussed in this section. The following abbreviations are used in the diagrams:

S—Start
P—Stop
R—Read
W—Write
A—Acknowledge
$\overline{\text{A}}$—No acknowledge

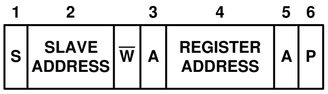
The NVT224 uses the following SMBus write protocols.

Send Byte

In this operation, the master device sends a single command byte to a slave device as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7−bit slave address followed by the write bit (low).
3. The addressed slave device asserts ACK on SDA.
4. The master sends a command code.
5. The slave asserts ACK on SDA.
6. The master asserts a stop condition on SDA and the transaction ends.

For the NVT224, the send byte protocol is used to write a register address to RAM for a subsequent single−byte read from the same address. This operation is shown in Figure 16.



1	2		3	4	5	6
S	SLAVE ADDRESS	$\overline{\text{W}}$	A	REGISTER ADDRESS	A	P

Figure 16. Setting a Register Address for Subsequent Read

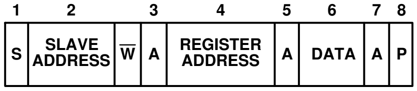
If the master is required to read data from the register immediately after setting up the address, it can assert a repeat start condition immediately after the final ACK and carry out a single−byte read without asserting an intermediate stop condition.

Write Byte

In this operation, the master device sends a command byte and one data byte to the slave device as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7−bit slave address followed by the write bit (low).
3. The addressed slave device asserts ACK on SDA.
4. The master sends a command code.
5. The slave asserts ACK on SDA.
6. The master sends a data byte.
7. The slave asserts ACK on SDA.
8. The master asserts a stop condition on SDA and the transaction ends.

The byte write operation is shown in Figure 17.



1	2		3	4	5	6	7	8
S	SLAVE ADDRESS	$\overline{\text{W}}$	A	REGISTER ADDRESS	A	DATA	A	P

Figure 17. Single−Byte Write to a Register

Read Operations

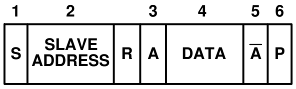
The NVT224 uses the following SMBus read protocols.

Receive Byte

This operation is useful when repeatedly reading a single register. The register address must be set up previously. In this operation, the master device receives a single byte from a slave device as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7−bit slave address followed by the read bit (high).
3. The addressed slave device asserts ACK on SDA.
4. The master receives a data byte.
5. The master asserts NO ACK on SDA.
6. The master asserts a stop condition on SDA, and the transaction ends.

In the NVT224, the receive byte protocol is used to read a single byte of data from a register whose address has previously been set by a send byte or write byte operation. This operation is shown in Figure 18.



1	2		3	4	5	6
S	SLAVE ADDRESS	R	A	DATA	$\overline{\text{A}}$	P

Figure 18. Single−Byte Read from a Register

Alert Response Address

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

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

1. $\overline{\text{SMBALERT}}$ is pulled low.
2. The master initiates a read operation and sends the alert response address (ARA = 0001 100). This general call address must not be used as a specific device address.
3. The device whose $\overline{\text{SMBALERT}}$ output is low responds to the alert response address, and the master reads its device address. The address of the device is now known and can be interrogated in the usual way.
4. If more than one device's $\overline{\text{SMBALERT}}$ output is low, the one with the lowest device address has priority in accordance with normal SMBus arbitration.
5. Once the NVT224 has responded to the alert response address, the master must read the status registers, and the $\overline{\text{SMBALERT}}$ is cleared only if the error condition has gone away.

SMBus Timeout

The NVT224 includes an SMBus timeout feature. If there is no SMBus activity for 35 ms, the NVT224 assumes that the bus is locked and releases the bus. This prevents the device from locking or holding the SMBus expecting data. Some SMBus controllers cannot handle the SMBus timeout feature, so it can be disabled.

Configuration Register 1 (0x40)

Bit 6 TODIS = 0; SMBus timeout enabled (default).

Bit 6 TODIS = 1; SMBus timeout disabled.

Virus Protection

To prevent rogue programs or viruses from accessing critical NVT224 register settings, the lock bit can be set. Setting Bit 1 of Configuration Register 1 (0x40) sets the lock bit and locks critical registers. In this mode, certain registers can no longer be written to until the NVT224 is powered down and powered up again. For more information on which registers are locked, see the Register Tables section.

Voltage Measurement Input

The NVT224 has one external voltage measurement channel. It can also measure its own supply voltage, V_{CC}. Pin 14 can measure V_{CCP}. The V_{CC} supply voltage measurement is carried out through the V_{CC} pin (Pin 3). The V_{CCP} input can be used to monitor a chipset supply voltage in computer systems.

Analog−to−Digital Converter

All analog inputs are multiplexed into the on−chip, successive approximation, analog−to−digital converter. This has a resolution of 10 bits. The basic input range is 0 V to 2.25 V, but the input has built−in attenuators to allow measurement of V_{CCP} without any external components. To allow for the tolerance of the supply voltage, the ADC produces an output of 3/4 full scale (decimal 768 or 300 hex) for the nominal input voltage and so has adequate headroom to deal with overvoltages.

Input Circuitry

The internal structure for the V_{CCP} analog input is shown in Figure 19. The input circuit consists of an input protection diode, an attenuator, and a capacitor to form a first−order, low−pass filter that gives the input immunity to high frequency noise.

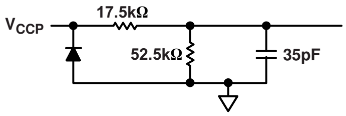


Figure 19. Structure of Analog Inputs

Voltage Measurement Registers

Register 0x21, V_{CCP} Reading = 0x00 default

Register 0x22, V_{CC} Reading = 0x00 default

V_{CCP} Limit Registers

Associated with the V_{CCP} measurement channel is a high and low limit register. Exceeding the programmed high or low limit causes the appropriate status bit to be set. Exceeding either limit can also generate $\overline{\text{SMBALERT}}$ interrupts.

Register 0x46, V_{CCP} Low Limit = 0x00 default

Register 0x47, V_{CCP} High Limit = 0xFF default

Table 2 shows the input ranges of the analog inputs and output codes of the 10−bit ADC.

When the ADC is running, it samples and converts a voltage input in 711 μs and averages 16 conversions to reduce noise; a measurement takes nominally 11.38 ms.

Extended Resolution Registers

Voltage measurements can be made with higher accuracy using the extended resolution registers (0x76 and 0x77). Whenever the extended resolution registers are read, the corresponding data in the voltage measurement registers is locked until their data is read. That is, if extended resolution is required, then the extended resolution register must be read first, immediately followed by the appropriate voltage measurement register.

Additional ADC Functions for Voltage Measurements

A number of other functions are available on the NVT224 to offer the system designer increased flexibility.

Turn−Off Averaging

For each voltage measurement read from a value register, 16 readings have been made internally, and the results averaged, before being placed into the value register. For instances where faster conversions are needed, setting Bit 4 of Configuration Register 2 (0x73) turns averaging off.

This effectively gives a reading 16 times faster (711 μs), but the reading may be noisier.

Bypass Voltage Input Attenuator

Setting Bit 5 of Configuration Register 2 (0x73) removes the attenuation circuitry from the V_{CCP} input. This allows the user to directly connect external sensors or to rescale the analog voltage measurement inputs for other applications. The input range of the ADC without the attenuators is 0 V to 2.25 V.

Single−Channel ADC Conversion

Setting Bit 6 of Configuration Register 2 (0x73) places the NVT224 into single−channel ADC conversion mode. In this mode, the NVT224 can be made to read a single voltage channel only. If the internal NVT224 clock is used, the selected input is read every 711 μs. The appropriate ADC channel is selected by writing to Bits [7:5] of the TACH1 minimum high byte register (0x55).

Table 1. Single−Channel ADC Conversion

Register 0x55, Bits [7:5]	Channel Selected
001	V_{CCP}
010	V_{CC}
101	Remote 1 Temperature
110	Local Temperature
111	Remote 2 Temperature

Configuration Register 2 (0x73)

Bit 4 = 1; averaging off.

Bit 5 = 1; bypass input attenuators.

Bit 6 = 1; single−channel convert mode.

TACH1 Minimum High Byte (0x55)

Bits [7:5] select the ADC channel for single−channel convert mode.

Table 2. 10−Bit ADC Output Code vs. V_{IN}

ADC Output			
V_{CC} (3.3 V_{IN}) (Note 1)	V_{CCP}	Decimal	Binary (10 Bits)
<0.0042	<0.00293	0	00000000 00
0.0042 to 0.0085	0.0293 to 0.0058	1	00000000 01
0.0085 to 0.0128	0.0058 to 0.0087	2	00000000 10
0.0128 to 0.0171	0.0087 to 0.0117	3	00000000 11
0.0171 to 0.0214	0.0117 to 0.0146	4	00000001 00
0.0214 to 0.0257	0.0146 to 0.0175	5	00000001 01
0.0257 to 0.0300	0.0175 to 0.0205	6	00000001 10
0.0300 to 0.0343	0.0205 to 0.0234	7	00000001 11
0.0343 to 0.0386	0.0234 to 0.0263	8	00000010 00
–	–	–	–
1.100 to 1.1042	0.7500 to 0.7529	256 (1/4 scale)	01000000 00
–	–	–	–
2.200 to 2.2042	1.5000 to 1.5029	512 (1/2 scale)	10000000 00
–	–	–	–
3.300 to 3.3042	2.2500 to 2.2529	768 (3/4 scale)	11000000 00
–	–	–	–
4.3527 to 4.3570	2.9677 to 2.9707	1013	11111101 01
4.3570 to 4.3613	2.9707 to 2.9736	1014	11111101 10
4.3613 to 4.3656	2.9736 to 2.9765	1015	11111101 11
4.3656 to 4.3699	2.9765 to 2.9794	1016	11111110 00
4.3699 to 4.3742	2.9794 to 2.9824	1017	11111110 01
4.3742 to 4.3785	2.9824 to 2.9853	1018	11111110 10
4.3785 to 4.3828	2.9853 to 2.9882	1019	11111110 11
4.3828 to 4.3871	2.9882 to 2.9912	1020	11111111 00
4.3871 to 4.3914	2.9912 to 2.9941	1021	11111111 01
4.3914 to 4.3957	2.9941 to 2.9970	1022	11111111 10
>4.3957	>2.9970	1023	11111111 11

1. The V_{CC} output codes listed assume that V_{CC} is 3.3 V and that V_{CC} should never exceed 3.6 V.

Temperature Measurement Method

Local Temperature Measurement

The NVT224 contains an on−chip band gap temperature sensor whose output is digitized by the on−chip, 10−bit ADC. The 8−bit MSB temperature data is stored in the temperature registers (0x25, 0x26, and 0x27). Because both positive and negative temperatures can be measured, the temperature data is stored in Offset 64 format or twos complement format, as shown in Table 3 and Table 4.

Theoretically, the temperature sensor and ADC can measure temperatures from −128°C to +127°C (or −64°C to +191°C in the extended temperature range) with a resolution of 0.25°C. However, this exceeds the operating temperature range of the device, so local temperature measurements outside the NVT224 operating temperature range are not possible.

Remote Temperature Measurement

The NVT224 can measure the temperature of two remote diode sensors or diode−connected transistors connected to Pin 10 and Pin 11 or to Pin 12 and Pin 13.

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

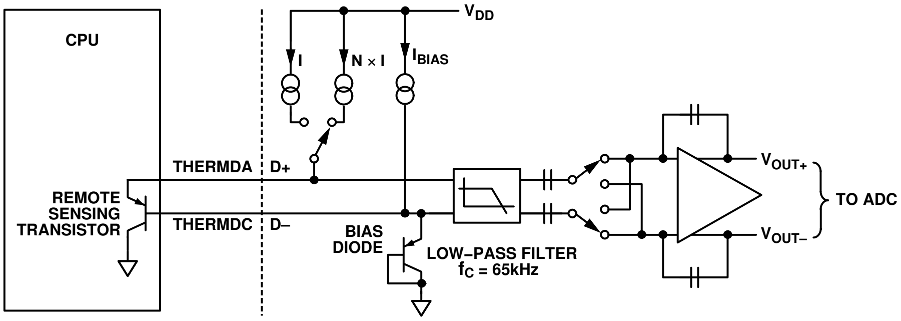


Figure 20. Signal Conditioning for Remote Diode Temperature Sensors

The technique used in the NVT224 is to measure the change in V_{BE} when the device is operated at two different currents.

This is given by:

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

where:

k is Boltzmann’s constant.
q is the charge on the carrier.
T is the absolute temperature in Kelvin.
N is the ratio of the two currents.

Figure 20 shows the input signal conditioning used to measure the output of a remote temperature sensor. This figure shows the external sensor as a substrate transistor, provided for temperature monitoring on some microprocessors. It could also be a discrete transistor such as a 2N3904/2N3906.

If a discrete transistor is used, the collector is not grounded and should be linked to the base. If a PNP transistor is used, the base is connected to the D input and the emitter to the D+ input. If an NPN transistor is used, the emitter is connected to the D input and the base to the D+ input. Figure 21 and Figure 22 show how to connect the NVT224 to an NPN or PNP transistor for temperature measurement. To prevent ground noise from 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.

To measure ΔV_{BE}, the sensor is switched between operating currents of I and N x I. The resulting waveform is passed through a 65 kHz low−pass filter to remove noise and 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 10−bit, twos complement format. To further reduce the effects of noise, digital filtering is performed by averaging the results of 16 measurement cycles.

A remote temperature measurement takes nominally 38 ms. The results of remote temperature measurements are stored in 10–bit, twos complement format, as shown in Table 3. The extra resolution for the temperature measurements is held in the Extended Resolution Register 2 (0x77). This gives temperature readings with a resolution of 0.25°C.

Noise Filtering

For temperature sensors operating in noisy environments, previous practice was to place a capacitor across the D+ pin and D− pin to help combat the effects of noise. However, large capacitance’s affect the accuracy of the temperature measurement, leading to a recommended maximum capacitor value of 1000 pF.

This capacitor reduces the noise but does not eliminate it. Sometimes, this sensor noise is a problem in a very noisy environment. In most cases, a capacitor is not required because differential inputs, by their very nature, have a high immunity to noise.

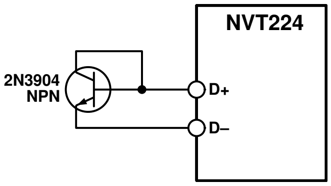


Figure 21. Measuring Temperature Using an NPN Transistor

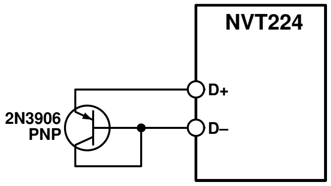


Figure 22. Measuring Temperature Using a PNP Transistor

Factors Affecting Diode Accuracy

Remote Sensing Diode

The NVT224 is designed to work with either substrate transistors built into processors or with discrete transistors. Substrate transistors are generally PNP types with the collector connected to the substrate. Discrete types can be either PNP or NPN transistors connected as a diode (base–shorted to the collector). If an NPN transistor is used, the collector and base are connected to D+ and the emitter to D–. If a PNP transistor is used, the collector and base are connected to D– and the emitter is connected to D+.

To reduce the error due to variations in both substrate and discrete transistors, a number of factors should be taken into consideration:

- The ideality factor, n_f, of the transistor is a measure of the deviation of the thermal diode from ideal behavior. The NVT224 is trimmed for an n_f value of 1.008. Use the following equation to calculate the error introduced at a temperature, T (°C), when using a transistor whose n_f does not equal 1.008. See the processor data sheet for the n_f values.

  $$\Delta T = (n_f - 1.008) \times (273.15\ k + T) \qquad \text{(eq. 2)}$$

  To factor this in, the user can write the ΔT value to the offset register. The NVT224 automatically adds it to or subtracts it from the temperature measurement.
- Some CPU manufacturers specify the high and low current levels of the substrate transistors. The high current level of the NVT224, I_{HIGH}, is 180 μA and the low level current, I_{LOW}, is 11 μA. If the NVT224 current levels do not match the current levels specified by the CPU manufacturer, it might be necessary to remove an offset. The CPU's data sheet advises whether this offset needs to be removed and how to calculate it. This offset can be programmed to the offset register. If more than one offset must be considered, the algebraic sum of these offsets must be programmed to the offset register.

If a discrete transistor is used with the NVT224, the best accuracy is obtained by choosing devices according to the following criteria:

- Base–emitter voltage greater than 0.25 V at 11 μA, at the highest operating temperature.
- Base–emitter voltage less than 0.95 V at 180 μA, at the lowest operating temperature.
- Base resistance less than 100 Ω.
- Small variation in h_{FE} (approximately 50 to 150) that indicates tight control of V_{BE} characteristics.

Transistors, such as 2N3904, 2N3906, or equivalents in SOT–23 packages, are suitable devices to use.

Table 3. Twos Complement Temperature Data Format

Temperature	**Digital Output (10–Bit)** (Note 1)
–128°C	1000 0000 **00** (diode fault)
–63°C	1100 0001 **00**
–50°C	1100 1110 **00**
–25°C	1110 0111 **00**
–10°C	1111 0110 **00**
0°C	0000 0000 **00**
10.25°C	0000 1010 **01**
25.5°C	0001 1001 **10**
50.75°C	0011 0010 **11**
75°C	0100 1011 **00**
100°C	0110 0100 **00**
125°C	0111 1101 **00**
127°C	0111 1111 **00**

1. Bold numbers denote 2 LSBs of measurement in the Extended Resolution Register 2 (0x77) with 0.25°C resolution.

Table 4. Extended Range, Temperature Data Format

Temperature	**Digital Output (10–Bit)** (Note 1)
–64°C	0000 0000 **00** (diode fault)
–63°C	0000 0001 **00**
–1°C	0011 1111 **00**
0°C	0100 0000 **00**
1°C	0100 0001 **00**
10°C	0100 1010 **00**
25°C	0101 1001 **00**
50°C	0111 0010 **00**
75°C	1000 1001 **00**
100°C	1010 0100 **00**
125°C	1011 1101 **00**
191°C	1111 1111 **00**

1. Bold numbers denote 2 LSBs of measurement in the Extended Resolution Register 2 (0x77) with 0.25°C resolution.

Nulling Out Temperature Errors

As CPUs run faster, it is more difficult to avoid high frequency clocks when routing the D+/D– traces around a system board. Even when recommended layout guidelines are followed, some temperature errors can still be attributable to noise coupled onto the D+/D– lines. Constant high frequency noise usually attenuates, or increases, temperature measurements by a linear, constant value.

The NVT224 has two temperature offset registers, Register 0x70 and Register 0x72, for the Remote 1 and Remote 2 temperature channels. By doing a one–time calibration of the system, the user can determine the offset caused by system board noise and null it out using the offset registers. The offset registers automatically add a twos complement 8–bit reading to every temperature measurement.

Changing Bit 1 of Configuration Register 5 (0x7C) changes the resolution and therefore the range of the temperature offset as either having a range of –63°C to +127°C, with a resolution of 1°C, or having a range of –63°C to +64°C, with a resolution of 0.5°C. This temperature offset can be used to compensate for linear temperature errors introduced by noise.

Temperature Offset Registers

Register 0x70, Remote 1 Temperature Offset = 0x00 (0°C default)

Register 0x71, Local Temperature Offset = 0x00 (0°C default)

Register 0x72, Remote 2 Temperature Offset = 0x00 (0°C default)

ADT7463/NVT224 Backwards Compatible Mode

By setting Bit 0 of Configuration Register 5 (0x7C), all temperature measurements are stored in the zone temperature value registers (0x25, 0x26, and 0x27) in twos complement in the range –128°C to +127°C. The temperature limits must be reprogrammed in twos complement.

If a twos complement temperature below –128°C is entered, the temperature is clamped to –128°C. In this mode, the diode fault condition remains –128°C = 1000 0000, while in the extended temperature range (–64°C to +191°C), the fault condition is represented by –64°C = 0000 0000.

Temperature Measurement Registers

Register 0x25, Remote 1 Temperature
Register 0x26, Local Temperature
Register 0x27, Remote 2 Temperature
Register 0x77, Extended Resolution 2 = 0x00 default
Bits [7:6] TDM2, Remote 2 Temperature LSBs.
Bits [5:4] LTMP, Local Temperature LSBs.
Bits [3:2] TDM1, Remote 1 Temperature LSBs.

Temperature Measurement Limit Registers

Associated with each temperature measurement channel are high and low limit registers. Exceeding the programmed high or low limit causes the appropriate status bit to be set. Exceeding either limit can also generate $\overline{\text{SMBALERT}}$ interrupts (depending on the way the interrupt mask register is programmed and assuming that $\overline{\text{SMBALERT}}$ is set as an output on the appropriate pin).

Register 0x4E, Remote 1 Temperature Low Limit = 0x81 default

Register 0x4F, Remote 1 Temperature High Limit = 0x7F default

Register 0x50, Local Temperature Low Limit = 0x81 default

Register 0x51, Local Temperature High Limit = 0x7F default

Register 0x52, Remote 2 Temperature Low Limit = 0x81 default

Register 0x53, Remote 2 Temperature High Limit = 0x7F default

Reading Temperature from the NVT224

It is important to note that temperature can be read from the NVT224 as an 8–bit value (with 1°C resolution) or as a 10–bit value (with 0.25°C resolution). If only 1°C resolution is required, the temperature readings can be read back at any time and in no particular order.

If the 10–bit measurement is required, this involves a two–register read for each measurement. The Extended Resolution Register 2 (0x77) should be read first. This causes all temperature reading registers to be frozen until all temperature reading registers have been read from. This prevents an MSB reading from being updated while its two LSBs are being read and vice versa.

Additional ADC Functions for Temperature Measurement

A number of other functions are available on the NVT224 to offer the system designer increased flexibility.

Turn–Off Averaging

For each temperature measurement read from a value register, 16 readings have actually been made internally, and the results averaged, before being placed into the value register. Sometimes it is necessary to take a very fast measurement. Setting Bit 4 of Configuration Register 2 (0x73) turns averaging off. The default round–robin cycle time takes 146.5 ms.

Table 5. Conversion Time with Averaging Disabled

Channel	Measurement Time (ms)
Voltage Channels	0.7
Remote Temperature 1	7
Remote Temperature 2	7
Local Temperature	1.3

When Bit 7 of Configuration Register 6 (0x10) is set, the default round–robin cycle time increases to 240 ms.

Table 6. Conversion Time with Averaging Enabled

Channel	Measurement Time (ms)
Voltage Channels	11
Remote Temperature 1	39
Remote Temperature 2	39
Local Temperature	12

Single−Channel ADC Conversions

Setting Bit 6 of Configuration Register 2 (0x73) places the NVT224 into single−channel ADC conversion mode. In this mode, the NVT224 can be made to read a single temperature channel only. The appropriate ADC channel is selected by writing to Bits [7:5] of the TACH1 minimum high byte register (0x55).

Table 7. Programming Single−Channel ADC Mode for Temperatures

Register 0x55, Bits [7:5]	Channel Selected
101	Remote 1 Temperature
110	Local Temperature
111	Remote 2 Temperature

Configuration Register 2 (0x73)

Bit 4 = 1, averaging off.

Bit 6 = 1, single−channel convert mode.

TACH1 Minimum High Byte Register (0x55)

Bits [7:5] select the ADC channel for single−channel convert mode.

Overtemperature Events

Overtemperature events on any of the temperature channels can be detected and dealt with automatically in automatic fan speed control mode. Register 0x6A to Register 0x6C are the $\overline{\text{THERM}}$ temperature limit registers. When a temperature exceeds its $\overline{\text{THERM}}$ temperature limit, all PWM outputs run at the maximum PWM duty cycle (Register 0x38, Register 0x39, and Register 0x3A). This effectively runs the fans at the fastest allowed speed.

The fans run at this speed until the temperature drops below $\overline{\text{THERM}}$ minus hysteresis. This can be disabled by setting the boost bit in Configuration Register 3 (0x78), Bit 2. The hysteresis value for the $\overline{\text{THERM}}$ temperature limit is the value programmed into Register 0x6D and Register 0x6E (hysteresis registers). The default hysteresis value is 4°C.

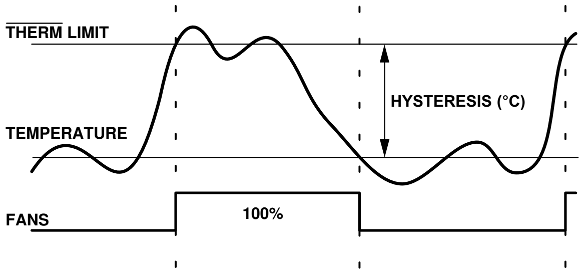


Figure 23. $\overline{\text{THERM}}$ Temperature Limit Operation

$\overline{\text{THERM}}$ can be disabled on specific temperature channels using Bits [7:5] of Configuration Register 5 (0x7C). $\overline{\text{THERM}}$ can also be disabled by:

- In Offset 64 mode, writing −64°C to the appropriate $\overline{\text{THERM}}$ Temperature Limit.
- In twos complement mode, writing −128°C to the appropriate $\overline{\text{THERM}}$ Temperature Limit.

Limits, Status Registers, and Interrupts

Limit Values

Associated with each measurement channel on the NVT224 are high and low limits. These can form the basis of system status monitoring; a status bit can be set for any out−of−limit condition and detected by polling the device. Alternatively, $\overline{\text{SMBALERT}}$ interrupts can be generated to flag out−of−limit conditions to a processor or microcontroller.

8−Bit Limits

The following is a list of 8−bit limits on the NVT224.

Voltage Limit Registers

Register 0x46, V_{CCP} Low Limit = 0x00 default

Register 0x47, V_{CCP} High Limit = 0xFF default

Register 0x48, V_{CC} Low Limit = 0x00 default

Register 0x49, V_{CC} High Limit = 0xFF default

Temperature Limit Registers

Register 0x4E, Remote 1 Temperature Low Limit = 0x81 default

Register 0x4F, Remote 1 Temperature High Limit = 0x7F default

Register 0x6A, Remote 1 $\overline{\text{THERM}}$ Temperature Limit = 0x64 default

Register 0x50, Local Temperature Low Limit = 0x81 default

Register 0x51, Local Temperature High Limit = 0x7F default

Register 0x6B, Local $\overline{\text{THERM}}$ Temperature Limit = 0x64 default

Register 0x52, Remote 2 Temperature Low Limit = 0x81 default

Register 0x53, Remote 2 Temperature High Limit = 0x7F default

Register 0x6C, Remote 2 $\overline{\text{THERM}}$ Temperature Limit = 0x64 default

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

Register 0x7A, $\overline{\text{THERM}}$ Timer Limit = 0x00 default

16−Bit Limits

The fan TACH measurements are 16−bit results. The fan TACH limits are also 16 bits, consisting of a high byte and low byte. Because fans running under speed or stalled are normally the only conditions of interest, only high limits exist for fan TACHs. Because the fan TACH period is actually being measured, exceeding the limit indicates a slow or stalled fan.

Fan Limit Registers

Register 0x54, TACH1 Minimum Low Byte = 0xFF default

Register 0x55, TACH1 Minimum High Byte = 0xFF default

Register 0x56, TACH2 Minimum Low Byte = 0xFF default

Register 0x57, TACH2 Minimum High Byte = 0xFF default

Register 0x58, TACH3 Minimum Low Byte = 0xFF default

Register 0x59, TACH3 Minimum High Byte = 0xFF default

Register 0x5A, TACH4 Minimum Low Byte = 0xFF default

Register 0x5B, TACH4 Minimum High Byte = 0xFF default

Out−of−Limit Comparisons

Once all limits have been programmed, the NVT224 can be enabled for monitoring. The NVT224 measures all voltage and temperature measurements in round−robin format and sets the appropriate status bit for out−of−limit conditions. TACH measurements are not part of this round−robin cycle. Comparisons are done differently, depending on whether the measured value is being compared to a high or low limit.

High Limit > Comparison Performed

Low Limit ≤ Comparison Performed

Voltage and temperature channels use a window comparator for error detecting and, therefore, have high and low limits. Fan speed measurements use only a low limit. This fan limit is needed only in manual fan control mode.

Analog Monitoring Cycle Time

The analog monitoring cycle begins when a 1 is written to the start bit (Bit 0) of Configuration Register 1 (0x40). By default, the NVT224 powers up with this bit set. The ADC measures each analog input in turn and, as each measurement is completed, the result is automatically stored in the appropriate value register. This round−robin monitoring cycle continues unless disabled by writing a 0 to Bit 0 of Configuration Register 1.

As the ADC is normally left to free−run in this manner, the time taken to monitor all the analog inputs is normally not of interest, because the most recently measured value of any input can be read out at any time.

For applications where the monitoring cycle time is important, it can easily be calculated. The total number of channels measured is

- One dedicated supply voltage input (V_{CCP} pin)
- Supply voltage (V_{CC} pin)
- Local temperature
- Two remote temperatures

As mentioned previously, the ADC performs round−robin conversions. The total monitoring cycle time for averaged voltage and temperature monitoring is 146 ms. The total monitoring cycle time for voltage and temperature monitoring with averaging disabled is 19 ms. The NVT224 is a derivative of the ADT7467. As a result, the total conversion time in the NVT224 is the same as the total conversion time of the ADT7467.

Fan TACH measurements are made in parallel and are not synchronized with the analog measurements in any way.

Interrupt Status Registers

The results of limit comparisons are stored in Interrupt Status Register 1 and Interrupt Status Register 2. The status register bit for each channel reflects the status of the last measurement and limit comparison on that channel. If a measurement is within limits, the corresponding status register bit is cleared to 0. If the measurement is out−of−limits, the corresponding status register bit is set to 1.

The state of the various measurement channels can be polled by reading the status registers over the serial bus. In Bit 7 (OOL) of Interrupt Status Register 1 (0x41), 1 means that an out−of−limit event has been flagged in Interrupt Status Register 2. This means that the user needs only to read Interrupt Status Register 2 when this bit is set. Alternatively, Pin 5 or Pin 9 can be configured as an $\overline{\text{SMBALERT}}$ output. This automatically notifies the system supervisor of an out−of−limit condition. Reading the status registers clears the appropriate status bit as long as the error condition that caused the interrupt has cleared.

Status register bits are sticky. Whenever a status bit is set, indicating an out−of−limit condition, it remains set even if the event that caused it has gone away (until read). The only way to clear the status bit is to read the status register after the event has gone away. Interrupt status mask registers (0x74 and 0x75) allow individual interrupt sources to be masked from causing an $\overline{\text{SMBALERT}}$. However, if one of these masked interrupt sources goes out−of−limit, its associated status bit is set in the interrupt status registers.

Interrupt Status Register 1 (0x41)

Bit 7 (OOL) = 1, denotes that a bit in Status Register 2 is set and that Interrupt Status Register 2 should be read.

Bit 6 (R2T) = 1, Remote 2 temperature high or low limit has been exceeded.

Bit 5 (LT) = 1, local temperature high or low limit has been exceeded.

Bit 4 (R1T) = 1, Remote 1 temperature high or low limit has been exceeded.

Bit 2 (V_{CC}) = 1, V_{CC} high or low limit has been exceeded.

Bit 1 (V_{CCP}) = 1, V_{CCP} high or low limit has been exceeded.

Interrupt Status Register 2 (0x42)

Bit 7 (D2) = 1, indicates an open or short on D2+/D2– inputs.

Bit 6 (D1) = 1, indicates an open or short on D1+/D1– inputs.

Bit 5 (F4P) = 1, indicates that Fan 4 has dropped below minimum speed. Alternatively, indicates that the $\overline{\text{THERM}}$ limit has been exceeded, if the $\overline{\text{THERM}}$ function is used.

Bit 4 (FAN3) = 1, indicates that Fan 3 has dropped below minimum speed.

Bit 3 (FAN2) = 1, indicates that Fan 2 has dropped below minimum speed.

Bit 2 (FAN1) = 1, indicates that Fan 1 has dropped below minimum speed.

Bit 1 (OVT) = 1, indicates that a $\overline{\text{THERM}}$ overtemperature limit has been exceeded.

$\overline{\text{SMBALERT}}$ Interrupt Behavior

The NVT224 can be polled for status, or an $\overline{\text{SMBALERT}}$ interrupt can be generated for out−of−limit conditions. Note how the $\overline{\text{SMBALERT}}$ output and status bits behave when writing interrupt handler software.

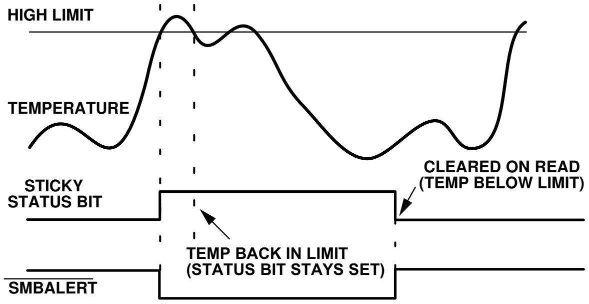

HIGH LIMIT
TEMPERATURE
STICKY STATUS BIT
CLEARED ON READ (TEMP BELOW LIMIT)
TEMP BACK IN LIMIT (STATUS BIT STAYS SET)
SMBALERT

Figure 24. $\overline{\text{SMBALERT}}$ and Status Bit Behavior

Figure 24 shows how the $\overline{\text{SMBALERT}}$ output and sticky status bits behave. Once a limit is exceeded, the corresponding status bit is set to 1. The status bit remains set until the error condition subsides and the status register is read. The status bits are referred to as sticky because they remain set until read by software. This ensures that an out−of−limit event cannot be missed if software is polling the device periodically. Note that the $\overline{\text{SMBALERT}}$ output remains low for the entire duration that a reading is out−of−limit and until the interrupt status register has been read. This has implications for how software handles the interrupt.

Handling $\overline{\text{SMBALERT}}$ Interrupts

To prevent the system from being tied up servicing interrupts, it is recommended to handle the $\overline{\text{SMBALERT}}$ interrupt as follows:

1. Detect the $\overline{\text{SMBALERT}}$ assertion.
2. Enter the interrupt handler.
3. Read the status registers to identify the interrupt source.
4. Mask the interrupt source by setting the appropriate mask bit in the interrupt mask registers (0x74 and 0x75).
5. Take the appropriate action for a given interrupt source.
6. Exit the interrupt handler.
7. Periodically poll the status registers. If the interrupt status bit has cleared, reset the corresponding interrupt mask bit to 0. This causes the $\overline{\text{SMBALERT}}$ output and status bits to behave as shown in Figure 25.

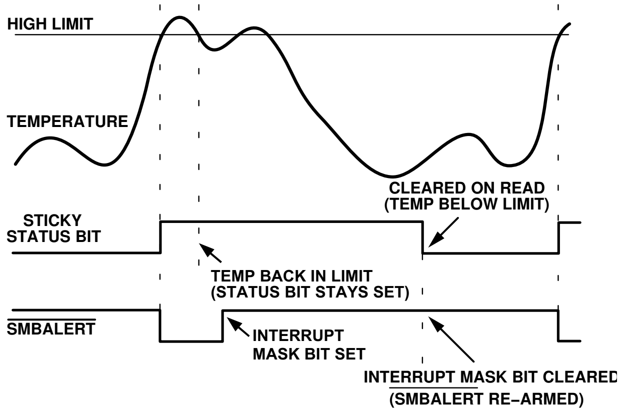

HIGH LIMIT
TEMPERATURE
CLEARED ON READ (TEMP BELOW LIMIT)
STICKY STATUS BIT
TEMP BACK IN LIMIT (STATUS BIT STAYS SET)
SMBALERT
INTERRUPT MASK BIT SET
INTERRUPT MASK BIT CLEARED (SMBALERT RE−ARMED)

Figure 25. How Masking the Interrupt Source Affects $\overline{\text{SMBALERT}}$ Output

Masking Interrupt Sources

Interrupt Mask Register 1 (0x74) and Interrupt Mask Register 2 (0x75) allow individual interrupt sources to be masked out to prevent $\overline{\text{SMBALERT}}$ interrupts. Note that masking an interrupt source prevents only the $\overline{\text{SMBALERT}}$ output from being asserted; the appropriate status bit is set normally.

Interrupt Mask Register 1 (0x74)

Bit 7 (OOL) = 1, masks $\overline{\text{SMBALERT}}$ for any alert condition flagged in Interrupt Status Register 2.

Bit 6 (R2T) = 1, masks $\overline{\text{SMBALERT}}$ for Remote 2 Temperature.

Bit 5 (LT) = 1, masks $\overline{\text{SMBALERT}}$ for Local Temperature.

Bit 4 (R1T) = 1, masks $\overline{\text{SMBALERT}}$ for Remote 1 Temperature.

Bit 2 (V_{CC}) = 1, masks $\overline{\text{SMBALERT}}$ for V_{CC} Channel.

Bit 1 (V_{CCP}) = 1, masks $\overline{\text{SMBALERT}}$ for V_{CCP} Channel.

Interrupt Mask Register 2 (0x75)

Bit 7 (D2) = 1, masks $\overline{\text{SMBALERT}}$ for Diode 2 errors.

Bit 6 (D1) = 1, masks $\overline{\text{SMBALERT}}$ for Diode 1 errors.

Bit 5 (F4P) = 1, masks $\overline{\text{SMBALERT}}$ for Fan 4 failure.

If the TACH4 pin is being used as the $\overline{\text{THERM}}$ input, this bit masks $\overline{\text{SMBALERT}}$ for a $\overline{\text{THERM}}$ event.

Bit 4 (FAN3) = 1, masks $\overline{\text{SMBALERT}}$ for Fan 3.

Bit 3 (FAN2) = 1, masks $\overline{\text{SMBALERT}}$ for Fan 2.

Bit 2 (FAN1) = 1, masks $\overline{\text{SMBALERT}}$ for Fan 1.

Bit 1 (OVT) = 1, masks $\overline{\text{SMBALERT}}$ for overtemperature (exceeding $\overline{\text{THERM}}$ limits).

Enabling the $\overline{\text{SMBALERT}}$ Interrupt Output

The $\overline{\text{SMBALERT}}$ interrupt function is disabled by default. Pin 5 or Pin 9 can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out−of−limit conditions.

Table 8. Configuring Pin 5 as SMBALERT Output

Register	Bit Setting
Configuration Register 3 (0x78)	[0] ALERT Enable = 1

Assigning THERM Functionality to a Pin

Pin 9 on the NVT224 has four possible functions: SMBALERT, THERM, GPIO, and TACH4. The user chooses the required functionality by setting Bit 0 and Bit 1 of Configuration Register 4 (0x7D).

Table 9. Pin 9 Configuration

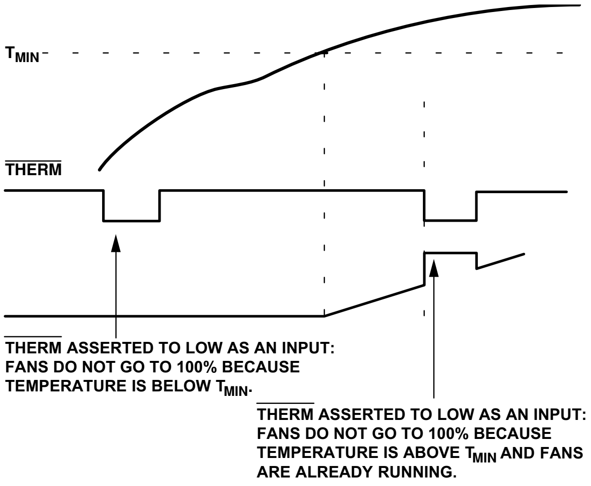
Bit 1	Bit 0	Function
0	0	TACH4
0	1	THERM
1	0	SMBALERT
1	1	GPIO

Once Pin 9 is configured as THERM, it must be enabled (Bit 1, Configuration Register 3 (0x78)).

THERM as an Input

When THERM is configured as an input, the user can time assertions on the THERM pin. This can be useful for connecting to the PROCHOT output of a CPU to gauge system performance.

The user can also set up the NVT224 so that, when the THERM pin is driven low externally, the fans run at 100%. The fans run at 100% for the duration of the time that the THERM pin is pulled low. This is done by setting the BOOST bit (Bit 2) in Configuration Register 3 (0x78) to 1. This works only if the fan is already running, for example, in manual mode when the current duty cycle is above 0x00 or in automatic mode when the temperature is above T_{MIN}. If the temperature is below T_{MIN} or if the duty cycle in manual mode is set to 0x00, pulling the THERM low externally has no effect. See Figure 26 for more information.



T_MIN
THERM
THERM ASSERTED TO LOW AS AN INPUT: FANS DO NOT GO TO 100% BECAUSE TEMPERATURE IS BELOW T_MIN.
THERM ASSERTED TO LOW AS AN INPUT: FANS DO NOT GO TO 100% BECAUSE TEMPERATURE IS ABOVE T_MIN AND FANS ARE ALREADY RUNNING.

Figure 26. Asserting THERM Low as an Input in Automatic Fan Speed Control Mode

THERM Timer

The NVT224 has an internal timer to measure THERM assertion time. For example, the THERM input can be connected to the PROCHOT output of a Pentium 4 CPU to measure system performance. The THERM input can also be connected to the output of a trip point temperature sensor.

The timer is started on the assertion of the NVT224's THERM input and stopped when THERM is un–asserted. The timer counts THERM times cumulatively, that is, the timer resumes counting on the next THERM assertion. The THERM timer continues to accumulate THERM assertion times until the timer is read (it is cleared on read) or until it reaches full scale. If the counter reaches full scale, it stops at that reading until cleared.

The 8–bit THERM timer status register (0x79) is designed so that the Bit 0 is set to 1 on the first THERM assertion. Once the cumulative THERM assertion time has exceeded 45.52 ms, Bit 1 of the THERM timer is set and Bit 0 becomes the LSB of the timer with a resolution of 22.76 ms, see Figure 27.

When using the THERM timer, be aware of the following. After a THERM timer read (Register 0x79), the following happens:

1. The contents of the timer are cleared on read.
2. The F4P bit (Bit 5) of Interrupt Status Register 2 needs to be cleared (assuming that the THERM timer limit has been exceeded).

If the THERM timer is read during a THERM assertion, the following happens:

3. The contents of the timer are cleared.
4. Bit 0 of the THERM timer is set to 1 (because a THERM assertion is occurring).
5. The THERM timer increments from zero.
6. If the THERM timer limit (Register 0x7A) = 0x00, the F4P bit is set.

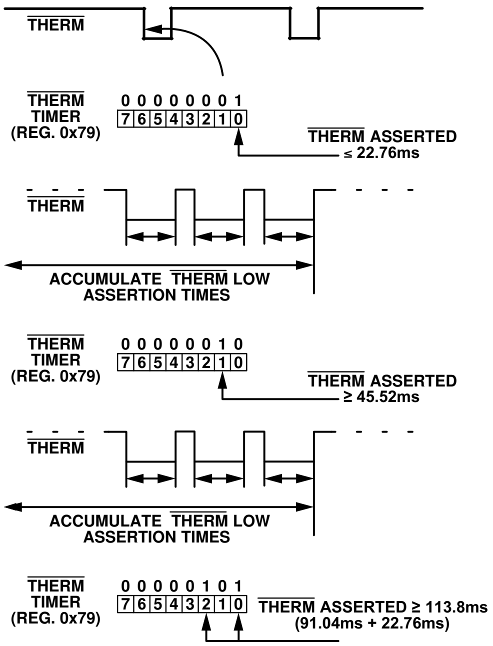


Figure 27. Understanding the THERM Timer

Generating SMBALERT Interrupts from A THERM Timer Events

The NVT224 can generate SMBALERTs when a programmable THERM timer limit has been exceeded. This allows the system designer to ignore brief, infrequent THERM assertions, while capturing longer THERM timer events. Register 0x7A is the THERM timer limit register. This 8−bit register allows a limit from 0 seconds (first THERM assertion) to 5.825 seconds to be set before an SMBALERT is generated. The THERM timer value is compared with the contents of the THERM timer limit register.

If the THERM timer value exceeds the THERM timer limit value, then the F4P bit (Bit 5) of Interrupt Status Register 2 is set, and an SMBALERT is generated. Note that the F4P bit (Bit 5) of Interrupt Mask Register 2 (0x75) masks out SMBALERTs if this bit is set to 1, although the F4P bit of Interrupt Status Register 2 is still set if the THERM timer limit is exceeded.

Figure 28 is a functional block diagram of the THERM timer, limit, and associated circuitry. Writing a value of 0x00 to the THERM timer limit register (0x7A) causes SMBALERT to be generated on the first THERM assertion. A THERM timer limit value of 0x01 generates an SMBALERT once cumulative THERM assertions exceed 45.52 ms.

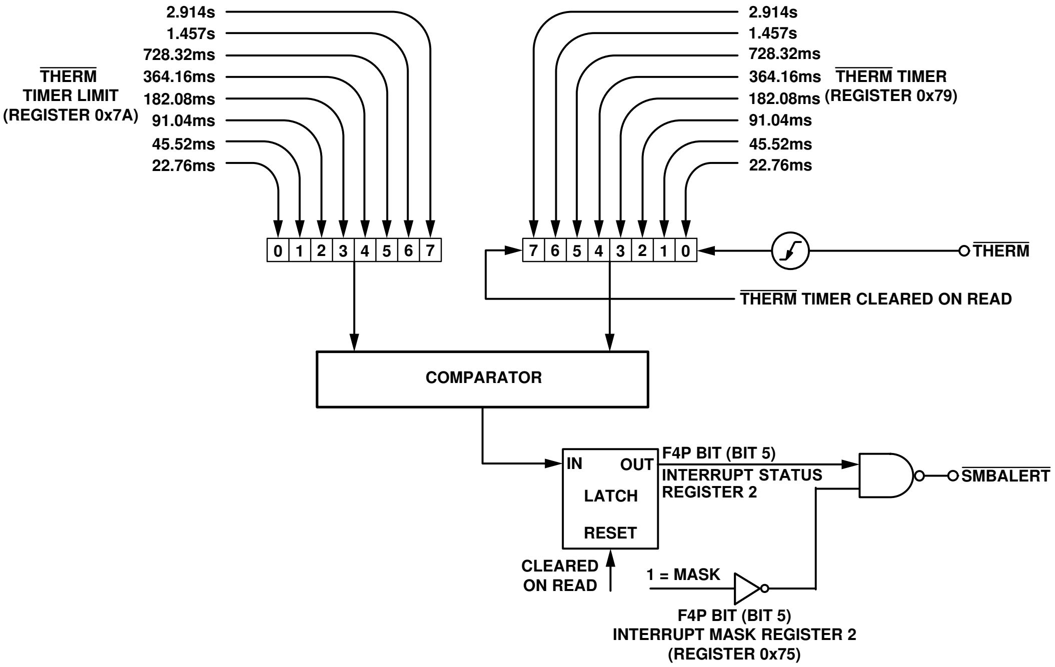


Figure 28. Functional Block Diagram of the NVT224 THERM Monitoring Circuitry

Configuring the THERM Behavior

1. Configure the relevant pin as the THERM timer input. Setting Bit 1 (THERM) of Configuration Register 3 (0x78) enables the THERM timer monitoring functionality. This is disabled on Pin 9 by default.
   Setting Bit 0 and Bit 1 (PIN9FUNC) of Configuration Register 4 (0x7D) enables THERM timer/output functionality on Pin 9 (Bit 1, THERM, of Configuration Register 3, must also be set). Pin 9 can also be used as TACH4.
2. Select the desired fan behavior for THERM timer events.
   Assuming that the fans are running, setting Bit 2 (BOOST bit) of Configuration Register 3 (0x78) causes all fans to run at 100% duty cycle whenever THERM is asserted. This allows fail−safe system cooling. If this bit is 0, the fans run at their current settings and are not affected by THERM events. If the fans are not already running when THERM is asserted, the fans do not run to full speed.
3. Select whether THERM timer events should generate SMBALERT interrupts.
   Bit 5 (F4P) of Interrupt Mask Register 2 (0x75), when set, masks out SMBALERTs when the THERM timer limit value is exceeded. This bit should be cleared if SMBALERTs based on THERM events are required.
4. Select a suitable THERM limit value.
   This value determines whether an SMBALERT is generated on the first THERM assertion or only if a cumulative THERM assertion time limit is exceeded. A value of 0x00 causes an SMBALERT to be generated on the first THERM assertion.
5. Select a THERM monitoring time.
   This value specifies how often OS− or BIOS−level software checks the THERM timer. For example, BIOS could read the THERM timer once an hour to determine the cumulative THERM assertion time. If, for example, the total THERM assertion time is <22.76 ms in Hour 1, >182.08 ms in Hour 2, and >2.914 s in Hour 3, this can indicate that system

performance is degrading significantly because THERM is asserting more frequently on an hourly basis.

Alternatively, OS− or BIOS−level software can time−stamp when the system is powered on. If an SMBALERT is generated due to the THERM timer limit being exceeded, another time−stamp can be taken. The difference in time can be calculated for a fixed THERM timer limit time. For example, if it takes one week for a THERM timer limit of 2.914 seconds to be exceeded and the next time it takes only one hour, this is an indication of a serious degradation in system performance.

Configuring the THERM Pin as an Output

In addition to monitoring THERM as an input, the NVT224 can optionally drive THERM low as an output. In cases where PROCHOT is bidirectional, THERM can be used to throttle the processor by asserting PROCHOT. The user can pre−program system−critical thermal limits. If the temperature exceeds a thermal limit by 0.25°C, THERM asserts low. If the temperature is still above the thermal limit on the next monitoring cycle, THERM stays low. THERM remains asserted low until the temperature is equal to or below the thermal limit. Because the temperature for that channel is measured only once for every monitoring cycle, after THERM asserts, it is guaranteed to remain low for at least one monitoring cycle.

The THERM pin can be configured to assert low if the Remote 1, local, or Remote 2 THERM temperature limit is exceeded by 0.25°C. The THERM temperature limit registers are at Register 0x6A, Register 0x6B, and Register 0x6C. Setting Bit 3 of Register 0x5F, Register 0x60, and Register 0x61 enables the THERM output feature for the Remote 1, local, and Remote 2 temperature channels, respectively. Figure 29 shows how the THERM pin asserts low as an output in the event of a critical over temperature.

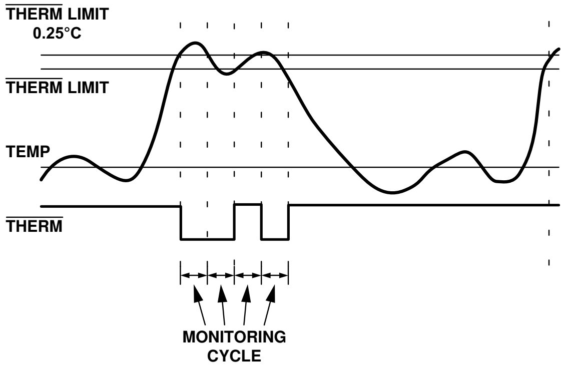


Figure 29. Asserting THERM as an Output, Based on Tripping THERM Limits

An alternative method of disabling THERM is to program the THERM temperature limit to 64°C or less in Offset 64 mode, or 128°C or less in twos complement mode; that is, for THERM temperature limit values less than 64°C or 128°C, respectively, THERM is disabled.

Enabling and Disabling THERM on Individual Channels

THERM can be enabled/disabled for individual or combinations of temperature channels using Bits [7:5] of Configuration Register 5 (0x7C).

THERM Hysteresis

Setting Bit 0 of Configuration Register 7 (0x11) disables THERM hysteresis.

If THERM hysteresis is enabled and THERM is disabled (Bit 2 of Configuration Register 4, 0x7D), the THERM pin does not assert low when a THERM event occurs. If THERM hysteresis is disabled and THERM is disabled (Bit 2 of Configuration Register 4, 0x7D, and assuming the appropriate pin is configured as THERM), the THERM pin asserts low when a THERM event occurs.

If THERM and THERM hysteresis are both enabled, the THERM output asserts as expected.

THERM Operation in Manual Mode

In manual mode, THERM events do not cause fans to go to full speed, unless Bit 3 of Configuration Register 6 (0x10) is set to 1.

Additionally, Bit 3 of Configuration Register 4 (0x7D) can be used to select PWM speed on THERM event (100% or maximum PWM).

Bit 2 in Configuration Register 4 (0x7D) can be set to disable THERM events from affecting the fans.

Fan Drive Using PWM Control

The NVT224 uses pulse−width modulation (PWM) to control fan speed. This relies on varying the duty cycle (or on/off ratio) of a square wave applied to the fan to vary the fan speed. The external circuitry required to drive a fan using PWM control is extremely simple. For 4−wire fans, the PWM drive may need only a pullup resistor. In many cases, the 4−wire fan PWM input has a built−in pullup resistor.

The NVT224 PWM frequency can be set to a selection of low frequencies or a single high PWM frequency. The low frequency options are usually used for 3−wire fans, while the high frequency option is usually used with 4−wire fans.

For 3−wire fans, a single N−channel MOSFET is the only drive device required. The specifications of the MOSFET depend on the maximum current required by the fan being driven. Typical notebook fans draw a nominal 170 mA, so SOT devices can be used where board space is a concern. In desktops, fans can typically draw 250 mA to 300 mA each. If several fans are driven in parallel from a single PWM output or drive larger server fans, the MOSFET must handle the higher current requirements.

The only other stipulation is that the MOSFET should have a gate voltage drive, V_{GS} < 3.3 V, for direct interfacing to the PWM output pin. The MOSFET should also have a low on resistance to ensure that there is not significant voltage drop across the FET, which would reduce the voltage applied across the fan and, therefore, the maximum operating speed of the fan.

Figure 30 shows how to drive a 3−wire fan using PWM control.

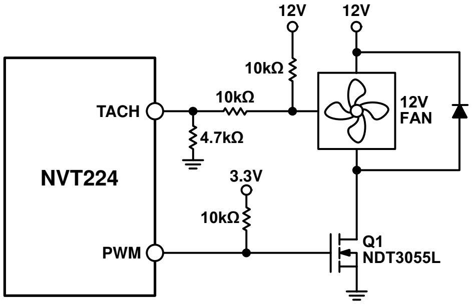


Figure 30. Driving a 3−Wire Fan Using an N−Channel MOSFET

Figure 30 uses a 10 kΩ pullup resistor for the TACH signal. This assumes that the TACH signal is an open−collector from the fan. In all cases, the TACH signal from the fan must be kept below 3.6 V maximum to prevent damaging the NVT224. If in doubt as to whether the fan used has an open−collector or totem pole TACH output, use one of the input signal conditioning circuits shown in the Fan Speed Measurement section.

Figure 31 shows a fan drive circuit using an NPN transistor such as a general−purpose MMBT2222. While these devices are inexpensive, they tend to have much lower current handling capabilities and higher on resistance than MOSFETs. When choosing a transistor, care should be taken to ensure that it meets the fan’s current requirements.

Ensure that the base resistor is chosen so that the transistor is saturated when the fan is powered on.

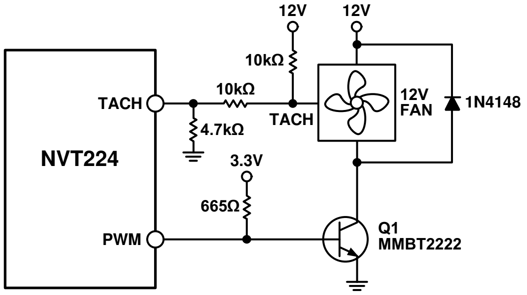


Figure 31. Driving a 3−Wire Fan Using an NPN Transistor

Because 4−wire fans are powered continuously, the fan speed is not switched on or off as with previous PWM driven/powered fans. This enables it to perform better than 3−wire fans, especially for high frequency applications. Figure 32 shows a typical drive circuit for 4−wire fans.

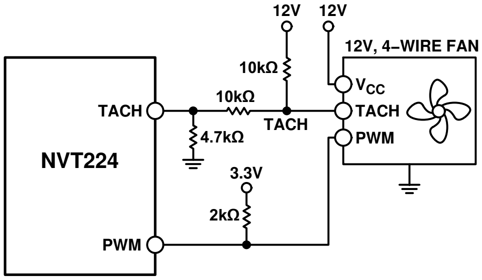


Figure 32. Driving a 4−Wire Fan

Driving Two Fans from PWM3

The NVT224 has four TACH inputs available for fan speed measurement but only three PWM drive outputs. If a fourth fan is used in the system, it should be driven from the PWM3 output in parallel with the third fan. Figure 33 shows how to drive two fans in parallel using low cost NPN transistors. Figure 34 shows the equivalent circuit using a MOSFET.

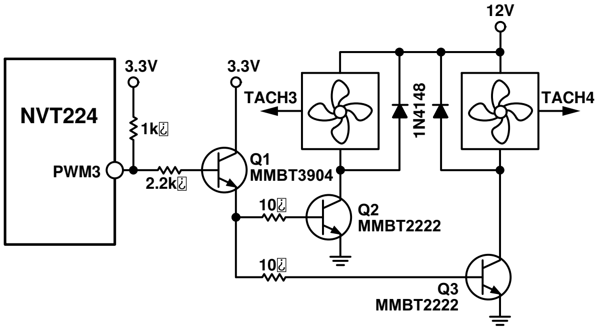


Figure 33. Interfacing Two Fans in Parallel to the PWM3 Output Using Low Cost NPN Transistors

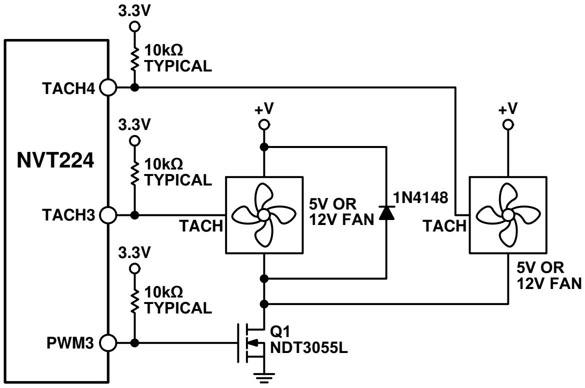


Figure 34. Interfacing Two Fans in Parallel to the PWM3 Output Using a Single N−Channel MOSFET

Because the MOSFET can handle up to 3.5 A, it is simply a matter of connecting another fan directly in parallel with the first. Care should be taken in designing drive circuits with transistors and FETs to ensure that the PWM pins are not required to source current and that they sink less than the 8 mA maximum current specified on the data sheet.

Driving up to Three Fans from PWM3

TACH measurements for fans are synchronized to particular PWM channels; for example, TACH1 is synchronized to PWM1. TACH3 and TACH4 are both synchronized to PWM3, so PWM3 can drive two fans. Alternatively, PWM3 can be programmed to synchronize TACH2, TACH3, and TACH4 to the PWM3 output. This allows PWM3 to drive two or three fans. In this case, the drive circuitry looks the same, as shown in Figure 33 and Figure 34. The SYNC bit in Register 0x62 enables this function.

Synchronization is not required in high frequency mode when used with 4−wire fans.

Bit 4 (SYNC) Enhanced Acoustics Register 1 (0x62)

SYNC = 1, synchronizes TACH2, TACH3, and TACH4 to PWM3.

TACH Inputs

Pin 4, Pin 6, Pin 7, and Pin 9, when configured as TACH inputs, are open−drain TACH inputs intended for fan speed measurement.

Signal conditioning in the NVT224 accommodates the slow rise and fall times typical of fan tachometer outputs. The maximum input signal range is 0 V to 3.6 V. In the event these inputs are supplied from fan outputs that exceed 0 V to 3.6 V, either resistive attenuation of the fan signal or diode clamping must be included to keep inputs within an acceptable range.

Figure 35 to Figure 38 show circuits for most common fan TACH outputs. If the fan TACH output has a resistive pullup to V_{CC}, it can be connected directly to the fan input, as shown in Figure 35.

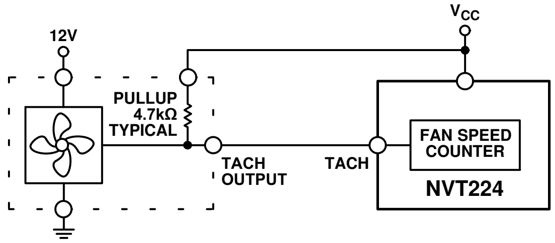
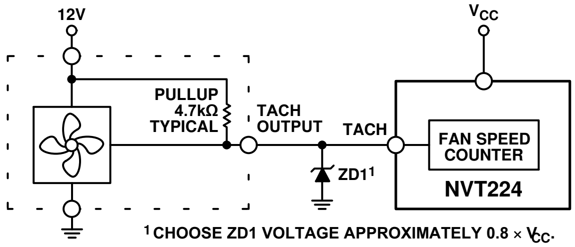


Figure 35. Fan with TACH Pullup to V_{CC}

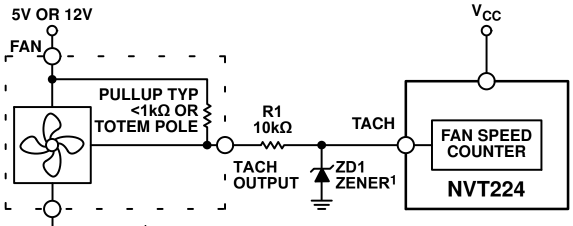
If the fan output has a resistive pullup to 12 V, or other voltage greater than 3.6 V, the fan output can be clamped with a Zener diode, as shown in Figure 36. The Zener diode voltage should be chosen so that it is greater than V_{IH} of the TACH input but less than 3.6 V, allowing for the voltage tolerance of the Zener. A value between 3.0 V and 3.6 V is suitable.



[1] CHOOSE ZD1 VOLTAGE APPROXIMATELY 0.8 × V_{CC}.

Figure 36. Fan with TACH Pullup to Voltage > 3.6 V (example, 12 V) Clamped with Zener Diode

If the fan has a strong pullup (less than 1 kΩ) to 12 V or a totem−pole output, then a series resistor can be added to limit the Zener current, as shown in Figure 37.



[1] CHOOSE ZD1 VOLTAGE APPROXIMATELY 0.8 × V_{CC}.

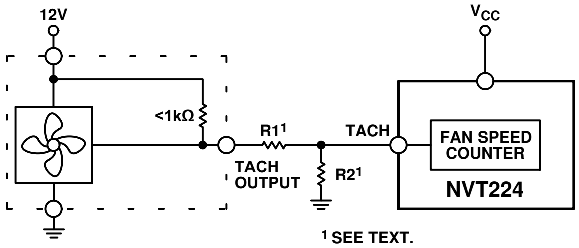
Figure 37. Fan with Strong TACH Pullup to > V_{CC} or Totem−Pole Output, Clamped with Zener and Resistor

Alternatively, a resistive attenuator can be used, as shown in Figure 38. R1 and R2 should be chosen such that:

$$2.0\ V < V_{PULLUP} \times R2/\left(R_{PULLUP} + R1 + R2\right) < 3.6\ V \qquad \text{(eq. 3)}$$

The fan inputs can have an input resistance of 160 kΩ to 5.1 kΩ to ground, which should be taken into account when calculating resistor values.

With a pullup voltage of 12 V and pullup resistor less than 1 kΩ, suitable values for R1 and R2 would be 100 kΩ and 33 kΩ, respectively. This gives a high input voltage of 2.95 V.



[1] SEE TEXT.

Figure 38. Fan with Strong TACH pullup to > V_{CC} or Totem−Pole Output, Attenuated with R1/R2

Fan Speed Measurement

The fan counter does not count the fan TACH output pulses directly because the fan speed could be less than 1000 RPM and it would take several seconds to accumulate a reasonably large and accurate count. Instead, the period of the fan revolution is measured by gating an on−chip 90 kHz oscillator into the input of a 16−bit counter for N periods of the fan TACH output (see Figure 39), so the accumulated count is actually proportional to the fan tachometer period and inversely proportional to the fan speed.

N, the number of pulses counted, is determined by the settings of Register 0x7B (TACH Pulses per Revolution register). This register contains two bits for each fan, allowing one, two (default), three, or four TACH pulses to be counted.

Measuring Fan TACH

When the NVT224 starts up, TACH measurements are locked. In effect, an internal read of the low byte has been made for each TACH input. The net result of this is that all TACH readings are locked until the high byte is read from the corresponding TACH registers. All TACH related interrupts are also ignored until the appropriate high byte is read.

Once the corresponding high byte has been read, TACH measurements are unlocked and interrupts are processed as normal.

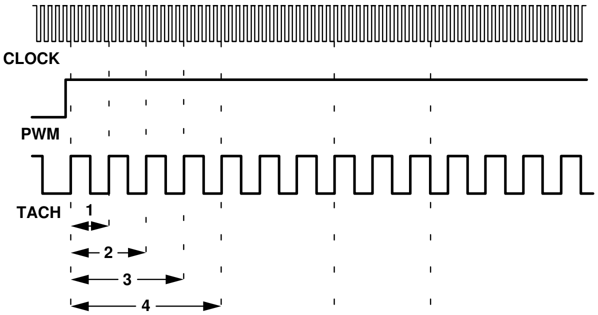


Figure 39. Fan Speed Measurement

Fan Speed Measurement Registers

The fan tachometer readings are 16–bit values consisting of a 2–byte read from the NVT224.

Register 0x28, TACH1 Low Byte = 0x00 default

Register 0x29, TACH1 High Byte = 0x00 default

Register 0x2A, TACH2 Low Byte = 0x00 default

Register 0x2B, TACH2 High Byte = 0x00 default

Register 0x2C, TACH3 Low Byte = 0x00 default

Register 0x2D, TACH3 High Byte = 0x00 default

Register 0x2E, TACH4 Low Byte = 0x00 default

Register 0x2F, TACH4 High Byte = 0x00 default

Reading Fan Speed from the NVT224

The measurement of fan speeds involves a 2–register read for each measurement. The low byte should be read first. This causes the high byte to be frozen until both high and low byte registers have been read, preventing erroneous TACH readings. The fan tachometer reading registers report back the number of 11.11 μs period clocks (90 kHz oscillator) gated to the fan speed counter, from the rising edge of the first fan TACH pulse to the rising edge of the third fan TACH pulse (assuming two pulses per revolution are being counted). Because the device is essentially measuring the fan TACH period, the higher the count value, the slower the fan is actually running. A 16–bit fan tachometer reading of 0xFFFF indicates that the fan either has stalled or is running very slowly (<100 RPM).

High Limit > Comparison Performed

Because the actual fan TACH period is being measured, falling below a fan TACH limit by 1 sets the appropriate status bit and can be used to generate an $\overline{\text{SMBALERT}}$.

Fan TACH Limit Registers

The fan TACH limit registers are 16–bit values consisting of two bytes.

Register 0x54, TACH1 Minimum Low Byte = 0xFF default

Register 0x55, TACH1 Minimum High Byte = 0xFF default

Register 0x56, TACH2 Minimum Low Byte = 0xFF default

Register 0x57, TACH2 Minimum High Byte = 0xFF default

Register 0x58, TACH3 Minimum Low Byte = 0xFF default

Register 0x59, TACH3 Minimum High Byte = 0xFF default

Register 0x5A, TACH4 Minimum Low Byte = 0xFF default

Register 0x5B, TACH4 Minimum High Byte = 0xFF default

Fan Speed Measurement Rate

The fan TACH readings are normally updated once every second. The FAST bit (Bit 3) of Configuration Register 3 (0x78), when set, updates the fan TACH readings every 250 ms.

If any of the fans are not being driven by a PWM channel but are powered directly from 5.0 V or 12 V, their associated dc bit in Configuration Register 3 should be set. This allows TACH readings to be taken on a continuous basis for fans connected directly to a dc source. For optimal results, the associated dc bit should always be set when using 4–wire fans.

Calculating Fan Speed

Assuming a fan with a two pulses per revolution (and two pulses per revolution being measured), fan speed is calculated by the following:

Fan Speed (RPM) = (90,000 x 60)/Fan TACH Reading where Fan TACH Reading is the 16–bit fan tachometer reading.

Example

TACH1 High Byte (Register 0x29) = 0x17

TACH1 Low Byte (Register 0x28) = 0xFF

What is Fan 1 speed in RPM?

Fan 1 TACH Reading = 0x17FF = 6143 (decimal)
RPM = (f x 60)/Fan 1 TACH Reading
RPM = (90,000 x 60)/6143
Fan Speed = 879 RPM

Fan Pulses per Revolution

Different fan models can output either 1, 2, 3, or 4 TACH pulses per revolution. Once the number of fan TACH pulses has been determined, it can be programmed into the TACH Pulses per Revolution register (Register 0x7B) for each fan.

Alternatively, this register can be used to determine the number or pulses per revolution output by a given fan. By plotting fan speed measurements at 100% speed with different pulses per revolution setting, the smoothest graph with the lowest ripple determines the correct pulses per revolution value.

TACH Pulses per Revolution Register

Bits [1:0] Fan 1 default = 2 pulses per revolution.

Bits [3:2] Fan 2 default = 2 pulses per revolution.

Bits [5:4] Fan 3 default = 2 pulses per revolution.

Bits [7:6] Fan 4 default = 2 pulses per revolution.
00 = 1 pulse per revolution
01 = 2 pulses per revolution
10 = 3 pulses per revolution
11 = 4 pulses per revolution

Fan Spin−Up

The NVT224 has a unique fan spin−up function. It spins the fan at 100% PWM duty cycle until two TACH pulses are detected on the TACH input. Once two TACH pulses have been detected, the PWM duty cycle goes to the expected running value, for example, 33%. The advantage is that fans have different spin−up characteristics and take different times to overcome inertia. The NVT224 runs the fans just fast enough to overcome inertia and is quieter on spin−up than fans programmed to spin up for a given spin−up time.

Fan Startup Timeout

To prevent the generation of false interrupts as a fan spins up (because it is below running speed), the NVT224 includes a fan startup timeout function. During this time, the NVT224 looks for two TACH pulses. If two TACH pulses are not detected, an interrupt is generated. Using Configuration Register 1 (0x40), Bit 5 (FSPDIS), this functionality can be changed (see the Disabling Fan Startup Timeout section).

PWM1, PWM2, PWM3 Configuration Registers (0x5C, 0x5D, and 0x5E)

Bits [2:0] SPIN, startup timeout for PWM1 = 0x5C, PWM2 = 0x5D, and PWM3 = 0x5E.
000 = No startup timeout
001 = 100 ms
010 = 250 ms default
011 = 400 ms
100 = 667 ms
101 = 1 sec
110 = 2 sec
111 = 4 sec

Disabling Fan Startup Timeout

Although fan startup makes fan spin−ups much quieter than fixed−time spin−ups, the option exists to use fixed spin−up times. Setting Bit 5 (FSPDIS) to 1 in Configuration Register 1 (0x40) disables the spin−up for two TACH pulses. Instead, the fan spins up for the fixed time as selected in Register 0x5C to Register 0x5E.

PWM Logic State

The PWM outputs can be programmed high for 100% duty cycle (non−inverted) or low for 100% duty cycle (inverted).

PWM1 Configuration Register (0x5C)

Bit 4 INV
0 = Logic high for 100% PWM duty cycle.
1 = Logic low for 100% PWM duty cycle.

PWM2 Configuration Register (0x5D)

Bit 4 INV
0 = Logic high for 100% PWM duty cycle.
1 = Logic low for 100% PWM duty cycle.

PWM3 Configuration Register (0x5E)

Bit 4 INV
0 = Logic high for 100% PWM duty cycle.
1 = Logic low for 100% PWM duty cycle.

Low Frequency Mode PWM Drive Frequency

The PWM drive frequency can be adjusted for the application. Register 0x5F to Register 0x61 configure the PWM frequency for PWM1 to PWM3, respectively. In high frequency mode, the PWM drive frequency is always 22.5 kHz.

High Frequency Mode PWM Drive

Setting Bit 3 of Register 0x5F, Register 0x60, and Register 0x61 enables high frequency mode for Fan 1, Fan 2, and Fan 3, respectively.

PWM Frequency Registers (0x5F to 0x61)

Bits [2:0] FREQ
000 = 11.0 Hz
001 = 14.7 Hz
010 = 22.1 Hz
011 = 29.4 Hz
100 = 35.3 Hz default
101 = 44.1 Hz
110 = 58.8 Hz
111 = 88.2 Hz

Fan Speed Control

The NVT224 controls fan speed using automatic mode and manual mode as follows:

- In automatic fan speed control mode, fan speed is automatically varied with temperature and without CPU intervention, once initial parameters are set up. The advantage of this is, if the system hangs, the user is guaranteed that the system is protected from overheating. For more information about how to program the automatic fan speed control loop, see the Programming the Automatic Fan Speed Control Loop section.
- In manual fan speed control mode, the NVT224 allows the duty cycle of any PWM output to be manually adjusted. This can be useful if the user wants to change fan speed at the software level or adjust PWM duty cycle output for test purposes. Bits [7:5] of Register 0x5C to Register 0x5E (PWM configuration) control the behavior of each PWM output.