

## Programming the Automatic Fan Speed Control Loop

By Mary Burke

### AUTOMATIC FAN SPEED CONTROL

The ADT7460/ADT7463 have a local temperature sensor and two remote temperature channels that may be connected to an on-chip diode-connected transistor on a CPU. These three temperature channels may be used as the basis for automatic fan speed control to drive fans using pulsewidth modulation (PWM). In general, the greater the number of fans in a system, the better the cooling, but this is to the detriment of system acoustics. Automatic fan speed control reduces acoustic noise by optimizing fan speed according to measured temperature. Reducing fan speed can also decrease system current consumption. The automatic fan speed control mode is very flexible owing to the number of programmable parameters, including  $T_{MIN}$  and  $T_{RANGE}$ , as discussed in detail later. The  $T_{MIN}$  and  $T_{RANGE}$  values for a

temperature channel and thus for a given fan are critical since these define the thermal characteristics of the system. The thermal validation of the system is one of the most important steps of the design process, so these values should be carefully selected.

### AIM OF THIS SECTION

The aim of this application note is not only to provide the system designer with an understanding of the automatic fan control loop, but to also provide step-by-step guidance as to how to most effectively evaluate and select the critical system parameters. To optimize the system characteristics, the designer needs to give some forethought to how the system will be configured, i.e., the number of fans, where they are located, and what temperatures are being measured in the particular

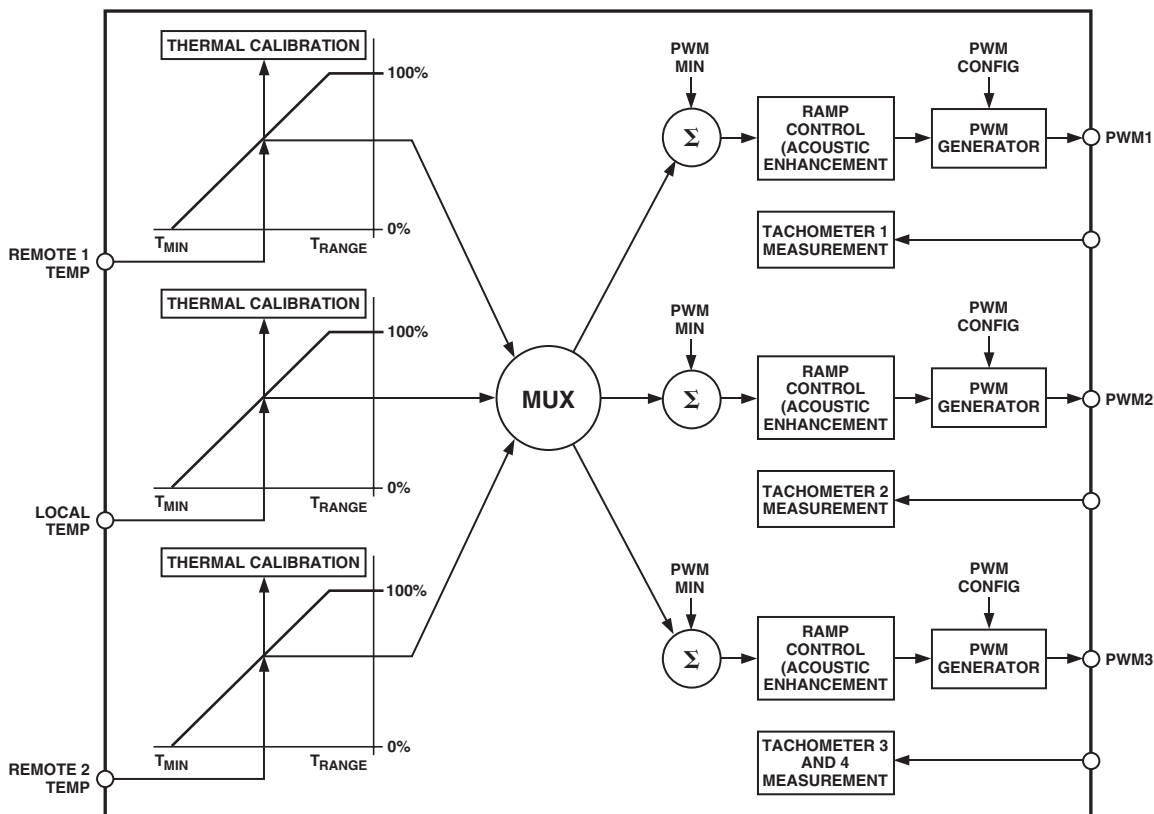


Figure 1. Automatic Fan Control Block Diagram

system. The mechanical or thermal engineer who is tasked with the actual system evaluation should also be involved at the beginning of the process.

## AUTOMATIC FAN CONTROL OVERVIEW

Figure 1 gives a top-level overview of the automatic fan control circuitry on the ADT7460/ADT7463. From a systems level perspective, up to three system temperatures can be monitored and used to control three PWM outputs. The three PWM outputs can be used to control up to four fans. The ADT7460/ADT7463 allow the speed of four fans to be monitored. Each temperature channel has a thermal calibration block. This allows the designer to individually configure the thermal characteristics of each temperature channel. For example, one may decide to run the CPU fan when CPU temperature increases above 60°C, and a chassis fan when the local temperature increases above 45°C. Note that at this stage, you have not assigned these thermal calibration settings to a particular fan drive (PWM) channel. The right side of the Block Diagram (Figure 1) shows controls that are fan-specific. The designer has individual control over parameters such as minimum PWM duty cycle, fan speed failure thresholds, and even ramp control of the PWM outputs. This ultimately allows graceful fan speed changes that are less perceptible to the system user.

## STEP 1: DETERMINING THE HARDWARE CONFIGURATION

During system design, the motherboard sensing and control capabilities should not be an afterthought, but

addressed early in the design stages. Decisions about how these capabilities are used should involve the system thermal/mechanical engineer. Ask the following questions:

1. What ADT7460/ADT7463 functionality will be used?

- PWM2 or SMBALERT?
- 2.5 V voltage monitoring or SMBALERT?
- 2.5 V voltage monitoring or processor power monitoring?
- TACH4 fan speed measurement or over-temperature THERM function?
- 5 V voltage monitoring or overtemperature THERM function?
- 12 V voltage monitoring or VID5 input?

The ADT7460/ADT7463 offers multifunctional pins that can be reconfigured to suit different system requirements and physical layouts. These multifunction pins are software programmable. Various pinout options are discussed in a separate application note.

2. How many fans will be supported in system, three or four? This will influence the choice of whether to use the TACH4 pin or to reconfigure it for the THERM function.

3. Is the CPU fan to be controlled using the ADT7460/ADT7463 or will it run at full speed 100% of the time?

If run at 100%, it will free up a PWM output, but the system will be louder.

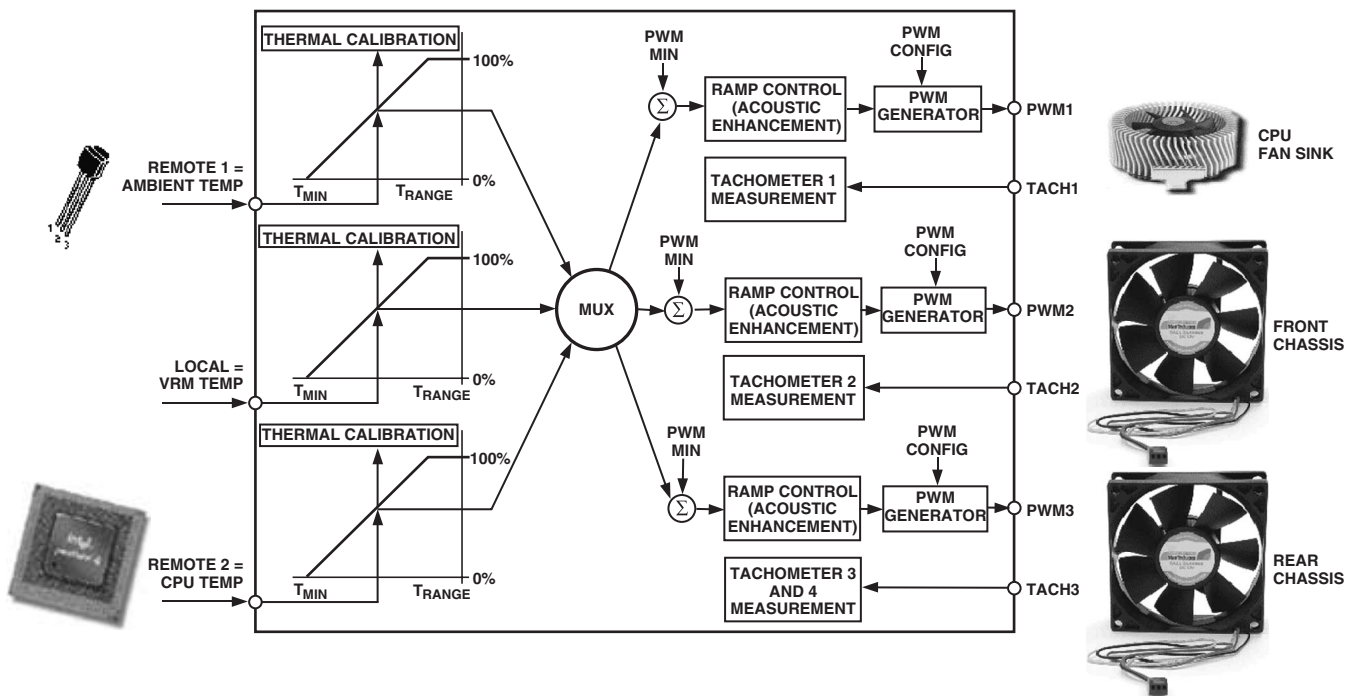


Figure 2. Hardware Configuration Example

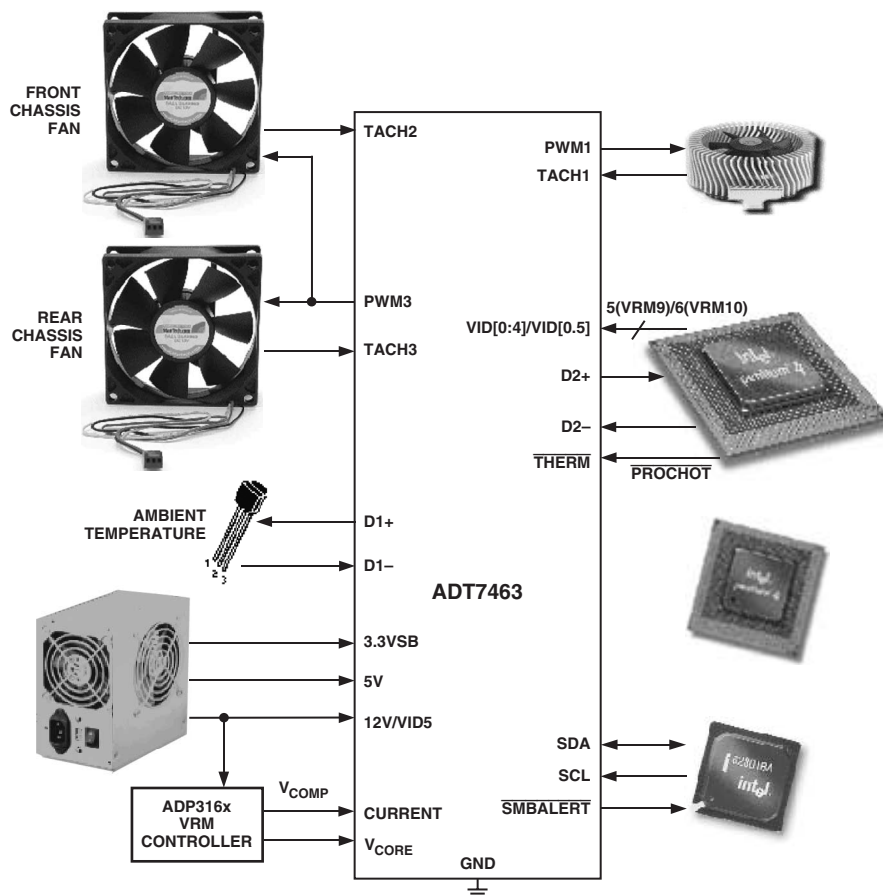


Figure 3. Recommended Implementation 1

4. Where will the ADT7460/ADT7463 be physically located in the system?

This influences the assignment of the temperature measurement channels to particular system thermal zones. For example, locating the ADT7460/ADT7463 close to the VRM controller circuitry allows the VRM temperature to be monitored using the local temperature channel.

#### RECOMMENDED IMPLEMENTATION 1

Configuring the ADT7460/ADT7463 as in Figure 3 provides the systems designer with the following features:

1. Six VID Inputs (VID0 to VID5) for VRM10 Support.
2. Two PWM Outputs for Fan Control of up to Three Fans. (The front and rear chassis fans are connected in parallel.)
3. Three TACH Fan Speed Measurement Inputs.
4.  $V_{CC}$  Measured Internally through Pin 4.
5. CPU Core Voltage Measurement ( $V_{CORE}$ ).
6. 2.5 V Measurement Input Used to Monitor CPU Current (connected to  $V_{COMP}$  output of ADP316x VRM controller). This is used to determine CPU power consumption.
7. 5 V Measurement Input.
8. VRM temperature uses local temperature sensor.
9. CPU Temperature Measured Using Remote 1 Temperature Channel.
10. Ambient Temperature Measured through Remote 2 Temperature Channel.
11. If not using VID5, this pin can be reconfigured as the 12 V monitoring input.
12. Bidirectional  $\overline{THERM}$  Pin. Allows monitoring of  $\overline{PROCHOT}$  output from Intel® P4 processor, for example, or can be used as an overtemperature  $\overline{THERM}$  output.
13.  $\overline{SMBALERT}$  System Interrupt Output.

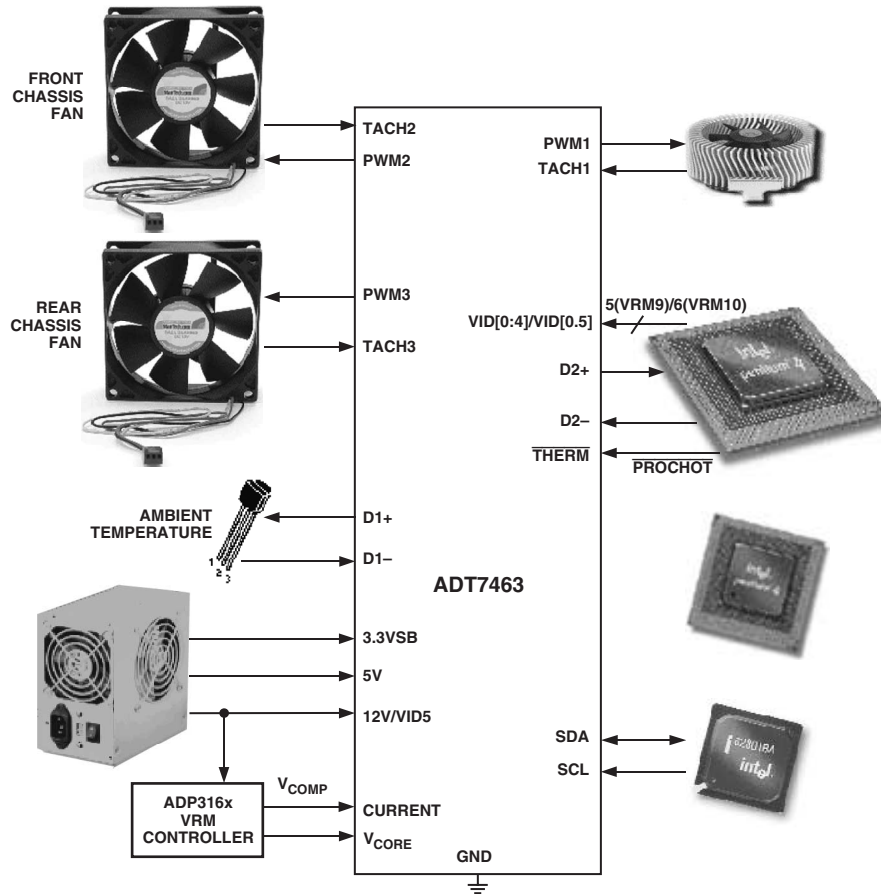


Figure 4. Recommended Implementation 2

### RECOMMENDED IMPLEMENTATION 2

Configuring the ADT7460/ADT7463 as in Figure 4 provides the systems designer with the following features:

1. Six VID Inputs (VID0 to VID5) for VRM10 Support.
2. Three PWM Outputs for Fan Control of up to Three Fans. (All three fans can be individually controlled.)
3. Three TACH Fan Speed Measurement Inputs.
4.  $V_{CC}$  Measured Internally through Pin 4.
5. CPU Core Voltage Measurement ( $V_{CORE}$ ).
6. 2.5 V Measurement Input Used to Monitor CPU Current (connected to  $V_{COMP}$  output of ADP316x VRM Controller). This is used to determine CPU power consumption.
7. 5 V Measurement Input.
8. VRM Temperature Uses Local Temperature Sensor.
9. CPU Temperature Measured Using Remote 1 Temperature Channel.
10. Ambient Temperature Measured through Remote 2 Temperature Channel.
11. If not using VID5, this pin can be reconfigured as the 12 V monitoring input.
12. BIDIRECTIONAL THERM Pin. Allows monitoring of PROCHOT output from Intel P4 processor, for example, or can be used as an overtemperature THERM output.

**STEP 2: CONFIGURING THE MUX – WHICH TEMPERATURE CONTROLS WHICH FAN?**

After the system hardware configuration is determined, the fans can be assigned to particular temperature channels. Not only can fans be assigned to individual channels, but the behavior of fans is also configurable. For example, fans can be run under automatic fan control, can run manually (under software control), or can run at the fastest speed calculated by multiple temperature channels. The MUX is the bridge between temperature measurement channels and the three PWM outputs.

Bits <7:5> (BHVR bits) of registers 0x5C, 0x5D, and 0x5E (PWM configuration registers) control the behavior of the fans connected to the PWM1, PWM2, and PWM3 outputs. The values selected for these bits determine how the MUX connects a temperature measurement channel to a PWM output.

**AUTOMATIC FAN CONTROL MUX OPTIONS**

**<7:5> (BHVR) REGISTERS 0x5C, 0x5D, 0x5E**

- 000 = Remote 1 Temp controls PWMx
- 001 = Local Temp controls PWMx
- 010 = Remote 2 Temp controls PWMx
- 101 = Fastest Speed calculated by Local and Remote 2 Temp controls PWMx
- 110 = Fastest Speed calculated by all three temperature channels controls PWMx

The "Fastest Speed Calculated" options pertain to the ability to control one PWM output based on multiple temperature channels. The thermal characteristics of the three temperature zones can be set to drive a single fan. An example would be if the fan turns on when Remote 1 temperature exceeds 60°C or if the local temperature exceeds 45°C.

**OTHER MUX OPTIONS**

**<7:5> (BHVR) REGISTERS 0x5C, 0x5D, 0x5E**

- 011 = PWMx runs full speed (default)
- 100 = PWMx disabled
- 111 = Manual Mode. PWMx is run under software control. In this mode, PWM duty cycle registers (registers 0x30 to 0x32) are writable and control the PWM outputs.

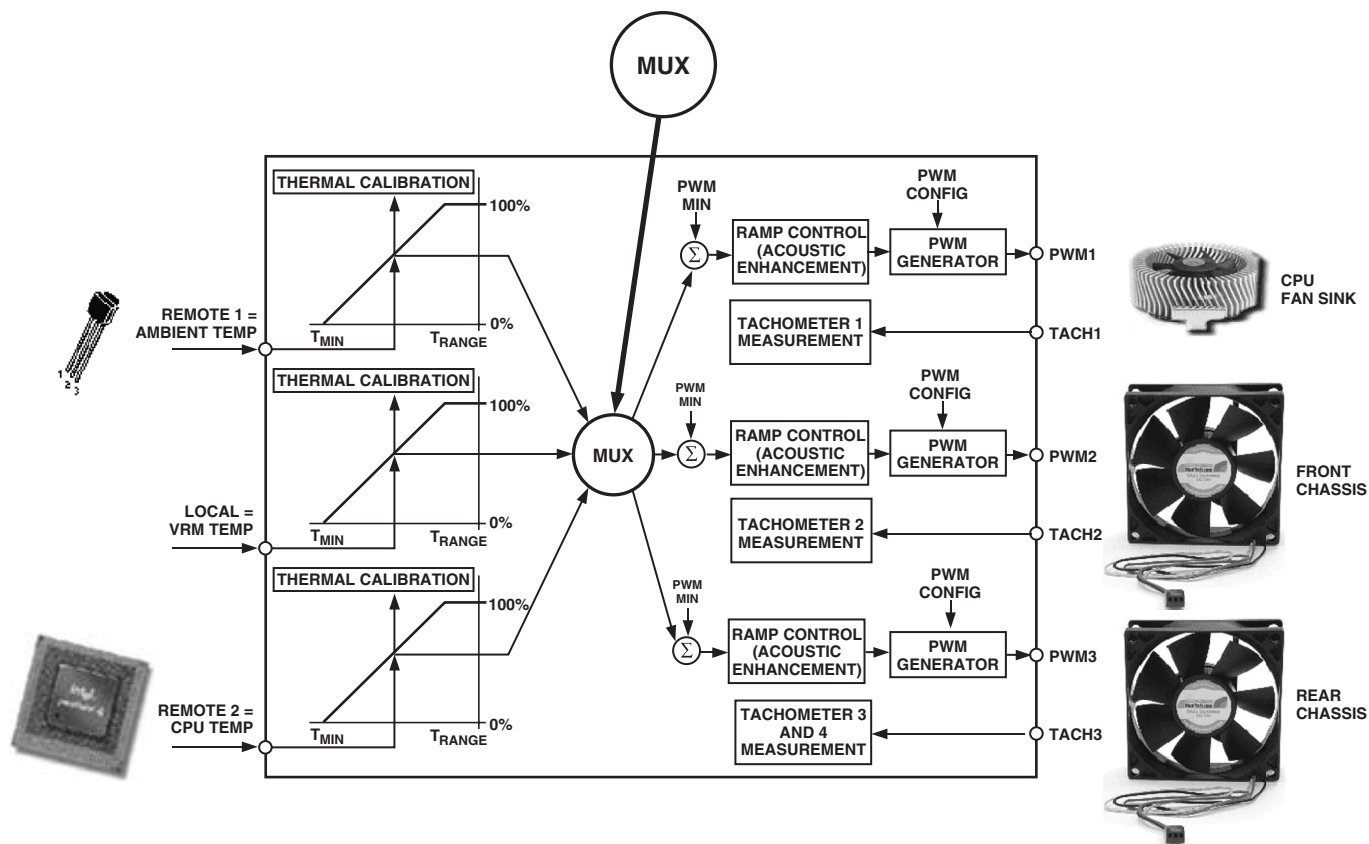


Figure 5. Assigning Temperature Channels to Fan Channels

## MUX CONFIGURATION EXAMPLE

This is an example of how to configure the MUX in a system using the ADT7460/ADT7463 to control three fans. The CPU fan sink is controlled by PWM1, the front chassis fan is controlled by PWM 2, and the rear chassis fan is controlled by PWM3. The MUX is configured for the following fan control behavior:

PWM1 (CPU fan sink) is controlled by the fastest speed calculated by the Local (VRM Temp) and Remote 2 (processor) temperature. In this case, the CPU fan sink is also being used to cool the VRM.

PWM2 (front chassis fan) is controlled by the Remote 1 temperature (ambient).

PWM3 (rear chassis fan) is controlled by the Remote 1 temperature (ambient).

## EXAMPLE MUX SETTINGS

### <7:5> (BHVR) PWM1 CONFIGURATION REG 0x5C

101 = Fastest speed calculated by Local and Remote 2 Temp controls PWM1.

### <7:5> (BHVR) PWM2 CONFIGURATION REG 0x5D

000 = Remote 1 Temp controls PWM2.

### <7:5> (BHVR) PWM3 CONFIGURATION REG 0x5E

000 = Remote 1 Temp controls PWM3.

These settings configure the MUX, as shown in Figure 6.

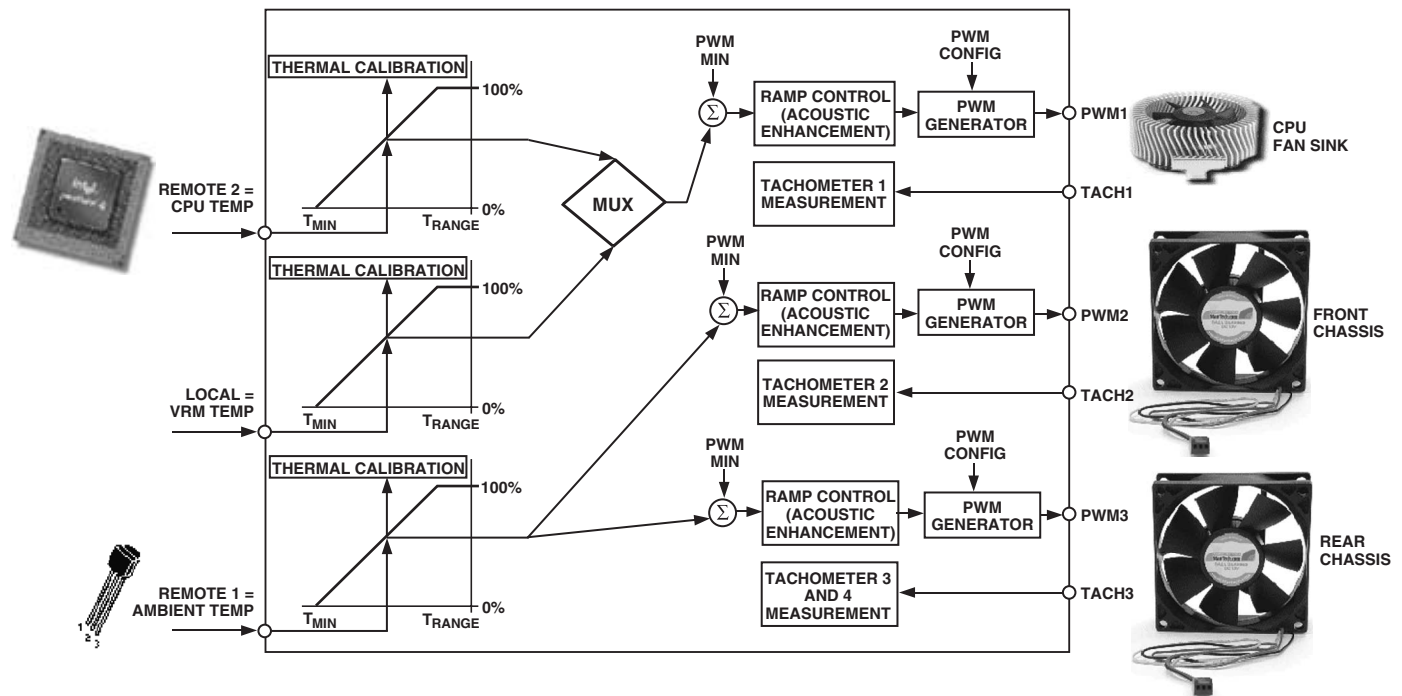


Figure 6. MUX Configuration Example



**STEP 3: DETERMINING  $T_{MIN}$  SETTING FOR EACH THERMAL CALIBRATION CHANNEL**

$T_{MIN}$  is the temperature at which the fans will start to turn on under automatic fan control. The speed at which the fan runs at  $T_{MIN}$  is programmed later. The  $T_{MIN}$  values chosen will be temperature channel specific, e.g., 25°C for ambient channel, 30°C for VRM temperature, and 40°C for processor temperature.

$T_{MIN}$  is an 8-bit two's complement value that can be programmed in 1°C increments. There is a  $T_{MIN}$  register associated with each temperature measurement channel: Remote 1, Local, and Remote 2 Temp. Once the  $T_{MIN}$  value is exceeded, the fan turns on and runs at minimum PWM duty cycle. The fan will turn off once temperature has dropped below  $T_{MIN} - T_{HYST}$  (detailed later).

To overcome fan inertia, the fan is spun up until two valid tach rising edges are counted. See the Fan Startup Timeout section of the ADT7460/ADT7463 data sheet for more details. In some cases, primarily for psycho-acoustic reasons, it is desirable that the fan never switches off below  $T_{MIN}$ . Bits <7:5> of enhance acoustics

Register 1 (Reg. 0x62), when set, keeps the fans running at PWM minimum duty cycle if the temperature should fall below  $T_{MIN}$ .

**$T_{MIN}$  REGISTERS**

- Reg. 0x67 Remote 1 Temp  $T_{MIN} = 0x5A$  (90°C default)
- Reg. 0x68 Local Temp  $T_{MIN} = 0x5A$  (90°C default)
- Reg. 0x69 Remote 2 Temp  $T_{MIN} = 0x5A$  (90°C default)

**ENHANCE ACOUSTICS REG 1 (REG. 0x62)**

- Bit 7 (MIN3) = 0**, PWM3 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .
- Bit 7 (MIN3) = 1**, PWM3 runs at PWM3 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .
- Bit 6 (MIN2) = 0**, PWM2 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .
- Bit 6 (MIN2) = 1**, PWM2 runs at PWM2 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .
- Bit 5 (MIN1) = 0**, PWM1 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .
- Bit 5 (MIN1) = 1**, PWM1 runs at PWM1 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .

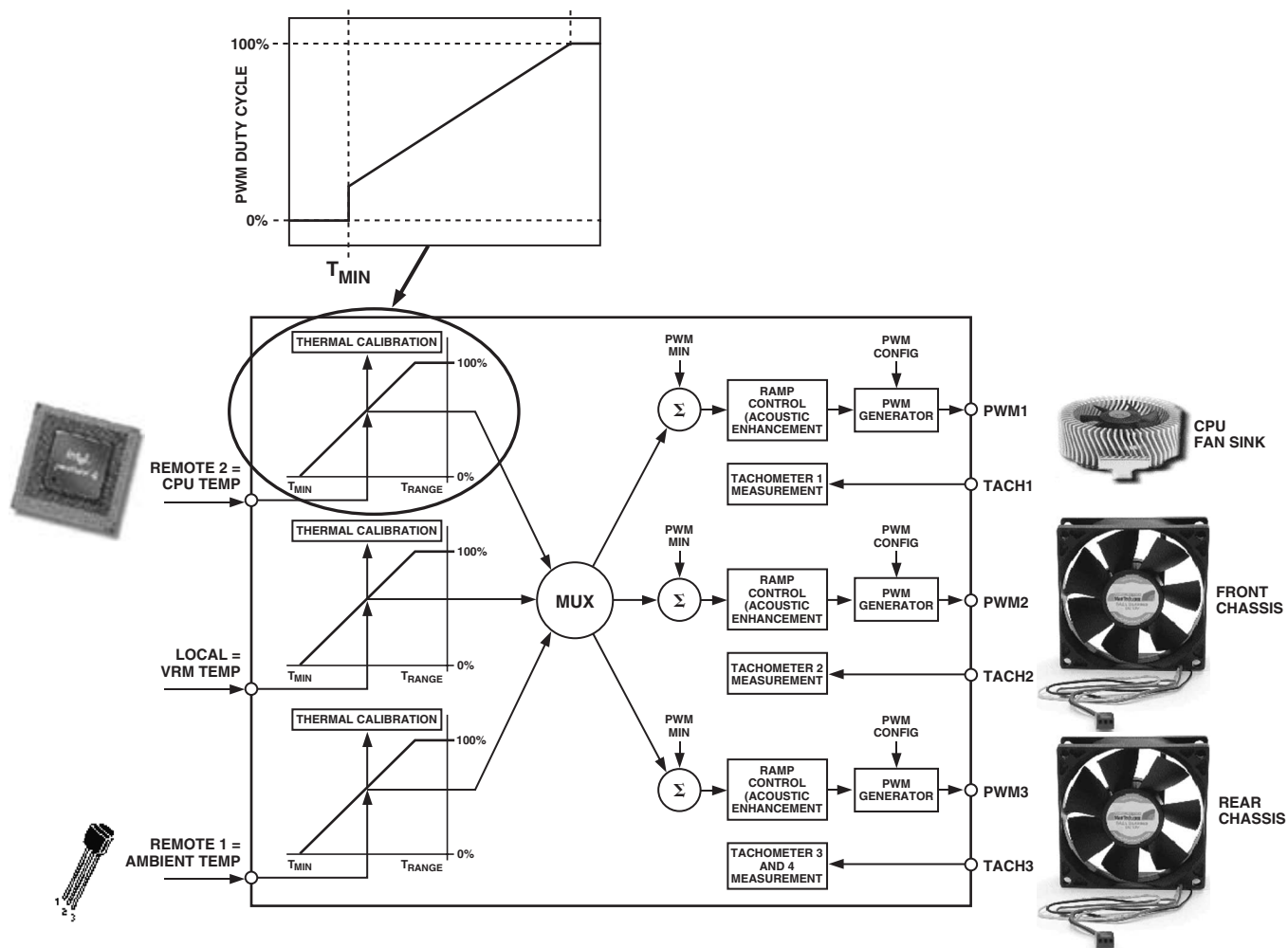


Figure 7. Understanding the  $T_{MIN}$  Parameter

## STEP 4: DETERMINING PWM<sub>MIN</sub> FOR EACH PWM (FAN) OUTPUT

PWM<sub>MIN</sub> is the minimum PWM duty cycle at which each fan in the system will run. It is also the “start” speed for each fan under automatic fan control once the temperature rises above T<sub>MIN</sub>. For maximum system acoustic benefit, PWM<sub>MIN</sub> should be as low as possible. Starting the fans at higher speeds than necessary will merely make the system louder than necessary. Depending on the fan used, the PWM<sub>MIN</sub> setting should be in the 20% to 33% duty cycle range. This value can be found through fan validation.

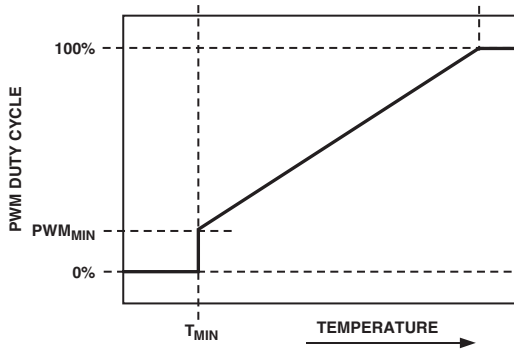


Figure 8. PWM<sub>MIN</sub> Determines Minimum PWM Duty Cycle

It is important to note that more than one PWM output can be controlled from a single temperature measurement channel. For example, Remote 1 Temp can control PWM1 and PWM2 outputs. If two different fans are used on PWM and PWM2, then the fan characteristics can be set up differently. As a result, Fan 1 driven by PWM1 can have a different PWM<sub>MIN</sub> value than that of Fan 2 connected to PWM2. Figure 9 illustrates this as PWM1<sub>MIN</sub> (front fan) is turned on at a minimum duty cycle of 20%, whereas PWM2<sub>MIN</sub> (rear fan) turns on at a minimum of 40% duty cycle. Note, however, that both fans turn on at exactly the same temperature, defined by T<sub>MIN</sub>.

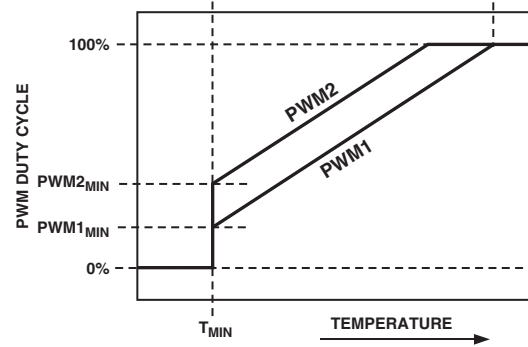


Figure 9. Operating Two Different Fans from a Single Temperature Channel

## PROGRAMMING THE PWM<sub>MIN</sub> REGISTERS

The PWM<sub>MIN</sub> registers are 8-bit registers that allow the minimum PWM duty cycle for each output to be configured anywhere from 0% to 100%. This allows minimum PWM duty cycle to be set in steps of 0.39%.

The value to be programmed into the PWM<sub>MIN</sub> register is given by:

$$\text{Value (decimal)} = \text{PWM}_{\text{MIN}}/0.39$$

Example 1: For a minimum PWM duty cycle of 50%,

$$\begin{aligned} \text{Value (decimal)} &= 50/0.39 = 128 \text{ decimal} \\ \text{Value} &= 128 \text{ decimal or } 80 \text{ hex} \end{aligned}$$

Example 2: For a minimum PWM duty cycle of 33%,

$$\begin{aligned} \text{Value (decimal)} &= 33/0.39 = 85 \text{ decimal} \\ \text{Value} &= 85 \text{ decimal or } 54 \text{ hex} \end{aligned}$$

## PWM<sub>MIN</sub> REGISTERS

- Reg. 0x64 PWM1 Min Duty Cycle = 0x80 (50% default)
- Reg. 0x65 PWM2 Min Duty Cycle = 0x80 (50% default)
- Reg. 0x66 PWM3 Min Duty Cycle = 0x80 (50% default)

## FAN SPEED AND PWM DUTY CYCLE

It should be noted that PWM duty cycle does not directly correlate to fan speed in RPM. Running a fan at 33% PWM duty cycle does not equate to running the fan at 33% speed. Driving a fan at 33% PWM duty cycle actually runs the fan at closer to 50% of its full speed. This is because fan speed in %RPM relates to the square root of PWM duty cycle. Given a PWM square wave as the drive signal, fan speed in RPM equates to:

$$\% \text{ fan speed} = \sqrt{\text{PWM duty cycle}} \times 10$$



**STEP 5: DETERMINING  $T_{RANGE}$  FOR EACH TEMPERATURE CHANNEL**

$T_{RANGE}$  is the range of temperature over which automatic fan control occurs once the programmed  $T_{MIN}$  temperature has been exceeded.  $T_{RANGE}$  is actually a temperature slope and not an arbitrary value, i.e., a  $T_{RANGE}$  of 40°C only holds true for  $PWM_{MIN} = 33\%$ . If  $PWM_{MIN}$  is increased or decreased, the effective  $T_{RANGE}$  is changed, as described later.

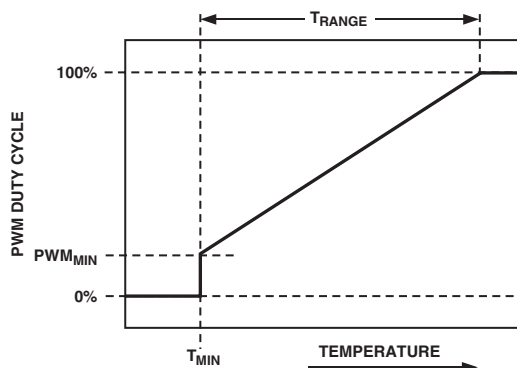


Figure 10.  $T_{RANGE}$  Parameter Affects Cooling Slope

The  $T_{RANGE}$  or fan control slope is determined by the following procedure:

1. Determine the maximum operating temperature for that channel, e.g., 70°C.
2. Determine experimentally the fan speed (PWM duty cycle value) that will not exceed the temperature at the worst-case operating points, e.g., 70°C is reached when the fans are running at 50% PWM duty cycle.
3. Determine the slope of the required control loop to meet these requirements.
4. Use best fit approximation to determine the most suitable  $T_{RANGE}$  value. ADT7460/ADT7463 evaluation software is available to calculate the best fit value. Ask your local Analog Devices representative for more details.

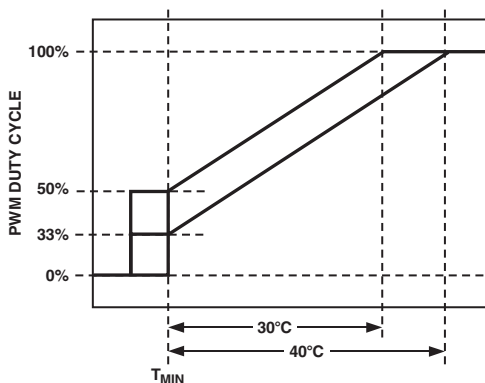


Figure 11. Adjusting  $PWM_{MIN}$  Affects  $T_{RANGE}$

$T_{RANGE}$  is implemented as a slope, which means as  $PWM_{MIN}$  is changed,  $T_{RANGE}$  changes but the actual slope remains the same. The higher the  $PWM_{MIN}$  value, the smaller the effective  $T_{RANGE}$  will be, i.e., the fan will reach full speed (100%) at a lower temperature.

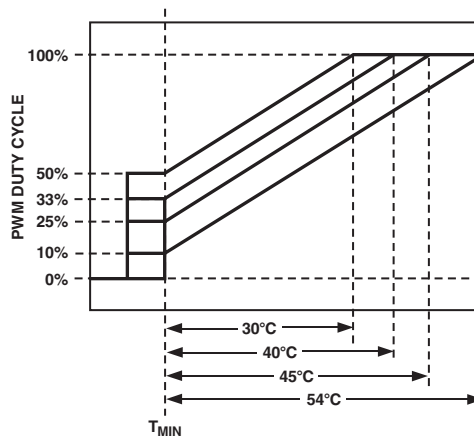


Figure 12. Increasing  $PWM_{MIN}$  Changes Effective  $T_{RANGE}$

For a given  $T_{RANGE}$  value, the temperature at which the fan will run at full speed for different  $PWM_{MIN}$  values can easily be calculated:

$$T_{MAX} = T_{MIN} + ((Max D. C. - Min D. C.) \times T_{RANGE}/170)$$

where

$T_{MAX}$  = Temperature at which the fan runs full speed

$T_{MIN}$  = Temperature at which the fan will turn on

Max D. C. = Maximum duty cycle (100%) = 255 decimal

Min D. C. =  $PWM_{MIN}$

$T_{RANGE}$  = PWM duty cycle versus temperature slope

**Example:** Calculate  $T_{MAX}$ , given  $T_{MIN} = 30^\circ C$ ,  $T_{RANGE} = 40^\circ C$ , and  $PWM_{MIN} = 10\%$  duty cycle = 26 decimal

$$T_{MAX} = T_{MIN} + (Max D. C. - Min D. C.) \times T_{RANGE}/170$$

$$T_{MAX} = 30^\circ C + (100\% - 10\%) \times 40^\circ C/170$$

$$T_{MAX} = 30^\circ C + (255 - 26) \times 40^\circ C/170$$

$$T_{MAX} = 84^\circ C \text{ (effective } T_{RANGE} = 54^\circ C)$$

**Example:** Calculate  $T_{MAX}$ , given  $T_{MIN} = 30^\circ C$ ,  $T_{RANGE} = 40^\circ C$ , and  $PWM_{MIN} = 25\%$  duty cycle = 64 decimal

$$T_{MAX} = T_{MIN} + (Max D. C. - Min D. C.) \times T_{RANGE}/170$$

$$T_{MAX} = 30^\circ C + (100\% - 25\%) \times 40^\circ C/170$$

$$T_{MAX} = 30^\circ C + (255 - 64) \times 40^\circ C/170$$

$$T_{MAX} = 75^\circ C \text{ (effective } T_{RANGE} = 45^\circ C)$$

**Example:** Calculate  $T_{MAX}$ , given  $T_{MIN} = 30^\circ C$ ,  $T_{RANGE} = 40^\circ C$ , and  $PWM_{MIN} = 33\%$  duty cycle = 85 decimal

$$T_{MAX} = T_{MIN} + (Max D. C. - Min D. C.) \times T_{RANGE}/170$$

$$T_{MAX} = 30^\circ C + (100\% - 33\%) \times 40^\circ C/170$$

$$T_{MAX} = 30^\circ C + (255 - 85) \times 40^\circ C/170$$

$$T_{MAX} = 70^\circ C \text{ (effective } T_{RANGE} = 40^\circ C)$$

# AN-613

**Example:** Calculate  $T_{MAX}$ , given  $T_{MIN} = 30^{\circ}C$ ,  $T_{RANGE} = 40^{\circ}C$ , and  $PWM_{MIN} = 50\%$  duty cycle = 128 decimal

$$T_{MAX} = T_{MIN} + (Max D. C. - Min D. C.) \times T_{RANGE}/170$$

$$T_{MAX} = 30^{\circ}C + (100\% - 50\%) \times 40^{\circ}C/170$$

$$T_{MAX} = 30^{\circ}C + (255 - 128) \times 40^{\circ}C/170$$

$$T_{MAX} = 60^{\circ}C \text{ (effective } T_{RANGE} = 30^{\circ}C)$$

## SELECTING A $T_{RANGE}$ SLOPE

The  $T_{RANGE}$  value can be selected for each temperature channel: Remote 1, Local, and Remote 2 Temp. Bits <7:4> ( $T_{RANGE}$ ) of registers 0x5F to 0x61 define the  $T_{RANGE}$  value for each temperature channel.

Table I. Selecting a  $T_{RANGE}$  Value

Bits <7:4>*	$T_{RANGE}$
0000	2°C
0001	2.5°C
0010	3.33°C
0011	4°C
0100	5°C
0101	6.67°C
0110	8°C
0111	10°C
1000	13.33°C
1001	16°C
1010	20°C
1011	26.67°C
1100	32°C (default)
1101	40°C
1110	53.33°C
1111	80°C

\* Register 0x5F configures Remote 1  $T_{RANGE}$   
 Register 0x60 configures Local  $T_{RANGE}$   
 Register 0x61 configures Remote 2  $T_{RANGE}$

## SUMMARY OF $T_{RANGE}$ FUNCTION

When using the automatic fan control function, the temperature at which the fan reaches full speed can be calculated by

$$T_{MAX} = T_{MIN} + T_{RANGE} \quad (1)$$

Equation 1 only holds true when  $PWM_{MIN} = 33\%$  PWM duty cycle.

Increasing or decreasing  $PWM_{MIN}$  will change the effective  $T_{RANGE}$ , although the fan control will still follow the same PWM duty cycle to temperature slope. The effective  $T_{RANGE}$  for different  $PWM_{MIN}$  values can be calculated using Equation 2.

$$T_{MAX} = T_{MIN} + (Max D. C. - Min D. C.) \times T_{RANGE}/170 \quad (2)$$

where:

$$(Max D. C. - Min D. C.) \times T_{RANGE}/170 = \text{effective } T_{RANGE} \text{ value.}$$

Remember that %PWM duty cycle does not correspond to %RPM. %RPM relates to the square root of the PWM duty cycle.

$$\% \text{ fan speed} = \sqrt{\text{PWM duty cycle}} \times 10$$

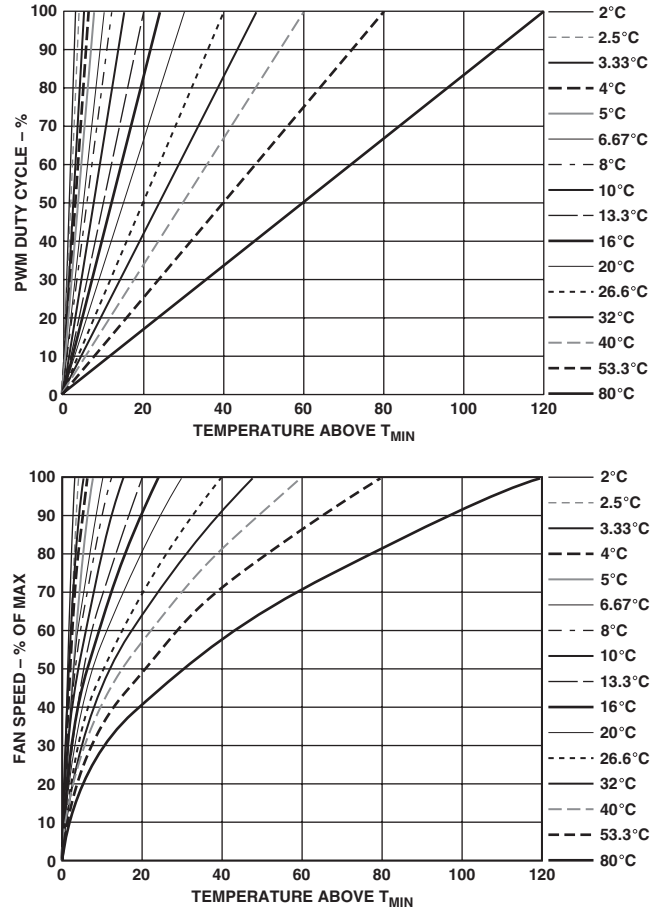


Figure 13.  $T_{RANGE}$  vs. Actual Fan Speed Profile

Figure 13 shows PWM duty cycle versus temperature for each  $T_{RANGE}$  setting. The lower graph shows how each  $T_{RANGE}$  setting affects fan speed versus temperature. As can be seen from the graph, the effect on fan speed is nonlinear. The graphs in Figure 13 assume that the fan starts from 0% PWM duty cycle. Clearly, the minimum PWM duty cycle,  $PWM_{MIN}$ , needs to be factored in to see how the loop actually performs in the system. Figure 14 shows how  $T_{RANGE}$  is affected when the  $PWM_{MIN}$  value is set to 20%. It can be seen that the fan will actually run at about 45% fan speed when the temperature exceeds  $T_{MIN}$ .

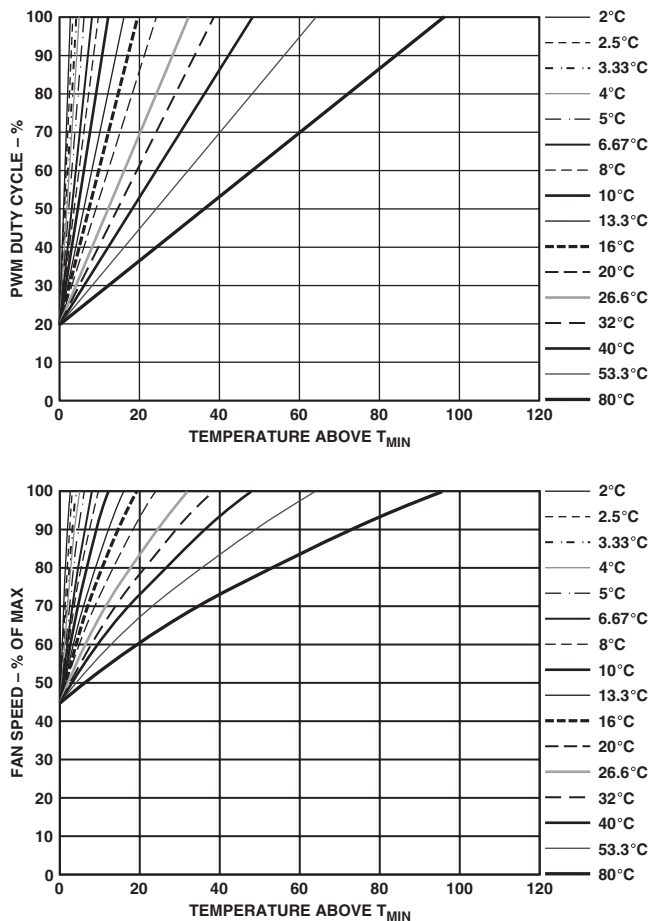


Figure 14.  $T_{RANGE}$ , % Fan Speed Slopes with  $PWM_{MIN} = 20\%$

**EXAMPLE: DETERMINING  $T_{RANGE}$  FOR EACH TEMPERATURE CHANNEL**

The following example is used to show how  $T_{MIN}$ ,  $T_{RANGE}$  settings might be applied to three different thermal zones. In this example, the following  $T_{RANGE}$  values apply:

- $T_{RANGE} = 80^{\circ}C$  for Ambient Temperature
- $T_{RANGE} = 53.3^{\circ}C$  for CPU Temperature
- $T_{RANGE} = 40^{\circ}C$  for VRM Temperature

This example uses the MUX configuration described in Step 2, with the ADT7460/ADT7463 connected as shown in Figure 6. Both CPU temperature and VRM temperature drive the CPU fan connected to PWM1. Ambient temperature drives the front chassis fan and rear chassis fan connected to PWM2 and PWM3.

The front chassis fan is configured to run at  $PWM_{MIN} = 20\%$ . The rear chassis fan is configured to run at  $PWM_{MIN} = 30\%$ .

The CPU fan is configured to run at  $PWM_{MIN} = 10\%$ .

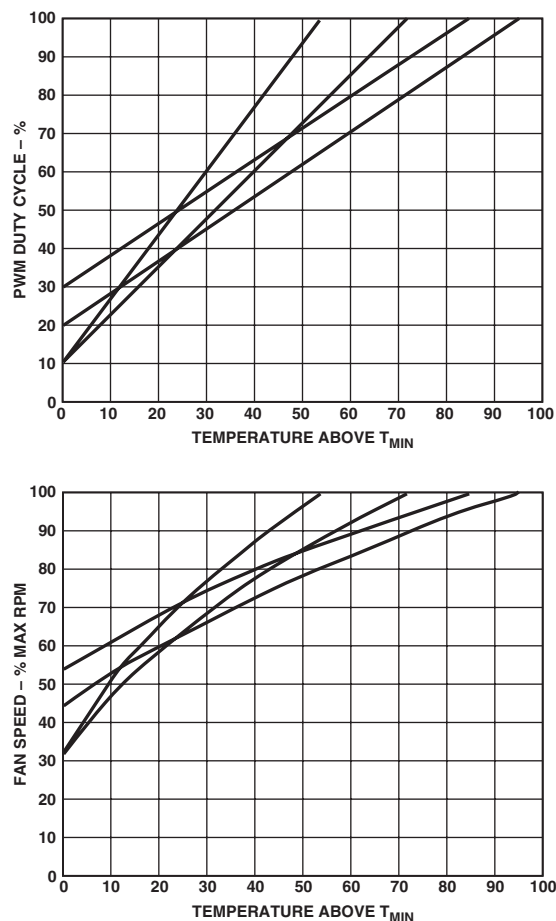


Figure 15.  $T_{RANGE}$ , % Fan Speed Slopes for VRM, Ambient, and CPU Temperature Channels

## STEP 6: DETERMINING $T_{THERM}$ FOR EACH TEMPERATURE CHANNEL

$T_{THERM}$  is the absolute maximum temperature allowed on a temperature channel. Above this temperature, a component such as the CPU or VRM may be operating beyond its safe operating limit. When the temperature measured exceeds  $T_{THERM}$ , all fans are driven at 100% PWM duty cycle (full speed) to provide critical system cooling. The fans remain running 100% until the temperature drops below  $T_{THERM} - \text{hysteresis}$ . The hysteresis value is the number programmed into hysteresis registers 0x6D and 0x6E. The default hysteresis value is 4°C.

The  $T_{THERM}$  limit should be considered the maximum worst-case operating temperature of the system. Since exceeding any  $T_{THERM}$  limit runs all fans at 100%, it has very negative acoustic effects. Ultimately, this limit should be set up as a failsafe, and one should ensure that it is not exceeded under normal system operating conditions.

Note that the  $T_{THERM}$  limits are nonmaskable and affect the fan speed no matter what automatic fan control settings are configured. This allows some flexibility since a  $T_{RANGE}$  value can be selected based on its slope, while a "hard limit," e.g., 70°C, can be programmed as  $T_{MAX}$  (the temperature at which the fan reaches full speed) by setting  $T_{THERM}$  to 70°C.

### THERM REGISTERS

Reg. 0x6A Remote 1  $\overline{THERM}$  limit = 0x64 (100°C default)

Reg. 0x6B Local Temp  $\overline{THERM}$  limit = 0x64 (100°C default)

Reg. 0x6C Remote 2  $\overline{THERM}$  limit = 0x64 (100°C default)

### HYSTERESIS REGISTERS

Reg. 0x6D Remote 1, Local Hysteresis Register

<7:4> = Remote 1 Temp Hysteresis (4°C default)

<3:0> = Local Temp Hysteresis (4°C default)

Reg. 0x6E Remote 2 Temp Hysteresis Register

<7:4> = Remote 2 Temp Hysteresis (4°C default)

Since each hysteresis setting is four bits, hysteresis values are programmable from 1°C to 15°C. It is not recommended that hysteresis values ever be programmed to 0°C, as this actually disables hysteresis. In effect, this would cause the fans to cycle between normal speed and 100% speed, creating unsettling acoustic noise.

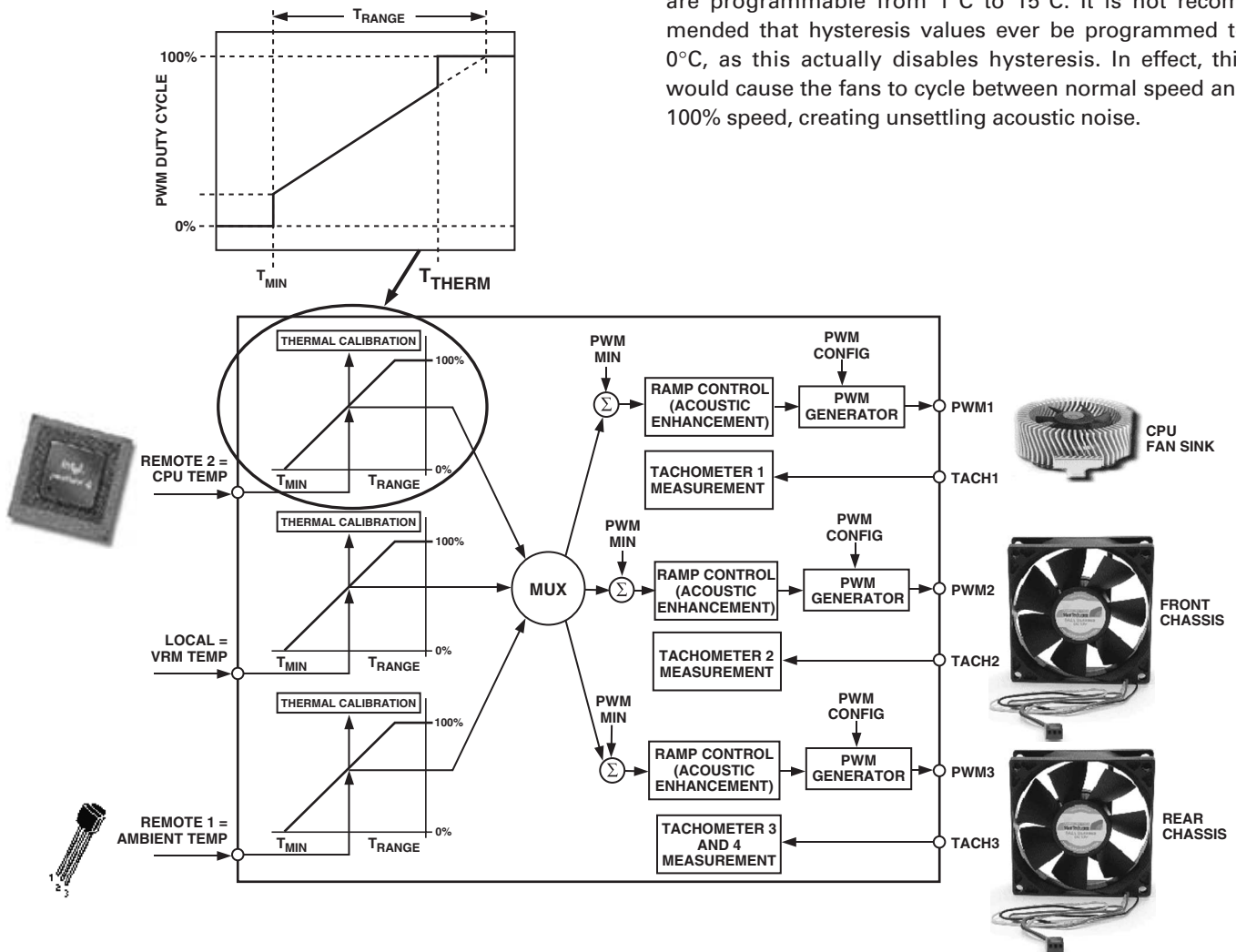


Figure 16. Understanding How  $T_{THERM}$  Relates to Automatic Fan Control

**STEP 7: DETERMINING  $T_{HYST}$  FOR EACH TEMPERATURE CHANNEL**

$T_{HYST}$  is the amount of extra cooling a fan provides after the temperature measured has dropped back below  $T_{MIN}$  before the fan turns off. The premise for temperature hysteresis ( $T_{HYST}$ ) is that without it, the fan would merely “chatter,” or cycle on and off regularly, whenever temperature is hovering at about the  $T_{MIN}$  setting.

The  $T_{HYST}$  value chosen will determine the amount of time needed for the system to cool down or heat up as the fan is turning on and off. Values of hysteresis are programmable in the range 1°C to 15°C. Larger values of  $T_{HYST}$  prevent the fans from chattering on and off as previously described. The  $T_{HYST}$  default value is set at 4°C.

Note that the  $T_{HYST}$  setting applies not only to the temperature hysteresis for fan turn on/off, but the same setting is used for the  $T_{THERM}$  hysteresis value described in Step 6. So programming registers 0x6D and 0x6E sets the hysteresis for both fan on/off and the  $\overline{THERM}$  function.

**HYSTERESIS REGISTERS**

**Reg. 0x6D Remote 1, Local Hysteresis Register**  
 <7:4> = Remote 1 Temp Hysteresis (4°C default)  
 <3:0> = Local Temp Hysteresis (4°C default)

**Reg. 0x6E Remote 2 Temp Hysteresis Register**  
 <7:4> = Remote 2 Temp Hysteresis (4°C default)

Note that in some applications, it is required that the fans not turn off below  $T_{MIN}$  but remain running at  $PWM_{MIN}$ . Bits <7:5> of Enhance Acoustics Register 1 (Reg. 0x62) allow the fans to be turned off, or to be kept spinning below  $T_{MIN}$ . If the fans are always on, the  $T_{HYST}$  value has no effect on the fan when the temperature drops below  $T_{MIN}$ .

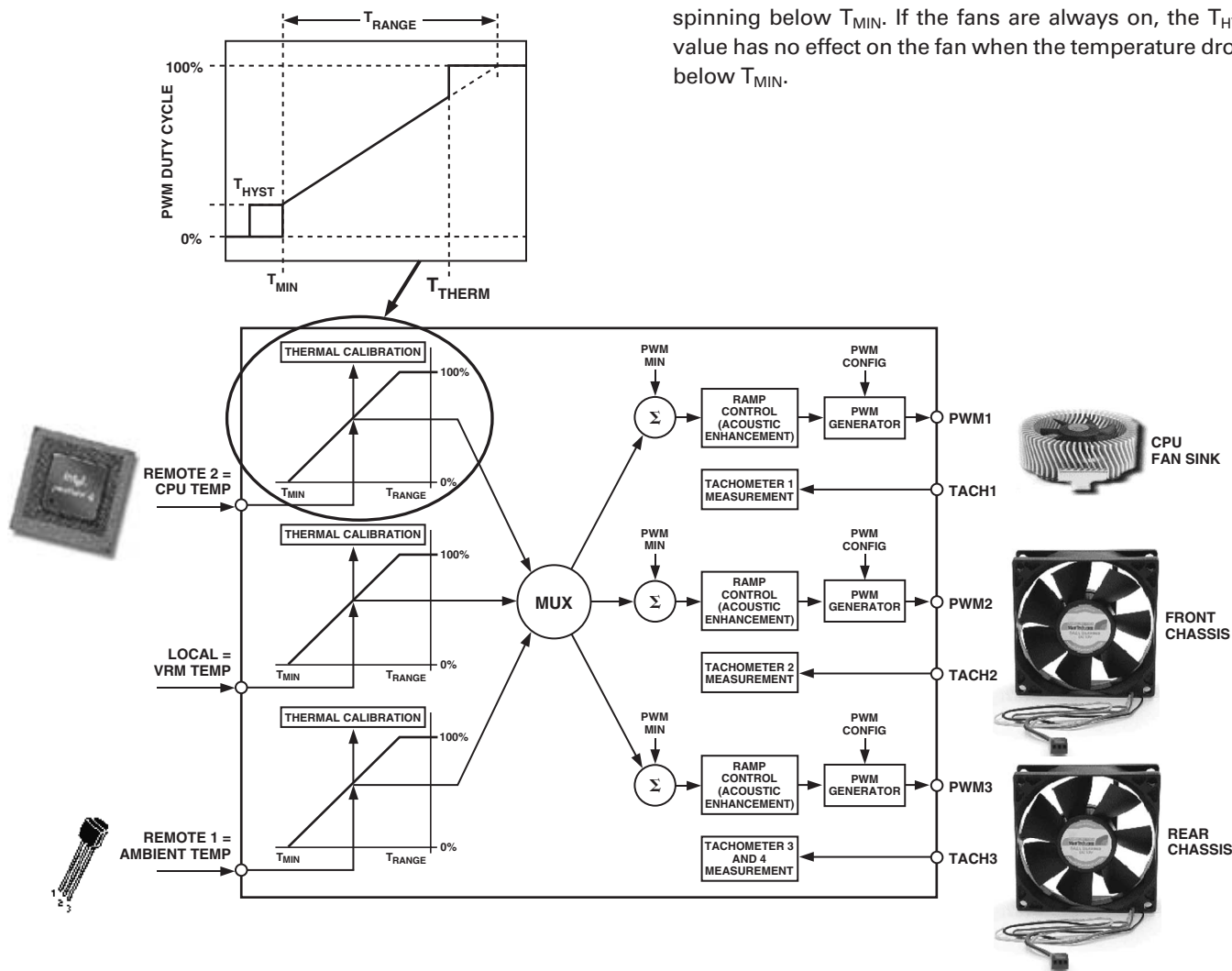


Figure 17. The  $T_{HYST}$  Value Applies to Fan On/Off Hysteresis and  $THERM$  Hysteresis

# AN-613

## ENHANCE ACOUSTICS REG 1 (REG. 0x62)

**Bit 7 (MIN3) = 0**, PWM3 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .

**Bit 7 (MIN3) = 1**, PWM3 runs at PWM3 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .

**Bit 6 (MIN2) = 0**, PWM2 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .

**Bit 6 (MIN2) = 1**, PWM2 runs at PWM2 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .

**Bit 5 (MIN1) = 0**, PWM1 is OFF (0% PWM duty cycle) when Temp is below  $T_{MIN} - T_{HYST}$ .

**Bit 5 (MIN1) = 1**, PWM1 runs at PWM1 minimum duty cycle below  $T_{MIN} - T_{HYST}$ .

## DYNAMIC $T_{MIN}$ CONTROL MODE

In addition to the automatic fan speed control mode described in the previous section, the ADT7460/ADT7463 have a mode that extends the basic automatic fan speed control loop. Dynamic  $T_{MIN}$  control allows the ADT7460/ADT7463 to intelligently adapt the system's cooling solution for best system performance or lowest possible system acoustics, depending on user or design requirements.

## AIM OF THIS SECTION

This section has two primary goals:

1. To show how dynamic  $T_{MIN}$  control alleviates the need for designing for worst-case conditions.
2. To illustrate how the dynamic  $T_{MIN}$  control function significantly reduces system design and validation time.

## DESIGNING FOR WORST-CASE CONDITIONS

When designing a system, you always design for worst-case conditions. In PC design, the worst-case conditions include, but are not limited to:

1. **Worst-Case Altitude.** A computer can be operated at different altitudes. The altitude affects the relative air density, which will alter the effectiveness of the fan cooling solution. For example, comparing 40°C air temperature at 10,000 ft to 20°C air temperature at sea level, relative air density is increased by 40%. This means that the fan can spin 40% slower, and make less noise, at sea level than at 10,000 ft while keeping the system at the same temperature at both locations.
2. **Worst-Case Fan.** Due to manufacturing tolerances, fan speeds in RPM are normally quoted with a tolerance of  $\pm 20\%$ . The designer needs to assume that the fan RPM can be 20% below tolerance. This translates to reduced system airflow and elevated system temperature. Note that fans 20% out of tolerance will negatively impact system acoustics since they run faster and generate more noise.

3. **Worst-Case Chassis Airflow.** The same motherboard can be used in a number of different chassis configurations. The design of the chassis and physical location of fans and components determine the system thermal characteristics. Moreover, for a given chassis, the addition of add-in cards, cables, or other system configuration options can alter the system airflow and reduce the effectiveness of the system cooling solution. The cooling solution can also be inadvertently altered by the end user, e.g., placing a computer against a wall can block the air ducts and reduce system airflow.

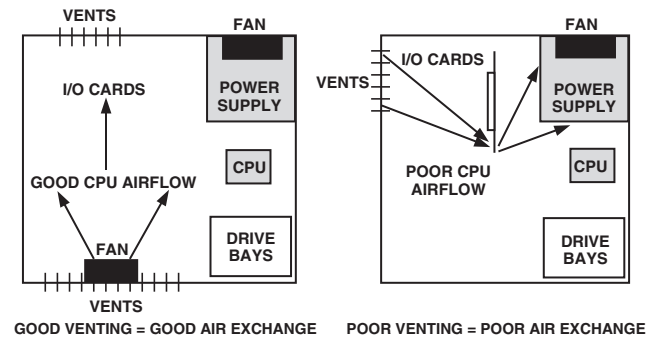


Figure 18. Chassis Airflow Issues

4. **Worst-Case Processor Power Consumption.** This is a data sheet maximum that does not necessarily reflect the true processor power consumption. Designing for worst-case CPU power consumption results in that the processor getting overcooled (generating excess system noise).
5. **Worst-Case Peripheral Power Consumptions.** The tendency is to design to data sheet maximums for these components (again overcooling the system).
6. **Worst-Case Assembly.** Every system manufactured is unique because of manufacturing variations. Heat sinks may be loose fitting or slightly misaligned. Too much or too little thermal grease may be used, or variations in application pressure for thermal interface material can affect the efficiency of the thermal solution. How can this be accounted for in every system? Again, the system is designed for the worst case.

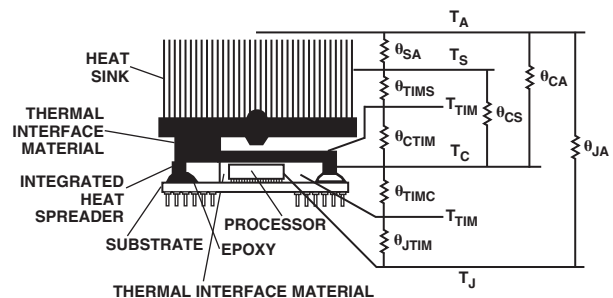


Figure 19. Thermal Model



The design usually accounts for worst-case conditions in all of these cases.

Note, however, that the actual system is almost never operated at worst-case conditions.

The alternative to designing for the worst case is to use the dynamic  $T_{MIN}$  control function.

### DYNAMIC $T_{MIN}$ CONTROL—OVERVIEW

Dynamic  $T_{MIN}$  Control mode builds upon the basic automatic fan control loop by adjusting the  $T_{MIN}$  value based on system performance and measured temperature. Why is this important?

Instead of designing for the worst case, the system thermals can be defined as “operating zones.” The ADT7460/ADT7463 will self-adjust its fan control loop to maintain an operating zone temperature or system target temperature. For example, you can specify that the ambient temperature in a system should be maintained at 50°C. If the temperature is below 50°C, the fans may not need to run or may run very slowly. If the temperature is higher than 50°C, the fans need to throttle up. How is this different from the automatic fan control mode?

The challenge presented by any thermal design is finding the right settings to suit the system’s fan control solution. This can involve designing for the worst case (as previously outlined), followed by weeks of system thermal characterization, and finally fan acoustic optimization (for psycho-acoustic reasons). Getting the most benefit from the automatic fan control mode involves characterizing the system to find the best  $T_{MIN}$  and  $T_{RANGE}$  settings for the control loop, and the best  $PWM_{MIN}$  value for the quietest fan speed setting. Using the ADT7460/ADT7463’s dynamic  $T_{MIN}$  control mode shortens the characterization time and alleviates tweaking the control loop settings because the device can self-adjust during system operation.

### DYNAMIC $T_{MIN}$ CONTROL—THE SPECIFICS

The dynamic  $T_{MIN}$  control mode is operated by specifying the “operating zone temperatures” required for the system. Associated with this control mode are three operating point registers, one for each temperature channel. This allows the system thermal solution to be broken down into distinct thermal zones, e.g., CPU operating temperature = 70°C, VRM operating temperature = 80°C, ambient operating temperature = 50°C. The ADT7460/ADT7463 will dynamically alter the control

solution to maintain each zone temperature as closely as possible to their target operating points.

### OPERATING POINT REGISTERS

Reg. 0x33 Remote 1 Operating Point = 0x64 (100°C)

Reg. 0x34 Local Temp Operating Point = 0x64 (100°C)

Reg. 0x35 Remote 2 Operating Point = 0x64 (100°C)

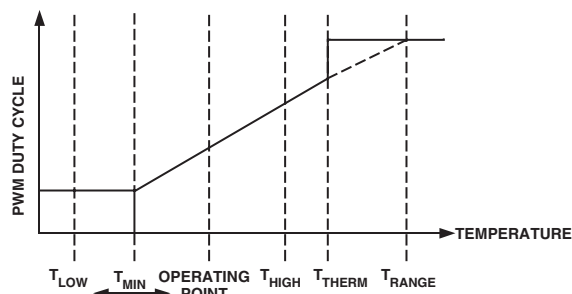


Figure 20. Dynamic  $T_{MIN}$  Control Loop

Figure 20 shows an overview of the parameters that affect the operation of the dynamic  $T_{MIN}$  control loop. A brief description of each parameter follows:

1.  $T_{LOW}$ . If temperature drops below the  $T_{LOW}$  limit, an error flag is set in a status register and an  $\overline{SMBALERT}$  interrupt can be generated.
2.  $T_{HIGH}$ . If temperature exceeds the  $T_{HIGH}$  limit, an error flag gets set in a status register and an  $\overline{SMBALERT}$  interrupt can be generated.
3.  $T_{MIN}$ . This is the temperature at which the fan turns on under automatic fan speed control.
4. Operating Point. This temperature defines the target temperature or optimal operating point for a particular temperature zone. The ADT7460/ADT7463 attempt to maintain system temperature at about the operating point by adjusting the  $T_{MIN}$  parameter of the control loop.
5.  $T_{THERM}$ . If temperature exceeds this critical limit, the fans can be run at 100% for maximum cooling.
6.  $T_{RANGE}$ . This programs the PWM duty cycle versus temperature control slope.

### DYNAMIC $T_{MIN}$ CONTROL PROGRAMMING

Since the dynamic  $T_{MIN}$  control mode is a basic extension of the automatic fan control mode, the automatic fan control mode parameters should be programmed first. Follow the seven steps in the Automatic Fan Control section of the ADT7460/ADT7463 data sheet before proceeding with dynamic  $T_{MIN}$  control mode programming.

## STEP 8: DETERMINING THE OPERATING POINT FOR EACH TEMPERATURE CHANNEL

The operating point for each temperature channel is the optimal temperature for that thermal zone. The hotter each zone is allowed to be, the quieter the system since the fans are not required to run at 100% all of the time. The ADT7460/ADT7463 will increase/decrease fan speeds as necessary to maintain operating point temperature. This allows for system-to-system variation and removes the need for worst-case design. As long as a sensible operating point value is chosen, any  $T_{MIN}$  value can be selected in the system characterization. If the  $T_{MIN}$  value is too low, the fans will run sooner than required, and the temperature will be below the operating point. In response, the ADT7460/ADT7463 will increase  $T_{MIN}$  to keep the fans off for longer and allow

the temperature zone to get closer to the operating point. Likewise, too high a  $T_{MIN}$  value will cause the operating point to be exceeded, and in turn, the ADT7460/ADT7463 will reduce  $T_{MIN}$  to turn the fans on earlier to cool the system.

## PROGRAMMING OPERATING POINT REGISTERS

There are three operating point registers, one associated with each temperature channel. These 8-bit registers allow the operating point temperatures to be programmed with 1°C resolution.

## OPERATING POINT REGISTERS

Reg. 0x33 Remote 1 Operating Point = 0x64 (100°C)

Reg. 0x34 Local Temp Operating Point = 0x64 (100°C)

Reg. 0x35 Remote 2 Operating Point = 0x64 (100°C)

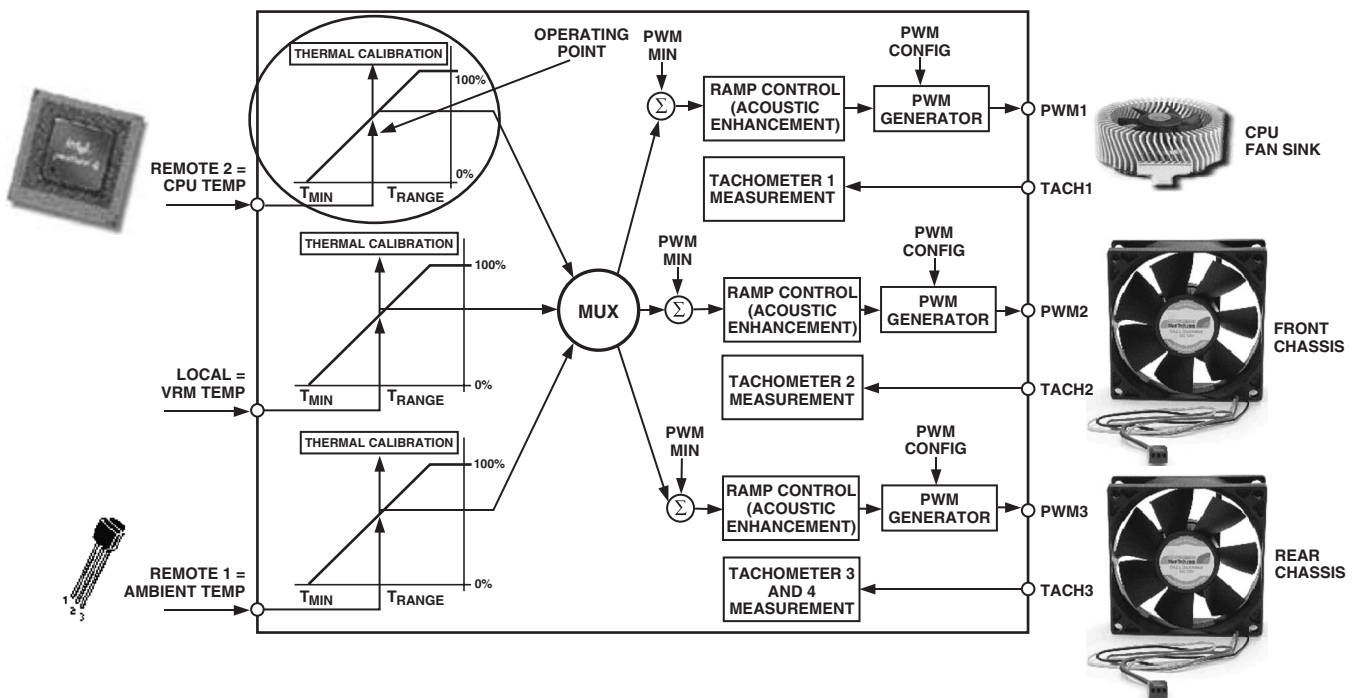


Figure 21. Operating Point Value Dynamically Adjusts Automatic Fan Control Settings

### STEP 9: DETERMINING THE HIGH AND LOW LIMITS FOR EACH TEMPERATURE CHANNEL

The low limit defines the temperature at which the  $T_{MIN}$  value will start to be increased if temperature falls below this value. This has the net effect of reducing the fan speed, allowing the system to get hotter. An interrupt can be generated when the temperature drops below the low limit.

The high limit defines the temperature at which the  $T_{MIN}$  value will start to be reduced if temperature increases above this value. This has the net effect of increasing fan speed in order to cool down the system. An interrupt can be generated when the temperature rises above the high limit.

### PROGRAMMING HIGH AND LOW LIMITS

There are six limit registers; a high limit and low limit are associated with each temperature channel. These 8-bit registers allow the high and low limit temperatures to be programmed with 1°C resolution.

### TEMPERATURE LIMIT REGISTERS

Reg. 0x4E Remote 1 Temp Low Limit = 0x81  
 Reg. 0x4F Remote 1 Temp High Limit = 0x7F  
 Reg. 0x50 Local Temp Low Limit = 0x81  
 Reg. 0x51 Local Temp High Limit = 0x7F  
 Reg. 0x52 Remote 2 Temp Low Limit = 0x81  
 Reg. 0x53 Remote 2 Temp High Limit = 0x7F

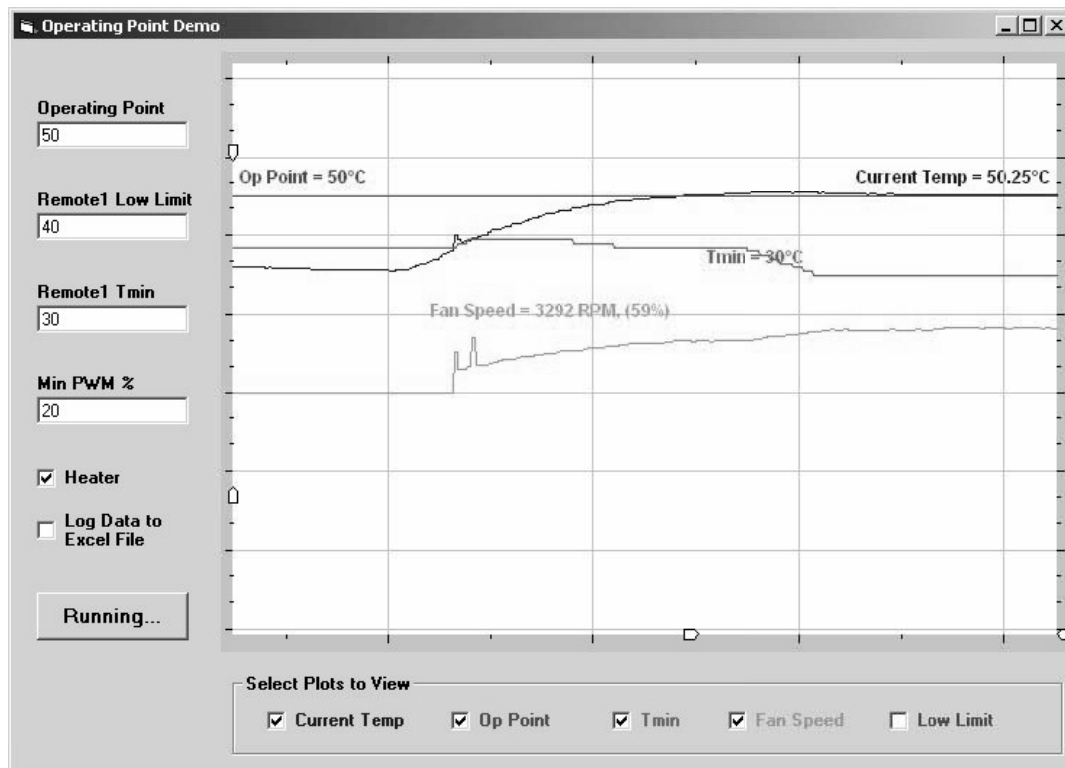


Figure 22. Dynamic  $T_{MIN}$  Control in Operation

## HOW DOES DYNAMIC $T_{MIN}$ CONTROL WORK?

The basic premise is as follows:

1. Set the target temperature for the temperature zone, which could be, for example, the Remote 1 thermal diode. This value is programmed to the Remote 1 operating temperature register.
2. As the temperature in that zone (Remote 1 temperature) rises toward and exceeds the operating point temperature,  $T_{MIN}$  is reduced and the fan speed increases.
3. As the temperature drops below the operating point temperature,  $T_{MIN}$  is increased, reducing the fan speed.

The loop operation is not as simple as described above. There are a number of conditions governing situations in which  $T_{MIN}$  can increase or decrease.

### SHORT CYCLE AND LONG CYCLE

The ADT7460/ADT7463 implement two loops, a short cycle and a long cycle. The short cycle takes place every  $n$  monitoring cycles. The long cycle takes place every  $2n$  monitoring cycles. The value of  $n$  is programmable for each temperature channel. The bits are located at the following register locations:

Remote 1 = CYR1 = Bits <2:0> of Calibration Control Register 2 (Addr = 0x37)

Local = CYL = Bits <5:3> of Calibration Control Register 2 (Addr = 0x37)

Remote 2 = CYR2 = Bits <7:6> of Calibration Control Register 2 and Bit 0 of Calibration Control Register 1 (Addr = 0x36)

Table II. Cycle Bit Assignments

CODE	Short Cycle	Long Cycle
000	8 cycles (1 s)	16 cycles (2 s)
001	16 cycles (2 s)	32 cycles (4 s)
010	32 cycles (4 s)	64 cycles (8 s)
011	64 cycles (8 s)	128 cycles (16 s)
100	128 cycles (16 s)	256 cycles (32 s)
101	256 cycles (32 s)	512 cycles (64 s)
110	512 cycles (64 s)	1024 cycles (128 s)
111	1024 cycles (128 s)	2048 cycles (256 s)

Care should be taken in choosing the cycle time. A long cycle time means that the  $T_{MIN}$  is not updated very often; if your system has very fast temperature transients, the dynamic  $T_{MIN}$  control loop will always be lagging. If you choose a cycle time that is too fast, the full benefit of changing  $T_{MIN}$  may not have been realized and you change again on the next cycle; in effect you would be overshooting. It is necessary to carry out some calibration to identify the most suitable response time.

**SHORT CYCLE**

Figure 23 displays the steps taken during the short cycle.

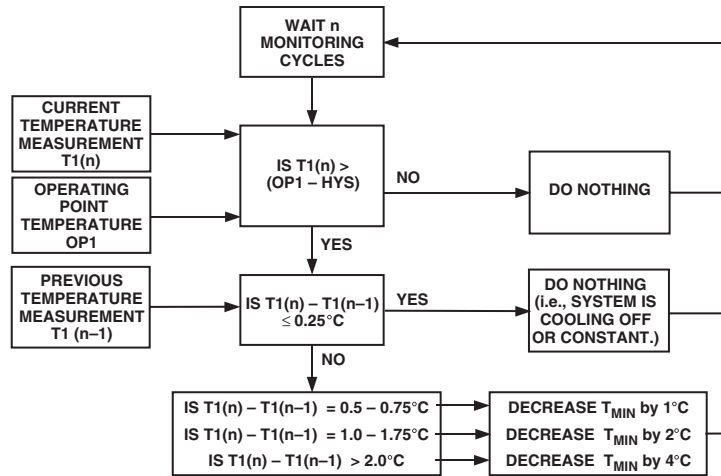


Figure 23. Short Cycle

**LONG CYCLE**

Figure 24 displays the steps taken during the long cycle.

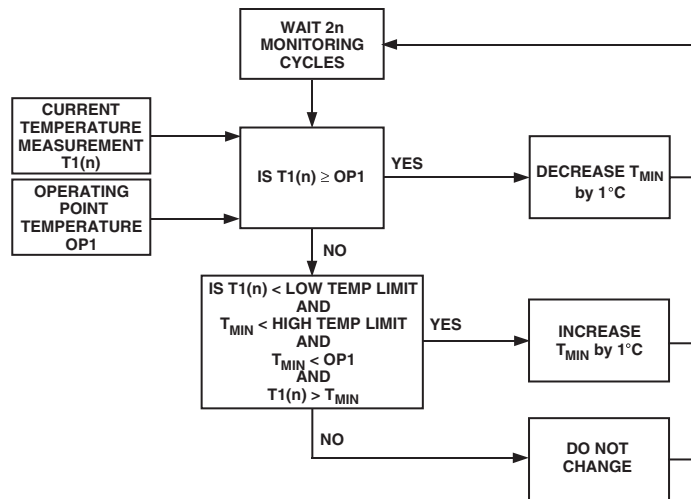


Figure 24. Long Cycle

## EXAMPLES

The following are examples of some circumstances that may cause  $T_{MIN}$  to increase or decrease or stay the same.

### NORMAL OPERATION—NO $T_{MIN}$ ADJUSTMENT

1. If measured temperature never exceeds the programmed operating point-hysteresis temperature, then  $T_{MIN}$  is not adjusted, i.e., remains at its current setting.
2. If measured temperature never drops below the low temperature limit, then  $T_{MIN}$  is not adjusted.

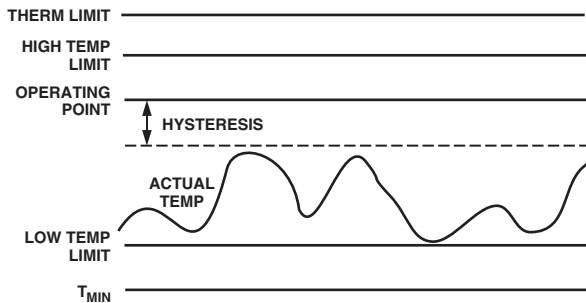


Figure 25. Temperature between Operating Point and Low Temperature Limit

Since neither the operating point-hysteresis temperature nor the low temperature limit has been exceeded, the  $T_{MIN}$  value is not adjusted and the fan runs at a speed determined by the fixed  $T_{MIN}$  and  $T_{RANGE}$  values defined in the automatic fan speed control mode.

### OPERATING POINT EXCEEDED— $T_{MIN}$ REDUCED

When the measured temperature is below the operating point temperature less the hysteresis,  $T_{MIN}$  remains the same.

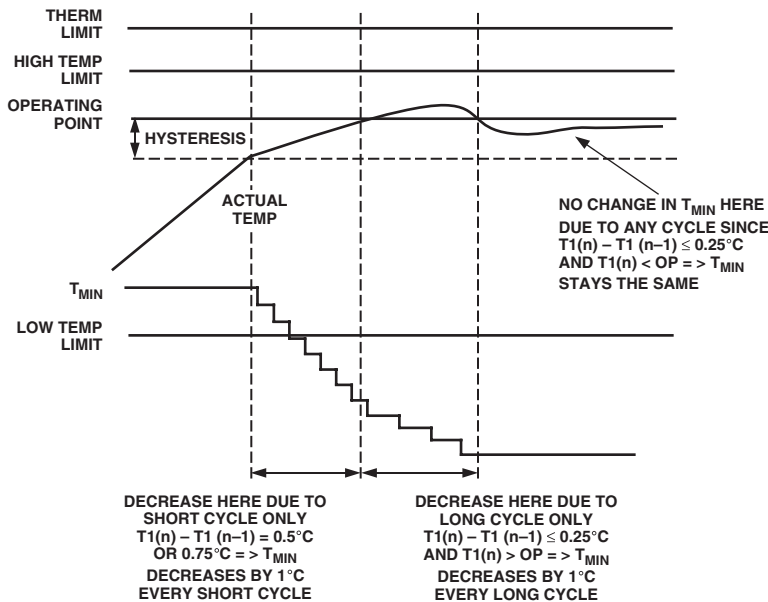


Figure 26. Effect of Exceeding Operating Point - Hysteresis Temperature

Once the temperature exceeds the operating temperature less the hysteresis ( $OP - Hys$ ), the  $T_{MIN}$  starts to decrease. This occurs during the short cycle; see Figure 23. The rate with which  $T_{MIN}$  decreases depends on the programmed value of  $n$ . It also depends on how much the temperature has increased between this monitoring cycle and the last monitoring cycle, i.e., if the temperature has increased by  $1^{\circ}C$ , then  $T_{MIN}$  is reduced by  $2^{\circ}C$ . Decreasing  $T_{MIN}$  has the effect of increasing the fan speed, thus providing more cooling to the system.

If the temperature is only slowly increasing in the range ( $OP - Hys$ ), i.e.,  $\leq 0.25^{\circ}C$  per short monitoring cycle, then  $T_{MIN}$  does not decrease. This allows small changes in temperature in the desired operating zone without changing  $T_{MIN}$ . The long cycle makes no change to  $T_{MIN}$  in the temperature range ( $OP - Hys$ ) since the temperature has not exceeded the operating temperature.

Once the temperature exceeds the operating temperature, the long cycle will cause  $T_{MIN}$  to reduce by  $1^{\circ}C$  every long cycle while the temperature remains above the operating temperature. This takes place in addition to the decrease in  $T_{MIN}$  that would occur due to the short cycle. In Figure 26, since the temperature is only increasing at a rate less than or equal to  $0.25^{\circ}C$  per short cycle, no reduction in  $T_{MIN}$  takes place during the short cycle.

Once the temperature has fallen below the operating temperature,  $T_{MIN}$  stays the same. Even when the temperature starts to increase slowly,  $T_{MIN}$  stays the same because the temperature increases at a rate  $\leq 0.25^{\circ}C$  per cycle.



**INCREASE  $T_{MIN}$  CYCLE**

When the temperature drops below the low temperature limit,  $T_{MIN}$  can increase in the long cycle. Increasing  $T_{MIN}$  has the effect of running the fan slower and therefore quieter. The long cycle diagram in Figure 24 shows the conditions that need to be true for  $T_{MIN}$  to increase. Here is a quick summary of those conditions and the reasons they need to be true.

$T_{MIN}$  can increase if

1. The measured temperature has fallen below the low temperature limit. This means the user must choose the low limit carefully. It should not be so low that the temperature will never fall below it because  $T_{MIN}$  would never increase and the fans would run faster than necessary.

AND

2.  $T_{MIN}$  is below the high temperature limit.  $T_{MIN}$  is never allowed to increase above the high temperature limit. As a result, the high limit should be sensibly chosen because it determines how high  $T_{MIN}$  can go.

AND

3.  $T_{MIN}$  is below the operating point temperature.  $T_{MIN}$  should never be allowed to increase above the operating point temperature since the fans would not switch on until the temperature rose above the operating point.

AND

4. The temperature is above  $T_{MIN}$ . The dynamic  $T_{MIN}$  control is turned off below  $T_{MIN}$ .

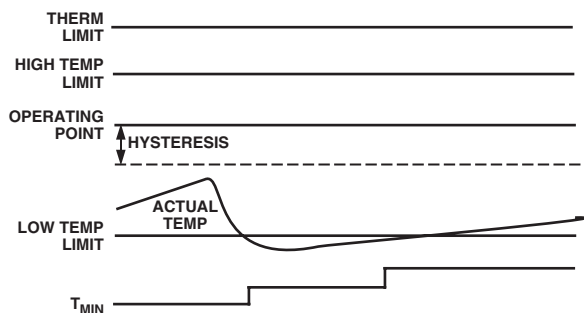


Figure 27. Increasing  $T_{MIN}$  for Quieter Operation

Figure 27 shows how  $T_{MIN}$  increases when the current temperature is above  $T_{MIN}$  and below the low temperature limit, and  $T_{MIN}$  is below the high temperature limit and below the operating point. Once the temperature rises above the low temperature limit,  $T_{MIN}$  stays the same.

**WHAT PREVENTS  $T_{MIN}$  FROM REACHING FULL SCALE?**

Since  $T_{MIN}$  is dynamically adjusted, it is undesirable for  $T_{MIN}$  to reach full scale (127°C) because the fan would never switch on. As a result,  $T_{MIN}$  is allowed to vary only within a specified range:

1. The lowest possible value to  $T_{MIN}$  is -127°C.
2.  $T_{MIN}$  cannot exceed the high temperature limit.
3. If the temperature is below  $T_{MIN}$ , the fan is switched off or is running at minimum speed and dynamic  $T_{MIN}$  control is disabled.

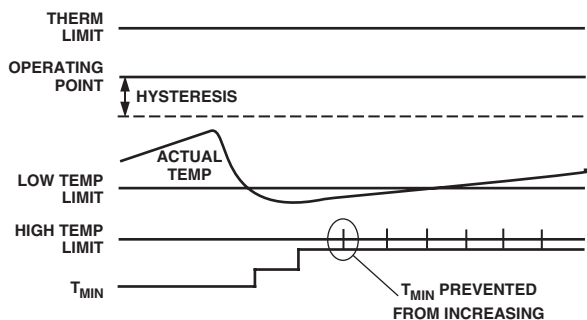


Figure 28.  $T_{MIN}$  Adjustments Limited by the High Temperature Limit

**STEP 10: DETERMINING WHETHER TO MONITOR  $\overline{\text{THERM}}$** 

Using the operating point limit ensures that the dynamic  $T_{\text{MIN}}$  control mode is operating in the best possible acoustic position while ensuring that the temperature never exceeds the maximum operating temperature. Using the operating point limit allows the  $T_{\text{MIN}}$  to be independent of system level issues because of its self-corrective nature.

In PC design, the operating point for the chassis is usually the worst-case internal chassis temperature.

The optimal operating point for the processor is determined by monitoring the thermal monitor in the Intel Pentium® 4 processor. To do this, the  $\overline{\text{PROCHOT}}$  output of the Pentium 4 is connected to the  $\overline{\text{THERM}}$  input of the ADT7460/ADT7463.

The operating point for the processor can be determined by allowing the current temperature to be copied to the operating point register when the  $\overline{\text{PROCHOT}}$  output pulls the  $\overline{\text{THERM}}$  input low on the ADT7460/ADT7463. This gives the maximum temperature at which the Pentium 4 can be run before clock modulation occurs.

**ENABLING  $\overline{\text{THERM}}$  TRIP POINT AS THE OPERATING POINT**

Bits <4:2> of dynamic  $T_{\text{MIN}}$  control Register 1 (Reg. 0x36) enable/disable  $\overline{\text{THERM}}$  monitoring to program the operating point.

**DYNAMIC  $T_{\text{MIN}}$  CONTROL REGISTER 1 (0x36)**

<2> **PHTR2 = 1** copies the Remote 2 current temperature to the Remote 2 operating point register if  $\overline{\text{THERM}}$  gets asserted. The operating point will contain the temperature at which  $\overline{\text{THERM}}$  is asserted. This allows the system to run as quietly as possible without system performance being affected.

**PHTR2 = 0** ignores any  $\overline{\text{THERM}}$  assertions. The Remote 2 operating point register will reflect its programmed value.

<3> **PHTL = 1** copies the local current temperature to the local temperature operating point register if  $\overline{\text{THERM}}$  gets asserted. The operating point will contain the temperature at which  $\overline{\text{THERM}}$  is asserted. This allows the system to run as quietly as possible without system performance being affected.

**PHTL = 0** ignores any  $\overline{\text{THERM}}$  assertions. The local temperature operating point register will reflect its programmed value.

<4> **PHTR1 = 1** copies the Remote 1 current temperature to the Remote 1 operating point register if  $\overline{\text{THERM}}$  gets asserted. The operating point will contain the temperature at which  $\overline{\text{THERM}}$  is asserted. This allows the system to run as quietly as possible without affecting system performance.

**PHTR1 = 0** ignores any  $\overline{\text{THERM}}$  assertions. The Remote 1 operating point register will reflect its programmed value.

**ENABLING DYNAMIC  $T_{\text{MIN}}$  CONTROL MODE**

Bits <7:5> of dynamic  $T_{\text{MIN}}$  control Register 1 (Reg. 0x36) enable/disable dynamic  $T_{\text{MIN}}$  control on the temperature channels.

**DYNAMIC  $T_{\text{MIN}}$  CONTROL REGISTER 1 (0x36)**

<5> **R2T = 1** enables dynamic  $T_{\text{MIN}}$  control on the Remote 2 temperature channel. The chosen  $T_{\text{MIN}}$  value will be dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

**R2T = 0** disables dynamic  $T_{\text{MIN}}$  control. The  $T_{\text{MIN}}$  value chosen will not be adjusted and the channel will behave as described in the Automatic Fan Control section.

<6> **LT = 1** enables dynamic  $T_{\text{MIN}}$  control on the local temperature channel. The chosen  $T_{\text{MIN}}$  value will be dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

**LT = 0** disables dynamic  $T_{\text{MIN}}$  control. The  $T_{\text{MIN}}$  value chosen will not be adjusted and the channel will behave as described in the Automatic Fan Control section.

<7> **R1T = 1** enables dynamic  $T_{\text{MIN}}$  control on the Remote 1 temperature channel. The chosen  $T_{\text{MIN}}$  value will be dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

**R1T = 0** disables dynamic  $T_{\text{MIN}}$  control. The  $T_{\text{MIN}}$  value chosen will not be adjusted and the channel will behave as described in the Automatic Fan Control section.

## ENHANCING SYSTEM ACOUSTICS

Automatic fan speed control mode reacts instantaneously to changes in temperature, i.e., the PWM duty cycle will respond immediately to temperature change. Any impulses in temperature can cause an impulse in fan noise. For psycho-acoustic reasons, the ADT7460/ADT7463 can prevent the PWM output from reacting instantaneously to temperature changes. Enhanced acoustic mode will control the maximum change in PWM duty cycle in a given time. The objective is to prevent the fan from cycling up and down and annoying the system user.

## ACOUSTIC ENHANCEMENT MODE OVERVIEW

Figure 29 gives a top-level overview of the automatic fan control circuitry on the ADT7460/ADT7463 and where acoustic enhancement fits in. Acoustic enhancement is intended as a post-design “tweak” made by a system or mechanical engineer evaluating best settings for the system. Having determined the optimal settings for the thermal solution, the engineer can adjust the system acoustics. The goal is to implement a system that is acoustically pleasing without causing user annoyance due to fan cycling. It is important to realize that although a system may pass an acoustic noise requirement spec, (e.g., 36 dB), if the fan is annoying, it will fail the consumer test.

## THE APPROACH

There are two different approaches to implementing system acoustic enhancement. The first method is temperature-centric. It involves “smoothing” transient temperatures as they are measured by a temperature source, e.g., Remote 1 temperature.

The temperature values used to calculate the PWM duty cycle values would be smoothed, reducing fan speed variation. However, this approach would cause an inherent delay in updating fan speed and would cause the thermal characteristics of the system to change. It would also cause the system fans to stay on longer than necessary, since the fan’s reaction is merely delayed. The user would also have no control over noise from different fans driven by the same temperature source. Consider controlling a CPU cooler fan (on PWM1) and a chassis fan (on PWM2) using Remote 1 temperature. Because the Remote 1 temperature is smoothed, both fans would be updated at exactly the same rate. If the chassis fan is much louder than the CPU fan, there is no way to improve its acoustics without changing the thermal solution of the CPU cooling fan.

The second approach is fan-centric. The idea is to control the PWM duty cycle driving the fan at a fixed rate, e.g., 6%. Each time the PWM duty cycle is updated, it is incremented by a fixed 6%. As a result, the fan ramps

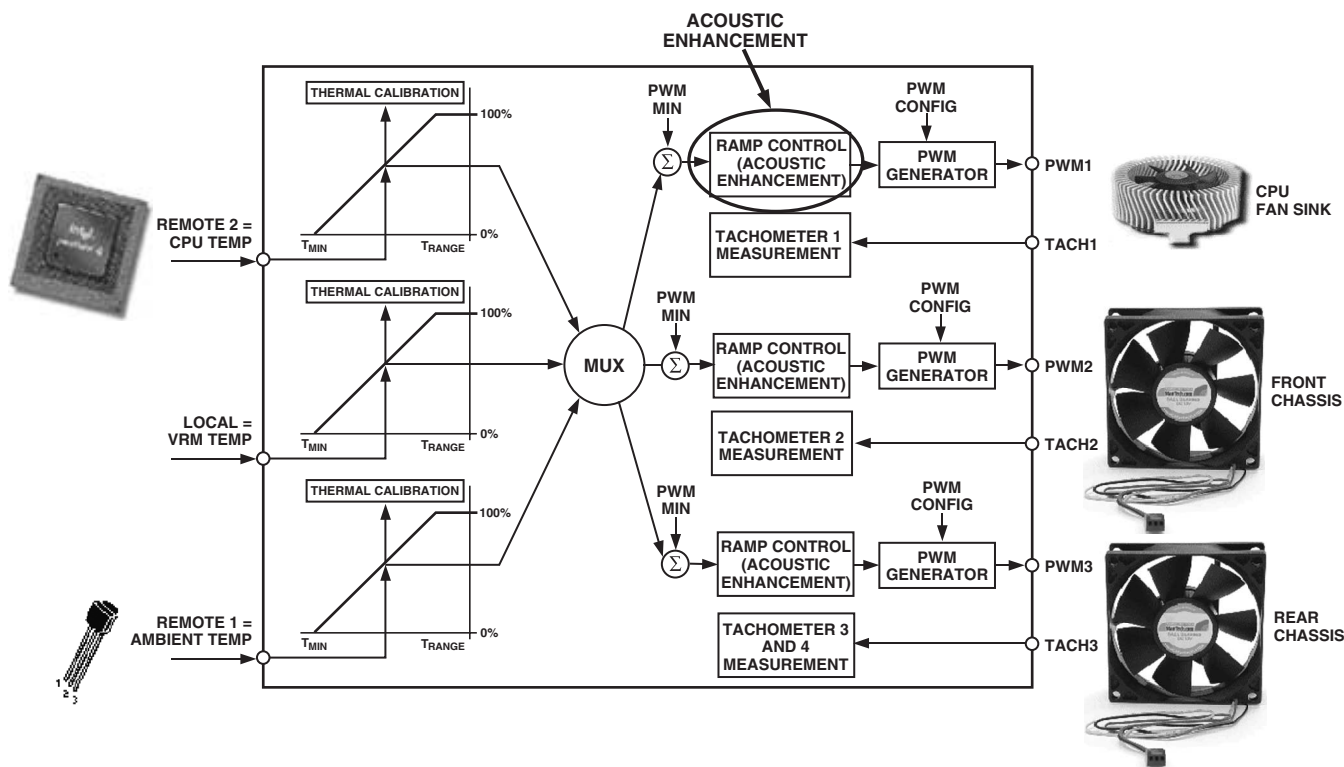


Figure 29. Acoustic Enhancement Smooths Fan Speed Variations under Automatic Fan Speed Control

smoothly to its newly calculated speed. If the temperature starts to drop, the PWM duty cycle immediately decreases by 6% every update. So the fan ramps smoothly up or down without inherent system delay. Consider controlling the same CPU cooler fan (on PWM1) and chassis fan (on PWM2) using Remote 1 temperature. The  $T_{MIN}$  and  $T_{RANGE}$  settings have already been defined in automatic fan speed control mode, i.e., thermal characterization of the control loop has been optimized. Now the chassis fan is noisier than the CPU cooling fan. So PWM2 can be placed into acoustic enhancement mode independently of PWM1. The acoustics of the chassis fan can therefore be adjusted without affecting the acoustic behavior of the CPU cooling fan, even though both fans are being controlled by Remote 1 temperature. This is exactly how acoustic enhancement works on the ADT7460/ADT7463.

### ENABLING ACOUSTIC ENHANCEMENT FOR EACH PWM OUTPUT

#### ENHANCE ACOUSTICS REGISTER 1 (Reg. 0x62)

<3> = 1 Enables acoustic enhancement on PWM1 output.

#### ENHANCE ACOUSTICS REGISTER 2 (Reg. 0x63)

<7> = 1 Enables acoustic enhancement on PWM2 output.

<3> = 1 Enables acoustic enhancement on PWM3 output.

### EFFECT OF RAMP RATE ON ENHANCED ACOUSTICS MODE

The PWM signal driving the fan will have a period,  $T$ , given by the PWM drive frequency,  $f$ , since  $T = 1/f$ . For a given PWM period,  $T$ , the PWM period is subdivided into 255 equal time slots. One time slot corresponds to the smallest possible increment in PWM duty cycle. A PWM signal of 33% duty cycle will thus be high for  $1/3 \times 255$  time slots and low for  $2/3 \times 255$  time slots. Therefore, 33% PWM duty cycle corresponds to a signal that is high for 85 time slots and low for 170 time slots.

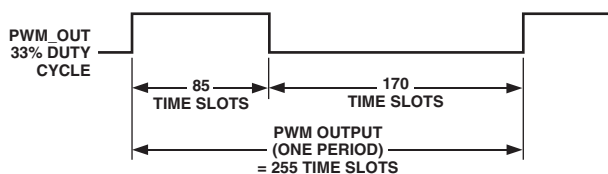


Figure 30. 33% PWM Duty Cycle Represented in Time Slots

The ramp rates in the enhanced acoustics mode are selectable from the values 1, 2, 3, 5, 8, 12, 24, and 48. The ramp rates are actually discrete time slots. For example, if the ramp rate = 8, then eight time slots will be added to the PWM high duty cycle each time the PWM duty cycle needs to be increased. If the PWM duty cycle value needs to be decreased, it will be decreased by eight time slots. Figure 31 shows how the enhanced acoustics mode algorithm operates.

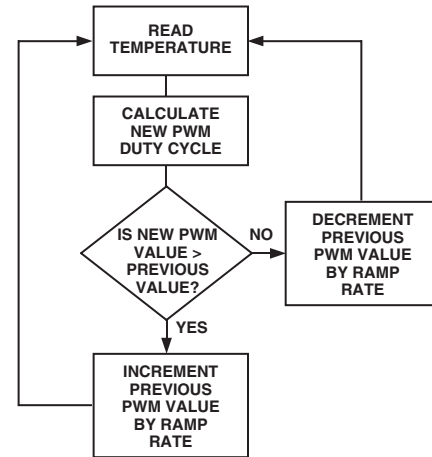


Figure 31. Enhanced Acoustics Algorithm

The enhanced acoustics mode algorithm calculates a new PWM duty cycle based on the temperature measured. If the new PWM duty cycle value is greater than the previous PWM value, the previous PWM duty cycle value is incremented by either 1, 2, 3, 5, 8, 12, 24, or 48 time slots, depending on the settings of the enhance acoustics registers. If the new PWM duty cycle value is less than the previous PWM value, the previous PWM duty cycle is decremented by 1, 2, 3, 5, 8, 12, 24, or 48 time slots. Each time the PWM duty cycle is incremented or decremented, it is stored as the previous PWM duty cycle for the next comparison.

A ramp rate of 1 corresponds to one time slot, which is  $1/255$  of the PWM period. In enhanced acoustics mode, incrementing or decrementing by 1 changes the PWM output by  $1/255 \times 100\%$ .

### STEP 11: DETERMINING THE RAMP RATE FOR ACOUSTIC ENHANCEMENT

The optimal ramp rate for acoustic enhancement can be found through system characterization after the thermal optimization has been finished. The effect of each ramp rate should be logged, if possible, to determine the best setting for a given solution.

#### ENHANCE ACOUSTICS REGISTER 1 (Reg. 0x62)

<2:0> ACOU Select the Ramp Rate for PWM1.

- 000 = 1 Time Slot = 35 seconds
- 001 = 2 Time Slots = 17.6 seconds
- 010 = 3 Time Slots = 11.8 seconds
- 011 = 5 Time Slots = 7 seconds
- 100 = 8 Time Slots = 4.4 seconds
- 101 = 12 Time Slots = 3 seconds
- 110 = 24 Time Slots = 1.6 seconds
- 111 = 48 Time Slots = 0.8 seconds

**ENHANCE ACOUSTICS REGISTER 2 (Reg. 0x63)**

<2:0> ACOU3 Select the ramp rate for PWM3.

- 000 = 1 Time Slot = 35 seconds
- 001 = 2 Time Slots = 17.6 seconds
- 010 = 3 Time Slots = 11.8 seconds
- 011 = 5 Time Slots = 7 seconds
- 100 = 8 Time Slots = 4.4 seconds
- 101 = 12 Time Slots = 3 seconds
- 110 = 24 Time Slots = 1.6 seconds
- 111 = 48 Time Slots = 0.8 seconds

<6:4> ACOU2 Select the ramp rate for PWM2.

- 000 = 1 Time Slot = 35 seconds
- 001 = 2 Time Slots = 17.6 seconds
- 010 = 3 Time Slots = 11.8 seconds
- 011 = 5 Time Slots = 7 seconds
- 100 = 8 Time Slots = 4.4 seconds
- 101 = 12 Time Slots = 3 seconds
- 110 = 24 Time Slots = 1.6 seconds
- 111 = 48 Time Slots = 0.8 seconds

Another way to view the ramp rates is the time it takes for the PWM output to ramp from 0% to 100% duty cycle for an instantaneous change in temperature. This can be tested by putting the ADT7460/ADT7463 into manual mode and changing the PWM output from 0% to 100% PWM duty cycle. The PWM output takes 35 seconds to reach 100% with a ramp rate of 1 time slot selected.

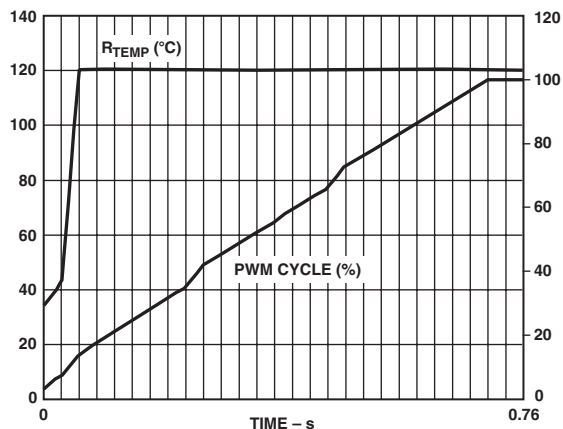


Figure 32. Enhanced Acoustics Mode with Ramp Rate = 48

Figure 32 shows remote temperature plotted against PWM duty cycle for enhanced acoustics mode. The ramp rate is set to 48, which would correspond to the fastest ramp rate. Assume that a new temperature reading is available every 115 ms. With these settings, it took approximately 0.76 seconds to go from 33% duty cycle to 100% duty cycle (full speed). Even though the temperature increased very rapidly, the fan ramps up to full speed gradually.

Figure 33 shows how changing the ramp rate from 48 to 8 affects the control loop. The overall response of the fan is slower. Since the ramp rate is reduced, it takes longer for the fan to achieve full running speed. In this case, it took approximately 4.4 seconds for the fan to reach full speed.

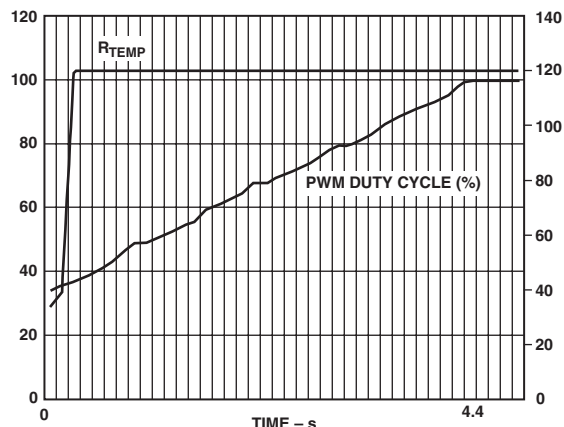


Figure 33. Enhanced Acoustics Mode with Ramp Rate = 8

Figure 34 shows the PWM output response for a ramp rate of 2. In this instance, the fan took about 17.6 seconds to reach full running speed.

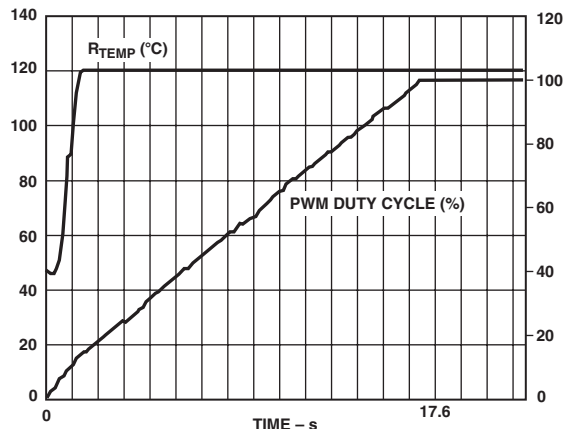


Figure 34. Enhanced Acoustics Mode with Ramp Rate = 2

Finally, Figure 35 shows how the control loop reacts to temperature with the slowest ramp rate. The ramp rate is set to 1, while all other control parameters remain the same. With the slowest ramp rate selected, it takes 35 seconds for the fan to reach full speed.

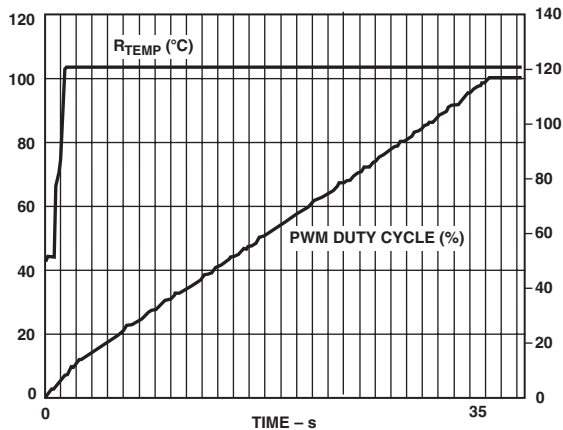


Figure 35. Enhanced Acoustics Mode with Ramp Rate = 1

As Figures 32 to 35 show, the rate at which the fan will react to temperature change is dependent on the ramp rate selected in the enhance acoustics registers. The higher the ramp rate, the faster the fan will reach the newly calculated fan speed.

Figure 36 shows the behavior of the PWM output as temperature varies. As the temperature is rising, the fan speed ramps up. Small drops in temperature will not affect the ramp-up function since the newly calculated fan speed will still be higher than the previous PWM value. The enhance acoustics mode allows the PWM output to be made less sensitive to temperature variations. This will be dependent on the ramp rate selected and programmed into the enhance acoustics registers.

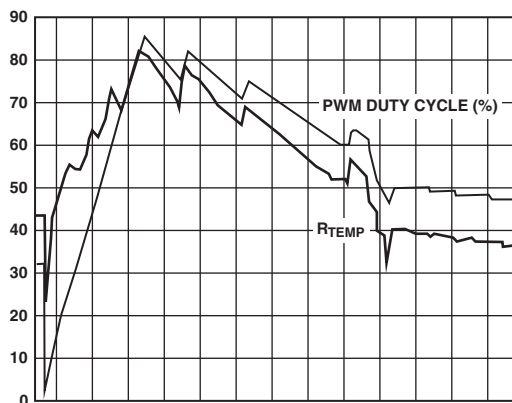


Figure 36. How Fan Reacts to Temperature Variation in Enhanced Acoustics Mode

## SLOWER RAMP RATES

The ADT7460/ADT7463 can be programmed for much longer ramp times by slowing the ramp rates. Each ramp rate can be slowed by a factor of 4.

### PWM1 CONFIGURATION REGISTER (Reg. 0x5C)

<3> SLOW = 1 slows the ramp rate for PWM1 by 4.

### PWM2 CONFIGURATION REGISTER (Reg. 0x5D)

<3> SLOW = 1 slows the ramp rate for PWM2 by 4.

### PWM3 CONFIGURATION REGISTER (Reg. 0x5E)

<3> SLOW = 1 slows the ramp rate for PWM3 by 4.

The following shows the ramp-up times when the SLOW bit is set for each PWM output.

### ENHANCE ACOUSTICS REGISTER 1 (Reg. 0x62)

<2:0> ACOU Select the ramp rate for PWM1.

- 000 = 140 seconds
- 001 = 70.4 seconds
- 010 = 47.2 seconds
- 011 = 28 seconds
- 100 = 17.6 seconds
- 101 = 12 seconds
- 110 = 6.4 seconds
- 111 = 3.2 seconds

### ENHANCE ACOUSTICS REGISTER 2 (Reg. 0x63)

<2:0> ACOU3 Select the ramp rate for PWM3.

- 000 = 140 seconds
- 001 = 70.4 seconds
- 010 = 47.2 seconds
- 011 = 28 seconds
- 100 = 17.6 seconds
- 101 = 12 seconds
- 110 = 6.4 seconds
- 111 = 3.2 seconds

<6:4> ACOU2 Select the ramp rate for PWM2.

- 000 = 140 seconds
- 001 = 70.4 seconds
- 010 = 47.2 seconds
- 011 = 28 seconds
- 100 = 17.6 seconds
- 101 = 12 seconds
- 110 = 6.4 seconds
- 111 = 3.2 seconds





