



## **Thermal Monitoring for Mellanox Systems with third party OS**

**Rev 0.1**

[www.mellanox.com](http://www.mellanox.com)

NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Mellanox Technologies, Inc.  
350 Oakmead Parkway Suite 100  
Sunnyvale, CA 94085  
U.S.A.  
[www.mellanox.com](http://www.mellanox.com)  
Tel: (408) 970-3400  
Fax: (408) 970-3403

Mellanox Technologies, Ltd.  
Beit Mellanox  
PO Box 586 Yokneam 20692  
Israel  
[www.mellanox.com](http://www.mellanox.com)  
Tel: +972 (0)4 909 7200 ; +972 (0)74 723 7200  
Fax: +972 (0)4 959 3245

© Copyright 2012. Mellanox Technologies. All rights reserved.

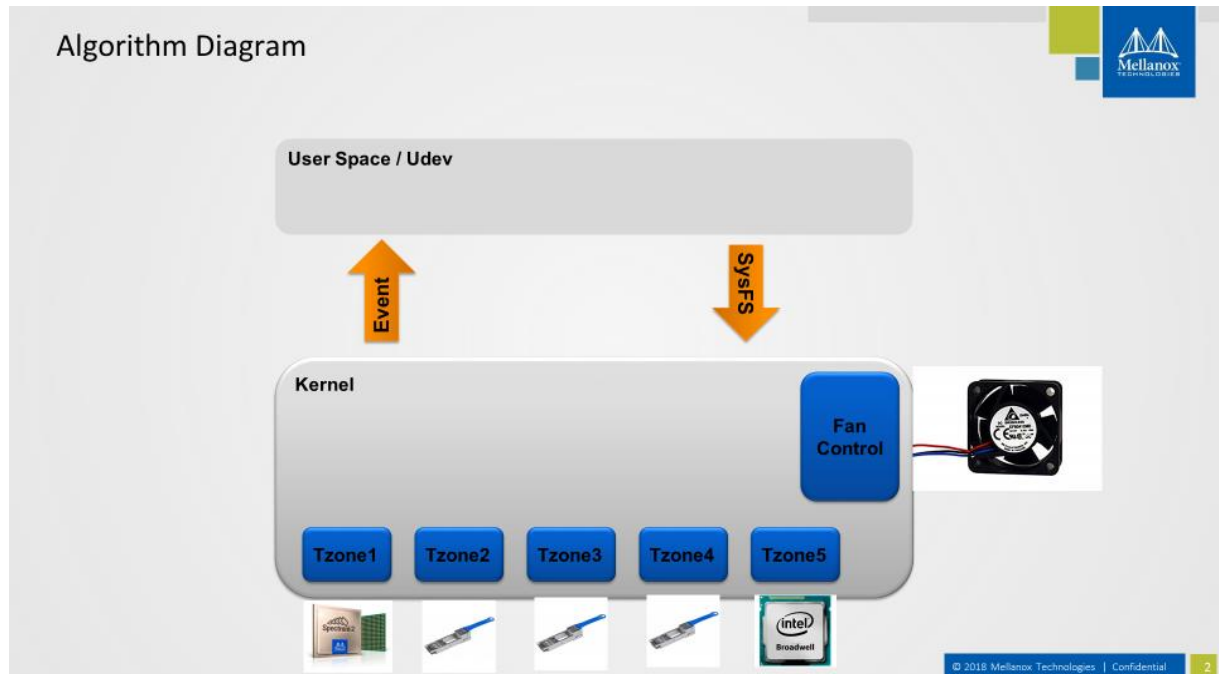
Mellanox®, BridgeX®, ConnectX®, CORE-Direct®, InfiniBridge®, InfiniHost®, InfiniScale®, PhyX®, Switch-X2®, Virtual Protocol Interconnect® and Voltaire® are registered trademarks of Mellanox Technologies, Ltd.

FabricIT™, MLNX-OS™ and Unbreakable-Link™ are trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
<b>2</b>	<b>Current Methodology and Problems Being Addressed .....</b>	<b>5</b>
<b>3</b>	<b>Solution summary.....</b>	<b>6</b>
3.1	Thermal monitoring architectural view.....	6



6

<b>4</b>	<b>Detailed Explanation.....</b>	<b>7</b>
4.1	Model description.....	8
4.1.1	Highest thermal zone calculation.....	9
4.1.2	Rationale .....	10
4.1.3	Initial state.....	11
4.1.4	Dynamic re-assignment of TZh (relevant only for TzD model).....	12
4.1.5	Dormant state .....	12
4.1.6	System Block Diagrams.....	13
4.2	PWM full speed policy .....	17
4.3	Dynamic minimal PWM speed policy .....	18
4.4	Thermal zone configuration .....	22
4.5	Power supply units PWM control policy .....	23
4.6	CPU thermal control policy .....	23
4.7	References .....	25

Figure 1 Algorithm Diagram .....	6
Figure 2 Kernel architectural view.....	14
Figure 3 Thermal Monitoring Block Diagram .....	<b>Error! Bookmark not defined.</b>
Figure 4 Thermal algorithm flow chart .....	17
Table 1 Terminology .....	7
Table 2 Thermal Area (TA) model with n thermal zones .....	8
Table 3 Thermal zones scores calculation.....	10
Table 4 Thermal zones initial state .....	11
Table 5 TZh transition from non TZD to non TZD .....	12
Table 6 TZh transition from TZD .....	12
Table 7 TZh transition to TZD.....	12
Table 8 Dormant state.....	13
Table 9 Dynamic PWM minimum speed table for system type 1 (Panther, Spider).....	19
Table 10 Dynamic PWM minimum speed table for system type 2 (Bulldog).....	20
Table 11 Dynamic PWM minimum speed table for system type 3 (Panther Streetfighter) ...	20
Table 12 Dynamic PWM minimum speed table for system type 4 (Boxer) .....	21
Table 13 Thermal zones configurations.....	22

# 1 Introduction

A method for performing switch or router system thermal control monitoring, such as that employed for complex system's configuration with a single FAN control and with multi points of temperature measure, is disclosed. The technique, referred as the Optimized Thermal Monitoring Algorithm (OTMA) approach, periodically re-calculates the highest temperature score of the all thermal zones and enforce thermal control to following the thermal zone with the highest score.

# 2 Current Methodology and Problems Being Addressed

The OTMA approach is targeted for the complex multiport any type network switches or routes required thermal management to improve reliability and prevent premature failure and save energy.

Such kind of systems are used to be equipped with big number of temperature sensors and limited or even single cooling device performing temperature monitoring and protection over the system. When multiple sensors are mapped to the same cooling device, the cooling device is should be set according the worst sensor from the sensors associated with this cooling device. The system shall implement cooling control based on thermal monitoring of the critical temperature sensors. The temperature control should keep the system in a desired operation temperature. Along with this requirement a thermal monitoring should not abuse the system resources and should be protected from the situation, when some of system temperature sensors requires to speed up cooling device, while some other requires speed down. Last case could lead to undesirable system thermal behavior, redundant noise and after all such system could get to malfunction behavior due to bad thermal monitoring.

There is now open source solution addressing the thermal monitoring for such class of systems and there are no any commercial software for that.

### 3 Solution summary

Currently there is no solution for the third-party OS.

The general purpose of the solution is to provide optimal thermal monitoring for Mellanox switches running third party OS, independently of the silicon type (Spectrum-2, Spectrum, SX or IB Switch), and the number of ports system equipped with.

The presented solution provides common solution for the thermal monitoring on Mellanox systems for Linux based switch operating systems. Algorithm will incorporated with existing Linux thermal control and will enhance it to allow more advance and accurate thermal control in Mellanox switches.

Existing Linux kernel algorithm does not support multiple sensors on same device. Each device is map to a thermal zone, however each thermal zone can affect fan rotation speed. This will result in collision between different thermal zones about how fast to spin the fans.

The presented approach for performing thermal control monitoring for systems with a single cooling system and multiple temperature sensors:

- introduces the definition of the highest thermal zone. highest thermal zone will be the zone that is currently most critical to the systems.
- provides the flexible approach, based on introducing of the "highest thermal zone" definition, which allows easy extension of the thermal monitoring, for example in the cases, when the additional thermal factors are to be considered (for example, next generation systems with Gearbox).
- allows to Decrease the amount of the context switching within the thermal monitoring. Highest thermal zone calculation is provided through the kernel space thread and user space should be involved only in case of the highest thermal zone re-assignment.

#### 3.1 Thermal monitoring architectural view

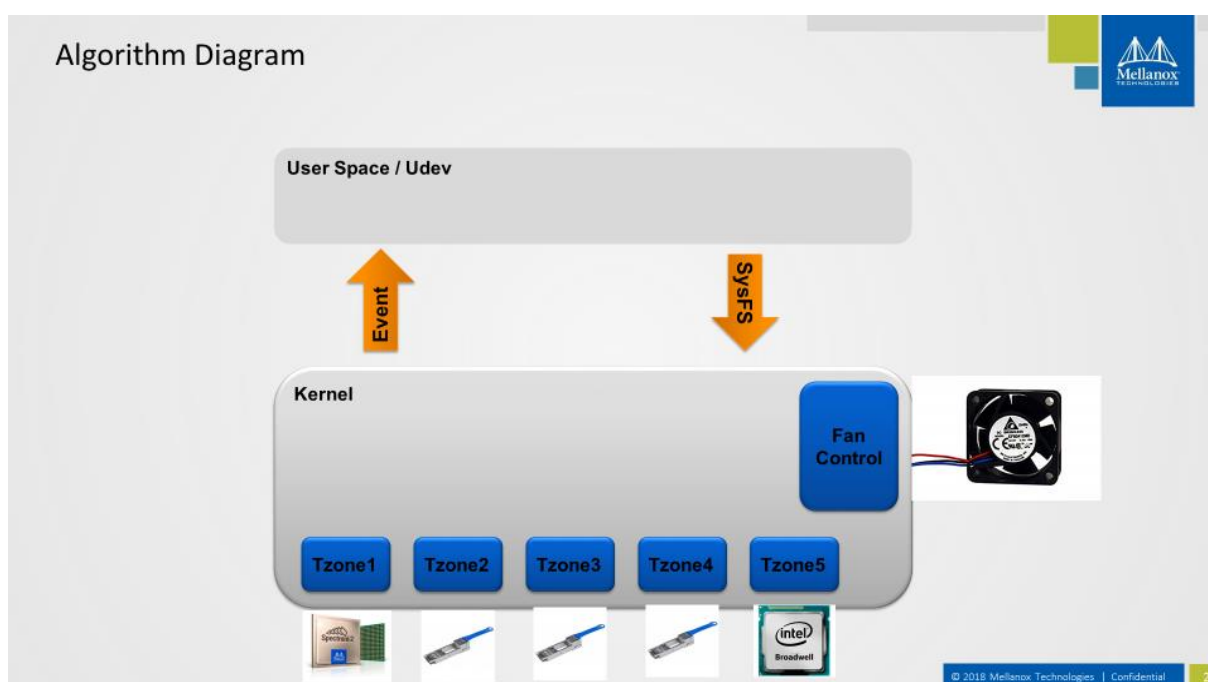


Figure 1 Algorithm Diagram

## 4 Detailed Explanation

Table 1 Terminology

Definitions	Description
OTMA	Optimized Thermal Monitoring Algorithm.
TA	Thermal Area - contains the number of the thermal zones (TZ<i>)</i> shared single cooling device, bound to all these zones.
TCG	Thermal Control Governor – user space application responsible for the thermal zones polling for finding one with the highest temperature score.
TZ<i></i>	Thermal Zone <i></i> - node containing thermal measurement sensor, definition of the temperature trip points and cooling device binding.
TZD	Thermal Zone Designated - thermal zone assigned for periodical re-calculation of the highest temperature score. It's possible to have a few TZD within the same TA. In such case TZDx will be associated with TZ<xi> (i form 1 to n1) zones, TZDy with TZ<yj> (j from 1 to n2), etc.  There is also possibility to control thermal algorithm with not TZD. In such case user space application is to be responsible for the highest thermal zone calculation (TCG).
TZH	Thermal Zone Highest - thermal zone with the highest temperature score.
TZM	Thermal Zone Mode - working mode of a thermal zone, where mode could be enabled or disabled. When thermal zone is in disabled mode, it is not the subject of monitoring.
TZP	Thermal Zone Policy - thermal governor for a thermal zone, where governor is responsible for performing throttling of the cooling device (FAN) associated with a thermal zone. The allowed policies are:  step wise policy - set a cooling device according to the thermal trends (high temperature - faster cooling device, lower temperature - slower cooling device);  standby policy - which does not change cooling device setting, until a thermal zone is not assigned as TZH.
TZTno<i></i>	Thermal Zone Trip point normal temperature of TZ<i></i>. When a thermal zone temperature is below this value - the temperature is below the desired temperature range.

TZThi<i>	Thermal Zone Trip point high temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZNtno<i> - the temperature is within the desired temperature range.
TZTho<i>	Thermal Zone <i> Trip point hot temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZThi<i> - the temperature is above the desired range but and cooling device speed should be increased.
TZTcr<i>	Thermal Zone <i> Trip point critical temperature of TZ<i>. When a thermal zone temperature is below this value and is above of TZTho<i> - the cooling device should set to the maximum speed. When a thermal zone temperature is below this value - it might cause system damage and thermal shutdown should be initiated.

## 4.1 Model description

Below is the TA model. It contains n thermal zones (where a thermal zone could be associated with any system component, like CPU, Network processor, memory, ports, PS units or just with some system critical points from thermal perspective).

TZD assignment is pre-defined within the TA and this role can't be changed on the fly. It could be a few TZD within the same TA. Any of TZ<i> can be set as TZD, but preferably to pick a thermal zone, which could not be removed. For example, CPU or network processor are good for such assignment, while transceiver or replaceable units are not so good for a such role. TZD is always enabled, since it's responsible for TZN re-calculation.

Only one thermal zone can be set at a given time as TZN. The cooling device manipulation is performed according to TZN state only (step wise policy), while for all others thermal zones the standby policy is applied (which means these thermal zones don't affect cooling device).

Same thermal zone could be TZD and TZN at the same time.

**Table 2 Thermal Area (TA) model with n thermal zones with TZD**

Name	TZNtno/TZNthi/ TZNtho/TZNtcr	TZD	TZN	TZP	TZM
TZ<1>	Variable	No	No	Standby	Disable
TZ<i>	Variable	Yes	No	Standby	Enable
TZ<k>	Variable	No	Yes	Step wise	Enable



TZ<n>	Variable	No	No	Standby	Disable
-------	----------	----	----	---------	---------

Such approach allows to reduce CPU utilization, since thermal control works over the maximum two thermal zones (TZD and TZH), while it services much more number of thermal zones.

It also helps to avoid competing between the thermal zones for the cooling device control and helps to avoid collisions between thermal zones, when one of them could require increasing of the cooling device speed, while another one could require its' reducing.

In case of alternative method, when highest thermal zone in TA is assigned by TCG the model look like in the below table.

**Table 3 Thermal Area (TA) model with n thermal zones with TCG**

Name	TZTno/TZThi/ TZTho/TZTcr	TZH	TZP	TZM
TZ<1>	Variable	No	Standby	Disable
TZ<i>	Variable	No	Standby	Disable
TZ<k>	Variable	Yes	Step wise	Enable
TZ<n>	Variable	No	Standby	Disable

#### 4.1.1 Highest thermal zone calculation

TZH is represented by 32 bits unsigned integer and calculated according to the next formula:

For  $T < TZ_{\langle t \rangle \langle i \rangle}$ , where  $t$  from  $\{no = 0, hi = 1, ho = 2, cr = 3\}$ :

$$TZ_{\langle i \rangle} \text{ score} = (T + (TZ_{\langle t \rangle \langle i \rangle} - T) / 2) / (TZ_{\langle t \rangle \langle i \rangle} - T) * 256^{**j};$$

It divides positive dividend  $T$  (current  $TZ_{\langle i \rangle}$  temperature read from thermal sensor) by positive divisor  $(t \text{ trip temperature minus } TZ_{\langle t \rangle \langle i \rangle} - T)$  rounds the result to closest integer and multiply it by 256 in power  $j$ .

Following this formula, if  $TZ_{<i>}$  is in trip point higher than  $TZ_{<k>}$ , the higher score is to be always assigned to  $TZ_{<i>}$ , independently of the real temperature.

If  $TZ_{<i>}$  and  $TZ_{<k>}$  are in the same trip point  $<t>$ , the higher score is to be assigned to the TZ, which is "closer" to its related trip point  $<t + 1>$  - respectively to  $TZ_{<t + 1> <i>}$  and  $TZ_{<t + 1> <k>}$ .

For  $T > TZ_{cr <i>}$  system thermal shutdown is to be performed.

$TZH \text{ score} = \text{MAX}(TZ_{<i>} \text{score});$

And TZ with the maximum score is to be assigned as TZH.

#### 4.1.2 Rationale

Consider the below trip points, temperatures and scores for a few thermal zones.

**Table 4 Thermal zones scores calculation**

Name	$TZT_{<no>}$	$TZT_{<hi>}$	$TZT_{<ho>}$	$TZT_{<cr>}$	Temperature	Score	Position
$TZ_{<i>}$ below normal trip	75	85	105	110	51	0x00000002	5
$TZ_{<j>}$ below normal trip	60	70	80	90	59	0x0000003C	3
$TZ_{<k>}$ below normal trip	66	76	88	98	63	0x00000010	4
$TZ_{<l>}$ above hot trip	60	70	80	90	82	0x000A0000	1
$TZ_{<m>}$ above high trip	60	70	80	90	62	0x00000900	2

According to the provided calculation method, the score for the thermal zone is assigned depending on thermal zone trip point location within the below ranges:

$Tx \leq TZ_{<x>no}$        $0x00000000 \leq Tx \text{ Score} \leq 0x000000FF$

$TZ_{<x>no} < Tx \leq TZ_{<x>hi}$        $0x000000FF < Tx \text{ Score} \leq 0x0000FF00$

$TZ_{<x>hi} < T_x \leq TZ_{<x>ho}$	$0x0000FF00 < T_x \text{ Score} \leq 0x00FF0000$
$TZ_{<x>ho} < T_x \leq TZ_{<x>cr}$	$0x00FF0000 \leq T_x \text{ Score} \leq 0xFF000000$
$T_x > TZ_{<x>cr}$ shutdown	$T_x \text{ Score}$ is $0xFFFFFFFF$ - subject for the system thermal

For two thermal zones located at the same kind of trip point, the higher score will be assigned to the zone, which closer to the next trip point, where distance is calculated as normalized value, between thermal zone real temperature and next trip point temperature.

Thus, the highest score will always be assigned objectively to the hottest thermal zone and the control of the cooling device will always be stuck to the thermal zone with the highest score.

#### 4.1.3 Initial state

At initial state TZD is always set as enabled with active step wise policy, while all others thermal zones are set as disabled and with standby policy. TZD is set according to the system configuration, for example to  $TZ_{<0>}$ . The thermal zone assigned as TZD will keep this role while the system is running.

**Table 5 Thermal zones initial state with TZD**

Name	Initial mode	Initial policy
TZD ( $TZ_{<0>}$ )	Enable	Step wise
$TZ_{<i>}$ (i from 1 to n)	Disable	Standby

For TCG method the initial state looks similar, but with no TZD.

**Table 6 Thermal zones initial state with TCG**

Name	Initial mode	Initial policy
$TZ_{<0>}$	Enable	Step wise
$TZ_{<i>}$ (i from 1 to n)	Disable	Standby

#### 4.1.4 Dynamic re-assignment of TZh (relevant only for TzD model)

When TzD during highest score re-calculation detects new TZh (for example Tz<y> should be assigned as TZh instead of Tz<x>), the next transition of mode and policy is performed:

**Table 7 TZh transition from non TzD to non TzD**

Name	Current mode	Current policy	Next mode	Next policy
Tz<x>	Enable	Step wise	Disable	Standby
Tz<y>	Disable	Standby	Enable	Step wise

**Table 8 TZh transition from TzD**

Name	Current mode	Current policy	Next mode	Next policy
Tz<x>	Enable	Step wise	Enable	Standby
Tz<y>	Disable	Standby	Enable	Step wise

**Table 9 TZh transition to TzD**

Name	Current mode	Current policy	Next mode	Next policy
Tz<x>	Enable	Step wise	Disable	Standby
Tz<y>	Enable	Standby	Enable	Step wise

#### 4.1.5 Dormant state

The dormant state means that cooling device speed is set to some constant value. System can get to the dormant state according to some system criteria. For example, it could be requirement to set cooling device to maximum speed in case some system unit, for example power supply, is absent, or some of tachometers is faulty. In such case the system will be in this state until power unit is not inserted back or until tachometer fault is not cleared. On exit from the dormant state system is always moved to the initial state.

**Table 10 Dormant state**

Name	Dormant mode	Dormant policy
TZ<i> (i from 0 to n)	Disable	Standby

#### **4.1.6 System Block Diagrams**

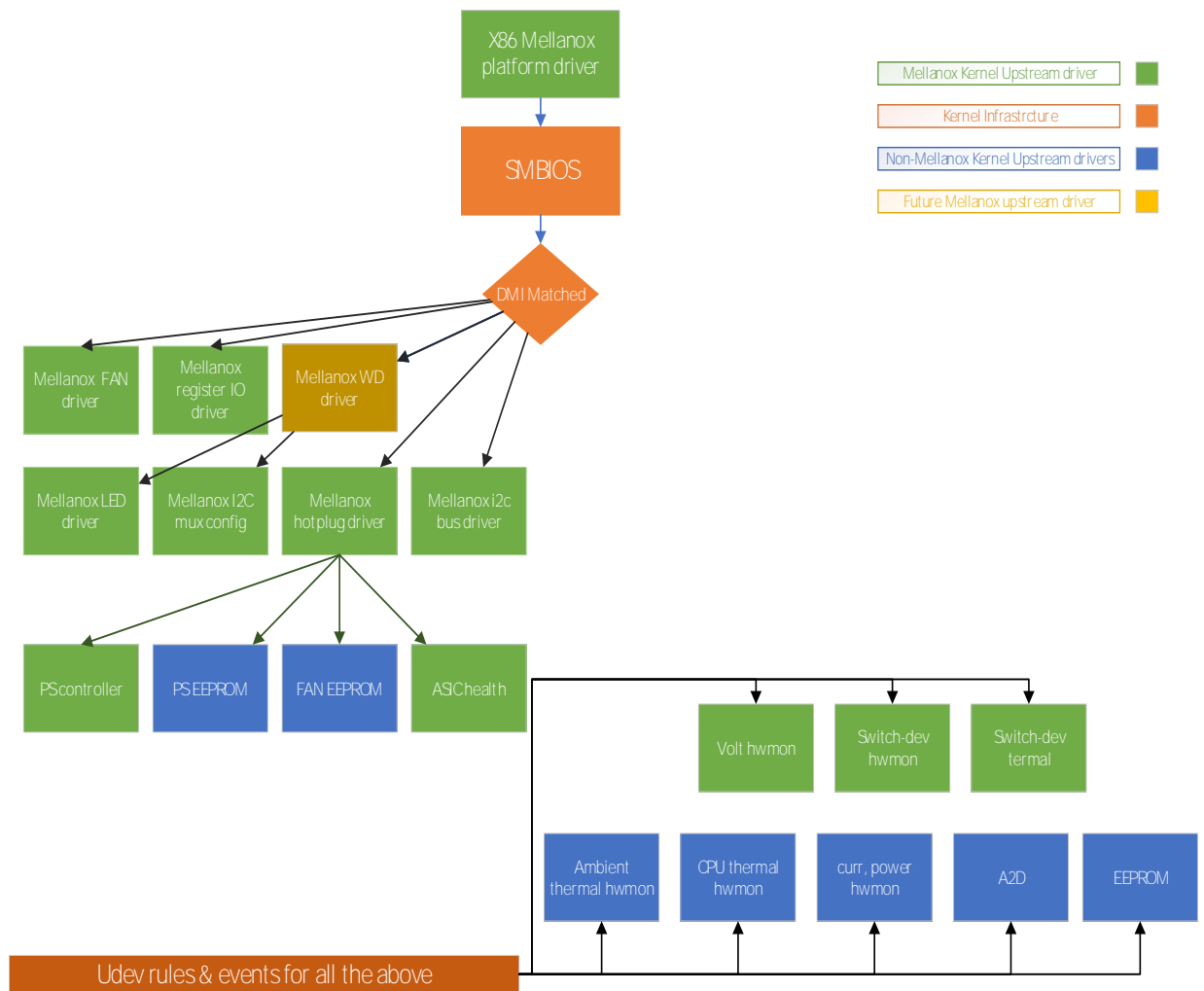


Figure 2 Kernel architectural view

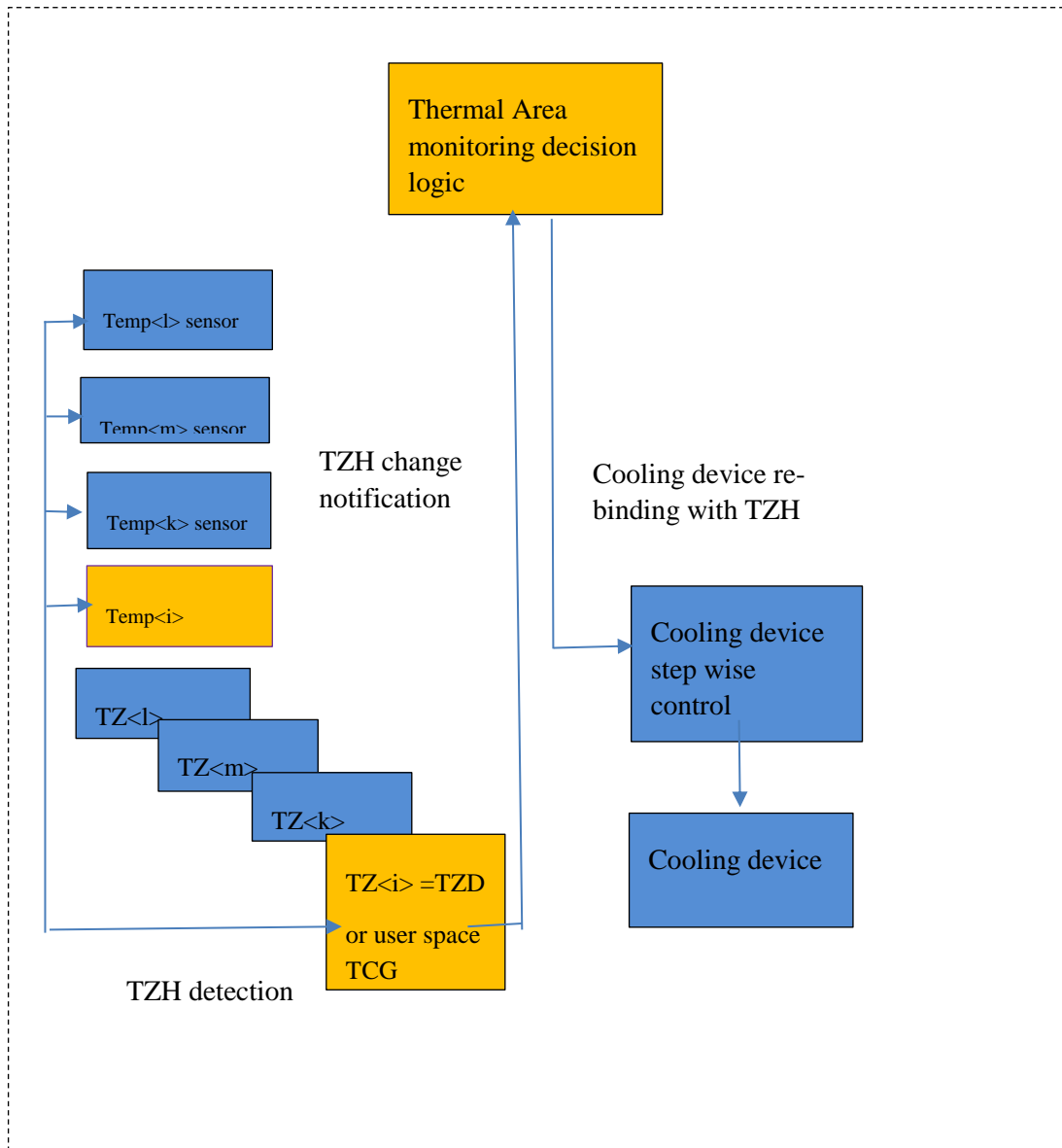


Figure 3 Thermal Monitoring Block Diagram

```
# Thermal zone score calculation normalized according to the trip point temperature.
# Parameters:
# $1 - the current thermal zone temperature;
# $2 - the closet trip temperature above the current temprature.
tz_score_calculate()
{
    delta=`expr '(' '$2 - $1' ')' / 2`
    score=`expr $delta + $shift`
}

# Thermal score calculation for ASIC thermal zone.
get_tz_asic_score()
{
    delta=0
    shift=1
    temp_curr=`cat $tz_temp`
    v1=`echo $temp_curr`
    v1=$((v1/1000))
    for ((t=0; t<$trip_points_num; t++)); do
        v2=`echo ${tz_asic_trips[t]}`
        if [ "$v1" -lt "$v2" ]; then
            tz_score_calculate $v1 $v2
            max_score=$score
            break
        fi
    done
    shift=$((shift*256))
}

# Thermal score calculation for modules' thermal zones
get_tz_module_score()
{
    delta=0
    shift=1
    temp_curr=`cat $thermal_path/mlxsw-module"$1"/thermal_zone_temp`
    v1=`echo $temp_curr`
    v1=$((v1/1000))
    for ((t=0; t<$trip_points_num; t++)); do
        v2=`echo ${tz_module_trips[t]}`
        if [ "$v1" -lt "$v2" ]; then
            tz_score_calculate $v1 $v2
            break
        fi
    done
    shift=$((shift*256))
}

# HTG (Highest Thermal Zone) detection.
# Re-assignment of TZM (Thermal Zone Mode) and TZP (Thermal Zone Policy) in case new thermal zone
# reaches the highest score. In such case previously hottest thermal zone mode and policy are set
# respectively to "disabled" and "user_space", while new hottest thermal zone to "enabled" and "step_wise",
get_tz_highest()
{
    score=0
    max_tz=0
    max_score=`cat $thermal_path/highest_score`
    get_tz_asic_score
    for ((p=1; p<=$max_ports; p+=1)); do
        if [ -L $thermal_path/mlxsw-module"$p"/thermal_zone_temp ]; then
            get_tz_module_score $p
            if [ "$score" -gt "$max_score" ]; then
                max_score=$score
                max_tz=$p
                echo $max_score > $thermal_path/highest_score
            fi
        fi
    done

    highest_tz_num=`cat $thermal_path/highest_tz_num`
    if [ "$max_tz" != "$highest_tz_num" ]; then
        if [ -L $thermal_path/highest_thermal_zone ]; then
            tzname=`basename "$(readlink -f $thermal_path/highest_thermal_zone)"`
            echo user_space > $thermal_path/highest_thermal_zone/thermal_zone_policy
            echo disabled > $thermal_path/highest_thermal_zone/thermal_zone_mode
            log_action_msg "Thermal zone $tzname: mode disabled, policy user_space"
            unlink $thermal_path/highest_thermal_zone
        fi
        if [ "$max_tz" != "0" ]; then
            ln -sf $thermal_path/mlxsw-module"$max_tz" $thermal_path/highest_thermal_zone
        else
            ln -sf $thermal_path/mlxsw $thermal_path/highest_thermal_zone
        fi
        echo $max_tz > $thermal_path/highest_tz_num
        echo $max_score > $thermal_path/highest_score
        echo step_wise > $thermal_path/highest_thermal_zone/thermal_zone_policy
        echo enabled > $thermal_path/highest_thermal_zone/thermal_zone_mode
        tzname=`basename "$(readlink -f $thermal_path/highest_thermal_zone)"`
        highest_tz=$tzname
        log_action_msg "Thermal zone $highest_tz: mode enabled, policy step_wise"
    fi
}
}
```

Figure 4 Highest temperature thermal zone calculation and assignment by TCG



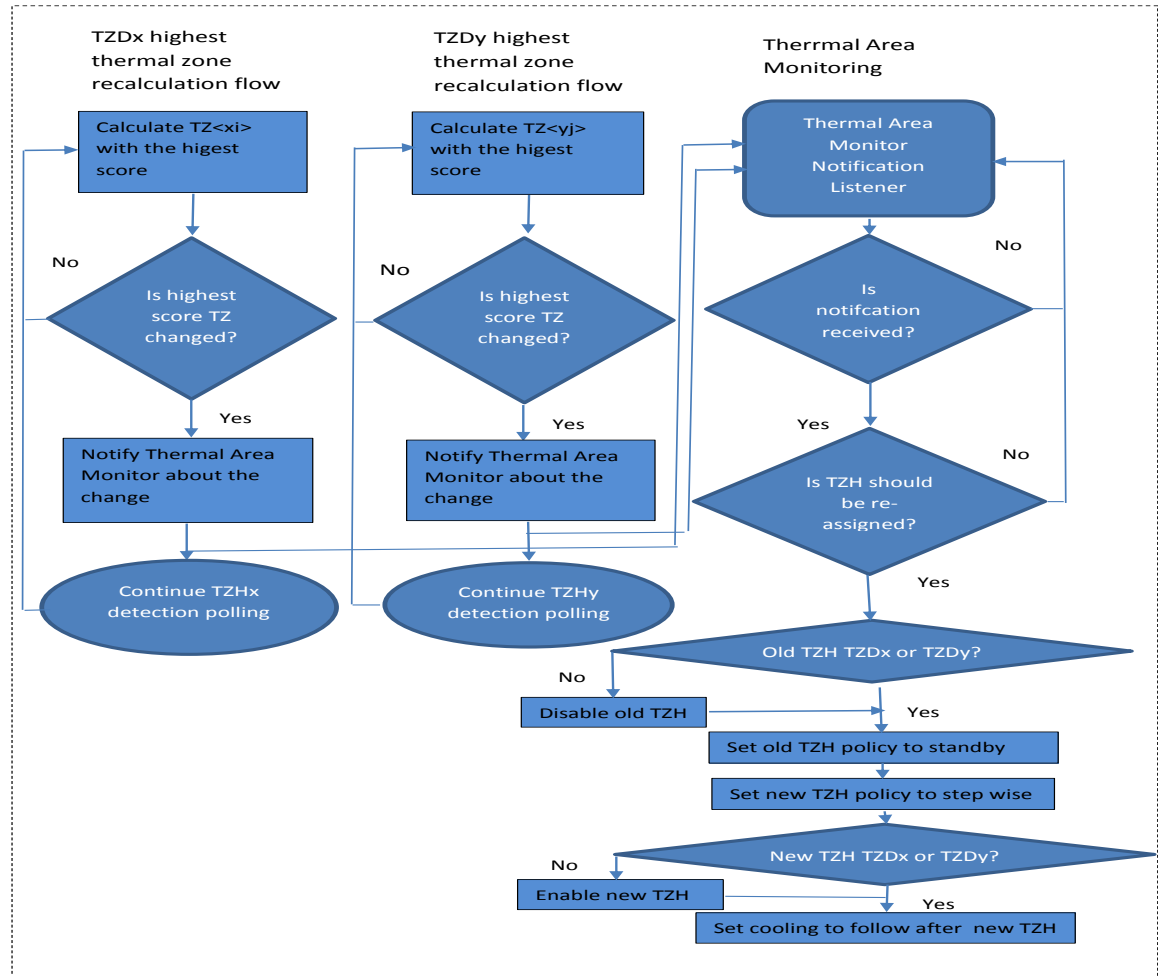


Figure 5 Thermal algorithm flow chart

## 4.2 PWM full speed policy

PWM full speed policy addresses the following features:

- Setting PWM to full speed if one of PS units is not present (in such case thermal monitoring in kernel is set to disabled state until the problem is not recovered). Such events will be reported to systemd journaling system.
- Setting PWM to full speed if one of FAN drawers is not present or one of tachometers is broken (in such case thermal monitoring in kernel is set to disabled

state until the problem is not recovered). Such events will be reported to systemd journaling system.

In the above cases all thermal zones are moved to the dormant state and PWM is to be at a full speed until all the states (absent PS unit or fan drawer, faulty fan) caused PWM full speed setting are not cleared. After all of them are cleared thermal zones are moved to initial state.

### 4.3 Dynamic minimal PWM speed policy

The dynamic setting depends on fan direction and cable type. For system with copper cables only or/and with trusted optic cable minimum PWM setting could be decreased according to the system definition. Such events will be reported to systemd journaling system.

There are per system type dynamic minimum tables, which define default minimum FAN. The cooling device bound to the thermal zones operates over the ten cooling logical levels. The default vector for the cooling levels is defined with the next PWM per level speeds:

20%	20%	30%	40%	50%	60%	70%	80%	90%	100%
-----	-----	-----	-----	-----	-----	-----	-----	-----	------

In case system dynamical minimum is changed for example from 20% to 60%, the cooling level vector will be dynamically updated as below:

60%	60%	60%	60%	60%	60%	70%	80%	90%	100%
-----	-----	-----	-----	-----	-----	-----	-----	-----	------

In such way the allowed PWM minimum is limited according to the system thermal requirements.

Per system thermal tables for the minimum PWM setting contain entries with the ambient temperature threshold values and relevant minimum speed setting. All Mellanox system are equipped with two ambient sensors: port side ambient sensor and fan side ambient sensor. The fan direction can be read from its EEPROM data, in case it is equipped with EEPROM device. If it doesn't equip with EEPROM, the direction is read from CPLD fan direction register. Or for the common case it can be calculated according to the next rule:

- if port side ambient sensor value is greater than fan side ambient sensor value - the direction is power to cable (forward);
- if port side ambient sensor value is less than fan side ambient sensor value - the direction is cable to power (reversed);
- if these values are equal: the direction is unknown.

For each system the following six vectors are defined:

- p2c\_dir\_trust\_tx: all cables with trusted sensors or with no sensors, FAN direction is power to cable (forward).
- p2c\_dir\_untrust\_tx: some cable sensor is untrusted; FAN direction is power to cable (forward).
- c2p\_dir\_trust\_tx: all cables with trusted sensors or with no sensors, FAN direction is cable to power (reversed).
- c2p\_dir\_untrust\_tx: some cable sensor is untrusted; FAN direction is cable to power (reversed).
- unk\_dir\_trust\_tx: all cables with trusted sensors or with no sensors, FAN direction is unknown.
- unk\_dir\_untrust\_tx: some cable sensor is untrusted; FAN direction is unknown.

Below is an example dynamic PWM setting for the different systems type.

**Table 11 Dynamic PWM minimum speed table for system type 1 (Panther, Spider)**

	PWM minimum [%]					
Direction	P2C		C2P		Unknown	
Cable reliability	trusted	untrusted	trusted	untrusted	trusted	untrusted
Ambient [C]						
<0	30	30	30	30	30	30
0 - 5	30	30	30	30	30	30
5 - 10	30	30	30	30	30	30
10 - 15	30	30	30	30	30	30
15 -20	30	30	30	30	30	30
20 - 25	30	30	40	40	40	40
25 -30	30	40	50	50	50	50

30 - 35	30	50	60	60	60	60
35 - 40	30	60	60	60	60	60
40- 45	50	60	60	60	60	60

**Table 12 Dynamic PWM minimum speed table for system type 2 (Bulldog)**

	PWM minimum [%]					
Direction	P2C		C2P		Unknown	
Cable reliability	trusted	untrusted	trusted	untrusted	trusted	untrusted
Ambient [C]						
<0	20	20	20	20	20	20
0 - 5	20	20	20	20	20	20
5 - 10	20	20	20	20	20	20
10 - 15	20	20	20	20	20	20
15 -20	20	30	20	20	20	30
20 - 25	20	30	20	20	20	30
25 -30	20	40	20	20	20	40
30 - 35	20	50	20	20	20	50
35 - 40	20	60	20	20	20	60
40- 45	20	60	30	30	30	60

**Table 13 Dynamic PWM minimum speed table for system type 3 (Panther Streetfighter)**

	PWM minimum [%]
--	-----------------

Direction	P2C		C2P		Unknown	
Cable reliability	trusted	untrusted	trusted	untrusted	trusted	untrusted
Ambient [C]						
<0	30	30	30	30	30	30
0 - 5	30	30	30	30	30	30
5 - 10	30	30	30	30	30	30
10 - 15	30	30	30	30	30	30
15 -20	30	30	30	40	30	40
20 - 25	30	30	30	40	30	40
25 -30	30	30	30	40	30	40
30 - 35	30	30	30	50	30	50
35 - 40	30	40	30	70	30	70
40- 45	30	50	30	70	30	70

**Table 14 Dynamic PWM minimum speed table for system type 4 (Boxer)**

	PWM minimum [%]					
Direction	P2C		C2P		Unknown	
Cable reliability	trusted	untrusted	trusted	untrusted	trusted	untrusted
Ambient [C]						
<0	20	20	20	20	20	20

0 - 5	20	20	20	20	20	20
5 - 10	20	20	20	20	20	20
10 - 15	20	20	20	20	20	20
15 - 20	20	30	20	20	20	30
20 - 25	20	40	20	30	20	40
25 - 30	20	40	20	40	20	40
30 - 35	20	50	20	50	20	50
35 - 40	20	60	20	60	20	60
40 - 45	20	60	20	60	20	60

## 4.4 Thermal zone configuration

Each thermal zone is configured with the thermal ranges, defined in the below table. Each zone, excepted last one, is configured with 5 Celsius hysteresis trips ( $t_h = 5C$ ). For ASIC thermal zone these ranges are set to the pre-defined value. For the QSFP modules thermal zones, these ranges are set according to the specific module warning and critical temperature thresholds:  $t_w$  – warning temperature threshold and  $t_c$  – critical temperature threshold.

**Table 15 Thermal zones configurations**

Thermal zone state	ASIC thermal zone	Module thermal zone	Thermal action
Cold	$t < 75\text{ C}$	$t < (t_w - 2 * t_h)$	Do nothing, keep PWM at minimal speed
Normal	$75\text{ C} \leq t < 85\text{ C}$	$(t_w - 2 * t_h) < t < t_w$	Perform hot algorithm
High	$85\text{ C} \leq t < 105\text{ C}$	$t_w \leq t < t_c$	Set PWM at full speed

Hot	$105\text{ C} \leq t < 110\text{ C}$	$t_c \leq t < (t_c + 2 * t_h)$	Keep PWM at full speed and produce warning message
Critical	$t \geq 110\text{ C}$	$t \geq (t_c + 2 * t_h)$	System thermal shutdown

## 4.5 Power supply units PWM control policy

If system's power supplies are equipped with the controlled cooling device, its cooling setting should follow the next rules

- Power supplies cooling devices should be set to the default value (usually 60% of PWM speed), defined per each system type;
- In case system's main cooling device is set above this default value, power supply's cooling device setting should follow main cooling device (for example if main cooling device is set to 80%, power supplies cooling devices should be set to 80%);
- In case system's main cooling device is set down (to x%), power supply's cooling devices should be set down, in case  $x\% \geq$  power supplies default speed value.

## 4.6 CPU thermal control policy

To be defined.

```
# Thermal monitoring loop with the relevant policies.
while true
do
    /bin/sleep $polling_time
    # Check if thermal is suspended. Continue loop from here if it is.
    # Validate thermal configuration.
    validate_thermal_configuration
    # Update PS unit fan speed
    update_psu_fan_speed
    # If one of PS units is out disable thermal zone and set PWM to the
    # maximum speed.
    get_psu_presence
    if [ $pwm_required_act -eq $pwm_max ]; then
        init_tz_highest
        continue
    fi
    # If one of tachometers is faulty disable thermal zone and set PWM
    # to the maximum speed.
    get_fan_faults
    if [ $pwm_required_act -eq $pwm_max ]; then
        init_tz_highest
        continue
    fi
    # Set dynamic FAN speed minimum, depending on ambient temperature,
    # presence of untrusted optical cables or presence of any cables
    # with untrusted temperature sensing.
    set_pwm_min_speed
    # Update cooling levels of FAN If dynamic minimum has been changed
    # since the last time.
    if [ $fan_dynamic_min -ne $fan_dynamic_min_last ]; then
        echo $fan_dynamic_min > $cooling_cur_state
        log_action_msg "FAN minimum speed is changed from $fan_from to $fan_to percent"
        fan_dynamic_min_last=$fan_dynamic_min
        echo $fan_to > $thermal_path/fan_dynamic_min
    fi
    # Enable ASIC thermal zone if it has been disabled before and it's
    # currently has highest score.
    mode=`cat $tz_mode`
    highest_tz_num=`cat $thermal_path/highest_tz_num`
    if [ $mode = "disabled" ] && [ $highest_tz_num = "0" ]; then
        echo enabled > $tz_mode
        log_action_msg "ASIC thermal zone is re-enabled"
        # System health (PS units or FANs) has been recovered. Set PWM
        # speed to dynamic speed minimum value and give to kernel
        # thermal algorithm can stabilize PWM speed if necessary.
        check_trip_min_vs_current_temp 2
    fi
    count=$((count+1))
    # Find highest theraml zone and re-assign mode and policy in case new one is found.
    get_tz_highest
done
```

Figure 6 Thermal monitoring flow



## 4.7 References

Reference code with implementation can be found at <https://github.com/Mellanox/hw-mgmt>.

It contains set of thermal control scripts and kernel patch for kernel v.5.1.