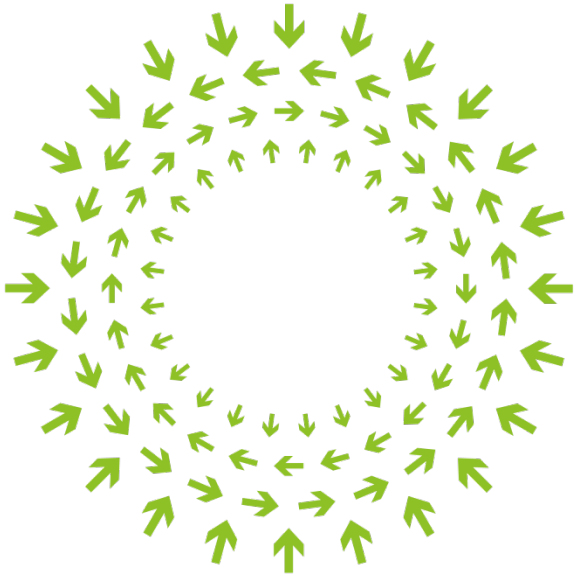

OPEN
Compute Project

Facebook Server
Fan Speed Control Interface

Draft Version 0.1

Author: Jacob Na, Thermal Engineer, Facebook

Contents

1 Overview

This document describes Facebook's FSC algorithm and its update methodology. Using the OpenIPMI Fan Speed Control is an intelligent method for controlling server fans to provide adequate cooling while managing thermal constraints and power efficiency. This document will help to manage FSC settings and FSC updates by using IPMI commands to vary the fan control profile on either local or remote systems.

1.1 License

2 Fan Speed Control Table Update Methodology

2.1 Introduction

Facebook wants to use the following IPMI commands to standardize how the server management controller, such as BMC, manages FSC parameters. The following information will explain the methodology of an FSC update by using open source IPMI commands. The FSC parameters should be writable and readable from the BMC, via IPMI commands. To avoid redesigning BMC firmware codes, the IPMI FSC functions should not be hard-coded, and should be flexible enough for Facebook to expand or improve the FSC algorithm in the future. Using IPMI commands to update the FSC can provide the following benefits:

- Easy implementation
- Openness
- Independency
- Flexible update control and FSC development

2.2 List of FSC IPMI Commands

There are four major FSC IPMI commands (Table 1). Detail explanations of each command are in Section 2.4

Table 1 IPMI FSC Commands

1. Enter/Exit FSC Update mode
2. Read/Write FSC Profile header
3. Read/Write FSC profiles
4. Read/Write Zone Control settings
5. Read/Write the pulse-width modulation (PWM) value and read the total number of PWM channels/IDs

2.3 Data Length

Please see the command tables for the data length. If not specified, data length should be a single byte.

2.4 Details of FSC IPMI Commands

Each IPMI command intended to interact with the FSC shall be communicated with the BMC by first sending a request message to the BMC and then reading the response message it sends back. OEM NetFcn should be “30h”.

2.4.1 Enter/Exit FSC Update Mode

In order to write FSC configuration data to the BMC, the user must enter into FSC update mode. In FSC update mode, the new FSC data is written to the temporary BMC memory space. In order to complete the FSC update, the user must exit the FSC update mode to upload the FSC data from the temporary BMC memory space to BMC flash and the runtime BMC memory space.

Reading FSC data does not require entering the FSC update mode. Either a byte of completion code or the BMC response message must be shown when a command is run. Request codes and response codes are listed in Table 2.

Table 2 Enter/Exit FSC Update Mode Table

Code	Command	Request, Response Data
61h	Enter FSC Update mode	**Request:**
		None
		Response:
		Byte 1 – Completion Code
		00h: Normal
		80h: Already in update mode
		CCh: Invalid data field
62h	Exit FSC Update mode	**Request:**
		Byte 1 – Exit FSC mode
		00h: Exit without saving
		01h: Exit with saving
		Response:
		Byte 1 – Completion Code
		00h: Normal
		D5h: Command not supported in present state
		D6h: Cannot execute command due to checksum mismatch

2.4.2 Read/Write FSC Profile Header

The FSC profile header contains the global information about the FSC as listed in Table 3. Table 4 shows the FSC IPMI commands. Section 4 contains detailed information about how to calculate the CRC-32 checksum in the FSC profile header. When the BMC is not ready, such as during the system boot-up, the PWM value in "Update mode PWM/Recovery mode PWM" must be used.

Table 3: FSC Profile Header

1. Total number of the FSC profiles
2. FSC control status: 0-inactive, 1-active
3. FSC version control: Major version [2 bytes], Minor version [2 bytes]
4. CRC-32 checksum value: [4 bytes]
5. Update mode PWM/Recovery mode PWM
6. PWM in boost mode, such as fan failure and sensor failure
7. Sensor sampling interval [ms]
8. Failure control
9. Manufacturer's ID
10. Development stage

Table 4 FSC IPMI Commands for Read/Write FSC Profile Header

Code	Command	Request, Response Data
63h	Read FSC profile header	**Request:** None
		Response: Byte 1 – Completion Code 00h: Normal C1h: Invalid command due to wrong state. D5h: Command not supported in present state.

		Byte 2 – Total number of FSC profiles Byte 3 – Control status 00: inactive 01: active Byte 4:5 – Major FSC version Byte 6:7 – Minor FSC version Byte 8:11 – CRC checksum Byte 12 – Update/Recovery mode PWM Byte 13 – Boost mode PWM Byte 14 – Sensor sampling interval [ms] Byte 15 – Failure control [7-3] - Reserved [2] - Fan failure boost enabled; 1 = enable, 0 = disable [1] - Sensor failure boost enabled; 1 = enable, 0 = disable [0] - Intrusion boost enabled. 1 = enable, 0 = disable Byte 16:18 – Manufacturer's ID Byte 19 – Development stage 00: EVT 01: DVT 02: PVT 03: MP
64h	Write FSC Profile Header	**Request:**
		Byte 1 – Total number of FSC profiles. Byte 2 – Control status 00: inactive 01: active Byte 3:4 – Major FSC version Byte 5:6 – Minor FSC version Byte 7:10 – CRC checksum Byte 11 – Update/Recovery mode PWM Byte 12 – Boost mode PWM Byte 13 – Sensor sampling interval [ms] Byte 14 – Failure control [7-3] - Reserved [2] - Fan failure boost enabled; 1 = enable, 0 = disable [1] - Sensor failure boost enabled; 1 = enable, 0 = disable [0] - Intrusion boost enabled. 1 = enable, 0 = disable Byte 15:17 – Manufacturer's ID Byte 18 – Development stage 00: EVT 01: DVT 02: PVT 03: MP
		Response:
		Byte 1 – Completion code 00h: Normal C1h: Invalid command due to wrong state C9h: Invalid data length D5h: Command not supported in present state

2.4.3 Read/Write FSC Profile

The FSC profile contains the data listed in Table 5. The Read FSC Profile command can read the data in the FSC profile. The Write FSC Profile command can create the inventory of different profiles. The profiles can be linked to the Sensor ID in the Zone Control Setting to apply for the FSC. The FSC profiles are identified by sequential profile ID. Each profile has an associated profile type, such as linear, non-linear and etc. The pre-defined profile types are listed in Table 6. The linear FSC type can be the table-driven fan control, which uses the piecewise linear function[1]. The non-linear FSC type can be the PID[5]-driven fan control. More description about the predefined data types is in Table 6.

The data sets in a linear-type profile are combinations of input and output data. Input and output data pairs store a one-byte sensor input first, followed by a one-byte output value. When the sensor value reports a value higher than the maximum value in the data sets, the highest output value in the data sets is used. When the sensor value reports a value lower than the minimum value in the data sets, the lowest output value in the data set is used. When the sensor value is between two data points, the output value should be calculated by the curve fit or the linear interpolation[2].

The unit of input data and unit of output data must follow the Sensor Unit Type Codes in table 43-15 in IPMI v2.0 specification4. The additional unit type code would be “code 91: %”

Table 5 Lists of Data in an FSC Profile

1. Profile ID
2. Profile Type: 0-linear, 1-nonlinear, 2-others-reserved
3. Hysteresis
4. Unit of data input
5. Unit of data output
6. Profile data length
7. Profile data sets

Table 6 Pre-defined Data Set

Type	Data Format Example		Description
	Input	Output	
Linear	Temperature	PWM [%]	Up to 64 pairs, ordered from low to high. Temperature is in Celsius.
	FAN RPM	CFM	Up to 64 pairs, ordered from low to high. FAN RPM corresponds to PWM [%] in 5% increment.
	Power (W)	PWM [%]	Up to 64 pairs, ordered from low to high.
Non-Linear	Control algorithm values [Kp, Ki, Kd, set point]	PWM [%]	Control algorithm constants and set point Kp = (1/1000 %PWM/C)- 4 bytes Ki, Kd = (1/1000 %PWM/C/Sensor sampling interval) – 4 bytes Setpoint – desired value

Table 7 Lists of Commands for Read/Write FSC Profile

Code	Command	Request, Response Data
65h	Read FSC Profile	**Request:**
		Byte 1 – Data source
		00: BMC flash
		01: temporary BMC memory space
		Byte 2 – Profile ID
		Response:
		Byte 1 – Completion code 00h: Normal C1h: Invalid command due to wrong state Byte 2 – Profile ID Byte 3 – Profile type 00: Linear FSC 01: Non-linear FSC (such as PID) 02: Reserved Byte 4:5 – Hysteresis Byte 6 – Unit of input data Byte 7 – Unit of output data Byte 8 – Profile data length Byte 9:N – Data sets: LSB first and MSB last in field in given profile ID

66h	Write FSC Profile	**Request:** Byte 1 – Profile ID Byte 2 – Profile type 00: Linear FSC 01: Non-linear FSC (such as PID) 02: Reserved Byte 3:4 – Hysteresis Byte 5 – Unit of input data Byte 6 – Unit of output data Byte 7 – Profile data length Byte 8:N – Data sets: LSB first and MSB last in field in given Profile ID
		Response: Byte 1 – Completion code 00h: Normal C1h: Invalid command due to wrong state C9h: Invalid data type or data length specified D5h: Command not supported in present state

2.4.4 Read/Write Zone Control Settings

The Zone control setting manages the group of sensors per zone as well as the links between the sensor ID and the FSC profile ID. The lists of zone control settings are in Table 7. The fan ID can be used in the PWM ID field. The data set should be combinations of the sensor ID, sensor owner ID, and profile ID. The sensor ID must match to IPMI SDR, and the sensor type/unit must be match to the unit of input data in FSC profile.

Table 8 List of Data in the Zone Control Setting

1. PWM ID
2. Fan PWM ramp rate per sensor sampling interval
3. Data length
4. Data sets: sensor ID, sensor owner ID, profile type, profile ID

Table 9 Lists of Commands for Zone Control Profile

Code	Command	Request, Response Data
67h	Read Zone Control setting	**Request:**
		Byte 1 – Data Source
		00: BMC flash
		01: temporary BMC memory space
		Byte 2 – PWM ID
		Response:
		Byte 1 – Completion code 00h: Normal C9h: Invalid data type or data length specified C7h: Requested data extends beyond data length C5h: Wrong PWM ID Byte 2 – PWM ID Byte 3 – Fan PWM ramp rate per sensor sampling interval Byte 4 – Data length Byte 5:N – Data sets
68h	Write Zone Control setting	**Request:**
		Byte 1 – PWM ID Byte 2 – Fan PWM ramp rate per sensor sampling interval Byte 3 – Data length

		Byte 4:N – Data sets
		Response:
		Byte 1 – Completion code 00h: Normal C1h: Invalid command due to wrong state C9h: Invalid data type or data length specified D5h: Command not supported in present state C5h: Wrong PWM ID

2.4.5 Get/Set PWM Value and Get the Total Number of PWM IDs

The following can help to read and write the PWM value. None of commands in Table 10 should require entering the FSC update mode. The total number of PWM IDs should match the total number of active fans in the server.

Table 10 Lists of Commands for PWM Read/Write

Code	Command	Request, Response Data
69h	Get PWM value	**Request:**
		Byte 1 – PWM ID
		Response:
		Byte 1 – Completion code
		00h: Normal C5h: Wrong PWM ID
		Byte 2 – Current PWM reading
70h	Set PWM value	**Request:**
		Byte 1 – PWM ID
		Byte 2 – PWM value
		Response:
		Byte 1 – Completion code 00h: Normal C5h: Wrong PWM ID
71h	Get the total number of PWM IDs	**Request:**
		None
		Response: Byte 1 – Completion code 00h: Normal Byte 2 – Total number of PWM IDs

3 CRC-32 and Checksum

A BMC checksum calculation shall be triggered on the entire FSC data when “Exit with saving mode” is run. This checksum calculation must use CRC with a polynomial length of 33 bits (CRC-323). The calculated checksum must match the data provided in the checksum field of the FSC Profile Header. This ensures that the new FSC data will be updated in flash and the FSC module in RAM from the special reserved BMC RAM. If a mismatch is found between the calculated checksum and the checksum provided in the checksum field of the FSC Profile Header, the saving operation shall be terminated and the BMC should send a checksum error using an IPMI response message.

3.1 Sequence in CRC-32 Generation

The CRC-32 checksum must be computed for all data elements in FSC profiles and FSC Profile header. The CRC-32 calculation must use the following sequences to avoid any mismatch of the generated checksum against the checksum field stored in the FSC profile header.

1. FSC Profile header
    a. From Top to Bottom in Table 3, excluding the Checksum field
2. FSC profile
    a. From Top to bottom in profile # 0 including profile ID
    b. i) For linear: input_0, output_0 - > input_1, output_1->…->input_N, output_N
       ii) For non-linear: target point, Kp, Ki, Kd
    c. From Top to bottom in profile # 1 including profile ID
    d. Continue until end of profile ID#
3. Zone control setting
    a. From top to bottom in zone control setting 0 including PWM ID
    b. From top to bottom in zone control setting 1 including PWM ID
    c. Continue until end of PWM ID#

4 References

1. http://en.wikipedia.org/wiki/Piecewise_linear_function
2. http://en.wikipedia.org/wiki/Linear_interpolation
3. http://en.wikipedia.org/wiki/Cyclic_redundancy_check
4. http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-gen-interface-spec-v2-rev1-4.pdf ; Page: 529
5. http://en.wikipedia.org/wiki/PID_controller