



Device Specification

NVIDIA Enthusiast System Architecture

Mark A. Overby
Chief Client / Workstation Software Architect

9 July 2007
Revision 0.90a

Revision History:

| <u>Date</u> | <u>Revision</u> | <u>Description</u> | <u>Author</u> |
|---------------|-----------------|--|---------------|
| 11 March 2007 | 0.40 | Initial draft of document with substantially complete outline and some sections of text completed | MAO |
| 19 March 2007 | 0.41 | Completed Clause 8 text. | MAO |
| 20 March 2007 | 0.41a | Added Annex B tables for pin configurations | MAO |
| 22 March 2007 | 0.42 | Minor editorial language clean up Revised clause 5.3 to include all types of sensors and devices permitted to be attached Added definition for ESA device Clarified text around string descriptors and serial numbers Changed sensor type to bus type Clarified units on voltages and current Added missing chip identification field from external tach element (TBD on AVL for those parts) Added Annex C for recommended sensor parts Added stub clause to describe how to format and use the microcontroller | MAO |
| 23 March 2007 | 0.43 | Updated clause stubs for tool chain and programming Deleted references to configuration memory and configuration memory clause | MAO |
| 2 April 2007 | 0.44 | Added review comments from previous on-line reviews Added text for architecture clauses | MAO |
| 8 April 2007 | 0.45 | Added text for configuration table description clauses Accepted comments from previous reviews Added RGB LED controls section (TBD data) Added Voltage Regulator controls section (TBD data) | MAO |
| 10 April 2007 | 0.50 | Clean up for first public review | MAO |
| 14 May 2007 | 0.80 | Added RGB LED details Added cable detection elements Minor editorial changes | MAO |
| 31 May 2007 | 0.81 | Added hardware strapping field to table description element | MAO |
| 5 June 2007 | 0.82 | Revised hardware strap mechanism Revised RGB (external) layout | MAO |
| 29 June 2007 | 0.90 | Added RGB permitted pin clause Corrected error in RGB identification value and AVL text Added thermal curves to zone descriptor element Added permitted pins for SPI CS signals Removed AVL for external tachometer elements Added external AVL for RGB LED controllers Removed Annex B and D | MAO |
| 9 July 2007 | 0.90a | Added requirement for all sensors to be in a zone Corrected minor typographical errors Replaced logo Fixed broken x-ref | MAO |

Legal Notices

*** Need full text. I've noted what I know ***

SPI is a registered trademark of Freescale Semiconductor Incorporated.
Intel is a registered trademark of the Intel Corporation.

Table of Contents

| | | |
|-------|--|----|
| 1 | Scope..... | 7 |
| 1.1 | Description of Clauses..... | 7 |
| 2 | References | 8 |
| 2.1 | Normative References..... | 8 |
| 2.2 | Approved References..... | 8 |
| 2.3 | Other Documents and References | 8 |
| 3 | Definitions, Symbols, Abbreviations, and Conventions | 9 |
| 3.1 | Definitions..... | 9 |
| 3.2 | Symbols and Abbreviations | 9 |
| 3.3 | Keywords..... | 10 |
| 3.4 | Conventions..... | 10 |
| 3.4.1 | Overview..... | 10 |
| 3.4.2 | Numeric conventions | 11 |
| 3.4.3 | Bit and byte ordering..... | 11 |
| 3.4.4 | Use of and/or | 12 |
| 3.4.5 | Hardware Pin and Port Symbols..... | 12 |
| 4 | Architecture..... | 13 |
| 4.1 | System Architecture..... | 13 |
| 4.2 | Hardware Architecture | 15 |
| 4.3 | Software Architecture | 15 |
| 5 | Hardware Design Requirements..... | 16 |
| 5.1 | Microcontroller Selection | 16 |
| 5.2 | Microcontroller Configuration..... | 16 |
| 5.2.1 | Clock Source | 16 |
| 5.2.2 | Power Source | 16 |
| 5.3 | Microcontroller Attached Devices | 17 |
| 5.3.1 | Fault LED (Mandatory) | 17 |
| 5.3.2 | I2C and SPI Sensors or Controllers..... | 17 |
| 5.3.3 | Analog Sensors | 17 |
| 5.3.4 | Status Indicator LED's | 17 |
| 5.3.5 | RGB LED's | 17 |
| 5.3.6 | PWM Controlled Devices..... | 18 |
| 5.3.7 | Automatic Thermal Control..... | 18 |
| 5.4 | Microcontroller Permitted Pin Configurations..... | 18 |
| 5.4.1 | Fault LED Connection..... | 18 |
| 5.4.2 | SMBUS or I2C Device Connection | 18 |
| 5.4.3 | SPI Device Connection..... | 18 |
| 5.4.4 | Analog Device (ADC Measurement) Connection..... | 18 |
| 5.4.5 | PWM Outputs and Tachometer Inputs..... | 20 |
| 5.4.6 | RGB LED Configurations..... | 20 |
| 5.4.7 | Status Indicator LED Configurations..... | 20 |
| 5.5 | Multifunction Sensors | 20 |
| 6 | Programming and Configuring ESA Microcontrollers..... | 21 |
| 6.1 | Configuration Tools..... | 21 |
| 6.2 | Configuration Procedure..... | 21 |
| 7 | Configuration Table Format | 22 |
| 7.1 | Overview..... | 22 |
| 7.2 | Sensor Zones | 22 |
| 7.3 | Axial Sensor Position..... | 23 |
| 8 | Table Elements Reference | 24 |
| 8.1 | Reference Overview | 24 |

3. Definitions, Symbols, Abbreviations, and Conventions

| | | |
|---------|--|----|
| 8.2 | Table Description Element..... | 24 |
| 8.3 | Common Element Fields | 26 |
| 8.3.1 | Common Element Header | 26 |
| 8.3.2 | Serial Bus Address Encoding | 26 |
| 8.4 | Digital Thermal Sensor Element | 28 |
| 8.5 | Analog Thermal Sensor Element | 29 |
| 8.6 | Microcontroller PWM Fan Element | 30 |
| 8.7 | Status LED Element | 32 |
| 8.8 | Water Level – Analog Element | 33 |
| 8.9 | Microcontroller Pump PWM Element | 34 |
| 8.10 | Power Supply Voltage Measurement Element..... | 35 |
| 8.11 | Current Measurement Element..... | 36 |
| 8.12 | External PWM Controller Element | 37 |
| 8.12.1 | Speed Block Encoding | 38 |
| 8.12.2 | Register Pair Fields | 38 |
| 8.13 | External Tachometer Element | 39 |
| 8.14 | Microcontroller RGB LED Element | 40 |
| 8.15 | External RGB LED Element..... | 41 |
| 8.16 | String Element | 43 |
| 8.17 | Zone Descriptor Element | 44 |
| Annex A | Pin Configuration Tables (Informative) | 46 |
| Annex B | Element Types by Numeric Value (Informative) | 48 |

List of Figures

| | |
|--|----|
| Figure 1 - ESA System Environment..... | 14 |
| Figure 2 - ESA Hardware Block Diagram..... | 15 |
| Figure 3 - Example Hall Effect Current Sensing Block Diagram | 19 |
| Figure 4 - Example High-Side Current Sensing Block Diagram | 19 |

List of Tables

| | |
|--|----|
| Table 1 - ISO and American numbering conventions examples | 11 |
| Table 2 - Example ordering of bits and bytes within an element | 12 |
| Table 3 – Power Consumption Maximums for USB Powered Devices | 16 |
| Table 4 - Table Description Element..... | 24 |
| Table 5 - Specification Revision Coding Values..... | 24 |
| Table 6 - ESA Device Type..... | 25 |
| Table 7 - Common Element Header..... | 26 |
| Table 8 - Bus Type Encoding..... | 26 |
| Table 9 – Device Address Encoding for I2C and SMBUS..... | 27 |
| Table 10 – Device Address Encoding for SPI | 27 |
| Table 11 - Digital Thermal Sensor Element | 28 |
| Table 12 – Analog Thermal Sensor Element | 29 |
| Table 13 – Analog Thermal Sensor Type Coding | 30 |
| Table 14 – Microcontroller PWM Fan Element..... | 31 |
| Table 15 – Status LED Element..... | 32 |
| Table 16 – LED color coding..... | 33 |
| Table 17 – LED purpose coding..... | 33 |
| Table 18 – Analog Water Level Element..... | 33 |
| Table 19 – Power Supply Voltage Measurement Element..... | 35 |
| Table 20 – Power Supply Current Measurement Element | 36 |
| Table 21 – External PWM Controller Element..... | 37 |
| Table 22 – Speed Block | 38 |
| Table 23 – Register Pair Fields | 38 |
| Table 24 – External Tachometer Element..... | 39 |
| Table 25 – Read Status / Tachometer Initialization Block | 40 |
| Table 26 – Microcontroller RGB LED Element | 40 |
| Table 27 – External RGB LED Element | 41 |
| Table 28 – RGB LED Controller Identifier | 42 |
| Table 29 - String Element | 43 |
| Table 30 – Zone Description Element..... | 44 |

1 Scope

This document describes the software and hardware architectures for the NVIDIA Enthusiast System Architecture environment. This document also describes the components that 3rd party implementers provide to enable the architecture for a supported component.

The Enthusiast System Architecture is an environment that provides a common user interface, control, and status monitoring that provides an enhanced experience for enthusiast class devices (e.g., water cooling, computer cases).

This document uses ISO and ANSI conventions and format and the structure of this document was designed from templates created by those national and international standards bodies. No standards body endorsement, submission, or approval is inferred by the usage of these templates.

1.1 Description of Clauses

Clause 1 (Scope) contains the scope of this specification

Clause 2 (References) contains related standards, specifications, and other documents

Clause 3 (Definitions) contains the definitions, symbols, abbreviations, and conventions used in this document

Clause 4 (Architecture) describes the software and hardware architecture

Clause 5 (Hardware Design Requirements) describes the hardware implementation requirements

Clause 6 (Microcontroller Programming) describes the tool set and procedures use to load and configure firmware

Clause 6 (Configuration Table Format) describes the configuration table used to configure the microcontroller environment

Clause 7 (Table Entry Reference) describes the possible configuration table entries and associated requirements

Clause 8 (Example Implementation) contains an informative example implementation based on the requirements in other clauses

2 References

2.1 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. At time of publication, the editions indicated were valid. All standards and specifications are subject to revision. Implementers of this specification are encouraged to investigate using the latest editions of the standards and specifications indicated in this clause.

2.2 Approved References

ISO/IEC 10646-1:2000, *Information Technology – Universal Multiple-Octet Codec Character Set*

2.3 Other Documents and References

For information on the current states of the listed document(s), or regarding availability, contact the indicated organization or company.

C8051F320/1 Datasheet

The C8051F320/1 datasheet may be obtained from Silicon Laboratories Incorporated (<http://www.silabs.com>).

The I²C-Bus Specification Revision 2.1

The I2C bus specification may be obtained from NXP Semiconductors (<http://www.nxp.com>).

System Management Bus (SMBUS) Specification 2.0

The SMBUS specification may be obtained from the SBS Implementers Forum (<http://www.smbus.org>).

Unicode 3.0

The Unicode 3.0 document(s) may be obtained from the Unicode Consortium (<http://www.unicode.org>).

Universal Serial Bus 2.0

Universal Serial Bus Language Identifiers (LANGIDs) 1.0

The USB documents may be obtained from the USB Implementers Forum (<http://www.usb.org>).

3 Definitions, Symbols, Abbreviations, and Conventions

3.1 Definitions

- 3.3.1 Element:** An element is a structure embedded in a configuration table that describes a device, sensor, or description of a device or sensor that is attached to the ESA microcontroller.
- 3.3.2 ESA Device:** An implementation of a support class of peripheral device (e.g., power supply) that includes an ESA microcontroller supported by this specification, attached sensors and devices, and configuration memories.
- 3.3.3 Enthusiast System Architecture:**
- 3.3.4 Pulse Width Modulation:** Square wave duty cycle transmission of a signal resulting in the variation of the average value of the waveform.
- 3.3.5 Serial Peripheral Interface:** A three wire serial bus used for communications to devices external to the microcontroller. The data into and out of the microcontroller is transmitted across separate physical pins, while a third pin generates a synchronous serial clock. In addition, the device being communicated with is selected through the use of a chip select input that is driven with an active-high input. The data may be latched on either the rising edge or falling edge of the clock signal depending on the device.
- 3.3.6 System Management Bus:** A two wire serial bus used for communications to devices external to the microcontroller. See the SMBUS 2.0 specification.

3.2 Symbols and Abbreviations

| | |
|--------|---|
| ADC | Analog Digital Converter or Conversion |
| ESA | Enthusiast System Architecture |
| ISO | International Organization for Standardization |
| LED | Light Emitting Diode |
| LSb | Least Significant Bit |
| LSB | Least Significant Byte |
| MSb | Most Significant Bit |
| MSB | Most Significant Byte |
| PWM | Pulse Width Modulation |
| SMBUS | System Management Bus |
| SPI | Serial Peripheral Interface |
| USB | Universal Serial Bus |
| UTF-16 | 16-bit Unicode Translation Format (see ISO/IEC 10646-1) |

3.3 Keywords

- 3.3.1 Expected:** A keyword used to describe the behavior of the hardware or software in the design models assumed by this specification.
- 3.3.2 Invalid:** A keyword used to describe an illegal or unsupported bit, byte, word, field, or code value. Receipt of an invalid bit, byte, word, field, or code value shall be reported as an error.
- 3.3.3 Mandatory:** A keyword indicating an item that is required to be implemented as defined in this specification
- 3.3.4 May:** A keyword that indicates flexibility of choice with no implied preference.
- 3.3.5 May not:** Keywords that indicate flexibility of choice with no implied preference.
- 3.3.6 Obsolete:** A keyword indicating that an item was previous defined but has been removed from the most recent version of the specification.
- 3.3.7 Optional:** A keyword that describes features that are not required to be implemented by this specification. However, if any optional feature defined by this specification is implemented, it shall be implemented as defined in this specification.
- 3.3.8 Reserved:** A keyword referring to bits, bytes, words, fields, and code values that are set aside for future specification. Their use and interpretation may be specified by future extensions to this specification. A reserved bit, byte, word, or field shall be set to zero. Recipients are not required to check reserved bits, bytes, words, or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error.
- 3.3.9 Shall:** A keyword indicating a mandatory requirement (equivalent to “is required”). Designers are required to implement all such requirements to ensure interoperability with technology designed to this specification.
- 3.3.10 Should:** A keyword indicating flexibility of choice with a preferred alternative (equivalent to “it is strongly recommended”).
- 3.3.11 Vendor Specific:** A keyword indicating specification of the referenced item is determined by the implementer of the device conforming to this specification.

3.4 Conventions

3.4.1 Overview

Certain words and terms used in this specification have a specific meaning beyond their normal English language meaning. These words and terms are defined in either 3.1 or in the text where they first appear.

The names of fields are in small uppercase (e.g., ZONE IDENTIFIER). When a field name is a concatenation of acronyms, uppercase letters may be used for readability (e.g., SMBADDRESS). Normal case is used when the contents of a field are being discussed. Fields containing only one bit are usually referred to as the NAME bit instead of the NAME field.

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values. Notes do not constitute any requirements for implementers and shall be considered informative.

3.4.2 Numeric conventions

A binary number is represented in this specification by any sequence of digits comprised of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this specification by any sequence of digits comprised of only the Western-Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., AB90h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., AB12 23CDh or AB12_23CDh).

A decimal number is represented in this specification by any sequence of digits comprised of only the Western-Arabic numbers 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 99).

When the value of a bit or field is not relevant, x or xx appears in place of a specific value.

This standard uses the ISO convention for representing decimal numbers (e.g., the thousands and higher numbers are separated by a space and a comma is used in the decimal point). Table 1 show examples of decimal numbers represented using the ISO and American conventions.

Table 1 - ISO and American numbering conventions examples

| ISO | American |
|--------------|--------------|
| 0,5 | 0.5 |
| 3,141 592 65 | 3.14159265 |
| 1 000 | 1,000 |
| 1 234 567,89 | 1,234,567.89 |

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no ordering relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, and 3-green) show an ordering relationship between the listed items.

3.4.3 Bit and byte ordering

In this specification, data structures may be referred to by a table. A table defines the complete ordering of elements (i.e., bits, bytes, and fields) within the structure. The ordering of elements within a table does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this specification, may constrain the order of the storage or transmission of the structure.

In any table, any element that is presented in a row below another element is more significant than the upper element, and any element presented to the left of another element in the same row is more significant than the element to the right.

If a table shows bit numbering (see Table 2), the least significant bit (LSb) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes (see Table 2), the most significant byte is represented at the highest number and each less significant byte or character has the lower number than the more significant byte.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number denoting the presence of additional bytes.

Table 2 shows a 16-bit word containing a single value where the LSb is on the upper right in bit 0 and the MSb is on the lower left in bit 15.

Table 2 - Example ordering of bits and bytes within an element

| Bit Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------|-----------------|--------|--------|--------|--------|--------|-------|----------------|
| 0 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
| 1 | Bit 15 (MSB) | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |

This formatting of bits and bytes is equivalent to little endian formatting.

3.4.4 Use of and/or

The text 'X and/or Y' means the same as:

- a) X;
- b) Y; or
- c) X and Y;

where X and Y are defined terms, descriptive elements, steps in a procedure, or other phrases used to define requirements in this specification.

3.4.5 Hardware Pin and Port Symbols

To describe a specific pin on the microcontroller, this specification uses a convention of describing the logical port and pin of that port. The microcontroller data sheet contains a mapping of these logical ports and pins to physical pins on the microcontroller package.

A pin port combination is described as Px.y, where x is the port number, and y is the pin number of that port. For example, P0.1 indicates pin 1 on port 0.

4 Architecture

4.1 System Architecture

The ESA architecture provides an environment to allow the monitoring and control of various devices that provide functionality to a personal computer. This is accomplished through the use of a USB interface paired with a microcontroller running NVIDIA-provided firmware. Customization of the platform is accomplished through the use of NVIDIA-provided tools that provide customization capability to the firmware on the microcontroller.

The ESA environment consists of software and hardware components provided by NVIDIA and provided by the implementer of this specification.

An ESA device consists of a USB interface to a set of sensors and controls for a supported class of device. For this revision of the specification, water cooling devices, power supplies, and chassis / computer case devices are supported. These devices interface with the PC through a standard set of physical, electrical, and mechanical interfaces using USB. The USB connections are made to USB ports provided by NVIDIA chipsets.

In the software environment, control and monitoring of the devices is provided through software running on the host processor. These applications communicate with the microcontroller through an NVIDIA-proprietary interface that is outside the scope of this specification.

Figure 1 shows an example operating environment of an ESA system.

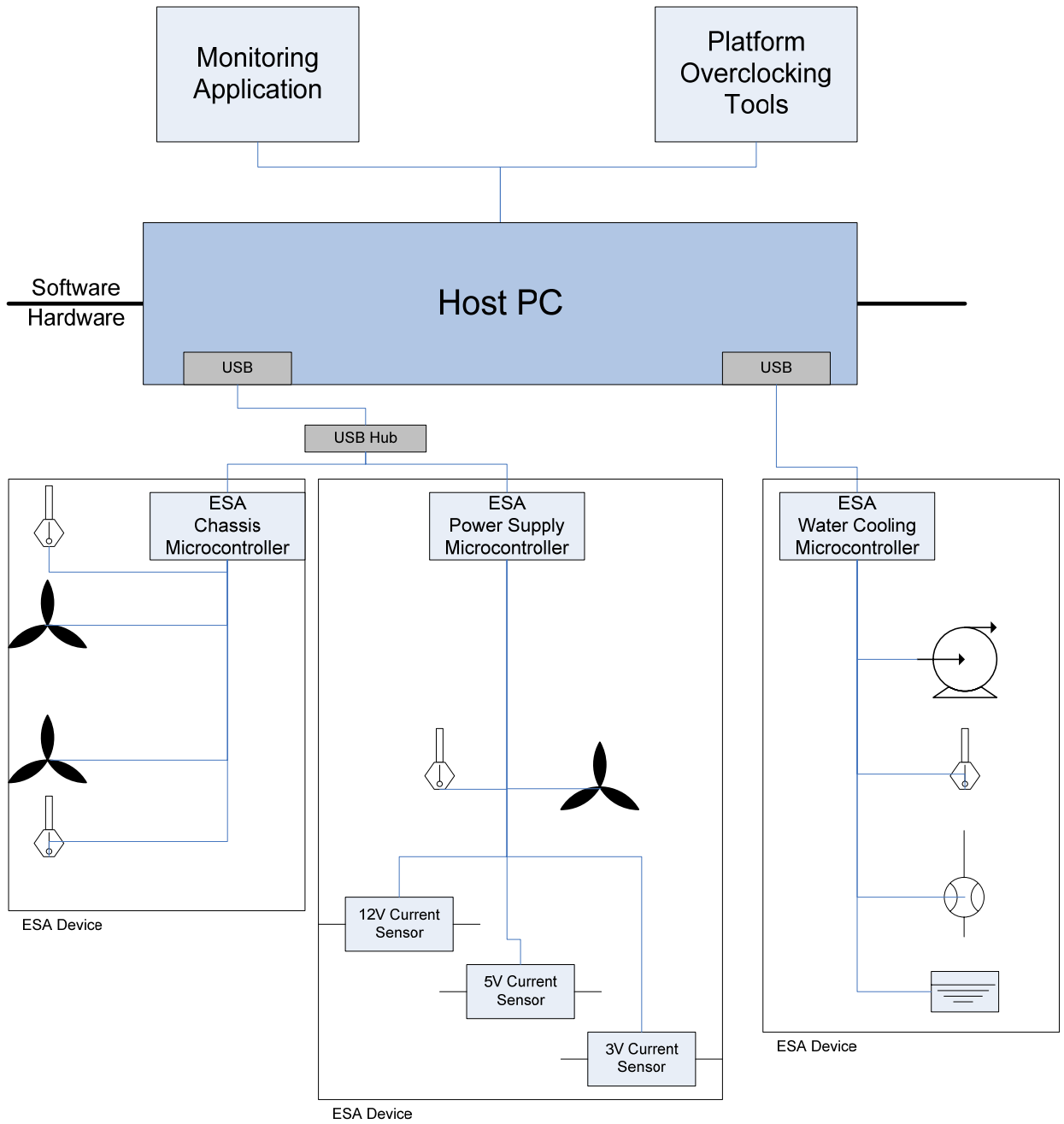


Figure 1 - ESA System Environment

4.2 Hardware Architecture

An ESA device consists of a microcontroller, supported by this specification (see clause 5.1), and one or more sensors or control devices. See Figure 2 for a representative block diagram of the architecture of an ESA device.

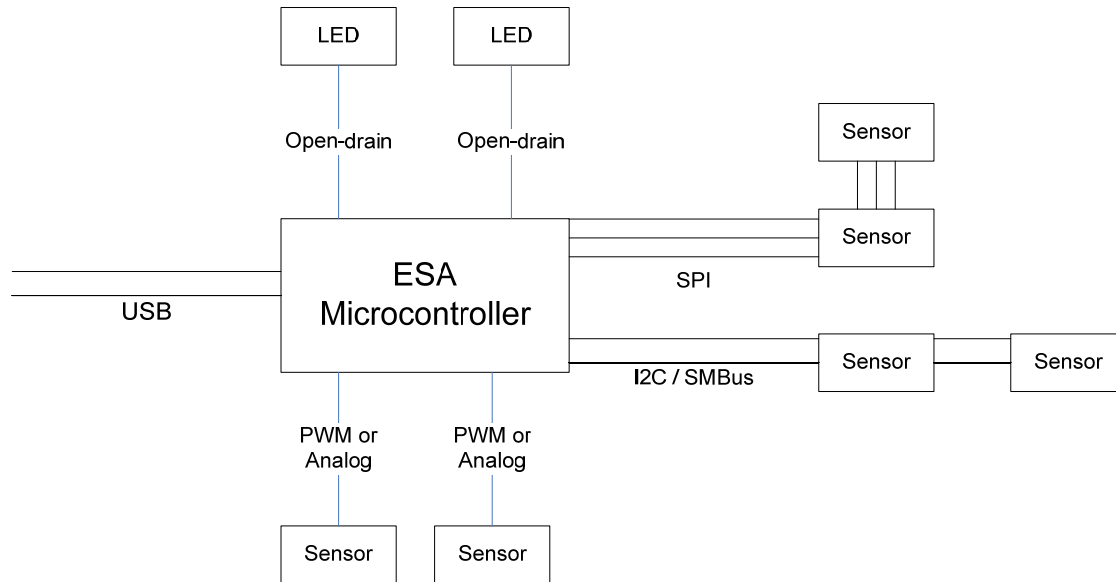


Figure 2 - ESA Hardware Block Diagram

The ESA microcontroller runs firmware provided by NVIDIA that implements the NVIDIA-proprietary communication interface between the microcontroller and the host system. The firmware also implements the controls necessary to interface with the sensor devices and controls to interface with the connected sensors, indicators, and controls.

4.3 Software Architecture

The sensors and devices attached to the microcontroller are controlled and their sensor data displayed through two applications provided by NVIDIA. The platform overclocking tools and monitor application provide a user interface to display the data read by the sensors and to provide controls for those elements that have a control factor. This data is integrated with the capabilities of other chipset and processor-specific controls such that all data related to overclocking or performance control of a system is displayed together for feedback to the user.

5 Hardware Design Requirements

5.1 Microcontroller Selection

Approved microcontrollers for this revision of the specification are:

- a) Silicon Laboratories C8051F320
- b) Silicon Laboratories C8051F321

In all respects, an ESA design shall conform to the stated electrical characteristics of these microcontrollers. In the event of a conflict between this document and the datasheet for these devices, the datasheet shall be controlling.

Implementers of ESA devices may select either of the approved microcontrollers. The C8051F320 provides four additional pins that may be used for devices than the C8051F321.

For further information on the microcontrollers see the C8051F320/1 datasheet.

5.2 Microcontroller Configuration

5.2.1 Clock Source

The ESA microcontroller is controlled by an internal oscillator and no external clock source is used.

5.2.2 Power Source

The ESA device may be powered from the USB bus or from its own external power source. If the device is configured to be powered from the USB bus, the total worst-case current consumption should not exceed 100mA. However, the device may consume up to 500mA in current, as permitted by the USB specification, when fully configured.

If the ESA device is configured to be powered from the USB bus, the device shall not exceed a current draw of 100mA on power-up without any data transfer, data sampling, or LED activation from attached devices (see Table 3).

Table 3 – Power Consumption Maximums for USB Powered Devices

| Device State | Maximum Power Consumption |
|--|---------------------------|
| Power-on Reset | 100mA |
| Recommended Operational Maximum | 100mA |
| Absolute Maximum After USB Configuration | 500mA |

5.3 Microcontroller Attached Devices

5.3.1 Fault LED (Mandatory)

All ESA devices shall implement a fault LED. The fault LED shall be red in color with a minimum intensity of 100mcd at 20mA of current. The fault LED shall be connected to the microcontroller in an open collector configuration with the microcontroller sinking the current for the LED. The LED shall be connected such that no more than 20mA of current flows through the circuit when activated.

The LED shall be mounted in a location that is user visible.

5.3.2 I2C and SPI Sensors or Controllers

I2C sensors and controllers may be used to implement functionality of an ESA device. All such sensors shall support a 400 kHz clock rate.

SPI sensors and controllers may be used to implement functionality of an ESA device. However, all SPI sensors and devices attached to the microcontroller shall latch the data on the falling edge of the clock signal.

In this revision of the specification thermal sensors, PWM controllers, and tachometer controllers are supported across a serial bus attachment.

5.3.3 Analog Sensors

The ESA microcontroller supports using analog sensors through analog to digital conversion using the ADC unit embedded in the microcontroller. All analog sensors in this revision of the specification shall be singled-ended in design and measured between a voltage reference supplied to the microcontroller by the ESA device (i.e., V_{REF}) and ground. All analog sensors shall use the same reference voltage to ensure proper measurement by the ESA microcontroller firmware.

In this revision of the specification analog thermal sensors, voltage measurement, and current measurement are supported using the ADC engine.

5.3.4 Status Indicator LED's

Indicator LED's are supported by the ESA microcontroller. LED's can be configured to represent various types of status indication (see Table 17).

5.3.5 RGB LED's

LED's that can display any color using RGB values are supported by the ESA microcontroller. RGB LED's are driven from the ESA microcontroller using PWM connections to the individual red, green, and blue inputs of the LED. Implementers shall use appropriate circuitry to supply the necessary current and drive to the LED(s) being controlled by the microcontroller. Due to limitations of the microcontroller, only one set of colors can be displayed by LEDs attached to the microcontroller; however with appropriate analog circuitry multiple such LEDs may be connected.

5.3.6 PWM Controlled Devices

ESA microcontrollers support using PWM to control various kinds of devices. Implementers shall use appropriate analog logic circuitry to supply the necessary current and drive capability to the device being controlled by PWM. For example, for a fan controlled by the microcontroller, the implementer shall implement a transistor or MOSFET power circuit with adequate protections for the current, electromagnetic force, and voltage requirements of the PWM controlled device.

5.3.7 Automatic Thermal Control

The ESA microcontroller may be configured to provide automatic thermal control of an associated thermal sensor and fan. Five steps are permitted in the automatic thermal control and are enabled through associating a thermal sensor and one or more fan controllers together. See clause 8.17 for further details on automatic thermal control and configuring the microcontroller for support of this feature.

5.4 *Microcontroller Permitted Pin Configurations*

The firmware for the microcontroller restricts certain types of devices to be configured only for use on certain physical pins of the microcontroller. This clause describes the pin selection that is permitted for a given type of device. An ESA device shall restrict the usage of pins to the permitted configurations as described in this clause. However, within a given permitted set of pins, the device may use any combination within the electrical, mechanical, and signal limits of the microcontroller (see C8051F320/1 datasheet).

Annex A contains a tabular format of permitted configurations. All annexes are considered informative text. In the event of a conflict between sub-clause 5.4 (and its sub-clauses) and the annex, sub-clause 5.4 (and its sub-clauses) shall be considered controlling.

5.4.1 Fault LED Connection

The fault LED (see 5.3.1) shall be connected to P1.0.

5.4.2 SMBUS or I2C Device Connection

All SMBUS or I2C sensors or devices shall be connected to the I2C or SMBUS pins on P0.3 and P0.4.

5.4.3 SPI Device Connection

All SPI devices shall be connected to the SPI bus pins of the microcontroller on P0.0, P0.1, and P0.2. The chip select or chip enable pin for a given SPI device may be connected to any unused port from P1.4 to P3.0. Unused PWM or tachometer pins may also be used (see 5.4.5).

5.4.4 Analog Device (ADC Measurement) Connection

If the ESA has any analog devices directly connected to the microcontroller, the implementation shall provide the voltage reference used for all analog components to pin P0.7.

5.4.4.1 Analog Thermal Sensor Circuits

ESA devices support thermal sensing using a single-ended analog circuit (e.g., thermistor voltage divider). Differential thermal sensing circuits shall not be supported or implemented by an ESA device. The analog circuit may be connected to any pin from P1.4 to P3.0.

For this revision of the specification only NTC-based thermistor voltage-divider circuits are supported.

5.4.4.2 Current Sensing Circuits

ESA devices support measuring consumed current by using a single-ended analog circuit where the voltage increases directly proportional to the current measured. The analog circuit may be to any pin from P1.4 to P3.0.

An example configuration that would meet the requirements of this clause would be a hall-effect integrated circuit where the output voltage increases as the current measured by the sensor increases (see Figure 3). Another example configuration that would meet the requirements of this clause would be a high-side current sensor/amplifier circuit (see Figure 4).

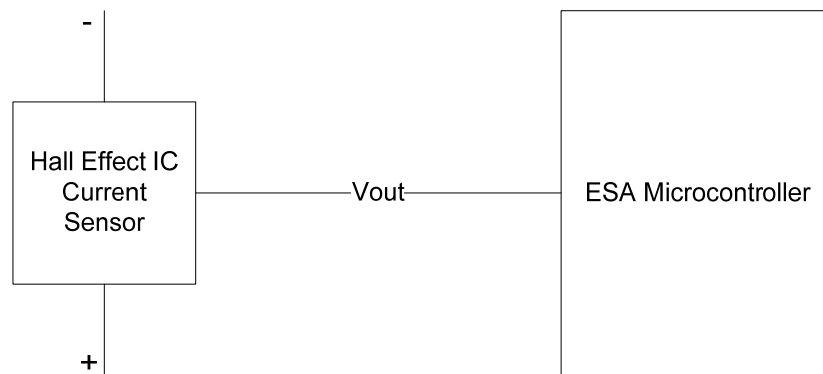


Figure 3 - Example Hall Effect Current Sensing Block Diagram

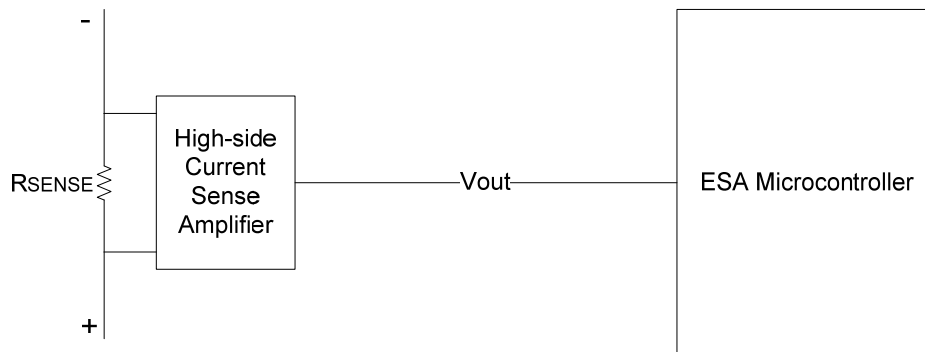


Figure 4 - Example High-Side Current Sensing Block Diagram

5.4.4.3 Voltage Sensing Circuits

ESA devices support measuring voltage levels by using a single-ended circuit where the voltage increases as the measured voltage increases. A simple voltage divider circuit would satisfy the requirements for this circuit. The configuration tables provide a scaling factor for the measured voltage in relation to the analog reference voltage provided to the microcontrollers ADC unit.

The circuit design shall ensure that the maximum ratings for the microcontroller are not exceeded under worst case measurement conditions.

The analog circuit may be connected to any pin from P1.4 to P3.0.

5.4.5 PWM Outputs and Tachometer Inputs

PWM managed devices (e.g., fans and water cooling pumps) are connected to the microcontroller on pins P0.5, P0.6, P1.1, P1.2, and P1.3. PWM outputs and tachometer inputs shall be connected to the permitted ports in the order that the ports are listed in this clause.

5.4.6 RGB LED Configurations

RGB LED connections for the red, green, and blue channels of the LED shall be restricted to connections on pins P0.5, P0.6, P1.1, P1.2, and P1.3. The channels for the LED configurations shall be connected in the order that the ports are listed in this clause.

5.4.7 Status Indicator LED Configurations

LED's may be attached to pins P1.4 to P3.0 without restriction. If there are no SPI devices connected to the microcontroller, status LED's may be connected to P0.0, P0.1, or P0.2. If there are no SMBUS or I2C devices attached to the microcontroller, pins P0.3 and P0.4 may be used. All LED's shall be configured as open-drain circuits with the microcontroller acting as the current sink for the circuit. In addition, the circuit shall not sink more than 20mA for a single LED. Restrictions apply to the maximum current that a given port on the microcontroller is permitted to sink. ESA devices shall comply with the restrictions given for maximum DC characteristics for the microcontroller (see C8051F320/1 datasheet).

5.5 Multifunction Sensors

An ESA device may use sensors that have multiple functions. A multifunction sensor may be a sensor that has more than one output for the measured variable (e.g., a thermal sensor with two separate temperature measurements) or a sensor with multiple distinct functions (e.g., a PWM controller and tachometer sensor) in one device.

To enable a sensor that has more than one function, each function has a separate configuration table entry to describe each portion of the sensor. See clause 6 for information about configuration tables.

6 Programming and Configuring ESA Microcontrollers

6.1 Configuration Tools

To implement the ESA microcontroller NVIDIA provides the main system firmware in an Intel Hex file format. In addition to the microcontroller firmware, NVIDIA also provides a tool to translate the configuration tables into microcontroller specific actions for the configuration of the device. The output of this tool is provided in an Intel HEX file format.

The implementer of the ESA device may select any programming tool for the supported microcontroller(s) that meets their requirements for the manufacturing environment. The programming tool must support both the OMF-51 and Intel HEX file format.

6.2 Configuration Procedure

To configure the ESA microcontroller after the device design has been completed, the implementer shall execute the following steps:

1. Create device and sensor configuration tables for the devices attached to the microcontroller.
2. Process configuration table through the NVIDIA configuration tool and obtain programmer output
3. Program microcontroller with reference firmware
4. Program microcontroller with table output from step 2

7 Configuration Table Format

7.1 Overview

The configuration table provides a virtual schematic and operation to the firmware of the sensors and devices attached to the microcontroller that make up the ESA device. Before programming the device with the reference firmware, the implementer of the ESA device creates a configuration table to be processed into a firmware table. This table is then programmed into the flash memory to be processed by the firmware during normal operation.

The configuration table is constructed as a raw binary file with various entries for each sensor type as describe in clause 8.

The configuration file describes not only the types of sensors attached, but the parameters, under which they operate, the type of data reported, as well as the location and grouping of the sensors. This information is conveyed to the microcontroller and other software applications that are part of the ESA architecture.

7.2 Sensor Zones

Grouping of related devices is accomplished through the use of zones. A zone is a logical group of sensors that provide related information about a given type of operation. For example, in a chassis device the fans located at the back of the case with a thermal sensor near them would be grouped as a single zone. The fans and thermal sensor at the front of the case would then be grouped as a separate single zone. This provides for logical control of the sensors to achieve common operation(s) such as thermal-based fan control.

A device shall not have more than 12 zones in this revision of the specification. All sensors shall be associated with a zone.

7.3 Axial Sensor Position

In addition to the grouping of sensors and devices into zones, the configuration tables also provide a mechanism to describe the physical position of the sensor. These measurements are made in centimeters from a reference datum in the enclosure where the sensors and devices are located.

All axial sensor positions shall have a reference datum in the x, y, and z axis of 0, 0, 0 where the datum point is located and the bottom left rear corner of the enclosure when the enclosure is positioned in the normal usage configuration.

For example, in a standard ATX chassis, the datum point would be the corner of the chassis viewed with the side cover removed at the base of the chassis to the rear on the left.

8 Table Elements Reference

8.1 Reference Overview

This clause describes each possible element in a configuration table and the required and optional fields in each element.

8.2 Table Description Element

The table description element describes the general parameters of the entire configuration table for the ESA device. All ESA devices shall have one table description element. The element shall be located at address zero of the configuration table memory.

The format of the table description element is described in Table 4.

Table 4 - Table Description Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|--------------------------|-------------|---|---|-------------------------|---|---|---|
| 0 | MAJOR REVISION | | | | MINOR REVISION | | | |
| 1 | TABLE LENGTH | | | | | | | |
| 2 | | | | | | | | |
| 3 | ESA DEVICE TYPE | | | | | | | |
| 4 | STRING INDEX | | | | | | | |
| 5 | SERIAL NUMBER | | | | | | | |
| 7 | | | | | | | | |
| 8 | SPWR | POWER UNITS | | | | | | |
| 9 | PORT HARDWARE REVISION 0 | | | | PIN HARDWARE REVISION 0 | | | |
| 10 | PORT HARDWARE REVISION 1 | | | | PIN HARDWARE REVISION 1 | | | |
| 11 | PORT HARDWARE REVISION 2 | | | | PIN HARDWARE REVISION 2 | | | |
| 12 | Reserved | | | | | | | |
| 15 | | | | | | | | |

The MAJOR REVISION and MINOR REVISION fields indicate what revision of the ESA specification that the ESA device complies with. MAJOR REVISION and MINOR REVISION shall be set according to Table 5.

Table 5 - Specification Revision Coding Values

| Specification Revision | MAJOR REVISION | MINOR REVISION |
|------------------------|----------------|----------------|
| 0.90 | 0 | 1 |
| 1.00 | 1 | 0 |

The TABLE LENGTH field shall be sent to the length of the configuration table, in bytes, including all elements.

The ESA DEVICE TYPE field identifies the category of device that is described by the configuration table. Table 6 lists the possible values for the ESA DEVICE TYPE field.

Table 6 - ESA Device Type

| Code | Description |
|-----------|----------------------|
| 00h | Reserved |
| 01h | Power Supply |
| 02h | Computer Case |
| 03h | Water Cooling Device |
| 04h – FFh | Reserved |

The STRING INDEX field identifies a string, by numeric index value, in this table that provides a description of the device or another type of identifier (e.g., product name). The value in this field directly corresponds to a string element (see 8.16) with the same numeric value. A value of zero in this field indicates that no string is provided. A device should provide a string for the table description element.

The SERIAL NUMBER field contains a unique serial number for each device, qualified with the ESA DEVICE TYPE field. Device serial number ranges are assigned to the implementer of this specification by NVIDIA at the time the license is granted. Implementers shall not use a serial number range outside of the range assigned to their product. Serial numbers across ESA device types need not be unique (e.g., a power supply and a water cooling device from the manufacturer may be identical).

If the device is self-powered and draws no power from the USB voltage rails, the SPWR field shall be set to one. If the SPWR field is set to zero, the configuration table shall also contain the maximum amount of power that the device consumes, in 4mA units rounded up to the next integer, in the POWER UNITS field.

The hardware revision fields are GPIO pins that are used to read a static value from the ESA device. The device provides pin and port combinations for up to 3 hardware revision bits, for a possible eight revisions. If a field is not used, the port and pin fields for that pin shall be set to FFh. The hardware revision strapping is optional to implement.

8.3 Common Element Fields

8.3.1 Common Element Header

All elements, with the exception of the table description element, in a configuration table contain a common header that identifies the element and describes the size of the element.

The format of the common header is described in Table 7.

Table 7 - Common Element Header

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|----------------|---|---|---|---|---|---|---|
| 0 | IDENTIFIER | | | | | | | |
| 1 | ELEMENT LENGTH | | | | | | | |
| 2 | | | | | | | | |

IDENTIFIER contains a unique value that indicates what type of element this common header is a part of. Each element has a unique value and this field shall be set to the value specified for the element.

ELEMENT LENGTH contains the length, in bytes, of the entire element descriptor including the common element header.

8.3.2 Serial Bus Address Encoding

Elements that describe a device attached through a serial bus use a common two-byte address field. The encoding of this two-byte value depends on the serial bus that is being enumerated. To describe the type of bus the device is attached to, a common sensor bus type field is used. The coding of this field is described in Table 8.

Table 8 - Bus Type Encoding

| Code | Bus Type |
|---------|------------------|
| 0h | I2C |
| 1h | SMBUS Byte Mode |
| 2h | SMBUS Block Mode |
| 3h | SPI |
| 4h – Fh | Reserved |

For each type of serial sensor, a DEVICE ADDRESS field is provided. The formatting of that field is described in following sub-clauses for each supported serial-bus type.

8.3.2.1 I2C and SMBUS Encoding

For devices using an I2C or SMBUS, Table 9 describes the format of the DEVICE ADDRESS field.

Table 9 – Device Address Encoding for I2C and SMBUS

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-------------|---|---|---|---|---|---|----------|
| 0 | BUS ADDRESS | | | | | | | Reserved |
| 1 | Reserved | | | | | | | |

The BUS ADDRESS field in this encoding shall be the seven-bit address transmitted across SMBUS or I2C that causes the device to respond to requests. Devices that require a 10-bit I2C address shall not be connected to the ESA microcontroller.

8.3.2.2 SPI Encoding

For devices using a SPI bus, Table 10 describes the format of the DEVICE ADDRESS field.

Table 10 – Device Address Encoding for SPI

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|----------|---|---|---|-------|---|---|---|
| 0 | CSPORT | | | | CSPIN | | | |
| 1 | Reserved | | | | | | | |

The CSPORT field contains the port number containing the pin that, when driven to an active-high signal, causes the device to decode the data on the SPI bus. The CSPIN field contains the pin number of the pin, within the port described by the CSPORT field, that when driven to an active-low signal causes the device to decode the data on the SPI bus.

8.4 Digital Thermal Sensor Element

A digital thermal sensor element describes a digital thermal sensor that is attached to the microcontroller through a serial interface.

The format of the digital thermal sensor element is described in Table 11.

Table 11 - Digital Thermal Sensor Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-------------------------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | Reserved | | | | BUS TYPE | | | |
| 9 | DEVICE ADDRESS | | | | | | | |
| 10 | | | | | | | | |
| 11 | TEMPERATURE LSB READ REGISTER | | | | | | | |
| 12 | | | | | | | | |
| 13 | TEMPERATURE MSB READ REGISTER | | | | | | | |
| 14 | | | | | | | | |
| 15 | MINIMUM VALUE | | | | | | | |
| 16 | | | | | | | | |
| 17 | MINIMUM TEMP | | | | | | | |
| 18 | MAXIMUM VALUE | | | | | | | |
| 19 | | | | | | | | |
| 20 | MAXIMUM TEMP | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 01h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The BUS TYPE field identifies the serial bus interface and general parameters of the thermal sensor. The field shall be set to a value from Table 8.

The DEVICE ADDRESS field contains the bus address of the thermal sensor. See 8.3.2 for the encoding of this field for the serial bus type.

The TEMPERATURE LSB READ REGISTER and TEMPERATURE MSB READ REGISTER contain the register addresses that will be transmitted across the serial bus to read the LSB and MSB, respectively, of the current temperature. If the device only provides 8-bit or less resolution, then the TEMPERATURE MSB READ REGISTER field shall be set to FFh.

The MINIMUM VALUE and MAXIMUM VALUE fields contain the minimum and maximum values, respectively that are read from the thermal sensor through the temperature register.

The MINIMUM TEMP and MAXIMUM TEMP fields contain the temperature that the MINIMUM VALUE and MAXIMUM VALUE fields represent. The units for these values are in degrees centigrade.

8.5 Analog Thermal Sensor Element

An analog thermal sensor element describes an analog circuit to measure temperature using the analog to digital conversion unit in the microcontroller. An implementer of an analog thermal circuit shall only implement single-ended thermal sensing circuits.

The format of the analog thermal sensor element is described in Table 12.

Table 12 – Analog Thermal Sensor Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|-----|---------------------|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | Reserved | | | | | THERMAL ANALOG TYPE | | |
| 9 | PORT | | | | PIN | | | |
| 10 | VALUE0 | | | | | | | |
| 11 | | | | | | | | |
| 12 | VALUE25 | | | | | | | |
| 13 | | | | | | | | |
| 14 | VALUE50 | | | | | | | |
| 15 | | | | | | | | |
| 16 | VALUE75 | | | | | | | |
| 17 | | | | | | | | |
| 18 | VALUE100 | | | | | | | |
| 19 | | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 02h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The THERMAL ANALOG TYPE field contains an identifier describing what kind of analog circuitry is connected to the microcontroller. For this revision of the specification, this field shall be set to a value from Table 13.

Table 13 – Analog Thermal Sensor Type Coding

| Code Value | Analog Thermal Sensor Type |
|------------|----------------------------|
| 0 | Reserved |
| 1h | NTC Thermistor |
| 2h – Fh | Reserved |

The PORT field contains the port number containing the pin that the thermistor circuit is connected to. The PIN field contains the pin number of the physical pin, within the port described by the PORT field, which the thermal sensing circuit is connected to.

The VALUE0, VALUE25, VALUE50, VALUE75, and VALUE100 are known as the temperature calibration fields.

To provide calibration information to the microcontroller, the configuration table contains temperature calibration fields on a ten bit scale for a given temperature. The calibration information is calculated by assuming that the voltage through the circuit at the measured point equal to the reference voltage would be equal to 11 1111 1111b. Likewise if the voltage through the circuit at the measured point equal to the ground voltage would be equal to 00 0000 0000b.

For example a given temperature calibration field that is equal to 1,25V when the reference voltage is 2,5V would be equal to 01 1111 1111b.

The VALUE0 field contains the 10-bit scaled value for a voltage that represents zero degrees centigrade.

The VALUE25 field contains the 10-bit scaled value for a voltage that represents 25 degrees centigrade.

The VALUE50 field contains the 10-bit scaled value for a voltage that represents 50 degrees centigrade.

The VALUE75 field contains the 10-bit scaled value for a voltage that represents 75 degrees centigrade.

The VALUE100 field contains the 10-bit scaled value for a voltage that represents 100 degrees centigrade.

8.6 Microcontroller PWM Fan Element

The microcontroller PWM fan element describes a fan connected to the one of the PWM controllers implemented by the ESA microcontroller.

The format of the microcontroller PWM fan element is described in Table 14.

Table 14 – Microcontroller PWM Fan Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|--------------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | PORT PWM | | | | PIN PWM | | | |
| 9 | PORT TACH | | | | PIN TACH | | | |
| 10 | PPR | | | | | | | |
| 11 | MINIMUM RPM | | | | | | | |
| 12 | | | | | | | | |
| 13 | MAXIMUM RPM | | | | | | | |
| 14 | | | | | | | | |
| 15 | MINIMUM DUTY CYCLE | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 03h.

The STRING INDEX field indicates the string element that describes this fan (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The PORT PWM field contains the port number containing the pin that the fan driver circuit is connected to. The PIN PWM field contains the pin number of the physical pin, within the port described by the PORT PWM field, which the fan driver circuit is connected to.

The PORT TACH field contains the port number containing the pin that the fan sensor circuit is connected to. The PIN TACH field contains the pin number of the physical pin, within the port described by the PORT TACH field, which the fan sensor circuit is connected to. If the fan connected to the PWM controller does not support a tachometer output back to the microcontroller, the PORT TACH and PIN TACH fields shall be set to Fh.

The PPR field specifies how many pulses are generated by the fan per revolution. If the fan is not configured for a tachometer output, this field shall be set to zero.

The MINIMUM RPM field contains a number of revolutions per minute that the fan should not be operated below to ensure safe operation.

The MAXIMUM RPM field indicates, if a PWM duty cycle of 100% is applied to the fan, the speed of the fan in revolutions per minute.

The MINIMUM DUTY CYCLE field indicates, for a total clock of 255 cycles, the maximum number of clocks that may be low during the PWM cycle to ensure safe operation of the fan.

8.7 Status LED Element

The status LED element describes an open-drain LED connected to the microcontroller.

The format of the status LED element is described in Table 15.

Table 15 – Status LED Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | PORT1 | | | | PIN1 | | | |
| 9 | COLOR1 | | | | PURPOSE1 | | | |
| 10 | PORT2 | | | | PIN2 | | | |
| 11 | COLOR2 | | | | PURPOSE2 | | | |
| 12 | PORT3 | | | | PIN3 | | | |
| 13 | COLOR3 | | | | PURPOSE3 | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 04h.

The STRING INDEX field indicates the string element that describes this LED (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this LED belongs.

There are three port, pin, color, and purpose fields in the status LED element and are referred to as a color sub-element. A single LED element describes a single, up to tri-color LED, where each color is activated by a separate anode or a third color is activated by turning on two anodes of the device. Any documentation for the first set of these fields applies to each set of the fields except as noted.

The PORT1 field contains the port number containing the pin that the LED circuit is connected to. The PIN1 field contains the pin number of the physical pin, within the port described by the PORT field, which the LED circuit is connected to.

If there is no connection for a color sub-element, the port and pin fields for that sub-element shall be set to Fh. In addition the color and purpose fields shall be set to zero.

If the third sub-element requires that the pins for the first two elements be activated to display, the port field shall be set to Eh and the pin field also set to Eh.

The COLOR1 field contains an encoding of the color the LED displays when activated. This field shall be encoded as described in Table 16.

Table 16 – LED color coding

| Code | Color |
|--------|----------|
| 0 | Reserved |
| 1 | Red |
| 2 | Green |
| 3 | Yellow |
| 4 | Blue |
| 5 | Orange |
| 6 – 15 | Reserved |

The PURPOSE1 field contains a general purpose for the status LED. ESA devices should provide a recommended use of the LED. The purpose field shall be encoded as described in Table 17.

Table 17 – LED purpose coding

| Code | Recommended Use |
|--------|-----------------------|
| 0 | No recommended use |
| 1 | Good status |
| 2 | Warning status |
| 3 | Failure status |
| 4 | User attention signal |
| 5 – 15 | Reserved |

8.8 Water Level – Analog Element

The analog water level element describes a singled-ended analog sensor that reports an overall water level for a water cooling reservoir.

The format of the analog water level element is described in Table 18.

Table 18 – Analog Water Level Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|-----|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | PORT | | | | PIN | | | |
| 9 | FULL | | | | | | | |
| 10 | | | | | | | | |
| 11 | EMPTY | | | | | | | |
| 12 | | | | | | | | |
| 13 | WARNING | | | | | | | |
| 14 | | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 05h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The PORT field contains the port number containing the pin that the analog circuit is connected to. The PIN field contains the pin number of the physical pin, within the port described by the PORT field, which the analog circuit is connected to.

To provide calibration information to the microcontroller, the configuration table contains water level calibration fields on a ten bit scale for a given water level. The calibration information is calculated by assuming that the voltage through the circuit at the measured point equal to the reference voltage would be equal to 11 1111 1111b. Likewise if the voltage through the circuit at the measured point equal to the ground voltage would be equal to 00 0000 0000b. These values should be scaled to accommodate the worse case scenario for the select sensors accuracy.

The FULL field describes a 10-bit scaled value indicating that the water reservoir is at maximum capacity.

The EMPTY field describes a 10-bit scaled value indicating that the water reservoir is at the minimum measurable value.

The WARN field describes a 10-bit scaled value indicating that the user should be notified that an unsafe water level condition exists or will soon exist.

8.9 Microcontroller Pump PWM Element

The microcontroller pump PWM element shares the same format at the microcontroller PWM fan element (see 8.6). However, the identifier field in the common header shall be set to 06h.

8.10 Power Supply Voltage Measurement Element

The power supply voltage element describes an analog voltage detection circuit.

The format of the power supply voltage element is described in Table 19.

Table 19 – Power Supply Voltage Measurement Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | PORT | | | | PIN | | | |
| 9 | VMAX | | | | VMIN | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 07h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The PORT field contains the port number containing the pin that the analog circuit is connected to. The PIN field contains the pin number of the physical pin, within the port described by the PORT field, which the analog circuit is connected to.

The voltage detection circuit described by this element is measured by the integrated ADC engine in the ESA microcontroller. Therefore, the VMAX and VMIN fields describe the voltages, in decivolt units, detected at the two extremes of the ADC conversion.

The VMAX field contains the voltage detected when the ADC engine is at full scale (e.g., 11 1111 1111b for the current microcontrollers). The VMIN field contains the voltage detected when the 10-bit ADC engine is at minimum scale (e.g., 00 0000 0000b for the current microcontrollers).

For example, if a voltage element reads 14 volts when the ADC reads 11 1111 1111b and 0 volts when the ADC reads 00 0000 0000b after conversion, then the VMAX field would be 140 and the VMIN field 0.

8.11 Current Measurement Element

The current element describes an analog sensing element to determine the amount of current flowing through a given power supply rail.

The format of the power supply current measurement element is described in Table 20.

Table 20 – Power Supply Current Measurement Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|-----|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | PORT | | | | PIN | | | |
| 9 | Reserved | | | | | | | |
| 10 | IMIN | | | | | | | |
| 11 | | | | | | | | |
| 12 | IMAX | | | | | | | |
| 13 | | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 08h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The PORT field contains the port number containing the pin that the analog circuit is connected to. The PIN field contains the pin number of the physical pin, within the port described by the PORT field, which the analog circuit is connected to.

A current measurement sensor circuit uses the ADC in the microcontroller to measure a specific voltage compared to the reference voltage in order to represent a current detected.

The IMIN field describes the current sensed, in deciamps, when the ADC detects a minimum scale value when compared with the reference voltage. The IMAX field describes the current sensed, in deciamps, when the ADC detects a maximum scale value when compared with the reference voltage.

8.12 External PWM Controller Element

The external PWM controller element describes a PWM controller attached to the ESA microcontroller through a serial bus.

The format of the external PWM controller element is described in Table 21.

Table 21 – External PWM Controller Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | Reserved | | | | BUS TYPE | | | |
| 9 | DEVICE ADDRESS | | | | | | | |
| 10 | | | | | | | | |
| 11 | DEFAULT SPEED BLOCK | | | | | | | |
| n | SPEED BLOCK | | | | | | | |
| n+1 | | | | | | | | |
| p | SPEED BLOCK | | | | | | | |
| p+1 | | | | | | | | |
| ... | SPEED BLOCK | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 09h.

The STRING INDEX field indicates the string element that describes this sensor (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this sensor belongs.

The BUS TYPE field identifies the serial bus interface and general parameters of the thermal sensor. The field shall be set to a value from Table 8.

The DEVICE ADDRESS field contains the bus address of the thermal sensor. See 8.3.2 for the encoding of this field for the serial bus type.

8.12.1 Speed Block Encoding

All external PWM controller elements shall have a DEFAULT SPEED BLOCK field. The default speed block is comprised of several fields that describe how to program the PWM controller to set a default speed on power-up. The fields that make up the DEFAULT SPEED BLOCK field are described in Table 22.

In addition to the default speed block, the element may provide one or more speed blocks to describe the register accesses necessary to implement the speed described by the SPEED field. Each speed block contains at least one register pair (see 8.12.2) to access the registers associated with the PWM controls. The device shall have no more than 5 speed blocks.

Table 22 – Speed Block

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|---|---|---|---|
| 0 | SPEED | | | | | | | |
| 1 | ENTRIES | | | | | | | |
| 2 | REGISTER PAIR 1 | | | | | | | |
| 5 | | | | | | | | |
| 6 | REGISTER PAIR N | | | | | | | |
| N | | | | | | | | |

The SPEED field contains an encoded value described what speed is set when the block of register writes are executed by the microcontroller across the serial bus. The speed shall be encoded in 100 RPM units.

The ENTRIES field contains how many register pair fields that this block contains. The block shall contain at least one register pair. The microcontroller, when setting the speed described by this block, executes each register pair in order when writing to the PWM controller.

8.12.2 Register Pair Fields

The register pair field is comprised of two sub-fields. Each register pair contains a register address and data that is written to. The sub-fields are described in Table 23.

Table 23 – Register Pair Fields

| Byte Offset | Field |
|-------------|------------------|
| 0 | REGISTER ADDRESS |
| 2 | REGISTER DATA |

The REGISTER ADDRESS sub-field contains the address within the device at the address referred to by the DEVICE ADDRESS field (see 8.3.2).

The REGISTER DATA sub-field contains the value that the microcontroller writes to the register addressed by the REGISTER ADDRESS sub-field. For example, if the controller requires that register 7 needs a value of 12 to be written to the device, the register address is set to 7 and the register data to 12.

A device that requires more than one register access in order to implement the required functionality has a register pair for each register access for that block.

8.13 External Tachometer Element

The external tachometer element describes a tachometer counter attached to the microcontroller through a serial bus.

The format of the external tachometer element is described in Table 24.

If the external tachometer element is associated with a device described by an external PWM controller element, the tachometer element shall immediately follow the PWM controller element in the table.

Table 24 – External Tachometer Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------------------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | Reserved | | | | BUS TYPE | | | |
| 9 | DEVICE ADDRESS | | | | | | | |
| 10 | | | | | | | | |
| 11 | RESERVED | | | | | | | |
| 12 | CHIP INITIALIZATION BLOCK | | | | | | | |
| N | | | | | | | | |
| N+1 | READ STATUS BLOCK | | | | | | | |
| P | | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 0Ah.

The STRING INDEX field indicates the string element that describes this controller (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this controller belongs.

The BUS TYPE field identifies the serial bus interface and general parameters of the tachometer controller. The field shall be set to a value from Table 8.

The DEVICE ADDRESS field contains the bus address of the tachometer controller. See 8.3.2 for the encoding of this field for the serial bus type.

The CHIP INITIALIZATION BLOCK field contains sub-fields to describe a set of registers to write, and associated data to write to those registers, to initialize the tachometer controller. The CHIP INITIALIZATION BLOCK field is described in Table 25.

Table 25 – Read Status / Tachometer Initialization Block

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|---|---|---|---|
| 0 | ENTRIES | | | | | | | |
| 1 | REGISTER PAIR 1 | | | | | | | |
| 4 | | | | | | | | |
| 5 | | | | | | | | |
| N | REGISTER PAIR N | | | | | | | |

The ENTRIES sub-field contains the number of register pair fields that follow this field.

Each register pair contains the register reads necessary to access the external tachometer controller and configure the tachometer for further operations. The format of the register pair fields is described in Table 23.

The READ STATUS BLOCK field contains sub-fields to describe a set of registers to read in order to obtain the current tachometer input. Table 25 describes the format of the READ STATUS BLOCK field.

8.14 Microcontroller RGB LED Element

The microcontroller RGB LED element describes and LED that is attached to the PWM pins of the microcontroller.

The format of the microcontroller RGB LED element is described in Table 26.

Table 26 – Microcontroller RGB LED Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------|---|---|---|-----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | RED PORT | | | | RED PIN | | | |
| 9 | GREEN PORT | | | | GREEN PIN | | | |
| 10 | BLUE PORT | | | | BLUE PIN | | | |

The COMMON HEADER field is defined in 8.3.1. The IDENTIFIER field in the common header shall be set to 0Eh.

The STRING INDEX field indicates the string element that describes this LED (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this LED belongs.

The port fields contain the port number containing the pin that the LED is connected to. The pin field contains the pin number of the physical pin, within the port described by the port field, which the LED is connected to. The red, green, and blue inputs to the LED are all described individually.

8.15 External RGB LED Element

The external RGB LED element describes an RGB LED that is attached to a SPI or I2C PWM controller.

The format of the external RGB LED element is described in Table 27.

Table 27 – External RGB LED Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------------------|---|---|---|----------|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING INDEX | | | | | | | |
| 4 | X POS | | | | | | | |
| 5 | Y POS | | | | | | | |
| 6 | Z POS | | | | | | | |
| 7 | ZONE IDENTIFIER | | | | | | | |
| 8 | Reserved | | | | BUS TYPE | | | |
| 9 | DEVICE ADDRESS | | | | | | | |
| 10 | | | | | | | | |
| 11 | CHIP IDENTIFIER | | | | | | | |
| 12 | RED ENTRIES | | | | | | | |
| 13 | RED REGISTER PAIR | | | | | | | |
| N | | | | | | | | |
| N+1 | GREEN ENTRIES | | | | | | | |
| N+2 | GREEN REGISTER PAIR | | | | | | | |
| P | | | | | | | | |
| P+1 | BLUE ENTRIES | | | | | | | |
| P+2 | BLUE REGISTER PAIR | | | | | | | |
| Q | | | | | | | | |

The COMMON HEADER field is defined in 8.3.1. The IDENTIFIER field in the common header shall be set to 0Dh.

The STRING INDEX field indicates the string element that describes this controller (see 8.16). A device should provide a string describing the element. If no string descriptor is provided, this field shall be set to zero.

The X POS, Y POS, and Z POS fields describe the distance, in centimeters; from the datum point of the enclosure holding the device (see 7.3). If measurement is not possible or is not meaningful for the device, all three fields shall be set to FFh.

The ZONE IDENTIFIER field contains the zone (see 7.2) to which this controller belongs.

The BUS TYPE field identifies the serial bus interface and general parameters of the LED controller. The field shall be set to a value from Table 8.

The DEVICE ADDRESS field contains the bus address of the tachometer controller. See 8.3.2 for the encoding of this field for the serial bus type.

The CHIP IDENTIFIER field contains a coded value indicating the manufacturer and model of the external RGB LED controller. ESA devices shall not describe externally attached RGB controllers that are not coded in Table 28. Based on the chip selected, the device may present an intensity option for the RGB LED.

Table 28 – RGB LED Controller Identifier

| Code Value | Manufacturer / Model |
|------------|----------------------|
| 01h | Maxim MAX7313 |
| 02h | NXP PCA9633 |
| 03h | NXP PCA9634 |

The RED ENTRIES, GREEN ENTRIES, AND BLUE ENTRIES fields contains the number of register pair fields that follow the field to set the red, green, and blue control component of the LED controller respectively.

Each register pair fields contain the register writes necessary to program the external RGB controller and configure the controller for further operations. The format of the register pair fields is described in 8.12.2.

8.16 String Element

The string element provides a UTF-16 encoded Unicode string describing another element entry in the configuration table.

The format of the string element is described in Table 29.

Table 29 - String Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|---------------|---|---|---|---|---|---|---|
| 0 | COMMON HEADER | | | | | | | |
| 2 | | | | | | | | |
| 3 | STRING ID | | | | | | | |
| 4 | LOCALE ID | | | | | | | |
| 5 | | | | | | | | |
| 6 | STRING DATA | | | | | | | |
| N | | | | | | | | |

The COMMON HEADER field is defined in 8.3.1. The IDENTIFIER field in the common header shall be set to 0Ch.

The STRING ID field contains the identifier for this string. The string identifier shall match at least one other non-string element. The use of the STRING ID field in this element is to identify a string that was referenced in another element. For example, if an instance of a string element has a string index of seven and a digital thermal sensor element also has a string index of seven, then the string element is the textual description of that thermal sensor. An implementation may provide multiple instances of the same string index value to describe an element in the configurations table for differing languages. However, for a given LOCALE ID all string identifiers shall be unique. A device shall not set the STRING ID to zero.

The LOCALE ID field contains the language identifier for the string. This field shall be set to the appropriate language for the string described by this block and be an approved language identifier for a USB device (see USB Language Identifiers). If a device provides a string for an element (i.e., a specific STRING INDEX value), it shall provide a string for language identifier 0409h (i.e. English (United States)). All other locale identifiers are optional for a given string index.

Due to operating system limitations, it is recommended that localized strings be omitted and only English strings provided for devices complying with this revision of the specification.

STRING DATA contains the Unicode string. The string shall not be null-terminated and shall be no more than 64 bytes in length.

8.17 Zone Descriptor Element

The zone descriptor element provides a description for a grouping of elements into a zone.

The format of the zone description element is described in Table 30.

Table 30 – Zone Description Element

| Bit/Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|----------|-----------------|---|---|---|---|---|------|--------|--|
| 0 | COMMON HEADER | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | ZONE IDENTIFIER | | | | | | | | |
| 4 | STRING INDEX | | | | | | | | |
| 5 | Reserved | | | | | | SAVE | ATHERM | |
| 6 | TEMPERATURE 0 | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | FAN SPEED 0 | | | | | | | | |
| 9 | | | | | | | | | |
| 10 | TEMPERATURE 1 | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | FAN SPEED 1 | | | | | | | | |
| 13 | | | | | | | | | |
| 14 | TEMPERATURE 2 | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | FAN SPEED 2 | | | | | | | | |
| 17 | | | | | | | | | |
| 18 | TEMPERATURE 3 | | | | | | | | |
| 19 | | | | | | | | | |
| 20 | FAN SPEED 3 | | | | | | | | |
| 21 | | | | | | | | | |
| 22 | TEMPERATURE 4 | | | | | | | | |
| 23 | | | | | | | | | |
| 24 | FAN SPEED 4 | | | | | | | | |
| 25 | | | | | | | | | |
| 26 | TEMPERATURE 5 | | | | | | | | |
| 27 | | | | | | | | | |
| 28 | FAN SPEED 5 | | | | | | | | |
| 29 | | | | | | | | | |

The COMMON HEADER field is described in 8.3.1. The identifier field in the header shall be set to 0Bh.

The ZONE IDENTIFIER field contains the numeric identifier that is being associated with a descriptive string. For any given zone identifier within another type of element, there shall be exactly one zone descriptor element for that value. For example, if a digital thermal sensor (see 8.4) has a seven in its ZONE IDENTIFIER field, then the zone descriptor element that describes the zone that the thermal sensor belongs to shall have a seven in its ZONE IDENTIFIER field.

The STRING INDEX field indicates the string element that describes this zone (see 8.16). A device should provide a string describing the zone. If no string descriptor is provided, this field shall be set to zero.

The `ATHERM` field, when set to one, indicates that this zone contains a thermal sensor and one or more fan or pump controls and that the temperature and fan speeds following this field contain valid data. In addition, setting this field to one indicates that the ESA microcontroller shall provide automatic fan control using the following values. Implementers shall not set this field to one unless there is one thermal sensor element and one or more fan or pump PWM elements in this zone.

Each temperature field in the element corresponds to a fan speed field with the same numeric identifier. These fields comprise a fan curve that defines automatic thermal control when enabled with the `ATHERM` field set to one. The temperatures shall be considered a minimum value at which the corresponding fan speed shall be enabled. The temperature fields shall be scaled values that are read directly from the sensor, analog or thermal. For example, if the thermal sensor for this zone is an analog sensor then the temperature field is scaled to correspond to the 10bit analog value. Likewise, the fan PWM output shall also be considered a scaled value corresponding to the PWM output of the associated fan controllers.

The `SAVE` field, when set to one, indicates that the fan curve information may be overridden with user settings transmitted to the microcontroller for default use and saved in the microcontroller's non-volatile memory.

Annex A Pin Configuration Tables (Informative)

The following tables contains the permitted pin configurations by pin

| Port.Pin | Permitted Devices or Connections | | | |
|--|----------------------------------|------------------|------------------|---------|
| P0.0 | SPI Sensor – SCK | LED ¹ | | |
| P0.1 | SPI Sensor – MISO | LED ¹ | | |
| P0.2 | SPI Sensor – MOSI | LED ¹ | | |
| P0.3 | SMBUS Device - SDA | I2C Device - SDA | LED ⁴ | |
| P0.4 | SMBUS Device - SCL | I2C Device - SCL | LED ⁴ | |
| P0.5 | PWM Output | Tachometer | LED | RGB LED |
| P0.6 | PWM Output | Tachometer | LED | RGB LED |
| P0.7 | Analog VRef | | | |
| P1.0 | Fault LED ² | | | |
| P1.1 | PWM Output | Tachometer | LED | RGB LED |
| P1.2 | PWM Output | Tachometer | LED | RGB LED |
| P1.3 | PWM Output | Tachometer | LED | RGB LED |
| P1.4 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P1.5 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P1.6 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P1.7 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.0 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.1 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.2 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.3 | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.4 ³ | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.5 ³ | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.6 ³ | Analog Thermal | Current Sense | Voltage Sense | LED |
| P2.7 ³ | Analog Thermal | Current Sense | Voltage Sense | LED |
| P3.0 | Analog Thermal | Current Sense | Voltage Sense | LED |
| Note 1 – LED's are permitted on these pins only when no SPI sensors are connected Note 2 – Fault LED is mandatory to implement Note 3 – These pins are not present on the C8051F321 part Note 4 – These devices are only permitted when no SMBUS or I2C sensors are connected | | | | |

| Port.Pin | Permitted Devices or Connections (continued) |
|--|--|
| P0.0 | |
| P0.1 | |
| P0.2 | |
| P0.3 | |
| P0.4 | |
| P0.5 | |
| P0.6 | |
| P0.7 | |
| P1.0 | |
| P1.1 | |
| P1.2 | |
| P1.3 | |
| P1.4 | SPI CS |
| P1.5 | SPI CS |
| P1.6 | SPI CS |
| P1.7 | SPI CS |
| P2.0 | SPI CS |
| P2.1 | SPI CS |
| P2.2 | SPI CS |
| P2.3 | SPI CS |
| P2.4 ³ | SPI CS |
| P2.5 ³ | SPI CS |
| P2.6 ³ | SPI CS |
| P2.7 ³ | SPI CS |
| P3.0 | SPI CS |
| Note 1 – LED's are permitted on these pins only when no SPI sensors are connected Note 2 – Fault LED is mandatory to implement Note 3 – These pins are not present on the C8051F321 part Note 4 – These devices are only permitted when no SMBUS or I2C sensors are connected | |

Annex B Element Types by Numeric Value (Informative)

| Value | Element Type |
|-----------|--|
| 01h | Digital Thermal Sensor Element |
| 02h | Analog Thermal Sensor Element |
| 03h | Microcontroller PWM Fan Element |
| 04h | Status LED Element |
| 05h | Water Level – Analog Element |
| 06h | Microcontroller Pump PWM Element |
| 07h | Power Supply Voltage Measurement Element |
| 08h | Current Measurement Element |
| 09h | External PWM Controller Element |
| 0Ah | External Tachometer Element |
| 0Bh | Zone Descriptor Element |
| 0Ch | String Element |
| 0Dh | External RGB LED Element |
| 0Eh | Microcontroller RGB LED Element |
| 0Fh – FFh | Reserved |