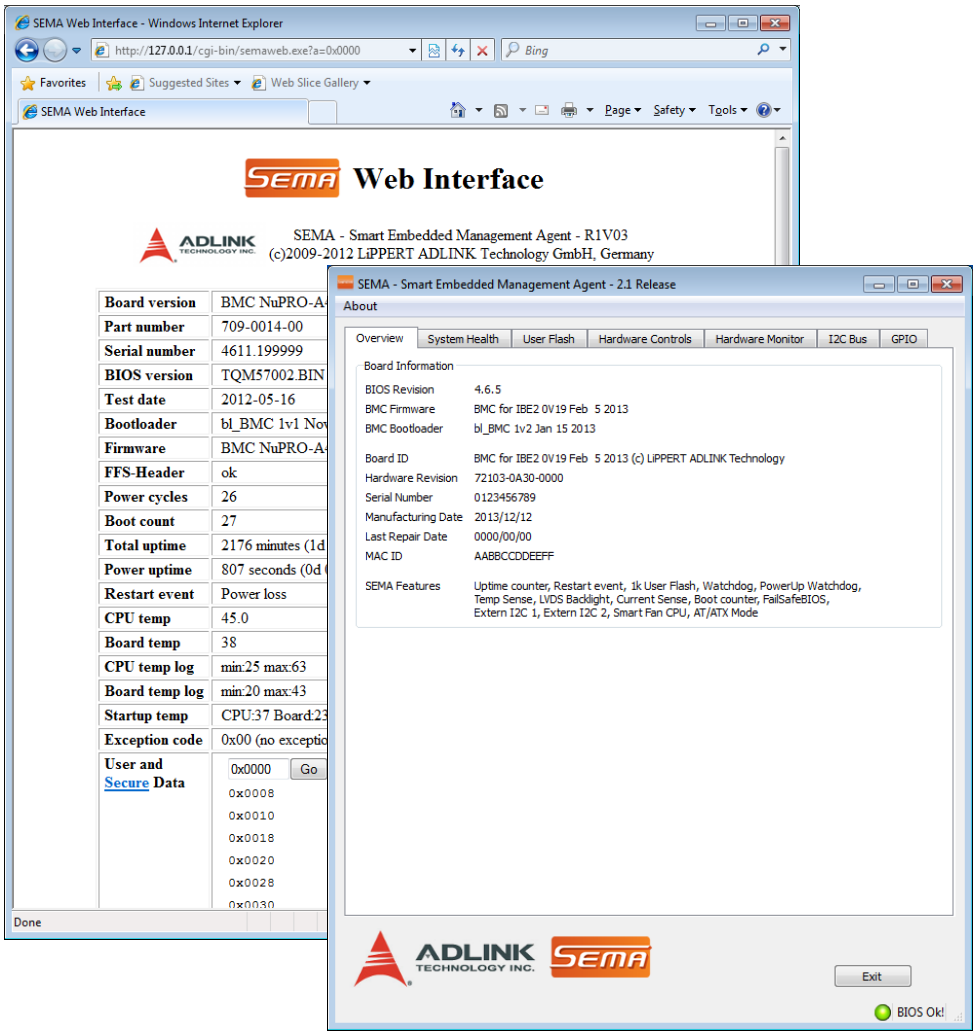
SEMA

Smart Embedded Management Agent

Software Manual



ADLINK
LiPPERT ADLINK Technology GmbH

Revision 1.3 / June 13
www.adlinktech.com

SEMA Software Manual

LiPPERT by ADLINK Document: SME-SEMA-R1V3.doc Revision 1.3

Trademarks

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows XP, Windows Vista, Windows 7 are trademarks of Microsoft Corporation. PS/2 is a trademark of International Business Machines, Inc. Intel and Intel Atom are trademarks of Intel Corporation. CoreExpress and CoreExpress logo are trademarks of LiPPERT ADLINK TECHNOLOGY GmbH. Qt, and the Qt logo are trademarks of Digia, Qt Corporation. All other trademarks appearing in this document are the property of their respective owners.

Table of Contents

1 Overview

1.1 Introduction

Today's Embedded PC Systems must combine optimal performance with low power consumption. This can be difficult to accomplish without the control and system management tools that help to find potential weaknesses in advance.

Similarly, systems must run stably and continuously, especially for critical applications in rough environments with severe shock & vibration and in extended temperature ranges. These tough requirements need a tool which is able to quickly, flexibly and precisely monitor and gather the performance and status information necessary from the hardware. This is exactly what the **S**mart **E**mbedded **M**anagement **A**gent (SEMA) does for you, the developer and end user.

Time-to-Market (TTM) and Total-Costs-of-Ownership (TCO) are key aspects to producing competitive products. To combine TTM and TCO in a reliable manner, a solid basis is fundamental. To assist in this endeavour, every new ADLINK Technology computer on module (COM) and single board computer (SBC) contains a SEMA Board Management Controller (BMC) device.

Initially designed to for power sequencing, the BMC has evolved with many new, helpful features through the years. Measuring the supply current to get a "snapshot" of the system's power consumption is only one of the new capabilities. Being compatible with the latest EAPI specification greatly simplifies the task of porting existing calls to SEMA to nearly zero!

Connecting the hardware to the operating system is also one of SEMA's key features. The BMC collects all relevant information from the chipset and other sources. Using the SMBus driver, the application layer fetches the data and presents it to the user. ADLINK also provides a ready-made application that shows the data in an easy-to-use graphical interface , suitable for system supervision and troubleshooting.

The SEMA tool can also be used from a command line (currently Windows and Linux are supported). This document describes how the SEMA tool is installed and what kind of information can be displayed.

1.2 SEMA Feature Overview

- Use your existing EAPI calls with SEMA
- Identify part number, serial number, BIOS and BMC version
- Read total operating hours
- Read up-time seconds counter since power on
- Read overall power cycles
- Read current temperature of CPU and board
- Get power-up temperature of CPU and board
- Log Min-/Max-Temperature of CPU and board
- Set, reset or disable the watchdog timer
- Access flash ROM for user data (512B, 1kB)
- Save data in write-protectable flash ROM (128B, fused)
- Get information about system restart events
- Enable, disable and control the LVDS backlight
- Read main current and power consumption
- Log data for spreadsheet calculation programs
- Access Hardware Monitor inputs (e.g. voltage)
- Control I^2C Busses up to 400kHz
- Smart Fan control and fan speed
- Support AT/ATX power supplies
- Save power using the ECO Mode
- Access COM Express GPIOs

2 Installation

2.1 Requirements

- ADLINK board with Board Management Controller (BMC) and SEMA support
- The SEMAGui program requires the Qt Library, version 4.8
    - QtCore4.dll and QtGui4.dll for Windows (included in the package you have downloaded)
    - libQtCore.so.4 and libQtGui.so.4 for Linux (to install Qt, please see the documentation of your Linux distribution)
- Under Windows you may need the Microsoft Visual C++ 2008 SP1 Redistributable Package (x86) to run SEMA. This file may be downloaded directly from Microsoft at: http://www.microsoft.com/download/en/details.aspx?id=5582

2.2 Windows

To access the SEMA microcontroller, a driver is required. The SEMA installation package installs the necessary driver.

2.3 Linux

Under Linux there is no driver necessary. However, to access the SEMA microcontroller, root privileges are mandatory.

Caution

If SEMA is operated in a Linux environment, the standard SMBus driver must be unloaded. Otherwise, this driver and SEMA will conflict.

Inside the kernel configuration, the standard SMBus driver may be found in:

Device Drivers -> I2C Support -> I2C Hardware Bus support

3 Flash Memory Organization

The SEMA microcontroller (BMC) provides 512, or 1024 bytes (depending on platform) for customer data and 128 bytes of write- and clear-protectable flash memory.

These memory areas are independent from the BIOS and are not cleared or restored during BIOS updates, which makes them ideal for storing serial numbers, keys, configuration data and other sensitive or board specific information.

3.1 Memory Map

No memory	0xFFFF 0x8080
Secure data	0x807F 0x8000
No memory	0x7FFF 0x0200 / 0x0400
User data	0x01FF / 0x03FF 0x0000

Table 1: Memory map

3.2 User Data

The “User Data Memory” can be used to store serial numbers, keys and configuration data. By factory default, the first 96 bytes of user data are preconfigured with the following data (table 2). Each value is padded with spaces (ASCII character 0x20), totalling a data length of 16 bytes each.

Address range	Contents
0x0000 – 0x000F	Hardware revision
0x0010 – 0x001F	Serial number
0x0020 – 0x002F	Last Repair Date
0x0030 – 0x003F	Manufacturing Date
0x0040 – 0x004F	Secondary part number (not on all platforms)
0x0050 – 0x005F	Secondary serial number (not on all platforms)
0x0060 – 0x006F	MAC ID

Table 2: Factory default user data memory organization

3.3 Secure Data

The secure data area is empty by factory default and may be protected against write accesses.

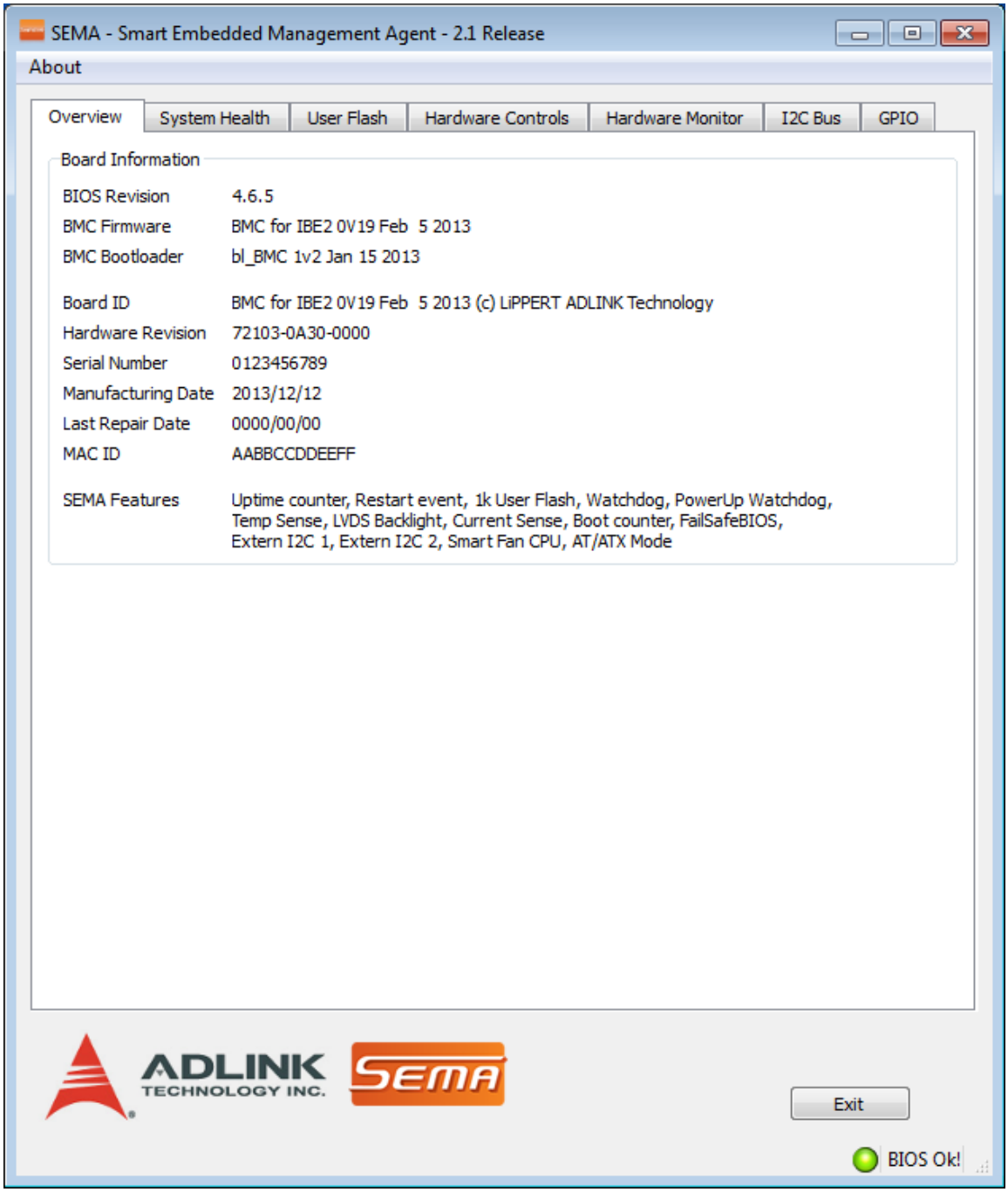
Caution **Once the secure area is write protected, it cannot be unlocked, rewritten or cleared!**

To write protect the secure area, a separate tool is necessary. Please contact your LiPPERT by ADLINK sales representative for further information.

4 Graphical User Interface

The graphical interface is available for Windows and Linux operating systems. To get started, simply run SEMAGui.exe (Windows) or semagui (Linux).

4.1 System Overview Tab



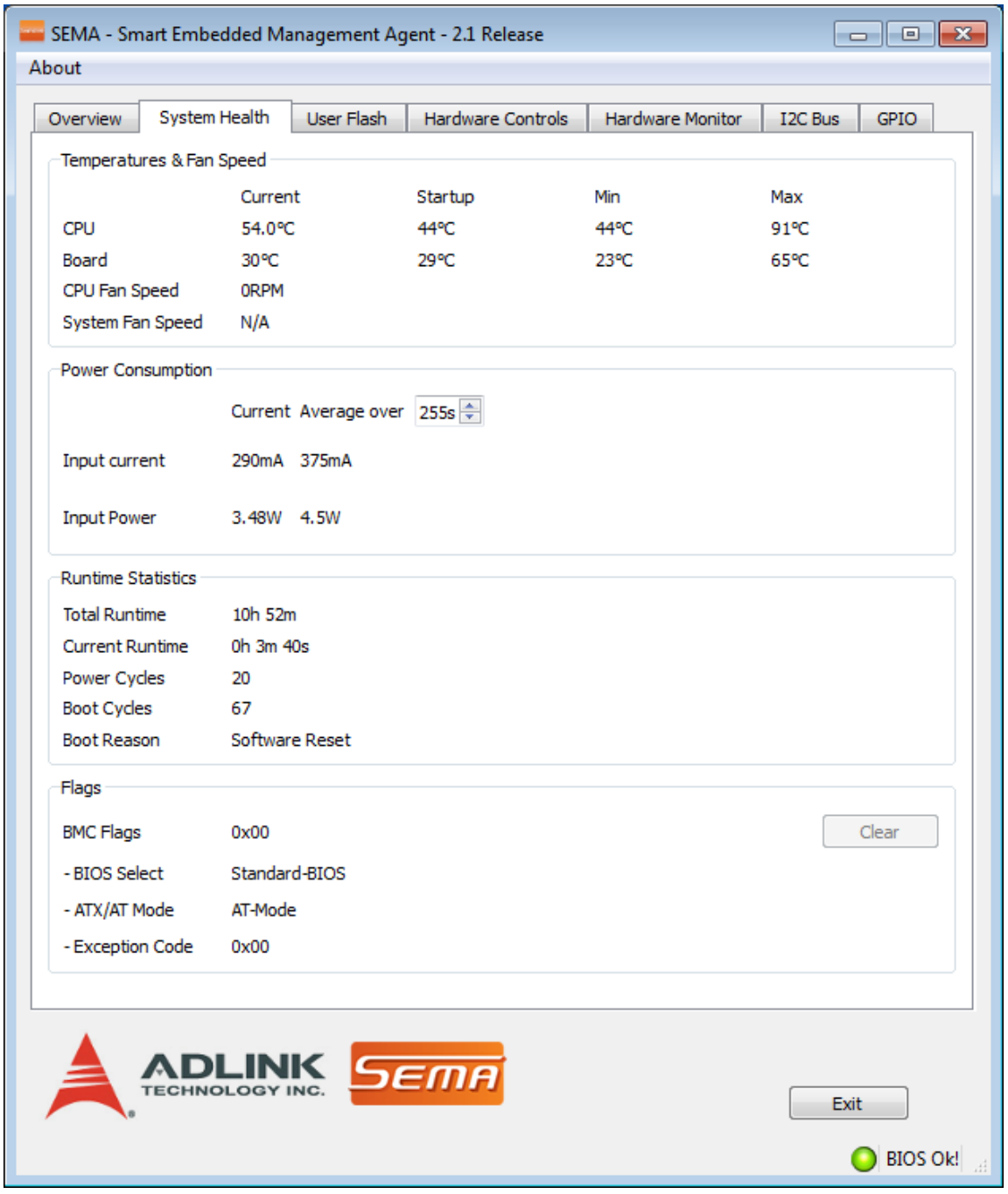
Board Information

This section shows general information about the board, BIOS Version and SEMA features supported. The first three lines show the firmware versions of the board. The BMC versions each contain an ID-string and the respective build date.

The next lines show the board ID, hardware revision, serial number, manufacturing/repair date and MAC ID of the board.

The last line shows all supported SEMA features for this board.

4.2 System Health Tab



Temperatures & Fan Speed

This section displays the current, start-up, minimum and maximum temperatures of the CPU and environment (board). The data is displayed in Celsius and updated every second.

Additionally, the current CPU and system fan speed is displayed in RPM.

Please note that not all platforms supply all information shown above. If so, "N/A" will be displayed.

Power Consumption

The power consumption section displays information about the main power supply. These readings are displayed in milliamperes (mA) for current, and watts (W) for power consumption, and are updated every second.

To filter out any spikes and to get average values over a certain interval of time, the number of seconds/samples can be selected in range from 0 to 255.

Please note that the averaging takes place within the graphical user interface and not within the BMC itself. The GUI software will collect the selected number of samples and present an average of those values.

Please note that not all platforms may supply the necessary current sensors. If so, "N/A" will be displayed and the section will be grayed out.

Runtime Statistics

Total	The total uptime of the system in hours and minutes
Current	Uptime since last boot in hours, minutes and seconds
Power cycles	Number of power cycles
Boot cycles	Number of HW/SW-Reset and successful power-ups
Boot reason	The event that caused the last reboot (e.g. power loss, power down, HW reset, etc.)

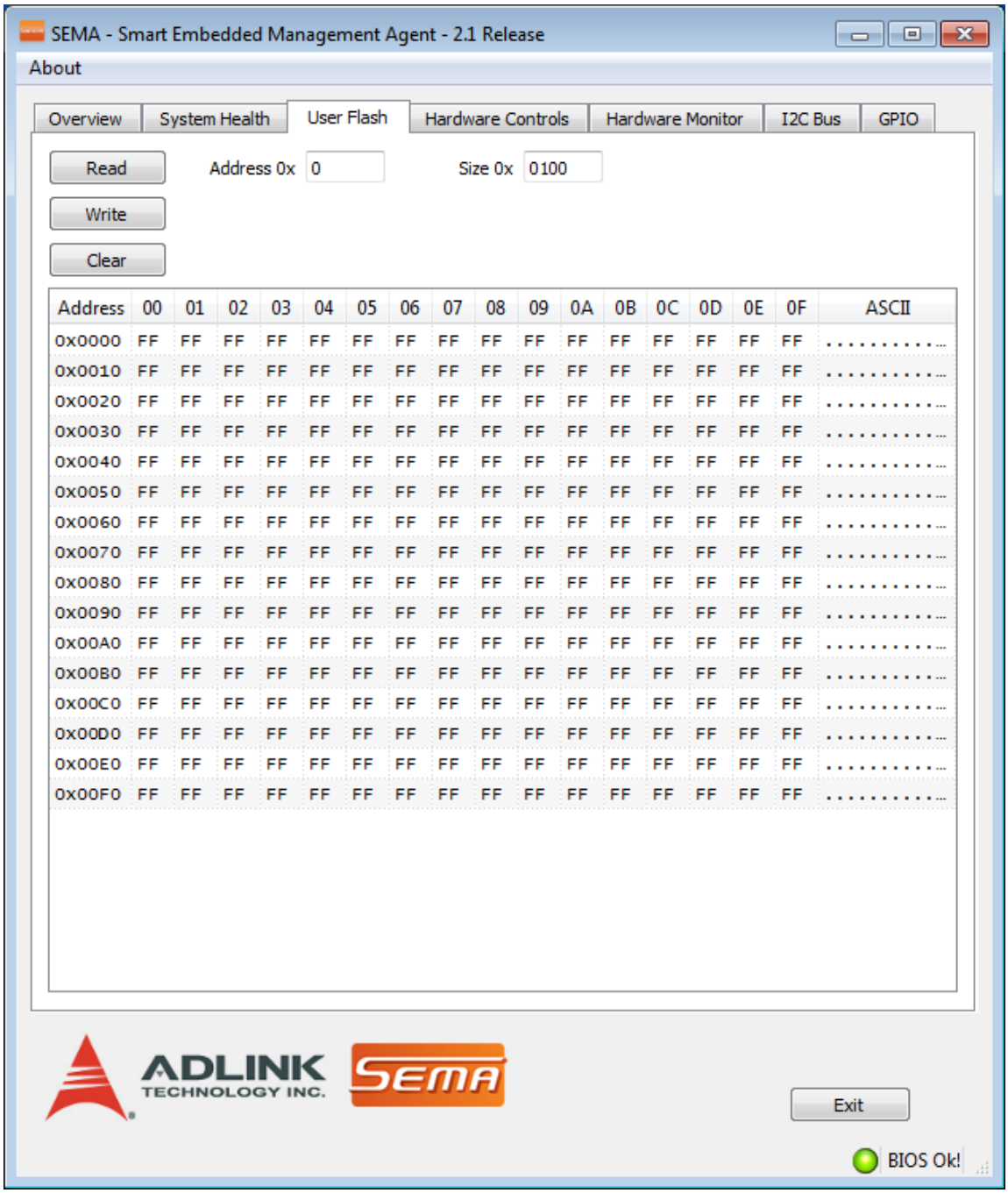
Flags

BMC flags display information about the internal status of the BMC. The content is board specific. Please refer to the technical manual of your board, chapter “SEMA functions”.

Additionally, this section provides information about:

- the currently active BIOS
- if the system is in AT or ATX mode
- and some Exception Code if applicable

4.3 User Flash Memory Tab



Memory

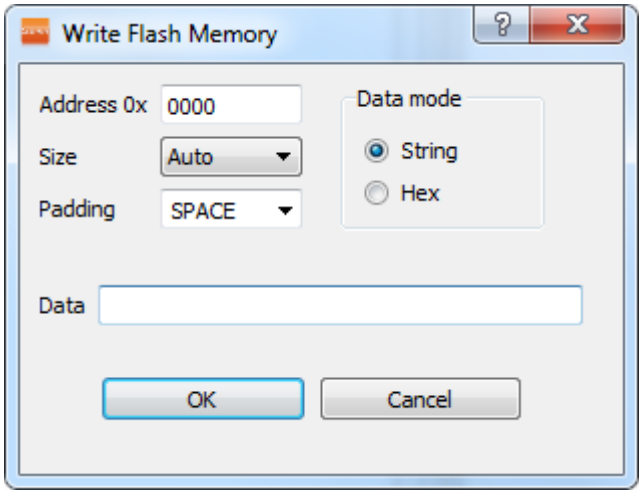
For memory organization, refer to Chapter 3, Flash Memory . address and size values are entered as hexadecimal values. Valid start addresses are multiples of 16 within the range: 0x0000-0x01F0 for 512 byte variants and 0x0000-0x03F0 for 1024 byte variants. The size can be any multiple of 16 bytes (up to the total memory size).

Example: To display the first 32 bytes from user flash memory, enter “0000” as address, “20” (hex 20 equals decimal 32) as size and press the [Read] button. The memory content is transferred from the BMC and displayed in the output window.

Each line contains the address of the first byte of this line, followed by 16 data bytes (displayed as hexadecimal values, prefixes “0x” omitted for better readability) followed by the corresponding printable ASCII characters.

The [Write] button opens the Write memory dialog box.

Write Memory



Address

The address is entered as hexadecimal value within the range of 0x0000..0x01FC (respectively 0x03FC for 1k board variants).

Size

The size can be any multiple of four, up to 32 bytes or “Auto”.

In Auto-mode, the number of bytes to write is determined by the amount of data entered in the data field below (max 32 bytes).

Padding

If the data length is not a multiple of four, the required number of pad (or fill) characters is appended.

In String-mode, a valid pad character can be any printable ASCII character or one of the predefined pad characters:

- SPACE character (ASCII character 0x20)
- NULL character (ASCII character 0x00)

In Hex-mode, any eight bit hex value (00..FF) is valid.

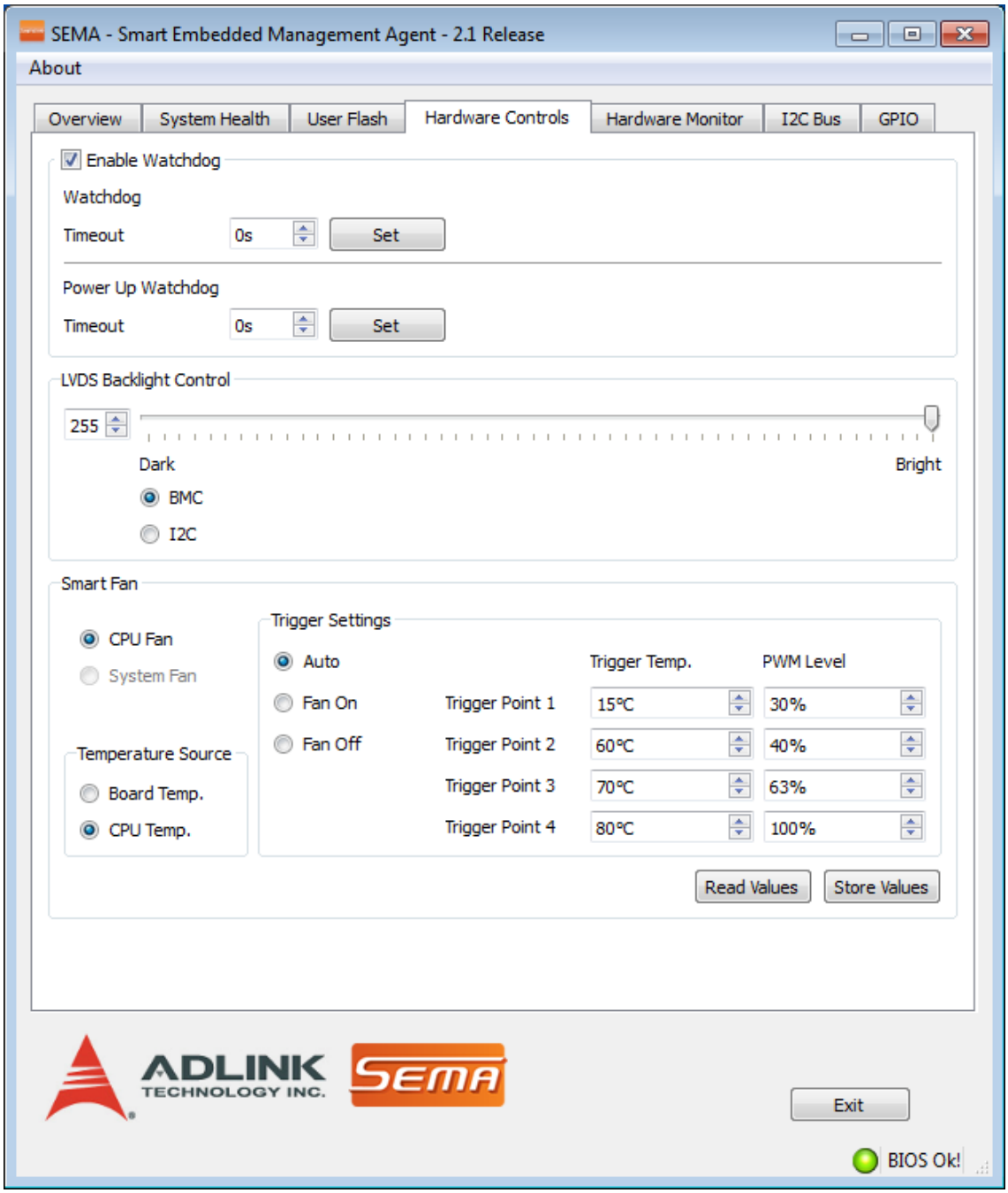
Data Mode

Currently supported modes are String and Hex.

String mode: The entered text (up to 32 characters) is converted to a byte stream. Please note that no termination character is appended.

Hex mode: Up to 64 hex digits (0..9, A..F) can be entered, resulting in up to 32 bytes of data.

4.4 Hardware Control Tab



Watchdog

The watchdog timeout value is given in seconds and can be set to 1-65535 seconds. A setting of 0 seconds will disable the watchdog. When enabled, be sure to trigger the watchdog repeatedly within the timeout period otherwise a system reset will be initiated.

Please note that not all platforms support watchdog functionality. If unavailable, the watchdog section will be grayed out.

Notes:

- When using the watchdog feature, be sure to have all partitions mounted read-only. Otherwise file system corruption and data loss may occur.
- It is NOT advisable to use the watchdog feature under Windows since it is recommended to restart a Windows environment using a safe shutdown procedure.

Backlight Control

Enables or disables the backlight of a display connected via LVDS. The backlight value can be adjusted by a slider or entered as a numerical value. Allowed values range from 0 to 255.

Please note that not all platforms support the necessary circuitry. If unavailable, this section will be grayed out.

The radio boxes select if the backlight is controlled by the BMC or the LVDS I2C interface.

Smart Fan

The Smart Fan section provides control of either the CPU or the system fan (if applicable). Which one is accessible can be also seen from SEMA Features item in the System Overview Tab (p. 7) or by using the "sema cap" command line (p. 20).

Both fans can have a different temperature source, independent of the other. The temperature source defines which temperature sensor will be used for calculating the RPM.

If the fan is set to "Auto", no other settings need to be made.

If set to “Off” the fan is turned off completely.

If set to “On” the fan runs at maximum RPM.

“Trigger Temp” and “PWM Level” determine the characteristic curve of the fan control. According to the settings in the screen capture shown above, the fan will:

- be turned off when the temperature drops below 15°C
- run with a PWM level of 30% if the temperature is above 15°C but below 60°C
- run with a PWM level of 40% if the temperature is above 40°C but below 70°C
- run with a PWM level of 63% if the temperature is above 48°C but below 80°C
- run with a PWM level of 100% if the temperature exceeds 55°C

If the temperature drops below one of the trigger points, the appropriate lower PWM value will be taken.

The button “Restore Values” reads out the current values of the board management controller and displays them in Trigger Settings.

The button “Save Values” stores the new settings to the board management controller.

4.5 Hardware Monitor Tab

The hardware monitor shows the “Power Consumption” and “Temperatures” information from the “System Health” tab, but in a graphical way in order to provide a better overview.

Enable Logging

When logging is enabled, all values that are queried every second (temperatures, power consumption etc.) are written to a log file. At startup, the logging feature is disabled by default. To enable logging, check the Enable Logging checkbox in the lower left corner and select a file. New data will be appended to the selected file; existing data will not be overwritten. Logging stops when the Enable Logging checkbox is unchecked or when the program is closed. Logging is suspended during the "File open …" dialog.

The data is written as plain ASCII text in tab-separated columns and therefore can easily be imported into any spreadsheet calculation program or other data processing tool. The first line written to the log file contains captions indicating the content and the respective unit of the data to be written below. If any particular data is not available (e.g. no sensor available), the corresponding column will be skipped.

The table below lists currently available items:

Caption	Content and unit
Total[m]	Total uptime in minutes
PwrUp[s]	Uptime since last power up in seconds
Tcpu[°C]	CPU temperature in Celsius
Tbrd[°C]	Board temperature in Celsius
Tcpumin[°C]	Minimal CPU temperature in Celsius
Tcpumax[°C]	Maximal CPU temperature in Celsius
Tbrdmin[°C]	Minimal board temperature in Celsius
Tbrdmax[°C]	Maximal board temperature in Celsius
Imp[mA]	Main power current in milliamperes
Pmp[W]	Power on main power current rail in watts
n[1]	Number of samples used for filtering
Impf[mA]	Filtered main power current in milliamperes
Pmpf[W]	Filtered power on main power current rail in watts
FanSpdSys[RPM]	CPU-Fan speed in revolutions per minute
FanSpdCPU[RPM]	System-Fan speed in revolutions per minute
'VoltageName'[V]	For each monitored voltage in volts

Table 3: Logfile columns (captions and contents)

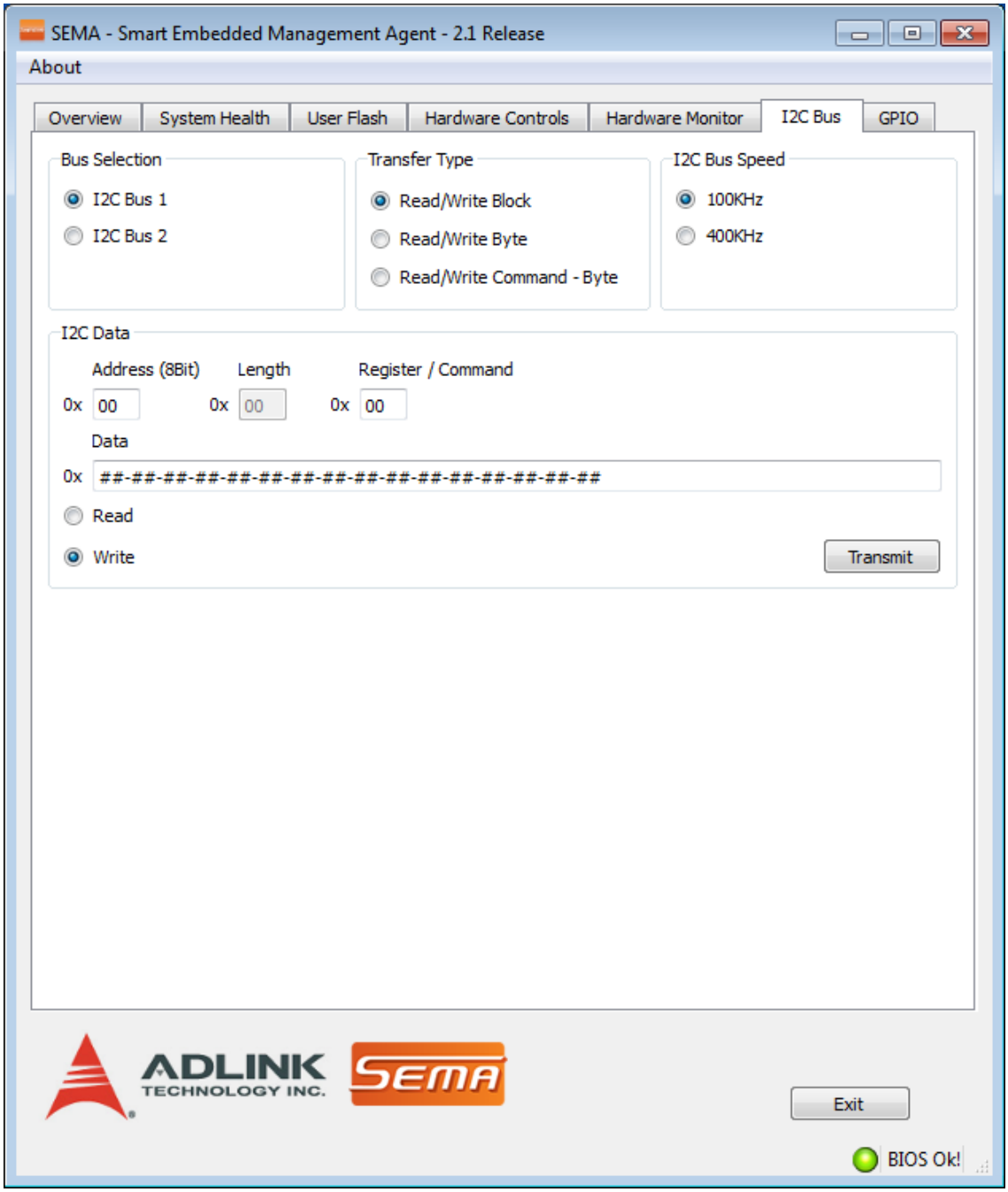
4.6 I²C Bus Tab

The board management controller (BMC) can access up to two external I²C busses.

Byte and block access for read and write are implemented. Both common bus speeds of 100kHz and 400kHz are supported.

To read data from or write data to the BMC an address (8-bit) must be given as well as the amount of data to be transferred (length) plus the register offset.

Data then has to be handed over in hex values forming a hex string (as shown below).



Transfer Types

The different transfer types will result in the following I2C bus activity.

Read/Write Block

- Write: Start + Address/Write + Register /Command + Length + Data[1] + Data[2] + ... + Data[Length] + Stop
- Read: Start + Address/Write + Register /Command + Start + Address/Read + Length + Data[1] + Data[2] + ... + Data[Length] + Stop

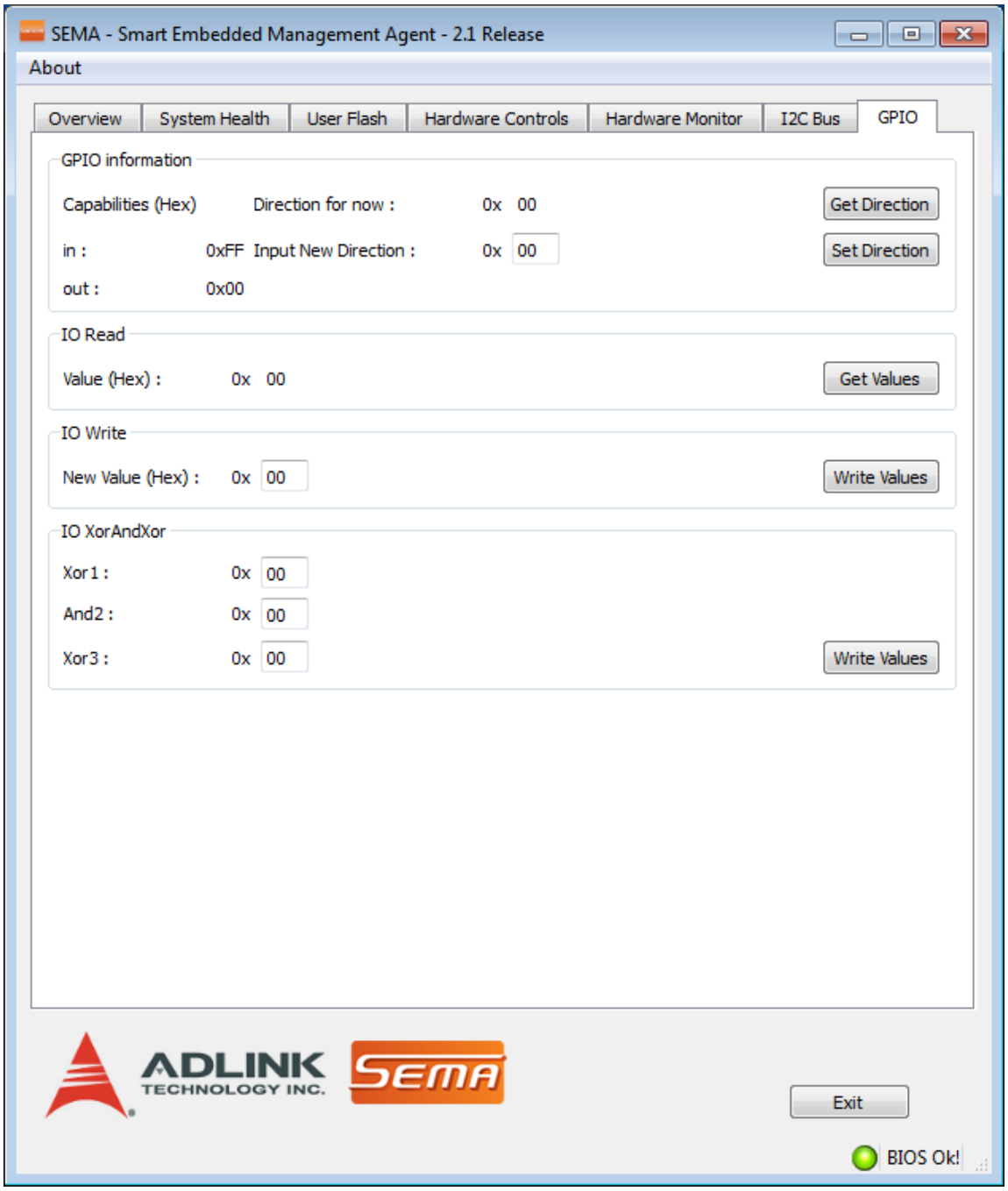
Read/Write Byte

- Write: Start + Address/Write + Data[1] + Stop
- Read: Start + Address/Read + Data[1] + Stop

Read/Write Command Byte

- Write: Start + Address/Write + Register /Command + Data[1] + Stop
- Read: Start + Address/Read + Register /Command + Data[1] + Stop

4.7 GPIO Tab



The GPIO Tab let you control the 8 GPIO provided by the COM Express board.

GPIO information

This section controls the GPIO direction. Each bit in the shown byte represents a GPIO. To set a GPIO for output the bit must set to 0, for input it must be 1. The “Get Direction” button read the current configuration and the “Set Direction” one sets the configuration to the value entered in the “Input New Direction” field.

IO Read

This section shows the current input values for all GPIOs. GPIOs configured as output will show their current output value.

IO Write

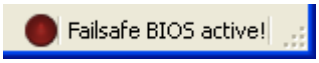
This section sets the outputs. GPIOs configured as input will not be affected.

IO XorAndXor

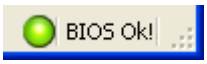
This section reads the GPIO input and applies the logic operation in order of the input fields. The result is set as output value.

4.8 Status Bar

The Status Bar of the SEMA windows displays a flashing red LED if the board has booted using the Failsafe Bios. This may occur if the main BIOS has been corrupted.



Failsafe BIOS active



Boot from normal BIOS

5 Command Line Interface

The SEMA command line interface is available for both Linux and Windows versions.

For a detailed description of the supported command line options, refer to the following sections.

5.1 General Options

`sema help`	Displays help screen with a brief list of available options

5.2 Get System Information

`sema` <options> [<option(s)>]

Supported options are:

`version`	Displays the board ID and version string
`hwrev`	Displays the board hardware revision
`serialno`	Displays serial number
`repair`	Displays last repair date
`date`	Displays manufacturing date
`mac`	Displays the MAC ID
`2hwrev`	Displays secondary board hardware revision (if present)
`2serialno`	Displays secondary serial number (if present)
`pwrcycles`	Displays the number of power cycles
`uptotal`	Displays total uptime
`upboot`	Displays uptime since last boot
`tempcpu`	Displays the current CPU temperature
`tempcpulog`	Displays the logged minimum and maximum CPU temperatures
`tempboard`	Displays the current board temperature
`tempboardlog`	Displays the logged minimum and maximum board temperatures
`tempboot`	Displays the CPU and board temperature at boot
`restart`	Displays the last restart event (e.g. “Hardware Reset”, “Power loss”, etc.)
`exception`	Displays exception code
`mpcurr`	Displays average main supply current (see note below)
`bmc`	Displays BMC bootloader and firmware ID strings including date codes and flash file system header status
`cap`	Displays BMC capabilities
`all`	Displays all available information

Notes:

- On some platforms, not all options may be available.
- All temperatures are displayed in Celsius.
- Any combination of parameters is allowed. If only one value is requested (e.g. pwrcycles) the output contains only the requested value without any descriptive text. This reduces the effort to integrate the provided information in script or batch files.
- Main supply current sampling: The BMC constantly samples the main supply current every 250ms and stores the last four values in an internal circular buffer. When the main power is queried, the largest of the four values is discarded and the average of the lower three samples is calculated and returned. The reason why the largest value is discarded is that the invocation of the query command itself may wake up the CPU from any power saving or sleep state and therefore may result in a higher current reading.
  Please keep in mind that this algorithm only yields a rough estimation of the average main supply current consumption over the last seconds and is not to be misunderstood as a precisely measured value.

5.3 Watchdog

```
sema wdt <seconds>
```

The SEMA command line interface can be used to set, retrigger and disable the watchdog timer.

The watchdog timeout value is given in seconds and can be set to 1-65535 seconds. A setting of 0 seconds will disable the watchdog. When enabled, be sure to trigger the watchdog repeatedly within the timeout period otherwise a system reset will be initiated.

Notes:

- When using the watchdog feature, be sure to have all partitions mounted read-only. Otherwise file system corruption and data loss may occur.
- It is NOT advisable to use the watchdog feature under Windows since it is recommended to restart a Windows environment using a safe shutdown procedure.

5.4 Memory Access

The SEMA command line interface can be used to access flash memory (read and write access).

Read Access

```
sema read <adr> <num>
```

Reads data from flash memory.

`<adr>` Start address

Valid start addresses: 0x0000..0x01FC/0x03FC (prefix “0x” can be omitted)

`<num>` Number of bytes (must be a multiple of four)

Write Access

```
sema write <mode> <adr> <num> <data> [<pad>]
```

Writes data to flash memory.

`<mode>` Data mode (a=ASCII, h=HEX)

`<adr>` Start address

Valid start addresses: 0x0000..0x01FC/0x03FC (prefix “0x” can be omitted)

`<num>` Number of bytes (up to 32 bytes, must be a multiple of four)

<data>	Data to write to memory (prefix “0x” can be omitted in hex mode)
<pad>	If <data> contains less bytes than <num>, memory is padded with <pad>

5.5 Backlight Control

The SEMA command line interface can be used to enable, disable or adjust LVDS backlight brightness.

```
sema bkget
```

Get current backlight setting. The returned value will be in range 0 (backlight off) to 255 (max. brightness).

```
sema bkset <value>
```

Set new backlight setting. Valid values are in range of 0 (backlight off) up to 255 (maximal brightness).

5.6 System Monitor

```
sema fanspeed
```

Displays the current speed of the CPU fan in rpm.

```
sema voltages
```

Prints a list of all monitored voltages, followed by the name of the voltage (e.g. CPU-Vcore, Vin) or the nominal value (e.g. 3.300 for the 3.3V rail).

```
sema volt <channel>
```

Displays only one single voltage, selected by parameter <channel>. In this case, no additional text is added for easier parsing in custom application.

5.7 Smart Fan Control

An ADLINK board may feature up to two software configurable Smart Fans. The command line interface can be used to read out and alter the current control settings. Each smart fan uses its dedicated control table that contains a set of temperature vs. fan speed combinations. The temperatures must be in strictly increasing order.

If the temperature rises above the next higher value, the fan speed is set to the associated speed value of the higher temperature value. If the temperature drops below the next lower value, the fan speed is set to the associated speed value of the lower temperature value. To minimize oscillation between two speed values, a hysteresis of two degrees Celsius is used.

```
sema getfan <idx>
```

Output will be in the following format:

```
sema setfan <idx> <t1> <s1> <t2> <s2> <t3> <s3> <t4> <s4>
```

"idx" is the fan index. Usually, index 0 controls the CPU fan, index 1 controls the system fan.

Each pair of t* / s* values forms a node of the control table. All temperatures (t1 .. t4) must be in strictly increasing order.

5.8 Failsafe-BIOS Support

If the board features Failsafe-BIOS support, the SEMA command line interface can be used to display the currently selected BIOS or to select a different one. The following BIOS options are available:

0 Main (or standard) BIOS
1 Failsafe BIOS: This will be selected if the main BIOS has become corrupted
2 External BIOS (not available on all platforms)

To read the currently selected BIOS or to select another one, use:

`sema getbios`	Displays currently selected BIOS
`sema setbios <num>`	Selects BIOS (<num> may be 0, 1 or 2)

5.9 Accessing I²C Devices on SMBus

The i2c* command family enables access to I²C devices that are connected to the SMBus. Several access modes are available to allow communication with different types of devices (e.g. EEPROMs, temperature sensors, GPIO expanders). For selection of the correct addressing scheme, please refer to the device's documentation.

The I²C slave device address <adr> is expected in 8-bit hex format with the 7-bit address in the upper seven bits and the lowest bit set as read (1) / write (0) bit accordingly.

Write access to a I²C device on SMBus

`sema i2cwb  <adr> <d>`

Write single byte (8-bit) data <d>.

`sema i2cwbb  <adr> <c> <d>`

Byte-wide (8-bit) command or register offset <c>, single byte (8-bit) data <d>.

`sema i2cwbw  <adr> <c> <d0> <d1>`

Byte-wide (8-bit) command or register offset <c>, word-wide (16-bit) data, split into two bytes <d0> and <d1>.

`sema i2cwwb  <adr> <c0> <c1> <d>`

Word-wide (16-bit) command or register offset (split into two bytes <c0> and <c1>), byte-wide (8-bit) data.

Read access to a I²C device on SMBus

`sema i2crb  <adr>`

Read single byte (8-bit).

`sema i2crbb  <adr> <c>`

Byte-wide (8-bit) command or register offset <c>, read single byte (8-bit) data.

`sema i2crbw  <adr> <c>`

Byte-wide (8-bit) command or register offset <c>, read word-wide (16-bit) data.

`sema i2crwb  <adr> <c0> <c1>`

Word-wide (16-bit) command or register offset (split into two bytes <c0> and <c1>), read single byte (8-bit) data.

5.10 Accessing I²C Devices on External I²C Busses

Some boards offer the possibility, to access up to two additional I²C busses (besides the SMBus). If external I²C busses are available, the corresponding bits (“Ext. I2C #1” and “Ext. I2C #2”) in the capability feature vector (accessed and displayed by the "sema cap" command line, see p. 20) are set to 1.

The external bus number <bus> must be 1 or 2.

The I²C slave device address <adr> is expected in 8-bit hex format with the 7-bit address in the upper seven bits and the lowest bit set as read (1) / write (0) bit accordingly.

Write access to a I²C device external I²C busses

```
sema i2cwblock <bus> <adr> <reg> <data>
```

Block-write to byte-wide (8-bit) command or register offset, data is expected as a string of hex characters of up to 32 bytes (e.g. <data>=”BADEAFFE0815” results in six bytes to be written).

```
sema i2cwbyte <bus> <adr> <data>
```

Byte-write to byte-wide (8-bit) command or register offset. Parameter <data> may be specified in decimal (0..255) or in hexadecimal representation (0x00..0xFF).

```
sema i2cwregbyte <bus> <adr> <reg> <data>
```

Byte-write to byte-wide (8-bit) command or register offset <reg> . Parameter <data> and <reg> may be specified in decimal (0..255) or in hexadecimal representation (0x00..0xFF).

Read access to a I2C device external I2C busses

```
sema i2crblock <bus> <adr> <reg> <num>
```

Block-read from byte-wide (8-bit) command or register offset, data is returned as a string of hex characters of up to 32 bytes (up to 64 hex digits).

```
sema i2crbyte <bus> <adr>
```

Byte-wide (8-bit) data read access.

```
sema i2crregbyte <bus> <adr> <reg>
```

Byte-wide (8-bit) data read access to command or register offset <reg> . Parameter <reg> may be specified in decimal (0..255) or in hexadecimal representation (0x00..0xFF).

5.11 GPIO Access

Some COM Express boards offer the possibility to access 8 GPIOS.

```
sema gpioread
```

Read input state of COM Express GPIOs.

```
sema gpiowrite <data>
```

Write <data> to COM Express GPIOs . Parameter <data> may be specified in decimal (0..255) or in hexadecimal representation (0x00..0xFF)

```
sema gpiogetdir
```

Read the current GPIO direction if a bit is set the GPIO works as input, otherwise as output.

```
sema gpiosetdir <dir>
```

Set the GPIO direction. A set bit marks the GPIO as Input, otherwise as output. Parameter <dir> may be specified in decimal (0..255) or in hexadecimal representation (0x00..0xFF).

5.12 Firmware Update

Installs updates to the BMC.

```
sema update <filename> [<delay> [down]]
```

`<filename>`	File that contains SBC firmware update
`<delay>`	Reset delay after finishing update procedure
`down`	The operating system will be shut down after finishing update procedure

After a firmware update, a BMC reset is mandatory and triggered by the BMC itself. If the delay option is not used, the default reset delay is 2 seconds.

Caution **When using the update feature, be sure to have your operating system in a safe state, all files closed and all partitions mounted read-only. Otherwise file system corruption and data loss may occur.**

If an update is necessary under Windows, it is strongly recommended to close all other applications before starting the update and use the <delay> and "down" options. Chose a delay time (0..255 seconds are selectable) that is long enough to safely shut down Windows. The board will stay powered up until the BMC restarts after the <delay> period has expired.

6 EAPI - Embedded Application Programming Interface

The EAPI specification provides a common programming interface for hardware features found on COM Express systems or other embedded form factors. The SEMA implementation of the EAPI library follows revision 1.0 of the PICMG EAPI Specification.

6.1 Installation

The EAPI library is available for Linux and Windows. The Windows library is called EAPI_1.DLL. The Linux version is called libEApiLIB.so.1.0. The number indicates the specification version of the EAPI implementation.

Windows

For Windows systems the SEMA Package provides the EAPI_1.dll. Copy this library in the working directory of your program or tool. For a global installation, copy the library to the C:\Windows\system32 directory.

Linux

Copy the libEApiLIP.so.X.Y file to /usr/lib/ directory. Then run ldconfig to create necessary symbolic links. For this step you will need root rights.

Note: Your program will need root rights to use the EAPI library.

6.2 Supported EAPI Functions

The ADLINK EAPI Library implements all EAPI functions, although not all features are supported by all hardware.

EAPI Function	Supported on ADLINK Hardware
EAPI Info	
EApiBoardGetStringA()	✓
EApiBoardGetValue()	-✓ Missing Features or only on supported Boards: • Chipset Temp • Voltage monitoring • Fan speed monitoring
EAPI Backlight	LVDS Backlight
EApiVgaGetBacklightEnable()	✓
EApiVgaSetBacklightEnable()	✓
EApiVgaGetBacklightBrightness()	✓
EApiVgaSetBacklightBrightness()	✓

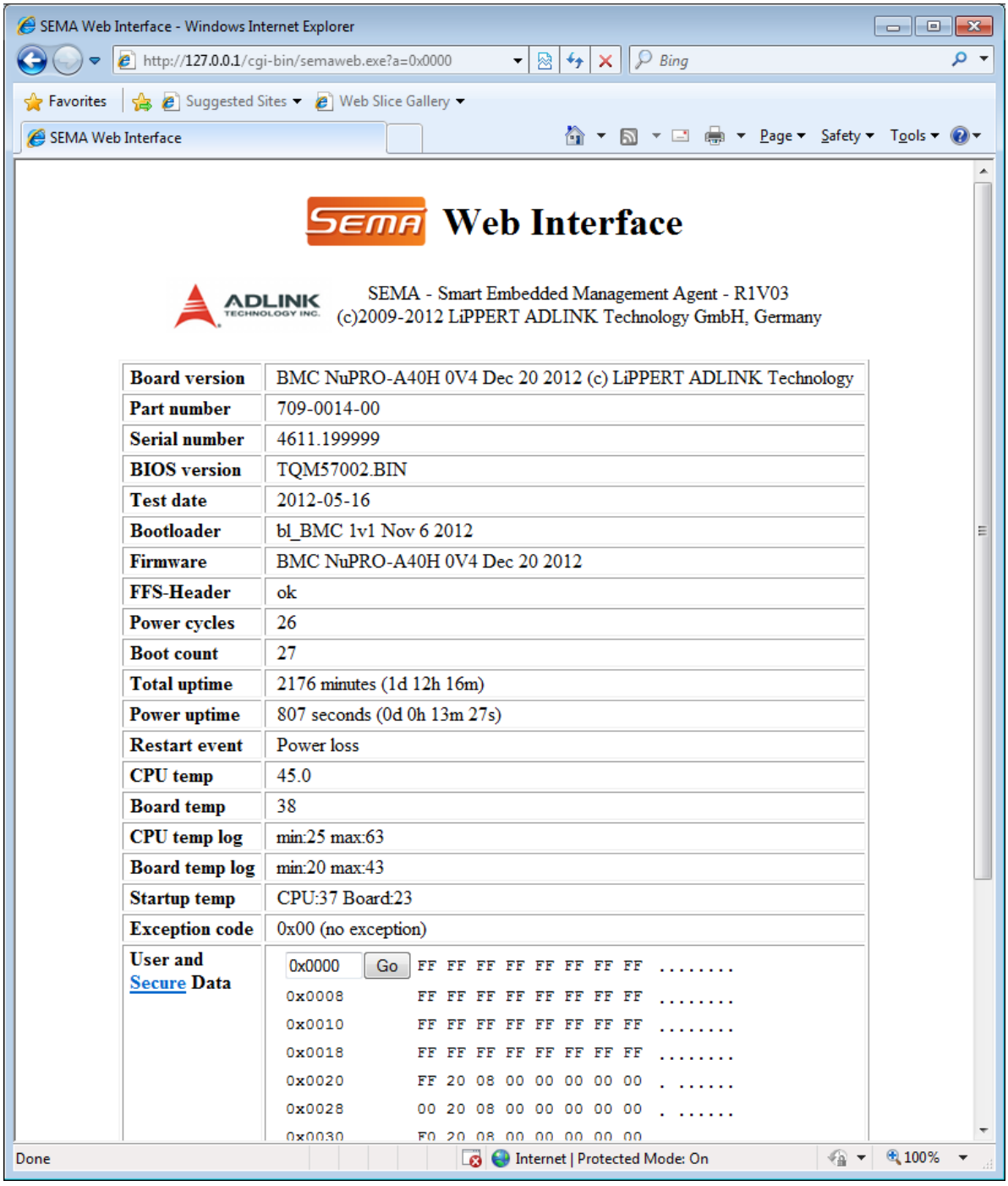
EAPI Internal Storage	Only User Data Memory
EApiStorageCap()	✓
EApiStorageAreaRead()	✓
EApiStorageAreaWrite()	✓
EAPI Watchdog	No Event Timeout
EApiWDogGetCap()	✓
EApiWDogStart()	✓
EApiWDogTrigger()	✓
EApiWDogStop()	✓
EAPI I2C	Only 7Bit addresses
EapiI2CGetBusCap()	✓
EapiI2CWriteReadRaw()	✓
EapiI2CReadTransfer ()	✓
EapiI2CWriteTransfer()	✓
EApiI2CProbeDevice()	-
EAPI GPIO	
EApiGPIOGetDirectionCaps()	-
EApiGPIOGetDirection()	-
EApiGPIOSetDirection()	-
EApiGPIOGetLevel()	-
EApiGPIOSetLevel()	-

Table 4: EAPI Function Overview

7 SEMA Web Interface

7.1 Overview

The SEMA web interface allows remote access of all main SEMA functions via any browser. To achieve this, a CGI program called SEMA Web Interface (or SEMAWeb) is used to connect the SEMA command line tool to a web server.



7.2 Security

Caution **Running a web server may introduce several and severe security risks for your system and network!**

Caution **Exposing SEMA on a web page means anyone with a browser can use it. Among other things this means anyone can reboot the board at any time. It is recommended to install the web interface on a trusted network only.**

If this installing the web interface on a trusted network is not possible, you may use an .htpasswd file to protect the web interface with passwords. If there is concern about transmitting plain text over the Internet, use digest authentication or enable encryption with HTTPS. This may require using a more feature rich server.

If there are more significant security concerns over whether or not the SEMAWeb interface might is the right solution for your application, we recommend calling the SEMA command line tool directly via an SSH connection. This is safer, much easier to set up properly, and will also save resources, especially if you use a lightweight server like **Dropbear** instead of the widely known **OpenSSH**. An SSH server additionally provides secure shell access for maintenance and makes updating your application possible via SCP for example.

7.3 Installation

Windows XP or Windows 7

To install the SEMAWeb utility under Windows XP or Windows 7, run the provided installation program: Sema(rev.1705).msi. This will install the SEMAWeb tool and the Internet Information Server (IIS) as web server. The setup process should be self explainatory. The installation will allow you to set an exception for the Windows Firewall.

If you wish to use another web server, please refer to the documentation provided by the appropriate server and its content regarding installing a CGI program, such as SEMAWeb.exe.

After installation, SEMAWeb is available using the following URL:

http://your.ip.address/cgi-bin/semaweb.exe

Note *Windows XP might need the Windows installation CD to finish the IIS installation!*

Note *It might be necessary also to change the CGI-BIN access user. Run **iis.msc.** Select from the website-tree the cgi-bin node, right click and select properties. Next, select the Directory Security tab and press Edit. Select a user for anonymous access which has the access rights to run the **SEMA** tool.*

Note *Windows Embedded Standard 7 may be supported. As a minimum requirement, the IIS7 package and its dependencies must be included in the Windows Embedded Standard 7 image. Please select Windows 7 as OS if queried by the installer.*

Linux

Web Server

The first component is a web server, to accept incoming network connections and serve the content to the user's browser. A server already running on the supervised system for some other purpose may be co-used, otherwise you will need to install one. It must support CGI, but virtually all do, so you may pick any server you feel comfortable with. Apache is full-featured and widely known, but clearly excessive for this purpose. We'll be using **thttpd** for the sake of this documentation.

You can download **thttpd** from its home page, but it is easier to install the appropriate software package for your distribution. For example if you are running Debian or Ubuntu, simply run:

```
apt-get install thttpd
```

Other distributions may use installpkg, yum, yast or rpm. Please refer to your distribution's manual for more information.

Thttpd's configuration file can often be found in /etc/thttpd/thttpd.conf, open it with any text editor to adapt it to your needs. We assume the following settings:

```
Nochroot
dir=/var/www
cgipat=/cgi-bin/*
```

Chroot means the web server will restrict all access to /var/www (whatever dir is set to) or below. This increases security, but means that in addition to the CGI programs, any referenced libraries, device files, etc., also need to be copied to this directory. This sandbox or chroot jail is rather tricky to set up and beyond the scope of this document. For example, see *man thttpd* to avoid cutting the server off from its own logfile. We recommend setting **nochroot** to avoid this obstacle at least initially until you get the SEMAWeb interface up and running. Then you may try and re-enable it if you want to.

Dir specifies the base directory for all web content, the "/" directory as seen by the browsers. If you installed a distribution-specific package then do not change this setting. Just know to use the appropriate path when reading this documentation.

cgipat is the pattern applied by the server to determine if a file is a CGI executable (alternatives are "*.cgi" or "/cgi-bin/*.cgi".) This option must be set and the SEMAWeb binary must match it.

If the web server is not running already or if you changed its configuration file then (re)start it with:

```
/etc/init.d/thttpd restart
```

By entering the supervised system's IP address in any browser's address bar you should now be able to connect from another computer to show whatever is already provided in /var/www/. If you are unable to, check if a firewall on the supervised system is blocking incoming(!) access to the HTTP port 80 or test the basic network setup with ping.

SEMAWeb

The second component is responsible for calling the SEMA command line tool and formatting its output as an HTML page, which it then passes to the web server. semaweb can be compiled from its source file semaweb.c with:

```
gcc -Wall -O2 -s -o semaweb semaweb.c
```

This is only necessary after changing one of the few configuration options #defined at the beginning of the source file. Install the resulting binary semaweb in the /var/www/cgi-bin/ directory where the web server expects to find CGI executables.

Also copy the logo image file to /var/www/img/ as specified by the "#define LOGO_IMAGE" line near the start of the source file. You may have to create the img/ directory first. Direct your browser to:

http://your.ip.address/cgi-bin/semaweb

Now the SEMA Web Interface headline and logo should be displayed, likely next to an error message about sema not being found. If not, check the files' permissions and the web server's CGI setting. semaweb must be executable by whatever user the web server runs as; often this is a dedicated user with minimum rights, named www-data or nobody.

SEMA

The last component is the SEMA executable which performs the actual hardware access. SEMA_COMMAND in semaweb.c defines how it is run. By default, this is just "sema" without a specific location, meaning sema will have to be installed somewhere in the standard search path to be found. For example, copy it to /usr/bin/ or specify the full path. It must also be executable by the web server user.

Next, you will see displayed something similar to "IO Access: Operation not permitted" and "Admin privileges needed" in your browser. This means that SEMAWeb succeeded in calling SEMA, but the latter did not have the rights to access the hardware. To make it work, the SEMA executable itself must run as root, even when being called by the unprivileged web server's user. One way to achieve this is to make SEMA a setuid root executable:

```
chown root:root sema
chmod 4755 sema
```

Alternatively you can set SEMA_COMMAND to "sudo sema" and add a suitable entry for the web server's user to /etc/sudoers. This would limit sema to this user, but would improve security only marginally, as logged in users with shell access are not the issue here

.

8 Getting Service

Contact us should you require any service or assistance.

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
新北市中和區建一路166號9樓
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110, San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 上海市浦东新区张江高科技园区芳春路300号 (201203)
300 Fang Chun Rd., Zhangjiang Hi-Tech Park,
Pudong New Area, Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology Beijing
Address: 北京市海淀区上地东路1号盈创动力大厦E座801室(100085)
Rm. 801, Power Creative E, No. 1, B/D
Shang Di East Rd., Beijing, 100085 China
Tel: +86-10-5885-8666
Fax: +86-10-5885-8625
Email: market@adlinktech.com

ADLINK Technology Shenzhen
Address: 深圳市南山区科技园南区高新南七道 数字技术园 A1栋2楼C区 (518057)
2F, C Block, Bldg. A1, Cyber-Tech Zone, Gao Xin Ave. Sec. 7,
High-Tech Industrial Park S., Shenzhen, 518054 China
Tel: +86-755-2643-4858
Fax: +86-755-2664-6353
Email: market@adlinktech.com

LiPPERT ADLINK Technology GmbH
Address: Hans-Thoma-Strasse 11, D-68163, Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

ADLINK Technology, Inc. (French Liaison Office)
Address: 15 rue Emile Baudot, 91300 Massy CEDEX, France
Tel: +33 (0) 1 60 12 35 66
Fax: +33 (0) 1 60 12 35 66
Email: france@adlinktech.com

ADLINK Technology Japan Corporation
Address: 〒101-0045 東京都千代田区神田鍛冶町3-7-4
神田374ビル4F
KANDA374 Bldg. 4F, 3-7-4 Kanda Kajicho,
Chiyoda-ku, Tokyo 101-0045, Japan
Tel: +81-3-4455-3722
Fax: +81-3-5209-6013
Email: japan@adlinktech.com

ADLINK Technology, Inc. (Korean Liaison Office)
Address: 서울시 서초구 서초동 1675-12 모인터빌딩 8층
8F Mointer B/D,1675-12, Seocho-Dong, Seocho-Gu,
Seoul 137-070, Korea
Tel: +82-2-2057-0565
Fax: +82-2-2057-0563
Email: korea@adlinktech.com

ADLINK Technology Singapore Pte. Ltd.
Address: 84 Genting Lane #07-02A, Cityneon Design Centre,
Singapore 349584
Tel: +65-6844-2261
Fax: +65-6844-2263
Email: singapore@adlinktech.com

ADLINK Technology Singapore Pte. Ltd. (Indian Liaison Office)
Address: 1st Floor, #50-56 (Between 16th/17th Cross) Margosa Plaza,
Margosa Main Road, Malleswaram, Bangalore-560055, India
Tel: +91-80-65605817, +91-80-42246107
Fax: +91-80-23464606
Email: india@adlinktech.com

9 Additional Information

CoreExpress™

http://www.CoreExpress.c om

Intel® Atom™

http://www.intel.com/products/centrino/atom/index.htm

http://www.intel.com/design/intarch/atom500/index.htm

Intel® System Controller Hub

http://www.intel.com/design/chipsets/embedded/SCHUS15W/techdocs.htm

USB

Universal Serial Bus (USB) connects computers, peripherals and more at www.usb.org

PCI-Express

PCI Express Specification, Revision 1.1 at www.pcisig.com/specifications/pciexpress/

ACPI

Advanced Configuration and Power Interface Specification (ACPI), Revision 3.0 at www.acpi.info/spec.htm

SMB

System Management Bus (SMBus) at www.smbus.org

EAPI

EAPI Specification 1.0 www.picmg.org/pdf/COM_EAPI_R1_0.pdf

Debian

Debian the universal OS at www.debian.org

thttpd

thttpd - tiny/turbo/throttling HTTP server at acme.com/software/thttpd/

IIS

Internet Information Services at www.iis.net

Qt

Qt Framework at http://qt.digia.com/

10 Revision History

Filename	Date	Edited by	Change
SME-LEMT-R0V1	2008-07-22	MH	Draft
SME-LEMT-R0V2	2008-07-28	MH	Added: write memory via command line
SME-LEMT-R0V3	2008-08-08	MH	Release version
SME-LEMT-R0V4	2009-21-08	MH	Added: - Part- and serial number - Updated screenshots - BIOS, SMC bootloader and firmware version - Unsupported features grayed out in GUI - Firmware update via command line
SME-LEMT-R0V5	2011-04-04	MH	Added: - Backlight control - Power-/Current monitoring - Updated screenshots
SME-LEMT-R0V6	2011-07-11	SLB	Added: - EAPI Chapter
SME-LEMT-R0V7	2011-10-19	SLB	Added: - LEMTWeb Chapter - Information about SMBus driver in Linux
SME-LEMT-R0V8	2012-02-29	SLB	Changed: - Chapter 4: new GUI description - Chapter 2: Requirements added

Filename	Date	Edited by	Change
SME-SEMA-R0V9	2013-01-24	MH	Changed: - LEMT -> SEMA, SMC -> BMC Added: - SMBus + ext. I2C bus access - System monitor (fan speed + voltages) - Smart fan control
SME-SEMA-R1V0	2013-01-30	MG	Final Release for SEMA
SME-SEMA-R1V1	2013-02-05	SLB	Changed: - Supported EAPI Functions adapted to new SEMA features - Nokia changed to Digia
SME-SEMA-R1V2	2013-04-29	MG	Second Release
SME-SEMA-R1V3	2013-06-27	SB	Changed: - Semagui chapter to reflect new features added with version 2.1 - Updated screenshot of Semagui -Sema command line tool chapter to reflect new features