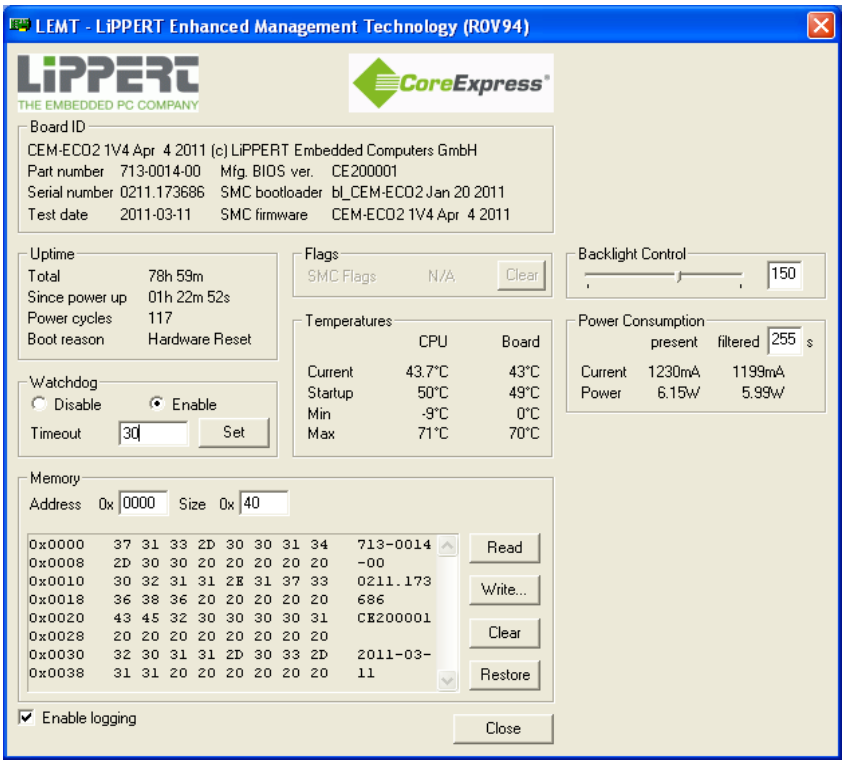
LEMT (LiPPERT Enhanced Management Technology)

LEMT Technical Manual



TME-LEMT-0V8.DOC
Revision 0.8 / April 11

Hans-Thoma-Str. 11
D-68163 Mannheim
http://www.lippertembedded.com/

Technical Manual LEMT

LiPPERT Document: TME-LEMT-0V8.DOC Revision 0.8

Trademarks

MS-DOS, Windows, Windows 95, Windows 98, Windows NT and Windows XP are trademarks of Microsoft Corporation. PS/2 is a trademark of International Business Machines, Inc. Intel and Solid State Drive are trademarks of Intel Corporation. Geode is a trademark of Advanced Micro Devices. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Table of Contents

1 OVERVIEW

The onboard System Management Controller (SMC) on many Lippert Boards implements power sequencing and LEMT (**L**iPPERT **E**nhanced **M**anagement **T**echnology) functionality. The microcontroller communicates via the System Management Bus with the CPU/Chipset.

The following functions are implemented permitting the user to:

- Set/Clear the Watchdog-timer
- Read the uptime seconds since power on
- Read the total ontime minutes
- Read the power cycles
- Read the current temperature of the CPU and Board
- Get PowerUp-Temperature of CPU and Board
- Min-/Max-Temperature Logger of CPU and Board
- Read Onboard-Voltages
- Read current drawn by the board
- Control the Backlight of the LVDS
- Get Fan-Speed
- Get access to 512 / 1024 Bytes of Flash-ROM for User data
- Save Data in 128 Bytes of Write-Protectable Flash-ROM (fused)
- Read the version information of the SMC Firmware
- Get the status information of the SMC

Some features are not included on several boards. This table gives you an overview of the implemented features:

Board	CoreExpress-ECO	CoreExpress-ECO2	CoreExpress-ARM	CRR-945GSE	CXR-GS45
Ontime Counters	✓	✓	✓	✓	✓
Power-Cycle Counter	✓	✓	✓	✓	✓
Watchdog Timer	✓	✓ (Type II)	✓ (Type II)	✓	✓
Voltages Monitor		✓	✓	✓	✓
Temperature Monitor	✓ (no PROCHOT#)	✓ (no PROCHOT#)	✓ (no PROCHOT#)	✓	✓
Fan Speed and Control				✓	✓
Manufacturing Data	✓	✓	✓	✓	✓
Status-/Version-Info of SMC	✓	✓	✓	✓	✓
User Flash Size	1024 Bytes	1024 Bytes	1024 Bytes	1024 Bytes	1024 Bytes
Power Monitor		✓	✓		
Backlight Control		✓ (optional)	✓ (optional)		

Board	CSR/CLR/ Hurricane-LX800	Thunderbird-E3100	Thunderbird-GM45	Hurricane-QM57	CLR-86DX
Ontime Counters	✓	✓	✓	✓	✓
Power-Cycle Counter	✓	✓	✓	✓	✓
Watchdog Timer	✓	✓	✓	✓ (Type II)	✓ (Type II)
Voltages Monitor		✓	✓	✓	✓
Temperature Monitor		✓	✓	✓	✓ (Board only)
Fan Speed and Control		✓	✓	✓	
Manufacturing Data	✓	✓	✓	✓	✓
Status-/Version-Info of SMC	✓	✓	✓	✓	✓
User Flash Size	512 Bytes	1024 Bytes	1024 Bytes	1024 Bytes	1024 Bytes
Power Monitor					
Backlight Control					

2 ACCESS TO THE SMC-CONTROLLER

The communication to the SMC-Controller is done through the SMBus 1.0-compliant host controller of the Chipset. This host controller provides a mechanism for the CPU to initiate communications with the SMC as slave device. Please look at the Chipset Datasheet for detailed information about the SMBus host controller. The SMC slave address is 0101000X, where X is the Read(1)-/Write(0)-bit.

BITS							
A7	**A6**	**A5**	**A4**	**A3**	**A2**	**A1**	**A0**
0	1	0	1	0	0	0	Read/Write

2.1 SMC STANDARD RECEIVER FORMAT (WRITE CYCLE)

The SMC receiver format is shown in Table 2.1.1.

START	SMC_addr	ACK	command	ACK	length	ACK	data_byte	ACK	n-bytes	ACK	STOP

Table 2.1.1 SMC *receiver* format

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit) (send by chipset)
ACK	SM-bus acknowledge by SMC
command	Selected Command (send by chipset)
length	Number of Data-bytes (send by chipset)
Data_byte	data byte transmitted to SMC receiver (send by chipset)
n-bytes	more data bytes transmitted to SMC receiver (send by chipset)
STOP	SM-bus stop condition

Table 2.1.2 Explanation of Table 2.1.1

2.2 SMC STANDARD TRANSMITTER FORMAT (READ CYCLE)

The SMC transmitter format is shown in Table 2.2.1.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	length	ACK(C)
data_byte	ACK(C)	n-bytes	ACK(C)	STOP					

Table 2.2.1 SMC *transmitter* format (direction of length and data-bytes changed)

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit) (send by chipset)
ACK(C)	SM-bus acknowledge (send by chipset)
ACK	SM-bus acknowledge (send by SMC)
command	Selected Command (send by chipset)
length	Number of Data-bytes (send by SMC)
data_byte	data byte transmitted to SMC receiver (send by SMC)
n-bytes	more data bytes transmitted to SMC receiver (send by SMC)
STOP	SM-bus stop condition

Table 2.2.2 Explanation of Table 2.2.1

3 COMMANDS OF THE SMC

To start communication between the bus master (SMBus host controller of Chipset) and the slave device (SMC), the host controller must initiate a Start condition followed by the address of the SMC. The SMC recognizes his address and responds with ACK. Following this, the host controller sends the Command Code (shown in Table 3.1).

COMMAND	DESCRIPTION
0x20	Set/Clear Watchdog-timer
0x21	Reserved for BIOS Timeout Counter
0x22	Set timeout of Watchdog-timer on power-up
0x26-0x28	Reserved for Softstraps
0x2E	Clear Exception Code
0x2F	Get SMC Capability
0x30	Read Version-String 1
0x31	Read Version-String 2
0x32	Read Total Ontime Minutes
0x33	Read Ontime Seconds since Power-Up
0x34	Read Power Cycles
0x35	Read SMC Flags
0x36	Get last System Restart Event
0x37	Read current CPU Temperature
0x38	Read current Board Temperature
0x39	Get Min/Max-Temperatures (Temperature Logger)
0x3A	Get Startup-Temperatures of CPU and Board
0x3B	Get number of PROCHOT#-Events
0x3C	Get number of TS#-Events
0x3D	Get SMC-Status
0x3F	Read Version-String of Bootloader
0x40	Set Address/Length for Flash Access
0x41	Write Data to User-Flash
0x42	Read Data from User-Flash
0x43	Clear all Data in User-Flash
0x44	Write-protect SECURE-Area
0x60-0x67	Get Voltages
0x68	Get Fan-Speed
0x69	Get Main Current (Power Monitor)
0x70-0x77	Get Board Manufacturing Data
0x80	Set Backlight PWM
0x81	Get Backlight PWM
0x90-0x97	Reserved for Board specific internal communication

Table 3.1 SMC Command Codes

4 DETAILED DESCRIPTION OF SMC COMMAND CODES

4.1 Set/Clear Watchdog-timer

After Start-Up the Watchdog is disabled or loaded with a startup value (Type II Watchdog) from the Flash memory. This command must be used to (re-)load or to clear the Timeout of the Watchdog-timer. The timeout value is given in seconds and has a 16-bit (two byte wide) size. So the Timeout of the Watchdog can be set to 1-65535 seconds. 0 disables the Watchdog.

START	SMC_addr	ACK	command	ACK	length	ACK	MSB_byte	ACK	LSB_byte	ACK	STOP

Table 4.1.1 SMC receiver format of the Watchdog-timer Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x20 = "Set/Clear Watchdog-timer" Command code
length	0x02 = 2 Bytes data length
MSB_byte	High-byte of Timeout-value in seconds
LSB_byte	Low-byte of Timeout-value in seconds
STOP	SM-bus stop condition

Table 4.1.2 Explanation of Table 4.1.1

4.2 Set/Clear initial timeout of Watchdog-timer

This command sets the Timeout value of the Watchdog-timer which is loaded at power-up. This initial timeout value is given in seconds and has a 16-bit (two byte wide) size. It can be set from 25 to 65535 seconds. 0 disables the loading of an initial timeout to the Watchdog.

START	SMC_addr	ACK	command	ACK	length	ACK	MSB_byte	ACK	LSB_byte	ACK	STOP

Table 4.2.1 SMC receiver format of the Watchdog-timer Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x22 = "Set/Clear initial Watchdog-timeout" Command code
length	0x02 = 2 Bytes data length
MSB_byte	High-byte of Timeout-value in seconds
LSB_byte	Low-byte of Timeout-value in seconds
STOP	SM-bus stop condition

Table 4.2.2 Explanation of Table 4.2.1

4.3 Clear Exception Code

This command clears the Flash stored Exception Code. Please look at the board technical manual chapter "LEMT functions" for details.

START	SMC_addr	ACK	command	ACK	STOP

Table 4.3.1 SMC receiver format of the Clear Exception Code Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x2E = "Clear Exception Code" Command code
STOP	SM-bus stop condition

Table 4.3.2 Explanation of Table 4.3.1

4.4 Get SMC Capability

This command reports the Capability of the SMC. Several features are controlled by the SMC on different hardware platforms. The Capability bytes gives the User the information about the implemented features (explained in Table 4.4.2).

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr	ACK	length
ACK	Byte 1 (MSB)	ACK	Byte 2 (LSB)	ACK	STOP			

Table 4.4.1 SMC transmitter format of the Get SMC Capability Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x2F = "Get SMC Capability" Command code
length	0x02 = 2 Bytes data length
Byte 1,2	Data-Bytes containing the Capability Bits Bit 11-15: set to 0 for future functionality Bit 10: Power Monitor (current sense) 0 = not provided, 1 = featured Bit 9: Watchdog Type II (initial timeout of Watchdog) 0 = not provided, 1 = featured Bit 8: Backlight control 0 = not provided, 1 = featured Bit 7: Bootloader timeout programable 0 = not provided, 1 = featured Bit 6: Storage of failure reason 0 = not provided, 1 = featured Bit 5: Voltage Monitor 0 = not provided, 1 = featured Bit 4: Temperatures 0 = not provided, 1 = featured Bit 3: Watchdog 0 = not provided, 1 = featured Bit 2: USER FLASH Size: 0 = 512 Bytes, 1 = 1024 Bytes Bit 1: System Restart Event 0 = not provided, 1 = featured Bit 0: Uptime & Power Cycles Counter 0 = not provided, 1 = featured
STOP	SM-bus stop condition

Table 4.4.2 Explanation of Table 4.4.1

4.5 Read Version-String 1/2

This command reads the Version Information of the SMC Firmware. Each command receives 1 length Byte and 32 Bytes of Data.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	length	ACK(C)
Byte 1	ACK(C)	…	ACK(C)	Byte 32	ACK(C)	STOP			

Table 4.5.1 SMC transmitter format of the Version String 1/2 Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x30 / 0x31 = "Read Version-String 1/2" Command code
length	0x20 = 32 Bytes data length
Byte 1-32	ASCII-Byte of Information String
STOP	SM-bus stop condition

Table 4.5.2 Explanation of Table 4.5.1

4.6 Read Total Ontime Minutes

This command reads the total Ontime Minutes of the System running in S0-State. The Ontime Minute-counter has a 32-bit (four byte wide) size and is count-up every minute in the Flash of the SMC. The Most Significant Byte is sent first, followed by the next highest Byte down to the Least Significant Byte (Table 4.6.1).

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	
length	ACK(C)	Byte 1 (MSB)	ACK(C)	…	ACK(C)	Byte 4 (LSB)	ACK(C)	STOP

Table 4.6.1 SMC transmitter format of the Total Ontime Minutes Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x32 = "Read Total Ontime Minutes" Command code
Length	0x04 = 4 Bytes data length
Byte 1-4	Value of Ontime Minute Counter
STOP	SM-bus stop condition

Table 4.6.2 Explanation of Table 4.6.1

4.7 Read Ontime Seconds

This command reads the Ontime Seconds of the System running in S0-State starting at last power-on. The Ontime Seconds-counter has a 32-bit (four byte wide) size and is count-up every second in the SMC. The counter is cleared when the system is removed from power. The Most Significant Byte is sent first, followed by the next highest Byte down to the Least Significant Byte (Table 4.7.1).

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK

length	ACK(C)	Byte 1 (MSB)	ACK(C)	…	ACK(C)	Byte 4 (LSB)	ACK(C)	STOP

Table 4.7.1 SMC transmitter format of the Ontime Seconds Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x33 = "Read Ontime Seconds" Command code
Length	0x04 = 4 Bytes data length
Byte 1-4	Value of Ontime Seconds Counter
STOP	SM-bus stop condition

Table 4.7.2 Explanation of Table 4.7.1

4.8 Read Power Cycles

This command reads the Power Cycles of the System. The Power-cycle-counter has a 32-bit (four byte wide) size and is updated every time when the System is Powered-ON in the Flash of the SMC. The Most Significant Byte is sent first, followed by the next highest Byte down to the Least Significant Byte (Table 4.8.1).

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	Byte 1 (MSB)	ACK(C)	…	ACK(C)	Byte 4 (LSB)	ACK(C)	STOP

Table 4.8.1 SMC transmitter format of the Power Cycles Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x34 = "Read Power Cycles" Command code
length	0x04 = 4 Bytes data length
Byte 1-4	Value of Power Cycle Counter
STOP	SM-bus stop condition

Table 4.8.2 Explanation of Table 4.8.1

4.9 Read SMC Flags

This command reads the SMC Flags and gives an information of the internal status of the SMC.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	data	ACK(C)	STOP			

Table 4.9.1 SMC transmitter format of the Read BIOS Flags Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x35 = "Read SMC Flags" Command code
length	0x01 = 1 Bytes data length
data	Reflects the SMC status Bit 5-7: System State 0 = S0-State 3 = S3-State 5 = S5-State other values are Board specific -> see TME of Board chapter "LEMT functions" Bit 0-4: Exception Code Board specific -> see TME of Board chapter "LEMT functions"
STOP	SM-bus stop condition

Table 4.9.2 Explanation of Table 4.9.1

4.10 Get last System Restart Event

This command reads the cause of the last System Restart. The returned code is explained in Table 4.10.2.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	data	ACK(C)	STOP			

Table 4.10.1 SMC transmitter format of the System Restart Event Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x36 = "System Restart Event" Command code
length	0x01 = 1 Bytes data length
data	Data-Byte containing the cause of the last restart 0x00: UNKNOWN Unknown Reason of Restart (should never reported) 0x20: SW_RESET A reset by Software caused the Restart of the system 0x30: HW_RESET A reset by Hardware caused the Restart of the system (e.g. Reset-Button) 0x40: WATCHDOG The Watchdog has restarted the system 0x50: BIOS_FAULT Main-BIOS is corrupted -> boot from Backup BIOS 0x60: POWER_DOWN The system was Shutdown (e.g. Power-Button, ACPI Shutdown) 0x70: POWER_LOSS The system is restarted after a Power-loss (e.g. external Power supply instable or switched off while the system was running) 0x80: POWER_CYCLE The system is restarted after a Power-cycle (e.g. internal Power supply has failed) 0x90: VIN_DROP The system is restarted after a Voltage Drop of the Main-Input-Voltage 0xA0: PWR_FAIL The system is restarted after a PWRFAIL detection of an internal power supply circuit
STOP	SM-bus stop condition

Table 4.10.2 Explanation of Table 4.10.1

4.11 Read current CPU Temperature

This command reads the current CPU Temperature. The CPU temperature is transmitted in two Bytes. The first byte is the full degree value, the second byte the value after the comma (Table 4.11.1). Both values are signed chars.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	Byte 1 (MSB)	ACK(C)	Byte 2 (LSB)	ACK(C)	STOP	

Table 4.11.1 SMC transmitter format of the CPU Temperature Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x37 = "Read current CPU Temperature" Command code
Length	0x02 = 2 Bytes data length
Byte 1	Full degree value of CPU temperature. The format of this byte is signed char.
Byte 2	Value after comma of CPU temperature
STOP	SM-bus stop condition

Table 4.11.2 Explanation of Table 4.11.1

4.12 Read current Board Temperature

This command reads the current Board Temperature. The Board temperature is transmitted in a signed char Byte.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	data	ACK(C)	STOP			

Table 4.12.1 SMC transmitter format of the Read current Board Temperature Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x38 = "Read current Board Temperature" Command code
length	0x01 = 1 Bytes data length
Data	Board temperature. The format of this byte is signed char.
STOP	SM-bus stop condition

Table 4.12.2 Explanation of Table 4.12.1

4.13 Get Min/Max-Temperatures (Temperature Logger)

This command reads the maximum and minimum reached CPU and Board Temperatures. The temperatures are transmitted in a signed char Bytes.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	
length	ACK(C)	MaxCPUTemp	ACK(C)	MinCPUTemp	ACK(C)	MaxBoardTemp	ACK(C)	
MinBoardTemp	ACK(C)	STOP						

Table 4.13.1 SMC transmitter format of the Get Min/Max-Temperatures Command

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x39 = "Get Min/Max-Temperatures" Command code
length	0x04 = 4 Bytes data length
MaxCPUTemp	Maximum CPU temperature. The format of this byte is signed char.
MinCPUTemp	Minimum CPU temperature. The format of this byte is signed char.
MaxBoardTemp	Maximum Board temperature. The format of this byte is signed char.
MinBoardTemp	Minimum Board temperature. The format of this byte is signed char.
STOP	SM-bus stop condition

Table 4.13.2 Explanation of Table 4.13.1

4.14 Get Startup-Temperatures of CPU and Board

This command reads the CPU and Board Temperatures at power-on time. The temperatures are transmitted in a signed char Bytes.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	CPU_temp	ACK(C)	Board_temp	ACK(C)	STOP	

Table 4.14.1 SMC transmitter format of the Get Startup-Temperatures of CPU and Board Command

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x3A = "Get Startup-Temperatures of CPU and Board" Command code
Length	0x02 = 2 Bytes data length
CPU_temp	CPU temperature at Startup. The format of this byte is signed char.
Board_temp	Board temperature at Startup. The format of this byte is signed char.
STOP	SM-bus stop condition

Table 4.14.2 Explanation of Table 4.14.1

4.15 Get Number of PROCHOT#-Events

This command reads the PROCHOT#-Event-counter. PROCHOT# (processor hot), is asserted by the CPU when the processor die temperature has reached its maximum operating temperature. Since Thermal Monitor 2 is enabled, the reduces it's clock speed and decreases it's Core Voltage when PROCHOT# is asserted. This counter gives the User an information of Cooling problems. This counter is cleared when the system is removed from power.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	
length	ACK(C)	Byte 1 (MSB)	ACK(C)	…	ACK(C)	Byte 4 (LSB)	ACK(C)	STOP

Table 4.15.1 SMC transmitter format of the Get Number of PROCHOT#-Events Command

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x3B = "Get Number of PROCHOT#-Events" Command code
length	0x04 = 4 Bytes data length
Byte 1-4	Value of PROCHOT#-Counter
STOP	SM-bus stop condition

Table 4.15.2 Explanation of Table 4.15.1

4.16 Get Number of TS#-Events

This command reads the TS#-Event-counter. TS# is asserted by a Temperature sensor when a device reaches it's critical temperature and released when the device is back into its normal temperature range. This counter gives the User an information of Temperature/Cooling problems. This counter is cleared when the system is removed from power.

Depending on the board this command reflects the status of different devices. Please look at the board technical manual chapter "LEMT functions" for details.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	Byte 1 (MSB)	ACK(C)	…	ACK(C)	Byte 4 (LSB)	ACK(C)	STOP

Table 4.16.1 SMC transmitter format of the Get Number of TS#-Events Command

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x3C = "Get Number of TS#-Events" Command code
length	0x04 = 4 Bytes data length
Byte 1-4	Value of TS#-Counter
STOP	SM-bus stop condition

Table 4.16.2 Explanation of Table 4.16.1

4.17 Read SMC Status

This command gives an information of the status of the SMC controlled signals.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	data	ACK(C)	STOP			

Table 4.17.1 SMC transmitter format of the Read SMC Status Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x3D = "Read SMC Status" Command code
length	0x01 = 1 Bytes data length
data	Reflects the status of the SMC controlled signals Board specific -> see TME of Board chapter "LEMT functions"
STOP	SM-bus stop condition

Table 4.17.2 Explanation of Table 4.17.1

4.18 Read Bootloader Version-String

This command reads the Version Information of the SMC Bootloader. Each command receives 1 length Byte and 24 Bytes of Data.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	length	ACK(C)
Byte 1	ACK(C)	…	ACK(C)	Byte 24	ACK(C)	STOP			

Table 4.18.1 SMC transmitter format of the Bootloader Version String Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x3F = "Read Bootloader Version-String" Command code
length	0x18 = 24 Bytes data length
Byte 1-24	ASCII-Byte of Version String
STOP	SM-bus stop condition

Table 4.18.2 Explanation of Table 4.18.1

4.19 Set Address/Length for Flash Access

This command prepares the read or write access to the SMC-Flash. For this, the address and the length information must be set in the first step. In the next step the Data can be read or write to the Flash. The address must have a 16-bit format and a length information about the number of bytes to write. For accesses to the USER-Area be sure that the address have 32-bit alignment and is within 1kB (or 512Bytes), so 0x0000, 0x0004, 0x0008, …, 0x03f8, 0x03fc are valid addresses. To access the SECURE-Area the address must be 0x8000 or greater but not higher than 0x807c. The memory map is show in Table 4.14.1.

The number of bytes to write must be a multiple of 4 and not more than 32 Bytes.

No Memory	0xFFFF 0x8080
SECURE-Data	0x807F 0x8000
No Memory	0x7FFF 0x0400
USER-Data	0x03FF 0x0000

Table 4.19.1 Memory-Map of SMC-Flash with 1024 Bytes User-Data

START	SMC_addr	ACK	command	ACK	length	ACK	MSB_Addr	ACK	LSB_Addr	ACK
Data_length	ACK	STOP								

Table 4.19.2 SMC receiver format of the Set Address/Length for Flash Access Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x40 = “Set Address/Length for Flash Access” Command code
length	0x03 = 3 Bytes data length
MSB_byte	High-byte of Flash-Address
LSB_byte	Low-byte of Flash-Address (32bit-alignment)
Data_length	Number of Data-Bytes to read or write (in steps of 4)
STOP	SM-bus stop condition

Table 4.19.3 Explanation of Table 4.19.2

4.20 Write Data to User-Flash

This command writes data to the selected address of the SMC-Flash. Up to 32 Bytes can be written in a single block access. The number of bytes to write must be a multiple of 4. This command takes up to 250ms of time (depends if the requested Flash-Area is clear or not). During this time the SMC is not accessible. To detect if the SMC is busy use a single read with 0xFF as command (like Read BIOS Flags, but use 0xFF instead of 0x35). If the SMC responds with 0xF0, the SMC is ready for next steps.

START	SMC_addr	ACK	command	ACK	length	ACK	Data0	ACK	…	ACK
Data_n	ACK	STOP								

Table 4.20.1 SMC receiver format of the Write Data to User-Flash Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x41 = “Write Data to User-Flash” Command code
Length	Number of Data_Bytes to write (=n)
Data0..n	4-32 Data-Bytes to write
STOP	SM-bus stop condition

Table 4.20.2 Explanation of Table 4.20.1

4.21 Read Data from User-Flash

This command reads data from the selected address of the SMC-Flash. Up to 32 Bytes can be written in a single block access. The number of bytes to write must be a multiple of 4.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	length	ACK(C)
Data0	ACK(C)	…	ACK(C)	Data_n	ACK(C)	STOP			

Table 4.21.1 SMC transmitter format of the Read Data from User-Flash Command

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	01010001b = 0x51 (Read access to SMC)
ACK	SM-bus acknowledge condition generated by receiver
command	0x42 = "Read Data from User-Flash" Command code
length	Number of Data_Bytes to write
Data0..n	4-32 Data-Bytes
STOP	SM-bus stop condition

Table 4.21.2 Explanation of Table 4.21.1

4.22 Clear all Data in User-Flash

This single command clears the total USER-Area of the SMC-Flash. To prevent unintentional erase of the User-Flash two Magic Bytes must be send with this command (see Table 4.22.2). This command takes up to 50ms of time. During this time the SMC is not accessible. To detect if the SMC is busy use a single read with 0xFF as command (like Read BIOS Flags, but use 0xFF instead of 0x35). If the SMC responds with 0xF0, the SMC is ready for next steps.

START	SMC_addr	ACK	command	ACK	length	ACK	Magic_1	ACK	Magic_2	ACK	STOP

Table 4.22.1 SMC receiver format of the Clear all Data in User-Flash Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	01010000b = 0x50 (8-bit) (Write access to SMC)
ACK	SM-bus acknowledge condition generated by receiver
command	0x43 = "Clear all Data in USER-Area" Command code
length	0x02 = Length of this Command
Magic_1	0xAE = Magic Byte 1
Magic_2	0xCD = Magic Byte 2
STOP	SM-bus stop condition

Table 4.22.2 Explanation of Table 4.22.1

4.23 Write-protect SECURE-Area

This single command enables the protection of total SECURE-Area against write accesses. Since the SECURE-Area is located at the bootloader region, the bootloader will also locked. No further bootloader update is possible when the SECURE-Area is locked. To prevent unintentional protect of the SECURE-Area two Magic Bytes must be send with this command (see Table 4.23.2).

Caution: If this security setting has been applied, the SECURE-Area of that device will not be rewritten by anyone (protected by hardware-fuse).

START	SMC_addr	ACK	command	ACK	length	ACK	Magic_1	ACK	Magic_2	ACK	STOP

Table 4.23.1 SMC receiver format of the Write-protect SECURE-Area Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	01010000b = 0x50 (8-bit) (Write access to SMC)
ACK	SM-bus acknowledge condition generated by receiver
command	0x44 = "Write-protect SECURE-Area" Command code
length	0x02 = Length of this Command
Magic_1	0xAE = Magic Byte 1
Magic_2	0xCE = Magic Byte 2
STOP	SM-bus stop condition

Table 4.23.2 Explanation of Table 4.23.1

4.24 Get Voltage

The 10-bit-ADC of the SMC is used to monitor the onboard Voltages. SMC provides 8 Channels which are separately accessed by Command codes as described in Table 4.24.2.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	MSB_byte	ACK(C)	LSB_byte	ACK(C)	STOP	

Table 4.24.1 SMC transmitter format of the Get Voltage Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	"Get Voltage" Command code: 0x60: Get Voltage 0 0x61: Get Voltage 1 0x62: Get Voltage 2 0x63: Get Voltage 3 0x64: Get Voltage 4 0x65: Get Voltage 5 0x66: Get Voltage 6 0x67: Get Voltage 7
length	0x02 = 2 Bytes data length
MSB byte LSB byte	16-Bit data value of Voltage
STOP	SM-bus stop condition

Table 4.24.2 Explanation of Table 4.24.1

The returned values are different for each board type. Please look at the board technical manual chapter "LEMT functions" for the equations to translate the returned values into voltages.

4.25 Get Fan Speed

This command reads the current Speed of the CPU Cooler Fan. It delivers the Fan Speed in RPM.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK
length	ACK(C)	MSB_byte	ACK(C)	LSB_byte	ACK(C)	STOP	

Table 4.25.1 SMC transmitter format of the Get Fan Speed Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x68 = "Get Fan Speed" Command code
length	0x02 = 2 Bytes data length
MSB byte LSB byte	16-Bit data value of Fan Speed (RPM)
STOP	SM-bus stop condition

Table 4.25.2 Explanation of Table 4.25.1

4.26 Get Main Current

This command reads 4 values representing the last 4 currents drawn by the board. The values are sampled every 250ms. The order of the 4 values is NOT in relationship to the time. The access to the SMC may increase the drawn current of the whole system. In this case, you still have 3 samples without the influence of the read access.

The returned values must be calculated by the user application into a current. Please look at the board technical manual chapter "LEMT functions" for the equation to calculate the returned value into a current.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	
length	ACK(C)	MSB_byte1	ACK(C)	LSB_byte1	ACK(C)	...	MSB_byte4	ACK(C)
LSB_byte4	ACK(C)	STOP						

Table 4.26.1 SMC transmitter format of the Get Main Current Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	0x69 = "Get Main Current" Command code
length	0x08 = 8 Bytes data length
MSB byte n LSB byte n	4 samples of 16-Bit data value of Main Current (n=1..4)
STOP	SM-bus stop condition

Table 4.26.2 Explanation of Table 4.26.1

4.27 Get Board Manufacturing Data

These commands reads the contents of the 8 Data-Fields. The Manufacturing Data gives information about the Board-Serial-Number, BIOS-Version, Test-Date, etc. Each command receives 1 length Byte and 16 Bytes of Data.

START	SMC_addr	ACK	command	ACK	START (Repeat)	SMC_addr+1	ACK	length	ACK(C)
Byte 1	ACK(C)	…	ACK(C)	Byte 16	ACK(C)	STOP			

Table 4.27.1 SMC transmitter format of the Get Board Manufacturing Data Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	0x50 (8-bit)
ACK	SM-bus acknowledge condition generated by receiver
command	"Get Board Manufacturing Data" Command code: 0x70: BOARD PART NUMBER 0x71: BOARD SERIAL NUMBER 0x72: BIOS VERSION 0x73: TEST DATE 0x74: For Future data 0x75: For Future data 0x76: For Future data 0x77: For Future data
length	0x10 = 16 Bytes data length
Byte 1-16	ASCII-Byte of Information String
STOP	SM-bus stop condition

Table 4.27.2 Explanation of Table 4.27.1

4.28 Set Backlight PWM

This command sets the duty of Backlight control output. The value of the PWM also controls the Backlight enable output. In case of PWM=0, the Backlight enable output will go inactive.

START	SMC_addr	ACK	command	ACK	length	ACK	PWM_value	ACK	STOP

Table 4.28.1 SMC receiver format of the Set Backlight PWM Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	01010000b = 0x50 (8-bit, write access to SMC)
ACK	SM-bus acknowledge condition generated by receiver
command	0x80 = "Set Backlight PWM" Command code
length	0x01 = Length of this Command
PWM_value	Duty setting (0-255)
STOP	SM-bus stop condition

Table 4.28.2 Explanation of Table 4.28.1

4.29 Get Backlight PWM

This command reads the current duty of the Backlight control output.

START	SMC_addr	ACK	command	ACK	length	ACK	PWM_value	ACK	STOP

Table 4.29.1 SMC receiver format of the Set Backlight PWM Command Code

ITEM	DESCRIPTION
START	SM-bus start condition
SMC_addr	01010000b = 0x50 (8-bit, write access to SMC)
ACK	SM-bus acknowledge condition generated by receiver
command	0x80 = "Set Backlight PWM" Command code
length	0x01 = Length of this Command
PWM_value	Duty setting (0-255)
STOP	SM-bus stop condition

Table 4.29.2 Explanation of Table 4.29.1

Appendix A, Contact Information

Headquarters

LiPPERT Embedded Computers GmbH
Hans-Thoma-Straße 11
68163 Mannheim
Germany

Phone	+49 621 4321410
Fax	+49 621 4321430
E-mail	sales@lippertembedded.com
	support@lippertembedded.com
Website	www.lippertembedded.com

US Office

LiPPERT Embedded Computers, Inc.
2220 Northmont Parkway, Suite 250
Duluth, GA 30096
USA

Phone	+1 (770) 295 0031
Fax	+1 (678) 417 6273
E-mail	ussales@lippertembedded.com
	support@lippertembedded.com
Website	www.lippertembedded.com

Appendix B, Getting Help

Should you have technical questions that are not covered by the respective manuals, please contact our support department at **support@lippertembedded.com**.

Please allow one working day for an answer!

Technical manuals as well as other literature for all LiPPERT products can be found in the *Products* section of LiPPERT's website www.lippertembedded.com. Simply locate the product in question and follow the link to its manual.

Returning Products for Repair

To return a product to LiPPERT for repair, you need to get a Return Material Authorization (RMA) number first.
Please fill in the RMA Request Form at http://www.lippertembedded.com/?id=rma and send it to us. We'll return it to you with the RMA number.

Deliveries without a valid RMA number are returned to sender at his own cost!

LiPPERT has a written Warranty and Repair Policy, which can be retrieved from http://www.lippertembedded.com/?id=rwp

It describes how defective products are handled and what the related costs are. Please read this document carefully before returning a product.

Appendix C, Further Resources

http://www.lippertembedded.com

LiPPERT Embedded Computers' website with news and detailed information.

http://www.intel.com

Datasheet of the CPU, Chipset and Ethernet-Controller.

http://www.smbus.org

Information about the System Management Bus (SMBus)

http://www.phoenix.com/en/customer+services/bios/awardbios

Additional BIOS information.

Appendix D, Revision History

Filename	Date	Edited by	Change
TME-LEMT-0V0.doc	2009-01-09	JS	Pre-release
TME-LEMT-0V1.doc	2009-02-17	JS	Function to read Bootloader-Version added Board specific Descriptions removed
TME-LEMT-0V2.doc	2009-05-07	JS	Text correction in Board Manufacturing Data
TME-LEMT-0V3.doc	2009-05-20	JS	SMC Flags defined Added POWER_CYCLE as System Restart Event
TME-LEMT-0V4.doc	2009-06-23	JS	Function to read SMC controlled signals added Added VIN_DROP as System Restart Event
TME-LEMT-0V5.doc	2010-02-05	JS	Board-table updated Command-table updated Clear Exception Code added Capability Register updated Get Number of TS#-Events added
TME-LEMT-0V6.doc	2010-02-19	JS	Restart Event table updated
TME-LEMT-0V7.doc	2010-03-19	JS	Added STARTREPEAT sequence in SMC Read Accesses
TME-LEMT-0V8.doc	2011-04-07	JS	Initial timeout of Watchdog-timer added Capability Register updated Get Main Current added (Power Monitor) Backlight control added