BP30

SW Driver
Power Management Specification

Edition 2006

Published by Neonseven s.r.l.,
Viale Stazione di Prosecco, 15

34010 Sgonico (Trieste) Italy

For questions on technology, delivery and prices please contact the Neonseven Offices in Italy Sgonico and Gorizia

Attention Please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact Neonseven.

Neonseven technologies may only be used in life-support devices or systems with the express written approval of Neonseven, if a failure of such technologies can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Table of Contents

1 Document Mission/Scope

1.1 Mission

This document is a technical specification about Power Management functionality in GLOBE1 and GLOBE2 based board. It provides information about last generation of Power Chipset provided by Infineon, E-POWERlite chipset.

1.2 Scope

This document is intended to describe driver functionality and programming interfaces used in GLOBE1 and GLOBE2 based board, so it is addressed to engineers and programmers who are interested in such activities. This technical specification is strictly dedicated to driver functionality about power management and is not intended to describe functionality about communications between E-GOLDlite and Power Management Chip.

2 List of Acronyms

Acronym	Meaning
IIC	Inter Integrated Circuits
I^2C	Same as above.
SMP 1.0	S/M-POWER Chip v <= 1.0
SMP 1.5	S/M-POWER v 1.5x
EPWRLT	E-POWERlite Chipset

3 Introduction

GLOBE based board is equipped by Infineon's E-POWERlite chip. The aim of this component is to provide an advanced power supply of the whole system, and is also used for battery management. It is designed for usage with E-GoldLite and Smarti-SD2 devices. In some release of GLOBE1 board is also used S/M-POWER v 1.5 and earlier, so the device driver described in this paper, support full detection and basic support for such devices. But it is optimized to support FULL functionality for E-POWERlite chipset only.
First releases of driver supported also SM-POWER v1.0 chipset. At present the driver officially supports only E-POWERlite chip. The current release still implements old chipset support but **it is NOT maintained, so it's not guaranteed to work correctly.** Also this documentation is updated only for E-POWERlite device.

3.1 Overview

E-POWERlite chip provides an advanced power supply, battery recharge and management functionalities. The aim of this document is to describe in details software drivers that control this device. For further technical specifications please also refer to official documentation provided by Infineon such as PMB6814 (E-POWERlite) Target specifications. All informations contained in these documents are assumed to be known.
In Figure 3-1 we can look at functional block diagram of E-POWERlite chipset series, with typical application circuit connected. For other release of chipset (i.e. S/M-POWER series) please refer to official technical references. E-POWERlite was developed for low cost platform developing, so it does not include some

regulators and features provided from S/M-POWER chip. The SW driver *epower.c* offers control of all features common to all chip versions. For special features the support is restricted to E-POWERlite chipset version only.

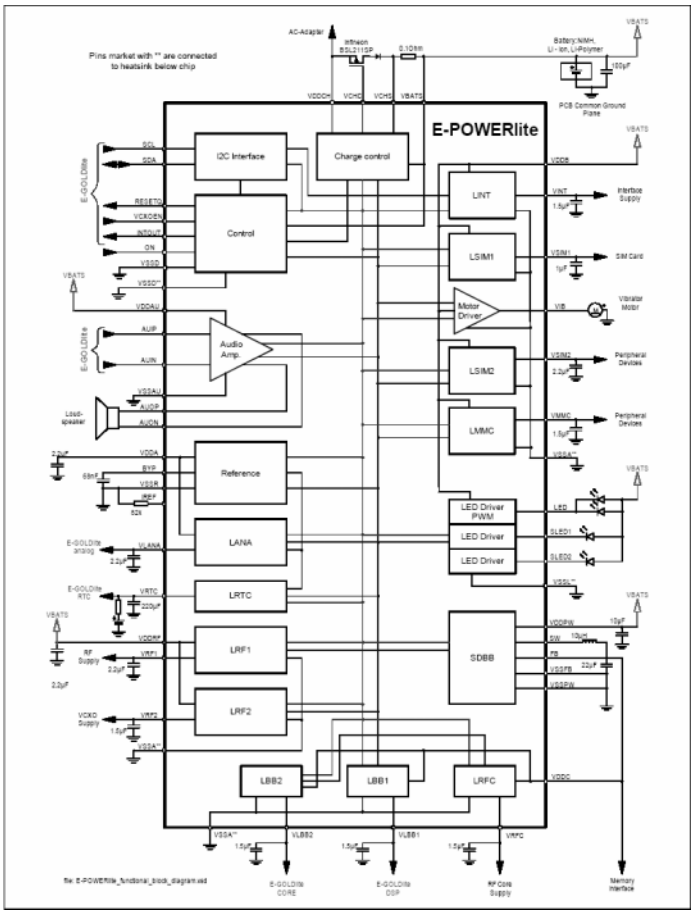

Figure 3-1

3.2 Power Control Registers

To manage the system supply, Power Chip provides a set of 8bit registers that controls all functionality of the regulators. Due to Power Chip version, we can have up to six power control registers, named **PWCTRL*n*** (where *n* is the number of register, from 1 to 6).
These registers must be programmed through IIC bus to have specific bit configuration. Each of these registers is bit mapped in different way and integrates one or more bit field, that control one or more regulator's behavior. The general form of bit field that control regulator's functionalities is reassumed in Table 3-1

FIELD NAME	Description	Comment
<RegName>V	Output Voltage Selection	Each regulator can be programmed to various Output Voltage level
<RegName>MD	On/Off function mode	Regulators state can be linked to VCXOEN input signal (from MCU) for power saving purposes

Table 3-1

VCXOEN is a control input that allows switching the state of Power Device. Usually it is provided from E-GOLDlite chip through a dedicated PIN. The status is VCXOEN=0 standby, VCXOEN=1 active.
Each regulator can be configured in a way that specifies its behavior when VCXOEN rises or lowers. This is reassumed in the following Table:

REGULATOR(S)	Meaning of VCXOEN signal
LBB1, LRFC (LRF0), LRF1, LRF2, (LRF3 if present)	Controls on/off state
LINT, LSIM, LSIM2, LMMC	Controls standby state
SDBB	Controls switching from PWM and PFM modes

Table 3-2

3.3 Other peripherals driver

The Power chip provides another set of registers that allow controlling special kind of peripherals. We have 2 registers that manage 2 signaling LED channels and BACKLIGHT programmable current, with PWM modulation for intensity dimming. These two registers are named LEDCTRL1 and LDCTRL2. Furthermore a register (DRVCTRL) controls vibrator driver, with different voltage level steps for vibrator dynamic control. Finally another set of registers manages the integrated audio amplifier features(AUDCTRL, AUDCTRL2)

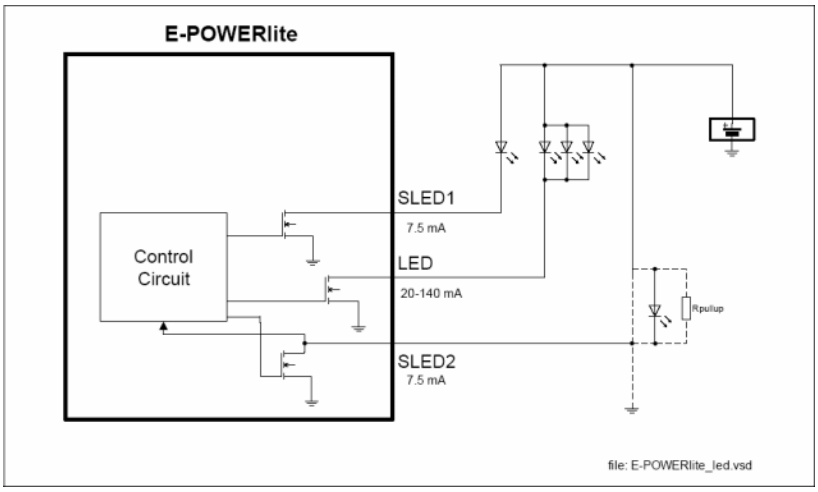

Figure 3-2

3.4 Battery Precharging and Charging features

The Power Chip series provides 2 registers to control and manage the battery charging (CHCTRL1, CHCTRL2). A precharge functionality with LED signaling is provided. Also different voltage levels for recharging could be chosen, allowing handling of several battery types (e.g. Li/Ion and NiMH).

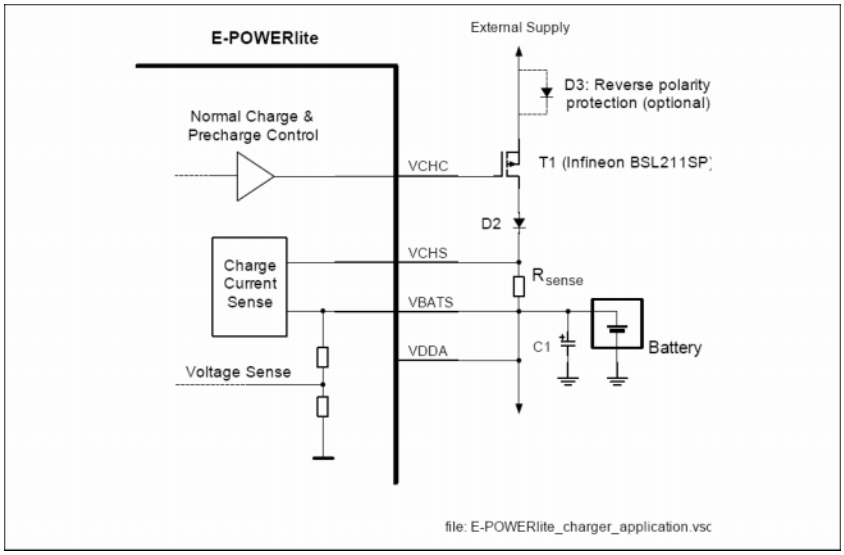

Figure 3-3

3.5 Interrupt channels

Another facility provided is an interrupt line to send to E-GOLDlite, named INTOUT. This signal is raised or lowered due to power or peripheral driver overtemp or current or charger status change. Each interrupt cause can be activated or deactivated (INTCTRL1 INTCTRL2 registers), then we can check for cause of interrupt from a read only status register (ISF).

4 Structure of Software Device Driver

The software driver that controls the device is developed in 3 different source files. The device is controlled directly from E-GOLDlite, via IIC bus. The lowest level of the driver provides to generate the correct switching for send and receive messages from Power device. Just one level above we have the part of driver that Read or Write to device registers (done calling the function defined in lower level). Finally there is the higher part of driver that switches on or off Regulators or other peripherals connected to Power device. Switching is done calling the function of the lower part of code. So we can define the following structure:

- pow_sm.c (in dwddrv/POW/src)
- pow.c (in dwddrv/POW/src)
- i2c.c (in dwddrv/I2C/src)

Starting from lower to higher level of source code. Following chart shows this structure

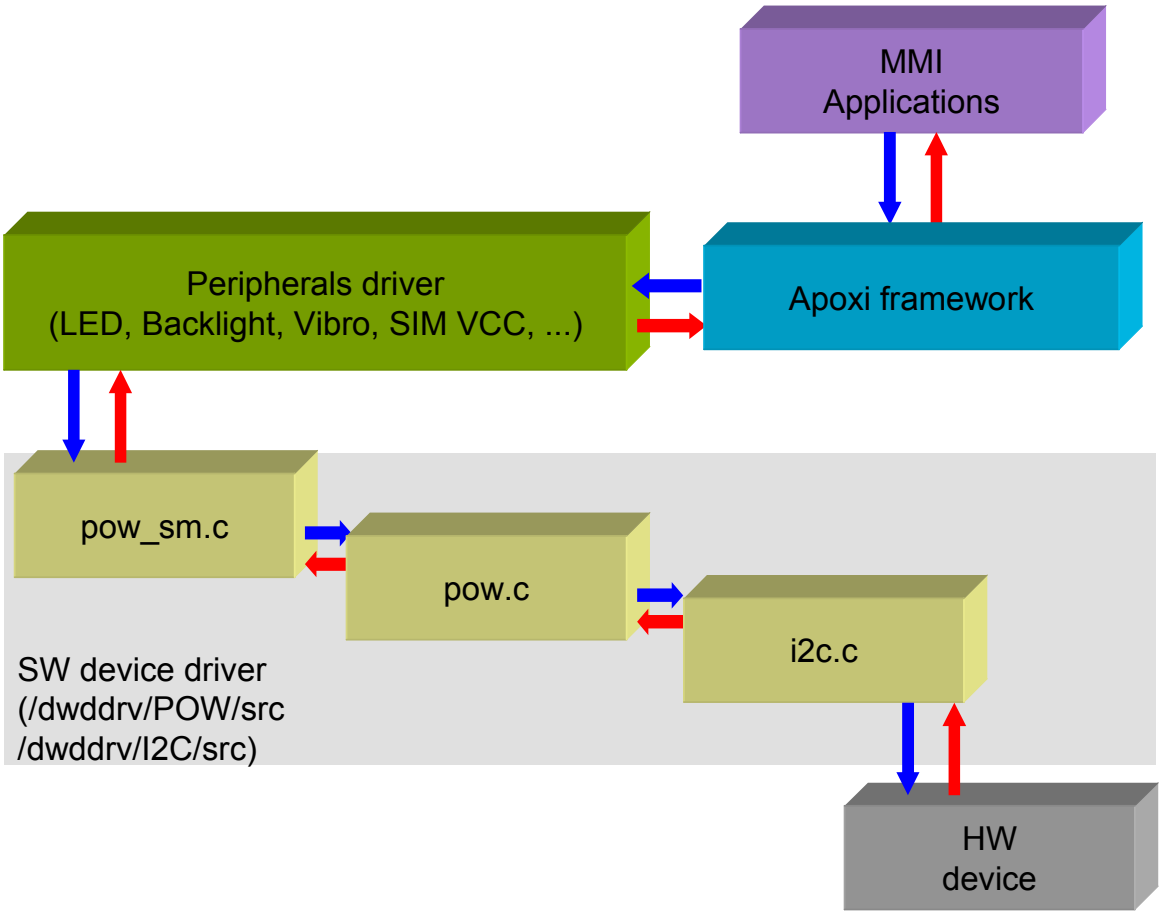

Figure 4-1

4.1 /dwddrv/I2C/i2c.c – IIC bus software device driver

Communication between E-GOLDlite and Power Chip is provided through IIC bus. This is the lowest level of code and works directly with two pins of E-GOLDlite, which are used as SCL and SDA. (See IIC bus technical reference for details). The software driver uses following functions, to implement all basic functions of IIC bus and to managet the communications with E-POWERlite. The driver also manages communications with other peripherals, such as a camera and an external DAC, if they are supported.

void **IIC_Init_Pin** (void);
void **IIC_Start** (void);
void **IIC_Stop** (void);
BOOLEAN **IIC_SendByte** (UINT8 IIC_Byte);
UINT8 **I2C_Write** (UINT8 *buf, UINT8 nof_bytes);
UINT8 **I2C_Read** (UINT8 addr, UINT8 reg, UINT8 nof_byte);
UINT8 **I2C_Read_Byte** (UINT8 send_ack);
UINT8 **I2C_Init** (void);

IIC_Init_Pin provide bus initialization. The other functions provide writing and reading one or more data to device.
Parameter *buf* is used to store data which have to be transferred.
The device address is the first byte of buf, and is set in proper driver (in case of E-POWERlite in pow.c, function epWrite, see below)
addr is referred to register's address where *data* has to be stored
nof_bytes is the number of bytes to transfer in case of multiple Read or Write operations.

4.2 /dwddrv/POW/pow.c – Communication between E-GOLDlite and Power device

Just above IIC software driver, we have a set of instructions that manage Read and Write operations. These are used to have a simplest interface from the point of view of the programming side.

void **epCommInit**(void);
unsigned int **epWrite**(unsigned int control_reg_adress, unsigned int control_byte, unsigned int checksum_byte);
unsigned int **epConfirm**(unsigned int checksum_byte);
unsigned int **epRead**(unsigned int status_reg_adress);
void **epContinuedRead**(void);
unsigned int **Calculate_CRC**(unsigned int byte3, unsigned int byte2, unsigned int byte1);

These functions recall directly the lower level functions, to manage correctly the IIC bus protocol. From a programming point of view: "Write this *control_byte* to *control_reg_address* and check for *checksum_byte*".
Checksum is used to confirm the data transfer has been successful. For further informations please refer to IIC bus technical specifications or Infineon's Power Chip Technical reference.

4.3 /dwddrv/POW/src/pow_sm.c – Regulators and Peripherals management

This is the higher level of the driver. The function set provided is very rich and offers powerful control for all the features of device. To improve functionality, a local structure has defined which stores the values of all registers settings. The power device's control registers are accessible only in write mode, so we save an internal copy of the status of register, to be able to know exactly actual configuration. Structure definition is the following:

```
typedef struct {
        unsigned char ChipID;

        unsigned char resctrl;
```

```
        unsigned char pwrctrl1;
        unsigned char pwrctrl2;
        unsigned char chctrl1;
        unsigned char intctrl1;
        unsigned char intctrl2;
        unsigned char pwrctrl3;
        unsigned char chctrl2;
        unsigned char testctrl1;
        unsigned char ledctrl;
        unsigned char drvctrl;
        unsigned char audctrl2;
        unsigned char smpaudctrl;
        unsigned char pwrctrl4;
        unsigned char testctrl2;
        unsigned char testctrl3;
        unsigned char pwrctrl5;
        unsigned char pwrctrl6;
        unsigned char ledctrl2;
        unsigned char testctrl4;
} PWR_device_control_status_type;
```

4.3.1 Auto detection of device.

Power management can be provided from S/M-POWER v 1.0, 1.5 or E-POWERlite. They offer similar features, but also they differs each other for few registers definition and bit mapping. For this reason they MUST be programmed in different ways. Another possible scenario is using of different version of chipset for the same board project.
To avoid damage to system or system malfunction, is suggested to provide auto detection of chipset version, to check the device at runtime. Following functions stores in **PWR_dev_status**.ChipID a value read from ID register of device.

unsigned int **epGet_device_version**(void);

The function is called at startup (through *epInitialSetup()*), in the form:

```
PWR_dev_status.ChipID = epGet_device_version();
```

4.3.2 Power supply control

E-POWERlite chipset is equipped with a set of control register that allows full control of power supply. Main control register are numbered from 1 to 6, and control each one a different power regulator. The register RESET, controls POWER OFF and SW generated reset of whole system.

```
 unsigned int epPWR1_set(short lbbvolt, short vanastatus, short settings1, short
settings2);
 unsigned int epPWR2_set(short pwmforce, short lwpwmode, short shcprot, short
outvsel);
 unsigned int epPWR3_set(unsigned int lbb, unsigned int lrf2set, unsigned int
lrf3set);
 unsigned int epPWR4_set(short lrf0set, short lrf1set);
 unsigned int epPWR5_set(short setsim12);
 unsigned int epPWR6_set(short setmmc);
 unsigned int epVANA_set(short on);
```

Full functionality and parameter description are described in the following tables. Each table describes function's parameters and examples.

unsigned int **epPWR1_set**(short lbbvolt, short vanastatus, short settings1, short settings2)

Manage settings for POWERCTRL1 register(LANA). For S/M-POWER v1.0 set also SIM mode and MMC. For S/M-POWERv1.5 set also LINT and LBB1&2 voltage levels

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	**lbbvolt**	short	**0**	Set LBB to 1.5V	
			1	Set LBB to 1.65V	
ALL	**vanastatus**	short	**0**	Set ANA off	
			1	Set ANA on	
SMP 1.0	**settings1**	short	**0x**	Set SIM off	
			1x	Set SIM on	
			x0	Set SIM2 off	
			x1	Set SIM2 on	
SMP 1.0	**settings2**	short	**0x**	VMMC set to 2.85V	
			1x	VMMC set to 1.8V	
			x0	MMC off	
			x1	MMC on	
SMP 1.5 EPWRLT	**settings1**	short	**0**	LINT standby	
			1	LINT on	
SMP 1.5 EPWRLT	**settings2**	short	**00**	LBB1 off LBB2 off	
			01	LBB1 on if VCXOEN=1 LBB2 on	
			10	LBB1 off LBB2 on	
			11	LBB1 on LBB2 on	

Table 4-1

epPWR2_set(short pwmforce, short lwpwmode, short shcprot, short outvsel)					
Set up PWRCTRL2 register regulators (SDBB mode and voltage levels)					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	**pwmforce**	short	**00x**	forced PWM mode	
			01x	switch to PFM if VCXOEN=0	
			11x	forced PFM mode	
			yy0	automatic switchback to PWM mode disabled	
			yy1	enabled	
ALL	**lwpwmode**	short	**0**	low power mode disabled	
			1	enabled	
ALL	**shcprot**	short	**0**	short circuit protection set to 650 mA	
			1	set to 850 mA	
ALL	**outvsel**	short	**00**	VDDC set to 1.50V	
			01	VDDC set to 1.80V	
			10	VDDC set to 1.86V	
			11	VDDC set to 1.92V	

Table 4-2

epPWR3_set(unsigned int lbb, unsigned int lrf2set, unsigned int lrf3set)					
Set up PWRCTRL3 register regulators (LRF2 and LRF3 voltage levels and mode). For S/M-POWER v1.0 also controls LBB1&2 mode. For E-POWERlite LRF3 is not provided					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
SMP 1.0	lbb	unsigned int	**00**	LBB1 off LBB2 off	
			01	LBB1 off LBB2 on	
			10	LBB1 on if VCXOEN=1 LBB2 on	
			11	LBB1 on LBB2 on	
SMP 1.0	lrf2set	unsigned int	**0xx**	VRF2 set to 2.85V	
			1xx	VRF2 set to 2.70V	
SMP 1.5	lrf2set	unsigned int	**0xx**	VRF2 set to 2.5V	
			1xx	VRF2 set to 2.70V	
EPWRLT	lrf2set	unsigned int	**0xx**	VRF2 set to 2.7V	
			1xx	VRF2 set to 2.5V	
ALL	lrf2set	unsigned int	**y00**	LRF2 off	
			y01	LRF2 on if VCXOEN=1	
			y10	LRF2 on if VCXOEN=1 or LRF3EN=1	
			y11	LRF2 on	
			0xx	VRF3 set to 2.65V	
			1xx	VRF3 set to 1.80V	
SMP 1.0 **SMP 1.5**	lrf3set	unsigned int	**y00**	LRF3 off	
			y01	LRF3 on if VCXOEN=1	
			y10	LRF3 on if LRF3EN=1	
			y11	LRF3 on	

Table 4-3

epPWR4_set(short lrf0set, short lrf1set)					
Set up PWRCTRL4 register regulators (LRF0/LRFC LRF1 voltage levels and mode). In S/M-POWER 1.0 lrf0set bit field has 1 more bit to program Voltage level for LRF0					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
SMP 1.0	lrf0set	short	**0xx**	set LRF0 to 2.5V	
			1xx	set LRF0 to 1.5V	
ALL	lrf0set	short	**y00**	LRF0 off	
			y01	LRF0 on if VCXOEN=1	
			y10	LRF0 on if LRF3EN=1	
			y11	LRF0 on	
ALL	lrf1set	short	**01xx**	set RF1 to 2.5V	
			10xx	set RF1 to 2.7V	
			11xx	set RF1 to 2.85V	
			yy00	LRF1 off	
			yy01	LRF1 on if VCXOEN=1	
			yy10	LRF1 on if LRF3EN=1	
			yy11	LRF1 on	

Table 4-4

epPWR5_set(short setsim12)					
Set up PWRCTRL5 register regulators (SIM1 & SIM2 voltage levels and mode). Supported only in S/M-POWER 1.5 and E-POWERlite					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
SMP 1.5 **EPWRLT (ONLY SIM regulator)**	setsim12	short	**0yyywzz**	VSIM2 set to 2.85V	
			1yyywzz	VSIM2 set to 1.8V	
			x000wzz	LSIM2 off	
			x111wzz	LSIM2 on	
			xyyy0zz	VSIM set to 2.85V	
			xyyy1zz	VSIM set to 1.8V	
			xyyyw00	LSIM off	
			xyyyw01	LSIM on if VCXOEN=1	
			xyyyw10	LSIM stdby	
			xyyyw11	LSIM on	

Table 4-5

epPWR6_set(short setmmc)

Set up PWRCTRL6 register regulators (MMC voltage levels and mode). Supported only in S/M-POWER v1.5 and E-POWERlite

CHIPSET Version	Parameter Name	Type	Values	Function	Example
SMP 1.5 **EPWRLT**	setmmc	short	**00yy**	MMC set to 2.85V	
			01yy	MMC set to 2.05V	
			10yy	MMC set to 2.50V	
			11yy	MMC set to 1.80V	
			xx00	MMC off	
			xx01	MMC stdby if VCXOEN=1	
			xx10	MMC stdby	
			xx11	MMC on	

Table 4-6

epVANA_set(short on)

Set on/off the LANA regulator. Used by audio driver to correctly power up and down the amplifier.

CHIPSET Version	Parameter Name	Type	Values	Function	Example
EPWRLT	on	short	**0**	LANA off	
			1	LANA on	

Table 4-7

4.3.3 Peripherals Control

Also a set of peripherals are provided inside Power chip. In details two LED channel could be driven, a backlight source, with programmable duty cycle, an audio amplifier with programmable muting and audio gain (and programmable common mode voltage in E-POWERlite). A driver for vibrator voltage is also integrated. The function set is:

```
epLED_SET(int Led_mask,int LED_settings);
epLED_PWM_SET(int LED_settings);
epLED_PWM_switch(ubyte LED_onoff);
POW_setVIBRATORCTRL(int pvvib);
POW_getVIBRATORCTRL(void);
epAudio_SET(int Gain, short Mute, short AudioON);
epLED_RED_Set(unsigned char on);
epLED_GREEN_Set(unsigned char on);
```

epLED_SET(int Led_mask, int LED_settings)

Sets LED 1 & 2 ON or OFF (for signaling). Also program LED driver current (for backlight)

CHIPSET Version	Parameter Name	Type	Values	Function	Example

ALL	Led_mask	int	**00**	LED1&2 off	
			01	LED1 on	
			10	LED2 on	
			11	LED1&2 on	
ALL	LED_settings	int	**000**	Power down LED	
			001	LED current = 20mA	
			010	LED current = 40mA	
			011	LED current = 60mA	
			100	LED current = 80mA	
			101	LED current = 100mA	
			110	LED current = 120mA	
			111	LED current = 140mA	

Table 4-8

epLED_PWM_SET(int LED_Settings)					
Set up the duty cycle of backlight LED driver channel					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	LED_Settings	int	0..63	Duty cycle of LED driver current (for backlight) set to 0.92^{value}	

Table 4-9

epLED_PWM_switch(ubyte LED_onoff)					
Switch on or off the backlight LED driver channel					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	LED_onoff	ubyte	0..1	Specifies if LED driver must be switched on or off	

Table 4-10

POW_setVIBRATORCTRL(int pvvib)；					
Set up the Voltage for vibro driver					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	pvvib	int	**0000..1000**	Value for Vib voltage 0.9 to 2.0 100mV step (0 is off)	

Table 4-11

POW_getVIBRATORCTRL(void)					
Return status of vibro driver level					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	void	-	-	-	

Table 4-12

epVIB_SET(int VIB_settings)					
Manage vibrator driver: S/M-POWER 1.5 has more voltage levels respect to S/M-POWER<=v1.0					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
SMP 1.0	VIB_settings	int	**000..100**	0.9V to 1.5V 100mV step (0 is off)	
SMP 1.5 **EPWRLT**	VIB_settings	int	**0000..1000**	0.9 to 2.0 100mV step (0 is off)	

Table 4-13

epAudio_SET(int Gain, short Mute, short AudioON)					
Sets built in audio amplifier: gain, muting and on/off status.					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL **(EPWRLT differs)**	Gain	int	**00**	Set gain to -6.0dB	
			01	Set gain to -1.2dB	
			10	Set gain to +2.7dB	
			11	Set gain to +6.0dB (.7.6dB in EPWRLT)	
ALL	Mute	short	**0**	Mute is off	
			1	Mute is on (no sound)	
ALL	AudioON	short	**0**	Amp is off	
			1	Amp is on	

Table 4-14

epAudio2_SET(short AudCM, short AudBOOST)					
Set up Common mode Voltage and BOOST mode. ****E-POWERLITE only****					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
EPWRLT	AudCM	short	**0**	Set Common Mode to 1.575V	
			1	Set Common Mode to	

				1.650V	
			2	Set Common Mode to 1.725V	
			3	Set Common Mode to 1.800V	
			4	Set Common Mode to 1.875V	
			5	Set Common Mode to 1.950V	
			6	Set Common Mode to 2.025V	
			7	Set Common Mode to 2.100V	
EPWRLT	AudBOOST	short	**0,1,3**	Normal mode	
			2	Boost mode on (Gain=12dB)	

Table 4-15

epLED_RED_Set(unsigned char on)					
Turns on or off the signaling LED2 (red in GLOBE2 platform)					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	on	unsigned char	0..1	on/off state for signaling LED1	

Table 4-16

epLED_GREEN_Set(unsigned char on)					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	on	unsigned char	0..1	on/off state for signaling LED2	

Table 4-17

This first group of functions set the whole register value, overwriting all bits. Following functions control only a part of the register, using the global structure **PWR_dev_status** for preserving bits that are not involved in settings.

```
unsigned int epSIMSetVolt_value(unsigned char Voltage);
unsigned int epLEDSwitch(unsigned char value);
unsigned int epLEDsetcurrent(unsigned char value, signed char mode);
unsigned int epLEDsetpwm(unsigned char value, signed char mode);
unsigned int epAudio_switch_onoff(void);
```

unsigned int **epSIMSetVolt_value**(unsigned char Voltage)

This function selects the voltage for SIM regulators (support for 1.8V SIM technology)

NOTE: This functionality is provided only by S/M-POWER v1.5 and E-POWERlite Chipset

CHIPSET Version	Parameter Name	Type	Values	Function	Example
SMP 1.5 **EPWRLT**	Voltage	unsigned char	**0**	SIM set to 2.85V	
			1	SIM set to 1.8V	

Table 4-18

unsigned int **epLEDSwitch**(unsigned char value)

Toggles on/off the status of SLED1 and SLED2 output

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	value	unsigned char	**00**	Toggle nothing	
			01	Toggle SLED1	
			10	Toggle SLED2	
			11	Toggle both	

Table 4-19

unsigned int **epLEDsetcurrent**(unsigned char value, signed char mode)

Set LED only current value (for backlight). Does not modify SLED1&2 status. Also can increment or decrement actual current setting

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	value	unsigned char	**000**	Turn OFF	
			001	Set LED current to 20mA	
			010	Set LED current to 40mA	
			011	Set LED current to 60mA	
			100	Set LED current to 80mA	
			101	Set LED current to 100mA	
			110	Set LED current to 120mA	
			111	**Set LED current to 140mA**	
ALL	mode	signed char	**0**	Set to value	

			+1	Increment by value	
			-1	Decrement by value	

Table 4-20

unsigned int **epLEDsetpwm**(unsigned char value, signed char mode)					
Set PWM for backlight. Very similar to epLED_PWM_SET (See Table 4-9 *pag.15)*					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	value	unsigned char	**0..63**	Set duty cycle to 0.92^{value}	
ALL	mode	signed char	**0**	Set to value	
			+1	Increment by value	
			-1	Decrement by value	

Table 4-21

unsigned int **epAudio_switch_onoff**(void)					
Toggles audio amplifier on and off					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	---	void	---	---	

Table 4-22

4.3.4 Other functions

Following functions are maintained for compatibility with previous version of code, but should not be used for S/M-POWER and E-POWERlite based board. The new set of functions described before, provides same functionality, also improved. **In this paper the following set is NOT documented.**

```
unsigned int epEnableCharger(void);
unsigned char epVoltSelect207(void);
unsigned int epDisableWatchdogTimer(void);
unsigned int epReadID(unsigned int max_size, unsigned char p_id);
unsigned int epForcePFM_Mode(void);
unsigned int epForcePWM_Mode(void);
```

Another set of special function, is reported below.

```
unsigned int epRESET(short resdn, short res);
unsigned int epAllOff(void);
unsigned int epInitialSetup(void);

unsigned int epEnableSimVoltage(void);
unsigned int epDisableSimVoltage(void);
```

The other two functions are self explicative. They are called from SIM activate/deactivate procedures.

unsigned int **epRESET**(short resdn, short res)

Generates a SW reset

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	resdn	short	0 or 1	if set to 1 generates EXT reset on POWERDOWN	
ALL	res	short	0 or 1	set to 1 to generates EXTERNAL + INTERNAL reset	

Table 4-23

unsigned int **epAllOff**(void)

Turns off all regulators except RTC, and is called when the system must shutdown.

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	---	void	---	---	

Table 4-24

unsigned int **epInitialSetup**(void)

Called once during bootstrap of the system and initialize all regulators to their start values and status.

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	---	void	---	---	

Table 4-25

unsigned int **epEnableSimVoltage**(void)

Turn on SIM regulator (settings for Voltage value must be done before)

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	---	void	---	---	

Table 4-26

unsigned int **epDisableSimVoltage**(void)

Turn off SIM regulator

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	---	void	---	---	

Table 4-27

4.3.5 Battery Charger source and Interrupt.

S/M-POWER and E-POWERlite can be programmed to recharge different type of battery, providing different Voltage and current values. Also may generate INTERRUPT signals when a change occurs in one of device's status registers (accessible Read only mode). Also this feature is programmable using following function set:

```
unsigned int epCHRG1_SET(short chrgset, short chrgmode, short vmax, short vlim,
short plslen);
unsigned int epCHRG2_SET(short chrgrvm, short chrgpreset, short currlim);

unsigned int epINT1_SET(int command);
unsigned int epINT2_SET(int command);
```

unsigned int **epCHRG1_SET**(short chrgset, short chrgmode, short vmax, short vlim, short plslen)

Setup charger ON/OFF, modes and parameters in S/M-POWER and E-POWERlite

CHIPSET Version	Parameter Name	Type	Values	Function	Example
ALL	chrgset	short	0	Charger OFF	
			1	Charger ON	
ALL	chrgmode	short	0	Set continuous MODE	
			1	Set pulse MODE	
ALL	vmax	short	0	Max Voltage value set to VCHMax1 (4.1V – Li-Ion battery)	
			1	Max Voltage value set to VCHMax2 (4.2V - Li-Ion battery)	
			2	RESERVED	
			3	Max Voltage value set to VCHMax3 (5.15V – Ni-MH battery)	
ALL	vlim	short	0	Shut down	
			1	Voltage limit in pulse charge	
ALL	plslen	short	0	pulse length 2.048ms	
			1	pulse length 4.096ms	
			2	pulse length 8.192ms	
			3	pulse length 16.4ms	
			4	pulse length 32.8ms	
			5	pulse length 65.5ms	
			6	pulse length 131ms	
			7	pulse length 262ms	

Table 4-28

unsigned int **epCHRG2_SET**(short chrgrvm, short chrgpreset, short currlim)					
Set CHRCTRL2 register for managing charger functionality: level for charging current measurement, precharging and charging current limit.					
CHIPSET Version	**Parameter Name**	**Type**	**Values**	**Function**	**Example**
ALL	chrgrvm	short	**0 - 7**	Sets chrgvm*100mA (from 0mA to 700mA)	
ALL	chrgpreset	short	**0**	Precharging on	
			1	Precharging off	
SMP 1.5 EPWRLT	currlim	short	**0 - 7**	Set battery charge current limit from 400mA (0) to 1100mA (7), steps of 100mA	

Table 4-29

unsigned int **epINT1_SET**(int command)					
Set INTCTRL1 register for enable (1) or disable (0) chip interrupts. *Use command= EICHV | EICHMD | ... | INTMD; due to which interrupt(s) you want to enable*					
CHIPSET Version	**BIT name**	**Type**	**Values**	**Function**	**Example**
ALL	EICHV	bitmask	**10000000 [Hex:80]**	Enable interrupt on remove or attach of charging voltage	
ALL	EICHMD	bitmask	**01000000 [Hex:40]**	Enable interrupt on change of charge mode	
ALL	EICHCAL	bitmask	**00100000 [Hex:20]**	Enable interrupt on current level triggered	
ALL		bitmask		NOT IMPLEMENTED (used in SMP 1.5 for USB)	
ALL	EISPWM	bitmask	**00001000 [Hex:08]**	Enable interrupt on switchback to PWM mode	
ALL	EIORLMMC	bitmask	**00000100 [Hex:04]**	Enable interrupt on MMC out of regulation	
ALL	EIORLSIM	bitmask	**00000010 [Hex:02]**	Enable interrupt on LSIM out of regulation	
ALL	INTMD	bitmask	**00000001**	Select sensitiveness of	

			[Hex:01]	interrupt on Falling (0) or rising (1) edge	

Table 4-30

unsigned int **epINT2_SET**(int command)					
Set INTCTRL2 register for enable (1) or disable (0) chip interrupts. Also enable or disable shutdown and protection features. *Use command= EIOTW | RAGOTW | ... | DEBUG; due to which function you want to enable*					
CHIPSET Version	**BIT name**	**Type**	**Values**	**Function**	**Example**
ALL	EIOTW	bitmask	**10000000** **[Hex:80]**	Enable interrupt on remove or attach of charging voltage	
ALL	RAGOTW	bitmask	**01000000** **[Hex:40]**	Enable interrupt on change of charge mode	
ALL			**00100000** **[Hex:20]**	Enable interrupt on current level triggered	
ALL	OTSEN	bitmask		NOT IMPLEMENTED (used in SMP 1.5 for USB)	
ALL			**00001000** **[Hex:08]**	RESERVED	
ALL			**00000100** **[Hex:04]**	RESERVED	
ALL	EION	bitmask	**00000010** **[Hex:02]**	Enable interrupt on change of level of pin ON	
ALL	DEBUG	bitmask	**00000001** **[Hex:01]**	Select sensitiveness of interrupt on Falling (0) or rising (1) edge	

Table 4-31

4.3.6 Phone Tool support.

Starting from Phone Tool version 9.8, a specific form for totally control E-POWERlite device is available. Such form dialogs directly with driver's API. To do this a particular inter communication interface is provided. See Figure 4-2 below.

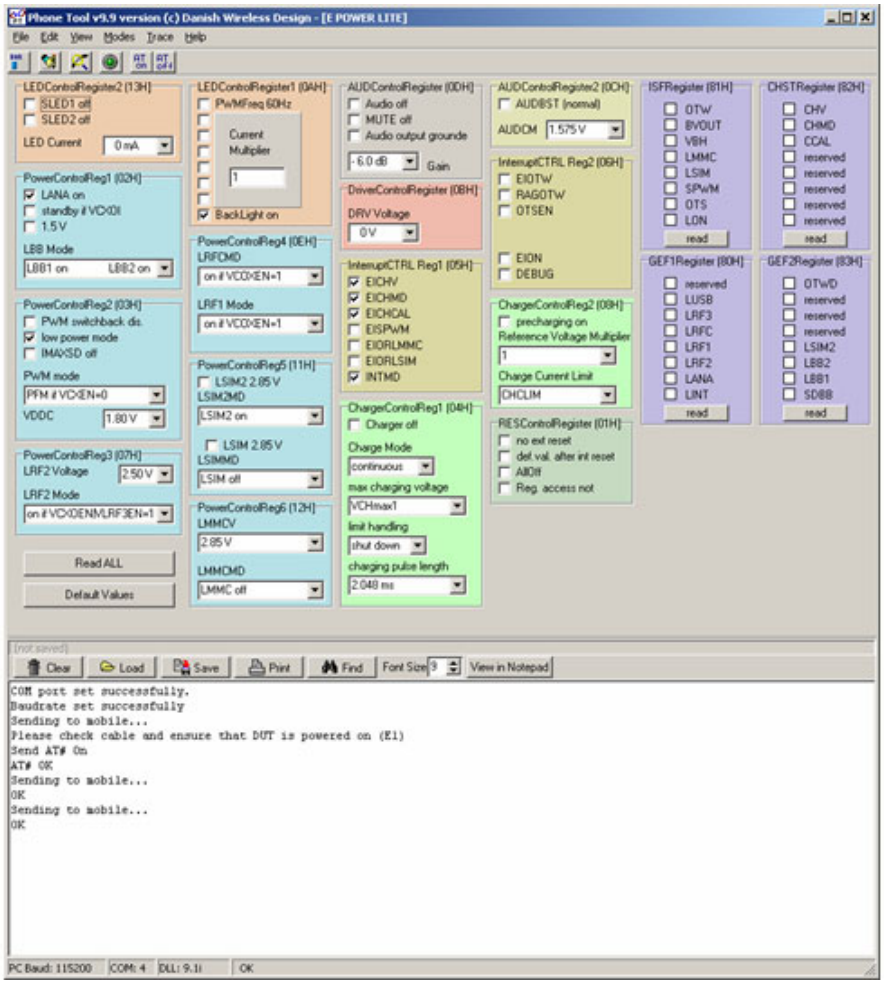

Figure 4-2

Three functions are added.

```
POW_power_chip_version_enum epGet_deviceID(void)
unsigned int POW_epRead(unsigned char epRegister)
unsigned int16 POW_ptest_generic_func(atctst_pow_man_generic_func_req_type
*func_req_ptr)
```

where

```
typedef enum {
    POW_chip_version_not_detected,
    POW_chip_version_failure,
    POW_chip_version_10,
    POW_chip_version_15,
    POW_chip_version_EPlite
} POW_power_chip_version_enum;
```

unsigned int **epGet_deviceID**(void)
Called from Phone Tool when upgrading MS information in general (main) form.

CHIPSET Version	Parameter	Type	Values	Function	Example
ALL		void			

Table 4-32

unsigned int **POW_epRead**(unsigned char epRegister)					
Called by atct command parser. Returns the actual state of device register epRegister					
CHIPSET Version	**Parameter**	**Type**	**Values**	**Function**	**Example**
ALL	epRegister	unsigned char		Specify which address has to be read	

Table 4-33

unsigned int16 **POW_ptest_generic_func**(atctst_pow_man_generic_func_req_type *func_req_ptr)					
Provides atct command parsing. This is specific for power management commands. See production_test_pc_dll_v7.3.pdf [1] *document for further information on DWD IO DLL*					
CHIPSET Version	**Parameter**	**Type**	**Values**	**Function**	**Example**
ALL	func_req_ptr	atctst_pow_man_generic_func_req_type (pointer)		See 1 for details	

Table 4-34

[1] This document is available under Clear Case in vob \dwdtools\dwdio\Documentation\Production_test_pc_dll\

5 References

5.1 External

[1] PMB6814 E-POWERlite Rev 2.0 2003-11-09; Infineon Technologies - Target specifications of device

[2] production_test_pc_dll_v7.3.pdf ; Danish Wireless Design (DWD) - Specification on DWD IO interface DLL

5.2 Internal

Title	Doc ID

6 Document change report

	Change Reference		Record of changes made to previous released version	
Rev	**Date**	**CR**	**Section**	**Comment**
1.0	26/05/2004			Document created
1.1	08/09/2004		4.3.6-Phone Tool support.	Added section (entirely)
			Table 4-7 Table 4-16 Table 4-17	Added
1.2	24/02/2005		N7 Form	Updated
			3 - Introduction	Added last lines: notes about chipset types support
2.0	01/02/2006		Whole document	Change for BP30 platform
			Figure 4-1	Changed
			4.1 /dwddrv/I2C/i2c.c – IIC bus software device driver	Updated to new filename and location – Changed interface
			4.2 /dwddrv/POW/pow.c – Communication between E-GOLDlite and Power device	Updated to new filename and location
			4.3 /dwddrv/POW/src/pow_sm.c – Regulators and Peripherals management	Updated to new filename and location – Extended Interface
			4.3.3 Peripherals Control	Table 4-10 added for epLED_PWM_switch (added) Added *POW_setVIBRATORCTRL* *POW_getVIBRATORCTRL* and related Table 4-11 & Table 4-12

7 Approval

Revision	Approver(s)	Date	Source/signature
1.0	Stefano Godeas	26/05/2004	Document stored on server

1.1	Stefano Godeas	08/09/2004	Document stored on server
1.2	Stefano Godeas	24/02/2005	Document stored on server
2.0	Stefano Godeas	01/02/2006	Document stored on server

8 Annex 1

None.