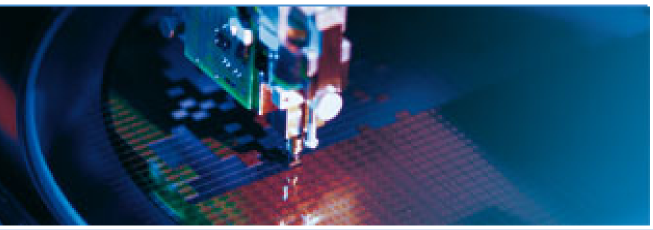
APPLICATION NOTE



An0061

The Guardian

Rev 2.3 – 28 July 2011 – ETH_An0061_Rev2.3

www.eurotech.com

EUROTECH

(This page has been intentionally left blank)

Trademarks

All trademarks both marked and unmarked appearing in this document are the property of their respective owners.

Revision History

REVISION	DESCRIPTION	DATE
2.0	First release	March 2008
2.1	Complete registers review	September 2008
2.2	Added registers: • VIN_SHUTDOWN_REG • VIN_PERIOD_REG Updated registers: • PWM_TB_REG	February 2008
2.3	Updated “0x32h” in “0x324h” at page 9	28 July 2011

(This page has been intentionally left blank)

Table of Contents

(This page has been intentionally left blank)

Chapter 1 The Guardian

DuraCOR systems come with a new integrated intelligence feature called "The Guardian".
It allows users to monitor the systems operating parameters and to manage the command execution logic even when the system is in standby mode.

“The Guardian”, allows for the following:

- Verification of voltages and internal temperatures; signalling alarms to the CPU
- Storage of the system status LOG an internal EEPROM
- Management of power-up and power-down of the system

“The Guardian” monitoring features

The Guardian is based on the 8051 compatible C8051F310 micro-controller (µC) and features the following:

- Odometer counter
- PWM signal generator
- Countdown monitor
- Voltage enabling logic
- Command execution logic
- Internal voltage monitor
- Internal temperature monitor

Once a second the Guardian checks that the internal voltage and temperature values are within permitted ranges; if these conditions are not realized the event is stored within an internal EEPROM and, if desired, an interrupt request is activated. (The description is valid for: Guardian: 1.12 or higher, Bootloader 1.5 or higher).

“The Guardian” Registers

The Guardian is managed with status and control registers implemented within it.
These registers are 1 byte long as follows:

REGISTER NAME	DESCRIPTION	ADDRESS
REV_L	Firmware revision, least significant byte	0x00
REV_H	Firmware revision, most significant byte	0x01
COMMAND_REG	Register command	0x02
SCRATCH_REG	Register scratch	0x03
CNT_TRIG_REG	Odometer trigger control	0x04
CNT_REG_H	Odometer counter, most significant byte	0x05
CNT_REG_L	Odometer counter, least significant byte	0x06
PWM_TB_REG	PWM time base control	0x07
PWM_DUTY_REG_H	PWM duty cycle control, most significant byte	0x08
PWM_DUTY_REG_L	PWM duty cycle control, least significant byte	0x09
SHUTDOWN_DELAY	Retarded shutdown control	0x0A
STATUS	µC status register	0x0B
VDD_MIN_LOW_REG	VDD minimum limit, least significant byte	0x0C
VDD_MIN_HIGH_REG	VDD minimum limit, most significant byte	0x0D
VDD_MAX_LOW_REG	VDD maximum limit, least significant byte	0x0E
VDD_MAX_HIGH_REG	VDD maximum limit, most significant byte	0x0F
VIN_MIN_LOW_REG	VIN minimum limit, least significant byte	0x10
VIN_MIN_HIGH_REG	VIN minimum limit, most significant byte	0x11
VIN_MAX_LOW_REG	VIN maximum limit, least significant byte	0x12

REGISTER NAME	DESCRIPTION	ADDRESS
VIN_MAX_HIGH_REG	VIN maximum limit, most significant byte	0x13
VCC3_MIN_LOW_REG	VCC3 minimum limit, least significant byte	0x14
VCC3_MIN_HIGH_REG	VCC3 minimum limit, most significant byte	0x15
VCC3_MAX_LOW_REG	VCC3 maximum limit, least significant byte	0x16
VCC3_MAX_HIGH_REG	VCC3 maximum limit, most significant byte	0x17
V12_MIN_LOW_REG	12V minimum limit, least significant byte	0x18
V12_MIN_HIGH_REG	12V minimum limit, most significant byte	0x19
V12_MAX_LOW_REG	12V maximum limit, least significant byte	0x1A
V12_MAX_HIGH_REG	12V maximum limit, most significant byte	0x1B
MICRO_TEMP_MIN_LOW_REG	Micro-controller temperature minimum limit, least significant byte	0x1C
MICRO_TEMP_MIN_HIGH_REG	Micro-controller temperature minimum limit, most significant byte	0x1D
MICRO_TEMP_MAX_LOW_REG	Micro-controller temperature maximum limit, least significant byte	0x1E
MICRO_TEMP_MAX_HIGH_REG	Micro-controller temperature maximum limit, most significant byte	0x1F
VCC3_ALW_MIN_LOW_REG	VCC_Always minimum limit, least significant byte	0x20
VCC3_ALW_MIN_HIGH_REG	VCC_Always minimum limit, most significant byte	0x21
VCC3_ALW_MAX_LOW_REG	VCC_Always maximum limit, least significant byte	0x22
VCC3_ALW_MAX_HIGH_REG	VCC_Always maximum limit, most significant byte	0x23
OUT_OF_RANGE_STATUS_REG	Status of the monitored values	0x24
MICRO_TEMP_LOW_REG	Micro-controller temperature value, least significant byte	0x25
MICRO_TEMP_HIGH_REG	Micro-controller temperature value, most significant byte	0x26
FLASH_ADDR_H_REG	Flash reading address, least significant byte	0x27
FLASH_ADDR_L_REG	Flash reading address, most significant byte	0x28
FLASH_DATA_REG	Flash reading data	0x29
VDD_LOW_REG	VDD value, least significant byte	0x2A
VDD_HIGH_REG	VDD value, most significant byte	0x2B
VIN_LOW_REG	VIN value, least significant byte	0x2C
VIN_HIGH_REG	VIN value, most significant byte	0x2D
VCC3_LOW_REG	VCC3 value, least significant byte	0x2E
VCC3_HIGH_REG	VCC3 value, most significant byte	0x2F
V12_LOW_REG	V12 value, least significant byte	0x30
V12_HIGH_REG	V12 value, most significant byte	0x31
VCC3_ALW_LOW_REG	VCC3_ALWAYS value, least significant byte	0x32
VCC3_ALW_HIGH_REG	VCC3_ALWAYS value, most significant byte	0x33
EEPROM_ADDR_REG	EEPROM address	0x34
EEPROM_DATA_REG	EEPROM data	0x35
EEPROM_STATUS_REG	EEPROM status	0x36
VIN_SHUTDOWN_REG	Counter for voltage limit	0x37
VIN_PERIOD_REG	Frequency period for voltage checking	0x38

Register Access

Access to the registers is done through an 8-Bit I/O port (by default mapped at 0x324h) on the ISA Bus; both reading and writing operations are possible.

To transmit a byte to the µC users should write it in the GUARDIAN_INTERFACE register. At the same time as this transmission, the µC will transmit a response byte to the CPU also in the GUARDIAN_INTERFACE register. A read operation in this register will allow you to get the received byte.

Read and write operations follow a protocol that controls the transmission of byte sequences.

Each sequence must be separated by a time interval of greater than 10ms.

The read operation of a register is made following this transmit/receive sequence:

	1ST BYTE	2ND BYTE	3RD BYTE	4TH BYTE
TX (to the µC)	'R' (0x52)	*Addr*	*Dummy*	*Dummy*
RX (from the µC)	*Dummy*	'R' (0x52)	*Addr*	*Data*

"Addr" is the address of the register to be read and "*Data"* is the read data. The µC performs an echo with the first 2-bytes received; the CPU can use this to validate the data transmitted.

The write operation of a register is made following this transmit/receive sequence:

	1^{ST} BYTE	2^{ND} BYTE	3^{RD} BYTE	4^{TH} BYTE
TX (to the µC)	'W' (0x57)	*Addr*	*Dummy*	*Dummy*
RX (from the µC)	*Dummy*	'W' (0x57)	*Addr*	*Data*

"Addr" is the address of the register to be written to and "*Data"* is the data to be written. The µC performs an echo with the first 3-bytes received; the CPU can use this to validate the data transmitted.

In the case that a sequence starts incorrectly (i.e. using bytes other than 'R' or 'W') the µC replies with the byte 'N' (0x4E) at the next transmission and then waits for a new sequence.

Registers can be accessed by reading [R], writing [W] or by write-only [WO] operations.

Some registers are set with a default value when the firmware is started (and the Guardian µC is powered on). This default value can be:

- A value set by Eurotech (described in more detail in the following paragraphs), if there are not any default values to load in the registers stored in the non-volatile memory of the µC
- A value previously stored by the application software in the non-volatile memory of the µC (Refer to the SAVE_DEFAULT command)

(This page has been intentionally left blank)

Chapter 2 Register description

REV_L & REV_H

The REV_L and REV_H registers contain the revision number of the µC firmware:

- "H" is the high byte contained in REV_H
- "L" is the low byte contained in REV_L

	REV_L							0X00
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	REV_L7 REV_L0							
Default	Depends on the revision							
Access	R							

- Allowable values: 0 to 254 (255 reserved)

	REV_H							0X01
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	REV_H7	REV_H6 REV_H0						
Default	0	Depends on the revision						
Access	R	R						

- Allowable values: 0 to 127

Bit 7 of REV_H is 0 during the Bootloader program execution.
This allows the CPU to verify which program (Bootloader or runtime) is running by reading bit 7.

COMMAND

The Guardian can execute the commands sent from the CPU.
A command is given by writing its corresponding code to the COMMAND register:

	COMMAND							0X02
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	COMMAND7 to COMMAND0							
Default	0							
Access	R/W							

When the command execution ends, the register value becomes: NO_COMMAND.
The CMD_DONE register signals to the CPU that the previous execution has been completed.
The CMD_DONE register is activated when a command to execute is loaded in the COMMAND register and deactivated when the command has been executed.
Possible commands are as follows:

NAME	CODE	DESCRIPTION
NO_COMMAND	0x00	-
RESTART	0x01	Guardian firmware re-start
SAVE_DEFAULT	0x02	Save register default values in the non volatile memory
RUNTIME_INVALIDATE	0x03	Runtime firmware Invalidation
ERASE_DEFAULTS	0x04	Erase the user default values
SHUTDOWN	0x05	Starts the shutdown procedure
ENABLE_PWM	0x06	Enable the PWM out
DISABLE_PWM	0x07	Disable the PWM out
EXCLUDE_KEY	0x08	Disable the n-KEY signal control
IRQ_ACK	0x09	Acknowledge IRQ
POST_ERROR	0x0A	Reserved
CPU_TEMP_ERROR	0x0B	Save the CPU over-temperature error in the memory
WRITE_TEMP_OFFSET	0x0C	Write the temperature offset in the flash
READ_FLASH	0x0D	Read the logs from the µC Flash
READ_EEPROM	0x0E	Read the EEPROM
WRITE_EEPROM	0x0F	Write the EEPROM

NAME	DESCRIPTION
RESTART	Activates a software reset of the µC that will restart the firmware running in the Guardian.
SAVE_DEFAULT	Saves all the write accessible register values of the Guardian into the non-volatile memory of the µC (flash memory). Stored values will be reloaded at the next start up.
RUNTIME_INVALIDATE	Erases the runtime firmware validity key. Used during the µC firmware update process, refer to Firmware update on page 30
ERASE_DEFAULTS	Erases the default register values previously stored into the flash memory of the µC. At the next start up the registers will load the default values.
SHUTDOWN	Powers down the CPU module after a number of seconds as defined in the SHUTDOWN_DELAY register and clears Flag KEY_EX of STATUS register.
ENABLE_PWM	Enables the PWM output with the parameters contained in the registers PWM_TB_REG, PWM_DUTY_REG_H and PWM_DUTY_REG_L.
DISABLE_PWM	Disables the PWM output. Following is the sequence to change the PWM run-time output parameters: • Send a DISABLE_PWM command • Modify the PWM_TB_REG, PWM_DUTY_REG_H and PWM_DUTY_REG_L registers • Send an ENABLE_PWM command.
EXCLUDE_KEY	Disables the nKEY signal status control. When activating the EXCLUDE_KEY function it is advisable to enable the Watchdog to prevent the system from permanently locking up due to Operating System boot time errors.
IRQ_ACK	Disable the interrupt line.
POST_ERROR	Reserved
CPU_TEMP_ERROR	Reserved.
WRITE_TEMP_OFFSET	Save the temperature offset of the µC to the flash
READ_FLASH	Read a byte from the flash of the µC. The reading address must be programmed in the FLASH_ADDR_H_REG and FLASH_ADDR_L_REG register before the command is executed. The read data is stored in the READ_FLASH register.
READ EEPROM	Read a byte from the EEPROM connected to the µC. The read address must be programmed in the EEPROM_ADDR_REG register before the command execution. The read data is stored in the EEPROM_DATA_REG register.
WRITE_EEPROM	Write a byte in the EEPROM connected to the µC. The write address and the data to write must be programmed respectively in the EEPROM_ADDR_REG and EEPROM_DATA_REG registers before the command is executed.

SCRATCH

	SCRATCH							0X03
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	SCRATCH7 to SCRATCH0							
Default	X							
Access	R/W							

This scratch read / write accessible register has no effect on the power monitor. It has been developed to be used by the user as a service / debugging register.

CNT_TRIG_REG

	CNT_TRIG_REG							0X04
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	CNT_TRIG_REG7 ÷ CNT_TRIG_REG0							
Default	0x20							
Access	R/W							

The *CNT_TRIG_REG* register allows the selection of the odometer trigger parameters as follows:

- 0x10: trigger on transition 1→0
- 0x20: trigger on transition 0→1
- 0x30: trigger on both transitions

CNT_REG_H, CNT_REG_L

CNT_REG_H and *CNT_REG_L* contain the most significant byte and least significant bytes of the odometer counter

	CNT_REG_H							0X05
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	CNT_REG_H7 to CNT_REG_H 0							
Default	0							
Access	R/W							

	CNT_REG_L							0X06
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	CNT_REG_L7 to CNT_REG_L0							
Default	0							
Access	R/W							

When the counter reaches FFFF it automatically reverts back to 0, and the ODOV bit of the STATUS register is set to 1.

PWM_TB_REG

The *PWM_TB_REG* register allows you to select the PWM frequency time base.

	PWM_TB_REG							0X07
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	PWM_TB_REG7 to PWM_TB_REG0							
Default	0x02							
Access	R/W							

Allowable values are:

- 0x01: 31.79 Hz
- 0x02: 95.37 Hz

PWM_DUTY_REG_H and PWM_DUTY_REG_L

	PWM_DUTY_REG_H							0X08
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	PWM_DUTY_REG_H7 to PWM_DUTY_REG_H0							
Default	0x80							
Access	R/W							

	PWM_DUTY_REG_L							0X09
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	PWM_DUTY_REG_L7 to PWM_DUTY_REG_L0							
Default	0x00							
Access	R/W							

The *PWM_DUTY_REG_H* and *PWM_DUTY_REG_L* registers define the PWM duty cycle according to the following equation:

Duty cycle = (65536 - (*PWM_DUTY_REG_L* + *PWM_DUTY_REG_H* * 256)) / 65536

The maximum allowed duty cycle (100 %) is obtained with:

- PWM_DUTY_REG_L = PWM_DUTY_REG_H = 0

The minimum allowed duty cycle (0.0015%) is obtained with:

- PWM_DUTY_REG_L = PWM_DUTY_REG_H = 0xFF.

Default values for the Duty cycle = 50%.

SHUTDOWN_DELAY

	SHUTDOWN_DELAY							0X0A
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	SHUTDOWN_DELAY 7 to SHUTDOWN_DELAY0							
Default	0x05							
Access	R/W							

The *SHUTDOWN_DELAY* register defines how many seconds the CPU has to wait between the receipt of the *SHUTDOWN* command and the CPU shutdown.

STATUS

	STATUS							0X0B
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	IRQ5_EN	IO_SEL1	IO_SEL0	KEY_EX	SHUTD	ODOV	PWM	WDT
Default	0x80							
Access	R/W							

The *STATUS* register defines the current µC status:

STATUS	DEFINITION
WDT	µC has been reset by the watchdog timer
PWM	PWM out is active
ODOV	Odometer counter has overflowed 1 indicates that the counter has overflowed (>FFFF) This bit must be cleared writing 0
SHUTD	Shutdown procedure is running
KEY_EX	Key input is disabled
IO_SEL0	Decode logic base register address selection
IO_SEL1	Decode logic base register address selection
IRQ5_EN	IRQ5 enabling

The PWM bit can be used to enable or disable the PWM output when the µC next starts.

The *IO_SEL0* and *IO_SEL1* bits can be used to modify the I/O space addresses of the DuraCOR internal register.

IO_SEL0	IO_SEL1	ADDRESS
0	0	0x320 ~ 0x327 (default)
0	1	0x310 ~ 0x317
1	0	0x330 ~ 0x337
1	1	0x340 ~ 0x347

The *IRQ5_EN* bit can be used to enable IRQ5 in the CPU.

VDD_MIN_LOW_REG and VDD_MIN_HIGH_REG

	VDD_MIN_LOW_REG							0X0C
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VDD_MIN_LOW_REG7 to VDD_MIN_LOW_REG0							
Default	0xE0							
Access	R/W							

	VDD_MIN_HIGH_REG							0X0D
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VDD_MIN_HIGH_REG7 to VDD_MIN_HIGH_REG0							
Default	0x02							
Access	R/W							

The *VDD_MIN_LOW_REG* and *VDD_MIN_HIGH_REG* registers contain the most significant byte and least significant bytes of the minimum threshold for the VDD alarm condition.

By reading the *VDD_MIN_LOW_REG* and *VDD_MIN_HIGH_REG* registers, it is possible to monitor the internal minimum threshold and calculate it using the following formula:

VDD = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 2 = ‘Read value’ * 0.0064516

For example, reading 2E0h (736 in decimal) the VDD will be:
736 * (3.3 / 1023) * 2 = 4.75 V

VDD_MAX_LOW_REG and VDD_MAX_HIGH_REG

	VDD_MAX_LOW_REG							0X0E
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VDD_MAX_LOW_REG7 to VDD_MAX_LOW_REG0							
Default	0x2D							
Access	R/W							

	VDD_MAX_HIGH_REG							0X0F
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VDD_MAX_HIGH_REG7 to VDD_MAX_HIGH_REG0							
Default	0x03							
Access	R/W							

The *VDD_MAX_LOW_REG* and *VDD_MAX_HIGH_REG* registers contain the most significant byte and least significant bytes of the maximum threshold allowed on VDD for alarm condition.

By reading the *VDD_MAX_LOW_REG* and *VDD_MAX_HIGH_REG* registers, it is possible to monitor the internal maximum threshold and calculate it using the following formula:

VDD = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 2 = ‘Read value’ * 0.0064516

For example, reading 32Dh (813 in decimal) the VDD will be:
813 * 0.0064516 = 5.25 V

VIN_MIN_LOW_REG and VIN_MIN_HIGH_REG

	VIN_MIN_LOW_REG							0X10
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VIN_MIN_LOW_REG7 to VIN_MIN_LOW_REG0							
Default	0xC8							
Access	R/W							

	VIN_MIN_HIGH_REG							0X11
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VIN_MIN_HIGH_REG7 to VIN_MIN_HIGH_REG0							
Default	0x00							
Access	R/W							

The *VIN_MIN_LOW_REG* and *VIN_MIN_HIGH_REG* registers contain respectively the most significant byte and least significant byte of the minimum threshold for the VIN alarm condition.

By reading the *VIN_MIN_LOW_REG* and *VIN_MIN_HIGH_REG* registers it is possible to monitor the internal minimum VIN voltage threshold, calculating it using the following formula:

VIN = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 13 = ‘Read value’ * 0.041935

For example, reading C8h (200 in decimal) the voltage will be:
200 * 0.041935= 8.39 V

VIN_MAX_LOW_REG and VIN_MAX_HIGH_REG

	VIN_MAX_LOW_REG							0X12
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VIN_MAX_LOW_REG7 to VIN_MAX_LOW_REG0							
Default	0x5A							
Access	R/W							

	VIN_MAX_HIGH_REG							0X13
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VIN_MAX_HIGH_REG7 to VIN_MAX_HIGH_REG0							
Default	0x03							
Access	R/W							

The *VIN_MAX_LOW_REG* and *VIN_MAX_HIGH_REG* registers contain respectively the most significant byte and least significant byte of the maximum threshold for the VIN alarm condition.

By reading the *VIN_MAX_LOW_REG* and *VIN_MAX_HIGH_REG* registers it is possible to monitor the internal maximum VIN voltage threshold, calculating it using the following formula:

VIN = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 13 = ‘Read value’ * 0.041935

For example, reading 35Ah (858 in decimal) the voltage will be:
858 * 0.041935= 36.0V

VCC3_MIN_LOW_REG and VCC3_MIN_HIGH_REG

	VCC3_MIN_LOW_REG							0X14
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_MIN_LOW_REG7 to VCC3_MIN_LOW_REG0							
Default	0xD1							
Access	R/W							

	VCC3_MIN_HIGH_REG							0X15
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_MIN_HIGH_REG7 to VCC3_MIN_HIGH_REG0							
Default	0x01							
Access	R/W							

The *VCC3_MIN_LOW_REG* and *VCC3_MIN_HIGH_REG* registers contain the most significant byte and least significant byte of the minimum threshold for the VCC3 alarm condition.

By reading the *VCC3_MIN_LOW_REG* and *VCC3_MIN_HIGH_REG* registers it is possible to monitor the internal minimum VCC3 voltage threshold, calculating it using the following formula:

VCC3 = 'Read value in decimal' * ((Vref = 3.3V) / 1023) * 2 = 'Read value' * 0.0064516

For example, reading 1D1h (465 in decimal) the voltage will be:
465 * 0.0064516= 3.0V

VCC3_MAX_LOW_REG and VCC3_MAX_HIGH_REG

	VCC3_MAX_LOW_REG							0X16
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_MAX_LOW_REG7 to VCC3_MAX_LOW_REG0							
Default	0x2E							
Access	R/W							

	VCC3_MAX_HIGH_REG							0X17
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_MAX_HIGH_REG7 to VCC3_MAX_HIGH_REG0							
Default	0x02							
Access	R/W							

The *VCC3_MAX_LOW_REG* and *VCC3_MAX_HIGH_REG* registers contain the most significant byte and least significant byte of the maximum threshold for the VCC3 alarm condition.

By reading the *VCC3_MAX_LOW_REG* and *VCC3_MAX_HIGH_REG* registers it is possible to monitor the internal maximum VCC3 voltage threshold, calculating it using the following formula:

VCC3 = 'Read value in decimal' * ((Vref = 3.3V) / 1023) * 2 = 'Read value' * 0.0064516

For example, reading 22Eh (558 in decimal) the voltage will be:
558 * 0.0064516= 3.6V

V12_MIN_LOW_REG and V12_MIN_HIGH_REG

	V12_MIN_LOW_REG							0X18
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	V12_MIN_LOW_REG7 to V12_MIN_LOW_REG0							
Default	0x00							
Access	RW							

	V12_MIN_HIGH_REG							0X19
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	V12_MIN_HIGH_REG7 to V12_MIN_HIGH_REG0							
Default	0x00							
Access	RW							

The *V12_MIN_LOW_REG* and *V12_MIN_HIGH_REG* registers contain the most significant and least significant bytes of the minimum threshold for the V12 alarm condition.

By reading the *V12_MIN_LOW_REG* and *V12_MIN_HIGH_REG* registers it is possible to monitor the internal minimum V12 voltage threshold, calculating it using the following formula:

V12 = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 4.32 = ‘Read value’ * 0.013954

For example, reading 0h (0 in decimal) the voltage will be: 0V

V12_MAX_LOW_REG and V12_MAX_HIGH_REG

	V12_MAX_LOW_REG							0X1A
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	V12_MAX_LOW_REG7 to V12_MAX_LOW_REG0							
Default	0x88							
Access	R/W							

	V12_MAX_HIGH_REG							0X1B
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	V12_MAX_HIGH_REG7 to V12_MAX_HIGH_REG0							
Default	0x03							
Access	R/W							

The *V12_MAX_LOW_REG* and *V12_MAX_HIGH_REG* registers contain the most significant and least significant bytes of the maximum threshold for the V12 alarm condition.

By reading the *V12_MAX_LOW_REG* and *V12_MAX_HIGH_REG* registers it is possible to monitor the internal maximum V12 voltage threshold, calculating it using the following formula:

V12 = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 4.32 = ‘Read value’ * 0.013954

For example, reading 388h (904 in decimal) the voltage will be:
904 * 0.013954= 12.6V

MICRO_TEMP_MIN_LOW_REG and MICRO_TEMP_MIN_HIGH_REG

	MICRO_TEMP_MIN_LOW_REG							0X1C
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	MICRO_TEMP_MIN_LOW_REG7 to MICRO_TEMP _MIN_LOW_REG0							
Default	0x00							
Access	R/W							

	MICRO_TEMP_MIN_HIGH_REG							0X1D
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	MICRO_TEMP_MIN_HIGH_REG7 to MICRO_TEMP_MIN_HIGH_REG0							
Default	0x00							
Access	R/W							

The *MICRO_TEMP_MIN_LOW_REG* and *MICRO_TEMP_MIN_HIGH_REG* registers contain the most and least significant byte of the minimum allowed temperature threshold for the micro-controller alarm condition.

By reading the *MICRO_TEMP_MIN_LOW_REG* and *MICRO_TEMP_MIN_HIGH_REG* registers it is possible to calculate the internal minimum micro-controller temperature threshold, using this formula:

'µC temperature in °C' = ('Read value in decimal' * (3300/1023) - 897) / 3.35

Example: Reading 0h the µC temperature will be: 0 °C

MICRO_TEMP_MAX_LOW_REG and MICRO_TEMP_MAX_HIGH_REG

	MICRO_TEMP_MAX_LOW_REG							0X1E
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	MICRO_TEMP_MAX_LOW_REG7 to MICRO_TEMP_MAX_LOW_REG0							
Default	0xFF							
Access	R/W							

	MICRO_TEMP_MAX_HIGH_REG							0X1F
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	MICRO_TEMP_MAX_HIGH_REG7 to MICRO_TEMP_MAX_HIGH_REG0							
Default	0xFF							
Access	R/W							

The *MICRO_TEMP_MAX_LOW_REG* and *MICRO_TEMP_MAX_HIGH_REG* registers contain the most and least significant byte of the maximum allowed micro-controller temperature threshold for the alarm condition.

By reading the *MICRO_TEMP_MAX_LOW_REG* and *MICRO_TEMP_MAX_HIGH_REG* registers it is possible to calculate the internal maximum micro-controller temperature threshold, using this formula:

'µC temperature in °C' = ('Read value in decimal' * (3300/1023) - 897) / 3.35

Example: Reading FFFFh (65535 in decimal) the µC temperature will be:

(65535 * (3300 / 1023) - 897) / 3.35 = 62837.6 °C

VCC3_ALW_MIN_LOW_REG and VCC3_ALW_MIN_HIGH_REG

	VCC3_ALW_MIN_LOW_REG							0X20
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_ALW_MIN_LOW_REG7 to VCC3_ALW_MIN_LOW_REG0							
Default	0xD1							
Access	R/W							

	VCC3_ALW_MIN_HIGH_REG							0X21
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_ALW_MIN_HIGH_REG7 to VCC3_ALW_MIN_HIGH_REG0							
Default	0x01							
Access	R/W							

The *VCC3_ALW_MIN_LOW_REG* and *VCC3_ALW_MIN_HIGH_REG* registers contain the most and least significant bytes of the minimum "VCC3 Always" threshold for the alarm condition. By reading these values it is possible to monitor the threshold using the following formula:

VCC3 Always = 'Read value in decimal' * ((Vref = 3.3V) / 1023) * 2 = 'Read value' * 0.0064516

For example, reading 1D1h (465 in decimal) the voltage will be: 3V

VCC3_ALW_MAX_LOW_REG and VCC3_ALW_MAX_HIGH_REG

	VCC3_ALW_MAX_LOW_REG							0X22
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_MAX_LOW_REG7 to VCC3_MAX_LOW_REG0							
Default	0x2E							
Access	R/W							

	VCC3_ALW_MAX_HIGH_REG							0X23
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name	VCC3_ALW_MAX_HIGH_REG7 to VCC3_ALW_MAX_HIGH_REG0							
Default	0x02							
Access	R/W							

The *VCC3_ALW_MAX_LOW_REG* and *VCC3_ALW_MAX_HIGH_REG* registers contain the most and least significant bytes of the maximum "VCC3 Always" threshold for the alarm condition. By reading these values it is possible to monitor the threshold using the following formula:

VCC3 Always = 'Read value in decimal' * ((Vref = 3.3V) / 1023) * 2 = 558 * 0.0064516 = 3.6V

OUT_OF_RANGE_STATUS_REG

	OUT_OF_RANGE_STATUS_REG							0X24
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name**	-	TEMP	V12	VCC3	VIN	VCC3_ALW	VDD	-
Default	0x00							
Access	R/W							

** Add "_OUT_OF_RANGE" to the value: i.e. "TEMP" becomes "TEMP_OUT_OF_RANGE"

By reading the *OUT_OF_RANGE_STATUS_REG* register users can determine the current status with regards to the relevant threshold value, possible values are:

- 0: value is within the Upper and Lower threshold values
- 1: Alarm condition, value is above or below the Upper or Lower threshold values

MICRO_TEMP_LOW_REG and MICRO_TEMP_HIGH_REG

	MICRO_TEMP_LOW_REG							0X25
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

	MICRO_TEMP_HIGH_REG							0X26
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

By reading the *MICRO_TEMP_LOW_REG* and *MICRO_TEMP_HIGH_REG* registers it is possible to calculate the internal micro-controller temperature value by using this formula:

'µC temperature in °C' = ('Read value in decimal' * (3300/1023) - 897) / 3.35 = 0°C

FLASH_ADDR_H_REG and FLASH_ADDR_L_REG

	FLASH_ADDR_H_REG							0X27
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

	FLASH_ADDR_L_REG							0X28
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

These registers contain the most and least significant bytes for the reading address of the flash page. This page is used for storing the log.

FLASH_DATA_REG

	FLASH_DATA_REG							0X29
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	R/W							

This register contains the read Byte from the Flash that is stored at the address identified by FLASH_ADDR_H_REG and FLASH_ADDR_L_REG

VDD_LOW_REG and VDD_HIGH_REG

	VDD_LOW_REG							0X2A
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

	VDD_HIGH_REG							0X2B
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

By reading the *VDD_LOW_REG* and *VDD_HIGH_REG* registers it is possible to monitor the VDD Internal voltage, it should then be calculated using the following formula:

VDD = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 2 = ‘Read value’ * 0.0064516

VIN_LOW_REG and VIN_HIGH_REG

	VIN_LOW_REG							0X2C
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

	VIN_HIGH_REG							0X2D
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

By reading the *VIN_LOW_REG* and *VIN_HIGH_REG* registers it is possible to monitor the internal VIN voltage, it should then be calculated using the following formula:

VIN = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 13 = ‘Read value’ * 0.041935

VCC3_LOW_REG and VCC3_HIGH_REG

	VCC3_LOW_REG							0X2E
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

	VCC3_HIGH_REG							0X2F
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

By reading the *VCC3_LOW_REG* and *VCC3_HIGH_REG* registers it is possible to monitor the internal VCC3 voltage, it should then be calculated using the following formula:

VCC3 = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 2 = ‘Read value’ * 0.0064516

V12_LOW_REG and V12_HIGH_REG

	V12_LOW_REG							0X30
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

	V12_HIGH_REG							0X31
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

By reading the *V12_LOW_REG* and *V12_HIGH_REG* registers it is possible to monitor the internal +12V voltage. This option is only available if the system has a DC/DC converter (+5V to +12V) present.

It should then be calculated using the following formula:

V12 = ‘Read value in decimal’ * ((Vref = 3.3V) / 1023) * 4.32 = ‘Read value’ * 0.013954

VCC3_ALWAYS_LOW_REG and VCC3_ALWAYS_HIGH_REG

	VCC3_ALWAYS _LOW_REG							0X32
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

	VCC3_ALWAYS _HIGH_REG							0X33
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	-							
Access	R							

By reading the *VCC3_ALWAYS_LOW_REG* and *VCC3_ALWAYS_HIGH_REG* registers it is possible to monitor the internal *VCC3_ALWAYS* voltage, it should then be calculated using the following formula:

VCC3_ALWAYS = 'Read value in decimal' * ((Vref = 3.3V) / 1023) * 2 = 'Read value' * 0.0064516

EEPROM_ADDR_REG

	EEPROM_ADDR_REG							0X34
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	R/W							

This register contains the read/write address of the EEPROM used to store logs.

EEPROM_DATA_REG

	EEPROM_DATA_REG							0X35
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	R							

This register can contain either:

- The byte read from the EEPROM at the address identified by EEPROM_ADDR_REG following a read command.
- The data to write to the address identified by EEPROM_ADDR_REG before the write command

EEPROM_STATUS_REG

	EEPROM_STATUS_REG							0X36
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

This register contains the status of the last read/write operation done in the EEPROM used to store the logs.

VIN_SHUTDOWN_REG

	VIN_SHUTDOWN_REG							0X37
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

The system will shutdown if VIN <= VIN_MIN for “VIN_SHUTDOWN_REG” consecutive times. If “VIN_SHUTDOWN_REG” = 0 this feature is disabled (default).

VIN_PERIOD_REG

	VIN_PERIOD_REG							0X38
Bit	B7	B6	B5	B4	B3	B2	B1	B0
Bit name								
Default	0x00							
Access	RW							

VIN is checked every “VIN_PERIOD_REG” msec.

(This page has been intentionally left blank)

Chapter 3 Event Storage

Once a second the Guardian checks that the monitored values fall within the specified ranges.
If any of these conditions are not met the event is stored in the EEPROM (5 bytes occupied).
If enabled IRQ5 is activated on the ISA bus in order to inform the CPU about Alarm condition.
If the condition persists, the CPU will be is informed every sixty seconds.

EEPROM structure

The first byte of the EEPROM is a pointer to the first free location.
The first event starts at the fifth location, following this kind of structure:

ADDRESS	CONTENT
0x00	Pointer to the first free location
0x01	Reserved
0x02	Reserved
0x03	Reserved
0x04	Reserved
0x05	Log 1
…	
0x0A	Log 2
…	
0x0F	Log 3
…	

Error log Structure

Each error is stored in 5 bytes of memory, following this rule:

BYTE #	CONTENT
1	The error type based on its code (see the "Error codes" paragraph below)
2	The most significant byte of the monitored features value
3	The least significant byte of the monitored features value
4	The hour when the event occurred
5	The minute when the event occurred

Time is initialised when the µC is turned on.
When the µC is turned on the monitored values will be stored in the flash even if they are in the error status or not. In this case the byte 0 contains a value that is equal to the error code plus 128.

Error codes

NAME	CODE	DESCRIPTION
VDD_ERROR	0x01	VDD is out of the permitted range
VCC3_ALWAYS_ERROR	0x02	VCC3 is out of the permitted range
VIN_ERROR	0x03	VIN is out of the permitted range
VCC3_ERROR	0x04	VCC3 is out of the permitted range
V12_ERROR	0x05	V12 is out of the permitted range
MICRO_TEMP_ERROR	0x06	µC temperature is out of the permitted range
CPI_TEMP_ERROR	0x07	Reserved
MICRO_WTD_ERROR	0x08	µC has been restarted by the watchdog
CPU_WTD_ERROR	0x09	CPU has been restarted by its watchdog
POST_ERROR	0x0A	Reserved
EEPROM_ERROR	0x0B	EEPROM Error

Bytes 2 and 3 of the error log are null in the last five error codes.

Power supply CPU logic

The Guardian firmware will not enable the POWERON line until the nKEY (hardware key) input is activated.

Disabling the nKEY input will immediately disable the POWERON line unless the Guardian receives an EXCLUDE_KEY command; in this second case the nKEY command will not affect the POWERON, and the only way to disable the CPU power line is to send the command SHUTDOWN to the Guardian.

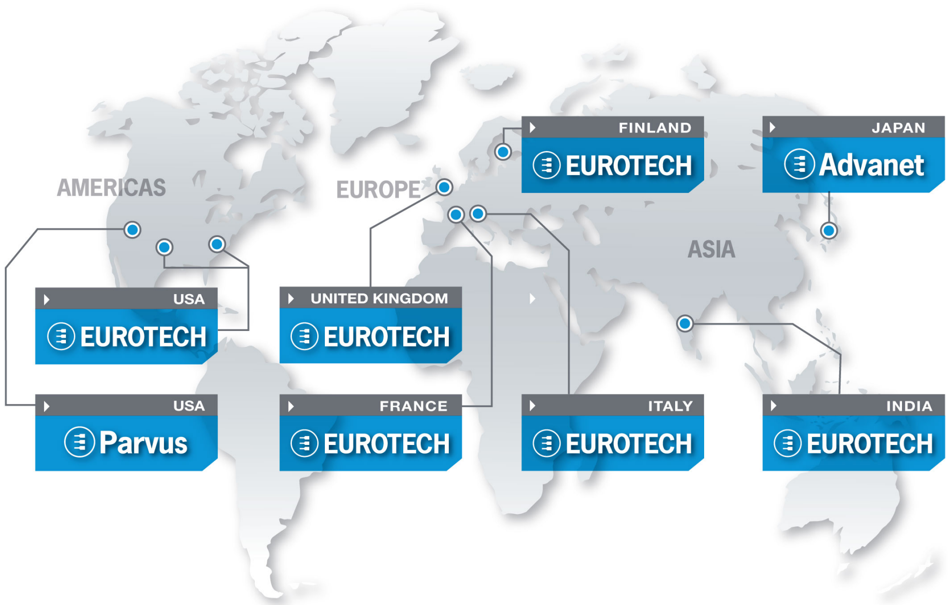
Reset power button

Pressing the reset push-button on the front panel of the DuraCOR system will reset the DC/DC output.

Firmware update

The Guardian supports a firmware updating procedure; for further details feel free to contact the Eurotech Technical Support .

Eurotech Worldwide Presence



AMERICAS

USA

EUROTECH

EUROTECH
Toll free +1 888.941.2224
Tel. +1 301.490.4007
Fax +1 301.490.4582
E-mail: sales.us@eurotech.com
E-mail: support.us@eurotech.com
Web: www.eurotech-inc.com

PARVUS
Tel. +1 800.483.3152
Fax +1 801.483.1523
E-mail: sales@parvus.com
E-mail: tsupport@parvus.com
Web: www.parvus.com

EUROPE

Italy

EUROTECH
Tel. +39 0433.485.411
Fax +39 0433.485.499
E-mail: sales.it@eurotech.com
E-mail: support.it@eurotech.com
Web: www.eurotech.com

United Kingdom

EUROTECH
Tel. +44 (0) 1223.403410
Fax +44 (0) 1223.410457
E-mail: sales.uk@eurotech.com
E-mail: support.uk@eurotech.com
Web: www.eurotech.com

France

EUROTECH
Tel. +33 04.72.89.00.90
Fax +33 04.78.70.08.24
E-mail: sales.fr@eurotech.com
E-mail: support.fr@eurotech.com
Web: www.eurotech.com

Finland

EUROTECH
Tel. +358 9.477.888.0
Fax +358 9.477.888.99
E-mail: sales.fi@eurotech.com
E-mail: support.fi@eurotech.com
Web: www.eurotech.com

ASIA

Japan

ADVANET
Tel. +81 86.245.2861
Fax +81 86.245.2860
E-mail: sales@advanet.co.jp
E-mail: tsupport@advanet.co.jp
Web: www.advanet.co.jp

India

EUROTECH
Tel. +91 80.43.35.71.17
E-mail: sales.in@eurotech.com
E-mail: support.in@eurotech.com
Web: www.eurotech.com

To find your nearest contact refer to: www.eurotech.com/contacts

EUROTECH

www.eurotech.com

EUROTECH HEADQUARTERS
Via Fratelli Solari 3/a
33020 Amaro (Udine) – ITALY
Phone: +39 0433.485.411
Fax: +39 0433.485.499

For full contact details go to: www.eurotech.com/contacts