



Embedded BIOS 2000 Quick Reference

CC5-RAVE • *CompactPCI*[®] FC-PGA 370 CPU

Document No. 2632 • Preliminary Edition 1

06-2002



Contents

About this Manual	4
Edition History	4
Related Documents	5
Nomenclature	5
Trade Marks	5
Legal Disclaimer - Liability Exclusion	5
Overview	6
Power-On Self-Test (POST)	7
The BIOS User Interface	7
Setup Screen System	9
Basic CMOS Configuration Screen	11
Date/Time	11
NumLock	11
Configuring Drive Assignment Order	12
Boot Order	12
Configuring Floppy Drive Types	13
Configuring IDE Drive Geometry	13
Miscellaneous Settings	14
Typematic Delay/Rate	14
Seek at Boot	14
Show "Hit Del"	14
Config Box	14
F1 Error Wait	14
Parity Checking	14
Memory Test Tick	14
INT 3 Instruction	14
Debugger Hex Case	14
Memory Test	15
Features Configuration Setup Screen	16
ACPI & Advanced Power Management	16
Graphical/Audio POST	17
System Management Mode	17
POST Memory Manager	18
System Management BIOS	18
Custom Configuration Setup Screen	19
Processor Throttle Mode	19
Scan PCI Buses	19
L2 Cache	20
Show BIOS Messages, Delay After Messages	20

Behaviour after Initializing Expansion ROM	20
Select Primary VGA	20
Select Ethernet BIOS	21
Digital DVI Support	21
BIOS Standard I/O Port	21
Parallel Port Configuration	22
Serial Port Configuration	22
IDE Configuration Setup Screen	23
Primary IDE Port	23
UDMA66	23
Master/Slave Timing Mode	24
IORDY Sampling Point	24
Recovery Time	24
Secondary IDE Port	24
Password Configuration	25
Burn-In Diagnostics Routines Setup Screen	26
System BIOS Debugger	27
Console Redirection	28
Addendum A: Embedded BIOS POST Codes	29
Addendum B: Embedded BIOS Beep & Blink Codes	32

About this Manual

This manual describes some of the setup functions incorporated in the on-board BIOS, provided with the CC5-RAVE. It is intended for the experienced user only. As the BIOS will be undergoing future development, information provided herein may differ from the actual appearance.

This manual is valid from BIOS build 1.00 off. For BIOS builds 0.94 and older please refer to the "BIOS Quick Reference CC5-RAVE", EKF document no. 2508. The particular BIOS version in use can be obtained from the starting screen when switching system power on (hit 'break key').

Edition History

EKF Document	Ed.	Contents/Changes	Author	Date
Text # 2632	1	1. Edition Embedded BIOS 2000 Quick Reference CC5-RAVE, English Preliminary edition, to be completed later on	jj	3 June 2002

Related Documents

For a description of the CC5-RAVE printed circuit board see the document "CC5-RAVE User Guide", available by download at <http://www.ekf.de/c/ccpu/cc5/cc5.html>.

Nomenclature

Numbers followed by a 'h' or with a '0x' prefix represent hexadecimal values.

Trade Marks

Some terms used herein are property of their respective owners, e.g.

Pentium, Celeron, Socket 370: ® Intel, *CompactPCI*: ® PICMG, Windows 98, Windows NT, Windows 2000: ® Microsoft

General Software™, the GS Logo, Embedded BIOS™, Embedded BIOS™ 2000, BIOSstart™, and Embedded DOS™ are trademarks or registered trademarks of General Software, Inc.

EKF does not claim this list to be complete.

Legal Disclaimer - Liability Exclusion

This manual has been edited as carefully as possible. We apologize for any potential mistake. Information provided herein is designated exclusively to the proficient user (system integrator, engineer). EKF can accept no responsibility for any damage caused by the use of this manual.

Overview

The CC5-RAVE board is licensed with a single copy of Embedded BIOS 2000 software from General Software, Inc. General Software may be reached on the web at <http://www.gensw.com>. The BIOS has been adapted to the particular CC5-RAVE hardware by EKF. BIOS updates will be available by download at <http://www.ekf.de/c/ccpu/cc5/cc5.html>. E-Mail support can be obtained directly by support@ekf.de. Please also report any issue to EKF.

The system's pre-boot environment of Embedded BIOS includes POST, Setup Screen System, Manufacturing Mode, Console Redirection, and Integrated BIOS Debugger. A REFLASH tool is also available to update the BIOS image with new builds of Embedded BIOS. Before using the system, please read the following to properly configure CMOS settings, and learn how to use the embedded features of the pre-boot firmware, Embedded BIOS. The last two sections of this manual provide the BIOS POST Codes and Beep codes.

This manual does not cover reprogramming of the Flash BIOS. Please download the latest BIOS version and the programming tool from <http://www.ekf.de/c/ccpu/cc5/firmware/>, un-zip the package and follow thoroughly the instructions provided within the readme.txt.

General Software's EMBEDDED BIOS brand BIOS (Basic Input/Output System) pre-boot firmware is the industry's standard product used by most designers of embedded X86 computer equipment in the world today. Its superior combination of configurability and functionality enables it to satisfy the most demanding ROM BIOS needs for embedded designers. Its modular architecture and high degree of configurability make it the most flexible BIOS in the world.

EMBEDDED BIOS offers a high degree of compatibility with past and current BIOS standards, allowing it to run off-the-shelf operating system software and application software. EMBEDDED BIOS has been tested with all industry standard operating systems, including versions of Windows, Linux, DOS, and real time operating systems. EMBEDDED BIOS is rigorously tested with programs such as AMI Diag, MSD, Check-It, Manifest, Q/A Plus, and so on, ensuring its compatibility with established desktop application standards. In addition to its standard data structures and programming interfaces, EMBEDDED BIOS provides support for industry-standard initiatives, including ACPI, APM, EI Torito, Legacy USB, MP, PCI, PMM, PXE, and SMBIOS (formerly DMI).

Power-On Self-Test (POST)

When the system is powered on, Embedded BIOS tests and initializes the hardware and programs, the chipset and other peripheral components. During this time, POST progress codes are written by the system BIOS to I/O port 80H, allowing the user to monitor the progress with a suitable emulator equipment. "Embedded BIOS POST Codes" in addendum A lists the POST codes and their meanings. During early POST, no video is available to display error messages should a critical error be encountered; therefore, POST uses beeps on the speaker (if the CC6-ACID super I/O board is attached) and the floppy drive LED (if there is one) to indicate the failure of a critical system component during this time. Consult "Embedded BIOS Beep Codes" on the addendum B pages for a list of Beep codes used by the system's BIOS.

The BIOS User Interface

The system BIOS can use the standard keyboard and video device, or use console redirection to demonstrate headless operation. For headless operation, remove the standard keyboard and screen devices and the system will boot unattended. If an RS232 cable is attached to COM1, a PC/ATstyle character-based POST is available from HyperTerminal, PROCOMM, or any other terminal emulator software that supports VT100 emulation.

When a keyboard and video device are attached, the CC5-RAVE can display either a traditional character-based PC BIOS display with memory count-up, or it can display a graphical POST with splash screen and progress icons. Both POST displays accept a key press to enter the setup screen, and both display boot-time progress activity displays. The graphical display shows the status of file system devices, but omits character-based PCI resource display. The text-based POST displays the memory count-up and the PCI resource assignment table.

The figure below shows the format of the text-based POST display. The display is very similar if console redirection through a COM port is used instead. PCI Bus numbers 00 and 01 belong to devices on the CC5-RAVE, while numbers from 02 off are related to components on the CPCI backplane and CPCI I/O modules. The screen image shown differs slightly after a system warm start, e.g. caused by pressing Ctrl-Alt-Del.

```
Embedded BIOS 2000 (tm), Copyright (C) 2001 General Software, Inc.
EKF CC5-RAVE (BIOS build: 1.10 [production] - 2002-04-30)
CPU: Pentium III 798 Mhz      Chipset: Intel 810 GMCH (R3) + 801AA ICH (R2)
```

```
00000640K Low Memory Passed      SO-DIMM: PC100U-222-620
00260096K Ext Memory Passed      MemCtrl: 100MHz 223 Trc=8
Wait.....
```

PCI Device Table:

Bus	Dev	Func	VendID	DevID	Rev	Class	IF	Description	IL	IRQ
00	00	00	8086	7124	03	06	00	Host Bridge		
00	01	00	8086	7125	03	03	00	VGA Display	A	10
00	1E	00	8086	2418	02	06	04	PCI-to-PCI Bridge		
00	1F	00	8086	2410	02	06	01	ISA Bridge		
00	1F	01	8086	2411	02	01	01	IDE Controller		
00	1F	02	8086	2412	02	0C	03	USB (UHCI)	D	10
00	1F	03	8086	2413	02	0C	05	SMBus Controller	B	10
01	04	00	8086	1209	09	02	00	Ethernet	A	10
01	06	00	104C	Ac28	00	06	04	PCI-to-PCI Bridge		
02	0A	00	1000	000C	01	01	00	SCSI Controller	A	10

```
Searching for El Torito Bootable Image ... No media.
Initializing IDE harddisk 00 . Success
```

If the graphical version of POST has been activated, the BIOS decompresses the main image, and can display multiple overlaid graphics at various points in POST. EKF could define the entire sequence and control the timing of the system for an embedded application, and can arrange to have different graphics displayed on each successive boot of the system. This feature is ideal for embedded systems that must show evidence of operation during startup, while the application loads underneath the splash screen. Once the application begins writing to the screen, the splash screen relinquishes control, providing a seamless graphical progression for the end user. Please contact EKF in order to arrange a custom specific splash screen (mail to support@ekf.de). When the system is powered on for the first time, you'll need to configure the system through the Setup Screen System (described later) before peripherals, such as disk drives, are recognized by the BIOS. The information is written to battery-backed CMOS RAM on the board's Real Time Clock. Should the board's battery fail, this information will be lost and the board will need to be reconfigured. The Basic Setup Screen provides an option to disable the graphical POST and switch to the legacy text-based version. This feature may not permanently disable the graphical POST if the BIOS adaptation calls for reverting to the graphical form after so many boots. If you find that the graphical POST comes back after several boots, it is because this option is enabled for this platform. EKF can arrange to control whether Setup can be used to dictate the policy, and whether it is permanent or temporary.

Setup Screen System

Without user interaction, the Embedded BIOS 2000 will display a short status message, and then attempts to boot from an available data source (see figure below). The suitable behaviour (boot device) can be selected from the BIOS Setup Screen System, among many other parameters.

```

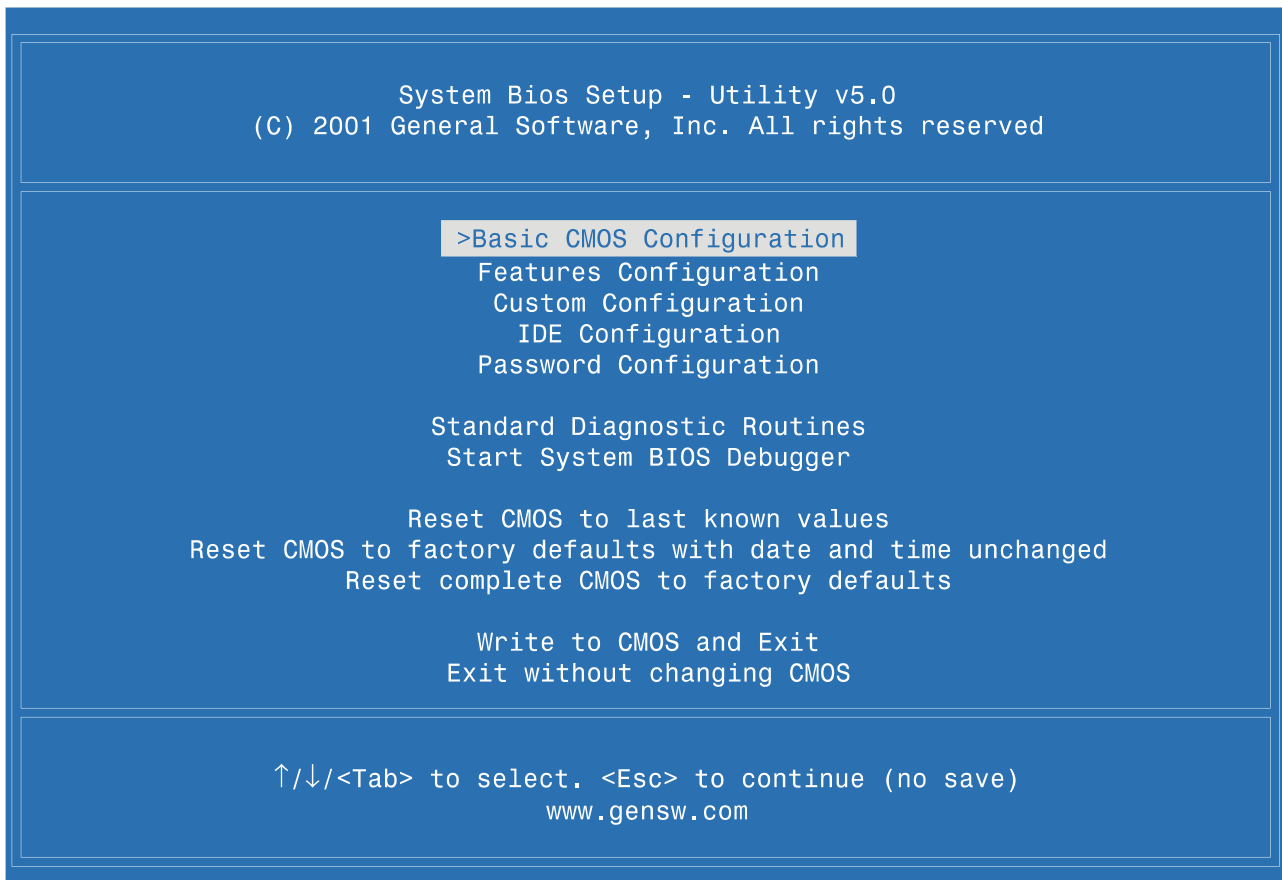
System BIOS Configuration, (C) 2001 General Software, Inc.

System CPU      : Pentium III      Low Memory      : 630KB
Coprocessor    : Enabled          Extended Memory : 254MB
Floppy 0 Type  : 1.44 MB, 3.5"    Serial Ports 1-2 : 03F8 02F8
Floppy 1 Type  : not installed     Serial Ports 3-4 :
IDE 0 Type     : 3                Parallel Ports  : 0378
IDE 1 Type     : 5                ROM Shadowing   : Enabled
Embedded BIOS Date : 2002-04-30   Manufacturing Mode : Disabled

ROM segment 0xc800 length 0x4000 reloc 0x9400
Etherboot 5.0.5 (GPL) Tagged ELF for [EEPROM100]
Boot from (N)etwork or from (L)ocal? N
Found Intel EtherExpressPro100 82559ER at 0xec00, ROM address 0xc000
Probing...[EEPROM100]Ethernet addr: 00:C0:88:F6:01:A0
Searching for server (DHCP)...

```

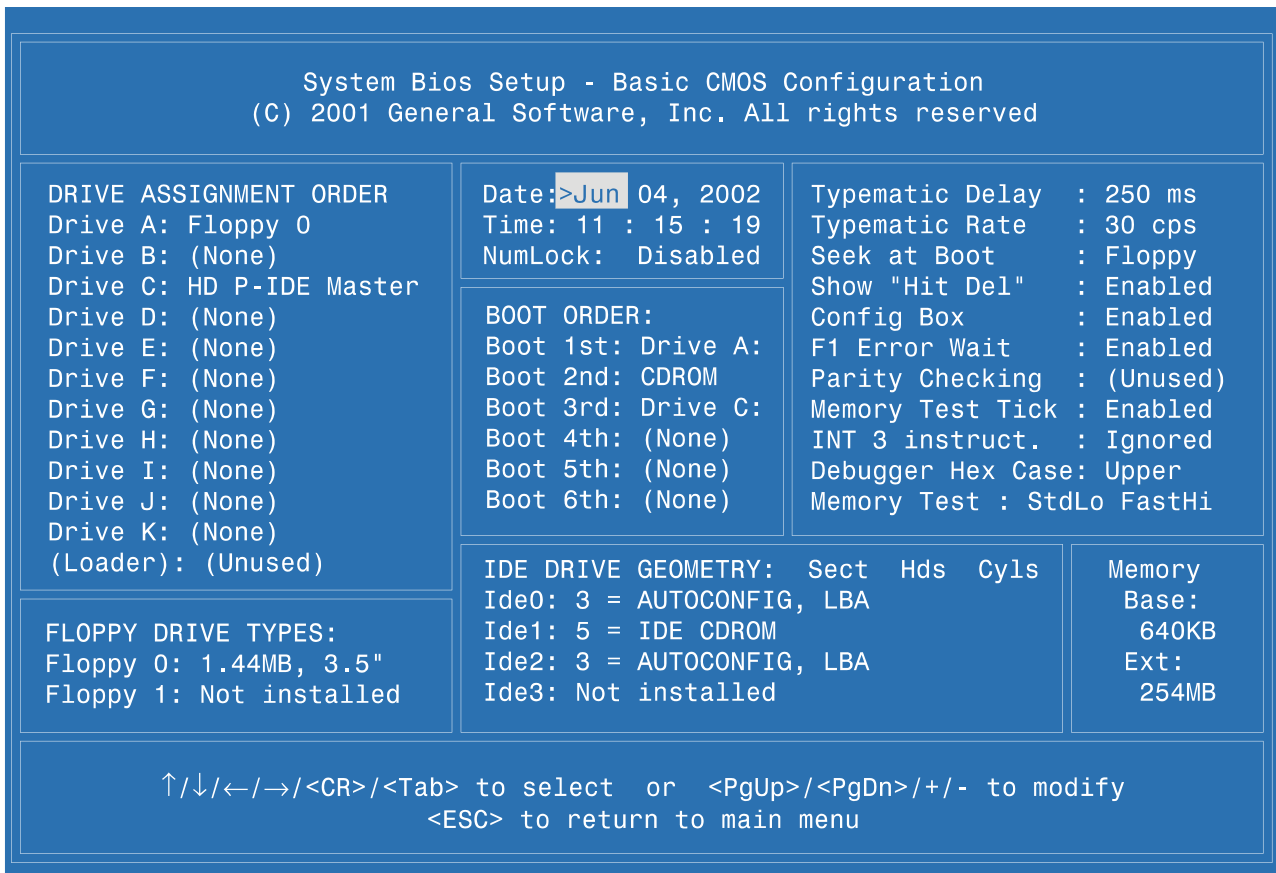
The computer system (here CC5-RAVE) is configured from within the Setup Screen System, which is a series of menus that can be invoked from POST by pressing the key. Once in the Setup Screen System, the user can navigate with the UP and DOWN arrow keys. Pressing ENTER opens the selected sub-menu screen (highlighted option, marked with an arrow).



Within the sub-menu screens, TAB and ENTER are used in addition to the cursor keys to advance to the next field, and '+' and '-' keys (or PgUp/PgDn) cycle through values, such as those in the Basic Configuration Screen, or the IDE Configuration Screen. Press ESC in order to return to the main menu screen. Modified BIOS adjustments can be either permanently saved ('Write to CMOS and Exit'), or abandoned ('Exit without changing CMOS').

Basic CMOS Configuration Screen

The system's drive types, boot activities, and POST optimizations are configured from the Basic Setup Screen. In order to use disk drives with your system, you must select appropriate assignments of drive types in the left-hand column. Then, if you are using true floppy and IDE drives (not memory disks that emulate these drives), you need to configure the drive types themselves in the Floppy Drive Types and IDE Drive Geometry sections. Finally, you'll need to configure the boot sequence in the middle of the screen. Once these selections have been made, your system is ready to use.



Date/Time

This section lets you set the real-time clock of the system. If the clock stands still, the buffer accumulator of the CMOS is probably depleted (or needs to be simply recharged). In this case exchange the accumulator (or set the board under power in order to recharge the accumulator) and set the RTC again. When detecting a failure in the CMOS CRC, caused by an exhausted accumulator, all CMOS options are automatically loaded from the BIOS Flash (factory default parameters) and have to be reconfigured (if necessary) by the user.

NumLock

Parameter *NumLock* defines the state of the <Num> key during boot.

Configuring Drive Assignment Order

Embedded BIOS 2000 allows the user to map a different file system to each drive letter. The BIOS allows file systems for each floppy (Floppy 0 and Floppy 1), and each IDE drive (Primary IDE Master, Primary IDE Slave), and the on-board CompactFlash socket (Secondary IDE Master). The figure above shows how the first floppy drive (Floppy 0) is assigned to drive A: in the system, and then how the first IDE drive (Primary IDE Master) is assigned to drive C: in the system. To switch two floppy disks around or two hard disks around, just map Floppy 0 to B: and Floppy 1 to A:, and for hard disks map the Primary IDE Master to D: and the Primary IDE Slave to C:. If you want the system to boot from the CompactFlash ATA card, it must be previously assigned one of the drive letters C: or D:, because the boot order options are currently restricted to drive letters A: - D: (see section "Boot Order").

Configuring is necessary for floppy disk drives, hard disk drives, and ATA CompactFlash cards. There is no need to configure CDROM drives. If the CDROM device is attached e.g. to the Primary IDE Slave port, select the option "Drive D: <none>", in our given example.

Caution: Take care to not skip drive A: when making floppy disk assignments, as well as drive C: when making hard disk assignments. The first floppy should be A:, and the first hard drive should be C:. Also, do not assign the same file system to more than one drive letter. Thus, Floppy 0 should not be used for both A: and B:. The BIOS permits this to allow embedded devices to alias drives, but desktop operating systems may not be able to maintain cache coherency with such a mapping in place.

Boot Order

This section defines the order in which the logical drives are scanned during the boot procedure. To avoid unnecessary delays of the boot procedure, you should only enter logical drives that are assigned to physical drives. Embedded BIOS supports up to five different user-defined steps in the boot sequence. When the entire system has been initialized, POST executes these steps in order until an operating system successfully loads. In addition, other pre-boot features can be run before, after, or between operating system load attempts. The following actions can be used:

Drive A: - D: or CDROM: Boot the operating system from specified drive. If "Loader" (see 'Drive Assignment Order') is set to "BootRecord" or "Unused", then the standard boot record will be invoked, causing DOS, Windows95/98, Windows NT, or other industry-standard operating systems to load (currently the loader option is preset to 'Unused' and cannot be changed by the user). If you want the system to boot from the CompactFlash ATA card, it must be previously assigned to one of the drive letters C: or D: (see section "Drive Assignment Order").

Debugger Launch the integrated BIOS Debugger. To return to the boot process from the debugger environment, type "G" at the debugger prompt and press ENTER.

None No action; POST proceeds to the next activity in the sequence.

Configuring Floppy Drive Types

If true floppy drive file systems (and not their emulators, such as ROM, RAM, or flash disks) are mapped to drive letters, then the floppy drives themselves must be configured in this section. Floppy 0 refers to the first floppy disk drive on the drive ribbon cable (normally drive A:), and Floppy 1 refers to the second drive (drive B:). If a floppy drive is not present, you must select "None", otherwise there will be long delays during the POST and system boot, caused by timeouts in the corresponding disk I/O routines.

Configuring IDE Drive Geometry

If true IDE disk file systems are mapped to drive letters, then the IDE drives themselves must be configured in this section. The following table shows the drive assignments for Ide 0 - Ide 3:

File System Name (Typical Usage)	Controller	Master/Slave
Ide 0 (external HD drive)	Primary (1f0h)	Master
Ide 1 (external device, HDD, DVD etc.)	Primary (1f0h)	Slave
Ide 2 (on-board CompactFlash)	Secondary (170h)	Master

To use the primary master IDE drive in your system (the typical case), just configure Ide 0 in this section, and map Ide 0 to drive C: in the Configuring Drive Assignments section. The IDE Drive Types section lets you select the type for each of the four IDE drives: Not Installed, User, Physical, LBA, or Phoenix/CHS.

User - This type allows the user to select the maximum cylinders, heads, and sectors per track associated with the IDE drive. This method is now rarely used since LBA is now in common use.

Physical - This type instructs the BIOS to query the drive's geometry from the controller on each POST. No translation on the drive's geometry is performed, so this type is limited to drives of 512 Mbytes or less. Commonly, this is used with embedded ATA PC Cards.

LBA - This type instructs the BIOS to query the drive's geometry from the controller on each POST, but then translate the geometry according to the industry-standard LBA convention. This supports up to 16-Gbyte drives. Use this method for all new drives.

Phoenix (CHS) - This type instructs the BIOS to query the drive's geometry from the controller on each POST, but then translate the geometry according to the Phoenix CHS convention. Using this type on a drive previously formatted with LBA or Physical geometry might show data as being missing or corrupted.

Geometry can either be entered directly in format cylinder/head/sector or can be automatically recognized, if this is supported. Automatical recognition supports physical addressing, LBA addressing and addressing according to Phoenix standard. For IDE drives smaller than 504MB and CD-ROM drives you should select option "AUTOCONFIG, PHYSICAL", for bigger drives select "AUTOCONFIG, LBA". Option "AUTOCONFIG, PHOENIX" is not in common use any more. Select "Not installed" for drives that are not installed.

Miscellaneous Settings

Typematic Delay/Rate

These parameters configure the keyboard interface and define the repetition rate of the characters. Normally you do not need to change these parameters.

Seek at Boot

This option defines drives to which drive a "SEEK" command is sent before the boot procedure. You can select 'NONE' (no drive), 'FLOPPY' (floppy drive A), 'IDE' (IDE drive C) and 'BOTH' (floppy and IDE drive). The default setting is 'NONE' to make the boot procedure as short as possible.

Show "Hit Del"

If this option is active, you have the choice to enter the Setup menu during the power-on self test. To do this, press the key (with monitor/keyboard) or <Ctrl> + <C> (with terminal).

Config Box

This option determines whether or not the configuration table is displayed on the boot console.

F1 Error Wait

If this option is active, the power-on self test will halt when detecting an error, and will prompt you to determine how to proceed. If you press <F1> the POST will continue despite the error, if you press you will enter the Setup menu and can check the configuration.

Parity Checking

This menu item is not used. It has no function.

Memory Test Tick

If this option is active, a click signal will be output on the speaker during the power on self test.

INT 3 Instruction

If your version of DOS, an application, or any OEM-supplied BIOS extensions have debugging code (i.e., "INT 3" instructions) remaining, then these would invoke the debugger automatically, if 'INT 3 instruct.' is set to 'Call Dbg'. To continue, use the "G" command. Normally this option is set to 'Ignored'.

Debugger Hex Case

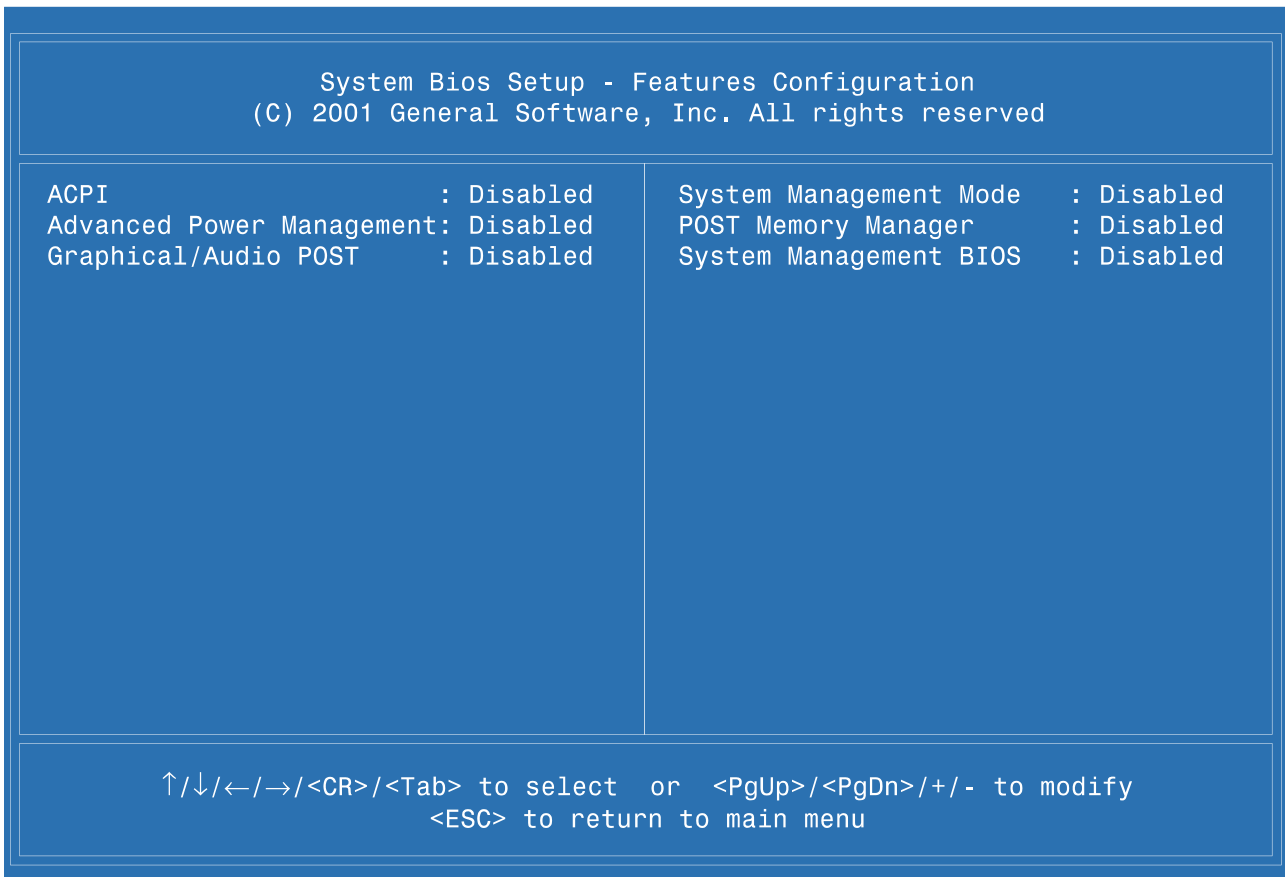
Determines whether the debugger function outputs hexadecimal numbers as lower case or upper case letters.

Memory Test

On POST, when testing the random access memory, there are several testing intensity options. Deeper testing requires more time, and vice versa.

Low Memory (0-640KB)	Extended Memory 1-256MB
FastLo	FastHi
StdLo	FastHi
FullLo	FastHi
FastLo	StdHi
StdLo	StdHi
FullLo	StdHi
FastLo	FullHi
StdLo	FullHi
FullLo	FullHi

Features Configuration Setup Screen



ACPI & Advanced Power Management

EMBEDDED BIOS 2000 provides support for power-sensitive applications by being compatible with the Advanced Power Management and Control (ACPI) Specification 2.0, and the older legacy PC Microsoft Advanced Power Management (APM) Specification 1.2. Please note that power management is restricted to the hardware capacities of the CC5-RAVE.

ACPI (Modern OS-Based Power Management)

ACPI gives control to the OS for a variety of functions that have been traditionally handled by BIOS or 3rd party software. The most commonly talked about of these uses is power management. With ACPI, the OS can control power flow, making sure that devices do not overheat, or that the battery maintains its life for as long as possible. APM also provides these functions, but in APM the BIOS really controls most aspects, while in ACPI it is the OS that does this, thus requiring one less step to go through to make sure everything works. More importantly, APM controls only power management, while ACPI truly gives all sorts of control to the OS. Operating systems that support ACPI include Windows 98 SE, Windows 2000, and Windows XP. The OS learns what it needs to know for ACPI by looking at information passed to it from the BIOS. The BIOS stores this information at the very top of physical memory, and in a format defined by the Advanced Configuration and Power Interface Specification. EMBEDDED

BIOS 2000 complies with both version 1.0 of this spec, and version 2.0, which is the most recent as of the release of this manual.

APM (Older Style Legacy PC Power Management)

If APM is enabled in the BIOS, the operating system and application may access APM BIOS services that can be used to control the power-consumption state of the system.

Graphical/Audio POST

Another facility, the Splash Screen, provides a graphical (and auditory) user interface that provides the end user with the impression that the system is booting immediately, and then transitions easily to the graphical environment of the run-time operating system. The Splash Screen supports animation, OEM branding, and even updates of sections of the Splash Screen graphics in the field through a BRAND.EXE utility.

System Management Mode

The Pentium III CPU supports three processor modes: real mode, protected mode, and system management mode (SMM). This third mode runs transparently to the other modes' execution, and can be entered only by a special interrupt which has no interrupt level assignment through the PICs. Instead, this interrupt is generated (typically) by the southbridge and is wired directly to an SMI# pin on the CPU. This means that the CPU enters and leaves SMM without allowing the operating system or application software to become aware of it. EMBEDDED BIOS provides a way to harness SMM to perform firmware-level work without the complexities of SMM mode programming, chipset programming, and assembly language programming. The SMM manager in the core BIOS, if enabled, calls the underlying Board Personality Module and Chipset Personality Module to program the SMM hardware, initializing it, and loading the protected SMRAM with an SMM operating environment, Firmbase. This run-time kernel provides a flat, 32-bit, protected mode environment within the protective envelope of SMM execution, and enables the OEM to create 32-bit firmware applications that run in this environment, completely transparently to the foreground operating system and its applications. Real applications for Firmbase include Legacy USB stacks, OS system death detection, remote access and control, and many other functions. The theory of operation of Firmbase, as well as the practical aspects of programming for that environment, are beyond the scope of this manual. A separate Firmbase SDK provides these details. For more information about Firmbase, contact General Software.

POST Memory Manager

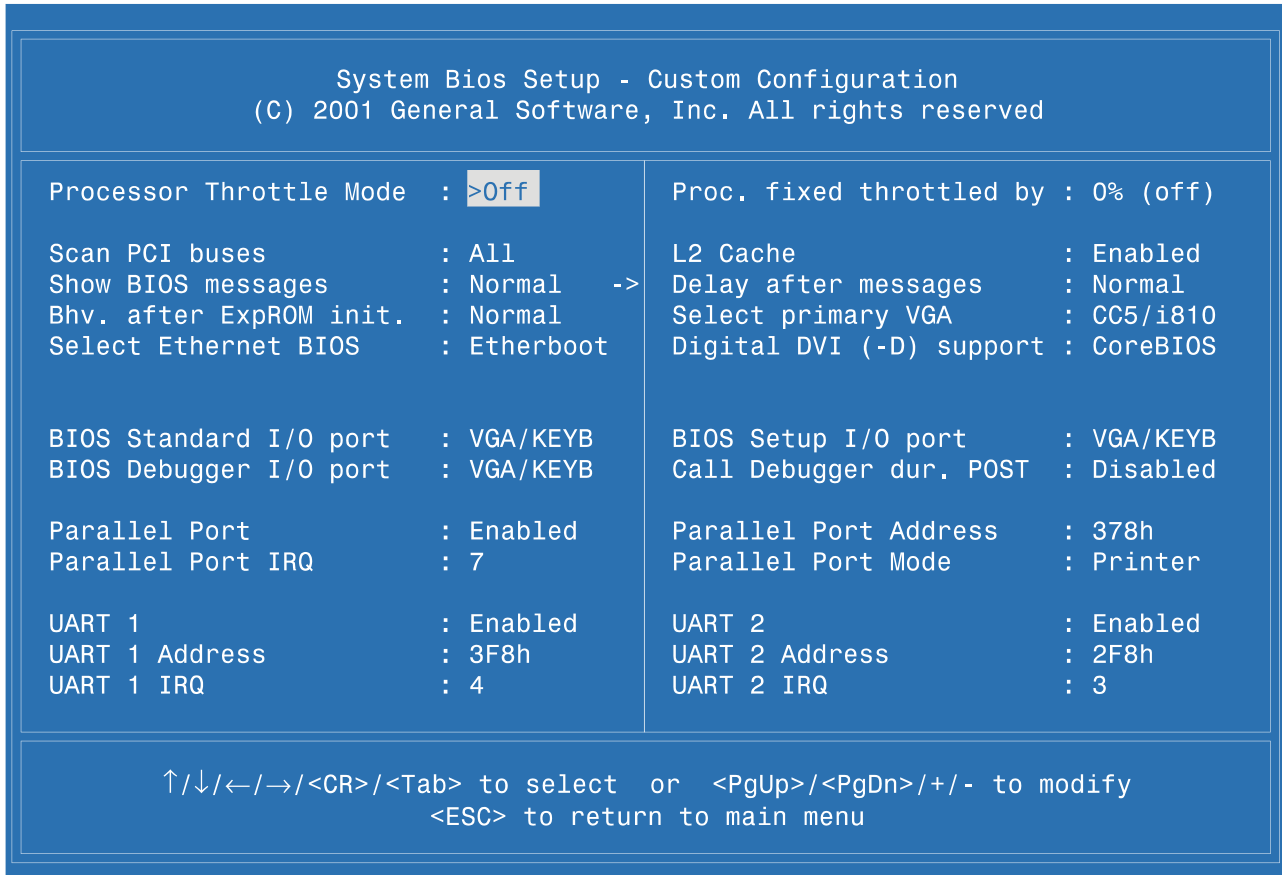
The Address Manager supports other memory managers in the system BIOS, including the POST Memory Manager (PMM), a subsystem that is called by PXE and other POST-time option ROMs to request memory from low or extended memory.

System Management BIOS

EMBEDDED BIOS provides the SMBIOS support necessary to support the Intel Preboot Execution Environment (PXE) firmware used in remote management and booting applications. System Management BIOS, or SMBIOS (also formerly known as Desktop Management Interface, or DMI), is a way for a system to return low level information about itself to the OS or to application level programs. Many attributes of a system are known only to the BIOS, yet may be useful to OS or application software. SMBIOS provides a standard interface for returning this information. This implementation conforms to the SMBIOS v2.3.1 spec.

Custom Configuration Setup Screen

The system's hardware-specific features are configured with the Custom Setup Screen. All features are straightforward except for the Redirect Debugger I/O option, which is an extra embedded feature that allows the user to select whether the Integrated BIOS Debugger should use standard keyboard and video or RS232 console redirection for interaction with the user. If no video is available, the debugger is always redirected.



Processor Throttle Mode

This option allows to reduce considerably the CPUs power consumption, useful for low power applications, or when high ambient temperatures can occur. If the full processor performance is not needed, the CPU can be throttled back either on a fixed or temperature controlled (future option) base, to 12.5/25/37.5/50/62.5/75/87.5%. Switch the throttle mode off if the full CPU performance is required.

Scan PCI Buses

On POST, the BIOS scans the PCI bus for attached devices. If this option is set to 'ALL', any PCI device is detected, also those components which are behind several PCI to PCI bridges. Setting the scan to '0 only' limits the enumerating procedure to the local CC5-RAVE chipset and speeds up the POST process. A value of 0-1 additionally includes scanning of local devices as the Ethernet controller and the PCI bridge. 0-2 would include all those devices outside the CC5-

RAVE on the backplane, that are attached directly w/o an additional PCI bridge on the I/O module. If the I/O module is equipped with a PCI bridge, detecting of devices behind this bridge (bus 3) would require setting of this option to 'ALL'.

L2 Cache

For maximum performance, leave the CPU second level cache enabled. Disabling the cache will slow down code processing dramatically, so this is merely a feature for testing and debugging.

Show BIOS Messages, Delay After Messages

This parameter can be adjusted to the value that most fits your needs. From 'Few' to 'Normal' or 'More', select the verbosity of BIOS messages you prefer. The 'DbgMode' is suitable when using the internal debugger. The amount of delay inserted after outputting each message to the screen can be controlled by the parameters 'Normal', 'FastBoot' and 'DbgMode'.

Behaviour after Initializing Expansion ROM

This parameter controls the action method which takes place when detecting external BIOS code on POST, typically provided on bootable I/O boards as SCSI adapters and networking interface controllers. You can select here to continue POST ('Normal'), or to append a '5s delay', or to wait for pressing a key ('Press key'), or to enter the 'Debugger'.

Select Primary VGA

- ▶ The CC5-RAVE is equipped with an i810 internal video display controller. Select 'CC5/i810' in order to use the boards DVI-I connector as primary display. In order to select an external video graphics adapter card, set the parameter value to 1.PCI ... 10.PCI accordingly. A setting of 'Legacy' establishes the VGA adapter as primary, which has been scanned as last video device during the enumeration of PCI components. The selectable options follow here in detail:
- ▶ CC5/i810 (factory default): Selects the VGA on CC5 (i810 internal graphics device) as the primary VGA device
- ▶ Legacy (old GS algorithm): In previous BIOS versions that VGA device became the primary VGA device which was found on the highest PCI bus. If more than one VGA device was found on that bus, the VGA device with the highest device number became primary VGA.
- ▶ 1. PCI: Primary VGA device becomes that VGA device which is found on the lowest PCI bus above CC5 (on CC5 systems this is always the first CompactPCI bus). If more than one VGA device is found on that bus, the VGA device with the lowest device number becomes primary VGA
- ▶ 2. -9. PCI:
- ▶ 10. PCI: Primary VGA device becomes that VGA device which is found on the highest PCI bus. If more than one VGA device is found on that bus, the VGA device with the highest device number becomes primary VGA

Select Ethernet BIOS

Booting from the network requires the on-board Ethernet controller BIOS to be activated. For connecting to Linux servers, EtherBoot (GPL) is a suitable selection. The Intel PXE currently is restricted to the 82559 Ethernet controller chip, which is available on the CC5-RAVE on special request only (normally, the 82559ER is stuffed, which contains a 82559 subset). The BootMng option requires additional third-party software.

Digital DVI Support

The digital part of the DVI connector (DVI-D) is enabled in the BIOS. This includes initialization of the i810 GMCH and the Panellink Transmitter. The entry "Digital DVI (-D) support" in the setup screen "Custom Configuration" allows to control this feature.

The available options are:

CoreBIOS : DVI-D is enabled by the core-BIOS (i.e. code written by EKF)

VideoBIOS: DVI-D is enabled by Video BIOS (Intel Video BIOS with AIM)

Disabled: DVI-D output is switched off

Please note:

Use of the option CoreBIOS is highly recommended. There have been issues observed with the Intel Video BIOS (option VideoBIOS). As a workaround, EKF has written own DVI-D initialization code (enabled with option CoreBIOS, which is also the factory default). The only disadvantage of CoreBIOS is that under Windows 2000 the Intel graphics driver (PV6.6 and older) always disables DVI-D when Windows starts. This effect should not occur when using the Intel Video BIOS (says Intel, although in fact it does, as of current). Therefore, to reenale DVI-D under Windows, EKF has written a software module (windvi), which can be downloaded from EKF's website. It should be unpacked and copied to the autostart folder, in order to compensate the erroneous Intel graphics driver, until VideoBIOS is available bug-free. Linux and DOS both don't disable DVI-D, hence there is no need for executing the windvi utility after OS start.

BIOS Standard I/O Port

Operation of the CC5-RAVE as a desktop PC requires monitor and keyboard to be attached (CC6-ACID needed for KBD). As an alternative, I/O might be redirected to the 'COM1' or 'COM2' serial interfaces (also requires CC6-ACID super I/O board). Attach a VT100 like ASCII terminal (or PC equipped with a terminal emulation program) to the selected port. Standard I/O redirection will be aborted on pressing any key on the PS/2 keyboard (the CMOS parameter itself is not changed).

BIOS Setup I/O Port

In a likewise manner as the standard I/O redirection, the BIOS setup procedure might be redirected to the 'COM1' or 'COM2' serial interfaces. Setup I/O redirection will be aborted on pressing any key on the PS/2 keyboard (the CMOS parameter itself remains unchanged).

BIOS Debugger I/O Port

In a likewise manner as the standard I/O redirection, the debugger operation might be redirected to the 'COM1' or 'COM2' serial interfaces. The BIOS is provided with an integrated debugger. The 'VGA/KEYB' mode requires the graphics hardware to work properly, and - more important - overwrites screen contents generated by the application which has to be debugged. Debugger I/O redirection cannot be aborted on pressing any key on the PS/2 keyboard

Call Debugger During POST

The integrated debugger can be invoked During POST. There are several levels available for selection:

Disabled: Do not invoke the debugger

Level 1: Invoke the debugger immediately before boot

Level 2: Invoke the debugger before ATA/IDE initializations (includes Level 1)

Level 3: Invoke the debugger after VGA initialization (includes Level 1 and 2)

Parallel Port Configuration

The parallel port can be configured by several parameters. If disabled, the settings of the remaining parameters have no meaning. If enabled, the parallel port IRQ (Level 5/7), address (278h, 378h, 3BCh), and operation mode (legacy centronics printer, SPP/EPP1.7, ECP/EPP1.7, SPP, SPP/EPP1.9, ECP, ECP/EPP1.9) can be configured to meet your individual needs.

Serial Port Configuration

The UART COM1/2 ports can be both individually disabled/enabled and configured with respect to their I/O addresses and IRQ levels. The available addressing range is 220h, 228h, 238h, 2E8h, 2F8h, 338h, 3E8h, 3F8h). You can select either of IRQ 3/4/5/7. If in doubt, select 3F8h IRQ 4 for UART1 (COM1) and 2F8h IRQ 3 for UART2 (COM2).

IDE Configuration Setup Screen

This screen provides for configuration of both the Primary IDE and the Secondary IDE interfaces. Each port can be individually enabled/disabled.

```

System Bios Setup - IDE Configuration
(c) 2002 by EKF Elektronik GmbH, D-59065 Hamm, Germany

Primary IDE port on CC5/CC6:
-----
IO access Prim. IDE (1F0) : >Enabled
U-DMA/66 support:
  CC5 only / with CC6 : AUTO/DIS
Master (IDE 0) Timing Mode: 7 (111)
  "   IORDY Sample Point : 3 clocks
  "   Recovery Time      : 1 clocks
Slave (IDE1) Timing Mode : 7 (111)
  "   IORDY Sample Point : 3 clocks
  "   Recovery Time      : 1 clocks

Secondary IDE port on CC5 (CFA):
-----
IO access Sec. IDE (170) : Enabled
U-DMA/66 support        : N/A
Master (IDE 2) Timing Mode: 4 (100)
  "   IORDY Sample Point : 4 clocks
  "   Recovery Time      : 3 clocks
Slave (IDE 3) Timing Mode : N/A
  "   IORDY Sample Point : N/A
  "   Recovery Time      : N/A

↑/↓/←/→/<CR>/<Tab> to select or <PgUp>/<PgDn>/+/- to modify
<ESC> to return to main menu
  
```

Primary IDE Port

UDMA66

On the CC5-RAVE, the primary IDE port is dedicated to the harddisk (master) and CD-ROM (slave) drives. Normally, this port must be enabled. You can select the method how the disk operating system will detect attached drives at the Primary IDE interface as Ultra ATA/66. You will have to distinct between two hardware configurations:

1. The CC5-RAVE comes without the super-I/O companion board CC6-ACID. The UDMA options at your choice are 'Dis' (disable, never allow UDMA disk operation) and 'Auto' (the IDE cable type in use, 40- or 80-pos., is detected in order to check if UDMA operation could be allowed).
2. The CC5-RAVE is accompanied by the CC6-ACID Low-Pin-Count super-I/O interface board, usually with the harddisk being mounted directly to this card. The available options are 'Dis' (never allow UDMA), 'Auto' (for auto-detecting of 80-pos. IDE cables), and 'Ena' (for simulating the 80-pos. IDE cable being present - this option is required for an HDD directly mounted on the CC6-ACID).

The UDMA capability is passed to the operating system. The actual data transfer mode however depends on the drives built in UDMA features and the treatment of the operating system (e.g. DOS will not be able to activate the UDMA mode). The BIOS itself does not use UDMA during POST.

Master/Slave Timing Mode

Individually selectable for master and slave, this 3-bit (0-7) parameter controls various access modes to the drives. Detailed information can be derived from the Intel IDE Programmer's Reference Manual at <http://developer.intel.com/design/chipsets/manuals/298236.htm>.

IORDY Sampling Point

Individually selectable for master and slave, this parameter controls a delay time for the drives IORDY signal (3-5 clock cycles). Detailed information see above.

Recovery Time

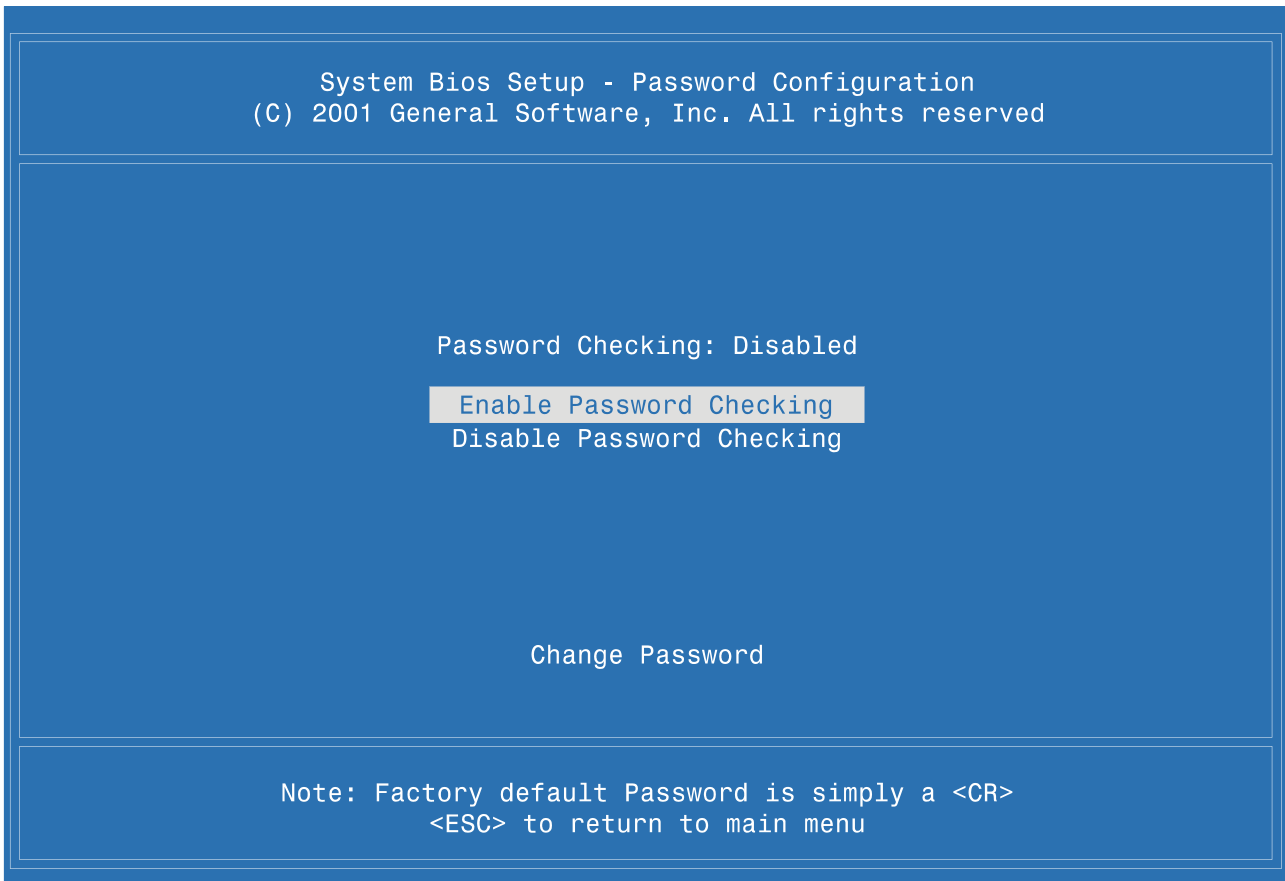
Individually selectable for master and slave, this parameter controls delays to recover from drive access (1-4 clock cycles). Detailed information see above.

Secondary IDE Port

This port is reserved for the ATA CompactFlash drive on the CC5-RAVE. The options are identical to the primary IDE port but the UDMA parameter. No IDE slave device is attached.

Password Configuration

This screen allows to define a password for the system, that is requested anytime before proceeding to either the BIOS Setup or the OS boot, if password checking is enabled. This feature therefore adds some level of security to the system in preventing unauthorized people from altering the BIOS setup parameters and starting the OS.



The factory default password is 'none' (simply enter <CR>, if password checking has been enabled without prior defining a user password). The option 'Change Password' allows to define a user specific key.

If you can't remember the password, there are two possible solutions to this problem: Either enter the debug mode (the integrated debugger can be invoked prior to entering the unknown password). Now alter the content of CMOS memory cells at the addresses 39h and 3Ah to 0 (this resets the password to 0000h = 'none'). When resetting the system, a CMOS Bad Checksum situation is detected, which causes the BIOS to replace the CMOS parameters to its factory defaults.

The second, harder way would be to carefully remove (soldering iron needed) the accumulator from the CC5-RAVE, which is responsible for data retention in the CMOS RAM. After a few seconds, insert (solder) the accumulator again. When switching on the system, the BIOS detects a checksum failure due to deleted data and installs the factory default parameters. All previous settings will be gone lost, e.g. date and time. You will have to reconfigure all BIOS settings now.

Burn-In Diagnostics Routines Setup Screen

Embedded systems may require automated burn-in testing in the development cycle. This facility is provided directly in the system's BIOS through the Burn-In Diagnostics routines setup screen. To use the system, selectively enable or disable features to be tested, and then enable the "Tests Begin on ESC?" option to cause the system test suite to be invoked. To repeat the system test battery continuously, you should also enable the "Continuous Testing" option. When continuous testing is started, the system will continue until an error is encountered. Caution: The disk I/O diagnostics perform write operations on those drives; therefore, only spare drives should be used which do not contain data that could be harmed by the test. Caution: The keyboard test may fail when in fact the hardware is operating within reasonable limits. This is because although the device may produce occasional errors, the BIOS retries operations when failures occur during normal operation of the system.

System Bios Setup - Burn-In Diagnostics			
(C) 2001 General Software, Inc. All rights reserved			
CPU Core	:>Disabled	BIOS Video Services	: Disabled
Floating Point Core	: Disabled	BIOS Equipment Services	: Disabled
Protected Mode	: Disabled	BIOS Low Memory Size	: Disabled
Low Memory (<1MB)	: Disabled	BIOS Block Disk Services	: Disabled
Extended Memory (>1MB)	: Disabled	BIOS Serial Services	: Disabled
DMA Controller(s)	: Disabled	BIOS System Services	: Disabled
CPU Int Controller(s)	: No Hdwr	BIOS Keyboard Services	: Disabled
Real-Time Clock	: Disabled	BIOS Parallel Services	: Disabled
Keyboard Controller	: Disabled	BIOS Time/Date Services	: Disabled
Video Controller/RAM	: Disabled	BIOS User Timer Tick	: Disabled
A20 Gate	: Disabled	Floppy Disk I/O	: Disabled
CPU Timer Controller	: No Hdwr	IDE Disk I/O	: Disabled
CMOS RAM & Battery	: Disabled	ROM Disk I/O	: No Hdwr
PC/AT Keyboard	: Disabled	RAM Disk I/O	: No Hdwr
Flash Read/Write/Update	: Disabled	RFD Disk I/O	: No Hdwr
Continuous Testing	: Disabled	Tests Begin on ESC?	: Disabled

↑/↓/←/→/<CR>/<Tab> to select or <PgUp>/<PgDn>/+/- to modify
<ESC> to return to main menu

System BIOS Debugger

The Embedded BIOS Integrated Debugger may be invoked from the Setup Screen main menu, or on encountering selected levels of the POST code (see the Custom Configuration Setup Screen - 'Call Debugger dur. POST' options), or by pressing <CTRL> <SHIFT> ! at any time. Once invoked, the debugger will display the debugger prompt:

EB43DBG:

and await debugger commands. To resume back to the Setup Screen main menu, type the following command, which instructs the debugger to "go":

EB43DBG: G <ENTER>

```
Type HELP for help, or G <Enter> to resume SETUP.
```

```
Embedded BIOS Debugger Breakpoint Trap
```

```
EAX = 00001D90  CS:EIP = 1D90:00000287  EFL = 0000E000  pl nz .. na .. po ?? nc
EBX = 756E1F0A  SS:ESP = 0000:0001FF2  EBP = 0000CF0  .. NT IOPL2 nv up di ..
ECX = 00002000  DS:ESI = E000:0003061C  FS = 2B18      .. .. id vp vi al vm rf
EDX = 0000184F  ES:EDI = 9F80:08001C38  GS = 0000
1D90:00000287 add [bx+si], al
```

```
EBDEBUG:
```

The debugger can be a valuable tool to aid the board verification process on new designs similar to the evaluation board. It supports a DOS SYMDEB-style command line interface, and can be used on the main console's keyboard and screen, or over a redirected connection to a terminal program (see "Console Redirection").

To activate the debugger at any time from the main console, press the left shift and the control keys together. A display will appear, containing the title, "Embedded BIOS Debugger Breakpoint Trap" and a snapshot of the processor general registers. To leave the debugger and resume the interrupted activity (whether POST, BIOS, DOS, Windows, or an application program), enter the "G" command (short for "go") and press ENTER. If you were at a DOS prompt when you entered

the debugger, then DOS will still be waiting for its command, and will not prompt again until you press ENTER again. The debugger can also be entered from the Setup Screen System, and as a boot activity (see "Basic CMOS Configuration Screen"), as a last ditch effort during board bring-up and development if no bootable device is available. If your version of DOS, an application, or any OEM-supplied BIOS extensions have debugging code (i.e., "INT 3" instructions) remaining, then these will invoke the debugger automatically as an option. To continue, use the "G" command. The debugger will respond on INT 3 instructions only if enabled (see the Basic CMOS Configuration Setup Screen). Normally this option is set to 'Ignored', so that debugging code in the application will not cause the debugger to be invoked.

A complete discussion of the debugger is beyond the scope of this chapter; however, complete documentation is available from General Software.

Console Redirection

The system can operate either with a standard PC/AT or PS/2 keyboard and VGA video monitor, or with a special emulation of a console over an RS232 cable connected to a host computer running a terminal program. When using the Console Redirection feature, the BIOS automatically switches its keyboard and screen functions to serial I/O over COM1/COM2 on the board (CC6-ACID required in addition to the CC5-RAVE). The hardware connection to the host computer requires a null modem cable.

Console redirection can be activated by means of the Custom Configuration Setup Screen. Individual selection of the best suited physical port for BIOS Standard I/O, Setup I/O and Debugger I/O is provided.

The software on the target can be any terminal emulation program that supports ANSI terminal mode, using 9600 baud, no parity, and one stop bit (Note: This can be modified by EKF during BIOS adaptation.) The program must be set to not use flow control, or the console may seem to stall or not accept input.

Caution if using HYPERTERMINAL from a PC: HYPERTERMINAL's default setting is to use flow control, which will render the console inoperative. To change this, create a new session, change the flow control setting to "none", save the session, and exit HYPERTERMINAL. Then reinvoke HYPERTERMINAL with the session and it will operate with the new flow control setting.

Addendum A: Embedded BIOS POST Codes

Embedded BIOS writes progress codes, also known as POST codes, to I/O port 80H during POST, in order to provide information to system integrators and developers about system faults. These POST are not displayed on the screen. However, the POST codes can be displayed by means of additional emulator equipment. For more information about POST codes, contact General Software.

Mnemonic	Code	System Progress Report
POST_STATUS_START	00h	Start POST (BIOS is executing).
POST_STATUS_CPUTEST	01h	Start CPU register test.
POST_STATUS_DELAY	02h	Start power-on delay.
POST_STATUS_DELAYDONE	03h	Power-on delay finished.
POST_STATUS_KDBATRDY	04h	Keyboard BAT finished.
POST_STATUS_DISABSHADOW	05h	Disable shadowing & cache.
POST_STATUS_CALCKSUM	06h	Compute ROM CRC, wait for KBC.
POST_STATUS_CKSUMGOOD	07h	CRC okay, KBC ready.
POST_STATUS_BATVRFY	08h	Verifying BAT command to KB.
POST_STATUS_KBDCMD	09h	Start KBC command.
POST_STATUS_KBDDATA	0ah	Start KBC data.
POST_STATUS_BLKUNBLK	0bh	Start pin 23,24 blocking & unblocking.
POST_STATUS_KBDNOP	0ch	Start KBC NOP command.
POST_STATUS_SHUTTEST	0dh	Test CMOS RAM shutdown register.
POST_STATUS_CMOSDIAG	0eh	Check CMOS checksum.
POST_STATUS_CMOSINIT	0fh	Initialize CMOS contents.
POST_STATUS_CMOSSTATUS	10h	Initialize CMOS status for date/time.
POST_STATUS_DISABDMAINT	11h	Disable DMA, PICs.
POST_STATUS_DISABPORTB	12h	Disable Port B, video display.
POST_STATUS_BOARD	13h	Initialize board, start memory bank detection.
POST_STATUS_TESTTIMER	14h	Start timer tests.
POST_STATUS_TESTTIMER2	15h	Test 8254 T2, for speaker, port B.
POST_STATUS_TESTTIMER1	16h	Test 8254 T1, for refresh.
POST_STATUS_TESTTIMER0	17h	Test 8254 T0, for 18.2Hz.
POST_STATUS_MEMREFRESH	18h	Start memory refresh.
POST_STATUS_TESTREFRESH	19h	Test memory refresh.
POST_STATUS_TEST15US	1ah	Test 15usec refresh ON/OFF time.
POST_STATUS_TEST64KB	1bh	Test base 64KB memory.
POST_STATUS_TESTDATA	1ch	Test data lines.
POST_STATUS_TESTADDR	20h	Test address lines.
POST_STATUS_TESTPARITY	21h	Test parity (toggling).
POST_STATUS_TESTMEMRDWR	22h	Test Base 64KB memory.
POST_STATUS_SYSINIT	23h	Prepare system for IVT initialization.
POST_STATUS_INITVECTORS	24h	Initialize vector table.
POST_STATUS_8042TURBO	25h	Read 8042 for turbo switch setting.
POST_STATUS_POSTTTURBO	26h	Initialize turbo data.
POST_STATUS_POSTVECTORS	27h	Modification of IVT.
POST_STATUS_MONOMODE	28h	Video in monochrome mode verified.
POST_STATUS_COLORMODE	29h	Video in color mode verified.
POST_STATUS_TOGGLEPARITY	2ah	Toggle parity before video ROM test.
POST_STATUS_INITBEFOREVIDEO	2bh	Initialize before video ROM check.
POST_STATUS_VIDEOROM	2ch	Passing control to video ROM.
POST_STATUS_POSTVIDEO	2dh	Control returned from video ROM.
POST_STATUS_CHECKEGAVGA	2eh	Check for EGA/VGA adapter.
POST_STATUS_TESTVIDEOMEMORY	2fh	No EGA/VGA found, test video memory.
POST_STATUS_RETRACE	30h	Scan for video retrace signal.
POST_STATUS_ALTDISPLAY	31h	Primary retrace failed.
POST_STATUS_ALTRETRACE	32h	Alternate found.
POST_STATUS_VRFYSWADAPTER	33h	Verify video switches.
POST_STATUS_SETDISPMODE	34h	Establish display mode.
POST_STATUS_CHECKSEG40A	35h	Initialize ROM BIOS data area.
POST_STATUS_SETCURSOR	36h	Set cursor for power-on msg.
POST_STATUS_PWRONDISPLAY	37h	Display power-on message.
POST_STATUS_SAVECURSOR	38h	Save cursor position.
POST_STATUS_BIOSIDENT	39h	Display BIOS identification string.

POST_STATUS_HITDEL 3ah Display "Hit to ..." message.
 POST_STATUS_VIRTUAL 40h Prepare protected mode test.
 POST_STATUS_DESCR 41h Prepare descriptor tables.
 POST_STATUS_ENTERVM 42h Enter virtual mode for memory test.
 POST_STATUS_ENABINT 43h Enable interrupts for diagnostics mode.
 POST_STATUS_CHECKWRAP1 44h Initialize data for memory wrap test.
 POST_STATUS_CHECKWRAP2 45h Test for wrap, find total memory size.
 POST_STATUS_HIGHPATTERNS 46h Write extended memory test patterns.
 POST_STATUS_LOWPATTERNS 47h Write conventional memory test patterns.
 POST_STATUS_FINDLOWMEM 48h Find low memory size from patterns.
 POST_STATUS_FINDHIMEM 49h Find high memory size from patterns.
 POST_STATUS_CHECKSEG40B 4ah Verify ROM BIOS data area again.
 POST_STATUS_CHECKDEL 4bh Check for pressed.
 POST_STATUS_CLEXTMEM 4ch Clear extended memory for soft reset.
 POST_STATUS_SAVEMEMSIZE 4dh Save memory size.
 POST_STATUS_COLD64TEST 4eh Cold boot: Display 1st 64KB memtest.
 POST_STATUS_COLDLOWTEST 4fh Cold boot: Test all of low memory.
 POST_STATUS_ADJUSTLOW 50h Adjust memory size for EBDA usage.
 POST_STATUS_COLDHITEST 51h Cold boot: Test high memory.
 POST_STATUS_REALMODETEST 52h Prepare for shutdown to real mode.
 POST_STATUS_ENTERREAL 53h Return to real mode.
 POST_STATUS_SHUTDOWN 54h Shutdown successful.
 POST_STATUS_DISABA20 55h Disable A20 line.
 POST_STATUS_CHECKSEG40C 56h Check ROM BIOS data area again.
 POST_STATUS_CHECKSEG40D 57h Check ROM BIOS data area again.
 POST_STATUS_CLRHITDEL 58h Clear "Hit " message.
 POST_STATUS_TESTDMAPAGE 59h Test DMA page register file.
 POST_STATUS_VRFYDISPMEM 60h Verify from display memory.
 POST_STATUS_TESTDMA0BASE 61h Test DMA0 base register.
 POST_STATUS_TESTDMA1BASE 62h Test DMA1 base register.
 POST_STATUS_CHECKSEG40E 63h Checking ROM BIOS data area again.
 POST_STATUS_CHECKSEG40F 64h Checking ROM BIOS data area again.
 POST_STATUS_PROGDMA 65h Program DMA controllers.
 POST_STATUS_INITINTCTRL 66h Initialize PICs.
 POST_STATUS_STARTKBDTEST 67h Start keyboard test.
 POST_STATUS_KBDRESET 80h Issue KB reset command.
 POST_STATUS_CHECKSTUCKKEYS 81h Check for stuck keys.
 POST_STATUS_INITCIRCBUFFER 82h Initialize circular buffer.
 POST_STATUS_CHECKLOCKEDKEYS 83h Check for locked keys.
 POST_STATUS_MEMSIZEMISMATCH 84h Check for memory size mismatch.
 POST_STATUS_PASSWORD 85h Check for password or bypass setup.
 POST_STATUS_BEFORESETUP 86h Password accepted.
 POST_STATUS_CALLSETUP 87h Entering setup system.
 POST_STATUS_POSTSETUP 88h Setup system exited.
 POST_STATUS_DISPPWRON 89h Display power-on screen message.
 POST_STATUS_DISPWAIT 8ah Display "Wait..." message.
 POST_STATUS_ENABSHADOW 8bh Shadow system & video BIOS.
 POST_STATUS_STDCMOSSETUP 8ch Load standard setup values from CMOS.
 POST_STATUS_MOUSE 8dh Test and initialize mouse.
 POST_STATUS_FLOPPY 8eh Test floppy disks.
 POST_STATUS_CONFIGFLOPPY 8fh Configure floppy drives.
 POST_STATUS_IDE 90h Test hard disks.
 POST_STATUS_CONFIGIDE 91h Configure IDE drives.
 POST_STATUS_CHECKSEG40G 92h Checking ROM BIOS data area.
 POST_STATUS_CHECKSEG40H 93h Checking ROM BIOS data area.
 POST_STATUS_SETMEMSIZE 94h Set base & extended memory sizes.
 POST_STATUS_SIZEADJUST 95h Adjust low memory size for EBDA.
 POST_STATUS_INITC8000 96h Initialize before calling C800h ROM.
 POST_STATUS_CALLC8000 97h Call ROM BIOS extension at C800h.
 POST_STATUS_POSTC8000 98h ROM C800h extension returned.
 POST_STATUS_TIMERPRNBASE 99h Configure timer/printer data.
 POST_STATUS_SERIALBASE 9ah Configure serial port base addresses.
 POST_STATUS_INITBEFORENPX 9bh Prepare to initialize coprocessor.
 POST_STATUS_INITNPX 9ch Initialize numeric coprocessor.
 POST_STATUS_POSTNPX 9dh Numeric coprocessor initialized.
 POST_STATUS_CHECKLOCKS 9eh Check KB settings.
 POST_STATUS_ISSUEKBDID 9fh Issue keyboard ID command.

POST_STATUS_RESETID 0a0h KB ID flag reset.
POST_STATUS_TESTCACHE 0a1h Test cache memory.
POST_STATUS_DISPSOFTERR 0a2h Display soft errors.
POST_STATUS_TYPEMATIC 0a3h Set keyboard typematic rate.
POST_STATUS_MEMWAIT 0a4h Program memory wait states.
POST_STATUS_CLRSCR 0a5h Clear screen.
POST_STATUS_ENABPTYNMI 0a6h Enable parity and NMIs.
POST_STATUS_INITE000 0a7h Initialize before calling ROM at E000h.
POST_STATUS_CALLE000 0a8h Call ROM BIOS extension at E000h.
POST_STATUS_POSTE000 0a9h ROM extension returned.
POST_STATUS_DISPCONFIG 0b0h Display system configuration box.
POST_STATUS_INT19BOOT 00h Call INT 19h bootstrap loader.
POST_STATUS_LOWMEMEXH 0b1h Test low memory exhaustively.
POST_STATUS_EXTMEMEXH 0b2h Test extended memory exhaustively.
POST_STATUS_PCIENUM 0b3h Enumerate PCI buses.

Addendum B: Embedded BIOS Beep & Blink Codes

Embedded BIOS tests much of the system hardware early in POST before messages can be displayed on the screen. When system failures are encountered at these early stages, POST uses beep codes (a sequence of tones on the speaker) and blinking sequences (visual signals, output to the floppy drive LED) to identify the source of the error. The following is a comprehensive list of POST beep & blink codes for the system BIOS. BIOS extensions, such as VGA ROMs and SCSI adapter ROMs, may use their own beep codes, including short/long sequences, or possibly beep codes that sound like the ones below. When diagnosing a system failure, remove these adapters if possible before making a final determination of the actual POST test that failed.

Mnemonic Code	Beep Count	Description of Problem
POST_BEEP_REFRESH	1	Memory refresh is not working
POST_BEEP_PARITY	2	Parity error found in 1st 64KB of memory
POST_BEEP_BASE64KB	3	Memory test of 1st 64KB failed
POST_BEEP_TIMER	4	T1 timer test failed
POST_BEEP_CPU	5	CPU test failed
POST_BEEP_GATEA20	6	Gate A20 test failed
POST_BEEP_DMA	7	DMA page/base register test failed
POST_BEEP_VIDEO	8	Video controller test failed
POST_BEEP_KEYBOARD	9	Keyboard test failed
POST_BEEP_SHUTDOWN	10	CMOS shutdown register test failed
POST_BEEP_CACHE	11	External cache test failed
POST_BEEP_BOARD	12	General board initialization failed
POST_BEEP_LOWMEM	13	Exhaustive low memory test failed
POST_BEEP_EXTMEM	14	Exhaustive extended memory test failed
POST_BEEP_CMOS	15	CMOS restart byte test failed
POST_BEEP_ADDRESS_LINE	16	Address line test failed
POST_BEEP_DATA_LINE	17	Data line test failed
POST_BEEP_INTERRUPT	18	Interrupt controller test failed
POST_BEEP_PASSWORD	1	Incorrect password used to access SETUP



EKF Elektronik GmbH
Philipp-Reis-Str. 4
D-59065 HAMM
(Germany)



Internet <http://www.ekf.de>
Fax. +49 (0)2381/6890-90
Tel. +49 (0)2381/6890-0
E-Mail sales@ekf.de