



# EP-P80P

## PCI Diagnostic & Troubleshooting Card.

### TRADEMARKS

All products and company names are trademarks or registered trademarks of their respective holders.

*These specifications are subject to change without notice.*

## **Table of Contents**

### ***Chapter 1: Introduction***

Component Checklist .....	5
Overview of EP-P80P .....	6

### ***Chapter 2: Card Features***

Directions .....	7
Detailed Board Layout .....	7

### ***Chapter 3: Diagnostic Card Usage***

System Voltages Detected .....	9
PCI Bus C/BE[3::0] # .....	9
AMI BIOS v6.24 Codes .....	10
Award BIOS v6.0 Codes .....	16

### ***Additional Sources of General Information***

#### **EPoX**

<http://www.epox.com>

<http://www.epox.com.tw>

<http://www.epox.nl>

#### **Intel Corporation**

<http://www.intel.com>

<http://developer.intel.com>

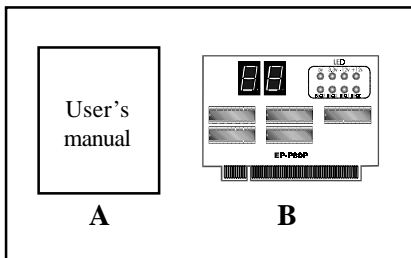
---

# ***Chapter 1***

## ***Introduction***

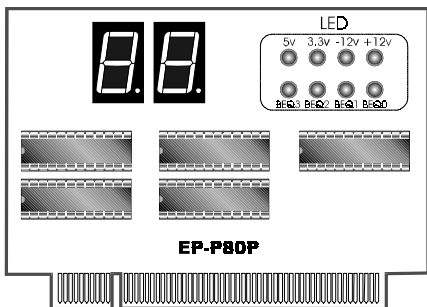
### **Components Checklist**

- A      EP-P80P User's manual
- B      EP-P80P PCI Diagnostic Card



## Overview of EP-P80P

- EP-P80P is a PCI based Diagnostic & Troubleshooting card for PCI local bus motherboards allowing realtime feedback of system boot status and current state of the PCI Byte Enable/Command signals.
- The EP-P80P can help IT professionals and system engineers in the troubleshooting and repair of non-responsive systems by obtaining system status direct from port I/O 80H. The card is also capable of decoding from the PCI bus C/BE as well checking the presence of +5, +3.3, -12 and 12V power lines.
- The EP-P80P manual includes a comprehensive list of common AMI and Award BIOS POST codes; see chapter 3.



# Chapter 2

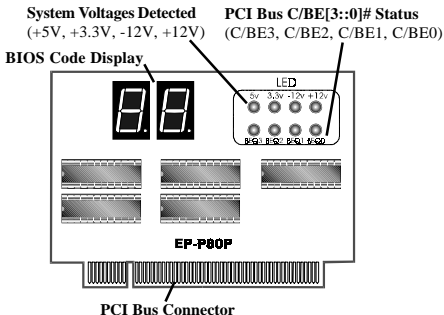
## Card Features

### Directions

Thank you for using EPoX products! This chapter will discuss some general steps needed in order to get your new diagnostic card installed and operational.

**Safety precautions:** Static electricity can severely damage your equipment. Handle the EP-P80P and any other device in your system with extreme care and avoid unnecessary contact with system components on the mainboard. Always work on an antistatic surface to avoid possible damage to the mainboard from static discharge. Always have the power supply unplugged and powered off when inserting and removing devices within the computer chassis.

### Detailed Board Layout





# **Chapter 3**

## **Diagnostic Card Usage**

### **System Voltages Detected**

These LED's represent the voltages they represent being present on the PCI bus. The diagnostic card is capable of detecting +5, +3.3, -12, and +12V from the PCI bus.

### **PCI Bus C/BE[3::0] # Description**

Note: A binary '1' below would represent the diagnostic card's corresponding LED being enabled (on), while '0' would be the LED disabled (off).

BEQ#:	PCI Bus
3 2 1 0	Command Type
0 0 0 0	Interrupt Acknowledge
0 0 0 1	Special Cycle
0 0 1 0	I/O Read
0 0 1 1	I/O Write
0 1 0 0	Reserved
0 1 0 1	Reserved
0 1 1 0	Memory Read
0 1 1 1	Memory Write
1 0 0 0	Reserved
1 0 0 1	Reserved
1 0 1 0	Configuration Read
1 0 1 1	Configuration Write
1 1 0 0	Memory Read Multiple
1 1 0 1	Dual Address Cycle
1 1 1 0	Memory Read Line
1 1 1 1	Memory Write and Invalidate

For more information about the PCI Bus C/BE signal please refer to *PCI Local Bus Specification Revision 2.0* or Intel datasheets.

**AMI BIOS v6.24 Codes**

Following is the checkpoint list in AMIBIOS in order of execution.

**Uncompressed INIT code checkpoints**

- D0 NMI is Disabled. CPU ID saved. Init code Checksum verification starting.
- D1 To do DMA init, Keyboard controller BAT test, start memory refresh and going to 4GB flat mode.
- D3 To start Memory sizing.
- D4 To comeback to real mode. Execute OEM patch. Set stack.
- D5 E000 ROM enabled. Init code is copied to segment 0 and control to be transferred to segment 0.
- D6 Control is in segment 0. To check <CTRL><HOME> key and verify main BIOS checksum. If either <CTRL><HOME> is pressed or main IOS checksum is bad, go to check point E0 else goto check point D7.
- D7 To pass control to Interface Module.
- D8 Main BIOS runtime code is to be decompressed.
- D9 Control to be passed to main BIOS in shadow RAM.

**Boot Block Recovery Code Check Points**

- E0 OnBoard Floppy Controller (if any) is initialized.  
To start base 512K memory test.
- E1 To initialise interrupt vector table.
- E2 To initialise DMA and interrupt controllers.
- E6 To enable floppy and timer IRQ, enable internal cache.
- ED Initialize floppy drive.
- EE Start looking for a diskette in drive A: and read 1st sector of the diskette.
- EF Floppy read error.
- F0 Start searching 'AMIBOOT.ROM' file in root directory.
- F1 MIBOOT.ROM' file not present in root directory.
- F2 Start reading FAT table and analyze FAT to find the clusters occupied by MIBOOT.ROM' file..
- F3 Start reading MIBOOT.ROM' file cluster by cluster.
- F4 MIBOOT.ROM' file not of proper size.
- F5 Disable internal cache.
- FB Detect Flash type present.



- FC Erase Flash.
- FD Program Flash.
- FF Flash program successful. BIOS is going to restart. Runtime code is uncompressed in F000 shadow ram
- 03 NMI is Disabled. To check soft reset/power-on.
- 05 BIOS stack set. Going to disable Cache if any.
- 06 POST code to be uncompressed.
- 07 CPU init and CPU data area init to be done.
- 08 CMOS checksum calculation to be done next.
- 0B Any initialization before keyboard BAT to be done next.
- 0C KB controller I/B free. To issue the BAT command to keyboard controller.
- 0E Any initialization after KB controller BAT to be done next.
- 0F Keyboard command byte to be written.
- 10 Going to issue Pin-23,24 blocking/unblocking command.
- 11 Going to check pressing of <INS> , <END> key during power-on.
- 12 To init CMOS if "Init CMOS in every boot" is set or <END> key is pressed. Going to disable DMA and Interrupt controllers.
- 13 Video display is disabled and port-B is initialized. Chipset init about to begin.
- 14 8254 timer test about to start.
- 19 About to start memory refresh test.
- 1A Memory Refresh line is toggling. Going to check 15us ON/OFF time.
- 23 To read 8042 input port and disable MegaKey GreenPC feature. Make BIOS code segment writeable.
- 24 To do any setup before Int vector init.
- 25 Interrupt vector initialization about to begin. To clear password if necessary.
- 27 Any initialization before setting video mode to be done.
- 28 Going for monochrome mode and color mode setting.
- 2A Different BUSes init (system, static, output devices) to start if present.
- 2B To give control for any setup required before optional video ROM check.
- 2C To look for optional video ROM and give control.
- 2D To give control to do any processing after video ROM returns control.
- 2E If EGA/VGA not found then do display memory R/W test.
- 2F EGA/VGA not found. Display memory R/W test about to begin.
- 30 Display memory R/W test passed. About to look for the retrace checking.

- 31 Display memory R/W test or retrace checking failed. To do alternate Display memory R/W test.
- 32 Alternate Display memory R/W test passed. To look for the alternate display retrace checking.
- 34 Video display checking over. Display mode to be set next.
- 37 Display mode set. Going to display the power on message.
- 38 Different BUSES init (input, IPL, general devices) to start if present.
- 39 Display different BUSES initialization error messages.
- 3A New cursor position read and saved. To display the Hit <DEL> message.
- 40 To prepare the descriptor tables.
- 42 To enter in virtual mode for memory test.
- 43 To enable interrupts for diagnostics mode.
- 44 To initialize data to check memory wrap around at 0:0.
- 45 Data initialized. Going to check for memory wrap around at 0:0 and finding the total system memory size.
- 46 Memory wrap around test done. Memory size calculation over. About to go for writing patterns to test memory.
- 47 Pattern to be tested written in extended memory. Going to write patterns in ase 640k memory.
- 48 Patterns written in base memory. Going to findout amount of memory below 1M memory.
- 49 Amount of memory below 1M found and verified. Going to findout amount of memory above 1M memory.
- 4B Amount of memory above 1M found and verified. Check for soft reset and going to clear memory below 1M for soft reset. (If power on, go to check point# 4Eh).
- 4C Memory below 1M cleared. (SOFT RESET) Going to clear memory above 1M.
- 4D Memory above 1M cleared. (SOFT RESET) Going to save the memory size. (Goto check point# 52h).
- 4E Memory test started. (NOT SOFT RESET) About to display the first 64k memory size.
- 4F Memory size display started. This will be updated during memory test. Going for sequential and random memory test.
- 50 Memory testing/initialization below 1M complete. Going to adjust displayed memory size for relocation/ shadow.
- 51 Memory size display adjusted due to relocation/ shadow. Memory test above 1M to follow.

- 52 Memory testing/initialization above 1M complete.  
Going to save memory size information.
- 53 Memory size information is saved. CPU registers are saved.  
Going to enter in real mode.
- 54 Shutdown successful, CPU in real mode. Going to disable  
gate A20 line and disable parity/NMI.
- 57 A20 address line, parity/NMI disable successful.  
Going to adjust memory size depending on relocation/shadow.
- 58 Memory size adjusted for relocation/shadow.  
Going to clear Hit <DEL> message.
- 59 Hit <DEL> message cleared. <WAIT...> message displayed.  
About to start DMA and interrupt controller test.
- 60 DMA page register test passed. To do DMA#1 base register test.
- 62 DMA#1 base register test passed. To do DMA#2 base register test.
- 65 DMA#2 base register test passed. To program DMA unit 1 and 2.
- 66 DMA unit 1 and 2 programming over.  
To initialize 8259 interrupt controller.
- 7F Extended NMI sources enabling is in progress.
- 80 Keyboard test started. clearing output buffer, checking for stuck key,  
to issue keyboard reset command.
- 81 Keyboard reset error/stuck key found. To issue keyboard controller  
interface test command.
- 82 Keyboard controller interface test over. To write command byte and  
init circular buffer.
- 83 Command byte written, Global data init done. To check for lock-key.
- 84 Lock-key checking over. To check for memory size mismatch with  
CMOS.
- 85 Memory size check done. To display soft error and check for password  
or bypass setup.
- 86 Password checked. About to do programming before setup.
- 87 Programming before setup complete. To uncompress SETUP code  
and execute CMOS setup.
- 88 Returned from CMOS setup program and screen is cleared.  
About to do programming after setup.
- 89 Programming after setup complete.  
Going to display power on screen message.
- 8B First screen message displayed. <WAIT...> message displayed.  
PS/2 Mouse check and extended BIOS data area allocation to be  
done.

- 8C Setup options programming after CMOS setup about to start.
- 8D Going for hard disk controller reset.
- 8F Hard disk controller reset done. Floppy setup to be done next.
- 91 Floppy setup complete. Hard disk setup to be done next.
- 95 Init of different BUSES optional ROMs from C800 to start.
- 96 Going to do any init before C800 optional ROM control.
- 97 Any init before C800 optional ROM control is over.  
Optional ROM check and control will be done next.
- 98 Optional ROM control is done. About to give control to do any  
required processing after optional ROM returns control and enable  
external cache.
- 99 Any initialization required after optional ROM test over.  
Going to setup timer data area and printer base address.
- 9A Return after setting timer and printer base address.  
Going to set the RS-232 base address.
- 9B Returned after RS-232 base address.  
Going to do any initialization before Coprocessor test.
- 9C Required initialization before Coprocessor is over.  
Going to initialize the Coprocessor next.
- 9D Coprocessor initialized.  
Going to do any initialization after Coprocessor test.
- 9E Initialization after Coprocessor test is complete. Going to check extd  
keyboard, keyboard ID and num-lock. Keyboard ID command to be
- A2 Going to display any soft errors.
- A3 Soft error display complete. Going to set keyboard typematic rate.
- A4 Keyboard typematic rate set. To program memory wait states.
- A5 Going to enable parity/NMI.
- A7 NMI and parity enabled. Going to do any initialization  
required before giving control to optional ROM at E000.
- A8 Initialization before E000 ROM control over.  
E000 ROM to get control next.
- A9 Returned from E000 ROM control. Going to do any  
initialization required after E000 optional ROM control.
- AA Initialization after E000 optional ROM control is over.  
Going to display the system configuration.
- AB To build MP table if needed.
- AC To uncompress DMI data and execute DMI POST init.
- B0 System configuration is displayed.
- B1 Going to copy any code to specific area.

00 Copying of code to specific area done. Going to give control to INT-19 boot loader.

For more information about AMI BIOS please refer to the AMI website at <http://www.ami.com>

**Award BIOS v6.0 Codes**

Following is the checkpoint list in Award BIOS in order of execution.

CFh Test CMOS R/W functionality.

C0h Early chipset initialization:

- Disable shadow RAM
- Disable L2 cache (socket 7 or below)
- Program basic chipset registers

C1h Detect memory

- Auto-detection of DRAM size, type and ECC.
- Auto-detection of L2 cache (socket 7 or below)

C3h Expand compressed BIOS code to DRAM

C5h Call chipset hook to copy BIOS back to E000 & F000 shadow RAM.

0h1 Expand the Xgroup codes locating in physical address 1000:0

02h Reserved

03h Initial Superio\_Early\_Init switch.

04h Reserved

05h 1. Blank out screen  
2. Clear CMOS error flag

06h Reserved

07h 1. Clear 8042 interface  
2. Initialize 8042 self-test

08h 1. Test special keyboard controller for Winbond 977 series Super I/O chips.  
2. Enable keyboard interface.

09h Reserved

0Ah 1. Disable PS/2 mouse interface (optional).  
2. Auto detect ports for keyboard & mouse followed by a port & interface swap (optional).  
3. Reset keyboard for Winbond 977 series Super I/O chips.

0Ch Reserved

0Dh Reserved

0Eh Test F000h segment shadow to see whether it is R/W-able or not. If test fails, keep beeping the speaker.

0Fh Reserved

10h Auto detect flash type to load appropriate flash R/W codes into the run time area in F000 for ESCD & DMI support.

- 11h Reserved
- 12h Use walking 1's algorithm to check out interface in CMOS circuitry. Also set real-time clock power status, and then check for override.
- 13h Reserved
- 14h Program chipset default values into chipset. Chipset default values are MODBINable by OEM customers.
- 15h Reserved
- 16h Initial Early\_Init\_Onboard\_Generator switch.
- 17h Reserved
- 18h Detect CPU information including brand, SMI type (Cyrix or Intel) and CPU level (586 or 686).
- 19h Reserved
- 1Ah Reserved
- 1Bh Initial interrupts vector table. If no special specified, all H/W interrupts are directed to PURIOUS\_INT\_HDLR & S/W interrupts to SPURIOUS\_soft\_HDLR.
- 1Ch Reserved
- 1Dh Initial EARLY\_PM\_INIT switch.
- 1Eh Reserved
- 1Fh Load keyboard matrix (notebook platform)
- 20h Reserved
- 21h HPM initialization (notebook platform)
- 22h Reserved
- 23h
1. Check validity of RTC value: e.g. a value of 5Ah is an invalid value for RTC minute.
  2. Load CMOS settings into BIOS stack. If CMOS checksum fails, use default value instead.
  3. Prepare BIOS resource map for PCI & PnP use. If ESCD is valid, take into consideration of the ESCD's legacy information.
  4. Onboard clock generator initialization. Disable respective clock resource to empty PCI & DIMM slots.
  5. Early PCI initialization:
    - Enumerate PCI bus number
    - Assign memory & I/O resource
    - Search for a valid VGA device & VGA BIOS, and put it into C000:0.
- 24h Reserved
- 25h Reserved
- 26h Reserved

- 27h Initialize INT 09 buffer
- 28h Reserved
- 29h
  1. Program CPU internal MTRR (P6 & PII) for 0-640K memory address.
  2. Initialize the APIC for Pentium class CPU.
  3. Program early chipset according to CMOS setup. Example:  
onboard IDE controller.
  4. Measure CPU speed.
  5. Invoke video BIOS.
- 2Ah Reserved
- 2Bh Reserved
- 2Ch Reserved
- 2Dh
  1. Initialize multi-language
  2. Put information on screen display, including Award title, CPU type, CPU speed ....
- 2Eh Reserved
- 2Fh Reserved
- 30h Reserved
- 31h Reserved
- 32h Reserved
- 33h Reset keyboard except Winbond 977 series Super I/O chips.
- 34h Reserved
- 35h Reserved
- 36h Reserved
- 37h Reserved
- 38h Reserved
- 39h Reserved
- 3Ah Reserved
- 3Bh Reserved
- 3Ch Test 8254
- 3Dh Reserved
- 3Eh Test 8259 interrupt mask bits for channel 1.
- 3Fh Reserved
- 40h Test 8259 interrupt mask bits for channel 2.
- 41h Reserved
- 42h Reserved
- 43h Test 8259 functionality.
- 44h Reserved
- 45h Reserved
- 46h Reserved



- 47h Initialize EISA slot
- 48h Reserved
- 49h
  1. Calculate total memory by testing the last doubleword of each 64K page.
  2. Program write allocation for AMD K5 CPU.
- 4Ah Reserved
- 4Bh Reserved
- 4Ch Reserved
- 4Dh Reserved
- 4Eh
  1. Program MTRR of M1 CPU
  2. Initialize L2 cache for P6 class CPU & program CPU with proper cacheable range.
  3. Initialize the APIC for P6 class CPU.
  4. On MP platform, adjust the cacheable range to smaller one in case the cacheable ranges between each CPU are not identical.
- 4Fh Reserved
- 50h Initialize USB
- 51h Reserved
- 52h Test all memory (clear all extended memory to 0)
- 53h Reserved
- 54h Reserved
- 55h Display number of processors (multi-processor platform)
- 56h Reserved
- 57h
  1. Display PnP logo
  2. Early ISA PnP initialization
    - Assign CSN to every ISA PnP device.
- 58h Reserved
- 59h Initialize the combined Trend Anti-Virus code.
- 5Ah Reserved
- 5Bh (Optional Feature) Show message for entering AWDFLASH.EXE from FDD
- 5Ch Reserved
- 5Dh
  1. Initialize Init\_Onboard\_Super\_IO switch.
  2. Initialize Init\_Onboard\_AUDIO switch.
- 5Eh Reserved
- 5Fh Reserved
- 60h Okay to enter Setup utility; i.e. not until this POST stage can users enter the CMOS setup utility.
- 61h Reserved

- 62h Reserved
- 63h Reserved
- 64h Reserved
- 65h Initialize PS/2 Mouse
- 66h Reserved
- 67h Prepare memory size information for function call:  
INT 15h ax=E820h
- 68h Reserved
- 69h Turn on L2 cache
- 6Ah Reserved
- 6Bh Program chipset registers according to items described in Setup & Auto-configuration table.
- 6Ch Reserved
- 6Dh
  1. Assign resources to all ISA PnP devices.
  2. Auto assign ports to onboard COM ports if the corresponding item in Setup is set to "AUTO".
- 6Eh Reserved
- 6Fh
  1. Initialize floppy controller
  2. Set up floppy related fields in 40:hardware.
- 70h Reserved
- 71h Reserved
- 72h Reserved
- 73h (Optional Feature) Enter AWDFLASH.EXE if :
  - AWDFLASH is found in floppy drive.
  - ALT+F2 is pressed
- 74h Reserved
- 75h Detect & install all IDE devices: HDD, LS120, ZIP, CDROM.....
- 76h Reserved
- 77h Detect serial ports & parallel ports.
- 78h Reserved
- 79h Reserved
- 7Ah Detect & install co-processor
- 7Bh Reserved
- 7Ch Reserved
- 7Dh Reserved
- 7Eh Reserved
- 7Fh
  1. Switch back to text mode if full screen logo is supported.
    - If errors occur, report errors & wait for keys
    - If no errors occur or F1 key is pressed to continue: wClear EPA

or customization logo.

80h Reserved

81h Reserved

### **E8POST.ASM starts**

82h 1. Call chipset power management hook.

2. Recover the text font used by EPA logo (not for full screen logo)

3. If password is set, ask for password.

83h Save all data in stack back to CMOS

84h Initialize ISA PnP boot devices

85h 1. USB final Initialization

2. NET PC: Build SYSID structure

3. Switch screen back to text mode

4. Set up ACPI table at top of memory.

5. Invoke ISA adapter ROMs

6. Assign IRQs to PCI devices

7. Initialize APM

8. Clear noise of IRQs.

86h Reserved

87h Reserved

88h Reserved

89h Reserved

90h Reserved

91h Reserved

92h Reserved

93h Read HDD boot sector information for Trend Anti-Virus code

94h 1. Enable L2 cache

2. Program boot up speed

3. Chipset final initialization.

4. Power management final initialization

5. Clear screen & display summary table

6. Program K6 write allocation

7. Program P6 class write combining

95h 1. Program daylight saving

2. Update keyboard LED & typematic rate

96h 1. Build MP table

2. Build & update ESCD

3. Set CMOS century to 20h or 19h

4. Load CMOS time into DOS timer tick

5. Build MSIRQ routing table.  
FFh Boot attempt (INT 19h)

For more information about Award BIOS please refer to the Award website at <http://www.award.com.tw>