# PI-6486V

## All-In-One STPC Client x86 CPU card with on-board VGA interface and Flash disk



Printed Aug. 2000

# Table Of Contents

# 1. Introductions

## 1.1 Overview

The PI-6486V comes equipped with SGS-THOMSON ST PC Client Pentium performance x86 CPU , it offer an 64-bit DRAM-controller, standard chipset functions (such as interrupt, DMA,counter/timer..etc), ISA bus controller. The PI-6486V also includes a high speed 32-bit TFT/LCDVGA controller with 135 MHz RAMDAC that allows connection of a TFT/LCD/CRT monitor

In addition, the PI-6486V has one PS/2 mouse port, two serial ports, a bi-direction high-speed parallel port, an PCI-bus hard disk drive interface and a floppy disk controller

Built-in 1.7MB Flash disk without installing any additional Flash Devices. In addition, the PI-6486V also has two onboard sockets can accept up to two DiskOnChp disk

The watchdog timer ensures that the CPU will be reset, if it stops due to a program or EMI problem, allowing the PI-6486V to be used in stand-alone systems or unmanned environments.

With its industrial grade reliability, The PI-6486V can withstand continuous operation in harsh industrial environments under 60C°(140F°) temperatures.

This card is specially designed as a compact all-in-one CPU card which incorporates single voltage power supply(+5V). This six layer CPU card turns any system into a 16-bit Pentium performance computer.

## 1.2 Packing list

- One half size CPU card
- One FDD cable (34-pin),one HDD cable (40-pin)
- One parallel port adapter (26-pin) kit
- One 5-pin DIN to 6 pin mini-DIN keyboard adapter
- One 3.5" utility diskette

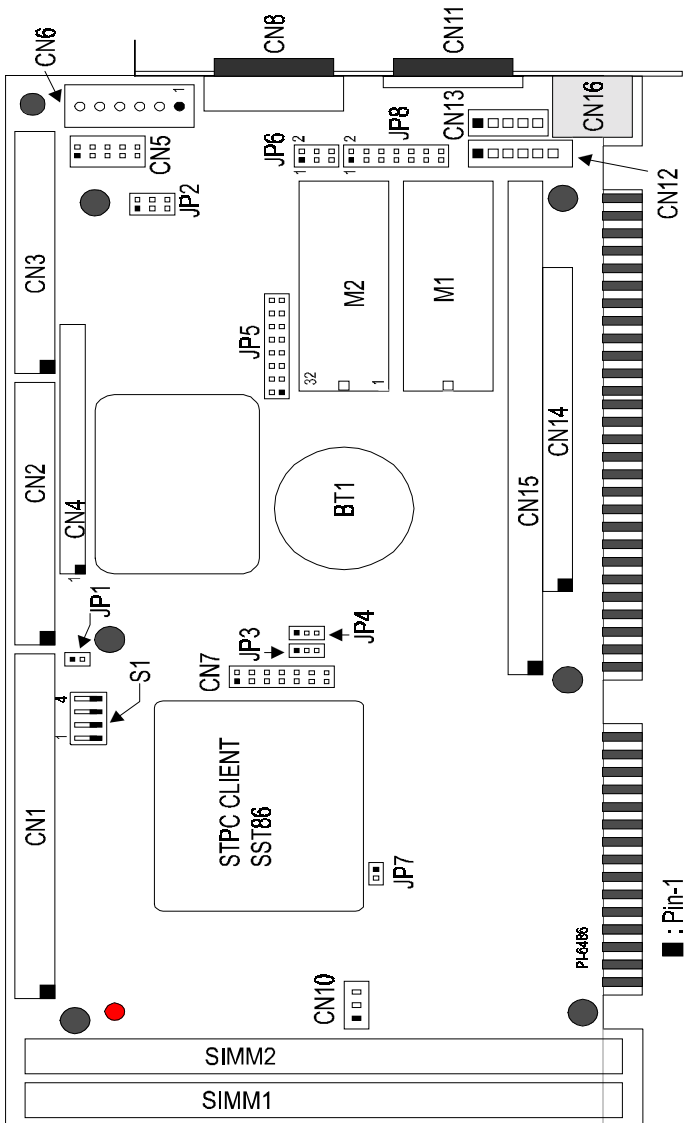




## 1.3 Features

- Fully IBM PC/AT Compatible
- On Board 66/75MHz ST PC Client 5ST86 Processor
- Internal 8K Bytes Unified Write-Back cache
- On-board CRT/LCD Flat Panel VGA Controller(For PI-6486V)
- On board CRT VGA Controller up to 1280x1024 resolution(For PI-6486C)
- Two 72-pin SIMM sockets up to 64MB(FP or EDO RAM)
- On-Board 4MB up to 8MB EDO-RAM for system memory(Option)
- Built-in Floppy Disk Driver Controller.
- On-board One Port PCI bus Enhance IDE Hard Disk Driver Controller
- Two 16550 Compatible RS-232C Serial Ports.
- One SPP/ECP/EPP Bidirection Parallel Port.
- On Board 2MB Flash ROM(Option) Disk(SSD) and Two 32-pin RAM/Flash ROM Sockets
- Provide Two M-Systems DiskOnChip Sockets
- Support PS/2 mouse connector
- 16 level timeout intervals watchdog timer from 1-160 seconds
- 16-bit PC/104 Piggiback connector
- Support Front Pannel KBD Input Connector.
- Support External Power Input Connector(P8)
- Single Power Supply(+5V/3A).
- Half Size buffered ISA-BUS(PC/AT) interface

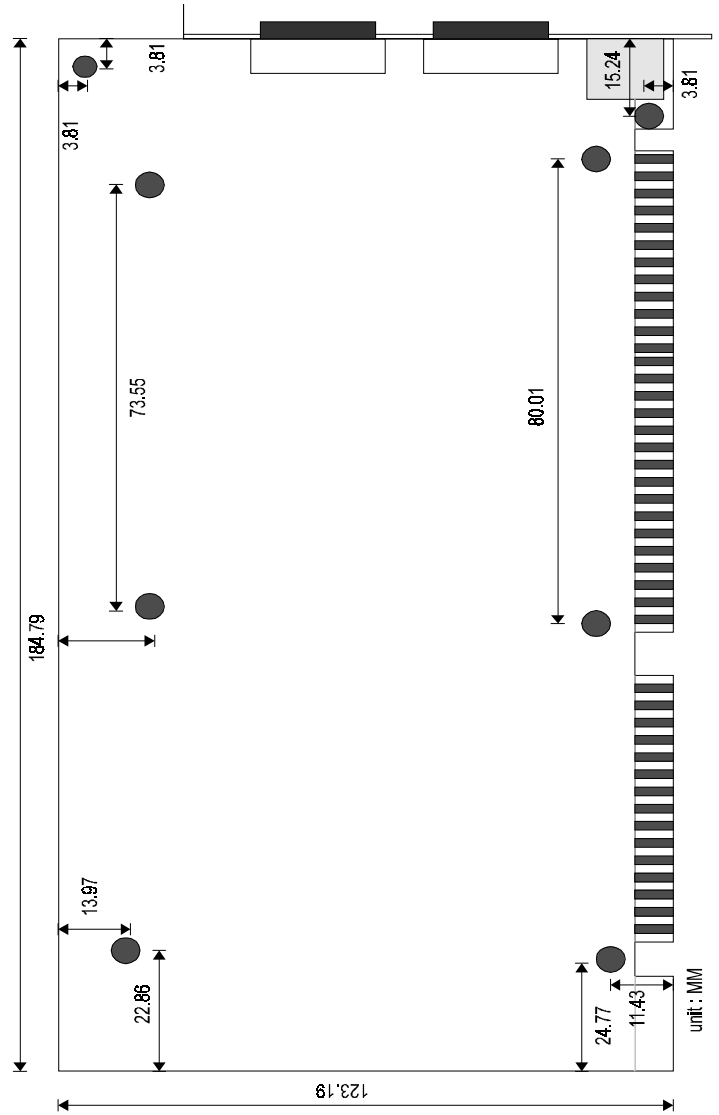
## 1.4 PCB layout








## 1.5 P.C.B dimension



# 2. Hardware description

## 2.1 Chipset

■ System Chipset :

The STPC86 chipset integrates a fully static x86 processor, fully compatible with standard x86 processors, and combine it with powerful chipset, graphics and video pipelines to provide a single consumer oriented PC compatible subsystem. The performance of this device is comparable with the performance of a typical P5 generation system.

The SST86 includes a powerful x86 processor core along with a 64-bit DRAM controller, advanced 64-bit accelerated graphics and video controller, a high speed PCI chip set functions

Graphics functions are controlled through the on-chip graphic engine in the ST PC Client and the monitor (or TV)display is produced through 2D graphics display engine. The graphics resolution supported is a maximum of 1280x1024 at 75 Hz refresh rate and is VGA and SVGA compatible

■ System memory

There are **On-board 4MB DRAM** (optional to 8MB) and two 64-bit memory banks on the board. Consists of two 72-pin SIMM sockets, which are designed to accommodate 2MB, 4MB, 8MB, 16MB, 32MB DRAM module providing the user with up to 64MB of paged main memory on the card.

■ Enhanced IDE features

The IDE specification has been updated to increase its capabilities and provided performance. Together these are referred to as "Enhanced IDE". The Enhanced IDE features comprise the followings:

1. Support for IDE hard disk drives larger than the former 528MB limit imposed by various technical factors.
2. Support for IDE devices other than hard disk drives, including IDE Tape backup and CD-ROM drives.
3. Support for faster data transfer rates, particularly with IDE controllers that have a VESA local bus interface.
4. BIOS setup will display all possible modes that include NORMAL, LBA & LARGE.





■ Solid state disk

1. On-board 1.7MB flash disk (PI-6486V only)

   PI-6486V provides on-board 1.7MB Flash disk that can be directly read/written under DOS prompt as regular floppy disk

2. DickOnChip

   In addition, the PI-6486V also provides two 32-pin sockets which can accept up to two DiskOnChip™ flash disks

3. SRAM disk

   Accept 512KB or 128KB SRAM devices

■ RS-232C serial Port

1. There are two full function UARTs which are full compatible with NS16C550 and has 16 bytes FIFOs Send/Receive registers. The data rates are independently programmable from 115.2K baud down to 50 baud.
2. Supported +5V(1A) Power Output or Ring Signal input By COM Port pin-9.(Jumper Selectable) for RS422/485 adapter

■ SVGA display

The graphics functions are controlled through the on-chip graphics engine in PCI Bus VGA controller(C&T65550) with On-board 1MB display memory, which supports TFT, STN, EL, Mono LCD and CRT. The graphics resolution supported is a maximum of 1024x768 256 colors and is VGA/SVGA compatible.

■ Watch dog timer

On-board watch dog timer provide 16 level from 1 to 160 seconds when the timer counts to preset value, and it sets a flag and optionally asserts NMI or RESET signal

■ Power Management

Four power saving modes: On, Doze, Standby, Suspend

# 3. Jumper settings on the PI-6486V

## 3.1 System memory setting: JP7

| JP7 | EDO-RAM | RAM |
|---|---|---|
| ON | Enabled | On-board EDO DRAM |
| OFF | Disabled | 72-pin DRAM on SIMM sockets |

**Note:**

1. Do not insert SIMM RAM modules onto the SIMM sockets when on-board EDO RAM enabled
2. If single 72-pin SIMM RAM is used, it should be inserted onto SIMM1 socket only

## 3.2 Battery backup for CMOS RAM and SRAM disk :JP3

| JP3 pin[1-2,2-3] | Function |
|---|---|
| 1-2 | Internal battery |
| 2-3 | Clear CMOS RAM |

1. The system should be turned off before Clear CMOS RAM
2. The SSD SRAM backup is not affected with this jumper setting





## 3.3 Watchdog setting : JP5

■ System signal setting:

The watchdog will generate a signal to system when it is not refreshed within a certain time. There are two way to assert a signal to system through RESET or MNI depended on the jumper setting shown as below:

| JP5 pin[1-2,3-4] | | |
|---|---|---|
| **Assert signal to** | **1-2** | **3-4** |
| RESET | ON | OFF |
| NMI | OFF | ON |
| Disabled | OFF | OFF |

■ watchdog timeout setting:

The watchdog timer is programmable in 16 level from 1 to 160 secs.

| Time out | JP5 pin[5-6,7-8,9-10,11-12,13-14,15-16] | | | | | |
|---|---|---|---|---|---|---|
| | 5-6 | 7-8 | 9-10 | 11-12 | 13-14 | 15-16 |
| 1 sec | on | off | off | off | off | off |
| 2 sec | on | off | off | off | off | on |
| 3 sec | on | off | off | off | on | off |
| 4 sec | on | off | off | off | on | on |
| 5 sec | on | off | off | on | off | off |
| 6 sec | on | off | off | on | off | on |
| 7 sec | on | off | off | on | on | off |
| 8 sec | on | off | off | on | on | on |
| 9 sec | on | off | on | off | off | off |
| 10 sec | on | off | on | off | off | on |
| 11 sec | on | off | on | off | on | off |
| 12 sec | on | off | on | off | on | on |
| 13 sec | on | off | on | on | off | off |

| | | | | | | |
|---|---|---|---|---|---|---|
| 14 sec | on | off | on | on | off | on |
| 15 sec | on | off | on | on | on | off |
| 16 sec | on | off | on | on | on | on |
| 10 sec | off | on | off | off | off | off |
| 20 sec | off | on | off | off | off | on |
| 30 sec | off | on | off | off | on | off |
| 40 sec | off | on | off | off | on | on |
| 50 sec | off | on | off | on | off | off |
| 60 sec | off | on | off | on | off | on |
| 70 sec | off | on | off | on | on | off |
| 80 sec | off | on | off | on | on | on |
| 90 sec | off | on | on | off | off | off |
| 100 sec | off | on | on | off | off | on |
| 110 sec | off | on | on | off | on | off |
| 120 sec | off | on | on | off | on | on |
| 130 sec | off | on | on | on | off | off |
| 140 sec | off | on | on | on | off | on |
| 150 sec | off | on | on | on | on | off |
| 160 sec | off | on | onf | on | on | on |

## 3.4 System memory occupied by Flash disk :JP8 pin[1-2]

The on-board flash disk occupies max. 24KB system memory area. The occupied memory addresses can be selected by jumper JP8 pin[1-2]

| System memory | JP3 pin 1-2 |
|---|---|
| D000:0000H-D5FF:FFFFH(24KB) | on |
| CA00:0000H-CFFF:FFFFH(24KB) | off |





## 3.5 Base port settings of on-board Flash disk:JP8 pin[3-4]

The on-board Flash disk occupies two I/O ports which can be set by jumper JP3 pin[3-4]

| I/O Base port | JP3 pin3-4 |
|---|---|
| 136H and 536H | on |
| 236H and 636H | off |

## 3.6 Configurations of on-board solid state disk

| JP8 | | | | | M0 on-board 1.7MB flash | M1 socket | M2 socket |
|---|---|---|---|---|---|---|---|
| 5-6 | 7-8 | 9-10 | 11-12 | 13-14 | | | |
| off | on | on | on | on | Enabled | SRAM | SRAM |
| off | off | on | off | on | Enabled | DiskOnChip | SRAM |
| off | off | off | off | off | Enabled | DiskOnChip | DiskOnChip |
| on | off | off | off | off | Disabled | DiskOnChip | DiskOnChip |

## 3.7 Setting RING or VCC of pin-9 of COM ports: JP6

The signal of pin-9 of each COM port can be set to “RING” or “Vcc” by jumper called “JP6”

| Pin 9 of COM port | JP6 | | | | Signal |
|---|---|---|---|---|---|
| | 2-4 | 4-6 | 1-3 | 3-5 | |
| COM 1,3(CN11) | on | off | | | Ring |
| | off | on | | | Vcc |
| COM 1,3(CN11) | | | on | off | Ring |
| | | | off | on | Vcc |

## 3.8 On-board TFT/LCD/CRT VGA BIOS setting: S1

VGA BIOS for eight most popular TFT/LCD display panels are pre-stored in on-board flash BIOS ROM. You can set switch called "S1" to select proper TFT/LCD panel BIOS

| S1 | | | | Panel type |
|---|---|---|---|---|
| Sw-1 | Sw-2 | Sw-3 | Sw-4 | |
| on | on | on | on | 1024 x 768 Dual scan STN color panel |
| on | on | on | off | 1280 x 1024 TFT color panel |
| on | on | off | on | 640 x 480 Dual scan STN color panel |
| on | on | off | off | 800 x 600 dual scan STN color panel |
| on | off | on | on | 640 x 480 12-bit TFT color panel |
| on | off | on | off | 640-480 18-bit TFT color panel |
| on | off | off | on | 1024 x 768 TFT color panel |
| on | off | off | off | 800 x 600 TFT color panel |

All modes shown above are also displayed on the CRT screen simultaneously

■ The SHF_CLK signal of Flat panel can be set "normal" or "inverted" mode

| JP4 | | SHF_CLK signal |
|---|---|---|
| 1-2 | 3-4 | |
| on | off | Inverted for EL panel |
| off | on | normal |





## 3.9 IRQ12 interrupt enable/disable setting:JP2

Because of "IRQ12" is appropriation for PS/2 mouse, the on-board chipset reserves "IRQ12" and has no pin-out to ISA bus. But in some application needs to use IRQ12 interrupt function. Here is a flexible way to generate a "**virtual IRQ12**" by redirecting IRQ12 on the ISA bus to IRQ15 by setting a jumper called JP2. When IRQ12 pin is triggered by ISA added-on card. The IRQ15 is interrupted and redirected to IRQ12 vector entry by TSR driver in the attached diskette

| JP2 | | Redirect IRQ12 to IRQ15 |
|---|---|---|
| 1-2 | 3-4 | |
| on | on | Enabled |
| off | off | Disabled |

**Note:**

1. When this functions is enabled. A TSR utility called "REDIRIRQ.EXE" should be added at first line of your "AUTOEXEC.BAT" file
2. When this function is enabled, the IRQ15 is functional disabled

# 4. Pin assignments of connectors

## 4.1 External power connector :CN6

| Pin No. | Signal |
|---|---|
| 1 | N.C |
| 2 | +5V DC |
| 3 | +12V DC |
| 4 | -12V DC |
| 5,6 | GND |

## 4.2 Keyboard connector :CN16 and CN13

| Pin No. | Signal |
|---|---|
| 1 | Clock |
| 2 | Data |
| 3 | NC |
| 4 | GND |
| 5 | +Vic |

## 4.3 Power L.E.D and Keyboard Lock :CN7 pin[6,8,10,12,14]

| Pin No. | Signal |
|---|---|
| 6 | Power L.E.D |
| 8 | N.C |
| 10 | GND |
| 12 | Key lock |
| 14 | GND |





## 4.4 Hardware reset connector :CN7 pin[1,2]

| Pin No. | signal |
|---|---|
| 1 | Reset input |
| 2 | GND |

## 4.5 CPU cooling fan power connector :CN10

| Pin No. | Signal |
|---|---|
| 1 | GND |
| 2 | +!2V DC |
| 3 | GND |

## 4.6 PS/2 mouse connector :CN12

| Pin No | Signal |
|---|---|
| 1 | MS_data |
| 2 | N.C |
| 3 | GND |
| 4 | +5Vdc |
| 5 | MS_clk |
| 6 | N.C |

## 4.7 External speaker connector :CN7 pin[7,9,11,13]

| Pin No. | Signal |
|---|---|
| 7 | +5V DC |
| 9,11,13 | Speaker out |

## 4.8 Hard Disk LED :CN7 pin[3,4]

| Pin No. | Signal |
|---|---|
| 3 | LED - |
| 4 | LED+ |

## 4.9 Hard disk connector :CN1

| Pin No. | Description | Pin No. | Description |
|---|---|---|---|
| 1 | -RST | 2 | Ground |
| 3 | D7 | 4 | D8 |
| 5 | D6 | 6 | D9 |
| 7 | D5 | 8 | D10 |
| 9 | D4 | 10 | D11 |
| 11 | D3 | 12 | D12 |
| 13 | D2 | 14 | D13 |
| 15 | D1 | 16 | D14 |
| 17 | D0 | 18 | D15 |
| 19 | GND | 20 | N.C |
| 21 | N.C | 22 | GND |
| 23 | IOW | 24 | GND |
| 25 | IOR | 26 | GND |
| 27 | IORDY | 28 | BALE |
| 29 | N.C. | 30 | GND |
| 31 | IRQ | 32 | IO16 |
| 33 | A1 | 34 | N.C. |
| 35 | A0 | 36 | A2 CS0 |
| 37 | CS0 | 38 | CS1 |
| 39 | -ACT | 40 | GND |





## 4.10Floppy disk connector :CN2

| Pin No. | Description | Pin No. | Description |
|---|---|---|---|
| 1 -33 (odd) | Ground | 2 | High density |
| 4,6 | Unused | 8 | Index |
| 10 | Motor enable A | 12 | Driver select B |
| 14 | Driver select A | 16 | Motor enable B |
| 18 | Direction | 20 | Step pulse |
| 22 | Write data | 24 | Wite enable |
| 26 | Track 0 | 28 | write protect |
| 30 | Read data | 32 | Select head |
| 34 | Disk change | | |

## 4.11Parallel port connector (CN3)

| Pin No. | Description | Pin No. | Description |
|---|---|---|---|
| 1 | -STROBE | 2 | DATA0 |
| 3 | DATA1 | 4 | DATA2 |
| 5 | DATA3 | 6 | DATA4 |
| 7 | DATA5 | 8 | DATA6 |
| 9 | DATA7 | 10 | ACK |
| 11 | BUSY | 12 | EMPTY |
| 13 | SELECT | 14 | Auto FEED |
| 15 | ERROR | 16 | INIT |
| 17 | INPUT | 18 | GND |
| 19 | GND | 20 | GND |
| 21 | GND | 22 | GND |
| 23 | GND | 24 | GND |
| 25 | GND | 26 | GND |

## 4.12COM port connectors: CN5 and CN11

■ COM[1,3] port (CN11)

| Pin No. | Signal |
|---|---|
| 1 | DCD |
| 2 | RX |
| 3 | TX |
| 4 | DTR |
| 5 | GND |
| 6 | DSR |
| 7 | RTS |
| 8 | CTS |
| 9 | RI (or Vcc)♦ |

**♦:ref to section 3.7**

■ COM[2,4] port (CN5)

| Pin No. | Signal |
|---|---|
| 1 | DCD |
| 2 | RX |
| 3 | TX |
| 4 | DTR |
| 5 | GND |
| 6 | DSR |
| 7 | RTS |
| 8 | CTS |
| 9 | RI (or Vcc) |

**♦:ref to section 3.7**





## 4.13 VGA CRT monitor connector: CN8

| pin No.(CN9) | signal |
|---|---|
| 1 | Red |
| 2 | Green |
| 3 | Blue |
| 4 | N.C |
| 5 | Ground |
| 6 | R/G/B return GND |
| 7 | R/G/B return GND |
| 8 | R/G/B return GND |
| 9 | N.C |
| 10 | Ground |
| 11 | N.C |
| 12 | Monitor ID bit |
| 13 | HSYNC |
| 14 | VSYNC |
| 15 | N.C |

## 4.14 Flat panel output connector: CN4

| Pin No. | Function | Pin No. | Function |
|---|---|---|---|
| 1 | GND | 2 | GND |
| 3 | SHF_CLK | 4 | M |
| 5 | HSYNC(LP) | 6 | VSYNC (FLM) |
| 7 | P0 | 8 | P1* |
| 9 | P2 | 10 | P3 |
| 11 | P4 | 12 | P5 |
| 13 | P6 | 14 | P7 |
| 15 | P8 | 16 | P9 |
| 17 | P10 | 18 | P11 |
| 19 | P12 | 20 | P13 |
| 21 | P14 | 22 | P15 |
| 23 | P16 | 24 | P17 |
| 25 | P18 | 26 | P19 |
| 27 | P20 | 28 | P21 |
| 29 | P22 | 30 | P23 |
| 31 | GND | 32 | GND |
| 33 | +5V | 34 | +5V |
| 35 | +5V | 36 | ENABKL* |
| 37 | ENAVEE | 38 | +12V |
| 39 | +12V | 40 | ENAVdd |
| 41 | N.C | 42 | GND |
| 43 | +3.3V | 44 | +3.3V |

The SHF_CLK signal of CN4 can be set to “normal” or “inverted” mode by setting jumper JP4 (see sec .3.8)

**“SHF_CLK”** should be set to “inverted” mode when EL type panel used





## 4.15 PC-104 ISA bus connectors :CN15,CN14

| CN12 | Name | CN12 | Name | CN12 | Name | CN12 | Name |
|---|---|---|---|---|---|---|---|
| A1 | -IOCHK | A17 | SA14 | B1 | GND | B17 | -DACK1 |
| A2 | SD7 | A18 | SA13 | B2 | Reset | B18 | DRQ1 |
| A3 | SD6 | A19 | SA12 | B3 | VCC | B19 | -REFSH |
| A4 | SD5 | A20 | SA11 | B4 | IRQ9 | B20 | SCLK |
| A5 | SD4 | A21 | SA10 | B5 | -5V | B21 | IRQ7 |
| A6 | SD3 | A22 | SA9 | B6 | DRQ2 | B22 | IRQ6 |
| A7 | SD2 | A23 | SA8 | B7 | -12V | B23 | IRQ5 |
| A8 | SD1 | A24 | SA7 | B8 | -WS0 | B24 | IRQ4 |
| A9 | SD0 | A25 | SA6 | B9 | +12V | B25 | IRQ3 |
| A10 | IORDY | A26 | SA5 | B10 | NC | B26 | -DACK2 |
| A11 | BAEN | A27 | SA4 | B11 | -SMWR | B27 | TC |
| A12 | SA19 | A28 | SA3 | B12 | -SMRD | B28 | BALE |
| A13 | SA18 | A29 | SA2 | B13 | -IOW | B29 | VCC |
| A14 | SA17 | A30 | SA1 | B14 | -IOR | B30 | OSC |
| A15 | SA16 | A31 | SA0 | B15 | -DACK3 | B31 | GND |
| A16 | SA15 | A32 | GND | B16 | -DRQ3 | B32 | GND |

| CN11 | Name | CN11 | Name | CN11 | Name | CN11 | Name |
|---|---|---|---|---|---|---|---|
| D1 | GND | D11 | -DACK5 | C1 | GND | C11 | -MWR |
| D2 | -MCS16 | D12 | DRQ5 | C2 | -SBHE | C12 | SD8 |
| D3 | -IOCS16 | D13 | -DACK6 | C3 | SA23 | C13 | SD9 |
| D4 | IRQ10 | D14 | DRQ6 | C4 | SA22 | C14 | S10 |
| D5 | IRQ11 | D15 | -DACK7 | C5 | SA21 | C15 | SD11 |
| D6 | IRQ12 | D16 | DRQ7 | C6 | SA20 | C16 | SD12 |
| D7 | IRQ13 | D17 | VCC | C7 | SA19 | C17 | SD13 |
| D8 | IRQ14 | D18 | MASTR | C8 | SA18 | C18 | SD14 |
| D9 | DACK0 | D19 | GND | C9 | SA17 | C19 | SD15 |
| D10 | DRQ0 | D20 | GND | C10 | -MRD | C20 | NC |

# 5. How to use Flash disk

## 5.1 About Flash disk

The PI-6486V provides on-board 1.7MB flash ROM disk and two disk sockets (M1,M2) that can accept 512KB/128KB low power SRAM, or DiskOnChip flash devices

There are three flash disk configurations of PI-6486V

On-board 1.7MB + SRAM on the M1,and M2 (PI-6486V only)

DiskOnChip flash only

On-board 1.7MB+ DiskOnChip flash (PI-6486V only)

## 5.2 Install on-board 1.7MB Flash disk + SRAMs on the M1& M2

1. Setting jumper JP3 properly (see sec. 3.4,3.5,3.6)
2. Inserts 128KB/512KB SRAMs into M1, M2 sockets, if necessary
3. Press "F10" at system POST to configure Flash Disk SETUP (see 5.5)
4. Boot system form normal CMOS setup sequence
5. Format on-board Flash disk and SRAM disks at DOS prompt as:

   *C>format <1.7MB Flash disk drive name> /S* (*Disk size=1.7MB)*

   *C>format <secondary SRAM disk drive name>* (*Disk size=M1+M2)*
6. Create "AUOTOEXEC.BAT" file in the flash disk
7. Copy "DISKSWAP.EXE" file into flash disk
8. Modifies "AUTOEXEC.BAT" file to have "DISKSWAP.EXE" at first line
9. Copy your application files into Flash disk by using "COPY" command
10. Press “F10” to enter Flash disk SETUP and change ‘System boot sequence item to "Flash disk" mode and press F7 to save
11. The system will boot DOS from Flash disk





## 5.3 Installing DiskOnChip only

The DiskOnChip™ Flash disk chip is produced by M-Systems. The DiskOnChip™ Flash disk occupies only 8KB system memory address, and can completely emulate a disk in PC system

1. Disable on-board 1.7MB Flash disk by setting jumper JP3 properly (see 3.4)
2. Insert DiskOnChip Flash device(s) into M1,or M2, or both
3. Boot system form normal CMOS setup sequence
4. Format DiskOnChip flash disk at DOS prompt as:

   *C>format D: /S* *(if one active hard disk existed )*

   Or

   C>format E: /S *(if two hard active disk existed )*
5. Copy your application files into DiskOnChip Flash disk by using "COPY" command
6. Reboot system and then modify System BIOS SETUP to have Hard Disk(s) not installed
7. The system will boot DOS from Flash disk

## 5.4 Installing on-board 1.7MB and diskOnChip simultaneously

1. Enable on-board 1.7MB flash disk by setting jumper JP8 properly (see3.4,3.5,3.6)
2. Insert DiskOnChip Flash device(s) into M1,or M2, or both
3. Press "F10" at system POST to configure Flash Disk SETUP (see 5.5)
4. Boot system form normal CMOS setup sequence and format on-board 1.7MB Flash disk at DOS prompt as: ***C>format <1.7MB disk drive name> /S***
1. Create "AUOTOEXEC.BAT" file in the flash disk ,and copy "DISKSWAP.EXE" file into flash disk
2. Modifies "AUTOEXEC.BAT" file to have "DISKSWAP.EXE" at first line
3. Copy your application files into Flash disk by using "COPY" command
4. Format DiskOnChip Flash disk at DOS prompt as:

   *C>format <DiskOnChip drive name> /S*
5. Reboot system and then modify the Flash disk setup to set "Boot Sequence" to "Flash disk" to boot up from Flash disk or to "CMOS setup" to boot up from DiskOnChip

## 5.5 On-board Flash disk setup

This ***SETUP*** is used to configure th***e*** On-board Flash disk. As AWARD/AMIBIOS POST executes, the following appears:

STPC ,xxMHz

Checkinh NVRAM.. Update OK!

..WAIT.....

…..

Hit <F10> to Run Advanced Flash disk SETUP

At this time, Hit <F10> to run Solid state disk SETUP. A SETUP window shows up on the screen.

.

At this time, Hit <F10> to run Solid state disk SETUP. A SETUP window shows up on the screen.

.

| STPC Client 5x86 CPU Card ROM/RAM Disk SetUp Vx.x | | |
|---|---|---|
| Primary drive of Flash Disk | :A | |
| Device(s) Configuration | :Primary=M0<br>:Secondary=M1+M2 | Flash<br>SRAM |
| System boot sequence | :Flash disk | |
| COM port for remote Flash PGM | :COM1 | |
| Auto detect & Boot from A: | :Enabled | |
| Function call Entry (INT83H) | :Disabled | |
| Up/Down:Select PgUp/PgDn:Select | Esc:No save | F7:Save |

F8 :Load Default Setup





Where

- Primary drive of Flash disk

  The on-board 1.7MB flash disk can emulate disk drive A:,B:,C:,D: or disabled.

  A=Drive A:

  B=Drive B:

  C=Drive C:

  D=Drive D:

  Disable: Disabled

- Device(s) Configuration

  Primary =M0 FLASH

  Secondary =M1+M2 SRAM

- System Boot Up Sequence

  The system may boot up from two following sequences as:

  CMOS SETUP=Boot up according to CMOS Setup sequence

  Flash disk =Boot up from on-board Flash disk, if DOS files existed in Flash

- COM Port for Remote Flash PGM

  If you are going to remotely access the flash disk with utility called "RDISK.EXE" . You should select which communication port is connected to communicate with host system.

  COM1=COM1 port (3F8H) IRQ4

  COM2=COM2 port (2F8H) IRQ3

  Disable=No remote Access required

- Auto detect and boot from Disk A:

  Enable= The system will automatically boot form disk A: If system files (IO. SYS, SMDOS.SYS) existed

  Disable=Normal boot sequence

- Function call Entry (INT83H)

  INT83H provides calls from which you can directly access on-board flash or SRAM device(s) without passing through DOS functions

  Enable=Enables INT83 function calls

  Disable= Disables INT83 function calls

- Selects and save SETUP

  PgDn=Moves reverse bar down

  Esc=Quit without saving SETUP

  F7=Quit and saves current SETUP to EEPROM

  F8=Load default setup values

## 5.6 Flash disk installation examples

- Example 1: Assume to use on board 1.7MB Flash disk as primary disk drive “A” and 128KB SRAM on the as secondary disk drive “B”

  Step 1. Press “F10” at BIOS POST to run Flash disk setup

  Step 2. Set “Primary ROM/RAM disk drive :A”,
  “Device(s) Configuration :Primary=M0 Flash”
  :Secondary=M1+M2 SRAM”

| STPC Client 5x86 CPU Card ROM/RAM Disk SetUp Vx.x | |
|---|---|
| Primary drive of Flash Disk | :A |
| Device(s) Configuration | :Primary=M0 Flash<br>:Secondary=M1+M2 SRAM |
| System boot sequence | :Flash disk |
| COM port for remote Flash PGM | :COM1 |
| Auto detect & Boot from A: | :Enabled |
| Function call Entry (INT83H) | :Disabled |
| Up/Down:Select PgUp/PgDn:Select | Esc:No save F7:Save |

  Step 3. Set “Boot sequence” to CMOS setup and press “F7” to save setup value and exit

  Step 4. Format B:/S (Primary disk drive) at DOS prompt

  Step 5. The total capacity of disk A: is 1700KB and B: is 128KB

  Step 6. Creat “Autoexec.bat” file in disk B:, and modify the first line in “autoexec.bat” file to have “@ Diskswap.exe” command

  Step 7. Copy “**DISKSWAP.EXE**” file to disk B: from provided utility diskette

  Step 8. reboot your system and press “F10” again and modify
  “System Boot Up Sequence :Flash disk”

**Note:** The file called "**Diskswap.exe**" in the provided diskette is used to swap the disk drive name when system boots from Flash disk





# 6. On board Flash/RAM disk BIOS function calls

The on-board Flash/RAM disk BIOS provides you the flexible functions that you can directly access the on-board memory device(s) without using file handler of DOS.

## 6.1 Function Call Entry:INT 83H

**Entry:** AH=1 ;Get Device type and size
DL=Device No.
=0 for on-board Flash
=1 for “M1” socket
=2 for “M2” socket

**Exit:** AH =Device Type =10H Flash/SRAM
=20H EPROM

CX =size (KB)

**Entry:** AH=2 ;Read Sectors From Memory
DL=Device No. ;device
CX=Sector No.
AL=Sectors

**Exit:** ES:BX= Destination buffer pointer

**Entry:** AH=3 ;Writes One Sector to memory
;device

DL=Device No. ;(0,1,2)

CX=Sector No. ;(0,1,2,……)

AL=Sectors. ;(1,2,……..)

**Exit:** ES:BX= Source buffer pointer

Carry =1 ;data writing error

Carry =0 ;completed

| | | |
|---|---|---|
| **Entry** | AH =4<br>BX=EEPROM address | ;Reads One Byte From EEPROM<br>;(0-499) |
| **Exit** | AL=Data be read out | |
| **Entry:** | AH=5<br>AL=Data be written<br>BX=EEPROM addr. | ;Writes One Byte to EEPROM |
| **Exit:** | Carry=1<br>Carry=0 | ;Data writing error<br>;completed |

## 6.2 Example written with C language:

```
#include "io.h"
#include "stdlib.h"
#include "conio.h"
#include "dos.h"
char sector_buffer[512];
main()
{
clrscr();
  if (check_funcall()==-1)              /* Check existence of Function call Entry */
     {
        printf ("Flash/RAM disk Function call not existed \n\r ");
        exit(0);
     }
  printf ("Flash/RAM disk Function call ready..\n\r ");
  read_devsec(2,0,1,sector_buffer);
  write_devsec(2,0,1,sector_buffer);
}
```





```
/************************************************/
/*        Function Calls of Flash/RAM disk    */
/************************************************/
int check_funcall()
/*    Check existence of Flash/RAM disk function call entry */

{
unsigned far *Int83_Ptr=0x0000020c; /* Points to INT 83H vector */
unsigned far *idptr;
idptr=MK_FP(*(Int83_Ptr),0);          /* Get Function call Parameter */
  if (*(idptr)!=0x5678)               /* Check Function call ID code */
     return(-1);                      /* Function call entry not existed */
  else
     return(1);                       /* Function call entry available */
}
read_devsec(int device_no,int sector_no,int sectors,char *buffer_ptr)
/* Directly read sectors from Flash/RAM disk function call entry
  where
  Entry  int device_no :device No. (0.....)
         int sector_no :start sector No. (0....)
         int sectors    :sectors be read    (1....)
         int buffer_ptr :data buffer pointer
  Exit   Datas in buffer_ptr                          */

{
  union REGS xr;
  struct SREGS sr;

  xr.h.ah=2;                          /* Function call no in AH */
  xr.h.dl=device_no&0xff;             /* Device No. in DL */
  xr.x.cx=sector_no;                  /* Sector No. in CX */
  xr.h.al=sectors;                    /* Sectors in AL */
```

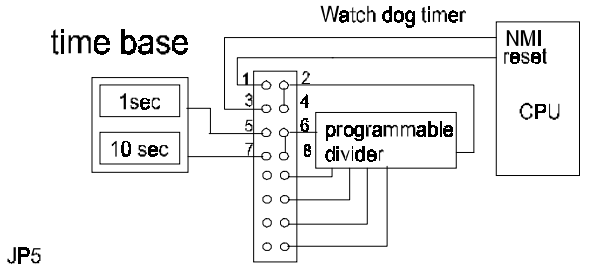
```
  xr.x.bx=FP_OFF(buffer_ptr);          /* Offset of Buffer pointer in BX */
  sr.es=FP_SEG(buffer_ptr);            /* Segment of Buffer pointer in ES */
  int86x(0x83,&xr,&xr,&sr);            /* Call function */

}

write_devsec(int device_no,int sector_no,int sectors,char *buffer_ptr)

/* Directly write sectors of device through function call entry

where
Entry   int device_no :device No. (0.....)
        int sector_no :start sector No. (0....)
        int sectors   :sectors be written   (1....)
        int buffer_ptr :data buffer pointer
Exit    None            */

{
  union REGS xr;
  struct SREGS sr;
  xr.h.ah=3;                           /* Function call no in AH */
  xr.h.dl=device_no&0xff;              /* Device No. in DL */
  xr.x.cx=sector_no;                   /* Sector No. in CX */
  xr.h.al=sectors;                     /* Sectors in AL */
  xr.x.bx=FP_OFF(buffer_ptr);          /* Offset of Buffer pointer in BX */
  sr.es=FP_SEG(buffer_ptr);            /* Segment of Buffer pointer in ES */
  int86x(0x83,&xr,&xr,&sr);            /* Call function */
}       End
```





# 7. How to use watch dog timer

The PI-6486V is equipped with a watch-dog timer that resets the system if processing comes to a standstill, Typically caused by electromagnetic interference (EMI) or software bugs. This feature ensures continued operation in industrial stand-alone and unmanned applications. The watchdog timer can be programmed in 16 level form 1 to 160 seconds. The “RESET” or “MNI” signal will be generated from watchdog timer, When timer out reaches




## 7.1 Watch dog timer control port

Ths on-board watchdog timer occupies two read/only input ports 443H and 43H

| Control port | Function |
|---|---|
| 443H | Enable watchdog and start to count |
| 43H | Disable watchdog |

## 7.2 Example program for watchdog timer

■ Example program written with assembly language (WDDEMO.ASM in diskette)

Assume set watchdog timer to 12 sec

```
;Enable watchdog timer and start to count
mov     dx, 443H        ;Points to watch dog timer Enable port
In      dx,al           ;enable and start to count
;user's application section
mov     dx,443H         ;Points to watch Enable port again
in      dx,al           ;enable and restart to count
                        ;this instruction should be recycled
                        within 12 sec in user's application sec.
;user's application section
;……
; disable watch dog timer
mov     dx, 43H         ;Points to watch dog disable port
In      dx,al           ;disable watchdog timer
ret                     ;exit
```





■ Example program written with C language (WDDEMO.C in diskette)

Assume set Time base to 12 sec

```
#include <dos.h>
#include <time.h>
#include <stdio.h>
/* ----------Example program-1 of watch dog timer ----------*/
void main ( )
 {
 Long Start_time,Cur_time;
 Int time_value,key=0 , time_base=1;
 Int signal=RESET;
 printf ("\n\r======= PI-6486 Watch Dog Timer Demostration ==========="
 inportb(0x443);                          /* enable and start to count */
 time (&Start_time);                      /* Get start time */
 printf ("\n\r Press any key to trigger or [ESC] to terminate>\n\r\n\r" );
 while (key!=0x1b)
{
while (kbhit()==0)
 {
 time (&Cur_time);                          /* Get current time */
 printf ("\r Remainded time :%3d sec. ",time_value-(int)(Cur_time-Start_time));
 }
 if ((key=getch())!=0x1b)
 {
Inportb(0x443);                           /* restart to count once*/
time (&Start_time);                       /* Get start time */
 } }
 inportb(0x43; }                           /* Disable Watch dog timer */
```

# 8. VGA driver installation

This chapter provides information on how to install VGA drivers that come in the floppy diskette with your PI-6486V card.

## 8.1 Contents in this diskette

- Standard Windows 95/98 driver :

  CHIPS95.DRV

  CHIPS95.VXD

  CHIP95.INF

## 8.2 How to install this driver in Windows 95/98

This section describes normal display driver installation procedures for Windows 95/98. Use these procedures when installing the display drivers provided in attached diskette.

- driver installation procedure

  1. Click Start, then Settings, then Control Panel.
  2. Start the "Display" applet program.
  3. Select the "Settings" page, push the "Change Display Type" button.
  4. Push the "Change" button in the "Adapter Type" area.
  5. Push the "Have Disk" button and press "OK".
  6. Specify the path to the new driver and press the <ENTER> key:

- Examples

  1. Insert the drivers disk in the A: floppy drive, and enter A:\.
  2. Type in the name of the directory where you copied the drivers, either on your local hard drive or on a network share.
  3. If you're not sure exactly where the drivers are, choose the "Browse..." button are find them.
  4. The "Select Device" dialog box will appear. Select the adapter that corresponds to the one you installed in your machine and click OK.
  5. Windows will copy the display drivers to the proper directories on your system.
  6. Continue choosing Close until asked to restart your machine from the "Systems Settings Change" dialog box.





7. After the system has restarted, you can go back into the Display applet and select alternate screen resolutions and color depths.

## 8.3 How to install this driver in Windows NT 4.0

This section describes normal display driver installation procedures for Windows NT 4.0 these procedures when installing the display drivers provided in attached diskette.

1. Install Windows NT as you normally would for a VGA display. First click the Start button, go to Settings and click on Control Panel. Choose the Display icon and click on the icon. In the Display Properties window, click on the Settings tab. Then click on Change Display Type. In the Change Display Type window, click on the Change button under Adapter Type. This will bring up the Select Device window.
2. In the Select Device window, click on the Other button. Enter source directory where the Windows NT driver files are located. Press <ENTER> and the name of the Chips and Technologies Video Accelerator driver will appear at the end of Models list box. Scroll to the end of the list box and double click on the driver. Once the installation is complete, the system must be shut down and restarted.
3. Upon restart, select the desired display settings from the Display property dialog box. Click on Test to test the newly selected graphics mode. A color test screen should appear, followed by the Testing Mode window. Click on Yes to continue. The Display Settings Change window will appear. Click on Restart Now for
4. the new settings to take effect.

# 9. CMOS setup

Once you enter the Award BIOS CMOS Setup Utility, the Main Menu will appear on the screen. The Main Menu allows you to select from several setup functions and two exit choices. Use the arrow keys to select among the items and press <Enter> to accept and enter the sub-menu.

| | |
|---|---|
| Standard CMOS Features<br>Advanced BIOS Features<br>Advanced Chipset Features<br>Integrated Peripherals<br>Power Management Setup<br>PNP/PCI Configuration | LOAD Fail-Safe Defaults<br>LOAD optimized Defaults<br>Set Password<br>Save & Exit Setup<br>Exit Without Saving |
| ESC: Quit<br>F10:Save & Exit | ↑ ↓ → ←:Select item |
| Time,Date,Hard disk Type | |

- Standard CMOS Features

  This setup page includes all the items in a standard, AT-compatible BIOS

- Advanced BIOS Features

  This setup page includes all the items of Award special enhanced features

- Advanced Chipset Features

  This setup page includes all the items of chipset special features.

- Integrated Peripherals

  This section page includes all the items of IDE hard drive and Programmed Input / Output features.".

- Power Management Setup

  This entry only appears if your system supports Power Managemen

- PNP / PCI Configuration Setup.

  This entry appears if your system supports PNP / PCI.





- LOAD Fail-Safe Defaults

  The BIOS defaults have been set by the manufacturer and represent settings which provide the minimum requirements for your system to operate.

- LOAD optimized Defaults

  The chipset defaults are settings which provide for maximum system performance. While Award has designed the custom BIOS to maximize performance, the manufacturer has the right to change these defaults to meet their needs.

- Set Password

  Change, set, or disable password. It allows you to limit access to the system and Setup, or just to Setup

- Save & Exit Setup

  Save CMOS value changes to CMOS and exit setup.

- Exit Without Save

  Abandon all CMOS value changes and exit setup.

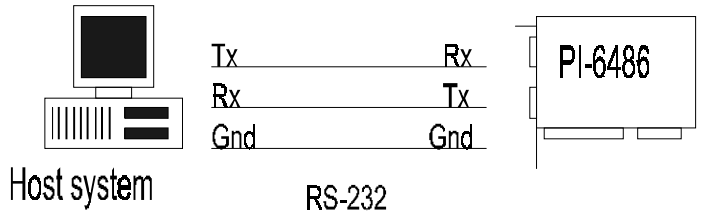
# 10. About RDISK.EXE utility

The RDISK.EXE program is a comprehensive driver which can help you to directly access the Flash disk in the remote system through RS-232 or RS-422 serial ports.

When RDISK.EXE be executed, it will be keep resident in host memory and redirects DOS disk I/O to remote system. The flash disk(s) in the remote system is logged into the host system as local disk drive(s) A: or B:, and all DOS commands are available for remote disk(s) such as COPY, DEL, FORMAT, MD....etc

The following procedures show you how to install RDISK.EXE program

1. Enter your host computer system BIOS SETUP menu and configure your host Floppy disks to have following configuration
2. Floppy disk A: 1.44MB or 1.2MB
3. Floppy disk B: 1.44MB or 1.2MB
4. Reboot your host system
5. Connects a RS-232 cable between the HOST and PI-6486V




6. Turn-on your remote system
7. Press "F10" key to run ROM/RAM disk SETUP. and
8. Set "Primary ROM/RAM disk " to A:,B:,C:,or D:
9. Enable "COM Port for Remote Flash PGM:" to COM1 or COM2
10. Press "F7" to save SETUP values
11. Type RDISK <Enter> at DOS prompt of host system
12. The following message shows up on the screen





```
==========   RDISK configurations   ==============
Primary Disk    = A:(local)
Secondary Disk= No used
COM port          = COM1
Baud Rate         = 57600
================================================
Reboot or turn-on remote system
      Type Rdisk /? for Helps
      Press [ESC] to abort
      Connecting (|) _
```

At this time, the RDISK are attempting link to remote system

Reboot your remote system

"Connecting (-) ....." message will show up after connecting completed

RDISK now returns to DOS and remote solid state disk(s) are ready to be accessed

Format remote disk(s): ( if flash disk in the remote system is unformatted)

Type one of the following commands to format remote solid state disk(s) which are installed at first time

```
C>FORMAT [remote disk drive] /T:16/N:32     for 512KB Flash disk
C>FORMAT [remote disk drive] /T:32/N:32     for 1.0MB Flash disk
C>FORMAT [remote disk drive] /T:48/N:32     for 1.5MB Flash disk
C>FORMAT [remote disk drive] /T:53/N:32     for 1.7MB Flash disk
C>FORMAT [remote disk drive] /T:69/N:32     for 1.5MB+512KB Flash disk
```

The disk(s) in the remote are already to be accessed by the host system

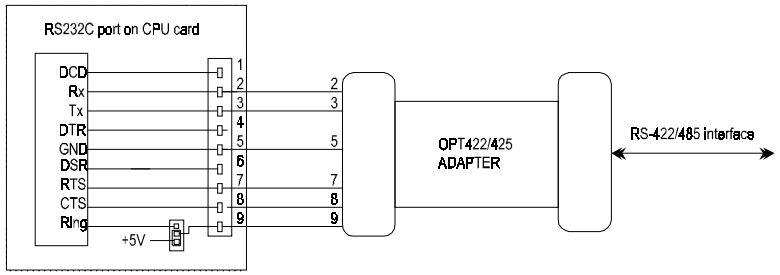
Type " RDISK /R " at DOS prompt can release this TSR driver from host system

Type " RDISK /? " at DOS prompt will show more information about RDISK

# 11. Converting RS-232 serial port(s) to RS-422/485

The RS-232C serial port(s) on this card can easy be converted to RS-422/485 interface by using OPT422/485 adapter

The OPT425 can convert four RS-232C level signals into RS-422 level( Tx,Rx,RTS,CST), and supplied by +5V through pin-9 of RS-232C connector as shown below.




Connection between RS-232C port and OPT422/425

See sec. 3.7 to set jumper called JP6 properly before using OPT425 adapter





# 12. Appendix A: Troubleshooting

Fatal errors are those which will not allow the system to continue the boot-up procedure.

These fatal errors are usually communicated through a series of audible beeps

| | | |
|---|---|---|
| 1 Beep | :Refresh Failure | The memory refresh circuitry of the motherboard is faulty |
| 2 beeps | :Parity Error | A parity error was detected in the base memory (the first block of 64KB) of the system |
| 3 beeps | :Base 64KB Memory Failure | A memory failure occurred within the first 64KB of memory |
| 4 beeps | :Timer Not Operational | Timer #1 on the system board has failed to function properly with the boot up procedure. |
| 5 beeps | :Processor Error | The CPU (Central Processing Unit) on the system board has generated an error |
| 6 beeps | :8042 Gate A20 Failure | This error message means that the BIOS is not able to switch the CPU into protected mode |
| 7 beeps | :Processor Interrupt Error | The CPU on the motherboard has generated an exception interrupt |
| 8 beeps | :Display Memory Error | The system video adapter has generated an error or its memory is faulty |
| 9 beeps | :ROM Checksum Error | The ROM checksum value does not match the value encoded in the |
| 10 beeps | :CMOS register error | The shutdown register for the CMOS memory has failed |

# 13. Appendix B : I/O Port Address Map

Each peripheral device in the system is assi-ned a set of 1/0 port addresses which also becomes the identity of the device. There are a total of 1 K port address space available. The following table lists the 1/0 port addresses used on the Industrial CPU Card.

| Address | Device Description |
|---|---|
| 000h - OlFh | DMA Controller #1 |
| 020h - 03Fh l | Interrput controller #l |
| 040h - 05Fh | Timer |
| 060h - 06Fh | Keyboard controller |
| 070h - 07Fh | Real Time Clock. NMI |
| 080h - 09Fh | DMA Page Register |
| OAOh - OBFh | Interrupt Controller #2 |
| OCOh – ODFh | DMA Controller #2 |
| OFOh | Clear Math Coprocessor Busy Signal |
| OF1h | Reset Math Coprocessor |
| 1F0h - IF7h | IDE Interface |
| 278h - 27Fh | Parallel Port #2(LPT2) |
| 2F8h - 2FFh | Serial port #2 (COM2) |
| 2B0h - 2DFh | Graphics adapter Controller |
| 378h - 3FFb | Parallel Port # I (LPT 1) |
| 360h - 36Fh | Network Ports |
| 3B0h - 3BFh | Monochrome & Printer adapter |
| 3C0h - 3CFh | EGA adapter |
| 3D0h- 3DFh | CGA adapter |
| 3F0h - 3F7h | Floppy Disk Controller |
| 3F8h - 3FFb | serial Port# I (COM I) |