Environmental Compatibility Monitor Program of Embedded and Industrial Computer

User Manual

(Ver. 1.0)

EVOC Software Technology Co., Ltd.

Preface

This program is designed for the real-time monitor of environmental parameters of EVOC embedded and industrial computers. Via the self-developed bottom hardware interface, it can monitor the real-time voltage of CPU, battery and power supply, fan speed and other parameters, as well as giving over-limit alert.

Content

Chapter 1 Hardware Requirements and Operating Environment 4

Hardware Requirements 4

Operating Environment..4

Chapter 2 Install and Uninstall 4

Install..4

Uninstall..4

Chapter 4 Parameters Setup 6

Power Supply Voltage Setup 6

Motherboard Voltage Setup 7

Temperature Setup..7

Other Setup..8

Color Setup 9

Chapter 5 Running of the Program 11

Real-time Monitor..11

Real-time Curve 13

History Curve..13

Hide 14

Monitor On/off 14

Sound Alert..14

Exit the System..14

Chapter 6 Hardware Interface Function Introduction (EvocHM.dll) 15

Chapter 7 Programming Introduction 19

Chapter 1 Hardware Requirements and Operating Environment

Hardware Requirements

The software is designed for EVOC embedded and industrial computers.

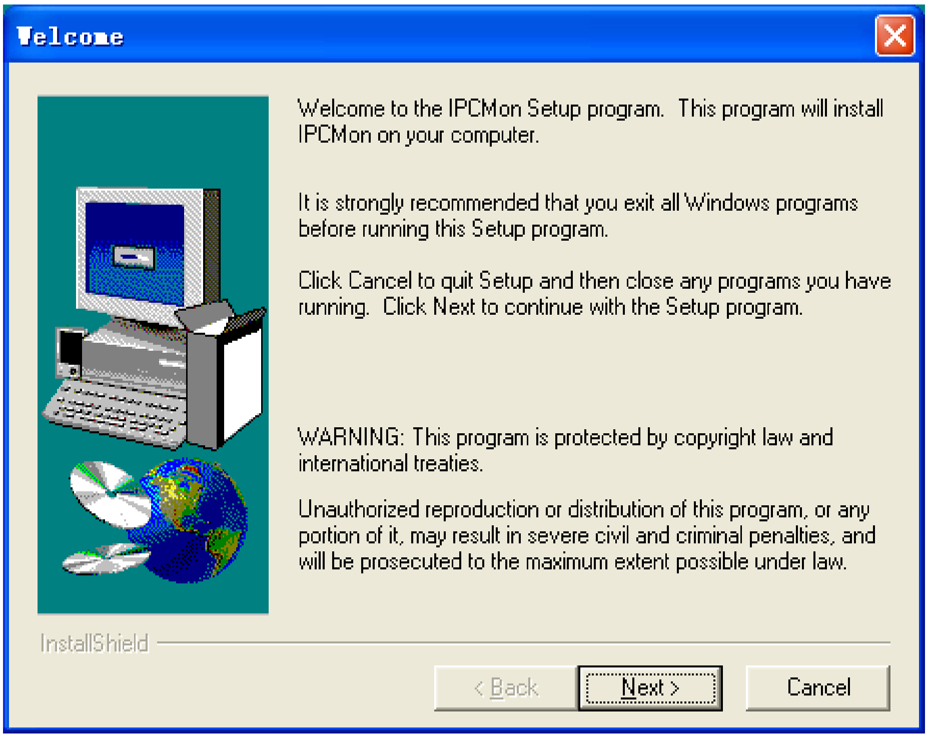
Operating Environment

The software is compatible with Win9X/Win2000/WinXP platforms.

Recommended resolution: 800 x 600.

Chapter 2 Install and Uninstall

Install



Welcome to the IPCMon Setup program. This program will install IPCMon on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

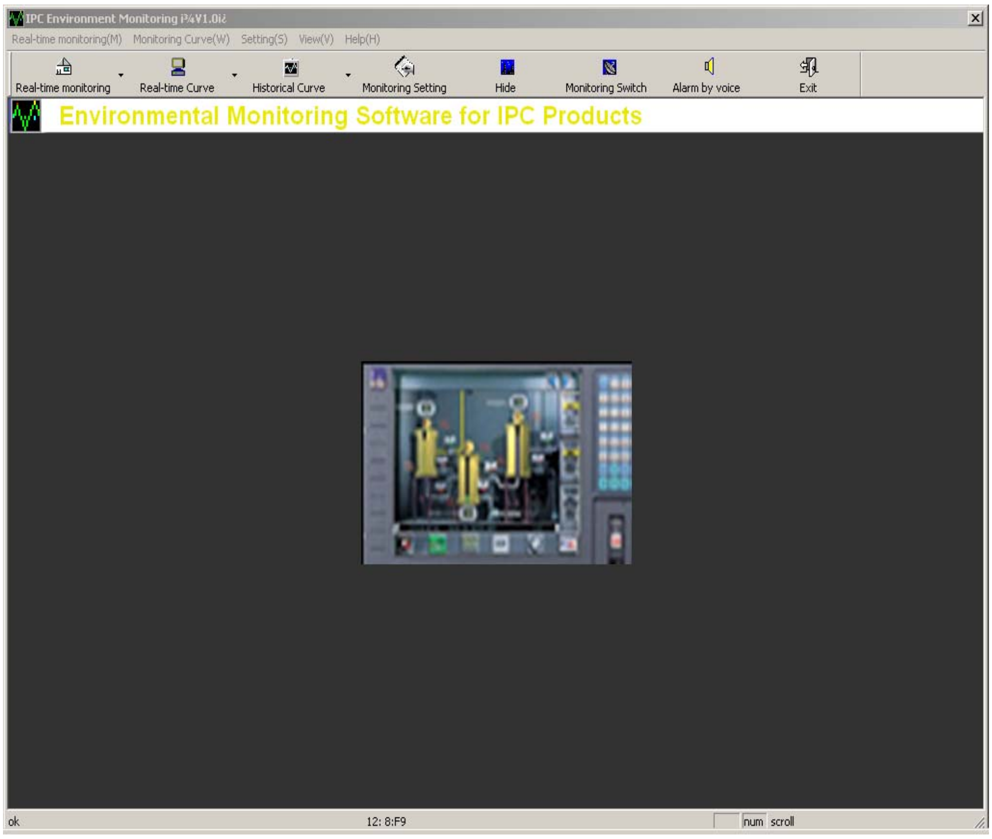
Run the Setup program to start installing, and then select the installation directory and follow the guide to finish the installation, finally restart the computer.

Uninstall

Find *Add or Remove Programs* in *Control Panel*, and select *Environmental Compatibility Monitor Program of Embedded and Industrial Computer* program, click *Add or Remove* button to eject remove confirm box, click *Yes* to uninstall the program and then click *OK* to finish the uninstalling.

Chapter 3 Software Function Menu

Start the *Environmental Compatibility Monitor of Embedded and Industrial Computer* program, and then you will see the following interface:



The function menus are on the top of the interface, users can click each menu to eject pull-down menu and then operate the corresponding function. The function menus include Real-time Monitor, Monitor Curve, Setup, View and Help.

Real-time Monitor includes:

Monitor On/Off, Power Supply Voltage Monitor, Motherboard Voltage Monitor, Exit.

Monitor Curve includes:

Real-time curve and History curve; real-time curve includes power supply voltage curve, motherboard voltage curve, temperature monitoring curve, fan speed voltage; history curve includes power supply voltage curve, motherboard voltage curve, temperature monitoring curve, fan speed curve.

Setup includes:

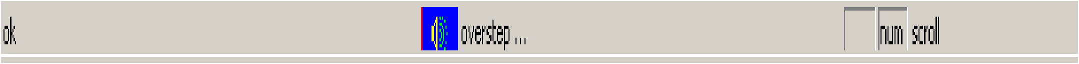
Alert setup, color setup, sound alert, start the cooling measures.

View includes: Tool bar, status bar and hide.

Help includes: About and EVOC website

The second raw is Tool Bar, including: Real-time Monitor, Real-time Curve, History Curve, Monitoring Setup, Hide, Monitor On/off, Sound Alert and Exit.

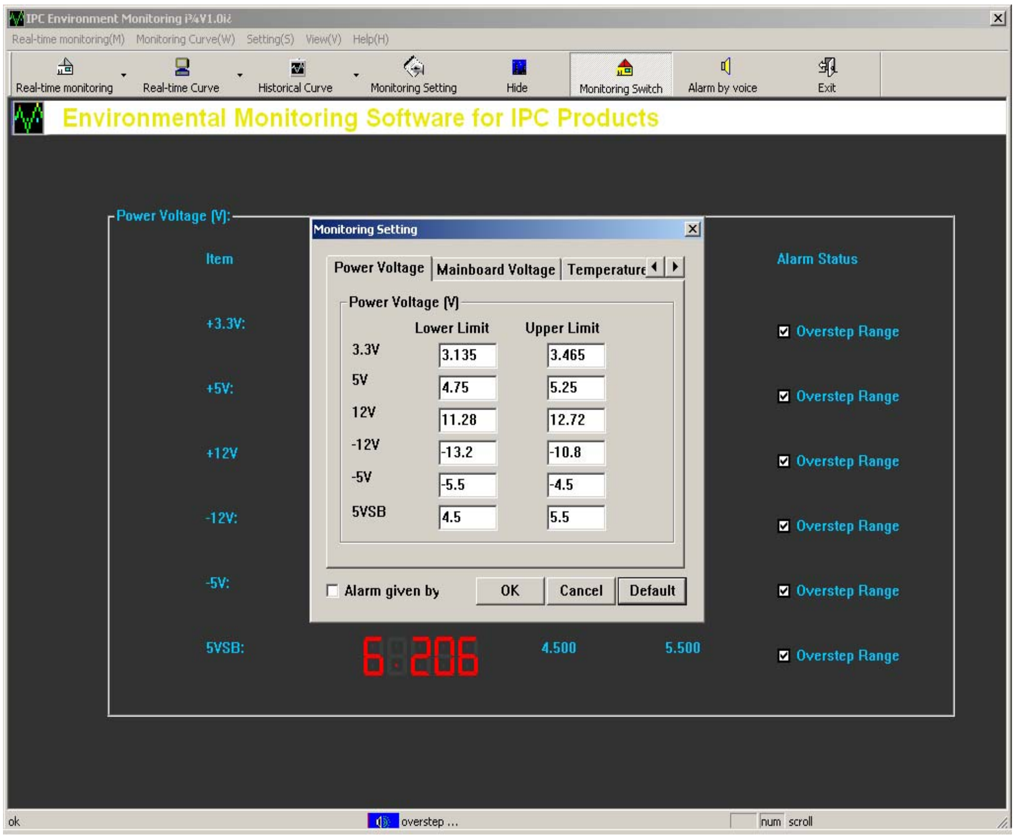
In the bottom is the Status Bar: Indicates the monitoring status.



Chapter 4 Parameters Setup

Power Supply Voltage Setup

The power supply voltage alerting limit can be set here.



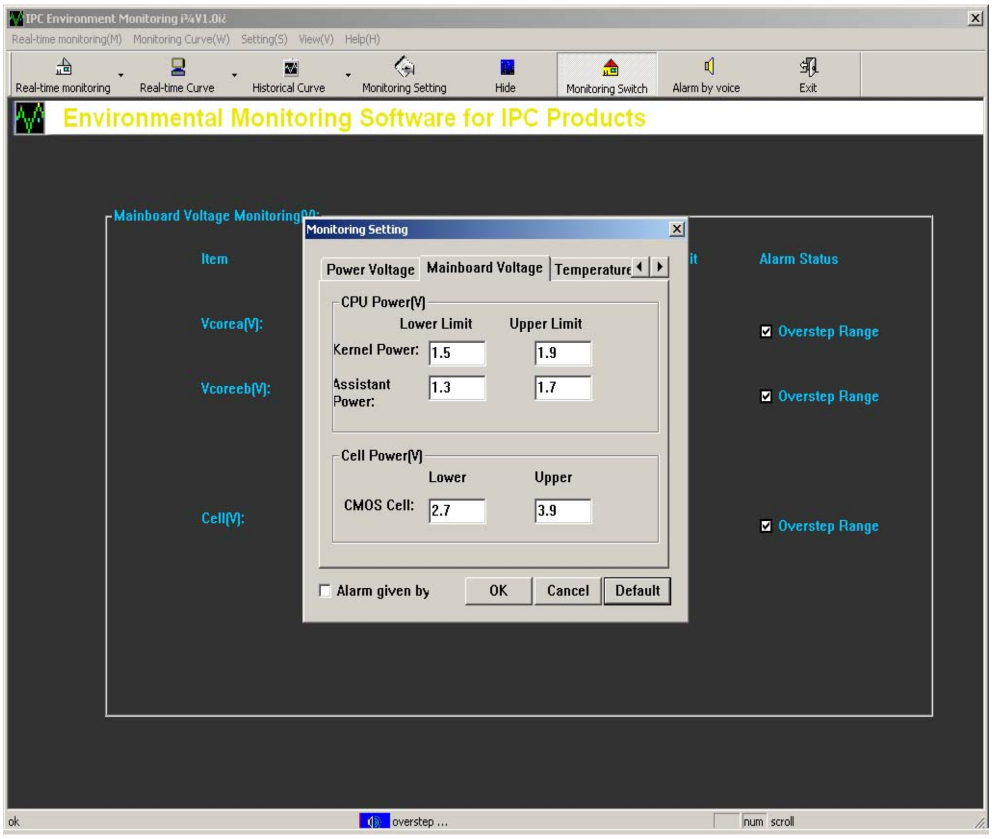
Power Voltage (V)

	Lower Limit	Upper Limit
3.3V	3.135	3.465
5V	4.75	5.25
12V	11.28	12.72
-12V	-13.2	-10.8
-5V	-5.5	-4.5
5VSB	4.5	5.5

After the setup, click *Yes* to save the configuration, click *Cancel* to cancel the configuration, click *Default* to restore the parameters to default status.

Motherboard Voltage Setup

The motherboard voltage alerting limit can be set here.



CPU Power(V)

	Lower Limit	Upper Limit
Kernel Power:	1.5	1.9
Assistant Power:	1.3	1.7

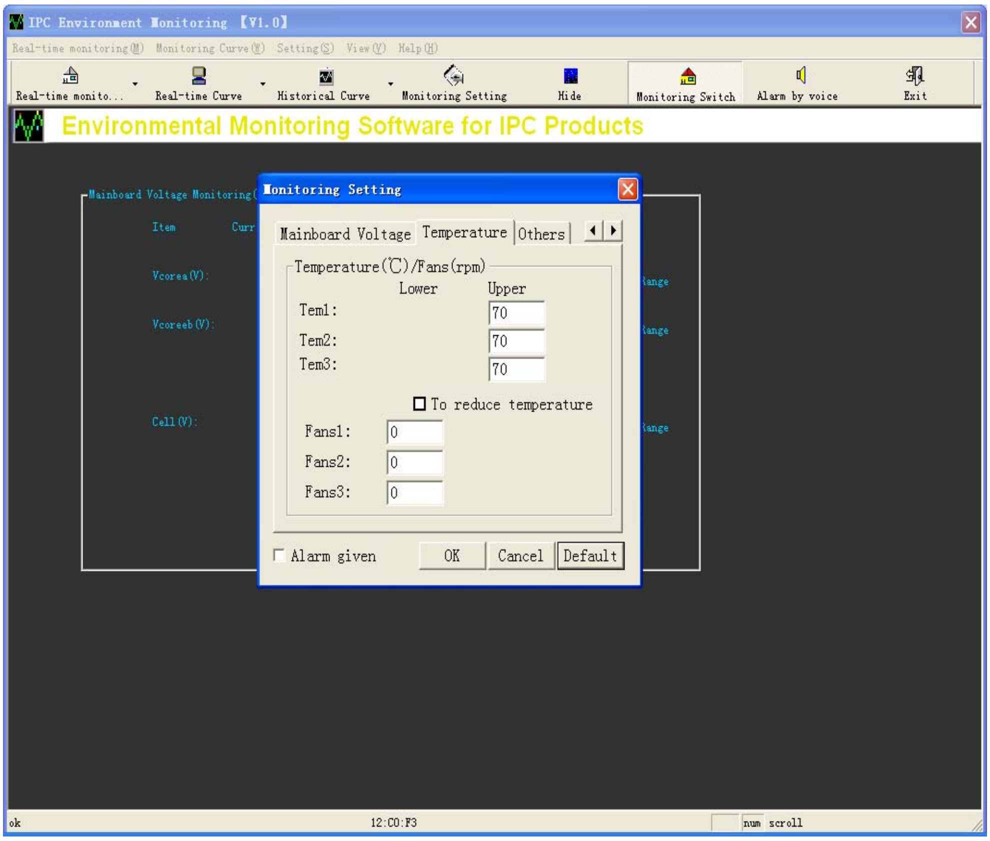
Cell Power(V)

	Lower	Upper
CMOS Cell:	2.7	3.9

After the setup, click *Yes* to save the configuration, click *Cancel* to cancel the configuration, click *Default* to restore the parameters to default status.

Temperature Setup

The upper limit of CPU temperature and system temperature and the lower limit of CPU fan speed can be set here.



Monitoring Setting

Temperature(℃)/Fans(rpm)

	Lower	Upper
Tem1:		70
Tem2:		70
Tem3:		70

□ To reduce temperature

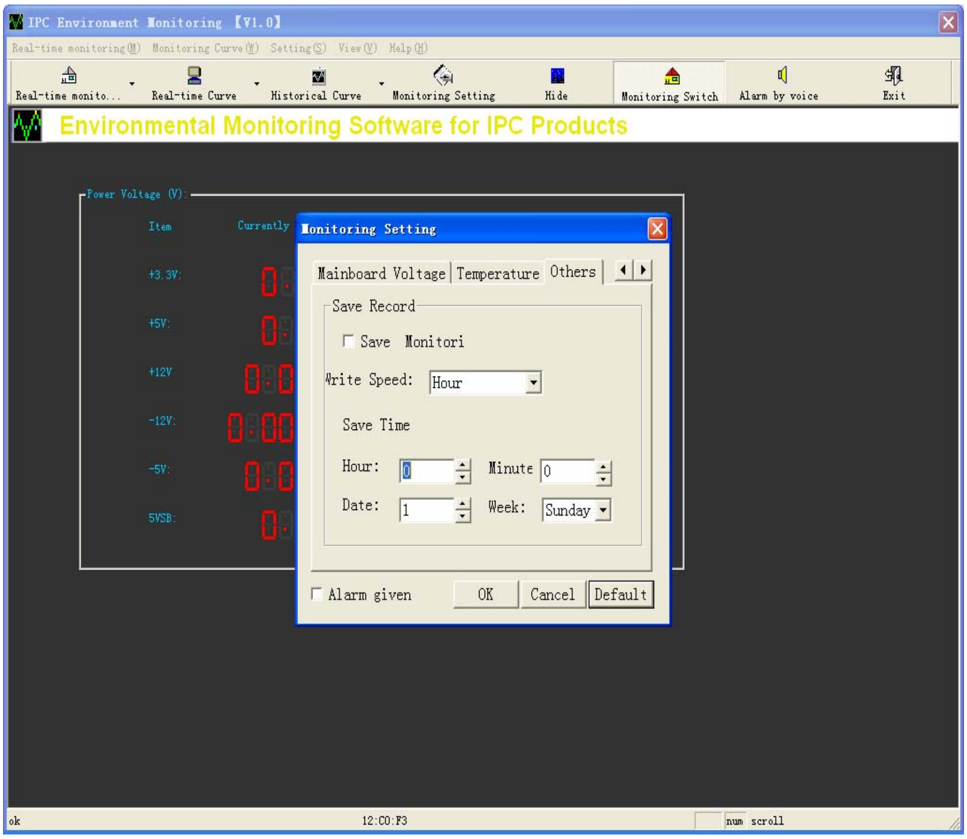
Fans1:	0
Fans2:	0
Fans3:	0

□ Alarm given

After the setup, click *Yes* to save the configuration, click *Cancel* to cancel the configuration, click *Default* to restore the parameters to default status.

Other Setup

The options of Saving Monitoring Record can be set here.



Monitoring Setting

Mainboard Voltage | Temperature | Others

Save Record

☐ Save Monitori

Write Speed: Hour

Save Time

Hour: 0 Minute 0

Date: 1 Week: Sunday

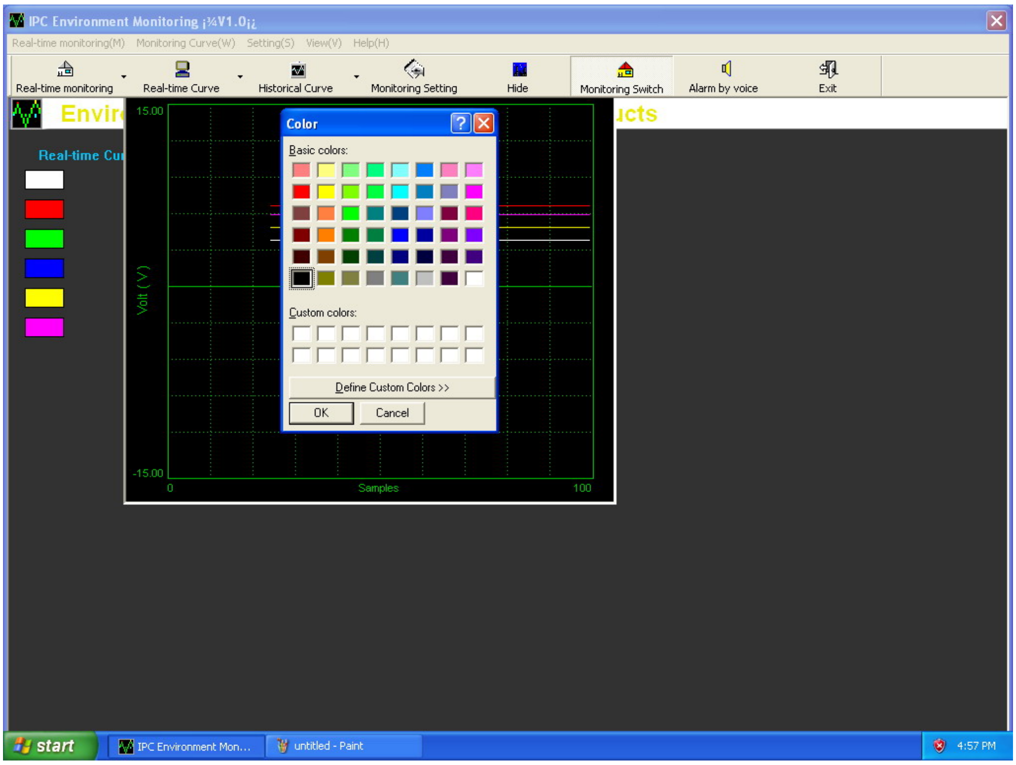
☐ Alarm given

OK Cancel Default

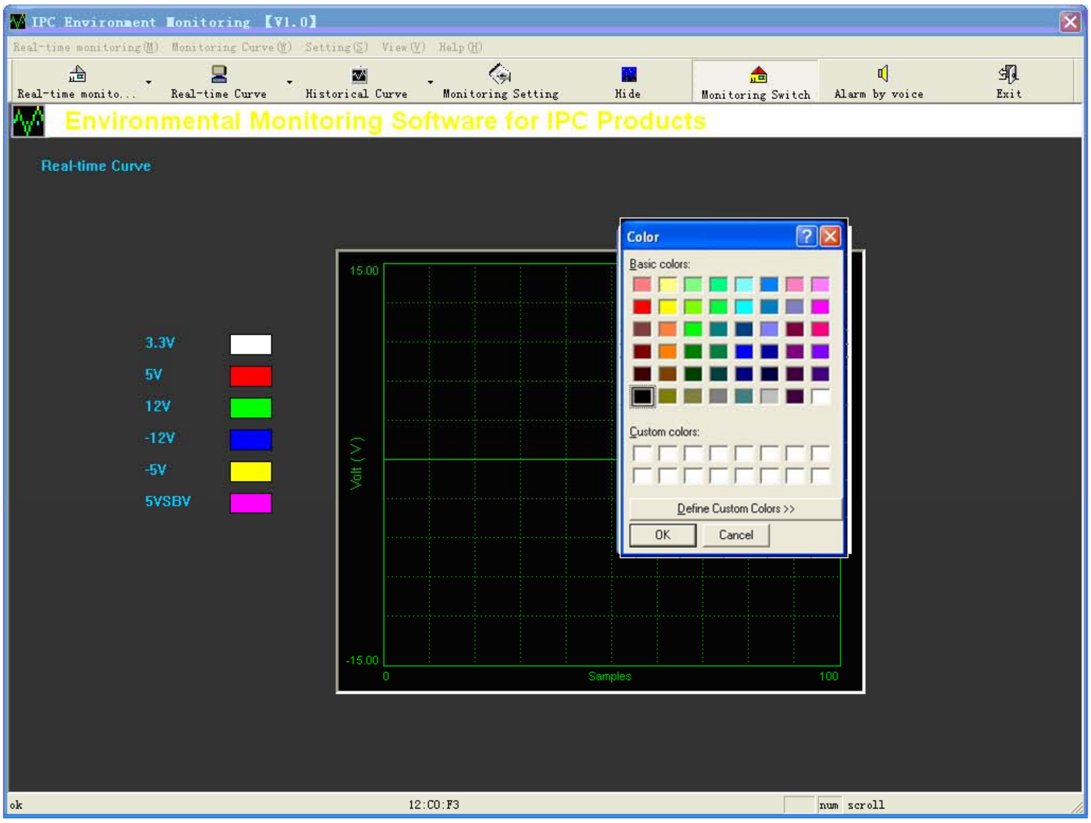
After the setup, click *Yes* to save the configuration, click *Cancel* to cancel the configuration, click *Default* to restore the parameters to default status.

Color Setup

LED Color Setup: Set the LED color.



Oscillograph grid color setup: Set the oscillograph grid color:

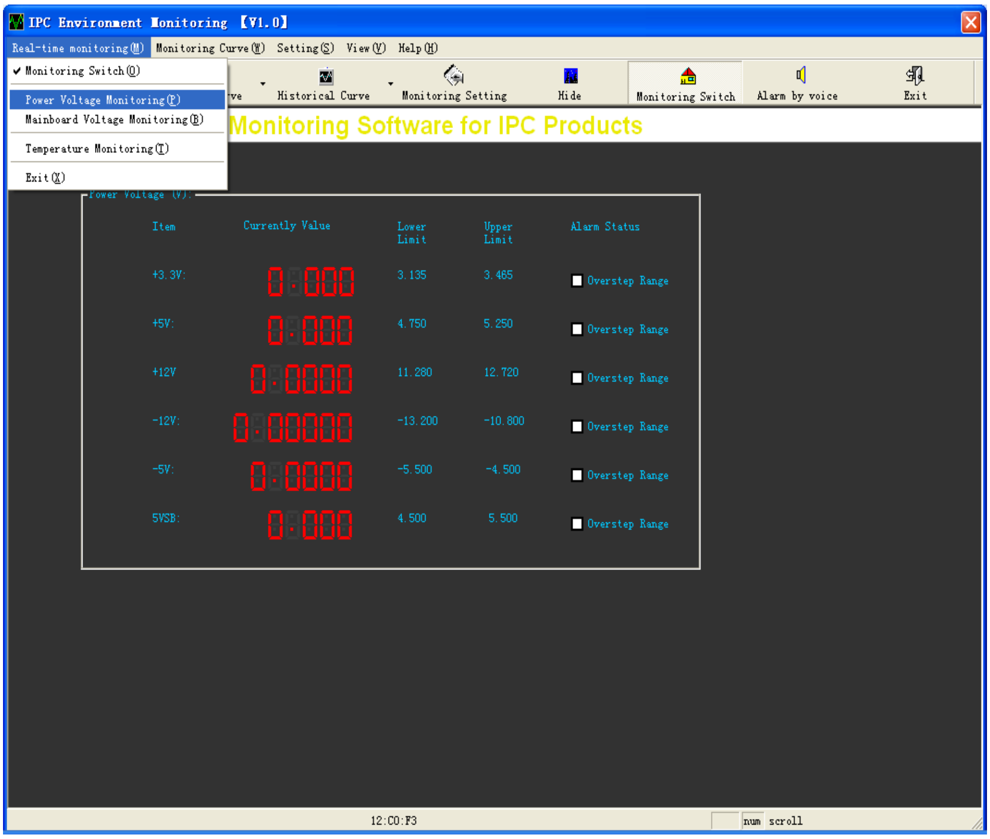


Chapter 5 Running of the Program

Real-time Monitor

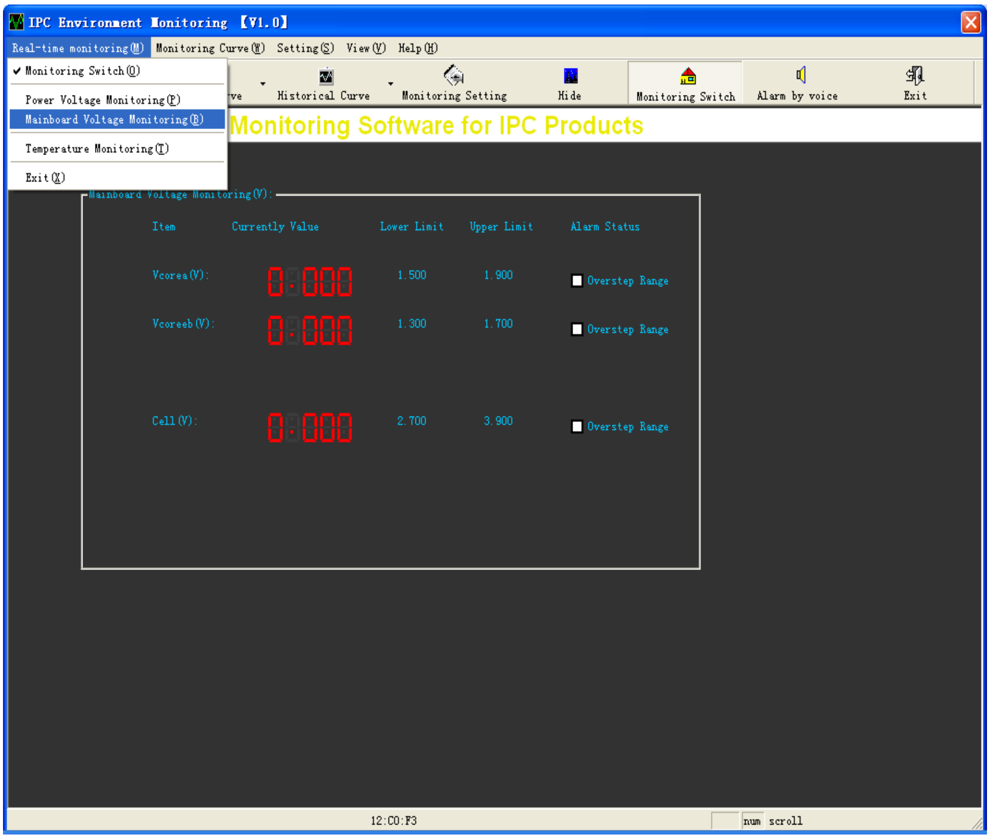
When the monitored value exceeds the limit value, it will give both status bar alert and sound alert.

(1) Power Supply Voltage Real-time Monitor: Indicates the real-time value of power supply voltage, upper and lower alerting limit and alerting status.



Item	Currently Value	Lower Limit	Upper Limit	Alarm Status
+3.3V:	0.000	3.135	3.465	□ Overstep Range
+5V:	0.000	4.750	5.250	□ Overstep Range
+12V	0.0000	11.280	12.720	□ Overstep Range
-12V:	0.00000	-13.200	-10.800	□ Overstep Range
-5V:	0.0000	-5.500	-4.500	□ Overstep Range
5VSB:	0.000	4.500	5.500	□ Overstep Range

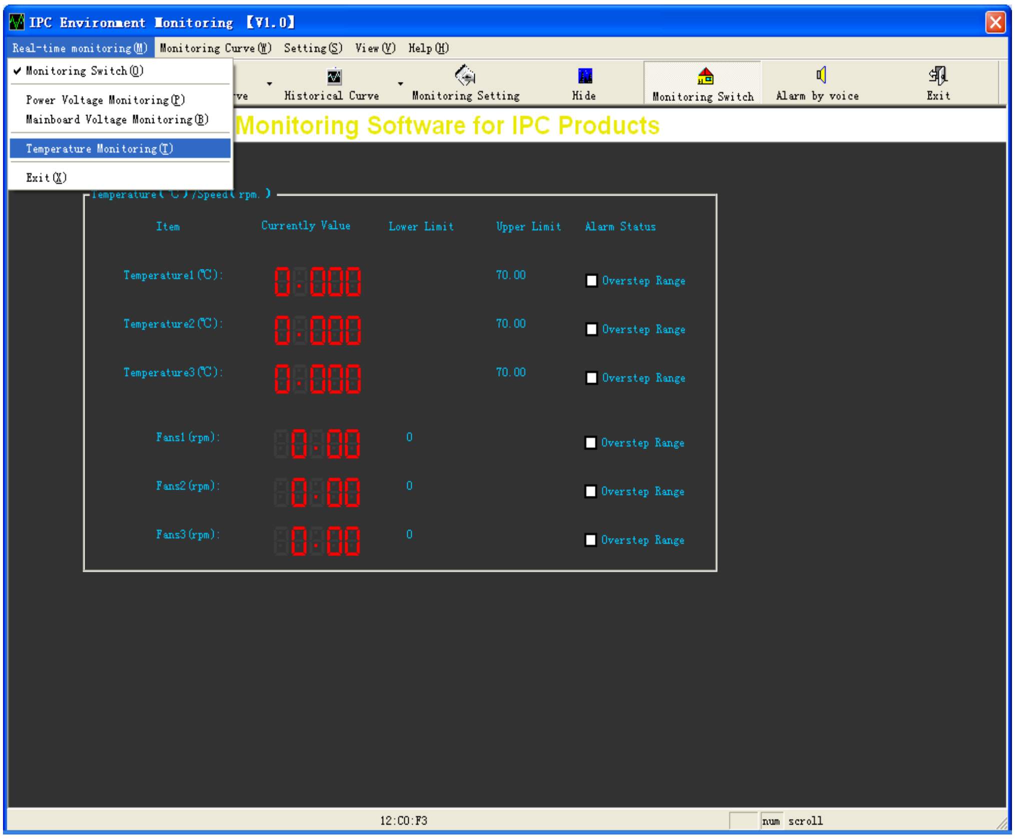
(2) Motherboard Voltage Real-time Monitor: Indicates the real-time value of power supply voltage, upper and lower alerting limit and alerting status.



Mainboard Voltage Monitoring(V):

Item	Currently Value	Lower Limit	Upper Limit	Alarm Status
Vcorea(V):	0.000	1.500	1.900	□ Overstep Range
Vcoreeb(V):	0.000	1.300	1.700	□ Overstep Range
Cell(V):	0.000	2.700	3.900	□ Overstep Range

(3) Temperature Real-time Monitor: Indicates CPU temperature, system temperature and fan speed real-time values, upper and lower alerting limit and alerting status.

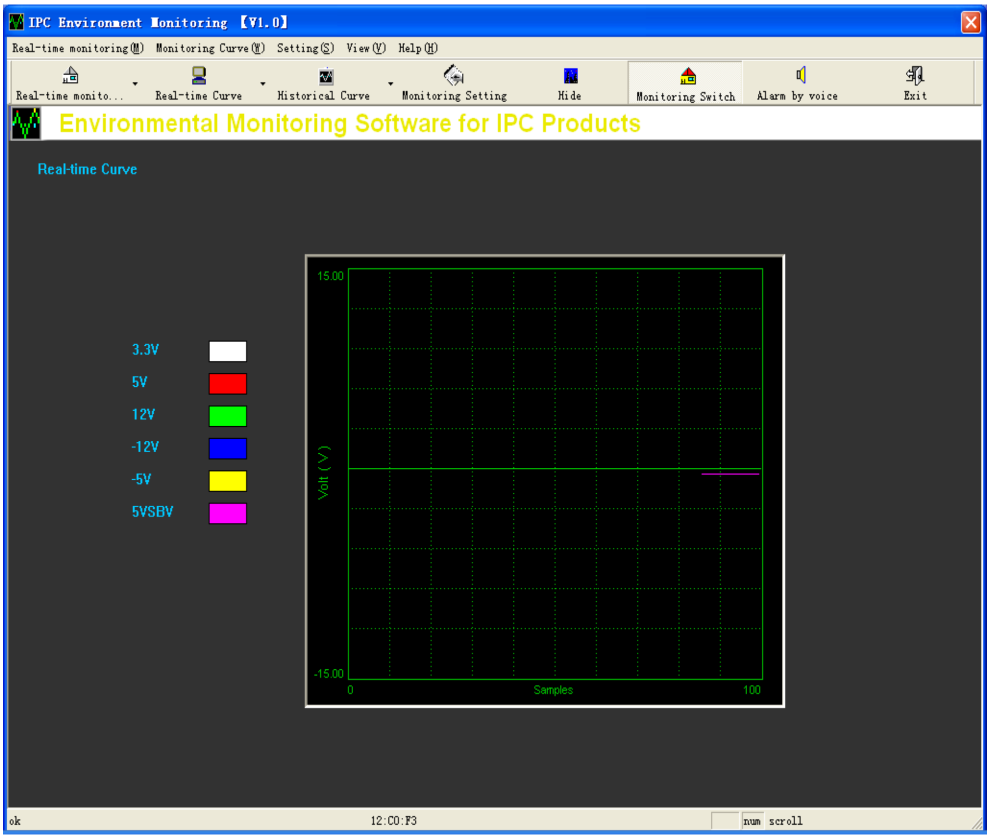


Temperature(℃)/Speed(rpm.)

Item	Currently Value	Lower Limit	Upper Limit	Alarm Status
Temperature1(℃):	0.000		70.00	□ Overstep Range
Temperature2(℃):	0.000		70.00	□ Overstep Range
Temperature3(℃):	0.000		70.00	□ Overstep Range
Fans1(rpm):	0.00	0		□ Overstep Range
Fans2(rpm):	0.00	0		□ Overstep Range
Fans3(rpm):	0.00	0		□ Overstep Range

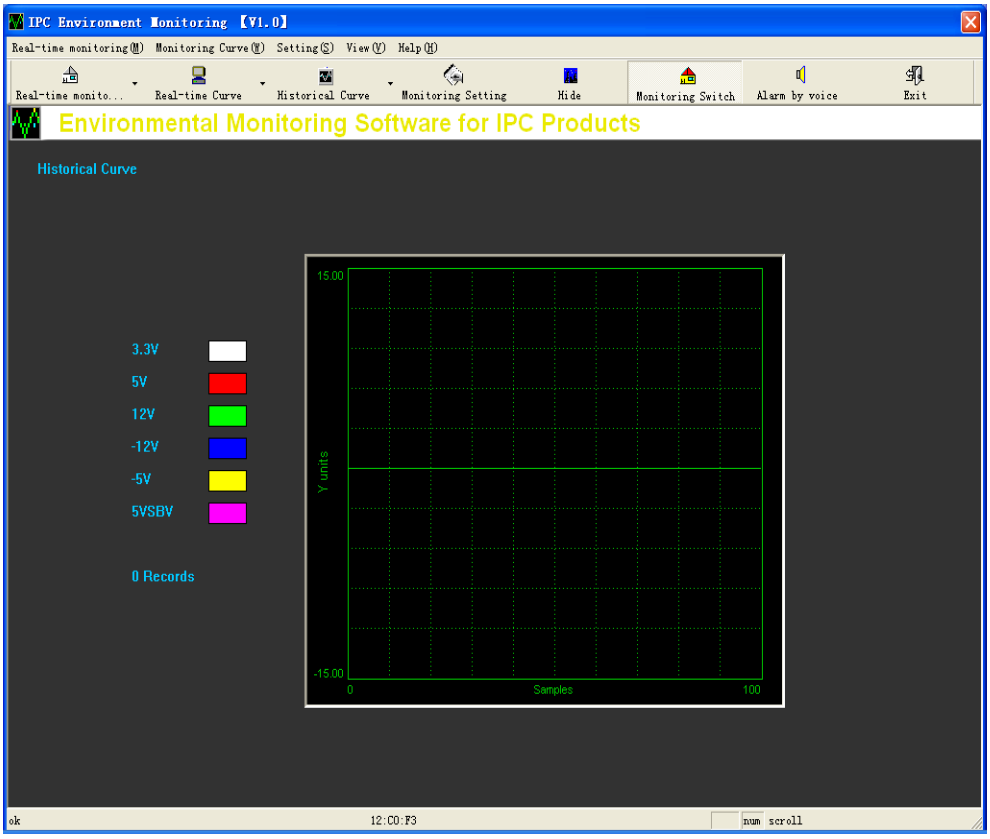
Real-time Curve

Indicates the real-time curve of power supply voltage, motherboard voltage, CPU and system temperature and fan speed.



History Curve

Indicates the history curve of power supply voltage, motherboard voltage, CPU and system temperature and fan speed.



Hide

Hide the monitoring interface.

Monitor On/off

Turns on/off the real-time monitor and indicates the real-time monitoring status.

Sound Alert

Turn on/off sound alert.

Exit the System

Click the ☒ on the top right corner, or click the Exit in Real-time Monitor menu, or click Exit button in tool bar, or right click the mouse and select Exit.

Chapter 6 Hardware Interface Function Introduction (EvocHM.dll)

int WINAPI HM_Initial (void);

Introduction: Initializes the hardware interface

Parameter: -

Return: 0 initialized normally

-1 initialized failed

int WINAPI HM_VCOREA_Read (double* volt);

Introduction: Read the CPU core voltage

Parameter: **volt:** Save the historical CPU core voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_VCOREB_Read (double* volt);

Introduction: Read the CPU boost voltage

Parameter: **volt:** Save the historical CPU boost voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_3_3V_Read (double* volt);

Introduction: Read +3.3V power supply voltage

Parameter: **volt:** Save the historical +3.3V power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_5V_Read (double* volt);

Introduction: Read +5V power supply voltage

Parameter: **volt:** Save the historical +5V power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_12V_Read (double* volt);

Introduction: Read +12V power supply voltage

Parameter: **volt:** Save the historical +12V power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_N12V_Read (double* volt);

Introduction: Read -12V power supply voltage

Parameter: **volt:** Save the historical -12V power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_N5V_Read (double* volt);

Introduction: Read -5V power supply voltage

Parameter: **volt:** Save the historical -5V power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_5VSB_Read (double* volt);

Introduction: Read 5VSB power supply voltage

Parameter: **volt:** Save the historical 5VSB power supply voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_VBAT_Read (double* volt);

Introduction: Read battery voltage

Parameter: **volt:** Save the historical battery voltage

Return: 0 read normally

-1 read failed

int WINAPI HM_Temp1_Read (double* temp);

Introduction: Read the temperature of sensor 1

Parameter: **temp:** Save the historical temperature values

Return: 0 read normally

-1 read failed

int WINAPI HM_Temp2_Read (double* temp);

Introduction: Read the temperature of sensor 2

Parameter: **temp:** Save the historical temperature values

Return: 0 read normally

-1 read failed

int WINAPI HM_Temp3_Read (double* temp);

Introduction: Read the temperature of sensor 3

Parameter: **temp:** Save the historical temperature values

Return: 0 read normally

-1 read failed

int WINAPI HM_Fan1_Read (unsigned int* speed);

Introduction: Read the speed of Fan 1

Parameter: speed: Save the historical fan speed

Return: 0 read normally

-1 read failed

int WINAPI HM_Fan2_Read (unsigned int* speed);

Introduction: Read the speed of Fan 2

Parameter: speed: Save the historical fan speed

Return: 0 read normally

-1 read failed

int WINAPI HM_Fan3_Read (unsigned int* speed);

Introduction: Read the speed of Fan 3

Parameter: speed: Save the historical fan speed

Return: 0 read normally

-1 read failed

Void WINAPI HM_Beep_On(unsigned int freq, unsigned char enable);

Introduction: Turn on/off sound alert

Parameter: freq: Alert frequency

Enable 0: turn off the sound alert
1: turn on the sound alert

Return: -

NOTE:

The values of CPU temperature, system temperature and CPU fan speed may be different from the locations of the sensors.

Chapter 7 Programming Introduction

(1) Compatible with Windows 98/2000/XP

(2) Call from VC++:

Add evochm.h and evochm.lib into VC++ application to call the function in evochm.dll.

(3) Call from C++builder:

Lead the lib file out of evocwdt.dll via the related programs of BC/C++builder develop kit. Then add the lib file and evochm.h into applications to call the functions in evochm.dll.

(4) Call from VB:

Then convert the evochm.h to evochm.BAS module file and add the module file into applications to call the functions in evochm.dll.

(5) Call from DELPHI:

Then convert the evochm.h to evochm.PAS declaration file and add the declaration file into applications to call the functions in evochm.dll.