Moxa Proactive Monitoring User's Manual

Edition 1.0, September 2015

www.moxa.com/product



Moxa Proactive Monitoring User's Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

© 2015 Moxa Inc. All rights reserved.

Trademarks

The MOXA logo is a registered trademark of Moxa Inc. All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

<u>Moxa An</u>	<u>iericas</u>	<u>Moxa Chi</u>	na (Shanghai office)
Toll-free:	1-888-669-2872	Toll-free:	800-820-5036
Tel:	+1-714-528-6777	Tel:	+86-21-5258-9955
Fax:	+1-714-528-6778	Fax:	+86-21-5258-5505
<u>Moxa Eu</u>	rope	<u>Moxa Asi</u>	a-Pacific
Tel:	+49-89-3 70 03 99-0	Tel:	+886-2-8919-1230
Fax:	+49-89-3 70 03 99-99	Fax:	+886-2-8919-1231

+91-80-4172-9088

+91-80-4132-1045

Tel: Fax:

Table of Contents

1.	Installing and Using Moxa Proactive Monitoring	
	Installing Moxa Proactive Monitoring	
	Installation Steps	
	Starting or Stopping the Moxa Proactive Monitoring Daemon	
	Monitoring System Status	
	How to display the Moxa Proactive Monitoring UI	
	How to use the Moxa Proactive Monitoring UI	
	Customizing Your Own Monitoring Items	1-5
	Setting the System Alarm	
	Changing Alarm Settings	1-7
	Performing an Alarm Action	
	Stopping an Alarm Action	
	Testing an Alarm Action	
2.	Moxa Proactive Monitoring API	
	API Functions	
	API Return Value Table	2-5

Installing and Using Moxa Proactive Monitoring

The following topics are covered in this chapter:

Installing Moxa Proactive Monitoring

- Installation Steps
- > Starting or Stopping the Moxa Proactive Monitoring Daemon

Monitoring System Status

- > How to display the Moxa Proactive Monitoring UI
- > How to use the Moxa Proactive Monitoring UI
- > Customizing Your Own Monitoring Items

Setting the System Alarm

- Changing Alarm Settings
- > Performing an Alarm Action
- > Stopping an Alarm Action
- > Testing an Alarm Action

Installing Moxa Proactive Monitoring

Installation Steps

- 1. Upload mxpromon_1.0.0_amd64.deb to the target machine.
- 2. Type the following command to install Moxa Proactive Monitoring:

root@Moxa:~# dpkg -i mxpromon_1.0.0_amd64.deb

Starting or Stopping the Moxa Proactive Monitoring Daemon

The Moxa Proactive Monitoring daemon **pro_mond** executes in the background when the system boots up, and then continuously monitors the status of the target machine. It logs alarm messages and performs alarm actions, if these features are activated. Take the following steps to stop or start the daemon.

How to stop the daemon

Type the following command to stop the daemon:

```
root@Moxa:~# /etc/init.d/pro_mond stop
```

How to start the daemon

Type the following command to start the daemon:

root@Moxa:~ # /etc/init.d/pro_mond start

How to restart the daemon

Type the following command to restart the daemon:

root@Moxa:~ # /etc/init.d/pro_mond restart

How to prevent the daemon from executing on boot up

If you don't want to execute the daemon when the system boots up, use the following command to turn it off:

root@Moxa:~ # insserv -r pro_mond

How to force the daemon to execute on boot up

If you want to execute the daemon when the system boots up, use the following command to turn it on:

root@Moxa:~ # insserv pro_mond

Monitoring System Status

NOTE We suggest using full screen viewing to display the Moxa Proactive Monitoring user interface. **Note:** <u>You cannot</u> <u>execute more than one instance of the Moxa Proactive Monitoring application at the same time</u>.

How to display the Moxa Proactive Monitoring UI

Use the following command to display the Moxa Proactive Monitoring user interface:

coot@Moxa:~ # pro_mon

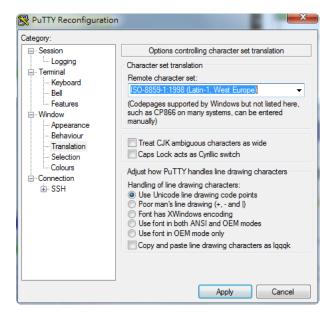
How to use the Moxa Proactive Monitoring UI

Four system status items will be displayed at the same time. To display other system status items, use your keyboard's **<Page Up>** or **<Page Down>** buttons to scroll up or down to the next page of status items.

Image: CPU Former: Mainboard[ON] Serial[ON] Network[ON] Setting Image: CPU Form Memory [ON] Disk[ON] Mainboard[ON] Serial[ON] Network[ON] Setting Image: CPU Form Memory Usage: 10.00 % Memory Usage: 10.00 % Memory Usage: 10.00 % Total Memory Size: 1068 Mbytes CPU CoreO Usage: 0.00 % CPU Temperature: 34 Degree C CPU Voltage: 0.624 V Mainboard Minboard Minboard Mainboard Temperature: 33 Degree C Mainboard Voltage: 5.056 V Image: CPU Voltage: 1008 Multiple for the state of the s	🗗 192.168.31.132 - PuTTY	
CFU CFU Joal Usage: 0.50 % CPU Logic Count : 2 CPU Core0 Usage: 0.00 % CPU Core1 Usage: 0.00 % CPU Voltage: 0.624 V PU Voltage: 0.624 V Mainboard Mainboard Mainboard Mainboard Mainboard Voltage: 5.056 V	[Moxa Pro	active Monitoring]
CPU Logic Count : 2 Total Memory Size : 1861 Mbytes CPU Core0 Usage : 0.00 % Available Memory Size : 1088 Mbytes CPU Core1 Usage : 0.624 V Available Memory Size : 1088 Mbytes Disk Mainboard /dev/sdal : Total:7456MB, Avail:5557MB, Usage:24% Mainboard Temperature : 33 Degree C Mainboard Voltage : 5.056 V Fille	CEU[ON] Memory[ON] Disk[ON] Main	<pre>board[ON] Serial[ON] Network[ON] Setting</pre>
/dev/sda1 : Total:7456MB, Avail:5557MB, Usage:24% Mainboard Temperature : 33 Degree C Mainboard Voltage : 5.056 V	CPU Logic Count : 2 CPU Core0 Usage : 0.00 % CPU Core1 Usage : 0.00 % CPU Temperature : 34 Degree C	Total Memory Size : 1861 Mbytes
		Mainboard Temperature : 33 Degree C Mainboard Voltage : 5.056 V

NOTE If garbage characters appear in the UI screen, as illustrated in the following screenshot, you may be able to rectify the situation by changing your computer's "Remote character set" from UTF-8 to a different encoding (ISO-8859-1, for example), and then restarting **pro_mon** (see the previous section).

3 192.168.31.132 - PuTTY	
[Moxa Proa	tive Monitoring]
CPU[ON] Memory[ON] Disk[ON] Mainbo	oard[ON] Serial[ON] Network[ON] Setting
x <mark>Cbî</mark> Tadaadaadadadadadadadadadadadadadadadad	xx _{Memoxy} x
xCPU Total Usage : 1.00 %	xxMemory Usage : 41.00 % x
xCPU Logic Count : 2	xxTotal Memory Size : 1861 Mbytes x
xCPU Core0 Usage : 0.99 %	xxAvailable Memory Size : 1088 Mbytes x
xCPU Corel Usage : 0.00 %	xx x
xCPU Temperature : 35 Degree C	xx x
xCPU Voltage : 0.816 V	xx x
x	XX X
	idyr addadadadadadadadadadadadadadadadadada
rqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq	xxMainboard x
x/dev/sda1 : Total:7456MB, Avail:5557MB, Usage:24%	xxMainboard Temperature : 33 Degree C x
X	xxMainboard Voltage : 5.056 V x
x	xx x
x	xx x
x	xx x
x	xx x
x	xx x
x	xx x
x	xx x
x	xx x
X	XX X
<u> </u>	14jmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
	seg0p>[1/2] seg0p>



Customizing Your Own Monitoring Items

 You can select your own system status to display in the UI window by turning on or off the status of different monitoring items. For example, if you don't want to show the CPU and memory status, you can turn off these features by pressing **<space>** or **<Enter>** on your computer's keyboard.

ở 192.168.31.132 - PuTTY	
[Moxa Preact	ive Nonitoring]
CPU[OFF] Memory[OFF] Disk[ON] Mainboa	rd[ON] Serial[ON] Network[ON] Setting
Disk /dev/sda1 : Total:7456MB, Avail:5557MB, Usage:24%	Mainboard Mainboard Temperature : 33 Degree C Mainboard Voltage : 5.056 V
Serial Serial P1 : Available Serial P2 : Available Serial P3 : Available Serial P4 : Available	Network Eth0 : Conneccted, Speed:1000Mb/s, Usage:0% Eth1 : Disconnectd
	≡
	<pre><pgup>[1/1]<pgdn5< pre=""></pgdn5<></pgup></pre>

 Furthermore, you can customize which items in each monitoring status category will be displayed. For example, if you don't want to see the CPU usage of each core, you can turn it off. To achieve this, select Setting and then press <Space> or <Enter> to enter the settings page.

P 1	192.168.31.132 -	PuTTY							×
				[Moxa Proact	tive Monitori	.ng]			^
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboa	ard[ON] Ser	cial[ON]	Network[ON]	Setting	
C			[Moxa]	Proactive Mo	nitoring Sett	ing Page]			
	CPU	Memory	Disk	Mainboard	Serial	Network	Setting	Information	
	Display CPU Total U CPU Logic C CPU Core0 U CPU Core1 U CPU Tempera CPU Voltage	Count [Show] Jsage [Show] Jsage [Show] ature [Show]				Jsage Alarm[Di ture Alarm[Di			E
								<pgup>[1/2]<pgd< td=""><td>n></td></pgd<></pgup>	n>

3. Choose which CPU items will be displayed, and which will be hidden.

B	192.168.31.132 -	PuTTY							x
Γ			[1	Moxa Proactiv	ve Monitori	ng]			^
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboard	I[ON] Ser	ial[ON]	Network[ON]	Setting	
	c c		[Moxa Pro	pactive Monit	oring Sett	ing Page]			
	C CPU	Memory	Disk Ma	ainboard	Serial	Network	Setting	Information	
	C Display CPU Total 1 CPU Logic (CPU Core0 1 CPU Core0 1 CPU Core1 1 CPU Temper/ CPU Voltage	Count[Show] Usage[Show] Usage[Show] ature[Show]		c		sage Alarm[ture Alarm[
			<save></save>			<cancel></cancel>			н
								<pg0p> [1/2] <pgdn< td=""><td></td></pgdn<></pg0p>	

4. For example, as shown below, you can turn off **CPU Usage**, and then press **<Tab>** to switch to the save menu to save the setting.

1 🛃	.92.168.31.132 -	PuTTY							x
			[Moxa Proact	ive Monit	coring]			^
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboa	rd[ON]	Serial[ON]	Network[ON]	Setting	
C			[Moxa Pr	coactive Mon	itoring :	Setting Page]			
C C C	CPU	Memory	Disk M	lainboard	Serial	Network	Setting	Information	
	CPU Logic (CPU Core0) CPU Core1)	Usage [Hidden] Cuage [Hidden] Usage [Hidden] ature [Show] e [Show]		Sav	Alarn	l Usage Alarm erature Alarm			
			<save></save>			<cancel></cancel>			E
								<pgup>[1/2]<pgdn< th=""><th>ļ</th></pgdn<></pgup>	ļ

5. Press **<Esc>** to return to the previous page. The CPU status will be updated in accordance with your selection.

🛃 192.168.31.132 - PuTTY				
	[Moxa Proact	ive Monitoring]		^
CPU[ON] Memory[ON]	Disk[ON] Mainboa	rd[ON] Serial[ON]	Network[ON]	Setting
CPU CPU Logic Count : 2 CPU Temperature : 35 Degree C CPU Voltage : 0.784 V		Memory Memory Usage : 41.00 Total Memory Size : Available Memory Siz	1861 Mbytes	
Disk /dev/sdal : Total:7456MB, Avail	:5557MB, Usage:24%	Mainboard Mainboard Temperatur Mainboard Voltage :		#
			<	J PgUp>[1/2] <pgdn> </pgdn>

Setting the System Alarm

Take the following steps to configure the system alarm settings.

Changing Alarm Settings

1. Click Setting.

B	192.168.31.132	- PuTTY							x
				Moxa Proactive	e Monitorin	g]			
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboard	[ON] Seri	al[ON]	Network[ON]	Setting	
									1
				coactive Monito					
	CPU	Memory	Disk N	lainboard S	Serial	Network	Setting	Information	
	Display			A	larm				
		Usage[Hidden] Count[Show]				age Alarm[E ure Alarm[E			
	CPU Core1	Usage[Hidden] Usage[Hidden]							
	CPU Temper CPU Voltag	rature[Show] ge[Show]							i
/									
									=
								<pgup>[1/2]<pgd< th=""><th></th></pgd<></pgup>	
									-

2. Click the CPU item under **[Moxa Proactive Monitoring Setting Page]** under **Alarm**, located near the middle of the page, and then configure the alarm settings here to monitor the system status and perform the alarm action when the system status is over the threshold value of the alarm settings.

8	192.168.31.132	- PuTTY							x
				[Moxa Proac	tive Monitor:	ing]			^
	CPU[ON]	Memory[ON]	Disk[ON] Mainbo	ard[ON] Se:	rial[ON]	Network[ON]	Setting	
			[Moxa	Proactive Mo	nitoring Set	ting Page]			
c	CPU	Memory	Disk	Mainboard	Serial	Network	Setting	Information	
	CPU Logic CPU Core0 CPU Core1	Usage[Hidden] Count[Show] Usage[Hidden] Usage[Hidden] cature[Show] ge[Show]				Jsage Alarm[ature Alarm[
			<save></save>			<cancel></cancel>			Ξ
								<pgup>[1/2]<pgdna< th=""><th>·</th></pgdna<></pgup>	·

- 3. To configure the alarm settings, press **<Space>** or **<Enter>** to turn on the alarm. The following example shows the CPU Temperature alarm setting.
- **NOTE** When you turn on the alarm option, by default the log message option will also turn on. If you don't want to turn on the log message option, you can turn it off yourself.

t بچ	192.168.31.132 -	ΡυΤΤΥ	<mark>I M</mark>	oxa Proac	tive Monitor	ing]			
C	CPU [ON]	Memory[ON]	Disk[ON] [Moxa Pro		ard[ON] Se nitoring Set	rial[ON] ting Page]	Network[ON]	Setting	
c	CPU	Memory	Disk Ma	inboard	Serial Alarm	Network	Setting	Informatio	n
	CPU Logic CPU Core0 CPU Core1	Usage[Hidden] Count[Show] Usage[Hidden] Usage[Hidden] ature[Show] e[Show]			CPU Temper Upper Th Lower Th Perform J		Enable] egrees C)[Dis egrees C)[Dis		
			<save></save>			<cancel></cancel>			
								<pgup>[1/2]</pgup>	<pgdn></pgdn>

4. After turning on the CPU Temperature alarm, you need to set the threshold value. For example, you can select the upper threshold item and press <e> to edit the upper threshold value of CPU Temperature as 10°C and then press <Space> or <Enter> to enable this option. In this case, when the CPU Temperature exceeds 10°C, the info on the window will change to red, and the alarm will be logged into the log file.

₽ 192.168.31.132	- PuTTY							x
		[]	Moxa Proact	ive Monito:	cing]			^
CBR[ON]	Memory[ON]	Disk[ON]	Mainboa	ard [ON] Se	erial[ON]	Network[ON]	Setting	
	Memory		pactive Mor ainboard	serial	sting Page] Network	Setting	Information	
CPU Logic CPU Core0 CPU Core1	Usage [Hidden] Count [Show] Usage [Hidden] Usage [Hidden] rature [Show] ge [Show]	Please, k	· · · ·	ue (-50 ~ : 0 <cance:< th=""><th>200) degrees C</th><th>able] ble] ees C)[Dis sable]</th><th></th><th></th></cance:<>	200) degrees C	able] ble] ees C)[Dis sable]		
Press <e></e>	to edit the valu	ie. <save></save>			<cancel></cancel>			E
							<pgup>[1/2]<pgd< th=""><th></th></pgd<></pgup>	

5. Next, press **<Tab>**, and then press **<Space>** or **<Enter>** to save your setting.

1 🧬	92.168.31.132	- PuTTY							x
			[Moxa Proact	ive Monitor	ring]			*
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboa	rd[ON] Se	erial[ON]	Network[ON]	Setting	
			[Moxa Pr	oactive Mon	itoring Set	cting Page]			
C	CPU	Memory	Disk M	ainboard	Serial	Network	Setting	Information	
	CPU Logic CPU Core0 CPU Core1 CPU Temper CPU Voltaç	Usage[Hidden] Count[Show] Usage[Hidden] Usage[Hidden] ature[Show] re[Show]	16.	Sav	ring Er Th Th m	nreshold(10	Enable] Degrees C)[Ena egrees C)[DIS	able] BDIej	
			<save></save>			<cancel></cancel>			=
								<pgup>[1/2]<pg< th=""><th></th></pg<></pgup>	

6. Continuing with the previous example, if the CPU temperature exceeds 10°C, the CPU Temperature will be highlighted in red, and the alarm will be logged in the log file. The log file path is:

/var/log/pro_mon/[date]_pro_mon_log

🛃 192.168.31.132 - PuTTY	
[Moxa Proact	ive Monitoring
CPU[ON] Memory[ON] Disk[ON] Mainboa	rd[ON] Serial[ON] Network[ON] Setting
CPU Logic Count : 2 CPU Temperature : 34 Degree C CPU Voltage : 0.624 V	Memory Memory Usage : 41.00 % Total Memory Size : 1861 Mbytes Available Memory Size : 1088 Mbytes
Disk /dev/sdal : Total:7456MB, Avail:5557MB, Usage:24%	Mainboard Mainboard Temperature : 33 Degree C Mainboard Voltage : 5.056 V
	E
	<pgup>[1/2]<pgdn></pgdn></pgup>

roo	t@Mox	ka:	~#	tail /	var/	log/p	ro	mon,	/20150902 pro	moi	n log								
Wed	Sep	2	18	:15:21	CST	2015		CPU	Temperature	(36	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:15:31	CST	2015		CPU	Temperature	(35	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:15:41	CST	2015		CPU	Temperature	(34	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:15:51	CST	2015		CPU	Temperature	(34	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:16:01	CST	2015		CPU	Temperature	(34	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:16:11	CST	2015		CPU	Temperature	(34	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:16:21	CST	2015		CPU	Temperature	(33	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:16:31	CST	2015		CPU	Temperature	(33	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
Wed	Sep	2	18	:16:41	CST	2015		CPU	Temperature	(34	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
				_	CST	2015		CPU	Temperature	(36	degrees	C)	is	over	the	upper	threshold	(10	degrees C)
roo	t@Mox	ka:	~#																

Performing an Alarm Action

1. If you want to perform the alarm action when the alarm occurs, check the **Perform alarm action** item, turn it on, and then save this setting.

B 1	92.168.31.132	- PuTTY						
			L	Moxa Proact	tive Monitor:	ing]		*
	CPU[ON]	Memory[ON]	Disk[ON]	Mainboa	ard[ON] Se:	rial[ON]	Network[ON]	Setting
C					litoring Set			
С	CPU	Memory	Disk N	fainboard	Serial	Network	Setting	Information
2	CPU Logic CPU Core0 CPU Core1	Usage[Hidden] Count[Show] Usage[Hidden] Usage[Hidden] rature[Show] ge[Show]			CPU Tempera Upper Th:		Enable] Degrees C)[En Lyrcco C [Dis	
								<pgup>[1/2]<pgdn></pgdn></pgup>

 After you save the setting, Moxa Proactive Monitoring will perform the alarm action when the alarm occurs. You can edit /sbin/mx_perform_alarm to change the alarm action.

The script will set the first digital output to high in the default setting.



NOTE If there is a relay on the device, the script will turn on the relay in the default setting.

Stopping an Alarm Action

- 1. To stop an alarm action, select setting and then click the terminate alarm action item to stop the alarm.
- **NOTE** If the alarm action is triggered, it will run nonstop until you press the terminate alarm action to stop it, even if the item returns to normal status.

<u>ب</u> 19	92.168.31.132 -	PuTTY		[Moxa Proac	tive Monitor	cing]			×
	CPU [ON]	Memory[ON]	Disk[ON]] Mainbo	ard[ON] Se	erial[ON]	Network[ON]	Setting	
	CPU	Memory		Proactive Mo Mainboard		tting Page] Network	Setting	Information	
		Nonitoring Rela	ted		e Alarm Act:				
	Terminate A	larm Action	eca		e Alarni Act.				
			<save></save>			<cancel></cancel>			=
								<pgup>[1/2]<pgd< td=""><td></td></pgd<></pgup>	

 Terminating an alarm action will execute the /sbin/mx_stop_alarm file. You can edit /sbin/mx_stop_alarm to change the stop alarm action.

The script will set the first digital output to low in the default setting.

root@Moxa:~# vi mx_stop_alarm
#!/bin/sh
<pre>mx_dio_control -s 0 -n 1</pre>

NOTE If there is a relay on the device, the script will turn off the relay in the default setting.

Testing an Alarm Action

If you want to test the **/sbin/mx_perform_alarm** and **/sbin/mx_stop_alarm** features, you can click the test alarm action to test it. It will execute **/sbin/mx_perform_alarm** for 3 seconds and then execute **/sbin/mx_stop_alarm**.

B	192.168.31.132 -	PuTTY							X
Γ				[Moxa Proac	tive Monitor:	ing]			*
	CPU[ON]	Memory[ON]	Disk[(ON] Mainbo	ard[ON] Sei	rial[ON]	Network[ON]	Setting	
	Scan Interv Alarm Relat	Memory ionitoring Rela ral(10 Secs) ed liarm Action Action for 3 S	Disk		Action for 3	Network	Setting	Information	
			<sav< th=""><th>=></th><th></th><th><cancel></cancel></th><th></th><th></th><th></th></sav<>	=>		<cancel></cancel>			
	_							<pgup>[1/2]<pgd< th=""><th>-</th></pgd<></pgup>	-

Moxa Proactive Monitoring API

An example of using the Moxa Proactive Monitoring API is stored in the folder **mx_pro_mon**. Refer to the sample code to learn how to apply the API functions described in this chapter.

The following topics are covered in this chapter:

- API Functions
- □ API Return Value Table

API Functions

Function	int get_average_cpu_usage(double *value)
Description	Obtain the average CPU utilization of the system.
Input	<value> The variable used to save the average CPU utilization of the system.</value>
Output	<value> The CPU utilization of the system.</value>
Return	Refer to the return value table.

Function	int get_cpu_count(int *value)
Description	Obtain the number of CPU cores in the system.
Input	<value> The variable used to save the number of CPU cores.</value>
Output	<value> The number of CPU cores.</value>
Return	Refer to the return value table.

Function	t get_cpu_usage(int index, double *value)						
Description	Obtain the specified CPU core utilization.						
Input	ndex> CPU number (0, 1, 2, 3)						
	<value> The variable used to save the CPU core utilization.</value>						
Output	<value> The specified CPU core utilization.</value>						
Return	Refer to the return value table.						

Function	nt get_mem_total_size(double *value)					
Description	Obtain the system's total memory size.					
Input	<value> The variable used to save the total memory size.</value>					
Output	<value> The system's total memory size.</value>					
Return	Refer to the return value table.					

Function	int get_mem_usage(int *value)
Description	Obtain the memory utilization of the system.
Input	<value> The variable used to save the memory utilization.</value>
Output	<value> The memory utilization.</value>
Return	Refer to the return value table.

Function	int get_mem_avail_size(double *value)
Description	Obtain the available memory size in the system.
Input	<value> The variable used to save the available memory size.</value>
Output	<value> The available memory size.</value>
Return	Refer to the return value table.

Function	int get_uart_count(int *value)
Description	Obtain the number of UART in the system.
Input	<value> The variable used to save the number of UARTs.</value>
Output	<value> The number of UARTs.</value>
Return	Refer to the return value table.

Function	int get_uart_status(int index, int *value)
Description	Obtain the specified UART status. (0) UART port is free, (1) in use
Input	<index> UART port (0, 1, 2)</index>
	<value> The variable used to save the specified UART status.</value>
Output	<value> The specified UART status. (0) UART port is free, (1) in use</value>
Return	Refer to the return value table.

Function	int get_eth_count(int *value)
Description	Obtain the number of Ethernet ports.
Input	<value> The variable used to save the number of Ethernet ports.</value>
Output	<value> The number of Ethernet ports.</value>
Return	Refer to the return value table.

Function	int get_eth_speed(int index, int *value)
Description	Obtain the specified Ethernet port speed.
Input	<index> The specified Ethernet port number (0, 1, 2)</index>
	<value> The variable used to save the specified Ethernet port speed.</value>
Output	<value> The specified Ethernet port speed.</value>
Return	Refer to the return value table.

Function	int get_eth_link(int index, int *value)
Description	Obtain the specified Ethernet port link status.
Input	<index> The specified Ethernet port number (0, 1, 2)</index>
	<value> The variable used to save specified Ethernet port link status.</value>
Output	<value> The specified Ethernet port link status.</value>
Return	Refer to the return value table.

Function	int get_eth_usage(int index, int *value)
Description	Obtain the specified Ethernet port utilization.
Input	<index> The specified Ethernet port number (0, 1, 2)</index>
	<value> The variable used to save the specified Ethernet port utilization.</value>
Output	<value> The specified Ethernet port utilization.</value>
Return	Refer to the return value table.

Function	int get_eth_link(int index, int *value)
Description	Obtain the specified Ethernet port link status.
Input	<index> The specified Ethernet port number (0, 1, 2)</index>
	<value> The variable used to save the specified Ethernet port link status.</value>
Output	<value> The specified Ethernet port link status.</value>
Return	Refer to the return value table.

Function	int get_disk_total_size(char *diskpath, double *value)
Description	Obtain the specified disk size.
Input	<diskpath> The specified disk location path</diskpath>
	<value> The variable used to save the specified disk size.</value>
Output	<value> The specified disk size.</value>
Return	Refer to the return value table.

Function	int get_disk_avail_size(char * diskpath, double *value)
Description	Obtain the amount of storage space available on the specified disk.
Input	<pre><diskpath> The specified disk location path</diskpath></pre>
	<value> The variable used to save the amount of storage space available on the specified disk.</value>
Output	<value> The amount of storage space available on the specified disk.</value>
Return	Refer to the return value table.

Function	int get_disk_usage(char *diskpath, double *value)
Description	Obtain the specified disk utilization.
Input	<pre><diskpath> The specified disk location path.</diskpath></pre>
	<value> The variable used to save the specified disk utilization.</value>
Output	<value> The specified disk utilization.</value>
Return	Refer to the return value table.

Function	int get_device_name(unsigned char *value)
Description	Obtain the device name of the platform.
Input	<value> The variable used to save the device name of the platform.</value>
Output	<value> The device name of the platform</value>
Return	Refer to the return value table.

Function	int get_bios_ver(unsigned char *value)
Description	Obtain the bios version.
Input	<value> The variable used to save the bios version.</value>
Output	<value> The bios version.</value>
Return	Refer to the return value table.

Function	int get_ser_num(unsigned char *value)
Description	The get_ser_num function is used to obtain the device serial number.
Input	<value> The variable used to save the device serial number</value>
Output	<value> The device serial number</value>
Return	Refer to the return value table.

Function	int get_pwr_status(int port, int *value)
Description	Obtain the power supply status.
Input	<port> The power supply number (0, 1)</port>
	<value> The variable used to save the status of the power supply.</value>
Output	<value> The status of the power supply.</value>
Return	Refer to the return value table.

Function	int set_relay(int port, int value)
Description	Set relay status by the input value (0) off (1) on.
Input	<port> relay port (0, 1)</port>
	<value> set status: (0) off, (1) on</value>
Output	None
Return	Refer to the return value table.

Function	int get_milli_volt(int index, unsigned int *value)
Description	Obtain the specified device voltage by index.
Input	< index> The Voltage type: (0) CPU, (1) Memory, (2) Mainboard
	<value> The variable used to save the specified device voltage.</value>
Output	<value> The specified device voltage.</value>
Return	Refer to the return value table.

Function	int get_temperature(int index, unsigned int *value)
Description	Obtain the specified device temperature.
Input	< index> The Voltage type: (0) CPU, (1) System
	<value> The variable used to save the specified device temperature.</value>
Output	<value> The specified device temperature.</value>
Return	Refer to the return value table.

API Return Value Table

value	Meaning
0	The operation has completed successfully.
-100	The parameter is invalid.
-101	The command popen failed to run. Unable to open the process.
-102	The command pclose failed to run. Unable to close the process.
-103	The system cannot open the device node.
-104	The system cannot close the device node.
-105	The IOCTL call made by the application program is not correct.
-106	Could not load the ini file.
-107	The source command was not found in the sh shell.
-108	The system cannot find the ini key.
-200	The system cannot get the device name.
-201	The system cannot get the sensor value.
-202	The system cannot get the BIOS version.
-203	The system cannot get the serial number.
-204	This system does not support relay.
-205	This system does not support power indicator.
-300	Failed to get Ethernet port status.