

# 产品说明书

**GM0-1613-01**

**智能物联主控板**

**版本：A00**

## 法律资讯

### 警告提示

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。



#### 危险

表示如果不采取相应的小心措施，将会导致死亡或者严重的人身伤害。



#### 警告

表示如果不采取相应的小心措施，可能导致死亡或者严重的人身伤害。



#### 小心

带有警告三角，表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

#### 注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

### 合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的合格人员进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

### GITSTAR产品

请注意下列说明：



#### 警告

GITSTAR产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到GITSTAR推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

## 免责声明

本公司保留对此手册更改的权利，产品后续相关变更时，恕不另行通知。对于任何因安装、使用不当、超规格使用而导致的直接、间接、有意或无意的损坏及隐患概不负责。

订购产品前，请向经销商详细了解产品性能是否符合您的需求。

**GITSTAR**是北京集特智能科技有限公司的注册商标。本手册所涉及到的其他商标，其所有权为相应的产品厂家所拥有。

## 保修条款：

产品保修条款详见官网说明。用户如另有要求，以双方签署的合同为准。

## 欲获更多信息请访问：

集特网站：<http://graest.com>

## 文档说明

### 本文档适用范围

本文档适用于**GITSTAR GM0-1613-01** 型号。

### 约定

在本文档中，术语“本板”或“产品”有时特指 **GITSTAR GM0-1613-01**产品。

### 说明

#### 安全相关注意事项

为避免财产损失以及出于个人安全方面的原因，请注意本入门指南中关

于安全方面的信息。文中使用警告三角来指示这些安全信息，警告三角的出现取决于潜在危险的程度。

## 历史

本说明书发布版本：

版本	时间
A00	2021.9

# 目录

1.产品介绍 .....	1
1.1 概述 .....	1
1.2 机械尺寸、重量与环境 .....	2
1.3 典型功耗 .....	2
1.4 微处理器 .....	2
1.5 芯片组 .....	3
1.6 系统内存 .....	3
1.7 显示功能 .....	3
1.8 网络功能 .....	3
1.9 音频功能 .....	3
1.10 电源特性 .....	3
1.11 扩展总线 .....	3
1.12 Watchdog 功能 .....	4
1.13 操作系统 .....	4
1.14 I/O 接口 .....	4
2.安装说明 .....	5
2.1 产品外形尺寸图 .....	5
2.2 接口位置示意图 .....	6
2.3 架构图 .....	7
2.4 跳线设置 .....	8
2.5 串口 .....	9
2.6 键盘/鼠标组合接口 .....	10

2.7 DVI 接口 .....	10
2.8 标准 DB15 VGA 接口 .....	11
2.9 LPT 接口 (IDC 连接器) .....	11
2.10 前面板接口 .....	12
2.11 FP2 扬声器输出接口 (脚距: 2.0mm JST) .....	12
2.12 USB 接口 .....	12
2.13 音频接口 .....	13
2.14 24Pin ATX 电源接口 .....	14
2.15 ATX 8Pin 12V 接口 .....	14
2.16 4PIN 风扇接头 .....	15
2.17 7Pin SATA 接口 .....	15
2.18 LVDS 背光控制接口(脚距: 2.0mm JST).....	<b>错误! 未定义书签。</b>
2.19 网络接口 .....	15
2.20 HDMI 接口 .....	16
2.21 GPIO 接口(脚距: 2.0mm) .....	16
3.BIOS 功能介绍 .....	17
4.驱动程序安装说明 .....	18
5.附录 .....	19
5.1 KIDI 简介 .....	19
5.1.1 KIDI 目的 .....	19
5.1.2 KIDI 优势 .....	20
5.2 KIDI 编程接口 .....	21

5.3 KIDI 函数调用步骤.....	30
5.3.1 硬件看门狗.....	31
5.3.2 用户可编程 GPIO.....	34
5.3.3 硬件监控.....	35
5.4 功能演示.....	37
5.4.1 看门狗测试.....	37
5.4.2 GPIO 测试.....	39
5.4.3 硬件监控测试.....	44
5.5 产品订购信息.....	42
5.6 缩略语.....	42

## 1.1 概述

GM0-1613-01采用Intel® H310C芯片组，支持Intel® Socket LGA1151封装的6th/7th/8th/9th Generation Intel® Core™、Pentium™、Celeron® G系Desktop系列CPU，产品提供2条288 Pin DDR4 DIMM内存插槽，支持Un-buffered Non-ECC 内存，支持双通道功能，单条内存插槽可支持最大内存容量16GB，总支持容量最大32GB，支持的内存频率2133MHz/2400MHz/2666MHz<sup>1</sup>。

支持VGA、DVI-D、HDMI 三个显示接口输出，可选择任意两个搭配，组成复制或扩展显示模式。

产品支持4个USB3.0 + 6个USB2.0；支持2个PCIe + 3个PCI设备扩展；板载2个Intel千兆网口；支持High Definition Audio Codec。

EM接口支持扩展独立显示模块，配合BIOS中间件，接收、主动读取相关信息，显示系统配置、状态以及故障或警告信息。

该主板符合标准ATX主板规范，能广泛应用自动化生产及检测、银行安防监控系统、智能楼宇管理系统（IBMS）、综合保安管理系统（ISMS）、综合交通管理系统（ITMS）、地铁综合监控系统、家庭智能化系统、呼叫中心系统、公共安全录音系统、各行业指挥调度中心、机器视觉（如自动化检测、图像自动分析处理）、工作站、保险、证券等行业录音监听监控系统等。

### GM0-1613-01系列主板规格：

型号	主要规格
GM0-1613-01	H310C, VGA+DVI+HDMI, PCIe(x16/ x4)+PCI*3, COM*6

<sup>1</sup> 2400MHz 需要 intel 第 7 代 kabylake CPU 支持，2666MHz 需要第 8/9 代 coffelake CPU 支持；

## 1.2 机械尺寸、重量与环境

- 外形尺寸：305mm（长）×244mm（宽）×37mm（高）
- 净重：0.5Kg；
- 工作环境：
  - 温度：0℃～60℃；
  - 湿度：5%～95%（非凝结状态）；
- 贮存环境：
  - 温度：-20℃～80℃；
  - 湿度：5% 95%（非凝结状态）；

## 1.3 典型功耗

典型功耗是基于以下配置闲置状态的数值。

CPU： I7 CPU (95W)

内存：DDR4/2400/8GB

硬盘：1个500G 3.5寸 SATA接口硬盘

操作系统：Win7 32位

- +5V @1.1A； +5%/-3%
- +3.3V@0.8A； +5%/-3%
- +12V@0.1A； +5%/-3%
- +12V P4@2.8A； +5%/-3%

## 1.4 微处理器

支持LGA1151封装Intel®Skylake/Kabylake/Coffeelake的Core™、Pentium™、Celeron™ 等 CPU。

## 1.5 芯片组

Intel® H310C

## 1.6 系统内存

最多提供 2 条 288Pin DDR4 内存插槽，支持 Un-buffered NON-ECC，支持双通道功能。单条内存插槽可支持最大内存容量 16GB，总支持最大内存容量 32GB。支持的内存频率 2133MHz/2400MHz/2666MHz，其中 2400MHz 需要 intel 第 7 代 kabylake CPU 支持，2666MHz 需要 intel 第 8/9 代 coffelake CPU 支持。

## 1.7 显示功能

- 板载VGA、DVI-D、HDMI显示接口，支持双显的复制、扩展桌面功能；DVI、HDMI支持热插拔功能；
- 最高分辨率支持：  
VGA： 1920×1200@60Hz；  
HDMI： 4096×2304@24Hz；  
DVI： 4096×2304@24Hz；

## 1.8 网络功能

提供2个10/100/1000Mbps网络接口，LAN1可支持网络唤醒功能；

## 1.9 音频功能

采用HDA标准，支持MIC-IN/LINE-IN/LINE-OUT音频接口。

## 1.10 电源特性

采用标准ATX 24+8PIN电源，支持来电自动开机功能。

## 1.11 扩展总线

- 提供三个PCI扩展插槽；
- 提供两个PCIe扩展插槽，一个PCIe x16插槽，一个PCIe x4插槽（插槽尾端开口，支持x8、x16长卡插入）。

## 1.12 Watchdog 功能

- 支持 255 级，可编程按分或秒；
- 支持看门狗超时中断或复位系统。

## 1.13 操作系统

- 支持操作系统：支持 Windows 10、Windows 8、Windows 7、Linux 等系统；

## 1.14 I/O 接口

- 提供 1 个并口，支持 SSP/EEP/ECP 工作模式；支持 BIOS 选择工作模式；
- 提供 6 个串口，其中 COM1、COM2 支持 RS-232/RS-422/RS-485 模式选择，支持 RS485 自动流控功能，COM3~COM6 支持 RS232，并可通过扩展模块扩展至 10 个 RS232 串口；
- 提供 4 个 SATA3.0 接口，默认支持 3 个，最大传输速率 6Gb/s；
- 提供 1 个 M.2(SATA3.0)接口，可支持 M-key 22110, 2280, 2260, 2242；
- 提供 6 个 USB2.0 接口，4 个 USB3.0 接口；CN8 仅支持一个 USB 设备；CN7 与 CN7-1 接口资源共用，2 种接口形式方便不同机箱前面板连接；
- 提供 1 个 PS/2 键盘/鼠标接口；
- 提供 1 个 8 路数字 I/O 接口。

### 提示：如何识别报警声

- 1、长鸣声为系统内存出错。
- 2、短“嘀”一声为开机声。







**提示：如何识别跳线、接口第一脚**

观察插头、插座旁边的文字标记，通常用加粗的线条或三角符号表示。

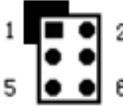
**2.4 跳线设置****1. CLR\_CMOS1：CMOS内容清除/保持设置（脚距：2.0mm）**

CMOS由主板上钮扣电池供电。清除CMOS会导致永久性消除以前系统配置并将其设为初始（工厂设置）系统设置。操作步骤：(1)关闭计算机，断开电源；(2)瞬间短接JCC1插针；(3)开计算机；(4)启动时按屏幕提示按键进入BIOS设置，重新加载最优缺省值；(5)保存并退出设置。设置方式如下：

	设置	功能
	1-2 开路	正常工作状态（Default）
	1-2 短路	清除 CMOS 内容，所有 BIOS 设置恢复成出厂值。

**2. 串口配置（脚距：2.0mm）**

主板的COM1、COM2支持（RS-232/RS-422/RS-485）模式，通过JP1~6设置：JP1~3设置对应COM1串口模式：

	COM1	RS232	RS485	RS422
JP1~3	JP1	1-2	3-4	5-6
		1-3	3-5	3-5
	JP2	2-4	4-6	4-6
		1-3	3-5	3-5
	JP3	2-4	4-6	4-6

JP4~6设置对应COM2串口模式:

 JP4~6	COM2	RS232	RS485	RS422
	JP4	1-2	3-4	5-6
	JP5	1-3	3-5	3-5
		2-4	4-6	4-6
	JP6	1-3	3-5	3-5
	2-4	4-6	4-6	

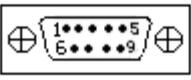
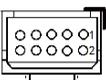
### 3. AT/ ATX模式选择 (脚距: 2.0mm)

通过设置JP7选择主板AT工作模式或ATX工作模式。主板工作在AT模式时, 来电自动开机, 不需要按POWER BUTTON。

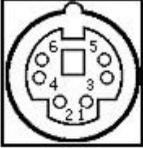
	设置	功能
	1-2 开路	ATX 模式 (Default)
	1-2 短路	AT 模式

## 2.5 串口

本主板支持6个RS232串口, 包括1个后置DB9接口、5个2.0mm Wafer接口, 接口的信号定义如下:

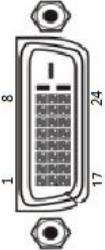
 COM1   COM2~4	管脚	信号名称		
		RS232	RS422	RS485
1	DCD#	TX-	Data-	
2	RXD	TX+	Data+	
3	TXD	RX+	--	
4	DTR#	RX-	--	
5	GND	GND	GND	
6	DSR#	--	--	
7	RTS#	--	--	
8	CTS#	--	--	
9	RI#	--	--	

## 2.6 键盘/鼠标组合接口

 <p>CN4</p>	管脚	信号名称
	1	KB_DATA
	2	MS_DATA
	3	GND
	4	+5V
	5	KB_CLK
	6	MS_CLK

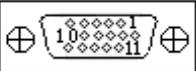
## 2.7 DVI 接口

本主板支持1个后置DVI-D接口，接口信号定义如下：

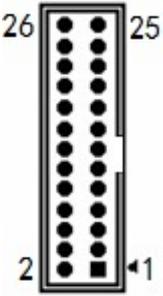
 <p>CN1(DVI)</p>	管脚	信号名称	管脚	信号名称
	1	DATA2-	13	NC
	2	DATA2+	14	+5V
	3	GND_DVI	15	GND
	4	NC	16	HOTPLUG
	5	NC	17	DATA0-
	6	DDCCLK	18	DATA0+
	7	DDCDATA	19	GND_DVI
	8	NC	20	NC
	9	DATA1-	21	NC
	10	DATA1+	22	GND_DVI
	11	GND_DVI	23	CLK+
	12	NC	24	CLK-

## 2.8 标准 DB15 VGA 接口

产品支持1个DB15 VGA接口，采用与DVI接口组合连接器，接口的定义如下：

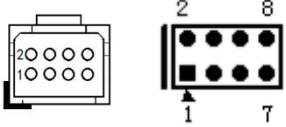
 CN1 (VGA)	管脚	信号名称	管脚	信号名称
	1	Red	2	Green
	3	Blue	4	NC
	5	GND	6	GND
	7	GND	8	GND
	9	NC	10	GND
	11	NC	12	DDCDATA
	13	HSYNC	14	VSYNC
	15	DDCCLK		

## 2.9 LPT 接口（IDC 连接器）

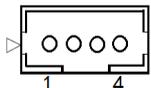
 LPT1 (脚距：2.54mm)	管脚	信号名称	管脚	信号名称
	1	STB#	2	AFD#
	3	PD0	4	ERR#
	5	PD1	6	INIT#
	7	PD2	8	SLIN#
	9	PD3	10	GND
	11	PD4	12	GND
	13	PD5	14	GND
	15	PD6	16	GND
	17	PD7	18	GND
	19	ACK#	20	GND
	21	BUSY	22	GND
	23	PE	24	GND

## 2.10 前面板接口

本主板支持两个前面板连接器，一个为2.0mm Wafer接口，一个为2.54mm接口，两接口信号定义相同，可根据需求选择不同连接器：

 <p>FP1 (间距 2.0)      FP1-1 (间距 2.54)</p>	管脚	信号名称	管脚	信号名称
	1	PWRBTN#	2	GND
	3	GND	4	RESET#
	5	HDD_LED-	6	HDD_LED+
	7	PWR_LED-	8	PWR_LED+

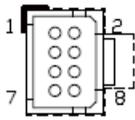
## 2.11 FP2 扬声器输出接口 (脚距: 2.0mm JST)

 <p>FP2</p>	管脚	信号名称
	1	BUZ+
	2	GND
	3	NC
	4	BUZ-

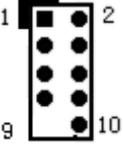
## 2.12 USB 接口

本主板支持4个USB3.0标准接口（采用双层USB与RJ45组合连接器），支持8个USB2.0接口，采用2.0mm Wafer连接器，其中连接器CN7与CN7-1（采用2.54mm 2x5插针）连接相同USB口，方便用户连接机箱前面板，接口的信号定义如下：

### 1. 2.0mm Wafer USB接口

 <p>CN7, CN8</p>	管脚	信号名称	管脚	信号名称
	1	+5V	2	+5V
	3	USB1_Data-	4	USB2_Data-
	5	USB1_Data+	6	USB2_Data+
	7	GND	8	GND

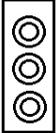
## 2. 2.54mm 插针USB接口

 <p>CN7-1</p>	管脚	信号名称	管脚	信号名称
	1	+5V_USB	2	+5V_USB
	3	USB1_DATA-	4	USB2_DATA-
	5	USB1_DATA+	6	USB2_DATA+
	7	GND	8	GND
9	NA	10	GND	

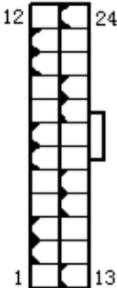
## 3. 标准USB3.0接口

 <p>CN2 (USB1/USB2) CN3 (USB3/USB4)</p>	管脚	信号名称
	1	+5V
	2	USB_DATA-
	3	USB_DATA+
	4	GND
	5	USB_SSRX-
	6	USB_SSRX+
	7	GND
	8	USB_SSTX-
9	USB_SSTX+	

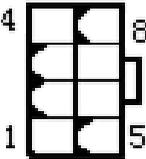
## 2.13 音频接口

	管脚	信号名称
	1	LINE-IN
	2	LINE-OUT
3	MIC-IN	

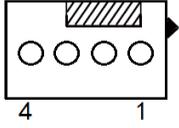
## 2.14 24Pin ATX 电源接口

 ATX1	管脚	信号名称	管脚	信号名称
	1	+3.3V	13	+3.3V
	2	+3.3V	14	-12V
	3	GND	15	GND
	4	+5V	16	PS_ON#
	5	GND	17	GND
	6	+5V	18	GND
	7	GND	19	GND
	8	PWROK	20	-5V
	9	+5VSB	21	+5V
	10	+12V	22	+5V
	11	+12V	23	+5V
	12	+3.3V	24	GND

## 2.15 ATX 8Pin 12V 接口

 ATX_12V1	管脚	信号名称	管脚	信号名称
	1	GND	5	+12V
	2	GND	6	+12V
	3	GND	7	+12V
	4	GND	8	+12V

## 2.16 4PIN 风扇接头

 <p>CPU_FAN1, SYS_FAN1</p>	管脚	信号名称
	1	GND
	2	+12V
	3	FAN_IO
	4	FAN_PWM

注：FAN\_IO：风扇转速输出； FAN\_PWM：风扇转速控制。

## 2.17 7Pin SATA 接口

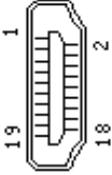
 <p>SATA1~4</p>	管脚	信号名称
	1	GND
	2	SATA_TX+
	3	SATA_TX-
	4	GND
	5	SATA_RX-
	6	SATA_RX+
	7	GND

## 2.19 网络接口

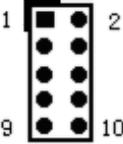
主板支持 2 个 10/100/1000Mbps 网络 RJ45 接口，包括 CN3 (LAN1) 和 CN4 (LAN2) 其中CN3 (LAN1) 支持网络唤醒。以太网接口指示灯包括 ACTLED 和 LILED，它们显示着LAN的活动指示状态和网络速度指示状态。请参考以下LED的状态描述：

活动状态指示 (单色：黄灯)		网速指示 (双色：橙绿)	
有数据传输	闪烁	1000Mbps	绿色
无数据传输	灭	100Mbps	橙色
		10Mbps	灭

## 2.20 HDMI 接口

 HDMI1	管脚	信号名称	管脚	信号名称
	1	TMDS DATA2+	2	TMDS DATA2 Shield
	3	TMDS DATA2-	4	TMDS DATA1+
	5	TMDS DATA1 Shield	6	TMDS DATA1-
	7	TMDS DATA0+	8	TMDS DATA0 Shield
	9	TMDS DATA0-	10	TMDS CLK+
	11	TMDS CLK Shield	12	TMDS CLK-
	13	CEC	14	RSVD(NC on device)
	15	SCL	16	SDA
	17	DDC/CEC Ground	18	+5V
	19	Hot Plug Detect	-	-

## 2.21 GPIO 接口 (脚距: 2.0mm)

 脚距: 2.0mm	管脚	信号名称	管脚	信号名称
	1	INPUT 0	2	OUTPUT 0
	3	INPUT 1	4	OUTPUT 1
	5	INPUT 2	6	OUTPUT 2
	7	INPUT 3	8	OUTPUT 3
	9	GND	10	NC/5v

### 3. BIOS 功能介绍

---

BIOS的设置及功能介绍，请参照《BIOS设置指南》。

#### 4. 驱动程序安装说明

---

本产品的驱动程序可依据配套光盘内容安装，在此不做介绍。

---

---

## 5.1 KIDI 简介

### 5.1.1 KIDI 目的

KIDI是集特智能研发的一种跨平台、安全、简单易用、易维护的智能部署接口规范，它包含丰富的操作系统下访问底层硬件的软件接口。

集特智能研发KIDI的目的：使用户在集特智能某款产品上开发的软件产品能通用于集特智能所有的硬件平台，达到软件与硬件无缝对接，同时极大程度地简化硬件访问编程工作，软件工程师可完全不需要了解任何驱动开发相关方面的知识，只要具有一定的C语言程序设计基础，即可根据KIDI驱动接口快速地开发出适合自己的软件产品。因此KIDI可帮助用户快速开发自己的软件产品，大大降低产品的后续维护工作量。

KIDI架构如图5.1所示：

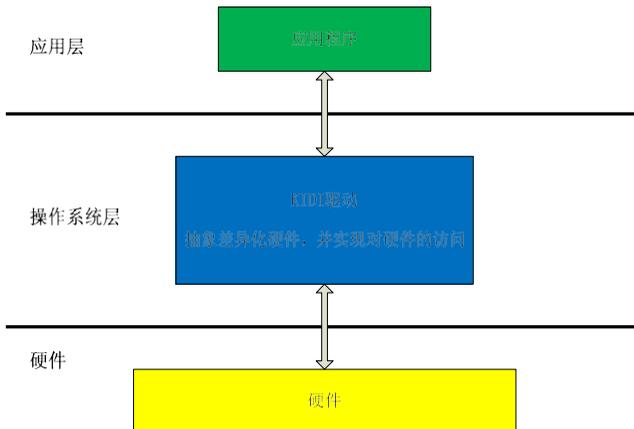


图 5.1 KIDI 架构

## 5.1.2 KIDI 优势

KIDI的初衷就是简化用户对底层硬件的访问，将用户从繁琐的驱动编程中解放出来，大大减少用户软件产品的重复设计和测试工作，帮助用户产品快速上市。因此KIDI软件具有以下优势：

### ★跨平台性

用户使用KIDI开发的软件产品，适用于集特智能所有硬件平台的机型，实现软件与硬件无缝对接。

### ★简单易用

KIDI提供了丰富的，简单易用的硬件访问接口，应用软件工程师可完全不需要任何驱动开发相关方面的知识，只要有一定的C语言程序设计基础，即可根据演示程序快速地开发出适合自己的软件产品。

### ★安全性

KIDI由集特智能研发团队编写，并经过严格测试，可避免因对系统硬件操作不当而造成系统异常问题。

### ★移植性好

针对不同的操作系统，KIDI提供了统一的接口函数，用户在某个版本的操作系统下开发的软件，可轻松移植到其它操作系统上。例如用户在Windows下利用KIDI开发的程序，可轻松地移植到任何Linux系统版本上，用户所要做的修改可能仅仅是更换KIDI库和修改Makefile（可参考KIDI演示程序）。

## ★易维护

KIDI提供给用户使用的所有接口均是高度抽象的，与具体硬件平台无关，因此用户在集特智能某款产品开发的软件（如看门狗，GPIO等）可安全地使用在控道智能其它硬件产品上，用户相同功能的软件产品仅需维护一套，大大节省软件产品的维护成本。

## ★成本低

KIDI简单易用，且具有平台无关性特点，可大大降低软件开发难度和缩短开发周期，可帮助系统集成商产品快速上市。

## 5.2 KIDI 编程接口

本章主要讲述如何使用KIDI，为用户编程提供参考。KIDI提供了丰富的接口函数，能满足用户对主板硬件访问的需求。本手册提供的函数接口具有良好的兼容性、可移植性和跨平台性，适用于集特智能公司目所有硬件产品，操作简单方便，可以大大缩短用户的开发周期，帮助用户产品快速上市。

### A、开发工具

运行环境：Windows XP/Windows Server2003/2008/2012/Win7/Win8/linux

开发工具：

★ Windows系统：Visual Studio2008

★ Linux系统：gcc

### B、需要引用的文件

当您进行程序开发时，需要引用以下文件：

Windows程序开发

→库文件: kidi32.dll, kidi32.lib, kidi64.dll和kidi64.lib;

→驱动文件: kidi32.sys和kidi64.sys;

→函数库头文件: kidi.h;

→kidi32和kidi64文件分别使用在32位和64位操作系统下;

### ★Linux程序开发

→库文件: libkidi32.a 或 libkidi64.a, 分别对应于 32 位和 64 位 Linux 系统;

→函数库头文件: kidi.h;

## C、数据结构定义

→状态返回码定义

KIDI中状态返回码宏定义描述如下:

```
KD_SUCCESS           //操作成功
KD_INVALID_PARAMETER //入口参数错误
KD_UNSUPPORTED       //功能不支持
KD_FAILED            //操作失败
```

→用户可编程GPIO宏定义

KIDI中GPIO功能定义的宏描述如下:

```
KD_GPIO_INPUT0      //对应于说明书中描述的GPIO输入引脚INPUT0
KD_GPIO_INPUT1      //对应于说明书中描述的GPIO输入引脚INPUT1
.....
KD_GPIO_INPUT31     //对应于说明书中描述的GPIO输入引脚INPUT31
```

KD\_GPIO\_OUTPUT0 //对应于说明书中描述的GPIO输出引脚OUTPUT0

KD\_GPIO\_OUTPUT1 //对应于说明书中描述的GPIO输出引脚OUTPUT1

.....

KD\_GPIO\_OUTPUT31 //对应于说明书中描述的GPIO输出引脚OUTPUT31

依次类推KD\_GPIO\_INPUTn和KD\_GPIO\_OUTPUTn分别对应说明书中描述的GPIO输入引脚的INPUTn和输出引脚的OUTPUTn。

→硬件监控宏定义

KD\_SYS1\_TEMP //第一个系统温度，系统温度传感器通常只设计一个

KD\_CPU1\_TEMP //第一个CPU的温度

KD\_CPU2\_TEMP //第二个CPU的温度，多路服务器平台才支持

KD\_CPU3\_TEMP //第三个CPU的温度，多路服务器平台才支持

KD\_CPU4\_TEMP //第四个CPU的温度，多路服务器平台才支持

KD\_SYS\_FAN1 //系统风扇1的转速

KD\_SYS\_FAN2 //系统风扇2的转速

KD\_SYS\_FAN3 //系统风扇3的转速

KD\_SYS\_FAN4 //系统风扇4的转速

KD\_CPU\_FAN1 //CPU风扇1的转速

KD\_CPU\_FAN2 //CPU风扇2的转速

KD\_CPU\_FAN3 //CPU风扇3的转速

KD\_CPU\_FAN4 //CPU风扇4的转速

KD_VOL_VCORE1	//CPU1的核心电压
KD_VOL_VCORE2	//CPU2的核心电压
KD_VOL_VCORE3	//CPU3的核心电压
KD_VOL_VCORE4	//CPU4的核心电压
KD_VOL_V3_3	//开关电源输出的3.3V电压
KD_VOL_V5	//开关电源输出的5.0V电压
KD_VOL_V12	//开关电源输出的12.0V电压

#### D、KIDI函数接口

→kd\_init

函数原型: int kd\_init();

函数功能: 初始化接口, 主要功能包括加载KIDI驱动和分配系统资源, 必须在进程中调用所有功能函数之前调用该函数。

输入参数: 无

输出参数: 无

返回值 : 若初始化成功, 函数返回值为KD\_SUCCESS, 否则初始化失败。

→ kd\_exit

函数原型: int kd\_exit ();

函数功能: 卸载KIDI驱动, 释放系统资源, 在进程结束之前调用。

输入参数: 无

输出参数: 无

返回值：若释放资源成功，函数返回值为KD\_SUCCESS，否则释放资源失败。

→kd\_get\_version

函数原型：unsigned char kd\_get\_version();

函数功能：获取KIDI固件版本。

输入参数：无

输出参数：无

返回值：返回KIDI固件版本。

→kd\_wdt\_start

函数原型：int kd\_wdt\_start(unsigned char timeout\_value);

函数功能：启动硬件看门狗，倒计时至0时系统会重启。

输入参数：timeout\_value - 看门狗硬件倒计时值，单位为秒

输出参数：无

返回值：若启动看门狗成功，函数返回值为KD\_SUCCESS，否则启动看门狗失败。

→kd\_wdt\_feed

函数原型：void kd\_wdt\_feed();

函数功能：对看门狗进行喂狗（以kd\_wdt\_start设置的倒计时值重新初始化看门狗）。从系统的效率和硬件兼容性考虑，不要太频繁喂狗，建议取值在10秒至倒计时值减

5秒之间。

输入参数：无

输出参数：无

返回值：无

→kd\_wdt\_stop

函数原型: void kd\_wdt\_stop();

函数功能: 停止看门狗。

输入参数: 无

输出参数: 无

返回值 : 无

→kd\_wdt\_get\_current\_timeout\_value

函数原型: unsigned char kd\_wdt\_get\_current\_timeout\_value();

函数功能: 获取看门狗当前剩余计数值。一般用于看门狗功能测试，  
目的是方便用户直观了解系统即将复位的时间或喂狗是否成功。

输入参数: 无

输出参数: 无

返回值 : 看门狗当前剩余计数值。

→ kd\_gpio\_get\_num

函数原型: int kd\_gpio\_get\_num();

函数功能: 获取主板上GPIO的个数。

输入参数: 无

输出参数: 无

返回值 : 主板上GPIO的个数。

返回值为0表示当前主板不支持GPIO功能。

→kd\_gpio\_set\_pin\_mode

函数原型: `int kd_gpio_set_pin_mode(unsigned char pin, unsigned char pin_mode);`

函数功能: 设置某个GPIO引脚的模式, 即输入或输出(相对CPU而言)。

输入参数: pin - GPIO引脚序号, 有效值为宏KD\_GPIO\_INPUTn或KD\_GPIO\_OUTPUTn

pin\_mode - GPIO引脚工作模式, 有效值为0: 输出模式, 1: 输入模式

输出参数: 无

返回值 : 若设置成功, 函数返回值为KD\_SUCCESS, 否则设置失败。

→kd\_gpio\_set\_pin\_level

函数原型: `int kd_gpio_set_pin_level(unsigned char pin, unsigned char pin_level);`

函数功能: 设置某个GPIO输出引脚的电平。

输入参数:

pin - GPIO引脚序号, 有效值为宏KD\_GPIO\_INPUTn或KD\_GPIO\_OUTPUTn

pin\_level - GPIO引脚输出电平, 有效值为0: 低电平, 1: 高电平

输出参数: 无

返回值 : 若设置成功, 函数返回值为KD\_SUCCESS, 否则设置失败。

→kd\_gpio\_get\_pin\_level

函数原型: `int kd_gpio_get_pin_level(unsigned char pin, unsigned char *pin_level);`

函数功能：读取某个GPIO 输入引脚的电平。

输入参数：

pin - GPIO引脚序号，有效值为宏KD\_GPIO\_INPUTn或KD\_GPIO\_OUTPUTn

pin\_level - GPIO引脚的输入电平，有效值为0：低电平，1：高电平

输出参数：无

返回值：若读取成功，函数返回值为KD\_SUCCESS，否则设置失败。

→kd\_get\_hwm\_temperature

函数原型：int kd\_get\_hwm\_temperature(unsigned char sensor, unsigned char \*temp);

函数功能：获取主板上温度传感器的温度值。

输入参数：sensor - 温度传感器宏定义（见2.3.3）

输出参数：\*temp - 温度值（单位：摄氏度，有效范围在0到127之间）

返回值：KD\_SUCCESS - 读取传感器温度成功

KD\_UNSUPPORTED - 当前温度传感器不存在

→kd\_get\_hwm\_fan\_speed

函数原型：int kd\_get\_hwm\_fan\_speed(unsigned char sensor, unsigned short \*fan\_speed);

函数功能：获取主板上的风扇转速。

输入参数：sensor - 风扇转速传感器宏定义（见2.3.3）

输出参数: \*fan\_speed - 风扇转速 (单位: 转/分钟)

返回值 : KD\_SUCCESS - 读取风扇转速成功

KD\_UNSUPPORTED - 当前风扇接口不存在

→ kd\_get\_hwm\_voltage

函数原型: int kd\_get\_hwm\_voltage(unsigned char sensor, float \*vol);

函数功能: 获取主板上电源引脚的电压值。

输入参数: sensor - 电压传感器宏定义 (见2.3.3)

输出参数: \*vol - 电压值 (单位: 伏特)

返回值 : KD\_SUCCESS - 读取电压值成功

KD\_UNSUPPORTED - 当前电压传感器不存在

### 5.3 KIDI 函数调用步骤

本章简要介绍 KIDI 函数调用步骤，方便用户快速利用 KIDI 函数开发自己的软件产品，帮助用户产品快速上市。因操作系统的保护机制，硬件访问都需要在特权模式下进行，因此使用 KIDI 编写的软件，需要具有管理员权限才能正确执行。

KIDI 函数调用的通用步骤如下：

- 1、调用 `kd_init` 函数加载 KIDI 驱动，申请系统资源。
- 2、KIDI 驱动首先检查当前主板是否支持 KIDI，如果不支持执行第 5 步，否则执行第 3 步。
- 3、调用 KIDI 其它函数访问主板硬件资源，例如调用 `kd_get_hwm_temperature` 获取主板温度传感器的温度值。
- 4、调用 `kd_exit` 函数卸载 KIDI 驱动，释放系统资源。
- 5、程序结束。

`kidi_init` 和 `kd_exit` 在同一个进程中只需在开始和结束各调用一次，KIDI 驱动调用的通用流程如图 5.2 所示：

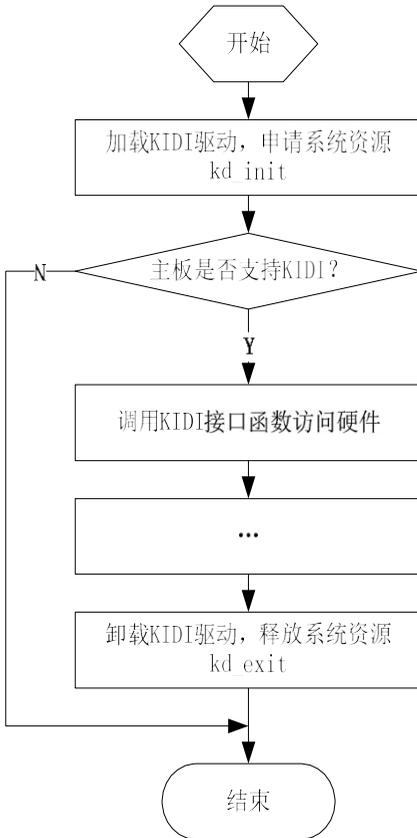


图 5.2 KIDI 驱动接口调用流程图

按功能分类，下面分别详细介绍 KIDI 各功能接口函数的调用步骤，包括硬件看门狗，用户可编程 GPIO，硬件监控等功能。

### 5.3.1 硬件看门狗

看门狗使用步骤如下：

- 1、调用 `kd_init` 函数加载 KIDI 驱动，申请系统资源。
- 2、KIDI 驱动首先检查当前主是否放支持 KIDI，如果不支持执行第 8 步，否则执

行第 3 步。

3、调用 `kd_wdt_start` 函数启动看门狗。

4、等待一定时间间隔，调用 `kd_wdt_feed` 函数进行喂狗。注意：此处时间间隔需要客户自行编程，在 CPU 负载过高的情况下，时间间隔有可能会不准，所以一定要留出充分的余量，建议取值在 10 秒至倒计时值减 5 秒之间。

5、判断是否需要停止看门狗，如果停止看门狗则执行第 6 步，否则循环执行第 4 步。

6、调用 `kd_wdt_stop` 函数停止看门狗。

7、调用 `kd_exit` 函数卸载 KIDI 驱动，释放系统资源。

8、程序结束

看门狗使用的流程图如图 5.3 所示：

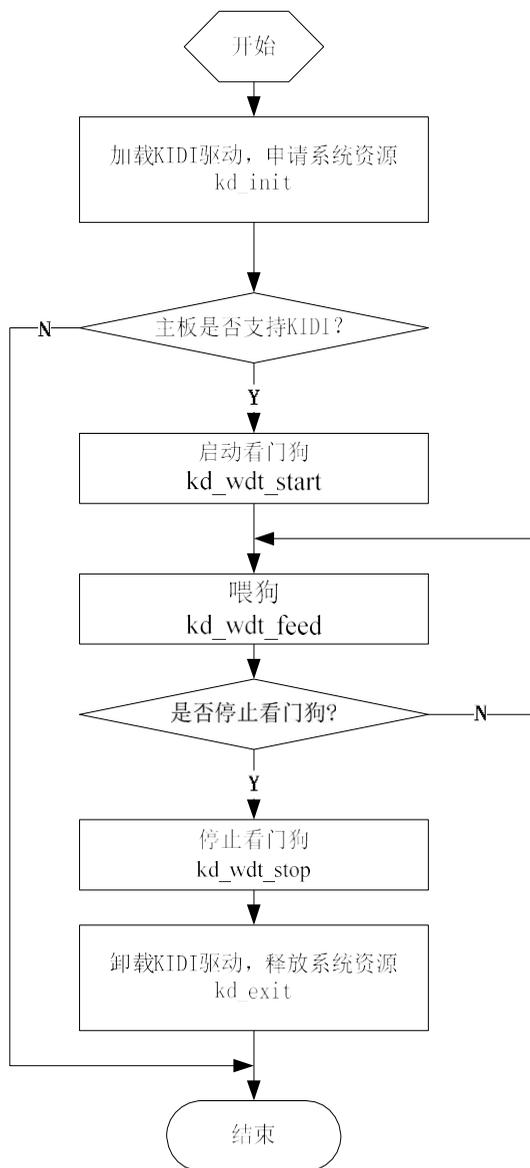


图 5.3 看门狗使用流程图

### 5.3.2 用户可编程 GPIO

用户可编程 GPIO 使用步骤如下：

- 1、调用 `kd_init` 函数加载 KIDI 驱动，申请系统资源。
- 2、KIDI 驱动首先检查当前主板是否支持 KIDI，如果不支持执行第 7 步，否则执行第 3 步。
- 3、调用 `kd_gpio_set_pin_mode` 函数配置各 GPIO 引脚的工作模式，即配置为输出模式还是输入模式，此处输入输出是相对于主板 CPU 而言的。
- 4、配置完各 GPIO 引脚的工作模式后，我们就可以调用 `kd_gpio_set_pin_level` 函数去设置各输出引脚的电平了，对于输入引脚，此函数的操作会被忽略，即不会对硬件产生影响。
- 5、调用 `kd_gpio_get_pin_level` 函数可获取 GPIO 引脚的电平状态，然后根据电平的状态去执行用户的业务操作。
- 6、调用 `kd_exit` 函数卸载 KIDI 驱动，释放系统资源。
- 7、程序结束

用户可编程 GPIO 使用的流程图如图 5.4 所示：

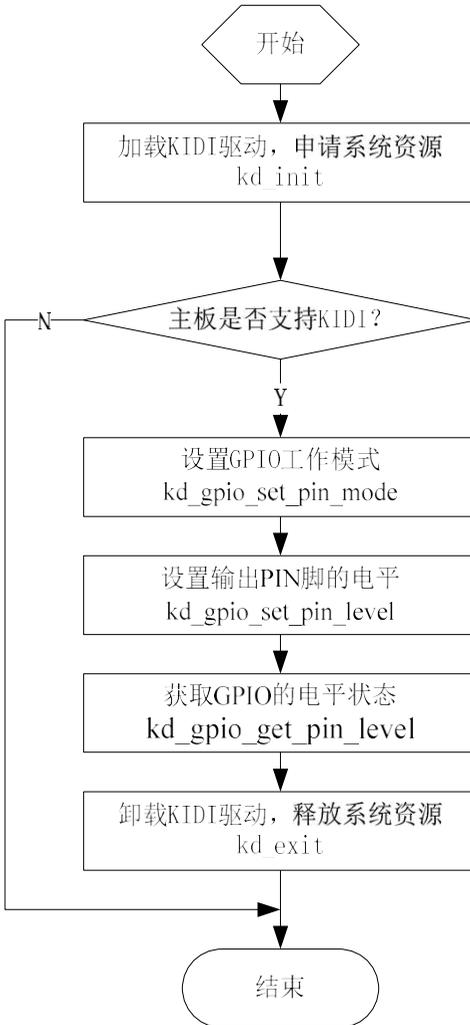


图 5.4 GPIO 使用流程图

### 5.3.3 硬件监控

KIDI 硬件监控各接口函数调用步骤类似，下面仅以温度侦测接口函数为例，介绍 KIDI 硬件监控接口函数的调步骤如下：

- 1、调用 `kd_init` 函数加载 KIDI 驱动，申请系统资源。
- 2、KIDI 驱动首先检查当前主板是否支持 KIDI，如果不支持执行第 5 步，否则执行第 3 步。
- 3、调用 `kd_get_hwm_temperature` 函数获取硬件温度。
- 4、调用 `kd_exit` 函数卸载 KIDI 驱动，释放系统资源。
- 5、程序结束

KIDI 硬件监控接口使用的流程图如图 5.5 所示：

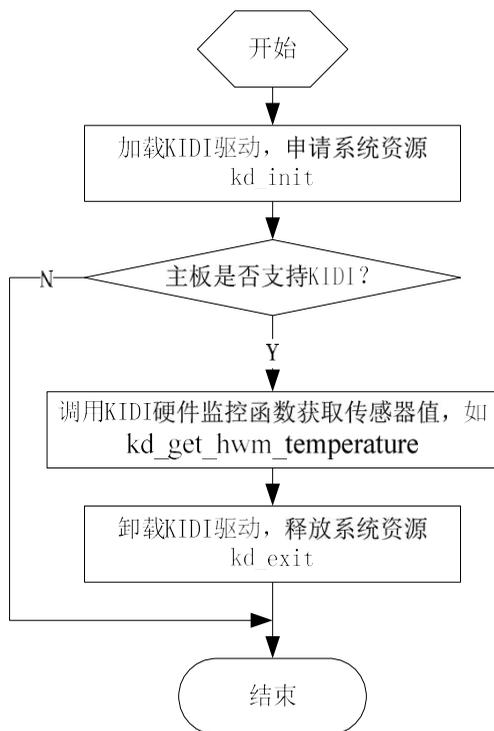


图 5.5 硬件监控接口使用流程图

## 5.4 功能演示

本章简要介绍 KIDI 演示程序的使用，用户可通过 KIDI 演示程序测试控道产品的相关功能，体验 KIDI 编程的简单及易用性。

大部分操作系统下硬件访问是受保护的，因此请以管理员权限运行 KIDI 产品中的演示程序，因 KIDI 产品中 Windows 和 Linux 的演示程序功能及各参数均一致，且使用非常简单，使用前，请先留意操作系统的版本，分别正确调用 32 位或 64 位的演示程序。下面仅介绍 32 位 linux 系统下 KIDI 演示程序，

### 5.4.1 看门狗测试

看门狗演示程序使用说明如下：

#### A、启动看门狗

测试硬件看门狗是否正常，验证系统是否在规定时间内复位，默认时间为 30 秒，命令为 `wdt_demo32 --start`，如下图所示：

```
[root@localhost kidi_src_for_linux_v1.0]# ./wdt_demo32
```

```
-----  
Kondoct WDT Test Utility for Linux
```

```
Version: 1.0
```

```
Author : Kondoct BIOS Team  
-----
```

```
Kidi Init Ok!
```

```
Kidi Version: 10
```

```
Start wdt passed!
```

```
system will reset after 0 seconds ...
```

```
system will reset after 0 seconds ...
```

```
system will reset after 30 seconds ...
```

B、喂狗

C、停止看门狗

```
[root@localhost kidi_src_for_linux_v1.0]#
[root@localhost kidi_src_for_linux_v1.0]#
[root@localhost kidi_src_for_linux_v1.0]# ./wdt_demo32 --stop
-----
Kondoct WDT Test Utility for Linux
Version: 1.0
Author : Kondoct BIOS Team
-----
Kidi Init Ok!
Kidi Version: 10
Stop wdt passed!
[root@localhost kidi_src_for_linux_v1.0]# █
```

## 5.4.2 GPIO 测试

GPIO 演示程序使用说明如下：

- A、将所有 GPIO 配置为输出模式，并输出低电平，命令为./gpio\_demo32 --low，然后用户可用万用表测量 GPIO 引脚的电平，验证 GPIO 功能是否正常。

```
[root@localhost kidi_src_for_linux_v1.0]#
[root@localhost kidi_src_for_linux_v1.0]# ./gpio_demo32 --low
-----
Kondoct GPIO Test Utility for Linux
Version: 1.0
Author : Kondoct BIOS Team
-----
Kidi Init Ok!
Kidi Version: 10
GPIO Count: 8
set all gpio pins output a low level, passed!
INPTU3 Pin Level: 0
[root@localhost kidi_src_for_linux_v1.0]# █
```

- B、将所有 GPIO 配置为输出模式，并输出高电平，参数为./gpio\_demo32 --high，然后用户可用万用表测量 GPIO 引脚电平，验证 GPIO 功能是否正常。

```
[root@localhost kidi_src_for_linux_v1.0]#  
[root@localhost kidi_src_for_linux_v1.0]# ./gpio_demo3  
-----  
Koooot GPIO Test Utility for Linux
```

### 5.4.3 硬件监控测试

硬件监控演示默认是演示主板上所有支持的传感器值，命令为./hwm\_demo32，每隔2秒刷新一次，如下图所示：

```
[root@localhost kidi_src_for_linux_v1.0]#  
[root@localhost kidi_src_for_linux_v1.0]# ./hwm_demo32
```

```
-----  
Konduct Hardware Monitor Test Utility for Linux  
Version: 1.0  
Author : Konduct BIOS Team  
-----
```

```
Kidi Init Ok!  
Kidi Version: 10  
system temperature1: 32  
cpu1 temperature   : 37  
cpu2 temperature   : unsupported  
cpu3 temperature   : unsupported  
cpu4 temperature   : unsupported
```

```
system fan1 speed  : N/A  
system fan2 speed  : unsupported  
system fan3 speed  : unsupported  
system fan4 speed  : unsupported  
cpu fan1 speed     : 5000  
cpu fan2 speed     : unsupported  
cpu fan3 speed     : unsupported  
cpu fan4 speed     : unsupported
```

```
cpu1 vcore         : 0.81  
cpu2 vcore         : unsupported  
cpu3 vcore         : unsupported  
cpu4 vcore         : unsupported  
V3.3              : 3.39  
V5                : 5.09  
V12               : 12.14
```

```
system temperature1: 32  
cpu1 temperature   : 37  
cpu2 temperature   : unsupported  
cpu3 temperature   : unsupported  
cpu4 temperature   : unsupported
```

N/A : 表示主板支持系统风扇1  
的侦测, 但此接口没有接风扇

```
system fan1 speed  : N/A  
system fan2 speed  : unsupported  
system fan3 speed  : unsupported  
system fan4 speed  : unsupported  
cpu fan1 speed     : 2137  
cpu fan2 speed     : unsupported  
cpu fan3 speed     : unsupported  
cpu fan4 speed     : unsupported
```

## 5.5 产品订购信息

请详见随机出货的产品配置订购信息或整机配套光盘，在此不做介绍。

## 5.6 缩略语

缩略语	述语	含义
AC	交流	交流
APM	高级电源管理	用于监视和降低 PC 功耗的工具
ACPI	高级配置与电源接口	
AHCI	高级主控接口/高级主机控制器接口	串行 ATA 的标准控制接口, Microsoft windows XP (高于 SP1 版本) 和 IAA 驱动程序支持该接口
APIC	高级可编程中断控制器	扩张的可编程中断控制器
ASCII	美国信息交换标准代码	
API	应用程序编程接口	
ATM	异步传输模式	
AT	高级技术	AT 电源
ATX	高级技术扩展	ATX 主板结构或 ATX 电源
BIOS	基本输入输出系统	BIOS 代码
bps	位/秒	数据传输速率的一种描述
BGA	球状矩阵排列	一种芯片封装形式
Buffer	缓冲器	
Battery	电池	
CD-ROM	只读光盘	大数据存储只读光盘
CD-RW	可读写光盘	刻录光盘
CE	欧洲共同体 (CE 认证符号)	欧盟产品的统一认证标示
CF	CF 卡	
CGA	彩色图形适配器	标准监视器接口
CLK	时钟脉冲	时钟信号
CMOS	互补金属氧化物半导体	

COM	串行通信端口	串行接口
CP	通信处理器	通信计算机
CPU	中央处理单元	
CRT	阴极射线管	CRT 显示器
CTS	清除发送	
Cache	高速缓冲存储器	
DRAM	动态随机存取存储器	
DDRAM	双数据随机存取存储器	带有高速接口的存储器芯片
DC	直流	
DCD	数据载波检测	数据载波信号检测
DMA	直接存储器存取	直接内存存取
DOS	磁盘操作系统	无图形界面的操作系统
DP	显示端口	Display Port
DQS	德国质量与环境管理体系认证机构	
DSR	数据设置就绪	操作就绪
DTR	数据终端就绪	
DVD	数字多功能光盘	
DVI-D	数字视频接口	数字显示器接口
DVI-I	数字视频接口	具备数字和 VGA 模拟信号的显示器接口
dB	分贝	一种纯计数单位, 表示两个量的比值大小
DCE	数据通信设备	数据通信终端设备
DOM	DOM 电子硬盘	Disk on module 的缩写
DOC	单芯片快闪磁盘	Disk on chip 的缩写
DDC	显示数据通道	显示器和主机通信总线标准
DDR	双倍数据速率	一种内存规范
Decode	指令解码	
DFP	数字平面显示器	
DHCP	动态主机设置协议	一种局域网的网络协议

DES	数据加密算法	一种对称加密算法
DIMM	双列直插式存储模块	
DMI	直接媒体接口	处理器 CPU 和 IO 控制器 (PCH, ICH) 间的数据传输通道
DVMT	动态分配共享显存技术	用于设定显卡可使用的共享内存大小的技术
ECC	错误检查修正	
ECP	扩展的功能端口	扩展的并行端口
EGA	增强型图形适配器	PC 和监视器的接口
ESD	静电释放	
EDID	扩展显示标识数据	一种 VESA 标准数据格式
EIDE	增强型 IDE	增强形电子集成驱动器
EISA	扩展工业标准体系结构	扩展的 ISA 标准
EM64T	64 位内存扩展技术	
EN	欧洲标准	成员国的国家标准必须与 EN 标准保持一致
EEPROM	电可擦可编程只读存储器	使用 EEPROM 芯片的子模块
EPP	增强型并行端口	
EMI	电磁干扰	
EMC	电磁兼容	
ES	能源之星	
ESCD	可扩展系统配置数据	
EIA	电子工业协会	
FAQ	常见问题解答	
FAT32	32 位文件分配表	
FDD	软盘驱动器	
FSB	前端总线	外部总线
Firmware	固件	固化的软件
FWH	固件中心	

FPU	浮点运算单元	
FTP	文件传输协议	
GND	接地	
GB	千兆位	
GUI	图形用户界面	
GMCH	图形和内存控制中心	
GPU	图形处理器	
HDD	硬盘驱动器	
HDTV	高清晰度电视	
HEX	十六进制	
HTML	超文本标记语言	用于创建 Internet 页面的脚本语言
HTTP	超文本传输协议	Internet 上的数据传输协议
Hardware	硬件	
HW Monitor	硬件监控	
Hz	赫兹	
HDMI	高清晰度多媒体接口	一种高清晰显示接口
HDA	高保真音频	
I/O	输入输出	计算机的数据输入/输出
IDE	电子集成驱动器	
IGD	集成的图形设备	
IP	防护等级	
IP	Internet 通信协议	
IR	红外线	一种低速近距离的无线传输
IRDA	红外线数据协会	用于通过 IR 模块传输数据的标准
IRQ	中断请求	
ISA	工业标准体系结构	用于扩展模块的总线

IPC	工业控制计算机	
ISO	国际标准化组织	
IT	信息技术	
IA	英特尔架构	
ICH	输入/输出控制中心	
ID	身份标识号码	
INI File	初始化文件	
IPMI	智能平台管理接口	
JEDEC	固态技术协会	存储设备的标准规范制定协会
KB	键盘	
Kbps	千位每秒钟	传输速率的一种
L2 Cache	二级缓存	
LAN	局域网	局限于本地通讯的计算机网络
LCD	液晶显示器	
LED	发光二极管	
LPT	打印机端口	
LVDS	低电压差分信号	
LBA	逻辑块寻址	一种机械硬盘的块寻址模式
LFP	LCD 平板	LCD 平板显示器
MAC	介质访问控制	
MMC	微型存储卡	32*24.5mm 格式的存储卡
MPI	用于编程设备的多点接口	
MS-DOS	微软磁盘操作系统	
MTBF	平均无故障间隔时间	
MB	兆字节	
MPEG	动态影像压缩标准	一种视频的压缩标准
MTTR	平均修复时间	
MBR	主引导记录	

MHz	兆赫兹	一种频率的单位
MCU	微控制单元	单片微型计算机的一种
MODEM	调制解调器	
NA	空	
NC	未连接	
NP	未安装	
NCQ	原生命令队列	自动将文件和磁盘存取重新排列, 以提高性能
NMI	不可屏蔽中断	
NTFS	新技术文件系统	Windows XP、Windows Server 2008、
NIC	网络接口卡	网络适配器
OS	操作系统	
OSD	屏幕显示菜单	
PATA	并行 ATA	存储接口标准的一种
PC	个人电脑	个人计算机
PCI	外围设备互连	PCI 总线
PCI-E	Peripheral Component Interconnect Express	一种高速串行差分全双工的总线传输规范
PCMCIA	个人计算机存储卡国际协会	
PEG	PCI EXPRESS 图形	
POST	开机自检	
PXE	预引导执行环境	用于通过网络运行没有硬盘数据 PC 的软件
PS2	PS2 接口设备	IBM 开发的 PS2 鼠标键盘标准接口形式
PICMG	PCI 工业计算机制造商协会	
RAID	独立磁盘冗余阵列	冗余硬盘阵列
RI	振铃输入	呼入唤醒信号
RS485	协调子层 485	设计用于多达 32 个节点的双向总线系统

RTC	实时时钟	
RTS	发送数据请求	
RXD	接收数据	
RF	射频	频率范围从 300KHz~30GHz 之间的无线发射电磁波
RSV	保留使用引脚	
RST	复位	Reset
SCSI	小型计算机系统接口	
SDK	软件开发工具包	
SSD	固态电子盘	
SV	标准电压	
SMART	自我检视, 分析和报告技术	硬盘错误诊断程序
SO-DIMM	小外形双列内存模组	笔记本内存的一种
SRAM	静态随机存取存储器	
SDRAM	同步动态随机存储器	
SVGA	超级视频图形阵列	
SW	软件	Software
S5	关机	所有硬件设备（包括电源）全部都关闭
S4	挂起到硬盘	内存信息写入硬盘，然后所有部件停止工作
S3	挂起到内存	将运行中的数据写入内存后关闭硬盘
S1	POS (Power on Suspend)	CPU 停止工作, 其他的硬件设备仍然正常工作
S0	电脑正常工作	所有硬件设备全部处于打开或正常工作的状态
SMB	全系统管理总线	
SMD	表面安装设备	

SMI	系统管理中断	只有触发 SMI 才能进入 SMM 模式
SMM	系统管理模式	
SPD	串行存在探测	一个 256KB 的 EEPROM
S/PDIF	索尼/飞利浦数字接口	一种民用数字音频接口协议
SCI	系统控制中断	硬件产生的通知 OS 的 ACPI 中断事件,
TFT	薄膜晶体管	LCD 平面屏幕类型
TxD	发送数据	数据传送信号
TDP	热量设计功耗	Thermal Design Power
TTL	晶体管 to 晶体管逻辑电路	
Turbo	瞬间加速自动超频技术	英特尔睿频加速技术 Turbo boost
TPM	可信平台模组	
USB	通用串行总线	
UPS	不间断电源	
UL	美国保险商实验室	UL 认证
U	服务器尺寸单位, 1U=44.45mm	
UEFI	统一的可扩展固件接口	Intel 全新固件接口标准, 采用 C 语言
VGA	视频图形阵列	满足工业标准的视频适配器
VRM	电压调整模块	
VT	虚拟化技术	通过 Internet 技术可以使用模拟封闭环境
VID	电压识别定义	
VSYNC	垂直同步	垂直同步刷新
VESA	视频电子标准协会	
VRAM	视讯随机存取存储器	
VIO	VIO 电压	PCI 总线设备上的 IO 电压
WDT	看门狗	Watch dog
WLAN	无线局域网	

WOL	网络唤醒	局域网唤醒
XGA	扩展图形阵列	最大分辨率支持 1024*768