



# PCM3-J1900

Rev A2

## 用户手册

## 版本

2017年6月

修订：A2

## 版权保护及声明

本手册为深圳灵江计算机技术有限公司的知识产权，内容受版权保护，版权所有。未经认可，不得以机械的，电子的或其它任何方式进行复制。除列明随产品配置的配件外，本手册包含的内容并不代表本公司的承诺，我们很小的编写此手册，但我们对于本手册的内容不保证完全正确，因为我们的产品一直在持续的改良及更新，故我方保留随时做出修改而不予另行通知的权利。对于任何安装、使用不当而导致的直接的、间接、有意或无意的损坏及隐患概不负责。您在订购产品前，请详细了解产品性能是否符合您的需求。

## 商标

本手册使用的所有商标均属于各自的商标持有者所有：

Intel和Celeron是Intel的注册商标

Win7和Windows XP是Microsoft的注册商标

---

## 装箱物品检查

请确认您所购买的主板包装盒是否完整，如果包装有所损坏、或是有任何配件欠缺的情形，请尽快与我们联系。

- 1块PCM3-J1900主板
- 1组串口转接电缆
- 1条SATA电缆
- 1条SATA-PWOR电缆
- 1个短路开关

## 订购信息

型号	描述
PCM3-N2930	板载超低功耗 Celeron J1900 CPU/5USB/SATA/2LAN AUDIO/2MPCIE/MSATA/HDMI/VGA/GPIO/4COM 双24 Bit LVDS

欲知更多信息欢迎访问：<http://www.ling-jiang.com>

---

---

1. 产品介绍.....	1
1. 1 简介.....	1
1. 2 环境与机械尺寸.....	6
2. 主板构造图.....	5
2. 1 功能接口标识描述.....	5
3. 主板安装.....	7
3. 1 安全指导.....	7
3. 2 系统内存的安装.....	8
3. 3 跳线设置.....	9
4. 板载接头和接口.....	14
5. 主板控制按钮、状态指示.....	23
6. GPIO 定义/编程指引.....	25
7. Watchdog 看门狗编程指引.....	26
8. BIOS 设置.....	28

---

# 用户手册

---

## 1. 产品介绍

---

PCM3-J1900基于Intel® Bay Trail架构的多核系列应用处理器,具有超强的图形处理能力及应用计算能力,是一款低功耗、高性能、结构紧凑的高清播放主板,采用Intel® Bay trail SOC芯片组,支持DDRIII 1333/1600MHz内存,最大可支持8GB,内部集成Intel® HD Graphics核心显卡,提供高性能的显示输出,支持VGA\HDMI\LVDS显示输出,同时使用可实现独立双显示,集成2个RTL千兆网卡,2个MPCIE插槽支持 WIFI/3G/4G 模块,1个MSATA可用SSD。PCM3-J1900具有多种双显示方式,以高性价比高性能的表现可广泛应用于媒体播放、广告、LCD大屏、通讯控制、医疗仪器、工业控制、交通控制、信息系统、金融设备、自动售票系统、汽车、军工和各种终端机市场领域。

Pcm3-J1900主板是款采用板载Intel Celeron J1900基于Bay Trail芯片集设计的高性能、高可靠工业主板,主要特点如下:

- ❖ 提供板载Celeron J1900 处理器
- ❖ 提供1条SODIMM DDR3L内存插槽,主板板载2/4GB内存,最大支持8GB内存
- ❖ 内建图形加速控制器 采用Intel R HD Graphics分配显存(最大达512MB)支持CRT、HDMI、LVDS同屏/分屏显示输出。
- ❖ 支持4个标准COMRS232端口,(1xRS485/232、3xRS232),1个USB 3.0,4个USB 2.0 1组4进4出8Bit GPIO。
- ❖ 2个10M\100M\1000Mbps网络接口,支持网络引导启动、网络唤醒。
- ❖ 一组专用MIC输入\音频输出接口,2组功放喇叭插座、1组PS2鼠标\键盘排座。
- ❖ 2个M\_PCIE扩展槽带SIM卡座。
- ❖ 1组SATA,1个MSATA扩展槽,
- ❖ 支持来电开机、4线/5线/8线电阻触摸屏(AMT技术)。

## 用户手册

---

### 处理器 (CPU)

集成 Intel® J1900 四核 22nm 处理器 2.0 GHz ,可兼容 Intel® Bay Trail 全系列 CPU , 可供客户灵活选择。

### 芯片组 (Chipset)

采用 Intel® Bay trail J1900 芯片组

### 内存 (DIMM)

提供 1 条 SODIMM DDR3L 内存扩展槽, **主板板载 2/4GB 内存, 最大支持 8GB 内存**

### 显示功能

Intel HD **Graphics** 显示芯片, 显存最高达 512 MHz  
VGA+HDMI 高清显示  
单/双通道 LVDS (18 bit/ 24 bit) 显示输出。

### 存储功能

1 个 SATA II, 1 个 MSATA 扩展槽

### 网络功能 (LAN)

2 个 1000Mbps 以太网控制器 (RTL8111G)  
支持网络引导启动、WOL 功能

### 音频功能 (AUDIO)

板载 ALC 662 芯片 MIC\_in、Line\_out、Line\_in  
功放 2\*2W

### USB 功能

1 个 USB 3.0 高速接口 支持 4.8GB/S  
4 个 USB 2.0 高速接口, 支持 480Mb/s 传输率

### I/O 功能

4 个标准 RS-232 串口 , ( COM1 : Rs485/232 )  
1 个 PS2 键盘和鼠标接口  
8 位 GIO 插针  
256 级看门狗  
2 组功放插座  
1 组 AUDIO 排座  
触摸屏接口: 支持 8 线/5 线/4 线电阻屏  
液晶屏 LED 灯管座

款款

# 用户手册

---

## 1. 2 环境与机械尺寸

---

◆ 工作环境:

温度:  $-40^{\circ}\text{C}\sim 80^{\circ}\text{C}$  ;

湿度: 5%~95% (非凝结状态);

◆ 储存环境:

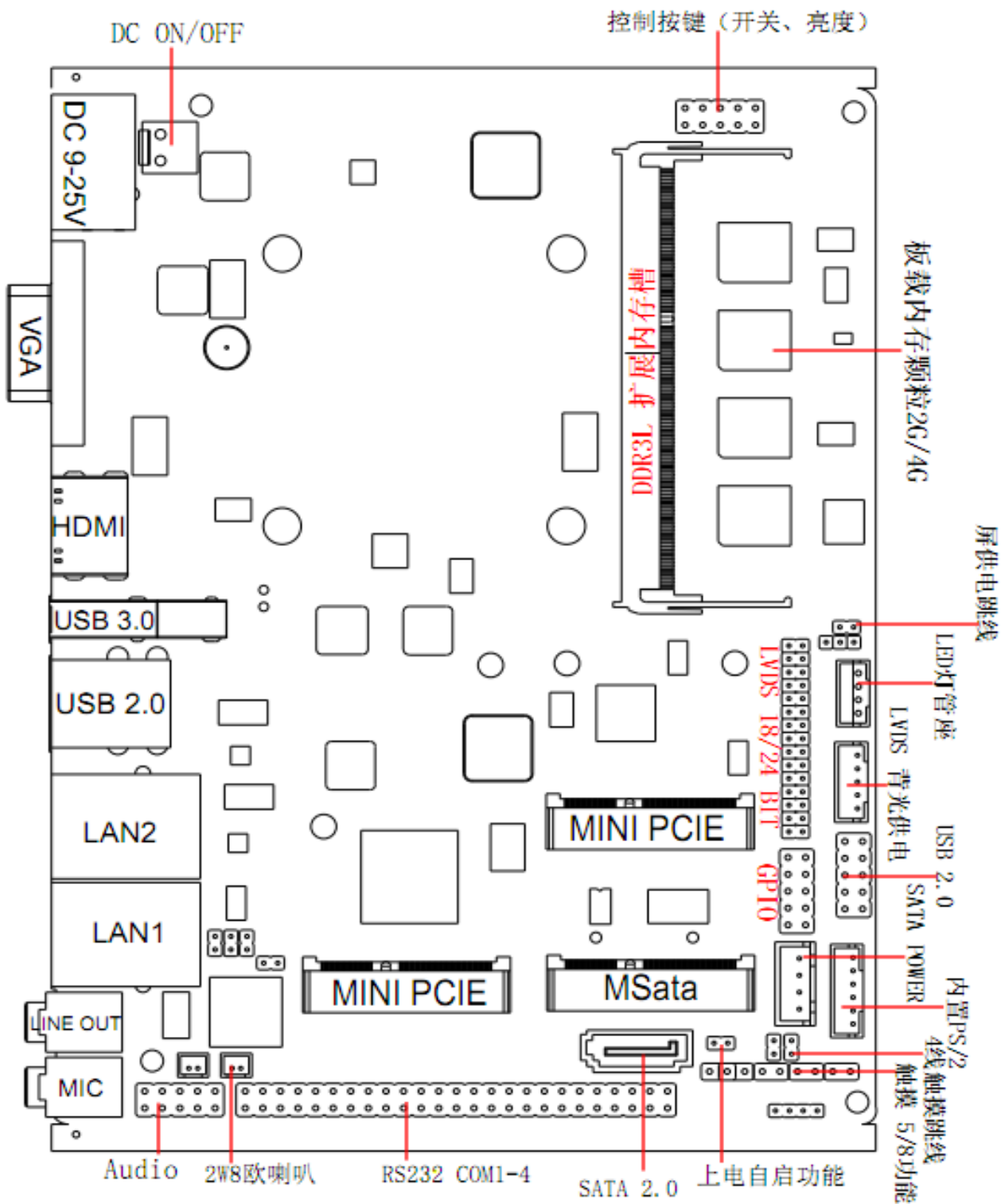
温度:  $-40^{\circ}\text{C}\sim 80^{\circ}\text{C}$  ;

湿度: 5%~95% (非凝结状态);

◆ 尺寸:

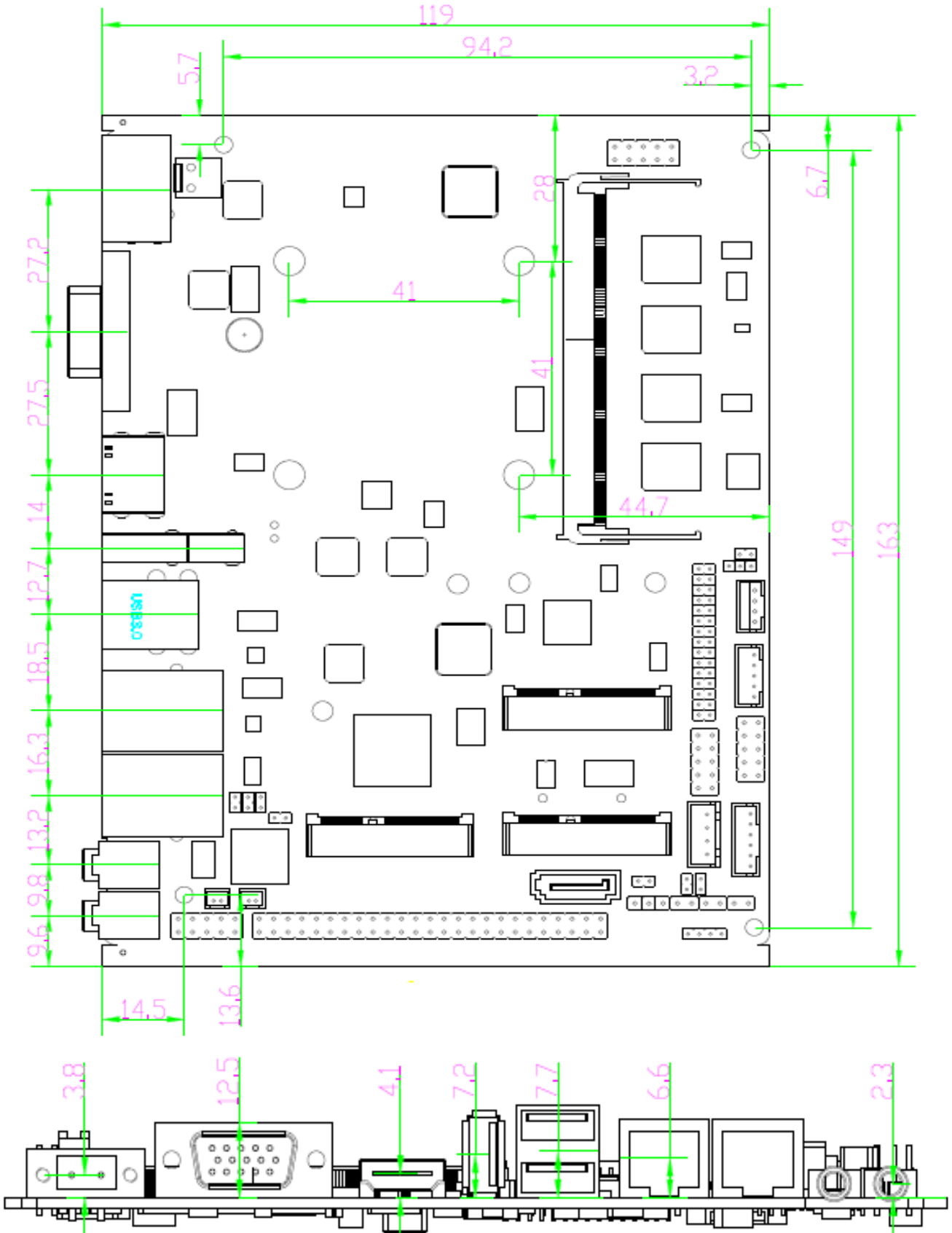
158mm × 119mm (6.5" × 4.6")

2. 主板构造图功能接口标识描述





主板端口尺寸图



### 3. 主板安装

---

---

#### 3.1 安全指导

---

- 1) 请仔细阅读本安全指导，并留意设备及手册上注明的所有注意事项和警告事项
- 2) 请妥善保管使用手册以备将来参考
- 3) 请保持本设备的干燥使其远离潮湿环境
- 4) 在将本设备与电源连接前请确认电源电压值并正确地针对9V ~ 25V电压做出调整
- 5) 请将电源线置于不会被践踏到的地方并且不要在电源线上堆置任何物件设备要有良好的接电线，避免静电损坏，进行安装前，请先断开电源，否则会损坏主板
- 6) 为了避免主板上的元件受到静电的损坏，绝不要把主板直接放到地毯等类似的地方，也要记住在接触主板前使用一个静电手腕带或接触金属通过边缘拿住整块主板安装，切勿接触芯片
- 7) 插拔任何扩展卡或内存模块前请将DC电源断开。
- 8) 不得将任何液体自开口处注入否则会产生严重损坏甚至导致电击
- 9) 如果发生以下情况请找技术服务人员处理：
  - ✧ 电源线 或插头损坏
  - ✧ 液体渗入设备内
  - ✧ 设备暴露在潮湿的环境中
  - ✧ 设备工作不正常或用户不能按照使用手册的指导使其正常工作
  - ✧ 设备跌落或受创，有明显的破损迹象



**注意：**如果BIOS电池换置不当会产生爆炸的危险请务必使用同一型号或者相当类型的且为制造商推荐的电池。

## 用户手册

---

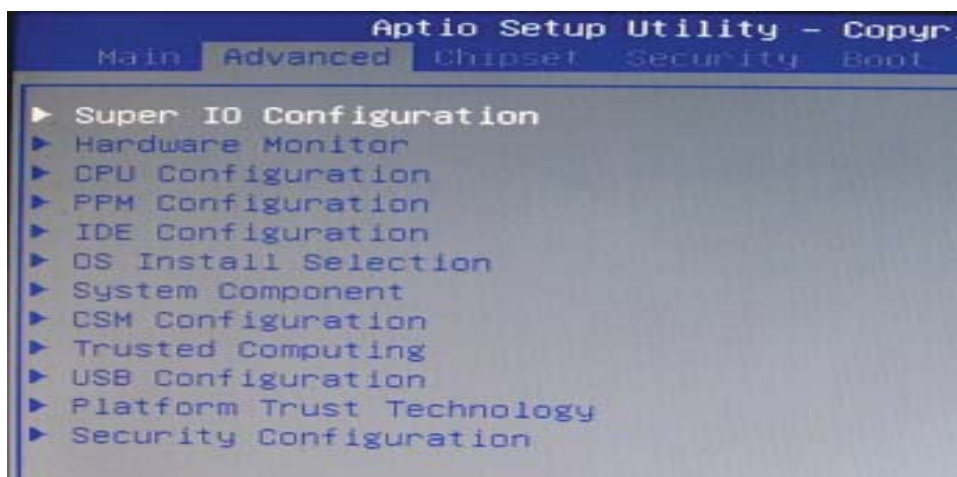
### 3. 2 系统内存的安装

---

主板提供1条 SO-DIMM 的 DDR3L内存插槽，选择安装内存条时，要注意以下几点：

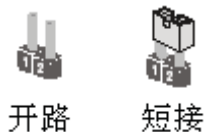
- 安装时，先对准DDR3L SDRAM 存储条与 SO-DIMM 插槽的缺口，用力插到位，再将 DDR3L SDRAM条向下并扣入 SO-DIMM 插槽，使 SO-DIMM 插槽两侧的手柄扣紧并锁住DDR3L SDRAM存储条
- 支持符合1066/1333MHz 规范的DDR3L SO-DIMM内存条
  
- 主板支持系统有 Windows7 Windows8 32/64BIT、Android  
安装以上几个不同版本的系统，需要进入BIOS设置

Advanced->OS Install Selection->OS Selection(Windows7 Windows8 Android)



## 3. 3 跳线设置

插图所示 JCC1 跳线方法。将跳线帽放置在针脚上时为“短接”；当针脚上未放置跳线帽时，此为“开路”。



### 插针 (JBATE)

### CMOS 状态设定

瞬间短接	清除CMOS
开路	正常状态 (默认设置)

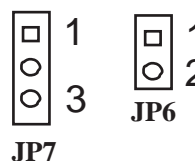


**注意：**清除 CMOS (瞬间短接 JCC1 位接针) 允许您清除 COMS 里的资料，重置系统参数到默认设置。在 COMS 里的资料包括系统设置资讯，例如系统密码，日期，时间及系统设置参数。您在执行此功能操作前，请先关闭电脑并拔掉电源线，等待十五秒钟之后，用跳线帽瞬间短接 JCC1 位接针。

### 插针 (JP7)

### LCD 屏的工作电压设定

1-2 短接	5 V( 默认设置)
2-3 短接	3.3 V
JP6短接	12V

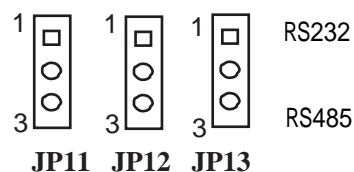


**注意：**在使用 LCD 屏前，请先了解其要求的工作电压，再通过改变 JP6/7插针的跳线帽状态来选择 LCD 屏的工作电压，以确保 LCD 屏稳定工作。

### 插针 ( JP11 JP12 JP13 )

### RS485跳线

1-2 短接	RS 232( 默认设置)
2-3 短接	RS 485

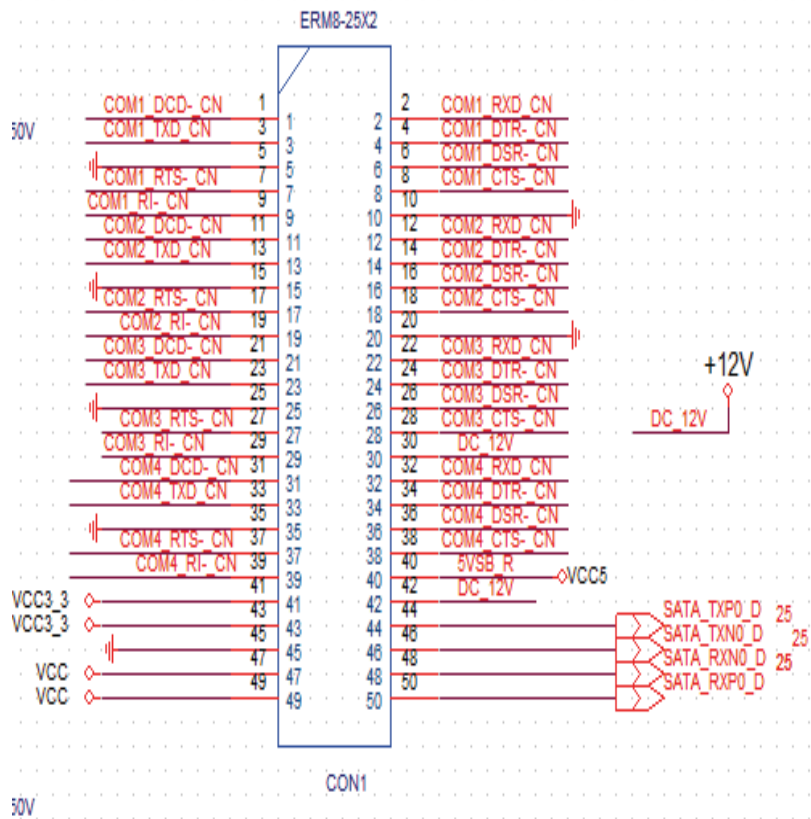


## 4.COM排针定义



板载接头和接口不是跳线，切勿将跳线帽放置在这些接头和接口上，将跳线帽放置接头和接口上将会导致主板的永久性损坏！

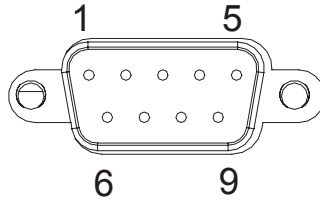
### 串口排针定义 (COM1-4)



主板提供 4 个可用的 RS232 标准串口，客户使用时可以通过配置的专用 COM 线转换出来标准 DB9 端口，可以使用标准 COM 口设备。

## 用户手册

### COM口定义 (DB9定义)

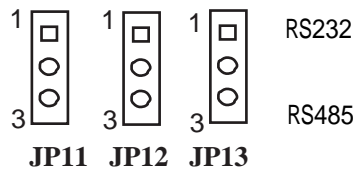


PIN	SIGNAL	DESCRIPTION
1	DCD	Data Carrier Detect
2	RXD	Receive Data TXD+ (A)
3	TXD	Transmit Data TXD- (B)
4	DTR	Data Terminal Ready
5	GND	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	VCC_COM	Voltage output, voltage select setting by J1

**COM1: RS485 2 (TXD+) A, 3 (TXD-) B**

1-2 短接                      R S 2 3 2(默认设置)

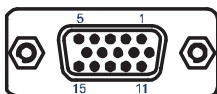
2-3 短接                      R S 4 8 5



## 用户手册

### 显示输出（VGA）接口

（标准的 DB15 显示输出接口 VGA1）



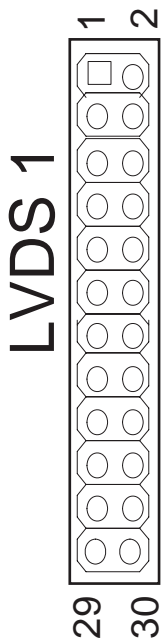
Signal Name	Pin	Pin	Signal Name
Red	1	2	GREEN
Blue	3	4	NC
GND	5	6	GND
GND	7	8	GND
5V	9	10	NC
NC	11	12	SDA
HSYNC	13	14	VSYNC
SCL			

主板提供一个标准的 DB15 显示接口，用户可直接连接显示设备使用。

### LVDS 显示输出接口

(30 针 LVDS1)

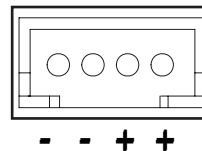
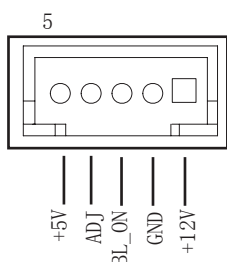
下面给出了单/双通道 LVDS (18-bit/24-bit) 接口定义：



管脚	信号名称	管脚	信号名称
1	VCC	2	VCC
3	VCC	4	GND
5	GND	6	GND
7	A_0-	8	A_0+
9	A_1-	10	A_1+
11	A_2-	12	A_2+
13	GND	14	GND
15	A_CLK-	16	A_CLK+
17	A_3-	18	A_3+
19	B_0-	20	B_0+
21	B_1-	22	B_1+
23	B_2-	24	B_2+
25	GND	26	GND
27	B_CLK-	28	B_CLK+
29	B_3-	30	B_3+

## LCD 背光接口 (INVERTER)

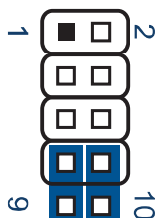
## 液晶屏LED灯管座



用户可根据需要选择使用此接口，用来连接 LCD 背光设备。

## AUDIO(音频)插座

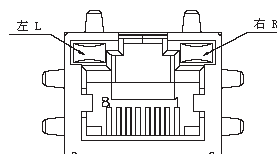
(2\*5 针 音频接针 AUDIO1 )



Jumper	Setting	Status
	1	LIN_R
	2	MIC1_JD
	3	LIN_L
	4	MIC_L
	5	MIC_R
	7	FRONT_JD
	8	LOUT_L
	9	LOUT_R
	6-10	GND

## 网络接口

(标准的 RJ-45 网络输入接口 LAN1)



主板提供一个标准的 10/100/1000Mbps RJ-45 以太网接口，用户可直接插上网络电缆便可使用。

RJ-45 以太网接口两侧有两盏状态指示灯：当右灯常亮时表示以太网处于链接状态；当左灯闪烁时表示网络处于活动状态。



## 用户手册

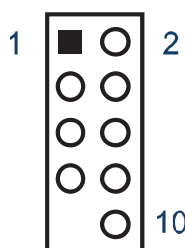
### USB2.0 标准接口

(标准的 USB Ver2.0 接口 USB 1/3 )

主板提供4个 USB Ver2.0 标准接口, 用户可直接连接标准的 USB 设备使用。

### USB 2.0 针座

(9 针F\_USB2 F\_USB3)

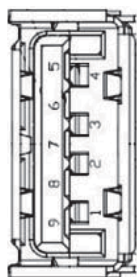


Pin	Signal Name	Pin	Signal Name
1	+5V	2	+5V
3	USB5-	4	USB6-
5	USB5+	6	USB6+
7	GND	8	GND
10	NC		

主板还提供两组可用的 USB 2.0 接针。如果后背板上的 USB 2.0 接口不够用, 使用这个 USB 2.0 接针可以通过 USB 专用转接电缆用来支持 4 个额外的 USB

### USB3.0 标准接口定义

(标准的 USB Ver3.0 接口 USB 2 )



Pin	Signal	Pin	Signal
1	+5V	2	USB_data-
3	USB_data+	4	GND
5	SSRX-	6	SSRX+
7	GND	8	SSTX-
9	SSTX+		

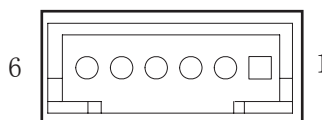
## 内接PS2

(6 针 KB1)

主板提供一个6 Pin白色边框的内接PS2接口 (KB/MS2)，可通过连接电缆连接使用。

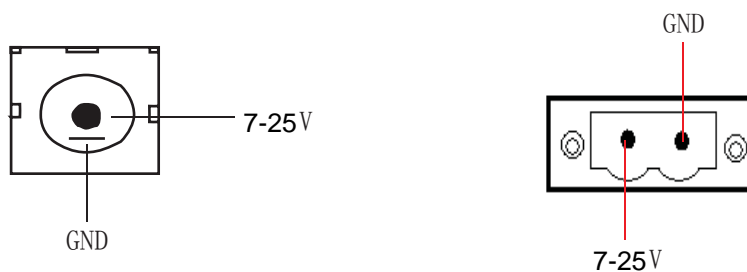
下面给出了 KB/MS2 插座定义：

管脚	信号名称
1	Keyboard data
2	GND
3	Mouse data
4	Keyboard clock
5	5V
6	Mouse clock



## 电源插座

a. DC电源插座 ( DC )



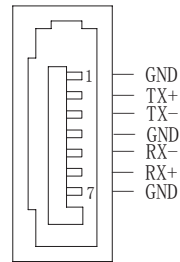
**注意：** 在连接电源前，请先确认DC定义是否正确，否则有可能损坏主板和电源。

# 用户手册

## SATA 插座 (SATA1、2)

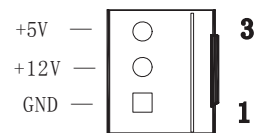
主板提供 2 个标准的 7Pin SATA II 接口,可使用标准 7 芯 SATA 转接电缆连接具有 SATA 接口的存储设备, (SATA3 为可选择使用端口)

SATA 接口定义:

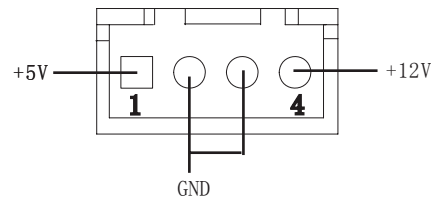


## 风扇接头 (FAN1)

用于连接 CPU、SYSTEM 风扇让黑线与地的接针脚相接。主板上的风扇接头同时也可兼容原来标准的 3 针风扇。



## SATA电源座 (sataPow)



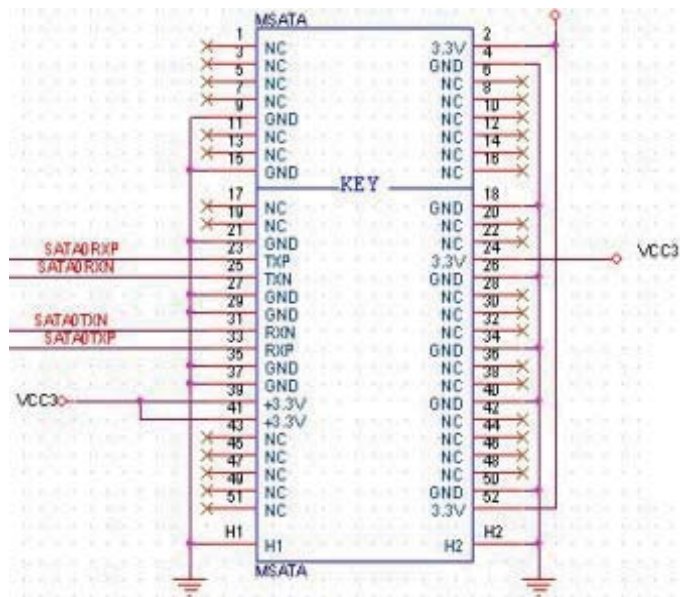
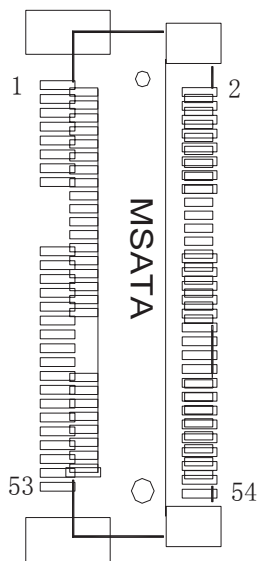
需要采用专用的转接电缆连接 sataPow 硬盘使用

# 用户手册

## MSATA 插座

主板提供 1 个标准的 MSATA 接口，可使用标准 MSATA 存储设备

MSATA 接口定义：

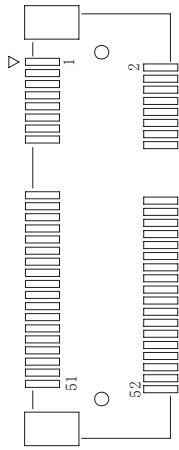


# 用户手册

## MPCIE 插座

主板提供 2 个标准的 MPCIE 接口 用来安装符合规范要求 MPCIE-EX1 总线设备,

下面给出了插槽的引脚定义:



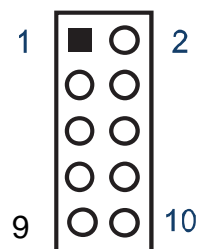
管脚	信号名称	管脚	信号名称
1	Wake#	2	+3.3V
3	NC	4	GND
5	NC	6	+1.5V
7	CLKREQ	8	NC
9	GND	10	NC
11	CLK-	12	NC
13	CLK+	14	NC
15	GND	16	NC
17	NC	18	GND
19	NC	20	NC
21	GND	22	PERST#
23	PER_NO	24	3.3VSB
25	PER_PO	26	GND
27	GND	28	+1.5V
29	GND	30	SMB_CLK
31	PET_NO	32	SMB_DATA
33	PET_PO	34	GND
35	GND	36	USB-
37	GND	38	USB+
39	NC	40	GND
41	NC	42	NC
43	GND	44	NC
45	NC	46	NC
47	NC	48	+1.5V
49	NC	50	GND
51	NC	52	+3.3V

## 5. 主板控制按钮、状态指示

系统前面板插针

(FP1 : 2\*5排针)

Pin	Signal	Pin	Signal
1	P LED-	2	RST+
3	P LED+	4	PWR-
5	HLED+	6	PWR+
7	HLED-	8	PWM_UP
9	RST-	10	PWM_DOWN



### 主板支持上电开机功能：上电开机功能分别由BIOS设置与硬件设置两种选择

BIOS上电开机：Chipset->PCH-I/O Configuration->Rsetore AC Power Loss

(这个选项默认是Power Off, 将它设置成Power ON, 保存退出就可以了)

硬件上机开机：在JP9排针上加跳线帽（短路JP9）即可以实现硬件上电开机功能

# 用户手册

8线触摸屏控制排针 (CON8 排针)

Pin	Signal	Pin	Signal
1	Y+	2	X+
3	SENSN	4	Y-
5	X-	6	Y+
7	X+	8	X-
8	GND		



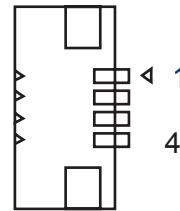
5线触摸屏控制排针 (CON8 排针共用)

Pin	Signal	Pin	Signal
1	Y+	2	X+
3	SENSE	4	Y-
5	X-		



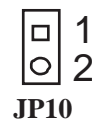
4线触摸屏控制排针 (CON4 排针)

Pin	Signal	Pin	Signal
1	Y-	2	X-
3	Y+	4	X+



## 插针 (JP10)

1-2 短接	4线触摸功能
1-2 开路	5/8线触摸功能 默认设置



开路



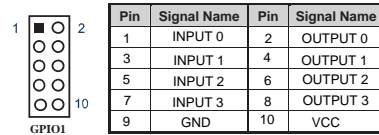
短接

## 6、GPIO 接针

(10针GPIO)

主板提供1个10接针8位GPIO，用户可以通过专用转接电缆引出使用

接针定义如下：



### GPIO 编程

```
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>

/*
GPIO1~8分别对应Super 10的GPIO32, 23, 33, 31, 36, 35, 34, 40; 现以GPIO1, 2, 3, 4配置成输入（输入为只读的）
GPIO5, 6, 7, 8配置成输出（输出为可写可写的）为例
*/

// 1. 初始化数字I/O
#define INEX_PORT 0x2E
#define DATA_PORT 0x2F
#define bit0 0x01
#define bit1 0x02
#define bit2 0x04
#define bit3 0x08
#define bit4 0x10
#define bit5 0x20
#define bit6 0x40
#define bit7 0x80

void main(void)
{
    int temp;
    outpwr(INEX_PORT, 0x87);
    outpwr(INEX_PORT, 0x87);
    outpwr(INEX_PORT, 0x2f);
    outpwr(DATA_PORT, 0x01);
    outpwr(INEX_PORT, 0x07);
    outpwr(DATA_PORT, 0x07);
    outpwr(INEX_PORT, 0x30);
    temp = inpwr(DATA_PORT) | bit2 | bit3 | bit4;
    outpwr(DATA_PORT, temp);
    //Enabled gpio3x
    outpwr(INEX_PORT, 0x1a);
    temp = inpwr(DATA_PORT) & (~bit5 | bit4);
    outpwr(DATA_PORT, temp);
    //Enabled gpio2x
    outpwr(INEX_PORT, 0x1c);
    temp = inpwr(DATA_PORT) & (~bit1);
    outpwr(DATA_PORT, temp);
    //复用引脚配置
    outpwr(INEX_PORT, 0x07);
    outpwr(DATA_PORT, 0x08);
    outpwr(INEX_PORT, 0xe2);
    temp = inpwr(DATA_PORT) & (~bit3);
    outpwr(DATA_PORT, temp);
    outpwr(INEX_PORT, 0xe3);
    temp = inpwr(DATA_PORT) & (~bit1 | bit2 | bit3 | bit4 | bit5 | bit6);
    outpwr(DATA_PORT, temp);
    outpwr(INEX_PORT, 0x07);
    outpwr(DATA_PORT, 0x08);
    outpwr(INEX_PORT, 0xe4);
    temp = inpwr(DATA_PORT) & (~bit0);
    //GPIO1, 2, 3, 4配置为GP1（输入），注意GPIO和各个bit的对应关系
    outpwr(INEX_PORT, 0x07);
    outpwr(DATA_PORT, 0x07);
    //GPIO1, 3, 4
    outpwr(INEX_PORT, 0xec);
    temp = inpwr(DATA_PORT) | bit2 | bit3 | bit1;
    outpwr(DATA_PORT, temp);
    //GPIO2
    outpwr(INEX_PORT, 0xe8);
    temp = inpwr(DATA_PORT);
    temp = temp | bit3;
    outpwr(DATA_PORT, temp);
    //GPIO5, 6, 7, 8配置为GPIO（输出）
    outpwr(INEX_PORT, 0x07);
    outpwr(DATA_PORT, 0x07);
    //GPIO5, 6, 7
    outpwr(INEX_PORT, 0xec);
    temp = inpwr(DATA_PORT) & (~bit6 | bit5 | bit4);
    outpwr(DATA_PORT, temp);
    //GPIO8
    outpwr(INEX_PORT, 0xf0);
    temp = inpwr(DATA_PORT);
    temp = temp & ~bit0;
    outpwr(DATA_PORT, temp);
    printf("Press ESC to exit\n");
    while(inpwr(0x60) != 0x01)
    {
        //GPIO5, 6, 7, 8输出低电平
        outpwr(INEX_PORT, 0x07);
        outpwr(DATA_PORT, 0x07);
        //GPIO5, 6, 7
        outpwr(INEX_PORT, 0xec);
        temp = inpwr(DATA_PORT) & (~bit6 | bit5 | bit4);
        outpwr(DATA_PORT, temp);
        //GPIO8
        outpwr(INEX_PORT, 0xf1);
        temp = inpwr(DATA_PORT);
        temp = temp & ~bit0;
        outpwr(DATA_PORT, temp);
        printf("Press output LOW\n");
        sleep(1);
        //GPIO5, 6, 7, 8输出高电平
        outpwr(INEX_PORT, 0x07);
        outpwr(DATA_PORT, 0x07);
        //GPIO5, 6, 7
        outpwr(INEX_PORT, 0xec);
        temp = inpwr(DATA_PORT) & (~bit6 | bit5 | bit4);
        outpwr(DATA_PORT, temp);
        //GPIO8
        outpwr(INEX_PORT, 0xf1);
        temp = inpwr(DATA_PORT);
        temp = temp | bit0;
        outpwr(DATA_PORT, temp);
        printf("Press output HIGH\n");
        sleep(1);
    }
}
//3. 结束编程
outpwr(INEX_PORT, 0xaa);
}
```



## Watchdog(看门狗编程地址)

```

#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>

#define INDEX_PORT 0x2E
#define DATA_PORT 0x2F
#define bit0 0x01
#define bit1 0x02
#define bit2 0x04
#define bit3 0x08
#define bit4 0x10
#define bit5 0x20
#define bit6 0x40
#define bit7 0x80

/*-----*/
// Name:sio_wdt_enabled()
// Description: Used to active wdt function
// Input: none
// Output: none
/*-----*/
void sio_wdt_enabled()
{
    int temp;
    outporth(INDEX_PORT, 0x07);
    outporth(DATA_PORT, 0x08);
    outporth(INDEX_PORT, 0x30);
    temp = inporth(DATA_PORT) | bit0;
    outporth(DATA_PORT, temp);
}

/*-----*/
// Name:sio_wdt_status_clear()
// Description: Used to clear wdt status
// Input: none
// Output: none
/*-----*/
void sio_wdt_status_clear()
{
    int temp = 0x00;
    outporth(INDEX_PORT, 0x07);
    outporth(DATA_PORT, 0x08);
    outporth(INDEX_PORT, 0x0f1);
    outporth(DATA_PORT, temp);
    outporth(INDEX_PORT, 0x0f2);
    temp = inporth(DATA_PORT) & (~bit4);
    outporth(DATA_PORT, temp);
}

/*-----*/
// Name:sio_wdt_mode_set(int mode)
// Description: Used to clear wdt status
// Input: 0, second mode; 1, minute mode
// Output: none
/*-----*/
void sio_wdt_mode_set(int mode)
{
    int temp = mode;
    int read_temp;
    outporth(INDEX_PORT, 0x07);
    outporth(DATA_PORT, 0x08);
    outporth(INDEX_PORT, 0x0f0);
    read_temp = inporth(DATA_PORT);
    if(temp) //if true, minute mode
    {
        read_temp = read_temp | bit3;
    }else //else, second mode
    {
        read_temp = read_temp & (~bit3);
    }
    outporth(DATA_PORT, read_temp);
}

/*-----*/
// Name:sio_wdt_restart_counter(int)
// Description: Used to clear wdt status
// Input: the time system to restart
// Output: none
/*-----*/
void sio_wdt_restart_counter(int counter)
{
    int temp = counter;
    outporth(INDEX_PORT, 0x07);
    outporth(DATA_PORT, 0x08);
    outporth(INDEX_PORT, 0x0f1);
    outporth(DATA_PORT, temp);
}

void main(void)
{
    int time;
    outporth(INDEX_PORT, 0x87);
    outporth(INDEX_PORT, 0x87);
    sio_wdt_status_clear();
    printf("please input wdt time /S\n");
    scanf("%d", &time);
    sio_wdt_mode_set(0);
    sio_wdt_restart_counter(time);
    sio_wdt_enabled();
}

```

### BIOS 设置

本部分描述如何运用BIOS配置程序设置您的系统。正确设置BIOS各项参数可使系统稳定可靠地工作,同时也能提升系统的整体性能,不恰当的甚至错误的BIOS参数设置则会使系统工作性能大为降低,使系统工作不稳定甚至无法正常工作。

当系统接通电源,正常开机后便可看见进入BIOS设置程序提示的信息,此时(其它时间无效)按下提示信息所指定的按键(通常为<F2>键)即可进入BIOS设置程序。CMOS中BIOS设置内容被破坏时系统也会要求进入BIOS设置程序,通过BIOS修改的所有设置值也都保存在系统的CMOS存储器中,该CMOS存储器由电池供电,即使切断外部电源其内容也不会丢失,除非执行清除CMOS内容的操作。

一旦您进入了AMI BIOS 设定程序,屏幕上会显示出主菜单。主菜单共提供了六种设定功能和两种退出选择。用户可通过方向键选择功能项目,按<Enter>键进入子菜单。

<↑>向前移一项; <↓>向后移一项; <←>向左移一项;

<→>向右移一项; <Enter>确定选择此选项;

<ESC>跳到退出菜单或者从子菜单回到主菜单

<F1 >主题帮助,仅在状态显示菜单和选择设定菜单有效

<F2>放弃设置但是不退出BIOS;

<F3>载入故障安全缺省值

<F4>保存并退出

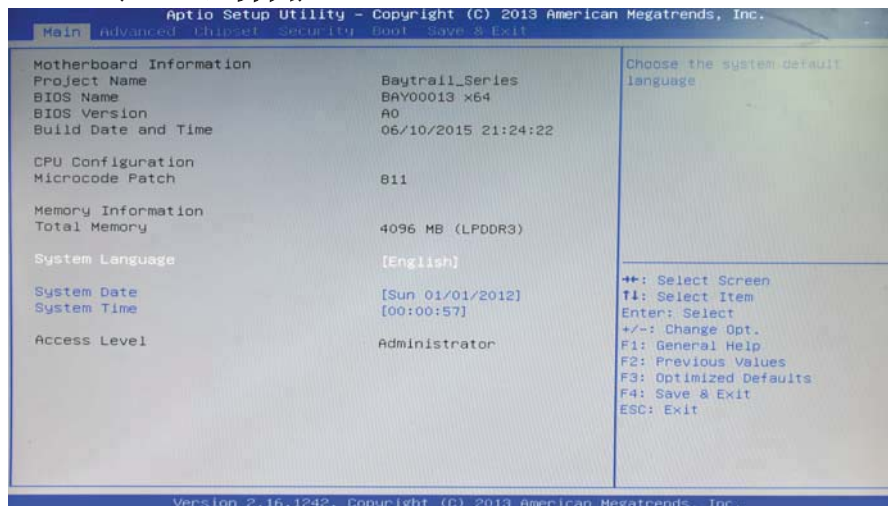
设置方法: 使用方向键移动白色高亮光标至设定处,按回车键进入设定菜单。



注意: 因BIOS程序会不时地更新,以下BIOS设置界面和描述仅供参考。

## 用户手册

### Main (BIOS 主界面)



当您进入 BIOS 设置程序时,主界面将会显现并显示系统概况。主菜单顶部显示的是控制菜单的控制键,主菜单的中部显示的是当前所选,第一个控制菜单的内容灰色信息是只读的内存及 CPU 信息。根据用户系统配置的改变自动调整。菜单右下部是本菜单所用的控制键,如果您需要帮助,按<F1>将显示相关信息帮助您。

#### Motherboard Information

BIOS 信息

#### BIOS Name

BAY00013 X64

#### Total Memory

该项显示 BIOS 检测到的可用内存大小。

#### Project Name

显示用户 CPU 详细信息如制造厂商、型号、参数等。

#### Build Date and Time

选择此选项用<+>/<->来设置目前的日期/时间,以月/日/年/时/分/秒的格式来表示。合理的范围是 Month/月 (Jan. ~Dec.), Date/日 (01~31), Year/年 (最大至 2099), Week/星期 (Mon. ~Sun.), Hour/时 (00~23), Minute/分 (00~59), Second/秒 (00~59)。

#### System language

选择此选项用<+>/<-> 来设置系统语言

#### System Date

系统日期

#### System Time

系统时间

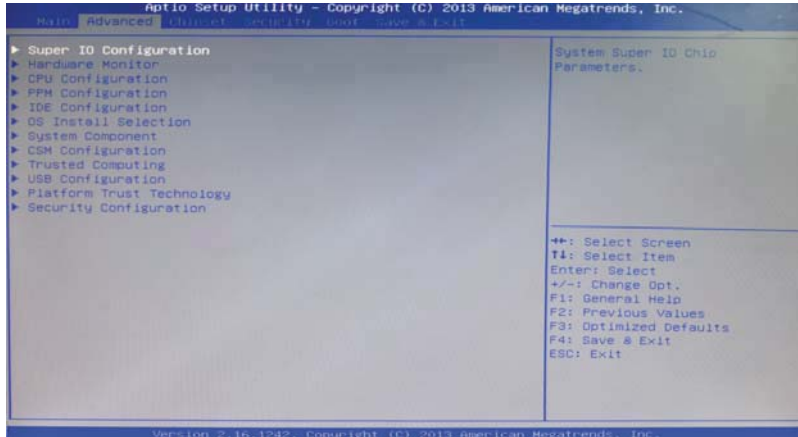
#### Access Level

管理权限

# 用户手册

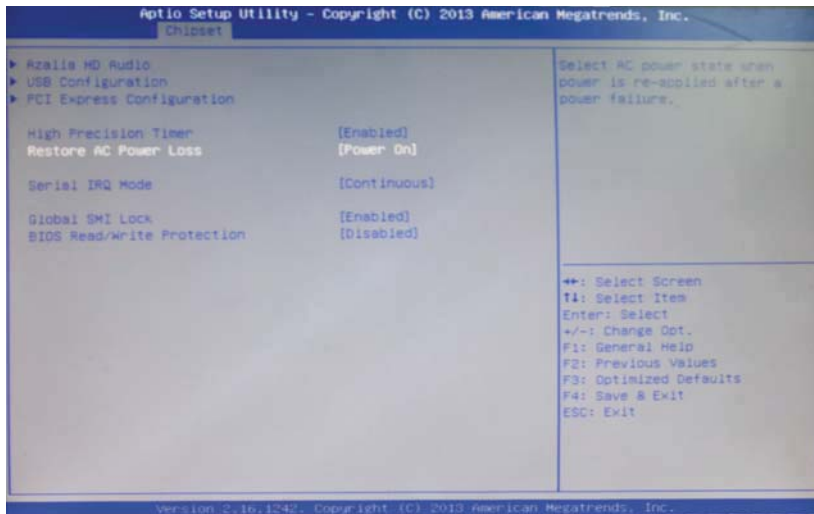
## 3.4 Advanced (高级 BIOS 设置)

此组选项设置系统的基本硬件配置。



## 3.5 Chipset (高级选项)

用户可以根据需求改变 Chipset 里面所需要的功能



### Restore AC Power Loss

当主板断电之后又恢复供电状态选项。

当选择为 Power On 时, 恢复供电时主板自动开机。

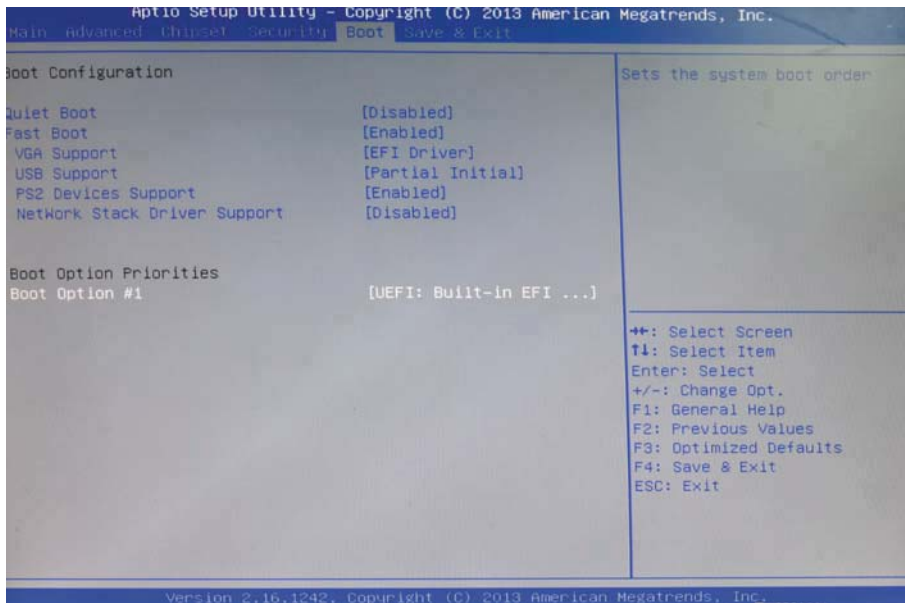
当选择为 Power Off 时, 恢复供电时需按主板开关键才能开机。

当选择为 Last State 时, 主板保持断电时的状态, 即断电时如在关机状态, 恢复供电需按开关键才可开机; 断电时如在开机状态, 恢复供电后则会自动开机。

# 用户手册

## Boot

启动管理设置



## SaveExit

保存与退出

