

# AIOT7-GMLi

## 用户手册



Version 1.0



## 免责声明

本手册内容系本公司知识产权，版权归本公司所有。本产品的所有部分，包括配件与软件等其所有权都归本公司所有。未经本公司书面许可，不得以任何形式对此手册和其中所包含的任何内容进行仿制、拷贝、摘抄或转译为其它语言文字。

我们本着对用户负责的态度精心地编写该手册，但不保证本手册的内容完全准确无误。本手册为纯技术文档，无任何暗示及影射第三方之内容，且不承担排版错误导致的用户理解歧义。若有任何因本手册或其所提到之产品的所有资讯，所引起直接或间接的信息流失或事业终止，本公司及其所属员工恕不为其担负任何责任。

由于我们的产品一直在持续的改良及更新，故本公司保留对本手册内容进行修正而不另行通知之权利。

## 版权声明

本手册中所提及之商标，均属其合法注册公司所有。

本手册所涉及到的产品名称仅做识别之用，其所有权归其制造商或品牌所有人。

# 目录

<b>第1章 综述</b> .....	<b>1</b>
1.1 包装清单 .....	1
1.2 主板规格 .....	2
1.3 主板结构图 .....	3
1.4 主板IO接口结构图 .....	3
1.5 主板布局图 .....	5
1.6 IO面板接口 .....	6
<b>第2章 硬件安装</b> .....	<b>8</b>
2.1 安装内存 .....	8
2.2 连接外部设备 .....	9
2.2.1 Serial ATA 连接器 .....	9
2.2.2 MINI_PCIE、PCIE_X1 插槽 .....	9
2.2.3 M.2-KEYM 插槽 .....	10
<b>第3章 跳线&amp;接头安装与设置</b> .....	<b>11</b>
3.1 各跳线设置说明 .....	11
3.2 跳线设置 .....	11
3.3 COM1/2/3/4/5/6 接口 .....	13
3.4 LPT 接口 .....	13
3.5 JGPIO 接口 .....	14
3.6 DRAWER 接口 .....	14
3.7 CPU_FUN 接口 .....	15
3.8 SMB 接口 .....	15
3.9 SYS_FUN 接口 .....	16
3.10 PS2 接口 .....	16
3.11 DC_OUT 主板取电接口 .....	17
3.12 HDD_PWR1/2 电源接口 .....	17
3.13 VGA_1 接口 .....	18
3.14 F_AUDIO1 接口 .....	18
3.15 SPEAKERS 接口 .....	19
3.16 USB 接口 .....	19
3.17 INVT 接口 .....	20
3.18 F_PANEL 接口 .....	20
3.19 JLPC 接口 .....	21
3.20 EDP 接口 .....	22
3.21 LVDS 接口 .....	23
<b>第4章 BIOS 设置</b> .....	<b>24</b>

4.1 BIOS解释说明	24
4.2 BIOS设定	24
4.2.1 进入BIOS设定程序	24
4.2.2 控制键位	24
4.3 Main	25
4.4 Advanced	26
4.5 Chipset	27
4.6 Security	28
4.7 Boot	29
4.8 Save & Exit	30
<b>第5章 安装驱动</b>	<b>31</b>
<b>第6章 WDT编程指导</b>	<b>32</b>
6.1 编程指导文档说明	32
6.1.1 涉及到的IO函数定义	32
6.2 系统WatchDog编程	33
6.2.1 WatchDog编程注意事项	33
6.2.2 编程示例	33
<b>第7章 GPIO编程指导</b>	<b>35</b>
7.1 编程指导文档说明	35
7.1.1 涉及到的函数定义	35
7.2 GPIO编程示例	37
7.2.1 内存地址说明	37
7.2.2 编程示例	38
<b>附录1 订购信息</b>	<b>39</b>
<b>附录2 产品有毒有害物质或元素标示表</b>	<b>40</b>

# 第1章 综述

## 1.1 包装清单

感谢您选用我们的产品。

请确认您所购买的主板包装是否完整，如果有包装损坏或是有任何配件短缺的情形，请尽快与您的经销商联系。

- ★ 主板 X 1
- ★ 驱动光盘 X 1(工业包装：2PCS/箱)
- ★ SATA硬盘转接线 X 1

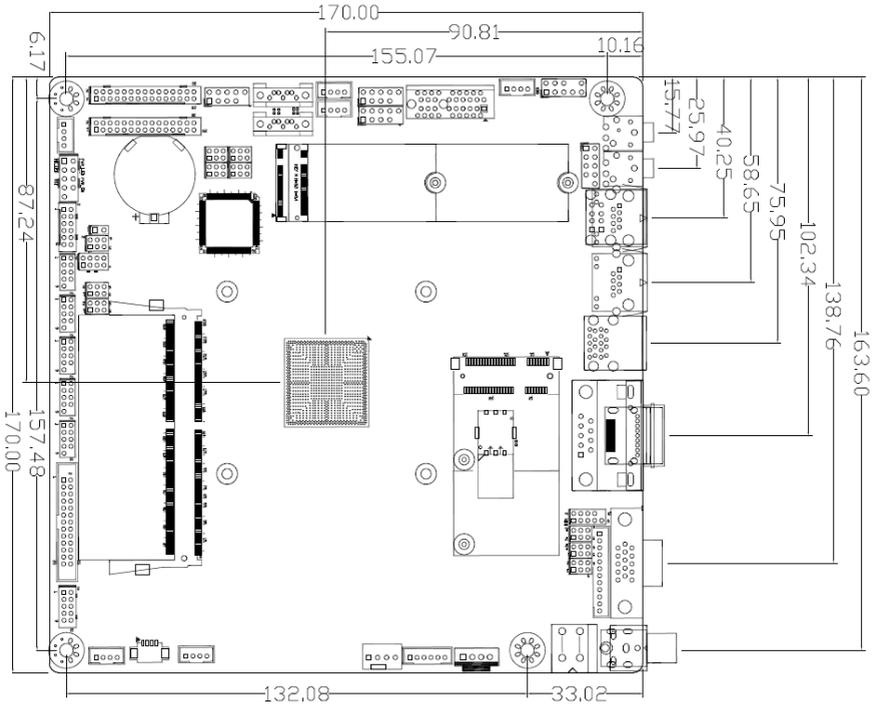


上述附带配件规格仅供参考，实际规格以实物为准，本公司保留修改之权利。

## 1.2 主板规格

处理器	- 采用Intel Gemini Lake系列J4005/J4025, J4105/J4125板贴CPU 标准品板贴J4005双核2.0睿频到2.7GHz或J4105四核1.5睿频到2.5GHz
芯片组	- Intel Gemini Lake SOC
内存	- 1 x DDR4 SODIMM内存插槽, 最高支持DDR4-2400单根最大8GB
显示控制器	- Intel CPU集成显示控制器UHD 600
显示接口	- 三显接口VGA+ HDMI+LVDS或eDP(跳线选择) HDMI 2.0 max resolution up to 4096*2160@60Hz LVDS, Dual Channel 24bit, max resolution up to 1920*1200@60 或eDP, max resolution up to 4096*2160@60Hz
存储	- 1个SATA3.0 - 1个M.2支持2242/2280 的SSD (不用M.2时可选SATA)
音频	- 后IO支持Mic-in, Line-out(单层), 支持机箱前面板Audio输出, 支持带 功放数字音频输出(3W, 4Pin wafer)
网络	- 板载2个Intel i211AT千兆网口 (低配板一个千兆口)
USB	- 2个USB3.0(后), 单网口板2个USB2.0(后) 双网口板8个USB2.0插针, 单网口板6个USB2.0插针 其中USB7~USB10是USB Hub后扩展
I <sup>2</sup> C/SM bus	- 一个4Pin SM Bus接口
串口	- 6个串口(可扩10串口) COM1支持RS422/485, COM2&COM3支持Pin9 +5/12V供电, COM3可选TTL电平
键盘鼠标接口	- PS/2键盘鼠标板内插针
数字 I/O	- 8位数字I/O, 提供电源和地, +5V电平
LPC 扩展接口	- 提供1个LPC局部总线接口 (可扩展安全加密模块、串口模块)
Cash Drawer	- 一个4Pin CashDrawer钱箱控制接口
并口/LPT	- 1个LPT并口插针
电源	- DC-Jack单电压+12V 输入(可选带防脱螺丝的DC-Jack), 另配2Pin凤凰端 子供电 DC-Out接口支持主板取+12v和+5v电500mA
扩展总线	- 1个Mini-PCle扩展(带SIM卡座支持4G模块) - 1个PCIe插槽
工作环境	- 工作温/湿度要求: 0~60°C, 10%~90%无凝结; 可扩展支持-20~70°C, 10%~90%无凝结; - 非工作温/湿度要求: -25~75°C, 5%~95%无凝结
Watch Dog	- 255级可编程秒/分, 支持超时中断或系统复位
BIOS	- AMI UEFI BIOS
操作系统	- Win10/Linux, 32位/64位
PCB外观配色	- PCB采用工业绿油漆喷涂

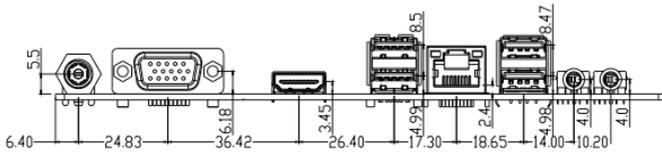
### 1.3 主板结构图



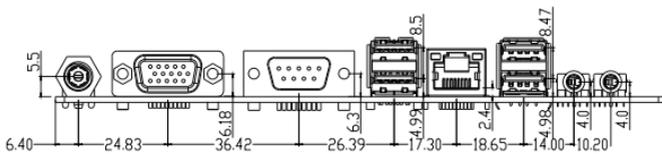
### 1.4 主板IO接口结构图

六角 DCIN 头 (可选):

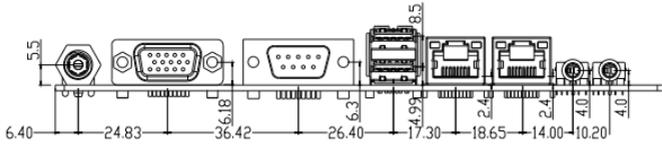
单千兆+HDMI



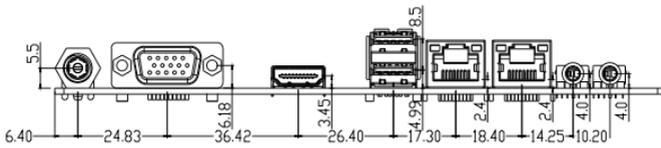
单千兆+COM



双千兆+COM

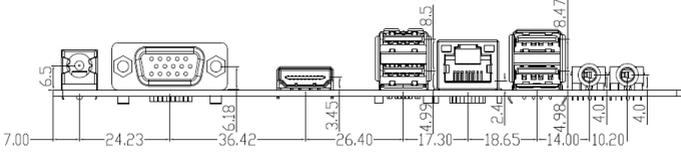


双千兆+HDMI

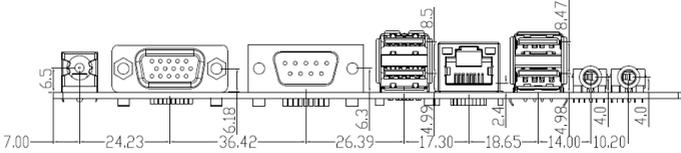


正方形 DCIN 头 (默认):

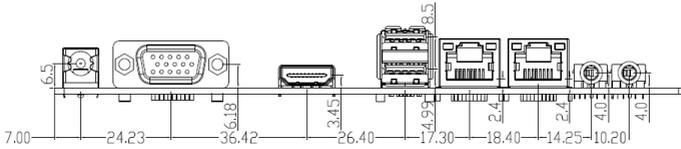
单千兆+HDMI



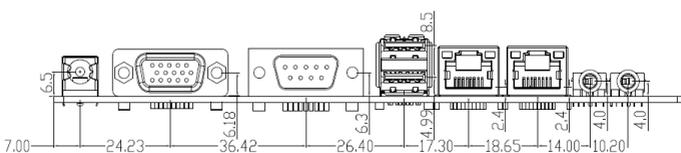
单千兆+COM



双千兆+HDMI



双千兆+COM

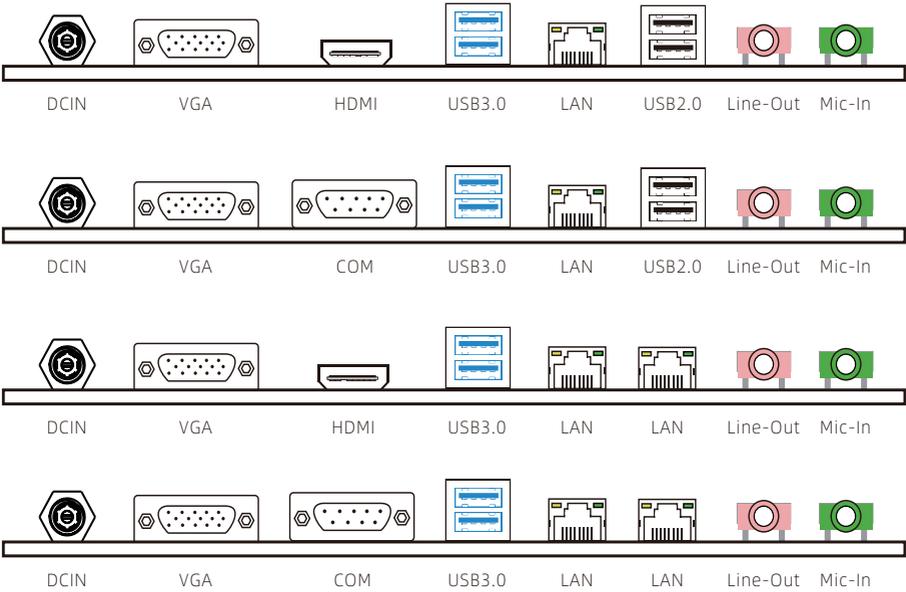


(此图片仅供参考, 请以实物为准)

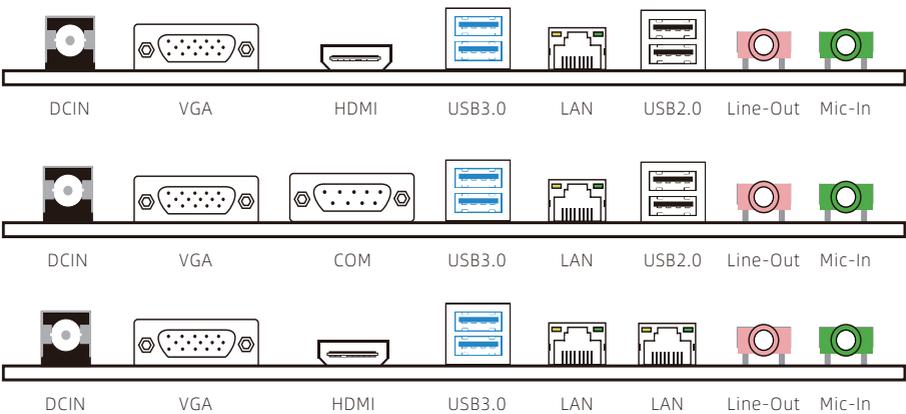


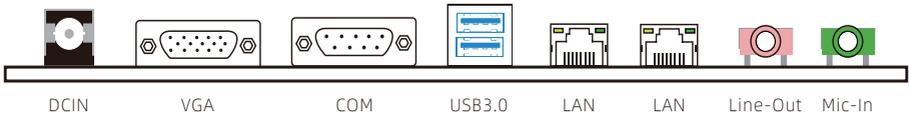
## 1.6 IO面板接口

六角 DCIN 头 (可选):



正方形 DCIN 头 (默认):





(此图片仅供参考，请以实物为准)

- DCIN: DC电源输入接口
- VGA: VGA显示接口
- COM: 串口
- HDMI: HDMI显示接口
- USB3.0: USB3.0接口
- LAN: RJ45 以太网接口
- USB2.0: USB2.0接口
- LINE\_OUT: 耳机插孔
- MIC\_IN: 麦克风插孔

# 第2章 硬件安装

## 2.1 安装内存

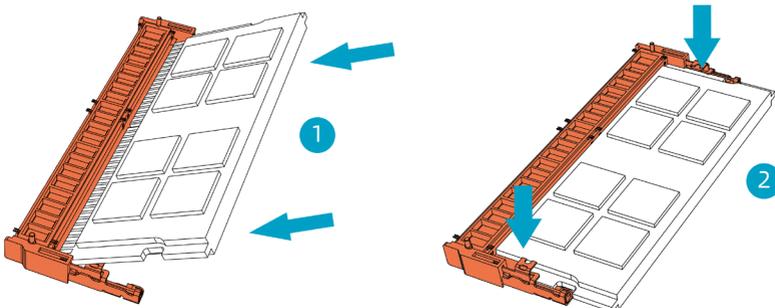
在开始安装内存前，请遵守以下的警告信息：

- 1.请先确认您所购买的内存适用本主板所支持的规格。
- 2.在安装或移除内存之前，请先确定电脑的电源已经关闭以免造成损毁。
- 3.内存设计有防呆标示，若您插入方向错误，内存就无法插入，此时请立刻更改插入方向。

安装内存：

1. 在安装或移除内存之前请先关掉电源，并且拔下AC电源线。
2. 小心握住内存条的两端，不要触碰到上面的金属接点。
3. 将内存条的金手指对齐内存条插槽，并且在方向上要注意金手指凹孔对上插槽的凸起点；
4. 将内存条斜30度插入内存槽处，然后将内存条往下压，压至可以听到“咔”的声响,说明内存已安装成功，可以使用。（注意：将内存条下压的力度，不可过大，以免损坏内存）
5. 要移除内存条，请将DIMM插槽两端的卡榫同时向外推，然后拿出内存条。

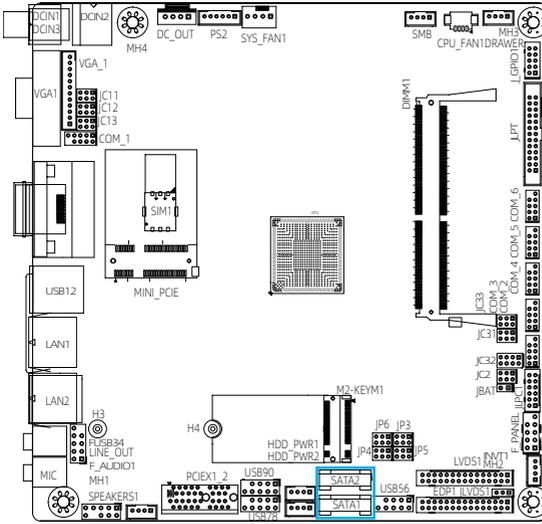
内存安装图示（仅供参考）：



**⚠ 注意：**静电会损害电脑或内存的电子元件，所以在进行以上步骤之前，请务必先短暂接触接地金属物体，以去除身上的静电。

## 2.2 连接外部设备

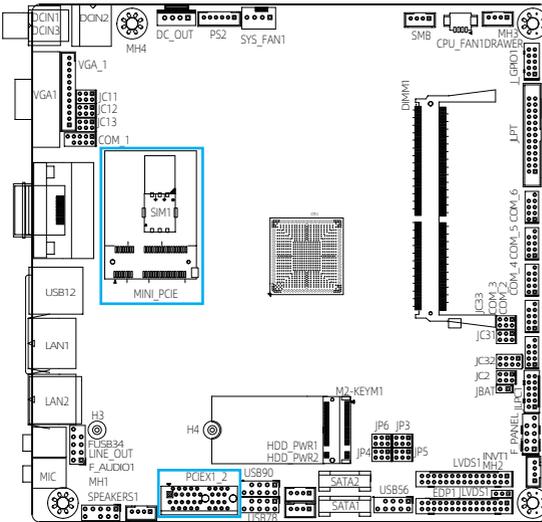
### 2.2.1 Serial ATA 连接器



该接口可支持使用Serial ATA排线来连接Serial ATA硬盘或是其它符合Serial ATA规范的设备。

备注：SATA2与M.2接口二选一，默认M.2。

### 2.2.2 MINI\_PCIE、PCIEX1 插槽

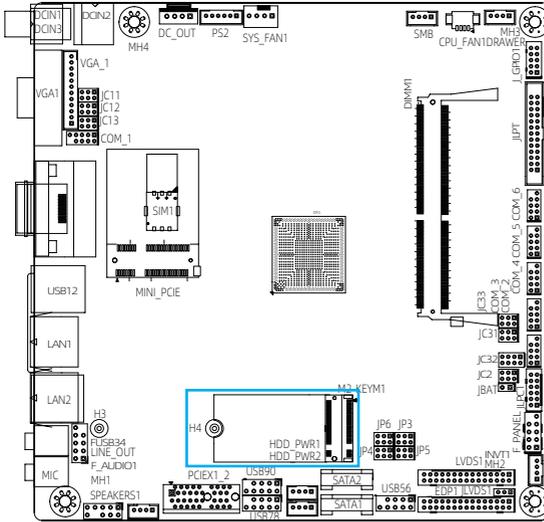


MINI\_PCIE为长卡，支持4G功能。

安装此卡时，请将卡斜30度插入，然后向下压至螺柱处，再用螺丝将其固定。

PCIEX1插槽。

### 2.2.3 M.2-KEYM 插槽



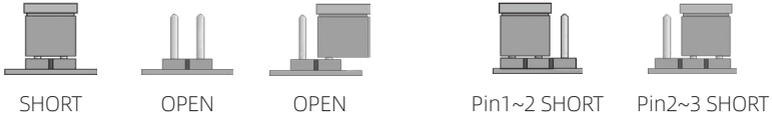
M.2插槽，支持SATA总线的M.2固态硬盘。安装此卡时，请将卡斜30度插入，然后向下压至螺柱处，再用螺丝将其固定。

# 第3章 跳线&接头安装与设置

## 3.1 各跳线设置说明

2针脚的接头：将跳线帽插入两个针脚将使其关闭（短路）。移除跳接帽或是插入其它针脚（为未来扩充预留）将会使其开启。

3针脚的接头：跳线帽可插入针脚1~2或针脚2~3使其关闭（短路）。



怎么辨认跳线的第1脚位置？

- 1.请仔细查看主板，凡有标明"1"或是有白色粗线标记的接脚即为1脚位置。
- 2.观看背板的焊盘，通常方型焊盘为第一脚。

## 3.2 跳线设置

此主板支持EDP或者LVDS接口的各种LCD显示屏，在点屏之前请做如下两步设置：

- 1.按照如下跳线设置说明，正确设置跳线；
- 2.在BIOS设置项中设置正确的显示屏类型和分辨率。

### JBAT 跳线设置

接脚	定义
SHORT	CLEAR CMOS
OPEN	NORMAL

### COM1 设置:

#### JC11/12/13 设置

RS232	RS422	RS485
JC11(1-2)	JC11(5-6)	JC11(3-4)
JC12(1-3)	JC12(3-5)	JC12(3-5)
JC12(2-4)	JC12(4-6)	JC12(4-6)
JC13(1-3)	JC13(3-5)	
JC13(2-4)	JC13(4-6)	

### COM2 设置:

#### JC2 设置

接脚	定义
1-2	5V
3-4	RI#
5-6	12V

COM3 设置:  
JC32/33 设置

TTL	RS232
JC32(3-4)	JC32(2-4)
JC32(7-8)	JC32(1-3)
JC33(1-3)	JC32(6-8)
JC33(2-4)	JC32(5-7)
	JC33(3-5)
	JC33(4-6)

JC31 设置

接脚	定义
1-2	5V
2-3	RI#
5-6	12V

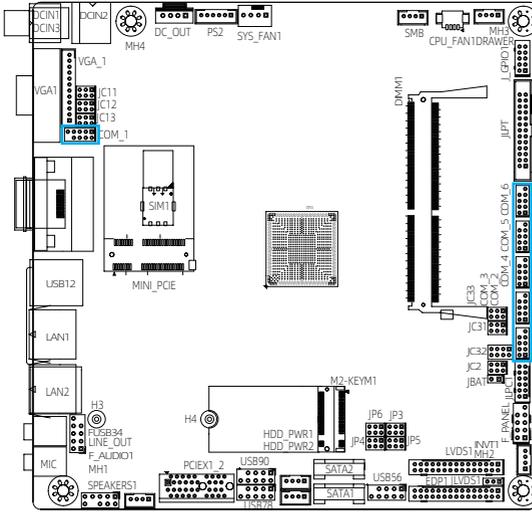
EDP/LVDS 设置  
JP3/4/5/6 跳线设置

LVDS	EDP
JP3(3-5)	JP3(1-3)
JP3(4-6)	JP3(2-4)
JP4(3-5)	JP4(1-3)
JP4(4-6)	JP4(2-4)
JP5(3-5)	JP5(1-3)
JP5(4-6)	JP5(2-4)
JP6(3-5)	JP6(1-3)
JP6(4-6)	JP6(2-4)

JLVDS 跳线(EDP或LVDS、LCDVCC设置)

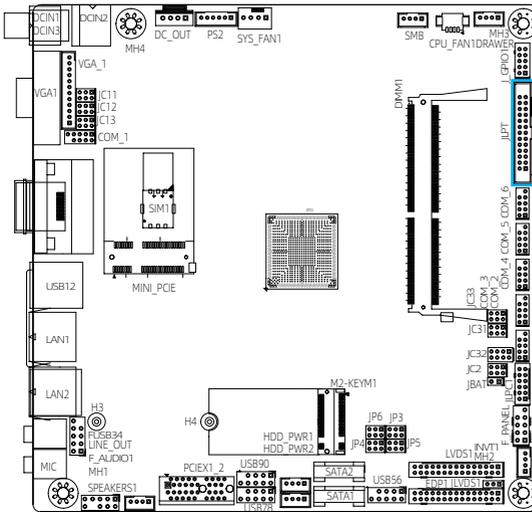
接脚	定义
1-2	3.3V
2-3	5V

### 3.3 COM1/2/3/4/5/6 接口



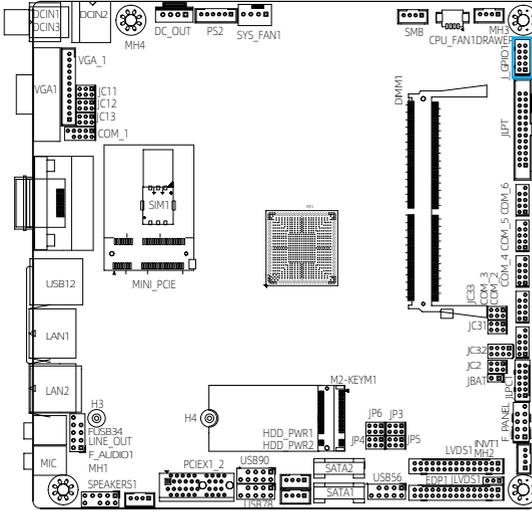
接脚	接脚定义	接脚	接脚定义
1	DCD	2	SIN
3	SOUT	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI		

### 3.4 JPT 接口



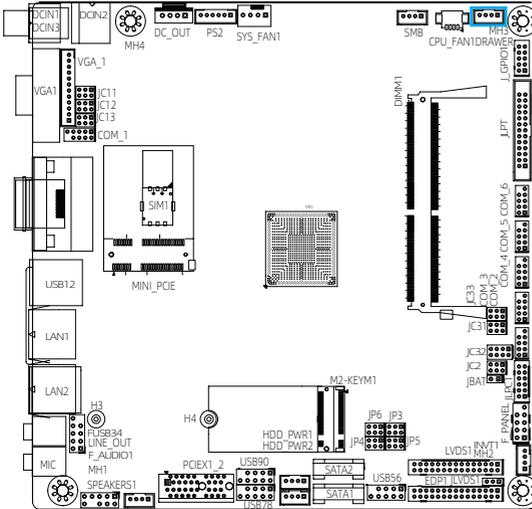
接脚	接脚定义	接脚	接脚定义
1	-STB	2	AFD
3	PRD0	4	ERRORJ
5	PRD1	6	-INIT
7	PRD2	8	-SLIN
9	PRD3	10	GND
11	PRD4	12	GND
13	PRD5	14	GND
15	PR6	16	GND
17	PR7	18	GND
19	ACKJ	20	GND
21	BUSY	22	GND
23	PE	24	GND
25	SLCTJ	26	GND

### 3.5 JGPIO 接口



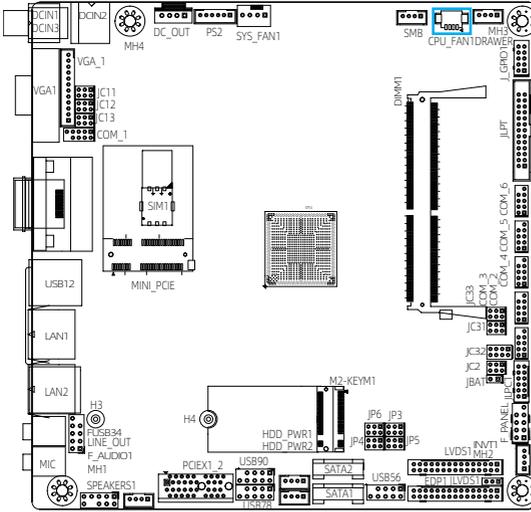
接脚	接脚定义	接脚	接脚定义
1	GPIO1	2	GPIO2
3	GPIO3	4	GPIO4
5	GPIO5	6	GPIO6
7	GPIO7	8	GPIO8
9	5V	10	GND

### 3.6 DRAWER 接口



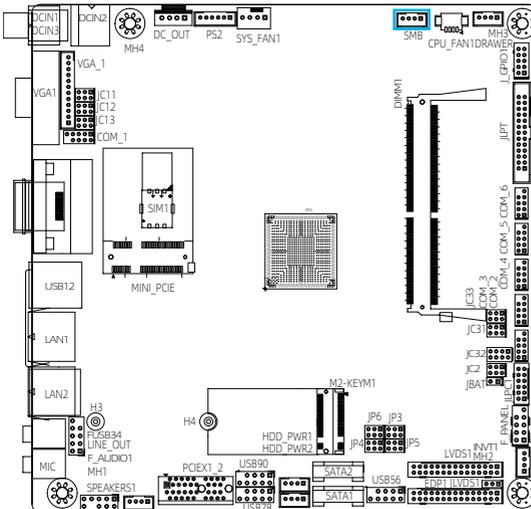
接脚	接脚定义
1	+5V_3V_DRAWER
2	SIO_GP52
3	SIO_GP51
4	GND

### 3.7 CPU\_FAN 接口



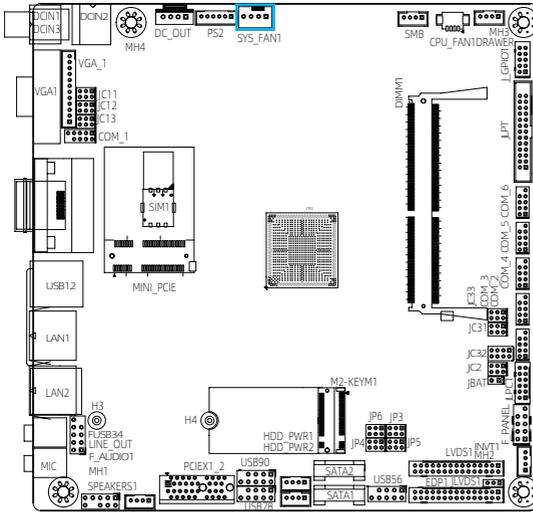
接脚	接脚定义
1	GND
2	+5V
3	FAN_IN
4	FAN_OUT

### 3.8 SMB ( SM Bus ) 接口



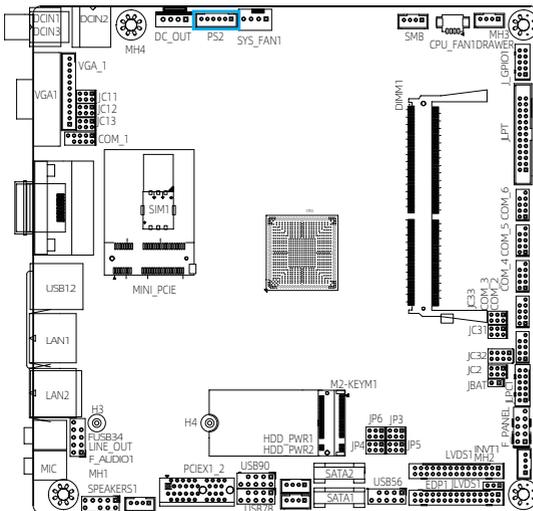
接脚	接脚定义
1	+5V_3V_SMBUS
2	SMB_CLK
3	SMB_DATA
4	GND

### 3.9 SYS\_FAN 接口



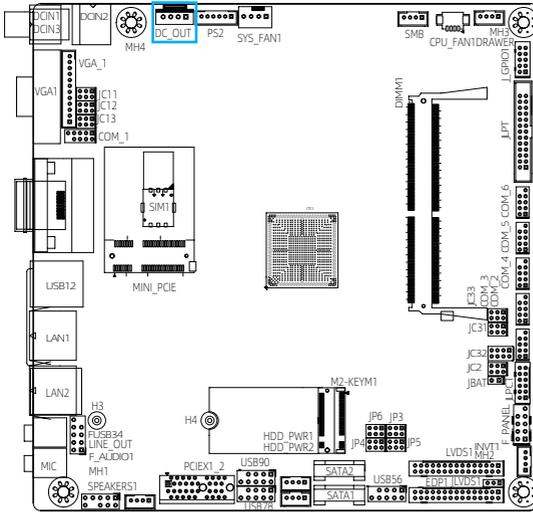
接脚	接脚定义
1	GND
2	12V
3	FAN_IN
4	FAN_OUT

### 3.10 PS2 接口



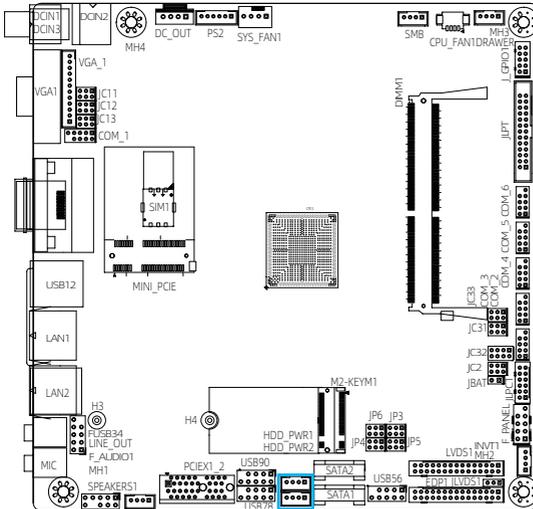
接脚	接脚定义
1	KB_CLK
2	KB_DATA
3	MS_CLK
4	GND
5	5V
6	MS_DATA

### 3.11 DC\_OUT 主板取电接口



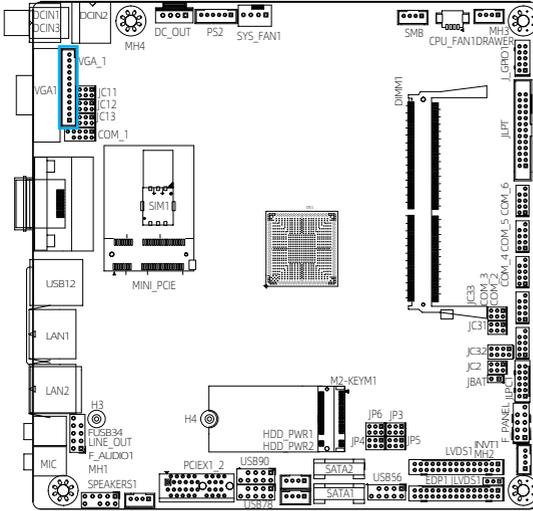
接口	接口定义
1	+12V
2	GND
3	GND
4	+5V

### 3.12 HDD\_PWR1/2 电源接口



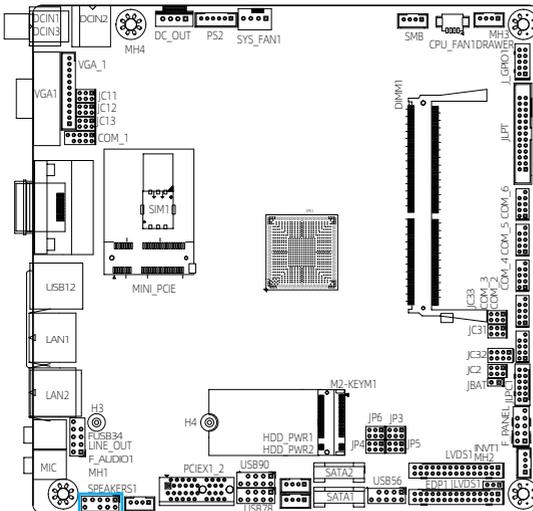
接口	接口定义
1	+12V
2	GND
3	GND
4	+5V

### 3.13 VGA\_1 接口



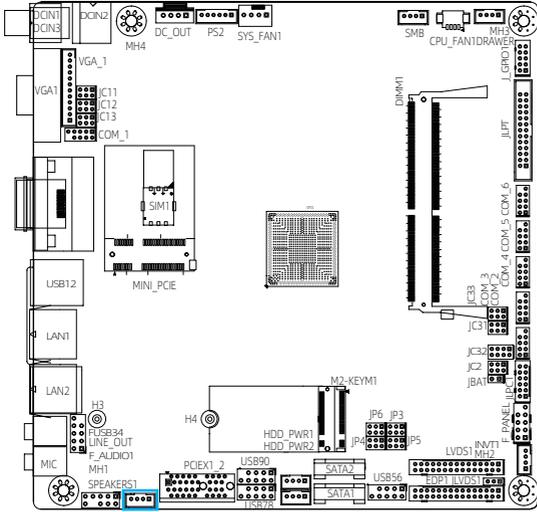
引脚	接脚定义	引脚	接脚定义
1	GND	2	V
3	H	4	GND
5	R	6	GND
7	G	8	GND
9	B	10	GND
11	DAT	12	CK

### 3.14 F\_AUDIO1 (前置面板音频接口)



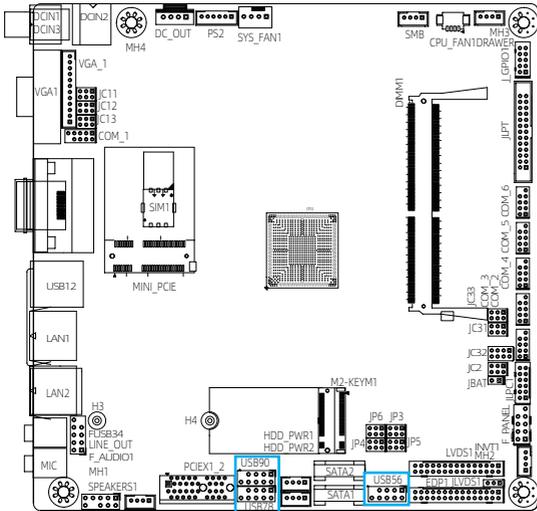
接脚	接脚定义
1	MIC_IN_L
2	GND
3	MIC_IN_R
4	NC
5	F_R
6	F_AUDIO_DET
7	GND
8	空
9	F_L
10	FRONT_IO_SENSE

### 3.15 SPEAKERS 接口（音频功放，可外接3W喇叭）



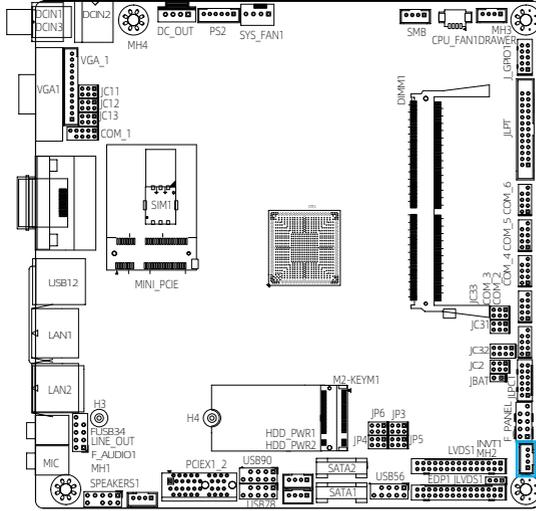
接脚	接脚定义
1	INTSPR-
2	INTSPR+
3	INTSPL-
4	INTSPL+

### 3.16 USB 接口



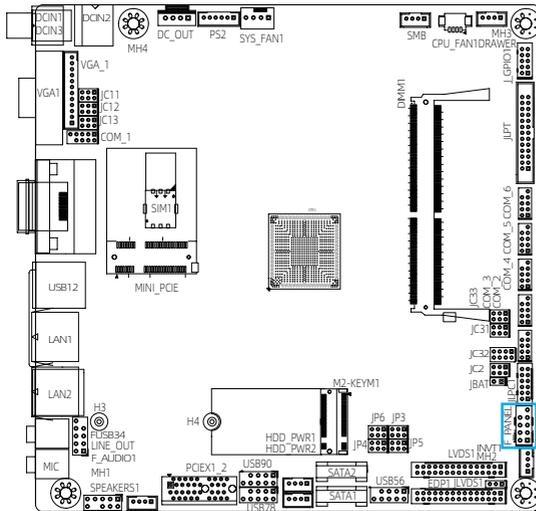
接脚	接脚定义
1	USBPWR3
2	USBPWR3
3	USB_N7_R
4	USB_N0_R
5	USB_P7_R
6	USB_PO_R
7	GND
8	GND
9	空
10	GND

### 3.17 INVT 接口



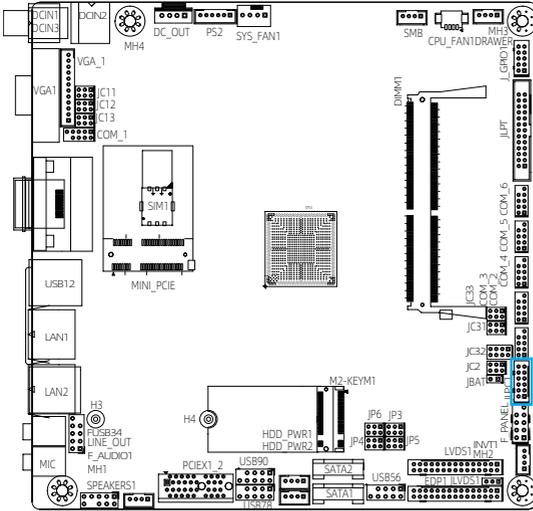
接脚	接脚定义
1	BLK_PWR(12V)
2	BLK_EN
3	PWM_OUT1
4	GND

### 3.18 F\_PANEL 接口



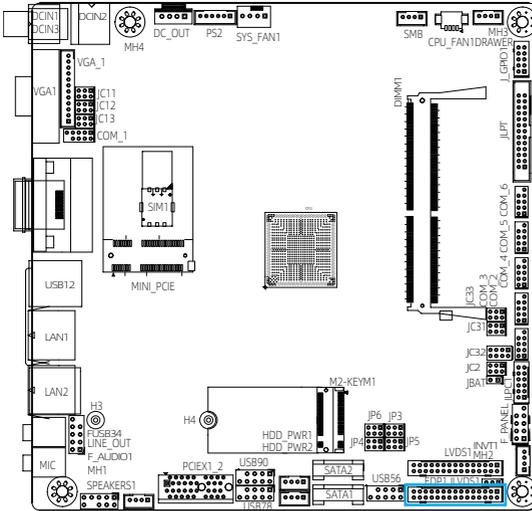
接脚	接脚定义	接脚	接脚定义
1	HDD_LED+	2	PWRLED+
3	HDD_LED-	4	PWRLED-
5	GND	6	PWR-SW
7	SYS-RST	8	GND
9	NC		

### 3.19 J1PC 接口（可外接串口卡或TPM模块）



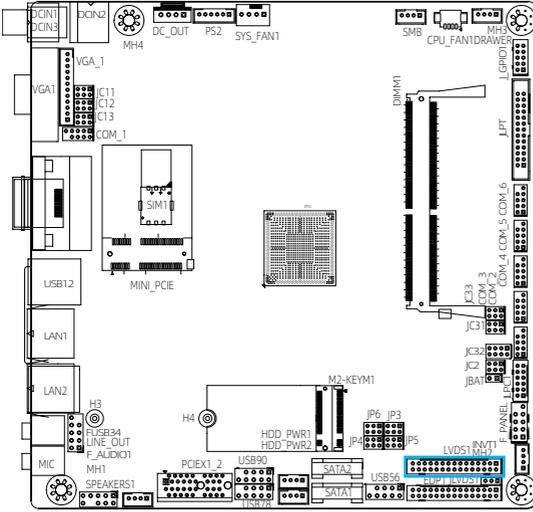
接脚	接脚定义	接脚	接脚定义
1	LAD3	2	3.3V_TPM
3	LAD2	4	SOC_PMU_PLTRST_N
5	LAD1	6	LPC_FRAME
7	LAD0	8	LPC_CLK
9	LPC_ILB_SERIRQ	10	GND
11	+3.3V	12	+5V

### 3.20 EDP 接口



接脚	接脚定义	接脚	接脚定义
1	BLK_PWR	2	BLK_PWR
3	BLK_PWR	4	GND
5	GND	6	NC
7	NC	8	NC
9	LCD_VCC	10	NC
11	LCD_VCC	12	PWM
13	GND	14	GND
15	VDDEN	16	AUXP_EDP
17	EDP_HDP	18	AUXN_EDP
19	GND	20	GND
21	EDP_P3	22	EDP_N3
23	EDP_P2	24	EDP_N2
25	GND	26	GND
27	EDP_P1	28	EDP_N1
29	EDP_P0	30	EDP_NO

### 3.21 LVDS 接口



接脚	接脚定义	接脚	接脚定义
1	VDD_PANEL	2	VDD_PANEL
3	VDD_PANEL	4	GND
5	GND	6	GND
7	LVDS_A_DATA0-	8	LVDS_A_DATA0+
9	LVDS_A_DATA1-	10	LVDS_A_DATA1+
11	LVDS_A_DATA2-	12	LVDS_A_DATA2+
13	GND	14	GND
15	LVDS_A_CLK-	16	LVDS_A_CLK+
17	LVDS_A_DATA3-	18	LVDS_A_DATA3+
19	LVDS_B_DATA0-	20	LVDS_B_DATA0+
21	LVDS_B_DATA1-	22	LVDS_B_DATA1+
23	LVDS_B_DATA2-	24	LVDS_B_DATA2+
25	GND	26	GND
27	LVDS_B_CLK-	28	LVDS_B_CLK+
29	LVDS_B_DATA3-	30	LVDS_B_DATA3+

## 第4章 BIOS 设置

### 4.1 BIOS解释说明

本主板使用AMI BIOS。BIOS全称为Basic Input Output System(基本输入输出系统)。它是存储在电脑主板上的一块ROM (Read-Only Memory)芯片中。当您开启电脑时，BIOS是最先运行的程序，它主要有以下几项功能：

- a.对您的电脑进行初始化和检测硬件，这个过程叫POST(Power On Self Test)。
- b.加载并运行您的操作系统。
- c.为您的电脑硬件提供最底层、最基本的控制。
- d.通过SETUP管理您的电脑。

被修改的BIOS资料会被存在一个以电池维持的CMOS RAM中，在电源切断时所存的资料室不会被丢失。一般情况下，系统运行正常时，无需修改BIOS。如果由于其他原因导致CMOS资料丢失时，须重新设定BIOS值。

### 4.2 BIOS设定

本章提供了BIOS Setup程序的信息，让用户可以自己配置优化系统设置。BIOS中一些未做过多说明的项目，属于非常用项目，在未完全了解其功能之前建议保持默认设置，不要随意更改。

如下情形您需要运行SETUP程序：

- a.系统自检时屏幕上出现错误信息，并要求进入SETUP程序；
- b.您想根据客户特征更改出厂时的默认设置。

注意：由于主板的BIOS版本在不断的升级，所以，本手册中有关BIOS的描述仅供参考。我们不保证本说明书中的相关内容与您所获得的信息的一致性。

#### 4.2.1 进入BIOS设定程序

打开电源或重新启动系统，在自检画面可看到如下信息，按<DEL>键即可进入BIOS设定程序。

Press <Delete> to enter SETUP

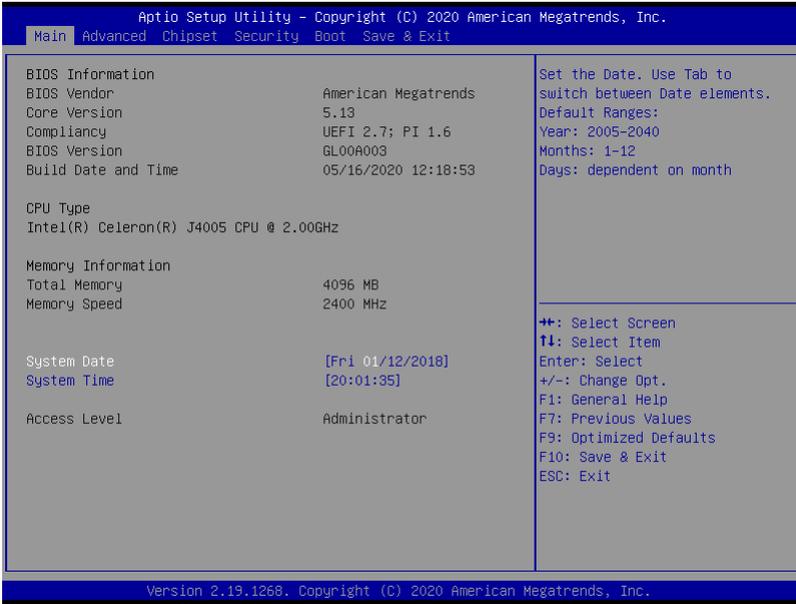
Press <F11> to enter Boot Menu

#### 4.2.2 控制键位

您可以用箭头键移动高亮度选项，按<Enter>键进行选择，按<F1>键寻求帮助，按<Esc>键退出。下列表格将详细列出如何运用键盘来引导系统程序设定。

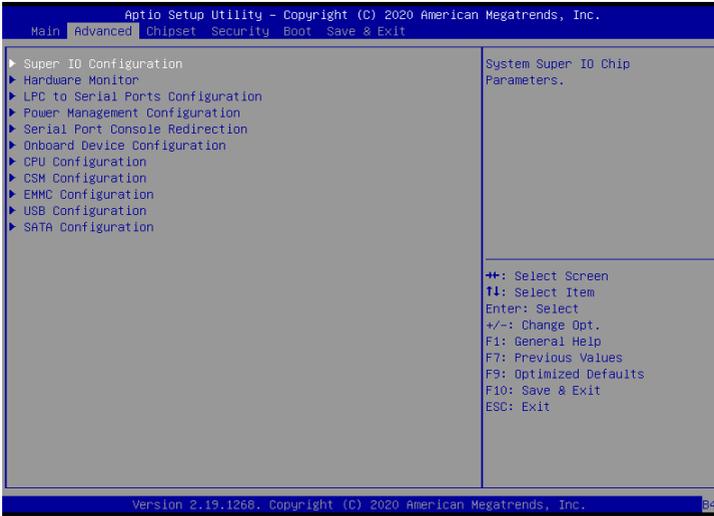
控制键	功能描述
← / →	移动左右箭头选择屏幕
↑ / ↓	移动上下箭头选择上下项目
+ / -	增加/减少数值或改变选择项
<Enter>	选定此选项，进入子菜单
<ESC>	返回主画面，或由主画面中结束CMOS SETUP程序
<F1>	显示相关辅助说明
<F7>	之前设定值
<F9>	载入最优化值的设定
<F10>	保存改变后的CMOS设定值并重启

## 4.3 Main



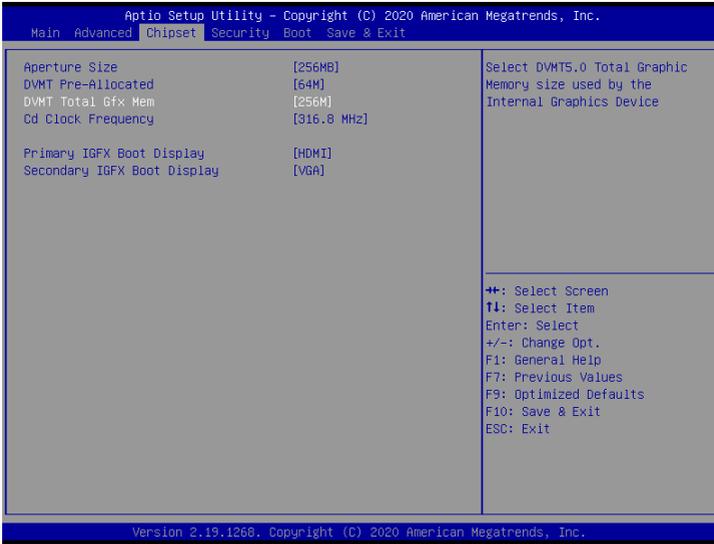
- BIOS Information(BIOS相关信息)
- System Date (系统日期设置)  
设置电脑的日期,格式为“星期,月/日/年”。
- System Time (系统时间设置)  
时间格式为<时><分><秒>。

## 4.4 Advanced



- **Super IO Configuration**  
此项为 Super IO设置信息，包括 COM 口中断号及地址设置。
- **Hardware Monitor**  
此项为监控当前的硬件状态包括CPU温度、电压等系统状态。
- **LPC to Serial Port Configuration**  
此项为系统高级IO芯片参数。
- **Power Management Configuration**  
此项为电源管理配置。
- **Serial Port Console Redirection**  
此项为串口重定向设置。
- **Onboard Device Configuration**  
此项为主板设备控制参数。
- **CPU Configuration**  
此项为处理器设置。
- **CSM Configuration**  
此项为CSM配置。
- **EMMC Configuration**  
此项为EMMC配置。
- **USB Configuration**  
此项为USB信息及控制选项。
- **SATA Configuration**  
此项为设置SATA设备。

## 4.5 Chipset



- **Aperture Size**

此项为孔径大小的选择。

可选项：128M, 256M, 512M.

- **DVMT Pre-Allocated**

根据内部图形设备使用的已分配（固定）图形显存大小选择DVMT5.0。

可选项：64M, 96M, 128M, 160M, 192M, 224M, 256M, 288M, 320M, 352M, 384M, 416M, 448M, 480M, 512M.

- **DVMT Total Gfx Mem**

选择内部图形设备使用的DVMT5.0总图形内存大小。

可选项：128M, 256M, MAX.

- **Cd Clock Frequency**

此项为选择平台支持的最高Cd时钟频率。

可选项：158.4MHz, 316.8MHz, 79.2MHz.

- **Primary IGFX Boot Display**

选择板载显卡主显示设备。

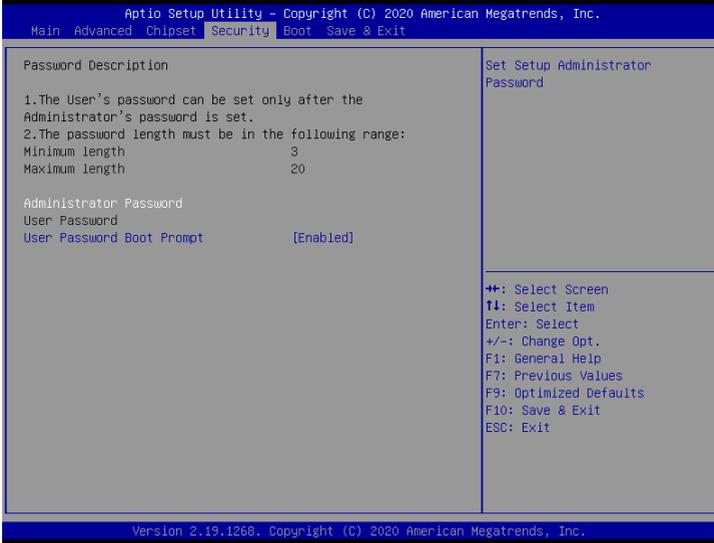
可选项：VGA, LVDS, HDMI.

- **Secondary IGFX Boot Disp**

此项为辅助IGFX引导显示的选择。

可选项：Disabled, VGA, HDMI.

## 4.6 Security



### • Administrator Password

设该选项被用来设置系统管理员密码，有以下这些步骤：

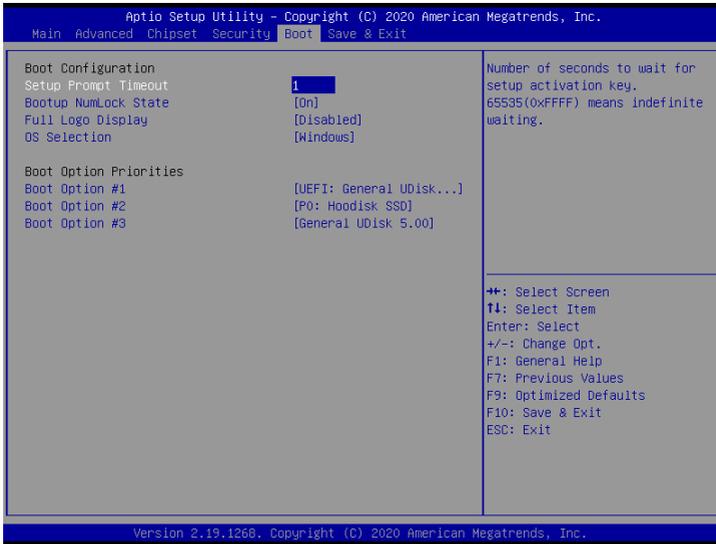
1. 选择Administrator Password设置项，按<Enter>键。
2. 在“Create New Password”对话框中输入3~20位要设定的字符或数字密码，输入完成按<Enter>键后，出现“Confirm Password”对话框，再一次输入密码以确认密码正确。若提示“Invalid Password!”，表示两次输入密码不匹配，请重新再输入一次。若要清除系统管理员密码，请选择“Administrator Password”，出现“Enter Current Password”对话框时，输入旧密码后出现“Create New Password”<Enter>密码即清除。

### • User Password Boot Prompt

在引导到操作系统之前验证管理员或用户密码。

可选项：Enabled,Disabled.

## 4.7 Boot



- **Setup Prompt Timeout**

设置开机界面停留的时间。

- **Bootup Numlock State**

设置系统启动后，Numlock的状态。当设定为On时，系统启动后将打开NumLock，小键盘的数字键有效。当设定为Off的时候，系统启动后Numlock关闭，小键盘方向键有效。

可选项：On,Off.

- **Full Logo Display**

启用或禁用图标。

可选项：Enabled, Disabled.

- **OS Selection**

此项为目标操作系统的选择。

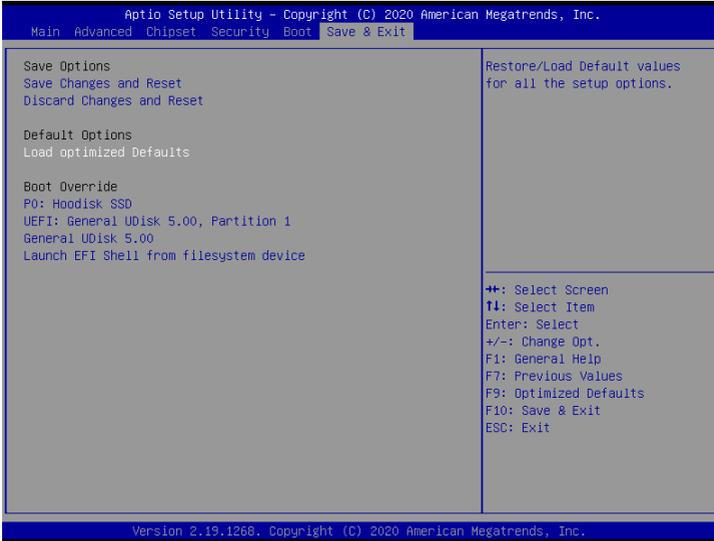
可选项：Windows, Android, MSDOS.

- **Boot Option #1/2/3**

此项为设置系统启动顺序。

可选项：P0: Hoodisk SSD, UEFI General UDisk 5.00,Partition 1, General UDisk 5.00, Disabled.

## 4.8 Save & Exit



- **Save Changes and Reset**  
保存更改并重启系统。
- **Discard Changes and Reset**  
不保存更改并重启系统。
- **Load optimized Default**  
还原/加载所有设置选项的默认值。
- **Launch EFI Shell from filesystem device**  
尝试从文件系统中加载运行Shellx64.efi。

## 第5章 安装驱动

请将主板驱动光盘放入光驱，光盘将自动运行，并弹出如下图所示界面。如果此界面没有出现，请双击运行X:\AUTORUN.EXE（假设光驱盘符为X:）。



(此图片仅供参考，请以实物为准)

请在以上界面依次点击您所需要安装的驱动，根据提示进行安装即可。

# 第6章 WDT编程指导

## 6.1 编程指导文档说明

本文档包含了AioT0-H110主板的WatchDog二次开发说明和软件样例代码。  
说明：本编程指导适用于WDT出自ITE8786 Super IO的相应信号的项目。

### 6.1.1 涉及到的IO函数定义

Linux系统下，需要包含的几个头文件：

```
#include <stdio.h>
#include <errno.h>
#include <sys/io.h>
```

**//从IO地址读1个Byte**

```
/*
  Name:IoRead8
  Input:  GpioAddress      - GPIO base address
  Output: GPIO port read value
  Description:
*/
unsigned char IoRead8(unsigned short GpioAddress) {
  unsigned char ret;
  ret = iopl(3);
  if (ret)
  {
    printf("ret = %d\n", ret);
    printf("errno = %d\n", errno);
    printf("error for iopl\n");
    return 1;
  }
  return inb(GpioAddress);
}
```

**//向IO地址写1个Byte**

```
unsigned char IoWrite8(unsigned short GpioAddress, unsigned char Data) {
  unsigned char ret;
  ret = iopl(3);
  if (ret)
  {
    printf("ret = %d\n", ret);
    printf("errno = %d\n", errno);
    printf("error for iopl\n");
    return 1;
  }
  outb(Data, GpioAddress);
  return 0;
}
```

## 6.2 系统WatchDog编程

### 6.2.1 WatchDog编程注意事项

- 1、WatchDog超时时间建议设置不低于5秒，以免软件系统在CPU负荷高时，不能及时复位Watchdog，引起系统不必要的重启。
- 2、在关闭操作系统和应用软件时，建议先关闭WatchDog或设置较长时间的Watchdog超时时间，以免操作系统未完全关闭时，出现Watchdog复位。

### 6.2.2 编程示例

说明：本编程指导适用于WDT出自IT8786 Super IO的相应信号的项目

#### WDT编程示例

```
#define SuperIO_Index_Port 0x2E
#define SuperIO_Data_Port 0x2F
#define WDT_LDN_DEVICE 0x07
```

#### //a. 进入Super IO的Configuration Mode

```
IoWrite8(SuperIO_Index_Port, 0x87);
IoWrite8(SuperIO_Index_Port, 0x01);
IoWrite8(SuperIO_Index_Port, 0x55);
IoWrite8(SuperIO_Index_Port, 0x55);
```

#### //b. 选择逻辑设备GPIO, LDN 7

```
IoWrite8(SuperIO_Index_Port, 0x07);
IoWrite8(SuperIO_Data_Port, WDT_LDN_DEVICE);
```

#### //c. Clear WDT状态寄存器

```
IoWrite8(SuperIO_Index_Port, 0x71);
Data8=IoWrite8(SuperIO_Data_Port, ((IoRead8(SuperIO_Data_Port)& ~0x10)|
0x01));
```

#### //d. 设置WDT的工作模式，以秒为单位或分为单位进行计时

```
IoWrite8(SuperIO_Index_Port, 0x72);
Data8 = IoRead8(SuperIO_Data_Port);
Data8 = (Data8&0xc0)|0xc0; //WDT计数模式以秒为单位
// Data8 = (Data8&0x40)|0x40; /*WDT计数模式以分为单位*/
IoWrite8(SuperIO_Data_Port, Data8 );
```

#### //e. 启动WDT，WDT范围为0~255，如果为0，则WDT处于停止模式，非0的数值为WDT计数器数值，这个时间之后系统将自动重启。

```
UINT8 Timer;
IoWrite8(SuperIO_Index_Port, 0x73);
```

loWrite8(SuperIO\_Data\_Port,Timer); //低字节Timer为用户自定义的1到255之间的任意值。

loWrite8(SuperIO\_Index\_Port,0x74);

loWrite8(SuperIO\_Data\_Port,0x00); //高字节Timer一般设置为0x00，因为有低字节的Timer配置已经足够用了。

#### **//f. 停用WDT功能**

loWrite8(SuperIO\_Index\_Port,0x73);

loWrite8(SuperIO\_Data\_Por, 0x00);

#### **//g. 重置（喂狗）WDT Timer**

loWrite8(SuperIO\_Index\_Port,0x73);

loWrite8(SuperIO\_Data\_Port,Timer); //Timer为用户自定义的1到255之间的任意值。

#### **//h. 退出Super IO的Configuration Mode**

loWrite8(SuperIO\_Index\_Port,0x02);

loWrite8(SuperIO\_Index\_Port,0x02); //退出Super IO Configuration Mode后，  
//SuperIO的各个逻辑设备将不能再被访问，除非再次进入Configuration Mode。

# 第7章 GPIO编程指导

## 7.1 编程指导文档说明

本文档包含了AioT7-GML主板的GPIO二次开发说明和软件样例代码。

说明：本编程指导适用于GPIO出自Intel PCH的相应信号的项目，不同板卡使用的具体GPIO信号可能不同。另外，请注意访问GPIO的内存地址为物理内存地址。

### 7.1.1 涉及到的IO函数定义（以BIOS中的定义为例）

```
/**
 Reads a 32-bit MMIO register.

 Reads the 32-bit MMIO register specified by Address. The 32-bit read
 value is
 returned. This function must guarantee that all MMIO read and write
 operations are serialized.

 If Address is not aligned on a 32-bit boundary, then ASSERT().

 If 32-bit MMIO register operations are not supported, then ASSERT().

 @param Address The MMIO register to read.

 @return The value read.

 **/
UINT32
EFIAPI
MmioRead32 (
    IN     UINTN      Address
)
{
    //
    // Make sure Address is aligned on a 32-bit boundary.
    //
    ASSERT ((Address & 3) == 0);
    return (UINT32)MmioReadWorker (Address, EfiCpuloWidthUint32);
}

/**
 Writes a 32-bit MMIO register.
```

Writes the 32-bit MMIO register specified by Address with the value specified by Value and returns Value. This function must guarantee that all MMIO read and write operations are serialized.

If Address is not aligned on a 32-bit boundary, then ASSERT().

If 32-bit MMIO register operations are not supported, then ASSERT().

@param Address The MMIO register to write.

@param Value The value to write to the MMIO register.

```
/**  
UINT32  
EFIAPI  
MmioWrite32 (  
IN  UINTN      Address,  
IN  UINT32     Value  
)  
{  
  //  
  // Make sure Address is aligned on a 32-bit boundary.  
  //  
  ASSERT ((Address & 3) == 0);  
  return (UINT32)MmioWriteWorker (Address, EfiCpuIoWidthUint32, Value);  
}  
/**
```

Reads memory-mapped registers in the EFI system memory space.

Reads the MMIO registers specified by Address with registers width specified by Width.

The read value is returned.If such operations are not supported,then ASSERT().

This function must guarantee that all MMIO read and write operations are serialized.

@param Address The MMIO register to read.

The caller is responsible for aligning the Address if required.

@param Width The width of the I/O operation.

@return Data read from registers in the EFI system memory space.

```
/**  
UINT64  
EFIAPI  
MmioReadWorker (  

```

```

    IN  UINTN          Address,
    IN  EFI_CPU_IO_PROTOCOL_WIDTH Width
)
{
    EFI_STATUS Status;
    UINT64 Data;
    Status = mCpulo->Mem.Read (mCpulo, Width, Address, 1, &Data);
    ASSERT_EFI_ERROR (Status);

```

```

    return Data;
}

```

```

/**

```

Writes memory-mapped registers in the EFI system memory space.

Writes the MMIO registers specified by Address with registers width and value specified by Width and Data respectively. Data is returned. If such operations are not supported, then ASSERT().

This function must guarantee that all MMIO read and write operations are serialized.

@param Address The MMIO register to read.  
The caller is responsible for aligning the Address

if required.

@param Width The width of the I/O operation.  
@param Data The value to write to the I/O port.

@return Data read from registers in the EFI system memory space.

```

**/

```

```

UINT64

```

```

EFIAPI

```

```

MmioWriteWorker (

```

```

    IN  UINTN          Address,
    IN  EFI_CPU_IO_PROTOCOL_WIDTH Width,
    IN  UINT64         Data
)
{

```

```

    EFI_STATUS Status;

```

```

    Status = mCpulo->Mem.Write (mCpulo, Width, Address, 1, &Data);
    ASSERT_EFI_ERROR (Status);

```

```

    return Data;
}

```

## 7.2 GIOP编程示例

### 7.2.1 8个GPIO对应的内存地址说明

```
GPIO1: 0xD0C50A30
GPIO2: 0xD0C50A40
GPIO3: 0xD0C50A50
GPIO4: 0xD0C50A60
GPIO5: 0xD0C509A0
GPIO6: 0xD0C509B0
GPIO7: 0xD0C509C0
GPIO8: 0xD0C509D0
```

## 7.2.2 编程示例（以GPIO1为例）

```
UINT Data32;
```

### GPIO输入的配置和读取

#### //a. 设置GPIO1为输入

```
Data32 = MmioRead32(0xFDAE0638);
MmioWrite32 (0xD0C50A30, Data32 &0xFFFFFDF0|0x00000100) ;
```

#### //b. 读取GPIO1作为输入时的电平状态

```
Data32 = (MmioRead32(0xD0C50A30)&0x02)>>1;
```

### GPIO输出的配置和读取

#### //a.设置GPIO1为输出

```
Data32 = MmioRead32(0xD0C50A30);
MmioWrite32 (0xD0C50A30, Data32 &0xFFFFFEFF|0x00000200) ;
```

#### //b.设置GPIO1为输出时的电平状态

```
Data32 = MmioRead32(0xD0C50A30);
MmioWrite32 (0xD0C50A30, Data32 &0xFFFFF0FE) ;//设置为低电平
或
MmioWrite32 (0xD0C50A30, Data32 |0x01) ;//设置为高电平
```

#### //c.读取GPIO1为输出时的电平状态

```
Data32 = (MmioRead32(0xD0C50A30)&0x01);
```

其他的GPIO访问方式类似，只需更换成对应的GPIO地址即可。

## 订购信息

产品型号	芯片组	内存	显示	存储	USB3	USB2	COM	LAN	SSD	PCIe
Alot7-GMLi1	J4025	1	3	2SATA	2	8	6	1 x i211	M.2	2
Alot7-GMLi2	J4025	1	3	2SATA	2	8	6	2 x i211	M.2	2
Alot7-GMLi3	J4105	1	3	2SATA	2	8	6	2 x i211	M.2	2
Alot7-GMLi4	J4125	1	3	2SATA	2	8	6	2 x i211	M.2	2



根据中华人民共和国信息产业部发布的《电子信息产品污染控制管理办法》所展开的 SJ/T11364-2014 标准要求，本产品污染控制标识以及有毒有害物质或元素标识说明如下：

产品有毒有害物质或元素标识：

产品中有毒有害物质或元素的名称及含量

部件名称	有毒有害物质或元素					
	铅(Pb)	汞(Hg)	镉(Cd)	六价铬(Cr(VI))	多溴联苯(PBB)	多溴二苯醚(PBDE)
PCB板	X	O	O	O	O	O
结构件	O	O	O	O	O	O
芯片	O	O	O	O	O	O
连接器	O	O	O	O	O	O
被动电子元器件	X	O	O	O	O	O
焊接金属	X	O	O	O	O	O
线材	O	O	O	O	O	O
其他耗材	O	O	O	O	O	O

O:表示该有毒有害物质在该部件所有均质材料中的含量均在 GB/T 26572 标准规定的限量要求以下。  
X:表示该有毒有害物质至少在部件的某一均质材料中的含量超出 GB/T 26572 标准规定的限量要求。  
备注：X位置的铅的含量超出 GB/T 26572 标准规定的限量要求，但符合欧盟 RoHS 指令的豁免条款。