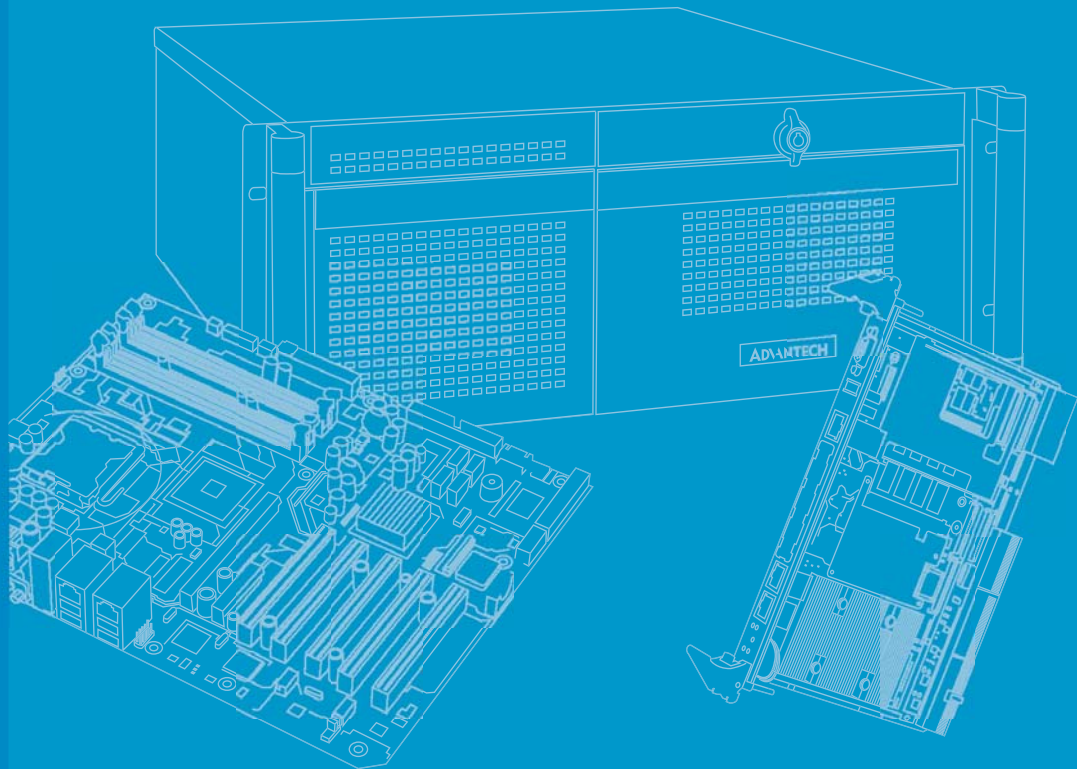


# 用户手册



## ITA-2231 系列

无风扇嵌入式  
Core i 级四核机架式工业电脑

**ADANTECH**

*Enabling an Intelligent Planet*

## 版权声明

随附本产品发行的文件为研华公司 2017 年版权所有，并保留相关权利。针对本手册中相关产品的说明，研华公司保留随时变更的权利，恕不另行通知。未经研华公司书面许可，本手册所有内容不得通过任何途径以任何形式复制、翻印、翻译或者传输。本手册以提供正确、可靠的信息为出发点。但是研华公司对于本手册的使用结果，或者因使用本手册而导致其它第三方的权益受损，概不负责。

## 认可声明

ITA-2231 系列为研华公司的商标。  
所有其他产品名或商标均为各自所属方的财产。

### 在线技术支持

关于技术支持和服务，请访问研华技术支持网站：  
<http://support.advantech.com.cn>

## 产品质量保证（两年）

从购买之日起，研华为原购买商提供两年的产品质量保证。但对那些未经授权的维修人员维修过的产品不予提供质量保证。研华对于不正确的使用、灾难、错误安装产生的问题有免责权利。

如果研华产品出现故障，在质保期内我们提供免费维修或更换服务。对于出保产品，我们将会酌情收取材料费、人工服务费用。请联系相关销售人员了解详细情况。

如果您认为您购买的产品出现了故障，请遵循以下步骤：

1. 收集您所遇到的问题信息（例如，CPU 主频、使用的研华产品及其它软件、硬件等）。请注意屏幕上出现的任何不正常信息显示。
2. 打电话给您的供货商，描述故障问题。请借助手册，产品和任何有帮助的信息。
3. 如果您的产品被诊断发生故障，请从您的供货商那里获得 RMA (Return Material Authorization) 序列号。这可以让我们尽快的进行故障产品的回收。
4. 请仔细的包装故障产品，并在包装中附上完整的售后服务卡片和购买日期证明（如销售发票）。我们对无法提供购买日期证明的产品不提供质量保证服务。
5. 把相关的 RMA 序列号写在外包装上，并将其运送给销售人员。

# 致客户

## 研华为客户提供的服务

研华的每一款产品都是严格按照规格生产的。这样，产品的可靠性在恶劣粗糙的工业环境下也可以得到保证。无论您购买的研华产品属于实验室还是工厂层，请坚信它会是已知的研华产品提供可靠性和易于操作性。客户的满意是我们最关注的。下面是研华客户服务指南。为保证您从我们的服务中获得最大的利益，请谨慎遵循下面的操作指南。

## 技术支持

我们衷心希望您购买的产品能够发挥最大的性能。如果您遇到技术问题，我们随时准备为您提供帮助。对于常见问题，您可以在产品文档中找到满意答案。这些答案通常比我们可以在电话上给您提供的答案更为详细。请先参考本手册。如果仍找不到方案，请搜集和故障有关的所有信息和问题，汇同你手边的资料，给您的经销商打电话。我们的经销商都是接受过专业培训的。通过您提供的产品信息，他们会为您提供所需要的技术支持。事实上，多数问题都是很微小的，通过电话咨询即可解决。此外，在每个工作日，研华工程师都为客户提供免费的技术支持。关于研华任意一款产品安装和操作方面的应用需求或具体信息，我们都时刻准备着为您提供相关的建议。

## 初始检查

打开包装时，用户需确认包装中含有下面所列各项：

- 1 x ITA-2231 系列工业电脑
- 1 x ITA-2231 附件盒
- 1 x 质保卡

如果其中任何一项缺失或者破损，请立即联系您的销售商或销售代表。装货前，我们已全面仔细检查过 ITA-2231 产品。因此您购买的产品应当是完好无损且运转正常的。在您打开 ITA-2231 产品的包装时，请检查是否有破损痕迹（例如，包装箱损坏、划痕、凹痕等）。如果产品有破损或者不符合规格，请立即联系我们的服务部门或您的销售商。同时也要通知搬运人员。请保留包装箱及包装材料以备搬运人员检查。检查之后，我们会给您提供维修或更换服务。

## 警告与注意

### **警告！**



在操作过程中，用户须特别注意该手册中的警告信息，以免造成人身伤害。

### **注意！**



该手册中的注意信息可帮助用户避免损坏硬件或丢失数据，例如：  
如果电池放置不正确，将有爆炸的危险。因此，只可以使用制造商推荐的同一种或者同等型号的电池进行替换。请按照制造商的指示处理旧电池。

### **注！**



此项提供其它额外信息。

## 安全指示

1. 请仔细阅读此安全操作说明。
2. 请妥善保存此用户手册供日后参考。
3. 用湿抹布清洗设备前，请从插座拔下电源线。请不要使用液体或去污喷雾剂清洗设备。
4. 对于使用电源线的设备，设备周围必须有容易接触到的电源插座。
5. 请不要在潮湿环境中使用设备。
6. 请在安装前确保设备放置在可靠的平面上，意外跌落可能会导致设备损坏。
7. 当您连接设备到电源插座上前，请确认电源插座的电压是否符合要求。
8. 请将电源线布置在人们不易绊到的位置，并不要在电源线上覆盖任何杂物。
9. 请注意设备上的所有警告标识。
10. 如果长时间不使用设备，请将其同电源插座断开，避免设备被超标的电压波动损坏。
11. 请不要让任何液体流入通风口，以免引起火灾或者短路。
12. 请不要自行打开设备。为了确保您的安全，请由经过认证的工程师来打开设备。
13. 如遇下列情况，请由专业人员来维修：
  - 电源线或者插头损坏；
  - 设备内部有液体流入；
  - 设备曾暴露在过于潮湿的环境中使用；
  - 设备无法正常工作，或您无法通过用户手册来使其正常工作；
  - 设备跌落或者损坏；
  - 设备有明显的外观破损。
14. 请不要把设备放置在超出我们建议的温度范围的环境，即不要低于  $-25^{\circ}\text{C}$  ( $-13^{\circ}\text{F}$ ) 或高于  $60^{\circ}\text{C}$  ( $140^{\circ}\text{F}$ )，否则可能会损坏设备。
15. 此为 A 级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对干扰采取切实可行的措施。
16. 本产品不带电线组件销售，应购买已通过 CCC 认证的电线组件。

**注意：**计算机配置了由电池供电的实时时钟电路，如果电池放置不正确，将有爆炸的危险。因此，只可以使用制造商推荐的同一种或者同等型号的电池进行替换。请按照制造商的指示处理旧电池。

根据 IEC 704-1:1982 的规定，操作员所在位置的声压级不可高于 70dB(A)。

**免责声明：**该安全指示符合 IEC 704-1 的要求。研华公司对其内容的准确性不承担任何法律责任。

## 安全措施 – 静电防护

为了保护您和您的设备免受伤害或损坏，请遵照以下安全措施：

- 操作设备之前，请务必断开机箱电源，以防触电。不可在电源接通时接触 CPU 卡或其他卡上的任何元件。
- 在更改任何配置之前请断开电源，以免在您连接跳线或安装卡时，瞬间电涌损坏敏感电子元件。
- 无论何时进行操作，请务必完全断开机箱电源。不可电源接通时进行设备连接，以避免瞬间电涌损坏敏感电子元件。只有专业技术人员才可以打开机箱。接触产品的母板、底板或附加卡前，请先确保您接地来移除身上附带的静电。由于现在的电子设备对静电十分敏感，为了安全起见，请使用接地腕环。请将所有电子元件放在无静电的表面或静电屏蔽袋中。



# 目录

## 第 1 章 产品概述 ..... 1

1.1	产品简介.....	2
1.2	产品规格.....	2
1.3	电源信息.....	3
	表 1.1: 电源.....	3
1.4	环境规格.....	3
	表 1.2: 环境规格.....	3
1.5	产品尺寸.....	4
	图 1.1: ITA-2231 产品尺寸.....	4
	图 1.2: ITA-2231 分解图.....	5
	表 1.3: 零件列表.....	5

## 第 2 章 硬件规格 ..... 7

2.1	简介.....	8
2.2	系统状态指示灯.....	8
	2.2.1 系统视图.....	8
	图 2.1: 前视图.....	8
	图 2.2: 后视图.....	8
	2.2.2 系统状态 LED 指示灯.....	8
	2.2.3 网口状态 LED 指示灯.....	9
	2.2.4 串口状态 LED 指示灯.....	9
2.3	跳线和接口.....	9
	2.3.1 跳线描述.....	9
	2.3.2 跳线和接口位置.....	10
	图 2.3: 主板接口及跳线位置图.....	10
	图 2.4: 背板接口及跳线位置图.....	11
	表 2.1: 跳线.....	11
	表 2.2: JCMOS1: Clear COMS 设定.....	11
	表 2.3: PSON1: 开机模式设定.....	12
	表 2.4: VCCGPIO1: GPIO 电压设定.....	12
2.4	I/O 接口.....	12
	图 2.5: ITA-2231 前部 I/O 接口图.....	12
	图 2.6: ITA-2231 后部 I/O 接口图.....	12
	2.4.1 COM 端口.....	13
	表 2.5: COM 端口.....	13
	2.4.2 DVI-D 接口.....	13
	表 2.6: DVI-D 接口.....	13
	2.4.3 DVI-I 接口.....	14
	表 2.7: DVI-I 接口.....	14
	2.4.4 DIO 接口.....	14
	表 2.8: DIO 接口.....	14
	2.4.5 USB 2.0 接口.....	15
	表 2.9: USB2.0 针脚定义.....	15
	2.4.6 USB3.0 接口.....	15
	表 2.10: USB3.0 针脚定义.....	15
	2.4.7 Audio in 接口.....	15
	2.4.8 LAN 接口.....	15
	表 2.11: LAN 接口.....	16
	2.4.9 凤凰端子接口.....	16
	表 2.12: 凤凰端子接口.....	16

## 第 3 章 系统安装 ..... 17

3.1	系统安装简介 .....	18
3.1.1	安装主板 mini-PCIe 卡和 M.2 卡 .....	18
	图 3.1: 安装 M.2 和 mini-PCIe 示意图 .....	18
3.1.2	安装硬盘模组 .....	18
	图 3.2: 装硬盘模组 .....	19
3.1.3	安装上盖 .....	19
	图 3.3: 安装上盖 .....	19
3.1.4	安装耳翼和把手 .....	20
	图 3.4: 安装脚架 .....	20
3.1.5	安装 ITA-EM 卡模组 .....	20
	图 3.5: 安装模组 ITA-EM 卡 .....	21

## 第 4 章 AMI BIOS 设置 ..... 23

4.1	简介 .....	24
	图 4.1: 设置程序初始页面 .....	24
4.2	进入设置界面 .....	24
	图 4.2: 按 <DEL> 键进入设置界面 .....	24
4.2.1	主设置界面 .....	25
	图 4.3: 主设置界面 .....	25
4.2.2	高级 BIOS 特性设置 .....	25
	图 4.4: 高级 BIOS 特性设置界面 .....	26
	图 4.5: Trusted Computing 设置 .....	26
	图 4.6: AMT Configuration 设置 .....	27
	图 4.7: PCH-FW Configuration .....	28
	图 4.8: 设置 F81866 Super IO Configuration .....	28
	图 4.9: 设置串口的设置 .....	29
	图 4.10: Hardware Monitor .....	30
	图 4.11: Serial Port Console Redirection .....	30
	图 4.12: CPU Configuration .....	31
	图 4.13: SATA Configuration .....	33
	图 4.14: Network Stack Configuration .....	34
4.2.3	高级芯片组特性设置 .....	38
	图 4.15: Intel Graphics Configuration .....	39
	图 4.16: PEG Port Configuration .....	41
	图 4.17: 设置 PCI Express Configuration .....	43
4.3	启动设置 .....	44
	图 4.18: 启动设置 .....	44
4.3.1	Boot Configuration .....	44
4.3.2	Hard Drive BBS Priorities .....	45
	图 4.19: Hard Drive BBS Priorities .....	45
4.4	安全设置 .....	45
	图 4.20: “Security” 设置界面 .....	45
4.5	Save & Exit 设置 .....	46
	图 4.21: “Setup Save & Exit Menu” 界面 .....	46

## 第 5 章 驱动安装 ..... 49

5.1	简介 .....	50
5.2	驱动安装 .....	50
5.2.1	芯片组 Windows 驱动安装 .....	50
5.2.2	VGA windows 驱动安装 .....	51
5.2.3	LAN windows 驱动安装 .....	51
5.2.4	AUDIO windows 驱动安装 .....	52
5.2.5	ME windows 驱动安装 .....	52
5.2.6	RST windows 驱动安装 .....	53



	5.2.7 Serial I/O windows 驱动安装.....	53
	5.2.8 USB3.0 windows 驱动安装.....	54
5.3	搭配 ITA-EM 网卡于 Windows8.1 及 Windows10 下网口排序操作说明.....	54
	5.3.1 简介.....	54
	5.3.2 移除 Inbox 驱动的步骤.....	54

## 第 6 章 GPIO 编程指南..... 59

6.1	ITA-2231 数字 I/O 定义（见 2.3.6）.....	60
6.2	配置顺序.....	60
6.3	Function call for reference.....	64

## 附录 A 看门狗定时器编程..... 69

A.1	看门狗定时器编程.....	70
A.1.1	看门狗定时器概述.....	70
A.1.2	编程看门狗定时器.....	70
	表 A.1: 看门狗定时器寄存器.....	70
A.1.3	编程示例.....	71



# 第 1 章

产品概述

## 1.1 产品简介

ITA-2231 是一款带 Core™ i7 级四核处理器，及宽电压输入的 2U 无风扇嵌入式机架式工业电脑，是研华公司推出的一款综合监控和通信管理系统。这款功能强大的计算平台能够 7 天 24 小时连续工作。

## 1.2 产品规格

- **芯片和芯片组：**英特尔 Core™ i7 处理器和英特尔 QM170 芯片组
- **BIOS：**AMI SPI 128 Mb Flash
- **内存：**板载 16GB DDR4-2133，支持 1 个 DDR4 SO-DIMM 插槽，最大可扩展 16GB 内存容量
- **显示：**集成显卡 HD530，共享系统内存最高达 512 MB SDRAM
- **显示模式：**
  - VGA 显示分辨率 1920 x 1200 @ 60 Hz
  - DVI 显示分辨率 1920 x 1200 @ 60 Hz
- **存储：**支持 1 个 M.2 插槽，1 个 3.5” 或 2 个 2.5” HDD 槽位
- **扩展槽：**支持 3 个 ITA-EM 插槽，支持 1 个 PCI104 接口，支援 1 个 Mini PCIe 插槽
- **以太网：**2 个 10/100/1000M 以太网 RJ45 接口
- **USB：**4 x USB 3.0，3 x USB 2.0
- **DVI-I：**1 个 DVI-I
- **DVI-D：**1 个 DVI-D
- **串口：**2 个 DB9 端口，支持 RS-232/422/485 切换
- **声音：**1 个扬声器输出接口，带 2 个 4w 放大器，1 个麦克风输入接口
- **预留区域（后面板，可选）：**1 个 PCI104 I/O 接口面板
- **ITA-EM I/O 模块（可选）：**
  - ITA-EM-SR21-10A1E 支持 8 个串口 RS-232/422/485
  - ITA-EM-SR21-00A1E 支持 8 个串口 RS-232/422/485 并支持 2.5KV 光耦隔离
  - ITA-EM-NC21-C0A1E 支持 8 个 10/100/1000M 以太网 RJ45 接口
  - ITA-EM-NC22-C0A1E 支持 4 个 10/100/1000M 以太网 RJ45 接口
  - ITA-EM-NC22-F0A1E 支持 4 个 10/100/1000M SFP 接口
- **产品尺寸（W x H x D）：**427 x 88 x 325 mm
- **净重：**
  - 单电：7.7kg
  - 双电：8kg

## 1.3 电源信息

ITA-2231 支持易维护电源模块，可选用双路或单路电源输入。

表 1.1：电源	
输入直 / 交流电压	110V DC/ 100-240V AC
输入电流	110 VDC --- 1.2 A 100-240 VAC --- 1.2-0.55 A
电源输入接口	3P 欧规端子
双电源输入方式	电源 1 AC+ 电源 2 DC 或 电源 1 AC+ 电源 2 AC 或 电源 1 DC+ 电源 2 DC

## 1.4 环境规格

表 1.2：环境规格	
工作温度	配 HDD: 0~40° C
	配工业级 SSD 卡: -25~60 ° C (带 0.7m/s 风流)
储存温度	-40~85° C
湿度	95% @ 40° C, 非凝结
振动	带 2.5-inch SSD: 2Grms @ 5~500 Hz, 随机, 1hr/axis
	带 2.5-inch HDD: 1Grms @ 5~500 Hz, 随机, 1hr/axis
	IEC60068-2-6 Sine 2G @ 5~500Hz, 1hr/axis
冲击	10G, IEC-68-2-27, 正弦波, 11ms 间隔
	带 2.5-inch HDD: 10G, IEC-68-2-27, 半正弦波, 11ms 间隔
安全	符合 CCC/CE/CB/UL 标准

## 1.5 产品尺寸

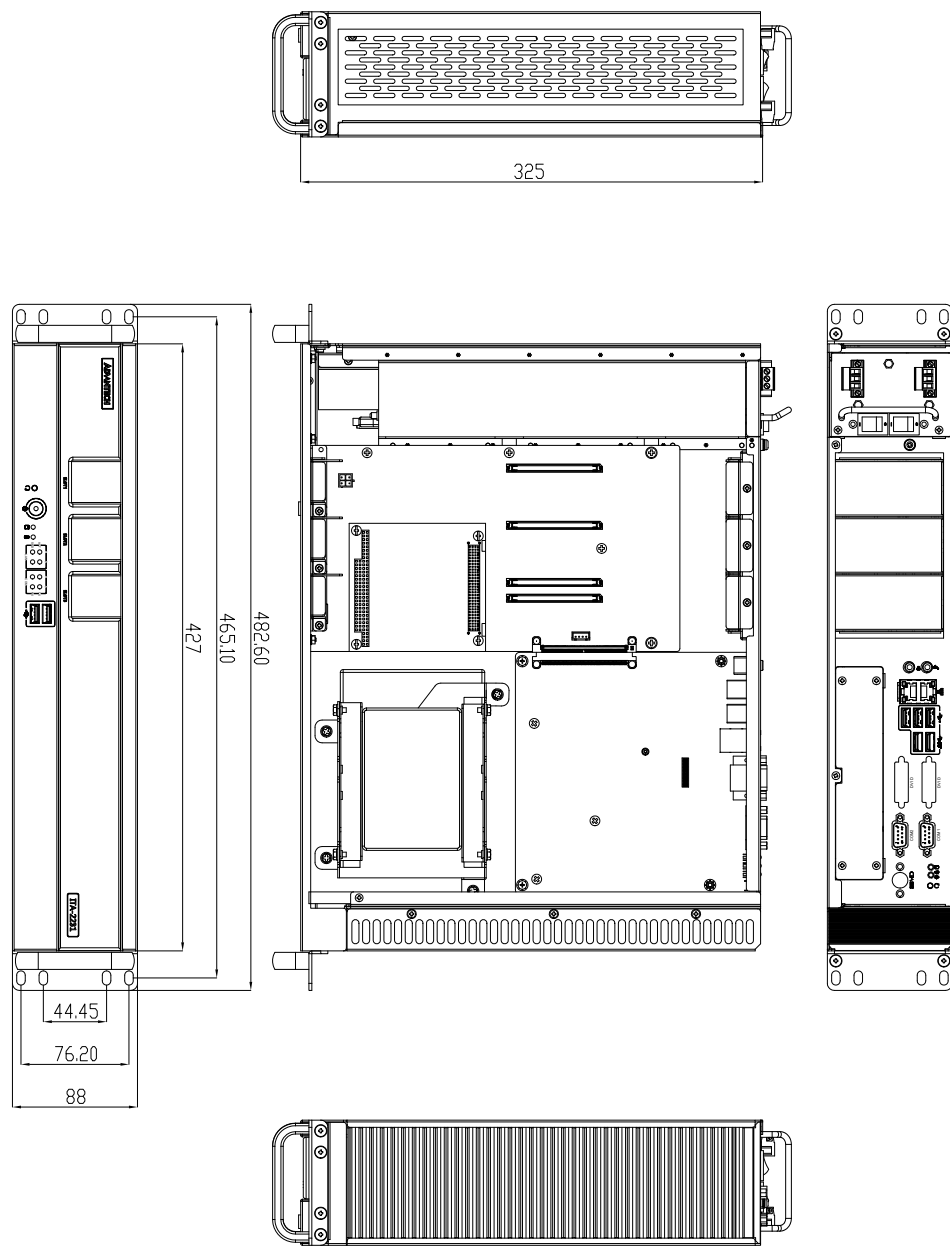


图 1.1: ITA-2231 产品尺寸

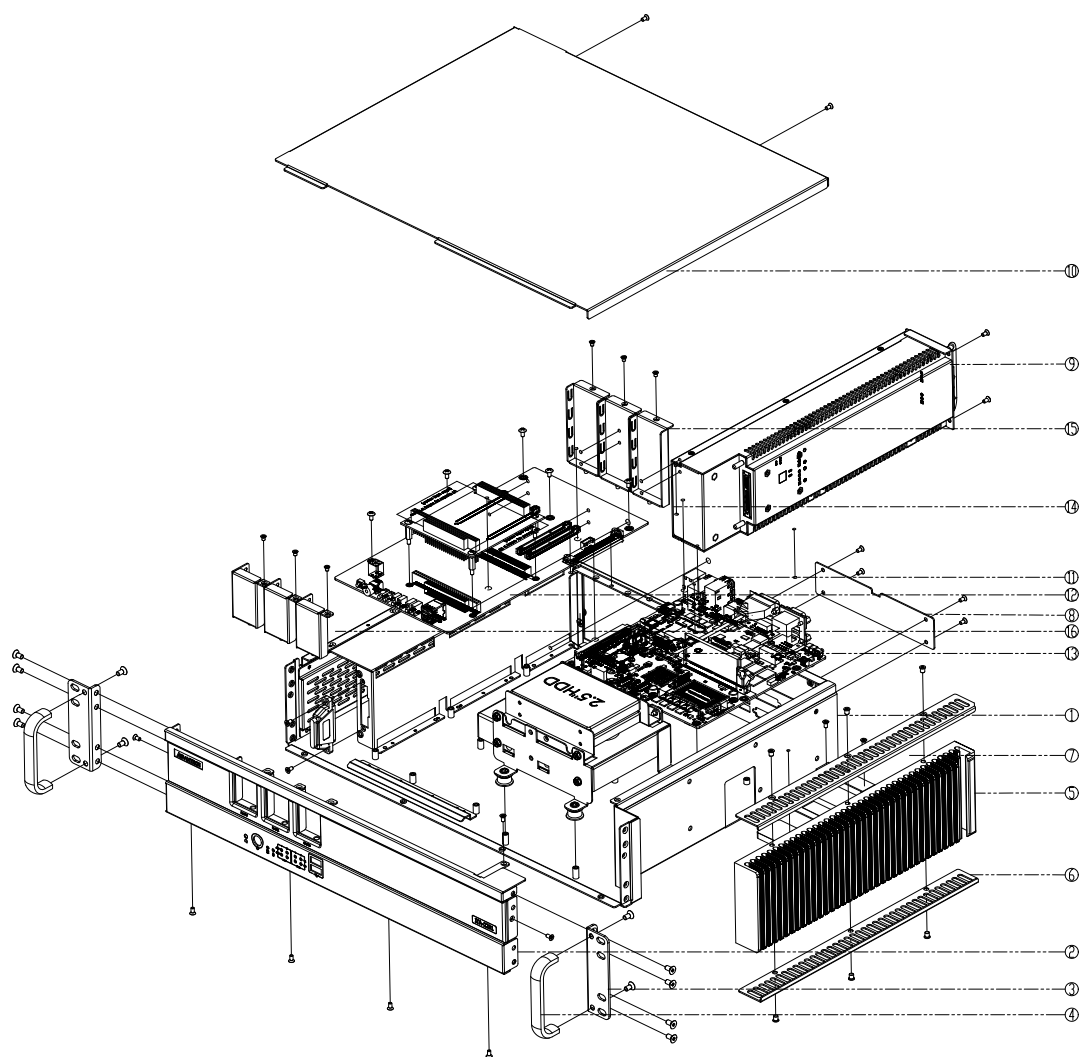


图 1.2: ITA-2231 分解图

表 1.3: 零件列表

1	底座	9	电源模组
2	前面板	10	上盖
3	耳翼	11	主板
4	把手	12	底板
5	散热片	13	3.5 HDD 架
6	护手 (下)	14	2.5 HDD 转接架
7	护手 (上)	15	空白挡片 (后)
8	PCI104 I/O 空白档片	16	空白挡片 (前)





## 第 2 章

硬件规格

## 2.1 简介

以下章节介绍了内部跳线设置和外部接口针脚分配信息，用于实现应用集成。

## 2.2 系统状态指示灯

### 2.2.1 系统视图



图 2.1：前视图



图 2.2：后视图

### 2.2.2 系统状态 LED 指示灯

前面板左下侧 LED 用于指示系统健康和活跃状态。

LED 定义的详细信息请参考下表。

项目	LED	状态	色彩	描述
1	故障	亮	红色	系统出现异常
2	硬盘 LED	闪	橘色	数据正在接收 / 传输
		灭		没有数据正在接收 / 传输

### 2.2.3 网口状态 LED 指示灯

前面板左下侧 LED 用于指示系统网络健康和活跃状态。

LED 定义的详细信息请参考下表。

项目	LED	状态	色彩	描述
1	1000M	亮	绿色	1000M 网络数据正在接收 / 传输
	100M	亮	橘色	100M 网络数据正在接收 / 传输
	10M	不亮		10M 网络数据正在接收 / 传输
	连接	闪烁	绿灯	网络正在连接

### 2.2.4 串口状态 LED 指示灯

前面板左下侧 LED 用于指示系统网络健康和活跃状态。

LED 定义的详细信息请参考下表。

项目	LED	状态	色彩	描述
1	TX Active	闪	绿色	串口数据正在发送
	TX No Data	不亮		串口无发送数据
	RX Active	闪	橘色	串口数据正在接收
	RX No Data	不亮		串口无接收数据

## 2.3 跳线和接口

### 2.3.1 跳线描述

用户可根据需要通过设置跳线配置 ITA-2231。跳线是用来连通电路的金属桥。它包括 2 个金属针脚和一个跳线帽（里面是金属夹片，外部是起保护作用的塑料套）。跳线帽可套住针脚将其连成通路。移走跳线帽则会断开线路。有时，一个跳线具有 3 个针脚，分别为针 1、2、3。这种情况下，用户可以任意选择连接针脚 1、2 或者针脚 2、3。



设备的跳线设置如下图所示：



进行跳线设置时，使用针鼻钳可能会有所帮助。若用户对应用的最佳硬件配置产生任何疑问，请在进行更改前联系当地的分销商或销售代表。通常情况下，用户仅需要一根标准电缆进行大多数连接。

### 2.3.2 跳线和接口位置

板卡带有一些接口和跳线，可供用户根据应用需要进行系统配置。每个接口和跳线的功能如下表所示。跳线和接口在电路板的位置可参照图 2.3，图 2.4 所示。

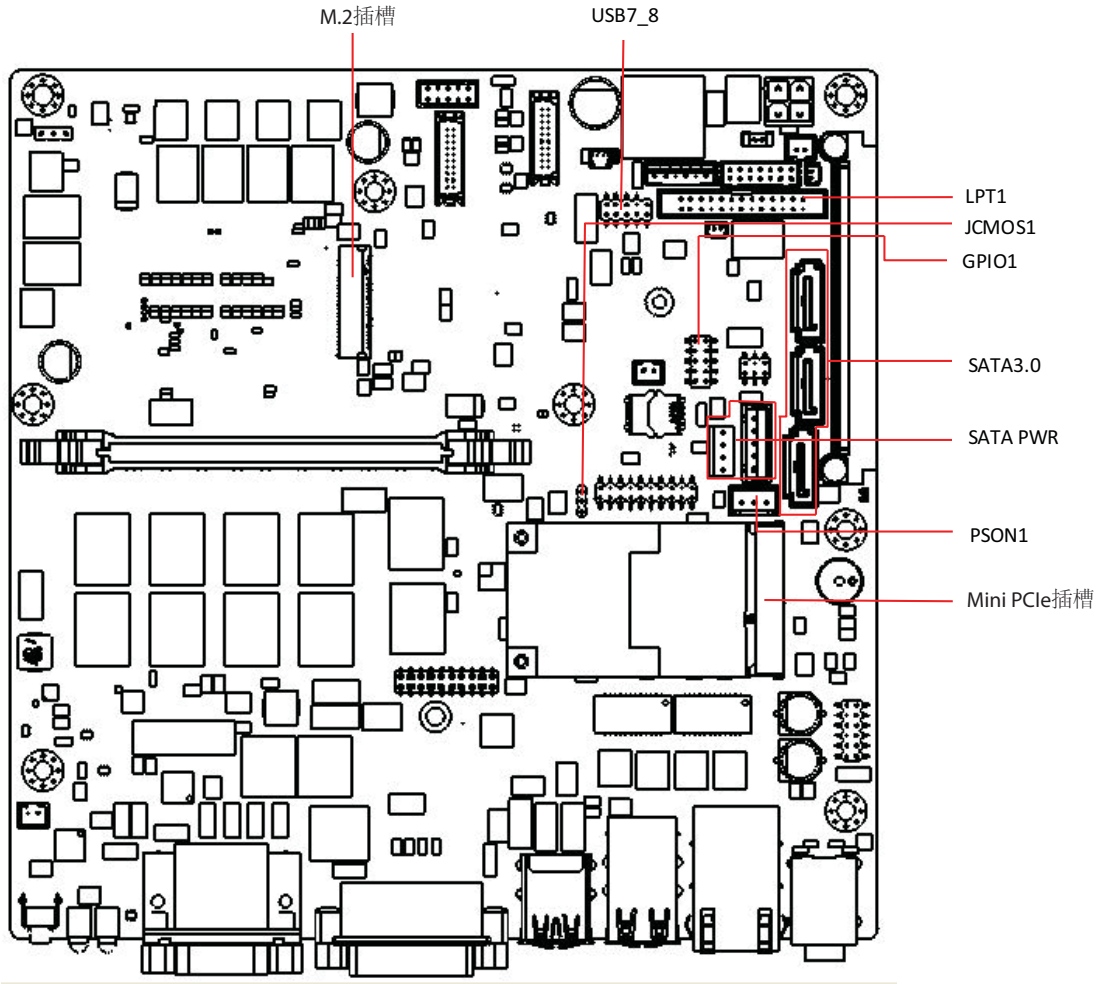


图 2.3： 主板接口及跳线位置图

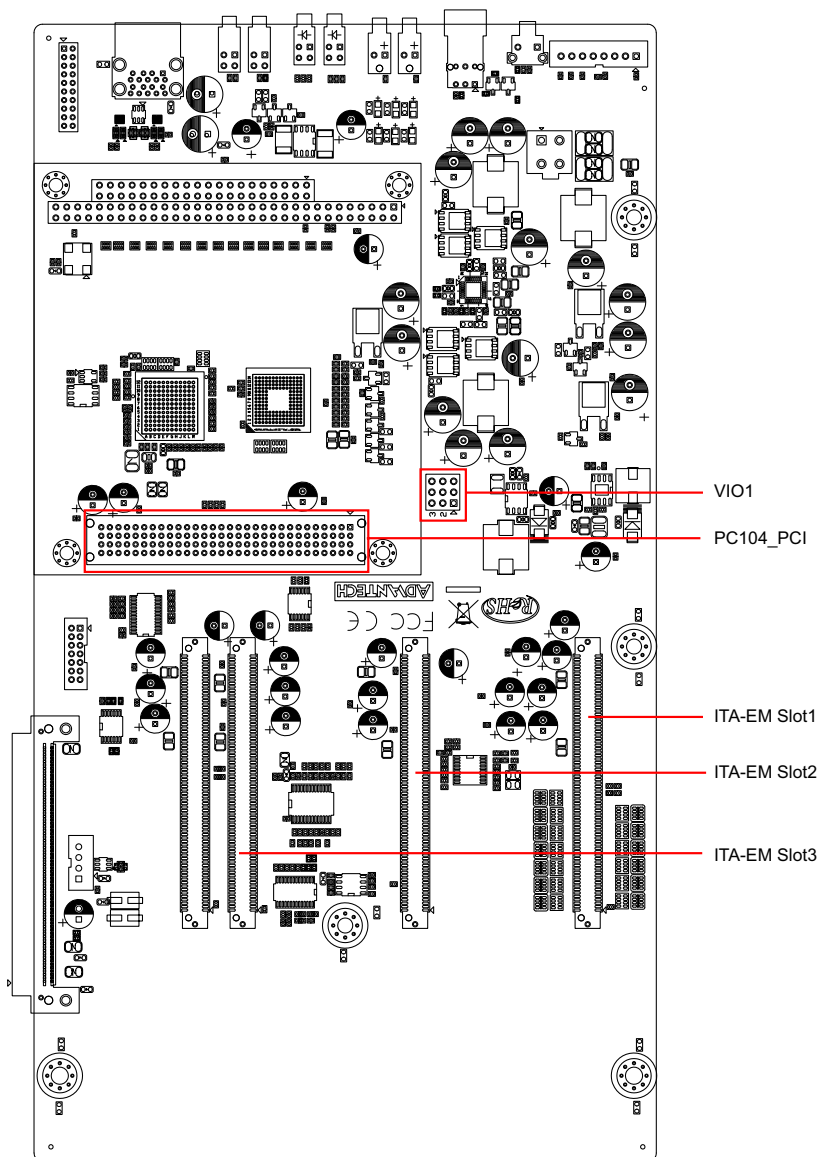


图 2.4：背板接口及跳线位置图

表 2.1：跳线	
名称	功能
JCMOS1	Clear CMOS 设定
PS0N1	开机模式设定
VCCGPIO1	GPIO 电压设定



默认设置



清除 CMOS 设定

表 2.2：JCMOS1: Clear COMS 设定	
闭合针脚	设置
1-2	正常 (+V3.3_SB)*
2-3	清除 CMOS 设定
* 代表默认设置	

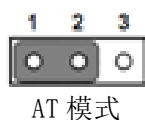


表 2.3: PS0N1: 开机模式设定

闭合针脚	设置
1-2	AT 模式
2-3	ATX 模式 *
* 代表默认设置	

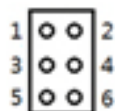


表 2.4: VCCGPIO1: GPIO 电压设定

闭合针脚	设置
1-3	+V5_SB*
2-4	+V3.3_SB
3-5	+V5
4-6	+V3.3
* 代表默认设置	

## 2.4 I/O 接口



图 2.5: ITA-2231 前部 I/O 接口图

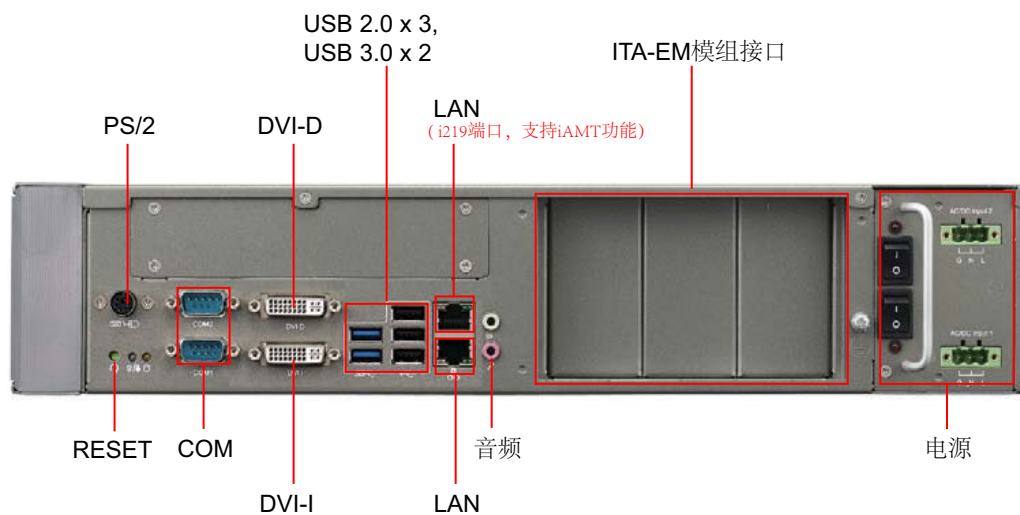


图 2.6: ITA-2231 后部 I/O 接口图

### 2.4.1 COM 端口

ITA-2231 提供了 2 个 DB9 RS-232/RS-422/RS-485 接口。

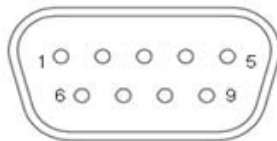


表 2.5: COM 端口

	RS-232	RS-422	RS-485
针脚	信号名	信号名	信号名
1	DCD	Tx-	DATA-
2	RxD	Tx+	DATA+
3	TxD	Rx+	NC
4	DTR	Rx-	NC
5	GND	GND	GND
6	DSR	NC	NC
7	RTS	NC	NC
8	CTS	NC	NC
9	RI	NC	NC

### 2.4.2 DVI-D 接口

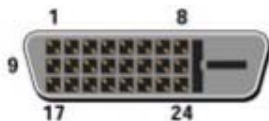


表 2.6: DVI-D 接口

针脚	信号	针脚	信号
1	T. M. D. S. Data2-	13	Ground
2	T. M. D. S. Data2+	14	+5V Power
3	Ground	15	Ground
4	Ground	16	Hot Plug Detect
5	Ground	17	T. M. D. S. Data0-
6	DDC Clock	18	T. M. D. S. Data0+
7	DDC Data	19	Ground
8	无连接	20	Ground
9	T. M. D. S. Data1-	21	Ground
10	T. M. D. S. Data1+	22	Ground
11	Ground	23	T. M. D. S. Clock+
12	Ground	24	T. M. D. S. Clock-

### 2.4.3 DVI-I 接口

ITA-2231 提供 1 个 DVI-I 接口。

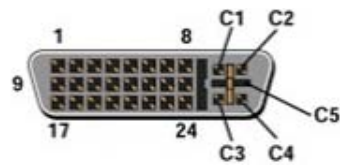


表 2.7: DVI-I 接口

针脚	信号	针脚	信号
1	T.M.D.S. Data2-	16	Hot Plug Detect
2	T.M.D.S. Data2+	17	T.M.D.S. Data0-
3	Ground	18	T.M.D.S. Data0+
4	VGA DCC Clock	19	Ground
5	VGA DCC Data	20	Ground
6	DDC Clock	21	Ground
7	DDC Data	22	DVI Detect
8	Analog Vertical Sync	23	T.M.D.S. Clock+
9	T.M.D.S. Data1-	24	T.M.D.S. Clock-
10	T.M.D.S. Data1+	C1	Analog Red
11	Ground	C2	Analog Green
12	Ground	C3	Analog Blue
13	Ground	C4	Analog Horizontal Sync
14	+5V Power	C5	Ground
15	Ground		

### 2.4.4 DIO 接口

ITA-2231 提供一个 8-bit DIO 接口，需要通过线缆连接。

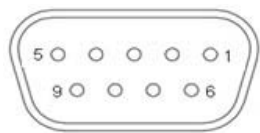


表 2.8: DIO 接口

针脚	信号	针脚	信号
1	GPI00	6	GPI04
2	GPI01	7	GPI05
3	GPI02	8	GPI06
4	GPI03	9	GPI07
5	GND		



### 2.4.5 USB 2.0 接口

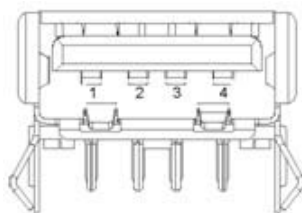


表 2.9: USB2.0 针脚定义

针脚	信号	针脚	信号
1	+V5 (VCC)	3	USB DATA+
2	USB DATA-	4	GND

### 2.4.6 USB3.0 接口

ITA-2231 提供 1 组共 2 个 USB3.0 接口，其符合 USB UHCI Rev. 3.0 标准，可以通过 BIOS 来禁用。

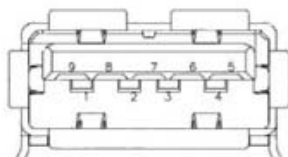
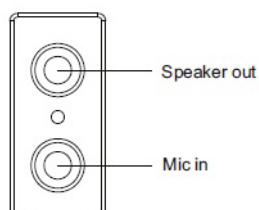


表 2.10: USB3.0 针脚定义

针脚	信号	针脚	信号
1	+V5 (VCC)	6	StdA_SSRX+
2	D-	7	GND_DRAIN
3	D+	8	StdA_SSTX-
4	GND	9	StdA_SSTX+
5	StdA_SSRX-		

### 2.4.7 Audio in 接口

ITA-2231 提供一个集成麦克风输入 / 扬声器输出的音频接口。



### 2.4.8 LAN 接口

板卡上带有 2 个 RJ45 以太网连接器，带有 LED 灯显示网络状态，符合 IEEE 802.3U 10/100/1000 Mbps 标准。

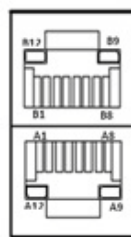


表 2.11: LAN 接口

针脚	信号名
A1/B1	MDI00+
A2/B2	MDI00-
A3/B3	MDI01+
A4/B4	MDI02+
A5/B5	MDI02-
A6/B6	MDI01-
A7/B7	MDI03+
A8/B8	MDI03-
A9/B9	LED GREEN-
A10/B10	LED GREEN+
A11/B11	1000M LED
A12/B12	100m/10M LED

#### 2.4.9 凤凰端子接口

ITA-2231 提供一个 3pin 凤凰端子的电源输入连接器。

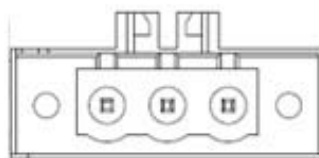


表 2.12: 凤凰端子接口

针脚	信号名
1	GND
2	N
3	L

## 第 3 章

### 系统安装

## 3.1 系统安装简介

以下步骤将指导用户组装 ITA-2231 中的各个模块。

### 3.1.1 安装主板 mini-PCIe 卡和 M.2 卡

1. ITA-2231 主板有提供 mini-PCIe 和 M.2 插槽各一个，使用时请注意区分；
2. 将 mini-PCIe 和 M.2 卡分别插进插槽内用螺丝固定即可。

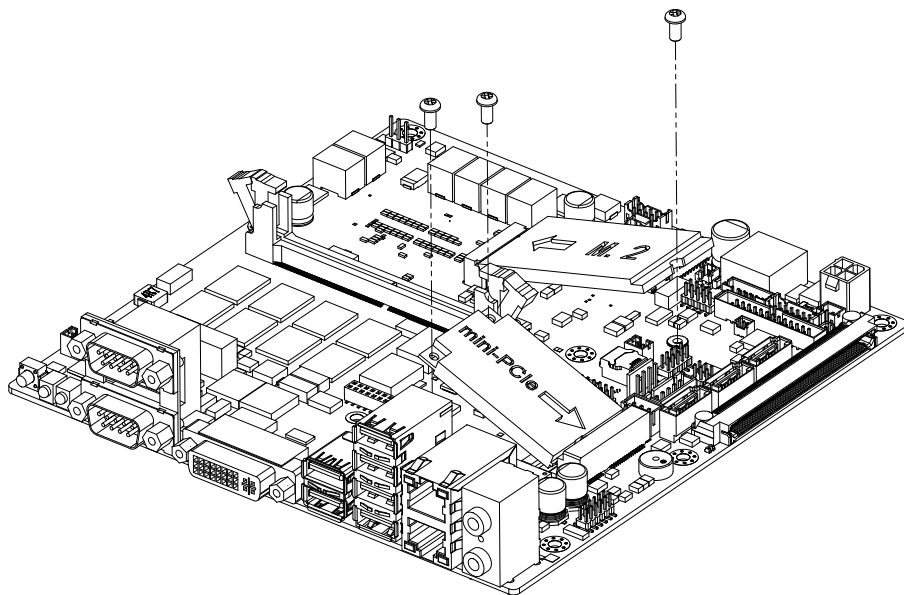


图 3.1：安装 M.2 和 mini-PCIe 示意图

### 3.1.2 安装硬盘模组

ITA-2231 预留有三个可以安装 2.5” 硬盘模组或两个 3.5” 硬盘模组的空间。请参考以下指导安装：

#### 3.1.2.1 安装 3.5” HDD：

1. 打开机箱上盖，将主板前端 HDD 固定架取出。
2. 将 3.5” HDD 铁壳面朝上放入 HDD 固定架内，HDD 两侧螺丝孔对准 rubber 中心孔，用螺丝固定。
3. Cable 接入 HDD 接口，然后将 HDD 模组按照 HDD 接口朝向背板的方向放进机箱，锁住，cable 接入主板即可。

#### 3.1.2.2 安装 2.5” HDD：

1. 将 HDD 按照铁壳面朝上的方向锁在附件盒中的 2.5” HDD 支架上，最大可支持三个 2.5” HDD。
2. 打开机箱上盖，将主板前端 HDD 固定架取出。
3. 将组装好的 2.5” HDD 模组装入 HDD 固定架，用螺丝固定。

4. Cable 接入 HDD 接口，然后将 HDD 模组按照 HDD 接口朝向背板的方向放进机箱，锁住，cable 接入主板即可。

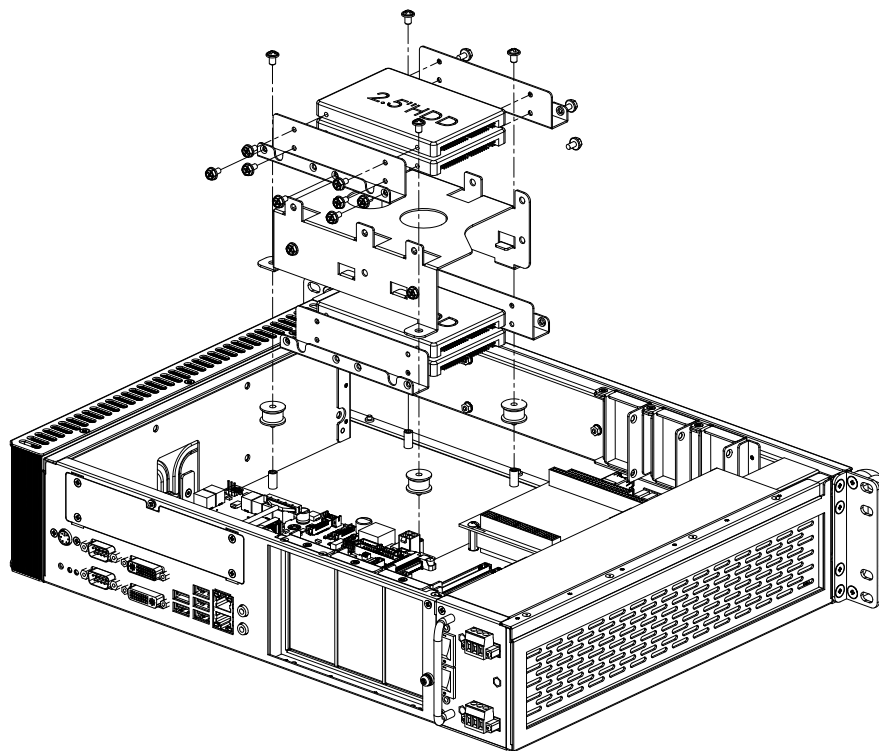


图 3.2：装硬盘模组

### 3.1.3 安装上盖

请按照下面步骤安装机箱顶盖。

1. 按照图 1.4 上盖所示的方向将切口端插入机箱前面板处，然后压好。
2. 用 2 颗螺丝将机箱顶盖固定。

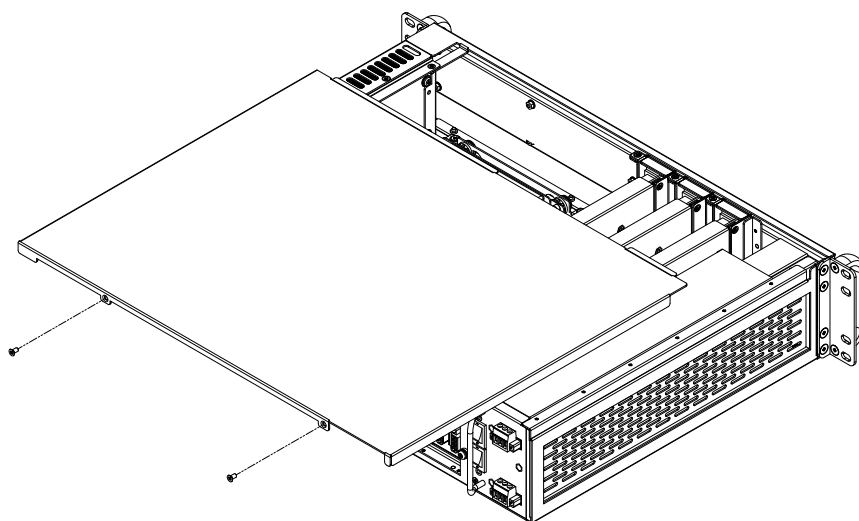


图 3.3：安装上盖

### 3.1.4 安装耳翼和把手

将 2 个耳翼与机箱侧壁处的螺丝孔对好，用螺丝锁好即可。将两个把手与耳翼外处的螺丝孔对好，用螺丝锁好即可。（如图 1.5）

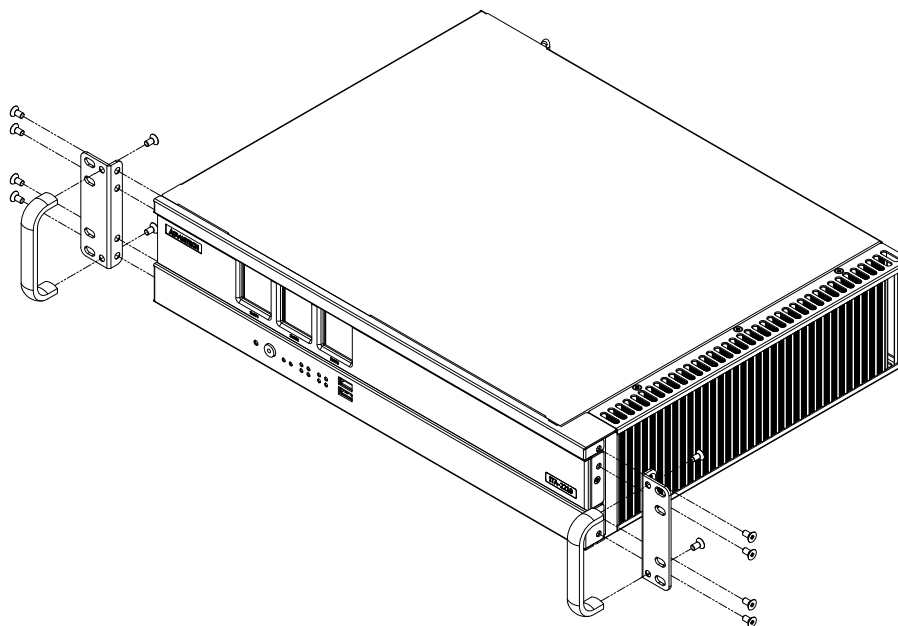


图 3.4：安装脚架

### 3.1.5 安装 ITA-EM 卡模组

请按照下面步骤安装模块 ITA-EM 卡。

1. 打开 ITA-2231 系列系统机箱顶盖。
2. 请按照下图所示插入 ITA-EM 模块。
  - a. 首先将模块 I/O 部分插入机箱 I/O 面板。
  - b. 然后将模块前部固定至机箱前面板。
  - c. 检查 ITA-EM 模块接口是否与底板上的 ITA-EM 插槽平行。
  - d. 检查模块是否完全插入。
3. 用螺丝将 ITA-EM 模块固定至 ITA-2231 系列系统。

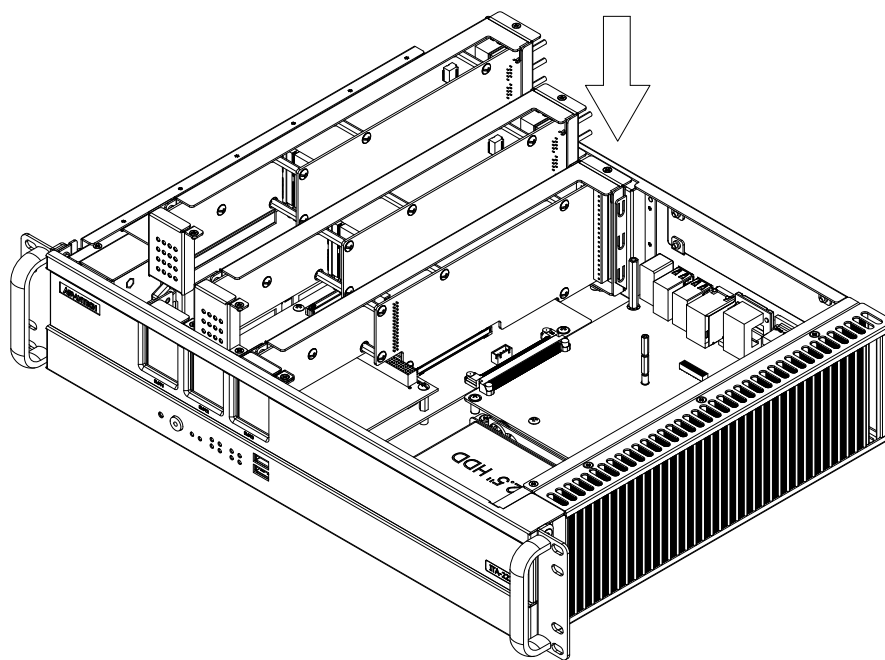


图 3.5: 安装模组 ITA-EM 卡





## 第 4 章

AMI BIOS 设置

## 4.1 简介

AMI BIOS 已被集成到多种工业级和嵌入式母板中，在近年来非常受欢迎。本章介绍如何配置 BIOS，使之适用于 ITA-2231 系列产品。用户可在 AMI BIOS 设置实用程序中更改 BIOS 设置、实现对 ITA-2231 系列特殊性能的控制。设置程序内包含多个菜单，允许用户对计算机性能进行设置。本章介绍 ITA-2231 系列 BIOS 设置的基本知识。

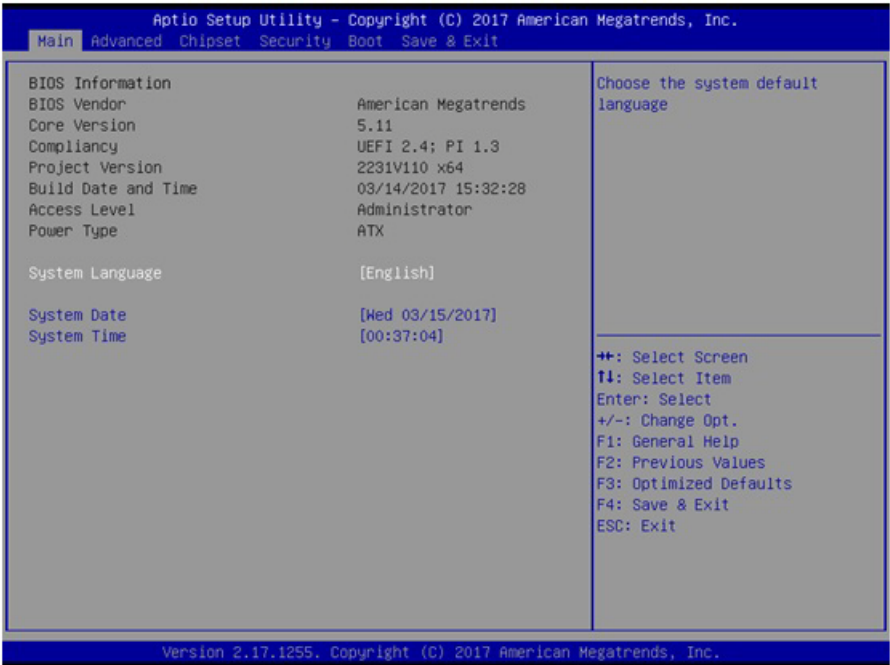


图 4.1： 设置程序初始页面

BIOS ROM 带有内置的设置程序，允许用户修改基本系统配置信息。这些信息保存在由电池供电的 CMOS 中，因此在电源关闭时仍不会丢失。

## 4.2 进入设置界面

开启计算机后，屏幕上将出现 POST（上电自检），显示 BIOS 和 CPU 信息。按 <Del> 键即可进入 BIOS 设置界面。

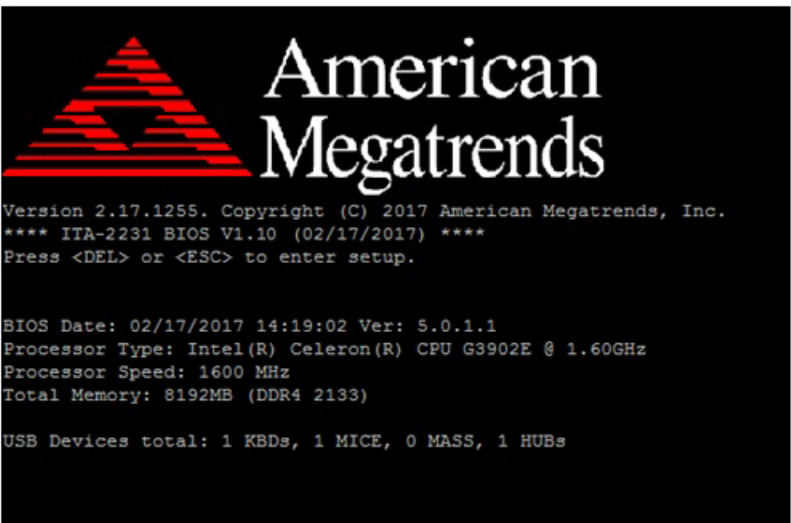


图 4.2： 按 <DEL> 键进入设置界面

### 4.2.1 主设置界面

首次进入设置实用程序时，即会进入主设置页面。点击“Main”标签即可随时返回主菜单。本节共介绍共有 2 个主设置选项。BIOS 主设置页面如下图所示：



图 4.3：主设置界面

BIOS 设置主菜单由 2 部分构成。左栏显示的是可以配置的所有项。灰色的为不可配置的项，蓝色的则相反。右边栏为箭头图例。箭头上方的空白区域是为文字信息预留的。如果在左边栏选择了某项，该项将以加亮白色字体显示，且在预留的文字信息处显示简介文字。

#### ■ System Time/System Date

此项可用于改变系统时间和日期。用户可使用方向键选中系统时间或系统日期，并可通过键盘输入新值。按 Tab 键或方向键可在各项间进行切换。日期的格式为 MM/DD/YY，时间的格式为 HH:MM:SS。

### 4.2.2 高级 BIOS 特性设置

从 ITA-2231 系列的设置界面点击“Advanced”标签即可进入高级 BIOS 设置界面。用户可选择左边边框中的项进入相应项的子菜单，如“CPU Configuration”。用户可使用方向键来选中相应的设置项。所有的高级 BIOS 设置选项都在此节中进行描述。如下图所示。子菜单将在后面进行介绍。

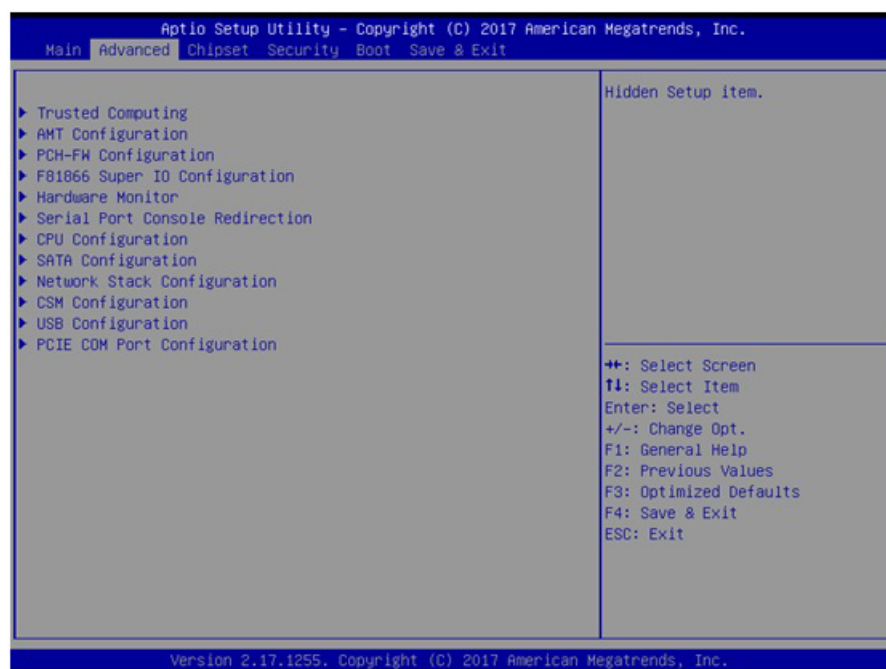


图 4.4: 高级 BIOS 特性设置界面

#### 4.2.2.1 Trusted Computing

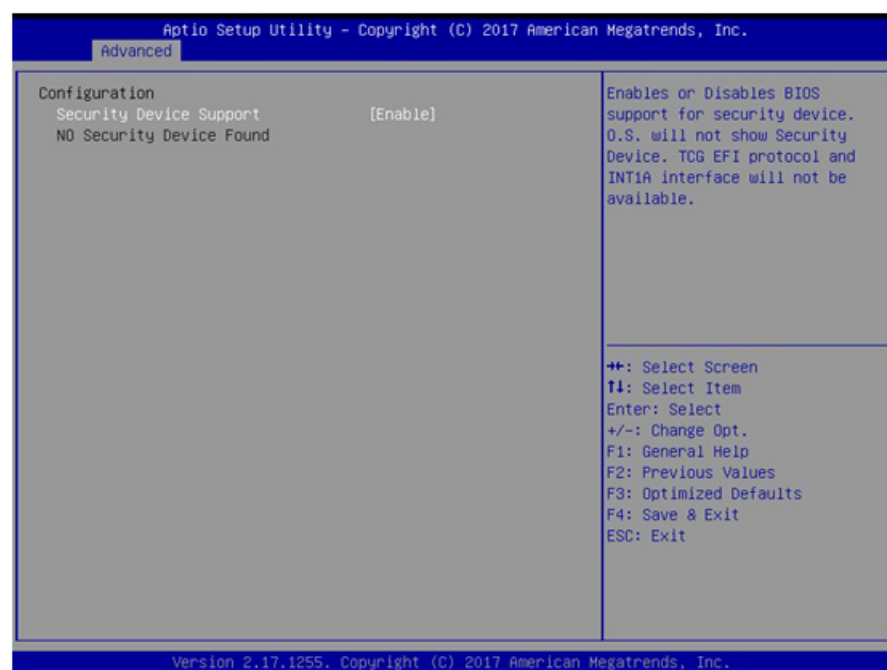


图 4.5: Trusted Computing 设置

- **Security Device Support**  
用于显示 BIOS 是否支持 TPM。

#### 4. 2. 2. 2 AMT Configuration

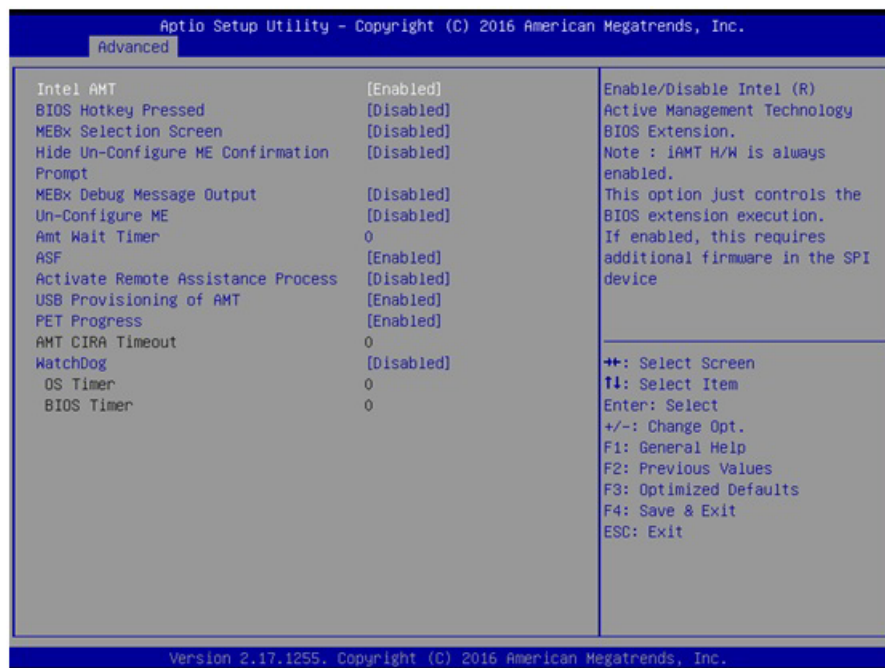


图 4.6: AMT Configuration 设置

- **Intel AMT**  
用于显示 BIOS 是否支持 AMT。
- **BIOS Hotkey Pressed**  
是否支持 BIOS 使用热键去触发 AMT。
- **MEBX Selection Screen**  
是否显示 MEBX 提示选项。
- **Hide Un-Configure ME Configuration Prompt**  
是否隐藏初始化 ME 配置的选项。
- **MEBX Debug Message Output**  
是否显示 MEBX 调试信息输出。
- **Un-Configure ME**  
是否初始化 AMT。
- **AMT Wait Timer**  
设置 AMT 等待时间。
- **ASF**  
是否支持 ASF。
- **Activate Remote Assistance Process**  
是否激活远程协助。
- **USB Provisioning of AMT**  
是否支持 AMT USB 配置。
- **PET Process**  
用户是否使用 PET Event 处理 PET 事件。
- **WatchDog**  
是否开启 AMT 开门狗。

4. 2. 2. 3 PCH-FW Configuration

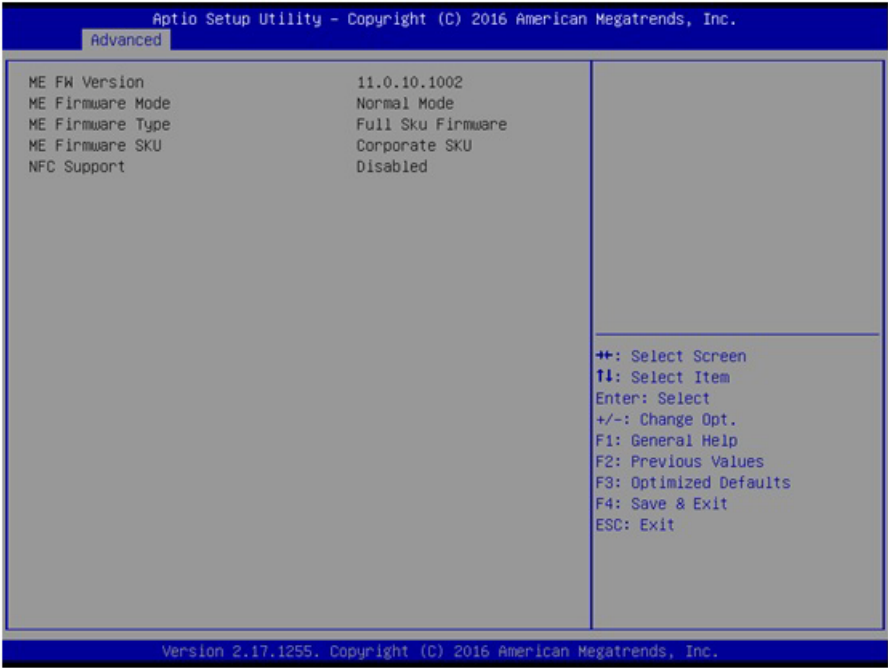


图 4.7: PCH-FW Configuration

显示当前 ME 的版本和型号。

4. 2. 2. 4 F81866 Super IO Configuration

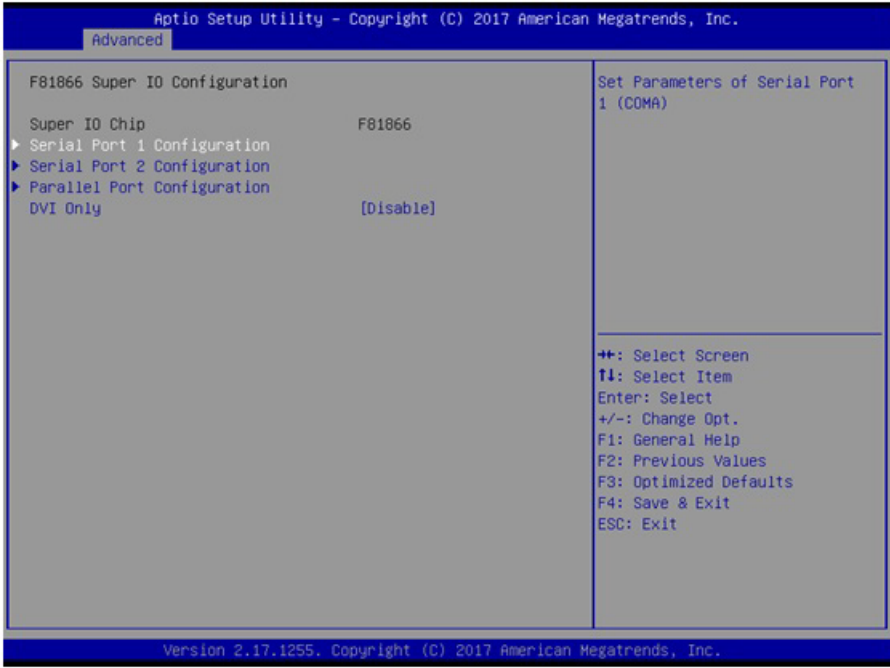


图 4.8: 设置 F81866 Super IO Configuration

- DVI only  
用户选择是否只接入 DVI 显示设备。

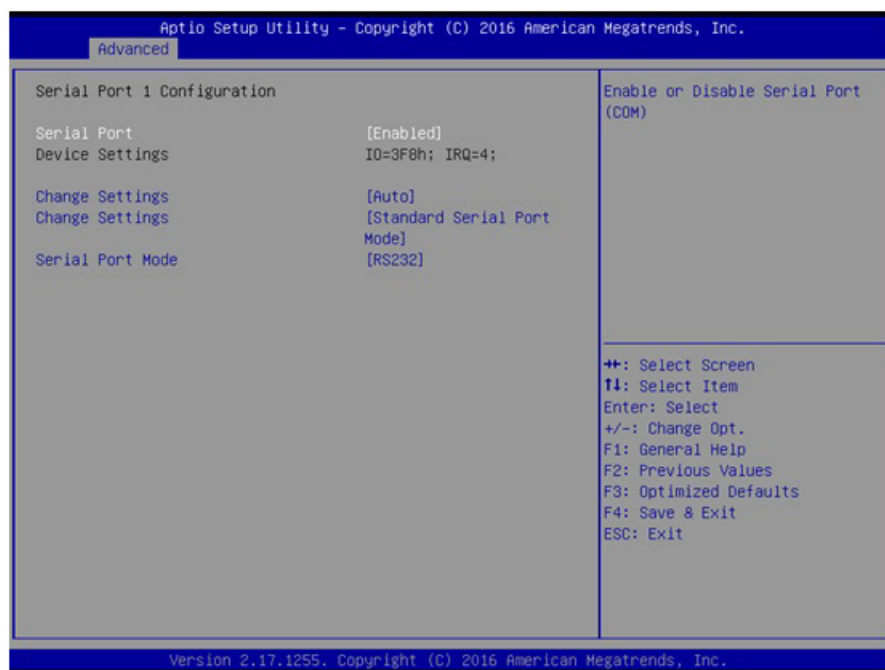


图 4.9： 设置串口的设置

- **Serial Port**  
是否开启串口。默认的设置开启。
- **Device Settings**  
显示串口当前的 IO 端口和 IRQ 数值。
- **Change settings[1]**  
选择串口的 IO 端口和 IRQ 数值。
- **Change settings[2]**  
选择串口的工作模式（半双工 / 全双工）。
- **Serial Port Mode**  
选择串口的模式（RS232/RS485）。

4. 2. 2. 5 Hardware Monitor

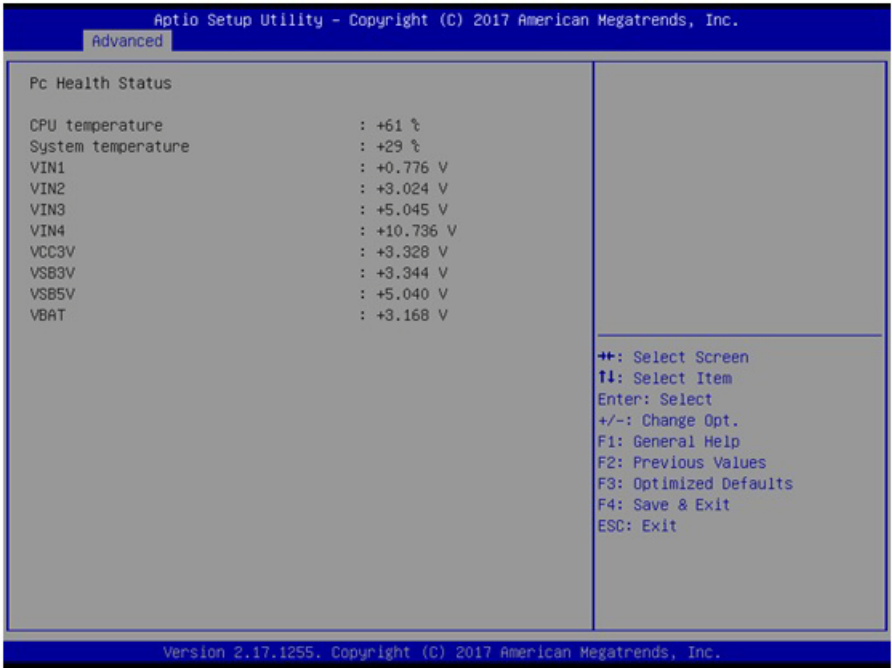


图 4.10: Hardware Monitor

此项用于做硬件安全侦测，BIOS 将显示当前的系统温度，CPU 温度，以及其他相关的电压值。以上参数都有一定的范围，系统不可以超过这些范围运行。

4. 2. 2. 6 Serial Port Console Redirection

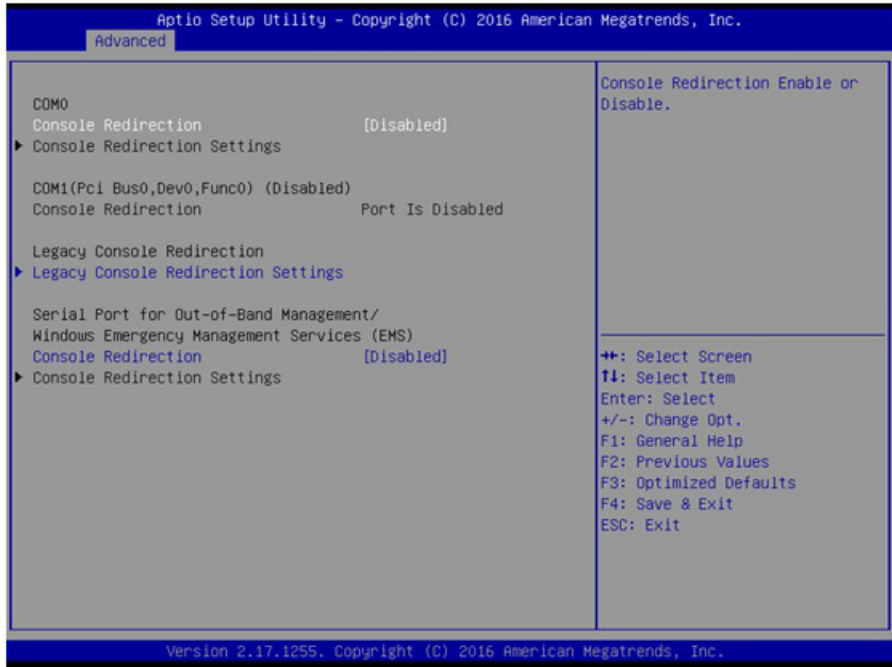
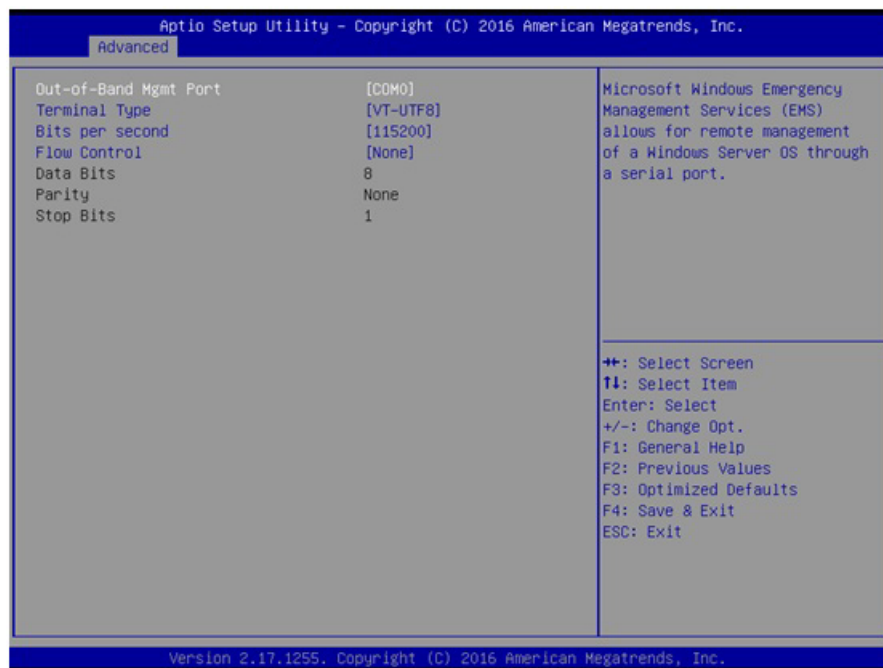


图 4.11: Serial Port Console Redirection

- Console redirection  
设置是否开启串口重定向功能。





#### ■ Console Redirection Settings

当开启串口重定向功能后，可以进入子页面进行详细的重定向设定。在这个页面中可以设定终端的类型和传输速率等功能。

#### 4.2.2.7 CPU Configuration

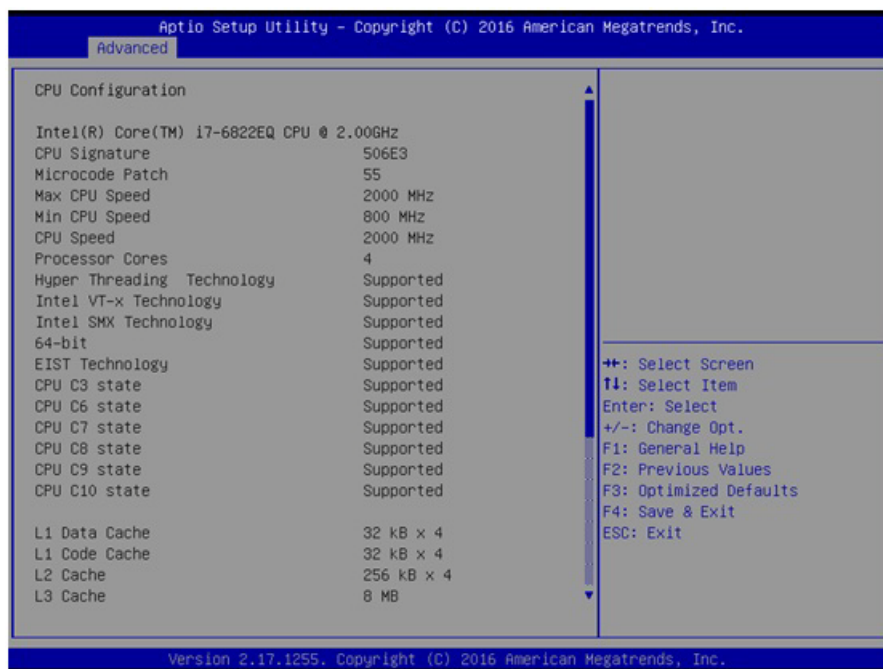
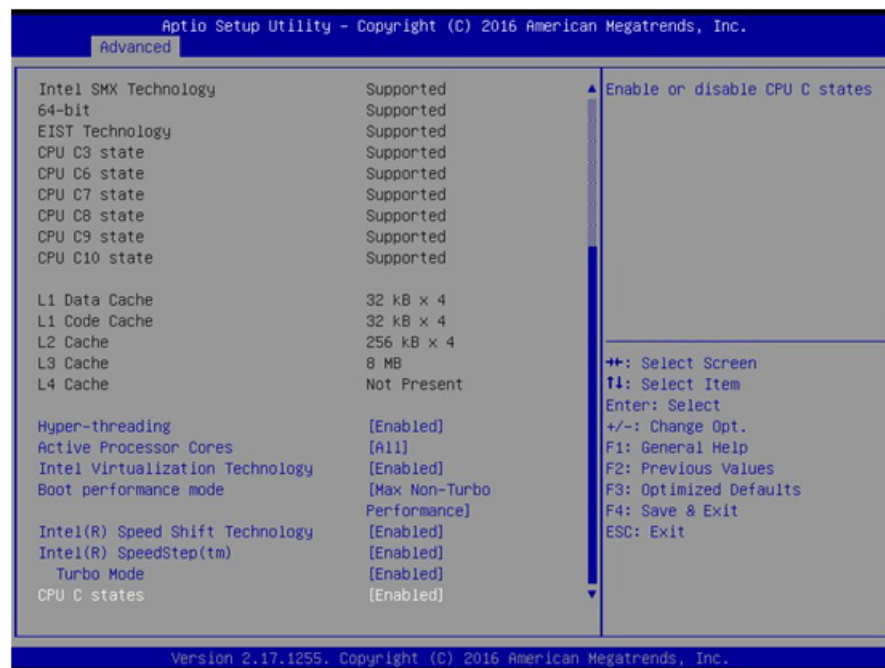


图 4.12: CPU Configuration



- **Hyper-threading**  
设置是否开启超线程。
- **Active Processor Cores**  
设置处理器的核心数。
- **Intel Virtualization Technology**  
是否开启 VT 技术。
- **Boot performance mode**  
设置 BIOS 在移交给操作系统前的 CPU 性能。
- **Intel(R) Speed shift Technology**  
是否开启 Intel(R) Speed shift Technology。
- **CPU C states**  
是否开启 C-State。

#### 4. 2. 2. 8 SATA Configuration

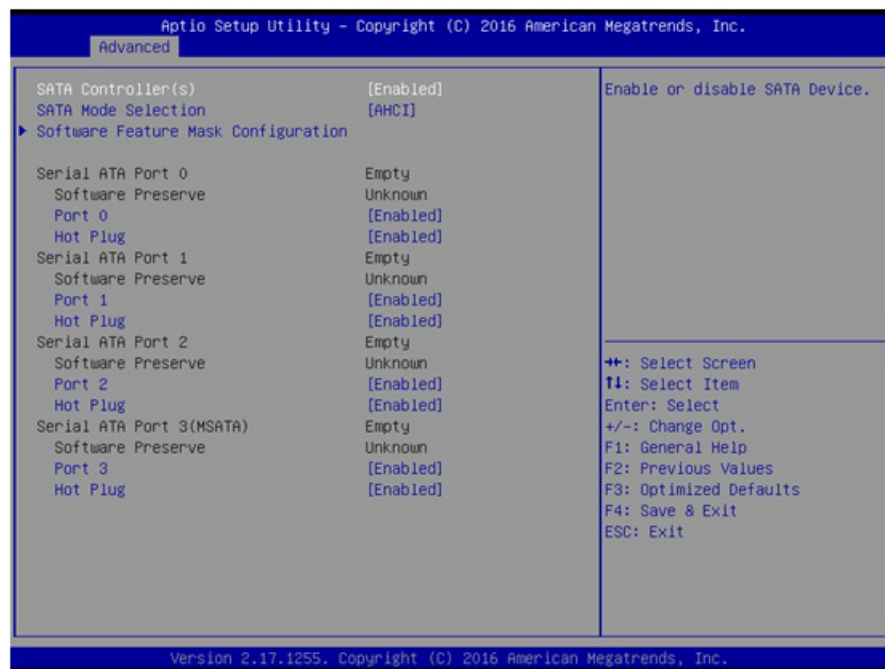


图 4.13: SATA Configuration

- **SATA Controller(s)**  
是否开启 SATA 控制器。
- **SATA Mode Selection**  
选择 SATA 的模式（AHCI 或者 RAID）。
- **SATA Port 0/1/2/MSATA**  
是否开启 SATA 端口 0/1/2。
- **Hot plug**  
设置 SATA 端口是否支持热插拔。

4.2.2.9 Network Stack Configuration

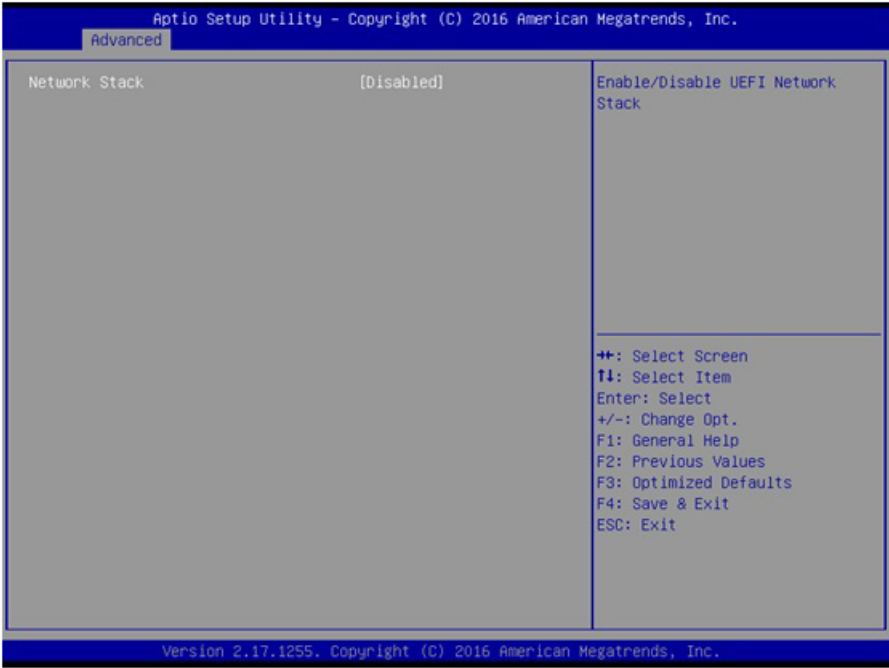
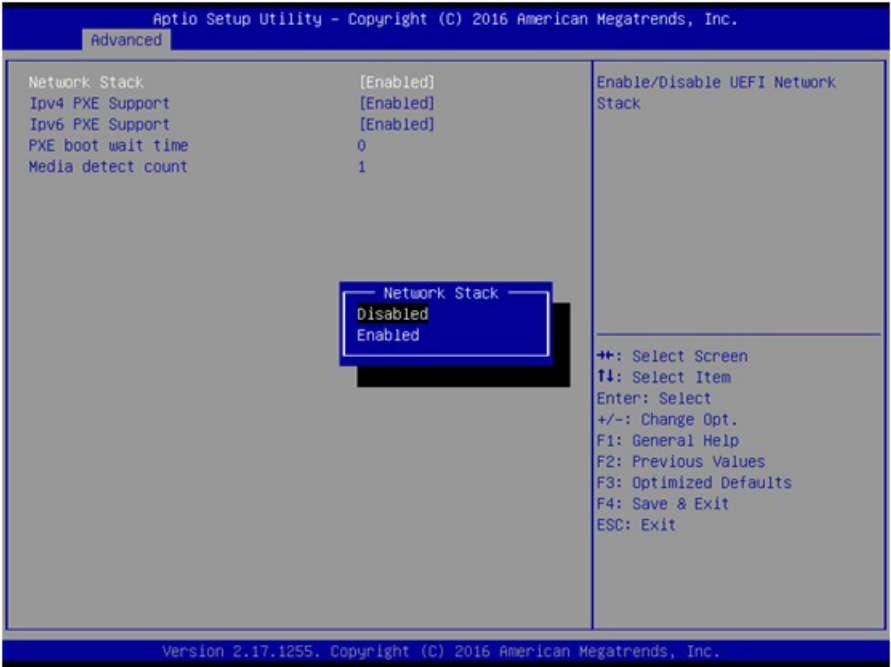


图 4.14: Network Stack Configuration

- **Network Stack**  
是否开启 UEFI 网络堆栈。



当开启网络堆栈后，会出现上图的设置。

- **Ipv4 PXE Support**  
是否开启对 IPV4 PXE 的支持。
- **Ipv6 PXE Support**  
是否开启对 IPV6 PXE 的支持。

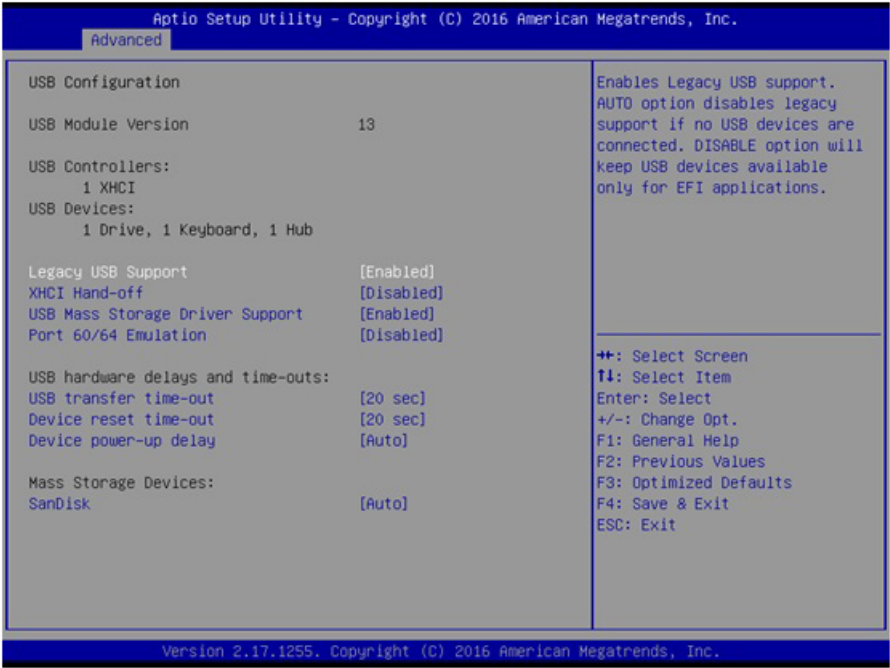
- **PXE boot wait time**  
设置等待按 ESC 键忽略 PXE 启动的时间。
- **Media detect count**  
设置检查多媒体的次数。

#### 4. 2. 2. 10 CSM Configuration



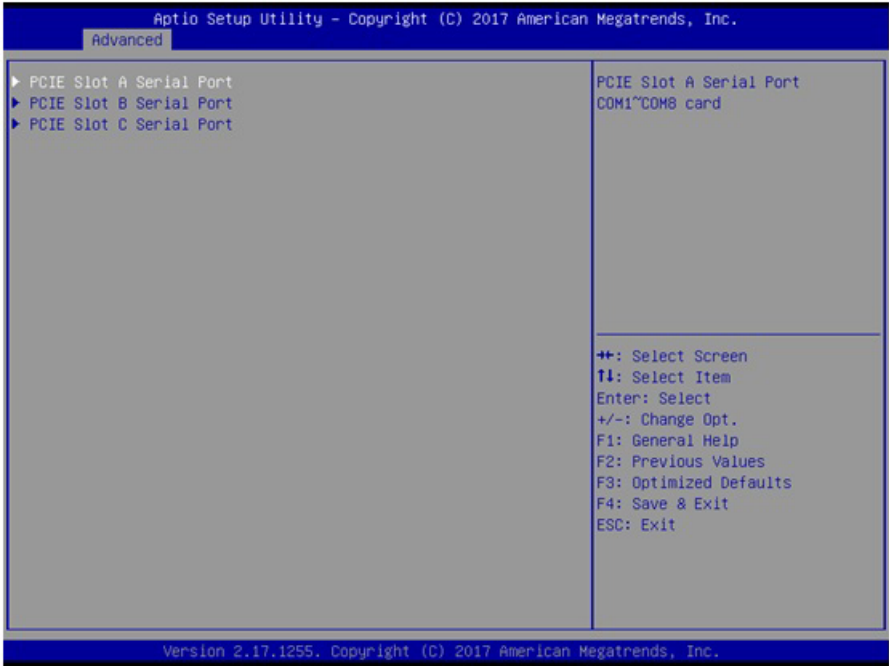
- **CSM Support**  
是否支持 CSM。
- **GateA20 Active**  
选择 GateA20 激活的时间。
- **Option ROM Messages**  
设置 option ROM 显示信息的模式。
- **INT19 Trap Response**  
设置 BIOS 执行 INT19 的时间
- **Boot option filter**  
选择 legacy/UEFI 启动的次序。
- **Network**  
选择网络执行 UEFI 还是 legacy 的 option rom。
- **Storage**  
选择存储执行 UEFI 还是 legacy 的 option rom。
- **Video**  
选择显示执行 UEFI 还是 legacy 的 option rom。
- **Other PCI devices**  
选择其他 PCI 设备执行 UEFI 还是 legacy 的 option rom。

4. 2. 2. 11USB Configuration

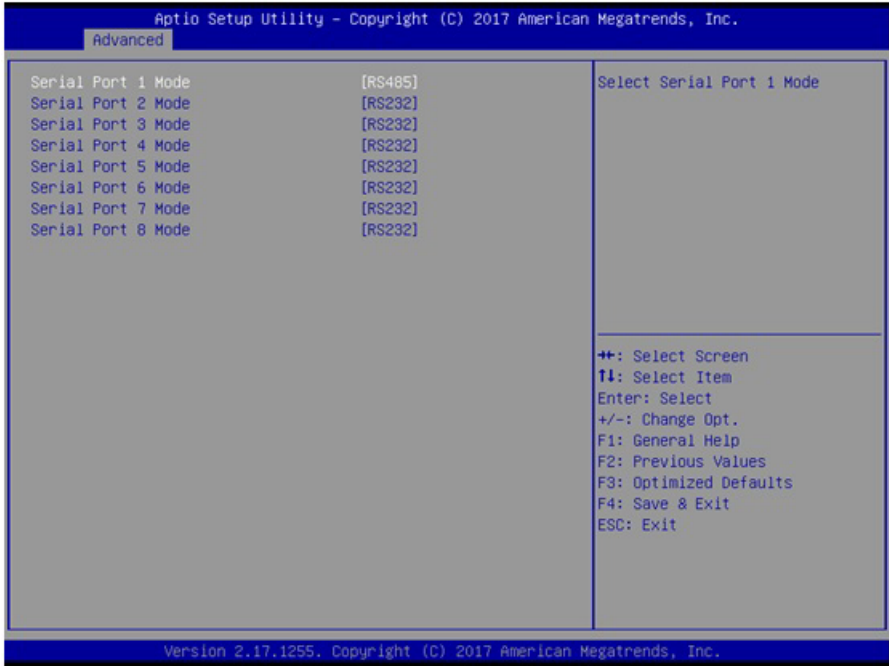


- **Legacy USB Support**  
是否支持传统 USB 设备。
- **XHCI Hand-off**  
是否支持 XHCI hand-off。是对操作系统没有 XHCI hand-off 的可变通方案。
- **USB Mass Storage Driver Support**  
是否支持 USB 存储设备。
- **Port 60/64 Emulation**  
是否打开 60/64 端口模拟的支持。对于不支持 usb 的操作系统，这个选项应该打开。
- **USB transfer time-out**  
设置 usb 传输数据的超时时间。
- **Device power-up delay**  
设置 usb 存储设备开始执行命令的超时时间。

4. 2. 2. 12PCIE COM Port Configuration



选择相应的 PCIE 串口卡进行设置。



- **Serial Port Mode**  
设置串口的模式，包括 RS232，RS422 以及 RS485。

### 4.2.3 高级芯片组特性设置



- **VT-d**  
是否支持 VT-d 技术。
- **Above 4GB MMIO BIOS assignment**  
是否开启超出 4G MMIO 的 BIOS 分配。



#### 4.2.3.1 Graphics Configuration

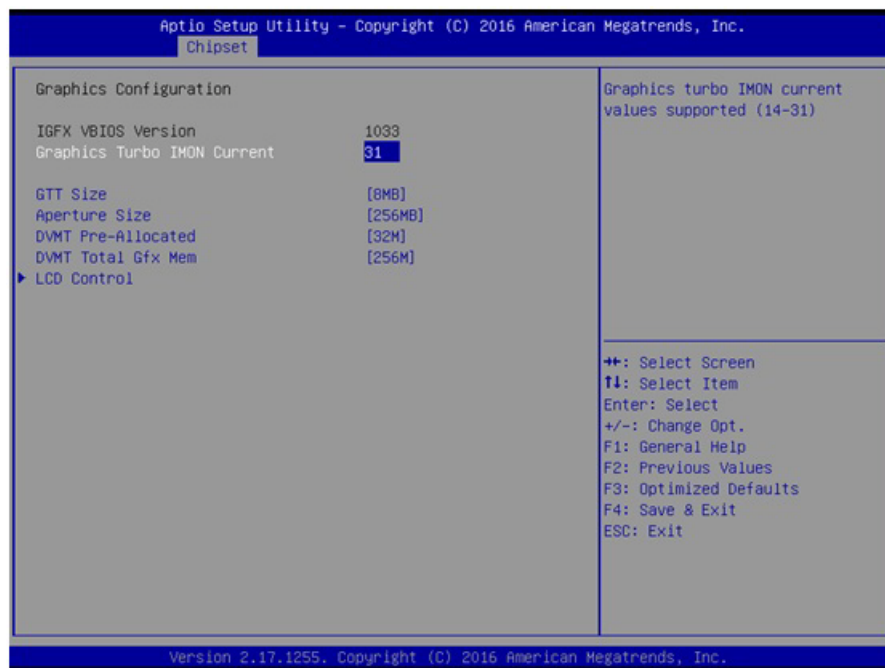


图 4.15: Intel Graphics Configuration

- **IGFX VBIOS Version**  
显示当前 VBIOS 的版本。
- **Graphics Turbo IMON Current**  
设置显示的 IMON 电流值。
- **GTT size**  
设置 GTT 大小。
- **Aperture size**  
设置 aperture 大小。
- **DVMT Pre-Allocated**  
设置 DVMT5.0 固定的显存大小
- **DVMT Total GFX Mem**  
设置 DVMT5.0 总共可用的显存大小。
- **LCD control**  
请参考下图。



- **Primary IGFX Boot Display**  
选择 POST 阶段可以显示的设备。ITA-2231 支持 HDMI，DVI，EDP 屏幕和 VGA 显示。
- **eDP LVDS Panel Type**  
选择 ITB-172 支持的屏幕类型。
- **Panel Scaling**  
是否选择屏幕自动缩放。
- **Backlight Control**  
选择控制背光的类型。
- **Spread Spectrum clock chip**  
设置展频的控制方式。
- **Active LFP**  
配置 LFP 的端口。
- **Panel Color Depth**  
选择 LFP 屏幕的色彩深度。

#### 4.2.3.2 PEG Port Configuration

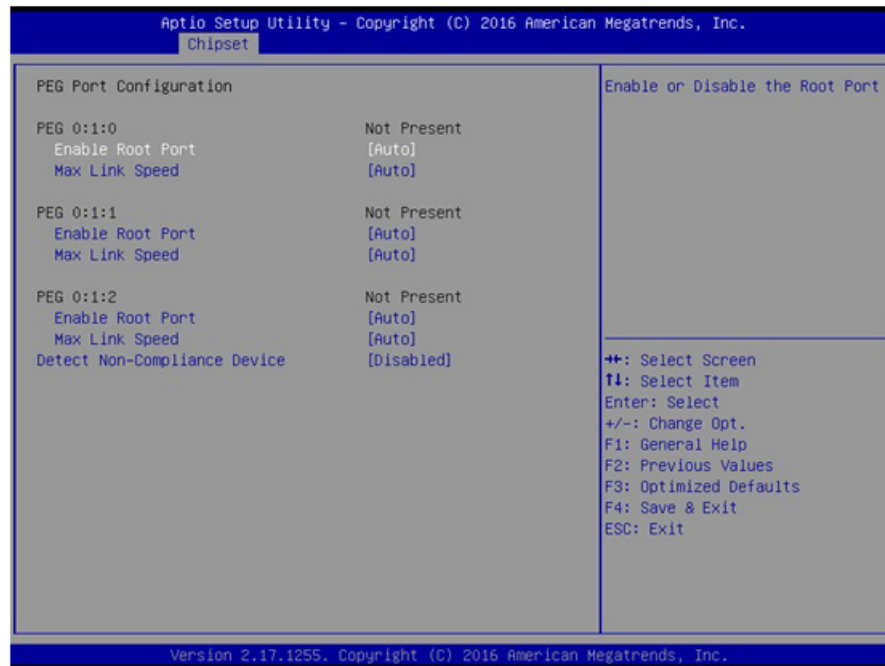
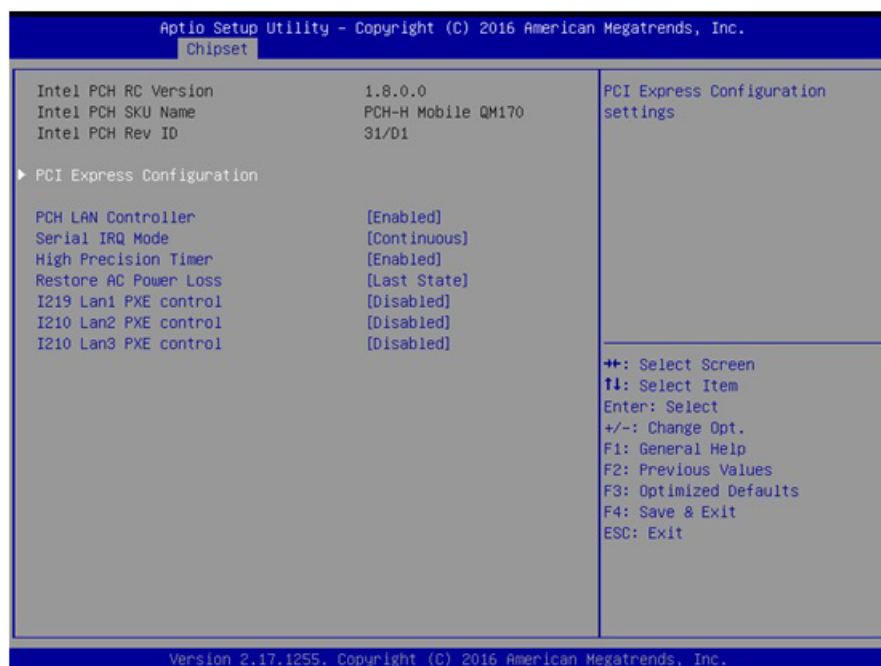


图 4.16: PEG Port Configuration

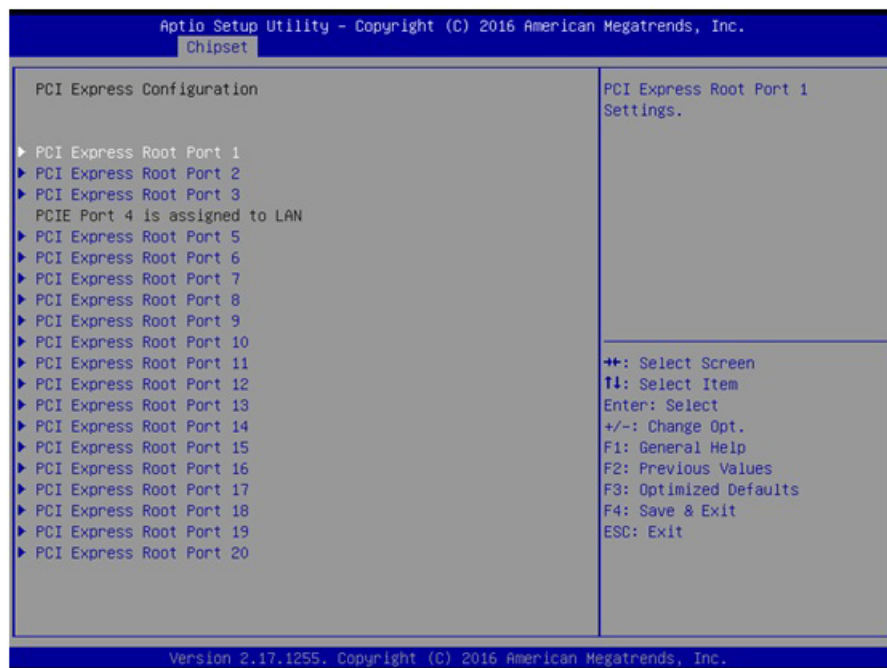
- **Enable Root Port**  
是否开启端口。
- **Max Link Speed**  
配置 PEG 端口最大的连接速率。
- **Detect Non-Compliance Device**  
是否检测不兼容设备。

#### PCH-I/O Configurations



- **Intel PCH RC Version**  
显示当前的 Intel PCH RC 版本。
- **Intel PCH SKU Name**  
显示当前的 Intel PCH 型号。
- **Intel PCH Rev ID**  
显示当前 Intel PCH 的版本信息。
- **PCI Express Configuration**  
请参考下一页。
- **PCH LAN Controller**  
是否开启 i219 网络控制器。
- **Serial IRQ Mode**  
配置 Serial IRQ 模式。
- **High Precision Timer**  
是否开启高精度时钟。
- **Restore AC Power Loss**  
选择下一次插入 AC 电源的状态。
- **I219/I210 LAN PXE Control**  
是否开启 I210/I219 的网络 PXE 控制。

## PCI Express Configuration



选择需要控制的 PCIE 端口。

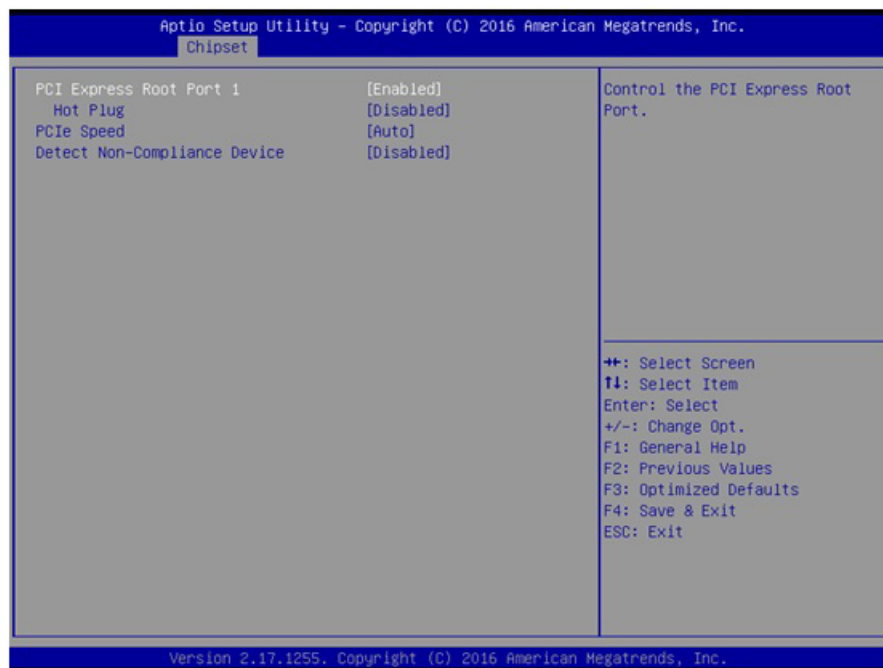


图 4.17: 设置 PCI Express Configuration

- **PCI Express Root Port**  
是否开启当前端口。
- **Hot plug**  
是否支持 PCIE 设备热插拔。
- **PCIe speed**  
设置 PCIE 端口速率。
- **Detect Non-Compliance device**  
是否检测不兼容设备。

## 4.3 启动设置

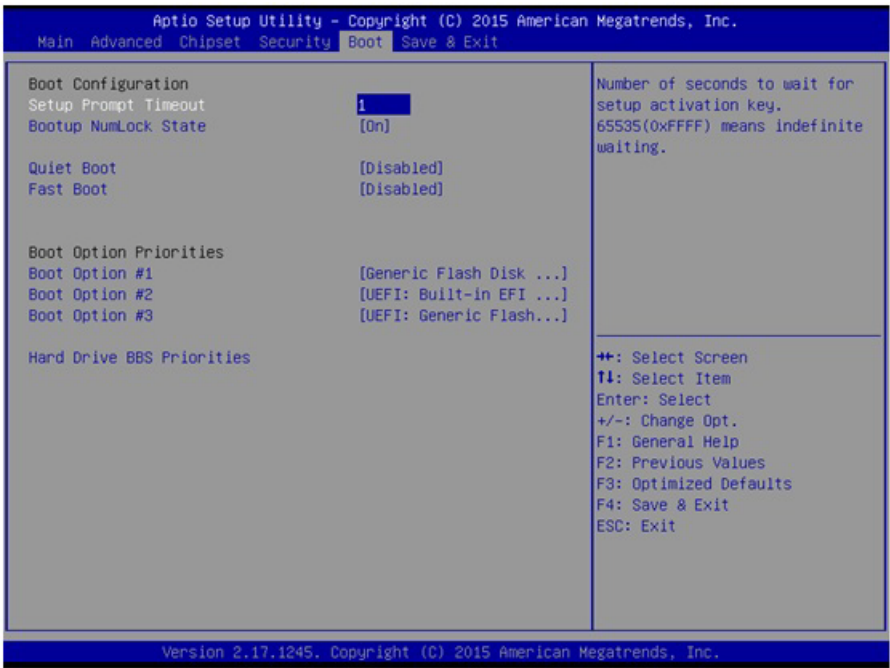


图 4.18： 启动设置

### 4.3.1 Boot Configuration

- **Setup Prompt Timeout**  
此项设置是按 Setup 快捷键的等待时间。如果在设置时间内没有按 Setup 快捷键就继续启动。
- **Bootup NumLock State**  
此项设置允许在系统上电至 DOS 系统后激活小键盘的数字锁功能。默认设置为“On”。  
On: 系统启动时处于数字锁开。  
Off: 启动时小键盘处于光标控制状态。
- **Quiet Boot**  
如果设置为“Disabled”，则 BIOS 将显示正常的 POST 信息。如果设置为“Enabled”屏幕上将出现 OEM 图标，而非 POST 信息。
- **Fast Boot**  
此项允许 BIOS 在启动过程中跳过一些检测步骤，从而减少系统启动的时间。默认设置为“Disabled”

### 4.3.2 Hard Drive BBS Priorities

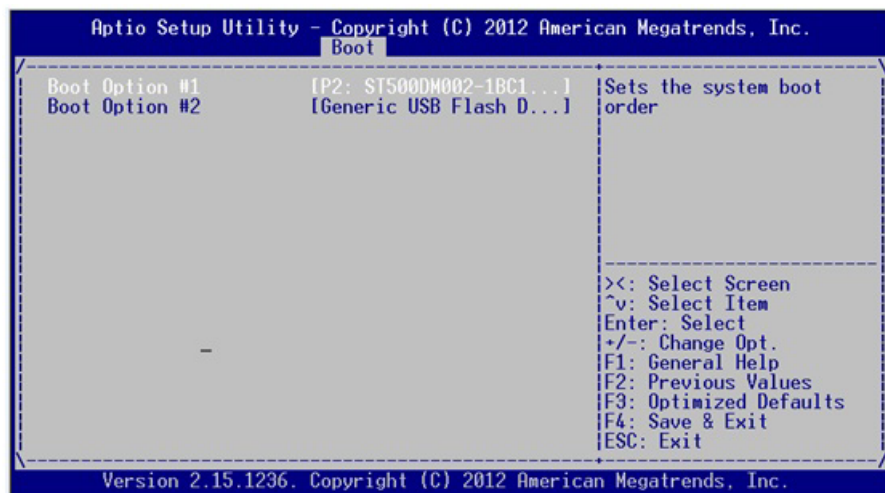


图 4.19: Hard Drive BBS Priorities

此项用于选择启动设备的启动顺序。

## 4.4 安全设置

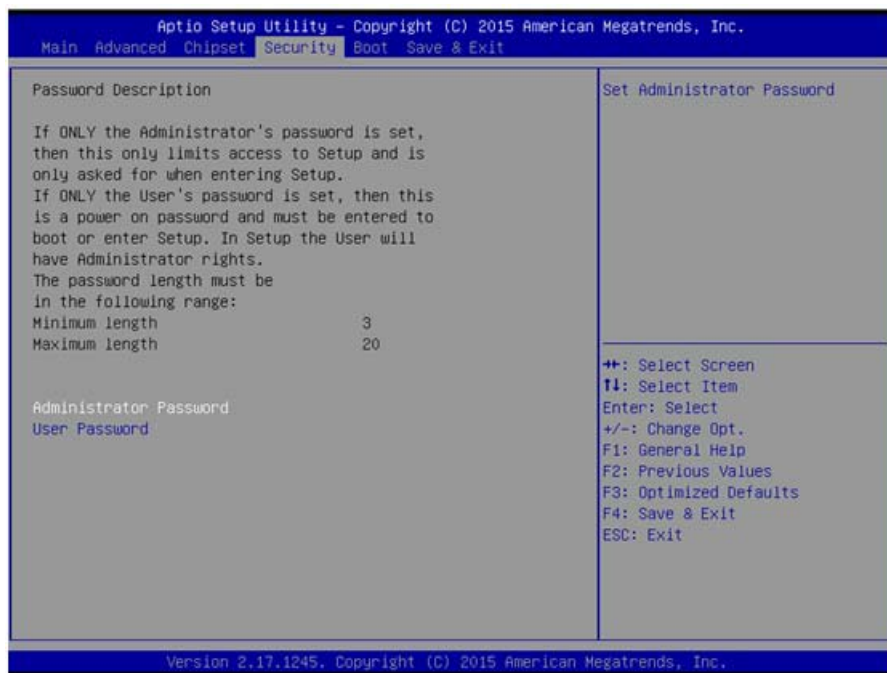


图 4.20: “Security” 设置界面

- **Administrator Password**  
用于设置管理员密码。
- **User Password**  
用于设置用户密码。

## 4.5 Save & Exit 设置

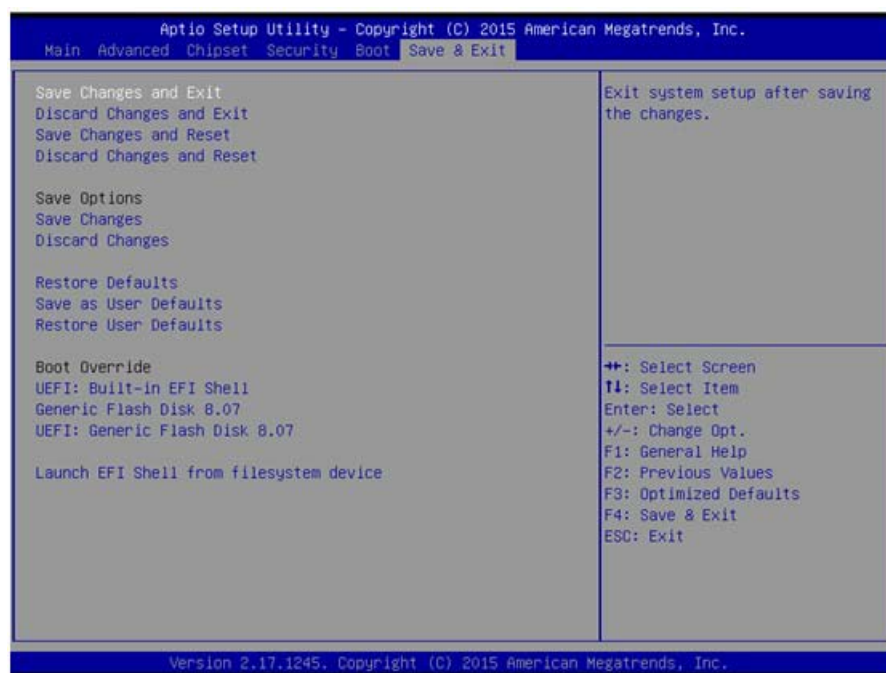


图 4.21: “Setup Save & Exit Menu” 界面

### ■ Save Changes and Exit

如果已完成系统配置，请选择此项退出设置。

1. 请从退出菜单中选择 “Save Changes and Exit” 并按下 <Enter> 键。屏幕上将出现以下信息：  
Save Configuration Changes and Exit Now?  
[Ok] [Cancel]

2. 选择 “Ok” 或 “Cancel”。

### ■ Discard Changes and Exit

选择此项将退出设置且不保存对系统配置的任何修改。

1. 请从退出菜单中选择 “Discard Changes and Exit” 并按下 <Enter> 键。屏幕上将出现以下信息：  
Discard Changes and Exit Setup Now?  
[Ok] [Cancel]

2. 选择 Ok 将放弃修改并退出设置。

### ■ Save Changes and Reset

如果用户已经完成系统配置，请选择此项保存修改，退出 BIOS 设置菜单并重启后系统配置参数即可生效。

1. 选择保存修改并重置并按下 <Enter> 键。屏幕上将出现以下信息：  
Save configuration and Reset? [Yes] [No]

2. 选择 “Ok” 或 “Cancel”。

### ■ Discard Changes and Reset

选择此项将退出设置且不保存对系统配置的任何修改。

1. 请从退出菜单中选择 “Discard Changes and Reset” 并按下 <Enter> 键。屏幕上将出现以下信息：  
Discard Changes and Reset Setup Now?  
[Ok] [Cancel]

2. 选择 Ok 将放弃修改并退出设置。



- **Save Changes**  
允许保存用户所作的任何修改。
- **Discard Changes**  
允许放弃用户所作的任何修改。
- **Restore Default**  
用于恢复系统默认值。
- **Save as User Default**  
用于将所作的修改保存为用户默认值。
- **Restore User Default**  
用于恢复用户默认值。
- **Boot Override**  
用于设置开机的装置。
- **Launch EFI Shell from file system device**  
允许从系统文件设备启动 EFI shell。



## 第 5 章

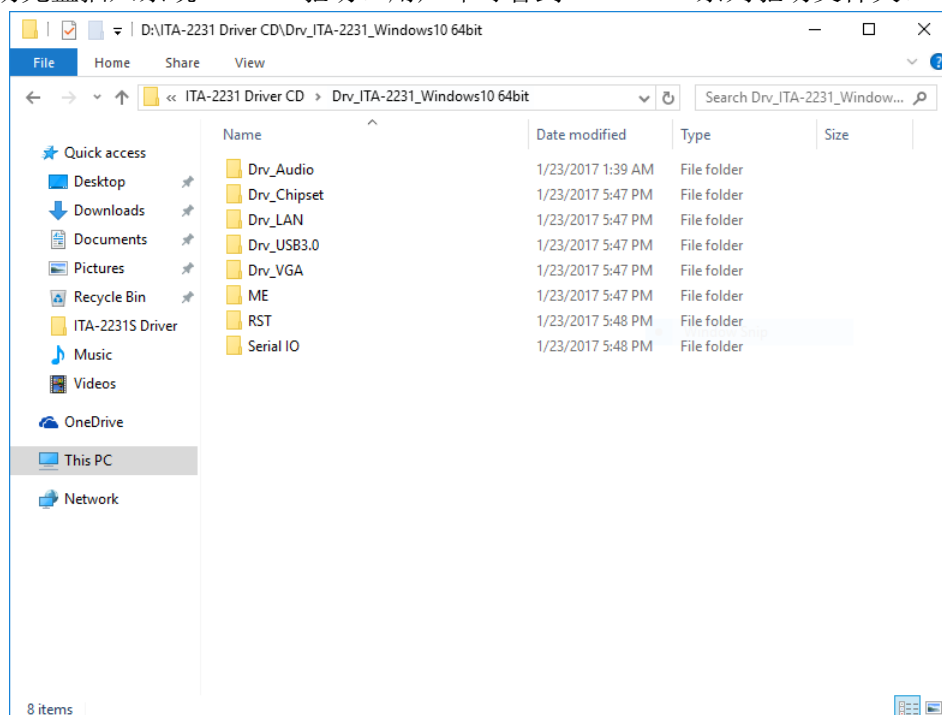
驱动安装

## 5.1 简介

研华为 Windows 程序开发人员提供了完整的设备驱动和软件。该设备驱动可应用于最通用的 Windows 编程工具中，如 Visual C++、Visual Basic、Borland C++ Builder 和 Borland Delphi。

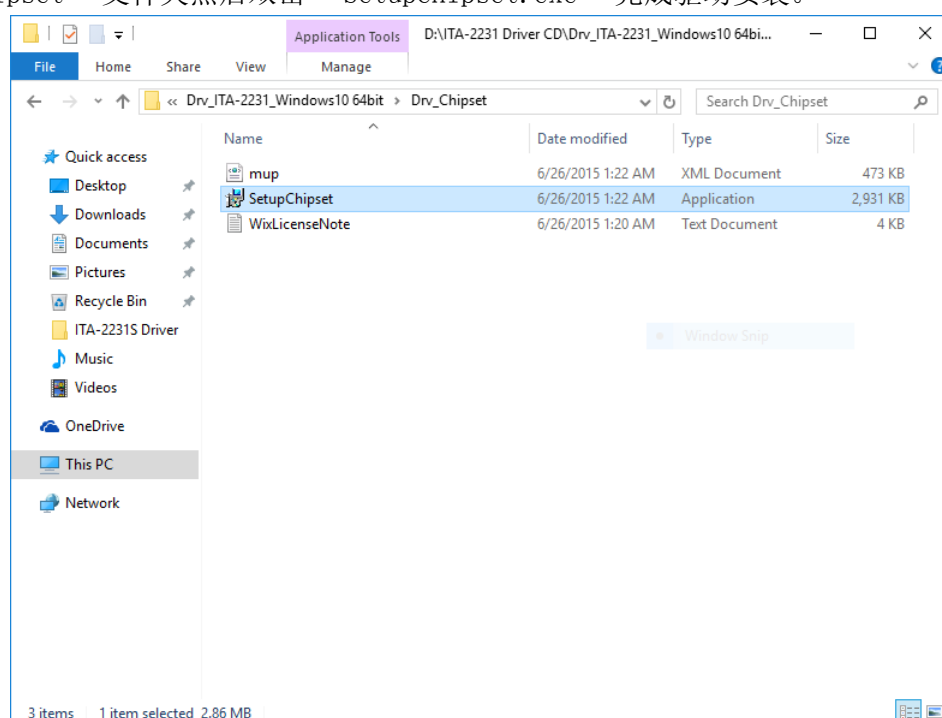
## 5.2 驱动安装

将驱动光盘插入系统 CD-ROM 驱动。用户即可看到 ITA-2231 系列驱动文件夹。



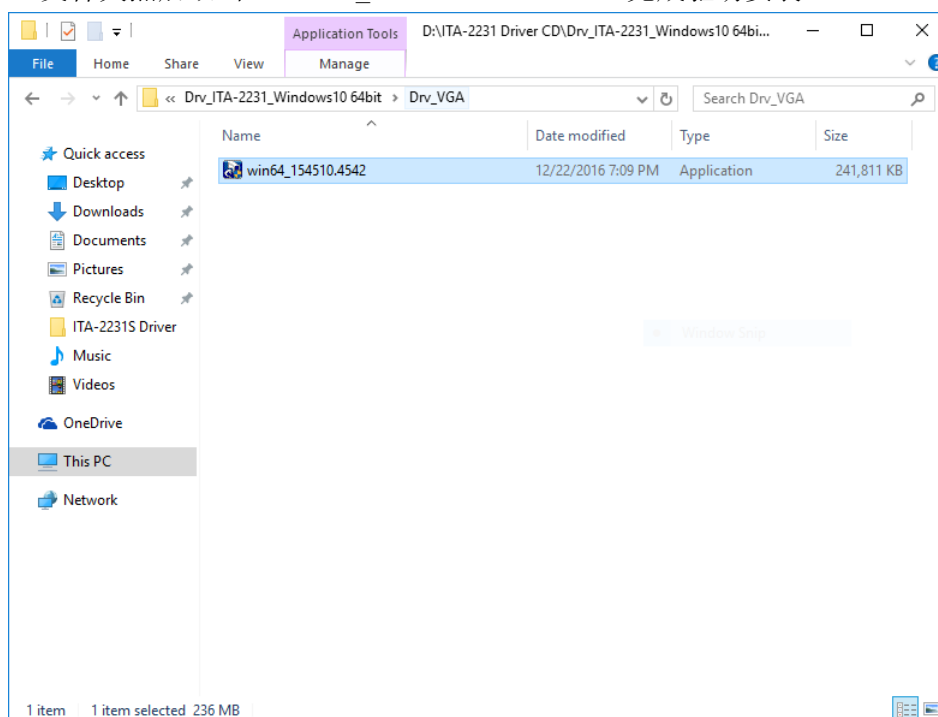
### 5.2.1 芯片组 Windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“Chipset”文件夹然后双击“SetupChipset.exe”完成驱动安装。



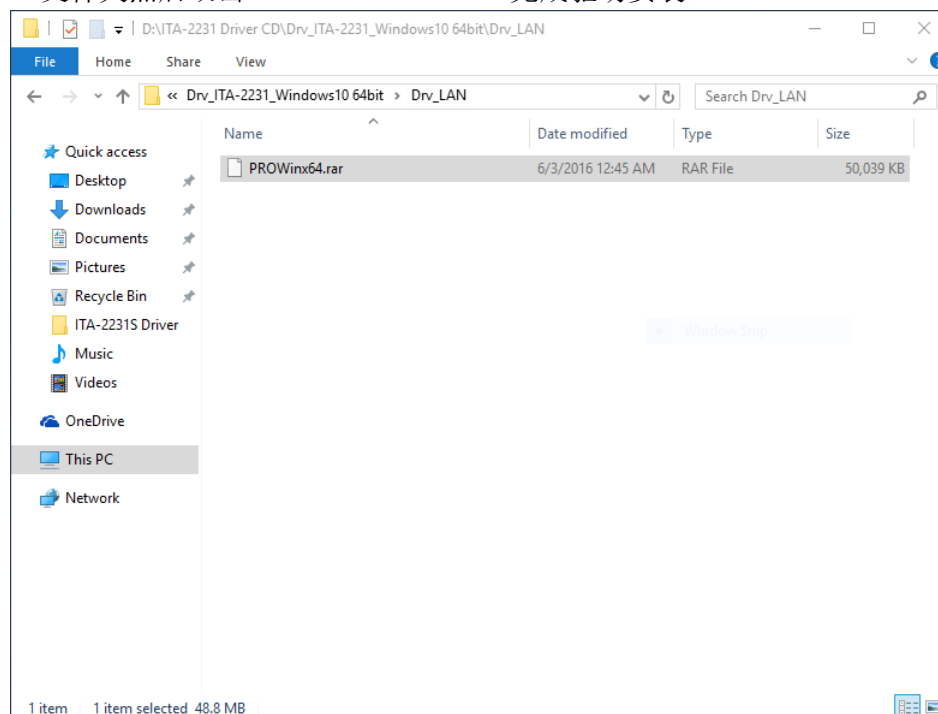
### 5.2.2 VGA windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“VGA”文件夹然后双击“win64\_154510.4542.exe”完成驱动安装。



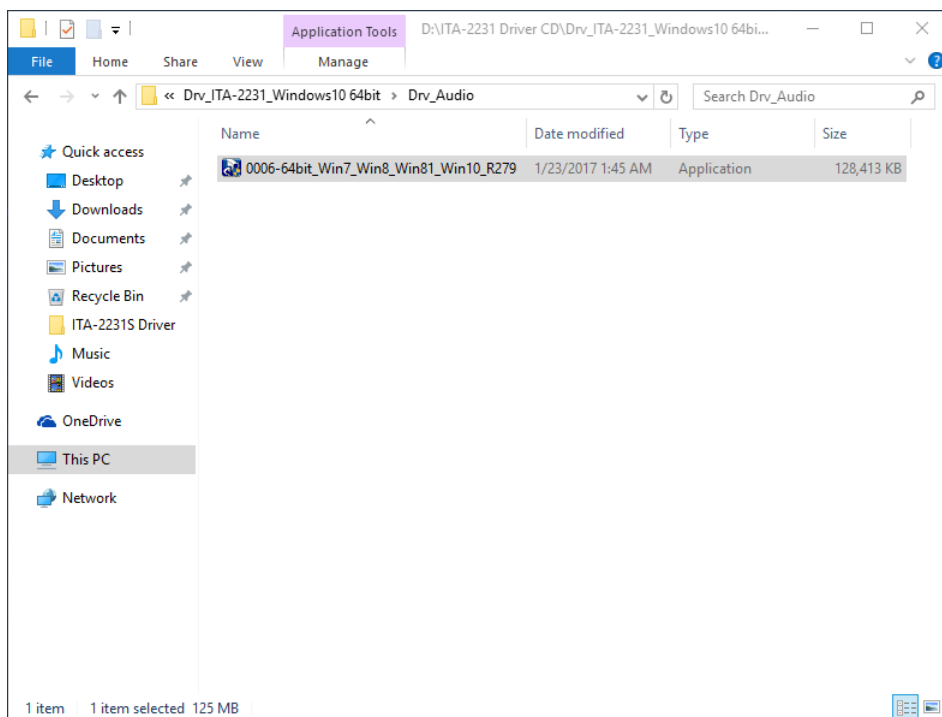
### 5.2.3 LAN windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“LAN”文件夹然后双击“PROWinx64.rar”完成驱动安装。



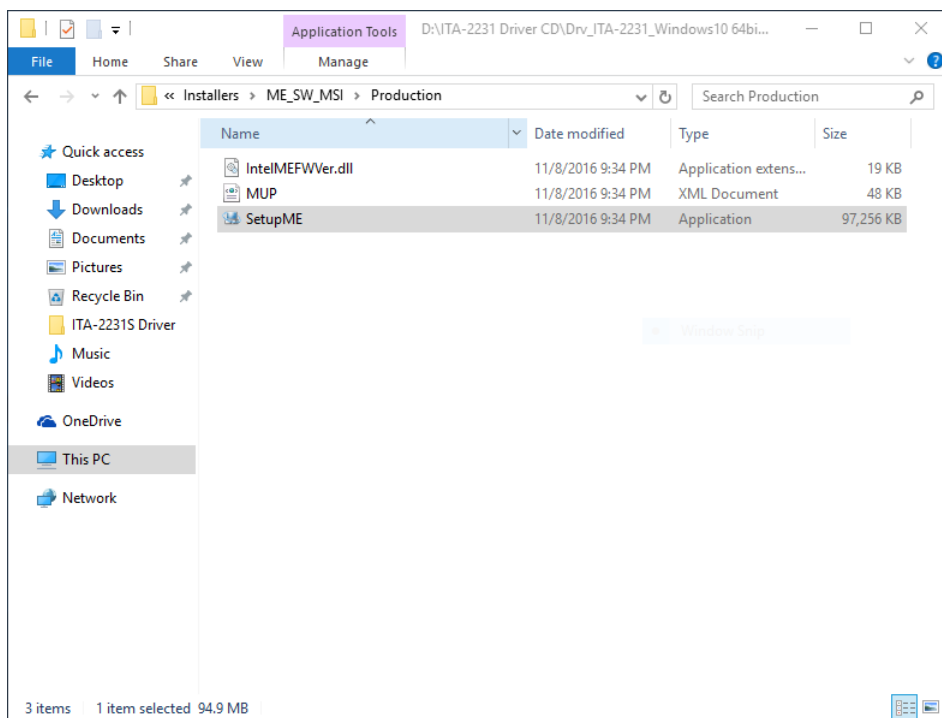
#### 5.2.4 AUDIO windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“Audio”文件夹然后双击“0006-64bit\_Win7\_Win8\_Win81\_Win10\_R279.exe”完成驱动安装。



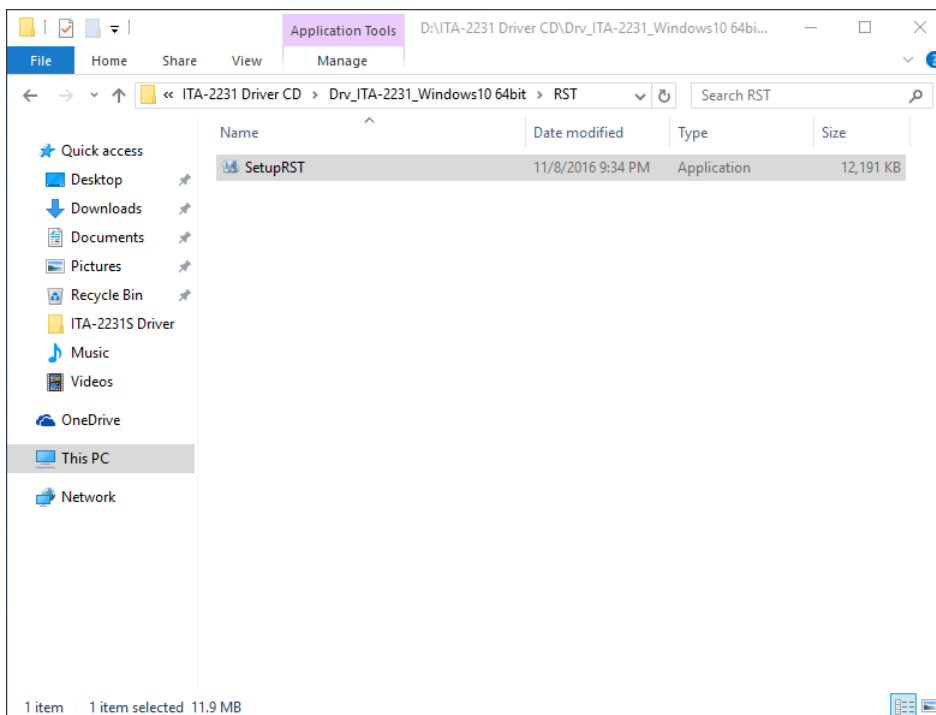
#### 5.2.5 ME windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“ME”文件夹然后双击“SetupME.exe”完成驱动安装。



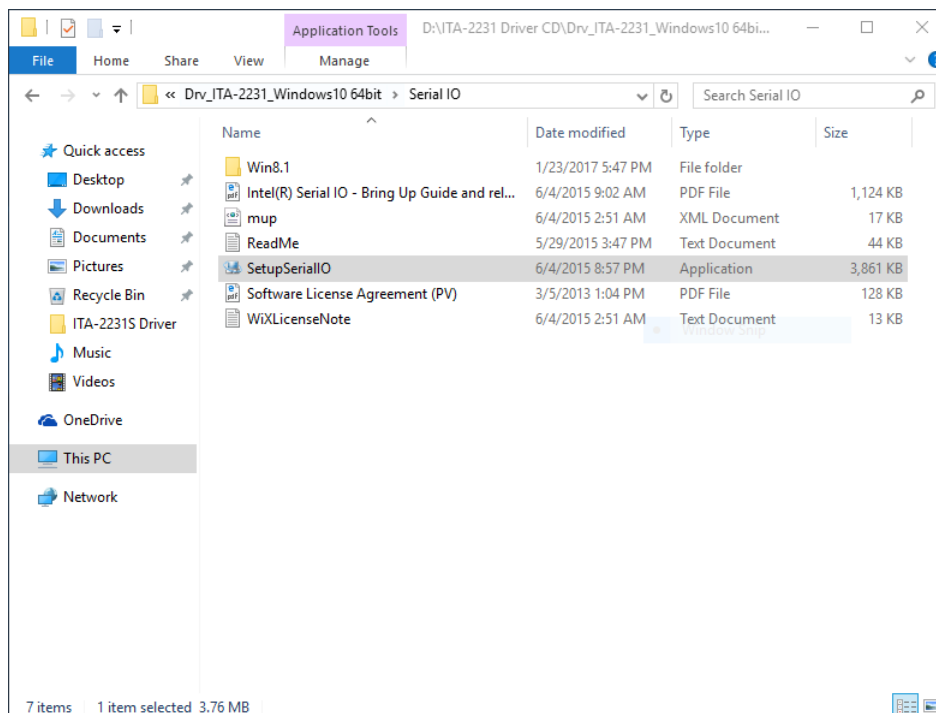
### 5.2.6 RST windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“RST”文件夹然后双击“SetupRST.exe”完成驱动安装。



### 5.2.7 Serial IO windows 驱动安装

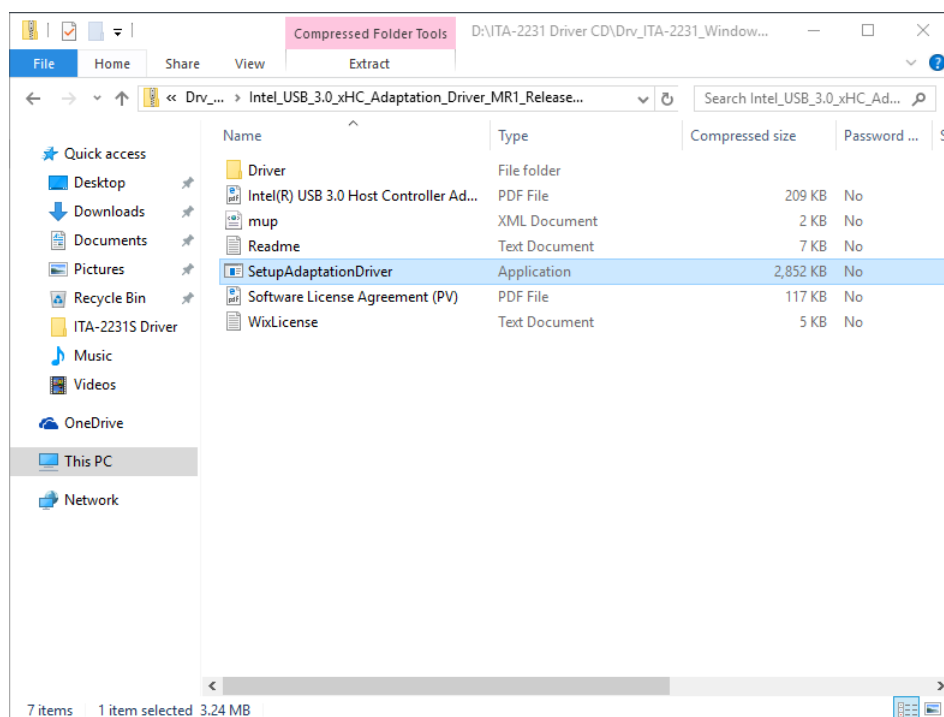
将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“Serial IO”文件夹然后双击“SetupSerialIO.exe”完成驱动安装。



### 5.2.8 USB3.0 windows 驱动安装

将驱动光盘插入到系统 CD-ROM 驱动，用户即可看到 ITA-2231 系列驱动文件夹。找到“USB 3.0”文件夹然后双击“SetupAdaptationDriver.exe”完成驱动安装。

备注：ITA-2231 在 win10 系统下不需要安装 USB3.0 驱动，直接使用 win10 OS 下自带的 USB 3.0 驱动即可。



## 5.3 搭配 ITA-EM 网卡于 Windows8.1 及 Windows10 下网口排序操作说明

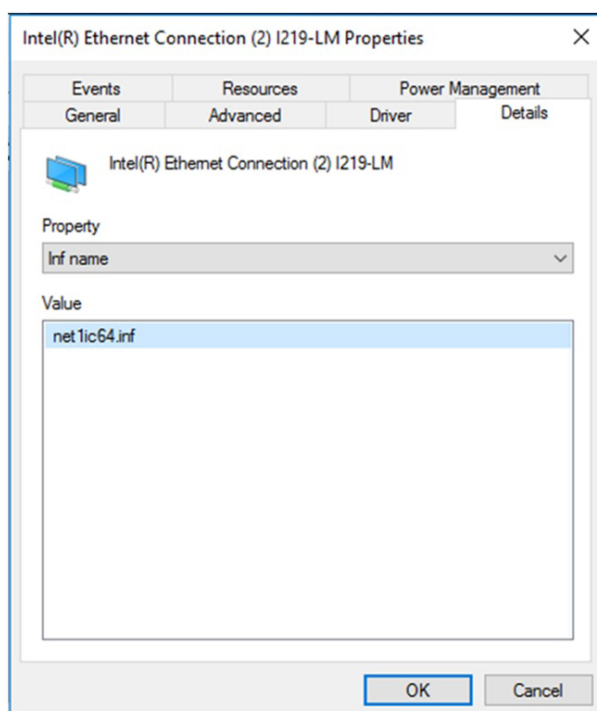
### 5.3.1 简介

因 windows 8.1 x64 系统自带驱动无法按照 ITA-EM 插槽顺序排列网口，故使用 SSID tool 将网口 ID 排序后，需要先将 windows8.1 x64 系统下的网卡驱动卸载，再重新安装对应的网卡驱动进行排序。

### 5.3.2 移除 Inbox 驱动的步骤

1. 在对应的网卡中找到 .inf 文件。





2. 按照如下操作指令，获取 .inf 的权限。

■ Cmd :

```
takeown /f "C:\Windows\INF\netlic64.inf"  
icacls "C:\Windows\INF\netlic64.inf" /grant administrators:F  
takeown /f "C:\Windows\INF\netlic64.PNF"  
icacls "C:\Windows\INF\netlic64.PNF" /grant administrators:F
```

3. 按照如下操作指令，移除 .inf 的文件。

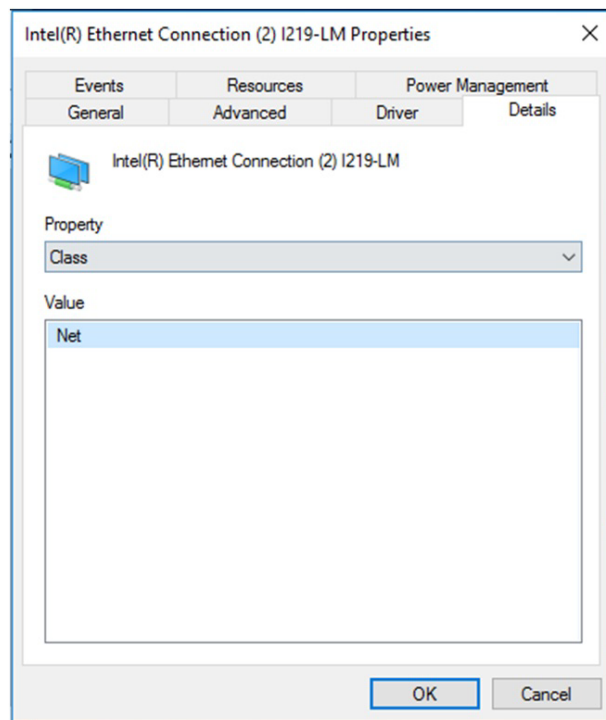
■ CMD :

```
del C:\Windows\INF\netlic64.inf  
del C:\Windows\INF\netlic64.PNF
```

4. 用 Devcon.exe 和如下操作指令将对应的网口驱动卸载。

■ CMD :

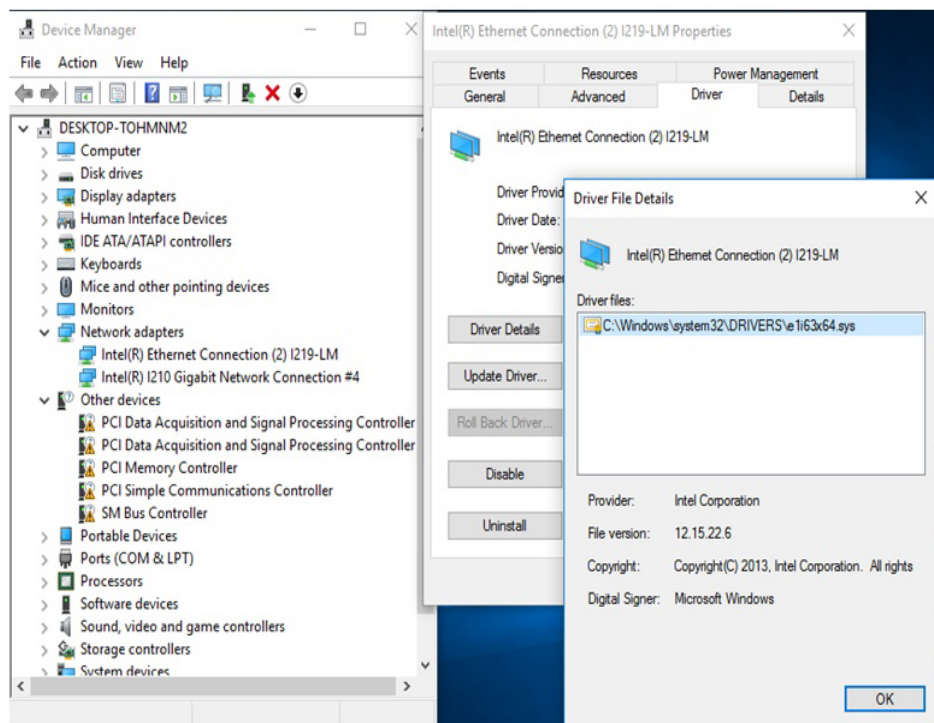
```
D:  
cdRemove_InBoxDriver  
devcon.exe /r remove =net
```



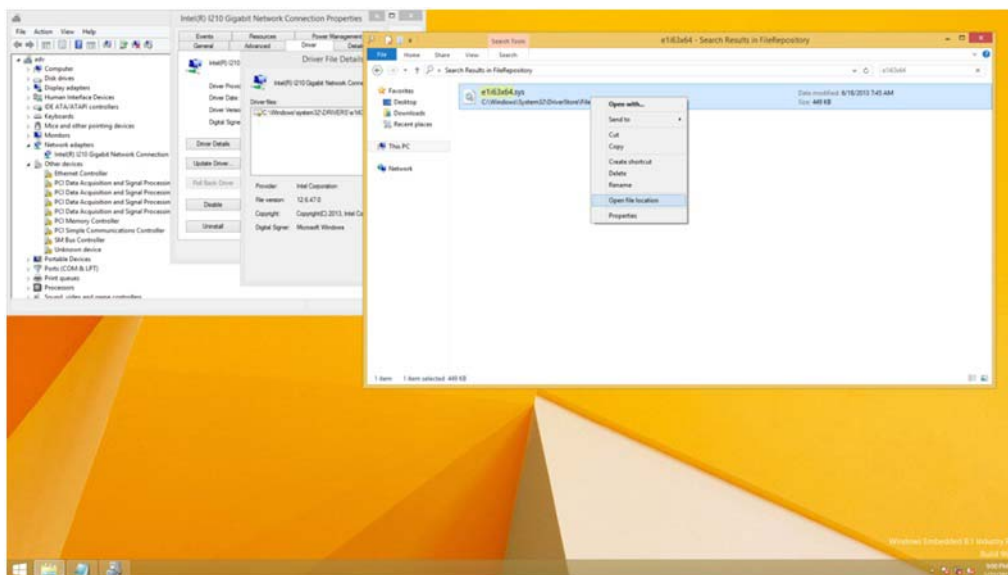
5. 用如下操作指令将对应网卡的 .sys 文件移除。

■ CMD :

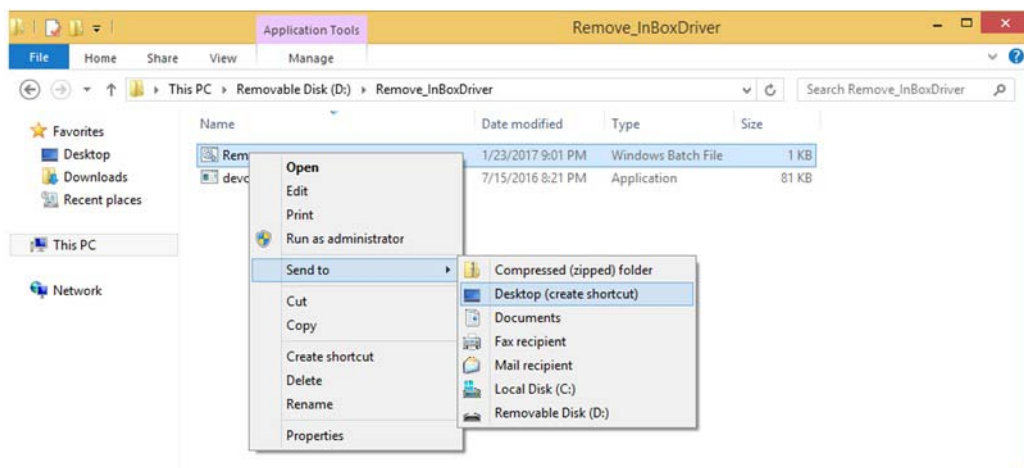
```
del C:\Windows\System32\drivers\eli63x64.sys
```

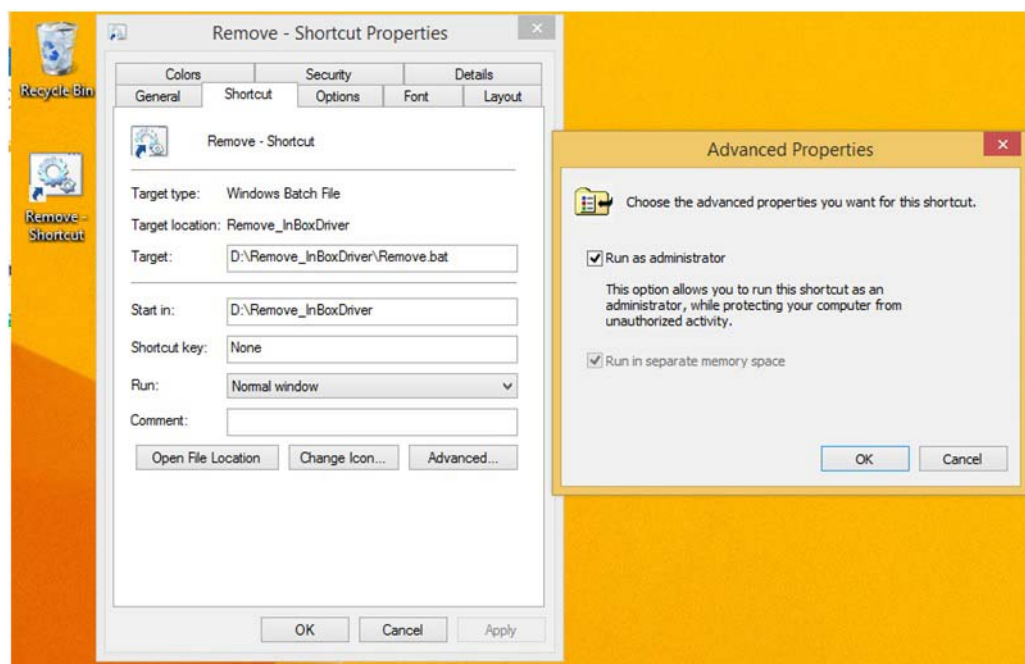


6. 照 5.3.2.5 中的操作指令找到对应路径中的文件  
C:\Windows\System32\DriverStore and search .sys file, 再按照如下操作指令将驱动存储里的驱动文件移除。  
■ CMD :  
del /  
QC:\Windows\System32\DriverStore\FileRepository\eld64x64.inf\_amd64\_16675c6772c15190



7. 完成批处理文件后, 重新生成桌面快捷方式, 最后双击快捷方式即可。





## 第 6 章

GPIO 编程指南

请仔细阅读并学习一下参考画面及蓝色粗体显示的源代码。  
请下载 NXP Semiconductors PCA9554 规格进行编程。

## 6.1 ITA-2231 数字 DIO 定义（见 2.3.6）

## 6.2 配置顺序

ITA-2231 的 GPIO 通过一个挂接在 ICH SMBUS 总线上的 PCA9554 GPIO IC 实现。  
因此对 GPIO IC 的设置和访问需要通过访问 ICH SMBUS controller 的 IO Space 来完成。  
下图是 ICH SMBUS IO Space 简图：

<b>SMB_BASE + Offset</b>	<b>Mnemonic</b>	<b>Register Name</b>	<b>Default</b>	<b>Type</b>
00h	HST_STS	Host Status	00h	R/WC, RO, R/WC (special)
02h	HST_CNT	Host Control	00h	R/W, WO
03h	HST_CMD	Host Command	00h	R/W
04h	XMIT_SLVA	Transmit Slave Address	00h	R/W
05h	HST_D0	Host Data 0	00h	R/W
06h	HST_D1	Host Data 1	00h	R/W

在 ITA-2231，上图中 **SMB\_BASE** 的 IO address 为 0xF40。  
具体的 SMBUS IO 控制访问的 code 在第 3 章节提供。

ITA-2231 上 GPIO 00 - GPIO 07 对应的 PCA9554 的 SMBUS slave address 为 0x40 (8bit address)：

**GPIO 00 - GPIO 07 : PCA9554 0x40 (I00 - I07)**

下图是 PCA9554 的示意图:

**Table 2. Pin description**

Symbol	Pin			Description
	DIP16, SO16, SSOP16, TSSOP16	HVQFN16	SSOP20	
A0	1	15	6	address input 0
A1	2	16	7	address input 1
A2	3	1	9	address input 2
IO0	4	2	10	input/output 0
IO1	5	3	11	input/output 1
IO2	6	4	12	input/output 2
IO3	7	5	14	input/output 3
V <sub>SS</sub>	8	6 <sup>[1]</sup>	15	supply ground
IO4	9	7	16	input/output 4
IO5	10	8	17	input/output 5
IO6	11	9	19	input/output 6
IO7	12	10	20	input/output 7
INT	13	11	1	interrupt output (open-drain)

PCA9554 register 图解:

### 6.1.1 Command byte

**Table 3. Command byte**

Command	Protocol	Function
0	read byte	Input Port register
1	read/write byte	Output Port register
2	read/write byte	Polarity Inversion register
3	read/write byte	Configuration register

The command byte is the first byte to follow the address byte during a write transmission. It is used as a pointer to determine which of the following registers will be written or read.

PCA9554 共有 4 个 register 来对 GPIO 做控制。

PCA9554 register 0:

6.1.2 Register 0 - Input Port register

This register is a read-only port. It reflects the incoming logic levels of the pins, regardless of whether the pin is defined as an input or an output by Register 3. Writes to this register have no effect.

The default 'X' is determined by the externally applied logic level, normally '1' when no external signal externally applied because of the internal pull-up resistors.

Table 4. Register 0 - Input Port register bit description

Bit	Symbol	Access	Value	Description
7	I7	read only	X	determined by externally applied logic level
6	I6	read only	X	
5	I5	read only	X	
4	I4	read only	X	
3	I3	read only	X	
2	I2	read only	X	
1	I1	read only	X	
0	I0	read only	X	

如果某个GPIO Pin被设置为Input，可以通过register 0的对应bit读取相应的Input值

PCA9554 register 1:

6.1.3 Register 1 - Output Port register

This register reflects the outgoing logic levels of the pins defined as outputs by Register 3. Bit values in this register have no effect on pins defined as inputs. Reads from this register return the value that is in the flip-flop controlling the output selection, **not** the actual pin value.

Table 5. Register 1 - Output Port register bit description

Legend: \* default value.

Bit	Symbol	Access	Value	Description
7	O7	R	1*	reflects outgoing logic levels of pins defined as outputs by Register 3
6	O6	R	1*	
5	O5	R	1*	
4	O4	R	1*	
3	O3	R	1*	
2	O2	R	1*	
1	O1	R	1*	
0	O0	R	1*	

如果某个GPIO Pin被设置为Output，可以通过往register 1的对应bit来输出Output值



PCA9554 register 2:

#### 6.1.4 Register 2 - Polarity Inversion register

This register allows the user to invert the polarity of the Input Port register data. If a bit in this register is set (written with '1'), the corresponding Input Port data is inverted. If a bit in this register is cleared (written with a '0'), the Input Port data polarity is retained.

**Table 6. Register 2 - Polarity Inversion register bit description**

Legend: \* default value.

Bit	Symbol	Access	Value	Description
7	N7	R/W	0*	inverts polarity of Input Port register data 0 = Input Port register data retained (default value) 1 = Input Port register data inverted
6	N6	R/W	0*	
5	N5	R/W	0*	
4	N4	R/W	0*	
3	N3	R/W	0*	
2	N2	R/W	0*	
1	N1	R/W	0*	
0	N0	R/W	0*	

如果某个 GPIO Pin 被设置成 Input，可以设置 register 2 的对应 bit 控制 Input pin 的 Polarity

PCA9554 register 3:

#### 6.1.5 Register 3 - Configuration register

This register configures the directions of the I/O pins. If a bit in this register is set, the corresponding port pin is enabled as an input with high-impedance output driver. If a bit in this register is cleared, the corresponding port pin is enabled as an output. At reset, the I/Os are configured as inputs with a weak pull-up to  $V_{DD}$ .

**Table 7. Register 3 - Configuration register bit description**

Legend: \* default value.

Bit	Symbol	Access	Value	Description
7	C7	R/W	1*	configures the directions of the I/O pins 0 = corresponding port pin enabled as an output 1 = corresponding port pin configured as input (default value)
6	C6	R/W	1*	
5	C5	R/W	1*	
4	C4	R/W	1*	
3	C3	R/W	1*	
2	C2	R/W	1*	
1	C1	R/W	1*	
0	C0	R/W	1*	

Register 3 用来设置每个 GPIO 是 Input 还是 Output:

如果对应 bit 为 '0'，则对应 GPIO pin 设置为 Output；

如果对应 bit 为 '1'，则对应 GPIO pin 设置为 Input。

示例:

以 ITA-2231 为例, 假定 GPIO 00 设置成 Output, GPIO 07 设置成 Input, 两个 pin 互联, 如何设定相应 register?

GPIO 00 对应 PCA9554 0x40 I00, 而 GPIO 07 对应 PCA9554 0x40 I07。

设置 GPIO 00 为 Output:

1. 读取 SMBUS slave 0x40 register 3 byte 值 ;
2. 将 step1 读取值 bit 0 设为 0 并写入 SMBUS slave 0x40 register 3 ;
3. 读取 SMBUS slave 0x40 register 1 byte 值;
4. 根据输出值为 Low 或者 High, 设置 step 3 读取值 bit 0 为 0 或 1 并写回 SMBUS slave 0x40 register 1。

设置 GPIO 07 为 Input 值 :

1. 读取 SMBUS slave 0x40 register 3 byte 值 ;
2. 将 step1 读取值 bit 7 设为 1 并写入 SMBUS slave 0x40 register 3 ;
3. 读取 SMBUS slave 0x40 register 0 byte 值;
4. 通过 step 3 bit7 的值判断 Input 值为 Low 还是 High。

## 6.3 Function call for reference

ICH SMBUS 访问代码

(以下代码是模拟 BIOS 对 SMBUS 的访问而实现的, 使用 Borand C++ 3.1 编译通过, 并在 DOS 下验证通过, 未在其他 OS 下验证)

```
#define SMBUS_PORT 0xF040//SMB_BASE 为 0xF040
```

```
typedef unsigned char BYTE;
```

```
////////////////////////////////////  
////////////////////////////////////
```

```
BYTE    smbus_read_byte(BYTE addr, BYTE offset)
```

```
// 读取 SMBUS Register byte 值, 一次读回一个 byte 值, 其中 addr 为 slave address  
如 0x40; offset 为 register offset
```

```
{
```

```
    int    i;
```

```
    BYTE data;
```

```
    outportb(SMBUS_PORT + 4, (addr | 1)); // 往 SMB_BASE + 4 写入 slave  
address (读取的时候需要设置 slave address bit 0 为 1, 所以这里有 addr|1)
```

```
    newiodelay(); // 延时
```

```
    newiodelay(); // 延时
```

```
    chk_smbus_ready(); // 判断 SMBUS 总线是否 ready
```

```
    outportb(SMBUS_PORT + 3, offset); // 往 SMB_BASE + 3 写入 register  
offset
```

```
    newiodelay(); // 延时
```

```
    newiodelay(); // 延时
```

```

        outportb(SMBUS_PORT + 2, 0x48); // 往 SMB_BASE + 2 写入 SMBUS 命令, 0x48
的意思是开始 Byte data 数据传输
        newiodelay(); // 延时
        newiodelay(); // 延时

        for (i = 0; i <= 0x100; i++)
        {
            newiodelay(); // 较长延时
        }

        chk_smbus_ready(); // 判断 SMBUS 总线是否 ready
        return(inportb(SMBUS_PORT + 5)); // 从 SMB_BASE + 5 读取读出的 byte 值
    }

    ///////////////////////////////////////////////////////////////////
    ///////////////////////////////////////////////////////////////////
void    smbus_write_byte(BYTE addr, BYTE offset, BYTE value)
// 写 SMBUS Register byte 值, 一次写一个 byte 值, 其中 addr 为 slave address 如
0x40; offset 为 register offset; value 为要写入的值
{
    int    i;

        outportb(SMBUS_PORT + 4, addr); // 往 SMB_BASE + 4 写入 slave address (
写入的时候需要设置 slave address bit 0 为 0)
        moredelay(); // 较长延时
        moredelay(); // 较长延时

        chk_smbus_ready(); // 判断 SMBUS 总线是否 ready

        outportb(SMBUS_PORT + 3, offset); // 往 SMB_BASE + 3 写入 register
offset
        moredelay(); // 较长延时
        moredelay(); // 较长延时

        outportb(SMBUS_PORT + 5, value); // 往 SMB_BASE + 5 写入数据 value
        moredelay(); // 较长延时
        moredelay(); // 较长延时

        outportb(SMBUS_PORT + 2, 0x48); // 往 SMB_BASE + 2 写入 SMBUS 命令, 0x48
的意思是开始 Byte data 数据传输
        moredelay(); // 较长延时
        moredelay(); // 较长延时

        for (i = 0; i <= 0x100; i++)
        {
            newiodelay(); // 较长延时

```

```

    }

    chk_smbus_ready(); // 判断 SMBUS 总线是否 ready
}

////////////////////////////////////
////////////////////////////////////
int    chk_smbus_ready()
// 判断 SMBUS 总线是否 ready 或者完成动作，基本上是做很长的等待，看 SMBUS 是否
// 传输完命令，因为出错情况基本不会出现，故 BIOS Code 在 SMBUS byte read 和 write
// 中并没有对本函数返回值做相应的判断
{
    int i, result = 1;
    BYTE data;

    for (i = 0; i <= 0x800; i++)
    {
        //SMB_BASE + 0 为 SMBUS status 值
        data = inportb(SMBUS_PORT); // 做一次读取 SMBUS status 值
        data = check_data(SMBUS_PORT); // 做多次读取 SMBUS status 值
        outportb(SMBUS_PORT, data); // 写回 SMBUS status 值，也就是清除
        status 值（在对应 bit 写入 1 代表清除 status）

        if (data & 0x02)
        {
            // 如果 bit 1 置位（代表命令完成），则 SMBUS 已经 ready
            result = 0; // SMBUS ready
            break;
        }

        if (!(data & 0xBF))
        {
            // 如果除 bit 2（代表 SMBUS 错误）以外的 bit 都为 0，则 SMBUS
            已经 ready
            result = 0; // SMBUS ready
            break;
        }

        if (data & 0x04)
        {
            // 如果 bit 2 置位（代表 SMBUS 错误），则 SMBUS 已经出现错误，
            这种情况很少出现
            result = 1; // SMBUS error
            break;
        }
    }

    return result;
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
BYTE    check_data(WORD addr)
{
    int i;
    BYTE data;

    for(i = 0; i <= 6; i++)
    {
        data = inportb(addr);
        if (data != 0)
            break;
    }

    returndata;
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
void    newiodelay()
// 短延时
{
    outportb(0xeb, 0); //IO port 0xeb 没有实际设备占用，通过往此 port 写值
    做延时功能，使用者可根据实际情况用其他方法代替
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
void    moredelay()
// 较长延时
{
    int i;
    for (i = 0; i < 20; i++)
    {
        outportb(0xeb, 0); //IO port 0xeb 没有实际设备占用，通过往此 port
        写值做延时功能，使用者可根据实际情况用其他方法代替
    }
}

```

\*\*\*\*\*

#### GPIO 模拟代码

(以第二章的 GPIO 00 和 GPIO 07 为例)

往 GPIO 00 输出 High:

```

data = smbus_read_byte(0x40, 0x03); // 读取 slave 0x40 register 3 byte
data &= 0xfe; // bit 0 设为 0

```

---

```
smbus_write_byte(0x40, 0x03, data)// 写回, GPIO 00 设为了 Output  
data = smbus_read_byte(0x40, 0x01)// 读取 slave 0x40 register 1  
data |= 0x01;//bit 0 设为 1, 代表 High  
smbus_write_byte(0x40, 0x01, data)// 写回, 输出 High 值
```

从 GPIO 07 读取 Input 值:

```
data = smbus_read_byte(0x40, 0x03);//读取 slave 0x40 register 3 byte  
data |= 0x80;//bit 7 设为 1  
smbus_write_byte(0x40, 0x03, data)// 写回, GPIO 07 设为了 Input  
data = smbus_read_byte(0x40, 0x00)// 读取 slave 0x40 register 0, 此时  
bit 7 的响应值就能得出 Input 是 Low 还是 High
```

## 附录 A

### 看门狗定时器编程

# A. 1 看门狗定时器编程

ITA-2231 看门狗定时器可用于监控系统软件操作，并在编程过程中出现软件故障时采取适当措施。本章节介绍了看门狗定时器操作以及如何编程。

## A. 1. 1 看门狗定时器概述

看门狗定时器内置于高级 I/O 控制器 SMSC SCH3114，提供了以下用户可编程功能：可通过用户编程启用或禁用。

定时器可设置为 1 ~ 255 秒或 1 ~ 255 分钟。

在软件复位定时器超时时产生中断或复位信号。

## A. 1. 2 编程看门狗定时器

看门狗定时器的 I/O 端口地址为 680h （hex）。

表 A. 1： 看门狗定时器寄存器		
地址：680 (hex)		
寄存器偏移	读 / 写	说明
65 (hex)	读 / 写	将定时器计时单位设置为秒或分钟： 向 bit 7 写入 0：将计时单位设置为分钟。（默认） 向 bit 7 写入 1：将计时单位设置为秒。
66 (hex)	读 / 写	0：停止定时器（默认）。 01~FF（hex）：计时值，单位为秒或分钟，取决于寄存器 65（hex）的设置值。 该值说明了看门狗定时器等待选通的时间达到何值时产生中断或复位信号。向该寄存器写入新值可以复位定时器，使其以新值开始计时。
67 (hex)	读 / 写	配置看门狗定时器 Bit 1：写入 1 启用键盘复位定时器，写入 0 禁用（默认）。 Bit 2 写入 1 启用鼠标复位定时器，写入 0 禁用（默认）。 Bit 7~4：设置看门狗定时器中断映射： 1111=IRQ15 ..... 0011=IRQ3 0010=IRQ2 0001=IRQ1 0000=Disable（默认）
68 (hex)	读 / 写	控制看门狗定时器 Bit 0：读取看门狗状态，1= 定时器超时。 Bit 2：写入 1 立即产生超时信号，并自动返回到 0（只写）。 Bit 3：写入 1 允许 P20 有效时触发定时器超时，写入 0 禁用（默认）。



### A. 1.3 编程示例

```

;-----
1.  启用看门狗定时器，并将超时间隔设置为 10 秒。
;-----
Mov dx, 6E5h ; 选择寄存器 65h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 65h
Mov al, 80h  ; 将计时单位设置为秒
Out dx, al

Mov dx, 6E6h ; 选择寄存器 66h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 66h
Mov al, 10   ; 将超时间隔设置为 10 秒然后开始计时
Out dx, al

;-----
2.  启用看门狗定时器功能并将超时间隔设置为 5 分钟。
;-----
Mov dx, 6E5h ; 选择寄存器 65h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 65h
Mov al, 00h  ; 将计时单位设置为分钟
Out dx, al

Mov dx, 6E6h ; 选择寄存器 66h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 66h
Mov al, 5    ; 将超时间隔设置为 5 分钟然后开始计时
Out dx, al

;-----
3.  启用鼠标复位看门狗定时器功能。
;-----
Mov dx, 6E7h ; 选择寄存器 67h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 67h
In  al, dx
Or  al, 4h   ; 启用鼠标复位看门狗定时器
Out dx, al

;-----
4.  启用键盘复位看门狗定时器功能。
;-----
Mov dx, 6E7h ; 选择寄存器 67h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 67h
In  al, dx
Or  al, 2h   ; 启用键盘复位看门狗定时器
Out dx, al

;-----
5.  定时器不计时时，产生超时信号
;-----
Mov dx, 6E8h ; 选择寄存器 68h，看门狗定时器的 I/O 端口地址 680h+ 寄存器偏移 68h
In  al, dx
Or  al, 4h   ; 立即产生超时信号
Out dx, al

```



*Enabling an Intelligent Planet*

[www.advantech.com.cn](http://www.advantech.com.cn)

使用前请检查核实产品的规格。本手册仅作为参考。

产品规格如有变更，恕不另行通知。

未经研华公司书面许可，本手册中的所有内容不得通过任何途径以任何形式复制、翻印、翻译或者传输。

所有的产品品牌或产品型号均为公司之注册商标。

© 研华公司 2017