



# 产品说明书

**GM0-5602**

版本：R01

## 法律资讯

### 警告提示

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

#### 危险

表示如果不采取相应的小心措施，将会导致死亡或者严重的人身伤害。

#### 警告

表示如果不采取相应的小心措施，可能导致死亡或者严重的人身伤害。

#### 小心

带有警告三角，表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

#### 注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

### 合格的专业人员

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

### GITSTAR产品

请注意下列说明：

#### 警告

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

## 免责声明

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

订购产品前，请向经销商详细了解产品性能是否符合您的需求。

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

## 保修条款：

产品保修条款详见官网说明。用户如另有要求，以双方签署的合同为准。

## 欲获更多信息请访问：

集特网站：<http://www.graest.com>

## 文档说明

### 本文档适用范围

本文档适用于**GITSTAR GM0-5602** 型号。

### 约定

在本文档中，术语“本板”或“产品”有时特指 **GITSTAR GM0-5602** 产品。

### 说明

#### 安全相关注意事项

为避免财产损失以及出于个人安全方面的原因，请注意本入门指南中关于安全方面的信息。文中使用警告三角来指示这些安全信息，警告三角的出现取决于潜在危险的程度。

### 历史

本说明书发布版本：

版本	时间
R01	2025.4

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，本公司保留对此手册更改的权利，且不另行通知。对于任何因安装、使用不当而导致的直接、间接、有意或无意的损坏及隐患概不负责。

订购产品前，请向经销商详细了解产品性能是否符合您的需求。本手册所涉及到的其他商标，其所有权为相应的产品厂家所拥有。

本手册内容受版权保护，版权所有。未经许可，不得以机械的、电子的或其它任何方式进行复制。

# 温馨提示

- 1、产品使用前，务必请仔细阅读产品说明书。
- 2、对未准备安装的主板，应将其保存在防静电保护袋中。
- 3、在从包装袋中拿主板前，应将手先置于接地金属物体上一会儿，以释放身体及手中的静电
- 4、在使用前，宜将主板置于稳固的平面上。
- 5、请保持主板的干燥，散热片的开口缝槽是用于通风，避免机箱内的部件过热。请勿将此类开口掩盖或堵塞。
- 6、在将主板与电源连接前，请确认电源电压值。
- 7、请将电源线置于不会被践踏的地方，且不要在电源线上堆置任何物件。
- 8、当您需连接或拔除任何设备前，须确定所有的电源线事先已被拔掉。
- 9、为避免人体被电击或产品被损坏，在每次对整机、板卡进行拔插或重新配置时，须先关闭交流电源或将交流电源线从电源插座中拔掉。
- 10、请留意手册上提到的所有注意和警告事项。
- 11、为避免频繁开关机对产品造成不必要的损伤，关机后，应至少等待30秒后再开机。
- 12、设备在使用过程中出现异常情况，请找专业人员处理。
- 13、请不要将本设备置于或保存在环境温度高于70°C上，否则会对设备造成伤害。



# 1 产品介绍

## 1.1 产品概述

GM0-5602 主板是采用国产海光处理器 C86 HG3350 的工业 ATX 主板，支持四条 DDR4 内存，最大可达到 128G，支持 VGA/HDMI/LVDS 任意双显，支持 PCIe/PCI 插槽，及 COM/GPIO/USB 等多种扩展端口。

GM0-5602 主板具备了国产安全自主可控产品的要求，产品性能稳定可靠，可兼容市面上主流的工控机箱的装配，可广泛应用于智能制造、工业控制、机器视觉、智能交通、机械自动化、安防监控、教育培训、通讯及信息采集等场景。



图 1-1 主板外观图

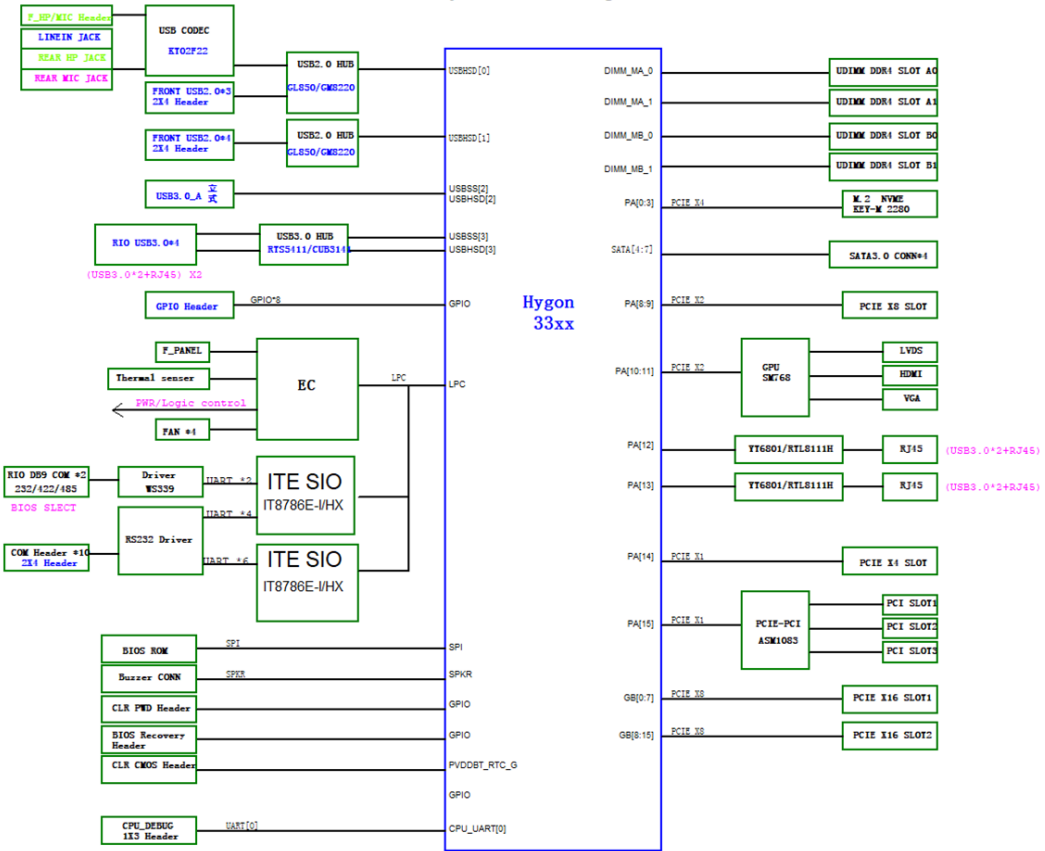
## 1.2 产品规格

名称		规格
CPU		8核 16线程 3.0GHz, 典型功耗
内存		4个 DDR4 插槽支持 UDIMM/ECC RDIMM 内存,最大可达 256G
网络		板载 2个 PCIE 千兆网卡 (RJ45)
显示		后 IO 板载 1个 VGA 接口, 支持最大分辨率 1920*1080
		后 IO 板载 1个 HDMI 接口, 支持最大分辨率 3840*2160
		板载 1个 LVDS 显示接口插座
		支持双独立显示
PCIe/PCI 扩展		2个 PCIe3.0 X16 插槽 (分别为 X8 信号)
		1个 PCIe3.0 X8 插槽 (分别为 X2 信号)
		1个 PCIe3.0 X4 插槽 (分别为 X1 信号)
		3个 32bit PCI 插槽
存储接口		板载 4个 SATA3.0 端口
		板载 1个 M.2 NVME 2280 连接器 (PCIe X2 信号)
IO 接口	外漏 IO	4个 USB3.0 端口
		2个 DB9 串口, 可支持 RS232/422/485 可调
		板载一组 Audio 接口 (MIC,Linein,Lineout)
	内部 IO	1个 USB3.0 Type A 位于主板内部
		7个 USB2.0 插座
		10个 COM 插座, 支持 RS232
		8个 GPIO 插座
		1个 HDaudio 插座
操作系统		麒麟、统信、方德等国产操作系统
		Windows7/10/11、Windows Server2019 等视窗操作系统
		Ubuntu、CentOS、RedHat 等商用 Linux 操作系统
看门狗计时器		System Reset
		1~255 秒可编程
电源		ATX 24+4pin 电源输入
散热		4个 4pin 2510 插座, 支持 4个 PWM 调速风扇
尺寸		ATX 305*244mm
温度	工作温度	-10°C~60°C
	储存温度	-40°C~85°C
湿度		5%~93% @40°C, 无凝结
震动 (5~500Hz)		工作: 1GrmsGrms, 非工作: 2Grms



### 1.3 主板框图

System block diagram



## 2 产品结构及安装说明

### 2.1 产品结构

#### 2.1.2 产品包装

GM0-5602主板采用飞机盒独立包装，出厂主板已锁紧CPU散热器，出厂包装内包含以下物料（COM线成品线材实物图）

	物料条码	规格参数型号	数量
标配	622-001-080104	主板 GM0-5602	1
标配	800-908-010301	太阳花CPU散热器 GCC103-01/出厂锁紧螺钉	1
标配	622-999-080104	GM0-5602-IO挡板	1
标配	800-996-110201	GITSTAR合格证	1
选配		线材辅料包（COM线*5/SATA3.0线）	1

#### 2.1.2 产品结构图示

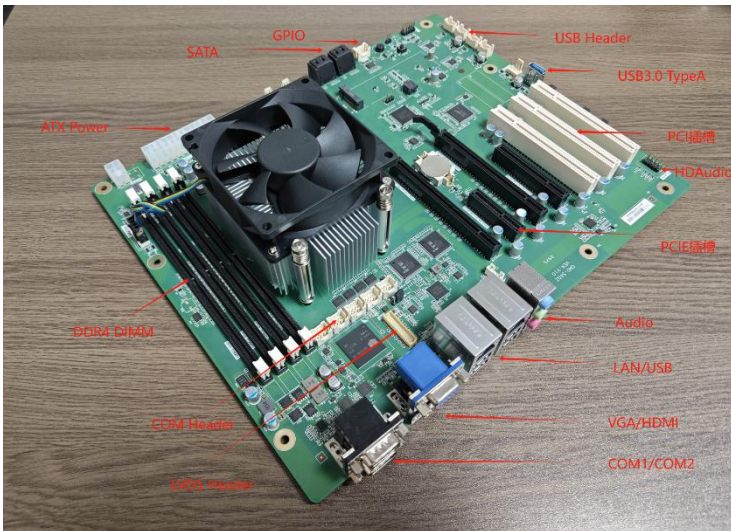


图 2-1 主板 I/O 图





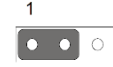
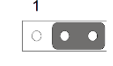
## 2.2 产品跳线及接口说明

### 2.2.1 COMS/PWD/BIOS\_Recovery 跳线设置(2.54mm 排针 3Pin)



图 2-2 主板跳线接口位置图

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

COMS_CLR	设置	功能
	1-2 短路	正常工作状态 (Default)
	2-3 短路	清除 CMOS, 所有 BIOS 设置恢复成出厂值。
PWD_CLR	设置	功能
	1-2 短路	正常工作状态 (Default)
	2-3 短路	Clear Password
BIOS_Recovery	设置	功能
	1-2 短路	正常工作状态 (Default)
	2-3 短路	BIOS Recovery

## 2.2.2 风扇接口 (2510 插座)

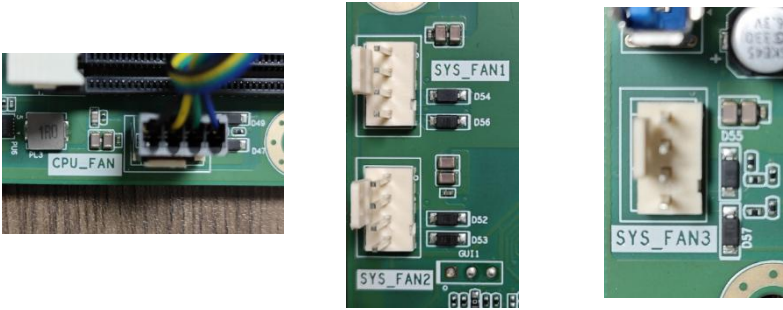


图 2-3 主板散热风扇位置图

	管脚	信号名称	功能
	1	GND	
	2	+12V	
CPU_FAN/SYS1~3FAN	3	FAN_IO	风扇转速输出
2510 插座限位第 3P	4	FAN_PWM	风扇转速控制

## 2.2.3 串口 (DB9\*2)

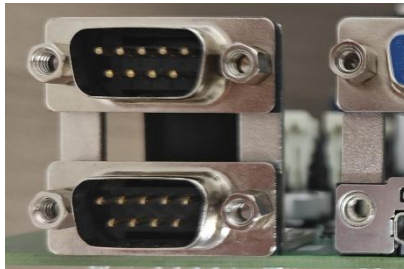
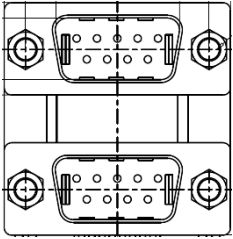


图 2-4 主板 COM1/2 位置图

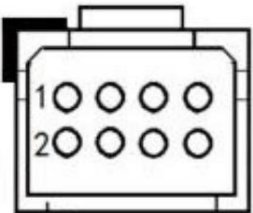
开机进BIOS选择串口的方式，支持RS232/422/485

	管脚	信号名称		
		RS232	RS422	RS485
	1	DCD#	TX-	Data-
	2	RXD	TX+	Data+
	3	TXD	RX+	
	4	DTR#	RX-	
	5	GND	GND	GND
	6	DSR#		
	7	RTS#		
<b>COM2(上)</b>	8	CTS#		
<b>COM1(下)</b>	9	RI#		

### 2.2.3 扩展 COM 插针 (PHB2.0 2\*4P)



图 2-5 主板 COM3~12 位置图

	Pin 脚	信号名称
		
	2	UART_RXD_1
	3	UART_TXD_2
	4	UART_RXD_2
	5	NC
	6	NC
	7	GND
<b>COM3~COM12</b>	8	GND

## 2.2.4 扩展 GPIO 插针 (PHB2.0 2\*5P)

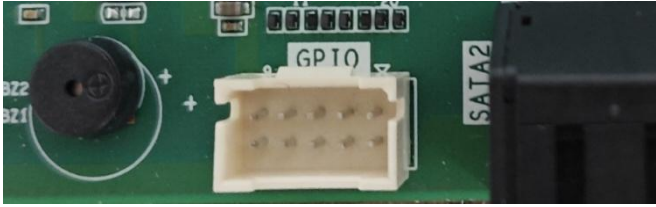
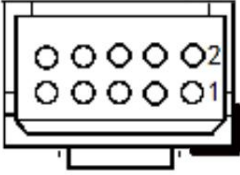


图 2-5 主板 GPIO 位置图

	Pin 脚	信号名称
	1	GPIO1
	2	GPIO2
	3	GPIO3
	4	GPIO4
	5	GPIO5
	6	GPIO6
	7	GPIO7
	8	GPIO8
	9	VCC-5V
<b>GPIO</b>	10	GND

## 2.2.5 扩展 USB2.0 插针 (PHB2.0 2\*5Pin/杜邦 2.54 2\*5Pin/PH2.0 4Pin)

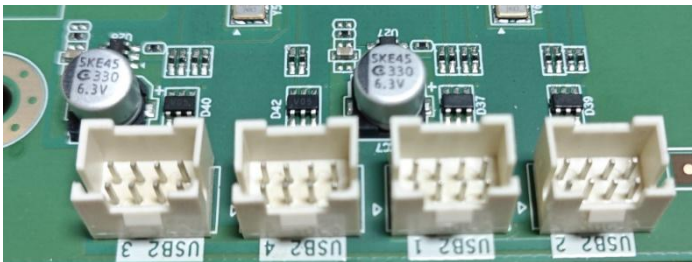
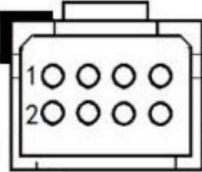
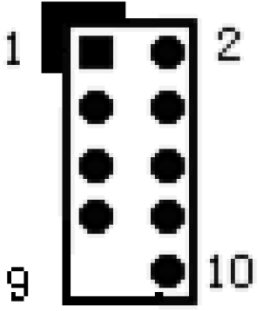


图 2-5 主板内置 USB2.0 插座位置图

为了兼容其他厂家的机箱前面板，集特主板特意为客户设计了三款USB2.0插座

	Pin 脚	信号名称
	1	USB2.0_5V
	2	USB2.0_5V
	3	USB2.0_DN_2
	4	USB2.0_DN_1
	5	USB2.0_DP_2
6	USB2.0_DP_1	
<b>USB2_1,USB2_2</b>	7	GND
<b>PHB2.0 2*4Pin</b>	8	GND

	Pin 脚	信号名称
	1	USB2.0_5V
	2	USB2.0_5V
	3	USB2.0_DN_2
	4	USB2.0_DN_1
	5	USB2.0_DP_2
	6	USB2.0_DP_1
	7	GND
8	GND	
<b>USB2_3</b>	9	NA 空/限位/防反插
<b>杜邦 2.54 2*5Pin</b>	10	GND

	Pin 脚	信号名称
	1	USB2.0_5V
	2	USB2.0_DN
	3	USB2.0_DP
<b>USB2_4</b>	4	GND
<b>PH2.0 4Pin</b>		

## 2.2.6 扩展 USB3.0 接口 (Type-A 标准接口)

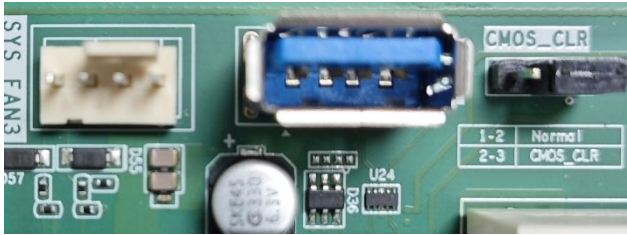
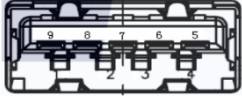


图 2-5 主板内置 USB3.0 插座位置图

 <b>USB1</b> <b>Type-A</b>	Pin 脚	信号名称
	1	USB2.0_5V
	2	USB2.0_5V
	3	USB2.0_DN_2
	4	USB2.0_DN_1
	5	USB2.0_DP_2
	6	USB2.0_DP_1
	7	GND
	8	GND
9		

### 2.2.7 音频接口 (Type-A 标准接口)



图 2-6 主板音频位置图



	Pin 脚	信号名称
	1	LINE_IN
	2	LINE_OUT
3	MIC_IN	

### 2.2.8 扩展音频插针 (杜邦 2.54 2\*5P/N8P)

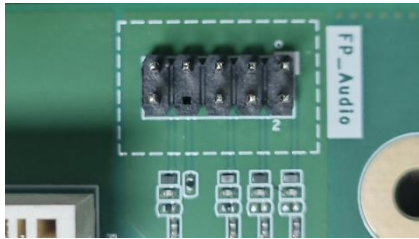
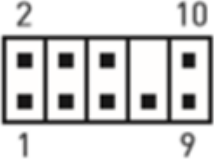


图 2-7 主板内置音频位置图

	Pin 脚	信号名称
	1	MIC_L
	2	GND
	3	MIC_R
	4	DET
	5	HP_R
	6	MIC-JD
	7	FRONT JD
8	NA	
<b>F_AUDIO1</b>	9	HP_L
<b>杜邦 2.54 2*5P/N8P</b>	10	HP-JD

### 2.2.9 网络接口 (RJ45 标准接口)



图 2-8 主板网口位置图

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

### 2.2.10 前面板接口 (PHB2.0 2\*5Pin/杜邦 2.54 2\*5Pin N10P)

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

管脚		信号名称
	1	FP_HDD_LED_P
	2	FP_PWR_LED_P
	3	FP_HDD_LED_N
	4	FP_PWR_LED_N
	5	GND
	6	FP_PWR_BNT_N
	7	FP_RST_BTN_N
	8	GND
<b>FP1(杜邦 2.54 2*5P)</b>	9	NC
<b>FP2(PHB2.0 2*5P)</b>	10	FP1 空/限位, FP2 NC

### 2.2.11 LVDS 背光接口 (PH2.0 4Pin)

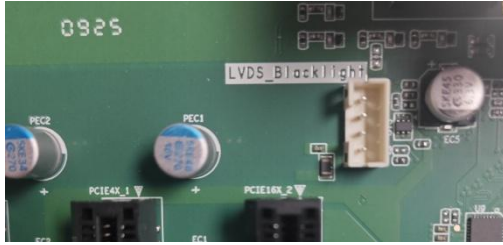
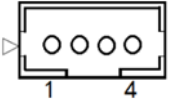




图 2-10 主板 LVDS 背光位置图

	管脚	信号名称
	1	12V
	2	PANEL_BLK_PWM
LCD_BLK1	3	PANEL_BLK_EN_C
PHB2.0 4P	4	GND

### 2.2.12 LVDS 背光电压跳线接口 (杜邦 2.54 3Pin)



图 2-11 主板 LVDS 背光位置图

LVDS_PWR1	设置	功能
	1-2 短路	5V (Default)
	2-3 短路	3.3V

## 2.2.13 LVDS 插针接口 (2\*20P 1.0 间距插座 创端 W100V40G2S)



图 2-12 主板 LVDS 位置图

		Pin 脚	功能	Pin 脚	
1		2	GND	2	GND
3	1	4	TX3_P	4	TX7_P
5	3	6	TX3_N	6	TX7_N
7	5	8	GND	8	GND
9	7	10	CLK1_P	10	CLK2_P
11	9	12	CLK1_N	12	CLK2_N
13	11	14	GND	14	GND
15	13	16	TX2_P	16	TX6_P
17	15	18	TX2_N	18	TX6_N
19	17	20	GND	20	GND
21	19	22	TX1_P	22	TX5_P
23	21	24	TX1_N	24	TX5_N
25	23	26	GND`	26	GND
27	25	28	TX0_P	28	TX4_P
29	27	30	TX0_N	30	TX4_N
31	29	32	GND	32	GND
33	31	34	NC	34	NC
35	33	36	5V	36	5V
37	35	38	5V	38	5V
39	37	40	5V	40	5V
<b>LVDS</b>					

## 2.2.14 固件升级接口 (杜邦 2.54 2\*4Pin N7P)

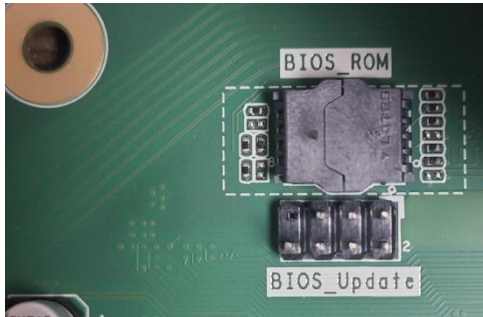
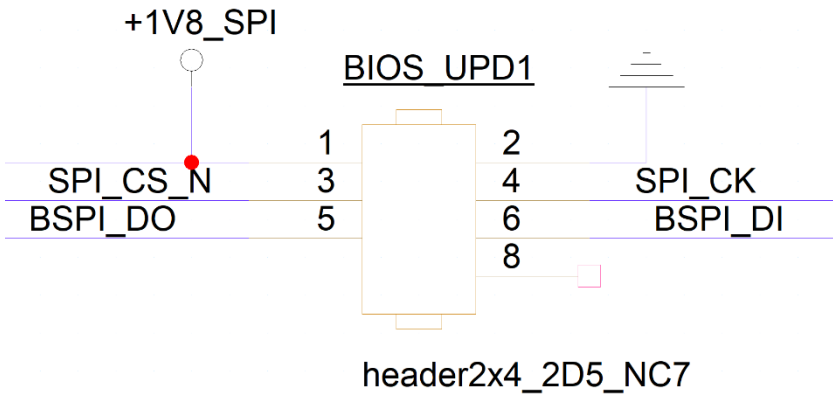


图 2-12 主板固件升级座位置图



★：非专业维修人员请勿使用

## 2.2.10 I/O 面板组件

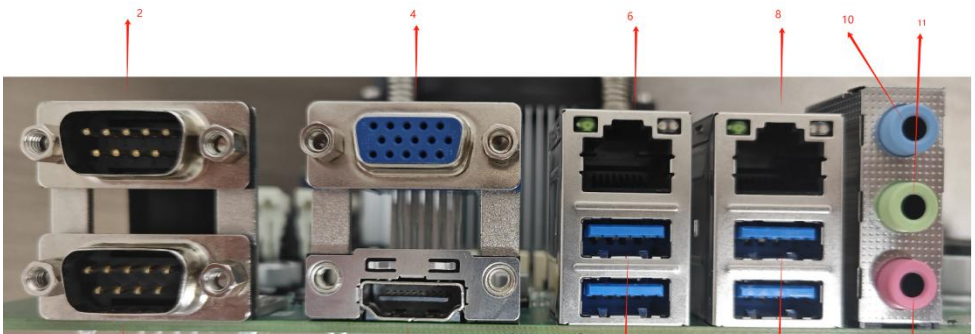


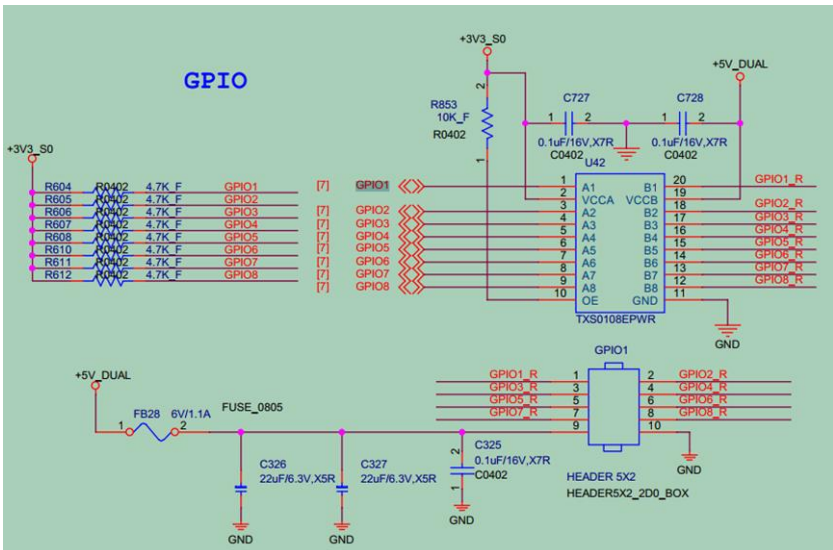
图 2-13 I/O 面板组件

编号	说明
1	COM1
2	COM2
3	HDMI 显示接口
4	VGA 显示接口
5	USB3.0 接口
6	千兆网络接口
7	USB3.0 接口
8	千兆网络接口
9	麦克风接口
10	音频输出接口
11	音频输入接口

## 3 GPIO 编程指导

### 3.1 GPIO 控制

控制 CPU 直出的 EGPIO96、EGPIO97、EGPIO98、EGPIO99、EGPIO100、AGPIO90、AGPIO89、AGPIO84，对应主板上 HEADER GPIO1~8。



## 3.2 GPIO 编程指导文档说明

本文档包含 GM0-5602 主板的 GPIO 二次开发说明和软件样例伪代码。

### 3.2.1 文档说明

CPU GPIO	HEADER GPIO
EGPI096	GPIO1
EGPI097	GPIO2
EGPI098	GPIO3
EGPI099	GPIO4
EGPI100	GPIO5
EGPI090	GPIO6
EGPI089	GPIO7
EGPI084	GPIO8

#### GPIOx00100 [GPIO Bank 1 Control Register] (GPIOBank1Ctl)

Reset: X0XX_XXXXh.	
Each GPIO pin is controlled by 4 bytes. These registers control GPIO bank 1 pins: GPIO64-GPIO127.	
FCH::GPIO::GPIOBank1Ctl_link[63:0]_aliasHOST; GPIOx001[[F:0][C.8,4,0]]; GPIO=FED8_1500h	
Bits	Description
31: 30	Reserved.
29	<b>WakeSts.</b> Read,Write-1-to-clear. Reset: Xb. 0=The pin didn't generate wake event. 1=The pin is a wake source. WakeSts Register.
28	<b>InterruptSts.</b> Read,Write-1-to-clear. Reset: Xb. 0=The pin didn't generate interrupt. 1=The pin is a interrupt source. InterruptSts Register.
27: 24	Reserved.
23	<b>OutputEnable.</b> Read-write. Reset: Xb. 1=Output is enabled on the pin. 0=Output is disabled on the pin. Note:I2C buffers are OD only and cannot support push-pull regardless if they are set in GPIO mode or I2C mode.
22	<b>OutputValue.</b> Read-write. Reset: Xb. 1=high. 0=low. When select I2C pad, OutputValue=don't care. Note:I2C buffers are OD only and cannot support push-pull regardless if they are set in GPIO mode or I2C mode.
21	<b>PullDownEnable.</b> Read-write. Reset: Xb. 1=Pull-down is enabled on the pin. 0=Pull-down is disabled on the pin. PullDownEnable is not applicable for I2C pad.

### 3.2.2 样例伪代码

样例伪代码为同时使 8 个 GPIO 输出高。具体函数定义如下：

```

#define IOMUX_BASE      0xFED80D00
#define IOMUX_GPIO1    0x60
#define IOMUX_GPIO2    0x61
#define IOMUX_GPIO3    0x62
#define IOMUX_GPIO4    0x63
#define IOMUX_GPIO5    0x64
#define IOMUX_GPIO6    0x5a
#define IOMUX_GPIO7    0x59
#define IOMUX_GPIO8    0x54

```

```

#define GPIO_BASE      0xFED81500
#define GPIO1_REG      0x0180
#define GPIO2_REG      0x0184
#define GPIO3_REG      0x0188
#define GPIO4_REG      0x018C
#define GPIO5_REG      0x0190
#define GPIO6_REG      0x0168
#define GPIO7_REG      0x0164
#define GPIO8_REG      0x0150

```

UINT8

EFIAPI

MmioWrite8 (

```

    IN      UINTN          Address,
    IN      UINT8         Value
)

```

```

{
    *(volatile UINT8*)Address = Value;
    return Value;
}

```

UINT32

EFIAPI



```

MmioRead32 (
    IN     UINTN           Address
)
{
    UINT32           Value;

    ASSERT ((Address & 3) == 0);
    Value = *(volatile UINT32*)Address;
    return Value;
}

```

UINT32

EFIAPI

```

MmioWrite32 (
    IN     UINTN           Address,
    IN     UINT32          Value
)
{
    ASSERT ((Address & 3) == 0);
    *(volatile UINT32*)Address = Value;
    return Value;
}

```

VOID

```

main(
    IN EFI_HANDLE ImageHandle,
    IN EFI_SYSTEM_TABLE *SystemTable,
    IN UINTN Argc,
    IN UINT16 **Argv )
{
    UINT32           Value;
    EFI_STATUS       Status = EFI_SUCCESS;

    //config GPIO1 (EGPIO96)

```

```

MmioWrite8(IOMUX_BASE + IOMUX_GPIO1, 0x01); //iomux as gpio
Value = MmioRead32(GPIO_BASE + GPIO1_REG);
MmioWrite32(GPIO_BASE + GPIO1_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
Value = MmioRead32(GPIO_BASE + GPIO1_REG);
Print(L"%a line=%d ,0xFED81680=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO2 (EGPIO97)
MmioWrite8(IOMUX_BASE + IOMUX_GPIO2, 0x01); //iomux as gpio
Value = MmioRead32(GPIO_BASE + GPIO2_REG);
MmioWrite32(GPIO_BASE + GPIO2_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
Value = MmioRead32(GPIO_BASE + GPIO2_REG);
Print(L"%a line=%d ,0xfed81684=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO3 (EGPIO98)
MmioWrite8(IOMUX_BASE + IOMUX_GPIO3, 0x01); //iomux as gpio
Value = MmioRead32(GPIO_BASE + GPIO3_REG);
MmioWrite32(GPIO_BASE + GPIO3_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
Value = MmioRead32(GPIO_BASE + GPIO3_REG);
Print(L"%a line=%d ,0xfed81688=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO4 (EGPIO99)
MmioWrite8(IOMUX_BASE + IOMUX_GPIO4, 0x01); //iomux as gpio
Value = MmioRead32(GPIO_BASE + GPIO4_REG);
MmioWrite32(GPIO_BASE + GPIO4_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
Value = MmioRead32(GPIO_BASE + GPIO4_REG);
Print(L"%a line=%d ,0xfed8168c=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO5 (EGPIO100)
MmioWrite8(IOMUX_BASE + IOMUX_GPIO5, 0x01); //iomux as gpio
Value = MmioRead32(GPIO_BASE + GPIO5_REG);

```

```

        MmioWrite32(GPIO_BASE + GPIO5_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
        Value = MmioRead32(GPIO_BASE + GPIO5_REG);
        Print(L"%a line=%d ,0xfed81690=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO6 (EGPIO90)
        MmioWrite8(IOMUX_BASE + IOMUX_GPIO6, 0x02); //iomux as gpio
        Value = MmioRead32(GPIO_BASE + GPIO6_REG);
        MmioWrite32(GPIO_BASE + GPIO6_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
        Value = MmioRead32(GPIO_BASE + GPIO6_REG);
        Print(L"%a line=%d ,0xfed81668=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO7 (EGPIO89)
        MmioWrite8(IOMUX_BASE + IOMUX_GPIO7, 0x02); //iomux as gpio
        Value = MmioRead32(GPIO_BASE + GPIO7_REG);
        MmioWrite32(GPIO_BASE + GPIO7_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
        Value = MmioRead32(GPIO_BASE + GPIO7_REG);
        Print(L"%a line=%d ,0xfed81664=%x\n", __FUNCTION__, __LINE__, Value);

//config GPIO8 (EGPIO84)
        MmioWrite8(IOMUX_BASE + IOMUX_GPIO8, 0x01); //iomux as gpio
        Value = MmioRead32(GPIO_BASE + GPIO8_REG);
        MmioWrite32(GPIO_BASE + GPIO8_REG, (Value | (BIT23 | BIT22)));
//output enable and output high
        Value = MmioRead32(GPIO_BASE + GPIO8_REG);
        Print(L"%a line=%d ,0xfed81650=%x\n", __FUNCTION__, __LINE__, Value);
}

```