

华北工控
NORCO

EMB-7540

V1.2

用户手册

USER'Manual



Industrial & Communication Computer 

做中国最可信赖的工控产品

EMB-7540

V1.2

深圳华北工控股份有限公司：0755-27331166

北京公司：010-82671166

上海公司：021-61212081

成都公司：028-85259319

沈阳公司：024-23960846

西安公司：029-88338386

南京公司：025-58015489

武汉公司：027-87858983

天津公司：022-23727100

新加坡公司：65-68530809

荷兰公司：31-040-2668554

更多产品信息请登陆：www.norco.com.cn

说 明

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。NORCO 是深圳华北工控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。

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，务必仔细阅读产品说明书。
2. 对未准备安装的板卡，应将其保存在防静电保护袋中。
3. 在从包装袋中拿板卡前，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. 在拿板卡时，需佩戴静电保护手套，并且应该养成只触及边缘部分的习惯。
5. 主板与电源连接时，请确认电源电压。
6. 为避免人体被电击或产品被损坏，在每次对主板、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。
7. 在对板卡进行搬动前，先将交流电源线从电源插座中拔掉。
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。
10. 设备在使用过程中出现异常情况，请找专业人员处理。
11. 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，可能需要用户对其干扰采取切实可行的措施。

目 录

第一章 产品介绍	1
1.1 硬件规格	1
第二章 硬件功能	- 3 -
2.1 接口位置和尺寸图	- 3 -
2.2 安装步骤	- 3 -
2.3 跳线功能设置	- 4 -
2.4 接口说明	- 4 -
2.4.1 串口,DI/DO	- 4 -
2.4.2 USB 接口 (USB, J1)	- 6 -
2.4.3 以太网接口 (LAN)	- 6 -
2.4.4 耳机接口 (HP, J1)	- 7 -
2.4.5 重启按键 (RESET)	- 9 -
2.4.6 LED 指示灯	- 9 -
2.4.7 显示接口 HDMI	- 10 -
2.4.8 接口 (TF)	- 15 -
2.4.9 电源接口 (PWR)	- 16 -
2.4.10 MIPI RX 接口 (J12, J13, J14)	- 17 -
2.4.11 前面板接口 (JFP)	- 20 -
第三章 软件功能	21
3.1 Linux 4.9.37 系统	21
3.1.1 mpp sample 简介	21
3.1.2 HDMI 部分	41
3.1.3 LCD 部分	41
3.1.4 USB 部分	41
3.1.5 COM 部分	41
3.1.6 TF 卡部分	41
3.1.7 继电器部分	42
3.2.8 以太网部分	42
3.2.9 声卡和麦克部分	42
3.2.10 摄像头	42
附 录	26

附一：术语表.....26

装箱清单

非常感谢您购买华北工控产品，在打开包装箱后请首先依据装箱清单检查配件，若发现物件有所损坏、或是有任何配件短缺的情况，请尽快与您的经销商联络。

■ EMB-7540 V1.2主板 1片

第一章

产 品 介 绍

华北工控
NORCO

第一章 产品介绍

1.1 硬件规格

尺寸

- Platform: HiSilicon ARM 平台

- 尺寸: 120mmX80mm

处理器

- CPU: 采用 HiSilicon Hi3559A 2*A73+3*A53+1*M7+2*G71+4*DSP+2*NNIE 多核异构处理器

系统内存

- 板载内存: 板载 4GB DDR4 内存

显示

- 提供 1 个 HDMI 接口

存储

- 提供 1 个 TF 卡座, 最大支持 64G

- 板载 16GB iNAND Flash

AUDIO

- 提供 1xMic(Header)

- 提供 1x Line out, 1x Mic (3.5mm 四节耳机孔)

LAN

- 提供 1 个 1000M RJ45(General terminal), 支持 10/100/1000Mbps 网络

I/O

- 串口: 1 个调试串口 232, 1*COM232, 1*COM485

- CMOS Sensor: (根据镜头规格设计转接板), 1x 8K30fps input, Or 2x 4K60fps input, Or

EMB-7540 V1.2

4x4K30fps input, Or 8x1080P30fps input, (搭配 ACP-250 转接板, 最多可支持 4 路 4Kx4K 输入)

- USB: 提供 1x Mini USB2.0(Micro USB), 1xUSB3.0(TYPE A)

扩展接口:

- 提供 2x DI, 1x DO (继电器控制输出);
- 提供 2x Gyroscope;
- 支持 GPIO, I2C, SPI 扩展
- 提供 1xRST BUTTON, 1x PWR LED

电源支持

- 支持单电源+12V 供电, 支持硬件及软件来电自启动功能

看门狗

- 支持硬件复位功能

操作环境

- 工作温度: $-20^{\circ}\text{C}\sim+65^{\circ}\text{C}$
- 存储温度: $-40^{\circ}\text{C}\sim+85^{\circ}\text{C}$
- 操作湿度: 5%~95%, 无凝露

第二章

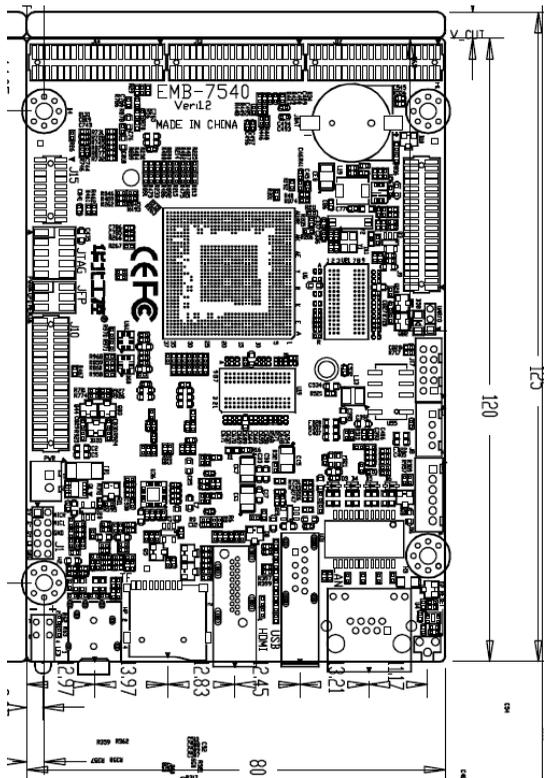
硬件功能

华北工控
NORCO

第二章 硬件功能

2.1 接口位置和尺寸图

下图为 EMB-7540 V1.2 的正面接口位置和尺寸图。在安装设备的过程中必须小心，对于有些部件，如果安装不正确，它将不能正常工作。



注意：操作时，请戴上静电手套，因为静电有可能会损坏部件。

2.2 安装步骤

请依照下列步骤组装您的电脑：

1. 参照用户手册将 EMB-7540 V1.2 上所有 Jumper（跳线帽）调整正确。
2. 安装其他扩展卡。

3. 连接所有信号线、电缆、面板控制线路以及电源供应器。

 **本主板关键元器件都是集成电路，而这些元件很容易因为遭受静电的影响而损坏。因此，请在正式安装主板之前，请先做好以下的准备：**

1. 拿主板时手握板边，尽可能不触及元器件和插头插座的引脚。
2. 接触集成电路元件（如 CPU、RAM 等）时，最好戴上防静电手环/手套。
3. 在集成电路元件未安装前，需将元件放在防静电垫或防静电袋内。
4. 在确认电源的开关处于断开位置后，再插上电源插头。

安装计算机配件之前

遵循以下安全原则有助于防止您的计算机受到潜在的损害并有助于确保您的人身安全。

1. 请确保您的计算机并未连接电源。
2. 接触集成电路元件（如 RAM 等）时，最好戴上防静电手环/手套。

2.3 跳线功能设置

在进行硬件设备安装之前请根据下表按照您的需要对相应的跳线进行设置。

提示：如何识别跳线、接口的第 1 针脚，观察插头插座旁边的文字标记，会用“1”或加粗的线条或三角符号表示；看看背面的焊盘，方型焊盘为第 1 针脚；所有跳线的针脚 1 旁都有 1 个白色箭头。

2.4 接口说明

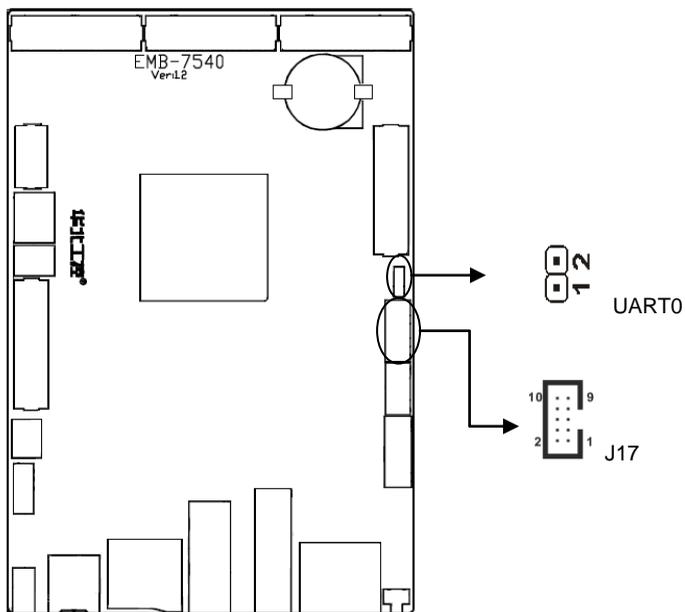
 **连接外部连接器时请先认真阅读本手册，以免对主板造成损坏！**

2.4.1 串口,DI/DO

提供 3 个串口，Uart0 为调试口（不作普通串口使用），

J17 支持 1*RS232, 1*RS485, 2x DI, 1x DO（继电器控制输出）。

EMB-7540 V1.2



UART0:

管脚	信号名称
1	COM0_TX
2	COM0_RX
3	GND

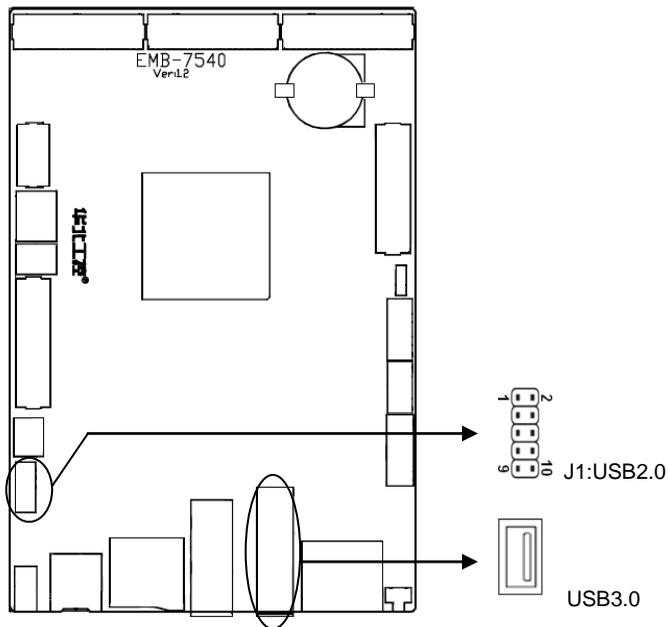
J17:

信号名称	管脚		信号名称
COM3_TXD	1	2	DO1
COM3_RXD	3	4	NO1
GND	5	6	GND
COM4_485+	7	8	IO_IN1
COM4_485-	9	10	IO_IN2

EMB-7540 V1.2

2.4.2 USB 接口 (USB, J1)

提供 1 个 USB2.0 小白插座 J1, 1 个 USB3.0(TYPE A)接口。



USB: 标准 USB3.0 TYPE A 接口定义

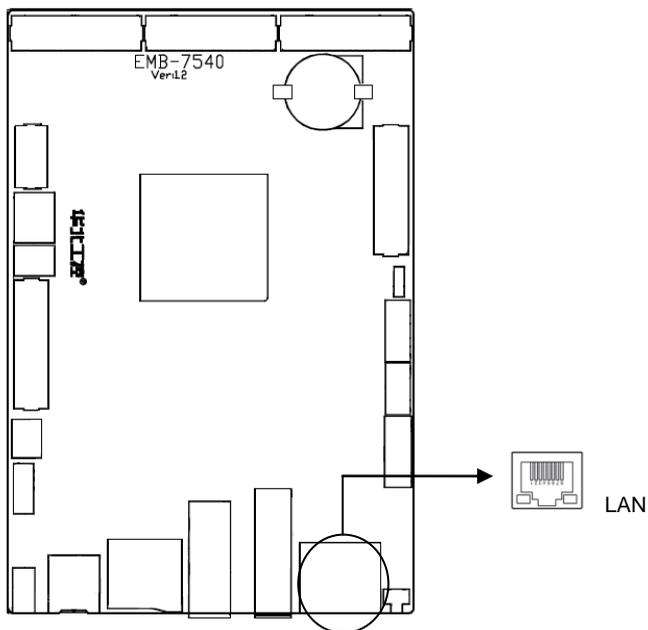
J1:

信号名称	管脚		信号名称
5V	1	2	MICR
USB-	3	4	MICL
USB+	5	6	GND
ID	7	8	/
GND	9	10	/

2.4.3 以太网接口 (LAN)

提供 1 个 RJ45 网络接口，黄色的表示数据传输状态，绿色的表示网络连接状态。

EMB-7540 V1.2



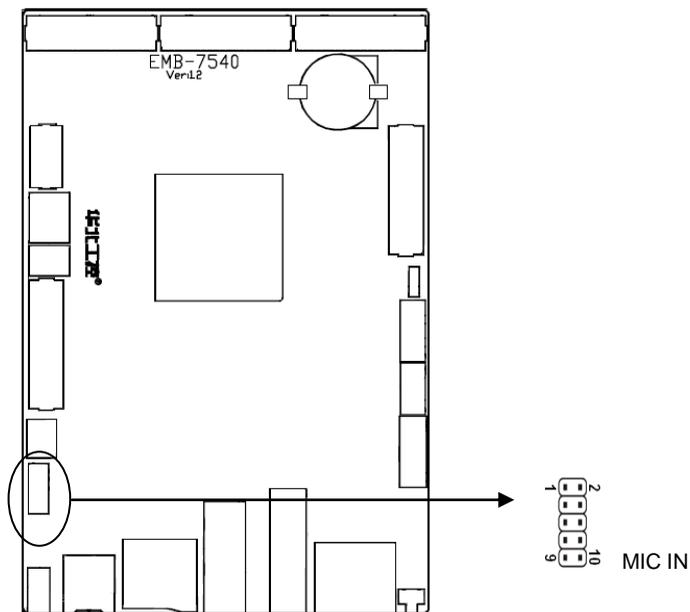
RJ45 LAN LED 状态描述:

LILED (绿色) 状态	功能	ACTLED (黄色) 状态	功能
亮	100/1000M 的连接	闪	进行数据传送
灭	10M 的连接或关闭	灭	数据传送停止

2.4.4 耳机接口 (HP、J1)

支持 2 路 MIC IN, 1 路 Headpone 输出, 提供 1 个四节耳机接口和一路单独的差分 MIC IN 插针接口。

EMB-7540 V1.2



HP: 标准四节耳机国际标准接口定义

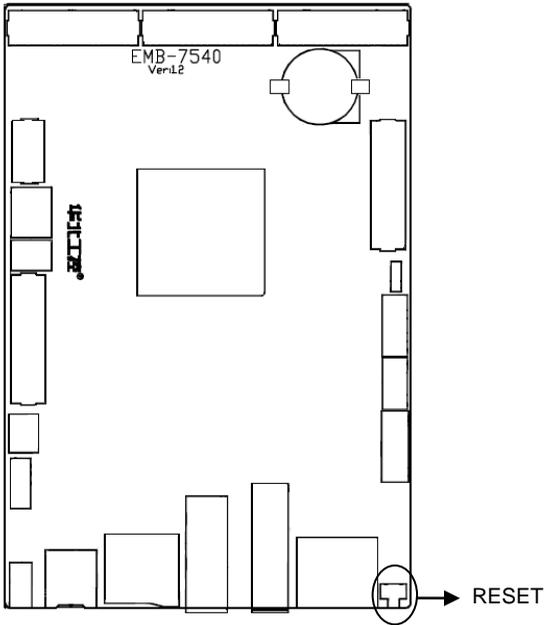


国际标准四段式插针结构

J1:

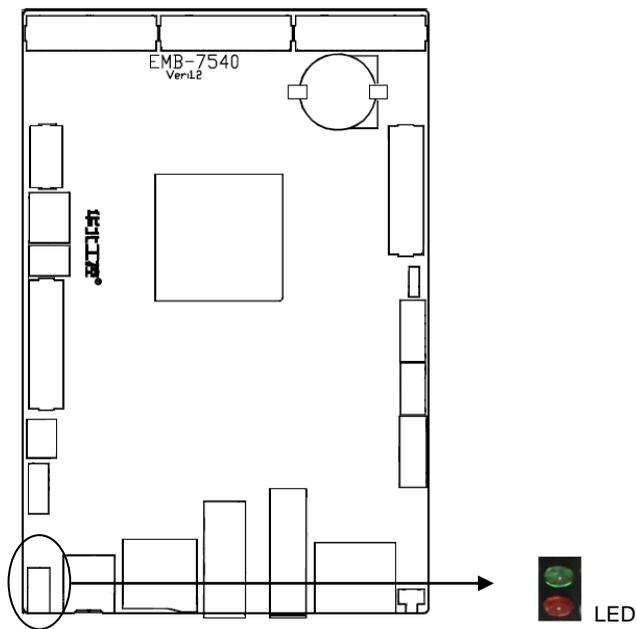
信号名称	管脚		信号名称
5V	1	2	MICR
USB-	3	4	MICL
USB+	5	6	GND
ID	7	8	/
GND	9	10	/

2.4.5 重启按键 (RESET)



2.4.6 LED 指示灯

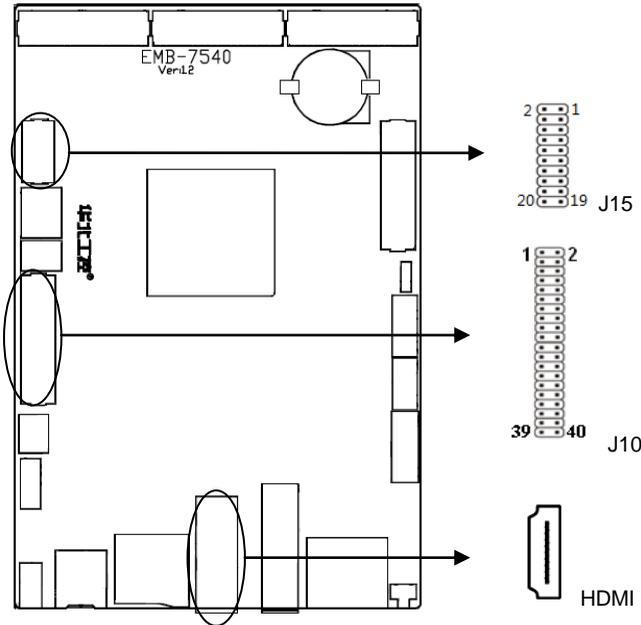
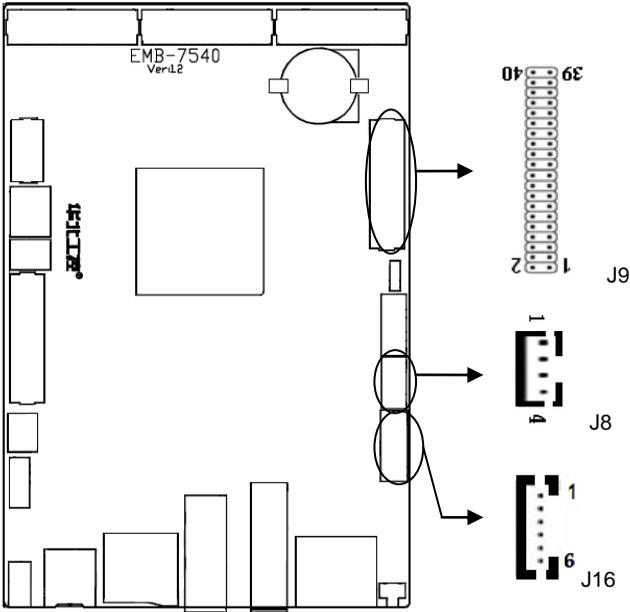
提供双层红色外露 LED 灯，上层灯表示运行灯，下层灯表示电源灯。



2.4.7 显示接口 HDMI

提供 1 个 HDMI 高清显示接口。

EMB-7540 V1.2



EMB-7540 V1.2

HDMI:

信号名称	管脚		信号名称
D2+	1	2	D2 Shield
D2-	3	4	D1+
D1 Shield	5	6	D1-
D0+	7	8	D0 Shield
D0-	9	10	CK+
CK Shield	11	12	CK-
CE Remote	13	14	NC
DDC CLK	15	16	DDC DATA
GND	17	18	+5V
HP DET	19	20	SHELL0
SHELL1	21	22	SHELL2
SHELL3	23	24	SHELL4
SHELL5	25	26	SHELL6
SHELL7	27	28	SHELL8
SHELL9	29	30	SHELL10
SHELL11	31		

J8: OUTPUT FOR AI Control

管脚	信号名称
1	DAMP-
2	DAMP+
3	DRIVE+
4	GND

MFTS:

管脚	信号名称
1	LSLDP
2	LSB2A
3	LSCK
4	LSDIO
5	LSL2B

EMB-7540 V1.2

6	LSB2L
7	LSRST
8	LSDET
9	GND
10	GND
11	VCC4
12	VCC4

J16: 触摸控制

管脚	信号名称
1	GND-
2	3.3V
3	TP_INT/GPIO16_7
4	TP_I2C10_SCL
5	TP_I2C10_SDA
6	TP_RST/GPIO16_4

J9:

信号名称	管脚		信号名称
UART1_RXD/CAN1_RXD_1	1	2	SDIO2_CDATA1
UART1_TXD/CAN1_TXD_1	3	4	SDIO2_CDATA2
UART2_RXD/UART1_RTSN	5	6	SDIO2_CDATA0
UART2_TXD/UART1_CTSN	7	8	SDIO2_CCLK_OUT
SHUB_I2C0_SCL	9	10	SDIO2_CCMD
SHUB_I2C0_SDA	11	12	SDIO2_CDATA3
SHUB_SPI1_SDO	13	14	GND
SHUB_SPI1_SCLK	15	16	SHUB_I2C1_SDA
SHUB_SPI1_CSN	17	18	SHUB_I2C1_SDA
SHUB_SPI1_SDI	19	20	GND
GND	21	22	SHUB_UART0_RXD

EMB-7540 V1.2

WL_REG_ON	23	24	SHUB_UART0_TXD
HOST_WAKEUP_BT	25	26	GND
PWR_EN1	27	28	SHUB_SPI0_SDO
PWR_EN2	29	30	SHUB_SPI0_SDI
PWR_EN0	31	32	SHUB_SPI0_SCLK
PWR_WAKEUP2	33	34	SHUB_SPI0_CSN
PWR_WAKEUP1	35	36	1V8_SHUB
PWR_WAKEUP0	37	38	1V8_SHUB
PWR_WAKEUP0	39	40	3V3_PER

J10:

信号名称	管脚		信号名称
SHUB_PWM_OUT1	1	2	DSI_D0P
GND	3	4	DSI_D0N
SPI4_CSN0	5	6	GND
SPI4_MISO	7	8	DSI_D1P
SPI4_MOSI/I2C8_SDA	9	10	DSI_D1N
SPI4_SCLK/I2C8_SCL	11	12	GND
LCD_RESET	13	14	DSI_CKP
LCD_DATA7	15	16	DSI_CKN
LCD_CLK	17	18	GND
LCD_DATA1	19	20	DSI_D2P
LCD_DATA0	21	22	DSI_D2N
LCD_HSYNC	23	24	GND
LCD_FOLLOW_CONTROL	25	26	DSI_D3P
LCD_DATA5	27	28	DSI_D3N
LCD_DE	29	30	GND
LCD_DATA6	31	32	GND

EMB-7540 V1.2

LCD_DATA4	33	34	1V8_PER
LCD_DATA3	35	36	1V8_PER
LCD_DATA2	37	38	3V3_PER
LCD_VSYNC	39	40	3V3_PER

J15:

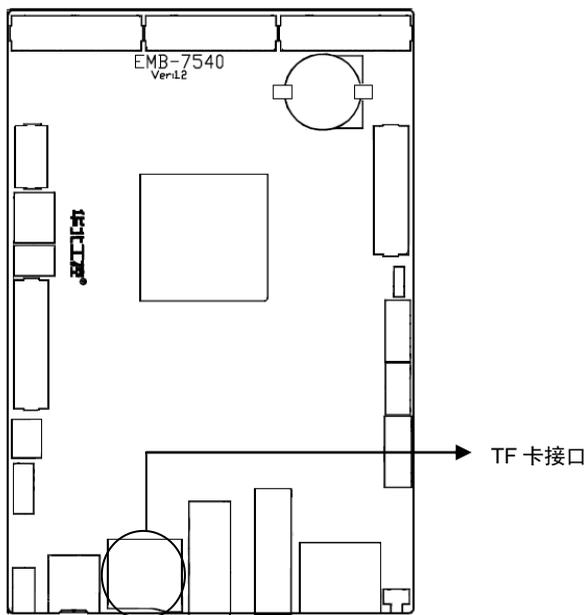
信号名称	管脚		信号名称
12V	1	2	3.3V
12V	3	4	3.3V
PCIE_RXM0	5	6	GND
PCIE_RXP0	7	8	PCIE_RSTN_3V3
GND	9	10	PCIE_CLK_REQ_N
PCIE_TXM0	11	12	NC
PCIE_TXP0	13	14	GPIO12_5
GND	15	16	GND
PCIE_REFCLKM	17	18	NC
PCIE_REFCLKP	19	20	NC

J12~J14:扩展 cmos sensor 插针, 支持 8 路 MIPI 输入。

2.4.8 接口 (TF)

提供一个 TF 卡接口。

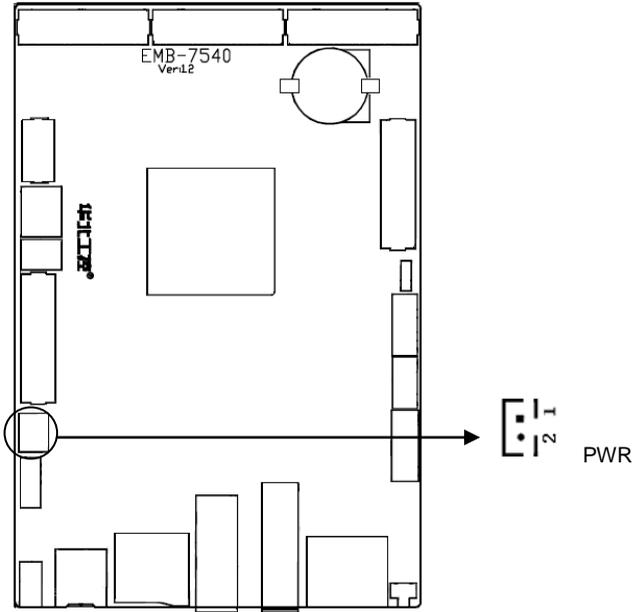
EMB-7540 V1.2



2.4.9 电源接口 (PWR)

板上提供一个 1x2 Pin 小白座电源输入接口。

EMB-7540 V1.2



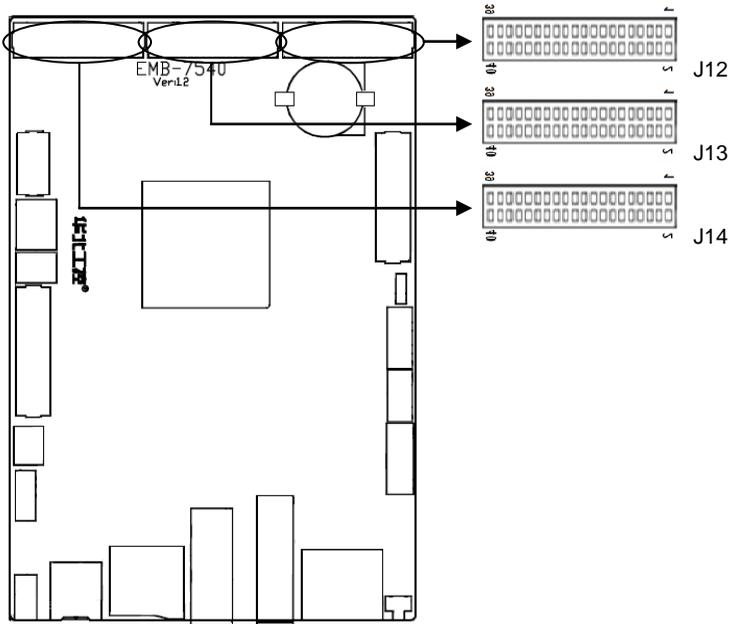
PWR:

管脚	信号名称
1	+12V
2	GND

2.4.10 MIPI RX 接口 (J12, J13, J14)

提供4路MIPI RX 4lane视频输入接口，接口类型：3个1.27mm间距2*20pin双排插针。

EMB-7540 V1.2



J12:

信号名称	管脚		信号名称
5V	1	2	5V
GND	3	4	GND
IR_CUT_0_CONTROL2	5	6	MIPI_RX3_D0N
IR_CUT_0_CONTROL1	7	8	MIPI_RX3_D0P
GND	9	10	MIPI_RX3_CK0N
MIPI_RX3_D1N	11	12	MIPI_RX3_CK0P
MIPI_RX3_D1P	13	14	MIPI_RX3_D2N
MIPI_RX3_CK1N	15	16	MIPI_RX3_D2P
MIPI_RX3_CK1P	17	18	GND
MIPI_RX3_D3N	19	20	MIPI_RX2_CK0N
MIPI_RX3_D3P	21	22	MIPI_RX2_CK0P
GND	23	24	MIPI_RX2_D2N
SENSOR_VS1/PCIE_USB3 _MODE0	25	26	MIPI_RX2_D2P

EMB-7540 V1.2

SENSOR_HS1/SFC_EMMC _BOOT_MODE	27	28	MIPI_RX2_D0N
SENSOR_CLK1_B	29	30	MIPI_RX2_D0P
SENSOR_CLK1_A	31	32	GND
SPI1_SCLK/I2C1_SCL	33	34	MIPI_RX2_D1N
SPI1_SDO/I2C1_SDA	35	36	MIPI_RX2_D1P
SPI1_SDI	37	38	MIPI_RX2_CK1N
SPI1_CSN1/I2C5_SDA	39	40	MIPI_RX2_CK1P

J13:

信号名称	管脚		信号名称
SPI1_CSN0/I2C5_SCL	1	2	MIPI_RX2_D3N
SENSOR_RSTN1	3	4	MIPI_RX2_D3P
GND	5	6	GND
MIPI_RX1_D2N	7	8	MIPI_RX1_CK0N
MIPI_RX1_D2P	9	10	MIPI_RX1_CK0P
MIPI_RX1_D0N	11	12	GND
MIPI_RX1_D0P	13	14	MIPI_RX1_D1N
GND	15	16	MIPI_RX1_D1P
MIPI_RX1_CK1N	17	18	MIPI_RX1_D3N
MIPI_RX1_CK1P	19	20	MIPI_RX1_D3P
GND	21	22	GND
MIPI_RX0_D2P	23	24	MIPI_RX0_CK0P
MIPI_RX0_D2P	25	26	MIPI_RX0_CK0N
GND	27	28	MIPI_RX0_D0P
MIPI_RX0_D3P	29	30	MIPI_RX0_D0N
MIPI_RX0_D3N	31	32	GND
MIPI_RX0_CK1P	33	34	MIPI_RX0_D1P
MIPI_RX0_CK1N	35	36	MIPI_RX0_D1N
GND	37	38	GND
SENSOR_VS0/SFC_DEVIC E_MODE	39	40	SPI0_CSN1/I2C4_SDA

EMB-7540 V1.2

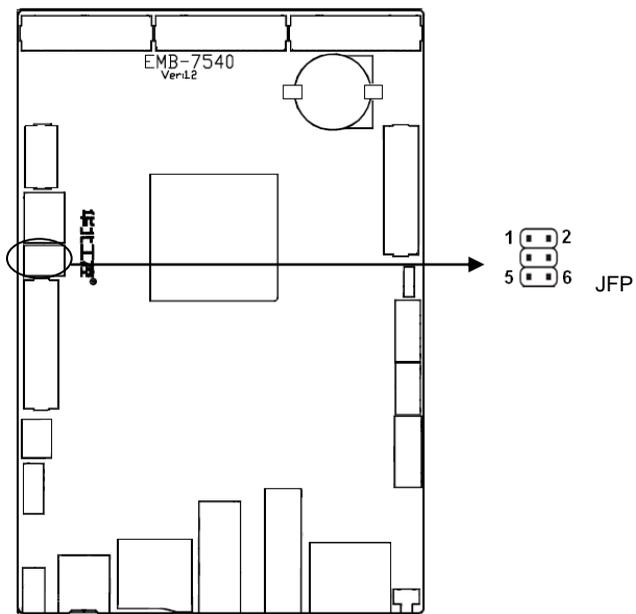
J14:

信号名称	管脚		信号名称
SENSOR_HS0	1	2	SPI0_CSN0/I2C4_SCL
SENSOR_RSTN0	3	4	SPI0_SDI
SENSOR_CLK0_B	5	6	SPI0_SDO/I2C0_SDA
SENSOR_CLK0_A	7	8	SPI0_SCLK/I2C0_SCL
GND	9	10	SPI3_CSN0/I2C7_SCL
SENSOR_VS2/PCIE_REFC LK_SEL	11	12	SPI3_CSN1/I2C7_SDA
SENSOR_HS2/PCIE_USB3 _MODE1	13	14	SPI3_SDI
SPI2_SCLK/I2C2_SCL	15	16	SPI3_SDO/I2C3_SDA
SPI2_SDO/I2C2_SDA	17	18	SPI3_SCLK/I2C3_SCL
SPI2_SDI	19	20	GND
SPI2_CSN1/I2C6_SDA	21	22	SENSOR_HS3/PCIE_D EEMPH_SEL
SPI2_CSN0/I2C6_SCL	23	24	SENSOR_VS3/BOOT_S EL4
SENSOR_RSTN2	25	26	SENSOR_RSTN3
SENSOR_CLK2_B	27	28	SENSOR_CLK3_B
SENSOR_CLK2_A	29	30	SENSOR_CLK3_A
GND	31	32	GND
1.2V	33	34	1.8V
1.2V	35	36	1.8V
12V	37	38	3.3V
12V	39	40	3.3V

2.4.11 前面板接口（JFP）

JFP用于连接至机箱前面板上所设的功能按钮和指示灯。

EMB-7540 V1.2



JFP:

信号名称	管脚		信号名称
PWR_BUTTON0	1	2	GND
UPDATE_MODE_N	3	4	GND
LSADC0_CH0	5	6	GND

BOOT_SEL2:UPDATE_MODE_N	FUNCTION
10	ENABLE UPDATE FROM SDIO0
11	DISABLE UPDATE FROM SDIO0

软件功能

华北工控
NORCO

第三章 软件功能

3.1 Linux 4.9.37 系统

3.1.1 mpp sample 简介

这些 sample 的具体用法详见公司 wiki 网站，如果没有做特殊说明，下面的所有命令都在在串口终端里面输入的。(wiki 网址：android1.norco.com.cn:7070/index.php/EMB-7540)

3.1.1.1 sample_venc 是视频编码录制样例。

~ # ./sample_venc

Usage : ./sample_venc [index]

index:

- 0) H.265e@4K@120fps + H264e@1080p@30fps.
- 1) H.265e@4K@60fps + H264e@4K@60fp.
- 2) Lowdelay:H.265e@4K@30fps.
- 3) IntraRefresh:H.265e@4K@60fps + H264e@4K@60fps.
- 4) Qpmap:H.265e@4K + H264e@4K@60fps.
- 5) Mjpeg@4K@60fps +Jpeg@4K.

shell 说明:

1. 运行 sample_venc (录制) 视频编码样例，需要接摄像头。
2. 使用方法: ./sample_venc [index]
3. 参数<索引>
4. 0) h265 编码,3840×2160 分辨率,每秒 120 帧 + h264 编码,1920x1080 分辨率,每秒 30 帧
5. 1) h265 编码,3840×2160 分辨率,每秒 60 帧 + h264 编码,3840×2160 分辨率,每秒 60 帧
6. 2) Lowdelay(低延时属性): h265 编码,3840×2160 分辨率,每秒 30 帧
7. 3) IntraRefresh(P 帧帧内刷新): h265 编码,3840×2160 分辨率,每秒 60 帧 + h264 编码,3840×2160 分辨率,每秒 60 帧
8. 4) Qpmap(是一种码率控制模式): h265 编码,3840×2160 分辨率 + h264 编码,3840×2160 分辨率,每秒 60 帧

9. 5) Mjpeg(Mjpeg 协议编码方式)的 3840x2160 分辨率,每秒 60 帧 + Jpeg(Jpeg 的编码)的 3840x2160 分辨率

3.1.1.2 sample_vdec 是视频解码播放样例。

~ # ./sample_vdec

Usage : ./sample_vdec <index> <IntfSync>

index:

- 0) VDEC(H265)-VPSS-VO
- 1) VDEC(H264)-VPSS-VO
- 2) VDEC(JPEG->YUV)-VPSS-VO
- 3) VDEC(JPEG->RGB)

IntfSync :

- 0) VO HDMI 4K@30fps.
- 1) VO HDMI 1080P@30fps.

shell 说明:

1. 运行 sample_vdec 视频解码显示样例, 解码文件目录下的视频和图片文件。
2. 使用提示: ./sample_vdec <索引> <Vo 接口时序类型>
3. 参数<索引>:
 4. 0) VDEC 解码器输入(H265 的编码格式视频)--->VPSS(视频处理)--->VO(视频输出)
 5. 1) VDEC 解码器输入(H264 的编码格式视频)--->VPSS(视频处理)--->VO(视频输出)
 6. 2) VDEC 解码器输入(JPEG 格式图片)--->VPSS(视频处理)--->VO(视频输出)
 7. 3) VDEC 解码器输入 JPEG 解码成 RGB 数据
8. 参数<IntfSync>:
 9. 0) 输出到 HDMI 3840x2160 分辨率,每秒 30 帧
 10. 1) 输出到 HDMI 1920x1080 分辨率,每秒 30 帧

3.1.1.3 sample_audio 是音频相关样例。

~ # ./sample_audio

Usage : ./sample_audio <index> <channel>

index:

- 0) start AI to AO loop
- 1) send audio frame to AENC channel from AI, save them
- 2) read audio stream from file, decode and send AO
- 3) start AI(VQE process), then send to AO
- 4) start AI to AO(Hdmi) loop
- 5) start AI to AO(Syschn) loop

channel:

- 0) mic0 input.
- 1) mic1 input.

shell 说明:

1. 运行 sample_audio 音频 (输入/输出/编码/解码) 样例
2. 使用提示: ./sample_audio <索引> <通道>
3. 以下内容是索引对应的功能
4. 0) 音频从输入到输出 (话筒功能)
5. 1) 采集音频输入帧发送到编码通道, 保存起来 (录音功能)
6. 2) 从文件读取音频流, 解码然后发送到输出 (解码播放功能)
7. 3) 通过音频输入声音质量增强处理, 然后输出音频 (声音质量增强处理功能)
8. 4) 采集音频输入到 HDMI 中音频输出 (HDMI 设备播放音频功能)
9. 5) 音频输入到系统音输出 (话筒功能)
10. 音频输入通道
11. 0) mic0 输入(HeadPhone)
12. 1) mic1 输入(插针)

3.1.1.4 sample_snap 是拍照的样例。

~ # ./sample_snap

Usage : ./sample_snap <index>

index:

0)double pipe offline, normal snap.

shell 说明:

1. 运行 sample_snap 拍照
2. 使用说明: ./sample_snap <索引>
3. 参数<索引>
4. 0)双 pipe 离线模式普通拍照

3.1.1.5 sample_dpu_main 是深度图的相关样例。(全称 Depth Process Unit)

~ # ./sample_dpu_main

Usage : ./sample_dpu_main <index>

index:

0) VI->VPSS->RECT->MATCH.

1) FILE->RECT->MATCH.

shell 说明:

1. 运行 sample_dpu_main
2. 使用方法: ./sample_dpu_main <索引>
3. <索引>
4. 0) DPU(Depth Process Unit)对输入的左图像和右图像经过校正和匹配计算得出深度图
5. 1) DPU(Depth Process Unit)对读取的文件的左图像和右图像经过校正和匹配计算得出深度图

3.1.1.6 sample_avs 是全景拼接的样例。(全称 Any View Stitching)

~ # ./sample_avs

Usage : ./sample_avs <index>

index:

- 0) 2 fisheye stitching, Normal projection.
- 1) 4 fisheye stitching, Cube map.
- 2) 4 pic no blend stitching.
- 3) 8 nonfisheye Equirectangular.
- 4) 2 fisheye stitching, Image stabilizing.
- 5) Generate lut.

shell 说明:

- 1. 运行 sample_avs
- 2. 使用说明: ./sample_avs <index>
- 3. <index>
- 4. 0) 2 个鱼眼镜头拼接, 等距柱面投影模式输出
- 5. 1) 4 个鱼眼镜头拼接, 正六面体投影模式输出
- 6. 2) 4 个摄像头非融合拼接 (硬拼)
- 7. 3) 目前暂不支持
- 8. 4) 目前暂不支持
- 9. 5)生成 LUT 表

3.1.1.7 sample_fisheye 是鱼眼模式的样例。

~ # ./sample_fisheye

Usage : ./sample_fisheye <index> <vo intf> <venc type>

index:

- 0) fisheye 360 panorama 2 half with ceiling mount.
- 1) fisheye 360 panorama and 2 normal PTZ with desktop mount.
- 2) fisheye 180 panorama and 2 normal dynamic PTZ with wall mount.
- 3) fisheye source picture and 3 normal PTZ with wall mount.
- 4) nine_lattice preview(Only images larger than or equal to 8M are supported).

vo intf:

- 0) vo HDMI output, default.
- 1) vo BT1120 output.

venc type:

0) H265, default.

1) H264.

shell 说明:

1. 运行 ./sample_fisheye

2. 使用说明 ./sample_fisheye <index> <vo intf> <venc type>

3. index:

4. 0) 2 个“半顶装模式”的鱼眼 360°全景视频

5. 1) 2 个普通“地装的 PTZ”鱼眼 360°全景视频

6. 2) 2 个普通“壁装的 PTZ”鱼眼 180°全景视频

7. 3) 三个鱼眼原画的普通 PTZ 壁装视频

8. 4) 九格预览视频(仅支持大于或等于 8m 的图像)

9. vo intf:

10. 0) 默认 HDMI 输出

11. 1) BT1120 输出

12. venc type:

13. 0) 默认 H265 编码

14. 1) H264 编码

3.1.1.8 sample_vio 是视频输入输出的样例。

~ # ./sample_vio

Usage : ./sample_vio <index> <intf>

index:

0)parallel SDR8 VI - VPSS - VO - HDMI. Embedded isp, phychn channel

preview.

1)online SDR8 VI - VPSS - VO - HDMI. 2 pipe, Embedded isp,

phychn channel pr.

EMB-7540 V1.2

2)offline SDR8 VI - VPSS - VO - HDMI. 4 pipe, Embedded isp, phychn channel pr.

3)online WDR+HDR10 VI - VPSS - VO - HDMI. Embedded isp, phychn channel preview.

4)online SDR8 VI - VO - HDMI. Embedded isp, LDC+ROTATE.

5)online SDR8 VI - VO - HDMI. Embedded isp, FreeRotation.

6)online SDR8 VI - VO - HDMI. Embedded isp, LDC+SPREAD.

7)online SDR8 VI - VPSS - VO - HDMI. Embedded isp, dump raw, phychn channel .

intf:

0) vo HDMI output, default.

1) vo BT1120 output.

shell 说明:

1. 运行 sample_vio

2. 使用方法: ./sample_vio <index> <intf>

3. index:

4. 0) 并行八位标清输入--> 视频处理 -->输出到 HDMI 嵌入式图像信号处理,物理通道预览

5. 1) 在线八位标清输入--> 视频处理 -->输出到 HDMI 2 管道的嵌入式图像信号处理,物理通道预览

6. 2) 离线八位标清输入-->视频处理 -->输出到 HDMI 4 管道的嵌入式图像信号处理,物理通道预览

7. 3) 在线十位高清+宽动态视频输入-->视频处理 -->输出到 HDMI. 嵌入式图像信号处理,物理通道预览

8. 4) 在线八位标清输入 -->输出到 HDMI 嵌入式图像信号处理, 镜头畸变校正和旋转

9. 5) 在线八位标清输入 -->输出到 HDMI 嵌入式图像信号处理, 自由角度旋转

10. 6) 在线八位标清输入 -->输出到 HDMI 嵌入式图像信号处理, 镜头畸变校正和展宽

11. 7) 在线八位标清输入 --> 视频处理 -->输出到 HDMI

12. intf:

13. 0) 默认 HDMI 输出

14. 1) BT1120 输出

3.1.1.9 sample_vgs 是视频图形子系统的样例。（全称 Video Graphics Sub-System）

./sample_vgs

Usage : ./sample_vgs <index> <intf>

index:

0) SDR10 COMPRESS, VI - VGS - VO - HDMI.

intf:

0) vo HDMI output, default.

1) vo BT1120 output.

shell 说明:

1. 运行 sample_vgs

2. 使用说明: ./sample_vgs <index> <intf>

3. index:

4. 0) 10 比特位压缩格式, 输入-->视频图像子系统-->HDMI 输出

5. intf:

6. 0) 默认 HDMI 输出

7. 1) BT1120 输出

3.1.1.10 sample_tde 是利用硬件为 OSD 和 GUI 提供快速的图形绘制功能的样例。（全称 Two Dimensional Engine）

~ # ./sample_tde

Usage : ./sample_tde <intf>

intf:

0) vo VGA output, default.

1) vo HDMI output.

shell 说明:

1. 运行 sample_tde
2. 使用说明: ./sample_tde <intf>
3. intf:
4. 0) 默认 VGA 输出演示
5. 1) HDMI 输出演示

3.1.1.11 sample_hifb 是 Framebuffer 功能的样例。

~ # ./sample_hifb

Usage : ./sample_hifb <index> <device> <intf>

/*****index*****/

please choose the case which you want to run:

- 0: ARGB8888 standard mode
- 1: ARGB1555 BUF_DOUBLE mode
- 2: ARGB1555 BUF_ONE mode
- 3: ARGB1555 BUF_NONE mode
- 4: ARGB1555 BUF_ONE mode with compress
- 5: ARGB8888 BUF_ONE mode with compress

/*****device*****/

- 0) VO device 0#, default.
- 1) VO device 1#.

/*****intf*****/

- 0) VO HDMI output, default.
- 1) VO BT1120 output.

shell 说明:

1. 运行 sample_hifb
2. 使用说明: ./sample_hifb <index> <device> <intf>

5. <index>

6. 请选择一下您想运行的情况之一

7. 0: ARGB8888 像素格式(标准模式)

8. 1: ARGB1555 像素格式双缓存模式

9. 2: ARGB1555 像素格式单缓存模式

10. 3: ARGB1555 像素格式无缓存模式

11. 4: ARGB1555 像素格式带压缩的单缓存模式

12. 5: ARGB8888 像素格式带压缩的单缓存模式

14. <device>

15. 0) 默认输出到超清显示设备 DHD0 (能够支持到 4K)

16. 1) 输出到高清显示设备 DHD1

18. <intf>

19. 0) 默认 HDMI 输出

20. 1) BT1120 输出

3.1.1.12 sample_awb_calibration 是自动白光平衡测量的样例。

~ # ./sample_awb_calibration

Usage : ./sample_awb_calibration <mode> <intf1> <intf2> <intf3>

mode:

0) Calculate Sample gain.

1) Adjust Sample gain according to Golden Sample.

intf1:

The value of Rgain of Golden Sample. 深红色区域的锐化增益控制。

intf2:

The value of Bgain of Golden Sample.

intf3:

The value of Alpha ranging from 0 to 1024 (The strength of adusting Sampe Gain

will increase with the value of Alpha) .

shell 说明:

1. 运行 sample_awb_calibration
2. 使用方法: ./sample_awb_calibration <mode> <intf1> <intf2> <intf3>
3. <mode>
4. 0) 计算例子中的增益
5. 1) 根据标准例子来校正例子中的增益
6. intf1:
7. 标准例子中的深红色区域的锐化增益控制值
8. intf2:
9. 标准例子中的深蓝色区域的锐化增益控制值
10. intf3:
11. 0~1024 范围的初始值(采样数据的增益强度会随着初始值的值增加而增加)

3.1.1.13 sample_modeswitch 是模式切换样例。

~ # ./sample_modeswitch

Usage : ./sample_modeswitch <index> <intf>

index:

0)mode switch wdr to linear for imx290 VI - VO - HDMI. Embedded isp, phychn channel preview.

1)mode switch linear to wdr for imx290 VI - VO - HDMI+BT1120. Embedded isp, phychn channel preview.

2)resolution 9M50FPS to 8M30FPS for imx477 VI - VO - HDMI. Embedded isp, phychn channel preview.

3)resolution 8M30FPS to 9M50FPS for imx477 VI - VO - HDMI. Embedded isp, phychn channel preview.

intf:

0) vo HDMI output, default.

1) vo BT1120 output.

shell 说明:

1. 运行 sample_modeswitch

2. 使用说明: ./sample_modeswitch <index> <intf>

3. index:

4. 0) imx290 的 wdr 模式到 linear 模式切换

5. 1) imx290 的 linear 模式到 wdr 模式切换

6. 2) imx477 的分辨率 9M50FPS 切换至 8M30FPS

7. 3) imx477 的分辨率 8M30FPS 切换至 9M50FPS

8. intf:

9. 0) 默认 HDMI 输出

10. 1) BT1120 输出

3.1.1.14 sample_ive_main 是智能加速引擎的样例。

~ # ./sample_ive_main

Usage : ./sample_ive_main <index> [complete] [encode] [vo]

index:

0)BgModel,<encode>:0, not encode;1,encode.<vo>:0,not call vo;1,call vo.(VI->VPSS->IVE->VGS->[VENC_H264]->[VO_HDMI]).

1)Gmm,<encode>:0, not encode;1,encode.<vo>:0,not call vo;1,call vo.(VI->VPSS->IVE->VGS->[VENC_H264]->[VO_HDMI]).

2)Occlusion detected.(VI->VPSS->IVE->VO_HDMI).

3)Motion detected.(VI->VPSS->IVE->VGS->VO_HDMI).

4)Canny,<complete>:0, part canny;1,complete canny.(FILE->IVE->FILE).

5)Gmm2.(FILE->IVE->FILE).

6)MemoryTest.(FILE->IVE->FILE).

7)Sobel.(FILE->IVE->FILE).

8)Ann.(FILE->IVE->STDIO).

9)St Lk.(FILE->IVE->FILE).

a)Svm.(FILE->IVE->STDIO).

b)Cnn.(FILE->IVE->STDIO).

shell 说明:

1. 运行 sample_ive_main

2. 使用说明: ./sample_ive_main <index> [complete] [encode] [vo]

3. index:

4. 0) 背景模型

5. 1) 高斯模型 Gmm

6. 2) 遮挡检测

7. 3) 运动检测

8. 4) 边缘检测

9. 5) 高斯模型 Gmm2

10 6) 内存测试

11 7) Sobel 算子分割

12 8) 图像检索 Ann

13 9) St 和 LK 光流法

14 a) 分类器 SVM

15 b) 神经网络 Cnn

3.1.1.15 sample_dis 是数字防抖的样例。(全称 Digital Image Stabilization)

~ # ./sample_dis

Usage : ./sample_dis <index> <intf>

index:

0)DIS-4DOF_GME.VI-VO VENC.

1)DIS-6DOF_GME.VI-VO VENC.

intf:

0) vo HDMI output, default.

1) vo BT1120 output.

shell 说明:

1. 运行 sample_dis

2. 使用说明: ./sample_dis <index> <intf>

3. index:

4. 0) DIS-4DOF_GME(四自由度 GME 算法,不使用陀螺仪), 输入—输出 同时 H265 格式录像(存储在当前)

5. 1) DIS-6DOF_GME(六自由度 GME 算法,不使用陀螺仪), 输入—输出 同时 H265 格式录像(存储在当前)

6. intf:

7. 0) HDMI 输出

8. 1) BT1120 输出

3.1.1.16 sample_dsp_main 是视觉处理加速的专用处理器 DSP 的样例。

~ # ./sample_dsp_main

shell 说明:

1. 四个 DSP 之 DSP0 的出图测试。

3.1.1.17 sample_nnie_main 是神经网络加速引擎样例。

~ # ./sample_nnie_main

Usage : ./sample_nnie_main <index>

index:

0) RFCN(VI->VPSS->NNIE->VGS->VO).

1) Segnet(Read File).

- 2) FasterRcnnAlexnet(Read File).
- 3) FasterRcnnDoubleRoiPooling(Read File).
- 4) Cnn(Read File).
- 5) SSD(Read File).
- 6) Yolov1(Read File).
- 7) Yolov2(Read File).
- 8) LSTM(Read File).

shell 说明:

- 1. 运行 sample_nnie_main
- 2. 使用说明: ./sample_nnie_main <index>
- 3. index:
- 4. 0) RFCN 模型, 从摄像头采集图像标定人形。
- 5. 1) 可训练的图像分割 Segnet
- 6. 2) 深度学习的目标检测 Alexnet
- 7. 3) 深度学习的目标检测 DoubleRoiPooling
- 8. 4) 神经网络 Cnn
- 9. 5) 可训练的 SSD 模型处理
- 10. 6) 神经网络模型 Yolov1
- 11. 7) 神经网络模型 Yolov2
- 12. 8) LSTM 模型

3.1.1.18 vi_bayerdump 是用来获取 sensor 原始图像工具的样例。

~ # ./vi_bayerdump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vi_bayerdump -h

Usage: ./vi_bayerdump [ViPipe] [nbit] [Compress] [FrmCnt] [ByteAlign] [RatioShow]

ViPipe:

0:ViPipe0 ~ 7:ViPipe 7

nbit:

The bit num to be dump

Compress:

Whether to dump compressed raw, default is 0

FrmCnt:

the count of frame to be dump

ByteAlign:

Whether convert to Byte align, default is 1

RatioShow:

Whether the file name of raw show ratio info, default is 1

e.g : ./vi_bayerdump 0 16 0 2 1 1

shell 说明:

注意: 此工具仅用于测试使用!

使用说明: ./vi_bayerdump [ViPipe] [nbit] [Compress] [FrmCnt] [ByteAlign] [RatioShow]

ViPipe: ViPipe 号 0~7 范围内

nbit: 原始数据的字节宽度 支持 8/10/12/14/16bit

Compress: 解码图像压缩模式: 是否转存原始图像压缩数据, 默认是0 0代表不压缩

FrmCnt: 获取 VI 框架个数

ByteAlign: 是否转换为字节对齐, 默认值为 1

RatioShow: 文件名上是否对比率信息原始展现, 默认为 1

3.1.1.19 vi_chn_dump 是输入通道视频帧转储的样例。

~ # ./vi_chn_dump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vi_chn_dump -h

Usage: ./vi_chn_dump [ViPipe] [ViChn] [FrmCnt] [ByteAlign]

1)ViPipe:

Vi pipe id

2)ViChn:

vi chn id

3)FrmCnt:

the count of frame to be dump

4)ByteAlign:

Whether convert to Byte align , default is 1

*)Example:

e.g : ./vi_chn_dump 0 0 2 1

shell 说明:

注意: 此工具仅用于测试!!!

使用说明: ./vi_chn_dump [ViPipe] [ViChn] [FrmCnt] [ByteAlign]

ViPipe: PIPE 号, 范围0~7

ViChn: vi 通道号

FrmCnt: 视频帧数

ByteAlign: 是否转换为字节对齐, 默认值为 1

3.1.1.20 vo_chn_dump 是输出通道视频帧转储的样例。

~ # ./vo_chn_dump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vo_chn_dump -h

Usage: ./vo_chn_dump [VoLayer] [VoChn] [Frmcnt].

1)VoLayer:

which layer to be dump 视频输出视频层号。

Default: 0

2)VoChn:

which channel to be dump

Default: 0

3)FrmCnt:

the count of frame to be dump

Default: 1

*)Example:

e.g : ./vo_chn_dump 0 0 1 (dump one YUV)

shell 说明:

注意: 此工具仅用于测试使用!!!

使用说明: ./vo_chn_dump [VoLayer] [VoChn] [Frmcnt].

VoLayer: 选择要转储的视频输出视频层号, 默认是0

VoChn: 选择要转储的视频输出通道, 默认是0

FrmCnt: 需要转储的视频帧个数, 默认 1

3.1.1.21 vo_screen_dump 是转储输出屏幕图像数据的样例。

~ # ./vo_screen_dump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vo_screen_dump -h

Usage: ./vo_screen_dump [VoLayer] [Frmcnt].

1)VoLayer:

Which layer to be dumped

Default: 0

2)FrmCnt:

The count of frame to be dumped

Default: 1

*)Example:

e.g : ./vo_screen_dump 0 1 (dump one YUV)

shell 说明:

注意: 此工具仅用于测试使用!!!

使用说明: ./vo_screen_dump [VoLayer] [FrmCnt].

VoLayer: 选择要转储的视频输出视频层号, 默认0

FrmCnt: 需要转储的视频帧个数, 默认1

3.1.1.22 vo_wbc_dump 是回写设备的输出图像数据转储的样例。

~ # ./vo_wbc_dump

NOTICE: This tool only can be used for TESTING !!!

usage: ./vo_wbc_dump 0 [frmCnt]. sample: ./vo_wbc_dump 0 5

Dev(0) HI_MPI_VO_SetWbcDepth errno 0xa00f8010

shell 说明:

注意: 此工具仅用于测试使用!!!

使用说明: ./vo_wbc_dump 0 [frmCnt].

第一个参数: 回写设备号

[frmCnt] 输出视频的帧数

3.1.1.23 vpss_chn_dump 是用户从 VPSS 通道获取 Cnt 个帧处理完成图像转储的样例。

~ # ./vpss_chn_dump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vpss_chn_dump -h

Usage: ./vpss_chn_dump [VpssGrp] [VpssChn] [FrmCnt]

1)VpssGrp:

Vpss group id

2)VpssChn:

vpss chn id

3)FrmCnt:

the count of frame to be dump

*)Example:

e.g : ./vpss_chn_dump 0 0 1

e.g : ./vpss_chn_dump 1 4 2

shell 说明:

注意: 此工具仅用于测试使用!!!

使用说明: ./vpss_chn_dump [VpssGrp] [VpssChn] [FrmCnt]

VpssGrp: VPSS GROUP 号

VpssChn: VPSS 通道号

FrmCnt: 转储的视频帧个数

3.1.1.24 vpss_src_dump 是转储视频缓存池帧数据的样例。

~ # ./vpss_src_dump -h

NOTICE: This tool only can be used for TESTING !!!

To see more usage, please enter: ./vpss_src_dump -h

Usage: ./vpss_src_dump [Grp] [Pipe]

1)VpssGrp:

Vpss group id

2)Pipe:

Vpss Grp pipe

*)Example:

e.g : ./vpss_src_dump 0 0

shell 说明:

注意: 此工具仅用于测试使用!!!

使用说明: ./vpss_src_dump [Grp] [Pipe]

VpssGrp: VPSS GROUP 号

Pipe: VPSS 组的管道号

3.1.2 HDMI 部分

支持 HDMI 输出

使用说明:使用 HDMI 线将主板与 4K 显示器相连, 运行视频解码样例, 会在 HDMI 上输出 4K 或 1080P 的图像。

3.1.3 LCD 部分

支持 lvds lcd 接口输出

需根据客户实际使用 LCD 屏订制驱动。

3.1.4 USB 部分

支持 1xUSB3.0 1x Mini USB2.

将 U 盘插入 USB 口, 会有很多提示信息, 其中下面标红的部分比较重要。

```
~ # usb 1-1: new high-speed USB device number 3 using xhci-hcd  
sd 0:0:0:0: [sda] Assuming drive cache: write through
```

```
sda: sda1
```

然后通过, mount -t vfat /dev/sda1 /tmp 命令进行 U 盘挂载, 在 tmp 目录可以看到 U 盘内容。

3.1.5 COM 部分

支持 2 个串口, 一个 debug 口。

1x COM232; 1x COM485; 一个 debug 串口。

分别对应的设备是/dev/ttyAMA3, /dev/ttyAMA4, /dev/ttyS000

3.1.6 TF 卡部分

支持

使用说明:将 tf 卡接入到主板, 重启系统, 在终端会有 mmc1: new high speed SD card at address 0001 提示。

启动之后通过命令: mount -t vfat /dev/mmcblk1p1 /tmp/进行挂载, 在 /tmp 目录下面就是 tp 卡里面的内容。

3.1.7 继电器部分

支持继电器控制。

通过操作 GPIO102 可以控制继电器的常开和闭合。

```
echo 102 > /sys/class/gpio/export           //导出 GPIO102
echo out > /sys/class/gpio/gpio102/direction //设置 GPIO 方向为输出
echo 1 > /sys/class/gpio/gpio102/value      //控制继电器常开
echo 0 > /sys/class/gpio/gpio102/value      //控制继电器常闭
echo 102 > /sys/class/gpio/unexport         //取消 GPIO102 的导出
```

3.2.8 以太网部分

支持，使用说明:链接设备网卡到局域网里面，在终端需要用 ifconfig 配置默认 IP。可以通过 ping 命令 ping 局域网里面的其他电脑。

3.2.9 声卡和麦克部分

支持，将带麦克的耳机输入。支持麦克输入，本地声卡输出、HDMI 声音输出

使用说明:

运行 ./sample_audio 样例:

./sample_audio 1 0 进行声音录制，这时说话会在当前目录生成 audio_chn0.aac 文件存储起来，运行./sample_audio 2 0 进行声音播放，播放存储的录音文件。

3.2.10 摄像头

目前支持四个索尼 imx 系列的 sensor 接口。

根据客户实际使用情况可以订制相同接口定义的其他 sensor。

附
录

华北工控
NORCO

附 录

附一：术语表

ACPI

高级配置和电源管理。ACPI规范允许操作系统控制计算机及其附加设备的大部分电能。

Windows 98/98SE, Windows 2000和Windows ME全部都支持此规范, 让用户能灵活管理系统的电能。

BIOS

基本输入/输出系统。是在PC中包含所有的输入/输出控制代码界面的软件。它在系统启动时进行硬件检测, 开始操作系统的运作, 在操作系统和硬件之间提供一个界面。BIOS是存储在一个只读存储器芯片内。

BUS

总线。在计算机系统中, 不同部件之间交换数据的通道, 是一组硬件线路。我们所指的BUS通常是CPU和主内存元件内部的局部线路。

Chipset

芯片组。是为执行一个或多个相关功能而设计的集成芯片。我们指的是由南桥和北桥组成的系统级芯片组, 他决定了主板的架构和主要功能。

CMOS

互补金属-氧化物半导体。是一种被广泛应用的半导体类型。它具有高速、低功耗的特点。我们指的CMOS是在主板上的CMOS RAM中预留的一部分空间, 用来保存日期、时间、系统信息和系统参数设定信息等。

COM

串口。一种通用的串行通信接口, 一般采用标准DB 9公头接口连接方式。

DIMM

双列直插式内存模块。是一个带有内存芯片组的小电路板。提供64bit的内存总线宽度。

DRAM

动态随机存取存储器。是一个普通计算机的通用内存类型。通常用一个晶体管和一个电容来存储一个位。随着技术的发展，DRAM的类型和规格已经在计算机应用中变得越来越多样化。例如现在常用的就有：SDRAM、DDR SDRAM和RDRAM。

LAN

局域网网络接口。一个小区域内相互关联的计算机组成的一个计算机网络，一般是在一个企事业单位或一栋建筑物。局域网一般由服务器、工作站、一些通信链接组成，一个终端可以通过电线访问数据和设备的任何地方，许多用户可以共享昂贵的设备和资源。

LED

发光二极管，一种半导体设备，当电流流过时它会被点亮，通常用来把信息非常直观地表示出来，例如表示电源已经导通或硬盘驱动器正在工作等。

PnP

即插即用。允许PC对外接设备进行自动配置，不用用户手动操作系统就可以自己工作的一种规格。为实现这个特点，BIOS支持PnP和一个PnP扩展卡都是必需的。

POST

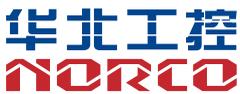
上电自检。在启动系统期间，BIOS会对系统执行一个连续的检测操作，包括检测RAM，键盘，硬盘驱动器等，看它们是否正确连接和是否正常工作。

PS/2

由IBM发展的一种键盘和鼠标连接的接口规范。PS/2是一个仅有6PIN的DIN接口，也可以用连接其他的设备，比如调制解调器。

USB

通用串行总线。一种适合低速外围设备的硬件接口，一般用来连接键盘、鼠标等。一台PC最多可以连接127个USB设备，提供一个12Mbit/s的传输带宽；USB支持热插拔和多数数据流功能，即在系统工作时可以插入USB设备，系统可以自动识别并让插入的设备正常。



敬请参阅

<http://www.norco.com.cn>

本手册所提供信息可不经事先通知进行变更

华北工控对所述信息保留解释权

