



正点原子公司名称：广州市星翼电子科技有限公司

原子哥在线教学平台：www.yuanzige.com

开源电子网 / 论坛：<http://www.openedv.com/forum.php>

正点原子淘宝店铺：<https://openedv.taobao.com>

正点原子官方网站：www.alientek.com

正点原子 B 站视频：<https://space.bilibili.com/394620890>

电话：020-38271790 传真：020-36773971

请关注正点原子公众号，资料发布更新我们会通知。

请下载原子哥 APP，数千讲视频免费学习，更快更流畅。



扫码关注正点原子公众号



扫码下载“原子哥”APP

文档更新说明

| 版本 | 版本更新说明 | 负责人 | 校审 | 发布日期 |
|------|------------------|------------------|------------------|-----------|
| V1.0 | 初稿 | 正点原子 linux 团队 | 正点原子 linux 团队 | 2022.3.11 |
| V1.1 | 修改部分描述以及增加部分客户案例 | 正点原子 linux 团队 | 正点原子 linux 团队 | 2022.7.5 |

目录

| | |
|---|----|
| 前言 | 5 |
| 第一章 最小系统设计..... | 6 |
| 1.1 电源设计说明..... | 6 |
| 1.2 系统启动配置..... | 6 |
| 1.3 复位信号..... | 7 |
| 1.4 核心板固件烧写电路..... | 7 |
| 第二章 外设接口注意事项..... | 8 |
| 2.1 USB 接口..... | 8 |
| 2.2 SDIO 接口 | 9 |
| 2.3 调试串口..... | 9 |
| 2.4 网口..... | 10 |
| 第三章 核心板上的部分管脚..... | 11 |
| 3.1 RGB-LCD 管脚..... | 11 |
| 3.2 GPIO5 (GPIO1-05) | 12 |
| 3.3 RTC 部分..... | 12 |
| 3.4 PMIC_ON_REQ 管脚 | 12 |
| 第四章 底板设计后样板的一些问题及调试思路..... | 13 |
| 4.1 核心板启动不了..... | 13 |
| 4.2 外设接口的一些问题..... | 14 |
| 4.3 使用 MFGTOOL 工具烧录固件的一些问题..... | 14 |
| 第五章 特殊需求..... | 14 |
| 5.1 底板有对静电、信号隔离之类的要求请自行增加对应的处理避免损坏核心板。 14 | |
| 5.2 自行设计供电电路需要注意的几个要求..... | 14 |
| 第六章 部分案例..... | 14 |

前言

本文档的主要内容是基于阿尔法开发板底板进行测试得到的,均适用于 B2B 核心板和邮票孔核心板。

核心板的具备的外设资源可以参考【正点原子】I.MX6ULL 核心板规格书.pdf 核心板上的引脚复用可以查看 imx6ull 核心板引脚分配图.pdf 或者使用 NXP 官方的引脚配置工具 CONFIG-TOOLS-IMX 进行调整。

规格书和引脚分配图这两个 pdf 在开发板资料 a 盘-> 硬件资料-> 核心板封装资料 下。资料盘下载链接 <https://pan.baidu.com/s/1inZtdngN-L3aVfoch2-sKA> 提取码: m65i

NXP 的官方引脚配置工具 CONFIG-TOOLS-IMX 的下载链接 [面向 i.MX 应用处理器的配置工具 NXP 半导体](#)

核心板引脚分布图 B2B 核心板:

| IMX6UL_CORE | | IMX6UL_CORE | |
|-------------|-------|-------------|------------------------------|
| CSI_HSYNC | J1 1 | J1 60 | CSI_VSYNC |
| CSI_MCLK | J1 2 | J1 59 | CSI_DATA3 |
| CSI_DATA2 | J1 3 | J1 58 | CSI_DATA7 |
| CSI_DATA6 | J1 4 | J1 57 | CSI_DATA1 |
| CSI_PIXCLK | J1 5 | J1 56 | CSI_DATA0 |
| CSI_DATA5 | J1 6 | J1 55 | CSI_DATA4 |
| LCD_DATA0 | J1 7 | J1 54 | SDI_CLK |
| LCD_DATA1 | J1 8 | J1 53 | SDI_CMD |
| LCD_DATA2 | J1 9 | J1 52 | SDI_DATA2 |
| LCD_DATA3 | J1 10 | J1 51 | SDI_DATA3 |
| LCD_DATA4 | J1 11 | J1 50 | SDI_DATA1 |
| LCD_DATA5 | J1 12 | J1 49 | SDI_DATA0 |
| LCD_DATA6 | J1 13 | J1 48 | SNVS_TAMPER9 CT_RST |
| LCD_DATA7 | J1 14 | J1 47 | GPIO_5 SDI_VSELECT |
| LCD_DATA8 | J1 15 | J1 46 | LCD_DE |
| LCD_DATA9 | J1 16 | J1 45 | LCD_PCLK |
| LCD_DATA10 | J1 17 | J1 44 | LCD_HSYNC |
| LCD_DATA11 | J1 18 | J1 43 | LCD_VSYNC |
| LCD_DATA12 | J1 19 | J1 42 | GPIO_5 |
| LCD_DATA13 | J1 20 | J1 41 | PMIC_ON_REQ VDD_COIN_3V |
| LCD_DATA14 | J1 21 | J1 40 | RESET |
| LCD_DATA15 | J1 22 | J1 39 | SNVS_TAMPER6 ENET1_INT_TREE# |
| LCD_DATA16 | J1 23 | J1 38 | ENET2_RXD0 |
| LCD_DATA17 | J1 24 | J1 37 | ENET2_RXD1 |
| LCD_DATA18 | J1 25 | J1 36 | ENET2_TXD0 |
| LCD_DATA19 | J1 26 | J1 35 | ENET2_TXD1 |
| LCD_DATA20 | J1 27 | J1 34 | ENET2_RXER |
| LCD_DATA21 | J1 28 | J1 33 | ENET2_CRSDV |
| LCD_DATA22 | J1 29 | J1 32 | ENET2_TXEN |
| LCD_DATA23 | J1 30 | J1 31 | ENET2_TX_CLK |

邮票孔核心板:

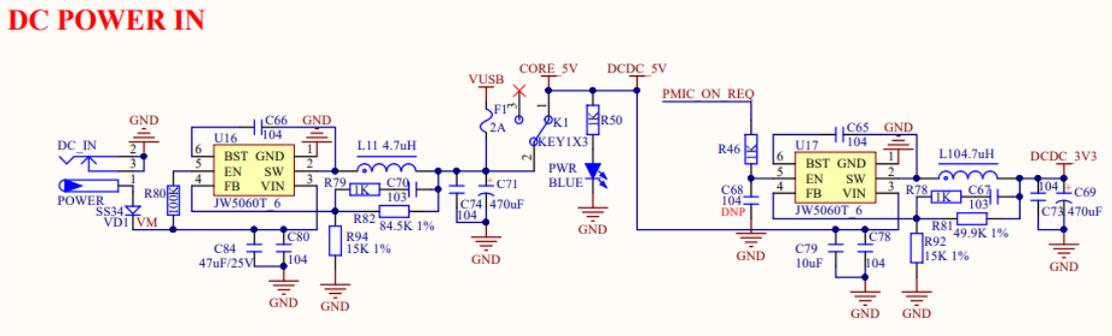
| | | | |
|---------------|-----|-----------------------|-----|
| CSI_VSYNC | 80 | GND | 80 |
| CSI_MCLK | 81 | ENET1_TX_CLK | 81 |
| CSI_DATA7 | 82 | ENET1_TX_DATA0 | 82 |
| CSI_DATA6 | 83 | ENET1_TX_DATA1 | 83 |
| CSI_PIXCLK | 84 | ENET1_TX_ERXENET_RXER | 84 |
| CSI_DATA5 | 85 | ENET1_TX_DATA2 | 85 |
| CSI_DATA4 | 86 | ENET1_TX_DATA3 | 86 |
| SDI_CMD | 87 | ENET1_TX_DATA4 | 87 |
| SDI_DATA2 | 88 | ENET1_TX_DATA5 | 88 |
| SDI_DATA3 | 89 | ENET1_TX_DATA6 | 89 |
| SDI_DATA1 | 90 | ENET1_TX_DATA7 | 90 |
| SDI_DATA0 | 91 | ENET1_TX_ERXENET_RXER | 91 |
| SNVS_TAMPER9 | 92 | ENET1_TX_DATA8 | 92 |
| GPIO_005 | 93 | ENET1_TX_DATA9 | 93 |
| GPIO_009 | 94 | ENET1_TX_DATA10 | 94 |
| GPIO_002 | 95 | ENET1_TX_DATA11 | 95 |
| SNVS_TAMPER2 | 96 | ENET1_TX_DATA12 | 96 |
| SNVS_TAMPER5 | 97 | ENET1_TX_DATA13 | 97 |
| SNVS_TAMPER7 | 98 | ENET1_TX_DATA14 | 98 |
| SNVS_TAMPER8 | 99 | ENET1_TX_DATA15 | 99 |
| BOOT_MODE0 | 100 | ENET1_TX_DATA16 | 100 |
| BOOT_MODE1 | 101 | ENET1_TX_DATA17 | 101 |
| BOOT_MODE2 | 102 | ENET1_TX_DATA18 | 102 |
| SNVS_TAMPER0 | 103 | ENET1_TX_DATA19 | 103 |
| SNVS_TAMPER1 | 104 | ENET1_TX_DATA20 | 104 |
| SNVS_TAMPER4 | 105 | ENET1_TX_DATA21 | 105 |
| 3V | 106 | ENET1_TX_DATA22 | 106 |
| 5V | 107 | ENET1_TX_DATA23 | 107 |
| GND | 108 | ENET1_TX_CLK | 108 |
| USB_OTG2_VBUS | 109 | ENET1_TX_DATA0 | 109 |
| USB_OTG1_DP | 110 | ENET1_TX_DATA1 | 110 |
| USB_OTG1_DM | 111 | ENET1_TX_DATA2 | 111 |
| USB_OTG1_DP | 112 | ENET1_TX_DATA3 | 112 |
| USB_OTG1_DM | 113 | ENET1_TX_DATA4 | 113 |
| GPIO_000 | 114 | ENET1_TX_DATA5 | 114 |
| GPIO_001 | 115 | ENET1_TX_DATA6 | 115 |
| UART1_TX_DATA | 116 | ENET1_TX_DATA7 | 116 |
| UART1_CTS_B | 117 | ENET1_TX_DATA8 | 117 |
| UART1_RX | 118 | ENET1_TX_DATA9 | 118 |
| UART1_RX_B | 119 | ENET1_TX_DATA10 | 119 |
| UART1_CTS_A | 120 | ENET1_TX_DATA11 | 120 |
| UART1_RX_B | 121 | ENET1_TX_DATA12 | 121 |
| UART1_CTS_A | 122 | ENET1_TX_DATA13 | 122 |
| UART1_RX_B | 123 | ENET1_TX_DATA14 | 123 |
| UART1_CTS_A | 124 | ENET1_TX_DATA15 | 124 |
| UART1_RX_B | 125 | ENET1_TX_DATA16 | 125 |
| UART1_CTS_A | 126 | ENET1_TX_DATA17 | 126 |
| UART1_RX_B | 127 | ENET1_TX_DATA18 | 127 |
| UART1_CTS_A | 128 | ENET1_TX_DATA19 | 128 |
| UART1_RX_B | 129 | ENET1_TX_DATA20 | 129 |
| UART1_CTS_A | 130 | ENET1_TX_DATA21 | 130 |
| UART1_RX_B | 131 | ENET1_TX_DATA22 | 131 |
| UART1_CTS_A | 132 | ENET1_TX_DATA23 | 132 |
| UART1_RX_B | 133 | ENET1_TX_CLK | 133 |

B2B 核心板和邮票孔引出的功能引脚是一样的,只是管脚引出的位置有一定的调整。详细请看具体封装。

第一章 最小系统设计

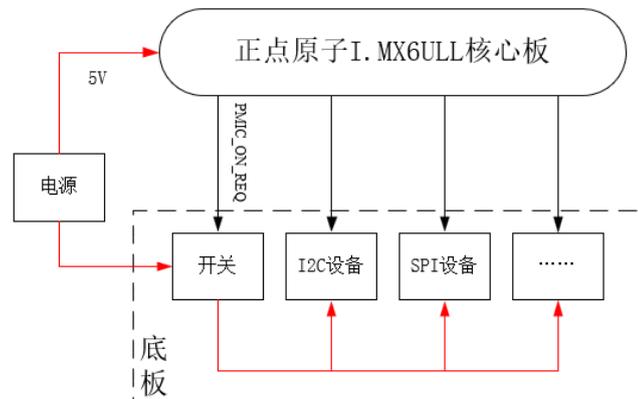
1.1 电源设计说明

核心板上电需要满足一定的上电时序要求才能正常启动。外设的供电需要晚于核心板。电源设计可以直接参考阿尔法开发板设计。



在阿尔法开发板上主电源从 12V 输入口输入, 经过 U16 输出 5V。5V 同时供给核心板以及 U17, U17 是底板外设的总电源来源并且被核心板通过 PMIC_ON_REQ 信号控制着输出。核心板上电完成后会将 PMIC_ON_REQ 信号拉高使能 U17 的输出, 给底板外设提供电源。(PMIC_ON_REQ 的输出能力较弱不能添加下拉电容, DNP 标识的电容是不焊的)

上电时序可以简化成下图:



1.2 系统启动配置

核心板的固件是存储在外置存储器上的例如 EMMC、NAND 或者 SD 卡。核心板想要正确的启动就需要有一个启动配置, 使得核心板可以在正确的存储器上读取到固件从而启动。设计时可以参考阿尔法开发板上的启动配置设计:

| SW | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-------------|
| D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | BOOT DEVICE |
| OFF | ON | OFF | OFF | OFF | OFF | OFF | OFF | USB |
| ON | OFF | OFF | OFF | OFF | OFF | ON | OFF | MicroSD |
| ON | OFF | ON | OFF | OFF | ON | ON | OFF | EMMC |
| ON | OFF | OFF | OFF | ON | OFF | OFF | ON | NAND |

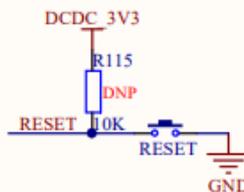
Title
*
Auth
lycre
Date
2019

使用八位拨码开关即可完成对四种模式配置的需求，这里使用的 3.3V 电源是底板外设总电源。

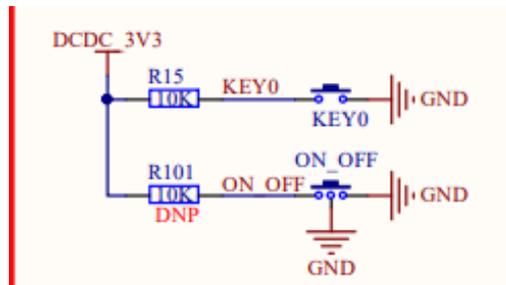
1.3 复位信号

核心板引出的 RESET 引脚是核心板的复位信号脚，此引脚在核心板内部有上拉，底板不需要再配置上拉电阻。阿尔法开发板上的设计：

RESET



核心板引出的 ON_OFF 引脚核心板的一个特殊功能引脚（可以实现类似电脑开关机的功能），此引脚在不使用的时候建议空置，不需要配置上下拉电阻。阿尔法开发板上的设计：



1.4 核心板固件烧写电路

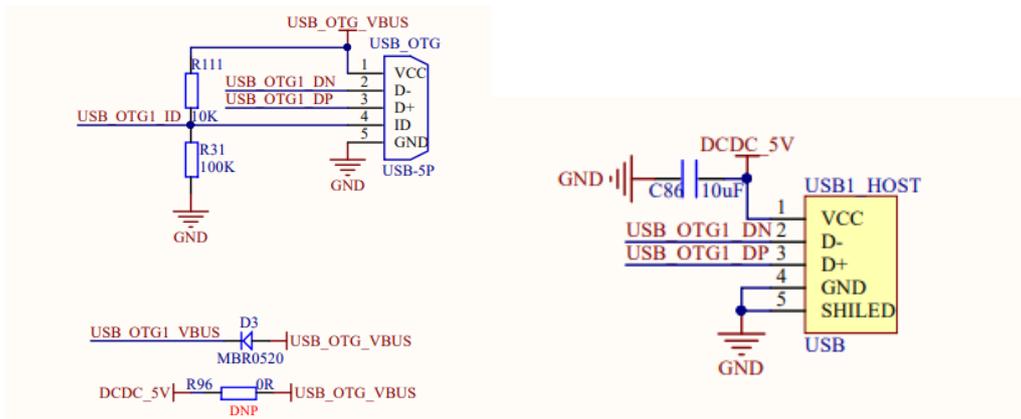
核心板烧写固件是通过 USB_OTG1 接口烧写的，不是使用 JTAG 接口，所以在设计底板的时候建议将 USB_OTG1 引出详细可以看 2.1USB 接口章节。调试串口是核心板的主要调试，同时在烧写固件的时候会有日志从调试串口输出，建议引出方便调试，引出详细可以看 2.3 调试串口章节。

第二章 外设接口注意事项

2.1 USB 接口

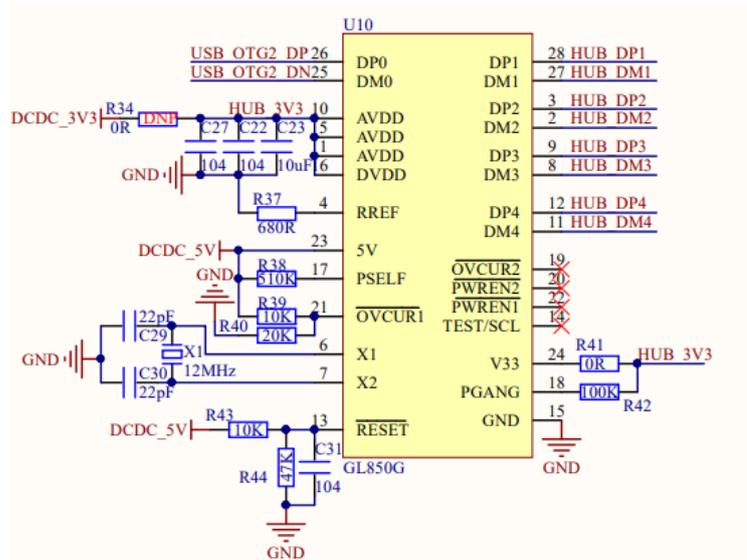
核心板上有集成两个 USB-PHY 均可以实现 OTG 功能（即可以做主机又可以做从机）。在阿尔法开发板上，将 USB_OTG1 做成 OTG 功能，烧写系统的时候需要使用这个接口。USB_OTG2 做成 HOST 功能连接 HUB 芯片可以扩展更多的 USB 接口。

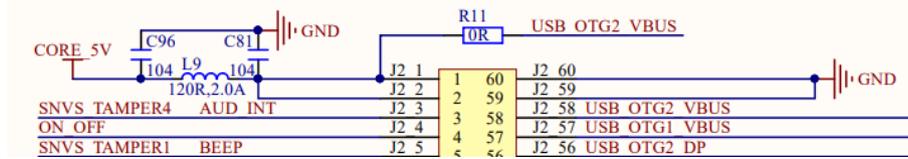
USB1:



阿尔法底板上 USB_OTG1 使用了 OTG 功能，使用 OTG 功能的时候需要增加一根 OTG-ID 脚，芯片上会根据这个 OTG-ID 引脚的状态去切换 USB_OTG1 做主机还是做从机。（ID 脚为高电平时为从机，为低电平时为主机。）USB_OTG1 是核心板的烧写接口在设计底板的时候建议引出。USB_OTG_VBUS 是接到核心板上的 USB_OTG1_VBUS 建议保持和开发板相同的设计。核心板烧录时此引脚为 USB-PHY 供电。

USB2:



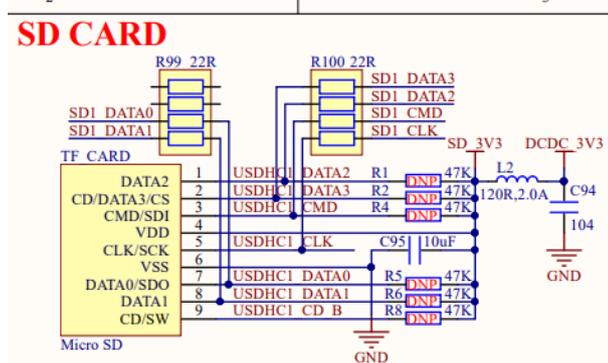


阿尔法底板上将 USB_OTG2 接到一个 HUB 芯片上拓展出 4 路 USB 接口可以外接其他设备,例如鼠标键盘等。将 USB_OTG2_VBUS 接到 5V 上,芯片的 USB-PHY 外设需要这路供电。

注意要点: USB 数据线是差分线,走线尽量满足 USB 走线的标准,避免由于走线导致的通信质量问题,同时由于 USB 外设需要的供电电流会偏大所以在设计的时候请考虑一下供电的走线宽度是否满足供电需求。核心板上有引出 USB_OTG1_VBUS 和 USB_OTG2_VBUS 两个引脚,这两个是芯片上 USB 部分的供电引脚,需要给 USB_OTG2_VBUS 上供给 5V。假如底板上没有供给 5V 给 USB_OTG2_VBUS 该引脚,可能会造成 USB 通信不正常。

2.2 SDIO 接口

核心板有引出一路 SDIO 接口供我们使用,默认阿尔法上是连接到 SD 卡槽,用于连接 SD 卡。可以用于调试阶段使用 SD 卡启动,或者量产阶段时使用 SD 卡进行烧录核心板。下图是阿尔法开发板上的 SD 卡槽设计:

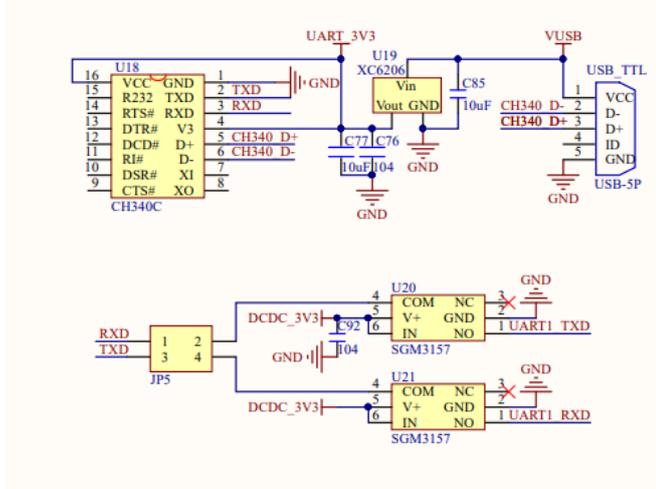


注意要点: SDIO 传输数据是并行传输的,在布线的时候建议尽量满足 SDIO 接口的布线需求。当外接一些耗电较大的外设时(如 SDIO-WIFI)电源走线需要满足外设的需求。图中的 USDHCI_CD_B 是检测 TF 卡热插拔的引脚在使用热插拔功能时需要连接上。有些 TF 卡槽的固定脚是需要连到地才能使卡槽正常的工作,在画 PCB 时可以确认一下卡槽的固定脚是否连接到地。

2.3 调试串口

核心板默认使用的串口的 UART1,在调试的时候建议将 UART1 引出。调试串口在阿尔法开发板上的设计:

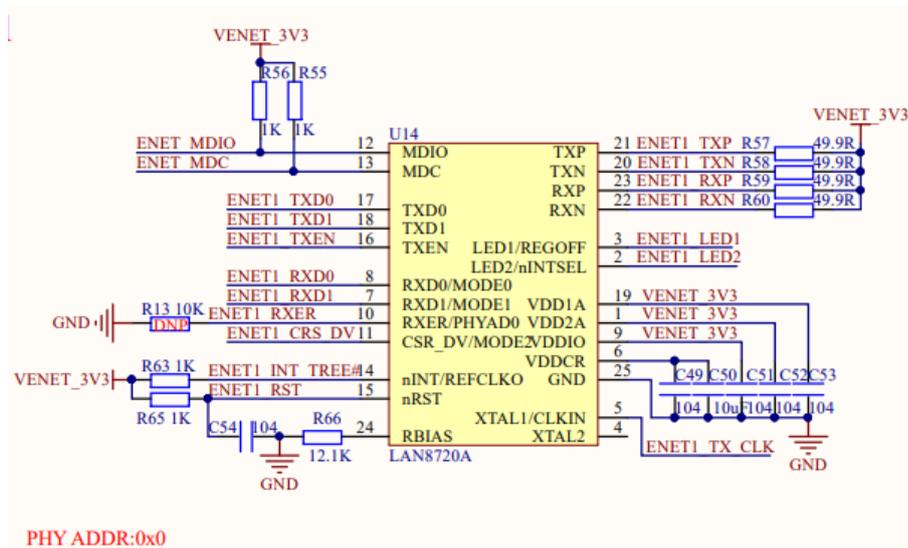
USB USART&USB POWER

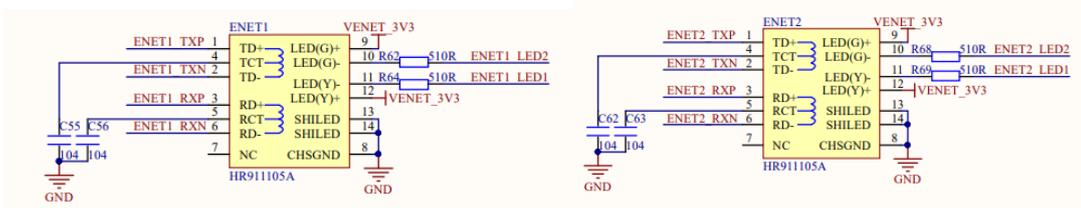
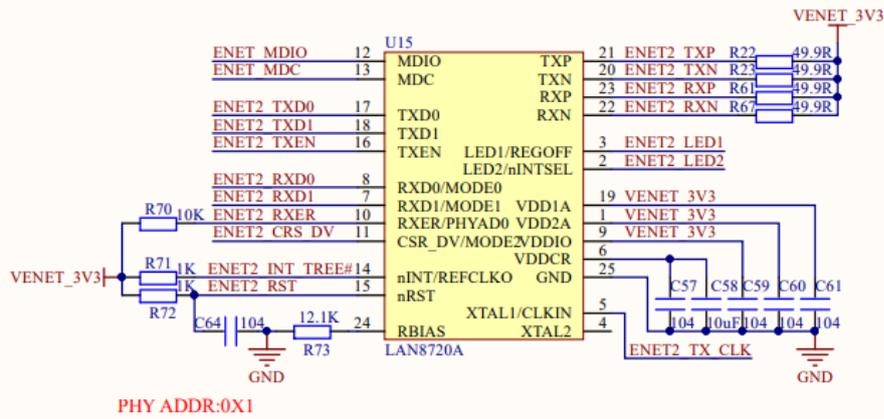


阿尔法开发板上将 CH340 和调试串口连接方便调试，且在两者之间增加了模拟开关用于隔离。模拟开关的作用是为了避免 CH340 通过调试串口的引脚干扰到核心板的启动。自己制作底板的时候可以根据自己的实际情况来设计。

2.4 网口

核心板上集成了两路的 100M 速率的 MAC 接口，需要和底板上的 PHY 芯片以及网络接口（HR911105A），三者才能形成完整的网口。下图是阿尔法开发板上的网络部分设计：





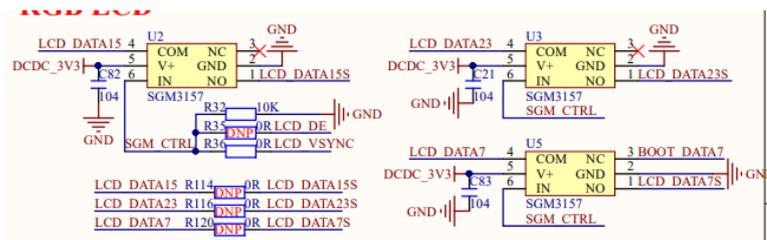
注意要点: PHY 芯片和核心板之间的布线不能过长，尽量将 PHY 芯片靠近核心板。由于核心板会通过一根引脚给 PHY 芯片提供 50MHz 时钟，该引脚的布线过长的话会导致信号质量不佳，PHY 不能正常识别。核心板通过 MDIO 和 MDC 两个引脚对 PHY 芯片进行管理和通信，数据的传输是通过 TX 和 RX 相关引脚。PHY 芯片和网络接口的数据线需要走差分线。走线最好满足一下相关的走线标准需求。

第三章 核心板上的部分管脚

3.1 RGB-LCD 管脚

LCD_DATA 管脚既是连接 LCD 屏幕的数据管脚又是启动时的启动配置引脚。在不使用 LCD 屏幕的时候可以将这部分引脚复用成其他功能，不过需要注意一下，连接的外设在上电的时候会不会对这部分引脚置高电平或者低电平，要避免在启动的时候干扰到核心板读取到的电平状态，假如没有读取到正确的电平会导致核心板不能正确的从外部存储上加加载 Linux 固件并启动。

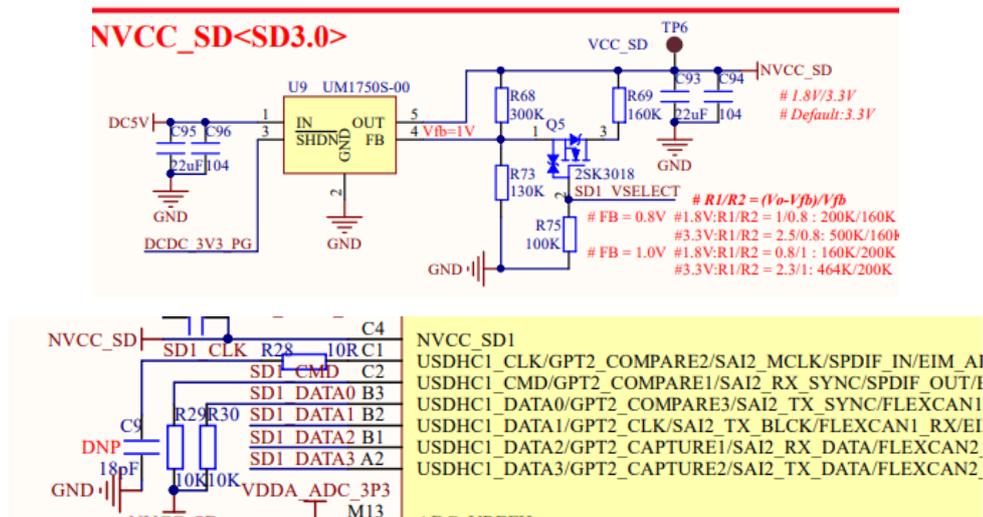
示例：在阿尔法开发板上，由于我们正点原子的屏幕使用了三个 LCD_DATA 引脚作为屏幕 ID 脚，在 ID 脚上有焊接上下拉电阻，由于焊接了上下拉电阻会影响到开发板的正常启动，故阿尔法开发板在 LCD 接口上添加了三个模拟开关进行隔离。如下图所示：



假如不使用我们正点原子的屏幕, 可以根据自己实际使用的屏幕或者转接芯片的具体情况来确定是否会干扰到核心板的正常启动。假设在不会影响的情况下可以直接直连。

3.2 GPIO5 (GPIO1-05)

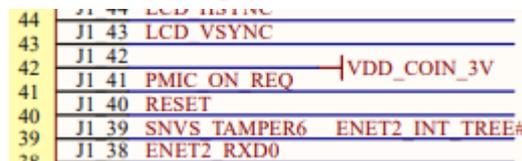
核心板上引出的 GPIO5 在使用的时候需要注意一下。这个引脚在核心板上被用于切换 SDIO1 接口的电压。当 GPIO5 为高电平时会把 SDIO1 接口的逻辑高电平设置成 1.8V, 为低电平时会把 SDIO1 接口的逻辑高电平设置成 3.3V。当需要使用 SDIO1 接口的管脚时, 请注意 GPIO5 的引脚电平应保持不变, 为了避免电压变化导致外设和核心板的逻辑电平不兼容, 从而导致外设或者核心板工作异常。当要使用 GPIO5 和 SDIO1 接口时, 请注意和 SDIO1 接口相接的外设是否兼容两种电平。不然还是建议将 GPIO5 进行空置, 只使用 SDIO1 接口。



3.3 RTC 部分

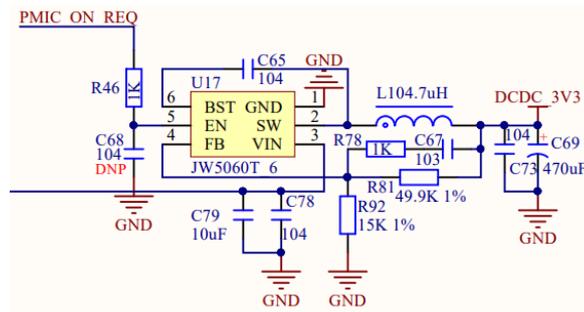
核心板上的 VDD_COIN_3V 引脚是用纽扣电池给芯片上集成的 RTC 部分供电的, 因为集成的 RTC 功耗较高所以比较推荐使用外部的 RTC 模块。当不使用芯片上自带的 RTC 模块时, 不建议将这个引脚直连底板的 3.3V 供电, 直接空置即可核心板上会有一路电源给这部分供电, 不需要底板再提供了。假如连接了底板的 3.3V 可能会引起核心板的启动异常或者供电异常。

核心板上集成的 RTC 模块, 在待机的时候消耗纽扣电池的电量会比较快, 根据自己的项目需求可以选择使用外部的 RTC 模块。(例如 pcf8563)



3.4 PMIC_ON_REQ 管脚

这个管脚是核心板使能底板 3.3V 供电的信号脚。当核心板上电后正常就会将这个管脚拉高使能底板的 3.3V 供电并进一步启动。这个引脚的负载能力很弱, 假如底板的 DCDC 转换芯片的使能脚需要较大的驱动能力请自行增加驱动电路。这个引脚在使用的時候请不要加上下拉电容, 容易造成核心板不能正常启动。



阿尔法开发板上与 PMIC_ON_REQ 相连的下拉电容是没有焊接上去的，焊上去会造成启动异常。

在使用三极管作为驱动电路的时候要注意三极管的驱动电流，当三极管的吸收过多的电流时会把 PMIC_ON_REQ 引脚的电压拉下去，这时需要注意一下是否会影响到核心板的电压正常。

第四章 底板设计后样板的一些问题及调试思路

4.1 核心板启动不了

- 1) 核心板直接供电 5V 是不会正常启动的，核心板需要底板有相应的启动电路配合才能启动。
- 2) 有了启动电路的话还需要满足对应的上电时序。核心板先接受 5V 的供电，核心板稳定后会 PMIC_ON_REQ 拉高，通过 PMIC_ON_REQ 使能底板的 3.3V 输出，3.3V 把启动的配置引脚拉高。核心板读取配置引脚进而启动对应存储上的固件。
- 3) 核心板对于 5V 和 3.3V 的是有一定要求的，部分的 DC-DC 电压转化芯片被使能输出后，需要经过一小段的上升时间才能稳定输出 5V。假如底板上的 3.3V 是经 5V 供给的 DCDC 芯片转化的，可能会因为核心板提前上电，提前输出 PMIC_ON_REQ 信号使能 3.3V 的输出，这时可能供电不稳导致底板外设使用的 3.3V 并未正确输出 3.3V（一般是低于 3.3V），有可能导致核心板在读取启动配置引脚时读到错误的配置从而未能启动到对应的储存上的固件。5V 和 3.3V 的上升时间在有条件的情况下尽量短一些，底板外设使用的 3.3V 最好在使能 DCDC 芯片输出后马上输出 3.3V。PMIC_ON_REQ 拉高后就会去读取拨码开关的配置信息。这个具体的时间 NXP 没有公开出来，笔者测试 没有多余的时间，基本是 PMIC_ON_REQ 拉高后就接着读取启动配置。没读到或者读到错误的配置就会不启动或者陷入假死。
- 4) 核心板进入了关机状态，这个只能测量核心板上的电压进行判断，正常核心板工作的话 ON_OFF 引脚和 PMIC_ON_REQ 引脚会输出高电平，关机状态 ON_OFF 引脚保持高电平 PMIC_ON_REQ 进入低电平。这时退出关机状态需要将 ON_OFF 引脚短时间接下地，待 PMIC_ON_REQ 恢复高电平即退出关机状态。
- 5) 底板外设上电先于核心板，造成电流倒灌导致不能正常启动。底板外设上电是需要晚于核心板上电，不然电流会经外设和核心板连接的 io 口倒灌核心板，从而影响核心板的正常启动。

4.2 外设接口的一些问题

- 1) USB 设备接上一直提示有新设备接入或者没有反应。需要检查一下核心板的 USB_OTG_VBUS 供电是否正常,USB 接口处的供电是否正常(USB 设备需要从母口取电),USB 数据线走线是否符合走线标准。
- 2) TF 卡没有反应或者初始化失败。有一些 TF 卡座需要将外壳接地才能使 TF 卡正常工作。数据传输不顺可以检查供电和数据线走线是不是符合 SDIO 的走线标准。
- 3) 网口识别正常但数据传输有问题或者识别不正常。PHY 芯片和核心板不能距离太远条件可以请尽量将 PHY 芯片和核心板靠近,避免走线过长导致的通信质量下降。走线尽量满足其需要的标准。
- 4) 使用 USB 连接电脑烧录, MFGTOOL 不识别, 检查 USB 数据线走线是否满足走线标准

4.3 使用 MFGTOOL 工具烧录固件的一些问题

- 1) MFGTOOL 工具烧录是通过 USB1 接口进行通信的, 要确保 USB1 的通信正常。
- 2) 使用 MFGTOOL 工具烧录需要将拨码开关设置成 USB 模式, 且 SD 卡接口不能接有可以启动的 SD 卡, 不然会启动 SD 卡上的固件。
- 3) MFGTOOL 工具识别不到核心板, 但是 USB1 接口是可以正常使用, 拨码开关设置正常, 也没有接入 SD 卡, 但是这时, MFGTOOL 工具就是识别不到核心板。可以检查一下核心板的 UART1 和 UART2 的引脚是不是连接到单片机模块或者外设上。由于外设或者模块的上电快于核心板导致连接的引脚处于通信状态, 干扰了核心板的正常启动, 如有的话可以暂时断开外设和模块的供电测试一下是否是干扰源。

第五章 特殊需求

5.1 底板有对静电、信号隔离之类的要求请自行增加对应的处理避免损坏核心板。

5.2 自行设计供电电路需要注意的几个要求

- 1) 核心板的供电输入电压是 5V 的, IO 管脚的耐压是 3.3V 的。外接外设时请注意接口电平是否满足耐压值。
- 2) 选用其他 DCDC 芯片的时候有几个建议需要留意一下。DCDC 芯片的输出电压上升不能慢, 需要快速一些, 最好可以保持和阿尔法开发板的一致。因为过慢的上升会导致核心板上电完成了, 通过 PMIC_ON_REQ 使能底板的 3.3V 输出, 但是这时输出 3.3V 的 DCDC 芯片由于输入的电压上升过慢, 输出的电压也会有一个上升过程, 从而影响到核心板读取启动配置。选用 DCDC 芯片尽量选择输出电压上升快的。

第六章 部分案例

- 1) 客户 A 比较信赖 TI 的电源芯片, 在样板上使用了 TI 的 DCDC 电源芯片作为核心板的供电源, 样板制作完后进行测试发现不能启动, 和客户交流探讨后, 发现给核心板供电 5V 的电源芯片以及核心板控制使能的 3.3V 电源芯片均存在较长时间的软启动时间, 经示波器实际

测量发现电压上升过程花费时间较长(有 5MS 甚至更长时间), 导致核心板不能正确读取拨码开关的设置造成启动错误, 更换芯片软启动时间参数和开发板上的 JW5060T (1.6MS) 相近的型号后, 样板正常启动。

- 2) 客户 B 制作样板后先贴保证最小系统的器件初步上电启动正常, 贴上外设后不能正常启动。和客户交流一番发现, 底板上有一个外设模块是使用 5V 供电的。断开这个外设模块的供电后, 启动正常。这个外设模块和核心板使用同一路 5V 导致外设模块先于核心板上电完成, 外设模块上电完成后就直接通过数据管脚发送数据。核心板所需上电时间要长于外设模块, 外设模块先上电完成后会直接通过数据管脚反馈数据信号给核心板, 此时核心板会被这些信号给干扰从而导致不能正常启动。
- 3) 客户 C 参照阿尔法板的原理图制作自己的底板, 发现不能正常启动, 经过和客户沟通发现阿尔法板原理图上的标注“DNP”的元件他也焊接上去了, 把需要去掉的元件去掉后正常启动。
- 4) 客户 D 参照阿尔法原理图制作自己的底板, 打样后使用 EMMC 启动正常, 使用 SD 卡启动并不能正常启动, 检查实物及原理图, 发现 SD 卡槽的固定脚并未接地以及 SDIO 接口上并未串接小电阻。给 SD 卡槽跳线接地及 SDIO 接口串接小电阻后正常。
- 5) 客户 E 参照阿尔法原理图制作自己的底板, 并使用我们正点原子的屏幕, 一开始使用 7 寸屏正常, 后面更改成 4.3-800*480 或者 10 寸屏发现不能正常启动, 换回 7 寸屏就可以。经排查发现底板上和阿尔法一样使用了模拟开关, 但是阿尔法上标记“DNP”的电阻也焊上去了, LCDDATA7、15、23 这三根数据线直连了, 导致模拟开关并没有起作用。正点原子的屏幕在这个三根数据线上有对应的上下拉电阻用于做屏幕的 ID 标识。恰好 4.3-800*480 和 10 寸屏在 LCDDATA7 这根数据线上有一个上拉电阻, 把直连电阻焊接上时, 再接上屏幕就会一直把 LCDDATA7 拉高, 这时就会影响到核心板的启动。解决办法: 把直连电阻去掉让模拟开关可以正常起作用, 或者去掉屏幕的识别 ID 电阻再手动加载内核和设备树启动。
- 6) 客户 F 自己的产品有需求把网口和 PHY 芯片放在距离核心板较远的地方。样板测试出现网络不通, 接网线没反应, 初始化网口失败。由于芯片的驱动能力较弱, 建议把 PHY 芯片放置在核心板附近, 网口要放置在距离较远的地方可以使用网口的延长线。
- 7) 客户 G 自己的产品上有多路的电源芯片需要使用 PMIC_ON_REQ 信号进行控制, 于是使用三极管作为驱动电路, 样板设计完成之后出现启动异常, 对核心板上的测试点电压进行测量发现电压异常, 客户在 PMIC_ON_REQ 引脚和三极管中间串接了一个电阻, 查阅三极管的数据手册疑似三极管吸收电流过大导致 PMIC_ON_REQ 引脚被拉下去, 客户把中间串接的电阻更换成阻值更大的之后问题得到改善。根本原因在三极管吸流过大, 在设计的时候可以预留一些设计, 方便出现相同问题时方便调试。
- 8) 更多案例后续持续收集更新