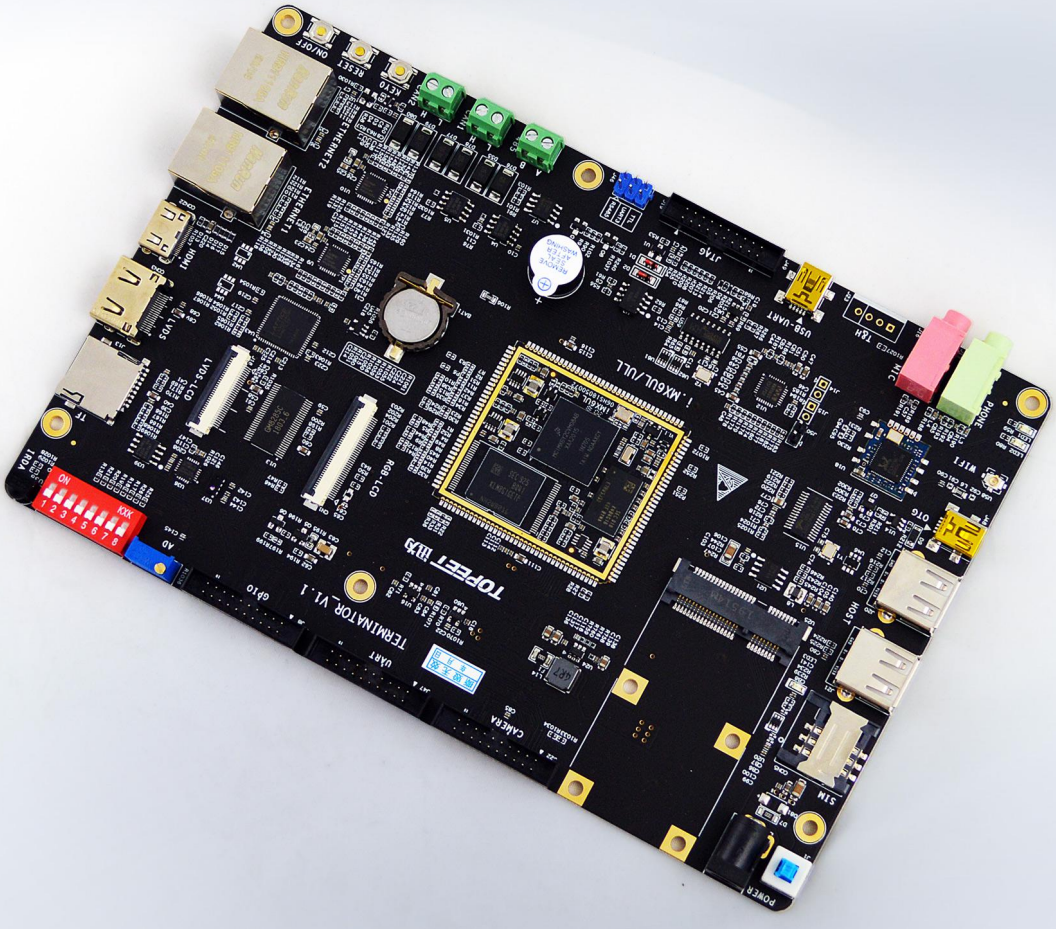
TOPEET北京迅为

让学习更简单 开发更轻松

1800+页使用手册 | 手把手视频教程 | 7x24小时稳定运行



迅为-iMX6ULL规格书

实战视频 | 资料教程 | 免费更新

选配
屏蔽罩版本

官方微信公众号

迅为商城

Beijing TOPEET Electronics Co.,Ltd.

公司简介

北京迅为电子有限公司位于北京海淀区中关村科技园，是国家认定的中关村高新技术企业，领先的嵌入式平台技术提供商。公司专注于嵌入式系统开发平台、新一代智能终端等产品的研发与生产。利用物联网、云计算等前沿技术，结合数据安全、人工智能等成熟应用，不断开拓创新，锐意进取，助力并加速智能终端、智能物联网、嵌入式边缘计算等领域的产品化与技术创新。

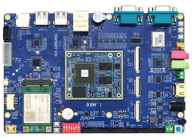
迅为扎根于嵌入式 Linux、Android 以及 ARM 系统架构层面的研发，多年的一线产品实战积累以及多领域开发经验，形成了从硬件原型设计、layout、硬件驱动，操作系统移植、中间件到上层应用全面的技术体系。核心技术涵盖高速数字电路设计、底层驱动、通信协议栈、操作系统优化、安全产品设计等方面。

凭借多年的行业实施经验，公司在相关产业链方面不断优化积累，形成了从元器件采购、加工制造等生产交付领域全方位优势，并与行业领先企业深入广泛合作。让合作伙伴把精力集中在自身产品的设计过程中，以提高开发设计效率，推进产品尽快上市并为之保驾护航。

作为一家技术驱动型企业，公司配备创新实验室，在电磁兼容、产品认证、复杂应用环境等领域具备很强的整改经验。可为用户在相关领域提供服务，解决 EMI 和 EMS 方面的棘手问题。

公司产品遍布全国，目前已成为国内嵌入式平台产品首选厂家。通过不断提升服务水平，赢得了客户普遍赞誉。新时期我们将进一步扎实精进，努力为中国嵌入式技术的发展做出自己的贡献。

产品系列


开发板

核心板

触控一体机

工控机

平台品牌

三星

恩智浦

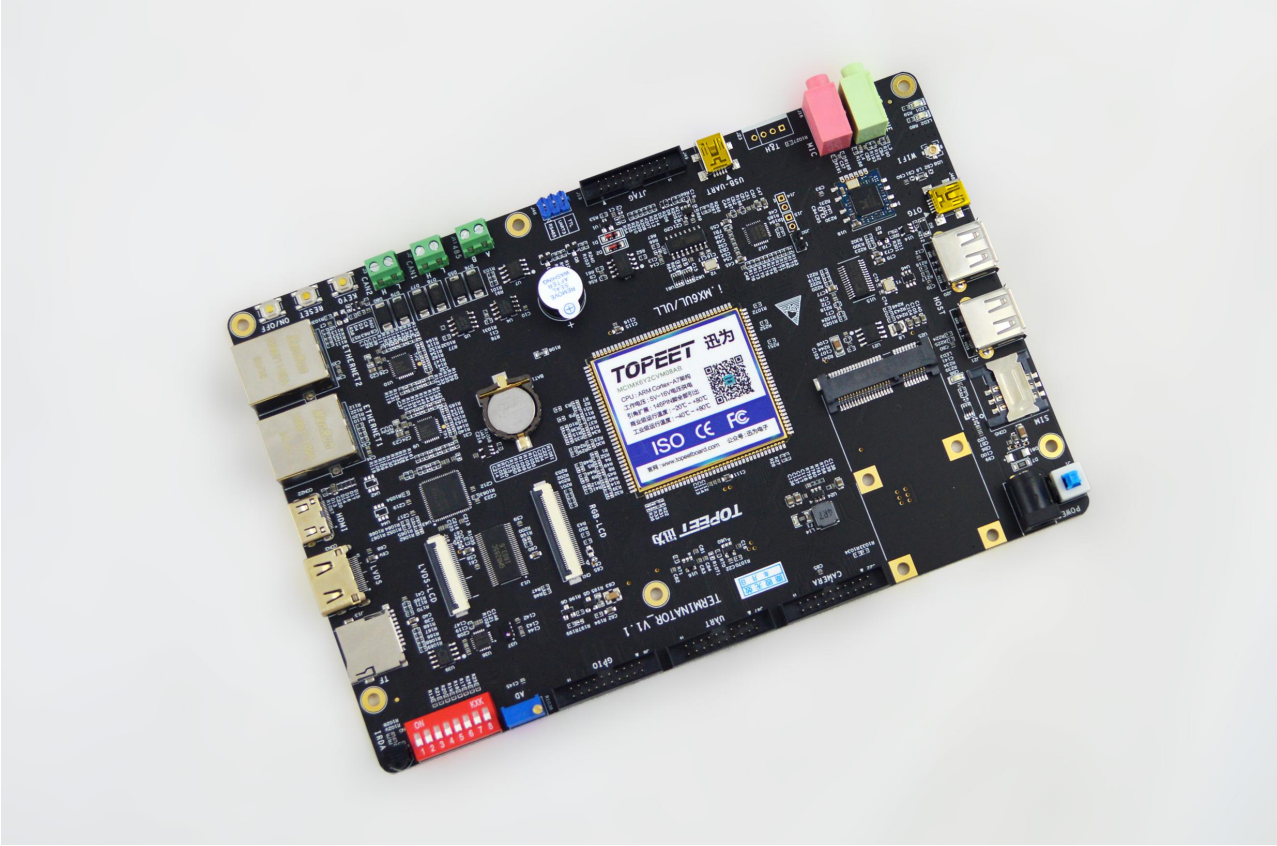
瑞芯微

龙芯

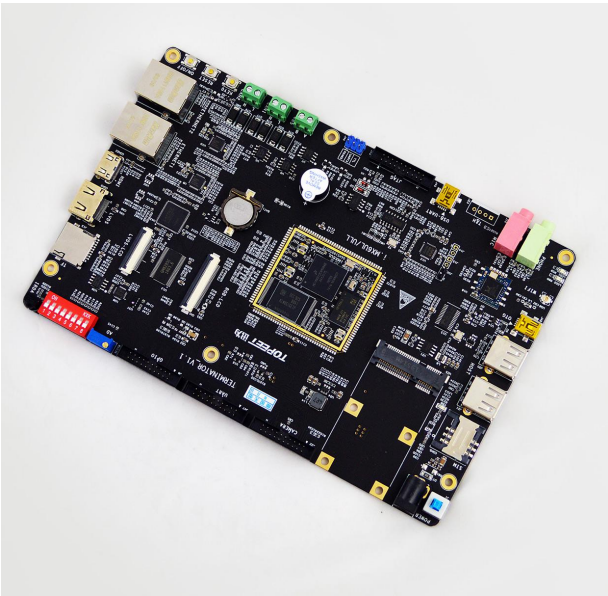
目录

公司简介 2
产品套餐 1
产品概述 2
可选模块 1
可选屏幕 1
核心板介绍 2
核心板认证 3
核心板硬件参数 4
底板介绍 5
底板硬件参数 6
详细尺寸 14
i.MX6ULL 终结者原理分析 15
软件资源 41
资料介绍 44
资料获取与后续更新 54

产品套餐



核心板可配屏蔽罩（默认不包含，如有需要联系客服）



基本型（不含 4G、温湿度模块）

豪华型（含 4G、温湿度模块）

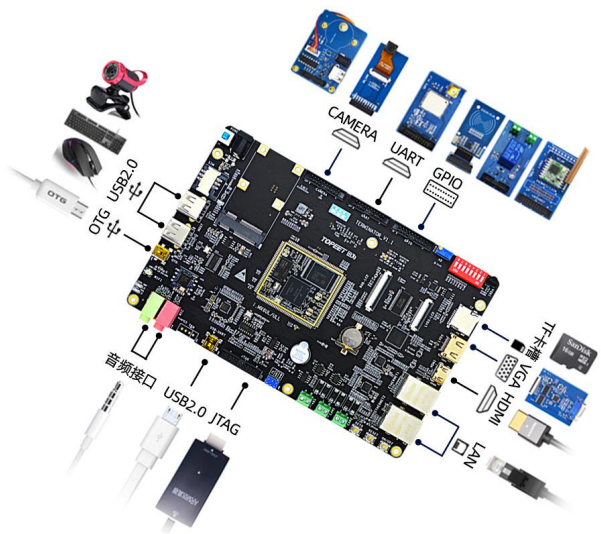
产品概述

1. 开发板预留了 JTAG 仿真接口，并给出了开发文档，可以实现在 JLINK 仿真器条件下的单步跟踪、断点调试等功能，使得开发研究 i.MX6ULL 处理器的相关技术更加直观便利。

2. 配套不同规格大小的屏幕供大家选择。迅为 7 寸 LVDS 屏、9.7 寸屏、10.1 寸屏 创造性的采用 HDMI 线连接方式，牢固耐用，使用方便，并兼容传统的连接方式。

3. 板载扩展了 HDMI 接口功能，您只要有一台 HDMI 接口的电脑显示器，或者电视投影等设备，配上鼠标，就可以直接驱动并使用这些显示设备了，这样看起来俨然是一台小电脑了。

4. 物联网时代，各种传感器的采集和处理技术是需要我们掌握的，开发板标配了各种传感器设备，通过对更多模块的研究和集成，可以更深入的理解和开发物联网设备。



可选模块

RFID模块

VGA模块

500万摄像头

USB摄像头

LORA模块

继电器

步进电机

GPS模块

可选屏幕

4.3寸屏

5寸屏

7寸屏-RGB

7寸屏-LVDS

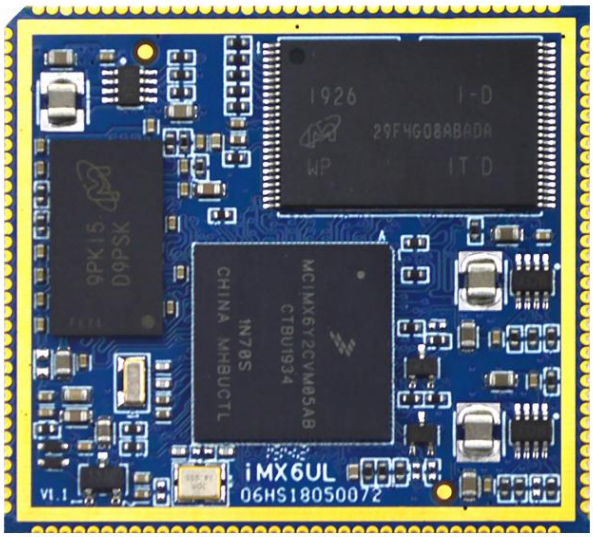
9.7寸屏

10.1寸屏

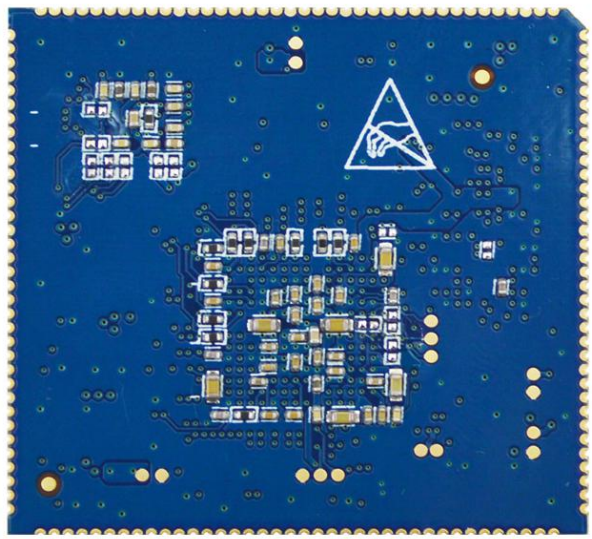
显示屏	4.3 寸屏	5 寸屏	7 寸屏-RGB	7 寸屏-LVDS	9.7 寸屏	10.1 寸屏
分辨率	480*272	800*480	1024*600	1280*800	1024*768	1024*600
触摸屏	电阻屏	电容屏-自选 电阻屏-自选	电容屏	电容屏	电容屏	电容屏
信号类型	RGB	RGB	RGB	LVDS	LVDS	LVDS
尺寸(mm)	110*83*6	140*83*8	190*110*11	197*119*10	243*189*10.5	250*155*9.5

核心板介绍

i. MX6ULL 工业级核心板板载资源丰富，通过邮票孔将 i.MX6ULL 的所有资源几乎全部引出，接口丰富，可以满足各种应用的需求。核心板的尺寸仅为 42mm*38mm，非常小巧。



（正面）



（背面）

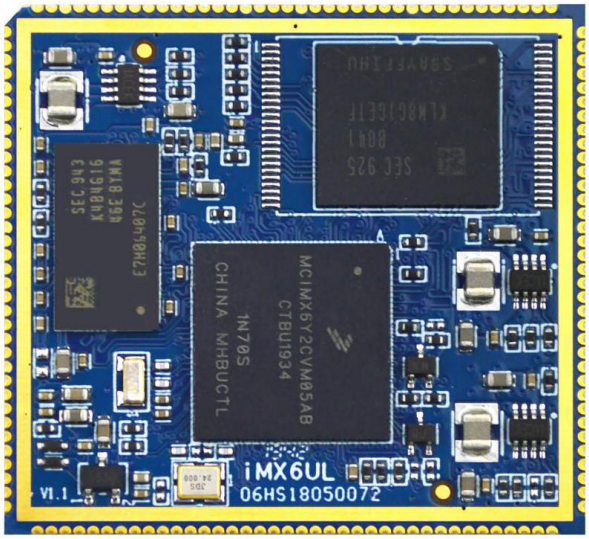
I.MX6ULL 工业级核心板板载资源：

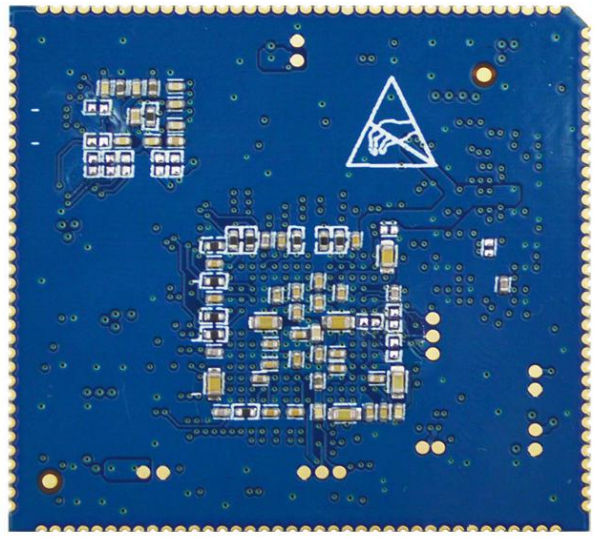
CPU：MCIMX6Y2CVM05AB(工业级)或 MCIMX6Y2CVM08AB（工业级），主频分别为 528MHz 和 800MHz(实际为 792MHz)，采用 BGA289 封装

DDR3：MT41K256M8，大小为 256MB 字节，工业级

NAND FLASH：MT29F4G08ABADAWP-IT，512M 字节，工业级

引脚扩展：采用 1.0mm 间距的邮票孔方式引出 146 个引脚





I.MX6ULL 商业级核心板板载资源：

CPU：MCIMX6Y2CVM05AB(工业级)或 MCIMX6Y2CVM08AB（工业级），主频分别为 528MHz 和 800MHz(实际为 792MHz)，采用 BGA289 封装

DDR3：K4B4G1646E-BY(M)，大小为 512MB 字节，商业级

EMMC：KLM8G1GET，8GB 字节，商业级

引脚扩展：采用 1.0mm 间距的邮票孔方式引出 146 个引脚

核心板认证

一次性通过 CE、FCC 认证

VERIFICATION-ПРОВЕРКА-檢証-VÉRIFICATION-검증-التحقق

Verification of Conformity

Certificate No.	:	CTL1805082021-EC
Applicant	:	Beijing TOPEET Electronics Co.,Ltd.
Address	:	409, New Material Plaza A, NO.7 MiddleFenghui Road, Haidian District, Beijing 100094, China
Product	:	Topeet ARM Mainboard
Trademark	:	TOPEET
Model(s)	:	iTOP-6UL
Manufacturer	:	Beijing TOPEET Electronics Co.,Ltd.
Address	:	409, New Material Plaza A, NO.7 MiddleFenghui Road, Haidian District, Beijing 100094, China
Test Report	:	CTL1805082021-E

Complies with the requirements of the
EC EMC Directive 2014/30/EU
Test Standards:

EN 55032: 2015
EN 55035:2017
EN 61000-3-2: 2014
EN 61000-3-3: 2013

Remarks:
Based on the voluntary assessment of the product sample and technical file, we confirm that the above-mentioned product meets the requirements of the EC directive.
The CE mark as show below can be used, under the responsibility of the manufacturer or the importer, after completion of an EC declaration of conformity and compliance with all relevant EC directives.

CE

For Chief Executive
May 15, 2018

Shenzhen CTL Testing Technology Co., Ltd.
Add:Floor 1-A,Baisha Technology Park,No.3011,Shahexi Road,Nanshan District,Shenzhen,China 518055
Tel:(86)0755-89486194 Web:www.ctl-lab.com E-mail:ctl@ctl-lab.com

VERIFICATION-ПРОВЕРКА-檢証-VÉRIFICATION-검증-التحقق

Supplier's Declaration of Conformity

Certificate No.	:	CTL1805082021-FC
Applicant	:	Beijing TOPEET Electronics Co.,Ltd.
Address	:	409, New Material Plaza A, NO.7 MiddleFenghui Road, Haidian District, Beijing 100094, China
Product	:	Topeet ARM Mainboard
Trademark	:	TOPEET
Model(s)	:	iTOP-6UL
Manufacturer	:	Beijing TOPEET Electronics Co.,Ltd.
Address	:	409, New Material Plaza A, NO.7 MiddleFenghui Road, Haidian District, Beijing 100094, China
Test Report	:	CTL1805082021-F

Test Standards:

FCC PART 15 Subpart B Class B

Remarks:
This device is complies with the part 15 of FCC rules. It's confirmed and found to comply with the requirements setup by ANSI C63.4 & FCC part 15 regulation for the evaluation of electromagnetic compatibility.

FC

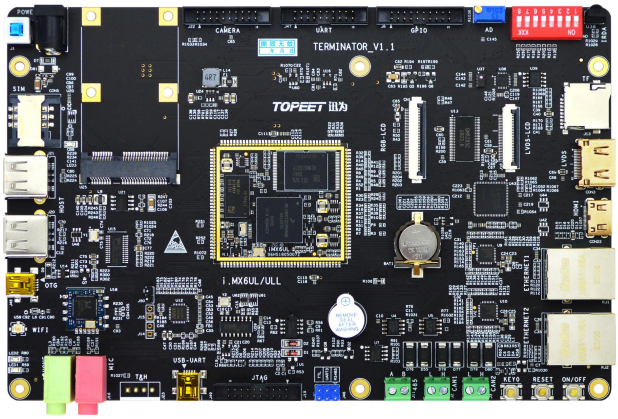
For Chief Executive
May 15, 2018

Shenzhen CTL Testing Technology Co., Ltd.
Add:Floor 1-A,Baisha Technology Park,No.3011,Shahexi Road,Nanshan District,Shenzhen,China 518055
Tel:(86)0755-89486194 Web:www.ctl-lab.com E-mail:ctl@ctl-lab.com

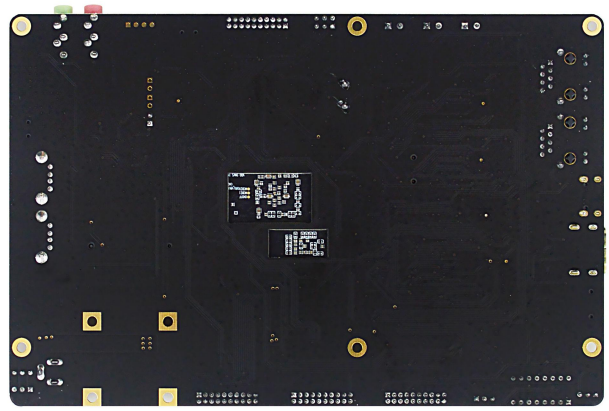
核心板硬件参数

核心板硬件参数	
尺寸	38mm*42mm
CPU	主频 528MHz/800MHz　ARM Cortex-A7 架构 单核
iMX6ULL 商业级	内存：512M　存储：8G EMMC
iMX6ULL 工业级	内存：256M　存储：512M FLASH
工作电压	5V~16V 电压供电
系统支持	Linux-QT 系统；Ubuntu core 系统；Youcto 系统；Buildroot 系统；
商业级运行温度	-20℃ ~ +80℃
工业级运行温度	-40℃ ~ +80℃
引脚间距	1.27mm
引角扩展	i.MX6 功能全部引出，146PIN 脚
连接方式	邮票孔
PCB	6 层沉金工艺

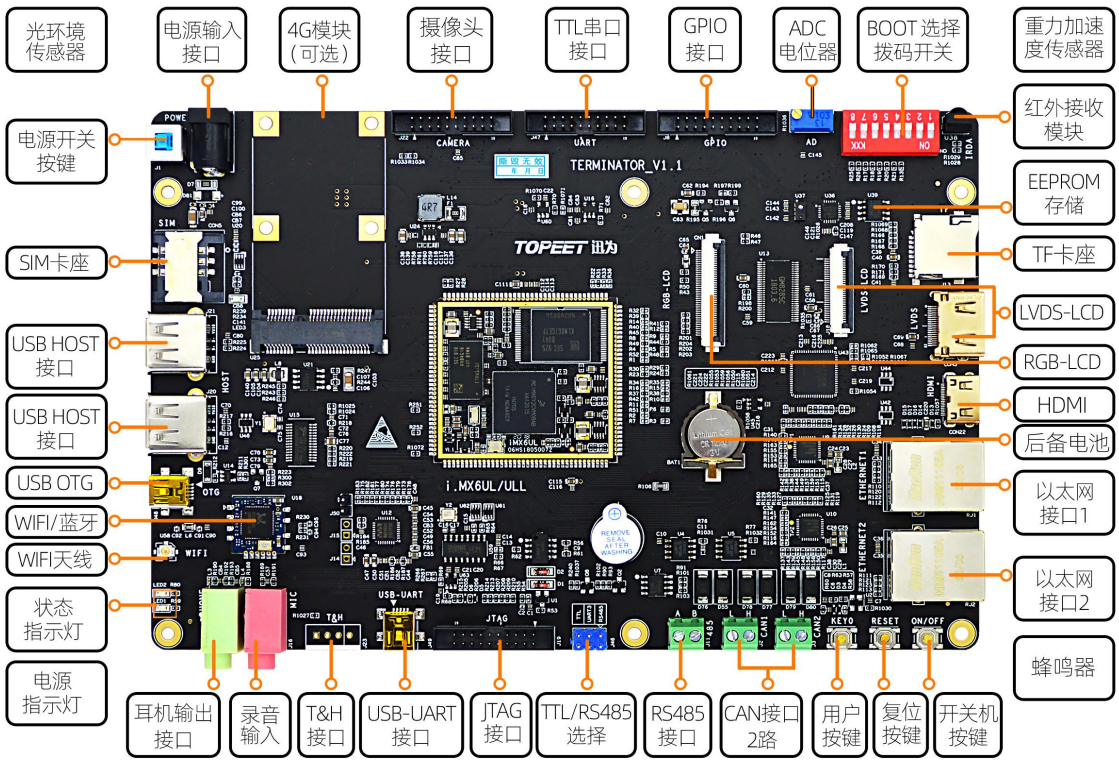
底板介绍


TERMINATOR_V1.1
TOPEET 迅为
i.MX6UL/ULL

（正面图）



（背面图）


光环境传感器
电源输入接口
4G模块（可选）
摄像头接口
TTL串口接口
GPIO接口
ADC电位器
BOOT选择拨码开关
重力加速度传感器
电源开关按键
红外接收模块
EEPROM存储
SIM卡座
TF卡座
LVDS-LCD
USB HOST接口
RGB-LCD
USB HOST接口
HDMI
后备电池
USB OTG
以太网接口1
WIFI/蓝牙
WIFI天线
以太网接口2
状态指示灯
蜂鸣器
电源指示灯
耳机输出接口
录音输入
T&H接口
USB-UART接口
JTAG接口
TTL/RS485选择
RS485接口
CAN接口2路
用户按键
复位按键
开关机按键
TERMINATOR_V1.1
TOPEET 迅为
i.MX6UL/ULL

底板硬件参数

底板硬件参数	
尺寸	12*19cm
POWER	5V~16V 电压供电
SWITCH	电源开关
EEPROM 存储	256 字节
4G 模块	1 个 全网通 （选配）
WIFI 蓝牙模块	支持
LVDS 接口	1 路
RGB 接口	1 路
CAN	2 路 CAN 总线接口
RS485	1 路 RS485 总线接口
MIC	支持录音输入
PHONE	支持耳机输出
USB OTG	1 路
USB HOST	2 路
USB-UART	1 路
mini HDMI	1 个
网口	2 路百兆以太网，RJ45 接口

TF 卡槽	1 个标准 TF 卡接口
SIM 卡槽	1 个标准 SIM 卡接口
T&H 接口	温湿度传感器接口
TTL/RS485 接口	TTL/RS485 接口二选一
BUZZER	1 个蜂鸣器
IRDA	1 个红外一体化接收头
ADC 电位器	支持
DIP SWITCH	8 位拨码开关
JTAG	1 个
CAMERA 接口	1 个 支持 500 万摄像头
TTL 串口接口	20PIN （gpio 引出：两个 ad，一个 spi，一路 ttl。一路 i2c，5 个 gpio，5v，3.3v 电源）
GPIO 接口	20PIN
重力加速度传感器	支持
红外接收模块	支持
光环境传感器	支持
RTC	实时时钟
LED	2 个
按键	3 个

i.MX6ULL 终结者底板资源说明

I.MX6ULL 终结者开发板板载资源非常丰富，几乎已经把 i.MX6ULL 的功能发挥到了极致。下面我们就来具体分析下 i.MX6ULL 终结者的硬件资源。我们从电源接口开始按照顺时针的方向依次介绍。

首先我们来看下底板的资源说明：

1. 电源 DC 输入接口

这是底板板载的一个电源输入接口（JACK1），采用了宽电压输入的电源芯片（JW5060T），外部可以输入 5V~16V 范围内的直流电源，通过电源芯片（JW5060T）最后会输出 5V 的直流电源给核心板和底板供电。由于输入的电源支持宽电压（5V~16V），所以我们可以很方便的找到一款电源适配器，来给开发板供电。

2. Mini PCIE 4G 模块接口

开发板板载了一个 Mini PCIE 的 4G 模块接口（U25），实际上是通过 USB 协议进行数据传输的，适配市面上大多数的 Mini PCIE 4G 模块，迅为电子使用的是移远公司的 EC20（支持全网通的 4G 模块），对于一些没有宽带网络的环境，我们可以通过 4G 模块来联网。

3. CAMERA 摄像头接口

开发板板载一路摄像头接口（可以接 CSI 的摄像头）（J22），通过 20pin 的座子引出，可以连接迅为电子的 500w 摄像头模块。

4. 串口接口

开发板通过一个 20pin 的连接器（J47）引出了两路 TTL（3.3V）的串口，同时引出了 5V，3.3V，1.8V 电源，可以很方便的给外接模块提供电源，通过此接口可以连接迅为电子的 GPS 模块，用户也可以连接自己的其它串口设备（需要 3.3v TTL 的串口设备）。

5. GPIO 接口

开发板通过一个 20pin 的连接器（J8）引出了一路 GPIO 接口，该接口中包括：两路 ADC，一路 SPI，5 个 GPIO，一路 TTL 串口，一路 I2C 接口，通过该接口可以连接迅为电子的继电器模块，RFID 模块，LORA 模块等。

6. ADC 电位器

板载了一路 ADC 电位器，电位器的输出电压范围是 0~3.3V，通过调节电位器的旋钮我们可以使用 i.MX6ULL 的 ADC 功能完成 0~3.3V 范围内的电压采集实验。

7. 拨码开关

i.MX6ULL 支持多种启动方式，例如 EMMC，TF 卡，NAND FLASH，QSPI，USB 串行下载等等，我们可以通过设置 i.MX6ULL 与启动方式相关的 IO 引脚的电平状态，来选择对应的启动方式，我们把对应的启动方式通过 8P 的拨码开关（SW1）引到了底板上，大家可以通过修改拨码开关上相应拨码的位置，来选择启动方式。开发板可以修改成四种启动方式，分别是：EMMC 启动，TF 卡启动，NAND FLASH 启动，USB 串行启动（OTG 烧写使用）。关于具体的设置方法，大家可以参考“5.1 i.MX6ULL 启动方式”章节。

8. 红外接收

开发板板载了一个 HS0038 红外接收模块（U38），接收的信号频率是 38kHz，通过该模块我们可以实现红外数据接收的实验，有助于我们掌握红外遥控的原理。

9. 光环境传感器

开发板板载了一个光环境三合一的传感器 AP3216C（U37），它具有环境光传感器，近（接近）距离传感器和红外传感器。通过它，i.MX6ULL 处理器可以感知到周围环境的光线变化，接近距离，从而可以实现类似手机自动调节背光的功能。

10.MPU6050 六轴传感器

开发板板载了一个六轴传感器芯片 MPU6050（U36），该芯片通过 I2C 总线与 i.MX6ULL 芯片进行数据通信。该芯片集成了陀螺仪与加速器，该传感器在姿态测量和加速度测量方面

应用广泛，大家可以通过该芯片来学习姿态测量方面的学习。

11.EEPROM 存储芯片

开发板板载了一个 EEPROM 存储芯片 AT24C02（U39），它和 i.MX6ULL 直接通过 I2C 总线进行数据传输。在我们实际开发产品的过程中，对于一些需要永久保存的配置数据，如果我们保存在 EMMC 或者 NAND FLASH 里面，我们在烧写系统的时候有可能会造成误删除，因此我们可以把这些数据保存在 EEPROM 存储芯片里面，它支持断电永久不丢失。

12.TF 卡

开发板板载了一路标准的 TF 卡接口（J3），可以方便我们扩展板子的存储功能。

13.LVDS 屏幕接口

开发板板载了两种形式的 LVDS 接口，一个是 HDMI 接口（CON3），另外一个是 30pin 的 FPC 座子（J18），这两个接口实际上是同一路 LVDS 信号，他们都是通过底板上的 RGB 转 LVDS 芯片 GM8285C（U13）转换出来的，可以用来连接迅为电子的 LVDS 屏幕，实现屏幕显示交互。

14.RGB 屏幕接口

开发板通过一个 40pin 的 FPC 座子（CN1）引出了 i.MX6ULL 的 RGB 数据信号，通过它我们可以连接迅为电子所有的 RGB 屏幕，实现屏幕显示交互。

15.标准 HDMI 接口

开发板板载了一路标准的 HDMI 接口（CON22），i.MX6ULL 芯片本身没有支持 HDMI 接口，我们通过一个 RGB 转 HDMI 的芯片 SiI9022A（U43）扩展出的 HDMI 接口，通过该接口，在没有 LCD 屏幕的情况下，我们可以使用它来连接普通的 HDMI 显示器，可以实现屏幕显示交互。

16.纽扣电池

开发板板载了一个 CR1220 的纽扣电池，当系统断电的情况下，通过这个纽扣电池可以给 i.MX6ULL 的 SNVS 域供电，从而保证 RTC 模块不断电供电（RTC 模块属于 SNVS 域），最终实现断电情况下实时时钟继续运行的目的。

17.以太网接口

开发板板载了两路以太网接口，ETHERNET1（RJ1）和 ETHERNET2（RJ2），i.MX6ULL 芯片内部支持两路 MAC 控制器，外部分别通过连接 PHY 芯片 KSZ8081（U9，U10），实现两路 10M/100M 网络接口，通过这两个 RJ45 网络接口我们可以实现网络通信功能。

18.开关机按键

开发板上引出了一路关机按键（ON/OFF），在开机运行的状态下，常按此按键可以使系统关机；在关机的状态下如果常按此按键，可以实现开机功能。

19.复位按键

开发板上引出了一个硬件复位按键（RESET），通过按下此按键，可以实现 i.MX6ULL 的复位。

20.用户按键 KEY0

开发板板载了一个输入按键（KEY0），可以通过软件配置成普通输入按键使用。

21.CAN 接口

开发板板载了两路 CAN 接口（CAN1、CAN2），分别通过 2 个绿色端子引出，每一路上是两个接口（H、L），我们在接线的时候一定要注意下：H 接 H，L 接 L（需要直连）。

22.RS485 接口

开发板板载了一路 RS485 接口（J11），通过一个绿色端子引出两个接口（A、B）和外部的其它 RS485 设备连接，我们在接线的时候一定要注意下：A 接 A，B 接 B（要直连）。

23.蜂鸣器

开发板板载了一个有源蜂鸣器（PZ1），通过 i.MX6ULL 的一个 GPIO 可以控制蜂鸣器的开或关。

24.RS485/TTL 串口选择接口

开发板板载的 RS485（J11）和 TTL 串口接口（J47）的选择接口，我们可以通过跳线帽选

择使用 RS485 还是使用 TTL 串口，这样设计的好处是通过一路 TT 串口我们可以连接 RS485 设备，也可以连接 TTL 串口设备，可以把硬件资源功能发挥到最大。

25.JTAG 接口

开发板板载了 JTAG 接口（J19），可以通过下载/调试器来调试 i.MX6ULL 处理器。

26.USB 转串口

开发板板载了一路 Mini USB 接口（J49），该接口是通过板载的 USB 转串口芯片 CH340（U59）引出的，它可以实现 TTL 串口和 USB 接口之间的数据转换，我们通过该接口可以在电脑上看到开发板的调试串口打印的信息，对于我们调试程序定位问题，很有帮助。

27.温湿度接口

开发板板载了一路温湿度采集接口（J23），该接口支持 DHT11 温湿度采集，同时也支持 DS18B20 温度采集模块。通过该接口我们可以实现 i.MX6ULL 温湿度的监控功能

28.MIC 接口

开发板板载了一个 MIC 录音输入接口（J16），我们可以使用标准的 3.5mm 的耳机连接此接口实现录音的功能。

29.PHONE 接口

开发板板载了一个 PHONE 音频输出接口，我们使用标准的 3.5mm 的耳机连接此接口可以实现音频的输出，例如听歌曲，看视频等功能。

30.系统电源指示灯

开发板板载了一个系统电源指示灯（LED1），通过该指示灯我们可以判断系统电源工作是否正常。

31.用户状态 LED

开发板板载了一个用户状态指示灯，该指示灯一端通过 LED2 指示灯连接到 3.3V 电源，另一端直接连接到 i.MX6ULL 的 GPIO 引脚上，我们可以通过软件控制 LED2 的亮灭状态。

32.WIFI/蓝牙二合一模块

开发板板载了一个 USB 接口的 WIFI/蓝牙二合一模块（U18）型号为 RTL8723，工作频段是 2.4GHz，支持 IEEE802.11bgn 等标准，支持蓝牙 4.0 协议。通过该模块我们可以连接到无线网络实现联网的功能，也可以通过蓝牙连接传输数据。

33.OTG1 接口

开发板板载的一个 Mini USB 接口（J78），用来实现从机通信，开发板通过此接口连接到电脑可以实现镜像烧写功能。

34.USB HOST 接口

开发板板载了两个标准的 USB HOST 接口（J20、J21），这两个 USB HOST 接口是 i.MX6ULL 的 OTG2 通过一个 USB HUB 芯片 FE1.1S（U15）扩展出四路 USB HOST 接口，其中的两路连接到了这两个 USB HOST 接口了（J20、J21），另外两路分别在 4G 模块和 WIFI/蓝牙模块上使用。

35.SIM 卡接口

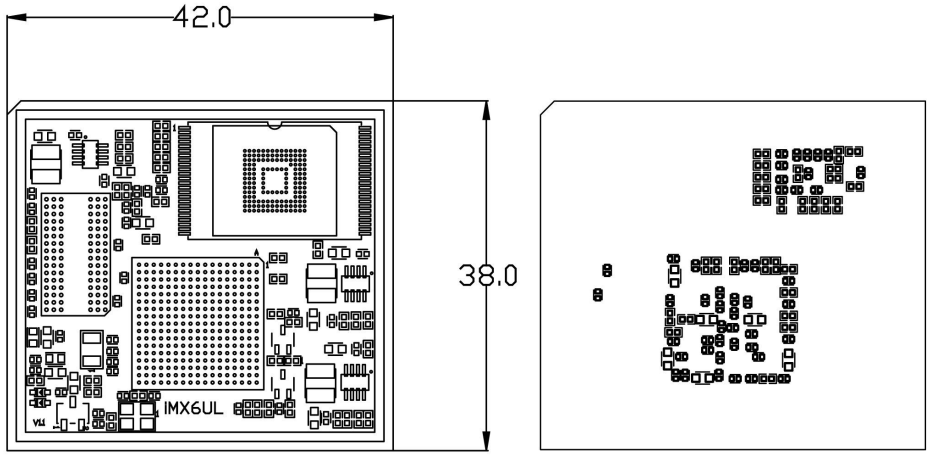
开发板板载了 SIM 卡接口（CON5），当我们使用 4G 模块的时候，需要在这里插入 SIM 卡。

36.电源开关

开发板板载了一个自锁式电源开关（J1），当我们把电源适配器连接到电源接口（JACK1），然后按下电源开关，开发板就会正常启动（确保拨码开关启动方式设置正确的前提下）。

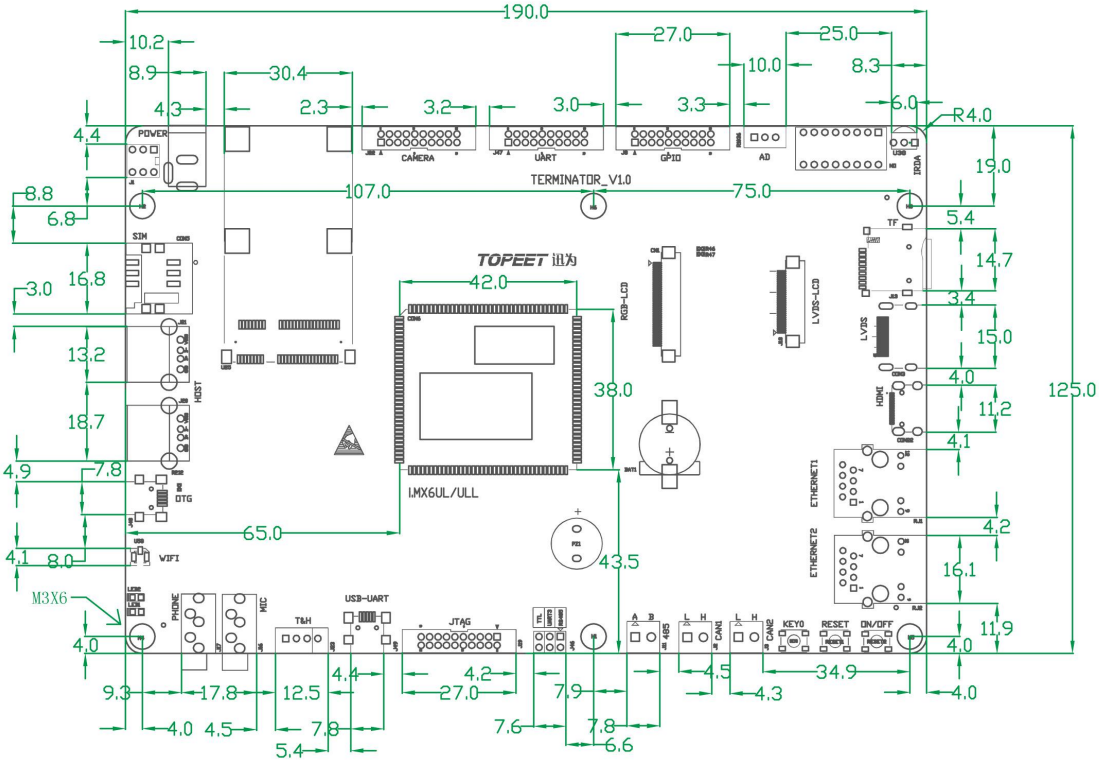
详细尺寸

核心板



底板

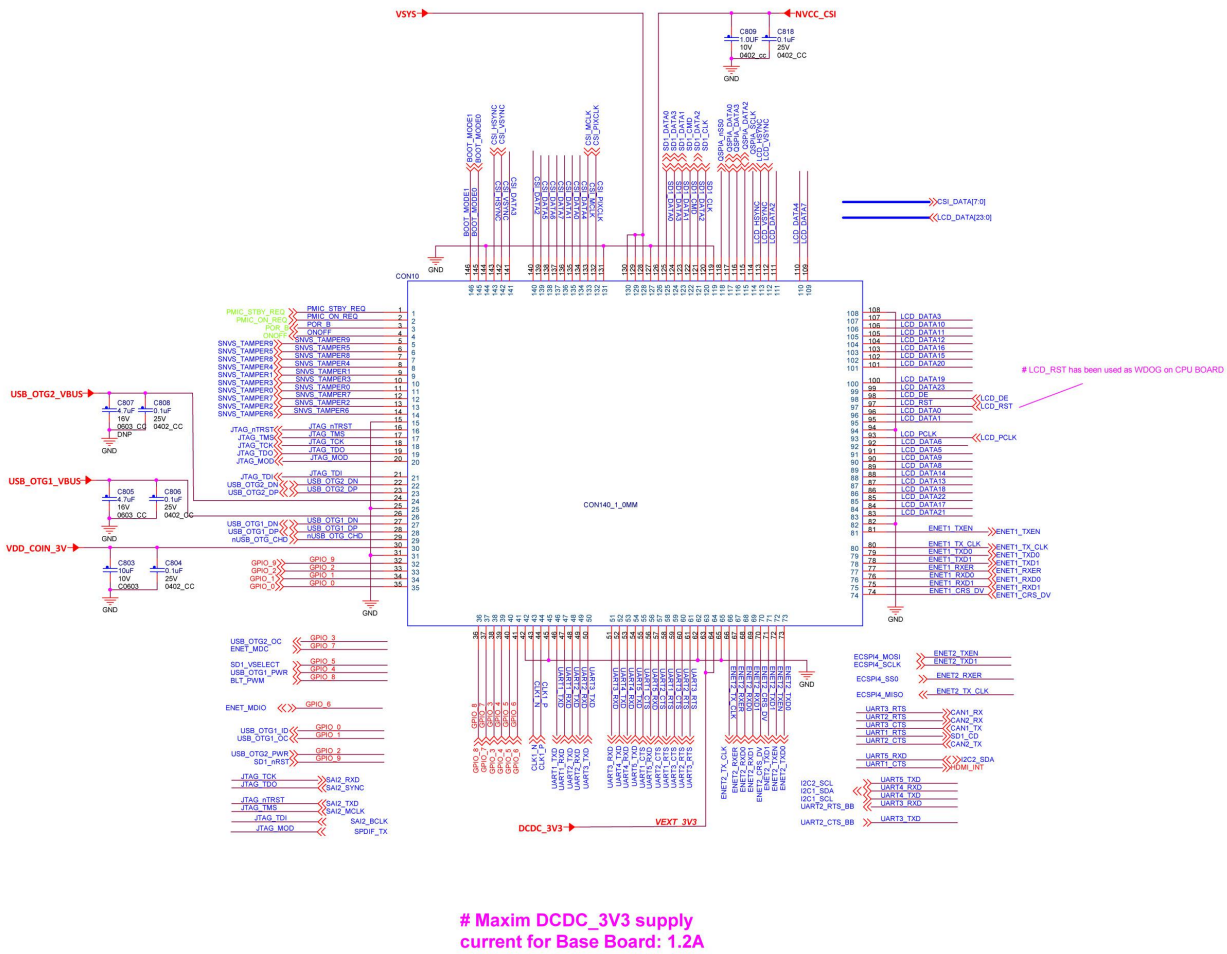
单位：MM　　误差：+-10%　　板厚：1.6　元器件最大高度:16



i.MX6ULL 终结者原理分析

1. 核心板引脚原理图

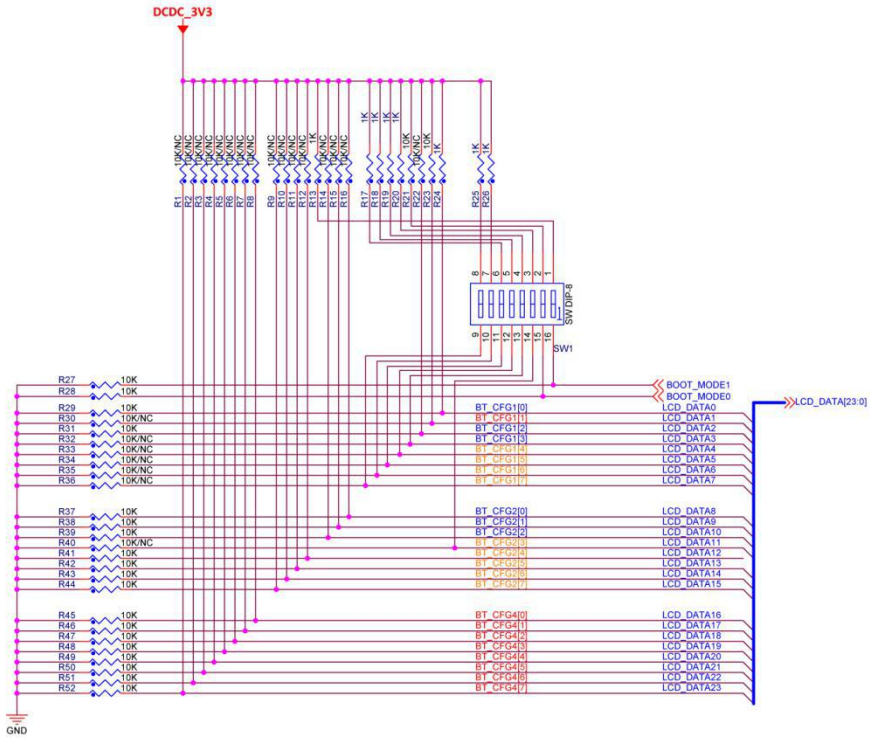
I.MX6ULL 终结者开发板采用核心板+底板的方式，核心板与底板硬件连接形式上采用的是邮票孔的方式，相比起连接器的方式此种方式具有连接性稳定，抗震动等优点。底板上相应的原理图如下图所示：



从上图我们可以看到邮票孔的封装一共引出了 146 个引脚，其中 i.MX6ULL 引出了 120 个 IO，另外还有电源，GND。考虑到信号完成性的要求，连接器上引出了尽可能多的 GND。

2. 启动方式原理部分

I.MX6ULL 支持很多种启动方式，我们可以通过设置与启动方式有关的 IO 状态来选择启动方式，具体的原理如下图所示：

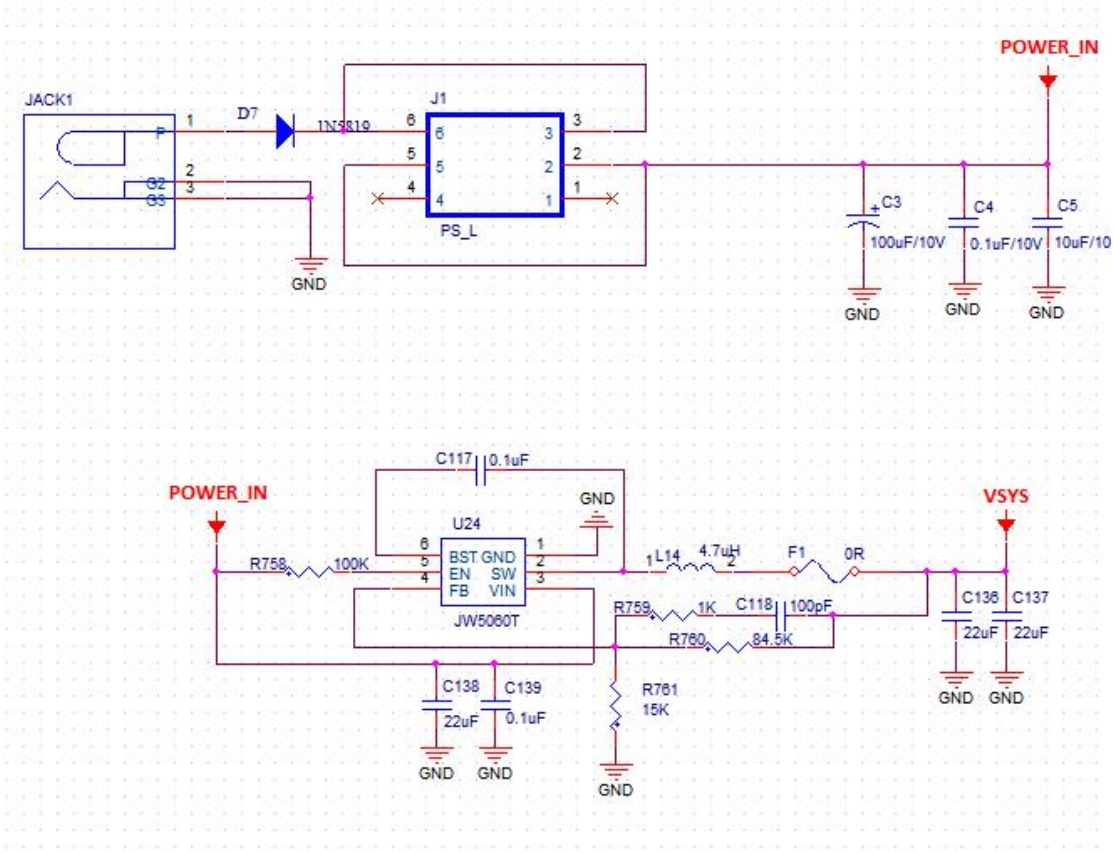


从上图我们可以看到，启动方式的 IO 中大多数的 IO 都是通过电阻下拉了，只有 8 位 IO 可以通过一个 8 位的拨码开关来选择对应的状态。具体的启动方式设置我们整理成了下面表格（关于启动方式更详细的说明，大家可以参考下 5.1 章节）：

1	2	3	4	5	6	7	8	启动设备
0	1	x	x	x	x	x	x	USB 串行启动
1	0	0	0	0	0	1	0	SD/TF 卡启动
1	0	1	0	0	1	1	0	EMMC 启动
1	0	0	0	1	0	0	1	NAND FLASH 启动

3. 系统电源接口

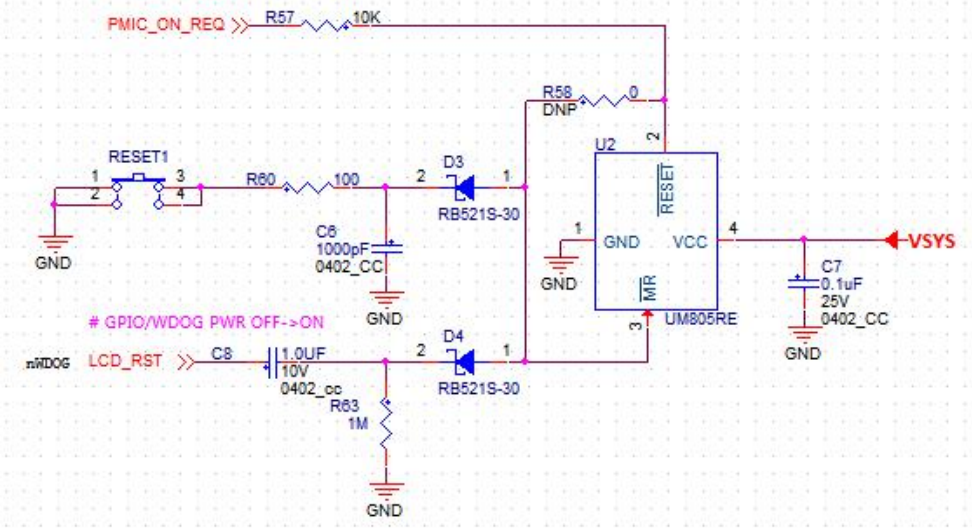
I.MX6ULL 开发板的电源供电部分原理如下图所示：



从上图我们可以看到电源部分使用了一个 DCDC 的电源芯片 U24，外部输入电源首先从 JACK1 输入，经过防反接二极管 D7 到达电源开关（J1），当我们按下电源开关（J1）的时候，电源会到达电源芯片（U24）的输入端，最终经过电源芯片会输出 5V 的电源给系统供电。由于我们使用的电源芯片 U24 是个宽电压芯片，允许输入的电压是 5V~16V 直流电源，所以我们可以很方便的就能找到一个与之匹配的电源适配器来使用，采用宽压电源芯片的优点是如果错接了 12V 的电源，板子也不会因为输入电压过高，而导致器件损坏（默认我们提供 5V 的电源适配器）。

4. 复位电路

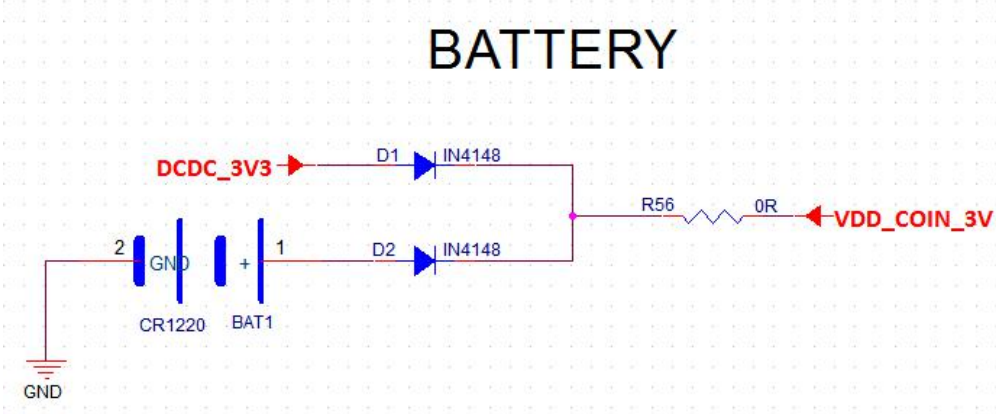
i.MX6ULL 终结者开发板的复位原理如下图所示：



从上图我们可以看到开发板是低电平产生复位。i.MX6ULL 终结者开发板通过专用的复位芯片来实现系统的复位。因为我们真正做产品的时候，有可能我们产品工作在环境非常恶略的环境下，比如电磁干扰之类的，复位引脚有可能在受到干扰的情况下，会发生瞬间的复位信号，如果这个信号直接接到处理器的复位引脚，处理器收到这个复位信号可能就会执行复位了，可是实际上我们并没有要求系统复位。因此我们可以使用专门的复位芯片，它能够排除瞬间的干扰，又可以防止系统在启动和关闭期间的误操作，保证系统的稳定。所以我们的终结者开发板不仅仅是一款学习板，而且也可以拿来作为参考，设计真正的产品。

5. 纽扣电池电路

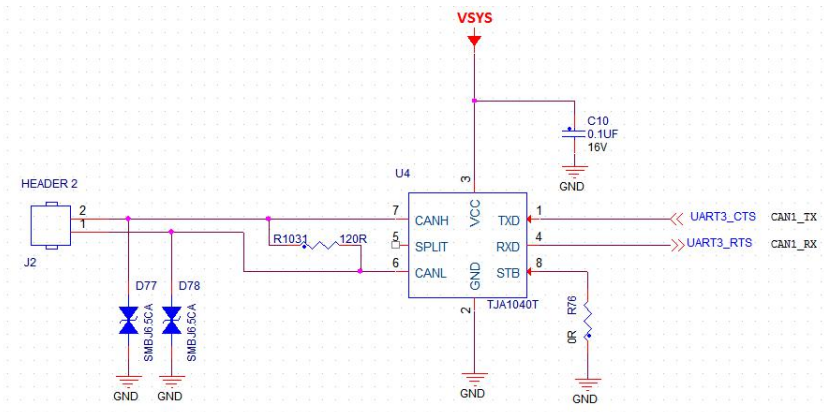
i.MX6ULL 终结者开发板的纽扣电池用来给 i.MX6ULL 的 SNVS 模块供电，保证在系统电源断电的情况下给 SNVS 模块提供持续的电源，原理图如下图所示：



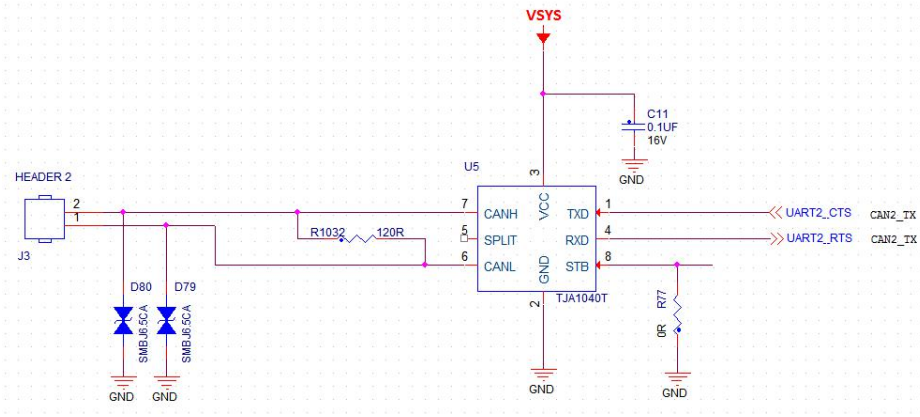
在上图中 VDD_COIN_3V 是连接到核心板的，最终给 i.MX6ULL 的 SNVS 模块供电，DCDC_3.3V 和纽扣电池（BAT1）同时给 VDD_COIN_3V 提供电源。DCDC_3.3V 是系统电源 5V 转换出来的，当系统电源断开以后，DCDC_3.3V 电源就会停止输出，此时纽扣电池会继续给 VDD_COIN_3V 提供持续的电源，最终使得 i.MX6ULL 的 SNVS 模块有 3V 的电源，从而使得 RTC 时钟模块继续运行（RTC 时钟模块属于 SNVS 模块）。

6. CAN 接口电路

i.MX6ULL 终结者开发板板载了两路 CAN 接口，原理图如下图所示：



CAN1 原理图

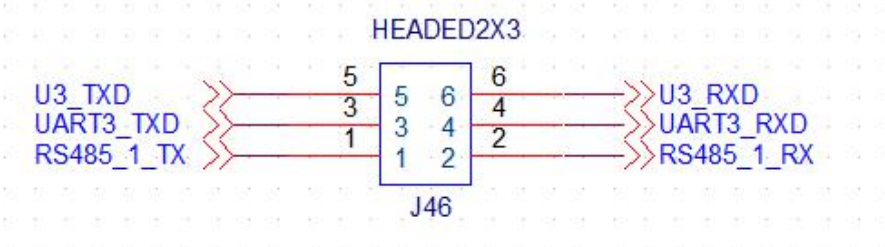


CAN2 原理图

i.MX6ULL 处理器芯片内部集成了两路 CAN 控制器，我们在底板上通过两个 CAN 的协议转换芯片（TJA1040T）分别引出了两路标准的 CAN 接口（H,L），其中 R1031 和 R1032 分别是两路 CAN 的终端匹配电阻。另外我们分别在两路 CAN 的数据线上加了 TVS 保护器件（D77，D78，D79，D80），使其能够达到抗静电的效果。

7. RS485/TTL 串口选择电路

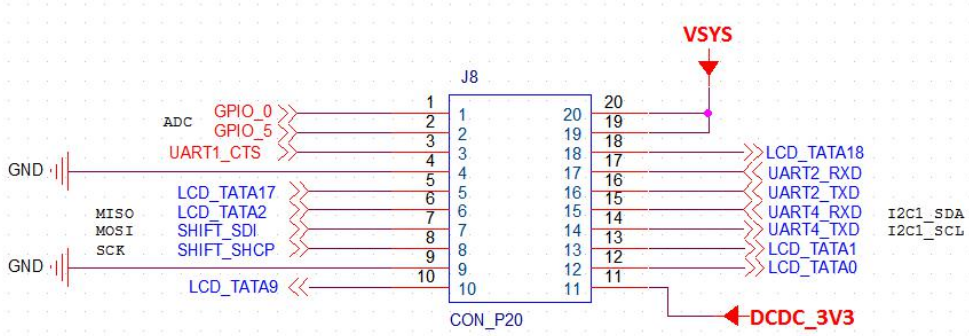
I.MX6ULL 终结者开发板板载了一个 RS485 接口，该 485 接口和 UART3 是复用的，因此我们增加了一个 RS485 和 UART3 的选择接口，原理图如下图所示：



从上图我们可以看到这个选择接口是用一个 2x3 的排针引出的，通过跳线帽可以选择是使用 RS485，还是使用 TTL 串口功能。例如，我们使用 RS485 功能，可以把上图中的 1 和 3、2 和 4 分别通过跳线帽短接起来；如果我们使用 TTL 串口功能，我们需要把 3 和 5、4 和 6 分别通过跳线帽短接起来。

8. GPIO 接口电路

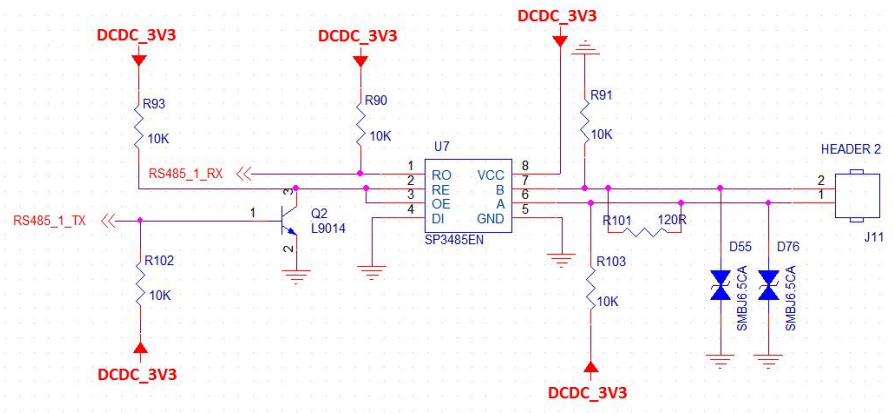
为了便于功能扩展，i.MX6ULL 终结者开发板通过一个 20pin 的排座，引出了一个 GPIO 扩展接口，原理如下图所示：



从上图可以看出该接口引出了两路 ADC，一路 SPI，一路 TTL 串口，一路 I2C，5 个 GPIO。通过这些接口我们可以很容易的扩展其他外设。

9. RS485 电路

I.MX6ULL 终结者开发板板载了一路 RS485 接口，原理图如下图所示：

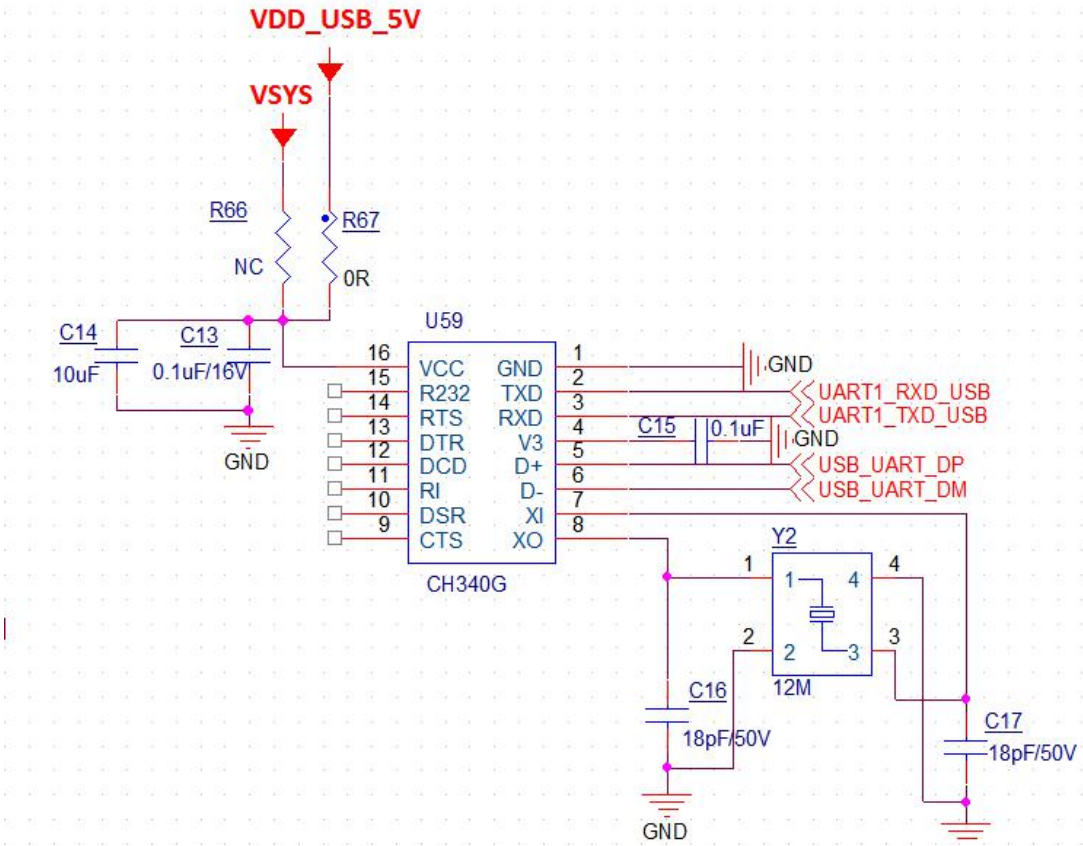


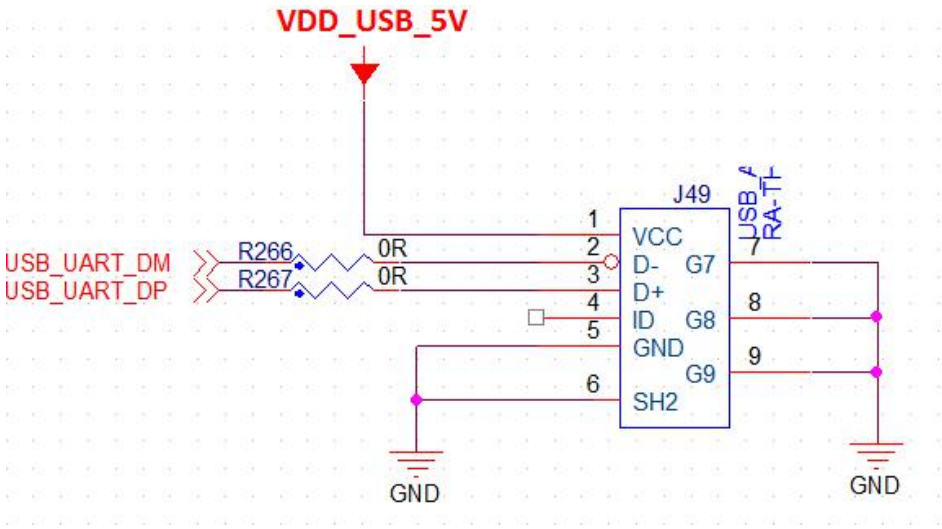
从上图我们可以看到 RS485 实际上是串口通过一个 485 协议转换芯片（SP3485EN）转换出来的，由于 SP3485EN 这个芯片是半双工的（收发不能同时进行），所以需要一个收发方向的控制引脚（SP3485EN 芯片的 2、3 引脚），一般这两个收发方向控制的引脚会连接到 CPU 处理器，软件除了要操作串口的收发，还需要控制这两个引脚的状态，这样会增加软件的工

作量。为了减少软件的工作量，我们的原理设计使用的是自收发的控制，参照我们的原理设计，我们的软件只需要实现串口的收发功能就可以，而不用去考虑设置 SP3485EN 的收发控制引脚的状态了。为了提高 485 接口的抗静电能力，我们在 485 的数据总线上加了 TVS 静电保护（D55，D76）电路。

10. USB 转串口电路

I.MX6ULL 终结者开发板板载了一个 USB 串口，原理图如下图所示：

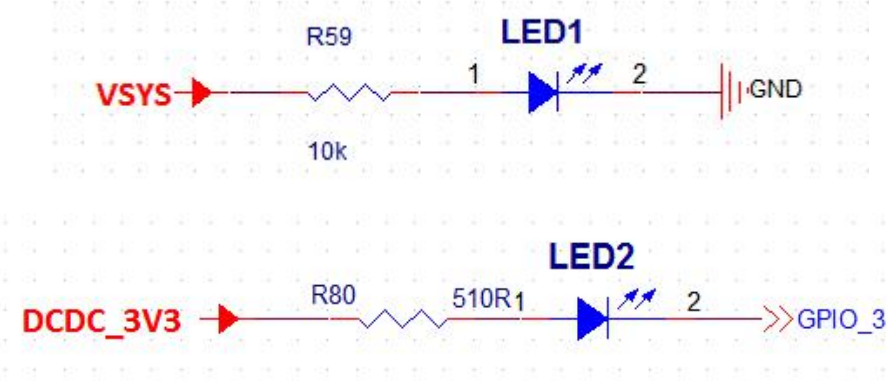




USB 转串口我们使用的是 CH340G 芯片，该芯片是由南京沁恒微电子研发生产的一款国产芯片。CH340G 的工作电压支持 3.3V、5V，甚至是 3V，从上图可以看到我们给 CH340G 的电压是 5V，并且是 Mini USB 接口提供的 5V 电源，与开发板上的电源是独立的，只要我们接上 USB 线 CH340G 就会上电。USB 转串口最终通过一个 Mini USB 座子（J49）引出。

11. LED 电路

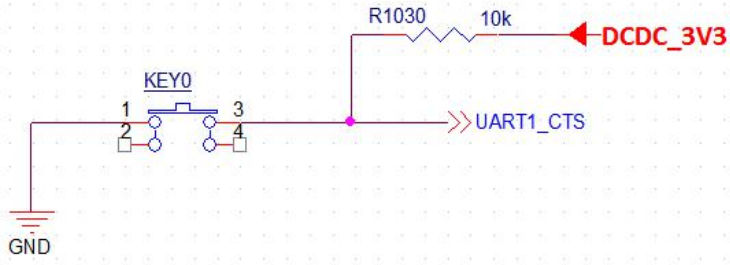
I.MX6ULL 终结者开发板板载了两个 LED 发光二极管，原理图如下图所示：



其中上面的 LED1 是系统电源指示灯。LED2 是用户 LED 灯，正极通过 510 欧的电阻连接到 3.3V 电源上，负极连接到 i.MX6ULL 的 GPIO_IO03 引脚上。

12. 按键电路

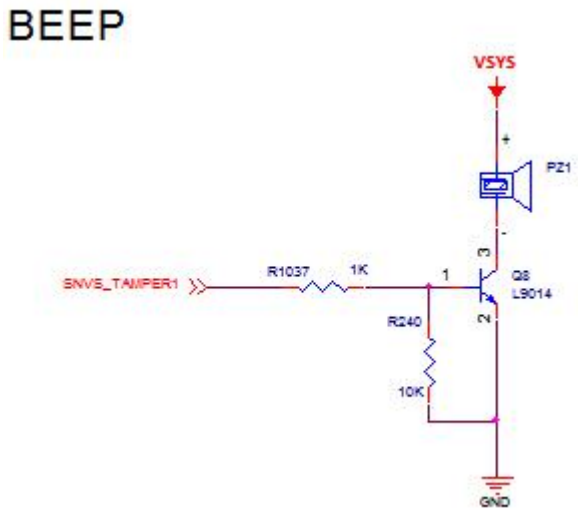
i.MX6ULL 终结者开发板板载了一个输入按键，原理如下图所示：



按键 KEY0 作为普通按键输入，一端接在 GND 上，另一端连接在 i.MX6ULL 的 UART1_CTS 引脚上，并且通过一个 10K 的电阻上拉到 3.3V。默认情况下 UART1_CTS 的引脚是高电平状态，按下按键的时候，UART1_CTS 引脚和 GND 直接连在一起，电平就会变成低。

13. 蜂鸣器电路

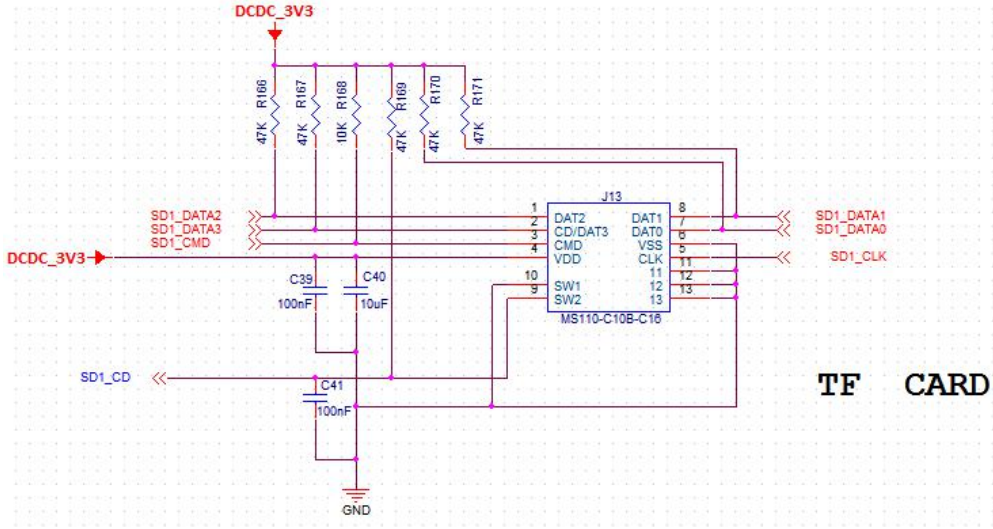
i.MX6ULL 终结者开发板板载了一个有源蜂鸣器，原理如下图所示：



蜂鸣器有两种：有源蜂鸣器和无源蜂鸣器。有源蜂鸣器内部自带了震荡电路，只需要接上电源，就会震荡发声；无源蜂鸣器需要外接一个定频（2~5KHz）的驱动信号，才会发声。为了电路设计简单，方便大家使用，我们使用的是有源蜂鸣器。蜂鸣器的开关通过控制三极管的通断来实现，我们使用 i.MX6ULL 的 SNVS_TAMPER1 这个 IO 来控制三极管的通断。

14. TF 卡电路

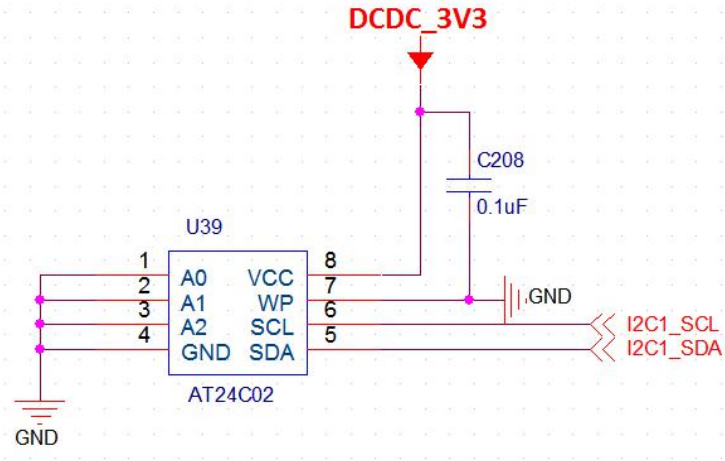
i.MX6ULL 终结者开发板板载了一个 TF 卡接口，原理如下图所示：



开发板采用标准的 TF 卡插座，采用 USDHC 驱动，SD1_DATA0、SD1_DATA1、SD1_DATA2、SD1_DATA3 是 4 位数据总线，分别连接到 i.MX6ULL 的 SD1_DATA0~SD1_DATA03 引脚上面。SD1_CMD 和 SD1_CLK 分别是 USDHC 的命令和时钟线，分别接到了 i.MX6ULL 的 SD1_CMD 和 SD1_CLK 引脚上了。SD1_CD 是 TF 卡的插拔检测引脚，通过该引脚状态可以检测是不是有 TF 卡连接。

15. EEPROM 电路

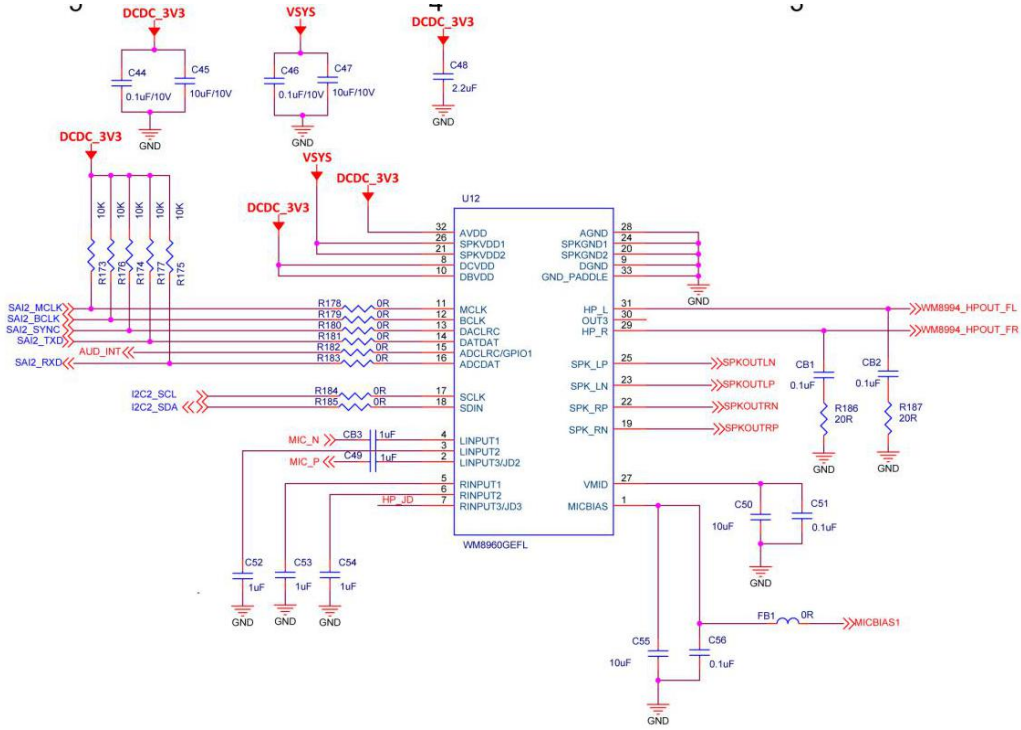
i.MX6ULL 终结者开发板板载了一个 EEPROM 存储芯片，原理如下图所示：

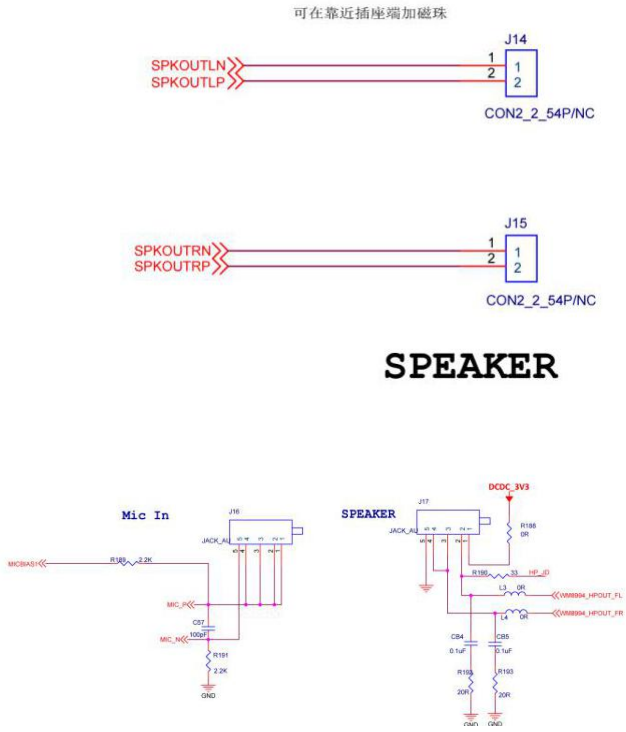


EEPROM 存储芯片我们使用的是 AT24C02 芯片，该芯片的容量是 2K bit，它与 i.MX6ULL 通过 I2C 总线进行通信。

16. 音频电路

i.MX6ULL 终结者开发板板载了一个音频编解码芯片 WM8960，原理如下图所示：





WM8960 是一款低功耗、立体声编解码芯片，内部集成了 24 位高性能的 DAC/ADC，并且支持 3D 音效等功能。采用 D 类扬声器驱动器，为 8Ω负载提供每通道 1W 功率。集成完整的麦克风接口和立体声耳机驱动器。由于无需单独的麦克风、扬声器或耳机放大器，因此显著降低了外部元件的需求。高级片上数字信号处理功能为麦克风或线路输入执行自动电平控制。

图中的 WM8960 的 SPK+和 SPK-分别通过两组排针引出，方便用户连接 8Ω 1W 的喇叭。J16（Mic In）是通过 3.5mm 的耳机接口实现立体声录音。J17（SPEAKER）是 3.5mm 的耳机接口，用来实现音频的输出。

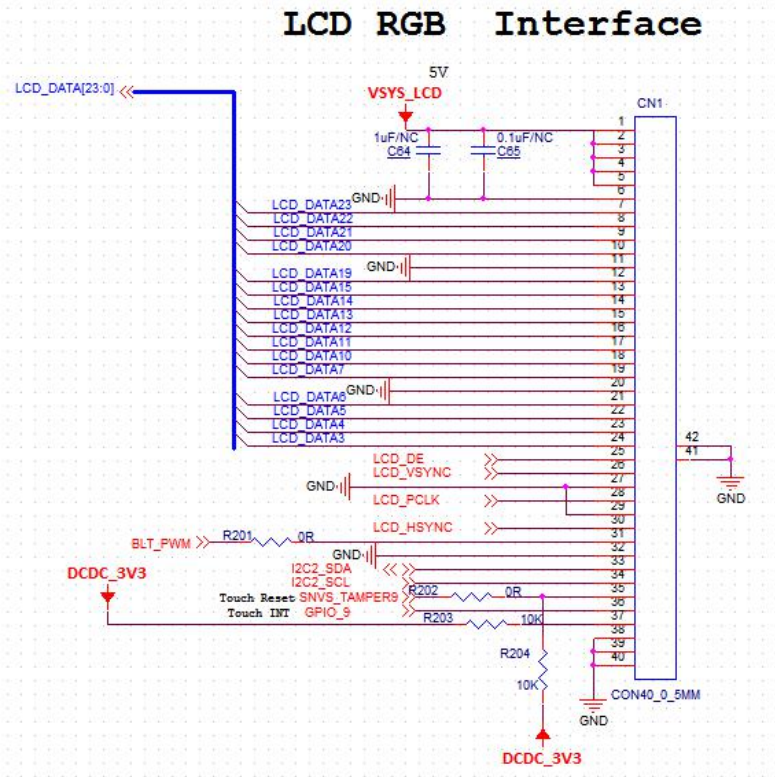
WM8960 芯片与 i.MX6ULL 通过 SAI 接口连接，图中的 SAI2_MCLK、SAI2_BCLK、SAI2_SYNC、SAI2_TXD、SAI2_RXD 分别接在 i.MX6ULL 的 JTAG_TMS、JTAG_TDI、JTAG_TDO、JTAG_TRST、JTAG_TCK 引脚上。

WM8960 还通过 I2C 连接到 i.MX6ULL 的 I2C2 总线上了，在使用 WM8960 之前，我们需

要通过这个 I2C 接口对他进行配置。

17. RGB 屏幕电路

i.MX6ULL 终结者开发板板载了一路 RGB 屏幕接口，如下图所示：

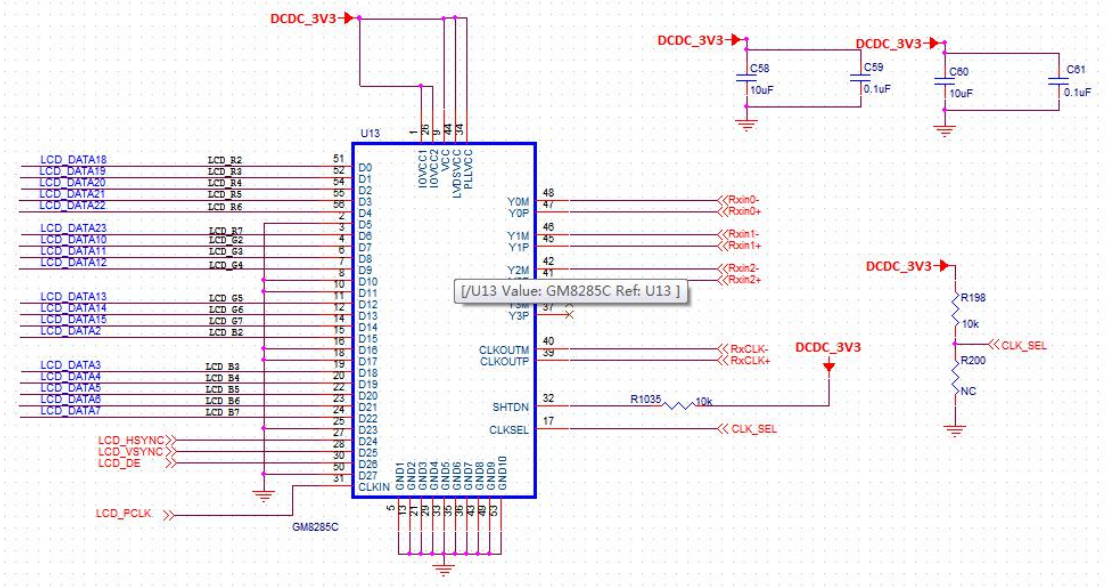


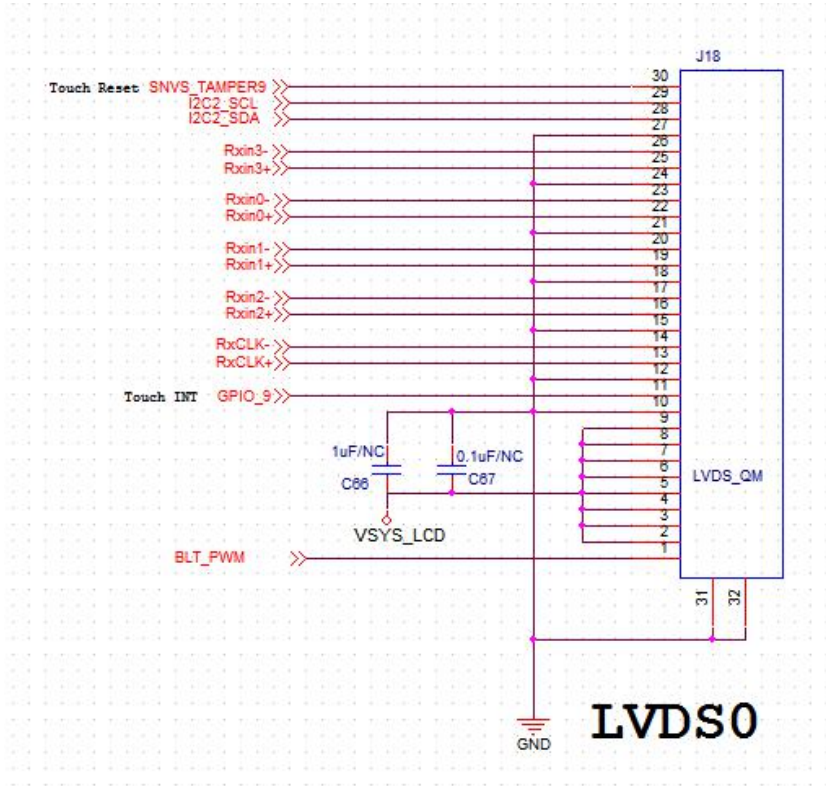
上图中 CN1 是 RGB 的屏幕接口，通过一个 40pin 的 FPC 座子引出，同时也支持触摸屏。该接口仅支持 RGB 接口的屏幕，目前迅为电子的 RGB 接口屏幕有 4.3 寸（480*272），5 寸（800*600），7 寸（1024*600）。

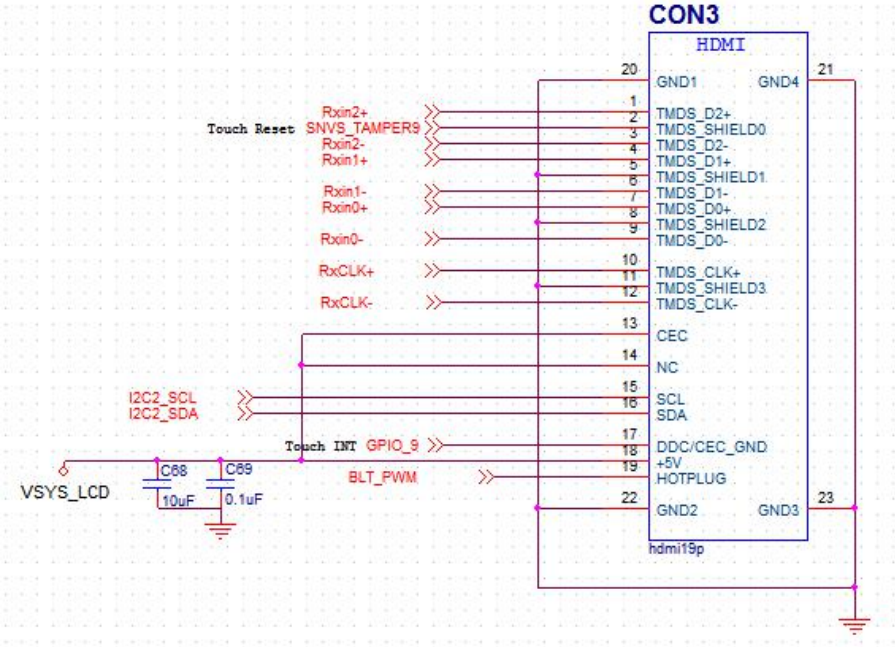
上图中的 BLT_PWM 是控制屏幕背光的引脚，他连接到了 i.MX6ULL 的 GPIO1_IO08 引脚。SNVS_TAMPER9 是控制触摸芯片复位的，它连接到了 i.MX6ULL 的 SNVS_TAMPER9 引脚上面。GPIO_9 是触摸的中断引脚，它连接到了 i.MX6ULL 的 GPIO1_IO09 引脚上面。I2C2_SDA 和 I2C2_SCL 是用于和触摸芯片通信的，他们分别连到了 i.MX6ULL 的 UART5_RX_DATA 和 UART5_TX_DATA 的引脚上面。

18. LVDS 屏幕接口

I.MX6ULL 开发板板载了两种不同接口形式的 LVDS 接口，原理如下图所示：





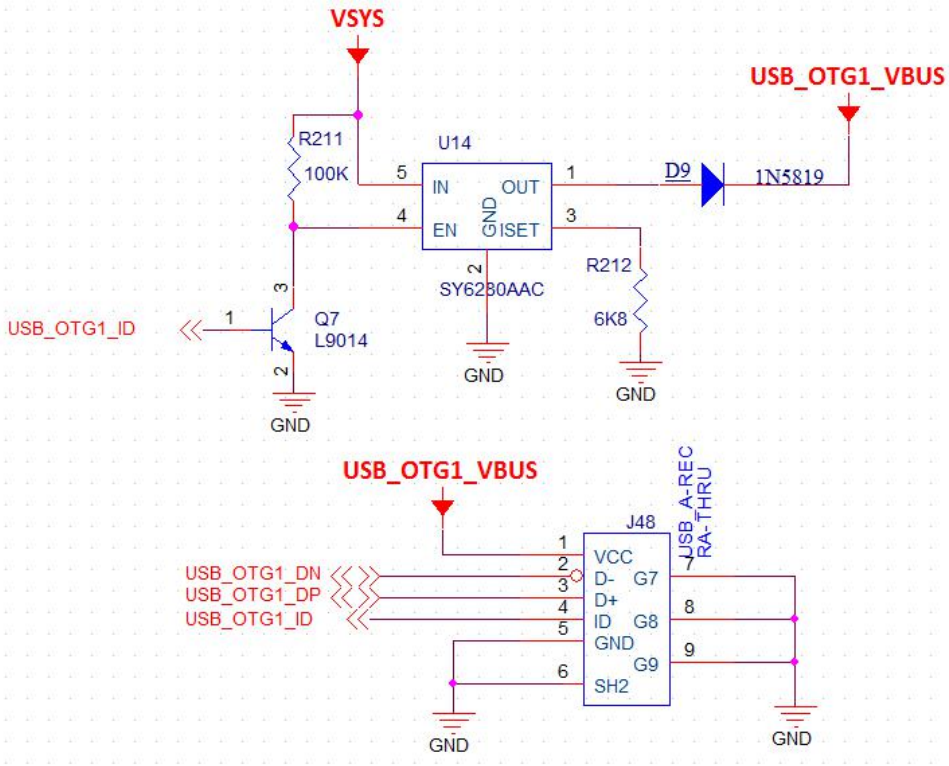


从上图我们可以看到 LVDS 接口是 RGB 信号通过芯片 GM8285C（U13）转换出来的，GM8285C 最大支持将 28 位并行数据转换为 4 对串行 LVDS 差分信号，同时并行输出 1 路 LVDS 差分时钟信号。I/O 电压支持 1.8V/3.3V。

在上图我们看到有两种 LVDS 屏幕接口：J18（30pin 的 FPC 座子）和 CON3（HDMI 座子）。这两个接口实际上是一路 LVDS 信号，只是对外引出的方式不一样，因为迅为电子的 LVDS 屏幕有两种接口（30pin FPC 座子和 HDMI 接口），所以为了适配迅为电子屏幕的接口，i.MX6ULL 终结者底板引出了这两种接口。（**大家一定要注意下：CON3（HDMI 座子）只能用来连接迅为电子的 LCD 屏幕，不能用来连接电脑显示器！！！**）。

19. USB OTG 电路

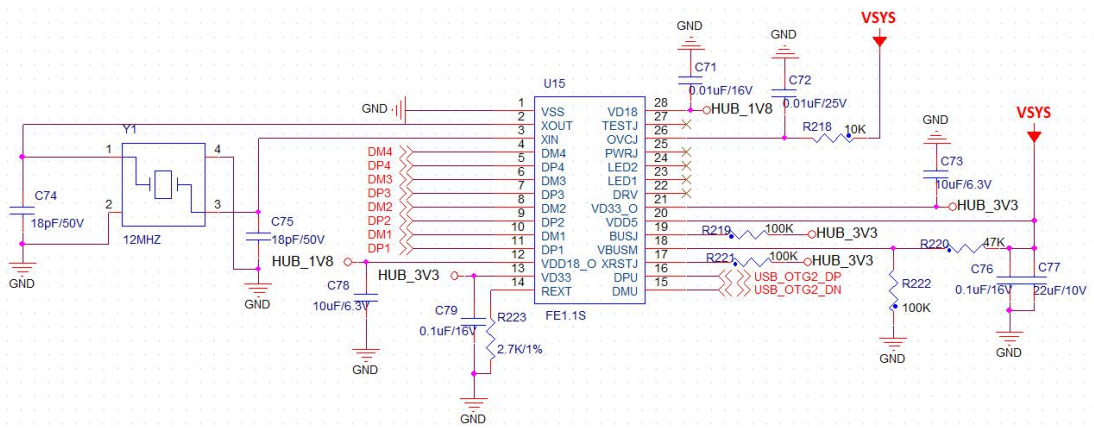
I.MX6ULL 终结者开发板板载了一路 USB OTG 接口，原理如下图所示：

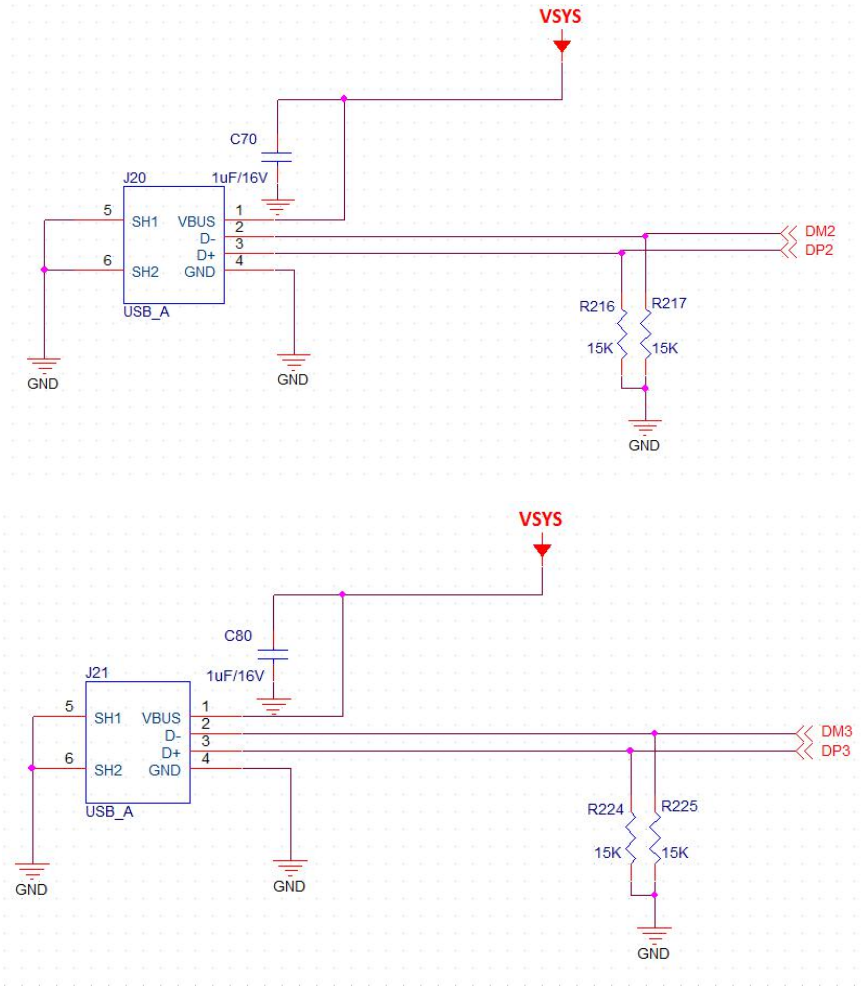


上图中 USB OTG 接口的 USB_OTG1_DN 和 USB_OTG1_DP 分别接到了 i.MX6ULL 的 USB_OTG1_DN 和 USB_OTG1_DP 引脚上了。USB OTG 接口通过一个 Mini USB 座子引出，我们可以通过此接口给开发板烧写镜像。

20. USB HOST 接口

i.MX6ULL 终结者开发板提供两路 USB HOST 接口，原理如下图所示：



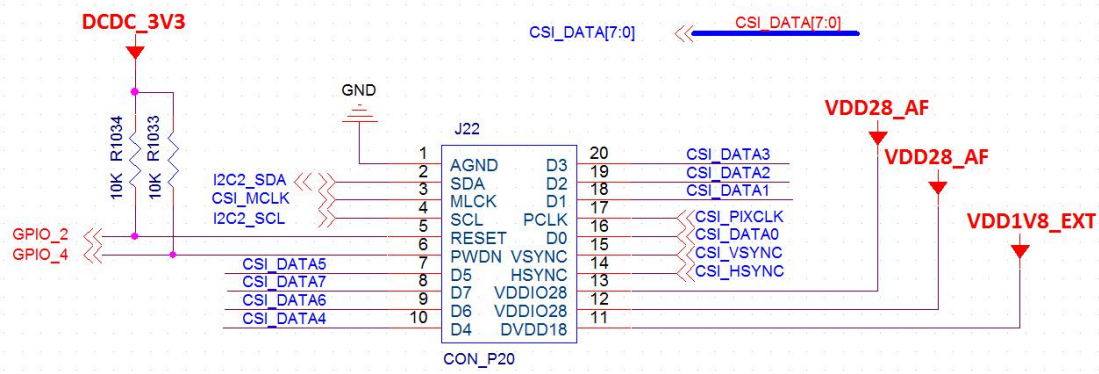


从图中我们可以看到两路 USB HOST 是 i.MX6ULL 的 USB OTG2 接口（USB_OTG2_DP，USB_OTG2_DN）通过一个芯片 FE1.1S（U15）扩展出的。FE1.1S 是一款高性能，低功耗的 USB HUB 芯片，他支持 USB2.0 协议，可以把一路 USB HOST，能扩展出 4 路 USB HOST 接口。

上图中的 J20 和 J21 是 FE1.1 扩展出的其中两路 USB HOST，通过这两个 USB 接口我们可以链接 USB Device（例如：U 盘，鼠标，键盘等等）。

21. CSI 摄像头接口电路

i.MX6ULL 终结者开发板板载了一路 CSI 的摄像头接口，原理如下图所示：

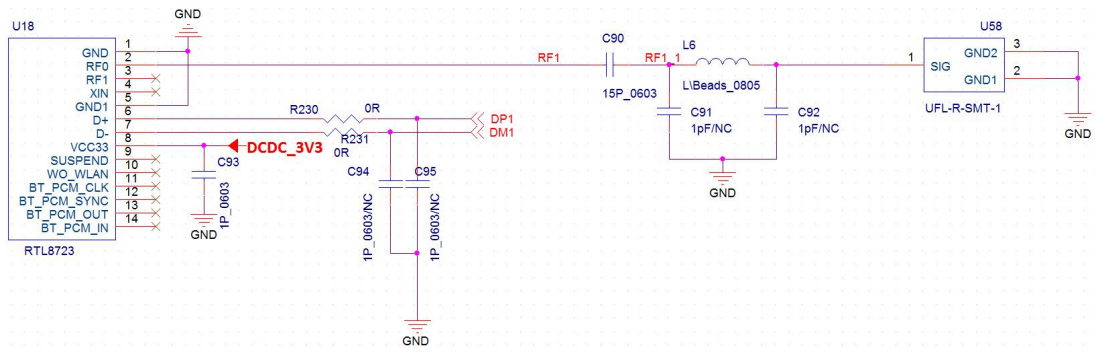


图中的 J22 接口可以用来连接迅为电子的 500w 摄像头模块。其中的 CSI_MCLK、CSI_PIXCLK、CSI_HSYNC、CSI_VSYNC、CSI_DATA0、CSI_DATA1、CSI_DATA2、CSI_DATA3、CSI_DATA4、CSI_DATA5、CSI_DATA6、CSI_DATA7 这些信号连接到 i.MX6ULL 摄像头模块相应的引脚上。

图中的 I2C2_SDA 和 I2C2_SCL 连接到 i.MX6ULL 的 I2C2 总线上，摄像头模块需要 i.MX6ULL 通过 I2C 初始化配置以后才能正常工作。GPIO_2 控制摄像头模块的复位，它连接到 i.MX6ULL 的 GPIO1_IO02 引脚上。GPIO_4 控制摄像头模块的开关使能，它连接到 i.MX6ULL 的 GPIO1_IO04 引脚上。

22. WIFI/蓝牙电路

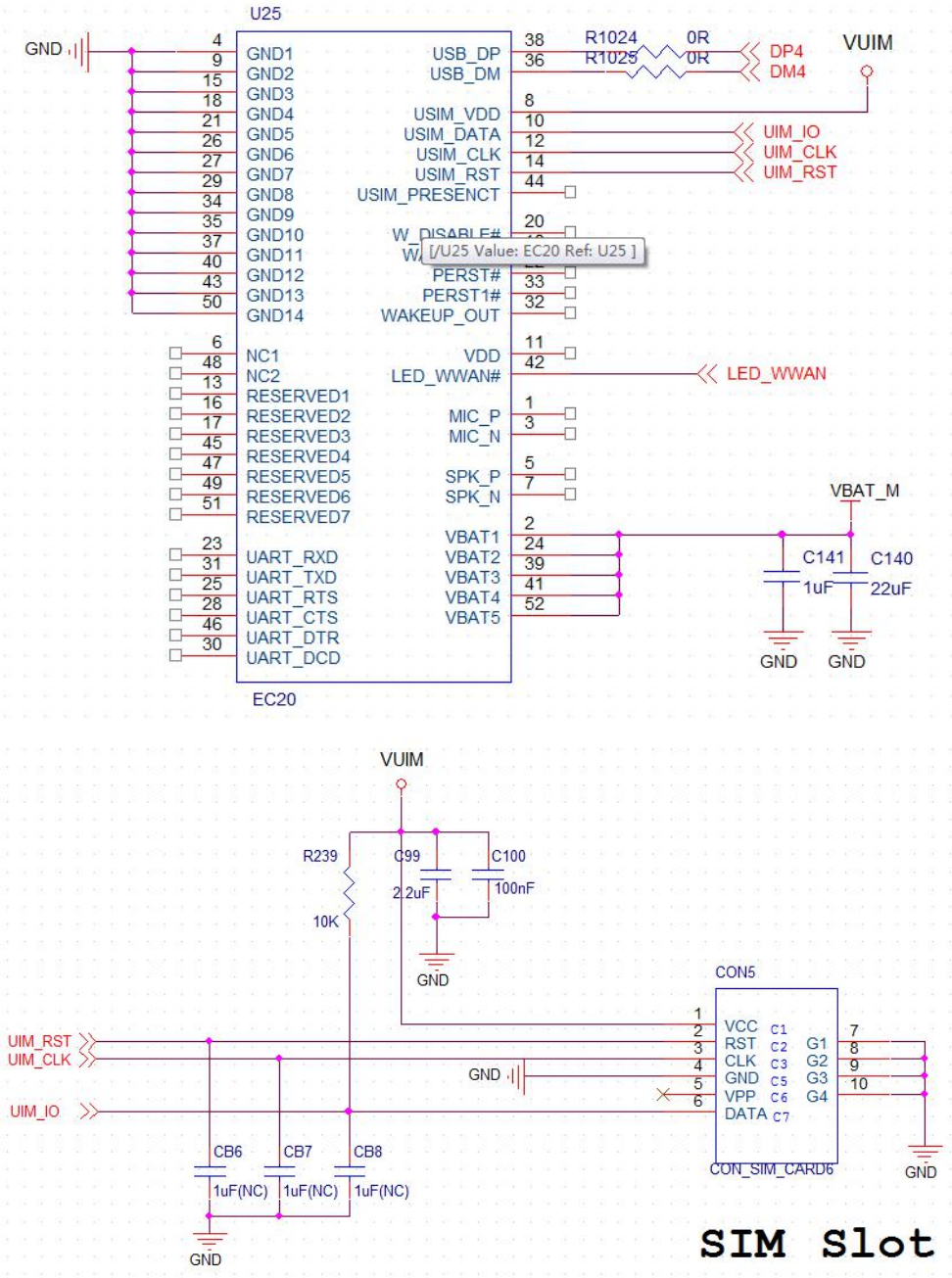
i.MX6ULL 终结者开发板板载了一个 WIFI/蓝牙二合一的模块，原理如下图所示：



我们使用的 WIFI/蓝牙二合一模块型号是 RTL8723，这个模块的电路非常简单，使用 3.3V 电源，与 i.MX6ULL 通过 USB 总线传输数据，对外引出了 ipex 的天线接口（U58）。

23. PCIE 4G 模块电路

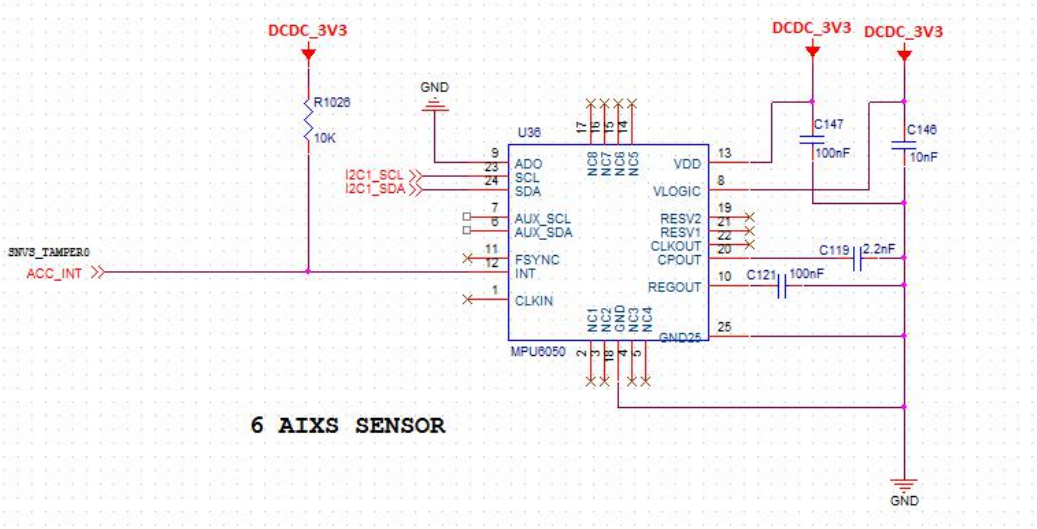
i.MX6ULL 终结者开发板板载了一个 Mini PCIE 的 4G 模块接口，原理如下图所示：



上图中 U25 是 Mini PCIE 座子，可以用来连接 Mini PCIE 的 4G 模块，比如移远的 EC20 模块，高新兴的 ME3630 模块。虽然我们使用的是 Mini PCIE 接口，实际上传输数据用到的是 USB 接口（使用的 FE1.1S 扩展出来的一路 USB 接口）。上图中的 CON5 是 SIM 卡插座，使用 4G 模块联网，我们必须要插入 SIM 卡。

24. 六轴传感器电路

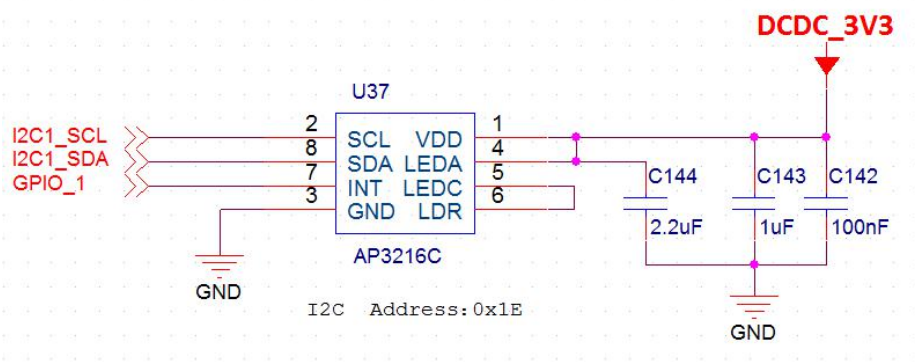
i..MX6ULL 终结者开发板板载了一个 6 轴重力加速度计，原理如下图所示：



从上图我们可以看到 6 轴重力加速度计芯片是 MPU6050（U36），该芯片内部集成了：三轴加速度传感器和三轴陀螺仪。我们使用 I2C 来访问它。I2C1_SCL 和 I2C1_SDA 分别连接到 i.MX6ULL 的 UART4_TX_DATA、UART4_RX_DATA 这两个 IO 上面了。

25. 光环境传感器电路

i.MX6ULL 终结者开发板板载了一个光环境传感器，如下图所示：

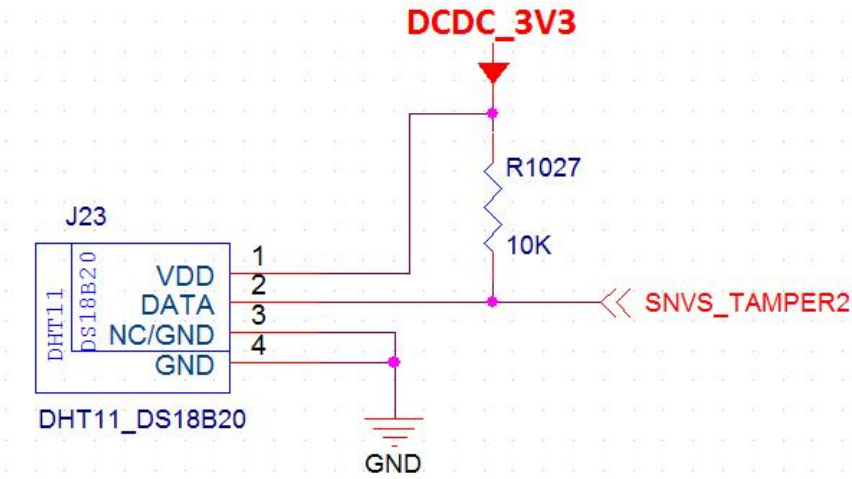


上图中的 U37 是光环境传感器 AP3216C，该芯片可以感应周围光线的强弱，接近距离和红外强度，使用的是 I2C 的接口，I2C1_SCL 和 I2C1_SDA 分别连接到 i.MX6ULL 的 UART4_TX_DATA、UART4_RX_DATA 这两个 IO 上面了，GPIO_1 是 AP3216C 的中断输出引脚，

连接在 i.MX6ULL 的 GPIO1_IO01 上面了。

26. 温湿度传感器电路

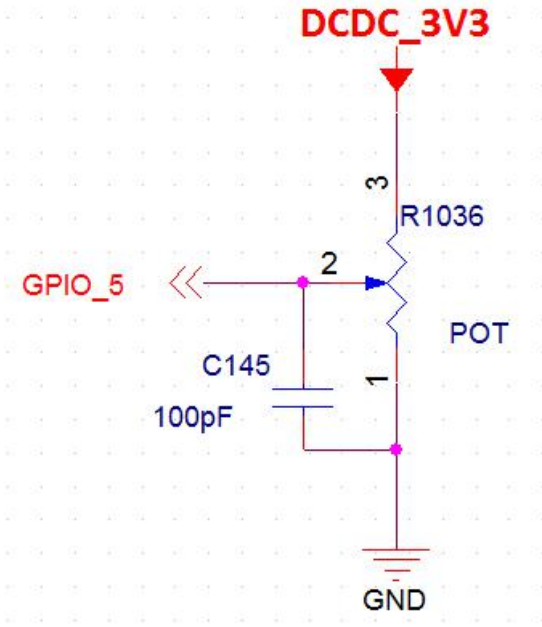
I.MX6ULL 终结者开发板板载了温湿度传感器接口，原理如下图所示：



从上图可以看到该接口可以兼容 DHT11（温湿度采集模块）和 DS18B20（温度采集模块）。该接口的电路比较简单，3.3V 电源供电，一个数据引脚直接连到 i.MX6ULL 的 SNVS_TAMPER2 引脚上了。

27. ADC 电位器电路

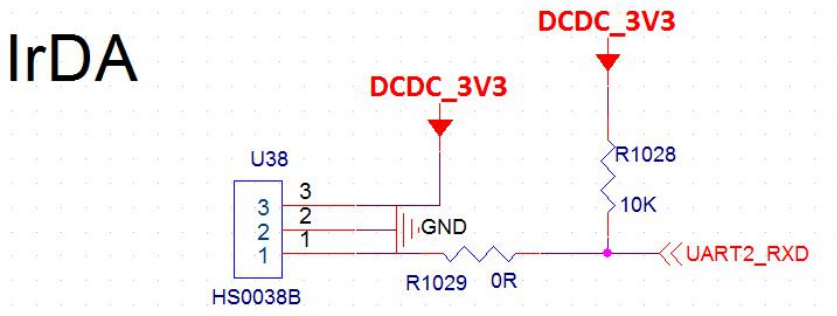
i.MX6ULL 终结者开发板板载了一个电位器，原理如下图所示：



从上图可以看到电位器（R1036）的原理很简单，使用 3.3V 供电，输出引脚通过 GPIO_5 链接到 i.MX6ULL 的 GPIO1_IO05 引脚上了。GPIO1_IO05 可以复用成 ADC 功能，这样就能够实现电位器的电压采集了，我们调节电位器上的旋钮，ADC 采集到的电压会在 0~3.3V 范围内变化。

28. 红外接收电路

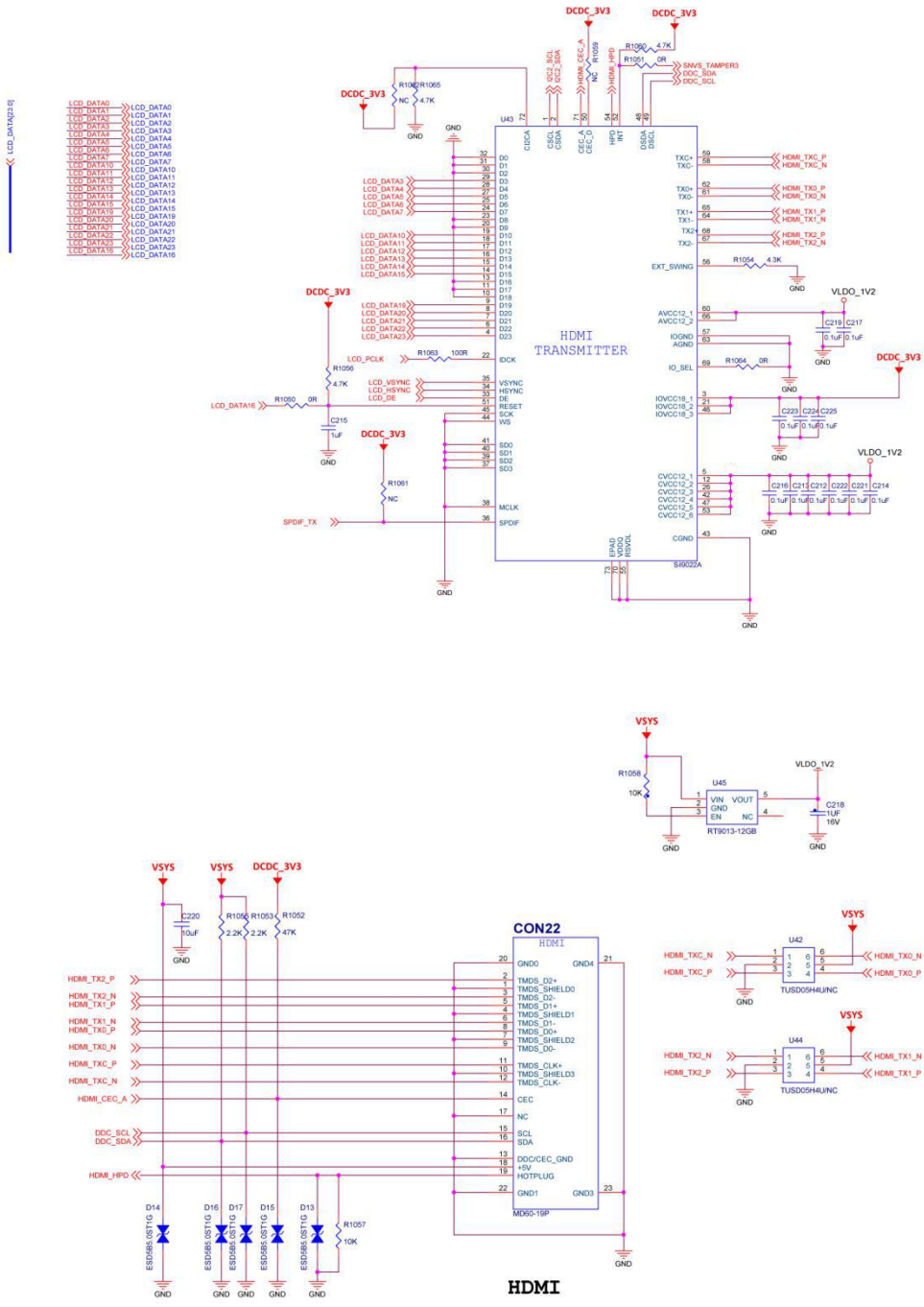
i.MX6ULL 终结者开发板板载了一个红外接收模块，原理如下图所示：



从上图可以看到红外接收模块（U38）使用的是 HS0038B 模块，该模块的原理很简单，使用 3.3V 供电，数据输出引脚 UART2_RXD 连接到 i.MX6ULL 的 UART2_RX_DATA 引脚上面了。

29. HDMI 接口电路

i.MX6ULL 终结者开发板板载了一路 HDMI 接口，原理如下图所示：

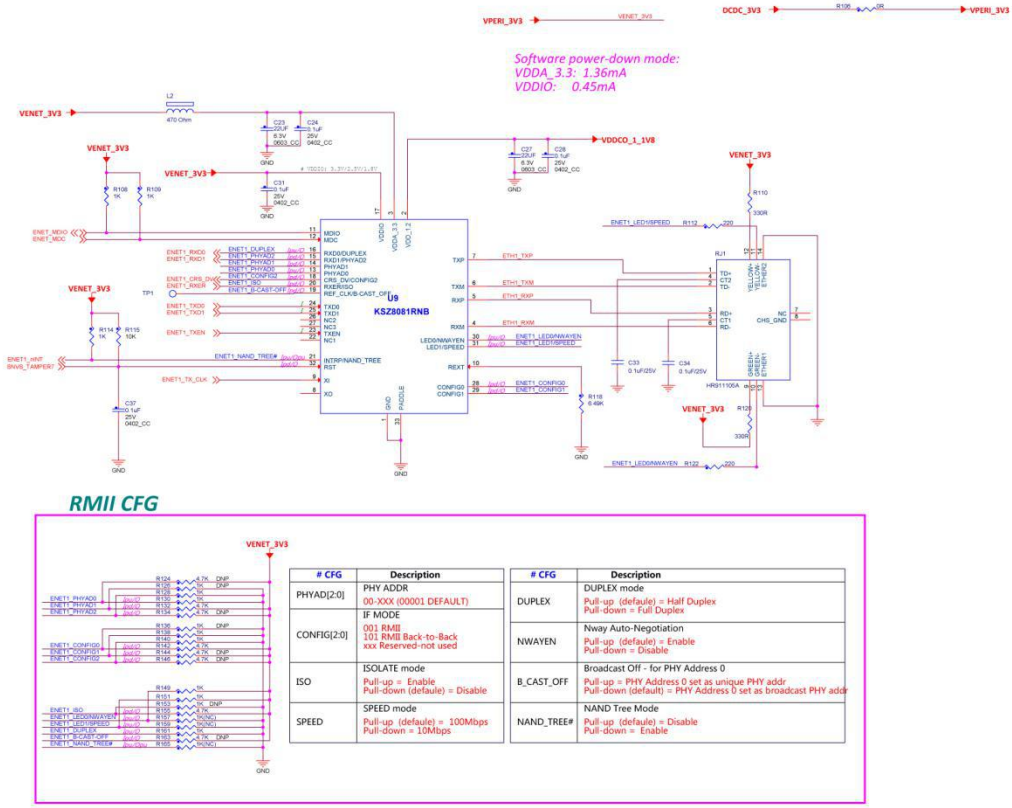


I.MX6ULL 处理器本身是不支持 HDMI 接口的，从上图我们可以看到 HDMI 接口是 RGB 接口通过 Sil9022A（U43）转换出来的。Sil9022A 可以将 24 位的 RGB 数据转换成标准的 HDMI 信号，i.MX6ULL 的 RGB 模块相关的引脚与该芯片直接相连。另外该还需要通过 I2C（I2C2_SCL、

I2C2_SDA）与 i.MX6ULL 进行连接，因为在使用之前，需要通过 I2C 对该芯片进行配置。CON22 是标准的 HDMI 接口，我们可以通过 HDMI 线，连接显示器到该接口。

30. 以太网电路

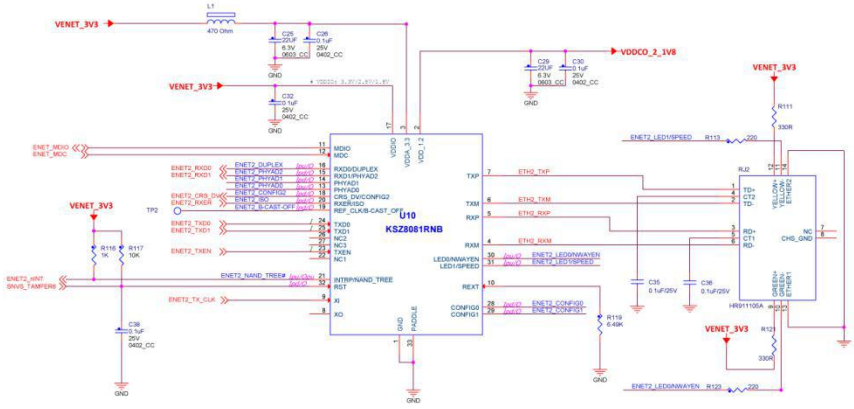
i.MX6ULL 终结者开发板板载了两路以太网接口，分别是 ETHERNET1 和 ETHERNET2，其中 ETHERNET1 的原理如下图所示：



# CFG	Description
PHYAD[2:0]	PHY ADDR 00-XXX (00001 DEFAULT)
CONFIG[2:0]	IF MODE 001 RMII 101 RMII Back-to-Back xxx Reserved-not used
ISO	ISOLATE mode Pull-up = Enable Pull-down (defaule) = Disable
SPEED	SPEED mode Pull-up (defaule) = 100Mbps Pull-down = 10Mbps

# CFG	Description
DUPLEX	DUPLEX mode Pull-up (defaule) = Half Duplex Pull-down = Full Duplex
NWAYEN	Nway Auto-Negotiation Pull-up (defaule) = Enable Pull-down = Disable
B_CAST_OFF	Broadcast Off - for PHY Address 0 Pull-up = PHY Address 0 set as unique PHY addr Pull-down (default) = PHY Address 0 set as broadcast PHY addr
NAND_TREE#	NAND Tree Mode Pull-up (defaule) = Disable Pull-down = Enable

ETHERNET1 的原理如下图所示：



RMII CFG

# CFG	Description
PHYAD[2:0]	PHY ADDR 00-XXX (00001 DEFAULT)
CONFIG[2:0]	IF MODE 001 RMII 101 RMII Back-to-Back xxx Reserved-not used
ISO	ISOLATE mode Pull-up = Enable Pull-down (defaule) = Disable
SPEED	SPEED mode Pull-up (defaule) = 100Mbps Pull-down = 10Mbps

# CFG	Description
DUPLEX	DUPLEX mode Pull-up (defaule) = Half Duplex Pull-down = Full Duplex
NWAYEN	Nway Auto-Negotiation Pull-up (defaule) = Enable Pull-down = Disable
B_CAST_OFF	Broadcast Off - for PHY Address 0 Pull-up = PHY Address 0 set as unique PHY addr Pull-down (default) = PHY Address 0 set as broadcast PHY addr
NAND_TREE#	NAND Tree Mode Pull-up (defaule) = Disable Pull-down = Enable

I.MX6ULL 内部集成了两个 MAC 控制器，每个 MAC 外接一个 PHY 芯片，就可以实现网络通信功能。我们使用的是 KSZ8081RNB 这个 PHY 芯片，该芯片一端与 i.MX6ULL 通过 RGMI 接口连接，另一端连接到带网络变压器的 RJ45 接口，组成一个 10M/100M 自协商的网卡。

ETNERNET1 和 ETHERNET2 通过 MDIO 接口与 i.MX6ULL 连接在一起，通过 MDIO 接口 i.MX6ULL 可以读写 PHY 芯片的寄存器，从而可以对 PHY 芯片进行配置。MDIO 接口由两根线组成 ENET_MDIO 和 ENET_MDC。这两根线分别连接到了 i.MX6ULL 的 GPIO1_IO06 和 GPIO1_IO07 引脚上面了。另外 ETHERNET1 和 ETHERNET2 分别有一个复位引脚，这两个复位引脚是 SNVS_TAMPER7 和 SNVS_TAMPER8，这两个引脚连接到 i.MX6ULL 的 SNVS_TAMPER7 和 SNVS_TAMPER8 引脚上了。

软件资源

裸机例程

例程描述			
1	基于汇编的 LED 例程	10	基于工程结构的按键定时器消抖
2	基于 C 语言的 LED 例程	11	基于工程结构的高精度延时例程
3	基于 SDK 的 LED 例程	12	基于工程结构的串口例程
4	基于工程结构的 LED 例程	13	基于工程结构的 printf 例程
5	基于工程结构的蜂鸣器例程	14	基于工程结构的 LCD 例程
6	基于工程结构的按键例程	15	基于工程结构的实时时钟例程
7	基于工程结构的时钟例程	16	基于工程结构的 I2C 通信例程
8	基于工程结构的中断例程	17	基于工程结构的触摸例程
9	基于工程结构的定时器例程	18	基于工程结构的背光例程

Linux 内核及文件系统源码结构

类型	描述	备注
ARM 交叉编译器	版本：4.9.4	提供软件工具
Uboot	版本：2016.03	提供源码 支持 LCD 显示；支持网口；支持

		EMMC,NAND FLASH；支持环境变量修改保存；
Linux 内核	版本：4.1.15	提供源码
文件系统	支持 Busybox（1.29.0）、buildroot（2019.02.9）、yocto（imx-4.1-krogoth）、Ubuntu（16.04.6）这四种文件系统	提供详细的制作文档
QT5	版本：5.6.1	提供详细的制作文档

驱动源码

驱动名称	描述	备注
RTC	实时时钟驱动	提供源码
CAN	CAN 驱动	提供源码
485	RS485 驱动	提供源码
串口	串口驱动	提供源码
USB HOST	USB HOST 驱动	提供源码
USB OTG	USB 从机驱动	提供源码
LED	GPIO 驱动	提供源码
按键 KEY	GPIO 驱动	提供源码

EMMC/TF 卡	USDHC 驱动	提供源码
EEPROM	AT24C02 驱动	提供源码
音频	WM8960 驱动	提供源码
屏幕	LCD 驱动	提供源码
摄像头	OV5640 驱动	提供源码
WIFI/蓝牙	RTL8723 驱动	提供源码
4G	EC20 驱动	提供源码
六轴传感器	MPU6050 驱动	提供源码
光环境传感器	AP3216C 驱动	提供源码
温湿度传感器	DHT11 驱动	提供源码
ADC 电位器	ADC 驱动	提供源码
红外接收	HS0038B 驱动	提供源码
HDMI	Sil9022A 驱动	提供源码
以太网	以太网驱动	提供源码

资料介绍

C 语 言 基 础 即 可 学 习
资料全部免费提供
关注VX公众号：迅为电子，回复：终结者，免费获取资料
所有教程由迅为原创，深知初学者痛点。从Linux入门到驱动开发再到系统移植，一步一步由浅入深， 逐步掌握嵌入式技术。
1 系统安装 学习Ubuntu系统安装、基础操作以及常用命令的使用
2 编译烧写 搭建开发环境，编译和烧写系统
3 裸机例程 学习裸机例程
4 应用开发 学习Linux应用程序的开发
5 驱动开发 学习Linux驱动程序的开发
6 系统移植 Uboot和Linux设备树驱动开发，驱动的移植和内核裁剪

使用手册
共计1800页 将持续更新
手册从硬件原理分析，搭建Ubuntu开发环境，系统烧写，移植Linux内核，移植Ubuntu，编译Busybox，buildroot，yocto，移植QT，交叉编译QT程序到开发板等，每个例子有详细的注释，每个命令详细的说明， 由迅为工作多年的工程师精心编写。
01 i.MX6ULL终结者开发板初识篇
开发板使用前装配、开发板的所有外设资源以及各个功能的硬件原理介绍，提供的软件资源、各个外设功能的测试方法。
前言 开发板使用前必读
必须注意的问题
一．光盘资料介绍
二 开发板使用前装配流程
第一章 开发板初体验
1.1 初识i.MX6ULL终结者开发板
1.2 i.MX6ULL终结者硬件资源说明
1.3 i.MX6ULL终结者底板原理分析
1.4 i.MX6ULL终结者软件资源
1.5 外设功能验证

02 初识Ubuntu系统篇

Ubuntu系统的安装以及常用的基础操作，通过本章节可以快速熟悉Ubuntu系统的基础使用，为后面的开发打下基础。

- 第二章 Ubuntu系统安装及配置
  - 2.1 虚拟机VMware的安装
  - 2.2 虚拟机安装Ubuntu系统
  - 2.3 Ubuntu系统初体验
  - 2.4 Ubuntu终端和常用命令
  - 2.5 Ubuntu常用命令
  - 2.6 apt-get软件下载工具
  - 2.7 文本编辑工具
  - 2.8 Linux文件系统
  - 2.9 Linux用户权限
- 第三章 Ubuntu下C编程入门
  - 3.1 Linux下编写第一个C程序
  - 3.2 gcc编译器
    - 3.2.1 gcc命令分析
    - 3.2.2 编译警告错误处理
    - 3.2.3 gcc编译流程
  - 3.3 初识Makefile
  - 3.4 Makefile语法

03 开发环境搭建篇

掌握如何搭建i.MX6ULL平台的开发环境。

- 第四章 开发环境搭建详解
  - 4.1 搭建SSH服务
  - 4.2 Windows安装ssh客户端
  - 4.3 搭建交叉编译环境
  - 4.4 SecureCRT 软件安装和使用
  - 4.5 安装CH340串口驱动
  - 4.6 Source Insight 软件安装和使用

04 i.MX6ULL处理器介绍篇

讲解i.MX6ULL处理器的启动方式以及镜像组成格式。

- 第五章 i.MX6ULL启动方式和镜像文件格式
  - 5.1 i.MX6ULL启动方式
  - 5.2 镜像格式
- 第六章 I.MX6ULL镜像烧写
  - 6.1 mfgtool烧写镜像
  - 6.2 屏幕设置

05 i.MX6ULL裸机篇

讲解A7架构的运行模式和16个通用寄存器的功能；掌握基础的ARM汇编指令；通过二十多个裸机实验，快速掌握i.MX6ULL处理器的各个功能模块（包括GPIO，中断，定时器，串口，RTC，I2C，PWM，LCD，触摸等）。

- 第七章 Cortex-A7架构
  - 7.1 Cortex-A7运行模式
  - 7.2 Cortex-A7寄存器组
- 第八章 初识ARM汇编
  - 8.1 GNU汇编语法
  - 8.2 ARM汇编指令
- 第九章 第一个汇编实验
  - 9.1 i.MX6ULL GPIO分析
  - 9.2 IO引脚配置
  - 9.3 GPIO配置
  - 9.4 GPIO时钟
  - 9.5 LED原理分析
  - 9.6 LED汇编程序
  - 9.7 编译LED汇编程序
- 第十章 C语言实现LED例程
  - 10.1 C程序版LED例程简介
  - 10.2 原理图分析
  - 10.3 程序编写
  - 10.4 修改Makefile
  - 17.4 编译及运行
- 第十八章 定时器按键例程
  - 18.1 定时器按键原理
  - 18.2原理分析
  - 18.3 程序设计
  - 18.4 编译及运行
- 第十九章 GPT定时器延时例程
  - 19.1 GPT 定时器简介
  - 19.2 GPT定时器延时原理
  - 19.3 原理分析
  - 19.4 程序设计
  - 16.5 编译及运行
- 第二十章 串口通信例程
  - 20.1 串口简介
  - 20.2 原理分析
  - 20.3 程序设计
  - 20.4 编译及运行
- 第二十一章 串口格式化函数例程
  - 21.1 原理分析

- 10.5 编译烧写验证
- 第十一章 移植官方SDK
  - 11.1 官方SDK简介
  - 11.2 硬件原理图分析
  - 11.3 SDK程序编写
  - 11.4 编译下载
- 第十二章 创建裸机工程
  - 12.1 创建工程目录结构
  - 12.2 原理分析
  - 12.3 程序设计
  - 12.4 编译及运行
- 第十三章 蜂鸣器例程
  - 13.1蜂鸣器原理
  - 13.2原理分析
  - 13.3程序设计
  - 13.4编译及运行
- 第十四章 按键例程
  - 14.1按键输入原理
  - 14.2原理分析
  - 14.3程序设计
  - 14.4编译及运行
- 第十五章 主频和时钟配置例程
  - 15.1 i.MX6ULL时钟系统
  - 15.2原理分析
  - 15.3程序设计
  - 15.4 编译及运行
- 第十六章 GPIO中断例程
  - 16.1中断系统简介
    - 16.1.1 中断向量表
    - 16.1.2 GIC控制器
  - 16.2原理分析
  - 16.3 程序设计
  - 16.4 编译及运行
- 第十七章 EPIT定时器例程
  - 17.1 EPIT定时器介绍
  - 17.2 原理分析
  - 17.3 程序设计
  - 21.2 程序设计
  - 21.3 编译及运行
- 第二十二章 LCD屏幕显示例程
  - 22.1 eLCDIF接口简介
  - 22.2 原理分析
  - 22.3 程序设计
  - 22.4 编译及运行
- 第二十三章 SRTC实时时钟例程
  - 23.1 SRTC简介
  - 23.2 原理分析
  - 23.3 程序设计
  - 23.4 编译及运行
- 第二十四章 I2C操作
  - 24.1 I2C简介
  - 24.2 原理分析
  - 24.3 程序设计
  - 24.4 编译及运行
- 第二十五章 电容触摸实验
  - 25.1 触摸屏简介
  - 25.2 原理分析
  - 25.3 程序设计
  - 25.4 编译及运行
- 第二十六章 屏幕背光调节例程
  - 26.1 背光调节简介
  - 26.2 原理分析
  - 26.3 程序设计
  - 26.4 编译及运行
- 第二十七章 MPU6050 六轴传感器例程
  - 27.1 MPU6050简介
  - 27.2 原理分析
  - 27.3程序设计
  - 27.4 编译运行
- 第二十八章 DHT11温湿度模块例程
  - 28.1 DHT11简介
  - 28.2 原理分析
  - 28.3 程序设计
  - 28.4 编译运行

06 i.MX6ULL镜像编译篇

讲解如何编译i.MX6ULL终结者的镜像（Uboot，Linux）。

- 第二十九章 编译终结者开发板的Uboot和内核
  - 29.1 安装第三方库
  - 29.2 编译终结者开发板Uboot
  - 29.3 编译终结者开发板Linux内核

07 Linux裁剪定制篇

讲解Linux内核的启动流程，如何进行内核的裁剪以及基于Busybox如何制作一个最小文件系统。

- 第三十章Linux内核启动流程
  - 30.1 Linux内核启动（一）：Linux内核自解压过程
  - 30.2 Linux内核启动（二）：ARM处理器相关设置
  - 30.3 Linux内核启动（三）：内核初始化
- 第三十一章linux内核裁剪与定制
  - 31.1 获取Linux源码
  - 31.2 NXP官方Linux源码编译
  - 31.3 在Linux中添加自己的开发板
  - 31.4 添加简单设备驱动
- 第三十二章Linux文件系统的构建
  - 32.1 根文件系统简介
  - 32.2 根文件系统目录介绍
  - 32.3 BusyBox构建根文件系统
  - 32.4 完善根文件系统
- Linux驱动开发

08 Linux驱动实战篇

二十多个驱动实战详细例程（包含字符驱动，设备树分析，定时器，中断，互斥操作，LED子系统，MISC子系统，INPUT子系统，LCD，RTC，串口，I2C，触摸，WIFI，4G等）。

- 第三十三章 字符设备驱动
  - 33.1 字符设备驱动开发
  - 33.2 常规字符设备驱动框架
  - 33.3 简化版字符设备驱动框架
  - 33.4 设备号
  - 33.5 驱动模块的安装与卸载
  - 33.6字符设备驱动实验
  - 33.7 自动创建设备节点
- 第三十四章Linux设备树
  - 34.1 什么是设备树
  - 34.2 DTS、DTC和DTB
  - 34.3 DTS设备树语法结构
  - 34.4 设备树实验
  - 34.5 设备树在系统中的体现
  - 34.6 特殊节点
  - 34.7 DTB文件格式
  - 34.8 binding信息文档
  - 34.9 设备树常用OF操作函数
- 第三十五章pinctrl子系统和GPIO子系统
  - 35.1 pinctrl子系统
  - 35.2 GPIO子系统
- 第三十六章Linux LED实验
  - 36.1 LED设备注册流程
  - 36.2 LED硬件原理图分析
  - 36.3 修改设备树文件
  - 36.4 编写LED驱动程序
  - 36.5 编写应用测试程序
  - 36.6 编译运行测试
- 第三十七章Linux 蜂鸣器实验
  - 37.1 蜂鸣器设备注册流程
  - 37.2 蜂鸣器硬件原理图分析
  - 37.3 修改设备树文件
  - 37.4 编写蜂鸣器驱动程序
  - 37.5 编写应用测试程序
  - 37.6 编译运行测试
- 第三十八章Linux并发与竞争
  - 38.1 并发与竞争
  - 38.2 中断屏蔽
  - 38.3 原子操作
  - 38.4 自旋锁
  - 38.5 信号量
  - 38.6 互斥体
- 第三十九章Linux并发与竞争实验
  - 39.1 原子操作
  - 39.2 自旋锁实验
  - 39.3 信号量实验
  - 39.4 互斥体实验
- 第四十章Linux按键输入实验
  - 40.1 Linux下按键驱动原理
  - 40.2 按键硬件原理图分析
  - 40.3 修改设备树文件
  - 40.4 按键驱动程序
  - 40.5 应用测试程序
  - 40.6 运行测试
  - 43.3 非阻塞IO实验
- 第四十四章 Linux异步通知实验
  - 44.1 异步通知
  - 44.2 编写实验程序
- 第四十五章 platform设备驱动
  - 45.1 Linux设备驱动分层和分离
  - 45.2 platform平台驱动模型
  - 45.3 实验程序
  - 45.4 运行测试
- 第四十六章 设备树下的platform驱动
  - 46.1 设备树下的platform设备
  - 46.2 实验程序编写
  - 46.3 运行测试
- 第四十七章 Linux内核自带LED灯驱动
  - 47.1 Linux内核中的LED灯驱动
  - 47.2 Linux自带LED驱动分析
  - 47.3 添加设备树节点
  - 47.4 运行测试
- 第四十八章 Linux MISC驱动
  - 48.1 misc设备驱动简介
  - 48.2 编写实验程序
  - 48.3 运行测试
- 第四十九章 Linux INPUT子系统实验
  - 49.1 input子系统
  - 49.2 编写实验程序
  - 49.3 运行测试
  - 49.4 linux自带按键驱动程序
- 第五十章 Linux LCD驱动实验
  - 50.1 Linux下LCD驱动简介
  - 50.2 编写LCD驱动程序
  - 50.3 运行测试
- 第五十一章 Linux RTC驱动实验
  - 51.1 Linux内核RTC驱动简介
  - 51.2 i.MX6UL下的RTC驱动
  - 51.3 Linux系统下RTC时间设置与查看
- 第五十二章 Linux I2C驱动实验
  - 52.1 Linux I2C驱动框架简介
  - 52.2 imx6ull的I2C总线驱动分析
  - 52.3 I2C设备驱动编写流程
  - 52.4 实验程序编写
  - 52.5 运行测试
- 第五十三章 Linux RS232/485驱动实验
  - 53.1 Linux下的UART驱动框架
  - 53.2 i.MX6UL UART驱动分析
  - 53.3 硬件原理图
  - 53.4 RS232驱动
  - 53.5 RS232驱动测试
  - 53.6 RS485测试
- 第五十四章 Linux 电容触摸屏实验
  - 54.1 Linux下电容触摸屏驱动框架
  - 54.2 硬件原理图
  - 54.3 实验程序编写
  - 54.4 运行测试
  - 54.5 tslib移植和使用

- 第四十一章Linux内核定时器实验
  - 41.1 Linux时间管理和内核定时器简介
  - 41.2 实验程序编写
  - 41.3 运行测试
- 第四十二章 Linux中断实验
  - 42.1 Linux中断简介
  - 42.2 实验程序编写
  - 42.3 运行测试
- 第四十三章 Linux阻塞和非阻塞IO实验
  - 43.1 阻塞和非阻塞IO
  - 43.2 阻塞IO实验
  - 54.6 使用Linux内核自带的ft5426驱动
- 第五十五章 Linux WIF驱动实验
  - 55.1 wif驱动添加和编译
  - 55.2 wireless tools 工具移植与测试
  - 55.3 wpa_supplicant 移植
  - 55.4 rtl8723 wif联网测试
- 第五十六章 Linux 4G通信实验
  - 56.1 4G网络连接
  - 56.2 EC20 4G模块配置
  - 56.3 EC20 4G模块联网测试

09 文件系统篇

讲解Linux文件系统的构成，目录结构、通过Buildroot制作Linux文件系统、使用Yocto构建文件系统；制作Ubuntu文件系统。

- 第五十七章 Linux系统开发
  - 57.1 根文件系统简介
  - 57.2 制作最小文件系统镜像
  - 57.3 最小Linux系统功能测试
  - 57.4 扩展文档
- 第五十八章 Buildoot文件系统构建篇
  - 58.1 buildroot简介
  - 58.2 buildroot构建文件系统
  - 58.3 buildroot文件系统测试
  - 58.4 buildroot添加支持第三方软件
  - 58.5 buildroot下配置busybox
- 第五十九章 Yocto文件系统构建篇
  - 59.1 注意事项
  - 59.2 在线构建文件系统
  - 59.3 离线构建文件系统
- 第六十章 Ubuntu文件系统
  - 60.1 ubuntu-base获取
  - 60.2 ubuntu-base构建
  - 60.3 Ubuntu文件系统测试
    - 60.3.1 打包烧写Ubuntu
    - 60.3.2 Ubuntu 根文件系统测试
- 第六十一章 Debian文件系统的构建

10 QT应用开发篇

讲解QT开发环境的搭建，创建第一个QT工程并如何编译运行。

- 第六十二章 QT应用开发
  - 62.1 c++入门
  - 62.2 Qt基础
  - 62.3 Qt系统移植
  - 62.4 Qt下控制硬件
  - 62.5 Qt for Android
  - 62.6 一键烧写QT程序到开发板
- 第六十三章 QT实战项目
  - 63.1 DHT11&网络编程实战练习
- 第六十四章 使用Yocto文件系统开发QT
  - 64.1 安装测试Qt Creator
  - 64.2 安装交叉编译工具及QT
  - 64.3 搭建IMX6UI/ULL QT Creator交叉编译环境
  - 64.4 QT工程的编译及测试
  - 64.5 文件系统

11 Linux系统开发进阶篇

- 第六十五章 Mfgtools修改单独只烧写Uboot，内核，文件
- 第六十六章 iTOP-i.MX6ULL开发板-修改开机进度条修改文
- 第六十七章 iTOP-i.MX6ULL开发板-Uboot-Logo修改文档
- 第六十八章 iTOP-i.MX6ULL开发板-内核-Logo 修改使用文
- 第六十九章 iTOP-i.MX6ULL开发板-移植Gdb+Gdbserver调
- 第七十章 iTOP-iMX6ULL开发板-移植ssh服务
- 第七十一章 iTOP-i.MX6ULL开发板-Jtag调试
- 第七十二章 iTOP-i.MX6ULL开发板-网络通信-套字节TCP
- 第七十三章 iTOP-i.MX6ULL开发板-网络通信-套字节UDP
- 第七十四章 iTOP-i.MX6ULL开发板-Tftp烧写和Nfs启动
- 第七十五章 iTOP-i.MX6ULL开发板-搭建Web服务器
- 第七十六章 iTOP-i.MX6ULL开发板-Modbus移植和使用
- 第七十七章 iTOP-i.MX6ULL开发板-USB转串口驱动Ch34X
- 第七十八章 iTOP-i.MX6ULL开发板-Sqlite3移植和使用
- 第七十九章 iTOP-i.MX6ULL开发板-Qt5.7Mplayer移植文档
- 第八十章 iTOP-i.MX6ULL开发板-移植OpenCv2.4.9
- 第八十一章 iTOP-i.MX6ULL开发板-移植OpenCv3.4.1
- 第八十二章 iTOP-i.MX6ULL开发板-网络测试方法
- 第八十三章 iTOP-i.MX6ULL开发板-Mqtt 移植教程
- 第八十四章 iTOP-i.MX6ULL开发板-Busybox移植DHCP
- 第八十五章 iTOP-i.MX6ULL开发板-C程序调用shell
- 第八十六章 iTOP-iMX6ULL开发板-八路串口配置

光盘资料

01 开发及烧写工具

交叉编译器，裸机镜像制作工具，i.MX6ULL的官方下载器，SDK包。

- 1.交叉编译器
- 2.裸机镜像制作工具
- 3.mfgtools_for_6ULL
- 4.i.MX6ULL SDK包

02 开发所需软件

虚拟机，Ubuntu系统镜像，驱动及各种调试软件等工具。

- 盘符格式化工具
- SSH
- SecureCRT7.1
- Qt creator中文插件
- Qt Creator
- CH340驱动(USB串口驱动)
- winhexv19.7.zip
- VMware-workstation-full-15.5.0-14665864.exe
- ubuntu-16.04.6-desktop-amd64.iso
- SSHSecureShellClient-3.2.9.exe
- sourceinsight40107-setup.exe
- SDFormatter(内存卡修复工具).exe
- MobaXterm_Installer_v12.3.zip
- hyperterminal-v2.4.8.zip
- FileZilla_3.39.0_win64-setup_bundled.exe

03 开发板硬件资料

芯片资料，底板的原理图，PCB图。核心板原理图。

- 01_开发板原理图 ---------->>
- 核心板原理图
- 底板原理图
- 02_芯片资料 ---------->>
- MPU6050资料
- FT5426(电容触摸屏IC)
- WM8960_v4.2.pdf
- UM805RE.pdf
- UM803RS.pdf
- TJA1040.pdf
- SY6280AAC.pdf
- SP3485EN.pdf
- SMF05C.pdf
- MIC5319-2.8YML.pdf
- L9014.pdf
- KSZ8081RNB.pdf
- JW5060T.pdf
- HS0038B.pdf
- HR911105A.pdf
- GT9147数据手册.pdf
- GT9147编程指南.pdf
- GM8285C数据手册3.1（20130707）.pdf

- Sil9022A编程手册.pdf
- SiI9022A9024A-DS-1076-C01.PDF
- RT9169-33GVL.pdf
- RT9013-18GB.pdf
- RT8065.pdf
- Quectel_EC20 R2.0_Mini_PCIe-C(Audio喉掛)_芭瑞挖寶
- OV5640_CSP3_DS_2.01_Ruisipusheng.pdf
- mt41k128m16-micron technology.pdf
- MPU6050.pdf
- MP2143DJ.pdf
- FE1.1S.pdf
- DS_K4B4G1646E-BY(M)_Rev1.1.pdf
- DHT11.pdf
- CH340G.pdf
- AT24C02.pdf
- AP3216C.pdf
- AO3416.pdf
- AO3415A.pdf
- 1、I.MX6UL芯片资料.zip
- 1、I.MX6ULL芯片资料.zip

04 裸机例程源码

leds，蜂鸣器，按键，ADC，UART等20个例程。

- 1_leds
- 2_led_C program
- 3_led_sdk
- 4_led_bsp
- 5_beep
- 6_key
- 7_clk
- 8_int
- 9_epit_timer
- 10_key_filter
- 11_gpt_delay
- 12_uart
- 13_printf
- 14_lcd
- 15_rtc
- 16_i2c
- 17_touchscreen
- 18_backlight
- 19_mpu6050
- 20_dht11

05 uboot linux源码

提供NXP官方原版源码和迅为修改好的Uboot和Linux源码。

- 终结者Uboot和Linux源码.zip
- NXP官方原版Uboot和Linux.zip

06 Linux驱动例程

GPIO，蜂鸣器，串口等20个例程及第三方库源码。

- 01_chartest
- 02_gpioled
- 03_beep
- 12_key_poll
- 13_key_signal
- 14_gpioled_platform

- 04_gpioled_atomic
- 05_gpioled_spinlock
- 06_gpioled_semaphore
- 07_gpioled_mutex
- 08_key
- 09_gpioled_timer
- 10_key_irq
- 11_key_waitqueue
- 15_gpioled_dts
- 16_beep_misc
- 17_key_input
- 18_ap3216
- 19_uart
- 20_multitouch
- 第三方库源码

07 文件系统源码

Busybox，buildroot，yocto，Ubuntut文件系统源码。

- 01_Busybox文件系统
- 02_Buildroot文件系统
- 03_Yocto文件系统
- 04_Ubuntu文件系统

08 开发板系统镜像

Uboot，Linux内核镜像，文件系统镜像。

- 01_Uboot
- 02_Linux内核镜像
- 03_文件系统镜像

09 QT综合例程

Qt应用例程。

- topeet_qt_demo_v1_0.tar.gz

10 其它参考资料

NXP官方DDR初始化与测试工具，格式化函数库及学习资料。

- 格式化函数库
- NXP官方DDR初始化与测试工具
- 怎样读Linux内核源代码.pdf
- 跟我一起写Makefile.pdf
- 多核编程之挑战与应对—孔晓华_计算机系统结构.ppt
- Uboot启动过程详解.pdf
- Linux系统命令及其使用详解.pdf
- Linux设备驱动开发详解.pdf
- Linux内核源代码(free).ppt
- Linux内核设计与实现_第三版_中文版.pdf
- Linux内核Makefile.pdf
- linuxC编程一站式学习.pdf
- C语言嵌入式系统编程修炼.pdf
- Cortex-A7 Technical ReferenceManua.pdf
- ARM入门宝典.pdf
- ARM开发工程师入门宝典.pdf
- ARM Cortex-A(armV7)编程手册V4.0.pdf
- ARM ArchitectureReference Manual ARMv7-A and

视频教程

视频精简，思路清晰，不做无用功
让所学每一个技术点都能真真切切落实到项目中，学即所用。
B站搜索'北京迅为'，免费观看视频教程 [https://space.bilibili.com/33785640]

01 嵌入式学习Linux学习方法篇

第二讲 事半功倍之Linux系统框架.mp4

第三讲 事半功倍之Linux应用程序编程框架.mp4

第四讲 事半功倍之Linux驱动架构的演进.mp4

第一讲 事半功倍之Linux框架学习法.mp4

02 嵌入式学习之Linux裸机篇

第1节 ARM体系结构简介.mp4

第2节 使用C语言按照GPIO操作流程点亮LED灯I.MX6ULL芯片...

第3节 初识GPIO以及GPIO操作流程.mp4

第4节 使用汇编按照GPIO操作流程点亮LED灯.mp4

第5节 掌握裸机程序的编译烧写以及测试流程.mp4

第6节 使用C语言按照GPIO操作流程点亮LED灯.mp4

第7节 掌握C程序的编译烧写以及测试流程.mp4

第8节 了解NXP提供的资料并使用官方SDK.mp4

第9节 在官方SDK的基础上构建一个通用裸机工程.mp4

第10节 用蜂鸣器学习GPIO输出功能.mp4

03 嵌入式学习之Linux入门篇

01-i.MX6ULL终结者开发板开箱初体验.mp4

02-i.MX6ULL终结者资料介绍.mp4

03-安装VMware Workstation Pro软件.mp4

04-获取并安装Ubuntu操作系统.mp4

05-Ubuntu系统介绍.mp4

06-Ubuntu启用root用户.mp4

07-Ubuntu使用apt-get下载.mp4

08-vim编辑器的使用.mp4

09-Ubuntu常用命令第一部分.mp4

10-Linux相对路径和路径.mp4

11-Linux家目录和根目录概念.mp4

12-Linux常用命令第二部分.mp4

13-Linux帮助手册讲解.mp4

14-Linux权限管理.mp4

15-Linux连接档概念.mp4

16-Linux目录结构讲解.mp4

17-Linux文件系统概念.mp4

18-Linux第一个程序

19-Linux环境变量讲解.mp4

20-编写第一个自己的Linux的命令.mp4

21-Linux工具之make工具和makefile文件.mp4

22-makefile基本语法(上).mp4

23-makefile基本语法(下).mp4

24-安装VMwareTool工具.mp4

25-SSH软件的安装和使用.mp4

26-samba的安装和使用.mp4

27-Source Insight的安装和使用.mp4

28-超级终端的安装和使用.mp4

29-交叉编译器的安装和使用.mp4

30-开发板固件烧录.mp4

31-借助U盘或者TF卡拷贝到程序到开发板.mp4

32-nfs服务器的搭建和使用.mp4

33-点亮第一个led灯.mp4

04 嵌入式学习之QT进阶篇

从 Windows 上位机开发，到 Linux 界面开发，再到手机 APP 开发，让你领略 QT 跨平台的魅力。

第1节-本期QT视频将会收获哪些知识？.mp4

第2节-在Windows上搭建QT开发环境.mp4

第3节-C++基础（上）.mp4

第4节-C++基础（下）.mp4

第5节-制作一个最简单的QT登录界面.mp4

第6节-QT信号和槽.mp4

第7节-仿写一个智能家居界面（上）.mp4

第8节-仿写一个智能家居界面（中）.mp4

第9节-仿写一个智能家居界面（下）.mp4

第10节-QT上位机开发之串口助手（上）.mp4

第11节-QT上位机开发之串口助手（下）.mp4

第12节-把QT程序打包成Windows软件.mp4

第13节-QT网络编程之TCP通信.mp4

第14节-QT网络编程之UDP通信.mp4

第15节-QT时间编程之QT时钟.mp4

第16节-QT系统移植之构建根文件系统.mp4

第17节-QT系统移植之交叉编译QT源码和触摸源码.mp4

第18节-QT系统移植之在开发板上部署QT运行环境.mp4

第19节-QT跨平台运行之把QT程序交叉编译到ARM开发板.mp4

第20节-QT跨平台运行之使用QT控制蜂鸣器.mp4

第21节-QT跨平台运行之把QT程序编译成手机APP.mp4

资料获取与后续更新

一. 资料的获取

(1) 百度网盘下载

方式一：您只需要关注 VX 公众号：迅为电子 ， 回复 ：终结者，免费获取产品资料

方式二：网盘的链接在购买开发板后可以在迅为电子技术支持 QQ 群下载。如果链接有更新，会在群里贴通告

(2) 视频配套教程

方式一： B 站搜‘北京迅为’， 入门视频教程免费学习，持续更新。

方式二：根据用户购买开发板的产品类别，迅为电子会提供相应的光盘。

二 后续更新

后续文档、视频等资料的更新，为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。关注“迅为电子”微信公众号，不定期分享教程、资料和行业干货及产品一线资料。

迅为电子

让学习更容易，让开发更简单

官方微信公众号

迅为商城

北京迅为电子有限公司

地址：北京市海淀区永丰产业基地丰慧中路 7 号新材料创业大厦

销售热线：010-58957586

售后服务热线：0312-6796610

开发板官网：http://topeetboard.com

方案官网：http://topeet.com