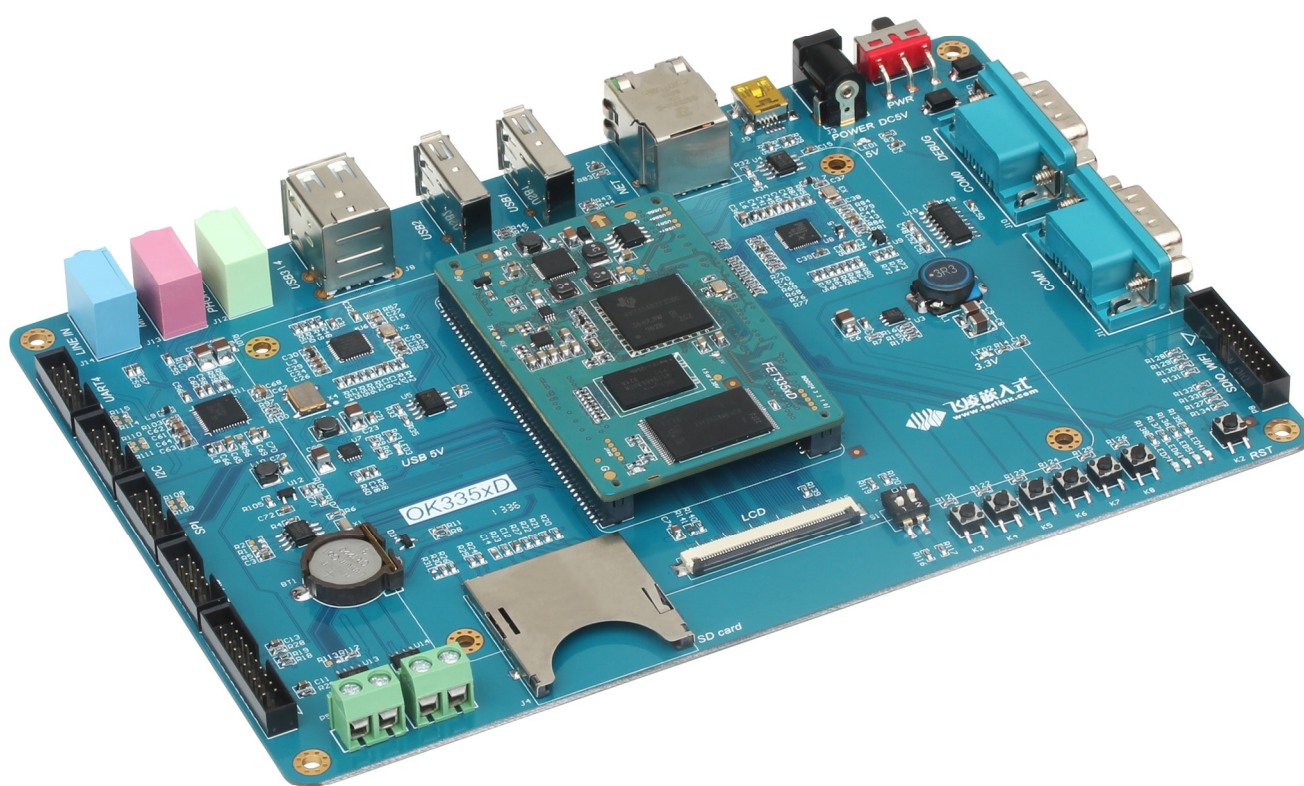


OK335xD

硬件手册



Devoted to create the best embedded products

注意事项与维护

产品使用环境

供电电压: DC5V \pm 10%
运行温度: 商用级 0 to + 70℃
工业级 -40 to + 85℃
湿度: 5% to 95%RH (不结露)



注意事项

- **请勿带电插拔核心板及外围模块！**

- 请遵循所有标注在产品上的警示和指引信息。
- 请保持本产品干燥。如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干。
- 使用中注意本产品的通风散热，避免温度过高造成元器件损坏。
- 请勿在多尘、脏乱的环境中使用或存放本产品。
- 请勿将本产品应用在冷热交替环境中，避免结露损坏元器件。
- 请勿粗暴对待本产品，跌落、敲打或剧烈晃动都可能损坏线路及元器件。
- 请勿使用有机溶剂或腐蚀性液体清洗本产品。
- 请勿自行修理、拆卸本公司产品，如产品出现故障请及时联系本公司进行维修。
- 擅自修改或使用未经授权的配件可能损坏本产品，由此造成的损坏将不予以保修。

如果产品出现故障，请联系飞凌技术服务部。

版权声明

本手册所有权由保定市飞凌嵌入式有限公司独家持有。未经本公司的书面许可，任何单位和个人无权以任何形式复制、传播、转载本手册的任何部分，否则一切后果由违者自负。

更新记录

日期	版本	更新内容
2014.03.26	V1.1	1.添加核心板引脚默认定义的硬性规定说明 2.OK335底板部分改动说明 3.添加实测功耗表
2013.08.03	V1.0	新文件导入

技术支持与更新

1 技术支持范围

- 1.1 本公司产品的软、硬件资源提供情况咨询；
- 1.2 本公司产品的软、硬件手册使用过程中遇到的问题；
- 1.3 本公司提供的 OEM、ODM 售后技术支持；
- 1.4 已购买本公司产品用户的资料丢失、更新后重新获取；
- 1.5 本公司产品的故障判断及售后维修服务；

2 技术讨论范围（非必解决问题）

- 2.1 源码的修改以及理解；
- 2.2 操作系统如何移植；
- 2.3 用户在自行修改以及开发中遇到的软硬件问题；

3 技术支持方式

- 3.1 电话（即时）：0312-3119192
- 3.2 邮箱（非即时）：
 - Linux 技术支持：linux@forlinx.com
 - Win CE 技术支持：wince@forlinx.com
 - Android 技术支持：android@forlinx.com
- 3.3 论坛（非即时）：<http://bbs.witech.com.cn>

4 技术支持时间

周一至周五：

上午	9:00—11:30
下午	13:30—17:00

公司按照国家法定节假日安排休息，在此期间无法提供技术支持，有问题请发邮箱或论坛技术支持区，我们会在工作日尽快给您回复。

5 资料下载方法：

请登陆“bbs.witech.com.cn”找到“[开发板资料下载](#)”选择对应平台下载

目 录

注意事项与维护	1
技术支持与更新	4
第一章 FET335xD核心板简介	7
1.1 FET335xD核心板概述	7
1.2 FET335xD系统框图	8
1.3 CPU功能框图	9
1.4 应用领域	9
第二章 FET335xD核心板说明	10
2.1 FET335xD核心处理器及电源管理IC	10
2.1.1 AM3354处理器特性	10
2.1.2 TPS65217电源管理芯片特性	12
2.2 FET335xD核心板结构图	13
2.2.1 FET335xD核心板外观图	13
2.2.2 FET335xD核心板结构尺寸图	13
2.3 FET335xD核心板性能参数	14
2.3.1 核心板配置资源	14
2.3.2 供电模式	14
2.3.3 测试功耗	14
2.3.4 工作环境测试	15
2.4 FET335xD核心板默认引脚定义图	16
2.5 FET335xD核心板引脚功能参考说明	18
2.5.1 FET335xD核心板默认引脚定义表	18
2.5.2 FET335xD核心板电源接口	23
2.5.3 FET335xD核心板系统控制接口	23
2.5.4 boot配置引脚	23
2.6 FET335xD核心板默认功能说明	28
2.6.1 通用存储控制器(GPMC)总线	28
2.6.2 LCD控制器	29
2.6.3 MAC控制器	30
2.6.4 通用串行接口(USB)	30
2.6.5 ADC控制器	31
2.6.6 测试仿真接口(JTAG)	31
2.6.7 MMC / SDIO 接口	31
2.6.8 I2C 接口	32
2.6.9 SPI 接口	32
2.6.10 UART 接口	32

2.6.11	485芯片接口	32
2.6.12	CAN接口	32
2.6.13	MCASP 接口	32
第三章	OK335xD底板说明与硬件设计指导	34
3.1	OK335xD底板及接口外围资源简述	34
3.1.1	OK335xD底板接口分布图	34
3.1.2	OK335xD底板布局丝印图	35
3.1.3	OK335xD底板功能列表	35
3.2	OK335xD底板功能详细介绍与硬件设计指导	36
3.2.1	电源插座和开关	36
3.2.2	boot配置引脚	36
3.2.3	核心板连接器	37
3.2.4	LCD液晶显示接口	38
3.2.5	SD卡	41
3.2.6	板载用户按键	43
3.2.7	演示LED	44
3.2.8	ADC模数转换	44
3.2.9	音频接口	45
3.2.10	UART串口	46
3.2.11	RS-485接口	47
3.2.12	USB 2.0 HOST接口	47
3.2.13	USB 2.0 OTG接口	48
3.2.14	SPI 接口	48
3.2.15	I2C接口	49
3.2.16	CAN接口	50
3.2.17	SDIO WiFi接口	50
3.2.18	千兆以太网	51
3.2.19	板载实时时钟	52
3.2.20	JTAG仿真调试接口	53
附录表		55
F1	FET335xD核心板引脚全功能定义	55

第一章 FET335xD核心板简介

1.1 FET335xD核心板概述

FET335xD核心板模块隶属于飞凌嵌入式技术有限公司推出的面向工业级开发平台的产品系列，旨在为更多的企业用户提供领先一步的嵌入式产品，以便于用户产品升级过程中缩短研发周期、节省技术成本，更快更优质的开发出自己的工业级产品。同时，FET335xD核心板模块搭载飞凌公司的OK335xD评估底板，即可用作教学开发平台。对于需要了解学习AM335x系列芯片、Cortex-A8架构及开发环境的用户，建议在购买FET335xD核心板的同时，购买OK335xD评估底板和配套的7吋电容屏或电阻屏。

FET335xD基于TI公司Sitara系列的ARM处理器AM335x，在设计上采用高集成度的系统模块(SOM: System On Module)形式的核心板，集成了容量512MB，频率800Mhz的DDR3内存颗粒，容量 256MB 的SLC Nandflash存储芯片，和TI公司专为AM335X处理器匹配设计的电源管理芯片，另外核心板模块上还添加了硬件看门狗，从而使系统更稳定运行。

核心板采用的Sitara系列ARM处理器AM3354（默认处理器），是一款高性能的Cortex-A8架构工业级处理器，其最高运行频率可达1GHz，采用45nm制程工艺，集成了基于ARM Cortex-A8的微处理器单元(MPU)，POWERVR SGX™ 图形加速器（用于3D 图形加速以支持显示和游戏效果），可编程实时单元和工业用通信子系统(PRU-ICSS)（从ARM 内核分离，从而实现了在传输和控制中配置灵活并拥有更高效率的可能），保证系统在低功耗运行的同时拥有高性能。

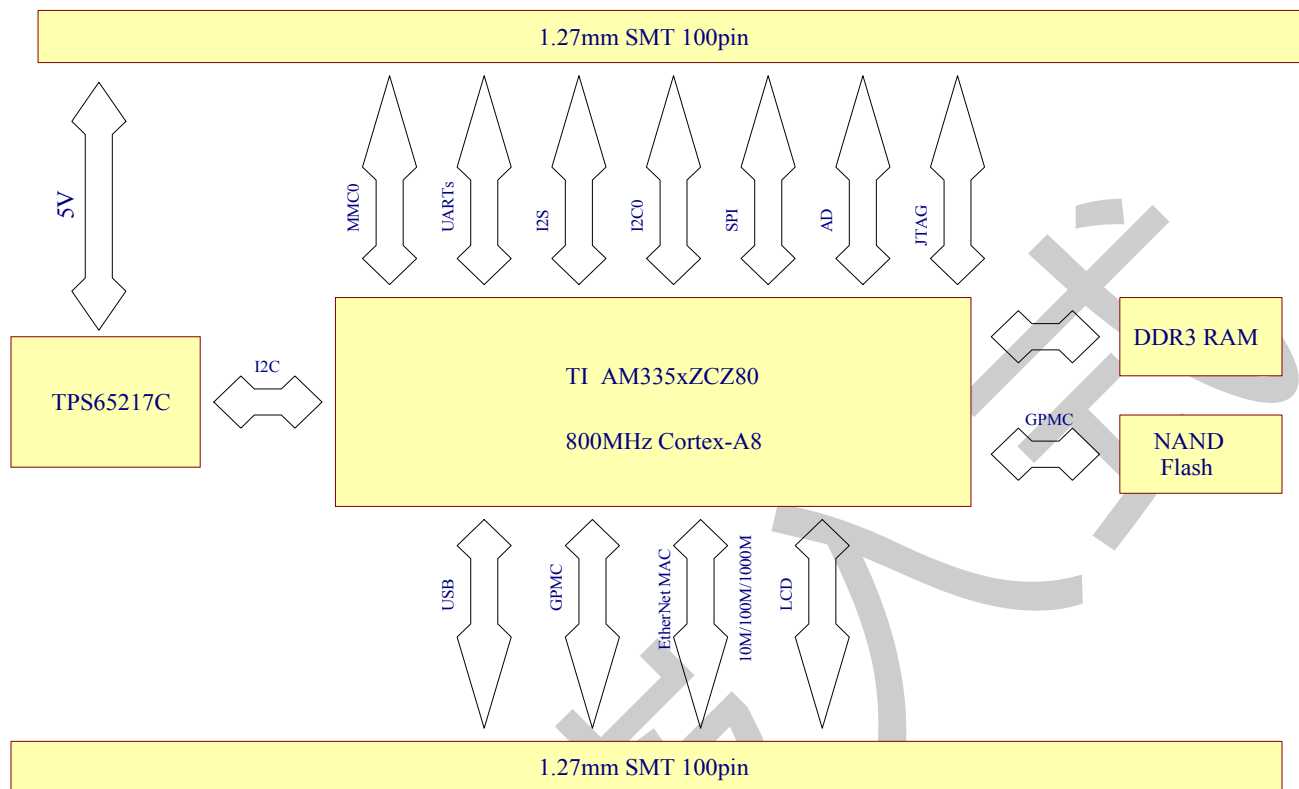
模块为8层PCB层叠结构，沉金工艺设计，连接方式采用1.27mm间距，2x50 pin双排插针(两组)结构。

FET335xD核心板功能简述：

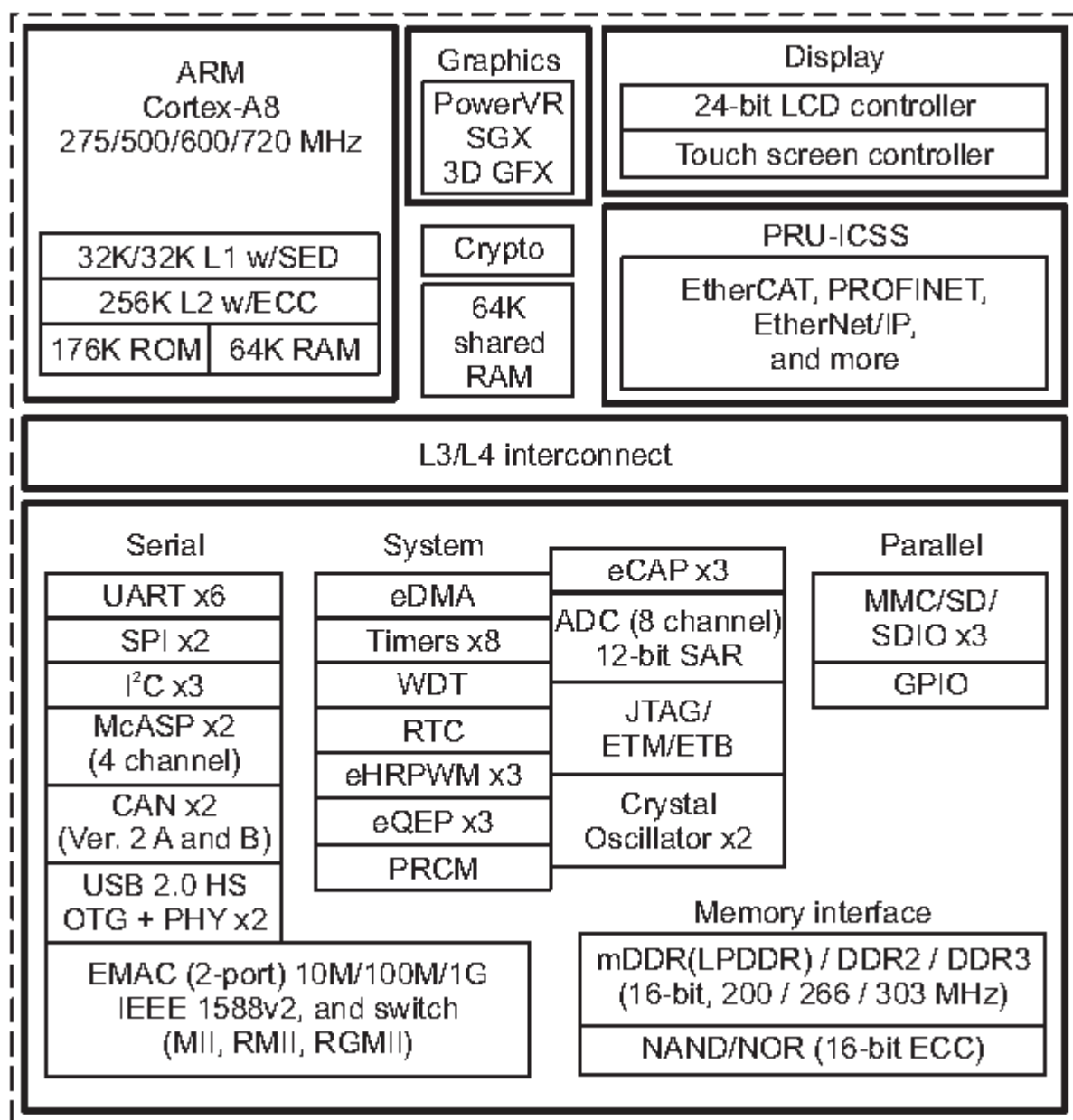
● RGB 16位/24位 LCD接口
● 系统总线 GPMC （8/16位数据 12位地址）
● 多路UARTs
● CAN接口
● 10/100/1000M RGMII/MII/RMII 接口
● USB2.0
● JTAG 仿真调试接口
● I2C接口
● SPI接口
● McASP 多通道音频串行接口
● ADC接口
● SD/MMC接口
● GPIOs
● 板载电源管理IC

注：因AM335x处理器功能强大，有些功能需要复用芯片引脚，故部分外围功能不能同时使用，设计模块时已做调配，大部分功能都不会受到影响。感兴趣的用户可以了解一下TI德州仪器的Pin Mux概念。

1.2 FET335xD系统框图



1.3 CPU功能框图



1.4 应用领域

适用于便携式数据终端、导航、游戏外设、智能家居、工业自动化、消费类医疗器械、打印机智能收费系统、测试和测量设备、教育控制台、高级玩具和单板计算机等多种领域。

第二章 FET335xD核心板说明

2.1 FET335xD核心处理器及电源管理IC

FET335xD模块的核心处理器采用了TI公司的AM335x系列中的AM3354，这款Cortex-A8架构的工业级CPU具有很高的性能。另外与之匹配设计的电源管理芯片是TPS65217，两个互相匹配设计的芯片协同作用可以使系统可以以更高的性能和更低的功耗运行。

2.1.1 AM3354处理器特性

- 275MHz, 500MHz, 600MHz, 720MHz 或 800MHz ARM Cortex™-A8 32位精简指令集(RISC)微处理器
 - o NEON™ 单指令流多数据流(SIMD)协处理器
 - o 具有单错检测（奇偶校验）的32KB L1指令高速缓存和32KB 数据高速缓存
 - o 具有错误纠正码(ECC) 的256KB L2高速缓存
- 支持移动双倍速率同步动态随机存储器(DDR[LPDDR]/DDR2/DDR3/DDR3L), 通用存储器(NAND, NOR, SRAM, 等)
 - o mDDR: 200MHz 时钟（400MHz 数据速率）
 - o DDR2: 266MHz 时钟（532MHz 数据速率）
 - o DDR3: 400MHz 时钟（800MHz 数据速率）
 - o 16位数据总线
 - o 支持7种存储芯片（NAND, NOR, Muxed-NOR, SRAM, 等）的8/16位异步存储器接口
 - o 使用BCH 码以支持4 位、8 位或者16位ECC
- SGX530 图形引擎
 - o 图像纹理架构最高传送速率为每秒20MPloy
 - o 通用可扩展着色引擎(包含像素和顶点着色功能性的多线程引擎)
 - o 支持Microsoft VS3.0, PS3.0 和OpenGL2.0 的高级着色程序特性集
 - o 支持Direct3D, OpenGL-ES 1.1 和2.0, OpenVG 1.0, 和OpenMax 的工业标准API 支持
 - o 高密度任务切换、负载均衡和电源管理
 - o 占用低CPU资源以达到交互的高级几何DMA驱动
 - o 可编程高质量图像防锯齿
 - o 用于统一存储器架构中操作系统运行的完全虚拟存储器寻址
- LCD 控制器
 - o 最高24位数据输出；每像素8位(RGB)
 - o 分辨率高达2048x2048 （具有最大频率126MHz 像素时钟）
 - o 集成LCD 接口显示驱动器(LIDD) 控制器
 - o 集成光栅控制器
 - o 集成DMA控制器,可凭借中断或者一个定时器从外部帧缓冲器中读取数据而不会增加处理器的负担
 - o 512字内部FIFO
 - o 支持的显示类型:
 - 字符显示-使用LCD 接口显示驱动器(LIDD) 控制器来为这些显示编程序
 - 无源矩阵LCD 显示-使用LCD 光栅显示控制器来为到无源显示的持续图形刷新供定时和数据
 - 有源矩阵LCD 显示-使用外部帧缓冲器空间和内部DMA控制器来驱动到控制面板的流数据
- 12位逐次逼近寄存器(SAR)ADC
 - o 每秒100K的采样速率

- o 可被配置为4线、5线、或者8线电阻式触摸屏控制器(TSC) 接口
- 可编程实时单元和工业用通信子系统(PRU-ICSS) (注: AM3354以上版本支持此项)
 - o 支持的协议包括EtherCAT®, PROFIBUS, PROFINET, Ether Net/IP™
 - o PRU-ICSS 内的外设
 - o 1个具有流量控制引脚的UART 端口, 支持的速率高达12Mbps
 - o 2个支持工业用以太网的MII 以太网端口,
 - o 1个MDIO 端口
 - o 1个增强型捕捉(eCAP)模块
- 电源复位和时钟管理(PRCM)模块
 - o 控制进入和离开待机和深度睡眠模式
 - o 负责睡眠排序, 电源域切换排序, 唤醒排序和电源域打开排序
- 实时时钟(RTC)
 - o 实时日期(日期/月份/年份)和时间(小时/分钟/秒)信息
 - o 内部集成32.768kHz晶体振荡器, RTC 逻辑控制器和1.1V 内部低压降稳压器(LDO)
 - o 独立的加电复位(RTC_PWRONRSTn) 输入引脚
 - o 用于外部唤醒事件的专用输入引脚(EXT_WAKEUP)
 - o 可编程警报器可被用于生成到PRCM(用于唤醒)或者Cortex-A8(用于事件通知)的内部中断
 - o 可编程警报器可与外部输出(PMIC_POWER_EN)一起使用以使电源管理IC能够恢复非RTC电源
- 外设
 - o 多达2个具有集成PHY 的USB 2.0 高速OTG端口
 - o 多达2个工业用千兆以太网MAC(10/100/1000Mbps)
 - 集成开关
 - 每个MAC支持MII/RMII/RGMII
 - 以太网MAC可独立运行其它功能
 - IEEE 1588v2 精准时间协议(PTP)
 - o 多达2个控制器局域网(CAN) 端口
 - 支持CAN 2.0版本
 - o 多达2个多通道音频串口(McASP)
 - 传输/接收时钟频率高达50 MHz
 - 带有独立TX/RX 时钟的McASP端口.
 - 支持时分复用(TDM)、I2S
 - 支持数字音频接口传输 (SPDIF, IEC60958- 1, 和AES-3格式)
 - 用于发送和接收的FIFO缓冲器 (256 字节)
 - o 最多6个UART
 - 所有UART支持IrDA和CIR模式
 - 所有UART支持RTS和CTS流量控制
 - UART1支持完全调制解调器(Modem)控制
 - o 最多2个主/从McSPI接口
 - 最多2个芯片选择
 - 最高48 MHz数据速率
 - o 最多3个MMC/SD/SDIO 端口
 - 1 位, 4 位和8 位MMC/SD/SDIO 模式
 - MMCS0 具有针对1.8V 或者3.3V 的运行的专用电源
 - 高达48MHz 的数据传输速率

- 支持卡检测和写保护
- 与MMC4.3和SD/SDIO 2.0规范兼容
- o 多达3个I2C主/从接口
 - 标准模式（高达100kHz）
 - 快速模式（高达400kHz）
- o 多达4 通道的通用IO (GPIO)
 - 每通道32个GPIO（与其它功能引脚复用）
 - GPIO 可被用作中断输入（每通道高达2 个中断输入）
 - 多达3个外部DMA 事件输入，此输入也可被用作中断输入
 - 8个32 位通用定时器
- 支持调试接口
 - o 支持器件边界扫描
 - o 支持IEEE1500
- DMA
 - o 片载增强型DMA 控制器（EDMA）具有3个第三方传送控制器(TPTC) 和1个第三方通道控制器 (TPCC)，此控制器最多支持64个可编程逻辑通道和8个QDMA通道。

2.1.2 TPS65217电源管理芯片特性

TPS65217 是一款单芯片电源管理 IC，设计用于 AM335x 系列应用处理器。支持单节锂离子电池和锂聚合物电池。

TPS65217 为双路径输入电源芯片，拥有 3 个降压 DCDC，4 个 LDO 并提供一路线性电池充电接口，还提供有一个高效升压转换器可以对 2 串 10 并结构的 LED 供电。TPS65217 可由 USB 端口、5V 电源适配器或锂离子电池的任意组合方式供电。额定运行温度范围为 -40°C 至 $+105^{\circ}\text{C}$ ，适合于工业应用。

TPS65217 拥有的 3 个高效的降压 DCDC 分别为处理器的核心，内存控制器和 I/O 提供电压。

TPS65217 在轻载时会进入低功率模式。对于低噪声场合，通过使用 I²C 接口设置使 TPS65217 的 DCDC 可以工作在固定的某一个低频率上。

4 个 LDO 中，LDO1 和 LDO2 是为了支持系统待机模式。在睡眠状态下，LDO1 和 LDO2 输出电流被限制在 100 μA 以降低静态电流。在正常模式下，它们可以支持 100 mA 的电流。LDO3 和 LDO4 可以支持 200 mA 电流并可被配置为负载开关。TPS65217 提供灵活的加电和断电时序并拥有诸如电源正常输出、按钮监视器等附加功能，TPS65217 内部集成的硬件复位功能和温度传感是为了保护电池。

最大输出电流2A

最高输出充电电流 700mA

耐压达 20V 的 USB 和 AC 输入

限制USB上的输入电流（100mA 500mA 1300mA 1800mA）

热调节、安全定时器

支持温度传感器输入

具有集成 FET 开关的 3 个降压 DCDC

2 个可调节 LDO

2 个可被配置为 LDO 的独立负载开关

欠压闭锁和电池故障比较器

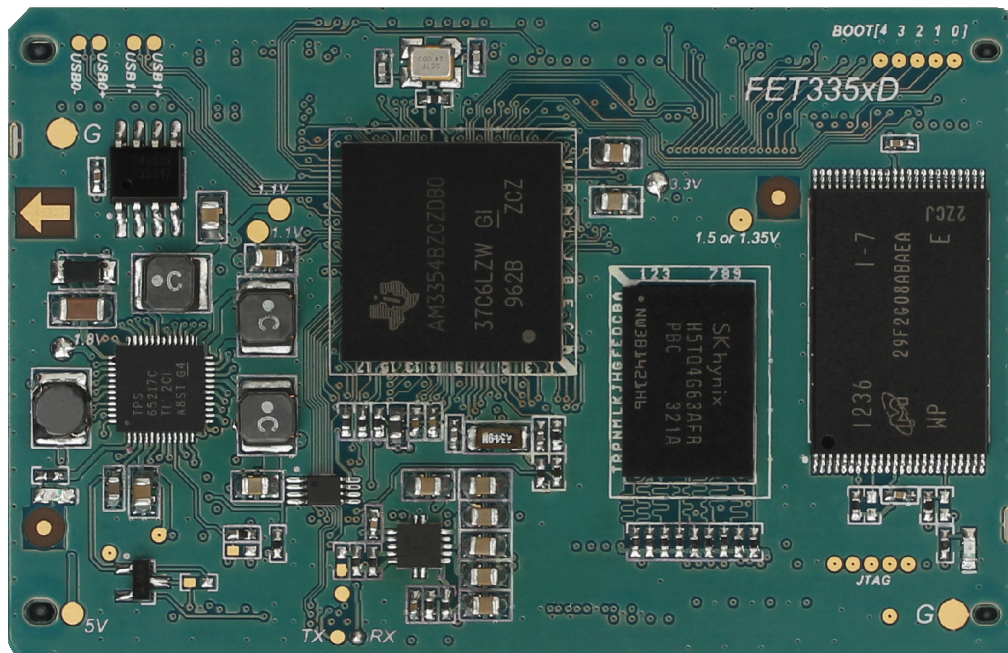
常开按钮监视器

硬件复位引脚

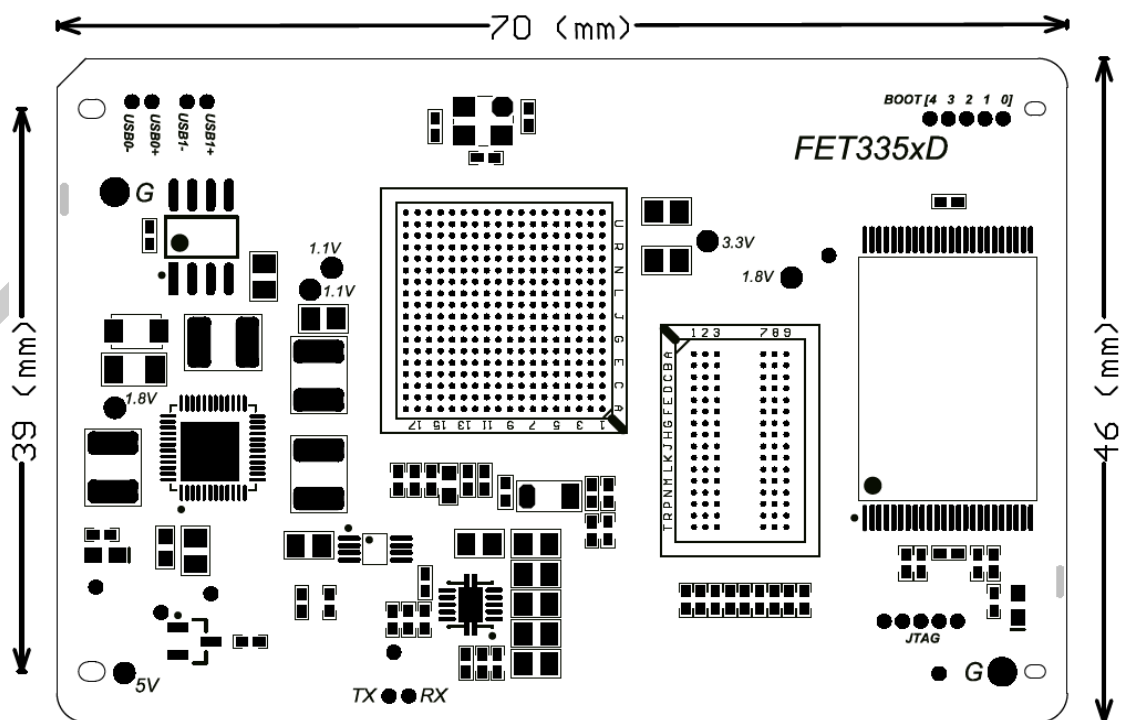
受密码保护的 I²C 寄存器（地址 0 x 24）

2.2 FET335xD核心板结构图

2.2.1 FET335xD核心板外观图



2.2.2 FET335xD核心板结构尺寸图



2.3 FET335xD核心板性能参数

2.3.1 核心板配置资源

硬件资源	规格	说明
CPU	AM3354	—
Nand Flash	256 MB	SLC工艺
内存	512MB	DDR3

软件资源	版本	说明
WinCE	WinCE 7.0	—
Linux	Linux 3.2	—
Android	Android2.3 & Android4.2	—

2.3.2 供电模式

外部USB供电（引脚：USB_DC）外部电池供电（引脚：BAT）

功能	引脚标号	规格				说明
		最小	典型	最大	单位	
主电源供电	DC5V	4.5	5	5.5	V	建议用5V/2A适配器
外部USB供电	USB_DC	4.5	—	5.5	V	—
外部电池供电	BAT	3.0	3.7	4.5	V	只限于给核心板供电

2.3.3 测试功耗

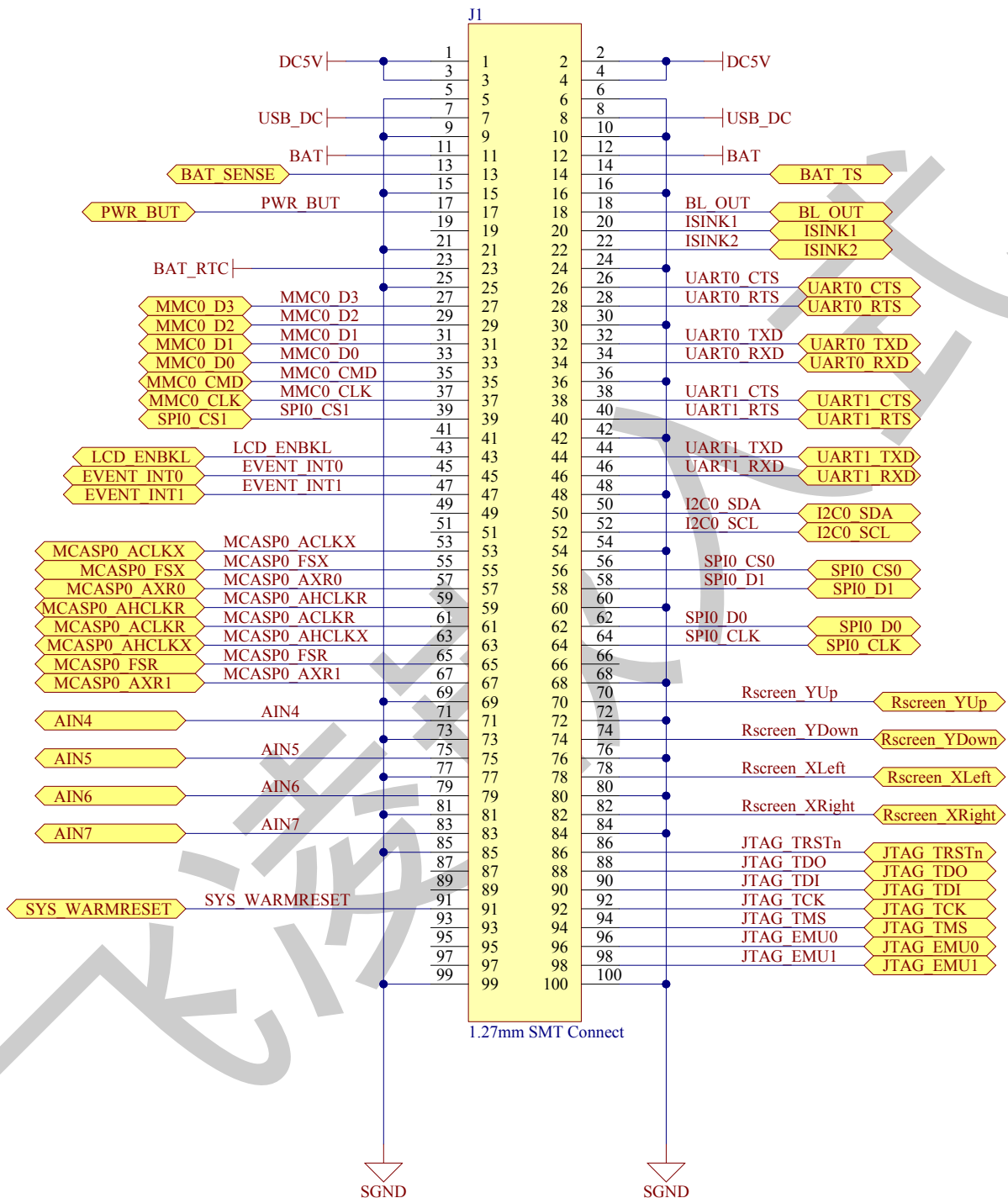
OK335xD 整机功耗测试（实测数据）						
编号	测试项目	供电电压	启动工作电流/功耗		对应项目操作工作电流/功耗	屏幕亮度
			瞬时峰值	稳定值		
1	整机功耗测试（带 7 寸屏+1 路 1000M 网络+USB+SD 卡+视频播放）	5V±0.05V	940mA/4.7W	900mA/4.5W	940mA/4.7W	60%
2	部分功耗测试 1（带 7 寸屏+视频播放）	5V±0.05V	790mA/3.95W	750mA/3.75W	800mA/4W	60%
3	部分功耗测试 2（带 7 寸屏+USB 传输）	5V±0.05V	860mA/4.3W	810mA/4.05W	810mA/4.05W	60%
4	部分功耗测试 3（带 7 寸屏+2 路 1000M 网络）	5V±0.05V	880mA/4.4W	840mA/4.2W	840mA/4.2W	60%
5	部分功耗测试 4（带 7 寸屏+SD 卡传输）	5V±0.05V	790mA/3.95W	750mA/3.75W	750mA/3.75W	60%
6	整机功耗测试（不带屏）	5V±0.05V	300mA/1.5W	270mA/1.35W	270mA/1.35W	—
7	核心板功耗测试（不带底板显示屏）	5V±0.05V	230mA/1.15W	190mA/0.95W	190mA/0.95W	—

8	屏幕亮度调节功耗（仅带 7 寸屏，无任何操作最小值-最大值）	5V±0.05V	790mA/3.95W	750mA/3.75W	占空比 10%： 550mA/2.75W 占空比 100%： 900mA/4.5W	10%-100%
---	--------------------------------	----------	-------------	-------------	--	----------

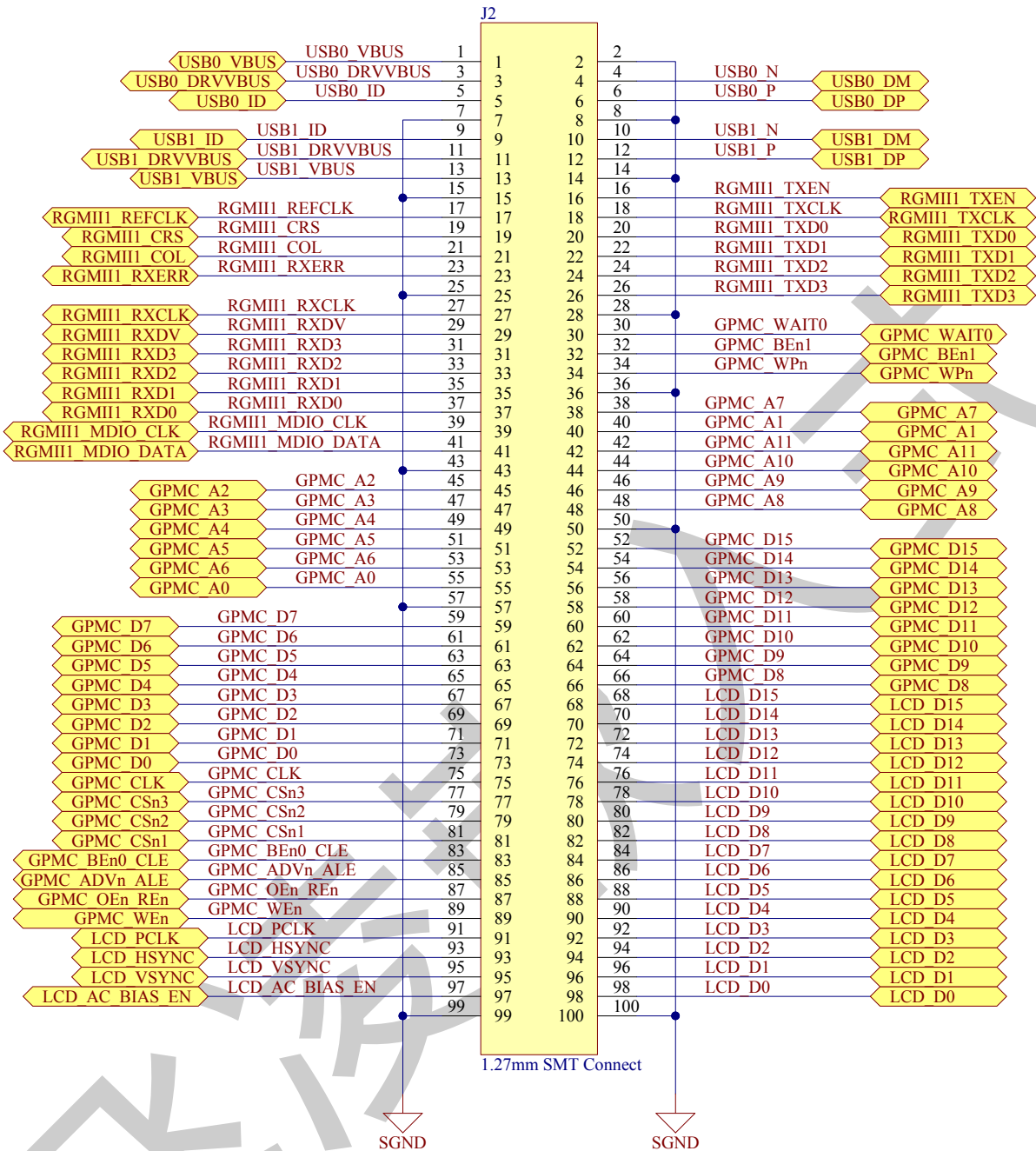
2.3.4 工作环境测试

测试项	参数	规格				说明
		最小	典型	最大	单位	
工作环境 温度	工作环境	-40	25	+85	℃	工业级版本
	存储环境	-40	25	+125	℃	
	工作环境	0	25	+70	℃	商业级版本
	存储环境	-40	25	+125	℃	
湿度	工作环境	10	—	90	%RH	不结露情况下
	存储环境	5	—	95	%RH	

2.4 FET335xD核心板默认引脚定义图



J1



J2

2.5 FET335xD核心板引脚功能参考说明

FET335xD核心板引脚通过二组间距为1.27mm的排母引出，每组排母100pin，排母J1，J2位置见核心板背面丝印层。

FET335xD核心板所采用的AM335x处理器包含有多种外围接口，为降低封装成本同时保持最大限度的功能，约半数的CPU引脚可以有8路信号功能复用，这些引脚功能复用的组合集是有一定量的，这些有效的集合被精心挑选来为用户提供更多可能的应用场景。德州仪器为此开发了一种Windows软件工具称为Pin Mux Utility，可以帮助系统设计师在做基于AM335x产品设计时适当的选择引脚功能资源配置。这种工具为选用特定外围接口的有效IO设置的提供了方法，确保选择的引脚多路复用配置都是AM335x支持的有效IO设置。

注：为便于用户理解与软硬件设计，飞凌的FET335xD核心板模块已经将模块每个引脚的“默认功能”做了硬性定义，即用户可以将核心板模块看做一个特定的“集成芯片”来使用而无需考虑其内部结构，关于核心板的主要特性会在本手册中说明。用户不可轻易更改引脚功能，否则将造成软件驱动冲突，对此若有疑问或是其他功能需求可联系飞凌的技术人员进行协调解决。

2.5.1 FET335xD核心板默认引脚定义表

Pin：核心板连接器引出引脚序号

Ball：核心板上直接连到CPU的引脚序号

Pin Name：核心板连接器引脚名(其中连接CPU的引脚定义只是以引脚第一功能命名，需注意核心板有一部分引脚默认功能并不是使用CPU引脚的第一功能，默认功能参照 **默认功能描述** 一栏)

默认功能描述：核心板引脚的默认功能配置定义

V：引脚信号电平

NC：未直接连到CPU的引脚

J1				
Pin	Ball	Pin Name	V	默认功能描述
1	NC	DC5V	5	电源输入
2	NC	DC5V	5	电源输入
3	NC	DC5V	5	电源输入
4	NC	DC5V	5	电源输入
5	NC	GND	0	地
6	NC	GND	0	地
7	NC	USB_DC	5	USB电源输入
8	NC	USB_DC	5	USB电源输入
9	NC	GND	0	地
10	NC	GND	0	地
11	NC	BAT	3.7	锂电池供电输入
12	NC	BAT	3.7	锂电池供电输入
13	NC	BAT_SENSE	3.3	电池传感器输入

14	NC	BAT_TS	3.3	电池温度传感器输入
15	NC	GND	0	地
16	NC	GND	0	地
17	NC	PWR_BUT	3.3	按键输入(低有效)
18	NC	BL_OUT	3.3	WLED反馈
19		NC		
20	NC	ISINK1	3.3	WLED SINK1电流输入
21	NC	GND	0	地
22	NC	ISINK2	3.3	WLED SINK2电流输入
23	NC	BAT_RTC	3.3	RTC电池输入
24	NC	GND	0	地
25	NC	GND	0	地
26	E18	UART0_CTS	3.3	UART0 清除发送
27	F17	MMC0_D3	3.3	MMC/SD/SDIO 数据3
28	E17	UART0_RTS	3.3	UART0 请求发送
29	F18	MMC0_D2	3.3	MMC/SD/SDIO 数据2
30	NC	GND	0	地
31	G15	MMC0_D1	3.3	MMC/SD/SDIO 数据1
32	E16	UART0_TXD	3.3	UART0 发送
33	G16	MMC0_D0	3.3	MMC/SD/SDIO数据0
34	E15	UART0_RXD	3.3	UART0接收
35	G18	MMC0_CMD	3.3	MMC/SD/SDIO 命令
36	NC	GND	0	地
37	G17	MMC0_CLK	3.3	MMC/SD/SDIO 时钟
38	D18	UART1_CTS	3.3	DCAN0_TX DCAN0发送
39	C15	SPI0_CS1	3.3	SPI0片选
40	D17	UART1_RTS	3.3	DCAN0_RX DCAN0接收
41		NC		
42	NC	GND	0	地
43	C18	LCD_ENBKL	3.3	ECPA0_IN_PWM0_OUT LCD背光控制 (PWM0输出)
44	D15	UART1_TXD	3.3	UART1发送
45	A15	EVENT_INTR0	3.3	外部DMA中断0
46	D16	UART1_RXD	3.3	UART1接收
47	D14	EVENT_INT1	3.3	外部DMA 中断 1
48	NC	GND	0	地
49		NC		
50	C17	I2C0_SDA	3.3	I2C0 数据
51		NC		
52	C16	I2C0_SCL	3.3	I2C0 时钟
53	A13	MCASP0_ACLKX	3.3	SPI1_SCLK
54	NC	GND	0	地

55	B13	MCASP0_FSX	3.3	SPI1_D0
56	A16	SPI0_CS0	3.3	I2C1_SCL时钟信号
57	D12	MCASP0_AXR0	3.3	SPI1_D1
58	B16	SPI0_D1	3.3	I2C1_SDA数据信号
59	C12	MCASP0_AHCLKR	3.3	SPI1_CS0
60	NC	GND	0	地
61	B12	MCASP0_ACLKR	3.3	MCASP1_ACLKX McASP1 发送位时钟
62	B17	SPI0_D0	3.3	UART2_TX UART2发送
63	A14	MCASP0_AHCLKX	3.3	MCASP1_AXR1 McASP1 串行数据
64	A17	SPI0_CLK	3.3	UART2_RX UART2接收
65	C13	MCASP0_FSR	3.3	MCASP1_FSX McASP1 发送帧同步
66		NC		
67	D13	MCASP0_AXR1	3.3	MCASP1_AXR0 McASP1 串行数据
68	NC	GND	0	地
69	NC	GND	0	地
70	B7	Rscreen_Yup	3.3	模拟输入2
71	C8	AIN4	3.3	模拟输入4
72	NC	GND	0	地
73	NC	GND	0	地
74	A7	Rscreen_YDown	3.3	模拟输入3
75	B8	AIN5	3.3	模拟输入5
76	NC	GND	0	地
77	NC	GND	0	地
78	B6	Rscreen_XLeft	3.3	模拟输入0
79	A8	AIN6	3.3	模拟输入6
80	NC	GND	0	地
81	NC	GND	0	地
82	A7	Rscreen_XRight	3.3	模拟输入1
83	C9	AIN7	3.3	模拟输入7
84	NC	GND	0	地
85	NC	GND	0	地
86	B10	JTAG_TRSTn	3.3	JTAG 测试复位
87		NC		
88	A11	JTAG_TDO	3.3	JTAG 测试数据输出
89		NC		
90	B11	JTAG_TDI	3.3	JTAG 测试数据输入
91	A10	SYS_WARMRESET	3.3	SYS_RESETn 系统复位
92	A12	JTAG_TCK	3.3	JTAG 测试时钟
93		NC		
94	C11	JTAG_TMS	3.3	JTAG 测试模式选择
95		NC		
96	C14	JTAG_EMU0	3.3	MISC 仿真0

97		NC		
98	B14	JTAG_EMU1	3.3	MISC 仿真1
99	NC	GND	0	地
100	NC	GND	0	地

J2

Pin	Ball	Pin Name	V	描述
1	P15	USB0_VBUS	5	USB0_VBUS
2	NC	GND	0	地
3	F16	USB0_DRVVBUS	3.3	USB0 VBUS控制输出
4	N18	USB0_DM	3.3	USB0数据 负
5	P16	USB0_ID	3.3	USB0 OTG 标识
6	N17	USB0_DP	3.3	USB0数据 正
7	NC	GND	0	地
8	NC	GND	0	地
9	P17	USB1_ID	3.3	USB1 OTG 标识
10	R18	USB1_DM	3.3	USB1数据 负
11	F15	USB1_DRVVBUS	3.3	USB1 VBUS控制输出
12	R17	USB1_DP	3.3	USB1数据 正
13	T18	USB1_VBUS	5	USB1_VBUS
14	NC	GND	0	地
15	NC	GND	0	地
16	J16	GMII1_TXEN	3.3	GMII1 发送使能
17	H18	RGMII1_REFCLK	3.3	MII1 参考时钟
18	K18	GMII1_TXCLK	3.3	GMII1 发送时钟
19	H17	RGMII1_CRS	3.3	MII1载波侦听
20	K17	GMII1_TXD0	3.3	MII1 发送数据0
21	H16	RGMII1_COL	3.3	MII1 冲突检测
22	K16	GMII1_TXD1	3.3	MII1 发送数据1
23	J15	RGMII1_RXERR	3.3	GPIO3_2 通用IO
24	K15	GMII1_TXD2	3.3	MII1 发送数据2
25	NC	GND	0	地
26	J18	GMII1_TXD3	3.3	MII1发送数据3
27	L18	GMII1_RXCLK	3.3	MII1接收时钟
28	NC	GND	0	地
29	J17	RGMII1_RXDV	3.3	MII1接收数据有效
30	T17	GPMC_WAIT0	3.3	GPMC等待
31	L17	GMII1_RXD3	3.3	MII1接收数据3
32	U18	GPMC_BEn1	3.3	GPMC 位使能1
33	L16	GMII1_RXD2	3.3	MII1 接收数据2

34	U17	GPMC_WPn	3.3	GPMC 写保护
35	L15	GMII1_RXD1	3.3	MII1 接收数据1
36	NC	GND	0	地
37	M16	GMII1_RXD0	3.3	MII1接收数据0
38	T15	GPMC_A7	3.3	GPIO1_23 通用IO
39	M18	GMII1_MDIO_CLK	3.3	MDIO 数据
40	V14	GPMC_A1	3.3	GPIO1_17 通用IO
41	M17	GMII1_MDIO_DATA	3.3	MDIO 数据
42	V17	GPMC_A11	3.3	GPMC 地址11
43	NC	GND	0	地
44	T16	GPMC_A10	3.3	GPMC 地址10
45	U14	GPMC_A2	3.3	GPIO1_18 通用IO
46	U16	GPMC_A9	3.3	GPIO1_25 通用IO
47	T14	GPMC_A3	3.3	GPIO1_19 通用IO
48	V16	GPMC_A8	3.3	GPIO1_24 通用IO
49	R14	GPMC_A4	3.3	GPIO1_20 通用IO
50	NC	GND	0	地
51	V15	GPMC_A5	3.3	GPIO1_21 通用IO
52	U13	GPMC_D15	3.3	GPMC 数据15
53	U15	GPMC_A6	3.3	GPIO1_22 通用IO
54	V13	GPMC_D14	3.3	GPMC 数据14
55	R13	GPMC_A0	3.3	GPIO1_16 通用IO
56	R12	GPMC_D13	3.3	GPMC 数据13
57	NC	GND	0	地
58	T12	GPMC_D12	3.3	GPMC 数据12
59	T9	GPMC_D7	3.3	GPMC 数据7
60	U12	GPMC_D11	3.3	GPMC 数据11
61	R9	GPMC_D6	3.3	GPMC 数据6
62	T11	GPMC_D10	3.3	GPMC 数据10
63	V8	GPMC_D5	3.3	GPMC 数据5
64	T10	GPMC_D9	3.3	GPMC 数据9
65	U8	GPMC_D4	3.3	GPMC 数据4
66	U10	GPMC_D8	3.3	GPMC 数据8
67	T8	GPMC_D3	3.3	GPMC数据3
68	T5	LCD_D15	3.3	LCD 数据15
69	R8	GPMC_D2	3.3	GPMC 数据2
70	V4	LCD_D14	3.3	LCD 数据14
71	V7	GPMC_D1	3.3	GPMC 数据1
72	V3	LCD_D13	3.3	LCD 数据13
73	U7	GPMC_D0	3.3	GPMC 数据0
74	V2	LCD_D12	3.3	LCD数据12
75	V12	GPMC_CLK	3.3	GPIO2_1 通用IO

76	U4	LCD_D11	3.3	LCD 数据11
77	T13	GPMC_CSn3	3.3	GPMC 片选
78	U3	LCD_D10	3.3	LCD 数据10
79	V9	GPMC_CS2n	3.3	GPMC 片选
80	U2	LCD_D9	3.3	LCD 数据9
81	U9	GPMC_CSn1	3.3	GPMC 片选
82	U1	LCD_D8	3.3	LCD 数据8
83	T6	GPMC_BEn0_CLE	3.3	GPMC 位使能 / 命令锁存使能
84	T4	LCD_D7	3.3	LCD 数据7
85	R7	GPMC_ADVn_ALE	3.3	GPMC 地址有效 / 地址锁存有效
86	T3	LCD_DAT6	3.3	LCD 数据6
87	T7	GPMC_OEn_REn	3.3	GPMC 输出/读 使能
88	T2	LCD_D5	3.3	LCD 数据5
89	U6	GPMC_WEn	3.3	GPMC 写使能
90	T1	LCD_D4	3.3	LCD 数据4
91	V5	LCD_PCLK	3.3	LCD 像素时钟
92	R4	LCD_D3	3.3	LCD 数据3
93	R5	LCD_HSYNC	3.3	LCD 水平同步
94	R3	LCD_D2	3.3	LCD 数据2
95	U5	LCD_VSYNC	3.3	LCD 垂直同步
96	R2	LCD_D1	3.3	LCD数据1
97	R6	LCD_AC_BIAS_EN	3.3	LCD AC 偏置使能
98	R1	LCD_D0	3.3	LCD 数据0
99	NC	GND	0	地
100	NC	GND	0	地

注：带*表示 AM3354不含此功能。

2.5.2 FET335xD核心板电源接口

FET335xD核心板使用电源管理芯片TPS65217对CPU提供灵活的电源配置和能耗管理

引脚	信号名称	类型	功能描述
(J1) 1 2 3 4	DC5V	POWER	电源输入端+5V，主电源供电
(J1) 7 8	USB_DC	POWER	外部USB供电
(J1) 11 12	BAT	POWER	外部电池供电
(J1) 23	BAT_RTC	POWER	RTC实时时钟备用电池输入

2.5.3 FET335xD核心板系统控制接口

引脚	信号名称	功能描述
(J1) 91	SYS_WARMRESET	核心板复位信号SYS_RESETh

2.5.4 boot配置引脚

LCD 0..15 引脚同时做为系统启动配置引脚使用。

FET335xD的默认启动配置为 01 00 00 0 0 10 0 00100

启动顺序 UART0 → XIP → MMC0 → NAND



附：系统启动配置引脚表格：

SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
For all boot modes: Crystal Frequency	For all boot modes: Set to 00b for normal operation	For XIP boot: Muxed or non-muxed device For NAND boot: must be 00b	For NAND and NANDI2C boot: NAND ECC For Fast External Boot: WAIT enable[2]	For XIP and NAND boot: Bus width[1]	For EMAC boot: PHY mode	For all boot modes: CLKOUT1 output enabled/disabled on XDMA_EVENT_INTR0					
CONTROL STATUS[23:22]	CONTROL STATUS[21:20]	CONTROL STATUS[19:18]	CONTROL STATUS[17]	CONTROL STATUS[16]	CONTROL STATUS[7:6]	CONTROL STATUS[5]	CONTROL STATUS[4:0]	1st	2nd	3rd	4th
							00000b	Reserved			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code[4]	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00001b	UART0	XIP w/ WAIT[2] (MUX2) [3]	MMIO	SPI0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00010b	UART0	SPI0	NAND	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00011b	UART0	SPI0	XIP (MUX2) [3]	MMIO
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00100b	UART0	XIP w/ WAIT[2] (MUX1)	MMIO	NAND

SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00101b	UART0	XIP (MUX1) [3]	SPI0	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00110b	EMAC1	SPI0	NAND	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	00111b	EMAC1	MMIO	XIP w/ WAIT[2] (MUX2) [3]	NAND
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 0	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01000b	EMAC1	MMIO	XIP (MUX2) [3]	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01001b	EMAC1	XIP w/ WAIT[2] (MUX1) [3]	NAND	MMIO

SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01010b	EMAC1	XIP (MUX1) [3]	SPI0	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01011b	USB0	NAND	SPI0	MMC0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01100b	USB0	NAND	XIP (MUX2) [3]	NANDI2C
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01101b	USB0	NAND	XIP (MUX1) [3]	SPI0
							01110b	Reserved			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For Fast External Boot: 00b = non-muxed device 10b = muxed device x1b = reserved	For Fast External Boot: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	01111b	Fast External Boot	UART0	EMAC1	Reserved

SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10000b	XIP (MUX1) [3]	UART0	EMAC1	MMC0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10001b	XIP w/ WAIT (MUX1) [2] [3]	UART0	EMAC1	MMC0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10010b	NAND	NANDI2C	USB0	UART0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10011b	NAND	NANDI2C	MMC0	UART0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10100b	NAND	NANDI2C	SPI0	EMAC1

SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For NAND boot: must be 00b	0 = ECC done by ROM 1 = ECC handled by NAND	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10101b	NANDI2C	MMC0	EMAC1	UART0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10110b	SPI0	MMC0	UART0	EMAC1
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	10111b	MMC0	SPI0	UART0	USB0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11000b	SPI0	MMC0	USB0	UART0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11001b	SPI0	MMC0	EMAC1	UART0
SYSBOOT[15:14]	SYSBOOT[13:12]	SYSBOOT[11:10]	SYSBOOT[9]	SYSBOOT[8]	SYSBOOT[7:6]	SYSBOOT[5]	SYSBOOT[4:0]	Boot Sequence			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11010b	XIP (MUX2) ^[3] 1	UART0	SPI0	MMC0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For XIP boot: 00b = non-muxed device 10b = muxed device x1b = reserved	Don't care for ROM code	0 = 8-bit device 1 = 16-bit device	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11011b	XIP w/ WAIT ^[2] (MUX2) ^[3] 1	UART0	SPI0	MMC0
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	Don't care for ROM code	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11100b	MMC1	MMC0	UART0	USB0
							11101b	Reserved			
							11110b	Reserved			
00b = 19.2MHz 01b = 24MHz 10b = 25MHz 11b = 26MHz	00b (all other values reserved)	For Fast External Boot: 00b = non-muxed device 10b = muxed device x1b = reserved	For Fast External Boot: 0 = WAIT signal not monitored during boot 1 = WAIT signal monitored during boot	0 = 8-bit device 1 = 16-bit device	00b = MII 01b = RMII 10b = RGMII with internal delay 11b = RGMII w/o internal delay	0 = CLKOUT1 disabled 1 = CLKOUT1 enabled	11111b	Fast External Boot	EMAC1	UART0	Reserved

- MMC1: MMC or SD card (MMC port 1)
- MMC0: MMC or SD card (MMC port 0)
- NAND / NANDI2C: NAND flash memory / EEPROM on I2C0
- XIP: NOR or other XIP device
- XIP w/ WAIT: NOR or other XIP device
- MUX1: Boot with XIP_MUX1 signals
- MUX2: Boot with XIP_MUX2 signals
- UART0: UART interface (UART port 0)
- EMAC1: Ethernet interface (EMAC port 1)
- SPI0: SPI EEPROM (SPI 0, CS0)
- USB0: USB interface (USB0)

2.6 FET335xD核心板默认功能说明

2.6.1 通用存储控制器(GPMC)总线

通用存储器控制器（GPMC）是用于连接外部存储设备：如伪SRAM，NAND闪存，NOR闪存，SRAM或现场可编程门阵列（FPGA）等

引脚	信号	描述
J2_73	GPMC_D0	GPMC 数据0
J2_71	GPMC_D1	GPMC 数据1
J2_69	GPMC_D2	GPMC 数据2
J2_67	GPMC_D3	GPMC 数据3
J2_65	GPMC_D4	GPMC 数据4
J2_63	GPMC_D5	GPMC 数据5
J2_61	GPMC_D6	GPMC 数据6
J2_59	GPMC_D7	GPMC 数据7
J2_66	GPMC_D8	GPMC 数据8
J2_64	GPMC_D9	GPMC 数据9
J2_62	GPMC_D10	GPMC 数据10
J2_60	GPMC_D11	GPMC 数据11
J2_58	GPMC_D12	GPMC 数据12
J2_56	GPMC_D13	GPMC 数据13
J2_54	GPMC_D14	GPMC 数据14
J2_52	GPMC_D15	GPMC 数据15
J2_55	GPMC_A0	GPMC 地址0
J2_40	GPMC_A1	GPMC 地址1
J2_45	GPMC_A2	GPMC 地址2
J2_47	GPMC_A3	GPMC 地址3
J2_49	GPMC_A4	GPMC 地址4
J2_51	GPMC_A5	GPMC 地址5
J2_53	GPMC_A6	GPMC 地址6
J2_38	GPMC_A7	GPMC 地址7
J2_48	GPMC_A8	GPMC 地址8
J2_46	GPMC_A9	GPMC 地址9
J2_44	GPMC_A10	GPMC 地址10
J2_42	GPMC_A11	GPMC 地址11
J2_75	GPMC_CLK	GPMC时钟
J2_79	GPMC_CS2n	GPMC 片选
J2_81	GPMC_CSn1	GPMC 片选
J2_82	GPMC_CSn1	GPMC 片选
J2_83	GPMC_BEn0_CLE	GPMC 位使能 / 命令锁存使能
J2_85	GPMC_ADVn_ALE	GPMC 地址有效 / 地址锁存有效
J2_87	GPMC_OEn_REn	GPMC 输出/读 使能
J2_89	GPMC_WEn	GPMC 写使能

2.6.2 LCD控制器

LCD控制器有两个独立的控制器(光栅控制器、LCD界面显示驱动控制器 (LIDD)), 最大分辨率为1366x768。最大帧频由图像大小与像素时钟速率决定。FET335xD默认LCD显示模式为16位显示模式(即RGB565模式)。

LCD 16位显示模式 (RGB565)

引脚	信号	描述
J2_68	LCD_D15	LCD 数据15
J2_70	LCD_D14	LCD 数据14
J2_72	LCD_D13	LCD 数据13
J2_74	LCD_D12	LCD数据12
J2_76	LCD_D11	LCD 数据11
J2_78	LCD_D10	LCD 数据10
J2_80	LCD_D9	LCD 数据9
J2_82	LCD_D8	LCD 数据8
J2_84	LCD_D7	LCD 数据7
J2_86	LCD_D6	LCD 数据6
J2_88	LCD_D5	LCD 数据5
J2_90	LCD_D4	LCD 数据4
J2_92	LCD_D3	LCD 数据3
J2_94	LCD_D2	LCD 数据2
J2_96	LCD_D1	LCD 数据1
J2_98	LCD_D0	LCD 数据0
J2_97	LCD_AC_BIAS_EN	LCD AC 偏置使能
J2_91	LCD_PCLK	LCD 像素时钟
J2_93	LCD_HSYNC	LCD 水平同步
J2_95	LCD_VSYNC	LCD 垂直同步

LCD 24位显示模式 (RGB888)

引脚	信号	描述
J2_52	LCD_D16	LCD 数据16
J2_54	LCD_D17	LCD 数据17
J2_56	LCD_D18	LCD 数据18
J2_58	LCD_D19	LCD 数据19
J2_60	LCD_D20	LCD 数据20
J2_62	LCD_D21	LCD 数据21
J2_64	LCD_D22	LCD 数据22
J2_66	LCD_D23	LCD 数据23
J2_68	LCD_D15	LCD 数据15
J2_70	LCD_D14	LCD 数据14
J2_72	LCD_D13	LCD 数据13
J2_74	LCD_D12	LCD 数据12

J2_76	LCD_D11	LCD 数据11
J2_78	LCD_D10	LCD 数据10
J2_80	LCD_D9	LCD 数据9
J2_82	LCD_D8	LCD 数据8
J2_84	LCD_D7	LCD 数据7
J2_86	LCD_D6	LCD 数据6
J2_88	LCD_D5	LCD 数据5
J2_90	LCD_D4	LCD 数据4
J2_92	LCD_D3	LCD 数据3
J2_94	LCD_D2	LCD 数据2
J2_96	LCD_D1	LCD 数据1
J2_98	LCD_D0	LCD 数据0
J2_97	LCD_AC_BIAS_EN	LCD AC 偏置使能
J2_91	LCD_PCLK	LCD 像素时钟
J2_93	LCD_HSYNC	LCD 水平同步
J2_95	LCD_VSYNC	LCD 垂直同步

2.6.3 MAC控制器

MAC控制器支持MII/RMII/RGMII协议，支持IEEE 1588v2 精准时间协议(PTP)

引脚	信号	描述
J2_16	RGMII1_TXEN	RGMII1 发送使能
J2_17	RGMII1_REFCLK	RGMII1 参考时钟
J2_18	RGMII1_TXCLK	GMII1 发送时钟
J2_19	RGMII1_CRS	RGMII1载波侦听
J2_20	RGMII1_TXD0	RGMII1 发送数据0
J2_21	RGMII1_COL	RGMII1 冲突检测
J2_22	RGMII1_TXD1	RGMII1 发送数据1
J2_23	RGMII1_RXERR	RGMII1 接收数据错误指示
J2_24	RGMII1_TXD2	RGMII1 发送数据2
J2_26	RGMII1_TXD3	RGMII1发送数据3
J2_27	RGMII1_RXCLK	RGMII1接收时钟
J2_29	RGMII1_RXDV	RGMII1接收数据有效
J2_31	RGMII1_RXD3	RGMII1接收数据3
J2_33	RGMII1_RXD2	RGMII1 接收数据2
J2_35	RGMII1_RXD1	RGMII1 接收数据1
J2_37	RGMII1_RXD0	RGMII1接收数据0
J2_39	GMII1_MDIO_CLK	MDIO 时钟
J2_41	GMII1_MDIO_DATA	MDIO 数据

2.6.4 通用串行接口（USB）

USB子系统有两个独立的USB2.0模块。

注:通过一个USB 2.0集线器，USB 1.1设备可以连接到USB2.0主机端口。

USB1.1设备可以直接连接到USB OTG端口。

引脚	信号	描述
J2_1	USB0_VBUS	USB0_VBUS
J2_3	USB0_DRVVBUS	USB0 VBUS控制输出
J2_4	USB0_DM	USB0数据 负
J2_5	USB0_ID	USB0 OTG 标识
J2_6	USB0_DP	USB0数据 正
J2_9	USB1_ID	USB1 OTG 标识
J2_10	USB1_DM	USB1数据 负
J2_11	USB1_DRVVBUS	USB1 VBUS控制输出
J2_12	USB1_DP	USB1数据 正
J2_13	USB1_VBUS	USB1_VBUS

2.6.5 ADC控制器

触摸屏控制器和模拟-数字转换器是一个8通道的通用模拟-数字转换器(ADC)，支持四线、五线、或8线电阻触摸屏(TS)。

引脚	信号	描述
J1_70	Rscreen_Yup (AIN2)	模拟输入2
J1_71	AIN4	模拟输入4
J1_74	Rscreen_YDown (AIN3)	模拟输入3
J1_75	AIN5	模拟输入5
J1_78	Rscreen_XLeft (AIN0)	模拟输入0
J1_79	AIN6	模拟输入6
J1_82	Rscreen_XRight (AIN1)	模拟输入1
J1_83	AIN7	模拟输入7

2.6.6 测试仿真接口 (JTAG)

引脚	信号	描述
J1_86	JTAG_TRSTn	JTAG 测试复位
J1_88	JTAG_TDO	JTAG 测试数据输出
J1_90	JTAG_TDI	JTAG 测试数据输入
J1_92	JTAG_TCK	JTAG 测试时钟
J1_94	JTAG_TMS	JTAG 测试模式选择
J1_96	JTAG_EMU0	MISC 仿真0
J1_98	JTAG_EMU1	MISC 仿真1

2.6.7 MMC / SDIO 接口

引脚	信号	描述
J1_27	MMC0_D3	MMC/SD/SDIO 数据3
J1_29	MMC0_D2	MMC/SD/SDIO 数据2

J1_31	MMC0_D1	MMC/SD/SDIO 数据1
J1_33	MMC0_D0	MMC/SD/SDIO数据0
J1_35	MMC0_CMD	MMC/SD/SDIO 命令
J1_37	MMC0_CLK	MMC/SD/SDIO 时钟

2.6.8 I2C 接口

引脚	信号	描述
J1_56	SPI0_CS0	I2C1_SCL时钟信号
J1_58	SPI0_D1	I2C1_SDA数据信号

2.6.9 SPI 接口

引脚	信号	描述
J1_53	MCASP0_ACLKX	SPI1_SCLK SPI1时钟信号
J1_55	MCASP0_FSX	SPI1_D0 SPI1数据信号
J1_57	MCASP0_AXR0	SPI1_D1 SPI1数据信号
J1_59	MCASP0_AHCLKR	SPI1_CS0 SPI1片选信号

2.6.10 UART 接口

引脚	信号	描述
J1_32	UART0_TXD	UART0 发送
J1_34	UART0_RXD	UART0 接收
J1_44	UART1_TXD	UART1 发送
J1_46	UART1_RXD	UART1 接收
J1_26	UART0_CTS	UART4 发送
J1_28	UART0_RTS	UART4 接收

2.6.11 485芯片接口

引脚	信号	描述
J1_28	SPI0_CLK	UART2 接收
J1_28	SPI0_D0	UART2 发送
J2_23	RGMII1_RXERR	GPIO3_2 通用IO

2.6.12 CAN接口

引脚	信号	描述
J1_28	UART1_RTS	DCAN0_RX
J2_23	UART1_CTS	DCAN0_TX

2.6.13 MCASP 接口

含2个多通道音频串口(McASP)、支持I2S接口、支持数字音频接口传输 (SPDIF, IEC60958- 1, 和AES-3 格式)

引脚	信号	描述
J1_53	MCASP0_ACLKX	McASP0发送时钟
J1_55	MCASP0_FSX	McASP0 发送帧同步

J1_57	MCASP0_AXR0	McASP0 串行数据
J1_59	MCASP0_AHCLKR	McASP0 接收主时钟
J1_61	MCASP0_ACLKR	McASP0 接收位时钟
	mcasp1_aclkx	McASP1 发送位时钟
J1_63	MCASP0_AHCLKX	McASP0 接收主时钟
	mcasp1_axr1	McASP1 串行数据
J1_65	MCASP0_FSR	McASP 接收帧同步
	mcasp1_fsx	McASP1 发送帧同步
J1_67	MCASP0_AXR1	mcasp0 串行数据
	mcasp1_axr0	mcasp1 串行数据

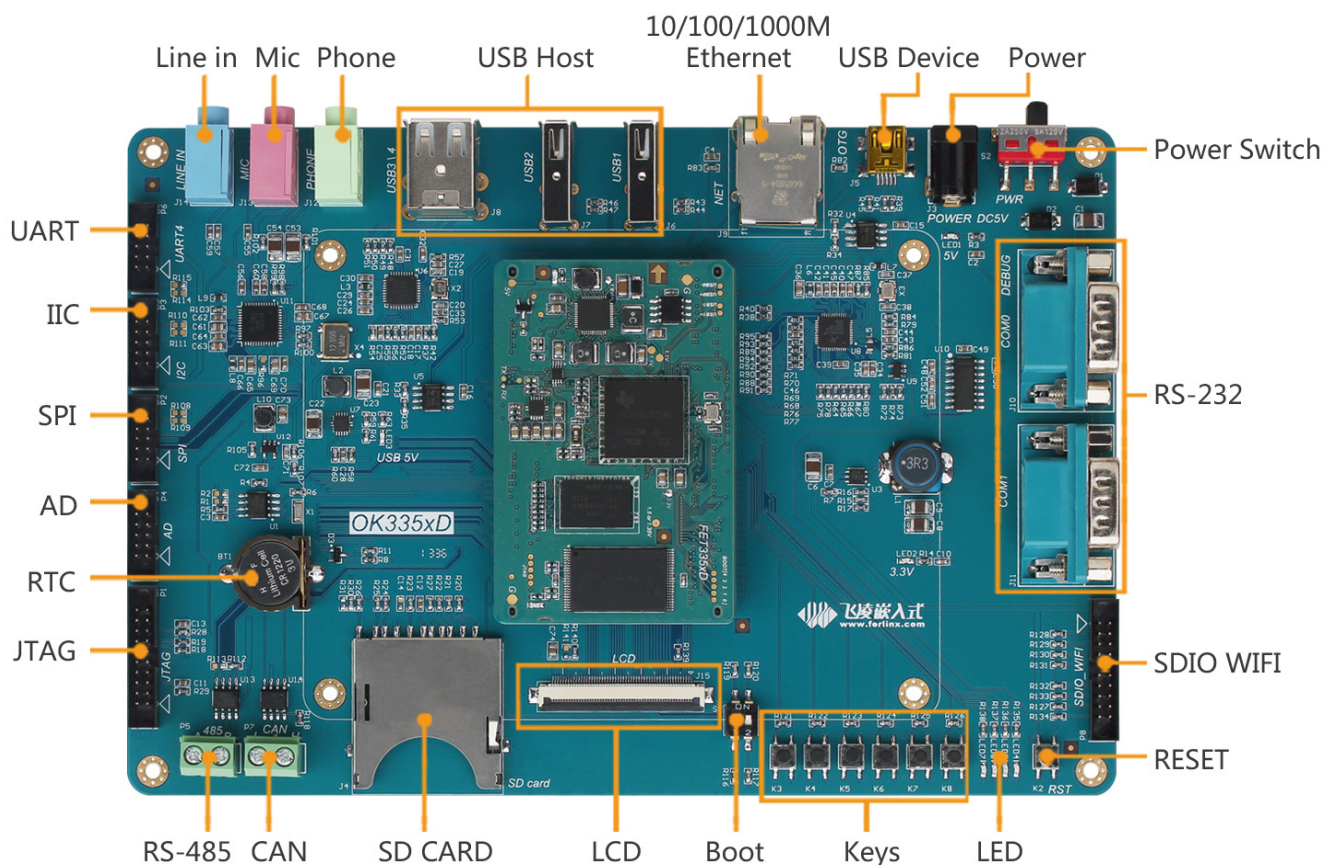
第三章 OK335xD底板说明与硬件设计指导

本章将要介绍飞凌公司生产的OK335xD底板，OK335xD底板是基于FET335xD核心板模块做的扩展底板。这块底板可以用于核心板用户的学习之用，也可作为一个用户基于核心板做硬件设计开发的参考板，便于用户对飞凌的FET335xD核心板进行功能性、可行性评估，从而达到为用户节约开发成本、避免大量设计问题和风险的目的。

注：OK335xD底板与FET335xD核心板使用时，所使用的核心板接口功能定义即为核心板上默认功能定义。

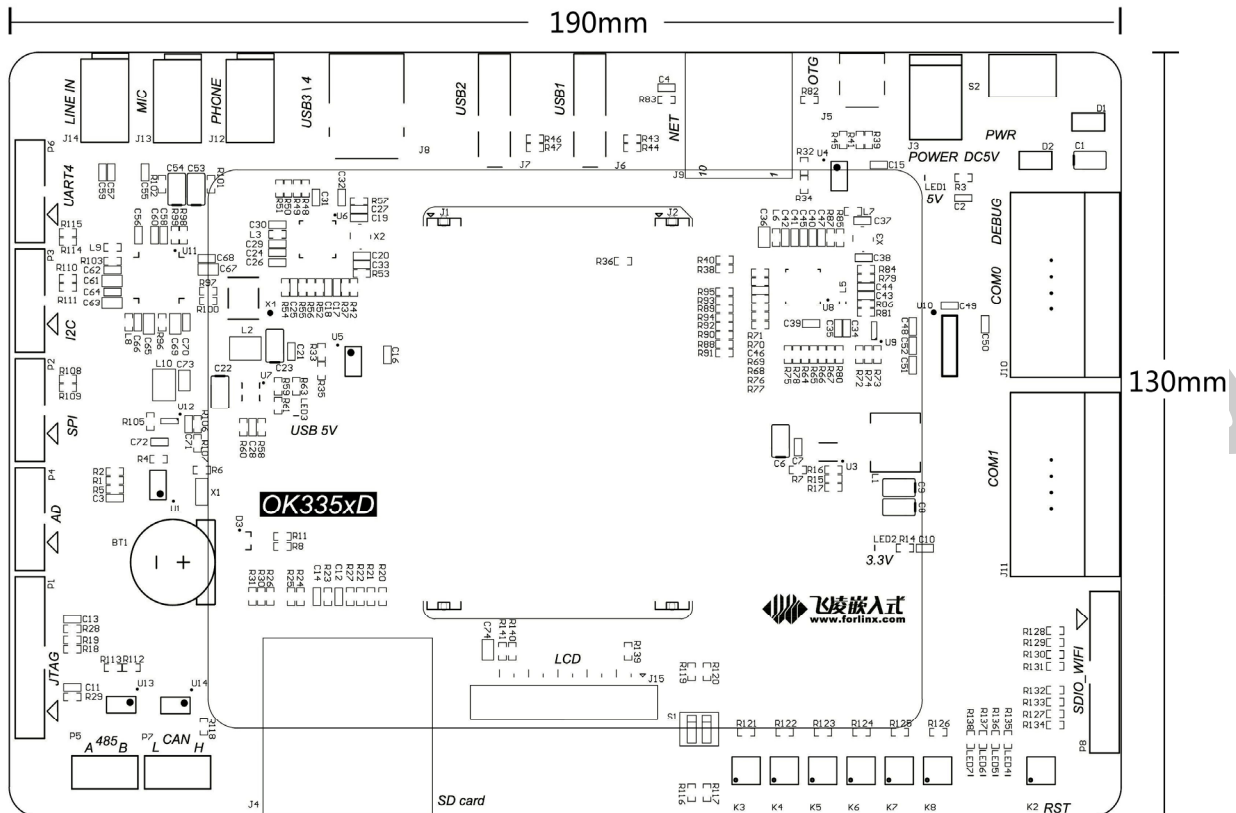
3.1 OK335xD底板及接口外围资源简述

3.1.1 OK335xD底板接口分布图



OK335xD底板尺寸规格为：长 190 mm,宽 130 mm。具体各安装孔位置、尺寸等可参考光盘资料里提供的底板PCB文件。

3.1.2 OK335xD底板布局丝印图



3.1.3 OK335xD底板功能列表

● 1路TFT液晶接口
● 1路RS-485
● 1路CAN现场总线
● 1路SD卡接口
● 6个按键
● 4只LED流水灯
● 1路SDIO WiFi
● 3路UART (2路RS-232 , 1路LVCMOS (3.3V))
● 4路USB 2.0 HOST
● 1路USB 2.0 OTG
● 1路千兆以太网
● 1组音频接口, 其中包括耳机输出, 话筒输入, 线路输入,
● 1路I2C
● 1路SPI
● 1组模拟量输入接口, 多达8路模拟量输入
● 板载实时时钟DS1337

3.2 OK335xD底板功能详细介绍与硬件设计指导

本节介绍OK335xD开发板上的各类资源，包括板载的各种功能接口和模块已经占用的核心板资源。我们在资料光盘中提供了完整的底板原理图，以供参考使用。若用户有自行设计FET335xD核心板扩展底板的需求，也可参照此底板原理图进行删减修改来达到用户自己的需求。

底板上更为详细的接口和引脚信息，请参考核心板及底板原理图，自行设计时也以原理图为准。

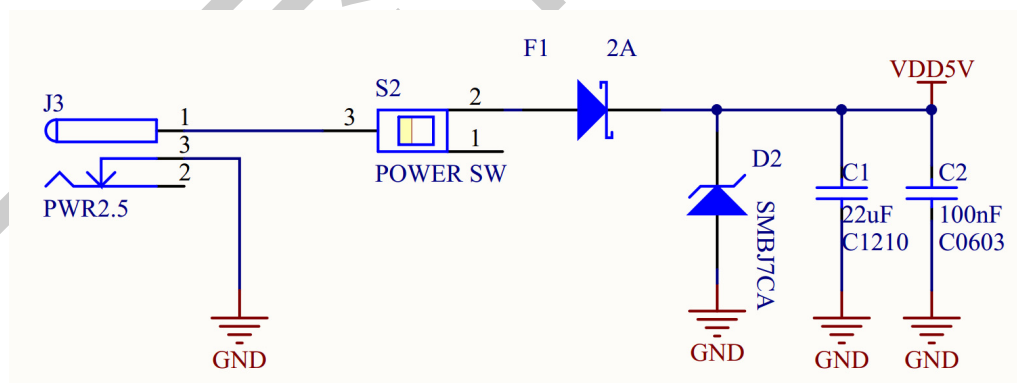
注：由于AM335x的管脚复用功能较多，在原理图的设计上，底板所用的网络标号一律和核心板一致，这样底板的设计更易对应核心板，保持电路连贯性。

FET335xD核心板最小系统：

FET335xD核心板的最小系统，即核心板核心板可以启动的最低条件，主要包括**电源5V供电**和正确的**boot引脚配置**，所以FET335xD核心板的最小系统是很简洁的。不过一般情况下，除最小系统外建议连接上一些外部设备，例如调试串口，否则用户无法判断系统是否启动。做好这些后再在此基础上根据飞凌提供的核心板默认接口定义来添加用户需要的功能。

3.2.1 电源插座和开关

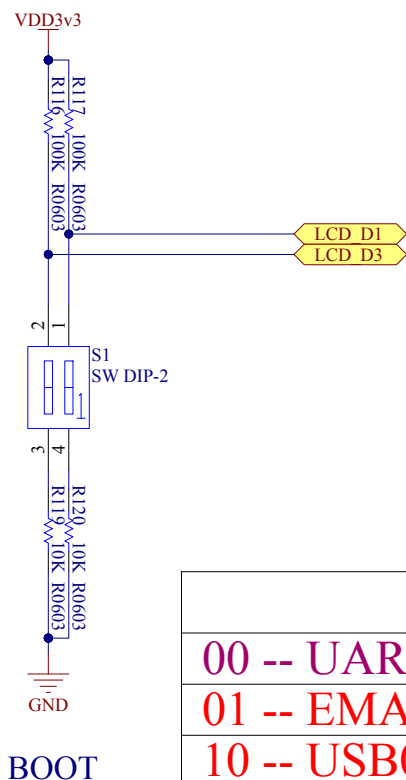
OK335xD开发板采用+5V DC来给核心板的DC5V引脚供电。板上J3为电源插座，直接连接配套的5V2A规格的电源适配器。对于+5V DC供电部分，单独给核心板供电的稳定电流为几百mA，但开机启动瞬间可能产生超过1A的电流，尤其是同时底板接显示屏的情况下，所以推荐使用5V2A及以上规格的电源适配器。电源插座旁边的拨动开关S2为电源开关，控制开发板电源的通断。



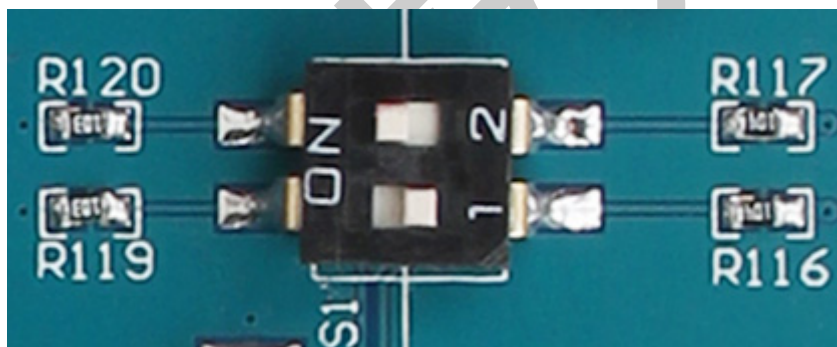
在电源供电处建议使用滤波电容。

3.2.2 boot配置引脚

AM335x系列有多种启动方式，在系统上电或复位后，通过读取系统启动配置引脚的状态，选择不同的启动方式。具体可参考AM335x的数据手册和本手册核心板部分。OK335xD开发板使用SYSBOOT1，SYSBOOT3两个选择位（原理图中分别为LCD_D1,LCD_D3）选择不同的启动方式，在底板上通过一个2位的拨码开关进行配置，电路图及照片如下所示：



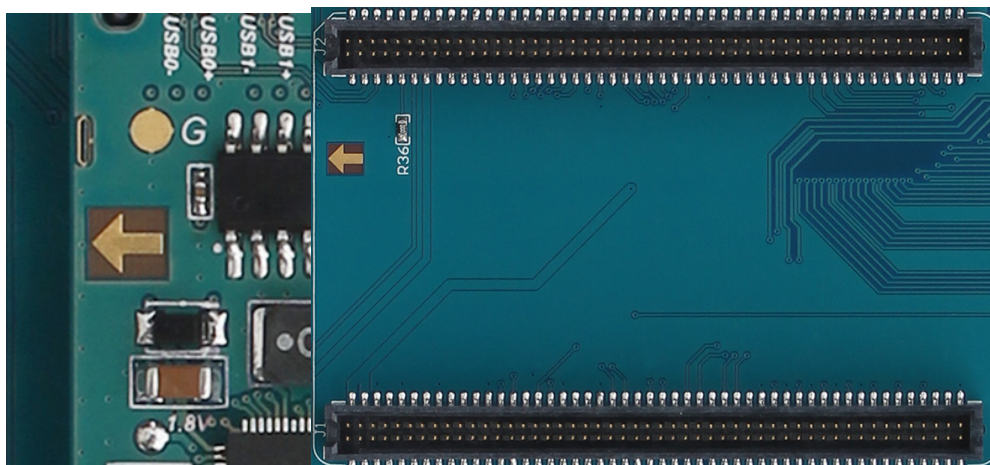
[1 3]
00 -- UART0_XIP MMC0 NAND
01 -- EMAC1_SPI0 NAND NANDI2C
10 -- USB0 NAND_XIP NANDI2C



3.2.3 核心板连接器

OK335xD开发板底板上的J1, J2为连接FET335xD核心板的接口，接口定义与核心板一致，具体可参考核心板的引脚定义。

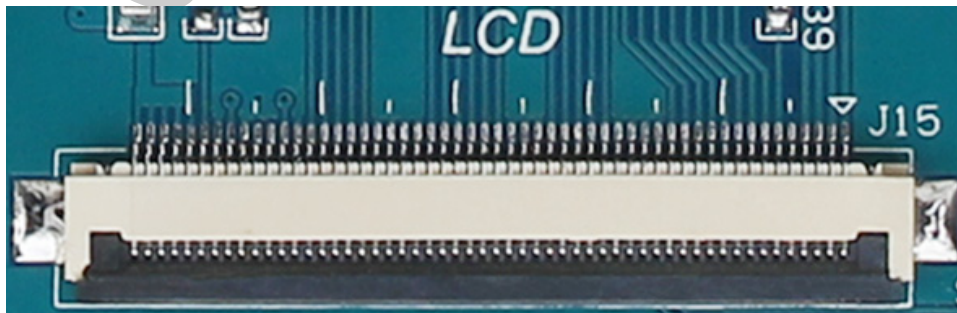
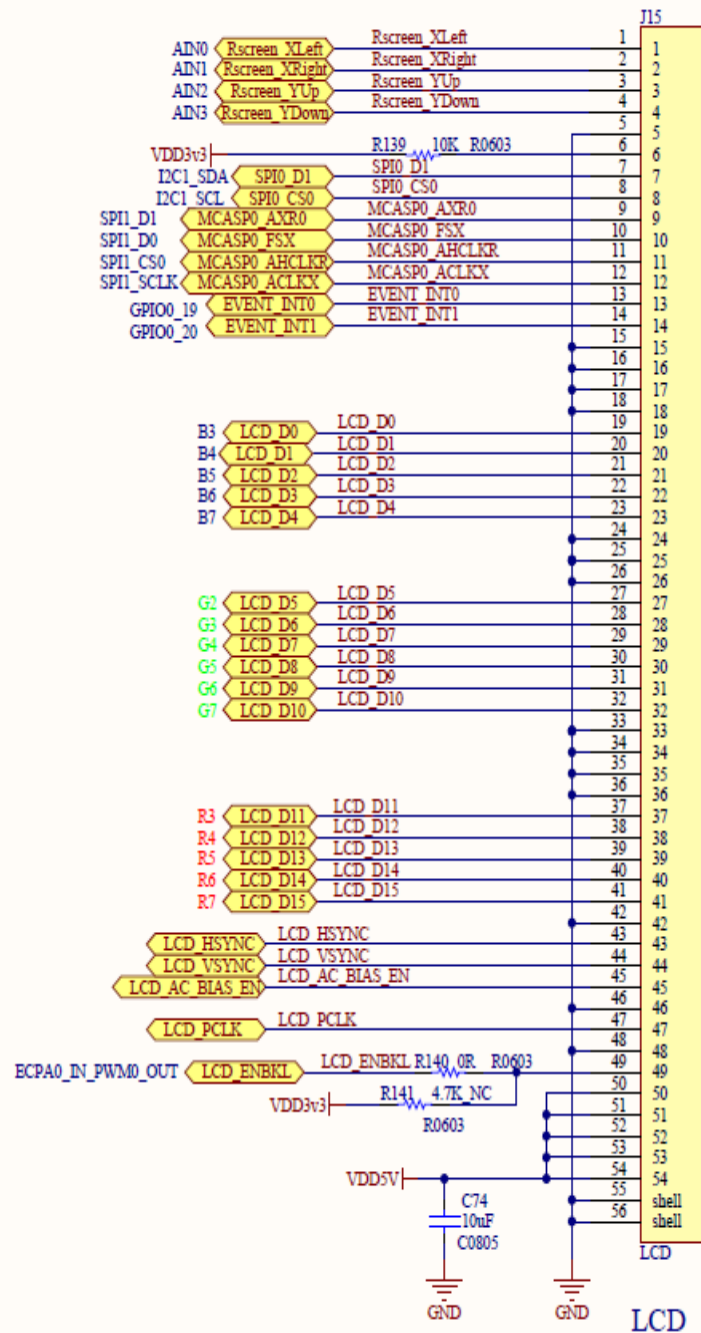
在FET335xD核心板与底板的连接上，采用通用插针的方式。当拔插核心板时，务必注意不要插反，否则可能会损坏OK335xD开发板及核心板。开发板上带有方向指示，如下图核心板（左）和底板（右）方向指示：



3.2.4 LCD液晶显示接口

OK335xD开发板提供了一组通用的液晶显示接口，通过一个54pin的FPC卡座引出，可连接飞凌公司生产的不同尺寸规格的电阻液晶屏和电容液晶屏。

由原理图中LCD部分可知OK335xD底板采用的是RGB565 16Bit模式的接法，电路已经将其他不用的数据引脚接地。如果想使用RGB888 24Bit模式接法的话，请确定多使用的8个数据引脚（LCD_16 ~ LCD_23）未因引脚复用功能而被其他功能占用。另需注意，现有软件驱动默认为RGB565 16Bit模式驱动，未做RGB888 24Bit模式的驱动。两模式显示效果无大差异，若无特殊要求建议采用默认模式接法。



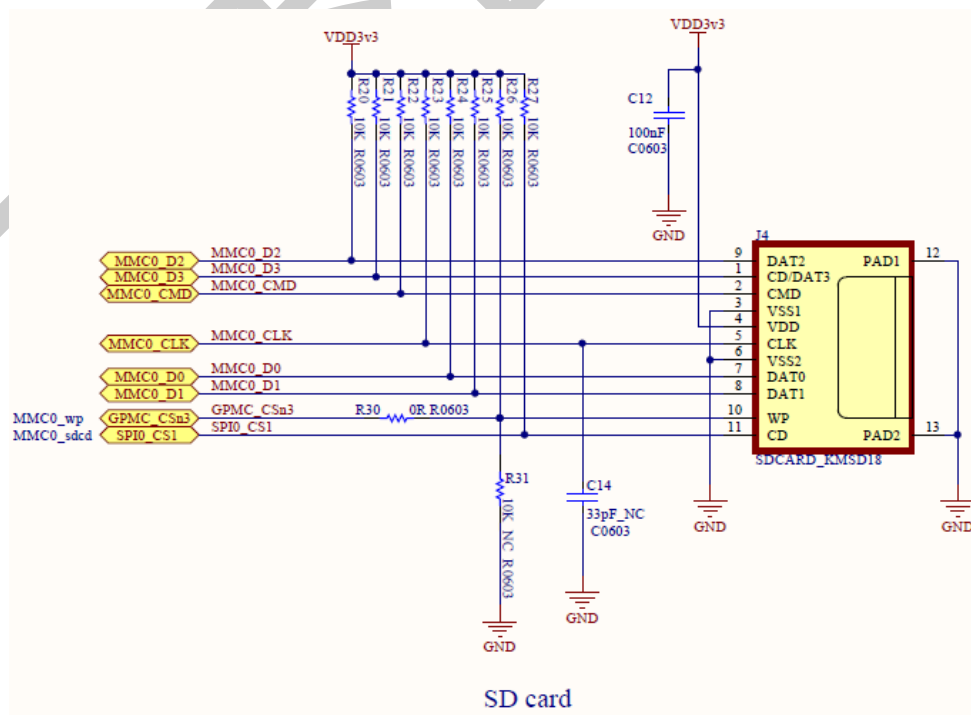
下面详细介绍该接口各个引脚的功能:

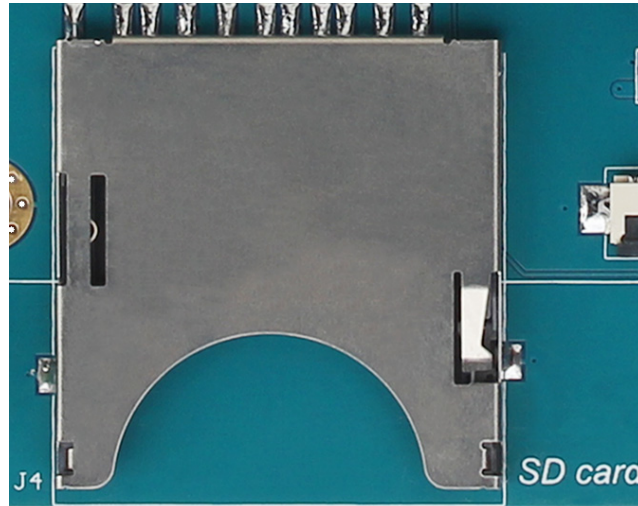
引脚序号	LCD接口信号名称	对应核心板引脚	功能描述	类型
1	AIN0	Rscreen_XLeft	电阻触摸屏X正信号	ANALOG
2	AIN1	Rscreen_XRight	电阻触摸屏X负信号	ANALOG
3	AIN2	Rscreen_YUp	电阻触摸屏Y正信号	ANALOG
4	AIN3	Rscreen_YDown	电阻触摸屏Y负信号	ANALOG
5	GND	GND	电源地	POWER
6	VDD3V3	NC	电源	(3.3V)
7	I2C1_SDA	SPI0_D1	I2C串行接口时钟信号	LVC MOS(3.3V)
8	I2C1_SCL	SPI0_CS0	I2C串行接口数据信号	LVC MOS(3.3V)
9	SPI1_D1	MCASP0_AXR0	SPI串行接口主入从出信号	LVC MOS(3.3V)
10	SPI1_D0	MCASP0_FSX	SPI串行接口主出从入信号	LVC MOS(3.3V)
11	SPI1_CS0	MCASP0_AHCLK R	SPI串行接口片选信号	LVC MOS(3.3V)
12	SPI1_SCLK	MCASP0_ACLKX	SPI串行接口时钟信号	LVC MOS(3.3V)
13	GPIO0_19	EVENT_INT0	中断信号1	LVC MOS(3.3V)
14	GPIO0_20	EVENT_INT1	中断信号2	LVC MOS(3.3V)
15	GND	GND	电源地	POWER
16	GND	GND	电源地	POWER
17	GND	GND	电源地	POWER
18	GND	GND	电源地	POWER
19	B3	LCD_D0	RGB接口蓝色数据位3	LVC MOS(3.3V)
20	B4	LCD_D1	RGB接口蓝色数据位4	LVC MOS(3.3V)
21	B5	LCD_D2	RGB接口蓝色数据位5	LVC MOS(3.3V)
22	B6	LCD_D3	RGB接口蓝色数据位6	LVC MOS(3.3V)
23	B7	LCD_D4	RGB接口蓝色数据位7	LVC MOS(3.3V)
24	GND	GND	电源地	POWER
25	GND	GND	电源地	POWER
26	GND	GND	电源地	POWER
27	G2	LCD_D5	RGB接口绿色数据位2	LVC MOS(3.3V)
28	G3	LCD_D6	RGB接口绿色数据位3	LVC MOS(3.3V)
29	G4	LCD_D7	RGB接口绿色数据位4	LVC MOS(3.3V)
30	G5	LCD_D8	RGB接口绿色数据位5	LVC MOS(3.3V)
31	G6	LCD_D9	RGB接口绿色数据位6	LVC MOS(3.3V)
32	G7	LCD_D10	RGB接口绿色数据位7	LVC MOS(3.3V)
33	GND	GND	电源地	POWER
34	GND	GND	电源地	POWER
35	GND	GND	电源地	POWER
36	GND	GND	电源地	POWER
37	R3	LCD_D11	RGB接口红色数据位3	LVC MOS(3.3V)
38	R4	LCD_D12	RGB接口红色数据位4	LVC MOS(3.3V)
39	R5	LCD_D13	RGB接口红色数据位5	LVC MOS(3.3V)

40	R6	LCD_D14	RGB接口红色数据位6	LVC MOS(3.3V)
41	R7	LCD_D15	RGB接口红色数据位7	LVC MOS(3.3V)
42	GND	GND	电源地	POWER
43	LCD_HSYNC	LCD_HSYNC	RGB接口水平扫描信号	LVC MOS(3.3V)
44	LCD_VSYNC	LCD_VSYNC	RGB接口垂直扫描信号	LVC MOS(3.3V)
45	LCD_AC_BIAS_EN	LCD_AC_BIAS_EN	RGB接口数据使能信号	LVC MOS(3.3V)
46	GND	GND	电源地	POWER
47	LCD_PCLK	LCD_PCLK	RGB接口像素时钟信号	LVC MOS(3.3V)
48	GND	GND	电源地	POWER
49	ECPA0_IN_PWM0_OUT	LCD_ENBKL	PWM背光调节信号	LVC MOS(3.3V)
50	VDD5V	DC5V	5V直流电源接口	POWER
51	VDD5V	DC5V	5V直流电源接口	POWER
52	VDD5V	DC5V	5V直流电源接口	POWER
53	VDD5V	DC5V	5V直流电源接口	POWER
54	VDD5V	DC5V	5V直流电源接口	POWER
55	LCD接口定位架1	NC	电源地	POWER
56	LCD接口定位架2	NC	电源地	POWER

注：本扩展底板中的I2C1_SDA、I2C1_SCL两个信号对应核心板引脚为SPI0_D1、SPI0_CS0，核心板的这两个引脚在配置上没有用作SPI0通信，而是用了同引脚上的I2C1的功能。

3.2.5 SD卡

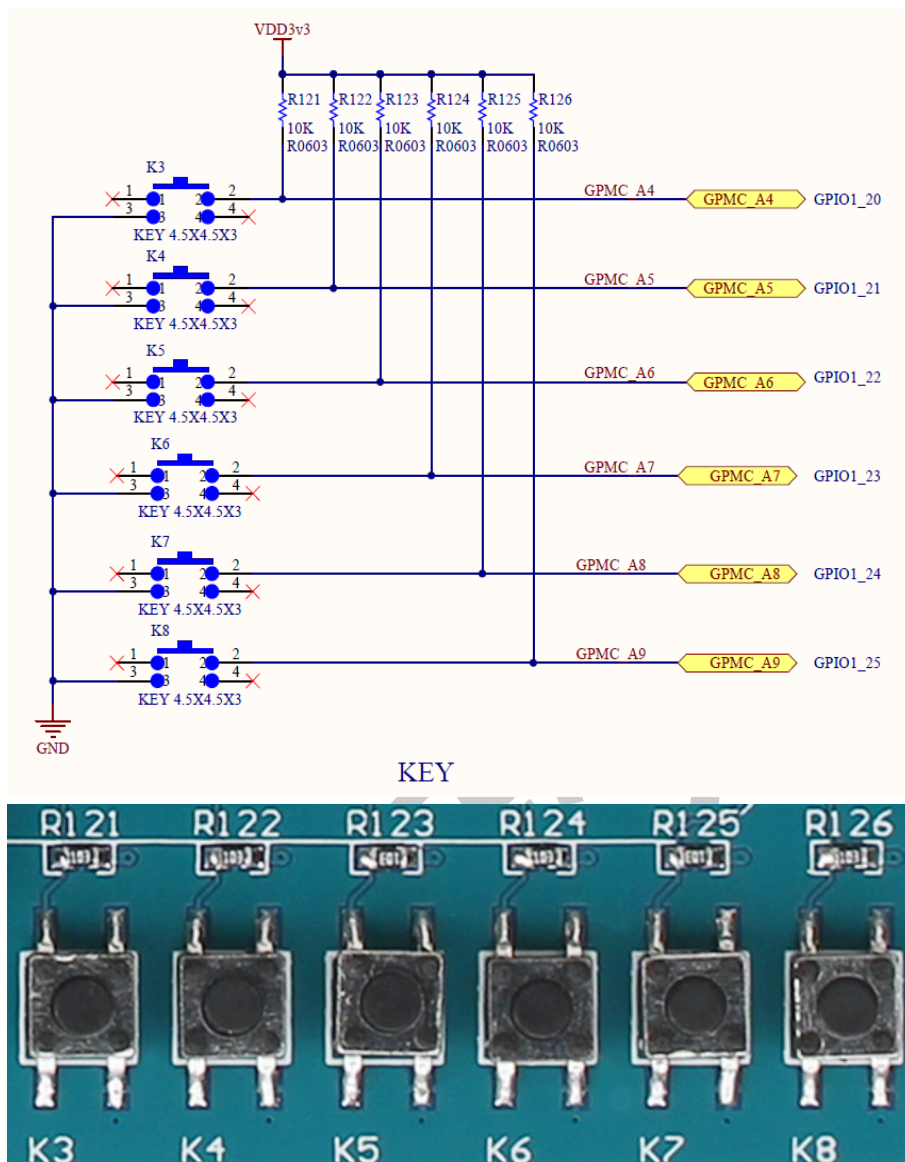




J4为SD卡接口，该接口兼容SD卡和/MMC卡，该接口有11个功能引脚，参照上图。该接口连接到了AM335X的MMC0接口，下面详细介绍该接口的引脚功能：

引脚序号	SD卡接口信号名称	对应核心板引脚	功能描述	类型
1	MMC0_D3	MMC0_D3	双向数据信号位3	LVC MOS(3.3V)
2	MMC0_CMD	MMC0_CMD	双向命令/响应信号	LVC MOS(3.3V)
3	GND	GND	电源地	POWER
4	VDD	VDD3.3V	电源	POWER
5	MMC0_CLK	MMC0_CLK	主机输入到SD卡的时钟信号	LVC MOS(3.3V)
6	GND	GND	电源地	POWER
7	MMC0_D0	MMC0_D0	双向数据信号位0	LVC MOS(3.3V)
8	MMC0_D1	MMC0_D1	双向数据信号位1	LVC MOS(3.3V)
9	MMC0_D2	MMC0_D2	双向数据信号位2	LVC MOS(3.3V)
10	MMC0_wp	GPMC_CSn3	SD卡座向主机输入的写保护状态信号	LVC MOS(3.3V)
11	MMC0_scd	SPI0_CS1	SD卡片选使能信号，低电平有效	LVC MOS(3.3V)

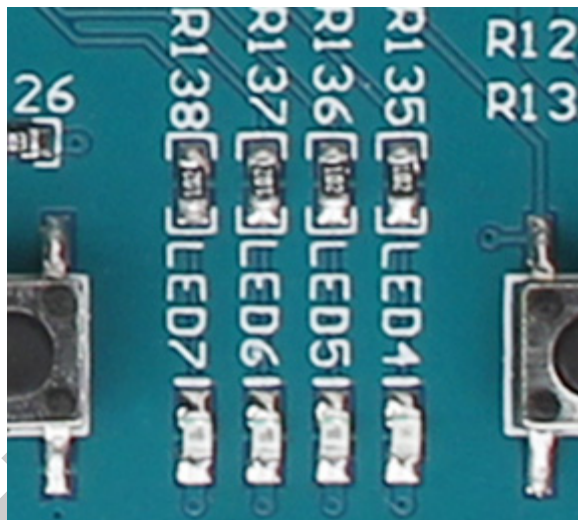
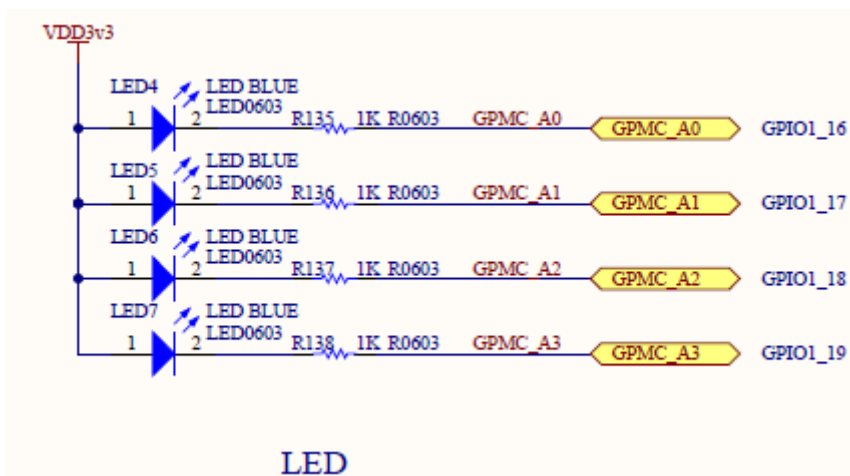
3.2.6 板载用户按键



如图所示，OK335xD开发板提供了6个用户按键，分别为K3、K4、K5、K6、K7、K8，用户开发自己的应用程序时可使用这些按键。该组按键连接到AM335x的外部中断引脚，在按键抬起状态下通过上拉电阻上拉到高电平，按键按下是为低电平。下面详细介绍按键的连接情况。

按键序号	按键接口信号名称	对应核心板引脚	功能描述	类型
1	K3	GPMC_A4	AM335x的GPIO1_20	LVC MOS(3.3V)
2	K4	GPMC_A5	AM335x的GPIO1_21	LVC MOS(3.3V)
3	K5	GPMC_A6	AM335x的GPIO1_22	LVC MOS(3.3V)
4	K6	GPMC_A7	AM335x的GPIO1_23	LVC MOS(3.3V)
5	K7	GPMC_A8	AM335x的GPIO1_24	LVC MOS(3.3V)
6	K8	GPMC_A9	AM335x的GPIO1_25	LVC MOS(3.3V)

3.2.7 演示LED



如图所示，OK335xD开发板提供4个LED指示灯，分别为LED4、LED5、LED6、LED7，该组LED指示灯分别连接到了AM335x的GPIO1_16、GPIO1_17、GPIO1_18、GPIO1_19引脚，使用LED指示灯时需要先将这种引脚配置为GPIO功能，引脚输出低电平时点亮LED指示灯，引脚输出高电平时，关闭LED指示灯。下面详细介绍LED的连接情况：

LED序号	LED接口信号名称	对应核心板引脚	功能描述	类型
1	LED4	GPMC_A0	AM335x的 GPIO1_16	LVC MOS(3.3V)
2	LED5	GPMC_A1	AM335x的 GPIO1_17	LVC MOS(3.3V)
3	LED6	GPMC_A2	AM335x的 GPIO1_18	LVC MOS(3.3V)
4	LED7	GPMC_A3	AM335x的 GPIO1_19	LVC MOS(3.3V)

3.2.8 ADC模数转换

OK335xD开发板提供一个2.0mm 2X5PIN的ADC接口，该接口引出了8路模拟输入通道，用于客户扩展功能使用。ADC接口的引脚顺序如下：白色三角形指示位置处，靠近白色三角形的引脚为第一引脚，第一

引脚序号	ADC接口信号名称	对应核心板引脚	功能描述	类型
1	AIN7	AIN7	AM335x的的ADC模拟输入通道7	ANALOG
2	AIN0	Rscreen_XRight	AM335x的ADC模拟输入通道0（于LCD4线电阻触摸屏共用）	ANALOG
3	AIN6	AIN6	AM335x的的ADC模拟输入通道6	ANALOG
4	AIN1	Rscreen_XLeft	AM335x的ADC模拟输入通道1（于LCD4线电阻触摸屏共用）	ANALOG
5	AIN5	AIN5	AM335x的的ADC模拟输入通道5	ANALOG
6	AIN2	Rscreen_YDown	AM335x的ADC模拟输入通道2（于LCD4线电阻触摸屏共用）	ANALOG
7	AIN4	AIN4	AM335x的的ADC模拟输入通道4	ANALOG
8	AIN3	Rscreen_YUp	AM335x的ADC模拟输入通道3（于LCD4线电阻触摸屏共用）	ANALOG
9	GND	GND	电源地	POWER
10	GND	GND	电源地	POWER

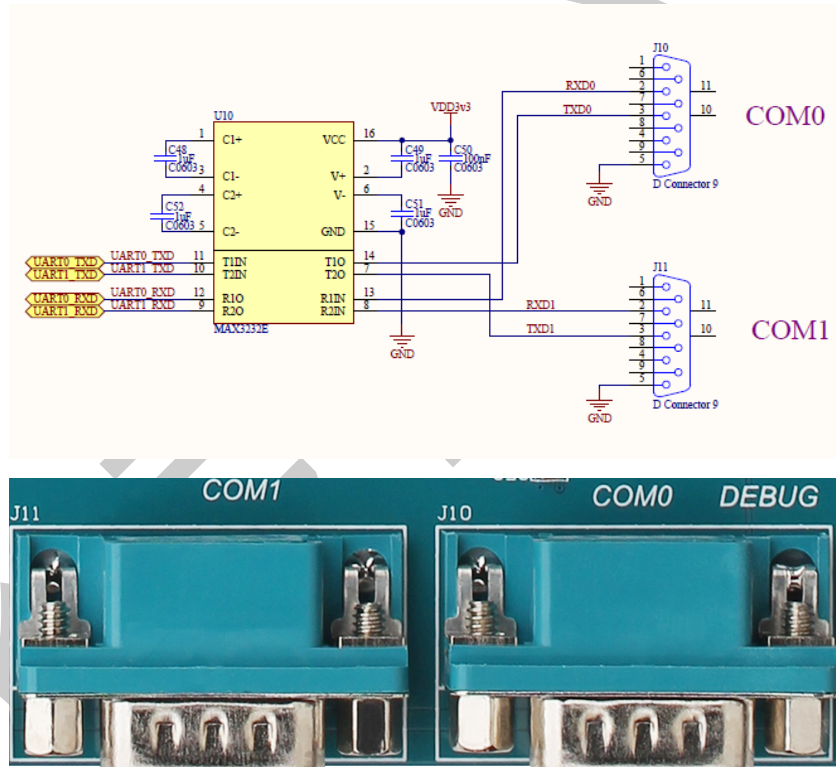
3.2.9 音频接口

The schematic diagram illustrates the connection of the ILV30AC3108 module to a microcontroller (U1). The microcontroller is connected to the module's pins for power, reset, SPI, I2C, and various control signals. The module includes a 100MHz oscillator (L8), a 3.3V regulator (R101), and a 1.8V regulator (R102). It also features a 100MHz oscillator (L9), a 3.3V regulator (R103), and a 1.8V regulator (R104). The module is connected to a PHONE jack (J12), a MIC jack (J13), and a LINE IN jack (J14).

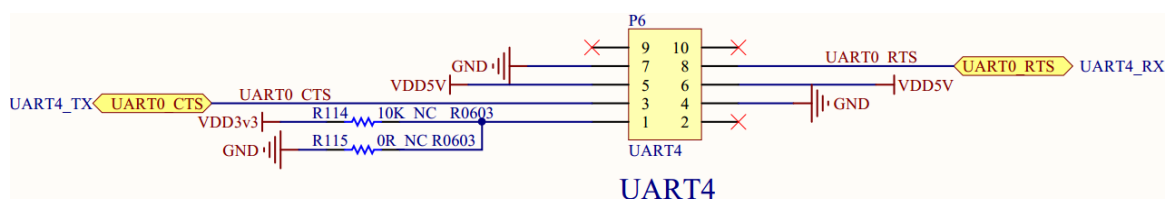


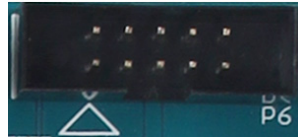
3.2.10 UART串口

OK335xD开发板提供两路RS232电平的通用异步串行接口，J10对应的为COM0，为系统调试串口，对应AM335x的UATR0；J11对应的为COM1，对应AM335x的UART1，用于用户扩展功能。这两个串行接口采用标准的DB9连接器。电平转换芯片采用MAX232ESE。引脚连接关系图片如下图所示：



为了提高开发板的使用灵活性，除了上述两路RS232电平的通用异步串行接口，还提供了一路3.3V电平接口的UART，对应AM335x的UART4，方便用户连接3.3V电平的串口设备，下面详细介绍此接口的引脚定义。（白色三角处为第一个引脚）





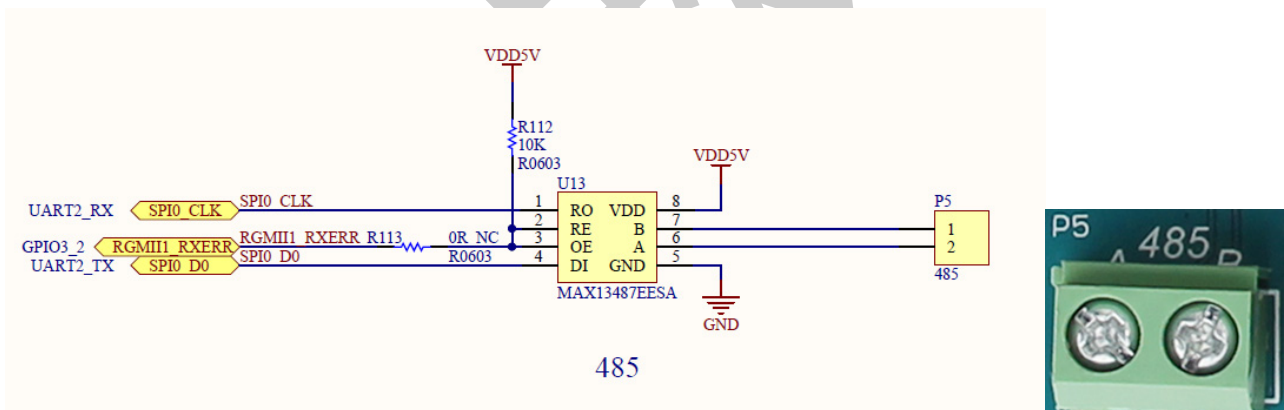
UART4接口引脚定义：

引脚序号	UART0接口信号名称	对应核心板引脚	功能描述	类型
3	UART4_TX	UART0_CTS	UART4接口数据发送信号	LVC MOS(3.3V)
4	GND	GND	电源地	POWER
5	VDD5V	DC5V	DC5V电源	POWER
6	VDD5V	DC5V	DC5V电源	POWER
7	GND	GND	电源地	POWER
8	UART4_RX	UART0_RTS	UART4接口数据接收信号	LVC MOS(3.3V)
1, 2, 9, 10	NC	--	--	--

3.2.11 RS-485接口

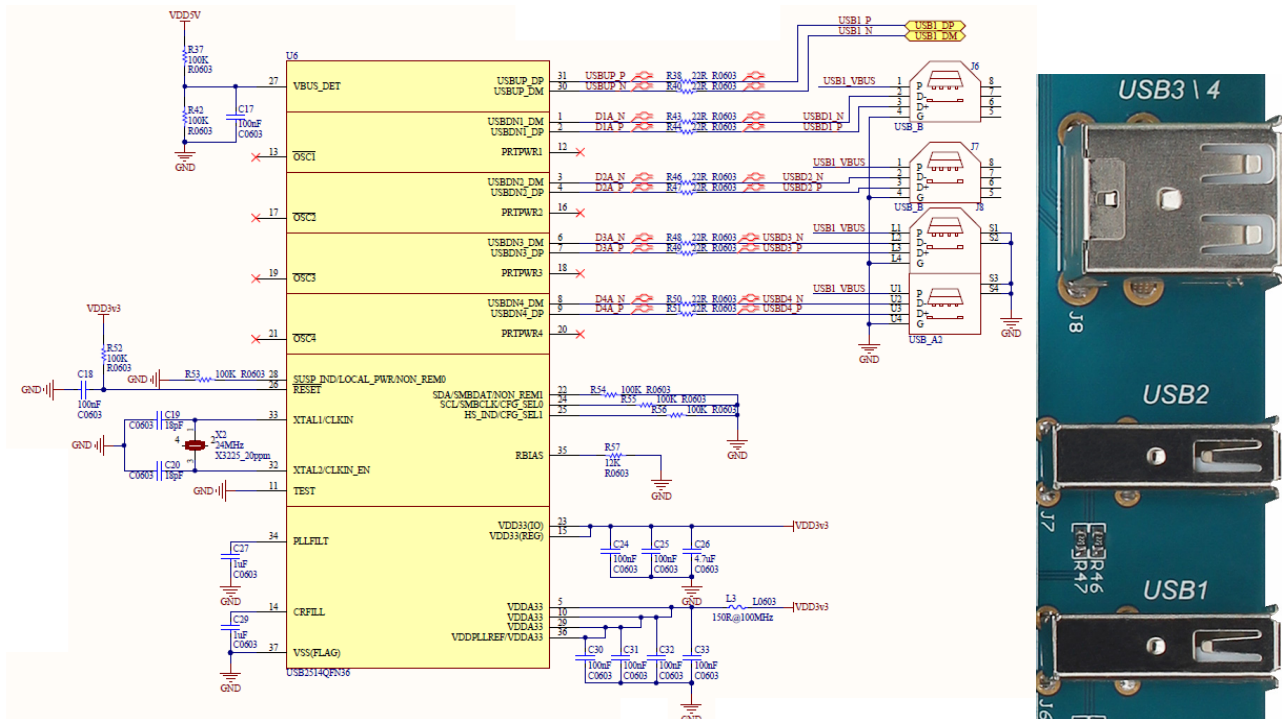
OK335xD开发板提供一路485接口，在485接口电路上，使用了AM335x的UART2。目前开发板已改为使用MAX485芯片（芯片引脚定义相同），对应软件驱动也为MAX485驱动，工作过程中控制收发数据由GPIO3_2来完成（R113的0R电阻需焊接）。

如下图所示：（MAX13487改为MAX485，电阻R113需焊接）



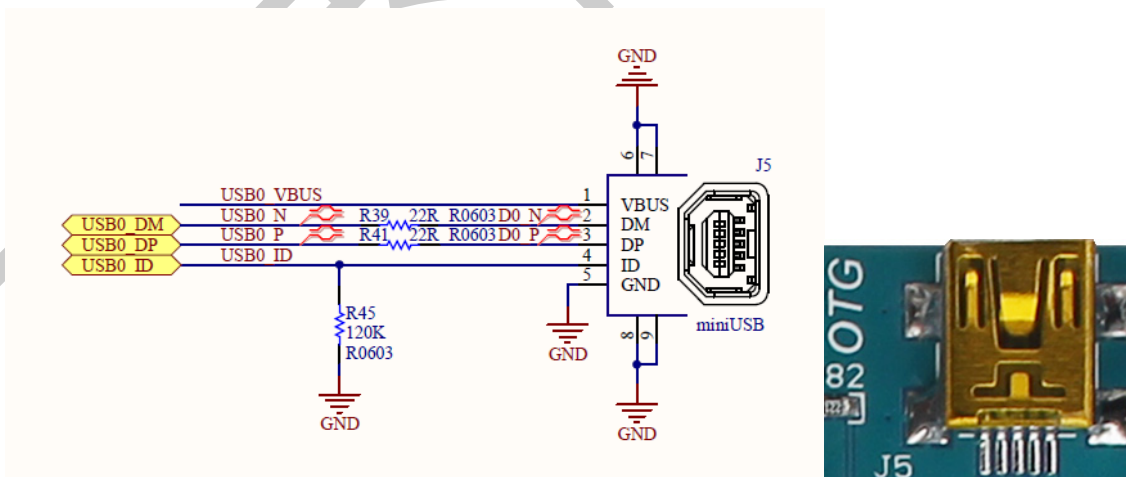
3.2.12 USB 2.0 HOST接口

如下图所示，OK335xD开发板将AM335x芯片的USB HOST接口（XUHDP，XUHDM）通过USB2514B芯片扩展出4个USB2.0 HOST接口，扩展了AM335x的USB HOST的数量，增强了OK335xD开发板开发板的整体性能。这四个USB HOST可以挂接一些诸如U盘、USB鼠标、USB键盘等通用的USB设备及飞凌公司生产的使用USB接口的3G、GPRS等模块。



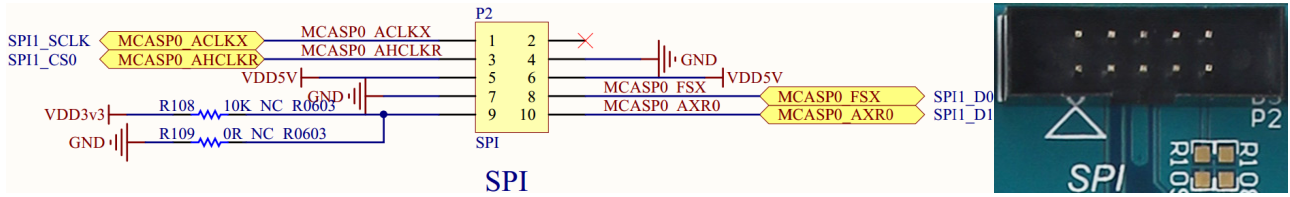
3.2.13 USB 2.0 OTG接口

OK335xD开发板上的J5对应USB 2.0 OTG接口，该接口是采用标准的Mini USB插座。USB OTG是USB On-The-Go的缩写，是近年发展起来的技术，On-The-Go，即OTG技术就是实现在没有Host的情况下，实现设备间的数据传送。例如数码相机直接连接到打印机上，通过OTG技术，连接两台设备间的USB口，将拍出的相片立即打印出来；也可以将数码照相机中的数据，通过OTG发送到USB接口的移动硬盘上，野外操作就没有必要携带价格昂贵的存储卡，或者背一个便携电脑。OK335xD开发板上USB 2.0 OTG如下所示：



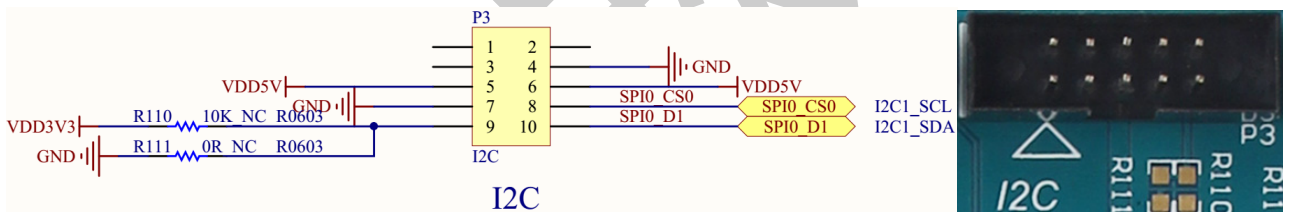
3.2.14 SPI 接口

OK335xD开发板提供了一路SPI接口，该接口对应的是AM335x的SPI1，另外，该接口还预留了电源、地等信号，方便用户的扩展。详细的引脚功能描述见下表。（白色三角处为第一个引脚）



引脚序号	SPI0接口信号名称	对应核心板引脚	功能描述	类型
1	SPI1_SCLK	MCASP0_ACLKX	SPI串行接口时钟输出信号	LVC MOS(3.3V)
2	NC			
3	SPI1_CS0	MCASP0_AHCLKR	SPI串行接口片选信号	LVC MOS(3.3V)
4	GND	GND	电源地	POWER
5	VDD	VDDIN	DC5V电源	POWER
6	VDD	VDDIN	DC5V电源	POWER
7	GND	GND	电源地	POWER
8	SPI1_D0	MCASP0_FSX	SPI串行接口主机输出从机输入信号	LVC MOS(3.3V)
9	逻辑电平预留	--	SPI0接口预留使能信号	
10	SPI1_D1	MCASP0_AXR0	SPI串行接口主机输入从机输出信号	LVC MOS(3.3V)

3.2.15 I2C接口

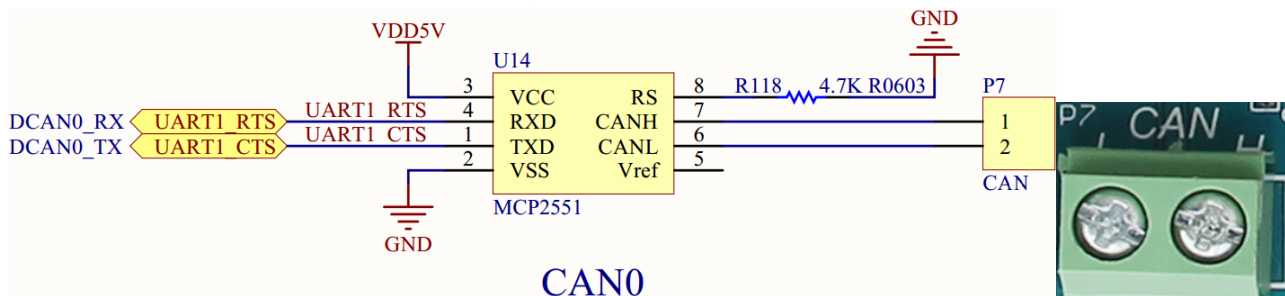


如上图所示，OK335xD开发板引出了一路I2C串行接口，I2C1。另外，该接口还预留了电源、地等信号，方便用户的扩展。详细的引脚功能描述见下表。

引脚序号	I2C接口信号名称	对应核心板引脚	功能描述	类型
1	NC	--	--	--
2	NC	--	--	--
3	NC	--	--	--
4	GND	GND	电源地	POWER
5	VDD5V	DC5V	DC5V电源	POWER
6	VDD5V	DC5V	DC5V电源	POWER
7	GND	GND	电源地	POWER
8	I2C1_SCL	SPI0_CS0	I2C串行接口1串行数据信号	LVC MOS(3.3V)
9	逻辑电平预留	--	SPI0接口预留使能信号	LVC MOS(3.3V)
10	I2C1_SDA	SPI0_D1	I2C串行接口1时钟输出信号	LVC MOS(3.3V)

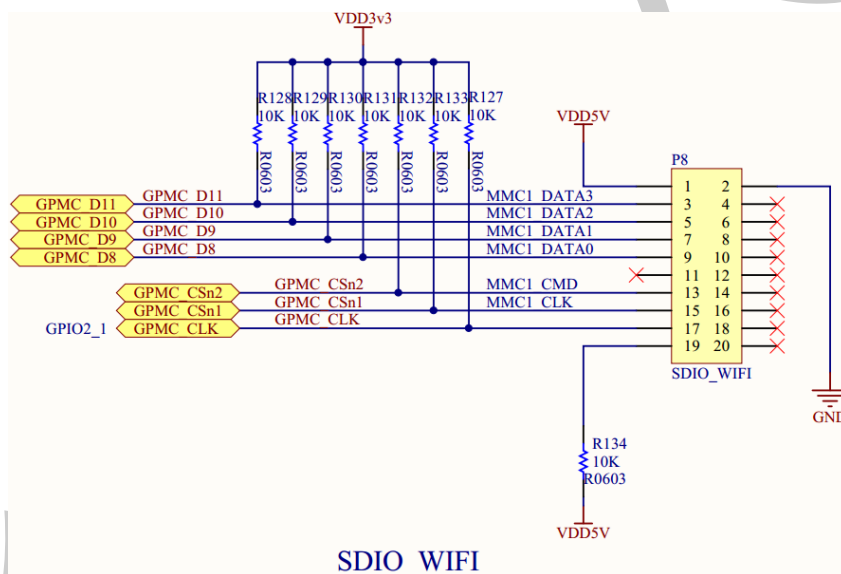
3.2.16 CAN接口

OK335xD开发板提供一路CAN接口，使用了AM335x的CAN0。如下图所示。



OK335xD开发板在FET335x核心板上引出CAN0接口，通过MCP2551进行CAN总线的物理连接，MCP2551支持高达1Mb/s的通讯速率，最多可连接112个CAN设备。

3.2.17 SDIO WiFi接口



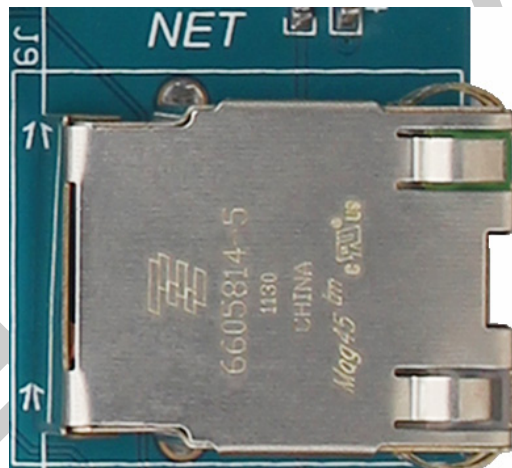
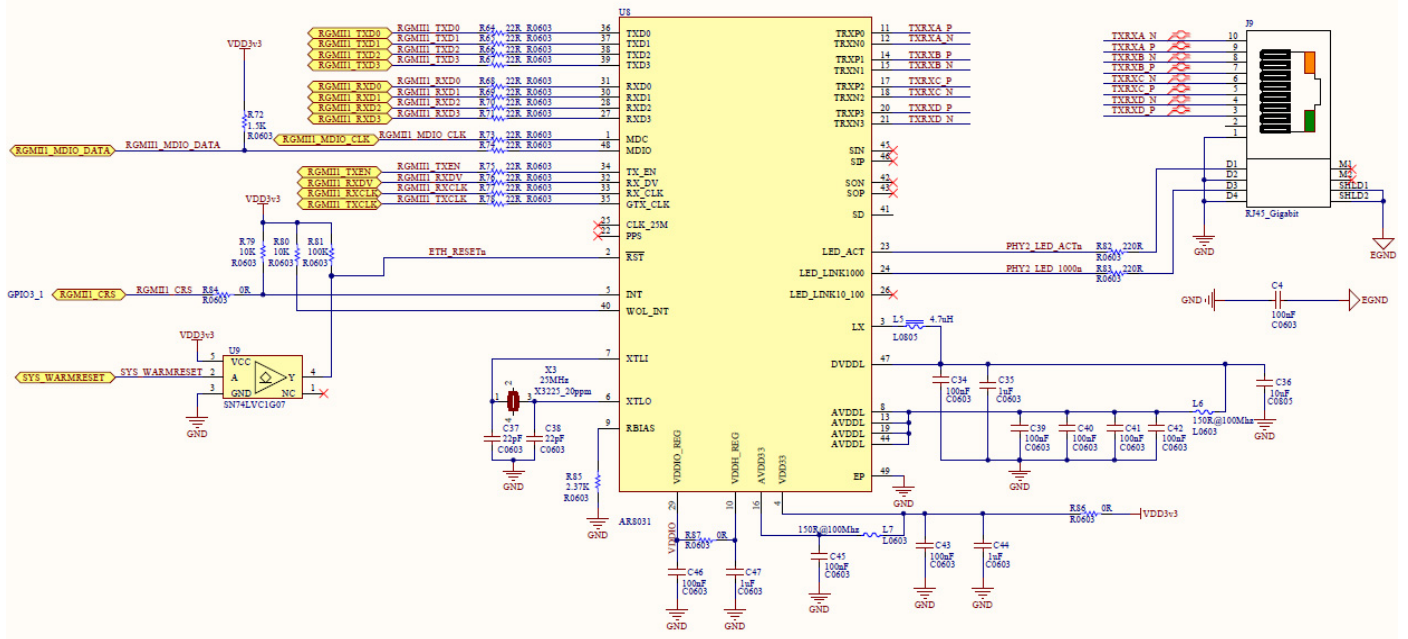
OK335xD开发板引出SD/MMC1作为一路SDIO接口，采用四位的总线宽度，该接口默认是配合飞凌公司的SDIO WiFi模块使用的，用户也可以根据该接口的引脚定义来连接自己的SDIO模块，也可以连接SD、MMC卡。详细的引脚说明见下表。

引脚序号	SDIO接口信号名称	对应核心板引脚	功能描述	类型
1	VDD5V	DC5V	DC5V电源信号	POWER

2	GND	GND	电源地	POER
3	MMC1_DATA3	GPMC_D11	SDIO接口数据位3信号	LVMOS(3.3V)
4	NC	NC	无连接	NC
5	MMC1_DATA2	GPMC_D10	SDIO接口数据位2信号	LVMOS(3.3V)
6	NC	NC	无连接	NC
7	MMC1_DATA1	GPMC_D9	SDIO接口数据位1信号	LVMOS(3.3V)
8	NC	NC	无连接	NC
9	MMC1_DATA0	GPMC_D8	SDIO接口数据位0信号	LVMOS(3.3V)
10	NC	NC	无连接	NC
11	NC	NC	无连接	NC
12	NC	NC	无连接	NC
13	MMC1_CMD	GPMC_CS _{n2}	SDIO接口命令指示信号	LVMOS(3.3V)
14	NC	NC	无连接	NC
15	MMC1_CLK	GPMC_CS _{n1}	SDIO接口时钟信号	LVMOS(3.3V)
16	NC	NC	无连接	NC
17	INT	GPMC_CLK	SDIO接口预留中断信号	LVMOS(3.3V)
18	NC	NC	无连接	NC
19	EN	XPWRRGTON	SDIO接口预留使能引脚	5V拉高
20	NC	NC	无连接	NC

3.2.18 千兆以太网

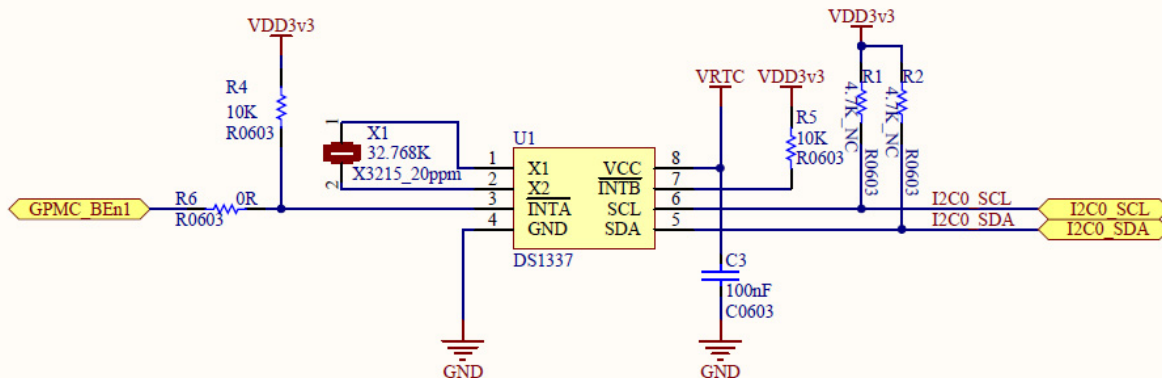
AM335x上最多可提供2路千兆以太网，考虑到其引脚存在复用情况，故在OK335xD开发板上为了尽可能多的实现功能，只引出了一路千兆以太网，使用RGMII接口形式，在底板上使用AR8031实现物理层的连接，如下图所示：



其他更为详细的信息请参考OK335xD开发板原理图。

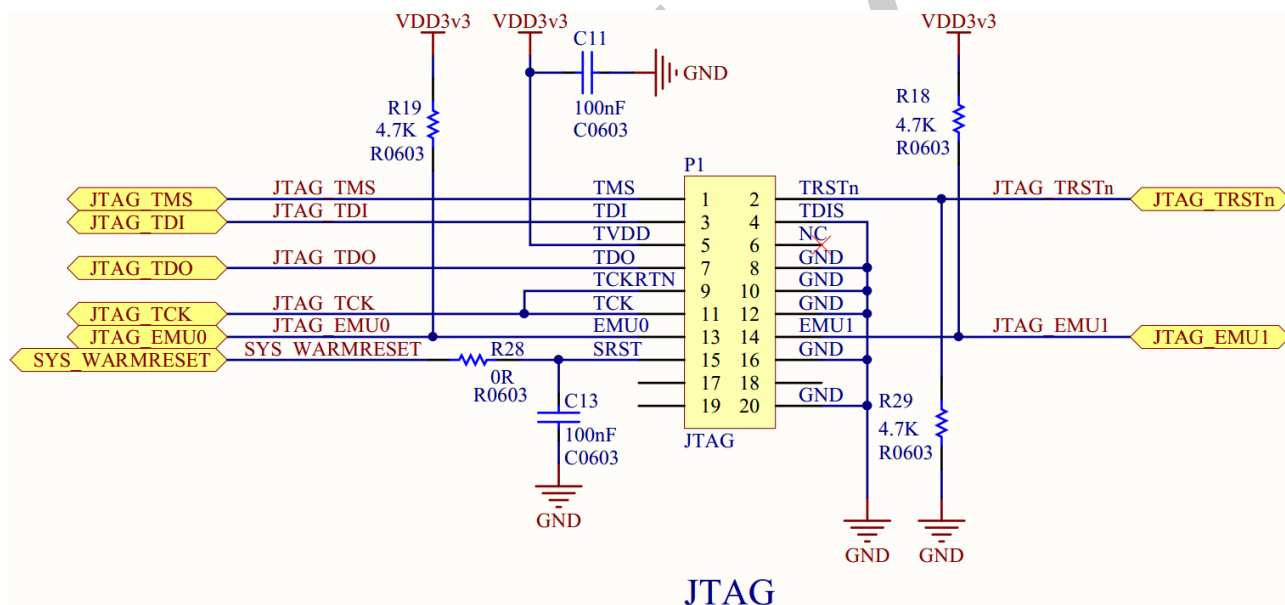
3.2.19 板载实时时钟

为了使OK335xD开发板发挥更大的实用性，弥补AM335x的不足，开发板上额外增加了一路实时时钟，使用时钟芯片DS1337实现，这样开发板上既可以使用AM335x自带的实时时钟功能，也可以使用DS1337实现实时时钟的管理，具体电路如下图



3.2.20 JTAG仿真调试接口

OK335xD开发板为用户提供了JTAG调试接口，方便用户通过仿真设备对AM335x进行仿真调试，仿真调试环境为TI提供的StartWare。对于此部分功能请您参考OK335xD开发板原理图和AM335x的数据手册。（白色三角处为第一个引脚）



引脚序号	JTAG接口信号名称	对应核心板引脚	功能描述	类型
1	TMS	JTAG_TMS	JTAG测试模式选择	LVC MOS(3.3V)
2	TRSTn	JTAG_TRSTn	JTAG测试系统复位信号	LVC MOS(3.3V)
3	TDI	JTAG_TDI	JTAG测试数据串行输入	LVC MOS(3.3V)

4	TDIS	--	接地	GND
5	TVDD	--	接电源	POWER
6	NC	NC	无连接	NC
7	TDO	JTAG_TDO	JTAG测试数据串行输出	LVC MOS(3.3V)
8	GND	--	接地	GND
9	TCKRTN	JTAG_TCK	JTAG测试时钟	LVC MOS(3.3V)
10	GND	--	接地	GND
11	TCK	JTAG_TCK	JTAG测试时钟	LVC MOS(3.3V)
12	GND	--	接地	GND
13	EMU0	JTAG_EMU0	MISC 仿真0	LVC MOS(3.3V)
14	EMU1	JTAG_EMU1	MISC 仿真1	LVC MOS(3.3V)
15	SRST	SYS_WARMRESET	系统复位	LVC MOS(3.3V)
16	GND	--	接地	GND
17	NC	NC	无连接	NC
18	NC	NC	无连接	NC
19	NC	NC	无连接	NC
20	GND	--	接地	GND

附录表

F1 FET335xD核心板引脚全功能定义

FET335xD核心板连接器引出接口（包括所引出引脚的全部复用功能）的引脚定义。用户在有多种功能扩展需求时可参阅此表做一个简单的了解，但若需了解更详细的情况建议用户查阅相关资料文档及芯片数据手册及参考手册等。编者水平有限，文中若有纰漏处还望批评指正，如对文档有疑问不明之处，请联系飞凌的技术人员进行分析解决。

J1

Pin	Ball	Pin Name	V	描述	
1	NC	DC5V	5	电源输入	
2	NC	DC5V	5	电源输入	
3	NC	DC5V	5	电源输入	
4	NC	DC5V	5	电源输入	
5	NC	GND	0	地	
6	NC	GND	0	地	
7	NC	USB_DC	5	USB电源输入	
8	NC	USB_DC	5	USB电源输入	
9	NC	GND	0	地	
10	NC	GND	0	地	
11	NC	BAT	3.7	锂电池供电输入	
12	NC	BAT	3.7	锂电池供电输入	
13	NC	BAT_SENSE	3.3	电池传感器输入	
14	NC	BAT_TS	3.3	电池温度传感器输入	
15	NC	GND	0	地	
16	NC	GND	0	地	
17	NC	PWR_BUT	3.3	按键输入(低有效)	
18	NC	BL_OUT	3.3	WLED反馈	
19		NC			
20	NC	ISINK1	3.3	WLED SINK1电流输入	
21	NC	GND	0	地	
22	NC	ISINK2	3.3	WLED SINK2电流输入	
23	NC	BAT_RTC	3.3	RTC电池输入	
24	NC	GND	0	地	
25	NC	GND	0	地	
26	E18	UART0_CTS	0	3.3	UART0 清除发送
		uart4_rxd	1	3.3	UART4 接收
		dcan1_tx	2	3.3	DCAN1 发送
		I2C1_SDA	3	3.3	I2C1 数据
		spi1_d0	4	3.3	SPI1 数据0

		timer7	5	3.3	定时器7/PWM输出
		prl_edc_sync0_out *	6	3.3	PRU_edc_sync0_out 数据输出
		gpio1_8	7	3.3	通用IO
27	F17	MMC0_D3	0	3.3	MMC/SD/SDIO 数据3
		gpmc_a20	1	3.3	GPMC地址20
		uart4_ctsn	2	3.3	UART 清除发送
		timer5	3	3.3	定时器5/PWM输出
		uart1_dcdn	4	3.3	UART1 数据载波检测
		prl_pru0_pru_r30_8*	5	3.3	PRU 数据输出
		prl_pru0_pru_r31_8*	6	3.3	PRU 数据输入
		gpio2_26	7	3.3	通用IO
28	E17	UART0_RTS	0	3.3	UART0 请求发送
		uart4_txd	1	3.3	UART4 发送
		dcan1_rx	2	3.3	DCAN1 接收
		I2C1_SCL	3	3.3	I2C1 时钟
		spi1_d1	4	3.3	SPI1 数据1
		spi1_cs0	5	3.3	SPI1 片选
		prl_edc_sync1_out *	6	3.3	PRU_edc_sync1_out数据输出
		gpio1_9	7	3.3	通用IO
29	F18	MMC0_D2	0	3.3	MMC/SD/SDIO 数据2
		gpmc_a21	1	3.3	GPMC 地址21
		uart4_rtsn	2	3.3	UART4 请求发送
		timer6	3	3.3	定时器6/PWM输出
		uart1_dsrn	4	3.3	UART1 发送准备
		prl_pru0_pru_r30_9*	5	3.3	PRU 数据输出
		prl_pru0_pru_r31_9*	6	3.3	PRU 数据输入
		gpio2_27	7	3.3	通用IO
30	NC	GND		0	地
31	G15	MMC0_D1	0	3.3	MMC/SD/SDIO 数据1
		gpmc_a22	1	3.3	GPMC 地址22
		uart5_ctsn	2	3.3	UART5清除发送
		uart3_rxd	3	3.3	UART3 接收
		uart1_dtrn	4	3.3	UART1数据终端就绪
		prl_pru0_pru_r30_10*	5	3.3	PRU 数据发送
		prl_pru0_pru_r31_10*	6	3.3	PRU 数据接收
		gpio2_28	7	3.3	通用IO
32	E16	UART0_TXD	0	3.3	UART0 发送
		spi1_cs1	1	3.3	SPI1片选
		dcan0_rx	2	3.3	DCAN0 接收
		I2C2_SCL	3	3.3	I2C2 时钟
		eCAP1_in_PWM1_out	4	3.3	增强型捕捉1输入或辅助PWM1输出
		prl_pru1_pru_r30_15*	5	3.3	PRU 数据输出

		pr1_pru1_pru_r31_15*	6	3.3	PRU 数据输入
		gpio1_11	7	3.3	通用IO
33	G16	MMC0_D0	0	3.3	MMC/SD/SDIO数据0
		gpmc_a23	1	3.3	GPMC 地址23
		uart5_rtsn	2	3.3	UART5请求发送
		uart3_txd	3	3.3	UART3发送
		uart1_rin	4	3.3	UART1 振铃指示
		pr1_pru0_pru_r30_11*	5	3.3	PRU 数据输出
		pr1_pru0_pru_r31_11*	6	3.3	PRU 数据输入
		gpio2_29	7	3.3	通用IO
34	E15	UART0_RXD	0	3.3	UART0接收
		spi1_cs0	1	3.3	SPI1片选
		dcan0_tx	2	3.3	DCAN0发送
		I2C2_SDA	3	3.3	I2C2 数据
		eCAP2_in_PWM2_out	4	3.3	增强型捕捉2输入或辅助PWM2输出
		pr1_pru1_pru_r30_14*	5	3.3	PRU 数据输出
		pr1_pru1_pru_r31_14*	6	3.3	PRU 数据输入
		gpio1_10	7	3.3	通用IO
35	G18	MMC0_CMD	0	3.3	MMC/SD/SDIO 命令
		gpmc_a25	1	3.3	GPMC 地址25
		uart3_rtsn	2	3.3	UART3 请求发送
		uart2_txd	3	3.3	UART2 发送
		dcan1_rx	4	3.3	DCAN1 接收
		pr1_pru0_pru_r30_13*	5	3.3	PRU 数据输出
		pr1_pru0_pru_r31_13*	6	3.3	PRU 数据输入
		gpio2_31	7	3.3	通用IO
36	NC	GND		0	地
37	G17	MMC0_CLK	0	3.3	MMC/SD/SDIO 时钟
		gpmc_a24	1	3.3	GPMC 地址24
		uart3_ctsn	2	3.3	UART3 清除发送
		uart2_rxd	3	3.3	UART2 接收
		dcan1_tx	4	3.3	DCAN1 发送
		pr1_pru0_pru_r30_12*	5	3.3	PRU 数据输出
		pr1_pru0_pru_r31_12*	6	3.3	PRU 数据输入
		gpio2_30	7	3.3	通用IO
38	D18	UART1_CTS	0	3.3	UART1清除发送
		timer6	1	3.3	定时器6/PWM输出
		dcan0_tx	2	3.3	DCAN0发送
		I2C2_SDA	3	3.3	I2C2 数据
		spi1_cs0	4	3.3	SPI1片选
		pr1_uart0_cts_n*	5	3.3	PRU_UART0 清除发送
		pr1_edc_latch0_in*	6	3.3	PRU 数据输入

		gpio0_12	7	3.3	通用IO
39	C15	SPI0_CS1	0	3.3	SPI0片选
		uart3_rxd	1	3.3	UART3接收
		eCAP1_in_PWM1_out	2	3.3	增强型捕捉1输入或辅助PWM1输出
		mmc0_pow	3	3.3	MMC/SD0 电源开关控制
		xdma_event_intr2	4	3.3	外部DMA中断2
		mmc0_sdcd	5	3.3	MMC/SD/SDIO卡检测
		CD/EMU4	6	3.3	MISC仿真4
		gpio0_6	7	3.3	通用IO
40	D17	UART1_RTS	0	3.3	UART1请求接收
		timer5	1	3.3	定时器5/PWM输出
		dcan0_rx	2	3.3	DCAN0发送
		I2C2_SCL	3	3.3	I2C2 clock
		spi1_cs1	4	3.3	SPI1片选
		prl_uart0_rts_n*	5	3.3	PRU_UART0请求发送
		prl_edc_latch1_in*	6	3.3	PRU数据输入
		gpio0_13	7	3.3	通用IO
41		NC			
42	NC	GND		0	地
43	C18	LCD_ENBKL	0	3.3	LCD背光控制（PWM0输出）
		uart3_txd	1	3.3	UART3 发送
		spi1_cs1	2	3.3	SPI1 发送
		prl_ecap0_ecap_capin_a_pwm_0*	3	3.3	PRU增强型捕捉0输入或辅助PWM0输出
		spi1_sclk	4	3.3	SPI1 时钟
		mmc0_sdwp	5	3.3	MMC/SD/SDIO 写保护
		xdma_event_intr2	6	3.3	外部DMA中断2
		gpio0_7	7	3.3	通用IO
44	D15	UART1_TXD	0	3.3	UART1发送
		mmc2_sdwp	1	3.3	MMC/SD/SDIO 写保护
		dcan1_rx	2	3.3	DCAN1接收
		I2C1_SCL	3	3.3	I2C1 时钟
		prl_uart0_txd*	5	3.3	PRU_UART0 发送
		prl_pru0_pru_r31_16*	6	3.3	PRU 数据输入
		gpio0_15	7	3.3	通用IO
45	A15	EVENT_INTR0	0	3.3	外部DMA中断0
		timer4	2	3.3	定时器5/PWM输出
		clkout1	3	3.3	时钟输出1
		spi1_cs1	4	3.3	SPI1 片选
		prl_pru1_pru_r31_16*	5	3.3	PRU 数据输入
		EMU2	6	3.3	MISC仿真2
		gpio0_19	7	3.3	通用IO

46	D16	UART1_RXD	0	3.3	UART1接收
		mmc1_sdwp	1	3.3	MMC/SD/SDIO 写保护
		dcan1_tx	2	3.3	DCAN1发送
		I2C1_SDA	3	3.3	I2C1数据
		pr1_uart0_rxd*	5	3.3	PRU_UART0接收
		pr1_pru1_pru_r31_16*	6	3.3	PRU 数据输入
		gpio0_14	7	3.3	通用IO
47	D14	EVENT_INT1	0	3.3	外部DMA 中断 1
		tc1kin	2	3.3	时钟输入
		CLKOUT2	3	3.3	时钟输出2
		timer7	4	3.3	定时器7 / PWM 输出
		pr1_pru0_pru_r31_16*	5	3.3	PRU数据输入
		EMU3	6	3.3	Misc 仿真3
		gpio0_20	7	3.3	通用IO
48	NC	GND		0	地
49		NC			
50	C17	I2C0_SDA	0	3.3	I2C0 数据
		timer4	1	3.3	定时器4 / PWM 输出
		uart2_ctsn	2	3.3	UART2 清除发送
		eCAP2_in_PWM2_out	3	3.3	增强型捕捉2输入/PWM2输出
		gpio3_5	7	3.3	通用IO
51		NC			
52	C16	I2C0_SCL	0	3.3	I2C0 时钟
		timer7	1	3.3	定时器7 / PWM 输出
		uart2_rtsn	2	3.3	UART2 请求发送
		eCAP1_in_PWM1_out	3	3.3	增强型捕捉1输入/PWM1输出
		gpio3_6	7	3.3	通用IO
53	A13	MCASP0_ACLKX	0	3.3	McASP0发送时钟
		ehrpwm0A	1	3.3	eHRPWM0 A 输出
		SPI1_SCLK	3	3.3	SPI1 时钟
		mmc0_sdcd	4	3.3	MMC/SD/SDIO 检测
		pr1_pru0_pru_r30_0*	5	3.3	PRU数据输出
		pr1_pru0_pru_r31_0*	6	3.3	PRU数据输入
		gpio3_14	7	3.3	通用IO
54	NC	GND		0	地
55	B13	MCASP0_FSX	0	3.3	McASP0 发送帧同步
		ehrpwm0B	1	3.3	eHRPWM0 B 输出
		SPI1_D0	3	3.3	SPI1数据0
		mmc1_sdcd	4	3.3	SD 卡检测
		pr1_pru0_pru_r30_18*	5	3.3	PRU数据输出
		pr1_pru0_pru_r31_1*	6	3.3	PRU数据输入
		gpio3_15	7	3.3	通用IO

56	A16	SPI0_CS0	0	3.3	SPI0片选
		mmc2_sdwp	1	3.3	MMC/SD/SDIO 写保护
		I2C1_SCL	2	3.3	I2C1 时钟
		ehrpwm0_synci	3	3.3	eHRPWM0同步
		pr1_uart0_txd*	4	3.3	PRU_UART0发送
		pr1_edio_data_in1*	5	3.3	PRU 数据输入
		pr1_edio_data_out1*	6	3.3	PRU 数据输出
		gpio0_5	7	3.3	通用IO
57	D12	MCASP0_AXR0	0	3.3	McASP0 串行数据
		ehrpwm0_trip_zone_input	1	3.3	eHRPWM0 堆栈输入
		SPI1_D1	3	3.3	SPI1数据
		mmc2_sdcd	4	3.3	SD 检测
		pr1_pru0_pru_r30_2*	5	3.3	PRU数据输出
		pr1_pru0_pru_r31_2*	6	3.3	PRU数据输入
		gpio3_16	7	3.3	通用IO
58	B16	SPI0_D1	0	3.3	SPI 数据
		mmc1_sdwp	1	3.3	MMC/SD/SDIO 写保护
		I2C1_SDA	2	3.3	I2C1 数据
		ehrpwm0_tripzone_input	3	3.3	eHRPWM0 堆栈区输入
		pr1_uart0_rxd*	4	3.3	PRU UART 接收
		pr1_edio_data_in0*	5	3.3	PRU 数据输入
		pr1_edio_data_out0*	6	3.3	PRU 数据输出
		gpio0_4	7	3.3	通用IO
59	C12	MCASP0_AHCLKR	0	3.3	McASP0 接收主时钟
		ehrpwm0_synci	1	3.3	eHRPWM0 同步输入
		mcasp0_axr2	2	3.3	McASP0 串行数据
		SPI1_CS0	3	3.3	SPI片选1
		eCAP2_in_PWM2_out	4	3.3	增强捕捉2输入和PWM2输出
		pr1_pru0_pru_r30_3*	5	3.3	PRU数据输出
		pr1_pru0_pru_r31_3*	6	3.3	PRU数据输入
		gpio3_17	7	3.3	通用IO
60	NC	GND		0	地
61	B12	MCASP0_ACLKR	0	3.3	McASP0 接收位时钟
		eQEP0A_in	1	3.3	eQEP0A 输入
		mcasp0_axr2	2	3.3	McASP0 串行数据
		mcasp1_aclkx	3	3.3	McASP1 发送位时钟
		mmc0_sdwp	4	3.3	SD 写保护
		pr1_pru0_pru_r30_4*	5	3.3	PRU数据输出
		pr1_pru0_pru_r31_4*	6	3.3	PRU数据输入
		GPIO3_18	7	3.3	通用IO
62	B17	SPI0_D0	0	3.3	SPI0数据
		UART2_TXD	1	3.3	UART2 发送

		I2C2_SCL	2	3.3	I2C2 时钟
		ehrpwm0B	3	3.3	eHRPWM0 B 输出
		pr1_uart0_rts_n*	4	3.3	PRU_ART0 请求发送
		pr1_edio_latch_in*	5	3.3	PRU_锁存器输入
		EMU3	6	3.3	Misc 仿真3
		gpio0_3	7	3.3	通用IO
63	A14	MCASP0_AHCLKX	0	3.3	McASP0 接收主时钟
		eQEP0_strobe	1	3.3	qQEP1 选通
		mcasp0_axr3	2	3.3	McASP0 串行数据
		mcasp1_axr1	3	3.3	McASP1 串行数据
		EMU4	4	3.3	Misc 仿真4
		pr1_prU0_pru_r30_7*	5	3.3	PRU0 数据输出
		pr1_pru0_pru_r31_7*	6	3.3	PRU0 数据输入
		GPIO3_21	7	3.3	通用IO
64	A17	SPI0_CLK	0	3.3	SPI0 时钟
		UART2_RXD	1	3.3	UART2 接收
		I2C2_SDA	2	3.3	I2C2数据
		ehrpwm0A	3	3.3	eHRPWM0 A 输出
		pr1_uart0_cts_n*	4	3.3	PRU_UART 清除发送
		pr1_edio_sof*	5	3.3	PRU_起始帧
		EMU2	6	3.3	Misc 仿真2
		gpio0_2*	7	3.3	通用IO
65	C13	MCASP0_FSR	0	3.3	McASP0 接收帧同步
		eQEP0B_in	1	3.3	eQEP0B输入
		mcasp0_axr3	2	3.3	mcasp0 串行数据
		mcasp1_fsx	3	3.3	McASP1 发送帧同步
		EMU2	4	3.3	Misc仿真2
		pr1_pru0_pru_r30_5*	5	3.3	PRU0数据输出
		pr1_pru0_pru_r31_5*	6	3.3	PRU0数据输入
		GPIO3_19	7	3.3	通用IO
66		NC			
67	D13	MCASP0_AXR1	0	3.3	mcasp0 串行数据
		eQEP0_index	1	3.3	eQEP0 索引
		mcasp1_axr0	3	3.3	mcasp1 串行数据
		EMU3	4	3.3	Misc 仿真3
		pr1_pru0_pru_r30_6*	5	3.3	PRU0 数据输出
		pr1_pru0_pru_r31_6*	6	3.3	PRU0 数据输入
		GPIO3_20	7	3.3	通用IO
68	NC	GND		0	地
69	NC	GND		0	地
70	B7	Rscreen_Yup		3.3	模拟输入2
71	C8	AIN4		3.3	模拟输入4

72	NC	GND	0	地
73	NC	GND	0	地
74	A7	Rscreen_YDown	3.3	模拟输入3
75	B8	AIN5	3.3	模拟输入5
76	NC	GND	0	地
77	NC	GND	0	地
78	B6	Rscreen_XLeft	3.3	模拟输入0
79	A8	AIN6	3.3	模拟输入6
80	NC	GND	0	地
81	NC	GND	0	地
82	A7	Rscreen_XRight	3.3	模拟输入1
83	C9	AIN7	3.3	模拟输入7
84	NC	GND	0	地
85	NC	GND	0	地
86	B10	JTAG_TRSTn	3.3	JTAG 测试复位
87		NC		
88	A11	JTAG_TDO	3.3	JTAG 测试数据输出
89		NC		
90	B11	JTAG_TDI	3.3	JTAG 测试数据输入
91	A10	SYS_WARMRESET	3.3	系统复位
92	A12	JTAG_TCK	3.3	JTAG 测试时钟
93		NC		
94	C11	JTAG_TMS	3.3	JTAG 测试模式选择
95		NC		
96	C14	JTAG_EMU0	3.3	MISC 仿真0
		GPIO3_7	3.3	通用IO
97		NC		
98	B14	JTAG_EMU1	3.3	MISC 仿真1
		GPIO3_8	3.3	通用IO
99	NC	GND	0	地
100	NC	GND	0	地

J2

Pin	Ball	Pin Name	V	描述
1	P15	USB0_VBUS	5	USB0_VBUS
2	NC	GND	0	地
3	F16	USB0_DRVVBUS	3.3	USB0 VBUS控制输出
		gpio0_18	3.3	通用IO
4	N18	USB0_DM	3.3	USB0数据 负
5	P16	USB0_ID	3.3	USB0 OTG 标识

6	N17	USB0_DP		3.3	USB0数据 正
7	NC	GND		0	地
8	NC	GND		0	地
9	P17	USB0_ID		3.3	USB1 OTG 标识
10	R18	USB1_DM		3.3	USB1数据 负
11	F15	USB1_DRVVBUS		3.3	USB1 VBUS控制输出
		GPIO3_13		3.3	通用IO
12	R17	USB1_DP		3.3	USB1数据 正
13	T18	USB1_VBUS		5	USB1_VBUS
14	NC	GND		0	地
15	NC	GND		0	地
16	J16	GMII1_TXEN	0	3.3	GMII1 发送使能
		mii1_txen	1	3.3	MII1 发送使能
		RGMII1_TXEN	2	3.3	RGMII1 发送控制
		timer4	3	3.3	定时器4/PWM输出
		mcasp1_axr0	4	3.3	McASP1 穿行数据
		eQEP0_index	5	3.3	eQEP0索引
		mmc2_cmd	6	3.3	MMC/SD/SDIO 命令
		gpio3_3	7	3.3	通用IO
17	H18	RGMII1_REFCLK	0	3.3	MII1 参考时钟
		xdma_event_intr2	1	3.3	外部 DMA 中断2
		spi1_cs0	2	3.3	SPI1 片选
		uart5_txd	3	3.3	UART5 发送
		mcasp1_axr3	4	3.3	McASP1 串行数据
		mmc0_pow	5	3.3	MMC/SD/SDIO 电源开关控制
		mcasp1_ahclkx	6	3.3	McASP1 发送主时钟
		gpio0_29	7	3.3	通用IO
18	K18	GMII1_TXCLK	0	3.3	GMII1 发送时钟
		uart2_rxd	1	3.3	UART2 接收
		RGMII1_TXCLK	2	3.3	RGMII1发送时钟
		mmc0_dat7	3	3.3	MMC/SD/SDIO 数据7
		mmc1_dat0	4	3.3	MMC/SD/SDIO 数据0
		uart1_dcdn	5	3.3	UART1数据载波检测
		mcasp0_aclkx	6	3.3	McASP0 发送位时钟
		gpio3_9	7	3.3	通用IO
19	H17	RGMII1_CRS	0	3.3	MII1载波侦听
		rmii1_crs	1	3.3	RMII1 载波侦听
		spi1_d0	2	3.3	SPI1 数据
		I2C1_SDA	3	3.3	I2C1 数据
		mcasp1_aclkx	4	3.3	McASP1 发送位时钟
		uart5_ctsn	5	3.3	UART5 清除发送
		uart2_rxd	6	3.3	UART2 接收

		gpio3_1	7	3.3	通用IO
20	K17	GMII1_TXD0	0	3.3	MII1 发送数据0
		rmii1_txd0	1	3.3	RMII1 发送数据0
		RGMII1_TXD0	2	3.3	RGMII1发送数据0
		mcasp1_axr2	3	3.3	McASP1串行数据
		mcasp1_aclkr	4	3.3	McASP1 接收位时钟
		eQEP0B_in	5	3.3	eQEP0B 正交输入
		mmc1_clk	6	3.3	MMC/SD/SDIO时钟
		gpio0_28	7	3.3	通用IO
21	H16	RGMII1_COL	0	3.3	MII1 冲突检测
		rmii2_refelk	1	3.3	RMII2 参考时钟
		spi1_sclk	2	3.3	SPI1 时钟
		uart5_rxd	3	3.3	UART5 接收数据
		mcasp1_axr2	4	3.3	McASP1 串行数据
		mmc2_dat3	5	3.3	MMC/SD/SDIO 数据3
		mcasp0_axr2	6	3.3	McASP0 串行数据
		gpio3_0	7	3.3	通用IO
22	K16	GMII1_TXD1	0	3.3	MII1 发送数据1
		rmii1_txd1	1	3.3	RMII1发送数据1
		RGMII1_TXD1	2	3.3	RGMII1发送数据1
		mcasp1_fsr	3	3.3	McASP1 接收帧同步
		mcasp1_axr1	4	3.3	McASP1 串行数据
		eQEP0A_in	5	3.3	eQEP0A 正交输入
		mmc1_cmd	6	3.3	MMC/SD/SDIO 命令
		gpio0_21	7	3.3	通用IO
23	J15	RGMII1_RXERR	0	3.3	MII1 接收数据错误指示
		rmii1_rxerr	1	3.3	RMII1 接收数据错指示
		spi1_d1	2	3.3	SPI1数据
		I2C1_SCL	3	3.3	I2C1时钟
		mcasp1_fsx	4	3.3	McASP1发送帧同步
		uart5_rtsn	5	3.3	UART5 请求发送
		uart2_txd	6	3.3	UART2 发送
		gpio3_2	7	3.3	通用IO
24	K15	GMII1_TXD2	0	3.3	MII1 发送数据2
		dcan0_rx	1	3.3	DCAN0 接收
		RGMII1_TXD2	2	3.3	RGMII1 发送数据2
		uart4_txd	3	3.3	UART4发送
		mcasp1_axr0	4	3.3	McASP1 串行数据
		mmc2_dat2	5	3.3	MMC/SD/SDIO 数据
		mcasp0_ahclkx	6	3.3	McASP0 发送主时钟
		gpio0_17	7	3.3	通用IO
25	NC	GND		0	地

26	J18	GMII1_TXD3	0	3.3	MII1发送数据3
		dcan0_tx	1	3.3	DCAN0 发送
		RGMII1_TXD3	2	3.3	RGMII1发送数据3
		uart4_rxd	3	3.3	UART4 接收
		mcasp1_fsx	4	3.3	McASP1 发送帧同步
		mmc2_dat1	5	3.3	MMC/SD/SDIO数据
		mcasp0_fsr	6	3.3	McASP0 接收帧同步
		gpio0_16	7	3.3	通用IO
27	L18	GMII1_RXCLK	0	3.3	MII1接收时钟
		uart2_txd	1	3.3	UART2发送
		RGMII1_RXCLK	2	3.3	RGMII1接收时钟
		mmc0_dat6	3	3.3	MMC/SD/SDIO 数据
		mmc1_dat1	4	3.3	MMC/SD/SDIO 数据
		uart1_dsrn	5	3.3	UART1数据设备就绪
		mcasp0_fsx	6	3.3	McASP0 发送帧同步
		gpio3_10	7	3.3	通用IO
28	NC	GND	0		地
29	J17	RGMII1_RXDV	0	3.3	MII1接收数据有效
		lcd_memory_clk	1	3.3	LCD 主时钟
		rgmii1_rctl	2	3.3	RGMII1 接收控制
		uart5_txd	3	3.3	UART5 发送数据
		mcasp1_aclkx	4	3.3	McASP1 发送位时钟
		mmc2_dat0	5	3.3	MMC/SD/SDIO 数据
		mcasp0_aclkr	6	3.3	McASP0 接收位时钟
		gpio3_4	7	3.3	通用 IO
30	T17	GPMC_WAIT0	0	3.3	GPMC等待
		gmii2_crs	1	3.3	MII2 载波侦听
		gpmc_csn4	2	3.3	GPMC 片选
		mii2_crs_dv	3	3.3	RMI2 载波侦听
		mmc1_sdcd	4	3.3	MMC/SD/SDIO 检测
		pr1_mii1_col	5	3.3	MII1 冲突检测
		uart4_rxd	6	3.3	UART4接收
		gpio0_30	7	3.3	通用 IO
31	L17	GMII1_RXD3	0	3.3	MII1接收数据3
		uart3_rxd	1	3.3	UART3接收
		RGMII1_RXD3	2	3.3	RGMII1 接收数据3
		mmc0_dat5	3	3.3	MMC/SD/SDIO 数据
		mmc1_dat2	4	3.3	MMC/SD/SDIO 数据
		uart1_dtrn	5	3.3	UART1数据终端准备好
		mcasp0_axr0	6	3.3	McASP0 串行数据
		gpio2_18	7	3.3	通用 IO
32	U18	GPMC_BEn1	0	3.3	GPMC 位使能1

		gmii2_col	1	3.3	MII2 冲突检测
		gpmc_csn6	2	3.3	GPMC 片选
		mmc2_dat3	3	3.3	MMC/SD/SDIO 数据3
		gpmc_dir	4	3.3	GPMC 数据方向
		pr1_mii1_rxlink	5	3.3	MII1 接收连接
		mcasp0_aclkr	6	3.3	McASP0 接收位时钟
		gpio1_28	7	3.3	通用 IO
33	L16	GMII1_RXD2	0	3.3	MII1 接收数据2
		uart3_txd	1	3.3	UART3 发送
		RGMII1_RXD2	2	3.3	RGMII1 接收数据2
		mmc0_dat4	3	3.3	MMC/SD/SDIO 数据
		mmc1_dat3	4	3.3	MMC/SD/SDIO 数据
		uart1_rin	5	3.3	UART1振铃指示
		mcasp0_axr1	6	3.3	McASP0 串行数据
		gpio2_19	7	3.3	通用 IO
34	U17	GPMC_WPn	0	3.3	GPMC 写保护
		gmii2_rxerr	1	3.3	MII2 接收数据错
		gpmc_csn5	2	3.3	GPMC 片选
		rmii2_rxerr	3	3.3	RMII2 接收数据错
		mmc2_sdcd	4	3.3	MMC/SD/SDIO 检测
		pr1_mii1_txen	5	3.3	MII1 允许发送
		uart4_txd	6	3.3	UART4 发送数据
		gpio_31	7	3.3	通用 IO
35	L15	GMII1_RXD1	0	3.3	MII1 接收数据1
		rmii1_rxd1	1	3.3	RMII接收数据1
		RGMII1_RXD1	2	3.3	RGMII接收数据1
		mcasp1_axr3	3	3.3	McASP1 串行数据
		mcasp1_fsr	4	3.3	McASP1 接收帧同步
		eQEP0_strobe	5	3.3	eQEP0 选通
		mmc2_clk	6	3.3	MMC/SD/SDIO 时钟
		gpio2_20	7	3.3	通用 IO
36	NC	GND		0	地
37	M16	GMII1_RXD0	0	3.3	MII1接收数据0
		rmii1_rxd0	1	3.3	RMII1接收数据0
		RGMII1_RXD0	2	3.3	RGMII1 接收数据0
		mcasp1_ahclkx	3	3.3	McASP1 发送主时钟
		mcasp1_ahclkr	4	3.3	McASP1 接收主时钟
		mcasp1_aclkr	5	3.3	McASP1 接收位时钟
		mcasp0_axr3	6	3.3	McASP0 串行数据
		gpio2_21	7	3.3	通用 IO
38	T15	GPMC_A7	0	3.3	GPMC地址7
		gmii2_rxclk	1	3.3	MII2 接收时钟

		rgmii2_rclk	2	3.3	RGMII2 接收时钟
		mmc2_dat5	3	3.3	MMC/SD/SDIO 数据5
		gpmc_a23	4	3.3	GPMC 地址23
		pr1_mii1_rxd1*	5	3.3	PRU_MII1 接收数据1
		eQEP1_strobe	6	3.3	eQEP1 选通
		gpio1_23	7	3.3	通用IO
39	M18	GMII1_MDIO_CLK	0	3.3	MDIO 数据
		timer5	1	3.3	定时器5 / PWM输出
		uart5_txd	2	3.3	UART5 发送
		uart3_rtsn	3	3.3	UART3 请求发送
		mmc0_sdwp	4	3.3	MMC/SD/SDIO 写保护
		mmc1_clk	5	3.3	MMC/SD/SDIO 时钟
		mmc2_clk	6	3.3	MMC/SD/SDIO 时钟
		gpio0_1	7	3.3	通用IO
40	V14	GPMC_A1	0	3.3	GPMC 地址1
		gmii2_rxdv	1	3.3	MII2 接收数据有效
		rgmii2_rctl	2	3.3	RGMII2 接收控制
		mmc2_dat0	3	3.3	MMC/SD/SDIO 数据
		gpmc_a17	4	3.3	GPMC 地址17
		pr1_mii1_txd3*	5	3.3	MII1 发送数据3
		ehrpwm0_synco	6	3.3	eHRPWM0 同步输出
		gpio1_17	7	3.3	通用IO
41	M17	GMII1_MDIO_DATA	0	3.3	MDIO 数据
		timer6	1	3.3	定时器6 / PWM输出
		uart5_rxd	2	3.3	UART5 接收
		uart3_ctsn	3	3.3	UART3 清除发送
		mmc0_sdccl	4	3.3	MMC/SD/SDIO 检测
		mmc1_cmd	5	3.3	MMC/SD/SDIO 命令
		mmc2_cmd	6	3.3	MMC/SD/SDIO命令
		gpio0_0	7	3.3	通用IO
42	V17	GPMC_A11	0	3.3	GPMC 地址11
		gmii2_rxd0	1	3.3	MII2 接收数据0
		rgmii2_rd0	2	3.3	RGMII2 接收数据0
		rmii2_rxd0	3	3.3	RMII2 接收数据0
		gpmc_a27	4	3.3	GPMC 地址27
		pr1_mii_rxr*	5	3.3	PRU_MII 接收数据错
		mcasp0_axr1	6	3.3	McASP 串行数据
		gpio1_27	7	3.3	通用IO
43	NC	GND		0	地
44	T16	GPMC_A10	0	3.3	GPMC 地址10
		gmii2_rxd1	1	3.3	MII2 接收数据1
		rgmii2_rd1	2	3.3	RGMII2 接收数据1

		rmii2_rxd1	3	3.3	RMII2 接收数据1
		gpmc_a26	4	3.3	GPMC 地址26
		prl_mii_rxdv*	5	3.3	PRU_MII 接收数据有效
		mcasp0_axr0	6	3.3	McASP 串行数据输出
		gpio1_26	7	3.3	通用IO
45	U14	GPMC_A2	0	3.3	GPMC 地址2
		gmii2_txd3	1	3.3	MII2 发送数据3
		rgmii2_td3	2	3.3	RGMII2 发送数据3
		mmc2_dat1	3	3.3	MMC/SD/SDIO 数据
		gpmc_a18	4	3.3	GPMC 地址18
		prl_mii1_txd2*	5	3.3	PRU_MII 发送数据2
		ehrpwm1A	6	3.3	eHRPWM1 A输出
		gpio1_18	7	3.3	通用IO
46	U16	GPMC_A9	0	3.3	GPMC 地址9
		gmii2_rxd2	1	3.3	MII2 接收数据2
		rgmii2_rd2	2	3.3	RGMII2 接收数据2
		mmc2_dat7	3	3.3	MMC/SD/SDIO 数据
		gpmc_a25	4	3.3	GPMC 地址25
		prl_mii_mr1_clk*	5	3.3	PRU_MII 接收时钟
		mcasp0_fsx	6	3.3	McASP0 发送帧同步
		gpio1_25	7	3.3	通用IO
47	T14	GPMC_A3	0	3.3	GPMC 地址3
		gmii2_txd2	1	3.3	MII2 发送数据2
		rgmii2_td2	2	3.3	RGMII2发送数据2
		mmc2_dat2	3	3.3	MMC/SD/SDIO 数据
		gpmc_a19	4	3.3	GPMC 地址19
		prl_mii1_txd1*	5	3.3	PRU_MII 发送数据1
		ehrpwm1B	6	3.3	eHRPWM1 B 输出
		gpio1_19	7	3.3	通用IO
48	V16	GPMC_A8	0	3.3	GPMC 地址8
		gmii2_rxd3	1	3.3	MII2 接收数据3
		rgmii2_rd3	2	3.3	RGMII2接收数据3
		mmc2_dat6	3	3.3	MMC/SD/SDIO 数据
		gpmc_a24	4	3.3	GPMC 地址24
		prl_mii1_rxd0*	5	3.3	PRU_MII 接收数据
		mcasp0_aclkx	6	3.3	McASP0 发送位时钟
		gpio1_24	7	3.3	通用IO
49	R14	GPMC_A4	0	3.3	GPMC 地址4
		gmii2_txd1	1	3.3	MII2 发送数据1
		rgmii2_td1	2	3.3	RGMII2发送数据1
		rmii2_txd1	3	3.3	RMII2发送数据1
		gpmc_a20	4	3.3	GPMC 地址20

		prl_mii1_txd0*	5	3.3	PRU_MII 发送数据
		eQEP1A_in	6	3.3	eQEP1A 正交输入
		gpio1_20	7	3.3	通用IO
50	NC	GND		0	地
51	V15	GPMC_A5	0	3.3	GPMC 地址5
		gmii2_txd0	1	3.3	RMII2 发送数据0
		rgmii2_txd0	2	3.3	RGMII2 发送数据0
		rmii2_txd0	3	3.3	RMII2 发送数据0
		gpmc_a21	4	3.3	GPMC 地址21
		prl_mii1_rxd3*	5	3.3	PRU_MII 接收数据
		eQEP1B_in	6	3.3	eQEP1B 正交输入
		gpio1_21	7	3.3	通用IO
52	U13	GPMC_D15	0	3.3	GPMC 数据15
		LCD_DAT16	1	3.3	LCD 数据16
		mmc1_dat7	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat3	3	3.3	MMC/SD/SDIO 数据
		eQEP2_strobe	4	3.3	eQEP2 选通
		prl_ecap0_ecap_capin_apwm_o*	5	3.3	增强捕捉输入 / PWM输出
		prl_pru0_pru_r31_15*	6	3.3	PRU数据输入
		gpio1_15	7	3.3	通用IO
53	U15	GPMC_A6	0	3.3	GPMC地址6
		gmii2_txclk	1	3.3	MII2发送时钟
		rgmii2_tclk	2	3.3	RGMII2发送时钟
		mmc2_dat4	3	3.3	MMC/SD/SDIO 数据
		gpmc_a22	4	3.3	GPMC地址22
		prl_mii1_rxd2*	5	3.3	PRU_MII 接收数据
		eQEP1_index	6	3.3	eQEP1 指示
		gpio1_22	7	3.3	通用IO
54	V13	GPMC_D14	0	3.3	GPMC 数据14
		LCD_DAT17	1	3.3	LCD 数据17
		mmc1_dat6	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat2	3	3.3	MMC/SD/SDIO 数据
		eQEP2_index	4	3.3	eQEP2 索引
		prl_mii0_txd0*	5	3.3	PRU_MII发送数据
		prl_pru0_pru_r31_14*	6	3.3	PRU数据输入
		gpio1_14	7	3.3	通用IO
55	R13	GPMC_A0	0	3.3	GPMC 地址0
		gmii2_txen	1	3.3	MII2 发送使能
		rgmii2_tctl	2	3.3	RGMII2 发送控制
		rmii2_txen	3	3.3	RMII2发送使能
		gpmc_a16	4	3.3	GPMC 地址16

		prl_mii_mt1_clk*	5	3.3	PRU_MII 发送时钟
		ehrpwm1_tripzone_input	6	3.3	eHRPWM1 堆栈输入
		gpio1_16	7	3.3	通用IO
56	R12	GPMC_D13	0	3.3	GPMC 数据13
		LCD_DAT18	1	3.3	LCD 数据18
		mmc1_dat5	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat1	3	3.3	MMC/SD/SDIO 数据
		eQEP2B_in	4	3.3	eQEP2B 正交输入
		prl_mii0_txd1*	5	3.3	PRU_MII 发送数据
		prl_pru0_pru_r30_15*	6	3.3	PRU数据输出
		gpio1_13	7	3.3	通用IO
57	NC	GND		0	地
58	T12	GPMC_D12	0	3.3	GPMC 数据12
		LCD_DAT19	1	3.3	LCD 数据19
		mmc1_dat4	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat0	3	3.3	MMC/SD/SDIO 数据
		eQEP2A_in	4	3.3	eQEP2A 正交输入
		prl_mii0_txd2*	5	3.3	PRU_MII 发送数据
		prl_pru0_pru_r30_14*	6	3.3	PRU数据发送
		gpio1_12	7	3.3	通用IO
59	T9	GPMC_D7	0	3.3	GPMC 数据7
		mmc1_dat7	1	3.3	MMC/SD/SDIO 数据
		gpio1_7	7	3.3	通用IO
60	U12	GPMC_D11	0	3.3	GPMC 数据11
		LCD_DAT20	1	3.3	LCD 数据20
		mmc1_dat3	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat7	3	3.3	MMC/SD/SDIO 数据
		ehrpwm0_synco	4	3.3	eHRPWM0 同步输出
		prl_mii0_txd3*	5	3.3	PRU_MII 发送
		gpio0_27	7	3.3	通用IO
61	R9	GPMC_D6	0	3.3	GPMC 数据6
		mmc1_dat6	1	3.3	MMC/SD/SDIO 数据
		gpio1_6	7	3.3	通用IO
62	T11	GPMC_D10	0	3.3	GPMC 数据10
		LCD_DAT21	1	3.3	LCD 数据21
		mmc1_dat2	2	3.3	MMC/SD/SDIO数据
		mmc2_dat6	3	3.3	MMC/SD/SDIO数据
		ehrpwm2_tripzone_input	4	3.3	eHRPWM2 堆栈区输入
		prl_mii0_txen*	5	3.3	PRU_MII发送使能
		gpio0_26	7	3.3	通用IO
63	V8	GPMC_D5	0	3.3	GPMC 数据5
		mmc1_dat5	1	3.3	MMC/SD/SDIO 数据

		gpio1_5	7	3.3	通用IO
64	T10	GPMC_D9	0	3.3	GPMC 数据9
		LCD_DAT22	1	3.3	LCD 数据22
		mmc1_dat1	2	3.3	MMC/SD/SDIO数据
		mmc2_dat5	3	3.3	MMC/SD/SDIO数据
		ehrpwm2B	4	3.3	eHRPWM2 B 输出
		pr1_mii0_col*	5	3.3	PRU_MII 冲突检测
		gpio0_23	7	3.3	通用IO
65	U8	GPMC_D4	0	3.3	GPMC 数据4
		mmc1_dat4	1	3.3	MMC/SD/SDIO 数据
		gpio1_4	7	3.3	通用IO
66	U10	GPMC_D8	0	3.3	GPMC 数据8
		LCD_DAT23	1	3.3	LCD 数据23
		mmc1_dat0	2	3.3	MMC/SD/SDIO 数据
		mmc2_dat4	3	3.3	MMC/SD/SDIO 数据
		ehrpwm2A	4	3.3	eHRPWM2 A 输出
		pr1_mii_mt0_clk*	5	3.3	PRU_MII 发送时钟
		gpio0_22	7	3.3	通用IO
67	T8	GPMC_D3	0	3.3	GPMC数据3
		mmc1_dat3	1	3.3	MMC/SD/SDIO 数据
		gpio1_3	7	3.3	通用IO
68	T5	LCD_D15	0	3.3	LCD 数据15
		gpmc_a19	1	3.3	GPMC地址19
		eQEP1_strobe	2	3.3	eQEP1 选通
		mcasp0_ahclkx	3	3.3	McASP0 发送主时钟
		mcasp0_axr3	4	3.3	McASP0 穿行数据
		pr1_mii0_rxdv*	5	3.3	PRU_MII 接收数据有效
		uart5_rtsn	6	3.3	UART5 请求发送
		gpio0_11	7	3.3	通用IO
69	R8	GPMC_D2	0	3.3	GPMC 数据2
		mmc1_dat2	1	3.3	MMC/SD/SDIO 数据
		gpio1_2	7	3.3	通用IO
70	V4	LCD_D14	0	3.3	LCD 数据14
		gpmc_a18	1	3.3	GPMC 地址18
		eQEP1_index	2	3.3	eQEP1 索引
		mcasp0_axr1	3	3.3	McASP1 串行数据
		uart5_rxd	4	3.3	UART5 接收
		pr1_mii_mr0_clk*	5	3.3	PRU_MII 接收时钟
		uart5_ctsn	6	3.3	UART5 清除发送
		gpio0_10	7	3.3	通用IO
71	V7	GPMC_D1	0	3.3	GPMC 数据1
		mmc1_dat1	1	3.3	MMC/SD/SDIO 数据

		gpio1_1	7	3.3	通用IO
72	V3	LCD_D13	0	3.3	LCD 数据13
		gpmc_a17	1	3.3	GPMC 地址17
		eQEP1B_in	2	3.3	eQEP1B 正交输入
		mcasp0_fsr	3	3.3	McASP0 接收帧同步
		mcasp0_axr3	4	3.3	McASP0 串行数据
		pr1_mii0_rxer*	5	3.3	PRU_MII 接收错误指示
		uart4_rtsn	6	3.3	UART4请求发送
		gpio0_9	7	3.3	通用IO
73	U7	GPMC_D0	0	3.3	GPMC 数据0
		mmc1_dat0	1	3.3	MMC/SD/SDIO 数据
		gpio1_0	7	3.3	通用IO
74	V2	LCD_D12	0	3.3	LCD数据12
		gpmc_a16	1	3.3	GPMC地址16
		eQEP1A_in	2	3.3	eQEP1A 正交输入
		mcasp0_aclkr	3	3.3	McASP 接收位时钟
		mcasp0_axr2	4	3.3	McASP0 串行数据
		pr1_mii0_rxlink*	5	3.3	PRU_MII 接收连接
		uart4_ctsn	6	3.3	UART4清除发送
		gpio0_8	7	3.3	通用IO
75	V12	GPMC_CLK	0	3.3	GPMC时钟
		Lcd_memory_clk	1	3.3	LCD 主时钟
		gpmc_wait1	2	3.3	GPMC等待
		MMC2_CLK	3	3.3	MMC/SD/SDIO 时钟
		pr1_mii1_crs*	4	3.3	PRU_MII 载波侦听
		pr1_mdio_mdclk*	5	3.3	PRU_MDIO 时钟
		mcasp0_fsr	6	3.3	McASP0 接收帧同步
		gpio2_1	7	3.3	通用IO
76	U4	LCD_D11	0	3.3	LCD 数据11
		gpmc_a15	1	3.3	GPMC 地址15
		ehrpwm1B	2	3.3	eHRPWM1 B 输出
		mcasp0_ahclkr	3	3.3	McASP0 接收主时钟
		mcasp0_axr2	4	3.3	McASP0 串行数据
		pr1_mii0_rxd0*	5	3.3	PRU_MII 接收
		uart3_rtsn	6	3.3	UART3 请求发送
		gpio2_17	7	3.3	通用IO
77	T13	GPMC_CS3	0	3.3	GPMC 片选
		gpmc_a3	1	3.3	GPMC 地址3
		rmii2_crs_dv	2	3.3	RMII2载波侦听
		MMC2_CMD	3	3.3	MMC/SD/SDIO 命令
		pr1_mii0_crs*	4	3.3	PRU_MII 载波侦听
		pr1_mdio_data*	5	3.3	MDIO 数据

		EMU4	6	3.3	仿真4
		gpio2_0	7	3.3	通用IO
78	U3	LCD_D10	0	3.3	LCD 数据10
		gpmc_a14	1	3.3	GPMC 地址14
		ehrpwm1A	2	3.3	eHRPWM1 A 输出
		mcasp0_axr0	3	3.3	McASP0 串行数据
		pr1_mii0_rxd1*	5	3.3	PRU_MII 接收数据
		uart3_ctsn	6	3.3	UART3 清除发送
		gpio2_16	7	3.3	通用IO
79	V9	GPMC_CS2n	0	3.3	GPMC 片选
		gpmc_be1n	1	3.3	GPMC 位使能
		mmc1_cmd	2	3.3	MMC/SD/SDIO 命令
		pr1_edio_data_in7*	3	3.3	PRU_数据输入
		pr1_edio_data_out7*	4	3.3	PRU_数据输出
		pr1_prul_pru_r30_13*	5	3.3	PRU_数据输出
		pr1_prul_pru_r31_13*	6	3.3	PRU_数据输入
		gpio1_31	7	3.3	通用IO
80	U2	LCD_D9	0	3.3	LCD 数据9
		gpmc_a13	1	3.3	GPMC 地址13
		ehrpwm0_synco	2	3.3	eHRPWM0 同步输出
		mcasp0_fsx	3	3.3	McASP0 发送帧同步
		uart5_rxd	4	3.3	UART5接收
		pr1_mii0_rxd2*	5	3.3	PRU_MII 接收
		uart2_rtsn	6	3.3	UART2允许发送
		gpio2_15	7	3.3	通用IO
81	U9	GPMC_CSn1	0	3.3	GPMC 片选
		gpmc_clk	1	3.3	GPMC 时钟
		mmc1_clk	2	3.3	MMC/SD/SDIO 时钟
		pr1_edio_data_in6*	3	3.3	PRU_数据输入
		pr1_edio_data_out6*	4	3.3	PRU_数据输出
		pr1_prul_pru_r30_12*	5	3.3	PRU_数据输出
		pr1_prul_pru_r31_12*	6	3.3	PRU_数据输入
		gpio1_30	7	3.3	通用 IO
82	U1	LCD_D8	0	3.3	LCD 数据8
		gpmc_a12	1	3.3	GPMC 地址12
		ehrpwm1_tripzone_input	2	3.3	eHRPWM1 堆栈输入
		mcasp0_aclkx	3	3.3	McASP0 发送位时钟
		uart5_txd	4	3.3	UART5发送
		pr1_mii0_rxd3*	5	3.3	PRU_MII 接收
		uart2_ctsn	6	3.3	UART2 清除发送
		gpio2_14	7	3.3	通用 IO
83	T6	GPMC_BEn0_CLE	0	3.3	GPMC 位使能 / 命令锁存使能

		timer5	2	3.3	定时器5 / PWM输出
		gpio2_5	7	3.3	通用 IO
84	T4	LCD_D7	0	3.3	LCD 数据7
		gpmc_a7	1	3.3	GPMC 地址7
		pr1_edio_data_in7*	2	3.3	PRU_数据输入
		eQEP2_strobe	3	3.3	qQEP2 选通
		pr1_edio_data_out7*	4	3.3	PRU_数据输出
		pr1_prul_pru_r30_7*	5	3.3	PRU_数据输出
		pr1_prul_pru_r31_7*	6	3.3	PRU_数据输入
		gpio2_13	7	3.3	通用 IO
85	R7	GPMC_ADVn_ALE	0	3.3	GPMC 地址有效 / 地址锁存有效
		timer4	2	3.3	定时器4 / PWM输出
		gpio2_2	7	3.3	通用 IO
86	T3	LCD_DAT6	0	3.3	LCD 数据6
		gpmc_a6	1	3.3	GPMC 地址6
		pr1_edio_data_in6*	2	3.3	PRU_数据输入
		eQEP2_index	3	3.3	eQEP2 索引
		pr1_edio_data_out6*	4	3.3	PRU_数据输出
		pr1_prul_pru_r30_6*	5	3.3	PRU_数据输出
		pr1_prul_pru_r31_6*	6	3.3	PRU_数据输入
		gpio2_12	7	3.3	通用 IO
87	T7	GPMC_OEn_REn	0	3.3	GPMC 输出/读 使能
		timer7	2	3.3	定时器4 / PWM输出
		gpio2_3	7	3.3	通用 IO
88	T2	LCD_D5	0	3.3	LCD 数据5
		gpmc_a5	1	3.3	GPMC 地址5
		pr1_mii0_tx0*	2	3.3	PRU_MII 发送
		eQEP2B_in	3	3.3	eQEP2B 正交输入
		pr1_prul_pru_r20_5*	5	3.3	PRU数据输出
		pr1_prul_pru_r31_5*	6	3.3	PRU数据输入
		gpio2_11	7	3.3	通用 IO
89	U6	GPMC_WEn	0	3.3	GPMC 写使能
		timer6	2	3.3	定时器6 / PWM输出
		gpio2_4	7	3.3	通用 IO
90	T1	LCD_D4	0	3.3	LCD 数据4
		gpmc_a4	1	3.3	GPMC 地址4
		pr1_mii0_tx1*	2	3.3	PRU_MII 接收
		eQEP2A_in	3	3.3	eQEP2A 正交输入
		pr1_prul_pru_r30_4*	5	3.3	PRU数据输出
		pr1_prul_pru_r31_4*	6	3.3	PRU数据输入
		gpio2_10	7	3.3	通用 IO
91	V5	LCD_PCLK	0	3.3	LCD 像素时钟

		gpmc_a10	1	3.3	GPMC 地址10
		prl_mii0_crs*	2	3.3	PRU_MII 载波侦听
		prl_edio_data_in4*	3	3.3	PRU数据输入
		prl_edio_data_out4*	4	3.3	PRU数据输出
		prl_prul_pru_r30_10*	5	3.3	PRU数据输出
		prl_prul_pru_r31_10*	6	3.3	PRU数据输入
		gpio2_24	7	3.3	通用 IO
92	R4	LCD_D3	0	3.3	LCD 数据3
		gpmc_a3	1	3.3	GPMC 地址3
		prl_mii0_txd2*	2	3.3	PRU_MII 发送
		ehrpwm0_synco	3	3.3	eHRPWM0 同步输出
		prl_prul_pru_r30_3*	5	3.3	PRU数据输出
		prl_prul_pru_r31_3*	6	3.3	PRU数据输入
		gpio2_9	7	3.3	通用 IO
93	R5	LCD_HSYNC	0	3.3	LCD 水平同步
		gpmc_a9	1	3.3	GPMC 地址9
		gpmc_a2	2	3.3	GPMC 地址2
		prl_edio_data_in3*	3	3.3	PRU数据输入
		prl_edio_data_out3*	4	3.3	PRU数据输出
		prl_prul_pru_r30_9*	5	3.3	PRU数据输出
		prl_prul_pru_r31_9*	6	3.3	PRU数据输入
		gpio2_23	7	3.3	通用 IO
94	R3	LCD_D2	0	3.3	LCD 数据2
		gpmc_a2	1	3.3	GPMC 地址2
		prl_mii0_txd3*	2	3.3	PRU_MII 发送
		ehrpwm2_tripzone_input	3	3.3	eHRPWM2 堆栈区输入
		prl_prul_pru_r30_2*	5	3.3	PRU数据输出
		prl_prul_pru_r31_2*	6	3.3	PRU数据输入
		gpio2_8	7	3.3	通用 IO
95	U5	LCD_VSYNC	0	3.3	LCD 垂直同步
		gpmc_a8	1	3.3	GPMC 地址8
		gpmc_a1	2	3.3	GPMC 地址1
		prl_edio_data_in2*	3	3.3	PRU数据输入
		prl_edio_data_out2*	4	3.3	PRU数据输出
		prl_prul_pru_r30_8*	5	3.3	PRU数据输出
		prl_prul_pru_r31_8*	6	3.3	PRU数据输入
		gpio2_22	7	3.3	通用 IO
96	R2	LCD_D1	0	3.3	LCD数据1
		gpmc_a1	1	3.3	GPMC 地址1
		prl_mii0_txen*	2	3.3	PRU_MII 发送使能
		ehrpwm2B	3	3.3	eHRPWM2 B 输出
		prl_prul_pru_r30_1*	5	3.3	PRU数据输出

		prl_pru1_pru_r31_1*	6	3.3	PRU数据输入
		gpio2_7	7	3.3	通用 IO
97	R6	LCD_AC_BIAS_EN	0	3.3	LCD AC 偏置使能
		gpmc_a11	1	3.3	GPMC 地址11
		prl_mii1_crs*	2	3.3	PRU_MII 载波侦听
		prl_edio_data_in5*	3	3.3	PRU数据输入
		prl_edio_data_out5*	4	3.3	PRU数据输出
		prl_pru1_pru_r30_11*	5	3.3	PRU数据输出
		prl_pru1_pru_r31_11*	6	3.3	PRU数据输入
		gpio2_25	7	3.3	通用 IO
98	R1	LCD_D0	0	3.3	LCD 数据0
		gpmc_a0	1	3.3	GPMC 地址0
		prl_mii_mt0_clk*	2	3.3	PRU_MII 发送时钟
		ehrpwm2A	3	3.3	eHRPWM2 A 输出
		prl_pru1_pru_r30_0*	5	3.3	PRU数据输出
		prl_pru1_pru_r31_0*	6	3.3	PRU数据输入
		gpio2_6	7	3.3	通用 IO
99	NC	GND		0	地
100	NC	GND		0	地