

## EM9283 工控主板数据手册

感谢您购买英创信息技术有限公司的产品：**EM9283 工控主板**。

EM9283 是一款面向手持式设备的高性价比嵌入式主板，以 FreeScale 的 iMX283 为其硬件核心，通过预装正版 Window CE6.0 实时多任务操作系统，为用户构造了可直接使用的通用嵌入式核心平台。用户可使用 Microsoft 提供的著名软件开发工具 Visual Studio 2005（简称 VS2005），在 EM9283 上直接开发应用程序。英创公司针对 EM9283 提供了完整的接口底层驱动以及丰富的应用程序范例，用户可在此基础上方便、快速地开发出各种手持式产品。

EM9283 主要特点：

- **完善的电源管理机制：**EM9283 具有作为手持式设备所必需的完整电源管理机制，在常规 Li 电池（1800mAH）供电情况下，具有 2 个工作日的持续工作时间。系统支持挂起/唤醒，一键开关机功能，系统完全待机时的电流小于 20uA。
- **强大的硬件配置：**EM9283 的 CPU 为 ARM926EJ-S，其运行主频高达 454MHz。配以 128MB 的系统内存，使 EM9283 能满足各种大型应用程序的运行需求。在显示方面，EM9283 即可支持彩色 LCD 显示，也可支持单色 LCD 显示，为客户整机设计提供了更为灵活的选择。
- **完备的标准接口资源：**针对 EM9283 主要在工业手持式设备上的应用，EM9283 配置了多达 5 路标准串口，以满足不同应用需求。此外 EM9283 还配置若干标准接口包括：（1）1 路 USB2.0 高速主控接口，1 路 USB2.0 OTG 接口；（2）标准 SD 卡接口，支持 SDHC 大容量存储；（3）1 路 SPI 接口；（4）1 路 I2C 接口总线；（5）4 路 PWM 输出；（6）2 路 AD 输入；（7）32 位 GPIO。
- **紧凑的外形尺寸：**EM9283 的外形尺寸为 40mm×40mm 正方形，采用两个 60 芯 0.8mm 间距的高品质连接器与应用底板相连，板间高度仅 4mm，具有良好的抗振动冲击性能，可完全满足绝大多数手持式设备对外形尺寸的要求。
- **产品的兼容特性：**EM9283 预装了微软的 Windows CE6.0 操作系统，Windows

CE6.0 是当前市场上最流行的嵌入式实时多任务操作系统之一。CE6.0 的一个主要优点是它的高性能的图形窗口显示功能，同时微软针对 CE 的应用开发推出一体化的开发工具 VS2005，该工具可支持客户采用 C/C++、C#、VB 等语言来开发系统应用程序。英创公司为 EM9283 的所有接口编写符合 CE 标准的驱动程序，这样用户在其他手持式设备上的应用程序可非常方便的移植到 EM9283 上来。

本手册详细介绍了 EM9283 的硬件配置、管脚定义及相关的技术指标，供用户使用时备查。此外，英创公司针对 EM9283 评估底板的使用编写有《EM9283 开发评估底板手册》。这两个手册都包含在英创为用户提供的产品开发光盘里面，用户也可以登录英创公司的网站下载相关资料的最新版本。

用户还可以访问英创公司网站或直接与英创公司联系以获得 EM9283 的其他相关资料。

英创信息技术有限公司联系方式如下：

地址：成都市高新区高朋大道 5 号博士创业园 B 座 404# 邮编：610041

联系电话：028-86180660 传真：028-85141028

网址：<http://www.emtronix.com> 电子邮件：[support@emtronix.com](mailto:support@emtronix.com)

注意：本手册的相关技术内容将会不断的完善，请客户适时从公司网站下载最新版本的数据手册，恕不另行通知。

## 目 录

1、主要技术指标.....	4
2、外形尺寸.....	6
3、模块信号管脚功能描述.....	7
3.1 EM9283 的 CN1 信号定义.....	8
3.2 EM9283 的 CN2 信号定义.....	11
3.3 EM9283 连接单色 LCD 模块.....	14
4、电源管理及相关说明.....	15
4.1 电源输入.....	15
4.2 电源输出.....	15
4.3 锂电池充电管理.....	16
4.4 电流限制.....	16
4.5 开关机与外部复位.....	17
4.6 电源直流电气特性.....	17
5、数字 IO 信号的基本电气特性.....	19
6、EM9283 的相关功能的说明.....	20
WDT 看门狗定时器.....	20
USB 接口.....	21
UART 异步串口.....	21
I <sup>2</sup> C 接口.....	22
SPI 同步串口.....	24
PWM 脉冲输出.....	24
AD 模拟通道.....	26
矩阵键盘接口.....	28
GPIO 通用数字 IO.....	29
7、版本历史.....	31

# 1、主要技术指标

## 核心单元

- 454MHz 主频的 ARM9 CPU
- 核心芯片为 Freescale 的 iMX283
- 128MB DDR2 系统内存，用户可用空间大于 100MB
- 128MB FLASH 存储器（512MB 可选）
- USB 接口支持 U 盘即插即用
- SD 接口，支持 SDHC 大容量存储
- 实时时钟 RTC
- 硬件看门狗（WDT），防止系统死锁

## 通讯接口配置

- 5 路标准 UART 高速串口，其波特率可达 3Mbps  
COM2: 支持 RTS/CTS 硬件流控，推荐作为无线通讯模块接口；  
COM3 – COM4: 3 线制，TTL 电平；  
COM5 – COM6: 3 线制，TTL 电平接口；与 GPIO 复用管脚；
- 1 路 USB 高速主控接口（HOST）
- 1 路 USB OTG 接口，支持微软的 ActiveSync 通讯协议
- 1 路 I2C 接口，主控模式，最高波特率 400kbps，与 GPIO 复用管脚
- 1 路 SPI 接口，主控半双工模式，最高波特率 10Mbps，与 GPIO 复用管脚
- 1 路调试串口（115200bps，8-N-1）

## 人机交互单元

- 多功能 LCD 接口，支持彩色或单色 LCD 显示
- 缺省配置为单色 LCD，分辨率 160×160，4-bit 灰度级。
- 彩色 LCD 接口规范：RGB 各 6-bit + 同步时钟信号
- 典型彩色显示分辨率：800×480、640×480、480×272
- 直接支持 4 线制电阻触摸屏

- 矩阵键盘接口，缺省配置 3×3 矩阵键盘
- 支持矩阵键盘灵活扩展，最大可扩展至 5×5 矩阵键盘

### 数字即模拟监控单元

- 32 位通用 GPIO0 – GPIO31，各位方向独立可控。
- 部分 GPIO 与系统的其他通讯功能复用管脚。
- 2 路低速 AD 采集通道，AD 分辨率为 12-bit。
- 支持对 CPU 温度、主板环境温度、供电电压的实时监测。

### 电源及模块机械参数

- 主供电电源：Li-ION 电池，典型规格：1800mAH
- 电流消耗：CPU 10%负载：100mA；系统挂起：8mA；系统关机：15uA
- 充电电源：+5V，支持 USB 电源充电方案
- +5V 充电：最大充电电流 500mA；典型充电时间：3.5 小时
- USB 充电：最大充电电流 300mA；典型充电时间：6 小时
- 工作温度：-10℃至 60℃
- 模块外形尺寸：40mm×40mm；板间高度：4mm
- 连接器规格：60 芯 0.8mm 间距 AXN 双排插座
- 连接器特色：波纹触点结构，抗振动冲击
- 连接器位置：2 个 AXN 插座分布于模块的两侧

### 基本软件环境

- 预装 Windows CE6.0 实时多任务操作系统
- 提供相应 SDK 开发包，包括各种接口驱动程序 API
- 支持 VS2005 应用程序集成开发环境
- 支持 ActiveSync 方式的应用程序源码调试及文件远程管理
- 支持用户自行修改开机启动画面和关机画面
- 提供典型应用参考程序源码

## 2、外形尺寸

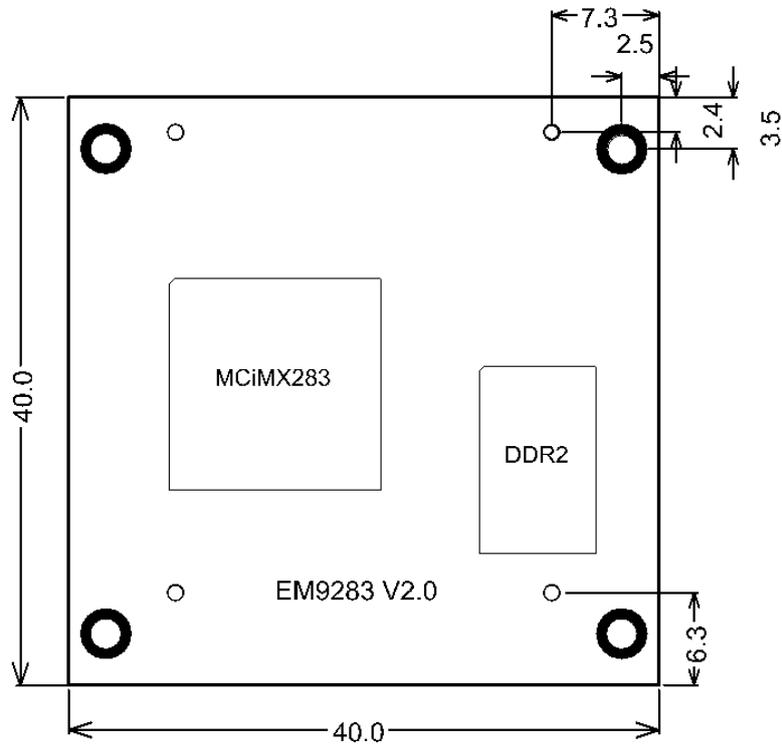


图 2.1 EM9283 元件面外形尺寸示意图 (单位: mm)

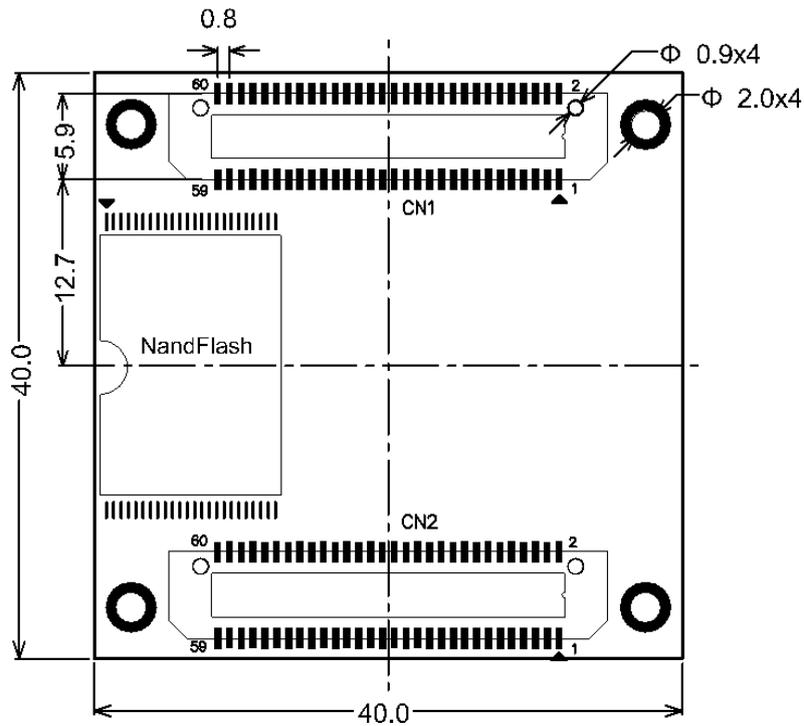


图 2.2 EM9283 焊接面外形尺寸示意图 (单位: mm)

### 3、模块信号管脚功能描述

EM9283 的使用是以模块形式，通过板上的相关插针，插在应用主板上，同时实现 EM9283 板卡的固定以及与应用主板的信号连接两个功能。EM9283 共有 2 组 60 芯、0.8mm 间距双排信号连接器，采用松下具有波纹型触点结构的 AXN 连接器插座，保证与应用主板的连接具有高度的耐振动、耐冲击性，可完全满足手持式设备运行环境的需求。插座的逻辑编号为 CN1 和 CN2，分别位于 EM9283 板卡模块的两端，如图 3.1 所示。

EM9283 的 CN1 主要包括电源、异步串口、USB、GPIO 等信号；而 CN2 主要包括 LCD 接口、矩阵键盘、数字 IO 等信号。CN1 和 CN2 的管脚编号均为奇偶排交错顺序编号，且 1#管脚处有三角箭头指示。

EM9283 所有管脚的信号电平，均为 LVTTTL (3.3V) 电平。除非特殊说明，输入管脚应避免接入 5V 电平信号。对低电平有效的信号，信号名称后均带“#”表示。

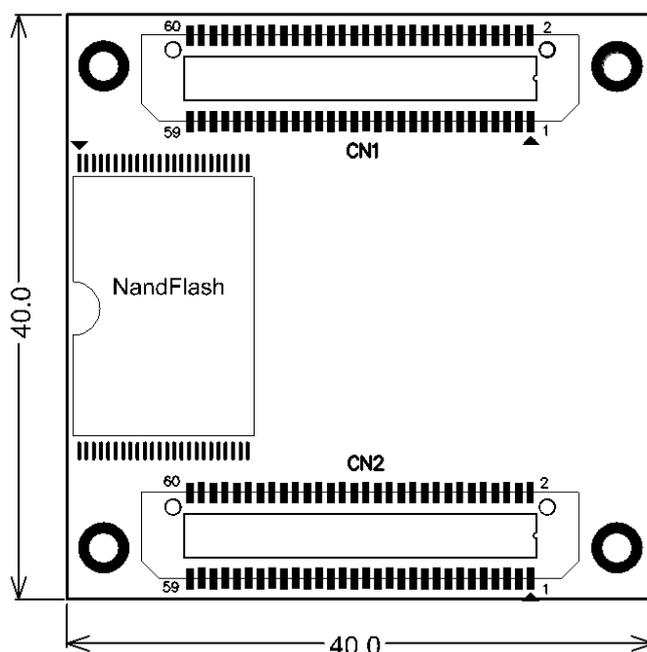


图 3.1 EM9283 焊接面上 CN1 和 CN2 所在位置示意图

以下对 EM9283 所有管脚信号列表逐一说明。

### 3.1 EM9283 的 CN1 信号定义

EM9283 的 CN1 主要包括电源、异步串口、USB、GPIO 等信号。CN1 各管脚的定义如下：

信号名称及简要描述	CN1		信号名称及简要描述
	PIN	PIN	
+5V 输入	1	2	+5V 输入
+5V 输入	3	4	+5V 输入
+5V 输入	5	6	+5V 输入
GND, 公共地 (电源地)	7	8	GND, 公共地 (电源地)
GND, 公共地 (电源地)	9	10	GND, 公共地 (电源地)
GND, 公共地 (电源地)	11	12	GND, 公共地 (电源地)
LI-ION_BATT, 锂电池输入	13	14	LI-ION_BATT, 锂电池输入
LI-ION_BATT, 锂电池输入	15	16	LI-ION_BATT, 锂电池输入
VDD4P2, 4.2V 输出	17	18	VDD4P2, 4.2V 输出
VDD4P2, 4.2V 输出	19	20	VDD4P2, 4.2V 输出
系统保留	21	22	VDDIO_3V3, 数字 3.3V 输出
系统保留	23	24	VDDIO_3V3, 数字 3.3V 输出
PSWITCH (一键开关机)	25	26	RSTIN#, 外部复位输入
USB0_OTG_ID	27	28	DBGSL#, 运行模式选择
USB0_OTG_DP	29	30	USB0_OTG_VBUS
USB0_OTG_DM	31	32	USB0_OTG_VBUS
USB1_HDP	33	34	USB1_VBUS
USB1_HDM	35	36	USB1_VBUS
DBG_RXD	37	38	DBG_TXD
RXD2, COM2 数据串行输入	39	40	TXD2, COM2 数据串行输出
GPIO0 / CTS2#	41	42	GPIO1 / RTS2#
GPIO2 / DSR2#	43	44	GPIO3 / DTR2#
GPIO4 / RI2#	45	46	GPIO5 / DCD2#
RXD3, COM3 数据串行输入	47	48	TXD3, COM3 数据串行输出
RXD4, COM4 数据串行输入	49	50	TXD4, COM4 数据串行输出
GPIO6 / RXD5	51	52	GPIO7 / TXD5
GPIO8 / RXD6	53	54	GPIO9 / TXD6
GPIO10 / I2C_SCL	55	56	GPIO11 / I2C_SDA
GPIO12 / SPI_MISO	57	58	GPIO13 / SPI_MOSI
GPIO14 / SPI_SCLK	59	60	GPIO15 / SPI_CS0N

关于 CN1 中相关信号的进一步说明：

**模块的电源管理**，CN1 上的 1# - 26#管脚都是有关各种电源的输入输出以及开关复位启动功能，可满足手持式设备对电源管理的基本需求。有关 EM9283 的电源管理机制，在第 4 节中有专门说明。

**USB OTG 端口**，EM9283 包含一个标准 USB OTG 接口，共 5 条引线：

USB OTG 接口定义	简要说明
USB0_OTG_DP	USB OTG 双向差分数据线
USB0_OTG_DM	USB OTG 双向差分数据线
USB0_OTG_VBUS	双向电源
GND	公共地
USB0_OTG_ID	连接类型标志，带上拉电阻。

上述 5 条引线可直接接到底板的微型 AB 插座(mini-AB)。在通常情况下，若连接带线使 USB\_OTG\_ID 变低（即微型 A 插头），则 EM9283 将作为主控端；若连接带线使 USB0\_OTG\_ID 保持高电平（即微型 B 插头），则 EM9283 将作为设备端。在实际使用中，若连接对端也具有 USB OTG 功能，则 USB OTG 将通过主机通信协议（HNP）根据实际连接的设备类型，动态切换主机和设备角色。因此即使 USB0\_OTG\_ID 的电平与设备类型不符，同样可以实现正常连接。

**异步串口 UART**，EM9283 上有 5 个 UART 串口可供应用程序使用，分别为 COM2 – COM6，其中的 COM5 和 COM6 与 GPIO 复用管脚。COM2 的硬件握手信号 CTS/RTS，以及 4 条 Modem 信号线也与 GPIO 复用管脚。当“COM2:”口打开时，仅有 COM2 口的收发数据线 RXD/TXD 是激活的，其他的控制线仍然处于 GPIO 的状态，只有通过 DCB 设置了相关参数后，对应的信号线才会被激活。另外 EM9283 还有一个独立的调试串口，主要是用于输出系统的相关信息，在正常使用中不需要引出调试串口。但在开发阶段，调试串口的输出的信息是有帮助的。调试串口的波特率为 115200bps，数据帧格式为 8-N-1。

**I2C 和 SPI 接口**，EM9283 带有 I2C 总线接口和 SPI 接口，在这些接口的工作模式中，EM9283 都是作为接口的 Master 方，与连接在总线或接口上的设备端进行通讯的。

**CN1 上的复用管脚**，除了 COM2 的 Modem 信号线外，在 CN1 上还有 I2C、SPI 接口信号与 GPIO 复用管脚。主板上电启动后，这些管脚的缺省配置是处于 GPIO 状态的，只有当应用程序打开接口的驱动设备文件“**I2C1:**”和“**SPI1:**”后，对应的管脚才会处于接口信号的状态。在 CN1 中包括 EM9283 的 16 位通用数字 IO: GPIO0 – GPIO15，它们都与某种接口功能复用管脚:

GPIO0	与 COM2 口的硬件流控信号 CTS#复用管脚。
GPIO1	与 COM2 口的硬件流控信号 RTS#复用管脚。
GPIO2	与 COM2 口的 Modem 信号 DSR#复用管脚。
GPIO3	与 COM2 口的 Modem 信号 DTR#复用管脚。
GPIO4	与 COM2 口的 Modem 信号 RI#复用管脚。
GPIO5	与 COM2 口的 Modem 信号 DCD#复用管脚。
GPIO6	与 COM5 的数据输入 RXD 复用管脚。
GPIO7	与 COM5 的数据输出 TXD 复用管脚。
GPIO8	与 COM6 的数据输入 RXD 复用管脚。
GPIO9	与 COM6 的数据输出 TXD 复用管脚。
GPIO10	与 I2C 总线的时钟信号 SCL 复用管脚。
GPIO11	与 I2C 总线的数据信号 SDA 复用管脚。
GPIO12	与 SPI 接口的数据输入信号 MISO 复用管脚。
GPIO13	与 SPI 接口的数据输出信号 MOSI 复用管脚。
GPIO14	与 SPI 接口的时钟控制信号 SCLK 复用管脚。
GPIO15	与 SPI 接口的片选控制信号 CS0N 复用管脚。

### 3.2 EM9283 的 CN2 信号定义

EM9283 的 CN2 管脚，主要包括 LCD 接口、矩阵键盘、数字 IO 等信号。CN2 各管脚的定义如下：

信号名称及简要描述	CN2		信号名称及简要描述
	PIN	PIN	
XM, 触摸屏 X 方向差分输入-	1	2	YM, 触摸屏 Y 方向差分输入-
XP, 触摸屏 X 方向差分输入+	3	4	YP, 触摸屏 Y 方向差分输入+
AIN1, 模拟单端输入	5	6	AIN0, 模拟单端输入
GND, 公共地 (电源地)	7	8	GND, 公共地 (电源地)
LCD_B2, 蓝色分量 LSB	9	10	LCD_B3
LCD_B4	11	12	LCD_B5
LCD_B6	13	14	LCD_B7, 蓝色分量 MSB
LCD_G2, 绿色分量 LSB	15	16	LCD_G3
LCD_G4	17	18	LCD_G5
LCD_G6	19	20	LCD_G7, 绿色分量 MSB
LCD_R2, 红色分量 LSB	21	22	LCD_R3
LCD_R4	23	24	LCD_R5
LCD_R6	25	26	LCD_R7, 红色分量 MSB
GND, 公共地 (电源地)	27	28	GND, 公共地 (电源地)
LCD_VSYNC, LCD 帧同步	29	30	LCD_HSYNC, LCD 行同步
LCD_DOTCLK, LCD 数据时钟	31	32	LCD_DEN, LCD 显示使能
GND, 公共地 (电源地)	33	34	GND, 公共地 (电源地)
系统保留	35	36	系统保留
LCD_PWR, LCD 电源控制	37	38	LCD_BKLIGHT, LCD 背光控制
GPIO31 / PWM4	39	40	GPIO30 / PWM3
GPIO29 / PWM2	41	42	GPIO28 / PWM1
GPIO27 / SD_D3	43	44	GPIO26 / SD_D2
GPIO25 / SD_D1	45	46	GPIO24 / SD_D0
GPIO23 / SD_CLK	47	48	GPIO22 / SD_CMD
GPIO21 / SD_PWR	49	50	GPIO20 / SD_DET
GPIO19 / KEYOUT4	51	52	GPIO18 / KEYIN4
GPIO17 / KEYOUT3	53	54	GPIO16 / KEYIN3
KEYOUT2, 矩阵键盘扫描输出	55	56	KEYIN2, 矩阵键盘状态输入
KEYOUT1, 矩阵键盘扫描输出	57	58	KEYIN1, 矩阵键盘状态输入
KEYOUT0, 矩阵键盘扫描输出	59	60	KEYIN0, 矩阵键盘状态输入

关于 CN2 中相关信号的进一步说明：

**矩阵键盘**，EM9283 矩阵键盘驱动的缺省配置是 3x3 的结构，即可支持 9 个按键。客

户可根据自身应用需要，通过注册表激活 KEYIN3 – KEYIN4，KEYOUT3 – KEYOUT4，最大可支持 5×5，即 25 个按键。

**显示接口**，EM9283 既可支持单色点阵 LCD，也可支持彩色 TFT LCD，单色 LCD 的连接说明见每 3.3 节“EM9283 连接单色 LCD 模块”。当连接 24 位彩色 TFT 屏时，EM9283 的 R2-R7、G2-G7、B2-B7 与彩色 TFT 屏的 LCD 对应管脚直接连接，TFT LCD 屏的 R0\R1\G0\G1\B0\B1 可直接接地。EM9283 提供了 LCD\_PWR 和 LCD\_BKLIGHT 两个引脚用于管理 LCD 屏的电源和背光。当系统关闭 LCD 显示时，LCD\_PWR 输出低电平的控制信号，反之，系统开启 LCD 显示时，LCD\_PWR 输出高电平。LCD\_BKLIGHT 是 1 路脉宽调制输出信号，用于 LCD 背光控制，当系统点亮 LCD 背光时，LCD\_BKLIGHT 输出占空比可调的 PWM 信号，系统关闭 LCD 背光时，LCD\_BKLIGHT 被置为低电平。EM9283 有专门的驱动程序管理 LCD 的背光，并在<控制面板>中提供有相应的设置界面。

**AD 通道**，EM9283 有 2 路 AD 通道 AIN1 和 AIN2，供应用程序监测慢变化的模拟信号。模拟输入量程为 0V – 3.6V，AD 分辨率为 12-bit。EM9283 提供有专门的设备驱动，应用程序只需打开设备文件“LDC1:”，通过适当的命令就可读取模拟通道的电压值了。

**CN2 中的 GPIO**：在缺省状态下，CN2 中的 GPIO16 – GPIO31 管脚均处于数字输入状态，当应用程序打开相应的设备驱动程序时，对应的管脚会自动切换到复用的功能管脚。复用功能如下表所示：

	功能复用
GPIO16	与矩阵键盘的状态输入 KEYIN3 复用管脚。
GPIO17	与矩阵键盘的扫描输出 KEYOUT3 复用管脚。
GPIO18	与矩阵键盘的状态输入 KEYIN4 复用管脚。
GPIO19	与矩阵键盘的扫描输出 KEYOUT4 复用管脚。
GPIO20	与 SD 卡物理检测信号 SD_DET 复用管脚。
GPIO21	与 SD 卡接口电源控制信号 SD_PWR 复用管脚。
GPIO22	与 SD 卡接口控制信号 SD_CMD 复用管脚。
GPIO23	与 SD 卡接口控制信号 SD_CLK 复用管脚。

GPIO24	与 SD 卡接口数据线 SD_D0 复用管脚。
GPIO25	与 SD 卡接口数据线 SD_D1 复用管脚。
GPIO26	与 SD 卡接口数据线 SD_D2 复用管脚。
GPIO27	与 SD 卡接口数据线 SD_D3 复用管脚。
GPIO28	与 PWM1 复用管脚。PWM1 设备文件名 = “PWM1:”
GPIO29	与 PWM2 复用管脚。PWM2 设备文件名 = “PWM2:”
GPIO30	与 PWM3 复用管脚。PWM3 设备文件名 = “PWM3:”
GPIO31	与 PWM4 复用管脚。PWM4 设备文件名 = “PWM4:”

### 3.3 EM9283 连接单色 LCD 模块

EM9283 的 LCD 接口既可支持彩色 TFT LCD 显示模块，也可支持单色 STN LCD 显示模块。在手持式设备中，从成本和供货稳定性方面考虑，分辨率为 160×160 的单色 LCD 模块已成为面向工业应用的手持式设备的主流选择之一。当 EM9283 与这类单色屏连接时，所需的信号如下表所示：

CN2	EM9283	单色 LCD 屏接口信号
13	LCD_B6	LCD_D0, LCD 数据总线（最低位 LSB）
14	LCD_B7	LCD_D1, LCD 数据总线
15	LCD_G2	LCD_D2, LCD 数据总线
16	LCD_G3	LCD_D3, LCD 数据总线
17	LCD_G4	LCD_D4, LCD 数据总线
18	LCD_G5	LCD_D5, LCD 数据总线
19	LCD_G6	LCD_D6, LCD 数据总线
20	LCD_G7	LCD_D7, LCD 数据总线（最高位 MSB）
29	LCD_VSYNC	LCD_RD#, LCD 读脉冲，低有效，Intel 时序
30	LCD_HSYNC	LCD_WE#, LCD 写脉冲，低有效，Intel 时序
31	LCD_DOTCLK	LCD_RS, LCD 命令/数据寄存器选择
32	LCD_DEN	LCD_CS#, LCD 片选，低有效
37	LCD_PWR	LCD 电源控制
38	LCD_BKLIGHT	LCD 背光控制信号

## 4、电源管理及相关说明

图 4.1 是 EM9283 的电源管理单元功能框图，包含了锂电池充电管理，高性能 DC-DC，内存、CPU、外设电源管理等模块。VDD4P2 和 VDDIO\_3V3 是 EM9283 提供的两路电源输出，供外部扩展电路使用。同时 EM9283 PMU 还支持一键开/关机、实时时钟等功能。

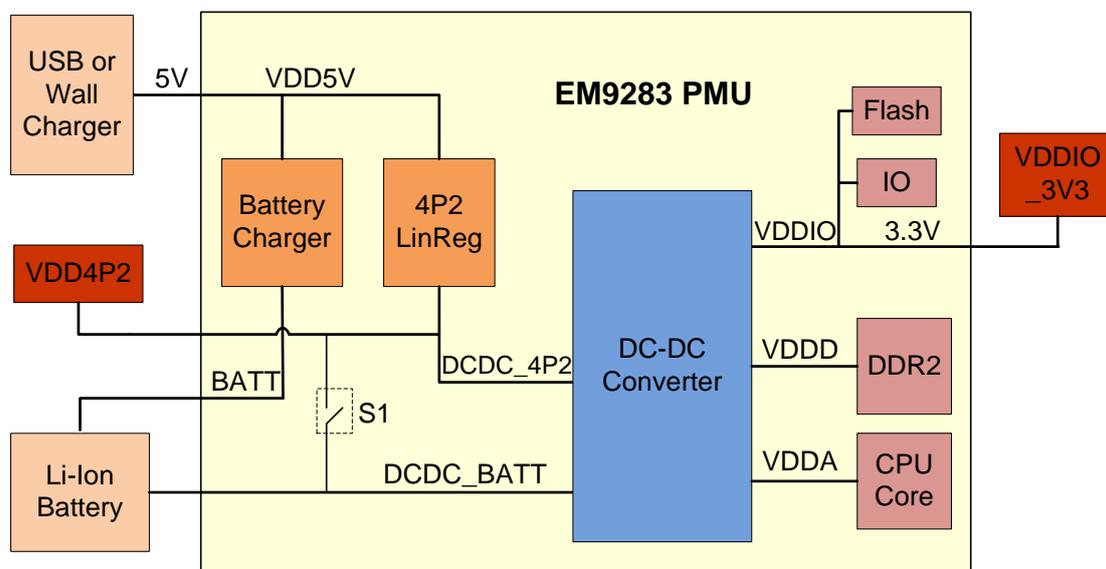


图 4.1 EM9283 电源管理功能框图

### 4.1 电源输入

EM9283 支持两种类型的电源供电方案：

1. +5V：外部供电输入
2. LI-ION\_BATT：锂电池供电输入

EM9283 可选择外部 5V 或锂电池作为系统电源，外部 5V 电源同时作为锂电池的充电电源。请使用标称电压为 3.7V 的锂电池为系统供电，在锂电池供电的情况下，当电池电压低于 3.0V 时，EM9283 将自动关机，以避免锂电池过放电。当电池电压低于 3.0V 时，EM9283 不能正常开机，此时需要连接 5V 电源为电池充电。一旦 EM9283 连接了 5V 电源，无论锂电池是否连接，系统均会正常启动。

如果要实现通过 USB 为系统供电，请将 USB0\_OTG\_VBUS 与 +5V 连接在一起。

### 4.2 电源输出

EM9283 提供两路电源输出：

1. VDD4P2：4.2V 输出

## 2. VDDIO\_3V3: 3.3V 输出

一旦 EM9283 连接了 5V 电源,内部的线性电源 4P2 线性稳压源就开始工作,输出 4.2V 的电压(简称 VDD4P2),同时 DC-DC 转换器选择 DCDC\_4P2 作为其输入电源。当 EM9283 没有连接 5V 电源,使用锂电池供电时,EM9283 电源管理单元内的单向开关 S1 将自动导通,VDD4P2 与锂电池直接相连,内部 DC-DC 转换器选择 DCDC\_BATT 作为供电电源。

EM9283 内部的 DC-DC 提供了三组电源 VDDA、VDDD、VDDIO,分别供 CPU 内核、系统内存和系统 Flash 使用。EM9283 为底板上的数字电路的供电的 VDDIO\_3V3 与系统 NandFlash 和 IO 共用一组电源 VDDIO。

**VDD4P2 与 VDDIO\_3V3 作为系统电源的一部份,在使用时需要特别注意不能超出其输出电流限制,否则将影响系统正常运行,甚至造成硬件上的损坏!**

## 4.3 锂电池充电管理

EM9283 直接使用 iMX283 的电池充电系统,对锂电池的充电过程分成恒流和恒压充电两个阶段。恒流充电阶段,系统将充电电压设置为 4.2V,输出恒定的电流为电池充电,电池电压逐步上升。当电池电压接近 4.2V 时,系统逐渐减小充电电流,并保持充电电压为 4.2V,开始恒压充电,至到电池充满。**当电池充满后,EM9283 会自动停止对锂电池的充电,但仍然会用一个微小的电流周期性的尝试对电池充电,所以当电池充满电后,应及时将充电器断开,不然可能会影响电池性能甚至损坏电池。**

## 4.4 电流限制

EM9283 可使用外部 5V 或 USB 为系统供电和作为充电电源,当使用 USB 供电时,根据目前 USB 规范,USB 口的输出电流上限为 500mA,所以当系统设置为使用 USB 供电时,EM9283 PMU 将限制系统的电大电流消耗为 450mA,既对锂电池的充电电流加上 VDD4P2 输出电流(包含 DCDC\_4P2 的电流),总电流小于 450mA。默认情况下,USB 供电时锂电池充电电流被设置为 300mA,但当 VDD4P2 负载加重,需要更多电流,而此时总电流已经达到 450mA 时,系统驱动会自动减小锂电池充电电流,优先供 VDD4P2 使用,4P2 线性稳压源具有电流的优先使用权。用户在设计基于 USB 充电的整机时,必须注意 USB 口最大电流的限制,使整机设备符合 USB 外设规范。

使用外部 5V 为 EM9283 供电时,EM9283 的最大电流限制为 780 mA,其中锂电池充电电流为 500 mA,4P2 线性稳压源同样具有电流的优先使用权。

为了保证整机设备的可靠工作,不推荐在对电池充电的同时使用设备。推荐的工作模式是首先把设备电池充满,然后在电池供电状态使用设备。这样的使用模式也符合绝大多数专

业手持式设备的应用需求。

#### 4.5 开关机与外部复位

EM9283 提供了一个开关机引脚 PSWITCH，在 PSWITCH 与 LI-ION\_BATT 引脚之间连接一个按键，就能实现一键开\关机、挂起\唤醒功能。PSWITCH 引脚功能仅在电池供电的情况下有效。

EM9283 支持外部手动复位，在外部 5V 电源供电时，将 RSTIN#拉低，系统将硬件重启，在锂电池供电时，将 RSTIN#拉低，系统会直接关机。建议在由电池供电的系统中，将 RSTIN#引脚悬空，由 PSWITCH 引脚统一管理系统开关机。

**禁止在 PSWITCH 与 RSTIN#引脚上连接上拉\下拉电阻或电容。**

#### 4.6 电源直流电气特性

EM9283 电源输入特性（表 4.2）

EM9283 管脚	最小值	典型值	最大值	单位
+5V 输入	4.75	5.00	5.25	V
LI-ION_BATT, 锂电池输入 <sup>1</sup>	3.0	3.7	4.24	V
PSWITCH <sup>2</sup>		LI-ION_BATT		

表 4.2 EM9283 电源输入特性

1. 锂电池单独供电时，电池电压低于 3.0V，系统自动关机。
2. PSWITCH 通过按键直接与锂电池相连。

EM9283 电源输出特性（表 4.3）

电源输出参数	最小值	典型值	最大值	单位
VDDIO_3V3 输出电压 <sup>1</sup>	3.0	3.3	3.4	V
VDDIO_3V3 最大输出电流 <sup>2</sup>			200	mA
VDD4P2 输出电压 <sup>3</sup> (4.2V)	-3	—	+3	%
VDD4P2 输出电流 <sup>4</sup> (USB 供电)			100	mA
VDD4P2 输出电流 <sup>4</sup> (+5V 电源供电)			400	mA
锂电池充电限制电压 (4.2V)	-2	—	+1	%
锂电池充电电流 <sup>5</sup> (USB 供电)			300	mA
锂电池充电电流 <sup>5</sup> (+5V 电源供电)			500	mA
锂电池停止充电电流 <sup>6</sup>			20	mA

表 4.3 EM9283 电源输出特性

1. 系统挂起时，VDDIO\_3V3 输出电压为 3.0V。
2. VDDIO\_3V3 与 EM9283 主板上的 NandFlash 和 CPU IO 共用一组 3.3V 电源。所以 VDDIO\_3V3 能提供的实际最大电流受系统其它电源负载和 IO 口电流使用情况的影响。
3. 测试条件为系统 5V 供电，VDD4P2 空载。当 EM9283 由锂电池供电时，VDD4P2 输出电压等于锂电池电源。
4. VDD4P2 能提供的实际最大电流受系统其它电源负载和 IO 口电流使用情况的影响。当由锂电池供电时，VDD4P2 的输出电流受锂电池最大输出电流的限制。
5. 锂电池充电电流会受系统其它电源负载影响而自动减小，详见 [4.4 电流限制](#)。
6. 当锂电池充电电流小于 20mA 时，EM9283 将停止充电。

EM9283 电源功耗指标（表 4.4）

主板运行状态	典型值	最大值	单位
系统运行(CPU 负载小于 10%) <sup>1, 2</sup>	100	—	mA
系统挂起 <sup>1, 2</sup>	8	—	mA
系统关机 <sup>1, 2, 3</sup>	15	—	uA

表 4.4 系统电流消耗

1. 典型值均是 EM9283 在锂电池供电的情况下,在 EM9283 评估底板上测得的平均值。
2. 电流消耗的最大值受实际电路情况的影响。
3. 系统关机时，CPU 的实时时钟电路会继续工作。

## 5、数字 IO 信号的基本电气特性

EM9283 共引出 32 位通用数字 IO（也称为 GPIO），均为 3.3V TTL 电平，具体电气参数如下表所示：

	Min（最小值）	Max（最大值）	简要说明
$V_{IL}$	-	0.8V	输入低电平
$V_{IH}$	2.0V	3.3V	输入高电平
$V_{OL}$	-	0.4V	输出低电平
$V_{OH}$	2.4V	-	输出高电平
$I_{OH}$	-8mA	-	输出高电平时源电流
$I_{OL}$	8mA	-	输出低电平时吸电流
$I_{IL}$	-	10uA	输入低电平时的泄漏电流
$I_{IH}$	-	10uA	输入高电平时的泄漏电流

表 5.1 GPIO 直流特性

EM9283 除电源接口，USB 接口和 ADC 接口（包括触摸屏接口）外，都是 3.3VTTL 电平信号，其 DC 电气特性与 EM9283 的 GPIO 一致，具体参数见表 5.1。

EM9283 的 GPIO 和串口上电时均为“高阻输入状态”，部份 IO 有内部上拉电阻，有内部上拉电阻的 IO 口，在上电时的状态就为“高电平输入状态”。而没有上拉电阻的 IO 口，上电时的电平就不确定，如果用户希望这些 IO 口在上电时有确定的电平状态，需要增加相应的上拉或下拉电阻。

EM9283 有内部上拉电阻的引脚如下表所示：

管脚	GPIO6	GPIO7	GPIO8	GPIO9	GPIO12	GPIO13	GPIO14
上拉电阻	47K						
管脚	GPIO15	GPIO20	GPIO22	GPIO24	GPIO25	GPIO26	GPIO27
上拉电阻	47K	10K	10K	47K	47K	47K	47K
管脚	RXD4	TXD4					
上拉电阻	47K	47K					

表 5.2 EM9283 管脚上拉电阻

## 6、EM9283 的相关功能的说明

### WDT 看门狗定时器

EM9283 直接使用了 iMX283 芯片内部的独立看门狗定时器，系统启动后设置看门狗的超时时间为 10 秒，且由 WinCE 内核的 Watchdog 线程按 5 秒间隔对看门狗进行刷新。此模式可以防止应用程序占用 CPU 的死循环，但对应用程序异常退出或挂起没有作用。

EM9283 为应用程序设计了专门的 WDT 驱动程序，应用程序可通过打开 WDT 设备文件“WDT1:”来接管对看门狗的操作，使之更为全面的监管应用程序行为的有效性。应用程序接管看门狗后，需按 5 秒的间隔对看门狗进行刷新操作。用户应用程序可通过 Read 函数来获取 WDT 加载周期，通过 Write 函数执行看门狗刷新操作，主要代码如下：

#### 打开 WDT 文件

```
HANDLE hWDT;
hWDT = CreateFile(_T("WDT1:"),           // name of device
    GENERIC_READ|GENERIC_WRITE,        // desired access
    FILE_SHARE_READ|FILE_SHARE_WRITE, // sharing mode
    NULL,                               // security attributes (ignored)
    OPEN_EXISTING,                      // creation disposition
    FILE_FLAG_RANDOM_ACCESS,           // flags/attributes
    NULL);                              // template file (ignored)
```

#### 获取 WDT 刷新周期

```
DWORD    dwWDTPeriod;                //in ms
DWORD    dwNumberOfBytesRead = 0;
BOOL     bRet;

bRet = ReadFile(hWDT, &dwWDTPeriod, sizeof(DWORD),
    &dwNumberOfBytesRead, NULL);
```

#### 执行 WDT 刷新操作

```
DWORD    dwNumberOfBytesWritten = 0;
BOOL     bRet;

bRet = WriteFile(hWDT, NULL, 0, &dwNumberOfBytesWritten, NULL);
```

## USB 接口

EM9283 可提供 2 个 USB 端口：一个高速主控接口，和一个 USB OTG 接口。EM9283 的 USB 主控接口一般用于连接用户需要的扩展设备，例如 USB 接口的 camera 等，所以在实际的手持式设备中 USB1 主控口不一定会引出。EM9283 的 USB OTG 接口，通常会在应用底板上，通过一个 miniAB 插座引出。在开发阶段，EM9283 通过 USB OTG 与开发主机的一个 USB 接口相连，开发主机通过微软的 ActiveSync 通讯协议，来对 EM9283 进行若干操作，包括把系统配置文件 userinfo.txt 拷贝到 EM9283 的“\NandFlash”中（userinfo.txt 用于在运行状态自动启动用户的应用程序）、在线调试用户的应用程序、管理 EM9283 上的文件等等。另外 ActiveSync 还把 USB 设备口映射成串口，占用串口逻辑号 COM1，所以 EM9283 真正的物理串口对应的逻辑编号从 COM2 开始。

## UART 异步串口

EM9283 为应用提供了 5 个串口，对应的逻辑编号为 COM2 – COM6，每个串口都支持高达 3Mbps 的波特率。其中 COM2 口配置有流控及 Modem 信号，其他的串口都是普通的三线制串口。对 COM2，只有当设置了流控使能后，对应的硬件流控信号才会被激活。需要主要代码如下：

打开串口设备文件

```
HANDLE hSer;
hSer = CreateFile(_T("COM2:"),           // name of device
    GENERIC_READ|GENERIC_WRITE,         // desired access
    FILE_SHARE_READ|FILE_SHARE_WRITE, // sharing mode
    NULL,                                // security attributes (ignored)
    OPEN_EXISTING,                       // creation disposition
    FILE_FLAG_RANDOM_ACCESS,             // flags/attributes
    NULL);                                // template file (ignored)
```

激活串口 RTS/CTS 硬件握手功能

```
DCB SerDCB;

SerDCB.DCBlength = sizeof(DCB);
GetCommState(hSer, &SerDCB);           // 从驱动读取当前DCB
```

```
SerDCB.fRtsControl = RTS_CONTROL_HANDSHAKE;
SetCommState(hSer, &SerDCB);           // 再设置回驱动
```

除了 COM2 口可配置硬件流控外，EM9283 的所有应用串口（包括 COM2 口）还可选择一个 GPIO 作为 RTS，实现硬件方向控制。可作为 RTS 硬件方向控制的 GPIO 有：GPIO10-GPIO31。应用程序设置串口 RTS 硬件方向控制的方法如下：

#### 打开串口设备文件

```
HANDLE hSer;
hSer = CreateFile(_T("COM6:"),           // name of device
    GENERIC_READ|GENERIC_WRITE,         // desired access
    FILE_SHARE_READ|FILE_SHARE_WRITE,   // sharing mode
    NULL,                                // security attributes (ignored)
    OPEN_EXISTING,                       // creation disposition
    FILE_FLAG_RANDOM_ACCESS,             // flags/attributes
    NULL);                                // template file (ignored)
```

#### 设置一位 GPIO 作为 RTS

```
DWORD dwRtsGpioPin = GPIO26;           //选择 GPIO26 作为 RTS

If (!DeviceIoControl (hSer,
    IOCTL_SET_UART_RTS_PIN,
    & dwRtsGpioPin, sizeof(DWORD),
    NULL, 0,
    NULL, NULL))
{
    // 出错处理。。。
}
```

#### 设置串口 RTS 控制模式

```
DCB SerDCB;

SerDCB.DCBlength = sizeof(DCB);
GetCommState(hSer, &SerDCB);         // 从驱动读取当前 DCB
SerDCB.fRtsControl = RTS_CONTROL_TOGGLE;
SetCommState(hSer, &SerDCB);         // 再设置回驱动
```

#### I<sup>2</sup>C 接口

EM9283 的 I<sup>2</sup>C 接口为 2 线制标准 I<sup>2</sup>C 接口，信号电平为 3.3V 的 TTL 电平（LVTTTL），最高传输波特率为 400kbps。在使用 I<sup>2</sup>C 接口时，应对 SCL 和 SDA 两个信号线均加 10K

的上拉电阻，在高波特率的情况下，上拉电阻是必须的。EM9283 板上已固化了面向 I<sup>2</sup>C 接口的 WinCE 标准驱动程序，应用程序只需打开文件名为“I2C1:”的文件对象，就可通过标准的 ReadFile (...) 和 WriteFile (...) 函数进行 I<sup>2</sup>C 数据传输了。

基本的 I<sup>2</sup>C 数据结构如下：

```
typedef struct
{
    BYTE    uHwAddr;    // 7-bit I2C 器件地址 + 1-bit 读写标志 (LSB)
    DWORD   dwCmd;      // 对 I2C 器件发送的命令，可选
    PBYTE   pDatBuf;    // 指向存储读写数据的 buffer
    DWORD   dwDatLen;   // 需要读写数据的字节长度
} I2C_INFO, *PI2C_INFO;
```

在上述结构中，dwCmd = 0xFFFFFFFF，表示无效命令，驱动程序会跳过命令的发送；若 dwCmd 的最高位 (D31) = 0，表示为单字节命令，最低字节 (D7 - D0) 将作为命令被发送；若 dwCmd 的最高位 (D31) = 1，表示为双字节命令，驱动程序会首先发送高字节 (D15 - D8)，然后再发送低字节 (D7 - D0) 命令。dwCmd 通常为 I<sup>2</sup>C 器件寄存器的起始地址。I<sup>2</sup>C 操作的主要代码如下：

打开 I<sup>2</sup>C 文件

```
HANDLE hI2C;
hI2C = CreateFile(_T("I2C1:"),           // name of device
    GENERIC_READ|GENERIC_WRITE,         // desired access
    FILE_SHARE_READ|FILE_SHARE_WRITE,   // sharing mode
    NULL,                                // security attributes (ignored)
    OPEN_EXISTING,                       // creation disposition
    FILE_FLAG_RANDOM_ACCESS,             // flags/attributes
    NULL);                                // template file (ignored)
```

从 I<sup>2</sup>C 器件读取数据

```
I2C_INFO  I2cInfo;
DWORD     dwNumberOfBytesRead = 0;
BOOL      bRet;

// I2C 数据结构初始化，注意数据 buffer 指针一定要指向有效 buffer
// ... ..
bRet = ReadFile(hI2C, & I2cInfo, sizeof(I2cInfo),
    &dwNumberOfBytesRead, NULL);
```

### 向 I<sup>2</sup>C 器件写入数据

```
I2C_INFO  I2cInfo;
DWORD     dwNumberOfBytesWritten = 0;
BOOL      bRet;

// I2C 数据结构初始化, 注意数据 buffer 指针一定要指向有效 buffer
// ... ..
bRet = WriteFile(hI2C, &I2cInfo, sizeof(I2cInfo),
                 &dwNumberOfBytesWritten, NULL);
```

用户可从 EM9283 的资料光盘中的 I2C 应用范例了解其详细的使用方法。

### SPI 同步串口

EM9283 的 SPI 接口为 4 线制标准 SPI 接口, 信号电平为 3.3V 的 TTL 电平 (LVTTTL), 最高传输波特率为 10Mbps。主要应用于设备内部各功能单元之间的短距离高速传输。EM9283 板上已固化了面向 SPI 接口的 WinCE 标准驱动程序, 应用程序只需打开文件名为“SPI1:”的文件对象, 就可通过 SPI 进行数据传输了。用户可从 EM9283 的资料光盘中的 SPI 应用范例了解其详细的使用方法。

### PWM 脉冲输出

EM9283 共有 4 路 PWM 输出, 其最高输出频率可达 12MHz, 但如果希望保证一定精度的占空比 (1%的精度), 则输出最高频率只能到 240KHz。EM9283 板上已固化了面向 PWM 输出的 WinCE 标准驱动程序, 应用程序只需打开文件名为“PWM1:”- “PWM4:”的文件对象, 再通过 WriteFile 设置启动 PWM 脉冲的参数 (频率和占空比) 即可, 应用程序也可通过 WriteFile 随时停止 PWM 的输出。典型的 PWM 应用, 包括为红外串口提供调制信号 (38.5KHz, 50%占空比)、为 ISO7816 提供时钟信号 (3.5712MHz, 9600bps 波特率)。

基本的 PWM 数据结构如下:

```
typedef struct
{
    DWORD     dwFreq;           // 单位为Hz, = 0: 停止PWM输出
    DWORD     dwDuty;          // 单位为%, 1 - 99
    DWORD     dwResolution;    // = 1: unit; = 10: 0.1 unit; = 100: 0.01 unit
} PWM_INFO, *PPWM_INFO;
```

上述结构中，一般分辨率 `dwResolution` 选为 1，这时 `dwFreq=1000000`，就表示输出脉冲频率为 1MHz；若 `dwResolution = 10`，`dwFreq=1000000` 则表示 100KHz。PWM 操作的主要代码如下：

#### 打开 PWM 文件

```
HANDLE hPWM;  
hWDT = CreateFile(_T("PWM1:"),           // name of device  
    GENERIC_READ|GENERIC_WRITE,         // desired access  
    FILE_SHARE_READ|FILE_SHARE_WRITE, // sharing mode  
    NULL,                                // security attributes (ignored)  
    OPEN_EXISTING,                       // creation disposition  
    FILE_FLAG_RANDOM_ACCESS,             // flags/attributes  
    NULL);                               // template file (ignored)
```

#### 执行 PWM 输出操作

```
PWM_INFO PwmInfo;  
DWORD    dwNumberOfBytesWritten = 0;  
BOOL     bRet;  
  
PwmInfo.dwFreq = 3571200;           // ISO7816 时钟: 3.5712MHz  
PwmInfo.dwDuty = 50;                // 占空比 50%  
PwmInfo.dwResolution = 1;  
bRet = WriteFile(hPWM, &PwmInfo, sizeof(PwmInfo),  
    &dwNumberOfBytesWritten, NULL);
```

#### 获取 PWM 设置的实际参数

```
PWM_INFO PwmInfo;  
DWORD    dwNumberOfBytesRead = 0;  
BOOL     bRet;  
  
bRet = ReadFile(hPWM, &PwmInfo, sizeof(PwmInfo),  
    &dwNumberOfBytesRead, NULL);
```

关闭设备文件，将停止 PWM 脉冲输出。

## AD 模拟通道

EM9283 共有 2 路低速的模拟 AD 通道 AIN0 和 AIN1，输入量程为 0 – 3.6V，AD 分辨率 12-bit。所谓低速通道，表示这两个通道只能用于外部的直流或慢变化类型的信号。除此之外，EM9283 还可提供对输入的+5V 电源电压、EM9283 板卡本身运行的环境温度以及核心 CPU 基片温度的监测。应用程序在打开 AD 驱动程序对应的设备文件“LCD1:”后，可多次调用 ReadFile 来读取各类数据。需要设置的命令及参数如下：

```
#define EM9280_DAQ_VOLTAGE_CH0      0
#define EM9280_DAQ_VOLTAGE_CH1      1
#define EM9280_DAQ_VDD_5V           2
#define EM9280_DAQ_CPU_TEMPERATURE   6
#define EM9280_DAQ_BOARD_TEMPERATURE 7

typedef struct
{
    DWORD    dwCmd;           // 命令码 = 0, 1, 2, ....
    DWORD    dwData;         // 返回的AD数据
    char     UnitName[16];    // 返回的单位字符串: "mV", "Kalvin"等
} DAQ_INFO, *PDAQ_INFO;
```

注意返回的温度参数均为开氏温度，转换成摄氏温度，大致减去 273 即可。在此基础上，应用程序的典型代码如下：

### 打开 AD 文件

```
HANDLE hLRADC;
hLRADC = CreateFile(_T("LDC1:"),           // name of device
    GENERIC_READ|GENERIC_WRITE,          // desired access
    FILE_SHARE_READ|FILE_SHARE_WRITE,    // sharing mode
    NULL,                                  // security attributes (ignored)
    OPEN_EXISTING,                        // creation disposition
    FILE_FLAG_RANDOM_ACCESS,              // flags/attributes
    NULL);                                 // template file (ignored)
```

### 读取模拟输入通道数据

```
DAQ_INFO DaqInfo;
DWORD    dwNumberOfBytesRead = 0;
BOOL     bRet;
float    f1;
char     Unit[16];
```

```
DaqInfo.dwCmd = EM9280_DAQ_VOLTAGE_CH0; //读取AIN1通道数据
dwNumberOfBytesRead = 0;
bRet = ReadFile(hLRADC, &DaqInfo, sizeof(DAQ_INFO),
                &dwNumberOfBytesRead, NULL);
if(!bRet)
{
    //出错处理
    //... ..
}
sscanf(DaqInfo.UnitName, "%f%s", &f1, Unit);
printf("AIN1 = %d(%s) -> %.1f%s\n", DaqInfo.dwData, DaqInfo.UnitName,
       (DaqInfo.dwData * f1), Unit);
```

在读取 AIN1 和 AIN2 通道的数据时，返回的 dwData 仅仅是 AD 的量化数据，在返回的单位字符串中，包括了量化电压间隔 f1 和单位，所以需要用 sscanf 解析之。实际测得的输入电压则为 DaqInfo.dwData \* f1。

#### 读取板卡环境温度数据

```
DaqInfo.dwCmd = EM9280_DAQ_BOARD_TEMPERATURE; //读取板卡温度
dwNumberOfBytesRead = 0;
bRet = ReadFile(hLRADC, &DaqInfo, sizeof(DAQ_INFO),
                &dwNumberOfBytesRead, NULL);
```

返回的数据 dwData 为开氏温度，大致减去 273 即为摄氏温度。EM9283 是利用三极管的 PN 结的温度特性进行测温的。由于器件的离散性，精度不是很高，一般误差在±3 度。在实际应用中，一般只有当环境温度很高时（比如 45℃或以上），才有温度检测的意义。因此该功能对设备运行于高温环境是有积极意义的。

## 矩阵键盘接口

EM9283 矩阵键盘驱动最大可支持 5x5，即 25 个按键。客户也可根据自身应用需要，通过注册表设置键盘格式为：KEYIN3 – KEYOUT3，KEYIN4 – KEYOUT3，KEYIN4 – KEYOUT4 等。具体的做法是修改 EM9283 注册表键

[HKEY\_LOCAL\_MACHINE\Drivers\BuiltIn\KPD1]下的"MKKeyPadFormat"的值。

MKeyPadFormat 值	简要描述
0	(KEYIN0~KEYIN2, KEYOUT0~KEYOUT2)
1	支持 4x3 (KEYIN0~KEYIN3, KEYOUT0~KEYOUT2)
2	支持 4x4 (KEYIN0~KEYIN3, KEYOUT0~KEYOUT3)
3	手持设备开发评估套件缺省配置 支持 5x4 (KEYIN0~KEYIN4, KEYOUT0~KEYOUT3)
4	通用开发评估套件缺省配置 支持 5x5 (KEYIN0~KEYIN4, KEYOUT0~KEYOUT4)

表 6.1 EM9283 矩阵键盘格式

系统启动时会自动加载矩阵键盘驱动程序，进行键盘扫描，当用户按下某个按键时，驱动程序会产生标准的 Windows 按键消息：“KeyDown”，“KeyUp”和对应的字符消息。针对 EM9283 不同的开发评估套件，各自缺省的虚拟键码如表 6.2 和表 6.3:

	KEYIN0	KEYIN1	KEYIN2	KEYIN3	KEYIN4
KEYOUT0	VK_RETURN	VK_DECIMAL	VK_0	VK_*	VK_BACK
KEYOUT1	VK_UP	VK_9	VK_8	VK_7	VK_RIGHT
KEYOUT2	VK_ESCAPE	VK_6	VK_5	VK_4	VK_DOWN
KEYOUT3	VK_TAB	VK_3	VK_2	VK_1	VK_LEFT
KEYOUT4	VK_SHIFT	VK_ADD	VK_SUBTRACT	VK_MULTIPLY	VK_DIVIDE

表 6.2 EM9283 手持设备开发评估虚拟键码

	KEYIN0	KEYIN1	KEYIN2	KEYIN3	KEYIN4
KEYOUT0	VK_ESCAPE	VK_0	VK_PERIOD	VK_BACK	VK_0
KEYOUT1	VK_ADD	VK_1	VK_2	VK_3	VK_0
KEYOUT2	VK_SUBTRACT	VK_4	VK_5	VK_6	VK_0
KEYOUT3	VK_MULTIPLY	VK_7	VK_8	VK_9	VK_0
KEYOUT4	VK_DIVIDE	VK_SPACE	VK_DECIMAL	VK_RETURN	VK_0

表 6.3 EM9283 通用评估虚拟键码

根据应用的需要，用户可通过修改注册表，重定义各个按键的虚拟键码，与虚拟键码对应的 16 进制值可以在 MSDN 上找到：

[http://msdn.microsoft.com/zh-cn/library/ms927178\(en-us\).aspx](http://msdn.microsoft.com/zh-cn/library/ms927178(en-us).aspx)

EM9283 矩阵键盘的 KEYIN3、KEYIN4 和 KEYOUT3、KEYOUT4 是与 GPIO 复用的，当没有使能这些位的键盘功能时，可作为独立的 GPIO 使用。

### GPIO 通用数字 IO

EM9283 的 32 位 GPIO0 – GPIO31 均为可独立方向可设置的通用数字 IO，所有 GPIO 的上电初始状态均为输入状态，部份 GPIO 带上拉电阻(见表 5.2)。EM9283 为应用程序提供了操作 GPIO 的驱动程序，其设备文件名为“PIO1:”。有关 GPIO 操作的使用方法在相应的范例代码中有详细的中文说明，这里不再赘述。EM9283 为了保持模块的紧凑尺寸及机械强度，其 GPIO 与主板的其它接口功能采用了管脚复用的设计，具体复用情况如下表所示：

管脚#	复用信号	复用功能简要说明
GPIO0	CTS2#	COM2 口的硬件流控信号，与 RTS 配对使用。
GPIO1	RTS2#	COM2 口的硬件流控信号，与 CTS 配对使用。
GPIO2	DSR2#	COM2 口的 Modem 控制信号
GPIO3	DTR2#	COM2 口的 Modem 控制信号
GPIO4	RI2#	COM2 口的 Modem 控制信号
GPIO5	DCD2#	COM2 口的 Modem 控制信号
GPIO6	RXD5	COM5 口数据输入
GPIO7	TXD5	COM5 口数据输出
GPIO8	RXD6	COM6 口数据输入

GPIO9	TXD6	COM6 口数据输出
GPIO10	I2C_SCL	I2C 总线时钟信号，平时为三态，要求上拉。
GPIO11	I2C_SDA	I2C 总线双向数据信号，平时为三态，要求上拉。
GPIO12	SPI_MISO	SPI 接口数据输入，Master 模式
GPIO13	SPI_MOSI	SPI 接口数据输出，Master 模式
GPIO14	SPI_SCLK	SPI 接口时钟输出，极性、相位可设置。
GPIO15	SPI_CS0N	SPI 接口片选控制输出，低电平有效。
GPIO16	KEYIN3	矩阵键盘状态输入，要求加上拉电阻。
GPIO17	KEYOUT3	矩阵键盘扫描输出，低电平有效。
GPIO18	KEYIN4	矩阵键盘状态输入，要求加上拉电阻。
GPIO19	KEYOUT4	矩阵键盘扫描输出，低电平有效。
GPIO20	SD_DET	SD 卡插入状态检测位，默认上升电平插入。
GPIO21	SD_PWR	SD 卡电源管理引脚
GPIO22	SD_CMD	SD 卡接口控制信号 SD_CMD。
GPIO23	SD_CLK	SD 卡接口控制信号 SD_CLK。
GPIO24	SD_D0	SD 卡接口双向数据信号 SD_D0。
GPIO25	SD_D1	SD 卡接口双向数据信号 SD_D1。
GPIO26	SD_D2	SD 卡接口双向数据信号 SD_D2。
GPIO27	SD_D3	SD 卡接口双向数据信号 SD_D3。
GPIO28	PWM1	脉宽调制输出。
GPIO29	PWM2	脉宽调制输出。
GPIO30	PWM3	脉宽调制输出。
GPIO31	PWM4	脉宽调制输出。

在系统启动后的初始状态，所有的 GPIO 都是有效的，一旦应用程序打开某个接口的设备文件，则对应的 GPIO 功能将被禁止。注意即使应用程序关闭了设备文件，对应的 GPIO 功能同样是被禁止的。因为在嵌入式系统中，不可能存在一条管脚动态复用的情况。

对于没有上位电阻的 GPIO 位，如果需要在系统上电后有明确的电平状态，需要用户在设计外围电路时，连接合适的上位或下拉电阻。

## 7、版本历史

手册版本	适用主板	简要描述	日期
V2.0	EM9283 V2.0	EM9283 工控主板数据手册	2012-11