

[ES-7860_111018-CH]

AU7860 SDK 使用说明

V0.1

目录

| | |
|------------------------------|----|
| AU7860 SDK 概述 | 1 |
| 1. AU7860 的特点及系统功能 | 1 |
| 2. AU7860 软件的层次 | 2 |
| 3. AU7860 SDK 包 | 3 |
| 4. AU7860 软件主流程 | 4 |
| GPIO 应用说明 | 7 |
| 1. GPIO 复用关系 | 7 |
| 2. 普通 GPIO 的使用 | 8 |
| 3. 上下拉电阻的配置及特点 | 8 |
| 4. 特殊应用 | 9 |
| ADC 应用说明 | 10 |
| 1. 如何设置 GPIO 口为 ADC 通道 | 10 |
| 2. 读取指定通道的 ADC 值 | 10 |
| 3. 芯片电源输入端电压采样 | 10 |
| 按键与红外应用说明 | 11 |
| 1. ADC 按键 | 11 |
| 2. 红外输入 | 11 |
| 段码 LCD 接口应用说明 | 12 |
| 1. LCD 接口使用流程 | 12 |
| 2. 应用实例 | 12 |
| RTC 应用说明 | 13 |
| 1. 时钟和闹钟配置 | 15 |
| PWM 应用说明 | 16 |
| 1. PWM 接口使用方法 | 16 |
| 2. 注意事项 | 16 |

AU7860 SDK 概述

1. AU7860 的特点及系统功能

AU7860 系列是山景开发的多应用、高性能音频 SOC 芯片，片上系统基于高性能增强型 51 MCU 运行。山景提供完善的 SDK 套件，方便客户二次开发。

AU7860 SOC 及其系统具有如下特点及功能：

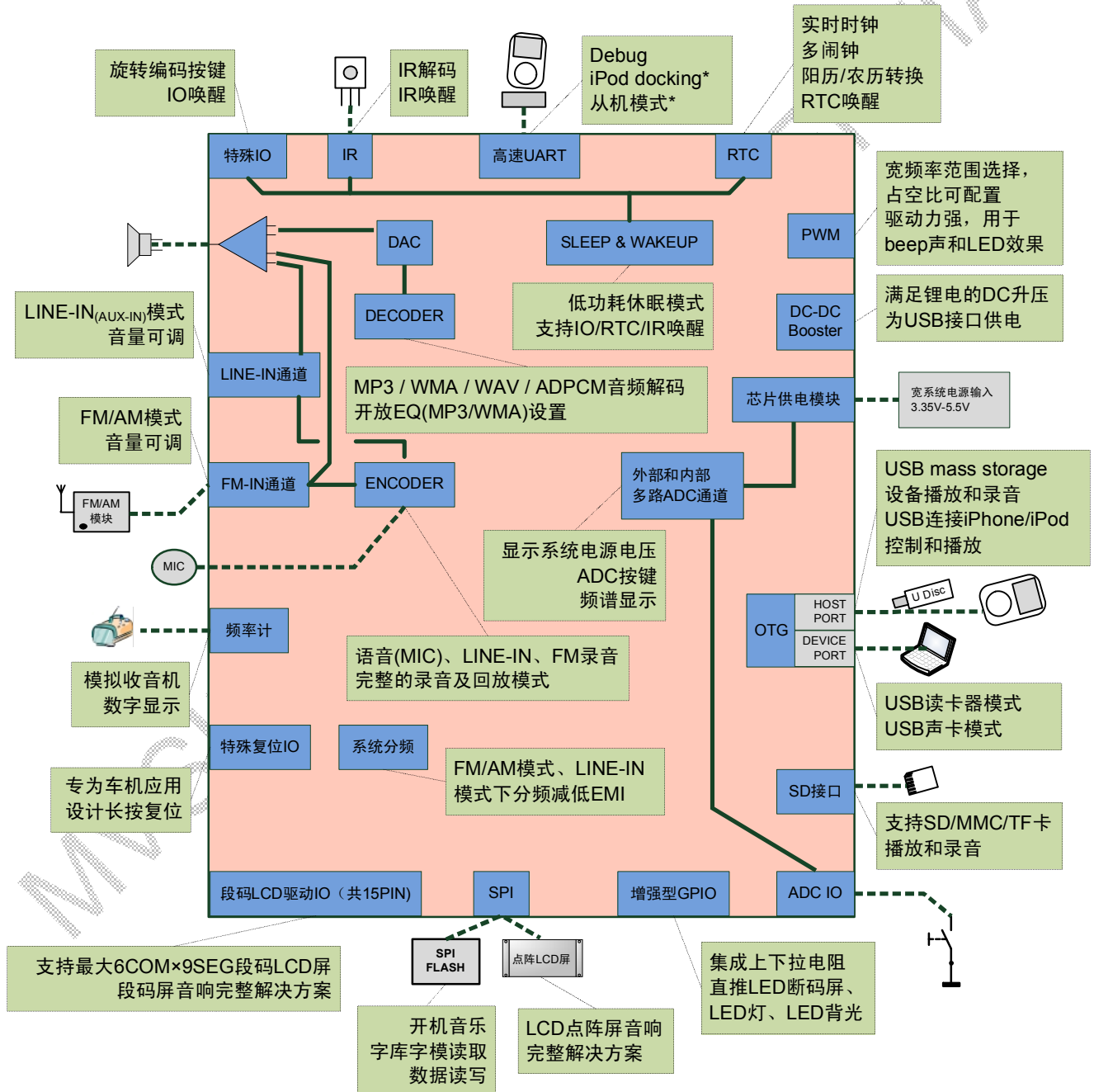


图 1：AU7860 系统的特点和功能



图 2: AU7860 SDK 软件功能

所有功能按逻辑和条理归类在 SDK 不同的文件夹下。

2. AU7860 软件的层次

上层应用代码将相关的底层模块组织在一起，实现高层次的逻辑功能。

上层应用代码主要有：播放流程、设备切换、主函数、FM 控制、录音控制等各种功能模式的控制代码。

上层应用代码全部以源代码的形式提供，便于客户做二次开发。

底层驱动实现对硬件的抽象。SDK 中将客户不需要关注的底层驱动打包成 LIB 库，另外，客户可能会修改的那部分底层驱动则不适合打包成 LIB 库，仍然是以源代码的形式提供，便于修改。

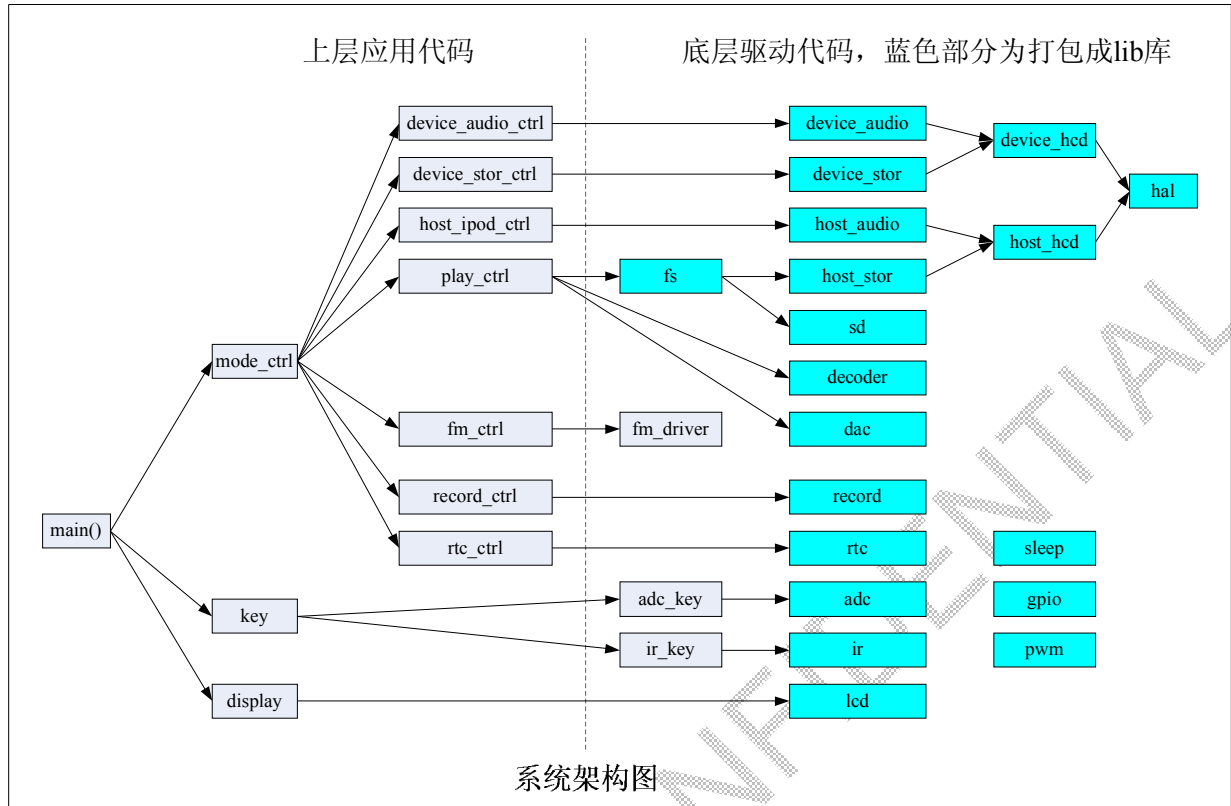


图 3: AU7860 SDK 软件层次

3. AU7860 SDK 包

AU7860 系统软件基本按照功能和层次划分，存放于 SDK 不同的文件夹下，简介如下：

| 文件夹 | 存放的主要代码 | 与用户相关性 |
|---------|--|---------------------------------|
| config | 各种系统实现的功能选择以及和该功能相关的配置，如 IO, buffer | 选择 SDK 具备的功能，定义功能 IO 等 |
| device | 系统模式控制代码和设备切换逻辑相关的代码 | 用户增加系统模式时，需要理解模式增加的方法和设备切换的代码逻辑 |
| display | 显示示例。 | 用户可以扩充自己更丰富的显示 |
| EEPROM | 如果断点记忆功能基于 24C02 E ² PROM 实现，这里存放了 24C02 和 I ² C 的代码 | |
| fs | 文件系统初始化，曲目和文件夹的预搜索，曲目和文件夹浏览 | |
| key | ADC 按键、旋转编码开关、IR 按键获取并解析为各种系统消息 | 用户根据产品按键的类型、数量、码值修改对应的键值表 |
| library | 各硬件模块驱动 | |
| line-in | LINE-IN (AUX-IN)模式控制示例 | |
| otg | USB 读卡器模式、iPhone USB 模式、USB 声卡模式控制示例 | |
| play | 播放控制逻辑，曲目信息、歌词同步显示 | |

| | | |
|--------|--|--|
| | 逻辑 | |
| radio | FM 模式示例 | |
| record | 录音和回放模式示例 | |
| rtc | 时钟和日历模式示例 | |
| system | 系统前台总控制代码含 main()函数，系统后台处理控制，断点记忆功能代码等 | |

表 1: SDK 包文件夹说明

各文件夹下包含的“.c”和“.h”文件，详见 SDK 软件包。

4. AU7860 软件主流程

AU7860 系统软件采用前后台方式构建和运行。

sysctrl.c 文件中的 main()函数实现了系统前台的主要功能模块的大循环，如图 4。

interrupt.c 文件中的几个函数实现了系统后台的几个中断服务功能，如图 5。

系统各功能模块内部多数采用状态机结构，各功能模块间采用消息传递和少量全局变量机制。

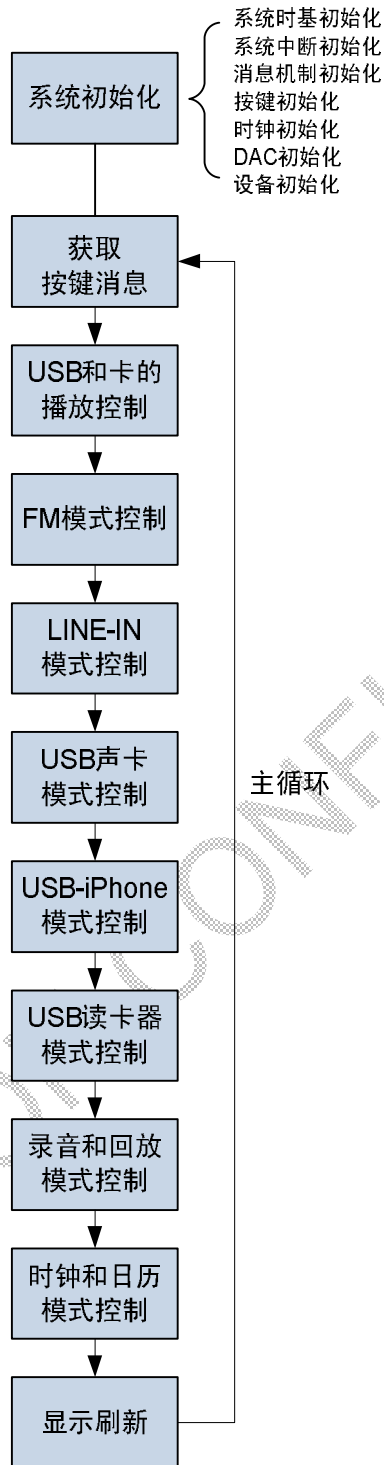


图 4：系统前台主流程

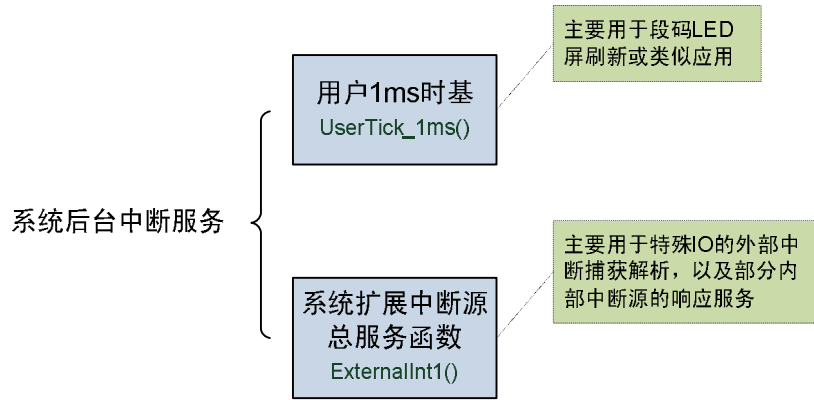


图 5: 系统后台中断服务

系统各模块的使用说明见随后各章节。

GPIO 应用说明

1. GPIO 复用关系

AU7860 共有 5 组 GPIO，除了可以作为普通 GPIO 使用外，还可以复用为其它功能，具体复用关系见下表 2。如何配置 IO 口的功能角色请参考 SDK 中的“gpio.h”文件。
 特别注意，多数 GPIO 口被多种功能复用，必须保证同一时刻一个 GPIO 口只扮演一种功能角色。

| IO 名称 | 普通 GPIO | ADC IO | LCD IO | SD 和 SPI IO | FM-IN IO | PWM IO | 其它功能 IO |
|-----------|---|----------------|--------|-------------|-------------|---------------------------------|--|
| GPIO_A[0] | 普通双向 3.3V IO，上下拉电阻可配置，分强弱 2 档 | ADC_CHANNEL_A0 | | | | | fast_uart_txd(o) |
| GPIO_A[1] | | ADC_CHANNEL_A1 | | | | | fast_uart_rxd(i) |
| GPIO_A[2] | | ADC_CHANNEL_A2 | | | | | |
| GPIO_A[3] | | | | | SD_DAT(io) | | |
| GPIO_A[4] | | | | | SD_CMD(io) | | |
| GPIO_A[5] | | | | | SD_SCK(o) | | |
| GPIO_A[6] | | | | | | FM-IN_1R | |
| GPIO_A[7] | | | | | | FM-IN_1L | |
| GPIO_B[0] | 增强电流驱动力的双向 3.3V IO，上下拉电阻可配置，分强弱 2 档，可直推 LED 段码屏 | | LCD0 | | | | fast_uart_txd(o) B 口与 A 口不能同时作为串口使用 |
| GPIO_B[1] | | | LCD1 | | | | fast_uart_rxd(i) B 口与 A 口不能同时作为串口使用 |
| GPIO_B[2] | | | LCD2 | | | | 8051_uart_txd(o) |
| GPIO_B[3] | | | LCD3 | | | | 8051_uart_rxd(i) |
| GPIO_B[4] | | | LCD4 | | SPI_MISO(i) | | |
| GPIO_B[5] | | | LCD5 | | SPI_SCK(o) | | |
| GPIO_B[6] | | | LCD6 | | SPI_MOSI(o) | | |
| GPIO_B[7] | | | LCD7 | | | | |
| GPIO_C[0] | 带中断触发的普通双向 3.3V IO，上下拉电阻可配置，分强弱 2 档； | | | | FM-IN_2R | PWM_CH_C0 | 可做外部唤醒 IO |
| GPIO_C[1] | | | | | FM-IN_2L | PWM_CH_C1 | 可做外部唤醒 IO |
| GPIO_C[2] | 带上拉，带中断触发，且耐压达 5V 的单纯输入口 | | | | | | 可做外部唤醒 IO |
| GPIO_D[0] | 普通双向 3.3V IO，上下拉电阻可配置，分强弱 2 档 | | LCD8 | | | PWM_CH_D0 不能与 PWM_CH_C0 同时使用 | |

| | | | | | | | |
|-----------|---------------|----------------|-------|--|--|---------------------------------------|--------------|
| GPIO_D[1] | | ADC_CHANNEL_D1 | | | | PWM_CH_D1 不能与 PWM_CH_C1 同时使用 | |
| GPIO_D[2] | | | LCD9 | | | PWM_CH_D2 | |
| GPIO_D[3] | | | LCD10 | | | PWM_CH_D3 | |
| GPIO_D[4] | | | LCD11 | | | | |
| GPIO_D[5] | | | LCD12 | | | | |
| GPIO_D[6] | | | LCD13 | | | | |
| GPIO_D[7] | | | LCD14 | | | | |
| GPIO_E[0] | 普通双向 3.3V IO, | ADC_CHANNEL_E0 | | | | | |
| GPIO_E[1] | 上下拉电阻可配 | ADC_CHANNEL_E1 | | | | | |
| GPIO_E[2] | 置, | ADC_CHANNEL_E2 | | | | | 5S_RST_KEY1 |
| GPIO_E[3] | 分强弱 2 档 | ADC_CHANNEL_E3 | | | | | 5S_RST_KEY2 |
| GPIO_E[4] | | ADC_CHANNEL_E4 | | | | | BOOSTER_CTRL |

表 2: GPIO 复用关系

注释:

FM-IN_R, FM-IN_L, 用于连接 FM 模拟音源 R 和 L 声道。

5S_RST_KEY1, 5S_RST_KEY2, 用于长按 5 秒复位系统。

BOOSTER_CTRL, 用于 DC-DC booster 的 PWM 控制信号输出。

各 IO 的详细连接方式, 请参考系统应用指南和参考设计图纸。

本章简介普通 GPIO 口的特性和注意事项, 作为其它功能 IO 的使用详见各功能章节。

2. 普通 GPIO 的使用

SDK 提供几个 API 函数对 GPIO 进行配置和读写操作, 详见“gpio.h”文件。

注意: 将普通 GPIO 配置为其它功能 IO 之后, 如果需要重新配置为普通 GPIO 使用, 需要调用相应的函数或者配置对应寄存器。

3. 上下拉电阻的配置及特点

内部上下拉电阻设置的真值表如下:

| GPIO_X_PU | GPIO_X_PD | 功能 |
|-----------|-----------|-----------------|
| 0 | 0 | 作为数字信号接口 (上拉有效) |
| 1 | 1 | 作为数字信号接口 (下拉有效) |
| 1 | 0 | 作为数字信号接口 (无上下拉) |

注: X 可以是 A/B/C/D/E 口。

AU7860 芯片, GPIO 上的电平 $\leq 1.40V$ 判断为逻辑 0, $\geq 1.60V$ 判断为逻辑 1, 设逻辑阈值 $V_{th} \approx 1.5V$, 针对于该 V_{th} 值:

弱上拉/下拉阻值 (约): 95K

强上拉/下拉阻值 (约): 28K

强、弱两档通过 **GPIO_X_DS** 寄存器配置，相应位为 1 时为强档。X 可以是 A/B/C/D/E 口。

4. 特殊应用

使用 GPIO 捕获外部信号并产生中断。
直接操作 GPIO 的控制寄存器。

ADC 应用说明

AU7860 集成 12 位精度的 ADC 硬件单元，支持共 12 路内部和外部模拟通道采样，包括：

- 9 路外部 IO 端口信号采样，详见 GPIO 复用关系表。
- 3 路内部信号采样。含，音频输出左右声道（ADC_CHANNEL_AUDIO_L 和 ADC_CHANNEL_AUDIO_R），常用于简单音频频谱显示；1 路芯片电源输入端电压采样，用于检测显示系统电源电压。

1. 如何设置 GPIO 口为 ADC 通道

使用 GPIO 口作为 ADC 通道时，GPIO 的输出控制寄存器使能位为 0，上拉电阻控制位为 0，下拉电阻控制位为 1，对应的 GPIO 口才能正确进入 ADC 模式。比如，设置 GPIO_A1 为 ADC 通道，需要使用以下代码：

```
ClrGpioRegBit (GPIO_A_OE, 1<<1); //设置输出使能位为 0
ClrGpioRegBit (GPIO_A_PU, 1<<1); //设置上拉电阻控制位为 0
SetGpioRegBit (GPIO_A_PD, 1<<1); //设置下拉电阻控制位为 1
```

3 路内部信号采样无需进行上述设置。

2. 读取指定通道的 ADC 值

函数接口请参考 SDK 中的“adc.h”。

一个 API 函数对指定通道进行 ADC 采样，输出 0~4095 的采样值。当然，输出值与参考电压源的选择也有关系，参考电压有 2 种，ADC_VREF_BG24 和 ADC_VREF_AVDD33，它们的区别如下：

| 宏定义 | 参考电压 | 支持的输入电压范围 | 理论输出采样值 |
|-----------------|-------------|-----------|------------------------|
| ADC_VREF_BG24 | 内部 2.4V 稳压源 | 0~2.4V | $(V_{in}/2.4)*4096$ |
| ADC_VREF_AVDD33 | AVDD33 | 0~AVDD33 | $(V_{in}/AVDD33)*4096$ |

使用何种参考电压取决于具体应用。

3. 芯片电源输入端电压采样

电源输入端 LDOIN 可检测的电压范围从 3.3V 至 4.6V。在“adc.h”中有 API 函数，返回该电压值，单位是毫伏。返回值范围为 3300~4600。此函数建议 1 秒以上调用一次，频繁调用会增加芯片功耗。

按键与红外应用说明

7860 SDK 中实现的按键驱动有 ADC 按键、IR 按键和旋转编码开关。

SDK 为不同的按键输入提供统一的接口，请参考“key.h”。

在 3 种按键功能都工作情况下，SDK 的默认逻辑查询顺序是：ADC 按键 → 旋转编码开关 → IR 按键。

1. ADC 按键

SDK 中的 ADC 驱动，即“adc_key.c”针对 AU7860 开发板系统的 ADC 按键电路设计。

按键动作与事件的对应关系定义在“adc_key.c”中的一个 2 维数组内。

用户根据实际产品电路调整消抖、按键与 ADC 值取值范围等，以及调整按键动作与消息事件的映射关系。

2. 红外输入

SDK 中的 IR 驱动，即“ir_key.c”针对 AU7860 开发板系统的 IR 遥控器及电路设计，IR 接收头连接到 GPIO_C[2] 口，兼容 5V 和 3.3V 两种电压的 IR 接收头。

IR 遥控器的键值定义在“adc_key.c”中的一个 1 维数组内，厂商编号由宏定义。

按键动作与消息事件的对应关系定义在“adc_key.c”中的一个 2 维数组内。

用户根据实际使用的 IR 遥控器，修改 IR 键值表和厂商编号宏定义。

段码 LCD 接口应用说明

AU7860 提供 15 个段码 LCD IO 口，具体见 GPIO 复用关系章节。

支持：

- 1/2 偏压、1/3 偏压、1/4 偏压
- 8 种 LCD 工作电压选择，包括 2.6V，2.7V，2.8V，2.9V，3.0V，3.1V，3.2V，3.3V。

另外，AU7860 支持休眠模式下刷屏。

1. LCD 接口使用流程

流程简图：

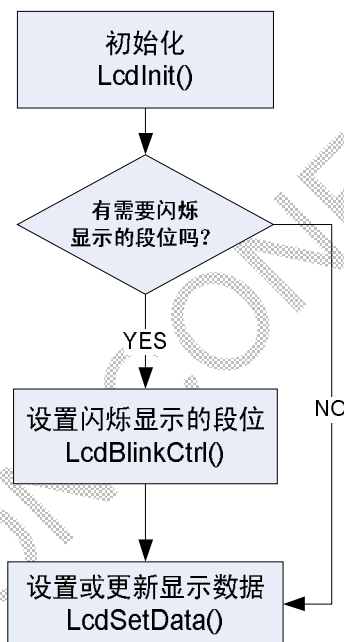
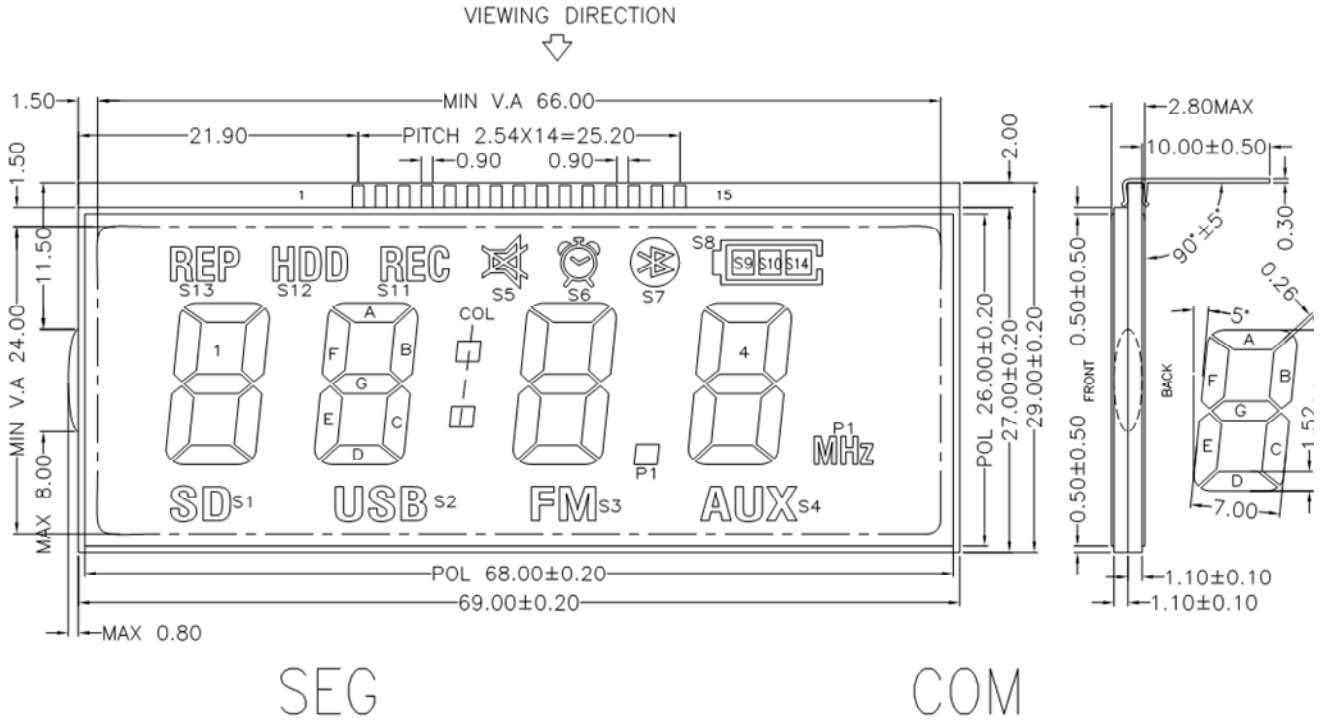


图 6: LCD 接口使用流程

API 函数及其使用方法，请参考 SDK 中的“lcdseg.h”文件。

2. 应用实例

假如使用的 LCD 屏的资料如下：



| PIN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------|------|------|------|------|----|----|----|----|----|----|----|----|-----|-----|-----|
| COM1 | COM1 | | | | 1F | 1A | 2F | 2A | 3F | 3A | 4F | 4A | S8 | COL | S11 |
| COM2 | | COM2 | | | 1G | 1B | 2G | 2B | 3G | 3B | 4G | 4B | S9 | S5 | S12 |
| COM3 | | | COM3 | | 1E | 1C | 2E | 2C | 3E | 3C | 4E | 4C | S10 | S6 | S13 |
| COM4 | | | | COM4 | S1 | 1D | S2 | 2D | S3 | 3D | P1 | 4D | S4 | S7 | S14 |

● 初始化

该液晶屏 4 个 COM 口，9 个 SEG 口，偏压为 1/3，工作电压为 3.3V。假设液晶屏的管脚和芯片的管脚连接关系如下：

| PIN1(COM1) | PIN2(COM2) | PIN3(COM3) | PIN4(COM4) | PIN5(SEG1) | PIN6(SEG2) | PIN7(SEG3) | PIN8(SEG4) |
|------------|------------|------------|------------|------------|------------|------------|------------|
| LCD1 | LCD2 | LCD3 | LCD4 | LCD5 | LCD6 | LCD7 | LCD8 |

| PIN9(SEG5) | PIN10(SEG6) | PIN11(SEG7) | PIN12(SEG8) | PIN13(SEG9) | PIN14(SEG10) | PIN15(SEG11) |
|------------|-------------|-------------|-------------|-------------|--------------|--------------|
| LCD9 | LCD10 | LCD11 | LCD12 | LCD13 | LCD14 | LCD15 |

那么初始化 LCD 显示模块：LcdInit (0x3fe0, 0x001e, LCD_BIAS_3, VLCD_3V3)。

● 更新显示缓存

由于使用 9 个 SEG 口，每个 COM 需要占用 2 个 byte 的 Buf，但只有 bit0~bit8 有效。

举例：如果显示“1234”和“USB”，电池显示满格，那么各个段位的点亮情况如下（1 为亮，0 为灭）：

| PIN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------|------|------|------|------|---|---|---|---|---|----|----|----|----|----|----|
| COM1 | | | | COM1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| COM2 | | | COM2 | | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| COM3 | | COM3 | | | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| COM4 | COM4 | | | | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

对应的 Buf 数据应该为：

COM1 对应 Buf[0:1]: 0x0168

COM2 对应 Buf[2:3]: 0x01FE

COM3 对应 Buf[4:5]: 0x01A6

COM4 对应 Buf[6:7]: 0x042C

因此，用如下代码：

```
BYTE Buf[8] = {0x01, 0x68, 0x01, 0xFE, 0x01, 0xA6, 0x04, 0x2C};  
LcdSetData(Buf);
```


RTC 应用说明

AU7860 集成 RTC 硬件单元，RTC 能够为系统提供以下功能：

- 实时时钟
- 最多 8 个闹钟
- 休眠模式下的闹钟唤醒

1. 时钟和闹钟配置

针对不同的应用需求，系统提供 2 套 API 函数，详见“rtc.h”文件。

应用 1：需要年、月、日、时、分、秒、星期设置和读取，闹钟模式支持每天、每周和单次 3 种的应用；

应用 2：仅需要时、分、秒设置和读取，闹钟模式仅支持每天和单次 2 种的应用。

关于闹钟标志获取、闹钟清除和关闭的 API 函数，也详见“rtc.h”文件。

注意：针对应用 1 有效的的时间设定范围为 1980 年 1 月 1 日 0 时 0 分 0 秒到 2099 年 12 月 31 日 23 时 59 分 59 秒。

PWM 应用说明

AU7860 支持 4 路 PWM 通道的分别设置，并能同时输出，可应用于触摸屏时钟，蜂鸣器，LED 呼吸效果等。PWM 通道与 GPIO 的复用关系，详见 GPIO 复用关系章节。

1. PWM 接口使用方法

具体 API 函数及使用方法，请参考“pwm.h”。

2. 注意事项

AU7860 的 PWM 的输出频率不受系统分频的影响。