版本: **1.0.1　2015** 年 **03** 月

Sword37C 平台

驱动库接口使用说明

© 2015 Anyka Technologies Corporation　All rights reserved.

声　　明

本手册的版权归安凯技术公司所有，受相关法律法规的保护。未经安凯技术公司的事先书面许可，任何人不得复制、传播本手册的内容。

本手册所涉及的知识产权归属安凯技术公司所有（或经合作商授权许可使用），任何人不得侵犯。

本手册不对包括但不限于下列事项担保：适销性、特殊用途的适用性；实施该用途不会侵害第三方的知识产权等权利。

安凯技术公司不对由使用本手册或执行本手册内容而带来的任何损害负责。

本手册是按当前的状态提供参考，随附产品或本书内容如有更改，恕不另行通知。

联 系 方 式

安凯（广州）微电子技术有限公司

地址：广州科学城科学大道 182 号创新大厦 C1 区 3 楼
电话: (86)-20-3221 9000

传真: (86)-20-3221 9258

邮编: 510663

销售热线:

(86)-20-3221 9499

电子邮箱:

sales@anyka.com

主页:

http: //www. anyka.com

Anyka Confidential For BOMEI Use Only

版 本 变 更 说 明

以下表格对于本文档的版本变更做一个简要的说明。版本变更仅限于技术内容的变更，不包括版式、格式、句法等的变更。

版本	说明	完成日期
V1.0.0	正式发布	2014 年 08 月
V1.0.1	1、LCD 控制增加 3 个控制函数，解决多通道显示闪屏问题 2、增加软重启函数 3、增加电压校准函数 4、增加驱动注意事项 5、修改参数描述的变化	2015 年 03 月

Anyka Confidential For BOMEI Use Only

目录

Anyka Confidential For BOMEI Use Only

Anyka Confidential For BOMEI Use Only

Anyka Confidential For BOMEI Use Only

1 简介

本文档为 Sword37C 系统应用平台的驱动库文档，主要包含“各个驱动模块的接口使用说明”和“常用驱动的修改方法”两部分。Sword37C 系统应用平台所包含的驱动模块有：Analog Control、Frequency、GUI、Init、LCD、MMC_SD_SDIO、MMU、RTC、UART、Camera、GPIO、I2C、Keypad、Timer、Sound 和 SPI Flash 等。

下面将逐个介绍各驱动的接口使用说明。

2 驱动库接口使用说明

2.1 Analog Control

2.1.1 功能概述

Analog Control 即模拟控制模块，主要用于控制音频的输入输出、声道控制以及输入输出的信号连接等。定义如下：

Enum ANALOG_CHANNEL

定义单通道或者双通道设置

Enumerator:

ANALOG_HP if chose mono, use left channel //若选择单通道则使用左通道

ANALOG_LINEOUT if chose mono, use left channel//若选择单通道则使用左通道

ANALOG_LINEIN if chose mono, use left channel//若选择单通道则使用左通道

ANALOG_MIC only one channel MIC//只有一个通道

Enum ANALOG_CHANNEL_STATE

定义通道状态：单通道或者双通道

Enumerator:

CHANNEL_MONO one channel(use left channel) //单通道（左通道）

CHANNEL_STEREO two channel. //双通道

Anyka Confidential For ROMEI Use Only

Enum ANALOG_SIGNAL_INPUT

定义音频输入模块

Enumerator:

INPUT_DAC = 1 //选择 DAC 输入音频

INPUT_LINEIN = 2 //选择 LINEIN 输入音频

INPUT_MIC = 4 //选择 MIC 输入音频

INPUT_ALL = 7, //选择以上三个模块同时输入音频

Enum ANALOG_SIGNAL_OUTPUT

定义音频输出模块

Enumerator:

OUTPUT_ADC = 1 //选择 ADC 输出音频

OUTPUT_HP = 2 //选择 HP 输出音频

OUTPUT_LINEOUT = 4 //选择 LINEOUT 输出音频

OUTPUT_ALL = 7, //选择以上三个模块同时输出音频

Enum ANALOG_SIGNAL_STATE

定义输入源和输出源之间的信号状态：连接或断开

Enumerator:

SIGNAL_CONNECT open one or all input signal to one or all output signal//连接输入输出信号

SIGNAL_DISCONNECT close input signal to output signal//断开输入输出信号

Enum ANALOG_VOLTAGE_MODE

定义参考电压类型设置

set mode of reference voltage generator

Enumerator:

MODE_AC AC mode. //交流模式

MODE_DC DC mode. //直流模式

Anyka Confidential For BOMEI Use Only

2.1.2 接口说明

本节将介绍 Analog Control 的各个接口说明。

T_BOOL analog_getchannel (ANALOG_CHANNEL module,
ANALOG_CHANNEL_STATE * state
)

获取信号通道状态（单通道或者双通道）

Get signal channel state, MONO or STEREO

参数说明:

[in] module 指 ANALOG_CHANNEL，详见 *ANALOG_CHANNEL* 定义

[out] state CHANNEL_MONO 或者 CHANNEL_STEREO

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

T_BOOL analog_getsignal (ANALOG_SIGNAL_INPUT analog_in,
ANALOG_SIGNAL_OUTPUT analog_out,
ANALOG_SIGNAL_STATE * state
)

获取输入源输出源之间的信号连接状态

get the signal connection state between input and output source

参数说明:

[in] analog_in ANALOG_SIGNAL_INPUT //参见定义

[in] analog_out ANALOG_SIGNAL_OUTPUT// 参见定义

[out] state SIGNAL_OPEN or SIGNAL_CLOSE// 参见定义

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

ANYKA Confidential For BOM Use Only

T_U32 analog_getvalue_ad5 (T_VOID)

获取 ADC1 AD5 的值，若输入电压为 0~AVDD，则返回值范围为 0~1023。

Get adc1 ad5 value. if input voltage from 0 to AVDD, it will return the value from 0 to 1023

返回值说明:

T_U32

T_U32 analog_getvalue_bat (T_VOID)

获取 ADC1 AD4 的值，若输入电压为 0~AVDD，则返回值范围为 0~1023。

get adc1 ad4 value. if input voltage from 0 to AVDD, it will return the value from 0 to 1023

返回值说明:

T_U32

T_BOOL analog_setchannel (ANALOG_CHANNEL module,
ANALOG_CHANNEL_STATE state
)

设置模拟模块通道为单通道或者双通道

set analog module channel to be MONO or STEREO

参数说明:

[in] module refer to ANALOG_CHANNEL

[in] state CHANNEL_MONO 或者 CHANNEL_STEREO

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

T_BOOL analog_setconnect (ANALOG_SIGNAL_INPUT analog_in,
ANALOG_SIGNAL_OUTPUT analog_out,
ANALOG_SIGNAL_STATE state
)

输入输出信号状态（连接或者断开）

Connect or disconnect the signal between input and output signal.

Anyka Confidential For BOMEI Use Only

参数说明:

[in] analog_in refer to ANALOG_SIGNAL_INPUT

[in] analog_out refer to ANALOG_SIGNAL_OUTPUT

[in] state SIGNAL_OPEN 或者 SIGNAL_CLOSE

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

T_BOOL analog_setgain_hp (T_U8 gain)

设置耳机增益

Set headphone gain.

参数说明:

[in] gain 对于 AK37XX 增益范围为 0~5, 5 为最大增益

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

T_BOOL analog_setgain_linein (T_U8 gain)

设置 LINEIN 增益

Set line-in gain.

参数说明:

[in] gain 增益范围为 0~3,1 为 0db

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

Anyka Confidential For BOMEI Use Only

T_BOOL analog_setgain_mic (T_U8 gain)

设置 MIC 增益

Set microphone gain.

参数说明:

[in] gain 增益范围为 0~7。

返回值说明:

T_BOOL

AK_TRUE 操作成功

AK_FALSE 操作失败

T_BOOL analog_setmode_voltage (ANALOG_VOLTAGE_MODE mode)

设置 DAC 模拟电压类型（交流和直流可选）

Set the mode of DAC analog voltage, it can be set to AC mode or DC mode

参数说明:

[in] mode 必须为 MODE_AC 或者 MODE_DC

返回值说明:

T_BOOL

T_BOOL analog_setsignal (ANALOG_SIGNAL_INPUT analog_in,
ANALOG_SIGNAL_OUTPUT analog_out,
ANALOG_SIGNAL_STATE state
)

连接或者断开输入输出信号

Connect or disconnect the signal between input and output signal.

参数说明:

[in] analog_in 指 ANALOG_SIGNAL_INPUT

[in] analog_out 指 ANALOG_SIGNAL_OUTPUT

[in] state 状态为 SIGNAL_OPEN 或者 SIGNAL_CLOSE

返回值说明:

T_BOOL

ANYKA Confidential For Internal Use Only

AK_TRUE 操作成功

AK_FALSE 操作失败

2.1.3 应用说明

2.1.3.1 HP检测

HP 是通过 ADC2 进行检测，通过读 A/D 的值的变化来判断这些设备的状态。

2.1.3.2 电池电量检测

AK37C 通过 wakeup 管理电源，电压检测则是通过 ADC1 来检测当前电池的电压值，可调用驱动库的接口 *analog_getvalue_bat*（）获得。

2.2 Codec

2.2.1 功能概述

Codec 编解码主要负责 JPEG 的编解码工作，其提供的接口包括编解码中断的使能和关闭等功能。

2.2.2 接口说明

T_VOID codec_intr_disable ()

关闭 JPEG 编解码中断

Disable jpeg codec interrupt

返回值说明：

T_VOID

T_BOOL codec_intr_enable ()

使能 JPEG 编解码中断

Enable jpeg codec interrupt

返回值说明：

T_BOOL

Anyka Confidential For BOMEI Use Only

T_BOOL codec_wait_finish ()

等待 JPEG 编解码完成

Wait jpeg codec finish

返回值说明:

T_VOID

2.3 Frequency

2.3.1 功能概述

Frequency，即频率管理。负责根据各应用的需求，协调设置一个系统合适的工作频率。应用只需申请自己需要工作的频率，然后由此模块协调管理，调用驱动库的频率设置接口，设置最合适的工作频率。

2.3.2 接口说明

T_U32 get_asic_freq (T_VOID)

获取当前 ASIC 的频率

Get current ASIC frequency.

返回值说明:

T_U32 the frequency of ASIC running //当前正在运行 ASIC 的频率

T_U32 get_chip_mainclk (T_U32 pll_value)

获取芯片的主频（Hz）。主频由 PLL 寄存器控制。

Get chip main clock (Hz).

Main clock is controlled by PLL register.

参数说明:

pll_value [in] PLL register value //PLL 寄存器的值

返回值说明:

T_U32 the main clock frequency = pll_value * 1000000L

T_BOOL get_cpu_2x_asic (T_VOID)

判断 CPU 频率是否为 ASIC 频率的两倍

Anyka Confidential For BOMEI Use Only

Judge whether CPU frequency is twice of ASIC frequency or not

返回值说明:

T_BOOL

AK_TRUE cpu frequency is twice of asic frequency //是

AK_FALSE cpu frequency is not twice of asic frequency //否

T_U32 get_cpu_freq (T_VOID)

获取 CPU 频率，其可能与 ASIC 频率相同或者是 ASIC 频率的两倍或三倍。

Get CPU frequency.

CPU frequency can the same as ASIC frequency or 2X/3X of that

返回值说明:

T_U32 the CPU frequency

T_U32 get_pll_value (T_VOID)

获取 PLL 寄存器的值，主频由 PLL 寄存器控制。

Get PLL register value.

Main clock is controlled by pll register.

返回值说明:

T_U32 the frequency of PLL (Mhz) //PLL 频率（Mhz）

T_BOOL set_asic_freq (T_U32 freq)

设置 ASIC 频率

Set ASIC frequency.

参数说明:

freq [in] the freq value to be set, refer to T_ASIC_FREQ definition //需要设定的频率值，参见 T_ASIC_FREQ 定义

返回值说明:

T_BOOL

AK_TRUE set ASIC frequency successful //设置成功

AK_FALSE set ASIC frequency unsuccessful //设置失败

Anyka Confidential For BOMEI Use Only

T_BOOL set_boost_mode (T_U32 pll_value,
T_BOOL enable
)

设置/退出高频模式，ASIC=PLL/2.5

Set or exit boost mode, asic = pll/2.5

参数说明:

[in] pll_value pll frequency //PLL 频率

[in] enable AK_TRUE is to enable boost mode //AK_TURE 为使能高频模式

返回值说明:

T_BOOL

AK_TRUE success to set/exit boost mode // 设置/退出成功

AK_FALSE fail to set/exit boost mode // 设置/退出失败

T_BOOL set_cpu_2x_asic (T_BOOL set_value)

设置是否将 CPU 频率设为 ASIC 频率的两倍，该函数只能将 CPU_CLK 设为 PLL1_CLK 或者 ASIC_CLK。

Set CPU frequency twice of ASIC frequency or not

This function just set cpu_clk = PLL1_clk or set cpu_clk = asic_clk

参数说明:

set_value set twice or not

返回值说明:

T_BOOL

AK_TRUE setting successful // 设置成功

AK_FALSE setting fail // 设置失败

T_BOOL set_cpu_3x_asic (T_U32 pll_value,
T_BOOL set_value
)

设置 CPU_CLK=PLL1_CLK=3*ASIC_CLK=300MHz。

注意事项:

Anyka Confidential For BOMEI Use Only

对于 AK8802 3ES，只有当 CPU 核心电压大于 1.3V 时才使用。在进入 3x config 之前将会存储旧的 PLL1 的值，退出 3x config 时也将会存储旧的 PLL 值。

Set cpu clock = pll1 clock = 3 * asic clock = 300MHz Warning: For AK8802 3ES, and only cpu core voltage higher 1.3V use only. Before entry 3x config, will store old pll1 value and when exit 3x cfg, it will restore old pll1 value.

参数说明:

pll_value [in] pll frequency

set_value AK_TRUE is to enable cpu3x

返回值说明:

T_BOOL

AK_TRUE,: cpu frequency three of asic frequency

AK_FALSE,: when cpu at 2x or 3x ready.

T_U32 set_cpu_low_mode (T_U32 asic,
T_BOOL enable
)

设置/退出 CPU 低频模式，CPU 频率=ASIC 频率。

Set or exit cpu low mode, cpu = asic

参数说明:

[in] asic asic frequency //ASIC 频率

[in] enable AK_TRUE is to enable cpu low mode// AK_TURE 为使能 CPU 低频模式

返回值说明:

T_BOOL

AK_TRUE success to set/exit cpu low mode //设置/退出成功

AK_FALSE fail to set/exit cpu low mode // 设置/退出失败

T_BOOL set_pll_value (T_U32 pll_value)

设置 PLL 寄存器的值，主频由 PLL 寄存器控制。

Set PLL register value.

Main clock is controlled by pll register.

参数说明:

Anyka Confidential For BOMEI Use Only

pll_value [in] pll register value(16-376)(Mhz) //PLL 寄存器的值（16-376）（Mhz）

返回值说明:

T_BOOL

AK_TRUE set pll frequency successful //设置 PLL 频率成功

AK_FALSE set pll frequency unsuccessful //设置 PLL 频率失败

2.4 Frequency Manage

2.4.1 功能概述

本模块是模块 Frequency 的一个补充。Frequency 模块只提供接口给需要变频的地方调用，是实现变频的功能接口，至于是什么时候调用就不关心。而本模块主要关心的是在什么时候需要变频。

本模块的中心思想是利用 CPU 占用率的值，CPU 一直处于最高值，ASIC 实现动态变频，从而不用关心系统处于哪个状态，屏蔽频率管理与模块之间的应用关系。

对于 DMA 等外设的频率管理，本模块提供接口申请与释放。如果有多个 DMA 设备同时申请，那么对应的频率是这些设备申请的频率相加后的结果。

2.4.2 接口说明

T_BOOL FreqMgr_Init(T_U32cpu_max_freq, FreqMgr_Callback CpuCb)

用于变频管理模块的初始化，一定要先调用 FreqMgr_Init。传进的参数 cpu max freq,生成频率节点表，设置初始 PLL、ASIC、CPU 的频率，启动一个定时器，读取 CPU 占用率的值，实现动态变频。参数 **CpuCb** 是返回值为 CPU 占用率的回调，由平台实现。

Frequency manager mode to initialize.

1. Will init ASIC frequency and driver list,
2. Start a timer to check CPU usage factor
3. Set the PLL and default ASIC, CPU 2X

参数说明:

param[in] cpu_max_freq: input the cpu max value,37xx chip the max is 280000000 (hz) //CPU 频率的最大值；

param[in] FreqMgr_Callback: get cpu usage factor call back function. //返回值是 CPU 占用率的回调。

Anyka Confidential For BOMEI Use Only

返回值说明:

T_BOOL

AK_TRUE initial is succeed //是

AK_FALSE initial is fail //否

T_hFreq FreqMgr_RequestFreq(T_U32 ReqFreq)

用于设备的申请最低可运行频率，参数是该设备的最低应用频率。如果是长时间占用申请，请谨慎使用。如果该设备在应用的过程中不能变频，可以用 LOCK/UNLOCK 来实现锁/解锁功能。一定要避免两个设备同时申请同时 LOCK 且申请频率小先申请的情况，因为已经 LOCK，频率大的再申请已经不能再变频。该接口有信号量保护。

Frequency manager mode to request asic min frequency for the specifically driver.

1. If a driver is need a specifically asic frequency for running, this function will be call

2. If call this function,must be call FreqMgr_CancelFreq function to cancel.

3. When the driver is runing and can't to change frequency, we will call FreqMgr_Lock to lock.

Notes: if request asic > current asic, and is lock, will return fail.

The DrvA request the least frequency frist, and then, DrvB request. Then need running the some time.

If the DrvA must be locked, the DrvB request frequency success, but will no to adjust asic frequency.

If DrvA'least frequency less then DrvB,the system may be make a mistake. So , the driver request frequency and need lock, the max value asic must be request first

参数说明:

param[in] ReqFreq: input the driver specifically min frequency for running //设备可运行的最低频率。

返回值说明:

T_hFreq:request frequency handle.//设备申请频率的句柄，用于取消申请的参数。

T_BOOL FreqMgr_CancelFreq(T_hFreq hFreq)

用于设备在已经有申请过频率的情况下取消申请频率。取消申请频率主要的工作是删除设备申请频率表上的信息，如果该设备已经 LOCK，则需要清掉 LOCK 标志。取消成功后，该设备的句柄一定要置为无效的。可以避免重复调用的情况。该接口有信号量保护。

Frequency manager mode to cancel asic min frequency what had requested for the specifically driver.

Anyka Confidential For BOMEI Use Only

1. A driver had requested specifically asic frequency this function will be call by cancle request.

2. If the driver had locked, when cancel, we will call FreqMgr_Lock to UnLock.

参数说明:

param[in] hFreq: input a handle what request freq return the handle //设备申请频率成功时返回的句柄。

返回值说明:

T_BOOL

AK_TRUE frequency cancel is succeed //是

AK_FALSE frequency cancel is fail //否

T_VOID FreqMgr_Lock(T_VOID)

如果某一设备，应用过程中不能变频。可以用以下接口锁住不变频。要求 LOCK 和 UNLOCK 必须配对出现。该接口有信号量保护。

Frequency manager mode to lock asic frequency not to be changed.

1. If a driver when is running and can't to change asic frequency, this function will be call

2. If call this function, must be kept lock and unlock one by one.

参数说明:

T_VOID

返回值说明:

T_VOID

T_VOID FreqMgr_UnLock(T_VOID)

如果一个设备已经 LOCK，那就调用此接口去 UNLOCK。

Frequency manager mode to UnLock asic frequency. must be keep lock and unlock one by one.

参数说明:

T_VOID

返回值说明:

T_VOID

Anyka Confidential For BOMEI Use Only

T_BOOL FreqMgr_IsLock(T_VOID)

取得是否有锁频

Frequency manager mode to get lock status

参数说明:

T_VOID

返回值说明:

T_BOOL

AK_TRUE frequency manage is lock //是

AK_FALSE frequency manage is Unlock //否

2.5 Init

2.5.1 功能概述

Init 初始化程序主要负责对各个库以及各模块的初始化动作，并创建必要的背景线程。支持的芯片类型定义如下：

Enum E_AKCHIP_TYPE

定义所有支持的芯片名称

chip name define all chip supported

Enumerator:

CHIP_8801 AK8801

CHIP_8802 AK8802

CHIP_9801 AK9801

CHIP_9802 AK9802

CHIP_9805 AK9805

CHIP_3771 AK3771

CHIP_3751 AK3751

CHIP_3751B AK3751B

CHIP_3760 AK3760

CHIP_RESERVE reserve

注：AK3753 型号驱动库功能与 AK3771 相同。

AnyKa Confidential For BOMEI Use Only

2.5.2 接口说明

T_BOOL drv_check_series_chip (E_AKCHIP_TYPE chip_type)

检查当前芯片是否为同一系列芯片

Check current chip is the same series or not

参数说明:

[in] chip_type chip type //芯片类型

返回值说明:

T_BOOL

if same series, return AK_TRUE //若为同一个系列，则返回 AK_TURE

T_pVOID drv_free (T_pVOID var)

释放内存

Memory free

参数说明:

var T_pVOID: address of memory to free //释放内存的地址

返回值说明:

void *

E_AKCHIP_TYPE drv_get_chip_type (T_VOID)

获取芯片型号

Get chip type

返回值说明:

T_VOID

T_U32 drv_get_ram_size (T_VOID)

获取 DRAM 容量

Get dram capacity

返回值说明:

T_U32 ram size, uint (byte) // 内存大小单位 BYTE

Anyka Confidential For BOMEI Use Only

void drv_init (T_PDRIVE_INITINFO drv_info)

初始化驱动库。应该在启动步骤时调用，启动中断模块和硬件设备如摄像头、LCD 等。

Driver library initialization

Should be called on start-up step, to initial interrupt module and register hardware as camera, LCD, etc.

返回值说明:

void

T_pVOID drv_malloc (T_U32 size)

内存配置

Memory allocate

参数说明:

size T_U32: size of memory to allocate // 内存配置大小

返回值说明:

void *

2.6 LCD

2.6.1 功能概述

LCD 模块主要负责对 LCD 相关的硬件设备的管理，并封装了将 RGB 和 YUV 等格式的图像数据显现在 LCD 上的接口，便于界面操作。对用户提供的接口主要包括 LCD 的开关、背光的开关、明暗度调整、RGB 显示、YUV 显示、矩形和圆形的绘制、清屏、TVOUT 的相关显示等。

Enum E_TV_OUT_CLK

定义 TV OUT 的时钟类型

TVOUT CLK type define. define the CLK type of TVOUT.

Enumerator:

TVOUT_CLK_INTERNAL Choose internal clk. //选择内部时钟

TVOUT_CLK_EXTERNAL Choose external clk. //选择外部时钟

Anyka Confidential For BOMEI Use Only

Enum E_TV_OUT_TYPE

定义 TV OUT 模式

TVOUT mode type define. define the mode type of TVOUT.

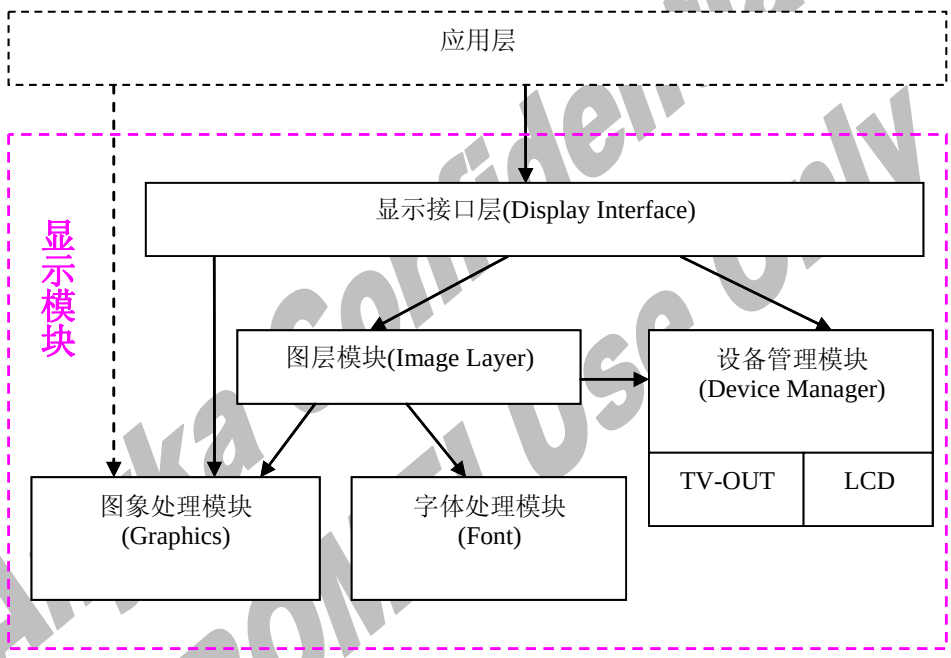
Enumerator:

TV_OUT_TYPE_PAL size: 720*576

TV_OUT_TYPE_NTSC size: 720*480

2.6.2 模块层次说明

LCD 模块运行的整体框图如下图所示：



1、应用层：应用层调用显示接口层提供的接口实现图形图象处理和显示的功能。

2、显示接口层(Display Interface)：为应用层提供图形图象处理、字体处理和显示相关的所有接口，屏幕显示的具体实现和显示设备的类型，并且是平台无关的。

3、图层模块(Image Layer)：维护图层对象的数据结构，对已创建的图层进行管理和维护，同时对图象处理模块和字体处理模块提供的接口进行封装，为显示接口层提供相关接口的具体实现。图层作为一个对象，包含显示空间、显示尺寸、颜色空间类型、透明度等成员，其方法是一组对该对象进行合法操作的函数。

ANYKA Confidential For BOM Use Only

4、图象处理模块(Graphics)：提供绘制图形（点、线、矩形、圆圈等等）、图像；RGB 与 YUV 之间相互转换；对象缩放、旋转等功能的具体实现接口，分 RGB 和 YUV 两种。应用层也可以直接使用这一模块提供的接口。

5、字体处理模块(Font)：提供字体处理，主要是在指定 buffer 显示字串（Unicode 字串、非 Unicode 字串、滚动 Unicode 字串等）的具体实现接口，分 RGB 和 YUV 两种。

6、设备管理模块(Device Manager)：驱动显示设备进行显示，同时对显示设备进行管理。

2.6.3 接口说明

T_eLCD_DEGREE lcd_degree (T_eLCD lcd)

旋转屏的显示角度

Get current LCD degree

参数说明:

[in] LCD selected LCD, must be LCD_0 or LCD_1

返回值说明:

T_eLCD_DEGREE

Degree of LCD //要旋转的角度值

T_U8 lcd_get_brightness (T_eLCD lcd)

获取亮度值

Get brightness value.

参数说明:

[in] LCD selected LCD, must be LCD_0 or LCD_1

返回值说明:

T_U8

current brightness value //当前亮度值

T_U32 lcd_get_hardware_height (T_eLCD lcd)

获取当前 LCD 硬件高度，该高度规格由 LCD 生产厂商确定。

Get current LCD's hardware height LCD hardware height is determined by the LCD factory, and won't be changed

Anyka Confidential For BOMEI Use Only

参数说明:

[in] lcd select the lcd, LCD_0 or LCD_1

返回值说明:

T_U32

LCD hardware height //LCD 硬件高度

T_U32 lcd_get_hardware_width (T_eLCD lcd)

获取当前 LCD 硬件宽度，该宽度规格由 LCD 生产厂商确定。

Get current LCD hardware width LCD hardware width is determined by the LCD factory, and won't be changed

参数说明:

[in] LCD select the LCD, LCD_0 or LCD_1

返回值说明:

T_U32

LCD hardware width //LCD 硬件宽度

E_LCD_TYPE lcd_get_type (T_VOID)

获取当前 LCD 类型

Get current LCD type

返回值说明:

E_LCD_TYPE

type of LCD //LCD 类型，详见 E_LCD_TYPE 定义

T_BOOL lcd_initial (T_VOID)

初始化 LCD

Initialize the LCD.

返回值说明:

T_VOID

Anyka Confidential For BOMEI Use Only

```
T_BOOL lcd_refresh_RGB  (  T_eLCD  lcd,
                           T_RECT *  dsp_rect,
                           T_U8 *  dsp_buf
                              )
```

将 RGB 图像刷新到 LCD 上

Refresh RGB picture to LCD.

参数说明:

[in] lcd selected LCD, must be LCD_0 or LCD_1

[in] dsp_rect display rectangle, source picture should lower than 800*1022 //显示区域的大小

[in] dsp_buf RGB buffer address //RGB 数据 buffer 地址

返回值说明:

T_BOOL

AK_TRUE refresh RGB channel successful //刷新 RGB 通道成功

AK_FALSE refresh RGB channel failed. //刷新 RGB 通道失败

注意事项:

Return failed:

display size bigger than 800*1022 //显示尺寸大于 800*1022

display size smaller than 18*18 //显示尺寸小于 18*18

```
T_BOOL lcd_refresh_RGB_ex  (  T_eLCD  lcd,
                              T_RECT *  dsp_rect,
                              T_RECT *  vir_rect,
                              T_U8 * dsp_buf
                             )
```

使用虚拟页面刷新 RGB 图像

Refresh RGB picture,use virtual page.

参数说明:

[in] lcd selected LCD, must be LCD_0 or LCD_1

[in] dsp_rect display rectangle,display page should lower than 800*1022 dsp_rect->left and top is the start postion of screen dsp_rect->width and height are display width and height //原图片大小

Anyka Confidential For MEI Use Only

[in] vir_rect virtual page rectangle(source page),virture page should lower than 1022*1022 vir_rect->left and top is the start postion of image vir_rect->width and height are image width and height //要显示图片大小

[in] dsp_buf RGB buffer //RGB 数据的 buffer 地址

返回值说明:

T_BOOL

AK_TRUE refresh rgb channel successful //RGB 通道刷新成功

AK_FALSE refresh rgb channel failed //RGB 通道刷新失败

注意事项:

Return failed:

source picture smaller than display size //源图像小于显示尺寸

source picture bigger than 1022*1022¡¢smaller than 18*18//源图像大于 1022*1022 或者小于 18*18

display size smaller than 18*18, bigger than 1022*1022 显示尺寸小于 18*18 或者大于 1022*1022

T_U8 lcd_set_brightness (T_eLCD lcd,
T_U8 brightness
)

设置 LCD 亮度值

Set brightness value.

参数说明:

[in] lcd selected LCD, must be LCD_0 or LCD_1

[in] brightness brightness value,use 0 to 7 //亮度值，取值范围为 0~7

返回值说明:

T_U8

new brightness value after setting //设置后的新亮度值

T_VOID lcd_turn_off (T_eLCD lcd)

关闭 LCD

Turn off the LCD.

Anyka Confidential For BOME Use Only

参数说明:

[in] lcd selected LCD, must be LCD_0 or LCD_1

返回值说明:

T_VOID

T_VOID lcd_turn_on (T_eLCD lcd)

打开 LCD

Turn on the LCD.

参数说明:

[in] lcd selected LCD, must be LCD_0 or LCD_1

返回值说明:

T_VOID

T_VOID lcd_tv_out_close (T_VOID)

关闭 TV OUT 功能

close TV-out function

返回值说明:

T_VOID

T_BOOL lcd_tv_out_open (E_TV_OUT_TYPE type)

打开 TV OUT 功能

Open TV-out function

参数说明:

[in] type TV_OUT_TYPE_PAL or TV_OUT_TYPE_NTSC //TV OUT 模式选择

返回值说明:

T_BOOL

AK_TRUE open TV out successful //打开 TV OUT 成功

AK_FALSE open TV out failed //打开 TV OUT 失败

注意事项:

return failed: type not TV_OUT_TYPE_PAL or TV_OUT_TYPE_NTSC //显示制式的选择

Anyka Confidential For BOMEI Use Only

T_VOID lcd_YUV_off (T_VOID)

关闭 YUV 通道

Close YUV channel

返回值说明:

T_VOID

T_VOID lcd_YUV_on (T_VOID)

打开 YUV 通道

Open YUV channel

返回值说明:

T_VOID

```
T_BOOL lcd_refreshdata_YUV1 (T_eLCD lcd,
                             T_U8 *srcY,
                             T_U8 *srcU,
                             T_U8 *srcV,
                             T_U16 src_width,
                             T_U16 src_height,
                             T_RECT *dsp_rect
                            )
```

准备 YUV 通道的数据

参数说明:

[in] lcd:选择 lcd 号，必须是 LCD_0 或者 LCD_1.

[in] srcY:YUV 数据 Y 的地址

[in] srcU:YUV 数据 U 的地址

[in]srcV:YUV 数据 V 的地址

[in]src_width:YUV 数据的宽度

[in]src_height:YUV 数据的长度

[in]dsp_rect：显示的窗口大小

返回值说明:

AK_TURE:成功

AK_FALSE:失败

T_BOOL lcd_refreshdata_RGB(T_eLCD lcd, T_RECT *dsp_rect, T_U8 *dsp_buf)

准备 RGB 的数据

参数说明：

[in] lcd:选择 lcd 号，必须是 LCD_0 或者 LCD_1.

[in]dsp_rect:显示的窗口 大小

[in]dsp_buf:待显示 RGB 数据的地址

返回值说明：

AK_TURE:成功

AK_FALSE:失败

T_BOOL lcd_refresh_output(T_eLCD lcd)

把准备好的 YUV 数据和 RGB 数据进行显示

参数说明：

[in] lcd:选择 lcd 号，必须是 LCD_0 或者 LCD_1.

返回值说明：

AK_TURE:成功

AK_FALSE:失败

Anyka Confidential For BOMEI Use Only

2.7 MMC_SD_SDIO

2.7.1 功能概述

本平台支持 MMC/SD 卡以及 SDIO 接口拓展。本模块提供的接口主要用于 MMC/SD/SDIO 设置，包括数据传输、时钟设置、SD controller（详见芯片 *Programmer's Guide*）设置等。

2.7.2 接口说明

T_BOOL emmc_switch_partition (T_pCARD_HANDLE handle,
T_eCARD_PARTITION part
)

用来进入设置 MMC 4.3 版本以上的 6 组功能模式。若 MMC 卡规格版本为 4.3 及以上，则应该调用该函数来切换分区。

mmc4.3 later card switch partition

If card specification version is mmc4.3 later, this function should be called to switch partition

参数说明:

[in] handle card handle, a pointer of void //SD 卡句柄，由 sd_initial 得到

[in] part the selected partition //所选择的分区

返回值说明:

T_BOOL

AK_TRUE,: switch successfully //切换成功

AK_FALSE,: switch failed //切换失败

T_VOID sd_free (T_pCARD_HANDLE handle)

关闭 SD 控制器

Close sd controller.

参数说明:

[in] handle card handle,a pointer of void //SD 卡句柄，由 sd_initial 得到

返回值说明:

T_VOID

T_VOID sd_get_info (T_pCARD_HANDLE handle,
T_U32 * total_block,
T_U32 * block_size
)

获取 SD 卡信息

Get SD card information

AnyKa Confidential For BOMEI Use Only

参数说明:

[in] handle card handle,a pointer of void //SD 卡句柄，由 sd_initial 得到

[out] total_block current sd's total block number //当前 SD 卡总块数

[out] block_size current sd's block size;1 block = 512 bytes //当前 SD 块大小；1 块=512 字节

返回值说明:

T_VOID

```
T_pCARD_HANDLE sd_initial (T_eCARD_INTERFACE cif,
                           T_U8 bus_mode
                          )
```

初始化 MMC 卡或者 COMOB 卡

Initialize MMC, SD or comob card

参数说明:

[in] cif card interface selected //所选卡接口

[in] bus_mode bus mode selected, can be USE_ONE_BUS or USE_FOUR_BUS //所选的总线模式；可以是 USE_ONE_BUS 或者 USE_FOUR_BUS

返回值说明:

T_pCARD_HANDLE

NON-NULL set initial successful, card type is MMC, SD or comob//初始化成功；卡类型为 MMC、SD 或者 COMOB

NULL set initial fail, card type is not MMC, SD or comob card //初始化失败；不是 MMC、SD 或 COMOB 卡

```
T_BOOL sd_read_block (T_pCARD_HANDLE handle,
                      T_U32  block_src,
                      T_U8 *  databuf,
                      T_U32  block_count
)
```

从 SD 卡读取数据

read data from sd card

参数说明:

Anyka Confidential for BOMEI Use Only

[in] handle card handle,a pointer of void //SD 卡句柄，由 sd_initial 得到

[in] block_src source block to read //读取的源块

[out] databuf data buffer to read //读取的数据缓存

[in] block_count size of blocks to be readed //将要读取的块大小

返回值说明:

T_BOOL

AK_TRUE,: read successfully //读取成功

AK_FALSE,: read failed //读取失败

```
T_BOOL sd_set_clock (T_pCARD_HANDLE handle,
                     T_U32 clock
                    )
```

设置 SD 接口的时钟

Set the sd interface clk

参数说明:

[in] handle card handle,a pointer of void //SD 卡句柄，由 sd_initial 得到

[in] clock clock to set //设置的时钟

返回值说明:

T_BOOL

```
T_BOOL sd_write_block (T_pCARD_HANDLE handle,
                       T_U32 block_dest,
                       const T_U8 * databuf,
                       T_U32 block_count
                          )
```

写数据到 SD 卡

Write data to sd card

参数说明:

[in] handle card handle,a pointer of void //SD 句柄，由 sd_initial 得到

[in] block_dest destation block to write //要写入的目标块

AnyKa Confidential For OMEI Use Only

[in] databuf data buffer to write //要写的数据缓存

[in] block_count size of blocks to be written //将要写入的块大小

返回值说明:

T_BOOL

AK_TRUE:write successfully //写入成功

AK_FALSE,: write failed //写入失败

T_BOOL sdio_enable_func (T_U8 func)

使能 SDIO 卡中的特定功能

Enable specific function in SDIO card

参数说明:

[in] func function to enable //使能的功能

返回值说明:

T_BOOL

AK_TRUE enable successfully //使能成功

AK_FALSE enable failed //使能失败

T_BOOL sdio_initial (T_U8 bus_mode)

初始化 SDIO 或者 COMBO 卡

Initialize SDIO or combo card

参数说明:

[in] bus_mode bus mode selected, can be USE_ONE_BUS or USE_FOUR_BUS //所选的总线模式，可以为 USE_ONE_BUS 或者 USE_FOUR_BUS

返回值说明:

T_BOOL

AK_TRUE set initial successful, card type is SDIO or combo//初始化成功，且为 SDIO 或者 COMBO 卡

AK_FALSE set initial fail, card type is not SDIO or combo //初始化失败，不是 SDIO 或者 COMBO 卡

Anyka Confidential For BOMEI Use Only

```
T_BOOL sdio_read_byte (T_U8 func,
                       T_U32  addr,
                       T_U8 *  rdata
                      )
```

从 SDIO 设备中读取 1 字节数据

Read one byte from SDIO card

参数说明:

[in] func function to read //要读取的功能

[in] addr register address to read//要读取的寄存器地址

[in] rdata data buffer for read data //读取数据的数据缓存

返回值说明:

T_BOOL

AK_TRUE read successfully //读取成功

AK_FALSE read failed //读取失败

```
T_BOOL sdio_read_multi (T_U8 func,
                        T_U32  src,
                        T_U32  count,
                        T_U8  opcode,
                        T_U8  rdata[]
                       )
```

从 SDIO 设备中读取几字节或者几块数据

Read multiple byte or block from SDIO card

参数说明:

[in] func function to read //要读取的功能

[in] src register address to read //要读取的寄存器地址

[in] count data size(number of byte) to read //要读取的数据大小（字节数）

[in] opcode fixed address or increasing address //固定地址还是增值地址

[in] rdata data buffer for read data //读取数据的数据缓存

返回值说明:

Confidential For Public Only

T_BOOL

AK_TRUE read successfully //读取成功

AK_FALSE read failed //读取失败

T_BOOL sdio_select_card (T_U32 addr)

选择或不选择某个 SDIO 设备，通过 SDIO 自身的相应地址可以选择该设备，其他地址均无法选择该设备，0 为全不选。

select or deselect a sdio device

The card is selected by its own relative address and gets deselected by any other address; address 0 deselects all

参数说明:

[in] addr the address to select card //卡地址：通过分配的地址来选择卡（见功能定义），如果地址是 0，则全不选中。

返回值说明:

T_BOOL

AK_TRUE select or deselect successfully 操作成功

AK_FALSE select or deselect failed 操作失败

T_BOOL sdio_set_block_len (T_U8 func,

T_U32 block_len

)

设置 SDIO 卡的块长度

Set block length to SDIO card

参数说明:

[in] func function to set block length //要设置块长度的功能

[in] block_len block length to set //要设置的块长度

返回值说明:

T_BOOL

AK_TRUE enable successfully //设置成功

AK_FALSE enable failed //设置失败

Anyka Confidential For BOM Use Only

T_BOOL sdio_set_int_callback (T_SDIO_INT_HANDLER cb)

设置 SDIO 中断回调函数

Set SDIO interrupt callback function

参数说明:

[in] cb callback function //回调函数

返回值说明:

T_BOOL

AK_TRUE set successfully 设置成功

AK_FALSE set failed 设置失败

T_BOOL sdio_write_byte (T_U8 func,
T_U32 addr,
T_U8 wdata
)

往 SDIO 卡中写入 1 字节数据

Write one byte to SDIO card

参数说明:

[in] func function to write //要写的功能

[in] addr register address to write //要写的寄存器地址

[in] wdata the write byte //要写的数据

返回值说明:

T_BOOL

AK_TRUE write successfully //写入成功

AK_FALSE write failed // 写入失败

T_BOOL sdio_write_multi (T_U8 func,
T_U32 dest,
T_U32 count,
T_U8 opcode,
T_U8 wdata[]

Anyka Confidential For BOMEI Use Only

)

写几 bytes 或者几 blocks 到 SIDO 卡中

Write multiple byte or block from SDIO card

参数说明:

[in] func function to read //读的功能

[in] dest register address to read //要读的起始地址

[in] count data size(number of byte) to read //要读的字节长度

[in] opcode fixed address or increasing address //固定地址还是增值地址

[in] wdata the wirte data //要写的数据

返回值说明:

T_BOOL

AK_TRUE write successfully //写入成功

AK_FALSE write failed //写入失败

2.8 MMU

2.8.1 功能概述

MMU（Memory Management Unit）内存管理单元是中央处理器（CPU）中用来管理虚拟存储器、物理存储器的控制线路，同时也负责虚拟地址映射为物理地址，以及提供硬件机制的内存访问授权。本节主要介绍 MMU 相关的接口说明。

2.8.2 接口说明

T_VOID MMU_Clean_All_DCache (T_VOID)

清除所有高速 buffer（DCACHE）

Clean all DCACHE

返回值说明:

T_VOID

T_VOID MMU_Clean_Invalidate_Dcache (T_VOID)

清除高速缓存（DCACHE）并使之无效

Anyka Confidential For BOMEI Use Only

Clean and invalidate DCACHE

返回值说明:

T_VOID

T_VOID MMU_DisableDCache (T_VOID)

关闭高速缓存（DCACHE）

Disable DCACHE

返回值说明:

T_VOID

T_VOID MMU_DisableICache (T_VOID)

关闭 ICACHE

Disable ICACHE

返回值说明:

T_VOID

T_VOID MMU_DisableMMU (T_VOID)

关闭 MMU（电源管理单元）

Disable MMU

返回值说明:

T_VOID

T_VOID MMU_DrainWriteBuffer (T_VOID)

排空写入缓存

Drain Write buffer.

返回值说明:

T_VOID

T_VOID MMU_EnableDCache (T_VOID)

打开 DCACHE

Enable DCACHE

返回值说明:

AnyKa Confidential For BOMEI Use Only

T_VOID

T_VOID MMU_EnableICache (T_VOID)

打开 ICACHE

Enable ICACHE

返回值说明:

T_VOID

T_VOID MMU_EnableMMU (T_VOID)

打开 MMU

Enable MMU

返回值说明:

T_VOID

T_VOID MMU_Init (T_U32 mmutt_start_addr)

初始化 MMU；1. 设置访问属性；2. 使能 ICACHE 和 DCACHE；3. 使能 MMU；4. 设置 TT（页表）起始地址

Initial MMU

1. Set access attributes.

2. Enable ICACHE and DCACHE.

3. Enable MMU.

4. Set TT(translation table) start address

参数说明:

[in] mmutt_start_addr MMU translation table's start address //MMU 页表的起始地址

返回值说明:

T_VOID

T_VOID MMU_InvalidateDCache (T_VOID)

关闭 DCACHE，DCACHE 中数据将会清除

Invalidate DCACHE

Data in DCACHE will be clear

返回值说明:

Anyka Confidential For BOMEI Use Only

T_VOID

T_VOID MMU_InvalidateICache (T_VOID)

关闭 ICACHE，ICACHE 中数据将会清除

Invalidate ICACHE; data in ICACHE will be clear

返回值说明:

T_VOID

T_VOID MMU_InvalidateIDCache (T_VOID)

同时关闭 ICACHE 和 DCACHE

返回值说明:

T_VOID

T_VOID MMU_InvalidateTLB (T_VOID)

使 TLB 无效（TLB：Translation lookaside buffer，即旁路转换缓冲或称为页表缓冲）

invalidate TLB

返回值说明:

T_VOID

```
T_VOID MMU_SetMTT ( T_U32 vaddrStart,
                    T_U32 vaddrEnd,
                    T_U32 paddrStart,
                    T_U32 attr
                  )
```

设置分区描述符

Set section descriptor

参数说明:

[in] vaddrStart start of virtual address //虚拟地址的起始位

[in] vaddrEnd end of virtual address //虚拟地址的末位

[in] paddrStart end of physical address //物理地址的末位

[in] attr attribute of access //存取属性

返回值说明:

T_VOID

2.9 RTC

2.9.1 功能概述

实时时钟模块（Real-Time Clock， RTC）由晶振及相关电路组成的时钟电路生成脉冲，RTC 经过变频产生系统时钟，提供稳定的时钟信号给后续电路。

本节主要介绍 RTC 相关接口使用说明。

2.9.2 接口说明

T_VOID rtc_enter_standby (T_VOID)

进入待机模式

Enter standby mode

返回值说明:

T_VOID

T_U16 rtc_exit_standby (T_VOID)

退出待机模式，返回原因

Exit standby mode and return the reason

返回值说明:

T_U32 the reason of exiting standby //退出待机模式的原因

low 8-bit is wakeup type, refer to T_WU_TYPE //低 8 位代表唤醒模块类型，参见 T_WU_TYPE 定义

Upper 8-bit stands for gpio number if WGPIO wakeup //高 8 位代表 WGPIO 唤醒后的 GPIO 数

T_BOOL rtc_get_alarm_status (T_VOID)

查询闹钟状态

Query alarm status

返回值说明:

T_BOOL

Anyka Confidential For BOMEI Use Only

AK_TRUE alarm has occurred //已设置闹钟

AK_FALSE alarm hasn't occurred //未设置闹钟

T_U32 rtc_get_alarmcount (T_VOID)

获取已经设置好的闹钟计时

Get alarm count that has been set.

返回值说明:

T_U32

The alarm count in seconds //按秒计算 alarm count

T_SYSTIME rtc_get_AlarmSystime (T_VOID)

从 RTC 获取闹钟系统时间

Get alarm system time from RTC

返回值说明:

T_SYSTIME system time structure //系统时间结构

T_U32 rtc_get_RTCcount (T_VOID)

获取 RTC 历时计时数，按秒算

Get RTCpassed count in seconds

返回值说明:

T_U32 the RTC count //RTC 计时数

T_SYSTIME rtc_get_RTCsystime (T_VOID)

从 RTC 上获取当前系统时间

Get current system time from rtc

返回值说明:

system time structure //时间结构

T_VOID rtc_init (T_U32 year)

初始化 RTC 模块

AnyKa Confidential For BOMEI Use Only

Init RTC module

参数说明:

year [in] current year //当前年份

返回值说明:

T_VOID

T_VOID rtc_set_AlarmBySystime (T_SYSTIME * systime)

根据系统时间设置闹钟

Set alarm by system time

参数说明:

systime system time structure //系统时间结构

返回值说明:

T_U32: day num //天数

T_VOID rtc_set_alarmcount (T_U32 rtc_wakeup_value)

设置 RTC 闹铃计时

Set RTCalarm count.

当 RTC COUNT 达到 ALARM COUNT 时，芯片将会从待机模式唤醒同时产生 RTC 中断

When the RTC count reaches to the alarm count, AK chip is woken up if in standby mode and RTC interrupt happens.

参数说明:

rtc_wakeup_value [in] alarm count in seconds // 闹铃计时值（单位：秒）

返回值说明:

T_VOID

T_VOID rtc_set_callback (T_fRTC_CALLBACK cb)

设置 RTC 事件回调处理函数

Set RTC alarm event callback handler

参数说明:

cb [in] the alarm event callback handler//alarm 事件回调处理函数

Anyka Confidential For BOMEI Use Only

返回值说明:

T_VOID

T_VOID rtc_set_powerdownalarm (T_BOOL alarm_on)

使能或关闭“关闭闹钟”功能

Enable or disable power down alarm

参数说明:

alarm_on [in] enable power down alarm or not //是否使能该功能

返回值说明:

T_VOID

T_VOID rtc_set_RTCbySystime (T_SYSTIME * systime)

根据系统时间是设置 RTC 寄存器值

Set RTC register value by system time

参数说明:

systime T_SYSTIME : system time structure //系统时间结构

返回值说明:

T_U32 day num //天数

T_VOID rtc_set_RTCcount (T_U32 rtc_value)

设置 RTC 起始计时值（单位：秒）

Set RTC start count value in seconds

参数说明:

rtc_value [in] the rtc count to be set //设置的 RTC 起始 count 值

返回值说明:

T_VOID

T_VOID rtc_set_wakeup_type (T_WU_TYPE type)

为退出待机模式设置唤醒类型

Set wakeup type for exiting standby mode

参数说明:

Anyka Confidential For BOMEI Use Only

type [in] wakeup type, WU_GPIO and WU_ALARM default opened //唤醒类型，默认状态下 WU_GPIO 和 WU_ALARM 为打开

返回值说明:

T_VOID

T_VOID rtc_set_wgpio (T_U32 wgpio_mask)

设置待机模式的唤醒 GPIO

Set wakeup gpio of standby mode

参数说明:

wgpio_mask [in] the wakeup gpio value //唤醒 GPIO 值

返回值说明:

T_VOID

T_VOID rtc_set_wpinLevel (T_BOOL wpinLevel)

设置唤醒 PIN 的激活电平状态

Set wakeuppin active level

参数说明:

wpinLevel [in] the wakeup signal active level 1:low active,0:high active //1：低电平有效；2：高电平有效

返回值说明:

T_VOID

Anyka Confidential For BOMEI Use Only

2.10 UART

2.10.1 功能概述

UART: Universal Asynchronous Receiver/Transmitter，通用异步接收/发送装置，其主要功能是使并行输入数据成为串行输出数据。

UART 支持的波特率（baud rate）定义如下：

UART_BAUD_9600 9600

UART_BAUD_19200 19200

UART_BAUD_38400 38400

UART_BAUD_57600 57600

UART_BAUD_115200 115200

UART_BAUD_460800 460800

2.10.2 接口说明

T_VOID uart_free (T_UART_ID uart_id)

关闭 UART（调用此函数之前必须先调用 uart_init()函数）

Close UART. Function uart_init() must be called before call this function

参数说明:

[in] uart_id UART ID

返回值说明:

T_VOID

T_BOOL uart_init (T_UART_ID uart_id,
T_U32 baud_rate,
T_U32 sys_clk
)

初始化 UART

Initialize UART.

注意事项:

根据 UART 的 ID、波特率和系统时钟初始化 UART。如果用户想要改变波特率或者系统时钟有变，用户应该调用此函数重新初始化 UART。在调用 UART 所有其它函数之前，必须先调用 uart_init()函数。

Initialize UART base on UART ID, baud rate and system clock. If user wants to change baud rate or system clock is changed, user should call this function to initialize UART again. Function uart_init() must be called before call any other UART functions

参数说明:

[in] uart_id UART ID

[in] baud_rate Baud rate, use UART_BAUD_9600, UART_BAUD_19200 ... //波特率

[in] sys_clk system clock //系统时钟

返回值说明:

Anyka Confidential For BOMEI Use Only

T_BOOL Init UART OK or not

AK_TRUE Successfully initialized UART //初始化 UART 成功

AK_FALSE Initializing UART failed. //初始化 UART 失败

T_VOID uart_on_change (T_U32 sys_clk)

根据系统时钟的变化改变 UART 的设置（调用此函数之前必须先调用 uart_init()函数）

Change UART setting according to system clock change

Function uart_init() must be called before call this function

参数说明:

[in] sys_clk system clock //系统时钟

返回值说明:

T_VOID

```
T_U32 uart_read ( T_UART_ID uart_id,
                  T_U8 * data,
                  T_U32 datalen
                )
```

从 UART Pool 中读取数据

Read data from uart pool

参数说明:

[in] uart_id UART ID

[out] *data buffer to recieve data,as big as you can //接收数据的 buffer，尽量大

[in] datalen data length to read //要读取的数据长度

返回值说明:

T_U32

the data length that has been read //已被读取的数据长度

```
T_BOOL uart_read_chr (T_UART_ID uart_id,
                      T_U8 * chr
                     )
```

从 UART 上读取一个字

Anyka Confidential For BOMEI Only

Read a character from UART.

该函数只有从 UART 获取了一个字之后才会返回；调用该函数前必须先调用 uart_init()函数。

This function will not return until get a character from UART Function uart_init() must be called before call this function

参数说明:

[in] uart_id UART ID

[out] *chr character for return //返回的字

返回值说明:

T_BOOL Got character or not

return AK_TRUE

```
T_VOID uart_set_callback (T_UART_ID uart_id,
                          T_fUART_CALLBACK callback_func
                          )
```

注册一个回调函数用来处理 UART 接收的数据；调用此函数之前必须先调用 uart_init()函数。

Register a callback function to process UART received data.

Function uart_init() must be called before call this function

参数说明:

[in] uart_id UART ID

[in] callback_func Callback function //回调函数

返回值说明:

T_VOID

```
T_VOID uart_set_datapool ( T_UART_ID uart_id,
                           T_U8 * pool,
                           T_U32 poollength
                         )
```

设置接收数据的库，之前必须调用 uart_set_callback 函数。

Set pool to receive data, must call before uart_set_callback

Anyka Confidential For BOMEI Use

参数说明:

[in] uart_id UART ID

[in] *pool buffer to recieve data,as big as you can //接收数据的 buffer，尽量设置成最大

[in] poollength buffer length //buffer 长度

返回值说明:

T_VOID

注意事项:

UART 接收的数据将会存储在库当中等待 uart_read()函数的读取。

The data received from UART will be stored in the pool waiting to be fetched by uart_read().

```
T_BOOL uart_setbaudrate ( T_UART_ID uart_id,
                          T_U32 baud_rate
                        )
```

改变 UART 波特率

Change uart baudrate

参数说明:

[in] uart_id UART ID

[in] baud_rate T_U32 new baud_rate to change //要改变成的新波特率

返回值说明:

T_BOOL

if success, return AK_TRUE //返回 AK_TURE 为成功

```
T_BOOL uart_setdataparity (T_UART_ID uart_id,
                           T_BOOL enable,
                           T_BOOL evenParity
                          )
```

设置 UART 奇偶校验位模式，使用偶数校验位或者奇数校验位或者两种都不使用

Set UART data parity mode; use even parity or odd parity or neither

参数说明:

[in] uart_id UART ID

AnyKa Confidential For BOMEI Use Only

[in] enable T_BOOL, enable data parity or not //是否使能数据奇偶校验位

[in] evenParity T_BOOL, if AK_TRUE enable even parity, else enable odd parity //AK_TURE 则使能偶数校验位，否则使能奇数校验位

返回值说明:

T_BOOL

If success, return AK_TRUE

```
T_BOOL uart_setflowcontrol (T_UART_ID uart_id,
                            T_BOOL enable
                            )
```

使能或关闭 UART 流控功能；调用该函数之前必须先调用 uart_init()函数

Enable or disable UART flow control

Function uart_init() must be called before call this function

参数说明:

[in] uart_id which uart to be set

[in] enable AK_TRUE to enable flow control //AK_TURE 则打开流控

返回值说明:

T_BOOL

AK_TRUE 打开成功

AK_FALSE 打开失败或者不支持

```
T_U32 uart_write ( T_UART_ID uart_id,
                   const T_U8 * data,
                   T_U32 data_len
                 )
```

写入串行数据到 UART

Write string data to UART.

根据 UART 的 ID 和数据长度写入数据到 UART；调用此函数之前必须先调用 uart_init()函数。

Write data to UART base on UART ID and data length Function uart_init() must be called before call this function

Anyka Confidential
For COMEI Use Only

参数说明:

[in] uart_id UART ID

[in] data Constant data pointer to be written to UART, this data needn't be ended with '\0' //要被写入 UART 的连续数据指针，该数据不需要以'\0'结尾。

[in] data_len Data length //数据长度

返回值说明:

T_U32

Length of the data which have been written to UART //已经写入 UART 的数据长度

2.11 GPIO

2.11.1 功能概述

General Purpose Input Output 简称为 GPIO，为 通用输入/输出或总线扩展器，当芯片组没有足够的 I/O 端口，或当系统需要采用远端串行通信或控制时，GPIO 产品能够提供额外的控制和监视功能。

2.11.2 数据结构

T_SHARE_COMPONENT

```
typedef struct
{
   T_U8        gpio_start; //模块引脚复用的开始 GPIO 号
   T_U8        gpio_end; //模块引脚复用的结束 GPIO 号
   T_U8        reg_num; //复用引脚配置寄存器的索引标号。0、1、2
   T_U32       bit_mask; //配置复用引脚的位
   T_U32       bit_value; //配置复用引脚为的具体值
   E_GPIO_PIN_SHARE_CONFIG module; //这些复用的引脚属于哪个模块
}T_SHARE_COMPONENT;
```

2.11.3 接口说明

```
T_VOID  gpio_share_pin_init(T_VOID)
```

AnyKa Confidential For BOMEI Use Only

初始化并设置默认的复用引脚模块

参数说明：无

返回值说明：无

T_U8 get_wGpio_Bit(T_U32 pin,T_U32* ctreg);

获取 GPIO 唤醒引脚的 bit 位

Get gpio wakeup pin bit

参数说明:

[in] pin gpio pin ID. //GPIO 引脚 ID

[in]ctreg 配置 GPIO 唤醒寄存器的索引。0 或者 1.

返回值说明:

T_U8

GPIO pin bit in wakeup register //在唤醒寄存器上的引脚位

T_U8 gpio_get_pin_level (T_U32 pin)

获取 GPIO 输入电平

Get GPIOinput level.

参数说明:

[in] pin GPIO pin ID.

返回值说明:

T_U8

1 level high //高电平

0 level low; //低电平

T_VOID gpio_init (T_VOID)

初始化 GPIO

Init GPIO.

返回值说明:

T_VOID

Anyka Confidential For BOMEI Use Only

T_VOID gpio_int_control (T_U32 pin,
T_U8 enable
)

GPIO 中断控制

GPIO interrupt control

参数说明:

[in] pin GPIO pin ID.

[in] enable 1 means enable interrupt. 0 means disable interrupt. //1 打开中断控制；0 为关闭中断控制

返回值说明:

T_VOID

T_VOID gpio_int_disableall (T_VOID)

关闭所有 GPIO 引脚中断控制

Disable all GPIO pin interrupt

返回值说明:

T_VOID

T_VOID gpio_int_restoreall (T_VOID)

恢复所有的 GPIO 中断.

Restore all GPIO pin interrupt

返回值说明:

T_VOID

T_BOOL gpio_pin_group_cfg (E_GPIO_PIN_SHARE_CONFIG PinCfg)

把一组 GPIO 引脚设置为用作特定模块

Set GPIO pin group as specified module used

参数说明:

[in] PinCfg enum data. the specified module //枚举指定的模块

返回值说明:

T_BOOL

Anyka Confidential For BOMEI Use Only

AK_TURE setting successful //设置成功

AK_FALSE setting failed //设置失败

T_VOID gpio_register_int_callback (T_U32 pin,
T_U8 polarity,
T_U8 enable,
T_fGPIO_CALLBACK callback
)

注册 GPIO 中断回调函数

Register one GPIO interrupt callback function.

参数说明:

[in] pin GPIO pin ID.

[in] polarity 1 means active high interrupt. 0 means active low interrupt. //1 为高电平有效中断；0 为低电平有效中断

[in] enable Initial interrupt state--enable or disable. //初始化的中断状态（使能或关闭）

[in] callback GPIO interrupt callback function. //GPIO 中断回调函数

返回值说明:

T_VOID

T_VOID gpio_set_int_p (T_U32 pin,
T_U8 polarity
)

设置 GPIO 中断极性

Set gpio interrupt polarity.

参数说明:

[in] pin gpio pin ID.

[in] polarity 1 means active high interrupt. 0 means active low interrupt. //1 为高电平有效中断；0 为低电平有效中断

返回值说明:

T_VOID

Anyka Confidential For BUEI Use Only

T_BOOL gpio_set_pin_as_gpio (T_U32 pin)

将 shared-pin GPIO 设置为用作 GPIO

Set GPIO share pin as GPIO

参数说明:

pin [in] GPIO pin ID

返回值说明:

T_BOOL

AK_TRUE set successfully //设置成功

AK_FALSE fail to set //设置失败

T_BOOL gpio_set_pin_attribute (T_U32 pin,
T_GPIO_PIN_ATTR attr,
T_BOOL enable
)

设置 GPIO 属性如 IE,PE 等

Set GPIO pin attribute as IE, PE, etc.

参数说明:

[in] pin the pin ID to set

[in] attr the atrribute to set

[in] enable enable the attribute or not //是否使能该属性

返回值说明:

T_BOOL

AK_TURE setting successful //设置成功

AK_FALSE setting failed //设置失败

T_VOID gpio_set_pin_dir (T_U32 pin,
T_U8 dir
)

设置 GPIO 信号方向

Set GPIO direction.

参数说明:

Anyka Confidential For BOMEI Use Only

[in] pin GPIO pin ID.

[in] dir 0 means input; 1 means output; //0 为输入；1 为输出

返回值说明:

T_VOID

T_VOID gpio_set_pin_level (T_U32 pin,
T_U8 level
)

设置 GPIO 输出电平

Set GPIO output level.

参数说明:

[in] pin GPIO pin ID.

[in] level 0 or 1.

返回值说明:

T_VOID

T_BOOL gpio_set_pull_down_r (T_U32 pin,
T_BOOL enable
)

GPIO 下拉电阻配置函数

GPIO pull-down resistance configuration function

参数说明:

[in] pin GPIO pin ID.

[in] enable 1 means enable pull down. 0 means disable pull down. //1 为打开下拉功能；0 为关闭下拉功能

返回值说明:

T_BOOL

AK_TURE configuration successful //配置成功

AK_FALSE configuration failed //配置失败

Anyka Confidential Property
BOM Reference Only

T_BOOL gpio_set_pull_up_r (T_U32 pin,
T_BOOL enable
)

GPIO 上拉电阻配置函数

GPIO pull-up resistance configuration function

参数说明:

[in] pin GPIO pin ID.

[in] enable 1 means enable pull up. 0 means disable pull up. //1 为打开上拉功能；0 为关闭上拉功能

返回值说明:

T_BOOL

AK_TURE configuration successful //配置成功

AK_FALSE configuration failed //配置失败

T_VOID gpio_set_wakeup_p (T_U32 pin,
T_BOOL polarity
)

设置 GPIO 唤醒极性

Set GPIO wake up polarity.

参数说明:

[in] pin GPIO pin ID.

[in] polarity 1 means high level. 0 means low level. //1 为高电平；0 为低电平

返回值说明:

T_VOID

2.12 Detector

2.12.1 功能概述

Detector 用于检测用户的一些连接操作，例如：SD/MMC card 的插拔、U 盘的插拔、充电器的插拔、耳机插拔以及跟 PC 连接线的插拔等等。

Anyka Confidential For BOMEI Use Only

2.12.2 接口说明

T_BOOL detector_register_gpio(T_pCSTR devname, T_U32 gpio_num,
T_BOOL active_level, T_BOOL interrupt_mode,
T_U32 interval_ms);

注册 GPIO 类型的 Detector

Register the detector of GPIO type.

参数说明:

[in] devname Name of the device to be detected. //被检测的设备名

Devname must point to a const string, because the detect module won't hold a copy of the device name. //devname 必须指向一个常量字符串

[in] gpio_num Number of the GPIO. //用于检测的 GPIO 编号

[in] active_level Active logic level, 0 or 1. //有效电平，表示设备连接时的 GPIO 电平

[in] interrupt_mode Detect type, AK_TRUE: interrupt, AK_FALSE: time. //检测方式

[in] interval_ms The interval of checking, in ms. //检测间隔时间

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

T_BOOL detector_register_adc(T_pCSTR *devname_list, T_U32 devnum,
const T_VOLTAGE_TABLE *pvoltage_table,
T_U32 voltageitem_num, T_U32 interval_ms）;

注册 ADC 类型的 Detector

Register the detector of ADC type.

参数说明:

[in] devname_list Name list of the devices to be detected. //被检测的设备名列表

Each pointer in the name list must point to a const string, because the detect module won't hold a copy of the device name. //列表中的指针必须指向常量字符串

[in] devnum Number of devices to be detected. //设备的数量

[in] pvoltage_table Voltage table . //电压表

Anyka Confidential For BETA Use Only

pvoltage_table must point to a const memory, because the detect module won't hold a copy of the Voltage table. // pvoltage_table 必须指向常量空间

[in] voltageitem_num Number of voltage item of voltage table. //项数

[in] interval_ms The interval of checking, in ms. //检测间隔时间

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

T_BOOL detector_init(T_VOID);

初始化 Detector 模块。

Initialize detector module.

参数说明:

T_VOID

返回值说明:

T_VOID

T_BOOL detector_free(T_VOID);

释放 Detector 模块。

Free detector module.

参数说明:

T_VOID

返回值说明:

T_VOID

T_BOOL detector_set_callback(T_pCSTR devname,
T_fDETECTOR_CALLBACK pcallbackfunc);

为名为 devname 的设备设置回调函数。当设备连接活或者断开时，回调函数就会被调用。

Set the callback function of device named by devname, the call back function will be called when the device inserted or removed.

参数说明:

[in] devname Name of the device to be detected. //设备名

Anyka Confidential For BOMEI Use Only

[in] pcallbackfunc Call back function of the device. //回调函数

Typedef T_VOID (*T_fDETECTOR_CALLBACK)(T_BOOL state); //回调函数原型

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

T_BOOL detector_enable(T_pCSTR devname, T_BOOL benable);)

使能或禁止设备 devname 的 Detector。

Enable or disable the detector.

不建议禁止 detector，因为在禁止的时间段内，设备的连接状态无法通知给上层应用。

It's not suggested to disable the detector. If the detector is disable, the connecting state of the device can't be informed the user.

参数说明:

[in] devname Name of the device to be detected. //设备名

[in] benable Enable or disable the detector //使能或禁止

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

T_BOOL detector_is_enabled(T_pCSTR devname, T_BOOL *pbenable);

检查设备名为 devname 的 detector 是使能还是禁止。

Determine whether the specified window is enabled.

参数说明:

[in] devname Name of the device to be detected. //设备名

[out] pbenable Pointer to a T_BOOL type variable for fetching the detector state. //获取状态

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

Anyka Confidential For BOMEI Use Only

T_BOOL detector_get_state(T_pCSTR devname，T_BOOL *pState);

获取设备名为 devname 的设备的连接状态。

Get the connecting state of the device named by devname.

参数说明:

[in] devname Name of the device to be detected. //设备名

[out] pState Pointer to a T_BOOL type variable for fetching the connecting state. //获取状态

返回值说明:

If the function succeeds, the return value is AK_TRUE; //成功

If the function fails, the return value is AK_FALSE; //失败

```
T_BOOL card_detect_reg(T_eCARD_INTERFACE card_type,
                       T_U32 gpio_num,
                       T_BOOL benable_dat3,
                       T_fDETECTOR_CALLBACK pcallbackfunc
                      )
```

注册 SD/MMC 的检测功能

参数说明：

[in]card_type：选择卡的类型，相关值参考 T_eCARD_INTERFACE 这个结构体。

[in]gpio_num：作为检测引脚的 GPIO 号。

[in]benable_dat3：是否使用卡的 data3 作为检测功能，AK_TURE 则使用。

[in]pcallbackfunc：底层检测与上层应用之间的回调函数。

返回值说明：

T_BOOL

AK_TURE，注册成功

AK_FALSE，写入失败

```
T_BOOL card_detect_enable(T_eCARD_INTERFACE card_type,T_BOOL benable);
```

使能 card 的检测功能，卡的检测在注册后，必须使能才能有效。

参数说明：

[in]card_type：选择卡的类型，相关值参考 T_eCARD_INTERFACE 这个结构体。

Anyka Confidential For BOMEI Use

[in]benable: AK_TURE 使能，AK_FALSE 禁止

返回值说明：

AK_TRUE ：成功

AK_FALSE ：失败

```
T_BOOL card_detector_get_state(T_eCARD_INTERFACE card_type, T_BOOL *pState);
```

获取卡的状态，是否有卡。

参数说明：

[in]card_type：选择卡的类型，相关值参考 T_eCARD_INTERFACE 这个结构体。

[out]pState：指针指向的值，AK_TURE 表示有卡，AK_FALSE 表示没卡。

返回值说明：

T_BOOL

AK_TRUE ：获取状态成功

AK_FALSE ：获取状态失败

2.13 I2C

2.13.1 功能概述

I2C（Inter－Integrated Circuit）总线为两线式串行总线，接口线少，控制方式简单，器件封装形式小，通信速率较高，用于连接微控制器及其外围设备。SCCB（Serial Camera Control Bus）接口定义与 I2C 接口定义相同，在此不再重复详述。

2.13.2 接口说明

```
T_VOID i2c_init  (  T_U32  pin_scl,
                    T_U32 pin_sda
                  )
```

初始化 I2C 接口 ，启动 I2C 接口

I2C interface initialization function.

Setup I2C interface

参数说明:

AnyKa Confidential For BOMEI Use Only

[in] pin_scl the pin assigned to SCL //SCL 线复用的引脚

[in] pin_sda the pin assigned to SDA //SDA 线复用的引脚

返回值说明:

T_VOID

```
T_BOOL i2c_read_data (T_U8 daddr,
                      T_U8  raddr,
                      T_U8 *  data,
                      T_U32 size
                     )
```

从 I2C 设备中读取数据

Read data from I2C device function

从指定地址寄存器读数据, 寄存器地址和数据宽度为 1 个字节

Read data from daddr's raddr register, raddr and data is byte width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[out] *data read output data store address//输出数据存储地址

[in] size read data size //读取数据大小

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_BOOL i2c_read_data2 (T_U8 daddr,
                       T_U16  raddr,
                       T_U8 *  data,
                       T_U32 size
                           )
```

从 I2C 设备中读取数据

Anyka Confidential
BOMEI Use Only

Read data from I2C device function

从指定地址寄存器读数据，寄存器地址宽度为 2 个字节（1 个字宽）；数据宽度为 1 个字节。

Read data from daddr's raddr register, raddr is word width, data is byte width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[out] *data read output data store address //输出数据存储地址

[in] size read data size //读取数据大小

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_BOOL i2c_read_data3 (T_U8 daddr,
                       T_U16  raddr,
                       T_U16 *  data,
                       T_U32 size
                           )
```

从 I2C 设备中读取数据

Read data from I2C device function

从指定地址寄存器读数据,寄存器地址和数据宽度为 2 个字节（1 个字宽）。

Read data from daddr's raddr register, raddr and data is word width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[out] data read output data store address//输出数据存储地址

[in] size read data size //读取数据大小

返回值说明:

T_BOOL return operation successful or failed

Anyka Confidential For BOM Only

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_BOOL i2c_read_data4 (T_U8 daddr,
                       T_U8 *  data,
                       T_U32  size
                      )
```

从 I2C 设备中读取数据

Read data from I2C device function

从指定地址寄存器读数据，不需要寄存器地址，数据宽度为 1 个字节。

Read data from daddr's raddr register, raddr is not required, data is byte width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[out] data read output data store address //输出数据存储地址

[in] size read data size //读取数据大小

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_VOID i2c_release ( T_U32  pin_scl,
                     T_U32  pin_sda
                   )
```

释放 I2C 总线

I2C interface release function.

参数说明:

[in] pin_scl the pin assigned to SCL //SCL 复用引脚

[in] pin_sda the pin assigned to SDA //SDA 复用引脚

返回值说明:

T_VOID

Anyka Confidential For OMEI Use Only

T_U32 i2c_set_cycle_delay (T_U32 delay)

设置 I2C 设备周期延迟时间，设置该函数可以改变 I2C 传输周期和频率

Set I2C device cycle delay time, the I2C transmit cycle and freq will be changed by setting this.

如果用户想复位 I2C 的传输频率，调用该函数之前应该先调用 i2c_init 函数。在传输数据过程中不可调用。默认延迟时间为 90 个 CPU CLK，由于使用 CPU LOOP 来延迟周期，I2C 周期和频率跟 CPU 频率也有关系。在默认延迟时间下，CPU 运行在 15MHz 和 84MHz 之间时，I2C 传输运行正常。如果 I2C 数据传输运行正常，建议不要复位 I2C 延迟时间，但是如果数据传输出错，I2C 频率过高或者过低，用户可使用该函数对 I2C 频率进行复位。

It should be call after i2c_init if want to reset the I2C transmit frequency. It should not be call in transmitting data. The default delay time value is 90, because we use CPU loop to delay, the cycle and freq will be relate to the CPU frequency too. In default delay time and the CPU run between 15MHz and 84MHz, the I2C transmit ok. If the data transmit ok, we suggest not reset the delay time. But if data transmit error, and the I2C freq is too high or too low, use this function to reset the freq of I2C.

参数说明:

[in] delay set delay time in every cycle by this value, this delay value is not exactitude to delay time, but can change the I2C cycle and freq by it //这个值是 I2C 读写延时设置的，通过它可以改变读写的周期和频率。

返回值说明:

T_U32

the delay value which is set //设置的延迟时间值

T_BOOL i2c_write_data (T_U8 daddr,
T_U8 raddr,
T_U8 * data,
T_U32 size
)

写数据到 I2C 设备

Write data to I2C device

写指定长度 length 的数据到 I2C 指定寄存器，寄存器地址和数据宽度为 1 个字节。

Write size length data to daddr's raddr register, raddr and data is byte width

参数说明:

Anyka Confidential For COMEI Use Only

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[in] *data write data's pointer //写数据的指针

[in] size write data's length //写数据的长度

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_BOOL i2c_write_data2 (T_U8 daddr,
                        T_U16 raddr,
                        T_U8 * data,
                        T_U32 size
                        )
```

写入数据到 I2C 设备

Write data to I2C device

写指定长度 length 的数据到 I2C 指定寄存器，寄存器地址为 2 个字节（1 个字宽），数据宽度为 1 个字节。

Write size length data to daddr's raddr register, raddr is word width, data is byte width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[in] *data write data's pointer //写数据指针

[in] size write data's length //写入数据的长度

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

Anyka Confidential For BOMEI Use Only

```
T_BOOL i2c_write_data3 (T_U8 daddr,
                        T_U16  raddr,
                        T_U16 *  data,
                            T_U32 size
                       )
```

写入数据到 I2C 设备

Write data to I2C device

写指定长度 length 的数据到 I2C 指定寄存器，寄存器地址和数据宽度为 2 个字节（1 个字宽）。

Write size length data to daddr's raddr register, raddr and data is word width

参数说明:

[in] daddr I2C device address //I2C 设备地址

[in] raddr register address //寄存器地址

[in] *data write data's pointer //写入数据指针

[in] size write data's length //写入数据的长度

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

```
T_BOOL i2c_write_data4 (T_U8 daddr,
                        T_U8 *  data,
                        T_U32  size
                       )
```

写入数据到 I2C 设备

Write data to I2C device

写指定长度 length 的数据到 I2C 指定寄存器中，数据宽度为 1 个字节。

Write size length data to daddr's raddr register, raddr is not required, data is byte width

参数说明:

[in] daddr I2C device address //I2C 设备地址

Anyka Confidential For OEM Use Only

[in] *data write data's pointer //写入数据的指针

[in] size write data's length //写入数据的长度

返回值说明:

T_BOOL return operation successful or failed

AK_FALSE operation failed //操作失败

AK_TRUE operation successful //操作成功

2.14 Keypad

2.14.1 功能概述

本节主要介绍有关 Keypad 的接口使用说明。用户可以根据相关接口设置相应的按键。有关键盘的相关定义如下：

Enum T_eKEY_PRESSMODE

定义按键模式，可以是单按模式或者多按模式

Press mode define, it can be single mode or multiple mode

Enumerator:

eSINGLE_PRESS single mode, only one key will be recognized, even there are other key pressed down, this is ordinary mode //单按模式（常用模式）：只识别一个按键，即使同时按下了多个按键。

eMULTIPLE_PRESS support more than one key pressed, only used in games //多按模式：支持两个或者以上按键同时按，只在游戏中有效

eKEYPRESSMODE_NUM the number of key press mode //按键模式的键数量

Enum T_KEYPADSTATUS

定义按键状态

Key status defininition

Enumerator:

eKEYDOWN key press down //按下

eKEYPRESS key press down and hold //按下并不动

eKEYUP key up //按键弹起状态

eKEYSTATUS_NUM num of key status //键状态数量

AnyKa Confidential For BOMEI Use Only

2.14.2 接口说明

T_BOOL keypad_reg_scanmode (T_U32 index,
T_KEYPAD_HANDLE * handler
)

注册键盘扫描模式

Register keypad scan mode

参数说明:

[in] index keypad index //键盘索引

[in] handler keypad handler //键盘处理函数

返回值说明:

T_BOOL

T_VOID keypad_disable_intr (T_VOID)

关闭键盘中断功能； 调用该函数之前必须调用 keypad_init()函数

Disable keypad interrupt

Function keypad_init() must be called before call this function

返回值说明:

T_VOID

T_VOID keypad_enable_intr (T_VOID)

打开键盘中断功能； 调用该函数之前必须调用 keypad_init()函数

Enable keypad interrupt

Function keypad_init() must be called before call this function

返回值说明:

T_VOID

T_BOOL keypad_get_key (T_KEYPAD * key)

获取当前按下的键，主要用于键盘回调函数

Get the keypad currently pressed, mainly used in keypad callback function

参数说明:

[in] key key structure pointer //键结构指针

Anyka Confidential For BOMEI Use Only

返回值说明:

T_BOOL

AK_TRUE get key success //获取成功

AK_FALSE get key failure, and key->keyID will kbNULL//获取失败；按键获取失败

T_eKEY_PRESSMODE keypad_get_pressmode (T_VOID)

获取按键模式（单按或者多按），多按模式只用于游戏中

Get keypad press mode single press or multiple press, now multiple press just use in game

返回值说明:

T_eKEY_PRESSMODE single press or multiple press //单按或者多按

eSINGLE_PRESS singel press //单按

eMULTIPLE_PRESS multiple press //多按

T_VOID keypad_init (T_fKEYPAD_CALLBACK callback_func,
T_U32 type_index,
const T_VOID * para
)

初始化键盘；在系统初始化中只需要调用一次，调用之前必须先调用 gpio_init()函数

Initialize keypad.

Only need to call one time during system init Function gpio_init() must be called before this function

参数说明:

[in] callback_func user defined Keypad callback function //用户定义的键盘回调函数

[in] type_index keypad type //键盘类型

[in] para keypad parameter //键盘参数

返回值说明:

T_VOID

T_S32 keypad_scan (T_VOID)

扫描键盘，或者当前按下的键盘的值；调用此函数之前必须先调用 keypad_init()函数

Anyka Confidential For BOMEI Only

Scan keypad and get the value of the keypad currently pressed

Function keypad_init() must be called before call this function

返回值说明:

T_S32

The pressed key's scan code //按下键的扫描模式

T_VOID keypad_set_delay (T_S32 keydown_delay,
T_S32 keyup_delay,
T_S32 keylong_delay,
T_S32 powerkey_long_delay,
T_S32 loopkey_delay
)

为各种键盘设置延迟值

Set all kinds of delay for keypad

参数说明:

[in] keylong_delay long key delay time (millisecond),must >0 //长按的延迟时间（必须大于 0 毫秒）

[in] keydown_delay long key delay time (millisecond),must >=0 //按下的延迟时间

[in] keyup_delay long key delay time (millisecond),must >=0 //按键弹起的延迟时间

[in] powerkey_long_delay long key delay time (millisecond),must >0 //电源键延迟时间

[in] loopkey_delay loop key delay time (millisecond),must >0 //回环键延迟时间

返回值说明:

T_VOID

T_BOOL keypad_set_pressmode (T_eKEY_PRESSMODE press_mode)

设置键盘按键模式，单按或者多按可选；目前多按仅在游戏中支持。

Set press mode for keypad single press or multiple press, now multiple press just use in game

参数说明:

[in] press_mode single press or multiple press //单按或者多按

返回值说明:

Anyka Confidential
BOMEI Use Only

T_BOOL

AK_TRUE,: set success //设置成功

AK_FALSE,: set failure //设置失败

2.14.3 配置说明

1、 按键的 GPIO 配置

3771 开发板配套的按键 是 4X3 矩阵的，需要用到 6 个 GPIO 口 3771 V1.1 版本配置可参考 2.11.3 章节 GPIO 口配置说明。

2、 按键添加

Keypad 根据扫描方式的不同划分不同类型：

a、矩阵式，M 个行 GPIO 和 N 个列 GPIO 对应 M*N 个 Key。

b、单 GPIO，每个 gpio 对应一个 Key。

c、 混合式，既有矩阵式又有单 gpio 式。

d、模拟键盘，同过 ad 输入，不同电压范围对应不同键值。

添加对一种新的类型的 Keypad 的支持，需要调用 keypad_reg_scanmode 函数向驱动库注册该类型 keypad 的初始化/获取键值等回调函数。对于不同扫描类型的键盘，通过如下结构，添加一款新的键盘只需实现对应的回调函数即可：

```
typedef struct
{
    T_VOID (*KeyPadInit)(T_f_H_KEYPAD_CALLBACK callback_func, const T_VOID *keypad_parm);//键盘的初试化函数

    T_S32  (*KeyPadScan)(T_VOID); //直接扫描键盘，得到当前按下按键的键值，只能得到按键的 id

    T_VOID (*KeyPadEnIntr)(T_VOID);//打开按键的中断

    T_VOID (*KeyPadDisIntr)(T_VOID);//关闭按键的中断

    T_eKEY_PRESSMODE (*GetMode)(T_VOID);//获取当前按键模式

    T_BOOL (*SetMode)(T_eKEY_PRESSMODE press_mode);//设置当前按键模式
```

Anyka Confidential For BOMEI Use Only

T_VOID (*SetDelay)(T_S32 keydown_delay, T_S32 keyup_delay, T_S32 keylong_delay, T_S32 powerkey_long_delay, T_S32 loopkey_delay);//调节按键参数

} T_KEYPAD_HANDLE;

3、 按键应用

使用按键前，必须得初始化：

T_VOID keypad_init(T_fKEYPAD_CALLBACK callback_func, T_U32 type_index, const T_VOID* para);

应用层可以调用以下接口获得按键值：

T_BOOL keypad_get_key(T_KEYPAD *key);

可以调用如下的两个接口使能或者关闭按键中断

T_VOID keypad_enable_intr(T_VOID);

T_VOID keypad_disable_intr(T_VOID);

2.15 Timer

2.15.1 功能概述

Timer 为定时器，功能是在指定的时间间隔内反复触发指定窗口的定时器事件。本节主要介绍定时器的相关接口使用说明。

2.15.2 接口说明

```
T_TIMER timer_start (T_TIMER_ID hardware_timer,
                     T_U32 milli_sec,
                     T_BOOL loop,
                     T_fTIMER_CALLBACK callback_func
                     )
```

启动定时器

Start timer.

当计数到时,调用定时器回调；定时器启动时要先调用 vtimer_init();

When the time reaches, the timer callback function will be called. Function vtimer_init() must be called before call this function

参数说明:

AnyKa Confidential For OEM Use Only

[in] hardware_timer hardware timer ID, uiTIMER0~uiTIMER3, cannot be uiTIMER4, because it is used for tick count //硬件定时器 ID，只能为 uiTIMER0~uiTIMER3；uiTIMER4 用于 Tick Count。

[in] milli_sec Specifies the time-out value, in millisecond. Caution, this value must can be divided by 20. //确定超时值，单位为毫秒。注意事项该值必须是 20 的倍数。

[in] loop loop or not //是否循环计数

[in] callback_func Timer callback function. If callback_func is not AK_NULL, then this callback function will be called when time reach. //定时计数满时,调用回调.

返回值说明:

T_TIMER timer ID user can stop the timer by this ID

ERROR_TIMER,: failed

T_VOID timer_stop (T_TIMER timer_id)

停止定时器；调用此函数之前必须先调用 vtimer_init()函数

Stop timer.

Function vtimer_init() must be called before call this function

参数说明:

[in] timer_id Timer ID, uiTIMER0~uiTIMER3, cannot be uiTIMER4, because it is used for tick count

返回值说明:

T_VOID

T_U32 get_tick_count (T_VOID)

从硬件定时器中获取毫秒 Tick 数

Get ms level tick count from hardware timer.

Tick count 调用此函数之前必须先调用 vtimer_init()函数。

Tick count value is calculated from timer5 Function vtimer_init() must be called before call this function

返回值说明:

T_U32

tick_count whose unit is ms //单位为毫秒（ms）

Anyka Confidential For ROMEI Use Only

T_U64 get_tick_count_us (T_VOID)

从硬件定时器中获取微秒 Tick 数

Get us level tick count from hardware timer.

调用此函数之前必须先调用 vtimer_init()函数。

Tick count value is calculated from timer5. Function vtimer_init() must be called before call this function

返回值说明:

T_U64

Return values:

tick_count whose unit is us //单位为微秒

T_VOID vtimer_free (T_VOID)

释放虚拟定时器和硬件定时器

Free virtual timer and hardware timer.

返回值说明:

T_VOID

T_U32 vtimer_get_cur_time (T_TIMER timerID)

获取定时器当前的值，单位为毫秒（ms）

Get Timer current value, count by ms.

调用此函数之前必须先调用 vtimer_init()函数

Function must called vtimer_init() before call this function

参数说明:

timerID [in] Timer ID

返回值说明:

T_U32 current value of that timer //定时器的当前值

T_U32 vtimer_get_time (T_TIMER timerID)

获取定时器的总延时，单位为毫秒。

Get Timer total delay, count by ms.

调用此函数之前必须先调用 vtimer_init()函数

Anyka Confidential For BOMEI Use Only

Function must called vtimer_init() before call this function

参数说明:

timerID [in] Timer ID

返回值说明:

T_U32 total delay of that timer //该定时器总延时

T_VOID vtimer_init (T_VOID)

初始化虚拟定时器和硬件定时器，然后打开硬件定时器中断系统

Initialize virtual timer and hardware timer. And then open the hardware timer interrupt;.

返回值说明:

T_VOID

T_TIMER vtimer_start (T_U32 milli_sec,
T_BOOL loop,
T_fVTIMER_CALLBACK callback_func
)

启动定时器

Start vtimer.

当计数到时，必须调用定时器回调函数。不管 loop 是 AK_TURE 还是 AK_FALSE，用户必须调用 vtimer_stop()函数来释放定时器 ID。调用此函数之前必须先调用 vtimer_init()函数。

When the time reaches, the vtimer callback function will be called. User must call function vtimer_stop() to free the timer ID, in spite of loop is AK_TRUE or AK_FALSE. Function must called vtimer_init() before call this function

参数说明:

milli_sec [in] Specifies the time-out value, in millisecond. Caution, this value must can be divided by 5. //定时数必须是 5 的倍数.

loop [in] oop or not //是否循环计数.

callback_func [in] Timer callback function. If callback_func is not AK_NULL, then this callback function will be called when time reaches. //定时器回调函数

返回值说明:

T_TIMER

Anyka Confidential For BOMEI Use

timer_ID user can stop the timer by this ID //通过该 ID 用户可终止相应的定时器

ERROR_TIMER start failed //启动失败

T_VOID vtimer_stop (T_TIMER timer_id)

终止定时器

Stop vtimer.

调用此函数之前必须先调用 vtimer_init()函数

Function must called vtimer_init() before call this function

参数说明:

timer_id [in] Timer ID

返回值说明:

T_VOID

T_U32 vtimer_validate_count (T_VOID)

获取未使用定时器数量

Get unused timer number.

调用此函数之前必须先调用 vtimer_init()函数

Function user should call vtimer_init() before calling this function

返回值说明:

T_U32 the number of unused timer //未使用定时器数量

Anyka Confidential For BOMEI Use Only

2.16 Camera

2.16.1 功能概述

本节主要介绍 camera 相关接口说明，其中包括 camera 参数设置、效果设置、模式设置以及中断设置等。

2.16.2 接口说明

T_BOOL cam_capture_JPEG (T_U8 * dstJPEG,
T_U32 * JPEGlength,
T_U32 timeout

```
                                    )
```

采集 JPEG 格式图像

Capture an image in JPEG format

参数说明:

[out] dstJPEG buffer to save the image data //存储图像数据的 buffer

[out] JPEGlength jpeg line length //JPEG 线长

[in] timeout time out value for capture //拍照超时值

返回值说明:

T_BOOL

AK_TRUE if success //成功

AK_FALSE if time out //超时

```
T_BOOL cam_capture_RGB ( T_U8 * dst,
                         T_U32 dstWidth,
                         T_U32 dstHeight,
                         T_U32 timeout
                        )
```

采集 RGB 格式图像

Capture an image in RGB format

参数说明:

[out] dst buffer to save the image data //存储图像数据的 buffer

[in] dstWidth desination width, the actual width of image in buffer //目标图像的宽度，buffer 中图像的实际宽度

[in] dstHeight desination height, the actual height of image in buffer//目标图像的高度，buffer 图像的实际高度

[in] timeout time out value for capture// 超时值

返回值说明:

T_BOOL

AK_TRUE if success

AK_FALSE if time out

Anyka Confidential For BOMEI

T_BOOL cam_capture_YUV (T_U8 * dstY,

T_U8 * dstU,

T_U8 * dstV,

T_U32 dstWidth,

T_U32 timeout

)

采集 YUV420 格式图像

Capture an image in YUV420 format

参数说明:

[out] dstY Y buffer to save the image data //Y 分量缓存

[out] dstU U buffer to save the image data // U 分量缓存

[out] dstV V buffer to save the image data // V 分量缓存

[in] dstWidth desination width, the actual width of image in buffer //目标图像宽度

[in] dstHeight desination height, the actual height of image in buffer //目标图像高度

[in] timeout time out value for capture //超时值

返回值说明:

T_BOOL

AK_TRUE if success

AK_FALSE if time out

T_VOID cam_close (T_VOID)

关闭 camera

Close camera

返回值说明:

T_VOID

T_CAMERA_TYPE cam_get_type (T_VOID)

获取摄像头类型

Get camera type

返回值说明:

Anyka Confidential For BOMEI Use Only

T_CAMERA_TYPE

camera type defined by mega pixels //其类型由像素定义

T_BOOL cam_open (T_VOID)

打开摄像头，此函数应该在摄像头复位初始化后使用

Open camera, should be done the after reset camera to initialize

返回值说明:

T_BOOL

AK_TRUE if successed //打开成功

AK_FALSE if failed //打开失败

T_VOID cam_set_feature (T_CAMERA_FEATURE feature_type,
T_U8 feature_setting
)

设置摄像头效果参数

Set camera feature

参数说明:

[in] feature_type camera feature type,such as night mode,AWB,contrast.etc.//比如夜间模式，AWB，对比度等

[in] feature_setting feature setting//效果参数设置

返回值说明:

T_VOID

T_U32 cam_set_framerate (float framerate)

手动设置摄像头帧率

Set camera frame rate manually.

参数说明:

[in] framerate camera output framerate //摄像头输出帧率

返回值说明:

T_U32

Anyka Confidential For BOMEI Use Only

0 if error mode //错误模式

1 if success //设置成功

T_BOOL cam_set_to_cap (T_U32 srcWidth,
T_U32 srcHeight
)

从预览模式切换成拍照模式

Wwitch from preview mode to capture mode

参数说明:

[in] srcWidth window width //窗口宽度

[in] srcHeight window height //窗口高度

返回值说明:

T_BOOL

AK_TRUE if successed

AK_FALSE if failed

T_BOOL cam_set_to_prev (T_U32 srcWidth,
T_U32 srcHeight
)

切换到预览模式

Switch camera reg to preview mode

参数说明:

[in] srcWidth window width

[in] srcHeight window height

返回值说明:

T_BOOL

AK_TRUE if successed

AK_FALSE if failed

Anyka Confidential For BOME Use Only

T_BOOL cam_set_to_record (T_U32 srcWidth,
T_U32 srcHeight
)

切换成录像模式

Switch camera to record mode

参数说明:

[in] srcWidth window width

[in] srcHeight window height

返回值说明:

T_BOOL

AK_TRUE if successed

AK_FALSE if failed

T_S32 cam_set_window (T_U32 srcWidth,
T_U32 srcHeight
)

设置摄像头窗口

Set camera window

参数说明:

[in] srcWidth window width //录像窗口宽度

[in] srcHeight window height //录像窗口高度

返回值说明:

T_S32

0 if error mode //错误模式

1 if success //成功

-1 if fail //失败

注意事项:

此函数的尺寸参数是为了从 camera 输出的图象中“挖取”一个窗口， “挖取”功能的具体实现由当前 AK 芯片的功能决定。有关该函数的调用，请用户参见 2.15.3“调用流程”。

Anyka Confidential For BOMEI Use Only

T_BOOL camstream_change (T_U32 dstWidth,
T_U32 dstHeight,
T_CAMERA_BUFFER * YUV1,
T_CAMERA_BUFFER * YUV2,
T_CAMERA_BUFFER * YUV3
)

更改摄像头配置

Change camera configuration

参数说明:

[in] dstWidth camera dest width //目标图像宽度

[in] dstHeight camera dest height //目标图像高度

[in] YUV1 store a frame YUV buffer, can be NULL //存储 YUV 数据缓存

[in] YUV2 store a frame YUV buffer, can be NULL //存储 YUV 数据缓存

[in] YUV3 store a frame YUV buffer, can be NULL //存储 YUV 数据缓存

返回值说明:

T_BOOL

AK_TRUE change successfully //操作成功

AK_FALSE change unsuccessfully //操作失败

T_CAMERA_BUFFER* camstream_get (T_VOID)

获取一帧数据

Get a frame data

返回值说明:

T_CAMERA_BUFFER*

Pointer of current frame data //当前帧数据指针

T_BOOL camstream_init (T_U32 dstWidth,
T_U32 dstHeight,
T_CAMERA_BUFFER * YUV1,
T_CAMERA_BUFFER * YUV2,

Anyka Confidential For BOMEI Use Only

T_CAMERA_BUFFER * YUV3

)

初始化摄像头中断模式

Initialize camera interrupt mode

参数说明:

[in] dstWidth camera dest width

[in] dstHeight camera dest height

[in] YUV1 store a frame YUV buffer, can be NULL

[in] YUV2 store a frame YUV buffer, can be NULL

[in] YUV3 store a frame YUV buffer, can be NULL

返回值说明:

T_BOOL

AK_TRUE initialize successfully

AK_FALSE initialize unsuccessfully

T_BOOL camstream_ready (T_VOID)

准备好一帧数据.

A frame is ready

返回值说明:

T_BOOL

AK_FALSE,: no data

AK_TRUE,: a frame is ready

T_VOID camstream_resume (T_VOID)

恢复采集数据.

Resume camera interface, start accept frame

返回值说明:

T_VOID

Anyka Confidential For BOMEI Use Only

T_VOID camstream_set_callback (T_fCAMSTREAMCALLBACK callback_func)

设置通知回调函数

Set notify callback function

参数说明:

[in] callback_func callback function

返回值说明:

T_VOID

T_VOID camstream_stop (T_VOID)

停止摄像头中断模式

Stop camera (interrupt mode)

返回值说明:

T_VOID

T_BOOL camstream_suspend (T_VOID)

暂停 camera 接口，只接受当前帧，不接受其它帧

Suspend camera interface, only accept current frame, other not accept

返回值说明:

T_BOOL

Return values:

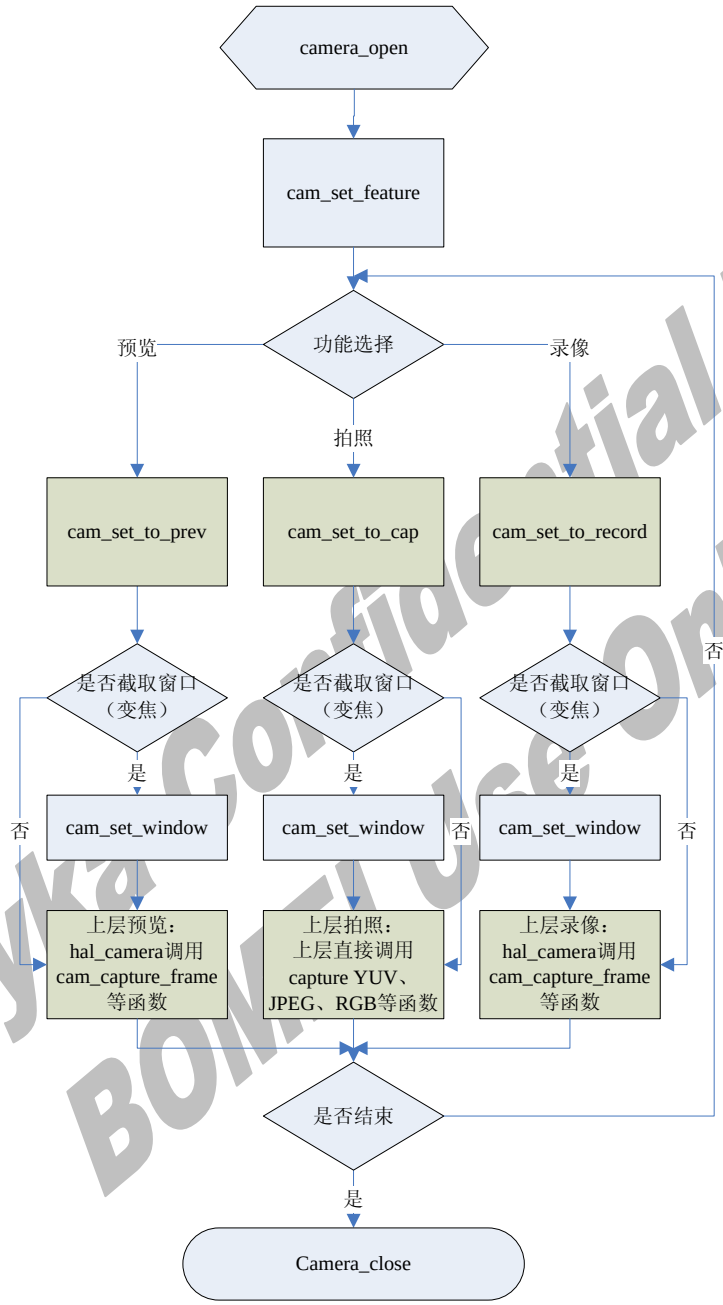
AK_FALSE,: suspend failed //暂停失败

AK_TRUE,: suspend successful //暂停成功

Anyka Confidential For BOMEI Use Only

2.16.3 调用流程

Camera 使用的接口调用基本流程图如下所示。



Anyka Confidential For BOM Use Only

图 2-1 Camera 函数调用流程

2.17 Sound Driver

2.17.1 功能概述

音频驱动主要是为设备上的多媒体播放提供音频驱动接口。音频驱动主要包括 ADC 驱动、DAC 驱动、I2S 发送驱动和 I2S 接收驱动等。以下为音频驱动的接口使用说明。

2.17.2 接口说明

T_VOID sound_cleanbuf (T_SOUND_DRV * handler)

清除音频 buffer

Clean sound buffer

参数说明:

[in] handler handler of the sound device //音频设备处理器

返回值说明:

T_VOID

T_BOOL sound_close (T_SOUND_DRV * handler)

关闭一个音频设备

Close a sound device.

参数说明:

[in] handler handler of the sound device

返回值说明:

T_BOOL

AK_TRUE close successful //关闭成功

AK_FALSE close failed //关闭失败

```
T_SOUND_DRV* sound_create (SOUND_DRIVER driver,
                           T_U32  OneBufSize,
                           T_U32  DABufNum,
                           T_fSOUND callback
                          )
```

Anyka Confidential For BOMEI Use Only

创建一个音频驱动，分配音频 buffer 并初始化 L2 内存。buffer 中完成一个读操作或者写操作后会调用该函数

Create a sound driver, it will malloc sound buffer and init L2

The callback function will be called when a read or write buffer complete, it can be AK_NULL and do nothing.

参数说明:

[in] driver sound driver, refer to SOUND_DRIVER

[in] OneBufSize buffer size //buffer 大小

[in] DABufNum buffer number //buffer 数量

[in] callback callback function or AK_NULL//回调函数或 AK_NULL

返回值说明:

T_SOUND_DRV

AK_NULL created failed //创建失败

T_VOID sound_delete (T_SOUND_DRV * handler)

删除音频驱动，释放音频 buffer

Delete sound driver and free sound buffer

参数说明:

[in] handler handler of the sound device

T_BOOL sound_endbuf (T_SOUND_DRV * handler,
T_U32 len
)

设置下一个数据 buffer 指针；在调用 sound_getbuf 函数、结束音频数据操作之后调用该函数

Set one buffer end

After calling sound_getbuf and finish the operation of sound data,call this function

参数说明:

[in] handler handler of the sound device

[in] len buffer len(use for write) //buffer 长度（用于写操作）

返回值说明:

Anyka Confidential For OMEI Use Only

T_BOOL

AK_TRUE successful //操作成功

AK_FALSE longer than one buffer's len //长度超过一个 buffer 的长度

T_BOOL sound_getbuf (T_SOUND_DRV * handler,
T_VOID ** pbuf,
T_U32 * len
)

获取 buffer 地址和 buffer 长度，可用于存储或接收音频数据

Get buffer address and buffer len, which can be used to fill or retrieve sound data

参数说明:

[in] handler handler of the sound device

[out] pbuf return buffer address or AK_NULL //返回 buffer 地址或者 AK_NULL

[out] len return buffer len or 0 //返回 buffer 长度或者 0

返回值说明:

T_BOOL

AK_TRUE get buffer successful

AK_FALSE get buffer failed

注意事项:

如果 sound_create 创建失败或者没有 buffer 返回，该函数将会返回失败

If sound_create failed or no buffer to return, it will return failed

T_U32 sound_getnum_fullbuf (T_SOUND_DRV * handler)

获取存储接收音频数据的 buffer 数

Get the number of buffers which have been filled or retrieved sound data

参数说明:

[in] handler handler of the sound device

返回值说明:

T_U32

value the value will from 0 to the number when create a sound set //获取音频数据 buf 数目

AnyKa Confidential For BOMEI Use Only

T_BOOL sound_open (T_SOUND_DRV * handler)

打开一个可用音频设备

Open a sound device and it can be used

参数说明:

[in] handler handler of the sound device

返回值说明:

T_BOOL

AK_TRUE open successful

AK_FALSE open failed

```
T_BOOL sound_realloc ( T_SOUND_DRV * handler,
                       T_U32 OneBufSize,
                       T_U32 DABufNum,
                       T_fSOUND callback
                     )
```

为音频驱动重新分配 buffer

Reallocate buffer for giving sound driver

参数说明:

[in] handler handler of the sound device

[in] OneBufSize buffer size //buffer 大小

[in] DABufNum buffer number //buffer 数

[in] callback callback function or AK_NULL

返回值说明:

T_BOOL

AK_TRUE realloc successful //操作成功

AK_FALSE realloc failed //操作失败

```
T_BOOL sound_setinfo ( T_SOUND_DRV * handler,
                       SOUND_INFO * info
                     )
```

设置音频设备的音频采样率，信道，每个 sample 的 bit 数

Anyka Confidential For BOMEI Use Only

Set sound sample rate, channel, bits per sample of the sound device.

参数说明:

[in] handler handler of the sound device

[in] info refer to SOUND_INFO

返回值说明:

T_BOOL

AK_TRUE set successful

AK_FALSE set failed

2.18 SPI Flash

2.18.1 功能概述

串行外设接口 (Serial Peripheral Interface--SPI) 总线系统是一种同步串行外设接口，它可以使 MCU 与各种外围设备以串行方式进行通信以交换信息。SPIFlash 为 SPI 接口的主要应用之一。

2.18.2 接口说明

T_BOOL spi_flash_erase (T_U32 sector)

擦除 spiflash 的一个分区

Erase one sector of spiflash

参数说明:

sector [in] sector number, unit is erase_size, refer to T_SFLASH_PARAM //分区号。单位为 erase_size，参见 *T_SFLASH_PARAM* 定义

返回值说明:

T_BOOL

T_U32 spi_flash_getid (T_VOID)

获取 spiflash ID 号

Get spiflash id

返回值说明:

T_U32

Anyka Confidential For BOMEI Use Only

T_U32 spiflash id

T_BOOL spi_flash_init(T_eSPI_ID spi_id, T_eSPI_BUS bus_width)

初始化 spiflash

spi flash init

参数说明:

[in] spi_id ： spi id, can be spi0 or spi1

[in]bus_width：SPI 总线的宽度，1、2 或者 4。

注意：总线的宽度选择是否有效，还需要结合 spiflash 的参数成员 flag，Spiflash 参数成员的设置请参考 spi_flash_set_param 这个函数。

返回值说明:

T_BOOL

T_BOOL spi_flash_read (T_U32 page,
T_U8 * buf,
T_U32 page_cnt
)

从 spiflash 的一页（page）中读取数据

Read data from one page of spiflash

参数说明:

page [in] page number //要读的起始页数

buf [in] buffer to store read data //存储读取数据的 buffer

page_cnt [in] the page count to read //页个数

返回值说明:

T_BOOL

T_VOID spi_flash_set_param (T_SFLASH_PARAM * sflash_param)

设置串行 flash 参数

Set param of serial flash

参数说明:

sflash_param [in] serial flash param //串行 flash 参数

Anyka Confidential For BOMEI Use

返回值说明:

T_VOID

```
T_BOOL spi_flash_write ( T_U32 page,
                         T_U8 * buf,
                         T_U32 page_cnt
                       )
```

写数据到 spiflash 的一页（page）里

Write data to one page of spiflash

参数说明:

page [in] page number

buf [in] buffer to be write

page_cnt [in] the page count to write

返回值说明:

T_BOOL

2.19 USB

2.19.1 功能概述

USB 为 Universal Serial BUS（通用串行总线）的缩写，而其中文简称为“通串线”，是一个外部总线标准，用于规范电脑与外部设备的连接和通讯。

本节介绍 USB Host 设置接口、USB debug 调试设置以及 UVC（USB Video Class 即插即用）设置等。UVC 控制定义如下：

```
Enum T_eUVC_CONTROL
```

定义 UVC 控制操作

Define uvc controls implemented.

Enumerator:

UVC_CTRL_BRIGHTNESS uvc device brightness control selector //UVC 设备亮度控制区

UVC_CTRL_CONTRAST uvc device contrast control selector //UVC 设备对比度控制区

UVC_CTRL_SATURATION uvc device saturation control selector //UVC 设备饱和度控制区

UVC_CTRL_RESOLUTION uvc device resolution control selector //UVC 设备分辨率控制区

UVC_CTRL_NUM uvc device control number //UVC 设备控制数

Anyka Confidential For BOMEI Use Only

Enum T_eUVC_STREAM_FORMAT

定义 UVC 帧格式

Define uvc frame format.

Enumerator:

UVC_STREAM_YUV uncompressed format //未压缩格式

UVC_STREAM_MJPEG motion jpeg format //MJPEG 格式

2.19.2 接口说明

T_VOID udisk_host_close (T_VOID)

关闭 U 盘主设备

Udisk host close function.

该函数在弹出 U 盘退出 U 盘主设备时由应用层调用

This function is called by application level when eject the udisk and exit the udisk host.

返回值说明:

T_VOID

T_VOID udisk_host_get_lun_info (T_U32 LUN,
T_pH_UDISK_LUN_INFO disk_info
)

获取逻辑单元号描述符

Get a logic unit number descriptor

参数说明:

[in] LUN Index of logic unit. //逻辑单元索引

[out] disk_info The information of the lun //逻辑单元号信息

返回值说明:

T_VOID.

T_U8 udisk_host_get_lun_num (T_VOID)

获取 U 盘所有的逻辑单元号数量

Get disk all logic unit number

Confidential For BOM Use Only

返回值说明:

T_U8

Total number of logic unit. //所有逻辑单元号总数

T_BOOL udisk_host_init (T_U32 mode)

初始化 U 盘主设备功能

Initialize udisk host function

分配 U 盘主设备 buffer，初始化数据结构，注册回调函数，打开 USB 控制器...

Allocate udisk host buffer, initialize data strcut, and register callback, open usb controller and phy.

参数说明:

[in] mode usb mode 1.1 or 2.0 //USB 模式 1.1 或者 2.0

返回值说明:

T_BOOL

AK_FALSE init failed //初始化失败

AK_TURE init successful //初始化成功

```
T_U32 udisk_host_read ( T_U32 LUN,
                        T_U8 data[],
                        T_U32 sector,
                        T_U32 size
                      )
```

USB 主设备从逻辑单元中读取分区数据

USB host read sector from logic unit

参数说明:

[in] LUN index of logic unit. //逻辑单元索引

[in] data Buffer to store data //存储数据 Buffer

[in] sector Start sector to read //读取的起始 sector

[in] size Total sector to read //读取的总 sector 大小

返回值说明:

T_U32

Confidential For Public Use Only

Really total sector have been read. //已经读取的总 sector

```
T_VOID udisk_host_set_callback (T_pfUDISK_HOST_CONNECT connect_callback,
                                T_pfUDISK_HOST_DISCONNECT disconnect_callback
                               )
```

U 盘主设备设置应用层回调函数

Udisk host set application level callback.

该函数必须在 U 盘主设备初始化之后由应用层调用

This function must be called by application level after udisk host initialization.

参数说明:

[in] connect_callback Application level callback //连接应用层回调函数

[in] disconnect_callback Application level callback //断开应用层回调函数

返回值说明:

T_VOID

```
T_U32 udisk_host_write ( T_U32 LUN,
                         T_U8 data[],
                         T_U32 sector,
                         T_U32 size
                       )
```

USB 主设备写入 sector 到逻辑单元

USB host write sector to logic unit

参数说明:

[in] LUN Index of logic unit. //逻辑单元索引

[in] data The write data //写入数据

[in] sector Start sector to write //写入起始 sector

[in] size Total sectors to write //写入总 sector

返回值说明:

T_U32

Really total sectors have been written. //已写总 sector 数

Anyka Confidential For BOM Only

T_VOID usbdebug_disable (T_VOID)

关闭 USB 从模式的 Debug 功能

Disable usb debug in usb slave mode

返回值说明:

T_VOID

T_BOOL usbdebug_enable (T_VOID)

使能（打开）USB 从模式 debug 功能

Enable usb debug in usb slave mode

返回值说明:

T_BOOL

Return values:

AK_FALSE means failed //操作失败

AK_TURE means successful //操作成功

T_U32 usbdebug_getstring (T_U8 * str,
T_U32 len
)

从 USB 获取字符串

Get string from usb

参数说明:

[out] str buffer to store input string //存储输入字符串 Buffer

[in] len buffer size //buffer 大小

返回值说明:

T_U32

void usbdebug_printf (T_U8 * str,
T_U32 len
)

从 USB 口打印

Print to usb

Anyka Confidential For BOMEI Use Only

参数说明:

[in] str string to be print //要打印的字符串

[in] len string length //字符串长度

返回值说明:

T_VOID

T_USB_EXT_CMD_HANDLER* usb_ext_probe (T_U8 ext_cmd)

扩展 USB 处理回调函数

USB probe pointer.

参数说明:

[in] ext_cmd USB ext cmd //USB 扩展 CMD

返回值说明:

T_USB_EXT_CMD_HANDLER: USB ext cmd pointer //USB 扩展的 CMD

T_BOOL usb_ext_scsi_reg (T_U8 ext_cmd,
T_USB_EXT_CMD_HANDLER * handler
)

注册 USB 扩展 CMD 回调

register usb extcmd

参数说明:

[in] ext_cmd USB ext cmd

[in] handler extcmd pointer //扩展 CMD 指针

返回值说明:

T_BOOL

注意事项事项:

Param [in] T_USB_EXT_CMD_HANDLER *handler is global variable pointer //参数

*T_USB_EXT_CMD_HANDLER *handle*r 为全局变量指针

T_BOOL usbdisk_addLUN (T_LUN_INFO * pAddLun)

USB 从设备磁盘 增加一个逻辑单元号

Anyka Confidential For BOMEI Only

USB slave bulk disk add a lun

该函数在主设备挂载 U 盘时调用

This function is called when host is mounting usb disk

参数说明:

[in] pAddLun struct of lun information. //逻辑单元号的信息结构

返回值说明:

T_BOOL

AK_FALSE means failed //操作失败

AK_TURE means successful //操作成功

T_BOOL usbdisk_changeLUN (T_LUN_INFO * pChgLun)

USB 从设备更换逻辑单元号

USB slave bulk disk change lun

当检测到 SD 卡时更换成 SD 卡的逻辑单元号

When SD card is detected, change the lun for sd card

参数说明:

[in] pChgLun struct of lun information. //逻辑单元号信息结构

返回值说明:

T_BOOL

AK_FALSE means failed

AK_TURE means successful

T_BOOL usbdisk_init (T_U32 mode)

初始化 U 盘功能

init USB disk function

初始化 U 盘 buffer，创建 HISR（高级中断），创建 U 盘任务，创建消息列队，创建事件组。

Initialize usb disk buffer, creat HISR, creat usb disk task, creat message queue, creat event group

参数说明:

[in] mode usb mode 1.1 or 2.0 //USB 模式 1.1 或者 2.0

Anyka Confidential For BOMEI Use Only

返回值说明:

T_BOOL

AK_FALSE init failed //初始化失败

AK_TURE init successful //初始化成功

T_VOID usbdisk_mboot_init (T_U32 mode)

初始化 Mass Boot

Mass boot init

在 USB 1.1 中 massboot 没有运行

Udisk reset, configure ok, this function must be called because mass boot at usb1.1 will not run enum

参数说明:

[in] mode usb mode 1.1 or 2.0

返回值说明:

T_BOOL

AK_FALSE init failed

AK_TURE init successful

T_BOOL usbdisk_mboot_set_cb (T_fMBOOT_HANDLE_CMD hnd_cmd,
T_fMBOOT_HANDLE_SEND hnd_send,
T_fMBOOT_HANDLE_RCV hnd_rcv
)

设置 procduce 的回调

Set procduce callback

在 procduce 中有用

Used by produce

参数说明:

[in] hnd_cmd handle cmd callback //USB CMD

[in] hnd_send handle send callback //发送回调

[in] hnd_rcv handle receive callback //接收回调

返回值说明:

ka Confidential For Use Only

T_BOOL

AK_FALSE set failed

AK_TURE set successful

T_VOID usbdisk_proc (T_VOID)

进入 USB 处理任务

Enter udisk task, poll udisk message

必须在启 USB 时才会被调用.

This function is added for bios version,and must be call after usbdisk_start

返回值说明:

T_VOID

T_BOOL usbdisk_set_str_desc (T_eSTR_DESC index,
T_CHR * str
)

设置字符串描述符

Initialize string descriptor with reference to device desc

这函数必须在启动 USB 后调用

The str is truncated to 10 characters or less, this func may be called before usbdisk_start

参数说明:

[in] index type of string descriptor //字符串描述符索引值

[in] str the start address of stirng //具体的字符串

返回值说明:

T_BOOL

AK_FALSE set failed

AK_TURE set successful

T_BOOL usbdisk_start (T_VOID)

启动 U 盘功能；该函数必须在 usbdisk_init 函数之后调用

Start USB disk function, this function must be call after usbdisk_init

分配 L2buffer，打开 USB 控制器，设置 U 盘回调函数，注册中断处理函数

Anyka Confidential For BOMEI Use Only

Allocate L2 buffer, open usb controller, set usb disk callback, and register interrupt process function

返回值说明:

T_BOOL

AK_FALSE means failed

AK_TURE means successful

T_VOID usbdisk_stop (T_VOID)

关闭 U 盘功能

Disable USB disk function.

关闭 USB 控制器，终止 U 盘任务，释放 buffer，释放数据结构

Close usb controller, terminate usb disk task, free buffer, free data structure.

返回值说明:

T_VOID

T_BOOL usb_detect (T_VOID)

检测是否有 USB 线插入

Detect whether USB cable is inserted or not

返回值说明:

T_BOOL

AK_TRUE usb cable in //USB 线插入

AK_FALSE usb cable not in //USB 线未插入

T_U8 usb_slave_getstate (T_VOID)

获取 USB 从设备状态

Get usb slave state.

返回值说明:

T_U8

USB_OK usb config ok,can transmit data //USB 配置 OK 可以传输数据

USB_ERROR usb error //USB 错误

USB_SUSPEND usb suspend by pc,can't to use //PC 中断 USB，USB 无法使用

Anyka Confidential For BOMEI Use Only

USB_NOTUSE usb close //USB 已关闭

USB_CONFIG usb config by pc //PC 配置 USB

T_VOID usb_slave_set_state (T_U8 stage)

设置 USB 从设备状态

Set USB slave state.

参数说明:

[in] stage T_U8.

返回值说明:

T_VOID

T_VOID uvc_get_frame_res (T_pUVC_FRAME_RES pFrameRes,
T_U32 FrameId
)

获取 UVC 帧分辨率

Get uvc frame resolution

参数说明:

[in] pFrameRes uvc frame resolution struct //UVC 帧分辨率结构

[in] FrameId uvc frame id //UVC 帧 ID

返回值说明:

T_VOID

T_BOOL uvc_init (T_U32 mode,
T_U8 format
)

初始化 UVC 描述符、YUVbuffer 以及 Map msg。

Initialize uvc descriptor,yuv buffer,and map msg.

参数说明:

[in] mode usb2.0 or usb1.0

[in] format The UVC frame format //UVC 帧格式

Anyka Confidential For BOMEI Use Only

返回值说明:

T_BOOL

AK_FALSE means failed

AK_TURE means successful

T_BOOL uvc_init_desc (T_VOID)

初始化 UVC 描述符；该函数必须在 uvc_set_ctrl 函数之后调用

Initialize uvc descriptor, MUST be called after uvc_set_ctrl.

返回值说明:

T_BOOL

AK_FALSE means failed

AK_TURE means successful

```
T_U32 uvc_parse_yuv ( T_U8 * pYUV,
                      T_U8 * y,
                      T_U8 * u,
                      T_U8 * v,
                      T_U32 width,
                      T_U32 height,
                      T_U8 yuv_format
                    )
```

解析 UVC 命令

USB Video Class parses yuv function.

参数说明:

[out] pYUV The buffer to store the YUV frame //存储 YUV 帧 buffer

[in] y The original y param addr //原始 Y 参数地址

[in] u The original u param addr //原始 U 参数地址

[in] v The original v param addr //原始 V 参数地址

[in] width The width of the yuv frame // YUV 帧宽度

[in] height The height of the yuv frame //YUV 帧高度

[in] yuv_format The format of yuv,yuv422 or yuv420 //YUV 格式（YUV/YUV422/YUV420）

返回值说明:

T_U32

The size of the yuv frame //YUV 帧大小

```
T_U32 uvc_payload ( T_U8 * pPayload,
                    T_U8 * pData,
                    T_U32 dwSize
                  )
```

加载 UVC 数据

USB Video Class payload packing function.

参数说明:

[out] pPayload The buffer to store the payload //加载缓存

[in] pData The original frame data to be packed //要打包的原始帧数据

[in] dwSize The size of the frame //帧大小

返回值说明:

T_U32

The size of the payload //加载 UVC 的长度

```
T_BOOL uvc_send ( T_U8 * data_buf,
                  T_U32 length
                )
```

通过 USB 发送帧数据

Send frame data via usb.

参数说明:

[in] data_buf : buffer to be send. //要发送的缓存

[in] length,: length of the buffer //缓存长度

返回值说明:

T_BOOL

AK_FALSE means failed

AK_TURE means successful

Anyka Confidential For Use Only

T_VOID uvc_set_callback (T_pUVC_VC_CTRL_CALLBACK vc_ctrl_callback,
T_pUVC_VS_CTRL_CALLBACK vs_ctrl_callback,
T_pUVC_FRAME_SENT_CALLBACK frame_sent_callback
)

设置 UVC 回调函数

Set UVC callback

该函数须在 uvc 初始化成功之后由应用层调用，来设置控制回调函数。

This function is called by application level to set control callback after uvc_init successful.

参数说明:

[in] vc_ctrl_callback implement video control interface controls //VC 控制回调

[in] vs_ctrl_callback implement video stream interface controls // VS 控制回调

[in] frame_sent_callback used to notify that a frame was sent completely //用于通知一帧数据已经完全发送

返回值说明:

T_VOID

T_BOOL uvc_set_ctrl (T_eUVC_CONTROL dwControl,
T_U32 ulMin,
T_U32 ulMax,
T_U32 ulDef,
T_U32 unRes
)

UVC 启动高级控制效果（设置控制效果如亮度、对比度、饱和度。该函数应在 uvc 初始化成功后调用）

USB Video Class setup the advanced control features.

参数说明:

[in] dwControl The advanced feature control selector //高级效果控制选择器

[in] ulMin The min value //最小值

[in] ulMax The max value //最大值

[in] ulDef The def value //默认值

Confidential For Device Only

[in] unRes The res value //参考值

返回值说明:

T_BOOL

AK_FALSE means failed //操作失败

AK_TURE means successful //操作成功

T_BOOL uvc_start (T_VOID)

启动 UVC

Start UVC.

返回值说明:

T_BOOL

T_VOID uvc_stop (T_VOID)

停止 UVC

Stop UVC.

返回值说明:

VOID

Anyka Confidential For BOMEI Use Only

2.20 MAC

媒体访问控制（Media Control Access, MAC）对应 OSI 标准的数据链路层，集成到 3C 的芯片内部，其与 PHY 之间通过标准的 MII (Medium Independent Interface)接口进行通信。

2.20.1 数据结构

T_MAC_STRUCT

typedef struct tagMAC_STRUCT

{

T_U8 gpio_pwr; //phy: 所用的 power gpio

T_U8 gpio_rst; // phy 所用的 reset gpio

T_MAC_DMA mac_dma; //记录 MAC 控制器使用的一些内存地址

```
    T_MAC_INFO mac_info ; //mac 控制器的一些信息
    T_hFreq freq_handle; //频率管理用
    fMAC_RECV_CALLBACK recv_cbk; //接受回调函数
    fMAC_STATUS_CALLBACK status_cbk; //状态回调函数
}
T_MAC_STRUCT;

T_MAC_DMA
typedef struct tagMAC_DMA
{
    T_U8 *tpd_ring_base;  //tpd 基址
    T_U8 *tpd_fifo_base; //tx fifo 基址
    T_U32 tpd_producer_index; //tx fifo 生产者地址
    T_U32 tpd_consumer_index; //tx fifo 消费者地址
    T_U8 *rfd_ring_base; //rfd 基址
    T_U8 *rrd_ring_base; //rrd 基址
    T_U8 *rfd_fifo_base; //rx fifo 基址
    T_U32 rfd_producer_index; //rx fifo 生产者地址
    T_U32 rfd_consumer_index; //rx fifo 消费者地址
    T_U32 rrd_producer_index; //rrd 生产者地址
    T_U32 rrd_consumer_index; //rrd 消费者地址
}
T_MAC_DMA;
```

Anyka Confidential For BOMEI Use Only

2.20.2 接口说明

```
T_BOOL mac_init(T_U8 *mac_addr,
                fMAC_RECV_CALLBACK recv_cbk,
                fMAC_STATUS_CALLBACK status_cbk,
                T_U8 gpio_pwr,
                T_U8 gpio_rst
```

)

初始化 MAC 功能。

参数说明：

[in] mac_addr：48 bit 的 mac 地址

[in]recv_cbk：接收数据的回调

[in]status_cbk：连接状态回调

[in]gpio_pwr：phy 的 power gpio 号

[in]gpio_rst,：phy 的 reset gpio 号

返回值说明：

T_BOOL

AK_TRUE：初始化成功

AK_FALSE：初始化失败

T_BOOL mac_close(T_VOID)

卸载 MAC 模块

参数说明：无

返回值说明：

T_BOOL

AK_TRUE：成功

AK_FALSE: 失败

T_U32 mac_send(T_U8 *data, T_U32 data_len);

发送数据

参数说明：

[in]data,：待发送的数据 buffer 地址

[in]data_len：待发送的数据长度

返回值说明：

T_U32

返回发送的数据长度

Anyka Confidential For BOMEI Use Only

T_VOID mac_phy_mcu_clk_25M(T_U8 pin)

设置芯片内部输出时钟的引脚

参数说明：

[in]pin：用于 25MHz 时钟输出的引脚号

返回值说明：无

2.21 watchdog

2.21.1 功能概述

Watchdog 用于防止程序进入死循环。当启动 watchdog 后，必须在规定的时间内进行喂狗，否则超时后，芯片会重启。

2.21.2 接口说明

T_VOID watchdog_timer_feed (T_VOID)

喂看门狗

Feed watch dog

参数说明：无

返回值说明:

T_VOID

注意事项:

This function must be called periodically, otherwise watchdog will expired and reset. //该函数必须定期调用，否则看门狗将会过期复位。

T_VOID watchdog_timer_start (T_U32 feed_time)

启动看门狗功能

Watch dog function start

参数说明：

[in] feed_time ：设定喂狗的时间（单位：ms），喂狗的时间必须小于这个设定值。

最大为 171798ms

返回值说明:

Anyka Confidential For BOMEI Use Only

T_VOID

T_VOID watchdog_timer_stop(T_VOID)

停止看门狗功能

Watch dog function stop

参数说明：无

返回值说明:

T_VOID

2.22 软重启

2.22.1 功能概述

软重启指系统在运行中，通过软件调用 soft_reset 函数，实现系统重新启动。

2.22.2 接口说明

T_VOID soft_reset(T_VOID)

软件重启系统

参数说明：无

返回值说明:

T_VOID

2.23 电压校准函数

2.23.1 功能概述

电压校准函数是要校准芯片内部 AD/DA 的基准电压，该基准电压会对模式按键和电池检测产生直接的影响。

2.23.2 接口说明

T_U32 Efuse_Rd(T_VOID)

校准芯片内部 AD/DA 的基准电压

参数说明：无

返回值说明:

Anyka Confidential For BOMEI Use Only

T_U32

“0”表示读取校准寄存器失败。

“~0”表示读取校准寄存器成功。

“bit7=1”表示芯片没烧录校准熔丝。

“bit7=0 && ~0”表示电压校准成功。

3 常用驱动修改方法

本章主要介绍本平台常用驱动的修改方法。

3.1 LCD

LCD 的相关接口，包括 LCD 初始化、LCD 打开和关闭、LCD 旋转和 LCD 刷新等，最终都会调用到具体的某款 LCD 的驱动函数，这些驱动函数定义与接口在同一个文件中，这个文件称为这款 LCD 的驱动文件或驱动。

LCD 驱动文件的格式基本如下：

```
/**
 * @FILENAME: lcd_xxx_xxxxx.c
 * @BRIEF
 * @AUTHOR
 * @DATE
 * @VERSION
 * @REF
 * @Note:
 */
#include "anyka_types.h"
#include "platform_hd_config.h"
#include "drv_lcd.h"

#ifdef USE_LCD_XXX_XXXXXX            //use to compile this file or not

#define LCD_ID XXXX                  //LCD id


/*macro & variable define*/

#if THIS_IS_A_DRIVER_FOR_RGB_LCD
/*T_RGBLCD_INFO JUST FOR RGB LCD*/
```

Anyka Confidential For BOMEI Use Only

```
    static T_RGBLCD_INFO SUPPORT_RGB_XXXXX_TABLE[] =
    {
        INTERLACE,      //Interlace(1) or progress(0)
        PHVG_POL,       //PCLK(bit3)Hsync(bit2), Vsync(bit1)
                        //Gate(bite0) polarity
        RGBorGBR,       //0=rgb,1=gbr
        THLEN,          //horizontal cycle, Unit PCLK
        THD,            //horizontal display period, Unit PCLK
        THF,            //Horizontal Front porch, Unit PCLK
        THPW,           //Horizontal Pulse width, Unit PCLK
        THB,            //Horizontal Back porch, Unit PCLK
        TVLEN,          //Vertical cycle, Unit Hsync
        TVD,            //Vertical display period, Unit line
        TVF,            //Vertical Front porch, Unit line
        TVPW,           //Vertical Pulse width, Unit line
        TVB,            //Vertical Back porch, Unit line
        pNAME,          //LCD module name or other string
    };
#elif THIS_IS_A_DRIVER_FOR_MPU_LCD
    /*command set just for MPU LCD*/
    static const T_U16 init_cmdset[][2] =
    {
        XX,XX,
        XX,XX,
        .....,
        {END_FLAG, END_FLAG}    //over send flag
    }
    static const T_U16 turnoff_cmdset[][2] =
    {
        XX,XX,
        XX,XX,
        .....,
        {END_FLAG, END_FLAG}    //over send flag
    }

    static const T_U16 turnon_cmdset[][2] =
    {
        XX,XX,
        XX,XX,
        .....,
        {END_FLAG, END_FLAG}    //over send flag
    }
#endif
```

Anyka Confidential For BOMEI Use Only

```
/*other variable define*/



/*function define*/
T_U32  lcd_read_id_func(T_eLCD lcd)
{
   //sentence
}

T_VOID lcd_init_func(T_eLCD lcd)
{
   //sentence
}

T_VOID lcd_turn_on_func(T_eLCD lcd)
{

}

T_VOID lcd_turn_off_func(T_eLCD lcd)
{
   //sentence
}

T_VOID lcd_set_disp_address_func(T_eLCD lcd, T_U32 x1, T_U32 y1, T_U32 x2, T_U32
y2)
{
   //sentence
}

T_BOOL lcd_rotate_func(T_eLCD lcd, T_eLCD_DEGREE degree)
{
   //sentence
}

T_VOID lcd_start_dma_func(T_eLCD lcd)
{
   //sentence
}

static T_LCD_FUNCTION_HANDLER function_handler =
{
   LCD_WIDTH,              // pannel size width
   LCD_HEIGHT,             // pannel size height
```

Anyka Confidential For BOMEI Use Only

```
    LCD_PCLK,              // Clock cycle, Unit Hz
    LCD_BUS_WIDTH ,        // data bus width(8,9,16,18 or 24)
    LCD_TYPE,              // interface type(0x01 MPU, 0x02 RGB, 0x03 TV-out)
    LCD_COLOR_SEL,                 // 0x00 4K color, 0x01 64K/256K color
    FLAG,                  //no use
    LCD_REG_INFO,          //refer to T_RGBLCD_INFO define

    lcd_read_id_func,      //only for MPU interface
    lcd_init_func,         //init for lcd driver
    lcd_turn_on_func,
    lcd_turn_off_func,
    lcd_set_disp_address_func,
    lcd_rotate_func,
    lcd_start_dma_func
};

static int lcd_reg(void)
{
    lcd_reg_dev(LCD_ID, &function_handler, AK_TRUE);
    return 0;
}

#ifdef __CC_ARM
#pragma arm section rwdata = "__initcall_", zidata = "__initcall_"
#endif
module_init(lcd_reg)
#ifdef __CC_ARM
#pragma arm section
#endif
#endif
```

宏 USE_LCD_XXX_XXXXXX 用来指示这个驱动文件是否会被编译(实质是，驱动文件里的函数、变量是否会被编译)，USE_LCD_XXX_XXXXXX 本来是定义在文件 platform_hd_config.h 中，也可以定义在 Makefile 里面。platform_hd_config.h 文件主要用来配置一些硬件相关的信息，对于 LCD，有如下配置：

```
//#define USE_LCD_HD66781
//#define USE_LCD_HX8312
//#define USE_LCD_DC8312
//#define USE_LCD_TRULY8312
//#define USE_LCD_ILI9320
//#define USE_LCD_SSD1289
```

Anyka Confidential For BOMEI Use Only

```
//#define USE_LCD_HX8347A
//#define USE_LCD_8907A
//#define USE_LCD_SPFD5408A
//#define USE_LCD_IS2102B
//#define USE_LCD_LQ043T3DX02
//#define USE_LCD_MPU_8907A
//#define USE_LCD_MPU_SPRING_8907
//#define  USE_LCD_RGB_TM050RDZ00
//#define  USE_LCD_RGB_A050VM01
//#define  USE_LCD_RGB_LQ043T3DX02
//#define  USE_LCD_RGB_A050VM01
//#define USE_LCD_RGB_HLY070ML209
//#define USE_LCD_RGB_HSD0701DW1
//#define USE_LCD_RGB_OTA5180A
//#define USE_LCD_MPU_ST7781
//#define USE_LCD_MPU_S6D04H0
```

若是针对 RGB LCD，需要定义 T_RGBLCD_INFO 结构体类型的变量 SUPPORT_RGB_XXXXX_TABLE, T_RGBLCD_INFO 的定义如下：

```
typedef struct
{
  T_BOOL  isInterlace;          ///< Interlace(1) or progress(0)
  T_U8    PHVG_POL;             ///< PCLK(bit3)Hsync(bit2), Vsync(bit1), Gate(bite0) polarity
  T_U16   RGBorGBR;             ///< 0=rgb,1=gbr
  T_U16   Thlen;                ///< horizontal cycle, Unit PCLK
  T_U16   Thd;                  ///<horizontal display period, Unit PCLK
  T_U8    Thf;          ///<Horizontal Front porch, Unit PCLK
  T_U8    Thpw;                 ///<Horizontal Pulse width, Unit PCLK
  T_U8    Thb;          ///<Horizontal Back porch, Unit PCLK
  T_U16   Tvlen;                ///<Vertical cycle, Unit Hsync
  T_U16   Tvd;                  ///<Vertical display period, Unit line
  T_U8    Tvf;          ///<Vertical Front porch, Unit line
  T_U8    Tvpw;                 ///<Vertical Pulse width, Unit line
  T_U8    Tvb;          ///<Vertical Back porch, Unit line
  T_U8    *pName;
}T_RGBLCD_INFO;
```

每个成员如下：

Anyka Confidential For BOMEI Use Only

isInterlace

扫描方式：隔行扫描(Interlace)还是连续扫描(progress)。当 isInterlace 为 1 时，表示隔行扫描，当 isInterlace 为 0 时，表示连续扫描。

PHVG_POL

指示 LCD 时钟 PCLK(bit 3)、行场同步信号 RGB_VOHSYNC(bit 2)、帧同步信号 RGB_VOVSYNC(bit 1)、数据同步信号 RGB_VOGATE(bit 0)的极性。1 表示 positive，0 表示 negative。

Thlen

horizontal cycle，单位是 PCLK。

Thd

horizontal display period，单位是 PCLK。

Thf

Horizontal Front porch，单位是 PCLK。

Thpw

Horizontal Pulse width，单位是 PCLK。

Thb

Horizontal Back porch，单位是 PCLK。

Tvlen

Vertical cycle，单位是 PCLK。

Tvd

Vertical display period，单位是 PCLK。

Tvf

Vertical Front porch，单位是 PCLK。

Tvpw

Vertical Pulse width，单位是 PCLK。

Tvb

Vertical Back porch，单位是 PCLK。

pName

字符串指针，表示 LCD 的型号或模组名。

T_RGBLCD_INFO 中的每个成员的信息，在每个 RGB LCD 的 specification 中都可以找到。

Anyka Confidential For BOMEI Use Only

若是针对 MPU LCD，需要定义初始化命令数组(init_cmdset)，turn off 命令数组(turnoff_cmdset)，turn on 命令数组(turnon_cmdset)，命令数组是一个二维数组，其中一维中的第一个成员表示寄存器的地址，第二个成员表示要给该寄存器写进去的数据。当寄存器的地址为 DELAY_FLAG 时，那么就是表示延时，延时时间由第二个成员指定，单位为 ms。当寄存器的地址为 END_FLAG，表示命令数组结束。

这些初始化命令数组、turn off 命令数组、turn on 命令数组的信息，在每个 MPU LCD 的 specification 中都可以找到。

函数 lcd_reg 调用驱动库接口 lcd_reg_dev 将本驱动的信息注册到 LCD 的驱动列表中。lcd_reg_dev 的原型如下：

T_BOOL lcd_reg_dev(T_U32 id, T_LCD_FUNCTION_HANDLER *handler,T_BOOL idx_sort_foward);

其中，id 为 LCD 的 identification，每个 MPU LCD 都有一个 ID，可以通过命令从 MPU LCD 中读出来。RGB LCD 没有 ID，由驱动编写者根据情况指定。handler 为 T_LCD_FUNCTION_HANDLER 结构体类型的指针，用于传递 LCD 的相关信息。idx_sort_foward 表示驱动在驱动列表中的注册顺序，1 时，表示正向，0 表示反向。

结构体类型 T_LCD_FUNCTION_HANDLER 定义了 LCD 的所有相关信息，用于函数 lcd_reg_dev 中。T_LCD_FUNCTION_HANDLER 的定义如下：

```
typedef struct
{
  T_U32   lcd_width;                  // pannel size width
  T_U32   lcd_height;                 // pannel size height
  T_U32   lcd_PClk_Hz;                // Clock cycle, Unit Hz
  T_U8    lcd_BusWidth;               // data bus width(8,9,16,18 or 24)
T_U8    lcd_type;                     // interface type(0x01 MPU, 0x02 RGB,
                                      // 0x03 TV-out)
  T_U8    lcd_color_sel;              // 0x00 4K color, 0x01 64K/256K color
  T_U8    flag;
T_VOID  *lcd_reginfo;                 // refer to T_RGBLCD_INFO
  T_U32  (*lcd_read_id_func)(T_eLCD lcd); // only for MPU interface
```

Anyka Confidential For BOMEI Use Only

```
    T_VOID (*lcd_init_func)(T_eLCD lcd);      // init for lcd driver
    T_VOID (*lcd_turn_on_func)(T_eLCD lcd);
    T_VOID (*lcd_turn_off_func)(T_eLCD lcd);
T_VOID (*lcd_set_disp_address_func)(T_eLCD lcd, T_U32 x1, T_U32 y1,
        T_U32 x2, T_U32 y2);
    T_BOOL (*lcd_rotate_func)(T_eLCD lcd, T_eLCD_DEGREE degree);
    T_VOID (*lcd_start_dma_func)(T_eLCD lcd);
}T_LCD_FUNCTION_HANDLER;
```

每个成员如下：

lcd_width：LCD 的屏宽，单位是 pixel；

lcd_height：LCD 的屏高，单位是 pixel；

lcd_PClk_Hz：LCD 的 clock，单位是 Hz；

lcd_BusWidth：LCD 的数据宽度。

lcd_type：LCD 的类型，1 表示 MPU LCD，2 表示 RGB LCD。

lcd_color_sel： LCD 的颜色深度，0 表示 4K，1 表示 64K 或 256K。

flag：reserve。

lcd_reginfo：根据 LCD 的类型指定。若是 RGB，就指向 T_RGBLCD_INFO 的结构体，若是 MPU，则为 AK_NULL。

lcd_read_id_func

lcd_init_func

lcd_turn_on_func

lcd_turn_off_func

lcd_set_disp_address_func

lcd_rotate_func

lcd_start_dma_func

相关的在驱动文件中定义的函数，如果没有定义，就赋值 AK_NULL。

通过 T_LCD_FUNCTION_HANDLER，可以将一款 LCD 的所有信息注册给驱动库。

module_init(lcd_reg)，module_init 是一个宏，用于将函数 lcd_reg 的地址放到固定的数据段中，启动时可以找到 lcd_reg 并能调用它，从而完成驱动的注册。

Anyka Confidential For BOMEI Use Only

当 LCD 的长宽发生变化时，平台的编译参数也要适当调整。因为 UI 的位置信息需要调整。如果需要选择相应的 LCD 屏，需要到 Command_XXX.txt 编译参数里定义，具体可参见《Sword37C 用户开发手册》2.1.3.1 章节。选择 LCD 以及大小尺寸示例如下：

LCD=MPU_SPRING_8907: LCD TYPE (MPU_SPRING_8907 / MPU_8907A / MPU_S6D04H0 / RGB_OTA5180A ETC. REF platform_hd_config.h)

LCDWIDTH=480 LCDHEIGHT=272：lcd480×272 的；

LCDWIDTH=320 LCDHEIGHT=240：lcd320×240 的；

3.2 电池检测

电池检测接口 analog_getvalue_bat 获取的是 ADC1 AD4 的值，支持芯片内部 2 分压或者 3 分压。通过内部或者外部分压，可以检测的电压范围是 2.3V～5V。驱动库默认芯片内部 3 分压。

若使用 AD5 检测电池电压，需要外部分压。AD5 本身可以检测的电压范围是 0V～(AVDD-0.7V)。

3.3 串口打印

串口打印默认使用 UART0，若使用其他串口进行打印，平台可以初始化其他串口。

3.4 按键方式

平台支持两种按键检测方式：扫描检测与模拟检测。扫描检测通过 GPIO 中断来检测，模拟检测通过 AD5 定时采样检测。

3.5 计时器

目前有 1ms 和 1us 两种 tick 数可选。这两种 Tick 数可以通过调用 Timer 驱动相应的函数接口来进行选择。请用户参见 2.15.2“接口说明”。

3.6 芯片选择

platform\Middle\SourceCode\init 目录下的 Fwl_multitask.c 文件里的 TMC_Task 接口中设置芯片的选择。系统初始化时修改。如:

Anyka Confidential For BOMEI Use Only

drv_info.chip= CHIP_AK3750

3.7 SPI Flash

如果需要更换一款 spiflash，则要通过修改配置 hal_spiflash.h 里的 T_SFLASH_PARAM 结构体参数。如下所示：

```
typedef struct tagSFLASH_PARAM
{
    T_U32   id;                 ///< flash id
    T_U32   total_size;         ///< flash total size in bytes
    T_U32   page_size;          ///< bytes per page
    T_U32   program_size;       ///< program size at 02h command
    T_U32   erase_size;         ///< erase size at d8h command
    T_U32   clock;              ///< spi clock, 0 means use default clock

    //chip character bits:
    //bit 0: under_protect flag, the serial flash under protection or not when power on
    //bit 1: fast read flag, the serial flash support fast read or not(command 0Bh)
    //bit 2: AAI flag, the serial flash support auto address increment word programming
    T_U8    flag;               ///< chip character bits
    T_U8    protect_mask;       ///< protect mask bits in status register:BIT2:BP0, BIT3:BP1,
BIT4:BP2, BIT5:BP3, BIT7:BPL
    T_U8    reserved1;
    T_U8    reserved2;
}T_SFLASH_PARAM;
```

Anyka Confidential For BOMEI Use Only

4 驱动注意事项

芯片有些资源存在复用的情况，如：GPIO 等。在使用这些资源的时候，要注意他们的竞争问题。下面对一些驱动常用的、需要注意的事项进行简要的说明。

4.1 GPIO的share pin

驱动库对 GPIO 的 share pin 的使用，在 gpio_cfg.c 文件的 gpio_share_pin_init 这个函数初始化了一组默认值。如果硬件使用的 GPIO 不是默认值，需要做相应的修改。

如串口 0，默认使用的 GPIO 配置如下：

```
static T_SHARE_UART m_share_uart1 = {
   /*.txd = */14,
   /*.rxd = */2,
   /*.cts = */INVALID_GPIO,
   /*.rts = */INVALID_GPIO,
};
```

默认使用 GPIO14 和 2 作为 TX 和 RX，没有硬件流控。如果硬件连接用了 32 和 33 作为 TX 和 RX，则该串口的配置值需要修改成：

```
static T_SHARE_UART m_share_uart1 = {
   /*.txd = */32,
   /*.rxd = */32,
   /*.cts = */INVALID_GPIO,
   /*.rts = */INVALID_GPIO,
};
```

要使用修改生效，在上层应用需要使用下面的函数

T_VOID gpio_share_pin_set(E_GPIO_PIN_SHARE_CONFIG module, T_U8 gpios[], T_U32 gpio_num)

该函数的使用可以参考“2.11 GPIO”

上面的操作只是配置了串口通信引脚的值，要使这些 GPIO share 成串口所用，还需要使用如下函数：

T_BOOL gpio_pin_group_cfg(E_GPIO_PIN_SHARE_CONFIG PinCfg)

该函数必须在 gpio_share_pin_set 之后调用。函数的使用可以参考“2.11 GPIO”

所用模块结构中不用的成员，都要设成 INVALID_GPIO，否则可能会出现异常情况。

4.2 硬件timer使用情况

37C 内部有 5 个硬件定时器，其中 timer0 和 1 可以复用为 PWM，如果 timer0 和 1 用作 PWM，则不能作为计时用。驱动对这 5 个驱动的使用情况。

Timer0,：留给 PWM0 用，暂时不用。

Timer1：作 PWM1，用在 LCD 的背光调节。

Anyka Confidential For BOMEI Use Only

Timer2：为 OS、Vtimer 和 tick 提供时钟基准。

Timer3：用在 watchdog。

Timer4：用在 USB host。

一般情况下，硬件定时器都使用默认分配，如果要做修改，请谨慎。

4.3 驱动库编译

当驱动库有修改时，需要重新编译，使修改生效。在驱动库的根目录下 build 文件夹内，有几个文件需要注意的，具体说明如下：

bld.bat：这个是编译平台驱动库的批处理文件。

bld_bios.bat：这个是编译 BIOS 的批处理文件。但目前 3750C 暂时不用 BIOS。

bld_burntool.bat：这个是编译烧录工具的驱动库批处理文件。

make_ads_ak37xx.pl：这个是生成 Makefile 的批处理文件，对应于 ADS1.2 编译器。

make_ak37xx.pl：这个是生成 Makefile 的批处理文件，对应于 GCC 编译器。

make_clean.bat：这个是清除批处理问题。

首次生成驱动库的时候，先运行 make_ads_ak37xx.pl（ADS1.2 编译器），生成 Makefile；再运行 bld.bat，生成的 drv_ak37xx.a 在上一层目录。

如果新增了*.c 文件，要运行 make_ads_ak37xx.pl，生成新的 Makefile，再运行 bld.bat。

如果修改的是*.h 文件而*.c 文件没改动，需要运行先 make_clean.bat，再运行 bld.bat。

Anyka Confidential For BOMEI Use Only