
IROT 平台抽象层接口

Rev 1.0

Release Date: 2018-04-08

1. 概述

1.1 目的

本文档主要提供给设备端平台开发者使用，用于平台抽象层的适配。

2. 平台抽象层接口描述

2.1 分配内存

函数原型: `void* irot_pal_memory_malloc(int size);`

功能描述: 分配指定长度的内存

参数描述: [IN] `size`, 要分配的内存长度

返回值: 成功, 返回内存的首地址; 错误, 返回 `NULL`

2.2 释放内存

函数原型: `irot_pal_memory_free(void* ptr);`

功能描述: 释放之前用 `irot_pal_memory_malloc` 分配的内存

参数描述: [IN] `ptr`, 用 `irot_pal_memory_malloc` 分配的内存的首地址

返回值: 无

2.3 获取随机数

函数原型: `irot_result_t irot_pal_get_random(uint8_t* buf, uint32_t len);`

功能描述: 获取随机数

参数描述: [OUT] `buf`, 输出数据的起始地址

[IN] 要生成的随机数的长度

返回值: 见 `irot_result_t`

2.4 base64 编码

函数原型: `irot_result_t irot_pal_base64_encode(const uint8_t* in, int in_len,
uint8_t* out, uint32_t* out_len);`

功能描述: 使用 `base64` 对指定数据进行编码

参数描述: [IN] `in`, 待编码数据的起始地址

[IN] in_en, 待编码数据的长度
[OUT] out, 编码后数据的起始地址
[IN/OUT] out_len, 输入为缓冲区长度, 输出实际编码后数据的长度
返回值: 见 irot_result_t

2.5 摘要算法

函数原型: irot_result_t irot_pal_hash_sum(const uint8_t* in, uint32_t in_len,
uint8_t* out, uint32_t* out_len,
digest_t type);

功能描述: 用指定算法对数据做摘要
参数描述: [IN] in, 原始数据的起始地址
[IN] in_len, 原始数据的长度
[OUT] out, 摘要数据的起始地址
[IN/OUT] out_len, 输入为缓冲区长度, 输出实际摘要数据的长度
[IN] type, 见类型定义 digest_t
返回值: 见 irot_result_t

2.6 日志输出

函数原型: void irot_pal_log(const char* fmt, ...);

功能描述: 按指定格式输出日志
参数描述: [IN] fmt, 格式字符串, 类似 printf
[IN] ..., 可变参数用于输出信息, 类似 printf
返回值: 无

3.附录

3.1 错误码定义

```
typedef enum
{
    IROT_SUCCESS                = 0, ///< The operation was successful.
    IROT_ERROR_GENERIC          = -1, ///< Non-specific cause.
    IROT_ERROR_BAD_PARAMETERS   = -2, ///< Input parameters were invalid.
    IROT_ERROR_SHORT_BUFFER     = -3, ///< The supplied buffer is too short for the output.
    IROT_ERROR_EXCESS_DATA      = -4, ///< Too much data for the requested operation was passed.
    IROT_ERROR_OUT_OF_MEMORY     = -5, ///< System out of memory resources.
    IROT_ERROR_COMMUNICATION     = -7, ///< Communication error
    IROT_ERROR_NOT_SUPPORTED     = -8, ///< The request operation is valid but is not supported in this implementation.
    IROT_ERROR_NOT_IMPLEMENTED   = -9, ///< The requested operation should exist but is not yet implementation.
    IROT_ERROR_TIMEOUT          = -10, ///< Communication Timeout
    IROT_ERROR_ITEM_NOT_FOUND    = -11, ///< Id2 is not exist
} irot_result_t;
```

3.2 类型定义

```
typedef enum
{
    DIGEST_TYPE_SHA1            = 0x00,
    DIGEST_TYPE_SHA224          = 0x01,
    DIGEST_TYPE_SHA256          = 0x02,
    DIGEST_TYPE_SHA384          = 0x03,
    DIGEST_TYPE_SHA512          = 0x04,
    DIGEST_TYPE_SM3              = 0x05,
} digest_t;
```