RTOS 模块适配标准接口

产品版本	资料版本	资料编号	资料更新说明
	V1.0		初始版本
	V1.1		1. 增加修改文件长度接口 2. 增加 FOTA 升级接口
	V1.2		1. 增加保存 Fota 升级信息到模块接口用于最小系统升级
	V1.3		1. 修改网络状态结构体 2. 增加 Fota 差分文件大小设置接口 3. 增加 TCP socket 断开回调

目录

类型说明

返回值类型

```
typedef enum {
    OEM_FAILURE = -1,
    OEM_SUCCESS = 0,
    OEM_PROCESS = 1,
    OEM_MAX
}oem_ret_e;
```

网络状态类型

```
typedef enum {
    NO_SERVICE = 0,
    SERVICE_AVAILABLE = 1,
    OEM_NET_MAX,
}oem_net_status;
```

数据链路状态类型

```
typedef enum {
    NO_PPP_SERVICE = 0,
    PPP_SERVICE_AVAILABLE = 1,
    OEM_PPP_MAX,
}oem_ppp_status;
```

TTS 数据编码类型

```
enum {
    TTS_STOP,
    TTS_PLAY_UTF16LE,       /*play ucs2 pcm */
    TTS_PLAY_GBK,                   /*play gbk*/
};
```

回调函数注册类型

```
typedef enum {
    CALL_BACK_REPORT_RECORD_DATA,                   // 上报录音数据
    CALL_BACK_REPORT_NETWORK_STATUS,                // 上报注网状态
```

```
    CALL_BACK_REPORT_AT_POC_CMD,                    // 上报 at+poc 指令
    CALL_BACK_REPORT_USER_MSG,                      // 上报用户消息
    CALL_BACK_REPORT_CONNECT_TCP_ID,                // 上报已连接 tcp sokcet 标识
    CALL_BACK_REPORT_RECV_TCP_ID,                   // 上报可读 tcp socket 标识
    CALL_BACK_REPORT_RECV_UDP_ID,                   // 上报可读 udp socket 标识
    CALL_BACK_REPORT_PLAY_BUFFER_REST_DATA,     // 上报 poc 播放缓存剩余
poc 数据
    CALL_BACK_REPORT_TTS_STATUS,                    // 上报 tts 播放状态
    CALL_BACK_REPORT_PPP_STATUS,                    // 上报 ppp 状态
    CALL_BACK_REPORT_TCP_CLOSE_ID,                  // 上报断开 tcp socket 标识
    CALL_BACK_MAX
}oem_poc_call_back_type_e;
```

SIM 卡状态类型

```
typedef enum
{
    POC_SIM_STATUS_NONE   =   0,
    POC_SIM_STATUS_READY,
    POC_SIM_STATUS_NOT_INSERT,
    POC_SIM_STATUS_MAX,
}oem_sim_status_type_e;
```

网络状态类型

```
typedef enum
{
  NETWORK_STATUS_NONE = 0,
  NETWORK_UNREGISTER ,
  NETWORK_REGISTER ,
}oem_network_status_type_e;
```

网络模式类型

```
typedef enum
{
  NETWORK_MODE_NONE = 0,
```

```
    NETWORK_MODE_CDMA = 1,
    NETWORK_MODE_HDR = 1 << 1,
    NETWORK_MODE_LTE = 1 << 2,
    NETWORK_MODE_GSM = 1 << 3,
    NETWORK_MODE_WCDMA = 1 << 4,
    NETWORK_MODE_TDCDMA = 1 << 5,
    NETWORK_MODE_MAX,
}oem_network_mode_type_e;
```

结构体说明

网络状态结构体

```
typedef struct
{
    oem_network_status_type_e network_status;
    oem_network_mode_type_e network_mode;
    unsigned int cell_id;
    unsigned int lac_or_tac;    //获取模块 LAC 或者 TAC
    char rssi;
    char mcc[4];        //add
    char mnc[4];        //add
}oem_network_info_s;
```

数据链路状态结构体

```
typedef enum
{
  PPP_STATUS_NONE = 0,
  PPP_STATUS_CONNECTING ,
  PPP_STATUS_CONNECTED    ,
  PPP_STATUS_DISCONNECTING    ,
  PPP_STATUS_DISCONNECTED,
}oem_ppp_status_type_e;
```

消息队列结构体

```
typedef struct {
    unsigned int MsgType;
    void* MsgData;
    unsigned int DataLen;
    unsigned short param;
}oem_poc_user_msg_s;
```

Fota 信息结构体

```
typedef struct
```

```
{
    char devNm[20];     //设备名字
    unsigned short devVersion;    //设备当前版本号
    char fotaAddr[50];              //Fota 服务器 IP 地址
    unsigned short fotaPort;      //Fota 服务器端口号
}fota_info_t;
```

回调函数说明

录音数据上报回调函数

typedef void (* fn_type_report_record_data_cb)(const char* data, int length);

网络状态上报回调函数

typedef void (* fn_type_report_network_status_cb)(oem_net_status result);

AT+POC 命令上报回调函数

typedef void (* fn_type_report_at_poc_cmd_cb)(char* buf ,int len);

用户自定义消息上报回调函数

typedef void (* fn_type_report_user_msg_cb)(oem_poc_user_msg_s msg);

TCP 连接成功上报回调函数

typedef void (* fn_type_report_connect_tcp_id_cb)(unsigned char socket_id);

TCP socket 可读上报回调函数

typedef void (* fn_type_report_recv_tcp_id_cb)(unsigned char socket_id);

UDP socket 可读上报回调函数

typedef void (* fn_type_report_recv_udp_id_cb)(unsigned char socket_id);

TCP socket 断开上报回调函数

typedef void (* fn_type_report_tcp_close_id_cb)(unsigned char socket_id);

播放缓冲区剩余数据量上报回调函数

typedef void (* fn_type_report_play_buffer_rest_data_cb)(int rest_data);

TTS 播放状态上报回调函数

typedef void (* fn_type_report_tts_status_cb)(int status);

数据链路状态上报回调函数

typedef void (* fn_type_report_ppp_status_cb)(oem_ppp_status status);

定时器回调函数

typedef void (* timercallbackfunc_t)(unsigned long param);

接口函数说明

POC 库初始化接口（模块调用 POC 库实现）

extern void OEM_PocInit(void);

数据链路相关接口

int OEM_pppOpen(void);

int OEM_pppClose(void);

oem_ppp_status_type_e OEM_getPppStatus(void);

网络状态接口

int OEM_GetNetworkInfo(oem_network_info_s *network_info);

IMEI 接口

int OEM_GetImeiInfo(char *imei);

SIM 卡状态接口

oem_sim_status_type_e OEM_getSimStatus();

int OEM_getSimIccid(char *uim_id);

音频操作接口

oem_ret_e OEM_StarRecord(void);

oem_ret_e OEM_StopRecord(void);

oem_ret_e OEM_StarPlay(void);

oem_ret_e OEM_StopPlay(void);

int OEM_Play(const char *data, int length);

int OEM_GetPlayBufferAvail(void);

oem_ret_e OEM_CleanPlayBuffer(void);

提示音接口

void OEM_PlayTone(int Type, int time);

extern void OEM_Play_Tone_Sound(int f1, int f2, int time);

串口数据发送接口

oem_ret_e OEM_SendUart(char *uf, int len);

定时器接口(单位为 tick 一个 tick 5ms)

void OEM_SetTimer(void **TimerRef, unsigned long timeout, unsigned long rescheduleTime, timercallbackfunc_t callbackfunc, unsigned long timerArgc);

备注

TimerRef 定时器指针

Timeout 超时时间单位 tick

rescheduleTime 是否重复 0 不重复 大于 0 重复

callbackfunc 回调函数指针

timerArgc 回调函数参数

void OEM_CancelTimer(void *TimerRef);

unsigned long OEM_Get_time(void);

备注

系统开机到现在经历的时间 单位 tick

void OEMSleep(int ticks)

TTS 接口

int OEM_TTS_Spk(int type, char * atxt);

socket 接口

1. TCP socket 接口

oem_ret_e OEMSocket_SetupTCP(unsigned char socket_id, char *ip, unsigned short port);

int OEMSocket_SendTCP(unsigned char socket_id, char *buffer, int buf_len);

int OEMSocket_ReadTcpData(unsigned char socket_id, const char *data, int length);

oem_ret_e OEMSocket_CloseTCP(unsigned char socket_id);

2. UDP socket 接口

int OEMSocket_SetupUDP(unsigned char socket_id);

int OEMSocket_SendUDP(unsigned char socket_id, char *ip, unsigned int port, char *buff, int length);

int OEMSocket_UdpReadData(unsigned char socket_id, char *data, unsigned short length);

int OEMSocket_CloseUDP(unsigned char socket_id);

3. DNS 接口

oem_ret_e OEMSocket_GetIpByName(char *name, char *ip);

消息队列接口

oem_ret_e OEM_PostMessage(oem_poc_user_msg_s msg);

备注：

调用该接口 Post 出的消息，将通过用户自定义消息上报回调通知 Poc 库，用于任务间调度

extern unsigned long OEM_PocGetMsgQueueRestSpace(void);

文件操作接口（标准 C 模式）

int OEMFile_Open(const char *path, const char *oflag);

int OEMFile_Close(unsigned int filedes);

int OEMFile_Seek(unsigned int filedes, long offset, int where);

long OEMFile_Tell(unsigned int filedes);

int OEMFile_Read(unsigned int filedes, void *buf, int size, int count);

int OEMFile_Write(unsigned int filedes, void *buf, int size, int count);

int OEMFile_ftruncate(unsigned int filedes, unsigned int size);

回调函数注册接口

void OEM_RegisterUserCallBack(oem_poc_call_back_type_e type, void *callback);

差分升级接口

void OEM_FotaUpdate(char *path);

void Fota_setInfo(fota_info_t data); //保存 Fota 升级信息到模块

备注:

如果模块不需要最小系统升级，该函数做空函数即可

int OEM_FotaSizeInit(int size)

备注:

如果模块不需要该函数做空函数即可

日志输出接口

void OEMLogPrintf(const char *fmt,...);