

应急广播协议处理模块移植文档

移植文档

|      |            |        |            |
|------|------------|--------|------------|
|      | 签名         |        | 日期         |
| 拟制   | 郑光明        |        | 2017-12-01 |
| 审核   |            |        |            |
| 批准   |            |        |            |
| 版本   | V1.0       | 正式发布日期 |            |
| 文件编号 | QR-A. 0-43 |        |            |

# 目录

|  |   |
|--|---|
| 目录 .....                                     | 2 |
| §1 概要 .....                                  | 1 |
| §1.1 文档名称 .....                              | 1 |
| §1.2 范围 .....                                | 1 |
| §1.3 模块功能 .....                              | 1 |
| §1.4 模块概述 .....                              | 1 |
| §2 模块的使用 .....                               | 2 |
| §2.1 需要链接的库和头文件 .....                        | 2 |
| §2.2 常量枚举变量 ( 宏 ) 描述 .....                   | 2 |
| §2.3 函数指针描述 .....                            | 2 |
| § 2.3.1 m_GS_EM_DevParamSet .....            | 2 |
| § 2.3.2 m_GS_EM_DevStatGet .....             | 3 |
| § 2.3.3 m_GS_EM_DevGetPsiInfo .....          | 3 |
| § 2.3.4 m_GS_EM_DevWaitPostBackService ..... | 3 |
| § 2.3.5 m_GS_EM_DevGetStorePath .....        | 4 |
| §2.4 数据结构描述 .....                            | 4 |
| § 2.4.1 GS_EM_DEV_IF_INIT_PARAM_S .....      | 4 |
| § 2.4.2 GS_EM_DEV_TTS_PARAM_S .....          | 4 |
| § 2.4.3 GS_EM_DEV_TUNER_PARAM_S .....        | 5 |
| § 2.4.4 GS_EM_DEV_FMT_PARAM_S .....          | 5 |
| § 2.4.5 GS_EM_DEV_FMR_PARAM_S .....          | 5 |
| § 2.4.6 GS_EM_DEV_DECODER_PARAM_S .....      | 6 |
| § 2.4.7 GS_EM_DEV_NET_ADDR_S .....           | 6 |

|  |   |
|--|---|
| § 2. 4. 8 GS_EM_DEV_AUD_BC_INFO_S .....                    | 6 |
| § 2. 4. 9 GS_EM_DEV_BC_PARAM_S .....                       | 6 |
| § 2. 4. 10 GS_EM_DEV_LOCALIP_S .....                       | 7 |
| § 2. 4. 11 GS_EM_DEV_SYS_PARAM_S .....                     | 7 |
| § 2. 4. 12 GS_EM_DEV_SYS_STAT_S .....                      | 7 |
| § 2. 4. 13 GS_EM_DEV_PSI_INFO_S .....                      | 7 |
| § 2. 4. 14 GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_START_S ..... | 8 |
| § 2. 4. 15 GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_End_S .....   | 8 |
| § 2. 4. 16 GS_EM_DEV_POSTBACK_SERV_S .....                 | 8 |
| §2. 5 API 函数列表.....  | 8 |
| § 2. 5. 1 API 函数列表.....                                    | 8 |
| § 2. 5. 2 GS_EM_ProtProcTaskCreate .....                   | 8 |
| § 2. 5. 3 GS_EM_ProtProcTaskDestroy .....                  | 9 |
| § 2. 5. 4 GS_EM_RemoteReConnect .....                      | 9 |
| §2. 6 移植示例 .....   | 9 |

## § 1 概要

### § 1.1 文档名称

本文档是应急广播协议处理模块的移植文档（以下简称 本文档）。

### § 1.2 范围

本文档主要是描述如何使用广播协议处理模块来进行 TCP 应急控制协议的解析并控制主体设备。

### § 1.3 模块功能

本模块主要是解析 TCP 接收的协议，从而控制本地设备执行相应的操作。

### § 1.4 模块概述

本模块将 TCP 的接收、协议的解析和设备的控制进行分离，通过回调的方式控制设备，这样同一个模块，可以控制不同功能的设备。

该模块在 PC 机 (Linux) 进行开发和测试，通过不同的编译在不同的环境进行运行，实现了可移植和独立开发测试。

## § 2 模块的使用

### § 2.1 需要链接的库和头文件

需要链接的库文件：“libgs\_prot\_proc\_xxx.a”，XXX 代表平台类型，如 libgs\_prot\_proc\_x86.a 代表 X86 平台即 PC 机编译的文件，其对应头文件为 “gs\_dev\_if.h” 和 “gs\_comm.h”。

### § 2.2 常量枚举变量（宏）描述

| 序号 | 常量名称                            | 说明              |
|----|---------------------------------|-----------------|
| 1  | GS_EM_DEV_MODULE_TAG_E          | 设备模块索引          |
| 2  | GS_EM_DEV_TTS_ENC_FMT_E         | TTS 文本编码格式      |
| 3  | GS_EM_DEV_TUNER_MODULATOR_E     | Tuner 调制方式      |
| 4  | GS_EM_DEV_TUNER_PLOAR_E         | Tuner 极化方式      |
| 5  | GS_EM_DEV_TUNER_SPECINV_E       | Tuner 频谱翻转      |
| 6  | GS_EM_DEV_FMT_SOUNDCHN_E        | FM 发送模块声道       |
| 7  | GS_EM_DEV_FMT_DEEMPH_E          | FM 发送模块预加重时间设置  |
| 8  | GS_EM_DEV_FMT_I2S_SAMPLE_FREQ_E | FM 发送模块 I2S 采样率 |
| 9  | GS_EM_DEV_FMR_SOUNDCHN_E        | FM 接收模块声道       |
| 10 | GS_EM_DEV_FMR_DEEMPH_E          | FM 接收模块预加重时间设置  |
| 11 | GS_EM_DEV_DECODER_VIDEO_MODE_E  | 解码器视频模式         |
| 12 | GS_EM_DEV_DECODER_TV_SYS_E      | 解码器视频输出制式       |
| 13 | GS_EM_DEV_DECODER_SOUNDCHN_E    | 解码器声道           |
| 14 | GS_EM_DEV_AUD_TRANS_PROT_TYPE_E | 音频传输协议类型        |
| 15 | GS_EM_DEV_AUD_ENC_FMT_E         | 音频编码格式          |
| 16 | GS_EM_DEV_BC_PARAM_TYPE_E       | 广播参数类型          |
| 17 | GS_EM_DEV_SYS_PARAM_TAG_E       | 系统参数类型索引        |
| 18 | GS_EM_DEV_WORK_STAT_E           | 工作状态            |
| 19 | GS_EM_DEV_POSTBACK_SERV_TAG_E   | 回传节目类型          |
| 20 | GS_EM_DEV_UDP_RECV_CHANNEL_E    | UDP 接收通道编号      |

### § 2.3 函数指针描述

#### § 2.3.1 m\_GS\_EM\_DevParamSet

函数原型

```
GS_S32 (*m_GS_EM_DevParamSet)(GS_EM_DEV_MODULE_TAG_E ModuleTag,  
GS_VOID *pParamInfo);
```

函数功能

模块回调函数，用于设备参数设置。

参数说明

**ModuleTag:** 设置参数的模块类型；

**pParamInfo:** 根据不同的模块，传送的不同的设置参数；

返回值

GS\_SUCCESS 成功

GS\_FAILURE 失败

### § 2.3.2 m\_GS\_EM\_DevStatGet

函数原型

GS\_S32 (\*m\_GS\_EM\_DevStatGet)(GS\_EM\_DEV\_MODULE\_TAG\_E ModuleTag,  
GS\_VOID \*pStatInfo);

函数功能

模块回调函数，用于设备状态获取。

参数说明

**ModuleTag:** 需要获取参数的模块类型；

返回值

GS\_SUCCESS 成功

GS\_FAILURE 失败

**pStatInfo:** 返回需要获取的设备类型的状态；

### § 2.3.3 m\_GS\_EM\_DevGetPsiInfo

函数原型

GS\_S32 (\*m\_GS\_EM\_DevGetPsiInfo)(GS\_S32 ChannelIndex,  
GS\_EM\_DEV\_PSI\_INFO\_S \*pPsiInfo);

函数功能

模块回调函数，用于获取指定通道的 PSI 信息。

参数说明

**ChannelIndex:** 输入通道类型，GS\_EM\_DEV\_UDP\_RECV\_CHANNEL\_E 定义；

返回值

GS\_SUCCESS 成功

GS\_FAILURE 失败

**pPsiInfo:** 解复用出来的 PSI 信息表；

### § 2.3.4 m\_GS\_EM\_DevWaitPostBackService

函数原型

GS\_S32  
(\*m\_GS\_EM\_DevWaitPostBackService)(GS\_EM\_DEV\_POSTBACK\_SERV\_S  
\*pPostBackServInfo);

函数功能

模块回调函数，阻塞等待设备回传业务。

参数说明

返回值

GS\_SUCCESS 成功

GS\_FAILURE 失败

pPostBackServInfo: 回传业务信息

### § 2.3.5 m\_GS\_EM\_DevGetStorePath

函数原型

GS\_VOID (\*m\_GS\_EM\_DevGetStorePath)(GS\_CHAR \*pPath);

函数功能

模块回调函数，用户获取设备标识注册文件的存储路径。

参数说明

返回值

GS\_SUCCESS 成功

GS\_FAILURE 失败

pPath: 注册文件存储的路径

## § 2.4 数据结构描述

### § 2.4.1 GS\_EM\_DEV\_IF\_INIT\_PARAM\_S

结构体定义:

```
typedef struct {
    /*
        m_ModuleTag:    设备内模块类型
        pParamInfo:     设备内相应模块的参数
    */
    GS_S32 (*m_GS_EM_DevParamSet)(GS_EM_DEV_MODULE_TAG_E ModuleTag, GS_VOID
*pParamInfo); /* 设备参数设置 */
    GS_S32 (*m_GS_EM_DevStatGet)(GS_EM_DEV_MODULE_TAG_E ModuleTag, GS_VOID *pStatInfo); /*
设备状态获取 */
    GS_S32 (*m_GS_EM_DevGetPsiInfo)(GS_S32 ChannelIndex, GS_EM_DEV_PSI_INFO_S *pPsiInfo); /*
ChannelIndex: 通道编号，搜索指定的哪路通道，不同设备不一样 */
    GS_S32 (*m_GS_EM_DevWaitPostBackService)(GS_EM_DEV_POSTBACK_SERV_S
*pPostBackServInfo); /* 阻塞等待回传业务 */

    GS_VOID (*m_GS_EM_DevGetStorePath)(GS_CHAR *pPath); /* 获取存储路径，返回值放到 pPath，如:
/mnt/mtdd */
} GS_EM_DEV_IF_INIT_PARAM_S;
```

结构体用途说明：用于创建模块是传入参数。

### § 2.4.2 GS\_EM\_DEV\_TTS\_PARAM\_S

结构体定义:

```
typedef struct{
    GS_S32                                m_Index; /* 模块索引，一个设备可能有多个模块 */

    GS_U8                                m_Speed; /* 语速 0-10 */
    GS_U8                                m_Volume; /* 音量 0-10 */
    GS_U8                                m_Tune; /* 语调 0-10 */
}
```

```

        GS_EM_DEV_TTS_ENC_FMT_E          m_TextEncFmt; /* 文本编码方式 */
        GS_S32                            m_TextLen; /* 文本长度 */
        GS_CHAR                           m_pText[65535]; /* 文本内容 */
    } GS_EM_DEV_TTS_PARAM_S;

```

结构体用途说明：传递 TTS 设置参数，具体说明见注释。

### § 2.4.3 GS\_EM\_DEV\_TUNER\_PARAM\_S

结构体定义：

```

typedef struct{
    GS_S32                            m_Index; /* 模块索引，一个设备可能有多个模块 */

    GS_U32                            m_Frequency; /* 频率，单位：KHz */
    GS_U32                            m_Symbol; /* 符号率，单位：KHz */
    GS_U32                            m_LocalFreq; /* 本地频率，单位：KHz */
    GS_EM_DEV_TUNER_MODULATOR_E      m_Modulation; /* 信号调制模式 */
    GS_EM_DEV_TUNER_PLOAR_E           m_PolarMethod; /* 极性方式 */
    GS_BOOL                           m_Switch_22K;
    GS_EM_DEV_TUNER_SPECINV_E         m_SpectInv; /* 频谱翻转 */
} GS_EM_DEV_TUNER_PARAM_S;

```

结构体用途说明：传递 Tuner 设置参数，具体说明见注释。

### § 2.4.4 GS\_EM\_DEV\_FMT\_PARAM\_S

结构体定义：

```

typedef struct{
    GS_S32                            m_Index; /* 模块索引，一个设备可能有多个模块 */

    GS_U32                            m_Freq; /* 频率，单位：KHz 范围：76MHz ~
108MHz */
    GS_BOOL                           m_RdsSwitch; /* RDS 开关，TRUE 开启发送
RDS，FALSE 关闭发送 RDS */
    GS_EM_DEV_FMT_SOUNDCHN_E          m_SoundChnMode; /* 发送声道模式 */
    GS_U32                            m_OutLvl; /* 输出电平，范围 0 - 100 */
    GS_BOOL                           m_MuteSwitch; /* 静音开关，TRUE 静音，FALSE
正常音量发送 */
    GS_U32                            m_TxFreqDev; /* 总的发送频率偏差，范围：
0~17595Hz 单位：Hz */
    GS_U32                            m_RdsFreqDev; /* RDS 频率偏差，范围：
0~4445Hz 单位：Hz */
    GS_EM_DEV_FMT_DEEMPH_E            m_PreEmphasisConst; /* 预加重设置 */

    GS_EM_DEV_FMT_I2S_SAMPLE_FREQ_E   m_SampleFreq; /* 采样率，仅针对于 I2S 输入 */
} GS_EM_DEV_FMT_PARAM_S;

```

结构体用途说明：传递 FM 发送模块设置参数，具体说明见注释。

### § 2.4.5 GS\_EM\_DEV\_FMR\_PARAM\_S

结构体定义：

```

typedef struct{
    GS_S32                            m_Index; /* 模块索引，一个设备可能有多个模块 */

    GS_U32                            m_Freq; /* 频率，单位：KHz 范围：60MHz ~
108MHz */
    GS_U8                             m_Vol; /* 音量 0-100 */
    GS_BOOL                           m_MuteSwitch; /* TRUE:静音 FALSE:不静音 */

```



```

        GS_BOOL                                m_RdsSwitch; /* TRUE:开启 Rds 接收 FALSE:关
闭 Rds 接收 */
        GS_EM_DEV_FMR_SOUNDCHN_E               m_SoundChnMode; /* AUTO:自动接收 MONO:强制接收
Mono 单声道 */
        GS_EM_DEV_FMR_DEEMPH_E                 m_DeEmphasisConst; /* 去加重常量 */
    } GS_EM_DEV_FMR_PARAM_S;

```

结构体用途说明：传递 FM 接收模块设置参数，具体说明见注释。

## § 2.4.6 GS\_EM\_DEV\_DECODER\_PARAM\_S

结构体定义：

```

typedef struct{
    GS_S32                                m_Index; /* 模块索引，一个设备可能有多
个模块 */

    GS_U8                                m_DecoderIndex; /* 解码芯片索引号，根
据解码板具体设置 */
    GS_EM_DEV_DECODER_TV_SYS_E           m_VideoOutputStandard; /* 视频输出制式
PAL/NTSC... */
    GS_EM_DEV_DECODER_VIDEO_MODE_E       m_VideoOutputMode; /* 视频输出模式 CVBS... */
    GS_EM_DEV_DECODER_SOUNDCHN_E         m_SoundChannel; /* 声道 LEFT/RIGHT/STEREO... */
    GS_U8                                m_AudioVolume; /* 音量 0 - 100 */

    GS_U16                                m_ServiceId; /* 节目号 ID */
    GS_U16                                m_PmtPid;
    GS_U16                                m_AudioPid;
} GS_EM_DEV_DECODER_PARAM_S;

```

结构体用途说明：传递解码器设置参数，具体说明见注释。

## § 2.4.7 GS\_EM\_DEV\_NET\_ADDR\_S

结构体定义：

```

typedef struct {
    GS_U8    m_Type; /* 0x01: IP + Port; 0x02: 域名 + 端口 */
    union {
        GS_U32    m_IpAddr;
        GS_CHAR    m_pDomain[64]; /* 域名最大长度限制为 63 个字符 */
    } m_HostName; /* 回传 IP 地址或域名 */
    GS_U16    m_Port;
} GS_EM_DEV_NET_ADDR_S;

```

结构体用途说明：传递网络地址设置参数，具体说明见注释。

## § 2.4.8 GS\_EM\_DEV\_AUD\_BC\_INFO\_S

结构体定义：

```

typedef struct {
    GS_EM_DEV_AUD_TRANS_PROT_TYPE_E m_AudTransProt; /* 音频数据传送协议类型 */
    GS_EM_DEV_AUD_ENC_FMT_E         m_AudEncFmt; /* 音频编码格式 */
    GS_EM_DEV_NET_ADDR_S            m_Addr; /* 网络地址 */
} GS_EM_DEV_AUD_BC_INFO_S;

```

结构体用途说明：传递音频广播设置参数，具体说明见注释。

## § 2.4.9 GS\_EM\_DEV\_BC\_PARAM\_S

结构体定义：

```
typedef struct{
    GS_EM_DEV_BC_PARAM_TYPE_E          m_Type;
    GS_U16                               m_Len;
    union {
        GS_EM_DEV_AUD_BC_INFO_S        m_AudInfo;
        GS_EM_DEV_NET_ADDR_S           m_PicInfo;
        GS_EM_DEV_NET_ADDR_S           m_VidInfo;
    } m_Data;
} GS_EM_DEV_BC_PARAM_S;
```

结构体用途说明：传递设备广播参数。

## § 2.4.10 GS\_EM\_DEV\_LOCALIP\_S

结构体定义：

```
typedef struct {
    GS_U32 m_IpAddr;
    GS_U32 m_IpMask;
    GS_U32 m_IpGate;
} GS_EM_DEV_LOCALIP_S;
```

结构体用途说明：本地 IP 设置参数。

## § 2.4.11 GS\_EM\_DEV\_SYS\_PARAM\_S

结构体定义：

```
typedef struct {
    GS_EM_DEV_SYS_PARAM_TAG_E          m_Tag;

    union {
        GS_EM_DEV_LOCALIP_S            m_LocalIp; /* 本地 IP 地址设置 */
        GS_EM_DEV_NET_ADDR_S           m_AudPostBack; /* 音频回传地址设置 */
        GS_EM_DEV_NET_ADDR_S           m_ServerTcpAddr; /* TCP 服务器地址设置 */
        GS_U8                           m_Vol; /* 全局音量 0 - 100 */
    } m_Param;
} GS_EM_DEV_SYS_PARAM_S;
```

结构体用途说明：设置本地系统参数。

## § 2.4.12 GS\_EM\_DEV\_SYS\_STAT\_S

结构体定义：

```
typedef struct {
    GS_EM_DEV_WORK_STAT_E              m_WorkStat;
    GS_U8                               m_pPhyId[18]; /* 设备物理 ID */
    GS_EM_DEV_LOCALIP_S                m_LocalIp;
    GS_EM_DEV_NET_ADDR_S               m_AudPostBackAddr;
    GS_EM_DEV_NET_ADDR_S               m_ServerAddr;
    GS_U8                               m_Vol; /* 全局音量 0 - 100 */
} GS_EM_DEV_SYS_STAT_S; /* 设备状态 */
```

结构体用途说明：获取设备当前设置的系统参数等状态。

## § 2.4.13 GS\_EM\_DEV\_PSI\_INFO\_S

结构体定义：

```
typedef struct {
    GS_U16 m_ServiceId; /* 节目号 ID */

    GS_U16 m_PmtPid;
```

```

GS_U16 m_VidPid;
GS_U16 m_AudPid;
GS_U16 m_PcrPid;

GS_EM_DEV_DECODER_TV_SYS_E m_VideoStandard; /* 视频输出制式 PAL/NTSC... */
GS_EM_DEV_DECODER_VIDEO_MODE_E m_VideoMode; /* 视频输出模式 CVBS... */
} GS_EM_DEV_PSI_INFO_S;

```

结构体用途说明：获取的通道 PSI 信息，用于设置解码器的参数。

## § 2.4.14 GS\_EM\_DEV\_POSTBACK\_SERV\_AUD\_TRANS\_START\_S

结构体定义：

```

typedef struct {
    GS_EM_DEV_AUD_TRANS_PROT_TYPE_E m_TransProt;
    GS_EM_DEV_AUD_ENC_FMT_E m_EncFmt;
    GS_EM_DEV_NET_ADDR_S m_PostBackAddr;
} GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_START_S;

```

结构体用途说明：回传业务音频传输开始内容。

## § 2.4.15 GS\_EM\_DEV\_POSTBACK\_SERV\_AUD\_TRANS\_End\_S

结构体定义：

```

typedef struct {
} GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_End_S;

```

结构体用途说明：回传业务音频传输结束内容。

## § 2.4.16 GS\_EM\_DEV\_POSTBACK\_SERV\_S

结构体定义：

```

typedef struct {
    GS_EM_DEV_POSTBACK_SERV_TAG_E m_ServTag;
    union {
        GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_START_S m_AudTransStart;
        GS_EM_DEV_POSTBACK_SERV_AUD_TRANS_End_S m_AudTransEnd;
    } m_ServParam;
} GS_EM_DEV_POSTBACK_SERV_S; /* 回调业务 */

```

结构体用途说明：回传业务结构。

# § 2.5 API 函数列表

## § 2.5.1 API 函数列表

| 序号 | 函数名称                         | 函数功能                  |
|----|------------------------------|-----------------------|
| 1  | GS_EM_ProtProcTaskCreate ()  | 创建模块，分配内存资源，启动协议处理线程； |
| 2  | GS_EM_ProtProcTaskDestroy () | 销毁所有协议处理线程、内存等资源；     |
| 3  | GS_EM_RemoteReConnect ()     | 设备主动触发的远程服务器重连        |

## § 2.5.2 GS\_EM\_ProtProcTaskCreate

函数原型:

GS\_S32 GS\_EM\_ProtProcTaskCreate(GS\_EM\_DEV\_IF\_INIT\_PARAM\_S \*pInitParam);

函数功能:

启动协议处理模块, 分配内存资源, 创建线程等资源。

参数说明:

见前节数据结构描述。

返回值:

GS\_SUCCESS 成功;

GS\_FAILURE 失败;

注意:

该函数不能重入, 且每个 GS\_EM\_ProtProcTaskCreate 函数必须对应一个 GS\_EM\_ProtProcTaskDestroy 函数。

### § 2.5.3 GS\_EM\_ProtProcTaskDestroy

函数原型:

GS\_VOID GS\_EM\_ProtProcTaskDestroy(GS\_VOID);

函数功能:

销毁资源

参数说明:

无

返回值:

无;

注意:

无;

### § 2.5.4 GS\_EM\_RemoteReConnect

函数原型:

GS\_VOID GS\_EM\_RemoteReConnect(GS\_VOID);

函数功能:

重连远程服务器。

参数说明:

无

返回值:

无;

注意:

该函数不能重入, 且该函数只能在 GS\_EM\_ProtProcTaskCreate 和 GS\_EM\_ProtProcTaskDestroy 之间调用。

## § 2.6 移植示例

见 demo 中的 C 文件!