

# SS7 Monitor Tool使用说明

## 1. 主要功能

监控E1端口状态，监控SS7\_Link状态，向SS7\_Link发原始包(六种ISUP消息类型)。

## 2. 监控E1端口状态

3秒钟刷新一次端口状态，注意监控如下状态信息：

Fas: 帧同步  
Mf: 复帧同步  
CRC4: CRC4同步  
RmtAlarm: 远端告警  
RmtMFArm: 远端复帧告警  
runTime: 运行总秒数  
RSLIPSc: 接收滑码错误总秒数  
TSLIPSc: 发送滑码错误总秒数  
LCVCRCnt: 接收线路码型错误总计数  
FOSCnt: 接收FAS错误总计数

## 3. 监控SS7\_Link状态

七号链路设备为主类型XMS\_DEVMAIN\_INTERFACE\_CH，子类型为XMS\_DEVSUB\_SS7\_LINK的设备。  
程序中只打开接口设备子类型为XMS\_DEVSUB\_SS7\_LINK的设备。

## 4. 发送原始包

### 4.1 目前支持6中类型消息的原始包：

ISUP_SM_BLO	电路闭塞
ISUP_SM_UBL	解除电路闭塞
ISUP_SM_CGB	群闭塞
ISUP_SM_CGU	解除群闭塞
ISUP_SM_RSC	电路复原
ISUP_SM_GRS	群复原

### 4.2 DPC, OPC同时为4位时表示国际的；DPC, OPC同时为6位时表示国内的；CIC必须为四位

例如：向第二个通道发送电路闭塞消息

消息类型：BLO

DPC: 0001aa

OPC: 0002bb

CIC: 0200

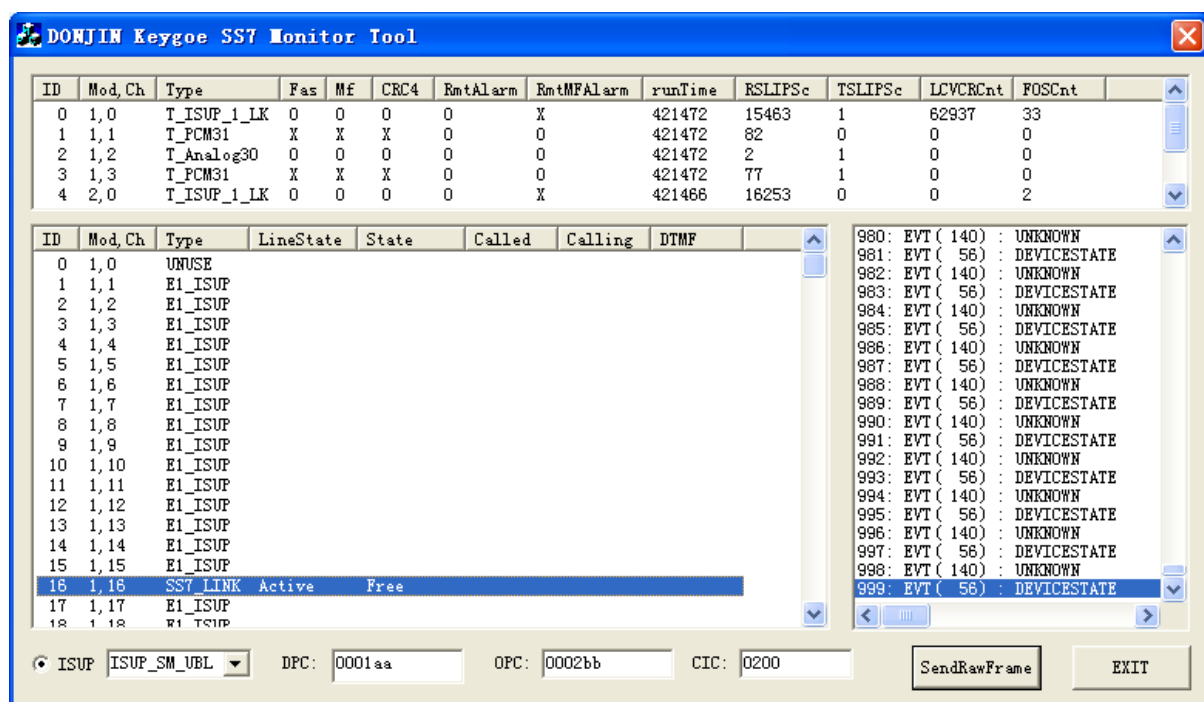


图4-1发送电路闭塞消息图

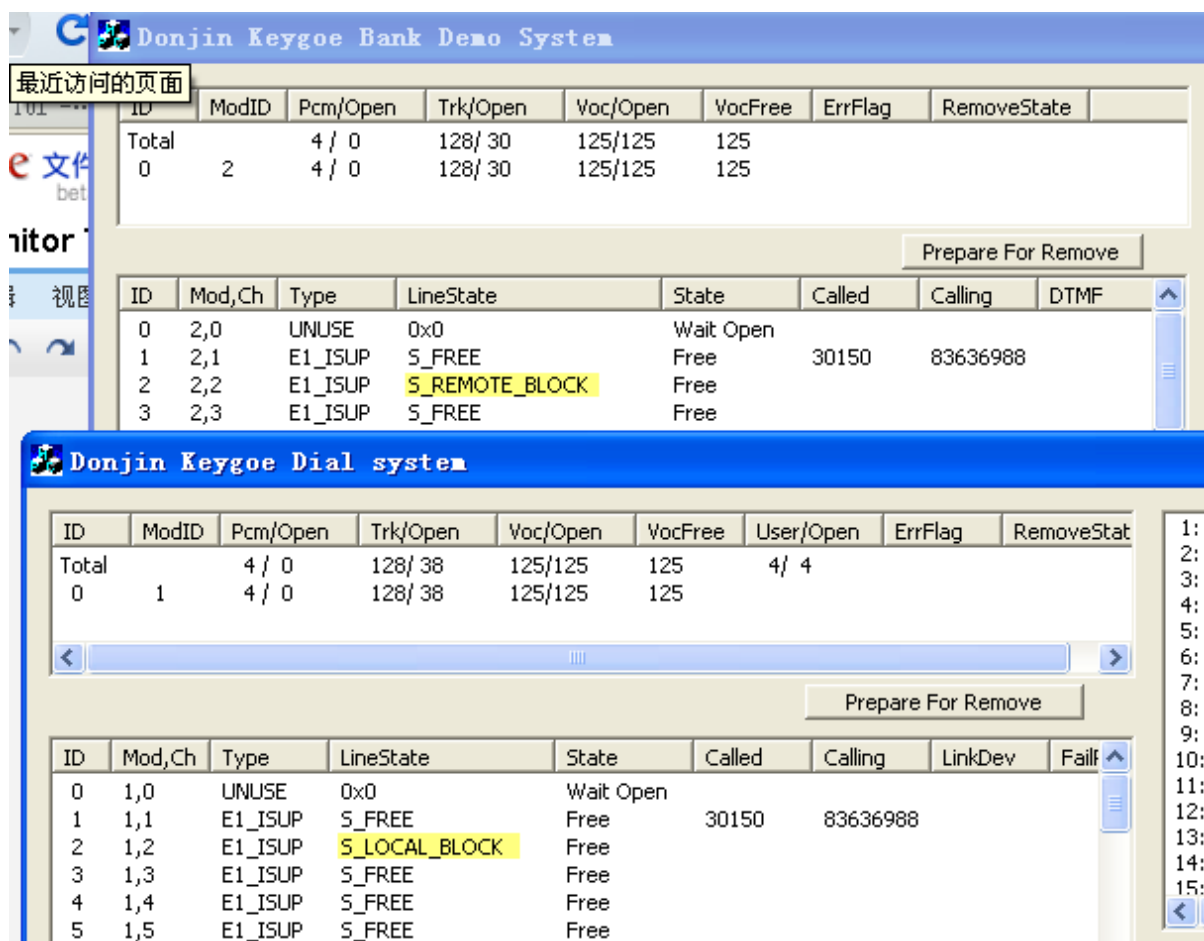


图4-2发送消息后，链路状态发生变化图

## 5. 注意事项

5.1 KeygoeV2.2.0sp1版本

5.2 更新sp1中的固件

5.3 需要在头文件DJAcDataDef.h加入

\*\*\*\*\*

```
typedef struct {
    DJ_U8    m_u8Lost_signal;    //1:无信号, 0: 有信号
    DJ_U8    m_u8Fas_align;      //0: 同步, 1: 失步
    DJ_U8    m_u8Mf_align;       //0: 同步, 1: 失步
    DJ_U8    m_u8Crc4_align;     //0: 同步, 1: 失步
    DJ_U8    m_u8Remote_alarm;    //0: 无告警, 1: 远端告警
    DJ_U8    m_u8Remote_MF_alarm; //0: 无告警, 1: 远端告警
    DJ_U8    m_u8Rx_level;       //0-15: *(-2.5db)
    DJ_U8    m_u8Tx_open;        //1:open circuit,
    DJ_U8    m_u8Tx_overlimit;   //1:current over limit
    DJ_U8    m_u8Port_Current_State; //0: error state; 1: normal working state
    DJ_U8    m_u8Port_Normal_Count; //seconds of digital port normal working
    DJ_U8    m_u8Port_Error_Count;  // seconds of digital port error
    DJ_U32    m_u32total_seconds;   //统计总时间
    DJ_U32    m_u32RLOS_seconds;    //接收失步总秒数
    DJ_U32    m_u32LRCL_seconds;    //接收无信号总秒数
    DJ_U32    m_u32RUA1_seconds;    //接收全1信号总秒数
    DJ_U32    m_u32RRA_seconds;     //远端告警总秒数
    DJ_U32    m_u32RDMA_seconds;    //远端CAS复帧告警总秒数
    DJ_U32    m_u32ILUT_seconds;    //接收电平过低总秒数
    DJ_U32    m_u32JALT_seconds;    //抖动超限总秒数
    DJ_U32    m_u32TOCD_seconds;    //发送开路错误总秒数
    DJ_U32    m_u32TCLE_seconds;    //发送短路错误总秒数
    DJ_U32    m_u32RSLIP_seconds;   //接收滑码错误总秒数
    DJ_U32    m_u32TSLIP_seconds;   //发送滑码错误总秒数
    DJ_U32    m_u32LCVCR_count;     //接收线路码型错误总计数
    DJ_U32    m_u32PCVCR_count;     //接收CRC4错误总计数
    DJ_U32    m_u32FOSCR_count;     //接收FAS错误总计数
    DJ_U32    m_u32EBCR_count;      //远端CRC4 E-BIT总计数
    DJ_U32    m_u32HDLC1_Tx_packets; //HDLC1发送包计数
    DJ_U32    m_u32HDLC1_Rx_goodpk;  //HDLC1接收正确包计数
    DJ_U32    m_u32HDLC1_Rx_badpk;   //HDLC1接收错误包计数
    DJ_U32    m_u32HDLC2_Tx_packets; //HDLC2发送包计数
    DJ_U32    m_u32HDLC2_Rx_goodpk;  //HDLC2接收正确包计数
    DJ_U32    m_u32HDLC2_Rx_badpk;   //HDLC2接收错误包计数
    DJ_U8    m_u8E1Type;            /*E1 type*/
    DJ_U8    m_u8Rfu[3];            /**/
} ACS_Digital_PortState_Data;
```

```
typedef struct {
    ACS_EVT_HEAD    m_AcsEvtHead;
    ACS_Digital_PortState_Data m_AcsDigitalPortStateData;
}ACS_Digital_PortState_Evt;
```

\*\*\*\*\*