ULISES G 5000 – UNIFICACION REDAN-ULISES.

INTERFAZ CONFIGURACION

Este documento se apoya en los ficheros:

Cfgpasa.hpp : estructura de configuración de la pasarela

Gw\_config.json : formato de datos de configuración de la pasarela REDAN.

Transferencia entre recursos de json y tablas de los recursos de la pasarela:

|  |  |  |  |
| --- | --- | --- | --- |
| **struct cfgConfigPasarela** | | |  |
| { | | |  |
| pthread\_mutex\_t tMutex; | | |  |
|  | | |  |
| char acIdSistema[CFG\_MAX\_LONG\_NOMBRE\_PASARELA +1]; | | | EMPLAZAMIENTO |
| char szNombre[CFG\_MAX\_LONG\_NOMBRE\_PASARELA+1]; | | | "general" "name" |
| char szDirCPU0[CFG\_MAX\_LONG\_URL+1]; | | | "general" |
| char szDirCPU1[CFG\_MAX\_LONG\_URL+1]; | | | "general" |
| char szDirVirtual[CFG\_MAX\_LONG\_URL+1]; | | |  |
| char szDirSrvConfig[CFG\_MAX\_LONG\_URL+1]; | | | "general" "ips" |
| int iNivelConsola; | | | "general" "nivelconsola" |
| int iPuertoConsola; | | | "general" "puertoconsola" |
| int iNivelIncidencias; | | | "general" "nivelIncidencias" |
| int iDual; | | | "general" "dual" |
| char acIdConfig[CFG\_MAX\_LONG\_ID\_CONFIG+1]; | | |  |
| int iModoSincronizacion; | | | MODO\_NTP (2) |
|  | | |  |
| int iSipPuertoLocal; | | | "sip""PuertoLocalSIP": |
| int iSipPeriodoSuperv; | | | "sip""PeriodoSupervisionSIP":  (90-1800 sg) |
| char acGrupoMulticast[CFG\_MAX\_LONG\_URL+1];  unsigned int uiPuertoMulticast; | | |  |
| int iSnmpPuertoRemoto; | | |  |
| int iRecPuertoRemoto; | | |  |
| int iNumRecursos; | | | Numero de recursos |
|  | | |  |
| struct LocalizadorRecurso asLocRec[MAX\_RECURSOS\_TIFX]; | | | Ver LocalizadorRecurso |
| struct cfgConfigRecurso sRecurso[MAX\_RECURSOS\_TIFX]; | | | Ver cfgConfigRecurso |
|  | | |  |
|  | | |  |
|  | | |  |
| char szDirSipRegistrar[NUM\_SIP\_REGISTRAR][MAX\_LONG\_DIRIP]; | | | "sip""Registrars" |
| char szDirSipProxy[NUM\_SIP\_PROXY][MAX\_LONG\_DIRIP]; | | | "sip""proxys" |
| char szDirSrvNtp[NUM\_SRV\_NTP][MAX\_LONG\_DIRIP]; | | | "sincro""servidores" |
| struct st\_Parametros\_generales\_sistema  parametrosgenerales | | | No se utiliza. Lo quitamos |
| **struct st\_numeracionats**  **plannumeracionats [N\_MAX\_CENTRALES];** | | | **Ver struct st\_numeracionats** |
| **struct st\_direccionamientoip**  **plandireccionamientoip[N\_MAX\_TV+N\_MAX\_TIFX];** | | | **Ver struct st\_direccionamientoip** |
| **struct st\_listatroncales**  **plantroncales [N\_MAX\_TRONCALES];** | | | **Ver struct st\_listatroncales** |
| **struct st\_listaredes planredes [N\_MAX\_REDES];** | | | **Ver struct st\_listaredes** |
| **struct st\_asignacionusuario\_tv**  **planasignacionusuarios [N\_MAX\_TV];** | | | **Ver struct st\_asignacionusuario\_tv** |
| **struct st\_asignacionrecursos\_gw**  **planasignacionrecursos[N\_MAX\_REC\_SISTEMA];** | | | **Ver struct st\_asignacionrecursos\_gw** |
| **struct st\_direccionamiento\_sip**  **plandireccionamientosip [N\_MAX\_TV];** | | | **Ver struct st\_direccionamiento\_sip** |
|  | | |  |
|  | | | |
|  | | | |
|  | | | |
| en el array struct cfgConfigRecurso sRecurso[MAX\_RECURSOS\_TIFX] los recursos se colocan según la posición física en el bastidor, es decir,  posición 0 : Slot 0 dispositivo 0  posición 2 : Slot 0 dispositivo 1  posición 3 : Slot 0 dispositivo 2  posición 4 : Slot 0 dispositivo 3  posición 5 : Slot 1 dispositivo 0  posición 6 : Slot 1 dispositivo 1  -----  posición 15 : Slot 3 dispositivo 3  para poder localizar y actualizar los recursos conforme se van creando, se utiliza el array:  **struct LocalizadorRecurso asLocRec[MAX\_RECURSOS\_TIFX];**  donde, por ejemplo, el primer recurso que se crea, irá en la posición 0 de asLocRec[] y ahí se indicará el slot y dispositivo que ocupa y así se podrá localizar la posición del recurso dentro de sRecurso[] | | | |
| **struct LocalizadorRecurso** |  | | |
| { |  | | |
| int iSlot; |  | | |
| int iDispositivo; |  | | |
|  |  | | |
|  |  | | |
| **struct cfgConfigRecurso** |  | | |
| { |  | | |
| struct cfgConfigGeneralRecurso sGeneral; | Ver cfgConfigGeneralRecurso | | |
| union Rec |  | | |
| { |  | | |
| struct cfgConfigRecAudio sAudio; | Ver cfgConfigRecAudio | | |
| struct cfgConfigRecDatos sDatos; | Por ahora nada | | |
| } uRec; |  | | |
| union uIf |  | | |
| { |  | | |
| struct cfgConfigIfRadio sRadio; | Ver cfgConfigIfRadio | | |
| struct cfgConfigIfLcen sLcen; | Ver cfgConfigIfLcen | | |
| struct cfgConfigIfR2N5 sR2N5; | Ver cfgConfigIfR2N5 | | |
| struct cfgConfigIfTlf sTlf; | Ver cfgConfigIfTlf | | |
| struct cfgConfigIfRdsi sRdsi; | Por ahora nada | | |
| } uIf; |  | | |
|  |  | | |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **struct cfgConfigGeneralRecurso** |  | | |
| { |  | | |
| char szNombre[CFG\_MAX\_LONG\_NOMBRE\_RECURSO+1]; | "recursos" "IdRecurso" | | |
| int iTipo; | Si "Radio\_o\_Telefonia": 1  iTipo = CFG\_REC\_TIPO\_AUDIO\_TXRX  Si "Radio\_o\_Telefonia": 0 | | |
| "radio"  "TipoRadio" | iTipo | |
| 0: Local Simple  1: Local PR  2: Local FD Simple  3: Local FD PR  4: Remoto RxTx  5: Remoto Tx  6: Remoto Rx | CFG\_REC\_TIPO\_AUDIO\_TXRX CFG\_REC\_TIPO\_AUDIO\_TXRX  CFG\_REC\_TIPO\_AUDIO\_TXRX  CFG\_REC\_TIPO\_AUDIO\_TXRX CFG\_REC\_TIPO\_AUDIO\_TXRX  CFG\_REC\_TIPO\_AUDIO\_TX  CFG\_REC\_TIPO\_AUDIO\_RX | |
| int iTipoIf; | Si "Radio\_o\_Telefonia": 0  iTipoIf= CFG\_IFREC\_TIPO\_RADIO  Si "Radio\_o\_Telefonia": 1 | | |
| "telefonia"  "TipoTelefonia" | | iTipoIf |
| 0: BL  1: BC  2: AB  3: R2  4: N5  6: LCEN  7: ATS/QSIG,  8: EYM\_PP | | CFG\_IFREC\_TIPO\_BL  CFG\_IFREC\_TIPO\_BC  CFG\_IFREC\_TIPO\_AB  CFG\_IFREC\_TIPO\_R2  CFG\_IFREC\_TIPO\_N5  CFG\_IFREC\_TIPO\_LCEN  CFG\_IFREC\_TIPO\_QSIG  CFG\_IFREC\_TIPO\_EYM\_PP |
| Faltarían (algunos sin implementar):  #define CFG\_IFREC\_TIPO\_DESCONOCIDO -1  #define CFG\_IFREC\_TIPO\_RDSI\_2BD  #define CFG\_IFREC\_TIPO\_RDSI\_30BD  #define CFG\_IFREC\_TIPO\_IO  #define CFG\_IFREC\_TIPO\_DATOS  #define CFG\_IFREC\_TIPO\_RRC  #define CFG\_IFREC\_TIPO\_EYM\_MARC | | |
|  |  | | |
| int iSlot; | "recursos" "SlotPasarela" | | |
| int iDispositivo; | "recursos" "NumDispositivoSlot" | | |
| char szDestino[CFG\_MAX\_LONG\_NOMBRE\_RECURSO+1]; | ¿La frecuencia fid??? | | |
|  |  | | |
| int iSnmpPuertoRemoto; |  | | |
| int iRecPuertoRemoto; |  | | |
| int iRecPuertoBase; | Puerto base para comunicar recursos radio con el modulo de grabación. Cada recurso utiliza un puerto iRecPuertoBase+numrecurso | | |
| int iEnableGI; | Habilita en un recurso la grabación. De momento sólo se habilitará para recursos radio remotos | | |
| **int iFlgUsarDiffServ; ¿????;** |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |

|  |  |  |
| --- | --- | --- |
| char szUriLocal[MAX\_LONG\_DIR\_URI]; | "recursos""Uri\_Local" | |
| Int iLLamadaAutomatica | 0/1 |
| int iRestriccion; | " recursos""restriccion"  “sin restricción” : 0  “con lista blanca”: 1  “con lista negra” : 2 | |
| char szListaBlanca[NUM\_LISTA\_BLANCA][MAX\_LONG\_DIR\_URI]; | " recursos""listablanca"  Cierra lista con NULL | |
| char szListaNegra[NUM\_LISTA\_NEGRA][MAX\_LONG\_DIR\_URI]; | " recursos""listanegra"  Cierra lista con NULL | |
| int iEnableRegistro | Estos dos hay que configurarlos por recurso  " recursos" "Registro\_Enable" | |
| char szClave[MAX\_LONG\_CLAVE]; | " recursos" "ClaveRegistro" | |
|  |  | |
| Struct cfgColateralPP sColateral; | Ver struct cfgColateralPP | |
|  |  | |
| **int size\_aListaEnlacesExternos;** |  | |
| **struct cfgEnlaceExterno aListaEnlacesExternos[MAX\_ENLACES\_EXTERNOS\_POR\_REC];** | **Los recursos de tf cuando se realiza llamada entrante llaman a todos los Puestos que tengan la tecla configurada** | |

|  |  |
| --- | --- |
| **struct cfgEnlaceExterno** |  |
| { |  |
| char szId[CFG\_MAX\_LONG\_ID\_ENLACE\_EXTERNO+1]; |  |
| }; |  |

|  |  |
| --- | --- |
| **struct cfgConfigRecAudio** |  |
|  |  |
| int iSelGananciaAgcTx; | "recursos""hardware" "AD\_AGC" |
| int iGananciaAgcTx; | "recursos""hardware" "AD\_Gain" |
| int iSelGananciaAgcRx; | "recursos""hardware" "DA\_AGC" |
| int iGananciaAgcRx; | "recursos""hardware" "AD\_Gain" |
| int iHaySupresionSilencio; |  |
| int iTamanoRtp; | "recursos""TamRTP" |
| int iCodecPreferido; | "recursos""Codec" |
| int iBufferJitterMin  int iBufferJitterMax | "Buffer\_jitter": "min": "Valor minimo", (en hex)  "max": "Valor maximo" (en hex) |

|  |  |
| --- | --- |
| **struct cfgConfigIfRadio** |  |
| { |  |
| int iTipoRadio; | "radio""TipoRadio" |
| char cTipoPtt; | "radio" "PTT":0 cTipoPtt:’h’  "radio" "PTT":1 cTipoPtt:’s’ |
| char cTipoSq; | "radio" "SQ":0 cTipoSq:’h’  "radio" "SQ":1 cTipoSq:’v’ |
| bool iEyM; | 0 |
|  |  |
|  |  |
| int iUmbralDetSq; | "radio" "UmbralVAD" (en Hex) |
| int iNivelTonoPtt; | -10 |
| int iMetodoBss; | "radio" "MetodoBSS" |
| int iNtz; | 0 |
| int TiempoMaxPtt; | "radio" "TiempoPTT" (en hex) |
| int iSupervisionPortadoraTx; | 0 |
| int iSupervisionModuladoraTx; | 0 |
| int iModoConfirmacionPtt; | "radio" "ModoConfPTT" |
| int iPeriodoRtpSqBss; | "radio" "RepSQyBSS" |
| int iNumPaquetesSqOff; | "radio" "DesactivacionSQ" Nueva version-ForzadoSQ |
| int iPttTimeOut; | "radio" "TimeoutPTT": |
| int iNumFlujosMezcla; | "radio" "NumFlujosAudio" |
| unsigned int KeepAlivePeriod; | "sip" "KeepAlivePeriod" |
| unsigned int KeepAliveMultiplier; | "sip" "KeepAliveMultiplier" |
| Int iSessionPrio |  |
| Int iPttPrio |  |
| Int iPrecisionAudio | Tipo de recurso radio (granularidad 5 o 20 ms)  PRECISION\_ESTRICTA 0  PRECISION\_NORMAL 1 |
| int iTiempoPttBloqueado; // 2000 por defecto. | 2000 |
| int iRetrasoPttOff; // 0 por defecto. | 0 |
| int iRetrasoSqOff; // 0 por defecto. | Tiempo desde que finaliza el sq del canal seleccionado en un grupo BSS hasta ee se permite otra comunicación si el resto de los receptores del grupo no han quitado su sq. |
|  |  |
| int iTiempo\_VentanaRx | "radio" "tm\_ventana\_rx" (en Hex) |
| int iClimaxDelay | "radio" "climax\_delay" 0/1 |
| Int iTmRetardoFijo | "radio" "modo\_compensacion" // viene el valor del tiempo de retardo fijo |
| Char cTipoConmutacion | " radio""colateral""tipoConmutacion" ("M"/”A”) |
| int iBssRtp | "radio" "bss\_rtp" |
|  | "radio" "tabla\_indices\_calidad" |
| Int iEvtPTT | Envia eventos de PTT ON/OFF (0:no 1:si) |
| Int iTjbd | Jitter Buffer Delay |
| Int iTGRSid | GRS Internal Delay (para radios remotas) |
|  |  |
|  |  |
| Struct cfgColateralRadio sColateral; | Ver struct cfgColateralRadio |
|  |  |

|  |  |
| --- | --- |
| **struct cfgColateralRadio** |  |
| { |  |
| char szDestino[CFG\_MAX\_LONG\_NOMBRE\_RECURSO+1]; | "radio""colateral""name" |
| Struct cfgEmplazamiento asEmpl[MAX\_NUM\_EMPL]; |  |
|  |  |
| Struct cfgEmplazamiento  {  char UriTxA[MAX\_LONG\_DIR\_URI];  char UriTxB[MAX\_LONG\_DIR\_URI];  char cActivoTx;  char UriRxA[MAX\_LONG\_DIR\_URI];  char UriRxB[MAX\_LONG\_DIR\_URI];  char cActivoRx;  } | "radio""colateral"  "emplazamientos": [{  "uriTxA": "uriA", "uriTxB": "uriB",  "activoTx": "A o B",  "uriRxA": "uriA",  "uriRxB": "uriB",  "activoRx": "A o B"  }] |

|  |  |
| --- | --- |
| **struct cfgConfigIfLcen** | Valores por defecto. Al dsp se le configura con datos grabados en el ejecutable directamente. No se modifican |
| { |  |
| int iUmbralRxTonoLcen; | -35 |
| int iNivelTxTonoLcen; | -18 |
| int iPeriodoRefrescoEstado; | 200 |
| int iTimeoutEstado; | 600 |
| int iLongRafagas; | 6 |
| int iT1; | 20 |
| int iT1Max; | 100 |
| int iT2; | 20 |
| int iT2Max; | 100 |
| int iT3; | 40 |
| int iT4; | 40 |
| int iT4Max; | 160 |
| int iT5; | 60 |
| int iT5Max; | 100 |
| int iT6; | 5000 |
| int iT6Max; | 6000 |
| int iT8; | 80 |
| int iT8Max; | 150 |
| int iT9; | 40 |
| int iT9Max; | 60 |
| int iT10; | 20 |
| int iT10Max; | 130 |
| int iT11; | 20 |
| int iT11Max; | 130 |
| int iT12; | 200 |
| int iT\_Release; | Tiempo generando tonos antes de liberar |
|  |  |

|  |  |
| --- | --- |
| **struct cfgConfigIfR2N5** | Los tiempos no se configuran, ni se pasan al dsp, es el dsp el que los tiene en su ejecutable |
| { |  |
| int iLado; | "telefonia""Lado"  CFG\_LADO\_A 0  CFG\_LADO\_B 1 |
|  |  |
| struct Tiempos |  |
| { |  |
| int iP1Min; | 0 |
| int iP1Nom; | 0 |
| int iP1Max; | 0 |
| int iP2; | 0 |
| int iP2Tout; | 0 |
| int iP3; | 0 |
| int iP3Tout; | 0 |
| int iP4; | 0 |
| int iP4Tout; | 0 |
| int iP5; | 0 |
| int iP5Tout; | 0 |
| int iP6; | 0 |
| int iP6Tout; | 00 |
| int iP7; | 0 |
| int iP7Tout; | 0 |
| int iP8; | 0 |
| int iP8Tout; | 0 |
| int iP9Min; | 0 |
| int iP9Nom; | 0 |
| int iP9Max; | 0 |
| int iP11; | 0 |
| int iP16Min; | 0 |
| int iP16Nom; | 0 |
| int iP16Max; | 0 |
| int iP18; | 0 |
| } sTiemA, sTiemB; |  |
| int iT\_Release; | Tiempo de generando tonos antes de liberar (def:5) |
| int iT\_Int\_Warning; | Tiempo duración de tonos Interrupt Warning (5-15 sg) (def:5) |
| **char szIdTroncal[CFG\_MAX\_LONG\_NOMBRE\_TRONCAL+1];** | **0** |
| **char szIdRed[CFG\_MAX\_LONG\_NOMBRE\_RED+1];** | **0** |
| char ntest\_local [LONG\_AB\_ATS]; | "telefonia""no\_test\_local" |
| char ntest\_remoto [LONG\_AB\_ATS]; | "telefonia""no\_test\_remoto" |
| struct st\_rangoATS rangos\_dst[MAX\_RANGOS\_ATS] | ]; /\*array de rangos de nuemros destino \*/ |
| struct st\_rangoATS rangos\_org[MAX\_RANGOS\_ATS] | ]; /\*array de rangos de nuemros origen \*/ |

|  |  |
| --- | --- |
| **struct st\_rangoATS** |  |
| { char inicial [LONG\_AB\_ATS]; | /\*0 para indicar fin de tabla\*/ |
| char final [LONG\_AB\_ATS]; |  |
| } |  |

|  |  |
| --- | --- |
|  |  |
| **struct cfgConfigIfTlf** |  |
| { |  |
| int iPatronRing; | 0 |
| int iPeriodoRing; | 50 |
| int iUmbralAltoPeriodoRing; | 0x0E |
| int iUmbralBajoPeriodoRing; | 0x3F |
| int iVRing; | 71300 |
| int iVBat; | 50000 |
| enum eType eTypeEyM; | "telefonia""t\_eym" |
| int iModoEyM; | "telefonia""h2h4" |
| int iLadoEyM; | "telefonia""ladoeym" |
| Int iDetectVox | 0: no detecta 1 : detecta Vox (def: 0) |
| Int iUmbralVox | Nivel (def:-50) |
| Int iTmInactividad | Tiempo inactividad (def: 10) |
| **char szIdRed[CFG\_MAX\_LONG\_NOMBRE\_RED+1];** |  |
|  |  |

|  |  |
| --- | --- |
| **struct cfgConfigIfRdsi** |  |
| { |  |
| int iLado; | "telefonia""Lado" |
| int iModo; | "telefonia""Modo" |
|  |  |
| char szIdTroncal[CFG\_MAX\_LONG\_NOMBRE\_TRONCAL+1]; |  |
| char szIdRed[CFG\_MAX\_LONG\_NOMBRE\_RED+1]; |  |
|  |  |

|  |  |
| --- | --- |
| **Struct cfgColateralPP** |  |
| { |  |
| int iRespuestaAutomatica | "telefonia""r\_automatica" |
| char szUriRemota[MAX\_LONG\_DIR\_URI]; | "telefonia" "uri\_remota" |
| int isuperv\_options | 0: no supervise 1: supervisa |
| int itm\_superv\_options | Tiempo entre options cuando se detecta colateral |
| int iColateralSCV | 0: el colateral es un recurso  1: el colateral es un SCV |

En el arranque de la pasasrela se llamará a la siguiente función:

Inicia(struct cfgConfigPasarela \*psCfgPasarela,

struct st\_config\_sistema \*psCfgSystem,

class CGestorCfg::CEventos \*poEvents );

|  |  |
| --- | --- |
| Inicia(struct cfgConfigPasarela \*psCfgPasarela,  struct st\_config\_sistema \*psCfgSystem,  class CEventosCfg \*poEventosCfg ); | |
| psCfgPasarela: | puntero a la estructura de configuración de la pasarela |
| psCfgSystem: | puntero a la estructura de configuración del sistema. En REDAN esta estructura solamente tiene parámetros inicializados con valores de defecto. |
| poEventosCfg | puntero a la estructura de eventos a los que llamará el módulo de configuración al analizar la configuración y detectar cambios. |

|  |  |
| --- | --- |
| int EvtReset() | Reset del módulo sw de la pasarela. Se llamará desde el módulo configuración cuando se estime que los cambios de configuración requieren una inicialización del sw. |
| int EvtReiniciaMulticast() | El módulo config llamaría a esta función con el cambio de los parámetros reaccionados con las comunicaciones multicast. No es aplicable en REDAN. |
| int EvtReiniciaSip() | El módulo config llamaría a esta función con el cambio de los parámetros reaccionados con las comunicaciones SIP.  De momento no se utiliza, puede ser necesario un reset hw, no solamente reiniciar el sw. |
| int EvtReiniciaSincro() | El módulo config llamará a esta función con el cambio de los parámetros reaccionados con la sincronización RTP |
| int EvtReiniciaSnmp() | El módulo config llamará a esta función con el cambio de los parámetros reaccionados con SNMP |
| int EvtReconfiguraEncamina() | El módulo config llamaría a esta función con el cambio de los parámetros reaccionados con el encaminamiento de las llamadas. No es aplicable en REDAN. |
| int EvtArrancaRecurso( int ) | El módulo config llamará a esta función cuando analice que se ha dado de alta un nuevo recurso en la configuración. |
| int EvtDetieneRecurso( int ) | El módulo config llamará a esta función cuando analice que se ha dado de baja un recurso en la configuración. |
| int EvtReconfiguraRecurso( int ) | El módulo config llamará a esta función cuando analice que se ha modificado algún parámetro de la configuración de un recurso existente. |

|  |  |
| --- | --- |
| **struct st\_no\_abonado** |  |
| unsigned int prefijo | ;/\*NO\_PREFIJO fin de tabla\*/ |
| char abonado [LONG\_AB]; |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **struct st\_planrutas** |  |
| { |  |
| unsigned char tiporuta; | /\* directa o alternativa,255 fin tabla(NO\_TIPO\_RUTA)\*/ |
| char listatroncales [N\_TRONCALES\_RUTA][LONG\_NAME\_TRONCAL]; |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **struct st\_rango** |  |
| char inicial [LONG\_AB\_ATS]; | /\*0 para indicar fin de tabla\*/ |
| char final [LONG\_AB\_ATS]; |  |
| struct st\_no\_abonado abonado; |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **struct st\_numeracionats** |  |
| { |  |
| unsigned char centralpropia | Mi Central,CENTRAL\_NO\_DEF para indicar fin tabla |
| unsigned char throwswitching |  |
| char no\_test [LONG\_AB\_ATS]; |  |
| struct st\_rango rangosoperador[N\_MAX\_RANGOS] | /\*array de rangos de operdores \*/ |
| struct st\_rango rangosprivilegiados[N\_MAX\_PRIV]; | /\*array de rangos de ope.privilegiados \*/ |
| struct st\_planrutas listarutas [N\_MAX\_RUTAS]; |  |
|  |  |

|  |  |
| --- | --- |
| **struct st\_direccionamientoip** |  |
| { |  |
| char idhost[LONG\_HOST\_NAME]; | /\*NO\_NAME fin de tabla\*/ |
| char ipred1[LONG\_IP]; |  |
| char ipred2[LONG\_IP]; |  |
| unsigned char tipohost; |  |

|  |  |
| --- | --- |
| **struct st\_plan\_recursos** |  |
| { |  |
| unsigned char tipo; |  |
| unsigned char idrec [LONG\_NAME\_REC]; | /\*0 para indicar fin de tabla NO\_REC\*/ |

|  |  |
| --- | --- |
| **struct st\_listatroncales** |  |
| { |  |
| char idtroncal [LONG\_NAME\_TRONCAL]; | /\*0 para indicar fin de tabla NO\_TRONCAL\*/ |
| char no\_test [LONG\_AB\_ATS]; |  |
| struct st\_plan\_recursos  listarecursos [N\_MAX\_REC\_BY\_TRONCAL]; |  |

|  |  |
| --- | --- |
| **struct st\_listaredes** |  |
| { |  |
| char idred [LONG\_NAME]; |  |
| unsigned int prefijo; | /\*prefijo 255 para indicar fin tabla\*/ |
| struct st\_plan\_recursos  listarecursos [N\_MAX\_REC\_BY\_NET]; |  |

|  |  |
| --- | --- |
| **struct st\_asignacionusuario\_tv** |  |
| { |  |
| char idusuario [LONG\_NAME\_TV]; |  |
| char idhost[LONG\_HOST\_NAME]; | /\*NO\_NAME fin de tabla NO\_HOST\*/ |

|  |  |
| --- | --- |
| **struct st\_asignacionrecursos\_gw** |  |
| { |  |
| char idrecurso [LONG\_NAME\_REC]; |  |
| char idhost[LONG\_HOST\_NAME]; | /\*NO\_NAME fin de tabla NO\_HOST\*/ |
|  |  |

|  |  |
| --- | --- |
| **struct st\_direccionamiento\_sip** |  |
| { |  |
| char idusuario [LONG\_NAME\_TV]; | /\*NO\_NAME fin de tabla NO\_USUARIO\*/ |
| struct st\_no\_abonado  array\_abonados[N\_ABONADOS\_BY\_TV]; |  |
|  |  |

#define MAX\_RECURSOS\_TIFX 16

#define MAX\_ENLACES\_EXTERNOS\_POR\_REC 32

/\*

\* Tipos EyM

\*/

enum eType

{

CFG\_EyM\_TYPE\_I,

CFG\_EyM\_TYPE\_II,

CFG\_EyM\_TYPE\_III,

CFG\_EyM\_TYPE\_IV,

CFG\_EyM\_TYPE\_V,

};

#define CFG\_TOTAL\_TYPES\_EYM 5

/\*

\* Modo EyM

\*/

#define CFG\_EyM\_2W 0

#define CFG\_EyM\_4W 1

#define CFG\_TOTAL\_MODOS\_EYM 2

/\*

\* Lado EyM

\*/

#define CFG\_EyM\_LADO\_EyM 0

#define CFG\_EyM\_LADO\_PLR 1

#define CFG\_TOTAL\_LADOS\_EYM 2

/\*

\* Rangos validos para algunos otros numeritos.

\*/

#define CFG\_SLOT\_NINGUNO -1

#define CFG\_MIN\_SLOT 1 // Numero minimo de slot para un recurso.

#define CFG\_MAX\_SLOT 4 // Numero minimo de slot para un recurso.

#define CFG\_DISPOSITIVO\_NINGUNO -1

#define CFG\_MIN\_DISPOSITIVO 1 // Numero minimo de de dispositivo (dentro de slot).

#define CFG\_MAX\_DISPOSITIVO 4 // Numero minimo de de dispositivo (dentro de slot).

#define CFG\_MIN\_FLUJOS\_PTT\_MEZCLA 2

#define CFG\_MAX\_FLUJOS\_PTT\_MEZCLA 7

/\*

\* Otros numeros.

\*/

#define CFG\_SIP\_SIN\_SUPERV 0

#define CFG\_MAX\_TIEMPO\_PTT\_INFINITO 0

#define CFG\_PTT\_LOCKOUT 1