#include "gt\_debug.h"

#include "hi\_unf\_disp.h"

#include "hi\_unf\_hdmi.h"

#include "hdi\_init\_platform.h"

#include "hdi\_init.h"

#include "hdi\_dmd.h"

#include "hdi\_demux.h"

#include "hdi\_osd.h"

#include "hdi\_av.h"

#include "hdi\_usb.h"

#include "hdi\_fp.h"

#include "hdi\_vfd.h"

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <string.h>

#include <pthread.h>

#include <assert.h>

#include <linux/fb.h>

#include <sys/mman.h>

#include <sys/ioctl.h>

#include <fcntl.h>

#include <unistd.h>

#include "hi\_unf\_ecs.h"

#include <time.h>

#ifndef ANDROID\_PLATFORM

#include <stdio.h>

#include <sys/shm.h>

#include <sys/stat.h>

#endif

#if defined(JT\_CHIP\_hisilicon3716mv410) || defined(JT\_CHIP\_hisilicon3716mv420)

#define HI\_DAC\_CVBS 3

#endif

#define HDI\_REVISION "HDI\_V00.00.00-"\_\_DATE\_\_

static JT\_U32 g\_is\_hdi\_init = JT\_FALSE;

static JT\_U32 g\_is\_gtscreen\_hdi\_init = JT\_FALSE;

static JT\_U32 g\_hdi\_init\_flag = 0;

static JT\_BOOL g\_hdi\_init\_vo\_flag = JT\_TRUE;

JT\_U8 timeForStandby [100] = "0:0/0-";

static OperatingModes\_t g\_cur\_operating\_mode = NORMAL\_MODE;

/\*0: 1280\*720; 1: 720\*576\*/

extern JT\_U8 u8hdiosd\_resolution;

/\*0: Auto, 1: PAL, 2: NTSC\*/

extern JT\_U8 u8hdiavDisplayFormat;

extern JT\_U8 u8hdiavAutoCheckDisplayFormat;

extern JT\_U8 u8hdiavHdcpEnabled;

extern JT\_BOOL bhdipvr\_safeview\_enabled;

extern JT\_BOOL bhdipvr\_cipher\_enabled;

extern JT\_BOOL g\_b8SNDEnableSmartVolume;

static void WDG\_Process(void \*param)

{

while (1)

{

JTOS\_TaskDelay(5000);

// JT\_ERROR(("WDG\_Process delay ClearWatchDog\n"));

#if defined(JT\_CHIP\_hisilicon3716cv200) || defined(JT\_CHIP\_hisilicon3712) || defined(JT\_CHIP\_hisilicon3796cv100) || defined(JT\_CHIP\_hisilicon3796mv100) || defined(JT\_CHIP\_hisilicon3798cv200) || defined(JT\_CHIP\_hisilicon3798mv200) || defined(JT\_CHIP\_hisilicon3716mv410) || defined(JT\_CHIP\_hisilicon3716mv420)

HI\_UNF\_WDG\_Clear(0);

#else

HI\_UNF\_WDG\_ClearWatchDog();

#endif

}

}

JT\_StatusCode\_t HDIInit(PST\_HDI\_PARAMETERS pstHdiParametersInit)

{

JT\_StatusCode\_t status\_code = JT\_ERROR\_ALREADY\_INITIALIZED;

JTHDIDMX\_SourceParams\_t demux\_source\_params;

JTHDIAV\_dispsettings\_t settings;

int time1,time2;

//EDU10658\_Test();

if(pstHdiParametersInit != NULL)

{

u8hdiosd\_resolution = pstHdiParametersInit->u8Resolution;

/\*0: Auto, 1: PAL, 2: NTSC\*/

u8hdiavDisplayFormat = pstHdiParametersInit->u8DisplayFormat;

u8hdiavAutoCheckDisplayFormat = pstHdiParametersInit->u8AutoDisplayFormat;

u8hdiavHdcpEnabled = pstHdiParametersInit->u8HDCPEnabled;

bhdipvr\_safeview\_enabled = pstHdiParametersInit->bsafeview\_cas\_enabled;

bhdipvr\_cipher\_enabled = pstHdiParametersInit->bPvr\_cipher\_enabled;

g\_b8SNDEnableSmartVolume = (pstHdiParametersInit->u8SmartVolume == 1) ? JT\_TRUE : JT\_FALSE;

}

if(!g\_is\_hdi\_init)

{

if ( !g\_is\_gtscreen\_hdi\_init )

{

status\_code = HDI\_InitPlatform();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTOS\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDIGPIO\_Initialize();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDIFLASH\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDIIR\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

}

#ifdef ANDROID\_PLATFORM

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP | HDI\_INIT\_IPPLAY ) )

#else

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP ) )

#endif

!= 0 )

{

status\_code = JTHDIFP\_Initialize();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDII2C\_Initialize();

JT\_ASSERT(status\_code == JT\_SUCCESS); }

#if 0

if (EDU10658\_GetDeviceID() == JT\_TRUE)

{

if (EDU10658\_GetDeviceID\_2() == JT\_TRUE)

{ JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_COMBO\_EXTERNALT2\_EXTERNALT2);

}

else

{ JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_COMBO\_CABLE\_TERRESTRIAL\_T2);

}

}

else if (tda10024\_check\_id() == JT\_TRUE)

{/\* EDC90078 also use it. \*/ JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_COMBO\_INTERCABLE\_EXTERNALCABLE);

}

else if (EDS10530\_GetDeviceID() == JT\_TRUE)

{

if (EDS10530\_GetDeviceID\_2() == JT\_TRUE)

{

JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_SYS\_DUAL\_S2);

}

else

{

JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_SYS\_SINGLE\_S2);

}

}

else if (EDT40607\_GetDeviceID() == JT\_TRUE)

{

JTHDIDMD\_SetSysDmdType(JT\_DMD\_DELIVERY\_SYS\_ISDBT);

}

#endif

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP | HDI\_INIT\_IPPLAY ) )

!= 0 )

{

status\_code= JTHDIDMX\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDIAV\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

#ifndef ANDROID\_PLATFORM

status\_code = JTHDIUSB\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

#endif

}

#ifdef ANDROID\_PLATFORM //

#else

/\* if (kmem\_cache\_init())

{

JT\_ERROR(("slab\_init Failed\n"));

}

else

{

kmem\_cache\_sizes\_init();

}

JTAPIDM\_Init();\*/

#endif

#if defined(JT\_CUSTOMER\_IVELOZITY) || defined(JT\_CUSTOMER\_MALDIVES)

#ifdef ANDROID\_PLATFORM

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_IPPLAY ) )

!= 0 )

#else

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_APP ) )

!= 0 )

#endif

#else

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_APP | HDI\_INIT\_IPPLAY ) )

!= 0 )

#endif

{

status\_code = JTHDIDMD\_Initialize();

JT\_ASSERT(status\_code == JT\_SUCCESS);

}

/\*

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP | HDI\_INIT\_IPPLAY ) )

!= 0 )

{

status\_code= JTHDIDMX\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS); status\_code = JTHDIAV\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

#ifndef ANDROID\_PLATFORM

status\_code = JTHDIUSB\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS); #endif

}

\*/

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP ) )

!= 0 )

{

status\_code = JTHDIOSD\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

}

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP | HDI\_INIT\_IPPLAY ) )

!= 0 )

{

status\_code = JTHDIPlayer\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

status\_code = JTHDINVRAM\_Initialize();

JT\_ASSERT(status\_code == JT\_SUCCESS);

}

#if !defined(ANDROID\_PLATFORM) && !defined(JT\_CHIP\_hisilicon3712) && defined(JT\_CLOSECAPTION\_ENABLED)

if ( ( HDIGetInitFlag() & ( HDI\_INIT\_HW | HDI\_INIT\_APP ) )

!= 0 )

{

status\_code = JTHDICC\_Init();

JT\_ASSERT(status\_code == JT\_SUCCESS);

}

#endif

//HDI\_InitWDG();//watchdog

//MediaPlayerInit(NULL);

JT\_INFO(("%s", get\_hdi\_revision()));

g\_is\_hdi\_init = JT\_TRUE;

}

#if defined(JT\_SUPPORT\_FUNCTION\_ADP\_OSD)

status\_code = JTHDIADP\_OSD\_Init();

if (JT\_SUCCESS != status\_code)

{

JT\_ERROR(("call JTHDIADP\_OSD\_Init failed !\n"));

}

#endif

#if defined(JT\_SUPPORT\_VFD)

JT\_INFO(("JTHDIVFD\_PT6302\_Init\_Entry\n"));

JTHDIVFD\_PT6302\_Init();

#endif

#if defined(JT\_CHIP\_hisilicon3796mv100)

if(JT\_SUCCESS != JTOS\_TaskCreate(&g\_rf\_task, "RF\_Check\_Key\_Task", (JTOS\_TaskPfn\_t)CheckKeyTask, NULL, NULL, JTOS\_TASK\_DEFAULT\_STACK\_SIZE, JTOS\_TASK\_PRIORITY\_LOW))

{

JT\_ERROR(("RF Check Key Task Crtate Failed!\n"));

}

#endif

#if defined(JT\_SUPPORT\_CEC\_FUNCTION)

JTHDIHDMI\_CEC\_Enable();

#endif

return status\_code;

}

JT\_StatusCode\_t HDI\_InitWDG(void)

{

JT\_StatusCode\_t status\_code = JT\_SUCCESS;

HI\_S32 ret =0;

HI\_U32 retvalue =0;

#if defined (JT\_CHIP\_hisilicon3716mv310)

ret = HI\_UNF\_WDG\_Init();

#else

ret = HI\_UNF\_WDG\_Open();

#endif

if(ret != 0)

{

JT\_ERROR(("\n HI\_UNF\_WDG\_Open %d\n",ret));

return JT\_FAILED;

}

#if defined(JT\_CHIP\_hisilicon3716cv200) || defined(JT\_CHIP\_hisilicon3712) || defined(JT\_CHIP\_hisilicon3796cv100) || defined(JT\_CHIP\_hisilicon3796mv100) || defined(JT\_CHIP\_hisilicon3798cv200) || defined(JT\_CHIP\_hisilicon3798mv200) || defined(JT\_CHIP\_hisilicon3716mv410) || defined(JT\_CHIP\_hisilicon3716mv420)

ret = HI\_UNF\_WDG\_Enable(0);

if(ret != 0)

{

JT\_ERROR(("\n HI\_UNF\_WDG\_Open %d\n",ret));

return JT\_FAILED;

}

HI\_UNF\_WDG\_SetTimeout(0,60000);//ms

ret = HI\_UNF\_WDG\_GetTimeout(0,&retvalue);

#else

ret = HI\_UNF\_WDG\_Enable();

if(ret != 0)

{

JT\_ERROR(("\n HI\_UNF\_WDG\_Open %d\n",ret));

return JT\_FAILED;

}

HI\_UNF\_WDG\_SetTimeout(60000);//ms

ret = HI\_UNF\_WDG\_GetTimeout(&retvalue);

#endif

//JT\_ERROR(("\n HI\_UNF\_WDG\_GetTimeout %d %d\n",ret,retvalue));

status\_code = JTOS\_TaskCreate(&g\_task\_id,"WDG\_Thread",(JTOS\_TaskPfn\_t)WDG\_Process,NULL,NULL,16 \* 1024,JTOS\_TASK\_PRIORITY\_HIGH);

if (status\_code != JT\_SUCCESS)

{

JT\_ERROR(("ERROR! Can not creat the thread of IR.\n"));

return JT\_FAILED;

}

return status\_code;

}

#include "gt\_debug.h"

#include "hdi\_init.h"

#include "hdi\_av.h"

#include "hdi\_flash.h"

#ifdef JT\_SUPPORT\_TTX\_FUNTION\_BY\_HISILICON

#include "hdi\_ttx.h"

#endif

#include "api\_init.h"

#include "api\_si.h"

#include "api\_epg.h"

#include "api\_dm.h"

#include "util\_timer.h"

#include "api\_timer\_time.h"

#include "api\_net.h"

#include "api\_flash.h"

#include "api\_search.h"

#include "api\_play.h"

#include "api\_ttx.h"

#include "api\_subt.h"

#include "api\_pip.h"

#ifdef JT\_HDCP\_ENABLED

#include "api\_hdcp.h"

#endif

#ifdef JT\_CB\_SUBTITLE\_ENABLED

#include "api\_stsubt.h"

#endif

#include "hdi\_cc.h"

#define DM\_SIGN\_STRING ("JT EMDB V0.11")

#ifdef JT\_CB\_SUBTITLE\_ENABLED

ST\_DeviceName\_t SUBT\_DeviceName[]={"SUBT0","SUBT1"};

#endif

#ifdef JT\_SUPPORT\_FAST\_SACAN\_FUNTION

#include "api\_fnt.h"

#include "api\_fst.h"

extern FST\_CHID\_t \*g\_fst\_chid\_p;

extern JTAPIFST\_FinishCallback g\_fst\_update\_finish\_callback;

static JTOS\_TaskId\_t g\_fst\_update\_task\_id = JT\_INVALID\_ID;

static JTOS\_TickId\_t search\_tick\_fnt\_version = JT\_INVALID\_ID;

static void search\_timer\_fnt\_version\_handler(JTOS\_TickId\_t tick\_id, void\* ptr);

JT\_BOOL JTAPIFNT\_CheckFSTVersion(JT\_U8 dmx\_id, JT\_U8 \*pTabFnt, int \*size);

#endif

static JT\_BOOL g\_is\_api\_init = JT\_FALSE;

#ifndef JT\_EMDB\_V2

JT\_BOOL g\_b8NeedToDefaultDM = JT\_FALSE;

#endif

#ifdef JT\_CAS\_ENABLED

static APICAS\_ServiceRegister\_t stapicas\_service[APICAS\_MAX];

void JTAPPView\_Init(void \*pWindow)

{

JT\_StatusCode\_t status\_code = JT\_FAILED;

APP\_MSGNoticeInit();

status\_code = JTOS\_QueueCreate(&g\_app\_Queue, (const JT\_S8\*)"APPQueue", 32); //max\_msg\_count: 32, 64

JT\_ASSERT(g\_app\_Queue != JT\_INVALID\_ID);

status\_code = JTOS\_TaskCreate(&g\_app\_task, (const JT\_S8\*)"APPTask", (JTOS\_TaskPfn\_t)APP\_MainTask, pWindow, NULL, JTOS\_TASK\_DEFAULT\_STACK\_SIZE, JTOS\_TASK\_PRIORITY\_NORMAL);

JT\_ASSERT(g\_app\_task != JT\_INVALID\_ID);

status\_code = JTAPITIMER\_LoadFromDM(timer\_get\_fun);

JT\_ASSERT(status\_code == JT\_SUCCESS);

JTAPITIME\_RegisterCallback(GetBookNotifyFunc());

#if defined(JT\_DYNAMIC\_PMT)

JTAPIPLAY\_RegisterDynamicPMTCallback(DynamicPMTCallback);

#endif

// JTHDIDMD\_RegisterCallback(0, Tuner0\_NoSignal\_EvtCallback);

RegisterPlayerFinishCallback();

JTHDIUSB\_RegisterCallback(Usb\_Callback);

#if defined(JT\_CAS\_CRYPTOGUARD)

RegisterCryptoGuardNotify();

CryptoGuard\_Cas\_Manage\_Init();

CryptoGuard\_Cas\_Public\_Init();

#endif

#if defined(JT\_CAS\_CTI)

RegisterCTINotify();

CTI\_LoadEmail();

#endif

#if defined(JT\_CAS\_VENUSCAS)

RegisterVenuscasNotify();

Venuscas\_LoadEmail();

#endif

#ifdef JT\_CAS\_GRIFFIN

RegisterGriffinNotify();

#endif

#ifdef JT\_CAS\_TOPREAL

RegisterToprealNotify();

#endif

#if defined(JT\_CAS\_CONAXCAS)

RegisterConaxcasNotify();

Conaxcas\_Cas\_Manage\_Init();

Conaxcas\_Cas\_Public\_Init();

#endif

#if defined(JT\_CAS\_SUMAVISION)

JT\_StatusCode\_t status;

status = Sumavision\_Cas\_Dialog\_Init();

if(status != JT\_SUCCESS)

{

JT\_ERROR(("INIT Sumavision Error! >>>>>>>>"));

return;

}

RegisterSumavisionNotify();

#endif

#if defined(SUPPORT\_SET\_VEDIO\_COLOR)

JTAPPAVCOLOR\_Init();

#endif

}

JT\_StatusCode\_t JTGUI\_Init()

{

JT\_StatusCode\_t enStatus = JT\_SUCCESS;

HDIIR\_TYPE\_t enActiveRemote = HDIIR\_TYPE\_MAX;

JTAPIDM\_SYSCFG\_OSDLANGUAGE\_t enOsdLanguage;

#if defined(JT\_OS\_LINUX)

if(JTHDIVFS\_FileExist("/home/db/FILE\_NOSEARCH\_AFTER\_FAKE\_STANDBY\_FLAG"))

{

g\_b8IsWakeUp\_Status = JT\_TRUE; JTHDIVFS\_FileDelete("/home/db/FILE\_NOSEARCH\_AFTER\_FAKE\_STANDBY\_FLAG");

}

#endif

#if defined(JT\_CHIP\_gx3113b)

/\*gx3113b/gx3201 chipset\*/

BLSM\_WGUI\_RELLENMASK = 0x0100;

bGXFontTheme = JT\_TRUE;

HIDE\_COLOR = 0x1164;

#elif defined(JT\_CHIP\_gx3201)|| defined(JT\_CHIP\_gx6601)||defined(JT\_CHIP\_gx3231)

BLSM\_WGUI\_RELLENMASK = 0xFF;

bGXFontTheme = JT\_TRUE;

HIDE\_COLOR = 0x1164;

#else

/\*Hisilicon 3716\*/

BLSM\_WGUI\_RELLENMASK = 0x8000;

bGXFontTheme = JT\_FALSE;

HIDE\_COLOR = 0xB208;

#endif

JTGUIOSD\_Init();

#if !defined(JT\_SUPPORT\_DISPLAY\_GUI\_BY\_HIGO)

JTGUITEXT\_InitFontLib(JT\_TRUE);

#endif

#ifndef JT\_SUPPORT\_DISPLAY\_GUI\_BY\_HIGO

JTGUIPNG\_Init();

#endif

KEY\_Init();

#if defined(JT\_REMOTE\_TYPE\_RCU\_A\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_A\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_B\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_B\_04)

enActiveRemote = HDIIR\_TYPE\_RCU\_B\_04;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_B\_06)

enActiveRemote = HDIIR\_TYPE\_RCU\_B\_06;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_C\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_C\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_D\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_D\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_E\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_E\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_K\_01)

enActiveRemote = HDIIR\_TYPE\_RCU\_K\_01;

#elif defined(JT\_REMOTE\_TYPE\_RCU\_WIDETV)

#if defined(JT\_CUSTOMER\_WIDETV)

enActiveRemote = = HDIIR\_TYPE\_RCU\_OSN;

#endif

#else

#error YOU MUST DEFINE Remote Type!!!!!!!!!!

#endif

#if defined(JT\_SUPPORT\_COMBO\_REMOTE\_RCU\_A\_01\_AND\_RCU\_B\_04)

enStatus = JTHDIIR\_RemoteTypeGet(&enActiveRemote);

if (enStatus != JT\_SUCCESS)

{

enActiveRemote = HDIIR\_TYPE\_RCU\_B\_04;

}

#endif

KEY\_RcuMapRegister(enActiveRemote);

#if defined(JT\_CUSTOMER\_WIDETV)&&defined(JT\_REMOTE\_TYPE\_RCU\_WIDETV)

enActiveRemote = HDIIR\_TYPE\_RCU\_OSN;

KEY\_RcuMapRegister(enActiveRemote);

#endif

WND\_Init();

JTAPP\_InitAV();

APPInit();

SoftwareVersion\_Init();

JTAPIDM\_SyscfgGetOSDLang(&enOsdLanguage);

JTGUI\_LANGUAGE\_MAP(enOsdLanguage);

#ifdef JT\_CB\_SUBTITLE\_ENABLED

subtitleDisplay\_Init();

JTAPISTSUBT\_RegisterPMTCallBack();

#endif

void \*pWindow = NULL;

JTAPPView\_Init(&pWindow);

MAINWND\_Setup();

}

#include "ui\_com.h"

#include "satellitemanage.h"

#include "satelliteedit.h"

#include "api\_dm\_sat.h"

#include "api\_dm\_tp.h"

#include "api\_dm.h"

#include "mainwnd.h"

#if defined(JT\_CUSTOMER\_MALDIVES)

#define POPUPBOX\_STYLE\_FOR\_MEDIANET

#endif

/\*==============================MACRO DEFINE===========================\*/

#define SATELLITEMANAGE\_WND\_NAME "Satellite Manage"

#define SATELLITEMANAGE\_REFRESH\_WND\_X 186

#define SATELLITEMANAGE\_REFRESH\_WND\_Y 100

#define SATELLITEMANAGE\_REFRESH\_WND\_W (WINDOW\_MAX\_WIDTH - SATELLITEMANAGE\_REFRESH\_WND\_X)

#define SATELLITEMANAGE\_REFRESH\_WND\_H (WINDOW\_MAX\_HEIGHT - 2\*SATELLITEMANAGE\_REFRESH\_WND\_Y)

#define SATELLITEMANAGE\_WND\_X 0

#define SATELLITEMANAGE\_WND\_Y 0

#define SATELLITEMANAGE\_WND\_W WINDOW\_MAX\_WIDTH

#define SATELLITEMANAGE\_WND\_H WINDOW\_MAX\_HEIGHT

//list box

#define SATELLITEMANAGE\_LIST\_W (SATELLITEMANAGE\_WND\_W \* 3 / 5)

#define SATELLITEMANAGE\_LIST\_H (SATELLITEMANAGE\_WND\_H \* 3 / 5)

#define SATELLITEMANAGE\_LIST\_X (SATELLITEMANAGE\_WND\_X + ((SATELLITEMANAGE\_WND\_W - SATELLITEMANAGE\_LIST\_W) >> 1))

#define SATELLITEMANAGE\_LIST\_Y (SATELLITEMANAGE\_WND\_Y + ((SATELLITEMANAGE\_WND\_H - SATELLITEMANAGE\_LIST\_H) >> 1) + 20)

#define SATELLITEMANAGE\_LIST\_ROW 5 //minum is 5

#define SATELLITEMANAGE\_ITEM\_H (SATELLITEMANAGE\_LIST\_H / (SATELLITEMANAGE\_LIST\_ROW + 2))

#define SATELLITEMANAGE\_ITEM\_X(index) SATELLITEMANAGE\_LIST\_X

#define SATELLITEMANAGE\_ITEM\_Y(index) (SATELLITEMANAGE\_LIST\_Y + (index < (SATELLITEMANAGE\_LIST\_ROW - 1) / 2 ? index : index + 2) \* SATELLITEMANAGE\_ITEM\_H)

#define SATELLITEMANAGE\_ITEM\_ID\_W (100 \* WINDOW\_MAX\_WIDTH / 1280)

#define SATELLITEMANAGE\_ITEM\_ID\_H (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define SATELLITEMANAGE\_ITEM\_ID\_X (212 \* WINDOW\_MAX\_WIDTH / 1280)

#define SATELLITEMANAGE\_ITEM\_MID\_Y (330 \* WINDOW\_MAX\_HEIGHT / 720)

#define SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_H (50 \* WINDOW\_MAX\_HEIGHT / 720)

#define SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_X (305 \* WINDOW\_MAX\_WIDTH / 1280)

#define SATELLITEMANAGE\_ITEM\_POSITION\_H (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define SATELLITEMANAGE\_ITEM\_POSITION\_X (780 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB0\_LABLE\_X (960 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB0\_LABLE\_Y (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define LNB0\_LABLE\_W (150 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB0\_LABLE\_H (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define LNB0\_LABLE\_ICON\_X (1000 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB0\_LABLE\_ICON\_Y(index) SATELLITEMANAGE\_ITEM\_Y(index)

#define LNB1\_LABLE\_X (1080 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB1\_LABLE\_Y (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define LNB1\_LABLE\_W LNB0\_LABLE\_W

#define LNB1\_LABLE\_H LNB0\_LABLE\_H

#define LNB1\_LABLE\_ICON\_X (1120 \* WINDOW\_MAX\_WIDTH / 1280)

#define LNB1\_LABLE\_ICON\_Y(index) SATELLITEMANAGE\_ITEM\_Y(index)

#define SATELLITEMANAGE\_BUTTON\_X(index) ((120 + (index) \* 200) \* WINDOW\_MAX\_WIDTH / 1280)

#define SATELLITEMANAGE\_BUTTON\_Y(index) (625 \* WINDOW\_MAX\_HEIGHT / 720)

#define SATELLITEMANAGE\_BUTTON\_INFO\_X(index) ((165 + (index) \* 200) \* WINDOW\_MAX\_WIDTH / 1280)

#define SATELLITEMANAGE\_BUTTON\_INFO\_Y(index) (635 \* WINDOW\_MAX\_HEIGHT / 720)

typedef enum

{

JTGUI\_LATITUDE\_MIN = 0,

JTGUI\_LATITUDE\_EAST = JTGUI\_LATITUDE\_MIN,

JTGUI\_LATITUDE\_WEST,

JTGUI\_LATITUDE\_MAX

}JTGUI\_LATITUDE\_t;

typedef enum

{

LNB\_DISABLE,

LNB0\_ON,

LNB1\_ON,

}JTGUI\_LNBStatus\_t;

typedef struct

{

JT\_U16\* pSatIdArray;

HSCROLL hScroll;

JT\_S32 S32ListCount; //SatCnt

JT\_U8 lnbCnt;

JT\_BOOL bChange;

JT\_BOOL bLoadData; //when S32ListCount changes, this must be true to reload

JT\_BOOL bInit;

}JTGUI\_SATELLITEMANAGE\_DATA\_t;

/\*=========================global variable==============================\*/

static HWND g\_hSatelliteManage = NULL;

static JT\_U8 \*g\_pu8Info[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_edit),

JTGUI\_FILLTITLE(msg\_add),

JTGUI\_FILLTITLE(msg\_delete\_full),

JTGUI\_FILLTITLE(msg\_select),

};

static JT\_StatusCode\_t JTGUI\_DrawMiddleBg(JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData)

{

//JTGUIBMP\_DisplayPartOfBitmap(SATELLITEMANAGE\_REFRESH\_WND\_X, SATELLITEMANAGE\_REFRESH\_WND\_Y,SATELLITEMANAGE\_REFRESH\_WND\_W,\

// SATELLITEMANAGE\_REFRESH\_WND\_H,SATELLITEMANAGE\_REFRESH\_WND\_X, SATELLITEMANAGE\_REFRESH\_WND\_Y,&ui\_BackgroudIfor,JT\_TRUE); JTGUI\_Draw\_Part\_of\_PNG(PATH\_BACKGROUND\_LEFT\_PNG,SATELLITEMANAGE\_REFRESH\_WND\_X, SATELLITEMANAGE\_REFRESH\_WND\_Y,SATELLITEMANAGE\_REFRESH\_WND\_W,SATELLITEMANAGE\_REFRESH\_WND\_H,SATELLITEMANAGE\_REFRESH\_WND\_X - PATH\_CUPID\_MENU3\_PNG\_X, SATELLITEMANAGE\_REFRESH\_WND\_Y);

//JTGUIBMP\_DisplayBitmap(186,255,ui\_SelectItemIfor.bWidth, ui\_SelectItemIfor.bHeight, 0, 0, &ui\_SelectItemIfor, JT\_TRUE);

JTGUIPNG\_DrawPngEx(PATH\_CUPID\_MENU3\_PNG, PATH\_CUPID\_MENU3\_PNG\_X, PATH\_CUPID\_MENU3\_PNG\_X, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

}

static JT\_StatusCode\_t JTGUI\_DrawWndBg(JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData)

{

JT\_S32 i;

JT\_S32 s32Num;

JTGUI\_TEXT\_t stText;

JTGUI\_BACKGROUND\_t stBg;

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stBg.stBgType = JTGUI\_BGTYPE\_NONE;

//JTGUIBMP\_DisplayBitmap(SATELLITEMANAGE\_WND\_X, SATELLITEMANAGE\_WND\_Y, ui\_BackgroudIfor.bWidth, ui\_BackgroudIfor.bHeight,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET, &ui\_BackgroudIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(0, 0,ui\_SubMenuIfor.bWidth, ui\_SubMenuIfor.bHeight, 0, 0, &ui\_SubMenuIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(186,255,ui\_SelectItemIfor.bWidth, ui\_SelectItemIfor.bHeight, 0, 0, &ui\_SelectItemIfor, JT\_TRUE);

JTGUI\_DrawWholeBG\_And\_Selector();

JTGUIPNG\_DrawPngEx(PATH\_INSTALLATION\_BIG\_ICON\_PNG, JTGUI\_SUBMENU\_BOX\_ICON\_X,JTGUI\_SUBMENU\_BOX\_ICON\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

stText.pu8Text = msg\_satellite\_manage[stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(JTGUI\_SUBWND\_TITLE\_X, JTGUI\_SUBWND\_TITLE\_Y, JTGUI\_SUBWND\_TITLE\_W, JTGUI\_SUBWND\_TITLE\_H, JTGUI\_COLOR\_WHITE, JTGUI\_FONTSIZE\_BIG, EN\_JTGUI\_TEXTALIGN\_LEFT, &stText, &stBg);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_RED\_PNG,SATELLITEMANAGE\_BUTTON\_X(0), SATELLITEMANAGE\_BUTTON\_Y(0), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_GREEN\_PNG,SATELLITEMANAGE\_BUTTON\_X(1), SATELLITEMANAGE\_BUTTON\_Y(1), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_YELLOW\_PNG,SATELLITEMANAGE\_BUTTON\_X(2), SATELLITEMANAGE\_BUTTON\_Y(2), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_BLUE\_PNG,SATELLITEMANAGE\_BUTTON\_X(3), SATELLITEMANAGE\_BUTTON\_Y(3), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_BACK\_BUTTON\_PNG, JTGUI\_SUBWND\_ICON\_BACK\_X,JTGUI\_SUBWND\_ICON\_BACK\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_MEDIUM);

for(i = 0; i < 4; i++)

{

JTGUITEXT\_DrawUnicodeTxt(SATELLITEMANAGE\_BUTTON\_INFO\_X(i), SATELLITEMANAGE\_BUTTON\_INFO\_Y(i), MSG\_LABLE\_W\*3, MSG\_LABLE\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET,\

g\_pu8Info[i][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

}

JTGUITEXT\_DrawUnicodeTxt(JTGUI\_SUBWND\_INFO\_BACK\_X, JTGUI\_SUBWND\_INFO\_BACK\_Y, MSG\_LABLE\_W, MSG\_LABLE\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, msg\_back[stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

//draw lnb0 label

switch (pstData->lnbCnt)

{

case 2:

{

stText.pu8Text = msg\_lnb1[stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(LNB1\_LABLE\_X, LNB1\_LABLE\_Y, LNB0\_LABLE\_W, LNB0\_LABLE\_H, JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_MEDIUM+5, EN\_JTGUI\_TEXTALIGN\_MIDDLE, &stText, &stBg);

}

case 1:

{

stText.pu8Text = msg\_lnb0[stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(LNB0\_LABLE\_X, LNB0\_LABLE\_Y, LNB0\_LABLE\_W, LNB0\_LABLE\_H, JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_MEDIUM+5, EN\_JTGUI\_TEXTALIGN\_MIDDLE, &stText, &stBg);

break;

}

default:

break;

}

return JT\_SUCCESS;

}

static JT\_StatusCode\_t satManage\_loadData(JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData)

{

JT\_U32 u32SatCnt = 0;

JT\_U32 i = 0;

JT\_U32 j = 0;

JT\_U16\* pSatIdTempArray = NULL;

pSatIdTempArray = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* MAX\_SAT\_NUMBER);

if (NULL == pSatIdTempArray)

{

JT\_ERROR(("#error: %d, %s: JTOS\_MemMalloc Failed!!!\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return JT\_FAILED;

}

if (NULL == pstData->pSatIdArray)

{

pstData->pSatIdArray = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* MAX\_SAT\_NUMBER);

if(NULL == pstData->pSatIdArray)

{

JT\_ERROR(("#error: %d, %s: JTOS\_MemMalloc Failed!!!\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return JT\_FAILED;

}

}

memset(pSatIdTempArray, 0, sizeof(JT\_U16) \* MAX\_SAT\_NUMBER);

memset(pstData->pSatIdArray, -1, sizeof(JT\_U16) \* MAX\_SAT\_NUMBER);

j = 0;

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_MAX, pSatIdTempArray, &u32SatCnt);

for (i = 0; i < u32SatCnt; i++)

{

if (pSatIdTempArray[i] < SAT\_ID\_ISDB\_T)

{

pstData->pSatIdArray[j] = pSatIdTempArray[i];

j++;

}

}

if (pstData->lnbCnt > 1)

{

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_1, pSatIdTempArray, &u32SatCnt);

for (; i < u32SatCnt; i++)

{

if (pSatIdTempArray[i] < SAT\_ID\_ISDB\_T)

{

pstData->pSatIdArray[j] = pSatIdTempArray[i];

j++;

}

}

}

pstData->S32ListCount = j;

SCROLL\_SetMaxItem(pstData->hScroll, pstData->S32ListCount);

JTOS\_MemFree(pSatIdTempArray);

pSatIdTempArray = NULL;

return JT\_SUCCESS;

}

static void InitLnb(JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData)

{

JT\_S32 i = 0;

JTHDIDMD\_TunerCapability\_t stTuner = {0};

JTHDIDMD\_GetTunerCapability(&stTuner);

for (i = 0; i < stTuner.tuner\_num; i++)

{

if (stTuner.tuner\_type[i] == JT\_DMD\_DELIVERY\_S2 || stTuner.tuner\_type[i] == JT\_DMD\_DELIVERY\_SATELLITE)

{

pstData->lnbCnt++;

}

}

}

static void JTGUI\_Draw\_LNB(JT\_U8 lnb, JT\_S32 y)

{

switch (lnb)

{

case JT\_DMD\_PORT\_0:

{

JTGUIPNG\_DrawPngEx(PATH\_ICON\_VOICEBAR3\_PNG, LNB0\_LABLE\_ICON\_X, y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

break;

}

case JT\_DMD\_PORT\_1:

{

JTGUIPNG\_DrawPngEx(PATH\_ICON\_VOICEBAR3\_PNG, LNB1\_LABLE\_ICON\_X, y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

break;

}

default:

{

break;

}

}

}

static JT\_BOOL JTGUI\_LIST\_DrawItem(HWND hWnd, JT\_S32 s32RowIndex, JT\_S32 s32ItemIndex, JT\_BOOL bSelect)

{

JTGUI\_SATELLITEMANAGE\_DATA\_t\* pstData = NULL;

JT\_U32 u32IndexInList = 0;

JT\_S32 s32Num;

JTAPIDM\_SatInfo\_t stSat = {0};

JT\_StatusCode\_t statusCode;

JT\_U8 tmpBuf[10];

JT\_S32 i = 0;

JT\_U16 otherIndex = 0;

JT\_S32 y = 0;

if (bSelect == JT\_FALSE)

{

return JT\_TRUE;

}

pstData = (JTGUI\_SATELLITEMANAGE\_DATA\_t \*)WND\_GetData(hWnd);

//selected

statusCode = JTAPIDM\_SatGetById(pstData->pSatIdArray[s32ItemIndex], &stSat);

if (statusCode == JT\_SUCCESS)

{

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_BIG);

snprintf(tmpBuf, 10, "%d", s32ItemIndex + 1);

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_ID\_X, SATELLITEMANAGE\_ITEM\_MID\_Y, SATELLITEMANAGE\_ITEM\_ID\_W, \

SATELLITEMANAGE\_ITEM\_ID\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, tmpBuf, 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 0, &s32Num);

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_X, SATELLITEMANAGE\_ITEM\_MID\_Y, 438, \

SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, stSat.name, 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 0, &s32Num);

snprintf(tmpBuf, 10, "%03d.%d%s", abs(stSat.position) / 10, abs(stSat.position) % 10, (stSat.position > 0) ? "E" : "W");

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_POSITION\_X, SATELLITEMANAGE\_ITEM\_MID\_Y, 200, \

SATELLITEMANAGE\_ITEM\_POSITION\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, tmpBuf, 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 0, &s32Num);

JTGUI\_Draw\_LNB(stSat.lnb\_port, SATELLITEMANAGE\_ITEM\_MID\_Y);

}

//others

for (i = 0 ; i < SATELLITEMANAGE\_LIST\_ROW; i++)

{

if (i == (SATELLITEMANAGE\_LIST\_ROW - 1) / 2)

{

//selected one

continue;

}

else

{

if (pstData->S32ListCount >= ((SATELLITEMANAGE\_LIST\_ROW + 1) / 2))

{

otherIndex = (s32ItemIndex + i + pstData->S32ListCount - (SATELLITEMANAGE\_LIST\_ROW - 1) / 2) % pstData->S32ListCount;

}

else

{

otherIndex = s32ItemIndex + i - (SATELLITEMANAGE\_LIST\_ROW - 1) / 2;

if (otherIndex < 0 || otherIndex >= pstData->S32ListCount)

{

continue;

}

}

}

statusCode = JTAPIDM\_SatGetById(pstData->pSatIdArray[otherIndex], &stSat);

if (statusCode == JT\_SUCCESS)

{

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_MEDIUM);

snprintf(tmpBuf, 10, "%d", otherIndex + 1);

y = SATELLITEMANAGE\_ITEM\_Y(i);

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_ID\_X, y, SATELLITEMANAGE\_ITEM\_ID\_W, \

SATELLITEMANAGE\_ITEM\_ID\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, tmpBuf, 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_X, y, 438, \

SATELLITEMANAGE\_ITEM\_SELECT\_NAME\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, stSat.name, 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

snprintf(tmpBuf, 10, "%03d.%d%s", abs(stSat.position) / 10, abs(stSat.position) % 10, (stSat.position > 0) ? "E" : "W");

JTGUITEXT\_DrawAsciiTxt(SATELLITEMANAGE\_ITEM\_POSITION\_X, y, 200, \

SATELLITEMANAGE\_ITEM\_POSITION\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, tmpBuf, 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

JTGUI\_Draw\_LNB(stSat.lnb\_port, y);

}

}

}

static JT\_StatusCode\_t JTGUI\_LIST\_DrawContent(JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData)

{

POINT stPoint;

stPoint.X = SATELLITEMANAGE\_LIST\_X;

stPoint.Y = SATELLITEMANAGE\_LIST\_Y;

pstData->hScroll = SCROLL\_Init(g\_hSatelliteManage, stPoint, SATELLITEMANAGE\_LIST\_W, SATELLITEMANAGE\_LIST\_H, 1, pstData->S32ListCount, 0, JT\_FALSE, JTGUI\_LIST\_DrawItem);

if (pstData->hScroll == NULL)

{

return JT\_ERROR\_INVALID\_HANDLE;

}

return JT\_SUCCESS;

}

static JT\_S32 JTGUI\_SATELLITEMANAGE\_Pro(HWND hWnd, JT\_S16 sMsg, JT\_S16 sKey, JT\_S32 iParam1, JT\_S32 iParam2)

{

JT\_S32 s32CurIndex, temp, index;

JTGUI\_BACKGROUND\_t stBg;

BITMAPTRUECOLOR \*pPopupBg;

JTGUI\_SATELLITEMANAGE\_DATA\_t \*pstData = NULL;

pstData = WND\_GetData(g\_hSatelliteManage);

if (pstData->S32ListCount != 0)

{

SCROLL\_GetIndex(pstData->hScroll, &temp, &s32CurIndex);

}

else

{

temp = 0;

s32CurIndex = 0;

}

switch(sMsg)

{

case WM\_CREATE:

{

memset(pstData, 0, sizeof(JTGUI\_SATELLITEMANAGE\_DATA\_t));

pstData->bChange = JT\_FALSE;

pstData->hScroll = NULL;

pstData->pSatIdArray = NULL;

pstData->S32ListCount = 0;

pstData->lnbCnt = 0;

pstData->bLoadData = JT\_TRUE;

pstData->bInit = JT\_FALSE;

InitLnb(pstData);

JTGUI\_LIST\_DrawContent(pstData);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case WM\_SHOW:

{

if (pstData->bLoadData == JT\_TRUE)

{

satManage\_loadData(pstData);

pstData->bLoadData = JT\_FALSE;

}

JTGUI\_DrawWndBg(pstData);

SCROLL\_StartNavi(pstData->hScroll);

if (pstData->bInit == JT\_FALSE)

{

JTHDIOSD\_RefreshScreen(SATELLITEMANAGE\_WND\_X, SATELLITEMANAGE\_WND\_Y, SATELLITEMANAGE\_WND\_W, SATELLITEMANAGE\_WND\_H);

pstData->bInit = JT\_TRUE;

}

else

{ JTHDIOSD\_RefreshScreen(SATELLITEMANAGE\_REFRESH\_WND\_X, SATELLITEMANAGE\_REFRESH\_WND\_Y, SATELLITEMANAGE\_REFRESH\_WND\_W, SATELLITEMANAGE\_REFRESH\_WND\_H);

}

}

break;

case WM\_KEYDOWN:

{

switch (sKey)

{

case KEY\_BACK:

{

if (pstData->bChange == JT\_TRUE)

{

JTGUI\_BACKGROUND\_t stBg;

JTGUI\_BTNFOCUS\_t ret;

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

//Background is already hard-define in popup Dialg

ret = JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_YESNO, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, NULL, POPDLG\_TITLE\_OFFSET\_Y, msg\_save\_changes[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

if (ret == POPUP\_BUTTON\_YES)

{

JTAPIDM\_Save();

}

else

{

JTAPIDM\_Reload();

}

}

WND\_PostMessage(hWnd, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_TERM, WND\_COMMAND\_TERM\_NORMAL\_MODE);

}

break;

case KEY\_OK:

{

JTAPIDM\_SatInfo\_t satInfo = {0};

JT\_S32 index = 0, tmp;

JT\_StatusCode\_t statusCode;

JTAPIDM\_SatGetById(pstData->pSatIdArray[s32CurIndex], &satInfo);

if (pstData->lnbCnt == 1)

{

if (satInfo.lnb\_port == 0)

{

statusCode = JTAPIDM\_SatSetLNBPortByID(pstData->pSatIdArray[s32CurIndex], JT\_DMD\_PORT\_MAX);

}

else

{

statusCode = JTAPIDM\_SatSetLNBPortByID(pstData->pSatIdArray[s32CurIndex], JT\_DMD\_PORT\_0);

}

}

else if (pstData->lnbCnt == 2)

{

JT\_U8 port = 0;

port = (satInfo.lnb\_port + 1) % JT\_DMD\_PORT\_MAX;

if (port > JT\_DMD\_PORT\_1)

{

port = JT\_DMD\_PORT\_MAX;

}

statusCode = JTAPIDM\_SatSetLNBPortByID(pstData->pSatIdArray[s32CurIndex], port);

}

if (statusCode == JT\_SUCCESS)

{

pstData->bChange = JT\_TRUE;

}

WND\_Show(hWnd, SW\_SHOW);

}

break;

case KEY\_UP:

case KEY\_DOWN:

{

if (pstData->S32ListCount != 0)

{

s32CurIndex = (s32CurIndex + (sKey == KEY\_UP ? -1 : 1) + pstData->S32ListCount) % pstData->S32ListCount;

SCROLL\_SetCurFocusItem(pstData->hScroll, s32CurIndex);

}

WND\_Show(hWnd, SW\_SHOW);

//JTGUI\_DrawMiddleBg(pstData);

break;

}

case KEY\_RED:

{

if (pstData->pSatIdArray[s32CurIndex] == (JT\_U16)-1)

{

JTGUI\_BACKGROUND\_t stBg;

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

//Background is already hard-define in popup Dialg

JTGUI\_DrawPopupBox2(g\_hSatelliteManage, JTGUI\_POPUP\_BTN\_OK, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, msg\_information[stGlobalVar.enLanguage], POPDLG\_TITLE\_OFFSET\_Y, msg\_no\_satellite\_selected[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

}

else

{

JT\_BOOL bChange;

JTGUI\_SatelliteEdit\_Setup(hWnd, pstData->pSatIdArray[s32CurIndex], &bChange, JTGUI\_SAT\_EDIT);

pstData->bInit = JT\_FALSE;

}

break;

}

case KEY\_GREEN:

{

if (pstData->S32ListCount >= 252)

{

JTGUI\_BACKGROUND\_t stBg = {.stBgType = JTGUI\_BGTYPE\_PICTURE};

JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_OK, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, NULL, POPDLG\_TITLE\_OFFSET\_Y, msg\_more\_than\_maxinum[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

break;

}

JTGUI\_SatelliteEdit\_Setup(hWnd, pstData->pSatIdArray[s32CurIndex], &pstData->bLoadData, JTGUI\_SAT\_ADD);

//reload whole interface

pstData->bInit = JT\_FALSE;

break;

}

#if defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

case KEY\_EPG:

#else

case KEY\_YELLOW:

#endif

{

JTGUI\_BACKGROUND\_t stBg;

JTGUI\_BTNFOCUS\_t ret;

JT\_StatusCode\_t statusCode;

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

//Background is already hard-define in popup Dialg

ret = JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_YESNO, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, NULL, POPDLG\_TITLE\_OFFSET\_Y, msg\_are\_you\_sure\_to\_delete[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

if (ret == POPUP\_BUTTON\_YES)

{

statusCode = JTAPIDM\_TpRemoveBySatId(pstData->pSatIdArray[s32CurIndex]);

statusCode = JTAPIDM\_SatRemoveById(pstData->pSatIdArray[s32CurIndex]);

if (statusCode == JT\_SUCCESS)

{

if (s32CurIndex == (pstData->S32ListCount - 1))

{

SCROLL\_SetCurFocusItem(pstData->hScroll, 0);

}

pstData->bChange = JT\_TRUE;

pstData->bLoadData = JT\_TRUE;

}

}

break;

}

case KEY\_BLUE:

{

JT\_U32 cntSat = 0;

JT\_U32 i = 0;

JTAPIDM\_SatInfo\_t stSat = {0};

JT\_U8 lnb\_port = 0;

JTAPIDM\_SatGetTotalNumber(&cntSat);

JTAPIDM\_SatGetByIndex(0, &stSat);

lnb\_port = stSat.lnb\_port != JT\_DMD\_PORT\_MAX ? JT\_DMD\_PORT\_MAX : JT\_DMD\_PORT\_0;

for (i = 0; i < cntSat; i++)

{

JTAPIDM\_SatSetLNBPortByIndex(i, lnb\_port);

}

pstData->bChange = JT\_TRUE;

pstData->bLoadData = JT\_TRUE;

WND\_Show(hWnd, SW\_SHOW);

break;

}

}

}

break;

case WM\_COMMAND:

{

switch(iParam1)

{

case WND\_COMMAND\_TERM:

{

if(pstData->pSatIdArray != NULL)

{

JTOS\_MemFree(pstData->pSatIdArray);

pstData->pSatIdArray = NULL;

}

if(pstData->hScroll != NULL)

{

SCROLL\_Term(pstData->hScroll);

pstData->hScroll = NULL;

}

WND\_Destroy(hWnd);

g\_hSatelliteManage = NULL;

break;

}

}

}

break;

default:

break;

}

return MAINWND\_DefProc(hWnd, sMsg, sKey, iParam1, iParam2);

}

JT\_StatusCode\_t JTGUI\_SATELLITEMANAGE\_Setup(HWND hParentWnd)

{

RECT tRect;

tRect.X = SATELLITEMANAGE\_WND\_X;

tRect.Y = SATELLITEMANAGE\_WND\_Y;

tRect.W = SATELLITEMANAGE\_WND\_W;

tRect.H = SATELLITEMANAGE\_WND\_H;

// WM\_CREATE

g\_hSatelliteManage = WND\_Create(SATELLITEMANAGE\_WND\_NAME, &tRect, sizeof(JTGUI\_SATELLITEMANAGE\_DATA\_t), JTGUI\_SATELLITEMANAGE\_Pro, hParentWnd);

if(g\_hSatelliteManage == NULL)

{

JT\_ERROR(("Error: SatelliteManage Created in failure ! %d\n", \_\_LINE\_\_));

return JT\_ERROR\_INVALID\_HANDLE;

}

WND\_SetFocus(g\_hSatelliteManage);

return JT\_SUCCESS;

}

HWND JTGUI\_SATELLITEMANAGE\_GetHandle()

{

return g\_hSatelliteManage;

}

#include "ui\_com.h"

#include "hdi\_dmd.h"

#include "hdi\_osd.h"

#include "api\_dm.h"

#include "api\_dm\_sat.h"

#include "motor\_setting.h"

#include "motorlimit\_setup.h"

#include "mainwnd.h"

#if defined(JT\_CUSTOMER\_MALDIVES)

#define POPUPBOX\_STYLE\_FOR\_MEDIANET

#endif

#define ANTENNA\_MENU\_NAME "Antenna Setup"

#define ANTENNA\_MENU\_WND\_X 0

#define ANTENNA\_MENU\_WND\_Y 0

#define ANTENNA\_MENU\_WND\_W WINDOW\_MAX\_WIDTH

#define ANTENNA\_MENU\_WND\_H WINDOW\_MAX\_HEIGHT

#define ANTENNA\_MENU\_LIST\_W (ANTENNA\_MENU\_WND\_W \* 3 / 5)

#define ANTENNA\_MENU\_LIST\_H (ANTENNA\_MENU\_WND\_H \* 3 / 5)

#define ANTENNA\_MENU\_LIST\_X (ANTENNA\_MENU\_WND\_X + ((ANTENNA\_MENU\_WND\_W - ANTENNA\_MENU\_LIST\_W) >> 1))

#define ANTENNA\_MENU\_LIST\_Y (ANTENNA\_MENU\_WND\_Y + ((ANTENNA\_MENU\_WND\_H - ANTENNA\_MENU\_LIST\_H) >> 1) - 20)

#define ANTENNA\_MENU\_LIST\_MAX 7

#define ANTENNA\_MENU\_SELECT\_NAME\_W (610 \* WINDOW\_MAX\_WIDTH / 1280)

#define ANTENNA\_MENU\_SELECT\_NAME\_H (40 \* WINDOW\_MAX\_HEIGHT / 720)

#define ANTENNA\_MENU\_SELECT\_NAME\_X (212 \* WINDOW\_MAX\_WIDTH / 1280)

#define ANTENNA\_MENU\_SELECT\_NAME\_DOWN\_Y (440 \* WINDOW\_MAX\_WIDTH / 1280)

#define ANTENNA\_MENU\_SELECT\_NAME\_MID\_Y (330 \* WINDOW\_MAX\_HEIGHT / 720)

#define EDITE\_BOX\_X 920

#define ITEM\_ARROW\_LEFT\_X (910 \* WINDOW\_MAX\_WIDTH / 1280)

#define ITEM\_ARROW\_RIGHT\_X (940 \* WINDOW\_MAX\_WIDTH / 1280)

#define ITEM\_ARROW\_W (100 \* WINDOW\_MAX\_WIDTH / 1280)

#define ITEM\_ARROW\_H (50 \* WINDOW\_MAX\_HEIGHT / 720)

#define ITEM\_ARROW\_Y (342 \* WINDOW\_MAX\_HEIGHT / 720)

#define ITEM\_PEN\_X 870

#define ITEM\_POP\_BG\_X (990 \* WINDOW\_MAX\_WIDTH / 1280)

#define ITEM\_POP\_BG\_Y (265 \* WINDOW\_MAX\_HEIGHT / 720)

static HWND g\_hANTENNA\_MENU = NULL;

typedef enum

{

JTGUI\_ANTENNA\_MENU\_MIN,

JTGUI\_ANTENNA\_MENU\_LNB\_PORT = JTGUI\_ANTENNA\_MENU\_MIN,

JTGUI\_ANTENNA\_MENU\_SATELLITE,

JTGUI\_ANTENNA\_MENU\_LNBFREQ,

JTGUI\_ANTENNA\_MENU\_DISEQC,

JTGUI\_ANTENNA\_MENU\_22K,

JTGUI\_ANTENNA\_MENU\_MOTORTYPE,

JTGUI\_ANTENNA\_MENU\_LONGTITUDE,

JTGUI\_ANTENNA\_MENU\_LATITUDE,

JTGUI\_ANTENNA\_MENU\_MOTORLIMIT,

JTGUI\_ANTENNA\_MENU\_MAX

}JTGUI\_ANTENNA\_MENU\_t;

typedef struct

{

JT\_U16 \*lnbSatId[2];

JT\_U32 satCnt[2];

HSCROLL hMenu;

JT\_U8 lnbCnt;

JTAPIDM\_SYSCFG\_LNBTYPE\_t lnbType[2];

JT\_U8 CurlnbPort;

JT\_U32 curSatIndex;

JT\_BOOL bChange;

JT\_U32 bRefreshWhole;

JTAPIDM\_SatInfo\_t curSat;

JT\_BOOL bTerm; //when exiting, can not open other window

}JTGUI\_ANTENNA\_MENU\_DATA\_t;

static JT\_U8 \*g\_MenuName[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_lnb\_port),

JTGUI\_FILLTITLE(msg\_satellite),

JTGUI\_FILLTITLE(msg\_lnb\_freq),

JTGUI\_FILLTITLE(msg\_diseqc),

JTGUI\_FILLTITLE(msg\_22k),

JTGUI\_FILLTITLE(msg\_motor\_type),

JTGUI\_FILLTITLE(msg\_local\_longitude),

JTGUI\_FILLTITLE(msg\_local\_latitude),

JTGUI\_FILLTITLE(msg\_motor\_limit),

};

static JT\_U8 \*g\_puLNBName[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_lnb0),

JTGUI\_FILLTITLE(msg\_lnb1),

};

static JT\_U8 \*g\_pu8OnOff[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_off),

JTGUI\_FILLTITLE(msg\_on)

};

static JT\_U8 \*g\_pu8Motor\_Type[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_diseqc1\_2),

JTGUI\_FILLTITLE(msg\_usals)

};

static JT\_U8 \*g\_pu8Diseqc[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_disable),

JTGUI\_FILLTITLE(msg\_port1),

JTGUI\_FILLTITLE(msg\_port2),

JTGUI\_FILLTITLE(msg\_port3),

JTGUI\_FILLTITLE(msg\_port4)

};

static JT\_U8 \*g\_pu8LnbFreq[] =

{

"5150",

"5750",

"5950",

"9750",

"10000",

"10050",

"10450",

"10600",

"10700",

"10750",

"11250",

"11300",

"5150-5750",

"5750-5150",

"9750-10600"

};

static void antennaSetup\_InitLnb(JT\_U8\* lnbCnt)

{

JT\_S32 i = 0;

JTHDIDMD\_TunerCapability\_t stTuner = {0};

\*lnbCnt = 0;

JTHDIDMD\_GetTunerCapability(&stTuner);

for (i = 0; i < stTuner.tuner\_num; i++)

{

if (stTuner.tuner\_type[i] == JT\_DMD\_DELIVERY\_S2 || stTuner.tuner\_type[i] == JT\_DMD\_DELIVERY\_SATELLITE)

{

(\*lnbCnt)++;

}

}

}

static void antennaSetup\_GetCurSatInfo(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

JTAPIDM\_SatGetById((ptData->lnbSatId[ptData->CurlnbPort])[ptData->curSatIndex], &ptData->curSat);

}

static void JTGUI\_Activate22KHzTone(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

JTAPIDM\_SatGetById((ptData->lnbSatId[ptData->CurlnbPort])[ptData->curSatIndex], &ptData->curSat);

JTHDIDMD\_Activate22KHzTone(ptData->curSat.lnb\_port, ptData->curSat.\_22K);

}

static void JTGUI\_DiseqcSwitchPort(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

JTAPIDM\_SatGetById((ptData->lnbSatId[ptData->CurlnbPort])[ptData->curSatIndex], &ptData->curSat);

JTHDIDiseqc\_SwitchPort(ptData->curSat.lnb\_port, ptData->curSat.diseqc, ptData->curSat.\_22K, ptData->curSat.pol, 0);

}

static void antennaSetup\_PopupSaveChange(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

JTGUI\_BACKGROUND\_t stBg;

JTGUI\_BTNFOCUS\_t ret;

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

if (ptData->bChange != JT\_TRUE)

{

return;

}

//Background is already hard-define in popup Dialg

ret = JTGUI\_DrawPopupBox2(g\_hANTENNA\_MENU, JTGUI\_POPUP\_BTN\_YESNO, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, NULL, POPDLG\_TITLE\_OFFSET\_Y, msg\_save\_changes[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

if (ret == POPUP\_BUTTON\_YES)

{ JTAPIDM\_SatModifyById((ptData->lnbSatId[ptData->CurlnbPort])[ptData->curSatIndex], &ptData->curSat);

JTAPIDM\_Save();

}

else

{

JTAPIDM\_Reload();

}

ptData->bChange = JT\_FALSE;

}

static void antennaSetup\_PopupWidget(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData, JT\_S32 curIndex)

{

JTGUI\_TEXT\_t \*ptext = NULL;

JT\_S32 i = 0;

switch(curIndex)

{

case JTGUI\_ANTENNA\_MENU\_LNB\_PORT:

{

JT\_U8 lnbCnt = 0;

JT\_U32 curLnbPort = ptData->CurlnbPort;

for (i = 0; i < 2; i++)

{

if (ptData->satCnt[i] > 0)

{

lnbCnt++;

}

}

ptext = (JTGUI\_TEXT\_t \*)JTOS\_MemMalloc(sizeof(JTGUI\_TEXT\_t) \* lnbCnt);

for (i = 0; i < lnbCnt; i++)

{

ptext[i].stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

ptext[i].pu8Text = g\_puLNBName[i][stGlobalVar.enLanguage];

}

JTGUI\_DrawComboBox(g\_hANTENNA\_MENU, ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ptext, lnbCnt, &curLnbPort);

if (ptData->CurlnbPort != curLnbPort)

{

ptData->bChange = JT\_TRUE;

antennaSetup\_PopupSaveChange(ptData);

ptData->CurlnbPort = curLnbPort;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_SATELLITE:

{

JT\_U8 (\*name)[SAT\_NAME\_MAXLEN];

JTAPIDM\_SatInfo\_t stSat = {0};

JT\_U32 curSatIndex = ptData->curSatIndex;

ptext = (JTGUI\_TEXT\_t \*)JTOS\_MemMalloc(sizeof(JTGUI\_TEXT\_t) \* ptData->satCnt[ptData->CurlnbPort]);

name = (JT\_U8(\*)[SAT\_NAME\_MAXLEN])JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* ptData->satCnt[ptData->CurlnbPort]);

for (i = 0; i < ptData->satCnt[ptData->CurlnbPort]; i++)

{

JTAPIDM\_SatGetById((ptData->lnbSatId[ptData->CurlnbPort])[i], &stSat);

ptext[i].stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

ptext[i].pu8Text = name[i];

strncpy((char \*)name[i], stSat.name, MAX\_SAT\_NUMBER);

}

JTGUI\_DrawComboBox(g\_hANTENNA\_MENU, ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ptext, ptData->satCnt[ptData->CurlnbPort], &curSatIndex);

if (ptData->curSatIndex != curSatIndex)

{

ptData->bChange == JT\_TRUE;

ptData->curSatIndex = curSatIndex;

}

JTGUI\_DiseqcSwitchPort(ptData);

JTGUI\_Activate22KHzTone(ptData);

JTOS\_MemFree(name);

break;

}

case JTGUI\_ANTENNA\_MENU\_LNBFREQ:

{

JT\_U32 lnbFreq = ptData->curSat.lnb\_freq;

ptext = (JTGUI\_TEXT\_t \*)JTOS\_MemMalloc(sizeof(JTGUI\_TEXT\_t) \* JTAPIDM\_LNBType\_MAX);

for (i = 0; i < JTAPIDM\_LNBType\_MAX; i++)

{

ptext[i].stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

ptext[i].pu8Text = g\_pu8LnbFreq[i];

}

JTGUI\_DrawComboBox(g\_hANTENNA\_MENU, ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ptext, JTAPIDM\_LNBType\_MAX, &lnbFreq);

if (lnbFreq != ptData->curSat.lnb\_freq)

{

ptData->curSat.lnb\_freq = lnbFreq;

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_DISEQC:

{

JT\_U32 diseqc = ptData->curSat.diseqc;

ptext = (JTGUI\_TEXT\_t \*)JTOS\_MemMalloc(sizeof(JTGUI\_TEXT\_t) \* JTAPIDM\_DiSEqcType\_MAX);

for (i = 0; i < JTAPIDM\_DiSEqcType\_MAX; i++)

{

ptext[i].stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

ptext[i].pu8Text = g\_pu8Diseqc[i][stGlobalVar.enLanguage];

}

JTGUI\_DrawComboBox(g\_hANTENNA\_MENU, ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ptext, JTAPIDM\_DiSEqcType\_MAX, &diseqc);

if (diseqc != ptData->curSat.diseqc)

{

ptData->curSat.diseqc = diseqc;

ptData->bChange = JT\_TRUE;

JTHDIDiseqc\_SwitchPort(ptData->curSat.lnb\_port, diseqc, ptData->curSat.\_22K, ptData->curSat.pol, 0);

}

break;

}

case JTGUI\_ANTENNA\_MENU\_22K:

{

JT\_U32 \_22k = ptData->curSat.\_22K;

ptext = (JTGUI\_TEXT\_t \*)JTOS\_MemMalloc(sizeof(JTGUI\_TEXT\_t) \* JTAPIDM\_22kType\_MAX);

for (i = 0; i < JTAPIDM\_22kType\_MAX; i++)

{

ptext[i].stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

ptext[i].pu8Text = g\_pu8OnOff[i][stGlobalVar.enLanguage];

}

JTGUI\_DrawComboBox(g\_hANTENNA\_MENU, ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ptext, JTAPIDM\_22kType\_MAX, &\_22k);

if (\_22k != ptData->curSat.\_22K)

{

ptData->curSat.\_22K = \_22k;

JTHDIDMD\_Activate22KHzTone(ptData->curSat.lnb\_port, \_22k);

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORTYPE:

{

JTGUI\_MOTOR\_SETTING\_Setup(g\_hANTENNA\_MENU, ptData->curSat.motor\_type, ptData->CurlnbPort);

ptData->bRefreshWhole = JT\_TRUE;

break;

}

case JTGUI\_ANTENNA\_MENU\_LONGTITUDE:

{

JT\_S16 s16Longitude;

JT\_U8 buf[16] = {0};

JT\_S8 flag = 0;

JTAPIDM\_SyscfgGetLocalLongitude(&s16Longitude);

if (s16Longitude < 0)

{

flag = -1;

}

else

{

flag = 1;

}

snprintf(buf, 16, "%03d", abs(s16Longitude));

JTGUI\_DrawEditer(g\_hANTENNA\_MENU, EDITE\_BOX\_X, JTGUI\_EDITE\_BOX\_Y, JTGUI\_EDITE\_BOX\_W, JTGUI\_EDITE\_BOX\_H,\

msg\_local\_longitude[stGlobalVar.enLanguage], buf);

sscanf(buf, "%d", &s16Longitude);

if (s16Longitude <= 180 && s16Longitude >= 0)

{

s16Longitude = flag \* s16Longitude;

JTAPIDM\_SyscfgSetLocalLongitude(s16Longitude);

ptData->bChange = JT\_TRUE;

}

else

{

WND\_PostMessage(g\_hANTENNA\_MENU, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_REPOST\_MSG, 0);

}

break;

}

case JTGUI\_ANTENNA\_MENU\_LATITUDE:

{

JT\_S16 s16Latitude;

JT\_U8 buf[16] = {0};

JT\_S8 flag = 0;

JTAPIDM\_SyscfgGetLocalLatitude(&s16Latitude);

if (s16Latitude < 0)

{

flag = -1;

}

else

{

flag = 1;

}

snprintf(buf, 16, "%03d", abs(s16Latitude));

JTGUI\_DrawEditer(g\_hANTENNA\_MENU, EDITE\_BOX\_X, JTGUI\_EDITE\_BOX\_Y, JTGUI\_EDITE\_BOX\_W, JTGUI\_EDITE\_BOX\_H,\

msg\_local\_latitude[stGlobalVar.enLanguage], buf);

sscanf(buf, "%d", &s16Latitude);

if (s16Latitude <= 90 && s16Latitude >= 0)

{

s16Latitude = s16Latitude \* flag;

JTAPIDM\_SyscfgSetLocalLatitude(s16Latitude);

ptData->bChange = JT\_TRUE;

}

else

{

WND\_PostMessage(g\_hANTENNA\_MENU, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_REPOST\_MSG, 0);

}

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORLIMIT:

{

JTGUI\_MOTOR\_LIMIT\_Setup(g\_hANTENNA\_MENU, ptData->curSat.lnb\_port);

ptData->bRefreshWhole = JT\_TRUE;

break;

}

}

if (ptext != NULL)

{

JTOS\_MemFree(ptext);

}

}

static void antennaSetup\_SwitchContent(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData, JT\_S32 curIndex, JT\_S32 step)

{

switch (curIndex)

{

case JTGUI\_ANTENNA\_MENU\_LNB\_PORT:

{

JT\_U8 lnbCnt = 0;

JT\_S32 i = 0;

JT\_S32 curLnbPort = 0;

for (i = 0; i < 2; i++)

{

if (ptData->satCnt[i] > 0)

{

lnbCnt++;

}

}

curLnbPort = (ptData->CurlnbPort + step + lnbCnt) % lnbCnt;

if (curLnbPort != ptData->CurlnbPort && ptData->bChange)

{

antennaSetup\_PopupSaveChange(ptData);

ptData->CurlnbPort = curLnbPort;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_SATELLITE:

{

JT\_S32 curSatIndex = 0;

curSatIndex = (ptData->curSatIndex + step + ptData->satCnt[ptData->CurlnbPort]) % ptData->satCnt[ptData->CurlnbPort];

if (curSatIndex != ptData->curSatIndex)

{

antennaSetup\_PopupSaveChange(ptData);

ptData->curSatIndex = curSatIndex;

JTGUI\_DiseqcSwitchPort(ptData);

JTGUI\_Activate22KHzTone(ptData);

}

break;

}

case JTGUI\_ANTENNA\_MENU\_LNBFREQ:

{

JT\_S32 lnbFreq = 0;

lnbFreq = (ptData->curSat.lnb\_freq + JTAPIDM\_LNBType\_MAX + step) % JTAPIDM\_LNBType\_MAX;

if (lnbFreq != ptData->curSat.lnb\_freq)

{

ptData->curSat.lnb\_freq = lnbFreq;

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_DISEQC:

{

JT\_S32 diseqc = 0;

diseqc = (ptData->curSat.diseqc + JTAPIDM\_DiSEqcType\_MAX + step) % JTAPIDM\_DiSEqcType\_MAX;

if (diseqc != ptData->curSat.diseqc)

{

ptData->curSat.diseqc = diseqc;

JTHDIDiseqc\_SwitchPort(ptData->curSat.lnb\_port, diseqc, ptData->curSat.\_22K, ptData->curSat.pol, 0);

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_22K:

{

JT\_S32 \_22k = 0;

\_22k = (ptData->curSat.\_22K + JTAPIDM\_22kType\_MAX + step) % JTAPIDM\_22kType\_MAX;

if (\_22k != ptData->curSat.\_22K)

{

JTHDIDMD\_Activate22KHzTone(ptData->curSat.lnb\_port, \_22k);

ptData->curSat.\_22K = \_22k;

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORTYPE:

{

JT\_S32 motor\_type = 0;

motor\_type = (ptData->curSat.motor\_type + JTAPIDM\_MotorType\_MAX + step) % JTAPIDM\_MotorType\_MAX;

if (motor\_type != ptData->curSat.motor\_type)

{

ptData->curSat.motor\_type = motor\_type;

ptData->bChange = JT\_TRUE;

}

break;

}

case JTGUI\_ANTENNA\_MENU\_LONGTITUDE:

{

JT\_S16 s16Longitude;

JTAPIDM\_SyscfgGetLocalLongitude(&s16Longitude);

s16Longitude = -s16Longitude;

JTAPIDM\_SyscfgSetLocalLongitude(s16Longitude);

ptData->bChange = JT\_TRUE;

break;

}

case JTGUI\_ANTENNA\_MENU\_LATITUDE:

{

JT\_S16 s16Latitude;

JTAPIDM\_SyscfgGetLocalLatitude(&s16Latitude);

s16Latitude = -s16Latitude;

JTAPIDM\_SyscfgSetLocalLatitude(s16Latitude);

ptData->bChange = JT\_TRUE;

break;

}

}

}

static JT\_StatusCode\_t JTGUI\_CheckSatCount(HWND hParentWnd)

{

JT\_U16 \*pLnb[2] = {0};

JT\_U32 lnbSatCnt[2] = {0};

JT\_U8 lnbCnt;

int i;

antennaSetup\_InitLnb(&lnbCnt);

pLnb[JT\_DMD\_PORT\_0] = (JT\_U16\*)JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* sizeof(JT\_U16));

if (pLnb[JT\_DMD\_PORT\_0] == NULL)

{

return JT\_ERROR\_NO\_MEMORY;

}

pLnb[JT\_DMD\_PORT\_1] = (JT\_U16\*)JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* sizeof(JT\_U16));

if (pLnb[JT\_DMD\_PORT\_1] == NULL)

{

return JT\_ERROR\_NO\_MEMORY;

}

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_0, pLnb[JT\_DMD\_PORT\_0], &lnbSatCnt[JT\_DMD\_PORT\_0]);

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_1, pLnb[JT\_DMD\_PORT\_1], &lnbSatCnt[JT\_DMD\_PORT\_1]);

//SAT\_ID\_ISDB\_T SAT\_ID\_UDP SAT\_ID\_DVB\_C SAT\_ID\_DVB\_T

if (pLnb[JT\_DMD\_PORT\_0][0] == SAT\_ID\_ISDB\_T)

{

lnbSatCnt[JT\_DMD\_PORT\_0] -= 4;

}

JTOS\_MemFree(pLnb[JT\_DMD\_PORT\_0]);

JTOS\_MemFree(pLnb[JT\_DMD\_PORT\_1]);

if ((lnbSatCnt[JT\_DMD\_PORT\_0] == 0 && lnbCnt == 1) ||

(lnbSatCnt[JT\_DMD\_PORT\_0] == 0 && lnbSatCnt[JT\_DMD\_PORT\_1] == 0 && lnbCnt == 2))

{

JTGUI\_BACKGROUND\_t stBg;

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

//Background is already hard-define in popup Dialg

JTGUI\_DrawPopupBox2(hParentWnd, JTGUI\_POPUP\_BTN\_OK, POPDLG\_X, POPDLG\_Y, POPDLG\_W, POPDLG\_H, \

&stBg, msg\_information[stGlobalVar.enLanguage], POPDLG\_TITLE\_OFFSET\_Y, msg\_no\_satellite\_selected[stGlobalVar.enLanguage], POPDLG\_CONTENT\_OFFSET\_X, POPDLG\_CONTENT\_OFFSET\_Y);

return JT\_FAILED;

}

else

{

return JT\_SUCCESS;

}

}

static JT\_StatusCode\_t JTGUI\_WND\_Bg()

{

JTGUI\_TEXT\_t stText;

JTGUI\_BACKGROUND\_t stBg;

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stBg.stBgType = JTGUI\_BGTYPE\_NONE;

//JTGUIBMP\_DisplayBitmap(ANTENNA\_MENU\_WND\_X, ANTENNA\_MENU\_WND\_Y, ui\_BackgroudIfor.bWidth, ui\_BackgroudIfor.bHeight,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET, &ui\_BackgroudIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(0, 0, ui\_SubMenuIfor.bWidth, ui\_SubMenuIfor.bHeight, 0, 0, &ui\_SubMenuIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(186, 255, ui\_SelectItemIfor.bWidth, ui\_SelectItemIfor.bHeight, 0, 0, &ui\_SelectItemIfor, JT\_TRUE);

JTGUI\_DrawWholeBG\_And\_Selector();

JTGUIPNG\_DrawPngEx(PATH\_INSTALLATION\_BIG\_ICON\_PNG, JTGUI\_SUBMENU\_BOX\_ICON\_X, JTGUI\_SUBMENU\_BOX\_ICON\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

stText.pu8Text = msg\_antenna\_setup[stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(JTGUI\_SUBWND\_TITLE\_X, JTGUI\_SUBWND\_TITLE\_Y, JTGUI\_SUBWND\_TITLE\_W, JTGUI\_SUBWND\_TITLE\_H, JTGUI\_COLOR\_WHITE, JTGUI\_FONTSIZE\_BIG, EN\_JTGUI\_TEXTALIGN\_LEFT, &stText, &stBg);

return JT\_SUCCESS;

}

static void JTGUI\_LIST\_GetMenuContent(JTGUI\_ANTENNA\_MENU\_DATA\_t\* ptData, JT\_S32 s32ItemIndex, JTGUI\_TEXT\_t \*pText, JT\_U32 bufSize)

{

if (pText == NULL || pText->pu8Text == NULL || s32ItemIndex >= JTGUI\_ANTENNA\_MENU\_MAX)

{

return;

}

memset(pText->pu8Text, 0, bufSize);

switch (s32ItemIndex)

{

case JTGUI\_ANTENNA\_MENU\_LNB\_PORT:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

pText->pu8Text = g\_puLNBName[ptData->CurlnbPort][stGlobalVar.enLanguage];

break;

}

case JTGUI\_ANTENNA\_MENU\_SATELLITE:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

snprintf(pText->pu8Text, bufSize, "%s(%d/%d)", ptData->curSat.name, ptData->curSatIndex + 1, ptData->satCnt[ptData->CurlnbPort]);

break;

}

case JTGUI\_ANTENNA\_MENU\_LNBFREQ:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

strncpy(pText->pu8Text, g\_pu8LnbFreq[ptData->curSat.lnb\_freq], bufSize);

break;

}

case JTGUI\_ANTENNA\_MENU\_DISEQC:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

pText->pu8Text = g\_pu8Diseqc[ptData->curSat.diseqc][stGlobalVar.enLanguage];

break;

}

case JTGUI\_ANTENNA\_MENU\_22K:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

pText->pu8Text = g\_pu8OnOff[ptData->curSat.\_22K][stGlobalVar.enLanguage];

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORTYPE:

{

pText->stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

pText->pu8Text = g\_pu8Motor\_Type[ptData->curSat.motor\_type][stGlobalVar.enLanguage];

break;

}

case JTGUI\_ANTENNA\_MENU\_LONGTITUDE:

{

JT\_S16 s16Longitude = 0;

JTAPIDM\_SyscfgGetLocalLongitude(&s16Longitude);

pText->stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

snprintf(pText->pu8Text, bufSize, "%03d.0%s", abs(s16Longitude), (s16Longitude >= 0) ? "E" : "W");

break;

}

case JTGUI\_ANTENNA\_MENU\_LATITUDE:

{

JT\_S16 s16Latitude = 0;

JTAPIDM\_SyscfgGetLocalLatitude(&s16Latitude);

pText->stTextType = JTGUI\_TEXT\_TYPE\_ASCII;

snprintf(pText->pu8Text, bufSize, "%03d.0%s", abs(s16Latitude), (s16Latitude >= 0) ? "N" : "S");

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORLIMIT:

{

JTAPIDM\_SYSCFG\_MOTORLIMITTYPE\_t enlimit;

JTAPIDM\_SyscfgGetMotorLimitType(&enlimit);

pText->stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

if (enlimit == JTAPIDM\_SYSCFG\_MOTORLIMITTYPE\_DISABLE)

{

pText->pu8Text = msg\_disable[stGlobalVar.enLanguage];

}

else

{

pText->pu8Text = msg\_enable[stGlobalVar.enLanguage];

}

break;

}

}

}

static JT\_S32 GetItemY(JT\_S32 idx)

{

if (idx < (ANTENNA\_MENU\_LIST\_MAX - 1) / 2)

{

return ANTENNA\_MENU\_LIST\_Y + idx \* ANTENNA\_MENU\_SELECT\_NAME\_H;

}

else

{

return ANTENNA\_MENU\_SELECT\_NAME\_DOWN\_Y + (idx - (ANTENNA\_MENU\_LIST\_MAX - 1) / 2) \* ANTENNA\_MENU\_SELECT\_NAME\_H;

}

}

static void drawIcon(JT\_S32 index)

{

switch (index)

{

case JTGUI\_ANTENNA\_MENU\_LNB\_PORT:

case JTGUI\_ANTENNA\_MENU\_SATELLITE:

case JTGUI\_ANTENNA\_MENU\_LNBFREQ:

case JTGUI\_ANTENNA\_MENU\_DISEQC:

case JTGUI\_ANTENNA\_MENU\_22K:

case JTGUI\_ANTENNA\_MENU\_MOTORTYPE:

{

JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_L\_PNG,ITEM\_ARROW\_LEFT\_X, ITEM\_ARROW\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_R\_PNG,ITEM\_ARROW\_RIGHT\_X, ITEM\_ARROW\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

break;

}

case JTGUI\_ANTENNA\_MENU\_LONGTITUDE:

case JTGUI\_ANTENNA\_MENU\_LATITUDE:

{

JTGUIPNG\_DrawPngEx(PATH\_PEN\_PNG, ITEM\_PEN\_X, ITEM\_ARROW\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

break;

}

case JTGUI\_ANTENNA\_MENU\_MOTORLIMIT:

{

break;

}

}

}

static JT\_BOOL JTGUI\_LIST\_DrawMenuItem(HWND hWnd, JT\_S32 s32RowIndex, JT\_S32 s32ItemIndex, JT\_BOOL bSelect)

{

JTGUI\_ANTENNA\_MENU\_DATA\_t\* ptData = NULL;

JT\_U8 tmpBuf[32];

JT\_S32 i = 0;

JT\_U16 otherIndex = 0;

JTGUI\_TEXT\_t stText = {0};

JTGUI\_BACKGROUND\_t stBg = {0};

JT\_U32 otherFontColor = 0;

if (bSelect == JT\_FALSE)

{

return JT\_TRUE;

}

stBg.stBgType = JTGUI\_BGTYPE\_NONE;

ptData = (JTGUI\_ANTENNA\_MENU\_DATA\_t \*)WND\_GetData(hWnd);

if (ptData->bChange == JT\_FALSE)

{

antennaSetup\_GetCurSatInfo(ptData);

}

//selected

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stText.pu8Text = g\_MenuName[s32ItemIndex][stGlobalVar.enLanguage];

if (s32ItemIndex == JTGUI\_ANTENNA\_MENU\_MOTORTYPE)

{

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X, ANTENNA\_MENU\_SELECT\_NAME\_MID\_Y, ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_MEDIUM, EN\_JTGUI\_TEXTALIGN\_LEFT , &stText, &stBg);

}

else

{

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X, ANTENNA\_MENU\_SELECT\_NAME\_MID\_Y, ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_BIG, EN\_JTGUI\_TEXTALIGN\_LEFT , &stText, &stBg);

}

stText.pu8Text = tmpBuf;

if (s32ItemIndex == JTGUI\_ANTENNA\_MENU\_SATELLITE)

{

JTGUI\_LIST\_GetMenuContent(ptData, s32ItemIndex, &stText, 28);

if(strlen(stText.pu8Text) >= 27)

{

strcat(stText.pu8Text , "...");

}

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X + 30, ANTENNA\_MENU\_SELECT\_NAME\_MID\_Y + 10, ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_MEDIUM, EN\_JTGUI\_TEXTALIGN\_RIGHT, &stText, &stBg);

}

else

{

JTGUI\_LIST\_GetMenuContent(ptData, s32ItemIndex, &stText, 32);

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X + 30, ANTENNA\_MENU\_SELECT\_NAME\_MID\_Y, ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

JTGUI\_COLOR\_HLCOLOR, JTGUI\_FONTSIZE\_BIG, EN\_JTGUI\_TEXTALIGN\_RIGHT, &stText, &stBg);

}

drawIcon(s32ItemIndex);

//others

for (i = 0 ; i < ANTENNA\_MENU\_LIST\_MAX; i++)

{

if (i == (ANTENNA\_MENU\_LIST\_MAX - 1) / 2)

{

//selected one

continue;

}

else

{

otherIndex = (s32ItemIndex + JTGUI\_ANTENNA\_MENU\_MAX + i - (ANTENNA\_MENU\_LIST\_MAX - 1) / 2) % JTGUI\_ANTENNA\_MENU\_MAX;

}

if (otherIndex>= JTGUI\_ANTENNA\_MENU\_LNB\_PORT && otherIndex <=JTGUI\_ANTENNA\_MENU\_LNBFREQ)

{

otherFontColor = JTGUI\_COLOR\_WHITE;

}

else

{

if (ptData->lnbType[ptData->CurlnbPort] == JTAPIDM\_SYSCFG\_LNBTYPE\_FIX)

{

if (otherIndex >=JTGUI\_ANTENNA\_MENU\_DISEQC && otherIndex <= JTGUI\_ANTENNA\_MENU\_22K)

{

otherFontColor = JTGUI\_COLOR\_WHITE;

}

else

{

otherFontColor = JTGUI\_COLOR\_GREY;

}

}

else

{

if (otherIndex >=JTGUI\_ANTENNA\_MENU\_DISEQC && otherIndex <= JTGUI\_ANTENNA\_MENU\_22K)

{

otherFontColor = JTGUI\_COLOR\_GREY;

}

else

{

otherFontColor = JTGUI\_COLOR\_WHITE;

}

if (!ptData->curSat.motor\_type)

{

if (otherIndex >= JTGUI\_ANTENNA\_MENU\_LONGTITUDE && otherIndex <= JTGUI\_ANTENNA\_MENU\_LATITUDE)

{

otherFontColor = JTGUI\_COLOR\_GREY;

}

}

}

}

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stText.pu8Text = g\_MenuName[otherIndex][stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X, GetItemY(i), ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

otherFontColor, JTGUI\_FONTSIZE\_MEDIUM, EN\_JTGUI\_TEXTALIGN\_LEFT | EN\_JTGUI\_TEXTALIGN\_CENTER, &stText, &stBg);

stText.pu8Text = tmpBuf;

JTGUI\_LIST\_GetMenuContent(ptData, otherIndex, &stText, 32);

JTGUI\_DrawTextLabel(ANTENNA\_MENU\_SELECT\_NAME\_X, GetItemY(i), ANTENNA\_MENU\_SELECT\_NAME\_W, ANTENNA\_MENU\_SELECT\_NAME\_H, \

otherFontColor, JTGUI\_FONTSIZE\_MEDIUM, EN\_JTGUI\_TEXTALIGN\_RIGHT | EN\_JTGUI\_TEXTALIGN\_CENTER, &stText, &stBg);

}

return JT\_TRUE;

}

static JT\_StatusCode\_t JTGUI\_Init\_scroll(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

POINT stPoint;

stPoint.X = ANTENNA\_MENU\_LIST\_X;

stPoint.Y = ANTENNA\_MENU\_LIST\_Y;

ptData->hMenu = SCROLL\_Init(g\_hANTENNA\_MENU, stPoint, ANTENNA\_MENU\_LIST\_W, ANTENNA\_MENU\_LIST\_H + 50, 1, JTGUI\_ANTENNA\_MENU\_MAX, 0, JT\_FALSE, JTGUI\_LIST\_DrawMenuItem);

if (ptData->hMenu == NULL)

{

JT\_ERROR(("JTGUI\_Init\_scroll Init Menu Failed\n"));

return JT\_ERROR\_INVALID\_HANDLE;

}

return JT\_SUCCESS;

}

static antennaSetup\_initData(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

memset(ptData, 0, sizeof(JTGUI\_ANTENNA\_MENU\_DATA\_t));

ptData->bChange = JT\_FALSE;

ptData->bRefreshWhole = JT\_TRUE;

ptData->bTerm = JT\_FALSE;

antennaSetup\_InitLnb(&ptData->lnbCnt);

if (ptData->satCnt[0] != 0)

{

ptData->CurlnbPort = 0;

ptData->curSatIndex = 0;

}

else if (ptData->satCnt[1] != 0)

{

ptData->CurlnbPort = 1;

ptData->curSatIndex = 0;

}

else

{

}

JTAPIDM\_SyscfgGetLNBType(JT\_DMD\_PORT\_0, &ptData->lnbType[JT\_DMD\_PORT\_0]);

JTAPIDM\_SyscfgGetLNBType(JT\_DMD\_PORT\_1, &ptData->lnbType[JT\_DMD\_PORT\_1]);

}

static JT\_StatusCode\_t antennaSetup\_loadData(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

JT\_U16 \*pSatIDTemp = NULL;

JT\_U32 i = 0, j = 0;

if (ptData == NULL)

{

return JT\_ERROR\_BAD\_PARAMETER;

}

pSatIDTemp = (JT\_U16\*)JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* sizeof(JT\_U16));

if (pSatIDTemp == NULL)

{

goto ERR;

}

if (ptData->lnbSatId[0] == NULL)

{

ptData->lnbSatId[0] = (JT\_U16\*)JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* sizeof(JT\_U16));

if (ptData->lnbSatId[0] == NULL)

{

goto ERR;

}

}

if (ptData->lnbSatId[1] == NULL)

{

ptData->lnbSatId[1] = (JT\_U16\*)JTOS\_MemMalloc(MAX\_SAT\_NUMBER \* sizeof(JT\_U16));

if (ptData->lnbSatId[1] == NULL)

{

goto ERR;

}

}

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_0, pSatIDTemp, &ptData->satCnt[0]);

for (i = 0; i < ptData->satCnt[0]; i++)

{

if (pSatIDTemp[i] < SAT\_ID\_ISDB\_T)

{

(ptData->lnbSatId[0])[j] = pSatIDTemp[i];

j++;

}

}

ptData->satCnt[0] = j;

JTAPIDM\_SatGetList(JT\_DMD\_PORT\_1, ptData->lnbSatId[1], &ptData->satCnt[1]);

JTOS\_MemFree(pSatIDTemp);

JTGUI\_DiseqcSwitchPort(ptData);

JTGUI\_Activate22KHzTone(ptData);

return JT\_SUCCESS;

ERR:

if (pSatIDTemp != NULL)

{

JTOS\_MemFree(pSatIDTemp);

}

return JT\_FAILED;

}

void JTGUI\_ANTENNA\_Term(JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData)

{

if (ptData == NULL)

{

return ;

}

if (ptData->hMenu != NULL)

{

SCROLL\_Term(ptData->hMenu);

ptData->hMenu = NULL;

}

if (ptData->lnbSatId[0] != NULL)

{

JTOS\_MemFree(ptData->lnbSatId[0]);

ptData->lnbSatId[0] = NULL;

}

if (ptData->lnbSatId[1] != NULL)

{

JTOS\_MemFree(ptData->lnbSatId[1]);

ptData->lnbSatId[1] = NULL;

}

}

static JT\_S32 JTGUI\_ANTENNA\_MENU\_Pro(HWND hWnd, JT\_S16 sMsg, JT\_S16 sKey, JT\_S32 iParam1, JT\_S32 iParam2)

{

JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData = NULL;

JT\_S32 temp = 0, curIndex = 0;

ptData = (JTGUI\_ANTENNA\_MENU\_DATA\_t \*)WND\_GetData(hWnd);

if(NULL == ptData)

{

JT\_ERROR((">>>#error: %d, %s: WND\_GetData Failed.\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return 0;

}

SCROLL\_GetIndex(ptData->hMenu, &temp, &curIndex);

switch (sMsg)

{

case WM\_CREATE:

{

antennaSetup\_initData(ptData);

antennaSetup\_loadData(ptData);

JTGUI\_Init\_scroll(ptData);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case WM\_SHOW:

{

JTGUI\_WND\_Bg();

SCROLL\_StartNavi(ptData->hMenu);

if (ptData->bRefreshWhole == JT\_TRUE)

{

ptData->bRefreshWhole = JT\_FALSE;

JTGUIOSD\_RefreshRectangle(ANTENNA\_MENU\_WND\_X, ANTENNA\_MENU\_WND\_Y, ANTENNA\_MENU\_WND\_W, ANTENNA\_MENU\_WND\_H);

}

else

{

JTGUIOSD\_RefreshRectangle(ANTENNA\_MENU\_WND\_X, ANTENNA\_MENU\_WND\_Y,ANTENNA\_MENU\_WND\_W, ANTENNA\_MENU\_WND\_H);

}

break;

}

case WM\_KEYDOWN:

{

switch (sKey)

{

case KEY\_UP:

case KEY\_DOWN:

{

JT\_S32 step = (sKey == KEY\_UP ? -1 : 1);

while(1)

{

curIndex = (step + curIndex + JTGUI\_ANTENNA\_MENU\_MAX) % JTGUI\_ANTENNA\_MENU\_MAX;

if (ptData->lnbType[ptData->CurlnbPort] == JTAPIDM\_SYSCFG\_LNBTYPE\_FIX)

{

if (curIndex >= JTGUI\_ANTENNA\_MENU\_LNB\_PORT && curIndex <= JTGUI\_ANTENNA\_MENU\_22K)

{

break;

}

}

else

{

if (curIndex < JTGUI\_ANTENNA\_MENU\_DISEQC || curIndex > JTGUI\_ANTENNA\_MENU\_22K)

{

if (!ptData->curSat.motor\_type)

{

if (curIndex > JTGUI\_ANTENNA\_MENU\_LATITUDE || curIndex < JTGUI\_ANTENNA\_MENU\_LONGTITUDE)

{

break;

}

}

else

{

break;

}

}

}

}

SCROLL\_SetCurFocusItem(ptData->hMenu, curIndex);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case KEY\_LEFT:

case KEY\_RIGHT:

{

antennaSetup\_SwitchContent(ptData, curIndex, sKey == KEY\_LEFT ? -1 : 1);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case KEY\_OK:

{

if (ptData->bTerm == JT\_FALSE)

{

antennaSetup\_PopupWidget(ptData, curIndex);

}

break;

}

case KEY\_BACK:

{

ptData->bTerm = JT\_TRUE;

if (ptData->bChange == JT\_TRUE)

{

antennaSetup\_PopupSaveChange(ptData);

}

WND\_PostMessage(hWnd, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_TERM, WND\_COMMAND\_TERM\_NORMAL\_MODE);

break;

}

}

}

case WM\_COMMAND:

{

switch(iParam1)

{

case WND\_COMMAND\_TERM:

{

JTGUI\_ANTENNA\_Term(ptData);

WND\_Destroy(hWnd);

g\_hANTENNA\_MENU = NULL;

#if defined(JT\_MENU\_KEEP\_VIDEO) && !defined(JT\_CUSTOMER\_MALDIVES)

JT\_U32 u32ChID = 0;

JT\_StatusCode\_t statusCode;

statusCode = JTAPIDM\_ServiceInfoListGetChid(g\_groupIndex, g\_serviceIndexInGroup, &u32ChID);

statusCode = JTAPIPLAY\_Open(u32ChID, AV\_DECODER\_MAIN);

JT\_ASSERT(JT\_SUCCESS == statusCode);

#endif

break;

}

case WND\_COMMAND\_REPOST\_MSG:

{

JTGUI\_PopupMessage(g\_hANTENNA\_MENU, msg\_information[stGlobalVar.enLanguage], msg\_wrong\_date\_format[stGlobalVar.enLanguage]);

break;

}

}

break;

}

default:

break;

}

return MAINWND\_DefProc(hWnd, sMsg, sKey, iParam1, iParam2);

}

JT\_StatusCode\_t JTGUI\_ANTENNA\_SETUP\_Setup(HWND hParentWnd)

{

RECT tRect;

JT\_BOOL bRet;

JTGUI\_ANTENNA\_MENU\_DATA\_t \*ptData = NULL;

JTGUI\_Check\_Handle(hParentWnd);

if (JTGUI\_CheckSatCount(hParentWnd) != JT\_SUCCESS)

{

#if defined(JT\_MENU\_KEEP\_VIDEO) && !defined(JT\_CUSTOMER\_MALDIVES)

JT\_U32 u32ChID = 0;

JT\_StatusCode\_t statusCode;

statusCode = JTAPIDM\_ServiceInfoListGetChid(g\_groupIndex, g\_serviceIndexInGroup, &u32ChID);

statusCode = JTAPIPLAY\_Open(u32ChID, AV\_DECODER\_MAIN);

JT\_ASSERT(JT\_SUCCESS == statusCode);

#endif

return JT\_FAILED;

}

if(bRet == JT\_FALSE)

{

return JT\_SUCCESS;

}

tRect.X = ANTENNA\_MENU\_WND\_X;

tRect.Y = ANTENNA\_MENU\_WND\_Y;

tRect.W = ANTENNA\_MENU\_WND\_W;

tRect.H = ANTENNA\_MENU\_WND\_H;

g\_hANTENNA\_MENU = WND\_Create(ANTENNA\_MENU\_NAME, &tRect, sizeof(JTGUI\_ANTENNA\_MENU\_DATA\_t), JTGUI\_ANTENNA\_MENU\_Pro, hParentWnd);

if(g\_hANTENNA\_MENU== NULL)

{

JT\_DEBUG(("error : Antenna Setup Created Failed!!! #%d\n", \_\_LINE\_\_));

return JT\_ERROR\_INVALID\_HANDLE;

}

WND\_SetFocus(g\_hANTENNA\_MENU);

return JT\_SUCCESS;

}

HWND JTGUI\_ANTENNA\_MENU\_GetHandle()

{

return g\_hANTENNA\_MENU;

}

#include "ui\_com.h"

#include "hdi\_dmd.h"

#include "hdi\_osd.h"

#include "api\_search.h"

#include "api\_dm.h"

#include "api\_timer\_time.h"

#include "mainwnd.h"

#if defined(JT\_CUSTOMER\_MALDIVES)

#define POPUPBOX\_STYLE\_FOR\_MEDIANET

#endif

#define SATELLITE\_SEARCH\_NAME "Satellite Search"

#define SATELLITE\_SEARCH\_WND\_X 0

#define SATELLITE\_SEARCH\_WND\_Y 0

#define SATELLITE\_SEARCH\_WND\_W WINDOW\_MAX\_WIDTH

#define SATELLITE\_SEARCH\_WND\_H WINDOW\_MAX\_HEIGHT

#define SATEL\_SEARCH\_BUTTON\_X(index) 155+(index)\*250

#define SATEL\_SEARCH\_BUTTON\_Y(index) 625

#define SATEL\_SEARCH\_BUTTON\_INFO\_X(index) 200+(index)\*1250

#define SATEL\_SEARCH\_BUTTON\_INFO\_Y(index) 635

#define ITEM\_LEFT 200

#define ITEM\_MIDDLE 335

#define ITEM\_Y(index) (index%2 ==0) ? (218-(index/2-1)\*45) : (460+((index+1)/2-1)\*45)

#define ITEM\_RIGHT\_X 815

#define ITEM\_HEIGHT 40

#define ITEM\_WIDTH 600

#define ITEM\_ARROW\_LEFT\_X 960

#define ITEM\_ARROW\_RIGHT\_X 990

#define ITEM\_ARROW\_W 100

#define ITEM\_ARROW\_H 50

#define ITEM\_ARROW\_Y 340

#define ITEM\_POP\_BG\_X 1040

#define ITEM\_POP\_BG\_Y 265

#define ITEM\_POP\_BG\_W 190

#define ITEM\_POP\_BG\_H 183

#define ITEM\_POP\_TEXT\_Y\_MID 345

#define TEXT\_SPACE\_LEFT 12

#define TEXT\_SPACE\_RIGHT 50

#define MIDDLE\_SPACE (30 \* WINDOW\_MAX\_WIDTH / 720)

static HWND g\_hSatellite\_Search = NULL;

static JT\_U8 g\_arrow\_left = JT\_FALSE;

static JT\_U8 g\_arrow\_right = JT\_FALSE;

#define SS\_DEBUG(x, arg...) //JT\_TRACE(x, ##arg)

typedef enum

{

JTGUI\_SATELLITE\_SEARCH\_MENU\_MIN,

JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE = JTGUI\_SATELLITE\_SEARCH\_MENU\_MIN,

JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA,

JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE,

JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH,

JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE,

JTGUI\_SATELLITE\_SEARCH\_MENU\_MAX

}JTGUI\_SATELLITE\_SEARCH\_MENU\_t;

typedef struct

{

JT\_U8 u8FTAIndex;

JT\_U8 u8ScanChannelIndex;

JT\_U8 u8NetworkSearchIndex;

JT\_U8 u8ScanModeIndex;

JT\_U8 au8Satellite[300][SAT\_NAME\_MAXLEN];

JT\_U16 u16SatelliteIndex;

JT\_U16 u16SatelliteCount;

JT\_U16 au16SatID[300];

JTGUI\_SATELLITE\_SEARCH\_MENU\_t enMenuFocusIndex;

JT\_U32 u32Timer;

}JTGUI\_SATELLITE\_SEARCH\_DATA\_t;

JT\_U8 \*g\_pu8BottomInfo[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_multi\_search),

JTGUI\_FILLTITLE(msg\_search)

};

JT\_U8 \*g\_pu8YesNo[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_no),

JTGUI\_FILLTITLE(msg\_yes)

};

JT\_U8 \*g\_pu8TvRadio[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_tv\_radio),

JTGUI\_FILLTITLE(msg\_tv),

JTGUI\_FILLTITLE(msg\_radio)

};

JT\_U8 \*g\_pu8ScanMode[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_auto\_scan),

JTGUI\_FILLTITLE(msg\_preset\_scan)

};

static JT\_StatusCode\_t JTGUI\_DrawEachItem(JT\_S32 s32X, JT\_S32 s32Y, JT\_U8 \*pu8Title, JT\_U8 \*pu8Content, JTGUI\_TEXTTYPE\_t stContentTextType, JT\_BOOL bSelect)

{

JTGUI\_TEXT\_t stText;

JTGUI\_FONTSIZE\_t fontsize;

JTGUI\_COLOR\_t enColor;

JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData;

ptData = (JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*)WND\_GetData(g\_hSatellite\_Search);

if(bSelect)

{

enColor = JTGUI\_COLOR\_HLCOLOR;

fontsize = JTGUI\_FONTSIZE\_BIG;

}

else

{

enColor = JTGUI\_COLOR\_WHITE;

fontsize = JTGUI\_FONTSIZE\_MEDIUM;

}

if(pu8Title == msg\_network\_search[stGlobalVar.enLanguage])

{

enColor = (ptData->u8ScanModeIndex == 0) ? JTGUI\_COLOR\_GREY : enColor;

}

JT\_S32 s32Num;

JTGUI\_BACKGROUND\_t stBg;

stBg.stBgType = JTGUI\_BGTYPE\_NONE;

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stText.pu8Text = pu8Title;

JTGUI\_DrawTextLabel(s32X,s32Y,JTGUI\_SUBWND\_TEXT\_W+100,ITEM\_HEIGHT, enColor, fontsize, \

EN\_JTGUI\_TEXTALIGN\_LEFT,&stText,&stBg);

stText.stTextType = stContentTextType;

stText.pu8Text = pu8Content;

//diaplay the satellite name with the medium fontsize

if ((pu8Title == msg\_satellite[stGlobalVar.enLanguage])&&(bSelect))

{

fontsize = JTGUI\_FONTSIZE\_MEDIUM;

snprintf(stText.pu8Text, 30, "%s", pu8Content);

if(strlen(stText.pu8Text) >= 29)

{

strcat(stText.pu8Text, "...");

}

JTGUI\_DrawTextLabel(s32X + 160, s32Y + 10, ITEM\_WIDTH - 60, ITEM\_HEIGHT, enColor, fontsize, EN\_JTGUI\_TEXTALIGN\_RIGHT, &stText, &stBg);

}

else

{

JTGUI\_DrawTextLabel(s32X + 100, s32Y, ITEM\_WIDTH, ITEM\_HEIGHT, enColor, fontsize, EN\_JTGUI\_TEXTALIGN\_RIGHT, &stText, &stBg);

}

return JT\_SUCCESS;

}

static JT\_StatusCode\_t JTGUI\_DrawIndexItem(JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData, JTGUI\_SATELLITE\_SEARCH\_MENU\_t enMenuFocusIndex, JT\_S32 s32X, JT\_S32 s32Y, JT\_BOOL bSelect)

{

switch(enMenuFocusIndex)

{

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE:

SS\_DEBUG("ptData->au8Satallite = %s\n", ptData->au8Satellite[ptData->u16SatelliteIndex]);

JTGUI\_DrawEachItem(s32X,s32Y,msg\_satellite[stGlobalVar.enLanguage], ptData->au8Satellite[ptData->u16SatelliteIndex], JTGUI\_TEXT\_TYPE\_ASCII, bSelect);

break;

case JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA:

JTGUI\_DrawEachItem(s32X,s32Y,msg\_fta\_only[stGlobalVar.enLanguage], g\_pu8YesNo[ptData->u8FTAIndex][stGlobalVar.enLanguage], JTGUI\_TEXT\_TYPE\_UNICODE, bSelect);

break;

case JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE:

JTGUI\_DrawEachItem(s32X,s32Y,msg\_scan\_channel[stGlobalVar.enLanguage], g\_pu8TvRadio[ptData->u8ScanChannelIndex][stGlobalVar.enLanguage], JTGUI\_TEXT\_TYPE\_UNICODE, bSelect);

break;

case JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH:

JTGUI\_DrawEachItem(s32X,s32Y,msg\_network\_search[stGlobalVar.enLanguage], g\_pu8YesNo[ptData->u8NetworkSearchIndex][stGlobalVar.enLanguage], JTGUI\_TEXT\_TYPE\_UNICODE, bSelect);

break;

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE:

JTGUI\_DrawEachItem(s32X,s32Y,msg\_scan\_mode[stGlobalVar.enLanguage], g\_pu8ScanMode[ptData->u8ScanModeIndex][stGlobalVar.enLanguage], JTGUI\_TEXT\_TYPE\_UNICODE, bSelect);

break;

default:

break;

}

}

static JT\_StatusCode\_t JTGUI\_WND\_Middle\_Bg()

{

//JTGUIBMP\_DisplayPartOfBitmap(ITEM\_LEFT, 80,WINDOW\_MAX\_WIDTH - ITEM\_LEFT, 500, ITEM\_LEFT, 80, &ui\_BackgroudIfor, JT\_TRUE);'

JTGUI\_Draw\_Part\_of\_PNG(PATH\_BACKGROUND\_LEFT\_PNG,ITEM\_LEFT, 80,WINDOW\_MAX\_WIDTH - ITEM\_LEFT, 500,ITEM\_LEFT - PATH\_CUPID\_MENU3\_PNG\_X, 80);

//JTGUIBMP\_DisplayBitmap(186,255,ui\_SelectItemIfor.bWidth, ui\_SelectItemIfor.bHeight, 0, 0, &ui\_SelectItemIfor, JT\_TRUE);

JTGUIPNG\_DrawPngEx(PATH\_CUPID\_MENU3\_PNG, PATH\_CUPID\_MENU3\_PNG\_X, PATH\_CUPID\_MENU3\_PNG\_Y, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

}

static JT\_StatusCode\_t JTGUI\_WND\_Bg()

{

JTGUI\_TEXT\_t stText;

JTGUI\_BACKGROUND\_t stBg;

JT\_S32 s32Num;

stText.stTextType = JTGUI\_TEXT\_TYPE\_UNICODE;

stBg.stBgType = JTGUI\_BGTYPE\_NONE;

//JTGUIBMP\_DisplayBitmap(SATELLITE\_SEARCH\_WND\_X, SATELLITE\_SEARCH\_WND\_Y, ui\_BackgroudIfor.bWidth, ui\_BackgroudIfor.bHeight,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET, &ui\_BackgroudIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(0, 0,ui\_SubMenuIfor.bWidth, ui\_SubMenuIfor.bHeight, 0, 0, &ui\_SubMenuIfor, JT\_TRUE);

//JTGUIBMP\_DisplayBitmap(186,255,ui\_SelectItemIfor.bWidth, ui\_SelectItemIfor.bHeight, 0, 0, &ui\_SelectItemIfor, JT\_TRUE);

JTGUI\_DrawWholeBG\_And\_Selector();

//JTGUIBMP\_DisplayBitmap(JTGUI\_SUBMENU\_BOX\_ICON\_X,JTGUI\_SUBMENU\_BOX\_ICON\_Y,ui\_Installation\_Big\_iconIfor.bWidth,ui\_Installation\_Big\_iconIfor.bHeight,0,0,&ui\_Installation\_Big\_iconIfor,JT\_TRUE);

JTGUIPNG\_DrawPngEx(PATH\_INSTALLATION\_BIG\_ICON\_PNG, JTGUI\_SUBMENU\_BOX\_ICON\_X,JTGUI\_SUBMENU\_BOX\_ICON\_Y,JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

stText.pu8Text = msg\_satellite\_search[stGlobalVar.enLanguage];

JTGUI\_DrawTextLabel(JTGUI\_SUBWND\_TITLE\_X, JTGUI\_SUBWND\_TITLE\_Y, JTGUI\_SUBWND\_TITLE\_W, JTGUI\_SUBWND\_TITLE\_H, JTGUI\_COLOR\_WHITE, JTGUI\_FONTSIZE\_BIG, EN\_JTGUI\_TEXTALIGN\_LEFT, &stText, &stBg);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_GREEN\_PNG,SATEL\_SEARCH\_BUTTON\_X(0), SATEL\_SEARCH\_BUTTON\_Y(0), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_FUNCTION\_YELLOW\_PNG,SATEL\_SEARCH\_BUTTON\_X(1) + 250, SATEL\_SEARCH\_BUTTON\_Y(1), JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_BACK\_BUTTON\_PNG,JTGUI\_SUBWND\_ICON\_BACK\_X,JTGUI\_SUBWND\_ICON\_BACK\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_MEDIUM);

JTGUITEXT\_DrawUnicodeTxt(SATEL\_SEARCH\_BUTTON\_INFO\_X(0), SATEL\_SEARCH\_BUTTON\_INFO\_Y(0), 500, MSG\_LABLE\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, g\_pu8BottomInfo[0][stGlobalVar.enLanguage], 0, \

JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

JTGUITEXT\_DrawUnicodeTxt(SATEL\_SEARCH\_BUTTON\_INFO\_X(1) + 530, SATEL\_SEARCH\_BUTTON\_INFO\_Y(1), MSG\_LABLE\_W\*2, MSG\_LABLE\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, g\_pu8BottomInfo[1][stGlobalVar.enLanguage], 0, \

JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

JTGUITEXT\_DrawUnicodeTxt(JTGUI\_SUBWND\_INFO\_BACK\_X, JTGUI\_SUBWND\_INFO\_BACK\_Y, MSG\_LABLE\_W, MSG\_LABLE\_H, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET, msg\_back[stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 0, &s32Num);

return JT\_SUCCESS;

}

static JT\_StatusCode\_t JTGUI\_DrawUI(JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData)

{

JTGUI\_COLOR\_t enColor;

JT\_S32 s32Index;

JT\_S32 i;

JTGUI\_WND\_Middle\_Bg();

if(g\_arrow\_left == JT\_FALSE)

{ JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_L\_PNG,ITEM\_ARROW\_LEFT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET); //

}

else

{ JTGUIPNG\_DrawPngEx(PATH\_ARROW\_BLUE\_L\_PNG,ITEM\_ARROW\_LEFT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

TIMER\_SET\_SAFE(ptData->u32Timer, g\_hSatellite\_Search, TIMER\_ONCE, 200);

g\_arrow\_left = JT\_FALSE;

}

if(g\_arrow\_right == JT\_FALSE)

{ JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_R\_PNG,ITEM\_ARROW\_RIGHT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

}

else

{

TIMER\_SET\_SAFE(ptData->u32Timer, g\_hSatellite\_Search, TIMER\_ONCE, 200); JTGUIPNG\_DrawPngEx(PATH\_ARROW\_BLUE\_R\_PNG,ITEM\_ARROW\_RIGHT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

g\_arrow\_right = JT\_FALSE;

}

s32Index = ptData->enMenuFocusIndex;

JTGUI\_DrawIndexItem(ptData, s32Index,ITEM\_LEFT, ITEM\_MIDDLE, JT\_TRUE);

//

for(i=0; i < 2; i++)

{

s32Index = (ptData->enMenuFocusIndex + JTGUI\_SATELLITE\_SEARCH\_MENU\_MAX - i - 1) % JTGUI\_SATELLITE\_SEARCH\_MENU\_MAX;

JTGUI\_DrawIndexItem(ptData, s32Index, ITEM\_LEFT, ITEM\_Y(2 \* (i + 1)), JT\_FALSE);

}

//

for(i = 0; i < 2; i++)

{

s32Index = (ptData->enMenuFocusIndex + i + 1) % JTGUI\_SATELLITE\_SEARCH\_MENU\_MAX;

JTGUI\_DrawIndexItem(ptData, s32Index,ITEM\_LEFT, ITEM\_Y(2\*i+1), JT\_FALSE);

}

return JT\_SUCCESS;

}

static void JTGUI\_ChangeItem(JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData, JT\_BOOL bKeyUpDown)

{

if( bKeyUpDown == 1 )

{

if(ptData->enMenuFocusIndex == JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE)

{

if(ptData->u8ScanModeIndex == 0)

{

ptData->enMenuFocusIndex = JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE;

return;

}

}

if(ptData->enMenuFocusIndex == JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE)

{

ptData->enMenuFocusIndex = JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE;

}

else

{

ptData->enMenuFocusIndex++;

}

}

else if(bKeyUpDown == -1)

{

if(ptData->enMenuFocusIndex == JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE)

{

if(ptData->u8ScanModeIndex == 0)

{

ptData->enMenuFocusIndex = JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE;

return;

}

}

if(ptData->enMenuFocusIndex == JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE)

{

ptData->enMenuFocusIndex = JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE;

}

else

{

ptData->enMenuFocusIndex--;

}

}

}

static JT\_StatusCode\_t fnItemDrawFunc(HWND hWnd, JT\_S32 iRowIndex, JT\_S32 iItemIndex, JT\_BOOL bSelect)

{

JT\_S32 s32WNum;

JT\_S32 s32X, s32Y;

JT\_U16 u16Width;

JT\_S32 s32\_num;

JT\_S32 s32\_len;

JT\_S32 s32\_lists = 5;

JT\_S32 i;

PLISTBOXWND\_DATA\_t pstData;

JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData;

pstData = (PLISTBOXWND\_DATA\_t)WND\_GetData(hWnd);

ptData = (JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*)WND\_GetData(g\_hSatellite\_Search);

if(iItemIndex>=0)

{

switch(ptData->enMenuFocusIndex)

{

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE:

{

JT\_S32 s32TotalItems = ptData->u16SatelliteCount;

if(s32TotalItems > 0)

{

s32\_lists = (s32\_lists >= s32TotalItems)?(s32TotalItems):(s32\_lists);

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_SMALL);

s32\_len = JTGUITEXT\_GetAsciiTxtWidth(16, ptData->au8Satellite[iItemIndex]);

JTGUITEXT\_DrawAsciiTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2,ITEM\_POP\_TEXT\_Y\_MID, s32\_len, 50, 0, 0, ptData->au8Satellite[iItemIndex], 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 1, &s32WNum);

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_SMALL);

for(i=0; i< (s32\_lists-1)/2; i++)

{

s32\_num = (iItemIndex + s32TotalItems -i -1) % s32TotalItems;

s32\_len = JTGUITEXT\_GetAsciiTxtWidth(16, ptData->au8Satellite[s32\_num]);

JTGUITEXT\_DrawAsciiTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2,ITEM\_POP\_TEXT\_Y\_MID-35\*(i+1), s32\_len, 40, 0, 0, ptData->au8Satellite[s32\_num], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

}

for(i=0; i<s32\_lists/2 ; i++)

{

s32\_num = (iItemIndex + i + 1) % s32TotalItems;

s32\_len = JTGUITEXT\_GetAsciiTxtWidth(16, ptData->au8Satellite[s32\_num]);

JTGUITEXT\_DrawAsciiTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2,ITEM\_POP\_TEXT\_Y\_MID+35\*(i+1), s32\_len, 40, 0, 0, ptData->au8Satellite[s32\_num], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

}

}

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA:

case JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH:

{

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_NORMAL);

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8YesNo[iItemIndex][stGlobalVar.enLanguage]), g\_pu8YesNo[iItemIndex][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID, s32\_len, 50, 0, 0, g\_pu8YesNo[iItemIndex][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 1, &s32WNum);

s32\_num = (iItemIndex +1) % 2;

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_SMALL);

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8YesNo[s32\_num][stGlobalVar.enLanguage]), g\_pu8YesNo[s32\_num][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID+40, s32\_len, 40, 0, 0, g\_pu8YesNo[s32\_num][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE:

{

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_NORMAL);

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8TvRadio[iItemIndex][stGlobalVar.enLanguage]), g\_pu8TvRadio[iItemIndex][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID, s32\_len, 50, 0, 0, g\_pu8TvRadio[iItemIndex][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 1, &s32WNum);

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_SMALL);

s32\_num = (iItemIndex +3-1) % 3;

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage]), g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID-40, s32\_len, 40, 0, 0, g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

s32\_num = (iItemIndex +1) % 3;

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage]), g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID+40, s32\_len, 40, 0, 0, g\_pu8TvRadio[s32\_num][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE:

{

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_NORMAL);

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8ScanMode[iItemIndex][stGlobalVar.enLanguage]), g\_pu8ScanMode[iItemIndex][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID, s32\_len, 50, 0, 0, g\_pu8ScanMode[iItemIndex][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_HLCOLOR, 0, JT\_FALSE, 1, &s32WNum);

s32\_num = (iItemIndex +1) % 2;

JTGUITEXT\_SetFontHeight(JTGUI\_FONTSIZE\_SMALL);

s32\_len = JTGUITEXT\_GetUnicodeTxtWidth(JTGUI\_GetUnicodeLen(g\_pu8ScanMode[s32\_num][stGlobalVar.enLanguage]), g\_pu8ScanMode[s32\_num][stGlobalVar.enLanguage]);

JTGUITEXT\_DrawUnicodeTxt(ITEM\_POP\_BG\_X+(ITEM\_POP\_BG\_W - s32\_len)/2, ITEM\_POP\_TEXT\_Y\_MID+40, s32\_len, 40, 0, 0, g\_pu8ScanMode[s32\_num][stGlobalVar.enLanguage], 0, JTGUI\_COLOR\_WHITE, 0, JT\_FALSE, 1, &s32WNum);

break;

}

}

}

pstData->u32ItemIndex = iItemIndex;

return JT\_SUCCESS;

}

static JT\_StatusCode\_t JTGUI\_Init\_Data(JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData)

{

JT\_StatusCode\_t statusCode;

JTAPIDM\_SatInfo\_t sat\_info;

JT\_U16 \*sat\_id\_array;

JT\_U32 sat\_cnt, err\_sat = 0, i, j = 0;

sat\_id\_array = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* 300);

if (sat\_id\_array == NULL)

{

SS\_DEBUG("there is no memery");

return JT\_FAILED;

}

memset(sat\_id\_array, 0, sizeof(JT\_U16) \* 300);

statusCode = JTAPIDM\_SatGetAllPortList(sat\_id\_array, &sat\_cnt);

JT\_ASSERT(statusCode == JT\_SUCCESS);

j = 0;

for (i = 0; i < sat\_cnt; i++)

{

JTAPIDM\_SatGetById(sat\_id\_array[i], &sat\_info);

if (sat\_info.id >= SAT\_ID\_ISDB\_T)

{

err\_sat++;

}

}

for (i = 0; i < sat\_cnt; i++)

{

JTAPIDM\_SatGetById(sat\_id\_array[i], &sat\_info);

if (sat\_info.id < SAT\_ID\_ISDB\_T)

{

ptData->au16SatID[j] = sat\_id\_array[i];

snprintf(ptData->au8Satellite[j], SAT\_NAME\_MAXLEN, "%s(%d/%d)", sat\_info.name, j + 1, sat\_cnt - err\_sat);//C T ISDB-T UDP

//snprintf(ptData->au8Satellite[j], SAT\_NAME\_MAXLEN, "%s", sat\_info.name);

j++;

}

}

ptData->u16SatelliteCount = j;

ptData->u32Timer = INVALID\_TIMER;

return JT\_SUCCESS;

}

static JT\_S32 JTGUI\_SATELLITE\_SEARCH\_Pro(HWND hWnd, JT\_S16 sMsg, JT\_S16 sKey, JT\_S32 iParam1, JT\_S32 iParam2)

{

JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*ptData;

JTAPI\_Time\_t time;

ptData = (JTGUI\_SATELLITE\_SEARCH\_DATA\_t \*)WND\_GetData(hWnd);

if(NULL == ptData)

{

JT\_ERROR((">>>#error: %d, %s: WND\_GetData Failed.\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

}

switch (sMsg)

{

case WM\_CREATE:

{

JTGUI\_Init\_Data(ptData);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case WM\_SHOW:

{

JTGUI\_WND\_Bg();

JTGUI\_DrawUI(ptData);

JTGUIOSD\_RefreshRectangle( SATELLITE\_SEARCH\_WND\_X, SATELLITE\_SEARCH\_WND\_Y, SATELLITE\_SEARCH\_WND\_W, SATELLITE\_SEARCH\_WND\_H);

break;

}

case WM\_TIMER:

{

if(ptData->u32Timer != INVALID\_TIMER && ptData->u32Timer == iParam1)

{ JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_L\_PNG,ITEM\_ARROW\_LEFT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIPNG\_DrawPngEx(PATH\_ARROW\_GREY\_R\_PNG,ITEM\_ARROW\_RIGHT\_X,ITEM\_ARROW\_Y,JTGUI\_NO\_OFFSET,JTGUI\_NO\_OFFSET);

JTGUIOSD\_RefreshRectangle(ITEM\_ARROW\_LEFT\_X,ITEM\_ARROW\_Y,ITEM\_ARROW\_W,ITEM\_ARROW\_H);

TIMER\_KILL\_SAFE(ptData->u32Timer);

}

break;

}

case WM\_KEYDOWN:

{

switch (sKey)

{

case KEY\_OK:

{

JT\_S16 s16Select;

JT\_U8 au8Index[5] = {ptData->u16SatelliteIndex, ptData->u8FTAIndex, ptData->u8ScanChannelIndex, ptData->u8NetworkSearchIndex, ptData->u8ScanModeIndex};

JT\_U8 au8TotalItems[5] = {ptData->u16SatelliteCount, 2, 3, 2, 2};

switch(ptData->enMenuFocusIndex)

{

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE:

case JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA:

case JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE:

case JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH:

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE:

{

s16Select = JTGUI\_DrawListBox\_(hWnd,ITEM\_POP\_BG\_X, ITEM\_POP\_BG\_Y, ITEM\_POP\_BG\_W, ITEM\_POP\_BG\_H,\

au8Index[ptData->enMenuFocusIndex], au8TotalItems[ptData->enMenuFocusIndex], JTGUI\_ITEM\_HEIGHT, "/theme/Background\_items.png", fnItemDrawFunc);

break;

}

}

if(s16Select != -1)

{

switch(ptData->enMenuFocusIndex)

{

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE:

{

ptData->u16SatelliteIndex = s16Select;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA:

{

ptData->u8FTAIndex = s16Select;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE:

{

ptData->u8ScanChannelIndex = s16Select;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH:

{

ptData->u8NetworkSearchIndex = s16Select;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE:

{

ptData->u8ScanModeIndex = s16Select;

break;

}

}

}

break;

}

case KEY\_DOWN:

{

JTGUI\_ChangeItem(ptData, 1);

JTGUI\_DrawUI(ptData);

JTGUIOSD\_RefreshRectangle( SATELLITE\_SEARCH\_WND\_X, SATELLITE\_SEARCH\_WND\_Y, SATELLITE\_SEARCH\_WND\_W, SATELLITE\_SEARCH\_WND\_H);

//WND\_Show(hWnd, SW\_SHOW);

break;

}

case KEY\_UP:

{

JTGUI\_ChangeItem(ptData, -1);

JTGUI\_DrawUI(ptData);

JTGUIOSD\_RefreshRectangle( SATELLITE\_SEARCH\_WND\_X, SATELLITE\_SEARCH\_WND\_Y, SATELLITE\_SEARCH\_WND\_W, SATELLITE\_SEARCH\_WND\_H);

//WND\_Show(hWnd, SW\_SHOW);

break;

}

case KEY\_LEFT:

case KEY\_RIGHT:

{

g\_arrow\_left = (KEY\_LEFT == sKey ?JT\_TRUE:JT\_FALSE);

g\_arrow\_right = (KEY\_RIGHT == sKey ?JT\_TRUE:JT\_FALSE);

switch(ptData->enMenuFocusIndex)

{

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SATELLITE:

{

if(sKey == KEY\_LEFT)

{

ptData->u16SatelliteIndex = (ptData->u16SatelliteIndex == 0) ? (ptData->u16SatelliteCount - 1) : (ptData->u16SatelliteIndex - 1);

}

else

{

ptData->u16SatelliteIndex = (ptData->u16SatelliteIndex == ptData->u16SatelliteCount - 1) ? 0 : (ptData->u16SatelliteIndex + 1);

}

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_FTA:

{

ptData->u8FTAIndex = !ptData->u8FTAIndex;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_CHANNELTYPE:

{

ptData->u8ScanChannelIndex = (ptData->u8ScanChannelIndex == 0) ? 2 : (ptData->u8ScanChannelIndex - 1);

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_NETWORKSEARCH:

{

ptData->u8NetworkSearchIndex = !ptData->u8NetworkSearchIndex;

break;

}

case JTGUI\_SATELLITE\_SEARCH\_MENU\_SCANMODE:

{

ptData->u8ScanModeIndex = !ptData->u8ScanModeIndex;

break;

}

}

JTGUI\_DrawUI(ptData);

JTGUIOSD\_RefreshRectangle( SATELLITE\_SEARCH\_WND\_X, SATELLITE\_SEARCH\_WND\_Y, SATELLITE\_SEARCH\_WND\_W, SATELLITE\_SEARCH\_WND\_H);

//WND\_Show(hWnd, SW\_SHOW);

break;

}

#if defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

case KEY\_EPG:

#else

case KEY\_YELLOW:

#endif

case KEY\_GREEN:

{

JT\_StatusCode\_t status\_code;

JTAPISEARCH\_SearchParameter\_t srch\_param;

JTAPIDM\_SatInfo\_t sat\_info;

memset((void \*)&srch\_param, 0x00, sizeof(srch\_param));

srch\_param.search\_mode = SEARCH\_DVB\_S;

switch(ptData->u8ScanModeIndex)

{

case 0:

{

#if defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

if(sKey == KEY\_EPG)

#else

if(sKey == KEY\_YELLOW)

#endif

{

srch\_param.search\_type = SEARCH\_SINGLE\_BLIND;

}

else

{

srch\_param.search\_type = SEARCH\_MULTI\_BLIND;

}

break;

}

case 1:

{

#if defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

if(sKey == KEY\_EPG)

#else

if(sKey == KEY\_YELLOW)

#endif

{

srch\_param.search\_type = SEARCH\_AUTO;

}

else

{

srch\_param.search\_type = SEARCH\_MULTI\_AUTO;

}

break;

}

}

srch\_param.search\_filter.sat\_filter.FTA = ptData->u8FTAIndex;

srch\_param.search\_filter.sat\_filter.search\_channel = ptData->u8ScanChannelIndex;

srch\_param.connect\_info.delivery\_type = JT\_DMD\_DELIVERY\_SATELLITE;

status\_code = JTAPIDM\_SatGetById((JT\_U32)ptData->au16SatID[ptData->u16SatelliteIndex], &sat\_info);

if(status\_code != JT\_SUCCESS)

{

JT\_ERROR(("JTAPIDM\_SatGetById Faild\n"));

break;

}

srch\_param.connect\_info.stream\_port = sat\_info.lnb\_port;

srch\_param.connect\_info.delivery.satellite.lnb\_freq = (JTHDIDMD\_LnbFreq\_t)sat\_info.lnb\_freq;

srch\_param.sat\_id = sat\_info.id;

JTGUI\_SearchResult(hWnd, &srch\_param);

break;

}

case KEY\_BACK:

{

WND\_PostMessage(hWnd, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_TERM, WND\_COMMAND\_TERM\_NORMAL\_MODE);

break;

}

}

}

case WM\_COMMAND:

{

switch(iParam1)

{

case WND\_COMMAND\_TERM:

{

WND\_Destroy(hWnd);

g\_hSatellite\_Search = NULL;

#if defined(JT\_MENU\_KEEP\_VIDEO) && !defined(JT\_CUSTOMER\_MALDIVES)

JT\_U32 u32ChID = 0;

JT\_StatusCode\_t statusCode;

statusCode = JTAPIDM\_ServiceInfoListGetChid(g\_groupIndex, g\_serviceIndexInGroup, &u32ChID);

statusCode = JTAPIPLAY\_Open(u32ChID, AV\_DECODER\_MAIN);

JT\_ASSERT(JT\_SUCCESS == statusCode);

#endif

break;

}

}

break;

}

default:

break;

}

return MAINWND\_DefProc(hWnd, sMsg, sKey, iParam1, iParam2);

}

static JT\_BOOL JTGUI\_CheckSatCount(HWND hWnd)

{

JT\_StatusCode\_t statusCode;

JT\_U16 \*pSatID = NULL;

JT\_U32 lnb0\_cnt, lnb1\_cnt;

JTGUI\_BACKGROUND\_t stBg;

BITMAPTRUECOLOR \*pPopupBg;

pSatID = (JT\_U16\*)JTOS\_MemMalloc(300 \* sizeof(pSatID));

if (pSatID == NULL)

{

SS\_DEBUG("No Memery.");

return JT\_FALSE;

}

statusCode = JTAPIDM\_SatGetList(JT\_DMD\_PORT\_0, pSatID, &lnb0\_cnt);

JT\_ASSERT(statusCode == JT\_SUCCESS);

statusCode = JTAPIDM\_SatGetList(JT\_DMD\_PORT\_1, pSatID, &lnb1\_cnt);

JT\_ASSERT(statusCode == JT\_SUCCESS);

if (lnb0\_cnt == 0 && lnb1\_cnt == 0)

{

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

pPopupBg = &stBg.unBgContent.stBitMap;

JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_OK, JTGUI\_POPUP\_LIST\_X, JTGUI\_POPUP\_LIST\_Y, JTGUI\_POPUP\_LIST\_W, JTGUI\_POPUP\_LIST\_H, \

&stBg, msg\_information[stGlobalVar.enLanguage], 33, msg\_no\_satellite\_selected[stGlobalVar.enLanguage], 90, 98);

JTOS\_MemFree(pSatID);

return JT\_FALSE;

}

JTOS\_MemFree(pSatID);

return JT\_TRUE;

}

JT\_StatusCode\_t JTGUI\_SATELLITE\_SEARCH\_Setup(HWND hParentWnd)

{

RECT tRect;

JT\_BOOL bRet;

JTGUI\_Check\_Handle(hParentWnd);

bRet = JTGUI\_CheckSatCount(hParentWnd);

if(bRet == JT\_FALSE)

{

return JT\_SUCCESS;

}

tRect.X = SATELLITE\_SEARCH\_WND\_X;

tRect.Y = SATELLITE\_SEARCH\_WND\_Y;

tRect.W = SATELLITE\_SEARCH\_WND\_W;

tRect.H = SATELLITE\_SEARCH\_WND\_H;

g\_hSatellite\_Search = WND\_Create(SATELLITE\_SEARCH\_NAME, &tRect, sizeof(JTGUI\_SATELLITE\_SEARCH\_DATA\_t), JTGUI\_SATELLITE\_SEARCH\_Pro, hParentWnd);

if(g\_hSatellite\_Search == NULL)

{

JT\_DEBUG(("error : Satellite Search Created Failed!!! #%d\n", \_\_LINE\_\_));

return JT\_ERROR\_INVALID\_HANDLE;

}

WND\_SetFocus(g\_hSatellite\_Search);

return JT\_SUCCESS;

}

#include "ui\_com.h"

#include "tplist.h"

#include "api\_dm\_tp.h"

#include "api\_dm\_sat.h"

#include "api\_dm\_syscfg.h"

#include "hdi\_dmd.h"

#include "hdi\_av.h"

#include "mainwnd.h"

#ifdef JT\_CUSTOMER\_MALDIVES

#define POPUPBOX\_STYLE\_FOR\_MEDIANET

#endif

/\*==============================MACRO DEFINE===========================\*/

#define TPLIST\_WND\_NAME "TPLIST"

#define TPLIST\_WND\_X 0

#define TPLIST\_WND\_Y 0

#define TPLIST\_WND\_W WINDOW\_MAX\_WIDTH

#define TPLIST\_WND\_H WINDOW\_MAX\_HEIGHT

//list box

#define TPLIST\_LIST\_W (TPLIST\_WND\_W \* 3 / 5)

#define TPLIST\_LIST\_H (TPLIST\_WND\_H \* 3 / 5 - 20)

#define TPLIST\_LIST\_X (TPLIST\_WND\_X + ((TPLIST\_WND\_W - TPLIST\_LIST\_W) >> 1))

#define TPLIST\_LIST\_ROW 7

#define TPLIST\_ITEM\_H (TPLIST\_LIST\_H / TPLIST\_LIST\_ROW)

#define TPLIST\_ITEM\_X(index) TPLIST\_LIST\_X

#define TPLIST\_ITEM\_FILETYPE\_W (TPLIST\_LIST\_W \* 3 / 5)

#define TPLIST\_ITEM\_FILETYPE\_H (TPLIST\_ITEM\_H \* 3 / 4)

#define TPLIST\_ITEM\_ID\_W (80)

#define TPLIST\_ITEM\_ID\_H (TPLIST\_ITEM\_H \* 3 / 4)

#define TPLIST\_ITEM\_FREQ\_W (200)

#define TPLIST\_ITEM\_FREQ\_H (TPLIST\_ITEM\_H \* 3 / 4)

#define TPLIST\_ITEM\_SYMBOLRATE\_W (240)

#define TPLIST\_ITEM\_SYMBOLRATE\_H (TPLIST\_ITEM\_ID\_H)

#define TPLIST\_ITEM\_POLARITY\_W (40)

#define TPLIST\_ITEM\_POLARITY\_H (TPLIST\_ITEM\_SYMBOLRATE\_H)

#define TPLIST\_REFRESH\_QS\_VALUE\_X 870

#define TPLIST\_REFRESH\_QS\_VALUE\_Y 510

#define TPLIST\_REFRESH\_QS\_VALUE\_W (WINDOW\_MAX\_WIDTH - TPLIST\_REFRESH\_QS\_VALUE\_X)

#define TPLIST\_REFRESH\_QS\_VALUE\_H 100

#define TPLIST\_ITEM\_PROGRESSBAR\_X 470

#define TPLIST\_ITEM\_PROGRESSBAR\_Y 535

#define TPLIST\_ITEM\_PROGRESSBAR\_W 14

#define TPLIST\_ITEM\_PROGRESSBAR\_H 14

#define TPLIST\_ITEM\_PROGRESSBAR\_INTERVAL\_Y 40

#define TPLIST\_SIGNAL\_QS\_VALUE\_X 870

#define TPLIST\_ITEM\_SIGNAL\_QUALITY\_Y 565

#define TPLIST\_ITEM\_SIGNAL\_QS\_X 232

#define TPLIST\_ITEM\_SIGNAL\_STRENGTH\_Y 525

#define TPLIST\_SIGNALBAR\_REFRESH\_X 202

#define TPLIST\_SIGNALBAR\_REFRESH\_Y 525

#define TPLIST\_SIGNALBAR\_REFRESH\_W (WINDOW\_MAX\_WIDTH - TPLIST\_SIGNALBAR\_REFRESH\_X)

#define TPLIST\_SIGNALBAR\_REFRESH\_H 180

#define TPLIST\_BUTTON\_X(index) 90+(index)\*170

#define TPLIST\_BUTTON\_Y(index) 625

#define TPLIST\_BUTTON\_INFO\_X(index) 135+(index)\*170

#define TPLIST\_BUTTON\_INFO\_Y(index) 635

#define TPLIST\_ITEM\_INDEX\_X 232

#define TPLIST\_ITEM\_FREQUECY\_X(index) 420

#define TPLIST\_ITEM\_SYMBOLRATE\_X(index) 750

#define TPLIST\_ITEM\_POLARITY\_X(index) 1020

#define TPLIST\_ITEM\_Y(index) (index%2 ==0) ? (275-(index/2-1)\*45) : (385+((index+1)/2-1)\*45)

#define TPLIST\_ITEM\_MID\_Y 323 //278

#define ARROW\_LEFT\_X 350

#define ARROW\_RIGHT\_X 850

#define ARROW\_WIDTH 60

#define ARROW\_HEIGHT 40

#define SATELLITENAME\_X 330

#define SATELLITENAME\_Y (JTGUI\_SUBWND\_TITLE\_Y+50)

#define SATELLITENAME\_W (WINDOW\_MAX\_WIDTH - 2\*SATELLITENAME\_X)

#define SATELLITENAME\_H 50

#define JT\_Check\_Status(x) \

{ \

if ((x) != JT\_SUCCESS) \

{ \

printf("@@@----Line : %d %s Failed Error No.%d-----------------------\n", \_\_LINE\_\_, \_\_FUNCTION\_\_); \

} \

}while(0)

#define SAT\_ID\_NUM\_MAX 300

#define TP\_ID\_NUM\_MAX 300

#define ASCEND 1

#define DESCEND -1

#define SIGNAL\_TIMER\_OUT 2000

typedef struct \_Tplist

{

JT\_BOOL bChange;

JT\_BOOL bFreshListBg;

JT\_BOOL bFind;

JT\_S32 s32Strength;

JT\_S32 s32Quality;

JT\_S32 fSQTimer;

JT\_S32 s32SatCnt;

JT\_S32 s32TpCnt;

JT\_S32 S32ListCount;

JT\_S32 s32CurIndex;

JT\_S32 s32SatIndex;

JT\_S32 s32SortOrder;

JT\_U32 u32PreConnectFreq;

JT\_U32 u32PreConnectSR;

JT\_U32 u32PreConnectPloar;

JT\_U32 u32PreConnectLnbFreq;

HSCROLL hScroll;

JT\_U16\* pSatIdArray;

JT\_U16\* pTpIdArray;

JT\_S32 u32Timer;

}JTGUI\_TPLIST\_DATA\_t;

/\*====================================================\*/

static HWND g\_hTplist = NULL;

static JT\_U8 g\_arrow\_left = JT\_FALSE;

static JT\_U8 g\_arrow\_right = JT\_FALSE;

static JT\_U8 \*g\_pu8Info[][LANGUAGE\_NUMBER\_MAX] =

{

JTGUI\_FILLTITLE(msg\_switch\_sat),

JTGUI\_FILLTITLE(msg\_search),

JTGUI\_FILLTITLE(msg\_edit),

JTGUI\_FILLTITLE(msg\_delete\_full),

JTGUI\_FILLTITLE(msg\_add)

};

/\*==================================================\*/

static JT\_StatusCode\_t JTGUI\_DrawTPListBg()

{

//JTGUIBMP\_DisplayPartOfBitmap(TPLIST\_ITEM\_INDEX\_X, TPLIST\_ITEM\_Y(2\*3), JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80,TPLIST\_ITEM\_INDEX\_X,TPLIST\_ITEM\_Y(2\*3), &ui\_BackgroudIfor, JT\_TRUE);

JTGUI\_Draw\_Part\_of\_PNG(PATH\_BACKGROUND\_LEFT\_PNG,TPLIST\_ITEM\_INDEX\_X, TPLIST\_ITEM\_Y(2\*3), JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80,TPLIST\_ITEM\_INDEX\_X - PATH\_CUPID\_MENU3\_PNG\_X, TPLIST\_ITEM\_Y(2\*3));

return JT\_SUCCESS;

}

static JT\_StatusCode\_t JTGUI\_SatData\_Init(JTGUI\_TPLIST\_DATA\_t \*pstData)

{

JT\_U32 u32SatCnt;

JT\_U32 i = 0;

JT\_U32 j = 0;

JT\_U16\* pSatIdTempArray = NULL;

JT\_U16 invisibleSatCnt = 0;

pSatIdTempArray = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* SAT\_ID\_NUM\_MAX);

if(NULL == pSatIdTempArray)

{

JT\_ERROR(("#error: %d, %s: JTOS\_MemMalloc Failed!!!\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return JT\_FAILED;

}

pstData->pSatIdArray = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* SAT\_ID\_NUM\_MAX);

if(NULL == pstData->pSatIdArray)

{

JT\_ERROR(("#error: %d, %s: JTOS\_MemMalloc Failed!!!\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return JT\_FAILED;

}

memset(pSatIdTempArray, 0, sizeof(JT\_U16) \* SAT\_ID\_NUM\_MAX);

memset(pstData->pSatIdArray, 0, sizeof(JT\_U16) \* SAT\_ID\_NUM\_MAX);

JT\_Check\_Status(JTAPIDM\_SatGetAllPortList(pSatIdTempArray, &u32SatCnt));

for(i = 0, j = 0; i < u32SatCnt, j < u32SatCnt; i++)

{

if (pSatIdTempArray[i] < SAT\_ID\_ISDB\_T)

{

pstData->pSatIdArray[j] = pSatIdTempArray[i];

j++;

}

else

{

invisibleSatCnt++;

}

}

pstData->s32SatCnt = u32SatCnt - invisibleSatCnt;

JTOS\_MemFree(pSatIdTempArray);

pSatIdTempArray = NULL;

return JT\_SUCCESS;

}

static JT\_StatusCode\_t JTGUI\_TpData\_Init(JTGUI\_TPLIST\_DATA\_t \*pstData)

{

JT\_U32 i = 0, count = 0;

JTAPIDM\_SatInfo\_t satInfo;

JTAPIDM\_TpInfo\_t\* pTpInfoArray = NULL;

pTpInfoArray = (JTAPIDM\_TpInfo\_t\*)JTOS\_MemMalloc(sizeof(JTAPIDM\_TpInfo\_t) \* SAT\_ID\_NUM\_MAX);

if(NULL != pstData->pTpIdArray)

{

JTOS\_MemFree(pstData->pTpIdArray);

pstData->pTpIdArray = NULL;

}

if(pstData->pTpIdArray == NULL)

{

pstData->pTpIdArray = (JT\_U16\*)JTOS\_MemMalloc(sizeof(JT\_U16) \* TP\_ID\_NUM\_MAX);

}

if(NULL == pTpInfoArray || NULL == pstData->pTpIdArray)

{

JT\_ERROR(("#error: %d, %s: JTOS\_MemMalloc Failed!!!\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

return JT\_FAILED;

}

memset(pTpInfoArray, 0, sizeof(JTAPIDM\_TpInfo\_t) \* SAT\_ID\_NUM\_MAX);

//display only tplist for sat lnb being selected

if(pstData->s32SortOrder == ASCEND)

{

for(i = pstData->s32SatIndex, count = 0; ((i + pstData->s32SatCnt) % pstData->s32SatCnt), count < pstData->s32SatCnt; i++, count++)

{

JT\_Check\_Status(JTAPIDM\_SatGetById(pstData->pSatIdArray[i], &satInfo));

if( (satInfo.lnb\_port == 0) && (pstData->s32SatIndex == i))

{

pstData->bFind = JT\_TRUE;

break;

}

else if((satInfo.lnb\_port != 0) && (pstData->s32SatIndex == i))

{

pstData->s32SatIndex = (pstData->s32SatIndex + pstData->s32SatCnt + 1) % pstData->s32SatCnt; //next

}

}

}

else

{

for(i = pstData->s32SatIndex, count = 0; ((i + pstData->s32SatCnt) % pstData->s32SatCnt), count < pstData->s32SatCnt; i--, count++)

{

JT\_Check\_Status(JTAPIDM\_SatGetById(pstData->pSatIdArray[i], &satInfo));

if( (satInfo.lnb\_port == 0) && (pstData->s32SatIndex == i))

{

pstData->bFind = JT\_TRUE;

break;

}

else if((satInfo.lnb\_port != 0) && (pstData->s32SatIndex == i))

{

pstData->s32SatIndex = (pstData->s32SatIndex + pstData->s32SatCnt - 1) % pstData->s32SatCnt;

}

}

}

if(pstData->bFind != JT\_TRUE)

{

pstData->s32TpCnt = 0;

pstData->S32ListCount = 0;

JTOS\_MemFree(pTpInfoArray);

pTpInfoArray = NULL;

return JT\_SUCCESS;

}

JT\_Check\_Status(JTAPIDM\_TpGetTpArrayBySatId(pstData->pSatIdArray[pstData->s32SatIndex], pTpInfoArray , &pstData->s32TpCnt));

pstData->S32ListCount = pstData->s32TpCnt;

pstData->bFind = JT\_FALSE;

for(i = 0; i < pstData->s32TpCnt; i++)

{

pstData->pTpIdArray[i] = pTpInfoArray[i].id;

}

JTOS\_MemFree(pTpInfoArray);

pTpInfoArray = NULL;

return JT\_SUCCESS;

}

static JT\_S32 JTGUI\_TPLIST\_Pro(HWND hWnd, JT\_S16 sMsg, JT\_S16 sKey, JT\_S32 iParam1, JT\_S32 iParam2)

{

JT\_S32 s32CurIndex;

JTGUI\_BACKGROUND\_t stBg;

BITMAPTRUECOLOR \*pPopupBg;

JTGUI\_TPLIST\_DATA\_t \*pstData = NULL;

pstData = WND\_GetData(g\_hTplist);

if(NULL == pstData)

{

JT\_ERROR(("#ERROR: %d, %s, WND\_GetData Failed.\n", \_\_LINE\_\_, \_\_FUNCTION\_\_));

exit(1);

}

s32CurIndex = pstData->s32CurIndex;

switch(sMsg)

{

case WM\_CREATE:

{

JT\_Check\_Status(JTGUI\_LIST\_InitData(pstData));

JT\_Check\_Status(JTGUI\_SatData\_Init(pstData));

JT\_Check\_Status(JTGUI\_TpData\_Init(pstData));

JT\_Check\_Status(JTGUI\_LIST\_DrawContent(pstData));

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_REPEAT, SIGNAL\_TIMER\_OUT);

WND\_Show(hWnd, SW\_SHOW);

break;

}

case WM\_SHOW:

{

JTGUI\_DrawWndBg(pstData);

JTGUI\_DrawSatName(pstData);

if(pstData->bChange == JT\_TRUE)

{

//a new tp has been added.

pstData->bChange = JT\_FALSE;

//reinitialization

SCROLL\_Term(pstData->hScroll);

JT\_Check\_Status(JTGUI\_TpData\_Init(pstData));

JT\_Check\_Status(JTGUI\_LIST\_DrawContent(pstData));

if( 0 != pstData->s32CurIndex)

{

SCROLL\_SetCurFocusItem(pstData->hScroll, (pstData->s32CurIndex)++);

}

else

{

SCROLL\_SetCurFocusItem(pstData->hScroll, pstData->s32CurIndex);

}

}

//JT\_Check\_Status(JTGUI\_LIST\_DrawContent(pstData));

SCROLL\_StartNavi(pstData->hScroll);

JTGUI\_DrawProcessQS(pstData->s32Strength, pstData->s32Quality);

JTHDIOSD\_RefreshScreen(TPLIST\_WND\_X, TPLIST\_WND\_Y, TPLIST\_WND\_W, TPLIST\_WND\_H);

}

break;

case WM\_TIMER:

{

if((pstData->fSQTimer != INVALID\_TIMER) && (pstData->fSQTimer == iParam1))

{

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_UpdateSignal(pstData);

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_REPEAT, SIGNAL\_TIMER\_OUT);

}

if((pstData->u32Timer != INVALID\_TIMER) && (pstData->u32Timer == iParam1))

{

//JTGUIPNG\_DrawPngEx("/theme/satellite\_grey\_L.png", ARROW\_LEFT\_X, JTGUI\_SUBWND\_TITLE\_Y+50, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

//JTGUIPNG\_DrawPngEx("/theme/satellite\_grey\_R.png", ARROW\_RIGHT\_X, JTGUI\_SUBWND\_TITLE\_Y+50, JTGUI\_NO\_OFFSET, JTGUI\_NO\_OFFSET);

//JTGUIOSD\_RefreshRectangle(ARROW\_LEFT\_X,JTGUI\_SUBWND\_TITLE\_Y+50,(ARROW\_RIGHT\_X-ARROW\_LEFT\_X+ARROW\_WIDTH),JTGUI\_SUBWND\_TITLE\_H);

TIMER\_KILL\_SAFE(pstData->u32Timer);

}

}

break;

case WM\_KEYDOWN:

{

switch (sKey)

{

case KEY\_BACK:

{

TIMER\_KILL\_SAFE(pstData->fSQTimer);

SCROLL\_Term(pstData->hScroll);

WND\_PostMessage(hWnd, WM\_COMMAND, KEY\_DUMMY, WND\_COMMAND\_TERM, WND\_COMMAND\_TERM\_NORMAL\_MODE);

break;

}

case KEY\_UP:

{

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_DrawTPListBg();

SCROLL\_MoveUp(pstData->hScroll);

SCROLL\_StartNavi(pstData->hScroll);

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_REPEAT, SIGNAL\_TIMER\_OUT);

JTHDIOSD\_RefreshScreen(TPLIST\_ITEM\_INDEX\_X, JTGUI\_SUBWND\_TITLE\_Y+60, JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80);

break;

}

case KEY\_DOWN:

{

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_DrawTPListBg();

SCROLL\_MoveDown(pstData->hScroll);

SCROLL\_StartNavi(pstData->hScroll);

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_REPEAT, SIGNAL\_TIMER\_OUT);

JTHDIOSD\_RefreshScreen(TPLIST\_ITEM\_INDEX\_X, JTGUI\_SUBWND\_TITLE\_Y+60, JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80);

break;

}

case KEY\_LEFT:

case KEY\_RIGHT:

{

g\_arrow\_left = (KEY\_LEFT == sKey ?JT\_TRUE:JT\_FALSE);

g\_arrow\_right = (KEY\_RIGHT == sKey ?JT\_TRUE:JT\_FALSE);

TIMER\_KILL\_SAFE(pstData->fSQTimer);

//switch sat

if(pstData->s32SatCnt == 0)

{

break;

}

pstData->s32SatIndex = ( (sKey == KEY\_LEFT) ? (pstData->s32SatIndex + pstData->s32SatCnt - 1) % pstData->s32SatCnt : \

(pstData->s32SatIndex + pstData->s32SatCnt + 1) % pstData->s32SatCnt );

pstData->s32SortOrder = (sKey == KEY\_LEFT) ? DESCEND : ASCEND;

SCROLL\_Term(pstData->hScroll);

JT\_Check\_Status(JTGUI\_DrawSatName(pstData));

JT\_Check\_Status(JTGUI\_TpData\_Init(pstData));

JT\_Check\_Status(JTGUI\_LIST\_DrawContent(pstData));

JTGUI\_DrawTPListBg();

SCROLL\_StartNavi(pstData->hScroll);

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_REPEAT, SIGNAL\_TIMER\_OUT);

JTHDIOSD\_RefreshScreen(TPLIST\_ITEM\_INDEX\_X, JTGUI\_SUBWND\_TITLE\_Y+50, JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80);

//WND\_Show(hWnd, SW\_SHOW);

break;

}

#if defined(JT\_REMOTE\_TYPE\_RCU\_B\_01)

case KEY\_EPG:

#else

case KEY\_YELLOW:

#endif

{

//delete

stBg.stBgType = JTGUI\_BGTYPE\_PICTURE;

pPopupBg = &stBg.unBgContent.stBitMap;

#if defined(POPUPBOX\_STYLE\_FOR\_MEDIANET)

stBg.unBgContent.stBitMap = ui\_bg\_popup\_medium\_medianetIfor;

if (POPUP\_BUTTON\_YES == JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_YESNO, JTGUI\_POPUP\_X(pPopupBg), JTGUI\_POPUP\_Y(pPopupBg), \

JTGUI\_POPUP\_W(pPopupBg), JTGUI\_POPUP\_H(pPopupBg), &stBg, NULL, 5,msg\_are\_you\_sure\_to\_delete[stGlobalVar.enLanguage], 5, JTGUI\_POPUP\_TEXT\_OFFSET\_Y))

#else

//stBg.unBgContent.stBitMap = ui\_bg\_popup\_mediumIfor;

if (POPUP\_BUTTON\_YES == JTGUI\_DrawPopupBox2(hWnd, JTGUI\_POPUP\_BTN\_YESNO, JTGUI\_POPUP\_LIST\_X, JTGUI\_POPUP\_LIST\_Y-200, \

JTGUI\_POPUP\_LIST\_W, JTGUI\_POPUP\_LIST\_H, &stBg, msg\_information[stGlobalVar.enLanguage], 33,msg\_are\_you\_sure\_to\_delete[stGlobalVar.enLanguage], 90, 98))

#endif

{

pstData->bFreshListBg = JT\_TRUE;

//delete tp

JTGUI\_DrawTPListBg();

JT\_Check\_Status(JTAPIDM\_TpRemoveById(pstData->pTpIdArray[pstData->s32CurIndex]));

JT\_Check\_Status(JTAPIDM\_Save());

SCROLL\_Term(pstData->hScroll);

JT\_Check\_Status(JTGUI\_TpData\_Init(pstData));

JT\_Check\_Status(JTGUI\_LIST\_DrawContent(pstData));

if( 0 != pstData->s32CurIndex)

{

SCROLL\_SetCurFocusItem(pstData->hScroll, --(pstData->s32CurIndex));

}

else

{

SCROLL\_SetCurFocusItem(pstData->hScroll, pstData->s32CurIndex);

}

SCROLL\_StartNavi(pstData->hScroll);

JTHDIOSD\_RefreshScreen(TPLIST\_ITEM\_INDEX\_X, JTGUI\_SUBWND\_TITLE\_Y+60, JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_W,JTGUI\_SUBWINDOW\_CHANGED\_ZONE\_H-80);

// WND\_Show(hWnd, SW\_SHOW);

}

break;

}

case KEY\_RED:

{

//search

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_TPLISTSEARCH\_Setup(hWnd, pstData->pTpIdArray[pstData->s32CurIndex], pstData->pSatIdArray[pstData->s32SatIndex]);

break;

}

case KEY\_GREEN:

{

//edit

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_TPLISTEdit\_Setup(hWnd, pstData->pTpIdArray[pstData->s32CurIndex], pstData->pSatIdArray[pstData->s32SatIndex], &pstData->bChange, JTGUI\_TP\_EDIT);

break;

}

case KEY\_BLUE:

{

//add

TIMER\_KILL\_SAFE(pstData->fSQTimer);

JTGUI\_TPLISTEdit\_Setup(hWnd, pstData->pTpIdArray[pstData->s32CurIndex], pstData->pSatIdArray[pstData->s32SatIndex], &pstData->bChange, JTGUI\_TP\_ADD);

break;

}

}

break;

}

case WM\_COMMAND:

{

switch(iParam1)

{

case WND\_COMMAND\_TIMER\_NOTICE:

{

if(pstData->fSQTimer == INVALID\_TIMER)

{

TIMER\_SET\_SAFE(pstData->fSQTimer, hWnd, TIMER\_ONCE, SIGNAL\_TIMER\_OUT);

}

break;

}

case WND\_COMMAND\_TERM:

{

if(pstData->pTpIdArray != NULL)

{

JTOS\_MemFree(pstData->pTpIdArray);

pstData->pTpIdArray = NULL;

}

else if(pstData->pTpIdArray != NULL)

{

JTOS\_MemFree(pstData->pSatIdArray);

pstData->pSatIdArray = NULL;

}

else if(pstData->hScroll != NULL)

{

SCROLL\_Term(pstData->hScroll);

pstData->hScroll = NULL;

}

TIMER\_KILL\_SAFE(pstData->fSQTimer);

WND\_Destroy(hWnd);

g\_hTplist = NULL;

#if defined(JT\_MENU\_KEEP\_VIDEO) && !defined(JT\_CUSTOMER\_MALDIVES)

JT\_U32 u32ChID = 0;

JT\_StatusCode\_t statusCode;

statusCode = JTAPIDM\_ServiceInfoListGetChid(g\_groupIndex, g\_serviceIndexInGroup, &u32ChID);

statusCode = JTAPIPLAY\_Open(u32ChID, AV\_DECODER\_MAIN);

JT\_ASSERT(JT\_SUCCESS == statusCode);

#endif

break;

}

}

break;

}

}

return MAINWND\_DefProc(hWnd, sMsg, sKey, iParam1, iParam2);

}

JT\_StatusCode\_t JTGUI\_TPLIST\_Setup(HWND hParentWnd)

{

RECT tRect;

JTGUI\_Check\_Handler(hParentWnd);

tRect.X = TPLIST\_WND\_X;

tRect.Y = TPLIST\_WND\_Y;

tRect.W = TPLIST\_WND\_W;

tRect.H = TPLIST\_WND\_H;

//WM\_CREATE

g\_hTplist = WND\_Create(TPLIST\_WND\_NAME, &tRect, sizeof(JTGUI\_TPLIST\_DATA\_t), JTGUI\_TPLIST\_Pro, hParentWnd);

if(g\_hTplist == NULL)

{

JT\_ERROR(("Error: TPLIST Created in failure ! %d\n", \_\_LINE\_\_));

return JT\_ERROR\_INVALID\_HANDLE;

}

WND\_SetFocus(g\_hTplist);

return JT\_SUCCESS;

}

HWND JTGUI\_TPLIST\_GetHandle()

{

return g\_hTplist;

}