在CHDICFunc\_QAMChart类中

# void CHDICFunc\_QAMChart::OnBnClickedclrqamchart()【1-1】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

//m\_QAMChart.Series(0).Clear();

KillTimer(MYTIMER);

memset(QAMPackBuffer,0,32\*8478);

readQAM\_ID=0;

writeQAM\_ID=0;

}

# void CHDICFunc\_QAMChart::OnBnClickedDisqamchart()【1-2】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

#if 0

#if 0

FILE \*fpBinary;

CString ExeCurrentDir=L"D:\\1.ts";

LPCSTR pszNonUnicode;

USES\_CONVERSION;

pszNonUnicode = W2A( ExeCurrentDir.LockBuffer( ) );

ExeCurrentDir.UnlockBuffer( );

if ((fpBinary = fopen(pszNonUnicode, "rb")) == NULL)

{

AfxMessageBox(L"Error");

return;

}

else

{

fread(DatTmp,1,64\*1024,fpBinary);

}

#endif

#if 0

QAM\_TaskFlag=FALSE;

Sleep(500);

Enable\_Constellation();

Write\_Register(0x31,0x03);

Sleep(100);

Write\_Register(0x37,0xB4);

Sleep(100);

Write\_Register(0x34,0x81);

Sleep(100);

//Write\_Register(0xB4,0x37);

//Sleep(100);

memset(QAMDataBuffer,0,64\*1024);

QAM\_TaskFlag=TRUE;

//if(QAM\_thread==NULL){

QAM\_thread=AfxBeginThread(QAM\_ReceiveingThread,NULL,THREAD\_PRIORITY\_TIME\_CRITICAL);

// }

Sleep(500);

memcpy(DatTmp,QAMDataBuffer,64\*1024);

#endif

for(int i=0;i<32\*1024;i++){

if(CalByteValue(DatTmp[i]) !=0 && CalByteValue(DatTmp[i+1]) !=0){

m\_QAMChart.Series(0).AddXY(CalByteValue(DatTmp[i]),CalByteValue(DatTmp[i+1]),L"",0);

}

i=i+1;

}

QAM\_TaskFlag=FALSE;

#endif

DataFlag=1;

Enable\_Constellation();

// set Register

Write\_Register(0x31,0x03);

Write\_Register(0x34,0x81);

Write\_Register(0x37,0xB4);

Sleep(200);

// start display

DataFlag=1;

ResendTaskRun = TRUE;

ReceiveTaskRun = TRUE;

Sleep(1000);

SetTimer(MYTIMER,1000,NULL);

}

在CHDICFunc\_SNRChart类中

# void CHDICFunc\_SNRChart::OnBnClickedbtnStartdisplay()【2-1】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

if(isOn==0){

SetTimer(MYTIMER,500,NULL);

isOn=1;

}

}

# void CHDICFunc\_SNRChart::OnBnClickedbtnStopdisplay()【2-2】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

KillTimer(MYTIMER);

isOn=0;

}

在HDICToolBus.cpp中

# void CHDICToolBus::OnAcceleratorEscape()【3-1】消息command

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),TRUE));

}

# void CHDICToolBus::OnAcceleratorReturn()【3-2】消息command

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),FALSE));

}

# void CHDICToolBus::OnBnClickedButtonI2cRead()【3-3】消息BN\_CLICKED

{

UCHAR Address = m\_Edit\_I2CAddress.GetIntValue();

UCHAR Length = m\_Edit\_I2CRead.GetIntValue();

UCHAR Buffer[64];

UCHAR Temp=0;

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

if(Length>20)

{

STTBX\_Print((L"Warning!!,Length>20!"));

return;

}

else if(Length==0)

{

STTBX\_Print((L"Warning!!!,Read Length =0!!!"));

return;

}

STTBX\_Print((L"Read Address:0x%02x,Length:0x%02x!\n",m\_Edit\_I2CAddress.GetIntValue(),m\_Edit\_I2CRead.GetIntValue()));

Read\_I2C(Address,Length,Buffer);

m\_Edit\_I2CData.SetData(Buffer,Length,20);

for(Temp=0;Temp<Length;Temp++)

{

STTBX\_Print((L" 0x%02x !\n",Buffer[Temp]));

}

}

# void CHDICToolBus::OnBnClickedButtonI2cWrite()【3-4】消息BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

UCHAR Address = m\_Edit\_I2CAddress.GetIntValue();

UCHAR Buffer[64];

UCHAR Length = m\_Edit\_I2CData.m\_length;

UCHAR Temp=0;

if(Length>20)

{

STTBX\_Print((L"Warning!!,Length>20!"));

return;

}

else if(Length==0)

{

STTBX\_Print((L"Warning!!!,Write Length =0!!!"));

return;

}

m\_Edit\_I2CData.GetData(Buffer,Length);

Write\_I2C(Address,Length,Buffer);

STTBX\_Print((L"Write Address:0x%02x,Length:0x%02x!\n",m\_Edit\_I2CAddress.GetIntValue(),Length));

for(Temp=0;Temp<Length;Temp++)

{

STTBX\_Print((L" 0x%02x !\n",Buffer[Temp]));

}

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnAcceleratorEscape()【4-1】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),TRUE));

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnAcceleratorReturn()【4-2】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),FALSE));

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedButtonDemodRead()【4-3】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

Do\_ReadRegister = 1;

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedButtonDemodReset()【4-4】消息：BN\_CLICKED

{

STTBX\_Print((L"Demod Soft Reset!!"));

Do\_SoftReset = 1;

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedButtonDemodSignaldetect()【4-5】消息：BN\_CLICKED

{

Do\_SignalDetect = 1;

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedButtonDemodVideoplay()【4-6】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

CString VLCdir;

DataFlag=0; // for TS

Sleep(100);

Enable\_TS(); // open TS

if(Check\_FileExist(GetCurrentWorkingDir()+L"\\HDIC\\VideoLAN\\VLC\\vlc.exe"))

{

VLCdir=GetCurrentWorkingDir()+L"\\HDIC\\VideoLAN\\VLC\\vlc.exe udp://@:12910";

}

else if(Check\_FileExist(L"C:\\Program Files\\VideoLAN\\VLC\\vlc.exe"))

{

VLCdir=L"C:\\Program Files\\VideoLAN\\VLC\\vlc.exe udp://@:12910";

}

else if(Check\_FileExist(L"D:\\Program Files\\VideoLAN\\VLC\\vlc.exe"))

{

VLCdir=L"D:\\Program Files\\VideoLAN\\VLC\\vlc.exe udp://@:12910";

}

else

{

AfxMessageBox(L"Cannot find VLC Player");

}

//memset(USBBuffer,0,32\*64\*1024);

//ReadID=0;

//WriteID=0;

//Sleep(100);

STARTUPINFO siStartInfo={0};

siStartInfo.cb = sizeof(STARTUPINFO);

CreateProcess(NULL,VLCdir.GetBuffer(),NULL,NULL,FALSE,0,NULL,NULL,&siStartInfo,&m\_piProcInfo);

//----------------------------------------------------//

//Sleep(500);

if(thread1==NULL)

thread1=AfxBeginThread(ReceiveingThread,NULL,THREAD\_PRIORITY\_TIME\_CRITICAL);

//if(thread2==NULL)

// thread2=AfxBeginThread(ResendThread,NULL,THREAD\_PRIORITY\_TIME\_CRITICAL);

ResendTaskRun = TRUE;

ReceiveTaskRun = TRUE;

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedButtonDemodWrite()【4-7】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

Do\_WriteRegister = 1;

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnBnClickedSavets()【4-8】消息：BN\_CLICKED

{

if(SaveToFile==0)

{

CFileDialog dlg(FALSE,NULL,NULL,OFN\_HIDEREADONLY|OFN\_OVERWRITEPROMPT,

L".ts(\*.ts)|\*.ts| |",AfxGetMainWnd());

if(dlg.DoModal()==IDOK)

{

m\_strPath=dlg.GetPathName()+L".ts";

fileWriteEx.Open(m\_strPath,CFile::modeCreate | CFile::modeWrite | CFile::typeBinary | CFile::modeNoTruncate);

SaveToFile = 1;

}

else

{

m\_SaveTS.SetCheck(FALSE);

}

}

else

{

SaveToFile = 0;

Sleep(10);

fileWriteEx.Close();

}

}

在CHDICToolDemod类中

# void CHDICToolDemod::OnCbnSelchangeComboDemodAutodetect()【4-9】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

STTBX\_Print((L"222"));

if(m\_Combo\_DemodAutoDetect.GetCurSel())

{

m\_Combo\_DemodCodeRate.EnableWindow();

m\_Combo\_DemodConstellation.EnableWindow();

m\_Combo\_DemodCarrierMode.EnableWindow();

m\_Combo\_DemodPNMode.EnableWindow();

m\_Combo\_DemodInterleave.EnableWindow();

if(Selected\_Demod\_Type==HDIC\_DEMOD\_2311)

{

m\_Combo\_DemodPnVariable.EnableWindow();

}

}

else

{

m\_Combo\_DemodCodeRate.EnableWindow(FALSE);

m\_Combo\_DemodConstellation.EnableWindow(FALSE);

m\_Combo\_DemodCarrierMode.EnableWindow(FALSE);

m\_Combo\_DemodPNMode.EnableWindow(FALSE);

m\_Combo\_DemodInterleave.EnableWindow(FALSE);

m\_Combo\_DemodPnVariable.EnableWindow(FALSE);

}

}

在CHDICToolDlg类中

# void CHDICToolDlg::OnAcceleratorEscape()【5-1】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),TRUE));

}

# void CHDICToolDlg::OnAcceleratorReturn()【5-2】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),FALSE));

}

# void CHDICToolDlg::OnBnClickedAutoCheck()【5-3】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

if(m\_MonitorThread==NULL)

{

m\_MonitorThread=AfxBeginThread(MonitorThread,NULL,THREAD\_PRIORITY\_TIME\_CRITICAL);

}

if(HDIC\_GetDemodRunFlag()==TRUE)

{

Monitor\_Enable = FALSE;

m\_CButton\_Monitor.SetCheck(FALSE);

return; //do nothing when demod is run.

}

if(Monitor\_Enable==FALSE)

{

Monitor\_Enable = TRUE;

m\_CButton\_Monitor.SetCheck(TRUE);

}

else

{

Monitor\_Enable = FALSE;

m\_CButton\_Monitor.SetCheck(FALSE);

}

}

# void CHDICToolDlg::OnBnClickedAutoCheck2()【5-4】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

if(ADD\_TOLOG==FALSE)

{

ADD\_TOLOG = TRUE;

m\_Check\_AddToLog.SetCheck(TRUE);

}

else

{

ADD\_TOLOG = FALSE;

m\_Check\_AddToLog.SetCheck(FALSE);

}

}

# void CHDICToolDlg::OnBnClickedButtonFilesave()【5-5】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

CString Cstr,Cname;

if(!m\_SaveLogToFile)

{

CFileDialog fDlg(TRUE,NULL,NULL ,OFN\_HIDEREADONLY|OFN\_OVERWRITEPROMPT,L"LOG Files (\*.log)|\*.log||",this);

INT\_PTR nResponse = fDlg.DoModal();

if (nResponse == IDOK)

{

m\_SaveLogToFile = TRUE;

Cstr.LoadStringW(IDS\_STRING\_SAVEFILESTOP);

Cname = fDlg.GetPathName();

if(!m\_Logfile.Open((LPCTSTR)Cname, CFile::modeCreate | CFile::modeWrite| CFile::modeNoTruncate))

{

HDIC\_Print(L"Can not open file %s!",Cname);

m\_SaveLogToFile = FALSE;

Cstr.LoadStringW(IDS\_STRING\_SAVEFILESTART);

m\_Static\_FileName.SetWindowText(L"");

return;

}

m\_Logfile.SeekToEnd();

m\_Static\_FileName.SetWindowText((LPCTSTR)Cname);

}

else if (nResponse == IDCANCEL)

{

Cstr.LoadStringW(IDS\_STRING\_SAVEFILESTART);

m\_Static\_FileName.SetWindowText(L"");

}

}

else

{

m\_SaveLogToFile=FALSE;

m\_Logfile.Close();

Cstr.LoadStringW(IDS\_STRING\_SAVEFILESTART);

m\_Static\_FileName.SetWindowText(L"");

}

m\_Button\_FileSave.SetWindowText((LPCTSTR)Cstr);

}

# void CHDICToolDlg::OnMenuChannel()【5-6】消息：COMMAND

{

if(!m\_Func\_Channel->bFuncChannelExist){

m\_Func\_Channel->Create(IDD\_Func\_Channel,GetDesktopWindow());

m\_Func\_Channel->ShowWindow(SW\_SHOW);

}

}

# void CHDICToolDlg::OnMenuExit()【5-7】消息：COMMAND

{

DestroyWindow();

CDialog::OnClose();

}

# void CHDICToolDlg::OnMenuQAM()【5-8】消息：COMMAND

{

if(!m\_Func\_QAMChart->bFuncQAMChartExist){

m\_Func\_QAMChart->Create(IDD\_Func\_QAMChart,GetDesktopWindow());

m\_Func\_QAMChart->ShowWindow(SW\_SHOW);

}

}

# void CHDICToolDlg::OnMenuRegWindow()【5-9】消息：COMMAND

{

if(!m\_RegWindow->bRegWindowExist){

m\_RegWindow->Create(IDD\_Func\_RegWindow,GetDesktopWindow());

m\_RegWindow->ShowWindow(SW\_SHOW);

}

CRect rc,rc2;

::GetWindowRect(this->m\_hWnd,&rc);

::GetWindowRect(this->m\_hWnd,&rc2);

//::MoveWindow(this->m\_hWnd,rc.bottom,rc.bottom,rc2.Width(),rc2.Height(),TRUE);

}

# void CHDICToolDlg::OnMenuSpectrum()【5-10】消息：COMMAND

{

if(!m\_Func\_SNRChart->bFuncSNRChartExist){

m\_Func\_SNRChart->Create(IDD\_Func\_SNRChart,GetDesktopWindow());

m\_Func\_SNRChart->ShowWindow(SW\_SHOW);

}

}

# void CHDICToolDlg::OnTcnSelchangeTab(NMHDR \*pNMHDR, LRESULT \*pResult) 【5-11】消息：COMMAND

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

\*pResult = 0;

CWnd \* pPage;

int iIndex;

iIndex=m\_Tab.GetCurSel();

if(iIndex >=0 && iIndex < m\_oPages.GetCount())

{

pPage = m\_oPages.GetAt(iIndex);

CRect oRect,oWRect,oPWRect,oIRect;

m\_Tab.GetItemRect(0,oIRect);

m\_Tab.GetClientRect(oRect);

m\_Tab.GetWindowRect(oWRect);

GetWindowRect(oPWRect);

pPage->SetWindowPos(&m\_Tab,oWRect.left-oPWRect.left-1,oWRect.top-oPWRect.top-6,oRect.Width()-6,oRect.Height()-oIRect.Height()-8,SWP\_SHOWWINDOW);

pPage->ShowWindow(SW\_SHOW);

for(int i=0;i<m\_oPages.GetCount();i++)

{

if(i!=iIndex)

{

pPage = m\_oPages.GetAt(i);

pPage->ShowWindow(SW\_HIDE);

}

}

}

else

{

for(int i=0;i<m\_oPages.GetCount();i++)

{

pPage = m\_oPages.GetAt(i);

pPage->ShowWindow(SW\_HIDE);

}

}

}

在类CHDICToolSetting中

# void CHDICToolSetting::OnAcceleratorEscape()【6-1】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),TRUE));

}

# void CHDICToolSetting::OnAcceleratorReturn()【6-2】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),FALSE));

}

# void CHDICToolSetting::OnBnClickedButtonDemodSet()【6-3】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

// ((CButton\*)(GetDlgItem( IDC\_AUTO\_CHECK)))->SetCheck(0);

STTBX\_Print((L"Demod Select: %s !!",DemodType[m\_ComboBox\_Demod.GetCurSel()]));

switch(m\_ComboBox\_Demod.GetCurSel())

{

case 0:

Selected\_Demod\_Type = HDIC\_DEMOD\_2311;

STTBX\_Print((L"CTTB......."));

HDIC2311\_HWReset();

break;

case 1:

Selected\_Demod\_Type = HDIC\_DEMOD\_2311\_CABLE;

STTBX\_Print((L"CABLE......."));

HDIC2311\_HWReset();

break;

case 2:

Selected\_Demod\_Type = HDIC\_DEMOD\_2910;

if(m\_ComboBox\_TunerType.GetCurSel()==1) //zero IF

{

STTBX\_Print((L"!!HDIC2910 Not Suppport Zero IF!!"));

}

if(Selected\_TunerI2C\_Type == TUNER\_I2C\_THROUGHDEMOD)

{

STTBX\_Print((L"!!HDIC2910 Not Suppport I2C Through Demod!!"));

}

HDIC2910\_HWReset();

break;

case 3:

Selected\_Demod\_Type = HDIC\_DEMOD\_2815;

if(m\_ComboBox\_TunerType.GetCurSel()==1) //zero IF

{

STTBX\_Print((L"!!HDIC2815 Not Suppport Zero IF!!"));

}

HDIC2815\_HWReset();

break;

case 4:

Selected\_Demod\_Type = HDIC\_DEMOD\_29L1;

HDIC29L1\_HWReset();

break;

case 5:

Selected\_Demod\_Type = HDIC\_DEMOD\_29L2;

HDIC29L2\_HWReset();

break;

case 6:

Selected\_Demod\_Type = HDIC\_DEMOD\_2816;

HDIC2816\_HWReset();

break;

case 7:

Selected\_Demod\_Type = HDIC\_DEMOD\_2912;

HDIC2912\_HWReset();

break;

#if HDIC2910B\_SUPPORT

case 8:

Selected\_Demod\_Type = HDIC\_DEMOD\_2910B;

HDIC2910B\_HWReset();

break;

#endif

default:

break;

}

((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.DialogUpdate();

Demod\_PC\_Control();

}

# void CHDICToolSetting::OnBnClickedButtonDemodSet2()【6-4】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

STTBX\_Print((L"Clock Select: %s !!",ClockSelect[m\_Combo\_ClockSelect.GetCurSel()]));

switch(m\_Combo\_ClockSelect.GetCurSel())

{

case 0:

Selected\_Clock\_Type = CLOCK\_304M;

STTBX\_Print((L"30.4M......."));

break;

case 1:

Selected\_Clock\_Type = CLOCK\_27M;

STTBX\_Print((L"27M......."));

break;

case 2:

Selected\_Clock\_Type = CLOCK\_16M;

STTBX\_Print((L"16M......."));

break;

case 3:

Selected\_Clock\_Type = CLOCK\_10M;

STTBX\_Print((L"10M......."));

break;

}

}

# void CHDICToolSetting::OnBnClickedButtonTunerSet()【6-5】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

STTBX\_Print((L"Tuner Select:%s %s!!\n",TunerType[m\_ComboBox\_TunerType.GetCurSel()],TunerModel[m\_ComboBox\_TunerType.GetCurSel()][m\_ComboBox\_TunerModel.GetCurSel()]));

switch(m\_ComboBox\_TunerType.GetCurSel())

{

case 0:

switch(m\_ComboBox\_TunerModel.GetCurSel())

{

case 0:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDAC;

break;

case 1:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDQG;

break;

case 2:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDQD;

break;

case 3:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDAD;

break;

case 4:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDAG;

break;

case 5:

Selected\_Tuner\_Type = TUNER\_SHARP\_2093;

break;

case 6:

Selected\_Tuner\_Type = TUNER\_SHARP\_8002;

break;

case 7:

Selected\_Tuner\_Type = TUNER\_SHARP\_2403;

break;

case 8:

Selected\_Tuner\_Type = TUNER\_QINGJIA\_500;

break;

case 9:

Selected\_Tuner\_Type = TUNER\_NUTUNE\_2107;

break;

case 10:

Selected\_Tuner\_Type = TUNER\_ENTROPIC\_4020;

break;

case 11:

Selected\_Tuner\_Type = TUNER\_PANASONIC\_ET35;

break;

case 12:

Selected\_Tuner\_Type = TUNER\_MxL5007T;

break;

case 13:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDAC7;

break;

case 14:

Selected\_Tuner\_Type = TUNER\_ALPS\_TDAE;

break;

case 15:

Selected\_Tuner\_Type = TUNER\_SHARP\_2092;

break;

case 16:

Selected\_Tuner\_Type = TUNER\_LW\_1310;

break;

case 17:

case 18:

Selected\_Tuner\_Type = TUNER\_LG\_C842D;

break;

case 19:

Selected\_Tuner\_Type = TUNER\_NXP\_18723;

break;

case 20:

Selected\_Tuner\_Type = TUNER\_ZHLONGT;

break;

case 21:

Selected\_Tuner\_Type = TUNER\_GTFD32L;

break;

case 22:

Selected\_Tuner\_Type = TUNER\_SHARP\_2094;

break;

case 23:

Selected\_Tuner\_Type = TUNER\_YTMB\_04;

break;

case 24:

Selected\_Tuner\_Type = TUNER\_FC\_5100;

break;

case 25:

Selected\_Tuner\_Type = TUNER\_NMI\_120;

break;

case 26:

Selected\_Tuner\_Type = TUNER\_LG\_155F;

break;

default:

break;

}

break;

case 1:

switch(m\_ComboBox\_TunerModel.GetCurSel())

{

case 0:

Selected\_Tuner\_Type = TUNER\_ADI\_MTV102;

break;

case 1:

Selected\_Tuner\_Type = TUNER\_ADI\_MTV803;

break;

case 2:

Selected\_Tuner\_Type = TUNER\_ADI\_MTV803\_304;

break;

case 3:

Selected\_Tuner\_Type = TUNER\_SHHIC\_T2;

break;

case 4:

Selected\_Tuner\_Type = TUNER\_FC\_2581;

break;

case 5:

Selected\_Tuner\_Type = TUNER\_MR\_1500;

break;

case 6:

Selected\_Tuner\_Type = TUNER\_FC\_0013;

break;

case 7:

Selected\_Tuner\_Type = TUNER\_RAON\_MTV808;

break;

default:

break;

}

break;

default:

break;

}

}

# void CHDICToolSetting::OnBnClickedRadioI2cNotonchip()【6-6】消息：BN\_CLICKED

{

Selected\_TunerI2C\_Type = TUNER\_I2C\_THROUGHDEMOD;

STTBX\_Print((L"TUNER I2C THROUGH DEMOD"));

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

}

# void CHDICToolSetting::OnBnClickedRadioI2cOnchip()【6-7】消息：BN\_CLICKED

{

Selected\_TunerI2C\_Type = TUENR\_I2C\_BYPASSDEMOD;

STTBX\_Print((L"TUNER I2C BYPASS DEMOD"));

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

}

# void CHDICToolSetting::OnCbnSelchangeComboTunerType()【6-8】消息：BN\_CLICKED

{

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

m\_ComboBox\_TunerModel.ResetContent();

InitComboBox(m\_ComboBox\_TunerModel,TunerModel[m\_ComboBox\_TunerType.GetCurSel()]);

}

在类CHDICToolTest中

# void CHDICToolTest::OnBnClickedButtonStart()【7-1】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

if(m\_TestThread==NULL)

{

m\_TestThread=AfxBeginThread(HDICTestThread,NULL,THREAD\_PRIORITY\_TIME\_CRITICAL);

}

Test\_Count = 0;

Failed\_Count = 0;

Switch\_Flag = 0;

Manual\_Detect = 0;

Frequency1 = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Channel1\_Frequency.GetIntValue();

Frequency2 = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Channel2\_Frequency.GetIntValue();

Time\_Spend = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Time\_Spend.GetIntValue();

Time\_Wait = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Time\_Wait.GetIntValue();

Dec\_Check = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Dec\_Value.GetCheck();

Mode\_Check = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Check\_Mode.GetCheck();

Status\_Value = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Status\_Value.GetIntValue();

And\_Value = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Static\_And.GetIntValue();

Status\_Check = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Combo\_Status.GetCurSel();

Test\_Total = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Test\_Times.GetIntValue();

Once\_Stop = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Test.m\_Check\_OnceStop.GetCheck();

if(Selected\_Clock\_Type == CLOCK\_304M)

{

NCO\_Value = HDIC\_NCO\_36167;

}

else if(Selected\_Clock\_Type == CLOCK\_27M)

{

NCO\_Value = HDIC\_NCO\_36167\_27;

}

else if(Selected\_Clock\_Type == CLOCK\_16M)

{

NCO\_Value = HDIC\_NCO\_36167\_16;

}

else if(Selected\_Clock\_Type == CLOCK\_10M)

{

NCO\_Value = HDIC\_NCO\_36167\_10;

}

if((Selected\_Tuner\_Type==TUNER\_ADI\_MTV102)||(Selected\_Tuner\_Type==TUNER\_ADI\_MTV803)||(Selected\_Tuner\_Type==TUNER\_ADI\_MTV803\_304)||(Selected\_Tuner\_Type==TUNER\_SHHIC\_T2)

||(Selected\_Tuner\_Type == TUNER\_FC\_2581)||(Selected\_Tuner\_Type==TUNER\_MR\_1500)||(Selected\_Tuner\_Type==TUNER\_FC\_0013))

{

TunerType = HDIC\_TUNER\_ZERO\_IF;

NCO\_Value = HDIC\_NCO\_0;

}

else if(/\*(Selected\_Tuner\_Type==TUNER\_SHARP\_8002)||\*/(Selected\_Tuner\_Type==TUNER\_SHARP\_2093)||(Selected\_Tuner\_Type==TUNER\_ALPS\_TDAC)

||(Selected\_Tuner\_Type==TUNER\_ALPS\_TDAC7)

||(Selected\_Tuner\_Type==TUNER\_SHARP\_2092))

{

if(Selected\_Clock\_Type == CLOCK\_304M)

{

NCO\_Value = HDIC\_NCO\_36;

}

else if(Selected\_Clock\_Type == CLOCK\_27M)

{

NCO\_Value = HDIC\_NCO\_36\_27;

}

else if(Selected\_Clock\_Type == CLOCK\_16M)

{

NCO\_Value = HDIC\_NCO\_36\_16;

}

else if(Selected\_Clock\_Type == CLOCK\_10M)

{

NCO\_Value = HDIC\_NCO\_36\_10;

}

}

else if(Selected\_Tuner\_Type==TUNER\_QINGJIA\_500)

{

NCO\_Value = HDIC\_NCO\_36125;

}

else if(Selected\_Tuner\_Type==TUNER\_MxL5007T)

{

NCO\_Value = HDIC\_NCO\_6;

}

else if(Selected\_Tuner\_Type == TUNER\_LW\_1310)

{

NCO\_Value = HDIC\_NCO\_5070;

}

if(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodDataOutput.GetCurSel()==1)

{

TsType = HDIC\_MPEG\_OUTPUT\_SERIAL;

}

if(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodTSValid.GetCurSel()==1)

{

CLK\_Valid = HDIC\_MPEG\_CLK\_VALIDAWAYS;

}

if(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodMPEGClock.GetCurSel()==1)

{

CLK\_Polarity = HDIC\_MPEG\_CLK\_INVERTED;

}

if(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodAutoDetect.GetCurSel()==1)

{

Manual\_Detect = 1;

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodPNMode.GetCurSel())

{

case 0:

PN\_Mode = HDIC\_PN\_945; //945

break;

case 1:

PN\_Mode = HDIC\_PN\_595; //595

break;

case 2:

PN\_Mode = HDIC\_PN\_420; //420

break;

default :

break;

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodConstellation.GetCurSel())

{

case 0:

QAM\_Mode = HDIC\_QAM4\_NR; //4QAM-NR

break;

case 1:

QAM\_Mode = HDIC\_QAM4; //4QAM

break;

case 2:

QAM\_Mode = HDIC\_QAM16; //16QAM

break;

case 3:

QAM\_Mode = HDIC\_QAM32; //32QAM

break;

case 4:

QAM\_Mode = HDIC\_QAM64; //64QAM

break;

default :

break;

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodInterleave.GetCurSel())

{

case 0:

Interleaver\_Length = HDIC\_INTERLEAVER\_720; //720

break;

case 1:

Interleaver\_Length = HDIC\_INTERLEAVER\_240; //240

break;

default :

break;

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodCodeRate.GetCurSel())

{

case 0:

Carrier\_Rate = HDIC\_RATE\_04; //0.4

break;

case 1:

Carrier\_Rate = HDIC\_RATE\_06; //0.6

break;

case 2:

Carrier\_Rate = HDIC\_RATE\_08; //0.8

break;

default :

break;

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodCarrierMode.GetCurSel())

{

case 0:

Carrier\_Mode = HDIC\_CARRIER\_SINGLE; //C=1

break;

case 1:

Carrier\_Mode = HDIC\_CARRIER\_MULTI; //C=3780

break;

default :

break;

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_DemodPnVariable.GetCurSel())

{

case 0:

PN\_Variable = HDIC\_PN\_VARIABLE; //pn variable

break;

case 1:

PN\_Variable = HDIC\_PN\_FIXED; //pn fixed

break;

default :

break;

}

}

switch(((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Combo\_CableMode.GetCurSel())

{

case 0:

Cable\_Mode = HDIC\_CABLE\_16QAM; //16

break;

case 1:

Cable\_Mode = HDIC\_CABLE\_32QAM; //32

break;

case 2:

Cable\_Mode = HDIC\_CABLE\_64QAM; //64

break;

case 3:

Cable\_Mode = HDIC\_CABLE\_128QAM; //128

break;

case 4:

Cable\_Mode = HDIC\_CABLE\_256QAM; //256

break;

default :

break;

}

Cable\_Rate = ((CHDICToolDlg \*)(theApp.m\_pMainWnd))->m\_Demod.m\_Edit\_Cable\_Rate.GetIntValue();

HDICTest\_Enable = TRUE;

STTBX\_Print((L"Test Begin...."));

}

# void CHDICToolTest::OnBnClickedButtonStop()【7-2】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

HDICTest\_Enable = FALSE;

STTBX\_Print((L"Test Stop...."));

}

在类CHDICToolTuner中

# void CHDICToolTuner::OnAcceleratorEscape()【8-1】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),TRUE));

}

# void CHDICToolTuner::OnAcceleratorReturn()【8-2】消息：COMMAND

{

::SetFocus(::GetNextDlgTabItem(m\_hWnd,::GetFocus(),FALSE));

}

# void CHDICToolTuner::OnBnClickedButtonTunerSet()【8-3】消息：BN\_CLICKED

{

CString Cstr;

BOOL Result;

UINT32 Frequency\_Value= m\_Edit\_TunerChannelFrq.GetIntValue();

Result = HDICTuner\_Set(Frequency\_Value);

if(Result==TRUE)

{

STTBX\_Print((L"tuner locked!\n"));

Cstr.Format(L"locked!");

}

else if(Result == FALSE)

{

STTBX\_Print((L"tuner unlocked!\n"));

Cstr.Format(L"Unlocked!");

}

m\_Edit\_TunerStatus.SetWindowText((LPCTSTR)Cstr);

}

# void CHDICToolTuner::OnEnChangeEditTunerChannelfrq()【8-4】消息：EN\_CHANGE

{

// TODO: 如¨?果?该?控?件t是º? RICHEDIT 控?件t，ê?则¨°它¨¹将?不?会¨¢

// 发¤¡é送¨ª该?通ª¡§知a，ê?除y非¤?重?写¡ä CDialog::OnInitDialog()

// 函¡¥数ºy并¡é调Ì¡Â用®? CRichEditCtrl().SetEventMask()，ê?

// 同ª?时º¡À将? ENM\_CHANGE 标À¨º志?“¡ã或¨°”¡À运?算?到Ì?掩¨²码?中D。¡ê

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

CString Cstrc,Cstrf;

int ChannelFrq,i;

m\_Edit\_TunerChannelFrq.GetWindowText(Cstrf);

ChannelFrq=\_wtoi(Cstrf);

for(i=0;i<DMB\_FREQUENCY\_COUNT\_TOTOAL;i++)

{

if(FrequencyTable[i]==ChannelFrq)

{

break;

}

}

if(m\_Edit\_TunerChannelFrq.m\_hWnd==GetFocus()->m\_hWnd)

{

if(i<DMB\_FREQUENCY\_COUNT\_TOTOAL)

{

Cstrc.Format((LPCTSTR)Format,ChannelTable[i]);

m\_Edit\_TunerChannelNum.SetWindowText((LPCTSTR)Cstrc);

}

else

{

m\_Edit\_TunerChannelNum.SetWindowTextW(\_T("??"));

}

}

}

# void CHDICToolTuner::OnEnChangeEditTunerChannelnum()【8-5】消息：EN\_CHANGE

{

// TODO: 如¨?果?该?控?件t是º? RICHEDIT 控?件t，ê?则¨°它¨¹将?不?会¨¢

// 发¤¡é送¨ª该?通ª¡§知a，ê?除y非¤?重?写¡ä CDialog::OnInitDialog()

// 函¡¥数ºy并¡é调Ì¡Â用®? CRichEditCtrl().SetEventMask()，ê?

// 同ª?时º¡À将? ENM\_CHANGE 标À¨º志?“¡ã或¨°”¡À运?算?到Ì?掩¨²码?中D。¡ê

// TODO: 在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?

int ChannelNum,i;

ChannelNum=m\_Edit\_TunerChannelNum.GetIntValue();

for(i=0;i<DMB\_FREQUENCY\_COUNT\_TOTOAL;i++)

{

if(ChannelTable[i]==ChannelNum)

{

break;

}

}

if(m\_Edit\_TunerChannelNum.m\_hWnd==GetFocus()->m\_hWnd)

{

if(i<DMB\_FREQUENCY\_COUNT\_TOTOAL)

{

m\_Edit\_TunerChannelFrq.SetIntValue(FrequencyTable[i]);

}

else

{

m\_Edit\_TunerChannelFrq.SetWindowTextW(\_T("???"));

}

}

}

在类CHexEdit中

# void CHexEdit::OnEditClear() 【9-1】消息：COMMAND

{

m\_currentAddress = m\_selStart;

SelDelete(m\_selStart, m\_selEnd);

RepositionCaret(m\_currentAddress);

RedrawWindow();

}

# void CHexEdit::OnEditCut()【9-2】消息：COMMAND

{

//OnEditCopy();

SelDelete(m\_selStart, m\_selEnd);

RedrawWindow();

}

# void CHexEdit::OnEditSelectAll()【9-3】消息：COMMAND

{

m\_selStart = 0;

m\_selEnd = m\_length;

DestroyCaret();

RedrawWindow();

}

在类CregWindow中

# void CRegWindow::OnBnClickedButtonLoadfile()【10-1】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

UpdateData();

CFileDialog dlg(TRUE, NULL, m\_filename, OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,\_T("CSV Files (\*.CSV)|\*.CSV|All Files (\*.\*)|\*.\*||"));

if(dlg.DoModal()==IDOK)

{

m\_LoadFilename = dlg.m\_ofn.lpstrFile;

UpdateData(FALSE);

}

else

{

return;

}

CStdioFileEx fileEx;

CString GetLine;

BOOL bOK = TRUE;

int i0,i1,i2,i3,i4;

int num=0;

if (fileEx.Open(dlg.GetPathName(), CFile::modeRead | CFile::typeText | CStdioFileEx::modeWriteUnicode))

{

CString RegNameStr,RegAddrStr,RegNumStr,RegValStr,RegDefStr;

for(int i=0;i<240 && fileEx.ReadString(GetLine);i++)

{

if(GetLine.Left(2)!=L"//" && GetLine !=L"")

{

num++;

// for Reg Name

i1=GetLine.Find(L",",0); // get 1st "," positon

RegNameStr=GetLine.Left(i1);

// for Reg Address

i2=GetLine.Find(L",",++i1); // get 2nd "," position

i0=i2-i1;

RegAddrStr=GetLine.Mid(i1,i0);

RegAddrStr.MakeUpper();

// for Reg Number

i3=GetLine.Find(L",",++i2); // get 3rd "," position

i0=i3-i2;

RegNumStr=GetLine.Mid(i2,i0);

RegNumStr.MakeUpper();

#if 1

// for Reg Value

i4=GetLine.Find(L",",++i3); // get 3rd "," position

i0=i4-i3;

RegValStr=GetLine.Mid(i3,i0);

RegValStr.MakeUpper();

// for Definition

i0=GetLine.GetLength();

RegDefStr=GetLine.Right(i0-i4-1);

//AfxMessageBox(RegDefStr);

#endif

#if 0

// for Reg Value

i0=GetLine.GetLength();

RegValStr=GetLine.Right(i0-i3-1);

RegValStr.MakeUpper();

#endif

/////////////////////////

// save value to array///

int valNum=\_tcstoul(RegAddrStr, 0, 16);

valRegAddress[num-1]=valNum;

strRegName[valNum]=RegNameStr;

strRegAddress[valNum]=RegAddrStr;

strRegNum[valNum]=RegNumStr;

strRegValue[valNum]=RegValStr; // length is 2 or 4

strRegDefinition[valNum]=RegDefStr;

}

}

}

fileEx.Close();

m\_RegPage->ShowPanelNumber(num);

UpdateData(FALSE);

}

# void CRegWindow::OnBnClickedButtonReadregall()【10-2】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

m\_RegPage->Read\_Register\_ALL();

for(int i=0;i<30;i++){

m\_Progress.SetPos(i);

}

}

# void CRegWindow::OnBnClickedButtonSavefile()【10-3】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

CFileDialog dlg(FALSE,NULL,NULL,OFN\_HIDEREADONLY|OFN\_OVERWRITEPROMPT,L"CSV Files (\*.CSV)|\*.CSV|All Files (\*.\*)|\*.\*||",AfxGetMainWnd());

if(dlg.DoModal()==IDOK)

{

CString strPath=dlg.GetPathName();

CStdioFileEx fileEx;

if(strPath.Right(4) !=L".csv") // add exname

{

m\_SaveFilename=dlg.GetPathName()+L".csv";

if (fileEx.Open(dlg.GetPathName()+L".csv",CFile::modeCreate | CFile::modeWrite | CFile::typeText/\* | CStdioFileEx::modeWriteUnicode\*/))

{

fileEx.WriteString(m\_RegPage->Get\_Para\_str());

}

}

else

{

m\_SaveFilename=dlg.GetPathName();

if (fileEx.Open(dlg.GetPathName(),CFile::modeCreate | CFile::modeWrite | CFile::typeText/\* | CStdioFileEx::modeWriteUnicode\*/))

{

fileEx.WriteString(m\_RegPage->Get\_Para\_str());

}

}

fileEx.Close();

}

}

# void CRegWindow::OnBnClickedButtonWriteregall()【10-4】消息：BN\_CLICKED

{

// TODO: Add your control notification handler code here

m\_RegPage->Write\_Register\_ALL();

for(int i=0;i<30;i++){

m\_Progress.SetPos(i);

}

}