//poweron music

case EN\_FACTORY\_PAGE\_AUDIO:

stGenSetting.g\_FactorySetting.u8PWROnMusicVolume= MApp\_ZUI\_ACT\_DecIncValue(bInc, stGenSetting.g\_FactorySetting.u8PWROnMusicVolume, 0, 100, 1);

MApp\_SaveFactorySetting();

return EN\_REDRAW\_ITEM;

// line out value

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadByte(0x112D2A);

u16TempVal = MApp\_ZUI\_ACT\_DecIncValue(bInc,u16TempVal,0,0xff,1);

MDrv\_WriteByte(0x112D2A,u16TempVal);

return EN\_REDRAW\_ITEM;

break;

//avc enable /disable

case EN\_FACTORY\_PAGE\_AUDIO:

stGenSetting.g\_SysSetting.fAutoVolume =!stGenSetting.g\_SysSetting.fAutoVolume;

MApi\_AUDIO\_EnableAutoVolume((BOOLEAN)stGenSetting.g\_SysSetting.fAutoVolume);

MDrv\_WriteByte(0x112D28, ((stGenSetting.g\_SysSetting.fAutoVolume) ? 0x1C: 0x18));

MApp\_SaveSysSetting();

return EN\_REDRAW\_ITEM;

//avc threshold

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadByte(0x112D28);

u16TempVal = MApp\_ZUI\_ACT\_DecIncValue(bInc,u16TempVal,0,0xff,1);

MDrv\_WriteByte(0x112D28,u16TempVal);

return EN\_REDRAW\_ITEM;

//drc enable

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadRegBit(0x112D82,BIT5);

if(u16TempVal == 0)

MDrv\_WriteRegBit(0x112D82,ENABLE,BIT5);

else

MDrv\_WriteRegBit(0x112D82,DISABLE,BIT5);

return EN\_REDRAW\_ITEM;

//drc threshold

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadByte(0x112D34);

u16TempVal = MApp\_ZUI\_ACT\_DecIncValue(bInc,u16TempVal,0,0xff,1);

MDrv\_WriteByte(0x112D34,u16TempVal);

return EN\_REDRAW\_ITEM;

//sound effect

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadRegBit(0x112D82,BIT6);

if(u16TempVal == 0)

MDrv\_WriteRegBit(0x112D82,ENABLE,BIT6);

else

MDrv\_WriteRegBit(0x112D82,DISABLE,BIT6);

return EN\_REDRAW\_ITEM;

// sound tone effect

case EN\_FACTORY\_PAGE\_AUDIO:

u16TempVal = MDrv\_ReadRegBit(0x112D82,BIT3);

if(u16TempVal == 0)

MDrv\_WriteRegBit(0x112D82,ENABLE,BIT3);

else

MDrv\_WriteRegBit(0x112D82,DISABLE,BIT3);

return EN\_REDRAW\_ITEM;