Video Library for CCP XS

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the CoDeSys Video library which enables configuration of the CCVideoXS application in a CoDeSys PLC program.

1.2 References

-

1.3 Revision history

Rev	Date	Author	Comments
1.0	2007-11-21	Fredrik Lans, CCS	Initial version
1.1	2008-01-22	Tobias Andersson, CCS	Start, stop added
1.2	2008-11-21	Lars Gustafsson, CCS	Start now has 5 input parameters

2 Background and principles

The PLC library described in this document enables control and configuration of the CCVideoXS application in a CoDeSys PLC program, if a CCP XS is used as target hardware. The CCP XS is equipped with a video input for connection of an external PAL or NTSC camera.

The library will give the user functions to start and stop the video application and to change the video window appearance. Possible changes are to minimize, restore, set position, set placement and to set full screen mode. There is also a function for setting video input channel.

The following list shows which files that are involved when using the Video Library.

Filename	Location	Comment
`Video.dll`	`CCP XS: \Program Files\SoftPLC\`	The soft plc video driver
`VideoLibrary.lib`	`PC: CoDeSys Lib Dir`	
`CCVideoXS.exe`	`CCP XS: \Windows\`	The video standalone application

3 Installation

In order to work, the Video library needs to have the Video driver and the CCVideoXS application installed on the target hardware. The Video driver dll used by the target runtime is called Video.dll. This file has to be located in the same directory as the runtime executable. In addition, the Video dll has to be registered in order to be loaded by the runtime. This is done by modifying the runtime configuration file rts3s.cfg also located in the runtime directory. Below are instructions for installing the Video driver and the CCVideoXS application.

3.1 Install Video driver

1. Locate your runtime directory on the target file system. The path is usually \Program Files\SoftPLC\. If there is no such directory, search for the file PLCCEARM.exe. The directory where you find the file is your runtime directory.

2. Copy the Video.dll file to the runtime directory. If the file is already there, your Video driver is already installed and you can skip the rest of this section.

3. Open the rts3s.cfg file in a text editor. It should look something like this:

   ```
   [IODRIVERLIST]
   MODULE0=CCPXS
   INITFCT0=IODrvInit
   MODULE1=3SCanFrame
   INITFCT1=IODrvInit
   [PLC]
   SETTINGS_FROM_CFG=Yes
   USE SYSINTR_TIMING AS TICK=Yes
   Files=
   BootMode=Run
   ```

   Now, insert these two lines in the IO driver list:

   ```
   MODULE2=Video
   INITFCT2=IODrvInit
   ```

   If there are more than two drivers present in the IO driver list, just increase the module number. Your file should now look something like this:

   ```
   [IODRIVERLIST]
   MODULE0=CCPXS
   INITFCT0=IODrvInit
   MODULE1=3SCanFrame
   INITFCT1=IODrvInit
   MODULE2=Video
   INITFCT2=IODrvInit
   [PLC]
   SETTINGS_FROM_CFG=Yes
   USE SYSINTR_TIMING AS TICK=Yes
   Files=
   BootMode=Run
   ```

4. Now you can check if your runtime loads the Video dll. If your runtime is already started, shut it down. Then, start it by double clicking on the file PLCCEARM.exe. Login to your PLC from CoDeSys. Go to the ‘resources’ tab and choose the PLC Browser. Enter ‘rtsinfo’ in the command line prompt. In the information you receive from the PLC, the loading of the Video driver should be present. If not, check that you have followed the steps above correctly.

5. Now, place the library file VideoLibrary.lib in your CCP XS target library directory in CoDeSys. Your library directory can be located by starting CoDeSys, creating a new project with CCP XS as target. Then, open the Project->Options dialog. Choose ‘Directories’. There, you can see the library paths used by CoDeSys.

6. Now you are ready to use the Video library.

3.2 Install CCVideoXS application

1. Open the Windows directory on the target file system.

2. Copy the CCVideoXS.exe file to the Windows directory. If the file is already there, your CCVideoXS application is already installed.

3. Now you are ready to use the CCVideoXS application.

4 Using the Video library

The following section will describe how you create a project where you use the Video library.

1. Create a new project in CoDeSys. Choose the proper target (CCP XS).

2. Go the Resources tab. Choose Library manager.

3. Choose Insert in the main menu. Choose the file VideoLibrary.lib.

4. Choose Insert again. Choose the file SysLibSem.lib. This library should already be installed with your CCP XS target installation.

5. In your program code, call the Video library functions to control the CCVideoXS.exe application. The functions are described in detail further down in this document.

6. Start the plc application.

5 Functions

5.1 XSVideo_StartVideo

Starts the video application

```
FUNCTION XSVideo_StartVideo : BOOL
VAR_INPUT
           bNoFrame              : BOOL;
           bStartMinimized       : BOOL;
           bCropToSize           : BOOL;
           iCropWidth            : INT;
           iCropHeight           : INT;
END_VAR
VAR
END_VAR
```

If the application starts successfully the function returns TRUE.

5.2 XSVideo_StopVideo

Closes the video application

```
FUNCTION XSVideo_StopVideo : BOOL
VAR_INPUT
END_VAR
VAR
END_VAR
```

5.3 XSVideo_MinimizeWin

Sets window to state minimized

```
FUNCTION XSVideo_MinimizeWin : BOOL
VAR_INPUT
           iDummy : INT;
END_VAR
VAR
END_VAR
```

The minimize function takes an integer value as input parameter. This input parameter is currently not used and can therefore be set to anything. If minimization succeeds, then value TRUE is returned.

5.4 XSVideo_RestoreWin

Restores the window size to the size it had before it was minimized.

```
FUNCTION XSVideo_RestoreWin : BOOL
VAR_INPUT
           iDummy : INT;
END_VAR
VAR
END_VAR
```

The restore function takes an integer value as input parameter. This input parameter is currently not used and can therefore be set to anything. If restore operation succeeds, then value TRUE is returned.

5.5 XSVideo_SetChannel

Sets video input channel.

```
FUNCTION XSVideo_SetChannel : BOOL
VAR_INPUT
           iChannel : INT;
END_VAR
VAR
END_VAR
```

The set channel function takes an integer value as input parameter. This input parameter determines which channel to set. If the operation succeeds, then value TRUE is returned.

5.6 XSVideo_SetFullScreen

Sets video window to full screen mode.

```
FUNCTION XSVideo_SetFullScreen : BOOL
VAR_INPUT
           iFullScreen : INT;
           (* When 1, picture will be set to fullscreen *)
           (* When 0, picture will be set not to fullscreen *)
END_VAR
VAR
END_VAR
```

The set full screen function takes an integer value as input parameter. This input parameter determines if full screen is set or not. Input = 1 for fullscreen and input = 0 for not fullscreen. If the operation succeeds, then value TRUE is returned.

5.7 XSVideo_SetWinPlace

Sets video window to specified position.

```
FUNCTION XSVideo_SetWinPlace : BOOL
VAR_INPUT
           iNewLeft : INT;
           iNewTop : INT;
END_VAR
VAR
END_VAR
```

The set windows place function takes two integer values as input parameters. The input parameters iNewLeft and iNewTop, determines the new left and top position to set for the window. If the operation succeeds, then value TRUE is returned.

5.8 XSVideo_SetWinSize

Sets video window to specified size.

```
FUNCTION XSVideo_SetWinSize : BOOL
VAR_INPUT
           iNewWidth : INT;
           iNewHeigth : INT;
END_VAR
VAR
END_VAR
```

The set windows size function takes two integer values as input parameters. The input parameters iNewWidth and iNewHeigth, determines the new width and height position to set for the window. If the operation succeeds, then value TRUE is returned.

6 Code example

This section gives an example of code for using the Video library. In order to use the code the libraries VideoLibrary.lib and the SysLibSem.lib has to be added to the project as described in previous sections.

```
(* Variable declaration*)
VAR
```

```
           Initialized : BOOL := FALSE;
END_VAR

(*First loop, call the SetWinPlace and SetWinSize functions. *)
IF initialized = FALSE THEN
           XSVideo_SetWinPlace(0, 0);
           XSVideo_SetWinSize(640, 440);
           Initialized := TRUE;
END_IF
```