CIFX provider

hilscher CIFX board

Version 1.0.6

User’s guide

Jan 17, 2017

[Remarks]

[Revision history]

Version	Date	Description
1.0.0.0	2010-6-18	First edition
1.0.1.0	2011-3-14	Added CaoWorkspace::AddController option Added variables. Added description of provider registration tool.
1.0.2.0	2011-5-24	Added variables.
1.0.3.0	2011-6-6	Added variables.
1.0.4.0	2011-6-20	Added variables.
1.0.5.0	2011-8-10	Added variables.
1.0.5	2012-7-17	Changed the version rule of document.
1.0.6	2013-1-22	Added CaoWorkspace::AddController option
	2017-1-17	Corrected some clerical errors. Added appendix.

[Supported models]

Model	Version	Note

Contents

1. Introduction

This is a user's guide of CIFX provider that accesses CIFX board manufactured by hilscher. Available CIFX boards are CIFX50-DN, CIFX50-CC, CIFX50-DP, and CIFX50-RE. For details about CIFX board, refer to CIFX device driver manual of hilscher.

> Attention: To use this CIFX provider, you need to install the device driver of the CIFX board. After the installation, you are required to register this provider registry. For information about registration, refer to Table 2-1.

2. Overview of provider

2.1. Overview

CIFX provider is CAO provider that wraps CIFX driver's APIs to operate it via ORiN technology. The file format is DLL (Dynamic Link Library) and it is automatically loaded from CAO engine when it is used. To use CIFX provider, you need to manually register the information of Table 2-1.

Table 2-1 CIFX provider

File name	CaoProvCIFX.dll
ProgID	CaoProv.hilscher.CIFX
Registry registration[1]	regsvr32 CaoProvCIFX.dll
Deregistration	regsvr32 /u CaoProvCIFX.dll

[1] You can register the provider with regsvr32.exe or RegCOM.exe ([Start]→[ORiN2]→[Tools]). CIFX provider cannot be registered unless the device driver of CIFX board has installed.

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

CIFX provider establishes communication with CIFX board when a Controller object is created.

Syntax AddController(<bstrCtrlName: BSTR>,<bstrProvName: BSTR>,
<bstrPcName: BSTR > [,<bstrOption: BSTR>])

bstrCtrlName : [in] Controller name
bstrProvName : [in] Provider name. Fixed to " CaoProv.hilscher.CIFX".
bstrPcName : [in] Computer name where provider runs
bstrOption : [in] Option character string

The following table lists the option character strings.

Table 2-2 Option character string of CaoWorkspace::AddController

Option	Description
Board[=<Board number>]	CIFX board number of connection destination (Default: 0)
Slot[=<Slot number>]	Identify the connection destination CIFX with the rotary switch. This option takes precedence over Board option.
Timeout[=<Communication timeout>]	Communication timeout (Default: 10)
ResetTimeout[=<Reset timeout>]	Reset timeout (Default: 10000)

2.2.2. CaoController::AddCommand method

For available command names and descriptions, refer to Table .

There is no option character string entry required.

Syntax AddCommand(<bstrName: BSTR > [,<bstrOption: BSTR>])

bstrName : [in] Any name
bstrOption : [in] Option character string (not used)

Table 2-3 Command class reserved word list

Reserved word (command)	Data type of parameter	Parameter setting
Reset	none	none

Init	none	none
HostStart	none	none
HostStop	none	none
BusOn	none	none
BusOff	none	none

2.2.3. CaoController::AddVariable method

This method generates a variable object that accesses CIFX board.

For variable naming, only the variables written in 2.3.1 are available. This method returns an error if other variable names are used.

Syntax AddVariable(<bstrName: BSTR > [,<bstrOption: BSTR>])

bstrName : [in] Any name

bstrOption : [in] Option character string

For the option character string, enter the buffer size as described below.

Table 2-4 Option character string of CaoController::AddVariable

Option	Description
Size[=<Data length>]	Buffer size (Unit : byte. Default: 1)

2.2.4. CaoController:: get_VariableNames property

Obtain variables described in 2.3.1.

2.2.5. CaoController::get_CommandNames property

Output the reserved word list that are available in Command class.

There is no option entry required.

For details, refer to Table .

2.2.6. CaoController::get_ID property

Obtains the currently connected CIFX board number.

2.2.7. CaoVariable::get_Value property

Obtain information about a specified variable. For information about variables, please refer to 2.3.1.

2.2.8. CaoVariable::put_Value property

Set information about a specified variable. For information about variables, please refer to 2.3.1.

2.2.9. CaoCommand::Execute property

Execute a specified command. For information about available commands, please refer to 2.2.2.

2.3. Variable list

2.3.1. Controller class

Table 2-5 Controller class User variable list

Variable name	Data type	Description	Attribute		Option
			get	put	Size
SYSPACKET	VT_ARRAY | VT_UI1	Obtain and set system packets. Sending packet is CIFX_PACKET structure.	✓	✓	-
CHANNELPACKET	VT_ARRAY | VT_UI1	Obtain and set channel packets. Sending packet is CIFX_PACKET structure.	✓	✓	-
IO?	VT_ARRAY | VT_UI1	Obtain and set the current value of I/O. For "?" in the end of variable name, please enter the offset value.	✓	✓	✓
FIO?	VT_ARRAY | VT_UI1	Obtain and set the current value of I/O. For "?" in the end of variable name, please enter the offset value.	✓	✓	✓

Table 2-6 Controller class System variable list

Variable name	Data type	Description	Attribute		Option
			get	put	Size
@FIRMWARE_NAME	VT_BSTR	Obtain a file name of firmware.	✓	-	-
@FIRMWARE_VERSION	VT_BSTR	Obtain the firmware version.	✓	-	-
@DEVICE_NO	VT_UI4	Obtain the manufacturing number of CIFX board.	✓	-	-
@SERIAL_NO	VT_UI4	Obtain the serial number of CIFX board.	✓	-	-

@RECEIVE_PACKET_CNT	VT_UI4	Obtain the number of packets receivable from CIFX board.	✓	-	-
@SEND_PACKET_CNT	VT_UI4	Obtain the number of packets transferable to CIFX board.	✓	-	-
@CHANNEL_OPEN	VT_BOOL	Check if Channel0 is opened or not. This value becomes 0 if firmware is not loaded in CIFX board at the timing of AddController.	✓	-	-
@HOST_STATE	VT_BOOL	Obtain the HOST state of Channel0. True: Ready-state False: Not Ready-state	✓	-	-
@BUS_STATE	VT_BOOL	Obtain the BUS status of Channel0. True: BUS ON False: BUS OFF	✓	-	-
@SLOT_NO	VT_UI4	Obtain the CIFX rotary switch number.	✓	-	-
@HW_OPTIONS	VT_ARRAY | VT_I2	Obtain the CIFX hardware option information.	✓	-	-
@MASTER_LICENSE	VT_BOOL	Check if the CIFX holds a master license. True: Holding a master license False: Not holding a master license	✓	-	-
@HW_REVISION	VT_UI4	Obtain the revision of hardware.	✓	-	-

3. Appendix

3.1. Motion of CIFX

To use CIFX provider, 2nd stage bootloader, field network's protocol stack, and WarmStartFile for setup are required.

Each file shall be stored in the following folder structure.

+ <BASEDIR>
| - - (2nd stage bootloader)
| - - + <Model number>_<Serial number> (Base directory of device. Example: 1250100_20217)
| | - - + Channel0 (File directory for Channel0)
| | | - - cifXeis.nxf (Protocol stack. File differs depending on the field network type.)
| | | - - warmstart.dat (WarmStartFile)

<BASEDIR> is stored in "HKLM¥SYSTEM¥CurrentControlSet¥Services¥CIFxDrv¥InstallDir" in the registry. Default setting is "C: ¥Program Files¥cifX Device Driver".

Model number is stored in DeviceNumber and serial number is stored in Serial number, both of which are under HKLM¥SYSTEM¥CurrentControlSet¥Services¥CIFxDrv¥Devices¥Board0.

These files are generated when the setup is done with "cifX setup", which is stored in the CD provided with CIFX and are imported at the startup of Windows. You can change the setting if you exclusively send warmstart file with xChannelPutPacket.

Table 3-1 shows the description of files.

Table 3-1 CIFX file

File	Description
2nd stage bootloader	This initializes the CIFX board before executing a protocol stack. The file is stored in a CD that is provided with CIFX. File name : NXCIF50-RTE.bin
Protocol stack	Firmware (Extension: nxm or nxf) This firmware determines the field network where the provider runs. However, available field networks are limited according to the board used. The following shows the available combinations of CIFX boards and files. If a file is saved with any other board combination, the board cannot be used. Files are stored in a CD that is provided with CIFX. CIFX50-DN: cifxdnm.nxf (DeviceNet Master), cifxdns.nxf (DeviceNet Slave) CIFX50-CC: cifxccs.nxf (CC-Link Slave) CIFX50-DP: cifxdpm.nxf (Profibus-DP Master), cifxdps.nxf (Profibus-DP Slave) CIFX50-RE: cifxecm.nxf (EtherCAT Master), cifxecs.nxf (EtherCAT Slave), cifxeim.nxf (EtherNet/IP Master), cifxeis.nxf (EtherNet/IP Slave), cifxomb.nxf (Modbus TCP/IP Server), cifxpls.nxf (PowerLink Slave), cifxpnm.nxf (Profinet Master), cifxpns.nxf (Profinet Slave), cifxs3s.nxf (SercosIII Slave)
WarmStartFile	This file configures each field network. This is a structured damp file. The structure to be used is different depending on the field network you select. For the structure selection, please refer to API manual of each field network. (Example : For EtnerNet/IP Adapter, EIP_APS_PACKET_WARMSTART_PRM_REQ_T) File name : warmstart.dat