EzProg-I Getting Started

(Version 4.2)

EzProg

ICP DAS CO., LTD.

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Technical Support

If you have problems about using the product, please contact ICP DAS Product Support.

Email: Service@icpdas.com

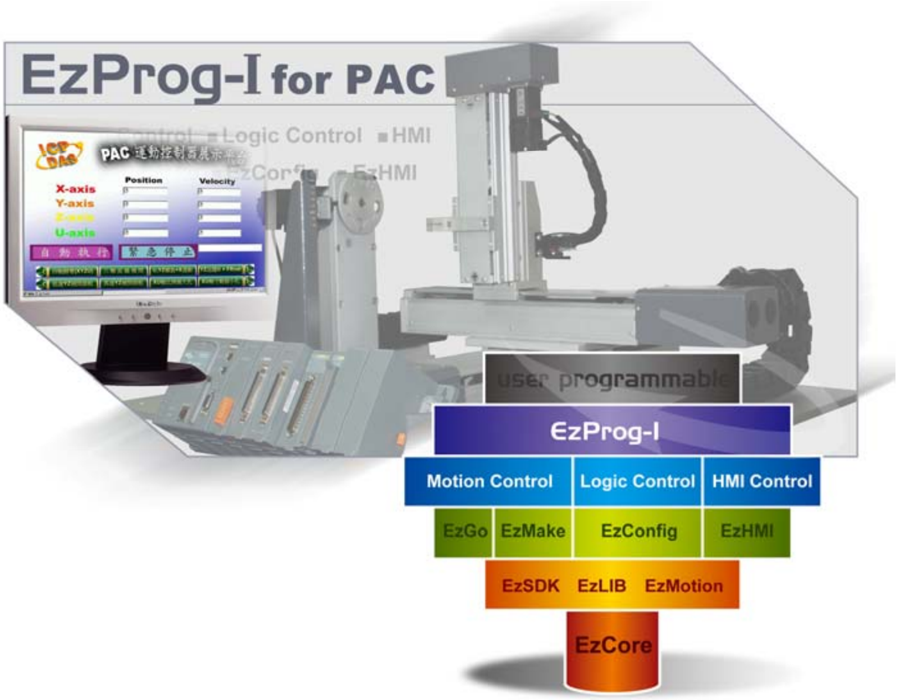
Table of Contents

1 Introduction 4
1.1 Library and utilities 5
1.2 Application framework 8
1.3 IO and register table 9
2 Development Tools Installation 10
2.1 SDK Installation Guide 10
2.2 Installation of EzProg-I Development Resources 15
2.3 EzProg-I Development Resources 19
2.3.1 Manuals 19
2.3.2 ActiveX 20
2.3.3 EzTemplate 20
2.3.4 Examples 21
3 EzTemplate Applications 22
3.1 Creating a simple user interface for monitoring digital IO statuses 22
3.2 Procedures on the MPAC 22
3.2.1 Hardware configuration 22
3.2.2 IO register mapping 23
3.3 Procedures on the PC 26
3.3.1 EzTemplate 26
3.3.2 Using EzTemplate 26
3.3.3 HMI design and IO register linking 29
4 Appendix 36
4.1 Adding EzHMI ActiveX controls to the Visual Studio Toolbox 36
4.2 Connecting to MPAC via Ethernet 39

1 Introduction

For the past few years the number of programmable automation controller (PAC) in the industry and their field of applications have grown rapidly. Requirements such as high performance, easy integration, extensibility, fast software development and short time to market are an increasing demand of the industry. ICPDAS offers the hardware and software solution to meet the demands. The EzCore engine provided by ICPDAS is a development kit to simplify and reduce the software development expenditure.

The main purpose of the EzCore engine is to assist the programmer to develop a logic control program on the Windows CE platform. The EzCore comprises of numerous tools, libraries and HMI to assist the system developer to implement the control operation. EzHMI is a set of HMI ActiveX controls, which consist of graphic controls for the user interface.
EzConfig is a utility to read, set and test the system framework configuration. The framework comprises of IO, retentive and none retentive registers. The motion control part offers the EzGo and EzMake utility to assist the developer to set up a motion control system by enabling him to configure and to test run the servo motors. The EzCore allows the system engineer to focus his energy on designing the logic control system and thereby achieving greater results in a shorter period of time.

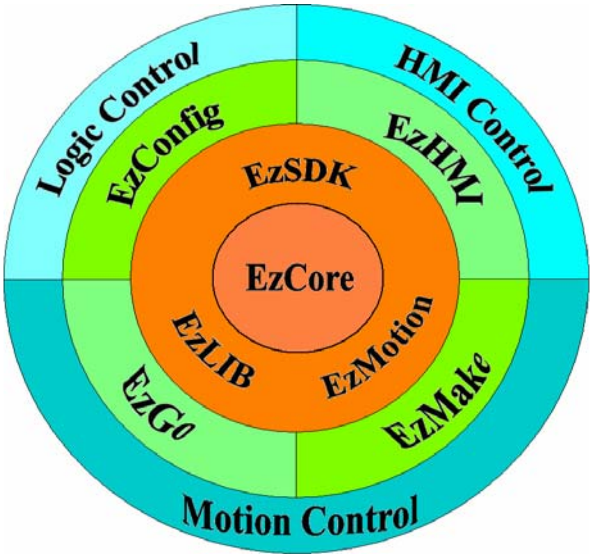


The EzProg-I is installed on the following directory on the Windows CE PAC:

- for MPAC: "\System_Disk\EzProg-I\"

1.1 Library and utilities

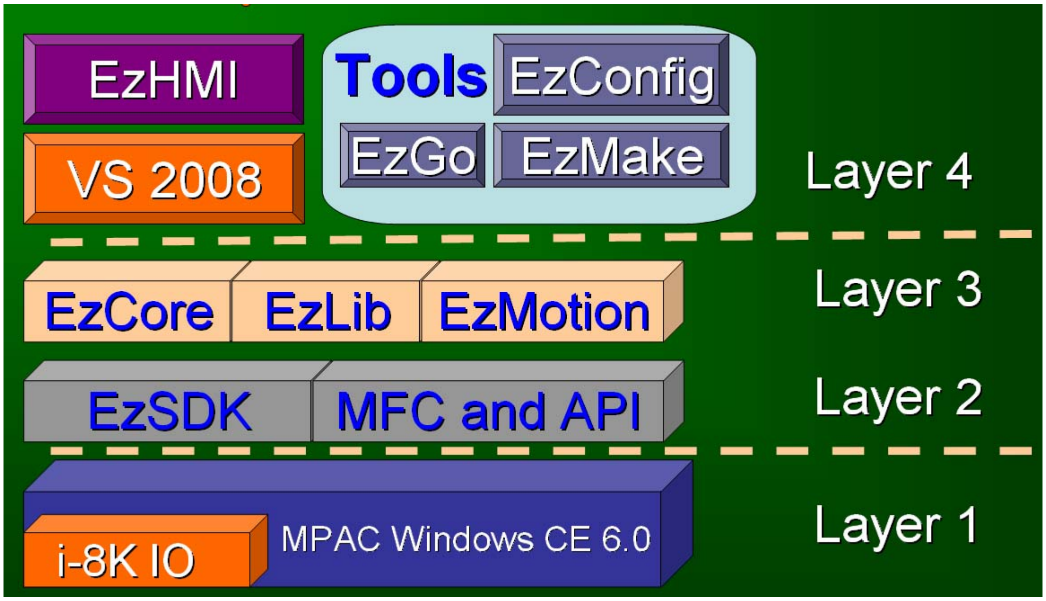
EzCore and VS2008:



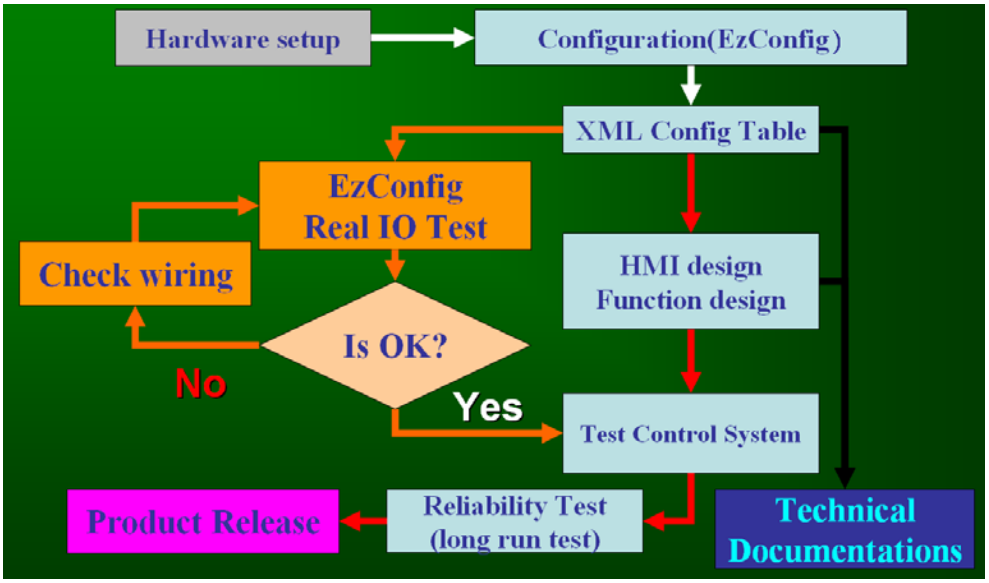
EzProg-I is a developing toolkit consisting of libraries, utilities and HMI controls for the Windows CE 6.0 platform. EzCore is the engine driving EzProg-I and its APIs are responsible for the communication between the hardware (IO), HMI controls and the registers. All the libraries are written for the VS2008 environment.

The basic development structure of EzProg-I:

- The lowest level consists of the real time WinCE operation system and the hardware driver.
- IO APIs and the MFC APIs are part of the next layer.
- One layer up is the main application layer composed of EzCore, EzLib and EzMotion.
- The top layer is made up of the control program created by VS2008, the human machine interface (EzHMI), the configuration and testing utilities.



Installing, configuration and testing the devices of the control system (IO, wiring, switches etc.) can be done without the actual control program. Therefore hardware and software implementation can be done independently from one another. This shortens the development time and reduces cost.

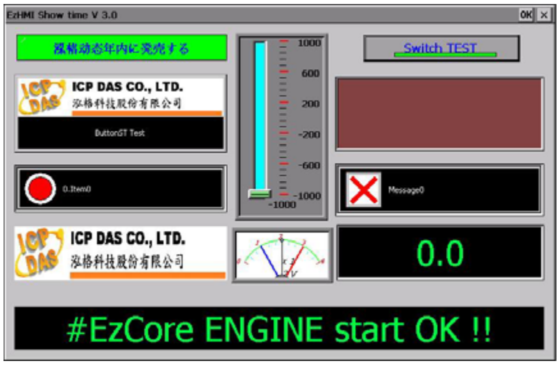


The main task of EzCore:

EzCore has a real time scanning engine which supports hardware interrupt and direct access of IO registers and other system related registers. EzCore offers eight real time interrupt service routine RTSR (similar to a task in a multitasking programmable logic controller) and eight user threads. The hardware interrupt is supported by the following modules

- DI module: 8094H in slot 1
- Motion modules: 8094A, 8094F, 8092F (in slot 1 to 3)

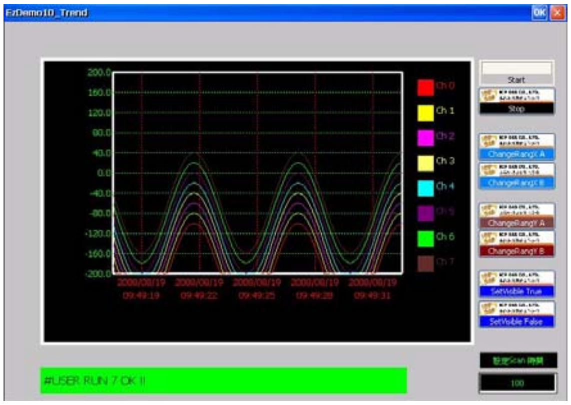
EzHMI ActiveX controls:



#EzCore ENGINE start OK !!

- EzProg-I EzHMI(ActiveX)
- Direct display of IO status
- Direct display of register values
- Multilanguage support
- Support flashing (alarm)
- Picture movement
- Font setting
- Color setting
- Enable/disable objects

EzLIB application library



#USER RUN 7 OK !!

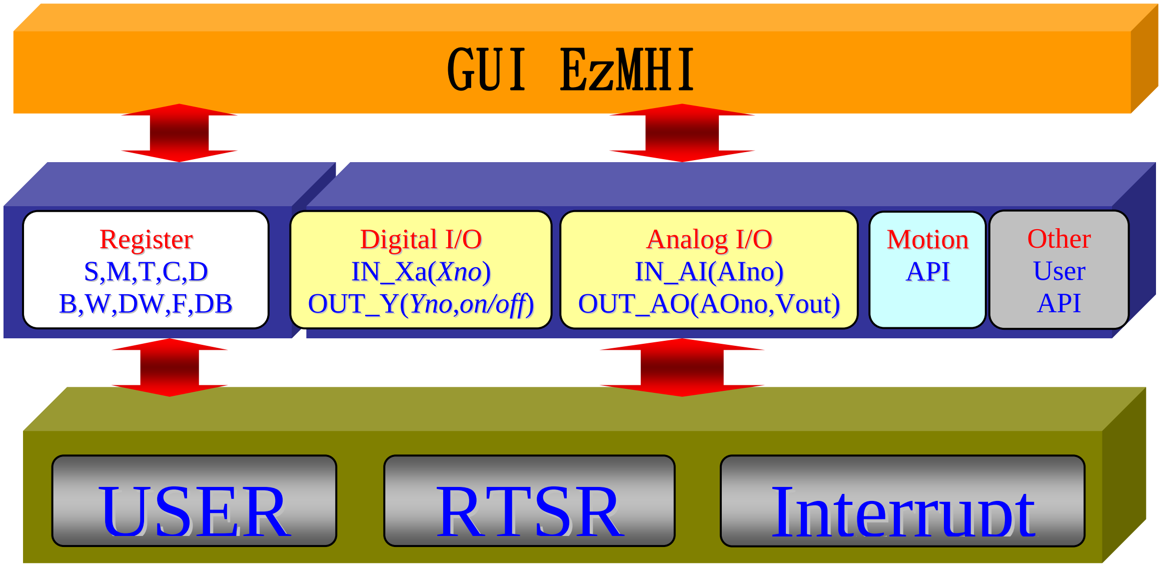
- EzLIB offers easy to use applications
- Various kinds of conversion functions
- Date, time functions
- Drawing functions
- BMP save to file function
- TCP/IP functions
- FTP functions
- Trend diagram support

Servo motor control

EzProg-I supports the following 2 axis and four axis motion modules:
i8092F, i8094, i8094F, i8094A, i8094H.
EzGo and EzMake are utilities for configuring, programming and testing the motion modules. For more information refer to the motion manuals.

1.2 Application framework

The EzProg-I main framework structure has the following layout:



System consists of three parts:

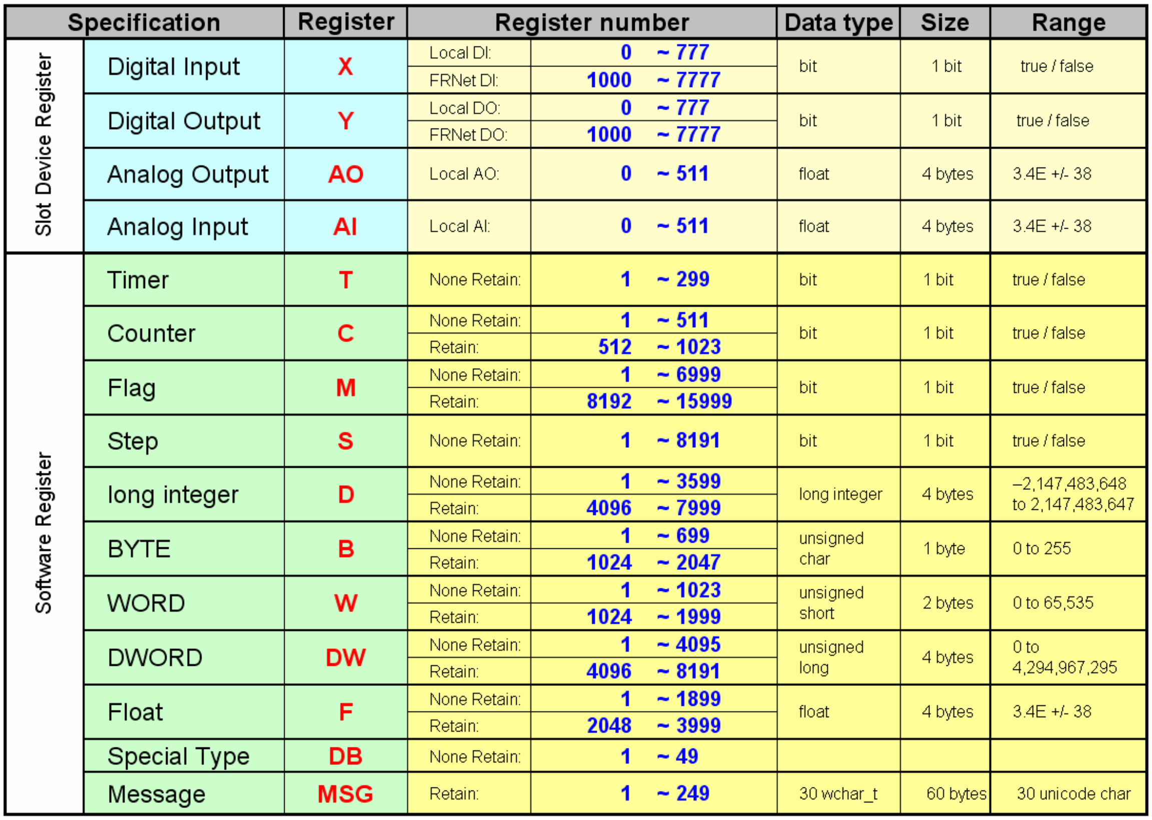
1. Top layer:
   EzHMI provides a number of ActiveX controls which allows the programmer to design a user interface on a WinCE system for monitoring and controlling purposes. The ActiveX controls can be directly linked to IO registers for displaying or manipulating of IO data. In addition the ActiveX controls support Multilanguage.

2. Middle layer
   EzProg1 has a section of memory called register where bit, integer, floating point and string values can be stored and accessed. These registers can be accessed by the upper and bottom layer. EzHMI controls can be linked directly to a register type and register number to access a value.

3. Bottom layer
   a. User thread: Programmer has to add code to the function. The code will be executed only once
   b. RTSR: Similar to plc coding. This API will be called at fixed time intervals.
   c. Hardware interrupt: The Interrupt has got the highest priority and immediately interrupts the execution of other tasks. After the interrupt handler has completed it execution code a thread with lower priority will be executed.

1.3 IO and register table

For a detailed description of the EzProg-IO and register table please refer to the EzProg-I_Tool manual.
Currently the following register types are defined:



Specification		Register	Register number			Data type	Size	Range
Slot Device Register	Digital Input	X	Local DI:	0	~ 777	bit	1 bit	true / false
			FRNet DI:	1000	~ 7777			
	Digital Output	Y	Local DO:	0	~ 777	bit	1 bit	true / false
			FRNet DO:	1000	~ 7777			
	Analog Output	AO	Local AO:	0	~ 511	float	4 bytes	3.4E +/- 38
	Analog Input	AI	Local AI:	0	~ 511	float	4 bytes	3.4E +/- 38
Software Register	Timer	T	None Retain:	1	~ 299	bit	1 bit	true / false
	Counter	C	None Retain:	1	~ 511	bit	1 bit	true / false
			Retain:	512	~ 1023			
	Flag	M	None Retain:	1	~ 6999	bit	1 bit	true / false
			Retain:	8192	~ 15999			
	Step	S	None Retain:	1	~ 8191	bit	1 bit	true / false
	long integer	D	None Retain:	1	~ 3599	long integer	4 bytes	−2,147,483,648 to 2,147,483,647
			Retain:	4096	~ 7999			
	BYTE	B	None Retain:	1	~ 699	unsigned char	1 byte	0 to 255
			Retain:	1024	~ 2047			
	WORD	W	None Retain:	1	~ 1023	unsigned short	2 bytes	0 to 65,535
			Retain:	1024	~ 1999			
	DWORD	DW	None Retain:	1	~ 4095	unsigned long	4 bytes	0 to 4,294,967,295
			Retain:	4096	~ 8191			
	Float	F	None Retain:	1	~ 1899	float	4 bytes	3.4E +/- 38
			Retain:	2048	~ 3999			
	Special Type	DB	None Retain:	1	~ 49			
	Message	MSG	Retain:	1	~ 249	30 wchar_t	60 bytes	30 unicode char

The following registers are already reserved for multi-language and virtual keyboard support:

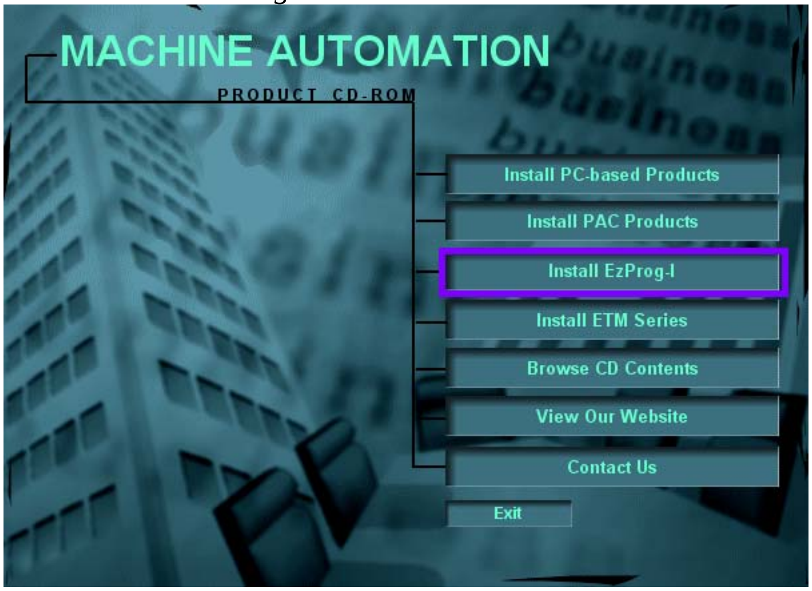
Register Type	Register Number	Note
D	8000	Multilanguage selection
M	16000	Enabling or disabling virtual keyboard support for the ColorEdit Control

2 Development Tools Installation

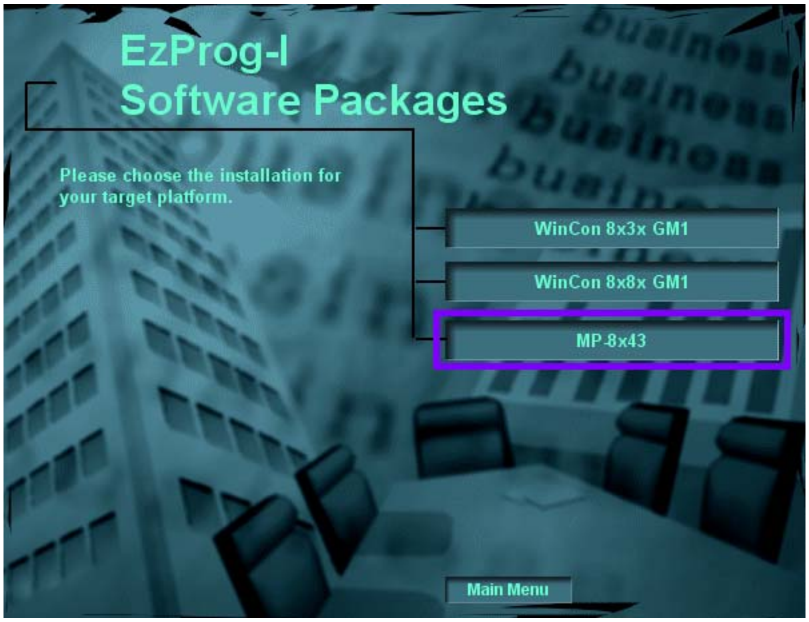
2.1 SDK Installation Guide

Step 1: Insert the "*MACHINE AUTOMATION*" DVD in your DVD drive. The Setup program should start automatically. In case the wizard fails to start double click the "Launch" execution file on the CD.

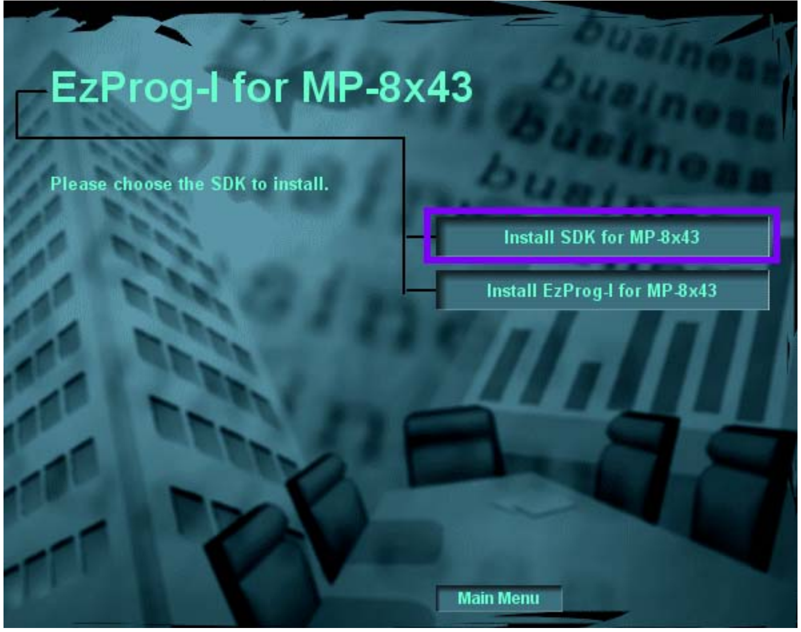
Step 2: Select "*Install EzProg-I*" from the list.



Step 3: Select "*MP-8x43*":



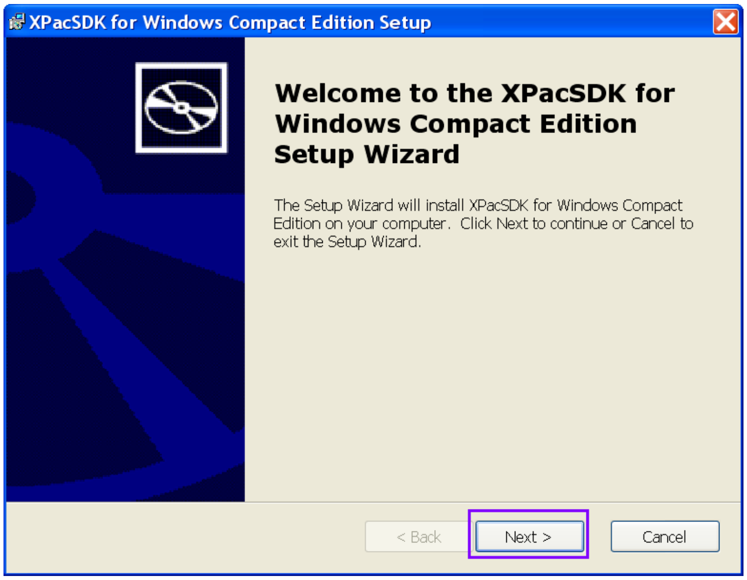
Step 4: Select “*Install SDK for Mp-8X43*”:



EzProg-I for MP-8x43

Please choose the SDK to install.

Step 5: Click ***Next***.

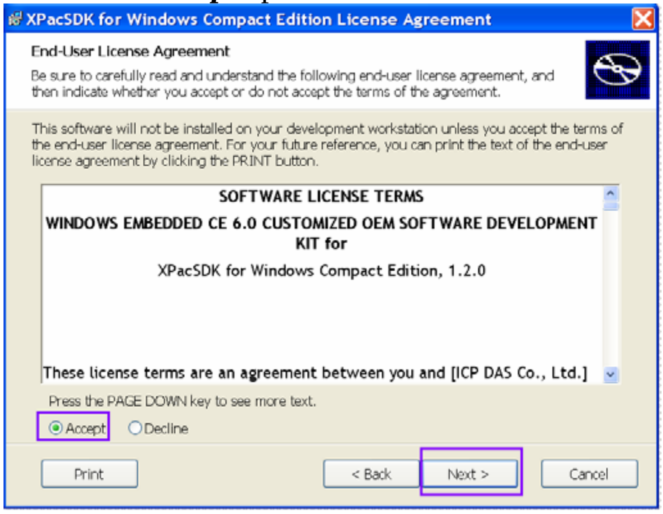


XPacSDK for Windows Compact Edition Setup

Welcome to the XPacSDK for Windows Compact Edition Setup Wizard

The Setup Wizard will install XPacSDK for Windows Compact Edition on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Step 6: Enable the ***Accept*** option and then click ***Next***.



XPacSDK for Windows Compact Edition License Agreement

End-User License Agreement

Be sure to carefully read and understand the following end-user license agreement, and then indicate whether you accept or do not accept the terms of the agreement.

This software will not be installed on your development workstation unless you accept the terms of the end-user license agreement. For your future reference, you can print the text of the end-user license agreement by clicking the PRINT button.

SOFTWARE LICENSE TERMS

WINDOWS EMBEDDED CE 6.0 CUSTOMIZED OEM SOFTWARE DEVELOPMENT KIT for

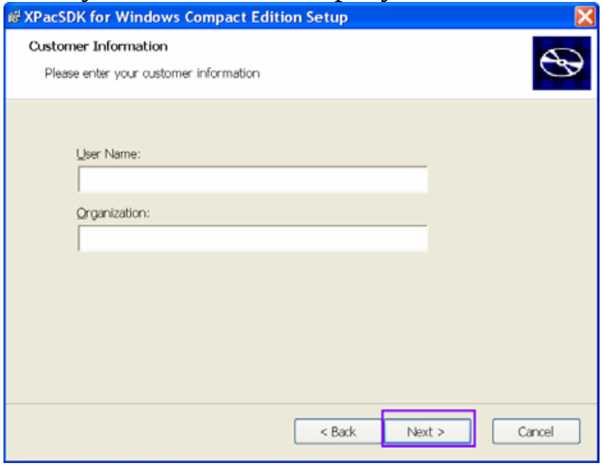
XPacSDK for Windows Compact Edition, 1.2.0

These license terms are an agreement between you and [ICP DAS Co., Ltd.]

Press the PAGE DOWN key to see more text.

● Accept ○ Decline

Step 7: Enter your name and company name and click ***Next***.



XPacSDK for Windows Compact Edition Setup

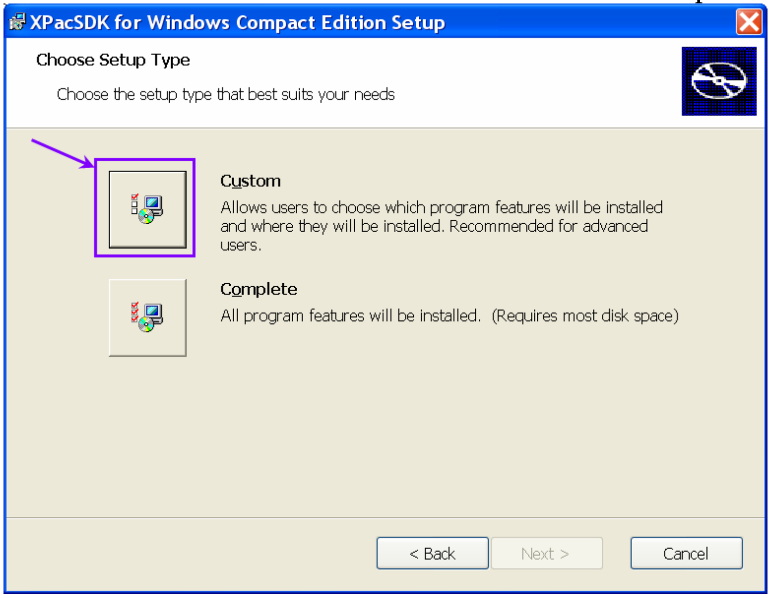
Customer Information

Please enter your customer information

User Name:

Organization:

Step 8: Click the "***Custom***" button. Do NOT install the "Complete" program.



XPacSDK for Windows Compact Edition Setup

Choose Setup Type

Choose the setup type that best suits your needs

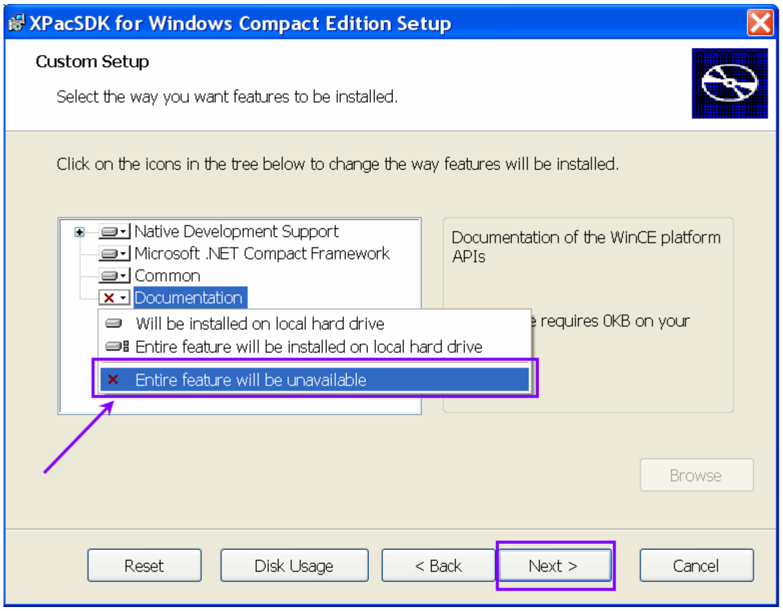
Custom

Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

Complete

All program features will be installed. (Requires most disk space)

Step 9: Disable the ***Documentation*** feature and click ***Next***.



XPacSDK for Windows Compact Edition Setup

Custom Setup

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

Native Development Support

Microsoft .NET Compact Framework

Common

Documentation

Will be installed on local hard drive

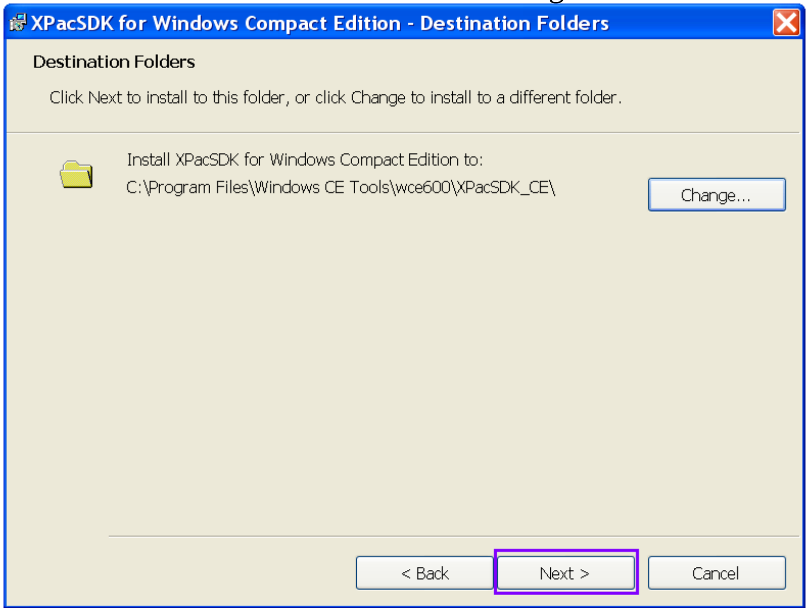
Entire feature will be installed on local hard drive

Entire feature will be unavailable

Documentation of the WinCE platform APIs

e requires 0KB on your

Step 10: Click ***Next***. It is recommended not to change the destination folder.

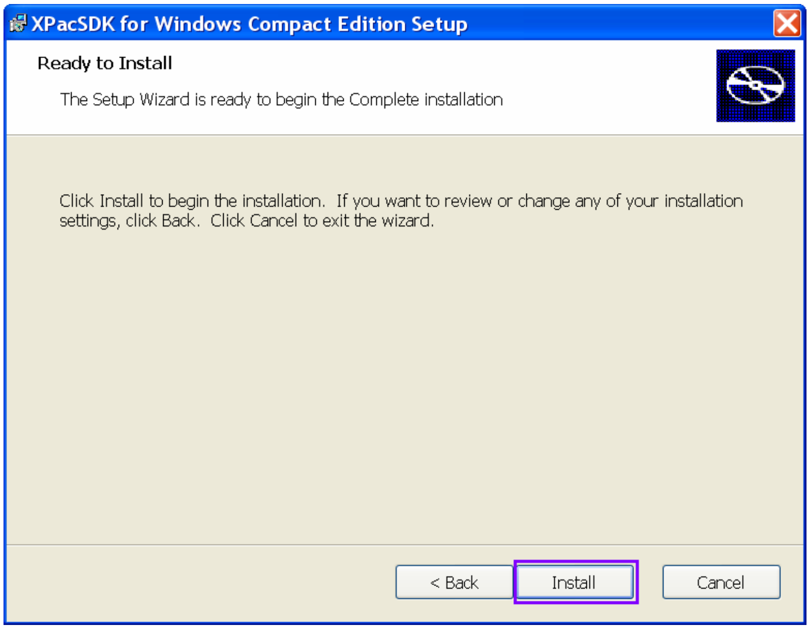

XPacSDK for Windows Compact Edition - Destination Folders

Destination Folders

Click Next to install to this folder, or click Change to install to a different folder.

Install XPacSDK for Windows Compact Edition to:
C:\Program Files\Windows CE Tools\wce600\XPacSDK_CE\

Step 11: Click "***Install***".


XPacSDK for Windows Compact Edition Setup

Ready to Install

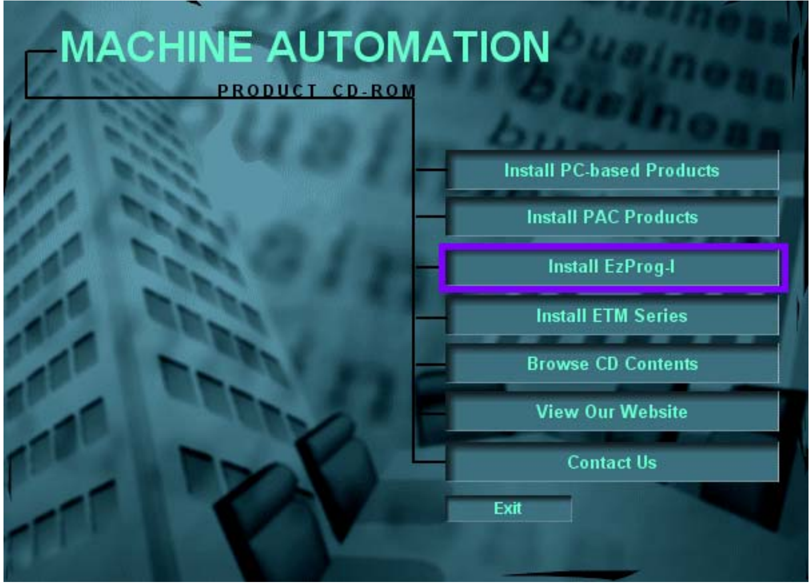
The Setup Wizard is ready to begin the Complete installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

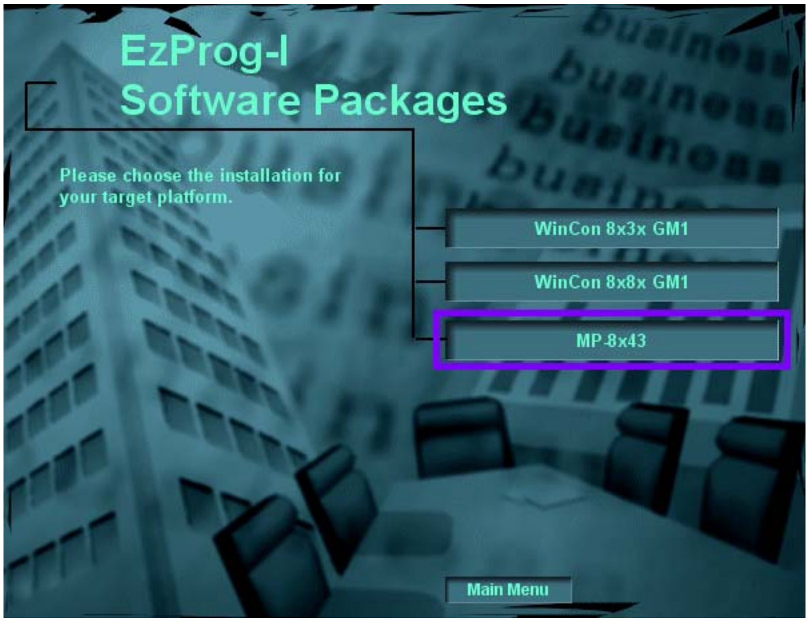
2.2 Installation of EzProg-I Development Resources

Step 1: Insert the "*MACHINE AUTOMATION*" DVD in your DVD drive. The Setup program should start automatically. In case the wizard fails to start double click the "Launch" execution file on the CD.

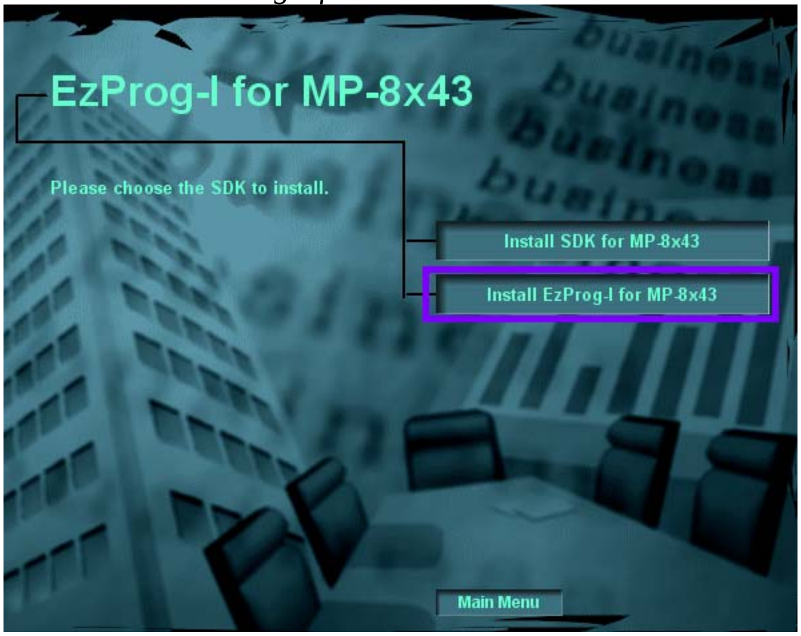
Step 2: Select "*Install EzProg-I*" from the list.



Step 3: Select "*MP-8x43*":

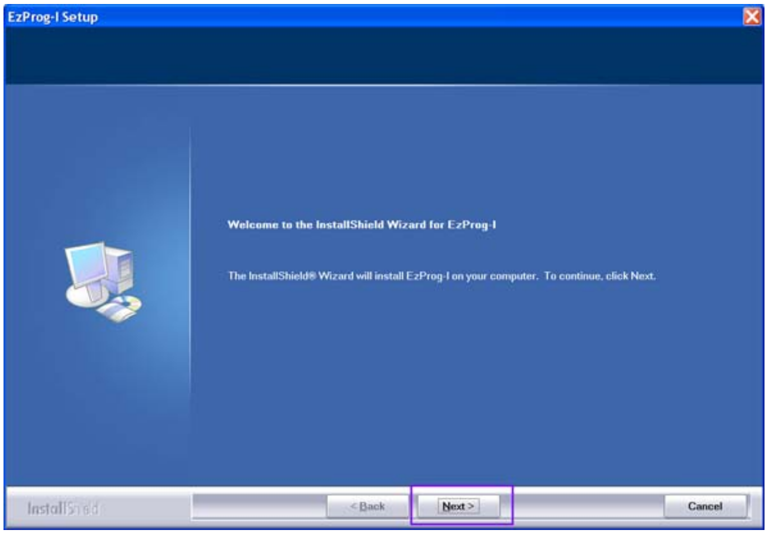


Step 4: Select “*Install EzProg-I for MP-8x43*”:



Please choose the SDK to install.

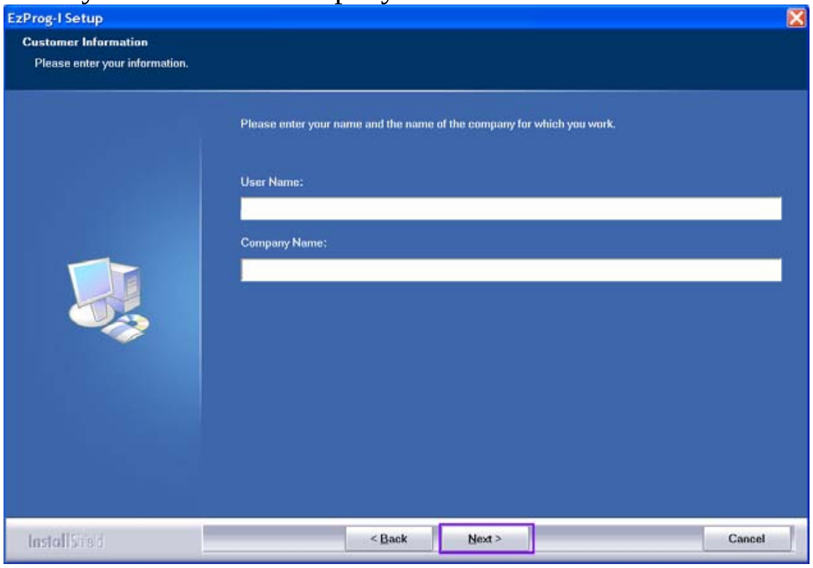
Step 5: Click ***Next***.



Welcome to the InstallShield Wizard for EzProg-I

The InstallShield® Wizard will install EzProg-I on your computer. To continue, click Next.

Step 6: Enter your name and company name and click ***Next***.



EzProg-I Setup

Customer Information

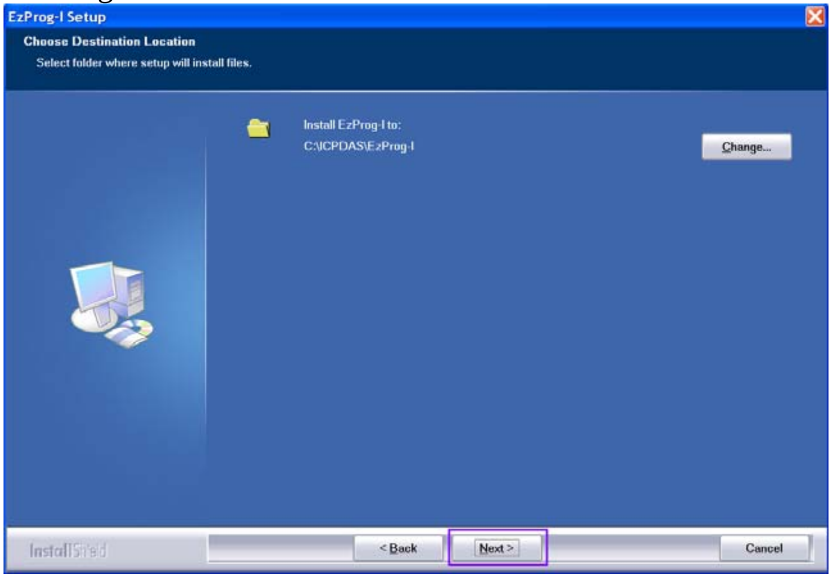
Please enter your information.

Please enter your name and the name of the company for which you work.

User Name:

Company Name:

Step 7: Click ***Next***. To guarantee consistency of the system it is recommended not to change the destination folder.



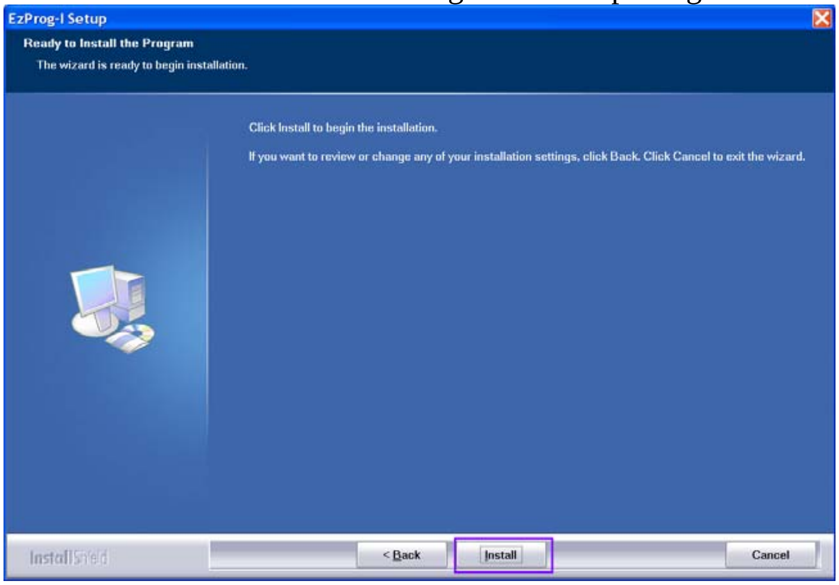
EzProg-I Setup

Choose Destination Location

Select folder where setup will install files.

Install EzProg-I to:
C:\ICPDAS\EzProg-I

Step 8: Click "***Install***" to install the EzProg-I software package.



EzProg-I Setup

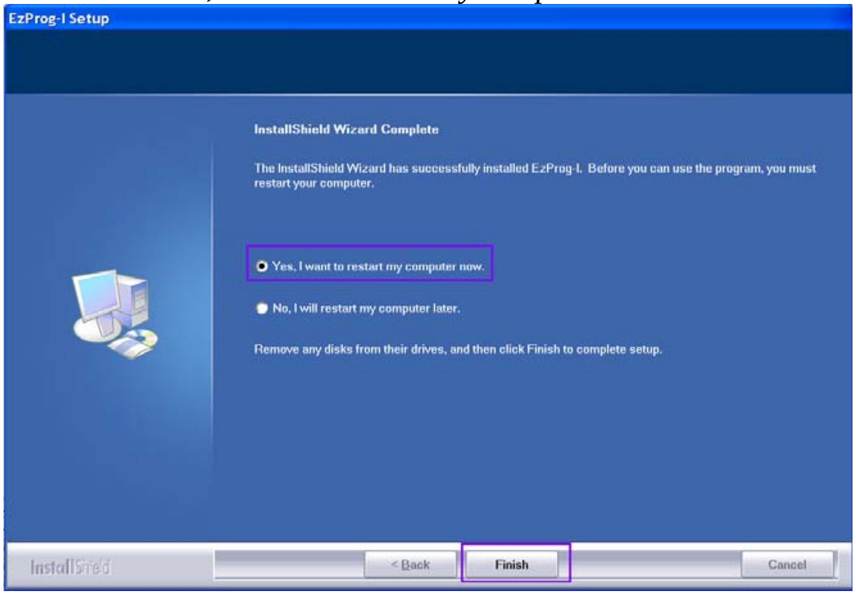
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Step 9: Enable the "*Yes, I want to restart my computer*" and click ***Finish***.



EzProg-I Setup

InstallShield Wizard Complete

The InstallShield Wizard has successfully installed EzProg-I. Before you can use the program, you must restart your computer.

● Yes, I want to restart my computer now.

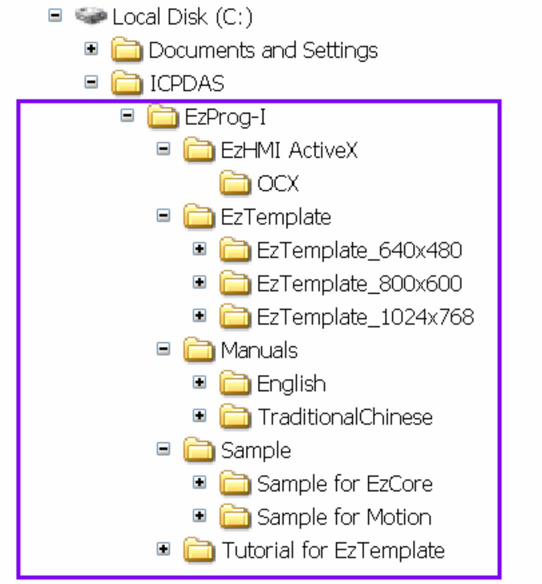
○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

After computer has restarted the EzProg-I software resources are ready to use.

2.3 EzProg-I Development Resources

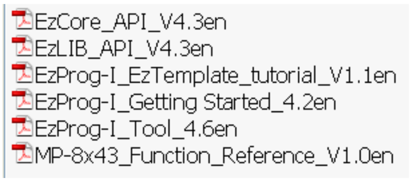
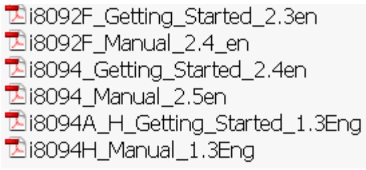
After installing of the EzProg-I resources the required manuals, ActiveXs and examples are installed in the *C:\ICPDAS\EZPROG-I* directory.



```
Local Disk (C:)
    Documents and Settings
    ICPDAS
        EzProg-I
            EzHMI ActiveX
                OCX
            EzTemplate
                EzTemplate_640x480
                EzTemplate_800x600
                EzTemplate_1024x768
            Manuals
                English
                TraditionalChinese
            Sample
                Sample for EzCore
                Sample for Motion
            Tutorial for EzTemplate
```

2.3.1 Manuals

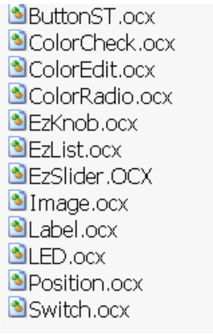
Manuals for EzProg-I are located in the following directory:
C:\ICPDAS\EZPROG-I\Manuals

Manuals	Directory
 EzCore_API_V4.3en EzLIB_API_V4.3en EzProg-I_EzTemplate_tutorial_V1.1en EzProg-I_Getting Started_4.2en EzProg-I_Tool_4.6en MP-8x43_Function_Reference_V1.0en	***C:\ICPDAS\EZPROG-I\Manuals\English***
 i8092F_Getting_Started_2.3en i8092F_Manual_2.4_en i8094_Getting_Started_2.4en i8094_Manual_2.5en i8094A_H_Getting_Started_1.3Eng i8094H_Manual_1.3Eng	***C:\ICPDAS\EZPROG-I\Manuals\English\Motion***

2.3.2 ActiveX

EzHMI ActiveXs for developing graphic user interfaces are located in the following directory:

C:\ICPDAS\EzProg-I\EzHMI ActiveX\OCX

EzHMI ActiveX	Directory
 ButtonST.ocx ColorCheck.ocx ColorEdit.ocx ColorRadio.ocx EzKnob.ocx EzList.ocx EzSlider.OCX Image.ocx Label.ocx LED.ocx Position.ocx Switch.ocx	***C:\ICPDAS\EzProg-I\EzHMI ActiveX\OCX***

2.3.3 EzTemplate

The EzTemplate provides the following default settings:

- All the basic EzProg-I settings are already done.
- The project includes 50 dialog pages which can be used as HMI pages. The programmer may select any of these pages for his HMI configuration.
- ButtonST control supports the viewing of the different HMI dialog pages during runtime
- Programming interfaces for user thread and RTSR. Eight user threads and real time service routine (multitasking) are available to run the control program in a multitasking and deterministic environment.
- Dialog pages for the following resolution are provided:
    - 640x480
    - 800x600
    - 1024x760

  The EzTemplate projects with the different dialog resolution are located in the following directory:

  C:\ICPDAS\EzProg-I\EzTemplate

  Tutorials for the EzTemplate are provided in the following directory:

  C:\ICPDAS\EzProg-I\Tutorial for EzTemplate

2.3.4 Examples

Ample examples are provided to demonstrate the usage of the EzCore and motion APIs:

C:\ICPDAS\EzProg-I\Sample\Sample for EzCore
C:\ICPDAS\EzProg-I\Sample\Sample for Motion

3 EzTemplate Applications

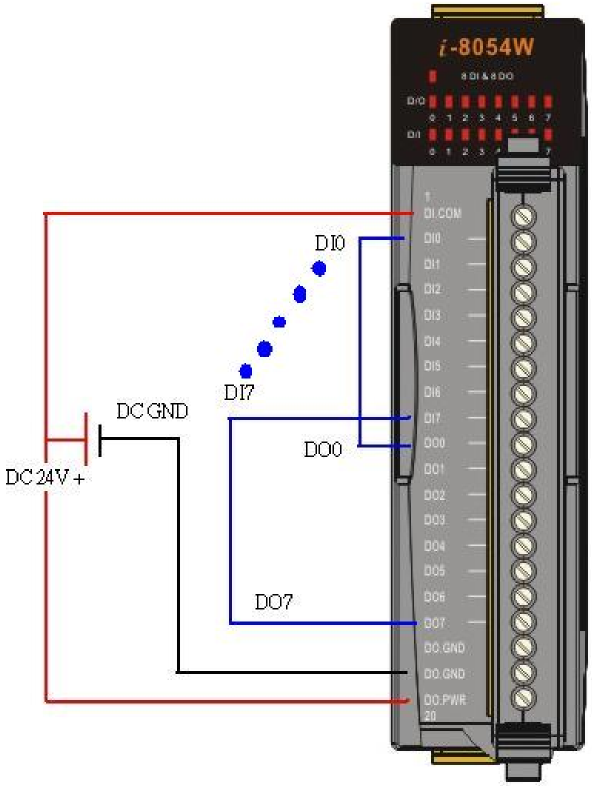
3.1 Creating a simple user interface for monitoring digital IO statuses

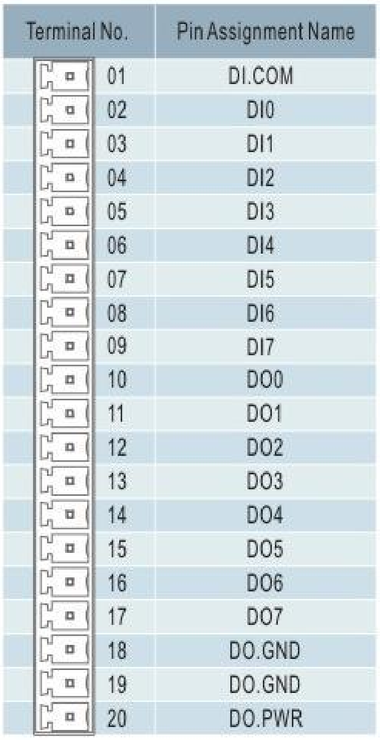
This example shows how to use the EzTemplate to link an EzHMI ActiveX object directly to digital IO channels without writing any code.

3.2 Procedures on the MPAC

3.2.1 Hardware configuration

For this demonstration the module i-8054W with eight digital input and eight digital output channels is being used. Plug the module in the first slot of the MPAC device and implement the wiring as shown in the following picture. Connect each digital output channel with the respective digital input channel counterpart. For example connect the first digital input channel (DI0) to the first digital output channel (DO0) and the second digital input channel (DI1) to the second digital output channel (DO1) and so forth.





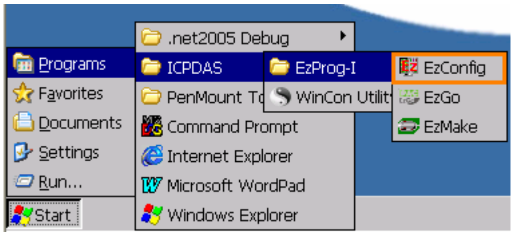
Terminal No.	Pin Assignment Name
01	DI.COM
02	DI0
03	DI1
04	DI2
05	DI3
06	DI4
07	DI5
08	DI6
09	DI7
10	DO0
11	DO1
12	DO2
13	DO3
14	DO4
15	DO5
16	DO6
17	DO7
18	DO.GND
19	DO.GND
20	DO.PWR

The modules supported by EzProg-I are listed on the following website:
http://www.icpdas.com/products/motion/EzProg-I_i8K.htm

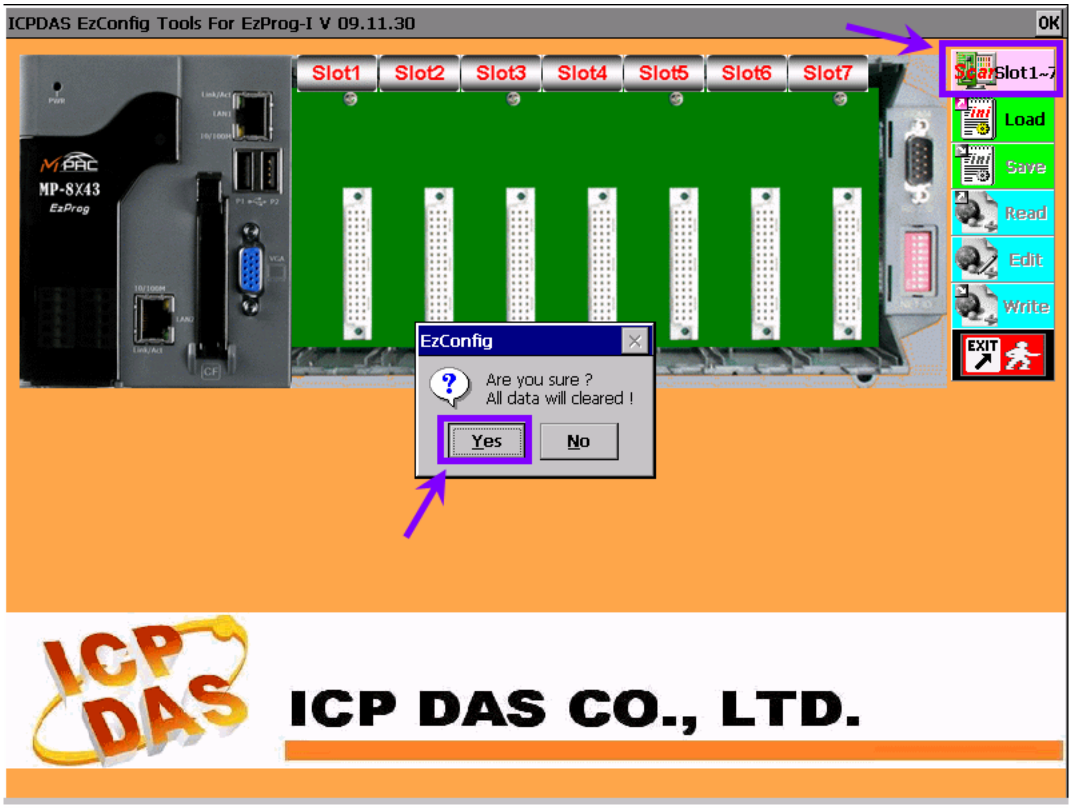
3.2.2 IO register mapping

Map the individual digital channels to IO registers as follows:

Step 1: Start the EzConfig utility on the MPAC:
Start →Program→ICPDAS→EzProg-I→EzConfig



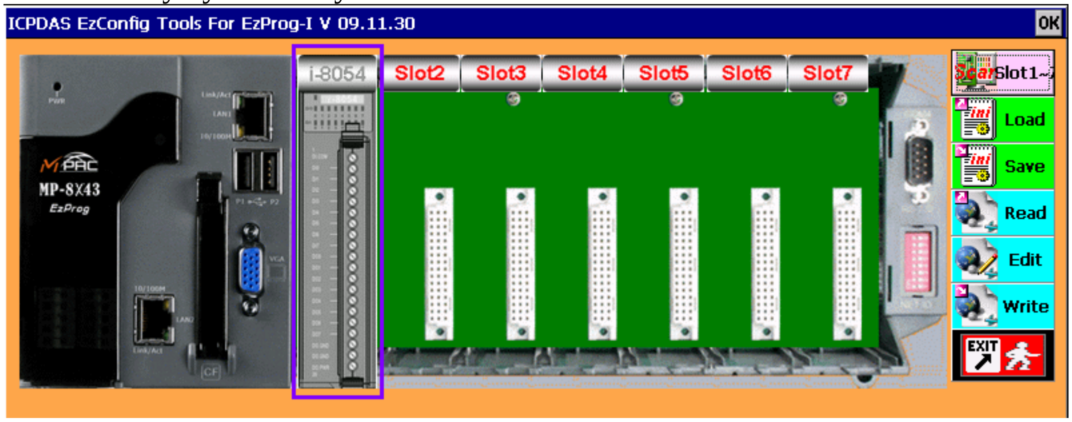
Step 2: Scan the slots to detect the digital module. Click the “Scan Slot 1~7” button and confirm your decision by clicking the “Yes” button on the message box.



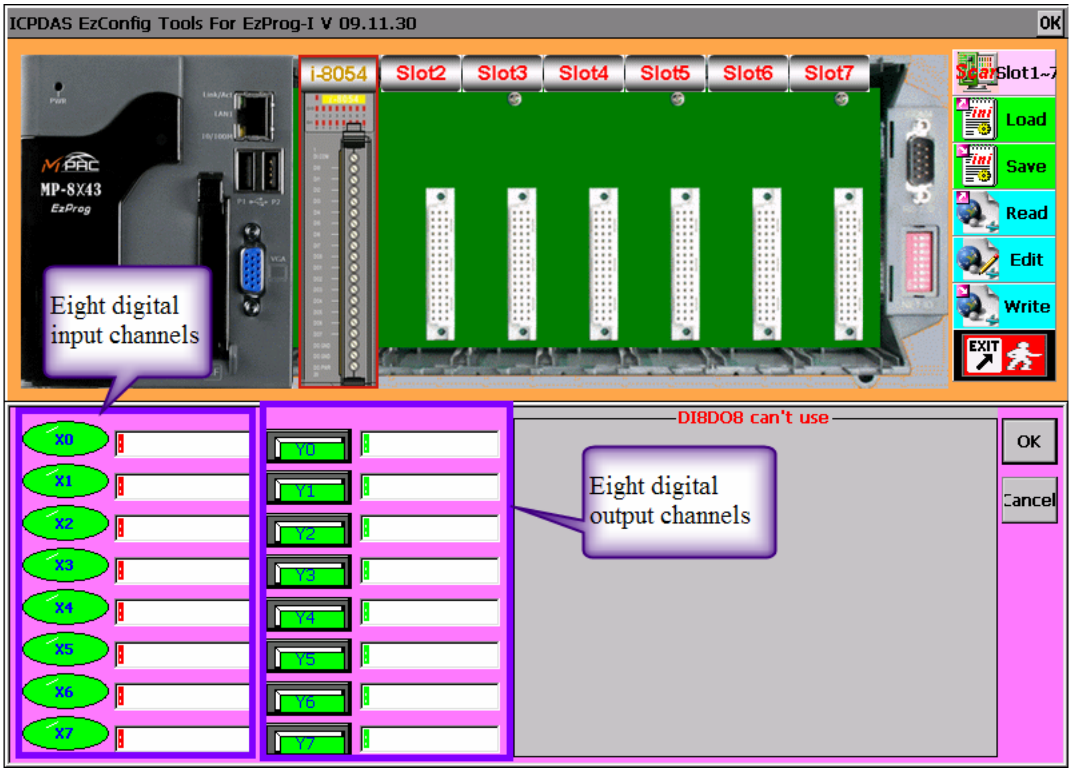
EzConfig

Are you sure ?
All data will cleared !

If the scanning process has been successful the module is displayed in the first slot together with its names. The IO register mapping is being done automatically by the utility as soon as the module has been identified.



Step 3: Click the module to display the register mapping:



The digital input channels are mapped to the registers X0 to X7 and the digital output channels are mapped to the registers Y0 to Y7. The statuses of the digital output channels can be directly changed by clicking the push buttons Y0 representing the channels.

Step 4: Click the "OK" button to close the register mapping window.

Step 5: Press "Save" to save the IO register mapping and "Exit" to close the program.

Notice: It is required to close the EzConfig utility in order for a program running on the EzCore engine to work properly.

3.3 Procedures on the PC

The EzTemplate is a template for Visual Studio 2008 to enable the user which is neither familiar with MFC nor Visual Studio 2008 to easily create a Window CE 6.0 application. The EzTemplate provides the user with all the necessary programming interfaces to start writing a real time logic controller with a human machine interfaces. Direct DIO control via the user interface, for example for displaying or setting a digital channel status, can be done without any programming.

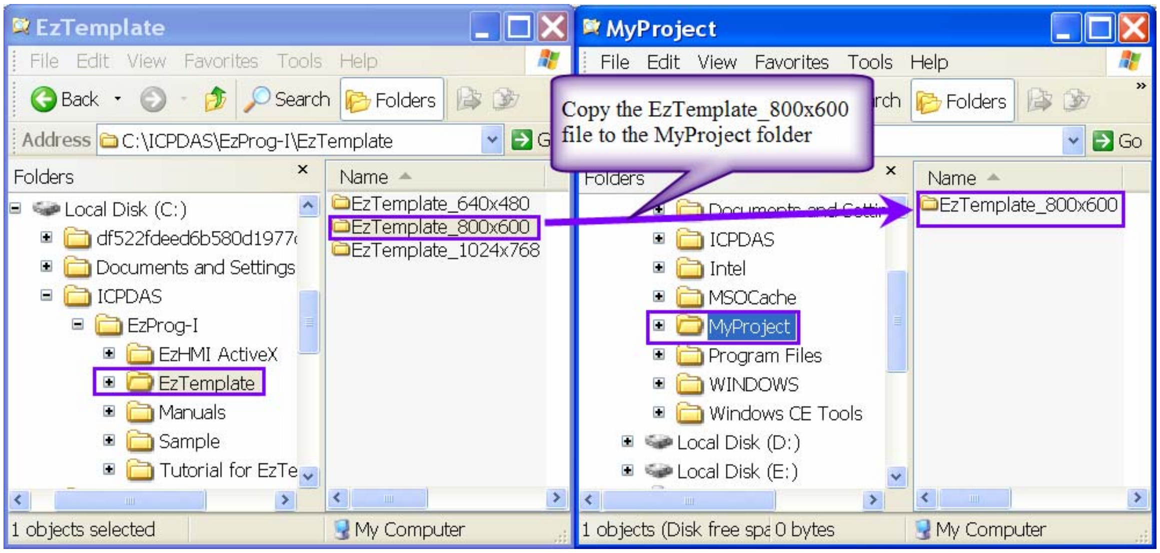
3.3.1 EzTemplate

All the necessary libraries and header files are included in the project for the system designer to make full use of the power of EzHMI controls. Dialog pages for the following screen resolutions are provided:

- 640x480
- 800x600
- 1024x760

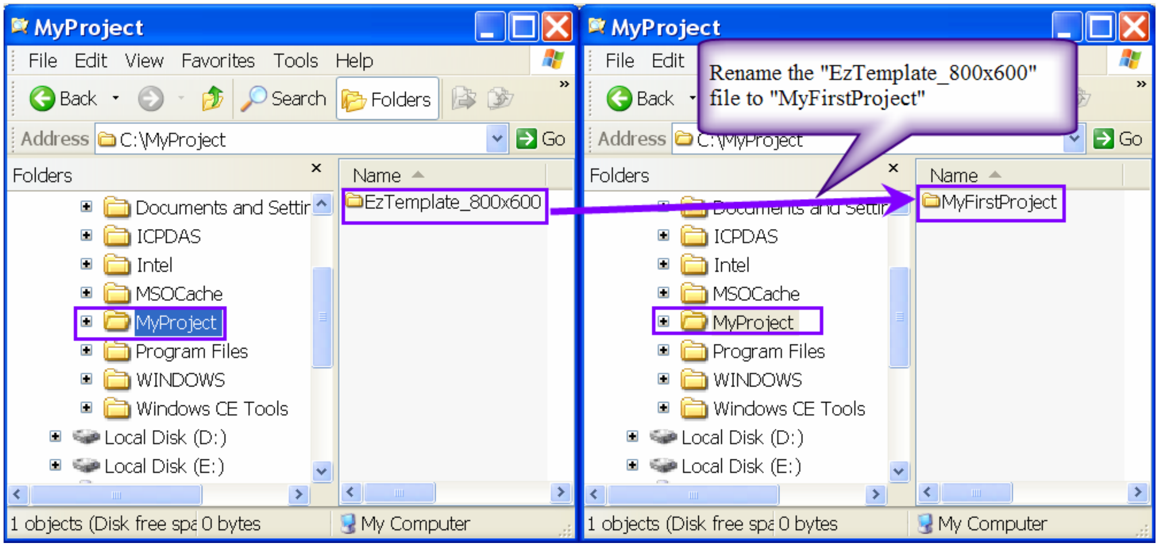
3.3.2 Using EzTemplate

Step 1: Copy the EzTemplate file from the directory ***C:\ICPDAS\EzProg-I\EzTemplate\EzTemplate_800x600*** to any directory of you choice. In this example the file is copied to the newly created "MyProject" directory.



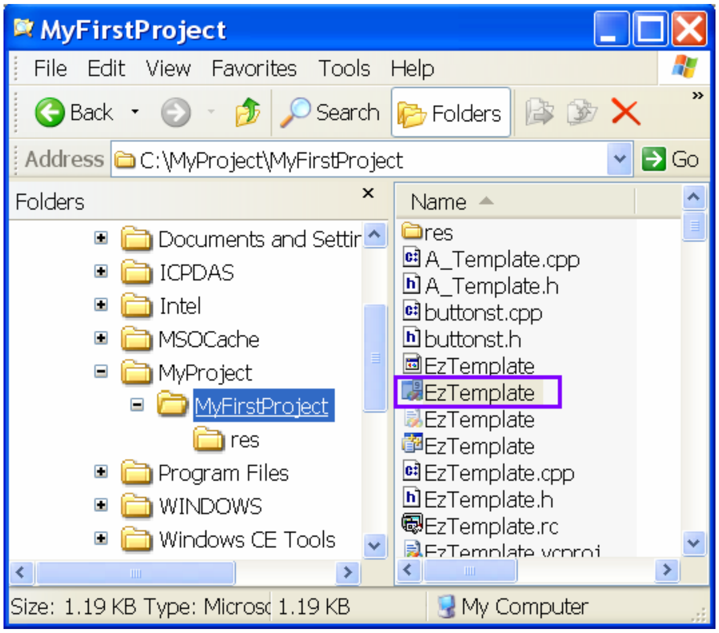
Copy the EzTemplate_800x600 file to the MyProject folder

Step 2: Rename the file "EzTemplate_800x600" to "MyFirstProject".



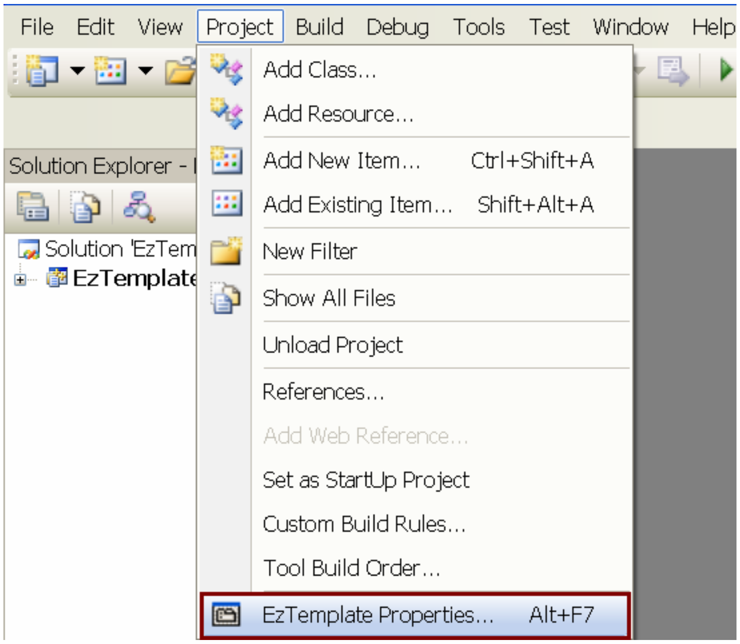
Rename the "EzTemplate_800x600" file to "MyFirstProject"

Step 3: Open the project file with Visual Studio 2008 by double clicking the solution file “EzTemplate.sln”

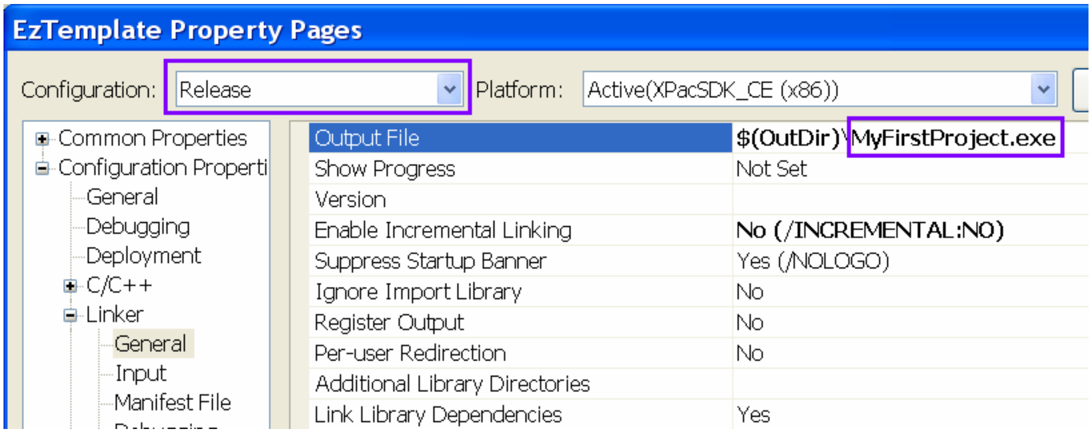


Step 4: Rename the execution file.

Open the Property Page: ***Project → EzTemplate Properties…***



Configuration Properties→ Linker→General
Select the "Release" configuration and change the output file to ***"$(OutDir)\MyFirstProject.exe***" as shown below and click "Apply".
Select the "Debug" configuration and change the output file to ***"$(OutDir)\MyFirstProject.exe***" and click "Apply".



Configuration: Release
Platform: Active(XPacSDK_CE (x86))

Output File	$(OutDir)\MyFirstProject.exe
Show Progress	Not Set
Version	
Enable Incremental Linking	No (/INCREMENTAL:NO)
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	
Link Library Dependencies	Yes

Step 5: Close the dialog by clicking "***OK***".

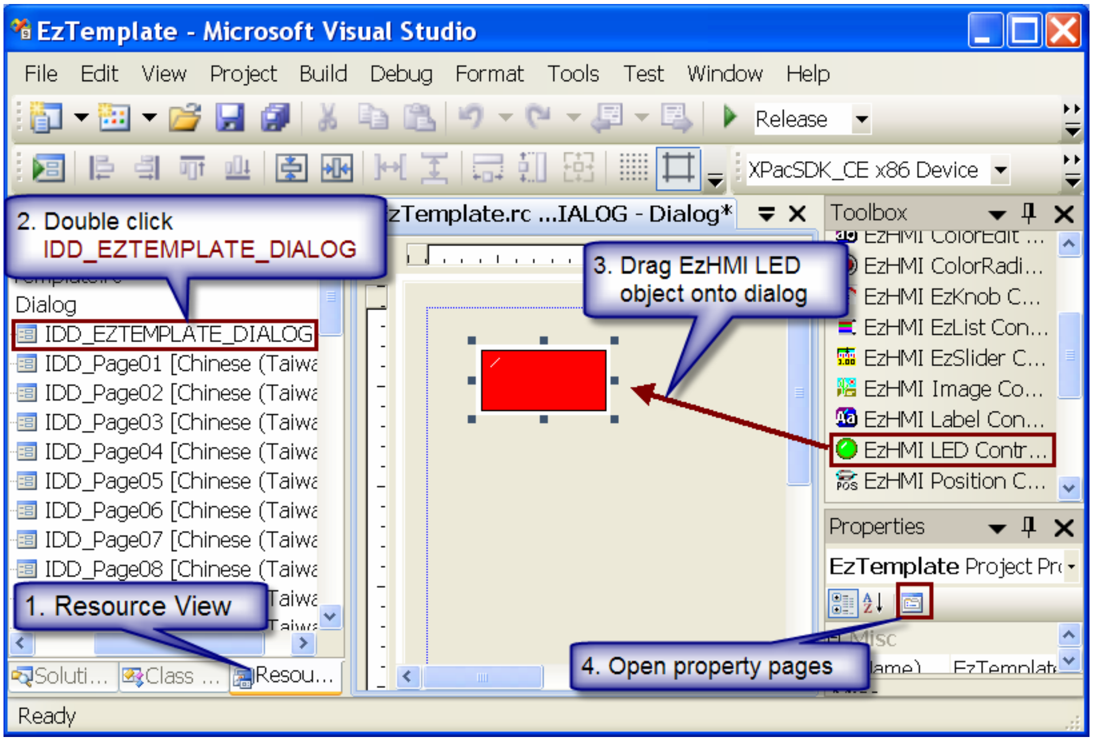
3.3.3 HMI design and IO register linking

Step 1: To add the EzHMI ActiveX controls to the Visual Studio toolbox follow the instructions described in appendix 4.1 at the end of this manual.

Step 2: Open the IDD_EZTEMPLATE_DIALOG window.

Step 3: Drag an EzHMI LED object onto the main dialog window.

Step 4: Open the property pages: Right click the LED object and select “Properties”. Click the “Property Pages” icon in the “Properties” window.



1. Resource View

2. Double click IDD_EZTEMPLATE_DIALOG

3. Drag EzHMI LED object onto dialog

4. Open property pages

EzTemplate - Microsoft Visual Studio
File Edit View Project Build Debug Format Tools Test Window Help
Release
XPacSDK_CE x86 Device
zTemplate.rc ...IALOG - Dialog*
Toolbox
Dialog
IDD_EZTEMPLATE_DIALOG
IDD_Page01 [Chinese (Taiwa
IDD_Page02 [Chinese (Taiwa
IDD_Page03 [Chinese (Taiwa
IDD_Page04 [Chinese (Taiwa
IDD_Page05 [Chinese (Taiwa
IDD_Page06 [Chinese (Taiwa
IDD_Page07 [Chinese (Taiwa
IDD_Page08 [Chinese (Taiwa
EzHMI ColorEdit ...
EzHMI ColorRadi...
EzHMI EzKnob C...
EzHMI EzList Con...
EzHMI EzSlider C...
EzHMI Image Co...
EzHMI Label Con...
EzHMI LED Contr...
EzHMI Position C...
Properties
EzTemplate Project Pr
Misc
EzTemplate
Soluti... Class ... Resou...
Ready

Make the following property settings:

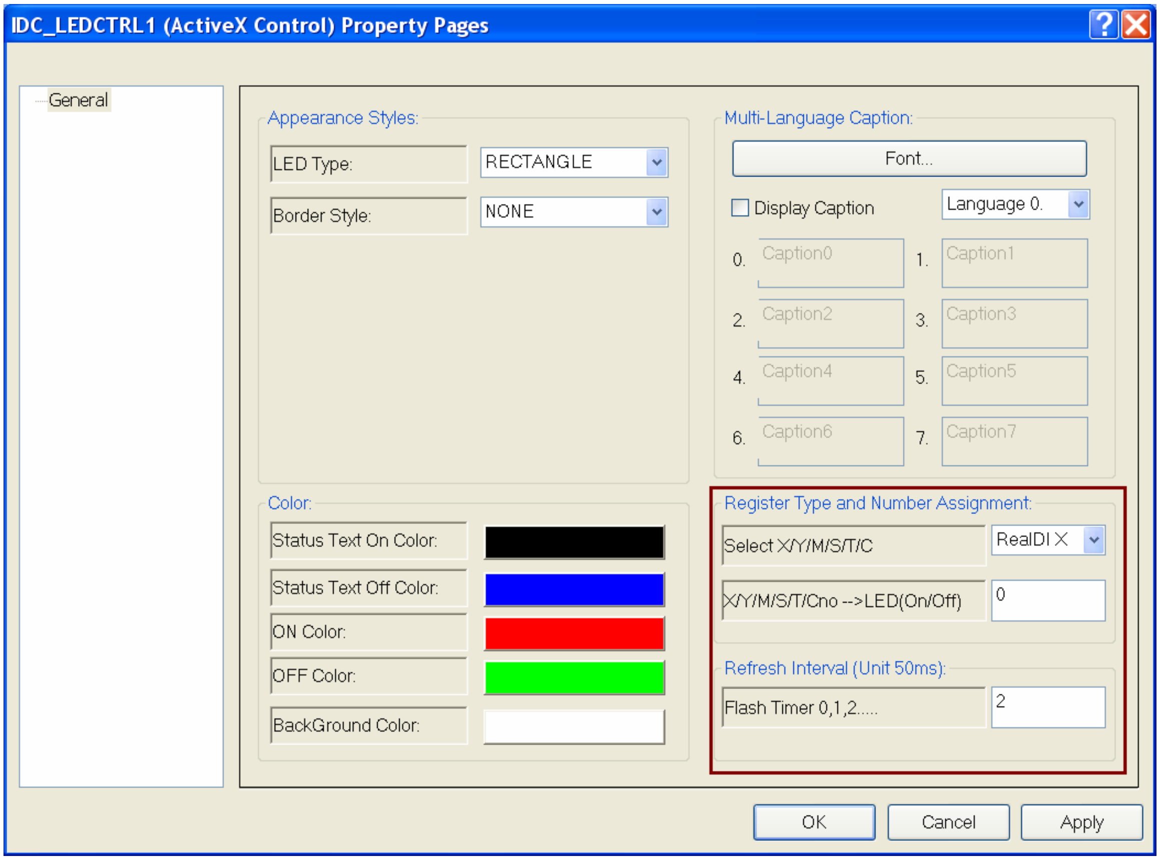
Step 5: Link the LED object to the X (digital input) register number 0.

- Register type: Select “RealDI X” for “Select X/Y/M/S/T/C”
- Register number: Assign zero to “X/Y/M/S/T/Cno→LED(On/Off)”

Step 6: Set the LED object refresh rate to 100 milliseconds.

- Assign “Flash Timer 0,1,2…” the value 2.

Step 7: Close the property page by clicking “OK”



Appearance Styles:

LED Type: RECTANGLE

Border Style: NONE

Color:

Status Text On Color: ■

Status Text Off Color: 🟦

ON Color: 🟥

OFF Color: 🟩

BackGround Color: □

Multi-Language Caption:

□ Display Caption Language 0.

0. Caption0 1. Caption1

2. Caption2 3. Caption3

4. Caption4 5. Caption5

6. Caption6 7. Caption7

Register Type and Number Assignment:

Select X/Y/M/S/T/C RealDI X

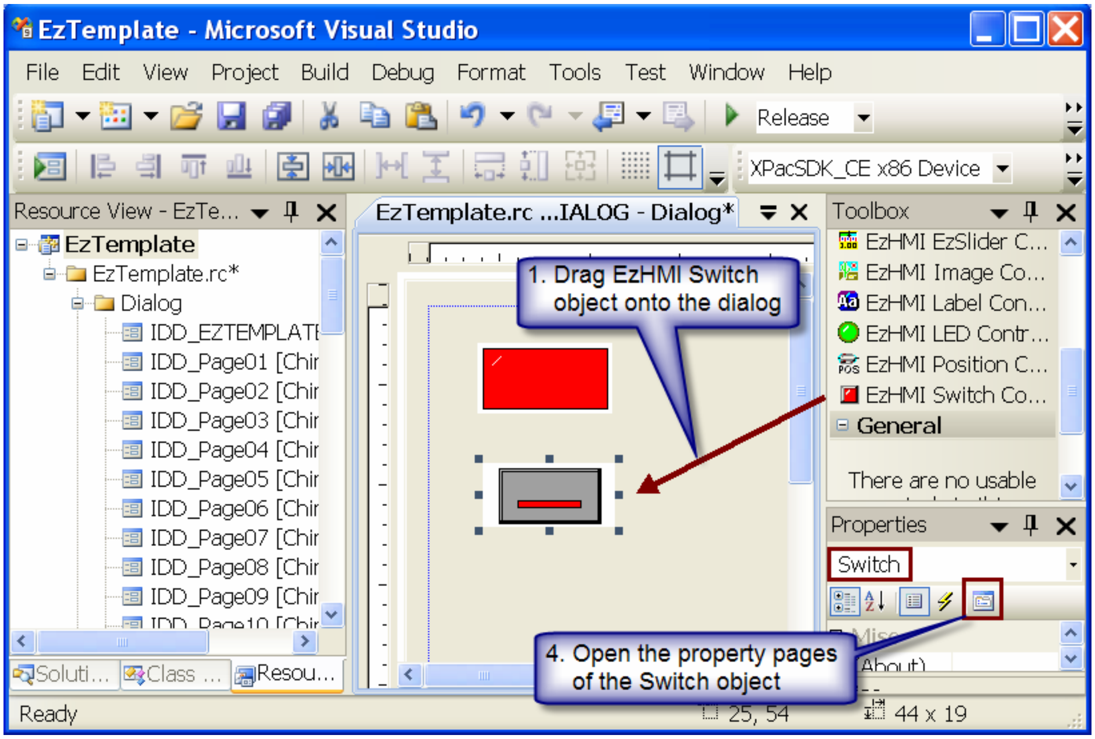
X/Y/M/S/T/Cno -->LED(On/Off) 0

Refresh Interval (Unit 50ms):

Flash Timer 0,1,2..... 2

Step 8: Drag an EzHMI SWITCH object onto the main dialog window.

Step 9: Open the property pages: Right click the SWITCH object and select "Properties".
Click the "Property Pages" icon in the "Properties" window.



1. Drag EzHMI Switch object onto the dialog

4. Open the property pages of the Switch object

Make the following property settings:

Step 10: Choose “SWITCH TOGGLE” as switch style.

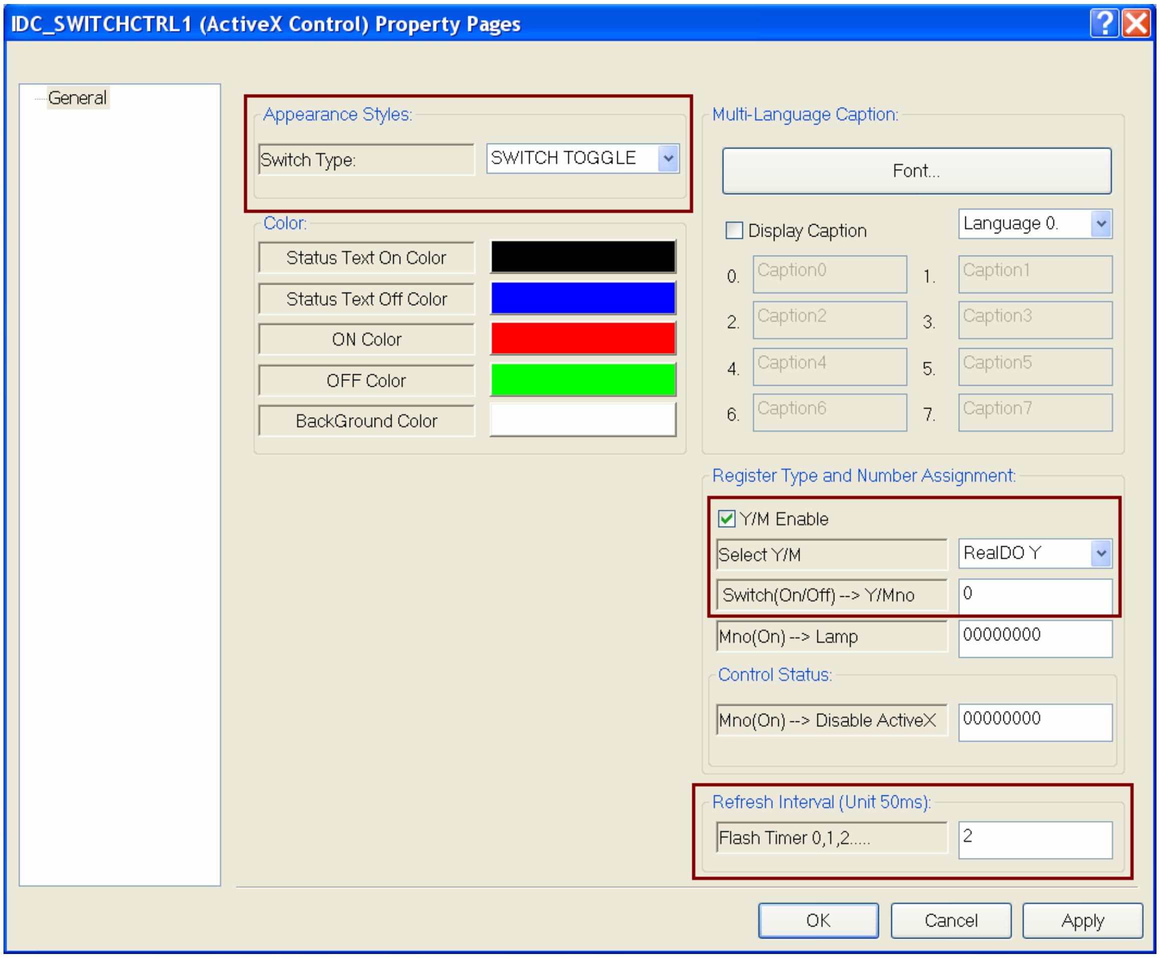
Step 11: Link the Switch object to the Y (digital output) register number 0.

- Register type: Select “RealDO Y” for “Select Y/M”
- Register number: Assign zero to “Switch (On/Off)→Y/Mno”

Step 12: Set the Switch object refresh rate to 100 milliseconds.

- Assign “Flash Timer 0,1,2…” the value 2.

Step 13: Close the property page by clicking “OK”



IDC_SWITCHCTRL1 (ActiveX Control) Property Pages

Appearance Styles:

Switch Type:	SWITCH TOGGLE

Color:

Status Text On Color	⬛
Status Text Off Color	🟦
ON Color	🟥
OFF Color	🟩
BackGround Color	⬜

Multi-Language Caption:

☐ Display Caption   Language 0.

0.	Caption0	1.	Caption1
2.	Caption2	3.	Caption3
4.	Caption4	5.	Caption5
6.	Caption6	7.	Caption7

Register Type and Number Assignment:

☑ Y/M Enable

Select Y/M	RealDO Y
Switch(On/Off) --> Y/Mno	0
Mno(On) --> Lamp	00000000

Control Status:

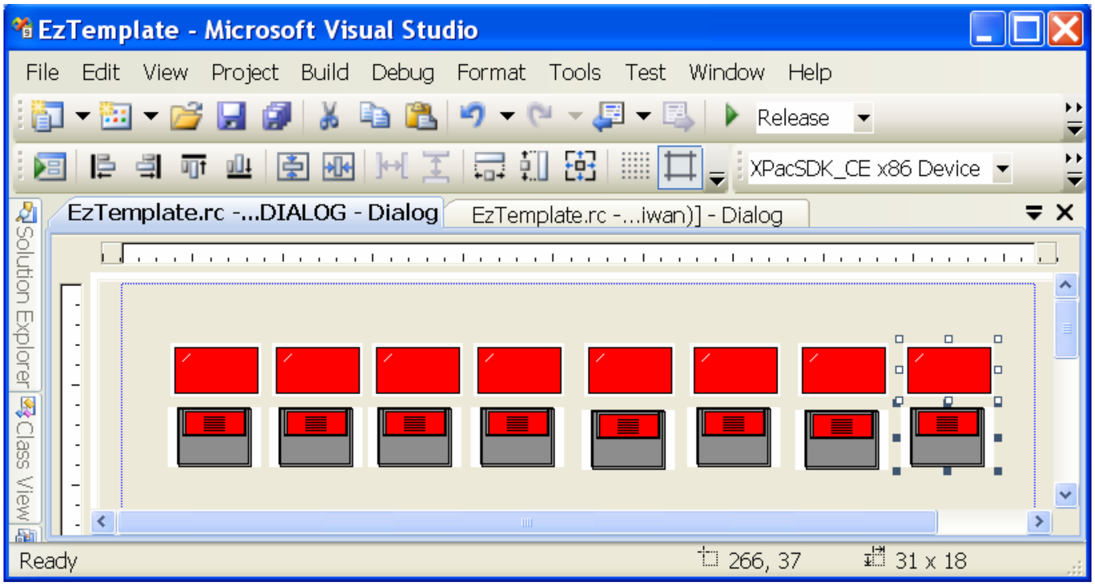
Mno(On) --> Disable ActiveX	00000000

Refresh Interval (Unit 50ms):

Flash Timer 0,1,2.....	2

Step 14: Copy and paste both LED and Switch objects eight times:

- Select both objects at once:
  - i. position the mouse pointer at the upper left corner of the objects
  - ii. hold left mouse button down and drag a rectangle over the LED and Switch objects
  - iii. release the mouse button
- Copy the objects (Ctrl+C)
- Paste the objects (Ctrl+V)


EzTemplate - Microsoft Visual Studio
File Edit View Project Build Debug Format Tools Test Window Help
Release
XPacSDK_CE x86 Device
EzTemplate.rc -...DIALOG - Dialog
EzTemplate.rc -...iwan)] - Dialog
Solution Explorer
Class View
Ready
266, 37
31 x 18

Open the property sheets for each copied object and make the following changes:

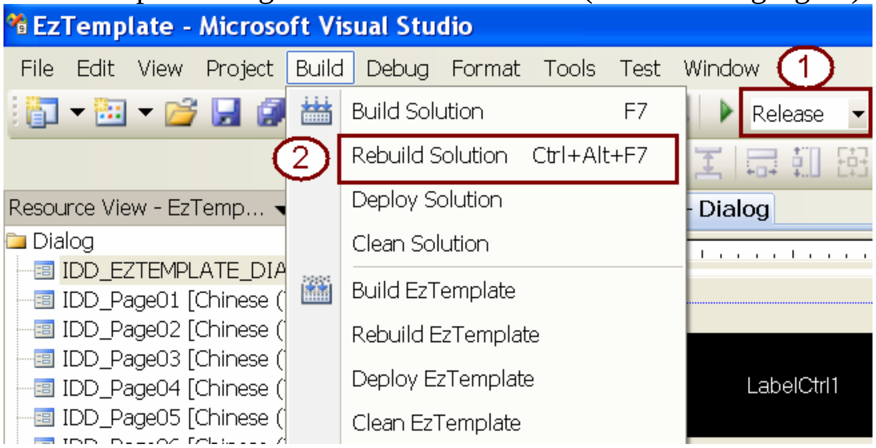
Step 15: Link the copied LED object in ascending sequence to the X register number 1 to 7.
Assign the first copied LED the register number 1 in the "X/Y/M/S/T/Cno→LED(On/Off)" text box and the second copied LED the number 2, etc.

Step 16: Link the copied Switch objects in ascending sequence to the Y (digital output) register number 1 to 7. Enter the register number 1 for the first copied Switch object in "Switch (On/Off)→Y/Mno" edit box and for the second Switch object the register number 2, etc.

Compile the project and download the execution file:

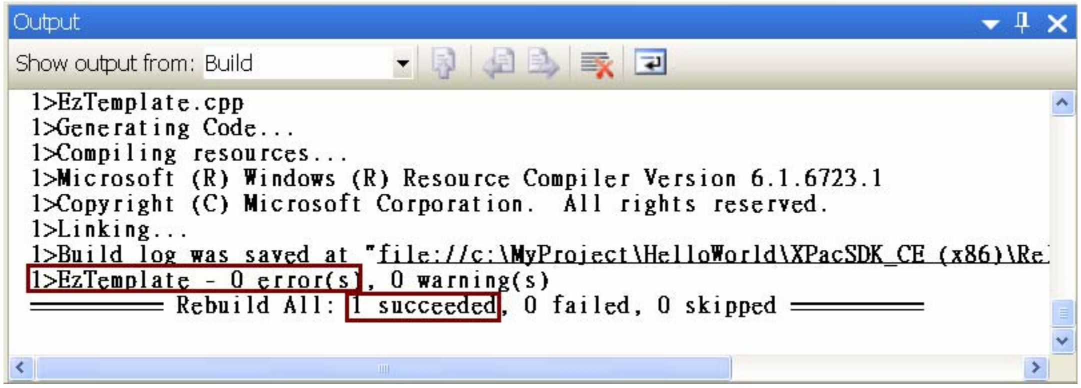
Creating an execution file:

Step 17: Set the compile configuration to release mode (see following figure): Release (1)



Step 18: Compile the project: Build→Rebuild Solution

If the building of the execution file was successful the VS2008 output window will display: “0 error(s)”



Output

Show output from: Build

```
1>EzTemplate.cpp
1>Generating Code...
1>Compiling resources...
1>Microsoft (R) Windows (R) Resource Compiler Version 6.1.6723.1
1>Copyright (C) Microsoft Corporation.  All rights reserved.
1>Linking...
1>Build log was saved at "file://c:\MyProject\HelloWorld\XPacSDK_CE (x86)\Re
1>EzTemplate - 0 error(s), 0 warning(s)
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

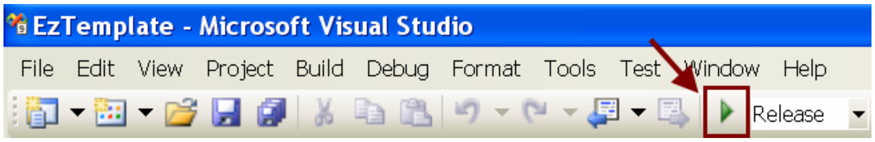
Connect to the target device:

Step 19: Follow the instructions described in Appendix 4.2 at the end of this manual.

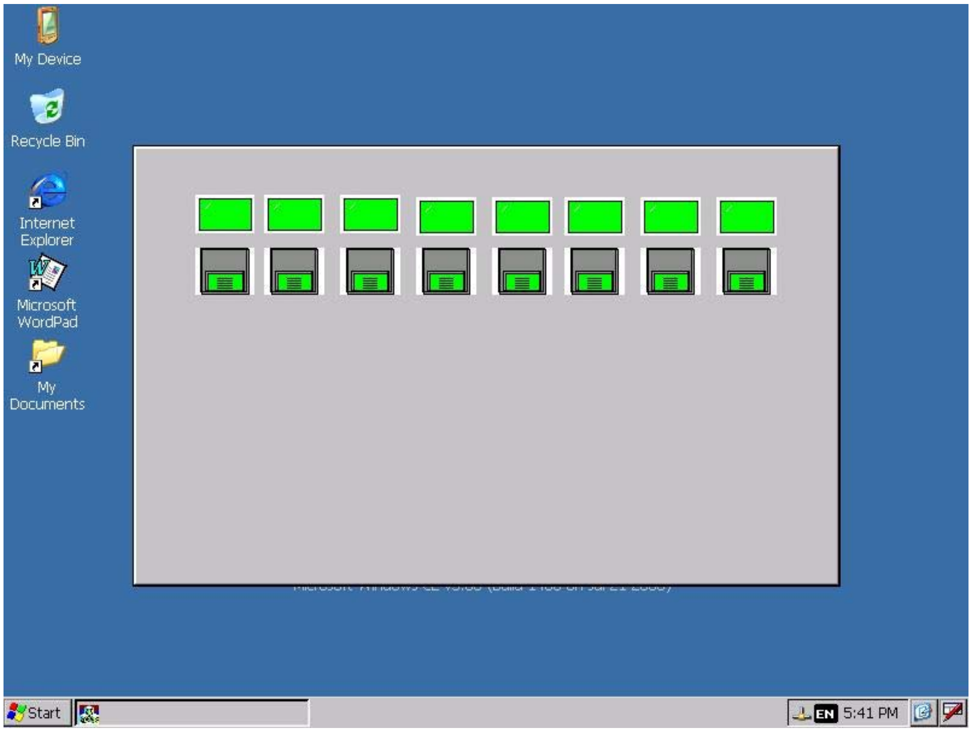
Download to the target device:

The execution file of the compiled project is downloaded with VS2008 as follows:

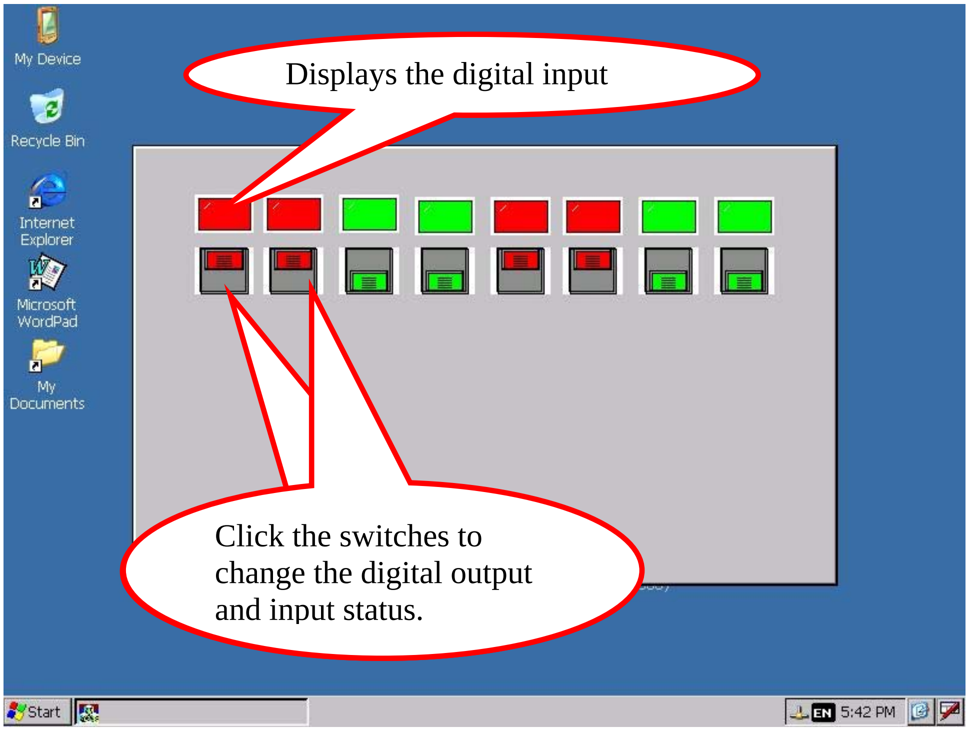
Step 20: Click the green arrow icon as indicated on the next figure.


EzTemplate - Microsoft Visual Studio
File Edit View Project Build Debug Format Tools Test Window Help
Release

The execution file are now downloaded to the following directory:
\Program Files\EzTemplate
After the downloading process has completed the program is being started automatically by VS2008.



Step 21: Click the switches to change the digital output status. This will change the input status of the connected channel.

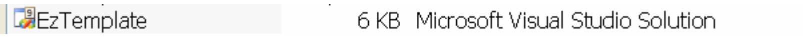

My Device
Recycle Bin
Internet Explorer
Microsoft WordPad
My Documents
Displays the digital input
Click the switches to change the digital output and input status.
Start
EN 5:42 PM

4 Appendix

4.1 Adding EzHMI ActiveX controls to the Visual Studio Toolbox

In order to use the EzHMI controls for the EzTemplate project the EzHMI ActiveX controls have to be added to the VS2008 toolbox. It is only necessary to load the ActiveX controls once to the VS2008. Once they have been added to the toolbox they will be always available even if you create a new project. The following steps describe the procedure:

STEP 1: Open one of the EzTemplate solution files: For example click the solution file

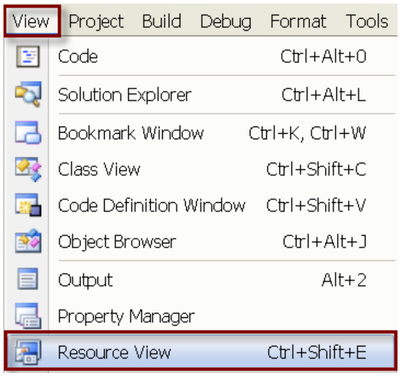


EzTemplate 6 KB Microsoft Visual Studio Solution

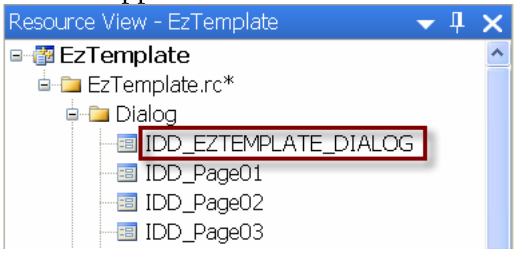
in the following directory to open the Visual Studio 2008:

C:\ICPDAS\EzProg-I\EzTemplate\EzTemplate_1024x768

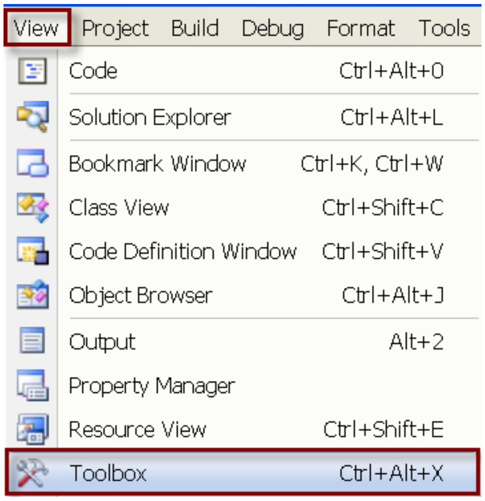
STEP 2: Click *View*→ *Resource View*


View
Project
Build
Debug
Format
Tools
Code Ctrl+Alt+0
Solution Explorer Ctrl+Alt+L
Bookmark Window Ctrl+K, Ctrl+W
Class View Ctrl+Shift+C
Code Definition Window Ctrl+Shift+V
Object Browser Ctrl+Alt+J
Output Alt+2
Property Manager
Resource View Ctrl+Shift+E

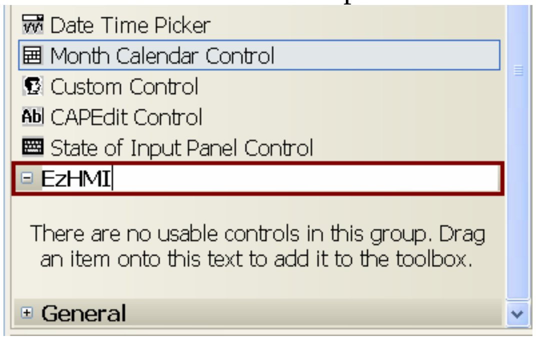
STEP 3: Double click the “IDD_EZTEMPLATE_DIALOG” resource. The dialog sheet will appear.



STEP 4: Click *View*→ *Toolbox* to display the toolbox


View
Project
Build
Debug
Format
Tools
Code Ctrl+Alt+0
Solution Explorer Ctrl+Alt+L
Bookmark Window Ctrl+K, Ctrl+W
Class View Ctrl+Shift+C
Code Definition Window Ctrl+Shift+V
Object Browser Ctrl+Alt+J
Output Alt+2
Property Manager
Resource View Ctrl+Shift+E
Toolbox Ctrl+Alt+X

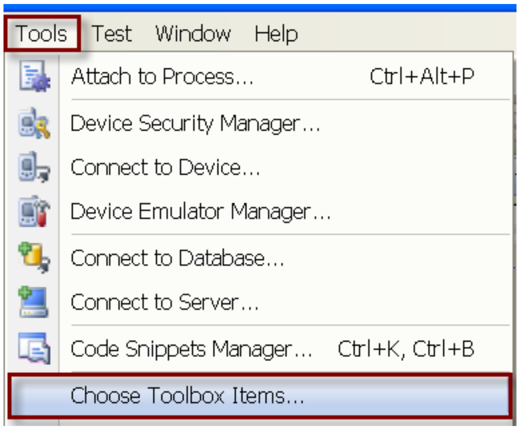
STEP 5: Right click the toolbox and select "Add Tab" from the popup window. Enter "EzHMI" as a tab name and press enter.



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

STEP 6: Click on the EzHMI tab

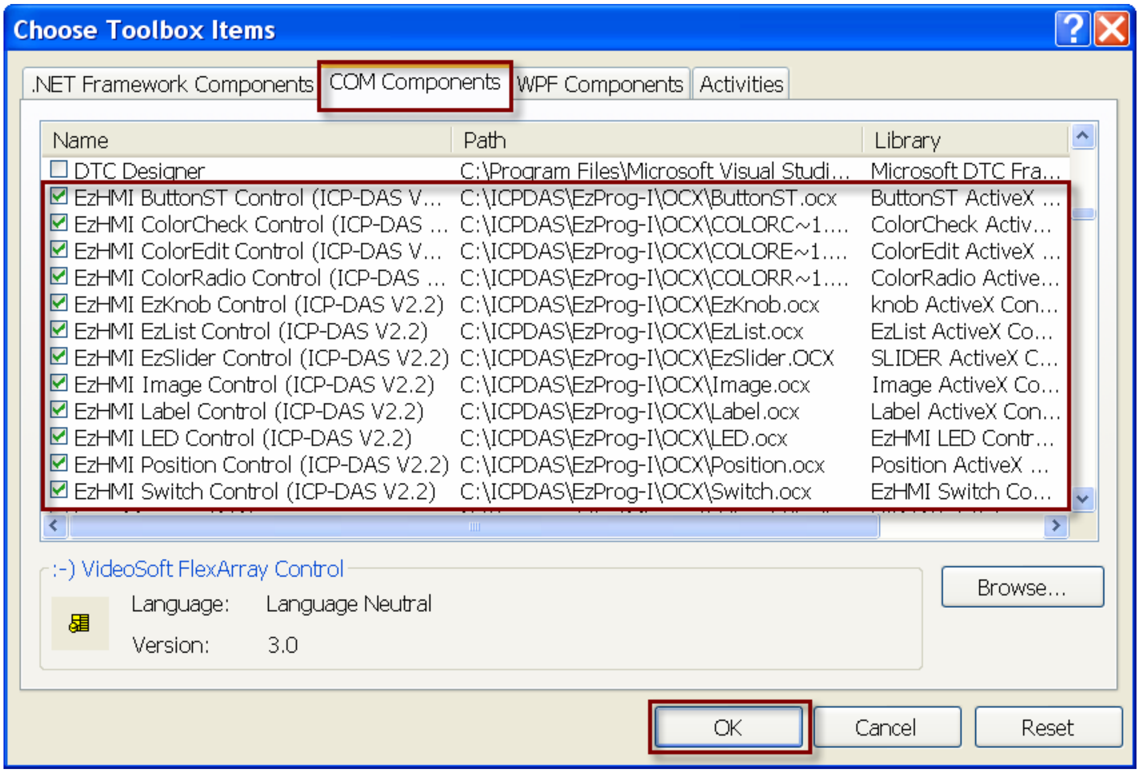
STEP 7: Click *Tools* → *Choose Toolbox Items …*



It will take a short time before the "Choose Toolbox Items" dialog window pops up.

STEP 8: Click the "*COM Components*" tab.

STEP 9: Select from the “*COM Components*” list the 12 EzHMI ActiveX controls and click “*OK*”.



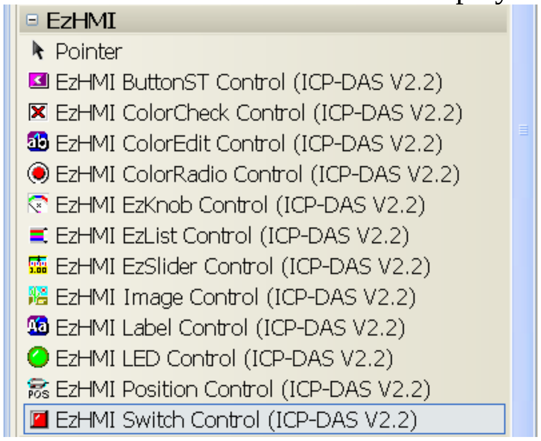
Name	Path	Library
□ DTC Designer	C:\Program Files\Microsoft Visual Studi…	Microsoft DTC Fra…
☑ EzHMI ButtonST Control (ICP-DAS V…	C:\ICPDAS\EzProg-I\OCX\ButtonST.ocx	ButtonST ActiveX …
☑ EzHMI ColorCheck Control (ICP-DAS …	C:\ICPDAS\EzProg-I\OCX\COLORC~1….	ColorCheck Activ…
☑ EzHMI ColorEdit Control (ICP-DAS V…	C:\ICPDAS\EzProg-I\OCX\COLORE~1….	ColorEdit ActiveX …
☑ EzHMI ColorRadio Control (ICP-DAS …	C:\ICPDAS\EzProg-I\OCX\COLORR~1….	ColorRadio Active…
☑ EzHMI EzKnob Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\EzKnob.ocx	knob ActiveX Con…
☑ EzHMI EzList Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\EzList.ocx	EzList ActiveX Co…
☑ EzHMI EzSlider Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\EzSlider.OCX	SLIDER ActiveX C…
☑ EzHMI Image Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\Image.ocx	Image ActiveX Co…
☑ EzHMI Label Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\Label.ocx	Label ActiveX Con…
☑ EzHMI LED Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\LED.ocx	EzHMI LED Contr…
☑ EzHMI Position Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\Position.ocx	Position ActiveX …
☑ EzHMI Switch Control (ICP-DAS V2.2)	C:\ICPDAS\EzProg-I\OCX\Switch.ocx	EzHMI Switch Co…

:-) VideoSoft FlexArray Control

Language: Language Neutral

Version: 3.0

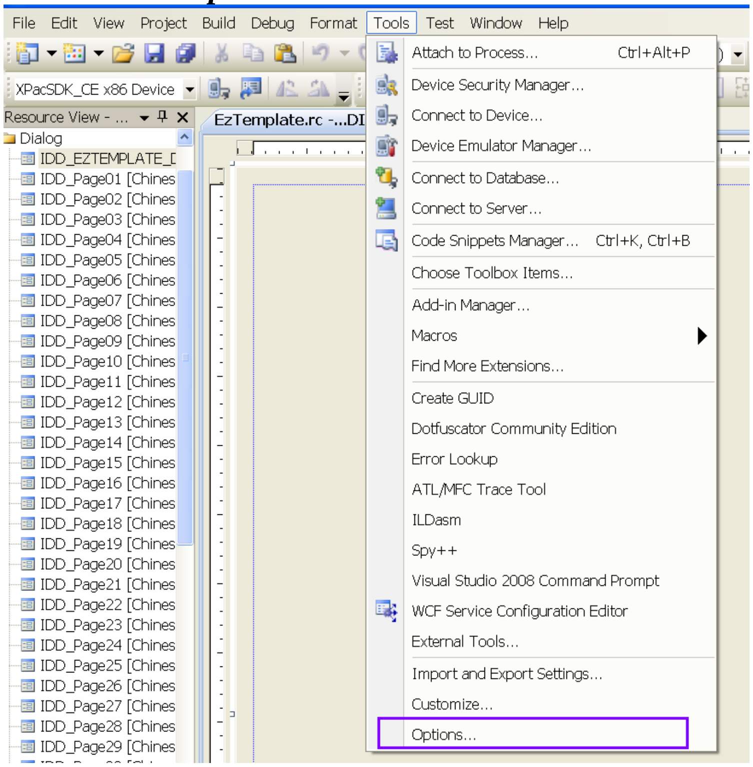
All the selected controls will be displayed beneath the EzHMI tab.


⊟ EzHMI

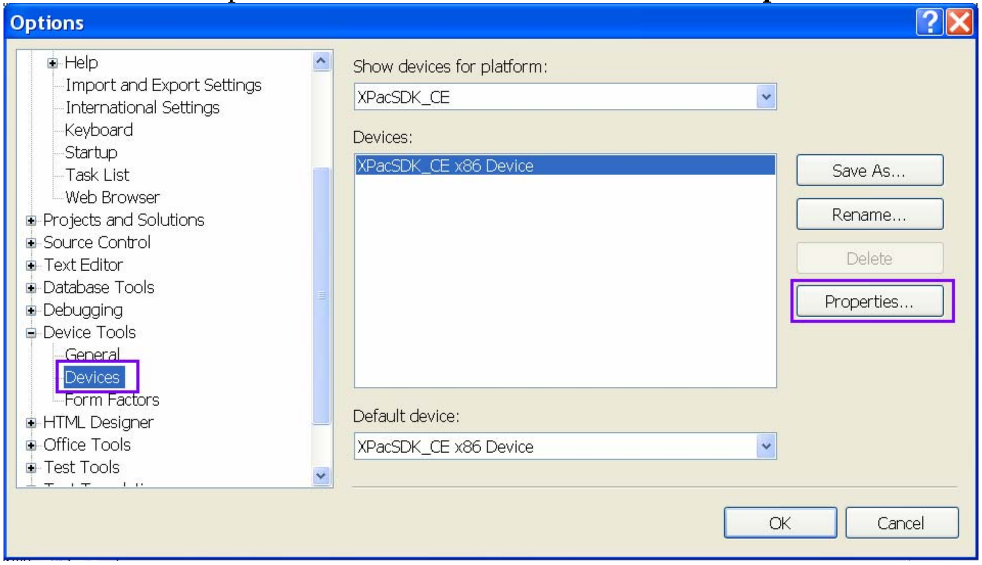
- Pointer
- EzHMI ButtonST Control (ICP-DAS V2.2)
- EzHMI ColorCheck Control (ICP-DAS V2.2)
- EzHMI ColorEdit Control (ICP-DAS V2.2)
- EzHMI ColorRadio Control (ICP-DAS V2.2)
- EzHMI EzKnob Control (ICP-DAS V2.2)
- EzHMI EzList Control (ICP-DAS V2.2)
- EzHMI EzSlider Control (ICP-DAS V2.2)
- EzHMI Image Control (ICP-DAS V2.2)
- EzHMI Label Control (ICP-DAS V2.2)
- EzHMI LED Control (ICP-DAS V2.2)
- EzHMI Position Control (ICP-DAS V2.2)
- EzHMI Switch Control (ICP-DAS V2.2)

4.2 Connecting to MPAC via Ethernet

Step 1: Click ***Tools→Options…***



Step 2: In the tree view open ***Device Tools→Devices*** and click ***Properties…***.

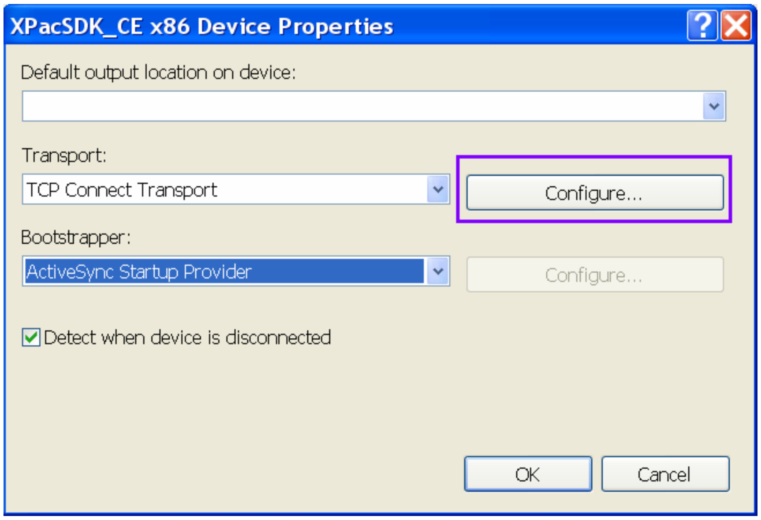


Show devices for platform: XPacSDK_CE

Devices: XPacSDK_CE x86 Device

Default device: XPacSDK_CE x86 Device

Step 3: In the property window click ***Configure…***.



XPacSDK_CE x86 Device Properties

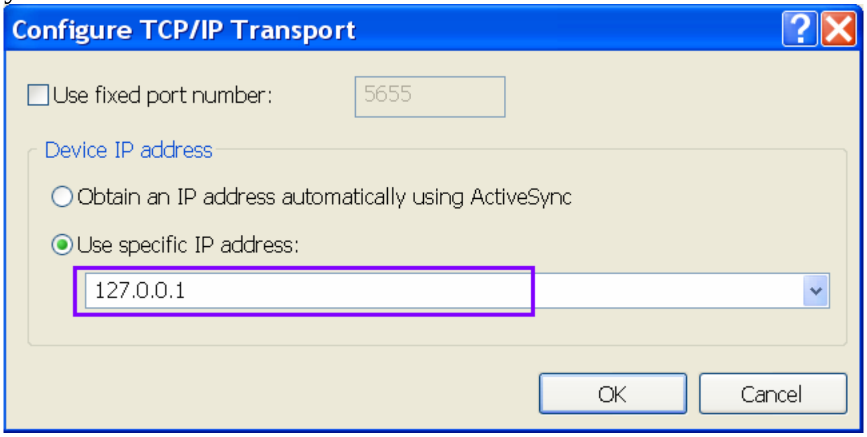
Default output location on device:

Transport: TCP Connect Transport

Bootstrapper: ActiveSync Startup Provider

☑ Detect when device is disconnected

Step 4: Select the "***Use specific IP address:***" option and enter the IP address of your MPAC and click ***OK***.



Configure TCP/IP Transport

☐ Use fixed port number: 5655

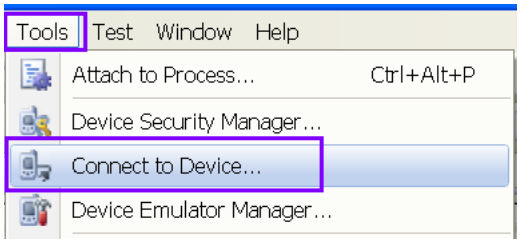
Device IP address

○ Obtain an IP address automatically using ActiveSync

● Use specific IP address: 127.0.0.1

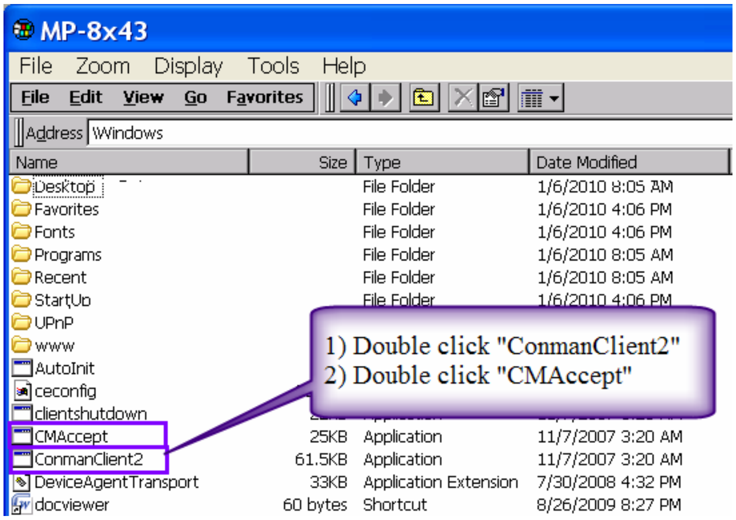
Step 5: Close all open dialog boxes by clicking ***OK***.

Step 6: Open the "Connect to Device" dialog window by clicking ***Tools →Connect to Device…***.



Tools
Test
Window
Help
Attach to Process... Ctrl+Alt+P
Device Security Manager...
Connect to Device...
Device Emulator Manager...

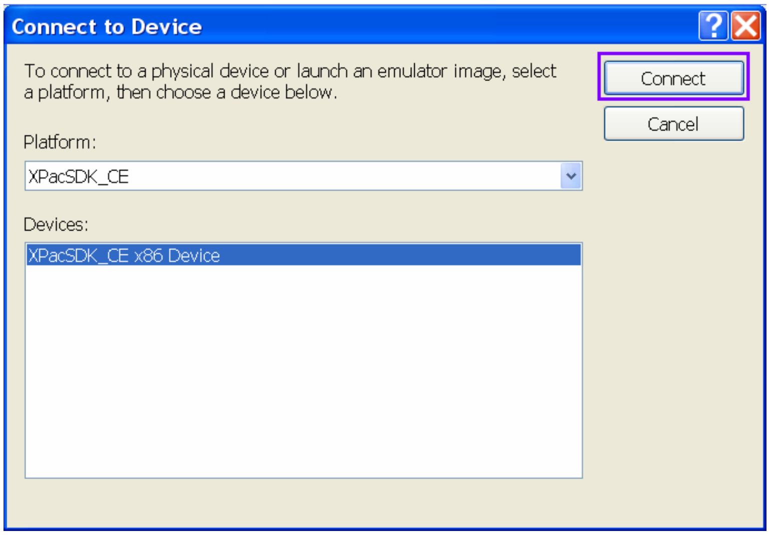
Step 7: On the MPAC device open the \Windows folder and first double click "***ConmanClient2***" and then "***CMAccept***" execution file.



Name	Size	Type	Date Modified
Desktop		File Folder	1/6/2010 8:05 AM
Favorites		File Folder	1/6/2010 4:06 PM
Fonts		File Folder	1/6/2010 4:06 PM
Programs		File Folder	1/6/2010 8:05 AM
Recent		File Folder	1/6/2010 8:05 AM
StartUp		File Folder	1/6/2010 4:06 PM
UPnP			
www			
AutoInit			
ceconfig			
clientshutdown			
CMAccept	25KB	Application	11/7/2007 3:20 AM
ConmanClient2	61.5KB	Application	11/7/2007 3:20 AM
DeviceAgentTransport	33KB	Application Extension	7/30/2008 4:32 PM
docviewer	60 bytes	Shortcut	8/26/2009 8:27 PM

1) Double click "ConmanClient2"
2) Double click "CMAccept"

Step 8: Establish a connection to MPAC by clicking the "***Connect***" button within three minutes after the "***ConmanClient2***" on the MPAC has been started.

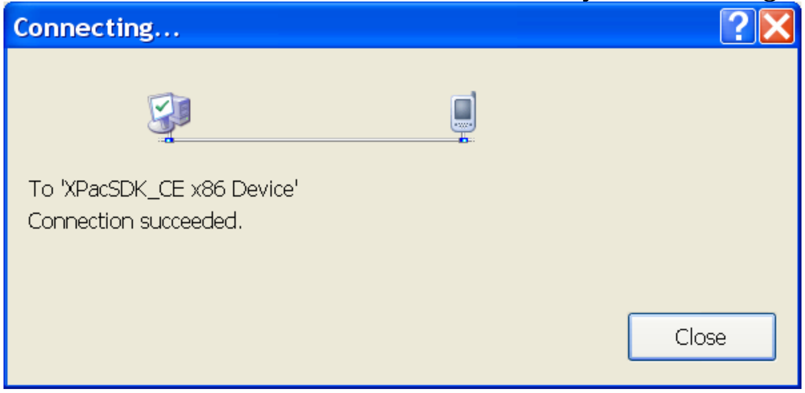


To connect to a physical device or launch an emulator image, select a platform, then choose a device below.

Platform: XPacSDK_CE

Devices: XPacSDK_CE x86 Device

Step 9: A successful connection will be confirmed by the following dialog box:



To 'XPacSDK_CE x86 Device'
Connection succeeded.