

# Chapter 10: C# .net 2008 Program Running In The XP-8xx7-CE6 Access To ISaGRAF Variables

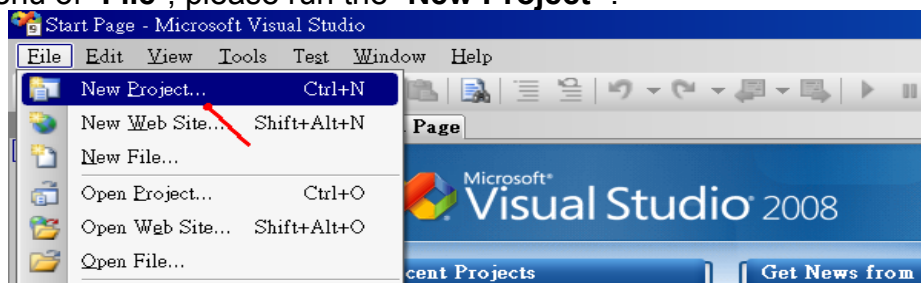
This chapter lists the procedure for creating the first demo program by Visual Studio .NET 2008 development tool. There is some sample programs in the XP-8xx7-CE6 CD-ROM.

XP-8xx7-CE6 CD-ROM : \napdos\isagrafxp-8xx7-ce6\xpce6-CSharp.net-2008-demo\  
wp\_CSharp01 : Digital I/O demo with one I-87055W in slot 1 of the XP-8xx7-CE6.  
wp\_CSharp02 : Analog I/O demo with one I-87024W in slot 2 and one I-8017HW in slot 3.  
wp\_CSharp03 : Read / Write ISaGRAF internal integers, timers and real variables. (No I/O)

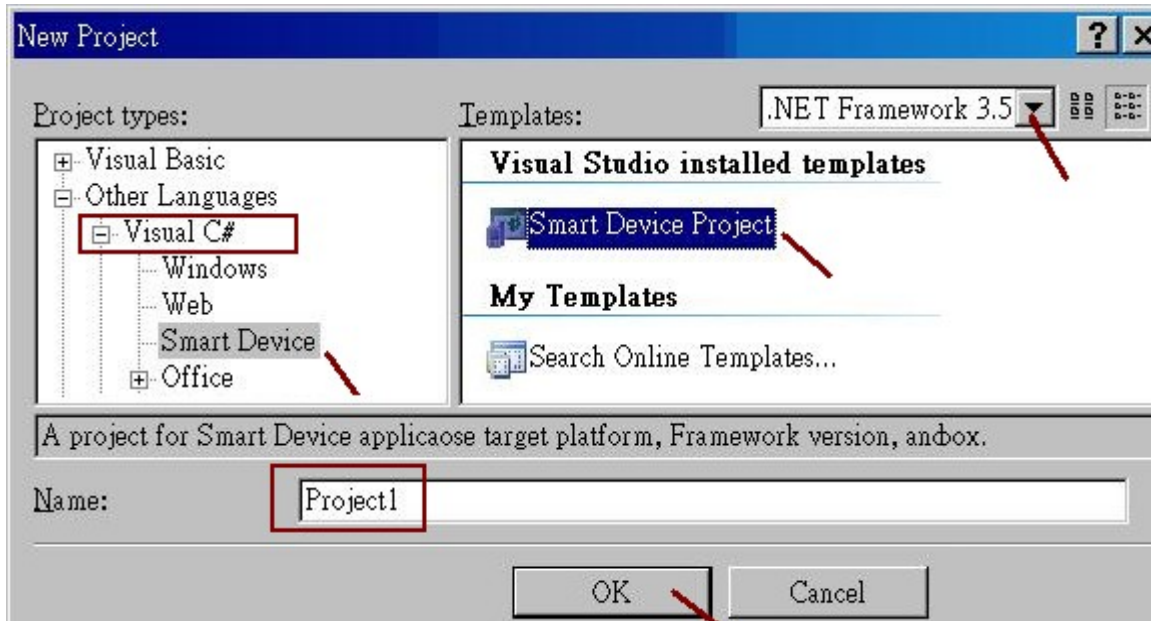
The related ISaGRAF demo project name are "wp\_vb01.pia" , "wp\_vb02.pia" and "wp\_vb03.pia" in the same directory.

## 10.1 Create a New Project

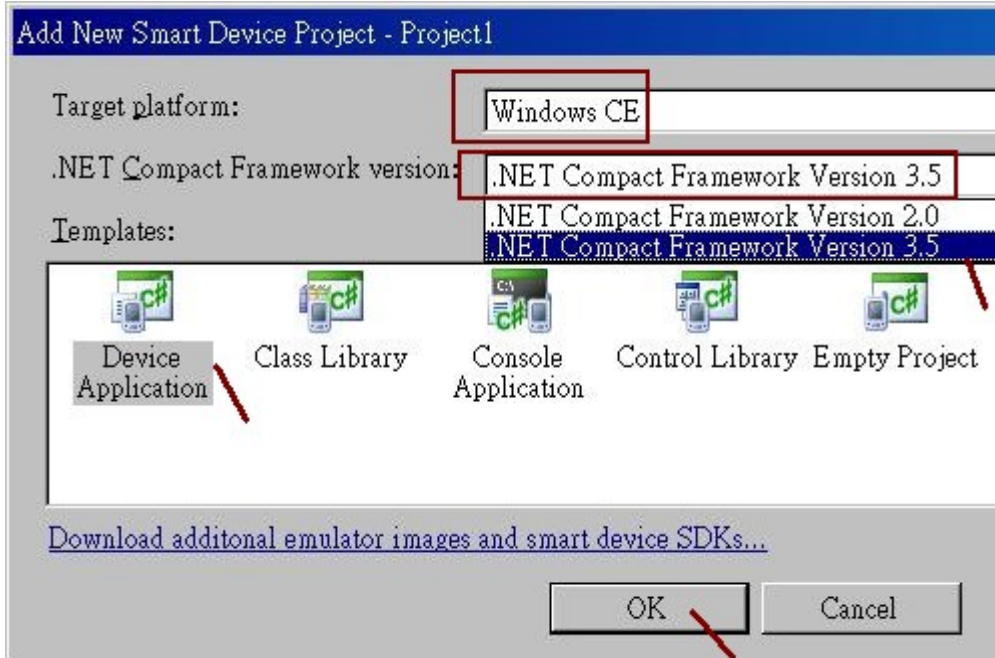
1. In the first, users need to open Microsoft Visual Studio .NET 2008 software. And then in the menu of "File", please run the "New Project".



2. Check the "Smart Device" on the left, then selecting the ".NET framework 3.5" and "Smart Device Project". Then entering a proper project name and the last click on "OK".



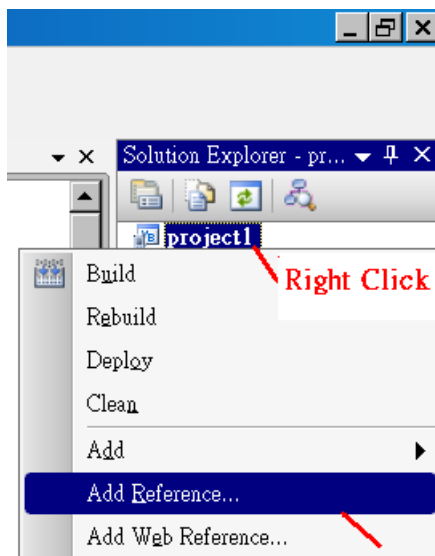
3. Select the "Device Application" and "Windows CE" and ".NET Compact Framework Version 3.5", then click on "OK".



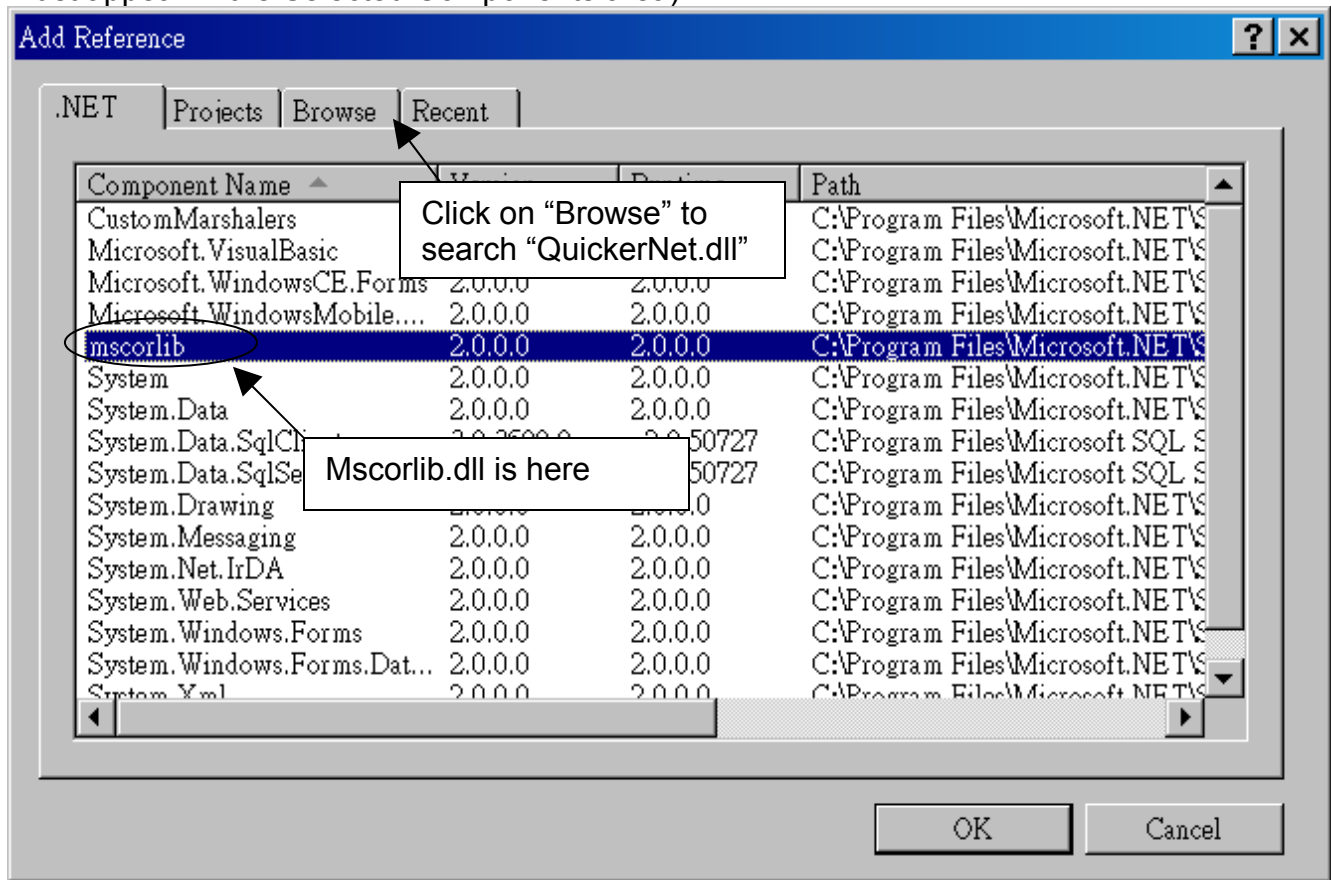
## 10.2 Add Project Reference for an Application

The "QuickerNet" library contains all modules' functions. Before you use the "Quicker" keyword in the program, you must add the "QuickerNet.dll" into the reference list of your application.

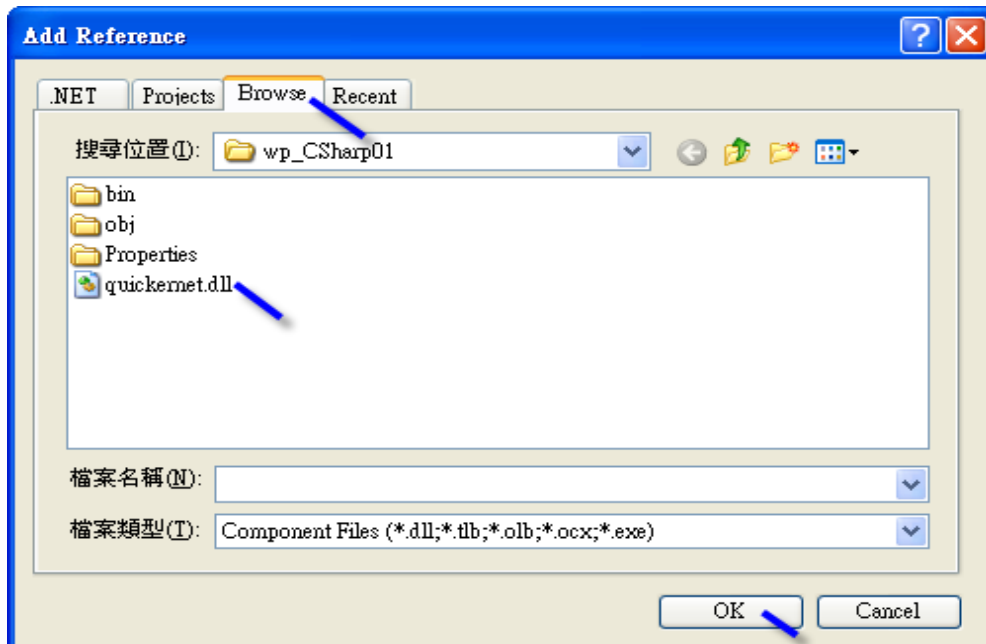
1. Right click on the Project name on the right hand side , then select "Add Reference ..."



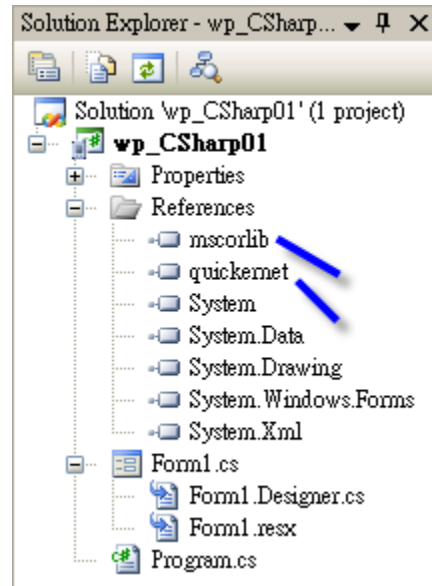
2. Select the “**mscorlib**” in the list box and click the button “**OK**” (the component “**mscorlib**” must appear in the Selected Components area)



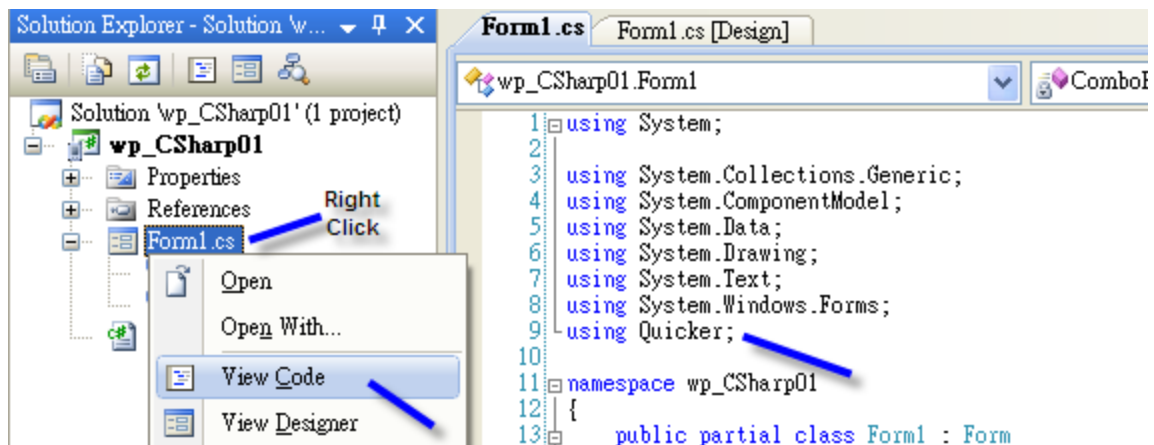
3. Click the “**Browse**” button. Select the “**QuickerNet.dll**” from **XP-8xx7-CE6 CD-ROM : \napdos\isagraf\xp-8xx7-ce6\xpce6-CSharp.net-2008-demo\wp\_CSharp01** subfolder or from your own location.



4. When both “mscorlib” and “QuickerNet.dll” are added, you can see them in the solution explorer as below



5. Right-click on the “Form1.cs” and select “View Code” from the pop-up. Move cursor to top and insert the “using Quicker;” in the first statements.

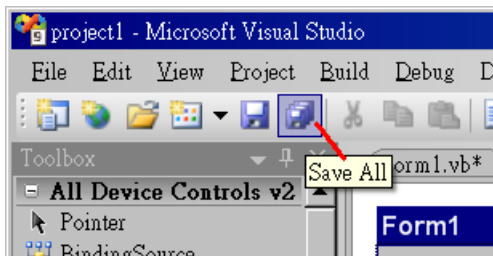


Then you can design all required objects and actions inside your C# Forms .

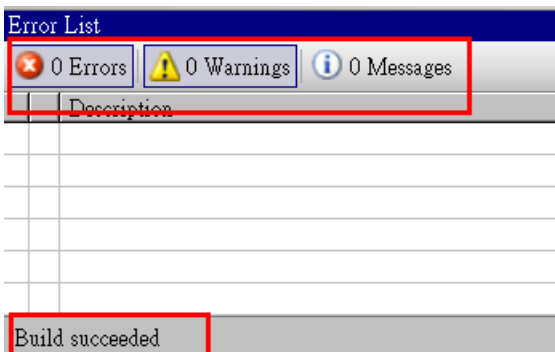
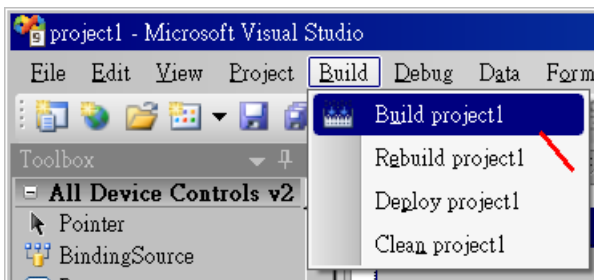
## 10.3 Compiling an Application Program

When you have finished writing a program, you can build an application by the following steps.

1. Remember to save at any time for safety.



2. Then compile (Build) the project . The result is listed in the “Error List” windows at the bottom .



3. You can find the execution file in

<Your C# .net Project folder> \bin\Release\ <project\_name>.exe

Please copy this execution file to the XP-8xx7-CE6 's \System\_Disk\ISaGRAF\ path to run it.

**Note:** User may copy the C#.net execution file to other path to run it but there should contain at least three DLL files with it or it can not run correctly.

For ex, the project1.exe can run in the \System\_Disk\User\ path if there is three plus one file in it. The “project1.exe”, “QuickerNet.dll”, “Quicker.dll” and “Mscorlib.dll” . (The “QuickerNet.dll”, “Quicker.dll” and “Mscorlib.dll” can be copied from the XP-8xx7-CE6 's “\System\_disk\ISaGRAF\” path)

## 10.4 QuickerNET.DLL

---

This section we will focus on the description of the application example of QuickerNET.DLL functions. There are some functions that can be used to R/W data from/to the ISaGRAF softlogic. The functions of QuickerNET.DLL can be clarified as two groups as depicted as below:

1. Digital R/W Functions
2. Analog R/W Functions

### 10.4.1 Digital R/W Functions

#### ■ UserSetCoil

##### **Description:**

This function is to set the value to a Boolean variable by Modbus network address.

##### **Syntax:**

**UserShare.UserSetCoil(ushort iUserAddress, byte iStatus)**

##### **Parameter:**

iUserAddress : Specify the Modbus Network Address of Variable (1 to 8191)

iStatus : Set the status. For instance, iStatus = 1 for True, iStatus = 0 for False

##### **Return Value:**

None

##### **Example:**

```
// Set the output variable of Modbus Network Address "1" to True.
```

```
UserShare.UserSetCoil(Convert.ToUInt16(1), 1);
```

##### **Demo program :**

XP-8xx7-CE6 CD-ROM:

```
\\napdos\isagraf\xp-8xx7-ce6\xpce6-CSharp.net-2008-demo\wp_CSharp01
```

## ■ UserGetCoil

### Description:

This function is to get the value from a boolean variable by Modbus network address.

### Syntax:

**UserShare.UserGetCoil(ushort iUserAddress, out byte iStatus)**

### Parameter:

iUserAddress : Specify the Modbus Network Address of Variable (1 to 8191)

iStatus : Get the variable status , iStatus = 1 for True, iStatus = 0 for False

### Return Value:

None

### Example:

```
// Get the variable status of Network Address "1".
```

```
byte iStatus;
```

```
UserShare.UserGetCoil(Convert.ToUInt16(1),out iStatus);
```

### Demo program :

XP-8xx7-CE6 CD-ROM:

\\napdos\isagraf\xp-8xx7-ce6\xpce6-csharp.net-2008-demo\wp\_csharp01

## 10.4.2 Analog R/W Functions

■ **UserSetReg\_short**   ■ **UserSetReg\_long**   ■ **UserSetReg\_float**

### Description:

These functions are to set 16-bit short integer , 32-bit long integer & 32-bit float value to the specified Modbus network address.

### Syntax:

```
UserShare.UserSetReg_Short(ushort iUserAddress, out int iStatus)
```

```
UserShare.UserSetReg_Long(ushort iUserAddress, out int iStatus)
```

```
UserShare.UserSetReg_Float(ushort iUserAddress, out float iStatus)
```

### Parameter:

iUserAddress : Specify the Network Address of Variable (1 to 8191)

iStatus : Set the short or long integer or float value.

### Example:

```
// Set a long value "1234567" to the variable of Modbus Network Address "1".
```

```
int temp1=1234567;
```

```
UserShare.UserSetReg_long(Convert.ToUInt16(1), out temp );
```

```
// Set a short value "-1234" to the variable of Modbus Network Address "3".
```

```
int temp2= -1234;
```

```
UserShare.UserSetReg_short(Convert.ToUInt16(3), out temp2 );
```

```
// Set a float value "2.174" to the variable of Modbus Network Address "4".
```

```
float temp3=2.174;
```

```
UserShare.UserSetReg_float(Convert.ToUInt16(4), out temp3 );
```

### Demo program :

XP-8xx7-CE6 CD-ROM:

1. \napdos\isagraf\xp-8xx7-ce6\xpce6-csharp.net-2008-demo\wp\_csharp02 for R/W analog I/O
2. \napdos\isagraf\xp-8xx7-ce6\xpce6-csharp.net-2008-demo\wp\_csharp03 for R/W internal long integer, Timer and Real (floating-point) values.

### Note:

**The long integer & timer & real variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of "User's Manual of ISaGRAF Embedded Controllers" or in the CD-ROM:\napdos\isagraf\xp-8xx7-ce6\english-manu\ "User\_Manual\_I\_8xx7.pdf")**



## ■UserGetReg\_short ■UserGetReg\_long ■UserGetReg\_float

### Description:

These functions are to get 16-bit short integer , 32-bit long integer & 32-bit float value from the specified Modbus network address.

### Syntax:

```
UserShare.UserGetReg_Short(ushort iUserAddress, out int iStatus)
```

```
UserShare.UserGetReg_Long(ushort iUserAddress, out int iStatus)
```

```
UserShare.UserGetReg_Float(ushort iUserAddress, out float iStatus)
```

### Parameter:

iUserAddress : Specify the Network Address of Variable (1 to 8191)

iStatus : Get the short or long integer or float value.

### Example:

```
float float_val  
short short_val  
int long_val
```

```
// Get float value of the variable of Modbus Network Address "7".  
UserShare.UserGetReg_float(Convert.ToUInt16(7),out float_val);
```

```
// Get long value of the variable of Modbus Network Address "9".  
UserShare.UserGetReg_long(Convert.ToUInt16(9),out long_val);
```

```
// Get short value of the variable of Modbus Network Address "11".  
UserShare.UserGetReg_short(Convert.ToUInt16(11),out short_val) ;
```

### Demo program :

XP-8xx7-CE6 CD-ROM:

1. \napdos\isagraf\xp-8xx7-ce6\xpce6-csharp.net-2008-demo\wp\_csharp02 for R/W analog I/O
2. \napdos\isagraf\xp-8xx7-ce6\xpce6-csharp.net-2008-demo\wp\_csharp03 for R/W internal long integer, Timer and Real (floating-point) values.

### Note:

**The long integer & timer & float variable's Network Address No. must occupy 2 No. in the ISaGRAF project (refer to section 4.2 of "User's Manual of ISaGRAF Embedded Controllers" or in the CD-ROM:\napdos\isagraf\xp-8xx7-ce6\english-manu\ "User\_Manual\_I\_8xx7.pdf")**