

CASIO
IT-3100 Series
Quick Start Guide
(Version 1.02)

CASIO Computer Co., Ltd.

Copyright ©2010. All rights reserved.

May 2010

Table of the Contents

	Editorial Record	4
	Preface	5
Chapter 1.	Product Overview	6
1.1	Features at a Glance	6
1.2	Library Configuration	7
1.3	Development Reference Manuals	8
1.4	Sample Program	9
Chapter 2.	Prerequisites	10
2.1	Skills Required	10
2.2	Hardware Required	11
2.3	Software Required	14
Chapter 3.	Installing SDK to PC	16
3.1	Application Development	16
3.2	Installing CASIO SDK Files	17
Chapter 4.	Connecting Power Supply to Cradle	21
4.1	HA-B61IO	21
Chapter 5.	Connecting the IT-3100 to PC	23
5.1	Installing IRXpress	24
5.1.1	Installing USB Driver	25
5.1.2	Installing RS-232C Driver	35
5.2	ActiveSync Connection via IrDA	42
5.2.1	Installing ActiveSync	42
5.2.2	Setting ActiveSync	43
5.3	IrDA Connection via Windows Mobile Device Center	44
Chapter 6.	Setting Up the Development Environment	45
6.1	Installing CAB Files	45
6.2	eMbedded Visual C++ 4.0	46
6.3	Visual Studio 2005	47
6.4	Visual Studio .NET 2003	48
6.5	Use of IT-3000 Software Legacy	49
6.6	Use of .NET Application Legacy	50
Chapter 7.	Device Emulator	51
7.1	Software Required	51
7.2	Starting up the Device Emulator	54
7.3	Using the Device Emulator	56
7.3.1	IT-3100 Device Emulator	56
7.3.2	I/O Simulator	57
7.3.3	Connecting via ActiveSync	62
7.4	Debugging Applications	65
7.4.1	Setting Up Build Configuration	65
7.4.2	Debugging Applications	65
Chapter 8.	eMbedded Visual C++	68
8.1	Building a Simple eVC++ 4.0 Test Program	68
8.2	Using CASIO Libraries from eVC++ 4.0	70
Chapter 9.	Visual Studio	72
9.1	Using CASIO .NET Libraries from VB	73
9.2	Using CASIO .NET Libraries from C#	75
9.3	Using CASIO Libraries from C++	77
Chapter 10.	Resources	81

No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan. Information in this document is subject to change without advance notice. CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2010 CASIO Computer Co., Ltd. All rights reserved.

Editorial Record

[illegible]

Preface

This guide clearly and concisely sets out the information developers need to know to get started with the CASIO IT-3100 series development. The best methods of connecting to your development system are covered and step by step instructions for installing and testing the CASIO SDKs are included.

The purpose of this guide is to get you to the point where you can start development; you should refer to the library manuals for detailed information on the specific APIs.

1. Product Overview

1.1 Features at a Glance

The IT-3100 features a built-in high speed thermal printer capable of printing up to 28 lines per second and other notable capabilities including the following functions.

Outstanding development environment

- Microsoft® Windows® CE 5.0 English Version as the built-in OS
- Visual Studio 2008
- Visual Studio 2005
- Visual Studio .NET 2003 (Windows® CE .NET Utilities v 1.1 for Visual Studio .NET 2003)
- eMbedded Visual C++ 4.0

Compatibility with various communication systems

- High speed infrared communication with IrDA Ver. 1.1
- Bluetooth® Version 1.2

Small size, light weight (improved portability)

- Dimensions : Approx. 52.5 (W) x 166 (D) x 30.5 (H) mm
- Weight : Approx. 145 g

Improved durability

- Impact resistance : 1.2 m in height *
- Dust/Water-splash proof : IP54 level (compliant with IEC60529 International Standard)

Superb scanning capability

- Readable 1D symbologies
UPC-A, UPC-E, EAN8, EAN13, Codabar, Code11, Code39, Code93, Code128, ITF, MSI, IATA, GS1 DataBar Omnidirectional, GS1 DataBar Limited, GS1 DataBar Expanded, GS1 DataBar Stacked, GS1 DataBar Expanded Stacked
- Readable 2D symbologies
PDF417, Micro PDF, Code49, Composite, Codablock F, TLC39, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec, DataMatrix, Maxicode, QR Code

CPU/Memory

- High-performance CPU
Marvell® PXA255 Application Processor (runs at maximum 400 MHz)
- Large-capacity memory
RAM : 128 MB
F-ROM : 96 MB (user area; approximately 30 MB)

* The drop durability height is a measured value resulting from actual testing. It does not necessarily guarantee the product from damage.

1.2 Library Configuration

The CASIO Software Development Kit (“SDK”) for IT-3100 series provides various libraries listed in the table. The compatibility on these libraries with ones for IT-3000, see Chapter 6.5 “Use of IT-3000 Software Legacy”.

Table 1.1

Library	Description	C++	VB C#
System Library	Library that is used to control the system.	Yes	Yes
Bluetooth Library	Library that is used to control the built-in Bluetooth module.	Yes	Yes
Printer Library	Library that is used to control the built-in Printer module.	Yes	Yes
Imager Library	Library that is used to control the built-in CMOS Imager.	Yes	Yes
MCR Library	Library that is used to control the built-in Magnetic Card Reader.	Yes	Yes
SOBR Library	Library that is used to control an external bar code reader connected to the 8-pin serial port.	Yes	Yes
GUI Library (note 2)	Library that is used to control the built-in Printer module in the print system.	Yes	Yes
Communication Port Library	Library that is used to control communication via the extension port.	Yes	-
JPEG Library	Library that is used to handle and manipulate JPEG image functions.	Yes	-
FLINK Library	Library that is used to control and carry out transmission/reception of files between PC and other device.	Yes	Yes

Notes:

- The abbreviations used in the table are;
C++ : Visual C++
VB : Visual Basic .NET
C# : Visual C# .NET
- The library changed its name from “Print Library” to “GUI Library”.

The names of the Dynamic Link Libraries for C++ and C#/VB for the device oriented libraries are listed in the table.

Table 1.2

Library	Dynamic Link Library	Dynamic Link Library (Class Library)
System Library	SystemLib.dll	SystemLibNet.dll
Bluetooth Library	BluetoothLib.dll	BluetoothLibNet.dll
Printer Library	PrinterLib.dll	PrinterLibNet.dll
Imager Library	ImagerLib.dll	ImagerLibNet.dll
MCR Library	MCReadLib.dll	MCReadLibNet.dll
SOBR Library	SerialObrLib.dll	SerialObrLibNet.dll
GUI Library	Cp780Lib.dll	Cp780LibVB.dll Cp780LibCS.dll
Communication Port Library	None	None
JPEG Library	JpegCe.dll	None
FLINK Library	FlinkLib.dll	MoFlinkLib.dll

1.3 Development Reference Manuals

The CASIO Software Development Kit (“SDK”) for IT-3100 series includes the development reference manuals listed in the table.

Table 1.3

Development Manual	Description
Quick Start Guide	This reference manual.
Hardware Manual	Reference manual that describes hardware specifications in detail on each dedicated option and IT-300 handheld printer terminal
Software Manual	Reference manual that describes software specifications in detail for all the software integrated in IT-3100 handheld printer terminal.
System Library Manual	Reference manual that describes individual functions in detail for System Library.
Bluetooth Library Manual	Reference manual that describes individual functions in detail for Bluetooth Library.
Printer Library Manual	Reference manual that describes individual functions in detail for Printer Library.
Imager Library Manual	Reference manual that describes individual functions in detail for Imager Library.
MCR Library Manual	Reference manual that describes individual functions in detail for MCR Library.
SOBR Library Manual	Reference manual that describes individual functions in detail for SOBR Library.
GUI Library Manual (note)	Reference manual that describes individual functions in detail for GUI Library.
Communication Port Library Manual	Reference manual that describes individual functions in detail for Communication Port Library.
JPEG Library Manual	Reference manual that describes individual functions in detail for JPEG Library.
FLINK Library Manual	Reference manual that describes individual functions in detail for FLINK Library.

Note:

The library changed its name from “Print Library” to “GUI Library”.

1.4 Sample Program

The following sample programs are included in the **IT-3100 SDK**.

Table 1.4

Sample Program	Description
Common Device Control Library	
BLUETOOTHLIBSAMPLE	Connects the terminal to a Bluetooth printer and prints out data.
CAMERALIBSAMPLE	Takes pictures and displays them on the screen. (note 2)
IMGLIBSAMPLE	Scans bar codes using the Imager with settings set with Imager setting file.
IMGLIBSAMPLE2	Program of IMGDemo.exe
OBRLIBSAMPLE	Reads a bar code using the Laser library. (note 2)
PRNLIBSAMPLE	Prints out data on the built-in printer.
SYSTEMLIBSAMPLE	Demonstrates LED and buzzer functions.
JPEG Library	
JPEGSAMPLE	Displays Jpeg files.
FLINK Library	
FLINKLIBSAMPLE	Demonstrate infrared communication via FLINK protocol between two terminals.

Notes:

- Two-character abbreviation showing the respective development platforms is added at the end of each folder name.

VC : Visual C++

VB : Visual Basic .NET

CS : Visual C# .NET
- Some of the sample programs such as **CAMERALIBSAMPLE** and **OBRLIBSAMPLE** in the table do not run because the respective devices are not integrated in the IT-3100.

2. Prerequisites

2.1 Skills Required

The following skills are required by developers aiming to develop application software for the IT-3100.

- Windows programming
- A good knowledge of one or more of the following
 - Visual C++
 - Visual Basic .NET
 - Visual C# .NET
 - Browser based applications (not covered in this guide)

The following skills or experience are also desirable.

- Windows CE devices
- ActiveSync
- Some networking experience

2.2 Hardware Required

The following models of the IT-3100 series (see Table 2.1) and dedicated options (see Table 2.2) are available.

- IT-3100

Table 2.1 Available models and features

Model no.	RAM	MCR	C-MOS	8-pin Serial
IT-3100M53E IT-3100M53E-CN	128 MB	No	No	Yes
IT-3100M54E IT-3100M54E-CN	128 MB	Yes	No	Yes
IT-3100M55E IT-3100M55E-CN	128 MB	Yes	Yes	No
IT-3100M55U	128 MB	Yes	Yes	No
IT-3100M56E IT-3100M56E-CN	128 MB	No	Yes	No
IT-3100M56U	128 MB	No	Yes	No

Note:

“-CN” attached at the end of model number denotes that the model is dedicated for the final destination of China. A note about compliance with the Chinese “RoHS” requirement promulgated by the Ministerial Decree No. 39 is included in the carton box; the RoHS compliant seal is affixed on the body and the seal of the packing material recycle marking is affixed on the carton box.

Table 2.2 Available options

Model no.	Description
HA-B61IO HA-B61IO-CN	Bridge Satellite Cradle
HA-B30CHG HA-B30CHG-CN	Cradle-type Battery Charger
HA-B34AT HA-B34AT-CN	Battery Charger Car Mount Unit
DT-9723LIC DT-9723LIC-CN	Battery Pack
DT-9721CHGE DT-9721CHGE-CN	Battery Charger
AD-S42120BE AD-S42120BE-CN	AC Adaptor with power cable accompanied
AD-S42120B	AC Adaptor without power cable accompanied
AC-CORD-EU	Power Cord for AD-S42120B/Europe
AC-CORD-US	Power Cord for AD-S42120B/USA and Canada
AC-CORD-TW	Power Cord for AD-S42120B/Taiwan
AC-CORD-AU	Power Cord for AD-S42120B/Australia
AC-CORD-KR	Power Cord for AD-S42120B/Korea
AD-S10095-AE	AC Adaptor
AD-S10095-AU	AC Adaptor
DT-827CAC DT-827CAC-CN	Car Power Cable
HA-B80AX HA-B80AX-CN	RS-232C Cable
HA-B90DCV	Screen Protect Cover
HA-B92PCV HA-B92PCV-CN	Splash Protect Cover
HA-B93PH HA-B93PH-CN	Paper Holder
DT-887AXA DT-887AXA-CN	RS-232C Cable
DT-888RSC	RS-422 Cable
DT-380USB	USB Cable

Note:

“-CN” attached at the end of model number denotes that the model is dedicated for the final destination of China. A note about compliance with the Chinese “RoHS” requirement promulgated by the Ministerial Decree No. 39 is included in the carton box; the RoHS compliant seal is affixed on the body and the seal of the packing material recycle marking is affixed on the carton box.

See the following page for external views of the IT-3100 and the dedicated options.

External views of the IT-3100 and the dedicated options

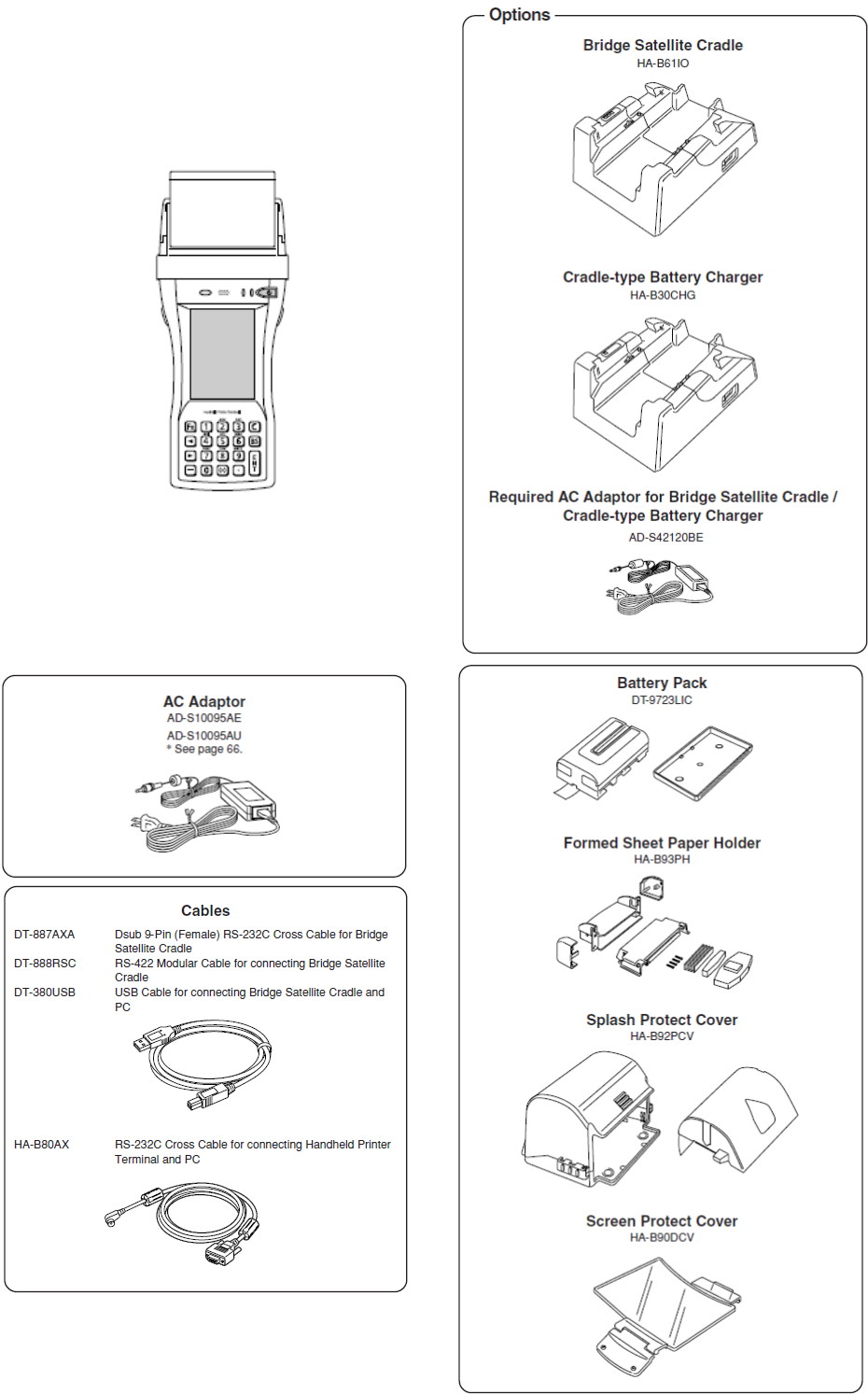


Figure 2.1

2.3 Software Required

PC Operating System

- Microsoft Windows 2000 Professional Service Pack 4 or later
- Or Microsoft Windows 2000 Server Service Pack 4 or later
- Or Microsoft Windows XP Professional Service Pack 2 or later
- Or Microsoft Windows 2003 Server Service Pack 1 or later
- Or Microsoft Windows Vista (Business / Ultimate)
- Or Microsoft Windows 7
- Or Microsoft Windows Server 2008

The following software tools and libraries are required in order to develop software for the IT-3100. Please ensure that you download or purchase the correct Microsoft tools as appropriate.

C/C++

- Microsoft Visual Studio 2008 (not free of charge)
- Microsoft Visual Studio 2005 (not free of charge)
- Microsoft eMbedded Visual C++ 4.0
Download for free from;
<http://www.Microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=1DACDB3D-50D1-41B2-A107-FA75AE960856>
- Microsoft eMbedded Visual C++ 4.0 SP4
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?familyid=4A4ED1F4-91D3-4DBE-986E-A812984318E5&displaylang=en>
- Microsoft ActiveSync 4.2 (or later)
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- Microsoft Windows Mobile Device Center 6.1 (for Windows Vista)
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?familyid=46F72DF1-E46A-4A5F-A791-09F07AAA1914&displaylang=en>

Visual Basic/Visual C#

- Microsoft Visual Studio 2008 (not free of charge)
- Microsoft Visual Studio 2005 (not free of charge)
- Microsoft Visual Studio .NET 2003 (not free of charge)
- Windows CE Utilities for Visual Studio .NET 2003 Add-on Pack 1.1
From Microsoft web site, see
<http://www.microsoft.com/downloads/details.aspx?familyid=7ec99ca6-2095-4086-b0cc-7c6c39b28762&displaylang=en>
- Microsoft ActiveSync 4.2 (or later)
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- Microsoft Windows Mobile Device Center 6.1 (for Windows Vista)
Download for free from;

<http://www.microsoft.com/downloads/details.aspx?familyid=46F72DF1-E46A-4A5F-A791-09F07AAA1914&displaylang=en>

CASIO IT-3100 SDK

Download the IT-3100 SDK from

http://www2.casio.co.jp/system_en/pa/PADealer/

(The site requires your user name and password. Enter your user name and password as issued by CASIO.)

3. Installing SDK to PC

3.1 Application Development

This chapter explains about what you need to set up for the development environment before starting your application development.

1. Installing Development Platform

Install Microsoft's development platform which supports Visual Studio 2005. For detail, refer to Chapter 2.3 "Software Required".

2. Installing CASIO SDK to PC

Install Casio's SDK and various libraries if necessary. For installation method, refer to Chapter 3.2 "Installing CASIO SDK Files".

3. Connecting IT-3100 to PC (via ActiveSync/Windows Mobile Device Center)

Connect the IT-3100 to PC via Microsoft's ActiveSync (for Windows XP or any other OS before Windows XP) or via Windows Mobile Device Center (for Windows Vista or later). For connection method, refer to Chapter 5 "Connecting the IT-3100 to PC".

4. Setting Up the Development Environment

Copy all necessary CASIO libraries to the IT-3100.

For detail, refer to Chapter 6 "Setting Up the Development Environment".

5. Installing the Device Emulator

Install the Device Emulator for the IT-3100. For installation method, refer to Chapter 7 "Device Emulator". If not necessary to install, go to 6 "Application Development" below.

6. Application Development

Now, the application development environment is set up and your development with the development platform can be started. After application is developed, transfer it to the Device Emulator or an actual terminal of the IT-3100 via ActiveSync or Windows Mobile Device Center for check on the operability. For application development method and transferring your application, refer to Chapter 8 "eMbedded Visual C++" and Chapter 9 "Visual Studio".

3.2 Installing CASIO SDK Files

Download the CASIO IT-3100 SDK installation CD image file from the following site and write it to a CD-ROM meadia.

http://www2.casio.co.jp/system_en/pa/PADealer/

(The site requires your user name and password. Enter your user name and password as issued by CASIO.)

Notes:

- If you had already installed IT-3100 SDK Ver. 1.xx, be sure to uninstall it before installing Ver. 2.xx.
- If any file of the IT-3100 SDK Ver. 1.xx is remained in the following folders after uninstalling, delete it manually.

C:\Program Files\CASIO\MBSYS

C:\Program Files\Windows CE Tools\wce500

- If your PC runs in the Windows Vista or later OS, first you must disable the User Account Control (“UAC”) by following the process below before installing the CASIO SDK Files.

Case of using Windows Vista and Windows Server 2008

- Navigate to **Control Panel** → **User Accounts** → **Enable or Disable the User Account Control**. Remove the check on **User Account Control (UAC) to protect your PC**, and then click **OK** button.

Case of using Windows 7

- Navigate to **Control Panel** → **User Accounts** → **Change User Account Control Settings**. Select **Never notify** in Choose when to be notified about changed to your computer setting.

Installing ExportSDK

1. When the CASIO IT-3100 SDK CD-ROM is inserted in the drive of your PC, the following menu screen appears automatically.



Figure 3.1

2. Click **Installation List** in Figure 3.1. The installation screen appears.



Figure 3.2

3. Click **IT-3100 ExportSDK** to display the Setup Wizard. Choose any one of the buttons in the screen.

Installing Library

The IT-3100 Library is available for the below development platforms. The library can be installed for each development platform in the **Installation List** screen.

- Microsoft eMbedded Visual C++ 4.0
- Microsoft Visual Studio .NET 2003
- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

Notes:

1. The Library header file (*.h) and the Import library file (*.lib) are installed in the following folder.

C:\Program Files\CASIO\MBSYS\include : Header file

C:\Program Files\CASIO\MBSYS\lib\ARMV4I : Import library file

2. The Class library DLL file (*.dll) is installed in the following folder.

C:\Program Files\CASIO\MBSYS\WindowsCE

3. When the Library is installed in PC, the Help file (*.chm) is installed in the following folder.

C:\Program Files\CASIO\MBSYS\HELP

The Help file can be accessed by navigating to **Start** menu → **All Programs** → **CASIO Basic Development Kit** → **Help**.

Installing Online Help

Besides the Help file described above, the Online Help is also available for the below development platforms. The installation of the Online Help is initiated in the **Installation List** for each development platform.

- Microsoft Visual Studio .NET 2003
- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

After the installation is complete, Microsoft Visual Studio starts up. The Online Help file can be accessed by navigating to **Help** → **Contents** menu.

Installing Sample Program

When the **Sample** is chosen in the **Installation List**, the sample program folder in the CD-ROM appears. The Sample program folder is available for each development platform. Copy one of the folders you wish to use into your PC.

- EVC for Microsoft embedded Visual C++
- VS2003 for Microsoft Visual Studio.NET 2003
- VS2005 for Microsoft Visual Studio 2005
- VS2008 for Microsoft Visual Studio 2008

If your sample program is with "Read-only" attribute set effect. Be sure to disable the attribute before using it.

4. Connecting Power Supply to Cradle

4.1 HA-B61IO

Use the dedicated AC adaptor (AD-S42120BE or AD-S42120B) for supplying power to the HA-B61IO Bridge Satellite Cradle. Ensure that you connect the AC adaptor to the cradle before starting communication between the IT-3100 and PC via the cradle.

Use the dedicated cable (DT-887AXA or DT-380USB) to connect the HA-B61IO with PC.

Follow the steps below to set up the DIP switch on the cradle and to connect the power supply to the IT-3100 via the cradle using the dedicated AC adaptor.

1. First, set up no. 6 of the DIP switch (see Figure 4.1) on the bottom of the HA-B61IO according to the cable you use, either DT-887AXA or DT-380USB.

Other switches are not needed to change. Make sure that all the switches except no. 6 are at the initial positions (see Figure 4.1).

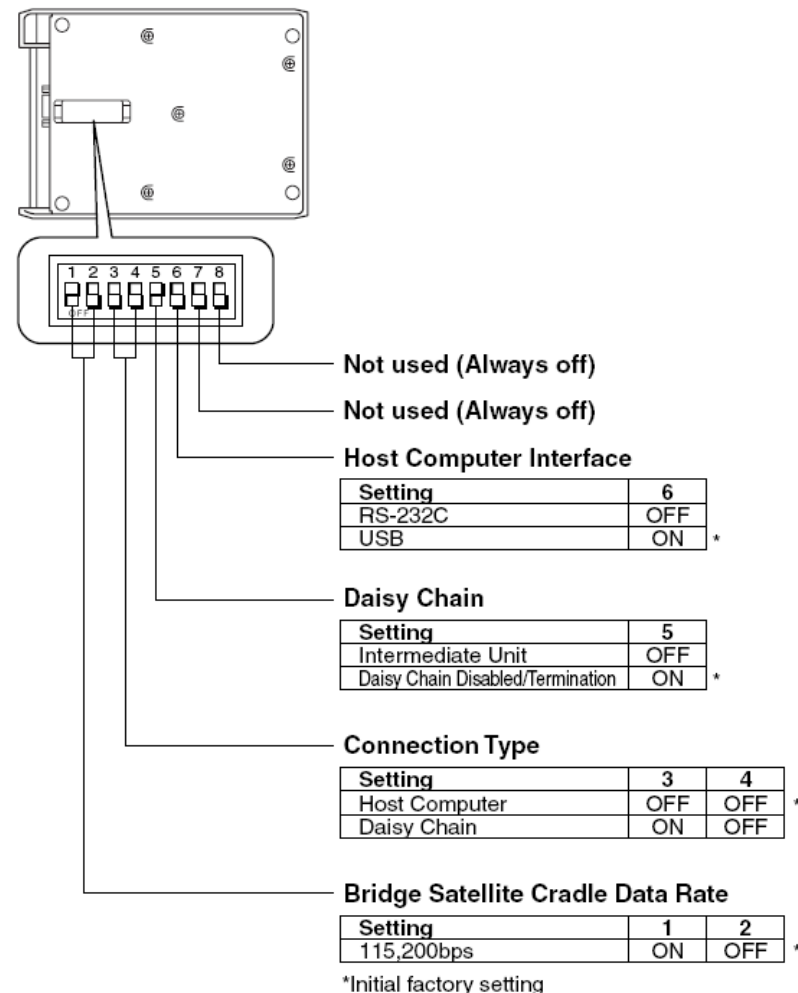


Figure 4.1

2. Plug the AC adaptor into the AC adaptor jack where “DCIN12V” is printed on the back of the cradle.
3. After connecting the power cable to the AC adaptor, plug in the plug to an electrical outlet.
4. Turn on the power switch on the cradle.
5. Connect the DT-887AXA cable to RS-232C port on the back of the cradle or DT-380USB cable to USB port and the other end of the cable to either RS-232C port or USB port on the PC.
6. Align the USB cradle mount holes on the back of the IT-3100 with the mount hooks on the cradle after aligning the contacts on the bottom of the IT-3100 with the power contacts of the cradle.

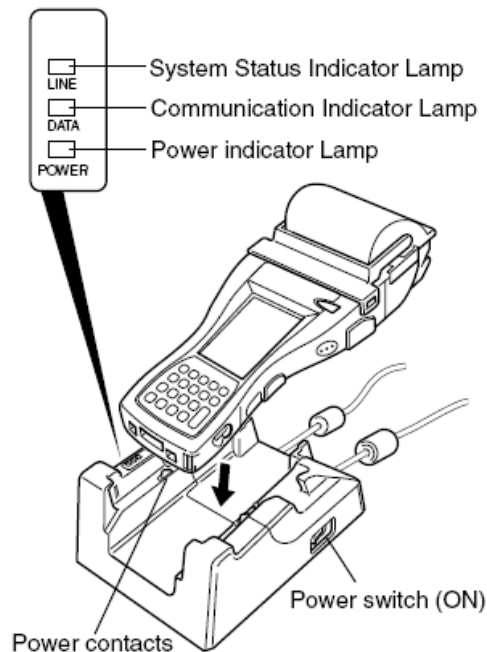


Figure 4.2

The power LED on the front of the cradle will light green if the IT-3100 has been properly mounted.

Status of the Power indicator lamp on IT-3100

- | | |
|--------|---|
| Orange | : Charging |
| Red | : Standby due to battery pack error or the surrounding temperature is out of the range (charging begins when the temperature returns within the correct range.) |
| Green | : Charging complete |

Notes:

- Always remove the IT-3100 from the cradle when switching the selector switch on the cradle.
- Never short the power contacts of the cradle. This damages the cradle.
- Do not subject the IT-3100 and cradle to vibration or impact during communication. This results in communication being interrupted.
- When mounting the IT-3100, securely attach it to the mount hooks of the cradle and check that the power LED on the front of the cradle lights green. Charging the battery pack or communication will not proceed if it is not mounted properly.

5. Connecting the IT-3100 to PC

To make connection establishment with PC, use one of the methods, depending on the OS your PC runs, described below.

- **ActiveSync** (for Windows XP or any other OS earlier)
Use the ActiveSync to connect the IT-3100 to PC if the PC runs in Windows XP or other OS earlier than Windows XP. The ActiveSync can be downloaded at the URL below.
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- **Windows Mobile Device Center** (for Windows Vista or later)
Use the Windows Mobile Device Center to connect the IT-3100 to PC if the PC runs in Windows Vista OS. The Windows Mobile Device Center (“WMDC”) can be downloaded at the URL below.
<http://www.microsoft.com/downloads/details.aspx?familyid=46F72DF1-E46A-4A5F-A791-09F07AAA1914&displaylang=en>

Notes:

- Always remove the IT-3100 from the cradle when switching the selector switch on the cradle.
- Never short-circuit the pins of the cradle. Shorting the pins will cause serious damage to the cradle.
- Do not vibrate or knock the cradle or the IT-3100 while communicating. Any vibration might cause disconnection of communication.

5.1 Installing IRXpress

This chapter describes about the installation procedure for **IRXpress** (“IrDA driver”) in the PC. In order to establish connection of IT-3100 with PC, the **IRXpress** driver must be pre-installed in the PC.

Install either USB driver or RS-232C driver in the PC depending on the cable you use to establish connection via the cradle. Method of installing USB and RS-232C drivers is dependant on the OS type of your PC.

Choose one of the drivers you wish to install in your PC and download the appropriate file depending on the OS type from the CASIO web site.

Table 5.1

Driver	Windows Server 2003 Windows XP Professional Windows 2000 Server Windows 2000 Professional	Windows Vista Business Windows Vista Ultimate
USB driver	IRXpressUSBIrDA.exe	SetupUSBIrDA.exe (see note 1) IRXpressUSBIrDA.exe
RS-232C driver	IRXpressSerialIrDA.exe	SetupSerialIrDA.exe (see note 2) IRXpressSerialIrDA.exe

Notes:

1. Download both files, **SetupUSBIrDA.exe** and **IRXpressUSBIrDA.exe**, in the same folder and then initiate **SetupUSBIrDA.exe**.
2. Download both files, **SetupSerialIrDA.exe** and **IRXpressSerialIrDA.exe**, in the same folder and then initiate **SetupSerialIrDA.exe**.
3. Be sure to close all applications if running before installing the driver. The installation of the driver prompts you to start up your PC.

5.1.1 Installing USB Driver

This chapter describes the USB driver installation method by the OS type your PC runs. Setting up communication speed is also required after installing the USB driver. See “Setting Up the USB Communication Speed” on page 32.

Windows Server 2003, Windows XP, Windows 2000

1. Initiate **IRXpressUSBIrDA.exe**.

When the preparation is complete, the background screen will change to InstallShield Wizard screen. The rest of screens in this installation method will show only pop-up screens and the background screen is omitted.



Figure 5.1

2. Click **Next >** button in Figure 5.1. The default installation path of the destination directory appears (see Figure 5.2).



Figure 5.2

3. The confirmation screen appears after clicking **Next >** button in Figure 5.2. If all detail appeared in the below screen is okay, click **Next >** button.

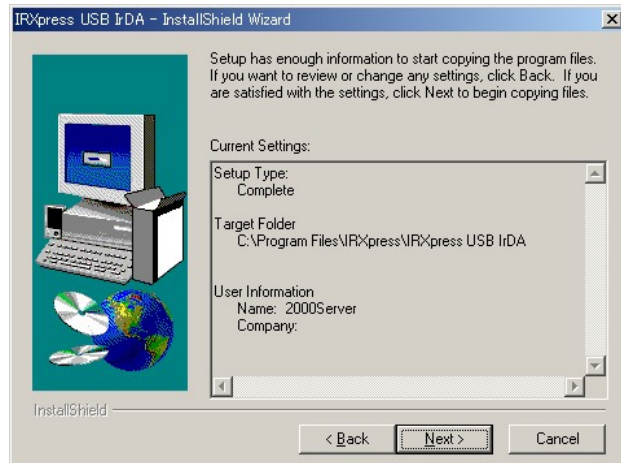


Figure 5.3

4. The installation screen appears. Choose the radio button of **Yes, I want to restart my computer now**, and **Finish** button.



Figure 5.4

5. Clicking **Finish** button starts up the PC again.
6. After starting up the PC, turn on the power on the cradle. The device recognition is carried out automatically.

Note:

In Windows 2000, device recognition on the cradle will be carried out automatically when the PC is started up. In WindowsXP and Windows Server 2003, the device recognition process must be carried out manually. For the procedure, refer to the following pages.

Continuing the installation process for Windows XP

1. After the PC has started up again, turn on the power on the cradle and then the below screen appears.



Figure 5.5

2. Choose **Install the software automatically [Recommended]** radio button in Figure 5.5. The below screen appears.



Figure 5.6

3. Clicking **Finish** button in Figure 5.6 starts up the PC again.

Continuing the installation process for Windows Server 2003

1. After the PC has started up again, turn on the power on the cradle.
2. Navigate to **Control Panel** → **System**.
3. Choose the **Device Manager** in **Hardware** tab.
4. Right-click **IRExpress USB IrDA Controller** in the Other Device and choose **Update Driver** in the menu. This initiates the screen below to appear.



Figure 5.7

5. Choose **Install the software automatically [Recommended]** radio button in Figure 5.7 and click **Next >** button. The screen below appears.



Figure 5.8

6. Click **Finish** button.

Continuing the installation process for Windows Vista

1. Initiate **SetupUSBIrDA.exe**. The User Account Control screen appears.

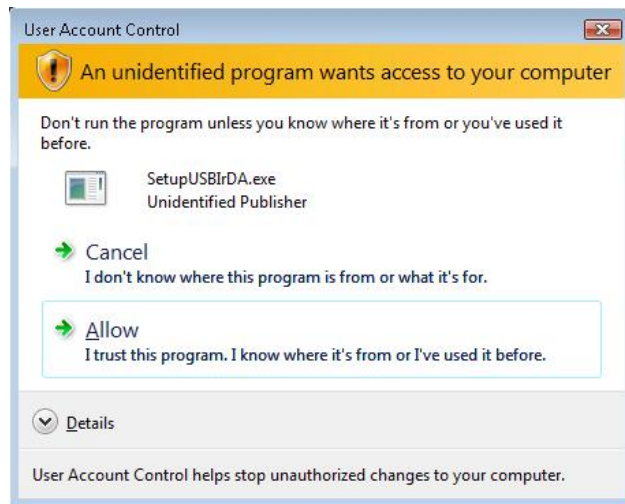


Figure 5.9

2. Choose **Allow** in Figure 5.9 to continue the installation. When the preparation is complete, the background screen changes to InstallShield Wizard screen (see Figure 5.10). The rest of screens in the installation process will show only pop-up screens and the background screen is omitted.

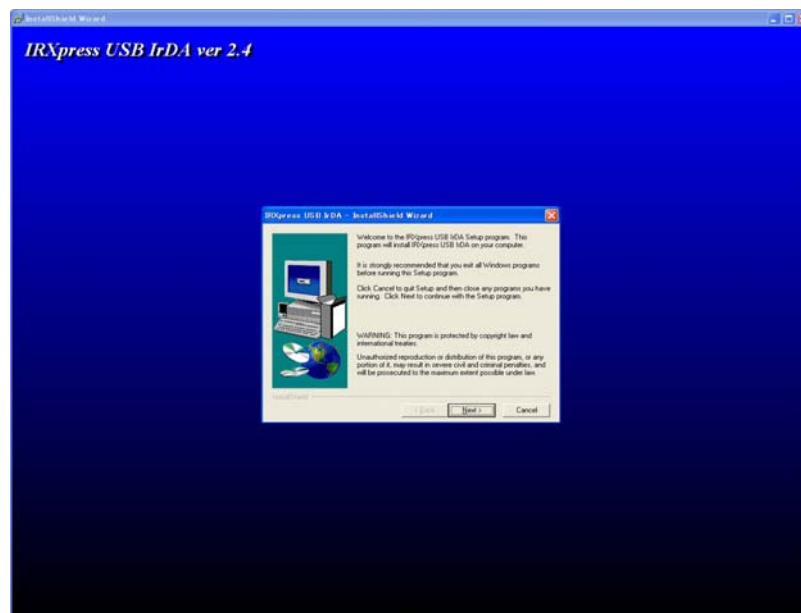


Figure 5.10

3. Click **Next >** button in Figure 5.10. The default installation path of Destination Directory appears in Figure 5.11.

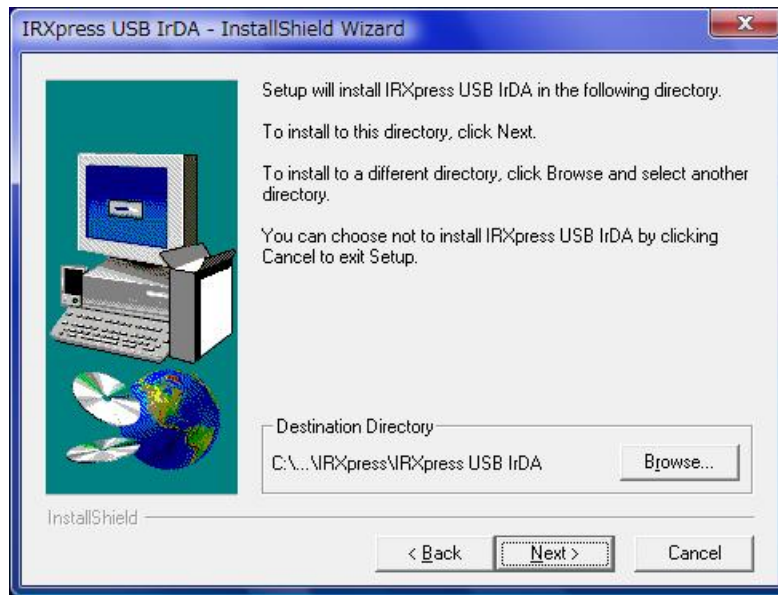


Figure 5.11

4. Specify your own installation path in Figure 5.11 and click **Next >** button. The confirmation screen below appears.

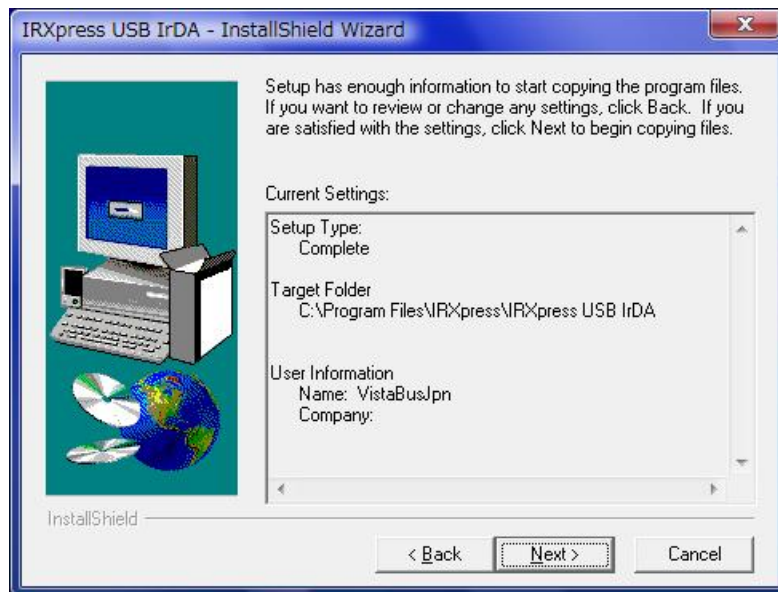


Figure 5.12

5. If all detail is okay in Figure 5.12, click **Next >** button. The screen below appears to indicate the completion of installing the driver.



Figure 5.13

6. Clicking **Finish** button in Figure 5.13 starts up the PC again.
7. After starting up the PC, turn on the power on the cradle. The device recognition is automatically carried out.

The subsequent screens in this page and the next page appear only once when the cradle is connected to the USB port on the PC for the first time after installing the driver.

1. After the PC has started up again, turn on the power on the cradle. The device recognition screen appears.



Figure 5.14

2. Choose **Locate and install driver software (recommended)** in Figure 5.14. The below screen appears.

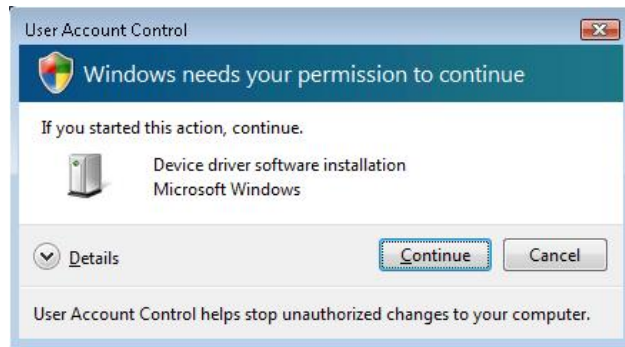


Figure 5.15

3. Click **C**ontinue button in Figure 5.15 and then the below screen appears.

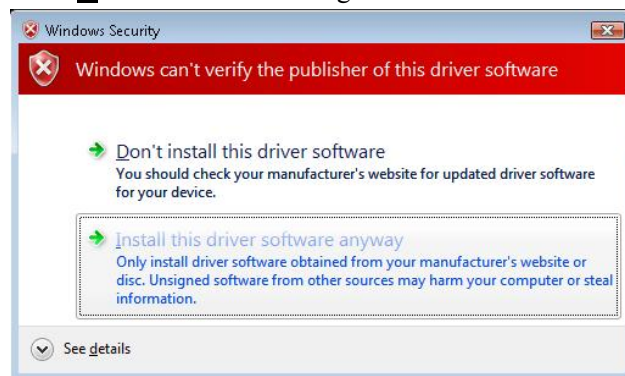


Figure 5.16

4. Choose **I**nstall this driver software anyway in Figure 5.16 and then the below screen appears.

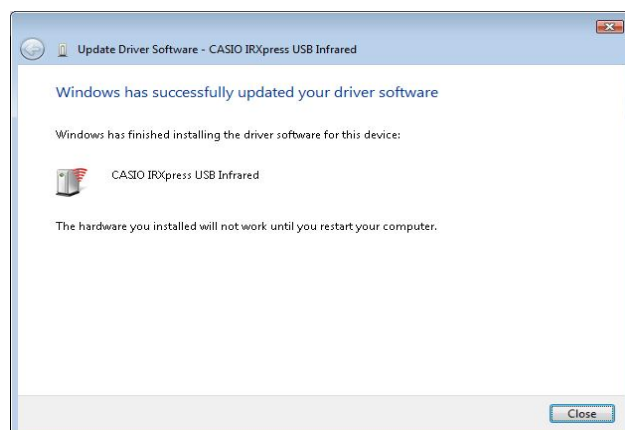


Figure 5.17

5. Click **C**lose button in Figure 5.17.

Setting Up the USB Communication Speed

After installing the USB driver, the communication speed must also be set up. Follow the procedure explained below to set up.

1. First, terminate **ActiveSync** if open.
2. Turn on the power on the cradle connected to PC.
3. Navigate to **Control Panel** → **System**.
4. Choose **Device Manager** in **Hardware** tab. If the cradle is connected to the PC and the power on the cradle is turned on, **CASIO IRXpress USB Infrared** appears in the **Device Manager** (see Figure 5.18).

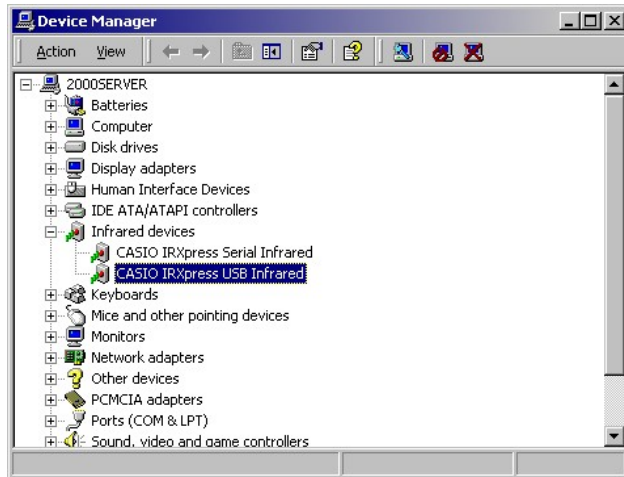


Figure 5.18

Note:

In Windows Vista, a screen asking you to continue with the administrator right appears. Click **Continue** button in the screen to navigate to the Device Manager.

5. Right-click **CASIO IRXpress USB Infrared** in the Infrared devices menu (see Figure 5.18) and then choose **Properties**.

6. Choose **Advanced** tab and then highlight **Maximum Connect Rate**. Choose a baud rate in **Value** field following the note below.

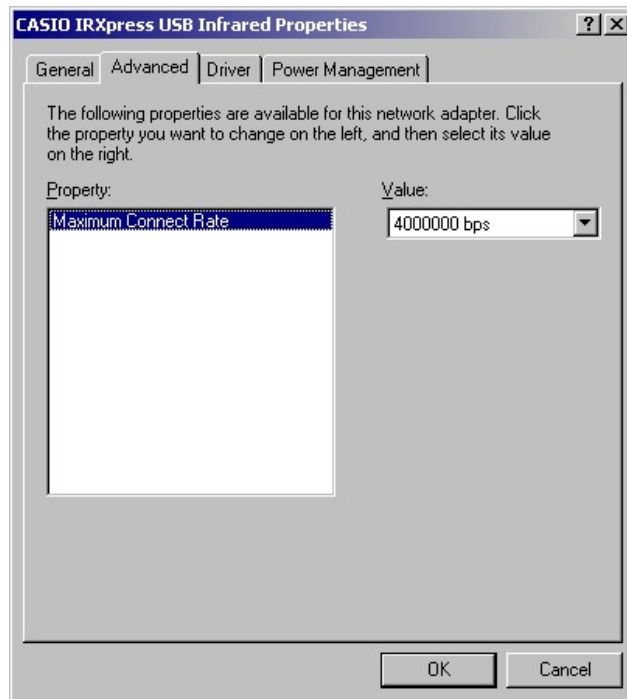


Figure 5.19

Note:

Choose a baud rate depending on the configuration.

In single configuration	: 4000000 bps or 115200 bps
In daisy chained configuration	: 115200 bps only

7. Click **OK** button.

5.1.2 Installing RS-232C Driver

This chapter describes the RS-232C driver installation method by the OS type your PC runs. The communication speed and the port must also be set up after installing the driver. See “Setting Up the RS-232C Communication Speed and Port” on page 39.

Windows Server 2003, Windows XP, Windows 2000

1. Initiate **IRXpressSerialIrDA.exe**.

When the preparation is complete, the background screen will change to InstallShield Wizard screen. The rest of screens in the subsequent installation procedure show only pop-up screens and the background screen are omitted.



Figure 5.20

2. Click **Next >** button in Figure 5.20. The below screen indicating the default path of destination directory appears.



Figure 5.21

3. Specify your own installation path in Figure 5.21 and click **Next >** button. The confirmation screen below appears. Click **Next >** button.

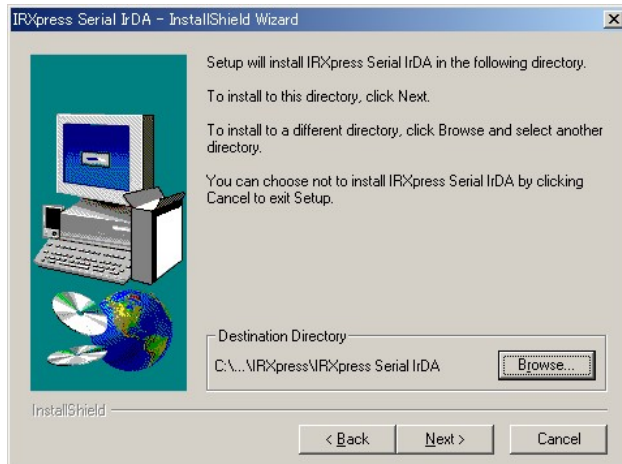


Figure 5.22

4. If all detail appeared in Figure 5.23 is okay, click **Next >** button.

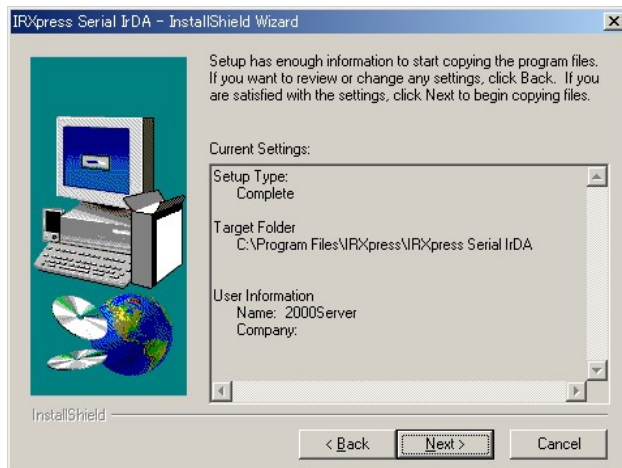


Figure 5.23

5. Clicking **Finish** button starts up the PC again.

Windows Vista

1. Initiate **SetupSerialIrDA.exe**. The User Account Control screen appears.

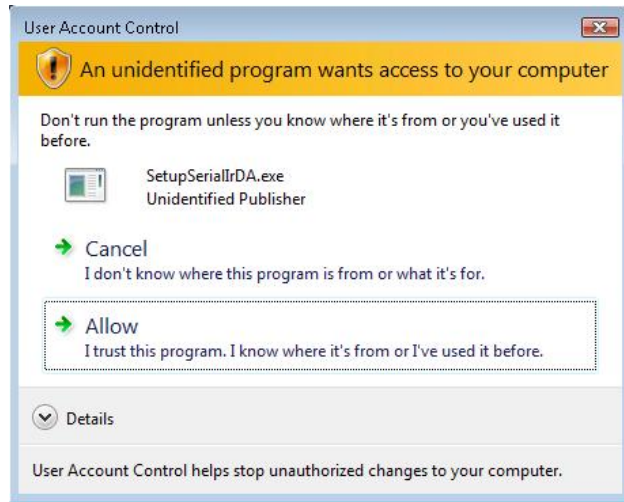


Figure 5.24

2. Choose **Allow** in Figure 5.24 to continue the installation.
When the preparation is complete, the background screen changes to InstallShield Wizard screen. The rest of screens in the subsequent installation procedure show only pop-up screens and the background screen are omitted.



Figure 5.25

3. Click **Next >** button in Figure 5.25. The below screen appears.

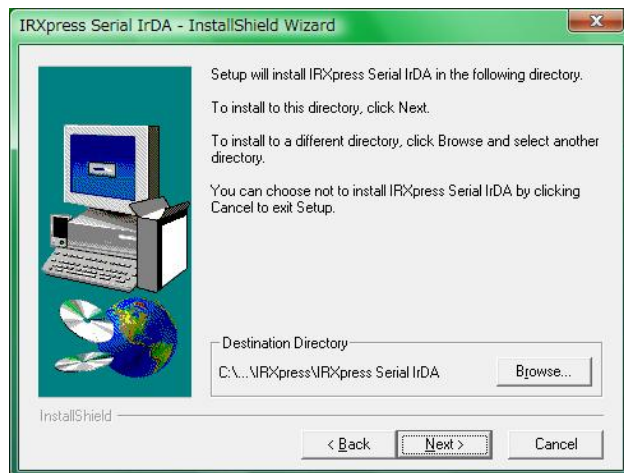


Figure 5.26

4. Specify your own installation path of destination directory in Figure 5.26 otherwise the default and click **Next >** button. The confirmation screen below appears.

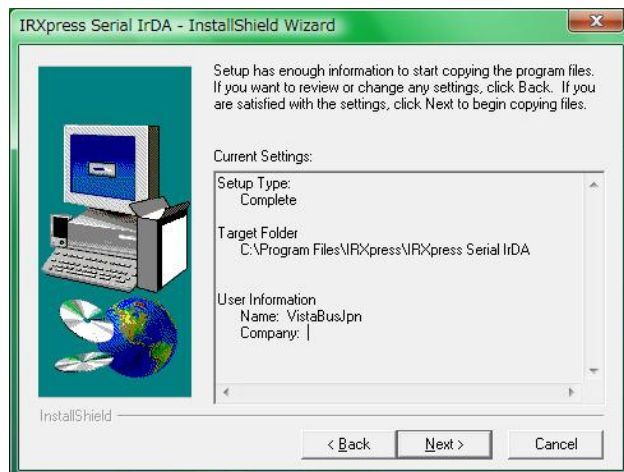


Figure 5.27

5. If all detail in Figure 5.27 is okay, click **Next >** button. The below screen appears.

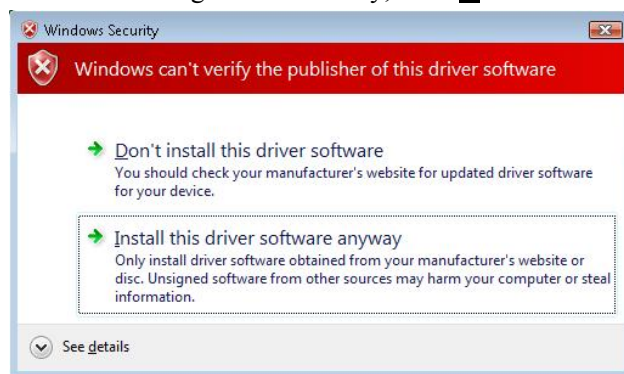


Figure 5.28

6. Choose **Install this driver software anyway** in Figure 5.28. The installation complete screen appears.



Figure 5.29

7. Choose **Yes, I want to restart my computer now** radio button in Figure 5.29 and click **Finish** button. This starts up the PC again.

Setting Up the RS-232C Communication Speed and Port

After installing the RS-232C driver, the communication speed and port must be also set up. Follow the procedure below.

1. First, terminate **ActiveSync** if open.
2. Navigate to **Control Panel** → **System**.
3. Choose **Device Manager** in **Hardware** tab.

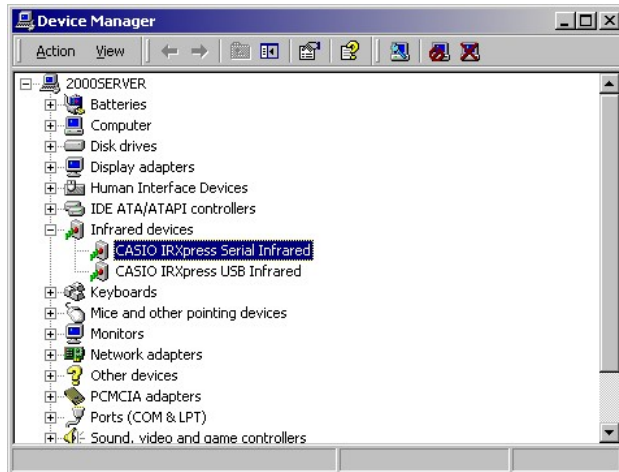


Figure 5.30

Note:

In Windows Vista, a screen asking you to continue with the administrator right appears. Click **Continue** button in the screen to navigate to the Device Manager.

4. Right-click **CASIO IRXpress Serial Infrared** in the Infrared devices menu (see Figure 5.30) and choose **Properties** in the menu.

RS-232C communication speed

5. Choose **Advanced** tab in **Properties** and highlight **Maximum Connect Rate**. Choose 115200 bps in the **Value** field.

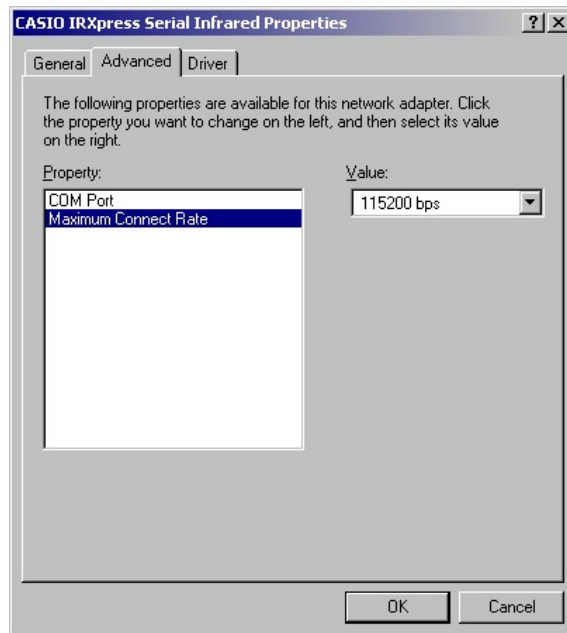


Figure 5.31

RS-232C port

6. Choose **Advanced** tab in **Properties** and highlight **COM Port**. Choose **COM 1** in the **Value** field. Note that the communication port you choose must not be occupied by other application.

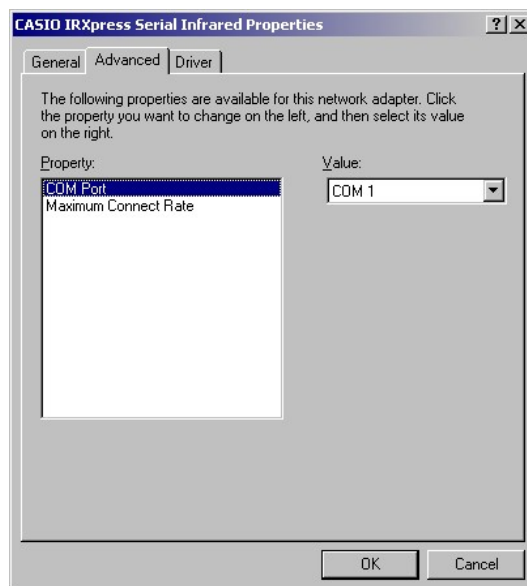


Figure 5.32

7. Click **OK** button in Figure 5.32.

5.2 ActiveSync Connection via IrDA

If you have already installed ActiveSync and connected the IT-3100 to the PC via IrDA, You may skip Chapter 5.2.1. You already have the IrDA driver and ActiveSync in your development environment.

5.2.1 Installing ActiveSync

1. Install ActiveSync first. Run the ActiveSync 'msi' file.

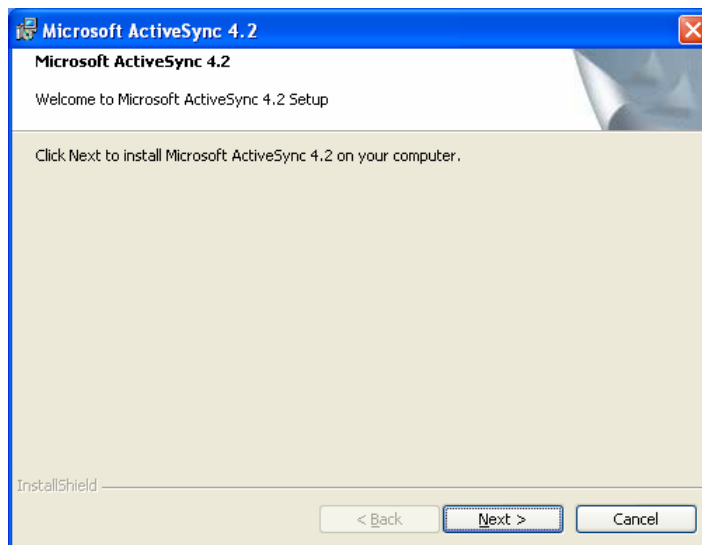


Figure 5.33

2. Click **I**nstall button.

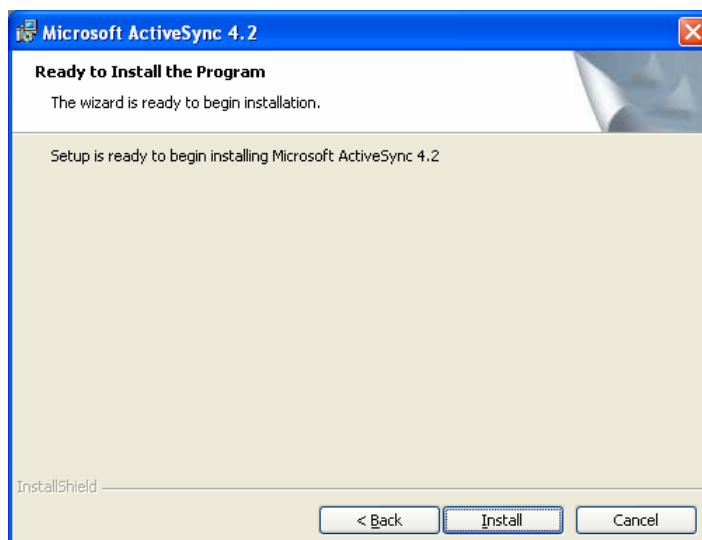


Figure 5.34

3. Click **F**inish button when the installation complete screen appears.

5.2.2 Setting ActiveSync

If already the ActiveSync is installed in your PC, change only the setting in **Connection Settings** ... by following the procedure below.

1. Navigate to **File** → **Connection Settings....** from the menu in ActiveSync. See Figure 5.35.

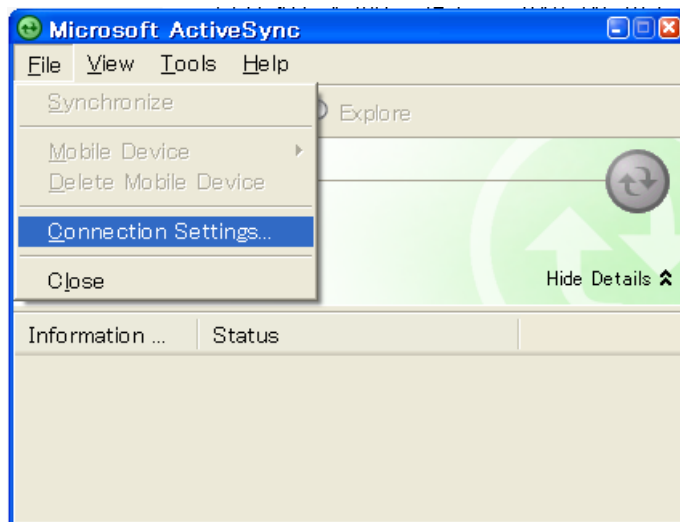


Figure 5.35

2. Check **Allow connections to one of the following**. Choose **Infrared Port (IR)** in the field and then click **OK** button.



Figure 5.36

3. Choose **PC Connection** in the **Control Panel** on the IT-3100.
4. Set **PC Connection** to **IrDA**.
5. Mount the IT-3100 on the cradle.
6. On the IT-3100, navigate to **Start** → **Programs** → **Communication** → **ActiveSync** to start up communication.

5.3 IrDA Connection via Windows Mobile Device Center

To establish connection via IrDA interface with PC runs in Windows Vista, use Windows Mobile Device Center (“WMDC”).

Note:

The WMDC version 6.1 or later supports the connection establishment via IrDA interface. Any other versions earlier are not interoperable with Windows CE devices.

Procedure

Follow the steps below:

1. Choose **PC Connection** in the **Control Panel** on the IT-3100.
2. Set **PC Connection** to **IrDA**.
3. Mount the IT-3100 on the cradle, and then follow a message appeared in the WMDC on the PC.
4. On the IT-3100, navigate to **Start** → **Programs** → **Communication** → **ActiveSync** to start up communication.

6. Setting Up the Development Environment

6.1 Installing CAB Files

1. After installing the library files, the CAB files in Table 6.1 are installed in the folder below.
C:\Program Files\CASIO\MBSYS\CAB

Table 6.1

Library	CAB file	Preinstalled
System Library	en_SystemLib.ARMV4I.CAB	Yes
Bluetooth Library	en_BluetoothLib.ARMV4I.CAB	Yes
Printer Library	en_PrinterLib.ARMV4I.CAB	Yes
Imager Library	en_ImagerLib.ARMV4I.CAB	Yes
MCR Library	en_MCReadLib.ARMV4I.CAB	No
SOBR Library	en_SerialOBRLib.ARMV4I.CAB	No
JPEG library	en_JPEG.ARMV4I.CAB	Yes
FLINK library	en_Flink.ARMV4I.CAB	Yes (note 2)

Notes:

1. The library with "Yes" in "Preinstalled" column is preinstalled in the IT-3100 and in the Device Emulator. Thus, it is not necessary to install it, unless it has been updated or changed.
 2. The CAB file, en_Flink.ARMV4I.CAB, in the table does not operate for the IT-3100. Use the FlinkLib.dll installed by default in the terminal.
2. Copy all the CAB files in Table 6.1 to any folder in the IT-3100 via ActiveSync.
 3. Carry out each CAB file.
 4. When the installation starts, the installation status appears.

6.2 eMbedded Visual C++ 4.0

The **IT-3100 SDK** is required to develop application software with eMbedded Visual C++ 4.0 (see note 1). Follow the steps below to install it.

1. Double click the **IT-3100_SDK.msi** file (see note 2) and follow the prompts that appear on the screen to install the SDK.
2. When prompted whether you want to install **Custom** or **Complete** installation, choose **Complete**.
3. When the installation is finished, start up eMbedded Visual C++ 4.0.
4. Go to Chapter 8 “eMbedded Visual C++” and follow the instruction to verify that the SDK has been installed correctly.

If eMbedded Visual C++ has been installed in your PC already, you will notice that you now have a new SDK and, once you select that new SDK, a new target device (IT-3100) in the comb-box menu in the toolbar. Also, if you use any of the Remote Tools in eVC++ then you will find IT-3100 is listed as a new target (for example, try the Remote Registry Editor). For more details, refer to Chapter 8 “eMbedded Visual C++”.

Notes:

1. If eMbedded Visual C++ 4.0 is used to develop application software, be sure to install Service Pack 4 prior to the development.
2. Other SDKs (e.g. standard SDK etc.) released before the **IT-3100 SDK** are also operable.
3. Application software developed using MFC (Microsoft Foundation Class) for CASIO IT-10 is not operable on the IT-3100.
4. Any application developed not using MFC is operable on the IT-3100.

6.3 Visual Studio 2005

Follow the steps in Chapter 5 “Connecting the IT-3100 to PC” before checking the steps below to confirm that you can connect to the IT-3100 from Visual Studio 2005.

1. Establish connection with the IT-3100 via ActiveSync.
2. Open the application project for VB or C# in Visual Studio 2005.
3. Click the button shown in the red box below (see Figure 6.1) to make sure that Visual Studio 2005 has recognized the connection established with the IT-3100 via ActiveSync. If it does not, start up ActiveSync again to establish connection.

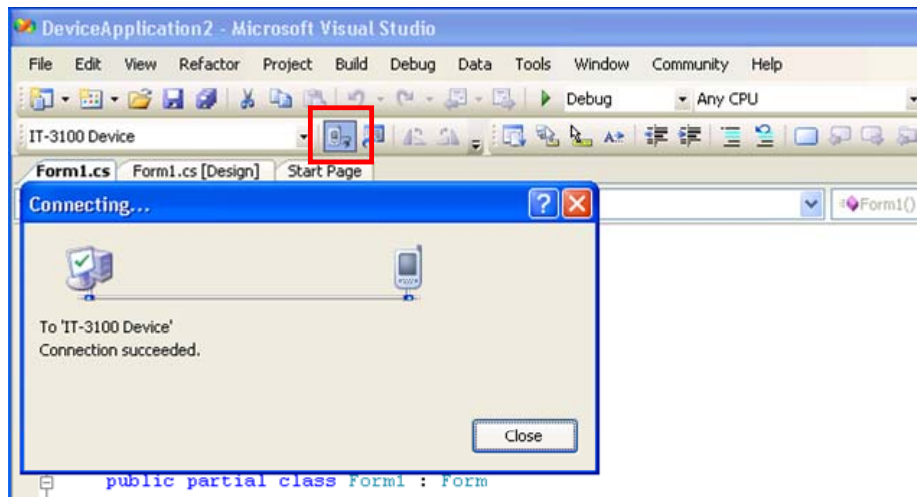


Figure 6.1

4. Choose **IT-3100 Device** in the pull-down menu box in Figure 6.2.

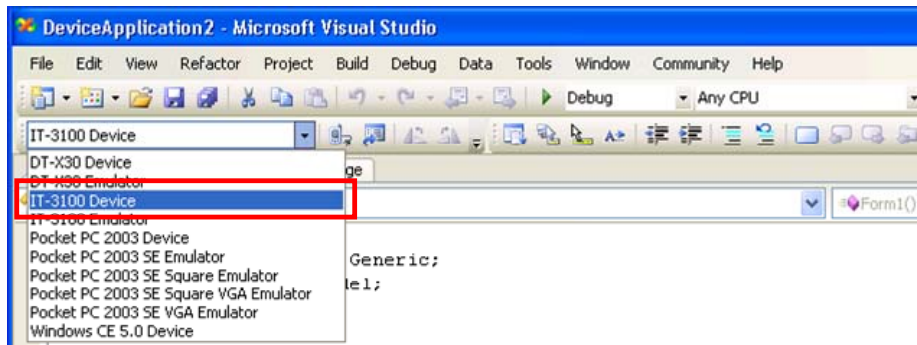


Figure 6.2

5. You will now be able to deploy solutions and also debug applications on the attached IT-3100 using the Visual Studio 2005 debugging features.

6.4 Visual Studio .NET 2003

Follow the steps in Chapter 5 “Connecting the IT-3100 to PC” before checking the steps below to confirm that you can connect to the IT-3100 from Visual Studio .NET 2003. Microsoft has released an add-on for Visual Studio .NET 2003 that allows you to set the target CPU for a connected device (Visual Studio is unable to detect the target CPU of non-Pocket PC devices).

Follow the steps below to install the add-on pack.

1. Download “Windows CE Utilities for Visual Studio .NET 2003 Add-on Pack 1.1” from the site described in Chapter 2.3 “Software Required”.
2. Establish connection via **ActiveSync** between the IT-3100 and PC using any of the methods described in this guide.
3. Navigate to **Tools** → **Select Windows CE Device CPU**.
 - In **Select the device architecture** pull-down menu box, choose ARMV4I.
 - Click **Configure**.
 - Re-start Visual Studio .NET 2003.

You will now, for example, be able to choose **Deploy <appname>** from the **Build** menu and your project will be directly deployed to the IT-3100. You will now also be able to remotely debug applications over your ActiveSync connection.

6.5 Use of IT-3000 Software Legacy

Many of the individual functions of the IT-3100 Libraries are not compatible with those used for the IT-3000 Library functions. The **IT-3100 SDK** does not support compatible library functions with the IT-3000 Library functions. However, your application software developed for IT-3000 can also run on the IT-3100 by building it with the procedure described below. This does not require the source code to change (see note 1).

1. Install the **IT-3100 SDK** in your PC.
2. Copy all header files, import library files and class library files of the libraries provided for IT-3000 into the same folder where the **IT-3100 SDK** is installed (see notes 2 and 3).
3. Read in the Project of application software into the development environment.
4. Specify **IT-3100 Device** as the target of the Project (see note 4).
5. Specify import library file or class library file provided for IT-3000 as the dependant file of the Project (see note 5).
6. Carry out the build.

Notes:

1. It is not recommended to use specific library functions with the function name “TDxxx” of the IT-3000’s CMOS Imager Library which are operable on DT-X10. Rewrite the source code of IT-3000 application software so that the CMOS Imager Library functions of the **IT-3100 SDK** can be operable.
2. The default destination folders to copy the header file, import library file and class library file are as follows.

Header file	C:\Program Files\CASIO\MBSys\INCLUDE
Import Library file	C:\Program Files\CASIO\MBSys\LIB\ARMV4I
Class Library file	C:\Program Files\CASIO\MBSys\WindowsCE
3. When copying these files, if there is any file in the destination folder that is the same file name with the source file do not overwrite it, but use the file of the **IT-3100 SDK** in the destination folder.
4. In Visual Studio .NET 2003, specify **Windows CE Device**.
5. The procedures by development platform are as follows.

C++ (eVC++ 4.0)	: Navigate to Project → Setting ... and then add the objective Import Library file in the Object/Library Module in Link tab.
C++ (VS2005)	: Navigate to Project → Properties of Project name and then add the objective Import Library file in the Add Dependant File by navigating to Configure Properties → Linker → Input .
VB	: Navigate to Project → Adding Reference and then add the objective
C#	Class Library file in Reference tab.

6.6 Use of .NET Application Legacy

This chapter explains how to make a .NET application run on the device installed with Compact Framework which runtime version is as same as the runtime version of Compact Framework which build a .NET application.

The application may not work as designed on the device installed with Compact Framework which runtime version is different from the runtime version of Compact Framework which you build an application, because event order depends on the runtime version of Compact Framework. In this case, following either way forces a application to run on Compact Framework which runtime version is as same as the runtime version of Compact Framework which you built an application and confirm its action.

1. To specify the compatibility mode in the device which install only newer runtime version of Compact Framework.

Create a text file with a file name that has ".exe.config" appended to the name of the application (for example, "appname.exe.config"), and specify the compatibility mode to a text file. The detail is shown below.

Example: To set an application to run on version 1.0

```
<configuration>
  <runtime>
    <compatibilityversion major="1" minor="0"/>
  </runtime>
</configuration>
```

Note:

Save the file to the folder that contains the application.

.NET Framework Developer's Guide

See "How to: Configure Runtime Version" in the following web site for its detail.

[http://msdn.microsoft.com/en-us/library/d5cd9b2c\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/d5cd9b2c(VS.80).aspx)

2. To install the version of Compact Framework runtime
If the application built with Compact Framework version 1.0 runs on the device where the runtime versions 1.0 and 2.0 are installed, the application runs on runtime version 1.0.

7. Device Emulator

The **Device Emulator** provides application developers with an environment that, without having the actual device available, allows them to debug basic functions and performance of an application at source level by stepping through the code.

7.1 Software Required

The **Device Emulator** requires the software(s) listed below before installing the emulator.

Using Visual Studio 2008 or Visual Studio 2005

- ActiveSync 4.2 or a later version (If required)
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- Visual Studio 2008 or Visual Studio 2005 (Required)
- CASIO IT-3100 SDK (Required)
- Standalone Device Emulator 3.0 (Optional. See note 2.)
<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=a6f6adaf-12e3-4b2f-a394-356e2c2fb114>

Using Visual Studio.NET 2003

- ActiveSync 4.2 or a later version (If required)
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- Standalone Device Emulator 3.0 (If required. See notes 1 and 2.)
<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=a6f6adaf-12e3-4b2f-a394-356e2c2fb114>
- Visual Studio .NET 2003 (Required)
- CASIO IT-3100 SDK (Required)

Using eMbedded Visual C++ 4.0

- ActiveSync 4.2 or a later version (Required)
<http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en>
- Standalone Device Emulator 3.0 (If required. See note 1.)
<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=a6f6adaf-12e3-4b2f-a394-356e2c2fb114>
- Microsoft eMbedded Visual C++ 4.0 (Required)
<http://www.Microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=1DACDB3D-50D1-41B2-A107-FA75AE960856>
- Microsoft eMbedded Visual C++ 4.0 ServicePack4 (Required)
<http://www.microsoft.com/downloads/details.aspx?familyid=4A4ED1F4-91D3-4DBE-986E-A812984318E5&displaylang=en>
- CASIO IT-3100 SDK (Required)

Notes:

1. The software is not required if your PC already has **Visual Studio 2005** installed.
2. To use Microsoft Device Emulator 3.0, follow the steps below.
 - 2-1. Install IT-3100 ExportSDK and Device Emulator in IT-3100 SDK.
 - 2-2. Download Microsoft Device Emulator 3.0 and install it.
 - 2-3. Edit the following file and save the change made.

C:\Program Files\Windows CE Tools\wce500\IT-3100\Emulation\IT-3100.cdes

Before you change, the default parameter in the file is described as follows.

`module=DevEmu500.exe`

Change the parameter to the one below. Be sure to describe the whole parameter in single one line.

`module=C:\Program Files\Microsoft Device
Emulator\1.0\DeviceEmulator.exe`

You can substitute Device Emulator Version 3.0 released in Visual Studio 2008 for an engine of Device Emulator included in the IT-3100 SDK by the procedure above.

Case of using after Windows Vista OS

If you use OS after Windows Vista (Windows 7 or Windows Server 2008), please set device emulation and I/O simulator as administrator.

Please check “Run this program as an administrator” of property from opening Explorer.
(Ex. Execute I/O simulator in Windows Vista as administrator.)

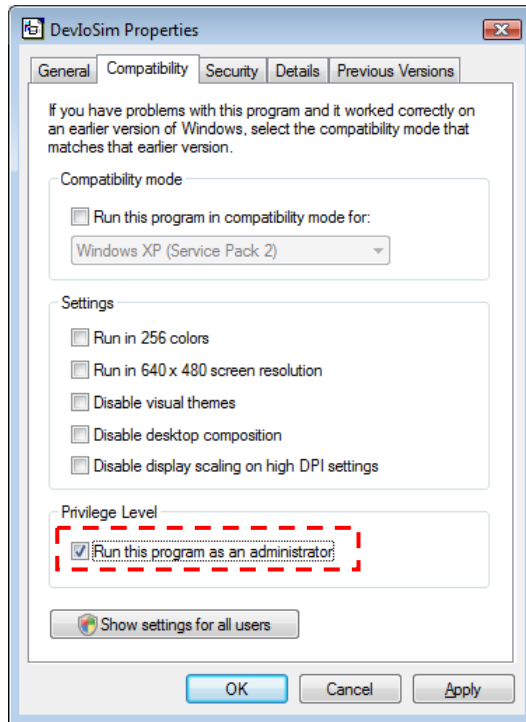


Figure 7.1

These program have installed the following location as default setting.

- Device emulator
C:\Program Files\Microsoft Device Emulator\1.0\DeviceEmulator.exe
- I/O simulator
C:\Program Files\Common Files\CASIO\Emulator\DevIoSim.exe

7.2 Starting up the Device Emulator

After installing all required software described in Chapter 7.1, follow the steps below to start up the **Device Emulator** on your PC.

1. Navigate to **Start** menu → **All Programs** → **CASIO Device Emulator** and click **IT-3100**.
2. Make sure that the IT-3100 **Device Emulator** has started up on the screen. See Figure 7.2. If Figure 7.2 and Figure 7.3 appear on your PC, you are ready to use the emulator.

Note.

If do not appear "IT-3100" in "CASIO Device Emulator", please execute "Add Device".

If you use OS after Windows Vista, please execute "DevEmuLoader.exe" as administrator by referring "Case of using after Windows Vista OS" section.

C:\Program Files\Common Files\CASIO\Emulator\DevEmuLoader.exe



Figure 7.2 IT-3100 Device Emulator

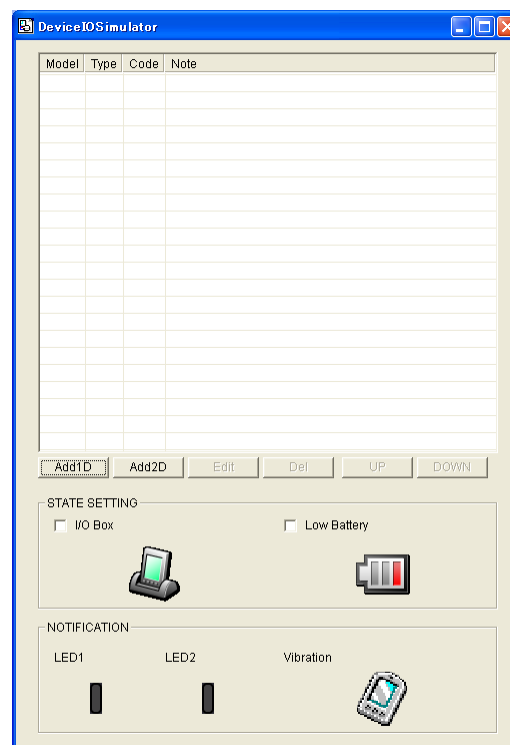


Figure 7.3 I/O Simulator

Terminology of **Emulator** and **Simulator**;

The **Emulator** described in this reference manual is a software application that behaves in a very similar way to the actual device by imitating individual hardware components or protocols present in the actual hardware.

On the other hand, the **Simulator** is also a software application that logically integrates application programming interfaces ("API") and certain other functions to allow debugging of the application

program using external events. The **Emulator** performs in a pseudo CPU and hardware environment and it is impossible for the application to recognize whether it is in the actual device environment or pseudo environment. However, actions carried out by the **Simulator** are not as alike to those performed by actual components but merely mimic them very closely.

7.3 Using the Device Emulator

7.3.1 IT-3100 Device Emulator

The **IT-3100 Device Emulator** emulates various operations carried out by the actual IT-3100 device on the PC's screen such as mouse operation, input on PC's keyboard, displaying execution of applications, and operations by actual devices such as the scanner. Figure 7.4 shows an emulated IT-3100 device on the screen of a PC.



Figure 7.4

Key Input

The emulator offers key input capability similar to that of the actual IT-3100 device. For instance, a key on the emulated keyboard of IT-3100 on the screen (see Figure 7.4) can be clicked with the PC mouse as well as key input made directly on the PC's keyboard.

Reading Bar Codes

The emulator enables bar codes pre-registered in the I/O Simulator (see Figure 7.3) to be input when clicking Trigger key on the emulated keyboard (see Figure 7.4). Note however that the Trigger key must be continuously pressed for a second or more otherwise an incorrect key input may result.

Sound

The emulator offers beep and sound capability similar to that of the actual IT-3100 device.

7.3.2 I/O Simulator

The I/O Simulator simulates registration of bar codes, generation of low battery warning, detection of terminal being mounted on cradle.

Registration of bar code symbologies

1. Registration

Click **ADD1D** or **ADD2D** button (circled in red in Figure 7.5) to go into the bar code registration mode.

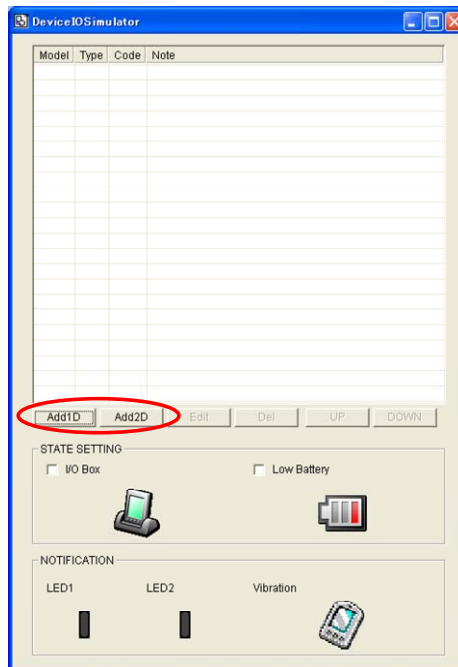


Figure 7.5

2. Bar code registration

Choose a bar code symbology in the **Code Type** pull-down menu that you wish to register in the **I/O Simulator**.

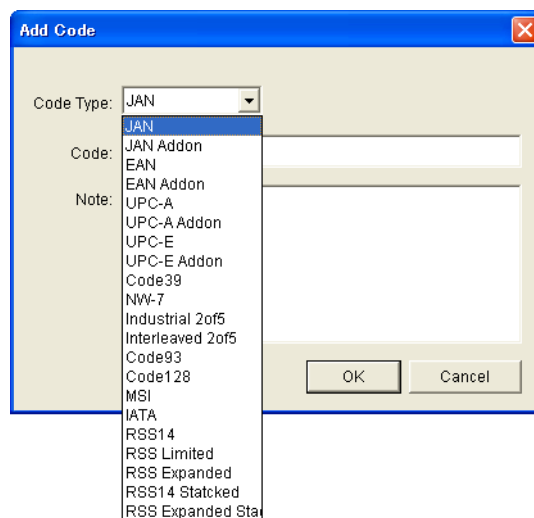


Figure 7.6

- Enter bar code data in the **Barcode** field (see Figure 7.7) and a note about the bar code in the **Note** field if necessary. Click **OK** button to complete the bar code registration.

Figure 7.7

- After completion of the bar codes registration, the screen in Figure 7.8 shows a list of bar codes that have been registered in the **I/O Simulator**. Prior to debugging with the **Device Emulator**, make sure that you register all bar codes you wish to use in debugging.

Figure 7.8

5. Editing registered bar code content

Highlight a bar code in the list of registered bar codes (see Figure 7.8) and click **Edit** button. Figure 7.9 appears for editing the bar code and its information.

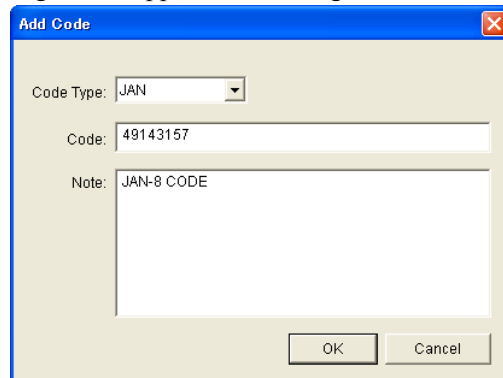
A dialog box titled "Add Code" with a blue header bar and a close button (X) in the top right corner. The dialog has a light beige background. It contains three input fields: "Code Type:" with a dropdown menu showing "JAN", "Code:" with a text box containing "49143157", and "Note:" with a larger text box containing "JAN-8 CODE". At the bottom right, there are two buttons: "OK" and "Cancel".

Figure 7.9

6. Deleting registered bar code content

Highlight a bar code in the list of registered bar codes (see Figure 7.8) and click the **Del** button. Dialogue screen in Figure 7.10 appears for you to confirm the deletion. If it is okay to delete, click **Yes** button, otherwise click **No** button.

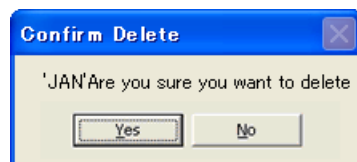
A dialog box titled "Confirm Delete" with a blue header bar and a close button (X) in the top right corner. The dialog has a light beige background. It contains a text box with the text "'JAN'Are you sure you want to delete". At the bottom, there are two buttons: "Yes" and "No".

Figure 7.10

Detection of Terminal in Cradle and Low Battery Warning

If you check the **I/O Box** and **Low Battery** boxes in STATE SETTING field (see Figure 7.11), the simulator simulates the respective events in the emulator.

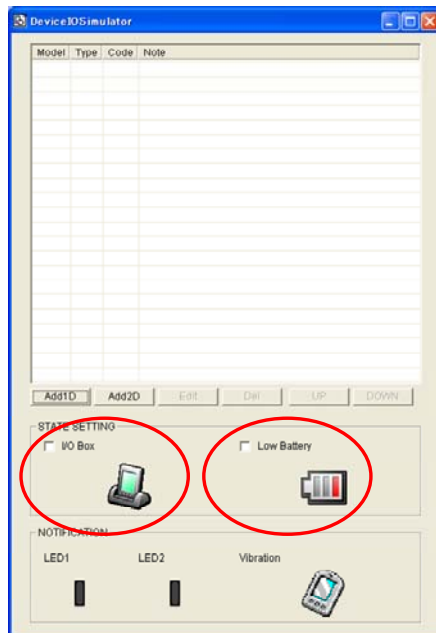


Figure 7.11

I/O Box

If this box is checked, a notification is issued that the connection between the IT-3100 **Device Emulator** and cradle has been established. This notification can be utilized by the application.

Low Battery

If this box is checked, a notification that a low battery state has occurred is raised. The icon in the toolbar in the emulated screen appears too. The notification can be utilized by the application to recognize the low battery state in the hardware.

Indications

The I/O Simulator expresses a change of state that occurred in the IT-3100 **Device Emulator**.

- LED

When the IT-3100 **Device Emulator** turns on the LED, the LED icon (LED2) in the I/O Simulator also turns on. See Figure 7.11.

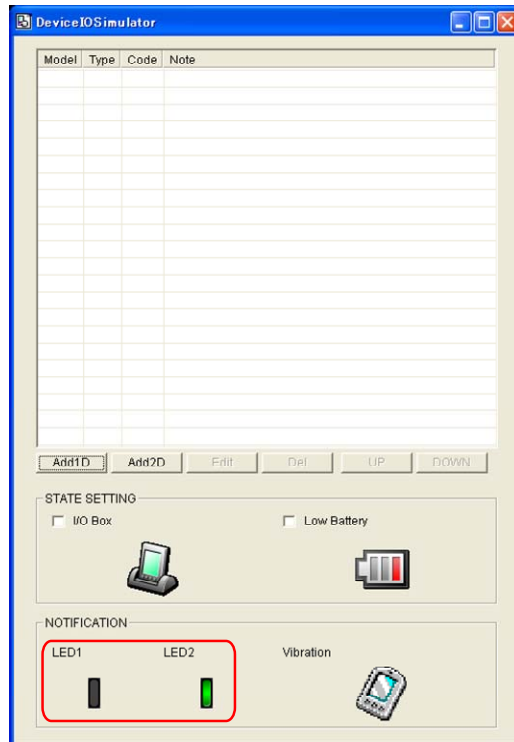


Figure 7.12

Note:

The vibration on the IT-3100 Device Emulator does not take effect.

7.3.3 Connecting via ActiveSync

If debugging with the **Device Emulator** is carried out in eMbedded Visual C++ 4.0 or Visual Studio 2008 or Visual Studio 2005, or transmission/reception of a file with the **Device Emulator** is carried out, **ActiveSync** must be used.

Setting ActiveSync

1. Start up **ActiveSync** and then navigate to **File** → **Connection Settings**

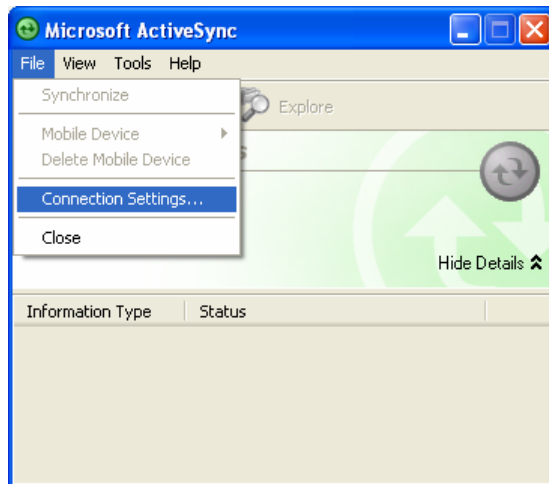


Figure 7.13

2. In **Connection Settings** screen, check in the **Allow connections to one of the following** box and choose **DMA** in the pull-down menu. See Figure 7.14.



Figure 7.14

Connection via ActiveSync

The ways to establish connection of the **Device Emulator** via **ActiveSync** are;

- Using Visual Studio 2008 or Visual Studio 2005
 - Using Standalone Device Emulator 3.0 (if Visual Studio 2008 or Visual Studio 2005 is not available.)
1. Start up the **Device Emulator** by referring to Chapter 7.2 “Starting up the Device Emulator”.
 2. Start up Visual Studio 2008 or Visual Studio 2005, and then navigate to **Tools** → **Device Emulator Manager**.
 3. Right-click **IT-3100 Emulator** in **Available Emulators** list and then choose **Cradle** in the popup menu. See Figure 7.15.

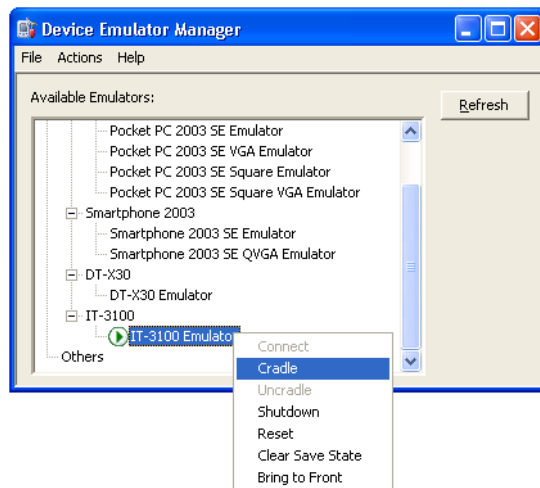


Figure 7.15


4. Make sure ActiveSync has started up and the  icon in the status bar appears. See the emulated screen of IT-3100 in Figure 7.15. The icon indicates that the connection via ActiveSync has been established. Figure 7.16 simultaneously appears.



Figure 7.16

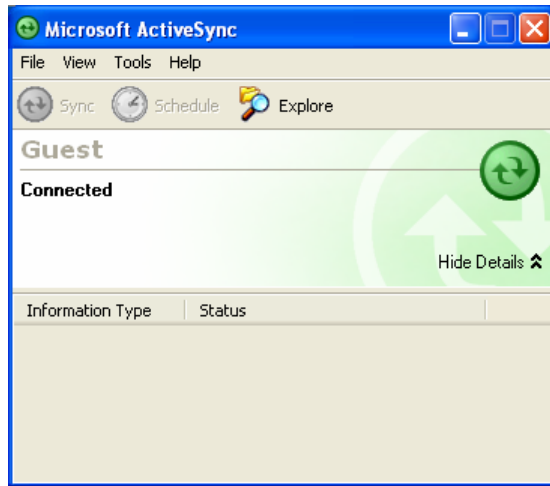


Figure 7.17

7.4 Debugging Applications

This chapter describes how to debug your application using the **Device Emulator**. Before starting to “Build”, establish a connection between the IT-3100 and your PC via ActiveSync by referring to Chapter 7.3 “Using the Device Emulator”.

For the basic order of developing an application, refer to Chapters 8 “eMbedded Visual C++” and 9 “Visual Studio”.

7.4.1 Setting Up Build Configuration

When using Visual Studio 2008 or Visual Studio 2005

- Choose **Debug** in the Solution Configurations pull-down menu in Visual Studio 2008 or Visual Studio 2005 and **IT-3100 Emulator** in the target device pull-down menu. See Figure 7.18.

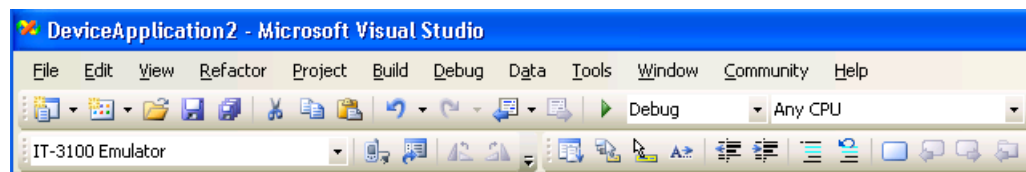


Figure 7.18

When using eMbedded Visual C++ 4.0

- Choose **IT-3100** in the Active WCE Configuration in the pull-down menu in eMbedded Visual C++ 4.0, **Win32 (WCE ARMV4I) Debug** in the Active Configuration pull-down menu, and **IT-3100 Device** in the Default Device pull-down menu.

7.4.2 Debugging Applications

Basic Debug Operation

The debug operation used for the **Device Emulator** in both Visual Studio 2005 and eMbedded Visual C++ 4.0 is the same as an ordinary debug operation using the actual terminal.

Debugging with the Device Emulator

With the **Device Emulator**, it is possible to set a break point in the source code of the application for step-by-step debugging.

When using Visual Studio 2008 or Visual Studio 2005

1. Navigate to **Debug** menu → **Start Debugging** to start up the debugger.

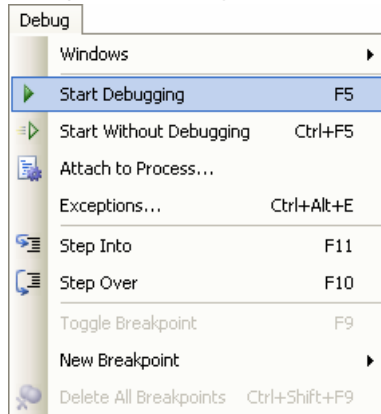


Figure 7.19

2. Similar to ordinary debugging operations with an actual IT-3100, the **Device Emulator** allows break point setting (circled in red in Figure 7.20) in the source code and step-by-step debugging.

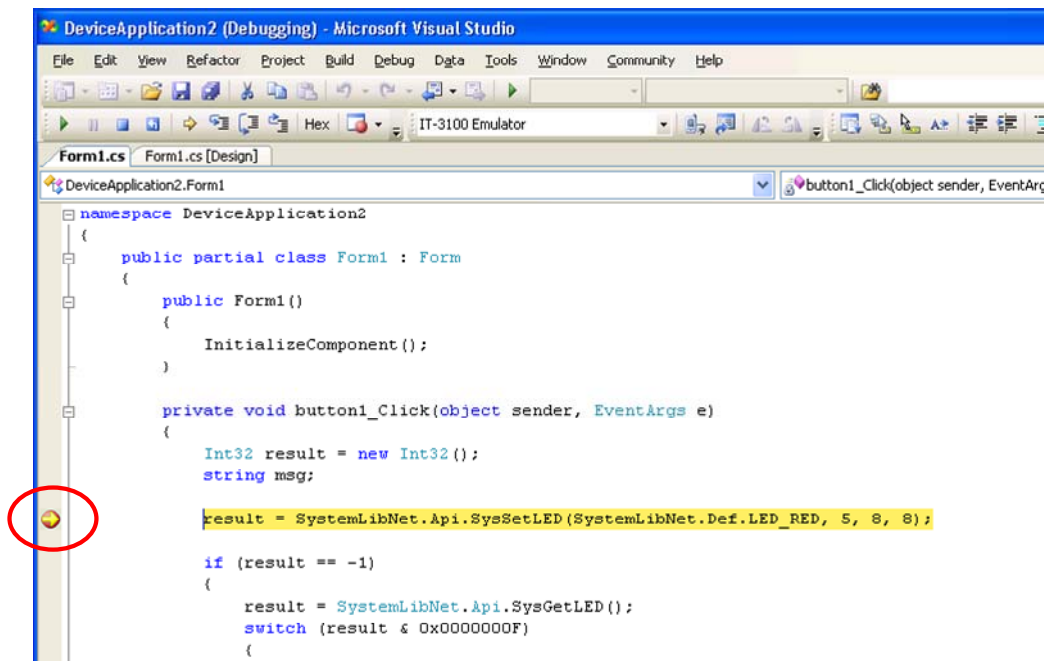


Figure 7.20

Note:

If the debugging operation does not stop at the break point, install Microsoft .NET Compact Framework 2.0 SP2 in your PC.

When using eMbedded Visual C++ 4.0

1. Establish a connection between the **Device Emulator** and PC via ActiveSync before starting debugging in eMbedded Visual C++ 4.0. For establishing connection via ActiveSync, refer to Chapter 7.3.3 “Connecting via ActiveSync”.
2. Navigate to **Build** in the menu bar → **Start Debug** → **Go** to start up debugging.

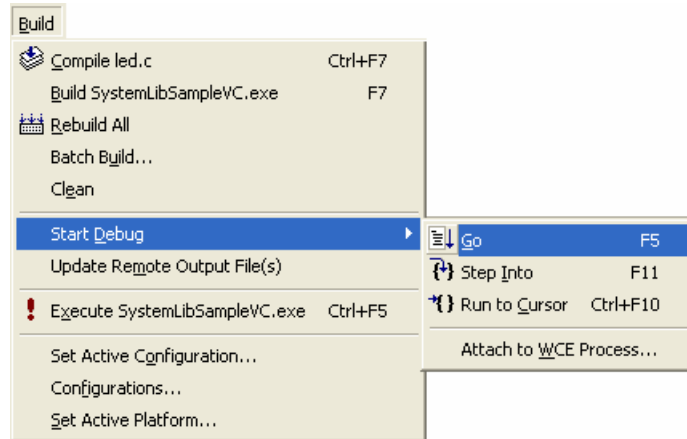


Figure 7.21

3. Similar to ordinary debugging operations with an actual IT-3100, the **Device Emulator** allows break point setting (circled in red in Figure 7.22) in the source code and step-by-step debugging.

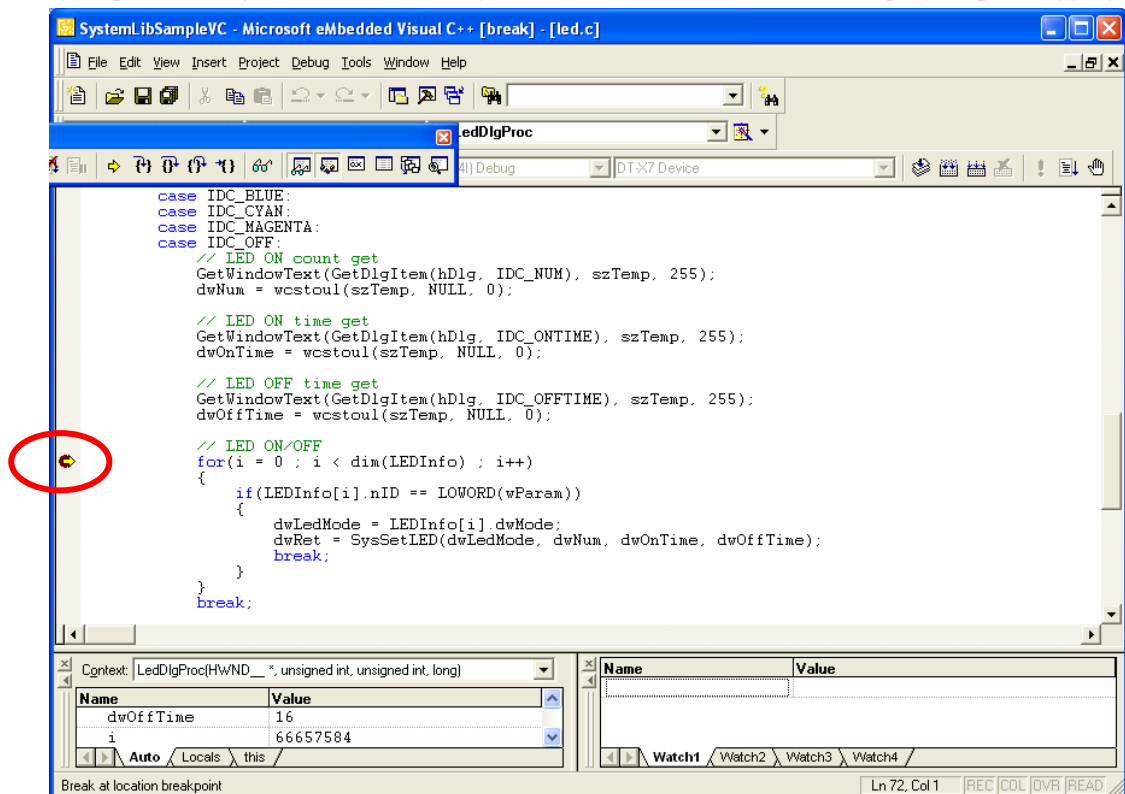


Figure 7.22

8. eMbedded Visual C++

8.1 Building a Simple eVC++ 4.0 Test Program

1. On the PC, initiate eVC++ 4.0.
2. Navigate to **File** → **New** in the **File** menu.

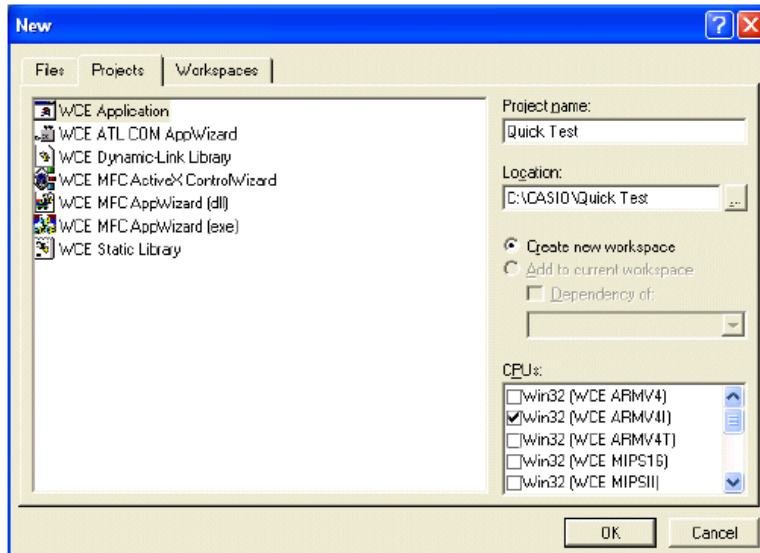


Figure 8.1

3. Highlight **WCE Application** in **Projects** tab. See Figure 8.1.
4. In **Location:** field, navigate to the folder where you want to create the new project.
5. In **Project name:** field, type in the name of the project.
6. In **CPUs:** pull-down menu, check **Win32 (WCE ARMV4I)**.
7. In the next dialog, leave **A typical “Hello World!” application** checked. Click **Finish** and then click **OK** button.
8. Below the Toolbar you will see a line of pull-down menu lists. Look for the one that indicates **the target devices available to you** and select **IT-3100** in the list. The right most pull-down menu list will change to **IT-3100 Device**.



Figure 8.2

9. Initiate an ActiveSync connection using one of the methods described in this guide.
10. Choose **Rebuild All** from **Build** menu (or use the appropriate icon on the Toolbar).

11. The program will be built and automatically downloaded to the IT-3100. By default, the program will be copied to the root folder on the terminal. Run the program to check that the process was successful. Note that for this basic example you will need to use the mouse emulator to close the application on the device.

You are now ready to begin development work with the IT-3100. The full operation of eVC++ 4.0 and the use of features such as remote debugging are beyond the scope of this guide. See the Chapter 10 “Resources” for detail on where to start if you are new to eMbedded Visual C++ development.

8.2 Using CASIO Libraries from eVC++ 4.0

The **IT-3100 SDK** provides the libraries for C++ applications. See Chapter 1.2 “Library Configuration” for the list of libraries provided.

Each library consists of a header file, a dll and a library file. The dlls are built into the ROM of the IT-3100 and you do not need to download them. The following is a simple example using one of the System Library functions in the simplest kind of WindowsCE program. The screen will flip 180° each time this program is executed.

1. Make sure all the “.h” files for the CASIO libraries are installed in **C:\Program Files\CASIO\MBSys\INCLUDE**. This is the default location but yours will be different if you install the SDK to another location.
2. Make sure all the “.lib” files for the CASIO libraries have been installed in **C:\Program Files\CASIO\MBSys\LIB\ARMV4I**
3. In eVC++ 4.0, select **New** from the **File** menu.
4. Highlight **WCE Application**. Choose a location and a name for the project and make sure the ARMV4I option is checked. Click **Next >** button.
5. On the next dialog click **A simple Windows CE application** option. Click **Finish** and then click **OK** button.
6. Make sure the IT-3100 is the selected SDK and target device in the pull-down list boxes.
7. Click **ClassView** tab in **Solution Explorer** and expand **classes** fully until you can see **WinMain()** class. Double click it.
8. At the top of the source file, under **#include stdafx.h** code add the following code.

```
#include <SystemLib.h>
```

9. Move to the line **// TO DO: Place code here** and add the following code.

```
int result;

result = SysGet180Rotate();
if( result == FALSE )
{
    SysSet180Rotate( TRUE );
}
else
{
    SysSet180Rotate( FALSE );
}
```

10. On the **Project** menu, navigate to **Add To Project** → **Files...**
11. Change **Files of type:** pull-down menu list to **Library files (.lib)**.
12. Navigate to the folder where the CASIO library files are stored (see step no. 2 in the previous page), highlight **SystemLib** file and click **OK** button.
13. Initiate **ActiveSync** to establish connection between the IT-3100 and the PC.
14. Choose **Build All** from **Build** menu. The project will be built and copied to the IT-3100.
15. Check the operation by running the program. You should find that each time the program is carried out, the screen flips 180°.

You should now be able to use the CASIO System Library in your C++ applications. See the Common Device Control Library manual for full detail of all the functions.

You can also use the CASIO samples as described in Chapter 1.4.

9. Visual Studio

This chapter explains the development procedure of the application that uses the following development platforms.

- Visual Studio 2008
 - Visual Studio 2005
 - Visual Studio .Net 2003
-
- The functions of the Common Device Control Library which control various individual devices integrated in the IT-3100 have different name spaces and names from those available for the previous CASIO handheld terminals. They are not compatible with the ones in the previous CASIO library. The exceptions are DT-X11 and IT-600 which do use the same libraries.
 - VB or C# applications developed with Visual Studio .NET 2003, but not with the CASIO dedicated libraries will run on the IT-3100.
 - Applications developed with any functions of CASIO dedicated library must be rewritten by replacing the dedicated functions with the appropriate functions from the Common Device Control Library.
 - New applications for the IT-3100 can be developed using either VB or C# in Visual Studio 2005.
 - It is recommended that Visual Studio .NET 2003 is used to modify applications developed using that tool for other handheld terminals when porting them to the IT-3100.
 - However, if Visual Studio 2005 is used to make modifications, the solution/project from Visual Studio .NET 2003 is automatically upgraded by Visual Studio 2005. This may result in different configuration of the output folder according to the parameter settings for the project file. Please take care when using Visual Studio 2005 to upgrade an existing application and always keep a backup of your existing project.

9.1 Using CASIO .NET Libraries from VB

The **IT-3100 SDK** provides the libraries for VB applications. See Chapter 1.2 “Library Configuration” for the list of libraries provided.

This chapter explains how to create and start up a simple test program using one of the System functions. The test program flashes the LED in red for 5 seconds.

1. Create a new VB Smart Device Application in Visual Studio.
2. Click Add Reference... from Project menu.
3. Click **Browse** and navigate to the folder where you have stored the CASIO .NET library files. Highlight **SystemLibNet.dll** and click **Open**. Click **OK** button.

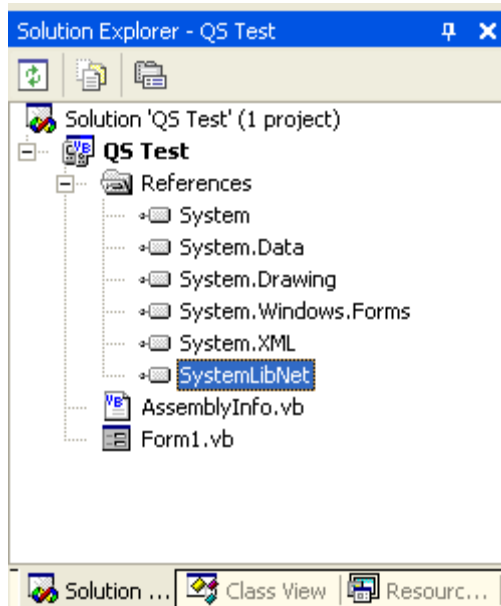


Figure 9.1

4. Add a button to the form, and double click it.

5. In the event function for the button click, add the following code.

```
Dim result As Int32
Dim msg As String

result = Calib.SystemLibNet.Api.SysSetLED( _
    Calib.SystemLibNet.Def.LED_RED, 5, 8, 8)

'.NET vales of "true" and "false" are "-1" and "0" respectively.
If result = -1 Then
    result = Calib.SystemLibNet.Api.SysGetLED()
    Select Case (result And &HF)
        Case Calib.SystemLibNet.Def.LED_OFF
            msg = "LED_OFF"
        Case Calib.SystemLibNet.Def.LED_RED
            msg = "LED_RED"
        Case Calib.SystemLibNet.Def.LED_GREEN
            msg = "LED_GREEN"
        Case Calib.SystemLibNet.Def.LED_ORANGE
            msg = "LED_ORANGE"
        Case Else
            msg = "LED_UNKNOWN"
    End Select
    MessageBox.Show(msg, "LED")
End If
```

Note:

If you type this code manually you should see the IntelliSense offer you suitable options as appropriate. If you do not, make sure you review steps 1 to 5 to make sure you have added the reference correctly.

6. Initiate **ActiveSync** to establish connection between the IT-3100 and PC.
7. Choose **Deploy <name of project>** on **Build** menu.
8. The project will be built and copied to the IT-3100. By default, it will be copied to **\Program Files\<name of project>** folder. **SystemLibNet.dll** will be deployed to the same folder.
9. Check that the program works correctly on the IT-3100.

You can also use the CASIO samples as described in Chapter 1.4.

9.2 Using CASIO .NET Libraries from C#

The **IT-3100 SDK** provides the libraries for C# applications. See Chapter 1.2 “Library Configuration” for the list of libraries provided.

This chapter explains how to create and start up a simple test program using one of the System functions. The test program flashes the LED in red for 5 seconds.

1. Create a new C# Smart Device Application in Visual Studio .NET.
2. Click Add Reference... from Project menu.
3. Click **Browse** and navigate to the folder where you have stored the CASIO .NET library files. Highlight **SystemLibNet.dll** and click **Open**. Click **OK** button.

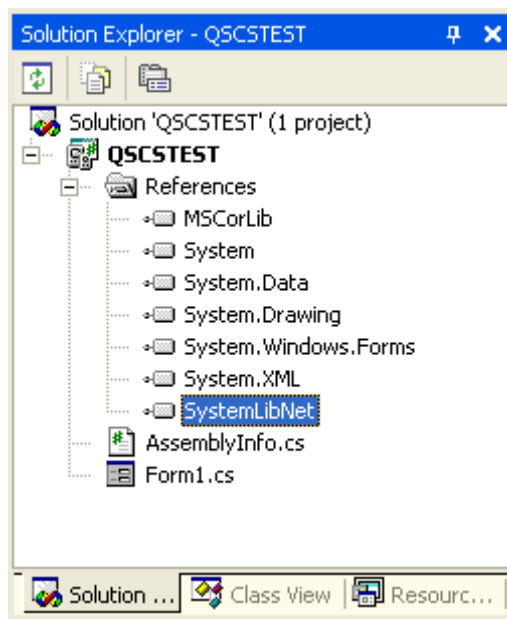


Figure 9.2

4. At the top of your source file add the following code.

```
using Calib;
```

5. Add a button to your form, double click it and add the following code:

```
Int32 result = new Int32();
string msg;

result = SystemLibNet.Api.SysSetLED(SystemLibNet.Def.LED_RED, 5,
8, 8);

// .Net values "true" and "false" are "-1" and "0" respectively.
if(result == -1)
{
    result = SystemLibNet.Api.SysGetLED();
    switch(result & 0x0000000F)
    {
        case SystemLibNet.Def.LED_OFF:
            msg = "LED_OFF";
            break;
        case SystemLibNet.Def.LED_RED:
            msg = "LED_RED";
            break;
        case SystemLibNet.Def.LED_GREEN:
            msg = "LED_GREEN";
            break;
        case SystemLibNet.Def.LED_ORANGE:
            msg = "LED_ORANGE";
            break;
        default:
            msg = "LED_UNKNOWN";
            break;
    }
    MessageBox.Show( msg, "LED" );
}
```

Note:

If you add this code manually you should see the IntelliSense offer you suitable options as appropriate. If you do not see this, then review steps 2 to 4 in the previous page to make sure you have not made a mistake.

6. Initiate **ActiveSync** to establish connection between the IT-3100 and PC.
7. Choose **Deploy <name of project>** on **Build** menu.
8. The project will be built and copied to the IT-3100. By default, it will be copied to **\Program Files\<name of project>** folder. SystemLibNet.dll will be deployed to the same folder.
9. Check that the program works correctly on the device.

You can also use the CASIO samples as described in Chapter 1.4.

9.3 Using CASIO Libraries from C++

This chapter explains how to create and start up a simple test program using one of the System functions. The test program flashes the LED in red for 5 seconds.

Before following the steps below, you will need to add the paths for the CASIO header and library files to Visual Studio 2008 or Visual Studio 2005. Choose **Tools-Options-Projects and Solutions-VC++ Directories**. Change the “Platform” to “IT-3100” and add the path to the CASIO header and library files to the respective lists for “Include files” and “Library files”.

1. First, create **Win32 Smart Device Project** with its device name **DeviceApp** for Visual C++ in Visual Studio 2008 or Visual Studio 2005.
2. Choose default platform in the Selected SDKs field in the Platforms of Win32 Smart Device Project Wizard and click < button.

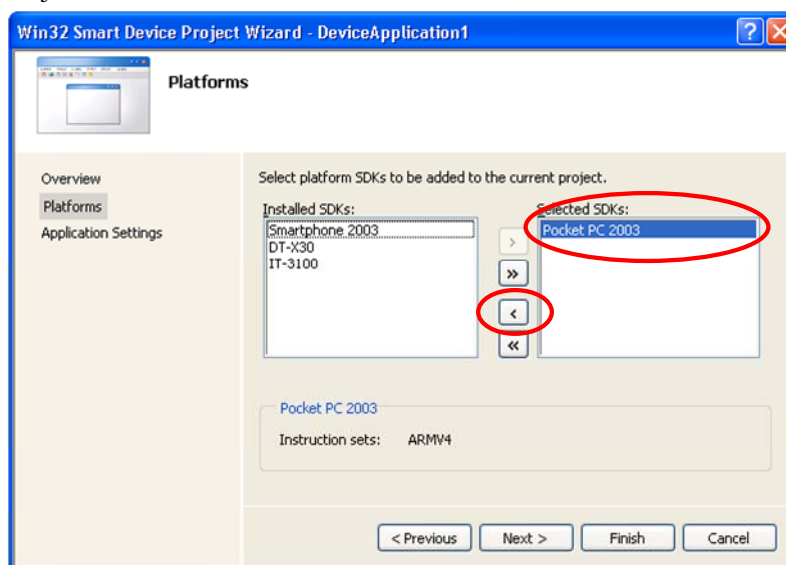


Figure 9.3

3. Choose **IT-3100** in the Installed SDKs field and click **>** button.

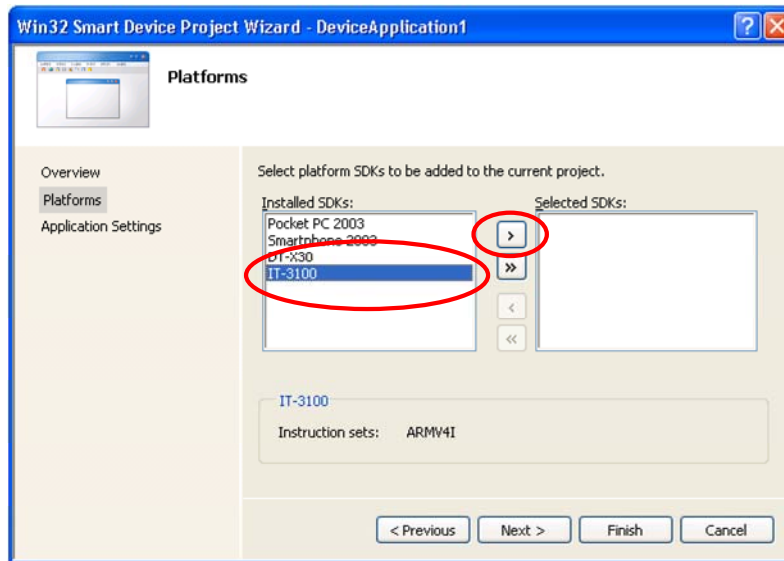


Figure 9.4

4. Click **Next >** button while **IT-3100** is kept being highlighted in the Selected SDKs field.

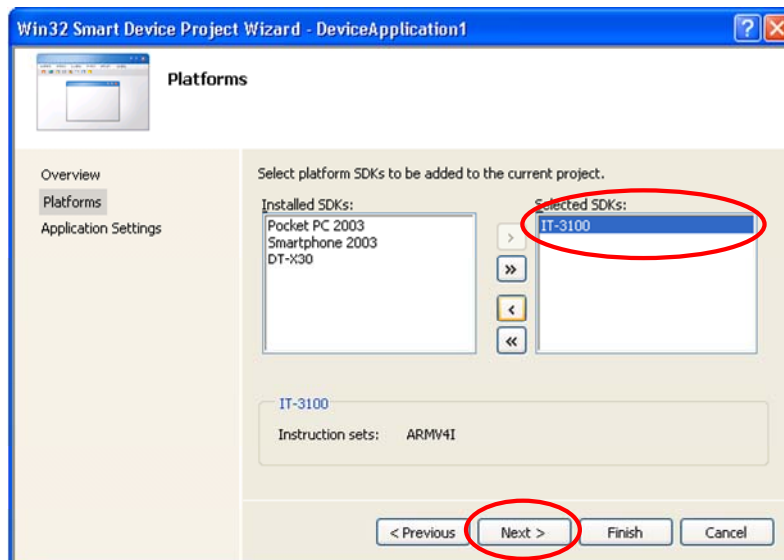


Figure 9.5

5. Choose **Console Application** for the Application type in the Application Settings menu, and click **Finish** button.
6. In Solution Explorer, click **DeviceApp.cpp** and then append the source code below subsequent to “#include <commctrl.h>”.

```
#include <SystemLib.h>
#if !defined(_countof)
#define _countof(_Array) (sizeof(_Array) / sizeof(_Array[0]))
#endif
```

7. Append the source code below in the main function of **DeviceApp.cpp**.

```
DWORD result;
TCHAR msg[16];

result = SysSetLED(LED_RED, 5, 8, 8);

if(result == TRUE)
{
    result = SysGetLED();
    switch(result & 0x0000000F) {
    case LED_OFF:
        wcscpy_s( msg, _countof(msg), TEXT("LED_OFF") );
        break;
    case LED_RED:
        wcscpy_s( msg, _countof(msg), TEXT("LED_RED") );
        break;
    case LED_GREEN:
        wcscpy_s( msg, _countof(msg), TEXT("LED_GREEN") );
        break;
    case LED_ORANGE:
        wcscpy_s( msg, _countof(msg), TEXT("LED_ORANGE") );
        break;
    default:
        wcscpy_s( msg, _countof(msg), TEXT("LED_UNKNOWN") );
        break;
    }
    MessageBox(NULL, msg, TEXT("LED"), MB_OK);
}
```

8. Choose **Properties** in the Project of Visual Studio 2008 or Visual Studio 2005, and navigate to **Configuration Properties** → **Linker** → **Input** → **Additional Dependencies** and then append **SystemLib.lib** (see red circle).

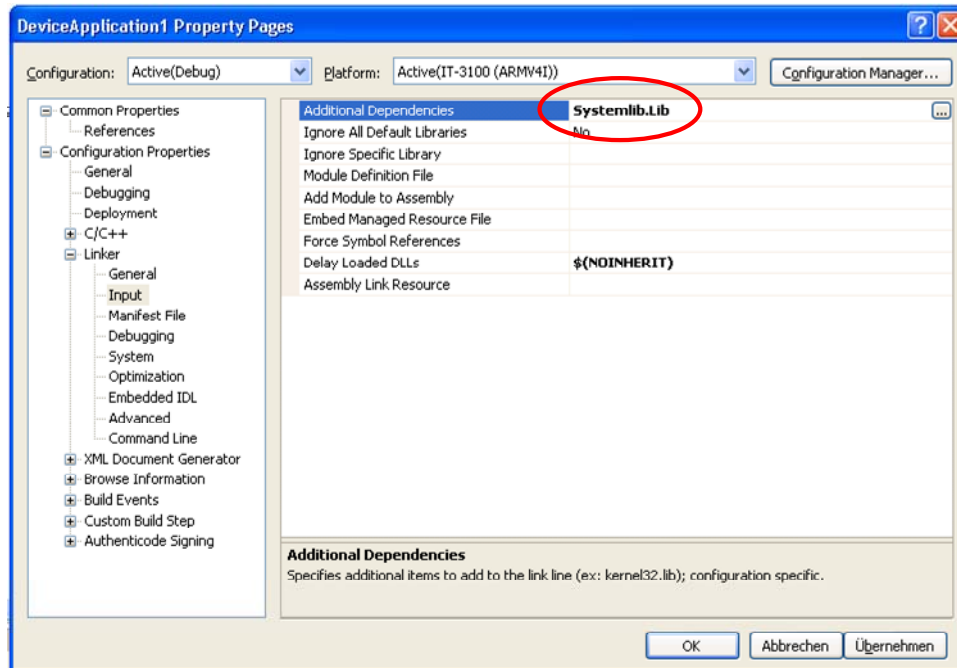


Figure 9.6

9. Choose **IT-3100 Device** for the Target Device to establish connection with the PC.
10. Choose **Debug** in Visual Studio 2008 or Visual Studio 2005 and then click either **Start Debugging** or **Start Without Debugging**.
11. The project will be built and copied to the “\Program Files\<name of project>” folder (by default) in the IT-3100.
12. Check that the program runs correctly on the IT-3100.

You can also use the CASIO samples as described in Chapter 1.4.

10. Resources

Microsoft's own <http://msdn.microsoft.com/mobility/> is an extremely comprehensive resource for programmers targeting WindowsCE .NET based devices. It includes links to most other useful web based resources. You will find detailed Software, Library and .NET Library manuals on <http://world.casio.com/system/pa>.