

CASIO
DT-X30 Series
Windows Mobile OS
Quick Start Guide

(Version 1.04)

CASIO Computer Co., Ltd.

Copyright ©2010. All rights reserved.

October 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 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 BDK to PC	15
3.1	Application Development	15
3.2	Installing CASIO BDK Files	16
Chapter 4.	Connecting Power Supply to Cradle	19
4.1	HA-G60IO	19
4.2	HA-G62IO	20
Chapter 5.	Connecting the DT-X30 to PC	21
5.1	Connection via USB Cradle	22
5.1.1	ActiveSync via USB	23
5.1.2	Windows Mobile Device Center via USB	23
5.2	Connection via IrDA	24
5.3	Connection via WLAN	25
5.4	Connection via Ethernet Cradle	30
5.5	Accessing Shared Network Drive on Your LAN	31
5.6	Direct TCP/IP Connection from Visual Studio	32
Chapter 6.	Setting Up the Development Environment	34
6.1	Installing CAB Files	34
6.2	Visual Studio 2005	35
Chapter 7.	Device Emulator	36
7.1	Software Required	36
7.2	Starting Up the Device Emulator	38
7.3	Using the Device Emulator	39
7.3.1	DT-X30WM Device Emulator	39
7.3.2	I/O Simulator	40
7.3.3	Connecting via ActiveSync	45
7.4	Debugging Applications	47
Chapter 8.	Visual Studio	49
8.1	Developing in C++	50
8.2	Developing in VB	54
8.3	Developing in C#	56
Chapter 9.	Resources	59

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 DT-X30 series handheld terminals with the integrated Windows Mobile® Version 6.1 OS. The best methods of connecting to your development system are covered and step by step instructions for installing and testing the CASIO Basic Development Kit (BDK) 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.

For DT-X30 series handheld terminals with the integrated Windows® CE OS, see DT-X30 Quick Start Guide available separately.

The words, “the terminal”, “the DT-X30”, “the DT-X30 series”, and “DT-X30WM device”, used in this reference manual mean DT-X30 series handheld terminals with the integrated Windows Mobile® as its OS unless otherwise noted.

1. Product Overview

1.1 Features at a Glance

The DT-X30 has been designed using the new concept of the Human-centered Design Processes and is capable of performing a wide variety of powerful functions.

The following is a brief overview of the features available on the DT-X30 series handheld terminals. For further detail on the hardware specifications, refer to DT-X30 Series Hardware Manual.

Outstanding development environment

- Microsoft® Windows Mobile® 6.1 English Version as the built-in OS
- Visual Studio 2008
- Visual Studio 2005

Capability of various communication systems

- Built-in ultra-small WLAN module compatible with the IEEE802.11b/g standard
- Built-in WWAN module combines GSM, GPRS and EDGE modules with GPS receiver designed for use in transportation and logistics.
- High speed infrared communication with IrDA Version. 1.3
- Bluetooth® Version 2.0
- Serial interface with USB version 1.1 (Host/Client)

Small size, light weight

- Dimensions : Approx. 87 (W) x 242 (D) x 46 (H) mm
- Weight : Approx. 550 g

Improved durability

- Impact resistance : 1.8 m in height *
- Dust/Water-splash proof : IP67 level (compliant with IEC60529 International Standard)

Capable of scanning industrial standard bar code symbologies

- Readable bar code symbologies:
UPC-A/E, EAN8, EAN13, Codabar, Code39, Code93, Code128/GS1-128, ITF, MSI, IATA, GS1 DataBar Omnidirectional, GS1 DataBar Limited, GS1 DataBar Expanded, GS1 DataBar Stacked, GS1 DataBar Expanded Stacked

CPU, Memory

- High-performance CPU
Marvell® PXA320 Application Processor (runs at maximum 624 MHz)
- Large-capacity memory
RAM : 128 MB
F-ROM : 128 MB (user area; approximately 60 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 DT-X30WM Basic Development Kit (“BDK”) provides various libraries including those listed in the table.

Table 1.1

Library	Description	C++	VB C#
System Library	Library that is used to control the system.	Yes	Yes
Laser Scanner Library	Library that is used to control the built-in laser scan engine.	Yes	Yes
Bluetooth Library	Library that is used to control the built-in Bluetooth module.	Yes	Yes
Camera Library (See note 2)	Library that is used to control the built-in Camera module.	Yes	Yes
Imager Library	Library that is used to control the built-in CMOS Imager.	Yes	Yes
WANGPRS Library	Library that is used to control the built-in WANGPRS module.	Yes	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

Note:

The abbreviations used in the table are;

C++ : Visual C++
VB : Visual Basic .NET
C# : Visual C# .NET

Note 2:

When you install patch file, DT-X30WM model is available to use CASIO Camera library.

Table 1.2 shows each file name of the Dynamic Link Library and Dynamic Link Class Library.

Table 1.2

Library	Dynamic Link Library	Dynamic Link Library (Class Library)
System Library	SystemLib.dll	SystemLibNet.dll
Laser Scanner Library	OBReadLib.dll	OBReadLibNet.dll
Bluetooth Library	BluetoothLib.dll	BluetoothLibNet.dll
Camera Library	CameraLib.dll	CameraLibNet.dll
Imager Library	ImagerLib.dll	ImagerLibNet.dll
WANGPRS Library	WangprsLib.dll	WangprsLibNet.dll
JPEG Library	JpegCe.dll	None
FLINK Library	FlinkLib.dll	MoFlinkLib.dll

1.3 Development Manuals

The DT-X30WM Basic Development Kit (“BDK”) provides various development reference manuals as described in the table below.

Table 1.3

Development Manual	Description
Quick Start Guide (See note 1)	This reference manual.
Software Manual (See note 1)	Reference manual that describes software specifications in detail for all the software integrated in DT-X30 handheld terminal.
Hardware Manual (See note 2)	Reference manual that describes hardware specifications in detail on each dedicated option and DT-X30 handheld terminal
System Library Manual	Reference manual that describes individual functions in detail for System Library.
Laser Scanner Library Manual	Reference manual that describes individual functions in detail for Laser Scanner Library.
Bluetooth Library Manual	Reference manual that describes individual functions in detail for Bluetooth Library.
Camera Library Manual	Reference manual that describes individual functions in detail for Camera Library.
Imager Library Manual	Reference manual that describes individual functions in detail for Imager Library.
WANGPRS Library Manual	Reference manual that describes individual functions in detail for WANGPRS 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.

Notes:

1. The dedicated reference manual for models with Windows Mobile® OS.
2. The hardware manual for models with either Windows Mobile® OS or Windows CE OS.

1.4 Sample Program

The following sample programs are included in the DT-X30WM Basic Development Kit.

Table 1.4

Sample Program	Description
WANGPRS Library	
GPRS_TUTORIAL	Demonstrates GPRS communication.

Note:

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

2. Prerequisites

2.1 Skills Required

The following skills are required by developers aiming to develop application software for the DT-X30 terminal.

- 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 DT-X30 series and dedicated options are available.

Table 2.1 List of available models

Model no.	OS (Note 1)	Scanner	WLAN (802.11 b/g)	WWAN (GSM, GPRS, EGPRS/EDGE)	GPS	Bluetooth	Camera
DT-X30R-15	Mobile	Laser	Yes	No	No	Yes	No
DT-X30GR-15	Mobile	Laser	Yes	Yes	Yes	Yes	No
DT-X30GR-15C	Mobile	Laser	Yes	Yes	Yes	Yes	Yes
DT-X30R-35	Mobile	Imager	Yes	No	No	Yes	No
DT-X30GR-35	Mobile	Imager	Yes	Yes	Yes	Yes	No
DT-X30GR-35C	Mobile	Imager	Yes	Yes	Yes	Yes	Yes
DT-X30G-35U	Mobile	Imager	No	Yes	Yes	Yes	No
DT-X30G-35UC	Mobile	Imager	No	Yes	Yes	Yes	Yes

Notes:

1. "Mobile" denotes that the model integrates Windows Mobile® 6.1 English Version for its operating system.
2. The table shows all the models with Windows Mobile® OS available current as of May 2009.

Table 2.2 List of the dedicated options

Option	Product	Model no.	Remark
Cradle	USB Cradle	HA-G60IO	
	Ethernet Cradle	HA-G62IO	
Battery	Battery Pack	HA-G20BAT	
Battery charger	Dual Battery Charger	HA-G32DCHG	
	Cradle-type Battery Charger	HA-G30CHG	
	Car Mounted-type Battery Charger	HA-G35CHG	
AC adaptor		AD-S42120B	For HA-G60IO, HA-G62IO, HA-G30CHG, HA-G32DCHG
Power Cable for AD-S42120B		AC-CORD-EU	For Europe
		AC-CORD-US	For USA/Canada
		AC-CORD-TW	For Taiwan
		AC-CORD-KR	For Korea
		AC-CORD-AU	For Australia/New Zealand
Cable	USB cable	DT-380USB	For cradle - PC
Others	Screen Protect Sheet	HA-G90PS5	

External views of the DT-X30 and the dedicated options

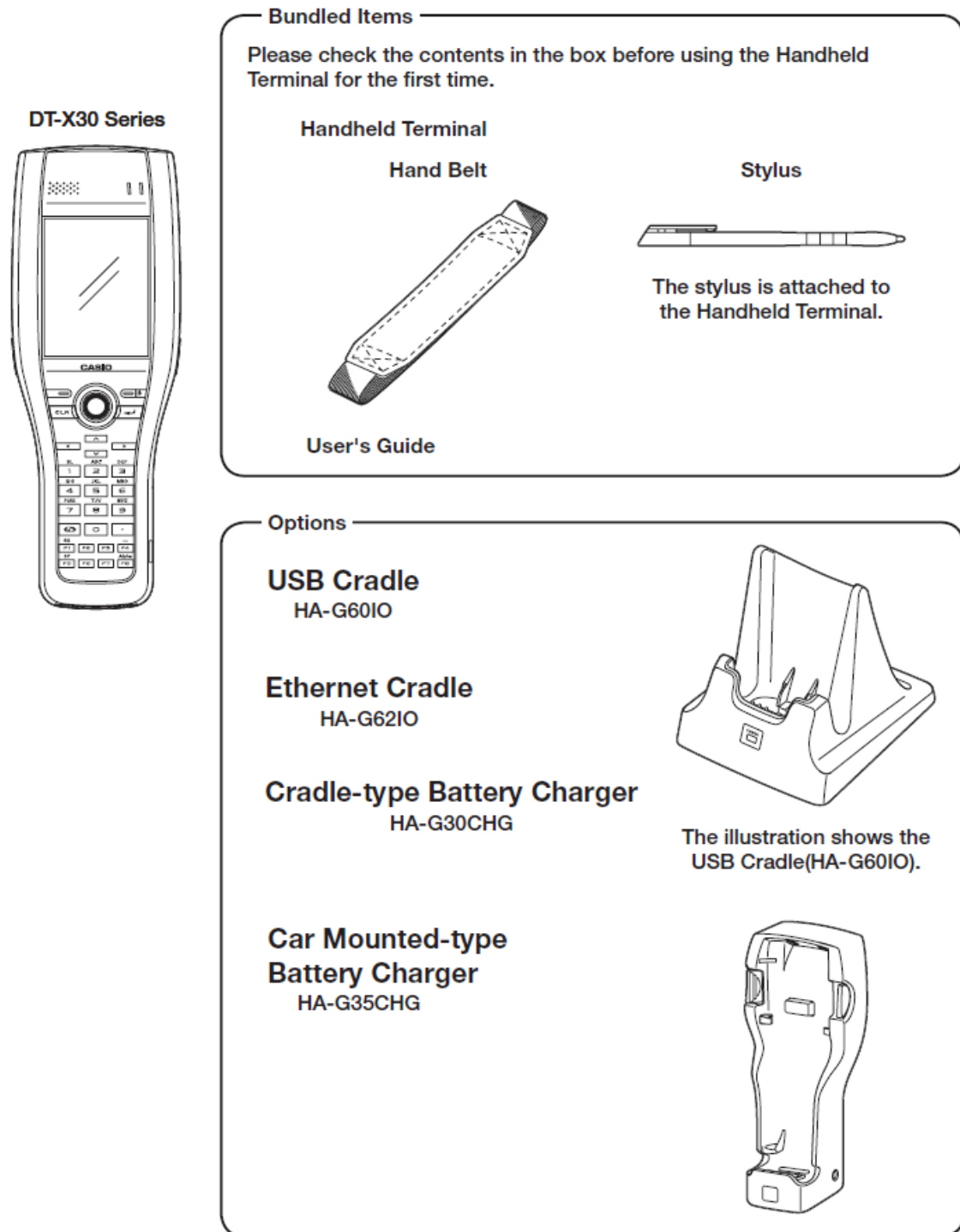


Figure 2.1

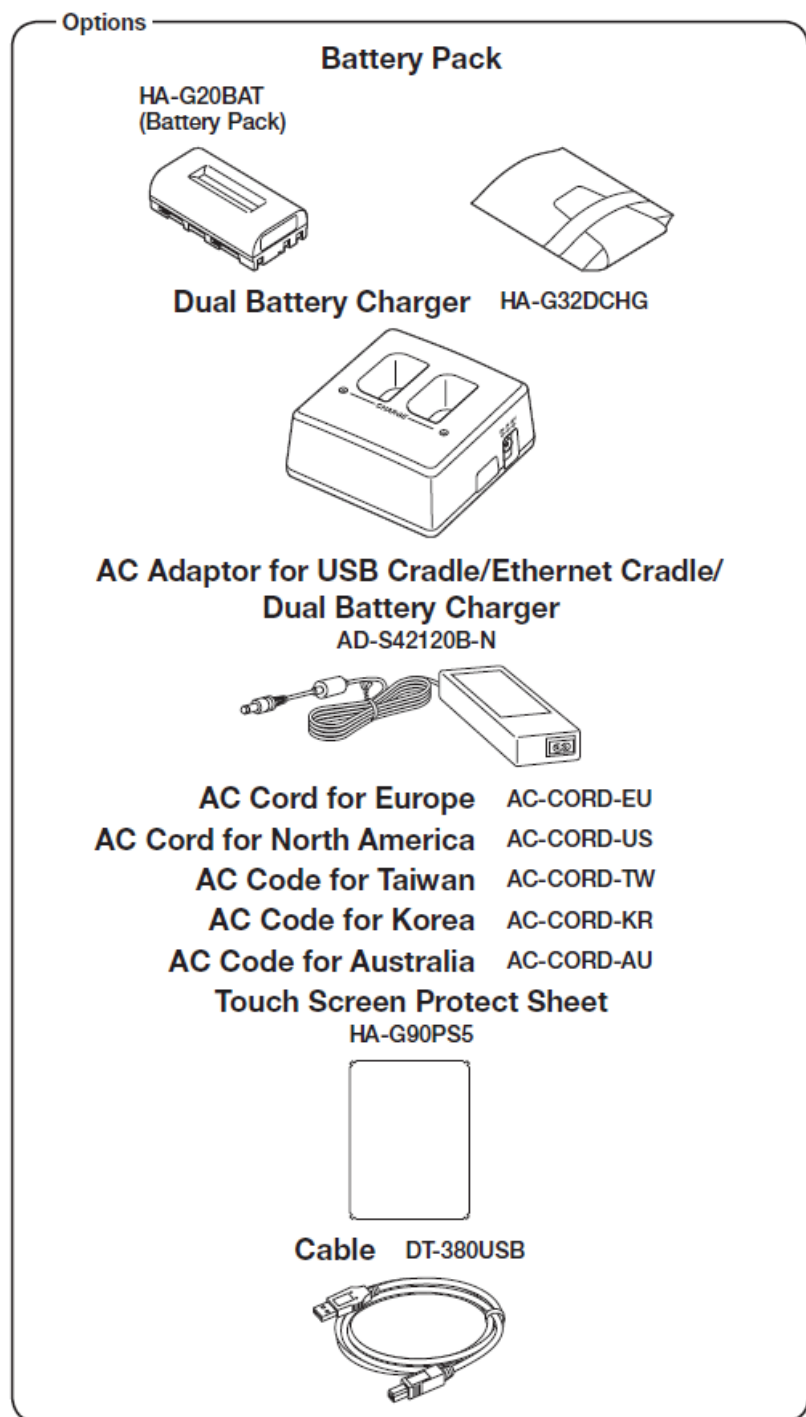


Figure 2.2

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

Development platform

The following software tools and libraries are required in order to develop software for the DT-X30. Please ensure that you download or purchase the correct Microsoft tools as appropriate.

- Microsoft Visual Studio 2008 + Service Pack 1 (or later) (not free of charge)
- Microsoft Visual Studio 2005 + Service Pack 1 (or later) (not free of charge)
- Microsoft ActiveSync 4.5 (or later)
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?familyid=9E641C34-6F7F-404D-A04B-DC09F8141141&displaylang=en>
- Microsoft Windows Mobile Device Center 6.1 (for Windows Vista or later)
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?familyid=46F72DF1-E46A-4A5F-A791-09F07AAA1914&displaylang=en>
- Microsoft Windows Mobile 6 Professional SDK
Download for free from;
<http://www.microsoft.com/downloads/details.aspx?familyid=06111A3A-A651-4745-88EF-3D48091A390B&displaylang=en>

CASIO DT-X30WM BDK

Download the DT-X30WM BDK 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 BDK 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 2008 and Visual Studio 2005. For detail, refer to Chapter 2.3 "Software Required".

2. Installing CASIO BDK to PC

Install Casio's BDK ("Basic Development Kit") and various libraries if necessary. For installation method, refer to Chapter 3.2 "Installing CASIO BDK Files".

3. Connecting DT-X30 to PC (via either ActiveSync or Windows Mobile Device Center)

Connect the DT-X30 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 DT-X30 to PC".

4. Setting up the Development Environment

Transfer the Casio's libraries to the DT-X30.

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

5. Installing the Device Emulator

Install the Device Emulator for the DT-X30. 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, transmit it to the Device Emulator or an actual terminal of the DT-X30 via ActiveSync or Windows Mobile Device Center for check on the operability. For application development method and transmitting your application, refer to Chapter 8 "Visual Studio".

3.2 Installing CASIO BDK Files

Download the CASIO DT-X30WM BDK installation CD image file from the following site and write it to a CD-ROM media.

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 DT-X30 BDK Ver. 1.xx, be sure to uninstall it before installing Ver. 2.xx.
- If any file of the DT-X30 BDK 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 BDK Files.
 - 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.

Installation steps

1. When the CASIO DT-X30WM BDK CD-ROM is inserted in the drive of PC, the following menu screen appears automatically.

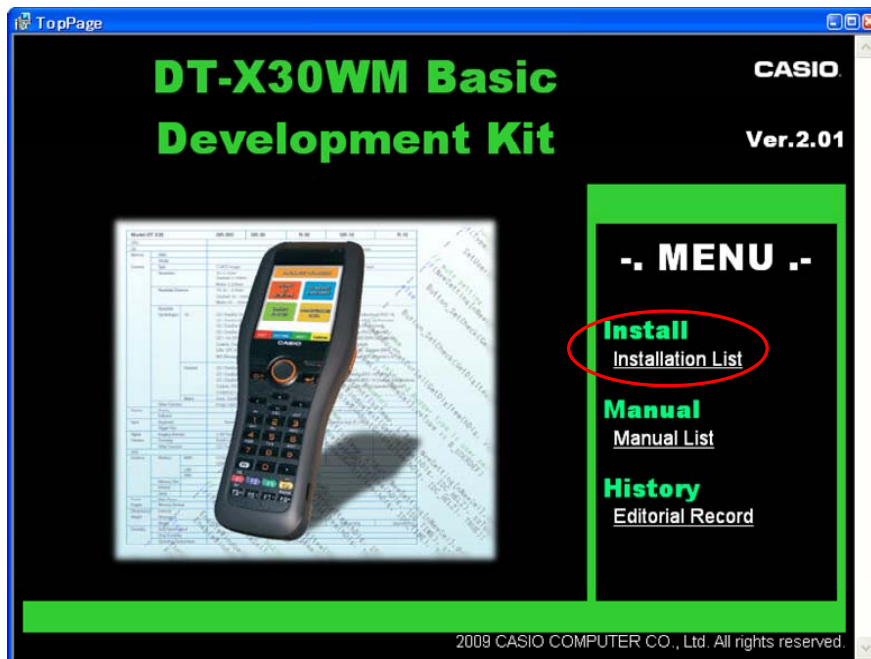


Figure 3.1

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



Figure 3.2

3. Choose the components in the screen.

Installing Library

The DT-X30 Library is available for the below development platforms.

- Microsoft Visual Studio 2008
- Microsoft Visual Studio 2005

Notes:

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

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**.

- Microsoft Visual Studio 2008
- Microsoft Visual Studio 2005

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 on the screen. The Sample program folder is available for the development platforms below. Copy it into your PC.

- VS2008 for Microsoft Visual Studio 2008
- VS2005 for Microsoft Visual Studio 2005

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-G60IO

Use the dedicated AC adaptor (AD-S42120B) for supplying power to the HA-G60IO USB Cradle. Ensure that you connect the AC adaptor to the cradle before starting communication between the DT-X30 and PC via the cradle. Follow the steps below to connect the power supply to the DT-X30 using the dedicated AC adaptor.

1. Plug the AC adaptor into the AC adaptor jack where “DCIN12V” is printed on the back of the cradle.
2. After connecting the power cable to the AC adaptor, plug in the plug to an electrical outlet.
3. Make sure the selector switch on the back of the cradle is set to position B.
4. Connect a USB cable (DT-380USB) to the USB client port on the back of the cradle, and then connect the other end of the cable to the PC. USB host port is used when connecting the cradle with other USB peripheral devices.
5. Align the USB cradle mount holes on the back of the DT-X30 with the mount hooks on the cradle after aligning the contacts on the bottom of the DT-X30 with the power contacts of the cradle. The power LED on the front of the cradle will light green if the DT-X30 has been properly mounted.

Status of Indicator 1 on DT-X30

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 DT-X30 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 DT-X30 and cradle to vibration or impact during communication. This results in communication being interrupted.
- When mounting the DT-X30, 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.

4.2 HA-G62IO

Use the dedicated AC adaptor (AD-S42120B) for supplying power to the HA-G62IO Ethernet Cradle. Ensure that you connect the AC adaptor to the cradle before starting communication between the DT-X30 and PC via the cradle. Follow the steps below to connect the power supply to the DT-X30 using the dedicated AC adaptor.

1. Plug the AC adaptor into the AC adaptor jack on the back of the Ethernet Cradle.
2. After connecting the AC adaptor to the power cable, plug in the plug of the power cable to an electrical outlet.
3. Set the selector switch on the back of the Ethernet cradle to the port that will be used. Set the switch to “LAN” to use the LAN port or to “USB” to use the USB port.
4. Before using the cradle ports, remove the caps from the ports. When using a LAN, connect one end of the LAN cable to the LAN port and the other end to the PC or hub. When using a USB connection, connect one end of the USB cable (DT-380USB) to the USB port and the other end to the PC.
5. Align the contacts on the underside of the DT-X30 with the power supply contacts on the Ethernet Cradle and then set the DT-X30 into the cradle so that mount holes in the back of the DT-X30 are aligned with the mount hooks on the cradle. Once the DT-X30 is properly set in the cradle, the power LED on the front of the Ethernet cradle lights green.

Status of Indicator 1 on DT-X30

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

Notes:

- Always remove the DT-X30 from the Ethernet Cradle before changing the selector switch setting.
- Never short out the power contacts of the cradle. This damages the cradle.
- Do not subject the DT-X30 and cradle to vibration or impact during communication. This results in communication being interrupted.
- When mounting the DT-X30, 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 battery pack or communication will not proceed if it is not mounted properly.
- The LAN and USB connections cannot be used concurrently.
- Always cap ports that are not being used. Using the Ethernet Cradle while the ports are uncapped can cause damage.

5. Connecting the DT-X30 to PC

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

- ActiveSync 4.5 or a later version (for Windows XP or any other OS earlier)
Use the ActiveSync to connect the DT-X30 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=9E641C34-6F7F-404D-A04B-DC09F8141141&displaylang=en>
- Windows Mobile Device Center 6.1 or a later version (for Windows Vista or later)
Use the Windows Mobile Device Center to connect the DT-X30 to PC if the PC runs in Windows Vista OS or later. 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>

5.1 Connection via USB Cradle

Before making connection using ActiveSync or Windows Mobile Device Center, check the USB Connection settings on the terminal.

Follow the steps below:

1. Navigate to **Start** → **Settings** → **System** tab → **USB Connection**.
2. Tap **Connect Utility** tab.

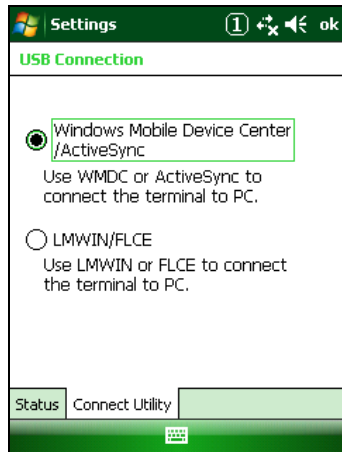


Figure 5.1

3. Confirm that “Windows Mobile Device Center/ActiveSync” is selected. If not, select the radio button and tap **ok**.

5.1.1 ActiveSync via USB

To establish connection via USB interface with PC runs in Windows XP or any other OS earlier, use ActiveSync.

Follow the steps below:

1. Install ActiveSync to the PC, if not installed.
2. Connect the USB cable to the PC and the other end to the USB Cradle and also the AC adaptor to the USB Cradle.
3. Make sure the selector switch on the back of the cradle is set to position B.
4. Put the DT-X30 on the cradle and confirm that the green LED on the front of the cradle is lit. If not, be sure the DT-X30 is positioned firmly on the cradle.
5. Follow a message appeared in the ActiveSync on the PC.

5.1.2 Windows Mobile Device Center via USB

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

Follow the steps below:

1. Install WMDC to the PC, if not installed.
2. Connect the USB cable to the PC and the other end to the USB Cradle and also the AC adaptor to the USB Cradle.
3. Make sure the selector switch on the back of the cradle is set to position B.
4. Put the DT-X30 on the cradle and confirm that the green LED on the front of the cradle is lit. If not, be sure the DT-X30 is positioned firmly on the cradle.
5. Follow a message appeared in the WMDC on the PC.

5.2 Connection via IrDA

If the PC has an IrDA interface, it is possible to connect the DT-X30 to the PC via IrDA using ActiveSync.

Follow the steps below:

1. Set the COM port used by ActiveSync on the PC to **Infrared Port(IR)**.
2. Place the DT-X30 so that the IrDA port located on the right side of the DT-X30 faces with the IrDA port on the PC. Communication can be established if the distance is between zero and approximately 1 m. In the case of 4Mbps rate, the maximum distance is approximately 30 cm.
3. On the DT-X30, from **Start** menu, navigate to **Programs** → **ActiveSync** to start up ActiveSync.
4. Tap **Menu** → **Connect via IR**.

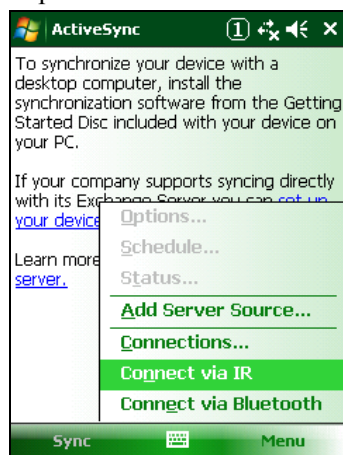


Figure 5.2

5.3 Connection via WLAN

To establish communication between the DT-X30 with WLAN module integrated (see Table 2.1 for applicable models) and PC via WLAN configuration, follow the steps, 1 to 6, below to set up a WLAN configuration on the DT-X30. After setting up the configuration, be sure to perform a site survey prior to starting communication via WLAN.

1. Navigate to **Start** → **Settings** → **Connection** tab → **WLANConfig** and then tap **IP** tab.

Figure 5.3

Table 5.1

Parameter	Description
Enable DHCP or Configure IP	Determines “Enable” or “Disable” for DHCP.
IP	Determines IP address.
MASK	Determines subnet mask.
GateWay	Determines default gateway.
DNS1	Determines primary DNS address.
DNS2	Determines secondary DNS address.
WINS1	Determines primary WINS address.
WINS2	Determines secondary WINS address.

If any of the settings in Figure 5.3 is omitted, the process described in the following table will automatically take place in the field.

Table 5.2

Parameter	Nothing is set (DHCP)	“Configure IP” is set
Enable DHCP or Configure IP	“Enable DHCP” is assumed.	“Configure IP” is set.
IP	Does not determine IP address.	Entered address is set as is.
MASK	Does not determine subnet mask.	Entered address is set as is.
GateWay	Does not determine gateway.	Entered address is set as is.
DNS1	Does not determine primary DNS address.	Entered address is set as is.
DNS2	Does not determine secondary DNS address.	Entered address is set as is.
WINS1	Does not determine primary WINS address.	Entered address is set as is.
WINS2	Does not determine secondary WINS address.	Entered address is set as is.

2. Tap **Basic** tab. Set up each parameter in the tab by referring to the descriptions for the parameters in Table 5.3.

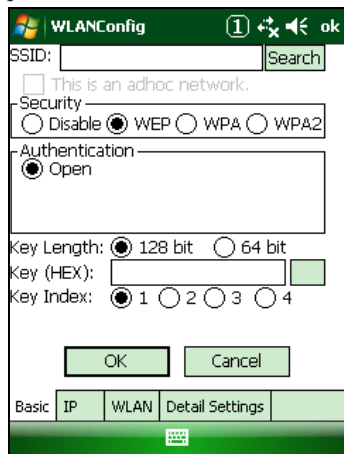


Figure 5.4

Table 5.3

Parameter		Description
SSID		Enter the SSID of the network you want to connect to.
Security	Disable	None.
	WEP	Open in Authentication field.
	WPA WPA2	PSK in Authentication field (if selected, the Key field must be set also.) EAP-PEAP in Authentication field EAP-TLS in Authentication field
Key		Enter 26 (maximum) alphanumeric digits (13 hex pairs) in the Key field if 128 bit radio button is selected. Or, enter 10 (maximum) alphanumeric digits (5 hex pairs) in the Key field if 64 bit radio button is selected. The field displays the number of characters that have been entered. ***** in the field implies that the key has been extracted from the ini file. If ***** in the field is edited, a new key has been deemed to be set. Or, if it has never been edited, the key extracted from the ini file becomes effective as is.

If **EAP-PEAP** radio button in Authentication field is selected, tap the **EAP-Properties** button that appears when selecting the **EAP-PEAP** radio button to set also the following parameters.

Table 5.4

Parameters in EAP-Properties screen	Description	Default
User name	Input a user name in alphanumeric (maximum 100 alphanumeric).	None
Password	Input a password in alphanumeric (maximum 100 alphanumeric). ***** in the field implies that the password has been extracted from the ini file. If ***** in the field is edited, a new password has been deemed to be set. Or, if it has never been edited, the password extracted from the ini file becomes effective as is.	None
Domain	Input a domain in alphanumeric (maximum 100 alphanumeric).	None
Validate server certificate	Set up the requisition for server certificate. With check mark : certificate is required. Without check mark: certificate is not required.	Certificate is not necessary

If **EAP-TLS** radio button in Authentication field is selected, tap the **EAP-Properties** button that appears when selecting the **EAP-TLS** radio button to set the following settings.

Table 5.5

Parameters in EAP-Properties screen	Description	Default
User name	Input a user name in alphanumeric (maximum 100 alphanumeric).	None
Certificate	Select a client certificate installed already (maximum 100 alphanumeric). Search button in the field will display a list of installed client certificates. Select one by highlighting it.	None
Domain	Input in alphanumeric (maximum 100 alphanumeric)	None
Validate server certificate	Set up the requisition for server certificate. With check mark : certificate is required. Without check mark : certificate is not required.	Certificate is required.

3. Tap **WLAN** tab.

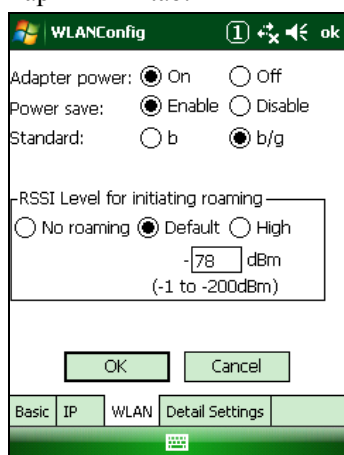


Figure 5.5

Table 5.6

Field / Radio Buttons	Description		Default
Adapter power	On	Enable power to the integrated WLAN module.	Yes
	Off	Disable power to the integrated WLAN module.	
Power save	Enable	Enable power save mode for the WLAN module.	Yes
	Disable	Disable power save mode for the WLAN module.	
Standard	b	Set up IEEE802.11b standard effect.	
	b/g	Set up IEEE802.11b/g standard effect.	Yes
RSSI Level for initiating roaming	No roaming	Set up “-100 dBm” for roaming starting threshold level, a level where communication via WLAN is practically impossible, so that roaming should not be carried out.	
	Default	Set up “-78 dBm” for the roaming starting threshold level.	Yes
	High	Set up “-72 dBm” for roaming starting threshold level, for faster (more frequent) roaming	

4. Tap **Detail Settings** tab.

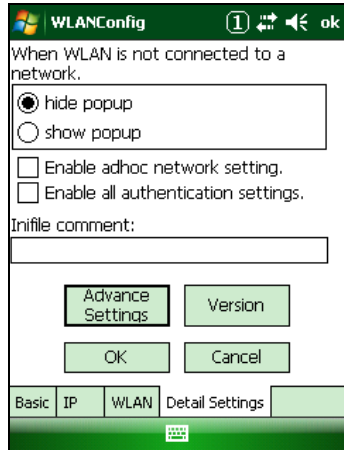


Figure 5.6

Table 5.7

Field / Radio Buttons		Description	Default
When WLAN is not connected to a network	hide popup	Hides popup window when WLAN is not connected to a network.	Yes
	show popup	Shows popup window when WLAN is not connected to a network.	
Enable AdHoc network setting		With check mark : enable the setting.	
		Without check mark : disable the setting.	Yes
Enable all authentication settings		With check mark : enable the setting.	
		Without check mark : disable the setting.	Yes
Infile comment		Enter a comment of up to 100 characters to be written in the ini file.	None

5. If **OK** button is clicked, the screen in Figure 5.7 appears. Tap **OK** button to perform a reset on the terminal so that the setting takes effect.

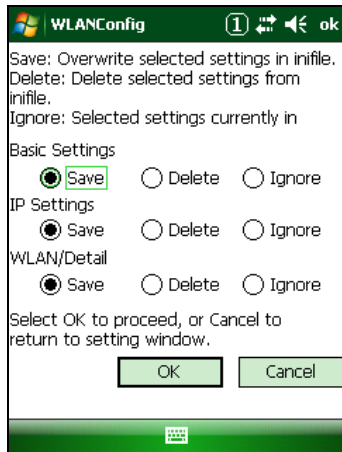


Figure 5.7

6. Check to make sure that the connection has been established using the **NetSearch** utility, and then navigating to the **Ping** function in there. Enter HostName first and then tap **Ping** to check that you are connected to the network correctly.

5.4 Connection via Ethernet Cradle

This chapter describes how to establish a high speed LAN connection on the WLAN non-integrated models with HA-G62IO Ethernet cradle.

Follow the steps below:

1. Connect the dedicated AC adapter to the Ethernet cradle as described in Chapter 4.2.
2. Connect one end of the network cable to the Ethernet cradle and the other end to the network hub.
3. Make sure the selector switch on the back of the HA-G62IO Ethernet cradle is set to the position “LAN”.
4. Place the DT-X30 in the cradle and navigate to **Start** → **Settings** → **Connections** tab → **Network Cards**.
5. The following screen appears. Double tap “AX88772”.

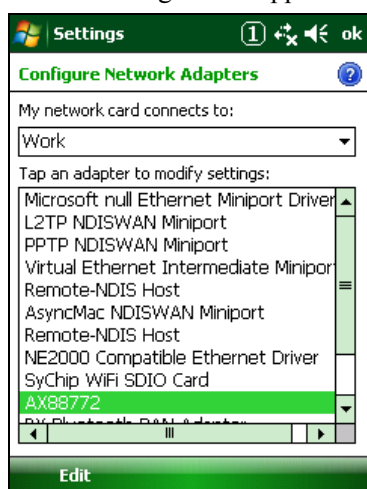


Figure 5.8

6. The following TCP/IP screen appears. Set up all the parameters in **IP Address** and **Name Servers** tabs as required and tap **ok** button.

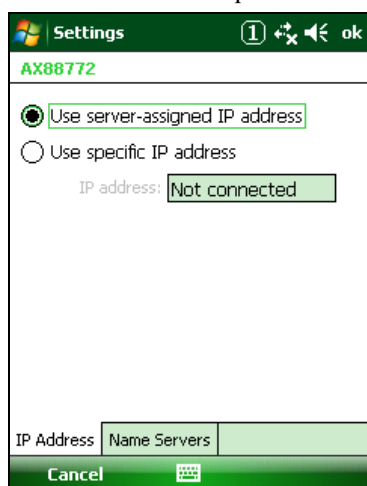


Figure 5.9

7. If the connection is established correctly, the icon (🔌) in the Taskbar changes to (🌐).

5.5 Accessing Shared Network Drive on Your LAN

Assuming you have a valid network connection established, you can access shared drives on your PC from the **File Explorer** on the DT-X30. The following shows the steps to initiate this.

1. Configure a network connection on the DT-X30.
2. Double tap My Device.
3. Type \\xxxx\ where xxxx is the network name of the PC.
4. A network logon dialog box will appear. Enter a valid User ID, Password and Network Domain.
5. Any shared network drives on the target PC will be displayed and you will be able to copy files freely between them and the DT-X30.

5.6 Direct TCP/IP Connection from Visual Studio

If you have a network connection to the DT-X30 (for example, via WLAN or the Ethernet cradle) then you can establish a direct link to the development PC without using ActiveSync.

For Visual Studio 2005

1. Download the files listed below to the DT-X30.

- Clientsshutdown.exe
- ConmanClient2.exe
- CMAccept.exe
- DeviceDMA.dll
- eDbgTL.dll
- TcpConnectionA.dll

The source folder in the PC:

C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i

The destination folder in the DT-X30:

\Windows

2. Run **ConmanClient2.exe** on the DT-X30.
3. Set the device IP address in Visual Studio 2005.
4. Navigate to **Tools** in the main menu of Visual Studio 2005 → **Options...** → **Device Tools** → **Devices**.
5. Choose **Windows Mobile 6 Professional Device** in the pull-down menu of **Devices:** and click **Properties...**
6. Click **Transport:** to access **Configure...** and set up **Device IP address** as shown in Figure 5.10.

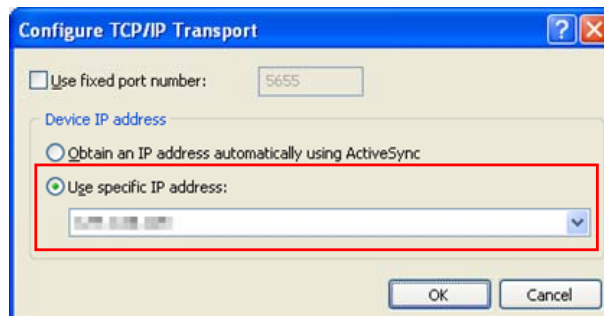


Figure 5.10

7. Run **CMAccept.exe** on the DT-X30.
8. Navigate to **Tools** in the main menu of Visual Studio 2005 → **Connect to Device...**

9. Choose **Windows Mobile 6 Professional Device** in the list of **Devices:** and click **Connect** button. The screen in Figure 5.11 if appear indicates the success of connection establishment.

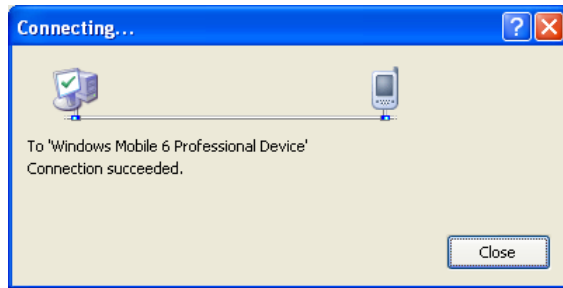


Figure 5.11

6. Setting Up the Development Environment

6.1 Installing CAB Files

1. After installing the library files, the CAB files in Table 6.1 will be installed in the folder below.

C:\Program Files\CASIO\MBSYS\CAB

Table 6.1

Library	CAB file	Preinstalled
System Library	en_SystemLibWM.ARMV4I.CAB	Yes
Laser Scanner Library	en_OBReadLibWM.ARMV4I.CAB	Yes
Bluetooth Library	en_BluetoothLibWM.ARMV4I.CAB	Yes
Imager Library	en_ImagerLibWM.ARMV4I.CAB	Yes
WANGPRS Library	en_WANGPRSLibWM.ARMV4I.CAB	No
JPEG library	enJPEGWM.ARMV4I.CAB	Yes
FLINK library	en_FlinkWM.ARMV4I.CAB	Yes (note 2)

Notes:

1. The library with "Yes" in "Preinstalled" column is preinstalled in the DT-X30 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_FlinkWM.ARMV4I.CAB, in the table does not operate for the DT-X30. Use the FlinkLib.dll installed by default in the terminal.
2. Copy all the CAB files in Table 6.1 to any folder on the DT-X30 via ActiveSync.
 3. Carry out each CAB file.
 4. When the installation starts, the installation status will appear.

6.2 Visual Studio 2005

Follow the steps in Chapter 5 “Connecting the DT-X30 to PC” before checking the steps below to confirm that you can connect to the DT-X30 from Visual Studio 2005.

1. Establish connection with the DT-X30 via ActiveSync.
2. Open the application project for VB or C# in Visual Studio 2005.
3. Choose **Windows Mobile 6 Professional Device** in the pull-down menu box.

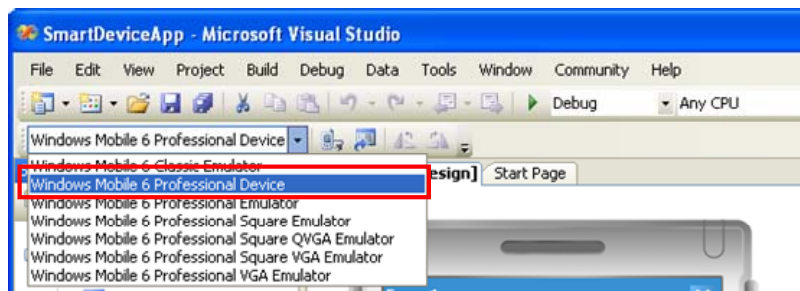


Figure 6.1

4. Click the button shown in the red box below (see Figure 6.2) to make sure that Visual Studio 2005 has recognized the connection established with the DT-X30 via ActiveSync. If it does not, start up ActiveSync again to establish connection.

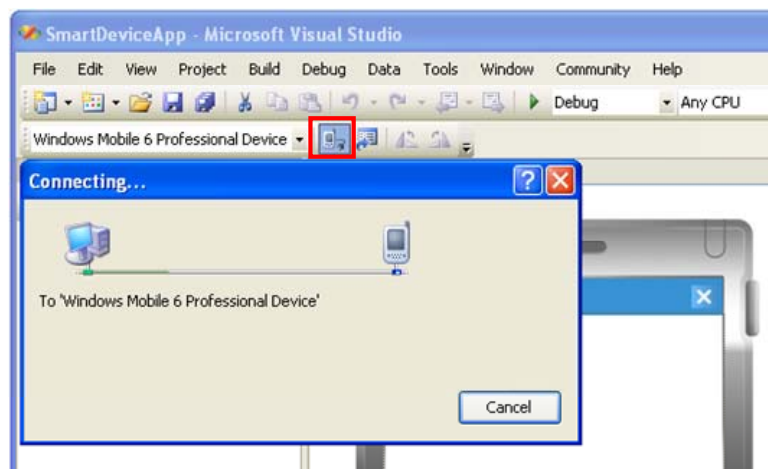


Figure 6.2

5. You will now be able to deploy solutions and also debug applications on the attached DT-X30 using the Visual Studio 2005 debugging features.

7. Device Emulator

The **Device Emulator** provides application developers with an environment that, without having the actual terminal 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.

- ActiveSync 4.5 or a later version (If required)
<http://www.microsoft.com/downloads/details.aspx?familyid=9E641C34-6F7F-404D-A04B-DC09F8141141&displaylang=en>
- Visual Studio 2008 or Visual Studio 2005
- Standalone Device Emulator 3.0
<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=a6f6adaf-12e3-4b2f-a394-356e2c2fb114>
- Windows Mobile 6 Professional SDK Refresh
<http://www.microsoft.com/downloads/details.aspx?familyid=06111A3A-A651-4745-88EF-3D48091A390B&displaylang=en>
- Windows Mobile 6 Emulator Image for CASIO DT-X30
- IO Simulator

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.)

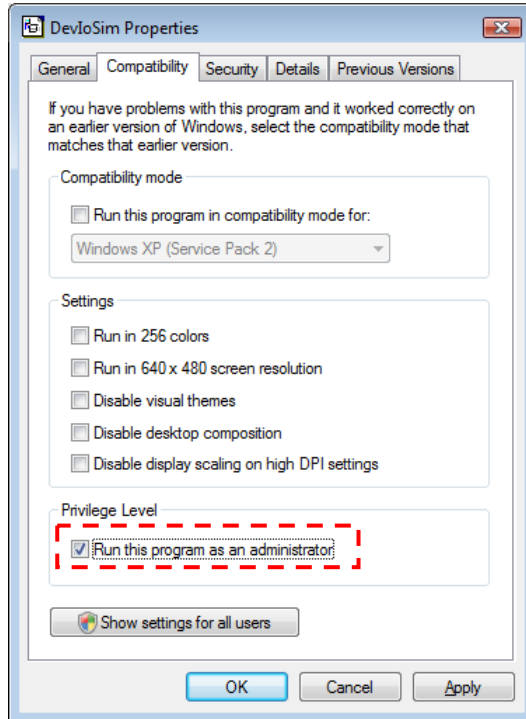


Figure 7.1

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 **DT-X30WM**.
2. Make sure that the DT-X30WM **Device Emulator** has started up on the screen. See Figure 7.2.
3. Navigate to Start menu → **All Programs** → **CASIO Device IO Simulator** and click **IO Simulator**.
4. Make sure that the **IO Simulator** has started up on the screen. See Figure 7.3.
5. If both Figure 7.2 and Figure 7.3 appear on your PC, you are ready to use the emulator.



Figure 7.2 DT-X30WM Emulator

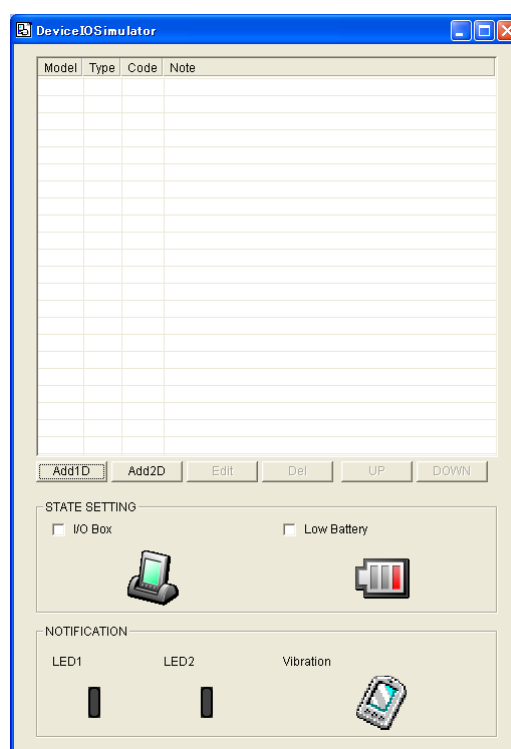


Figure 7.3 I/O Simulator

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 DT-X30WM Device Emulator

The **DT-X30WM Device Emulator** emulates various operations carried out by the actual DT-X30WM 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 DT-X30WM device on the screen of a PC.



Figure 7.4

Key Input

The emulator offers key input capability similar to that of the actual DT-X30WM device. For instance, a key on the emulated keyboard of DT-X30WM 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 DT-X30WM 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 the 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.

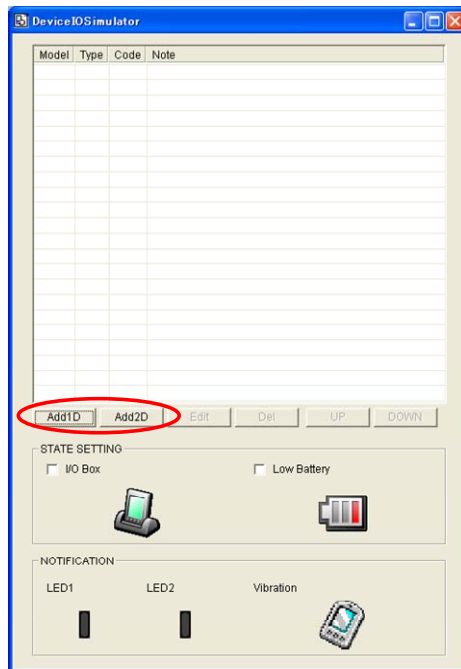


Figure 7.5

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**.

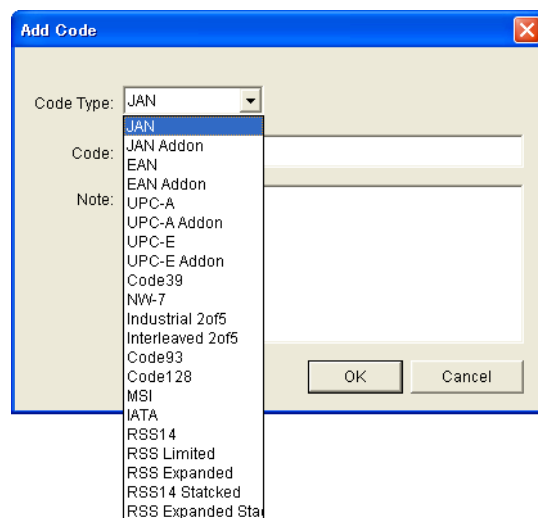


Figure 7.6

- Registration of bar code and note
Enter bar code data in the **Code** 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.

- Registration of bar code and note
Enter bar code data in the **Code** 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.

Add Code

Code Type: JAN

Code: 49143157

Note: JAN-8 CODE

OK Cancel

Figure 7.7

4. **Completion of registration**
After completion of the bar codes registration, the screen in Figure 7.8 shows a list of 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.

4. Completion of registration
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.

[illegible]

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.

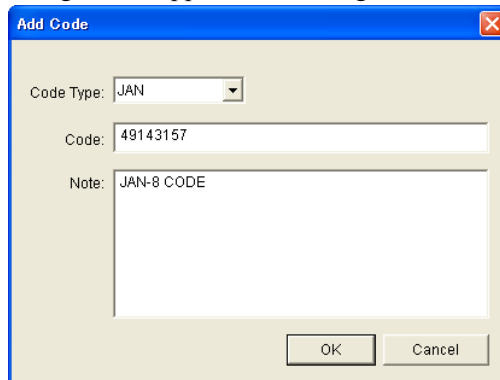
A dialog box titled "Add Code" with a blue header bar and a close button. It contains three input fields: "Code Type:" with a dropdown menu showing "JAN", "Code:" with a text box containing "49143157", and "Note:" with a text box containing "JAN-8 CODE". At the bottom right are "OK" and "Cancel" buttons.

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.

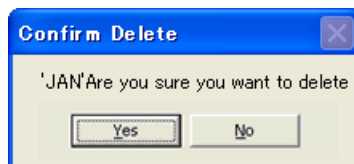
A dialog box titled "Confirm Delete" with a blue header bar and a close button. It contains a text box with the text "'JAN'Are you sure you want to delete". At the bottom are "Yes" and "No" buttons.

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.

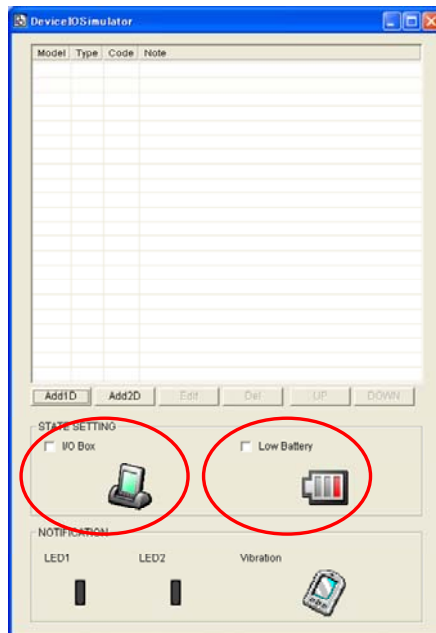


Figure 7.11

I/O Box

If this box is checked, a notification is issued that the connection between the DT-X30WM **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 DT-X30WM **Device Emulator**.

LED

When the DT-X30WM **Device Emulator** turns on the LED, the LED icon (LED2) in the I/O Simulator also turns on. See Figure 7.12.

Vibration

When the DT-X30WM **Device Emulator** vibrates, the vibration icon in the I/O Simulator also turns on. See Figure 7.12.

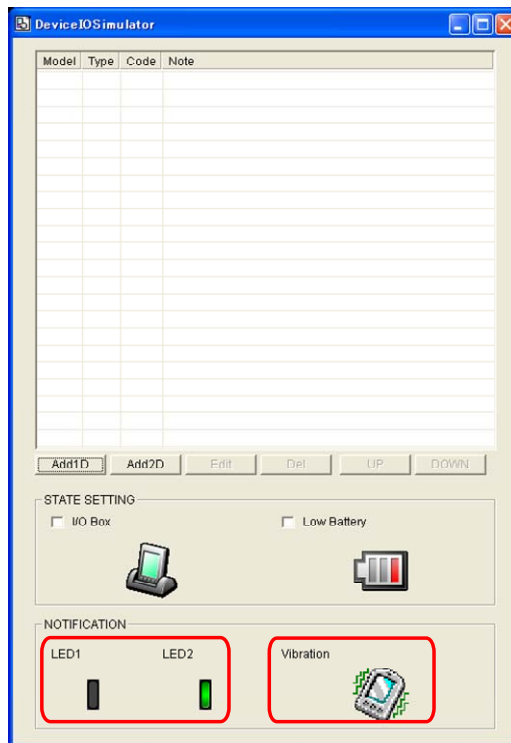


Figure 7.12

7.3.3 Connecting via ActiveSync

If debugging with the **Device Emulator** is carried out in 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**

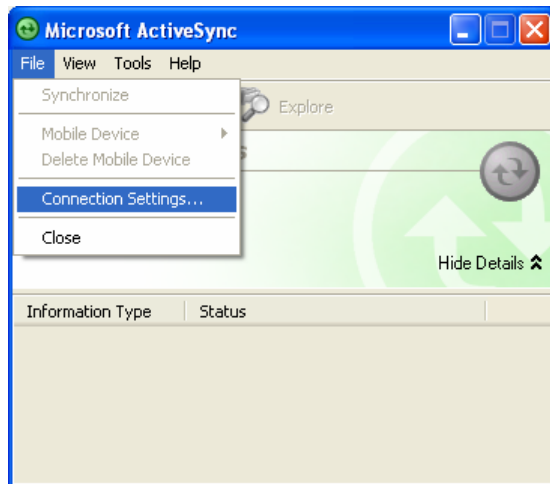


Figure 7.13

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.

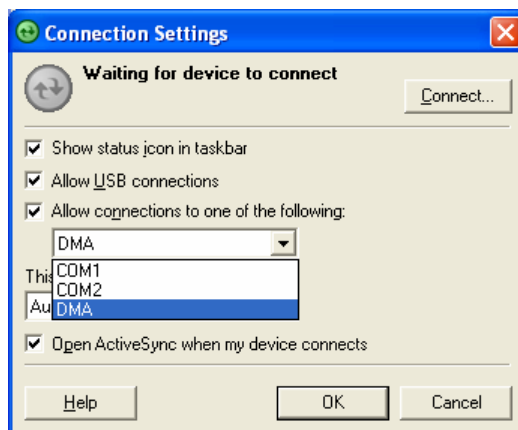


Figure 7.14

Connection via ActiveSync

The way to establish connection of the **Device Emulator** via **ActiveSync** is;

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**. Right-click **DT-X30WM Emulator** in the list and then choose **Cradle** in the popup menu. See Figure 7.15.



Figure 7.15



3. Make sure ActiveSync has started up and the  icon in the status bar changes to the  icon. See the emulated screen of DT-X30WM in Figure 7.16. The icon indicates that the connection via ActiveSync has been established.



Figure 7.16

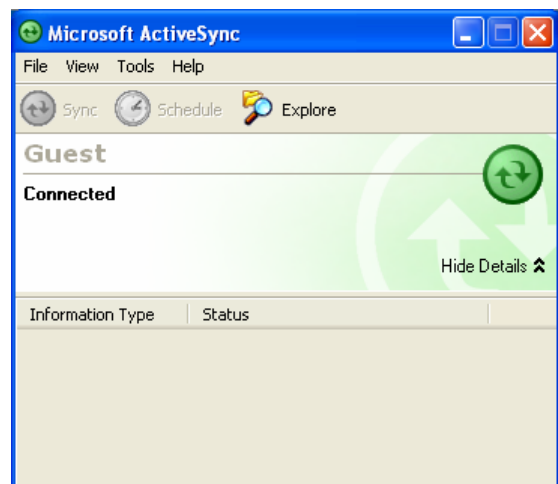


Figure 7.17

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 **DT-X30WM Emulator** 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 Chapter 8 “Visual Studio”.

Setting Build Configuration

Choose **Debug** in the Solution Configurations pull-down menu in Visual Studio 2008 or Visual Studio 2005 and **DT-X30WM Emulator** in the target device pull-down menu. See Figure 7.18.

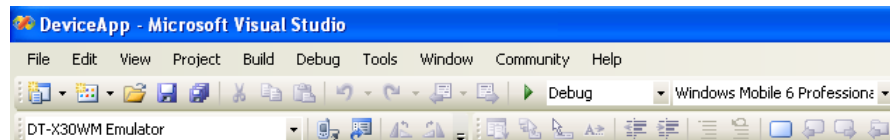


Figure 7.18

Debugging Applications

Basic Debug Operation

The debug operation used for the **Device Emulator** in Visual Studio 2008 or Visual Studio 2005 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.

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

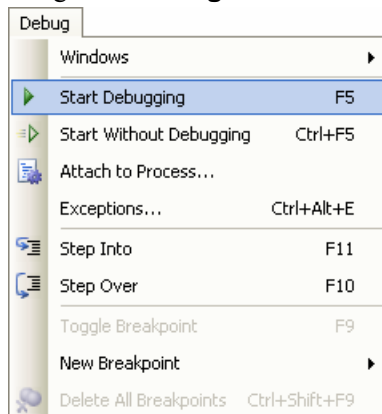


Figure 7.19

2. Similar to ordinary debugging operations with an actual DT-X30WM, the **Device Emulator** allows break point setting (circled in red in Figure 7.20) in the source code and step-by-step debugging.

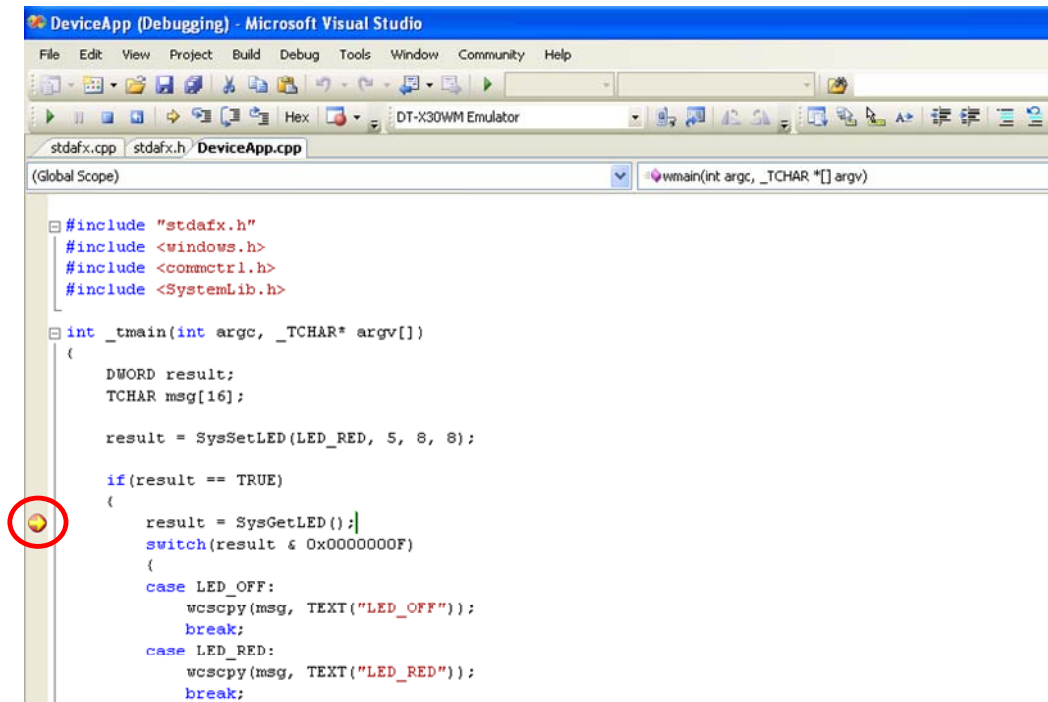


Figure 7.20

8. Visual Studio

This chapter describes application development steps in Visual Studio 2008 or Visual Studio 2005 to create an example program. The example program referred to in this chapter is an application program developed in three development environments - C++, Visual Basic and C# - using the CASIO Common Device Control Library or Common Device Control Class Library. The example program for all three environments repeats turning on and off the LED to blink for a period of 5 seconds. See Chapters 8.1 “Developing in C++”, 8.2 “Developing in VB” and 8.3 “Developing in C#” for each development environment.

For the list of the libraries provided by the DT-X30 BDK, refer to Chapter 1.2 “Library Configuration”. For detail about each function of the Common Device Control Library and Common Device Control Class Library, refer to Common Device Control Library Manual.

The example program in this chapter runs also in the Device Emulator. To use the emulator, rename **DT-X30 Device** to **DT-X30 Emulator**.

Notes:

- Functions of the Common Device Control Library which control various individual devices integrated in the DT-X30 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-X7, DT-X11, and IT-600 that use the same libraries.
- 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.
- Take care when using Visual Studio 2008 or Visual Studio 2005 to upgrade an existing application and always keep a backup of your existing project.

8.1 Developing in C++

This chapter describes necessary steps to develop the example application program (see Chapter 8) in C++ environment using the Common Device Control Library.

Before proceeding to creating the project, check if the installation folder for the Common Device Control Library has been registered in Visual Studio 2008 or Visual Studio 2005 by following steps, 1 to 8, below.

1. Navigate to **Tools** → **Options ...** and open **Options** screen.
2. Navigate to Projects and Solutions → Directories.
3. Choose Windows Mobile 6 Professional SDK (ARMV4I) in Platforms.
4. Choose Include files in **Show directories for:**.
5. If the installation folder has not been registered, append the following.
C:\Program Files\CASIO\MBSYS\include
6. Choose Library files in **Show directories for:**.
7. If the files have not been registered, append the following.
C:\Program Files\CASIO\MBSYS\lib\ARMV4I
8. Click **OK** button to close the screen.

Application Development Procedure

Subsequent steps show how to create and start up a simple program using one of the system functions. When running the program, the LED will blink in red for 5 seconds.

1. First, create **Win32 Smart Device Project** with its device name **SmartDeviceApp** for Visual C++ in Visual Studio 2008 or Visual Studio 2005.
2. Choose **Pocket PC 2003** in the Selected BDKs (SDKs in the screen) field in the Platforms of Win32 Smart Device Project Wizard and click **<** button.

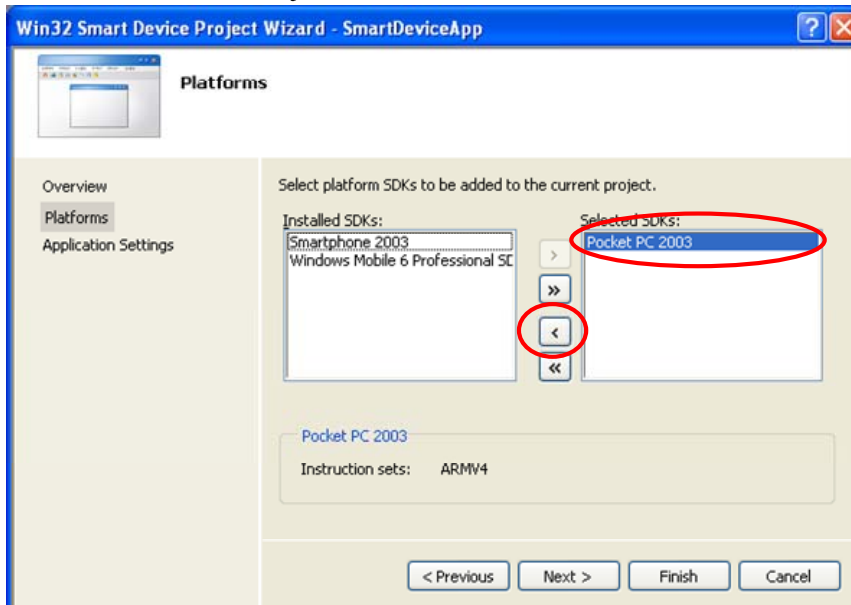


Figure 8.1

Choose **Windows Mobile 6 Professional SDK** in the **Installed SDKs** field and click **>** button.

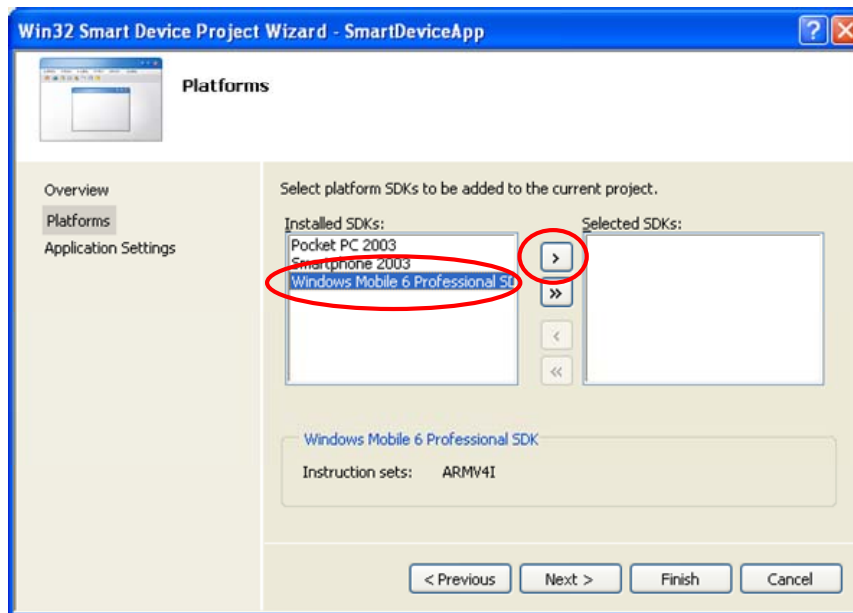


Figure 8.2

3. Click **Next >** button while **DT-X30** is kept being highlighted in the **Selected SDKs** field.

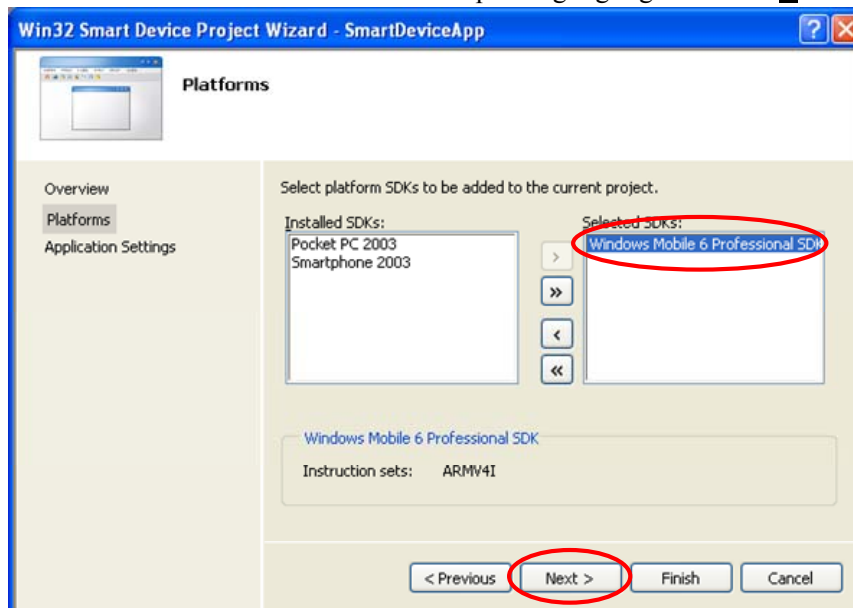


Figure 8.3

4. Choose Console Application for the Application type in the Application Settings menu, and click **Finish** button.
5. In Solution Explorer, click **SmartDeviceApp.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
```

6. Append the source code below in the main function of **SmartDeviceApp.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;
    case LED_BLUE:
        wcscpy_s( msg, _countof(msg), TEXT("LED_BLUE") );
        break;
    case LED_CYAN:
        wcscpy_s( msg, _countof(msg), TEXT("LED_CYAN") );
        break;
    case LED_MAGENTA:
        wcscpy_s( msg, _countof(msg), TEXT("LED_MAGENTA") );
        break;
    default:
        wcscpy_s( msg, _countof(msg), TEXT("LED_UNKNOWN") );
        break;
    }
    MessageBox(NULL, msg, TEXT("LED"), MB_OK);
}
```

7. 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 in Figure 8.4).

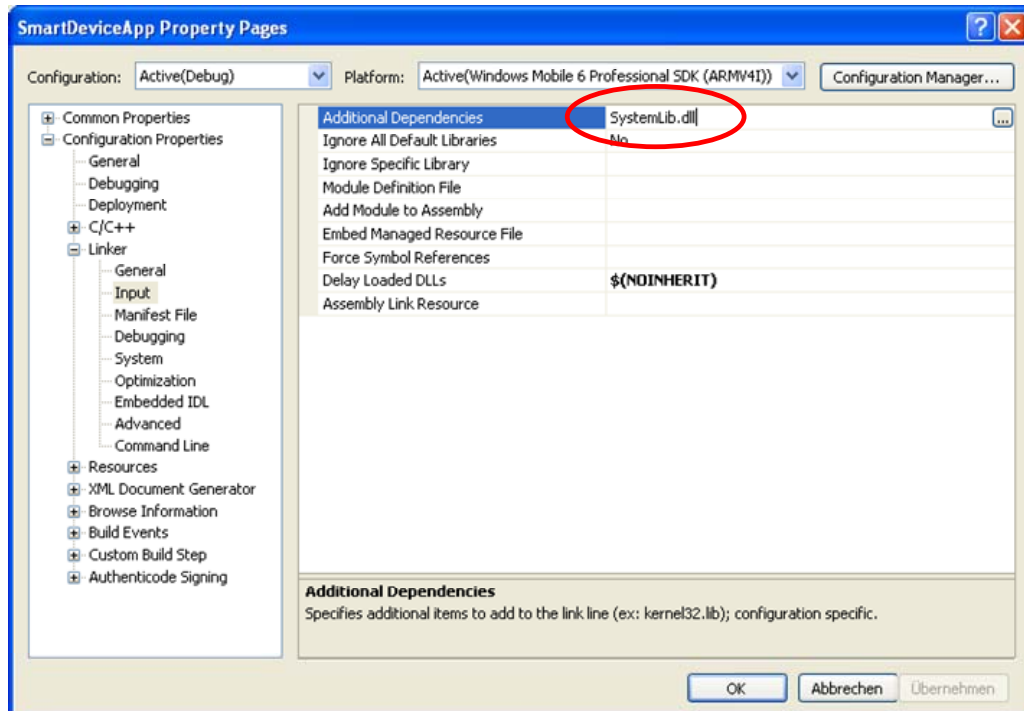


Figure 8.4

8. Choose **Windows Mobile 6 Professional Device** for the Target Device to establish connection with the PC.
9. Choose Debug in Visual Studio 2008 or Visual Studio 2005 and then click either Start Debugging or Start Without Debugging.
10. The project will be built and copied to the “\Program Files\<name of project>” folder by default in the DT-X30.
11. Check that the program runs correctly on the DT-X30.

8.2 Developing in VB

This chapter describes necessary steps to develop the example application program (see Chapter 8) in VB environment using the Common Device Control Class Library.

Application Development Procedure

Subsequent steps show how to create and start up a simple program using one of the system functions. When running the program, the LED will blink in red for 5 seconds.

1. Create a new VB Smart Device Application in Visual Studio.
2. In Solution Explorer, right-click **References** and click **Add Reference...**
3. Click **Browse** and navigate to the folder where you have stored the CASIO .NET library files. Highlight **SystemLibNet.dll** and click it to open. Click **OK** button.

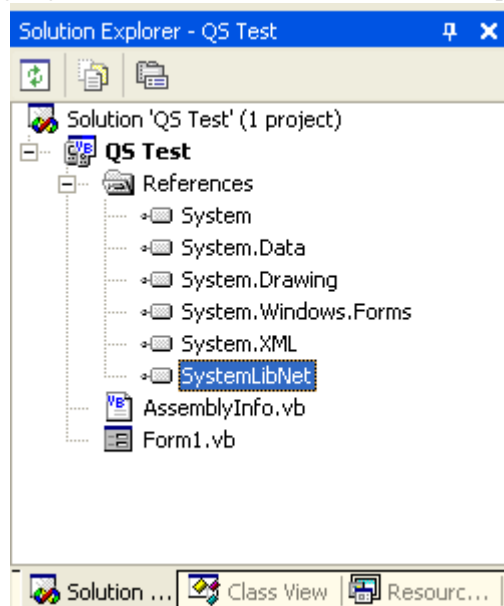


Figure 8.5

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

5. In the event function for the button to 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 Calib.SystemLibNet.Def.LED_BLUE
            msg = "LED_BLUE"
        Case Calib.SystemLibNet.Def.LED_CYAN
            msg = "LED_CYAN"
        Case Calib.SystemLibNet.Def.LED_MAGENTA
            msg = "LED_MAGENTA"
        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. Choose **Windows Mobile 6 Professional Device** for the Target Device to establish connection with the PC.
7. Choose Deploy <name of project> on Build menu.
8. The project will be built and copied to the DT-X30. 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 DT-X30.

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

8.3 Developing in C#

This chapter describes necessary steps to develop the example application program (see Chapter 8) in C# environment using the Common Device Control Class Library.

Application Development Procedure

Subsequent steps show how to create and start up a simple program using one of the system functions. When running the program, the LED will blink in red for 5 seconds.

1. Create a new C# Smart Device Application in Visual Studio .NET.
2. In **Solution Explorer**, right click **References** and click **Add Reference...**
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.

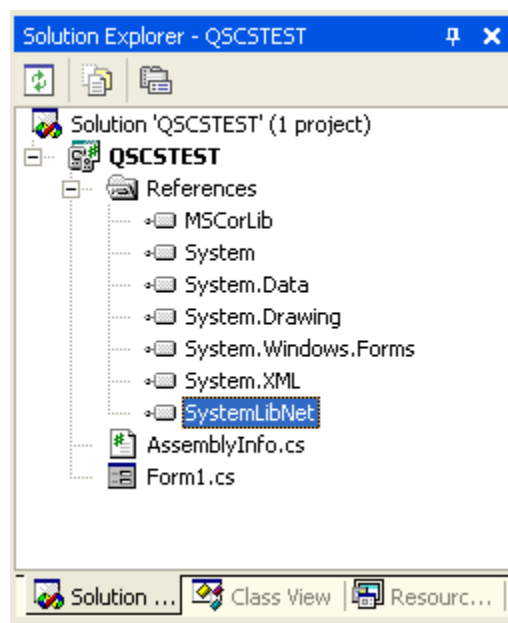


Figure 8.6

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

```
using Calib;
```


5. Add a button to your form, rename it **LED** and double click it and then 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;
        case SystemLibNet.Def.LED_BLUE:
            msg = "LED_BLUE";
            break;
        case SystemLibNet.Def.LED_CYAN:
            msg = "LED_CYAN";
            break;
        case SystemLibNet.Def.LED_MAGENTA:
            msg = "LED_MAGENTA";
            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. Choose **Windows Mobile 6 Professional Device** for the Target Device to establish connection with the PC.
7. Select Deploy <name of project> on Build menu.
8. The project will be built and copied to the DT-X30. 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. 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 and Library manuals on <http://world.casio.com/system/pa>.