Driver Design Guidlines for BE-300

Version 0.54, October 9, 2001
Casio Computer Co., Ltd

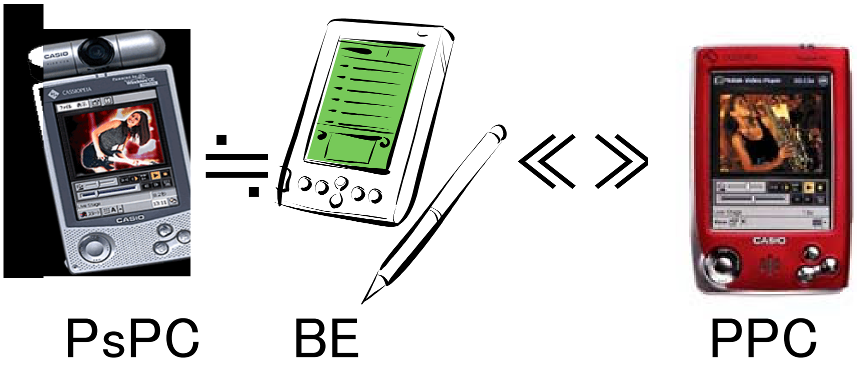
Revision log

Version No.	Date of revision	Details of revision
0.51	2001/05/14	First version
0.52	2001/07/06	“WIL” change into “BE-300”
0.53	2001/08/28	Development Environment maintenance. (p.15~)
	2001/08/28	20pin,CF WakeUp (p7,p11)
0.54	2001/10/09	A little description was changed.

1. Introduction

This manual contains information for developing drivers for the Cassiopeia BE-300. We will use the following abbreviations: BE-300 (Cassiopeia BE-300), PsPC (Palm Size PC), PPC (PocketPC), H/PC (HandheldPC).

1.1. BE Driver Design Overview (differences between BE and PPC, PsPC, H/PC)



If you have already developed other drivers for Windows CE (for example HPC, PsPC, PPC) creating drivers for Cassiopeia BE will be very easy. In most cases driver modification will be simple. However, since the installer is unique to BE, you must use BE installation tools.

Also, when newly developing drivers for the BE-300, you will be able to follow almost the same steps as those used when developing drivers for PsPC and PPC.

Using PsPC, PPC, HPC standard embedded drivers

Your hardware will work as is. Cassiopeia BE supports PPC standard embedded drivers. Refer to this for details on embedded drivers.

Using drivers you made that have no user interface

If your drivers work with Cassiopeia PPC they will most likely work with BE. If they work with other manufacturers' PPCs or PsPCs, they can be made to run by compiling with MIPS code.

If you have drivers that have a user interface and work with PsPC.

In most cases your drivers will work with no problems. There is no need to modify the driver core. From an appearance point of view you may feel the need to modify parts of the user interface code, but drivers should work with minimal modification.

If you have drivers that have a user interface and work with PPC.

Although there is no need to modify the driver core, you will need to modify parts of the PPC user interface code. You will need to modify so that you use the PsPC or BE UI libraries instead of the PPC UI library.

Newly developing drivers for the BE-300.

Obtain the driver development tool from Microsoft and the SDK for BE-300 from Cassio. You can develop BE-300 drivers in the same way as Windows CE (PsPC, PPC) drivers.

You do not need to modify the driver core.
Change sections made with the PPC UI to PsPC UI.
Do not save user data to RAM space; save it to Flash space.
Use Casio tools to create installers.

For details on the development environment see "6. Development Environment".

2. BE System

2.1. Hardware

The following illustration shows hardware components of a Cassiopeia BE.

Touch screen

The *touch screen* is a LCD covered by a *resistive touch panel*. The LCD has a portrait orientation with a 240 x 320 pixel resolution.

Key board

Some keys are put on the front panel.

Buzzer

Buzzer alarms with some information. Application can control it.

Audio

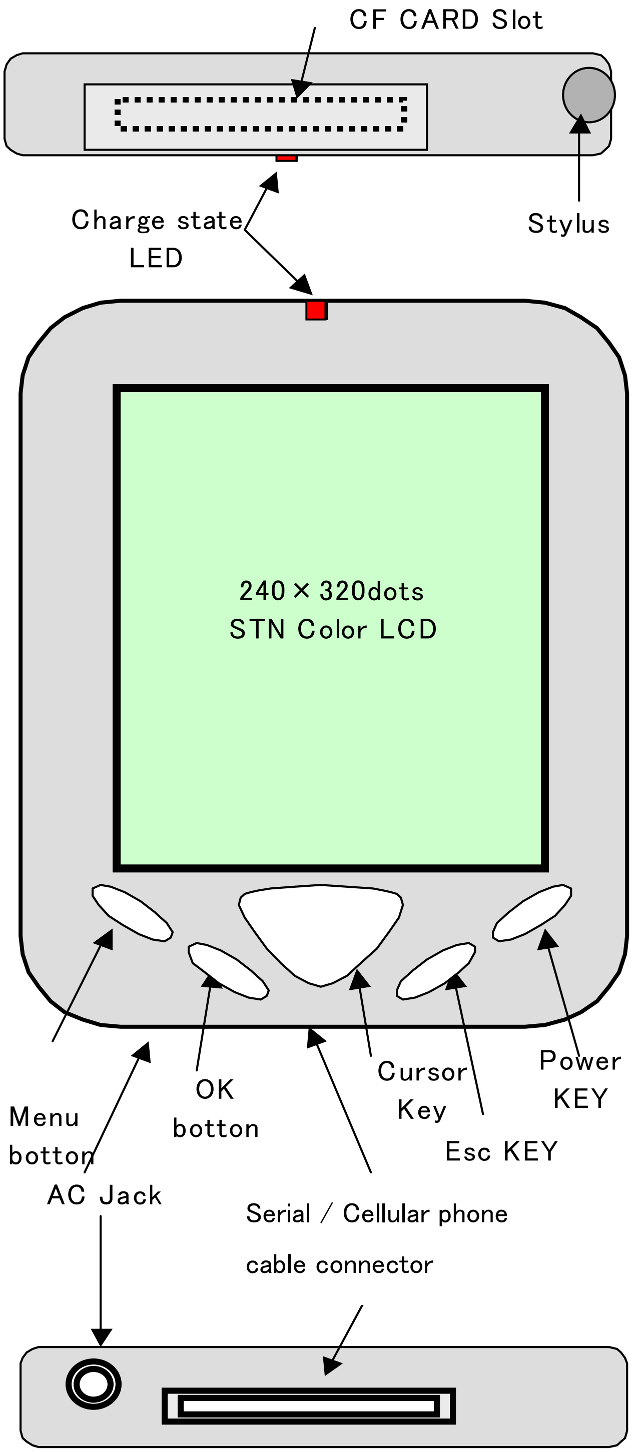
Audio output is supported with external headphone. You can use it to play sound. The BE does not support Audio input device.

Memory

BE has a least 2Mbyte built-in-flash memory for storage memory and 4M byte RAM for work memory.

Communication

You can connect PC with serial or USB. You can also connect to Network with wireless cable, CF wireless card.



2.2. Software

OS

Windows CE 3.0

Device manager

GWES

File manager

Communication

Built-in drivers

Wave device for plating audio

Wired Serial driver

USB serial class driver

PCMCIA socket driver

Card drivers (plug & play drivers)

ATA memory disk card driver

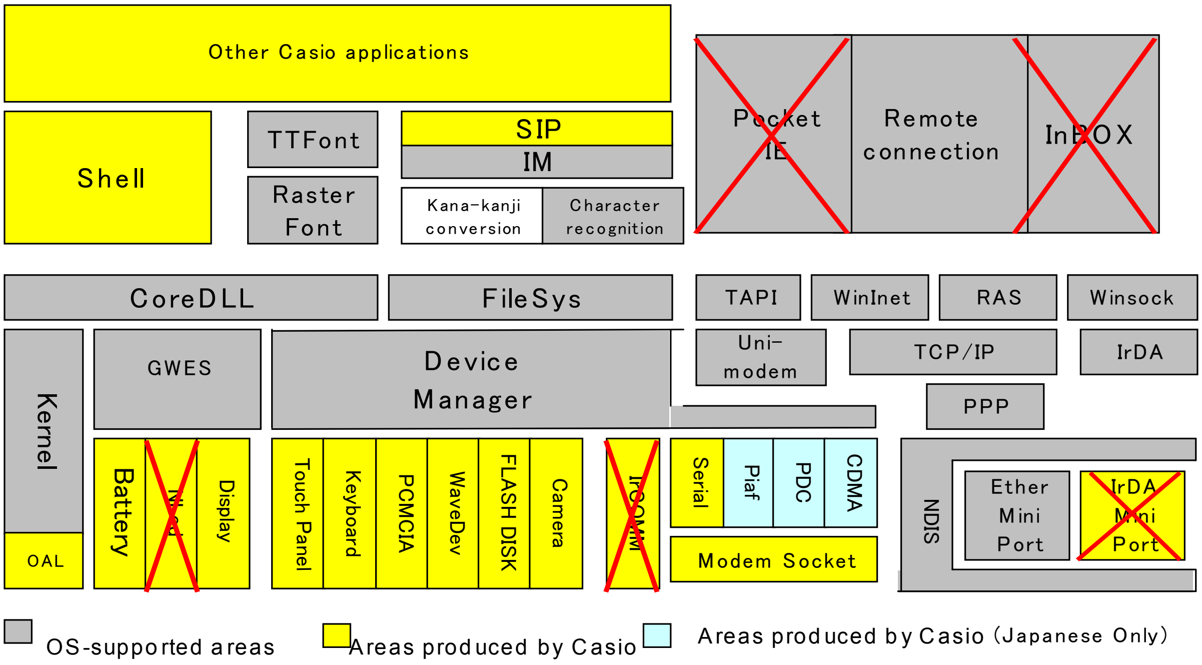
Hays compatible modem card driver

NE200 compatible LAN card driver

Memory

Built-in Flash disk memory for storage memory

Fig 2-1 Cassiopeia BE software structure



3. BE Hardware Interface

3.1. CF, PCMCIA Interface

3.1.1 Hardware Specification

Table 1

Item	CF	PCMCIA
Operating voltage	3.3V	3.3/5.0V
AC Spec	CFA standard	PC Card Standard
BUS width	16bit	16bit

3.1.2 Restrictions and Hints

Here we will introduce some hints on how to develop for hardware restrictions peculiar to Cassiopeia and for functions not supported by standard drivers.

~~Using a card modem to turn Cassiopeia power on~~

~~The card must have power when Cassiopeia power is OFF.~~

~~If there is power in the card, you can use its interrupt (IREQ/STSCHG) to WakeUp. In that case the interrupts must be enabled. You will have to update the driver in order to use these functions (turning power OFF but keeping the card powered, WakeUp using an interrupt). For detailed technical information either see Help for PB (Platform Builder 3.0) etc, or ask Microsoft.~~

Using an card to turn Cassiopeia power OFF.

When certain events are generated by applications running on Cassiopeia, you can do this by calling either a system function, or the power OFF function in the Cassiopeia library. For detailed technical information concerning system functions refer to PB or other Help; for the Cassiopeia library refer to the Cassiopeia BE SDK.

Power ON/OFF specification and control

When turning card power ON, according to PCMCIA specifications, the following minimum times are required: 10ms for the power supply to stabilize, 300ms for card power to stabilize. It follows that power on will take 310ms. Minimum card reset time is 1ms.

All power ON/OFF control is done from the Card Services API. Refer to Help for PB3.0, etc.

~~Keeping card power on when Cassiopeia power is OFF.~~

~~You must make a new card driver. With the card driver, when configuring the card in the PC socket (when calling **CardRequestConfiguration()**, set the CARD_CONIFG_INFO structure's *fAttributes* to CFG_ATTR_KEEP_POWERED.~~

Required driver software development, testing

You can develop with PB. Please refer to this.

3.2. Casio 20P Interface Standard

The following diagram shows the 20-pin connector located at the bottom of the unit. Since the 20-pin interface identifies the connecting cable type, it can interface to many machines.

Fig 3-1 Location of the 20-pin connector on the unit, and the position of pin 1

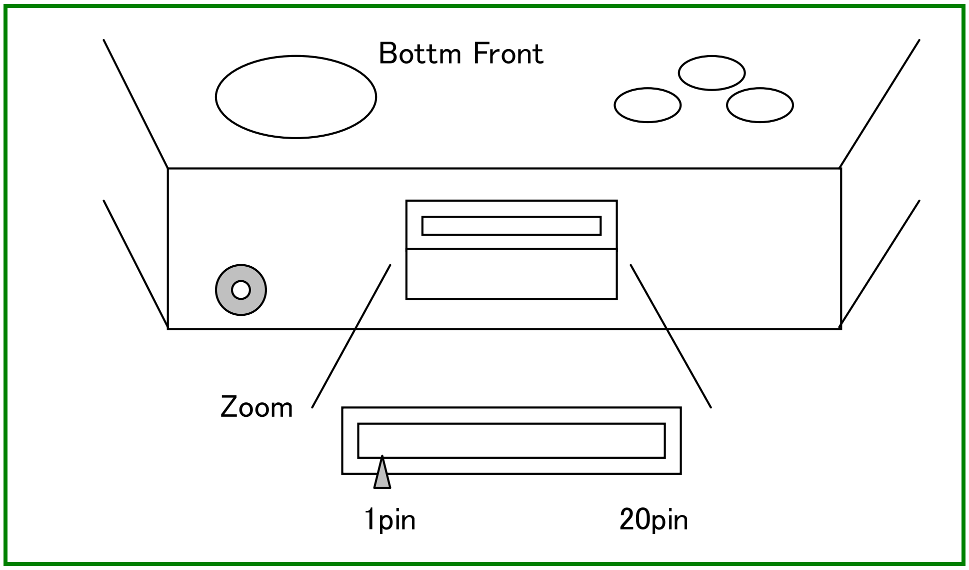


Table 3-1 Pin Layout of the 20-pin connector

PinNo	Name	Description	PinNo	Name	Description
1	ADPIN	Adapter Input 5.9V	11	RXD	RS-232C Data Line
2	SEL3	Cable Selector3	12	TXD	RS-232C Data Line
3	GND	Digital Ground	13	RTS	RS-232C Control Line
4	SEL0	Cable Selector0	14	DTR	RS-232C Control Line
5	SEL1	Cable Selector1	15	DSR	RS-232C Control Line
6	SEL2	Cable Selector2	16	NC	
7	SHDNB	Line Receiver control	17	+Data	USB I/O
8	NC		18	-Data	USB I/O
9	CTS	RS-232C Control Line	19	V3	3.3V
10	DCD	RS-232C Control Line	20	ADGND	AC Adaptor Ground

The SEL(3-0) combination will determine the interface mode.

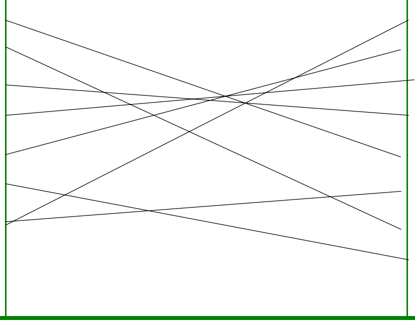
Only the RS-232C mode interface has been released to the public.

Table 2 Interface setting SEL combinations

SELNO	Interface	SELNO	Interface
0-000	RS-232C (modem) reserved for future use	1-000	RS-232C (PC connection mode)
0-001	Reserve	1-001	PIAFS32(DDI-P) interface
0-010	Reserve	1-010	Reserve
0-011	CdmaOne Cable interface	1-011	PDC,PDC-P interface
0-100	Reserve	1-100	USB connection interface
0-101	Reserve	1-101	PIAFS32 (ASTEL/DOCOM) interface
00-11X	NC, Disable	01-11X	Reserve

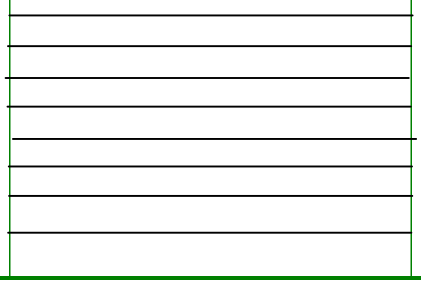
3.2.1 Connections from the 20-Pin Connector to Peripheral Circuits

Connecting from the 20-pin connector to a PC (Active Sync)

PinNo	CE side	Sample connection	PC side	PinNo
3	GND		DCD	1
9	CTS		RxD	2
10	DCD		TxD	3
11	RxD		DTR	4
12	TxD		GND	5
13	RTS		DSR	6
14	DTR		RTS	7
15	DSR		CTS	8
16	RI		RI	9

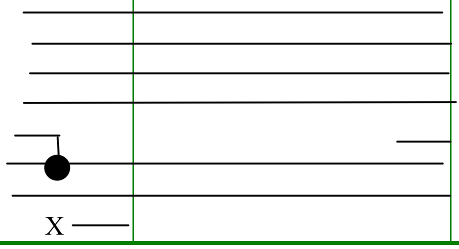
Connecting from the 20-pin connector to a modem

Straight output for each signal.

PinNo	CE side	Sample connection	Modem side	PinNo
3	GND		GND	
9	CTS		CTS	
10	RTS		RTS	
11	TxD		TxD	
12	RxD		RxD	
13	DSR		DSR	
14	DCD		DCD	
15	DTR		DTR	
16	RI		RI	

Connecting from the cradle to a modem.

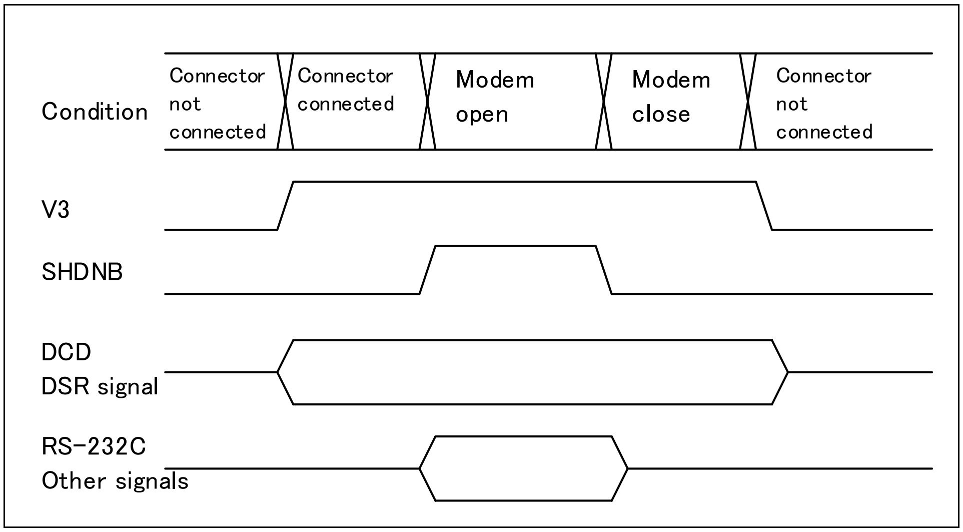
With some connections you must be careful with the DCD signal. In the cradle DCD and DSR are shorted.

PinNo	CE side	Unit process	Sample connection	Modem side	PinNo
	CTS			CTS	
	RTS			RTS	
	TxD			TxD	
	RxD			RxD	
	DSR			DSR	
	DCD			DCD	
	DTR			DTR	
	RI			RI	

3.2.2 20-Pin Power Control

Power control to peripheral circuits (a modem)

The unit provides a continuous 3.3V power supply. If you want to use a power saving mode when the modem is not in use, use the SHDNB signal. This signal will only become Hi when an application in the unit uses the modem.



Unit power ON control from a modem or PC

A modem cannot turn the unit ON.

When the unit is starting up, as soon the DCD signal is detected as becoming active, applications registered in

CeRunAppAtEvent(CeRunAppAtEvent(TEXT("AtPcCnct.exe.exe"), NOTIFICATION_EVENT_NONE);)

will be automatically started. "AtPcCnct.exe" (PC Connect) is started by default.

Since the RING pin is not supported, ON control using this pin is impossible.

Turning unit power off from a modem

A modem cannot turn the unit OFF. Install an application that will support this in the unit.

3.2.3 Running ActiveSync (Provisional)

Prohibiting Start

When applications have not been started, PC Connect will start if Active is input to the DCD pin[1].

To prevent this only make DCD Active for Modem Open. Be careful when using DSR and DCD with the current cradle because the unit shorts them.

Using your application, call the following functions and remove the PC Connect start setting. With this method, even if even if you insert the unit into a standard cradle PC Connect will not start—exercise caution.

```
CeRunAppAtEvent(TEXT("AtPcCnct.exe"), NOTIFICATION_EVENT_NONE);
CeRunAppAtEvent(TEXT("YourApplication.exe"),NOTIFICATION_EVENT_RS232_DETECTED);
```

---

[1] For a USB connection this will happen when DTR (the signal corresponding to DCD) is set in the USB class driver.

3.3. USB Interface

Although BE does have a USB interface, it is a device, not a host. Therefore you cannot use this interface to communicate on a 1 to 1 basis with another device. At present

USB Standard

Supports Universal Serial Bus Specification Revision 1.1 (http://www.usb.org/)

Supports USB full speed mode (12Mbps).

PC Connection

OS	Windows2000 ,Windows98,{testing Windows95(OSR2.2 or later)}and subsequent OS
Software	PC Connect (When developing, you are possible to use ActiveSync version 3.1.)

1. When PC power is ON and the USB cable is attached the only COM port you can use is USB.
2. You cannot connect with COM ports other than those set in the Control Panel.

4. BE User Interface Design Guide

You will need to modify parts of the user interface.

1. Location used for control items like Command Bar, View List.
2. Location where user settings etc are saved in the registry and files

Please make modifications after referring to the following.

Windows

PPC, PsPC for Windows can be used with no modification.

Shell Control

Shell API functionality and interface have changed significantly. Either change these functions to other ones or use functions supported by BE.

Common Control

Since Common Control uses the same implementation as PsPC, you should note that the appearance will be different that that of PPC. If the appearance is undesirable modify it.

Other Controls

Be careful when using Command Bar, List View, etc. Refer to the BE "Application Design Guidelines" and make modifications.

Saving User data in registry and files

With BE, when you save registry and files to RAM, unlike PPC, a reset will delete them. Make certain to write user settings etc to memory disk.

Registry

When you have changed the registry use CLBFlushRegistry to write all RAM resident registry to memory disk.

Files

The same as the registry, files saved in RAM will be deleted; you must save them on the memory disk, which is obtained with the GetSystemDisk function.??

Help

Help can only support very simple items. We recommend that you make the UI so that Help is not necessary.

5. Driver Installers

5.1. Installing from a PC

You cannot use a general Windows application installer's setup as is. Please use Casio's tools.

5.2. Installing from the Web

Casio will supply installation tools for this too (detailed specifications are now being written).

6. Development Environment

6.1. Software Development Tools

The following tools are required to develop BE-300 drivers. 1,2 is from Microsoft; 3 must be obtained from Casio.

Please Ask Microsoft for information concerning driver development methods and development tool utilization methods for Windows CE.

1. Microsoft Windows CE Platform Builder 3.0
2. Microsoft eMbedded Visual Tools 3.0
3. Cassiopeia BE-300 SDK

Additionally a USB or serial cable is required to connect the BE-300 unit to a PC and a BE-300.

6.2. Software Development Environment

The BE-300 uses an Embedded platform employing the Microsoft OS, Windows® CE 3.0.
If you have already developed other drivers for Windows CE (for example HPC, PsPC, PPC) , creating drivers for BE-300 will be very easy. In most cases driver modification will be simple. However, since the installer is unique to BE-300, you must use BE-300 installation tools.
By importing SDK to a tool, you can develop to BE-300 the same with developing the software of PsPC or PPC.

* BE-300 SDK can develop only the C/C++ language of the eMbedded Visual Tools 3.0.

6.2.1 Using drivers you made for PsPC or PPC

If your drivers work with Cassiopeia PPC and the drives don't have User Interface, they will most likely work with BE-300. If they work with other manufacturers' PPCs or PsPCs, they can be made to run by compiling with MIPS32 code.

If you have drivers that have a user interface and work with PsPC.

In most cases, you don't need to modify your drivers. From an appearance point of view you may feel the need to modify parts of the user interface code, you can modify by the following environment.

If you have drivers that have a user interface and work with PPC.

You have to modify the part of user interface code. You are able to use the following environment for modifying.

Microsoft embedded Visual Tools 3.0 + BE-300 + E-300 SDK

6.2.2 Making new drivers for BE-300

If you make new drivers for BE-300, you can choose the followings ways.

Case 1

For the Part Of Drivers

Microsoft Windows CE Platform Builder 3.0 + BE-300

For the Part of User Interface

Microsoft embedded Visual Tools 3.0 + BE-300 + BE-300-SDK

Case 2

For the Part of Drivers

Microsoft VC++6.0 + Microsoft Windows CE Tool Kit for VC++6.0 + BE-300

For the Part of User Interface

Microsoft embedded Visual Tools 3.0 + BE-300 + BE-300-SDK

The DDK of Microsoft Windows CE Tool Kit for VC++6.0 is supported until CE2.11 only. If you hope to develop the drivers that use the APIs supported since CE3.0 later, you have to choose the way of "Case1".

6.3. Debug

6.3.1 The method of Debug.

Debugging can be performed with outputting creation and the debugging message of a log trace file to PC.

If you can make a CEPC code, your driver code can debug logically by CEPC.
You debug user interface using the eMbedded Visual Tools.
After logically debugging, you re-build with our MIPS code and check whether your driver can work on our device.

6.3.2 The method of Remote debugging

The remote debugging described here requires Microsoft Windows CE Platform Builder 3.0 and Microsoft eMbedded Visual Tools.

You can debug by setting the include file path and liblary path of Platform Builder to eMbedded Visual Tools.
(remote debugging, registry edit, the download to a target, etc.)

The cautions about use of remote debugging:

When you want to remote debugger, you must set the execute file to load your driver (DLL) Actually your driver is loaded by device.exe. However you can not select this.
eMbedded Visual Tools try to load device.exe and it will be exited soon because device.exe is already loaded.
Then you have to create a test application to load your driver instead of device.exe.
This means, you can not debug at the same condition of final but you can debug somehow your driver APIs at the pseudo environment.
After debugging, you re-build at the release condition and check whether your driver can work on our device.

Installation and Setting:

1. Install BE-300 SDK into eMbedded Visual Tools according to the installer guide of BE-300 SDK.

2.Setup ActiveSync on both your PC and BE-300 device according to BE-300 SDK installer guide.

3. Create your project for card driver.
  - Execute "Microsoft eMbedded Visual C++"
  - Create a your card driver project:
    - On File menu, click New, click Project Tab and select WCE dynamic-link library
    - Input your project name
      ex. project name: SampleCard
    - Select Win32 (WCE MIPS) in CPUs selection and press OK button
    - Select an empty project and click finish button

3.Modify project setting
  - Seclet platform

  - On Build menu, click "Set Active platform .." and click CASIO_BE300
- Set path to platform builder's include
  - On Tools menu, click "options", click Directories tab, Select Include file list in "Show directories for" catalogue.
  - Add the following paths in Directories list to the top of list.

    - %_WINCEROOT%¥PUBLIC¥COMMON¥DDK¥INC
    - %_WINCEROOT%¥PUBLIC¥COMMON¥OAK¥INC

  - Set path to platform builder's library
    - On the same Options dialog and at the same tab.
      - Select "library file" in "Show directories for" catalogue.
  - Add the following paths in Directories list to the top of list.

    - %_WINCEROOT%¥PUBLIC¥COMMON¥SDK¥LIB¥MIPS¥R4100¥CE¥RETAIL

- On the same Options dialog, click download tab
  - remove "always tranfer cef executable" check

4. Write your code
- Write your code,
  - Write driver code according to the guide of platform builder.
  - Write user interface code according to the guide of eMbedded visual tools and CASSIOPEIA BE documentation.

ex) If you want a sample driver, copy atadisk driver from platform builder tree.
  - Copy atadisk folder to your work folder
    - ex) Copy to SampleCard¥src
  - In this case, you have to modify project setting
  - Add define file option: /def:"ATADISK.def" in linker option
  - Add entry option: /Entry:"ATADiskEntry"