CC864/UC864 Windows CE 6.0 User Guide

1vv0300821 rev. 2 – 2010-07-14

Disclaimer

The information contained in this document is the proprietary information of Telit Communications S.p.A. and its affiliates ("TELIT"). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Telit, is strictly prohibited.

Telit makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Telit does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information.

Telit disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules.

Telit reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will, nevertheless be incorporated into new editions of this application note.

Copyright: Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights are reserved.

Applicable Products

Product
UC864-E
UC864-E-DUAL
UC864-E-AUTO
UC864-G
UC864-K
UC864-WD
UC864-WDU
CC864-DUAL
CC864-SINGLE

Contents

1. Introduction

1.1. Scope

This user guide serves the following purpose:

- Provides details about the CC864/UC864 Telit Modules.
- Describes how to compile, include in a kernel image and use the CC864/UC864 Windows CE 6.0 driver for a headless system.

1.2. Audience

This User Guide is intended for software developers who develop applications using CC864/UC864 modem.

1.3. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit's Technical Support Center (TTSC) at:

TS-EMEA@telit.com
TS-NORTHAMERICA@telit.com
TS-LATINAMERICA@telit.com
TS-APAC@telit.com

Alternatively, use:
http://www.telit.com/en/products/technical-support-center/contact.php
For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:
http://www.telit.com
To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).
Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.
Telit appreciates feedback from the users of our information.

1.4. Product Overview

UC864 modules contains a fully featured HSDPA modem and UMTS/GSM/GPRS module, compatible with the other Telit GSM/GPRS modules.
CC864 modules contains a fully features CDMA modem compatible with the other Telit GSM/GPRS modules.

1.5. Document Organization

This manual contains the following chapters:

- “Chapter 1, Introduction” provides a scope for this manual, target audience, technical contact information, and text conventions.
- “Chapter 2, System setup” describes how to add the CC864/UC864 USB driver in a Windows CE 6.0 image.
- “Chapter 3, Device Driver” describes how to use the CC864/UC864 driver for interacting with the CC864/UC864 modem.

1.6. Text Conventions

This section lists the paragraph and font styles used for the various types of information presented in this user guide.

Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

Format	Content
`Arial monospaced`	Windows CE shell commands at command prompt, filesystem paths, source code examples and menu items

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

1.7. Document history

Revision	Date	Changes
ISSUE #0	2009-04-10	First Release
ISSUE #1	2009-10-08	Added support of all UC864 and CC864-DUAL Telit Modules
ISSUE #2	2010-07-14	Updated Applicability Table

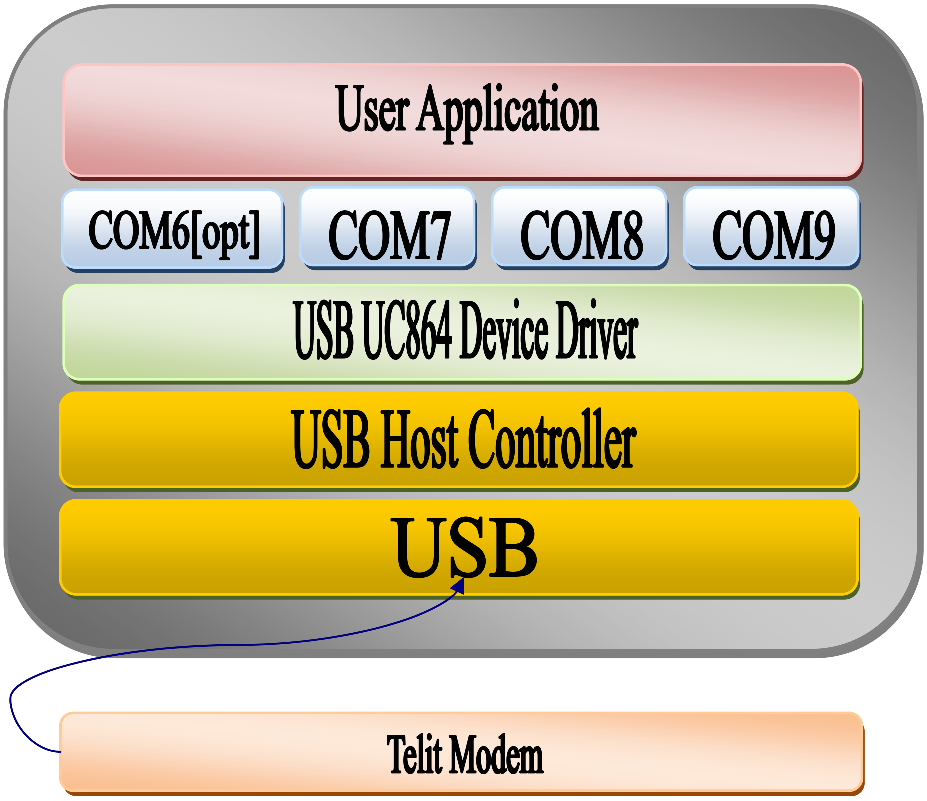
2. System setup

In this chapter it is described the general architecture of a Windows CE system with attached an CC864/UC864 module, the requirements needed for using the CC864/UC864 and how to integrate the CC864/UC864 device driver in a Windows CE image. The following table shows the reference platforms which this guide has been tested for:

Device	Operating System
AT92SAM9260	Windows CE 6.0 R2

2.1. General Overview

In the following image it is shown a diagram of USB software/hardware interaction in a Windows CE system with a CC864/UC864 attached:



From the bottom of the stack:

- Telit modem connected through an USB port.
- USB: serial bus for interfacing devices to a host computer.
- USB Host Controller: a combination of hardware and software that is responsible for the following actions:
  - Detecting the insertion and removal of USB devices
  - Managing flow control between the host and USB devices
  - Managing data flow between the host and USB devices
  - Collecting status

  - Providing power to attached USB devices

- USB CC864/UC864 Device Driver: piece of software that allows the CC864/UC864 to be seen by the user application as connected through serial ports rather than USB port.
- Virtual serial ports (COM6, COM7, COM8 and COM9): serial ports created by the CC864/UC864 device driver for accessing the modem; they can be used as real physical serial ports.

UC864-E, UC864-E-AUTO, UC864-K, UC864-WD, UC864-WDU, CC864-K:

- COM7: Telit HSDPA modem port, used for normal modem/application interaction (e.g. AT commands sending, data connection…).
- COM8: Telit Diagnostics, used for firmware update.
- COM9: Telit Auxiliary, used for debugging purposes.

UC864-G, CC864-DUAL:

- COM6: Telit HSDPA modem port, used for normal modem/application interaction (e.g. AT commands sending, data connection…).
- COM7: Telit Diagnostics, used for firmware update.
- COM8: Telit NMEA, used for firmware update.
- COM9: Telit Auxiliary, used for debugging purposes.

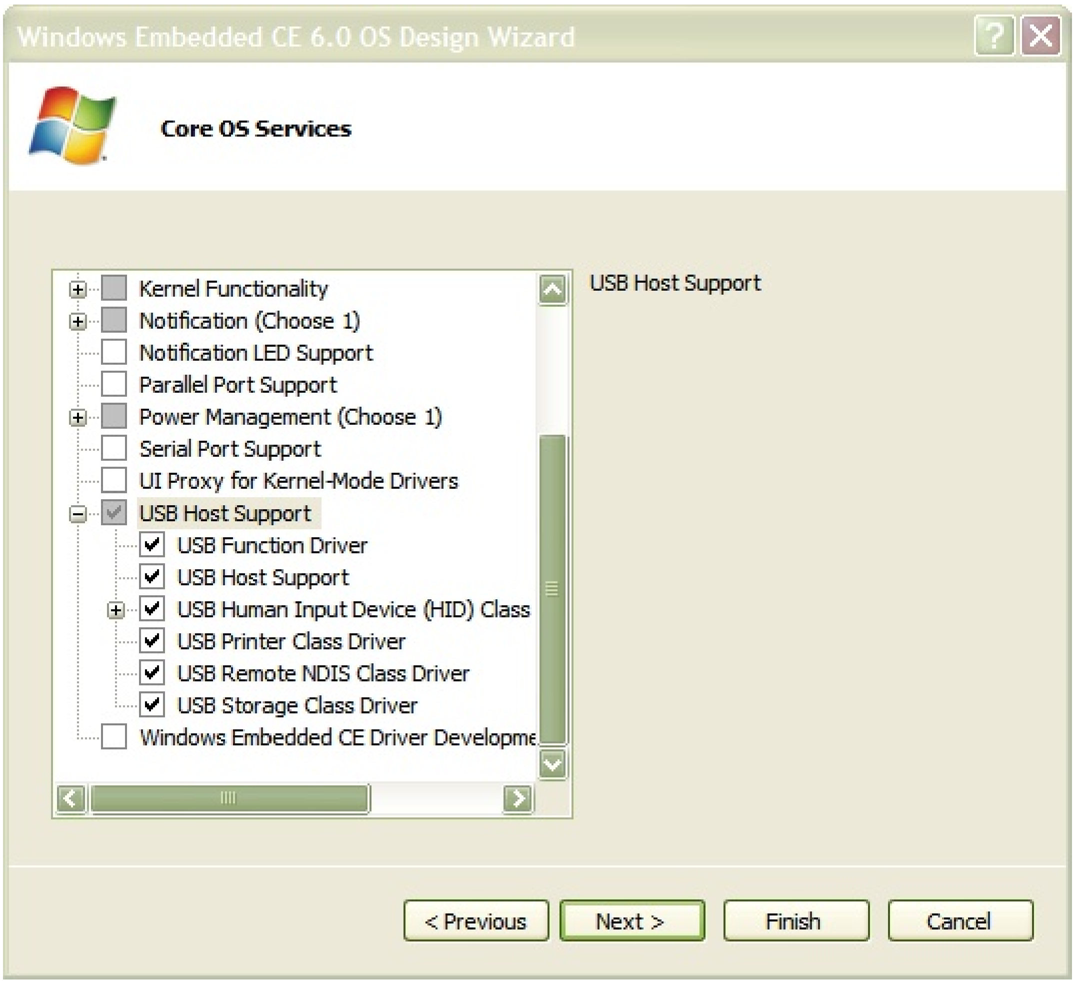
- User Application: piece of software written by the customer that uses CC864/UC864 features through virtual serial ports.

2.2. Pre-requirements

For correctly building a Windows CE 6.0 image with the CC864/UC864 USB driver on a headless system the following pre-requirements need to be satisfied:

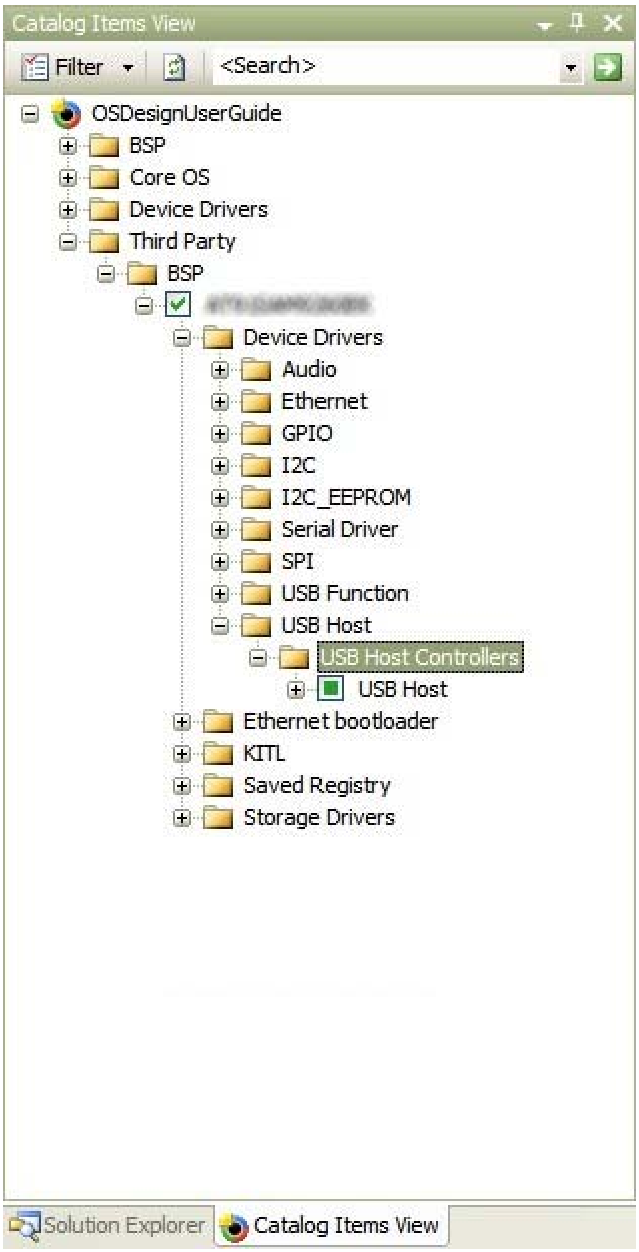
- Microsoft Visual Studio 2005, with Platform Builder for CE 6.0.
- An OS design with the USB host controller supported.
- Knowledge of Windows CE OS image and subproject creation.

Please note that there are different ways for enabling the USB host controller and they depend on the considered hardware. In our test platform the USB host controller can be enabled during the initial image creation wizard (in the Core OS Services step), as the following image suggests:



- ▨ Kernel Functionality
- ▨ Notification (Choose 1)
- ☐ Notification LED Support
- ☐ Parallel Port Support
- ▨ Power Management (Choose 1)
- ☐ Serial Port Support
- ☐ UI Proxy for Kernel-Mode Drivers
- ☑ USB Host Support
  - ☑ USB Function Driver
  - ☑ USB Host Support
  - ☑ USB Human Input Device (HID) Class
  - ☑ USB Printer Class Driver
  - ☑ USB Remote NDIS Class Driver
  - ☑ USB Storage Class Driver
- ☐ Windows Embedded CE Driver Developme

or it can be added using the catalog as the following image suggests:



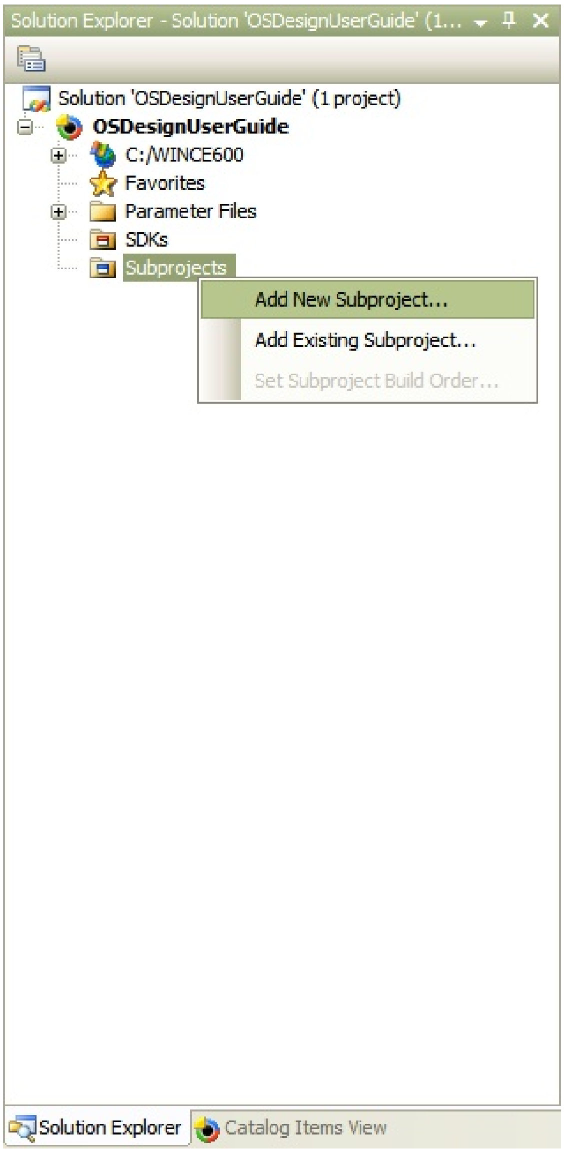
Please note again that this step could be different for your platform.

2.3. Adding CC864/UC864 support in the OS image

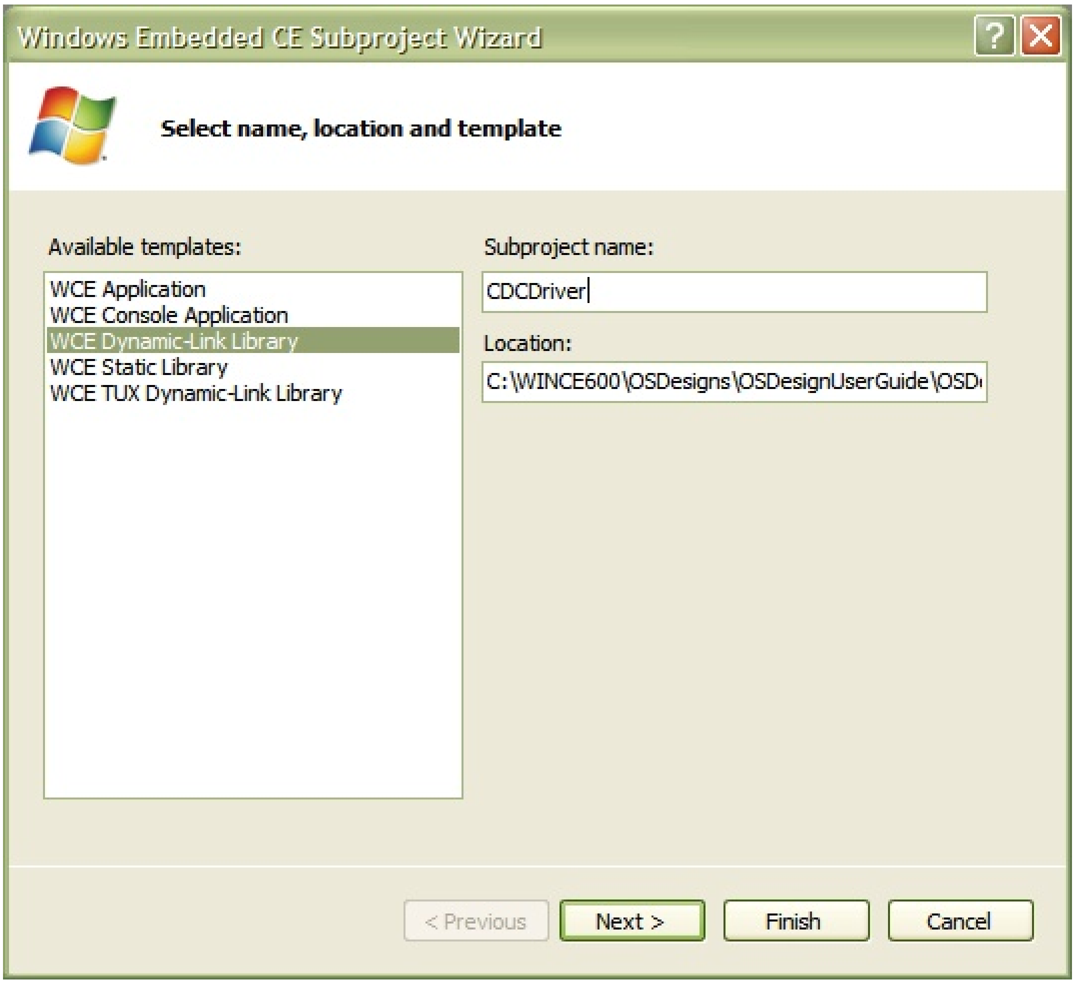
2.3.1. Compiling from source code

For adding CC864/UC864 support in the OS image, having the source code files, follow these steps:

- Having loaded the solution for your custom OS design right click on the *Subprojects* folder and choose the voice *Add new subproject*.


Solution Explorer - Solution 'OSDesignUserGuide' (1...
Solution 'OSDesignUserGuide' (1 project)
OSDesignUserGuide
C:/WINCE600
Favorites
Parameter Files
SDKs
Subprojects
Add New Subproject...
Add Existing Subproject...
Set Subproject Build Order...
Solution Explorer
Catalog Items View

- In the wizard choose WCE Dynamic-Link library and type *CDCDriver* as the subproject name.



Windows Embedded CE Subproject Wizard

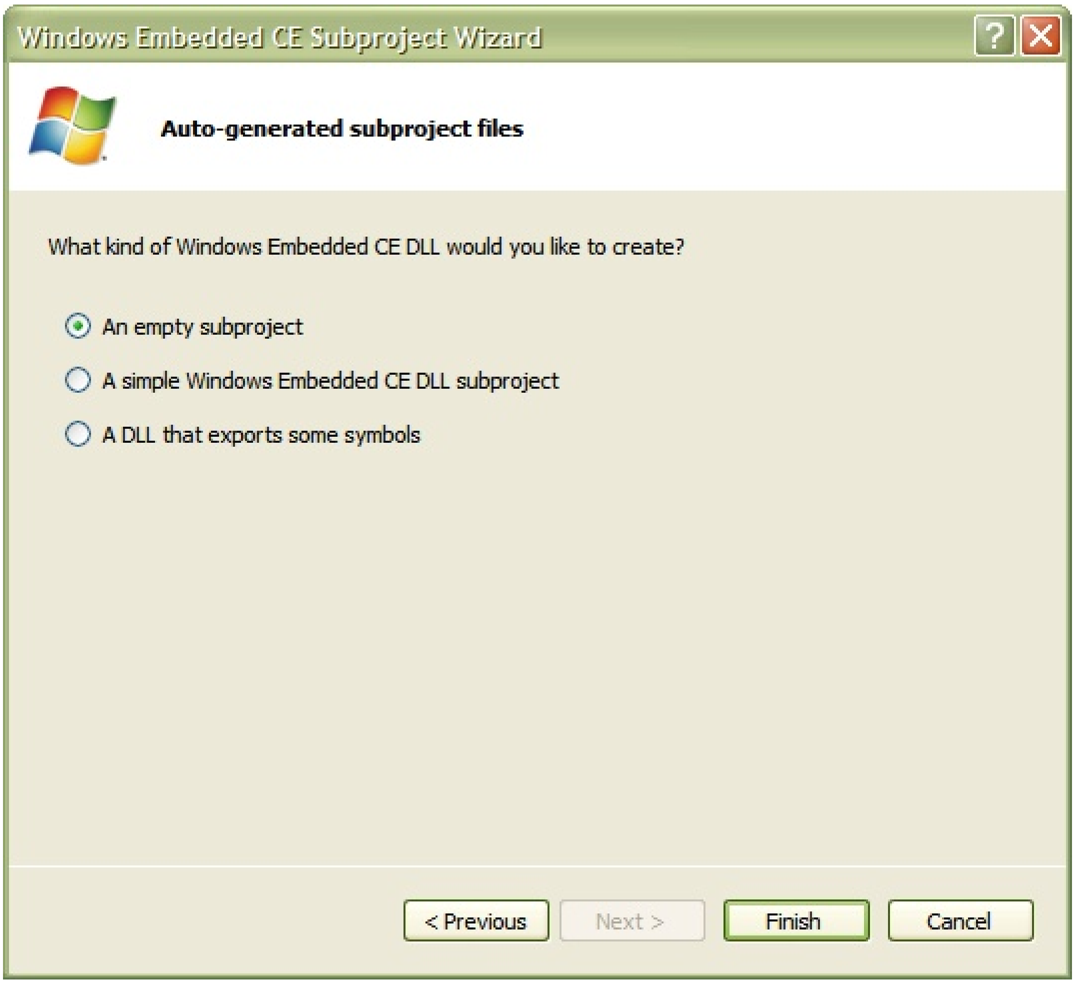
Select name, location and template

Available templates:
- WCE Application
- WCE Console Application
- WCE Dynamic-Link Library
- WCE Static Library
- WCE TUX Dynamic-Link Library

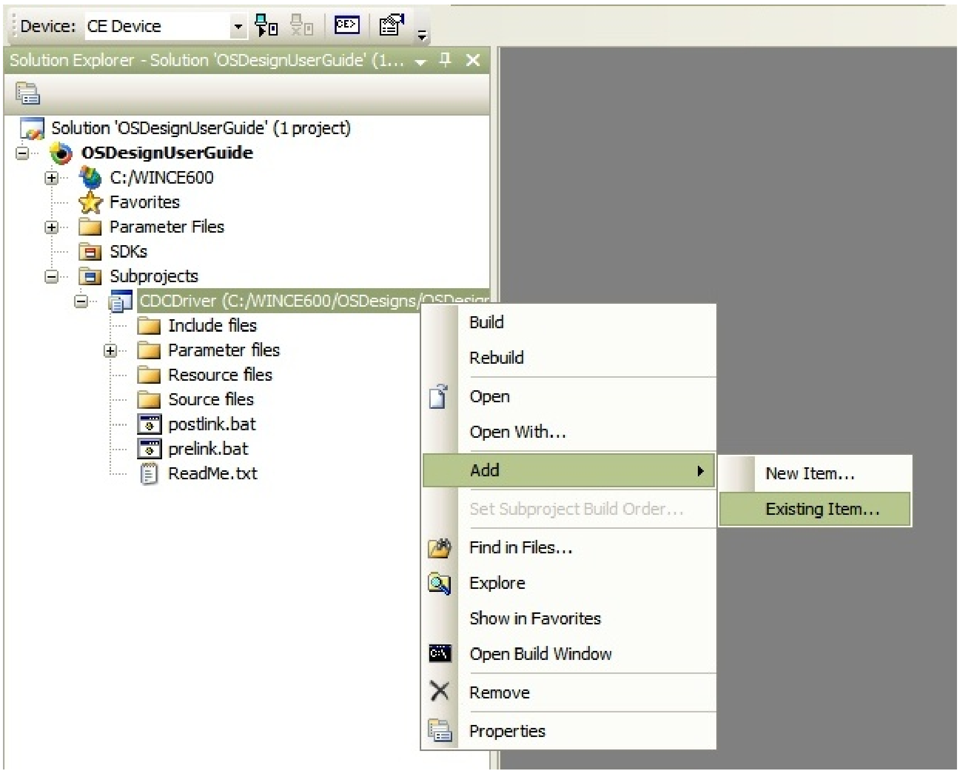
Subproject name: CDCDriver

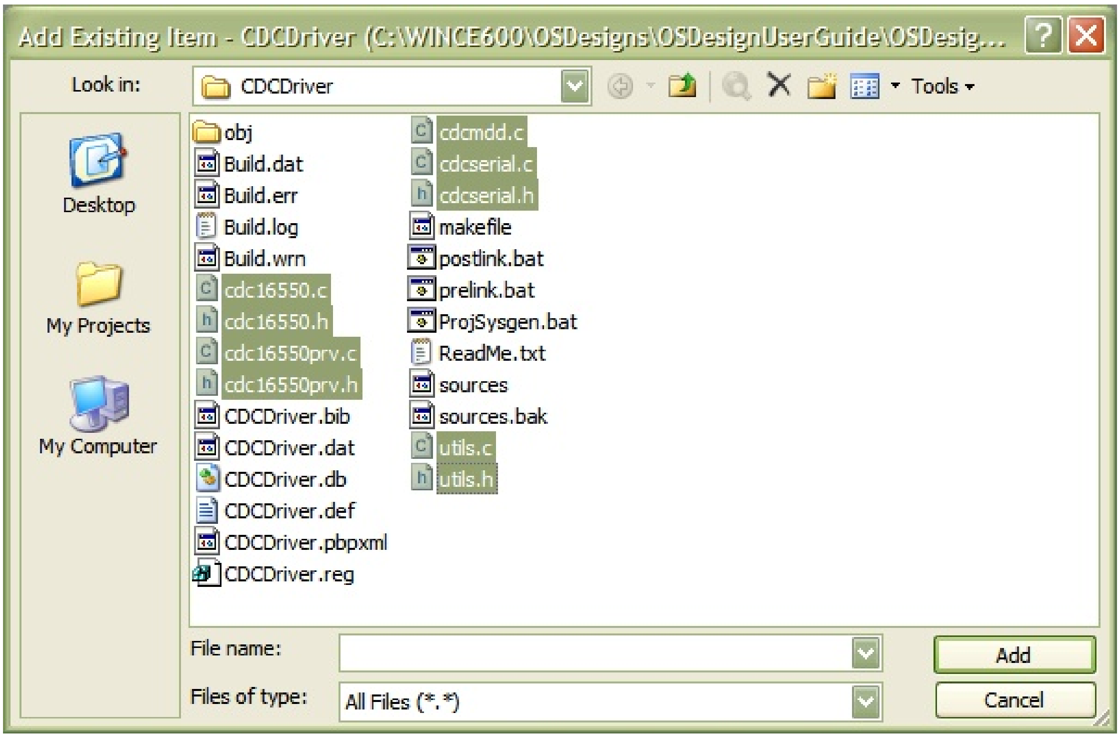
Location: C:\WINCE600\OSDesigns\OSDesignUserGuide\OSD

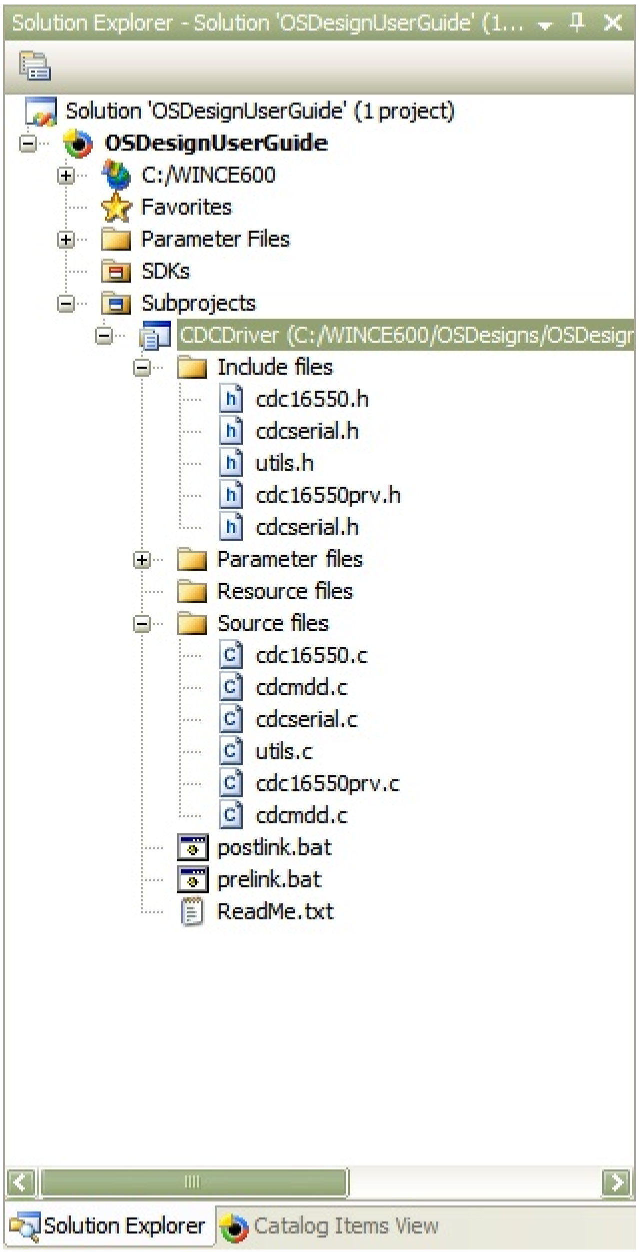
Click *Next* and in the new window choose *An Empty Subproject*. Click *Finish*.


Windows Embedded CE Subproject Wizard
Auto-generated subproject files
What kind of Windows Embedded CE DLL would you like to create?
● An empty subproject
○ A simple Windows Embedded CE DLL subproject
○ A DLL that exports some symbols
< Previous
Next >
Finish
Cancel

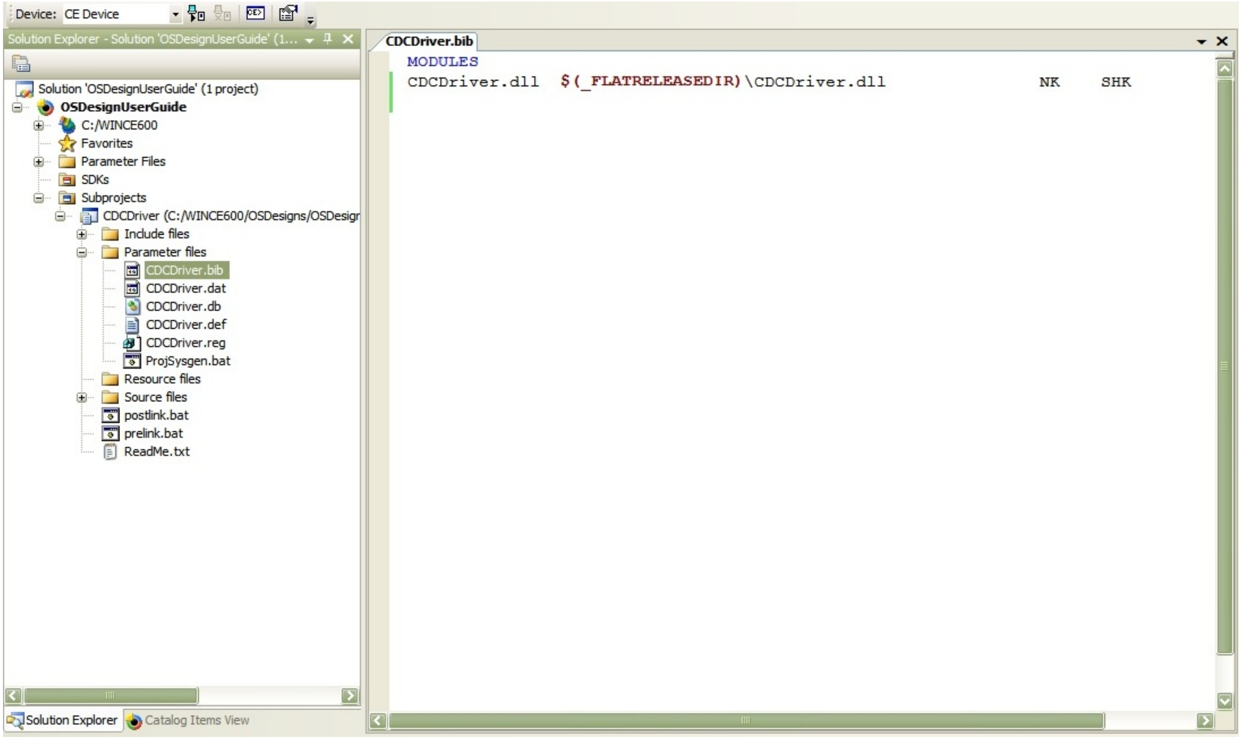
- Copy the following files inside the directory of the CDCDriver project and add them to the project (right click on the subproject name and choose the voice *Add → Existing Item*):
    - cdc16550.h, cdc16550.c
    - cdc16550prv.h, cdc16550prv.c
    - cdcserial.h, cdcserial.c
    - cdcmdd.c
    - utils.h, utils.c


Device: CE Device
Solution Explorer - Solution 'OSDesignUserGuide' (1...
Solution 'OSDesignUserGuide' (1 project)
OSDesignUserGuide
C:/WINCE600
Favorites
Parameter Files
SDKs
Subprojects
CDCDriver (C:/WINCE600/OSDesigns/OSDesign
Include files
Parameter files
Resource files
Source files
postlink.bat
prelink.bat
ReadMe.txt
Build
Rebuild
Open
Open With...
Add
Set Subproject Build Order...
Find in Files...
Explore
Show in Favorites
Open Build Window
Remove
Properties
New Item...
Existing Item...


Add Existing Item - CDCDriver (C:\WINCE600\OSDesigns\OSDesignUserGuide\OSDesig...
Look in:
CDCDriver
Tools
Desktop
My Projects
My Computer
obj
Build.dat
Build.err
Build.log
Build.wrn
cdc16550.c
cdc16550.h
cdc16550prv.c
cdc16550prv.h
CDCDriver.bib
CDCDriver.dat
CDCDriver.db
CDCDriver.def
CDCDriver.pbpxml
CDCDriver.reg
cdcmdd.c
cdcserial.c
cdcserial.h
makefile
postlink.bat
prelink.bat
ProjSysgen.bat
ReadMe.txt
sources
sources.bak
utils.c
utils.h
File name:
Files of type:
All Files (*.*)
Add
Cancel

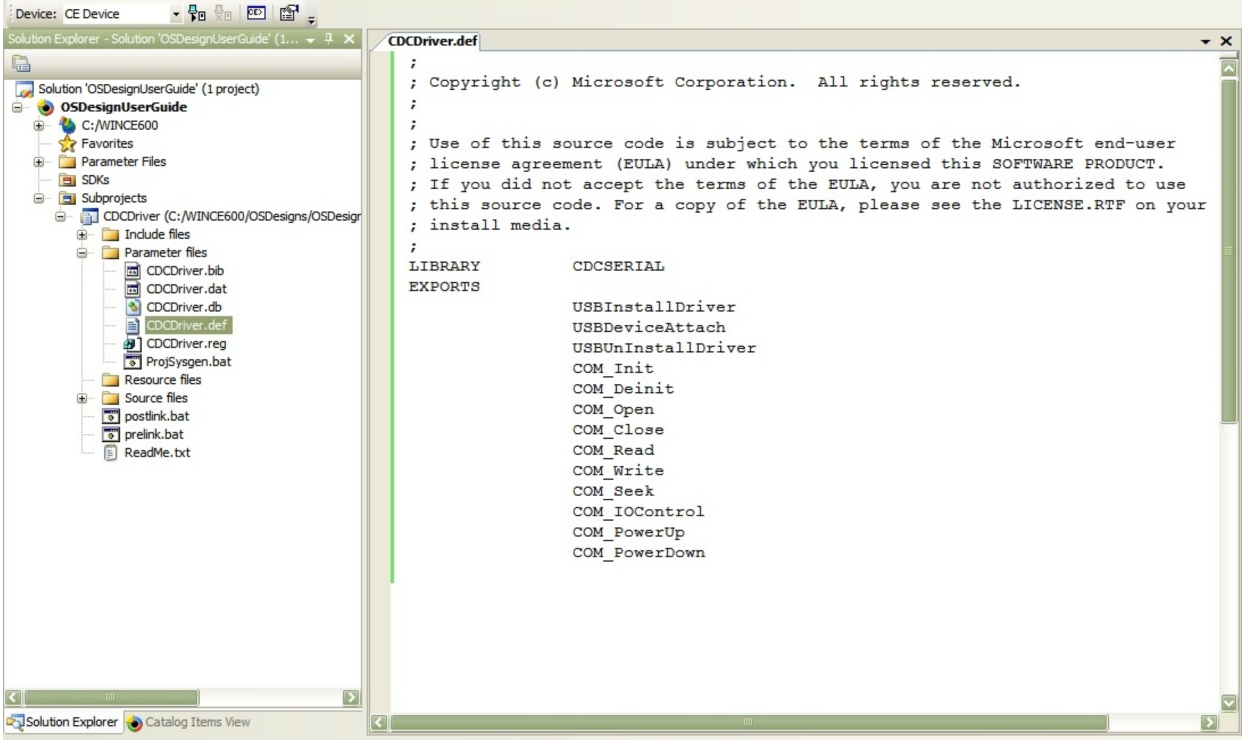


- Open the file *CDCDriver.bib* and add the word **SHK** at the end of the second line (after the word **NK**) as the image suggests.



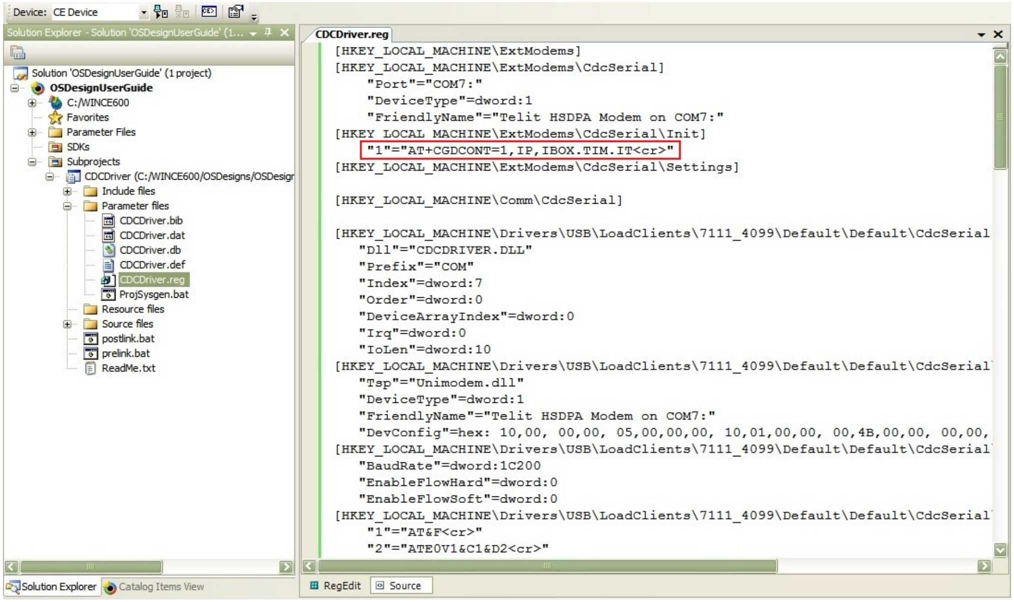
```
MODULES
CDCDriver.dll   $(_FLATRELEASEDIR)\CDCDriver.dll               NK     SHK
```

- Open the file *CDCDriver.def* and paste the content of the same file provided with the driver source code as the image suggests.



```
;
; Copyright (c) Microsoft Corporation.  All rights reserved.
;
;
; Use of this source code is subject to the terms of the Microsoft end-user
; license agreement (EULA) under which you licensed this SOFTWARE PRODUCT.
; If you did not accept the terms of the EULA, you are not authorized to use
; this source code. For a copy of the EULA, please see the LICENSE.RTF on your
; install media.
;
LIBRARY         CDCSERIAL
EXPORTS
                USBInstallDriver
                USBDeviceAttach
                USBUnInstallDriver
                COM_Init
                COM_Deinit
                COM_Open
                COM_Close
                COM_Read
                COM_Write
                COM_Seek
                COM_IOControl
                COM_PowerUp
                COM_PowerDown
```

- Open the file *CDCDriver.reg* and paste the content of the same file provided with the driver source code. Pay attention to change the following lines according to your needs:
    - `[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial\Init]`: this registry key holds the initialization commands for the modem. In this key there should be set the `AT+CGDCONT` command for configuring the context.

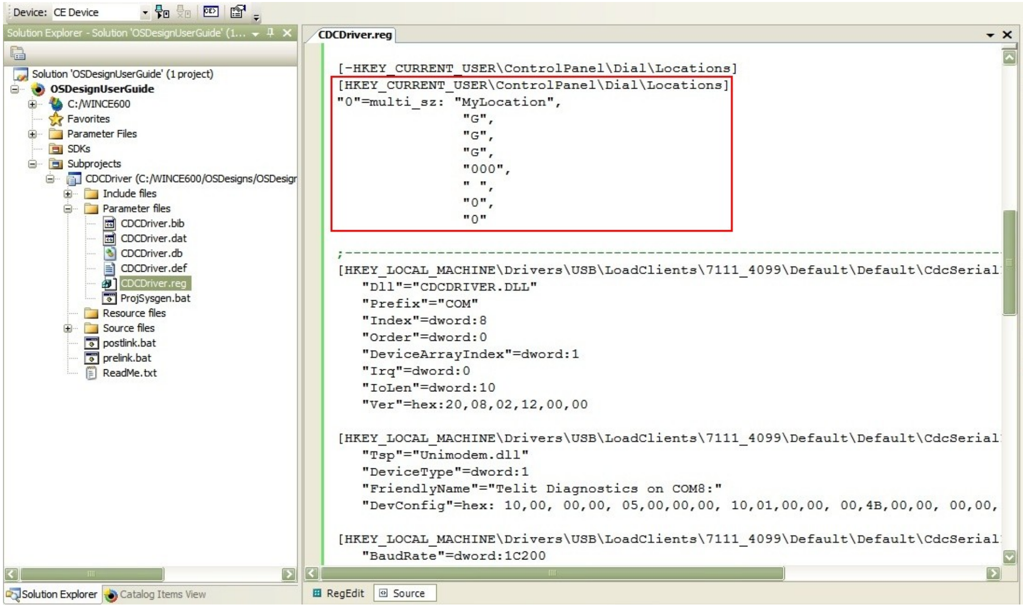


```
[HKEY_LOCAL_MACHINE\ExtModems]
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial]
    "Port"="COM7:"
    "DeviceType"=dword:1
    "FriendlyName"="Telit HSDPA Modem on COM7:"
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial\Init]
    "1"="AT+CGDCONT=1,IP,IBOX.TIM.IT<cr>"
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial\Settings]

[HKEY_LOCAL_MACHINE\Comm\CdcSerial]

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Dll"="CDCDRIVER.DLL"
   "Prefix"="COM"
   "Index"=dword:7
   "Order"=dword:0
   "DeviceArrayIndex"=dword:0
   "Irq"=dword:0
   "IoLen"=dword:10
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Tsp"="Unimodem.dll"
   "DeviceType"=dword:1
   "FriendlyName"="Telit HSDPA Modem on COM7:"
   "DevConfig"=hex: 10,00, 00,00, 05,00,00,00, 10,01,00,00, 00,4B,00,00, 00,00,
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "BaudRate"=dword:1C200
   "EnableFlowHard"=dword:0
   "EnableFlowSoft"=dword:0
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
    "1"="AT&F<cr>"
    "2"="ATE0V1&C1&D2<cr>"
```

    - `[HKEY_CURRENT_USER\ControlPanel\Dial\Locations]`: this registry key specifies a location for dialing.



```
[-HKEY_CURRENT_USER\ControlPanel\Dial\Locations]
[HKEY_CURRENT_USER\ControlPanel\Dial\Locations]
"0"=multi_sz: "MyLocation",
               "G",
               "G",
               "G",
               "000",
               " ",
               "0",
               "0"

;-----------------------------------------------------------------------------------
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Dll"="CDCDRIVER.DLL"
   "Prefix"="COM"
   "Index"=dword:8
   "Order"=dword:0
   "DeviceArrayIndex"=dword:1
   "Irq"=dword:0
   "IoLen"=dword:10
   "Ver"=hex:20,08,02,12,00,00

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Tsp"="Unimodem.dll"
   "DeviceType"=dword:1
   "FriendlyName"="Telit Diagnostics on COM8:"
   "DevConfig"=hex: 10,00, 00,00, 05,00,00,00, 10,01,00,00, 00,4B,00,00, 00,00,

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "BaudRate"=dword:1C200
```

- Open the file *sources* (it cannot be seen from the Visual Studio subproject tree, you can find it inside the project directory) and follow these steps:

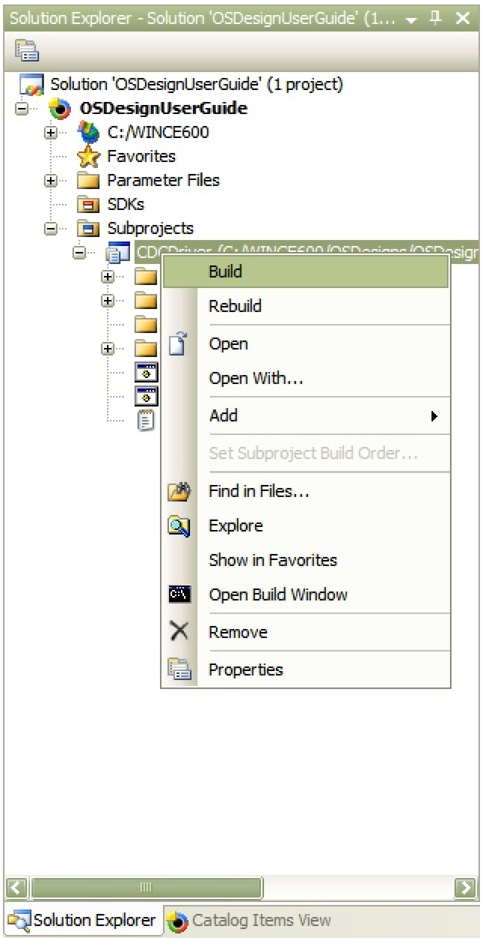
  - Add to the *TARGETLIBS* variable the following lines:

    ```
    $(_COMMONSDKROOT)\lib\$(_CPUINDPATH)\ntcompat.lib \
    $(_COMMONOAKROOT)\lib\$(_CPUINDPATH)\usbclient.lib \
    $(_SYSGENOAKROOT)\lib\$(_CPUINDPATH)\usbd.lib \
    ```

  - At the end of the file add the following lines:

    ```
    DLLENTRY=DllEntry
    INCLUDES= \
    $(_COMMONDDKROOT)\inc;$(_COMMONOAKROOT)\inc;..\..\common; \
    CDEFINES=$(CDEFINES) -DUSE_NEW_SERIAL_MODEL -DTELIT
    ```

- Right click on the name of the subproject and choose the item *Build*. After a successful building recreate the OS image.



Now you should have the CC864/UC864 driver integrated into your OS image. Start your system with the just created OS image and, after the boot, plug the device into your target: if the connection succeeds you should see the three virtual serial ports ready to be used.

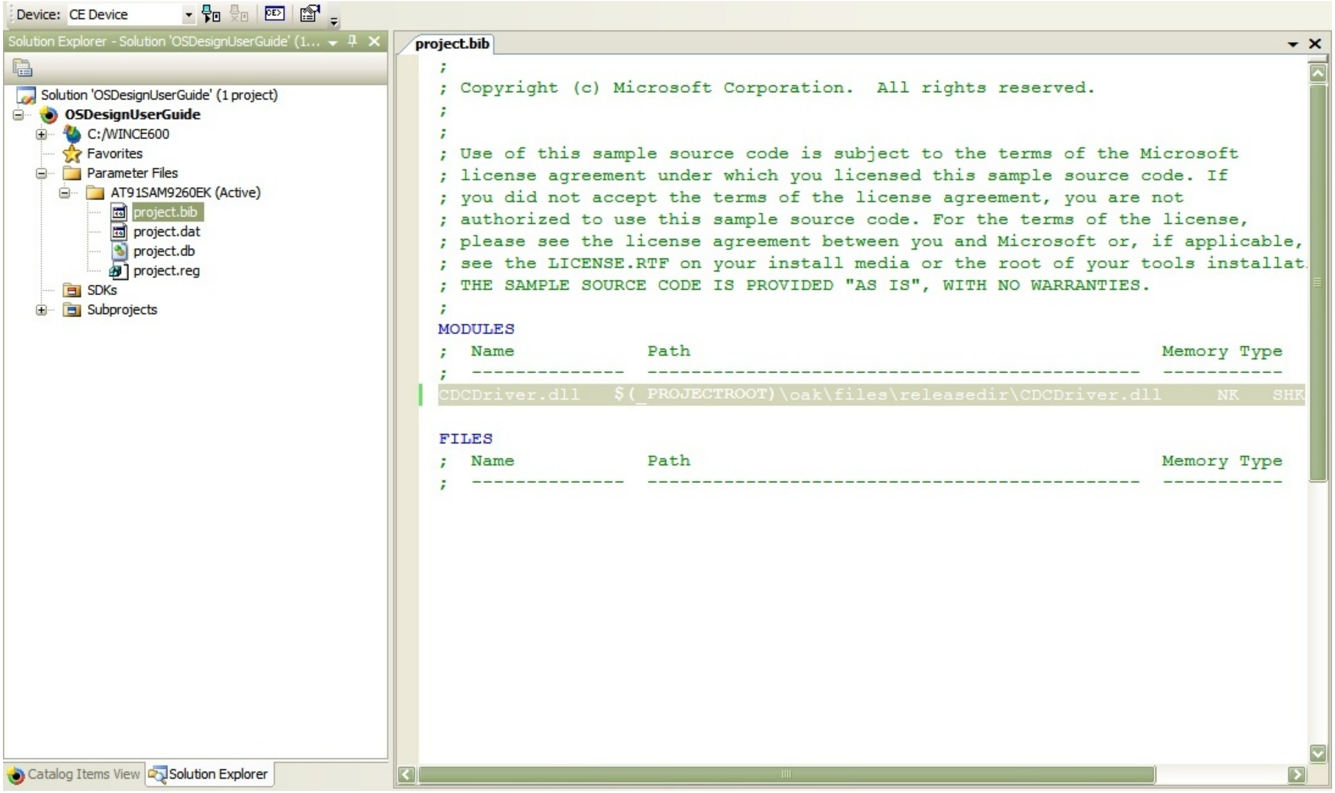
2.3.2. Using the binary

For adding CC864/UC864 support in the OS image, having the binary *CDCDriver.dll*, follow these steps:

- In the *Solution Explore*, under the folder *Parameter Files* open the folders called as your BSP name and open the file *project.bib*. Under the *Modules* section add the following line:

```
CDCDriver.dll
$(_PROJECTROOT)\[path_where_the_dll_is]\CDCDriver.dll      NK
      SHK
```

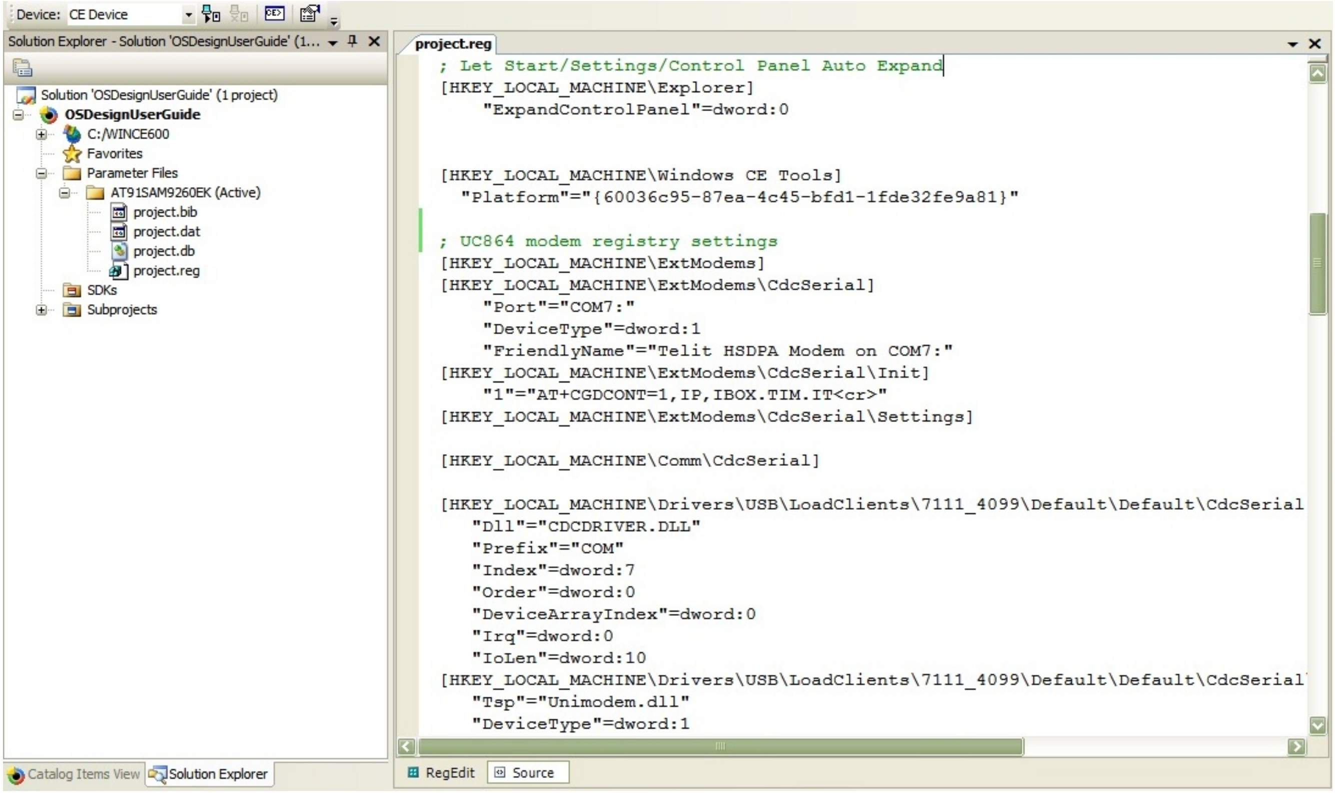
paying attention to change the path to the correct one (the place where the dll can be found).



```
;
; Copyright (c) Microsoft Corporation.  All rights reserved.
;
;
; Use of this sample source code is subject to the terms of the Microsoft
; license agreement under which you licensed this sample source code. If
; you did not accept the terms of the license agreement, you are not
; authorized to use this sample source code. For the terms of the license,
; please see the license agreement between you and Microsoft or, if applicable,
; see the LICENSE.RTF on your install media or the root of your tools installat
; THE SAMPLE SOURCE CODE IS PROVIDED "AS IS", WITH NO WARRANTIES.
;
MODULES
;  Name            Path                                           Memory Type
;  --------------  ---------------------------------------------  -----------
CDCDriver.dll   $(_PROJECTROOT)\oak\files\releasedir\CDCDriver.dll     NK    SHK

FILES
;  Name            Path                                           Memory Type
;  --------------  ---------------------------------------------  -----------
```

- Open the file *project.reg* and, at its end, paste the content of the file *CDCDriver.reg*.



```
; Let Start/Settings/Control Panel Auto Expand
[HKEY_LOCAL_MACHINE\Explorer]
    "ExpandControlPanel"=dword:0


[HKEY_LOCAL_MACHINE\Windows CE Tools]
  "Platform"="{60036c95-87ea-4c45-bfd1-1fde32fe9a81}"

; UC864 modem registry settings
[HKEY_LOCAL_MACHINE\ExtModems]
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial]
    "Port"="COM7:"
    "DeviceType"=dword:1
    "FriendlyName"="Telit HSDPA Modem on COM7:"
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial\Init]
    "1"="AT+CGDCONT=1,IP,IBOX.TIM.IT<cr>"
[HKEY_LOCAL_MACHINE\ExtModems\CdcSerial\Settings]

[HKEY_LOCAL_MACHINE\Comm\CdcSerial]

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Dll"="CDCDRIVER.DLL"
   "Prefix"="COM"
   "Index"=dword:7
   "Order"=dword:0
   "DeviceArrayIndex"=dword:0
   "Irq"=dword:0
   "IoLen"=dword:10
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\7111_4099\Default\Default\CdcSerial
   "Tsp"="Unimodem.dll"
   "DeviceType"=dword:1
```

- Recreate the OS image.

Now you should have the CC864/UC864 driver integrated into your OS image. Start your system with the just created OS image and, after the boot, plug the device into your target: if the connection succeeds you should see the three virtual serial ports described in paragraph 2.1 ready to be used.

3. Using the CC864/UC864

In this chapter it is explained how to programmatically use the CC864/UC864 (via the serial port API) and how to setup a PPP connection.

3.1. The serial API

Application can interact with the CC864/UC864 through the virtual serial ports created by the driver (for example COM7). Windows CE 6.0 has a complete API for dealing with serial ports; following you can find all the most important calls with code examples. Further information can be found in Microsoft Developer Network.

3.1.1. CreateFile

This function creates, opens, or truncates a file, COM port, device, service, or console. It returns a handle to access the object.

Header:

winbase.h

Library:

coredll.lib

Syntax:

```
HANDLE CreateFile(
  LPCTSTR lpFileName,
  DWORD dwDesiredAccess,
  DWORD dwShareMode,
  LPSECURITY_ATTRIBUTES lpSecurityAttributes,
  DWORD dwCreationDisposition,
  DWORD dwFlagsAndAttributes,
  HANDLE hTemplateFile
);
```

Parameters:

lpFileName

[in] Pointer to a null-terminated string that specifies the name of the object, such as file, COM port, disk device, or console, to create or open.

dwDesiredAccess

[in] Type of access to the object. An application can obtain read-only access, write-only access, read/write access, or device-query access.

dwShareMode

[in] Share mode for the object. If this parameter is set to zero, the object cannot be shared. Subsequent open operations on the object fail until the handle is closed.

This parameter can be set to one or more values.

lpSecurityAttributes

[in] Not used.

dwCreationDisposition

[in] Action to take on files that exist, and which action to take when files do not exist.

dwFlagsAndAttributes

[in] File attributes and flags for the file.

hTemplateFile

[in] Ignored; as a result, this function does not copy the extended attributes to the new file.

Return Value:

An open handle to the specified file indicates success. If the specified file exists before the function call and dwCreationDisposition is set to CREATE_ALWAYS or OPEN_ALWAYS, a call to GetLastError returns ERROR_ALREADY_EXISTS, even though the function has succeeded. If the file does not exist before the call, GetLastError

returns zero. INVALID_HANDLE_VALUE indicates failure. To get extended error information, call GetLastError.

Example:

```
HANDLE hModem;
hModem = CreateFile( TEXT("COM7:"),
                    GENERIC_READ | GENERIC_WRITE,
                    0,
                    NULL,
                    OPEN_EXISTING,
                    0,
                    NULL);
if (hModem == INVALID_HANDLE_VALUE)
   // error opening port; abort
```

Further information on the parameters' values can be found at http://msdn.microsoft.com/en-us/library/aa914735.aspx.

3.1.2. WriteFile

This function writes data to a file. WriteFile starts writing data to the file at the position indicated by the file pointer.

Header:

winbase.h

Library:

coredll.lib

Syntax:

```
BOOL WriteFile(
  HANDLE hFile,
  LPCVOID lpBuffer,
  DWORD nNumberOfBytesToWrite,
  LPDWORD lpNumberOfBytesWritten,
  LPOVERLAPPED lpOverlapped
);
```

Parameters:

hFile

[in] Handle to the file to be written to (returned by CreateFile). The file handle must have been created with GENERIC_WRITE access to the file.

lpBuffer

[in] Pointer to the buffer containing the data to write to the file.

nNumberOfBytesToWrite

[in] Number of bytes to write to the file.

lpNumberOfBytesWritten

[out] Pointer to the number of bytes written by this function call. **WriteFile** sets this value to zero before taking action or checking errors.

lpOverlapped

[in] Unsupported; set to NULL.

Return Value:

Nonzero indicates success. Zero indicates failure.

Example:

```
#define BYTES_TO_BE_WRITTEN 4
char atTest[]="AT\r\n";
DWORD written;
HANDLE hModem;
hModem = CreateFile( TEXT("COM7:"),
                    GENERIC_READ | GENERIC_WRITE,
                    0,
                    NULL,
                    OPEN_EXISTING,
                    0,
                    NULL);
if (hModem == INVALID_HANDLE_VALUE)
   // error opening port; abort
if (!WriteFile(hModem, atTest, BYTES_TO_BE_WRITTEN, &written,
NULL))
   // error writing  bytes; abort
```

Further information on the parameters' values can be found at http://msdn.microsoft.com/en-us/library/ms892380.aspx.

3.1.3. ReadFile

This function reads data from a file, starting at the position indicated by the file pointer.

Header:

winbase.h

Library:

coredll.lib

Syntax:

```
BOOL ReadFile(
  HANDLE hFile,
  LPVOID lpBuffer,
  DWORD nNumberOfBytesToRead,
  LPDWORD lpNumberOfBytesRead,
  LPOVERLAPPED lpOverlapped
);
```

Parameters:

hFile
[in] Handle to the file to be read. The file handle must have been created with GENERIC_READ access to the file. This parameter cannot be a socket handle.

lpBuffer
[out] Pointer to the buffer that receives the data read from the file.

nNumberOfBytesToRead
[in] Number of bytes to be read from the file.

lpNumberOfBytesRead
[out] Pointer to the number of bytes read. **ReadFile** sets this value to zero before doing taking action or checking errors.

lpOverlapped
[in] Unsupported; set to NULL.

Return Value:

Nonzero indicates success. Zero indicates failure.

Example:

```
#define BYTES_TO_BE_WRITTEN 4
#define BYTES_TO_BE_READ 9
char atTest[]="AT\r\n";
char atAnswer[10];
DWORD written;
DWORD read;
HANDLE hModem;
hModem = CreateFile( TEXT("COM7:"),
                    GENERIC_READ | GENERIC_WRITE,
                    0,
                    NULL,
                    OPEN_EXISTING,
                    0,
                    NULL);
if (hModem == INVALID_HANDLE_VALUE)
   // error opening port; abort
if (!WriteFile(hModem, atTest, BYTES_TO_BE_WRITTEN, &written,
NULL))
   // error writing bytes; abort
Sleep(500);
if (!ReadFile(hModem, atAnswer, BYTES_TO_BE_READ, &read, NULL))
   // error reading bytes; abort
```

Further information on the parameters' values can be found at http://msdn.microsoft.com/en-us/library/ms891445.aspx.

3.1.4. CloseHandle

This function closes an open object handle.

Header:

winbase.h

Library:

coredll.lib

Syntax:

```
BOOL CloseHandle(
  HANDLE hObject
);
```

Parameters:

hObject
[in] Handle to an open object.

Return value:

Nonzero indicates success. Zero indicates failure.

Example:

```
HANDLE hModem;
hModem = CreateFile( TEXT("COM7:"),
                    GENERIC_READ | GENERIC_WRITE,
                    0,
                    NULL,
                    OPEN_EXISTING,
                    0,
                    NULL);
if (hModem == INVALID_HANDLE_VALUE)
   // error opening port; abort
CloseHandle(hModem);
```

Further information can be found at http://msdn.microsoft.com/en-us/library/ms923946.aspx.

3.2. PPP Connection

There are several ways to setup a PPP connection for a Windows CE system; in this guide it is described one of the possible methods suitable for headless devices. For further instructions on how to create a subproject refer to MSDN (http://msdn.microsoft.com/en-us/library/aa913961.aspx).

- Open Visual Studio 2005 and load your Os Image Design solution.
- In the solution explorer create a Windows CE Console Application subproject with the source files provided by Microsoft in the directory, starting from your Windows CE root (typically WINCE600), \PUBLIC\COMMON\OAK\DRIVERS\NETSAMP\RASENTRY, compile it and add to

the OS image. This program adds an entry to the default RAS phonebook from information stored in a configuration file.

- Create the RAS phonebook configuration file (for example called telitUC864.ras). In the directory, starting from your Windows CE root (typically WINCE600), \PUBLIC\COMMON\OAK\DRIVERS\NETSAMP\RASENTRY, there is the file *rasentry.txt* that explains the rules for creating this kind of file. Following there is an example:

```
Name=Telit RAS
UseCountryAndAreaCodes=N
Phone=[Your provider phone number]
SpecificIpAddr=N
SpecificNameServers=N
DeviceType=modem
DeviceName=Telit HSDPA Modem on COM7:
SwCompression=N
IpHeaderCompression=N
DialModifier=[Custom AT commands if needed]
SpecificNameServers=Y
DnsAddr=[Your provider primary DNS address]
AltDnsAddr=[Your provider alternative DNS address]
```

- Create a Windows CE Console Application subproject with the source files provided by Microsoft in the directory, starting from your Windows CE root (typically WINCE600), \PUBLIC\COMMON\OAK\DRIVERS\NETSAMP\RASDIAL compile it and add to the OS image. This program setups the PPP connection according to the selected RAS phonebook entry.
- Start your target and, after the boot, plug the UC864. Upload the created RAS phonebook configuration file to the target (for example using the File Viewer tool that you can find in the menu *Target → Remote Tools → File Viewer*).
- Launch in the target the application *RASENTRY* with the path of the RAS phonebook configuration file as the first argument; you can run an application in your system by choosing the menu voice *Target → Run Programs* and selecting the desired application.
- Launch in the target the application *RASDIAL* with the name of the created RAS phonebook entry (Telit RAS in the above example) as the first argument.

If all is successful the PPP connection should be setup: you can launch the *ipconfig* application for checking the target ip address.