

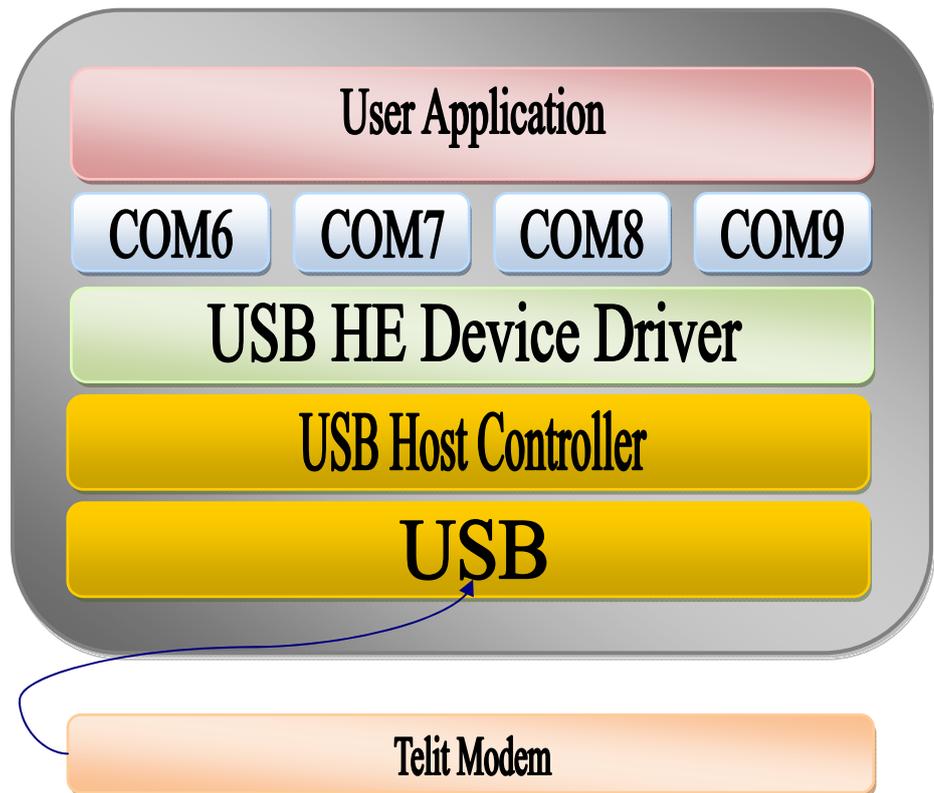
HE Windows CE 6.0 User Guide

1VV0300927 rev. 1 – 2011-11-03



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 HE module 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

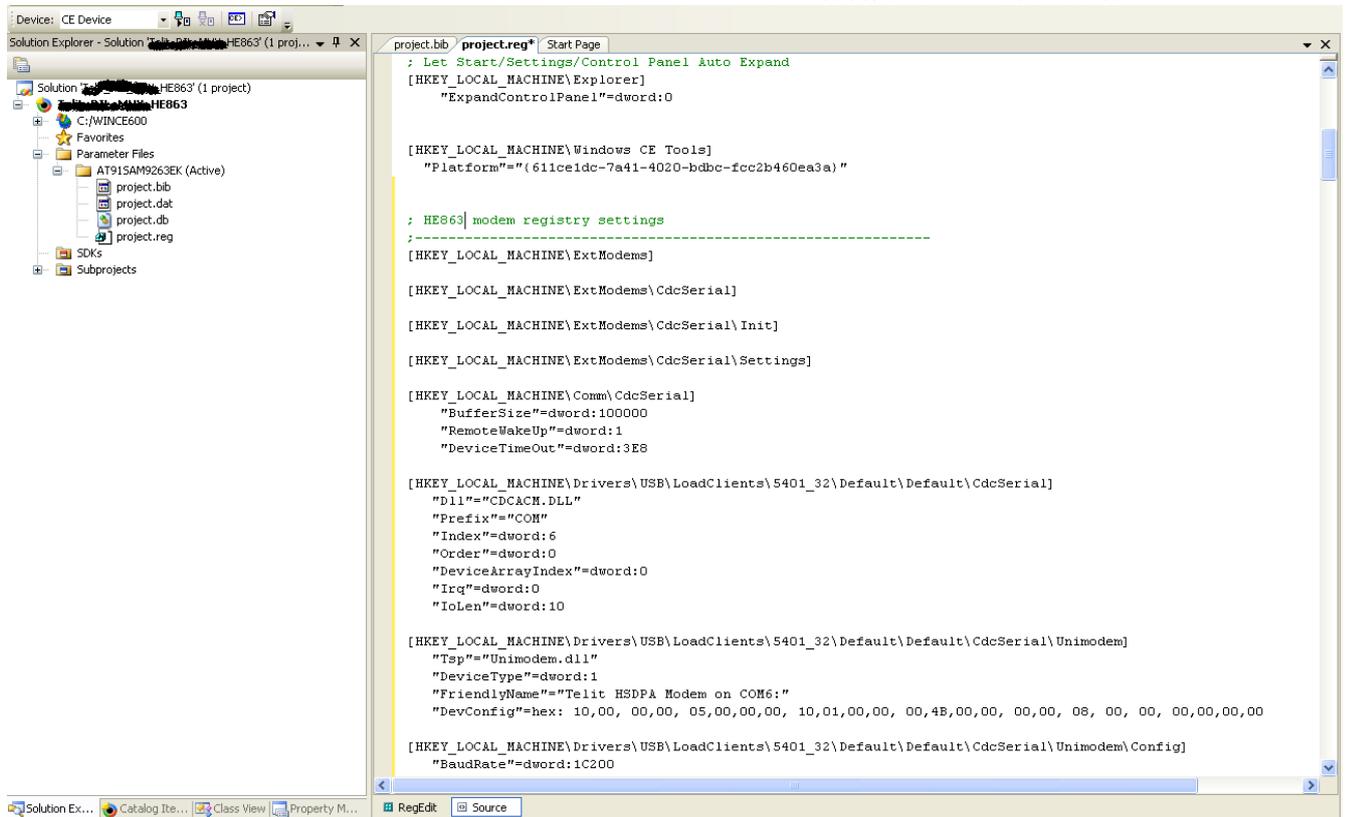


- USB HE Device Driver: piece of software that allows the HE to be seen by the user application as connected through serial ports rather than USB port.
- Virtual serial ports (COM6, COM7, COM8 and COM9): serial ports created by the HE device driver for accessing the modem; they can be used as real physical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

HE:

- COM6: Telit HSDPA modem port, used for normal modem/application interaction (e.g. AT commands sending, data connection...).
 - COM7: Telit on COM7.
 - COM8: Telit on COM8.
 - COM9: Telit Auxiliary, used for debugging purposes.
- User Application: piece of software written by the customer that uses HE features through virtual serial ports.





- Recreate the OS image.

Now you should have the HE 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 four virtual serial ports described in paragraph 2.1 ready to be used.



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("COM6:"),
    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.



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 telitHE863.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 COM6:
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 HE. 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.

